**📄 ApiController.cs (CS, 27 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using QuanLyBanHang.Models.APIModels;

3 │ using QuanLyBanHang.Models.QLBHEntity;

4 │

5 │ namespace QuanLyBanHang.Controllers

6 │ {

7 │ public abstract class ApiController : ControllerBase

8 │ {

9 │ private UserToken? \_objToken;

10 │ public UserToken? objToken

11 │ {

12 │ get => \_objToken;

13 │ set

14 │ {

15 │ \_objToken = value;

16 │ dbcontext.userId = \_objToken.userid;

17 │ }

18 │ }

19 │

20 │ public QuanLyBanHangContext dbcontext { get; }

21 │

22 │ public ApiController(QuanLyBanHangContext dbcontext)

23 │ {

24 │ this.dbcontext = dbcontext;

25 │ }

26 │ }

27 │ }

────────────────────────────────────────────────────────────

**📄 BanHangController.cs (CS, 1109 lines)**

1 │ ﻿using DeviceDetectorNET;

2 │ using Microsoft.AspNetCore.Mvc;

3 │ using Microsoft.AspNetCore.SignalR;

4 │ using Microsoft.EntityFrameworkCore;

5 │ using Microsoft.IdentityModel.Tokens;

6 │ using QuanLyBanHang.Common;

7 │ using QuanLyBanHang.Common.Attributes;

8 │ using QuanLyBanHang.Models.APIModels;

9 │ using QuanLyBanHang.Models.ExModel.CAT;

10 │ using QuanLyBanHang.Models.ExModel.OPS;

11 │ using QuanLyBanHang.Models.ExModel.StoredProcedureModel;

12 │ using QuanLyBanHang.Models.ExModel.SYS;

13 │ using QuanLyBanHang.Models.QLBHEntity;

14 │ using QuanLyBanHang.Models.VCEntity;

15 │ using QuanLyBanHang.Models.VCEntity.ExModel;

16 │ using QuanLyBanHang.Services;

17 │

18 │ namespace QuanLyBanHang.Controllers.OPS

19 │ {

20 │ public class BanHangController : BaseController

21 │ {

22 │ private QuanLyVoucherContext vcContext { get; set; }

23 │ private IEInvoiceService \_service;

24 │ private readonly IHubContext<ChatHUB> \_chatHub;

25 │ public BanHangController(QuanLyBanHangContext dbcontext, QuanLyVoucherContext voucherContext, IEInvoiceService s...

26 │ {

27 │ m\_MenuID = MenuID.opsBanHang.GetEnumGuid();

28 │ vcContext = voucherContext;

29 │ \_chatHub = chatHub;

30 │ \_service = service;

31 │ }

32 │ public class cHoaDonBan

33 │ {

34 │ public string? Result { get; set; }

35 │ public string? ResultText { get; set; }

36 │ }

37 │ public class cCT\_HoaDon

38 │ {

39 │ public string? Ma { get; set; }

40 │ public string? MaHangHoa { get; set; }

41 │ public string? MaVach { get; set; }

42 │ public string? SoLuong { get; set; }

43 │ public string? DonGia { get; set; }

44 │ public string? GiaGiam { get; set; }

45 │ public string? ThanhTien { get; set; }

46 │ public string? SoLuongTraLai { get; set; }

47 │ public string? ThanhTienTraLai { get; set; }

48 │

49 │ }

50 │ private bool IsMobile()

51 │ {

52 │ var userAgent = Request.Headers["User-Agent"].ToString();

53 │ if (string.IsNullOrEmpty(userAgent))

54 │ return false; // không có UA ⇒ coi như desktop

55 │

56 │ var dd = new DeviceDetector(userAgent);

57 │ dd.Parse();

58 │

59 │ return dd.IsMobile() || dd.IsTablet();

60 │ }

61 │ [TokenUserCheckHTTP]

62 │ public ActionResult Index(string MaKhoHang, string MaNganh)

63 │ {

64 │ if (IsMobile())

65 │ {

66 │ return RedirectToAction("mBanHang", new { MaKhoHang, MaNganh });

67 │ }

68 │ int MaKho = Cast.ToInt((MaKhoHang + "").Decrypt());

69 │ if (MaKho <= 0)

70 │ {

71 │ return RedirectToAction("ChonKho");

72 │ }

73 │ var user = new sys\_UserModel(dbcontext).GetItem(objToken.userid);

74 │ string[] strKhoHang = (user.KhoHang + "").Split(';');

75 │ if (!strKhoHang.Contains(Cast.ToInt(MaKho).ToString()))

76 │ {

77 │ return RedirectToAction("ChonKho");

78 │ }

79 │

80 │ DateTime NgayHD = new NgayKetCaModel(dbcontext).LayNgayKetCa(MaKho);

81 │

82 │ ViewData["ListHinhThuc\_TT"] = new HinhThucTTModel(dbcontext).GetData();

83 │ ViewData["ListGiamGia"] = new GiamGiaModel(dbcontext).GetData(objToken.branch);

84 │ ViewData["NgayHD"] = string.Format("{0:dd/MM/yyyy}", NgayHD);

85 │ ViewData["MaNganh"] = MaNganh;

86 │ ViewData["MaKho"] = MaKhoHang;

87 │ ViewData["TenKho"] = new DanhMucKhoModel(dbcontext).GetItem(MaKho).Ten;

88 │ return View();

89 │ }

90 │ [TokenUserCheckHTTP]

91 │ public ActionResult mBanHang(string MaKhoHang, string MaNganh)

92 │ {

93 │ if (!IsMobile())

94 │ {

95 │ return RedirectToAction("Index", new { MaKhoHang, MaNganh });

96 │ }

97 │ int MaKho = Cast.ToInt((MaKhoHang + "").Decrypt());

98 │ if (MaKho <= 0)

99 │ {

100 │ return RedirectToAction("ChonKho");

101 │ }

102 │ var user = new sys\_UserModel(dbcontext).GetItem(objToken.userid);

103 │ string[] strKhoHang = (user.KhoHang + "").Split(';');

104 │ if (!strKhoHang.Contains(Cast.ToInt(MaKho).ToString()))

105 │ {

106 │ return RedirectToAction("ChonKho");

107 │ }

108 │

109 │ DateTime NgayHD = new NgayKetCaModel(dbcontext).LayNgayKetCa(MaKho);

110 │

111 │ ViewData["NgayHD"] = string.Format("{0:dd/MM/yyyy}", NgayHD);

112 │ ViewData["MaNganh"] = MaNganh;

113 │ ViewData["MaKho"] = MaKhoHang.Decrypt();

114 │ return View();

115 │ }

116 │ [TokenUserCheckHTTP]

117 │ public ActionResult ChonKho()

118 │ {

119 │ var user = new sys\_UserModel(dbcontext).GetItem(objToken.userid);

120 │

121 │

122 │ string[] strKhoHang = (user.KhoHang + "").Split(';');

123 │ List<int> lstIDKhoHang = new List<int>();

124 │ foreach (string item in strKhoHang)

125 │ {

126 │ if (Cast.ToInt(item.Trim()) > 0)

127 │ {

128 │ lstIDKhoHang.Add(Cast.ToInt(item.Trim()));

129 │ }

130 │ }

131 │

132 │ ViewData["ListKhoHang"] = new DanhMucKhoModel(dbcontext).GetDataByIds(lstIDKhoHang, objToken.branch).Where(t...

133 │

134 │ return View("ChonKho");

135 │ }

136 │ [TokenUserCheckHTTP]

137 │ public ActionResult getPhuongThucThanhToan()

138 │ {

139 │ return Json(new ResponseModel(QLBHConst.OK, "", new HinhThucTTModel(dbcontext).GetData()));

140 │ }

141 │ [TokenUserCheckHTTP]

142 │ public ActionResult getLopHangTheoKho(string MaKhoHang)

143 │ {

144 │ int MaKho = Cast.ToInt((MaKhoHang + "").Decrypt());

145 │ return Json(new ResponseModel(QLBHConst.OK, "", new LopHangHoaModel(dbcontext).GetLopHangTheoKho(MaKho)));

146 │ }

147 │ [TokenUserCheckHTTP]

148 │ public ActionResult getOrderCount(string MaKhoHang)

149 │ {

150 │ int MaKho = Cast.ToInt((MaKhoHang + "").Decrypt());

151 │ var orderList = dbcontext.OpsOrderDetail.Where(x => x.MaKho == MaKho).ToList();

152 │ int waitingCount = orderList.Where(x => x.TrangThai == (int)OrderStatus.WaitConfirm).Count();

153 │ int confirmedCount = orderList.Where(x => x.TrangThai == (int)OrderStatus.Confirmed).Count();

154 │ return Json(new ResponseModel(QLBHConst.OK, "", new { waitingCount, confirmedCount }));

155 │ }

156 │ [TokenUserCheckHTTP]

157 │ public ActionResult Edit(string Ma, string TuNgay, string DenNgay, string NguoiBan, string MaKhoHang, string MaN...

158 │ {

159 │ int MaChiNhanhInt = dbcontext.SysBranch.First(x => x.Ma == objToken.branch).Nid ?? 0;

160 │ int MaChiNhanhVoucher = dbcontext.SysBranch.First(x => x.Ma == objToken.branch).MaChiNhanhVoucher ?? 0;

161 │ int MaKho = Cast.ToInt((MaKhoHang + "").Decrypt());

162 │ if (MaKho <= 0)

163 │ {

164 │ return RedirectToAction("ChonKho");

165 │ }

166 │ Models.QLBHEntity.SysUser user = new sys\_UserModel(dbcontext).GetItem(objToken.userid);

167 │ string[] strKhoHang = (user.KhoHang + "").Split(';');

168 │ if (!strKhoHang.Contains(Cast.ToInt(MaKho).ToString()))

169 │ {

170 │ return RedirectToAction("ChonKho");

171 │ }

172 │ OpsHoaDon item = new HoaDonBanModel(dbcontext, vcContext).GetItem(Cast.ToInt(Ma.Decrypt())) ?? new OpsHoaDon...

173 │ if (item.NgayHoaDon.Date != DateTime.Now.Date)

174 │ {

175 │ List<VPermission> ListPermission = GetVPermissions();

176 │ if (!ListPermission.Any(t => t.PermissionId == Permission.SuaHoaDonQuaNgay.GetEnumGuid()))

177 │ {

178 │ return Json(QLBHConst.ResponseNoPermission);

179 │ }

180 │ }

181 │

182 │ List<OpsChiTietHoaDon> ListChiTiet = new ChiTietHoaDonModel(dbcontext).GetData(item.Ma) ?? new List<OpsChiTi...

183 │

184 │ var voucherRs = new ThanhToanVoucherModel(vcContext).GetVoucher(item.MaVachThe ?? "", MaChiNhanhVoucher, Cas...

185 │ var voucher = voucherRs.Voucher;

186 │

187 │ ViewData["itemHoaDonBan"] = item;

188 │ ViewData["ListChiTiet"] = ListChiTiet;

189 │ ViewData["Voucher"] = voucher;

190 │ ViewData["ListHangHoa"] = new GiaBanTheoKhoModel(dbcontext).DanhSachTatCaTonKhoHangHoaVaGiaBanTheoKho(Cast.T...

191 │ ViewData["ListHinhThuc\_TT"] = new HinhThucTTModel(dbcontext).GetData();

192 │ ViewData["ListGiamGia"] = new GiamGiaModel(dbcontext).GetData(objToken.branch);

193 │ ViewData["TuNgay"] = TuNgay == "" ? string.Format("{0:dd/MM/yyyy}", DateTime.Now) : TuNgay;

194 │ ViewData["DenNgay"] = DenNgay == "" ? string.Format("{0:dd/MM/yyyy}", DateTime.Now) : DenNgay;

195 │ ViewData["KhoHang"] = new DanhMucKhoModel(dbcontext).GetItem(MaKho);

196 │ ViewData["NguoiBan"] = NguoiBan + "";

197 │ ViewData["TraHang"] = "0";

198 │ ViewData["MaNganh"] = MaNganh;

199 │ return View("Edit");

200 │ }

201 │ [TokenUserCheckHTTP()]

202 │ public ActionResult TraHang(string Ma, string TuNgay, string DenNgay, string NguoiBan, string MaKhoHang, string ...

203 │ {

204 │ int MaChiNhanhInt = dbcontext.SysBranch.First(x => x.Ma == objToken.branch).Nid ?? 0;

205 │ int MaChiNhanhVoucher = dbcontext.SysBranch.First(x => x.Ma == objToken.branch).MaChiNhanhVoucher ?? 0;

206 │ int MaKho = Cast.ToInt((MaKhoHang + "").Decrypt());

207 │ if (MaKho <= 0)

208 │ {

209 │ return RedirectToAction("ChonKho");

210 │ }

211 │ Models.QLBHEntity.SysUser user = new sys\_UserModel(dbcontext).GetItem(objToken.userid);

212 │ string[] strKhoHang = (user.KhoHang + "").Split(';');

213 │ if (!strKhoHang.Contains(Cast.ToInt(MaKho).ToString()))

214 │ {

215 │ return RedirectToAction("ChonKho");

216 │ }

217 │ OpsHoaDon item = new HoaDonBanModel(dbcontext, vcContext).GetItem(Cast.ToInt(Ma.Decrypt())) ?? new OpsHoaDon...

218 │ if (item.NgayHoaDon.Date != DateTime.Now.Date)

219 │ {

220 │ List<VPermission> ListPermission = GetVPermissions();

221 │ if (!ListPermission.Any(t => t.PermissionId == Permission.TraHangQuaNgay.GetEnumGuid()))

222 │ {

223 │ return Json(QLBHConst.ResponseNoPermission);

224 │ }

225 │ }

226 │

227 │ List<OpsChiTietHoaDon> ListChiTiet = new ChiTietHoaDonModel(dbcontext).GetData(item.Ma) ?? new List<OpsChiTi...

228 │ var voucherRs = new ThanhToanVoucherModel(vcContext).GetVoucher(item.MaVachThe ?? "", MaChiNhanhVoucher, Cas...

229 │ var voucher = voucherRs.Voucher;

230 │

231 │

232 │ ViewData["itemHoaDonBan"] = item;

233 │ ViewData["ListChiTiet"] = ListChiTiet;

234 │ ViewData["Voucher"] = voucher;

235 │ ViewData["ListHangHoa"] = new GiaBanTheoKhoModel(dbcontext).DanhSachTatCaTonKhoHangHoaVaGiaBanTheoKho(Cast.T...

236 │ ViewData["ListHinhThuc\_TT"] = new HinhThucTTModel(dbcontext).GetData();

237 │ ViewData["ListGiamGia"] = new GiamGiaModel(dbcontext).GetData(objToken.branch);

238 │ ViewData["TuNgay"] = TuNgay == "" ? string.Format("{0:dd/MM/yyyy}", DateTime.Now) : TuNgay;

239 │ ViewData["DenNgay"] = DenNgay == "" ? string.Format("{0:dd/MM/yyyy}", DateTime.Now) : DenNgay;

240 │ ViewData["KhoHang"] = new DanhMucKhoModel(dbcontext).GetItem(MaKho);

241 │ ViewData["NguoiBan"] = NguoiBan + "";

242 │ ViewData["TraHang"] = "1";

243 │ ViewData["MaNganh"] = MaNganh;

244 │ return View("Edit");

245 │ }

246 │ [HttpPost]

247 │ [TokenUserCheckHTTP]

248 │ public async Task<ActionResult> Update(string Ma, string HinhThuc\_TT, string GiamGia, string ThanhTien, string K...

249 │ {

250 │ int MaChiNhanhInt = dbcontext.SysBranch.First(x => x.Ma == objToken.branch).Nid ?? 0;

251 │ int MaChiNhanhVoucher = dbcontext.SysBranch.First(x => x.Ma == objToken.branch).MaChiNhanhVoucher ?? 0;

252 │ int MaKho;

253 │

254 │ if (ListChiTiet == null || ListChiTiet.Count == 0)

255 │ {

256 │ return Json(new ResponseModel(QLBHConst.Error, "Danh sách chi tiết không được trống", null));

257 │ }

258 │

259 │ // Xử lý tương tự như getListHangHoa và Ma

260 │ if (int.TryParse(MaKhoHang, out int directMaKhoValue))

261 │ {

262 │ // Nếu là số nguyên, sử dụng trực tiếp

263 │ MaKho = directMaKhoValue;

264 │ }

265 │ else

266 │ {

267 │ // Nếu không phải số nguyên, thử giải mã

268 │ try

269 │ {

270 │ MaKho = Cast.ToInt((MaKhoHang).Decrypt());

271 │ }

272 │ catch

273 │ {

274 │ // Nếu không thể giải mã, trả về lỗi

275 │ return Json(new ResponseModel(QLBHConst.Error, "Mã kho hàng không hợp lệ", null));

276 │ }

277 │ }

278 │

279 │

280 │ if (MaKho <= 0)

281 │ {

282 │ return RedirectToAction("ChonKho");

283 │ }

284 │ var user = new sys\_UserModel(dbcontext).GetItem(objToken.userid);

285 │ string[] strKhoHang = (user.KhoHang + "").Split(';');

286 │ if (!strKhoHang.Contains(Cast.ToInt(MaKho).ToString()))

287 │ {

288 │ return RedirectToAction("ChonKho");

289 │ }

290 │

291 │ int \_Ma;

292 │

293 │ // Xử lý tương tự như getListHangHoa

294 │ if (int.TryParse(Ma, out int directValue))

295 │ {

296 │ // Nếu là số nguyên, sử dụng trực tiếp

297 │ \_Ma = directValue;

298 │ }

299 │ else

300 │ {

301 │ // Nếu không phải số nguyên, thử giải mã

302 │ try

303 │ {

304 │ \_Ma = Cast.ToInt((Ma).Decrypt());

305 │ }

306 │ catch

307 │ {

308 │ // Nếu không thể giải mã, trả về lỗi

309 │ return Json(new ResponseModel(QLBHConst.Error, "Mã kho hàng không hợp lệ", null));

310 │ }

311 │ }

312 │

313 │ if (Cast.ToInt(HoanLai) < 0 && Cast.ToInt(HinhThuc\_TT) != (int)HinhThucThanhToan.AR && Cast.ToInt(HinhThuc\_T...

314 │ {

315 │ return Json(new ResponseModel(QLBHConst.Error, "Số tiền khách đưa không đủ"));

316 │ }

317 │ if (Cast.ToInt(HinhThuc\_TT) != (int)HinhThucThanhToan.TienMat && Cast.ToInt(HinhThuc\_TT) != (int)HinhThucTha...

318 │ && Cast.ToInt(HinhThuc\_TT) != (int)HinhThucThanhToan.The && Cast.ToInt(HinhThuc\_TT) != (int)HinhThucThan...

319 │ {

320 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng chọn hình thức thanh toán"));

321 │ }

322 │

323 │ if (!MaVachThe.IsNullOrEmpty())

324 │ {

325 │ var voucherRs = new ThanhToanVoucherModel(vcContext).GetVoucher(MaVachThe ?? "", MaChiNhanhVoucher, Cast...

326 │ var voucher = voucherRs.Voucher;

327 │ int? Result = voucherRs.Result;

328 │ if ((Result ?? 0) == 0)

329 │ {

330 │ return Json(new ResponseModel(QLBHConst.Error, "Thẻ không tồn tại"));

331 │ }

332 │ else if ((Result ?? 0) == -1)

333 │ {

334 │ return Json(new ResponseModel(QLBHConst.Error, "Thẻ không áp dụng tại chi nhánh này"));

335 │ }

336 │ else if ((Result ?? 0) == -2)

337 │ {

338 │ return Json(new ResponseModel(QLBHConst.Error, "Thẻ không áp dụng cho ngành này"));

339 │ }

340 │ else if ((Result ?? 0) == -3)

341 │ {

342 │ return Json(new ResponseModel(QLBHConst.Error, "Thẻ chưa kích hoạt"));

343 │ }

344 │ else if ((Result ?? 0) == -4)

345 │ {

346 │ return Json(new ResponseModel(QLBHConst.Error, "Thẻ chưa đến ngày sử dụng"));

347 │ }

348 │ else if ((Result ?? 0) == -5)

349 │ {

350 │ return Json(new ResponseModel(QLBHConst.Error, "Thẻ đã hết hạn sử dụng"));

351 │ }

352 │ }

353 │ if (Cast.ToInt(HinhThuc\_TT) == (int)HinhThucThanhToan.The && MaVachThe.IsEmpty())

354 │ {

355 │ return Json(new ResponseModel(QLBHConst.Error, "Voucher không hợp lệ"));

356 │ }

357 │ if (Cast.ToInt(HinhThuc\_TT) == (int)HinhThucThanhToan.ChuyenKhoan && RequestID.IsEmpty())

358 │ {

359 │ return Json(new ResponseModel(QLBHConst.Error, "Mã thanh toán không hợp lệ"));

360 │ }

361 │ List<VPermission> ListPermission = GetVPermissions();

362 │ if (Cast.ToInt(KhuyenMaiTienMat) > 0 && !ListPermission.Any(t => t.PermissionId == Permission.KhuyenMaiTienM...

363 │ {

364 │ return Json(new ResponseModel(QLBHConst.Error, "Bạn không có quyền nhập khuyến mãi tiền m��...

365 │ }

366 │

367 │ if (\_Ma == 0 && ListPermission.Any(t => t.PermissionId == Permission.Add.GetEnumGuid()))//Add

368 │ {

369 │ cHoaDonBan res = new HoaDonBanModel(dbcontext, vcContext).Add(objToken.branch, Cast.ToInt(HinhThuc\_TT), ...

370 │ if (res.Result == QLBHConst.OK)

371 │ {

372 │ var kho = dbcontext.StockDanhMucKho.FirstOrDefault(x => x.Ma == MaKho);

373 │

374 │ List<int> hoaDons = new List<int>

375 │ {

376 │ Cast.ToInt(res.ResultText)

377 │ };

378 │ TayNinhServiceBackgroundWorker.AddJob(new TayNinhServiceJob { ListMaHoaDon = hoaDons });

379 │ var res2 = GetThongTinIn(Cast.ToInt(res.ResultText));

380 │ return Json(new ResponseModel(QLBHConst.OK, QLBHConst.AddSuccessfully, new

381 │ {

382 │ hoadon = res2.hd,

383 │ cthoadon = res2.ct,

384 │ user = res2.usr

385 │ }));

386 │ }

387 │ else

388 │ {

389 │ return Json(new ResponseModel(res.Result, res.ResultText));

390 │ }

391 │ }

392 │ else if (\_Ma != 0 && ListPermission.Any(t => t.PermissionId == Permission.Edit.GetEnumGuid()))//Edit

393 │ {

394 │ OpsHoaDon item = new HoaDonBanModel(dbcontext, vcContext).GetItem(Cast.ToInt(Ma.Decrypt())) ?? new OpsHo...

395 │ if (item.NgayHoaDon.Date != DateTime.Now.Date)

396 │ {

397 │ if (!ListPermission.Any(t => t.PermissionId == Permission.SuaHoaDonQuaNgay.GetEnumGuid()))

398 │ {

399 │ return Json(QLBHConst.ResponseNoPermission);

400 │ }

401 │ }

402 │ cHoaDonBan res = new HoaDonBanModel(dbcontext, vcContext).Edit(\_Ma, objToken.branch, Cast.ToInt(HinhThuc...

403 │ return Json(new ResponseModel(res.Result, res.ResultText));

404 │ }

405 │ else

406 │ {

407 │ return Json(QLBHConst.ResponseNoPermission);

408 │ }

409 │ }

410 │ [HttpPost]

411 │ [TokenUserCheckHTTP]

412 │ public ActionResult getVoucher(string MaVachThe, string MaNganh)

413 │ {

414 │

415 │ int \_MaNganh;

416 │

417 │ if (int.TryParse(MaNganh, out int directValue))

418 │ {

419 │ // Nếu là số nguyên, sử dụng trực tiếp

420 │ \_MaNganh = directValue;

421 │ }

422 │ else

423 │ {

424 │ // Nếu không phải số nguyên, thử giải mã

425 │ try

426 │ {

427 │ \_MaNganh = Cast.ToInt((MaNganh).Decrypt());

428 │ }

429 │ catch

430 │ {

431 │ // Nếu không thể giải mã, trả về lỗi

432 │ return Json(new ResponseModel(QLBHConst.Error, "Mã kho hàng không hợp lệ", null));

433 │ }

434 │ }

435 │

436 │ DateTime dtNow = DateTime.Now;

437 │ int MaChiNhanhInt = dbcontext.SysBranch.First(x => x.Ma == objToken.branch).Nid ?? 0;

438 │ int MaChiNhanhVoucher = dbcontext.SysBranch.First(x => x.Ma == objToken.branch).MaChiNhanhVoucher ?? 0;

439 │ var voucherRs = new ThanhToanVoucherModel(vcContext).GetVoucher(MaVachThe ?? "", MaChiNhanhVoucher, \_MaNganh...

440 │ var voucher = voucherRs.Voucher;

441 │ int? Result = voucherRs.Result;

442 │ if ((Result ?? 0) == 0)

443 │ {

444 │ return Json(new ResponseModel(QLBHConst.Error, "Thẻ không tồn tại"));

445 │ }

446 │ else if ((Result ?? 0) == -1)

447 │ {

448 │ return Json(new ResponseModel(QLBHConst.Error, "Thẻ không áp dụng tại chi nhánh này"));

449 │ }

450 │ else if ((Result ?? 0) == -2)

451 │ {

452 │ return Json(new ResponseModel(QLBHConst.Error, "Thẻ không áp dụng cho ngành này"));

453 │ }

454 │ else if ((Result ?? 0) == -3)

455 │ {

456 │ return Json(new ResponseModel(QLBHConst.Error, "Thẻ chưa kích hoạt"));

457 │ }

458 │ else if ((Result ?? 0) == -4)

459 │ {

460 │ return Json(new ResponseModel(QLBHConst.Error, "Thẻ chưa đến ngày sử dụng"));

461 │ }

462 │ else if ((Result ?? 0) == -5)

463 │ {

464 │ return Json(new ResponseModel(QLBHConst.Error, "Thẻ đã hết hạn sử dụng"));

465 │ }

466 │ else

467 │ {

468 │ return Json(new ResponseModel(QLBHConst.OK, "", new { sodu = voucher.SoDu }));

469 │ }

470 │ }

471 │ [HttpPost]

472 │ [TokenUserCheckHTTP(Permission.Delete)]

473 │ public ActionResult Delete(string Ma)

474 │ {

475 │ Ma = Ma.Decrypt();

476 │ OpsHoaDon item = new HoaDonBanModel(dbcontext, vcContext).GetItem(Cast.ToInt(Ma.Decrypt())) ?? new OpsHoaDon...

477 │ if (item == null)

478 │ {

479 │ return Json(QLBHConst.ResponseNotExist);

480 │ }

481 │ if (item.NgayHoaDon.Date != DateTime.Now.Date)

482 │ {

483 │ List<VPermission> ListPermission = GetVPermissions();

484 │ if (!ListPermission.Any(t => t.PermissionId == Permission.SuaHoaDonQuaNgay.GetEnumGuid()))

485 │ {

486 │ return Json(QLBHConst.ResponseNoPermission);

487 │ }

488 │ }

489 │

490 │ int res = new HoaDonBanModel(dbcontext, vcContext).Delete(Cast.ToInt(Ma), objToken.userid);

491 │ if (res == -2)

492 │ {

493 │ return Json(QLBHConst.ResponseNotExist);

494 │ }

495 │ else if (res == -3)

496 │ {

497 │ return Json(new ResponseModel(QLBHConst.Error, "Hóa đơn đã khóa sổ, không thể xóa"));

498 │ }

499 │ else if (res == -4)

500 │ {

501 │ return Json(new ResponseModel(QLBHConst.Error, "Lỗi xóa chi tiết hóa đơn"));

502 │ }

503 │ else if (res == -5)

504 │ {

505 │ return Json(new ResponseModel(QLBHConst.Error, "Lỗi xóa thanh toán thẻ"));

506 │ }

507 │ else if (res == 1)

508 │ {

509 │ return Json(QLBHConst.ResponseDeleteSuccessfully);

510 │ }

511 │ else

512 │ {

513 │ return Json(QLBHConst.ResponseHaveError);

514 │ }

515 │ }

516 │ [HttpPost]

517 │ [TokenUserCheckHTTP]

518 │ public ActionResult KhoaSo(string Ma, string Value)

519 │ {

520 │ List<VPermission> ListPermission = GetVPermissions();

521 │ if (Value == "1")

522 │ {

523 │ if (ListPermission.Any(t => t.PermissionId == Permission.KhoaSo.GetEnumGuid()))

524 │ {

525 │ int res = new HoaDonBanModel(dbcontext, vcContext).KhoaSo(Cast.ToInt(Ma.Decrypt()), objToken.userid,...

526 │ if (res == 1)

527 │ {

528 │ return Json(new ResponseModel(QLBHConst.OK, "Khóa sổ thành công"));

529 │ }

530 │ else if (res == -2)

531 │ {

532 │ return Json(QLBHConst.ResponseNotExist);

533 │ }

534 │ else if (res == -3)

535 │ {

536 │ return Json(new ResponseModel(QLBHConst.Error, "Đã khóa sổ bởi người khác"));

537 │ }

538 │ else

539 │ {

540 │ return Json(QLBHConst.ResponseHaveError);

541 │ }

542 │ }

543 │ else

544 │ {

545 │ return Json(QLBHConst.ResponseNoPermission);

546 │ }

547 │ }

548 │ else

549 │ {

550 │ if (ListPermission.Any(t => t.PermissionId == Permission.MoKhoaSo.GetEnumGuid()))

551 │ {

552 │ int res = new HoaDonBanModel(dbcontext, vcContext).KhoaSo(Cast.ToInt(Ma.Decrypt()), objToken.userid,...

553 │ if (res == 1)

554 │ {

555 │ return Json(new ResponseModel(QLBHConst.OK, "Mở khóa sổ thành công"));

556 │ }

557 │ else if (res == -2)

558 │ {

559 │ return Json(QLBHConst.ResponseNotExist);

560 │ }

561 │ else if (res == -3)

562 │ {

563 │ return Json(new ResponseModel(QLBHConst.Error, "Đã mở khóa sổ bởi người khác"));

564 │ }

565 │ else

566 │ {

567 │ return Json(QLBHConst.ResponseHaveError);

568 │ }

569 │ }

570 │ else

571 │ {

572 │ return Json(QLBHConst.ResponseNoPermission);

573 │ }

574 │ }

575 │ }

576 │ [HttpPost]

577 │ [TokenUserCheckHTTP(Permission.TraHang)]

578 │ public ActionResult TraHang(string Ma, string TienTraLaiKhach, string TienTraLaiThe, string GhiChu, List<cCT\_Hoa...

579 │ {

580 │ int \_Ma = Cast.ToInt(Ma.Decrypt());

581 │ OpsHoaDon item = new HoaDonBanModel(dbcontext, vcContext).GetItem(\_Ma);

582 │ if (item == null)

583 │ {

584 │ return Json(QLBHConst.ResponseNotExist);

585 │ }

586 │ int iTienTraLaiKhach = Cast.ToInt(TienTraLaiKhach);

587 │ int iTienTraLaiThe = Cast.ToInt(TienTraLaiThe);

588 │ int iTongTien = 0;

589 │ foreach (cCT\_HoaDon ct in ListHangTra)

590 │ {

591 │ if (Cast.ToInt(ct.SoLuong) < Cast.ToInt(ct.SoLuongTraLai))

592 │ {

593 │ return Json(new ResponseModel(QLBHConst.Error, "Số lượng trả lại vượt quá số lượng...

594 │ }

595 │ if (Cast.ToInt(ct.SoLuongTraLai) == 0)

596 │ {

597 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng nhập số lượng trả cho hàng hóa ...

598 │ }

599 │ iTongTien += Cast.ToInt(ct.ThanhTienTraLai);

600 │ }

601 │ if (iTongTien != (iTienTraLaiKhach + iTienTraLaiThe))

602 │ {

603 │ return Json(new ResponseModel(QLBHConst.Error, "Tổng tiền tính sai. Vui lòng thử lại"));

604 │ }

605 │ if (item.NgayHoaDon.Date != DateTime.Now.Date)

606 │ {

607 │ List<VPermission> ListPermission = GetVPermissions();

608 │ if (!ListPermission.Any(t => t.PermissionId == Permission.TraHangQuaNgay.GetEnumGuid()))

609 │ {

610 │ return Json(new ResponseModel(QLBHConst.Error, "Bạn không có quyền trả hàng qua ngày"));

611 │ }

612 │ }

613 │

614 │ try

615 │ {

616 │ cHoaDonBan res = new HoaDonBanModel(dbcontext, vcContext).TraHang(\_Ma, iTienTraLaiKhach, iTienTraLaiThe,...

617 │ if (res.Result == QLBHConst.OK)

618 │ {

619 │ if (item.NgayHoaDon.Date != DateTime.Now.Date)

620 │ {

621 │ TonKhoBackgroundWorker.AddJob(new TonKhoJob

622 │ {

623 │ NamThang = string.Format("{0:yyyyMM}", DateTime.Now),

624 │ MaKho = Cast.ToInt(item.MaKho),

625 │ UserId = objToken.userid,

626 │ MaChiNhanh = objToken.branch

627 │ });

628 │ }

629 │ else

630 │ {

631 │ TonKhoBackgroundWorker.AddJob(new TonKhoJob

632 │ {

633 │ NamThang = string.Format("{0:yyyyMM}", item.NgayHoaDon),

634 │ MaKho = Cast.ToInt(item.MaKho),

635 │ UserId = objToken.userid,

636 │ MaChiNhanh = objToken.branch

637 │ });

638 │ }

639 │ }

640 │ return Json(new ResponseModel(res.Result, res.ResultText));

641 │ }

642 │ catch (Exception ex)

643 │ {

644 │ return Json(new ResponseModel(QLBHConst.Error, "Có lỗi xảy ra: " + ex.Message));

645 │ }

646 │ }

647 │ [HttpPost]

648 │ [TokenUserCheckHTTP]

649 │ public ActionResult KetCa(string MaKhoHang)

650 │ {

651 │ int MaKho;

652 │

653 │ if (int.TryParse(MaKhoHang, out int directValue))

654 │ {

655 │ // Nếu là số nguyên, sử dụng trực tiếp

656 │ MaKho = directValue;

657 │ }

658 │ else

659 │ {

660 │ // Nếu không phải số nguyên, thử giải mã

661 │ try

662 │ {

663 │ MaKho = Cast.ToInt((MaKhoHang).Decrypt());

664 │ }

665 │ catch

666 │ {

667 │ // Nếu không thể giải mã, trả về lỗi

668 │ return Json(new ResponseModel(QLBHConst.Error, "Mã kho hàng không hợp lệ", null));

669 │ }

670 │ }

671 │ //int MaKho = Cast.ToInt(MaKhoHang.Decrypt());

672 │ if (MaKho != 0)

673 │ {

674 │ decimal SoTien = 0;

675 │ int res = new NgayKetCaModel(dbcontext).KetCa(MaKho, DateTime.Now, objToken.userid, ref SoTien);

676 │ if (res == 1)

677 │ {

678 │ return Json(new ResponseModel(QLBHConst.OK, "Kết ca thành công.\nDoanh thu: " + string.Format("{...

679 │ }

680 │ else

681 │ {

682 │ return Json(new ResponseModel(QLBHConst.Error, "Kết ca thất bại"));

683 │ }

684 │ }

685 │ return Json(new ResponseModel(QLBHConst.Error, "Không tìm thấy kho hàng"));

686 │ }

687 │ [HttpPost]

688 │ [TokenUserCheckHTTP]

689 │ public ActionResult GetThongTinInAPI(string Ma)

690 │ {

691 │

692 │ int maHoaDon;

693 │

694 │ // Thử chuyển đổi trực tiếp nếu là số nguyên

695 │ if (int.TryParse(Ma, out maHoaDon))

696 │ {

697 │ // Nếu chuyển đổi thành công, sử dụng giá trị này

698 │ }

699 │ else

700 │ {

701 │ // Nếu không phải số nguyên, giả định nó đã được mã hóa

702 │ try

703 │ {

704 │ maHoaDon = Cast.ToInt(Ma.Decrypt());

705 │ }

706 │ catch

707 │ {

708 │ return Json(new ResponseModel(QLBHConst.Error, "Mã hóa đơn không hợp lệ", null));

709 │ }

710 │ }

711 │ var res = GetThongTinIn(maHoaDon);

712 │ return Json(new ResponseModel(QLBHConst.OK, "", new

713 │ {

714 │ hoadon = res.hd,

715 │ cthoadon = res.ct,

716 │ user = res.usr

717 │ }));

718 │ }

719 │

720 │ [HttpPost]

721 │ [TokenUserCheckHTTP]

722 │ public ActionResult GetThongTinInHoaDonTraAPI(string Ma)

723 │ {

724 │ var item = new HoaDonBanModel(dbcontext, vcContext).GetItem(Cast.ToInt(Ma.Decrypt())) ?? new OpsHoaDon();

725 │ List<sp\_ChiTietHoaDonTraResult> ListChitiet = new ChiTietHoaDonModel(dbcontext).GetFullDataTra(item.Ma);

726 │ var usr = new sys\_UserModel(dbcontext).GetItem(objToken.userid);

727 │ return Json(new ResponseModel(QLBHConst.OK, "", new

728 │ {

729 │ hoadon = item,

730 │ cthoadon = ListChitiet,

731 │ user = usr

732 │ }));

733 │ }

734 │ [TokenUserCheckHTTP(Permission.InLaiHoaDon)]

735 │ public ActionResult InHoaDon(string Ma, string FirstPrint)

736 │ {

737 │ OpsHoaDon item = new HoaDonBanModel(dbcontext, vcContext).GetItem(Cast.ToInt(Ma.Decrypt())) ?? new OpsHoaDon...

738 │ List<sp\_ChiTietHoaDonResult> ListChitiet = new ChiTietHoaDonModel(dbcontext).GetFullData(item.Ma);

739 │ Models.QLBHEntity.SysUser usr = new sys\_UserModel(dbcontext).GetItem(objToken.userid);

740 │ ViewData["itemHoaDonBan"] = item;

741 │ ViewData["ListChitiet"] = ListChitiet;

742 │ ViewData["NhanVien"] = usr.FullName;

743 │ return View("InHoaDon");

744 │ }

745 │ [TokenUserCheckHTTP(Permission.InLaiHoaDon)]

746 │ public ActionResult InHoaDonTra(string Ma)

747 │ {

748 │ OpsHoaDon item = new HoaDonBanModel(dbcontext, vcContext).GetItem(Cast.ToInt(Ma.Decrypt())) ?? new OpsHoaDon...

749 │ List<sp\_ChiTietHoaDonResult> ListChitiet = new ChiTietHoaDonModel(dbcontext).GetFullData(item.Ma);

750 │ Models.QLBHEntity.SysUser usr = new sys\_UserModel(dbcontext).GetItem(objToken.userid);

751 │ ViewData["itemHoaDonBan"] = item;

752 │ ViewData["ListChitiet"] = ListChitiet;

753 │ ViewData["NhanVien"] = usr.FullName;

754 │ return View("InHoaDonTra");

755 │ }

756 │ [TokenUserCheckHTTP]

757 │ public ActionResult Preview(string TuNgay, string DenNgay, string NguoiBan, string MaKhoHang, string ChayTonKho,...

758 │ {

759 │ int MaKho = Cast.ToInt((MaKhoHang + "").Decrypt());

760 │ if (MaKho <= 0)

761 │ {

762 │ return RedirectToAction("ChonKho");

763 │ }

764 │ Models.QLBHEntity.SysUser user = new sys\_UserModel(dbcontext).GetItem(objToken.userid);

765 │ string[] strKhoHang = (user.KhoHang + "").Split(';');

766 │ if (!strKhoHang.Contains(Cast.ToInt(MaKho).ToString()))

767 │ {

768 │ return RedirectToAction("ChonKho");

769 │ }

770 │ if (NguoiBan == null)

771 │ {

772 │ NguoiBan = objToken.userid + "";

773 │ }

774 │ DateTime CurrDate = DateTime.Now;

775 │

776 │ DateTime dtpTuNgay = TuNgay.ToDateTime2(DateTime.Now)!.Value;

777 │ DateTime dtpDenNgay = DenNgay.ToDateTime2(DateTime.Now)!.Value;

778 │

779 │ ViewData["TuNgay"] = TuNgay;

780 │ ViewData["DenNgay"] = DenNgay;

781 │ ViewData["NguoiBan"] = NguoiBan;

782 │ ViewData["KhoHang"] = new DanhMucKhoModel(dbcontext).GetItem(MaKho);

783 │ ViewData["ListNguoiBan"] = new sys\_UserModel(dbcontext).GetData();

784 │ ViewData["MaNganh"] = MaNganh;

785 │

786 │ return View("Preview");

787 │ }

788 │ [HttpPost]

789 │ [TokenUserCheckHTTP]

790 │ public ActionResult getDanhSachHoaDon(string TuNgay, string DenNgay, string MaNhanVien, string MaKhoHang)

791 │ {

792 │ int MaKho;

793 │

794 │ // Xử lý tương tự như getListHangHoa và Ma

795 │ if (int.TryParse(MaKhoHang, out int directMaKhoValue))

796 │ {

797 │ // Nếu là số nguyên, sử dụng trực tiếp

798 │ MaKho = directMaKhoValue;

799 │ }

800 │ else

801 │ {

802 │ // Nếu không phải số nguyên, thử giải mã

803 │ try

804 │ {

805 │ MaKho = Cast.ToInt((MaKhoHang).Decrypt());

806 │ }

807 │ catch

808 │ {

809 │ // Nếu không thể giải mã, trả về lỗi

810 │ return Json(new ResponseModel(QLBHConst.Error, "Mã kho hàng không hợp lệ", null));

811 │ }

812 │ }

813 │

814 │ Models.QLBHEntity.SysUser user = new sys\_UserModel(dbcontext).GetItem(objToken.userid);

815 │ string[] strKhoHang = (user.KhoHang + "").Split(';');

816 │

817 │

818 │ DateTime CurrDate = DateTime.Now;

819 │

820 │ DateTime dtpTuNgay = TuNgay.ToDateTime2(DateTime.Now)!.Value;

821 │ DateTime dtpDenNgay = DenNgay.ToDateTime2(DateTime.Now)!.Value;

822 │

823 │ var listData = new HoaDonBanModel(dbcontext, vcContext).GetData(dtpTuNgay, dtpDenNgay, objToken.branch, MaKh...

824 │ .Select(t => new

825 │ {

826 │ HoaDon\_ID = t.Ma.Encrypt(),

827 │ Ngay\_HD = string.Format("{0:dd/MM/yyyy}", t.NgayHoaDon),

828 │ So\_Seri = t.SoSeri + "",

829 │ NguoiBan = t.NguoiBan + "",

830 │ TenHinhThuc = t.TenHinhThuc + "",

831 │ Trigia = t.Trigia,

832 │ GiaGiam = t.GiaGiam,

833 │ NguoiBanID = t.NguoiThanhToan + "",

834 │ KhoaSo = (t.KhoaSo ?? false) ? 1 : 0,

835 │ MaKho = t.MaKho + "",

836 │ TenKho = t.TenKho + "",

837 │ TraHang = (t.TraHang ?? false) ? 1 : 0,

838 │ HoaDonTrongNgay = (t.NgayHoaDon.Date == DateTime.Today.Date) ? 1 : 0,

839 │ KhuyenMaiTienMat = t.KhuyenMaiTienMat + "",

840 │ HinhThucTT = t.HinhThucTT + "",

841 │ RequestID = t.RequestID + "",

842 │ WebOrderId = t.WebOrderId + ""

843 │ }).ToList();

844 │ return Json(new ResponseModel(QLBHConst.OK, "", listData));

845 │ }

846 │ [HttpPost]

847 │ [TokenUserCheckHTTP]

848 │ public ActionResult getListHangHoa(string MaKhoHang)

849 │ {

850 │ int MaKho;

851 │

852 │ // Kiểm tra xem chuỗi đầu vào có phải là số nguyên không

853 │ if (int.TryParse(MaKhoHang, out int directValue))

854 │ {

855 │ // Nếu là số nguyên, sử dụng trực tiếp

856 │ MaKho = directValue;

857 │ }

858 │ else

859 │ {

860 │ // Nếu không phải số nguyên, thử giải mã

861 │ try

862 │ {

863 │ MaKho = Cast.ToInt((MaKhoHang).Decrypt());

864 │ }

865 │ catch

866 │ {

867 │ // Nếu không thể giải mã, trả về lỗi

868 │ return Json(new ResponseModel(QLBHConst.Error, "Mã kho hàng không hợp lệ", null));

869 │ }

870 │ }

871 │

872 │ DateTime NgayHD = new NgayKetCaModel(dbcontext).LayNgayKetCa(MaKho);

873 │ return Json(new ResponseModel(QLBHConst.OK, "", new GiaBanTheoKhoModel(dbcontext).DanhSachTonKhoHangHoaVaGia...

874 │ }

875 │

876 │ public (OpsHoaDon hd, List<sp\_ChiTietHoaDonResult> ct, Models.QLBHEntity.SysUser usr) GetThongTinIn(int MaHoaDon...

877 │ {

878 │ OpsHoaDon item = new HoaDonBanModel(dbcontext, vcContext).GetItem(MaHoaDon) ?? new OpsHoaDon();

879 │ List<sp\_ChiTietHoaDonResult> ListChitiet = new ChiTietHoaDonModel(dbcontext).GetFullData(item.Ma);

880 │ var usr = new sys\_UserModel(dbcontext).GetItem(objToken?.userid);

881 │ return (item, ListChitiet, usr);

882 │ }

883 │

884 │

885 │ [HttpPost]

886 │ [TokenUserCheckHTTP(Permission.NotCheck)]

887 │ public async Task<IActionResult> GenVietQr(string sotien)

888 │ {

889 │ string billnumber = QuanLyBanHang.Common.Utility.Generate12CharCode();

890 │ string noidung = "Thanh toan Tanione";

891 │ var service = new VietinBankService(dbcontext, false); //test

892 │ //var service = new VietinBankService(dbcontext, false);

893 │ try

894 │ {

895 │ string QRData = await service.GenQRByVietQR(sotien, billnumber, noidung);

896 │ string vac = service.GetVacCode();

897 │ return Json(new ResponseModelSuccess("Tạo QR thành công", new

898 │ {

899 │ QRData = QRData,

900 │ SoHoaDon = vac + billnumber,

901 │ NgayIn = DateTime.Now.ToString("dd/MM/yyyy HH:mm"),

902 │ SoTien = sotien,

903 │ SoHoaDonKhachHang = noidung,

904 │ }));

905 │ }

906 │ catch (Exception ex)

907 │ {

908 │ return Json(new ResponseModelError(ex.Message));

909 │ }

910 │ }

911 │

912 │ [HttpGet]

913 │ [TokenUserCheckHTTP]

914 │ public ActionResult ChonKhoApp()

915 │ {

916 │ var user = new sys\_UserModel(dbcontext).GetItem(objToken.userid);

917 │

918 │ string[] strKhoHang = (user.KhoHang + "").Split(';');

919 │ List<int> lstIDKhoHang = new List<int>();

920 │ foreach (string item in strKhoHang)

921 │ {

922 │ if (Cast.ToInt(item.Trim()) > 0)

923 │ {

924 │ lstIDKhoHang.Add(Cast.ToInt(item.Trim()));

925 │ }

926 │ }

927 │

928 │ // Lấy danh sách kho hàng giống như trong action ChonKho gốc

929 │ var listKho = new DanhMucKhoModel(dbcontext)

930 │ .GetDataByIds(lstIDKhoHang, objToken.branch)

931 │ .Where(t => t.BanHang ?? false)

932 │ .ToList(); // Trả về toàn bộ đối tượng StockDanhMucKho

933 │

934 │ return Json(new ResponseModel(QLBHConst.OK, "", listKho));

935 │ }

936 │

937 │ [HttpPost]

938 │ [TokenUserCheckHTTP()]

939 │ public ActionResult TraHangAPI(string Ma, string TuNgay, string DenNgay, string NguoiBan, string MaKhoHang, stri...

940 │ {

941 │ try

942 │ {

943 │ int MaChiNhanhInt = dbcontext.SysBranch.First(x => x.Ma == objToken.branch).Nid ?? 0;

944 │ int MaChiNhanhVoucher = dbcontext.SysBranch.First(x => x.Ma == objToken.branch).MaChiNhanhVoucher ?? 0;

945 │ // FIX: Xử lý MaKhoHang giống như trong Update method

946 │ int MaKho;

947 │ if (int.TryParse(MaKhoHang, out int directMaKhoValue))

948 │ {

949 │ // Nếu là số nguyên, sử dụng trực tiếp

950 │ MaKho = directMaKhoValue;

951 │ }

952 │ else

953 │ {

954 │ // Nếu không phải số nguyên, thử giải mã

955 │ try

956 │ {

957 │ MaKho = Cast.ToInt((MaKhoHang).Decrypt());

958 │ }

959 │ catch

960 │ {

961 │ return Json(new ResponseModel(QLBHConst.Error, "Mã kho hàng không hợp lệ", null));

962 │ }

963 │ }

964 │

965 │ if (MaKho <= 0)

966 │ {

967 │ return Json(new ResponseModel(QLBHConst.Error, "Mã kho không hợp lệ", null));

968 │ }

969 │

970 │ Models.QLBHEntity.SysUser user = new sys\_UserModel(dbcontext).GetItem(objToken.userid);

971 │ string[] strKhoHang = (user.KhoHang + "").Split(';');

972 │ if (!strKhoHang.Contains(Cast.ToInt(MaKho).ToString()))

973 │ {

974 │ return Json(new ResponseModel(QLBHConst.Error, "Không có quyền truy cập kho hàng", null));

975 │ }

976 │

977 │ // FIX: Xử lý Ma giống như trong Update method

978 │ int \_Ma;

979 │ if (int.TryParse(Ma, out int directValue))

980 │ {

981 │ // Nếu là số nguyên, sử dụng trực tiếp

982 │ \_Ma = directValue;

983 │ }

984 │ else

985 │ {

986 │ // Nếu không phải số nguyên, thử giải mã

987 │ try

988 │ {

989 │ \_Ma = Cast.ToInt((Ma).Decrypt());

990 │ }

991 │ catch

992 │ {

993 │ return Json(new ResponseModel(QLBHConst.Error, "Mã hóa đơn không hợp lệ", null));

994 │ }

995 │ }

996 │

997 │ OpsHoaDon item = new HoaDonBanModel(dbcontext, vcContext).GetItem(\_Ma) ?? new OpsHoaDon();

998 │ if (item.Ma == 0)

999 │ {

1000 │ return Json(new ResponseModel(QLBHConst.Error, "Không tìm thấy hóa đơn", null));

1001 │ }

1002 │

1003 │ if (item.NgayHoaDon.Date != DateTime.Now.Date)

1004 │ {

1005 │ List<VPermission> ListPermission = GetVPermissions();

1006 │ if (!ListPermission.Any(t => t.PermissionId == Permission.TraHangQuaNgay.GetEnumGuid()))

1007 │ {

1008 │ return Json(new ResponseModel(QLBHConst.Error, "Không có quyền trả hàng qua ngày", null)...

1009 │ }

1010 │ }

1011 │

1012 │ List<OpsChiTietHoaDon> ListChiTiet = new ChiTietHoaDonModel(dbcontext).GetData(item.Ma) ?? new List<OpsC...

1013 │

1014 │ // FIX: Xử lý MaNganh giống như trong Update method

1015 │ int maNganhInt;

1016 │ if (int.TryParse(MaNganh, out int directMaNganhValue))

1017 │ {

1018 │ maNganhInt = directMaNganhValue;

1019 │ }

1020 │ else

1021 │ {

1022 │ try

1023 │ {

1024 │ maNganhInt = Cast.ToInt((MaNganh).Decrypt());

1025 │ }

1026 │ catch

1027 │ {

1028 │ return Json(new ResponseModel(QLBHConst.Error, "Mã ngành không hợp lệ", null));

1029 │ }

1030 │ }

1031 │

1032 │ var voucherRs = new ThanhToanVoucherModel(vcContext).GetVoucher(item.MaVachThe ?? "", MaChiNhanhVoucher,...

1033 │ var voucher = voucherRs.Voucher;

1034 │

1035 │ var responseData = new

1036 │ {

1037 │ itemHoaDonBan = item,

1038 │ ListChiTiet = ListChiTiet,

1039 │ Voucher = voucher,

1040 │ ListHangHoa = new GiaBanTheoKhoModel(dbcontext).DanhSachTatCaTonKhoHangHoaVaGiaBanTheoKho(Cast.ToInt...

1041 │ ListHinhThuc\_TT = new HinhThucTTModel(dbcontext).GetData(),

1042 │ ListGiamGia = new GiamGiaModel(dbcontext).GetData(objToken.branch),

1043 │ TuNgay = TuNgay == "" ? string.Format("{0:dd/MM/yyyy}", DateTime.Now) : TuNgay,

1044 │ DenNgay = DenNgay == "" ? string.Format("{0:dd/MM/yyyy}", DateTime.Now) : DenNgay,

1045 │ KhoHang = new DanhMucKhoModel(dbcontext).GetItem(MaKho),

1046 │ NguoiBan = NguoiBan + "",

1047 │ TraHang = "1",

1048 │ MaNganh = MaNganh

1049 │ };

1050 │ TonKhoBackgroundWorker.AddJob(new TonKhoJob { MaChiNhanh = objToken.branch, MaKho = MaKho, NamThang = st...

1051 │ return Json(new ResponseModel(QLBHConst.OK, "Lấy thông tin trả hàng thành công", responseData));

1052 │ }

1053 │ catch (Exception ex)

1054 │ {

1055 │ return Json(new ResponseModel(QLBHConst.Error, "Có lỗi xảy ra: " + ex.Message, null));

1056 │ }

1057 │ }

1058 │

1059 │ [HttpPost]

1060 │ [TokenUserCheckHTTP(Permission.NotCheck)]

1061 │ public ActionResult CheckVietQRPayment(string soHoaDon)

1062 │ {

1063 │ try

1064 │ {

1065 │ if (string.IsNullOrEmpty(soHoaDon))

1066 │ {

1067 │ return Json(new ResponseModel(QLBHConst.Error, "Số hóa đơn không được để trống", null...

1068 │ }

1069 │

1070 │ // Tìm VietQR record theo số hóa đơn

1071 │ var vietQrRecord = dbcontext.VietQr

1072 │ .Where(x => x.SoHoaDon == soHoaDon) // Bỏ filter Deleted

1073 │ .FirstOrDefault();

1074 │

1075 │ if (vietQrRecord == null)

1076 │ {

1077 │ return Json(new ResponseModel(QLBHConst.Error, "Không tìm thấy thông tin thanh toán", null));

1078 │ }

1079 │

1080 │ // Kiểm tra trạng thái đã thanh toán

1081 │ var isDaThanhToan = vietQrRecord.TrangThai == (int)TrangThaiVietQR.DaThanhToan;

1082 │

1083 │ var responseData = new

1084 │ {

1085 │ SoHoaDon = vietQrRecord.SoHoaDon,

1086 │ SoTien = vietQrRecord.SoTien,

1087 │ DaThanhToan = isDaThanhToan,

1088 │ NgayTao = vietQrRecord.CreatedDate?.ToString("dd/MM/yyyy HH:mm"),

1089 │ NgayThu = vietQrRecord.NgayThu?.ToString("dd/MM/yyyy HH:mm")

1090 │ };

1091 │

1092 │ return Json(new ResponseModel(QLBHConst.OK,

1093 │ isDaThanhToan ? "Đã thanh toán thành công" : "Chưa thanh toán",

1094 │ responseData));

1095 │ }

1096 │ catch (Exception ex)

1097 │ {

1098 │ return Json(new ResponseModel(QLBHConst.Error, "Có lỗi xảy ra: " + ex.Message, null));

1099 │ }

1100 │ }

1101 │ [HttpPost]

1102 │ [APIKeyCheck]

1103 │ public ActionResult DaChuyenKhoan(string soHoaDon)

1104 │ {

1105 │ \_chatHub.Clients.All.SendAsync("payOK", new { MaThanhToan = soHoaDon });

1106 │ return Json(new ResponseModel(QLBHConst.OK, "Hoàn tất thanh toán"));

1107 │ }

1108 │ }

1109 │ }

────────────────────────────────────────────────────────────

**📄 BangKeThuMuaKhongHoaDonController.cs (CS, 340 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using QuanLyBanHang.Common;

3 │ using QuanLyBanHang.Common.Attributes;

4 │ using QuanLyBanHang.Models.APIModels;

5 │ using QuanLyBanHang.Models.ExModel.CAT;

6 │ using QuanLyBanHang.Models.QLBHEntity;

7 │ using System.Data;

8 │

9 │ namespace QuanLyBanHang.Controllers.OPS

10 │ {

11 │ public class BangKeThuMuaKhongHoaDonController : BaseController

12 │ {

13 │ private readonly IWebHostEnvironment \_hostingEnvironment;

14 │

15 │ public BangKeThuMuaKhongHoaDonController(QuanLyBanHangContext dbcontext, MemoryCacheManager memoryCache, IWebHos...

16 │ {

17 │ m\_MenuID = MenuID.opsBangKeThuMuaKhongHoaDon.GetEnumGuid();

18 │ \_hostingEnvironment = hostingEnvironment;

19 │ }

20 │

21 │ public class BangKeThuMuaKhongHoaDon

22 │ {

23 │ public string NgayHoaDon { get; set; }

24 │ public string MaNhaCungCap { get; set; }

25 │ public string TenNhaCungCap { get; set; }

26 │ public string DiaChi { get; set; }

27 │ public string CCCD { get; set; }

28 │ public string MaHangHoa { get; set; }

29 │ public string TenHangHoa { get; set; }

30 │ public decimal? SoLuong { get; set; }

31 │ public decimal? DonGia { get; set; }

32 │ public decimal? ThanhTien { get; set; }

33 │ public string GhiChu { get; set; }

34 │ public string NguoiLienHe { get; set; }

35 │ }

36 │

37 │ [TokenUserCheckHTTP]

38 │ public IActionResult Index()

39 │ {

40 │ ViewData["ListNhaCC"]= new NhaCCModel(dbcontext).GetData(objToken.branch);

41 │ ViewData["ListHangHoa"] = new TblHangHoaModel(dbcontext).GetData(objToken.branch);

42 │

43 │ return View();

44 │ }

45 │

46 │ [HttpPost]

47 │ [TokenUserCheckHTTP]

48 │ public ActionResult getListData(string TuNgay, string DenNgay, string MaNhaCC, string MaHangHoa)

49 │ {

50 │ DateTime TuNgayConvert = TuNgay.ToDateTime2(DateTime.Now).Value.Date;

51 │ DateTime DenNgayConvert = DenNgay.ToDateTime2(DateTime.Now).Value.Date;

52 │

53 │ return Json(new ResponseModel(QLBHConst.OK, "", getBangKeThuMua(TuNgayConvert, DenNgayConvert, Cast.ToInt(Ma...

54 │ }

55 │

56 │ private List<BangKeThuMuaKhongHoaDon> getBangKeThuMua(DateTime dtpTuNgay, DateTime dtpDenNgay, int MaNhaCungCap,...

57 │ {

58 │ var listData = (from ct in dbcontext.StockChiTietNhapHang

59 │ join nh in dbcontext.StockNhapHang on ct.MaNhapHang equals nh.Ma

60 │ join ncc in dbcontext.TblNhaCc on nh.MaNhaCc equals ncc.Ma

61 │ join hh in dbcontext.TblHangHoa on ct.MaHangHoa equals hh.Ma

62 │ where !(hh.Deleted ?? false) && !(ct.Deleted ?? false) && (nh.KhongHoaDon ?? false)

63 │ && (MaNhaCungCap == 0 || MaNhaCungCap == ncc.Ma)

64 │ && (MaHangHoa == 0 || MaHangHoa == hh.Ma)

65 │ && nh.NgayHoaDon != null && nh.NgayHoaDon.Value.Date >= dtpTuNgay.Date

66 │ && nh.NgayHoaDon.Value.Date <= dtpDenNgay.Date

67 │ select new BangKeThuMuaKhongHoaDon

68 │ {

69 │ NgayHoaDon = string.Format("{0:dd/MM/yyyy}", nh.NgayHoaDon.Value),

70 │ MaNhaCungCap = ncc.Ma + "",

71 │ TenNhaCungCap = ncc.TenNhaCc + "",

72 │ DiaChi = ncc.DiaChi == null ? "" : ncc.DiaChi,

73 │ CCCD = ncc.Cccd == null ? "" : ncc.Cccd,

74 │ MaHangHoa = ct.MaHangHoa + "",

75 │ TenHangHoa = hh.Ten + "",

76 │ SoLuong = ct.SoLuong,

77 │ DonGia = ct.GiaVon,

78 │ ThanhTien = (ct.ThanhTien ?? 0),

79 │ GhiChu = nh.SoHoaDon + "",

80 │ NguoiLienHe = ncc.NguoiLienHe + ""

81 │ }).ToList();

82 │ return listData;

83 │ }

84 │

85 │ [HttpPost]

86 │ [TokenUserCheckHTTP]

87 │ public ActionResult getDanhMuc()

88 │ {

89 │ var listnhacungcap = dbcontext.TblNhaCc.Where(t => !(t.Deleted ?? false)).Select(t => new

90 │ {

91 │ ma = t.Ma + "",

92 │ ten = t.TenNhaCc + ""

93 │ }).ToList();

94 │ var listhanghoa = dbcontext.TblHangHoa.Where(t => !(t.Deleted ?? false)).Select(t => new

95 │ {

96 │ ma = t.Ma + "",

97 │ ten = t.Ten + ""

98 │ }).ToList();

99 │

100 │ return Json(new ResponseModel(QLBHConst.OK, "", new { listnhacungcap, listhanghoa }));

101 │ }

102 │

103 │ [TokenUserCheckHTTP]

104 │ public ActionResult ExportExcel(DateTime TuNgay, DateTime DenNgay, int MaNhaCungCap, int MaHangHoa, string FileT...

105 │ {

106 │ var ListDataDetail = getBangKeThuMua(TuNgay, DenNgay, MaNhaCungCap, MaHangHoa);

107 │

108 │ if (ListDataDetail != null && ListDataDetail.Count > 0)

109 │ {

110 │ string FileName = string.Format("BangKeThuMuaKhongHoaDon\_{0:ddMMyyyy}\_{1:ddMMyyyy}", TuNgay, DenNgay);

111 │ string templatePath = Path.Combine(\_hostingEnvironment.ContentRootPath, "wwwroot", "templates", "BangKeT...

112 │ if (TuNgay > DenNgay)

113 │ {

114 │ return Json(new ResponseModel(QLBHConst.Error, "Từ ngày không được lớn hơn đến ngày")...

115 │ }

116 │ if (!System.IO.File.Exists(templatePath))

117 │ {

118 │ return Json(new ResponseModel(QLBHConst.Error, "Không tìm thấy template file"));

119 │ }

120 │ if (ListDataDetail == null || ListDataDetail.Count == 0)

121 │ {

122 │ return Json(new ResponseModel(QLBHConst.Error, "Không có dữ liệu để xuất file"));

123 │ }

124 │

125 │ // Kiểm tra template tồn tại

126 │ if (System.IO.File.Exists(templatePath))

127 │ {

128 │ int DongBatDau = 17; // Dòng bắt đầu dữ liệu trong template

129 │

130 │ // Chuẩn bị DataTable cho dữ liệu chính

131 │ DataTable dtDetail = new DataTable();

132 │ dtDetail.Columns.Add("Ngày tháng mua hàng");

133 │ dtDetail.Columns.Add("Người bán");

134 │ dtDetail.Columns.Add("Địa chỉ");

135 │ dtDetail.Columns.Add("Số CCCD/CMND");

136 │ dtDetail.Columns.Add("Tên hàng hóa, dịch vụ");

137 │ dtDetail.Columns.Add("Số lượng");

138 │ dtDetail.Columns.Add("Đơn giá");

139 │ dtDetail.Columns.Add("Thành tiền");

140 │ dtDetail.Columns.Add("Tổng");

141 │ dtDetail.Columns.Add("Ghi chú");

142 │

143 │ decimal tong = 0;

144 │ foreach (var item in ListDataDetail)

145 │ {

146 │ dtDetail.Rows.Add(new object[]

147 │ {

148 │ item.NgayHoaDon,

149 │ item.NguoiLienHe,

150 │ item.DiaChi,

151 │ item.CCCD,

152 │ item.TenHangHoa,

153 │ item.SoLuong,

154 │ item.DonGia,

155 │ item.ThanhTien,

156 │ item.GhiChu,

157 │ });

158 │ tong += Cast.ToDecimal(item.ThanhTien);

159 │ }

160 │

161 │ // Tính toán vị trí dòng cho footer

162 │ int IndexRow = DongBatDau + dtDetail.Rows.Count;

163 │ int bottomidx = IndexRow + 2;

164 │

165 │ // Chuẩn bị giá trị riêng lẻ

166 │ List<ExcelValue> ListGiaTriRiengLe = new List<ExcelValue>

167 │ {

168 │ new ExcelValue {

169 │ Range = "D7",

170 │ Value = string.Format("Ngày {0:dd} tháng {0:MM} năm {0:yyyy}", DateTime.Today)

171 │ },

172 │ new ExcelValue {

173 │ Range = "A9",

174 │ Value = "Tên doanh nghiệp: " + QLBHConst.TenDoanhNghiep

175 │ },

176 │ new ExcelValue {

177 │ Range = "A10",

178 │ Value = "Địa chỉ: " + QLBHConst.DiaChiDoanhNghiep

179 │ },

180 │ new ExcelValue {

181 │ Range = "A11",

182 │ Value = "Địa chỉ nơi tổ chức thu mua: " + ListDataDetail[0].DiaChi

183 │ },

184 │ new ExcelValue {

185 │ Range = "A12",

186 │ Value = "Người phụ trách thu mua: " + QLBHConst.NguoiPhuTrach

187 │ },

188 │ new ExcelValue {

189 │ Range = "H9",

190 │ Value = "Mã số thuế: " + QLBHConst.MaSoThue

191 │ }

192 │ };

193 │

194 │ // Chuẩn bị giá trị riêng lẻ 2 (cho footer và định dạng)

195 │ List<ExcelValue> ListGiaTriRiengLe2 = new List<ExcelValue>

196 │ {

197 │

198 │ // Tổng giá trị

199 │ new ExcelValue {

200 │ Range = $"A{bottomidx}",

201 │ Value = $"Tổng giá trị hàng hóa mua vào: {tong:#,##0.00}",

202 │ Format = new ExcelFormat {

203 │ HorizontalAlignment = eHorizontalAlignment.Left

204 │ }

205 │ },

206 │

207 │ // Người lập bảng kê

208 │ new ExcelValue {

209 │ Range = $"B{bottomidx + 2}",

210 │ Value = "Người lập bảng kê",

211 │ Format = new ExcelFormat {

212 │ IsBold = true,

213 │ HorizontalAlignment = eHorizontalAlignment.Center

214 │ }

215 │ },

216 │

217 │ // Ngày tháng năm

218 │ new ExcelValue {

219 │ Range = $"G{bottomidx + 2}",

220 │ Value = $"Ngày {DateTime.Today:dd} tháng {DateTime.Today:MM} năm {DateTime.Today:yyyy}",

221 │ Format = new ExcelFormat {

222 │ HorizontalAlignment = eHorizontalAlignment.Center,

223 │ IsItalic = true

224 │ }

225 │ },

226 │

227 │ // Ký ghi rõ họ tên

228 │ new ExcelValue {

229 │ Range = $"B{bottomidx + 3}",

230 │ Value = "(Ký, ghi rõ họ tên)",

231 │ Format = new ExcelFormat {

232 │ HorizontalAlignment = eHorizontalAlignment.Center

233 │ }

234 │ },

235 │

236 │ // Giám đốc doanh nghiệp

237 │ new ExcelValue {

238 │ Range = $"G{bottomidx + 3}",

239 │ Value = "Giám đốc doanh nghiệp",

240 │ Format = new ExcelFormat {

241 │ IsBold = true,

242 │ HorizontalAlignment = eHorizontalAlignment.Center

243 │ }

244 │ },

245 │

246 │ // Ký tên đóng dấu

247 │ new ExcelValue {

248 │ Range = $"G{bottomidx + 4}",

249 │ Value = "(Ký tên, đóng dấu)",

250 │ Format = new ExcelFormat {

251 │ HorizontalAlignment = eHorizontalAlignment.Center,

252 │ WrapText = false

253 │ }

254 │ },

255 │

256 │ // Tên giám đốc

257 │ new ExcelValue {

258 │ Range = $"G{bottomidx + 8}",

259 │ Value = QLBHConst.GiamDocDoanhNghiep,

260 │ Format = new ExcelFormat {

261 │ IsBold = true,

262 │ HorizontalAlignment = eHorizontalAlignment.Center,

263 │ WrapText = false

264 │ }

265 │ },

266 │

267 │ // Ghi chú

268 │ new ExcelValue {

269 │ Range = $"A{bottomidx + 9}",

270 │ Value = "Ghi chú",

271 │ Format = new ExcelFormat {

272 │ IsBold = true

273 │ }

274 │ },

275 │

276 │ // Ghi chú 1

277 │ new ExcelValue {

278 │ Range = $"A{bottomidx + 10}:I{bottomidx + 10}",

279 │ Value = QLBHConst.Note1,

280 │ KhongGopO = false, // Cho phép gộp ô

281 │ Format = new ExcelFormat {

282 │ WrapText = true,

283 │ HorizontalAlignment = eHorizontalAlignment.Left,

284 │ Height = 57

285 │ }

286 │ },

287 │

288 │ // Ghi chú 2

289 │ new ExcelValue {

290 │ Range = $"A{bottomidx + 11}:I{bottomidx + 11}",

291 │ Value = QLBHConst.Note2,

292 │ KhongGopO = false, // Cho phép gộp ô

293 │ Format = new ExcelFormat {

294 │ WrapText = true,

295 │ HorizontalAlignment = eHorizontalAlignment.Left,

296 │ Height = 35

297 │ }

298 │ }

299 │ };

300 │

301 │ // Sử dụng ExportToExcel.FromTemplate để xuất file

302 │ Stream fileStream = ExportToExcel.FromTemplateBangKe(

303 │ templatePath,

304 │ FileType,

305 │ ListGiaTriRiengLe,

306 │ dtDetail,

307 │ true,

308 │ DongBatDau,

309 │ "A",

310 │ "I",

311 │ ListGiaTriRiengLe2

312 │ );

313 │

314 │ string contentType = (FileType.ToLower() == "pdf") ? "application/pdf" : "application/ms-excel";

315 │ string FileTypeension = (FileType.ToLower() == "pdf") ? ".pdf" : ".xlsx";

316 │ string fileNameWithExt = FileName + FileTypeension;

317 │

318 │ return new FileStreamResult(fileStream, contentType)

319 │ {

320 │ FileDownloadName = fileNameWithExt

321 │ };

322 │ }

323 │ }

324 │

325 │ return Json(new ResponseModel(QLBHConst.Error, "", "Không có dữ liệu hoặc template không tồn tạ...

326 │ }

327 │

328 │ [HttpPost]

329 │ public JsonResult CheckExport(DateTime TuNgay, DateTime DenNgay, int MaNhaCungCap, int MaHangHoa)

330 │ {

331 │ var list = getBangKeThuMua(TuNgay, DenNgay, MaNhaCungCap, MaHangHoa);

332 │

333 │ if (list == null || !list.Any())

334 │ return Json(new ResponseModel(QLBHConst.Error, "Không có dữ liệu để xuất"));

335 │

336 │ return Json(new ResponseModel(QLBHConst.OK, "OK"));

337 │ }

338 │

339 │ }

340 │ }

────────────────────────────────────────────────────────────

**📄 BaoCaoDoanhThuTheoNgayController.cs (CS, 412 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using Microsoft.EntityFrameworkCore;

3 │ using QuanLyBanHang.Common;

4 │ using QuanLyBanHang.Common.Attributes;

5 │ using QuanLyBanHang.Models.APIModels;

6 │ using QuanLyBanHang.Models.ExModel.CAT;

7 │ using QuanLyBanHang.Models.ExModel.StoredProcedureModel;

8 │ using QuanLyBanHang.Models.ExModel.SYS;

9 │ using QuanLyBanHang.Models.QLBHEntity;

10 │ using System.Data;

11 │

12 │ namespace QuanLyBanHang.Controllers.OPS

13 │ {

14 │ public class BaoCaoDoanhThuTheoNgayController : BaseController

15 │ {

16 │ private readonly IWebHostEnvironment \_hostingEnvironment;

17 │

18 │ public BaoCaoDoanhThuTheoNgayController(QuanLyBanHangContext dbcontext, MemoryCacheManager memoryCache, IWebHost...

19 │ {

20 │ m\_MenuID = MenuID.opsBaoCao.GetEnumGuid();

21 │ \_hostingEnvironment = hostingEnvironment;

22 │ }

23 │

24 │ [TokenUserCheckHTTP]

25 │ public IActionResult Index()

26 │ {

27 │ ViewData["ListUser"] = new NguoiDungModel(dbcontext).GetData();

28 │ ViewData["ListKho"] = new DanhMucKhoModel(dbcontext).GetData(objToken.branch).Where(t => t.BanHang ?? false)...

29 │ ViewData["MaNhanVien"] = objToken.userid;

30 │

31 │ return View();

32 │ }

33 │

34 │ [TokenUserCheckHTTP]

35 │ public ActionResult getListData(DateTime TuNgay, DateTime DenNgay, Guid NguoiBan, string LoaiHang, string FileTy...

36 │ {

37 │ DenNgay = DenNgay.Date.AddDays(1).AddTicks(-1);

38 │

39 │ var query1 = from hd in dbcontext.OpsHoaDon.AsNoTracking().Where(x => x.MaChiNhanh == objToken.branch)

40 │ join ct in dbcontext.OpsChiTietHoaDon.AsNoTracking() on hd.Ma equals ct.MaHoaDon

41 │ where !(hd.Deleted ?? false) && !(ct.Deleted ?? false)

42 │ && hd.NgayHoaDon.Date >= TuNgay.Date && hd.NgayHoaDon.Date <= DenNgay.Date

43 │ && (NguoiBan == Guid.Empty || hd.NguoiThanhToan == NguoiBan) && (MaKho == 0 || hd.MaKho == MaKh...

44 │ && (HinhThucTT == 0 || hd.HinhThucTt == HinhThucTT)

45 │ && hd.HinhThucTt != (int)HinhThucThanhToan.ChuyenKhoan

46 │ select new sp\_BaoCaoDoanhThuTheoNguoiBanTheoCa

47 │ {

48 │ NgayHoaDon = hd.NgayHoaDon.Date,

49 │ NguoiBan = hd.NguoiThanhToan,

50 │ SoSeri = hd.SoSeri,

51 │ MaVach = ct.MaVach,

52 │ MaHangHoa = ct.MaHangHoa,

53 │ SoLuong = ct.SoLuong,

54 │ DonGia = ct.DonGia,

55 │ GiaGiam = ct.GiaGiam,

56 │ MaKho = hd.MaKho,

57 │ SoLuongTra = 0.0m,

58 │ SoLuongCK = 0,

59 │ HinhThucTt = hd.HinhThucTt,

60 │ KhuyenMaiTienMat = hd.KhuyenMaiTienMat ?? 0,

61 │ GhiChu = hd.GhiChu ?? string.Empty,

62 │ RequestID = string.Empty,

63 │ NgayTao = hd.CreatedDate ?? DateTime.Now

64 │ };

65 │ var query2 = from hd in dbcontext.OpsHoaDon.AsNoTracking().Where(x => x.MaChiNhanh == objToken.branch)

66 │ join ct in dbcontext.OpsChiTietHoaDon.AsNoTracking() on hd.Ma equals ct.MaHoaDon

67 │ where !(hd.Deleted ?? false) && !(ct.Deleted ?? false)

68 │ && hd.NgayHoaDon.Date >= TuNgay.Date && hd.NgayHoaDon.Date <= DenNgay.Date

69 │ && (NguoiBan == Guid.Empty || hd.NguoiThanhToan == NguoiBan) && (MaKho == 0 || hd.MaKho == MaKh...

70 │ && (HinhThucTT == 0 || hd.HinhThucTt == HinhThucTT)

71 │ && hd.HinhThucTt == (int)HinhThucThanhToan.ChuyenKhoan

72 │ select new sp\_BaoCaoDoanhThuTheoNguoiBanTheoCa

73 │ {

74 │ NgayHoaDon = hd.NgayHoaDon.Date,

75 │ NguoiBan = hd.NguoiThanhToan,

76 │ SoSeri = hd.SoSeri,

77 │ MaVach = ct.MaVach,

78 │ MaHangHoa = ct.MaHangHoa,

79 │ SoLuong = 0,

80 │ DonGia = ct.DonGia,

81 │ GiaGiam = ct.GiaGiam,

82 │ MaKho = hd.MaKho,

83 │ SoLuongTra = 0.0m,

84 │ SoLuongCK = ct.SoLuong,

85 │ HinhThucTt = hd.HinhThucTt,

86 │ KhuyenMaiTienMat = hd.KhuyenMaiTienMat ?? 0,

87 │ GhiChu = hd.GhiChu ?? string.Empty,

88 │ RequestID = hd.RequestId ?? string.Empty,

89 │ NgayTao = hd.CreatedDate

90 │ };

91 │ var query3 = from hd in dbcontext.OpsHoaDon.AsNoTracking().Where(x => x.MaChiNhanh == objToken.branch)

92 │ join ct in dbcontext.OpsChiTietHoaDon.AsNoTracking() on hd.Ma equals ct.MaHoaDon

93 │ where !(hd.Deleted ?? false) && !(ct.Deleted ?? false)

94 │ && hd.NgayHoaDon.Date >= TuNgay.Date && hd.NgayHoaDon.Date <= DenNgay.Date

95 │ && (NguoiBan == Guid.Empty || ct.NguoiTra == NguoiBan) && (MaKho == 0 || hd.MaKho == MaKho)

96 │ && (ct.SoLuongTraLai ?? 0) > 0 && ct.NgayTra != null

97 │ select new sp\_BaoCaoDoanhThuTheoNguoiBanTheoCa

98 │ {

99 │ NgayHoaDon = ct.NgayTra.Value.Date,

100 │ NguoiBan = ct.NguoiTra ?? Guid.Empty,

101 │ SoSeri = hd.SoSeri,

102 │ MaVach = ct.MaVach,

103 │ MaHangHoa = ct.MaHangHoa,

104 │ SoLuong = 0,

105 │ DonGia = ct.DonGia,

106 │ GiaGiam = ct.GiaGiam,

107 │ MaKho = hd.MaKho,

108 │ SoLuongTra = ct.SoLuongTraLai ?? 0.0m,

109 │ SoLuongCK = 0,

110 │ HinhThucTt = hd.HinhThucTt,

111 │ KhuyenMaiTienMat = 0,

112 │ GhiChu = string.Empty,

113 │ RequestID = string.Empty,

114 │ NgayTao = ct.NgayTra

115 │ };

116 │

117 │ var query1Result = query1.ToList();

118 │ var query2Result = query2.ToList();

119 │ var query3Result = query3.ToList();

120 │

121 │ var lstHoaDon = query1Result.Concat(query2Result).Concat(query3Result).ToList();

122 │

123 │ // 2. Tối ưu GroupBy bằng cách chỉ chọn các trường cần thiết

124 │ var data = lstHoaDon

125 │ .GroupBy(t => new

126 │ {

127 │ t.NgayHoaDon,

128 │ t.NguoiBan,

129 │ t.SoSeri,

130 │ t.MaHangHoa,

131 │ t.MaVach,

132 │ t.MaKho,

133 │ t.HinhThucTt,

134 │ t.KhuyenMaiTienMat,

135 │ t.GhiChu,

136 │ t.RequestID,

137 │ t.NgayTao

138 │ })

139 │ .Select(t => new

140 │ {

141 │ t.Key.NgayHoaDon,

142 │ t.Key.NguoiBan,

143 │ t.Key.SoSeri,

144 │ t.Key.MaHangHoa,

145 │ t.Key.MaVach,

146 │ t.Key.MaKho,

147 │ t.Key.HinhThucTt,

148 │ SoLuong = (double)t.Sum(x => x.SoLuong ?? 0),

149 │ DonGia = (double)t.Max(x => x.DonGia ?? 0),

150 │ SoLuongTra = (double)t.Sum(x => x.SoLuongTra),

151 │ SoLuongCK = (double)t.Sum(x => x.SoLuongCK ?? 0),

152 │ GiaGiam = (double)t.Max(x => x.GiaGiam ?? 0),

153 │ t.Key.KhuyenMaiTienMat,

154 │ t.Key.GhiChu,

155 │ t.Key.RequestID,

156 │ t.Key.NgayTao

157 │ })

158 │ .ToList();

159 │

160 │ // 3. Tính toán tổng số lượng cho mỗi hóa đơn trước

161 │ var invoiceTotals = data

162 │ .GroupBy(x => x.SoSeri)

163 │ .ToDictionary(

164 │ g => g.Key,

165 │ g => g.Sum(x => x.SoLuong + x.SoLuongCK)

166 │ );

167 │

168 │ // 4. Lấy tất cả ID cần thiết

169 │ var hangHoaIds = data.Select(x => x.MaHangHoa).Distinct().ToList();

170 │ var nguoiBanIds = data.Select(x => x.NguoiBan).Distinct().ToList();

171 │ var maKhoIds = data.Select(x => x.MaKho).Distinct().ToList();

172 │ var hinhThucIds = data.Select(x => x.HinhThucTt).Distinct().ToList();

173 │

174 │ // 5. Truy vấn dữ liệu liên quan một lần và lưu vào Dictionary

175 │ var listHH = dbcontext.VHangHoa.AsNoTracking().Where(x => x.MaChiNhanh == objToken.branch)

176 │ .Where(t => hangHoaIds.Contains(t.Ma))

177 │ .ToDictionary(t => t.Ma);

178 │

179 │ var listUser = dbcontext.SysUser.AsNoTracking()

180 │ .Where(t => nguoiBanIds.Contains(t.Ma))

181 │ .ToDictionary(t => t.Ma);

182 │

183 │ var listKho = dbcontext.StockDanhMucKho.AsNoTracking().Where(x => x.MaChiNhanh == objToken.branch)

184 │ .Where(t => maKhoIds.Contains(t.Ma))

185 │ .ToDictionary(t => t.Ma);

186 │

187 │ var listHTTT = dbcontext.TblHinhThucThanhToan.AsNoTracking()

188 │ .Where(t => hinhThucIds.Contains(t.Ma))

189 │ .ToDictionary(t => t.Ma);

190 │

191 │ // 6. Tạo kết quả cuối cùng với Dictionary lookup thay vì FirstOrDefault

192 │ var listData = data.Select(t =>

193 │ {

194 │ listUser.TryGetValue(t.NguoiBan ?? Guid.Empty, out var u);

195 │ listHH.TryGetValue(t.MaHangHoa ?? 0, out var hh);

196 │ listKho.TryGetValue(t.MaKho ?? 0, out var kho);

197 │ listHTTT.TryGetValue(t.HinhThucTt, out var hinhThuc);

198 │

199 │ invoiceTotals.TryGetValue(t.SoSeri, out var totalQuantity);

200 │ totalQuantity = totalQuantity == 0 ? 1 : totalQuantity;

201 │

202 │ var khuyenMaiTienMat = t.KhuyenMaiTienMat == 0 ? 0 :

203 │ (int)((double)t.KhuyenMaiTienMat / totalQuantity \* (t.SoLuong + t.SoLuongCK));

204 │

205 │ return new sp\_BaoCaoDoanhThuTheoNguoiBanTheoCa

206 │ {

207 │ TuNgay = TuNgay.Date,

208 │ DenNgay = DenNgay.Date,

209 │ NgayHoaDon = t.NgayHoaDon,

210 │ NguoiBan = t.NguoiBan,

211 │ Username = u?.Username ?? "",

212 │ Fullname = u?.FullName ?? "",

213 │ TenHang = hh?.TenHangHoa ?? "",

214 │ LopHang = hh?.MaLopHang ?? 0,

215 │ TenLop = hh?.TenLopHang ?? "",

216 │ SoSeri = t.SoSeri,

217 │ MaVach = t.MaVach,

218 │ MaKT = hh?.MaKt ?? "",

219 │ Dvt = hh?.MaDvt ?? 0,

220 │ TenDVT = hh?.TenDvt ?? "",

221 │ MaKho = t.MaKho ?? 0,

222 │ TenKho = kho?.Ten ?? "",

223 │ SoLuong = (decimal)(t.SoLuong + t.SoLuongCK),

224 │ DonGia = (decimal)t.DonGia,

225 │ SoLuongTra = (decimal)t.SoLuongTra,

226 │ ThanhTien = (decimal)((t.SoLuong + t.SoLuongCK) \* t.DonGia),

227 │ GiamGia = (decimal)(((t.SoLuong + t.SoLuongCK) \* t.DonGia) \* t.GiaGiam / 100.0),

228 │ ThanhTienTra = (decimal)(t.SoLuongTra \* t.DonGia \* (100.0 - t.GiaGiam) / 100.0),

229 │ TienChuyenKhoan = (decimal)(t.SoLuongCK \* t.DonGia),

230 │ TongCong = (decimal)(

231 │ ((t.SoLuong + t.SoLuongCK) \* t.DonGia)

232 │ - (((t.SoLuong + t.SoLuongCK) \* t.DonGia) \* t.GiaGiam / 100.0)

233 │ - (t.SoLuongTra \* t.DonGia \* (100.0 - t.GiaGiam) / 100.0)

234 │ - (t.SoLuongCK \* t.DonGia)

235 │ - khuyenMaiTienMat

236 │ ),

237 │ MaHinhThuc = t.HinhThucTt,

238 │ TenHinhThuc = hinhThuc?.Ten ?? "",

239 │ KhuyenMaiTienMat = khuyenMaiTienMat,

240 │ GhiChu = t.GhiChu,

241 │ RequestID = t.RequestID,

242 │ NgayTao = t.NgayTao

243 │ };

244 │ }).ToList();

245 │

246 │ var SumThanhTien = listData.Sum(t => t.ThanhTien);

247 │ var SumThanhTienTra = listData.Sum(t => t.ThanhTienTra);

248 │ var SumGiamGia = listData.Sum(t => t.GiamGia);

249 │ var SumTongCong = listData.Sum(t => t.TongCong);

250 │ var SumSoLuong = listData.Sum(t => t.SoLuong ?? 0);

251 │ var SumSoLuongTra = listData.Sum(t => t.SoLuongTra ?? 0);

252 │ var SumKhuyenMaiTM = listData.Sum(t => t.KhuyenMaiTienMat);

253 │ var SumTongCongChuyenKhoan = listData.Sum(t => t.TienChuyenKhoan);

254 │

255 │ if (FileType == null)

256 │ {

257 │ return Json(new ResponseModel(QLBHConst.OK, "", new

258 │ {

259 │ listData = listData.Select(t => new

260 │ {

261 │ TuNgay = t.TuNgay.ToString("dd/MM/yyyy"),

262 │ DenNgay = t.DenNgay.ToString("dd/MM/yyyy"),

263 │ Ngay\_HD = t.NgayHoaDon.ToString("dd/MM/yyyy"),

264 │ NguoiBan = t.NguoiBan.ToString(),

265 │ Tendangnhap = t.Username,

266 │ Hoten = t.Fullname,

267 │ LopHang = t.LopHang.ToString(),

268 │ TenLop = t.TenLop,

269 │ So\_Seri = t.SoSeri,

270 │ MaVach = t.MaVach,

271 │ MaKT = t.MaKT,

272 │ TenHang = t.TenHang,

273 │ dvt = t.Dvt,

274 │ TenDVT = t.TenDVT,

275 │ MaKho = t.MaKho.ToString(),

276 │ TenKho = t.TenKho,

277 │ SoLuong = t.SoLuong,

278 │ DonGia = t.DonGia,

279 │ SoLuongTra = t.SoLuongTra,

280 │ ThanhTien = t.ThanhTien,

281 │ GiamGia = t.GiamGia,

282 │ ThanhTienTra = t.ThanhTienTra,

283 │ TongCong = t.TongCong,

284 │ TenHinhThuc = t.TenHinhThuc,

285 │ KhuyenMaiTienMat = t.KhuyenMaiTienMat,

286 │ TienChuyenKhoan = t.TienChuyenKhoan,

287 │ GhiChu = t.GhiChu,

288 │ RequestID = t.RequestID

289 │ }).ToList(),

290 │ SumThanhTien,

291 │ SumThanhTienTra,

292 │ SumGiamGia,

293 │ SumTongCong,

294 │ SumSoLuong,

295 │ SumSoLuongTra,

296 │ SumKhuyenMaiTM,

297 │ SumTongCongChuyenKhoan

298 │ }));

299 │ }

300 │

301 │ StockDanhMucKho drKho = new DanhMucKhoModel(dbcontext).GetItem(MaKho);

302 │

303 │ DataTable dt = new DataTable();

304 │ dt.Columns.Add("Ngày HĐ");

305 │ dt.Columns.Add("Ngày tạo");

306 │ dt.Columns.Add("Thu ngân");

307 │ dt.Columns.Add("Tên lớp");

308 │ dt.Columns.Add("Mã KT");

309 │ dt.Columns.Add("Tên hàng");

310 │ dt.Columns.Add("Số seri");

311 │ dt.Columns.Add("Mã vạch");

312 │ dt.Columns.Add("SL bán", typeof(decimal));

313 │ dt.Columns.Add("SL trả", typeof(decimal));

314 │ dt.Columns.Add("ĐVT");

315 │ dt.Columns.Add("Đơn giá", typeof(decimal));

316 │ dt.Columns.Add("Thành tiền", typeof(decimal));

317 │ dt.Columns.Add("Giảm giá", typeof(decimal));

318 │ dt.Columns.Add("KMTM", typeof(decimal));

319 │ dt.Columns.Add("Thành tiền trả", typeof(decimal));

320 │ dt.Columns.Add("Chuyển khoản", typeof(decimal));

321 │ dt.Columns.Add("Tổng cộng", typeof(decimal));

322 │ dt.Columns.Add("Hình thức TT");

323 │ dt.Columns.Add("Ghi chú");

324 │ dt.Columns.Add("Số hóa đơn ck");

325 │

326 │ decimal sumThanhTien = 0, sumGiamGia = 0, sumTongCong = 0;

327 │ decimal sumKMTM = 0, sumCK = 0, sumTra = 0, sumSL = 0, sumSLTra = 0;

328 │

329 │ foreach (var item in listData)

330 │ {

331 │ dt.Rows.Add(new object[]

332 │ {

333 │ string.Format("'{0:dd/MM/yyyy}", item.NgayHoaDon),

334 │ string.Format("'{0:dd/MM/yyyy HH:mm}", item.NgayTao),

335 │ item.Fullname ?? "",

336 │ item.TenLop ?? "",

337 │ item.MaKT ?? "",

338 │ item.TenHang ?? "",

339 │ item.SoSeri ?? "",

340 │ item.MaVach ?? "",

341 │ item.SoLuong ?? 0.0M,

342 │ item.SoLuongTra ?? 0.0M,

343 │ item.TenDVT ?? "",

344 │ item.DonGia ?? 0.0M,

345 │ item.ThanhTien ,

346 │ item.GiamGia,

347 │ item.KhuyenMaiTienMat,

348 │ item.ThanhTienTra ,

349 │ item.TienChuyenKhoan,

350 │ item.TongCong,

351 │ item.TenHinhThuc ?? "",

352 │ item.GhiChu ?? "",

353 │ "'" + (item.RequestID ?? "")

354 │ });

355 │

356 │ sumThanhTien += item.ThanhTien;

357 │ sumGiamGia += item.GiamGia;

358 │ sumTongCong += item.TongCong;

359 │ sumKMTM += item.KhuyenMaiTienMat;

360 │ sumCK += item.TienChuyenKhoan;

361 │ sumTra += item.ThanhTienTra;

362 │ sumSL += item.SoLuong ?? 0;

363 │ sumSLTra += item.SoLuongTra ?? 0;

364 │ }

365 │

366 │ // Thêm dòng tổng cộng

367 │ dt.Rows.Add(new object[]

368 │ {

369 │ "TỔNG CỘNG", "", "", "", "", "", "", "",

370 │ sumSL,

371 │ sumSLTra,

372 │ "",

373 │ DBNull.Value,

374 │ sumThanhTien,

375 │ sumGiamGia,

376 │ sumKMTM,

377 │ sumTra,

378 │ sumCK,

379 │ sumTongCong,

380 │ "", "", ""

381 │ });

382 │

383 │ // Chuẩn bị dữ liệu riêng lẻ (header, tiêu đề...)

384 │ List<ExcelValue> headerData = new List<ExcelValue>

385 │ {

386 │ new ExcelValue

387 │ {

388 │ Range = "A2",

389 │ Value = $"Từ {TuNgay:dd/MM/yyyy} đến {DenNgay:dd/MM/yyyy}"

390 │ },

391 │ new ExcelValue

392 │ {

393 │ Range = "A3",

394 │ Value = $"Kho hàng: {(MaKho == 0 ? "Tất cả kho" : drKho?.Ten ?? "")}"

395 │ }

396 │ };

397 │

398 │

399 │ string fileName = FileType == "Excel" ? "BaoCaoDoanhThuTheoNgay.xlsx" : "BaoCaoDoanhThuTheoNgay.pdf";

400 │ string contentType = FileType == "Excel" ? "application/ms-excel" : "application/pdf";

401 │

402 │ string fileTemplate = Path.Combine(\_hostingEnvironment.ContentRootPath, "wwwroot", "templates", "BaoCaoDoanh...

403 │

404 │ int DongBatDau = 5;

405 │ Stream x = ExportToExcel.FromTemplate(fileTemplate, FileType, headerData, dt, true, DongBatDau, "A", ExportT...

406 │ return new FileStreamResult(x, contentType)

407 │ {

408 │ FileDownloadName = fileName

409 │ };

410 │ }

411 │ }

412 │ }

────────────────────────────────────────────────────────────

**📄 BaoCaoModel.cs (CS, 265 lines)**

1 │ ﻿using Microsoft.EntityFrameworkCore;

2 │ using QuanLyBanHang.Common;

3 │ using QuanLyBanHang.Models.ExModel.StoredProcedureModel;

4 │ using QuanLyBanHang.Models.QLBHEntity;

5 │ using System.Globalization;

6 │

7 │ namespace QuanLyBanHang.Models.ExModel.OPS

8 │ {

9 │ public class BaoCaoModel : Base<sp\_BaoCaoDoanhThuTheoNgay>

10 │ {

11 │ public BaoCaoModel(DbContext dbContext) : base(dbContext)

12 │ {

13 │ }

14 │

15 │ QuanLyBanHangContext dbContext

16 │ {

17 │ get { return \_dbContext as QuanLyBanHangContext; }

18 │ set { return; }

19 │ }

20 │ public class HoaDonTemp

21 │ {

22 │ public DateTime NgayHD { get; set; }

23 │ public Guid? NguoiThanhToan { get; set; }

24 │ public string SoSeri { get; set; }

25 │ public string MaVach { get; set; }

26 │ public int? MaHangHoa { get; set; }

27 │ public decimal SoLuong { get; set; }

28 │ public decimal? DonGia { get; set; }

29 │ public decimal? GiaGiam { get; set; }

30 │ public int? MaKho { get; set; }

31 │ public decimal SoLuongTra { get; set; }

32 │ public decimal SoLuongCK { get; set; }

33 │ public int HinhThucTt { get; set; }

34 │ public decimal KhuyenMaiTienMat { get; set; }

35 │ public string GhiChu { get; set; }

36 │ public string RequestID { get; set; }

37 │ public DateTime? NgayTao { get; set; }

38 │ }

39 │

40 │ public List<sp\_BaoCaoDoanhThuTheoNgay> GetBaoCaoDoanhThuTheoNgay(string tuNgay, string denNgay)

41 │ {

42 │ var ngayBatDau = DateTime.ParseExact(tuNgay, "yyyy/MM/dd", CultureInfo.InvariantCulture);

43 │ var ngayKetThuc = DateTime.ParseExact(denNgay, "yyyy/MM/dd", CultureInfo.InvariantCulture);

44 │

45 │ var query = from hd in dbContext.OpsHoaDon

46 │ join ct in dbContext.OpsChiTietHoaDon on hd.Ma equals ct.MaHoaDon

47 │ join hh in dbContext.TblHangHoa on ct.MaHangHoa equals hh.Ma into hhGroup

48 │ from hh in hhGroup.DefaultIfEmpty()

49 │ join lh in dbContext.TblLopHangHoa on hh.MaLopHang equals lh.Ma into lhGroup

50 │ from lh in lhGroup.DefaultIfEmpty()

51 │ join dv in dbContext.TblDonViTinh on hh.MaDvt equals dv.Ma into dvGroup

52 │ from dv in dvGroup.DefaultIfEmpty()

53 │ where hd.NgayHoaDon >= ngayBatDau && hd.NgayHoaDon <= ngayKetThuc

54 │ && !(hd.Deleted ?? false)

55 │ select new sp\_BaoCaoDoanhThuTheoNgay

56 │ {

57 │ TuNgay = tuNgay,

58 │ DenNgay = denNgay,

59 │ NgayHoaDon = hd.NgayHoaDon,

60 │ LopHang = hh.MaLopHang,

61 │ TenLop = lh.Ten,

62 │ SoSeri = hd.SoSeri,

63 │ MaVach = ct.MaVach,

64 │ TenHang = hh.Ten,

65 │ Dvt = hh.MaDvt,

66 │ TenDVT = dv.Ten,

67 │ SoLuong = ct.SoLuong ?? 0,

68 │ DonGia = ct.DonGia ?? 0,

69 │ ThanhTien = ct.SoLuong \* ct.DonGia,

70 │ GiamGia = (ct.SoLuong \* ct.DonGia) \* hd.GiamGia / 100,

71 │ TongCong = (ct.SoLuong \* ct.DonGia) - ((ct.SoLuong \* ct.DonGia) \* hd.GiamGia / 100),

72 │ HinhThucTT = hd.HinhThucTt

73 │ };

74 │

75 │ return query.ToList();

76 │ }

77 │

78 │ public List<sp\_BaoCaoDoanhThuTheoNguoiBanTheoCa> BaoCaoDoanhThuTheoNguoiBanTheoCa(DateTime TuNgay, DateTime DenN...

79 │ {

80 │ var list1 = (

81 │ from hd in dbContext.OpsHoaDon

82 │ join ct in dbContext.OpsChiTietHoaDon on hd.Ma equals ct.MaHoaDon

83 │ where !(hd.Deleted ?? false) && !(ct.Deleted ?? false)

84 │ && hd.NgayHoaDon.Date >= TuNgay.Date && hd.NgayHoaDon.Date <= DenNgay.Date

85 │ && (NguoiBan == Utility.defaultUID || hd.NguoiThanhToan == NguoiBan)

86 │ && (MaKho == 0 || hd.MaKho == MaKho)

87 │ && (HinhThucTT == 0 || hd.HinhThucTt == HinhThucTT)

88 │ && hd.HinhThucTt != (int)HinhThucThanhToan.ChuyenKhoan

89 │ select new HoaDonTemp

90 │ {

91 │ NgayHD = hd.NgayHoaDon.Date,

92 │ NguoiThanhToan = hd.NguoiThanhToan,

93 │ SoSeri = hd.SoSeri,

94 │ MaVach = ct.MaVach,

95 │ MaHangHoa = ct.MaHangHoa,

96 │ SoLuong = ct.SoLuong ?? 0,

97 │ DonGia = ct.DonGia,

98 │ GiaGiam = ct.GiaGiam,

99 │ MaKho = hd.MaKho,

100 │ SoLuongTra = 0,

101 │ SoLuongCK = 0,

102 │ HinhThucTt = hd.HinhThucTt,

103 │ KhuyenMaiTienMat = hd.KhuyenMaiTienMat ?? 0,

104 │ GhiChu = hd.GhiChu,

105 │ RequestID = "",

106 │ NgayTao = hd.CreatedDate

107 │ }).ToList();

108 │

109 │ var list2 = (

110 │ from hd in dbContext.OpsHoaDon

111 │ join ct in dbContext.OpsChiTietHoaDon on hd.Ma equals ct.MaHoaDon

112 │ where !(hd.Deleted ?? false) && !(ct.Deleted ?? false)

113 │ && hd.NgayHoaDon.Date >= TuNgay.Date && hd.NgayHoaDon.Date <= DenNgay.Date

114 │ && (NguoiBan == Utility.defaultUID || hd.NguoiThanhToan == NguoiBan)

115 │ && (MaKho == 0 || hd.MaKho == MaKho)

116 │ && (HinhThucTT == 0 || hd.HinhThucTt == HinhThucTT)

117 │ && hd.HinhThucTt == (int)HinhThucThanhToan.ChuyenKhoan

118 │ select new HoaDonTemp

119 │ {

120 │ NgayHD = hd.NgayHoaDon.Date,

121 │ NguoiThanhToan = hd.NguoiThanhToan,

122 │ SoSeri = hd.SoSeri,

123 │ MaVach = ct.MaVach + "",

124 │ MaHangHoa = ct.MaHangHoa,

125 │ SoLuong = 0,

126 │ DonGia = ct.DonGia,

127 │ GiaGiam = ct.GiaGiam,

128 │ MaKho = hd.MaKho,

129 │ SoLuongTra = 0,

130 │ SoLuongCK = ct.SoLuong ?? 0,

131 │ HinhThucTt = hd.HinhThucTt,

132 │ KhuyenMaiTienMat = hd.KhuyenMaiTienMat ?? 0,

133 │ GhiChu = hd.GhiChu + "",

134 │ RequestID = hd.RequestId ?? "",

135 │ NgayTao = hd.CreatedDate

136 │ }).ToList();

137 │

138 │ var list3 = (

139 │ from hd in dbContext.OpsHoaDon

140 │ join ct in dbContext.OpsChiTietHoaDon on hd.Ma equals ct.MaHoaDon

141 │ where !(hd.Deleted ?? false) && !(ct.Deleted ?? false)

142 │ && hd.NgayHoaDon.Date >= TuNgay.Date && hd.NgayHoaDon.Date <= DenNgay.Date

143 │ && (NguoiBan == Utility.defaultUID || ct.NguoiTra == NguoiBan)

144 │ && (MaKho == 0 || hd.MaKho == MaKho)

145 │ && (ct.SoLuongTraLai ?? 0) > 0

146 │ && ct.NgayTra != null

147 │ select new HoaDonTemp

148 │ {

149 │ NgayHD = ct.NgayTra.Value.Date,

150 │ NguoiThanhToan = ct.NguoiTra,

151 │ SoSeri = hd.SoSeri,

152 │ MaVach = ct.MaVach + "",

153 │ MaHangHoa = ct.MaHangHoa,

154 │ SoLuong = 0,

155 │ DonGia = ct.DonGia,

156 │ GiaGiam = ct.GiaGiam,

157 │ MaKho = hd.MaKho,

158 │ SoLuongTra = ct.SoLuongTraLai ?? 0,

159 │ SoLuongCK = 0,

160 │ HinhThucTt = hd.HinhThucTt,

161 │ KhuyenMaiTienMat = 0,

162 │ GhiChu = "",

163 │ RequestID = "",

164 │ NgayTao = ct.NgayTra

165 │ }).ToList();

166 │

167 │ var lstHoaDon = list1.Concat(list2).Concat(list3).ToList();

168 │

169 │

170 │ var data = lstHoaDon

171 │ .GroupBy(t => new

172 │ {

173 │ t.NgayHD,

174 │ t.NguoiThanhToan,

175 │ t.SoSeri,

176 │ t.MaHangHoa,

177 │ t.MaVach,

178 │ t.MaKho,

179 │ t.HinhThucTt,

180 │ t.KhuyenMaiTienMat,

181 │ t.GhiChu,

182 │ t.RequestID,

183 │ t.NgayTao

184 │ })

185 │ .Select(g => new

186 │ {

187 │ NgayHD = g.Key.NgayHD,

188 │ NguoiThanhToan = g.Key.NguoiThanhToan,

189 │ SoSeri = g.Key.SoSeri,

190 │ MaHangHoa = g.Key.MaHangHoa,

191 │ MaVach = g.Key.MaVach,

192 │ MaKho = g.Key.MaKho,

193 │ HinhThucTt = g.Key.HinhThucTt,

194 │ SoLuong = (double)g.Sum(x => x.SoLuong),

195 │ DonGia = (double)(g.Max(x => x.DonGia) ?? 0),

196 │ SoLuongTra = (double)g.Sum(x => x.SoLuongTra),

197 │ SoLuongCK = (double)g.Sum(x => x.SoLuongCK),

198 │ GiaGiam = (double)(g.Max(x => x.GiaGiam ?? 0)),

199 │ KhuyenMaiTienMat = (double)g.Key.KhuyenMaiTienMat,

200 │ GhiChu = g.Key.GhiChu,

201 │ RequestID = g.Key.RequestID,

202 │ NgayTao = g.Key.NgayTao

203 │ })

204 │ .ToList();

205 │

206 │

207 │ List<VHangHoa> listHH = new List<VHangHoa>();

208 │ List<SysUser> listUser = new List<SysUser>();

209 │ List<StockDanhMucKho> listKho = new List<StockDanhMucKho>();

210 │ int loop = data.Count / 2000;

211 │ if (loop \* 2000 < data.Count) loop += 1;

212 │ for (int i = 0; i < loop; i++)

213 │ {

214 │ listHH.AddRange(dbContext.VHangHoa.Where(t => data.Skip(i \* 2000).Take(2000).Select(x => x.MaHangHoa).Co...

215 │ listUser.AddRange(dbContext.SysUser.Where(t => data.Skip(i \* 2000).Take(2000).Select(x => x.NguoiThanhTo...

216 │ listKho.AddRange(dbContext.StockDanhMucKho.Where(t => data.Skip(i \* 2000).Take(2000).Select(x => x.MaKho...

217 │ }

218 │ List<TblHinhThucThanhToan> listHTTT = dbContext.TblHinhThucThanhToan.ToList();

219 │ SysUser u = new SysUser();

220 │ VHangHoa hh = new VHangHoa();

221 │ StockDanhMucKho kho = new StockDanhMucKho();

222 │ return data.Select(t => new sp\_BaoCaoDoanhThuTheoNguoiBanTheoCa

223 │ {

224 │ TuNgay = TuNgay.Date,

225 │ DenNgay = DenNgay.Date,

226 │ NgayHoaDon = t.NgayHD,

227 │ NguoiBan = t.NguoiThanhToan,

228 │ Username = (u = listUser.FirstOrDefault(x => x.Ma == t.NguoiThanhToan) ?? new SysUser()).Username + "",

229 │ Fullname = u.FullName + "",

230 │ TenHang = (hh = listHH.FirstOrDefault(x => x.Ma == t.MaHangHoa) ?? new VHangHoa()).TenHangHoa + "",

231 │ LopHang = hh.MaLopHang,

232 │ TenLop = hh.TenLopHang + "",

233 │ SoSeri = t.SoSeri,

234 │ MaVach = t.MaVach,

235 │ MaKT = hh.MaKt + "",

236 │ Dvt = hh.MaDvt ?? 0,

237 │ TenDVT = hh.TenDvt + "",

238 │ MaKho = t.MaKho,

239 │ TenKho = (listKho.FirstOrDefault(x => x.Ma == t.MaKho) ?? new StockDanhMucKho()).Ten + "",

240 │ SoLuong = (int)(t.SoLuong + t.SoLuongCK),

241 │ DonGia = (int)t.DonGia,

242 │ SoLuongTra = (int)t.SoLuongTra,

243 │ ThanhTien = (decimal)((t.SoLuong + t.SoLuongCK) \* t.DonGia),

244 │ GiamGia = (decimal)(((t.SoLuong + t.SoLuongCK) \* t.DonGia) \* t.GiaGiam / 100.0),

245 │ ThanhTienTra = (decimal)(t.SoLuongTra \* t.DonGia \* (100.0 - t.GiaGiam) / 100.0),

246 │ TienChuyenKhoan = (decimal)(t.SoLuongCK \* t.DonGia),

247 │ TongCong = (decimal)

248 │ (

249 │ ((t.SoLuong + t.SoLuongCK) \* t.DonGia) //Thành tiền

250 │ - (((t.SoLuong + t.SoLuongCK) \* t.DonGia) \* t.GiaGiam / 100.0) //Giảm giá

251 │ - (t.SoLuongTra \* t.DonGia \* (100.0 - t.GiaGiam) / 100.0) //Trả hàng

252 │ - (t.SoLuongCK \* t.DonGia) //Chuyển khoản

253 │ - (t.KhuyenMaiTienMat == 0 ? 0 : (int)(t.KhuyenMaiTienMat / data.Where(x => x.SoSeri == t.So...

254 │ ),

255 │ MaHinhThuc = t.HinhThucTt,

256 │ TenHinhThuc = (listHTTT.FirstOrDefault(x => x.Ma == t.HinhThucTt) ?? new TblHinhThucThanhToan()).Ten + "...

257 │ KhuyenMaiTienMat = t.KhuyenMaiTienMat == 0 ? 0 : (int)(t.KhuyenMaiTienMat / data.Where(x => x.SoSeri == ...

258 │ GhiChu = t.GhiChu,

259 │ RequestID = t.RequestID,

260 │ NgayTao = t.NgayTao

261 │ }).ToList();

262 │ }

263 │

264 │ }

265 │ }

────────────────────────────────────────────────────────────

**📄 BaoCaoVatTuController.cs (CS, 120 lines)**

1 │ ﻿using System.Linq;

2 │ using Microsoft.AspNetCore.Mvc;

3 │ using QuanLyBanHang.Common;

4 │ using QuanLyBanHang.Common.Attributes;

5 │ using QuanLyBanHang.Models.APIModels;

6 │ using QuanLyBanHang.Models.ExModel.CAT;

7 │ using QuanLyBanHang.Models.ExModel.OPS;

8 │ using QuanLyBanHang.Models.QLBHEntity;

9 │

10 │ namespace QuanLyBanHang.Controllers.OPS

11 │ {

12 │ public class BaoCaoVatTuController : BaseController

13 │ {

14 │ public BaoCaoVatTuController(QuanLyBanHangContext dbcontext, MemoryCacheManager memoryCache) : base(dbcontext, m...

15 │ {

16 │ m\_MenuID = MenuID.rptBaoCaoVatTu.GetEnumGuid();

17 │ }

18 │ [TokenUserCheckHTTP]

19 │ public IActionResult Index()

20 │ {

21 │ ViewData["ListKho"] = new DanhMucKhoModel(dbcontext).GetDataByUser(objToken.userid, objToken.branch);

22 │ return View();

23 │ }

24 │ [HttpPost]

25 │ public IActionResult getListData(int maKho, int nam, int thang)

26 │ {

27 │

28 │ try

29 │ {

30 │ var tieuHaoDinhMucQuery = from hd in dbcontext.OpsHoaDon

31 │ where hd.MaKho == maKho &&

32 │ hd.NgayHoaDon.Year == nam &&

33 │ hd.NgayHoaDon.Month == thang

34 │ join cthd in dbcontext.OpsChiTietHoaDon on hd.Ma equals cthd.MaHoaDon

35 │ join a in dbcontext.TblHangHoa on cthd.MaHangHoa equals a.Ma

36 │ join b in dbcontext.VtthDinhMuc on a.MaDinhMuc equals b.Ma

37 │ join c in dbcontext.VtthChiTietDinhMuc on b.Ma equals c.MaDinhMuc

38 │ group new { cthd, c } by c.MaVatTu into g

39 │ select new

40 │ {

41 │ MaVatTu = g.Key,

42 │ SoLuongTieuHaoDinhMuc = g.Sum(x => x.cthd.SoLuong \* x.c.SoLuong)

43 │ };

44 │

45 │ var tieuHaoThucTeQuery = from d in dbcontext.VtthLichSuXuatKhay

46 │ where d.MaKhoXuat == maKho &&

47 │ d.NgayXuat.HasValue &&

48 │ d.NgayXuat.Value.Year == nam &&

49 │ d.NgayXuat.Value.Month == thang

50 │ group d by d.MaVatTu into g

51 │ select new

52 │ {

53 │ MaVatTu = g.Key,

54 │ SoLuongTieuHaoThucTe = g.Sum(x => x.SoLuongXuat)

55 │ };

56 │

57 │ var dinhMucList = tieuHaoDinhMucQuery.ToList();

58 │ var thucTeList = tieuHaoThucTeQuery.ToList();

59 │

60 │ var allMaVatTu = dinhMucList.Select(x => x.MaVatTu)

61 │ .Union(thucTeList.Select(x => x.MaVatTu))

62 │ .Distinct()

63 │ .ToList();

64 │

65 │ var hangHoaDict = (from hh in dbcontext.TblHangHoa

66 │ join dvt in dbcontext.TblDonViTinh on hh.MaDvt equals dvt.Ma into gj

67 │ from dvt in gj.DefaultIfEmpty()

68 │ where allMaVatTu.Contains(hh.Ma)

69 │ select new

70 │ {

71 │ hh.Ma,

72 │ MaKT = hh.MaKt,

73 │ Ten = hh.Ten,

74 │ GiaVon = hh.GiaVon ?? 0,

75 │ TenDVT = dvt != null ? dvt.Ten : "(Chưa có DVT)"

76 │ }).ToDictionary(x => x.Ma, x => x);

77 │

78 │

79 │ var finalReport = allMaVatTu.Select(maVT =>

80 │ {

81 │ var dm = dinhMucList.FirstOrDefault(x => x.MaVatTu == maVT);

82 │ var tt = thucTeList.FirstOrDefault(x => x.MaVatTu == maVT);

83 │ var hh = hangHoaDict.ContainsKey(maVT ?? 0)

84 │ ? hangHoaDict[maVT ?? 0]

85 │ : new { Ma = 0, MaKT = "", Ten = "(Không rõ)", GiaVon = 0m, TenDVT = "(Không rõ)" };

86 │

87 │ var slDinhMuc = dm?.SoLuongTieuHaoDinhMuc ?? 0;

88 │ var slThucTe = tt?.SoLuongTieuHaoThucTe ?? 0;

89 │ var chenhLech = slThucTe - slDinhMuc;

90 │ var giaTriChenhLech = chenhLech \* hh.GiaVon;

91 │

92 │ return new

93 │ {

94 │ MaKT = hh.MaKT,

95 │ MaKho = maKho,

96 │ DVT = hh.TenDVT,

97 │ MaVatTu = maVT,

98 │ TenVatTu = hh.Ten,

99 │ TenDonViTinh = hh.TenDVT,

100 │ SoLuongTieuHaoDinhMuc = slDinhMuc,

101 │ SoLuongTieuHaoThucTe = slThucTe,

102 │ ChenhLech = chenhLech,

103 │ GiaVon = hh.GiaVon,

104 │ GiaTriChenhLech = giaTriChenhLech

105 │ };

106 │ })

107 │ .OrderBy(x => x.MaVatTu)

108 │ .ToList();

109 │

110 │

111 │ return Json(new ResponseModelSuccess("", finalReport));

112 │ }

113 │ catch (Exception ex)

114 │ {

115 │ return Json(new ResponseModelError(ex.Message));

116 │ }

117 │ }

118 │

119 │ }

120 │ }

────────────────────────────────────────────────────────────

**📄 BartenderController.cs (CS, 160 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using Microsoft.AspNetCore.SignalR;

3 │ using QuanLyBanHang.Common;

4 │ using QuanLyBanHang.Common.Attributes;

5 │ using QuanLyBanHang.Models.APIModels;

6 │ using QuanLyBanHang.Models.ExModel.CAT;

7 │ using QuanLyBanHang.Models.ExModel.OPS;

8 │ using QuanLyBanHang.Models.ExModel.SYS;

9 │ using QuanLyBanHang.Models.QLBHEntity;

10 │ using static QuanLyBanHang.Models.ExModel.OPS.OrderDetailModel;

11 │

12 │ namespace QuanLyBanHang.Controllers.OPS

13 │ {

14 │ public class BartenderController : BaseController

15 │ {

16 │ private readonly IHubContext<ChatHUB> \_chatHub;

17 │ public BartenderController(QuanLyBanHangContext dbcontext, MemoryCacheManager memoryCache, IHubContext<ChatHUB> ...

18 │ {

19 │ \_chatHub = chatHub;

20 │ m\_MenuID = MenuID.opsBartender.GetEnumGuid();

21 │ }

22 │

23 │ [TokenUserCheckHTTP]

24 │ public ActionResult Index(string MaKhoHang)

25 │ {

26 │ int MaKho = Cast.ToInt((MaKhoHang + "").Decrypt());

27 │ if (MaKho <= 0)

28 │ {

29 │ return RedirectToAction("ChonKho");

30 │ }

31 │ var user = new sys\_UserModel(dbcontext).GetItem(objToken.userid);

32 │ string[] strKhoHang = (user.KhoHang + "").Split(';');

33 │ if (!strKhoHang.Contains(Cast.ToInt(MaKho).ToString()))

34 │ {

35 │ return RedirectToAction("ChonKho");

36 │ }

37 │

38 │ DateTime NgayHD = new NgayKetCaModel(dbcontext).LayNgayKetCa(MaKho);

39 │ ViewData["MaKho"] = MaKho;

40 │ ViewData["TenKho"] = new DanhMucKhoModel(dbcontext).GetItem(MaKho).Ten;

41 │

42 │ return View();

43 │ }

44 │ [TokenUserCheckHTTP]

45 │ public ActionResult ChonKho()

46 │ {

47 │ var user = new sys\_UserModel(dbcontext).GetItem(objToken.userid);

48 │

49 │

50 │ string[] strKhoHang = (user.KhoHang + "").Split(';');

51 │ List<int> lstIDKhoHang = new List<int>();

52 │ foreach (string item in strKhoHang)

53 │ {

54 │ if (Cast.ToInt(item.Trim()) > 0)

55 │ {

56 │ lstIDKhoHang.Add(Cast.ToInt(item.Trim()));

57 │ }

58 │ }

59 │

60 │ ViewData["ListKhoHang"] = new DanhMucKhoModel(dbcontext).GetDataByIds(lstIDKhoHang, objToken.branch).Where(t...

61 │

62 │ return View("ChonKho");

63 │ }

64 │ [HttpPost]

65 │ [TokenUserCheckHTTP]

66 │ public ActionResult getOrderList(string MaKho)

67 │ {

68 │ var orderList = (from o in dbcontext.OpsOrder.Where(x => !(x.DaThucHien ?? false) && x.MaPhong != null)

69 │ join od in dbcontext.OpsOrderDetail.Where(x => x.MaKho == Cast.ToInt(MaKho) && x.TrangThai ...

70 │ join p in dbcontext.TblPhongBenh.Where(x => !(x.Deleted ?? false)) on o.MaPhong equals p.Ma

71 │ select new

72 │ {

73 │ MaDonHang = o.Ma,

74 │ MaDonHangChiTiet = od.Ma,

75 │ MaPhong = o.MaPhong,

76 │ TenPhong = p.Ten,

77 │ TenKH = o.TenKh,

78 │ TrangThai = od.TrangThai,

79 │ ThoiGianDatHang = od.ThoiGianDatMon == null ? null : od.ThoiGianDatMon.Value.ToString("...

80 │ ThoiGianXacNhan = od.ThoiGianXacNhan == null ? null : od.ThoiGianXacNhan.Value.ToString...

81 │ }).ToList();

82 │ var listData = orderList.Select(t => new

83 │ {

84 │ MaDonHang = t.MaDonHang,

85 │ MaDonHangChiTiet = t.MaDonHangChiTiet,

86 │ MaPhong = t.MaPhong,

87 │ TenPhong = t.TenPhong,

88 │ TenKH = t.TenKH,

89 │ TrangThai = t.TrangThai,

90 │ DanhSachMon = getDanhSachMon(t.MaDonHangChiTiet),

91 │ TongTien = getDanhSachMon(t.MaDonHangChiTiet).Sum(x => (x.GiaBan ?? 0) \* (x.SoLuong ?? 0)),

92 │ ThoiGianDatHang = t.ThoiGianDatHang,

93 │ ThoiGianXacNhan = t.ThoiGianXacNhan,

94 │ }).ToList();

95 │ return Json(new ResponseModel(QLBHConst.OK, "", listData));

96 │ }

97 │ private List<OrderItemModel> getDanhSachMon(string MaDonHangChiTiet)

98 │ {

99 │ var listData = (from oi in dbcontext.OpsOrderItems.Where(x => x.MaDonHangChiTiet == MaDonHangChiTiet)

100 │ join hh in dbcontext.TblHangHoa.Where(x => !(x.Deleted ?? false)) on oi.MaHangHoa equals hh....

101 │ select new OrderItemModel

102 │ {

103 │ Ma = oi.Ma,

104 │ MaDonHang = oi.MaDonHang,

105 │ MaDonHangChiTiet = oi.MaDonHangChiTiet,

106 │ MaHangHoa = oi.MaHangHoa,

107 │ TenHangHoa = hh.Ten,

108 │ HinhAnh = hh.HinhAnh,

109 │ GiaBan = oi.GiaBan,

110 │ SoLuong = oi.SoLuong,

111 │ ThanhTien = oi.ThanhTien,

112 │ GhiChu = oi.GhiChu,

113 │ DaChuanBi = oi.DaChuanBi,

114 │ }).ToList();

115 │ return listData;

116 │ }

117 │ [HttpPost]

118 │ [TokenUserCheckHTTP]

119 │ public ActionResult setReadyOrder(string MaDonHangChiTiet)

120 │ {

121 │ var orderDetail = dbcontext.OpsOrderDetail.FirstOrDefault(x => x.Ma == MaDonHangChiTiet);

122 │ if (orderDetail == null) return Json(new ResponseModelError("Đơn hàng chi tiết không tồn tại"));

123 │

124 │ var order = dbcontext.OpsOrder.FirstOrDefault(x => x.Ma == orderDetail.MaDonHang);

125 │ var orderItems = dbcontext.OpsOrderItems.Where(x => x.MaDonHangChiTiet == MaDonHangChiTiet).ToList();

126 │ orderDetail.TrangThai = (int)OrderStatus.Ready;

127 │ dbcontext.Update(orderDetail);

128 │ dbcontext.SaveChanges();

129 │

130 │ orderItems.ForEach(item => item.DaChuanBi = true);

131 │ dbcontext.UpdateRange(orderItems);

132 │ dbcontext.SaveChanges();

133 │ \_chatHub.Clients.Group(order.Uuid + "").SendAsync("readyOrder", new

134 │ {

135 │ MaDonHang = orderDetail.Ma,

136 │ MaKho = orderDetail.MaKho,

137 │

138 │ });

139 │ \_chatHub.Clients.Group("OrderGroup").SendAsync("readyOrder", new

140 │ {

141 │ MaDonHang = orderDetail.Ma,

142 │ MaKho = orderDetail.MaKho,

143 │

144 │ });

145 │ return Json(new ResponseModelSuccess("Đã chuẩn bị đơn hàng thành công"));

146 │ }

147 │

148 │ [HttpPost]

149 │ [TokenUserCheckHTTP]

150 │ public ActionResult setReadyOrderItem(string MaDonHangChiTiet, string MaMon, bool TrangThai)

151 │ {

152 │ var orderItem = dbcontext.OpsOrderItems.FirstOrDefault(x => x.MaDonHangChiTiet == MaDonHangChiTiet && x.Ma =...

153 │ orderItem.DaChuanBi = TrangThai;

154 │ dbcontext.Update(orderItem);

155 │ dbcontext.SaveChanges();

156 │

157 │ return Json(new ResponseModelSuccess("Đã chuẩn bị món thành công"));

158 │ }

159 │ }

160 │ }

────────────────────────────────────────────────────────────

**📄 Base.cs (CS, 82 lines)**

1 │ ﻿using System;

2 │ using System.Linq.Expressions;

3 │ using Microsoft.EntityFrameworkCore;

4 │

5 │ namespace QuanLyBanHang.Models

6 │ {

7 │ public class Base<T> : IDisposable where T : class

8 │ {

9 │ protected readonly DbContext \_dbContext;

10 │ public Base(DbContext dbContext)

11 │ {

12 │ \_dbContext = dbContext;

13 │ }

14 │

15 │ public T First(Expression<Func<T, bool>> predicate)

16 │ {

17 │ try

18 │ {

19 │ return \_dbContext.Set<T>().First(predicate);

20 │ }

21 │ catch

22 │ {

23 │ return null;

24 │ }

25 │ }

26 │

27 │ public int Add(T item, bool isE = false)

28 │ {

29 │ try

30 │ {

31 │ \_dbContext.Set<T>().Add(item);

32 │ return \_dbContext.SaveChanges() > 0 ? 1 : 0;

33 │ }

34 │ catch (Exception e)

35 │ {

36 │ return -1;

37 │ }

38 │ }

39 │

40 │ public int Update(T item, bool isE = false, bool v = false)

41 │ {

42 │ try

43 │ {

44 │ \_dbContext.Set<T>().Update(item);

45 │ return \_dbContext.SaveChanges() > 0 ? 1 : 0;

46 │ }

47 │ catch

48 │ {

49 │ return -1;

50 │ }

51 │ }

52 │ public IQueryable<T> Get(Expression<Func<T, bool>> predicate)

53 │ {

54 │ return \_dbContext.Set<T>().Where(predicate);

55 │ }

56 │ public IQueryable<T> Get()

57 │ {

58 │ return \_dbContext.Set<T>().Where(t => true);

59 │ }

60 │ public int Delete(T item, bool isE = false, bool v = false)

61 │ {

62 │ try

63 │ {

64 │ \_dbContext.Set<T>().Remove(item);

65 │ return \_dbContext.SaveChanges() > 0 ? 1 : 0;

66 │ }

67 │ catch

68 │ {

69 │ return -1;

70 │ }

71 │ }

72 │ public bool Any(Expression<Func<T, bool>> predicate)

73 │ {

74 │ return \_dbContext.Set<T>().Any(predicate);

75 │ }

76 │

77 │ public void Dispose()

78 │ {

79 │ \_dbContext.Dispose();

80 │ }

81 │ }

82 │ }

────────────────────────────────────────────────────────────

**📄 BaseController.cs (CS, 64 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using QuanLyBanHang.Common;

3 │ using QuanLyBanHang.Common.Attributes;

4 │ using QuanLyBanHang.Models.APIModels;

5 │ using QuanLyBanHang.Models.ExModel.SYS;

6 │ using QuanLyBanHang.Models.QLBHEntity;

7 │

8 │ namespace QuanLyBanHang.Controllers

9 │ {

10 │ [LogAction]

11 │ public abstract class BaseController : Controller

12 │ {

13 │ private UserToken? \_objToken;

14 │ public UserToken? objToken

15 │ {

16 │ get => \_objToken;

17 │ set

18 │ {

19 │ \_objToken = value;

20 │ dbcontext.userId = \_objToken.userid;

21 │ }

22 │ }

23 │ public Guid m\_MenuID;

24 │ public bool isAPI;

25 │

26 │ public bool isAdmin

27 │ {

28 │ get

29 │ {

30 │ try

31 │ {

32 │ return dbcontext.VPermission.Any(t => t.RoleId == RoleID.Admin.GetEnumGuid() && t.MenuId == m\_MenuID...

33 │ }

34 │ catch

35 │ {

36 │ }

37 │ return false;

38 │ }

39 │ }

40 │ public List<VPermission> GetVPermissions(Guid? MenuId = null)

41 │ {

42 │ Guid menuID = MenuId ?? m\_MenuID;

43 │ return dbcontext.VPermission.Where(t=> t.MenuId == menuID && t.UserId == \_objToken.userid && t.BranchId == o...

44 │ }

45 │ public QuanLyBanHangContext dbcontext { get; }

46 │ public MemoryCacheManager memoryCache { get; }

47 │

48 │ public BaseController(QuanLyBanHangContext dbcontext, MemoryCacheManager memoryCache)

49 │ {

50 │ this.dbcontext = dbcontext;

51 │ this.memoryCache = memoryCache;

52 │ }

53 │

54 │ protected override void Dispose(bool disposing)

55 │ {

56 │ if (disposing)

57 │ {

58 │ //dbcontext.Dispose();

59 │ GC.Collect();

60 │ }

61 │ base.Dispose(disposing);

62 │ }

63 │ }

64 │ }

────────────────────────────────────────────────────────────

**📄 BasicAuthCheck.cs (CS, 33 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using Microsoft.AspNetCore.Mvc.Filters;

3 │ using QuanLyBanHang.Controllers;

4 │ using System.Text;

5 │

6 │ namespace QuanLyBanHang.Common.Attributes

7 │ {

8 │ public class BasicAuthCheck : ActionFilterAttribute

9 │ {

10 │ private string client\_id = AppController.passcode;

11 │ private string client\_secret = AppController.passcode;

12 │ public BasicAuthCheck()

13 │ {

14 │ }

15 │

16 │ public BasicAuthCheck(string client\_id, string client\_secret)

17 │ {

18 │ this.client\_id = client\_id;

19 │ this.client\_secret = client\_secret;

20 │ }

21 │

22 │ public override void OnActionExecuting(ActionExecutingContext context)

23 │ {

24 │ var byteArray = Encoding.ASCII.GetBytes($"{client\_id}:{client\_secret}");

25 │ string apikey = string.Format("Basic {0}", Convert.ToBase64String(byteArray));

26 │ if (!(context.HttpContext.Request.Headers["Authorization"] == apikey))

27 │ {

28 │ context.Result = new UnauthorizedResult();

29 │ }

30 │ base.OnActionExecuting(context);

31 │ }

32 │ }

33 │ }

────────────────────────────────────────────────────────────

**📄 BearerAuthCheckAttribute.cs (CS, 61 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using Microsoft.AspNetCore.Mvc.Filters;

3 │ using QuanLyBanHang.Controllers;

4 │ using System.Text;

5 │

6 │ namespace QuanLyBanHang.Common.Attributes

7 │ {

8 │ public class BearerAuthCheckAttribute : ActionFilterAttribute

9 │ {

10 │ private string tokenKey = AppController.passcode;

11 │ private long expireTime = 60;

12 │ public BearerAuthCheckAttribute()

13 │ {

14 │ }

15 │

16 │ public BearerAuthCheckAttribute(string tokenKey)

17 │ {

18 │ this.tokenKey = tokenKey;

19 │ }

20 │ public BearerAuthCheckAttribute(string tokenKey, long expireTime)

21 │ {

22 │ this.tokenKey = tokenKey;

23 │ this.expireTime = expireTime;

24 │ }

25 │ public override void OnActionExecuting(ActionExecutingContext context)

26 │ {

27 │ try

28 │ {

29 │ string token = context.HttpContext.Request.Headers["Authorization"].ToString().Replace("Bearer ", "");

30 │ byte[] data = Convert.FromBase64String(token);

31 │ long timestamp = BitConverter.ToInt64(data, 0);

32 │ DateTime creationTime = DateTime.FromBinary(timestamp);

33 │ byte[] extractedKey = data.Skip(sizeof(long)).ToArray();

34 │ string key = Encoding.UTF8.GetString(extractedKey).Remove(0, 1);

35 │ if (key != Utility.GetSHA512(string.Concat(tokenKey, creationTime.ToString("yyyyMMddHHmmss"))))

36 │ {

37 │ context.Result = new UnauthorizedResult();

38 │ }

39 │ if (creationTime < DateTime.UtcNow.AddSeconds(-expireTime))

40 │ {

41 │ context.Result = new UnauthorizedResult();

42 │ }

43 │ }

44 │ catch (Exception ex)

45 │ {

46 │ context.Result = new UnauthorizedResult();

47 │ }

48 │ base.OnActionExecuting(context);

49 │ }

50 │

51 │ public static string GetToken(string tokenKey)

52 │ {

53 │ DateTime now = DateTime.UtcNow;

54 │ byte[] time = BitConverter.GetBytes(now.ToBinary());

55 │ string key = Utility.RandomString(1) + Utility.GetSHA512(string.Concat(tokenKey, now.ToString("yyyyMMddHHmms...

56 │ byte[] keyByte = Encoding.UTF8.GetBytes(key);

57 │ string token = Convert.ToBase64String(time.Concat(keyByte).ToArray());

58 │ return token;

59 │ }

60 │ }

61 │ }

────────────────────────────────────────────────────────────

**📄 BranchController.cs (CS, 226 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using QuanLyBanHang.Common;

3 │ using QuanLyBanHang.Common.Attributes;

4 │ using QuanLyBanHang.Models.APIModels;

5 │ using QuanLyBanHang.Models.QLBHEntity;

6 │

7 │ namespace QuanLyBanHang.Controllers.SYS

8 │ {

9 │ public class BranchController : BaseController

10 │ {

11 │ public BranchController(QuanLyBanHangContext dbcontext, MemoryCacheManager memoryCache) : base(dbcontext, memory...

12 │ {

13 │ m\_MenuID = MenuID.sysBranch.GetEnumGuid();

14 │ }

15 │

16 │ [HttpPost]

17 │ [TokenUserCheckHTTP(Permission.NotCheck)]

18 │ public ActionResult getBranch()

19 │ {

20 │ List<SysBranch> sysBranches = dbcontext.SysBranch.Where(x => !(x.Deleted ?? false)).ToList();

21 │ return Json(new ResponseModel(QLBHConst.OK, "", sysBranches.Select(i => new

22 │ {

23 │ ma = i.Ma,

24 │ code = i.Code,

25 │ ten = i.Ten,

26 │ diachi = i.DiaChi,

27 │ trangthai = i.Status,

28 │ })));

29 │ }

30 │

31 │ [HttpPost]

32 │ [TokenUserCheckHTTP(Permission.NotCheck)]

33 │ public ActionResult getBranchDangNhap()

34 │ {

35 │ List<SysBranch> sysBranches = dbcontext.SysBranch.Where(x => !(x.Deleted ?? false) && x.Ma == objToken.branc...

36 │ return Json(new ResponseModel(QLBHConst.OK, "", sysBranches.Select(i => new

37 │ {

38 │ ma = i.Ma,

39 │ code = i.Code,

40 │ ten = i.Ten,

41 │ diachi = i.DiaChi,

42 │ trangthai = i.Status,

43 │ })));

44 │ }

45 │

46 │ [HttpPost]

47 │ [TokenUserCheckHTTP(Permission.NotCheck)]

48 │ public ActionResult getBranchByRoleUser()

49 │ {

50 │ List<SysUserRole> lstUserRole = dbcontext.SysUserRole.Where(t => t.UserId == objToken.userid && !(t.Deleted ...

51 │ List<SysBranch> sysBranches = dbcontext.SysBranch.Where(x => lstUserRole.Select(t => t.BranchId).Contains(x....

52 │ return Json(new ResponseModel(QLBHConst.OK, "", sysBranches.Select(i => new

53 │ {

54 │ ma = i.Ma,

55 │ code = i.Code,

56 │ ten = i.Ten,

57 │ diachi = i.DiaChi,

58 │ trangthai = i.Status,

59 │ })));

60 │ }

61 │

62 │ [TokenUserCheckHTTP]

63 │ public ActionResult Index()

64 │ {

65 │ ViewData["ListData"] = dbcontext.SysBranch.Where(x => !(x.Deleted ?? false)).ToList();

66 │ return View();

67 │ }

68 │

69 │ [TokenUserCheckHTTP(Permission.Add)]

70 │ public ActionResult Add()

71 │ {

72 │ return View("Form");

73 │ }

74 │

75 │ [TokenUserCheckHTTP(Permission.Edit)]

76 │ public ActionResult Edit(string Ma)

77 │ {

78 │ ViewData["item"] = dbcontext.SysBranch.FirstOrDefault(x => x.Ma == Ma.GetGuid() && !(x.Deleted ?? false));

79 │ return View("Form");

80 │ }

81 │ private bool CheckExists(Guid Ma, string Code)

82 │ {

83 │ return dbcontext.SysBranch.Any(t => t.Ma != Ma && t.Code == Code && !(t.Deleted ?? false));

84 │ }

85 │ [HttpPost]

86 │ [TokenUserCheckHTTP(Permission.Add)]

87 │ public ActionResult AddAPI(string Code, string Ten, string PhapNhan, string DiaChi, string MaSoThue, string Tran...

88 │ {

89 │ if (CheckExists(Utility.defaultUID, Code))

90 │ {

91 │ return Json(new ResponseModel(QLBHConst.Error, "Trùng mã chi nhánh"));

92 │ }

93 │

94 │ if (CheckExists(Utility.defaultUID, SoTaiKhoan))

95 │ {

96 │ return Json(new ResponseModel(QLBHConst.Error, "Trùng mã tài khoản đơn vị"));

97 │ }

98 │ if (Code == null || Code.IsEmpty())

99 │ {

100 │ return Json(new ResponseModel(QLBHConst.Error, "Chưa có Code Chi nhánh"));

101 │ }

102 │

103 │

104 │ var item = new SysBranch

105 │ {

106 │ Code = Code,

107 │ Ten = Ten,

108 │ TenVietTat = TenVietTat,

109 │ SoTaiKhoan = SoTaiKhoan,

110 │ PhapNhan = PhapNhan,

111 │ DiaChi = DiaChi,

112 │ MaSoThue = MaSoThue,

113 │ Status = TrangThai == "1",

114 │ };

115 │ if (NgayKhoaSo != null)

116 │ {

117 │ DateTime NgayKhoaSoDate = NgayKhoaSo.ToDateTime2(DateTime.Now)!.Value;

118 │ item.NgayKhoaSo = NgayKhoaSoDate;

119 │ }

120 │

121 │ if (!Logo.IsEmpty())

122 │ {

123 │ item.Logo = Logo;

124 │ }

125 │ dbcontext.SysBranch.Add(item);

126 │ dbcontext.SaveChanges();

127 │ return Json(QLBHConst.ResponseAddSuccessfully);

128 │ }

129 │ [HttpPost]

130 │ [TokenUserCheckHTTP(Permission.Edit)]

131 │ public ActionResult UpdateAPI(string Ma, string Code, string Ten, string PhapNhan, string DiaChi, string MaSoThu...

132 │ {

133 │ if (CheckExists(Ma.GetGuid(), Code))

134 │ {

135 │ return Json(new ResponseModel(QLBHConst.Error, "Trùng mã chi nhánh"));

136 │ }

137 │ SysBranch item = dbcontext.SysBranch.FirstOrDefault(x => x.Ma == Ma.GetGuid() && !(x.Deleted ?? false));

138 │ if (item == null)

139 │ {

140 │ return Json(QLBHConst.ResponseNotExist);

141 │ }

142 │ if (!Logo.IsEmpty())

143 │ {

144 │ item.Logo = Logo;

145 │ }

146 │ if (CheckExists(Ma.GetGuid(), SoTaiKhoan))

147 │ {

148 │ return Json(new ResponseModel(QLBHConst.Error, "Trùng mã tài khoản đơn vị"));

149 │ }

150 │ if (Code == null ||Code.IsEmpty())

151 │ {

152 │ return Json(new ResponseModel(QLBHConst.Error, "Chưa có Code chi nhánh"));

153 │ }

154 │

155 │ item.Code = Code;

156 │ item.Ten = Ten;

157 │ item.TenVietTat = TenVietTat;

158 │ item.SoTaiKhoan = SoTaiKhoan;

159 │ if (NgayKhoaSo != null)

160 │ {

161 │ DateTime NgayKhoaSoDate = NgayKhoaSo.ToDateTime2(DateTime.Now)!.Value;

162 │ item.NgayKhoaSo = NgayKhoaSoDate;

163 │ }

164 │

165 │ item.PhapNhan = PhapNhan;

166 │ item.DiaChi = DiaChi;

167 │ item.MaSoThue = MaSoThue;

168 │ item.Status = TrangThai == "1";

169 │ dbcontext.SysBranch.Update(item);

170 │ dbcontext.SaveChanges();

171 │ return Json(QLBHConst.ResponseUpdateSuccessfully);

172 │ }

173 │

174 │ [HttpPost]

175 │ [TokenUserCheckHTTP(Permission.Delete)]

176 │ public ActionResult DeleteAPI(string Ma)

177 │ {

178 │ var item = dbcontext.SysBranch.FirstOrDefault(x => x.Ma == Ma.GetGuid());

179 │ if (item == null)

180 │ {

181 │ return Json(QLBHConst.ResponseNotExist);

182 │ }

183 │ item.Deleted = true;

184 │ dbcontext.SaveChanges();

185 │ return Json(new ResponseModel(QLBHConst.OK, "Đã xóa thành công"));

186 │ }

187 │

188 │

189 │ [HttpPost]

190 │ [TokenUserCheckHTTP(Permission.NotCheck)]

191 │ public ActionResult getBranchByUserMenu(string UserID, string MenuID)

192 │ {

193 │ //"C55F1D52-43F3-4256-AFF9-302991D7AEE2"

194 │ //var listRolePermission = (from rp in dbcontext.SysRolePermission.Where(t=>(t.Status ?? false) && !(t.Delet...

195 │ // join rm in dbcontext.SysRoleMenu.Where(t => (t.Status ?? false) && !(t.Deleted ?...

196 │ // join ur in dbcontext.SysUserRole.Where(t => (t.Status ?? false) && !(t.Deleted ?...

197 │ // join cn in dbcontext.SysBranch.Where(t => (t.Status ?? false) && !(t.Deleted ?? ...

198 │ // where ur.UserId == UserID.GetGuid() && rm.MenuId == MenuID.GetGuid()

199 │ // select new

200 │ // {

201 │ // ma = cn.Ma +"",

202 │ // code = cn.Code + "",

203 │ // ten = cn.Ten,

204 │ // }).ToList();

205 │

206 │ var listChiNhanh = (from rm in dbcontext.SysRoleMenu.Where(t => (t.Status ?? false) && !(t.Deleted ?? false)...

207 │ join ur in dbcontext.SysUserRole.Where(t => (t.Status ?? false) && !(t.Deleted ?? false)...

208 │ join cn in dbcontext.SysBranch.Where(t => (t.Status ?? false) && !(t.Deleted ?? false)) ...

209 │ where ur.UserId == UserID.GetGuid() && rm.MenuId == MenuID.GetGuid()

210 │ orderby cn.Ten

211 │ select new

212 │ {

213 │ ma = cn.Ma + "",

214 │ code = cn.Code + "",

215 │ ten = cn.Ten,

216 │ trangthai = cn.Status,

217 │ }).Distinct().ToList();

218 │

219 │ //var listData = listRolePermissionView;

220 │ return Json(new ResponseModel(QLBHConst.OK, "", listChiNhanh));

221 │ }

222 │

223 │

224 │

225 │ }

226 │ }

────────────────────────────────────────────────────────────

**📄 CRC16.cs (CS, 69 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │ using System.Linq;

4 │ using System.Text;

5 │ using System.Threading.Tasks;

6 │ using static System.Net.Mime.MediaTypeNames;

7 │

8 │ namespace QuanLyBanHang.Common.BIDV

9 │ {

10 │ public class CRC16

11 │ {

12 │ private static int[] TABLE = {

13 │ 0x0000, 0xC0C1, 0xC181, 0x0140, 0xC301, 0x03C0, 0x0280, 0xC241,

14 │ 0xC601, 0x06C0, 0x0780, 0xC741, 0x0500, 0xC5C1, 0xC481, 0x0440,

15 │ 0xCC01, 0x0CC0, 0x0D80, 0xCD41, 0x0F00, 0xCFC1, 0xCE81, 0x0E40,

16 │ 0x0A00, 0xCAC1, 0xCB81, 0x0B40, 0xC901, 0x09C0, 0x0880, 0xC841,

17 │ 0xD801, 0x18C0, 0x1980, 0xD941, 0x1B00, 0xDBC1, 0xDA81, 0x1A40,

18 │ 0x1E00, 0xDEC1, 0xDF81, 0x1F40, 0xDD01, 0x1DC0, 0x1C80, 0xDC41,

19 │ 0x1400, 0xD4C1, 0xD581, 0x1540, 0xD701, 0x17C0, 0x1680, 0xD641,

20 │ 0xD201, 0x12C0, 0x1380, 0xD341, 0x1100, 0xD1C1, 0xD081, 0x1040,

21 │ 0xF001, 0x30C0, 0x3180, 0xF141, 0x3300, 0xF3C1, 0xF281, 0x3240,

22 │ 0x3600, 0xF6C1, 0xF781, 0x3740, 0xF501, 0x35C0, 0x3480, 0xF441,

23 │ 0x3C00, 0xFCC1, 0xFD81, 0x3D40, 0xFF01, 0x3FC0, 0x3E80, 0xFE41,

24 │ 0xFA01, 0x3AC0, 0x3B80, 0xFB41, 0x3900, 0xF9C1, 0xF881, 0x3840,

25 │ 0x2800, 0xE8C1, 0xE981, 0x2940, 0xEB01, 0x2BC0, 0x2A80, 0xEA41,

26 │ 0xEE01, 0x2EC0, 0x2F80, 0xEF41, 0x2D00, 0xEDC1, 0xEC81, 0x2C40,

27 │ 0xE401, 0x24C0, 0x2580, 0xE541, 0x2700, 0xE7C1, 0xE681, 0x2640,

28 │ 0x2200, 0xE2C1, 0xE381, 0x2340, 0xE101, 0x21C0, 0x2080, 0xE041,

29 │ 0xA001, 0x60C0, 0x6180, 0xA141, 0x6300, 0xA3C1, 0xA281, 0x6240,

30 │ 0x6600, 0xA6C1, 0xA781, 0x6740, 0xA501, 0x65C0, 0x6480, 0xA441,

31 │ 0x6C00, 0xACC1, 0xAD81, 0x6D40, 0xAF01, 0x6FC0, 0x6E80, 0xAE41,

32 │ 0xAA01, 0x6AC0, 0x6B80, 0xAB41, 0x6900, 0xA9C1, 0xA881, 0x6840,

33 │ 0x7800, 0xB8C1, 0xB981, 0x7940, 0xBB01, 0x7BC0, 0x7A80, 0xBA41,

34 │ 0xBE01, 0x7EC0, 0x7F80, 0xBF41, 0x7D00, 0xBDC1, 0xBC81, 0x7C40,

35 │ 0xB401, 0x74C0, 0x7580, 0xB541, 0x7700, 0xB7C1, 0xB681, 0x7640,

36 │ 0x7200, 0xB2C1, 0xB381, 0x7340, 0xB101, 0x71C0, 0x7080, 0xB041,

37 │ 0x5000, 0x90C1, 0x9181, 0x5140, 0x9301, 0x53C0, 0x5280, 0x9241,

38 │ 0x9601, 0x56C0, 0x5780, 0x9741, 0x5500, 0x95C1, 0x9481, 0x5440,

39 │ 0x9C01, 0x5CC0, 0x5D80, 0x9D41, 0x5F00, 0x9FC1, 0x9E81, 0x5E40,

40 │ 0x5A00, 0x9AC1, 0x9B81, 0x5B40, 0x9901, 0x59C0, 0x5880, 0x9841,

41 │ 0x8801, 0x48C0, 0x4980, 0x8941, 0x4B00, 0x8BC1, 0x8A81, 0x4A40,

42 │ 0x4E00, 0x8EC1, 0x8F81, 0x4F40, 0x8D01, 0x4DC0, 0x4C80, 0x8C41,

43 │ 0x4400, 0x84C1, 0x8581, 0x4540, 0x8701, 0x47C0, 0x4680, 0x8641,

44 │ 0x8201, 0x42C0, 0x4380, 0x8341, 0x4100, 0x81C1, 0x8081, 0x4040,};

45 │ public static string caculate(string message)

46 │ {

47 │ int crc = 0xFFFF; // initial value

48 │ int polynomial = 0x1021; // 0001 0000 0010 0001 (0, 5, 12)

49 │

50 │ byte[] bytes = Encoding.UTF8.GetBytes(message);

51 │ foreach (byte b in bytes)

52 │ {

53 │ for (int i = 0; i < 8; i++)

54 │ {

55 │ bool bit = ((b >> (7 - i) & 1) == 1);

56 │ bool c15 = ((crc >> 15 & 1) == 1);

57 │ crc <<= 1;

58 │ if (c15 ^ bit)

59 │ {

60 │ crc ^= polynomial;

61 │ }

62 │ }

63 │ }

64 │ crc &= 0xffff;

65 │ string CRC16 = crc.ToString("X").ToUpper();// Integer.toHexString(crc).toUpperCase();

66 │ return QRPack.LeftPad(CRC16, 4, "0");

67 │ }

68 │ }

69 │ }

────────────────────────────────────────────────────────────

**📄 Cast.cs (CS, 295 lines)**

1 │ ﻿using System.Globalization;

2 │ using System.Reflection;

3 │

4 │ namespace QuanLyBanHang.Common

5 │ {

6 │ public static class Cast

7 │ {

8 │ public static string ToString(object? o)

9 │ {

10 │ return (o ?? "").ToString() ?? "";

11 │ }

12 │

13 │ public static byte ToByte(object? o)

14 │ {

15 │ byte result = 0;

16 │ if (!Utility.IsNull(o))

17 │ {

18 │ byte.TryParse(o.ToString(), out result);

19 │ }

20 │ return result;

21 │ }

22 │

23 │ public static int ToInt(object? o)

24 │ {

25 │ return (int)ToDouble(o);

26 │ }

27 │

28 │ public static float ToFloat(object? o)

29 │ {

30 │ float result = 0f;

31 │ if (!Utility.IsNull(o))

32 │ {

33 │ float.TryParse(o.ToString(), out result);

34 │ }

35 │ return result;

36 │ }

37 │

38 │ public static double ToDouble(object? o)

39 │ {

40 │ double result = 0.0;

41 │ if (!Utility.IsNull(o))

42 │ {

43 │ double.TryParse(o.ToString(), out result);

44 │ }

45 │ return result;

46 │ }

47 │

48 │ public static decimal ToDecimal(object? o)

49 │ {

50 │ decimal result = 0m;

51 │ if (!Utility.IsNull(o))

52 │ {

53 │ decimal.TryParse(o.ToString(), out result);

54 │ }

55 │ return result;

56 │ }

57 │

58 │ public static bool ToBoolean(object? o)

59 │ {

60 │ bool result = false;

61 │ if (!Utility.IsNull(o))

62 │ {

63 │ bool.TryParse(o.ToString(), out result);

64 │ }

65 │ return result;

66 │ }

67 │

68 │ public static DateTime ToDateTime(object? o)

69 │ {

70 │ DateTime result = new DateTime(2000, 1, 1);

71 │ if (!Utility.IsNull(o))

72 │ {

73 │ DateTime.TryParse(o.ToString(), out result);

74 │ }

75 │ return result;

76 │ }

77 │

78 │ public static DateTime ToDateTime(string dateTime, string Format)

79 │ {

80 │ DateTime result = new DateTime(2000, 1, 1);

81 │ try

82 │ {

83 │ if (!Utility.IsNull(dateTime))

84 │ {

85 │ return DateTime.ParseExact(dateTime, Format, new CultureInfo("en-US"), DateTimeStyles.None);

86 │ }

87 │ }

88 │ catch

89 │ {

90 │ }

91 │ return result;

92 │ }

93 │

94 │ public static TimeSpan ToTimeSpan(object o)

95 │ {

96 │ TimeSpan result = new TimeSpan(0L);

97 │ if (!Utility.IsNull(o))

98 │ {

99 │ TimeSpan.TryParse(o.ToString(), out result);

100 │ }

101 │ return result;

102 │ }

103 │

104 │ public static object Nz(object o, object oReplace)

105 │ {

106 │ object result;

107 │ if (!Utility.IsNull(o))

108 │ {

109 │ result = o;

110 │ }

111 │ else

112 │ {

113 │ result = oReplace;

114 │ }

115 │ return result;

116 │ }

117 │

118 │ public static object TryConvert(PropertyInfo pro, object o)

119 │ {

120 │ string text = pro.PropertyType.ToString();

121 │ object result;

122 │ if (text.Contains("Decimal"))

123 │ {

124 │ result = ToDecimal(o);

125 │ }

126 │ else if (text.Contains("Double"))

127 │ {

128 │ result = ToDouble(o);

129 │ }

130 │ else if (text.Contains("Int32"))

131 │ {

132 │ result = ToInt(o);

133 │ }

134 │ else if (text.Contains("Boolean"))

135 │ {

136 │ result = ToBoolean(o);

137 │ }

138 │ else if (text.Contains("DateTime"))

139 │ {

140 │ result = ToDateTime(o);

141 │ }

142 │ else if (text.Contains("String"))

143 │ {

144 │ result = Nz(o, "").ToString() ?? "";

145 │ }

146 │ else

147 │ {

148 │ result = o;

149 │ }

150 │ return result;

151 │ }

152 │

153 │ public static short ToShort(object o)

154 │ {

155 │ return (short)ToDouble(o);

156 │ }

157 │ }

158 │

159 │ public static class CastExtend

160 │ {

161 │ public static string ToString(this object o)

162 │ {

163 │ return Cast.ToString(o);

164 │ }

165 │

166 │ public static byte ToByte(this object o)

167 │ {

168 │ return Cast.ToByte(o);

169 │ }

170 │

171 │ public static int ToInt(this object o)

172 │ {

173 │ return Cast.ToInt(o);

174 │ }

175 │

176 │ public static float ToFloat(this object o)

177 │ {

178 │ return Cast.ToFloat(o);

179 │ }

180 │

181 │ public static double ToDouble(this object o)

182 │ {

183 │ return Cast.ToDouble(o);

184 │ }

185 │

186 │ public static decimal ToDecimal(this object o)

187 │ {

188 │ return Cast.ToDecimal(o);

189 │ }

190 │

191 │ public static bool ToBoolean(this object o)

192 │ {

193 │ return Cast.ToBoolean(o);

194 │ }

195 │

196 │ public static DateTime ToDateTime(this object o)

197 │ {

198 │ return Cast.ToDateTime(o);

199 │ }

200 │ public static DateTime? ToDateTime2(this object o, DateTime? df = null)

201 │ {

202 │ string tmp = (o + "");

203 │ if (tmp.IsEmpty())

204 │ {

205 │ return df;

206 │ }

207 │ try

208 │ {

209 │ DateTime x;

210 │ if (tmp.Contains("T") || tmp.Contains(" "))//Datetime

211 │ {

212 │ if (DateTime.TryParseExact(tmp, tmp.Contains("-") ? "yyyy-MM-ddTHH:mm:ss" : "dd/MM/yyyy HH:mm:ss", n...

213 │ {

214 │ return x;

215 │ }

216 │ if (DateTime.TryParseExact(tmp, tmp.Contains("-") ? "yyyy-MM-ddThh:mm:ss" : "dd/MM/yyyy hh:mm:ss", n...

217 │ {

218 │ return x;

219 │ }

220 │ if (DateTime.TryParseExact(tmp, tmp.Contains("-") ? "yyyy-MM-ddTHH:mm" : "dd/MM/yyyyTHH:mm", null, D...

221 │ {

222 │ return x;

223 │ }

224 │ if (DateTime.TryParseExact(tmp, tmp.Contains("-") ? "yyyy-MM-dd HH:mm" : "dd/MM/yyyy HH:mm", null, D...

225 │ {

226 │ return x;

227 │ }

228 │ if (DateTime.TryParseExact(tmp, tmp.Contains("-") ? "yyyy-MM-dd hh:mm:ss tt" : "dd/MM/yyyy hh:mm:ss ...

229 │ {

230 │ return x;

231 │ }

232 │ return df;

233 │ }

234 │ else//Date

235 │ {

236 │ if (tmp.Contains("-") || tmp.Contains("/"))

237 │ {

238 │ if (DateTime.TryParseExact(tmp, tmp.Contains("-") ? "yyyy-MM-dd" : "dd/MM/yyyy", null, DateTimeS...

239 │ {

240 │ return x;

241 │ }

242 │ return df;

243 │ }

244 │ else

245 │ {

246 │ if (Cast.ToInt(tmp) > 100000)//yyyyMd hoặc dMyyyy hoặc yyyyMMdd hoặc ddMMyyyy

247 │ {

248 │ if (tmp.Length == 8) //yyyyMMdd hoặc ddMMyyyy

249 │ {

250 │ //coming soon

251 │ }

252 │ }

253 │ else if (Cast.ToInt(tmp) > 1900)//yyyy

254 │ {

255 │ return new DateTime(Cast.ToInt(o + ""), 1, 1);

256 │ }

257 │ }

258 │ }

259 │ }

260 │ catch { }

261 │ return df;

262 │ }

263 │

264 │ public static bool IsToday(this DateTime? dateTime)

265 │ {

266 │ var now = DateTime.Now;

267 │ return dateTime?.Day == now.Day && dateTime?.Month == now.Month && dateTime?.Year == now.Year;

268 │ }

269 │

270 │ public static DateTime ToDateTime(this string dateTime, string Format)

271 │ {

272 │ return Cast.ToDateTime(dateTime, Format);

273 │ }

274 │

275 │ public static TimeSpan ToTimeSpan(this object o)

276 │ {

277 │ return Cast.ToTimeSpan(o);

278 │ }

279 │

280 │ public static object Nz(this object o, object oReplace)

281 │ {

282 │ return Cast.Nz(o, oReplace);

283 │ }

284 │

285 │ public static object TryConvert(this PropertyInfo pro, object o)

286 │ {

287 │ return Cast.TryConvert(pro, o);

288 │ }

289 │

290 │ public static short ToShort(object o)

291 │ {

292 │ return Cast.ToShort(o);

293 │ }

294 │ }

295 │ }

────────────────────────────────────────────────────────────

**📄 CatLoaiDichVu.cs (CS, 33 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class CatLoaiDichVu

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string? Code { get; set; }

11 │

12 │ public string? Ten { get; set; }

13 │

14 │ public string? MoTa { get; set; }

15 │

16 │ public bool? Status { get; set; }

17 │

18 │ public Guid? UserCreated { get; set; }

19 │

20 │ public DateTime? CreatedDate { get; set; }

21 │

22 │ public Guid? UserModified { get; set; }

23 │

24 │ public DateTime? ModifiedDate { get; set; }

25 │

26 │ public bool? Deleted { get; set; }

27 │

28 │ public Guid? UserDeleted { get; set; }

29 │

30 │ public DateTime? DeletedDate { get; set; }

31 │

32 │ public Guid? MaChiNhanh { get; set; }

33 │ }

────────────────────────────────────────────────────────────

**📄 CatNhomDichVu.cs (CS, 33 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class CatNhomDichVu

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int? MaDanhMuc { get; set; }

11 │

12 │ public string? Code { get; set; }

13 │

14 │ public string? Ten { get; set; }

15 │

16 │ public bool? Status { get; set; }

17 │

18 │ public Guid? UserCreated { get; set; }

19 │

20 │ public DateTime? CreatedDate { get; set; }

21 │

22 │ public Guid? UserModified { get; set; }

23 │

24 │ public DateTime? ModifiedDate { get; set; }

25 │

26 │ public bool? Deleted { get; set; }

27 │

28 │ public Guid? UserDeleted { get; set; }

29 │

30 │ public DateTime? DeletedDate { get; set; }

31 │

32 │ public Guid? MaChiNhanh { get; set; }

33 │ }

────────────────────────────────────────────────────────────

**📄 ChatHUB.cs (CS, 31 lines)**

1 │ ﻿using System.Collections.Concurrent;

2 │ using Microsoft.AspNetCore.SignalR;

3 │

4 │ namespace QuanLyBanHang.Common

5 │ {

6 │ public class ChatHUB : Hub

7 │ {

8 │ private static readonly ConcurrentDictionary<string, string> \_connections = new();

9 │

10 │ public override Task OnConnectedAsync()

11 │ {

12 │ \_connections.TryAdd(Context.ConnectionId, Context.UserIdentifier ?? "anonymous");

13 │ return base.OnConnectedAsync();

14 │ }

15 │

16 │ public override Task OnDisconnectedAsync(Exception exception)

17 │ {

18 │ \_connections.TryRemove(Context.ConnectionId, out \_);

19 │ return base.OnDisconnectedAsync(exception);

20 │ }

21 │

22 │ public static int ConnectionCount => \_connections.Count;

23 │ public async Task JoinGroup(string uuid = "")

24 │ {

25 │ var groupName = uuid == "" ? "OrderGroup" : uuid;

26 │

27 │ await Groups.AddToGroupAsync(Context.ConnectionId, groupName);

28 │

29 │ }

30 │ }

31 │ }

────────────────────────────────────────────────────────────

**📄 ChiTietDinhMucDVKTModel.cs (CS, 141 lines)**

1 │ ﻿using Microsoft.EntityFrameworkCore;

2 │ using QuanLyBanHang.Models.QLBHEntity;

3 │

4 │ namespace QuanLyBanHang.Models.ExModel.VTTH

5 │ {

6 │ public class ChiTietDinhMucDVKTModel : Base<VtthChiTietDinhMuc>

7 │ {

8 │ public ChiTietDinhMucDVKTModel(DbContext dbContext) : base(dbContext)

9 │ {

10 │ }

11 │

12 │ QuanLyBanHangContext dbContext

13 │ {

14 │ get { return \_dbContext as QuanLyBanHangContext; }

15 │ set { return; }

16 │ }

17 │

18 │ public VtthChiTietDinhMuc GetItem(int Ma)

19 │ {

20 │ return First(t => t.Ma == Ma && !(t.Deleted ?? false));

21 │ }

22 │

23 │ public List<VtthChiTietDinhMuc> GetData(int MaDinhMuc)

24 │ {

25 │ return Get(t => t.MaDinhMuc == MaDinhMuc && !(t.Deleted ?? false)).ToList();

26 │ }

27 │

28 │ public int SaveListChiTiet(int maDinhMucCha, List<VtthChiTietDinhMuc> ctsFromClient, Guid userid)

29 │ {

30 │ try

31 │ {

32 │ // Lấy danh sách chi tiết hiện có trong DB cho định mức cha này

33 │ var ctsOldInDb = \_dbContext.Set<VtthChiTietDinhMuc>()

34 │ .Where(t => t.MaDinhMuc == maDinhMucCha && !(t.Deleted ?? false))

35 │ .ToList();

36 │

37 │ // --- Tách biệt các mục từ client ---

38 │ // 1. Các mục mới hoàn toàn (có Ma = 0 hoặc Ma không tồn tại trong ctsOldInDb)

39 │ // 2. Các mục đã tồn tại và có thể cần cập nhật (Ma != 0 và Ma có trong ctsOldInDb)

40 │ // 3. Các mục trong DB cần xóa (có trong ctsOldInDb nhưng không có trong danh sách Ma != 0 t�...

41 │

42 │ var newItemsFromClient = new List<VtthChiTietDinhMuc>();

43 │ var existingItemsFromClientMap = new Dictionary<int, VtthChiTietDinhMuc>();

44 │

45 │ foreach (var itemClient in ctsFromClient)

46 │ {

47 │ if (itemClient.Ma == 0) // Đây là mục mới client muốn thêm

48 │ {

49 │ newItemsFromClient.Add(itemClient);

50 │ }

51 │ else // Đây là mục client nghĩ là đã tồn tại (Ma != 0)

52 │ {

53 │ if (!existingItemsFromClientMap.ContainsKey(itemClient.Ma))

54 │ {

55 │ existingItemsFromClientMap.Add(itemClient.Ma, itemClient);

56 │ }

57 │ else

58 │ {

59 │ // Lỗi: Client gửi nhiều mục chi tiết có cùng Ma (khác 0).

60 │ // Điều này không nên xảy ra nếu Ma là khóa chính và client xử lý đúng.

61 │ // Bạn có thể throw exception ở đây hoặc ghi log và bỏ qua mục trùng lặp.

62 │ // Ví dụ: throw new ArgumentException($"Client gửi nhiều chi tiết trùng Mã (Ma={i...

63 │ Console.WriteLine($"Cảnh báo: Client gửi nhiều chi tiết trùng Mã (Ma={itemClient....

64 │ }

65 │ }

66 │ }

67 │

68 │ // --- Xử lý các mục trong Database (ctsOldInDb) ---

69 │ var itemsToDeleteInDb = new List<VtthChiTietDinhMuc>();

70 │ var itemsToUpdateInDb = new List<VtthChiTietDinhMuc>();

71 │

72 │ foreach (var oldItemDb in ctsOldInDb)

73 │ {

74 │ if (existingItemsFromClientMap.TryGetValue(oldItemDb.Ma, out var correspondingClientItem))

75 │ {

76 │ // Mục này có trong DB và cũng có trong danh sách (Ma != 0) từ client -> Cập nhật

77 │ oldItemDb.MaDvt = correspondingClientItem.MaDvt;

78 │ oldItemDb.SoLuong = correspondingClientItem.SoLuong;

79 │ oldItemDb.DonGiaTb = correspondingClientItem.DonGiaTb;

80 │ oldItemDb.ThanhTien = correspondingClientItem.ThanhTien;

81 │ oldItemDb.GhiChu = correspondingClientItem.GhiChu;

82 │ oldItemDb.KhongTruKho = correspondingClientItem.KhongTruKho;

83 │ oldItemDb.UserModified = userid;

84 │ oldItemDb.ModifiedDate = DateTime.Now;

85 │ itemsToUpdateInDb.Add(oldItemDb); // Thêm vào danh sách cần gọi UpdateRange

86 │ }

87 │ else

88 │ {

89 │ // Mục này có trong DB nhưng không có trong danh sách (Ma != 0) từ client -> Xóa

90 │ oldItemDb.Deleted = true;

91 │ oldItemDb.DeletedDate = DateTime.Now;

92 │ oldItemDb.UserDeleted = userid;

93 │ itemsToDeleteInDb.Add(oldItemDb); // Thêm vào danh sách cần gọi UpdateRange

94 │ }

95 │ }

96 │

97 │ // --- Xử lý các mục mới từ Client ---

98 │ foreach (var newItem in newItemsFromClient)

99 │ {

100 │ newItem.MaDinhMuc = maDinhMucCha; // Gán Ma của định mức cha

101 │ newItem.CreatedDate = DateTime.Now;

102 │ newItem.UserCreated = userid;

103 │ newItem.Deleted = false; // Đảm bảo là chưa bị xóa

104 │ // Ma sẽ được tự động tạo bởi DB nếu là cột IDENTITY.

105 │ // Nếu Ma không phải IDENTITY, bạn cần cơ chế tạo Ma mới ��...

106 │ }

107 │

108 │ // --- Áp dụng thay đổi vào DbContext ---

109 │ bool hasChanges = itemsToDeleteInDb.Any() || itemsToUpdateInDb.Any() || newItemsFromClient.Any();

110 │

111 │ if (hasChanges)

112 │ {

113 │ if (itemsToDeleteInDb.Any())

114 │ {

115 │ \_dbContext.UpdateRange(itemsToDeleteInDb);

116 │ }

117 │

118 │ if (itemsToUpdateInDb.Any())

119 │ {

120 │ \_dbContext.UpdateRange(itemsToUpdateInDb);

121 │ }

122 │

123 │ if (newItemsFromClient.Any())

124 │ {

125 │ \_dbContext.AddRange(newItemsFromClient);

126 │ }

127 │

128 │ \_dbContext.SaveChanges();

129 │ }

130 │

131 │ return 1; // Thành công

132 │ }

133 │ catch (Exception ex) // Bắt lỗi cụ thể hơn nếu cần

134 │ {

135 │ // Ghi log lỗi ex ở đây để debug

136 │ Console.WriteLine($"Lỗi trong SaveListChiTiet: {ex.ToString()}");

137 │ return 0; // Thất bại

138 │ }

139 │ }

140 │ }

141 │ }

────────────────────────────────────────────────────────────

**📄 ChiTietHoaDonModel.cs (CS, 310 lines)**

1 │ ﻿using Microsoft.EntityFrameworkCore;

2 │ using QuanLyBanHang.Common;

3 │ using QuanLyBanHang.Models.ExModel.StoredProcedureModel;

4 │ using QuanLyBanHang.Models.QLBHEntity;

5 │ using static QuanLyBanHang.Controllers.OPS.BanHangController;

6 │

7 │ namespace QuanLyBanHang.Models.ExModel.OPS

8 │ {

9 │ public class ChiTietHoaDonModel : Base<OpsChiTietHoaDon>

10 │ {

11 │ public ChiTietHoaDonModel(DbContext dbContext) : base(dbContext)

12 │ {

13 │ }

14 │ QuanLyBanHangContext dbContext

15 │ {

16 │ get { return \_dbContext as QuanLyBanHangContext; }

17 │ set { return; }

18 │ }

19 │ public OpsChiTietHoaDon GetItem(int Ma)

20 │ {

21 │ return First(t => t.Ma == Ma && !(t.Deleted ?? false));

22 │ }

23 │ public List<sp\_ChiTietHoaDonResult> GetFullData(int Ma)

24 │ {

25 │ var chiTietHoaDon = (

26 │ from ct in dbContext.OpsChiTietHoaDon.AsNoTracking().Where(t => !(t.Deleted ?? false) && t.MaHoaDon == M...

27 │ join hh in dbContext.TblHangHoa on ct.MaHangHoa equals hh.Ma

28 │

29 │ select new sp\_ChiTietHoaDonResult

30 │ {

31 │ TenHang = hh.Ten,

32 │ SoLuong = ct.SoLuong,

33 │ DonGia = ct.DonGia,

34 │ GiaGiam = ct.GiaGiam,

35 │ ThanhTien = (ct.SoLuong ?? 0) \* (ct.DonGia ?? 0)

36 │ }).ToList();

37 │

38 │ return chiTietHoaDon;

39 │ }

40 │ public List<sp\_ChiTietHoaDonTraResult> GetFullDataTra(int Ma)

41 │ {

42 │ var chiTietTraLai = (

43 │ from ct in dbContext.OpsChiTietHoaDon.AsNoTracking().Where(t => !(t.Deleted ?? false) && t.MaHoaDon == M...

44 │ join hh in dbContext.TblHangHoa on ct.MaHangHoa equals hh.Ma

45 │ select new sp\_ChiTietHoaDonTraResult

46 │ {

47 │ TenHang = hh.Ten,

48 │ SoLuongTraLai = ct.SoLuongTraLai,

49 │ DonGia = ct.DonGia,

50 │ GiaGiam = ct.GiaGiam,

51 │ ThanhTienTraLai = (ct.SoLuongTraLai ?? 0) \* (ct.DonGia ?? 0),

52 │ NgayTra = ct.NgayTra

53 │ }).ToList();

54 │

55 │ return chiTietTraLai;

56 │ }

57 │

58 │ public List<OpsChiTietHoaDon> GetData(int HoaDon\_ID)

59 │ {

60 │ return Get(t => t.MaHoaDon == HoaDon\_ID && !(t.Deleted ?? false)).OrderBy(t => t.Ma).ToList();

61 │ }

62 │ public int SaveListChiTiet(int HoaDon\_ID, int MaKho, List<cCT\_HoaDon> cts, Guid UserID, DateTime dtNgayTao, int ...

63 │ {

64 │ message = "";

65 │ TriGia = 0;

66 │ int row = 0;

67 │

68 │ try

69 │ {

70 │ var hd = dbContext.OpsHoaDon.FirstOrDefault(t => t.Ma == HoaDon\_ID);

71 │ if (hd == null)

72 │ {

73 │ message = "Hóa đơn không tồn tại.";

74 │ return -999;

75 │ }

76 │

77 │ // Xoá các chi tiết không còn trong danh sách mới

78 │ var ct\_olds = dbContext.OpsChiTietHoaDon.Where(t => t.MaHoaDon == HoaDon\_ID).ToList();

79 │ foreach (var ct\_old in ct\_olds)

80 │ {

81 │ if (!cts.Any(t => Cast.ToInt(t.Ma) == ct\_old.Ma))

82 │ {

83 │ ct\_old.Deleted = true;

84 │ ct\_old.UserDeleted = UserID;

85 │ ct\_old.DeletedDate = dtNgayTao;

86 │ }

87 │ }

88 │

89 │ row = 1;

90 │ foreach (var ct in cts)

91 │ {

92 │ int ma\_ct = Cast.ToInt(ct.Ma);

93 │ int maHangHoa = Cast.ToInt(ct.MaHangHoa);

94 │ int soLuongYeuCau = Cast.ToInt(ct.SoLuong);

95 │

96 │ // Kiểm tra hàng hóa gốc

97 │ var hangHoa = dbContext.TblHangHoa

98 │ .FirstOrDefault(h => h.Ma == maHangHoa && !(h.Deleted ?? false));

99 │

100 │ if (hangHoa == null)

101 │ {

102 │ message = $"Hàng hóa không tồn tại. Dòng {row}";

103 │ return -999;

104 │ }

105 │

106 │ // Lấy thông tin giá và kiểm tra ChoPhepBanAm tại VGiaBanTheoKho

107 │ var hhGia = dbContext.VGiaBanTheoKho

108 │ .FirstOrDefault(t => t.MaHangHoa == maHangHoa && t.MaKho == MaKho);

109 │

110 │ if (hhGia == null)

111 │ {

112 │ message = $"Không tìm thấy thông tin giá bán cho hàng hóa. Dòng {row}";

113 │ return -999;

114 │ }

115 │

116 │ bool choPhepBanAm = hhGia.ChoPhepBanAm ?? false;

117 │

118 │ // Kiểm tra hàng cha

119 │ var danhSachCon = dbContext.TblHangHoa

120 │ .Where(h => h.ParentId == maHangHoa && !(h.Deleted ?? false))

121 │ .ToList();

122 │

123 │ bool laHangCha = danhSachCon.Any();

124 │

125 │ if (laHangCha)

126 │ {

127 │ int soLuongConLai = soLuongYeuCau;

128 │

129 │ foreach (var hangCon in danhSachCon)

130 │ {

131 │ var tonkho = dbContext.StockTonKhoBanHang

132 │ .FirstOrDefault(t => t.MaKho == MaKho &&

133 │ t.NamThang == hd.NgayHoaDon.ToString("yyyyMM") &&

134 │ t.MaHangHoa == hangCon.Ma);

135 │

136 │ int slTon = (int?)tonkho?.SltonCuoiKy ?? 0;

137 │

138 │ if (!choPhepBanAm && slTon <= 0)

139 │ continue;

140 │

141 │ int slTru = choPhepBanAm ? soLuongConLai : Math.Min(slTon, soLuongConLai);

142 │

143 │ if (slTru > 0)

144 │ {

145 │ dbContext.OpsChiTietHoaDon.Add(new OpsChiTietHoaDon

146 │ {

147 │ MaChiNhanhInt = MaChiNhanhInt,

148 │ MaHoaDon = HoaDon\_ID,

149 │ MaHangHoa = hangCon.Ma,

150 │ MaVach = ct.MaVach,

151 │ MaDvt = hangCon.MaDvt ?? 0,

152 │ SoLuong = slTru,

153 │ DonGia = Cast.ToDecimal(ct.DonGia),

154 │ GiaGiam = Cast.ToDecimal(ct.GiaGiam),

155 │ ThanhTien = Cast.ToDecimal(ct.ThanhTien) \* slTru / soLuongYeuCau,

156 │ UserCreated = UserID,

157 │ CreatedDate = dtNgayTao

158 │ });

159 │

160 │ TriGia += Cast.ToDecimal(ct.ThanhTien) \* slTru / soLuongYeuCau;

161 │ soLuongConLai -= slTru;

162 │

163 │ if (soLuongConLai <= 0)

164 │ break;

165 │ }

166 │ }

167 │

168 │ if (soLuongConLai > 0)

169 │ {

170 │ message = $"Không đủ tồn kho cho hàng hóa: {hangHoa.Ten} (mã: {maHangHoa})";

171 │ return -999;

172 │ }

173 │ }

174 │ else

175 │ {

176 │ // Hàng đơn, không có hàng con

177 │ var tonkho = dbContext.StockTonKhoBanHang

178 │ .FirstOrDefault(t => t.MaKho == MaKho &&

179 │ t.NamThang == hd.NgayHoaDon.ToString("yyyyMM") &&

180 │ t.MaHangHoa == maHangHoa);

181 │

182 │ int tonCuoiKy = (int?)tonkho?.SltonCuoiKy ?? 0;

183 │

184 │ if (!choPhepBanAm && tonCuoiKy < soLuongYeuCau)

185 │ {

186 │ message = $"Không đủ tồn kho cho hàng hóa: {hangHoa.Ten}. Dòng {row}";

187 │ return -999;

188 │ }

189 │

190 │ if (ma\_ct == 0)

191 │ {

192 │ dbContext.OpsChiTietHoaDon.Add(new OpsChiTietHoaDon

193 │ {

194 │ MaChiNhanhInt = MaChiNhanhInt,

195 │ MaHoaDon = HoaDon\_ID,

196 │ MaHangHoa = maHangHoa,

197 │ MaVach = ct.MaVach,

198 │ MaDvt = hhGia.MaDvt ?? 0,

199 │ SoLuong = soLuongYeuCau,

200 │ DonGia = Cast.ToDecimal(ct.DonGia),

201 │ GiaGiam = Cast.ToDecimal(ct.GiaGiam),

202 │ ThanhTien = Cast.ToDecimal(ct.ThanhTien),

203 │ UserCreated = UserID,

204 │ CreatedDate = dtNgayTao

205 │ });

206 │ }

207 │ else

208 │ {

209 │ var ct\_old = dbContext.OpsChiTietHoaDon.FirstOrDefault(t => t.Ma == ma\_ct && !(t.Deleted ?? ...

210 │ if (ct\_old == null)

211 │ {

212 │ message = $"Chi tiết không tồn tại. Dòng {row}";

213 │ return -999;

214 │ }

215 │

216 │ if (!choPhepBanAm && (tonCuoiKy + ct\_old.SoLuong < soLuongYeuCau))

217 │ {

218 │ message = $"Không đủ tồn kho (sửa) cho hàng hóa: {hangHoa.Ten}. Dòng {row}";

219 │ return -999;

220 │ }

221 │

222 │ ct\_old.SoLuong = soLuongYeuCau;

223 │ ct\_old.DonGia = Cast.ToDecimal(ct.DonGia);

224 │ ct\_old.GiaGiam = Cast.ToDecimal(ct.GiaGiam);

225 │ ct\_old.ThanhTien = Cast.ToDecimal(ct.ThanhTien);

226 │ ct\_old.ModifiedDate = dtNgayTao;

227 │ ct\_old.UserModified = UserID;

228 │ }

229 │

230 │ TriGia += Cast.ToDecimal(ct.ThanhTien);

231 │ }

232 │

233 │ dbContext.SaveChanges();

234 │ row++;

235 │ }

236 │

237 │

238 │ return 0;

239 │ }

240 │ catch (Exception ex)

241 │ {

242 │ message = $"Lỗi hệ thống: {ex.Message}";

243 │ return -999;

244 │ }

245 │ }

246 │

247 │ public int SaveTraHang(int HoaDon\_ID, List<cCT\_HoaDon> cts, Guid UserID, Guid MaChiNhanh, DateTime dtNgayTao, ou...

248 │ {

249 │ message = "";

250 │ ErrorCode = 0;

251 │ TongSoTienTraLai = 0;

252 │

253 │

254 │ int row = 0;

255 │ int ma\_ct = 0;

256 │ decimal thanhtien = 0;

257 │ try

258 │ {

259 │ row = 1;

260 │ ErrorCode = -2;

261 │ foreach (cCT\_HoaDon ct in cts)

262 │ {

263 │

264 │ if (int.TryParse(ct.Ma, out int directMaValue))

265 │ {

266 │ // Nếu là số nguyên, sử dụng trực tiếp

267 │ ma\_ct = directMaValue;

268 │ }

269 │ else

270 │ {

271 │ // Nếu không phải số nguyên, thử giải mã

272 │ try

273 │ {

274 │ ma\_ct = Cast.ToInt((ct.Ma).Decrypt());

275 │ }

276 │ catch

277 │ {

278 │ message = "Mã chi tiết không hợp lệ. Dòng " + row;

279 │ return -999;

280 │ }

281 │ }

282 │

283 │ //ma\_ct = Cast.ToInt(ct.Ma.Decrypt());

284 │ OpsChiTietHoaDon? ct\_old = dbContext.OpsChiTietHoaDon.Where(t => t.Ma == ma\_ct && !(t.Deleted ?? fal...

285 │ if (ct\_old == null)

286 │ {

287 │ return -row \* 1000;

288 │ }

289 │ ct\_old.SoLuongTraLai = Cast.ToInt(ct.SoLuongTraLai);

290 │ thanhtien = ct\_old.DonGia ?? 0 \* Cast.ToInt(ct.SoLuongTraLai);

291 │ thanhtien = thanhtien - thanhtien \* ((ct\_old.GiaGiam ?? 0) / 100);

292 │ ct\_old.ThanhTienTraLai = Cast.ToInt(thanhtien);

293 │ ct\_old.NguoiTra = UserID;

294 │ ct\_old.NgayTra = dtNgayTao;

295 │ dbContext.SaveChanges();

296 │ TongSoTienTraLai += Cast.ToInt(thanhtien);

297 │ row++;

298 │ }

299 │ ErrorCode = 0;

300 │ return 0;

301 │ }

302 │ catch (Exception ex)

303 │ {

304 │ message = ex.Message;

305 │ return row;

306 │ }

307 │

308 │ }

309 │ }

310 │ }

────────────────────────────────────────────────────────────

**📄 ChiTietKiemKeModel.cs (CS, 207 lines)**

1 │ ﻿using System.Data.Common;

2 │ using Microsoft.EntityFrameworkCore;

3 │ using QuanLyBanHang.Common;

4 │ using QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ namespace QuanLyBanHang.Models.ExModel.OPS

7 │ {

8 │ public class ChiTietKiemKeModel : Base<StockChiTietKiemKe>

9 │ {

10 │ public ChiTietKiemKeModel(DbContext dbContext) : base(dbContext)

11 │ {

12 │ }

13 │ QuanLyBanHangContext dbContext

14 │ {

15 │ get { return \_dbContext as QuanLyBanHangContext; }

16 │ set { return; }

17 │ }

18 │

19 │ public StockChiTietKiemKe GetItem(int Ma)

20 │ {

21 │ return First(t => t.Ma == Ma);

22 │ }

23 │

24 │ public List<StockChiTietKiemKe> GetData(int MaKiemKe)

25 │ {

26 │ return Get(t => t.MaKiemKe == MaKiemKe).ToList();

27 │ }

28 │

29 │ public List<VChiTietKiemKe> GetListChiTietKiemKe(int MaKiemKe, Guid MaChiNhanh)

30 │ {

31 │ return dbContext.VChiTietKiemKe.Where(t => t.MaKiemKe == MaKiemKe && t.MaChiNhanh == MaChiNhanh).ToList();

32 │ }

33 │

34 │ public List<VChiTietKiemKe> GetListChiTietKiemKe(DateTime TuNgay, DateTime DenNgay, int MaKho, Guid MaChiNhanh)

35 │ {

36 │ return dbContext.VChiTietKiemKe.Where(t => t.NgayKiemKe.Value.Date >= TuNgay && t.NgayKiemKe.Value.Date<=Den...

37 │ }

38 │

39 │ public int SaveListChiTiet(int MaKiemKe, List<StockChiTietKiemKe> cts, Guid UserID, DateTime dtNgayTao, bool Del...

40 │ {

41 │ int row = 0;

42 │ int ma\_ct = 0;

43 │

44 │ try

45 │ {

46 │ if (DeleteOld)

47 │ {

48 │ // Lấy danh sách chi tiết kiểm kê cũ chưa bị xóa

49 │ var cts\_old = dbContext.StockChiTietKiemKe

50 │ .Where(t => t.MaKiemKe == MaKiemKe && !(t.Deleted ?? false))

51 │ .ToList();

52 │

53 │ foreach (var ct\_old in cts\_old)

54 │ {

55 │ // Tìm chi tiết kiểm kê trong danh sách mới

56 │ var ct = cts.FirstOrDefault(t => Cast.ToInt(t.Ma.Decrypt()) == ct\_old.Ma);

57 │

58 │ // Nếu không tìm thấy, đánh dấu đã xóa

59 │ if (ct == null)

60 │ {

61 │ ct\_old.Deleted = true;

62 │ ct\_old.UserDeleted = UserID;

63 │ ct\_old.DeletedDate = dtNgayTao;

64 │ dbContext.SaveChanges();

65 │ }

66 │ }

67 │ }

68 │ row = 1;

69 │ var kk = dbContext.StockKiemKe.FirstOrDefault(x => x.Ma == MaKiemKe);

70 │ var br = dbContext.SysBranch.FirstOrDefault(x => x.Ma == kk.MaChiNhanh);

71 │ foreach (var ct in cts)

72 │ {

73 │ ma\_ct = Cast.ToInt(ct.Ma.Decrypt());

74 │

75 │ if (ma\_ct == 0)

76 │ {

77 │ // Thêm mới chi tiết kiểm kê

78 │ var ct\_new = new StockChiTietKiemKe

79 │ {

80 │ MaKiemKe = MaKiemKe,

81 │ MaChiNhanhInt = br.Nid ?? 0,

82 │ MaHangHoa = Cast.ToInt(ct.MaHangHoa),

83 │ SoLuong = ct.SoLuong,

84 │ UserCreated = UserID,

85 │ CreatedDate = dtNgayTao,

86 │ };

87 │

88 │ dbContext.StockChiTietKiemKe.Add(ct\_new);

89 │ }

90 │ else

91 │ {

92 │ // Cập nhật chi tiết kiểm kê hiện có

93 │ var ct\_old = dbContext.StockChiTietKiemKe

94 │ .FirstOrDefault(t => t.Ma == ma\_ct && (t.Deleted ?? false) == false);

95 │

96 │ if (ct\_old == null)

97 │ return row;

98 │ ct\_old.MaChiNhanhInt = br.Nid ?? 0;

99 │ ct\_old.MaHangHoa = Cast.ToInt(ct.MaHangHoa);

100 │ ct\_old.SoLuong = ct.SoLuong;

101 │ ct\_old.UserModified = UserID;

102 │ ct\_old.ModifiedDate = dtNgayTao;

103 │ }

104 │

105 │ dbContext.SaveChanges();

106 │ row++;

107 │ }

108 │

109 │ return 0;

110 │ }

111 │ catch

112 │ {

113 │ return row;

114 │ }

115 │ }

116 │ public int Add(int MaKiemKe, int MaHangHoa, int SoLuong, Guid UserID)

117 │ {

118 │ try

119 │ {

120 │ // Tìm kiếm chi tiết kiểm kê hiện có

121 │ var drItem = dbContext.StockChiTietKiemKe

122 │ .FirstOrDefault(t => t.MaKiemKe == MaKiemKe &&

123 │ t.MaHangHoa == MaHangHoa &&

124 │ !(t.Deleted ?? false));

125 │

126 │ if (drItem != null)

127 │ {

128 │ // Cập nhật chi tiết kiểm kê hiện có

129 │ drItem.SoLuong = (drItem.SoLuong ?? 0) + SoLuong;

130 │ drItem.UserModified = UserID;

131 │ drItem.ModifiedDate = DateTime.Now;

132 │

133 │ // EF Core tự động theo dõi thay đổi của entity

134 │ \_dbContext.SaveChanges();

135 │ return 1;

136 │ }

137 │ else

138 │ {

139 │ // Tạo chi tiết kiểm kê mới

140 │ drItem = new StockChiTietKiemKe

141 │ {

142 │ MaKiemKe = MaKiemKe,

143 │ MaHangHoa = MaHangHoa,

144 │ SoLuong = SoLuong,

145 │ UserCreated = UserID,

146 │ CreatedDate = DateTime.Now

147 │ };

148 │

149 │ // Thêm entity mới vào DbContext

150 │ dbContext.StockChiTietKiemKe.Add(drItem);

151 │ dbContext.SaveChanges();

152 │ return 1;

153 │ }

154 │ }

155 │ catch (Exception)

156 │ {

157 │ return 0;

158 │ }

159 │ }

160 │ public int Edit(int Ma, int SoLuong, string GhiChu, Guid UserID)

161 │ {

162 │ try

163 │ {

164 │ // Tìm chi tiết kiểm kê theo mã

165 │ var drItem = dbContext.StockChiTietKiemKe

166 │ .FirstOrDefault(t => t.Ma == Ma && !(t.Deleted ?? false));

167 │

168 │ if (drItem != null)

169 │ {

170 │ // Cập nhật thông tin chi tiết kiểm kê

171 │ drItem.SoLuong = SoLuong;

172 │ drItem.UserModified = UserID;

173 │ drItem.ModifiedDate = DateTime.Now;

174 │

175 │ // Lưu thay đổi vào cơ sở dữ liệu

176 │ \_dbContext.SaveChanges();

177 │ return 1;

178 │ }

179 │

180 │ return -2; // Không tìm thấy chi tiết kiểm kê

181 │ }

182 │ catch (Exception)

183 │ {

184 │ return 0; // Lỗi khi cập nhật

185 │ }

186 │ }

187 │ public int Delete(int Ma, Guid UserID)

188 │ {

189 │ StockChiTietKiemKe drItem = GetItem(Ma);

190 │

191 │ if (drItem != null)

192 │ {

193 │ DateTime dtNgayTao = DateTime.Now;

194 │ drItem.Deleted = true;

195 │ drItem.DeletedDate = dtNgayTao;

196 │ drItem.UserDeleted = UserID;

197 │

198 │ if (Update(drItem) > 0)

199 │ {

200 │ return 1;

201 │ }

202 │ return 0;

203 │ }

204 │ return -2;

205 │ }

206 │ }

207 │ }

────────────────────────────────────────────────────────────

**📄 ChiTietLichSuGiaModel.cs (CS, 107 lines)**

1 │ ﻿using Microsoft.EntityFrameworkCore;

2 │ using QuanLyBanHang.Common;

3 │ using QuanLyBanHang.Models.QLBHEntity;

4 │ using static QuanLyBanHang.Controllers.OPS.LichSuGiaController;

5 │

6 │ namespace QuanLyBanHang.Models.ExModel.OPS

7 │ {

8 │ public class ChiTietLichSuGiaModel : Base<TblChiTietLichSuGia>

9 │ {

10 │ public ChiTietLichSuGiaModel(DbContext dbContext) : base(dbContext)

11 │ {

12 │ }

13 │ QuanLyBanHangContext dbContext

14 │ {

15 │ get { return \_dbContext as QuanLyBanHangContext; }

16 │ set { return; }

17 │ }

18 │ public TblChiTietLichSuGia GetItem(int ma)

19 │ {

20 │ return dbContext.TblChiTietLichSuGia

21 │ .First(t => t.Ma == ma && (t.Deleted == null || t.Deleted == false));

22 │ }

23 │

24 │ public TblChiTietLichSuGia GetItem(int maLichSu, int maHangHoa, int maKho)

25 │ {

26 │ return dbContext.TblChiTietLichSuGia

27 │ .AsNoTracking()

28 │ .First(t =>

29 │ t.MaLichSu == maLichSu

30 │ && t.MaHangHoa == maHangHoa

31 │ && EF.Functions.Like(";" + t.MaKho + ";", "%;" + maKho.ToString() + ";%")

32 │ && (t.Deleted == null || t.Deleted == false)

33 │ );

34 │ }

35 │

36 │ public List<TblChiTietLichSuGia> GetData(int maLichSu)

37 │ {

38 │ return dbContext.TblChiTietLichSuGia

39 │ .Where(t => t.MaLichSu == maLichSu && (t.Deleted == null || t.Deleted == false))

40 │ .ToList();

41 │ }

42 │

43 │ public int SaveListChiTiet(int maLichSu, List<cCT\_LichSuGia> cts, Guid userId, DateTime dtNgayTao, bool deleteOl...

44 │ {

45 │ int row = 0;

46 │ try

47 │ {

48 │ if (deleteOld)

49 │ {

50 │ var ctOlds = dbContext.TblChiTietLichSuGia

51 │ .Where(t => t.MaLichSu == maLichSu && !(t.Deleted ?? false))

52 │ .ToList();

53 │

54 │ foreach (var ctOld in ctOlds)

55 │ {

56 │ var ctNew = cts.FirstOrDefault(t => Cast.ToInt(t.Ma.Decrypt()) == ctOld.Ma);

57 │ if (ctNew == null)

58 │ {

59 │ ctOld.Deleted = true;

60 │ ctOld.UserDeleted = userId;

61 │ ctOld.DeletedDate = dtNgayTao;

62 │ }

63 │ }

64 │ }

65 │

66 │ row = 1;

67 │ foreach (var ct in cts)

68 │ {

69 │ int maChiTiet = Cast.ToInt(ct.Ma.Decrypt());

70 │ var ctOld = dbContext.TblChiTietLichSuGia

71 │ .FirstOrDefault(t => t.Ma == maChiTiet && !(t.Deleted ?? false));

72 │

73 │ if (ctOld == null)

74 │ {

75 │ ctOld = new TblChiTietLichSuGia

76 │ {

77 │ MaLichSu = maLichSu,

78 │ MaHangHoa = ct.MaHangHoa,

79 │ GiaBanCuTheoKho = Utility.SetItemMultiSelect(ct.GiaBanCu),

80 │ GiaBanMoi = ct.GiaBanMoi,

81 │ GhiChu = ct.GhiChu,

82 │ MaKho = Utility.SetItemMultiSelect(ct.MaKho),

83 │ UserCreated = userId,

84 │ CreatedDate = dtNgayTao

85 │ };

86 │ dbContext.TblChiTietLichSuGia.Add(ctOld);

87 │ }

88 │ else

89 │ {

90 │ ctOld.GiaBanCuTheoKho = Utility.SetItemMultiSelect(ct.GiaBanCu);

91 │ ctOld.GiaBanMoi = ct.GiaBanMoi;

92 │ ctOld.MaKho = Utility.SetItemMultiSelect(ct.MaKho);

93 │ ctOld.GhiChu = ct.GhiChu;

94 │ }

95 │ row++;

96 │ }

97 │

98 │ \_dbContext.SaveChanges();

99 │ return 0;

100 │ }

101 │ catch

102 │ {

103 │ return row;

104 │ }

105 │ }

106 │ }

107 │ }

────────────────────────────────────────────────────────────

**📄 ChiTietNhapHangModel.cs (CS, 247 lines)**

1 │ ﻿using Microsoft.EntityFrameworkCore;

2 │ using NetTopologySuite.Index.HPRtree;

3 │ using QuanLyBanHang.Common;

4 │ using QuanLyBanHang.Models.QLBHEntity;

5 │ using static QuanLyBanHang.Controllers.OPS.NhapHangController;

6 │

7 │ namespace QuanLyBanHang.Models.ExModel.OPS

8 │ {

9 │ public class ChiTietNhapHangModel : Base<StockChiTietNhapHang>

10 │ {

11 │ public ChiTietNhapHangModel(DbContext dbContext) : base(dbContext)

12 │ {

13 │ }

14 │

15 │ QuanLyBanHangContext dbContext

16 │ {

17 │ get { return \_dbContext as QuanLyBanHangContext; }

18 │ set { return; }

19 │ }

20 │

21 │ public StockChiTietNhapHang? GetItem(int ma)

22 │ {

23 │ return First(t => t.Ma == ma);

24 │ }

25 │

26 │ public List<StockChiTietNhapHang> GetData(int maNhapHang)

27 │ {

28 │ return Get()

29 │ .Where(t => t.MaNhapHang == maNhapHang)

30 │ .ToList();

31 │ }

32 │

33 │ public List<VChiTietNhapHang> GetListChiTietNhapHang(Guid MaChiNhanh, DateTime tuNgay, DateTime denNgay, int maN...

34 │ {

35 │ return dbContext.VChiTietNhapHang

36 │ .Where(t => (t.NgayHoaDon ?? t.CreatedDate).Value.Date >= tuNgay &&

37 │ (t.NgayHoaDon ?? t.CreatedDate).Value.Date <= denNgay &&

38 │ (maNhaCC == 0 || t.MaNhaCc == maNhaCC) &&

39 │ (maKho == 0 || t.MaKho == maKho) && t.MaChiNhanh == MaChiNhanh)

40 │ .ToList();

41 │ }

42 │

43 │ public int SaveListChiTiet(int maNhapHang, List<cCT\_NhapHang> cts, Guid userId, DateTime ngayTao)

44 │ {

45 │ var nhapHang = dbContext.StockNhapHang.FirstOrDefault(x => x.Ma == maNhapHang);

46 │ if (nhapHang == null) return 0;

47 │

48 │ var chiNhanh = dbContext.SysBranch.FirstOrDefault(x => x.Ma == nhapHang.MaChiNhanh);

49 │ var maChiNhanhInt = chiNhanh?.Nid ?? 0;

50 │

51 │ var maHangHoaSet = cts.Select(x => x.MaHangHoa).Distinct().ToList();

52 │ var hangHoas = dbContext.TblHangHoa

53 │ .Where(x => maHangHoaSet.Contains(x.Ma))

54 │ .ToDictionary(x => x.Ma);

55 │

56 │ var existingItems = dbContext.StockChiTietNhapHang

57 │ .Where(t => t.MaNhapHang == maNhapHang && !(t.Deleted ?? false))

58 │ .ToList();

59 │

60 │ // --- Đánh dấu xóa các dòng không còn tồn tại ---

61 │ foreach (var item in existingItems)

62 │ {

63 │ var matched = cts.Any(ct =>

64 │ Cast.ToInt(ct.Ma.Decrypt()) == item.Ma || // dòng cũ

65 │ (ct.MaHangHoa == item.MaHangHoa &&

66 │ Cast.ToDateTime(ct.HanSD, "dd/MM/yyyy") == item.HanSd) // dòng trùng mã nhưng khác HSD

67 │ );

68 │

69 │ if (!matched)

70 │ {

71 │ item.Deleted = true;

72 │ item.UserDeleted = userId;

73 │ item.DeletedDate = ngayTao;

74 │ }

75 │ }

76 │

77 │ // --- Xử lý từng dòng chi tiết ---

78 │ foreach (var ct in cts)

79 │ {

80 │ var maCt = Cast.ToInt(ct.Ma.Decrypt());

81 │ var hanSd = Cast.ToDateTime(ct.HanSD, "dd/MM/yyyy");

82 │ var (tt, ttCk, ttVat) = CalculateAmounts(ct, nhapHang);

83 │

84 │ if (maCt == 0)

85 │ {

86 │ // Thêm mới

87 │ dbContext.StockChiTietNhapHang.Add(new StockChiTietNhapHang

88 │ {

89 │ MaNhapHang = maNhapHang,

90 │ MaChiNhanhInt = maChiNhanhInt,

91 │ MaHangHoa = ct.MaHangHoa,

92 │ SoLuong = (int)ct.SoLuong,

93 │ GiaVon = (decimal)ct.GiaVon,

94 │ HanSd = hanSd,

95 │ ChietKhauTm = (decimal?)ct.ChietKhauTM,

96 │ ChietKhauPt = (decimal?)ct.ChietKhauPT,

97 │ Vat = ct.VAT,

98 │ ThanhTien = tt,

99 │ ThanhTienSauCk = ttCk,

100 │ ThanhTienSauVat = ttVat,

101 │ UserCreated = userId,

102 │ CreatedDate = ngayTao

103 │ });

104 │ }

105 │ else

106 │ {

107 │ // Cập nhật

108 │ var item = existingItems.FirstOrDefault(x => x.Ma == maCt);

109 │ if (item != null)

110 │ {

111 │ item.MaChiNhanhInt = maChiNhanhInt;

112 │ item.MaHangHoa = ct.MaHangHoa;

113 │ item.SoLuong = (int)ct.SoLuong;

114 │ item.GiaVon = (decimal)ct.GiaVon;

115 │ item.HanSd = hanSd;

116 │ item.ChietKhauTm = (decimal?)ct.ChietKhauTM;

117 │ item.ChietKhauPt = (decimal?)ct.ChietKhauPT;

118 │ item.Vat = ct.VAT;

119 │ item.ThanhTien = tt;

120 │ item.ThanhTienSauCk = ttCk;

121 │ item.ThanhTienSauVat = ttVat;

122 │ item.UserModified = userId;

123 │ item.ModifiedDate = ngayTao;

124 │ }

125 │ }

126 │

127 │ // Cập nhật giá vốn nếu cần

128 │ if (hangHoas.TryGetValue(ct.MaHangHoa, out var hh))

129 │ {

130 │ var giaMoi = Cast.ToDecimal(ct.GiaVon);

131 │ if (hh.GiaVon != giaMoi)

132 │ {

133 │ hh.GiaVon = giaMoi;

134 │ dbContext.Update(hh);

135 │ }

136 │ }

137 │ }

138 │

139 │ dbContext.SaveChanges();

140 │ return 0;

141 │ }

142 │

143 │ private (decimal, decimal, decimal) CalculateAmounts(cCT\_NhapHang ct, StockNhapHang nhapHang)

144 │ {

145 │ var thanhtien = ct.SoLuong \* ct.GiaVon;

146 │ var thanhtiensauCK = CalculateDiscount(ct, nhapHang, thanhtien);

147 │ var thanhtiensauVAT = CalculateVAT(ct, nhapHang, thanhtiensauCK);

148 │

149 │ return (

150 │ (decimal)Math.Round(thanhtien, 4),

151 │ (decimal)Math.Round(thanhtiensauCK, 4),

152 │ (decimal)Math.Round(thanhtiensauVAT, 4)

153 │ );

154 │ }

155 │

156 │ private double CalculateDiscount(cCT\_NhapHang ct, StockNhapHang nhapHang, double thanhtien)

157 │ {

158 │ if (nhapHang.TienChietKhau.GetValueOrDefault() != 0)

159 │ {

160 │ ct.ChietKhauTM = 0;

161 │ ct.ChietKhauPT = 0;

162 │ }

163 │

164 │ return thanhtien - (ct.ChietKhauTM.GetValueOrDefault() + thanhtien \* ct.ChietKhauPT.GetValueOrDefault() / 10...

165 │ }

166 │

167 │ private double CalculateVAT(cCT\_NhapHang ct, StockNhapHang nhapHang, double thanhtiensauCK)

168 │ {

169 │ return nhapHang.ThueVat != 0

170 │ ? thanhtiensauCK

171 │ : thanhtiensauCK \* (100.0 + ct.VAT) / 100.0;

172 │ }

173 │

174 │ private void AddNewItem(int maNhapHang, cCT\_NhapHang ct, Guid userId, DateTime ngayTao,

175 │ decimal thanhtien, decimal thanhtiensauCK, decimal thanhtiensauVAT)

176 │ {

177 │ var nh = dbContext.StockNhapHang.FirstOrDefault(x => x.Ma == maNhapHang);

178 │ var br = dbContext.SysBranch.FirstOrDefault(x => x.Ma == nh.MaChiNhanh);

179 │

180 │ var newItem = new StockChiTietNhapHang

181 │ {

182 │ MaNhapHang = maNhapHang,

183 │ MaChiNhanhInt = br.Nid ?? 0,

184 │ MaHangHoa = ct.MaHangHoa,

185 │ SoLuong = (int)ct.SoLuong,

186 │ GiaVon = (int)ct.GiaVon,

187 │ HanSd = Cast.ToDateTime(ct.HanSD, "dd/MM/yyyy"),

188 │ ChietKhauTm = (decimal?)ct.ChietKhauTM,

189 │ ChietKhauPt = (decimal?)ct.ChietKhauPT,

190 │ Vat = ct.VAT,

191 │ ThanhTien = thanhtien,

192 │ ThanhTienSauCk = thanhtiensauCK,

193 │ ThanhTienSauVat = thanhtiensauVAT,

194 │ UserCreated = userId,

195 │ CreatedDate = ngayTao

196 │ };

197 │

198 │ dbContext.StockChiTietNhapHang.Add(newItem);

199 │ var MaHH = ct.MaHangHoa;

200 │ var GiaVon = Cast.ToDecimal(ct.GiaVon);

201 │ var hh = dbContext.TblHangHoa.FirstOrDefault(x => x.Ma == MaHH);

202 │ var oldGiaVon = hh?.GiaVon;

203 │ if (hh != null && GiaVon != oldGiaVon)

204 │ {

205 │ hh.GiaVon = GiaVon;

206 │ dbContext.Update(hh);

207 │ }

208 │ }

209 │

210 │ private void UpdateExistingItem(int maCt, cCT\_NhapHang ct, Guid userId, DateTime ngayTao,

211 │ decimal thanhtien, decimal thanhtiensauCK, decimal thanhtiensauVAT)

212 │ {

213 │ var existingItem = dbContext.StockChiTietNhapHang.Find(maCt);

214 │ if (existingItem == null) return;

215 │

216 │ var nh = dbContext.StockNhapHang.FirstOrDefault(x => x.Ma == existingItem.MaNhapHang);

217 │ var br = dbContext.SysBranch.FirstOrDefault(x => x.Ma == nh.MaChiNhanh);

218 │

219 │ existingItem.MaChiNhanhInt = br.Nid ?? 0;

220 │ existingItem.MaHangHoa = ct.MaHangHoa;

221 │ existingItem.SoLuong = (int)ct.SoLuong;

222 │ existingItem.GiaVon = (decimal)ct.GiaVon;

223 │ existingItem.HanSd = Cast.ToDateTime(ct.HanSD, "dd/MM/yyyy");

224 │ existingItem.ChietKhauTm = (decimal?)ct.ChietKhauTM;

225 │ existingItem.ChietKhauPt = (decimal?)ct.ChietKhauPT;

226 │ existingItem.Vat = ct.VAT;

227 │ existingItem.ThanhTien = thanhtien;

228 │ existingItem.ThanhTienSauCk = thanhtiensauCK;

229 │ existingItem.ThanhTienSauVat = thanhtiensauVAT;

230 │ existingItem.UserModified = userId;

231 │ existingItem.ModifiedDate = ngayTao;

232 │

233 │ dbContext.StockChiTietNhapHang.Update(existingItem);

234 │

235 │ var MaHH = ct.MaHangHoa;

236 │ var GiaVon = Cast.ToDecimal(ct.GiaVon);

237 │ var hh = dbContext.TblHangHoa.FirstOrDefault(x => x.Ma == MaHH);

238 │ var oldGiaVon = hh?.GiaVon;

239 │ if (hh != null && GiaVon != oldGiaVon)

240 │ {

241 │ hh.GiaVon = GiaVon;

242 │ dbContext.Update(hh);

243 │ }

244 │ }

245 │ }

246 │ }

247 │

────────────────────────────────────────────────────────────

**📄 ChiTietTonKhoModel.cs (CS, 49 lines)**

1 │ ﻿using Microsoft.EntityFrameworkCore;

2 │ using QuanLyBanHang.Common;

3 │ using QuanLyBanHang.Models.QLBHEntity;

4 │

5 │ namespace QuanLyBanHang.Models.ExModel.OPS

6 │ {

7 │ public class ChiTietTonKhoModel : Base<StockChiTietTonKho>

8 │ {

9 │ public ChiTietTonKhoModel(DbContext dbContext) : base(dbContext)

10 │ {

11 │ }

12 │

13 │ QuanLyBanHangContext dbContext

14 │ {

15 │ get { return \_dbContext as QuanLyBanHangContext; }

16 │ set { return; }

17 │ }

18 │

19 │ public int UpdateGhiChu(List<StockChiTietTonKho> ListChiTiet, Guid UserID)

20 │ {

21 │ int res = 0;

22 │ DateTime dtpNgaySua = DateTime.Now;

23 │ try

24 │ {

25 │ foreach (StockChiTietTonKho item in ListChiTiet)

26 │ {

27 │ StockChiTietTonKho ct = Get(t => t.Ma == Cast.ToInt(item.Ma)).First();

28 │ if (ct != null)

29 │ {

30 │ ct.GhiChu = item.GhiChu;

31 │ ct.UserModified = UserID;

32 │ ct.ModifiedDate = dtpNgaySua;

33 │

34 │ if (Update(ct) > 0)

35 │ {

36 │ res++;

37 │ }

38 │ }

39 │

40 │ }

41 │ }

42 │ catch (Exception ex)

43 │ {

44 │ return 0;

45 │ }

46 │ return res;

47 │ }

48 │ }

49 │ }

────────────────────────────────────────────────────────────

**📄 ChiTietXuatKhoModel.cs (CS, 162 lines)**

1 │ ﻿using Microsoft.EntityFrameworkCore;

2 │ using QuanLyBanHang.Common;

3 │ using QuanLyBanHang.Controllers.OPS;

4 │ using QuanLyBanHang.Models.QLBHEntity;

5 │ using QuanLyBanHang.Models.ViewModels;

6 │

7 │ namespace QuanLyBanHang.Models.ExModel.OPS

8 │ {

9 │ public class ChiTietXuatKhoModel : Base<StockChiTietXuatKho>

10 │ {

11 │ public ChiTietXuatKhoModel(DbContext dbContext) : base(dbContext)

12 │ {

13 │ }

14 │ QuanLyBanHangContext dbContext

15 │ {

16 │ get { return \_dbContext as QuanLyBanHangContext; }

17 │ set { return; }

18 │ }

19 │ public StockChiTietXuatKho GetItem(int Ma)

20 │ {

21 │ return First(t => t.Ma == Ma);

22 │ }

23 │

24 │ public List<StockChiTietXuatKho> GetData(int MaPhieuXuat)

25 │ {

26 │ return Get(t => t.MaPhieuXuat == MaPhieuXuat).ToList();

27 │ }

28 │ public List<ChiTietXuatKhoViewModel> GetViewChiTietXuatKho(Guid MaChiNhanh)

29 │ {

30 │ var result = (from ctxk in dbContext.StockChiTietXuatKho.Where(x => !(x.Deleted ?? false))

31 │ join xk in dbContext.StockXuatKho.Where(x => !(x.Deleted ?? false) && x.MaChiNhanh == MaChiNha...

32 │ join hh in dbContext.TblHangHoa.Where(x => !(x.Deleted ?? false)) on ctxk.MaHangHoa equals hh....

33 │ join dvt in dbContext.TblDonViTinh.Where(x => !(x.Deleted ?? false)) on hh.MaDvt equals dvt.Ma

34 │ join k in dbContext.StockDanhMucKho.Where(x => !(x.Deleted ?? false)) on xk.MaKho equals k.Ma

35 │ join l in dbContext.TblLopHangHoa.Where(x => !(x.Deleted ?? false)) on hh.MaLopHang equals l.M...

36 │ join k2 in dbContext.StockDanhMucKho.Where(x => !(x.Deleted ?? false)) on xk.MaKhoChuyenDen eq...

37 │ from k2 in khoChuyenDenGroup.DefaultIfEmpty()

38 │

39 │ select new ChiTietXuatKhoViewModel

40 │ {

41 │ Ma = ctxk.Ma,

42 │ MaPhieuXuat = ctxk.MaPhieuXuat,

43 │ MaHangHoa = ctxk.MaHangHoa,

44 │ MaVach = hh.MaVach,

45 │ TenHang = hh.Ten,

46 │ SoLuong = ctxk.SoLuong,

47 │ GiaVon = ctxk.GiaVon,

48 │ MaDVT = ctxk.MaDvt,

49 │ TenDVT = dvt.Ten,

50 │ TenLop = l.Ten,

51 │ NgayXuat = xk.NgayXuat,

52 │ SoPhieu = xk.SoPhieu,

53 │ KhoaSo = xk.KhoaSo,

54 │ GhiChu = xk.GhiChu,

55 │ MaKho = xk.MaKho,

56 │ TenKho = k.Ten,

57 │ GhiChuChiTiet = ctxk.GhiChu,

58 │ MaKT = hh.MaKt,

59 │ MaKhoChuyenDen = xk.MaKhoChuyenDen,

60 │ KhoChuyenDen = k2 != null ? k2.Ten : null,

61 │ MaChiNhanh = xk.MaChiNhanh ?? Utility.defaultUID

62 │ }).ToList();

63 │

64 │ return result;

65 │ }

66 │

67 │ public List<ChiTietXuatKhoViewModel> GetListChiTietXuatKho(int MaPhieuXuat, Guid MaChiNhanh)

68 │ {

69 │ return GetViewChiTietXuatKho(MaChiNhanh).Where(t => t.MaPhieuXuat == MaPhieuXuat).ToList();

70 │ }

71 │

72 │ public List<ChiTietXuatKhoViewModel> GetListChiTietXuatKho(DateTime TuNgay, DateTime DenNgay, int MaKho, Guid Ma...

73 │ {

74 │ return GetViewChiTietXuatKho(MaChiNhanh).Where(t => t.NgayXuat >= TuNgay && t.NgayXuat <= DenNgay && (MaKho ...

75 │ }

76 │ public int SaveListChiTiet(int MaXuatKho, List<ChiTietXuatKho> cts, Guid UserID, DateTime dtNgayTao)

77 │ {

78 │ int row = 0;

79 │ int ma\_ct = 0;

80 │ try

81 │ {

82 │ StockXuatKho? xk = dbContext.StockXuatKho.FirstOrDefault(t => t.Ma == MaXuatKho && !(t.Deleted ?? false)...

83 │ if (xk == null)

84 │ {

85 │ return -2;

86 │ }

87 │

88 │ List<StockChiTietXuatKho> cts\_old = dbContext.StockChiTietXuatKho.Where(t => t.MaPhieuXuat == MaXuatKho ...

89 │ foreach (StockChiTietXuatKho ct\_old in cts\_old)

90 │ {

91 │ ChiTietXuatKho? ct = cts.FirstOrDefault(t => Cast.ToInt(t.Ma.Decrypt()) == ct\_old.Ma);

92 │ if (ct == null)

93 │ {

94 │ ct\_old.Deleted = true;

95 │ ct\_old.UserDeleted = UserID;

96 │ ct\_old.DeletedDate = dtNgayTao;

97 │ dbContext.Update(ct\_old);

98 │ dbContext.SaveChanges();

99 │ }

100 │ }

101 │ row = 1;

102 │ foreach (ChiTietXuatKho ct in cts)

103 │ {

104 │ ma\_ct = Cast.ToInt(ct.Ma.Decrypt());

105 │ if (ma\_ct == 0)

106 │ {

107 │ decimal soluongton = dbContext.StockTonKho.Where(t => t.MaKho == xk.MaKho && t.MaHangHoa == Cast...

108 │ if (soluongton < Cast.ToInt(ct.SoLuong))

109 │ {

110 │ return -1000 \* row;

111 │ }

112 │ var br = dbContext.SysBranch.FirstOrDefault(x => x.Ma == xk.MaChiNhanh);

113 │

114 │ StockChiTietXuatKho ct\_new = new StockChiTietXuatKho

115 │ {

116 │ MaPhieuXuat = MaXuatKho,

117 │ MaChiNhanhInt = br.Nid ?? 0,

118 │ MaHangHoa = Cast.ToInt(ct.MaHangHoa),

119 │ MaDvt = Cast.ToInt(ct.MaDVT),

120 │ SoLuong = Cast.ToInt(ct.SoLuong),

121 │ GiaVon = Cast.ToDecimal(ct.GiaVon),

122 │ GhiChu = ct.GhiChu,

123 │ UserCreated = UserID,

124 │ CreatedDate = dtNgayTao,

125 │ };

126 │ dbContext.StockChiTietXuatKho.Add(ct\_new);

127 │ }

128 │ else

129 │ {

130 │ StockChiTietXuatKho? ct\_old = dbContext.StockChiTietXuatKho.FirstOrDefault(t => t.Ma == ma\_ct &&...

131 │ if (ct\_old == null)

132 │ return -1;

133 │ decimal soluongton = dbContext.StockTonKho.Where(t => t.MaKho == xk.MaKho && t.MaHangHoa == Cast...

134 │ if (soluongton + ct\_old.SoLuong < Cast.ToInt(ct.SoLuong))

135 │ {

136 │ return -1000 \* row;

137 │ }

138 │ var br = dbContext.SysBranch.FirstOrDefault(x => x.Ma == xk.MaChiNhanh);

139 │

140 │ ct\_old.MaHangHoa = Cast.ToInt(ct.MaHangHoa);

141 │ ct\_old.MaChiNhanhInt = br.Nid ?? 0;

142 │ ct\_old.MaDvt = Cast.ToInt(ct.MaDVT);

143 │ ct\_old.SoLuong = Cast.ToInt(ct.SoLuong);

144 │ ct\_old.GiaVon = Cast.ToDecimal(ct.GiaVon);

145 │ ct\_old.GhiChu = ct.GhiChu;

146 │ ct\_old.UserModified = UserID;

147 │ ct\_old.ModifiedDate = dtNgayTao;

148 │ dbContext.Update(ct\_old);

149 │ }

150 │

151 │ dbContext.SaveChanges();

152 │ row++;

153 │ }

154 │ return 0;

155 │ }

156 │ catch (Exception ex)

157 │ {

158 │ return row;

159 │ }

160 │ }

161 │ }

162 │ }

────────────────────────────────────────────────────────────

**📄 ChiTietXuatKhoViewModel.cs (CS, 27 lines)**

1 │ ﻿namespace QuanLyBanHang.Models.ViewModels

2 │ {

3 │ public class ChiTietXuatKhoViewModel

4 │ {

5 │ public int Ma { get; set; }

6 │ public int? MaPhieuXuat { get; set; }

7 │ public int? MaHangHoa { get; set; }

8 │ public string? MaVach { get; set; }

9 │ public string? TenHang { get; set; }

10 │ public decimal? SoLuong { get; set; }

11 │ public decimal? GiaVon { get; set; }

12 │ public int? MaDVT { get; set; }

13 │ public string? TenDVT { get; set; }

14 │ public string? TenLop { get; set; }

15 │ public DateTime? NgayXuat { get; set; }

16 │ public string? SoPhieu { get; set; }

17 │ public bool? KhoaSo { get; set; }

18 │ public string? GhiChu { get; set; }

19 │ public int? MaKho { get; set; }

20 │ public string? TenKho { get; set; }

21 │ public string? GhiChuChiTiet { get; set; }

22 │ public string? MaKT { get; set; }

23 │ public int? MaKhoChuyenDen { get; set; }

24 │ public string? KhoChuyenDen { get; set; }

25 │ public Guid MaChiNhanh { get; set; }

26 │ }

27 │ }

────────────────────────────────────────────────────────────

**📄 ChucVuController.cs (CS, 135 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using QuanLyBanHang.Common;

3 │ using QuanLyBanHang.Common.Attributes;

4 │ using QuanLyBanHang.Models.APIModels;

5 │ using QuanLyBanHang.Models.ExModel.SYS;

6 │ using QuanLyBanHang.Models.QLBHEntity;

7 │

8 │ namespace QuanLyBanHang.Controllers.SYS

9 │ {

10 │ public class ViewListChucVuModel : TblChucVu

11 │ {

12 │ public string? TenNhomChucVu { get; set; }

13 │ }

14 │ public class ChucVuController : BaseController

15 │ {

16 │ public ChucVuController(QuanLyBanHangContext dbcontext, MemoryCacheManager memoryCache) : base(dbcontext, memory...

17 │ {

18 │ m\_MenuID = MenuID.tblChucVu.GetEnumGuid();

19 │ }

20 │ [TokenUserCheckHTTP]

21 │ public IActionResult Index()

22 │ {

23 │ var data = (

24 │ from cv in new tbl\_ChucVuModel(dbcontext).GetData()

25 │ join \_nhcv in new tbl\_NhomChucVuModel(dbcontext).GetData() on cv.MaNhomChucVu equals \_nhcv.Ma into \_...

26 │ from nhcv in \_nhcv.DefaultIfEmpty()

27 │ select new ViewListChucVuModel

28 │ {

29 │ Ma = cv.Ma,

30 │ Code = cv.Code,

31 │ Ten = cv.Ten,

32 │ TenNhomChucVu = nhcv?.Ten + "",

33 │ Status = cv.Status,

34 │ }

35 │ ).ToList();

36 │

37 │ ViewData["ListData"] = data;

38 │ return View();

39 │ }

40 │ [TokenUserCheckHTTP(Permission.Add)]

41 │ public ActionResult Add()

42 │ {

43 │ ViewData["ListNhomChucVu"] = new tbl\_NhomChucVuModel(dbcontext).GetData();

44 │ return View("Form");

45 │ }

46 │

47 │ [TokenUserCheckHTTP(Permission.Edit)]

48 │ public ActionResult Edit(string Ma)

49 │ {

50 │ ViewData["ListNhomChucVu"] = new tbl\_NhomChucVuModel(dbcontext).GetData();

51 │ if (Ma == null || Ma == "")

52 │ {

53 │ return View("Index");

54 │ }

55 │

56 │ ViewData["item"] = new tbl\_ChucVuModel(dbcontext).GetItem(Ma.GetGuid());

57 │ return View("Form");

58 │ }

59 │

60 │ private bool CheckExists(Guid Ma, string Code)

61 │ {

62 │ return new tbl\_ChucVuModel(dbcontext).CheckExists(Ma, Code);

63 │ }

64 │

65 │ [HttpPost]

66 │ [TokenUserCheckHTTP(Permission.Add)]

67 │ public ActionResult AddAPI(string Code, string Ten, string TrangThai, string MaNhomChucVu)

68 │ {

69 │ bool StatusBool = TrangThai == ((int)Status.Active).ToString();

70 │ Guid MaNhomChucVuGuid = MaNhomChucVu == null || MaNhomChucVu == "" ? Utility.defaultUID : MaNhomChucVu.GetGu...

71 │

72 │ if (CheckExists(Utility.defaultUID, Code))

73 │ {

74 │ return Json(new ResponseModel(QLBHConst.Error, "Trùng mã"));

75 │ }

76 │ var res = new tbl\_ChucVuModel(dbcontext).Add(Code, Ten, StatusBool, MaNhomChucVuGuid, objToken!.userid);

77 │ if (res.row == 1)

78 │ {

79 │ return Json(QLBHConst.ResponseAddSuccessfully);

80 │ }

81 │ else

82 │ {

83 │ return Json(new ResponseModel(QLBHConst.Error, "Đã có lỗi xảy ra!"));

84 │ }

85 │ }

86 │

87 │ [HttpPost]

88 │ [TokenUserCheckHTTP(Permission.Edit)]

89 │ public ActionResult UpdateAPI(string Ma, string Code, string Ten, string TrangThai, string MaNhomChucVu)

90 │ {

91 │ Guid MaGuid = Ma == null || Ma == "" ? Utility.defaultUID : Ma.GetGuid();

92 │ Guid MaNhomChucVuGuid = MaNhomChucVu == null || MaNhomChucVu == "" ? Utility.defaultUID : MaNhomChucVu.GetGu...

93 │ var issetItem = new tbl\_ChucVuModel(dbcontext).Any(x => x.Ma == MaGuid);

94 │

95 │ if (!issetItem)

96 │ {

97 │ return Json(new ResponseModel(QLBHConst.Error, "Không tìm thấy dữ liệu!"));

98 │ }

99 │

100 │ bool StatusBool = TrangThai == ((int)Status.Active).ToString();

101 │ var res = new tbl\_ChucVuModel(dbcontext).Edit(new Guid(Ma), Code, Ten, StatusBool, MaNhomChucVuGuid, objToke...

102 │

103 │ if (res == 1)

104 │ {

105 │ return Json(QLBHConst.ResponseUpdateSuccessfully);

106 │ }

107 │ else

108 │ {

109 │ return Json(QLBHConst.ResponseUpdateFailure);

110 │ }

111 │ }

112 │

113 │ [HttpPost]

114 │ [TokenUserCheckHTTP(Permission.Delete)]

115 │ public ActionResult DeleteAPI(string Ma)

116 │ {

117 │ Guid MaGuid = Ma == null || Ma == "" ? Utility.defaultUID : Ma.GetGuid();

118 │

119 │ var res = new tbl\_ChucVuModel(dbcontext).Delete(MaGuid, objToken!.userid);

120 │

121 │ if (res == 1)

122 │ {

123 │ return Json(new ResponseModel(QLBHConst.OK, "Đã xóa thành công"));

124 │ }

125 │ else if (res == -2)

126 │ {

127 │ return Json(new ResponseModel(QLBHConst.Error, "Không tìm thấy dữ liệu!"));

128 │ }

129 │ else

130 │ {

131 │ return Json(new ResponseModel(QLBHConst.Error, "Đã có lỗi xảy ra!"));

132 │ }

133 │ }

134 │ }

135 │ }

────────────────────────────────────────────────────────────

**📄 CreateInvoiceRequest.cs (CS, 92 lines)**

1 │ ﻿namespace QuanLyBanHang.Models.EInvoiceService.Viettel

2 │ {

3 │ public class CreateInvoiceRequest

4 │ {

5 │ public GeneralInvoiceInfo? GeneralInvoiceInfo { get; set; }

6 │ public BuyerInfo? BuyerInfo { get; set; }

7 │ public List<Payment>? Payments { get; set; }

8 │ public List<ItemInfo>? ItemInfo { get; set; }

9 │ public List<TaxBreakdown>? TaxBreakdowns { get; set; }

10 │ public SummarizeInfo? SummarizeInfo { get; set; }

11 │ public List<Metadata>? Metadata { get; set; }

12 │ }

13 │

14 │ public class GeneralInvoiceInfo

15 │ {

16 │ public string? InvoiceType { get; set; } // Loại hóa đơn (TT78: 1-6)

17 │ public string? TemplateCode { get; set; } // Mẫu hóa đơn

18 │ public string? InvoiceSeries { get; set; } // Ký hiệu hóa đơn

19 │ public string? CurrencyCode { get; set; } // Mã tiền tệ (VND, USD...)

20 │ public decimal ExchangeRate { get; set; } // Tỷ giá

21 │ public string? AdjustmentType { get; set; } // Loại điều chỉnh (1, 3, 5, 7)

22 │ public bool PaymentStatus { get; set; } // Trạng thái thanh toán

23 │ public bool CusGetInvoiceRight { get; set; } // Cho phép tra cứu hóa đơn

24 │ public long? InvoiceIssuedDate { get; set; } // Unix timestamp hoặc null

25 │ public string? TransactionUuid { get; set; } // UUID duy nhất để tránh trùng giao dịch

26 │ }

27 │

28 │ public class BuyerInfo

29 │ {

30 │ public string? BuyerName { get; set; }

31 │ public string? BuyerLegalName { get; set; }

32 │ public string? BuyerTaxCode { get; set; }

33 │ public string? BuyerAddressLine { get; set; }

34 │ public string? BuyerPhoneNumber { get; set; }

35 │ public string? BuyerEmail { get; set; }

36 │ public string? BuyerIdNo { get; set; }

37 │ public string? BuyerIdType { get; set; }

38 │ public string? BuyerNotGetInvoice { get; set; } // "0" hoặc "1"

39 │ }

40 │

41 │ public class Payment

42 │ {

43 │ public string? PaymentMethodName { get; set; }

44 │ }

45 │

46 │ public class ItemInfo

47 │ {

48 │ public int LineNumber { get; set; }

49 │ public int Selection { get; set; } // 1: Hàng hóa, 2: Ghi chú, 3: Chiết khấu...

50 │ public string? ItemCode { get; set; }

51 │ public string? ItemName { get; set; }

52 │ public string? UnitName { get; set; }

53 │ public decimal? Quantity { get; set; }

54 │ public decimal? UnitPrice { get; set; }

55 │ public decimal? ItemTotalAmountWithoutTax { get; set; }

56 │ public decimal? ItemTotalAmountAfterDiscount { get; set; }

57 │ public decimal? ItemTotalAmountWithTax { get; set; }

58 │ public decimal? TaxPercentage { get; set; }

59 │ public decimal? TaxAmount { get; set; }

60 │ public decimal? Discount { get; set; }

61 │ public decimal? ItemDiscount { get; set; }

62 │ public string? ItemNote { get; set; }

63 │ public bool? IsIncreaseItem { get; set; }

64 │ }

65 │

66 │ public class TaxBreakdown

67 │ {

68 │ public decimal TaxPercentage { get; set; }

69 │ public decimal TaxableAmount { get; set; }

70 │ public decimal TaxAmount { get; set; }

71 │ }

72 │

73 │ public class SummarizeInfo

74 │ {

75 │ public decimal SumOfTotalLineAmountWithoutTax { get; set; }

76 │ public decimal TotalAmountAfterDiscount { get; set; }

77 │ public decimal TotalAmountWithoutTax { get; set; }

78 │ public decimal TotalTaxAmount { get; set; }

79 │ public decimal TotalAmountWithTax { get; set; }

80 │ public string? TotalAmountWithTaxInWords { get; set; }

81 │ public decimal DiscountAmount { get; set; }

82 │ }

83 │

84 │ public class Metadata

85 │ {

86 │ public string? KeyTag { get; set; }

87 │ public string? StringValue { get; set; }

88 │ public string? ValueType { get; set; }

89 │ public string? KeyLabel { get; set; }

90 │ }

91 │

92 │ }

────────────────────────────────────────────────────────────

**📄 CreateInvoiceResponse.cs (CS, 19 lines)**

1 │ ﻿namespace QuanLyBanHang.Models.EInvoiceService.Viettel

2 │ {

3 │ public class CreateInvoiceResponse

4 │ {

5 │ public string? ErrorCode { get; set; }

6 │ public string? Description { get; set; }

7 │ public InvoiceResult? Result { get; set; }

8 │ }

9 │

10 │ public class InvoiceResult

11 │ {

12 │ public string? SupplierTaxCode { get; set; }

13 │ public string? InvoiceNo { get; set; }

14 │ public string? TransactionID { get; set; }

15 │ public string? ReservationCode { get; set; }

16 │ public string? CodeOfTax { get; set; }

17 │ }

18 │

19 │ }

────────────────────────────────────────────────────────────

**📄 CtHoaDon.cs (CS, 47 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class CtHoaDon

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int HoaDonId { get; set; }

11 │

12 │ public int HangHoaId { get; set; }

13 │

14 │ public string MaVach { get; set; } = null!;

15 │

16 │ public int? MaDvt { get; set; }

17 │

18 │ public int SoLuong { get; set; }

19 │

20 │ public int DonGia { get; set; }

21 │

22 │ public int? GiaGiam { get; set; }

23 │

24 │ public int ThanhTien { get; set; }

25 │

26 │ public int? NguoiTao { get; set; }

27 │

28 │ public DateTime? NgayTao { get; set; }

29 │

30 │ public int? NguoiSua { get; set; }

31 │

32 │ public DateTime? NgaySua { get; set; }

33 │

34 │ public bool? DaXoa { get; set; }

35 │

36 │ public int? NguoiXoa { get; set; }

37 │

38 │ public DateTime? NgayXoa { get; set; }

39 │

40 │ public int? SoLuongTraLai { get; set; }

41 │

42 │ public int? ThanhTienTraLai { get; set; }

43 │

44 │ public DateTime? NgayTra { get; set; }

45 │

46 │ public int? NguoiTra { get; set; }

47 │ }

────────────────────────────────────────────────────────────

**📄 CtLichSuGia.cs (CS, 37 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class CtLichSuGia

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int? MaLichSu { get; set; }

11 │

12 │ public int? HangHoaId { get; set; }

13 │

14 │ public string? MaKho { get; set; }

15 │

16 │ public int? GiaBanCu { get; set; }

17 │

18 │ public int? GiaBanMoi { get; set; }

19 │

20 │ public string? GiaBanCuTheoKho { get; set; }

21 │

22 │ public string? GhiChu { get; set; }

23 │

24 │ public int? NguoiTao { get; set; }

25 │

26 │ public DateTime? NgayTao { get; set; }

27 │

28 │ public int? NguoiSua { get; set; }

29 │

30 │ public DateTime? NgaySua { get; set; }

31 │

32 │ public bool? DaXoa { get; set; }

33 │

34 │ public int? NguoiXoa { get; set; }

35 │

36 │ public DateTime? NgayXoa { get; set; }

37 │ }

────────────────────────────────────────────────────────────

**📄 CtNhapHang.cs (CS, 45 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class CtNhapHang

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int MaNhapHang { get; set; }

11 │

12 │ public int MaHangHoa { get; set; }

13 │

14 │ public decimal SoLuong { get; set; }

15 │

16 │ public decimal GiaVon { get; set; }

17 │

18 │ public DateTime? HanSd { get; set; }

19 │

20 │ public decimal? ChietKhauTm { get; set; }

21 │

22 │ public decimal? ChietKhauPt { get; set; }

23 │

24 │ public int? Vat { get; set; }

25 │

26 │ public decimal? ThanhTien { get; set; }

27 │

28 │ public decimal? ThanhTienSauCk { get; set; }

29 │

30 │ public decimal? ThanhTienSauVat { get; set; }

31 │

32 │ public int? NguoiTao { get; set; }

33 │

34 │ public DateTime? NgayTao { get; set; }

35 │

36 │ public int? NguoiSua { get; set; }

37 │

38 │ public DateTime? NgaySua { get; set; }

39 │

40 │ public bool? DaXoa { get; set; }

41 │

42 │ public int? NguoiXoa { get; set; }

43 │

44 │ public DateTime? NgayXoa { get; set; }

45 │ }

────────────────────────────────────────────────────────────

**📄 DanhMucKhoController.cs (CS, 345 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using Microsoft.EntityFrameworkCore;

3 │ using QuanLyBanHang.Common;

4 │ using QuanLyBanHang.Common.Attributes;

5 │ using QuanLyBanHang.Models.APIModels;

6 │ using QuanLyBanHang.Models.QLBHEntity;

7 │ using QuanLyBanHang.Models.VCEntity;

8 │

9 │ namespace QuanLyBanHang.Controllers.CAT

10 │ {

11 │ public class DanhMucKhoController : BaseController

12 │ {

13 │ private readonly QuanLyVoucherContext vcContext;

14 │ private readonly ILogger<DanhMucKhoController> \_logger;

15 │

16 │ public DanhMucKhoController(

17 │ QuanLyBanHangContext dbcontext,

18 │ QuanLyVoucherContext voucherContext,

19 │ MemoryCacheManager memoryCache,

20 │ ILogger<DanhMucKhoController> logger)

21 │ : base(dbcontext, memoryCache)

22 │ {

23 │ m\_MenuID = MenuID.catDanhMucKho.GetEnumGuid();

24 │ vcContext = voucherContext;

25 │ \_logger = logger;

26 │ }

27 │

28 │ [TokenUserCheckHTTP]

29 │ public IActionResult Index()

30 │ {

31 │ \_logger.LogInformation("Người dùng {UserId} truy cập trang danh sách Kho.", objToken.userid);

32 │ return View();

33 │ }

34 │

35 │ [TokenUserCheckHTTP(Permission.Add)]

36 │ public async Task<IActionResult> Add()

37 │ {

38 │ \_logger.LogInformation("Người dùng {UserId} truy cập trang thêm mới Kho.", objToken.userid);

39 │ try

40 │ {

41 │ ViewData["MaNganhVoucher"] = await GetNganhVoucherListAsync();

42 │ ViewData["MaLoaiKho"] = await dbcontext.StockLoaiKho.Where(t => !(t.Deleted ?? false)).ToListAsync();

43 │ ViewData["MaPhongBan"] = await dbcontext.TblPhongBan.Where(t => !(t.Deleted ?? false) && t.MaChiNhanh ==...

44 │ return View("Form");

45 │ }

46 │ catch (Exception ex)

47 │ {

48 │ \_logger.LogError(ex, "Lỗi khi tải dữ liệu cho trang thêm mới Kho.");

49 │ return RedirectToAction("Index");

50 │ }

51 │ }

52 │

53 │ [TokenUserCheckHTTP(Permission.Edit)]

54 │ public async Task<IActionResult> Edit(string Ma)

55 │ {

56 │ \_logger.LogInformation("Người dùng {UserId} yêu cầu chỉnh sửa kho có mã (đã mã hóa): {MaKho...

57 │

58 │ if (!Utility.TryDecryptToInt(Ma, out int id))

59 │ {

60 │ \_logger.LogWarning("Mã kho không hợp lệ được cung cấp: {MaKho}", Ma);

61 │ return Json(new ResponseModel(QLBHConst.Error, "Mã không hợp lệ"));

62 │ }

63 │

64 │ try

65 │ {

66 │ var item = await dbcontext.StockDanhMucKho

67 │ .AsNoTracking()

68 │ .FirstOrDefaultAsync(t => t.Ma == id && !(t.Deleted ?? false) && t.MaChiNhanh == objToken.branch);

69 │

70 │ if (item == null)

71 │ {

72 │ \_logger.LogWarning("Không tìm thấy kho có ID: {KhoId} để chỉnh sửa.", id);

73 │ }

74 │

75 │ ViewData["MaNganhVoucher"] = await GetNganhVoucherListAsync();

76 │ ViewData["MaLoaiKho"] = await dbcontext.StockLoaiKho.Where(t => !(t.Deleted ?? false)).ToListAsync();

77 │ ViewData["MaPhongBan"] = await dbcontext.TblPhongBan.Where(t => !(t.Deleted ?? false) && t.MaChiNhanh ==...

78 │ ViewData["item"] = item;

79 │ return View("Form");

80 │ }

81 │ catch (Exception ex)

82 │ {

83 │ \_logger.LogError(ex, "Lỗi khi tải dữ liệu để chỉnh sửa kho ID: {KhoId}", id);

84 │ return RedirectToAction("Index");

85 │ }

86 │ }

87 │

88 │ [HttpPost]

89 │ [TokenUserCheckHTTP(Permission.Add)]

90 │ public async Task<IActionResult> AddAPI(string Code, string Ten, string DienGiai, bool MayBanHang, bool BanHang,...

91 │ string TaiKhoanCo, string TaiKhoanKho, int MaNganhVoucher, string ThoiG...

92 │ {

93 │ \_logger.LogInformation("Bắt đầu xử lý AddAPI cho kho mới. Code: {KhoCode}, Ten: {KhoTen} bởi ng�...

94 │

95 │ var result = CheckRequire(Code, Ten, DienGiai);

96 │ if (result != null) return Json(result);

97 │

98 │ if (CheckExists(Code))

99 │ {

100 │ \_logger.LogWarning("Thêm mới thất bại: Mã kho '{KhoCode}' đã tồn tại.", Code);

101 │ return Json(new ResponseModel(QLBHConst.Error, "Mã kho đã tồn tại. Vui lòng nhập mã kho khác...

102 │ }

103 │

104 │ try

105 │ {

106 │ var item = new StockDanhMucKho

107 │ {

108 │ UserCreated = objToken.userid,

109 │ CreatedDate = DateTime.Now,

110 │ MaChiNhanh = objToken.branch

111 │ };

112 │

113 │ BindStockDanhMucKho(item, Code, Ten, DienGiai, MayBanHang, BanHang, NhapHang, TrangThai, TenKhoOrder, Ba...

114 │ TaiKhoanCo, TaiKhoanKho, MaNganhVoucher, ThoiGianMoBan, isAllDay, WebOrder, MaLoai...

115 │

116 │ await dbcontext.StockDanhMucKho.AddAsync(item);

117 │ await dbcontext.SaveChangesAsync();

118 │

119 │ \_logger.LogInformation("Thêm mới kho có ID: {KhoId} và Code: {KhoCode} thành công.", item.Ma, ite...

120 │ return Json(QLBHConst.ResponseAddSuccessfully);

121 │ }

122 │ catch (Exception ex)

123 │ {

124 │ \_logger.LogError(ex, "Đã xảy ra lỗi nghiêm trọng khi xử lý AddAPI cho kho Code: {KhoCode}.",...

125 │ return Json(new ResponseModel(QLBHConst.Error, "Đã có lỗi xảy ra từ hệ thống."));

126 │ }

127 │ }

128 │

129 │ [HttpPost]

130 │ [TokenUserCheckHTTP(Permission.Edit)]

131 │ public async Task<IActionResult> UpdateAPI(string Ma, string Code, string Ten, string DienGiai, bool MayBanHang,...

132 │ string TaiKhoanCo, string TaiKhoanKho, int MaNganhVoucher, string Thoi...

133 │ {

134 │ \_logger.LogInformation("Bắt đầu xử lý UpdateAPI cho kho mã (đã mã hóa): {MaKho} bởi người ...

135 │

136 │ if (!Utility.TryDecryptToInt(Ma, out int id))

137 │ {

138 │ \_logger.LogWarning("UpdateAPI thất bại: Mã kho không hợp lệ: {MaKho}", Ma);

139 │ return Json(new ResponseModel(QLBHConst.Error, "Mã không hợp lệ"));

140 │

141 │ }

142 │

143 │ var result = CheckRequire(Code, Ten, DienGiai);

144 │ if (result != null) return Json(result);

145 │

146 │ try

147 │ {

148 │ var item = await dbcontext.StockDanhMucKho.FirstOrDefaultAsync(x => x.Ma == id && !(x.Deleted ?? false))...

149 │ if (item == null)

150 │ {

151 │ \_logger.LogWarning("UpdateAPI thất bại: Không tìm thấy kho có ID: {KhoId}", id);

152 │ return Json(QLBHConst.ResponseNotExist);

153 │ }

154 │

155 │ BindStockDanhMucKho(item, Code, Ten, DienGiai, MayBanHang, BanHang, NhapHang, TrangThai, TenKhoOrder, Ba...

156 │ TaiKhoanCo, TaiKhoanKho, MaNganhVoucher, ThoiGianMoBan, isAllDay, WebOrder, MaLoai...

157 │

158 │ item.UserModified = objToken.userid;

159 │ item.ModifiedDate = DateTime.Now;

160 │

161 │ await dbcontext.SaveChangesAsync();

162 │ \_logger.LogInformation("Cập nhật kho có ID: {KhoId} thành công.", id);

163 │ return Json(QLBHConst.ResponseUpdateSuccessfully);

164 │ }

165 │ catch (Exception ex)

166 │ {

167 │ \_logger.LogError(ex, "Lỗi khi cập nhật kho ID: {KhoId}", id);

168 │ return Json(new ResponseModel(QLBHConst.Error, "Đã có lỗi xảy ra từ hệ thống."));

169 │ }

170 │ }

171 │

172 │ [HttpPost]

173 │ [TokenUserCheckHTTP(Permission.Delete)]

174 │ public async Task<IActionResult> DeleteAPI(string Ma)

175 │ {

176 │ \_logger.LogInformation("Bắt đầu xử lý DeleteAPI cho kho mã (đã mã hóa): {MaKho} bởi người ...

177 │

178 │ if (!Utility.TryDecryptToInt(Ma, out int id))

179 │ {

180 │ \_logger.LogWarning("DeleteAPI thất bại: Mã kho không hợp lệ: {MaKho}", Ma);

181 │ return Json(new ResponseModel(QLBHConst.Error, "Mã không hợp lệ."));

182 │

183 │ }

184 │

185 │ try

186 │ {

187 │ var item = await dbcontext.StockDanhMucKho.FirstOrDefaultAsync(x => x.Ma == id);

188 │ if (item == null)

189 │ {

190 │ \_logger.LogWarning("DeleteAPI thất bại: Không tìm thấy kho có ID: {KhoId}", id);

191 │ return Json(QLBHConst.ResponseNotExist);

192 │ }

193 │

194 │ item.Deleted = true;

195 │ item.UserDeleted = objToken.userid;

196 │ item.DeletedDate = DateTime.Now;

197 │

198 │ await dbcontext.SaveChangesAsync();

199 │ \_logger.LogInformation("Đã xóa (đánh dấu) kho có ID: {KhoId} thành công.", id);

200 │ return Json(QLBHConst.ResponseDeleteSuccessfully);

201 │ }

202 │ catch (Exception ex)

203 │ {

204 │ \_logger.LogError(ex, "Lỗi khi xóa kho ID: {KhoId}", id);

205 │ return Json(new ResponseModel(QLBHConst.Error, "Đã có lỗi xảy ra từ hệ thống."));

206 │ }

207 │ }

208 │

209 │ [TokenUserCheckHTTP(Permission.NotCheck)]

210 │ public async Task<IActionResult> getListData()

211 │ {

212 │ \_logger.LogInformation("Bắt đầu lấy dữ liệu danh sách kho cho người dùng {UserId}, chi nhán...

213 │ try

214 │ {

215 │ var listData = await (

216 │ from kho in dbcontext.StockDanhMucKho.AsNoTracking()

217 │ join stock in dbcontext.StockLoaiKho.AsNoTracking() on kho.MaLoaiKho equals stock.Ma into gStock

218 │ from stock in gStock.DefaultIfEmpty()

219 │ join phongban in dbcontext.TblPhongBan.AsNoTracking() on kho.MaPhongBan equals phongban.Ma into gPho...

220 │ from phongban in gPhongBan.DefaultIfEmpty()

221 │ where !(kho.Deleted ?? false) && kho.MaChiNhanh == objToken.branch

222 │ orderby kho.Ma

223 │ select new

224 │ {

225 │ maEncrypt = kho.Ma.Encrypt(),

226 │ kho.Ma,

227 │ kho.Code,

228 │ kho.Ten,

229 │ kho.DienGiai,

230 │ kho.MayBanHangTuDong,

231 │ kho.BanHang,

232 │ kho.NhapHang,

233 │ kho.TaiKhoanCo,

234 │ kho.TaiKhoanKho,

235 │ kho.MaNganhVoucher,

236 │ kho.ThoiGianBatDau,

237 │ kho.ThoiGianKetThuc,

238 │ trangthai = kho.Status ?? false,

239 │ tentrangthai = (kho.Status ?? false) ? "Hoạt động" : "Hết hoạt động",

240 │ TenKho = stock != null ? stock.Ten : "",

241 │ TenPhongBan = phongban != null ? phongban.Ten : ""

242 │ }

243 │ ).ToListAsync();

244 │

245 │ \_logger.LogInformation("Lấy thành công {Count} bản ghi danh sách kho.", listData.Count);

246 │

247 │ return Json(new ResponseModel(QLBHConst.OK, "", listData));

248 │ }

249 │ catch (Exception ex)

250 │ {

251 │ \_logger.LogError(ex, "Lỗi khi thực hiện getListData.");

252 │ return Json(new ResponseModel(QLBHConst.Error, "Lỗi hệ thống khi lấy danh sách."));

253 │ }

254 │ }

255 │

256 │ [HttpPost]

257 │ [TokenUserCheckHTTP(Permission.NotCheck)]

258 │ public IActionResult getListKhoNhap()

259 │ {

260 │ \_logger.LogInformation("Người dùng {UserId} lấy danh sách kho nhập.", objToken.userid);

261 │ try

262 │ {

263 │ var data = dbcontext.StockDanhMucKho.Where(x => x.MaChiNhanh == objToken.branch

264 │ && (x.Status ?? false) && !(x.Deleted ?? false))

265 │ .Select(t => new

266 │ {

267 │ ma = t.Ma + "",

268 │ code = t.Code + "",

269 │ ten = t.Ten + ""

270 │ }).ToList();

271 │ return Json(new ResponseModelSuccess("", data));

272 │ }

273 │ catch (Exception ex)

274 │ {

275 │ \_logger.LogError(ex, "Lỗi khi thực hiện getListKhoNhap.");

276 │ return Json(new ResponseModel(QLBHConst.Error, "Lỗi hệ thống khi lấy danh sách kho nhập."));

277 │ }

278 │ }

279 │

280 │ #region Private Methods

281 │

282 │ private async Task<List<TblNganh>> GetNganhVoucherListAsync()

283 │ {

284 │ return await vcContext.TblNganh

285 │ .AsNoTracking()

286 │ .Where(x => !(x.DaXoa ?? false))

287 │ .ToListAsync();

288 │ }

289 │

290 │ private ResponseModel CheckRequire(string Code, string Ten, string DienGiai)

291 │ {

292 │ if (Code.IsEmpty()) return new ResponseModel(QLBHConst.Error, "Vui lòng nhập mã kho");

293 │ if (Ten.IsEmpty()) return new ResponseModel(QLBHConst.Error, "Vui lòng nhập tên");

294 │ if (DienGiai.IsEmpty()) return new ResponseModel(QLBHConst.Error, "Vui lòng nhập diễn giải");

295 │ return null;

296 │ }

297 │

298 │ private bool CheckExists(string Code)

299 │ {

300 │ return dbcontext.StockDanhMucKho

301 │ .AsNoTracking()

302 │ .Any(t => t.Code == Code && !(t.Deleted ?? false));

303 │ }

304 │

305 │ private void BindStockDanhMucKho(StockDanhMucKho item, string Code, string Ten, string DienGiai,

306 │ bool MayBanHang, bool BanHang, bool NhapHang, bool TrangThai, string TenKhoOrde...

307 │ string TaiKhoanCo, string TaiKhoanKho, int MaNganhVoucher,

308 │ string ThoiGianMoBan, string isAllDay, bool WebOrder, string MaLoaiKho, string ...

309 │ {

310 │ var (startTime, endTime) = ParseTimeRange(ThoiGianMoBan, isAllDay);

311 │ item.Code = Code;

312 │ item.Ten = Ten;

313 │ item.TenHienThi = TenKhoOrder;

314 │ item.MayBanHangTuDong = MayBanHang;

315 │ item.NhapHang = NhapHang;

316 │ item.DienGiai = DienGiai;

317 │ item.BanHang = BanHang;

318 │ item.HienThiTrongWebOrder = WebOrder;

319 │ item.Status = TrangThai;

320 │ item.TaiKhoanCo = TaiKhoanCo;

321 │ item.TaiKhoanKho = TaiKhoanKho;

322 │ item.MaNganhVoucher = MaNganhVoucher;

323 │ item.ThoiGianBatDau = startTime;

324 │ item.ThoiGianKetThuc = endTime;

325 │ item.MaChiNhanh = item.MaChiNhanh != null ? item.MaChiNhanh : objToken.branch;

326 │ item.Bartender = Bartender;

327 │ item.MaLoaiKho = MaLoaiKho.ToInt();

328 │ item.MaPhongBan = MaPhongBan.GetGuid();

329 │ }

330 │

331 │ private (int?, int?) ParseTimeRange(string timeString, string isAllDay)

332 │ {

333 │ if (isAllDay == "true" || string.IsNullOrWhiteSpace(timeString)) return (null, null);

334 │ var parts = timeString.Split('|');

335 │ if (parts.Length == 2)

336 │ {

337 │ return (Cast.ToInt(parts[0]), Cast.ToInt(parts[1]));

338 │ }

339 │ \_logger.LogWarning("Định dạng ThoiGianMoBan không hợp lệ: {TimeString}", timeString);

340 │ return (null, null);

341 │ }

342 │

343 │ #endregion

344 │ }

345 │ }

────────────────────────────────────────────────────────────

**📄 DanhMucKhoModel.cs (CS, 86 lines)**

1 │ ﻿using Microsoft.EntityFrameworkCore;

2 │ using QuanLyBanHang.Common;

3 │ using QuanLyBanHang.Models.QLBHEntity;

4 │

5 │ namespace QuanLyBanHang.Models.ExModel.CAT

6 │ {

7 │ public class DanhMucKhoModel : Base<StockDanhMucKho>

8 │ {

9 │ public DanhMucKhoModel(DbContext dbContext) : base(dbContext)

10 │ {

11 │ }

12 │

13 │ QuanLyBanHangContext dbContext

14 │ {

15 │ get { return \_dbContext as QuanLyBanHangContext; }

16 │ set { return; }

17 │ }

18 │

19 │ public StockDanhMucKho GetItem(int ma)

20 │ {

21 │ return dbContext.StockDanhMucKho.FirstOrDefault(t => t.Ma == ma && !(t.Deleted ?? false));

22 │ }

23 │

24 │ public List<StockDanhMucKho> GetData(Guid maChiNhanh, List<int> khoIds = null)

25 │ {

26 │ var query = Get(t => !(t.Deleted ?? false) && t.MaChiNhanh == maChiNhanh);

27 │ if (khoIds != null && khoIds.Any())

28 │ query = query.Where(t => khoIds.Contains(t.Ma));

29 │

30 │ return query.OrderBy(t => t.Ten).ToList();

31 │ }

32 │ public List<StockDanhMucKho> GetDataByListID(List<int> lstID, Guid MaChiNhanh)

33 │ {

34 │ return Get(t => !(t.Deleted ?? false) && lstID.Contains(t.Ma) && t.MaChiNhanh == MaChiNhanh).OrderBy(t => t....

35 │ }

36 │ public List<StockDanhMucKho> DanhSachMayBanHangTuDong(Guid maChiNhanh)

37 │ {

38 │ return Get(t => !(t.Deleted ?? false) &&

39 │ (t.MayBanHangTuDong ?? false) &&

40 │ t.MaChiNhanh == maChiNhanh)

41 │ .OrderBy(t => t.Ten)

42 │ .ToList();

43 │ }

44 │

45 │ public List<StockDanhMucKho> GetDataByUser(Guid maNhanVien, Guid maChiNhanh)

46 │ {

47 │ var user = dbContext.SysUser

48 │ .AsNoTracking()

49 │ .FirstOrDefault(x => !(x.Deleted ?? false) && x.Ma == maNhanVien);

50 │

51 │ if (user == null || string.IsNullOrWhiteSpace(user.KhoHang))

52 │ return new List<StockDanhMucKho>();

53 │

54 │ var khoIds = user.KhoHang

55 │ .Split(';', StringSplitOptions.RemoveEmptyEntries)

56 │ .Select(Cast.ToInt)

57 │ .Where(id => id > 0)

58 │ .Distinct()

59 │ .ToList();

60 │

61 │ return Get(t => !(t.Deleted ?? false) &&

62 │ t.MaChiNhanh == maChiNhanh &&

63 │ khoIds.Contains(t.Ma))

64 │ .OrderBy(t => t.Ma)

65 │ .ToList();

66 │ }

67 │

68 │ public List<StockDanhMucKho> GetDataByIds(List<int> khoIds, Guid maChiNhanh)

69 │ {

70 │ if (khoIds == null || !khoIds.Any()) return new List<StockDanhMucKho>();

71 │

72 │ return dbContext.StockDanhMucKho

73 │ .AsNoTracking()

74 │ .Where(k => khoIds.Contains(k.Ma) && k.MaChiNhanh == maChiNhanh && !(k.Deleted ?? false))

75 │ .OrderBy(k => k.Ten)

76 │ .ToList();

77 │ }

78 │ public List<StockDanhMucKho> DanhMucKhoOrder(Guid MaChiNhanh)

79 │ {

80 │ var now = DateTime.Now;

81 │ return Get(t => !(t.Deleted ?? false) && t.MaChiNhanh == MaChiNhanh && (t.HienThiTrongWebOrder ?? false)

82 │ && t.MaChiNhanh == MaChiNhanh && (t.ThoiGianBatDau ?? 0) <= now.TimeOfDay.TotalHours && (t.ThoiGianKetThuc ?...

83 │ }

84 │

85 │ }

86 │ }

────────────────────────────────────────────────────────────

**📄 DebugController.cs (CS, 46 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using QuanLyBanHang.Common;

3 │ using QuanLyBanHang.Models.QLBHEntity;

4 │ using QuanLyBanHang.Services;

5 │

6 │ namespace QuanLyBanHang.Controllers.SYS

7 │ {

8 │ [ApiController]

9 │ [Route("api/debug")]

10 │ public class DebugController : BaseController

11 │ {

12 │ public DebugController(QuanLyBanHangContext dbcontext, MemoryCacheManager memoryCache) : base(dbcontext, memoryC...

13 │ {

14 │ }

15 │

16 │ public IActionResult Index()

17 │ {

18 │ return View();

19 │ }

20 │

21 │ [HttpGet("tonkho-queue")]

22 │ public IActionResult GetTonKhoQueueLength()

23 │ {

24 │ return Ok(new

25 │ {

26 │ TonKhoQueueLength = TonKhoBackgroundWorker.QueueLength

27 │ });

28 │ }

29 │ [HttpGet("tonkho-queue-list")]

30 │ public IActionResult PeekJobs()

31 │ {

32 │ return Ok(new

33 │ {

34 │ Jobs = TonKhoBackgroundWorker.GetQueueSnapshot()

35 │ });

36 │ }

37 │ [HttpGet("signalr-connection-count")]

38 │ public IActionResult GetSignalRConnectionCount()

39 │ {

40 │ return Ok(new

41 │ {

42 │ SignalRConnectionCount = ChatHUB.ConnectionCount

43 │ });

44 │ }

45 │ }

46 │ }

────────────────────────────────────────────────────────────

**📄 DinhMucDVKTController.cs (CS, 268 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using Microsoft.EntityFrameworkCore;

3 │ using QuanLyBanHang.Common;

4 │ using QuanLyBanHang.Common.Attributes;

5 │ using QuanLyBanHang.Models.APIModels;

6 │ using QuanLyBanHang.Models.ExModel.VTTH;

7 │ using QuanLyBanHang.Models.QLBHEntity;

8 │

9 │ namespace QuanLyBanHang.Controllers.VTTH

10 │ {

11 │ public class DinhMucDVKTController : BaseController

12 │ {

13 │ public DinhMucDVKTController(QuanLyBanHangContext dbcontext, MemoryCacheManager memoryCache) : base(dbcontext, m...

14 │ {

15 │ m\_MenuID = MenuID.vtthDinhMucDVKT.GetEnumGuid();

16 │ }

17 │

18 │ public class ChiTietDinhMuc

19 │ {

20 │ public int STT { get; set; }

21 │ public string? Ma { get; set; }

22 │ public string? MaDinhMuc { get; set; }

23 │ public string? MaHangHoa { get; set; }

24 │ public string? MaVatTu { get; set; }

25 │ public string? MaDvt { get; set; }

26 │ public string? SoLuong { get; set; }

27 │ public string? DonGiaTB { get; set; }

28 │ public string? ThanhTien { get; set; }

29 │ public string? GhiChu { get; set; }

30 │ public string? KhongTruKho { get; set; }

31 │ }

32 │ [TokenUserCheckHTTP()]

33 │ public IActionResult Index()

34 │ {

35 │ return View();

36 │ }

37 │ private string ConvertListChiTiet(List<ChiTietDinhMuc> cts, out List<VtthChiTietDinhMuc> listcts)

38 │ {

39 │ string error = "";

40 │ listcts = new List<VtthChiTietDinhMuc>();

41 │ try

42 │ {

43 │ foreach (var ct in cts)

44 │ {

45 │ listcts.Add(new VtthChiTietDinhMuc

46 │ {

47 │ Ma = ct.Ma.ToInt(),

48 │ MaDinhMuc = ct.MaDinhMuc.ToInt(),

49 │ MaHangHoa = ct.MaHangHoa.ToInt(),

50 │ MaVatTu = ct.MaVatTu.ToInt(),

51 │ MaDvt = ct.MaDvt.ToInt(),

52 │ SoLuong = ct.SoLuong.ToDecimal(),

53 │ DonGiaTb = ct.DonGiaTB.ToDecimal(),

54 │ ThanhTien = ct.ThanhTien.ToDecimal(),

55 │ GhiChu = ct.GhiChu,

56 │ KhongTruKho = ct.KhongTruKho == "1"

57 │ });

58 │ }

59 │ }

60 │ catch (Exception ex)

61 │ {

62 │ error = ex.Message;

63 │ }

64 │ return error;

65 │ }

66 │

67 │ [HttpPost]

68 │ [DisableRequestSizeLimit]

69 │ [TokenUserCheckHTTP(Permission.Add)]

70 │ public IActionResult AddAPI(string Code, string Ten, string QuyTrinhKT, string TaiLieuThamKhao, string GhiChu, s...

71 │ {

72 │ if (Ten.IsEmpty())

73 │ {

74 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng nhập tên"));

75 │ }

76 │

77 │ if (cts == null || cts.Count == 0)

78 │ {

79 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng nhập chi tiết"));

80 │ }

81 │ List<VtthChiTietDinhMuc> listcts;

82 │ string err = ConvertListChiTiet(cts, out listcts);

83 │ if (!err.IsEmpty())

84 │ {

85 │ return Json(new ResponseModel(QLBHConst.Error, err));

86 │ }

87 │ var result = new DinhMucDVKTModel(dbcontext).Add(Code, Ten, QuyTrinhKT, TaiLieuThamKhao

88 │ , GhiChu, TrangThai == "1", objToken!.userid, listcts, objToken!.branch);

89 │

90 │ if (result == 1)

91 │ {

92 │ return Json(new ResponseModel(QLBHConst.OK, "Lưu thành công"));

93 │ }

94 │ else if (result == -1)

95 │ {

96 │ return Json(new ResponseModel(QLBHConst.Error, "Trùng mã định mức"));

97 │ }

98 │ else if (result == -2)

99 │ {

100 │ return Json(new ResponseModel(QLBHConst.Error, "Trùng vật tư y tế"));

101 │ }

102 │ else if (result == -3)

103 │ {

104 │ return Json(new ResponseModel(QLBHConst.Error, "Trùng tên thương mại"));

105 │ }

106 │ else

107 │ {

108 │ return Json(new ResponseModel(QLBHConst.Error, "Lưu thất bại"));

109 │ }

110 │ }

111 │

112 │ [HttpPost]

113 │ [DisableRequestSizeLimit]

114 │ [TokenUserCheckHTTP(Permission.Edit)]

115 │ public IActionResult EditAPI(string Ma, string Code, string Ten, string QuyTrinhKT, string TaiLieuThamKhao, stri...

116 │ {

117 │ if (Code.IsEmpty())

118 │ {

119 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng nhập mã định mức"));

120 │ }

121 │

122 │ if (Ten.IsEmpty())

123 │ {

124 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng chọn nhà cung cấp"));

125 │ }

126 │

127 │ if (cts == null || cts.Count == 0)

128 │ {

129 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng nhập chi tiết nhập kho"));

130 │ }

131 │ List<VtthChiTietDinhMuc> listcts;

132 │ string err = ConvertListChiTiet(cts, out listcts);

133 │ if (!err.IsEmpty())

134 │ {

135 │ return Json(new ResponseModel(QLBHConst.Error, err));

136 │ }

137 │

138 │ int result = new DinhMucDVKTModel(dbcontext).Edit(Ma.ToInt(), Code, Ten, QuyTrinhKT, TaiLieuThamKhao

139 │ , GhiChu, TrangThai == "1", objToken!.userid, listcts);

140 │

141 │ if (result == 1)

142 │ {

143 │ return Json(new ResponseModel(QLBHConst.OK, "Lưu thành công"));

144 │ }

145 │ else if (result == -1)

146 │ {

147 │ return Json(new ResponseModel(QLBHConst.Error, "Trùng mã định mức"));

148 │ }

149 │ else if (result == -2)

150 │ {

151 │ return Json(new ResponseModel(QLBHConst.Error, "Trùng vật tư y tế"));

152 │ }

153 │ else if (result == -3)

154 │ {

155 │ return Json(new ResponseModel(QLBHConst.Error, "Trùng tên thương mại"));

156 │ }

157 │ else

158 │ {

159 │ return Json(new ResponseModel(QLBHConst.Error, "Lưu thất bại"));

160 │ }

161 │ }

162 │

163 │ [HttpPost]

164 │ [TokenUserCheckHTTP(Permission.Delete)]

165 │ public IActionResult DeleteAPI(string Ma)

166 │ {

167 │ var res = new DinhMucDVKTModel(dbcontext).Delete(Ma.ToInt(), objToken!.userid);

168 │ return Json(Utility.MakeResponseFromDeleteResult(res.ToString()));

169 │ }

170 │

171 │ [HttpPost]

172 │ [TokenUserCheckHTTP(Permission.NotCheck)]

173 │ public IActionResult GetListDinhMucDVKT(string MaDinhMuc, string TenDinhMuc)

174 │ {

175 │ var codeDM = (MaDinhMuc ?? "").ToUpper();

176 │ var tenDM = (TenDinhMuc ?? "").ToUpper();

177 │

178 │

179 │ var lstHDQuery = dbcontext.Set<VtthDinhMuc>().AsNoTracking().Where(hd =>

180 │ !(hd.Deleted ?? false)

181 │ && hd.CreatedDate != null

182 │ && (string.IsNullOrEmpty(codeDM) || (hd.Code ?? "").ToUpper().Contains(codeDM))

183 │ && (string.IsNullOrEmpty(tenDM) || (hd.Ten ?? "").ToUpper().Contains(tenDM))

184 │ );

185 │

186 │ var lstHD = lstHDQuery.Select(hd => new

187 │ {

188 │ ima = hd.Ma,

189 │ ma = hd.Ma.ToString(),

190 │ codedinhmuc = hd.Code ?? "",

191 │ tendinhmuc = hd.Ten ?? "",

192 │ quytrinhkt = hd.QuyTrinhKt ?? "",

193 │ tailieuthamkhao = hd.TaiLieuThamKhao ?? "",

194 │ ghichu = hd.GhiChu ?? "",

195 │ trangthai = hd.Status ?? false ? "1" : "0",

196 │ ordby = hd.CreatedDate.Value.ToString("yyyyMMddHHmmssfff")

197 │ }).ToList();

198 │

199 │ var maDinhMucList = lstHD.Select(t => t.ima).ToList();

200 │

201 │ var lstCT = (from ct in dbcontext.Set<VtthChiTietDinhMuc>().AsNoTracking()

202 │ join hh in dbcontext.Set<TblHangHoa>() on ct.MaHangHoa equals hh.Ma into \_hh

203 │ from hh in \_hh.DefaultIfEmpty()

204 │ join vt in dbcontext.Set<TblHangHoa>() on ct.MaVatTu equals vt.Ma into \_vt

205 │ from vt in \_vt.DefaultIfEmpty()

206 │ join dvt in dbcontext.Set<TblDonViTinh>() on ct.MaDvt equals dvt.Ma into \_dvt

207 │ from dvt in \_dvt.DefaultIfEmpty()

208 │ where !(ct.Deleted ?? false) && ct.MaDinhMuc.HasValue && maDinhMucList.Contains(ct.MaDinhMuc.Va...

209 │ select new

210 │ {

211 │ ma = ct.Ma.ToString(),

212 │ madinhmuc = ct.MaDinhMuc.ToString(),

213 │ madvt = ct.MaDvt.ToString(),

214 │ tendvt = dvt.Ten ?? "",

215 │ mahh = ct.MaHangHoa.ToString(),

216 │ mavattu = ct.MaVatTu.ToString(),

217 │ tenhh = hh.Ten ?? "",

218 │ codehh = hh.MaKt ?? "",

219 │ codevattu = vt.MaKt ?? "",

220 │ tenvattu = vt.Ten ?? "",

221 │ soluong = string.Format("{0:#,##0.###}", ct.SoLuong),

222 │ dongiatb = string.Format("{0:#,##0.##}", ct.DonGiaTb),

223 │ thanhtien = string.Format("{0:#,##0.##}", ct.ThanhTien),

224 │ ghichu = ct.GhiChu ?? "",

225 │ khongtrukho = ct.KhongTruKho ?? false ? "1" : "0"

226 │ }).ToList();

227 │

228 │ return Json(new ResponseModel(QLBHConst.OK, "", new

229 │ {

230 │ listdinhmuc = lstHD,

231 │ listchitiet = lstCT

232 │ }));

233 │ }

234 │

235 │ [HttpPost]

236 │ [TokenUserCheckHTTP(Permission.NotCheck)]

237 │ public IActionResult getDanhSachHangHoaVaVatTu()

238 │ {

239 │

240 │ var lstHH = (from hh in dbcontext.TblHangHoa.Where(x => !(x.Deleted ?? false) && x.MaChiNhanh == objToken.br...

241 │ join dvt in dbcontext.TblDonViTinh on hh.MaDvt equals dvt.Ma

242 │ select new

243 │ {

244 │ mahh = hh.Ma.ToString(),

245 │ madvt = hh.MaDvt.ToString(),

246 │ codehh = hh.MaKt,

247 │ tenhh = hh.Ten,

248 │ tendvt = dvt.Ten,

249 │ dongiatb = hh.GiaVon

250 │ }).ToList();

251 │ var lstVT = (from hh in dbcontext.TblHangHoa.Where(x => (x.VatTu ?? false) == true && !(x.Deleted ?? false) ...

252 │ join dvt in dbcontext.TblDonViTinh on hh.MaDvt equals dvt.Ma

253 │ select new

254 │ {

255 │ mavattu = hh.Ma + "",

256 │ codevattu = hh.MaKt + "",

257 │ tenvattu = hh.Ten + "",

258 │ madvt = dvt.Ma + "",

259 │ tendvt = dvt.Ten + "",

260 │ }).ToList();

261 │ return Json(new ResponseModelSuccess("", new

262 │ {

263 │ lstHH,

264 │ lstVT

265 │ }));

266 │ }

267 │ }

268 │ }

────────────────────────────────────────────────────────────

**📄 DinhMucDVKTModel.cs (CS, 280 lines)**

1 │ ﻿using Microsoft.EntityFrameworkCore;

2 │ using QuanLyBanHang.Common.MultiServer;

3 │ using QuanLyBanHang.Models.QLBHEntity;

4 │

5 │ namespace QuanLyBanHang.Models.ExModel.VTTH

6 │ {

7 │ public class DinhMucDVKTModel : Base<VtthDinhMuc>

8 │ {

9 │ public DinhMucDVKTModel(DbContext dbContext) : base(dbContext)

10 │ {

11 │ }

12 │ QuanLyBanHangContext dbContext

13 │ {

14 │ get { return \_dbContext as QuanLyBanHangContext; }

15 │ set { return; }

16 │ }

17 │ public class ChiTietVatTuCongKhaiTheoDinhMuc

18 │ {

19 │ public Guid MaVatTu { get; set; }

20 │ public Guid MaDVT { get; set; }

21 │ public Guid MaCTNhap { get; set; }

22 │ public string? SoLo { get; set; }

23 │ public DateTime? HanSuDung { get; set; }

24 │ public decimal SoLuong { get; set; }

25 │ }

26 │

27 │ public VtthDinhMuc GetItem(int Ma)

28 │ {

29 │ return First(t => t.Ma == Ma && !(t.Deleted ?? false));

30 │ }

31 │ public List<VtthDinhMuc> GetData()

32 │ {

33 │ return Get(t => !(t.Deleted ?? false)).OrderByDescending(t => t.CreatedDate).ToList();

34 │ }

35 │

36 │ private bool CheckExists(int Ma, string Code)

37 │ {

38 │ return Any(t => t.Ma != Ma && t.Code == Code && !(t.Deleted ?? false));

39 │ }

40 │

41 │ public int Add(

42 │ string Code,

43 │ string Ten,

44 │ string QuyTrinhKT,

45 │ string TaiLieuThamKhao,

46 │ string GhiChu,

47 │ bool TrangThai,

48 │ Guid UserId,

49 │ List<VtthChiTietDinhMuc> listct,

50 │ Guid branch

51 │ )

52 │ {

53 │ if (HasDuplicate(listct, x => x.MaHangHoa)) return -2;

54 │ if (HasDuplicate(listct, x => x.MaVatTu)) return -3;

55 │

56 │ try

57 │ {

58 │ using var transaction = \_dbContext.Database.BeginTransaction();

59 │

60 │ var item = new VtthDinhMuc

61 │ {

62 │ Ten = Ten,

63 │ QuyTrinhKt = QuyTrinhKT,

64 │ TaiLieuThamKhao = TaiLieuThamKhao,

65 │ MaChiNhanh = branch,

66 │ GhiChu = GhiChu,

67 │ Status = TrangThai,

68 │ UserCreated = UserId,

69 │ CreatedDate = DateTime.Now

70 │ };

71 │

72 │ dbContext.VtthDinhMuc.Add(item);

73 │ dbContext.SaveChanges();

74 │

75 │ int res = new ChiTietDinhMucDVKTModel(\_dbContext).SaveListChiTiet(item.Ma, listct, UserId);

76 │ if (res == 0)

77 │ {

78 │ transaction.Rollback();

79 │ return 0;

80 │ }

81 │

82 │ var sophieu = MultiServer.GetData("VtthDinhMuc", "Code").Result;

83 │ item.Code = $"{sophieu:000000}";

84 │

85 │ \_dbContext.SaveChanges(); // Save code cập nhật

86 │ transaction.Commit();

87 │ return 1;

88 │ }

89 │ catch

90 │ {

91 │ return 0;

92 │ }

93 │ }

94 │

95 │ bool HasDuplicate(List<VtthChiTietDinhMuc> list, Func<VtthChiTietDinhMuc, int?> selector)

96 │ {

97 │ return list

98 │ .Where(e => selector(e).HasValue && selector(e).Value != 0)

99 │ .GroupBy(selector)

100 │ .Any(g => g.Count() > 1);

101 │ }

102 │

103 │ public int Edit(int Ma, string Code, string Ten, string QuyTrinhKT,

104 │ string TaiLieuThamKhao, string GhiChu, bool TrangThai, Guid UserId, List<VtthChiTietDinhMuc> listct)

105 │ {

106 │ var item = GetItem(Ma);

107 │ if (item == null)

108 │ return -4;

109 │

110 │ if (HasDuplicate(listct, x => x.MaHangHoa)) return -2;

111 │ if (HasDuplicate(listct, x => x.MaVatTu)) return -3;

112 │

113 │ using var transaction = \_dbContext.Database.BeginTransaction();

114 │ try

115 │ {

116 │ item.Code = Code;

117 │ item.Ten = Ten;

118 │ item.QuyTrinhKt = QuyTrinhKT;

119 │ item.TaiLieuThamKhao = TaiLieuThamKhao;

120 │ item.GhiChu = GhiChu;

121 │ item.Status = TrangThai;

122 │ item.UserModified = UserId;

123 │ item.ModifiedDate = DateTime.Now;

124 │

125 │ \_dbContext.Update(item);

126 │ \_dbContext.SaveChanges();

127 │

128 │ int res = new ChiTietDinhMucDVKTModel(\_dbContext).SaveListChiTiet(Ma, listct, UserId);

129 │ if (res == 0)

130 │ {

131 │ transaction.Rollback();

132 │ return 0;

133 │ }

134 │

135 │ transaction.Commit();

136 │ return 1;

137 │ }

138 │ catch

139 │ {

140 │ transaction.Rollback();

141 │ return 0;

142 │ }

143 │ }

144 │

145 │ public int Delete(int Ma, Guid UserID)

146 │ {

147 │ VtthDinhMuc item = GetItem(Ma);

148 │

149 │ if (item != null)

150 │ {

151 │ item.Deleted = true;

152 │ item.DeletedDate = DateTime.Now;

153 │ item.UserDeleted = UserID;

154 │ if (Update(item) > 0)

155 │ {

156 │ return 1;

157 │ }

158 │ return 0;

159 │ }

160 │ return -2;

161 │ }

162 │

163 │ //public List<ChiTietXuatKho> setListVatTuTruKho(int MaDinhMuc, int MaKho, int Nam, int Thang)

164 │ //{

165 │ // string sNamThang = Nam.ToString() + Thang.ToString();

166 │ // List<ChiTietVatTuCongKhaiTheoDinhMuc> list = new List<ChiTietVatTuCongKhaiTheoDinhMuc>();

167 │ // List<StockChiTietXuatKho> listctXuat = new List<StockChiTietXuatKho>();

168 │ // var chitiet = new ChiTietDinhMucDVKTModel(\_dbContext).GetData(MaDinhMuc);

169 │ // foreach (var ch in chitiet)

170 │ // {

171 │ // decimal SoLuongThucXuat = ch.SoLuong ?? 0;

172 │ // List<VTonKho> listTK = \_dbContext.Set<VTonKho>().Where(t => t.NamThang == sNamThang

173 │ // && t.MaKho == MaKho && t.MaHangHoa == ch.MaHangHoa && !(t.TreoNhapKho ?? false) && t.Sl...

174 │ // .ToList();

175 │

176 │ // decimal TongSoLuongTon = listTK.Sum(y => y.SltonCuoiKy ?? 0);

177 │ // // nếu có 1 hàng hoá trong dv kỹ thuật ko đủ thì ko cho công khai

178 │ // if (TongSoLuongTon < ch.SoLuong)

179 │ // {

180 │ // return new List<ChiTietXuatKho>();

181 │ // }

182 │ // else

183 │ // {

184 │ // foreach (VTonKho tk in listTK)

185 │ // {

186 │ // decimal soluongxuat = Cast.ToDecimal(tk.SltonCuoiKy ?? 0) >= SoLuongThucXuat ? SoLuongThucXuat...

187 │ // StockChiTietXuatKho ct = new StockChiTietXuatKho();

188 │ // ct.MaVatTu = tk.MaVatTu;

189 │ // ct.MaCtnhapKho = tk.MaCtnhap;

190 │ // ct.MaDvt = tk.MaDvt;

191 │ // ct.SoLuongThucXuat = soluongxuat;

192 │ // ct.SoLo = tk.SoLo;

193 │ // ct.HanSuDung = tk.HanSuDung;

194 │ // listctXuat.Add(ct);

195 │ // SoLuongThucXuat = SoLuongThucXuat - soluongxuat;

196 │

197 │ // if (SoLuongThucXuat <= 0)

198 │ // {

199 │ // break;

200 │ // }

201 │ // }

202 │ // }

203 │ // }

204 │ // return listctXuat;

205 │ //}

206 │

207 │ //public int setCongKhaiDichVuCoDinhMuc(Guid MaDichVu, Guid MaKho, int Nam, int Thang)

208 │ //{

209 │ // CatDichVu dvkt = new cat\_DichVuModel(\_dbContext).GetItem(MaDichVu);

210 │ // List<MedChiTietXuatKho> listctXuat = new List<MedChiTietXuatKho>();

211 │

212 │ // List<SerChiTietDinhMuc> chitiet = new ser\_ChiTietDinhMucDVKTModel(\_dbContext).GetData(dvkt.MaDinhMuc ?? Ut...

213 │ // if (chitiet.Count == 0)

214 │ // {

215 │ // return 0;

216 │ // }

217 │ // else

218 │ // {

219 │ // foreach (SerChiTietDinhMuc ch in chitiet)

220 │ // {

221 │ // decimal SoLuongThucXuat = ch.SoLuong ?? 0;

222 │ // List<VTonKho> listTK = \_dbContext.Set<VTonKho>().Where(t => t.Nam == Nam && t.Thang == Thang

223 │ // && t.MaKho == MaKho && t.MaHangHoa == ch.MaHangHoa && !(t.TreoNhapKho ?? false) && ...

224 │ // .OrderBy(t => t.HanSuDung).OrderBy(t => t.MaCtnhap).ToList();

225 │

226 │ // decimal TongSoLuongTon = listTK.Sum(y => y.SoLuongTon ?? 0);

227 │ // // nếu có 1 hàng hoá trong dv kỹ thuật ko đủ thì ko cho công khai

228 │ // if (TongSoLuongTon < ch.SoLuong)

229 │ // {

230 │ // return -1; // không đủ tồn kho

231 │ // }

232 │ // else

233 │ // {

234 │ // foreach (VTonKho tk in listTK)

235 │ // {

236 │ // decimal soluongxuat = Cast.ToDecimal(tk.SoLuongTon ?? 0) >= SoLuongThucXuat ? SoLuongThucX...

237 │ // MedChiTietXuatKho ct = new MedChiTietXuatKho();

238 │ // ct.MaVatTu = tk.MaVatTu;

239 │ // ct.MaCtnhapKho = tk.MaCtnhap;

240 │ // ct.MaDvt = tk.MaDvt;

241 │ // ct.SoLuongThucXuat = soluongxuat;

242 │ // ct.SoLo = tk.SoLo;

243 │ // ct.HanSuDung = tk.HanSuDung;

244 │ // listctXuat.Add(ct);

245 │ // SoLuongThucXuat = SoLuongThucXuat - soluongxuat;

246 │

247 │ // if (SoLuongThucXuat <= 0)

248 │ // {

249 │ // break;

250 │ // }

251 │ // }

252 │ // }

253 │ // }

254 │ // // cập nhật vào chi tiết sử dụng dịch vụ

255 │

256 │ // // cập nhật tồn kho

257 │ // List<MedTonKho> listTKTheoMaKho = \_dbContext.Set<MedTonKho>().Where(t => t.Nam == Nam && t.Thang == Th...

258 │ // List<MedTonKho> list\_capnhattonkho = (from tk in listTKTheoMaKho

259 │ // join xk in listctXuat on tk.MaVatTu equals xk.MaVatTu

260 │ // where tk.MaCtnhap == xk.MaCtnhapKho

261 │ // select new { a = tk, b = xk }).ToList().Select(t =>

262 │ // {

263 │ // t.a.MaVatTu = t.b.MaVatTu ?? Utility.defaultUID;

264 │ // t.a.MaCtnhap = t.b.MaCtnhapKho ?? Utility.defaultUID;

265 │ // t.a.SoLuongXuat = (t.a.SoLuongXuat ?? 0) + t.b.SoLuongThucXu...

266 │ // t.a.SoLuongTon = (t.a.SoLuongDauKy ?? 0) + (t.a.SoLuongNhap ...

267 │ // return t.a;

268 │ // }).ToList();

269 │

270 │ // if (list\_capnhattonkho != null && list\_capnhattonkho.Count > 0)

271 │ // {

272 │ // \_dbContext.Set<MedTonKho>().UpdateRange(list\_capnhattonkho);

273 │ // \_dbContext.SaveChanges();

274 │ // }

275 │

276 │ // return listctXuat.Count;

277 │ // }

278 │ //}

279 │ }

280 │ }

────────────────────────────────────────────────────────────

**📄 DiscordEmbed.cs (CS, 23 lines)**

1 │ ﻿using QuanLyBanHang.Common;

2 │ using System.Text.Json.Serialization;

3 │

4 │ namespace HRMUngTuyen.Models.DTO

5 │ {

6 │ public class DiscordEmbed

7 │ {

8 │ [JsonPropertyName("title")]

9 │ public string? Title { get; set; }

10 │

11 │ [JsonPropertyName("description")]

12 │ public string? Description { get; set; }

13 │

14 │ [JsonPropertyName("color")]

15 │ public int Color { get; set; } = QLBHConst.DISCORD\_COLOR; // Màu xanh mặc định

16 │

17 │ [JsonPropertyName("fields")]

18 │ public List<DiscordEmbedField> Fields { get; set; } = new();

19 │

20 │ [JsonPropertyName("footer")]

21 │ public DiscordEmbedFooter? Footer { get; set; }

22 │ }

23 │ }

────────────────────────────────────────────────────────────

**📄 DiscordEmbedField.cs (CS, 16 lines)**

1 │ ﻿using System.Text.Json.Serialization;

2 │

3 │ namespace HRMUngTuyen.Models.DTO

4 │ {

5 │ public class DiscordEmbedField

6 │ {

7 │ [JsonPropertyName("name")]

8 │ public string Name { get; set; }

9 │

10 │ [JsonPropertyName("value")]

11 │ public string Value { get; set; }

12 │

13 │ [JsonPropertyName("inline")]

14 │ public bool Inline { get; set; }

15 │ }

16 │ }

────────────────────────────────────────────────────────────

**📄 DiscordEmbedFooter.cs (CS, 10 lines)**

1 │ ﻿using System.Text.Json.Serialization;

2 │

3 │ namespace HRMUngTuyen.Models.DTO

4 │ {

5 │ public class DiscordEmbedFooter

6 │ {

7 │ [JsonPropertyName("text")]

8 │ public string Text { get; set; }

9 │ }

10 │ }

────────────────────────────────────────────────────────────

**📄 DiscordSerilogSink.cs (CS, 66 lines)**

1 │ ﻿using HRMUngTuyen.Models.DTO;

2 │ using QuanLyBanHang.Common;

3 │ using QuanLyBanHang.Services;

4 │ using Serilog.Core;

5 │ using Serilog.Events;

6 │

7 │ namespace HRMUngTuyen.Services.Logs

8 │ {

9 │ public class DiscordSerilogSink : ILogEventSink

10 │ {

11 │ private readonly IServiceProvider \_serviceProvider;

12 │ private readonly IFormatProvider? \_formatProvider;

13 │ private DiscordNotificationService? \_notificationService; // Dịch vụ sẽ được lấy sau

14 │

15 │ // 1. Nhận IServiceProvider thay vì INotificationService

16 │ public DiscordSerilogSink(IServiceProvider serviceProvider, IFormatProvider? formatProvider = null)

17 │ {

18 │ \_serviceProvider = serviceProvider;

19 │ \_formatProvider = formatProvider;

20 │ }

21 │

22 │ public void Emit(LogEvent logEvent)

23 │ {

24 │ // Chỉ gửi log khi mức độ là Error hoặc cao hơn

25 │ if (logEvent.Level < LogEventLevel.Error)

26 │ {

27 │ return;

28 │ }

29 │

30 │ // 2. Lấy dịch vụ INotificationService một cách "lười biếng" (lazy)

31 │ // Nó chỉ được lấy ở lần đầu tiên một log lỗi thực sự cần được gửi đi.

32 │ \_notificationService ??= \_serviceProvider.GetRequiredService<DiscordNotificationService>();

33 │

34 │ var message = logEvent.RenderMessage(\_formatProvider);

35 │

36 │ var embed = new DiscordEmbed

37 │ {

38 │ Title = "🚨 Lỗi Hệ Thống",

39 │ Color = 15158332, // Màu đỏ

40 │ Description = $"Một lỗi đã xảy ra vào lúc: \*\*{logEvent.Timestamp:HH:mm:ss dd/MM/yyyy}\*\*"

41 │ };

42 │

43 │ embed.Fields.Add(new DiscordEmbedField { Name = "Level", Value = logEvent.Level.ToString(), Inline = true })...

44 │

45 │ if (logEvent.Properties.TryGetValue("SourceContext", out var sourceContext))

46 │ {

47 │ embed.Fields.Add(new DiscordEmbedField { Name = "Category", Value = sourceContext.ToString().Trim('"'), ...

48 │ }

49 │

50 │ embed.Fields.Add(new DiscordEmbedField { Name = "Message", Value = message, Inline = false });

51 │ if (logEvent.Exception != null)

52 │ {

53 │ embed.Fields.Add(new DiscordEmbedField { Name = "Exception Type", Value = logEvent.Exception.GetType().N...

54 │

55 │ var stackTrace = logEvent.Exception.StackTrace?.Length > 1020

56 │ ? logEvent.Exception.StackTrace[..1020] + "..."

57 │ : logEvent.Exception.StackTrace;

58 │

59 │ embed.Fields.Add(new DiscordEmbedField { Name = "Stack Trace", Value = $"```{stackTrace ?? "N/A"}```", I...

60 │ }

61 │

62 │ // Gửi đi bằng dịch vụ đã được lấy

63 │ \_ = \_notificationService.SendRichNotificationAsync(NotificationChannel.Error, embed);

64 │ }

65 │ }

66 │ }

────────────────────────────────────────────────────────────

**📄 DiscordSinkExtensions.cs (CS, 18 lines)**

1 │ ﻿using Serilog;

2 │ using Serilog.Configuration;

3 │

4 │ namespace HRMUngTuyen.Services.Logs

5 │ {

6 │ public static class DiscordSinkExtensions

7 │ {

8 │ public static LoggerConfiguration Discord(

9 │ this LoggerSinkConfiguration sinkConfiguration,

10 │ IServiceProvider services, // 'services' ở đây chính là IServiceProvider

11 │ IFormatProvider? formatProvider = null)

12 │ {

13 │ // Không lấy dịch vụ ở đây nữa.

14 │ // Thay vào đó, truyền thẳng IServiceProvider vào trong Sink.

15 │ return sinkConfiguration.Sink(new DiscordSerilogSink(services, formatProvider));

16 │ }

17 │ }

18 │ }

────────────────────────────────────────────────────────────

**📄 DonViTinh.cs (CS, 27 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class DonViTinh

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string Ten { get; set; } = null!;

11 │

12 │ public bool? TrangThai { get; set; }

13 │

14 │ public int? NguoiTao { get; set; }

15 │

16 │ public DateTime? NgayTao { get; set; }

17 │

18 │ public int? NguoiSua { get; set; }

19 │

20 │ public DateTime? NgaySua { get; set; }

21 │

22 │ public bool? DaXoa { get; set; }

23 │

24 │ public int? NguoiXoa { get; set; }

25 │

26 │ public DateTime? NgayXoa { get; set; }

27 │ }

────────────────────────────────────────────────────────────

**📄 DonViTinhController.cs (CS, 214 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using Microsoft.EntityFrameworkCore;

3 │ using QuanLyBanHang.Common;

4 │ using QuanLyBanHang.Common.Attributes;

5 │ using QuanLyBanHang.Models.APIModels;

6 │ using QuanLyBanHang.Models.QLBHEntity;

7 │

8 │ namespace QuanLyBanHang.Controllers.CAT

9 │ {

10 │ public class DonViTinhController : BaseController

11 │ {

12 │ private readonly ILogger<DonViTinhController> \_logger;

13 │

14 │ public DonViTinhController(

15 │ QuanLyBanHangContext dbcontext,

16 │ MemoryCacheManager memoryCache,

17 │ ILogger<DonViTinhController> logger) : base(dbcontext, memoryCache)

18 │ {

19 │ m\_MenuID = MenuID.catDonViTinh.GetEnumGuid();

20 │ \_logger = logger;

21 │ }

22 │

23 │ [TokenUserCheckHTTP]

24 │ public IActionResult Index()

25 │ {

26 │ \_logger.LogInformation("Người dùng {UserId} truy cập trang danh sách Đơn vị tính.", objToken.use...

27 │ return View();

28 │ }

29 │

30 │ [TokenUserCheckHTTP(Permission.Add)]

31 │ public IActionResult Add()

32 │ {

33 │ \_logger.LogInformation("Người dùng {UserId} truy cập trang thêm mới Đơn vị tính.", objToken.us...

34 │ return View("Form");

35 │ }

36 │

37 │ [TokenUserCheckHTTP(Permission.Edit)]

38 │ public async Task<IActionResult> Edit(string Ma)

39 │ {

40 │ \_logger.LogInformation("Người dùng {UserId} yêu cầu chỉnh sửa ĐVT có mã (đã mã hóa): {MaDV...

41 │ if (!Utility.TryDecryptToInt(Ma, out int id))

42 │ {

43 │ \_logger.LogWarning("Mã ĐVT không hợp lệ được cung cấp: {MaDVT}", Ma);

44 │ return Json(new ResponseModel(QLBHConst.Error, "Mã không hợp lệ"));

45 │ }

46 │

47 │ try

48 │ {

49 │ var item = await dbcontext.TblDonViTinh

50 │ .AsNoTracking()

51 │ .FirstOrDefaultAsync(t => t.Ma == id && !(t.Deleted ?? false));

52 │

53 │ if (item == null)

54 │ {

55 │ \_logger.LogWarning("Không tìm thấy ĐVT có ID: {DVTId} để chỉnh sửa.", id);

56 │ }

57 │

58 │ ViewData["item"] = item;

59 │ return View("Form");

60 │ }

61 │ catch (Exception ex)

62 │ {

63 │ \_logger.LogError(ex, "Lỗi khi tải dữ liệu để chỉnh sửa ĐVT ID: {DVTId}", id);

64 │ return RedirectToAction("Index");

65 │ }

66 │ }

67 │

68 │ [HttpPost]

69 │ [TokenUserCheckHTTP(Permission.Edit)]

70 │ public async Task<IActionResult> UpdateAPI(string Ma, string Ten, string TrangThai)

71 │ {

72 │ \_logger.LogInformation("Người dùng {UserId} bắt đầu xử lý lưu (Thêm/Sửa) ĐVT. Mã (đã mã...

73 │ if (!Utility.TryDecryptToInt(Ma, out int id))

74 │ {

75 │ \_logger.LogWarning("UpdateAPI thất bại: Mã ĐVT không hợp lệ: {MaDVT}", Ma);

76 │ return Json(new ResponseModel(QLBHConst.Error, "Mã không hợp lệ"));

77 │ }

78 │

79 │ var result = CheckRequire(Ten, TrangThai);

80 │ if (result != null)

81 │ return Json(result);

82 │

83 │ bool isNew = id == 0;

84 │

85 │ if (isNew)

86 │ {

87 │ \_logger.LogInformation("Thêm mới ĐVT với tên: {TenDVT}", Ten);

88 │ try

89 │ {

90 │ var item = new TblDonViTinh

91 │ {

92 │ Ten = Ten,

93 │ Status = TrangThai == "1",

94 │ CreatedDate = DateTime.Now,

95 │ UserCreated = objToken.userid

96 │ };

97 │

98 │ await dbcontext.TblDonViTinh.AddAsync(item);

99 │ await dbcontext.SaveChangesAsync();

100 │ \_logger.LogInformation("Thêm mới ĐVT có ID: {DVTId} thành công.", item.Ma);

101 │ return Json(QLBHConst.ResponseAddSuccessfully);

102 │ }

103 │ catch (Exception ex)

104 │ {

105 │ \_logger.LogError(ex, "Lỗi khi thêm mới ĐVT. Dữ liệu: Tên={TenDVT}", Ten);

106 │ return Json(new ResponseModel(QLBHConst.Error, "Lỗi hệ thống khi thêm mới."));

107 │ }

108 │ }

109 │ else

110 │ {

111 │ \_logger.LogInformation("Cập nhật ĐVT có ID: {DVTId}", id);

112 │ try

113 │ {

114 │ var item = await dbcontext.TblDonViTinh.FirstOrDefaultAsync(x => x.Ma == id && !(x.Deleted ?? false)...

115 │ if (item == null)

116 │ {

117 │ \_logger.LogWarning("UpdateAPI thất bại: Không tìm thấy ĐVT có ID: {DVTId}", id);

118 │ return Json(QLBHConst.ResponseNotExist);

119 │ }

120 │

121 │ item.Ten = Ten;

122 │ item.Status = TrangThai == "1";

123 │ item.UserModified = objToken.userid;

124 │ item.ModifiedDate = DateTime.Now;

125 │

126 │ await dbcontext.SaveChangesAsync();

127 │ \_logger.LogInformation("Cập nhật ĐVT có ID: {DVTId} thành công.", id);

128 │ return Json(QLBHConst.ResponseUpdateSuccessfully);

129 │ }

130 │ catch (Exception ex)

131 │ {

132 │ \_logger.LogError(ex, "Lỗi khi cập nhật ĐVT có ID: {DVTId}", id);

133 │ return Json(new ResponseModel(QLBHConst.Error, "Lỗi hệ thống khi cập nhật."));

134 │ }

135 │ }

136 │ }

137 │

138 │ [HttpPost]

139 │ [TokenUserCheckHTTP(Permission.Delete)]

140 │ public async Task<IActionResult> DeleteAPI(string Ma)

141 │ {

142 │ \_logger.LogInformation("Người dùng {UserId} yêu cầu xóa ĐVT. Mã (đã mã hóa): {MaDVT}", objToke...

143 │ if (!Utility.TryDecryptToInt(Ma, out int id))

144 │ {

145 │ \_logger.LogWarning("DeleteAPI thất bại: Mã ĐVT không hợp lệ: {MaDVT}", Ma);

146 │ return Json(new ResponseModel(QLBHConst.Error, "Mã không hợp lệ"));

147 │ }

148 │

149 │ try

150 │ {

151 │ var item = await dbcontext.TblDonViTinh.FirstOrDefaultAsync(x => x.Ma == id);

152 │ if (item == null)

153 │ {

154 │ \_logger.LogWarning("DeleteAPI thất bại: Không tìm thấy ĐVT có ID: {DVTId}", id);

155 │ return Json(QLBHConst.ResponseNotExist);

156 │ }

157 │

158 │ item.Deleted = true;

159 │ item.UserDeleted = objToken.userid;

160 │ item.DeletedDate = DateTime.Now;

161 │

162 │ await dbcontext.SaveChangesAsync();

163 │ \_logger.LogInformation("Đã xóa (đánh dấu) ĐVT có ID: {DVTId} thành công.", id);

164 │ return Json(QLBHConst.ResponseDeleteSuccessfully);

165 │ }

166 │ catch (Exception ex)

167 │ {

168 │ \_logger.LogError(ex, "Lỗi khi xóa ĐVT có ID: {DVTId}", id);

169 │ return Json(new ResponseModel(QLBHConst.Error, "Lỗi hệ thống khi xóa."));

170 │ }

171 │ }

172 │

173 │ private static ResponseModel CheckRequire(string Ten, string TrangThai)

174 │ {

175 │ if (Ten.IsEmpty())

176 │ return new ResponseModel(QLBHConst.Error, "Vui lòng nhập tên");

177 │

178 │ if (TrangThai.IsEmpty())

179 │ return new ResponseModel(QLBHConst.Error, "Vui lòng nhập trạng thái");

180 │

181 │ return null;

182 │ }

183 │

184 │ [TokenUserCheckHTTP(Permission.NotCheck)]

185 │ public async Task<IActionResult> getListData()

186 │ {

187 │ \_logger.LogInformation("Bắt đầu lấy danh sách Đơn vị tính.");

188 │ try

189 │ {

190 │ var listData = await dbcontext.TblDonViTinh

191 │ .AsNoTracking()

192 │ .Where(x => !(x.Deleted ?? false))

193 │ .Select(n => new

194 │ {

195 │ maEncrypt = n.Ma.Encrypt(),

196 │ Ma = n.Ma,

197 │ Ten = n.Ten,

198 │ trangthai = n.Status ?? false,

199 │ tentrangthai = (n.Status ?? false) ? "Hoạt động" : "Hết hoạt động",

200 │ })

201 │ .OrderBy(t => t.Ma)

202 │ .ToListAsync();

203 │

204 │ \_logger.LogInformation("Lấy thành công {Count} bản ghi Đơn vị tính.", listData.Count);

205 │ return Json(new ResponseModel(QLBHConst.OK, "", listData));

206 │ }

207 │ catch (Exception ex)

208 │ {

209 │ \_logger.LogError(ex, "Lỗi khi lấy danh sách Đơn vị tính.");

210 │ return Json(new ResponseModel(QLBHConst.Error, "Lỗi hệ thống khi lấy danh sách."));

211 │ }

212 │ }

213 │ }

214 │ }

────────────────────────────────────────────────────────────

**📄 DonViTinhModel.cs (CS, 42 lines)**

1 │ ﻿using Microsoft.EntityFrameworkCore;

2 │ using QuanLyBanHang.Models.QLBHEntity;

3 │

4 │ namespace QuanLyBanHang.Models.ExModel.CAT

5 │ {

6 │ public class DonViTinhModel : Base<TblDonViTinh>

7 │ {

8 │ public DonViTinhModel(DbContext dbContext) : base(dbContext)

9 │ {

10 │ }

11 │ QuanLyBanHangContext dbContext

12 │ {

13 │ get { return \_dbContext as QuanLyBanHangContext; }

14 │ set { return; }

15 │ }

16 │ public TblDonViTinh GetItem(int Ma)

17 │ {

18 │ return First(t => t.Ma == Ma && !(t.Deleted ?? false));

19 │ }

20 │

21 │ public string TenDonViTinh(int MaDV)

22 │ {

23 │ return First(t => t.Ma == MaDV && !(t.Deleted ?? false)).Ten ?? "";

24 │ }

25 │

26 │ public List<TblDonViTinh> GetData()

27 │ {

28 │ return Get(t => !(t.Deleted ?? false)).OrderBy(t => t.Ten).ToList();

29 │ }

30 │

31 │

32 │ public string TenDonViTinhTheoHH(int MaHH)

33 │ {

34 │ var donViTinh = Get(t => dbContext.TblHangHoa.Any(h => h.Ma == MaHH && h.MaDvt == t.Ma)

35 │ && !(t.Deleted ?? false))

36 │ .OrderBy(t => t.Ten)

37 │ .FirstOrDefault();

38 │

39 │ return donViTinh?.Ten ?? "";

40 │ }

41 │ }

42 │ }

────────────────────────────────────────────────────────────

**📄 Enum.cs (CS, 526 lines)**

1 │ ﻿using QuanLyBanHang.Common.Attributes;

2 │

3 │ namespace QuanLyBanHang.Common

4 │ {

5 │ public enum LopHang

6 │ {

7 │ MatKinh = 336,

8 │ }

9 │

10 │ public enum TrangThaiVietQR

11 │ {

12 │ ChoThanhToan = 1,

13 │ DaThanhToan = 2,

14 │ DaHuy = 3,

15 │ DaGachNoTrenHIS = 4,

16 │ }

17 │ public enum RoleID

18 │ {

19 │ [EnumGuid("25DBCF99-51EE-4D0F-B0F3-A376F77A82A3")]

20 │ Admin = 1,

21 │ }

22 │ public enum GroupNotification

23 │ {

24 │ ThongBaoOrder = 1,

25 │ }

26 │ public enum TrangThaiQR

27 │ {

28 │ ChoThanhToan = 1,

29 │ DaThanhToan = 2,

30 │ DaHuy = 3,

31 │ DaDung = 4

32 │ }

33 │ public enum LoaiNhapXuat

34 │ {

35 │ PhieuNhapTuNCC = 1,

36 │ PhieuNhapChuyenKho = 2,

37 │ PhieuXuatChuyenKho = 3,

38 │ PhieuXuatTieuHao = 4,

39 │ PhieuXuatTraNhaCC = 5,

40 │ PhieuXuatKiemKe = 6,

41 │ PhieuXuatThanhLy = 7,

42 │ PhieuNhapDongBo = 8,

43 │ }

44 │ public enum LoaiKho

45 │ {

46 │ KhoNguyenVatLieu = 1,

47 │ KhoBanHang = 2,

48 │ KhoTieuHao = 3,

49 │ }

50 │ public enum ELoaiBaoCao

51 │ {

52 │ BaoCaoTonKhoTheoThang = 1,

53 │ BaoCaoTonKhoTheoNgay = 2,

54 │ BaoCaoTongHopNhapHang = 3,

55 │ BaoCaoTongHopXuatHang = 4,

56 │ BaoCaoNhapHangImportFast = 5,

57 │ BaoCaoTonKhoCoGiaVon = 6

58 │ }

59 │ public enum HinhThucThanhToan

60 │ {

61 │ TienMat = 1,

62 │ The = 2,

63 │ ChuyenKhoan = 3,

64 │ AR = 4

65 │ }

66 │ public enum Permission

67 │ {

68 │ [EnumGuid()]

69 │ NotCheck,

70 │

71 │ [EnumGuid("AE00BFF4-A49A-4303-BF5F-AC9E54A28F9C")]

72 │ Add,

73 │

74 │ [EnumGuid("D3E25BC4-658B-41A1-B9F8-793890D06771")]

75 │ Edit,

76 │

77 │ [EnumGuid("CE245AA0-8B94-427A-9941-98DF3AA21A0C")]

78 │ Delete,

79 │

80 │ [EnumGuid("A91DD6BA-A892-4D9E-95E1-1F06E4FE0334")]

81 │ ResetPassword,

82 │

83 │ [EnumGuid("BF31E07A-9EC7-4920-9EEB-D2986E14725A")]

84 │ ShowAll,

85 │

86 │ [EnumGuid("35B668B3-2630-4B5E-88E6-35321555224E")]

87 │ ShowAllBranch,

88 │

89 │ [EnumGuid("2E420BAD-93EA-40F0-B96D-DA92AC0D5E52")]

90 │ LoginAs,

91 │

92 │ [EnumGuid("388582bb-7273-4f00-8a9e-dfb4a0568da2")]

93 │ View,

94 │

95 │ [EnumGuid("52385B85-8DD2-4167-B2A1-FE6ED54E2D53")]

96 │ PhanQuyenRole,

97 │

98 │ [EnumGuid("B7D8E2E5-147F-42BB-8079-2A5C3102352B")]

99 │ CheckHetHang,

100 │

101 │ [EnumGuid("27D04B6E-D369-4B11-9E2E-F6E468133586")]

102 │ CheckHangHoaVatTu,

103 │

104 │ [EnumGuid("81C5600C-91D8-48A4-9ADB-A17CE1B8B8D7")]

105 │ KhoaSo,

106 │

107 │ [EnumGuid("956B6824-926A-42ED-B31F-7805DB1161E1")]

108 │ MoKhoaSo,

109 │

110 │ [EnumGuid("d5ae8055-0175-4c47-826e-04a43bcdda6a")]

111 │ SuaGiaBan,

112 │

113 │ [EnumGuid("D4F59862-B8A5-4237-B137-643352A2EE0A")]

114 │ SuaGiaVon,

115 │

116 │ [EnumGuid("afd7511e-0250-4025-9a10-989178bfb9d9")]

117 │ DuyetSuaGiaBan,

118 │

119 │ [EnumGuid("870168f3-9c33-42d4-ac16-9fd43b603b24")]

120 │ BoDuyetSuaGiaBan,

121 │

122 │ [EnumGuid("154D0649-3184-4AF3-BF6E-43C156974568")]

123 │ KhuyenMaiTienMat,

124 │

125 │ [EnumGuid("D48C9FA6-32DE-454D-8F69-1F708AC9F3F5")]

126 │ SuaHoaDonQuaNgay,

127 │

128 │ [EnumGuid("4580B1BE-478F-4453-9C15-A2F89EBC1329")]

129 │ TraHang,

130 │

131 │ [EnumGuid("26DEC8F3-BB84-4951-BD34-09131FB0ECAE")]

132 │ TraHangQuaNgay,

133 │

134 │ [EnumGuid("B6D73F55-B287-4149-9477-07624D9B9D32")]

135 │ InLaiHoaDon,

136 │

137 │ [EnumGuid("B82D853E-C680-4E2E-A31B-7289226B6BDF")]

138 │ InLaiHoaDonTraHang,

139 │

140 │ [EnumGuid("018ecf49-a24a-4f4e-b9df-9f5189d19bbf")]

141 │ ChotTonKho,

142 │

143 │ [EnumGuid("e4aabaa1-61b7-4c9e-b268-81b23538297d")]

144 │ NhapVTTH,

145 │

146 │ [EnumGuid("8185ec4d-e624-4d47-b203-6e8083ab6706")]

147 │ XuatVTTH,

148 │

149 │ [EnumGuid("b30b5b43-7799-460f-aefd-79c015856fd4")]

150 │ XacNhanTraVTTH,

151 │

152 │ [EnumGuid("088aa2b8-6dab-4f61-a367-1c1d804b49fb")]

153 │ TraVTTH,

154 │

155 │ [EnumGuid("567cf06a-9a7d-4d99-8f5b-881814f3b96e")]

156 │ SuaPhieuNhapDaDuyet,

157 │

158 │ [EnumGuid("23327f7b-9984-4be2-b8c1-ef57ba055fea")]

159 │ SuaSoHoaDonNhapKho,

160 │

161 │ [EnumGuid("4d78fef2-b3db-42cb-96b7-32a8102549a3")]

162 │ SuaLoPhieuDaNhapKho,

163 │

164 │ [EnumGuid("5ce271f2-94a0-43a9-9128-5958cbb5e89a")]

165 │ ChoPhepDoiMa,

166 │

167 │ [EnumGuid("E45F0D07-99A7-46CD-B31A-F3F01A788E0F")]

168 │ ManageMember,

169 │

170 │ [EnumGuid("38da4241-fb53-4b1d-8121-465a0f22c4aa")]

171 │ DuyetNhapKho,

172 │

173 │ [EnumGuid("37730b83-97de-4eaa-84e3-cf78fb5bb3c8")]

174 │ SuaChietKhau,

175 │

176 │ [EnumGuid("9950E8B9-66DE-400A-8833-A7C1C32AEC1E")]

177 │ CapNhatTrangThaiDonHang,

178 │

179 │ [EnumGuid("A83FC8BE-FEEC-44F7-848E-1A76A3708CB5")]

180 │ ThanhToanDonHang,

181 │

182 │ [EnumGuid("24F1E4E1-5D1F-4B5C-8C08-C7E11E449504")]

183 │ XacNhanThanhToan,

184 │ }

185 │ public enum MenuID

186 │ {

187 │ [EnumGuid("4EDFE951-E2C3-4833-A8D8-C20A9C00EA35")]

188 │ sysRole,

189 │

190 │ [EnumGuid("F4475CFC-C795-436C-BE95-EC0146CC3FED")]

191 │ sysRoleMenu,

192 │

193 │ [EnumGuid("0468AE23-4C75-4420-99BF-86638B3254EC")]

194 │ sysRolePermission,

195 │

196 │ [EnumGuid("D93DE4BC-E887-4D4A-906E-70DF42B585E0")]

197 │ sysPermission,

198 │

199 │ [EnumGuid("FDEB130E-D677-40FB-850B-AB0A618B1A80")]

200 │ sysMenu,

201 │

202 │

203 │ [EnumGuid("2DEE3EF1-0E82-41EF-B76B-FD6D5AA21CB0")]

204 │ sysUser,

205 │

206 │ [EnumGuid("629B3B2D-A8F1-478C-9F4A-0B24FCAF7794")]

207 │ sysUserRole,

208 │

209 │ [EnumGuid("EC43ECAB-3F47-4C1F-85D5-32AECF61D384")]

210 │ sysLichSuDangNhap,

211 │

212 │ [EnumGuid("182ED01B-61A3-49A4-B8A4-738CD8D05D5F")]

213 │ sysNhomThongBao,

214 │

215 │ [EnumGuid("04581A0A-5F25-4AEB-8957-3CFA017F0B64")]

216 │ sysBranch,

217 │

218 │ [EnumGuid("06C9C5CF-51D9-4546-810B-2765893F9F6A")]

219 │ catNhaCC,

220 │

221 │ [EnumGuid("EB06196E-7841-4E0A-A25D-33CEA81D4EFA")]

222 │ catDanhMucKho,

223 │

224 │ [EnumGuid("9B1275CD-0727-43C1-A4CE-42540ED77AD6")]

225 │ catPhongBenh,

226 │

227 │ [EnumGuid("B0150345-01A5-4F4B-80C7-44B3212B7D78")]

228 │ catDonViTinh,

229 │

230 │ [EnumGuid("78C5B8C4-6A15-465F-93A9-6359F4CBB9EF")]

231 │ catLopHang,

232 │

233 │ [EnumGuid("3EBD756D-7C1A-4B00-B130-F3CD3E0332A8")]

234 │ catHangHoa,

235 │

236 │ [EnumGuid("EBF8CD92-6809-4267-9832-FCF9E6F066CD")]

237 │ catGiaBanTheoKho,

238 │

239 │ [EnumGuid("903702c7-ccab-40c7-8052-003603337666")]

240 │ catThueVat,

241 │

242 │ [EnumGuid("316AC987-F024-43F0-A219-33DA9E8F25A4")]

243 │ opsTonKho,

244 │

245 │ [EnumGuid("5D361012-1C7F-4397-B4FF-66DA331D237D")]

246 │ opsGiamGia,

247 │

248 │ [EnumGuid("73356206-2D85-4A29-A8C9-692A30DA586E")]

249 │ rptBangKeThuMuaKhongHoaDon,

250 │

251 │ [EnumGuid("9F4AE769-F78F-4B63-BF9C-7A6E2C0AB1EA")]

252 │ opsLichSuGia,

253 │

254 │ [EnumGuid("10E9CE06-31D8-4AC8-BCD9-A3B60BF2C1B3")]

255 │ opsKiemKe,

256 │

257 │ [EnumGuid("BDCB50A8-DCDA-483F-BFA8-1684D0CA2296")]

258 │ opsBartender,

259 │

260 │ [EnumGuid("C2284F81-D499-46C3-9269-B6079182CBC0")]

261 │ opsOrderTaiQuay,

262 │

263 │ [EnumGuid("33365D27-A731-4D4E-A5EC-C44304B01488")]

264 │ opsBaoCao,

265 │

266 │ [EnumGuid("2134A282-BB04-475E-B26D-C95B9E984299")]

267 │ opsNhapHang,

268 │

269 │ [EnumGuid("07051E71-2A85-4714-A507-D6A6B50B560E")]

270 │ vcKiemTraVouCher,

271 │

272 │ [EnumGuid("ADEC6D15-1E48-4870-BCFD-E4F34AC50041")]

273 │ opsXuatHang,

274 │

275 │ [EnumGuid("25C2F278-2A77-4E95-9834-FCAB8FB426E8")]

276 │ opsBanHang,

277 │

278 │ [EnumGuid("27ad8854-c138-413c-a345-fb653ae0ccc6")]

279 │ tblPhongBan,

280 │

281 │ [EnumGuid("F067B632-29B9-442B-864F-310D35204D6B")]

282 │ tblChucVu,

283 │

284 │ [EnumGuid("73356206-2D85-4A29-A8C9-692A30DA586E")]

285 │ opsBangKeThuMuaKhongHoaDon,

286 │

287 │ [EnumGuid("235D2F99-C6A9-4855-850F-C6288123689B")]

288 │ sysSetUpMayIn,

289 │

290 │ [EnumGuid("C982F290-44C9-4F3A-BE9E-2EE6CB539AA6")]

291 │ opsWebOrder,

292 │

293 │ [EnumGuid("07051E71-2A85-4714-A507-D6A6B50B560E")]

294 │ opsKiemTraVoucher,

295 │

296 │ [EnumGuid("3A5399E1-8D69-4831-B34F-40EDE532253C")]

297 │ vtthDinhMucDVKT,

298 │

299 │ [EnumGuid("67cceac9-c3a8-4166-b955-036630256ac4")]

300 │ vtthKhay,

301 │

302 │ [EnumGuid("b94d8cd9-9f33-432e-bc1c-ec05a2027e48")]

303 │ vtthNhapXuatKhay,

304 │

305 │ [EnumGuid("0d8ece37-d82d-4325-8a7d-a5d1b9e38c96")]

306 │ vtthNhapXuat,

307 │

308 │ [EnumGuid("18013196-d6fd-4e3a-9ce5-2edf810e0046")]

309 │ opsNhapKho,

310 │

311 │ [EnumGuid("b38390eb-343a-4876-b5b3-06feefd7ca86")]

312 │ catLoaiKho,

313 │

314 │ [EnumGuid("6c249ead-5134-4046-a61c-ed25969cc7f5")]

315 │ catVatTu,

316 │

317 │ [EnumGuid("67607CD8-AB9C-4886-A18A-608379F39EDD")]

318 │ rptBaoCaoVatTu,

319 │ }

320 │

321 │ public enum Status

322 │ {

323 │ Active = 1,

324 │ Inactive = 0

325 │ }

326 │ public enum LoaiGiaoDichTonKho

327 │ {

328 │ NhapNCC,

329 │ NhapChuyenKho,

330 │ XuatKho,

331 │ BanHang,

332 │ TraHang

333 │ }

334 │ public enum OrderStatus

335 │ {

336 │ Ordering = 0,

337 │ WaitConfirm = 1,

338 │ Confirmed = 2,

339 │ Ready = 3,

340 │ Paid = 4,

341 │ Done = 5,

342 │ Cancel = 6

343 │

344 │ }

345 │ public enum LoaiTieuDung

346 │ {

347 │ Khac = 0,

348 │ ThucAn = 1,

349 │ ThucUong = 2,

350 │ }

351 │ public enum ActionStatus

352 │ {

353 │ [EnumGuid("da241f20-4e82-4699-b490-9b076cb94ff9")]

354 │ Success = 1,

355 │ [EnumGuid("e538bd6a-cfef-4841-8cd2-cc97cacb930f")]

356 │ Error = 0,

357 │ [EnumGuid("b1bbead6-a20e-49f8-8c96-6fce8ad92b3d")]

358 │ Exist = -1,

359 │ [EnumGuid("7437465f-7e93-4f5f-ab07-48e4e8a26d85")]

360 │ NotExit = -2,

361 │ [EnumGuid("102b5b66-b367-4d25-8b16-b2dcd3e0d545")]

362 │ Disable = -3,

363 │ [EnumGuid("efb68b17-8bea-4e05-8e44-2cb82d463fed")]

364 │ NotMatch = -4,

365 │ [EnumGuid("91fd8594-7a1e-44bb-91cd-38e468545c81")]

366 │ Using = -5,

367 │ [EnumGuid("09cd51dd-3994-4022-a2d9-f277ea546fa2")]

368 │ TransactionError = -6,

369 │ }

370 │ public enum WebsiteUsing

371 │ {

372 │ all,

373 │ qlcv,

374 │ tmtt,

375 │ xncovid

376 │ }

377 │ public enum Branch

378 │ {

379 │ [EnumGuid("79d88348-7bb9-4278-884e-52de1607541a")]

380 │ TCT,

381 │

382 │ [EnumGuid("5750b1d3-68eb-42f0-814e-04987618b129")]

383 │ NhaMay1,

384 │

385 │ [EnumGuid("275d6ca7-ece5-43f4-816e-07d4c84d9c98")]

386 │ NhaMay2,

387 │

388 │ [EnumGuid("1d6def78-727f-4ea2-a09e-921d3af747a0")]

389 │ NhaMay3,

390 │

391 │ [EnumGuid("c918de3b-e93b-472a-885f-a0a13f9fd0b3")]

392 │ NhaMay4,

393 │

394 │ [EnumGuid("1780ed6c-97e3-45a7-a559-95ea4e6973d8")]

395 │ NhaMay5,

396 │

397 │ [EnumGuid("0bfce959-16fe-4f38-8b8d-3d9d56067f50")]

398 │ NhaMay6,

399 │

400 │ [EnumGuid("40ae8217-52be-4461-81c2-5ab4758bfe62")]

401 │ PhanPhoiHongHung,

402 │

403 │ [EnumGuid("9ef973ed-faa9-46bf-90aa-b7f21993c16d")]

404 │ PhanPhoiHungDuy,

405 │

406 │ [EnumGuid("0cc509ea-b071-4ddb-9c07-05ad4709dd30")]

407 │ KhoiKyThuat,

408 │

409 │ [EnumGuid("e0064746-ffa1-4aa4-9001-dc1601e759b4")]

410 │ KhoiDichVu,

411 │

412 │ [EnumGuid("123a3afd-db50-4ad6-a24a-5afb2d302e16")]

413 │ BVDaKhoaHongHung,

414 │

415 │ [EnumGuid("03dc8024-b69c-4b19-975c-3c550f70f140")]

416 │ NGOAI,

417 │ [EnumGuid("cc2c0680-952b-435d-828f-e380fc14cda5")]

418 │ VictoryHotel,

419 │ [EnumGuid("61798c88-5d5e-4675-8ad3-fde61da22c4f")]

420 │ TTGTCana,

421 │ [EnumGuid("7e1b3a07-70d7-48b5-ac78-ee7c8c11ee0f")]

422 │ CNCT,

423 │ }

424 │

425 │ public enum LoaiNhapXuatVTTH

426 │ {

427 │ Nhap = 3,

428 │ Xuat = 2,

429 │ Tra = 1,

430 │ }

431 │ public enum LoaiNhapKho

432 │ {

433 │ PNMN = 1,

434 │ PNCK = 2,

435 │ PXCK = 3,

436 │ PXTH = 4,

437 │ PXTR = 5,

438 │ PXKK = 6

439 │ }

440 │ public enum TrangThaiNhapKho

441 │ {

442 │ NhapKho = 1,

443 │ DaXacNhan = 2,

444 │ DaThanhToan = 3

445 │ }

446 │ public enum LoaiXuatKho

447 │ {

448 │ XuatHuHong = 1,

449 │ XuatHetDate = 2,

450 │ XuatKiemKe = 3,

451 │ XuatSuDung = 4,

452 │ XuatKhac = 5,

453 │ HoanKho = 6,

454 │ XuatDuTru = 7,

455 │ XuatThanhLy = 8,

456 │ XuatKyGui = 9,

457 │ HoanKhoNoiTru = 1009,

458 │ XuatTongHopToaNoiTru = 1010,

459 │ XuatBuTruc = 1011

460 │ }

461 │ public enum TrangThaiXuatKho

462 │ {

463 │ XuatKho = 1,

464 │ DaDuyet = 2

465 │ }

466 │ public enum TrangThaiDeXuatVTTH

467 │ {

468 │ LapPhieu = 1,

469 │ DaDuyet = 2,

470 │ DaXuat = 3,

471 │ DaNhan = 4,

472 │ HoanTat = 5

473 │ }

474 │ public enum KhoVatTu

475 │ {

476 │ KHOCHANDUOC = 1,

477 │

478 │ KHOCHANVATTU = 111,

479 │

480 │ KHOLE = 2,

481 │

482 │ KHONHATHUOCTRUNGTAM = 3,

483 │

484 │ KHONHATHUOCSO1 = 4,

485 │

486 │ KHONHATHUOCSO2 = 5,

487 │

488 │ TUTRUCGAYME = 1015,

489 │

490 │ TUTRUCPHONGMO = 39,

491 │

492 │ TUTRUCHOISUC = 99,

493 │

494 │ TUTRUCTIENME = 999,

495 │

496 │ KHOMAU = 1000,

497 │

498 │ KHOOXYHCQT = 1005,

499 │

500 │ TuTrucLCK\_RHM = 1006,

501 │

502 │ CDHATUTRUC = 1007,

503 │

504 │ KYGUI\_TTTM = 1008,

505 │

506 │ TUTRUC\_TTTM = 1009,

507 │

508 │ KYGUI\_TIEMCHUNG = 1010,

509 │

510 │ VTTHXN = 1011,

511 │

512 │ }

513 │

514 │ public enum NotificationChannel

515 │ {

516 │ Error

517 │ }

518 │

519 │

520 │ public enum KhoTieuHao

521 │ {

522 │ VTTHBEP = 52,

523 │ VTTHCF = 51,

524 │ VTTHCHTI = 48,

525 │ }

526 │ }

────────────────────────────────────────────────────────────

**📄 EnumExtend.cs (CS, 160 lines)**

1 │ ﻿using QuanLyBanHang.Common.Attributes;

2 │ using System.Reflection;

3 │ namespace QuanLyBanHang.Common

4 │ {

5 │ public static class EnumExtend

6 │ {

7 │ public static Guid GetEnumGuid(this Branch v) => GetEnumGuidV(v);

8 │ public static Guid GetEnumGuid(this MenuID v) => GetEnumGuidV(v);

9 │ public static Guid GetEnumGuid(this Permission v) => GetEnumGuidV(v);

10 │ public static Guid GetEnumGuid(this RoleID v) => GetEnumGuidV(v);

11 │ public static Guid GetEnumGuid(this ActionStatus v) => GetEnumGuidV(v);

12 │ private static Guid GetEnumGuidV(Enum e)

13 │ {

14 │ MemberInfo[] memInfo = e.GetType().GetMember(e.ToString());

15 │ if (memInfo != null && memInfo.Length > 0)

16 │ {

17 │ object[] attrs = memInfo[0].GetCustomAttributes(typeof(EnumGuidAttribute), false);

18 │ if (attrs != null && attrs.Length > 0)

19 │ return ((EnumGuidAttribute)attrs[0]).Guid;

20 │ }

21 │ throw new ArgumentException("Enum " + e.ToString() + " chưa được khởi tạo!");

22 │ }

23 │

24 │ }

25 │

26 │ public static class StatusExtend

27 │ {

28 │ #region Int

29 │ public static bool isActive(this int value)

30 │ {

31 │ return value == Status.Active.toInt();

32 │ }

33 │

34 │ public static bool isInactive(this int value)

35 │ {

36 │ return value == Status.Inactive.toInt();

37 │ }

38 │ #endregion

39 │ #region Status

40 │ public static bool isActive(this Status value)

41 │ {

42 │ return value == Status.Active;

43 │ }

44 │

45 │ public static bool isInactive(this Status value)

46 │ {

47 │ return value == Status.Inactive;

48 │ }

49 │ #endregion

50 │ }

51 │

52 │ public static class ActionStatusExtend

53 │ {

54 │ public static int toInt(this Status toInt)

55 │ {

56 │ return (int)toInt;

57 │ }

58 │ public static int toInt(this ActionStatus toInt)

59 │ {

60 │ return (int)toInt;

61 │ }

62 │

63 │ #region Int

64 │ public static bool isSuccess(this int value)

65 │ {

66 │ return value == ActionStatus.Success.toInt();

67 │ }

68 │ public static bool isError(this int value)

69 │ {

70 │ return value == ActionStatus.Error.toInt();

71 │ }

72 │ public static bool isDisable(this int value)

73 │ {

74 │ return value == ActionStatus.Disable.toInt();

75 │ }

76 │ public static bool isExist(this int value)

77 │ {

78 │ return value == ActionStatus.Exist.toInt();

79 │ }

80 │ public static bool isNotExit(this int value)

81 │ {

82 │ return value == ActionStatus.NotExit.toInt();

83 │ }

84 │ public static bool isNotMatch(this int value)

85 │ {

86 │ return value == ActionStatus.NotMatch.toInt();

87 │ }

88 │ public static bool isUsing(this int value)

89 │ {

90 │ return value == ActionStatus.Using.toInt();

91 │ }

92 │ public static bool isTransactionError(this int value)

93 │ {

94 │ return value == ActionStatus.TransactionError.toInt();

95 │ }

96 │ #endregion

97 │ #region ActionStatus

98 │ public static bool isSuccess(this ActionStatus value)

99 │ {

100 │ return value == ActionStatus.Success;

101 │ }

102 │ public static bool isError(this ActionStatus value)

103 │ {

104 │ return value == ActionStatus.Error;

105 │ }

106 │ public static bool isDisable(this ActionStatus value)

107 │ {

108 │ return value == ActionStatus.Disable;

109 │ }

110 │ public static bool isNotExit(this ActionStatus value)

111 │ {

112 │ return value == ActionStatus.NotExit;

113 │ }

114 │ public static bool isNotMatch(this ActionStatus value)

115 │ {

116 │ return value == ActionStatus.NotMatch;

117 │ }

118 │ public static bool isUsing(this ActionStatus value)

119 │ {

120 │ return value == ActionStatus.Using;

121 │ }

122 │ public static bool isTransactionError(this ActionStatus value)

123 │ {

124 │ return value == ActionStatus.TransactionError;

125 │ }

126 │ #endregion

127 │ #region GUID

128 │ public static bool isSuccess(this Guid value)

129 │ {

130 │ return value == ActionStatus.Success.GetEnumGuid();

131 │ }

132 │ public static bool isError(this Guid value)

133 │ {

134 │ return value == ActionStatus.Error.GetEnumGuid();

135 │ }

136 │ public static bool isDisable(this Guid value)

137 │ {

138 │ return value == ActionStatus.Disable.GetEnumGuid();

139 │ }

140 │ public static bool isNotExit(this Guid value)

141 │ {

142 │ return value == ActionStatus.NotExit.GetEnumGuid();

143 │ }

144 │ public static bool isNotMatch(this Guid value)

145 │ {

146 │ return value == ActionStatus.NotMatch.GetEnumGuid();

147 │ }

148 │ public static bool isUsing(this Guid value)

149 │ {

150 │ return value == ActionStatus.Using.GetEnumGuid();

151 │ }

152 │ public static bool isTransactionError(this Guid value)

153 │ {

154 │ return value == ActionStatus.TransactionError.GetEnumGuid();

155 │ }

156 │ #endregion

157 │ }

158 │

159 │ }

160 │

────────────────────────────────────────────────────────────

**📄 EnumGuidAttribute.cs (CS, 16 lines)**

1 │ ﻿namespace QuanLyBanHang.Common.Attributes

2 │ {

3 │ class EnumGuidAttribute : Attribute

4 │ {

5 │ public Guid Guid;

6 │

7 │ public EnumGuidAttribute(string guid)

8 │ {

9 │ Guid = new Guid(guid);

10 │ }

11 │ public EnumGuidAttribute()

12 │ {

13 │ Guid = Utility.defaultUID;

14 │ }

15 │ }

16 │ }

────────────────────────────────────────────────────────────

**📄 ErrorHandlerMiddleware.cs (CS, 79 lines)**

1 │ ﻿using HRMUngTuyen.Models.DTO;

2 │ using QuanLyBanHang.Services;

3 │ using System.Net;

4 │ using System.Text.Json;

5 │

6 │ namespace QuanLyBanHang.Common

7 │ {

8 │ public class ErrorHandlerMiddleware

9 │ {

10 │ private readonly RequestDelegate \_next;

11 │ private readonly ILogger<ErrorHandlerMiddleware> \_logger;

12 │ // THÊM VÀO: Dịch vụ để gửi thông báo

13 │ private readonly DiscordNotificationService \_notificationService;

14 │

15 │ public ErrorHandlerMiddleware(

16 │ RequestDelegate next,

17 │ ILogger<ErrorHandlerMiddleware> logger,

18 │ DiscordNotificationService notificationService)

19 │ {

20 │ \_next = next;

21 │ \_logger = logger;

22 │ \_notificationService = notificationService;

23 │ }

24 │

25 │ public async Task InvokeAsync(HttpContext context)

26 │ {

27 │ try

28 │ {

29 │ // Chuyển quyền điều khiển cho middleware tiếp theo trong pipeline

30 │ await \_next(context);

31 │ }

32 │ catch (Exception ex)

33 │ {

34 │ // Nếu có bất kỳ lỗi nào xảy ra, nó sẽ bị bắt lại ở đây.

35 │

36 │ // 1. Ghi log lỗi chi tiết bằng Serilog

37 │ \_logger.LogError(ex, "Một lỗi không mong muốn đã xảy ra: {Message}", ex.Message);

38 │

39 │ // 2. GỬI THÔNG BÁO LỖI QUA DISCORD

40 │ try

41 │ {

42 │ var embed = new DiscordEmbed

43 │ {

44 │ Title = "🚨 Lỗi Hệ Thống Tanione",

45 │ Color = 15158332, // Màu đỏ

46 │ Description = $"\*\*Đã xảy ra lỗi nghiêm trọng tại:\*\* `{context.Request.Path}`"

47 │ };

48 │ embed.Fields.Add(new DiscordEmbedField { Name = "Loại lỗi", Value = $"`{ex.GetType().Name}`", In...

49 │ embed.Fields.Add(new DiscordEmbedField { Name = "Thông báo", Value = ex.Message, Inline = false })...

50 │

51 │ // Không gửi stack trace đầy đủ qua Discord để tránh quá dài, chỉ cần thông tin ...

52 │ // Log chi tiết đã có trên server nhờ Serilog.

53 │

54 │ await \_notificationService.SendRichNotificationAsync(NotificationChannel.Error, embed);

55 │ }

56 │ catch (Exception notificationEx)

57 │ {

58 │ // Nếu việc gửi thông báo cũng bị lỗi, hãy ghi log lại

59 │ \_logger.LogError(notificationEx, "Không thể gửi thông báo lỗi đến Discord.");

60 │ }

61 │

62 │

63 │ // 3. Chuẩn bị một response lỗi thân thiện cho client

64 │ var response = context.Response;

65 │ response.ContentType = "application/json";

66 │ response.StatusCode = (int)HttpStatusCode.InternalServerError; // Lỗi 500

67 │

68 │ var errorResponse = new

69 │ {

70 │ message = "Hệ thống đã gặp phải một lỗi không mong muốn. Vui lòng thử lại sau....

71 │ statusCode = response.StatusCode

72 │ };

73 │

74 │ var jsonResponse = JsonSerializer.Serialize(errorResponse);

75 │ await response.WriteAsync(jsonResponse);

76 │ }

77 │ }

78 │ }

79 │ }

────────────────────────────────────────────────────────────

**📄 ErrorResponse.cs (CS, 11 lines)**

1 │ ﻿namespace QuanLyBanHang.Models.EInvoiceService.Viettel

2 │ {

3 │ public class ErrorResponse

4 │ {

5 │ public int Code { get; set; }

6 │ public string? Message { get; set; }

7 │ public string? Data { get; set; }

8 │ public string? ErrorCode { get; set; }

9 │ }

10 │

11 │ }

────────────────────────────────────────────────────────────

**📄 ErrorViewModel.cs (CS, 9 lines)**

1 │ namespace QuanLyBanHang.Models

2 │ {

3 │ public class ErrorViewModel

4 │ {

5 │ public string? RequestId { get; set; }

6 │

7 │ public bool ShowRequestId => !string.IsNullOrEmpty(RequestId);

8 │ }

9 │ }

────────────────────────────────────────────────────────────

**📄 ExportToExcel.cs (CS, 837 lines)**

1 │ ﻿using Microsoft.Extensions.Caching.Memory;

2 │ using Spire.Xls;

3 │

4 │ //using OfficeOpenXml;

5 │ //using OfficeOpenXml.Style;

6 │ using System.Data;

7 │ using System.Drawing;

8 │

9 │ namespace QuanLyBanHang.Common

10 │ {

11 │ public class ExportToExcel

12 │ {

13 │ private readonly MemoryCacheManager \_memoryCache;

14 │ public ExportToExcel(MemoryCacheManager memoryCache)

15 │ {

16 │ \_memoryCache = memoryCache;

17 │ }

18 │

19 │ public Stream Test(DataTable dt)

20 │ {

21 │ //var fileName = "Test.pdf";

22 │ //var contentType = "application/pdf";

23 │ Stream outStream = \_memoryCache.GetValueObject("WorkBook-" + 1, () =>

24 │ {

25 │ Stream outStream = new MemoryStream();

26 │ Workbook book = new Workbook();

27 │ Worksheet sheet = book.Worksheets[0];

28 │ sheet.InsertDataTable(dt, true, 1, 1);

29 │ book.ConverterSetting.SheetFitToPage = true;

30 │ //book.SaveToFile("C:\\Users\\PC\\Desktop\\insertTableToExcel.pdf", FileFormat.PDF);

31 │ book.SaveToStream(outStream, Spire.Xls.FileFormat.PDF);

32 │ outStream.Position = 0;

33 │ outStream.Seek(0, SeekOrigin.Begin);

34 │ return outStream;

35 │ }, new MemoryCacheEntryOptions().SetSlidingExpiration(TimeSpan.FromMinutes(5)));

36 │

37 │

38 │ return outStream;

39 │ // new FileStreamResult(outStream, contentType)

40 │ //{

41 │ // FileDownloadName = fileName

42 │ //};

43 │ }

44 │

45 │ public Stream TestPDF(DataTable dt, string type)

46 │ {

47 │ Stream outStream = new MemoryStream();

48 │ Workbook book = new Workbook();

49 │ Worksheet sheet = book.Worksheets[0];

50 │ sheet.InsertDataTable(dt, true, 1, 1);

51 │ sheet.PageSetup.PaperSize = PaperSizeType.PaperA4;

52 │ sheet.Range["E1:E10"].Merge();

53 │ sheet.Range["E1"].Value2 = "AAAAAAAA";

54 │ if (type == "Excel")

55 │ {

56 │ book.SaveToStream(outStream, Spire.Xls.FileFormat.Version2007);

57 │ }

58 │ else

59 │ {

60 │ book.SaveToStream(outStream, Spire.Xls.FileFormat.PDF);

61 │ }

62 │

63 │ outStream.Position = 0;

64 │ outStream.Seek(0, SeekOrigin.Begin);

65 │ return outStream;

66 │

67 │ }

68 │ public static Stream FromDataTable(DataTable dtDetail, string FileType, int startRow = 1, int startColumn = 1,

69 │ List<ColumnFormat>? ListHeaderFormat = null, List<ExcelFooter>? ListFooter = null, List<ColumnFormat>? ListC...

70 │ List<ExcelValue>? ListGiaTriRiengLe = null, bool FreezeTable = false, ePaperSize paperSize = ePaperSize.A4,

71 │ ePageOrientation pageOrientation = ePageOrientation.Portrait, PageSetup pageSetup = null)

72 │ {

73 │ Stream outStream = new MemoryStream();

74 │ Workbook book = new Workbook();

75 │ Worksheet sheet = book.Worksheets[0];

76 │ sheet.InsertDataTable(dtDetail, true, startRow, startColumn);

77 │ //sheet.Range[$"A{startRow + 1}:M{startRow + dtDetail.Rows.Count + 1}"].VerticalAlignment = VerticalAlignTyp...

78 │

79 │ for (int i = 0; i < dtDetail.Columns.Count; i++)

80 │ {

81 │ string cell = GetExcelColumnName(startColumn + i) + startRow.ToString();

82 │ ExcelFormat format = new ExcelFormat

83 │ {

84 │ HorizontalAlignment = eHorizontalAlignment.Center,

85 │ VerticalAlignment = eVerticalAlignment.Center,

86 │ WrapText = true,

87 │ IsBold = true

88 │ };

89 │ FormatRange(sheet, cell, format, book);

90 │ }

91 │

92 │ List<int> ColHeaderFormated = new List<int>();

93 │ if (ListHeaderFormat != null && ListHeaderFormat.Count > 0)

94 │ {

95 │ foreach (var item in ListHeaderFormat)

96 │ {

97 │ foreach (var c in item.ColumnIndex)

98 │ {

99 │ string cell = GetExcelColumnName(startColumn + c) + startRow.ToString();

100 │ FormatRange(sheet, cell, item.Format, book);

101 │ ColHeaderFormated.Add(c);

102 │ }

103 │ }

104 │ }

105 │

106 │ if (ListColumnFormat != null && ListColumnFormat.Count > 0)

107 │ {

108 │ foreach (var item in ListColumnFormat)

109 │ {

110 │ foreach (var c in item.ColumnIndex)

111 │ {

112 │ string col = GetExcelColumnName(c + startColumn);

113 │ string startcell = col + (startRow + 1).ToString();

114 │ string endcell = col + (startRow + 1 + dtDetail.Rows.Count).ToString();

115 │ string cell = startcell + ":" + endcell;

116 │ FormatRange(sheet, cell, item.Format, book);

117 │ }

118 │ }

119 │ }

120 │

121 │ int haveFooter = 0;

122 │ if (ListFooter != null && ListFooter.Count > 0)

123 │ {

124 │ haveFooter = 1;

125 │ foreach (var item in ListFooter)

126 │ {

127 │ foreach (var c in item.ColumnIndex)

128 │ {

129 │ string col = GetExcelColumnName(c + startColumn);

130 │ string cell = col + (startRow + 1 + dtDetail.Rows.Count).ToString();

131 │ if (item.IsFormula)

132 │ {

133 │ sheet.Range[cell].Formula = item.Text;

134 │ }

135 │ else

136 │ {

137 │ sheet.Range[cell].Value2 = item.Text;

138 │ }

139 │ if (item.Format == null)

140 │ {

141 │ item.Format = new ExcelFormat

142 │ {

143 │ IsBold = true

144 │ };

145 │ }

146 │ FormatRange(sheet, cell, item.Format, book);

147 │ }

148 │ }

149 │ }

150 │

151 │ var sc = GetExcelColumnName(startColumn);

152 │ var ec = GetExcelColumnName(startColumn + dtDetail.Columns.Count - 1);

153 │

154 │ var rang = sc + startRow.ToString() + ":" + ec + (startRow + dtDetail.Rows.Count + haveFooter).ToString();

155 │ if (rang.Split(":").Length == 2)

156 │ {

157 │ sheet.Range[rang].BorderInside(LineStyleType.Thin, System.Drawing.Color.Black);

158 │ }

159 │ sheet.Range[rang].BorderAround(LineStyleType.Thin, System.Drawing.Color.Black);

160 │ sheet.Range[rang].IsWrapText = true;

161 │ sheet.Range[rang].VerticalAlignment = VerticalAlignType.Center;

162 │ if (FreezeTable)

163 │ {

164 │ sheet.FreezePanes(startRow + 1, startColumn);

165 │ }

166 │ string startCol = GetExcelColumnName(startColumn);

167 │ var maxCol = startColumn + dtDetail.Columns.Count - 1;

168 │ var maxRow = startRow + dtDetail.Rows.Count + haveFooter;

169 │

170 │ if (ListGiaTriRiengLe != null && ListGiaTriRiengLe.Count > 0)

171 │ {

172 │ foreach (var item in ListGiaTriRiengLe)

173 │ {

174 │ string cell = item.Range.Split(':')[0];

175 │ if (!item.KhongGopO)

176 │ {

177 │ sheet.Range[item.Range].Merge();

178 │ }

179 │ if (item.IsCheckBox)

180 │ {

181 │ var address = GetAddress(item.Range);

182 │ Spire.Xls.Core.ICheckBox cb = sheet.CheckBoxes.AddCheckBox(address.Row, address.ColumnIdx, item....

183 │ cb.CheckState = item.CheckedState ? CheckState.Checked : CheckState.Unchecked;

184 │ cb.Left = item.CheckBoxLeft;

185 │ cb.Top = item.CheckBoxTop;

186 │ cb.Text = item.Value + "";

187 │ }

188 │ else if (!item.KhongDienGiaTri)

189 │ {

190 │ sheet.Range[cell].Value2 = item.Value;

191 │ }

192 │

193 │ FormatRange(sheet, item.Range, item.Format, book);

194 │ }

195 │ //Méo hiểu sao gom chung 1 for thì ko định dạng được nên phải tách ra như này chứ k...

196 │ foreach (var item in ListGiaTriRiengLe.Where(t => t.Format != null && (t.Format.Width > 0 || t.Format.He...

197 │ {

198 │ var address = GetAddress(item.Range, 1);

199 │ if (address.ColumnIdx > maxCol)

200 │ {

201 │ maxCol = GetColumnNumber(address.ColumnName);

202 │ }

203 │ if (address.Row > maxRow)

204 │ {

205 │ maxRow = address.Row;

206 │ }

207 │ if (item.Format != null)

208 │ {

209 │ if (item.Format.Width > 0)

210 │ {

211 │ sheet.SetColumnWidth(address.ColumnIdx, item.Format.Width);

212 │ }

213 │ if (item.Format.Height > 0)

214 │ {

215 │ sheet.SetRowHeight(address.Row, item.Format.Height);

216 │ }

217 │ }

218 │ }

219 │ }

220 │

221 │ string endCol = GetExcelColumnName(maxCol);

222 │ if (pageSetup != null && (pageSetup.PrintTitleColumns + "") != "AUTO")

223 │ {

224 │ sheet.PageSetup.PrintTitleColumns = pageSetup.PrintTitleColumns + "";

225 │ }

226 │ else

227 │ {

228 │ sheet.PageSetup.PrintTitleColumns = "$" + startCol + ":$" + endCol;

229 │ }

230 │

231 │ if (pageSetup != null && (pageSetup.PrintArea + "") != "AUTO")

232 │ {

233 │ sheet.PageSetup.PrintArea = pageSetup.PrintArea + "";

234 │ }

235 │ else

236 │ {

237 │ sheet.PageSetup.PrintArea = "A1:" + endCol + maxRow.ToString();

238 │ }

239 │

240 │ sheet.PageSetup.PaperSize = (PaperSizeType)paperSize;

241 │ sheet.PageSetup.Orientation = (PageOrientationType)pageOrientation;

242 │ sheet.PageSetup.FitToPagesWide = (pageSetup == null || pageSetup.FitToPagesWide == null) ? 1 : pageSetup.Fit...

243 │ sheet.PageSetup.FitToPagesTall = (pageSetup == null || pageSetup.FitToPagesTall == null) ? 0 : pageSetup.Fit...

244 │ sheet.PageSetup.LeftMargin = (pageSetup == null || pageSetup.LeftMargin == null) ? 0.25 : pageSetup.LeftMarg...

245 │ sheet.PageSetup.RightMargin = (pageSetup == null || pageSetup.RightMargin == null) ? 0.25 : pageSetup.RightM...

246 │ sheet.PageSetup.TopMargin = (pageSetup == null || pageSetup.TopMargin == null) ? 0.25 : pageSetup.TopMargin....

247 │ sheet.PageSetup.BottomMargin = (pageSetup == null || pageSetup.BottomMargin == null) ? 0.25 : pageSetup.Bott...

248 │ if (FileType == "Excel")

249 │ {

250 │ book.SaveToStream(outStream, Spire.Xls.FileFormat.Version2007);

251 │ }

252 │ else

253 │ {

254 │ book.SaveToStream(outStream, Spire.Xls.FileFormat.PDF);

255 │ }

256 │ outStream.Position = 0;

257 │ outStream.Seek(0, SeekOrigin.Begin);

258 │ return outStream;

259 │ }

260 │

261 │ public static (int ColumnIdx, string ColumnName, int Row) GetAddress(string range, int pos = 0)

262 │ {

263 │ if (range.IsEmpty())

264 │ {

265 │ return (-1, "", -1);

266 │ }

267 │ string numbers = "0123456789";

268 │ var tmp = range.Split(":");

269 │ var cell = tmp.Length == 2 ? tmp[pos] : tmp[0];

270 │ int idx = 0;

271 │ for (int i = 0; i < cell.Length; i++)

272 │ {

273 │ if (numbers.IndexOf(cell[i]) >= 0)

274 │ {

275 │ idx = i;

276 │ break;

277 │ }

278 │ }

279 │ return (GetColumnNumber(cell.Substring(0, idx)), cell.Substring(0, idx), cell.Substring(idx).ToInt());

280 │ }

281 │

282 │ public static void FormatRange(Worksheet sheet, string range, ExcelFormat format, Workbook wb)

283 │ {

284 │ if (format == null) return;

285 │ if (!format.BackgroundColor.IsEmpty())

286 │ {

287 │ sheet.Range[range].Style.Color = ColorTranslator.FromHtml(format.BackgroundColor);

288 │ }

289 │ if (!format.ForeColor.IsEmpty())

290 │ {

291 │ sheet.Range[range].Style.Font.Color = ColorTranslator.FromHtml(format.ForeColor);

292 │ }

293 │

294 │ if (format.Font != null)

295 │ {

296 │ sheet.Range[range].Style.Font.FontName = format.Font.Name;

297 │ sheet.Range[range].Style.Font.Size = format.Font.Size;

298 │ }

299 │ sheet.Range[range].Style.Font.IsBold = format.IsBold;

300 │ sheet.Range[range].Style.Font.IsItalic = format.IsItalic;

301 │ if (!format.NumberFormat.IsEmpty())

302 │ {

303 │ sheet.Range[range].NumberFormat = format.NumberFormat;

304 │ }

305 │ sheet.Range[range].IsWrapText = format.WrapText;

306 │ sheet.Range[range].Style.HorizontalAlignment = (HorizontalAlignType)format.HorizontalAlignment;

307 │ sheet.Range[range].Style.VerticalAlignment = (VerticalAlignType)format.VerticalAlignment;

308 │ var address = GetAddress(range);

309 │ if (format.Width > 0)

310 │ {

311 │ //sheet.Range[range].ColumnWidth = format.Width;

312 │ sheet.SetColumnWidth(address.ColumnIdx, format.Width);

313 │ }

314 │ if (format.Height > 0)

315 │ {

316 │ //sheet.Range[range].RowHeight = format.Height;

317 │ sheet.SetRowHeight(address.Row, format.Height);

318 │ }

319 │ if (format.Border)

320 │ {

321 │ if (range.Split(":").Length == 2)

322 │ {

323 │ sheet.Range[range].BorderInside(LineStyleType.Thin, System.Drawing.Color.Black);

324 │ }

325 │ sheet.Range[range].BorderAround(LineStyleType.Thin, System.Drawing.Color.Black);

326 │ }

327 │ if (format.ListMultiFormatCell != null && format.ListMultiFormatCell.Count > 0)

328 │ {

329 │ RichText rtx = sheet.Range[range].RichText;

330 │ foreach (var item in format.ListMultiFormatCell)

331 │ {

332 │ try

333 │ {

334 │ ExcelFont f = wb.CreateFont();

335 │ f.IsBold = item.IsBold;

336 │ f.IsItalic = item.IsItalic;

337 │ f.FontName = item.Font.FontFamily.Name;

338 │ f.Size = item.Font.Size.ToDouble();

339 │ if (!item.ForeColor.IsEmpty())

340 │ {

341 │ f.Color = ColorTranslator.FromHtml(item.ForeColor);

342 │ }

343 │ if (item.To > item.From)

344 │ {

345 │ rtx.SetFont(item.From, item.To, f);

346 │ }

347 │ }

348 │ catch { }

349 │ }

350 │ }

351 │ }

352 │ public static string GetExcelColumnName(int columnNumber)

353 │ {

354 │ string columnName = "";

355 │

356 │ while (columnNumber > 0)

357 │ {

358 │ int modulo = (columnNumber - 1) % 26;

359 │ columnName = Convert.ToChar('A' + modulo) + columnName;

360 │ columnNumber = (columnNumber - modulo) / 26;

361 │ }

362 │

363 │ return columnName;

364 │ }

365 │ public static int GetColumnNumber(string name)

366 │ {

367 │ int number = 0;

368 │ int pow = 1;

369 │ for (int i = name.Length - 1; i >= 0; i--)

370 │ {

371 │ number += (name[i] - 'A' + 1) \* pow;

372 │ pow \*= 26;

373 │ }

374 │

375 │ return number;

376 │ }

377 │ public static Stream FromTemplate(string fileName, string FileType, List<ExcelValue> ListGiaTriRiengLe, DataTabl...

378 │ {

379 │

380 │ Stream outStream = new MemoryStream();

381 │ Workbook book = new Workbook();

382 │ book.LoadFromFile(fileName);

383 │ Worksheet sheet = book.Worksheets[0];

384 │

385 │

386 │

387 │ //Điền giá trị riêng lẻ

388 │ FormatGiaTriRiengLe(book, sheet, ListGiaTriRiengLe);

389 │

390 │ if (TableGiaTriBang.Rows.Count > 0)

391 │ {

392 │ sheet.InsertRow(DongBatDau + 1, TableGiaTriBang.Rows.Count);

393 │ }

394 │ sheet.InsertDataTable(TableGiaTriBang, false, DongBatDau + 1, GetColumnNumber(CotBatDau));

395 │ int cotbatdau = GetColumnNumber(CotBatDau);

396 │ int cotketthuc = GetColumnNumber(CotKetThuc);

397 │ for (int c = cotbatdau; c <= cotketthuc; c++)

398 │ {

399 │ // Lấy 2 vùng (range) cần sao chép định dạng

400 │ CellRange sourceRange = sheet.Range[string.Format("{0}{1}", GetExcelColumnName(c), DongBatDau)];

401 │ int idxCotDau = DongBatDau + (TableGiaTriBang.Rows.Count == 0 ? 0 : 1);

402 │ CellRange targetRange = sheet.Range[string.Format("{0}{1}:{2}{3}", GetExcelColumnName(c), idxCotDau, Get...

403 │

404 │ // Sao chép font

405 │ targetRange.Style.Font.FontName = sourceRange.Style.Font.FontName;

406 │ targetRange.Style.Font.Size = sourceRange.Style.Font.Size;

407 │ targetRange.Style.Font.IsBold = sourceRange.Style.Font.IsBold;

408 │ targetRange.Style.Font.IsItalic = sourceRange.Style.Font.IsItalic;

409 │ targetRange.Style.Font.Underline = sourceRange.Style.Font.Underline; // Sao chép gạch chân

410 │ targetRange.Style.Font.Color = sourceRange.Style.Font.Color;

411 │

412 │ // Sao chép căn lề

413 │ targetRange.Style.HorizontalAlignment = sourceRange.Style.HorizontalAlignment; // Căn lề ngang

414 │ targetRange.Style.VerticalAlignment = sourceRange.Style.VerticalAlignment; // Căn lề dọc

415 │

416 │ // Sao chép định dạng số

417 │ targetRange.Style.NumberFormat = sourceRange.Style.NumberFormat;

418 │

419 │ // Sao chép nền

420 │ targetRange.Style.Color = sourceRange.Style.Color;

421 │

422 │ // Sao chép viền

423 │ targetRange.Borders[BordersLineType.EdgeTop].LineStyle = sourceRange.Borders[BordersLineType.EdgeTop].Li...

424 │ targetRange.Borders[BordersLineType.EdgeBottom].LineStyle = sourceRange.Borders[BordersLineType.EdgeBott...

425 │ targetRange.Borders[BordersLineType.EdgeLeft].LineStyle = sourceRange.Borders[BordersLineType.EdgeLeft]....

426 │ targetRange.Borders[BordersLineType.EdgeRight].LineStyle = sourceRange.Borders[BordersLineType.EdgeRight...

427 │

428 │ // Sao chép độ co giãn (wrap text)

429 │ targetRange.Style.WrapText = sourceRange.Style.WrapText;

430 │

431 │ }

432 │ //Điền giá trị riêng lẻ 2

433 │ FormatGiaTriRiengLe(book, sheet, ListGiaTriRiengLe2);

434 │

435 │ AdjustRowHeights(sheet, ListGiaTriRiengLe);

436 │ //AdjustRowHeights(sheet, ListGiaTriRiengLe2);

437 │

438 │ //Xóa dòng định dạng

439 │ if (XoaDongTemplate)

440 │ {

441 │ sheet.DeleteRow(DongBatDau);

442 │ }

443 │

444 │ if (FileType == "Excel")

445 │ {

446 │ book.SaveToStream(outStream, Spire.Xls.FileFormat.Version2007);

447 │ }

448 │ else

449 │ {

450 │

451 │ // Tự động điều chỉnh kích thước

452 │ book.SaveToStream(outStream, Spire.Xls.FileFormat.PDF);

453 │ }

454 │ outStream.Position = 0;

455 │ outStream.Seek(0, SeekOrigin.Begin);

456 │ return outStream;

457 │ }

458 │

459 │ public static Stream FromTemplateBangKe(string fileName, string FileType, List<ExcelValue> ListGiaTriRiengLe, Da...

460 │ {

461 │

462 │ Stream outStream = new MemoryStream();

463 │ Workbook book = new Workbook();

464 │ book.LoadFromFile(fileName);

465 │ Worksheet sheet = book.Worksheets[0];

466 │

467 │

468 │

469 │ //Điền giá trị riêng lẻ

470 │ FormatGiaTriRiengLe(book, sheet, ListGiaTriRiengLe);

471 │

472 │ if (TableGiaTriBang.Rows.Count > 0)

473 │ {

474 │ sheet.InsertRow(DongBatDau + 1, TableGiaTriBang.Rows.Count);

475 │ }

476 │ sheet.InsertDataTable(TableGiaTriBang, false, DongBatDau + 1, GetColumnNumber(CotBatDau));

477 │ int cotbatdau = GetColumnNumber(CotBatDau);

478 │ int cotketthuc = GetColumnNumber(CotKetThuc);

479 │ for (int c = cotbatdau; c <= cotketthuc; c++)

480 │ {

481 │ // Lấy 2 vùng (range) cần sao chép định dạng

482 │ CellRange sourceRange = sheet.Range[string.Format("{0}{1}", GetExcelColumnName(c), DongBatDau)];

483 │ int idxCotDau = DongBatDau + (TableGiaTriBang.Rows.Count == 0 ? 0 : 1);

484 │ CellRange targetRange = sheet.Range[string.Format("{0}{1}:{2}{3}", GetExcelColumnName(c), idxCotDau, Get...

485 │

486 │ // Sao chép font

487 │ targetRange.Style.Font.FontName = sourceRange.Style.Font.FontName;

488 │ targetRange.Style.Font.Size = sourceRange.Style.Font.Size;

489 │ targetRange.Style.Font.IsBold = sourceRange.Style.Font.IsBold;

490 │ targetRange.Style.Font.IsItalic = sourceRange.Style.Font.IsItalic;

491 │ targetRange.Style.Font.Underline = sourceRange.Style.Font.Underline; // Sao chép gạch chân

492 │ targetRange.Style.Font.Color = sourceRange.Style.Font.Color;

493 │

494 │ // Sao chép căn lề

495 │ targetRange.Style.HorizontalAlignment = sourceRange.Style.HorizontalAlignment; // Căn lề ngang

496 │ targetRange.Style.VerticalAlignment = sourceRange.Style.VerticalAlignment; // Căn lề dọc

497 │

498 │ // Sao chép định dạng số

499 │ targetRange.Style.NumberFormat = sourceRange.Style.NumberFormat;

500 │

501 │ // Sao chép nền

502 │ targetRange.Style.Color = sourceRange.Style.Color;

503 │

504 │ // Sao chép viền

505 │ targetRange.Borders[BordersLineType.EdgeTop].LineStyle = sourceRange.Borders[BordersLineType.EdgeTop].Li...

506 │ targetRange.Borders[BordersLineType.EdgeBottom].LineStyle = sourceRange.Borders[BordersLineType.EdgeBott...

507 │ targetRange.Borders[BordersLineType.EdgeLeft].LineStyle = sourceRange.Borders[BordersLineType.EdgeLeft]....

508 │ targetRange.Borders[BordersLineType.EdgeRight].LineStyle = sourceRange.Borders[BordersLineType.EdgeRight...

509 │

510 │ // Sao chép độ co giãn (wrap text)

511 │ targetRange.Style.WrapText = sourceRange.Style.WrapText;

512 │

513 │ }

514 │ //Điền giá trị riêng lẻ 2

515 │ FormatGiaTriRiengLe(book, sheet, ListGiaTriRiengLe2);

516 │

517 │ AdjustRowHeights(sheet, ListGiaTriRiengLe);

518 │ AdjustRowHeights(sheet, ListGiaTriRiengLe2);

519 │

520 │ //Xóa dòng định dạng

521 │ if (XoaDongTemplate)

522 │ {

523 │ sheet.DeleteRow(DongBatDau);

524 │ }

525 │

526 │ if (FileType == "Excel")

527 │ {

528 │ book.SaveToStream(outStream, Spire.Xls.FileFormat.Version2007);

529 │ }

530 │ else

531 │ {

532 │

533 │ // Thiết lập hướng trang ngang

534 │ sheet.PageSetup.Orientation = PageOrientationType.Landscape;

535 │ sheet.PageSetup.Zoom = 85;

536 │

537 │ // Tự động điều chỉnh kích thước

538 │ book.SaveToStream(outStream, Spire.Xls.FileFormat.PDF);

539 │ }

540 │ outStream.Position = 0;

541 │ outStream.Seek(0, SeekOrigin.Begin);

542 │ return outStream;

543 │ }

544 │ static void FormatGiaTriRiengLe(Workbook book, Worksheet sheet, List<ExcelValue> ListGiaTriRiengLe)

545 │ {

546 │ if (ListGiaTriRiengLe != null)

547 │ {

548 │ foreach (ExcelValue item in ListGiaTriRiengLe)

549 │ {

550 │ if (item.IsImage)

551 │ {

552 │ if (!item.ImagePath.IsEmpty())

553 │ {

554 │ ExcelPicture picture = sheet.Pictures.Add(item.ImageY, item.ImageX, item.ImagePath);

555 │ picture.Width = item.WidthImage; // Chiều rộng tính bằng điểm ảnh

556 │ picture.Height = item.HeightImage; // Chiều cao tính bằng điểm ảnh

557 │ picture.Left = picture.Left + item.LeftImage;

558 │ picture.Top = picture.Top + item.TopImage;

559 │ }

560 │ }

561 │ else if (item.IsCheckBox)

562 │ {

563 │ var idxchk = 0;

564 │ if (item.RightToLeft)

565 │ {

566 │ idxchk = (item.Value + "").Length + 1;

567 │ }

568 │ var spacing = "";

569 │ if (item.CheckBoxSpacing > 0)

570 │ {

571 │ for (int i = 0; i < item.CheckBoxSpacing; i++)

572 │ {

573 │ spacing += " ";

574 │ }

575 │ }

576 │ char chkbox = item.CheckedState ? (char)254 : (char)168;

577 │ sheet.Range[item.Range].Text = item.RightToLeft ? (item.Value + spacing + chkbox) : (chkbox + sp...

578 │ RichText rtx = sheet.Range[item.Range].RichText;

579 │ try

580 │ {

581 │ ExcelFont f = book.CreateFont();

582 │ f.FontName = "Wingdings";

583 │ f.Size = item.Format != null && item.Format.Font != null ? item.Format.Font.Size : 12;

584 │ rtx.SetFont(idxchk, idxchk + 1, f);

585 │ }

586 │ catch { }

587 │

588 │ }

589 │ else

590 │ {

591 │ FormatRange(sheet, item.Range, item.Format, book);

592 │ if (!item.KhongGopO)

593 │ {

594 │ sheet.Range[item.Range].Merge();

595 │ }

596 │ if (!item.KhongDienGiaTri)

597 │ {

598 │ if (item.IsFormula)

599 │ {

600 │ sheet.Range[item.Range].Formula = item.Value.ToString();

601 │ }

602 │ else

603 │ {

604 │ sheet.Range[item.Range].Value2 = item.Value;

605 │ }

606 │ }

607 │ }

608 │ }

609 │ }

610 │ }

611 │ private static void AdjustRowHeights(Worksheet sheet, List<ExcelValue> values)

612 │ {

613 │ foreach (var item in values)

614 │ {

615 │ if (item.Format != null && item.Format.Height > 0)

616 │ {

617 │ try

618 │ {

619 │ CellRange range = sheet.Range[item.Range];

620 │ int rowIndex = range.Row;

621 │ sheet.SetRowHeight(rowIndex, item.Format.Height);

622 │ }

623 │ catch (Exception ex)

624 │ {

625 │ System.Diagnostics.Debug.WriteLine($"Error setting row height: {ex.Message}");

626 │ }

627 │ }

628 │ }

629 │ }

630 │ static void AdjustRowHeight(Worksheet sheet, int rowIndex)

631 │ {

632 │ // Điều chỉnh chiều cao hàng thủ công bằng cách đo nội dung của các ô

633 │ foreach (CellRange range in sheet.Rows[rowIndex].CellList)

634 │ {

635 │ if (range.Style.WrapText && range.Text != null && range.Text.Length > 0)

636 │ {

637 │ float fontSize = (float)range.Style.Font.Size;

638 │ FontFamily ff = new FontFamily(range.Style.Font.FontName);

639 │ Font font = new Font(ff, fontSize);

640 │ SizeF size = range.MeasureString(range.Text);

641 │ double cellWidth = 0;

642 │

643 │ CellRange mergeArea = range.MergeArea;

644 │ if (mergeArea != null && (mergeArea.Row != range.Row || mergeArea.Column != range.Column || mergeAre...

645 │ {

646 │ for (int col = mergeArea.Column; col < mergeArea.Column + mergeArea.Columns.Length; col++)

647 │ {

648 │ cellWidth += sheet.Columns[col - 1].ColumnWidth;

649 │ }

650 │ }

651 │ else

652 │ {

653 │ cellWidth += sheet.Columns[range.Column - 1].ColumnWidth;

654 │ }

655 │ cellWidth = cellWidth == 0 ? 8.43 : cellWidth;//8.43 là chiều rộng mặc định của 1 cột ...

656 │ double textWidth = 0;

657 │ using (Graphics g = Graphics.FromImage(new Bitmap(1, 1)))

658 │ {

659 │ SizeF textSize = g.MeasureString(range.Text, font);

660 │ // Điều chỉnh chiều cao hàng dựa trên chiều cao của văn bản

661 │ textWidth = textSize.Width / 0.75; // Chuyển đổi từ điểm sang pixel

662 │ }

663 │ int lines = (int)Math.Ceiling((double)(textWidth / cellWidth) / 9.4);

664 │ lines = lines == 0 ? 1 : lines;

665 │ double newHeight = lines \* size.Height;

666 │ if (newHeight > sheet.Rows[rowIndex].RowHeight)

667 │ {

668 │ sheet.Rows[rowIndex].RowHeight = lines \* size.Height;

669 │ }

670 │ }

671 │ }

672 │ }

673 │

674 │ }

675 │

676 │ public class ExcelValue

677 │ {

678 │ /// <summary>

679 │ /// Giá trị của ô

680 │ /// </summary>

681 │ public object Value { get; set; }

682 │ /// <summary>

683 │ /// Vùng dữ liệu

684 │ /// </summary>

685 │ public string Range { get; set; }

686 │ /// <summary>

687 │ /// Chỉ format ô không điền giá trị vào ô

688 │ /// </summary>

689 │ public bool KhongDienGiaTri { get; set; }

690 │ /// <summary>

691 │ /// Xác định có muốn gộp ô trong Range lại không

692 │ /// </summary>

693 │ public bool KhongGopO { get; set; } = false;

694 │ public ExcelFormat Format { get; set; }

695 │ /// <summary>

696 │ /// Giá trị là checkbox

697 │ /// </summary>

698 │ public bool IsCheckBox { get; set; } = false;

699 │ /// <summary>

700 │ /// Trạng thái ô Checkbox True, False

701 │ /// </summary>

702 │ public bool CheckedState { get; set; } = false;

703 │ /// <summary>

704 │ /// False: ô check nằm bên trái chữ, True: ô check nằm bên phải chữ

705 │ /// </summary>

706 │ public bool RightToLeft { get; set; } = false;

707 │

708 │ /// <summary>

709 │ /// Khoảng trống giữa chữ và checkbox

710 │ /// </summary>

711 │ public uint CheckBoxSpacing { get; set; } = 1;

712 │ /// <summary>

713 │ /// Giá trị là công thức excel bắt đầu bằng dấu =

714 │ /// </summary>

715 │ public bool IsFormula { get; set; } = false;

716 │ /// <summary>

717 │ /// Giá trị là hình ảnh

718 │ /// </summary>

719 │ public bool IsImage { get; set; } = false;

720 │ /// <summary>

721 │ /// ĐƯờng dẫn hình ảnh

722 │ /// </summary>

723 │ public string ImagePath { get; set; } = "";

724 │

725 │ /// <summary>

726 │ /// cột chèn hình

727 │ /// </summary>

728 │ public int ImageX { get; set; }

729 │

730 │ /// <summary>

731 │ /// Dòng chèn hình

732 │ /// </summary>

733 │ public int ImageY { get; set; }

734 │ /// <summary>

735 │ /// Chiều ngang hình

736 │ /// </summary>

737 │ public int WidthImage { get; set; }

738 │ /// <summary>

739 │ /// Chiều cao hình

740 │ /// </summary>

741 │ public int HeightImage { get; set; }

742 │

743 │ /// <summary>

744 │ /// Di chuyển hình qua trái, phải xíu

745 │ /// </summary>

746 │ public int LeftImage { get; set; } = 0;

747 │ /// <summary>

748 │ /// Di chuyển hình lên, xuống xíu

749 │ /// </summary>

750 │ public int TopImage { get; set; } = 0;

751 │

752 │ public int CheckBoxHeight { get; set; } = 15;

753 │ public int CheckBoxWidth { get; set; } = 100;

754 │ public int CheckBoxLeft { get; set; } = 0;

755 │ public int CheckBoxTop { get; set; } = 0;

756 │ }

757 │ public class ExcelFormat

758 │ {

759 │ public string NumberFormat { get; set; }

760 │ public Font Font { get; set; }

761 │ public string BackgroundColor { get; set; }

762 │ public string ForeColor { get; set; }

763 │ public bool WrapText { get; set; }

764 │ public bool IsBold { get; set; }

765 │ public bool IsItalic { get; set; }

766 │ public double Width { get; set; }

767 │ public double Height { get; set; }

768 │ public bool Border { get; set; }

769 │ public eHorizontalAlignment HorizontalAlignment { get; set; }

770 │ public eVerticalAlignment VerticalAlignment { get; set; }

771 │ public List<MultiFormatCell> ListMultiFormatCell { get; set; }

772 │ }

773 │ public class ColumnFormat

774 │ {

775 │ public List<int> ColumnIndex { get; set; }

776 │ public ExcelFormat Format { get; set; }

777 │ }

778 │ public class ExcelFooter

779 │ {

780 │ public List<int> ColumnIndex { get; set; }

781 │ public string Text { get; set; }

782 │ public bool IsFormula { get; set; }

783 │ public ExcelFormat Format { get; set; }

784 │ }

785 │

786 │ public class MultiFormatCell

787 │ {

788 │ public int From { get; set; }

789 │ public int To { get; set; }

790 │ public Font Font { get; set; }

791 │ public string ForeColor { get; set; }

792 │ public bool IsBold { get; set; }

793 │ public bool IsItalic { get; set; }

794 │ }

795 │

796 │ public class PageSetup

797 │ {

798 │ public string? PrintTitleColumns { get; set; }

799 │ public string? PrintArea { get; set; }

800 │ public int? FitToPagesWide { get; set; }

801 │ public int? FitToPagesTall { get; set; }

802 │ public double? LeftMargin { get; set; }

803 │ public double? RightMargin { get; set; }

804 │ public double? TopMargin { get; set; }

805 │ public double? BottomMargin { get; set; }

806 │ }

807 │

808 │ public enum ePaperSize

809 │ {

810 │ A4 = PaperSizeType.PaperA4,

811 │ A5 = PaperSizeType.PaperA5

812 │ }

813 │ public enum ePageOrientation

814 │ {

815 │ Portrait = PageOrientationType.Portrait,

816 │ Landscape = PageOrientationType.Landscape,

817 │ }

818 │ public enum eHorizontalAlignment

819 │ {

820 │ General = HorizontalAlignType.General,

821 │ Left = HorizontalAlignType.Left,

822 │ Center = HorizontalAlignType.Center,

823 │ Right = HorizontalAlignType.Right,

824 │ Fill = HorizontalAlignType.Fill,

825 │ Justify = HorizontalAlignType.Justify,

826 │ CenterAcrossSelection = HorizontalAlignType.CenterAcrossSelection,

827 │ Distributed = HorizontalAlignType.Distributed

828 │ }

829 │ public enum eVerticalAlignment

830 │ {

831 │ Top = VerticalAlignType.Top,

832 │ Center = VerticalAlignType.Center,

833 │ Bottom = VerticalAlignType.Bottom,

834 │ Justify = VerticalAlignType.Justify,

835 │ Distributed = VerticalAlignType.Distributed

836 │ }

837 │ }

────────────────────────────────────────────────────────────

**📄 Extension.cs (CS, 207 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc.ViewFeatures;

2 │ using Microsoft.Extensions.Caching.Memory;

3 │ using Newtonsoft.Json;

4 │ using System.Text;

5 │ using System.Xml;

6 │ using System.Xml.Serialization;

7 │

8 │ namespace QuanLyBanHang.Common

9 │ {

10 │ public static class Exten

11 │ {

12 │ public static string Decrypt(this string toDecrypt)

13 │ {

14 │ return Utility.Decrypt(toDecrypt);

15 │ }

16 │ public static string Encrypt(this string toEncrypt)

17 │ {

18 │ return Utility.Encrypt(toEncrypt);

19 │ }

20 │ public static string Decrypt(this int toDecrypt)

21 │ {

22 │ return Utility.Decrypt(toDecrypt + "");

23 │ }

24 │ public static string Encrypt(this int toEncrypt)

25 │ {

26 │ return Utility.Encrypt(toEncrypt + "");

27 │ }

28 │

29 │ public static bool IsEmpty(this string? str)

30 │ {

31 │ return str == null || str == "";

32 │ }

33 │

34 │ public static Guid? GetGuidNull(this string? str)

35 │ {

36 │ try

37 │ {

38 │ return str.IsEmpty() ? null : Guid.Parse(str);

39 │ }

40 │ catch

41 │ {

42 │ return Utility.defaultUID;

43 │ }

44 │ }

45 │ public static Guid GetGuid(this string? str)

46 │ {

47 │ try

48 │ {

49 │ return Guid.Parse(str);

50 │ }

51 │ catch

52 │ {

53 │ return Utility.defaultUID;

54 │ }

55 │ }

56 │ public static bool IsNullOrEmpty(this Guid? str)

57 │ {

58 │ try

59 │ {

60 │ return str == null || str == Utility.defaultUID;

61 │ }

62 │ catch

63 │ {

64 │ return true;

65 │ }

66 │ }

67 │ public static bool IsEmpty(this Guid str)

68 │ {

69 │ try

70 │ {

71 │ return str == Utility.defaultUID;

72 │ }

73 │ catch

74 │ {

75 │ return true;

76 │ }

77 │ }

78 │ public static bool IsNotEmpty(this Guid? str)

79 │ {

80 │ return !str.IsNullOrEmpty();

81 │ }

82 │ public static bool IsNotEmpty(this Guid str)

83 │ {

84 │ return !str.IsEmpty();

85 │ }

86 │ public static Guid DefaultIfNull(this Guid? str)

87 │ {

88 │ return str ?? Utility.defaultUID;

89 │ }

90 │ public static DateTime ParseDate(this string? str)

91 │ {

92 │ try

93 │ {

94 │ return DateTime.ParseExact(str, str.Contains("-") ? "yyyy-MM-dd" : "dd/MM/yyyy", null);

95 │ }

96 │ catch

97 │ {

98 │ return DateTime.Now;

99 │ }

100 │ }

101 │ public delegate T CallBack<T>();

102 │ public delegate void CallBackData<T>(List<T> b);

103 │ public static T? DeepCopy<T>(this T self)

104 │ {

105 │ var serialized = JsonConvert.SerializeObject(self);

106 │ return JsonConvert.DeserializeObject<T>(serialized);

107 │ }

108 │ public class Utf8StringWriter : StringWriter

109 │ {

110 │ public override Encoding Encoding => Encoding.UTF8;

111 │ }

112 │ public static string SerializeChild<T>(this T value)

113 │ {

114 │ return Serialize(value, new XmlSerializerNamespaces(new[] { XmlQualifiedName.Empty }));

115 │ }

116 │ public static string Serialize<T>(this T value, XmlSerializerNamespaces? namespaces = null)

117 │ {

118 │ if (value == null) return string.Empty;

119 │ var xmlSerializer = new XmlSerializer(typeof(T));

120 │ using (var stringWriter = new Utf8StringWriter())

121 │ {

122 │ using (var xmlWriter = XmlWriter.Create(stringWriter, new XmlWriterSettings { Indent = true, OmitXmlDecl...

123 │ {

124 │ xmlSerializer.Serialize(xmlWriter, value, namespaces);

125 │ return stringWriter.ToString();

126 │ }

127 │ }

128 │ }

129 │ public static bool checkRole(this ViewDataDictionary<dynamic> viewData, RoleID role)

130 │ {

131 │ return Utility.CheckRole(viewData, role.GetEnumGuid());

132 │ }

133 │ public static bool checkPermis(this ViewDataDictionary<dynamic> viewData, Permission permission)

134 │ {

135 │ return Utility.checkTokenPermission(viewData, permission.GetEnumGuid());

136 │ }

137 │ public static T getViewData<T>(this ViewDataDictionary<dynamic> viewData, string key)

138 │ {

139 │ return Utility.getViewData<T>(viewData, key);

140 │ }

141 │

142 │ private static MemoryCacheEntryOptions cacheEntryOptions = new MemoryCacheEntryOptions().SetSlidingExpiration(Ti...

143 │ private static List<string> \_key = new List<string>();

144 │ public static T GetValueObject<T>(this IMemoryCache \_memoryCache, string key, CallBack<T> callBack, MemoryCacheE...

145 │ {

146 │ if (!\_memoryCache.TryGetValue(key, out T? obj))

147 │ {

148 │ obj = callBack();

149 │ \_memoryCache.Set(key, obj, option ?? cacheEntryOptions);

150 │ if (!\_key.Any(t => t == key))

151 │ {

152 │ \_key.Add(key);

153 │ }

154 │ }

155 │ return obj ?? Activator.CreateInstance<T>();

156 │ }

157 │

158 │

159 │ public static T GetValue<T>(this IMemoryCache \_memoryCache, string key, CallBack<T> callBack, MemoryCacheEntryOp...

160 │ {

161 │ return callBack();

162 │ if (!\_memoryCache.TryGetValue(key, out T? obj))

163 │ {

164 │ obj = callBack();

165 │ \_memoryCache.Set(key, obj, option ?? cacheEntryOptions);

166 │ if (!\_key.Any(t => t == key))

167 │ {

168 │ \_key.Add(key);

169 │ }

170 │ }

171 │ return obj.DeepCopy();

172 │ }

173 │ public delegate Task<T> CallBackAsync<T>();

174 │

175 │ public static async Task<T> GetValueAsync<T>(this IMemoryCache \_memoryCache, string key, CallBackAsync<T> callBa...

176 │ {

177 │ return await callBack();

178 │ if (!\_memoryCache.TryGetValue(key, out T? obj))

179 │ {

180 │ obj = await callBack();

181 │ \_memoryCache.Set(key, obj, option ?? cacheEntryOptions);

182 │ if (!\_key.Any(t => t == key))

183 │ {

184 │ \_key.Add(key);

185 │ }

186 │ }

187 │ return obj.DeepCopy();

188 │ }

189 │ public static void Clear(this IMemoryCache \_memoryCache, Func<string, bool>? predicate)

190 │ {

191 │ List<string> keyRemove = new List<string>();

192 │ foreach (var k in predicate != null ? \_key.Where(predicate) : \_key)

193 │ {

194 │ try

195 │ {

196 │ keyRemove.Add(k);

197 │ \_memoryCache.Remove(k);

198 │ }

199 │ catch { }

200 │ }

201 │ foreach (var key in keyRemove)

202 │ {

203 │ \_key.Remove(key);

204 │ }

205 │ }

206 │ }

207 │ }

────────────────────────────────────────────────────────────

**📄 FileController.cs (CS, 614 lines)**

1 │ ﻿using QuanLyBanHang.Common.Attributes;

2 │ using QuanLyBanHang.Models.APIModels;

3 │ using Microsoft.AspNetCore.Mvc;

4 │ using QuanLyBanHang.Common;

5 │ using System.IO;

6 │ using QuanLyBanHang.Models.QLBHEntity;

7 │ using Microsoft.AspNetCore.SignalR;

8 │ using SixLabors.ImageSharp;

9 │ using SixLabors.ImageSharp.Formats.Jpeg;

10 │ using SixLabors.ImageSharp.PixelFormats;

11 │ using iText.Kernel.Pdf;

12 │ using iText.Kernel.Utils;

13 │ using iText.IO.Source;

14 │ using iText.Bouncycastle;

15 │ using System.Diagnostics;

16 │

17 │ namespace QuanLyBanHang.Controllers

18 │ {

19 │ public class FileController : BaseController

20 │ {

21 │ private readonly IWebHostEnvironment \_hostingEnvironment;

22 │ private readonly string wwwroot = "wwwroot";

23 │

24 │ public FileController(QuanLyBanHangContext dbcontext, IWebHostEnvironment hostingEnvironment, MemoryCacheManager...

25 │ {

26 │ \_hostingEnvironment = hostingEnvironment;

27 │ }

28 │

29 │ [HttpPost]

30 │ [DisableRequestSizeLimit]

31 │ [TokenUserCheckHTTP(Permission.NotCheck)]

32 │ public ActionResult UploadImage()

33 │ {

34 │ SysUser? user = dbcontext.SysUser.Where(e => e.Ma == objToken!.userid).FirstOrDefault();

35 │ return \_uploadFile("/upload/", user?.Username ?? "");

36 │

37 │ }

38 │

39 │

40 │ private ActionResult \_uploadFile(string urlImagePath, string module)

41 │ {

42 │ if (!module.IsEmpty())

43 │ {

44 │ module += "/";

45 │ }

46 │ var filelist = HttpContext.Request.Form.Files;

47 │ if (filelist.Count > 0 && filelist[0].FileName != "")

48 │ {

49 │ for (int i = 0; i < filelist.Count; i++)

50 │ {

51 │ IFormFile file = filelist[i];

52 │ string fileExtension = Path.GetExtension(file.FileName);

53 │ if (QLBHConst.AllowImageExt.FirstOrDefault(t => t.Equals(fileExtension.ToUpper())) == null)

54 │ {

55 │ return Json(new ResponseModel(QLBHConst.Error, "Định dạng ." + fileExtension + " không h��...

56 │ }

57 │ if (file.Length / 1048576 > 500)

58 │ {

59 │ return Json(new ResponseModel(QLBHConst.Error, "Dung lượng file quá lớn !!!"));

60 │ }

61 │ }

62 │

63 │ string filePath = wwwroot + urlImagePath + module;

64 │

65 │ if (!Directory.Exists(filePath))

66 │ {

67 │ Directory.CreateDirectory(filePath);

68 │ }

69 │

70 │ List<string> filesURL = new List<string>();

71 │ for (int i = 0; i < filelist.Count; i++)

72 │ {

73 │ IFormFile file = filelist[i];

74 │ string filename = Utility.RandomString(15) + Path.GetExtension(file.FileName);

75 │ string copy = Path.Combine(\_hostingEnvironment.ContentRootPath, filePath, filename);

76 │ using (var stream = System.IO.File.Create(copy))

77 │ {

78 │ file.CopyTo(stream);

79 │ }

80 │ filesURL.Add(urlImagePath + module + filename);

81 │ }

82 │ if (filelist.Count == filesURL.Count)

83 │ {

84 │ return Json(new ResponseModel(QLBHConst.OK, "Thông tin lưu thành công !!", filesURL));

85 │ }

86 │ else

87 │ {

88 │ return Json(new ResponseModel(QLBHConst.Error, ""));

89 │ }

90 │ }

91 │

92 │ return Json(new ResponseModel(QLBHConst.Error, "Không có file trong tệp upload"));

93 │ }

94 │

95 │

96 │ [HttpPost]

97 │ public ActionResult UploadImagePublic()

98 │ {

99 │ var filelist = HttpContext.Request.Form.Files;

100 │ if (filelist.Count > 0 && filelist[0].FileName != "")

101 │ {

102 │ for (int i = 0; i < filelist.Count; i++)

103 │ {

104 │ IFormFile file = filelist[i];

105 │ string fileExtension = Path.GetExtension(file.FileName);

106 │ if (QLBHConst.AllowImageExt.FirstOrDefault(t => t.Equals(fileExtension.ToUpper())) == null)

107 │ {

108 │ return Json(new ResponseModel(QLBHConst.Error, "Định dạng ." + fileExtension + " không h��...

109 │ }

110 │ if (file.Length / 1048576 > 50)

111 │ {

112 │ return Json(new ResponseModel(QLBHConst.Error, "Dung lượng file quá lớn !!!"));

113 │ }

114 │ }

115 │ List<string> filesURL = new List<string>();

116 │ string urlImagePath = "/upload/";

117 │ for (int i = 0; i < filelist.Count; i++)

118 │ {

119 │ IFormFile file = filelist[i];

120 │ string filename = Utility.RandomString(15) + Path.GetExtension(file.FileName);

121 │ string copy = Path.Combine(\_hostingEnvironment.ContentRootPath, "wwwroot" + urlImagePath, filename);

122 │ using (var stream = System.IO.File.Create(copy))

123 │ {

124 │ file.CopyTo(stream);

125 │ }

126 │ filesURL.Add(urlImagePath + filename);

127 │ }

128 │ if (filelist.Count == filesURL.Count)

129 │ {

130 │ return Json(new ResponseModel(QLBHConst.OK, "Thông tin lưu thành công !!", filesURL));

131 │ }

132 │ else

133 │ {

134 │ return Json(new ResponseModel(QLBHConst.Error, ""));

135 │ }

136 │ }

137 │

138 │ return Json(new ResponseModel(QLBHConst.Error, "Không có file trong tệp upload"));

139 │ }

140 │

141 │ [HttpPost]

142 │ [DisableRequestSizeLimit]

143 │ [TokenUserCheckHTTP()]

144 │ public ActionResult UploadFile()

145 │ {

146 │ var filelist = HttpContext.Request.Form.Files;

147 │ string module = "";

148 │ try

149 │ {

150 │ var dict = Request.Form.ToDictionary(x => x.Key, x => x.Value.ToString());

151 │ if (dict.Count > 0)

152 │ {

153 │ module = dict["module"];

154 │ }

155 │ }

156 │ catch { }

157 │ if (filelist.Count > 0 && filelist[0].FileName != "")

158 │ {

159 │ List<string> AllAllowExt = new List<string>();

160 │ AllAllowExt.AddRange(QLBHConst.AllowDocumentExt);

161 │ AllAllowExt.AddRange(QLBHConst.AllowImageExt);

162 │ AllAllowExt.AddRange(QLBHConst.AllowAudioExt);

163 │ AllAllowExt.AddRange(QLBHConst.AllowVideoExt);

164 │ Utility.CheckPath(QLBHConst.UploadPath);

165 │ Utility.CheckPath(QLBHConst.DocumentPath);

166 │ Utility.CheckPath(QLBHConst.ImagePath);

167 │ Utility.CheckPath(QLBHConst.AudioPath);

168 │ Utility.CheckPath(QLBHConst.VideoPath);

169 │ for (int i = 0; i < filelist.Count; i++)

170 │ {

171 │ IFormFile file = filelist[i];

172 │ string fileExtension = Path.GetExtension(file.FileName);

173 │ if (AllAllowExt.FirstOrDefault(t => t.Equals(fileExtension.ToUpper())) == null)

174 │ {

175 │ return Json(new ResponseModel("Error", "Định dạng ." + fileExtension + " không hợp lệ!...

176 │ }

177 │ if (Math.Round((double)file.Length / 1048576) > QLBHConst.MaxFileSize)

178 │ {

179 │ return Json(new ResponseModel("Error", "Dung lượng file quá lớn !!!"));

180 │ }

181 │ }

182 │

183 │ List<string> filesURL = new List<string>();

184 │ try

185 │ {

186 │ for (int i = 0; i < filelist.Count; i++)

187 │ {

188 │ IFormFile file = filelist[i];

189 │ string fileExtension = Path.GetExtension(file.FileName);

190 │ string filename = Utility.RandomString(30) + fileExtension;

191 │ string copy = "";

192 │

193 │ if (QLBHConst.AllowDocumentExt.FirstOrDefault(t => t.Equals(fileExtension.ToUpper())) != null)

194 │ {

195 │ if (module != "")

196 │ {

197 │ string tmp = Path.Combine(QLBHConst.DocumentPath, module);

198 │ Utility.CheckPath(tmp);

199 │ copy = Path.Combine(tmp, filename);

200 │ }

201 │ else

202 │ {

203 │ copy = Path.Combine(QLBHConst.DocumentPath, filename);

204 │ }

205 │

206 │ }

207 │ else if (QLBHConst.AllowImageExt.FirstOrDefault(t => t.Equals(fileExtension.ToUpper())) != null)

208 │ {

209 │ if (module != "")

210 │ {

211 │ string tmp = Path.Combine(QLBHConst.ImagePath, module);

212 │ Utility.CheckPath(tmp);

213 │ copy = Path.Combine(tmp, filename);

214 │ }

215 │ else

216 │ {

217 │ copy = Path.Combine(QLBHConst.ImagePath, filename);

218 │ }

219 │ }

220 │ else if (QLBHConst.AllowAudioExt.FirstOrDefault(t => t.Equals(fileExtension.ToUpper())) != null)

221 │ {

222 │ if (module != "")

223 │ {

224 │ string tmp = Path.Combine(QLBHConst.AudioPath, module);

225 │ Utility.CheckPath(tmp);

226 │ copy = Path.Combine(tmp, filename);

227 │ }

228 │ else

229 │ {

230 │ copy = Path.Combine(QLBHConst.AudioPath, filename);

231 │ }

232 │ }

233 │ else if (QLBHConst.AllowVideoExt.FirstOrDefault(t => t.Equals(fileExtension.ToUpper())) != null)

234 │ {

235 │ if (module != "")

236 │ {

237 │ string tmp = Path.Combine(QLBHConst.VideoPath, module);

238 │ Utility.CheckPath(tmp);

239 │ copy = Path.Combine(tmp, filename);

240 │ }

241 │ else

242 │ {

243 │ copy = Path.Combine(QLBHConst.VideoPath, filename);

244 │ }

245 │ }

246 │ using (var stream = System.IO.File.Create(copy))

247 │ {

248 │ file.CopyTo(stream);

249 │ }

250 │ if (module != "")

251 │ {

252 │ filesURL.Add(Path.Combine(QLBHConst.DownloadPath, module, filename)); //QLBHConst.Down...

253 │ }

254 │ else

255 │ {

256 │ filesURL.Add(QLBHConst.DownloadPath + filename);

257 │ }

258 │

259 │ }

260 │ }

261 │ catch (Exception ex)

262 │ {

263 │ Debug.WriteLine(ex);

264 │ }

265 │ if (filelist.Count == filesURL.Count)

266 │ {

267 │ return Json(new ResponseModel("OK", "Thông tin lưu thành công !!!", filesURL));

268 │ }

269 │ else

270 │ {

271 │ return Json(new ResponseModel("Error", "Có lỗi xảy ra !!!", filesURL));

272 │ }

273 │ }

274 │ return Json(new ResponseModel("Error", "Không có file trong tệp upload !!!"));

275 │ }

276 │ [HttpPost]

277 │ public ActionResult UploadFilePublic()

278 │ {

279 │ var filelist = HttpContext.Request.Form.Files;

280 │ if (filelist.Count > 0 && filelist[0].FileName != "")

281 │ {

282 │ List<string> AllAllowExt = new List<string>();

283 │ AllAllowExt.AddRange(QLBHConst.AllowDocumentExt);

284 │ AllAllowExt.AddRange(QLBHConst.AllowImageExt);

285 │ AllAllowExt.AddRange(QLBHConst.AllowAudioExt);

286 │ AllAllowExt.AddRange(QLBHConst.AllowVideoExt);

287 │ Utility.CheckPath(QLBHConst.UploadPath);

288 │ Utility.CheckPath(QLBHConst.DocumentPath);

289 │ Utility.CheckPath(QLBHConst.ImagePath);

290 │ Utility.CheckPath(QLBHConst.AudioPath);

291 │ Utility.CheckPath(QLBHConst.VideoPath);

292 │ for (int i = 0; i < filelist.Count; i++)

293 │ {

294 │ IFormFile file = filelist[i];

295 │ string fileExtension = Path.GetExtension(file.FileName);

296 │ if (AllAllowExt.FirstOrDefault(t => t.Equals(fileExtension.ToUpper())) == null)

297 │ {

298 │ return Json(new ResponseModel("Error", "Định dạng ." + fileExtension + " không hợp lệ!...

299 │ }

300 │ if (Math.Round((double)file.Length / 1048576) > QLBHConst.MaxFileSize)

301 │ {

302 │ return Json(new ResponseModel("Error", "Dung lượng file quá lớn !!!"));

303 │ }

304 │ }

305 │

306 │ List<string> filesURL = new List<string>();

307 │ try

308 │ {

309 │ for (int i = 0; i < filelist.Count; i++)

310 │ {

311 │ IFormFile file = filelist[i];

312 │ string fileExtension = Path.GetExtension(file.FileName);

313 │ string filename = Utility.RandomString(30) + fileExtension;

314 │ string copy = "";

315 │

316 │ if (QLBHConst.AllowDocumentExt.FirstOrDefault(t => t.Equals(fileExtension.ToUpper())) != null)

317 │ {

318 │ copy = Path.Combine(QLBHConst.DocumentPath, filename);

319 │ }

320 │ else if (QLBHConst.AllowImageExt.FirstOrDefault(t => t.Equals(fileExtension.ToUpper())) != null)

321 │ {

322 │ copy = Path.Combine(QLBHConst.ImagePath, filename);

323 │ }

324 │ else if (QLBHConst.AllowAudioExt.FirstOrDefault(t => t.Equals(fileExtension.ToUpper())) != null)

325 │ {

326 │ copy = Path.Combine(QLBHConst.AudioPath, filename);

327 │ }

328 │ else if (QLBHConst.AllowVideoExt.FirstOrDefault(t => t.Equals(fileExtension.ToUpper())) != null)

329 │ {

330 │ copy = Path.Combine(QLBHConst.VideoPath, filename);

331 │ }

332 │ using (var stream = System.IO.File.Create(copy))

333 │ {

334 │ file.CopyTo(stream);

335 │ }

336 │

337 │ filesURL.Add(QLBHConst.DownloadPath + filename);

338 │ }

339 │ }

340 │ catch (Exception ex)

341 │ {

342 │

343 │ }

344 │ if (filelist.Count == filesURL.Count)

345 │ {

346 │ return Json(new ResponseModel("OK", "Thông tin lưu thành công !!!", filesURL));

347 │ }

348 │ else

349 │ {

350 │ return Json(new ResponseModel("Error", "Có lỗi xảy ra !!!", filesURL));

351 │ }

352 │ }

353 │ return Json(new ResponseModel("Error", "Không có file trong tệp upload !!!"));

354 │ }

355 │ [HttpGet]

356 │ public ActionResult DownloadFile(string filename, string view = "0")

357 │ {

358 │ if (filename == null || filename == "")

359 │ {

360 │ return NotFound();

361 │ }

362 │ string p = \_hostingEnvironment.EnvironmentName;

363 │ string fileExtension = Path.GetExtension(filename).ToUpper();

364 │ string fullPath = "";

365 │ if (QLBHConst.AllowDocumentExt.Any(t => t.Equals(fileExtension)))

366 │ {

367 │ fullPath = QLBHConst.DocumentPath + "\\" + filename;

368 │ }

369 │ else if (QLBHConst.AllowImageExt.FirstOrDefault(t => t.Equals(fileExtension)) != null)

370 │ {

371 │ fullPath = QLBHConst.ImagePath + "\\" + filename;

372 │ }

373 │ else if (QLBHConst.AllowAudioExt.FirstOrDefault(t => t.Equals(fileExtension)) != null)

374 │ {

375 │ fullPath = QLBHConst.AudioPath + "\\" + filename;

376 │ }

377 │ else if (QLBHConst.AllowVideoExt.FirstOrDefault(t => t.Equals(fileExtension)) != null)

378 │ {

379 │ fullPath = QLBHConst.VideoPath + "\\" + filename;

380 │ }

381 │ if (!System.IO.File.Exists(fullPath))

382 │ {

383 │ return NotFound();

384 │ }

385 │

386 │ string contentType = "application/msword";

387 │ switch (fileExtension)

388 │ {

389 │ //Document

390 │ case ".DOC":

391 │ contentType = "application/msword";

392 │ break;

393 │ case ".DOCX":

394 │ contentType = "application/vnd.openxmlformats-officedocument.wordprocessingml.document";

395 │ break;

396 │ case ".XLS":

397 │ contentType = "application/vnd.ms-excel";

398 │ break;

399 │ case ".XLSX":

400 │ contentType = "application/vnd.openxmlformats-officedocument.spreadsheetml.sheet";

401 │ break;

402 │ case ".XLSB":

403 │ contentType = "application/vnd.ms-excel.sheet.binary.macroEnabled.12";

404 │ break;

405 │ case ".PPT":

406 │ contentType = "application/vnd.ms-powerpoint";

407 │ break;

408 │ case ".PPTX":

409 │ contentType = "application/vnd.openxmlformats-officedocument.presentationml.presentation";

410 │ break;

411 │ case ".PDF":

412 │ contentType = "application/pdf";

413 │ break;

414 │ //Image

415 │ case ".JPG":

416 │ case ".JPE":

417 │ contentType = "image/jpeg";

418 │ break;

419 │ case ".BMP":

420 │ contentType = "image/bmp";

421 │ break;

422 │ case ".GIF":

423 │ contentType = "application/gif";

424 │ break;

425 │ case ".PNG":

426 │ contentType = "image/png";

427 │ break;

428 │ //Audio

429 │ case ".MP3":

430 │ contentType = "audio/mpeg";

431 │ break;

432 │ case ".AAC":

433 │ contentType = "audio/aac";

434 │ break;

435 │ case ".WMA":

436 │ contentType = "audio/x-ms-wma";

437 │ break;

438 │ case ".WAV":

439 │ contentType = "audio/wav";

440 │ break;

441 │ //Video

442 │ case ".AVI":

443 │ contentType = "video/x-msvideo";

444 │ break;

445 │ case ".MP4":

446 │ contentType = "video/mp4";

447 │ break;

448 │ case ".WMV":

449 │ contentType = "video/x-ms-wmv";

450 │ break;

451 │ case ".FLV":

452 │ contentType = "video/x-flv";

453 │ break;

454 │ case ".3GP":

455 │ contentType = "video/3gpp";

456 │ break;

457 │ default:

458 │ break;

459 │ }

460 │

461 │ var filestream = System.IO.File.OpenRead(fullPath);

462 │

463 │ /\* var net = new System.Net.WebClient();

464 │ var data = net.DownloadData(fullPath);

465 │ var content = new System.IO.MemoryStream(data);\*/

466 │ if (view == "1")

467 │ {

468 │ return File(filestream, contentType);

469 │ }

470 │ return File(filestream, contentType, filename, enableRangeProcessing: true);

471 │ }

472 │ [HttpPost]

473 │ [TokenUserCheckHTTP()]

474 │ public void DeleteFile(string Files)

475 │ {

476 │

477 │ string[] arr\_File = (Files + "").Split(';');

478 │ int idx = -1;

479 │ string filename = "";

480 │ foreach (string file in arr\_File)

481 │ {

482 │ if (!file.Trim().IsEmpty())

483 │ {

484 │ try

485 │ {

486 │ filename = file;

487 │ idx = file.IndexOf("filename=");

488 │ if (idx > -1)

489 │ {

490 │ filename = filename.Substring(idx + 9);

491 │ }

492 │ if (filename.StartsWith("\\"))

493 │ {

494 │ filename = filename.Substring(1);

495 │ }

496 │ string fileExtension = Path.GetExtension(filename).ToUpper();

497 │ string fullPath = "";

498 │ if (QLBHConst.AllowDocumentExt.Any(t => t.Equals(fileExtension)))

499 │ {

500 │ fullPath = QLBHConst.DocumentPath + "\\" + filename;

501 │ }

502 │ else if (QLBHConst.AllowImageExt.FirstOrDefault(t => t.Equals(fileExtension)) != null)

503 │ {

504 │ fullPath = QLBHConst.ImagePath + "\\" + filename;

505 │ }

506 │ else if (QLBHConst.AllowAudioExt.FirstOrDefault(t => t.Equals(fileExtension)) != null)

507 │ {

508 │ fullPath = QLBHConst.AudioPath + "\\" + filename;

509 │ }

510 │ else if (QLBHConst.AllowVideoExt.FirstOrDefault(t => t.Equals(fileExtension)) != null)

511 │ {

512 │ fullPath = QLBHConst.VideoPath + "\\" + filename;

513 │ }

514 │ if (System.IO.File.Exists(fullPath))

515 │ {

516 │ System.IO.File.Delete(fullPath);

517 │ }

518 │ }

519 │ catch { }

520 │ }

521 │ }

522 │ }

523 │

524 │ [HttpPost]

525 │ [TokenUserCheckHTTP()]

526 │ public IActionResult AddFileHoSoBenhAn(string NoiDung, string OldFilePath)

527 │ {

528 │ string path = "";

529 │ Utility.CheckPath(QLBHConst.HoSoBenhAnPath);

530 │ if (OldFilePath.IsEmpty() || (OldFilePath + "").IsEmpty())

531 │ {

532 │ string filename = Utility.RandomString(30) + ".txt";

533 │ path = Path.Combine(QLBHConst.HoSoBenhAnPath, filename);

534 │ }

535 │ else

536 │ {

537 │ path = OldFilePath;

538 │ }

539 │

540 │ System.IO.File.WriteAllText(path, NoiDung);

541 │ return Json(new ResponseModel("OK", "Thông tin lưu thành công !!!", path));

542 │ }

543 │

544 │ [HttpPost]

545 │ public async Task<IActionResult> UploadCompressedImageFileAsync([FromForm] IFormFile File)

546 │ {

547 │ var file = File;

548 │ if (file == null || file.Length == 0)

549 │ return Ok(new { success = false, message = "File không tồn tại hoặc rỗng.", file\_name\_old = fil...

550 │

551 │ try

552 │ {

553 │ byte[] processedBytes;

554 │ string fileName = file.FileName;

555 │ string extension = Path.GetExtension(fileName).ToLowerInvariant();

556 │

557 │ if (Utility.IsImage(extension))

558 │ {

559 │ using (var image = Image.Load<Rgba32>(file.OpenReadStream()))

560 │ using (var ms = new MemoryStream())

561 │ {

562 │ image.SaveAsJpeg(ms, new JpegEncoder { Quality = 80 });

563 │ processedBytes = ms.ToArray();

564 │ fileName = Utility.RandomString(10) + ".jpg";

565 │ }

566 │ }

567 │ else if (extension == ".pdf")

568 │ {

569 │ using (var ms = new MemoryStream())

570 │ {

571 │ file.CopyTo(ms);

572 │ ms.Position = 0;

573 │ processedBytes = CompressPdf(ms);

574 │ }

575 │ }

576 │ else

577 │ {

578 │ using (var inputStream = file.OpenReadStream())

579 │ using (var ms = new MemoryStream())

580 │ {

581 │ inputStream.CopyTo(ms);

582 │ processedBytes = ms.ToArray();

583 │ }

584 │ }

585 │

586 │ var url = await Utility.UploadFile(processedBytes, fileName);

587 │ return Ok(new { success = true, url, file\_name\_old = file.FileName });

588 │ }

589 │ catch (Exception ex)

590 │ {

591 │ return Ok(new { success = false, message = ex.Message, url = "", file\_name\_old = "" });

592 │ }

593 │ }

594 │

595 │ public static byte[] CompressPdf(Stream inputStream)

596 │ {

597 │ using (var outputStream = new MemoryStream())

598 │ {

599 │ var writerProperties = new WriterProperties()

600 │ .SetCompressionLevel(CompressionConstants.BEST\_COMPRESSION);

601 │

602 │ using (var pdfReader = new PdfReader(inputStream))

603 │ using (var pdfWriter = new PdfWriter(outputStream, writerProperties))

604 │ using (var pdfDocument = new PdfDocument(pdfReader, pdfWriter))

605 │ {

606 │ pdfDocument.SetFlushUnusedObjects(true);

607 │ }

608 │

609 │ return outputStream.ToArray();

610 │ }

611 │ }

612 │ }

613 │

614 │ }

────────────────────────────────────────────────────────────

**📄 FirebaseUtility.cs (CS, 432 lines)**

1 │ ﻿using FirebaseAdmin.Messaging;

2 │ using FirebaseAdmin;

3 │ using Google.Apis.Auth.OAuth2;

4 │ using System;

5 │ using QuanLyBanHang.Models.QLBHEntity;

6 │ using QuanLyBanHang.Common;

7 │ using Newtonsoft.Json;

8 │ using static QuanLyBanHang.Models.ExModel.OPS.OrderDetailModel;

9 │

10 │ namespace HongHungHIS.Common.Firebase

11 │ {

12 │ public class FirebaseUtility

13 │ {

14 │ private class \_MesThongBaoApp

15 │ {

16 │ public readonly string id;

17 │ public readonly string title;

18 │ public readonly string body;

19 │ public readonly string token;

20 │ public readonly bool enableReminder;

21 │ public readonly int badge;

22 │

23 │ public \_MesThongBaoApp(string id, string title, string body, string token, bool enableReminder, int badge)

24 │ {

25 │ this.id = id;

26 │ this.title = title;

27 │ this.body = body;

28 │ this.token = token;

29 │ this.enableReminder = enableReminder;

30 │ this.badge = badge;

31 │ }

32 │ }

33 │

34 │ public static readonly int idMax = 100000000;

35 │ private static readonly Random random = new Random();

36 │ private static FirebaseMessaging mess;

37 │ static FirebaseUtility()

38 │ {

39 │ var app = FirebaseApp.Create(new AppOptions()

40 │ {

41 │ Credential = GoogleCredential.FromJson("{\r\n \"type\": \"service\_account\",\r\n \"project\_id\": \"tan...

42 │ });

43 │ mess = FirebaseMessaging.GetMessaging(app);

44 │ }

45 │

46 │ public static IEnumerable<List<T>> Chunk<T>(List<T> source, int chunkSize)

47 │ {

48 │ for (int i = 0; i < source.Count; i += chunkSize)

49 │ {

50 │ yield return source.Skip(i).Take(chunkSize).ToList();

51 │ }

52 │ }

53 │

54 │ private static List<\_MesThongBaoApp> createMessage(List<SysUser> users, string title, string body)

55 │ {

56 │ users = users.Where(e => !e.Fcm.IsEmpty()).ToList();

57 │ int length = users.Count();

58 │ var mess = users.Select(e => new { MaUser = e.Ma, TieuDe = title, NoiDung = body }).ToList();

59 │ var tokens = new Dictionary<Guid, string>();

60 │ var enableReminder = new Dictionary<Guid, bool>();

61 │ for (int i = 0; i < length; i++)

62 │ {

63 │ tokens[users[i].Ma] = users[i].Fcm;

64 │ enableReminder[users[i].Ma] = true;

65 │ //enableReminder[users[i].Ma] = users[i].Reminder ?? true;

66 │ }

67 │ List<\_MesThongBaoApp> res = new List<\_MesThongBaoApp>();

68 │

69 │ for (int i = 0; i < length; i++)

70 │ {

71 │ var msg = mess[i];

72 │ res.Add(new \_MesThongBaoApp("", msg.TieuDe + "", msg.NoiDung + "",

73 │ tokens[msg.MaUser] + "", enableReminder[msg.MaUser], 0));

74 │ }

75 │ return res;

76 │ }

77 │

78 │ private static void sendMessage(List<\_MesThongBaoApp> res, string action, SoundModel sound)

79 │ {

80 │ res = res.Where(e => e.enableReminder).ToList();

81 │ var chunk = Chunk(res, 500).ToList();

82 │ for (int k = 0; k < chunk.Count; k++)

83 │ {

84 │ List<Message> messages = new List<Message>();

85 │ for (int i = 0; i < chunk[k].Count; i++)

86 │ {

87 │ var msg = chunk[k][i];

88 │ messages.Add(new Message

89 │ {

90 │ Token = msg.token,

91 │ Notification = new Notification

92 │ {

93 │ Title = msg.title,

94 │ Body = msg.body,

95 │

96 │ },

97 │ Data = new Dictionary<string, string>

98 │ {

99 │ { "action", action },

100 │ { "title", msg.title },

101 │ { "body", msg.body },

102 │ { "id" , msg.id },

103 │ { "badge" , msg.badge+"" }

104 │ },

105 │ Android = new AndroidConfig

106 │ {

107 │ Priority = Priority.High,

108 │ Notification = new AndroidNotification

109 │ {

110 │ Sound = sound.androidName, //android <26

111 │ ChannelId = sound.androidChannel, //android >=26

112 │ },

113 │ },

114 │ Apns = new ApnsConfig

115 │ {

116 │ Headers = new Dictionary<string, string>()

117 │ {

118 │ { "apns-priority", "10" },

119 │ },

120 │ Aps = new Aps

121 │ {

122 │ //Sound = "marimba.wav",

123 │ ContentAvailable = true,

124 │ CriticalSound = new CriticalSound()

125 │ {

126 │ Name = sound.iosName,

127 │ Volume = 1,

128 │ Critical = true,

129 │ },

130 │ Badge = msg.badge,

131 │ },

132 │

133 │ },

134 │

135 │ });

136 │ }

137 │ mess.SendEachAsync(messages);

138 │ }

139 │ }

140 │

141 │

142 │ //public static async Task NotifyAction(BackgroundSyncDBServices \_backgroundServices, List<SysUser> users, strin...

143 │ //{

144 │ // saveMessage(\_backgroundServices, users, title, body, (mess) =>

145 │ // {

146 │ // sendMessage(mess, FirebaseActionKey.notifyAction, sound);

147 │ // return true;

148 │ // });

149 │ //}

150 │

151 │ public static void NotifyActionWithNoSave(List<SysUser> users, string title, string body, SoundModel sound)

152 │ {

153 │ var mess = createMessage(users, title, body);

154 │ sendMessage(mess, FirebaseActionKey.notifyAction, sound);

155 │ }

156 │

157 │ public static void sendRequestLogin(SysUser User, string connectionID, string sign)

158 │ {

159 │ \_ = mess.SendAsync(new Message

160 │ {

161 │ Token = User.Fcm,//User.FCM,

162 │ Notification = new Notification

163 │ {

164 │ Title = "Cảnh báo bảo mật",

165 │ Body = "Tài khoản của bạn đang được đăng nhập ở một thiết bị khác. Vui lòn...

166 │ ImageUrl = "https://tapioca.hungduy.vn/assets/images/tanione-logo.jpg"

167 │ },

168 │ Data = new Dictionary<string, string>

169 │ {

170 │ { "action", FirebaseActionKey.requestLogin },

171 │ { "connectionId",connectionID },

172 │ { "sign",sign }

173 │ }

174 │ });

175 │ }

176 │

177 │

178 │

179 │

180 │

181 │ public static int randomId()

182 │ {

183 │ return random.Next(idMax);

184 │ }

185 │

186 │ /// <summary>

187 │ /// Dùng để tạo tin nhắn cho tất cả thiết bị di động được kết nối

188 │ /// </summary>

189 │ /// <param name="group">

190 │ /// Các thiết bị đã kết nối với group được chọn sẽ nhận được tin nhắn

191 │ /// </param>

192 │ /// <param name="action">

193 │ /// Dùng các action có trong <see cref="ReminderActionKey"/> để trigger chức năng sau khi tắt tin nh�...

194 │ /// </param>

195 │ public static void RegisterAlarm(List<string> tokens, int? id, string title, string body, string action, SoundMo...

196 │ {

197 │ return; //Chưa có gọi được đâu vì chưa có xin quyền Exact alarms

198 │ //tokens = new List<string> { "fD7pc1-ArUbDuoFHfwp\_Im:APA91bGvC6GO7YQeBa6QpWPw6cML4dsaC1ES7VXnhvw435DgSVRybq...

199 │ if (id == null)

200 │ {

201 │ id = randomId();

202 │ }

203 │ var chunk = Chunk(tokens, 500).ToList();

204 │ for (int k = 0; k < chunk.Count; k++)

205 │ {

206 │ List<Message> messages = new List<Message>();

207 │ for (int i = 0; i < chunk[k].Count; i++)

208 │ {

209 │ messages.Add(new Message

210 │ {

211 │ Token = chunk[k][i],

212 │ Data = new Dictionary<string, string>

213 │ {

214 │ { "action", FirebaseActionKey.registerAlarm },

215 │ { "id", id.ToString() },

216 │ { "title", title },

217 │ { "body", body },

218 │ { "register\_action", action },

219 │ { "time", string.Format("{0:yyyy-MM-ddTHH:mm:ss}", time)},

220 │ { "android\_channel", sound.androidChannel },

221 │ { "ios\_sound", sound.iosName },

222 │ },

223 │ Android = new AndroidConfig

224 │ {

225 │ Priority = Priority.High,

226 │

227 │ },

228 │ Apns = new ApnsConfig

229 │ {

230 │ Headers = new Dictionary<string, string>()

231 │ {

232 │ { "apns-priority", "10" },

233 │ },

234 │ Aps = new Aps

235 │ {

236 │ ContentAvailable = true,

237 │ },

238 │

239 │ },

240 │

241 │ });

242 │ }

243 │ mess.SendEachAsync(messages);

244 │ }

245 │ }

246 │

247 │ public static void CancelAlarm(List<string> tokens, int? id)

248 │ {

249 │ if (id == null)

250 │ {

251 │ return;

252 │ }

253 │ var chunk = Chunk(tokens, 500).ToList();

254 │ for (int k = 0; k < chunk.Count; k++)

255 │ {

256 │ List<Message> messages = new List<Message>();

257 │ for (int i = 0; i < chunk[k].Count; i++)

258 │ {

259 │ messages.Add(new Message

260 │ {

261 │ Token = chunk[k][i],

262 │ Data = new Dictionary<string, string>

263 │ {

264 │ { "action", FirebaseActionKey.cancelAlarm },

265 │ { "id", id.ToString() },

266 │ },

267 │ Android = new AndroidConfig

268 │ {

269 │ Priority = Priority.High,

270 │

271 │ },

272 │ Apns = new ApnsConfig

273 │ {

274 │ Headers = new Dictionary<string, string>()

275 │ {

276 │ { "apns-priority", "10" },

277 │ },

278 │ Aps = new Aps

279 │ {

280 │

281 │ ContentAvailable = true,

282 │ },

283 │

284 │ },

285 │

286 │ });

287 │ }

288 │ mess.SendEachAsync(messages);

289 │ }

290 │ }

291 │

292 │

293 │ public static void NotifyOrderAction(List<SysUser> users, OrderNotificationData orderData, SoundModel sound)

294 │ {

295 │ users = users.Where(e => !e.Fcm.IsEmpty()).ToList();

296 │

297 │ var chunk = Chunk(users.Select(u => u.Fcm).ToList(), 500).ToList();

298 │

299 │ for (int k = 0; k < chunk.Count; k++)

300 │ {

301 │ List<Message> messages = new List<Message>();

302 │

303 │ for (int i = 0; i < chunk[k].Count; i++)

304 │ {

305 │ var token = chunk[k][i];

306 │

307 │ messages.Add(new Message

308 │ {

309 │ Token = token,

310 │ Notification = new Notification

311 │ {

312 │ Title = $"Đơn hàng mới - #{orderData.OrderCode.Substring(Math.Max(0, orderData.OrderCod...

313 │ Body = $"{orderData.TenKho} - {orderData.TenPhong} - ({orderData.ItemCount} món)",

314 │ },

315 │ Data = new Dictionary<string, string>

316 │ {

317 │ { "action", FirebaseActionKey.orderNotification },

318 │ { "orderId", orderData.OrderCode },

319 │ { "orderMaDonHang", orderData.MaDonHang.ToString() },

320 │ { "maKho", orderData.MaKho.ToString() },

321 │ { "maPhong", orderData.MaPhong.ToString() },

322 │ { "tenPhong", orderData.TenPhong },

323 │ { "tenKho", orderData.TenKho },

324 │ { "tenKH", orderData.TenKH },

325 │ { "sdt", orderData.SDT },

326 │ { "itemCount", orderData.ItemCount.ToString() },

327 │ { "danhSachMon", JsonConvert.SerializeObject(orderData.DanhSachMon) },

328 │ { "createdTime", DateTime.Now.ToString("yyyy-MM-ddTHH:mm:ss") }

329 │ },

330 │ Android = new AndroidConfig

331 │ {

332 │ Priority = Priority.High,

333 │ Notification = new AndroidNotification

334 │ {

335 │ Sound = sound.androidName,

336 │ ChannelId = sound.androidChannel,

337 │ },

338 │ },

339 │ Apns = new ApnsConfig

340 │ {

341 │ Headers = new Dictionary<string, string>()

342 │ {

343 │ { "apns-priority", "10" },

344 │ },

345 │ Aps = new Aps

346 │ {

347 │ ContentAvailable = true,

348 │ CriticalSound = new CriticalSound()

349 │ {

350 │ Name = sound.iosName,

351 │ Volume = 1,

352 │ Critical = true,

353 │ },

354 │ Badge = 1,

355 │ },

356 │ },

357 │ });

358 │ }

359 │

360 │ mess.SendEachAsync(messages);

361 │ }

362 │ }

363 │

364 │ // Model cho Order Notification Data

365 │ public class OrderNotificationData

366 │ {

367 │ public string OrderCode { get; set; }

368 │ public int MaDonHang { get; set; }

369 │ public int MaKho { get; set; }

370 │ public int MaPhong { get; set; }

371 │ public int MaNganhVoucher { get; set; }

372 │ public string TenPhong { get; set; }

373 │ public string TenKho { get; set; }

374 │ public string TenKH { get; set; }

375 │ public string SDT { get; set; }

376 │ public int ItemCount { get; set; }

377 │ public List<CartItem> DanhSachMon { get; set; }

378 │

379 │ public OrderNotificationData(string orderCode, int maDonHang, int maKho, int maPhong,int maNganhVoucher,

380 │ string tenPhong, string tenKho, string tenKH, string sdt, List<CartItem> danhSachMon)

381 │ {

382 │ OrderCode = orderCode;

383 │ MaDonHang = maDonHang;

384 │ MaKho = maKho;

385 │ MaPhong = maPhong;

386 │ MaNganhVoucher = maNganhVoucher;

387 │ TenPhong = tenPhong;

388 │ TenKho = tenKho;

389 │ TenKH = tenKH;

390 │ SDT = sdt;

391 │ DanhSachMon = danhSachMon ?? new List<CartItem>();

392 │ ItemCount = DanhSachMon.Count;

393 │ }

394 │ }

395 │ }

396 │ }

397 │

398 │

399 │ /// <summary>

400 │ /// Chức năng của tin nhắn firebase

401 │ /// </summary>

402 │ public class FirebaseActionKey

403 │ {

404 │ public static string requestLogin = "request\_login";

405 │ public static string registerAlarm = "register\_alarm";

406 │ public static string cancelAlarm = "cancel\_alarm";

407 │ public static string notifyAction = "notify\_action";

408 │ public static string orderNotification = "order\_notification";

409 │

410 │

411 │ }

412 │

413 │ public class FirebaseSound

414 │ {

415 │ public static readonly SoundModel none = new SoundModel("", "", "default");

416 │ public static readonly SoundModel channel = new SoundModel("marimba", "fa\_channel", "fa\_channel.wav");

417 │ public static readonly SoundModel subchannel = new SoundModel("marimba\_sub", "fa\_sub\_channel", "fa\_sub\_channel.wav")...

418 │ }

419 │

420 │ public class SoundModel

421 │ {

422 │ public SoundModel(string androidName, string androidChannel, string iosName)

423 │ {

424 │ this.androidName = androidName;

425 │ this.androidChannel = androidChannel;

426 │ this.iosName = iosName;

427 │ }

428 │

429 │ public string androidName;

430 │ public string androidChannel;

431 │ public string iosName;

432 │ }

────────────────────────────────────────────────────────────

**📄 GenQrRequest.cs (CS, 37 lines)**

1 │ ﻿namespace QuanLyBanHang.Models.VietinBank

2 │ {

3 │ public class GenQrRequest: ISignData

4 │ {

5 │ public string RequestId { get; set; }

6 │ public string ProviderId { get; set; }

7 │ public string MerchantId { get; set; }

8 │ public string Channel { get; set; }

9 │ public string Version { get; set; }

10 │ public string ClientIP { get; set; }

11 │ public string Language { get; set; }

12 │ public string ClientDt { get; set; }

13 │ public string Signature { get; set; }

14 │ public GenQrData Data { get; set; }

15 │

16 │ public string GetSignData()

17 │ {

18 │ return RequestId + ProviderId + MerchantId + ClientDt + Data.AccountNumber;

19 │ }

20 │

21 │ public void SetSignature(string signature)

22 │ {

23 │ Signature = signature;

24 │ }

25 │ }

26 │

27 │ public class GenQrData

28 │ {

29 │ public string AccountNumber { get; set; }

30 │ public string Amount { get; set; }

31 │ public string StoreLabel { get; set; }

32 │ public string ReferenceLabel { get; set; }

33 │ public string CustomerLabel { get; set; }

34 │ public string TerminalLabel { get; set; }

35 │ public string PurposeOfTrans { get; set; }

36 │ }

37 │ }

────────────────────────────────────────────────────────────

**📄 GenQrResponse.cs (CS, 34 lines)**

1 │ ﻿using static QuanLyBanHang.Models.VietinBank.ResponseData;

2 │

3 │ namespace QuanLyBanHang.Models.VietinBank

4 │ {

5 │ public class GenQrResponse : IResponseData

6 │ {

7 │ public string RequestId { get; set; }

8 │ public string ProviderId { get; set; }

9 │ public string MerchantId { get; set; }

10 │ public string Channel { get; set; }

11 │ public string Version { get; set; }

12 │ public string Language { get; set; }

13 │ public string ClientIP { get; set; }

14 │ public string ClientDt { get; set; }

15 │ public string Signature { get; set; }

16 │ public GenQrResponseData Data { get; set; }

17 │ public StatusOutputBaseApi Status { get; set; }

18 │

19 │ public string GetSignData()

20 │ {

21 │ return RequestId + ProviderId + MerchantId + ClientDt + Status.StatusCode;

22 │ }

23 │

24 │ public void SetSignature(string signature)

25 │ {

26 │ Signature = signature;

27 │ }

28 │ }

29 │

30 │ public class GenQrResponseData

31 │ {

32 │ public string Base64QRCode { get; set; }

33 │ }

34 │ }

────────────────────────────────────────────────────────────

**📄 GiaBanTheoKhoController.cs (CS, 261 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using Microsoft.EntityFrameworkCore;

3 │ using QuanLyBanHang.Common;

4 │ using QuanLyBanHang.Common.Attributes;

5 │ using QuanLyBanHang.Models.APIModels;

6 │ using QuanLyBanHang.Models.ExModel.CAT;

7 │ using QuanLyBanHang.Models.QLBHEntity;

8 │ using System.Data;

9 │

10 │ namespace QuanLyBanHang.Controllers.CAT

11 │ {

12 │ public class GiaBanTheoKhoController : BaseController

13 │ {

14 │ private readonly IWebHostEnvironment \_hostingEnvironment;

15 │ private readonly ILogger<GiaBanTheoKhoController> \_logger;

16 │

17 │ public GiaBanTheoKhoController(

18 │ QuanLyBanHangContext dbcontext,

19 │ MemoryCacheManager memoryCache,

20 │ IWebHostEnvironment hostingEnvironment,

21 │ ILogger<GiaBanTheoKhoController> logger)

22 │ : base(dbcontext, memoryCache)

23 │ {

24 │ m\_MenuID = MenuID.catGiaBanTheoKho.GetEnumGuid();

25 │ \_hostingEnvironment = hostingEnvironment;

26 │ \_logger = logger;

27 │ }

28 │

29 │ [TokenUserCheckHTTP]

30 │ public ActionResult Index(string MaKho)

31 │ {

32 │ \_logger.LogInformation("Người dùng {UserId} truy cập trang Giá bán theo kho. Mã kho: {MaKho}", objT...

33 │ try

34 │ {

35 │ LoadViewDataLists(MaKho);

36 │ return View();

37 │ }

38 │ catch (Exception ex)

39 │ {

40 │ \_logger.LogError(ex, "Lỗi khi tải dữ liệu cho trang Giá bán theo kho. Mã kho: {MaKho}", MaKho...

41 │ return View(); // Cân nhắc trả về một view báo lỗi

42 │ }

43 │ }

44 │

45 │ [TokenUserCheckHTTP(Permission.Add)]

46 │ public ActionResult Add(string MaKho)

47 │ {

48 │ \_logger.LogInformation("Người dùng {UserId} truy cập trang thêm mới Giá bán theo kho. Mã kho: {M...

49 │ LoadViewDataLists(MaKho);

50 │ return View("Form");

51 │ }

52 │

53 │ [TokenUserCheckHTTP(Permission.Edit)]

54 │ public ActionResult Edit(string Ma, string MaKho)

55 │ {

56 │ \_logger.LogInformation("Người dùng {UserId} yêu cầu chỉnh sửa Giá bán theo kho. Mã (đã mã h...

57 │ if (!Utility.TryDecryptToInt(Ma, out int id))

58 │ {

59 │ \_logger.LogWarning("Mã không hợp lệ được cung cấp: {Ma}", Ma);

60 │ return Json(new ResponseModel(QLBHConst.Error, "Mã không hợp lệ."));

61 │ }

62 │

63 │ try

64 │ {

65 │ LoadViewDataLists(MaKho);

66 │ ViewData["item"] = new GiaBanTheoKhoModel(dbcontext).GetItem(id, objToken.branch);

67 │ return View("Form");

68 │ }

69 │ catch (Exception ex)

70 │ {

71 │ \_logger.LogError(ex, "Lỗi khi tải dữ liệu để chỉnh sửa Giá bán theo kho ID: {Id}", id);

72 │ return RedirectToAction("Index", new { MaKho });

73 │ }

74 │ }

75 │

76 │ [HttpPost]

77 │ [TokenUserCheckHTTP(Permission.Add)]

78 │ public ActionResult AddAPI(string Ma, string MaKho, string MaHangHoa, string GiaBan, string GhiChu)

79 │ {

80 │ \_logger.LogInformation("Người dùng {UserId} thêm mới Giá bán. Kho: {MaKho}, Hàng hóa: {MaHangHoa}...

81 │ var validation = CheckRequireGiaBan(MaKho, MaHangHoa, GiaBan);

82 │ if (validation != null)

83 │ return Json(validation);

84 │

85 │ try

86 │ {

87 │ var result = new GiaBanTheoKhoModel(dbcontext).Add(

88 │ Cast.ToInt(MaKho),

89 │ Cast.ToInt(MaHangHoa),

90 │ Cast.ToInt(GiaBan),

91 │ GhiChu,

92 │ objToken.userid,

93 │ objToken.branch);

94 │

95 │ \_logger.LogInformation("Kết quả thêm mới Giá bán: {Result}", result);

96 │ return Json(Utility.MakeResponseFromAddResult(Cast.ToString(result)));

97 │ }

98 │ catch (Exception ex)

99 │ {

100 │ \_logger.LogError(ex, "Lỗi khi thêm mới Giá bán. Kho: {MaKho}, Hàng hóa: {MaHangHoa}", MaKho, Ma...

101 │ return Json(new ResponseModel(QLBHConst.Error, "Lỗi hệ thống khi thêm mới."));

102 │ }

103 │ }

104 │

105 │ [HttpPost]

106 │ [TokenUserCheckHTTP(Permission.Add)] // Should be Permission.Edit

107 │ public ActionResult UpdateAPI(string Ma, string MaKho, string MaHangHoa, string GiaBan, string GhiChu)

108 │ {

109 │ \_logger.LogInformation("Người dùng {UserId} cập nhật Giá bán. Mã (đã mã hóa): {Ma}", objToken...

110 │ if (!Utility.TryDecryptToInt(Ma, out int id))

111 │ {

112 │ \_logger.LogWarning("UpdateAPI thất bại: Mã không hợp lệ: {Ma}", Ma);

113 │ return Json(new ResponseModel(QLBHConst.Error, "Mã không hợp lệ"));

114 │ }

115 │

116 │ var validation = CheckRequireGiaBan(MaKho, MaHangHoa, GiaBan);

117 │ if (validation != null)

118 │ return Json(validation);

119 │

120 │ try

121 │ {

122 │ var result = new GiaBanTheoKhoModel(dbcontext).Edit(

123 │ id,

124 │ Cast.ToInt(MaKho),

125 │ Cast.ToInt(MaHangHoa),

126 │ Cast.ToInt(GiaBan),

127 │ GhiChu,

128 │ objToken.userid,

129 │ objToken.branch);

130 │

131 │ \_logger.LogInformation("Kết quả cập nhật Giá bán ID {Id}: {Result}", id, result);

132 │ return Json(Utility.MakeResponseFromAddResult(Cast.ToString(result)));

133 │ }

134 │ catch (Exception ex)

135 │ {

136 │ \_logger.LogError(ex, "Lỗi khi cập nhật Giá bán ID: {Id}", id);

137 │ return Json(new ResponseModel(QLBHConst.Error, "Lỗi hệ thống khi cập nhật."));

138 │ }

139 │ }

140 │

141 │ [HttpPost]

142 │ [TokenUserCheckHTTP(Permission.Delete)]

143 │ public ActionResult Delete(string Ma)

144 │ {

145 │ \_logger.LogInformation("Người dùng {UserId} yêu cầu xóa Giá bán. Mã (đã mã hóa): {Ma}", objTo...

146 │ if (!Utility.TryDecryptToInt(Ma, out int id))

147 │ {

148 │ \_logger.LogWarning("Delete thất bại: Mã không hợp lệ: {Ma}", Ma);

149 │ return Json(new ResponseModel(QLBHConst.Error, "Mã không hợp lệ"));

150 │ }

151 │

152 │ try

153 │ {

154 │ var result = new GiaBanTheoKhoModel(dbcontext).Delete(id, objToken.userid, objToken.branch);

155 │ \_logger.LogInformation("Kết quả xóa Giá bán ID {Id}: {Result}", id, result);

156 │ return Json(Utility.MakeResponseFromDeleteResult(Cast.ToString(result)));

157 │ }

158 │ catch (Exception ex)

159 │ {

160 │ \_logger.LogError(ex, "Lỗi khi xóa Giá bán ID: {Id}", id);

161 │ return Json(new ResponseModel(QLBHConst.Error, "Lỗi hệ thống khi xóa."));

162 │ }

163 │ }

164 │

165 │ [TokenUserCheckHTTP]

166 │ public ActionResult XuatExcel(string MaKho)

167 │ {

168 │ \_logger.LogInformation("Người dùng {UserId} yêu cầu xuất Excel Giá bán theo kho. Mã kho: {MaKho}...

169 │ try

170 │ {

171 │ var khoId = Cast.ToInt(MaKho);

172 │ var kho = new DanhMucKhoModel(dbcontext).GetItem(khoId);

173 │ if (kho == null)

174 │ {

175 │ \_logger.LogWarning("Xuất Excel thất bại: Không tìm thấy kho với Mã kho: {MaKho}", MaKho...

176 │ return Json(new ResponseModel(QLBHConst.Error, "Không tìm thấy kho"));

177 │ }

178 │

179 │ var listData = (from gb in dbcontext.TblGiaBanTheoKho.AsNoTracking()

180 │ join k in dbcontext.StockDanhMucKho.AsNoTracking() on gb.MaKho equals k.Ma

181 │ join hh in dbcontext.TblHangHoa.AsNoTracking() on gb.MaHangHoa equals hh.Ma

182 │ join dvt in dbcontext.TblDonViTinh.AsNoTracking() on hh.MaDvt equals dvt.Ma

183 │ join lh in dbcontext.TblLopHangHoa.AsNoTracking() on hh.MaLopHang equals lh.Ma

184 │ where !(gb.Deleted ?? false) &&

185 │ gb.MaChiNhanh == objToken.branch &&

186 │ gb.MaKho == khoId

187 │ orderby gb.Ma

188 │ select new

189 │ {

190 │ hh.MaVach,

191 │ hh.MaKt,

192 │ LopHang = lh.Ten,

193 │ hh.Ten,

194 │ TenDvt = dvt.Ten,

195 │ GiaBan = gb.GiaBan, // Giữ kiểu số để định dạng trong Excel

196 │ gb.GhiChu

197 │ }).ToList();

198 │

199 │ \_logger.LogInformation("Tìm thấy {Count} bản ghi để xuất Excel.", listData.Count);

200 │

201 │ var dtb = new DataTable();

202 │ dtb.Columns.Add("STT");

203 │ dtb.Columns.Add("Lớp hàng");

204 │ dtb.Columns.Add("Mã vạch");

205 │ dtb.Columns.Add("Mã KT");

206 │ dtb.Columns.Add("Tên hàng hóa");

207 │ dtb.Columns.Add("ĐVT");

208 │ dtb.Columns.Add("Giá bán", typeof(decimal));

209 │ dtb.Columns.Add("Ghi chú");

210 │

211 │ int index = 1;

212 │ foreach (var item in listData)

213 │ {

214 │ dtb.Rows.Add(index++, item.LopHang, item.MaVach, item.MaKt, item.Ten, item.TenDvt, item.GiaBan, item...

215 │ }

216 │

217 │ string filePath = Path.Combine(\_hostingEnvironment.ContentRootPath, "wwwroot", "templates", "GiaBanTheoK...

218 │ string fileName = $"GiaBanTheoKho\_{kho.Ten.Replace(" ", "\_")}\_{DateTime.Now:yyyyMMdd}.xlsx";

219 │ string contentType = "application/vnd.openxmlformats-officedocument.spreadsheetml.sheet";

220 │

221 │ var customValues = new List<ExcelValue>

222 │ {

223 │ new() { Range = $"A2:{ExportToExcel.GetExcelColumnName(dtb.Columns.Count)}2", Value = $"{kho.Ten}" }

224 │ };

225 │

226 │ var stream = ExportToExcel.FromTemplate(filePath, "Excel", customValues, dtb, true, 5, "A", ExportToExce...

227 │

228 │ \_logger.LogInformation("Xuất Excel thành công.");

229 │ return new FileStreamResult(stream, contentType) { FileDownloadName = fileName };

230 │ }

231 │ catch (Exception ex)

232 │ {

233 │ \_logger.LogError(ex, "Lỗi nghiêm trọng trong quá trình xuất Excel cho kho: {MaKho}", MaKho);

234 │ return Json(new ResponseModel(QLBHConst.Error, "Đã có lỗi xảy ra trong quá trình tạo file Exc...

235 │ }

236 │ }

237 │

238 │ private void LoadViewDataLists(string MaKho)

239 │ {

240 │ \_logger.LogDebug("Bắt đầu tải ViewData lists cho kho: {MaKho}", MaKho);

241 │ ViewData["ListKho"] = new DanhMucKhoModel(dbcontext).GetData(objToken.branch);

242 │ ViewData["ListHH"] = new TblHangHoaModel(dbcontext).GetData(objToken.branch);

243 │ ViewData["ListDVT"] = new DonViTinhModel(dbcontext).GetData();

244 │ ViewData["ListLopHang"] = new LopHangHoaModel(dbcontext).GetData(objToken.branch);

245 │ ViewData["ListData"] = new GiaBanTheoKhoModel(dbcontext).GetGiaBanTheoKho(Cast.ToInt(MaKho), objToken.branch...

246 │ ViewData["MaKho"] = MaKho;

247 │ \_logger.LogDebug("Tải xong ViewData lists.");

248 │ }

249 │

250 │ private static ResponseModel CheckRequireGiaBan(string MaKho, string MaHangHoa, string GiaBan)

251 │ {

252 │ if (string.IsNullOrWhiteSpace(MaKho))

253 │ return new ResponseModel(QLBHConst.Error, "Vui lòng chọn kho");

254 │ if (string.IsNullOrWhiteSpace(MaHangHoa))

255 │ return new ResponseModel(QLBHConst.Error, "Vui lòng chọn hàng hoá");

256 │ if (string.IsNullOrWhiteSpace(GiaBan))

257 │ return new ResponseModel(QLBHConst.Error, "Vui lòng nhập giá bán");

258 │ return null;

259 │ }

260 │ }

261 │ }

────────────────────────────────────────────────────────────

**📄 GiaBanTheoKhoModel.cs (CS, 265 lines)**

1 │ ﻿using Microsoft.EntityFrameworkCore;

2 │ using QuanLyBanHang.Common;

3 │ using QuanLyBanHang.Models.QLBHEntity;

4 │ using QuanLyBanHang.Models.ViewModels;

5 │

6 │ namespace QuanLyBanHang.Models.ExModel.CAT

7 │ {

8 │ public class GiaBanTheoKhoModel : Base<TblGiaBanTheoKho>

9 │ {

10 │ public GiaBanTheoKhoModel(DbContext dbContext) : base(dbContext)

11 │ {

12 │ }

13 │

14 │ QuanLyBanHangContext dbContext

15 │ {

16 │ get { return \_dbContext as QuanLyBanHangContext; }

17 │ set { return; }

18 │ }

19 │

20 │ public TblGiaBanTheoKho? GetItem(int ma, Guid maChiNhanh)

21 │ {

22 │ return dbContext.TblGiaBanTheoKho

23 │ .AsNoTracking()

24 │ .FirstOrDefault(t => t.Ma == ma && !(t.Deleted ?? false) && t.MaChiNhanh == maChiNhanh);

25 │ }

26 │

27 │ public TblGiaBanTheoKho? GetItem(int maKho, int maHangHoa, Guid maChiNhanh)

28 │ {

29 │ return dbContext.TblGiaBanTheoKho

30 │ .AsNoTracking()

31 │ .FirstOrDefault(t => t.MaKho == maKho && t.MaHangHoa == maHangHoa && !(t.Deleted ?? false) && t.MaChiNha...

32 │ }

33 │

34 │ public List<TblGiaBanTheoKho> GetData(Guid maChiNhanh)

35 │ {

36 │ return dbContext.TblGiaBanTheoKho

37 │ .AsNoTracking()

38 │ .Where(t => !(t.Deleted ?? false) && t.MaChiNhanh == maChiNhanh)

39 │ .OrderBy(t => t.Ma)

40 │ .ToList();

41 │ }

42 │

43 │ public List<TblGiaBanTheoKho> GetGiaBanTheoKho(int maKho, Guid maChiNhanh)

44 │ {

45 │ return dbContext.TblGiaBanTheoKho

46 │ .AsNoTracking()

47 │ .Where(t => t.MaKho == maKho && !(t.Deleted ?? false) && t.MaChiNhanh == maChiNhanh)

48 │ .OrderBy(t => t.Ma)

49 │ .ToList();

50 │ }

51 │

52 │ public List<TblGiaBanTheoKho> GetAllHangHoaTheoKho(int maHangHoa, Guid maChiNhanh)

53 │ {

54 │ return dbContext.TblGiaBanTheoKho

55 │ .AsNoTracking()

56 │ .Where(t => t.MaHangHoa == maHangHoa && !(t.Deleted ?? false) && t.MaChiNhanh == maChiNhanh)

57 │ .OrderBy(t => t.Ma)

58 │ .ToList();

59 │ }

60 │ public List<TonKhoVaGiaBanTheoKhoModel> DanhSachTonKhoHangHoaVaGiaBanTheoKho(int MaKho, string NamThang, Guid Ma...

61 │ {

62 │ var query = from gia in dbContext.TblGiaBanTheoKho.AsNoTracking().Where(t=>t.MaKho == MaKho)

63 │ join hh in dbContext.TblHangHoa.AsNoTracking().Where(x=> (x.Status ?? false)) on gia.MaHangHoa e...

64 │ join lh in dbContext.TblLopHangHoa.AsNoTracking() on hh.MaLopHang equals lh.Ma

65 │ join tk in dbContext.StockTonKhoBanHang.AsNoTracking()

66 │ .Where(t => t.NamThang == NamThang && t.MaChiNhanh == MaChiNhanh && t.MaKho == MaKho)

67 │ on gia.MaHangHoa equals tk.MaHangHoa into \_tk

68 │ from tk in \_tk.DefaultIfEmpty()

69 │ join dv in dbContext.TblDonViTinh.AsNoTracking() on hh.MaDvt equals dv.Ma into \_dv

70 │ from dv in \_dv.DefaultIfEmpty()

71 │ where

72 │ !(hh.Deleted ?? false) &&

73 │ !(hh.HetHang ?? false) &&

74 │ !(lh.Deleted ?? false) &&

75 │ (lh.Status ?? false) &&

76 │ !(dv.Deleted ?? false) &&

77 │ (NoiBo || !(hh.BanNoiBo ?? false)) &&

78 │ ((lh.ChoPhepBanAm ?? false) || (tk != null && (tk.SltonCuoiKy ?? 0) > 0))

79 │

80 │ select new TonKhoVaGiaBanTheoKhoModel

81 │ {

82 │ Ma = gia.Ma,

83 │ MaVach = hh.MaVach,

84 │ TenHang = hh.Ten,

85 │ GiaBan = gia.GiaBan ?? 0,

86 │ DVT = hh.MaDvt ?? 0,

87 │ LopHang = hh.MaLopHang ?? 0,

88 │ TenLop = lh.Ten ?? "",

89 │ MaKho = gia.MaKho ?? 0,

90 │ TenKho = "",

91 │ TenDVT = dv.Ten ?? "",

92 │ SLTon = tk == null ? 0 : Convert.ToInt64(tk.SltonCuoiKy ?? 0),

93 │ HinhAnh = hh.HinhAnh ?? "",

94 │ BanNoiBo = hh.BanNoiBo ?? false,

95 │ MaHangHoa = hh.Ma,

96 │ LoaiTieuDung = lh.MaLoaiTieuDung ?? 0

97 │ };

98 │

99 │ var listData = query.ToList();

100 │

101 │ return listData;

102 │ }

103 │

104 │ public List<TonKhoVaGiaBanTheoKhoModel> DanhSachThucAnVaThucUong(string NamThang, Guid MaChiNhanh, bool NoiBo = ...

105 │ {

106 │ var currentHour = DateTime.Now.TimeOfDay.TotalHours;

107 │

108 │ var query = (from gia in dbContext.TblGiaBanTheoKho.AsNoTracking()

109 │ join hh in dbContext.TblHangHoa.AsNoTracking() on gia.MaHangHoa equals hh.Ma

110 │ join lh in dbContext.TblLopHangHoa.AsNoTracking() on hh.MaLopHang equals lh.Ma

111 │ join kho in dbContext.StockDanhMucKho.AsNoTracking() on gia.MaKho equals kho.Ma

112 │ join tk in dbContext.StockTonKhoBanHang.AsNoTracking()

113 │ .Where(t => t.NamThang == NamThang && t.MaChiNhanh == MaChiNhanh)

114 │ on new { MaHangHoa = hh.Ma, MaKho = kho.Ma } equals new { tk.MaHangHoa, tk.MaKho } in...

115 │ from tk in \_tk.DefaultIfEmpty()

116 │ join dv in dbContext.TblDonViTinh.AsNoTracking() on hh.MaDvt equals dv.Ma into \_dv

117 │ from dv in \_dv.DefaultIfEmpty()

118 │ where

119 │ gia.MaChiNhanh == MaChiNhanh &&

120 │ hh.MaChiNhanh == MaChiNhanh &&

121 │ lh.MaChiNhanh == MaChiNhanh &&

122 │ !(hh.Deleted ?? false) &&

123 │ !(hh.HetHang ?? false) &&

124 │ !(lh.Deleted ?? false) &&

125 │ (lh.Status ?? false) &&

126 │ !(kho.Deleted ?? false) &&

127 │ (kho.ThoiGianBatDau ?? 0) <= currentHour && (kho.ThoiGianKetThuc ?? 23) >= currentHour &&

128 │ (lh.ThoiGianBatDau ?? 0) <= currentHour && (lh.ThoiGianKetThuc ?? 23) >= currentHour &&

129 │ lh.MaLoaiTieuDung != (int)LoaiTieuDung.Khac &&

130 │ (NoiBo || !(hh.BanNoiBo ?? false)) &&

131 │ (kho.HienThiTrongWebOrder ?? false) &&

132 │ ((lh.ChoPhepBanAm ?? false) || (tk != null && (tk.SltonCuoiKy ?? 0) > 0))

133 │ select new TonKhoVaGiaBanTheoKhoModel

134 │ {

135 │ Ma = gia.Ma,

136 │ MaVach = hh.MaVach,

137 │ TenHang = hh.Ten,

138 │ GiaBan = gia.GiaBan ?? 0,

139 │ DVT = hh.MaDvt ?? 0,

140 │ LopHang = hh.MaLopHang ?? 0,

141 │ TenLop = lh.Ten,

142 │ MaKho = gia.MaKho ?? 0,

143 │ TenKho = kho.Ten,

144 │ TenDVT = dv.Ten ?? "",

145 │ SLTon = tk == null ? 0 : Convert.ToInt64(tk.SltonCuoiKy ?? 0),

146 │ HinhAnh = hh.HinhAnh ?? "",

147 │ BanNoiBo = hh.BanNoiBo ?? false,

148 │ MaHangHoa = hh.Ma,

149 │ LoaiTieuDung = lh.MaLoaiTieuDung ?? 0

150 │ }).ToList();

151 │

152 │

153 │ return query.ToList();

154 │ }

155 │

156 │ public List<TonKhoVaGiaBanTheoKhoModel\_AllStock> DanhSachTatCaTonKhoHangHoaVaGiaBanTheoKho(int maKho, string nam...

157 │ {

158 │ var listParentId = dbContext.VHangHoa.AsNoTracking().Where(x => x.MaChiNhanh == maChiNhanh).Select(x => x.Pa...

159 │ var listData = (from gia in dbContext.TblGiaBanTheoKho.AsNoTracking()

160 │ join hh in dbContext.TblHangHoa.AsNoTracking().Where(t => !listParentId.Contains(t.Ma))

161 │ on gia.MaHangHoa equals hh.Ma

162 │ join lh in dbContext.TblLopHangHoa.AsNoTracking()

163 │ on hh.MaLopHang equals lh.Ma

164 │ join tk in dbContext.StockTonKhoBanHang.AsNoTracking()

165 │ .Where(t => t.NamThang == namThang && t.MaKho == maKho && t.MaChiNhanh == maChiNhanh)

166 │ on gia.MaHangHoa equals tk.MaHangHoa into g\_tk

167 │ from tk in g\_tk.DefaultIfEmpty()

168 │ join dv in dbContext.TblDonViTinh.AsNoTracking()

169 │ on hh.MaDvt equals dv.Ma into g\_dv

170 │ from dv in g\_dv.DefaultIfEmpty()

171 │ where gia.MaKho == maKho &&

172 │ gia.MaChiNhanh == maChiNhanh &&

173 │ !(gia.Deleted ?? false) &&

174 │ !(hh.Deleted ?? false) &&

175 │ !(hh.HetHang ?? false) &&

176 │ !(lh.Deleted ?? false) &&

177 │ (lh.Status ?? false)

178 │ select new TonKhoVaGiaBanTheoKhoModel\_AllStock

179 │ {

180 │ Ma = gia.Ma,

181 │ MaVach = hh.MaVach,

182 │ MaHangHoa = hh.Ma,

183 │ TenHang = hh.Ten,

184 │ GiaBan = gia.GiaBan ?? 0,

185 │ DVT = hh.MaDvt ?? 0,

186 │ LopHang = hh.MaLopHang ?? 0,

187 │ TenLop = lh.Ten,

188 │ MaKho = gia.MaKho ?? 0,

189 │ TenKho = "",

190 │ TenDVT = dv.Ten,

191 │ SLTon = tk.SltonCuoiKy ?? 0,

192 │ HinhAnh = hh.HinhAnh ?? ""

193 │ }).ToList();

194 │

195 │ return listData;

196 │ }

197 │

198 │ public int Add(int maKho, int maHangHoa, int giaBan, string ghiChu, Guid userID, Guid maChiNhanh)

199 │ {

200 │ if (CheckExists(0, maHangHoa, maKho, maChiNhanh))

201 │ return -3;

202 │ var br = dbContext.SysBranch.FirstOrDefault(x => x.Ma == maChiNhanh);

203 │

204 │ var itemAdd = new TblGiaBanTheoKho

205 │ {

206 │ MaChiNhanh = maChiNhanh,

207 │ MaChiNhanhInt = br.Nid ?? 0,

208 │ MaKho = maKho,

209 │ MaHangHoa = maHangHoa,

210 │ GiaBan = giaBan,

211 │ GhiChu = ghiChu,

212 │ CreatedDate = DateTime.Now,

213 │ UserCreated = userID

214 │ };

215 │

216 │ dbContext.TblGiaBanTheoKho.Add(itemAdd);

217 │ return dbContext.SaveChanges() > 0 ? 1 : 0;

218 │ }

219 │

220 │ public int Edit(int ma, int maKho, int maHangHoa, int giaBan, string ghiChu, Guid userID, Guid maChiNhanh)

221 │ {

222 │ if (CheckExists(ma, maHangHoa, maKho, maChiNhanh))

223 │ return -3;

224 │

225 │ var item = dbContext.TblGiaBanTheoKho.FirstOrDefault(x =>

226 │ x.Ma == ma && !(x.Deleted ?? false) && x.MaChiNhanh == maChiNhanh);

227 │

228 │ if (item == null)

229 │ return -2;

230 │

231 │ item.MaKho = maKho;

232 │ item.MaHangHoa = maHangHoa;

233 │ item.GiaBan = giaBan;

234 │ item.GhiChu = ghiChu;

235 │ item.ModifiedDate = DateTime.Now;

236 │ item.UserModified = userID;

237 │

238 │ dbContext.TblGiaBanTheoKho.Update(item);

239 │ return dbContext.SaveChanges() > 0 ? 1 : 0;

240 │ }

241 │

242 │ public int Delete(int ma, Guid userID, Guid maChiNhanh)

243 │ {

244 │ var item = dbContext.TblGiaBanTheoKho.FirstOrDefault(x =>

245 │ x.Ma == ma && !(x.Deleted ?? false) && x.MaChiNhanh == maChiNhanh);

246 │

247 │ if (item == null)

248 │ return -2;

249 │

250 │ item.Deleted = true;

251 │ item.DeletedDate = DateTime.Now;

252 │ item.UserDeleted = userID;

253 │

254 │ dbContext.TblGiaBanTheoKho.Update(item);

255 │ return dbContext.SaveChanges() > 0 ? 1 : 0;

256 │ }

257 │

258 │ private bool CheckExists(int ma, int maHangHoa, int maKho, Guid maChiNhanh)

259 │ {

260 │ return dbContext.TblGiaBanTheoKho

261 │ .Any(t => t.Ma != ma && t.MaHangHoa == maHangHoa && t.MaKho == maKho &&

262 │ !(t.Deleted ?? false) && t.MaChiNhanh == maChiNhanh);

263 │ }

264 │ }

265 │ }

────────────────────────────────────────────────────────────

**📄 GiamGia.cs (CS, 33 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class GiamGia

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int? Giatri { get; set; }

11 │

12 │ public string? TenCt { get; set; }

13 │

14 │ public DateTime? TuNgay { get; set; }

15 │

16 │ public DateTime? DenNgay { get; set; }

17 │

18 │ public bool? TrangThai { get; set; }

19 │

20 │ public DateTime? NgayTao { get; set; }

21 │

22 │ public int? NguoiTao { get; set; }

23 │

24 │ public DateTime? NgaySua { get; set; }

25 │

26 │ public int? NguoiSua { get; set; }

27 │

28 │ public bool? DaXoa { get; set; }

29 │

30 │ public int? NguoiXoa { get; set; }

31 │

32 │ public DateTime? NgayXoa { get; set; }

33 │ }

────────────────────────────────────────────────────────────

**📄 GiamGiaController.cs (CS, 120 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using QuanLyBanHang.Common;

3 │ using QuanLyBanHang.Common.Attributes;

4 │ using QuanLyBanHang.Models.APIModels;

5 │ using QuanLyBanHang.Models.ExModel.CAT;

6 │ using QuanLyBanHang.Models.QLBHEntity;

7 │

8 │ namespace QuanLyBanHang.Controllers.OPS

9 │ {

10 │ public class GiamGiaController : BaseController

11 │ {

12 │ public GiamGiaController(QuanLyBanHangContext dbcontext, MemoryCacheManager memoryCache) : base(dbcontext, memor...

13 │ {

14 │ m\_MenuID = MenuID.opsGiamGia.GetEnumGuid();

15 │ }

16 │

17 │ [TokenUserCheckHTTP]

18 │ public IActionResult Index() => View();

19 │

20 │ [TokenUserCheckHTTP(Permission.Add)]

21 │ public IActionResult Add() => View("Form");

22 │

23 │ [TokenUserCheckHTTP(Permission.Edit)]

24 │ public IActionResult Edit(string Ma)

25 │ {

26 │ if (!Utility.TryDecryptToInt(Ma, out int id))

27 │ return BadRequest("Mã không hợp lệ");

28 │

29 │ var item = new GiamGiaModel(dbcontext).GetItem(id);

30 │ ViewData["item"] = item;

31 │ return View("Form");

32 │ }

33 │

34 │ [HttpPost]

35 │ [TokenUserCheckHTTP(Permission.Add)]

36 │ public IActionResult AddAPI(string Ten, string GiaTri, string TuNgay, string DenNgay, string Status)

37 │ {

38 │ var validation = CheckRequire(Ten, GiaTri, TuNgay, DenNgay, Status);

39 │ if (validation != null) return Json(validation);

40 │

41 │ if (!DateTime.TryParse(TuNgay, out DateTime tuNgay) || !DateTime.TryParse(DenNgay, out DateTime denNgay))

42 │ return Json(new ResponseModel(QLBHConst.Error, "Định dạng ngày không hợp lệ"));

43 │

44 │ int giaTri = Cast.ToInt(GiaTri);

45 │ bool trangThai = Status == "1";

46 │

47 │ int result = new GiamGiaModel(dbcontext).Add(Ten, giaTri, tuNgay.Date, denNgay.Date, trangThai, objToken.use...

48 │

49 │ return Json(Utility.MakeResponseFromAddResult(result.ToString()));

50 │ }

51 │

52 │ [HttpPost]

53 │ [TokenUserCheckHTTP(Permission.Edit)]

54 │ public IActionResult UpdateAPI(string Ma, string Ten, string GiaTri, string TuNgay, string DenNgay, string Statu...

55 │ {

56 │ if (!Utility.TryDecryptToInt(Ma, out int id))

57 │ return Json(new ResponseModel(QLBHConst.Error, "Mã không hợp lệ"));

58 │

59 │ var validation = CheckRequire(Ten, GiaTri, TuNgay, DenNgay, Status);

60 │ if (validation != null) return Json(validation);

61 │

62 │ if (!DateTime.TryParse(TuNgay, out DateTime tuNgay) || !DateTime.TryParse(DenNgay, out DateTime denNgay))

63 │ return Json(new ResponseModel(QLBHConst.Error, "Định dạng ngày không hợp lệ"));

64 │

65 │ int giaTri = Cast.ToInt(GiaTri);

66 │ bool trangThai = Status == "1";

67 │

68 │ int result = new GiamGiaModel(dbcontext).Edit(id, Ten, giaTri, tuNgay.Date, denNgay.Date, trangThai, objToke...

69 │

70 │ return Json(Utility.MakeResponseFromEditResult(result.ToString()));

71 │ }

72 │

73 │ [HttpPost]

74 │ [TokenUserCheckHTTP(Permission.Delete)]

75 │ public IActionResult DeleteAPI(string Ma)

76 │ {

77 │ if (!Utility.TryDecryptToInt(Ma, out int id))

78 │ return Json(new ResponseModel(QLBHConst.Error, "Mã không hợp lệ"));

79 │

80 │ int result = new GiamGiaModel(dbcontext).Delete(id, objToken.userid);

81 │ return Json(Utility.MakeResponseFromDeleteResult(result.ToString()));

82 │ }

83 │

84 │ [TokenUserCheckHTTP(Permission.NotCheck)]

85 │ public IActionResult getListData()

86 │ {

87 │ var listData = new GiamGiaModel(dbcontext).GetData(objToken.branch)

88 │ .Select(n => new

89 │ {

90 │ maEncrypt = n.Ma.Encrypt(),

91 │ ma = n.Ma,

92 │ ten = n.Ten,

93 │ giatri = n.Giatri,

94 │ tungay = n.TuNgay?.ToString("dd/MM/yyyy"),

95 │ denngay = n.DenNgay?.ToString("dd/MM/yyyy"),

96 │ trangthai = n.Status ?? false,

97 │ tentrangthai = (n.Status ?? false) ? "Hoạt động" : "Hết hoạt động"

98 │ })

99 │ .ToList();

100 │

101 │ return Json(new ResponseModel(QLBHConst.OK, "", listData));

102 │ }

103 │

104 │ private static ResponseModel CheckRequire(string ten, string giaTri, string tuNgay, string denNgay, string statu...

105 │ {

106 │ if (string.IsNullOrWhiteSpace(ten))

107 │ return new ResponseModel(QLBHConst.Error, "Vui lòng nhập tên");

108 │ if (string.IsNullOrWhiteSpace(giaTri))

109 │ return new ResponseModel(QLBHConst.Error, "Vui lòng nhập giá trị giảm");

110 │ if (string.IsNullOrWhiteSpace(tuNgay))

111 │ return new ResponseModel(QLBHConst.Error, "Vui lòng nhập từ ngày");

112 │ if (string.IsNullOrWhiteSpace(denNgay))

113 │ return new ResponseModel(QLBHConst.Error, "Vui lòng nhập đến ngày");

114 │ if (string.IsNullOrWhiteSpace(status))

115 │ return new ResponseModel(QLBHConst.Error, "Vui lòng chọn trạng thái");

116 │

117 │ return null;

118 │ }

119 │ }

120 │ }

────────────────────────────────────────────────────────────

**📄 GiamGiaModel.cs (CS, 91 lines)**

1 │ ﻿using Microsoft.EntityFrameworkCore;

2 │ using QuanLyBanHang.Models.QLBHEntity;

3 │

4 │ namespace QuanLyBanHang.Models.ExModel.CAT

5 │ {

6 │ public class GiamGiaModel : Base<TblGiamGia>

7 │ {

8 │ private QuanLyBanHangContext Context => \_dbContext as QuanLyBanHangContext;

9 │

10 │ public GiamGiaModel(DbContext dbContext) : base(dbContext) { }

11 │

12 │ public TblGiamGia GetItem(int ma)

13 │ {

14 │ return Context.TblGiamGia

15 │ .AsNoTracking()

16 │ .FirstOrDefault(t => t.Ma == ma && !(t.Deleted ?? false));

17 │ }

18 │

19 │ public List<TblGiamGia> GetData(Guid maChiNhanh)

20 │ {

21 │ return Context.TblGiamGia

22 │ .AsNoTracking()

23 │ .Where(t => !(t.Deleted ?? false) && t.MaChiNhanh == maChiNhanh)

24 │ .OrderBy(t => t.CreatedDate)

25 │ .ToList();

26 │ }

27 │

28 │ public int Add(string ten, int giaTri, DateTime tuNgay, DateTime denNgay, bool trangThai, Guid userID, Guid maCh...

29 │ {

30 │ if (CheckExists(0, ten))

31 │ return -3;

32 │

33 │ var item = new TblGiamGia

34 │ {

35 │ MaChiNhanh = maChiNhanh,

36 │ Ten = ten,

37 │ TuNgay = tuNgay,

38 │ DenNgay = denNgay,

39 │ Giatri = giaTri,

40 │ Status = trangThai,

41 │ UserCreated = userID,

42 │ CreatedDate = DateTime.Now

43 │ };

44 │

45 │ Context.TblGiamGia.Add(item);

46 │ return Context.SaveChanges() > 0 ? 1 : 0;

47 │ }

48 │

49 │ public int Edit(int ma, string ten, int giaTri, DateTime tuNgay, DateTime denNgay, bool trangThai, Guid userID)

50 │ {

51 │ if (CheckExists(ma, ten))

52 │ return -3;

53 │

54 │ var item = Context.TblGiamGia.FirstOrDefault(t => t.Ma == ma && !(t.Deleted ?? false));

55 │ if (item == null)

56 │ return -2;

57 │

58 │ item.Ten = ten;

59 │ item.TuNgay = tuNgay;

60 │ item.DenNgay = denNgay;

61 │ item.Giatri = giaTri;

62 │ item.Status = trangThai;

63 │ item.UserModified = userID;

64 │ item.ModifiedDate = DateTime.Now;

65 │

66 │ Context.TblGiamGia.Update(item);

67 │ return Context.SaveChanges() > 0 ? 1 : 0;

68 │ }

69 │

70 │ public int Delete(int ma, Guid userID)

71 │ {

72 │ var item = Context.TblGiamGia.FirstOrDefault(t => t.Ma == ma && !(t.Deleted ?? false));

73 │ if (item == null)

74 │ return -2;

75 │

76 │ item.Deleted = true;

77 │ item.DeletedDate = DateTime.Now;

78 │ item.UserDeleted = userID;

79 │

80 │ Context.TblGiamGia.Update(item);

81 │ return Context.SaveChanges() > 0 ? 1 : 0;

82 │ }

83 │

84 │ private bool CheckExists(int ma, string ten)

85 │ {

86 │ return Context.TblGiamGia

87 │ .AsNoTracking()

88 │ .Any(t => t.Ma != ma && t.Ten == ten && !(t.Deleted ?? false));

89 │ }

90 │ }

91 │ }

────────────────────────────────────────────────────────────

**📄 HangHoa.cs (CS, 47 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class HangHoa

7 │ {

8 │ public int HangHoaId { get; set; }

9 │

10 │ public string? MaKt { get; set; }

11 │

12 │ public string MaVach { get; set; } = null!;

13 │

14 │ public int LopHang { get; set; }

15 │

16 │ public string TenHang { get; set; } = null!;

17 │

18 │ public int Dvt { get; set; }

19 │

20 │ public int? GiaBan { get; set; }

21 │

22 │ public decimal? GiaVon { get; set; }

23 │

24 │ public string? HinhAnh { get; set; }

25 │

26 │ public string? GhiChu { get; set; }

27 │

28 │ public DateTime? NgayTao { get; set; }

29 │

30 │ public int NguoiTao { get; set; }

31 │

32 │ public bool? TrangThai { get; set; }

33 │

34 │ public DateTime? NgaySua { get; set; }

35 │

36 │ public int? NguoiSua { get; set; }

37 │

38 │ public bool? DaXoa { get; set; }

39 │

40 │ public int? NguoiXoa { get; set; }

41 │

42 │ public DateTime? NgayXoa { get; set; }

43 │

44 │ public bool? BanNoiBo { get; set; }

45 │

46 │ public bool? HetHang { get; set; }

47 │ }

────────────────────────────────────────────────────────────

**📄 HangHoaController.cs (CS, 504 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using Microsoft.EntityFrameworkCore;

3 │ using QuanLyBanHang.Common;

4 │ using QuanLyBanHang.Common.Attributes;

5 │ using QuanLyBanHang.Models.APIModels;

6 │ using QuanLyBanHang.Models.ExModel.CAT;

7 │ using QuanLyBanHang.Models.QLBHEntity;

8 │ using System.Data;

9 │ using static iText.StyledXmlParser.Jsoup.Select.Evaluator;

10 │

11 │ namespace QuanLyBanHang.Controllers.CAT

12 │ {

13 │ public class HangHoaController : BaseController

14 │ {

15 │ private readonly IWebHostEnvironment \_hostingEnvironment;

16 │ private readonly ILogger<HangHoaController> \_logger;

17 │

18 │ public HangHoaController(QuanLyBanHangContext dbcontext, IWebHostEnvironment hostingEnvironment, MemoryCacheMana...

19 │ {

20 │ m\_MenuID = MenuID.catHangHoa.GetEnumGuid();

21 │ \_hostingEnvironment = hostingEnvironment;

22 │ \_logger = logger;

23 │ }

24 │ [TokenUserCheckHTTP]

25 │ public IActionResult Index()

26 │ {

27 │ \_logger.LogInformation("Người dùng {UserId} truy cập trang danh sách Hàng hóa.", objToken.userid);

28 │ return View();

29 │ }

30 │

31 │ [HttpPost]

32 │ public ActionResult generateCode()

33 │ {

34 │ \_logger.LogInformation("Người dùng yêu cầu tạo mã vạch mới.");

35 │ try

36 │ {

37 │ string code;

38 │ do { code = Utility.RandomStringNumber(13); }

39 │ while (dbcontext.TblHangHoa.Any(t => t.MaVach == code));

40 │

41 │ \_logger.LogInformation("Đã tạo thành công mã vạch mới: {MaVach}", code);

42 │ return Json(new ResponseModel(QLBHConst.OK, code));

43 │ }

44 │ catch (Exception ex)

45 │ {

46 │ \_logger.LogError(ex, "Lỗi khi tạo mã vạch tự động.");

47 │ return Json(new ResponseModel(QLBHConst.Error, "Lỗi hệ thống khi tạo mã vạch."));

48 │ }

49 │ }

50 │

51 │ [TokenUserCheckHTTP(Permission.Add)]

52 │ public ActionResult Add()

53 │ {

54 │ \_logger.LogInformation("Người dùng {UserId} truy cập trang thêm mới Hàng hóa.", objToken.userid);

55 │ try

56 │ {

57 │ ViewData["ListDinhMuc"] = dbcontext.VtthDinhMuc.Where(x => !(x.Deleted ?? false)).ToList();

58 │ ViewData["ListData"] = GetListDataParent(objToken.branch);

59 │ ViewData["ListDVT"] = dbcontext.TblDonViTinh.Where(x => !(x.Deleted ?? false)).ToList();

60 │ ViewData["ListLopHang"] = dbcontext.TblLopHangHoa.Where(x => !(x.Deleted ?? false)).ToList();

61 │ ViewData["ListKho"] = new DanhMucKhoModel(dbcontext).GetDataByUser(objToken.userid, objToken.branch);

62 │ return View("Form");

63 │ }

64 │ catch (Exception ex)

65 │ {

66 │ \_logger.LogError(ex, "Lỗi khi tải dữ liệu cho trang thêm mới Hàng hóa.");

67 │ return View("Index");

68 │ }

69 │ }

70 │

71 │ [TokenUserCheckHTTP(Permission.Edit)]

72 │ public ActionResult Edit(string Ma)

73 │ {

74 │ \_logger.LogInformation("Người dùng {UserId} yêu cầu chỉnh sửa Hàng hóa. Mã (đã mã hóa): {M...

75 │ try

76 │ {

77 │ Ma = Ma.Decrypt();

78 │ int id = Ma.ToInt();

79 │ var kho = new DanhMucKhoModel(dbcontext).GetDataByUser(objToken.userid, objToken.branch);

80 │

81 │ ViewData["ListDinhMuc"] = dbcontext.VtthDinhMuc.Where(x => !(x.Deleted ?? false)).ToList();

82 │ ViewData["ListData"] = GetListDataParent(objToken.branch);

83 │ ViewData["ListDVT"] = dbcontext.TblDonViTinh.Where(x => !(x.Deleted ?? false)).ToList();

84 │ ViewData["ListLopHang"] = dbcontext.TblLopHangHoa.Where(x => !(x.Deleted ?? false)).ToList();

85 │ ViewData["ListKho"] = kho;

86 │ ViewData["item"] = dbcontext.TblHangHoa.FirstOrDefault(x => !(x.Deleted ?? false) && x.Ma == id);

87 │ ViewData["ListGiaBanTaiKho"] = new GiaBanTheoKhoModel(dbcontext).GetAllHangHoaTheoKho(id, objToken.branc...

88 │

89 │ return View("Form");

90 │ }

91 │ catch (Exception ex)

92 │ {

93 │ \_logger.LogError(ex, "Lỗi khi tải dữ liệu để chỉnh sửa Hàng hóa. Mã (đã mã hóa): {...

94 │ return RedirectToAction("Index");

95 │ }

96 │ }

97 │

98 │ [TokenUserCheckHTTP]

99 │ public ActionResult GetListData(string FileType)

100 │ {

101 │ \_logger.LogInformation("Người dùng {UserId} yêu cầu lấy danh sách Hàng hóa. Loại file: {FileTy...

102 │

103 │ Guid MaChiNhanh = objToken.branch;

104 │ var rawData = (from h in dbcontext.TblHangHoa.Where(t => !(t.Deleted ?? false) && t.MaChiNhanh == MaChiNhanh...

105 │ join d in dbcontext.TblDonViTinh.Where(t => !(t.Deleted ?? false)) on h.MaDvt equals d.Ma int...

106 │ from d in tempDVT.DefaultIfEmpty()

107 │ join l in dbcontext.TblLopHangHoa.Where(t => !(t.Deleted ?? false) && t.MaChiNhanh == MaChiNh...

108 │ from l in tempLop.DefaultIfEmpty()

109 │ join hh in dbcontext.TblHangHoa.Where(t => !(t.Deleted ?? false)) on h.ParentId equals hh.Ma ...

110 │ from hh in tempHH.DefaultIfEmpty()

111 │ select new

112 │ {

113 │ h,

114 │ d,

115 │ l,

116 │ hh

117 │ }).ToList();

118 │

119 │ var listData = rawData.Select(x => new

120 │ {

121 │ maEncrypt = x.h.Ma.ToString().Encrypt(),

122 │ ma = x.h.Ma + "",

123 │ maKt = x.h.MaKt + "",

124 │ maVach = x.h.MaVach + "",

125 │ tenHang = x.h.Ten ?? "",

126 │ giaBan = x.h.GiaBan ?? 0,

127 │ giaVon = x.h.GiaVon ?? 0,

128 │ donViTinh = x.d?.Ten ?? "",

129 │ maLopHang = x.l?.Ma,

130 │ tenLopHang = x.l?.Ten ?? "",

131 │ banNoiBo = x.h.BanNoiBo,

132 │ hetHang = x.h.HetHang,

133 │ vatTu = x.h.VatTu,

134 │ hinhAnh = x.h.HinhAnh ?? "",

135 │ trangThai = x.h.Status ?? false,

136 │ giaBanFormat = string.Format("{0:#,##0.00}", x.h.GiaBan ?? 0),

137 │ tenParent = x.hh?.Ten ?? ""

138 │ }).ToList();

139 │

140 │ \_logger.LogInformation("Lấy thành công {Count} bản ghi Hàng hóa.", listData.Count);

141 │

142 │ if (FileType == null)

143 │ {

144 │ return Json(new ResponseModelSuccess("", listData));

145 │ }

146 │

147 │ DataTable dtb = new();

148 │ dtb.Columns.Add("STT");

149 │ dtb.Columns.Add("Mã kế toán");

150 │ dtb.Columns.Add("Mã vạch");

151 │ dtb.Columns.Add("Lớp hàng");

152 │ dtb.Columns.Add("Tên hàng");

153 │ dtb.Columns.Add("ĐVT");

154 │ dtb.Columns.Add("Giá bán", typeof(decimal));

155 │

156 │ int index = 0;

157 │

158 │ foreach (var item in listData)

159 │ {

160 │ index++;

161 │ dtb.Rows.Add(new object[]

162 │ {

163 │ index,

164 │ item.maKt,

165 │ item.maVach,

166 │ item.tenLopHang,

167 │ item.tenHang,

168 │ item.donViTinh,

169 │ item.giaBanFormat

170 │ });

171 │

172 │ }

173 │ string fileName = FileType == "Excel" ? "DanhSachHangHoa.xlsx" : "DanhSachHangHoa.pdf";

174 │ string contentType = FileType == "Excel" ? "application/ms-excel" : "application/pdf";

175 │

176 │ string fileTemplate = Path.Combine(\_hostingEnvironment.ContentRootPath, "wwwroot", "templates", "DanhSachHan...

177 │ List<ExcelValue> ListGiaTriRiengLe = new()

178 │ {

179 │ };

180 │ int DongBatDau = 5;

181 │ Stream x = ExportToExcel.FromTemplate(fileTemplate, FileType, ListGiaTriRiengLe, dtb, true, DongBatDau, "A",...

182 │ \_logger.LogInformation("Bắt đầu xuất {Count} bản ghi Hàng hóa ra file {FileType}.", listData.Coun...

183 │

184 │ return new FileStreamResult(x, contentType)

185 │ {

186 │ FileDownloadName = fileName

187 │ };

188 │ }

189 │

190 │ [HttpPost]

191 │ [TokenUserCheckHTTP(Permission.Edit)]

192 │ public ActionResult Update(string Ma, string MaKT, string MaVach, string LopHang, string Ten, string DVT, string...

193 │ {

194 │ \_logger.LogInformation("Người dùng {UserId} bắt đầu xử lý lưu (Thêm/Sửa) Hàng hóa. Mã (đ...

195 │

196 │ Ma = Ma.Decrypt();

197 │

198 │ if (MaVach.IsEmpty())

199 │ {

200 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng nhập mã vạch"));

201 │ }

202 │

203 │ if (LopHang.IsEmpty())

204 │ {

205 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng lớp hàng"));

206 │ }

207 │

208 │ Ten = Utility.RemoveSpecialCharacters(Ten + "");

209 │ MaVach = Utility.RemoveSpecialCharacters(MaVach + "");

210 │ MaKT = Utility.RemoveSpecialCharacters(MaKT + "");

211 │

212 │ if (Ten.IsEmpty())

213 │ {

214 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng nhập tên"));

215 │ }

216 │

217 │ if (DVT.IsEmpty())

218 │ {

219 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng chọn đơn vị"));

220 │ }

221 │

222 │ if (Cast.ToDecimal(GiaVon) < 0)

223 │ {

224 │ return Json(new ResponseModel(QLBHConst.Error, "Giá vốn không hợp lệ"));

225 │ }

226 │

227 │ string FileExtension = "";

228 │

229 │

230 │ if (!HinhAnh.IsEmpty())

231 │ {

232 │ FileExtension = HinhAnh.Substring(HinhAnh.LastIndexOf("."));

233 │

234 │ if (FileExtension != "" && !Utility.IsImage(FileExtension))

235 │ {

236 │ return Json(new ResponseModel(QLBHConst.Error, "Hình ảnh không hợp lệ"));

237 │ }

238 │ }

239 │ List<StockDanhMucKho> lstKho = new DanhMucKhoModel(dbcontext).GetData(objToken?.branch ?? Utility.defaultUID...

240 │ StockDanhMucKho drKho;

241 │ ListGiaBanTheoKho = ListGiaBanTheoKho ?? new List<TblGiaBanTheoKho>();

242 │ foreach (TblGiaBanTheoKho item in ListGiaBanTheoKho)

243 │ {

244 │ drKho = lstKho.First(t => t.Ma == Cast.ToInt(item.MaKho));

245 │ if (drKho == null)

246 │ {

247 │ return Json(new ResponseModel(QLBHConst.Error, "Kho " + item.MaKho + " không tồn tại"));

248 │ }

249 │ if (Cast.ToInt(item.GiaBan) < 0)

250 │ {

251 │ return Json(new ResponseModel(QLBHConst.Error, "Giá bán của kho " + drKho.Ten + " không hợp l...

252 │ }

253 │ }

254 │

255 │ if (Ma == "0")//Add

256 │ {

257 │ \_logger.LogInformation("Thực hiện thêm mới hàng hóa: {TenHangHoa}", Ten);

258 │

259 │ string result = Cast.ToString(new TblHangHoaModel(dbcontext).Add(ref MaKT, MaVach, Cast.ToInt(LopHang), ...

260 │ objToken.userid, HinhAnh, ListGiaBanTheoKho, Cast.ToBoolean(BanNoiBo), Cast.ToBoolean(HetHang), objT...

261 │

262 │ ResponseModel responseModel = Utility.MakeResponseFromAddResult(result);

263 │ \_logger.LogInformation("Kết quả thêm mới hàng hóa {TenHangHoa}: {Result}", Ten, result);

264 │

265 │ return Json(responseModel);

266 │ }

267 │ else if (Ma != "0")//Edit

268 │ {

269 │ \_logger.LogInformation("Thực hiện cập nhật hàng hóa ID: {IdHangHoa}", Ma.Decrypt());

270 │

271 │ string result = Cast.ToString(new TblHangHoaModel(dbcontext).Edit(Cast.ToInt(Ma), MaKT, MaVach, Cast.ToI...

272 │ GhiChu, objToken.userid, HinhAnh, ListGiaBanTheoKho, Cast.ToBoolean(BanNoiBo), Cast.ToBoolean(HetHan...

273 │

274 │ ResponseModel responseModel = Utility.MakeResponseFromEditResult(result);

275 │

276 │ \_logger.LogInformation("Kết quả cập nhật hàng hóa ID {IdHangHoa}: {Result}", Ma.Decrypt(), res...

277 │

278 │ if (responseModel != null)

279 │ {

280 │ return Json(responseModel);

281 │ }

282 │ }

283 │

284 │ return Json(QLBHConst.ResponseHaveError);

285 │ }

286 │ [HttpPost]

287 │ [TokenUserCheckHTTP(Permission.CheckHetHang)]

288 │ public ActionResult UpdateStockStatus(string Ma, bool bChecked)

289 │ {

290 │ \_logger.LogInformation("Người dùng {UserId} cập nhật trạng thái Hết hàng={IsOutOfStock} cho HH...

291 │ try

292 │ {

293 │ Ma = Ma.Decrypt();

294 │ var item = dbcontext.TblHangHoa.FirstOrDefault(x => !(x.Deleted ?? false) && x.Ma == Ma.ToInt());

295 │ if (item != null)

296 │ {

297 │ item.HetHang = bChecked;

298 │ dbcontext.Update(item);

299 │ dbcontext.SaveChanges();

300 │ return Json(QLBHConst.ResponseSuccessfully);

301 │ }

302 │ \_logger.LogWarning("Không tìm thấy hàng hóa để cập nhật trạng thái Hết hàng. Mã (đ�...

303 │ return Json(QLBHConst.NotExist);

304 │ }

305 │ catch (Exception ex)

306 │ {

307 │ \_logger.LogError(ex, "Lỗi khi cập nhật trạng thái Hết hàng cho HH mã (đã mã hóa): {MaHa...

308 │ return Json(new ResponseModel(QLBHConst.Error, "Lỗi hệ thống."));

309 │ }

310 │ }

311 │ [HttpPost]

312 │ [TokenUserCheckHTTP(Permission.CheckHangHoaVatTu)]

313 │ public ActionResult setHangHoaVatTu(string Ma, bool bChecked)

314 │ {

315 │ \_logger.LogInformation("Người dùng {UserId} cập nhật trạng thái Vật Tư={IsOutOfStock} cho HH m...

316 │ try

317 │ {

318 │ Ma = Ma.Decrypt();

319 │ var item = dbcontext.TblHangHoa.FirstOrDefault(x => !(x.Deleted ?? false) && x.Ma == Ma.ToInt());

320 │ if (item != null)

321 │ {

322 │ item.VatTu = bChecked;

323 │ dbcontext.Update(item);

324 │ dbcontext.SaveChanges();

325 │ return Json(QLBHConst.ResponseSuccessfully);

326 │ }

327 │ return Json(QLBHConst.NotExist);

328 │ }

329 │ catch (Exception ex)

330 │ {

331 │ \_logger.LogError(ex, "Lỗi khi cập nhật trạng thái Vật Tư cho HH mã (đã mã hóa): {MaHang...

332 │ return Json(new ResponseModel(QLBHConst.Error, "Lỗi hệ thống."));

333 │ }

334 │ }

335 │ [HttpPost]

336 │ [TokenUserCheckHTTP(Permission.Edit)]

337 │ public ActionResult setNoiBo(string Ma, bool bChecked)

338 │ {

339 │ \_logger.LogInformation("Người dùng {UserId} cập nhật trạng thái Nội Bộ={IsOutOfStock} cho HH ...

340 │ try

341 │ {

342 │ Ma = Ma.Decrypt();

343 │ var item = dbcontext.TblHangHoa.FirstOrDefault(x => !(x.Deleted ?? false) && x.Ma == Ma.ToInt());

344 │ if (item != null)

345 │ {

346 │ item.BanNoiBo = bChecked;

347 │ dbcontext.Update(item);

348 │ dbcontext.SaveChanges();

349 │ return Json(QLBHConst.ResponseSuccessfully);

350 │ }

351 │ return Json(QLBHConst.NotExist);

352 │ }

353 │ catch (Exception ex)

354 │ {

355 │ \_logger.LogError(ex, "Lỗi khi cập nhật trạng thái Nội Bộ cho HH mã (đã mã hóa): {MaHan...

356 │ return Json(new ResponseModel(QLBHConst.Error, "Lỗi hệ thống."));

357 │ }

358 │ }

359 │ [HttpPost]

360 │ [TokenUserCheckHTTP(Permission.Delete)]

361 │ public IActionResult DeleteAPI(string Ma)

362 │ {

363 │ \_logger.LogInformation("Người dùng {UserId} yêu cầu xóa Hàng hóa. Mã (đã mã hóa): {MaHangHoa}...

364 │ try

365 │ {

366 │ Ma = Ma.Decrypt();

367 │ string result = Cast.ToString(new TblHangHoaModel(dbcontext).Delete(Cast.ToInt(Ma), objToken.userid));

368 │ \_logger.LogInformation("Kết quả xóa hàng hóa ID {IdHangHoa}: {Result}", Ma, result);

369 │ return Json(Utility.MakeResponseFromDeleteResult(result));

370 │ }

371 │ catch (Exception ex)

372 │ {

373 │ \_logger.LogError(ex, "Lỗi khi xóa Hàng hóa mã (đã mã hóa): {MaHangHoa}", Ma);

374 │ return Json(new ResponseModel(QLBHConst.Error, "Lỗi hệ thống khi xóa."));

375 │ }

376 │ }

377 │

378 │ private List<dynamic> GetListDataParent(Guid maChiNhanh)

379 │ {

380 │ \_logger.LogDebug("Bắt đầu lấy danh sách hàng hóa cha cho chi nhánh: {MaChiNhanh}", maChiNhanh);

381 │

382 │ // Bước 1: Lấy các mã hàng là con (ParentId > 0)

383 │ var listToHashSet = dbcontext.TblHangHoa.AsNoTracking()

384 │ .Where(t => t.MaLopHang == (int)LopHang.MatKinh

385 │ && (t.ParentId ?? 0) > 0

386 │ && !(t.Deleted ?? false)

387 │ && t.MaChiNhanh == maChiNhanh)

388 │ .Select(x => x.Ma)

389 │ .ToHashSet();

390 │

391 │ // Bước 2: Lấy các mặt hàng là "cha"

392 │ var listData = dbcontext.TblHangHoa.AsNoTracking()

393 │ .Where(t => t.MaLopHang == (int)LopHang.MatKinh

394 │ && !(t.Deleted ?? false)

395 │ && t.MaChiNhanh == maChiNhanh

396 │ && !listToHashSet.Contains(t.Ma))

397 │ .Select(h => new

398 │ {

399 │ maEncrypt = h.Ma.ToString().Encrypt(),

400 │ ma = h.Ma,

401 │ maKt = h.MaKt ?? "",

402 │ maVach = h.MaVach ?? "",

403 │ tenHang = h.Ten ?? "",

404 │ giaBan = h.GiaBan ?? 0,

405 │ giaVon = h.GiaVon ?? 0,

406 │ maDvt = h.MaDvt,

407 │ donViTinh = "",

408 │ maLopHang = 0,

409 │ tenLopHang = "",

410 │ banNoiBo = h.BanNoiBo,

411 │ hetHang = h.HetHang,

412 │ vatTu = h.VatTu,

413 │ hinhAnh = h.HinhAnh ?? "",

414 │ trangThai = h.Status ?? false,

415 │ giaBanFormat = string.Format("{0:#,##0.00}", h.GiaBan ?? 0),

416 │ parentId = h.ParentId

417 │ })

418 │ .ToList<dynamic>();

419 │

420 │ return listData;

421 │ }

422 │

423 │ [HttpPost]

424 │ [TokenUserCheckHTTP]

425 │ public ActionResult GetListDataForApp(int? MaKho = null)

426 │ {

427 │ try

428 │ {

429 │ Guid MaChiNhanh = objToken.branch;

430 │

431 │ // Lấy danh sách kho của user

432 │ var listKhoUser = new DanhMucKhoModel(dbcontext).GetDataByUser(objToken.userid, objToken.branch);

433 │

434 │ // Nếu không truyền mã kho, lấy kho đầu tiên

435 │ int maKhoFilter = MaKho ?? listKhoUser.FirstOrDefault()?.Ma ?? 0;

436 │

437 │ var rawData = (from h in dbcontext.TblHangHoa.Where(t => !(t.Deleted ?? false) && t.MaChiNhanh == MaChiN...

438 │ join d in dbcontext.TblDonViTinh.Where(t => !(t.Deleted ?? false)) on h.MaDvt equals d.Ma...

439 │ from d in tempDVT.DefaultIfEmpty()

440 │ join l in dbcontext.TblLopHangHoa.Where(t => !(t.Deleted ?? false) && t.MaChiNhanh == MaC...

441 │ from l in tempLop.DefaultIfEmpty()

442 │ join hh in dbcontext.TblHangHoa.Where(t => !(t.Deleted ?? false)) on h.ParentId equals hh...

443 │ from hh in tempHH.DefaultIfEmpty()

444 │ // Join với kho để lấy thông tin kho và giá theo kho

445 │ join gbtk in dbcontext.TblGiaBanTheoKho.Where(x => x.MaKho == maKhoFilter) on h.Ma equals...

446 │ from gbtk in tempGBTK.DefaultIfEmpty()

447 │ join k in dbcontext.StockDanhMucKho.Where(x => !(x.Deleted ?? false) && x.Ma == maKhoFilt...

448 │ from k in tempKho.DefaultIfEmpty()

449 │ select new { h, d, l, hh, gbtk, k }).ToList();

450 │

451 │ var listData = rawData.Select(x => new

452 │ {

453 │ maEncrypt = x.h.Ma.ToString().Encrypt(),

454 │ ma = x.h.Ma + "",

455 │ maKt = x.h.MaKt + "",

456 │ maVach = x.h.MaVach + "",

457 │ tenHang = x.h.Ten ?? "",

458 │ giaBan = x.h.GiaBan ?? 0,

459 │ giaVon = x.h.GiaVon ?? 0,

460 │ donViTinh = x.d?.Ten ?? "",

461 │ maLopHang = x.l?.Ma,

462 │ tenLopHang = x.l?.Ten ?? "",

463 │ banNoiBo = x.h.BanNoiBo,

464 │ hetHang = x.h.HetHang,

465 │ vatTu = x.h.VatTu,

466 │ hinhAnh = x.h.HinhAnh ?? "",

467 │ trangThai = x.h.Status ?? false,

468 │ giaBanFormat = string.Format("{0:#,##0.00}", x.h.GiaBan ?? 0),

469 │ tenParent = x.hh?.Ten ?? "",

470 │

471 │ // Thông tin kho

472 │ maKho = x.k?.Ma ?? 0,

473 │ tenKho = x.k?.Ten ?? "",

474 │ giaBanTaiKho = x.gbtk?.GiaBan ?? x.h.GiaBan ?? 0, // Nếu không có giá theo kho thì lấy giá ...

475 │ giaBanTaiKhoFormat = string.Format("{0:#,##0.00}", x.gbtk?.GiaBan ?? x.h.GiaBan ?? 0)

476 │ }).ToList();

477 │

478 │ return Json(new ResponseModelSuccess("", listData));

479 │ }

480 │ catch (Exception ex)

481 │ {

482 │ return Json(new ResponseModel(QLBHConst.Error, "Có lỗi xảy ra: " + ex.Message));

483 │ }

484 │ }

485 │

486 │ [TokenUserCheckHTTP(Permission.Add)]

487 │ public ActionResult AddRFID()

488 │ {

489 │ \_logger.LogInformation("Người dùng {UserId} truy cập trang thêm mới RFID Hàng hóa.", objToken.use...

490 │ try

491 │ {

492 │ ViewData["ListData"] = GetListDataParent(objToken.branch);

493 │ ViewData["ListDVT"] = dbcontext.TblDonViTinh.Where(x => !(x.Deleted ?? false)).ToList();

494 │

495 │ return View("FormRFID");

496 │ }

497 │ catch (Exception ex)

498 │ {

499 │ \_logger.LogError(ex, "Lỗi khi tải dữ liệu cho trang thêm mới Hàng hóa.");

500 │ return View("Index");

501 │ }

502 │ }

503 │ }

504 │ }

────────────────────────────────────────────────────────────

**📄 HangHoaModel.cs (CS, 316 lines)**

1 │ ﻿using Microsoft.EntityFrameworkCore;

2 │ using QuanLyBanHang.Common;

3 │ using QuanLyBanHang.Models.QLBHEntity;

4 │

5 │ namespace QuanLyBanHang.Models.ExModel.CAT

6 │ {

7 │ public class TblHangHoaModel : Base<TblHangHoaModel>

8 │ {

9 │ public TblHangHoaModel(DbContext dbContext) : base(dbContext)

10 │ {

11 │ }

12 │ QuanLyBanHangContext dbContext

13 │ {

14 │ get { return \_dbContext as QuanLyBanHangContext; }

15 │ set { return; }

16 │ }

17 │

18 │ public TblHangHoa? GetItem(int Ma)

19 │ {

20 │ return dbContext.TblHangHoa.FirstOrDefault(t => t.Ma == Ma && !(t.Deleted ?? false));

21 │ }

22 │

23 │ public List<TblHangHoa> GetData(Guid MaChiNhanh)

24 │ {

25 │ return dbContext.TblHangHoa.Where(t => !(t.Deleted ?? false) && t.MaChiNhanh == MaChiNhanh).OrderBy(t => t.T...

26 │ }

27 │ public VHangHoa GetTblHangHoaTheoMaVach(string MaVach, Guid MaChiNhanh)

28 │ {

29 │ return dbContext.VHangHoa.First(t => t.MaVach == MaVach && t.MaChiNhanh == MaChiNhanh);

30 │ }

31 │ public List<VHangHoa> LayDanhSachTblHangHoa(Guid MaChiNhanh)

32 │ {

33 │ var listParentId = dbContext.VHangHoa.AsNoTracking().Where(x => x.MaChiNhanh == MaChiNhanh).Select(x=>x.Pare...

34 │ return dbContext.VHangHoa.AsNoTracking().Where(x => !listParentId.Contains(x.Ma) && x.MaChiNhanh == MaChiNha...

35 │ }

36 │

37 │ public int Add(ref string MaKT, string MaVach, int LopHang, string Ten, int DVT, int GiaBan,

38 │ decimal GiaVon, string GhiChu, Guid UserID, string HinhAnh, List<TblGiaBanTheoKho> ListGiaBanTheoKho, bool B...

39 │ {

40 │

41 │ if (CheckExists(0, MaVach))

42 │ return -3;

43 │ if (!MaKT.IsEmpty())

44 │ {

45 │ if (CheckExistsMAKT(0, MaKT))

46 │ return -3;

47 │ }

48 │

49 │ var br = dbContext.SysBranch.FirstOrDefault(x => x.Ma == MaChiNhanh);

50 │ if (br == null)

51 │ {

52 │ throw new Exception("Không tìm thấy chi nhánh");

53 │ }

54 │ TblHangHoa itemAdd = new TblHangHoa

55 │ {

56 │ MaKt = MaKT,

57 │ HinhAnh = HinhAnh != "" ? HinhAnh : "",

58 │ MaVach = MaVach,

59 │ MaLopHang = LopHang,

60 │ MaDvt = DVT,

61 │ GiaBan = GiaBan,

62 │ GiaVon = GiaVon,

63 │ GhiChu = GhiChu,

64 │ Ten = Ten,

65 │ CreatedDate = DateTime.Now,

66 │ UserCreated = UserID,

67 │ BanNoiBo = BanNoiBo,

68 │ HetHang = HetHang,

69 │ VatTu = VatTu,

70 │ Status = TrangThai,

71 │ MaChiNhanh = MaChiNhanh,

72 │ MaChiNhanhInt = br.Nid ?? 0,

73 │ MaDinhMuc = MaDinhMuc,

74 │ ParentId = MaParent

75 │ };

76 │

77 │ dbContext.Add(itemAdd);

78 │ dbContext.SaveChanges();

79 │

80 │ List<TblLichSuChinhGia> lstLS = new List<TblLichSuChinhGia>();

81 │

82 │ lstLS.Add(new TblLichSuChinhGia

83 │ {

84 │ MaHangHoa = itemAdd.Ma,

85 │ GiaCu = 0,

86 │ GiaMoi = GiaVon,

87 │ UserCreated = itemAdd.UserCreated,

88 │ CreatedDate = itemAdd.CreatedDate,

89 │ GhiChu = "Khởi tạo giá vốn từ danh mục hàng hóa",

90 │

91 │ });

92 │ if (ListGiaBanTheoKho != null)

93 │ {

94 │ foreach (TblGiaBanTheoKho gb in ListGiaBanTheoKho)

95 │ {

96 │ TblGiaBanTheoKho? old = new GiaBanTheoKhoModel(dbContext).GetItem(Cast.ToInt(gb.MaKho), itemAdd.Ma, ...

97 │ if (old == null)

98 │ {

99 │ dbContext.TblGiaBanTheoKho.Add(new TblGiaBanTheoKho

100 │ {

101 │ MaChiNhanh = MaChiNhanh,

102 │ MaKho = Cast.ToInt(gb.MaKho),

103 │ MaHangHoa = itemAdd.Ma,

104 │ GiaBan = Cast.ToInt(gb.GiaBan),

105 │ GhiChu = "Thêm mới từ danh mục hàng hóa",

106 │ CreatedDate = DateTime.Now,

107 │ UserCreated = UserID

108 │ });

109 │ lstLS.Add(new TblLichSuChinhGia

110 │ {

111 │ MaKho = Cast.ToInt(gb.MaKho),

112 │ MaHangHoa = itemAdd.Ma,

113 │ GiaCu = 0,

114 │ GiaMoi = Cast.ToInt(gb.GiaBan),

115 │ UserCreated = itemAdd.UserCreated,

116 │ CreatedDate = itemAdd.CreatedDate,

117 │ GhiChu = "Khởi tạo giá bán từ danh mục hàng hóa"

118 │ });

119 │ }

120 │ else

121 │ {

122 │ if ((old.GiaBan ?? 0) != Cast.ToInt(gb.GiaBan))

123 │ {

124 │ lstLS.Add(new TblLichSuChinhGia

125 │ {

126 │ MaKho = Cast.ToInt(gb.MaKho),

127 │ MaHangHoa = itemAdd.Ma,

128 │ GiaCu = (old.GiaBan ?? 0),

129 │ GiaMoi = Cast.ToInt(gb.GiaBan),

130 │ UserModified = itemAdd.UserCreated,

131 │ ModifiedDate = itemAdd.CreatedDate,

132 │ GhiChu = "Cập nhật giá bán từ danh mục hàng hóa"

133 │ });

134 │ }

135 │ old.GiaBan = Cast.ToInt(gb.GiaBan);

136 │ old.GhiChu = "Cập nhật từ danh mục hàng hóa";

137 │ old.ModifiedDate = DateTime.Now;

138 │ dbContext.TblGiaBanTheoKho.Update(old);

139 │ }

140 │ dbContext.SaveChanges();

141 │

142 │ }

143 │ }

144 │ if (lstLS.Count > 0)

145 │ {

146 │ dbContext.TblLichSuChinhGia.AddRange(lstLS);

147 │ }

148 │ VHangHoa hh = GetHangHoaTheoMaVach(MaVach);

149 │ if (hh != null)

150 │ {

151 │ MaKT = hh.MaKt;

152 │ }

153 │ dbContext.SaveChanges();

154 │

155 │ return 1;

156 │

157 │ }

158 │ public int Edit(int Ma, string MaKT, string MaVach, int LopHang, string Ten, int DVT, int GiaBan, decimal GiaVon...

159 │ string GhiChu, Guid UserID, string HinhAnh, List<TblGiaBanTheoKho> ListGiaBanTheoKho, bool BanNoiBo, bool He...

160 │ {

161 │ if (!MaVach.IsEmpty() && CheckExists(Ma, MaVach))

162 │ return -3;

163 │

164 │ if (!MaKT.IsEmpty() && CheckExistsMAKT(Ma, MaKT))

165 │ return -3;

166 │

167 │ TblHangHoa itemUpdate = GetItem(Ma);

168 │

169 │ if (itemUpdate != null)

170 │ {

171 │ decimal giacu = itemUpdate.GiaVon ?? 0;

172 │ itemUpdate.MaKt = MaKT;

173 │ itemUpdate.MaVach = MaVach;

174 │ itemUpdate.MaLopHang = LopHang;

175 │ itemUpdate.MaDvt = DVT;

176 │ itemUpdate.GiaBan = GiaBan;

177 │ itemUpdate.GhiChu = GhiChu;

178 │ itemUpdate.Ten = Ten;

179 │ itemUpdate.GiaVon = GiaVon;

180 │ itemUpdate.UserModified = UserID;

181 │ itemUpdate.ModifiedDate = DateTime.Now;

182 │ itemUpdate.HinhAnh = HinhAnh != "" ? HinhAnh : "/upload/200x200.png";

183 │ itemUpdate.BanNoiBo = BanNoiBo;

184 │ itemUpdate.HetHang = HetHang;

185 │ itemUpdate.VatTu = VatTu;

186 │ itemUpdate.MaChiNhanh = MaChiNhanh;

187 │ itemUpdate.Status = TrangThai;

188 │ itemUpdate.MaDinhMuc = MaDinhMuc;

189 │ itemUpdate.ParentId = MaParent;

190 │

191 │ dbContext.Update(itemUpdate);

192 │

193 │ List<TblLichSuChinhGia> lstLS = new List<TblLichSuChinhGia>();

194 │ if (giacu != GiaVon)

195 │ {

196 │ lstLS.Add(new TblLichSuChinhGia

197 │ {

198 │ MaHangHoa = itemUpdate.Ma,

199 │ GiaCu = giacu,

200 │ GiaMoi = GiaVon,

201 │ UserCreated = itemUpdate.UserCreated,

202 │ CreatedDate = itemUpdate.CreatedDate,

203 │ GhiChu = "Cập nhật giá vốn từ danh mục hàng hóa"

204 │ });

205 │ }

206 │ GiaBanTheoKhoModel model = new GiaBanTheoKhoModel(dbContext);

207 │ foreach (TblGiaBanTheoKho gb in ListGiaBanTheoKho)

208 │ {

209 │ TblGiaBanTheoKho old = model.GetItem(Cast.ToInt(gb.MaKho), itemUpdate.Ma, MaChiNhanh);

210 │ if (old == null)

211 │ {

212 │ model.Add(new TblGiaBanTheoKho

213 │ {

214 │ MaChiNhanh = MaChiNhanh,

215 │ MaKho = Cast.ToInt(gb.MaKho),

216 │ MaHangHoa = itemUpdate.Ma,

217 │ GiaBan = Cast.ToInt(gb.GiaBan),

218 │ GhiChu = "Thêm mới từ danh mục hàng hóa",

219 │ CreatedDate = DateTime.Now,

220 │ UserCreated = UserID

221 │ }, true);

222 │ lstLS.Add(new TblLichSuChinhGia

223 │ {

224 │ MaKho = Cast.ToInt(gb.MaKho),

225 │ MaHangHoa = itemUpdate.Ma,

226 │ GiaCu = 0,

227 │ GiaMoi = Cast.ToInt(gb.GiaBan),

228 │ UserCreated = itemUpdate.UserCreated,

229 │ CreatedDate = itemUpdate.CreatedDate,

230 │ GhiChu = "Thêm mới giá vốn từ danh mục hàng hóa"

231 │ });

232 │ }

233 │ else

234 │ {

235 │ if ((old.GiaBan ?? 0) != Cast.ToInt(gb.GiaBan))

236 │ {

237 │ lstLS.Add(new TblLichSuChinhGia

238 │ {

239 │ MaKho = Cast.ToInt(gb.MaKho),

240 │ MaHangHoa = itemUpdate.Ma,

241 │ GiaCu = (old.GiaBan ?? 0),

242 │ GiaMoi = Cast.ToInt(gb.GiaBan),

243 │ UserCreated = itemUpdate.UserCreated,

244 │ CreatedDate = itemUpdate.CreatedDate,

245 │ GhiChu = "Cập nhật giá vốn từ danh mục hàng hóa"

246 │ });

247 │ }

248 │ old.GiaBan = Cast.ToInt(gb.GiaBan);

249 │ old.GhiChu = "Cập nhật từ danh mục hàng hóa";

250 │ old.CreatedDate = DateTime.Now;

251 │ model.Update(old);

252 │ }

253 │

254 │ if (lstLS.Count > 0)

255 │ {

256 │ dbContext.TblLichSuChinhGia.AddRange(lstLS);

257 │ }

258 │ }

259 │ dbContext.SaveChanges();

260 │ return 1;

261 │

262 │ }

263 │

264 │ return -2;

265 │ }

266 │

267 │ public int Delete(int Ma, Guid UserID)

268 │ {

269 │ TblHangHoa drItem = GetItem(Ma);

270 │

271 │ if (drItem != null)

272 │ {

273 │ //Check children

274 │ TblHangHoa drHH = dbContext.TblHangHoa.FirstOrDefault(t => t.Ma == Ma);

275 │

276 │ if (drHH == null)

277 │ return -1;

278 │ else

279 │ {

280 │ drItem.Deleted = true;

281 │ drItem.DeletedDate = DateTime.Now;

282 │ drItem.UserDeleted = UserID;

283 │

284 │ dbContext.SaveChanges();

285 │ return 1;

286 │

287 │ }

288 │ }

289 │

290 │ return -2;

291 │ }

292 │ private bool CheckExists(int Ma, string MaVach)

293 │ {

294 │ return dbContext.TblHangHoa.Any(t => t.Ma != Ma && t.MaVach == MaVach && (t.Status ?? false) && !(t.Deleted ...

295 │ }

296 │ private bool CheckExistsMAKT(int Ma, string MaKT)

297 │ {

298 │ return dbContext.TblHangHoa.Any(t => t.Ma != Ma && t.MaKt == MaKT && !(t.Deleted ?? false));

299 │ }

300 │

301 │ internal void Update(TblHangHoa hh)

302 │ {

303 │ throw new NotImplementedException();

304 │ }

305 │

306 │ public VHangHoa GetHangHoaTheoMaVach(string MaVach)

307 │ {

308 │ return dbContext.VHangHoa.First(t => t.MaVach == MaVach && (t.Status ?? false));

309 │ }

310 │

311 │ public List<VHangHoa> LayDanhSachHangHoa()

312 │ {

313 │ return dbContext.VHangHoa.AsNoTracking().Where(x => (x.Status ?? false)).OrderBy(t => t.TenHangHoa).ToList()...

314 │ }

315 │ }

316 │ }

────────────────────────────────────────────────────────────

**📄 HangHoaRfidController.cs (CS, 105 lines)**

1 │ ﻿using EFCore.BulkExtensions;

2 │ using Microsoft.AspNetCore.Mvc;

3 │ using Microsoft.EntityFrameworkCore;

4 │ using QuanLyBanHang.Common;

5 │ using QuanLyBanHang.Common.Attributes;

6 │ using QuanLyBanHang.Models.APIModels;

7 │ using QuanLyBanHang.Models.DTO;

8 │ using QuanLyBanHang.Models.QLBHEntity;

9 │

10 │ namespace QuanLyBanHang.Controllers.CAT

11 │ {

12 │ public class HangHoaRfidController : BaseController

13 │ {

14 │ public HangHoaRfidController(QuanLyBanHangContext dbcontext, MemoryCacheManager memoryCache) : base(dbcontext, m...

15 │ {

16 │ }

17 │

18 │ public IActionResult Index()

19 │ {

20 │ return View();

21 │ }

22 │

23 │ [HttpPost]

24 │ [TokenUserCheckHTTP(Permission.Add)]

25 │ public async Task<IActionResult> AddAPIRfid(List<RfidHangHoa> listHangHoa, string MaHangHoa, string Status)

26 │ {

27 │ if (listHangHoa == null || !listHangHoa.Any())

28 │ {

29 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng quét ít nhất 1 mã RFID"));

30 │ }

31 │

32 │ if (!int.TryParse(MaHangHoa, out int maHangHoa))

33 │ {

34 │ return Json(new ResponseModel(QLBHConst.Error, "Mã hàng hoá không hợp lệ"));

35 │ }

36 │

37 │ var checkHH = dbcontext.TblHangHoa

38 │ .AsNoTracking()

39 │ .FirstOrDefault(t => t.Ma == maHangHoa

40 │ && (t.ParentId ?? 0) == 0

41 │ && t.MaChiNhanh == objToken.branch

42 │ && !(t.Deleted ?? false));

43 │

44 │ if (checkHH == null)

45 │ {

46 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng chọn hàng hoá"));

47 │ }

48 │

49 │ var maChiNhanhInt = dbcontext.SysBranch

50 │ .AsNoTracking()

51 │ .Where(t => t.Ma == objToken.branch)

52 │ .Select(t => t.Nid)

53 │ .FirstOrDefault();

54 │

55 │ var rfidList = listHangHoa.Select(x => x.RfID).ToList();

56 │

57 │ var rfidTrung = dbcontext.TblHangHoa

58 │ .AsNoTracking()

59 │ .Where(t => rfidList.Contains(t.MaVach) && (t.Status ?? false) && !(t.Deleted ?? false))

60 │ .Select(t => t.MaVach)

61 │ .ToList();

62 │

63 │ if (rfidTrung.Any())

64 │ {

65 │ string msg = "Các mã RFID đã tồn tại: " + string.Join(", ", rfidTrung);

66 │ return Json(new ResponseModel(QLBHConst.Error, msg));

67 │ }

68 │

69 │ List<TblHangHoa> items = new List<TblHangHoa>();

70 │

71 │ foreach (var item in listHangHoa)

72 │ {

73 │ int? maDvt = int.TryParse(item.MaDVT, out var tmpMaDvt) ? tmpMaDvt : (int?)null;

74 │ decimal? giaVon = decimal.TryParse(item.GiaVon, out var tmpGiaVon) ? tmpGiaVon : (decimal?)null;

75 │

76 │ items.Add(new TblHangHoa

77 │ {

78 │ MaVach = item.RfID,

79 │ ParentId = maHangHoa,

80 │ Status = Status == "1",

81 │ UserCreated = objToken.userid,

82 │ CreatedDate = DateTime.Now,

83 │ MaChiNhanh = objToken.branch,

84 │ MaChiNhanhInt = maChiNhanhInt ?? 0,

85 │ Ten = item.TenHangHoa,

86 │ MaDvt = maDvt,

87 │ MaLopHang = checkHH.MaLopHang,

88 │ GiaVon = giaVon

89 │ });

90 │ }

91 │

92 │ try

93 │ {

94 │ await dbcontext.BulkInsertAsync(items);

95 │

96 │ return Json(new ResponseModel(QLBHConst.OK, $"Đã thêm hàng hoá"));

97 │

98 │ }

99 │ catch (Exception ex)

100 │ {

101 │ return Json(new ResponseModel(QLBHConst.Error, ex.Message));

102 │ }

103 │ }

104 │ }

105 │ }

────────────────────────────────────────────────────────────

**📄 HappyStoreContext.cs (CS, 1601 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │ using Microsoft.EntityFrameworkCore;

4 │

5 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

6 │

7 │ public partial class HappyStoreContext : DbContext

8 │ {

9 │ public HappyStoreContext()

10 │ {

11 │ }

12 │

13 │ public HappyStoreContext(DbContextOptions<HappyStoreContext> options)

14 │ : base(options)

15 │ {

16 │ }

17 │

18 │ public virtual DbSet<BidvQr> BidvQr { get; set; }

19 │

20 │ public virtual DbSet<CtHoaDon> CtHoaDon { get; set; }

21 │

22 │ public virtual DbSet<CtLichSuGia> CtLichSuGia { get; set; }

23 │

24 │ public virtual DbSet<CtNhapHang> CtNhapHang { get; set; }

25 │

26 │ public virtual DbSet<DonViTinh> DonViTinh { get; set; }

27 │

28 │ public virtual DbSet<GiamGia> GiamGia { get; set; }

29 │

30 │ public virtual DbSet<HangHoa> HangHoa { get; set; }

31 │

32 │ public virtual DbSet<Hhimport> Hhimport { get; set; }

33 │

34 │ public virtual DbSet<HinhThucTt> HinhThucTt { get; set; }

35 │

36 │ public virtual DbSet<HisLichSuChinhGia> HisLichSuChinhGia { get; set; }

37 │

38 │ public virtual DbSet<HoaDonBan> HoaDonBan { get; set; }

39 │

40 │ public virtual DbSet<KkBienBanKiemKe> KkBienBanKiemKe { get; set; }

41 │

42 │ public virtual DbSet<KkChiTietKiemKe> KkChiTietKiemKe { get; set; }

43 │

44 │ public virtual DbSet<KkChiTietTonKho> KkChiTietTonKho { get; set; }

45 │

46 │ public virtual DbSet<LichSuGia> LichSuGia { get; set; }

47 │

48 │ public virtual DbSet<LopHang> LopHang { get; set; }

49 │

50 │ public virtual DbSet<NccChti> NccChti { get; set; }

51 │

52 │ public virtual DbSet<NccDược> NccDược { get; set; }

53 │

54 │ public virtual DbSet<NguoiDung> NguoiDung { get; set; }

55 │

56 │ public virtual DbSet<NhaCc> NhaCc { get; set; }

57 │

58 │ public virtual DbSet<NhapHang> NhapHang { get; set; }

59 │

60 │ public virtual DbSet<SysDsbaoCao> SysDsbaoCao { get; set; }

61 │

62 │ public virtual DbSet<SysDsdieuKienBaoCao> SysDsdieuKienBaoCao { get; set; }

63 │

64 │ public virtual DbSet<SysMenu> SysMenu { get; set; }

65 │

66 │ public virtual DbSet<SysPrinter> SysPrinter { get; set; }

67 │

68 │ public virtual DbSet<SysQuyen> SysQuyen { get; set; }

69 │

70 │ public virtual DbSet<SysRole> SysRole { get; set; }

71 │

72 │ public virtual DbSet<SysRoleMenu> SysRoleMenu { get; set; }

73 │

74 │ public virtual DbSet<SysRoleQuyen> SysRoleQuyen { get; set; }

75 │

76 │ public virtual DbSet<SysStatus> SysStatus { get; set; }

77 │

78 │ public virtual DbSet<SysUserRole> SysUserRole { get; set; }

79 │

80 │ public virtual DbSet<TblChiTietXuatKho> TblChiTietXuatKho { get; set; }

81 │

82 │ public virtual DbSet<TblCtKetCa> TblCtKetCa { get; set; }

83 │

84 │ public virtual DbSet<TblDanhMucKho> TblDanhMucKho { get; set; }

85 │

86 │ public virtual DbSet<TblGiaBanTheoKho> TblGiaBanTheoKho { get; set; }

87 │

88 │ public virtual DbSet<TblGiuongBenh> TblGiuongBenh { get; set; }

89 │

90 │ public virtual DbSet<TblKyKho> TblKyKho { get; set; }

91 │

92 │ public virtual DbSet<TblLoaiNhapXuat> TblLoaiNhapXuat { get; set; }

93 │

94 │ public virtual DbSet<TblNgayKetCa> TblNgayKetCa { get; set; }

95 │

96 │ public virtual DbSet<TblPhongBan> TblPhongBan { get; set; }

97 │

98 │ public virtual DbSet<TblPhongBenh> TblPhongBenh { get; set; }

99 │

100 │ public virtual DbSet<TblThueVat> TblThueVat { get; set; }

101 │

102 │ public virtual DbSet<TblTransTrinm> TblTransTrinm { get; set; }

103 │

104 │ public virtual DbSet<TblXuatKho> TblXuatKho { get; set; }

105 │

106 │ public virtual DbSet<ThongTinShop> ThongTinShop { get; set; }

107 │

108 │ public virtual DbSet<TmpOrder> TmpOrder { get; set; }

109 │

110 │ public virtual DbSet<TmpOrderDetail> TmpOrderDetail { get; set; }

111 │

112 │ public virtual DbSet<TmpUpdateTonKho> TmpUpdateTonKho { get; set; }

113 │

114 │ public virtual DbSet<TonDauKy1> TonDauKy1 { get; set; }

115 │

116 │ public virtual DbSet<TonDayKy2> TonDayKy2 { get; set; }

117 │

118 │ public virtual DbSet<TonKho> TonKho { get; set; }

119 │

120 │ public virtual DbSet<TonKhoBanHang> TonKhoBanHang { get; set; }

121 │

122 │ public virtual DbSet<TonKhoK> TonKhoK { get; set; }

123 │

124 │ public virtual DbSet<TonKhoTuNgayDenNgay> TonKhoTuNgayDenNgay { get; set; }

125 │

126 │ public virtual DbSet<VChiTietKiemKe> VChiTietKiemKe { get; set; }

127 │

128 │ public virtual DbSet<VChiTietNhapHang> VChiTietNhapHang { get; set; }

129 │

130 │ public virtual DbSet<VChiTietXuatKho> VChiTietXuatKho { get; set; }

131 │

132 │ public virtual DbSet<VGiaBanTheoKho> VGiaBanTheoKho { get; set; }

133 │

134 │ public virtual DbSet<VHangHoa> VHangHoa { get; set; }

135 │

136 │ public virtual DbSet<VNhaCc> VNhaCc { get; set; }

137 │

138 │ public virtual DbSet<VNhapHang> VNhapHang { get; set; }

139 │

140 │ public virtual DbSet<VQuyen> VQuyen { get; set; }

141 │

142 │ public virtual DbSet<VTonKho> VTonKho { get; set; }

143 │

144 │ public virtual DbSet<VTonKhoBanHang> VTonKhoBanHang { get; set; }

145 │

146 │ public virtual DbSet<VTonKhoTuNgayDenNgay> VTonKhoTuNgayDenNgay { get; set; }

147 │

148 │ protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

149 │ => optionsBuilder.UseSqlServer("Name=ConnectionStrings:HappyStore");

150 │

151 │ protected override void OnModelCreating(ModelBuilder modelBuilder)

152 │ {

153 │ modelBuilder.Entity<BidvQr>(entity =>

154 │ {

155 │ entity.HasKey(e => e.Ma);

156 │

157 │ entity.ToTable("bidv\_QR");

158 │

159 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

160 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

161 │ entity.Property(e => e.MaQr)

162 │ .HasMaxLength(50)

163 │ .IsUnicode(false)

164 │ .HasColumnName("MaQR");

165 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

166 │ entity.Property(e => e.NoiDung).HasMaxLength(500);

167 │ entity.Property(e => e.Qrdata)

168 │ .HasMaxLength(500)

169 │ .HasColumnName("QRData");

170 │ entity.Property(e => e.SoHoaDon).HasMaxLength(100);

171 │ });

172 │

173 │ modelBuilder.Entity<CtHoaDon>(entity =>

174 │ {

175 │ entity.HasKey(e => e.Ma);

176 │

177 │ entity.ToTable("CT\_HoaDon");

178 │

179 │ entity.Property(e => e.HangHoaId).HasColumnName("HangHoa\_ID");

180 │ entity.Property(e => e.HoaDonId).HasColumnName("HoaDon\_ID");

181 │ entity.Property(e => e.MaDvt).HasColumnName("MaDVT");

182 │ entity.Property(e => e.MaVach)

183 │ .HasMaxLength(20)

184 │ .IsUnicode(false);

185 │ entity.Property(e => e.NgaySua).HasColumnType("datetime");

186 │ entity.Property(e => e.NgayTao).HasColumnType("datetime");

187 │ entity.Property(e => e.NgayTra).HasColumnType("datetime");

188 │ entity.Property(e => e.NgayXoa).HasColumnType("datetime");

189 │ });

190 │

191 │ modelBuilder.Entity<CtLichSuGia>(entity =>

192 │ {

193 │ entity.HasKey(e => e.Ma).HasName("PK\_CT\_LichSu");

194 │

195 │ entity.ToTable("CT\_LichSuGia");

196 │

197 │ entity.Property(e => e.GhiChu).HasMaxLength(100);

198 │ entity.Property(e => e.HangHoaId).HasColumnName("HangHoa\_ID");

199 │ entity.Property(e => e.MaKho).HasMaxLength(500);

200 │ entity.Property(e => e.NgaySua).HasColumnType("datetime");

201 │ entity.Property(e => e.NgayTao).HasColumnType("datetime");

202 │ entity.Property(e => e.NgayXoa).HasColumnType("datetime");

203 │ });

204 │

205 │ modelBuilder.Entity<CtNhapHang>(entity =>

206 │ {

207 │ entity.HasKey(e => e.Ma);

208 │

209 │ entity.ToTable("CT\_NhapHang");

210 │

211 │ entity.Property(e => e.ChietKhauPt)

212 │ .HasColumnType("decimal(20, 4)")

213 │ .HasColumnName("ChietKhauPT");

214 │ entity.Property(e => e.ChietKhauTm)

215 │ .HasColumnType("decimal(20, 4)")

216 │ .HasColumnName("ChietKhauTM");

217 │ entity.Property(e => e.GiaVon).HasColumnType("decimal(20, 4)");

218 │ entity.Property(e => e.HanSd)

219 │ .HasColumnType("datetime")

220 │ .HasColumnName("HanSD");

221 │ entity.Property(e => e.NgaySua).HasColumnType("datetime");

222 │ entity.Property(e => e.NgayTao)

223 │ .HasDefaultValueSql("(getdate())")

224 │ .HasColumnType("datetime");

225 │ entity.Property(e => e.NgayXoa).HasColumnType("datetime");

226 │ entity.Property(e => e.SoLuong).HasColumnType("decimal(20, 4)");

227 │ entity.Property(e => e.ThanhTien).HasColumnType("decimal(20, 4)");

228 │ entity.Property(e => e.ThanhTienSauCk)

229 │ .HasColumnType("decimal(20, 4)")

230 │ .HasColumnName("ThanhTienSauCK");

231 │ entity.Property(e => e.ThanhTienSauVat)

232 │ .HasColumnType("decimal(20, 4)")

233 │ .HasColumnName("ThanhTienSauVAT");

234 │ entity.Property(e => e.Vat).HasColumnName("VAT");

235 │ });

236 │

237 │ modelBuilder.Entity<DonViTinh>(entity =>

238 │ {

239 │ entity.HasKey(e => e.Ma);

240 │

241 │ entity.Property(e => e.NgaySua).HasColumnType("datetime");

242 │ entity.Property(e => e.NgayTao)

243 │ .HasDefaultValueSql("(getdate())")

244 │ .HasColumnType("datetime");

245 │ entity.Property(e => e.NgayXoa).HasColumnType("datetime");

246 │ entity.Property(e => e.Ten).HasMaxLength(100);

247 │ entity.Property(e => e.TrangThai).HasDefaultValueSql("((1))");

248 │ });

249 │

250 │ modelBuilder.Entity<GiamGia>(entity =>

251 │ {

252 │ entity.HasKey(e => e.Ma);

253 │

254 │ entity.Property(e => e.DenNgay).HasColumnType("datetime");

255 │ entity.Property(e => e.NgaySua).HasColumnType("datetime");

256 │ entity.Property(e => e.NgayTao)

257 │ .HasDefaultValueSql("(getdate())")

258 │ .HasColumnType("datetime");

259 │ entity.Property(e => e.NgayXoa).HasColumnType("datetime");

260 │ entity.Property(e => e.TenCt)

261 │ .HasMaxLength(100)

262 │ .HasColumnName("TenCT");

263 │ entity.Property(e => e.TrangThai).HasDefaultValueSql("((1))");

264 │ entity.Property(e => e.TuNgay).HasColumnType("datetime");

265 │ });

266 │

267 │ modelBuilder.Entity<HangHoa>(entity =>

268 │ {

269 │ entity.ToTable(tb => tb.HasTrigger("trg\_HangHoa\_ins"));

270 │

271 │ entity.Property(e => e.HangHoaId).HasColumnName("HangHoa\_ID");

272 │ entity.Property(e => e.Dvt).HasColumnName("DVT");

273 │ entity.Property(e => e.GhiChu).HasMaxLength(100);

274 │ entity.Property(e => e.GiaVon).HasColumnType("decimal(20, 4)");

275 │ entity.Property(e => e.HinhAnh).HasMaxLength(100);

276 │ entity.Property(e => e.MaKt)

277 │ .HasMaxLength(20)

278 │ .IsUnicode(false)

279 │ .HasColumnName("MaKT");

280 │ entity.Property(e => e.MaVach)

281 │ .HasMaxLength(20)

282 │ .IsUnicode(false);

283 │ entity.Property(e => e.NgaySua).HasColumnType("datetime");

284 │ entity.Property(e => e.NgayTao)

285 │ .HasDefaultValueSql("(getdate())")

286 │ .HasColumnType("datetime");

287 │ entity.Property(e => e.NgayXoa).HasColumnType("datetime");

288 │ entity.Property(e => e.TenHang).HasMaxLength(200);

289 │ entity.Property(e => e.TrangThai)

290 │ .IsRequired()

291 │ .HasDefaultValueSql("((1))");

292 │ });

293 │

294 │ modelBuilder.Entity<Hhimport>(entity =>

295 │ {

296 │ entity

297 │ .HasNoKey()

298 │ .ToTable("HHImport$");

299 │

300 │ entity.Property(e => e.Dvt)

301 │ .HasMaxLength(255)

302 │ .HasColumnName("DVT");

303 │ entity.Property(e => e.F10).HasMaxLength(255);

304 │ entity.Property(e => e.F11).HasMaxLength(255);

305 │ entity.Property(e => e.F12).HasMaxLength(255);

306 │ entity.Property(e => e.F13).HasMaxLength(255);

307 │ entity.Property(e => e.F14).HasMaxLength(255);

308 │ entity.Property(e => e.F15).HasMaxLength(255);

309 │ entity.Property(e => e.F16).HasMaxLength(255);

310 │ entity.Property(e => e.F17).HasMaxLength(255);

311 │ entity.Property(e => e.F18).HasMaxLength(255);

312 │ entity.Property(e => e.F19).HasMaxLength(255);

313 │ entity.Property(e => e.F20).HasMaxLength(255);

314 │ entity.Property(e => e.GiáBán).HasColumnName("Giá bán");

315 │ entity.Property(e => e.GiáVốn).HasColumnName("Giá vốn");

316 │ entity.Property(e => e.LớpHàngHoá)

317 │ .HasMaxLength(255)

318 │ .HasColumnName("Lớp hàng hoá");

319 │ entity.Property(e => e.Madvt)

320 │ .HasMaxLength(255)

321 │ .HasColumnName("MADVT");

322 │ entity.Property(e => e.MãBệnhViện)

323 │ .HasMaxLength(255)

324 │ .HasColumnName("Mã Bệnh viện");

325 │ entity.Property(e => e.MãCode)

326 │ .HasMaxLength(255)

327 │ .HasColumnName("Mã Code");

328 │ entity.Property(e => e.TênHàngHoá)

329 │ .HasMaxLength(255)

330 │ .HasColumnName("Tên Hàng hoá");

331 │ });

332 │

333 │ modelBuilder.Entity<HinhThucTt>(entity =>

334 │ {

335 │ entity.HasKey(e => e.HinhThucId);

336 │

337 │ entity.ToTable("HinhThuc\_TT");

338 │

339 │ entity.Property(e => e.HinhThucId).HasColumnName("HinhThuc\_ID");

340 │ entity.Property(e => e.NgayXoa).HasColumnType("datetime");

341 │ entity.Property(e => e.TenHinhThuc).HasMaxLength(50);

342 │ entity.Property(e => e.TrangThai)

343 │ .IsRequired()

344 │ .HasDefaultValueSql("((1))");

345 │ });

346 │

347 │ modelBuilder.Entity<HisLichSuChinhGia>(entity =>

348 │ {

349 │ entity.HasKey(e => e.Ma);

350 │

351 │ entity.ToTable("his\_LichSuChinhGia");

352 │

353 │ entity.Property(e => e.GhiChu).HasMaxLength(200);

354 │ entity.Property(e => e.GiaCu).HasColumnType("decimal(18, 4)");

355 │ entity.Property(e => e.GiaMoi).HasColumnType("decimal(18, 4)");

356 │ entity.Property(e => e.NgaySua).HasColumnType("datetime");

357 │ });

358 │

359 │ modelBuilder.Entity<HoaDonBan>(entity =>

360 │ {

361 │ entity.HasKey(e => e.HoaDonId);

362 │

363 │ entity.ToTable(tb => tb.HasTrigger("trg\_HoaDonBan\_del"));

364 │

365 │ entity.Property(e => e.HoaDonId).HasColumnName("HoaDon\_ID");

366 │ entity.Property(e => e.GhiChu).HasMaxLength(100);

367 │ entity.Property(e => e.GiaGiam).HasDefaultValueSql("((0))");

368 │ entity.Property(e => e.HinhThucTt).HasColumnName("HinhThucTT");

369 │ entity.Property(e => e.KhoaSo).HasDefaultValueSql("((0))");

370 │ entity.Property(e => e.MaVachThe).HasMaxLength(50);

371 │ entity.Property(e => e.NgayHd)

372 │ .HasColumnType("datetime")

373 │ .HasColumnName("Ngay\_HD");

374 │ entity.Property(e => e.NgaySua).HasColumnType("datetime");

375 │ entity.Property(e => e.NgayTao).HasColumnType("datetime");

376 │ entity.Property(e => e.NgayThanhToan).HasColumnType("datetime");

377 │ entity.Property(e => e.NgayXoa)

378 │ .HasDefaultValueSql("(getdate())")

379 │ .HasColumnType("datetime");

380 │ entity.Property(e => e.NguoiXoa).HasDefaultValueSql("((0))");

381 │ entity.Property(e => e.RequestId)

382 │ .HasMaxLength(50)

383 │ .IsUnicode(false)

384 │ .HasColumnName("RequestID");

385 │ entity.Property(e => e.SoSeri)

386 │ .HasMaxLength(50)

387 │ .IsUnicode(false)

388 │ .HasColumnName("So\_Seri");

389 │ entity.Property(e => e.TienThua).HasDefaultValueSql("((0))");

390 │ entity.Property(e => e.WebOrderId)

391 │ .HasMaxLength(50)

392 │ .IsUnicode(false);

393 │ });

394 │

395 │ modelBuilder.Entity<KkBienBanKiemKe>(entity =>

396 │ {

397 │ entity.HasKey(e => e.Ma);

398 │

399 │ entity.ToTable("kk\_BienBanKiemKe", tb =>

400 │ {

401 │ tb.HasTrigger("tg\_kk\_BienBanKiemKe\_ins");

402 │ tb.HasTrigger("trg\_kk\_BienBanKiemKe\_del");

403 │ });

404 │

405 │ entity.Property(e => e.GhiChu).HasMaxLength(150);

406 │ entity.Property(e => e.NgayKiemKe).HasColumnType("datetime");

407 │ entity.Property(e => e.NgaySua).HasColumnType("datetime");

408 │ entity.Property(e => e.NgayTao)

409 │ .HasDefaultValueSql("(getdate())")

410 │ .HasColumnType("datetime");

411 │ entity.Property(e => e.NgayXoa).HasColumnType("datetime");

412 │ entity.Property(e => e.NguoiKiemKe).HasMaxLength(50);

413 │ entity.Property(e => e.SoPhieu).HasMaxLength(50);

414 │ });

415 │

416 │ modelBuilder.Entity<KkChiTietKiemKe>(entity =>

417 │ {

418 │ entity.HasKey(e => e.Ma);

419 │

420 │ entity.ToTable("kk\_ChiTietKiemKe");

421 │

422 │ entity.Property(e => e.GhiChu).HasMaxLength(2000);

423 │ entity.Property(e => e.NgaySua).HasColumnType("datetime");

424 │ entity.Property(e => e.NgayTao)

425 │ .HasDefaultValueSql("(getdate())")

426 │ .HasColumnType("datetime");

427 │ entity.Property(e => e.NgayXoa).HasColumnType("datetime");

428 │ entity.Property(e => e.SoLuongKiemKe).HasColumnType("decimal(20, 4)");

429 │ });

430 │

431 │ modelBuilder.Entity<KkChiTietTonKho>(entity =>

432 │ {

433 │ entity.HasKey(e => e.Ma);

434 │

435 │ entity.ToTable("kk\_ChiTietTonKho");

436 │

437 │ entity.Property(e => e.GhiChu).HasMaxLength(2000);

438 │ entity.Property(e => e.NgaySua).HasColumnType("datetime");

439 │ entity.Property(e => e.NgayTao)

440 │ .HasDefaultValueSql("(getdate())")

441 │ .HasColumnType("datetime");

442 │ entity.Property(e => e.SoLuongTon).HasColumnType("decimal(20, 4)");

443 │ });

444 │

445 │ modelBuilder.Entity<LichSuGia>(entity =>

446 │ {

447 │ entity.HasKey(e => e.Ma);

448 │

449 │ entity.ToTable(tb => tb.HasTrigger("trg\_LichSuGia\_del"));

450 │

451 │ entity.Property(e => e.LyDo).HasMaxLength(250);

452 │ entity.Property(e => e.NgaySua).HasColumnType("datetime");

453 │ entity.Property(e => e.NgayTao)

454 │ .HasDefaultValueSql("(getdate())")

455 │ .HasColumnType("datetime");

456 │ entity.Property(e => e.NgayXoa).HasColumnType("datetime");

457 │ entity.Property(e => e.TieuDe).HasMaxLength(250);

458 │ });

459 │

460 │ modelBuilder.Entity<LopHang>(entity =>

461 │ {

462 │ entity.HasKey(e => e.LopId);

463 │

464 │ entity.Property(e => e.LopId).HasColumnName("Lop\_ID");

465 │ entity.Property(e => e.NgaySua).HasColumnType("datetime");

466 │ entity.Property(e => e.NgayTao)

467 │ .HasDefaultValueSql("(getdate())")

468 │ .HasColumnType("datetime");

469 │ entity.Property(e => e.NgayXoa).HasColumnType("datetime");

470 │ entity.Property(e => e.TenLop).HasMaxLength(50);

471 │ entity.Property(e => e.TrangThai)

472 │ .IsRequired()

473 │ .HasDefaultValueSql("((1))");

474 │ });

475 │

476 │ modelBuilder.Entity<NccChti>(entity =>

477 │ {

478 │ entity

479 │ .HasNoKey()

480 │ .ToTable("'NCC CHTI$'");

481 │

482 │ entity.Property(e => e.MãHis)

483 │ .HasMaxLength(255)

484 │ .HasColumnName("Mã His");

485 │ entity.Property(e => e.MãNccFast)

486 │ .HasMaxLength(255)

487 │ .HasColumnName("Mã NCC Fast");

488 │ entity.Property(e => e.TênNccFast)

489 │ .HasMaxLength(255)

490 │ .HasColumnName("Tên NCC Fast");

491 │ entity.Property(e => e.TênNhàCungCấpHis)

492 │ .HasMaxLength(255)

493 │ .HasColumnName("Tên nhà cung cấp HIS#");

494 │ });

495 │

496 │ modelBuilder.Entity<NccDược>(entity =>

497 │ {

498 │ entity

499 │ .HasNoKey()

500 │ .ToTable("'NCC DƯỢC$'");

501 │

502 │ entity.Property(e => e.F10).HasMaxLength(255);

503 │ entity.Property(e => e.F11).HasMaxLength(255);

504 │ entity.Property(e => e.F12).HasMaxLength(255);

505 │ entity.Property(e => e.F13).HasMaxLength(255);

506 │ entity.Property(e => e.F14).HasMaxLength(255);

507 │ entity.Property(e => e.F15).HasMaxLength(255);

508 │ entity.Property(e => e.F16).HasMaxLength(255);

509 │ entity.Property(e => e.F17).HasMaxLength(255);

510 │ entity.Property(e => e.F18).HasMaxLength(255);

511 │ entity.Property(e => e.F19).HasMaxLength(255);

512 │ entity.Property(e => e.F20).HasMaxLength(255);

513 │ entity.Property(e => e.F21).HasMaxLength(255);

514 │ entity.Property(e => e.F22).HasMaxLength(255);

515 │ entity.Property(e => e.F5).HasMaxLength(255);

516 │ entity.Property(e => e.F6).HasMaxLength(255);

517 │ entity.Property(e => e.F7).HasMaxLength(255);

518 │ entity.Property(e => e.F8).HasMaxLength(255);

519 │ entity.Property(e => e.F9).HasMaxLength(255);

520 │ entity.Property(e => e.MãNccFast)

521 │ .HasMaxLength(255)

522 │ .HasColumnName("Mã NCC Fast");

523 │ entity.Property(e => e.MãNccHis)

524 │ .HasMaxLength(255)

525 │ .HasColumnName("Mã NCC HIS");

526 │ entity.Property(e => e.TênNccFast)

527 │ .HasMaxLength(255)

528 │ .HasColumnName("Tên NCC Fast");

529 │ entity.Property(e => e.TênNhàCungCấpHis)

530 │ .HasMaxLength(255)

531 │ .HasColumnName("Tên nhà cung cấp HIS#");

532 │ });

533 │

534 │ modelBuilder.Entity<NguoiDung>(entity =>

535 │ {

536 │ entity.HasKey(e => e.NId);

537 │

538 │ entity.Property(e => e.NId).HasColumnName("nID");

539 │ entity.Property(e => e.Anh).HasMaxLength(50);

540 │ entity.Property(e => e.DepartmentId)

541 │ .HasMaxLength(50)

542 │ .HasColumnName("DepartmentID");

543 │ entity.Property(e => e.Hoten).HasMaxLength(100);

544 │ entity.Property(e => e.Image).HasMaxLength(100);

545 │ entity.Property(e => e.KhoHang)

546 │ .HasMaxLength(200)

547 │ .IsUnicode(false);

548 │ entity.Property(e => e.Matkhau)

549 │ .HasMaxLength(200)

550 │ .IsUnicode(false);

551 │ entity.Property(e => e.Msnv)

552 │ .HasMaxLength(30)

553 │ .IsUnicode(false)

554 │ .HasColumnName("MSNV");

555 │ entity.Property(e => e.NgaySua).HasColumnType("datetime");

556 │ entity.Property(e => e.NgayTao)

557 │ .HasDefaultValueSql("(getdate())")

558 │ .HasColumnType("datetime");

559 │ entity.Property(e => e.NgayXoa).HasColumnType("datetime");

560 │ entity.Property(e => e.Photo).HasColumnType("image");

561 │ entity.Property(e => e.Tendangnhap)

562 │ .HasMaxLength(20)

563 │ .IsUnicode(false);

564 │ entity.Property(e => e.TrangThai)

565 │ .IsRequired()

566 │ .HasDefaultValueSql("((1))");

567 │ });

568 │

569 │ modelBuilder.Entity<NhaCc>(entity =>

570 │ {

571 │ entity.HasKey(e => e.Ma);

572 │

573 │ entity.ToTable("NhaCC", tb => tb.HasTrigger("trg\_NhaCC\_ins"));

574 │

575 │ entity.Property(e => e.Cccd)

576 │ .HasMaxLength(30)

577 │ .IsUnicode(false)

578 │ .HasColumnName("CCCD");

579 │ entity.Property(e => e.DiaChi).HasMaxLength(100);

580 │ entity.Property(e => e.DienThoai).HasMaxLength(20);

581 │ entity.Property(e => e.Email).HasMaxLength(50);

582 │ entity.Property(e => e.MaKt)

583 │ .HasMaxLength(20)

584 │ .HasColumnName("MaKT");

585 │ entity.Property(e => e.Mst)

586 │ .HasMaxLength(30)

587 │ .HasColumnName("MST");

588 │ entity.Property(e => e.NgaySua).HasColumnType("datetime");

589 │ entity.Property(e => e.NgayTao)

590 │ .HasDefaultValueSql("(getdate())")

591 │ .HasColumnType("datetime");

592 │ entity.Property(e => e.NgayXoa).HasColumnType("datetime");

593 │ entity.Property(e => e.NguoiLienHe).HasMaxLength(80);

594 │ entity.Property(e => e.TenNhaCc)

595 │ .HasMaxLength(100)

596 │ .HasColumnName("TenNhaCC");

597 │ entity.Property(e => e.TrangThai)

598 │ .IsRequired()

599 │ .HasDefaultValueSql("((1))");

600 │ });

601 │

602 │ modelBuilder.Entity<NhapHang>(entity =>

603 │ {

604 │ entity.HasKey(e => e.Ma);

605 │

606 │ entity.ToTable(tb =>

607 │ {

608 │ tb.HasTrigger("trg\_NhapHang\_del");

609 │ tb.HasTrigger("trg\_NhapHang\_ins");

610 │ });

611 │

612 │ entity.Property(e => e.GhiChu).HasMaxLength(500);

613 │ entity.Property(e => e.KhoaSo).HasDefaultValueSql("((0))");

614 │ entity.Property(e => e.MaLoaiNx).HasColumnName("MaLoaiNX");

615 │ entity.Property(e => e.NgayHoaDon).HasColumnType("datetime");

616 │ entity.Property(e => e.NgayNhapHang).HasColumnType("datetime");

617 │ entity.Property(e => e.NgaySua).HasColumnType("datetime");

618 │ entity.Property(e => e.NgayTao)

619 │ .HasDefaultValueSql("(getdate())")

620 │ .HasColumnType("datetime");

621 │ entity.Property(e => e.NgayXoa).HasColumnType("datetime");

622 │ entity.Property(e => e.NhaCc).HasColumnName("NhaCC");

623 │ entity.Property(e => e.SoHd)

624 │ .HasMaxLength(50)

625 │ .HasColumnName("SoHD");

626 │ entity.Property(e => e.SoHoaDonNcc)

627 │ .HasMaxLength(50)

628 │ .HasColumnName("SoHoaDonNCC");

629 │ entity.Property(e => e.ThueVat).HasColumnName("ThueVAT");

630 │ entity.Property(e => e.TienChietKhau).HasColumnType("decimal(20, 4)");

631 │ });

632 │

633 │ modelBuilder.Entity<SysDsbaoCao>(entity =>

634 │ {

635 │ entity.HasKey(e => e.Ma);

636 │

637 │ entity.ToTable("sys\_DSBaoCao");

638 │

639 │ entity.Property(e => e.DienGiai).HasMaxLength(100);

640 │ entity.Property(e => e.NgayXoa).HasColumnType("datetime");

641 │ entity.Property(e => e.TenBaoCao).HasMaxLength(50);

642 │ entity.Property(e => e.TrangThai).HasDefaultValueSql("((1))");

643 │ });

644 │

645 │ modelBuilder.Entity<SysDsdieuKienBaoCao>(entity =>

646 │ {

647 │ entity.HasKey(e => e.Ma);

648 │

649 │ entity.ToTable("sys\_DSDieuKien\_BaoCao");

650 │

651 │ entity.Property(e => e.CotHienThi)

652 │ .HasMaxLength(50)

653 │ .IsUnicode(false);

654 │ entity.Property(e => e.LoaiControl)

655 │ .HasMaxLength(50)

656 │ .IsUnicode(false);

657 │ entity.Property(e => e.NgayXoa).HasColumnType("datetime");

658 │ entity.Property(e => e.TrangThai).HasDefaultValueSql("((1))");

659 │ });

660 │

661 │ modelBuilder.Entity<SysMenu>(entity =>

662 │ {

663 │ entity.HasKey(e => e.Ma);

664 │

665 │ entity.ToTable("sys\_Menu");

666 │

667 │ entity.Property(e => e.Code).HasMaxLength(100);

668 │ entity.Property(e => e.DuongDan).HasMaxLength(300);

669 │ entity.Property(e => e.Icon).HasMaxLength(200);

670 │ entity.Property(e => e.IconWeb).HasMaxLength(100);

671 │ entity.Property(e => e.NgaySua).HasColumnType("datetime");

672 │ entity.Property(e => e.NgayTao)

673 │ .HasDefaultValueSql("(getdate())")

674 │ .HasColumnType("datetime");

675 │ entity.Property(e => e.NgayXoa).HasColumnType("datetime");

676 │ entity.Property(e => e.Ten).HasMaxLength(300);

677 │ entity.Property(e => e.ThamSo).HasMaxLength(300);

678 │ entity.Property(e => e.TrangThai).HasDefaultValueSql("((1))");

679 │ entity.Property(e => e.Url).HasMaxLength(100);

680 │ });

681 │

682 │ modelBuilder.Entity<SysPrinter>(entity =>

683 │ {

684 │ entity.HasKey(e => e.Ma);

685 │

686 │ entity.ToTable("sys\_Printer");

687 │

688 │ entity.Property(e => e.TenMayIn).HasMaxLength(100);

689 │ entity.Property(e => e.TenMayTinh).HasMaxLength(50);

690 │ });

691 │

692 │ modelBuilder.Entity<SysQuyen>(entity =>

693 │ {

694 │ entity.HasKey(e => e.Ma);

695 │

696 │ entity.ToTable("sys\_Quyen");

697 │

698 │ entity.Property(e => e.Code).HasMaxLength(50);

699 │ entity.Property(e => e.MoTa).HasMaxLength(300);

700 │ entity.Property(e => e.NgaySua).HasColumnType("datetime");

701 │ entity.Property(e => e.NgayTao)

702 │ .HasDefaultValueSql("(getdate())")

703 │ .HasColumnType("datetime");

704 │ entity.Property(e => e.NgayXoa).HasColumnType("datetime");

705 │ entity.Property(e => e.Ten).HasMaxLength(250);

706 │ entity.Property(e => e.TrangThai).HasDefaultValueSql("((1))");

707 │ });

708 │

709 │ modelBuilder.Entity<SysRole>(entity =>

710 │ {

711 │ entity.HasKey(e => e.Ma);

712 │

713 │ entity.ToTable("sys\_Role");

714 │

715 │ entity.Property(e => e.Code).HasMaxLength(50);

716 │ entity.Property(e => e.MoTa).HasMaxLength(300);

717 │ entity.Property(e => e.NgaySua).HasColumnType("datetime");

718 │ entity.Property(e => e.NgayTao)

719 │ .HasDefaultValueSql("(getdate())")

720 │ .HasColumnType("datetime");

721 │ entity.Property(e => e.NgayXoa).HasColumnType("datetime");

722 │ entity.Property(e => e.Ten).HasMaxLength(250);

723 │ entity.Property(e => e.TrangThai).HasDefaultValueSql("((1))");

724 │ });

725 │

726 │ modelBuilder.Entity<SysRoleMenu>(entity =>

727 │ {

728 │ entity.HasKey(e => e.Ma);

729 │

730 │ entity.ToTable("sys\_RoleMenu");

731 │

732 │ entity.Property(e => e.ChoPhepTruyCap).HasDefaultValueSql("((1))");

733 │ entity.Property(e => e.NgaySua).HasColumnType("datetime");

734 │ entity.Property(e => e.NgayTao)

735 │ .HasDefaultValueSql("(getdate())")

736 │ .HasColumnType("datetime");

737 │ entity.Property(e => e.NgayXoa).HasColumnType("datetime");

738 │ entity.Property(e => e.TrangThai).HasDefaultValueSql("((1))");

739 │ });

740 │

741 │ modelBuilder.Entity<SysRoleQuyen>(entity =>

742 │ {

743 │ entity.HasKey(e => e.Ma);

744 │

745 │ entity.ToTable("sys\_RoleQuyen");

746 │

747 │ entity.Property(e => e.NgaySua).HasColumnType("datetime");

748 │ entity.Property(e => e.NgayTao)

749 │ .HasDefaultValueSql("(getdate())")

750 │ .HasColumnType("datetime");

751 │ entity.Property(e => e.NgayXoa).HasColumnType("datetime");

752 │ entity.Property(e => e.TrangThai).HasDefaultValueSql("((1))");

753 │ });

754 │

755 │ modelBuilder.Entity<SysStatus>(entity =>

756 │ {

757 │ entity.HasKey(e => e.Id).HasName("PK\_sysStatus");

758 │

759 │ entity.ToTable("sys\_Status");

760 │

761 │ entity.Property(e => e.Id).HasColumnName("ID");

762 │ entity.Property(e => e.Page).HasMaxLength(50);

763 │ entity.Property(e => e.StatusMes).HasMaxLength(50);

764 │ });

765 │

766 │ modelBuilder.Entity<SysUserRole>(entity =>

767 │ {

768 │ entity.HasKey(e => e.Ma);

769 │

770 │ entity.ToTable("sys\_UserRole");

771 │

772 │ entity.Property(e => e.NgaySua).HasColumnType("datetime");

773 │ entity.Property(e => e.NgayTao)

774 │ .HasDefaultValueSql("(getdate())")

775 │ .HasColumnType("datetime");

776 │ entity.Property(e => e.NgayXoa).HasColumnType("datetime");

777 │ entity.Property(e => e.TrangThai).HasDefaultValueSql("((1))");

778 │ });

779 │

780 │ modelBuilder.Entity<TblChiTietXuatKho>(entity =>

781 │ {

782 │ entity.HasKey(e => e.Ma).HasName("PK\_tbl\_ChiTietPhieuXuat");

783 │

784 │ entity.ToTable("tbl\_ChiTietXuatKho");

785 │

786 │ entity.Property(e => e.GhiChu).HasMaxLength(500);

787 │ entity.Property(e => e.GiaVon).HasColumnType("decimal(20, 4)");

788 │ entity.Property(e => e.MaDvt).HasColumnName("MaDVT");

789 │ entity.Property(e => e.NgaySua).HasColumnType("datetime");

790 │ entity.Property(e => e.NgayTao)

791 │ .HasDefaultValueSql("(getdate())")

792 │ .HasColumnType("datetime");

793 │ entity.Property(e => e.NgayXoa).HasColumnType("datetime");

794 │ entity.Property(e => e.SoLuong).HasColumnType("decimal(20, 4)");

795 │ });

796 │

797 │ modelBuilder.Entity<TblCtKetCa>(entity =>

798 │ {

799 │ entity.HasKey(e => e.Ma);

800 │

801 │ entity.ToTable("tbl\_CT\_KetCa");

802 │

803 │ entity.Property(e => e.NgayBan).HasColumnType("datetime");

804 │ entity.Property(e => e.NgayKetCa).HasColumnType("datetime");

805 │ entity.Property(e => e.SoTien).HasColumnType("decimal(18, 0)");

806 │ });

807 │

808 │ modelBuilder.Entity<TblDanhMucKho>(entity =>

809 │ {

810 │ entity.HasKey(e => e.Ma);

811 │

812 │ entity.ToTable("tbl\_DanhMucKho");

813 │

814 │ entity.Property(e => e.Code)

815 │ .HasMaxLength(50)

816 │ .IsUnicode(false);

817 │ entity.Property(e => e.DienGiai).HasMaxLength(200);

818 │ entity.Property(e => e.NgaySua).HasColumnType("datetime");

819 │ entity.Property(e => e.NgayTao)

820 │ .HasDefaultValueSql("(getdate())")

821 │ .HasColumnType("datetime");

822 │ entity.Property(e => e.NgayXoa).HasColumnType("datetime");

823 │ entity.Property(e => e.TaiKhoanCo)

824 │ .HasMaxLength(20)

825 │ .IsUnicode(false);

826 │ entity.Property(e => e.TaiKhoanKho)

827 │ .HasMaxLength(20)

828 │ .IsUnicode(false);

829 │ entity.Property(e => e.Ten).HasMaxLength(150);

830 │ entity.Property(e => e.TrangThai).HasDefaultValueSql("((1))");

831 │ });

832 │

833 │ modelBuilder.Entity<TblGiaBanTheoKho>(entity =>

834 │ {

835 │ entity.HasKey(e => e.Ma);

836 │

837 │ entity.ToTable("tbl\_GiaBanTheoKho", tb => tb.HasTrigger("trg\_tbl\_GiaBanTheoKho\_Ins"));

838 │

839 │ entity.Property(e => e.GhiChu).HasMaxLength(100);

840 │ entity.Property(e => e.NgaySua).HasColumnType("datetime");

841 │ entity.Property(e => e.NgayTao)

842 │ .HasDefaultValueSql("(getdate())")

843 │ .HasColumnType("datetime");

844 │ entity.Property(e => e.NgayXoa).HasColumnType("datetime");

845 │ entity.Property(e => e.TrangThai).HasDefaultValueSql("((1))");

846 │ });

847 │

848 │ modelBuilder.Entity<TblGiuongBenh>(entity =>

849 │ {

850 │ entity.HasKey(e => e.Ma);

851 │

852 │ entity.ToTable("tbl\_GiuongBenh");

853 │

854 │ entity.Property(e => e.Code).HasMaxLength(20);

855 │ entity.Property(e => e.NgaySua).HasColumnType("datetime");

856 │ entity.Property(e => e.NgayTao)

857 │ .HasDefaultValueSql("(getdate())")

858 │ .HasColumnType("datetime");

859 │ entity.Property(e => e.NgayXoa).HasColumnType("datetime");

860 │ entity.Property(e => e.Ten).HasMaxLength(200);

861 │ entity.Property(e => e.TrangThai).HasDefaultValueSql("((1))");

862 │ });

863 │

864 │ modelBuilder.Entity<TblKyKho>(entity =>

865 │ {

866 │ entity.HasKey(e => e.Ma);

867 │

868 │ entity.ToTable("tbl\_KyKho");

869 │

870 │ entity.Property(e => e.NgaySua).HasColumnType("datetime");

871 │ entity.Property(e => e.NgayTao)

872 │ .HasDefaultValueSql("(getdate())")

873 │ .HasColumnType("datetime");

874 │ entity.Property(e => e.NgayXoa).HasColumnType("datetime");

875 │ });

876 │

877 │ modelBuilder.Entity<TblLoaiNhapXuat>(entity =>

878 │ {

879 │ entity.HasKey(e => e.Ma);

880 │

881 │ entity.ToTable("tbl\_LoaiNhapXuat");

882 │

883 │ entity.Property(e => e.Code).HasMaxLength(20);

884 │ entity.Property(e => e.NgaySua).HasColumnType("datetime");

885 │ entity.Property(e => e.NgayTao)

886 │ .HasDefaultValueSql("(getdate())")

887 │ .HasColumnType("datetime");

888 │ entity.Property(e => e.NgayXoa).HasColumnType("datetime");

889 │ entity.Property(e => e.NhapKho).HasComment("1: Nhập kho, 0:Xuất kho");

890 │ entity.Property(e => e.Ten).HasMaxLength(200);

891 │ });

892 │

893 │ modelBuilder.Entity<TblNgayKetCa>(entity =>

894 │ {

895 │ entity.HasKey(e => e.Ma).HasName("PK\_tblNgayKeCa");

896 │

897 │ entity.ToTable("tbl\_NgayKetCa");

898 │

899 │ entity.Property(e => e.NgayBanHang).HasColumnType("datetime");

900 │ entity.Property(e => e.NgaySua).HasColumnType("datetime");

901 │ entity.Property(e => e.NgayTao)

902 │ .HasDefaultValueSql("(getdate())")

903 │ .HasColumnType("datetime");

904 │ entity.Property(e => e.NgayXoa).HasColumnType("datetime");

905 │ });

906 │

907 │ modelBuilder.Entity<TblPhongBan>(entity =>

908 │ {

909 │ entity.HasKey(e => e.Ma);

910 │

911 │ entity.ToTable("tbl\_PhongBan");

912 │

913 │ entity.Property(e => e.Code).HasMaxLength(50);

914 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

915 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

916 │ entity.Property(e => e.GroupZalo).HasMaxLength(100);

917 │ entity.Property(e => e.MaCn).HasColumnName("MaCN");

918 │ entity.Property(e => e.MoTa).HasMaxLength(300);

919 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

920 │ entity.Property(e => e.Ten).HasMaxLength(200);

921 │ entity.Property(e => e.TenVietTat).HasMaxLength(100);

922 │ });

923 │

924 │ modelBuilder.Entity<TblPhongBenh>(entity =>

925 │ {

926 │ entity.HasKey(e => e.Ma);

927 │

928 │ entity.ToTable("tbl\_PhongBenh");

929 │

930 │ entity.Property(e => e.Code)

931 │ .HasMaxLength(20)

932 │ .IsUnicode(false);

933 │ entity.Property(e => e.NgaySua).HasColumnType("datetime");

934 │ entity.Property(e => e.NgayTao)

935 │ .HasDefaultValueSql("(getdate())")

936 │ .HasColumnType("datetime");

937 │ entity.Property(e => e.NgayXoa).HasColumnType("datetime");

938 │ entity.Property(e => e.Ten).HasMaxLength(200);

939 │ entity.Property(e => e.TrangThai).HasDefaultValueSql("((1))");

940 │ });

941 │

942 │ modelBuilder.Entity<TblThueVat>(entity =>

943 │ {

944 │ entity.HasKey(e => e.Ma);

945 │

946 │ entity.ToTable("tbl\_ThueVAT");

947 │

948 │ entity.Property(e => e.NgaySua).HasColumnType("datetime");

949 │ entity.Property(e => e.NgayTao)

950 │ .HasDefaultValueSql("(getdate())")

951 │ .HasColumnType("datetime");

952 │ entity.Property(e => e.NgayXoa).HasColumnType("datetime");

953 │ entity.Property(e => e.Ten).HasMaxLength(120);

954 │ entity.Property(e => e.TrangThai).HasDefaultValueSql("((1))");

955 │ });

956 │

957 │ modelBuilder.Entity<TblTransTrinm>(entity =>

958 │ {

959 │ entity.ToTable("tbl\_Trans\_TRINM");

960 │

961 │ entity.Property(e => e.Id).HasColumnName("ID");

962 │ entity.Property(e => e.Pcode)

963 │ .HasMaxLength(20)

964 │ .IsUnicode(false)

965 │ .HasColumnName("PCode");

966 │ entity.Property(e => e.ProductId).HasColumnName("ProductID");

967 │ entity.Property(e => e.TransCost).HasDefaultValueSql("((0))");

968 │ entity.Property(e => e.TransDisCount).HasDefaultValueSql("((0))");

969 │ entity.Property(e => e.TransQty).HasDefaultValueSql("((0))");

970 │ entity.Property(e => e.TransTotal).HasComputedColumnSql("([TransQty]\*[TransCost]-([TransDisCount]\*([TransQty...

971 │ });

972 │

973 │ modelBuilder.Entity<TblXuatKho>(entity =>

974 │ {

975 │ entity.HasKey(e => e.Ma).HasName("PK\_tbl\_XuatKhoMayBanHang");

976 │

977 │ entity.ToTable("tbl\_XuatKho", tb =>

978 │ {

979 │ tb.HasTrigger("tg\_tbl\_XuatKho\_ins");

980 │ tb.HasTrigger("trg\_tbl\_XuatKho\_del");

981 │ });

982 │

983 │ entity.Property(e => e.GhiChu).HasMaxLength(500);

984 │ entity.Property(e => e.KhoaSo).HasDefaultValueSql("((0))");

985 │ entity.Property(e => e.MaLoaiNx).HasColumnName("MaLoaiNX");

986 │ entity.Property(e => e.NgaySua).HasColumnType("datetime");

987 │ entity.Property(e => e.NgayTao)

988 │ .HasDefaultValueSql("(getdate())")

989 │ .HasColumnType("datetime");

990 │ entity.Property(e => e.NgayXoa).HasColumnType("datetime");

991 │ entity.Property(e => e.NgayXuat).HasColumnType("datetime");

992 │ entity.Property(e => e.SoPhieu)

993 │ .HasMaxLength(30)

994 │ .IsUnicode(false);

995 │ });

996 │

997 │ modelBuilder.Entity<ThongTinShop>(entity =>

998 │ {

999 │ entity.HasKey(e => e.Ma);

1000 │

1001 │ entity.Property(e => e.DiaChi).HasMaxLength(150);

1002 │ entity.Property(e => e.DienThoai).HasMaxLength(15);

1003 │ entity.Property(e => e.Logo).HasColumnType("image");

1004 │ entity.Property(e => e.Ten).HasMaxLength(50);

1005 │ });

1006 │

1007 │ modelBuilder.Entity<TmpOrder>(entity =>

1008 │ {

1009 │ entity.HasKey(e => e.Ma);

1010 │

1011 │ entity.ToTable("tmp\_Order");

1012 │

1013 │ entity.Property(e => e.CodePhong).HasMaxLength(50);

1014 │ entity.Property(e => e.SoDienThoai)

1015 │ .HasMaxLength(20)

1016 │ .IsUnicode(false);

1017 │ entity.Property(e => e.TenKh)

1018 │ .HasMaxLength(50)

1019 │ .HasColumnName("TenKH");

1020 │ entity.Property(e => e.Uuid)

1021 │ .HasMaxLength(50)

1022 │ .HasColumnName("UUID");

1023 │ });

1024 │

1025 │ modelBuilder.Entity<TmpOrderDetail>(entity =>

1026 │ {

1027 │ entity.ToTable("tmp\_OrderDetail");

1028 │

1029 │ entity.Property(e => e.Id)

1030 │ .HasMaxLength(50)

1031 │ .HasColumnName("id");

1032 │ entity.Property(e => e.ConfirmedTime)

1033 │ .HasColumnType("datetime")

1034 │ .HasColumnName("confirmed\_time");

1035 │ entity.Property(e => e.Danhsachmon).HasColumnName("danhsachmon");

1036 │ entity.Property(e => e.MaHoaDon).HasColumnName("ma\_hoa\_don");

1037 │ entity.Property(e => e.MaKho).HasColumnName("ma\_kho");

1038 │ entity.Property(e => e.MaNganh).HasColumnName("ma\_nganh");

1039 │ entity.Property(e => e.Note)

1040 │ .HasMaxLength(50)

1041 │ .HasColumnName("note");

1042 │ entity.Property(e => e.OrderTime)

1043 │ .HasColumnType("datetime")

1044 │ .HasColumnName("order\_time");

1045 │ entity.Property(e => e.Status).HasColumnName("status");

1046 │ });

1047 │

1048 │ modelBuilder.Entity<TmpUpdateTonKho>(entity =>

1049 │ {

1050 │ entity

1051 │ .HasNoKey()

1052 │ .ToTable("tmpUpdateTonKho");

1053 │

1054 │ entity.Property(e => e.Dvt)

1055 │ .HasMaxLength(50)

1056 │ .HasColumnName("DVT");

1057 │ entity.Property(e => e.LopHang).HasMaxLength(50);

1058 │ entity.Property(e => e.MaVach).HasMaxLength(50);

1059 │ entity.Property(e => e.TenHangHoa).HasMaxLength(500);

1060 │ });

1061 │

1062 │ modelBuilder.Entity<TonDauKy1>(entity =>

1063 │ {

1064 │ entity

1065 │ .HasNoKey()

1066 │ .ToTable("TonDauKy1$");

1067 │

1068 │ entity.Property(e => e.Dvt)

1069 │ .HasMaxLength(255)

1070 │ .HasColumnName("DVT");

1071 │ entity.Property(e => e.HàmLượng)

1072 │ .HasMaxLength(255)

1073 │ .HasColumnName("Hàm lượng");

1074 │ entity.Property(e => e.MãBv)

1075 │ .HasMaxLength(255)

1076 │ .HasColumnName("Mã BV");

1077 │ entity.Property(e => e.MãCũ)

1078 │ .HasMaxLength(255)

1079 │ .HasColumnName("Mã cũ");

1080 │ entity.Property(e => e.NgàyHh)

1081 │ .HasColumnType("datetime")

1082 │ .HasColumnName("Ngày HH");

1083 │ entity.Property(e => e.Stt).HasColumnName("stt");

1084 │ entity.Property(e => e.SốLô)

1085 │ .HasMaxLength(255)

1086 │ .HasColumnName("Số lô");

1087 │ entity.Property(e => e.TiềnTồnCk).HasColumnName("Tiền Tồn CK");

1088 │ entity.Property(e => e.TiềnTồnĐk).HasColumnName("Tiền tồn ĐK");

1089 │ entity.Property(e => e.TênDượcVậtTư)

1090 │ .HasMaxLength(255)

1091 │ .HasColumnName("Tên Dược-Vật tư");

1092 │ entity.Property(e => e.TênHoạtChất)

1093 │ .HasMaxLength(255)

1094 │ .HasColumnName("Tên hoạt chất");

1095 │ entity.Property(e => e.TồnCk).HasColumnName("Tồn CK");

1096 │ entity.Property(e => e.TồnĐk).HasColumnName("Tồn ĐK");

1097 │ entity.Property(e => e.ĐơnGiá).HasColumnName("Đơn giá");

1098 │ });

1099 │

1100 │ modelBuilder.Entity<TonDayKy2>(entity =>

1101 │ {

1102 │ entity

1103 │ .HasNoKey()

1104 │ .ToTable("TonDayKy2$");

1105 │

1106 │ entity.Property(e => e.Dvt)

1107 │ .HasMaxLength(255)

1108 │ .HasColumnName("DVT");

1109 │ entity.Property(e => e.HàmLượng)

1110 │ .HasMaxLength(255)

1111 │ .HasColumnName("Hàm lượng");

1112 │ entity.Property(e => e.MãBv)

1113 │ .HasMaxLength(255)

1114 │ .HasColumnName("Mã BV");

1115 │ entity.Property(e => e.MãCũ)

1116 │ .HasMaxLength(255)

1117 │ .HasColumnName("Mã cũ");

1118 │ entity.Property(e => e.NgàyHh)

1119 │ .HasColumnType("datetime")

1120 │ .HasColumnName("Ngày HH");

1121 │ entity.Property(e => e.Stt).HasColumnName("STT");

1122 │ entity.Property(e => e.SốLô)

1123 │ .HasMaxLength(255)

1124 │ .HasColumnName("Số lô");

1125 │ entity.Property(e => e.TiềnTồnCk).HasColumnName("Tiền Tồn CK");

1126 │ entity.Property(e => e.TiềnTồnĐk).HasColumnName("Tiền tồn ĐK");

1127 │ entity.Property(e => e.TênDượcVậtTư)

1128 │ .HasMaxLength(255)

1129 │ .HasColumnName("Tên Dược-Vật tư");

1130 │ entity.Property(e => e.TênHoạtChất)

1131 │ .HasMaxLength(255)

1132 │ .HasColumnName("Tên hoạt chất");

1133 │ entity.Property(e => e.TồnCk).HasColumnName("Tồn CK");

1134 │ entity.Property(e => e.TồnĐk).HasColumnName("Tồn ĐK");

1135 │ entity.Property(e => e.ĐơnGiá).HasColumnName("Đơn giá");

1136 │ });

1137 │

1138 │ modelBuilder.Entity<TonKho>(entity =>

1139 │ {

1140 │ entity.HasKey(e => e.Ma);

1141 │

1142 │ entity.Property(e => e.GiaVon).HasColumnType("decimal(18, 4)");

1143 │ entity.Property(e => e.NamThang)

1144 │ .HasMaxLength(6)

1145 │ .IsUnicode(false);

1146 │ entity.Property(e => e.NgayXoa).HasColumnType("datetime");

1147 │ entity.Property(e => e.SlbanTrongKy)

1148 │ .HasDefaultValueSql("((0))")

1149 │ .HasColumnType("decimal(20, 4)")

1150 │ .HasColumnName("SLBanTrongKy");

1151 │ entity.Property(e => e.SlnhapTrongKy)

1152 │ .HasDefaultValueSql("((0))")

1153 │ .HasColumnType("decimal(20, 4)")

1154 │ .HasColumnName("SLNhapTrongKy");

1155 │ entity.Property(e => e.SltonCuoiKy)

1156 │ .HasDefaultValueSql("((0))")

1157 │ .HasColumnType("decimal(20, 4)")

1158 │ .HasColumnName("SLTonCuoiKy");

1159 │ entity.Property(e => e.SltonDauKy)

1160 │ .HasDefaultValueSql("((0))")

1161 │ .HasColumnType("decimal(20, 4)")

1162 │ .HasColumnName("SLTonDauKy");

1163 │ entity.Property(e => e.SlxuatKhac)

1164 │ .HasColumnType("decimal(20, 4)")

1165 │ .HasColumnName("SLXuatKhac");

1166 │ entity.Property(e => e.SlxuatTieuDung)

1167 │ .HasColumnType("decimal(20, 4)")

1168 │ .HasColumnName("SLXuatTieuDung");

1169 │ entity.Property(e => e.SlxuatTra)

1170 │ .HasColumnType("decimal(18, 4)")

1171 │ .HasColumnName("SLXuatTra");

1172 │ });

1173 │

1174 │ modelBuilder.Entity<TonKhoBanHang>(entity =>

1175 │ {

1176 │ entity.HasKey(e => new { e.NamThang, e.MaKho, e.MaHangHoa }).HasName("PK\_TonKhoBanHang\_1");

1177 │

1178 │ entity.Property(e => e.NamThang)

1179 │ .HasMaxLength(6)

1180 │ .IsUnicode(false);

1181 │ entity.Property(e => e.Ma).ValueGeneratedOnAdd();

1182 │ entity.Property(e => e.SlbanTrongKy)

1183 │ .HasColumnType("decimal(20, 4)")

1184 │ .HasColumnName("SLBanTrongKy");

1185 │ entity.Property(e => e.SlnhapTrongKy)

1186 │ .HasColumnType("decimal(20, 4)")

1187 │ .HasColumnName("SLNhapTrongKy");

1188 │ entity.Property(e => e.SltonCuoiKy)

1189 │ .HasColumnType("decimal(20, 4)")

1190 │ .HasColumnName("SLTonCuoiKy");

1191 │ entity.Property(e => e.SltonDauKy)

1192 │ .HasColumnType("decimal(20, 4)")

1193 │ .HasColumnName("SLTonDauKy");

1194 │ entity.Property(e => e.SlxuatKhac)

1195 │ .HasColumnType("decimal(20, 4)")

1196 │ .HasColumnName("SLXuatKhac");

1197 │ entity.Property(e => e.SlxuatTieuDung)

1198 │ .HasColumnType("decimal(20, 4)")

1199 │ .HasColumnName("SLXuatTieuDung");

1200 │ entity.Property(e => e.SlxuatTra)

1201 │ .HasColumnType("decimal(18, 4)")

1202 │ .HasColumnName("SLXuatTra");

1203 │ });

1204 │

1205 │ modelBuilder.Entity<TonKhoK>(entity =>

1206 │ {

1207 │ entity.HasKey(e => e.Ma);

1208 │

1209 │ entity.ToTable("TonKho\_K");

1210 │

1211 │ entity.Property(e => e.NamThang)

1212 │ .HasMaxLength(6)

1213 │ .IsUnicode(false);

1214 │ entity.Property(e => e.SlbanTrongKy)

1215 │ .HasDefaultValueSql("((0))")

1216 │ .HasColumnName("SLBanTrongKy");

1217 │ entity.Property(e => e.SlnhapTrongKy)

1218 │ .HasDefaultValueSql("((0))")

1219 │ .HasColumnName("SLNhapTrongKy");

1220 │ entity.Property(e => e.SltonCuoiKy)

1221 │ .HasDefaultValueSql("((0))")

1222 │ .HasColumnName("SLTonCuoiKy");

1223 │ entity.Property(e => e.SltonDauKy)

1224 │ .HasDefaultValueSql("((0))")

1225 │ .HasColumnName("SLTonDauKy");

1226 │ entity.Property(e => e.SlxuatKhac).HasColumnName("SLXuatKhac");

1227 │ entity.Property(e => e.SlxuatTieuDung).HasColumnName("SLXuatTieuDung");

1228 │ });

1229 │

1230 │ modelBuilder.Entity<TonKhoTuNgayDenNgay>(entity =>

1231 │ {

1232 │ entity.HasKey(e => e.Ma);

1233 │

1234 │ entity.ToTable("TonKho\_TuNgay\_DenNgay");

1235 │

1236 │ entity.Property(e => e.NamThang)

1237 │ .HasMaxLength(6)

1238 │ .IsUnicode(false);

1239 │ entity.Property(e => e.SlbanTrongKy1)

1240 │ .HasDefaultValueSql("((0))")

1241 │ .HasColumnName("SLBanTrongKy1");

1242 │ entity.Property(e => e.SlbanTrongKy2)

1243 │ .HasDefaultValueSql("((0))")

1244 │ .HasColumnName("SLBanTrongKy2");

1245 │ entity.Property(e => e.SlnhapTrongKy1)

1246 │ .HasDefaultValueSql("((0))")

1247 │ .HasColumnName("SLNhapTrongKy1");

1248 │ entity.Property(e => e.SlnhapTrongKy2)

1249 │ .HasDefaultValueSql("((0))")

1250 │ .HasColumnName("SLNhapTrongKy2");

1251 │ entity.Property(e => e.SltonCuoiKy1)

1252 │ .HasComputedColumnSql("(([SLTonDauKy1]+[SLNhapTrongKy1])-[SLBanTrongKy1])", false)

1253 │ .HasColumnName("SLTonCuoiKy1");

1254 │ entity.Property(e => e.SltonCuoiKy2)

1255 │ .HasComputedColumnSql("(((([SLTonDauKy1]+[SLNhapTrongKy1])-[SLBanTrongKy1])+[SLNhapTrongKy2])-[SLBanTron...

1256 │ .HasColumnName("SLTonCuoiKy2");

1257 │ entity.Property(e => e.SltonDauKy1)

1258 │ .HasDefaultValueSql("((0))")

1259 │ .HasColumnName("SLTonDauKy1");

1260 │ });

1261 │

1262 │ modelBuilder.Entity<VChiTietKiemKe>(entity =>

1263 │ {

1264 │ entity

1265 │ .HasNoKey()

1266 │ .ToView("vChiTietKiemKe");

1267 │

1268 │ entity.Property(e => e.GhiChu).HasMaxLength(2000);

1269 │ entity.Property(e => e.MaKt)

1270 │ .HasMaxLength(20)

1271 │ .IsUnicode(false)

1272 │ .HasColumnName("MaKT");

1273 │ entity.Property(e => e.MaVach)

1274 │ .HasMaxLength(20)

1275 │ .IsUnicode(false);

1276 │ entity.Property(e => e.NgayKiemKe).HasColumnType("datetime");

1277 │ entity.Property(e => e.NguoiKiemKe).HasMaxLength(50);

1278 │ entity.Property(e => e.SoLuongKiemKe).HasColumnType("decimal(20, 4)");

1279 │ entity.Property(e => e.SoPhieu).HasMaxLength(50);

1280 │ entity.Property(e => e.TenDvt)

1281 │ .HasMaxLength(100)

1282 │ .HasColumnName("TenDVT");

1283 │ entity.Property(e => e.TenHang).HasMaxLength(200);

1284 │ entity.Property(e => e.TenKho).HasMaxLength(150);

1285 │ entity.Property(e => e.TenLop).HasMaxLength(50);

1286 │ });

1287 │

1288 │ modelBuilder.Entity<VChiTietNhapHang>(entity =>

1289 │ {

1290 │ entity

1291 │ .HasNoKey()

1292 │ .ToView("vChiTietNhapHang");

1293 │

1294 │ entity.Property(e => e.ChietKhauPt)

1295 │ .HasColumnType("decimal(20, 4)")

1296 │ .HasColumnName("ChietKhauPT");

1297 │ entity.Property(e => e.ChietKhauTm)

1298 │ .HasColumnType("decimal(31, 15)")

1299 │ .HasColumnName("ChietKhauTM");

1300 │ entity.Property(e => e.GhiChu).HasMaxLength(500);

1301 │ entity.Property(e => e.GiaVon).HasColumnType("decimal(20, 4)");

1302 │ entity.Property(e => e.HanSd)

1303 │ .HasColumnType("datetime")

1304 │ .HasColumnName("HanSD");

1305 │ entity.Property(e => e.MaDvt).HasColumnName("MaDVT");

1306 │ entity.Property(e => e.MaKt)

1307 │ .HasMaxLength(20)

1308 │ .IsUnicode(false)

1309 │ .HasColumnName("MaKT");

1310 │ entity.Property(e => e.MaKtncc)

1311 │ .HasMaxLength(20)

1312 │ .HasColumnName("MaKTNCC");

1313 │ entity.Property(e => e.MaVach)

1314 │ .HasMaxLength(20)

1315 │ .IsUnicode(false);

1316 │ entity.Property(e => e.NgayHoaDon).HasColumnType("datetime");

1317 │ entity.Property(e => e.NgayNhapHang).HasColumnType("datetime");

1318 │ entity.Property(e => e.NgayTao).HasColumnType("datetime");

1319 │ entity.Property(e => e.NhaCc).HasColumnName("NhaCC");

1320 │ entity.Property(e => e.SoHd)

1321 │ .HasMaxLength(50)

1322 │ .HasColumnName("SoHD");

1323 │ entity.Property(e => e.SoHoaDonNcc)

1324 │ .HasMaxLength(50)

1325 │ .HasColumnName("SoHoaDonNCC");

1326 │ entity.Property(e => e.SoLuong).HasColumnType("decimal(20, 4)");

1327 │ entity.Property(e => e.TenDvt)

1328 │ .HasMaxLength(100)

1329 │ .HasColumnName("TenDVT");

1330 │ entity.Property(e => e.TenHang).HasMaxLength(200);

1331 │ entity.Property(e => e.TenKho).HasMaxLength(150);

1332 │ entity.Property(e => e.TenLop).HasMaxLength(50);

1333 │ entity.Property(e => e.TenNhaCc)

1334 │ .HasMaxLength(100)

1335 │ .HasColumnName("TenNhaCC");

1336 │ entity.Property(e => e.ThanhTien).HasColumnType("decimal(20, 4)");

1337 │ entity.Property(e => e.ThanhTienSauCk)

1338 │ .HasColumnType("decimal(20, 4)")

1339 │ .HasColumnName("ThanhTienSauCK");

1340 │ entity.Property(e => e.ThanhTienSauVat)

1341 │ .HasColumnType("decimal(20, 4)")

1342 │ .HasColumnName("ThanhTienSauVAT");

1343 │ entity.Property(e => e.Vat).HasColumnName("VAT");

1344 │ });

1345 │

1346 │ modelBuilder.Entity<VChiTietXuatKho>(entity =>

1347 │ {

1348 │ entity

1349 │ .HasNoKey()

1350 │ .ToView("vChiTietXuatKho");

1351 │

1352 │ entity.Property(e => e.GhiChu).HasMaxLength(500);

1353 │ entity.Property(e => e.GhiChuChiTiet).HasMaxLength(500);

1354 │ entity.Property(e => e.GiaVon).HasColumnType("decimal(20, 4)");

1355 │ entity.Property(e => e.KhoChuyenDen).HasMaxLength(150);

1356 │ entity.Property(e => e.MaDvt).HasColumnName("MaDVT");

1357 │ entity.Property(e => e.MaKt)

1358 │ .HasMaxLength(20)

1359 │ .IsUnicode(false)

1360 │ .HasColumnName("MaKT");

1361 │ entity.Property(e => e.MaVach)

1362 │ .HasMaxLength(20)

1363 │ .IsUnicode(false);

1364 │ entity.Property(e => e.NgayXuat).HasColumnType("datetime");

1365 │ entity.Property(e => e.SoLuong).HasColumnType("decimal(20, 4)");

1366 │ entity.Property(e => e.SoPhieu)

1367 │ .HasMaxLength(30)

1368 │ .IsUnicode(false);

1369 │ entity.Property(e => e.TenDvt)

1370 │ .HasMaxLength(100)

1371 │ .HasColumnName("TenDVT");

1372 │ entity.Property(e => e.TenHang).HasMaxLength(200);

1373 │ entity.Property(e => e.TenKho).HasMaxLength(150);

1374 │ entity.Property(e => e.TenLop).HasMaxLength(50);

1375 │ });

1376 │

1377 │ modelBuilder.Entity<VGiaBanTheoKho>(entity =>

1378 │ {

1379 │ entity

1380 │ .HasNoKey()

1381 │ .ToView("vGiaBanTheoKho");

1382 │

1383 │ entity.Property(e => e.Dvt).HasColumnName("DVT");

1384 │ entity.Property(e => e.MaVach)

1385 │ .HasMaxLength(20)

1386 │ .IsUnicode(false);

1387 │ entity.Property(e => e.SltonCuoiKy)

1388 │ .HasColumnType("decimal(20, 4)")

1389 │ .HasColumnName("SLTonCuoiKy");

1390 │ entity.Property(e => e.TenDvt)

1391 │ .HasMaxLength(100)

1392 │ .HasColumnName("TenDVT");

1393 │ entity.Property(e => e.TenHang).HasMaxLength(200);

1394 │ entity.Property(e => e.TenKho).HasMaxLength(150);

1395 │ entity.Property(e => e.TenLop).HasMaxLength(50);

1396 │ });

1397 │

1398 │ modelBuilder.Entity<VHangHoa>(entity =>

1399 │ {

1400 │ entity

1401 │ .HasNoKey()

1402 │ .ToView("vHangHoa");

1403 │

1404 │ entity.Property(e => e.Dvt).HasColumnName("DVT");

1405 │ entity.Property(e => e.GhiChu).HasMaxLength(100);

1406 │ entity.Property(e => e.GiaVon).HasColumnType("decimal(20, 4)");

1407 │ entity.Property(e => e.HangHoaId).HasColumnName("HangHoa\_ID");

1408 │ entity.Property(e => e.HinhAnh).HasMaxLength(100);

1409 │ entity.Property(e => e.MaKt)

1410 │ .HasMaxLength(20)

1411 │ .IsUnicode(false)

1412 │ .HasColumnName("MaKT");

1413 │ entity.Property(e => e.MaVach)

1414 │ .HasMaxLength(20)

1415 │ .IsUnicode(false);

1416 │ entity.Property(e => e.TenDvt)

1417 │ .HasMaxLength(100)

1418 │ .HasColumnName("TenDVT");

1419 │ entity.Property(e => e.TenHang).HasMaxLength(200);

1420 │ entity.Property(e => e.TenLop).HasMaxLength(50);

1421 │ });

1422 │

1423 │ modelBuilder.Entity<VNhaCc>(entity =>

1424 │ {

1425 │ entity

1426 │ .HasNoKey()

1427 │ .ToView("vNhaCC");

1428 │

1429 │ entity.Property(e => e.DiaChi).HasMaxLength(100);

1430 │ entity.Property(e => e.DienThoai).HasMaxLength(20);

1431 │ entity.Property(e => e.Email).HasMaxLength(50);

1432 │ entity.Property(e => e.Ma).ValueGeneratedOnAdd();

1433 │ entity.Property(e => e.MaKt)

1434 │ .HasMaxLength(20)

1435 │ .HasColumnName("MaKT");

1436 │ entity.Property(e => e.NgaySua).HasColumnType("datetime");

1437 │ entity.Property(e => e.NgayTao).HasColumnType("datetime");

1438 │ entity.Property(e => e.NgayXoa).HasColumnType("datetime");

1439 │ entity.Property(e => e.NguoiLienHe).HasMaxLength(80);

1440 │ entity.Property(e => e.TenNhaCc)

1441 │ .HasMaxLength(100)

1442 │ .HasColumnName("TenNhaCC");

1443 │ });

1444 │

1445 │ modelBuilder.Entity<VNhapHang>(entity =>

1446 │ {

1447 │ entity

1448 │ .HasNoKey()

1449 │ .ToView("vNhapHang");

1450 │

1451 │ entity.Property(e => e.GhiChu).HasMaxLength(500);

1452 │ entity.Property(e => e.NgayHoaDon).HasColumnType("datetime");

1453 │ entity.Property(e => e.NgayNhapHang).HasColumnType("datetime");

1454 │ entity.Property(e => e.NgayTao).HasColumnType("datetime");

1455 │ entity.Property(e => e.NhaCc).HasColumnName("NhaCC");

1456 │ entity.Property(e => e.SoHd)

1457 │ .HasMaxLength(50)

1458 │ .HasColumnName("SoHD");

1459 │ entity.Property(e => e.SoHoaDonNcc)

1460 │ .HasMaxLength(50)

1461 │ .HasColumnName("SoHoaDonNCC");

1462 │ entity.Property(e => e.TenKho).HasMaxLength(150);

1463 │ entity.Property(e => e.TenLoaiNhapXuat).HasMaxLength(200);

1464 │ entity.Property(e => e.TenNhaCc)

1465 │ .HasMaxLength(100)

1466 │ .HasColumnName("TenNhaCC");

1467 │ entity.Property(e => e.ThanhTien).HasColumnType("numeric(38, 6)");

1468 │ entity.Property(e => e.ThueVat).HasColumnName("ThueVAT");

1469 │ entity.Property(e => e.TienChietKhau).HasColumnType("decimal(20, 4)");

1470 │ });

1471 │

1472 │ modelBuilder.Entity<VQuyen>(entity =>

1473 │ {

1474 │ entity

1475 │ .HasNoKey()

1476 │ .ToView("vQuyen");

1477 │

1478 │ entity.Property(e => e.HoTen).HasMaxLength(100);

1479 │ entity.Property(e => e.TenDangNhap)

1480 │ .HasMaxLength(20)

1481 │ .IsUnicode(false);

1482 │ });

1483 │

1484 │ modelBuilder.Entity<VTonKho>(entity =>

1485 │ {

1486 │ entity

1487 │ .HasNoKey()

1488 │ .ToView("vTonKho");

1489 │

1490 │ entity.Property(e => e.GiaVon).HasColumnType("decimal(20, 4)");

1491 │ entity.Property(e => e.NamThang)

1492 │ .HasMaxLength(6)

1493 │ .IsUnicode(false);

1494 │ entity.Property(e => e.NgayNhapHang).HasColumnType("datetime");

1495 │ entity.Property(e => e.SlbanTrongKy)

1496 │ .HasColumnType("decimal(20, 4)")

1497 │ .HasColumnName("SLBanTrongKy");

1498 │ entity.Property(e => e.SlnhapTrongKy)

1499 │ .HasColumnType("decimal(20, 4)")

1500 │ .HasColumnName("SLNhapTrongKy");

1501 │ entity.Property(e => e.SltonCuoiKy)

1502 │ .HasColumnType("decimal(20, 4)")

1503 │ .HasColumnName("SLTonCuoiKy");

1504 │ entity.Property(e => e.SltonDauKy)

1505 │ .HasColumnType("decimal(20, 4)")

1506 │ .HasColumnName("SLTonDauKy");

1507 │ entity.Property(e => e.SlxuatKhac)

1508 │ .HasColumnType("decimal(20, 4)")

1509 │ .HasColumnName("SLXuatKhac");

1510 │ entity.Property(e => e.SlxuatTieuDung)

1511 │ .HasColumnType("decimal(20, 4)")

1512 │ .HasColumnName("SLXuatTieuDung");

1513 │ entity.Property(e => e.SlxuatTra)

1514 │ .HasColumnType("decimal(18, 4)")

1515 │ .HasColumnName("SLXuatTra");

1516 │ entity.Property(e => e.SoHd)

1517 │ .HasMaxLength(50)

1518 │ .HasColumnName("SoHD");

1519 │ entity.Property(e => e.TenHangHoa).HasMaxLength(200);

1520 │ entity.Property(e => e.TriGiaBan).HasColumnType("decimal(38, 6)");

1521 │ entity.Property(e => e.TriGiaCuoiKy).HasColumnType("decimal(38, 6)");

1522 │ entity.Property(e => e.TriGiaDauKy).HasColumnType("decimal(38, 6)");

1523 │ entity.Property(e => e.TriGiaNhap).HasColumnType("decimal(38, 6)");

1524 │ });

1525 │

1526 │ modelBuilder.Entity<VTonKhoBanHang>(entity =>

1527 │ {

1528 │ entity

1529 │ .HasNoKey()

1530 │ .ToView("vTonKhoBanHang");

1531 │

1532 │ entity.Property(e => e.LoGanNhat).HasColumnType("datetime");

1533 │ entity.Property(e => e.MaKt)

1534 │ .HasMaxLength(20)

1535 │ .IsUnicode(false)

1536 │ .HasColumnName("MaKT");

1537 │ entity.Property(e => e.MaVach)

1538 │ .HasMaxLength(20)

1539 │ .IsUnicode(false);

1540 │ entity.Property(e => e.NamThang)

1541 │ .HasMaxLength(6)

1542 │ .IsUnicode(false);

1543 │ entity.Property(e => e.SlbanTrongKy)

1544 │ .HasColumnType("decimal(20, 4)")

1545 │ .HasColumnName("SLBanTrongKy");

1546 │ entity.Property(e => e.SlnhapTrongKy)

1547 │ .HasColumnType("decimal(20, 4)")

1548 │ .HasColumnName("SLNhapTrongKy");

1549 │ entity.Property(e => e.SltonCuoiKy)

1550 │ .HasColumnType("decimal(20, 4)")

1551 │ .HasColumnName("SLTonCuoiKy");

1552 │ entity.Property(e => e.SltonDauKy)

1553 │ .HasColumnType("decimal(20, 4)")

1554 │ .HasColumnName("SLTonDauKy");

1555 │ entity.Property(e => e.SlxuatKhac)

1556 │ .HasColumnType("decimal(20, 4)")

1557 │ .HasColumnName("SLXuatKhac");

1558 │ entity.Property(e => e.SlxuatTieuDung)

1559 │ .HasColumnType("decimal(20, 4)")

1560 │ .HasColumnName("SLXuatTieuDung");

1561 │ entity.Property(e => e.SlxuatTra)

1562 │ .HasColumnType("decimal(18, 4)")

1563 │ .HasColumnName("SLXuatTra");

1564 │ entity.Property(e => e.SoLuongGanNhat).HasColumnType("decimal(20, 4)");

1565 │ entity.Property(e => e.TenDv)

1566 │ .HasMaxLength(100)

1567 │ .HasColumnName("TenDV");

1568 │ entity.Property(e => e.TenHangHoa).HasMaxLength(200);

1569 │ entity.Property(e => e.TenLop).HasMaxLength(50);

1570 │ });

1571 │

1572 │ modelBuilder.Entity<VTonKhoTuNgayDenNgay>(entity =>

1573 │ {

1574 │ entity

1575 │ .HasNoKey()

1576 │ .ToView("vTonKho\_TuNgay\_DenNgay");

1577 │

1578 │ entity.Property(e => e.Dvt)

1579 │ .HasMaxLength(100)

1580 │ .HasColumnName("DVT");

1581 │ entity.Property(e => e.Mavach)

1582 │ .HasMaxLength(20)

1583 │ .IsUnicode(false);

1584 │ entity.Property(e => e.NamThang)

1585 │ .HasMaxLength(6)

1586 │ .IsUnicode(false);

1587 │ entity.Property(e => e.Slban).HasColumnName("SLBan");

1588 │ entity.Property(e => e.Slnhap).HasColumnName("SLNhap");

1589 │ entity.Property(e => e.SltonCuoi).HasColumnName("SLTonCuoi");

1590 │ entity.Property(e => e.SltonDau).HasColumnName("SLTonDau");

1591 │ entity.Property(e => e.SoHd)

1592 │ .HasMaxLength(50)

1593 │ .HasColumnName("SoHD");

1594 │ entity.Property(e => e.TenHang).HasMaxLength(200);

1595 │ });

1596 │

1597 │ OnModelCreatingPartial(modelBuilder);

1598 │ }

1599 │

1600 │ partial void OnModelCreatingPartial(ModelBuilder modelBuilder);

1601 │ }

────────────────────────────────────────────────────────────

**📄 Hhimport.cs (CS, 47 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class Hhimport

7 │ {

8 │ public string? MãBệnhViện { get; set; }

9 │

10 │ public string? MãCode { get; set; }

11 │

12 │ public string? LớpHàngHoá { get; set; }

13 │

14 │ public double? MaLopHang { get; set; }

15 │

16 │ public string? TênHàngHoá { get; set; }

17 │

18 │ public string? Dvt { get; set; }

19 │

20 │ public string? Madvt { get; set; }

21 │

22 │ public double? GiáBán { get; set; }

23 │

24 │ public double? GiáVốn { get; set; }

25 │

26 │ public string? F10 { get; set; }

27 │

28 │ public string? F11 { get; set; }

29 │

30 │ public string? F12 { get; set; }

31 │

32 │ public string? F13 { get; set; }

33 │

34 │ public string? F14 { get; set; }

35 │

36 │ public string? F15 { get; set; }

37 │

38 │ public string? F16 { get; set; }

39 │

40 │ public string? F17 { get; set; }

41 │

42 │ public string? F18 { get; set; }

43 │

44 │ public string? F19 { get; set; }

45 │

46 │ public string? F20 { get; set; }

47 │ }

────────────────────────────────────────────────────────────

**📄 HinhThucTTModel.cs (CS, 21 lines)**

1 │ ﻿using Microsoft.EntityFrameworkCore;

2 │ using QuanLyBanHang.Models.QLBHEntity;

3 │

4 │ namespace QuanLyBanHang.Models.ExModel.CAT

5 │ {

6 │ public class HinhThucTTModel : Base<TblHinhThucThanhToan>

7 │ {

8 │ public HinhThucTTModel(DbContext dbContext) : base(dbContext)

9 │ {

10 │ }

11 │ public TblHinhThucThanhToan GetItem(int Ma)

12 │ {

13 │ return First(t => t.Ma == Ma && !(t.Deleted ?? false));

14 │ }

15 │

16 │ public List<TblHinhThucThanhToan> GetData()

17 │ {

18 │ return Get(t => !(t.Deleted ?? false)).OrderBy(t => t.Ma).ToList();

19 │ }

20 │ }

21 │ }

────────────────────────────────────────────────────────────

**📄 HinhThucTt.cs (CS, 19 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class HinhThucTt

7 │ {

8 │ public int HinhThucId { get; set; }

9 │

10 │ public string TenHinhThuc { get; set; } = null!;

11 │

12 │ public bool? TrangThai { get; set; }

13 │

14 │ public bool? DaXoa { get; set; }

15 │

16 │ public int? NguoiXoa { get; set; }

17 │

18 │ public DateTime? NgayXoa { get; set; }

19 │ }

────────────────────────────────────────────────────────────

**📄 HisLichSuChinhGia.cs (CS, 23 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class HisLichSuChinhGia

7 │ {

8 │ public long Ma { get; set; }

9 │

10 │ public int? MaHangHoa { get; set; }

11 │

12 │ public int? MaKho { get; set; }

13 │

14 │ public decimal? GiaCu { get; set; }

15 │

16 │ public decimal? GiaMoi { get; set; }

17 │

18 │ public string? GhiChu { get; set; }

19 │

20 │ public int? NguoiSua { get; set; }

21 │

22 │ public DateTime? NgaySua { get; set; }

23 │ }

────────────────────────────────────────────────────────────

**📄 HoaDonBan.cs (CS, 71 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class HoaDonBan

7 │ {

8 │ public int HoaDonId { get; set; }

9 │

10 │ public int? MaKho { get; set; }

11 │

12 │ public int? MaMayBanHang { get; set; }

13 │

14 │ public int? MaPhongBenh { get; set; }

15 │

16 │ public DateTime NgayHd { get; set; }

17 │

18 │ public string SoSeri { get; set; } = null!;

19 │

20 │ public bool DaXoa { get; set; }

21 │

22 │ public int NguoiBan { get; set; }

23 │

24 │ public int HinhThucTt { get; set; }

25 │

26 │ public int Trigia { get; set; }

27 │

28 │ public int? GiaGiam { get; set; }

29 │

30 │ public int TienKhachTra { get; set; }

31 │

32 │ public int? TienThua { get; set; }

33 │

34 │ public string? MaVachThe { get; set; }

35 │

36 │ public int? TienThe { get; set; }

37 │

38 │ public string? GhiChu { get; set; }

39 │

40 │ public DateTime? NgayTao { get; set; }

41 │

42 │ public int? NguoiTao { get; set; }

43 │

44 │ public DateTime? NgayXoa { get; set; }

45 │

46 │ public int? NguoiXoa { get; set; }

47 │

48 │ public bool? KhoaSo { get; set; }

49 │

50 │ public DateTime? NgaySua { get; set; }

51 │

52 │ public int? NguoiSua { get; set; }

53 │

54 │ public int? MaThanhToan { get; set; }

55 │

56 │ public DateTime? NgayThanhToan { get; set; }

57 │

58 │ public int? NguoiThanhToan { get; set; }

59 │

60 │ public int? TongTienTraLai { get; set; }

61 │

62 │ public int? TienTraLaiKhach { get; set; }

63 │

64 │ public int? TienTraLaiThe { get; set; }

65 │

66 │ public int? KhuyenMaiTienMat { get; set; }

67 │

68 │ public string? RequestId { get; set; }

69 │

70 │ public string? WebOrderId { get; set; }

71 │ }

────────────────────────────────────────────────────────────

**📄 HoaDonBanModel.cs (CS, 756 lines)**

1 │ ﻿using Microsoft.EntityFrameworkCore;

2 │ using QuanLyBanHang.Common;

3 │ using QuanLyBanHang.Models.ExModel.StoredProcedureModel;

4 │ using QuanLyBanHang.Models.QLBHEntity;

5 │ using QuanLyBanHang.Models.VCEntity;

6 │ using QuanLyBanHang.Models.VCEntity.ExModel;

7 │ using QuanLyBanHang.Services;

8 │ using static QuanLyBanHang.Controllers.OPS.BanHangController;

9 │

10 │ namespace QuanLyBanHang.Models.ExModel.OPS

11 │ {

12 │ public class HoaDonBanModel : Base<OpsHoaDon>

13 │ {

14 │ QuanLyVoucherContext vcContext;

15 │

16 │ public HoaDonBanModel(DbContext dbContext, QuanLyVoucherContext voucherDbContext) : base(dbContext)

17 │ {

18 │ vcContext = voucherDbContext;

19 │ }

20 │ QuanLyBanHangContext dbContext

21 │ {

22 │ get { return \_dbContext as QuanLyBanHangContext; }

23 │ set { return; }

24 │ }

25 │ public OpsHoaDon GetItem(int Ma)

26 │ {

27 │ return First(t => t.Ma == Ma && !(t.Deleted ?? false));

28 │ }

29 │ public List<sp\_HoaDonTheoNgayResult> GetData(DateTime TuNgay, DateTime DenNgay, Guid MaChiNhanh, int MaKho, Guid...

30 │ {

31 │

32 │ var hoaDonList = (

33 │ from hd in dbContext.OpsHoaDon.Where(x => !(x.Deleted ?? false) && x.MaChiNhanh == MaChiNhanh)

34 │ join nd in dbContext.SysUser on hd.NguoiThanhToan equals nd.Ma

35 │ join tt in dbContext.TblHinhThucThanhToan on hd.HinhThucTt equals tt.Ma

36 │ join k in dbContext.StockDanhMucKho on hd.MaKho equals k.Ma into khoGroup

37 │ from k in khoGroup.DefaultIfEmpty()

38 │ where hd.NgayHoaDon.Date >= TuNgay.Date && hd.NgayHoaDon.Date <= DenNgay.Date

39 │ && hd.MaKho == MaKho

40 │ && (NguoiThanhToan == Utility.defaultUID || hd.NguoiThanhToan == NguoiThanhToan)

41 │ select new sp\_HoaDonTheoNgayResult

42 │ {

43 │ Ma = hd.Ma,

44 │ NgayHoaDon = hd.NgayHoaDon,

45 │ SoSeri = hd.SoSeri,

46 │ NguoiBan = nd.Username,

47 │ HinhThucTT = hd.HinhThucTt,

48 │ TenHinhThuc = tt.Ten,

49 │ Trigia = hd.PhaiThu,

50 │ GiaGiam = (hd.PhaiThu \* (hd.GiamGia ?? 0)) / (100 - (hd.GiamGia ?? 0)),

51 │ NguoiThanhToan = hd.NguoiThanhToan ?? Utility.defaultUID,

52 │ KhoaSo = hd.KhoaSo,

53 │ MaKho = hd.MaKho,

54 │ TenKho = k.Ten,

55 │ TraHang = (hd.TongTienTraLai ?? 0) == 0 ? false : true,

56 │ KhuyenMaiTienMat = hd.KhuyenMaiTienMat,

57 │ RequestID = hd.RequestId,

58 │ WebOrderId = hd.WebOrderId

59 │ }).ToList();

60 │

61 │ return hoaDonList;

62 │ }

63 │

64 │ public cHoaDonBan Add(Guid MaChiNhanh, int HinhThuc\_TT, int GiamGia, int ThanhTien, int KhachDua, int HoanLai, s...

65 │ {

66 │ try

67 │ {

68 │ int mahd = 0;

69 │ int row = 0;

70 │ int errorCode = 0;

71 │ int PhuThu = 0;

72 │ int MaChiNhanhInt = dbContext.SysBranch.First(x => x.Ma == MaChiNhanh).Nid ?? 0;

73 │ int MaChiNhanhVoucher = dbContext.SysBranch.First(x => x.Ma == MaChiNhanh).MaChiNhanhVoucher ?? 0;

74 │ int UserIdInt = dbContext.SysUser.First(x => x.Ma == UserID).NId;

75 │ DateTime dtNgayTao = DateTime.Now;

76 │ DateTime dtNgayHD = new NgayKetCaModel(dbContext).LayNgayKetCa(MaKho);

77 │ int? Result = 0;

78 │ sp\_GetVoucherResult? voucher = null;

79 │ if (!MaVachThe.IsEmpty())

80 │ {

81 │ var voucherRs = new ThanhToanVoucherModel(vcContext).GetVoucher(MaVachThe ?? "", MaChiNhanhVoucher, ...

82 │ voucher = voucherRs.Voucher;

83 │ Result = voucherRs.Result;

84 │ }

85 │

86 │

87 │ using (var transaction = dbContext.Database.BeginTransaction())

88 │ {

89 │ try

90 │ {

91 │ if (HinhThuc\_TT == (int)HinhThucThanhToan.TienMat || HinhThuc\_TT == (int)HinhThucThanhToan.AR ||...

92 │ {

93 │ Tienthe = 0;

94 │ }

95 │ else

96 │ {

97 │ if ((voucher?.SoDu ?? 0) >= ThanhTien)

98 │ {

99 │ Tienthe = ThanhTien;

100 │ KhachDua = 0;

101 │ HoanLai = 0;

102 │ }

103 │ else

104 │ {

105 │ Tienthe = voucher?.SoDu ?? 0;

106 │ PhuThu = KhachDua - HoanLai;

107 │ }

108 │ }

109 │

110 │ if (HinhThuc\_TT == (int)HinhThucThanhToan.AR || HinhThuc\_TT == (int)HinhThucThanhToan.ChuyenKhoa...

111 │ {

112 │ HoanLai = 0;

113 │ }

114 │

115 │

116 │

117 │ VietQr? vietQR = null;

118 │ if (HinhThuc\_TT == (int)HinhThucThanhToan.ChuyenKhoan)

119 │ {

120 │ vietQR = dbContext.VietQr.Where(t => !(t.Deleted ?? false) && t.SoHoaDon == RequestID).Order...

121 │ if (vietQR == null)

122 │ {

123 │ return new cHoaDonBan { Result = QLBHConst.Error, ResultText = "Mã thanh toán không t...

124 │ }

125 │ else if ((vietQR.SoTien ?? 0) != ThanhTien)

126 │ {

127 │ return new cHoaDonBan { Result = QLBHConst.Error, ResultText = "Số tiền không khớ...

128 │ }

129 │ else if ((vietQR.TrangThai ?? 1) == (int)TrangThaiQR.ChoThanhToan)

130 │ {

131 │ return new cHoaDonBan { Result = QLBHConst.Error, ResultText = "Giao dịch đang chờ ...

132 │ }

133 │ else if ((vietQR.TrangThai ?? 1) == (int)TrangThaiQR.DaHuy)

134 │ {

135 │ return new cHoaDonBan { Result = QLBHConst.Error, ResultText = "Giao dịch đã bị h�...

136 │ }

137 │ else if ((vietQR.TrangThai ?? 1) == (int)TrangThaiQR.DaDung)

138 │ {

139 │ return new cHoaDonBan { Result = QLBHConst.Error, ResultText = "Đã tạo hóa đơn ch...

140 │ }

141 │ }

142 │ else

143 │ {

144 │ RequestID = "";

145 │ }

146 │ //Thêm hóa đơn

147 │ OpsHoaDon hd = new OpsHoaDon();

148 │ hd.NgayHoaDon = dtNgayHD;

149 │ Thread.Sleep(1);

150 │ hd.SoSeri = DateTime.Now.ToString("ddMMyyHHmmssfff");

151 │ hd.MaChiNhanh = MaChiNhanh;

152 │ hd.MaChiNhanhInt = MaChiNhanhInt;

153 │ hd.NguoiThanhToan = UserID;

154 │ hd.NgayThanhToan = dtNgayTao;

155 │ hd.HinhThucTt = HinhThuc\_TT;

156 │ hd.PhaiThu = ThanhTien;

157 │ hd.GiamGia = GiamGia;

158 │ hd.TienKhachTra = KhachDua;

159 │ hd.TienThua = HoanLai;

160 │ hd.GhiChu = GhiChu;

161 │ hd.MaVachThe = MaVachThe;

162 │ hd.TienThe = Tienthe;

163 │ hd.MaKho = MaKho;

164 │ hd.UserCreated = UserID;

165 │ hd.CreatedDate = dtNgayTao;

166 │ hd.KhuyenMaiTienMat = KhuyenMaiTienMat;

167 │ hd.RequestId = RequestID;

168 │ hd.WebOrderId = webOrderId;

169 │

170 │ Add(hd);

171 │ dbContext.SaveChanges();

172 │

173 │ mahd = hd.Ma;

174 │ decimal TriGiaNew = 0;

175 │ string message = "";

176 │ int res = new ChiTietHoaDonModel(dbContext).SaveListChiTiet(mahd, MaKho, cts, UserID, dtNgayTao,...

177 │ if (res != 0)

178 │ {

179 │ throw new Exception(message);

180 │

181 │ }

182 │ if ((TriGiaNew - KhuyenMaiTienMat) != hd.PhaiThu)

183 │ {

184 │ errorCode = -5;

185 │ throw new Exception("");

186 │ }

187 │ if (voucher != null)

188 │ {

189 │ int? MaThanhToan = 0;

190 │ MaThanhToan = ThemThanhToan(dtNgayTao.ToString("ddMMyyHHmmssfff"), MaChiNhanhVoucher, MaNgan...

191 │ if ((MaThanhToan ?? 0) <= 0)

192 │ {

193 │ errorCode = -6;

194 │ throw new Exception("");

195 │ }

196 │ }

197 │ if (HinhThuc\_TT == (int)HinhThucThanhToan.ChuyenKhoan && vietQR != null)

198 │ {

199 │ vietQR.TrangThai = (int)TrangThaiQR.DaDung;

200 │ vietQR.ModifiedDate = DateTime.Now;

201 │ vietQR.UserModified = UserID;

202 │ dbContext.SaveChanges();

203 │ }

204 │

205 │ transaction.Commit();

206 │ }

207 │ catch (Exception ex)

208 │ {

209 │ transaction.Rollback();

210 │ switch (errorCode)

211 │ {

212 │ case -1:

213 │ return new cHoaDonBan { Result = QLBHConst.Error, ResultText = "Thêm hóa đơn thất ...

214 │ case -2:

215 │ return new cHoaDonBan { Result = QLBHConst.Error, ResultText = "Thêm hóa đơn thất ...

216 │ case -3:

217 │ return new cHoaDonBan { Result = QLBHConst.Error, ResultText = "Hàng hóa dòng " + row...

218 │ case -4:

219 │ return new cHoaDonBan { Result = QLBHConst.Error, ResultText = "Hàng hóa dòng " + row...

220 │ case -5:

221 │ return new cHoaDonBan { Result = QLBHConst.Error, ResultText = "Thêm hóa đơn thất ...

222 │ case -6:

223 │ return new cHoaDonBan { Result = QLBHConst.Error, ResultText = "Thêm hóa đơn thất ...

224 │ case -999://Từ SaveListChiTiet

225 │ return new cHoaDonBan { Result = QLBHConst.Error, ResultText = ex.Message };

226 │ }

227 │ return new cHoaDonBan { Result = QLBHConst.Error, ResultText = "Thêm hóa đơn thất bại. K...

228 │ }

229 │ }

230 │ TonKhoBackgroundWorker.AddJob(new TonKhoJob

231 │ {

232 │ NamThang = string.Format("{0:yyyyMM}", dtNgayHD),

233 │ MaKho = Cast.ToInt(MaKho),

234 │ UserId = UserID,

235 │ MaChiNhanh = MaChiNhanh

236 │ });

237 │

238 │ return new cHoaDonBan { Result = QLBHConst.OK, ResultText = mahd.ToString() };

239 │ }

240 │ catch (Exception ex)

241 │ {

242 │ return new cHoaDonBan { Result = QLBHConst.Error, ResultText = "Có lỗi xảy ra trong quá trình th�...

243 │ }

244 │

245 │ }

246 │

247 │ public cHoaDonBan Edit(int Ma, Guid MaChiNhanh, int HinhThuc\_TT, int GiamGia, int ThanhTien, int KhachDua, int H...

248 │ {

249 │ try

250 │ {

251 │ int MaChiNhanhInt = dbContext.SysBranch.First(x => x.Ma == MaChiNhanh).Nid ?? 0;

252 │ int MaChiNhanhVoucher = dbContext.SysBranch.First(x => x.Ma == MaChiNhanh).MaChiNhanhVoucher ?? 0;

253 │ int UserIdInt = dbContext.SysUser.First(x => x.Ma == UserID).NId;

254 │ if (KiemTraKhoaSo(Ma))

255 │ {

256 │ return new cHoaDonBan { Result = QLBHConst.Error, ResultText = "Hóa đơn đã khóa sổ không th...

257 │ }

258 │ OpsHoaDon hd = dbContext.OpsHoaDon.First(t => t.Ma == Ma);

259 │ if (hd == null)

260 │ {

261 │ return new cHoaDonBan { Result = QLBHConst.Error, ResultText = "Hóa đơn không tồn tại!" };

262 │ }

263 │ int row = 0;

264 │ int errorCode = 0;

265 │ sp\_GetVoucherResult voucher = null;

266 │ int PhuThu = 0;

267 │

268 │ if (!MaVachThe.IsEmpty())

269 │ {

270 │ int? Result = 0;

271 │ var voucherRs = new ThanhToanVoucherModel(vcContext).GetVoucher(MaVachThe ?? "", MaChiNhanhVoucher, ...

272 │ voucher = voucherRs.Voucher;

273 │ Result = voucherRs.Result;

274 │ }

275 │ using (var transaction = dbContext.Database.BeginTransaction())

276 │ {

277 │

278 │ DateTime dtNgayTao = DateTime.Now;

279 │ try

280 │ {

281 │

282 │ int TienThe\_Old = hd.TienThe ?? 0;

283 │ if (HinhThuc\_TT == (int)HinhThucThanhToan.The)

284 │ {

285 │ if ((voucher.SoDu + (hd.TienThe ?? 0)) >= ThanhTien)

286 │ {

287 │ Tienthe = ThanhTien;

288 │ KhachDua = 0;

289 │ HoanLai = 0;

290 │ }

291 │ else

292 │ {

293 │ Tienthe = ((voucher.SoDu ?? 0) + (hd.TienThe ?? 0));

294 │ PhuThu = KhachDua - HoanLai;

295 │ }

296 │ }

297 │

298 │ //Thêm hóa đơn

299 │

300 │ //if(hd == null)

301 │ //{

302 │ // return new cHoaDonBan { Result = QLBHConst.Error, ResultText = QLBHConst.NotExist };

303 │ //}

304 │

305 │ //hd.Ngay\_HD = dtNgayTao;

306 │ hd.Deleted = false;

307 │ hd.NguoiThanhToan = UserID;

308 │ hd.HinhThucTt = HinhThuc\_TT;

309 │ hd.PhaiThu = ThanhTien;

310 │ hd.GiamGia = GiamGia;

311 │ hd.TienKhachTra = KhachDua;

312 │ hd.TienThua = HoanLai;

313 │ hd.GhiChu = GhiChu;

314 │ hd.MaVachThe = MaVachThe;

315 │ hd.TienThe = Tienthe;

316 │ hd.MaKho = MaKho;

317 │ hd.KhuyenMaiTienMat = KhuyenMaiTienMat;

318 │

319 │ dbContext.SaveChanges();

320 │ decimal TriGiaNew = 0;

321 │ string message = "";

322 │ int res = new ChiTietHoaDonModel(dbContext).SaveListChiTiet(hd.Ma, MaKho, cts, UserID, dtNgayTao...

323 │ if (res != 0)

324 │ {

325 │ //if (rowError <= -1000)

326 │ //{

327 │ // errorCode = -99999;

328 │ // row = rowError / -1000;

329 │ // throw new Exception("Số lượng tồn kho không đủ. Dòng " + row.ToString());

330 │ //}

331 │ //else

332 │ //{

333 │ // row = rowError;

334 │ throw new Exception(message);

335 │ //}

336 │ }

337 │ if ((TriGiaNew - KhuyenMaiTienMat) != hd.PhaiThu)

338 │ {

339 │ errorCode = -5;

340 │ throw new Exception("");

341 │ }

342 │

343 │ if ((hd.MaThanhToan ?? 0) > 0 && voucher != null)

344 │ {

345 │ int? Result = 0;

346 │ Result = SuaThanhToan(hd.MaThanhToan ?? 0, voucher.Ma ?? 0, hd.PhaiThu.ToInt(), Tienthe - Ti...

347 │ if ((Result ?? 0) <= 0)

348 │ {

349 │ errorCode = -6;

350 │ throw new Exception("");

351 │ }

352 │ }

353 │ transaction.Commit();

354 │ }

355 │ catch (Exception ex)

356 │ {

357 │ transaction.Rollback();

358 │ switch (errorCode)

359 │ {

360 │ case -1:

361 │ return new cHoaDonBan { Result = QLBHConst.Error, ResultText = "Sửa hóa đơn thất ...

362 │ case -2:

363 │ return new cHoaDonBan { Result = QLBHConst.Error, ResultText = "Sửa hóa đơn thất ...

364 │ case -3:

365 │ return new cHoaDonBan { Result = QLBHConst.Error, ResultText = "Hàng hóa dòng " + row...

366 │ case -4:

367 │ return new cHoaDonBan { Result = QLBHConst.Error, ResultText = "Hàng hóa dòng " + row...

368 │ case -5:

369 │ return new cHoaDonBan { Result = QLBHConst.Error, ResultText = "Sửa hóa đơn thất ...

370 │ case -6:

371 │ return new cHoaDonBan { Result = QLBHConst.Error, ResultText = "Sửa hóa đơn thất ...

372 │ case -999://Từ SaveListChiTiet

373 │ return new cHoaDonBan { Result = QLBHConst.Error, ResultText = ex.Message };

374 │ }

375 │ return new cHoaDonBan { Result = QLBHConst.Error, ResultText = "Sửa hóa đơn thất bại. K...

376 │ }

377 │ }

378 │ new TonKhoModel(dbContext).TinhTonKhoBanHang(Cast.ToInt(MaKho), string.Format("{0:yyyyMM}", hd.NgayHoaDo...

379 │ TonKhoBackgroundWorker.AddJob(new TonKhoJob

380 │ {

381 │ NamThang = string.Format("{0:yyyyMM}", hd.NgayHoaDon),

382 │ MaKho = Cast.ToInt(MaKho),

383 │ UserId = UserID,

384 │ MaChiNhanh = MaChiNhanh

385 │ });

386 │ return new cHoaDonBan { Result = QLBHConst.OK, ResultText = QLBHConst.UpdateSuccessfully };

387 │ }

388 │ catch (Exception ex)

389 │ {

390 │ return new cHoaDonBan { Result = QLBHConst.Error, ResultText = "Có lỗi xảy ra trong quá trình s��...

391 │ }

392 │

393 │ }

394 │ public int Delete(int Ma, Guid UserID)

395 │ {

396 │ int UserIdInt = dbContext.SysUser.First(x => x.Ma == UserID).NId;

397 │

398 │ using (var dbContextTransaction = dbContext.Database.BeginTransaction())

399 │ {

400 │

401 │ DateTime dtNgayTao = DateTime.Now;

402 │ int? result = 0;

403 │ try

404 │ {

405 │ OpsHoaDon? drItem = dbContext.OpsHoaDon.FirstOrDefault(t => t.Ma == Ma);

406 │ if (drItem == null)

407 │ {

408 │ return -2;

409 │ }

410 │ if ((drItem.KhoaSo ?? false))

411 │ {

412 │ return -3;

413 │ }

414 │ drItem.Deleted = true;

415 │ drItem.DeletedDate = dtNgayTao;

416 │ drItem.UserDeleted = UserID;

417 │ dbContext.SaveChanges();

418 │ List<OpsChiTietHoaDon> cts = dbContext.OpsChiTietHoaDon.Where(t => t.MaHoaDon == drItem.Ma).ToList()...

419 │ result = -4;

420 │ foreach (var ct in cts)

421 │ {

422 │ ct.Deleted = true;

423 │ ct.DeletedDate = dtNgayTao;

424 │ ct.UserDeleted = UserID;

425 │ dbContext.SaveChanges();

426 │ }

427 │ if ((drItem.MaThanhToan ?? 0) > 0)

428 │ {

429 │ result = XoaThanhToan(drItem.MaThanhToan ?? 0, UserIdInt, dtNgayTao);

430 │ if ((result ?? 0) <= 0)

431 │ {

432 │ result = -5;

433 │ throw new Exception("");

434 │ }

435 │ }

436 │ dbContextTransaction.Commit();

437 │ return 1;

438 │ }

439 │ catch

440 │ {

441 │ dbContextTransaction.Rollback();

442 │ return result ?? 0;

443 │ }

444 │ }

445 │ }

446 │ public bool KiemTraKhoaSo(int HoaDon\_ID)

447 │ {

448 │ OpsHoaDon hd = GetItem(HoaDon\_ID);

449 │ if (hd == null)

450 │ return false;

451 │ return hd.KhoaSo ?? false;

452 │ }

453 │ public int KhoaSo(int Ma, Guid UserID, bool Value)

454 │ {

455 │ OpsHoaDon drItem = GetItem(Ma);

456 │

457 │ if (drItem != null)

458 │ {

459 │ if ((drItem.KhoaSo ?? false) == Value)

460 │ {

461 │ return -3;

462 │ }

463 │ DateTime dtNgayTao = DateTime.Now;

464 │ drItem.KhoaSo = Value;

465 │ drItem.ModifiedDate = dtNgayTao;

466 │ drItem.UserModified = UserID;

467 │

468 │ Update(drItem);

469 │ dbContext.SaveChanges();

470 │ return 1;

471 │ }

472 │

473 │ return -2;

474 │ }

475 │ public cHoaDonBan TraHang(int Ma, int TienTraLaiKhach, int TienTraLaiThe, string GhiChu, List<cCT\_HoaDon> cts, G...

476 │ {

477 │ try

478 │ {

479 │ if (KiemTraKhoaSo(Ma))

480 │ {

481 │ return new cHoaDonBan { Result = QLBHConst.Error, ResultText = "Hóa đơn đã khóa sổ không th...

482 │ }

483 │ int row = 0;

484 │ int errorCode = 0;

485 │ sp\_GetVoucherResult voucher = null;

486 │ int MaChiNhanhInt = dbContext.SysBranch.First(x => x.Ma == MaChiNhanh).Nid ?? 0;

487 │ int MaChiNhanhVoucher = dbContext.SysBranch.First(x => x.Ma == MaChiNhanh).MaChiNhanhVoucher ?? 0;

488 │ int UserIdInt = dbContext.SysUser.First(x => x.Ma == UserID).NId;

489 │ OpsHoaDon? hd = dbContext.OpsHoaDon.FirstOrDefault(t => t.Ma == Ma);

490 │ using (var dbContextTransaction = dbContext.Database.BeginTransaction())

491 │ {

492 │

493 │ DateTime dtNgayTao = DateTime.Now;

494 │ try

495 │ {

496 │ if (hd == null)

497 │ {

498 │ return new cHoaDonBan { Result = QLBHConst.Error, ResultText = QLBHConst.NotExist };

499 │ }

500 │ if (!hd.MaVachThe.IsEmpty())

501 │ {

502 │ int? Result = 0;

503 │ var voucherRs = new ThanhToanVoucherModel(vcContext).GetVoucher(hd.MaVachThe ?? "", MaChiNha...

504 │ voucher = voucherRs.Voucher;

505 │ Result = voucherRs.Result;

506 │ }

507 │

508 │ int TongTienTraLai = 0;

509 │ string message = "";

510 │ int rowError = new ChiTietHoaDonModel(dbContext).SaveTraHang(hd.Ma, cts, UserID, MaChiNhanh, dtN...

511 │ if (errorCode != 0)

512 │ {

513 │ if (rowError <= -1000)

514 │ {

515 │ errorCode = -99999;

516 │ row = rowError / -1000;

517 │ throw new Exception("Hàng hóa không tồn tại. Dòng " + row.ToString());

518 │ }

519 │ else

520 │ {

521 │ row = rowError;

522 │ throw new Exception(message);

523 │ }

524 │ }

525 │

526 │ if ((hd.MaThanhToan ?? 0) > 0 && (hd.TienThe ?? 0) > 0 && voucher != null)

527 │ {

528 │ if (TongTienTraLai > (hd.TienThe ?? 0))

529 │ {

530 │ TienTraLaiThe = (hd.TienThe ?? 0);

531 │ TienTraLaiKhach = TongTienTraLai - (hd.TienThe ?? 0);

532 │

533 │ }

534 │ else

535 │ {

536 │ TienTraLaiThe = TongTienTraLai;

537 │ TienTraLaiKhach = 0;

538 │ }

539 │ int? Result = 0;

540 │ Result = TraHangThanhToan(hd.MaThanhToan ?? 0, voucher.Ma ?? 0, TienTraLaiThe, UserIdInt, dt...

541 │ if ((Result ?? 0) <= 0)

542 │ {

543 │ errorCode = -6;

544 │ throw new Exception("");

545 │ }

546 │ }

547 │ else

548 │ {

549 │ TienTraLaiKhach = TongTienTraLai;

550 │ TienTraLaiThe = 0;

551 │ }

552 │ hd.GhiChu = GhiChu;

553 │ hd.TienTraLaiThe = TienTraLaiThe;

554 │ hd.TienTraLaiKhach = TienTraLaiKhach;

555 │ hd.TongTienTraLai = TienTraLaiKhach + TienTraLaiThe;

556 │ dbContext.SaveChanges();

557 │ dbContextTransaction.Commit();

558 │ }

559 │ catch (Exception ex)

560 │ {

561 │ dbContextTransaction.Rollback();

562 │ switch (errorCode)

563 │ {

564 │ case -1:

565 │ return new cHoaDonBan { Result = QLBHConst.Error, ResultText = "Trả hàng thất bại...

566 │ case -2:

567 │ return new cHoaDonBan { Result = QLBHConst.Error, ResultText = "Trả hàng thất bại...

568 │ case -3:

569 │ return new cHoaDonBan { Result = QLBHConst.Error, ResultText = "Hàng hóa dòng " + row...

570 │ case -4:

571 │ return new cHoaDonBan { Result = QLBHConst.Error, ResultText = "Hàng hóa dòng " + row...

572 │ case -5:

573 │ return new cHoaDonBan { Result = QLBHConst.Error, ResultText = "Trả hàng thất bại...

574 │ case -6:

575 │ return new cHoaDonBan { Result = QLBHConst.Error, ResultText = "Trả hàng thất bại...

576 │ }

577 │ return new cHoaDonBan { Result = QLBHConst.Error, ResultText = "Trả hàng thất bại. Kiểm...

578 │ }

579 │ }

580 │ TonKhoBackgroundWorker.AddJob(new TonKhoJob

581 │ {

582 │ NamThang = string.Format("{0:yyyyMM}", DateTime.Now),

583 │ MaKho = hd.MaKho ?? 0,

584 │ UserId = UserID,

585 │ MaChiNhanh = MaChiNhanh

586 │ });

587 │ return new cHoaDonBan { Result = QLBHConst.OK, ResultText = QLBHConst.UpdateSuccessfully };

588 │ }

589 │ catch (Exception ex)

590 │ {

591 │ return new cHoaDonBan { Result = QLBHConst.Error, ResultText = "Có lỗi xảy ra trong quá trình tr�...

592 │ }

593 │ }

594 │

595 │ public int ThemThanhToan(

596 │ string code, int maChiNhanh, int maNganh, int maBanHang, int giaTriThanhToan,

597 │ int tienThe, int phuThu, string ghiChu, int nguoiTao, DateTime ngayTao, int hoaDon\_Id)

598 │ {

599 │ var banHang = vcContext.TblBanHang.FirstOrDefault(bh => bh.Ma == maBanHang);

600 │ if (banHang == null) return -1;

601 │

602 │ int soDu = banHang.SoDu;

603 │ int conLai = soDu - tienThe;

604 │ if (conLai < 0) return -1;

605 │

606 │ var thanhToan = new TblThanhToan

607 │ {

608 │ Code = code,

609 │ MaChiNhanh = maChiNhanh,

610 │ MaNganh = maNganh,

611 │ MaBanHang = maBanHang,

612 │ GiaTriThanhToan = tienThe,

613 │ SoDu = conLai,

614 │ PhuThu = 0,

615 │ GhiChu = ghiChu,

616 │ NguoiTao = nguoiTao,

617 │ NgayTao = ngayTao

618 │ };

619 │

620 │ vcContext.TblThanhToan.Add(thanhToan);

621 │ vcContext.SaveChanges();

622 │

623 │ int maThanhToan = thanhToan.Ma;

624 │ if (maThanhToan > 0)

625 │ {

626 │ banHang.SoDu = conLai;

627 │ banHang.NguoiSua = nguoiTao;

628 │ banHang.NgaySua = ngayTao;

629 │

630 │ var hoaDon = dbContext.OpsHoaDon.FirstOrDefault(h => h.Ma == hoaDon\_Id);

631 │ if (hoaDon != null)

632 │ {

633 │ hoaDon.MaThanhToan = maThanhToan;

634 │ hoaDon.NgayThanhToan = ngayTao;

635 │ }

636 │

637 │ dbContext.SaveChanges();

638 │ vcContext.SaveChanges();

639 │ return maThanhToan;

640 │ }

641 │

642 │ return -1;

643 │ }

644 │

645 │ public int SuaThanhToan(

646 │ int ma,

647 │ int maBanHang,

648 │ int giaTriThanhToan,

649 │ int tienThe,

650 │ int phuThu,

651 │ string ghiChu,

652 │ int nguoiSua,

653 │ DateTime ngaySua)

654 │ {

655 │ try

656 │ {

657 │ var banHang = vcContext.TblBanHang.FirstOrDefault(bh => bh.Ma == maBanHang);

658 │ if (banHang == null) return 0;

659 │

660 │ int soDuHienTai = banHang.SoDu;

661 │ int conLai = soDuHienTai - tienThe;

662 │

663 │ if (conLai < 0) return -1; // Số dư không đủ

664 │

665 │ var thanhToanCanSua = vcContext.TblThanhToan.FirstOrDefault(tt => tt.Ma == ma);

666 │ if (thanhToanCanSua == null) return 0;

667 │

668 │ thanhToanCanSua.GiaTriThanhToan = tienThe;

669 │ thanhToanCanSua.SoDu = conLai;

670 │ thanhToanCanSua.GhiChu = ghiChu;

671 │ thanhToanCanSua.NguoiSua = nguoiSua;

672 │ thanhToanCanSua.NgaySua = ngaySua;

673 │

674 │ banHang.SoDu = conLai;

675 │ banHang.NguoiSua = nguoiSua;

676 │ banHang.NgaySua = ngaySua;

677 │

678 │ vcContext.SaveChanges();

679 │

680 │ return 1;

681 │ }

682 │ catch

683 │ {

684 │ return 0;

685 │ }

686 │ }

687 │

688 │ public int TraHangThanhToan(

689 │ int maThanhToan,

690 │ int maBanHang,

691 │ int tienTraLai,

692 │ int nguoiSua,

693 │ DateTime ngaySua)

694 │ {

695 │ try

696 │ {

697 │ var thanhToan = vcContext.TblThanhToan.FirstOrDefault(tt => tt.Ma == maThanhToan);

698 │ if (thanhToan == null) return 0;

699 │

700 │ thanhToan.GiaTriThanhToan -= tienTraLai;

701 │ thanhToan.SoDu += tienTraLai;

702 │ thanhToan.NguoiSua = nguoiSua;

703 │ thanhToan.NgaySua = ngaySua;

704 │

705 │ var banHang = vcContext.TblBanHang.FirstOrDefault(bh => bh.Ma == maBanHang);

706 │ if (banHang == null) return 0;

707 │

708 │ banHang.SoDu += tienTraLai;

709 │ banHang.NguoiSua = nguoiSua;

710 │ banHang.NgaySua = ngaySua;

711 │

712 │ vcContext.SaveChanges();

713 │ return 1;

714 │ }

715 │ catch

716 │ {

717 │ return 0;

718 │ }

719 │ }

720 │

721 │ public int XoaThanhToan(

722 │ int ma,

723 │ int nguoiXoa,

724 │ DateTime ngayXoa)

725 │ {

726 │ try

727 │ {

728 │ var thanhToan = vcContext.TblThanhToan.FirstOrDefault(tt => tt.Ma == ma);

729 │ if (thanhToan == null) return 0;

730 │

731 │ int maBanHang = thanhToan.MaBanHang;

732 │ int giaTriThanhToan = thanhToan.GiaTriThanhToan;

733 │

734 │ thanhToan.DaXoa = true;

735 │ thanhToan.NguoiXoa = nguoiXoa;

736 │ thanhToan.NgayXoa = ngayXoa;

737 │

738 │ var banHang = vcContext.TblBanHang.FirstOrDefault(bh => bh.Ma == maBanHang);

739 │ if (banHang == null) return 0;

740 │

741 │ banHang.SoDu += giaTriThanhToan;

742 │ banHang.NguoiSua = nguoiXoa;

743 │ banHang.NgaySua = ngayXoa;

744 │

745 │ vcContext.SaveChanges();

746 │

747 │ return 1;

748 │ }

749 │ catch

750 │ {

751 │ return 0;

752 │ }

753 │ }

754 │

755 │ }

756 │ }

────────────────────────────────────────────────────────────

**📄 HomeController.cs (CS, 482 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using QuanLyBanHang.Common;

3 │ using QuanLyBanHang.Common.Attributes;

4 │ using QuanLyBanHang.Models.APIModels;

5 │ using QuanLyBanHang.Models.HappyStoreEntity;

6 │ using QuanLyBanHang.Models.QLBHEntity;

7 │

8 │ namespace QuanLyBanHang.Controllers.SYS

9 │ {

10 │ public class HomeController : BaseController

11 │ {

12 │ private HappyStoreContext storeContext { get; set; }

13 │ public HomeController(QuanLyBanHangContext dbcontext, HappyStoreContext \_storecontext, IWebHostEnvironment hosti...

14 │ {

15 │ QLBHConst.wwwRoot = hostingEnvironment.WebRootPath;

16 │ m\_MenuID = Utility.defaultUID;

17 │ storeContext = \_storecontext;

18 │ }

19 │

20 │

21 │ [TokenUserCheckHTTP()]

22 │ public IActionResult Index()

23 │ {

24 │ //List<SysMenu> ListMenu = new sys\_MenuModel(dbcontext).GetData();

25 │ //List<SysMenu> ListMenuCon = new List<SysMenu>();

26 │ //ListMenuCon = ListMenu.Where(t => (t.MaPhongKham ?? 0) == 0 && (t.ParentId != (int)MenuID.pkPhongKham || t...

27 │ //SysUser CurrentUser = new sys\_UserModel(dbcontext).GetItem(objToken!.userid);

28 │ //List<SysMenu> ListMenu = new sys\_MenuModel(dbcontext).GetData();

29 │ //List<string> ListPK = (CurrentUser!.PhongKhamDieuTri + "").Split(';').ToList();

30 │ //ListMenu = ListMenu.Where(t => (t.MaPhongKham ?? 0) != 0 && t.ParentId == (int)MenuID.pkPhongKham && ListP...

31 │ if (isAPI)

32 │ {

33 │ return new JsonResult(new ResponseModelSuccess(""));

34 │ }

35 │ return View();

36 │ }

37 │ [HttpPost]

38 │ [TokenUserCheckHTTP(Permission.NotCheck)]

39 │ public IActionResult getTimeNow()

40 │ {

41 │ return Json(new ResponseModelSuccess("", string.Format("{0:yyyy-MM-ddTHH:mm:ss}", DateTime.Now)));

42 │ }

43 │

44 │ [HttpPost]

45 │ [TokenUserCheckHTTP]

46 │ public IActionResult syncDataDanhMuc()

47 │ {

48 │ //// Sử dụng .AsNoTracking() khi đọc dữ liệu từ context cũ nếu không có ý định cập n...

49 │ //var oldNCC = storeContext.NhaCc.AsNoTracking().Where(x => !(x.DaXoa ?? false)).ToList();

50 │ //var oldDVT = storeContext.DonViTinh.AsNoTracking().Where(x => !(x.DaXoa ?? false)).ToList();

51 │ //var oldLopHang = storeContext.LopHang.AsNoTracking().Where(x => !(x.DaXoa ?? false)).ToList();

52 │ //var oldHanghoa = storeContext.HangHoa.AsNoTracking().Where(x => !(x.DaXoa ?? false)).ToList();

53 │ //var oldKho = storeContext.TblDanhMucKho.AsNoTracking().Where(x => !(x.DaXoa ?? false)).ToList();

54 │ //var oldGiaban = storeContext.TblGiaBanTheoKho.AsNoTracking().Where(x => !(x.DaXoa ?? false)).ToList();

55 │ //var oldPhong = storeContext.TblPhongBenh.AsNoTracking().Where(x => !(x.DaXoa ?? false)).ToList();

56 │

57 │ //var oldUser = storeContext.NguoiDung.AsNoTracking().Where(x => !(x.DaXoa ?? false)).ToList();

58 │ //var presentUser = dbcontext.SysUser.AsNoTracking().Where(x => !(x.Deleted ?? false)).ToList();

59 │

60 │ ////var oldNhaphang = storeContext.NhapHang.AsNoTracking().Where(x => !(x.DaXoa ?? false) && x.NgayHoaDon.Va...

61 │ ////var oldChiTietNhapHang = storeContext.CtNhapHang.AsNoTracking().Where(x => !(x.DaXoa ?? false)).ToList()...

62 │ ////var oldXuatKho = storeContext.TblXuatKho.AsNoTracking().Where(x => !(x.DaXoa ?? false) && x.NgayTao.Valu...

63 │ ////var oldChiTietXuatKho = storeContext.TblChiTietXuatKho.AsNoTracking().Where(x => !(x.DaXoa ?? false)).To...

64 │ ////var oldHoaDon = storeContext.HoaDonBan.AsNoTracking().Where(x => !x.DaXoa && x.NgayHd.Month == 4).ToList...

65 │ ////var oldChiTietHoaDon = storeContext.CtHoaDon.AsNoTracking().Where(x => !(x.DaXoa ?? false)).ToList();

66 │

67 │ //// Các phép chiếu (Select) để tạo đối tượng mới

68 │

69 │ //var newNCCQuery = oldNCC.Select(t => new TblNhaCc

70 │ //{

71 │ // MaChiNhanh = objToken.branch,

72 │ // MaKt = t.MaKt,

73 │ // TenNhaCc = t.TenNhaCc,

74 │ // DiaChi = t.DiaChi,

75 │ // DienThoai = t.DienThoai,

76 │ // Mst = t.Mst,

77 │ // Email = t.Email,

78 │ // NguoiLienHe = t.NguoiLienHe,

79 │ // Cccd = t.Cccd,

80 │ // Status = t.TrangThai, // Giả sử TrangThai tương ứng với Status mới

81 │ // // DaXoa = false, // Nếu có trường này, thường là false cho bản ghi...

82 │ //});

83 │ //var newDVTQuery = oldDVT.Select(t => new TblDonViTinh

84 │ //{

85 │ // Ten = t.Ten,

86 │ // Status = true,

87 │ // // DaXoa = false,

88 │ //});

89 │ //var newLopHangQuery = oldLopHang.Select(t => new TblLopHangHoa

90 │ //{

91 │ // MaChiNhanh = objToken.branch,

92 │ // Ten = t.TenLop,

93 │ // ChoPhepBanAm = t.ChoPhepBanAm,

94 │ // ThoiGianBatDau = t.ThoiGianBatDau,

95 │ // ThoiGianKetThuc = t.ThoiGianKetThuc,

96 │ // QuanLyQuaNfc = null, // Xác minh

97 │ // Status = true,

98 │ // MaLoaiTieuDung = 0, // Xác minh

99 │ // // DaXoa = false,

100 │ //});

101 │ //var newKhoQuery = oldKho.Select(t => new StockDanhMucKho

102 │ //{

103 │ // MaChiNhanh = objToken.branch,

104 │ // Code = t.Code,

105 │ // Ten = t.Ten,

106 │ // DienGiai = t.DienGiai,

107 │ // MayBanHangTuDong = t.MayBanHangTuDong,

108 │ // BanHang = t.BanHang,

109 │ // NhapHang = t.NhapHang,

110 │ // ThoiGianBatDau = t.ThoiGianBatDau,

111 │ // ThoiGianKetThuc = t.ThoiGianKetThuc,

112 │ // TaiKhoanCo = t.TaiKhoanCo,

113 │ // TaiKhoanKho = t.TaiKhoanKho,

114 │ // MaNganhVoucher = null, // Xác minh

115 │ // HienThiTrongWebOrder = null, // Xác minh

116 │ // TenHienThi = null, // Xác minh

117 │ // Status = true,

118 │ // // DaXoa = false,

119 │ //});

120 │ //var newPhongQuery = oldPhong.Select(t => new Models.QLBHEntity.TblPhongBenh

121 │ //{

122 │ // MaChiNhanh = objToken.branch,

123 │ // Ten = t.Ten,

124 │ // Status = true,

125 │ // NoiBo = t.NoiBo,

126 │ // MaCu = t.Ma,

127 │ // Code = t.Code

128 │ // // DaXoa = false,

129 │ //});

130 │ //using (var transaction = dbcontext.Database.BeginTransaction())

131 │ //{

132 │ // try

133 │ // {

134 │ // //var deleteNCC = dbcontext.TblNhaCc.Where(x => x.MaChiNhanh == objToken.branch);

135 │ // //var deleteDVT = dbcontext.TblDonViTinh;

136 │ // //var deleteLopHang = dbcontext.TblLopHangHoa.Where(x => x.MaChiNhanh == objToken.branch);

137 │ // //var deleteHangHoa = dbcontext.TblHangHoa.Where(x => x.MaChiNhanh == objToken.branch);

138 │ // //var deleteKho = dbcontext.StockDanhMucKho.Where(x => x.MaChiNhanh == objToken.branch);

139 │ // //var deleteGiaban = dbcontext.TblGiaBanTheoKho.Where(x => x.MaChiNhanh == objToken.branch);

140 │ // //var deletePhong = dbcontext.TblPhongBenh.Where(x => x.MaChiNhanh == objToken.branch);

141 │

142 │ // //var deleteNhaphang = dbcontext.StockNhapHang.Where(x => x.MaChiNhanh == objToken.branch);

143 │ // //var deleteChiTietNhapHang = dbcontext.StockChiTietNhapHang.Where(x => deleteNhaphang.Select(t =>...

144 │ // //var deleteXuatKho = dbcontext.StockNhapHang.Where(x => x.MaChiNhanh == objToken.branch);

145 │ // //var deleteChiTietXuatKho = dbcontext.StockChiTietNhapHang.Where(x => deleteXuatKho.Select(t => t...

146 │ // //var deleteHoaDon = dbcontext.OpsHoaDon.Where(x => x.MaChiNhanh == objToken.branch);

147 │ // //var deleteChiTietHoaDon = dbcontext.OpsChiTietHoaDon.Where(x => deleteHoaDon.Select(t => t.Ma).C...

148 │

149 │ // //dbcontext.RemoveRange(deleteNCC);

150 │ // //dbcontext.RemoveRange(deleteDVT);

151 │ // //dbcontext.RemoveRange(deleteLopHang);

152 │ // //dbcontext.RemoveRange(deleteHangHoa);

153 │ // //dbcontext.RemoveRange(deleteKho);

154 │ // //dbcontext.RemoveRange(deleteGiaban);

155 │ // dbcontext.RemoveRange(deletePhong);

156 │

157 │ // //dbcontext.RemoveRange(deleteNhaphang);

158 │ // //dbcontext.RemoveRange(deleteChiTietNhapHang);

159 │ // //dbcontext.RemoveRange(deleteXuatKho);

160 │ // //dbcontext.RemoveRange(deleteChiTietXuatKho);

161 │ // //dbcontext.RemoveRange(deleteHoaDon);

162 │ // //dbcontext.RemoveRange(deleteChiTietHoaDon);

163 │

164 │ // dbcontext.SaveChanges();

165 │

166 │ // //// 1. Nhà Cung Cấp (NCC)

167 │ // //var newNCCList = newNCCQuery.ToList();

168 │ // //dbcontext.TblNhaCc.AddRange(newNCCList);

169 │ // //dbcontext.SaveChanges();

170 │ // //var dictNCC = oldNCC

171 │ // // .Zip(newNCCList, (oldItem, newItem) => new { OldId = oldItem.Ma, NewId = newItem.Ma })

172 │ // // .ToDictionary(x => x.OldId, x => x.NewId);

173 │

174 │ // //// 2. Đơn Vị Tính (DVT)

175 │ // //var newDVTList = newDVTQuery.ToList();

176 │ // //dbcontext.TblDonViTinh.AddRange(newDVTList);

177 │ // //dbcontext.SaveChanges();

178 │ // //var dictDVT = oldDVT

179 │ // // .Zip(newDVTList, (oldItem, newItem) => new { OldId = oldItem.Ma, NewId = newItem.Ma })

180 │ // // .ToDictionary(x => x.OldId, x => x.NewId);

181 │

182 │ // //// 3. Lớp Hàng

183 │ // //var newLopHangList = newLopHangQuery.ToList();

184 │ // //dbcontext.TblLopHangHoa.AddRange(newLopHangList);

185 │ // //dbcontext.SaveChanges();

186 │ // //var dictLopHang = oldLopHang

187 │ // // .Zip(newLopHangList, (oldItem, newItem) => new { OldId = oldItem.LopId, NewId = newItem.Ma }...

188 │ // // .ToDictionary(x => x.OldId, x => x.NewId);

189 │

190 │ // //// 4. Kho

191 │ // //var newKhoList = newKhoQuery.ToList();

192 │ // //dbcontext.StockDanhMucKho.AddRange(newKhoList);

193 │ // //dbcontext.SaveChanges();

194 │ // //var dictKho = oldKho

195 │ // // .Zip(newKhoList, (oldItem, newItem) => new { OldId = oldItem.Ma, NewId = newItem.Ma })

196 │ // // .ToDictionary(x => x.OldId, x => x.NewId);

197 │

198 │ // // 5. Phòng Bệnh

199 │ // var newPhongList = newPhongQuery.ToList();

200 │ // dbcontext.TblPhongBenh.AddRange(newPhongList);

201 │ // dbcontext.SaveChanges();

202 │ // //var dictPhong = oldPhong

203 │ // // .Zip(newPhongList, (oldItem, newItem) => new { OldId = oldItem.Ma, NewId = newItem.Ma })

204 │ // // .ToDictionary(x => x.OldId, x => x.NewId);

205 │

206 │ // // 6. Hàng Hóa

207 │ // //var newHangHoaList = oldHanghoa.Select(t => new TblHangHoa

208 │ // //{

209 │ // // MaChiNhanh = objToken.branch,

210 │ // // MaKt = t.MaKt,

211 │ // // MaVach = t.MaVach,

212 │ // // MaLopHang = dictLopHang.TryGetValue(t.LopHang, out var newLopHangId) ? newLopHangId : (int?)...

213 │ // // Ten = t.TenHang,

214 │ // // MaDvt = dictDVT.TryGetValue(t.Dvt, out var newDvtId) ? newDvtId : (int?)null,

215 │ // // GiaBan = t.GiaBan,

216 │ // // GiaVon = t.GiaVon,

217 │ // // HinhAnh = t.HinhAnh,

218 │ // // GhiChu = t.GhiChu,

219 │ // // BanNoiBo = t.BanNoiBo,

220 │ // // HetHang = t.HetHang,

221 │

222 │ // //}).ToList();

223 │ // //dbcontext.TblHangHoa.AddRange(newHangHoaList);

224 │ // //dbcontext.SaveChanges();

225 │ // //var dictHangHoa = oldHanghoa

226 │ // // .Zip(newHangHoaList, (oldItem, newItem) => new { OldId = oldItem.HangHoaId, NewId = newItem....

227 │ // // .ToDictionary(x => x.OldId, x => x.NewId);

228 │

229 │ // ////// 7. Giá Bán Theo Kho

230 │ // //var newGiaBanList = oldGiaban.Select(t =>

231 │ // //{

232 │ // // if (dictKho.TryGetValue(t.MaKho ?? 0, out var newMaKhoId) &&

233 │ // // dictHangHoa.TryGetValue(t.MaHangHoa ?? 0, out var newMaHangHoaId))

234 │ // // {

235 │ // // return new Models.QLBHEntity.TblGiaBanTheoKho

236 │ // // {

237 │ // // MaChiNhanh = objToken.branch,

238 │ // // MaKho = newMaKhoId,

239 │ // // MaHangHoa = newMaHangHoaId,

240 │ // // GiaBan = t.GiaBan,

241 │ // // GhiChu = t.GhiChu,

242 │ // // Status = true,

243 │ // // };

244 │ // // }

245 │

246 │ // // return null;

247 │ // //}).Where(gb => gb != null).ToList();

248 │

249 │ // //dbcontext.TblGiaBanTheoKho.AddRange(newGiaBanList);

250 │ // //dbcontext.SaveChanges();

251 │

252 │ // ////user

253 │ // //var notInPresentUser = oldUser.Where(o => !presentUser.ToList().Select(x => x.Username).Contains...

254 │ // //var newUserList = notInPresentUser

255 │ // // .Select(t => new Models.QLBHEntity.SysUser

256 │ // // {

257 │ // // NId = t.NId,

258 │ // // BranchId = Branch.BVDaKhoaHongHung.GetEnumGuid(),

259 │ // // Username = t.Tendangnhap,

260 │ // // Password = Utility.GetSHA512($"{t.Tendangnhap}@2025"),

261 │ // // FullName = t.Hoten,

262 │ // // Msnv = t.Msnv,

263 │ // // Status = true,

264 │

265 │ // // });

266 │ // //dbcontext.SysUser.AddRange(newUserList);

267 │ // //dbcontext.SaveChanges();

268 │

269 │ // //var dictUser = newUserList

270 │ // // .Zip(newUserList, (oldItem, newItem) => new { OldId = oldItem.NId, NewId = newItem.Ma })

271 │ // // .ToDictionary(x => x.OldId, x => x.NewId);

272 │

273 │ // ////8. Nhập hàng

274 │ // //var newNhapHang = oldNhaphang.Select(t =>

275 │ // //{

276 │ // // if (dictKho.TryGetValue(t.MaKho ?? 0, out var newMaKhoId) &&

277 │ // // dictNCC.TryGetValue(t.NhaCc, out var newMaNhaCC))

278 │ // // {

279 │ // // return new Models.QLBHEntity.StockNhapHang

280 │ // // {

281 │ // // MaChiNhanh = objToken.branch,

282 │ // // MaKho = newMaKhoId,

283 │ // // MaLoaiNx = t.MaLoaiNx,

284 │ // // MaNhaCc = newMaNhaCC,

285 │ // // SoHoaDon = t.SoHd,

286 │ // // SoHoaDonNcc = t.SoHoaDonNcc,

287 │ // // NgayHoaDon = t.NgayHoaDon,

288 │ // // NgayNhapHang = t.NgayNhapHang,

289 │ // // TienChietKhau = t.TienChietKhau,

290 │ // // ThueVat = t.ThueVat,

291 │ // // GhiChu = t.GhiChu,

292 │ // // KhongHoaDon = t.KhongHoaDon

293 │

294 │ // // };

295 │ // // }

296 │

297 │ // // return null;

298 │ // //}).Where(gb => gb != null).ToList();

299 │

300 │ // //dbcontext.StockNhapHang.AddRange(newNhapHang);

301 │ // //dbcontext.SaveChanges();

302 │

303 │ // //var dictNhapHang = oldNhaphang

304 │ // // .Zip(newNhapHang, (oldItem, newItem) => new { OldId = oldItem.Ma, NewId = newItem.Ma })

305 │ // // .ToDictionary(x => x.OldId, x => x.NewId);

306 │

307 │

308 │ // ////9. Chi tiết Nhập hàng

309 │ // //var newChitietNhapHang = oldChiTietNhapHang.Select(t =>

310 │ // //{

311 │ // // if (dictNhapHang.TryGetValue(t.Ma, out var newMaNhapHang) &&

312 │ // // dictHangHoa.TryGetValue(t.MaHangHoa, out var newMaHanghoa))

313 │ // // {

314 │ // // return new Models.QLBHEntity.StockChiTietNhapHang

315 │ // // {

316 │ // // MaNhapHang = newMaNhapHang,

317 │ // // MaHangHoa = newMaHanghoa,

318 │ // // SoLuong = t.SoLuong,

319 │ // // GiaVon = t.GiaVon,

320 │ // // HanSd = t.HanSd,

321 │ // // ChietKhauTm = t.ChietKhauTm,

322 │ // // ChietKhauPt = t.ChietKhauPt,

323 │ // // Vat = t.Vat,

324 │ // // ThanhTien = t.ThanhTien,

325 │ // // ThanhTienSauCk = t.ThanhTienSauCk,

326 │ // // ThanhTienSauVat = t.ThanhTienSauVat,

327 │

328 │ // // };

329 │ // // }

330 │

331 │ // // return null;

332 │ // //}).Where(gb => gb != null).ToList();

333 │

334 │ // //dbcontext.StockChiTietNhapHang.AddRange(newChitietNhapHang);

335 │ // //dbcontext.SaveChanges();

336 │

337 │

338 │ // ////10. Xuất kho

339 │ // //var newXuatKho = oldXuatKho.Select(t =>

340 │ // //{

341 │ // // if (dictKho.TryGetValue(t.MaKho ?? 0, out var newMaKhoId) &&

342 │ // // dictKho.TryGetValue(t.MaKhoChuyenDen ?? 0, out var newMaKhoChuyenDen))

343 │ // // {

344 │ // // return new Models.QLBHEntity.StockXuatKho

345 │ // // {

346 │ // // MaChiNhanh = objToken.branch,

347 │ // // SoPhieu = t.SoPhieu,

348 │ // // NgayXuat = t.NgayXuat,

349 │ // // MaKho = newMaKhoId,

350 │ // // MaMayBanHang = t.MaMayBanHang,

351 │ // // MaLoaiNx = t.MaLoaiNx,

352 │ // // GhiChu = t.GhiChu,

353 │ // // MaKhoChuyenDen = newMaKhoChuyenDen,

354 │ // // KhoaSo = t.KhoaSo,

355 │

356 │ // // };

357 │ // // }

358 │

359 │ // // return null;

360 │ // //}).Where(gb => gb != null).ToList();

361 │

362 │ // //dbcontext.StockXuatKho.AddRange(newXuatKho);

363 │ // //dbcontext.SaveChanges();

364 │

365 │ // //var dictXuatKho = oldXuatKho

366 │ // // .Zip(newXuatKho, (oldItem, newItem) => new { OldId = oldItem.Ma, NewId = newItem.Ma })

367 │ // // .ToDictionary(x => x.OldId, x => x.NewId);

368 │

369 │

370 │ // ////11. Chi tiết Xuất kho

371 │ // //var newChitietXuatKho = oldChiTietXuatKho.Select(t =>

372 │ // //{

373 │ // // if (dictHangHoa.TryGetValue(t.MaHangHoa ?? 0, out var newMaHanghoa) &&

374 │ // // dictDVT.TryGetValue(t.MaDvt ?? 0, out var newMaDVT))

375 │ // // {

376 │ // // return new Models.QLBHEntity.StockChiTietXuatKho

377 │ // // {

378 │ // // MaPhieuXuat = t.MaPhieuXuat,

379 │ // // MaHangHoa = newMaHanghoa,

380 │ // // MaDvt = newMaDVT,

381 │ // // SoLuong = t.SoLuong,

382 │ // // GiaVon = t.GiaVon,

383 │ // // GhiChu = t.GhiChu,

384 │

385 │ // // };

386 │ // // }

387 │

388 │ // // return null;

389 │ // //}).Where(gb => gb != null).ToList();

390 │

391 │ // //dbcontext.StockChiTietXuatKho.AddRange(newChitietXuatKho);

392 │ // //dbcontext.SaveChanges();

393 │

394 │ // ////12. Bán hàng

395 │ // //var newHoaDonList = oldHoaDon.Select(t =>

396 │ // //{

397 │ // // if (dictKho.TryGetValue(t.MaKho ?? 0, out var newMaKhoId) &&

398 │ // // dictUser.TryGetValue(t.NguoiBan, out var newNguoiBan))

399 │ // // {

400 │ // // return new Models.QLBHEntity.OpsHoaDon

401 │ // // {

402 │ // // MaChiNhanh = objToken.branch,

403 │ // // MaKho = newMaKhoId,

404 │ // // MaMayBanHang = t.MaMayBanHang,

405 │ // // MaPhongBenh = t.MaPhongBenh,

406 │ // // NgayHoaDon = t.NgayHd,

407 │ // // SoSeri = t.SoSeri,

408 │ // // HinhThucTt = t.HinhThucTt,

409 │ // // GiamGia = t.GiaGiam,

410 │ // // PhaiThu = t.Trigia,

411 │ // // TienKhachTra = t.TienKhachTra,

412 │ // // TienThua = t.TienThua,

413 │ // // MaVachThe = t.MaVachThe,

414 │ // // TienThe = t.TienThe,

415 │ // // TienTraLaiKhach = t.TienTraLaiKhach,

416 │ // // TienTraLaiThe = t.TienTraLaiThe,

417 │ // // KhuyenMaiTienMat = t.KhuyenMaiTienMat,

418 │ // // RequestId = t.RequestId,

419 │ // // WebOrderId = t.WebOrderId,

420 │ // // MaThanhToan = t.MaThanhToan,

421 │ // // NgayThanhToan = t.NgayThanhToan,

422 │ // // NguoiThanhToan = newNguoiBan,

423 │ // // TongTienTraLai = t.TongTienTraLai,

424 │ // // GhiChu = t.GhiChu,

425 │ // // KhoaSo = t.KhoaSo,

426 │

427 │

428 │ // // };

429 │ // // }

430 │

431 │ // // return null;

432 │ // //}).Where(gb => gb != null).ToList();

433 │

434 │ // //dbcontext.OpsHoaDon.AddRange(newHoaDonList);

435 │ // //dbcontext.SaveChanges();

436 │

437 │ // //var dictHoaDon = oldHoaDon

438 │ // // .Zip(newHoaDonList, (oldItem, newItem) => new { OldId = oldItem.HoaDonId, NewId = newItem.Ma...

439 │ // // .ToDictionary(x => x.OldId, x => x.NewId);

440 │

441 │ // ////Chi tiết hóa đơn

442 │ // //var newChiTietHoaDon = oldChiTietHoaDon.Select(t =>

443 │ // //{

444 │ // // if (dictHoaDon.TryGetValue(t.HoaDonId, out var newMaHoaDon) &&

445 │ // // dictHangHoa.TryGetValue(t.HangHoaId, out var newMaHangHoa) &&

446 │ // // dictDVT.TryGetValue(t.MaDvt ?? 0, out var newMaDVT))

447 │ // // {

448 │ // // return new Models.QLBHEntity.OpsChiTietHoaDon

449 │ // // {

450 │ // // MaHoaDon = newMaHoaDon,

451 │ // // MaHangHoa = newMaHangHoa,

452 │ // // MaVach = t.MaVach,

453 │ // // MaDvt = newMaDVT,

454 │ // // SoLuong = t.SoLuong,

455 │ // // DonGia = t.DonGia,

456 │ // // GiaGiam = t.GiaGiam,

457 │ // // ThanhTien = t.ThanhTien,

458 │ // // SoLuongTraLai = t.SoLuongTraLai,

459 │ // // ThanhTienTraLai = t.ThanhTienTraLai,

460 │ // // NgayTra = t.NgayTra

461 │

462 │ // // };

463 │ // // }

464 │

465 │ // // return null;

466 │ // //}).Where(gb => gb != null).ToList();

467 │

468 │ // //dbcontext.OpsChiTietHoaDon.AddRange(newChiTietHoaDon);

469 │ // dbcontext.SaveChanges();

470 │

471 │ // transaction.Commit();

472 │ // }

473 │ // catch (Exception ex)

474 │ // {

475 │ // transaction.Rollback();

476 │ // return Json(new ResponseModelError(ex.Message));

477 │ // }

478 │ //}

479 │ return Json(new ResponseModelSuccess("Sync data thành công"));

480 │ }

481 │ }

482 │ }

────────────────────────────────────────────────────────────

**📄 ISignData.cs (CS, 8 lines)**

1 │ ﻿namespace QuanLyBanHang.Models.VietinBank

2 │ {

3 │ public interface ISignData

4 │ {

5 │ public string GetSignData();

6 │ public void SetSignature(string signature);

7 │ }

8 │ }

────────────────────────────────────────────────────────────

**📄 InqBillRequest.cs (CS, 66 lines)**

1 │ ﻿namespace QuanLyBanHang.Models.VietinBank

2 │ {

3 │ public class InqBillRequest : ISignData

4 │ {

5 │ public InqBillHeader Header { get; set; }

6 │ public InqBillData Data { get; set; }

7 │

8 │ public string GetSignData()

9 │ {

10 │ return Data.TransId + Data.ChannelId + Data.TransTime + Data.TransType + Data.CustCode + Data.CustName + Dat...

11 │ }

12 │

13 │ public void SetSignature(string signature)

14 │ {

15 │ Header.Signature = signature;

16 │ }

17 │ }

18 │

19 │ public class InqBillHeader

20 │ {

21 │ public string MsgId { get; set; }

22 │ public string MsgType { get; set; }

23 │ public string ChannelId { get; set; }

24 │ public string ProviderId { get; set; }

25 │ public string MerchantId { get; set; }

26 │ public string ProductId { get; set; }

27 │ public string Timestamp { get; set; }

28 │ public string Username { get; set; }

29 │ public string Encrypt { get; set; }

30 │ public string Signature { get; set; }

31 │ public string RecordNum { get; set; }

32 │ public string Token { get; set; }

33 │ public string ClientIP { get; set; }

34 │ public string Preseve1 { get; set; }

35 │ public string Preseve2 { get; set; }

36 │ public string Preseve3 { get; set; }

37 │ public object AdditionalProperties { get; set; }

38 │ }

39 │

40 │ public class InqBillData

41 │ {

42 │ public string TransId { get; set; }

43 │ public string ChannelId { get; set; }

44 │ public string TransTime { get; set; }

45 │ public string TransType { get; set; }

46 │ public string ServiceType { get; set; }

47 │ public string PaymentType { get; set; }

48 │ public string PaymentMethod { get; set; }

49 │ public string CustCode { get; set; }

50 │ public string CustName { get; set; }

51 │ public string CustAcct { get; set; }

52 │ public string IdCard { get; set; }

53 │ public string CustAddr { get; set; }

54 │ public string CustStreet { get; set; }

55 │ public string PhoneNo { get; set; }

56 │ public string Email { get; set; }

57 │ public string AreaCode { get; set; }

58 │ public string ProvinceCode { get; set; }

59 │ public string Tid { get; set; }

60 │ public object AddInfo { get; set; }

61 │ public string Preseve1 { get; set; }

62 │ public string Preseve2 { get; set; }

63 │ public string Preseve3 { get; set; }

64 │ public object AdditionalProperties { get; set; }

65 │ }

66 │ }

────────────────────────────────────────────────────────────

**📄 InqBillResponse.cs (CS, 107 lines)**

1 │ ﻿namespace QuanLyBanHang.Models.VietinBank

2 │ {

3 │ public class InqBillResponse : ISignData

4 │ {

5 │ public InqBillHeader Header { get; set; }

6 │ public InqBillResponseData Data { get; set; }

7 │

8 │ public string GetSignData()

9 │ {

10 │ return Data.Details.TransId + Data.Details.TransTime + Data.Details.CustCode + Data.Details.CustName + Data....

11 │ }

12 │

13 │ public void SetSignature(string signature)

14 │ {

15 │ Header.Signature = signature;

16 │ }

17 │ }

18 │

19 │ public class InqBillResponseData

20 │ {

21 │ public InqBillError Errors { get; set; }

22 │ public InqBillDetails Details { get; set; }

23 │ }

24 │

25 │ public class InqBillError

26 │ {

27 │ public string ErrorCode { get; set; }

28 │ public string ErrorDesc { get; set; }

29 │ }

30 │

31 │ public class InqBillDetails

32 │ {

33 │ public string TransId { get; set; }

34 │ public string TransTime { get; set; }

35 │ public string CustCode { get; set; }

36 │ public string CustName { get; set; }

37 │ public string CustAddr { get; set; }

38 │ public string CustPhone { get; set; }

39 │ public string CustEmail { get; set; }

40 │ public string CustAcct { get; set; }

41 │ public string BankCode { get; set; }

42 │ public string BranchCode { get; set; }

43 │ public string BillId { get; set; }

44 │ public string BillTerm { get; set; }

45 │ public string Amount { get; set; }

46 │ public string AmountMin { get; set; }

47 │ public string Fee { get; set; }

48 │ public string CurrencyCode { get; set; }

49 │ public string OutPayDate { get; set; }

50 │ public string ExpireDate { get; set; }

51 │ public string PaymentType { get; set; }

52 │ public string PriceList { get; set; }

53 │ public string Quota { get; set; }

54 │ public string Consume { get; set; }

55 │ public string IdxStart { get; set; }

56 │ public string IdxEnd { get; set; }

57 │ public string Vat { get; set; }

58 │ public string RaCode { get; set; }

59 │ public string RaName { get; set; }

60 │ public string ProvinceCode { get; set; }

61 │ public string ProvinceName { get; set; }

62 │ public string AreaCode { get; set; }

63 │ public string AreaName { get; set; }

64 │ public string TaxCode { get; set; }

65 │ public List<InqBillSubBill> BillInfo1 { get; set; }

66 │ public object BillInfo2 { get; set; }

67 │ public object BillInfo3 { get; set; }

68 │ public string Preseve1 { get; set; }

69 │ public string Preseve2 { get; set; }

70 │ public string Preseve3 { get; set; }

71 │ }

72 │

73 │ public class InqBillSubBill

74 │ {

75 │ public string CustCode { get; set; }

76 │ public string CustName { get; set; }

77 │ public string CustAddr { get; set; }

78 │ public string CustPhone { get; set; }

79 │ public string CustEmail { get; set; }

80 │ public string CustAcct { get; set; }

81 │ public string BankCode { get; set; }

82 │ public string BranchCode { get; set; }

83 │ public string BillId { get; set; }

84 │ public string BillTerm { get; set; }

85 │ public string Amount { get; set; }

86 │ public string AmountMin { get; set; }

87 │ public string Fee { get; set; }

88 │ public string CurrencyCode { get; set; }

89 │ public string OutPayDate { get; set; }

90 │ public string ExpireDate { get; set; }

91 │ public string PaymentType { get; set; }

92 │ public string PriceList { get; set; }

93 │ public string Quota { get; set; }

94 │ public string Consume { get; set; }

95 │ public string IdxStart { get; set; }

96 │ public string IdxEnd { get; set; }

97 │ public string Vat { get; set; }

98 │ public string RaCode { get; set; }

99 │ public string RaName { get; set; }

100 │ public string ProvinceCode { get; set; }

101 │ public string ProvinceName { get; set; }

102 │ public string AreaCode { get; set; }

103 │ public string AreaName { get; set; }

104 │ public string TaxCode { get; set; }

105 │ public object AddInfo { get; set; }

106 │ }

107 │ }

────────────────────────────────────────────────────────────

**📄 Inv.cs (CS, 13 lines)**

1 │ using System.Xml.Serialization;

2 │

3 │ namespace QuanLyBanHang.Models.EInvoiceService

4 │ {

5 │ public class Inv

6 │ {

7 │ [XmlElement("key")]

8 │ public string? Key { get; set; }

9 │

10 │ [XmlElement("Invoice")]

11 │ public Invoice? Invoice { get; set; }

12 │ }

13 │ }

────────────────────────────────────────────────────────────

**📄 InvLogVnpt.cs (CS, 21 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class InvLogVnpt

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string? InvKey { get; set; }

11 │

12 │ public string? SoapEnvelope { get; set; }

13 │

14 │ public string? StatusCode { get; set; }

15 │

16 │ public string? ReasonPhrase { get; set; }

17 │

18 │ public string? Result { get; set; }

19 │

20 │ public DateTime? CreatedDate { get; set; }

21 │ }

────────────────────────────────────────────────────────────

**📄 Invoice.cs (CS, 72 lines)**

1 │ using System.Xml.Serialization;

2 │

3 │ namespace QuanLyBanHang.Models.EInvoiceService

4 │ {

5 │ public class Invoice

6 │ {

7 │ public string? InvoiceNo { get; set; }

8 │ public string? CusCode { get; set; }

9 │ public string? CusName { get; set; }

10 │ public string? CusAddress { get; set; }

11 │ public string? CusPhone { get; set; }

12 │ public string? CusTaxCode { get; set; }

13 │ public string? PaymentMethod { get; set; }

14 │ public string? KindOfService { get; set; }

15 │

16 │ [XmlArray("Products")]

17 │ [XmlArrayItem("Product")]

18 │ public List<Product>? Products { get; set; }

19 │

20 │ public string? Total { get; set; }

21 │ public string? DiscountAmount { get; set; }

22 │ public string? VATRate { get; set; }

23 │ public string? VATAmount { get; set; }

24 │ public string? Amount { get; set; }

25 │ public string? AmountInWords { get; set; }

26 │ public string? Extra { get; set; }

27 │ public string? ArisingDate { get; set; }

28 │ public string? PaymentStatus { get; set; }

29 │ public string? EmailDeliver { get; set; }

30 │ public string? ComName { get; set; }

31 │ public string? ComAddress { get; set; }

32 │ public string? ComTaxCode { get; set; }

33 │ public string? ComFax { get; set; }

34 │ public string? ResourceCode { get; set; }

35 │ public string? GrossValue { get; set; }

36 │ public string? GrossValue0 { get; set; }

37 │ public string? VatAmount0 { get; set; }

38 │ public string? GrossValue5 { get; set; }

39 │ public string? VatAmount5 { get; set; }

40 │ public string? GrossValue10 { get; set; }

41 │ public string? VatAmount10 { get; set; }

42 │ public string? Buyer { get; set; }

43 │ public string? Name { get; set; }

44 │ public string? ComPhone { get; set; }

45 │ public string? ComBankName { get; set; }

46 │ public string? ComBankNo { get; set; }

47 │ public string? CreateDate { get; set; }

48 │ public string? DiscountRate { get; set; }

49 │ public string? CusSignStatus { get; set; }

50 │ public string? CreateBy { get; set; }

51 │ public string? PublishBy { get; set; }

52 │ public string? Note { get; set; }

53 │ public string? ProcessInvNote { get; set; }

54 │ public string? Fkey { get; set; }

55 │ public string? GrossValue\_NonTax { get; set; }

56 │ public string? CurrencyUnit { get; set; }

57 │ public string? ExchangeRate { get; set; }

58 │ public string? ConvertedAmount { get; set; }

59 │ public string? Extra1 { get; set; }

60 │ public string? Extra2 { get; set; }

61 │ public string? SMSDeliver { get; set; }

62 │ public string? LDDNBo { get; set; }

63 │ public string? HDSo { get; set; }

64 │ public string? HVTNXHang { get; set; }

65 │ public string? TNVChuyen { get; set; }

66 │ public string? PTVChuyen { get; set; }

67 │ public string? HDKTSo { get; set; }

68 │ public string? HDKTNgay { get; set; }

69 │ public string? DiscountNonTax { get; set; }

70 │ public string? DiscountOther { get; set; }

71 │ }

72 │ }

────────────────────────────────────────────────────────────

**📄 Invoices.cs (CS, 10 lines)**

1 │ using System.Xml.Serialization;

2 │ namespace QuanLyBanHang.Models.EInvoiceService

3 │ {

4 │ [XmlRoot("Invoices")]

5 │ public class Invoices

6 │ {

7 │ [XmlElement("Inv")]

8 │ public List<Inv>? InvList { get; set; }

9 │ }

10 │ }

────────────────────────────────────────────────────────────

**📄 KhayController.cs (CS, 546 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using Microsoft.EntityFrameworkCore;

3 │ using QuanLyBanHang.Common;

4 │ using QuanLyBanHang.Common.Attributes;

5 │ using QuanLyBanHang.Models.APIModels;

6 │ using QuanLyBanHang.Models.ExModel.VTTH;

7 │ using QuanLyBanHang.Models.QLBHEntity;

8 │ using System.Data;

9 │

10 │ namespace QuanLyBanHang.Controllers.VTTH

11 │ {

12 │ public class KhayController : BaseController

13 │ {

14 │ private readonly IWebHostEnvironment \_hostingEnvironment;

15 │

16 │ public KhayController(QuanLyBanHangContext dbcontext, MemoryCacheManager memoryCache, IWebHostEnvironment hostin...

17 │ {

18 │ m\_MenuID = MenuID.vtthKhay.GetEnumGuid();

19 │ \_hostingEnvironment = hostingEnvironment;

20 │ }

21 │

22 │ [TokenUserCheckHTTP(Permission.NotCheck)]

23 │ public IActionResult Index(string NhapKhay)

24 │ {

25 │ bool? nhapKhayFilter = null;

26 │ if (!string.IsNullOrEmpty(NhapKhay))

27 │ {

28 │ nhapKhayFilter = NhapKhay == "1";

29 │ }

30 │

31 │ var query = dbcontext.VtthKhay.AsNoTracking().Where(t => !(t.Deleted ?? false) && t.MaChiNhanh == objToken.b...

32 │ .Select(khay => new

33 │ {

34 │ khay,

35 │ tongNhap = dbcontext.VtthNhapXuatKhay

36 │ .Where(nx => nx.MaKhay == khay.Ma && nx.LoaiPhieu == (int)LoaiNhapXuatVTTH.Nhap)

37 │ .Sum(nx => (decimal?)nx.SoLuongNhap) ?? 0,

38 │ tongXuat = dbcontext.VtthNhapXuatKhay

39 │ .Where(nx => nx.MaKhay == khay.Ma && nx.LoaiPhieu == (int)LoaiNhapXuatVTTH.Xuat)

40 │ .Sum(nx => (decimal?)nx.SoLuongXuat) ?? 0,

41 │ tongTra = dbcontext.VtthNhapXuatKhay

42 │ .Where(nx => nx.MaKhay == khay.Ma && nx.LoaiPhieu == (int)LoaiNhapXuatVTTH.Tra && (nx.XacNhanTra ?? ...

43 │ .Sum(nx => (decimal?)nx.SoLuongTra) ?? 0,

44 │ })

45 │ .AsEnumerable()

46 │ .Select(t => new vKhay

47 │ {

48 │ Ma = t.khay.Ma,

49 │ MaKhoTieuHao = t.khay.MaKhoTieuHao,

50 │ MaKhoNguyenLieu = t.khay.MaKhoNguyenLieu,

51 │ MaVatTu = t.khay.MaVatTu,

52 │ DinhMucKhay = t.khay.DinhMucKhay,

53 │ DaNhap = t.khay.DaNhap,

54 │ KhayTreo = t.khay.KhayTreo,

55 │ MoTa = t.khay.MoTa,

56 │ ConLai = t.tongNhap - t.tongXuat - t.tongTra,

57 │ Status = t.khay.Status,

58 │

59 │ });

60 │ if (nhapKhayFilter.HasValue)

61 │ {

62 │ query = query.Where(t => (t.DaNhap ?? false) == nhapKhayFilter.Value);

63 │ }

64 │

65 │ ViewData["ListData"] = query.ToList();

66 │

67 │ ViewData["ListVatTu"] = dbcontext.TblHangHoa

68 │ .AsNoTracking()

69 │ .Where(vt => (vt.VatTu ?? false) && !(vt.Deleted ?? false) && (vt.Status ?? false))

70 │ .ToList();

71 │ ViewData["ListKho"] = dbcontext.StockDanhMucKho

72 │ .AsNoTracking()

73 │ .Where(x => x.MaChiNhanh == objToken.branch)

74 │ .ToList();

75 │

76 │ ViewData["NhapKhay"] = NhapKhay;

77 │

78 │ return View();

79 │ }

80 │

81 │ [TokenUserCheckHTTP(Permission.Add)]

82 │ public IActionResult Add()

83 │ {

84 │ ViewData["ListKho"] = dbcontext.StockDanhMucKho.AsNoTracking().Where(x => x.MaChiNhanh == objToken.branch &&...

85 │ ViewData["ListPhongBan"] = dbcontext.TblPhongBan.AsNoTracking().Where(t => (t.Status ?? false) && !(t.Delete...

86 │ ViewData["ListVatTu"] = (from vt in dbcontext.TblHangHoa

87 │ where (vt.VatTu ?? false) && !(vt.Deleted ?? false) && (vt.Status ?? false)

88 │ select vt).ToList();

89 │ return View("Form");

90 │ }

91 │

92 │ [TokenUserCheckHTTP(Permission.Edit)]

93 │ public IActionResult Edit(string Ma)

94 │ {

95 │ ViewData["ListKho"] = dbcontext.StockDanhMucKho.AsNoTracking().Where(x => x.MaChiNhanh == objToken.branch &&...

96 │ ViewData["ListPhongBan"] = dbcontext.TblPhongBan.AsNoTracking().Where(t => (t.Status ?? false) && !(t.Delete...

97 │ ViewData["ListVatTu"] = (from vt in dbcontext.TblHangHoa.AsNoTracking()

98 │ where (vt.VatTu ?? false) && !(vt.Deleted ?? false) && (vt.Status ?? false)

99 │ select vt).ToList();

100 │ ViewData["item"] = dbcontext.VtthKhay.FirstOrDefault(t => t.Ma == Ma.ToInt() && !(t.Deleted ?? false));

101 │ return View("Form");

102 │ }

103 │

104 │ [TokenUserCheckHTTP(Permission.NotCheck)]

105 │ public IActionResult ChiTiet(string Ma)

106 │ {

107 │ ViewData["Ma"] = Ma;

108 │ return View("Edit");

109 │ }

110 │

111 │ [HttpPost]

112 │ [TokenUserCheckHTTP(Permission.Add)]

113 │ public IActionResult AddAPI(string MaKhoNhap, string MaVatTu, string DinhMucKhay, string MoTa, string Status, st...

114 │ {

115 │ if (MaVatTu.IsEmpty())

116 │ {

117 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng chọn vật tư"));

118 │ }

119 │ if (DinhMucKhay.IsEmpty())

120 │ {

121 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng nhập định mức khay"));

122 │ }

123 │ if (Status.IsEmpty())

124 │ {

125 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng chọn trạng thái"));

126 │ }

127 │ if (MaKhoNhap.IsEmpty())

128 │ {

129 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng chọn kho nhận"));

130 │ }

131 │ if (MaKhoXuat.IsEmpty())

132 │ {

133 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng chọn kho xuất"));

134 │ }

135 │ if (MaKhoNhap == MaKhoXuat)

136 │ {

137 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng nhập kho xuất khác kho nhận hàng"));

138 │ }

139 │ StockDanhMucKho? khovt = dbcontext.StockDanhMucKho.FirstOrDefault(t => t.MaChiNhanh == objToken.branch

140 │ && !(t.Deleted ?? false));

141 │ var br = dbcontext.SysBranch.FirstOrDefault(x => x.Ma == objToken.branch);

142 │ try

143 │ {

144 │ VtthKhay item = new VtthKhay

145 │ {

146 │ MaKhoNguyenLieu = MaKhoNhap.ToInt(),

147 │ MaKhoTieuHao = MaKhoXuat.ToInt(),

148 │ MaVatTu = MaVatTu.ToInt(),

149 │ DinhMucKhay = DinhMucKhay.ToDecimal(),

150 │ MoTa = MoTa,

151 │ Status = Status == "1",

152 │ CreatedDate = DateTime.Now,

153 │ UserCreated = objToken.userid,

154 │ MaChiNhanh = objToken.branch,

155 │ MaChiNhanhInt = br.Nid ?? 0,

156 │ };

157 │ dbcontext.Add(item);

158 │ dbcontext.SaveChanges();

159 │

160 │ return Json(new ResponseModel(QLBHConst.OK, "Đã thêm"));

161 │ }

162 │ catch (Exception ex)

163 │ {

164 │ return Json(new ResponseModel(QLBHConst.Error, ex.Message));

165 │ }

166 │ }

167 │

168 │ [HttpPost]

169 │ [TokenUserCheckHTTP(Permission.Edit)]

170 │ public IActionResult UpdateAPI(string Ma, string MaKhoNhap, string MaVatTu, string DinhMucKhay, string MoTa, str...

171 │ {

172 │ VtthKhay? item = dbcontext.VtthKhay.FirstOrDefault(t => t.Ma == Ma.ToInt() && !(t.Deleted ?? false));

173 │ if (item == null)

174 │ {

175 │ return Json(new ResponseModel(QLBHConst.Error, "Khay vật tư không tồn tại"));

176 │ }

177 │ if (item.KhayTreo ?? false)

178 │ {

179 │ return Json(new ResponseModel(QLBHConst.Error, "Khay vật tư đang treo không sửa được"));

180 │ }

181 │ if (item.DaNhap ?? false)

182 │ {

183 │ return Json(new ResponseModel(QLBHConst.Error, "Khay vật tư đang đầy không sửa được"));

184 │ }

185 │ if (MaVatTu.IsEmpty())

186 │ {

187 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng chọn vật tư"));

188 │ }

189 │ if (DinhMucKhay.IsEmpty())

190 │ {

191 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng nhập định mức khay"));

192 │ }

193 │ if (Status.IsEmpty())

194 │ {

195 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng chọn trạng thái"));

196 │ }

197 │ if (MaKhoNhap.IsEmpty())

198 │ {

199 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng chọn kho nhận"));

200 │ }

201 │ if (MaKhoXuat.IsEmpty())

202 │ {

203 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng chọn kho xuất"));

204 │ }

205 │ if (MaKhoNhap == MaKhoXuat)

206 │ {

207 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng nhập kho xuất khác kho nhận hàng"));

208 │ }

209 │

210 │ try

211 │ {

212 │

213 │ VtthNhapXuatKhay? nx = dbcontext.VtthNhapXuatKhay.FirstOrDefault(t => t.MaKhay == item.Ma && !(t.Deleted...

214 │ if (nx == null)

215 │ {

216 │ StockDanhMucKho? khovt = dbcontext.StockDanhMucKho.FirstOrDefault(t => t.MaChiNhanh == objToken.bran...

217 │ && !(t.Deleted ?? false));

218 │

219 │ item.MaKhoNguyenLieu = MaKhoNhap.ToInt();

220 │ item.MaKhoTieuHao = MaKhoXuat.ToInt();

221 │ }

222 │ item.DinhMucKhay = DinhMucKhay.ToDecimal();

223 │ item.MoTa = MoTa;

224 │ item.Status = Status == "1";

225 │ item.MaVatTu = MaVatTu.ToInt();

226 │ item.ModifiedDate = DateTime.Now;

227 │ item.UserModified = objToken.userid;

228 │

229 │ dbcontext.Update(item);

230 │ dbcontext.SaveChanges();

231 │ return Json(new ResponseModel(QLBHConst.OK, "Đã cập nhật"));

232 │ }

233 │ catch (Exception ex)

234 │ {

235 │ return Json(new ResponseModel(QLBHConst.Error, ex.Message));

236 │ }

237 │ }

238 │

239 │ [HttpPost]

240 │ [TokenUserCheckHTTP(Permission.Delete)]

241 │ public IActionResult DeleteAPI(string Ma)

242 │ {

243 │ VtthKhay? item = dbcontext.VtthKhay.FirstOrDefault(t => t.Ma == Ma.ToInt() && !(t.Deleted ?? false));

244 │

245 │ if (item == null)

246 │ {

247 │ return Json(new ResponseModel(QLBHConst.Error, "Khay vật tư không tồn tại"));

248 │ }

249 │ VtthNhapXuatKhay? nx = dbcontext.VtthNhapXuatKhay.FirstOrDefault(t => t.MaKhay == Ma.ToInt() && !(t.Deleted ...

250 │ if (nx != null)

251 │ {

252 │ return Json(new ResponseModel(QLBHConst.Error, "Khay vật tư đã có nhập hoặc xuất không th��...

253 │ }

254 │

255 │ try

256 │ {

257 │ item.Deleted = true;

258 │ item.DeletedDate = DateTime.Now;

259 │ item.UserDeleted = objToken.userid;

260 │

261 │ dbcontext.Update(item);

262 │ dbcontext.SaveChanges();

263 │ return Json(new ResponseModel(QLBHConst.OK, "Đã xóa"));

264 │ }

265 │ catch (Exception ex)

266 │ {

267 │ return Json(new ResponseModel(QLBHConst.Error, ex.Message));

268 │ }

269 │ }

270 │

271 │ [HttpPost]

272 │ public IActionResult getListKhayTheoBrand()

273 │ {

274 │ return Json(new ResponseModelSuccess("", dbcontext.VtthKhay.AsNoTracking().Where(e => (e.MaChiNhanh == objTo...

275 │ && (e.Status ?? false) && !(e.Deleted ?? false)).Select(t => new

276 │ {

277 │ ma = t.Ma + "",

278 │ khaytreo = t.KhayTreo ?? false ? "1" : "0",

279 │ khayday = t.DaNhap ?? false ? "1" : "0",

280 │ trangthai = t.Status ?? false ? "1" : "0"

281 │ }).ToList()));

282 │ }

283 │

284 │

285 │ [TokenUserCheckHTTP(Permission.NotCheck)]

286 │ public IActionResult getListExcelKhayTheoKhoaPhong(string FileType)

287 │ {

288 │ var listData = (from vtth in dbcontext.VtthKhay.AsNoTracking()

289 │ join vt in dbcontext.TblHangHoa.AsNoTracking() on vtth.MaVatTu equals vt.Ma

290 │ join hh in dbcontext.TblHangHoa.AsNoTracking() on vt.Ma equals hh.Ma

291 │ join k in dbcontext.StockDanhMucKho.AsNoTracking() on vtth.MaKhoNguyenLieu equals k.Ma

292 │ join kx in dbcontext.StockDanhMucKho.AsNoTracking() on vtth.MaKhoTieuHao equals kx.Ma

293 │ join kp in dbcontext.SysBranch.AsNoTracking() on vtth.MaChiNhanh equals kp.Ma

294 │ where (vtth.MaChiNhanh == objToken.branch)

295 │ && (vtth.Status ?? false) && !(vtth.Deleted ?? false)

296 │ && (vt.VatTu ?? false) && !(vt.Deleted ?? false)

297 │ select new

298 │ {

299 │ ma = vtth.Ma + "",

300 │ manhapkhay = vtth.MaNhapKhay + "",

301 │ codevt = vt.MaKt + "",

302 │ tenkhonhap = k.Ten + "",

303 │ tenkhoxuat = kx.Ten + "",

304 │ tenvt = vt.Ten?? "",

305 │ dinhmuckhay = vtth.DinhMucKhay ?? 0,

306 │ mota = vtth.MoTa ?? "",

307 │ tenchinhanh = kp.Ten + "",

308 │ khaytreo = vtth.KhayTreo ?? false ? "Khay treo" : "",

309 │ khayday = vtth.DaNhap ?? false ? "Đã nhập" : "",

310 │ trangthai = vtth.Status ?? false ? "Hoạt động" : "Không hoạt động",

311 │ ngaytao = vtth.CreatedDate

312 │ })

313 │ .OrderByDescending(x => x.ngaytao.HasValue ? x.ngaytao : DateTime.MaxValue)

314 │ .ToList();

315 │

316 │ if (FileType.IsEmpty())

317 │ {

318 │ return Json(new ResponseModelSuccess("OK", listData));

319 │ }

320 │ string fileName = FileType == "Excel" ? "BaoCaoKhayVTTH.xlsx" : "BaoCaoKhayVTTH.pdf";

321 │ string contentType = FileType == "Excel" ? "application/ms-excel" : "application/pdf";

322 │ int startRow = 5;

323 │ List<ExcelValue> ListGiaTriRiengLe = new List<ExcelValue>();

324 │ List<ColumnFormat>? ListHeaderFormat = new List<ColumnFormat>();

325 │ List<ExcelFooter>? ListFooter = new List<ExcelFooter>();

326 │ List<ColumnFormat>? ListColumnFormat = new List<ColumnFormat>();

327 │

328 │ DataTable dt = new DataTable();

329 │ dt.Columns.Add("STT", typeof(int));

330 │ dt.Columns.Add("Code Khay", typeof(string));

331 │ dt.Columns.Add("Code Vật tư", typeof(string));

332 │ dt.Columns.Add("Tên vật tư", typeof(string));

333 │ dt.Columns.Add("Tên kho xuất", typeof(string));

334 │ dt.Columns.Add("Tên kho nhập", typeof(string));

335 │ dt.Columns.Add("Chi Nhánh", typeof(string));

336 │ dt.Columns.Add("Định mức", typeof(decimal));

337 │ dt.Columns.Add("Mô tả", typeof(string));

338 │ var i = 1;

339 │ foreach (var data in listData)

340 │ {

341 │ dt.Rows.Add(new object[]

342 │ {

343 │ i++,

344 │ data.ma,

345 │ data.codevt,

346 │ data.tenvt,

347 │ data.tenkhoxuat,

348 │ data.tenkhonhap,

349 │ data.tenchinhanh.ToUpper(),

350 │ data.dinhmuckhay,

351 │ data.mota,

352 │ });

353 │ }

354 │ string fileTemplate = Path.Combine(\_hostingEnvironment.ContentRootPath, "wwwroot", "templates", "BaoCaoKhayV...

355 │

356 │ try

357 │ {

358 │ Stream x = ExportToExcel.FromTemplate(fileTemplate, FileType, ListGiaTriRiengLe, dt, true, startRow, "A"...

359 │ return new FileStreamResult(x, contentType)

360 │ {

361 │ FileDownloadName = fileName

362 │ };

363 │ }

364 │ catch (Exception ex)

365 │ {

366 │

367 │ Console.WriteLine("Error generating Excel file: " + ex.Message);

368 │ return BadRequest("Failed to generate Excel file.");

369 │ }

370 │ }

371 │

372 │ //---------------API DÙNG CHO APP ---------------//

373 │

374 │ /// <summary>

375 │ /// API lấy danh sách khay VTTH với thông tin đầy đủ

376 │ /// </summary>

377 │ [HttpPost]

378 │ [TokenUserCheckHTTP(Permission.NotCheck)]

379 │ public IActionResult getKhayListAPI(string NhapKhay)

380 │ {

381 │ try

382 │ {

383 │ bool? nhapKhayFilter = null;

384 │ if (!string.IsNullOrEmpty(NhapKhay))

385 │ {

386 │ nhapKhayFilter = NhapKhay == "1";

387 │ }

388 │

389 │ var query = (from vtth in dbcontext.VtthKhay.AsNoTracking()

390 │ join vt in dbcontext.TblHangHoa.AsNoTracking() on vtth.MaVatTu equals vt.Ma into vtGroup

391 │ from vt in vtGroup.DefaultIfEmpty()

392 │ join dvt in dbcontext.TblDonViTinh.AsNoTracking() on vt.MaDvt equals dvt.Ma into dvtGroup

393 │ from dvt in dvtGroup.DefaultIfEmpty()

394 │ join khoNL in dbcontext.StockDanhMucKho.AsNoTracking() on vtth.MaKhoNguyenLieu equals khoNL...

395 │ from khoNL in khoNLGroup.DefaultIfEmpty()

396 │ join khoTH in dbcontext.StockDanhMucKho.AsNoTracking() on vtth.MaKhoTieuHao equals khoTH.Ma...

397 │ from khoTH in khoTHGroup.DefaultIfEmpty()

398 │ join pb in dbcontext.TblPhongBan.AsNoTracking() on vtth.MaPhongBan equals pb.Ma into pbGrou...

399 │ from pb in pbGroup.DefaultIfEmpty()

400 │ join cn in dbcontext.SysBranch.AsNoTracking() on vtth.MaChiNhanh equals cn.Ma into cnGroup

401 │ from cn in cnGroup.DefaultIfEmpty()

402 │ where !(vtth.Deleted ?? false) && vtth.MaChiNhanh == objToken.branch

403 │ select new

404 │ {

405 │ ma = vtth.Ma,

406 │ maVatTu = vtth.MaVatTu ?? 0,

407 │ codeVatTu = vt != null ? (vt.MaKt ?? "") : "",

408 │ tenVatTu = vt != null ? (vt.Ten ?? "") : "",

409 │ tenKhoNguyenLieu = khoNL != null ? (khoNL.Ten ?? "") : "",

410 │ tenKhoTieuHao = khoTH != null ? (khoTH.Ten ?? "") : "",

411 │ tenPhongBan = pb != null ? (pb.Ten ?? "") : "",

412 │ dinhMucKhay = vtth.DinhMucKhay ?? 0,

413 │ moTa = vtth.MoTa ?? "",

414 │ daNhap = vtth.DaNhap ?? false,

415 │ khayTreo = vtth.KhayTreo ?? false,

416 │ status = vtth.Status ?? false,

417 │ maNhapKhay = vtth.MaNhapKhay ?? 0,

418 │ ngayTao = vtth.CreatedDate, // Giữ nguyên DateTime

419 │ tenChiNhanh = cn != null ? (cn.Ten ?? "") : "",

420 │ tenDvt = dvt.Ten,

421 │ });

422 │

423 │ // Áp dụng filter nếu có

424 │ if (nhapKhayFilter.HasValue)

425 │ {

426 │ query = query.Where(t => t.daNhap == nhapKhayFilter.Value);

427 │ }

428 │

429 │ // Sắp xếp theo DateTime gốc và lấy data về client

430 │ var rawData = query.OrderByDescending(x => x.ngayTao).ToList();

431 │

432 │ // Format ngày tháng ở client side

433 │ var listData = rawData.Select(t => new

434 │ {

435 │ ma = t.ma,

436 │ maVatTu = t.maVatTu,

437 │ codeVatTu = t.codeVatTu,

438 │ tenVatTu = t.tenVatTu,

439 │ tenKhoNguyenLieu = t.tenKhoNguyenLieu,

440 │ tenKhoTieuHao = t.tenKhoTieuHao,

441 │ tenPhongBan = t.tenPhongBan,

442 │ dinhMucKhay = t.dinhMucKhay,

443 │ moTa = t.moTa,

444 │ daNhap = t.daNhap,

445 │ khayTreo = t.khayTreo,

446 │ status = t.status,

447 │ maNhapKhay = t.maNhapKhay,

448 │ ngayTao = t.ngayTao?.ToString("dd/MM/yyyy") ?? "", // Format ở đây

449 │ tenChiNhanh = t.tenChiNhanh,

450 │ tenDvt = t.tenDvt,

451 │ }).ToList();

452 │

453 │ return Json(new ResponseModelSuccess("OK", listData));

454 │ }

455 │ catch (Exception ex)

456 │ {

457 │ return Json(new ResponseModel(QLBHConst.Error, $"Lỗi khi lấy danh sách khay: {ex.Message}"));

458 │ }

459 │ }

460 │

461 │ /// <summary>

462 │ /// API lấy thông tin khay theo mã (cho QR Scanner)

463 │ /// </summary>

464 │ [HttpPost]

465 │ [TokenUserCheckHTTP(Permission.NotCheck)]

466 │ public IActionResult getKhayByMaAPI(string MaKhay)

467 │ {

468 │ try

469 │ {

470 │ if (string.IsNullOrEmpty(MaKhay))

471 │ {

472 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng nhập mã khay"));

473 │ }

474 │

475 │ var rawKhayData = (from vtth in dbcontext.VtthKhay.AsNoTracking()

476 │ join vt in dbcontext.TblHangHoa.AsNoTracking() on vtth.MaVatTu equals vt.Ma into vtGr...

477 │ from vt in vtGroup.DefaultIfEmpty()

478 │ join dvt in dbcontext.TblDonViTinh.AsNoTracking() on vt.MaDvt equals dvt.Ma into dvtG...

479 │ from dvt in dvtGroup.DefaultIfEmpty()

480 │ join khoNL in dbcontext.StockDanhMucKho.AsNoTracking() on vtth.MaKhoNguyenLieu equals...

481 │ from khoNL in khoNLGroup.DefaultIfEmpty()

482 │ join khoTH in dbcontext.StockDanhMucKho.AsNoTracking() on vtth.MaKhoTieuHao equals kh...

483 │ from khoTH in khoTHGroup.DefaultIfEmpty()

484 │ join pb in dbcontext.TblPhongBan.AsNoTracking() on vtth.MaPhongBan equals pb.Ma into ...

485 │ from pb in pbGroup.DefaultIfEmpty()

486 │ join cn in dbcontext.SysBranch.AsNoTracking() on vtth.MaChiNhanh equals cn.Ma into cn...

487 │ from cn in cnGroup.DefaultIfEmpty()

488 │ where vtth.Ma == MaKhay.ToInt()

489 │ && !(vtth.Deleted ?? false)

490 │ && vtth.MaChiNhanh == objToken.branch

491 │ select new

492 │ {

493 │ ma = vtth.Ma,

494 │ maVatTu = vtth.MaVatTu ?? 0,

495 │ codeVatTu = vt != null ? (vt.MaKt ?? "") : "",

496 │ tenVatTu = vt != null ? (vt.Ten ?? "") : "",

497 │ tenKhoNguyenLieu = khoNL != null ? (khoNL.Ten ?? "") : "",

498 │ tenKhoTieuHao = khoTH != null ? (khoTH.Ten ?? "") : "",

499 │ tenPhongBan = pb != null ? (pb.Ten ?? "") : "",

500 │ dinhMucKhay = vtth.DinhMucKhay ?? 0,

501 │ moTa = vtth.MoTa ?? "",

502 │ daNhap = vtth.DaNhap ?? false,

503 │ khayTreo = vtth.KhayTreo ?? false,

504 │ status = vtth.Status ?? false,

505 │ maNhapKhay = vtth.MaNhapKhay ?? 0,

506 │ ngayTao = vtth.CreatedDate, // Giữ nguyên DateTime

507 │ tenChiNhanh = cn != null ? (cn.Ten ?? "") : "",

508 │ tenDvt = dvt.Ten,

509 │ }).FirstOrDefault();

510 │

511 │ if (rawKhayData == null)

512 │ {

513 │ return Json(new ResponseModel(QLBHConst.Error, "Không tìm thấy khay hoặc khay không thuộc c...

514 │ }

515 │

516 │ // Format ngày tháng ở client side

517 │ var khayData = new

518 │ {

519 │ ma = rawKhayData.ma,

520 │ maVatTu = rawKhayData.maVatTu,

521 │ codeVatTu = rawKhayData.codeVatTu,

522 │ tenVatTu = rawKhayData.tenVatTu,

523 │ tenKhoNguyenLieu = rawKhayData.tenKhoNguyenLieu,

524 │ tenKhoTieuHao = rawKhayData.tenKhoTieuHao,

525 │ tenPhongBan = rawKhayData.tenPhongBan,

526 │ dinhMucKhay = rawKhayData.dinhMucKhay,

527 │ moTa = rawKhayData.moTa,

528 │ daNhap = rawKhayData.daNhap,

529 │ khayTreo = rawKhayData.khayTreo,

530 │ status = rawKhayData.status,

531 │ maNhapKhay = rawKhayData.maNhapKhay,

532 │ ngayTao = rawKhayData.ngayTao?.ToString("dd/MM/yyyy") ?? "",

533 │ tenChiNhanh = rawKhayData.tenChiNhanh,

534 │ tenDvt = rawKhayData.tenDvt,

535 │ };

536 │

537 │ return Json(new ResponseModelSuccess("OK", khayData));

538 │ }

539 │ catch (Exception ex)

540 │ {

541 │ return Json(new ResponseModel(QLBHConst.Error, $"Lỗi khi lấy thông tin khay: {ex.Message}"));

542 │ }

543 │ }

544 │

545 │ }

546 │ }

────────────────────────────────────────────────────────────

**📄 KhayNlBep.cs (CS, 15 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class KhayNlBep

7 │ {

8 │ public string? MãHàngHoá { get; set; }

9 │

10 │ public string? TênHàngHoá { get; set; }

11 │

12 │ public string? Đvt { get; set; }

13 │

14 │ public double? Khây { get; set; }

15 │ }

────────────────────────────────────────────────────────────

**📄 KhayThChti.cs (CS, 15 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class KhayThChti

7 │ {

8 │ public string? MãHàngHoá { get; set; }

9 │

10 │ public string? TênHàngHoá { get; set; }

11 │

12 │ public string? Đvt { get; set; }

13 │

14 │ public double? Khây { get; set; }

15 │ }

────────────────────────────────────────────────────────────

**📄 KhayVtBep.cs (CS, 15 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class KhayVtBep

7 │ {

8 │ public string? MãHàngHoá { get; set; }

9 │

10 │ public string? TênHàngHoá { get; set; }

11 │

12 │ public string? Đvt { get; set; }

13 │

14 │ public double? Khây { get; set; }

15 │ }

────────────────────────────────────────────────────────────

**📄 KhayVtCafe.cs (CS, 15 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class KhayVtCafe

7 │ {

8 │ public string? MãHàngHoá { get; set; }

9 │

10 │ public string? TênHàngHoá { get; set; }

11 │

12 │ public string? Đvt { get; set; }

13 │

14 │ public double? Khây { get; set; }

15 │ }

────────────────────────────────────────────────────────────

**📄 KhoBep.cs (CS, 25 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class KhoBep

7 │ {

8 │ public double? Stt { get; set; }

9 │

10 │ public string? MãVạch { get; set; }

11 │

12 │ public string? MãHàngHoá { get; set; }

13 │

14 │ public string? TênHàngHoá { get; set; }

15 │

16 │ public string? Đvt { get; set; }

17 │

18 │ public double? GiáVốnChưaVat { get; set; }

19 │

20 │ public double? SlTồn { get; set; }

21 │

22 │ public double? ThuếVat { get; set; }

23 │

24 │ public string? HạnSửDụngDdMmYyyy { get; set; }

25 │ }

────────────────────────────────────────────────────────────

**📄 KhoVtthBep.cs (CS, 25 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class KhoVtthBep

7 │ {

8 │ public double? Stt { get; set; }

9 │

10 │ public string? MãVạch { get; set; }

11 │

12 │ public string? MãHàngHoá { get; set; }

13 │

14 │ public string? TênHàngHoá { get; set; }

15 │

16 │ public string? Đvt { get; set; }

17 │

18 │ public double? GiáVốnChưaVat { get; set; }

19 │

20 │ public double? SlTồn { get; set; }

21 │

22 │ public double? ThuếVat { get; set; }

23 │

24 │ public string? HạnSửDụngDdMmYyyy { get; set; }

25 │ }

────────────────────────────────────────────────────────────

**📄 KiemKeController.cs (CS, 843 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using Microsoft.EntityFrameworkCore;

3 │ using QuanLyBanHang.Common;

4 │ using QuanLyBanHang.Common.Attributes;

5 │ using QuanLyBanHang.Models.APIModels;

6 │ using QuanLyBanHang.Models.ExModel.CAT;

7 │ using QuanLyBanHang.Models.ExModel.OPS;

8 │ using QuanLyBanHang.Models.ExModel.StoredProcedureModel;

9 │ using QuanLyBanHang.Models.ExModel.SYS;

10 │ using QuanLyBanHang.Models.QLBHEntity;

11 │ using System.Data;

12 │

13 │ namespace QuanLyBanHang.Controllers.OPS

14 │ {

15 │ public class KiemKeController : BaseController

16 │ {

17 │ public class HangHoaTheoKho

18 │ {

19 │ public int Ma { get; set; }

20 │ public string? MaKT { get; set; }

21 │ public string? MaVach { get; set; }

22 │ public int? MaLopHang { get; set; }

23 │ public int? MaChiNhanh { get; set; }

24 │ public string? TenLopHang { get; set; }

25 │ public string? TenHangHoa { get; set; }

26 │ public int? MaDvt { get; set; }

27 │ public string? TenDVT { get; set; }

28 │ public decimal? GiaBan { get; set; }

29 │ public decimal? GiaVon { get; set; }

30 │ public string? HinhAnh { get; set; }

31 │ public string? GhiChu { get; set; }

32 │ public bool? HetHang { get; set; }

33 │

34 │ public decimal SLTon { get; set; }

35 │ }

36 │ private readonly IWebHostEnvironment \_hostingEnvironment;

37 │

38 │ public KiemKeController(QuanLyBanHangContext dbcontext, MemoryCacheManager memoryCache, IWebHostEnvironment host...

39 │ {

40 │ m\_MenuID = MenuID.opsKiemKe.GetEnumGuid();

41 │ \_hostingEnvironment = hostingEnvironment;

42 │ }

43 │

44 │ [TokenUserCheckHTTP]

45 │ public IActionResult Index(string MaKho)

46 │ {

47 │ ViewData["ListUser"] = new NguoiDungModel(dbcontext).GetData();

48 │ ViewData["ListKho"] = new DanhMucKhoModel(dbcontext).GetDataByUser(objToken.userid, objToken.branch);

49 │

50 │

51 │ return View();

52 │ }

53 │

54 │ [TokenUserCheckHTTP(Permission.Add)]

55 │ public ActionResult Add()

56 │ {

57 │ ViewData["ListKho"] = new DanhMucKhoModel(dbcontext).GetDataByUser(objToken.userid, objToken.branch);

58 │ ViewData["ListHangHoa"] = new TblHangHoaModel(dbcontext)

59 │ .LayDanhSachTblHangHoa(objToken.branch)

60 │ .Where(t => !(t.HetHang ?? false))

61 │ .ToList();

62 │

63 │ return View("Form");

64 │ }

65 │ [TokenUserCheckHTTP(Permission.Edit)]

66 │ public ActionResult Edit(string Ma)

67 │ {

68 │ var kk = new BienBanKiemKeModel(dbcontext).GetItem(Cast.ToInt(Ma.Decrypt()));

69 │

70 │ if (kk == null)

71 │ {

72 │ return View("../PageNotFound/Index");

73 │ }

74 │ ViewData["item"] = kk;

75 │ ViewData["ListChiTiet"] = new ChiTietKiemKeModel(dbcontext).GetListChiTietKiemKe(kk.Ma, objToken.branch);

76 │ ViewData["ListKho"] = new DanhMucKhoModel(dbcontext).GetDataByUser(objToken.userid, objToken.branch);

77 │

78 │ return View("Form");

79 │ }

80 │ [TokenUserCheckHTTP]

81 │ public ActionResult In(string Ma)

82 │ {

83 │ Ma = Ma.Decrypt();

84 │ var nh = new BienBanKiemKeModel(dbcontext).GetItem(Cast.ToInt(Ma));

85 │

86 │ List<VChiTietKiemKe> ct = new ChiTietKiemKeModel(dbcontext).GetListChiTietKiemKe(nh == null ? 0 : nh.Ma, obj...

87 │ ViewData["Item"] = nh ?? new StockKiemKe();

88 │ ViewData["User"] = new NguoiDungModel(dbcontext).GetItem(nh.UserCreated);

89 │ ViewData["ListChiTiet"] = ct ?? new List<VChiTietKiemKe>();

90 │ ViewData["Kho"] = new DanhMucKhoModel(dbcontext).GetItem(nh == null ? 0 : (nh.MaKho ?? 0));

91 │ return View("In");

92 │

93 │ }

94 │ public List<HangHoaTheoKho> GetListHangHoaTheoKhoDeKiemKe(int MaKho, Guid MaChiNhanh, string NamThang)

95 │ {

96 │ var result = (

97 │ from hh in dbcontext.TblHangHoa.AsNoTracking().Where(x => x.MaChiNhanh == MaChiNhanh)

98 │ join lh in dbcontext.TblLopHangHoa.AsNoTracking().Where(x => x.MaChiNhanh == MaChiNhanh) on hh.MaLopHang...

99 │ join dvt in dbcontext.TblDonViTinh.AsNoTracking() on hh.MaDvt equals dvt.Ma

100 │ join tk in dbcontext.StockTonKhoBanHang.AsNoTracking().Where(t => t.NamThang == NamThang && t.MaChiNhanh...

101 │

102 │ where !(hh.Deleted ?? false && tk.SltonCuoiKy > 0)

103 │

104 │ select new HangHoaTheoKho

105 │ {

106 │ Ma = hh.Ma,

107 │ MaKT = hh.MaKt,

108 │ MaVach = hh.MaVach,

109 │ MaLopHang = hh.MaLopHang,

110 │ TenLopHang = lh.Ten,

111 │ TenHangHoa = hh.Ten,

112 │ MaDvt = hh.MaDvt,

113 │ TenDVT = dvt.Ten,

114 │ GiaBan = hh.GiaBan,

115 │ GiaVon = hh.GiaVon,

116 │ HinhAnh = hh.HinhAnh,

117 │ GhiChu = hh.GhiChu,

118 │ HetHang = hh.HetHang,

119 │ SLTon = tk.SltonCuoiKy ?? 0

120 │ }).ToList();

121 │ return result;

122 │ }

123 │ [HttpPost]

124 │ [TokenUserCheckHTTP]

125 │ public ActionResult getListHangHoaKiemKe(int MaKho, string NamThang)

126 │ {

127 │ var listData = GetListHangHoaTheoKhoDeKiemKe(MaKho, objToken.branch, NamThang);

128 │ return Json(new ResponseModelSuccess("", listData));

129 │ }

130 │ [HttpPost]

131 │ [TokenUserCheckHTTP]

132 │ public ActionResult getListDVT()

133 │ {

134 │ var listData = new DonViTinhModel(dbcontext).GetData();

135 │ return Json(new ResponseModelSuccess("", listData));

136 │ }

137 │ #region "App API"

138 │ [HttpPost]

139 │ [TokenUserCheckHTTP]

140 │ public ActionResult Update(string Ma, string MaKho, string NgayKiemKe, string NguoiKiemKe, string GhiChu, string...

141 │ {

142 │ int \_Ma = Cast.ToInt(Ma.Decrypt());

143 │

144 │ if (ListChiTiet == null)

145 │ {

146 │ ListChiTiet = new List<StockChiTietKiemKe>();

147 │ }

148 │ if (Cast.ToInt(MaKho) == 0)

149 │ {

150 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng chọn kho"));

151 │ }

152 │ int i = 1;

153 │ foreach (StockChiTietKiemKe ct in ListChiTiet)

154 │ {

155 │ if (Cast.ToInt(ct.MaHangHoa) == 0)

156 │ {

157 │ return Json(new ResponseModel(QLBHConst.Error, "Hàng hóa không tồn tại. Chi tiết số " + i...

158 │ }

159 │ if (Cast.ToDecimal(ct.SoLuong) <= 0)

160 │ {

161 │ return Json(new ResponseModel(QLBHConst.Error, "Số lượng không hợp lệ. Chi tiết số " +...

162 │ }

163 │ i++;

164 │ }

165 │

166 │ if (NguoiKiemKe == "")

167 │ {

168 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng nhập tên người kiểm kê"));

169 │ }

170 │ DateTime dtpNgayKiemKe;

171 │ if (NgayKiemKe == "")

172 │ {

173 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng nhập ngày kiểm kê"));

174 │ }

175 │ else

176 │ {

177 │ dtpNgayKiemKe = DateTime.ParseExact(NgayKiemKe, "dd/MM/yyyy", null);

178 │ }

179 │ int Nam, Thang;

180 │ if (KyKiemKe == "")

181 │ {

182 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng nhập kỳ kiểm kê"));

183 │ }

184 │ else

185 │ {

186 │ try

187 │ {

188 │ Thang = Cast.ToInt(KyKiemKe.Split('/')[0]);

189 │ Nam = Cast.ToInt(KyKiemKe.Split('/')[1]);

190 │ if (Thang < 1 || Thang > 12 || Nam < 2000)

191 │ {

192 │ return Json(new ResponseModel(QLBHConst.Error, "Kỳ kiểm kê không hợp lệ"));

193 │ }

194 │ }

195 │ catch

196 │ {

197 │ return Json(new ResponseModel(QLBHConst.Error, "Kỳ kiểm kê không hợp lệ"));

198 │ }

199 │ }

200 │

201 │ List<VPermission> ListPermission = GetVPermissions();

202 │

203 │ if (\_Ma == 0 && ListPermission.Any(t => t.PermissionId == Permission.Add.GetEnumGuid()))//Add

204 │ {

205 │ var res = new BienBanKiemKeModel(dbcontext).Add(Cast.ToInt(MaKho), dtpNgayKiemKe, NguoiKiemKe, GhiChu, o...

206 │

207 │ return Json(new ResponseModel(res.Result, res.ResultText));

208 │ }

209 │ else if (\_Ma != 0 && ListPermission.Any(t => t.PermissionId == Permission.Edit.GetEnumGuid()))//Edit

210 │ {

211 │ var res = new BienBanKiemKeModel(dbcontext).Edit(\_Ma, Cast.ToInt(MaKho), dtpNgayKiemKe, NguoiKiemKe, Ghi...

212 │

213 │ return Json(new ResponseModel(res.Result, res.ResultText));

214 │ }

215 │ else

216 │ {

217 │ return Json(QLBHConst.ResponseNoPermission);

218 │ }

219 │ }

220 │

221 │ [HttpPost]

222 │ [TokenUserCheckHTTP(Permission.ChotTonKho)]

223 │ public ActionResult ChotTonKho(string MaKiemKe)

224 │ {

225 │ var res = new BienBanKiemKeModel(dbcontext).ChotKiemKe(Cast.ToInt(MaKiemKe.Decrypt()), objToken.userid);

226 │ return Json(new ResponseModel(res.Result, res.ResultText));

227 │ }

228 │

229 │ [HttpPost]

230 │ [TokenUserCheckHTTP(Permission.Delete)]

231 │ public ActionResult Delete(string Ma)

232 │ {

233 │ Ma = Ma.Decrypt();

234 │ int res = new BienBanKiemKeModel(dbcontext).Delete(Cast.ToInt(Ma), objToken.userid);

235 │ if (res == -2)

236 │ {

237 │ return Json(QLBHConst.ResponseNotExist);

238 │ }

239 │ else if (res == 1)

240 │ {

241 │ return Json(QLBHConst.ResponseDeleteSuccessfully);

242 │ }

243 │ else

244 │ {

245 │ return Json(QLBHConst.ResponseHaveError);

246 │ }

247 │ }

248 │

249 │ [HttpPost]

250 │ [TokenUserCheckHTTP]

251 │ public ActionResult getListData(string TuNgay, string DenNgay, string MaKho)

252 │ {

253 │ // Chuyển đổi ngày tháng

254 │ DateTime TuNgayConvert = TuNgay.ToDateTime2(DateTime.Now).Value.Date;

255 │ DateTime DenNgayConvert = DenNgay.ToDateTime2(DateTime.Now).Value.Date;

256 │ int maKhoInt = Cast.ToInt(MaKho);

257 │

258 │ // Sử dụng AsNoTracking() để tăng hiệu suất truy vấn chỉ đọc

259 │ var ListData = new BienBanKiemKeModel(dbcontext).GetData(objToken.branch, TuNgayConvert, DenNgayConvert, maK...

260 │

261 │ // Lấy chỉ những ID cần thiết

262 │ var nguoiTaoIds = ListData.Where(t => t.UserCreated.HasValue).Select(t => t.UserCreated.Value).Distinct().To...

263 │ var nguoiSuaIds = ListData.Where(t => t.UserModified.HasValue).Select(t => t.UserModified.Value).Distinct()....

264 │ var maKhoIds = ListData.Where(t => t.MaKho.HasValue).Select(t => t.MaKho.Value).Distinct().ToList();

265 │

266 │ // Kết hợp danh sách ID người dùng

267 │ var allUserIds = nguoiTaoIds.Union(nguoiSuaIds).ToList();

268 │

269 │ // Truy vấn dữ liệu liên quan một lần và lưu vào Dictionary để tìm kiếm nhanh

270 │ var userDict = new NguoiDungModel(dbcontext).GetDataByIds(allUserIds)

271 │ .ToDictionary(u => u.Ma, u => u);

272 │

273 │ var khoDict = new DanhMucKhoModel(dbcontext).GetDataByIds(maKhoIds, objToken.branch)

274 │ .ToDictionary(k => k.Ma, k => k);

275 │

276 │ // Tạo kết quả cuối cùng

277 │ var result = ListData.Select(t => new

278 │ {

279 │ ma = t.Ma.Encrypt() + "",

280 │ makho = t.MaKho + "",

281 │ kykiemke = t.NgayKiemKe.Value.ToString("dd/MM/yyyy"),

282 │ tenkho = t.MaKho.HasValue && khoDict.TryGetValue(t.MaKho.Value, out var kho) ? kho.Ten : "",

283 │ ngaykiemke = t.NgayKiemKe.Value.ToString("dd/MM/yyyy"),

284 │ sophieu = t.SoKiemKe + "",

285 │ ghichu = t.DienGiai + "",

286 │ nguoikiemke = t.NguoiKiemKe + "",

287 │ nguoitao = t.UserCreated.HasValue && userDict.TryGetValue(t.UserCreated.Value, out var nguoiTao) ? nguoi...

288 │ ngaytao = t.NgayLap.HasValue ? t.NgayLap.Value.ToString("dd/MM/yyyy") : "",

289 │ nguoisua = t.UserModified.HasValue && userDict.TryGetValue(t.UserModified.Value, out var nguoiSua) ? ngu...

290 │ ngaysua = t.ModifiedDate.HasValue ? t.ModifiedDate.Value.ToString("dd/MM/yyyy") : "",

291 │ thang = t.Thang,

292 │ nam = t.Nam

293 │ }).ToList();

294 │

295 │ return Json(new ResponseModel(QLBHConst.OK, "", result));

296 │

297 │ }

298 │

299 │ [HttpPost]

300 │ [TokenUserCheckHTTP]

301 │ public ActionResult getBienBanKiemKeTheoMa(string Ma)

302 │ {

303 │ try

304 │ {

305 │ // Giải mã và chuyển đổi mã kiểm kê

306 │ int maKiemKe = Cast.ToInt(Ma.Decrypt());

307 │

308 │ // Lấy thông tin biên bản kiểm kê

309 │ var kk = new BienBanKiemKeModel(dbcontext).GetItem(maKiemKe);

310 │

311 │ if (kk == null)

312 │ {

313 │ return Json(QLBHConst.ResponseNotExist);

314 │ }

315 │

316 │ // Lấy thông tin người tạo và người sửa (nếu có)

317 │ Guid? nguoiTaoId = kk.UserCreated;

318 │ Guid? nguoiSuaId = kk.UserModified;

319 │

320 │ // Lấy thông tin kho

321 │ int? maKho = kk.MaKho;

322 │

323 │ // Chỉ lấy thông tin người dùng và kho cần thiết

324 │ var nguoiTao = nguoiTaoId.HasValue

325 │ ? new NguoiDungModel(dbcontext).GetByCreatedId(nguoiTaoId.Value)

326 │ : null;

327 │

328 │ var nguoiSua = nguoiSuaId.HasValue

329 │ ? new NguoiDungModel(dbcontext).GetByModifiedId(nguoiSuaId.Value)

330 │ : null;

331 │

332 │ var kho = maKho.HasValue

333 │ ? new DanhMucKhoModel(dbcontext).GetItem(maKho.Value)

334 │ : null;

335 │

336 │ // Tạo kết quả trả về

337 │ var result = new

338 │ {

339 │ ma = kk.Ma.Encrypt(),

340 │ makho = kk.MaKho.ToString(),

341 │ tenkho = kho?.Ten ?? "",

342 │ ngaykiemke = kk.NgayKiemKe.Value.ToString("dd/MM/yyyy"),

343 │ sophieu = kk.SoKiemKe ?? "",

344 │ ghichu = kk.DienGiai ?? "",

345 │ nguoikiemke = kk.NguoiKiemKe ?? "",

346 │ nguoitao = nguoiTao?.FullName ?? "",

347 │ ngaytao = kk.CreatedDate.HasValue ? kk.CreatedDate.Value.ToString("dd/MM/yyyy") : "",

348 │ nguoisua = nguoiSua?.FullName ?? "",

349 │ ngaysua = kk.ModifiedDate.HasValue ? kk.ModifiedDate.Value.ToString("dd/MM/yyyy") : ""

350 │ };

351 │

352 │ return Json(new ResponseModel(QLBHConst.OK, "", result));

353 │ }

354 │ catch (Exception ex)

355 │ {

356 │ // Xử lý ngoại lệ

357 │ return Json(new ResponseModel(QLBHConst.Error, ex.Message, null));

358 │ }

359 │ }

360 │

361 │ [HttpPost]

362 │ [TokenUserCheckHTTP]

363 │ public ActionResult getListChiTietBienBanKiemKe(string MaKiemKe)

364 │ {

365 │ List<VChiTietKiemKe> ListData = new ChiTietKiemKeModel(dbcontext).GetListChiTietKiemKe(Cast.ToInt(MaKiemKe.D...

366 │

367 │ return Json(new ResponseModel(QLBHConst.OK, "", ListData.Select(t => new

368 │ {

369 │ ma = t.Ma.Encrypt() + "",

370 │ makiemke = t.MaKiemKe + "",

371 │ mahanghoa = t.MaHangHoa + "",

372 │ tenhanghoa = t.TenHang + "",

373 │ mavach = t.MaVach + "",

374 │ soluongkiemke = t.SoLuongKiemKe + "",

375 │ tendonvitinh = t.TenDvt + "",

376 │ tenlop = t.TenLop + "",

377 │ ghichu = t.DienGiai + ""

378 │ }).ToList()));

379 │ }

380 │

381 │ [HttpPost]

382 │ [TokenUserCheckHTTP]

383 │ public ActionResult getListHangHoa()

384 │ {

385 │ var list = new TblHangHoaModel(dbcontext).LayDanhSachHangHoa().ToList();

386 │ return Json(new ResponseModel(QLBHConst.OK, "", new TblHangHoaModel(dbcontext).LayDanhSachHangHoa().Select(t...

387 │ {

388 │ mavach = t.MaVach + "",

389 │ mahanghoa = t.Ma + "",

390 │ tenhanghoa = t.TenHangHoa + "",

391 │ madonvitinh = t.MaDvt + "",

392 │ tendonvitinh = t.TenDvt + ""

393 │ }).ToList()));

394 │ }

395 │

396 │ [HttpPost]

397 │ [TokenUserCheckHTTP]

398 │ public ActionResult getHangHoaTheoMaVach(string MaVach)

399 │ {

400 │ VHangHoa hh = new TblHangHoaModel(dbcontext).GetHangHoaTheoMaVach(MaVach);

401 │ if (hh == null)

402 │ {

403 │ return Json(QLBHConst.ResponseNotExist);

404 │ }

405 │ return Json(new ResponseModel(QLBHConst.OK, "", new { mavach = hh.MaVach + "", mahanghoa = hh.Ma + "", tenha...

406 │ }

407 │

408 │ [HttpPost]

409 │ [TokenUserCheckHTTP]

410 │ public ActionResult getListKhoHang()

411 │ {

412 │

413 │ return Json(new ResponseModel(QLBHConst.OK, "", new DanhMucKhoModel(dbcontext).GetDataByUser(objToken.userid...

414 │ {

415 │ makho = t.Ma + "",

416 │ tenkho = t.Ten + "",

417 │ maybantudong = (t.MayBanHangTuDong ?? false) ? "1" : "0"

418 │ }).ToList()));

419 │ }

420 │ [HttpPost]

421 │ [TokenUserCheckHTTP]

422 │ public ActionResult getKhoHangTheoMa(string MaKho)

423 │ {

424 │ StockDanhMucKho kho = new DanhMucKhoModel(dbcontext).GetItem(Cast.ToInt(MaKho));

425 │ if (kho == null)

426 │ {

427 │ return Json(QLBHConst.ResponseNotExist);

428 │ }

429 │ return Json(new ResponseModel(QLBHConst.OK, "", new { makho = kho.Ma + "", tenkho = kho.Ten + "", maybantudo...

430 │ }

431 │

432 │ [HttpPost]

433 │ [TokenUserCheckHTTP]

434 │ public ActionResult UpdateBienBanKiemKeOnly(string Ma, string MaKho, string Thang, string Nam, string NgayKiemKe...

435 │ {

436 │ int \_Ma = Cast.ToInt(Ma.Decrypt());

437 │ if (Cast.ToInt(MaKho) == 0)

438 │ {

439 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng chọn kho"));

440 │ }

441 │ DateTime dtpNgayKiemKe;

442 │ if (NgayKiemKe == "")

443 │ {

444 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng nhập ngày kiểm kê"));

445 │ }

446 │ else

447 │ {

448 │ dtpNgayKiemKe = DateTime.ParseExact(NgayKiemKe, "dd/MM/yyyy", null);

449 │ }

450 │ int iNam, iThang;

451 │

452 │ try

453 │ {

454 │ iThang = Cast.ToInt(Thang);

455 │ iNam = Cast.ToInt(Nam);

456 │ if (iThang < 1 || iThang > 12 || iNam < 2000)

457 │ {

458 │ return Json(new ResponseModel(QLBHConst.Error, "Kỳ kiểm kê không hợp lệ"));

459 │ }

460 │ }

461 │ catch

462 │ {

463 │ return Json(new ResponseModel(QLBHConst.Error, "Kỳ kiểm kê không hợp lệ"));

464 │ }

465 │

466 │ List<VPermission> ListPermission = GetVPermissions();

467 │

468 │ if (\_Ma == 0 && ListPermission.Any(t => t.PermissionId == Permission.Add.GetEnumGuid()))//Add

469 │ {

470 │

471 │ var res = new BienBanKiemKeModel(dbcontext).AddBienBanOnly(Cast.ToInt(MaKho), dtpNgayKiemKe, NguoiKiemKe...

472 │ return Json(new ResponseModel(res.Result, res.ResultText));

473 │ }

474 │ else if (\_Ma != 0 && ListPermission.Any(t => t.PermissionId == Permission.Edit.GetEnumGuid()))//Edit

475 │ {

476 │ var res = new BienBanKiemKeModel(dbcontext).EditBienBanOnly(\_Ma, Cast.ToInt(MaKho), dtpNgayKiemKe, Nguoi...

477 │ return Json(new ResponseModel(res.Result, res.ResultText));

478 │ }

479 │ else

480 │ {

481 │ return Json(QLBHConst.ResponseNoPermission);

482 │ }

483 │ }

484 │

485 │ [HttpPost]

486 │ [TokenUserCheckHTTP(Permission.Add)]

487 │ public ActionResult AddChiTietKiemKe(string MaKiemKe, string MaHangHoa, string SoLuong, string GhiChu)

488 │ {

489 │ MaKiemKe = MaKiemKe.Decrypt();

490 │ if (Cast.ToInt(MaKiemKe) == 0)

491 │ {

492 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng chọn biên bản kiểm kê"));

493 │ }

494 │ if (Cast.ToInt(MaHangHoa) == 0)

495 │ {

496 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng chọn hàng hóa"));

497 │ }

498 │ if (Cast.ToDecimal(SoLuong) <= 0M)

499 │ {

500 │ return Json(new ResponseModel(QLBHConst.Error, "Số lượng không hợp lệ"));

501 │ }

502 │

503 │ int res = new ChiTietKiemKeModel(dbcontext).Add(Cast.ToInt(MaKiemKe), Cast.ToInt(MaHangHoa), SoLuong.ToInt()...

504 │

505 │ if (res == 1)

506 │ {

507 │ return Json(QLBHConst.ResponseAddSuccessfully);

508 │ }

509 │

510 │ return Json(QLBHConst.ResponseAddFailure);

511 │ }

512 │

513 │ [HttpPost]

514 │ [TokenUserCheckHTTP(Permission.Edit)]

515 │ public ActionResult EditChiTietKiemKe(string MaChiTiet, string SoLuong, string GhiChu)

516 │ {

517 │ MaChiTiet = MaChiTiet.Decrypt();

518 │ if (Cast.ToInt(MaChiTiet) == 0)

519 │ {

520 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng chọn chi tiết cần sửa"));

521 │ }

522 │ if (Cast.ToDecimal(SoLuong) <= 0M)

523 │ {

524 │ return Json(new ResponseModel(QLBHConst.Error, "Số lượng không hợp lệ"));

525 │ }

526 │ int res = new ChiTietKiemKeModel(dbcontext).Edit(Cast.ToInt(MaChiTiet), SoLuong.ToInt(), GhiChu, objToken.us...

527 │

528 │ if (res == 1)

529 │ {

530 │ return Json(QLBHConst.ResponseUpdateSuccessfully);

531 │ }

532 │ else if (res == -1)

533 │ {

534 │ return Json(QLBHConst.ResponseNotExist);

535 │ }

536 │ return Json(QLBHConst.ResponseUpdateFailure);

537 │ }

538 │

539 │ [HttpPost]

540 │ [TokenUserCheckHTTP(Permission.Delete)]

541 │ public ActionResult DeleteChiTietKiemKe(string MaChiTiet)

542 │ {

543 │ MaChiTiet = MaChiTiet.Decrypt();

544 │ if (Cast.ToInt(MaChiTiet) == 0)

545 │ {

546 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng chọn chi tiết cần xóa"));

547 │ }

548 │

549 │ int res = new ChiTietKiemKeModel(dbcontext).Delete(Cast.ToInt(MaChiTiet), objToken.userid);

550 │

551 │ if (res == 1)

552 │ {

553 │ return Json(QLBHConst.ResponseUpdateSuccessfully);

554 │ }

555 │ else if (res == -2)

556 │ {

557 │ return Json(QLBHConst.ResponseNotExist);

558 │ }

559 │ return Json(QLBHConst.ResponseUpdateFailure);

560 │ }

561 │ [HttpPost]

562 │ [TokenUserCheckHTTP(Permission.Delete)]

563 │ public ActionResult UpdateGhiChu(string MaKiemKe, List<StockChiTietTonKho> ListChiTiet)

564 │ {

565 │ MaKiemKe = MaKiemKe.Decrypt();

566 │ var kk = new BienBanKiemKeModel(dbcontext).GetItem(Cast.ToInt(MaKiemKe));

567 │

568 │ if (kk == null)

569 │ {

570 │ return Json(QLBHConst.ResponseNotExist);

571 │ }

572 │ int res = new ChiTietTonKhoModel(dbcontext).UpdateGhiChu(ListChiTiet, objToken.userid);

573 │ if (res > 0)

574 │ {

575 │ return Json(QLBHConst.ResponseUpdateSuccessfully);

576 │ }

577 │

578 │ return Json(QLBHConst.ResponseUpdateFailure);

579 │ }

580 │ #endregion

581 │

582 │ [TokenUserCheckHTTP]

583 │ public ActionResult BaoCao(string TuNgay, string DenNgay, string MaKho, string MaKiemKe)

584 │ {

585 │ MaKiemKe = MaKiemKe.Decrypt();

586 │ int intMaKiemKe = MaKiemKe.ToInt();

587 │

588 │ StockKiemKe kk = new BienBanKiemKeModel(dbcontext).GetItem(intMaKiemKe);

589 │ var result = GetBaoCaoKiemKe(TuNgay, DenNgay, MaKho, MaKiemKe);

590 │ ViewData["item"] = kk;

591 │ ViewData["ListData"] = result;

592 │ ViewData["ListKho"] = new DanhMucKhoModel(dbcontext).GetDataByUser(objToken.userid, objToken.branch);

593 │ return View("BaoCao");

594 │ }

595 │ private List<sp\_BaoCaoKiemKe> GetBaoCaoKiemKe(string TuNgay, string DenNgay, string MaKho, string MaKiemKe)

596 │ {

597 │ int intMaKiemKe = MaKiemKe.ToInt();

598 │ var result = (

599 │ from hh in dbcontext.TblHangHoa.AsNoTracking().Where(x => x.MaChiNhanh == objToken.branch)

600 │ where !(hh.Deleted ?? false)

601 │ join lh in dbcontext.TblLopHangHoa.AsNoTracking().Where(l => !(l.Deleted ?? false))

602 │ on hh.MaLopHang equals lh.Ma into lhJoin

603 │ from lopHang in lhJoin.DefaultIfEmpty()

604 │

605 │ join cttk in dbcontext.StockChiTietTonKho.AsNoTracking().Where(ct => ct.MaKiemKe == intMaKiemKe)

606 │ on hh.Ma equals cttk.MaHangHoa into cttkJoin

607 │ from chiTietTonKho in cttkJoin.DefaultIfEmpty()

608 │

609 │ join ctkk in dbcontext.StockChiTietKiemKe.AsNoTracking().Where(ct => ct.MaKiemKe == intMaKiemKe)

610 │ on hh.Ma equals ctkk.MaHangHoa into ctkkJoin

611 │ from chiTietKiemKe in ctkkJoin.DefaultIfEmpty()

612 │

613 │ where (chiTietTonKho.SoLuongTon ?? 0) != 0 || (chiTietKiemKe.SoLuong ?? 0) != 0

614 │

615 │ select new sp\_BaoCaoKiemKe

616 │ {

617 │ Ma = chiTietTonKho.Ma,

618 │ TenLop = lopHang.Ten,

619 │ MaKT = hh.MaKt,

620 │ MaVach = hh.MaVach,

621 │ TenHang = hh.Ten,

622 │ SoLuongTon = chiTietTonKho.SoLuongTon,

623 │ SoLuongKiemKe = chiTietKiemKe.SoLuong,

624 │ SoLuongChenhLech = (chiTietTonKho.SoLuongTon ?? 0m) - (chiTietKiemKe.SoLuong ?? 0m),

625 │ GhiChu = chiTietTonKho.GhiChu

626 │ }).ToList();

627 │ return result;

628 │ }

629 │ [TokenUserCheckHTTP]

630 │ public ActionResult ExportExcel(string TuNgay, string DenNgay, string MaKho, string MaKiemKe, string FileType)

631 │ {

632 │ DateTime TuNgayConvert = TuNgay.ToDateTime2(DateTime.Now).Value.Date;

633 │ DateTime DenNgayConvert = DenNgay.ToDateTime2(DateTime.Now).Value.Date;

634 │

635 │ //List<VChiTietKiemKe> listData = new ChiTietKiemKeModel(dbcontext).GetListChiTietKiemKe(TuNgayConvert, DenN...

636 │ var listData = (

637 │ from ctkk in dbcontext.StockChiTietKiemKe.AsNoTracking()

638 │ join cttk in dbcontext.StockChiTietTonKho.AsNoTracking()

639 │ on new { ctkk.MaHangHoa, ctkk.MaKiemKe } equals new { cttk.MaHangHoa, cttk.MaKiemKe } into cttkJoin

640 │ from cttk in cttkJoin.DefaultIfEmpty()

641 │

642 │ join hh in dbcontext.TblHangHoa.AsNoTracking()

643 │ on ctkk.MaHangHoa equals hh.Ma

644 │ join dvt in dbcontext.TblDonViTinh.AsNoTracking()

645 │ on hh.MaDvt equals dvt.Ma

646 │ join lop in dbcontext.TblLopHangHoa.AsNoTracking()

647 │ on hh.MaLopHang equals lop.Ma

648 │ join kk in dbcontext.StockKiemKe.AsNoTracking()

649 │ on ctkk.MaKiemKe equals kk.Ma

650 │ join kho in dbcontext.StockDanhMucKho.AsNoTracking()

651 │ on kk.MaKho equals kho.Ma

652 │

653 │ where !(ctkk.Deleted ?? false)

654 │ && !(kk.Deleted ?? false)

655 │ && !(hh.Deleted ?? false)

656 │ && hh.MaChiNhanh == objToken.branch

657 │ //&& ctkk.MaKiemKe == intMaKiemKe

658 │

659 │ select new

660 │ {

661 │ Ma = ctkk.Ma,

662 │ MaKiemKe = ctkk.MaKiemKe,

663 │ MaHangHoa = ctkk.MaHangHoa,

664 │ MaVach = hh.MaVach,

665 │ MaKT = hh.MaKt,

666 │ TenHang = hh.Ten,

667 │ MaChiNhanh = hh.MaChiNhanh,

668 │ SoLuongKiemKe = ctkk.SoLuong ?? 0,

669 │ SoLuongTon = cttk.SoLuongTon ?? 0,

670 │ SoLuongChenhLech = (cttk.SoLuongTon ?? 0) - (ctkk.SoLuong ?? 0),

671 │ TenDVT = dvt.Ten,

672 │ MaDvt = dvt.Ma,

673 │ TenLop = lop.Ten,

674 │ NgayKiemKe = kk.NgayKiemKe,

675 │ MaKho = kk.MaKho,

676 │ TenKho = kho.Ten,

677 │ SoKiemKe = kk.SoKiemKe,

678 │ DienGiai = kk.DienGiai,

679 │ Thang = kk.Thang,

680 │ Nam = kk.Nam,

681 │ GhiChu = ctkk.GhiChu,

682 │ NguoiKiemKe = kk.NguoiKiemKe

683 │ }).Distinct().ToList();

684 │

685 │

686 │ DataTable dtb = new();

687 │ dtb.Columns.Add("STT");

688 │ dtb.Columns.Add("Ngày kiểm kê");

689 │ dtb.Columns.Add("Người kiểm kê");

690 │ dtb.Columns.Add("Kho hàng");

691 │ dtb.Columns.Add("Mã KT");

692 │ dtb.Columns.Add("Mã vạch");

693 │ dtb.Columns.Add("Tên hàng hóa");

694 │ dtb.Columns.Add("ĐVT");

695 │ dtb.Columns.Add("Số lượng kiểm kê", typeof(decimal));

696 │ dtb.Columns.Add("Số lượng tồn", typeof(decimal));

697 │ dtb.Columns.Add("Số lượng chênh lệch", typeof(decimal));

698 │ dtb.Columns.Add("Ghi chú");

699 │

700 │ int index = 0;

701 │

702 │ foreach (var item in listData)

703 │ {

704 │ index++;

705 │ dtb.Rows.Add(new object[]

706 │ {

707 │ index,

708 │ string.Format("'{0:dd/MM/yyyy}", item.NgayKiemKe),

709 │ item.NguoiKiemKe ?? "",

710 │ item.TenKho ?? "",

711 │ item.MaKT ?? "",

712 │ item.MaVach ?? "",

713 │ item.TenHang ?? "",

714 │ item.TenDVT ?? "",

715 │ item.SoLuongKiemKe,

716 │ item.SoLuongTon,

717 │ item.SoLuongChenhLech,

718 │ item.GhiChu ?? "",

719 │ });

720 │

721 │ }

722 │ string fileName = FileType == "Excel" ? "DanhSachKiemKe.xlsx" : "DanhSachKiemKe.pdf";

723 │ string contentType = FileType == "Excel" ? "application/ms-excel" : "application/pdf";

724 │

725 │ string fileTemplate = Path.Combine(\_hostingEnvironment.ContentRootPath, "wwwroot", "templates", "DanhSachKie...

726 │ List<ExcelValue> ListGiaTriRiengLe = new()

727 │ {

728 │ new() { Range = $"A2:{ExportToExcel.GetExcelColumnName(dtb.Columns.Count)}2", Value = $"Từ ngày {TuNg...

729 │ };

730 │ int DongBatDau = 5;

731 │ Stream x = ExportToExcel.FromTemplate(fileTemplate, FileType, ListGiaTriRiengLe, dtb, true, DongBatDau, "A",...

732 │ return new FileStreamResult(x, contentType)

733 │ {

734 │ FileDownloadName = fileName

735 │ };

736 │ }

737 │

738 │ [HttpPost]

739 │ public JsonResult CheckBaoCao(string MaKiemKe, string TuNgay, string DenNgay, string MaKho)

740 │ {

741 │ MaKiemKe = MaKiemKe.Decrypt();

742 │ int id = MaKiemKe.ToInt();

743 │

744 │ var result = dbcontext.StockChiTietTonKho

745 │ .Where(x => x.MaKiemKe == id && (x.SoLuongTon ?? 0) != 0)

746 │ .Any();

747 │

748 │ if (!result)

749 │ return Json(new ResponseModel(QLBHConst.Error, "Cần lấy số lượng tồn kho trước khi xem chi...

750 │

751 │ return Json(new ResponseModel(QLBHConst.OK, "Hợp lệ"));

752 │ }

753 │

754 │ [HttpPost]

755 │ [TokenUserCheckHTTP]

756 │ public ActionResult BaoCaoApp(string MaKiemKe)

757 │ {

758 │ MaKiemKe = MaKiemKe.Decrypt();

759 │ int intMaKiemKe = MaKiemKe.ToInt();

760 │

761 │ StockKiemKe kk = new BienBanKiemKeModel(dbcontext).GetItem(intMaKiemKe);

762 │

763 │ if (kk == null)

764 │ {

765 │ return Json(new ResponseModel(QLBHConst.Error, "Không tìm thấy biên bản kiểm kê"));

766 │ }

767 │

768 │ // Truy vấn danh sách hàng hóa và kiểm kê

769 │ var listData = (

770 │ from hh in dbcontext.TblHangHoa.AsNoTracking()

771 │ where !(hh.Deleted ?? false)

772 │ join lh in dbcontext.TblLopHangHoa.AsNoTracking().Where(l => !(l.Deleted ?? false))

773 │ on hh.MaLopHang equals lh.Ma into lhJoin

774 │ from lopHang in lhJoin.DefaultIfEmpty()

775 │

776 │ join dvt in dbcontext.TblDonViTinh.AsNoTracking().Where(d => !(d.Deleted ?? false))

777 │ on hh.MaDvt equals dvt.Ma into dvtJoin

778 │ from donViTinh in dvtJoin.DefaultIfEmpty()

779 │

780 │ join cttk in dbcontext.StockChiTietTonKho.AsNoTracking().Where(ct => ct.MaKiemKe == intMaKiemKe)

781 │ on hh.Ma equals cttk.MaHangHoa into cttkJoin

782 │ from chiTietTonKho in cttkJoin.DefaultIfEmpty()

783 │

784 │ join ctkk in dbcontext.StockChiTietKiemKe.AsNoTracking().Where(ct => ct.MaKiemKe == intMaKiemKe)

785 │ on hh.Ma equals ctkk.MaHangHoa into ctkkJoin

786 │ from chiTietKiemKe in ctkkJoin.DefaultIfEmpty()

787 │

788 │ where (chiTietTonKho.SoLuongTon ?? 0) != 0 || (chiTietKiemKe.SoLuong ?? 0) != 0

789 │

790 │ select new sp\_BaoCaoKiemKe

791 │ {

792 │ Ma = chiTietTonKho.Ma,

793 │ TenLop = lopHang.Ten,

794 │ MaKT = hh.MaKt,

795 │ MaVach = hh.MaVach,

796 │ TenHang = hh.Ten,

797 │ SoLuongTon = chiTietTonKho.SoLuongTon,

798 │ SoLuongKiemKe = chiTietKiemKe.SoLuong,

799 │ SoLuongChenhLech = (chiTietTonKho.SoLuongTon ?? 0m) - (chiTietKiemKe.SoLuong ?? 0m),

800 │ GhiChu = chiTietTonKho.GhiChu

801 │ }

802 │ ).ToList();

803 │

804 │ // Kiểm tra xem biên bản đã chốt tồn kho hay chưa

805 │ bool daLaySoLieu = dbcontext.StockChiTietTonKho

806 │ .Where(x => x.MaKiemKe == intMaKiemKe)

807 │ .Any();

808 │

809 │ if (!daLaySoLieu)

810 │ {

811 │ return Json(new ResponseModel(QLBHConst.Error, "Cần lấy số lượng tồn kho trước khi xem bá...

812 │ }

813 │

814 │ // Trả về kết quả JSON

815 │ return Json(new ResponseModel(QLBHConst.OK, "", listData.Select(t => new

816 │ {

817 │ ma = t.Ma,

818 │ tenlop = t.TenLop + "",

819 │ makt = t.MaKT + "",

820 │ mavach = t.MaVach + "",

821 │ tenhang = t.TenHang + "",

822 │ soluongton = t.SoLuongTon ?? 0,

823 │ soluongkiemke = t.SoLuongKiemKe ?? 0,

824 │ chenhLech = t.SoLuongChenhLech ?? 0,

825 │ ghichu = t.GhiChu + ""

826 │ }).ToList()));

827 │ }

828 │

829 │ [HttpPost]

830 │ [TokenUserCheckHTTP]

831 │ public ActionResult getDanhSachHangHoaForApp()

832 │ {

833 │ var listHangHoa = new TblHangHoaModel(dbcontext)

834 │ .LayDanhSachTblHangHoa(objToken.branch)

835 │ .Where(t => !(t.HetHang ?? false))

836 │ .ToList();

837 │

838 │ return Json(new ResponseModel(QLBHConst.OK, "", listHangHoa));

839 │ }

840 │ }

841 │

842 │

843 │ }

────────────────────────────────────────────────────────────

**📄 KiemTraVoucherController.cs (CS, 55 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using QuanLyBanHang.Common;

3 │ using QuanLyBanHang.Common.Attributes;

4 │ using QuanLyBanHang.Models.APIModels;

5 │ using QuanLyBanHang.Models.QLBHEntity;

6 │ using QuanLyBanHang.Models.VCEntity;

7 │

8 │ namespace QuanLyBanHang.Controllers.OPS

9 │ {

10 │ public class KiemTraVoucherController : BaseController

11 │ {

12 │ private QuanLyVoucherContext vcContext { get; set; }

13 │

14 │ public KiemTraVoucherController(QuanLyBanHangContext dbcontext, MemoryCacheManager memoryCache, QuanLyVoucherCon...

15 │ {

16 │ m\_MenuID = MenuID.opsKiemTraVoucher.GetEnumGuid();

17 │ vcContext = voucherContext;

18 │ }

19 │

20 │ [TokenUserCheckHTTP]

21 │ public IActionResult Index(string MaVach)

22 │ {

23 │ ViewData["MaVach"] = MaVach;

24 │ return View();

25 │ }

26 │

27 │ [TokenUserCheckHTTP]

28 │ public ActionResult getThongTinVoucher(string MaVach)

29 │ {

30 │ MaVach = MaVach ?? "";

31 │ if (string.IsNullOrWhiteSpace(MaVach))

32 │ {

33 │ return Json(new ResponseModel(QLBHConst.Error, "Mã vạch không hợp lệ"));

34 │ }

35 │ var bh = vcContext.TblBanHang.FirstOrDefault(t => t.MaVach == MaVach);

36 │ if (bh == null)

37 │ {

38 │ return Json(new ResponseModel(QLBHConst.Error, "Không tìm thấy hóa đơn theo mã vạch"));

39 │ }

40 │ var vc = vcContext.TblVoucher.FirstOrDefault(t => t.Ma == bh.MaVoucher);

41 │ var kh = vcContext.TblKhachHang.FirstOrDefault(t => t.Ma == bh.MaKh);

42 │ var listTT = vcContext.TblThanhToan.Where(t => t.MaBanHang == bh.Ma).ToList();

43 │

44 │ return Json(new ResponseModel(QLBHConst.OK, "Thành công", new

45 │ {

46 │ tenkh = kh != null ? kh.Ten + "" : "",

47 │ voucher = vc != null ? vc.Ten + "" : "",

48 │ giatri = bh.GiaTri + "",

49 │ sodu = bh.SoDu + "",

50 │ listTT

51 │ }));

52 │

53 │ }

54 │ }

55 │ }

────────────────────────────────────────────────────────────

**📄 KkChiTietKiemKe.cs (CS, 31 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class KkChiTietKiemKe

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int? MaKiemKe { get; set; }

11 │

12 │ public int? MaHangHoa { get; set; }

13 │

14 │ public decimal? SoLuongKiemKe { get; set; }

15 │

16 │ public string? GhiChu { get; set; }

17 │

18 │ public int? NguoiTao { get; set; }

19 │

20 │ public DateTime? NgayTao { get; set; }

21 │

22 │ public int? NguoiSua { get; set; }

23 │

24 │ public DateTime? NgaySua { get; set; }

25 │

26 │ public bool? DaXoa { get; set; }

27 │

28 │ public int? NguoiXoa { get; set; }

29 │

30 │ public DateTime? NgayXoa { get; set; }

31 │ }

────────────────────────────────────────────────────────────

**📄 KkChiTietTonKho.cs (CS, 25 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class KkChiTietTonKho

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int? MaKiemKe { get; set; }

11 │

12 │ public int? MaHangHoa { get; set; }

13 │

14 │ public decimal? SoLuongTon { get; set; }

15 │

16 │ public int? NguoiTao { get; set; }

17 │

18 │ public DateTime? NgayTao { get; set; }

19 │

20 │ public string? GhiChu { get; set; }

21 │

22 │ public int? NguoiSua { get; set; }

23 │

24 │ public DateTime? NgaySua { get; set; }

25 │ }

────────────────────────────────────────────────────────────

**📄 KyKhoModel.cs (CS, 73 lines)**

1 │ ﻿using Microsoft.EntityFrameworkCore;

2 │ using QuanLyBanHang.Common;

3 │ using QuanLyBanHang.Models.QLBHEntity;

4 │

5 │ namespace QuanLyBanHang.Models.ExModel.OPS

6 │ {

7 │ public class KyKhoModel : Base<TblKyKho>

8 │ {

9 │ public KyKhoModel(DbContext dbContext) : base(dbContext)

10 │ {

11 │ }

12 │ QuanLyBanHangContext dbContext

13 │ {

14 │ get { return \_dbContext as QuanLyBanHangContext; }

15 │ set { return; }

16 │ }

17 │ public void UpdatekyKho(int MaKho, int Nam, int Thang, bool Khoa, Guid UserID, Guid MaChiNhanh)

18 │ {

19 │ TblKyKho? drKhoaSo = dbContext.TblKyKho.FirstOrDefault(t => t.MaKho == MaKho && t.Nam == Nam && t.Thang == T...

20 │

21 │ if (drKhoaSo == null)

22 │ {

23 │ TblKyKho itemAdd = new TblKyKho

24 │ {

25 │ MaKho = MaKho,

26 │ Nam = Nam,

27 │ Thang = Thang,

28 │ KhoaSo = Khoa,

29 │ UserCreated = UserID,

30 │ CreatedDate = DateTime.Now

31 │ };

32 │ dbContext.Add(itemAdd);

33 │ dbContext.SaveChanges();

34 │ }

35 │ else

36 │ {

37 │ drKhoaSo.KhoaSo = Khoa;

38 │ drKhoaSo.UserModified = UserID;

39 │ drKhoaSo.ModifiedDate = DateTime.Now;

40 │ dbContext.Update(drKhoaSo);

41 │ dbContext.SaveChanges();

42 │ }

43 │ }

44 │

45 │

46 │ public bool KiemTraKyKhoTruocKhiTinhTonKyHT(int Nam, int Thang, int MaKho)

47 │ {

48 │ DateTime? KyKhoTruoc = null;

49 │ string sThang = "00" + Cast.ToString(Thang);

50 │ sThang = sThang.Substring(sThang.Length - 2, 2);

51 │ DateTime KyKhoHT = DateTime.ParseExact("01/" + sThang + "/" + Nam, "dd/MM/yyyy", null);

52 │ KyKhoTruoc = KyKhoHT.AddMonths(-1);

53 │ TblKyKho? drKhoaSo = dbContext.TblKyKho.FirstOrDefault(t => t.Nam == KyKhoTruoc.Value.Year && t.Thang == KyK...

54 │

55 │ if (drKhoaSo == null)

56 │ return false;

57 │ else if (drKhoaSo != null && !(drKhoaSo.KhoaSo ?? false))

58 │ return false;

59 │ else

60 │ return true;

61 │ }

62 │

63 │ public DateTime LayKyKhoSau(int Nam, int Thang)

64 │ {

65 │ DateTime KyKhoSau = DateTime.Now;

66 │ string sThang = "00" + Cast.ToString(Thang);

67 │ sThang = sThang.Substring(sThang.Length - 2, 2);

68 │ DateTime KyKhoHT = DateTime.ParseExact("01/" + sThang + "/" + Nam, "dd/MM/yyyy", null);

69 │ KyKhoSau = KyKhoHT.AddMonths(1);

70 │ return KyKhoSau;

71 │ }

72 │ }

73 │ }

────────────────────────────────────────────────────────────

**📄 LichSuDangNhapController.cs (CS, 55 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using QuanLyBanHang.Common;

3 │ using QuanLyBanHang.Common.Attributes;

4 │ using QuanLyBanHang.Models;

5 │ using QuanLyBanHang.Models.APIModels;

6 │ using QuanLyBanHang.Models.QLBHEntity;

7 │

8 │ namespace QuanLyBanHang.Controllers.SYS

9 │ {

10 │ public class LichSuDangNhapController : BaseController

11 │ {

12 │

13 │ public LichSuDangNhapController(QuanLyBanHangContext dbcontext, MemoryCacheManager memoryCache) : base(dbcontext...

14 │ {

15 │ m\_MenuID = MenuID.sysLichSuDangNhap.GetEnumGuid();

16 │ }

17 │

18 │ [TokenUserCheckHTTP()]

19 │ public IActionResult Index()

20 │ {

21 │ ViewData["data"] = new Base<SysToken>(dbcontext).Get(t => t.Token != null && t.UserId == objToken!.userid).T...

22 │ return View();

23 │ }

24 │

25 │ [TokenUserCheckHTTP()]

26 │ public IActionResult DeleteAPI(string ma)

27 │ {

28 │ SysToken token = new Base<SysToken>(dbcontext).First(t => t.Ma == ma.GetGuid());

29 │ if (token != null)

30 │ {

31 │ token.Revoked = true;

32 │ new Base<SysToken>(dbcontext).Update(token);

33 │ //\_chatHub.Clients.All.SendAsync("Revoke", token.Token);

34 │ return Json(new ResponseModelSuccess("Thao tác thành công"));

35 │ }

36 │ return Json(new ResponseModelError("Không tìm thấy"));

37 │ }

38 │

39 │ [TokenUserCheckHTTP()]

40 │ public IActionResult UpdateConnectionID(string connectionID, string token)

41 │ {

42 │ var sign = token.Split('.');

43 │ SysToken sysT = new Base<SysToken>(dbcontext).First(t => t.Token != null && sign[1].Contains(t.Token) && !(t...

44 │ if (sysT != null)

45 │ {

46 │ sysT.ConnectionId = connectionID;

47 │ new Base<SysToken>(dbcontext).Update(sysT);

48 │ return Json(new ResponseModelSuccess("Thao tác thành công"));

49 │ }

50 │ return Json(new ResponseModelSuccess("Không tìm thấy"));

51 │ }

52 │

53 │

54 │ }

55 │ }

────────────────────────────────────────────────────────────

**📄 LichSuGia.cs (CS, 29 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class LichSuGia

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string TieuDe { get; set; } = null!;

11 │

12 │ public string LyDo { get; set; } = null!;

13 │

14 │ public bool TrangThai { get; set; }

15 │

16 │ public int NguoiTao { get; set; }

17 │

18 │ public DateTime NgayTao { get; set; }

19 │

20 │ public int? NguoiSua { get; set; }

21 │

22 │ public DateTime? NgaySua { get; set; }

23 │

24 │ public bool? DaXoa { get; set; }

25 │

26 │ public int? NguoiXoa { get; set; }

27 │

28 │ public DateTime? NgayXoa { get; set; }

29 │ }

────────────────────────────────────────────────────────────

**📄 LichSuGiaController.cs (CS, 324 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using QuanLyBanHang.Common;

3 │ using QuanLyBanHang.Common.Attributes;

4 │ using QuanLyBanHang.Models.APIModels;

5 │ using QuanLyBanHang.Models.ExModel.CAT;

6 │ using QuanLyBanHang.Models.ExModel.OPS;

7 │ using QuanLyBanHang.Models.QLBHEntity;

8 │

9 │ namespace QuanLyBanHang.Controllers.OPS

10 │ {

11 │ public class LichSuGiaController : BaseController

12 │ {

13 │ public LichSuGiaController(QuanLyBanHangContext dbcontext, MemoryCacheManager memoryCache) : base(dbcontext, mem...

14 │ {

15 │ m\_MenuID = MenuID.opsLichSuGia.GetEnumGuid();

16 │ }

17 │

18 │ public class cLichSuGia

19 │ {

20 │ public string TieuDe { get; set; }

21 │ public DateTime? NgayChinh { get; set; }

22 │ public string LyDo { get; set; }

23 │ public bool TrangThai { get; set; }

24 │ public int NguoiChinh { get; set; }

25 │ public string Result { get; set; }

26 │ public string ResultText { get; set; }

27 │ }

28 │

29 │ public class cCT\_LichSuGia

30 │ {

31 │ public string Ma { get; set; }

32 │ public int MaHangHoa { get; set; }

33 │ public string GiaBanCu { get; set; }

34 │ public int GiaBanMoi { get; set; }

35 │ public string GhiChu { get; set; }

36 │ public string MaKho { get; set; }

37 │ }

38 │

39 │ [TokenUserCheckHTTP()]

40 │ public IActionResult Index()

41 │ {

42 │ return View();

43 │ }

44 │

45 │ [TokenUserCheckHTTP(Permission.SuaGiaBan)]

46 │ public ActionResult Add()

47 │ {

48 │ ViewData["ListHangHoa"] = new TblHangHoaModel(dbcontext).GetData(objToken.branch);

49 │ ViewData["ListDVT"] = new DonViTinhModel(dbcontext).GetData();

50 │ ViewData["ListKho"] = new DanhMucKhoModel(dbcontext).GetDataByUser(objToken.userid, objToken.branch);

51 │ return View("Form");

52 │ }

53 │

54 │ [TokenUserCheckHTTP(Permission.SuaGiaBan)]

55 │ public ActionResult Edit(string Ma)

56 │ {

57 │ Ma = Ma.Decrypt();

58 │

59 │ TblLichSuGia ls = new LichSuGiaModel(dbcontext).GetItem(Cast.ToInt(Ma));

60 │

61 │ List<TblChiTietLichSuGia> ct = new ChiTietLichSuGiaModel(dbcontext).GetData(ls == null ? 0 : ls.Ma);

62 │

63 │ ViewData["item"] = ls ?? new TblLichSuGia();

64 │ ViewData["ListChiTiet"] = ct ?? new List<TblChiTietLichSuGia>();

65 │ ViewData["ListHangHoa"] = new TblHangHoaModel(dbcontext).GetData(objToken.branch);

66 │ ViewData["ListDVT"] = new DonViTinhModel(dbcontext).GetData();

67 │ ViewData["ListKho"] = new DanhMucKhoModel(dbcontext).GetDataByUser(objToken.userid, objToken.branch);

68 │ return View("Form");

69 │ }

70 │

71 │ [TokenUserCheckHTTP]

72 │ public ActionResult Detail(string Ma)

73 │ {

74 │ Ma = Ma.Decrypt();

75 │ TblLichSuGia ls = new LichSuGiaModel(dbcontext).GetItem(Cast.ToInt(Ma));

76 │ List<TblChiTietLichSuGia> ct = new ChiTietLichSuGiaModel(dbcontext).GetData(ls == null ? 0 : ls.Ma);

77 │ ViewData["item"] = ls ?? new TblLichSuGia();

78 │ ViewData["ListChiTiet"] = ct ?? new List<TblChiTietLichSuGia>();

79 │ ViewData["ListHangHoa"] = new TblHangHoaModel(dbcontext).GetData(objToken.branch);

80 │ ViewData["ListDVT"] = new DonViTinhModel(dbcontext).GetData();

81 │ ViewData["ListKho"] = new DanhMucKhoModel(dbcontext).GetDataByUser(objToken.userid, objToken.branch);

82 │ ViewData["ViewOnly"] = true;

83 │ return View("Form");

84 │ }

85 │

86 │ [HttpPost]

87 │ [TokenUserCheckHTTP]

88 │ public ActionResult DuyetChinhSuaGia(string MaLichSu, string value)

89 │ {

90 │ List<VPermission> ListPermission = GetVPermissions();

91 │

92 │ if (value == "1")

93 │ {

94 │ if (ListPermission.Any(t => t.PermissionId == Permission.DuyetSuaGiaBan.GetEnumGuid()))

95 │ {

96 │ int res = new LichSuGiaModel(dbcontext).DuyetChinhGiaBan(Cast.ToInt(MaLichSu.Decrypt()), objToken.us...

97 │ if (res == 1)

98 │ {

99 │ return Json(new ResponseModel(QLBHConst.OK, "Duyệt thành công"));

100 │ }

101 │ else if (res == -2)

102 │ {

103 │ return Json(QLBHConst.ResponseNotExist);

104 │ }

105 │ else if (res == -3)

106 │ {

107 │ return Json(new ResponseModel(QLBHConst.Error, "Đã được duyệt bởi người khác"));

108 │ }

109 │ else if (res <= -1000)

110 │ {

111 │ return Json(new ResponseModel(QLBHConst.Error, "Lỗi khi cập nhật giá ở dòng " + (res /...

112 │ }

113 │ else

114 │ {

115 │ return Json(QLBHConst.ResponseHaveError);

116 │ }

117 │ }

118 │ else

119 │ {

120 │ return Json(QLBHConst.ResponseNoPermission);

121 │ }

122 │ }

123 │ else

124 │ {

125 │ if (ListPermission.Any(t => t.PermissionId == Permission.BoDuyetSuaGiaBan.GetEnumGuid()))

126 │ {

127 │ int res = new LichSuGiaModel(dbcontext).DuyetChinhGiaBan(Cast.ToInt(MaLichSu.Decrypt()), objToken.us...

128 │ if (res == 1)

129 │ {

130 │ return Json(new ResponseModel(QLBHConst.OK, "Bỏ duyệt thành công"));

131 │ }

132 │ else if (res == -2)

133 │ {

134 │ return Json(QLBHConst.ResponseNotExist);

135 │ }

136 │ else if (res == -3)

137 │ {

138 │ return Json(new ResponseModel(QLBHConst.Error, "Đã bỏ duyệt bởi người khác"));

139 │ }

140 │ else if (res <= -1000)

141 │ {

142 │ return Json(new ResponseModel(QLBHConst.Error, "Lỗi khi cập nhật giá ở dòng " + (res /...

143 │ }

144 │ else

145 │ {

146 │ return Json(QLBHConst.ResponseHaveError);

147 │ }

148 │ }

149 │ else

150 │ {

151 │ return Json(QLBHConst.ResponseNoPermission);

152 │ }

153 │ }

154 │ }

155 │ [HttpPost]

156 │ [TokenUserCheckHTTP(Permission.SuaGiaBan)]

157 │ public ActionResult Update(string Ma, string TieuDe, string LyDo, List<cCT\_LichSuGia> ListChiTiet)

158 │ {

159 │ int \_Ma = Cast.ToInt(Ma.Decrypt());

160 │

161 │ if (ListChiTiet == null)

162 │ {

163 │ ListChiTiet = new List<cCT\_LichSuGia>();

164 │ }

165 │

166 │ if (TieuDe.IsEmpty())

167 │ {

168 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng nhập tiêu đề"));

169 │ }

170 │

171 │ if (LyDo.IsEmpty())

172 │ {

173 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng nhập lý do"));

174 │ }

175 │

176 │ if (\_Ma == 0)//Add

177 │ {

178 │ cLichSuGia res = new LichSuGiaModel(dbcontext).Add(TieuDe, LyDo, objToken.userid, ListChiTiet);

179 │

180 │ return Json(new ResponseModel(res.Result, res.ResultText));

181 │ }

182 │ else

183 │ {

184 │

185 │ cLichSuGia res = new LichSuGiaModel(dbcontext).Edit(\_Ma, TieuDe, LyDo, objToken.userid, ListChiTiet);

186 │

187 │ return Json(new ResponseModel(res.Result, res.ResultText));

188 │ }

189 │

190 │ }

191 │ [HttpPost]

192 │ [TokenUserCheckHTTP(Permission.SuaGiaBan)]

193 │ public ActionResult Delete(string Ma)

194 │ {

195 │ Ma = Ma.Decrypt();

196 │ int res = new LichSuGiaModel(dbcontext).Delete(Cast.ToInt(Ma), objToken.userid);

197 │ if (res == -2)

198 │ {

199 │ return Json(QLBHConst.ResponseNotExist);

200 │ }

201 │ else if (res == -3)

202 │ {

203 │ return Json(new ResponseModel(QLBHConst.Error, "Đã duyệt, không thể xóa"));

204 │ }

205 │ else if (res == 1)

206 │ {

207 │ return Json(QLBHConst.ResponseDeleteSuccessfully);

208 │ }

209 │ else

210 │ {

211 │ return Json(QLBHConst.ResponseHaveError);

212 │ }

213 │ }

214 │ [HttpPost]

215 │ [TokenUserCheckHTTP(Permission.SuaGiaBan)]

216 │ public ActionResult setCTLichSuGia(List<cCT\_LichSuGia> ListData, string MaLichSu)

217 │ {

218 │ MaLichSu = MaLichSu.Decrypt();

219 │ TblLichSuGia ls = new LichSuGiaModel(dbcontext).GetItem(Cast.ToInt(MaLichSu));

220 │ if (ls == null)

221 │ {

222 │ return Json(QLBHConst.ResponseNotExist);

223 │ }

224 │ int result = new ChiTietLichSuGiaModel(dbcontext).SaveListChiTiet(ls.Ma, ListData, objToken.userid, DateTime...

225 │ if (result == 0)

226 │ {

227 │ return Json(new ResponseModel(QLBHConst.OK, "Cập nhật chi tiết phiếu nhập hàng thành công."...

228 │ }

229 │ else

230 │ {

231 │ return Json(new ResponseModel(QLBHConst.Error, "Cập nhật chi tiết phiếu nhập hàng thất bạ...

232 │ }

233 │

234 │ }

235 │ [HttpPost]

236 │ [TokenUserCheckHTTP]

237 │ public ActionResult getListData(string TuNgay, string DenNgay)

238 │ {

239 │ try

240 │ {

241 │ DateTime TuNgayConvert = TuNgay.ToDateTime2(DateTime.Now).Value.Date;

242 │ DateTime DenNgayConvert = DenNgay.ToDateTime2(DateTime.Now).Value.Date;

243 │

244 │ List<TblLichSuGia> lstLichSuGia;

245 │ lstLichSuGia = new LichSuGiaModel(dbcontext).GetData(TuNgayConvert, DenNgayConvert);

246 │

247 │ if (lstLichSuGia == null)

248 │ return Json(new ResponseModel(QLBHConst.Error, QLBHConst.NotExist));

249 │

250 │ return Json(new ResponseModel(QLBHConst.OK, QLBHConst.Successfully, lstLichSuGia.Select(t => new

251 │ {

252 │ ma = (t.Ma + "").Encrypt(),

253 │ tieude = t.TieuDe + "",

254 │ lydo = t.LyDo + "",

255 │ status = t.Status,

256 │ ngaytao = t.CreatedDate.Value.ToString("dd/MM/yyyy")

257 │ }).ToList()));

258 │ }

259 │ catch (Exception ex)

260 │ {

261 │ return Json(new ResponseModel(QLBHConst.Error, QLBHConst.HaveError + ". " + ex.Message));

262 │ }

263 │ }

264 │

265 │ [HttpPost]

266 │ [TokenUserCheckHTTP]

267 │ public ActionResult getListCTLichSuGia(string MaLichSu)

268 │ {

269 │ try

270 │ {

271 │ MaLichSu = MaLichSu.Decrypt();

272 │ List<TblChiTietLichSuGia> lstCTLichSuGia = new ChiTietLichSuGiaModel(dbcontext).GetData(Cast.ToInt(MaLic...

273 │ List<TblHangHoa> lstHH = new TblHangHoaModel(dbcontext).GetData(objToken.branch);

274 │ List<TblDonViTinh> lstDVT = new DonViTinhModel(dbcontext).GetData();

275 │

276 │ var lstTmp = (from h in lstHH

277 │ join dvt in lstDVT on h.MaDvt equals dvt.Ma into dvtJoin

278 │ from dvt in dvtJoin.DefaultIfEmpty()

279 │ select new

280 │ {

281 │ MaHangHoa = h.Ma,

282 │ TenHang = h.Ten,

283 │ DVT = h.MaDvt,

284 │ TenDVT = dvt?.Ten ?? ""

285 │ }).ToList();

286 │

287 │ var result = lstCTLichSuGia.Select(t =>

288 │ {

289 │ var hangHoa = lstTmp.FirstOrDefault(y => y.MaHangHoa == t.MaHangHoa);

290 │ return new

291 │ {

292 │ ma = t.Ma.ToString(),

293 │ mahanghoa = t.MaHangHoa.ToString(),

294 │ tenhanghoa = hangHoa?.TenHang ?? "",

295 │ madonvitinh = hangHoa?.DVT,

296 │ tendonvitinh = hangHoa?.TenDVT ?? "",

297 │ giabancu = t.GiaBanCu?.ToString() ?? "",

298 │ giabanmoi = t.GiaBanMoi?.ToString() ?? "",

299 │ ghichu = t.GhiChu ?? ""

300 │ };

301 │ }).ToList();

302 │ return Json(new ResponseModel(QLBHConst.OK, QLBHConst.Successfully, result));

303 │ }

304 │ catch (Exception ex)

305 │ {

306 │ return Json(new ResponseModel(QLBHConst.Error, QLBHConst.HaveError + ". " + ex.Message));

307 │ }

308 │ }

309 │ [HttpPost]

310 │ [TokenUserCheckHTTP]

311 │ public ActionResult getGiaBanCu(string MaHangHoa, string MaKho)

312 │ {

313 │ try

314 │ {

315 │ var gb = dbcontext.TblGiaBanTheoKho.FirstOrDefault(t => t.MaHangHoa == Cast.ToInt(MaHangHoa) && t.MaKho ...

316 │ return Json(new ResponseModel(QLBHConst.OK, "", new { giabancu = gb == null ? 0 : gb.GiaBan }));

317 │ }

318 │ catch (Exception ex)

319 │ {

320 │ return Json(new ResponseModel(QLBHConst.Error, QLBHConst.HaveError + ". " + ex.Message));

321 │ }

322 │ }

323 │ }

324 │ }

────────────────────────────────────────────────────────────

**📄 LichSuGiaModel.cs (CS, 256 lines)**

1 │ ﻿using Microsoft.EntityFrameworkCore;

2 │ using QuanLyBanHang.Common;

3 │ using QuanLyBanHang.Models.QLBHEntity;

4 │ using static QuanLyBanHang.Controllers.OPS.LichSuGiaController;

5 │

6 │ namespace QuanLyBanHang.Models.ExModel.OPS

7 │ {

8 │ public class LichSuGiaModel : Base<TblLichSuGia>

9 │ {

10 │ public LichSuGiaModel(DbContext dbContext) : base(dbContext)

11 │ {

12 │ }

13 │ QuanLyBanHangContext dbContext

14 │ {

15 │ get { return \_dbContext as QuanLyBanHangContext; }

16 │ set { return; }

17 │ }

18 │

19 │ public TblLichSuGia GetItem(int ma)

20 │ {

21 │ return dbContext.TblLichSuGia.First(t => t.Ma == ma && !(t.Deleted ?? false));

22 │ }

23 │

24 │ public List<TblLichSuGia> GetData(DateTime tuNgay, DateTime denNgay)

25 │ {

26 │ return dbContext.TblLichSuGia

27 │ .Where(t => t.CreatedDate.Value.Date >= tuNgay && t.CreatedDate.Value.Date <= denNgay && !(t.Deleted ?? ...

28 │ .ToList();

29 │ }

30 │

31 │ public cLichSuGia Add(string tieuDe, string lyDo, Guid userID, List<cCT\_LichSuGia> listChiTiet)

32 │ {

33 │ using var transaction = \_dbContext.Database.BeginTransaction();

34 │ try

35 │ {

36 │ var dtNgayTao = DateTime.Now;

37 │ var ls = new TblLichSuGia

38 │ {

39 │ TieuDe = tieuDe,

40 │ LyDo = lyDo,

41 │ UserCreated = userID,

42 │ CreatedDate = dtNgayTao,

43 │ Status = false

44 │ };

45 │ dbContext.TblLichSuGia.Add(ls);

46 │ dbContext.SaveChanges();

47 │

48 │ var rowError = new ChiTietLichSuGiaModel(\_dbContext).SaveListChiTiet(ls.Ma, listChiTiet, userID, dtNgayT...

49 │ if (rowError != 0)

50 │ {

51 │ throw new Exception("Lỗi chi tiết dòng số " + rowError);

52 │ }

53 │

54 │ transaction.Commit();

55 │ return new cLichSuGia { Result = QLBHConst.OK, ResultText = QLBHConst.AddSuccessfully };

56 │ }

57 │ catch (Exception ex)

58 │ {

59 │ transaction.Rollback();

60 │ return new cLichSuGia { Result = QLBHConst.Error, ResultText = "Thêm thất bại. " + ex.Message };

61 │ }

62 │ }

63 │

64 │ public cLichSuGia Edit(int ma, string tieuDe, string lyDo, Guid userId, List<cCT\_LichSuGia> listChiTiet)

65 │ {

66 │ using var transaction = \_dbContext.Database.BeginTransaction();

67 │ int row = -1;

68 │ DateTime dtNgayTao = DateTime.Now;

69 │ try

70 │ {

71 │ // Lấy bản ghi lịch sử giá cần sửa (chưa xóa, chưa duyệt)

72 │ var ls = dbContext.TblLichSuGia

73 │ .FirstOrDefault(t => t.Ma == ma && !(t.Deleted ?? false));

74 │ if (ls == null)

75 │ {

76 │ return new cLichSuGia { Result = QLBHConst.Error, ResultText = QLBHConst.NotExist };

77 │ }

78 │ if (ls.Status)

79 │ {

80 │ return new cLichSuGia { Result = QLBHConst.Error, ResultText = "Đã duyệt, không thể sửa" };

81 │ }

82 │ row = 0;

83 │

84 │ // Cập nhật thông tin

85 │ ls.TieuDe = tieuDe;

86 │ ls.LyDo = lyDo;

87 │ ls.UserModified = userId;

88 │ ls.ModifiedDate = dtNgayTao;

89 │

90 │ dbContext.SaveChanges();

91 │

92 │ // Cập nhật danh sách chi tiết (xóa cũ, thêm/sửa mới)

93 │ int rowError = new ChiTietLichSuGiaModel(\_dbContext).SaveListChiTiet(ls.Ma, listChiTiet, userId, dtNgayT...

94 │ if (rowError != 0)

95 │ {

96 │ row = rowError;

97 │ throw new Exception("");

98 │ }

99 │

100 │ transaction.Commit();

101 │ return new cLichSuGia { Result = QLBHConst.OK, ResultText = QLBHConst.UpdateSuccessfully };

102 │ }

103 │ catch (Exception ex)

104 │ {

105 │ transaction.Rollback();

106 │ if (row == -1)

107 │ return new cLichSuGia { Result = QLBHConst.Error, ResultText = "Cập nhật thất bại." + ex.Mes...

108 │ else if (row == 0)

109 │ return new cLichSuGia { Result = QLBHConst.Error, ResultText = "Cập nhật thất bại. Kiểm tr...

110 │ else

111 │ return new cLichSuGia { Result = QLBHConst.Error, ResultText = "Cập nhật thất bại. Kiểm tr...

112 │ }

113 │ }

114 │ public int Delete(int ma, Guid userId)

115 │ {

116 │ var drItem = dbContext.TblLichSuGia.FirstOrDefault(t => t.Ma == ma && !(t.Deleted ?? false));

117 │ if (drItem == null)

118 │ {

119 │ return -2; // Không tồn tại

120 │ }

121 │

122 │ if (drItem.Status)

123 │ {

124 │ return -3; // Đã duyệt, không thể xóa

125 │ }

126 │

127 │ DateTime dtNgayTao = DateTime.Now;

128 │

129 │ drItem.Deleted = true;

130 │ drItem.DeletedDate = dtNgayTao;

131 │ drItem.UserDeleted = userId;

132 │

133 │ try

134 │ {

135 │ \_dbContext.SaveChanges();

136 │ return 1; // Xóa mềm thành công

137 │ }

138 │ catch

139 │ {

140 │ return 0; // Lỗi khi lưu thay đổi

141 │ }

142 │ }

143 │

144 │ public int DuyetChinhGiaBan(int ma, Guid userId, bool value)

145 │ {

146 │ using var transaction = \_dbContext.Database.BeginTransaction();

147 │ DateTime dtNgayTao = DateTime.Now;

148 │ int row = 0;

149 │ try

150 │ {

151 │ var drItem = dbContext.TblLichSuGia

152 │ .FirstOrDefault(t => t.Ma == ma && !(t.Deleted ?? false));

153 │

154 │ if (drItem == null)

155 │ return -2;

156 │

157 │ if (drItem.Status == value)

158 │ return -3;

159 │

160 │ drItem.Status = value;

161 │ drItem.ModifiedDate = dtNgayTao;

162 │ drItem.UserModified = userId;

163 │ dbContext.SaveChanges();

164 │

165 │ var cts = dbContext.TblChiTietLichSuGia

166 │ .Where(t => t.MaLichSu == ma && !(t.Deleted ?? false))

167 │ .ToList();

168 │

169 │ foreach (var ct in cts)

170 │ {

171 │ row++;

172 │ var maKhos = (ct.MaKho ?? "").Split(';', StringSplitOptions.RemoveEmptyEntries);

173 │ foreach (var makhoStr in maKhos)

174 │ {

175 │ int makho = Cast.ToInt(makhoStr);

176 │ int hangHoaId = Cast.ToInt(ct.MaHangHoa);

177 │ if (makho != 0 && hangHoaId != 0)

178 │ {

179 │ var gbtk = dbContext.TblGiaBanTheoKho

180 │ .FirstOrDefault(t => t.MaKho == makho && t.MaHangHoa == hangHoaId && !(t.Deleted ?? fals...

181 │ if (gbtk != null)

182 │ {

183 │ if (value)

184 │ {

185 │ gbtk.GiaBan = ct.GiaBanMoi;

186 │ gbtk.GhiChu = "Cập nhật từ chỉnh giá";

187 │ gbtk.UserModified = userId;

188 │ gbtk.ModifiedDate = dtNgayTao;

189 │ dbContext.SaveChanges();

190 │ }

191 │ else

192 │ {

193 │ // Lấy giá cũ đã chỉnh gần nhất (đã duyệt)

194 │ var ls = (from a in dbContext.TblChiTietLichSuGia

195 │ join b in dbContext.TblLichSuGia on a.MaLichSu equals b.Ma

196 │ where a.MaHangHoa == hangHoaId

197 │ && !(a.Deleted ?? false)

198 │ && (a.MaKho + "").Contains(";" + makho + ";")

199 │ && !(b.Deleted ?? false)

200 │ && b.Status

201 │ orderby a.Ma descending

202 │ select a).FirstOrDefault();

203 │

204 │ if (ls != null)

205 │ {

206 │ var giabancus = (ls.GiaBanCuTheoKho ?? "").Split(';', StringSplitOptions.RemoveE...

207 │ foreach (var gbc in giabancus)

208 │ {

209 │ var tmp = gbc.Split('\_');

210 │ if (tmp.Length >= 2 && Cast.ToInt(tmp[0]) == makho)

211 │ {

212 │ gbtk.GiaBan = Cast.ToInt(tmp[1]);

213 │ gbtk.GhiChu = "Cập nhật từ chỉnh giá";

214 │ gbtk.UserModified = userId;

215 │ gbtk.ModifiedDate = dtNgayTao;

216 │ \_dbContext.SaveChanges();

217 │ break;

218 │ }

219 │ }

220 │ }

221 │ }

222 │ }

223 │ else

224 │ {

225 │ if (value)

226 │ {

227 │ gbtk = new TblGiaBanTheoKho

228 │ {

229 │ MaHangHoa = hangHoaId,

230 │ MaKho = makho,

231 │ GiaBan = ct.GiaBanMoi,

232 │ GhiChu = "Thêm mới từ chỉnh giá",

233 │ Status = true,

234 │ UserCreated = userId,

235 │ CreatedDate = dtNgayTao

236 │ };

237 │ dbContext.TblGiaBanTheoKho.Add(gbtk);

238 │ dbContext.SaveChanges();

239 │ }

240 │ }

241 │ }

242 │ }

243 │ }

244 │

245 │ transaction.Commit();

246 │ return 1;

247 │ }

248 │ catch

249 │ {

250 │ transaction.Rollback();

251 │ return -2;

252 │ }

253 │ }

254 │ }

255 │

256 │ }

────────────────────────────────────────────────────────────

**📄 LoaiNhapXuatModel.cs (CS, 30 lines)**

1 │ ﻿using Microsoft.EntityFrameworkCore;

2 │ using QuanLyBanHang.Models.QLBHEntity;

3 │

4 │ namespace QuanLyBanHang.Models.ExModel.OPS

5 │ {

6 │ public class LoaiNhapXuatModel : Base<StockLoaiNhapXuat>

7 │ {

8 │ public LoaiNhapXuatModel(DbContext dbContext) : base(dbContext)

9 │ {

10 │ }

11 │ public StockLoaiNhapXuat GetItem(int Ma)

12 │ {

13 │ return First(t => t.Ma == Ma && !(t.Deleted ?? false));

14 │ }

15 │

16 │ public List<StockLoaiNhapXuat> GetData()

17 │ {

18 │ return Get(t => !(t.Deleted ?? false)).OrderBy(t => t.Ten).ToList();

19 │ }

20 │

21 │ public List<StockLoaiNhapXuat> DanhSachLoaiNhapKho()

22 │ {

23 │ return Get(t => !(t.Deleted ?? false) && (t.NhapKho ?? false)).OrderBy(t => t.Ten).ToList();

24 │ }

25 │ public List<StockLoaiNhapXuat> DanhSachLoaiXuatKho()

26 │ {

27 │ return Get(t => !(t.Deleted ?? false) && !(t.NhapKho ?? false)).OrderBy(t => t.Ten).ToList();

28 │ }

29 │ }

30 │ }

────────────────────────────────────────────────────────────

**📄 LogActionAttribute.cs (CS, 44 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc.Filters;

2 │ using QuanLyBanHang.Controllers;

3 │

4 │ namespace QuanLyBanHang.Common.Attributes

5 │ {

6 │ public class LogActionAttribute : ActionFilterAttribute

7 │ {

8 │ public override void OnActionExecuting(ActionExecutingContext context)

9 │ {

10 │ var controller = context.Controller as BaseController;

11 │ if (controller == null)

12 │ return;

13 │

14 │ var logger = context.HttpContext.RequestServices.GetService(typeof(ILogger<LogActionAttribute>)) as ILogger;

15 │

16 │ var actionName = context.ActionDescriptor.DisplayName;

17 │ var userId = controller.objToken?.userid;

18 │ var menuId = controller.m\_MenuID;

19 │

20 │ logger?.LogInformation("➡️ [START] {Action} | User: {UserId} | Menu: {MenuId}", actionName, userId, menu...

21 │ }

22 │

23 │ public override void OnActionExecuted(ActionExecutedContext context)

24 │ {

25 │ var controller = context.Controller as BaseController;

26 │ if (controller == null)

27 │ return;

28 │

29 │ var logger = context.HttpContext.RequestServices.GetService(typeof(ILogger<LogActionAttribute>)) as ILogger;

30 │

31 │ var actionName = context.ActionDescriptor.DisplayName;

32 │ var userId = controller.objToken?.userid;

33 │

34 │ if (context.Exception == null)

35 │ {

36 │ logger?.LogInformation("✅ [END] {Action} | User: {UserId}", actionName, userId);

37 │ }

38 │ else

39 │ {

40 │ logger?.LogError(context.Exception, "❌ [ERROR] {Action} | User: {UserId}", actionName, userId);

41 │ }

42 │ }

43 │ }

44 │ }

────────────────────────────────────────────────────────────

**📄 LogVietQr.cs (CS, 25 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class LogVietQr

7 │ {

8 │ public Guid Ma { get; set; }

9 │

10 │ public Guid? MaNganHang { get; set; }

11 │

12 │ public string? RequestId { get; set; }

13 │

14 │ public string? Payload { get; set; }

15 │

16 │ public string? SubPayload { get; set; }

17 │

18 │ public string? Error { get; set; }

19 │

20 │ public string? Description { get; set; }

21 │

22 │ public DateTime? CreateDate { get; set; }

23 │

24 │ public bool? IsTest { get; set; }

25 │ }

────────────────────────────────────────────────────────────

**📄 LoginController.cs (CS, 267 lines)**

1 │ ﻿using HongHungHIS.Common.Firebase;

2 │ using Microsoft.AspNetCore.Mvc;

3 │ using NetTools;

4 │ using Newtonsoft.Json;

5 │ using QuanLyBanHang.Common;

6 │ using QuanLyBanHang.Common.Attributes;

7 │ using QuanLyBanHang.Models.APIModels;

8 │ using QuanLyBanHang.Models.ExModel.SYS;

9 │ using QuanLyBanHang.Models.QLBHEntity;

10 │ using System.Net;

11 │

12 │ namespace QuanLyBanHang.Controllers.SYS

13 │ {

14 │ public class LoginController : BaseController

15 │ {

16 │ private class Login

17 │ {

18 │ public string Message { get; set; }

19 │ public int Result { get; set; }

20 │ public string ResultText { get; set; }

21 │ public string url { get; set; }

22 │ }

23 │

24 │ private class LoginCount

25 │ {

26 │ public string id { get; set; }

27 │ public string name { get; set; }

28 │ public string ip { get; set; }

29 │ public int logincount { get; set; }

30 │ public DateTime date { get; set; }

31 │ }

32 │ private static List<LoginCount> loginCount = new List<LoginCount>();

33 │ private readonly IWebHostEnvironment \_env;

34 │ private bool isDev { get { return \_env.IsDevelopment(); } set { return; } }

35 │ //private bool isDev { get { return true; } set { return; } }

36 │ //private bool isDev { get { return false; } set { return; } }

37 │

38 │ public LoginController(IWebHostEnvironment env, QuanLyBanHangContext dbcontext, MemoryCacheManager memoryCache) ...

39 │ {

40 │ m\_MenuID = MenuID.sysUser.GetEnumGuid();

41 │ \_env = env;

42 │ }

43 │

44 │ public IActionResult Index()

45 │ {

46 │ ViewData["ListBranch"] = dbcontext.SysBranch.Where(t => !(t.Deleted ?? false) && (t.Status ?? false)).ToList...

47 │ ViewData["LogoBranch"] = dbcontext.SysLogo.Where(t => !(t.Deleted ?? false)).ToList();

48 │ return View();

49 │ }

50 │

51 │ private string getIP()

52 │ {

53 │ return Request.Headers["X-Real-IP"].ToString() ?? "";

54 │ }

55 │ [HttpPost]

56 │ public IActionResult Authenticate(string Username, string Password, string Remember, string id, string device, s...

57 │ {

58 │ Login login = new Login();

59 │ string token = "";

60 │ if (Username == "" || Password == "" || Branch == "" || Branch == null)

61 │ {

62 │ login.Result = 0;

63 │ login.ResultText = "Thông tin đăng nhập còn trống";

64 │ goto message;

65 │ }

66 │ else

67 │ login.Message = "";

68 │

69 │ string ip = getIP();

70 │

71 │ var loginLog = loginCount.FirstOrDefault(t => t.id == id && t.ip == ip);

72 │ if (loginLog != null && loginLog.logincount >= 3) // một id và ip chỉ được nhập sai tối đa 3 l...

73 │ {

74 │ if (loginLog.date != null)

75 │ {

76 │ if (loginLog.date.AddMinutes(5) >= DateTime.Now)

77 │ {

78 │

79 │ login.Result = 0;

80 │ login.ResultText = "Bạn đăng nhập sai quá nhiều. Vui lòng thử lại sau " + (int)(lo...

81 │ goto message;

82 │ }

83 │ else

84 │ {

85 │ loginLog.logincount = 0;

86 │ }

87 │ }

88 │ }

89 │ if (ip != "" && loginCount.Where(t => t.name == Username && t.date.AddMinutes(5) >= DateTime.Now).Count() >=...

90 │ // một username chỉ được nhập sai tối đa 3 lần tối đa 2 id. Nếu quá 3 lần tài khoả...

91 │ {

92 │ login.Result = 0;

93 │ login.ResultText = "Tài khoản " + Username + " đã nhập sai quá nhiều! Vui lòng thử lại sa...

94 │ goto message;

95 │ }

96 │

97 │ var x = new sys\_UserModel(dbcontext).Authenticate(Username, Password, new Guid(Branch));

98 │ login.Result = x.Item1;

99 │

100 │ if (x.Item1 == 1)//Success

101 │ {

102 │ //string? url = HttpContext.Session.GetString("url");

103 │

104 │ DateTime now = DateTime.Now;

105 │ DateTime expiresDate;

106 │ if (now.Month == 12)

107 │ {

108 │ expiresDate = new DateTime(now.Year + 1, 1, 1);

109 │ }

110 │ else

111 │ {

112 │ expiresDate = new DateTime(now.Year, now.Month + 1, 1);

113 │ }

114 │ token = signToken(new UserToken(x.Item2!.Ma, Branch.GetGuid(), id), device);

115 │ string sign = token.Split('.')[1];

116 │

117 │ // Thêm biến tạm để kiểm tra

118 │ bool enableTestingFcm = true;

119 │

120 │ if (!isDev || enableTestingFcm)

121 │ FirebaseUtility.sendRequestLogin(x.Item2, connectionid, sign);

122 │ Response.Cookies.Append("token", token, new() { Expires = expiresDate });

123 │

124 │ if (loginLog != null)

125 │ {

126 │ loginLog.logincount = 0;

127 │ }

128 │ if (Remember == "1")

129 │ {

130 │ Response.Cookies.Append("Username", Username, new() { Expires = now.AddYears(1) });

131 │ Response.Cookies.Append("Password", Password, new() { Expires = now.AddYears(1) });

132 │ Response.Cookies.Append("Branch", Branch, new() { Expires = now.AddYears(1) });

133 │ }

134 │ else

135 │ {

136 │ Response.Cookies.Append("Username", "", new() { Expires = now.AddDays(-1) });

137 │ Response.Cookies.Append("Password", "", new() { Expires = now.AddDays(-1) });

138 │ Response.Cookies.Append("Branch", "", new() { Expires = now.AddDays(-1) });

139 │ }

140 │

141 │

142 │ login.ResultText = "";

143 │ login.url = "/";

144 │ }

145 │ else if (x.Item1 == 0)

146 │ {

147 │ login.ResultText = "Tài khoản đã bị vô hiệu hóa";

148 │ }

149 │ else

150 │ {

151 │ if (loginLog == null)

152 │ {

153 │ loginLog = new LoginCount

154 │ {

155 │ name = Username,

156 │ id = id,

157 │ ip = ip,

158 │ logincount = 0

159 │ };

160 │ loginCount.Add(loginLog);

161 │ }

162 │ loginLog.logincount++;

163 │ //loginLog.logincount = 1;

164 │ login.ResultText = "Thông tin đăng nhập không chính xác. Tài khoản sẽ bị khóa nếu đă...

165 │ if (loginLog.logincount >= 3)

166 │ {

167 │ login.ResultText = "Bạn đã nhập sai tài khoản quá ba lần !!";

168 │ loginLog.date = DateTime.Now;

169 │ }

170 │ }

171 │

172 │ message:

173 │ //return Json(new ResponseModel(login.Result == 1 ? QLBHConst.OK : QLBHConst.Error, "Vui lòng xác nhận t...

174 │ return Json(new ResponseModel(login.Result == 1 ? (isDev ? QLBHConst.DEVLogin : QLBHConst.OK) : QLBHConst.Er...

175 │ }

176 │

177 │ private string signToken(UserToken objToken, string devicename, bool? accept = null)

178 │ {

179 │ string obj = Utility.Base64Encode(JsonConvert.SerializeObject(objToken));

180 │ string sign = Utility.GetSHA256(obj + AppController.passcode + DateTime.Now.ToString("yyyyMM"));

181 │ string token = obj + "." + sign;

182 │ List<SysToken> loginData = dbcontext.SysToken.Where(t => !(t.Revoked ?? false) && !(t.IsMobile ?? false) && ...

183 │ if (loginData.Count > 0)

184 │ {

185 │ foreach (var login in loginData)

186 │ {

187 │ login.Revoked = true;

188 │ }

189 │ dbcontext.SysToken.UpdateRange(loginData);

190 │ dbcontext.SaveChanges();

191 │ }

192 │

193 │ if (!dbcontext.SysToken.Any(t => t.Token == sign && !(t.Revoked ?? false)))

194 │ {

195 │ dbcontext.SysToken.Add(new SysToken

196 │ {

197 │ UserId = objToken.userid,

198 │ Token = sign,

199 │ Ip = getIP(),

200 │ DevicesId = objToken.id,

201 │ IsMobile = false,

202 │ //Accepted = accept ?? true,

203 │ Accepted = accept ?? isDev,

204 │ DeviceName = devicename ?? "Không xác định",

205 │ LastSeen = DateTime.Now,

206 │ CreatedDate = DateTime.Now,

207 │ });

208 │ dbcontext.SaveChanges();

209 │ }

210 │ return token;

211 │ }

212 │

213 │ [HttpGet]

214 │ [TokenUserCheckHTTP(Permission.LoginAs)]

215 │ public IActionResult LoginAs(string userid)

216 │ {

217 │ string ip = getIP();

218 │ var rangeA = IPAddressRange.Parse("172.16.10.0/255.255.255.0");

219 │ if (!\_env.IsDevelopment() && (ip == "" || !rangeA.Contains(IPAddress.Parse(ip))))

220 │ {

221 │ return Redirect("/pagenotfound/LoginAsDenied?ip=" + ip);

222 │ }

223 │ DateTime now = DateTime.Now;

224 │ DateTime expiresDate;

225 │ if (now.Month == 12)

226 │ {

227 │ expiresDate = new DateTime(now.Year + 1, 1, 1);

228 │ }

229 │ else

230 │ {

231 │ expiresDate = new DateTime(now.Year, now.Month + 1, 1);

232 │ }

233 │ objToken.loginAs = true;

234 │ objToken.idAs = new Guid(userid);

235 │ string token = signToken(objToken, "LoginAs", true);

236 │ Response.Cookies.Append("token", token, new() { Expires = expiresDate });

237 │ //resign token

238 │ return Redirect("/home");

239 │ }

240 │

241 │ [HttpGet]

242 │ [TokenUserCheckHTTP(Permission.NotCheck)]

243 │ public IActionResult Logout()

244 │ {

245 │ if (objToken?.loginAs ?? false)

246 │ {

247 │ DateTime now = DateTime.Now;

248 │ DateTime expiresDate;

249 │ if (now.Month == 12)

250 │ {

251 │ expiresDate = new DateTime(now.Year + 1, 1, 1);

252 │ }

253 │ else

254 │ {

255 │ expiresDate = new DateTime(now.Year, now.Month + 1, 1);

256 │ }

257 │ objToken.loginAs = false;

258 │ objToken.idAs = null;

259 │ string token = signToken(objToken, "LoginAs", true);

260 │ Response.Cookies.Append("token", token, new() { Expires = expiresDate });

261 │ return Redirect("/home");

262 │ }

263 │ Response.Cookies.Delete("token");

264 │ return Redirect("/login");

265 │ }

266 │ }

267 │ }

────────────────────────────────────────────────────────────

**📄 LopHang.cs (CS, 33 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class LopHang

7 │ {

8 │ public int LopId { get; set; }

9 │

10 │ public string TenLop { get; set; } = null!;

11 │

12 │ public int? NguoiTao { get; set; }

13 │

14 │ public DateTime NgayTao { get; set; }

15 │

16 │ public bool? TrangThai { get; set; }

17 │

18 │ public bool? ChoPhepBanAm { get; set; }

19 │

20 │ public DateTime? NgaySua { get; set; }

21 │

22 │ public int? NguoiSua { get; set; }

23 │

24 │ public bool? DaXoa { get; set; }

25 │

26 │ public int? NguoiXoa { get; set; }

27 │

28 │ public DateTime? NgayXoa { get; set; }

29 │

30 │ public int? ThoiGianBatDau { get; set; }

31 │

32 │ public int? ThoiGianKetThuc { get; set; }

33 │ }

────────────────────────────────────────────────────────────

**📄 LopHangController.cs (CS, 232 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using QuanLyBanHang.Common;

3 │ using QuanLyBanHang.Common.Attributes;

4 │ using QuanLyBanHang.Models.APIModels;

5 │ using QuanLyBanHang.Models.QLBHEntity;

6 │

7 │ namespace QuanLyBanHang.Controllers.CAT

8 │ {

9 │ public class LopHangController : BaseController

10 │ {

11 │ private readonly ILogger<LopHangController> \_logger;

12 │

13 │ public LopHangController(

14 │ QuanLyBanHangContext dbcontext,

15 │ MemoryCacheManager memoryCache,

16 │ ILogger<LopHangController> logger) : base(dbcontext, memoryCache)

17 │ {

18 │ m\_MenuID = MenuID.catLopHang.GetEnumGuid();

19 │ \_logger = logger;

20 │ }

21 │

22 │ [TokenUserCheckHTTP]

23 │ public IActionResult Index()

24 │ {

25 │ \_logger.LogInformation("Người dùng {UserId} truy cập trang danh sách Lớp hàng.", objToken.userid);

26 │ return View();

27 │ }

28 │

29 │ [TokenUserCheckHTTP(Permission.Add)]

30 │ public ActionResult Add()

31 │ {

32 │ \_logger.LogInformation("Người dùng {UserId} truy cập trang thêm mới Lớp hàng.", objToken.userid)...

33 │ try

34 │ {

35 │ ViewData["ListLoaiTieuDung"] = dbcontext.TblLoaiTieuDung.Where(t => !(t.Deleted ?? false)).ToList();

36 │ return View("Form");

37 │ }

38 │ catch (Exception ex)

39 │ {

40 │ \_logger.LogError(ex, "Lỗi khi tải dữ liệu cho trang thêm mới Lớp hàng.");

41 │ return View("Error");

42 │ }

43 │ }

44 │

45 │ [TokenUserCheckHTTP(Permission.Edit)]

46 │ public ActionResult Edit(string Ma)

47 │ {

48 │ \_logger.LogInformation("Người dùng {UserId} yêu cầu chỉnh sửa Lớp hàng. Mã (đã mã hóa): {...

49 │ try

50 │ {

51 │ Ma = Ma.Decrypt();

52 │ int id = Ma.ToInt();

53 │ ViewData["ListLoaiTieuDung"] = dbcontext.TblLoaiTieuDung.Where(t => !(t.Deleted ?? false)).ToList();

54 │ ViewData["item"] = dbcontext.TblLopHangHoa.FirstOrDefault(t => t.Ma == id && !(t.Deleted ?? false));

55 │ return View("Form");

56 │ }

57 │ catch (Exception ex)

58 │ {

59 │ \_logger.LogError(ex, "Lỗi khi tải dữ liệu để chỉnh sửa Lớp hàng. Mã (đã mã hóa): ...

60 │ return RedirectToAction("Index");

61 │ }

62 │ }

63 │

64 │ [HttpPost]

65 │ [TokenUserCheckHTTP(Permission.Add)]

66 │ public IActionResult AddAPI(string Ma, string Ten, string Status, string BanAm, string ThoiGianMoban, string MaL...

67 │ {

68 │ \_logger.LogInformation("Người dùng {UserId} thêm mới Lớp hàng: {TenLopHang}", objToken.userid, Ten...

69 │ var result = CheckRequire(Ten, Status);

70 │ if (result != null)

71 │ return Json(result);

72 │

73 │ try

74 │ {

75 │ int? thoiGianBatDau = null;

76 │ int? thoiGianKetThuc = null;

77 │ if (isAllDay.Equals("false") && !string.IsNullOrEmpty(ThoiGianMoban))

78 │ {

79 │ thoiGianBatDau = Cast.ToInt(ThoiGianMoban.Split('|')[0]);

80 │ thoiGianKetThuc = Cast.ToInt(ThoiGianMoban.Split('|')[1]);

81 │ }

82 │

83 │ TblLopHangHoa item = new TblLopHangHoa

84 │ {

85 │ Ten = Ten,

86 │ Status = Status == "1",

87 │ ChoPhepBanAm = BanAm == "1",

88 │ ThoiGianBatDau = thoiGianBatDau,

89 │ ThoiGianKetThuc = thoiGianKetThuc,

90 │ MaChiNhanh = objToken.branch,

91 │ PrinterName = PrinterName,

92 │ UserCreated = objToken.userid,

93 │ CreatedDate = DateTime.Now,

94 │ MaLoaiTieuDung = MaLoaiTieuDung.ToInt(),

95 │ };

96 │

97 │ dbcontext.TblLopHangHoa.Add(item);

98 │ dbcontext.SaveChanges();

99 │ \_logger.LogInformation("Thêm mới Lớp hàng ID {LopHangId} thành công.", item.Ma);

100 │ return Json(QLBHConst.ResponseAddSuccessfully);

101 │ }

102 │ catch (Exception ex)

103 │ {

104 │ \_logger.LogError(ex, "Lỗi khi thêm mới Lớp hàng: {TenLopHang}", Ten);

105 │ return Json(new ResponseModel(QLBHConst.Error, "Lỗi hệ thống khi thêm mới."));

106 │ }

107 │ }

108 │

109 │ [HttpPost]

110 │ [TokenUserCheckHTTP(Permission.Edit)]

111 │ public ActionResult UpdateAPI(string Ma, string Ten, string Status, string BanAm, string ThoiGianMoBan, string M...

112 │ {

113 │ \_logger.LogInformation("Người dùng {UserId} cập nhật Lớp hàng. Mã (đã mã hóa): {MaLopHang}",...

114 │ try

115 │ {

116 │ Ma = Ma.Decrypt();

117 │ int id = Ma.ToInt();

118 │ var result = CheckRequire(Ten, Status);

119 │ if (result != null)

120 │ return Json(result);

121 │

122 │ var item = dbcontext.TblLopHangHoa.FirstOrDefault(x => x.Ma == id && !(x.Deleted ?? false));

123 │ if (item == null)

124 │ {

125 │ \_logger.LogWarning("Không tìm thấy Lớp hàng ID: {LopHangId} để cập nhật.", id);

126 │ return Json(QLBHConst.ResponseNotExist);

127 │ }

128 │

129 │ if (isAllDay.Equals("true"))

130 │ {

131 │ item.ThoiGianBatDau = null;

132 │ item.ThoiGianKetThuc = null;

133 │ }

134 │ else

135 │ {

136 │ item.ThoiGianBatDau = Cast.ToInt(ThoiGianMoBan.Split('|')[0]);

137 │ item.ThoiGianKetThuc = Cast.ToInt(ThoiGianMoBan.Split('|')[1]);

138 │ }

139 │

140 │ item.Ten = Ten;

141 │ item.Status = Status == "1";

142 │ item.ChoPhepBanAm = BanAm == "1";

143 │ item.PrinterName = PrinterName;

144 │ item.UserModified = objToken.userid;

145 │ item.ModifiedDate = DateTime.Now;

146 │ item.MaLoaiTieuDung = MaLoaiTieuDung.ToInt();

147 │

148 │ dbcontext.SaveChanges();

149 │ \_logger.LogInformation("Cập nhật Lớp hàng ID {LopHangId} thành công.", id);

150 │ return Json(QLBHConst.ResponseUpdateSuccessfully);

151 │ }

152 │ catch (Exception ex)

153 │ {

154 │ \_logger.LogError(ex, "Lỗi khi cập nhật Lớp hàng. Mã (đã mã hóa): {MaLopHang}", Ma);

155 │ return Json(new ResponseModel(QLBHConst.Error, "Lỗi hệ thống khi cập nhật."));

156 │ }

157 │ }

158 │

159 │ [HttpPost]

160 │ [TokenUserCheckHTTP(Permission.Delete)]

161 │ public IActionResult DeleteAPI(string Ma)

162 │ {

163 │ \_logger.LogInformation("Người dùng {UserId} yêu cầu xóa Lớp hàng. Mã (đã mã hóa): {MaLopHang...

164 │ try

165 │ {

166 │ Ma = Ma.Decrypt();

167 │ int id = Ma.ToInt();

168 │ var item = dbcontext.TblLopHangHoa.FirstOrDefault(x => x.Ma == id);

169 │ if (item == null)

170 │ {

171 │ \_logger.LogWarning("Không tìm thấy Lớp hàng ID: {LopHangId} để xóa.", id);

172 │ return Json(QLBHConst.ResponseNotExist);

173 │ }

174 │

175 │ item.Deleted = true;

176 │ item.UserDeleted = objToken.userid;

177 │ item.DeletedDate = DateTime.Now;

178 │ dbcontext.SaveChanges();

179 │

180 │ \_logger.LogInformation("Đã xóa (đánh dấu) Lớp hàng ID {LopHangId} thành công.", id);

181 │ return Json(QLBHConst.ResponseDeleteSuccessfully);

182 │ }

183 │ catch (Exception ex)

184 │ {

185 │ \_logger.LogError(ex, "Lỗi khi xóa Lớp hàng. Mã (đã mã hóa): {MaLopHang}", Ma);

186 │ return Json(new ResponseModel(QLBHConst.Error, "Lỗi hệ thống khi xóa."));

187 │ }

188 │ }

189 │

190 │ private static ResponseModel CheckRequire(String Ten, String Status)

191 │ {

192 │ if (Ten.IsEmpty())

193 │ return new ResponseModel(QLBHConst.Error, "Vui lòng nhập tên");

194 │ if (Status.IsEmpty())

195 │ return new ResponseModel(QLBHConst.Error, "Vui lòng nhập trạng thái");

196 │ return null;

197 │ }

198 │

199 │ [TokenUserCheckHTTP(Permission.NotCheck)]

200 │ public IActionResult getListData()

201 │ {

202 │ \_logger.LogInformation("Người dùng {UserId} yêu cầu lấy danh sách Lớp hàng.", objToken.userid);

203 │ try

204 │ {

205 │ var listData = (from lh in dbcontext.TblLopHangHoa

206 │ join ltd in dbcontext.TblLoaiTieuDung on lh.MaLoaiTieuDung equals ltd.Ma into ltdJoin

207 │ from ltd in ltdJoin.DefaultIfEmpty() // LEFT JOIN

208 │ where !(lh.Deleted ?? false) && lh.MaChiNhanh == objToken.branch

209 │ select new

210 │ {

211 │ maEncrypt = lh.Ma.Encrypt(),

212 │ Ma = lh.Ma,

213 │ Ten = lh.Ten,

214 │ ChoPhepBanAm = lh.ChoPhepBanAm,

215 │ ThoiGianBatDau = lh.ThoiGianBatDau,

216 │ ThoiGianKetThuc = lh.ThoiGianKetThuc,

217 │ trangthai = lh.Status ?? false,

218 │ tentrangthai = (lh.Status ?? false) ? "Hoạt động" : "Hết hoạt động",

219 │ TenLoaiTieuDung = ltd != null ? ltd.LoaiTieuDung : ""

220 │ }).OrderBy(t => t.Ma).ToList();

221 │

222 │ \_logger.LogInformation("Lấy thành công {Count} bản ghi Lớp hàng.", listData.Count);

223 │ return Json(new ResponseModel(QLBHConst.OK, "", listData));

224 │ }

225 │ catch (Exception ex)

226 │ {

227 │ \_logger.LogError(ex, "Lỗi khi lấy danh sách Lớp hàng cho chi nhánh {MaChiNhanh}", objToken.bran...

228 │ return Json(new ResponseModel(QLBHConst.Error, "Lỗi hệ thống khi lấy danh sách."));

229 │ }

230 │ }

231 │ }

232 │ }

────────────────────────────────────────────────────────────

**📄 LopHangHoaModel.cs (CS, 132 lines)**

1 │ ﻿using Microsoft.EntityFrameworkCore;

2 │ using QuanLyBanHang.Common;

3 │ using QuanLyBanHang.Models.QLBHEntity;

4 │ using QuanLyBanHang.Models.ViewModels;

5 │

6 │ namespace QuanLyBanHang.Models.ExModel.CAT

7 │ {

8 │ public class LopHangTheoKho

9 │ {

10 │ public int Ma { get; set; }

11 │ public string? Ten { get; set; }

12 │ }

13 │ public class TopLopHang

14 │ {

15 │ public int Ma {get;set;}

16 │ public int MaKho {get;set;}

17 │ public string? Ten {get;set;}

18 │ public string? HinhAnh { get; set; }

19 │ }

20 │

21 │ public class LopHangHoaModel : Base<TblLopHangHoa>

22 │ {

23 │ public LopHangHoaModel(DbContext dbContext) : base(dbContext)

24 │ {

25 │ }

26 │ QuanLyBanHangContext dbContext

27 │ {

28 │ get { return \_dbContext as QuanLyBanHangContext; }

29 │ set { return; }

30 │ }

31 │ public TblLopHangHoa GetItem(int Ma, Guid MaChiNhanh)

32 │ {

33 │ return First(t => t.Ma == Ma && !(t.Deleted ?? false) && t.MaChiNhanh == MaChiNhanh);

34 │ }

35 │

36 │ public List<TblLopHangHoa> GetData(Guid MaChiNhanh)

37 │ {

38 │ return Get(t => !(t.Deleted ?? false)).OrderBy(t => t.Ten).ToList();

39 │ }

40 │ public List<TopLopHang> GetTopLopHang(Guid MaChiNhanh)

41 │ {

42 │ var NamThang = DateTime.Now.ToString("yyyyMM");

43 │ var nowHour = DateTime.Now.TimeOfDay.TotalHours;

44 │ var listTK = dbContext.StockTonKhoBanHang.AsNoTracking().Where(t => t.NamThang == NamThang && t.MaChiNhanh =...

45 │ var listKho = dbContext.StockDanhMucKho.AsNoTracking().Where(x => !(x.Deleted ?? false) && x.MaChiNhanh == M...

46 │ var listHH = dbContext.TblHangHoa.AsNoTracking().Where(t => !(t.Deleted ?? false) && !(t.HetHang ?? false) &...

47 │ var listGiaBanTheoKho = dbContext.TblGiaBanTheoKho.AsNoTracking().Where(t => t.MaChiNhanh == MaChiNhanh ).To...

48 │ var listDVT = dbContext.TblDonViTinh.AsNoTracking().Where(t => !(t.Deleted ?? false)).ToList();

49 │ var listLopHang = dbContext.TblLopHangHoa.AsNoTracking().Where(t => !(t.Deleted ?? false) && t.MaChiNhanh ==...

50 │

51 │ var giabantheokho = (from gia in listGiaBanTheoKho

52 │ join hh in listHH on gia.MaHangHoa equals hh.Ma

53 │ join lh in listLopHang on hh.MaLopHang equals lh.Ma

54 │ join kho in listKho on gia.MaKho equals kho.Ma

55 │ join tk in listTK on gia.MaHangHoa equals tk.MaHangHoa into \_tk

56 │ from tk in \_tk.DefaultIfEmpty()

57 │ join dv in listDVT on hh.MaDvt equals dv.Ma into \_dv

58 │ from dv in \_dv.DefaultIfEmpty()

59 │ where ((tk != null && !(lh.ChoPhepBanAm ?? false) && (tk.SltonCuoiKy ?? 0) > 0) || (lh.ChoPh...

60 │ (

61 │ (lh.ThoiGianBatDau == null || lh.ThoiGianKetThuc == null) ||

62 │ (lh.ThoiGianBatDau <= nowHour && nowHour < lh.ThoiGianKetThuc)

63 │ ) &&

64 │ (

65 │ (kho.ThoiGianBatDau == null || kho.ThoiGianKetThuc == null) ||

66 │ (kho.ThoiGianBatDau <= nowHour && nowHour < kho.ThoiGianKetThuc)

67 │ )) &&

68 │ (kho.HienThiTrongWebOrder ?? false) && (lh.MaLoaiTieuDung ?? 0) != (int)LoaiTieuDung.Kha...

69 │ select new TonKhoVaGiaBanTheoKhoModel

70 │ {

71 │ Ma = gia.Ma,

72 │ MaVach = hh.MaVach,

73 │ TenHang = hh.Ten,

74 │ GiaBan = gia.GiaBan ?? 0,

75 │ DVT = hh.MaDvt ?? 0,

76 │ LopHang = hh.MaLopHang ?? 0,

77 │ TenLop = lh?.Ten ?? "",

78 │ MaKho = gia.MaKho ?? 0,

79 │ TenKho = "",

80 │ TenDVT = dv?.Ten ?? "",

81 │ SLTon = tk == null ? 0 : Convert.ToInt64(tk.SltonCuoiKy ?? 0),

82 │ HinhAnh = hh.HinhAnh + "",

83 │ BanNoiBo = hh.BanNoiBo ?? false,

84 │ MaHangHoa = hh.Ma,

85 │ LoaiTieuDung = lh.MaLoaiTieuDung ?? 0

86 │ }).ToList();

87 │ // Group theo LopHang và MaKho, lấy random 1 item trong mỗi group để lấy HinhAnh

88 │ var lophangList = giabantheokho

89 │ .Where(x => !string.IsNullOrEmpty(x.HinhAnh))

90 │ .GroupBy(x => new { x.LopHang, x.MaKho })

91 │ .Select(g =>

92 │ {

93 │ var random = new Random();

94 │ var randomItem = g.OrderBy(x => random.Next()).FirstOrDefault();

95 │ return new TopLopHang

96 │ {

97 │ Ma = g.Key.LopHang,

98 │ MaKho = g.Key.MaKho,

99 │ Ten = randomItem?.TenLop ?? "",

100 │ HinhAnh = randomItem?.HinhAnh ?? "",

101 │ };

102 │ })

103 │ .ToList();

104 │

105 │ return lophangList;

106 │ }

107 │ public List<LopHangTheoKho> GetLopHangTheoKho(int MaKho)

108 │ {

109 │ var nowHour = DateTime.Now.TimeOfDay.TotalHours;

110 │ var namThang = DateTime.Now.ToString("yyyyMM");

111 │ var result = (

112 │ from tk in dbContext.StockTonKhoBanHang.Where(x=> x.MaKho == MaKho && x.NamThang == namThang)

113 │ join hh in dbContext.TblHangHoa on tk.MaHangHoa equals hh.Ma

114 │ join lh in dbContext.TblLopHangHoa on hh.MaLopHang equals lh.Ma

115 │ where

116 │ ((tk.SltonCuoiKy ?? 0) > 0 || (lh.ChoPhepBanAm ?? false)) &&

117 │ (

118 │ (lh.ThoiGianBatDau == null || lh.ThoiGianKetThuc == null) ||

119 │ (lh.ThoiGianBatDau <= nowHour && nowHour < lh.ThoiGianKetThuc)

120 │ )

121 │ group lh by new { lh.Ma, lh.Ten } into g

122 │ select new LopHangTheoKho

123 │ {

124 │ Ma = g.Key.Ma,

125 │ Ten = g.Key.Ten

126 │ }

127 │ ).ToList();

128 │ return result;

129 │ }

130 │

131 │ }

132 │ }

────────────────────────────────────────────────────────────

**📄 MapStoredProcedure.cs (CS, 34 lines)**

1 │ ﻿using Microsoft.EntityFrameworkCore;

2 │ using QuanLyBanHang.Models.ExModel.StoredProcedureModel;

3 │ using QuanLyBanHang.Models.StoredProcedureModel;

4 │

5 │ namespace QuanLyBanHang.Models.QLBHEntity

6 │ {

7 │ public partial class QuanLyBanHangContext

8 │ {

9 │ public virtual DbSet<sp\_RoleMenuResult> sp\_RoleMenu { get; set; }

10 │ public virtual DbSet<sp\_RolePermissionResult> sp\_RolePermission { get; set; }

11 │ public virtual DbSet<sp\_BaoCaoTonKhoTheoNgayResult> sp\_BaoCaoTonKhoTheoNgay { get; set; }

12 │ public virtual DbSet<sp\_NhapHangImportFASTResult> sp\_NhapHangImportFAST { get; set; }

13 │ partial void OnModelCreatingPartial(ModelBuilder modelBuilder)

14 │ {

15 │ modelBuilder.Entity<sp\_RoleMenuResult>(entity =>

16 │ {

17 │ entity.HasNoKey();

18 │ });

19 │ modelBuilder.Entity<sp\_RolePermissionResult>(entity =>

20 │ {

21 │ entity.HasNoKey();

22 │ });

23 │ modelBuilder.Entity<sp\_BaoCaoTonKhoTheoNgayResult>(entity =>

24 │ {

25 │ entity.HasNoKey();

26 │ });

27 │ modelBuilder.Entity<sp\_NhapHangImportFASTResult>(entity =>

28 │ {

29 │ entity.HasNoKey();

30 │ });

31 │ modelBuilder.Entity<sp\_BaoCaoKiemKe>().HasNoKey();

32 │ }

33 │ }

34 │ }

────────────────────────────────────────────────────────────

**📄 MemoryCacheManager.cs (CS, 33 lines)**

1 │ ﻿using Microsoft.Extensions.Caching.Memory;

2 │ using Microsoft.Extensions.Options;

3 │

4 │ namespace QuanLyBanHang.Common

5 │ {

6 │ public sealed class MemoryCacheManager : IMemoryCache

7 │ {

8 │ readonly IOptions<MemoryCacheOptions> \_optionsAccessor;

9 │

10 │ IMemoryCache \_inner;

11 │

12 │ public MemoryCacheManager(IOptions<MemoryCacheOptions> optionsAccessor)

13 │ {

14 │ \_optionsAccessor = optionsAccessor;

15 │ \_inner = new MemoryCache(\_optionsAccessor);

16 │ }

17 │

18 │ public void Reset()

19 │ {

20 │ var old = \_inner;

21 │ \_inner = new MemoryCache(\_optionsAccessor);

22 │ old.Dispose();

23 │ }

24 │

25 │ public void Dispose() => \_inner.Dispose();

26 │

27 │ public ICacheEntry CreateEntry(object key) => \_inner.CreateEntry(key);

28 │

29 │ public void Remove(object key) => \_inner.Remove(key);

30 │

31 │ public bool TryGetValue(object key, out object value) => \_inner.TryGetValue(key, out value);

32 │ }

33 │ }

────────────────────────────────────────────────────────────

**📄 MemoryMetrics.cs (CS, 85 lines)**

1 │ ﻿using System.Diagnostics;

2 │ using System.Runtime.InteropServices;

3 │

4 │ namespace QuanLyBanHang.Common

5 │ {

6 │ public class MemoryMetrics

7 │ {

8 │ public double Total;

9 │ public double Used;

10 │ public double Free;

11 │ }

12 │

13 │ public class MemoryMetricsClient

14 │ {

15 │ public MemoryMetrics GetMetrics()

16 │ {

17 │ if (IsUnix())

18 │ {

19 │ return GetUnixMetrics();

20 │ }

21 │

22 │ return GetWindowsMetrics();

23 │ }

24 │

25 │ private bool IsUnix()

26 │ {

27 │ var isUnix = RuntimeInformation.IsOSPlatform(OSPlatform.OSX) ||

28 │ RuntimeInformation.IsOSPlatform(OSPlatform.Linux);

29 │

30 │ return isUnix;

31 │ }

32 │

33 │ private MemoryMetrics GetWindowsMetrics()

34 │ {

35 │ var output = "";

36 │

37 │ var info = new ProcessStartInfo();

38 │ info.FileName = "wmic";

39 │ info.Arguments = "OS get FreePhysicalMemory,TotalVisibleMemorySize /Value";

40 │ info.RedirectStandardOutput = true;

41 │

42 │ using (var process = Process.Start(info))

43 │ {

44 │ output = process.StandardOutput.ReadToEnd();

45 │ }

46 │

47 │ var lines = output.Trim().Split("\n");

48 │ var freeMemoryParts = lines[0].Split("=", StringSplitOptions.RemoveEmptyEntries);

49 │ var totalMemoryParts = lines[1].Split("=", StringSplitOptions.RemoveEmptyEntries);

50 │

51 │ var metrics = new MemoryMetrics();

52 │ metrics.Total = Math.Round(double.Parse(totalMemoryParts[1]) / 1024, 0);

53 │ metrics.Free = Math.Round(double.Parse(freeMemoryParts[1]) / 1024, 0);

54 │ metrics.Used = metrics.Total - metrics.Free;

55 │

56 │ return metrics;

57 │ }

58 │

59 │ private MemoryMetrics GetUnixMetrics()

60 │ {

61 │ var output = "";

62 │

63 │ var info = new ProcessStartInfo("free -m");

64 │ info.FileName = "/bin/bash";

65 │ info.Arguments = "-c \"free -m\"";

66 │ info.RedirectStandardOutput = true;

67 │

68 │ using (var process = Process.Start(info))

69 │ {

70 │ output = process.StandardOutput.ReadToEnd();

71 │ Console.WriteLine(output);

72 │ }

73 │

74 │ var lines = output.Split("\n");

75 │ var memory = lines[1].Split(" ", StringSplitOptions.RemoveEmptyEntries);

76 │

77 │ var metrics = new MemoryMetrics();

78 │ metrics.Total = double.Parse(memory[1]);

79 │ metrics.Used = double.Parse(memory[2]);

80 │ metrics.Free = double.Parse(memory[3]);

81 │

82 │ return metrics;

83 │ }

84 │ }

85 │ }

────────────────────────────────────────────────────────────

**📄 MenuController.cs (CS, 78 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using QuanLyBanHang.Common;

3 │ using QuanLyBanHang.Common.Attributes;

4 │ //using QuanLyBanHang.Models.ExModel.CAT;

5 │ using QuanLyBanHang.Models.ExModel.SYS;

6 │ using QuanLyBanHang.Models.QLBHEntity;

7 │

8 │ namespace QuanLyBanHang.Controllers.SYS

9 │ {

10 │ public class MenuController : BaseController

11 │ {

12 │ public MenuController(QuanLyBanHangContext dbcontext, MemoryCacheManager memoryCache) : base(dbcontext, memoryCa...

13 │ {

14 │ m\_MenuID = MenuID.sysMenu.GetEnumGuid();

15 │ }

16 │ [TokenUserCheckHTTP()]

17 │ public IActionResult Index()

18 │ {

19 │ ViewData["ListData"] = new sys\_MenuModel(dbcontext).GetData();

20 │ //ViewData["ListDataPhongKham"] = new cat\_PhongKhamModel(dbcontext).GetData();

21 │ return View();

22 │ }

23 │ [TokenUserCheckHTTP(Permission.Add)]

24 │ public IActionResult Add()

25 │ {

26 │ ViewData["ListMenu"] = new sys\_MenuModel(dbcontext).GetData().Where(t => (t.Status ?? false)).ToList();

27 │ return View("Form");

28 │ }

29 │

30 │ [TokenUserCheckHTTP(Permission.Edit)]

31 │ public IActionResult Edit(string ID)

32 │ {

33 │ //

34 │ ViewData["item"] = new sys\_MenuModel(dbcontext).GetItem(ID.GetGuid());

35 │ ViewData["ListMenu"] = new sys\_MenuModel(dbcontext).GetData().Where(t => (t.Status ?? false)).ToList();

36 │ return View("Form");

37 │ }

38 │

39 │ [HttpPost]

40 │ [TokenUserCheckHTTP(Permission.Edit)]

41 │ public IActionResult UpdateAPI(string ID, string Code, string ParentID, string Name, string Sequence, string Lin...

42 │ , string Icon, string Status, string ChildOfMenu, string UsingFor)

43 │ {

44 │

45 │ int result = new sys\_MenuModel(dbcontext).Edit(ID.GetGuid(), Code, ParentID.GetGuidNull(), Name, Cast.ToInt(...

46 │ , Link, Icon, Status == "1", objToken!.userid, ChildOfMenu.GetGuidNull(), UsingFor);

47 │ memoryCache.Clear(t => t.Contains(QLBHConst.cacheKeyMenu));

48 │ return Json(Utility.MakeResponseFromEditResult(Cast.ToString(result)));

49 │ }

50 │

51 │ [HttpPost]

52 │ [TokenUserCheckHTTP(Permission.Add)]

53 │ public IActionResult AddAPI(string ID, string Code, string ParentID, string Name, string Sequence, string Link

54 │ , string Icon, string Status, string ChildOfMenu, string UsingFor)

55 │ {

56 │

57 │ var result = new sys\_MenuModel(dbcontext).Add(Code, ParentID.GetGuidNull(), Name, Cast.ToInt(Sequence), Link...

58 │ Icon, Status == "1", objToken!.userid, ChildOfMenu.GetGuidNull(), UsingFor);

59 │ memoryCache.Clear(t => t.Contains(QLBHConst.cacheKeyMenu));

60 │ if (result.row > 0 && (Link + "").ToLower().Contains("menuid="))

61 │ {

62 │ new sys\_MenuModel(dbcontext).Edit(result.id ?? Utility.defaultUID, Code, ParentID.GetGuidNull(), Name, C...

63 │ , Link + result.id.ToString(), Icon, Status == "1", objToken!.userid, ChildOfMenu.GetGuidNul...

64 │ }

65 │ return Json(Utility.MakeResponseFromAddResult(Cast.ToString(result.row)));

66 │ }

67 │

68 │ [HttpPost]

69 │ [TokenUserCheckHTTP(Permission.Delete)]

70 │ public IActionResult DeleteAPI(string ID)

71 │ {

72 │ //

73 │ int result = new sys\_MenuModel(dbcontext).Delete(ID.GetGuid(), objToken!.userid);

74 │ memoryCache.Clear(t => t.Contains(QLBHConst.cacheKeyMenu));

75 │ return Json(Utility.MakeResponseFromDeleteResult(result+""));

76 │ }

77 │ }

78 │ }

────────────────────────────────────────────────────────────

**📄 MiddlewareLogAPI.cs (CS, 143 lines)**

1 │ ﻿using QuanLyBanHang.Models.APIModels;

2 │ using QuanLyBanHang.Models.QLBHEntity;

3 │ using Microsoft.IdentityModel.Tokens;

4 │ using Newtonsoft.Json;

5 │ using Newtonsoft.Json.Linq;

6 │ using System.Diagnostics;

7 │ using System.IO;

8 │ using System.Text;

9 │ namespace QuanLyBanHang.Common

10 │ {

11 │ public class MiddlewareLogAPI

12 │ {

13 │ private readonly RequestDelegate \_next;

14 │ public MiddlewareLogAPI(RequestDelegate next)

15 │ {

16 │ \_next = next;

17 │ }

18 │ public async Task Invoke(HttpContext context)

19 │ {

20 │ if (context.Request.Method == "POST" && !context.Request.Headers["token"].IsNullOrEmpty())

21 │ {

22 │ bool isNext = false;

23 │ try

24 │ {

25 │ var \_hisHhhContext = context.RequestServices.GetRequiredService<QuanLyBanHangContext>();

26 │

27 │ string requestBody = "";

28 │ string responseBody = "";

29 │ UserToken? user = new UserToken(JsonConvert.DeserializeObject<JObject>(Utility.Base64Decode(((string...

30 │ string ip = "";

31 │ if (!context.Request.Headers["X-Real-IP"].IsNullOrEmpty())

32 │ {

33 │ ip = context.Request.Headers["X-Real-IP"]!;

34 │ }

35 │ var request = context.Request;

36 │ if (request.Method == HttpMethods.Post && request.ContentLength > 0)

37 │ {

38 │ request.EnableBuffering();

39 │ var buffer = new byte[Convert.ToInt32(request.ContentLength)];

40 │ await request.Body.ReadAsync(buffer, 0, buffer.Length);

41 │ requestBody = Encoding.UTF8.GetString(buffer);

42 │ request.Body.Position = 0;

43 │ }

44 │ var timer = new Stopwatch();

45 │ timer.Start();

46 │ if (\_hisHhhContext.SysVariable.Any(t => t.Code == "LogResponse" && t.Value == "1"))

47 │ {

48 │ Stream originalBody = context.Response.Body;

49 │ try

50 │ {

51 │ using (var memStream = new MemoryStream())

52 │ {

53 │ context.Response.Body = memStream;

54 │ await \_next(context);

55 │ isNext = true;

56 │

57 │ memStream.Position = 0;

58 │ responseBody = new StreamReader(memStream).ReadToEnd();

59 │

60 │ memStream.Position = 0;

61 │ await memStream.CopyToAsync(originalBody);

62 │ }

63 │

64 │ }

65 │ catch (Exception ex)

66 │ {

67 │ \_ = Utility.sendAsync(ex.Message, "Middleware", "Tanione");

68 │ }

69 │ finally

70 │ {

71 │ context.Response.Body = originalBody;

72 │ }

73 │ }

74 │ else

75 │ {

76 │ isNext = true;

77 │ await \_next(context);

78 │ }

79 │ timer.Stop();

80 │ //\_hisHhhContext.SysLogAccess.Add(new SysLogAccess

81 │ //{

82 │ // UserId = user?.userid ?? 0,

83 │ // DeviceId = user?.id,

84 │ // Ip = ip,

85 │ // Path = request.Path,

86 │ // Request = requestBody,

87 │ // Response = responseBody,

88 │ // CreateDate = DateTime.Now,

89 │ // ExecTime = timer.ElapsedMilliseconds+"ms",

90 │ //});

91 │ /\* \_ = Utility.sendAsync(JsonConvert.SerializeObject(new accessLog

92 │ {

93 │ UserId = user?.userid ?? Utility.defaultUID,

94 │ DeviceId = user?.id,

95 │ Ip = ip,

96 │ Path = request.Path,

97 │ Request = requestBody,

98 │ Response = responseBody,

99 │ ExecTime = timer.ElapsedMilliseconds + "ms",

100 │ }),"", "HungDuyFinanceAccounting");\*/

101 │

102 │ // \_hisHhhContext.SaveChanges();

103 │ }

104 │ catch (Exception ex)

105 │ {

106 │ \_ = Utility.sendAsync(ex.Message, "Middleware", "Tanione");

107 │ if (!isNext)

108 │ {

109 │ await \_next(context);

110 │ }

111 │ //TODO write Excetpion to log

112 │ }

113 │ }

114 │ else

115 │ {

116 │ try

117 │ {

118 │ await \_next(context);

119 │ }

120 │ catch (Exception ex)

121 │ {

122 │ \_ = Utility.sendAsync(ex.Message, "Middleware", "Tanione");

123 │ }

124 │ }

125 │ }

126 │ }

127 │ public class accessLog

128 │ {

129 │ public Guid? UserId { get; set; }

130 │

131 │ public string? DeviceId { get; set; }

132 │

133 │ public string? Ip { get; set; }

134 │

135 │ public string? Path { get; set; }

136 │

137 │ public string? Request { get; set; }

138 │

139 │ public string? Response { get; set; }

140 │

141 │ public string? ExecTime { get; set; }

142 │ }

143 │ }

────────────────────────────────────────────────────────────

**📄 MultiSQL.cs (CS, 90 lines)**

1 │ ﻿using QuanLyBanHang.Common.MultiServer;

2 │ using Microsoft.EntityFrameworkCore;

3 │ using Microsoft.EntityFrameworkCore.ChangeTracking;

4 │ using Microsoft.EntityFrameworkCore.Metadata;

5 │ using Microsoft.EntityFrameworkCore.Metadata.Internal;

6 │ using Newtonsoft.Json;

7 │ using System.Collections.Concurrent;

8 │

9 │ namespace QuanLyBanHang.Models.QLBHEntity

10 │ {

11 │ public partial class QuanLyBanHangContext

12 │ {

13 │ public Guid? userId { get; set; }

14 │ public override int SaveChanges()

15 │ {

16 │ // ChangeTracker.DetectChanges();

17 │ // var seed = VSeed.ToList();

18 │ // MultiServer.Instance.setSeed(seed);

19 │ // Parallel.ForEach(

20 │ // ChangeTracker.Entries().Where(t => t.State == EntityState.Added)

21 │ // .Select(t => Model.FindEntityType(t.Entity.GetType())?.GetTableName())

22 │ // .Distinct(), (tablename) =>

23 │ // {

24 │ // if(tablename != null)

25 │ // {

26 │ // var s = seed.FirstOrDefault(t=>t.TableName == tablename);

27 │ // if(s != null)

28 │ // {

29 │ // MultiServer.Instance.checkSeed(tablename, s.CurrentIdentityValue ?? 1, this);

30 │ // }

31 │ // }

32 │ // });

33 │ // \_ = MultiServer.Instance.shareSeed();

34 │ ChangeTracker.DetectChanges();

35 │ var now = DateTime.Now;

36 │ var entity = ChangeTracker.Entries();

37 │ foreach (var entry in entity.Where(t => t.State == EntityState.Added))

38 │ {

39 │ var t = entry.Entity;

40 │ var type = t.GetType();

41 │ var CreatedDate = type.GetProperty("CreatedDate");

42 │ if (CreatedDate != null && CreatedDate.PropertyType == typeof(DateTime?))

43 │ {

44 │ CreatedDate.SetValue(t, now);

45 │ }

46 │ var UserCreated = type.GetProperty("UserCreated");

47 │ if (UserCreated != null && UserCreated.PropertyType == typeof(Guid?) && userId != null)

48 │ {

49 │ UserCreated.SetValue(t, userId);

50 │ }

51 │ }

52 │ foreach (var entry in entity.Where(t => t.State == EntityState.Modified))

53 │ {

54 │ var t = entry.Entity;

55 │ var type = t.GetType();

56 │ var s = t.GetType().GetProperty("Deleted");//detect delete

57 │ if (s != null && s.PropertyType == typeof(bool?))

58 │ {

59 │ if ((bool)(s.GetValue(t) ?? false))

60 │ {

61 │ var DeletedDate = type.GetProperty("DeletedDate");

62 │ if (DeletedDate != null && DeletedDate.PropertyType == typeof(DateTime?))

63 │ {

64 │ DeletedDate.SetValue(t, now);

65 │ }

66 │ var UserDeleted = type.GetProperty("UserDeleted");

67 │ if (UserDeleted != null && UserDeleted.PropertyType == typeof(Guid?) && userId != null)

68 │ {

69 │ UserDeleted.SetValue(t, userId);

70 │ }

71 │ continue; // bypass check Modified

72 │ }

73 │ }

74 │ var ModifiedDate = type.GetProperty("ModifiedDate");

75 │ if (ModifiedDate != null && ModifiedDate.PropertyType == typeof(DateTime?))

76 │ {

77 │ ModifiedDate.SetValue(t, now);

78 │ }

79 │ var UserModified = type.GetProperty("UserModified");

80 │ if (UserModified != null && UserModified.PropertyType == typeof(Guid?) && userId != null)

81 │ {

82 │ UserModified.SetValue(t, userId);

83 │ }

84 │ }

85 │ return base.SaveChanges();

86 │ }

87 │ }

88 │ }

89 │ //Multi SQL ID Generator

90 │ //ConcurrentStack<long> ids = new ConcurrentStack<long>();

────────────────────────────────────────────────────────────

**📄 MultiServer.cs (CS, 50 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using QuanLyBanHang.Models.QLBHEntity;

3 │

4 │ namespace QuanLyBanHang.Common.MultiServer

5 │ {

6 │ public class MultiServer : Controller

7 │ {

8 │ private static IServiceProvider \_serviceProvider;

9 │ private static object \_lock = new object();

10 │ public static void Configure(IServiceProvider serviceProvider)

11 │ {

12 │ \_serviceProvider = serviceProvider;

13 │ }

14 │ public static bool IsDevelopment { get; set; } = false;

15 │ public static async Task<double> GetData(string table, string col, bool resetByDate = false, int count = 1)

16 │ {

17 │ using (var scope = \_serviceProvider.CreateScope())

18 │ {

19 │ lock (\_lock)

20 │ {

21 │ var qlbh = scope.ServiceProvider.GetRequiredService<QuanLyBanHangContext>();

22 │ var s = qlbh.SysStt.FirstOrDefault(t => t.TableName == table && t.Colname == col);

23 │ if (s == null)

24 │ {

25 │ s = new SysStt()

26 │ {

27 │ Ma = Guid.NewGuid(),

28 │ TableName = table,

29 │ Colname = col,

30 │ CreateDate = DateTime.Now,

31 │ TableMax = 0

32 │ };

33 │ qlbh.SysStt.Add(s);

34 │ }

35 │ else

36 │ {

37 │ if (resetByDate && s.CreateDate.Value.Day != DateTime.Now.Day)

38 │ {

39 │ s.CreateDate = DateTime.Now;

40 │ s.TableMax = 0;

41 │ }

42 │ }

43 │ s.TableMax += count;

44 │ qlbh.SaveChanges();

45 │ return (double)s.TableMax;

46 │ }

47 │ }

48 │ }

49 │ }

50 │ }

────────────────────────────────────────────────────────────

**📄 NccChti.cs (CS, 15 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class NccChti

7 │ {

8 │ public string? MãHis { get; set; }

9 │

10 │ public string? TênNhàCungCấpHis { get; set; }

11 │

12 │ public string? MãNccFast { get; set; }

13 │

14 │ public string? TênNccFast { get; set; }

15 │ }

────────────────────────────────────────────────────────────

**📄 NccDược.cs (CS, 51 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class NccDược

7 │ {

8 │ public string? MãNccHis { get; set; }

9 │

10 │ public string? TênNhàCungCấpHis { get; set; }

11 │

12 │ public string? MãNccFast { get; set; }

13 │

14 │ public string? TênNccFast { get; set; }

15 │

16 │ public string? F5 { get; set; }

17 │

18 │ public string? F6 { get; set; }

19 │

20 │ public string? F7 { get; set; }

21 │

22 │ public string? F8 { get; set; }

23 │

24 │ public string? F9 { get; set; }

25 │

26 │ public string? F10 { get; set; }

27 │

28 │ public string? F11 { get; set; }

29 │

30 │ public string? F12 { get; set; }

31 │

32 │ public string? F13 { get; set; }

33 │

34 │ public string? F14 { get; set; }

35 │

36 │ public string? F15 { get; set; }

37 │

38 │ public string? F16 { get; set; }

39 │

40 │ public string? F17 { get; set; }

41 │

42 │ public string? F18 { get; set; }

43 │

44 │ public string? F19 { get; set; }

45 │

46 │ public string? F20 { get; set; }

47 │

48 │ public string? F21 { get; set; }

49 │

50 │ public string? F22 { get; set; }

51 │ }

────────────────────────────────────────────────────────────

**📄 NetPrinterController.cs (CS, 52 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using QuanLyBanHang.Common;

3 │ using QuanLyBanHang.Models.APIModels;

4 │ using QuanLyBanHang.Models.QLBHEntity;

5 │ using System.Drawing;

6 │ using System.Drawing.Printing;

7 │

8 │ namespace QuanLyBanHang.Controllers.SYS

9 │ {

10 │ public class NetPrinterController : BaseController

11 │ {

12 │ public NetPrinterController(QuanLyBanHangContext dbcontext, MemoryCacheManager memoryCache) : base(dbcontext, me...

13 │ {

14 │ }

15 │ private string printContent = "";

16 │ [HttpPost]

17 │ public IActionResult PrintBill(string printerName, string content)

18 │ {

19 │ if (string.IsNullOrWhiteSpace(printerName))

20 │ return Json(new ResponseModelError($"printerName null"));

21 │

22 │ printContent = content;

23 │

24 │ try

25 │ {

26 │ PrintDocument p1 = new PrintDocument();

27 │ p1.PrinterSettings.PrinterName = printerName;

28 │

29 │ if (!p1.PrinterSettings.IsValid)

30 │ return Json(new ResponseModelError($"Không tìm thấy máy in: {printerName}"));

31 │

32 │

33 │ p1.PrintPage += new PrintPageEventHandler(Page\_Print);

34 │ p1.Print();

35 │ return Json(new ResponseModelSuccess($"Đã gửi lệnh in"));

36 │

37 │ }

38 │ catch (Exception ex)

39 │ {

40 │ return Json(new ResponseModelError($"Lỗi khi in: {ex.Message}"));

41 │ }

42 │ }

43 │

44 │ private void Page\_Print(object sender, PrintPageEventArgs ev)

45 │ {

46 │ using Brush b = new SolidBrush(Color.Black);

47 │ using Font printFont = new Font("Lucida Console", 12);

48 │ ev.Graphics.DrawString(printContent, printFont, b, 0, 0, StringFormat.GenericDefault);

49 │ printContent = "";

50 │ }

51 │ }

52 │ }

────────────────────────────────────────────────────────────

**📄 NgayKetCaModel.cs (CS, 101 lines)**

1 │ ﻿using Microsoft.EntityFrameworkCore;

2 │ using QuanLyBanHang.Common;

3 │ using QuanLyBanHang.Models.ExModel.StoredProcedureModel;

4 │ using QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ namespace QuanLyBanHang.Models.ExModel.OPS

7 │ {

8 │ public class NgayKetCaModel : Base<TblNgayKetCa>

9 │ {

10 │ public NgayKetCaModel(DbContext dbContext) : base(dbContext)

11 │ {

12 │ }

13 │ QuanLyBanHangContext dbContext

14 │ {

15 │ get { return \_dbContext as QuanLyBanHangContext; }

16 │ set { return; }

17 │ }

18 │ public DateTime LayNgayKetCa(int MaKho)

19 │ {

20 │ var kc = First(t => t.MaKho == MaKho && !(t.Deleted ?? false));

21 │ DateTime dt = DateTime.Now;

22 │ return kc == null ? dt : (kc.NgayBanHang ?? dt);

23 │ }

24 │ public int KetCa(int MaKho, DateTime NgayKetCa, Guid UserID, ref decimal SoTien)

25 │ {

26 │ SoTien = 0;

27 │ var drKetCa = dbContext.TblNgayKetCa.Where(t => t.MaKho == MaKho && !(t.Deleted ?? false)).FirstOrDefault();

28 │ if (drKetCa == null)

29 │ {

30 │ var itemAdd = new TblNgayKetCa

31 │ {

32 │ MaKho = MaKho,

33 │ NgayBanHang = NgayKetCa,

34 │ UserCreated = UserID,

35 │ CreatedDate = DateTime.Now,

36 │ };

37 │ if (Add(itemAdd) > 0)

38 │ {

39 │ new TonKhoModel(dbContext).TinhTonKhoBanHang(Cast.ToInt(MaKho), string.Format("{0:yyyyMM}", NgayKetC...

40 │ SoTien = LuuCTKetCa(MaKho, NgayKetCa, UserID);

41 │ return 1;

42 │ }

43 │

44 │ return 0;

45 │ }

46 │ else

47 │ {

48 │ DateTime? NgayBanHangHT = drKetCa.NgayBanHang;

49 │ if (NgayBanHangHT != NgayKetCa)

50 │ {

51 │ drKetCa.NgayBanHang = NgayKetCa;

52 │ drKetCa.UserModified = UserID;

53 │ drKetCa.ModifiedDate = DateTime.Now;

54 │ if (Update(drKetCa) > 0)

55 │ {

56 │ if (NgayBanHangHT != null)

57 │ {

58 │ if (NgayBanHangHT.Value.Month != NgayKetCa.Month)

59 │ {

60 │

61 │ new TonKhoModel(dbContext).TinhTonKhoBanHang(Cast.ToInt(MaKho), string.Format("{0:yyyyMM...

62 │ new TonKhoModel(dbContext).TinhTonKhoBanHang(Cast.ToInt(MaKho), string.Format("{0:yyyyMM...

63 │ }

64 │ }

65 │ SoTien = LuuCTKetCa(MaKho, NgayBanHangHT.Value, UserID);

66 │ return 1;

67 │ }

68 │ }

69 │ else

70 │ {

71 │ new TonKhoModel(dbContext).TinhTonKhoBanHang(Cast.ToInt(MaKho), string.Format("{0:yyyyMM}", NgayKetC...

72 │ SoTien = LuuCTKetCa(MaKho, NgayBanHangHT.Value, UserID);

73 │ return 1;

74 │ }

75 │

76 │ return 0;

77 │ }

78 │ }

79 │ public decimal LuuCTKetCa(int MaKho, DateTime NgayBan, Guid UserId)

80 │ {

81 │ List<sp\_BaoCaoDoanhThuTheoNguoiBanTheoCa> lstData = new BaoCaoModel(dbContext)

82 │ .BaoCaoDoanhThuTheoNguoiBanTheoCa(NgayBan, NgayBan, UserId, "", MaKho);

83 │ decimal SoTien = 0;

84 │ if (lstData != null)

85 │ {

86 │ SoTien = lstData.Sum(t => t.TongCong);

87 │ }

88 │ var item = new TblChiTietKetCa

89 │ {

90 │ MaKho = MaKho,

91 │ NgayBan = NgayBan.Date,

92 │ NgayKetCa = DateTime.Now,

93 │ MaUser = UserId,

94 │ SoTien = SoTien

95 │ };

96 │ dbContext.TblChiTietKetCa.AddRange(item);

97 │ dbContext.SaveChanges();

98 │ return item.SoTien ?? 0;

99 │ }

100 │ }

101 │ }

────────────────────────────────────────────────────────────

**📄 NguoiDung.cs (CS, 45 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class NguoiDung

7 │ {

8 │ public int NId { get; set; }

9 │

10 │ public string Tendangnhap { get; set; } = null!;

11 │

12 │ public string Hoten { get; set; } = null!;

13 │

14 │ public string Matkhau { get; set; } = null!;

15 │

16 │ public string? DepartmentId { get; set; }

17 │

18 │ public string? Anh { get; set; }

19 │

20 │ public string? KhoHang { get; set; }

21 │

22 │ public bool? TrangThai { get; set; }

23 │

24 │ public int? NguoiTao { get; set; }

25 │

26 │ public DateTime NgayTao { get; set; }

27 │

28 │ public int? Quyen { get; set; }

29 │

30 │ public byte[]? Photo { get; set; }

31 │

32 │ public string? Image { get; set; }

33 │

34 │ public int? NguoiSua { get; set; }

35 │

36 │ public DateTime? NgaySua { get; set; }

37 │

38 │ public bool? DaXoa { get; set; }

39 │

40 │ public int? NguoiXoa { get; set; }

41 │

42 │ public DateTime? NgayXoa { get; set; }

43 │

44 │ public string? Msnv { get; set; }

45 │ }

────────────────────────────────────────────────────────────

**📄 NguoiDungModel.cs (CS, 49 lines)**

1 │ ﻿using Microsoft.EntityFrameworkCore;

2 │ using QuanLyBanHang.Models.QLBHEntity;

3 │

4 │ namespace QuanLyBanHang.Models.ExModel.SYS

5 │ {

6 │ public class NguoiDungModel : Base<SysUser>

7 │ {

8 │ public NguoiDungModel(DbContext dbContext) : base(dbContext)

9 │ {

10 │ }

11 │

12 │ QuanLyBanHangContext dbContext

13 │ {

14 │ get { return \_dbContext as QuanLyBanHangContext; }

15 │ set { return; }

16 │ }

17 │

18 │ public SysUser GetItem(Guid? UserID)

19 │ {

20 │ return First(t => t.Ma == UserID && !(t.Deleted??false));

21 │ }

22 │

23 │ public List<SysUser> GetData()

24 │ {

25 │ return Get(t => !(t.Deleted ?? false)).OrderByDescending(t => t.Ma).ToList();

26 │ }

27 │

28 │ public List<SysUser> GetDataByIds(List<Guid> userIds)

29 │ {

30 │ return dbContext.SysUser

31 │ .Where(u => userIds.Contains(u.Ma))

32 │ .ToList();

33 │ }

34 │

35 │ public SysUser GetByCreatedId(Guid id)

36 │ {

37 │ return dbContext.SysUser

38 │ .AsNoTracking() // Tối ưu hiệu suất cho truy vấn chỉ đọc

39 │ .First(u => u.UserCreated == id);

40 │ }

41 │

42 │ public SysUser GetByModifiedId(Guid id)

43 │ {

44 │ return dbContext.SysUser

45 │ .AsNoTracking() // Tối ưu hiệu suất cho truy vấn chỉ đọc

46 │ .First(u => u.UserModified == id);

47 │ }

48 │ }

49 │ }

────────────────────────────────────────────────────────────

**📄 NhaCCController.cs (CS, 225 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using QuanLyBanHang.Common;

3 │ using QuanLyBanHang.Common.Attributes;

4 │ using QuanLyBanHang.Models.APIModels;

5 │ using QuanLyBanHang.Models.QLBHEntity;

6 │

7 │ namespace QuanLyBanHang.Controllers.CAT

8 │ {

9 │ public class NhaCCController : BaseController

10 │ {

11 │ private readonly ILogger<NhaCCController> \_logger;

12 │

13 │ public NhaCCController(

14 │ QuanLyBanHangContext dbcontext,

15 │ MemoryCacheManager memoryCache,

16 │ ILogger<NhaCCController> logger) : base(dbcontext, memoryCache)

17 │ {

18 │ m\_MenuID = MenuID.catNhaCC.GetEnumGuid();

19 │ \_logger = logger;

20 │ }

21 │

22 │ [TokenUserCheckHTTP]

23 │ public IActionResult Index()

24 │ {

25 │ \_logger.LogInformation("Người dùng {UserId} truy cập trang danh sách Nhà cung cấp.", objToken.user...

26 │ return View();

27 │ }

28 │

29 │ [TokenUserCheckHTTP(Permission.Add)]

30 │ public ActionResult Add()

31 │ {

32 │ \_logger.LogInformation("Người dùng {UserId} truy cập trang thêm mới Nhà cung cấp.", objToken.use...

33 │ return View("Form");

34 │ }

35 │

36 │ [TokenUserCheckHTTP(Permission.Edit)]

37 │ public ActionResult Edit(string Ma)

38 │ {

39 │ \_logger.LogInformation("Người dùng {UserId} yêu cầu chỉnh sửa Nhà cung cấp. Mã (đã mã hóa...

40 │ try

41 │ {

42 │ Ma = Ma.Decrypt();

43 │ int id = Ma.ToInt();

44 │ ViewData["item"] = dbcontext.TblNhaCc.FirstOrDefault(t => t.Ma == id && !(t.Deleted ?? false));

45 │ return View("Form");

46 │ }

47 │ catch (Exception ex)

48 │ {

49 │ \_logger.LogError(ex, "Lỗi khi tải dữ liệu để chỉnh sửa Nhà cung cấp. Mã (đã mã hó...

50 │ return RedirectToAction("Index");

51 │ }

52 │ }

53 │

54 │ [HttpPost]

55 │ [TokenUserCheckHTTP(Permission.Edit)] // Gộp cả Add và Edit

56 │ public ActionResult UpdateAPI(string Ma, string Ten, string TrangThai, string? MaKt, string? DiaChi, string? Die...

57 │ {

58 │ \_logger.LogInformation("Người dùng {UserId} bắt đầu xử lý lưu (Thêm/Sửa) Nhà cung cấp. M�...

59 │ try

60 │ {

61 │ Ma = Ma.Decrypt();

62 │ var result = CheckRequire(Ten, TrangThai);

63 │ if (result != null)

64 │ return Json(result);

65 │

66 │ var br = dbcontext.SysBranch.FirstOrDefault(x => x.Ma == objToken.branch);

67 │ if (br == null)

68 │ {

69 │ \_logger.LogWarning("Không tìm thấy chi nhánh {MaChiNhanh} khi đang lưu Nhà cung cấp.", obj...

70 │ return Json(new ResponseModelError("Không tìm thấy chi nhánh"));

71 │ }

72 │

73 │ if (Ma.Equals("0"))

74 │ {

75 │ \_logger.LogInformation("Thêm mới Nhà cung cấp: {TenNCC}", Ten);

76 │ var item = new TblNhaCc

77 │ {

78 │ TenNhaCc = Ten,

79 │ Status = TrangThai == "1",

80 │ MaKt = MaKt,

81 │ DiaChi = DiaChi,

82 │ DienThoai = DienThoai,

83 │ Mst = Mst,

84 │ Email = Email,

85 │ NguoiLienHe = NguoiLienHe,

86 │ Cccd = Cccd,

87 │ MaChiNhanh = objToken.branch,

88 │ MaChiNhanhInt = br.Nid ?? 0,

89 │ UserCreated = objToken.userid,

90 │ CreatedDate = DateTime.Now,

91 │ };

92 │ dbcontext.TblNhaCc.Add(item);

93 │ dbcontext.SaveChanges();

94 │ \_logger.LogInformation("Thêm mới Nhà cung cấp ID {NCCId} thành công.", item.Ma);

95 │ return Json(QLBHConst.ResponseAddSuccessfully);

96 │ }

97 │ else

98 │ {

99 │ int id = Ma.ToInt();

100 │ \_logger.LogInformation("Cập nhật Nhà cung cấp ID: {NCCId}", id);

101 │ var item = dbcontext.TblNhaCc.FirstOrDefault(x => x.Ma == id && !(x.Deleted ?? false));

102 │ if (item == null)

103 │ {

104 │ \_logger.LogWarning("Không tìm thấy Nhà cung cấp ID: {NCCId} để cập nhật.", id);

105 │ return Json(QLBHConst.ResponseNotExist);

106 │ }

107 │

108 │ item.TenNhaCc = Ten; item.Status = TrangThai == "1"; item.MaKt = MaKt;

109 │ item.DiaChi = DiaChi; item.DienThoai = DienThoai; item.Mst = Mst;

110 │ item.Email = Email; item.NguoiLienHe = NguoiLienHe; item.Cccd = Cccd;

111 │ item.MaChiNhanh = objToken.branch; item.UserModified = objToken.userid;

112 │ item.ModifiedDate = DateTime.Now;

113 │

114 │ dbcontext.SaveChanges();

115 │ \_logger.LogInformation("Cập nhật Nhà cung cấp ID {NCCId} thành công.", id);

116 │ return Json(QLBHConst.ResponseUpdateSuccessfully);

117 │ }

118 │ }

119 │ catch (Exception ex)

120 │ {

121 │ \_logger.LogError(ex, "Lỗi khi xử lý lưu Nhà cung cấp. Mã (đã mã hóa): {MaNCC}", Ma);

122 │ return Json(new ResponseModel(QLBHConst.Error, "Lỗi hệ thống khi lưu."));

123 │ }

124 │ }

125 │

126 │ [HttpPost]

127 │ [TokenUserCheckHTTP(Permission.Delete)]

128 │ public IActionResult DeleteAPI(string Ma)

129 │ {

130 │ \_logger.LogInformation("Người dùng {UserId} yêu cầu xóa Nhà cung cấp. Mã (đã mã hóa): {MaNCC...

131 │ try

132 │ {

133 │ Ma = Ma.Decrypt();

134 │ int id = Ma.ToInt();

135 │ var item = dbcontext.TblNhaCc.FirstOrDefault(x => x.Ma == id);

136 │ if (item == null)

137 │ {

138 │ \_logger.LogWarning("Không tìm thấy Nhà cung cấp ID: {NCCId} để xóa.", id);

139 │ return Json(QLBHConst.ResponseNotExist);

140 │ }

141 │

142 │ item.Deleted = true;

143 │ item.UserDeleted = objToken.userid;

144 │ item.DeletedDate = DateTime.Now;

145 │ dbcontext.SaveChanges();

146 │

147 │ \_logger.LogInformation("Đã xóa (đánh dấu) Nhà cung cấp ID {NCCId} thành công.", id);

148 │ return Json(QLBHConst.ResponseDeleteSuccessfully);

149 │ }

150 │ catch (Exception ex)

151 │ {

152 │ \_logger.LogError(ex, "Lỗi khi xóa Nhà cung cấp. Mã (đã mã hóa): {MaNCC}", Ma);

153 │ return Json(new ResponseModel(QLBHConst.Error, "Lỗi hệ thống khi xóa."));

154 │ }

155 │ }

156 │

157 │ private static ResponseModel CheckRequire(String Ten, String Status)

158 │ {

159 │ if (Ten.IsEmpty())

160 │ return new ResponseModel(QLBHConst.Error, "Vui lòng nhập tên");

161 │ if (Status.IsEmpty())

162 │ return new ResponseModel(QLBHConst.Error, "Vui lòng nhập trạng thái");

163 │ return null;

164 │ }

165 │

166 │ [TokenUserCheckHTTP(Permission.NotCheck)]

167 │ public IActionResult getListData()

168 │ {

169 │ \_logger.LogInformation("Người dùng {UserId} yêu cầu lấy danh sách Nhà cung cấp.", objToken.user...

170 │ try

171 │ {

172 │ var listData = dbcontext.TblNhaCc.Where(x => !(x.Deleted ?? false) && x.MaChiNhanh == objToken.branch)

173 │ .Select(n => new

174 │ {

175 │ maEncrypt = n.Ma.Encrypt(),

176 │ n.Ma,

177 │ n.MaKt,

178 │ n.TenNhaCc,

179 │ n.DiaChi,

180 │ n.DienThoai,

181 │ n.Email,

182 │ n.NguoiLienHe,

183 │ n.Cccd,

184 │ trangthai = n.Status ?? false,

185 │ tentrangthai = n.Status ?? false ? "Hoạt động" : "Hết hoạt động",

186 │ }).OrderBy(t => t.Ma).ToList();

187 │

188 │ \_logger.LogInformation("Lấy thành công {Count} bản ghi Nhà cung cấp.", listData.Count);

189 │ return Json(new ResponseModel(QLBHConst.OK, "", listData));

190 │ }

191 │ catch (Exception ex)

192 │ {

193 │ \_logger.LogError(ex, "Lỗi khi lấy danh sách Nhà cung cấp cho chi nhánh {MaChiNhanh}", objToken....

194 │ return Json(new ResponseModel(QLBHConst.Error, "Lỗi hệ thống khi lấy danh sách."));

195 │ }

196 │ }

197 │

198 │ [HttpPost]

199 │ [TokenUserCheckHTTP(Permission.NotCheck)]

200 │ public IActionResult getListNhaCungCap()

201 │ {

202 │ \_logger.LogInformation("Người dùng {UserId} yêu cầu lấy danh sách Nhà cung cấp (rút gọn).", ...

203 │ try

204 │ {

205 │ var listData = dbcontext.TblNhaCc.Where(x => !(x.Deleted ?? false) && x.MaChiNhanh == objToken.branch)

206 │ .Select(t => new

207 │ {

208 │ ma = t.Ma + "",

209 │ code = t.MaKt + "",

210 │ ten = t.TenNhaCc + "",

211 │ nguoilienhe = t.NguoiLienHe + "",

212 │ sodienthoainlh = t.DienThoai + "",

213 │ diachi = t.DiaChi + "",

214 │ }).ToList();

215 │

216 │ return Json(new ResponseModelSuccess("", listData));

217 │ }

218 │ catch (Exception ex)

219 │ {

220 │ \_logger.LogError(ex, "Lỗi khi lấy danh sách Nhà cung cấp (rút gọn) cho chi nhánh {MaChiNhanh...

221 │ return Json(new ResponseModel(QLBHConst.Error, "Lỗi hệ thống khi lấy danh sách."));

222 │ }

223 │ }

224 │ }

225 │ }

────────────────────────────────────────────────────────────

**📄 NhaCCModel.cs (CS, 27 lines)**

1 │ ﻿using Microsoft.EntityFrameworkCore;

2 │ using QuanLyBanHang.Models.QLBHEntity;

3 │

4 │ namespace QuanLyBanHang.Models.ExModel.CAT

5 │ {

6 │ public class NhaCCModel : Base<TblNhaCc>

7 │ {

8 │ public NhaCCModel(DbContext dbContext) : base(dbContext)

9 │ {

10 │ }

11 │ QuanLyBanHangContext dbContext

12 │ {

13 │ get { return \_dbContext as QuanLyBanHangContext; }

14 │ set { return; }

15 │ }

16 │ public TblNhaCc GetItem(int? Ma)

17 │ {

18 │ return First(t => t.Ma == Ma && !(t.Deleted ?? false));

19 │ }

20 │

21 │ public List<TblNhaCc> GetData(Guid brandId)

22 │ {

23 │ return Get(t => !(t.Deleted ?? false) && t.MaChiNhanh == brandId).OrderBy(t => t.TenNhaCc).ToList();

24 │ }

25 │

26 │ }

27 │ }

────────────────────────────────────────────────────────────

**📄 NhaCc.cs (CS, 41 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class NhaCc

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string? MaKt { get; set; }

11 │

12 │ public string TenNhaCc { get; set; } = null!;

13 │

14 │ public string? DiaChi { get; set; }

15 │

16 │ public string? DienThoai { get; set; }

17 │

18 │ public string? Mst { get; set; }

19 │

20 │ public string? Email { get; set; }

21 │

22 │ public string? NguoiLienHe { get; set; }

23 │

24 │ public string? Cccd { get; set; }

25 │

26 │ public bool? TrangThai { get; set; }

27 │

28 │ public DateTime NgayTao { get; set; }

29 │

30 │ public int NguoiTao { get; set; }

31 │

32 │ public DateTime? NgaySua { get; set; }

33 │

34 │ public int? NguoiSua { get; set; }

35 │

36 │ public bool? DaXoa { get; set; }

37 │

38 │ public int? NguoiXoa { get; set; }

39 │

40 │ public DateTime? NgayXoa { get; set; }

41 │ }

────────────────────────────────────────────────────────────

**📄 NhapHang.cs (CS, 47 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class NhapHang

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string? SoHd { get; set; }

11 │

12 │ public DateTime? NgayHoaDon { get; set; }

13 │

14 │ public DateTime NgayNhapHang { get; set; }

15 │

16 │ public int NhaCc { get; set; }

17 │

18 │ public decimal? TienChietKhau { get; set; }

19 │

20 │ public int? ThueVat { get; set; }

21 │

22 │ public int? MaKho { get; set; }

23 │

24 │ public int? MaLoaiNx { get; set; }

25 │

26 │ public string? GhiChu { get; set; }

27 │

28 │ public bool? KhongHoaDon { get; set; }

29 │

30 │ public int? NguoiTao { get; set; }

31 │

32 │ public DateTime? NgayTao { get; set; }

33 │

34 │ public int? NguoiSua { get; set; }

35 │

36 │ public DateTime? NgaySua { get; set; }

37 │

38 │ public bool? KhoaSo { get; set; }

39 │

40 │ public bool? DaXoa { get; set; }

41 │

42 │ public int? NguoiXoa { get; set; }

43 │

44 │ public DateTime? NgayXoa { get; set; }

45 │

46 │ public string? SoHoaDonNcc { get; set; }

47 │ }

────────────────────────────────────────────────────────────

**📄 NhapHangController.cs (CS, 519 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using QuanLyBanHang.Common;

3 │ using QuanLyBanHang.Common.Attributes;

4 │ using QuanLyBanHang.Models.APIModels;

5 │ using QuanLyBanHang.Models.ExModel.CAT;

6 │ using QuanLyBanHang.Models.ExModel.OPS;

7 │ using QuanLyBanHang.Models.ExModel.SYS;

8 │ using QuanLyBanHang.Models.QLBHEntity;

9 │ using System.Data;

10 │

11 │ namespace QuanLyBanHang.Controllers.OPS

12 │ {

13 │ public class NhapHangController : BaseController

14 │ {

15 │ private readonly IWebHostEnvironment \_hostingEnvironment;

16 │

17 │ public NhapHangController(QuanLyBanHangContext dbcontext, MemoryCacheManager memoryCache, IWebHostEnvironment ho...

18 │ {

19 │ m\_MenuID = MenuID.opsNhapHang.GetEnumGuid();

20 │ \_hostingEnvironment = hostingEnvironment;

21 │

22 │ }

23 │

24 │ public class cNhapHang

25 │ {

26 │ public string Ma { get; set; }

27 │ public string SoHD { get; set; }

28 │ public string Ngaynhaphang { get; set; }

29 │ public string NhaCC { get; set; }

30 │ public string Tienchietkhau { get; set; }

31 │ public string ThueVAT { get; set; }

32 │ public string GhiChu { get; set; }

33 │ public string Result { get; set; }

34 │ public string ResultText { get; set; }

35 │ }

36 │

37 │ public class cCT\_NhapHang

38 │ {

39 │ public string Ma { get; set; }

40 │ public int MaHangHoa { get; set; }

41 │ public double SoLuong { get; set; }

42 │ public double GiaVon { get; set; }

43 │ public string HanSD { get; set; }

44 │ public double? ChietKhauTM { get; set; }

45 │ public double? ChietKhauPT { get; set; }

46 │ public int VAT { get; set; }

47 │ }

48 │

49 │ [TokenUserCheckHTTP]

50 │ public IActionResult Index()

51 │ {

52 │ ViewData["ListMaNCC"] = new NhaCCModel(dbcontext).GetData(objToken.branch).Where(t => t.Status == true).ToLi...

53 │ ViewData["ListKho"] = new DanhMucKhoModel(dbcontext).GetData(objToken.branch);

54 │ return View();

55 │ }

56 │

57 │ [TokenUserCheckHTTP(Permission.Add)]

58 │ public ActionResult Add()

59 │ {

60 │ ViewData["ListHangHoa"] = new TblHangHoaModel(dbcontext).LayDanhSachTblHangHoa(objToken.branch).ToList();

61 │ ViewData["ListVAT"] = dbcontext.TblThueVat

62 │ .Where(t => !(t.Deleted ?? false))

63 │ .OrderBy(x => x.GiaTri)

64 │ .ToList();

65 │ ViewData["ListDVT"] = dbcontext.TblDonViTinh.Where(t => !(t.Deleted ?? false)).ToList();

66 │ ViewData["ListNCC"] = dbcontext.TblNhaCc.Where(t => !(t.Deleted ?? false) && t.MaChiNhanh == objToken.branch...

67 │ ViewData["ListKhoNhap"] = new DanhMucKhoModel(dbcontext).GetDataByUser(objToken.userid, objToken.branch)

68 │ .Where(t => (t.Status ?? false) && (t.NhapHang ?? false)).ToList();

69 │ return View("Form");

70 │ }

71 │

72 │ [TokenUserCheckHTTP(Permission.Edit)]

73 │ public ActionResult Edit(string Ma)

74 │ {

75 │ Ma = Ma.Decrypt();

76 │ VNhapHang nh = new NhapHangModel(dbcontext).LayPhieuNhapHang(Cast.ToInt(Ma), objToken.branch);

77 │ if (nh == null)

78 │ {

79 │ return Json(QLBHConst.ResponseNotExist);

80 │ }

81 │ List<VChiTietNhapHang> ct = new vChiTietNhapHangModel(dbcontext).GetData(nh.Ma);

82 │

83 │ ViewData["item"] = nh;

84 │ ViewData["ListChiTiet"] = ct ?? new List<VChiTietNhapHang>();

85 │ ViewData["ListVAT"] = dbcontext.TblThueVat

86 │ .Where(t => !(t.Deleted ?? false))

87 │ .OrderBy(x => x.GiaTri)

88 │ .ToList();

89 │ ViewData["ListDVT"] = dbcontext.TblDonViTinh.Where(t => !(t.Deleted ?? false)).ToList();

90 │ ViewData["ListNCC"] = dbcontext.TblNhaCc.Where(t => !(t.Deleted ?? false) && t.MaChiNhanh == objToken.branch...

91 │ ViewData["ListKhoNhap"] = new DanhMucKhoModel(dbcontext).GetDataByUser(objToken.userid, objToken.branch)

92 │ .Where(t => (t.Status ?? false) && (t.NhapHang ?? false)).ToList();

93 │ ViewData["ListHangHoa"] = new TblHangHoaModel(dbcontext).LayDanhSachTblHangHoa(objToken.branch).ToList();

94 │ if ((nh.ThueVat ?? 0) != 0 || (nh.TienChietKhau ?? 0) != 0)

95 │ {

96 │ double tongthanhtien = (double)((ct.Sum(t => t.ThanhTienSauCk) ?? 0) - (nh.TienChietKhau ?? 0));

97 │ double tongthanhtienVAT = tongthanhtien \* (double)(nh.ThueVat) / 100.0;

98 │ ViewData["TongThueVAT"] = Math.Round(tongthanhtienVAT, 3);

99 │ double tongthanhtoan = Math.Round(tongthanhtien, 3) + Math.Round(tongthanhtienVAT, 3);

100 │ ViewData["TongThanhToan"] = tongthanhtoan;

101 │ }

102 │ else

103 │ {

104 │ ViewData["TongThueVAT"] = (ct.Sum(t => t.ThanhTienSauVat) ?? 0) - (ct.Sum(t => t.ThanhTienSauCk) ?? 0);

105 │ double tongthanhtoan = ((double?)ct.Sum(t => t.ThanhTienSauVat) ?? 0.0);

106 │ ViewData["TongThanhToan"] = tongthanhtoan;

107 │ }

108 │ return View("Form");

109 │ }

110 │

111 │ [HttpPost]

112 │ [TokenUserCheckHTTP]

113 │ public ActionResult Update(string Ma, string NgayNhapHang, string SoHD, string NhaCC, string MaKho, string TienC...

114 │ , string ThueVAT, string GhiChu, string NgayHoaDon, string KhongHoaDon

115 │ , List<cCT\_NhapHang> ListChiTiet, string SoHoaDonNCC)

116 │ {

117 │ int \_Ma = Cast.ToInt(Ma.Decrypt());

118 │

119 │ if (ListChiTiet == null)

120 │ {

121 │ ListChiTiet = new List<cCT\_NhapHang>();

122 │ }

123 │

124 │ if (String.IsNullOrEmpty(NhaCC))

125 │ {

126 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng chọn nhà cung cấp"));

127 │ }

128 │

129 │ if (String.IsNullOrEmpty(MaKho))

130 │ {

131 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng chọn kho nhập"));

132 │ }

133 │

134 │ if (TienChietKhau != "")

135 │ {

136 │ int tienck = 0;

137 │ if (!int.TryParse(TienChietKhau, out tienck))

138 │ {

139 │ return Json(new ResponseModel(QLBHConst.Error, "Tiền chiết khấu không hợp lệ"));

140 │ }

141 │ }

142 │ else

143 │ {

144 │ TienChietKhau = "0";

145 │ }

146 │

147 │ if (ThueVAT == "")

148 │ {

149 │ ThueVAT = "0";

150 │ }

151 │ DateTime dtNgayNhap, dtNgayHoaDon;

152 │ if (String.IsNullOrEmpty(NgayNhapHang))

153 │ {

154 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng nhập ngày nhập hàng"));

155 │ }

156 │ else

157 │ {

158 │ dtNgayNhap = Cast.ToDateTime(NgayNhapHang, "dd/MM/yyyy HH:mm");

159 │ }

160 │ if (String.IsNullOrEmpty(NgayHoaDon))

161 │ {

162 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng nhập ngày hóa đơn"));

163 │ }

164 │ else

165 │ {

166 │ dtNgayHoaDon = Cast.ToDateTime(NgayHoaDon, "dd/MM/yyyy");

167 │ }

168 │

169 │ TblKyKho? drKhoaSo = new KyKhoModel(dbcontext).Get(t => t.Nam == dtNgayNhap.Year

170 │ && t.Thang == dtNgayNhap.Month

171 │ && t.MaKho == Cast.ToInt(MaKho)

172 │ && (t.KhoaSo ?? false)).FirstOrDefault();

173 │

174 │ if (drKhoaSo != null)

175 │ {

176 │ return Json(new ResponseModel(QLBHConst.Error, "Kỳ kho đã khoá sổ"));

177 │ }

178 │

179 │ int ThangTruoc = dtNgayNhap.Month - 1;

180 │ int Nam = dtNgayNhap.Year;

181 │ if (ThangTruoc == 0)

182 │ {

183 │ ThangTruoc = 12;

184 │ Nam -= 1;

185 │ }

186 │ TblKyKho? drKhoaSoThangTruoc = new KyKhoModel(dbcontext).Get(t => t.Nam == Nam

187 │ && t.Thang == ThangTruoc

188 │ && t.MaKho == Cast.ToInt(MaKho)).FirstOrDefault();

189 │

190 │ if (drKhoaSoThangTruoc != null)

191 │ {

192 │ if (!(drKhoaSoThangTruoc.KhoaSo ?? false))

193 │ {

194 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng khóa sổ kỳ trước trước khi thao...

195 │ }

196 │ }

197 │ var kho = dbcontext.StockDanhMucKho.FirstOrDefault(x => x.Ma == Cast.ToInt(MaKho));

198 │ if(kho.MaLoaiKho == (int)LoaiKho.KhoTieuHao)

199 │ {

200 │ return Json(new ResponseModel(QLBHConst.Error, "Không thể nhập hàng vào kho tiêu hao"));

201 │ }

202 │ List<VPermission> ListPermission = GetVPermissions();

203 │

204 │ if (\_Ma == 0 && ListPermission.Any(t => t.PermissionId == Permission.Add.GetEnumGuid()))

205 │ {

206 │ cNhapHang res = new NhapHangModel(dbcontext).Add(SoHD, dtNgayNhap, dtNgayHoaDon, Cast.ToInt(NhaCC), Cast...

207 │ return Json(new ResponseModel(res.Result, res.ResultText));

208 │ }

209 │ else if (\_Ma != 0 && ListPermission.Any(t => t.PermissionId == Permission.Edit.GetEnumGuid()))

210 │ {

211 │ cNhapHang res = new NhapHangModel(dbcontext).Edit(\_Ma, SoHD, dtNgayNhap, dtNgayHoaDon, Cast.ToInt(NhaCC)...

212 │ return Json(new ResponseModel(res.Result, res.ResultText));

213 │ }

214 │ else

215 │ {

216 │ return Json(QLBHConst.ResponseNoPermission);

217 │ }

218 │ }

219 │

220 │ [TokenUserCheckHTTP(Permission.NotCheck)]

221 │ public IActionResult getListData(string TuNgay, string DenNgay, string MaKho, string MaNCC)

222 │ {

223 │ DateTime TuNgayConvert = TuNgay.ToDateTime2(DateTime.Now).Value.Date;

224 │ DateTime DenNgayConvert = DenNgay.ToDateTime2(DateTime.Now).Value.Date;

225 │

226 │ List<VNhapHang> ListNhapKho;

227 │ if (Cast.ToInt(MaNCC) == 0)

228 │ {

229 │ if (Cast.ToInt(MaKho) == 0)

230 │ {

231 │ ListNhapKho = new NhapHangModel(dbcontext).LayDanhSachPhieuNhapHang(TuNgayConvert, DenNgayConvert, o...

232 │ }

233 │ else

234 │ {

235 │ ListNhapKho = new NhapHangModel(dbcontext).LayDanhSachPhieuNhapHangTheoKho(TuNgayConvert, DenNgayCon...

236 │ }

237 │ }

238 │ else

239 │ {

240 │ if (Cast.ToInt(MaKho) == 0)

241 │ {

242 │ ListNhapKho = new NhapHangModel(dbcontext).LayDanhSachPhieuNhapHangTheoNhaCungCap(TuNgayConvert, Den...

243 │ }

244 │ else

245 │ {

246 │ ListNhapKho = new NhapHangModel(dbcontext).LayDanhSachPhieuNhapHangTheoKhoVaNhaCungCap(TuNgayConvert...

247 │ }

248 │ }

249 │ var listData = new

250 │ {

251 │ lstdata = ListNhapKho.Select(t => new

252 │ {

253 │ ma = t.Ma.Encrypt(),

254 │ ordby = string.Format("{0:yyyyMMddHHmm}", t.NgayHoaDon == null ? t.CreatedDate : t.NgayNhapHang),

255 │ ngaynhaphang = string.Format("{0:dd/MM/yyyy HH:mm}", t.NgayHoaDon == null ? t.CreatedDate : t.NgayNh...

256 │ sohd = t.SoHoaDon + "",

257 │ ordby2 = string.Format("{0:yyyyMMdd}", t.NgayHoaDon == null ? t.NgayNhapHang : t.NgayHoaDon),

258 │ ngayhoadon = string.Format("{0:dd/MM/yyyy}", t.NgayHoaDon == null ? t.NgayNhapHang : t.NgayHoaDon),

259 │ tenncc = t.TenNhaCc + "",

260 │ tenkho = t.TenKho + "",

261 │ ghichu = t.GhiChu + "",

262 │ thanhtien = string.Format("{0:#,##0.000}", t.ThanhTien),

263 │ khoaso = (t.KhoaSo ?? false) ? "1" : "0",

264 │ sohoadonncc = t.SoHoaDonNcc + ""

265 │ }).ToList(),

266 │ tongtien = string.Format("{0:#,##0.000}", ListNhapKho.Sum(t => t.ThanhTien ?? 0))

267 │ };

268 │ return Json(new ResponseModel(QLBHConst.OK, "", listData));

269 │ }

270 │

271 │ [HttpPost]

272 │ [TokenUserCheckHTTP(Permission.Delete)]

273 │ public ActionResult Delete(string Ma)

274 │ {

275 │ Ma = Ma.Decrypt();

276 │ int res = new NhapHangModel(dbcontext).Delete(Cast.ToInt(Ma), objToken.userid, objToken.branch);

277 │ if (res == -2)

278 │ {

279 │ return Json(QLBHConst.ResponseNotExist);

280 │ }

281 │ else if (res == -3)

282 │ {

283 │ return Json(new ResponseModel(QLBHConst.Error, "Phiếu nhập kho đã khóa sổ, không thể xóa"))...

284 │ }

285 │ else if (res == 1)

286 │ {

287 │ return Json(QLBHConst.ResponseDeleteSuccessfully);

288 │ }

289 │ else

290 │ {

291 │ return Json(QLBHConst.ResponseHaveError);

292 │ }

293 │ }

294 │

295 │ [HttpPost]

296 │ [TokenUserCheckHTTP]

297 │ public ActionResult KhoaSo(string MaNhapHang, string Value)

298 │ {

299 │ List<VPermission> ListPermission = GetVPermissions();

300 │ if (Value == "1")

301 │ {

302 │ if (ListPermission.Any(t => t.PermissionId == Permission.KhoaSo.GetEnumGuid()))

303 │ {

304 │ int res = new NhapHangModel(dbcontext).KhoaSo(Cast.ToInt(MaNhapHang.Decrypt()), objToken.userid, tru...

305 │ if (res == 1)

306 │ {

307 │ return Json(new ResponseModel(QLBHConst.OK, "Khóa sổ thành công"));

308 │ }

309 │ else if (res == -2)

310 │ {

311 │ return Json(QLBHConst.ResponseNotExist);

312 │ }

313 │ else if (res == -3)

314 │ {

315 │ return Json(new ResponseModel(QLBHConst.Error, "Đã khóa sổ bởi người khác"));

316 │ }

317 │ else

318 │ {

319 │ return Json(QLBHConst.ResponseHaveError);

320 │ }

321 │ }

322 │ else

323 │ {

324 │ return Json(QLBHConst.ResponseNoPermission);

325 │ }

326 │ }

327 │ else

328 │ {

329 │ if (ListPermission.Any(t => t.PermissionId == Permission.MoKhoaSo.GetEnumGuid()))

330 │ {

331 │ int res = new NhapHangModel(dbcontext).KhoaSo(Cast.ToInt(MaNhapHang.Decrypt()), objToken.userid, fal...

332 │ if (res == 1)

333 │ {

334 │ return Json(new ResponseModel(QLBHConst.OK, "Mở khóa sổ thành công"));

335 │ }

336 │ else if (res == -2)

337 │ {

338 │ return Json(QLBHConst.ResponseNotExist);

339 │ }

340 │ else if (res == -3)

341 │ {

342 │ return Json(new ResponseModel(QLBHConst.Error, "Đã mở khóa sổ bởi người khác"));

343 │ }

344 │ else

345 │ {

346 │ return Json(QLBHConst.ResponseHaveError);

347 │ }

348 │ }

349 │ else

350 │ {

351 │ return Json(QLBHConst.ResponseNoPermission);

352 │ }

353 │ }

354 │ }

355 │

356 │ [HttpPost]

357 │ [TokenUserCheckHTTP]

358 │ public ActionResult KhoaSoNhieu(List<string> ListMaNhapHang)

359 │ {

360 │ List<VPermission> ListPermission = GetVPermissions();

361 │ if (ListPermission.Any(t => t.PermissionId == Permission.KhoaSo.GetEnumGuid()))

362 │ {

363 │ List<int> lstMa = ListMaNhapHang.Select(t => Cast.ToInt(t.Decrypt())).Where(t => t > 0).ToList();

364 │ int res = new NhapHangModel(dbcontext).KhoaSoNhieu(lstMa, objToken.userid, objToken.branch);

365 │ if (res == 1)

366 │ {

367 │ return Json(new ResponseModel(QLBHConst.OK, "Khóa sổ thành công"));

368 │

369 │ }

370 │ else

371 │ {

372 │ return Json(new ResponseModel(QLBHConst.OK, "Khóa sổ không thành công. Liên hệ IT!!!"));

373 │ }

374 │ }

375 │ return Json(QLBHConst.ResponseNoPermission);

376 │ }

377 │

378 │ [TokenUserCheckHTTP(Permission.Add)]

379 │ public ActionResult In(string Ma)

380 │ {

381 │ Ma = Ma.Decrypt();

382 │ StockNhapHang nh = new NhapHangModel(dbcontext).GetItem(Cast.ToInt(Ma));

383 │ if (nh == null)

384 │ {

385 │ return Json(QLBHConst.ResponseNotExist);

386 │ }

387 │ List<VChiTietNhapHang> ct = new vChiTietNhapHangModel(dbcontext).GetData(nh.Ma) ?? new List<VChiTietNhapHang...

388 │

389 │ ViewData["nhacc"] = new NhaCCModel(dbcontext).GetItem(nh.MaNhaCc);

390 │ ViewData["nguoinhap"] = new NguoiDungModel(dbcontext).GetItem(nh.UserCreated);

391 │ ViewData["ListChiTiet"] = ct;

392 │

393 │ double tongthanhtienVAT = 0;

394 │ double tongthanhtoan = 0;

395 │ if ((nh.ThueVat ?? 0) != 0)

396 │ {

397 │ double tongthanhtien = (double)((ct.Sum(t => t.ThanhTienSauCk) ?? 0) - (nh.TienChietKhau ?? 0));

398 │ tongthanhtienVAT = tongthanhtien \* (double)(nh.ThueVat) / 100.0;

399 │ tongthanhtoan = Math.Round(tongthanhtien, 3) + Math.Round(tongthanhtienVAT, 3);

400 │ }

401 │ else

402 │ {

403 │ tongthanhtienVAT = (double)((ct.Sum(t => t.ThanhTienSauVat) ?? 0) - (ct.Sum(t => t.ThanhTienSauCk) ?? 0)...

404 │ tongthanhtoan = (double)(ct.Sum(t => t.ThanhTienSauVat) ?? 0);

405 │ }

406 │ nh.TienChietKhau = ct.Sum(t => t.ThanhTien) - ct.Sum(t => t.ThanhTienSauCk);

407 │ ViewData["TongThueVAT"] = Math.Round(tongthanhtienVAT, 3);

408 │ ViewData["TongThanhToan"] = tongthanhtoan;

409 │ ViewData["TienBangChu"] = Common.Utility.NumberToText(tongthanhtoan);

410 │

411 │ ViewData["item"] = nh;

412 │ ViewData["itemKho"] = new DanhMucKhoModel(dbcontext).GetItem(nh.MaKho ?? 0);

413 │ return View();

414 │ }

415 │

416 │ [TokenUserCheckHTTP]

417 │ public ActionResult Detail(string Ma)

418 │ {

419 │ Ma = Ma.Decrypt();

420 │ VNhapHang nh = new NhapHangModel(dbcontext).LayPhieuNhapHang(Cast.ToInt(Ma), objToken.branch);

421 │ if (nh == null)

422 │ {

423 │ return Json(QLBHConst.ResponseNotExist);

424 │ }

425 │ List<VChiTietNhapHang> ct = new vChiTietNhapHangModel(dbcontext).GetData(nh.Ma);

426 │ ViewData["item"] = nh;

427 │ ViewData["ListChiTiet"] = ct ?? new List<VChiTietNhapHang>();

428 │ ViewData["ListNCC"] = new NhaCCModel(dbcontext).GetData(objToken.branch);

429 │ ViewData["ListHangHoa"] = new TblHangHoaModel(dbcontext).LayDanhSachTblHangHoa(objToken.branch);

430 │ ViewData["ListVAT"] = new ThueVATModel(dbcontext).GetData();

431 │ ViewData["ListDVT"] = new DonViTinhModel(dbcontext).GetData();

432 │ ViewData["ListKhoNhap"] = new DanhMucKhoModel(dbcontext).GetDataByUser(objToken.userid, objToken.branch)

433 │ .Where(t => (t.Status ?? false) && (t.NhapHang ?? false)).ToList();

434 │ ViewData["ViewOnly"] = true;

435 │ if ((nh.ThueVat ?? 0) != 0 || (nh.TienChietKhau ?? 0) != 0)

436 │ {

437 │ double tongthanhtien = (double)((ct.Sum(t => t.ThanhTienSauCk) ?? 0) - (nh.TienChietKhau ?? 0));

438 │ double tongthanhtienVAT = tongthanhtien \* (double)(nh.ThueVat) / 100.0;

439 │ ViewData["TongThueVAT"] = Math.Round(tongthanhtienVAT, 3);

440 │ double tongthanhtoan = Math.Round(tongthanhtien, 3) + Math.Round(tongthanhtienVAT, 3);

441 │ ViewData["TongThanhToan"] = tongthanhtoan;

442 │ }

443 │ else

444 │ {

445 │ ViewData["TongThueVAT"] = (ct.Sum(t => t.ThanhTienSauVat) ?? 0) - (ct.Sum(t => t.ThanhTienSauCk) ?? 0);

446 │ double tongthanhtoan = ((double?)ct.Sum(t => t.ThanhTienSauVat)) ?? 0.0;

447 │ ViewData["TongThanhToan"] = tongthanhtoan;

448 │ }

449 │ return View("Form");

450 │ }

451 │

452 │ [TokenUserCheckHTTP(Permission.NotCheck)]

453 │ public IActionResult ExportExcel(string TuNgay, string DenNgay, string MaKho, string MaNCC, string FileType)

454 │ {

455 │ DateTime TuNgayConvert = TuNgay.ToDateTime2(DateTime.Now).Value.Date;

456 │ DateTime DenNgayConvert = DenNgay.ToDateTime2(DateTime.Now).Value.Date;

457 │

458 │ var listData = new ChiTietNhapHangModel(dbcontext).GetListChiTietNhapHang(objToken.branch, TuNgayConvert, De...

459 │

460 │ DataTable dtb = new();

461 │ dtb.Columns.Add("Ngày nhập hàng");

462 │ dtb.Columns.Add("Số hóa đơn");

463 │ dtb.Columns.Add("Ngày hóa đơn");

464 │ dtb.Columns.Add("Tên nhà cung cấp");

465 │ dtb.Columns.Add("Tên kho");

466 │ dtb.Columns.Add("Lớp hàng");

467 │ dtb.Columns.Add("Mã hàng");

468 │ dtb.Columns.Add("Mã vạch");

469 │ dtb.Columns.Add("Tên hàng hóa");

470 │ dtb.Columns.Add("ĐVT");

471 │ dtb.Columns.Add("Số lượng", typeof(decimal));

472 │ dtb.Columns.Add("Đơn giá", typeof(decimal));

473 │ dtb.Columns.Add("Chiếc khấu", typeof(decimal));

474 │ dtb.Columns.Add("Thành tiền", typeof(decimal));

475 │ dtb.Columns.Add("VAT", typeof(decimal));

476 │ dtb.Columns.Add("Thanh toán", typeof(decimal));

477 │ dtb.Columns.Add("Ghi chú");

478 │

479 │ foreach (var item in listData)

480 │ {

481 │ dtb.Rows.Add(new object[]

482 │ {

483 │ string.Format("'{0:dd/MM/yyyy}", item.NgayHoaDon == null ? item.CreatedDate : item.NgayNhapHang),

484 │ item.SoHoaDon ?? "",

485 │ string.Format("'{0:dd/MM/yyyy}", item.NgayHoaDon == null ? item.NgayNhapHang : item.NgayHoaDon),

486 │ item.TenNhaCc ?? "",

487 │ item.TenKho ?? "",

488 │ item.TenLopHang ?? "",

489 │ item.MaKt ?? "",

490 │ item.MaVach ?? "",

491 │ item.TenHangHoa ?? "",

492 │ item.TenDvt ?? "",

493 │ item.SoLuong,

494 │ item.GiaVon,

495 │ item.ChietKhauTm + (item.ChietKhauPt / 100 \* (item.SoLuong \* item.GiaVon)),

496 │ item.ThanhTien,

497 │ item.Vat ?? 0,

498 │ item.ThanhTienSauVat,

499 │ item.GhiChu ?? ""

500 │ });

501 │ }

502 │

503 │ string fileName = FileType == "Excel" ? "BaoCaoNhapHang.xlsx" : "BaoCaoNhapHang.pdf";

504 │ string contentType = FileType == "Excel" ? "application/ms-excel" : "application/pdf";

505 │

506 │ string fileTemplate = Path.Combine(\_hostingEnvironment.ContentRootPath, "wwwroot", "templates", "BaoCaoNhapH...

507 │ List<ExcelValue> ListGiaTriRiengLe = new()

508 │ {

509 │ new() { Range = $"A2:{ExportToExcel.GetExcelColumnName(dtb.Columns.Count)}2", Value = $"Từ ngày {TuNg...

510 │ };

511 │ int DongBatDau = 5;

512 │ Stream x = ExportToExcel.FromTemplate(fileTemplate, FileType, ListGiaTriRiengLe, dtb, true, DongBatDau, "A",...

513 │ return new FileStreamResult(x, contentType)

514 │ {

515 │ FileDownloadName = fileName

516 │ };

517 │ }

518 │ }

519 │ }

────────────────────────────────────────────────────────────

**📄 NhapHangModel.cs (CS, 352 lines)**

1 │ ﻿using Microsoft.EntityFrameworkCore;

2 │ using QuanLyBanHang.Common;

3 │ using QuanLyBanHang.Models.ExModel.CAT;

4 │ using QuanLyBanHang.Models.QLBHEntity;

5 │ using QuanLyBanHang.Services;

6 │ using static QuanLyBanHang.Controllers.OPS.NhapHangController;

7 │

8 │ namespace QuanLyBanHang.Models.ExModel.OPS

9 │ {

10 │

11 │ public class NhapHangModel : Base<StockNhapHang>

12 │ {

13 │

14 │ public NhapHangModel(DbContext dbContext) : base(dbContext)

15 │ {

16 │ }

17 │

18 │ QuanLyBanHangContext dbContext

19 │ {

20 │ get { return \_dbContext as QuanLyBanHangContext; }

21 │ set { return; }

22 │ }

23 │

24 │ public StockNhapHang GetItem(int Ma)

25 │ {

26 │ return First(t => t.Ma == Ma && !(t.Deleted??false));

27 │ }

28 │ public List<StockNhapHang> GetData(DateTime TuNgay, DateTime DenNgay, Guid BrandId)

29 │ {

30 │ return Get(t => t.NgayNhapHang.Value.Date >= TuNgay.Date && t.NgayNhapHang.Value.Date <= DenNgay.Date && !(t...

31 │ }

32 │ public List<StockNhapHang> GetData(DateTime TuNgay, DateTime DenNgay, int MaNhaCC, Guid BrandId)

33 │ {

34 │ return Get(t => t.NgayNhapHang.Value.Date >= TuNgay.Date && t.NgayNhapHang.Value.Date <= DenNgay.Date && t.M...

35 │ }

36 │ public List<VNhapHang> LayDanhSachPhieuNhapHang(DateTime TuNgay, DateTime DenNgay, Guid BrandId)

37 │ {

38 │ return dbContext.VNhapHang.Where(t => t.NgayNhapHang.Value.Date >= TuNgay.Date && t.NgayNhapHang.Value.Date ...

39 │ }

40 │ public List<VNhapHang> LayDanhSachPhieuNhapHangTheoNhaCungCap(DateTime TuNgay, DateTime DenNgay, int MaNhaCC, Gu...

41 │ {

42 │ return dbContext.VNhapHang.Where(t => t.NgayNhapHang.Value.Date >= TuNgay.Date && t.NgayNhapHang.Value.Date ...

43 │ }

44 │ public List<VNhapHang> LayDanhSachPhieuNhapHangTheoKho(DateTime TuNgay, DateTime DenNgay, int MaKho, Guid BrandI...

45 │ {

46 │ return dbContext.VNhapHang.Where(t => t.NgayNhapHang.Value.Date >= TuNgay.Date && t.NgayNhapHang.Value.Date ...

47 │ }

48 │ public List<VNhapHang> LayDanhSachPhieuNhapHangTheoKhoVaNhaCungCap(DateTime TuNgay, DateTime DenNgay, int MaKho,...

49 │ {

50 │ return dbContext.VNhapHang.Where(t => t.NgayNhapHang.Value.Date >= TuNgay.Date && t.NgayNhapHang.Value.Date ...

51 │ }

52 │ public VNhapHang LayPhieuNhapHang(int Ma, Guid BrandId)

53 │ {

54 │ return dbContext.VNhapHang.First(t => t.Ma == Ma && t.MaChiNhanh == BrandId);

55 │ }

56 │

57 │ public cNhapHang Add(string soHD, DateTime ngayNhapHang, DateTime ngayHoaDon, int nhaCC, int maKho, int tienChie...

58 │ {

59 │ using var transaction = dbContext.Database.BeginTransaction();

60 │ try

61 │ {

62 │

63 │ var ngayTao = DateTime.Now;

64 │ var br = dbContext.SysBranch.FirstOrDefault(x => x.Ma == brandId);

65 │ var nhapHang = new StockNhapHang

66 │ {

67 │ SoHoaDon = soHD,

68 │ MaLoaiNx = (int)LoaiNhapXuat.PhieuNhapTuNCC,

69 │ NgayNhapHang = ngayNhapHang,

70 │ NgayHoaDon = ngayHoaDon,

71 │ MaNhaCc = nhaCC,

72 │ MaKho = maKho,

73 │ TienChietKhau = tienChietKhau,

74 │ ThueVat = thueVAT,

75 │ GhiChu = ghiChu,

76 │ UserCreated = userId,

77 │ CreatedDate = ngayTao,

78 │ KhoaSo = false,

79 │ KhongHoaDon = khongHoaDon,

80 │ SoHoaDonNcc = soHoaDonNCC,

81 │ MaChiNhanh = brandId,

82 │ MaChiNhanhInt = br.Nid ?? 0,

83 │ };

84 │

85 │ dbContext.StockNhapHang.Add(nhapHang);

86 │ dbContext.SaveChanges();

87 │

88 │ var resultRow = new ChiTietNhapHangModel(dbContext).SaveListChiTiet(nhapHang.Ma, listChiTiet, userId, ng...

89 │ if (resultRow != 0)

90 │ {

91 │ throw new Exception($"Lỗi tại dòng chi tiết số {resultRow}");

92 │ }

93 │

94 │ transaction.Commit();

95 │ // Xử lý tồn kho sau khi thêm

96 │ TonKhoBackgroundWorker.AddJob(new TonKhoJob

97 │ {

98 │ NamThang = $"{ngayNhapHang:yyyyMM}",

99 │ MaKho = maKho,

100 │ UserId = userId,

101 │ MaChiNhanh = brandId

102 │ });

103 │

104 │ return new cNhapHang

105 │ {

106 │ Result = QLBHConst.OK,

107 │ ResultText = QLBHConst.AddSuccessfully

108 │ };

109 │ }

110 │ catch (Exception ex)

111 │ {

112 │ transaction.Rollback();

113 │ return new cNhapHang

114 │ {

115 │ Result = QLBHConst.Error,

116 │ ResultText = $"Thêm thất bại. {ex.Message}"

117 │ };

118 │ }

119 │ }

120 │

121 │ public cNhapHang Edit(int ma, string soHD, DateTime ngayNhapHang, DateTime ngayHoaDon, int nhaCC, int maKho, int...

122 │ {

123 │ using var transaction = dbContext.Database.BeginTransaction();

124 │ try

125 │ {

126 │ var nhapHang = dbContext.StockNhapHang.FirstOrDefault(t => t.Ma == ma && !(t.Deleted ?? false));

127 │ if (nhapHang == null)

128 │ {

129 │ return new cNhapHang { Result = QLBHConst.Error, ResultText = QLBHConst.NotExist };

130 │ }

131 │ if (nhapHang.KhoaSo ?? false)

132 │ {

133 │ return new cNhapHang { Result = QLBHConst.Error, ResultText = "Phiếu nhập đã khóa sổ, khôn...

134 │ }

135 │ var br = dbContext.SysBranch.FirstOrDefault(x => x.Ma == nhapHang.MaChiNhanh);

136 │ var ngaySua = DateTime.Now;

137 │

138 │ nhapHang.SoHoaDon = soHD;

139 │ nhapHang.NgayNhapHang = ngayNhapHang;

140 │ nhapHang.NgayHoaDon = ngayHoaDon;

141 │ nhapHang.MaNhaCc = nhaCC;

142 │ nhapHang.MaKho = maKho;

143 │ nhapHang.TienChietKhau = tienChietKhau;

144 │ nhapHang.ThueVat = thueVAT;

145 │ nhapHang.GhiChu = ghiChu;

146 │ nhapHang.KhongHoaDon = khongHoaDon;

147 │ nhapHang.SoHoaDonNcc = soHoaDonNCC;

148 │ nhapHang.UserModified = userId;

149 │ nhapHang.ModifiedDate = ngaySua;

150 │ nhapHang.MaChiNhanh = br.Ma;

151 │ nhapHang.MaChiNhanhInt = br.Nid ?? 0;

152 │

153 │ dbContext.SaveChanges();

154 │

155 │ var resultRow = new ChiTietNhapHangModel(dbContext).SaveListChiTiet(nhapHang.Ma, listChiTiet, userId, ng...

156 │ if (resultRow != 0)

157 │ {

158 │ throw new Exception($"Lỗi tại dòng chi tiết số {resultRow}");

159 │ }

160 │

161 │ transaction.Commit();

162 │ TonKhoBackgroundWorker.AddJob(new TonKhoJob

163 │ {

164 │ NamThang = $"{ngayNhapHang:yyyyMM}",

165 │ MaKho = maKho,

166 │ UserId = userId,

167 │ MaChiNhanh = MaChiNhanh

168 │ });

169 │

170 │ return new cNhapHang

171 │ {

172 │ Result = QLBHConst.OK,

173 │ ResultText = QLBHConst.UpdateSuccessfully

174 │ };

175 │ }

176 │ catch (Exception ex)

177 │ {

178 │ transaction.Rollback();

179 │ return new cNhapHang

180 │ {

181 │ Result = QLBHConst.Error,

182 │ ResultText = $"Cập nhật thất bại. {ex.Message}"

183 │ };

184 │ }

185 │ }

186 │

187 │ public int Delete(int Ma, Guid UserID, Guid MaChiNhanh)

188 │ {

189 │ var drItem = GetItem(Ma);

190 │

191 │ if (drItem == null)

192 │ return -2;

193 │

194 │ if (drItem.KhoaSo == true)

195 │ return -3;

196 │

197 │ drItem.Deleted = true;

198 │ drItem.DeletedDate = DateTime.Now;

199 │ drItem.UserDeleted = UserID;

200 │

201 │ var updateResult = Update(drItem);

202 │ if (updateResult > 0)

203 │ {

204 │ if (drItem.NgayNhapHang.HasValue)

205 │ {

206 │ var thangNam = $"{drItem.NgayNhapHang.Value:yyyyMM}";

207 │ TonKhoBackgroundWorker.AddJob(new TonKhoJob

208 │ {

209 │ NamThang = thangNam,

210 │ UserId = UserID,

211 │ MaChiNhanh = MaChiNhanh

212 │ });

213 │ }

214 │

215 │ return 1;

216 │ }

217 │

218 │ return 0;

219 │ }

220 │

221 │ public int KhoaSo(int Ma, Guid UserID, bool Value, Guid BrandID)

222 │ {

223 │ StockNhapHang drItem = GetItem(Ma);

224 │

225 │ if (drItem != null)

226 │ {

227 │ if ((drItem.KhoaSo ?? false) == Value)

228 │ {

229 │ return -3;

230 │ }

231 │ drItem.KhoaSo = Value;

232 │ drItem.ModifiedDate = DateTime.Now;

233 │ drItem.UserModified = UserID;

234 │

235 │ if (Update(drItem) > 0)

236 │ {

237 │ List<TblLichSuChinhGia> lstLS = new List<TblLichSuChinhGia>();

238 │ if (Value)

239 │ {

240 │ List<StockChiTietNhapHang> cts = new ChiTietNhapHangModel(dbContext).GetData(Ma);

241 │ TblHangHoaModel model = new TblHangHoaModel(dbContext);

242 │ foreach (StockChiTietNhapHang item in cts)

243 │ {

244 │ TblHangHoa hh = model.GetItem(item.MaHangHoa.Value);

245 │

246 │ if (hh != null)

247 │ {

248 │ if ((hh.GiaVon ?? 0) != item.GiaVon)

249 │ {

250 │ lstLS.Add(new TblLichSuChinhGia

251 │ {

252 │ MaHangHoa = hh.Ma,

253 │ GiaCu = (hh.GiaVon ?? 0),

254 │ GiaMoi = item.GiaVon,

255 │ UserModified = UserID,

256 │ ModifiedDate = drItem.ModifiedDate,

257 │ GhiChu = "Cập nhật giá vốn từ khóa sổ nhập hàng"

258 │ });

259 │ }

260 │ hh.GiaVon = item.GiaVon;

261 │

262 │ dbContext.Set<TblHangHoa>().Update(hh);

263 │ dbContext.SaveChanges();

264 │ }

265 │ }

266 │ }

267 │ if (lstLS.Count > 0)

268 │ {

269 │ dbContext.TblLichSuChinhGia.AddRange(lstLS);

270 │ dbContext.SaveChanges();

271 │ }

272 │ return 1;

273 │ }

274 │ return 0;

275 │ }

276 │

277 │ return -2;

278 │ }

279 │

280 │ public int KhoaSoNhieu(List<int> listMa, Guid userId, Guid brandId)

281 │ {

282 │ try

283 │ {

284 │ var now = DateTime.Now;

285 │

286 │ // Lấy danh sách phiếu chưa khóa

287 │ var nhapHangs = dbContext.StockNhapHang

288 │ .Where(t => !(t.Deleted ?? false) && listMa.Contains(t.Ma) && !(t.KhoaSo ?? false))

289 │ .ToList();

290 │

291 │ foreach (var nh in nhapHangs)

292 │ {

293 │ nh.KhoaSo = true;

294 │ nh.ModifiedDate = now;

295 │ nh.UserModified = userId;

296 │ }

297 │

298 │ dbContext.SaveChanges();

299 │

300 │ // Lấy chi tiết của các phiếu

301 │ var cts = dbContext.StockChiTietNhapHang

302 │ .Where(t => !(t.Deleted ?? false) && listMa.Contains(t.Ma))

303 │ .OrderBy(t => t.Ma)

304 │ .ToList();

305 │

306 │ var hangHoaModel = new TblHangHoaModel(dbContext);

307 │ var lstLS = new List<TblLichSuChinhGia>();

308 │

309 │ foreach (var item in cts)

310 │ {

311 │ var hh = hangHoaModel.GetItem(item.MaHangHoa.Value);

312 │ if (hh == null) continue;

313 │

314 │ var giaCu = hh.GiaVon ?? 0;

315 │ if (giaCu != item.GiaVon)

316 │ {

317 │ lstLS.Add(new TblLichSuChinhGia

318 │ {

319 │ MaHangHoa = hh.Ma,

320 │ GiaCu = giaCu,

321 │ GiaMoi = item.GiaVon,

322 │ UserModified = userId,

323 │ ModifiedDate = now,

324 │ GhiChu = "Cập nhật giá vốn từ khóa sổ nhập hàng"

325 │ });

326 │ }

327 │

328 │ hh.GiaVon = item.GiaVon;

329 │ dbContext.Set<TblHangHoa>().Update(hh);

330 │ }

331 │

332 │ if (lstLS.Any())

333 │ {

334 │ dbContext.TblLichSuChinhGia.AddRange(lstLS);

335 │ }

336 │

337 │ dbContext.SaveChanges();

338 │ return 1;

339 │ }

340 │ catch (Exception ex)

341 │ {

342 │ return 0;

343 │ }

344 │ }

345 │

346 │

347 │ public List<VChiTietNhapHang> GetListChiTietNhapHang(DateTime TuNgay, DateTime DenNgay, int MaNhaCC, int MaKho)

348 │ {

349 │ return dbContext.VChiTietNhapHang.Where(t => (t.NgayHoaDon == null ? t.CreatedDate.Value.Date : t.NgayNhapHa...

350 │ }

351 │ }

352 │ }

────────────────────────────────────────────────────────────

**📄 NhapXuatKhayController.cs (CS, 752 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using Microsoft.EntityFrameworkCore;

3 │ using QuanLyBanHang.Common;

4 │ using QuanLyBanHang.Common.Attributes;

5 │ using QuanLyBanHang.Models.APIModels;

6 │ using QuanLyBanHang.Models.QLBHEntity;

7 │ using QuanLyBanHang.Models.ViewModels;

8 │ using System.Data;

9 │

10 │ namespace QuanLyBanHang.Controllers.VTTH

11 │ {

12 │ public class NhapXuatKhayController : BaseController

13 │ {

14 │ private readonly IWebHostEnvironment \_hostingEnvironment;

15 │

16 │ public NhapXuatKhayController(QuanLyBanHangContext dbcontext, MemoryCacheManager memoryCache, IWebHostEnvironmen...

17 │ {

18 │ m\_MenuID = MenuID.vtthNhapXuatKhay.GetEnumGuid();

19 │ \_hostingEnvironment = hostingEnvironment;

20 │ }

21 │ [TokenUserCheckHTTP(Permission.NotCheck)]

22 │ public IActionResult Index(string TuNgay, string DenNgay, string LoaiNXTra)

23 │ {

24 │ try

25 │ {

26 │ // Validation và set default values

27 │ DateTime dtpTuNgay = (TuNgay.ToDateTime2() ?? DateTime.Now).Date;

28 │ DateTime dtpDenNgay = (DenNgay.ToDateTime2() ?? DateTime.Now).Date;

29 │ int loaiNXT = LoaiNXTra.ToInt();

30 │

31 │ // Lấy dữ liệu cơ bản với join

32 │ var baseQuery = from t in dbcontext.VtthNhapXuatKhay.AsNoTracking()

33 │ join k in dbcontext.VtthKhay.AsNoTracking() on t.MaKhay equals k.Ma into kGroup

34 │ from k in kGroup.DefaultIfEmpty()

35 │ join hh in dbcontext.TblHangHoa.Where(x => !(x.Deleted ?? false) && (x.VatTu ?? false))....

36 │ join dvt in dbcontext.TblDonViTinh.AsNoTracking() on hh.MaDvt equals dvt.Ma into dvtGrou...

37 │ from dvt in dvtGroup.DefaultIfEmpty()

38 │ where (t.MaChiNhanh == objToken.branch || t.MaChiNhanh == Utility.defaultUID) &&

39 │ !(t.Deleted ?? false) &&

40 │ (loaiNXT == 0 || t.LoaiPhieu == loaiNXT) &&

41 │ ((t.NgayNhap.HasValue && t.NgayNhap.Value.Date >= dtpTuNgay.Date && t.NgayNhap.Val...

42 │ (t.NgayXuat.HasValue && t.NgayXuat.Value.Date >= dtpTuNgay.Date && t.NgayXuat.Val...

43 │ (t.NgayTra.HasValue && t.NgayTra.Value.Date >= dtpTuNgay.Date && t.NgayTra.Value....

44 │ select new

45 │ {

46 │ t.Ma,

47 │ t.LoaiPhieu,

48 │ t.NgayNhap,

49 │ t.NgayXuat,

50 │ t.NgayTra,

51 │ t.MaKhay,

52 │ k.MaVatTu,

53 │ CodeVatTu = hh.MaKt ?? "",

54 │ TenVatTu = hh.Ten ?? "",

55 │ TenDVT = dvt != null ? dvt.Ten : "",

56 │ t.SoLuongNhap,

57 │ t.SoLuongXuat,

58 │ t.SoLuongTra,

59 │ k.DinhMucKhay,

60 │ KhayTreo = k != null ? (k.KhayTreo ?? false) : false,

61 │ t.XacNhanTra,

62 │ DaNhap = k != null ? (k.DaNhap ?? false) : false,

63 │ Status = k != null ? (k.Status ?? false) : false

64 │ };

65 │

66 │ var listDataTH = baseQuery.ToList();

67 │ List<NhapXuatTraVTTH> listData = new List<NhapXuatTraVTTH>();

68 │

69 │ // SỬA: Xử lý từng loại phiếu riêng biệt

70 │ // 1. Xử lý Nhập (LoaiPhieu = 3)

71 │ if (loaiNXT == 0 || loaiNXT == (int)LoaiNhapXuatVTTH.Nhap)

72 │ {

73 │ var nhapData = listDataTH

74 │ .Where(t => t.LoaiPhieu == (int)LoaiNhapXuatVTTH.Nhap &&

75 │ t.NgayNhap.HasValue &&

76 │ t.NgayNhap.Value.Date >= dtpTuNgay.Date &&

77 │ t.NgayNhap.Value.Date <= dtpDenNgay.Date &&

78 │ (t.SoLuongNhap ?? 0) > 0)

79 │ .GroupBy(x => new

80 │ {

81 │ x.NgayNhap,

82 │ x.MaKhay,

83 │ x.MaVatTu,

84 │ x.CodeVatTu,

85 │ x.TenVatTu,

86 │ x.TenDVT,

87 │ x.DinhMucKhay,

88 │ })

89 │ .Select(g => new NhapXuatTraVTTH

90 │ {

91 │ LoaiNX = "Nhập",

92 │ Ngay = g.Key.NgayNhap?.Date ?? DateTime.MinValue,

93 │ MaKhay = g.Key.MaKhay?.ToString() ?? "",

94 │ MaVatTu = g.Key.MaVatTu?.ToString() ?? "",

95 │ CodeVatTu = g.Key.CodeVatTu,

96 │ TenVatTu = g.Key.TenVatTu,

97 │ TenDVT = g.Key.TenDVT,

98 │ SoLuong = g.Sum(e => e.SoLuongNhap ?? 0),

99 │ DinhMucKhay = g.Key.DinhMucKhay ?? 0,

100 │ KhayTreo = g.Any(e => e.KhayTreo) ? "1" : "0",

101 │ XacNhanTra = g.Any(e => e.XacNhanTra ?? false) ? "1" : "0",

102 │ GhiChu = TaoGhiChu(g.FirstOrDefault()?.DaNhap ?? false,

103 │ g.FirstOrDefault()?.KhayTreo ?? false,

104 │ g.FirstOrDefault()?.Status ?? false)

105 │ })

106 │ .ToList();

107 │

108 │ listData.AddRange(nhapData);

109 │ }

110 │

111 │ // 2. Xử lý Xuất (LoaiPhieu = 2)

112 │ if (loaiNXT == 0 || loaiNXT == (int)LoaiNhapXuatVTTH.Xuat)

113 │ {

114 │ var xuatData = listDataTH

115 │ .Where(t => t.LoaiPhieu == (int)LoaiNhapXuatVTTH.Xuat &&

116 │ t.NgayXuat.HasValue &&

117 │ t.NgayXuat.Value.Date >= dtpTuNgay.Date &&

118 │ t.NgayXuat.Value.Date <= dtpDenNgay.Date &&

119 │ (t.SoLuongXuat ?? 0) > 0)

120 │ .GroupBy(x => new

121 │ {

122 │ x.NgayXuat,

123 │ x.MaKhay,

124 │ x.MaVatTu,

125 │ x.CodeVatTu,

126 │ x.TenVatTu,

127 │ x.TenDVT,

128 │ x.DinhMucKhay,

129 │ })

130 │ .Select(g => new NhapXuatTraVTTH

131 │ {

132 │ LoaiNX = "Xuất",

133 │ Ngay = g.Key.NgayXuat?.Date ?? DateTime.MinValue,

134 │ MaKhay = g.Key.MaKhay?.ToString() ?? "",

135 │ MaVatTu = g.Key.MaVatTu?.ToString() ?? "",

136 │ CodeVatTu = g.Key.CodeVatTu,

137 │ TenVatTu = g.Key.TenVatTu,

138 │ TenDVT = g.Key.TenDVT,

139 │ SoLuong = g.Sum(e => e.SoLuongXuat ?? 0),

140 │ DinhMucKhay = g.Key.DinhMucKhay ?? 0,

141 │ KhayTreo = g.Any(e => e.KhayTreo) ? "1" : "0",

142 │ XacNhanTra = g.Any(e => e.XacNhanTra ?? false) ? "1" : "0",

143 │ GhiChu = TaoGhiChu(g.FirstOrDefault()?.DaNhap ?? false,

144 │ g.FirstOrDefault()?.KhayTreo ?? false,

145 │ g.FirstOrDefault()?.Status ?? false)

146 │ })

147 │ .ToList();

148 │

149 │ listData.AddRange(xuatData);

150 │ }

151 │

152 │ // 3. Xử lý Trả (LoaiPhieu = 1)

153 │ if (loaiNXT == 0 || loaiNXT == (int)LoaiNhapXuatVTTH.Tra)

154 │ {

155 │ var traData = listDataTH

156 │ .Where(t => t.LoaiPhieu == (int)LoaiNhapXuatVTTH.Tra &&

157 │ t.NgayTra.HasValue &&

158 │ t.NgayTra.Value.Date >= dtpTuNgay.Date &&

159 │ t.NgayTra.Value.Date <= dtpDenNgay.Date &&

160 │ (t.SoLuongTra ?? 0) > 0)

161 │ .GroupBy(x => new

162 │ {

163 │ x.NgayTra,

164 │ x.MaKhay,

165 │ x.MaVatTu,

166 │ x.CodeVatTu,

167 │ x.TenVatTu,

168 │ x.TenDVT,

169 │ x.DinhMucKhay,

170 │ })

171 │ .Select(g => new NhapXuatTraVTTH

172 │ {

173 │ LoaiNX = "Trả",

174 │ Ngay = g.Key.NgayTra?.Date ?? DateTime.MinValue,

175 │ MaKhay = g.Key.MaKhay?.ToString() ?? "",

176 │ MaVatTu = g.Key.MaVatTu?.ToString() ?? "",

177 │ CodeVatTu = g.Key.CodeVatTu,

178 │ TenVatTu = g.Key.TenVatTu,

179 │ TenDVT = g.Key.TenDVT,

180 │ SoLuong = g.Sum(e => e.SoLuongTra ?? 0),

181 │ DinhMucKhay = g.Key.DinhMucKhay ?? 0,

182 │ KhayTreo = g.Any(e => e.KhayTreo) ? "1" : "0",

183 │ XacNhanTra = g.Any(e => e.XacNhanTra ?? false) ? "1" : "0",

184 │ GhiChu = TaoGhiChu(g.FirstOrDefault()?.DaNhap ?? false,

185 │ g.FirstOrDefault()?.KhayTreo ?? false,

186 │ g.FirstOrDefault()?.Status ?? false)

187 │ })

188 │ .ToList();

189 │

190 │ listData.AddRange(traData);

191 │ }

192 │

193 │ // Sắp xếp kết quả

194 │ listData = listData.OrderByDescending(x => x.Ngay)

195 │ .ThenBy(x => x.LoaiNX)

196 │ .ThenBy(x => x.TenVatTu)

197 │ .ToList();

198 │

199 │ ViewData["ListData"] = listData;

200 │ ViewData["LoaiNXTra"] = LoaiNXTra ?? "";

201 │ ViewData["TuNgay"] = dtpTuNgay;

202 │ ViewData["DenNgay"] = dtpDenNgay;

203 │

204 │ return View();

205 │ }

206 │ catch (Exception ex)

207 │ {

208 │ // Log error và return empty view

209 │ ViewData["ListData"] = new List<NhapXuatTraVTTH>();

210 │ ViewData["Error"] = ex.Message;

211 │ return View();

212 │ }

213 │ }

214 │

215 │ /// <summary>

216 │ /// Helper method để tạo ghi chú trạng thái khay

217 │ /// </summary>

218 │ private string TaoGhiChu(bool daNhap, bool khayTreo, bool status)

219 │ {

220 │ var ghiChu = new List<string>();

221 │

222 │ ghiChu.Add(daNhap ? "Khay đầy" : "Khay rỗng");

223 │

224 │ if (khayTreo)

225 │ ghiChu.Add("đang treo");

226 │

227 │ if (!status)

228 │ ghiChu.Add("đã khóa");

229 │

230 │ return string.Join(" - ", ghiChu);

231 │ }

232 │

233 │ [HttpPost]

234 │ [TokenUserCheckHTTP(Permission.NhapVTTH)]

235 │ public async Task<IActionResult> setNhapKhayVatTuTheoMa(string MaKhay)

236 │ {

237 │ using var transaction = await dbcontext.Database.BeginTransactionAsync();

238 │ try

239 │ {

240 │ var khay = await dbcontext.VtthKhay.FirstOrDefaultAsync(x => x.Ma == MaKhay.ToInt() && !(x.Deleted ?? fa...

241 │

242 │ if (khay == null)

243 │ return Json(new ResponseModel(QLBHConst.Error, "Khay không tồn tại"));

244 │

245 │ // Validation

246 │ if (!(khay.Status ?? false))

247 │ return Json(new ResponseModel(QLBHConst.Error, "Khay ở trạng thái không hoạt động"));

248 │ if (khay.KhayTreo ?? false)

249 │ return Json(new ResponseModel(QLBHConst.Error, "Khay đang treo"));

250 │ if (khay.DaNhap ?? false)

251 │ return Json(new ResponseModel(QLBHConst.Error, "Khay đang đầy"));

252 │ if (!khay.DinhMucKhay.HasValue || khay.DinhMucKhay <= 0)

253 │ return Json(new ResponseModel(QLBHConst.Error, "Khay chưa cấu hình định mức hợp lệ"));

254 │

255 │ var vattu = await dbcontext.TblHangHoa.FirstOrDefaultAsync(x => x.Ma == khay.MaVatTu && !(x.Deleted ?? f...

256 │

257 │ if (vattu == null)

258 │ return Json(new ResponseModel(QLBHConst.Error, "Chưa chọn vật tư cho khay"));

259 │

260 │ // Sử dụng TonKhoVatTuModel để kiểm tra tồn kho

261 │ var tonKhoModel = new TonKhoVatTuModel(dbcontext);

262 │ var coTheTonKho = await tonKhoModel.KiemTraTonKhoTruocXuat(

263 │ khay.MaVatTu.Value, khay.MaKhoNguyenLieu.Value, khay.DinhMucKhay ?? 0);

264 │

265 │ if (!coTheTonKho)

266 │ {

267 │ var tonHienTai = tonKhoModel.GetSoLuongTonTheoVatTu(khay.MaVatTu ?? 0, khay.MaKhoNguyenLieu ?? 0);

268 │ return Json(new ResponseModel(QLBHConst.Error,

269 │ $"Không đủ tồn kho. Hiện tại: {tonHienTai}, cần: {khay.DinhMucKhay}"));

270 │ }

271 │

272 │ await tonKhoModel.NhapKhay(

273 │ khay.Ma,

274 │ khay.DinhMucKhay ?? 0,

275 │ objToken.userid,

276 │ $"Nhập khay {khay.Ma}",

277 │ objToken.branch

278 │ );

279 │

280 │ // Cập nhật trạng thái khay

281 │ khay.DaNhap = true;

282 │ khay.MaNhapKhay = 0;

283 │ khay.ModifiedDate = DateTime.Now;

284 │ khay.UserModified = objToken.userid;

285 │

286 │ dbcontext.VtthKhay.Update(khay);

287 │ await dbcontext.SaveChangesAsync();

288 │

289 │ await transaction.CommitAsync();

290 │

291 │ return Json(new ResponseModel(QLBHConst.OK, "Đã nhập vật tư vào khay"));

292 │ }

293 │ catch (Exception ex)

294 │ {

295 │ await transaction.RollbackAsync();

296 │ return Json(new ResponseModel(QLBHConst.Error, ex.Message));

297 │ }

298 │ }

299 │

300 │

301 │ [HttpPost]

302 │ [TokenUserCheckHTTP(Permission.XuatVTTH)]

303 │ public async Task<IActionResult> setXuatTraKhayVatTuTheoMa(int maKhay, string loaiPhieu, decimal soLuong)

304 │ {

305 │ using var transaction = await dbcontext.Database.BeginTransactionAsync();

306 │ try

307 │ {

308 │ var khay = await dbcontext.VtthKhay.FirstOrDefaultAsync(x => x.Ma == maKhay && !(x.Deleted ?? false));

309 │ if (khay == null)

310 │ return Json(new ResponseModel(QLBHConst.Error, "Không tìm thấy khay"));

311 │

312 │ if (!(khay.Status ?? false))

313 │ return Json(new ResponseModel(QLBHConst.Error, "Khay không hoạt động"));

314 │

315 │ if (khay.MaVatTu == null)

316 │ return Json(new ResponseModel(QLBHConst.Error, "Chưa gán vật tư cho khay"));

317 │

318 │ var maVatTu = khay.MaVatTu.Value;

319 │ var branch = objToken.branch;

320 │ var userId = objToken.userid;

321 │

322 │ switch (loaiPhieu.ToLower())

323 │ {

324 │ case "nhap":

325 │ if (khay.DaNhap ?? false)

326 │ return Json(new ResponseModel(QLBHConst.Error, "Khay đang đầy"));

327 │ if (khay.KhayTreo ?? false)

328 │ return Json(new ResponseModel(QLBHConst.Error, "Khay đang treo"));

329 │ if (soLuong != khay.DinhMucKhay)

330 │ return Json(new ResponseModel(QLBHConst.Error, $"Số lượng nhập phải bằng định ...

331 │

332 │ await new TonKhoVatTuModel(dbcontext).NhapKhay(maKhay, soLuong, userId, "Nhập khay", branch);

333 │ khay.DaNhap = true;

334 │ khay.MaNhapKhay = 0;

335 │ break;

336 │

337 │ case "xuat":

338 │ if (!(khay.DaNhap ?? false))

339 │ return Json(new ResponseModel(QLBHConst.Error, "Khay rỗng, không thể xuất"));

340 │

341 │ var tonXuat = await TinhTonKhoTrenKhay(maKhay);

342 │ if (soLuong > tonXuat)

343 │ return Json(new ResponseModel(QLBHConst.Error, $"Số lượng xuất vượt quá số còn...

344 │

345 │ await new TonKhoVatTuModel(dbcontext).XuatKhay(maKhay, soLuong, userId, "Xuất khay", branch);

346 │ break;

347 │

348 │ case "tra":

349 │ if (!(khay.DaNhap ?? false))

350 │ return Json(new ResponseModel(QLBHConst.Error, "Khay rỗng, không thể trả"));

351 │ if (khay.KhayTreo ?? false)

352 │ return Json(new ResponseModel(QLBHConst.Error, "Khay đang treo, chờ xác nhận"));

353 │

354 │ var tonTra = await TinhTonKhoTrenKhay(maKhay);

355 │ if (soLuong != tonTra)

356 │ return Json(new ResponseModel(QLBHConst.Error, $"Phải trả toàn bộ số lượng còn ...

357 │

358 │ await new TonKhoVatTuModel(dbcontext).TraKhay(maKhay, soLuong, userId, "Trả khay", branch);

359 │ khay.KhayTreo = true;

360 │ break;

361 │

362 │ default:

363 │ return Json(new ResponseModel(QLBHConst.Error, "Loại phiếu không hợp lệ (nhap/xuat/tra)...

364 │ }

365 │

366 │ khay.ModifiedDate = DateTime.Now;

367 │ khay.UserModified = userId;

368 │ dbcontext.VtthKhay.Update(khay);

369 │ await dbcontext.SaveChangesAsync();

370 │ await transaction.CommitAsync();

371 │

372 │ var thongBao = loaiPhieu.ToLower() switch

373 │ {

374 │ "nhap" => "Đã nhập vật tư vào khay",

375 │ "xuat" => "Đã xuất vật tư từ khay",

376 │ "tra" => "Đã trả khay, chờ xác nhận",

377 │ \_ => "Thành công"

378 │ };

379 │

380 │ return Json(new ResponseModel(QLBHConst.OK, thongBao));

381 │ }

382 │ catch (Exception ex)

383 │ {

384 │ await transaction.RollbackAsync();

385 │ return Json(new ResponseModel(QLBHConst.Error, ex.Message));

386 │ }

387 │ }

388 │

389 │ [HttpPost]

390 │ [TokenUserCheckHTTP(Permission.XacNhanTraVTTH)]

391 │ public async Task<IActionResult> setXacNhanTra(string Ma)

392 │ {

393 │ using var transaction = await dbcontext.Database.BeginTransactionAsync();

394 │ try

395 │ {

396 │

397 │ // Lấy danh sách phiếu trả cần xác nhận

398 │ var phieuTra = await dbcontext.VtthNhapXuatKhay

399 │ .Where(x => x.MaKhay == Cast.ToInt(Ma) && !(x.XacNhanTra ?? false) && (x.SoLuongTra ?? 0) > 0 && !(x...

400 │ .OrderByDescending(x => x.CreatedDate)

401 │ .FirstOrDefaultAsync();

402 │

403 │ if (phieuTra == null)

404 │ return Json(new ResponseModel(QLBHConst.Error, "Không có phiếu trả cần xác nhận hoặc đ...

405 │

406 │ var tonKhoModel = new TonKhoVatTuModel(dbcontext);

407 │

408 │ await tonKhoModel.XacNhanTraKhay(phieuTra.Ma, objToken.userid, objToken.branch);

409 │

410 │ // Cập nhật trạng thái xác nhận

411 │ phieuTra.XacNhanTra = true;

412 │ phieuTra.ModifiedDate = DateTime.Now;

413 │ phieuTra.UserModified = objToken.userid;

414 │ dbcontext.VtthNhapXuatKhay.Update(phieuTra);

415 │

416 │ // Cập nhật trạng thái khay (gỡ treo)

417 │ var maKhay = phieuTra.MaKhay;

418 │ var khay = await dbcontext.VtthKhay.FirstOrDefaultAsync(x => x.Ma == maKhay && !(x.Deleted ?? false));

419 │ if (khay != null)

420 │ {

421 │ khay.KhayTreo = false;

422 │ khay.DaNhap = false;

423 │ khay.ModifiedDate = DateTime.Now;

424 │ khay.UserModified = objToken.userid;

425 │ dbcontext.VtthKhay.Update(khay);

426 │ }

427 │

428 │ await dbcontext.SaveChangesAsync();

429 │ await transaction.CommitAsync();

430 │

431 │ return Json(new ResponseModel(QLBHConst.OK, "Đã xác nhận trả khay thành công."));

432 │ }

433 │ catch (Exception ex)

434 │ {

435 │ await transaction.RollbackAsync();

436 │ return Json(new ResponseModel(QLBHConst.Error, $"Lỗi xác nhận trả khay: {ex.Message}"));

437 │ }

438 │ }

439 │

440 │ private async Task<decimal> TinhTonKhoTrenKhay(int maKhay)

441 │ {

442 │ var nhapXuatKhay = await dbcontext.VtthNhapXuatKhay

443 │ .Where(t => t.MaKhay == maKhay && !(t.Deleted ?? false))

444 │ .ToListAsync();

445 │

446 │ decimal tongNhap = nhapXuatKhay

447 │ .Sum(t => t.SoLuongNhap ?? 0);

448 │

449 │ decimal tongXuat = nhapXuatKhay

450 │ .Sum(t => t.SoLuongXuat ?? 0);

451 │

452 │ decimal tongTraDaXacNhan = nhapXuatKhay

453 │ .Where(t => t.XacNhanTra == true)

454 │ .Sum(t => t.SoLuongTra ?? 0);

455 │

456 │ return tongNhap - tongXuat - tongTraDaXacNhan;

457 │ }

458 │

459 │ [TokenUserCheckHTTP(Permission.NotCheck)]

460 │ public IActionResult getListExcelNhapXuatKhayTheoKhoaPhongVatLoai(string TuNgay, string DenNgay, string MaPhongB...

461 │ {

462 │ DateTime dtpTuNgay = (TuNgay.ToDateTime2() ?? DateTime.Now).Date;

463 │ DateTime dtpDenNgay = (DenNgay.ToDateTime2() ?? DateTime.Now).Date;

464 │ int? loaiNXTraInt = LoaiNXTra.ToInt();

465 │ List<NhapXuatTraVTTH> listData = new();

466 │

467 │ var rawData = (from t in dbcontext.VtthNhapXuatKhay.AsNoTracking()

468 │ join k in dbcontext.VtthKhay.AsNoTracking() on t.MaKhay equals k.Ma

469 │ join hh in dbcontext.TblHangHoa.AsNoTracking() on k.MaVatTu equals hh.Ma

470 │ join dvt in dbcontext.TblDonViTinh.AsNoTracking() on hh.MaDvt equals dvt.Ma

471 │ where !(t.Deleted ?? false) &&

472 │ (

473 │ (t.NgayNhap != null && t.NgayNhap >= dtpTuNgay && t.NgayNhap < dtpDenNgay.AddDays(1...

474 │ (t.NgayXuat != null && t.NgayXuat >= dtpTuNgay && t.NgayXuat < dtpDenNgay.AddDays(1...

475 │ (t.NgayTra != null && t.NgayTra >= dtpTuNgay && t.NgayTra < dtpDenNgay.AddDays(1))

476 │ )

477 │ select new NhapXuatTraVTTH

478 │ {

479 │ Ma = t.Ma.ToString(),

480 │ NgayNhap = t.NgayNhap,

481 │ NgayXuat = t.NgayXuat,

482 │ NgayTra = t.NgayTra,

483 │ MaKhay = t.MaKhay.ToString(),

484 │ MaVatTu = hh.Ma.ToString(),

485 │ CodeVatTu = hh.MaKt ?? "",

486 │ TenVatTu = hh.Ten ?? "",

487 │ TenDVT = dvt.Ten ?? "",

488 │ SoLuongNhap = t.SoLuongNhap ?? 0,

489 │ SoLuongXuat = t.SoLuongXuat ?? 0,

490 │ SoLuongTra = t.SoLuongTra ?? 0,

491 │ DinhMucKhay = k.DinhMucKhay ?? 0,

492 │ KhayTreo = (k.KhayTreo ?? false) ? "1" : "0",

493 │ XacNhanTra = (t.XacNhanTra ?? false) ? "1" : "0",

494 │ GhiChu = $"{(k.DaNhap ?? false ? "Khay đầy" : "Khay rỗng")}{(k.KhayTreo ?? false ? "...

495 │ LoaiVatTu = k.MaKhoTieuHao == (int)KhoTieuHao.VTTHCHTI ? "VTTH cửa hàng tiện ích"

496 │ : k.MaKhoTieuHao == (int)KhoTieuHao.VTTHCF ? "VTTH cafe"

497 │ : k.MaKhoTieuHao == (int)KhoTieuHao.VTTHBEP ? "VTTH bếp"

498 │ : "Không xác định"

499 │ }).ToList();

500 │

501 │ // Cấu hình từng loại lọc

502 │ var typeConfigs = new List<(string Label, Func<NhapXuatTraVTTH, DateTime?> GetDate, Func<NhapXuatTraVTTH, bo...

503 │ {

504 │ ("Nhập", x => x.NgayNhap, x => x.NgayNhap >= dtpTuNgay && x.NgayNhap < dtpDenNgay.AddDays(1) && x.SoLu...

505 │ ("Xuất", x => x.NgayXuat, x => x.NgayXuat >= dtpTuNgay && x.NgayXuat < dtpDenNgay.AddDays(1) && x.SoLu...

506 │ ("Trả", x => x.NgayTra, x => x.NgayTra >= dtpTuNgay && x.NgayTra < dtpDenNgay.AddDays(1) && x.SoLuon...

507 │ };

508 │

509 │ void AddDataFiltered(string label, Func<NhapXuatTraVTTH, DateTime?> getDate, Func<NhapXuatTraVTTH, bool> pre...

510 │ {

511 │ var filtered = rawData

512 │ .Where(predicate)

513 │ .Select(t =>

514 │ {

515 │ var date = getDate(t);

516 │ if (date == null) return null;

517 │

518 │ return new NhapXuatTraVTTH

519 │ {

520 │ LoaiNX = label,

521 │ Ngay = date.Value,

522 │ MaKhay = t.MaKhay,

523 │ MaVatTu = t.MaVatTu,

524 │ CodeVatTu = t.CodeVatTu,

525 │ TenVatTu = t.TenVatTu,

526 │ TenDVT = t.TenDVT,

527 │ SoLuong = label == "Nhập" ? t.SoLuongNhap : label == "Xuất" ? t.SoLuongXuat : t.SoLuongT...

528 │ DinhMucKhay = t.DinhMucKhay,

529 │ GhiChu = t.GhiChu,

530 │ KhayTreo = t.KhayTreo,

531 │ XacNhanTra = t.XacNhanTra,

532 │ LoaiVatTu = t.LoaiVatTu

533 │ };

534 │ })

535 │ .Where(x => x != null)

536 │ .GroupBy(x => new

537 │ {

538 │ x.LoaiNX,

539 │ x.Ngay,

540 │ x.MaKhay,

541 │ x.MaVatTu,

542 │ x.CodeVatTu,

543 │ x.TenVatTu,

544 │ x.TenDVT,

545 │ x.DinhMucKhay,

546 │ x.GhiChu,

547 │ x.KhayTreo,

548 │ x.XacNhanTra,

549 │ x.LoaiVatTu

550 │ })

551 │ .Select(g => new NhapXuatTraVTTH

552 │ {

553 │ LoaiNX = g.Key.LoaiNX,

554 │ Ngay = g.Key.Ngay,

555 │ MaKhay = g.Key.MaKhay,

556 │ MaVatTu = g.Key.MaVatTu,

557 │ CodeVatTu = g.Key.CodeVatTu,

558 │ TenVatTu = g.Key.TenVatTu,

559 │ TenDVT = g.Key.TenDVT,

560 │ SoLuong = g.Sum(x => x.SoLuong),

561 │ DinhMucKhay = g.Key.DinhMucKhay,

562 │ GhiChu = g.Key.GhiChu,

563 │ KhayTreo = g.Key.KhayTreo,

564 │ XacNhanTra = g.Key.XacNhanTra,

565 │ LoaiVatTu = g.Key.LoaiVatTu

566 │ });

567 │

568 │ listData.AddRange(filtered);

569 │ }

570 │

571 │ foreach (var (label, getDate, predicate) in typeConfigs)

572 │ {

573 │ if (loaiNXTraInt == 0 || label == ((LoaiNhapXuatVTTH)loaiNXTraInt).ToString())

574 │ {

575 │ AddDataFiltered(label, getDate, predicate);

576 │ }

577 │ }

578 │

579 │ if (string.IsNullOrEmpty(FileType))

580 │ {

581 │ return Json(new ResponseModelSuccess("OK", listData));

582 │ }

583 │

584 │ // ===== Excel Export =====

585 │ string fileName = FileType == "Excel" ? "BaoCaoNhapXuatKhayVatTu.xlsx" : "BaoCaoNhapXuatKhayVatTu.pdf";

586 │ string contentType = FileType == "Excel" ? "application/ms-excel" : "application/pdf";

587 │ int startRow = 6;

588 │

589 │ var listExcelHeader = new List<ExcelValue>

590 │ {

591 │ new ExcelValue

592 │ {

593 │ Range = "A4:I4",

594 │ Value = $"Từ ngày {dtpTuNgay:dd/MM/yyyy} đến ngày {dtpDenNgay:dd/MM/yyyy}",

595 │ Format = new ExcelFormat

596 │ {

597 │ Font = new System.Drawing.Font("Times New Roman", 9.0f),

598 │ IsBold = true,

599 │ HorizontalAlignment = eHorizontalAlignment.Center,

600 │ VerticalAlignment = eVerticalAlignment.Top,

601 │ Height = 25

602 │ }

603 │ }

604 │ };

605 │

606 │ DataTable dt = new DataTable();

607 │ dt.Columns.Add("STT", typeof(int));

608 │ dt.Columns.Add("Loai Vat Tu", typeof(string));

609 │ dt.Columns.Add("Loại NX", typeof(string));

610 │ dt.Columns.Add("Ngày", typeof(string));

611 │ dt.Columns.Add("Tên thương mại", typeof(string));

612 │ dt.Columns.Add("Tên vật tư", typeof(string));

613 │ dt.Columns.Add("ĐVT", typeof(string));

614 │ dt.Columns.Add("Ghi chú", typeof(string));

615 │ dt.Columns.Add("Định mức", typeof(decimal));

616 │ dt.Columns.Add("Số lượng", typeof(decimal));

617 │

618 │ int i = 1;

619 │ foreach (var item in listData)

620 │ {

621 │ dt.Rows.Add(i++, item.LoaiVatTu, item.LoaiNX, item.Ngay.Value.ToString("dd/MM/yyyy"), item.CodeVatTu, it...

622 │ item.TenDVT, item.GhiChu, item.DinhMucKhay, item.SoLuong);

623 │ }

624 │

625 │ string fileTemplate = Path.Combine(\_hostingEnvironment.ContentRootPath, "wwwroot", "templates", "BaoCaoNhapX...

626 │

627 │ try

628 │ {

629 │ var stream = ExportToExcel.FromTemplate(fileTemplate, FileType, listExcelHeader, dt, true, startRow, "A"...

630 │ return new FileStreamResult(stream, contentType)

631 │ {

632 │ FileDownloadName = fileName

633 │ };

634 │ }

635 │ catch (Exception ex)

636 │ {

637 │ Console.WriteLine("Export error: " + ex.ToString());

638 │ return Json(new ResponseModel(QLBHConst.Error, "Lỗi xuất file."));

639 │ }

640 │ }

641 │

642 │ //[HttpGet]

643 │ //[TokenUserCheckHTTP(Permission.NhapVTTH)]

644 │ //public IActionResult setNhapCacKhayVatTuTheoKhoaPhong(string MaKhoaPhong, string MaNhapKhoDauKy)

645 │ //{

646 │ // // lấy ra khay đã nhập nhưng chưa có mã nhập khay // đưa data tồn của khay ban đầu v�...

647 │ // var listkhay = dbcontext.VtthKhay.Where(t => t.MaPhongBan == MaKhoaPhong.GetGuid() && (t.DaNhap ?? false)

648 │ // && (t.MaNhapKhay != null || t.MaNhapKhay != 0) && !(t.Deleted ?? false)).OrderBy(x => x.MaVatTu).T...

649 │ // int sokhay = 0;

650 │ // foreach (VtthKhay khay in listkhay)

651 │ // {

652 │ // var vattu = dbcontext.StockVatTu.FirstOrDefault(t => t.Ma == khay.MaVatTu && !(t.Deleted ?? false));

653 │ // if (vattu == null)

654 │ // {

655 │ // return Json(new ResponseModel(QLBHConst.Error, "Chưa chọn vật tư cho khay"));

656 │ // }

657 │ // List<VTonKhoVatTu> listTK = new List<VTonKhoVatTu>();

658 │ // List<VtthNhapXuatKhay> cts = new List<VtthNhapXuatKhay>();

659 │ // // cập nhật nhập xuất

660 │ // // cập nhật code nhập xuất

661 │ // // kiểm tra tồn kho

662 │ // var namThang = DateTime.Now.Year + DateTime.Now.Month + "";

663 │ // // lấy tồn kho theo phiếu nhập đầu kỳ dành cho VTTH

664 │ // var listMaCtnhap = dbcontext.StockChiTietNhapKho

665 │ // .Where(t => t.MaNhapKho == MaNhapKhoDauKy.ToInt())

666 │ // .Select(x => x.Ma)

667 │ // .ToList(); // <-- Thêm .ToListAsync() ở đây

668 │

669 │ // // Sau đó sử dụng listMaCtnhap đã ở trong bộ nhớ cho điều kiện Contains

670 │ // listTK = dbcontext.VTonKhoVatTu.Where(t => t.MaVatTu == khay.MaVatTu && t.NamThang == namThang && t.M...

671 │ // && listMaCtnhap.Contains(t.MaCtnhap) // <-- Sửa ở đây

672 │ // && !(t.TreoNhapKho ?? false))

673 │ // .ToList(); // <-- Thêm .ToListAsync() ở đây và đổi thành as...

674 │

675 │ // decimal soluongnhap = khay.DinhMucKhay ?? 0; decimal soluong = 0;

676 │ // // tạo chi tiết nhập

677 │ // int MaNhapKhay = 0;

678 │ // foreach (VTonKhoVatTu tk in listTK)

679 │ // {

680 │ // if (tk.SoLuongTon >= soluongnhap)

681 │ // {

682 │ // soluong = soluongnhap;

683 │ // soluongnhap = 0;

684 │ // }

685 │ // else

686 │ // {

687 │ // soluong = tk.SoLuongTon ?? 0;

688 │ // soluongnhap = soluongnhap - soluong;

689 │ // }

690 │

691 │ // VtthNhapXuatKhay nhap = new VtthNhapXuatKhay

692 │ // {

693 │ // NgayNhap = DateTime.Now,

694 │ // MaKhay = khay.Ma,

695 │ // SoLuongNhap = soluong,

696 │ // CreatedDate = DateTime.Now,

697 │ // UserCreated = objToken.userid,

698 │ // };

699 │ // cts.Add(nhap);

700 │ // if (soluongnhap == 0)

701 │ // {

702 │ // break;

703 │ // }

704 │ // }

705 │ // if (soluongnhap > 0)

706 │ // {

707 │ // return Json(new ResponseModel(QLBHConst.Error, "Vui lòng kiểm tra tồn kho "));

708 │ // }

709 │

710 │ // using (var transaction = dbcontext.Database.BeginTransaction())

711 │ // {

712 │ // try

713 │ // {

714 │ // // nhập khay vật tư

715 │ // // cập nhật trạng thái khay và code nhập

716 │ // dbcontext.AddRange(cts);

717 │ // dbcontext.SaveChanges();

718 │

719 │ // //khay.DaNhap = true;

720 │ // khay.MaNhapKhay = MaNhapKhay;

721 │ // khay.ModifiedDate = DateTime.Now;

722 │ // khay.UserModified = objToken.userid;

723 │

724 │ // dbcontext.Update(khay);

725 │ // dbcontext.SaveChanges();

726 │ // transaction.Commit();

727 │

728 │ // sokhay++;

729 │ // // chạy tồn kho VTTH

730 │ // //new TonKhoModel(dbcontext).TinhTonKhoChan(DateTime.Now.Year, DateTime.Now.Month, khay.MaKhoN...

731 │ // //new TonKhoModel(dbcontext).TinhTonKhoVTTH(DateTime.Now.Year, DateTime.Now.Month, khay.MaKho ...

732 │

733 │ // }

734 │ // catch (Exception ex)

735 │ // {

736 │ // transaction.Rollback();

737 │ // return Json(new ResponseModel(QLBHConst.Error, ex.Message));

738 │ // }

739 │ // }

740 │ // }

741 │

742 │ // if (sokhay == listkhay.Count)

743 │ // {

744 │ // return Json(new ResponseModel(QLBHConst.OK, "Đã nhập vật tư vào " + sokhay + " khay"));

745 │ // }

746 │ // else

747 │ // {

748 │ // return Json(new ResponseModel(QLBHConst.Error, "Đã nhập vật tư vào " + sokhay + " khay"));

749 │ // }

750 │ //}

751 │ }

752 │ }

────────────────────────────────────────────────────────────

**📄 NhapXuatTraVTTH.cs (CS, 30 lines)**

1 │ ﻿namespace QuanLyBanHang.Models.ViewModels

2 │ {

3 │ public class NhapXuatTraVTTH

4 │ {

5 │ public string? Ma { get; set; }

6 │ public string? LoaiNX { get; set; }

7 │ public string? MaNhapKhay { get; set; }

8 │ public string? MaPhongBan { get; set; }

9 │ public string? TenPhongBan { get; set; }

10 │ public DateTime? NgayNhap { get; set; }

11 │ public DateTime? NgayXuat { get; set; }

12 │ public DateTime? NgayTra { get; set; }

13 │ public DateTime? Ngay { get; set; }

14 │ public string? MaKhay { get; set; }

15 │ public string? MaVatTu { get; set; }

16 │ public string? CodeVatTu { get; set; }

17 │ public string? TenVatTu { get; set; }

18 │ public string? TenDVT { get; set; }

19 │ public string? MaCTNhapKho { get; set; }

20 │ public decimal DinhMucKhay { get; set; }

21 │ public decimal SoLuong { get; set; }

22 │ public decimal SoLuongNhap { get; set; }

23 │ public decimal SoLuongXuat { get; set; }

24 │ public decimal SoLuongTra { get; set; }

25 │ public string? GhiChu { get; set; }

26 │ public string? KhayTreo { get; set; }

27 │ public string? XacNhanTra { get; set; }

28 │ public string? LoaiVatTu { get; set; }

29 │ }

30 │ }

────────────────────────────────────────────────────────────

**📄 NhomThongBaoController.cs (CS, 178 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using Microsoft.EntityFrameworkCore;

3 │ using QuanLyBanHang.Common;

4 │ using QuanLyBanHang.Common.Attributes;

5 │ using QuanLyBanHang.Models.APIModels;

6 │ using QuanLyBanHang.Models.QLBHEntity;

7 │

8 │ namespace QuanLyBanHang.Controllers.SYS

9 │ {

10 │ public class NhomThongBaoController : BaseController

11 │ {

12 │ public NhomThongBaoController(QuanLyBanHangContext dbcontext, MemoryCacheManager memoryCache) : base(dbcontext, ...

13 │ {

14 │ m\_MenuID = MenuID.sysNhomThongBao.GetEnumGuid();

15 │ }

16 │

17 │ [TokenUserCheckHTTP]

18 │ public IActionResult Index()

19 │ {

20 │ return View();

21 │ }

22 │ [TokenUserCheckHTTP(Permission.Add)]

23 │ public IActionResult Add()

24 │ {

25 │ return View("Form");

26 │ }

27 │ [TokenUserCheckHTTP(Permission.Edit)]

28 │ public IActionResult Edit(string Ma)

29 │ {

30 │ var id = Cast.ToInt(Ma.Decrypt());

31 │

32 │ var item = dbcontext.TblThongBao.AsNoTracking().FirstOrDefault(t => t.Ma == id && !(t.Deleted ?? false));

33 │ if (item == null) return NotFound("Không tìm thấy");

34 │ ViewData["item"] = item;

35 │ return View("Form");

36 │ }

37 │ [HttpPost]

38 │ [TokenUserCheckHTTP(Permission.Edit)]

39 │ public IActionResult UpdateAPI(string Ma, string Ten, string TrangThai)

40 │ {

41 │ var id = Cast.ToInt(Ma.Decrypt());

42 │

43 │ var result = CheckRequire(Ten, TrangThai);

44 │ if (result != null)

45 │ return Json(result);

46 │

47 │ bool isNew = id == 0;

48 │

49 │ if (isNew)

50 │ {

51 │ var item = new TblThongBao

52 │ {

53 │ TenNhom = Ten,

54 │ TrangThai = TrangThai == "1",

55 │ MaChiNhanh = objToken.branch,

56 │ CreatedDate = DateTime.Now,

57 │ UserCreated = objToken.userid

58 │ };

59 │

60 │ dbcontext.TblThongBao.Add(item);

61 │ dbcontext.SaveChanges();

62 │ return Json(QLBHConst.ResponseAddSuccessfully);

63 │ }

64 │ else

65 │ {

66 │ var item = dbcontext.TblThongBao.FirstOrDefault(x => x.Ma == id && !(x.Deleted ?? false));

67 │ if (item == null)

68 │ return Json(QLBHConst.ResponseNotExist);

69 │

70 │ item.TenNhom = Ten;

71 │ item.TrangThai = TrangThai == "1";

72 │ item.UserModified = objToken.userid;

73 │ item.ModifiedDate = DateTime.Now;

74 │

75 │ dbcontext.SaveChanges();

76 │ return Json(QLBHConst.ResponseUpdateSuccessfully);

77 │ }

78 │ }

79 │

80 │ [HttpPost]

81 │ [TokenUserCheckHTTP(Permission.Delete)]

82 │ public IActionResult DeleteAPI(string Ma)

83 │ {

84 │ var id = Cast.ToInt(Ma.Decrypt());

85 │

86 │ var item = dbcontext.TblThongBao.FirstOrDefault(x => x.Ma == id);

87 │ if (item == null) return Json(QLBHConst.ResponseNotExist);

88 │

89 │ item.Deleted = true;

90 │ item.UserDeleted = objToken.userid;

91 │ item.DeletedDate = DateTime.Now;

92 │

93 │ dbcontext.SaveChanges();

94 │ return Json(QLBHConst.ResponseDeleteSuccessfully);

95 │ }

96 │

97 │ private static ResponseModel CheckRequire(string Ten, string TrangThai)

98 │ {

99 │ if (Ten.IsEmpty())

100 │ return new ResponseModel(QLBHConst.Error, "Vui lòng nhập tên");

101 │

102 │ if (TrangThai.IsEmpty())

103 │ return new ResponseModel(QLBHConst.Error, "Vui lòng nhập trạng thái");

104 │

105 │ return null;

106 │ }

107 │

108 │ [TokenUserCheckHTTP(Permission.NotCheck)]

109 │ public IActionResult getListData()

110 │ {

111 │ var listData = dbcontext.TblThongBao

112 │ .AsNoTracking()

113 │ .Where(x => !(x.Deleted ?? false) && x.MaChiNhanh == objToken.branch)

114 │ .Select(n => new

115 │ {

116 │ eMa = n.Ma.Encrypt(),

117 │ Ma = n.Ma,

118 │ Ten = n.TenNhom,

119 │ TrangThai = n.TrangThai ?? false,

120 │ })

121 │ .OrderBy(t => t.Ma)

122 │ .ToList();

123 │

124 │ return Json(new ResponseModel(QLBHConst.OK, "", listData));

125 │ }

126 │ [TokenUserCheckHTTP(Permission.NotCheck)]

127 │ public IActionResult getMemberList(string groupId)

128 │ {

129 │

130 │ var listData = (from user in dbcontext.SysUser.AsNoTracking().Where(x => !(x.Deleted ?? false))

131 │ join role in dbcontext.SysUserRole.AsNoTracking()

132 │ .Where(r => r.BranchId == objToken.branch) on user.Ma equals role.UserId

133 │ join nhom in dbcontext.TblNhomThongBao.AsNoTracking()

134 │ .Where(x => !(x.Deleted ?? false) && x.MaNhom == Cast.ToInt(groupId.Decrypt()))

135 │ on user.Ma equals nhom.MaUser into joined

136 │ from j in joined.DefaultIfEmpty()

137 │ select new

138 │ {

139 │ MaUser = user.Ma,

140 │ Ten = user.FullName,

141 │ IsMember = j != null,

142 │ eMa = (j != null ? j.Ma.Encrypt() : null)

143 │ }).ToList();

144 │

145 │

146 │ return Json(new ResponseModel(QLBHConst.OK, "", listData));

147 │ }

148 │ [TokenUserCheckHTTP(Permission.NotCheck)]

149 │ public IActionResult addMember(string GroupId, string UserId)

150 │ {

151 │ var item = dbcontext.TblNhomThongBao.FirstOrDefault(x => !(x.Deleted ?? false) && x.MaUser == UserId.GetGuid...

152 │ if(item != null) return Json(new ResponseModel(QLBHConst.Error, QLBHConst.ObjectExists));

153 │ var newItem = new TblNhomThongBao

154 │ {

155 │ MaNhom = Cast.ToInt(GroupId.Decrypt()),

156 │ MaUser = UserId.GetGuidNull(),

157 │ UserCreated = objToken.userid,

158 │ CreatedDate = DateTime.Now

159 │ };

160 │ dbcontext.Add(newItem);

161 │ dbcontext.SaveChanges();

162 │

163 │ return Json(new ResponseModel(QLBHConst.OK,QLBHConst.AddSuccessfully));

164 │ }

165 │ [TokenUserCheckHTTP(Permission.NotCheck)]

166 │ public IActionResult deleteMember(string GroupId, string UserId)

167 │ {

168 │ var item = dbcontext.TblNhomThongBao.FirstOrDefault(x => !(x.Deleted ?? false) && x.MaUser == UserId.GetGuid...

169 │ if (item == null) return Json(new ResponseModel(QLBHConst.Error, QLBHConst.NotExist));

170 │ item.Deleted = true;

171 │ item.UserDeleted = objToken.userid;

172 │ dbcontext.Update(item);

173 │ dbcontext.SaveChanges();

174 │

175 │ return Json(new ResponseModel(QLBHConst.OK, QLBHConst.DeleteSuccessfully));

176 │ }

177 │ }

178 │ }

────────────────────────────────────────────────────────────

**📄 NotEqual.cs (CS, 25 lines)**

1 │ ﻿namespace QuanLyBanHang.Common

2 │ {

3 │ public class NotEqual : IRouteConstraint

4 │ {

5 │ private string \_match = String.Empty;

6 │

7 │ public NotEqual(string match)

8 │ {

9 │ \_match = match;

10 │ }

11 │

12 │ public bool Match(HttpContext? httpContext, IRouter? route, string parameterName, RouteValueDictionary values, R...

13 │ {

14 │ try

15 │ {

16 │ return String.Compare(values[parameterName]!.ToString(), \_match, true) != 0;

17 │ }

18 │ catch

19 │ {

20 │ return false;

21 │ }

22 │ }

23 │

24 │ }

25 │ }

────────────────────────────────────────────────────────────

**📄 NumberToWordsHelper.cs (CS, 78 lines)**

1 │ ﻿namespace QuanLyBanHang.Common

2 │ {

3 │ public static class NumberToWordsHelper

4 │ {

5 │ private static readonly string[] ChuSo = { "không", "một", "hai", "ba", "bốn", "năm", "sáu", "bảy", "t�...

6 │ private static readonly string[] DonVi = { "", "nghìn", "triệu", "tỷ" };

7 │

8 │ public static string DocSoThanhChu(decimal number)

9 │ {

10 │ if (number == 0) return "không đồng";

11 │

12 │ string s = number < 0 ? "âm " : "";

13 │ number = Math.Abs(number);

14 │

15 │ var parts = new List<string>();

16 │ int donViIndex = 0;

17 │

18 │ while (number > 0)

19 │ {

20 │ int phan = (int)(number % 1000);

21 │ if (phan != 0)

22 │ {

23 │ var docDuTram = phan > 100;

24 │ var chu = DocBaSo(phan, docDuTram);

25 │ if (donViIndex > 0) chu += " " + DonVi[donViIndex];

26 │

27 │ parts.Insert(0, chu);

28 │ }

29 │ number /= 1000;

30 │ donViIndex++;

31 │ }

32 │

33 │ return s + string.Join(" ", parts).Trim() + " đồng";

34 │ }

35 │

36 │ private static string DocBaSo(int number, bool docDuTramPhiaTruoc)

37 │ {

38 │ int tram = number / 100;

39 │ int chuc = (number % 100) / 10;

40 │ int donvi = number % 10;

41 │ string result = "";

42 │

43 │ if (tram != 0)

44 │ {

45 │ result += ChuSo[tram] + " trăm";

46 │ }

47 │ else if (docDuTramPhiaTruoc && (chuc != 0 || donvi != 0))

48 │ {

49 │ result += "không trăm";

50 │ }

51 │

52 │ if (chuc != 0)

53 │ {

54 │ if (chuc == 1)

55 │ result += " mười";

56 │ else

57 │ result += " " + ChuSo[chuc] + " mươi";

58 │ }

59 │ else if (donvi != 0 && tram != 0)

60 │ {

61 │ result += " lẻ";

62 │ }

63 │

64 │ if (donvi != 0)

65 │ {

66 │ if (chuc != 0 && donvi == 1)

67 │ result += " mốt";

68 │ else if (chuc != 0 && donvi == 5)

69 │ result += " lăm";

70 │ else

71 │ result += " " + ChuSo[donvi];

72 │ }

73 │

74 │ return result.Trim();

75 │ }

76 │ }

77 │

78 │ }

────────────────────────────────────────────────────────────

**📄 OldTonKho.cs (CS, 53 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class OldTonKho

7 │ {

8 │ public long Ma { get; set; }

9 │

10 │ public int MaChiNhanhInt { get; set; }

11 │

12 │ public string? NamThang { get; set; }

13 │

14 │ public int? MaKho { get; set; }

15 │

16 │ public int? MaHangHoa { get; set; }

17 │

18 │ public int? MaNhapHang { get; set; }

19 │

20 │ public decimal? GiaVon { get; set; }

21 │

22 │ public DateTime? HanSd { get; set; }

23 │

24 │ public decimal? SltonDauKy { get; set; }

25 │

26 │ public decimal? SlnhapTrongKy { get; set; }

27 │

28 │ public decimal? SlxuatKhac { get; set; }

29 │

30 │ public decimal? SlxuatTieuDung { get; set; }

31 │

32 │ public decimal? SlxuatTra { get; set; }

33 │

34 │ public decimal? SlbanTrongKy { get; set; }

35 │

36 │ public decimal? SltonCuoiKy { get; set; }

37 │

38 │ public Guid? UserCreated { get; set; }

39 │

40 │ public DateTime? CreatedDate { get; set; }

41 │

42 │ public Guid? UserModified { get; set; }

43 │

44 │ public DateTime? ModifiedDate { get; set; }

45 │

46 │ public bool? Deleted { get; set; }

47 │

48 │ public Guid? UserDeleted { get; set; }

49 │

50 │ public DateTime? DeletedDate { get; set; }

51 │

52 │ public Guid? MaChiNhanh { get; set; }

53 │ }

────────────────────────────────────────────────────────────

**📄 OldTonKhoBanHang.cs (CS, 33 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class OldTonKhoBanHang

7 │ {

8 │ public long Ma { get; set; }

9 │

10 │ public int MaChiNhanhInt { get; set; }

11 │

12 │ public string NamThang { get; set; } = null!;

13 │

14 │ public int MaKho { get; set; }

15 │

16 │ public int MaHangHoa { get; set; }

17 │

18 │ public decimal? SltonDauKy { get; set; }

19 │

20 │ public decimal? SlnhapTrongKy { get; set; }

21 │

22 │ public decimal? SlxuatKhac { get; set; }

23 │

24 │ public decimal? SlxuatTieuDung { get; set; }

25 │

26 │ public decimal? SlxuatTra { get; set; }

27 │

28 │ public decimal? SlbanTrongKy { get; set; }

29 │

30 │ public decimal? SltonCuoiKy { get; set; }

31 │

32 │ public Guid MaChiNhanh { get; set; }

33 │ }

────────────────────────────────────────────────────────────

**📄 OpsChiTietHoaDon.cs (CS, 57 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class OpsChiTietHoaDon

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int MaChiNhanhInt { get; set; }

11 │

12 │ public int? MaHoaDon { get; set; }

13 │

14 │ public int? MaHangHoa { get; set; }

15 │

16 │ public string? MaVach { get; set; }

17 │

18 │ public int? MaDvt { get; set; }

19 │

20 │ public decimal? SoLuong { get; set; }

21 │

22 │ public decimal? DonGia { get; set; }

23 │

24 │ public decimal? GiaGiam { get; set; }

25 │

26 │ public decimal? ThanhTien { get; set; }

27 │

28 │ public decimal? SoLuongTraLai { get; set; }

29 │

30 │ public decimal? ThanhTienTraLai { get; set; }

31 │

32 │ public DateTime? NgayTra { get; set; }

33 │

34 │ public Guid? UserCreated { get; set; }

35 │

36 │ public DateTime? CreatedDate { get; set; }

37 │

38 │ public Guid? UserModified { get; set; }

39 │

40 │ public DateTime? ModifiedDate { get; set; }

41 │

42 │ public bool? Deleted { get; set; }

43 │

44 │ public Guid? UserDeleted { get; set; }

45 │

46 │ public DateTime? DeletedDate { get; set; }

47 │

48 │ public Guid? NguoiTra { get; set; }

49 │

50 │ public int? ThueVat { get; set; }

51 │

52 │ public decimal? TienThue { get; set; }

53 │

54 │ public string? MaTaiKhoanDoanhThu { get; set; }

55 │

56 │ public string? MaTaiKhoanThue { get; set; }

57 │ }

────────────────────────────────────────────────────────────

**📄 OpsHoaDon.cs (CS, 87 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class OpsHoaDon

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int MaChiNhanhInt { get; set; }

11 │

12 │ public Guid? MaChiNhanh { get; set; }

13 │

14 │ public int? MaKho { get; set; }

15 │

16 │ public int? MaMayBanHang { get; set; }

17 │

18 │ public int? MaPhongBenh { get; set; }

19 │

20 │ public DateTime NgayHoaDon { get; set; }

21 │

22 │ public string SoSeri { get; set; } = null!;

23 │

24 │ public int HinhThucTt { get; set; }

25 │

26 │ public int? MaGiamGia { get; set; }

27 │

28 │ public decimal? ThanhTien { get; set; }

29 │

30 │ public decimal? GiamGia { get; set; }

31 │

32 │ public decimal PhaiThu { get; set; }

33 │

34 │ public int TienKhachTra { get; set; }

35 │

36 │ public int? TienThua { get; set; }

37 │

38 │ public string? MaVachThe { get; set; }

39 │

40 │ public int? TienThe { get; set; }

41 │

42 │ public decimal? TienTraLaiKhach { get; set; }

43 │

44 │ public decimal? TienTraLaiThe { get; set; }

45 │

46 │ public decimal? KhuyenMaiTienMat { get; set; }

47 │

48 │ public string? RequestId { get; set; }

49 │

50 │ public string? WebOrderId { get; set; }

51 │

52 │ public int? MaThanhToan { get; set; }

53 │

54 │ public DateTime? NgayThanhToan { get; set; }

55 │

56 │ public Guid? NguoiThanhToan { get; set; }

57 │

58 │ public decimal? TongTienTraLai { get; set; }

59 │

60 │ public string? GhiChu { get; set; }

61 │

62 │ public bool? KhoaSo { get; set; }

63 │

64 │ public string? SoHoaDonThue { get; set; }

65 │

66 │ public Guid? TransactionUuid { get; set; }

67 │

68 │ public string? SupplierTaxCode { get; set; }

69 │

70 │ public string? TransactionId { get; set; }

71 │

72 │ public string? ReservationCode { get; set; }

73 │

74 │ public Guid? UserCreated { get; set; }

75 │

76 │ public DateTime? CreatedDate { get; set; }

77 │

78 │ public Guid? UserModified { get; set; }

79 │

80 │ public DateTime? ModifiedDate { get; set; }

81 │

82 │ public bool? Deleted { get; set; }

83 │

84 │ public Guid? UserDeleted { get; set; }

85 │

86 │ public DateTime? DeletedDate { get; set; }

87 │ }

────────────────────────────────────────────────────────────

**📄 OpsOrder.cs (CS, 29 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class OpsOrder

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int MaChiNhanhInt { get; set; }

11 │

12 │ public Guid MaChiNhanh { get; set; }

13 │

14 │ public int? MaPhong { get; set; }

15 │

16 │ public string? CodePhong { get; set; }

17 │

18 │ public string? TenKh { get; set; }

19 │

20 │ public string? Uuid { get; set; }

21 │

22 │ public string? SoDienThoai { get; set; }

23 │

24 │ public bool? DaThucHien { get; set; }

25 │

26 │ public bool? TaiQuay { get; set; }

27 │

28 │ public Guid? MaNvthuNgan { get; set; }

29 │ }

────────────────────────────────────────────────────────────

**📄 OpsOrderDanhSachMon.cs (CS, 23 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class OpsOrderDanhSachMon

7 │ {

8 │ public string Ma { get; set; } = null!;

9 │

10 │ public int MaDonHang { get; set; }

11 │

12 │ public string MaDonHangChiTiet { get; set; } = null!;

13 │

14 │ public int? MaHangHoa { get; set; }

15 │

16 │ public decimal? GiaBan { get; set; }

17 │

18 │ public int? SoLuong { get; set; }

19 │

20 │ public decimal? ThanhTien { get; set; }

21 │

22 │ public string? GhiChu { get; set; }

23 │ }

────────────────────────────────────────────────────────────

**📄 OpsOrderDetail.cs (CS, 31 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class OpsOrderDetail

7 │ {

8 │ public string Ma { get; set; } = null!;

9 │

10 │ public int MaChiNhanhInt { get; set; }

11 │

12 │ public int? MaDonHang { get; set; }

13 │

14 │ public int? TrangThai { get; set; }

15 │

16 │ public string? GhiChu { get; set; }

17 │

18 │ public DateTime? ThoiGianDatMon { get; set; }

19 │

20 │ public DateTime? ThoiGianXacNhan { get; set; }

21 │

22 │ public Guid? NvxacNhan { get; set; }

23 │

24 │ public Guid? NvgiaoHang { get; set; }

25 │

26 │ public int? MaKho { get; set; }

27 │

28 │ public int? MaNganh { get; set; }

29 │

30 │ public bool? DaThucHien { get; set; }

31 │ }

────────────────────────────────────────────────────────────

**📄 OpsOrderItems.cs (CS, 25 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class OpsOrderItems

7 │ {

8 │ public string Ma { get; set; } = null!;

9 │

10 │ public int MaDonHang { get; set; }

11 │

12 │ public string MaDonHangChiTiet { get; set; } = null!;

13 │

14 │ public int? MaHangHoa { get; set; }

15 │

16 │ public decimal? GiaBan { get; set; }

17 │

18 │ public int? SoLuong { get; set; }

19 │

20 │ public decimal? ThanhTien { get; set; }

21 │

22 │ public string? GhiChu { get; set; }

23 │

24 │ public bool? DaChuanBi { get; set; }

25 │ }

────────────────────────────────────────────────────────────

**📄 OrderController.cs (CS, 915 lines)**

1 │ ﻿using HongHungHIS.Common.Firebase;

2 │ using Microsoft.AspNetCore.Mvc;

3 │ using Microsoft.AspNetCore.SignalR;

4 │ using Microsoft.EntityFrameworkCore;

5 │ using QuanLyBanHang.Common;

6 │ using QuanLyBanHang.Common.Attributes;

7 │ using QuanLyBanHang.Models.APIModels;

8 │ using QuanLyBanHang.Models.ExModel.CAT;

9 │ using QuanLyBanHang.Models.ExModel.OPS;

10 │ using QuanLyBanHang.Models.ExModel.StoredProcedureModel;

11 │ using QuanLyBanHang.Models.ExModel.SYS;

12 │ using QuanLyBanHang.Models.QLBHEntity;

13 │ using QuanLyBanHang.Models.VCEntity;

14 │ using QuanLyBanHang.Models.VCEntity.ExModel;

15 │ using static HongHungHIS.Common.Firebase.FirebaseUtility;

16 │ using static QuanLyBanHang.Controllers.OPS.BanHangController;

17 │ using static QuanLyBanHang.Models.ExModel.OPS.OrderDetailModel;

18 │

19 │

20 │

21 │ namespace QuanLyBanHang.Controllers.OPS

22 │ {

23 │ public class OrderController : BaseController

24 │ {

25 │

26 │

27 │ private readonly IHubContext<ChatHUB> \_chatHub;

28 │ private readonly IWebHostEnvironment \_env;

29 │ private QuanLyVoucherContext vcContext { get; set; }

30 │ public OrderController(QuanLyBanHangContext dbcontext, MemoryCacheManager memoryCache, QuanLyVoucherContext vouc...

31 │ {

32 │ m\_MenuID = MenuID.opsWebOrder.GetEnumGuid();

33 │ \_chatHub = chatHub;

34 │ vcContext = voucherContext;

35 │ \_env = env;

36 │ }

37 │

38 │

39 │ // GET: Order

40 │ public ActionResult Index(string @params)

41 │ {

42 │ if (!string.IsNullOrEmpty(@params))

43 │ {

44 │ string maDecode = @params.Split(';')[0].Decrypt();

45 │ string bNoiBo = @params.Split(';')[1].Decrypt();

46 │ int ma = Cast.ToInt(maDecode);

47 │ var phongBenhList = dbcontext.TblPhongBenh.Where(x => !(x.Deleted ?? false)).ToList();

48 │

49 │ var phongBenh = phongBenhList.FirstOrDefault(phong =>

50 │ (phong.MaCu != null && phong.MaCu == ma) ||

51 │ (phong.MaCu == null && phong.Ma == ma)

52 │ );

53 │

54 │ if (phongBenh == null || (bNoiBo != "true" && bNoiBo != "false"))

55 │ {

56 │ return View("QRNotFound");

57 │ }

58 │ var chiNhanh = dbcontext.SysBranch.FirstOrDefault(x => x.Ma == phongBenh.MaChiNhanh);

59 │ ViewData["phong"] = phongBenh;

60 │ ViewData["ChiNhanh"] = chiNhanh;

61 │ ViewData["checkNoiBo"] = bNoiBo;

62 │ ViewData["isDev"] = \_env.IsDevelopment();

63 │ ViewData["bBartender"] = chiNhanh?.CoQuayPhaChe ?? false;

64 │ return View();

65 │ }

66 │ return View("QRNotFound");

67 │

68 │ }

69 │

70 │ [TokenUserCheckHTTP]

71 │ public ActionResult ViewOrder()

72 │ {

73 │ var user = new sys\_UserModel(dbcontext).GetItem(objToken.userid);

74 │ ViewBag.User = user.FullName;

75 │ ViewBag.ImgUser = user.HinhAnh ?? "/images/avatar.png";

76 │ return View();

77 │ }

78 │

79 │

80 │ [HttpPost]

81 │ public ActionResult getMenuData(int MaKho, string MaPhong, bool NoiBo = false)

82 │ {

83 │ var maChiNhanh = new PhongBenhModel(dbcontext).GetItem(Cast.ToInt(MaPhong)).MaChiNhanh ?? Utility.defaultUID...

84 │ DateTime NgayHD = new NgayKetCaModel(dbcontext).LayNgayKetCa(MaKho);

85 │

86 │ var listData = new GiaBanTheoKhoModel(dbcontext).DanhSachTonKhoHangHoaVaGiaBanTheoKho(Cast.ToInt(MaKho), str...

87 │ return Json(new ResponseModel(QLBHConst.OK, "", listData));

88 │ }

89 │ [HttpPost]

90 │ public ActionResult getMyOrder(string UUID, int MaPhong)

91 │ {

92 │ var order = new OrderModel(dbcontext).getMyOrders(MaPhong, UUID);

93 │ if (order != null)

94 │ {

95 │ var orderDetailList = new OrderDetailModel(dbcontext, \_chatHub).GetListOrderByOrderId(order.Ma);

96 │ var listData = orderDetailList.Where(x => (x.TrangThai < (int)OrderStatus.Paid)).Select(t => new

97 │ {

98 │ Ma = t.Ma,

99 │ TrangThai = t.TrangThai + "",

100 │ GhiChu = t.GhiChu,

101 │ MaKho = t.MaKho,

102 │ MaNganh = t.MaNganh,

103 │ DanhSachMon = new OrderDetailModel(dbcontext, \_chatHub).GetDanhSachMon(t.Ma)

104 │ }).ToList();

105 │ return Json(new ResponseModel(QLBHConst.OK, "", listData));

106 │

107 │ }

108 │ return Json(new ResponseModel(QLBHConst.Error, "", new List<OpsOrderDetail>()));

109 │ }

110 │

111 │ [HttpPost]

112 │ public ActionResult getDanhMucKho(string MaPhong)

113 │ {

114 │ var maChiNhanh = new PhongBenhModel(dbcontext).GetItem(Cast.ToInt(MaPhong)).MaChiNhanh ?? Utility.defaultUID...

115 │ var listKhoHang = new DanhMucKhoModel(dbcontext).DanhMucKhoOrder(maChiNhanh);

116 │ return Json(new ResponseModel(QLBHConst.OK, "", listKhoHang));

117 │ }

118 │ [HttpPost]

119 │ public ActionResult getTopLopHang(string MaPhong)

120 │ {

121 │ var maChiNhanh = new PhongBenhModel(dbcontext).GetItem(Cast.ToInt(MaPhong)).MaChiNhanh ?? Utility.defaultUID...

122 │ var listData = new LopHangHoaModel(dbcontext).GetTopLopHang(maChiNhanh);

123 │ return Json(new ResponseModel(QLBHConst.OK, "", listData));

124 │ }

125 │ [HttpPost]

126 │ public ActionResult getThucAnThucUong(string MaPhong, bool NoiBo = false)

127 │ {

128 │ var maChiNhanh = new PhongBenhModel(dbcontext).GetItem(Cast.ToInt(MaPhong)).MaChiNhanh ?? Utility.defaultUID...

129 │ var listData = new GiaBanTheoKhoModel(dbcontext).DanhSachThucAnVaThucUong(string.Format("{0:yyyyMM}", DateTi...

130 │ return Json(new ResponseModel(QLBHConst.OK, "", listData));

131 │ }

132 │

133 │

134 │ [HttpPost]

135 │ public async Task<ActionResult> sendOrder(int MaPhong, string UUID, string TenKH, string SDT, List<OrderDetailMo...

136 │ {

137 │ using (var transaction = dbcontext.Database.BeginTransaction())

138 │ {

139 │ try

140 │ {

141 │ var maChiNhanh = new PhongBenhModel(dbcontext).GetItem(Cast.ToInt(MaPhong)).MaChiNhanh ?? Utility.de...

142 │ int MaDonHang = new OrderModel(dbcontext).addOrder(MaPhong, UUID, TenKH, SDT, maChiNhanh);

143 │ new OrderDetailModel(dbcontext, \_chatHub).setOrderDetail(orderList, MaDonHang);

144 │ string tenPhong = new PhongBenhModel(dbcontext).GetItem(MaPhong).Ten;

145 │

146 │ foreach (var order in orderList)

147 │ {

148 │ string? tenKho = new DanhMucKhoModel(dbcontext).GetItem(order.MaKho ?? 0).Ten;

149 │

150 │ // Query users như cũ

151 │ var userListToSendMessage = (

152 │ from user in dbcontext.SysUser

153 │ where user.KhoHang != null &&

154 │ (user.KhoHang.Contains($"{order.MaKho};") ||

155 │ user.KhoHang.StartsWith($"{order.MaKho};") ||

156 │ user.KhoHang.EndsWith($";{order.MaKho}") ||

157 │ user.KhoHang == order.MaKho.ToString()) &&

158 │ !(user.Deleted ?? false) &&

159 │ !string.IsNullOrEmpty(user.Fcm)

160 │ join nhom in dbcontext.TblNhomThongBao.Where(x => x.MaNhom == (int)GroupNotification.ThongBa...

161 │ join tb in dbcontext.TblThongBao.Where(x => !(x.Deleted ?? false)) on nhom.MaNhom equals tb....

162 │ select user

163 │ ).Distinct().ToList();

164 │

165 │ bool fcmSent = false;

166 │ if (userListToSendMessage.Any())

167 │ {

168 │ try

169 │ {

170 │ // Tạo OrderNotificationData với thông tin structured

171 │ var orderNotificationData = new OrderNotificationData(

172 │ orderCode: order.Ma,

173 │ maDonHang: MaDonHang,

174 │ maKho: order.MaKho ?? 0,

175 │ maNganhVoucher: order.MaNganh ?? 0,

176 │ maPhong: MaPhong,

177 │ tenPhong: tenPhong,

178 │ tenKho: tenKho ?? "",

179 │ tenKH: TenKH,

180 │ sdt: SDT,

181 │ danhSachMon: order.DanhSachMon ?? new List<CartItem>()

182 │ );

183 │

184 │ // Sử dụng method mới

185 │ FirebaseUtility.NotifyOrderAction(userListToSendMessage, orderNotificationData, Firebase...

186 │ fcmSent = true;

187 │ }

188 │ catch

189 │ {

190 │ fcmSent = false;

191 │ }

192 │ }

193 │

194 │ // Fallback Telegram nếu cần

195 │ //if (!fcmSent)

196 │ //{

197 │ // string messageText = $"Bạn có đơn hàng mới {order.Ma} ({order?.DanhSachMon?.Count}...

198 │ // await TelegramBotHelper.SendTelegramMessageAsync(messageText);

199 │ //}

200 │ };

201 │

202 │ transaction.Commit();

203 │ }

204 │ catch (Exception ex)

205 │ {

206 │ transaction.Rollback();

207 │ return Json(new ResponseModel(QLBHConst.Error, ex.Message));

208 │ }

209 │ return Json(new ResponseModel(QLBHConst.OK, "Gửi yêu cầu đặt món thành công"));

210 │ }

211 │ }

212 │

213 │ [HttpPost]

214 │ public ActionResult cancelOrderDetail(int MaPhong, string UUID, string MaDonHangChiTiet)

215 │ {

216 │ using (var transaction = dbcontext.Database.BeginTransaction())

217 │ {

218 │ try

219 │ {

220 │ new OrderDetailModel(dbcontext, \_chatHub).deleteOrderDetail(MaDonHangChiTiet);

221 │ transaction.Commit();

222 │ }

223 │ catch (Exception ex)

224 │ {

225 │ transaction.Rollback();

226 │ return Json(new ResponseModel(QLBHConst.Error, ex.Message));

227 │ }

228 │

229 │ return Json(new ResponseModel(QLBHConst.OK, "Hủy đơn thành công"));

230 │ }

231 │

232 │ }

233 │ [HttpPost]

234 │ public ActionResult deleteAllOrderDetail(int MaPhong, string UUID)

235 │ {

236 │ using (var transaction = dbcontext.Database.BeginTransaction())

237 │ {

238 │ try

239 │ {

240 │ var order = new OrderModel(dbcontext).getMyOrders(MaPhong, UUID);

241 │ var orderDetailList = new OrderDetailModel(dbcontext, \_chatHub).GetListOrderByOrderId(order.Ma).Wher...

242 │ dbcontext.RemoveRange(orderDetailList);

243 │ foreach (var orderDetail in orderDetailList)

244 │ {

245 │ var orderItemList = dbcontext.OpsOrderItems.Where(x => x.MaDonHangChiTiet == orderDetail.Ma);

246 │ dbcontext.RemoveRange(orderItemList);

247 │ \_chatHub.Clients.Group("OrderGroup").SendAsync("deleteOrder", new { MaDonHang = orderDetail.Ma, ...

248 │ };

249 │ if (new OrderDetailModel(dbcontext, \_chatHub).GetListOrderByOrderId(order.Ma).Count == 0)

250 │ {

251 │ dbcontext.Remove(order);

252 │ }

253 │ dbcontext.SaveChanges();

254 │ transaction.Commit();

255 │ }

256 │ catch (Exception ex)

257 │ {

258 │ transaction.Rollback();

259 │ return Json(new ResponseModel(QLBHConst.Error, ex.Message));

260 │ }

261 │

262 │ return Json(new ResponseModel(QLBHConst.OK, "Hủy đơn thành công"));

263 │ }

264 │

265 │ }

266 │ [HttpPost]

267 │ [TokenUserCheckHTTP]

268 │ public ActionResult getKho()

269 │ {

270 │ var user = new sys\_UserModel(dbcontext).GetItem(objToken.userid);

271 │ string[] strKhoHang = (user.KhoHang + "").Split(';');

272 │ List<int> lstIDKhoHang = new List<int>();

273 │ foreach (string item in strKhoHang)

274 │ {

275 │ if (Cast.ToInt(item.Trim()) > 0)

276 │ {

277 │ lstIDKhoHang.Add(Cast.ToInt(item.Trim()));

278 │ }

279 │ }

280 │ var listKhoHangOfUser = new DanhMucKhoModel(dbcontext).GetDataByListID(lstIDKhoHang, objToken.branch).Where(...

281 │

282 │ return Json(new ResponseModel(QLBHConst.OK, "", listKhoHangOfUser));

283 │ }

284 │ [HttpPost]

285 │ [TokenUserCheckHTTP]

286 │ public ActionResult getListHinhThucTC()

287 │ {

288 │ return Json(new ResponseModel(QLBHConst.OK, "", new HinhThucTTModel(dbcontext).GetData()));

289 │ }

290 │ [HttpPost]

291 │ [TokenUserCheckHTTP]

292 │ public ActionResult getRoomList(int MaKho)

293 │ {

294 │ return Json(new ResponseModel(QLBHConst.OK, "", new OrderDetailModel(dbcontext, \_chatHub).getDanhSachPhongCo...

295 │ }

296 │ [HttpPost]

297 │ [TokenUserCheckHTTP]

298 │ public ActionResult getAllOrderByRoom(int MaKho, int MaPhong)

299 │ {

300 │ return Json(new ResponseModel(QLBHConst.OK, "", new OrderDetailModel(dbcontext, \_chatHub).getDanhSachDonHang...

301 │ }

302 │ [HttpPost]

303 │ [TokenUserCheckHTTP]

304 │ public ActionResult getAllOrder(int MaKho)

305 │ {

306 │ return Json(new ResponseModel(QLBHConst.OK, "", new OrderDetailModel(dbcontext, \_chatHub).getDanhSachDonHang...

307 │ }

308 │ [HttpPost]

309 │ [TokenUserCheckHTTP]

310 │ public ActionResult getDanhSachMon(string MaDonHangChiTiet)

311 │ {

312 │ return Json(new ResponseModel(QLBHConst.OK, "", new OrderDetailModel(dbcontext, \_chatHub).GetDanhSachMon(MaD...

313 │ }

314 │

315 │ [HttpPost]

316 │ [TokenUserCheckHTTP(Permission.CapNhatTrangThaiDonHang)]

317 │ public ActionResult confirmOrder(string MaDonHangChiTiet)

318 │ {

319 │ var orderDetail = new OrderDetailModel(dbcontext, \_chatHub).GetItem(MaDonHangChiTiet);

320 │ var order = new OrderModel(dbcontext).GetItem(orderDetail.MaDonHang ?? 0);

321 │ if (orderDetail == null || order == null) return Json(new ResponseModel(QLBHConst.Error, "Không tìm thấy...

322 │ var phong = dbcontext.TblPhongBenh.FirstOrDefault(x => x.Ma == order.MaPhong);

323 │ var kho = dbcontext.StockDanhMucKho.FirstOrDefault(x => x.Ma == orderDetail.MaKho);

324 │ var orderItemList = dbcontext.OpsOrderItems.Where(x => x.MaDonHangChiTiet == MaDonHangChiTiet).ToList();

325 │ orderDetail.TrangThai = (int)OrderStatus.Confirmed;

326 │ orderDetail.NvxacNhan = objToken.userid;

327 │ orderDetail.ThoiGianXacNhan = DateTime.Now;

328 │ dbcontext.Update(orderDetail);

329 │ dbcontext.SaveChanges();

330 │ \_chatHub.Clients.Group(order.Uuid + "").SendAsync("confirmOrder", new

331 │ {

332 │ MaDonHang = orderDetail.Ma,

333 │ MaKho = orderDetail.MaKho,

334 │

335 │ });

336 │ \_chatHub.Clients.Group("OrderGroup").SendAsync("confirmOrder", new

337 │ {

338 │ MaDonHang = orderDetail.Ma,

339 │ MaKho = orderDetail.MaKho,

340 │

341 │ });

342 │ var br = dbcontext.SysBranch.FirstOrDefault(x => x.Ma == objToken.branch);

343 │

344 │ var hanghoa = (from hh in dbcontext.TblHangHoa

345 │ join lh in dbcontext.TblLopHangHoa on hh.MaLopHang equals lh.Ma into gj

346 │ from lh in gj.DefaultIfEmpty()

347 │ select new

348 │ {

349 │ hh.Ma,

350 │ TenHang = hh.Ten,

351 │ PrinterName = lh != null ? lh.PrinterName : ""

352 │ }).ToDictionary(x => x.Ma, x => (x.TenHang, x.PrinterName));

353 │

354 │ return Json(new ResponseModel(QLBHConst.OK, "Xác nhận đơn hàng thành công",

355 │ new

356 │ {

357 │ ThoiGianXacNhan = orderDetail.ThoiGianXacNhan.Value.ToString("dd/MM/yyyy HH:mm"),

358 │ DanhSachMon = orderItemList.Select(t =>

359 │ {

360 │ hanghoa.TryGetValue(t.MaHangHoa ?? 0, out var info);

361 │ return new

362 │ {

363 │ TenHang = info.TenHang ?? "(Không xác định)",

364 │ SoLuong = t.SoLuong ?? 0,

365 │ GhiChu = t.GhiChu ?? "",

366 │ PrinterName = info.PrinterName

367 │ };

368 │ }),

369 │

370 │ PrintDanhSachMon = br.CoQuayPhaChe ?? false,

371 │ TenPhong = phong.Ten

372 │ }));

373 │ }

374 │

375 │ [HttpPost]

376 │ [TokenUserCheckHTTP(Permission.CapNhatTrangThaiDonHang)]

377 │ public ActionResult setReadyOrder(string MaDonHangChiTiet)

378 │ {

379 │ var orderDetail = new OrderDetailModel(dbcontext, \_chatHub).GetItem(MaDonHangChiTiet);

380 │ var order = new OrderModel(dbcontext).GetItem(orderDetail.MaDonHang ?? 0);

381 │ if (orderDetail == null || order == null) return Json(new ResponseModel(QLBHConst.Error, "Không tìm thấy...

382 │

383 │

384 │

385 │ orderDetail.TrangThai = (int)OrderStatus.Ready;

386 │ orderDetail.NvgiaoHang = objToken.userid;

387 │ dbcontext.Update(orderDetail);

388 │ if (order.TaiQuay ?? false)

389 │ {

390 │ orderDetail.TrangThai = (int)OrderStatus.Done;

391 │ order.DaThucHien = true;

392 │ dbcontext.Update(order);

393 │ dbcontext.Update(orderDetail);

394 │ }

395 │ dbcontext.SaveChanges();

396 │ \_chatHub.Clients.Group(order.Uuid + "").SendAsync("readyOrder", new { MaDonHang = orderDetail.Ma, MaKho = or...

397 │ \_chatHub.Clients.Group("OrderGroup").SendAsync("readyOrder", new { MaDonHang = orderDetail.Ma, MaKho = order...

398 │ return Json(new ResponseModel(QLBHConst.OK, "Đã chuẩn bị món", new { ThoiGianXacNhan = orderDetail.Th...

399 │ }

400 │ [HttpPost]

401 │ [TokenUserCheckHTTP(Permission.XacNhanThanhToan)]

402 │ public ActionResult payConfirm(string MaDonHangChiTiet)

403 │ {

404 │ var orderDetail = new OrderDetailModel(dbcontext, \_chatHub).GetItem(MaDonHangChiTiet);

405 │ var order = new OrderModel(dbcontext).GetItem(orderDetail.MaDonHang ?? 0);

406 │ if (orderDetail == null || order == null) return Json(new ResponseModel(QLBHConst.Error, "Không tìm thấy...

407 │

408 │ orderDetail.TrangThai = (int)OrderStatus.Done;

409 │ orderDetail.DaThucHien = true;

410 │ dbcontext.Update(orderDetail);

411 │

412 │ var orderDetailList = dbcontext.OpsOrderDetail.Where(x => x.MaDonHang == order.Ma).ToList();

413 │ var paidOrderList = orderDetailList.Where(x => x.TrangThai == (int)OrderStatus.Done).ToList();

414 │ if (orderDetailList.Count == paidOrderList.Count)

415 │ {

416 │ order.DaThucHien = true;

417 │ }

418 │ dbcontext.Update(order);

419 │

420 │ var hoaDon = dbcontext.OpsHoaDon.FirstOrDefault(x => x.WebOrderId == MaDonHangChiTiet);

421 │ if (hoaDon != null)

422 │ {

423 │ hoaDon.NguoiThanhToan = objToken.userid;

424 │ hoaDon.NgayThanhToan = DateTime.Now;

425 │ dbcontext.Update(hoaDon);

426 │ }

427 │

428 │ dbcontext.SaveChanges();

429 │ \_chatHub.Clients.Group(order.Uuid + "").SendAsync("payConfirmed", new { MaDonHang = orderDetail.Ma, MaKho = ...

430 │ \_chatHub.Clients.Group("OrderGroup").SendAsync("payConfirmed", new { MaDonHang = orderDetail.Ma, MaKho = ord...

431 │ return Json(new ResponseModel(QLBHConst.OK, "Đã hoàn tất đơn hàng", new { ThoiGianXacNhan = orderDet...

432 │ }

433 │ [HttpPost]

434 │ [TokenUserCheckHTTP(Permission.ThanhToanDonHang)]

435 │ public ActionResult pay(string orderId, int HinhThuc\_TT, int ThanhTien, int KhachDua, int HoanLai, string GhiChu...

436 │ {

437 │ var orderDetail = new OrderDetailModel(dbcontext, \_chatHub).GetItem(orderId);

438 │ var order = new OrderModel(dbcontext).GetItem(orderDetail.MaDonHang ?? 0);

439 │ var orderItems = dbcontext.OpsOrderItems.Where(x => x.MaDonHangChiTiet == orderId);

440 │ int MaChiNhanhInt = dbcontext.SysBranch.First(x => x.Ma == objToken.branch).Nid ?? 0;

441 │ int MaChiNhanhVoucher = dbcontext.SysBranch.First(x => x.Ma == objToken.branch).MaChiNhanhVoucher ?? 0;

442 │ var GiamGia = 0;

443 │ var TienThe = 0;

444 │ if (orderDetail == null)

445 │ {

446 │ return Json(new ResponseModel(QLBHConst.Error, "Không tìm thấy đơn hàng"));

447 │ }

448 │ if (Cast.ToInt(HoanLai) < 0 && Cast.ToInt(HinhThuc\_TT) != (int)HinhThucThanhToan.AR && Cast.ToInt(HinhThuc\_T...

449 │ {

450 │ return Json(new ResponseModel(QLBHConst.Error, "Số tiền khách đưa không đủ"));

451 │ }

452 │ if (Cast.ToInt(HinhThuc\_TT) != (int)HinhThucThanhToan.TienMat && Cast.ToInt(HinhThuc\_TT) != (int)HinhThucTha...

453 │ && Cast.ToInt(HinhThuc\_TT) != (int)HinhThucThanhToan.The && Cast.ToInt(HinhThuc\_TT) != (int)HinhThucThan...

454 │ {

455 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng chọn hình thức thanh toán"));

456 │ }

457 │ if (HinhThuc\_TT == (int)HinhThucThanhToan.ChuyenKhoan && RequestId.IsEmpty())

458 │ {

459 │ return Json(new ResponseModel(QLBHConst.Error, "Mã thanh toán không hợp lệ"));

460 │ }

461 │ if (HinhThuc\_TT == (int)HinhThucThanhToan.The && MaThe.IsEmpty())

462 │ {

463 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng nhập mã thẻ"));

464 │ }

465 │

466 │ if (HinhThuc\_TT == (int)HinhThucThanhToan.The && !MaThe.IsEmpty())

467 │ {

468 │

469 │ var voucherRs = new ThanhToanVoucherModel(vcContext).GetVoucher(MaThe ?? "", MaChiNhanhVoucher, orderDet...

470 │ var voucher = voucherRs.Voucher;

471 │ int? Result = voucherRs.Result;

472 │ if ((Result ?? 0) == 0)

473 │ {

474 │ return Json(new ResponseModel(QLBHConst.Error, "Thẻ không tồn tại"));

475 │ }

476 │ else if ((Result ?? 0) == -1)

477 │ {

478 │ return Json(new ResponseModel(QLBHConst.Error, "Thẻ không áp dụng tại chi nhánh này"));

479 │ }

480 │ else if ((Result ?? 0) == -2)

481 │ {

482 │ return Json(new ResponseModel(QLBHConst.Error, "Thẻ không áp dụng cho ngành này"));

483 │ }

484 │ else if ((Result ?? 0) == -3)

485 │ {

486 │ return Json(new ResponseModel(QLBHConst.Error, "Thẻ chưa kích hoạt"));

487 │ }

488 │ else if ((Result ?? 0) == -4)

489 │ {

490 │ return Json(new ResponseModel(QLBHConst.Error, "Thẻ chưa đến ngày sử dụng"));

491 │ }

492 │ else if ((Result ?? 0) == -5)

493 │ {

494 │ return Json(new ResponseModel(QLBHConst.Error, "Thẻ đã hết hạn sử dụng"));

495 │ }

496 │ }

497 │

498 │

499 │

500 │ List<VPermission> ListPermission = GetVPermissions();

501 │ if (ListPermission.Any(t => t.PermissionId == Permission.ThanhToanDonHang.GetEnumGuid()))

502 │ {

503 │ List<cCT\_HoaDon> dsCTHD = orderItems.Select(t => new cCT\_HoaDon()

504 │ {

505 │ Ma = t.Ma,

506 │ MaHangHoa = t.MaHangHoa.ToString(),

507 │ MaVach = "",

508 │ SoLuong = t.SoLuong.ToString(),

509 │ DonGia = t.GiaBan.ToString(),

510 │ GiaGiam = "0",

511 │ ThanhTien = (t.SoLuong \* t.GiaBan).ToString(),

512 │ SoLuongTraLai = "0",

513 │ ThanhTienTraLai = "0"

514 │

515 │ }).ToList();

516 │ cHoaDonBan res = new HoaDonBanModel(dbcontext, vcContext).Add(objToken.branch, HinhThuc\_TT, GiamGia, Tha...

517 │ if (res.Result == QLBHConst.OK)

518 │ {

519 │ orderDetail.TrangThai = (int)OrderStatus.Paid;

520 │ orderDetail.NvgiaoHang = objToken.userid;

521 │ dbcontext.Update(orderDetail);

522 │ dbcontext.SaveChanges();

523 │

524 │ \_chatHub.Clients.Group(order.Uuid + "").SendAsync("payOrder", new { MaDonHang = orderDetail.Ma, MaKh...

525 │ \_chatHub.Clients.Group("OrderGroup").SendAsync("payOrder", new { MaDonHang = orderDetail.Ma, MaKho =...

526 │

527 │

528 │

529 │ var (hd, ct, usr) = GetThongTinIn(Cast.ToInt(res.ResultText));

530 │ return Json(new ResponseModel(QLBHConst.OK, "Thao tác thành công", new

531 │ {

532 │ hoadon = hd,

533 │ cthoadon = ct,

534 │ user = usr

535 │ }));

536 │ }

537 │ else return Json(new ResponseModel(res.Result, res.ResultText));

538 │ }

539 │ else return Json(QLBHConst.ResponseNoPermission);

540 │ }

541 │

542 │ public (OpsHoaDon hd, List<sp\_ChiTietHoaDonResult> ct, Models.QLBHEntity.SysUser usr) GetThongTinIn(int MaHoaDon...

543 │ {

544 │ OpsHoaDon item = new HoaDonBanModel(dbcontext, vcContext).GetItem(MaHoaDon) ?? new OpsHoaDon();

545 │ List<sp\_ChiTietHoaDonResult> ListChitiet = new ChiTietHoaDonModel(dbcontext).GetFullData(item.Ma);

546 │ Models.QLBHEntity.SysUser usr = new NguoiDungModel(dbcontext).GetItem(objToken.userid);

547 │ return (item, ListChitiet, usr);

548 │ }

549 │

550 │ [HttpPost]

551 │ public ActionResult testSendOrderFCM(int? MaKho = null, string SDT = "")

552 │ {

553 │ try

554 │ {

555 │ List<QuanLyBanHang.Models.QLBHEntity.SysUser> targetUsers = new List<QuanLyBanHang.Models.QLBHEntity.Sys...

556 │

557 │ // Kiểm tra tham số hợp lệ

558 │ bool hasSDT = !string.IsNullOrEmpty(SDT) && SDT.Trim() != "";

559 │ bool hasMaKho = MaKho.HasValue && MaKho.Value > 0;

560 │

561 │ if (!hasSDT && !hasMaKho)

562 │ {

563 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng nhập MaKho hoặc SDT (không được �...

564 │ }

565 │

566 │ // Case 1: Có cả SDT và MaKho - lấy user theo SDT trong kho đó

567 │ if (hasSDT && hasMaKho)

568 │ {

569 │ var userBySDTAndKho = dbcontext.SysUser

570 │ .Where(u => u.SoDienThoai == SDT.Trim() &&

571 │ u.KhoHang != null &&

572 │ (u.KhoHang.Contains($"{MaKho};") ||

573 │ u.KhoHang.StartsWith($"{MaKho};") ||

574 │ u.KhoHang.EndsWith($";{MaKho}") ||

575 │ u.KhoHang == MaKho.ToString()) &&

576 │ !(u.Deleted ?? false) &&

577 │ !string.IsNullOrEmpty(u.Fcm))

578 │ .FirstOrDefault();

579 │

580 │ if (userBySDTAndKho != null)

581 │ {

582 │ targetUsers.Add(userBySDTAndKho);

583 │ }

584 │ else

585 │ {

586 │ return Json(new ResponseModel(QLBHConst.Error, $"Không tìm thấy user với SĐT: {SDT} trong...

587 │ }

588 │ }

589 │ // Case 2: Chỉ có SDT - lấy user theo SDT

590 │ else if (hasSDT && !hasMaKho)

591 │ {

592 │ var userBySDT = dbcontext.SysUser

593 │ .Where(u => u.SoDienThoai == SDT.Trim() &&

594 │ !(u.Deleted ?? false) &&

595 │ !string.IsNullOrEmpty(u.Fcm))

596 │ .FirstOrDefault();

597 │

598 │ if (userBySDT != null)

599 │ {

600 │ targetUsers.Add(userBySDT);

601 │ }

602 │ else

603 │ {

604 │ return Json(new ResponseModel(QLBHConst.Error, $"Không tìm thấy user với SĐT: {SDT} hoặ...

605 │ }

606 │ }

607 │ // Case 3: Chỉ có MaKho - lấy tất cả user trong kho

608 │ else if (!hasSDT && hasMaKho)

609 │ {

610 │ targetUsers = dbcontext.SysUser

611 │ .Where(u => u.KhoHang != null &&

612 │ (u.KhoHang.Contains($"{MaKho};") ||

613 │ u.KhoHang.StartsWith($"{MaKho};") ||

614 │ u.KhoHang.EndsWith($";{MaKho}") ||

615 │ u.KhoHang == MaKho.ToString()) &&

616 │ !(u.Deleted ?? false) &&

617 │ !string.IsNullOrEmpty(u.Fcm))

618 │ .ToList();

619 │

620 │ if (!targetUsers.Any())

621 │ {

622 │ return Json(new ResponseModel(QLBHConst.Error, $"Không tìm thấy user nào quản lý kho {Ma...

623 │ }

624 │ }

625 │

626 │ // Tạo thông tin test

627 │ string tenKho = hasMaKho ?

628 │ (new DanhMucKhoModel(dbcontext).GetItem(MaKho.Value)?.Ten ?? $"Kho {MaKho}") :

629 │ "Kho Test";

630 │

631 │ string fcmTitle = "🔔 [TEST] Đơn hàng mới";

632 │ string fcmBody = $"Đơn TEST001 (3 món)\n📍 Phòng Test - {tenKho}\n👤 Khách Test - 0123456789\n�...

633 │

634 │ // Gửi FCM

635 │ FirebaseUtility.NotifyActionWithNoSave(targetUsers, fcmTitle, fcmBody, FirebaseSound.channel);

636 │

637 │ // Thông tin response

638 │ var userInfo = targetUsers.Select(u => new

639 │ {

640 │ Ten = u.FullName,

641 │ SDT = u.SoDienThoai,

642 │ KhoHang = u.KhoHang,

643 │ CoFCM = !string.IsNullOrEmpty(u.Fcm)

644 │ }).ToList();

645 │

646 │ string resultMessage = "";

647 │ if (hasSDT && hasMaKho)

648 │ {

649 │ resultMessage = $"Đã gửi test FCM cho user SĐT {SDT} trong kho {MaKho}";

650 │ }

651 │ else if (hasSDT)

652 │ {

653 │ resultMessage = $"Đã gửi test FCM cho user SĐT {SDT}";

654 │ }

655 │ else if (hasMaKho)

656 │ {

657 │ resultMessage = $"Đã gửi test FCM cho {targetUsers.Count} user(s) trong kho {MaKho}";

658 │ }

659 │

660 │ return Json(new ResponseModel(QLBHConst.OK, resultMessage, new

661 │ {

662 │ ThamSoNhan = new

663 │ {

664 │ MaKho = hasMaKho ? MaKho : null,

665 │ SDT = hasSDT ? SDT : null,

666 │ LoaiTest = hasSDT && hasMaKho ? "SDT + MaKho" : (hasSDT ? "Chỉ SDT" : "Chỉ MaKho")

667 │ },

668 │ SoLuongUser = targetUsers.Count,

669 │ DanhSachUser = userInfo,

670 │ NoiDungGui = new

671 │ {

672 │ Title = fcmTitle,

673 │ Body = fcmBody

674 │ }

675 │ }));

676 │ }

677 │ catch (Exception ex)

678 │ {

679 │ return Json(new ResponseModel(QLBHConst.Error, $"Lỗi test FCM: {ex.Message}"));

680 │ }

681 │ }

682 │

683 │ //dùng cho app///

684 │ [HttpPost]

685 │ [TokenUserCheckHTTP]

686 │ public ActionResult getAllOrdersByUser()

687 │ {

688 │ try

689 │ {

690 │ // 1. Lấy danh sách kho user quản lý

691 │ var user = new sys\_UserModel(dbcontext).GetItem(objToken.userid);

692 │ string[] strKhoHang = (user.KhoHang + "").Split(';');

693 │ List<int> lstIDKhoHang = strKhoHang

694 │ .Where(item => Cast.ToInt(item.Trim()) > 0)

695 │ .Select(item => Cast.ToInt(item.Trim()))

696 │ .ToList();

697 │ if (!lstIDKhoHang.Any())

698 │ {

699 │ return Json(new ResponseModel(QLBHConst.OK, "User không có quyền quản lý kho nào", new List<...

700 │ }

701 │

702 │ // 2. Query tất cả đơn hàng - KHÔNG format DateTime trong LINQ

703 │ var orderTotals = dbcontext.OpsOrderItems.GroupBy(i => i.MaDonHangChiTiet)

704 │ .Select(g => new

705 │ {

706 │ MaDonHangChiTiet = g.Key,

707 │ TongTien = g.Sum(x => x.ThanhTien),

708 │ TongSoMon = g.Sum(x => x.SoLuong)

709 │ });

710 │

711 │ // 3. Query chính - Lấy DateTime gốc, format sau

712 │ var rawOrders = (from od in dbcontext.OpsOrderDetail

713 │ where lstIDKhoHang.Contains(od.MaKho ?? 0) && od.TrangThai < (int)OrderStatus.Done

714 │ join o in dbcontext.OpsOrder on od.MaDonHang equals o.Ma

715 │ join p in dbcontext.TblPhongBenh on o.MaPhong equals p.Ma into p0

716 │ from p in p0.DefaultIfEmpty()

717 │ join k in dbcontext.StockDanhMucKho on od.MaKho equals k.Ma into k0

718 │ from k in k0.DefaultIfEmpty()

719 │ join u in dbcontext.SysUser on od.NvgiaoHang equals u.Ma into u0

720 │ from u in u0.DefaultIfEmpty()

721 │ join t in orderTotals on od.Ma equals t.MaDonHangChiTiet

722 │ select new

723 │ {

724 │ MaDonHang = o.Ma,

725 │ MaDonHangChiTiet = od.Ma,

726 │ MaKho = od.MaKho,

727 │ TenKho = k.Ten,

728 │ MaPhong = o.MaPhong,

729 │ TenPhong = o.MaPhong != null ? p.Ten : (o.TaiQuay ?? false) ? "Tại quầy" : "",

730 │ GhiChu = od.GhiChu,

731 │ TenKhachHang = o.TenKh,

732 │ SoDienThoai = o.SoDienThoai,

733 │ TongTien = t.TongTien ?? 0,

734 │ TongSoMon = t.TongSoMon ?? 0,

735 │ TrangThai = od.TrangThai + "",

736 │ ThoiGianDatHang = od.ThoiGianDatMon,

737 │ ThoiGianXacNhan = od.ThoiGianXacNhan,

738 │ Bartender = k.Bartender ?? false,

739 │ TaiQuay = o.TaiQuay ?? false,

740 │ NoiBo = o.MaPhong != null ? (p.NoiBo ?? false) : false,

741 │ MaNguoiThanhToan = u.Ma.ToString() ?? "",

742 │ NguoiThanhToan = u.FullName ?? "",

743 │ MaNganhVoucher = k.MaNganhVoucher

744 │ })

745 │ .OrderByDescending(x => x.ThoiGianDatHang)

746 │ .ToList();

747 │

748 │ // 4. Format DateTime sau khi lấy dữ liệu từ database

749 │ var allOrders = rawOrders.Select(x => new

750 │ {

751 │ x.MaDonHang,

752 │ x.MaDonHangChiTiet,

753 │ x.MaKho,

754 │ x.TenKho,

755 │ x.MaPhong,

756 │ x.TenPhong,

757 │ x.GhiChu,

758 │ x.TenKhachHang,

759 │ x.SoDienThoai,

760 │ x.TongTien,

761 │ x.TongSoMon,

762 │ x.TrangThai,

763 │ ThoiGianDatHang = x.ThoiGianDatHang?.ToString("dd/MM/yyyy HH:mm"),

764 │ ThoiGianXacNhan = x.ThoiGianXacNhan?.ToString("dd/MM/yyyy HH:mm"),

765 │ x.Bartender,

766 │ x.TaiQuay,

767 │ x.NoiBo,

768 │ x.MaNguoiThanhToan,

769 │ x.NguoiThanhToan,

770 │ x.MaNganhVoucher

771 │ }).ToList();

772 │

773 │ return Json(new ResponseModel(QLBHConst.OK, "Lấy dữ liệu thành công", allOrders));

774 │ }

775 │ catch (Exception ex)

776 │ {

777 │ return Json(new ResponseModel(QLBHConst.Error, $"Có lỗi xảy ra: {ex.Message}"));

778 │ }

779 │ }

780 │

781 │

782 │ //[HttpPost]

783 │ //[TokenUserCheckHTTP]

784 │ //public ActionResult getVoucher(string MaThe, int MaNganh)

785 │ //{

786 │ // DateTime dtNow = DataUtils.GetDate();

787 │ // int? Result = 0;

788 │

789 │ // sp\_GetVoucherResult voucher = new HoaDonBanModel().GetVoucher(MaThe, QLBHConst.MaChiNhanh, MaNganh, ref Re...

790 │ // if ((Result ?? 0) == 0)

791 │ // {

792 │ // return Json(new ResponseModel(QLBHConst.Error, "Thẻ không tồn tại"));

793 │ // }

794 │ // else if ((Result ?? 0) == -1)

795 │ // {

796 │ // return Json(new ResponseModel(QLBHConst.Error, "Thẻ không áp dụng tại chi nhánh này"));

797 │ // }

798 │ // else if ((Result ?? 0) == -2)

799 │ // {

800 │ // return Json(new ResponseModel(QLBHConst.Error, "Thẻ không áp dụng cho ngành này"));

801 │ // }

802 │ // else if ((Result ?? 0) == -3)

803 │ // {

804 │ // return Json(new ResponseModel(QLBHConst.Error, "Thẻ chưa kích hoạt"));

805 │ // }

806 │ // else if ((Result ?? 0) == -4)

807 │ // {

808 │ // return Json(new ResponseModel(QLBHConst.Error, "Thẻ chưa đến ngày sử dụng"));

809 │ // }

810 │ // else if ((Result ?? 0) == -5)

811 │ // {

812 │ // return Json(new ResponseModel(QLBHConst.Error, "Thẻ đã hết hạn sử dụng"));

813 │ // }

814 │ // else

815 │ // {

816 │ // return Json(new ResponseModel(QLBHConst.OK, "", new { sodu = voucher.SoDu }));

817 │ // }

818 │ //}

819 │ //[HttpPost]

820 │ //public ActionResult getTonKhoByKho(int MaKho)

821 │ //{

822 │ // string NamThang = DateTime.Now.ToString("yyyyMM");

823 │ // HSDataContext dataContext = (new Base<GiaBanTheoKho>().DC as HSDataContext);

824 │ // var listTK = dataContext.TonKhoBanHangs.Where(t => t.NamThang == NamThang && t.MaKho == MaKho).ToList();

825 │ // var listHH = dataContext.HangHoas.Where(t => t.TrangThai && !(t.DaXoa ?? false) && !(t.HetHang ?? false))....

826 │ // var listGiaBanTheoKho = dataContext.tbl\_GiaBanTheoKhos.Where(t => t.MaKho == MaKho).ToList();

827 │ // var listDVT = dataContext.DonViTinhs.Where(t => !(t.DaXoa ?? false)).ToList();

828 │ // var listLopHang = dataContext.LopHangs.Where(t => !(t.DaXoa ?? false)).ToList();

829 │

830 │ // var listData = (from gia in listGiaBanTheoKho

831 │ // join hh in listHH on gia.MaHangHoa equals hh.HangHoa\_ID

832 │ // join lh in listLopHang on hh.LopHang equals lh.Lop\_ID

833 │ // join \_tk in listTK on gia.MaHangHoa equals \_tk.MaHangHoa into \_tk

834 │ // from tk in \_tk.DefaultIfEmpty()

835 │ // join \_dv in listDVT on hh.DVT equals \_dv.Ma into \_dv

836 │ // from dv in \_dv.DefaultIfEmpty()

837 │ // where ((tk != null && !(lh.ChoPhepBanAm ?? false) && (tk.SLTonCuoiKy ?? 0) > 0) || (lh.Cho...

838 │ // select new

839 │ // {

840 │ // Ma = gia.Ma,

841 │ // MaVach = hh.MaVach,

842 │ // TenHang = hh.TenHang,

843 │ // GiaBan = gia.GiaBan,

844 │ // DVT = hh.DVT,

845 │ // LopHang = hh.LopHang,

846 │ // TenLop = lh.TenLop,

847 │ // MaKho = gia.MaKho,

848 │ // TenKho = "",

849 │ // TenDVT = dv.Ten,

850 │ // SLTon = tk == null ? 0 : tk.SLTonCuoiKy ?? 0,

851 │ // HinhAnh = hh.HinhAnh + "",

852 │ // BanNoiBo = hh.BanNoiBo ?? false,

853 │ // MaHangHoa = hh.HangHoa\_ID,

854 │ // ChoPhepBanAm = lh.ChoPhepBanAm,

855 │ // ThoiGianBatDau = lh.ThoiGianBatDau ?? -1,

856 │ // ThoiGianKetThuc = lh.ThoiGianKetThuc ?? -1

857 │ // }).ToList();

858 │

859 │ // return Json(new ResponseModel(QLBHConst.OK, "", listData));

860 │

861 │ //}

862 │

863 │ //[HttpPost]

864 │ //public ActionResult getCategoriesFoodAndDrink()

865 │ //{

866 │ // int[] categoryIdList = new int[] { 74, 70, 67, 37, 38, 83, 84, 36, 39, 61, 62, 63, 64, 71, 72, 65, 66 };

867 │ // string NamThang = DateTime.Now.ToString("yyyyMM");

868 │ // HSDataContext dataContext = (new Base<GiaBanTheoKho>().DC as HSDataContext);

869 │ // var listTK = dataContext.TonKhoBanHangs.Where(t => t.NamThang == NamThang && (t.MaKho == 6 || t.MaKho == 7...

870 │ // var listHH = dataContext.HangHoas.Where(t => t.TrangThai && !(t.DaXoa ?? false) && !(t.HetHang ?? false))....

871 │ // var listGiaBanTheoKho = dataContext.tbl\_GiaBanTheoKhos.Where(t => t.MaKho == 6 || t.MaKho == 7).ToList();

872 │ // var listDVT = dataContext.DonViTinhs.Where(t => !(t.DaXoa ?? false)).ToList();

873 │ // var listLopHang = dataContext.LopHangs.Where(t => !(t.DaXoa ?? false) && categoryIdList.Contains(t.Lop\_ID)...

874 │

875 │ // var listData = (from gia in listGiaBanTheoKho

876 │ // join hh in listHH on gia.MaHangHoa equals hh.HangHoa\_ID

877 │ // join lh in listLopHang on hh.LopHang equals lh.Lop\_ID

878 │ // join \_tk in listTK on gia.MaHangHoa equals \_tk.MaHangHoa into \_tk

879 │ // from tk in \_tk.DefaultIfEmpty()

880 │ // join \_dv in listDVT on hh.DVT equals \_dv.Ma into \_dv

881 │ // from dv in \_dv.DefaultIfEmpty()

882 │ // where ((tk != null && !(lh.ChoPhepBanAm ?? false) && (tk.SLTonCuoiKy ?? 0) > 0) || (lh.Cho...

883 │ // select new

884 │ // {

885 │ // Lop\_ID = lh.Lop\_ID,

886 │ // TenLop = lh.TenLop,

887 │ // MaKho = gia.MaKho,

888 │ // HinhAnh = hh.HinhAnh,

889 │ // BanNoiBo = hh.BanNoiBo,

890 │ // ThoiGianBatDau = lh.ThoiGianBatDau,

891 │ // ThoiGianKetThuc = lh.ThoiGianKetThuc

892 │ // }).ToList();

893 │

894 │ // var groupedData = listData

895 │ // .GroupBy(x => new { x.Lop\_ID, x.TenLop, x.MaKho, x.ThoiGianBatDau, x.ThoiGianKetThuc })

896 │ // .Select(g => new

897 │ // {

898 │ // Lop\_ID = g.Key.Lop\_ID,

899 │ // TenLop = g.Key.TenLop,

900 │ // MaKho = g.Key.MaKho,

901 │ // HinhAnh = g.OrderBy(x => Guid.NewGuid()).FirstOrDefault()?.HinhAnh,

902 │ // BanNoiBo = g.Any(x => x.BanNoiBo ?? false),

903 │ // ThoiGianBatDau = g.Key.ThoiGianBatDau ?? -1,

904 │ // ThoiGianKetThuc = g.Key.ThoiGianKetThuc ?? -1

905 │ // })

906 │ // .ToList();

907 │

908 │

909 │ // return Json(new ResponseModel(QLBHConst.OK, "", groupedData));

910 │

911 │ //}

912 │

913 │

914 │ }

915 │ }

────────────────────────────────────────────────────────────

**📄 OrderDetailModel.cs (CS, 361 lines)**

1 │ ﻿using Humanizer;

2 │ using System.Diagnostics;

3 │ using Microsoft.EntityFrameworkCore;

4 │ using Newtonsoft.Json;

5 │ using QuanLyBanHang.Common;

6 │ using QuanLyBanHang.Controllers;

7 │ using QuanLyBanHang.Models.QLBHEntity;

8 │ using QuanLyBanHang.Models.ExModel.CAT;

9 │ using Microsoft.AspNetCore.SignalR;

10 │ using System;

11 │ using iText.Commons.Actions.Contexts;

12 │

13 │ namespace QuanLyBanHang.Models.ExModel.OPS

14 │ {

15 │ public class OrderDetailModel : Base<OpsOrderDetail>

16 │ {

17 │ public class CartItem

18 │ {

19 │ public string? Ma { get; set; }

20 │ public int MaHangHoa { get; set; }

21 │ public string? MaVach { get; set; }

22 │ public int Code { get; set; }

23 │ public string? HinhAnh { get; set; }

24 │ public string? Ten { get; set; }

25 │ public int GiaBan { get; set; }

26 │ public int SoLuong { get; set; }

27 │ public string? GhiChu { get; set; }

28 │

29 │ }

30 │ public class OrderDetailModelConvert

31 │ {

32 │ public string? Ma { get; set; }

33 │

34 │ public int? MaHoaDon { get; set; }

35 │

36 │ public string? TrangThai { get; set; }

37 │

38 │ public string? GhiChu { get; set; }

39 │

40 │ public string? ThoiGianDatMon { get; set; }

41 │

42 │ public string? ThoiGianXacNhan { get; set; }

43 │

44 │ public int? MaKho { get; set; }

45 │

46 │ public int? MaNganh { get; set; }

47 │

48 │ public List<CartItem>? DanhSachMon { get; set; }

49 │

50 │ public bool? Deleted { get; set; }

51 │ }

52 │ public class PhongTheoKho

53 │ {

54 │ public int Ma {get;set;}

55 │ public string? Ten {get;set;}

56 │ public string? Code {get;set;}

57 │ public int SoDon { get; set; }

58 │ }

59 │ public class DonHangTheoKhoVaPhong

60 │ {

61 │ public int MaDonHang { get; set; }

62 │ public string MaDonHangChiTiet { get; set; } = null!;

63 │ public int? MaKho { get; set; }

64 │ public string? TenKho { get; set; }

65 │ public int? MaPhong { get; set; }

66 │ public string? TenPhong { get; set; }

67 │ public string? GhiChu { get; set; }

68 │ public string? TenKhachHang { get; set; }

69 │ public string? SoDienThoai { get; set; }

70 │ public decimal TongTien { get; set; }

71 │ public int? TongSoMon { get; set; }

72 │ public string? TrangThai { get; set; }

73 │ public string? ThoiGianDatHang { get; set; }

74 │ public string? ThoiGianXacNhan { get; set; }

75 │ public bool Bartender { get; set; }

76 │ public bool TaiQuay { get; set; }

77 │ public bool NoiBo { get; set; }

78 │ public string? NguoiThanhToan { get; set; }

79 │ }

80 │ public partial class OrderItemModel

81 │ {

82 │ public string Ma { get; set; }

83 │

84 │ public int MaDonHang { get; set; }

85 │

86 │ public string MaDonHangChiTiet { get; set; }

87 │

88 │ public int? MaHangHoa { get; set; }

89 │

90 │ public string? TenHangHoa { get; set; }

91 │

92 │ public string? HinhAnh { get; set; }

93 │

94 │ public decimal? GiaBan { get; set; }

95 │

96 │ public int? SoLuong { get; set; }

97 │

98 │ public decimal? ThanhTien { get; set; }

99 │

100 │ public string? GhiChu { get; set; }

101 │

102 │ public bool? DaChuanBi { get; set; }

103 │ }

104 │ private readonly IHubContext<ChatHUB> \_chatHub;

105 │ public OrderDetailModel(DbContext dbContext, IHubContext<ChatHUB> chatHub) : base(dbContext)

106 │ {

107 │ \_chatHub = chatHub;

108 │ }

109 │ QuanLyBanHangContext dbContext

110 │ {

111 │ get { return \_dbContext as QuanLyBanHangContext; }

112 │ set { return; }

113 │ }

114 │ public OpsOrderDetail GetItem(string Ma)

115 │ {

116 │ return First(t => t.Ma == Ma);

117 │ }

118 │ public List<OpsOrderDetail> GetListOrderByOrderId(int orderId)

119 │ {

120 │ return Get(t => t.MaDonHang == orderId && t.TrangThai != (int)OrderStatus.Cancel).ToList();

121 │ }

122 │ public List<OpsOrderDetail> GetAll()

123 │ {

124 │ return Get().ToList();

125 │ }

126 │ public void deleteOrderDetail(string MaDonHangChiTiet)

127 │ {

128 │ var orderDetail = GetItem(MaDonHangChiTiet);

129 │ var order = dbContext.OpsOrder.FirstOrDefault(x => x.Ma == orderDetail.MaDonHang);

130 │ if (orderDetail == null)

131 │ {

132 │ throw new Exception("Đơn hàng không tồn tại");

133 │ }

134 │ dbContext.Remove(orderDetail);

135 │ dbContext.SaveChanges();

136 │

137 │ var orderItemList = dbContext.OpsOrderItems.Where(x => x.MaDonHangChiTiet == MaDonHangChiTiet).ToList();

138 │ dbContext.RemoveRange(orderItemList);

139 │ dbContext.SaveChanges();

140 │ if(GetListOrderByOrderId(order.Ma).Count == 0)

141 │ {

142 │ dbContext.Remove(order);

143 │ dbContext.SaveChanges();

144 │ }

145 │ \_chatHub.Clients.Group("OrderGroup").SendAsync("deleteOrder", new

146 │ {

147 │ MaDonHang = orderDetail.Ma,

148 │ MaKho = orderDetail.MaKho,

149 │ MaPhong = order.MaPhong,

150 │

151 │ });

152 │ }

153 │ public void setOrderDetail(List<OrderDetailModelConvert> orderList, int MaDonHang)

154 │ {

155 │ var order = new OrderModel(dbContext).GetItem(MaDonHang);

156 │ foreach(OrderDetailModelConvert item in orderList)

157 │ {

158 │ var kho = dbContext.StockDanhMucKho.FirstOrDefault(x => !(x.Deleted ?? false) && x.Ma == item.MaKho);

159 │ if (kho == null) throw new Exception("Kho hàng không tồn tại");

160 │ if (DateTime.Now.TimeOfDay.TotalHours < (kho.ThoiGianBatDau ?? 0) || DateTime.Now.TimeOfDay.TotalHours >...

161 │ {

162 │ throw new Exception($"Thời gian mở bán: {kho.ThoiGianBatDau}h - {kho.ThoiGianKetThuc}h\nVui lò...

163 │ }

164 │ var orderDetail = new OpsOrderDetail

165 │ {

166 │ Ma = item.Ma + "",

167 │ MaDonHang = MaDonHang,

168 │ TrangThai = (int)OrderStatus.WaitConfirm,

169 │ GhiChu = item.GhiChu,

170 │ ThoiGianDatMon = DateTime.Now,

171 │ ThoiGianXacNhan = null,

172 │ MaKho = item.MaKho,

173 │ MaNganh = item.MaNganh,

174 │ MaChiNhanhInt = order.MaChiNhanhInt,

175 │ };

176 │ dbContext.Add(orderDetail);

177 │ dbContext.SaveChanges();

178 │

179 │ var danhSachMon = item.DanhSachMon;

180 │ var namThang = DateTime.Now.ToString("yyyyMM");

181 │ foreach (var mon in danhSachMon)

182 │ {

183 │ VGiaBanTheoKho? hh = dbContext.VGiaBanTheoKho.FirstOrDefault(t => t.MaHangHoa == Cast.ToInt(mon.MaHa...

184 │ StockTonKhoBanHang? tonkho = dbContext.StockTonKhoBanHang.FirstOrDefault(t =>

185 │ t.MaKho == item.MaKho && t.NamThang == namThang && t.MaHangHoa == Cast.ToInt(mon.MaHangHoa));

186 │ var slTon = tonkho?.SltonCuoiKy ?? 0;

187 │

188 │ if (slTon < mon.SoLuong && !(hh?.ChoPhepBanAm ?? false))

189 │ {

190 │ throw new Exception($"Số lượng tồn kho không đủ. Hàng hóa: {mon.Ten}. Tồn kho: {s...

191 │ }

192 │ var orderItem = new OpsOrderItems

193 │ {

194 │ Ma = mon.Ma + "",

195 │ MaDonHang = orderDetail.MaDonHang ?? 0,

196 │ MaDonHangChiTiet = orderDetail.Ma,

197 │ MaHangHoa = mon.MaHangHoa,

198 │ GiaBan = mon.GiaBan,

199 │ SoLuong = mon.SoLuong,

200 │ ThanhTien = mon.GiaBan \* mon.SoLuong,

201 │ GhiChu = mon.GhiChu,

202 │ };

203 │ dbContext.Add(orderItem);

204 │ dbContext.SaveChanges();

205 │

206 │ }

207 │ \_chatHub.Clients.Group("OrderGroup").SendAsync("newOrder", new

208 │ {

209 │ MaDonHang = orderDetail.Ma,

210 │ MaKho = orderDetail.MaKho,

211 │ TenKH = order.TenKh,

212 │ CodePhong = order.CodePhong,

213 │ MaPhong = order.MaPhong,

214 │ DanhSachMon = item.DanhSachMon,

215 │ TenKho = new DanhMucKhoModel(dbContext).GetItem(orderDetail.MaKho ?? 0).Ten

216 │ });

217 │ }

218 │

219 │

220 │ }

221 │ public List<PhongTheoKho> getDanhSachPhongCoDonHangTheoKho(int MaKho, Guid MaChiNhanh)

222 │ {

223 │ var query = from phong in dbContext.TblPhongBenh.Where(x => !(x.Deleted ?? false) && x.MaChiNhanh == MaChiNh...

224 │ join order in dbContext.OpsOrder on phong.Ma equals order.MaPhong

225 │ join detail in dbContext.OpsOrderDetail.Where(x => x.TrangThai < (int)OrderStatus.Done) on order...

226 │ where detail.MaKho == MaKho

227 │ select new

228 │ {

229 │ phong.Ma,

230 │ phong.Ten,

231 │ order.CodePhong,

232 │ OrderId = order.Ma ,

233 │ MaDonHangChiTiet = detail.Ma,

234 │ };

235 │ var test = query.ToList();

236 │ var groupedQuery = from p in query

237 │ group p by new { p.Ma, p.Ten, p.CodePhong } into roomGroup

238 │ select new PhongTheoKho

239 │ {

240 │ Ma = roomGroup.Key.Ma,

241 │ Ten = roomGroup.Key.Ten,

242 │ Code = roomGroup.Key.CodePhong,

243 │ SoDon = roomGroup.Select(x => x.MaDonHangChiTiet).Distinct().Count()

244 │ };

245 │

246 │ return groupedQuery.ToList();

247 │

248 │ }

249 │ public List<DonHangTheoKhoVaPhong> getDanhSachDonHangTheoKhoVaPhong(int MaKho, int MaPhong)

250 │ {

251 │ var orderTotals = dbContext.OpsOrderItems.GroupBy(i => i.MaDonHangChiTiet)

252 │ .Select(g => new

253 │ {

254 │ MaDonHangChiTiet = g.Key,

255 │ TongTien = g.Sum(x => x.ThanhTien),

256 │ TongSoMon = g.Sum(x => x.SoLuong)

257 │

258 │ });

259 │ var kho = dbContext.StockDanhMucKho.FirstOrDefault(x => x.Ma == MaKho);

260 │ var result = (from od in dbContext.OpsOrderDetail

261 │ where od.MaKho == MaKho && od.TrangThai < (int)OrderStatus.Done

262 │ join o in dbContext.OpsOrder on od.MaDonHang equals o.Ma

263 │ where o.MaPhong == MaPhong

264 │ join p in dbContext.TblPhongBenh on o.MaPhong equals p.Ma into p0

265 │ from p in p0.DefaultIfEmpty()

266 │ join k in dbContext.StockDanhMucKho on od.MaKho equals k.Ma into k0

267 │ from k in k0.DefaultIfEmpty()

268 │ join u in dbContext.SysUser on od.NvgiaoHang equals u.Ma into u0

269 │ from u in u0.DefaultIfEmpty()

270 │ join t in orderTotals on od.Ma equals t.MaDonHangChiTiet

271 │ select new DonHangTheoKhoVaPhong

272 │ {

273 │ MaDonHang = o.Ma,

274 │ MaDonHangChiTiet = od.Ma,

275 │ MaKho = od.MaKho,

276 │ TenKho = k.Ten,

277 │ MaPhong = o.MaPhong,

278 │ TenPhong = o.MaPhong != null ? p.Ten : (o.TaiQuay ?? false) ? "Tại quầy" : "",

279 │ GhiChu = od.GhiChu,

280 │ TenKhachHang = o.TenKh,

281 │ SoDienThoai = o.SoDienThoai,

282 │ TongTien = t.TongTien ?? 0,

283 │ TongSoMon = t.TongSoMon ?? 0,

284 │ TrangThai = od.TrangThai + "",

285 │ ThoiGianDatHang = od.ThoiGianDatMon == null ? null : od.ThoiGianDatMon.Value.ToString("dd...

286 │ ThoiGianXacNhan = od.ThoiGianXacNhan == null ? null : od.ThoiGianXacNhan.Value.ToString("d...

287 │ Bartender = kho.Bartender ?? false,

288 │ TaiQuay = o.TaiQuay ?? false,

289 │ NoiBo = o.MaPhong != null ? (p.NoiBo ?? false) : false,

290 │ NguoiThanhToan = u.FullName

291 │ })

292 │ .ToList();

293 │ return result;

294 │ }

295 │

296 │ public List<DonHangTheoKhoVaPhong> getDanhSachDonHangTheoKho(int MaKho)

297 │ {

298 │ var orderTotals = dbContext.OpsOrderItems.GroupBy(i => i.MaDonHangChiTiet)

299 │ .Select(g => new

300 │ {

301 │ MaDonHangChiTiet = g.Key,

302 │ TongTien = g.Sum(x => x.ThanhTien),

303 │ TongSoMon = g.Sum(x => x.SoLuong)

304 │ });

305 │ var kho = dbContext.StockDanhMucKho.FirstOrDefault(x => x.Ma == MaKho);

306 │ var result = (from od in dbContext.OpsOrderDetail

307 │ where od.MaKho == MaKho && od.TrangThai < (int)OrderStatus.Done

308 │ join o in dbContext.OpsOrder on od.MaDonHang equals o.Ma

309 │ join p in dbContext.TblPhongBenh on o.MaPhong equals p.Ma into p0

310 │ from p in p0.DefaultIfEmpty()

311 │ join k in dbContext.StockDanhMucKho on od.MaKho equals k.Ma into k0

312 │ from k in k0.DefaultIfEmpty()

313 │ join u in dbContext.SysUser on od.NvgiaoHang equals u.Ma into u0

314 │ from u in u0.DefaultIfEmpty()

315 │ join t in orderTotals on od.Ma equals t.MaDonHangChiTiet

316 │ select new DonHangTheoKhoVaPhong

317 │ {

318 │ MaDonHang = o.Ma,

319 │ MaDonHangChiTiet = od.Ma,

320 │ MaKho = od.MaKho,

321 │ TenKho = k.Ten,

322 │ MaPhong = o.MaPhong,

323 │ TenPhong = o.MaPhong != null ? p.Ten : (o.TaiQuay ?? false) ? "Tại quầy" : "",

324 │ GhiChu = od.GhiChu,

325 │ TenKhachHang = o.TenKh,

326 │ SoDienThoai = o.SoDienThoai,

327 │ TongTien = t.TongTien ?? 0,

328 │ TongSoMon = t.TongSoMon ?? 0,

329 │ TrangThai = od.TrangThai + "",

330 │ ThoiGianDatHang = od.ThoiGianDatMon == null ? null : od.ThoiGianDatMon.Value.ToString("dd/...

331 │ ThoiGianXacNhan = od.ThoiGianXacNhan == null ? null : od.ThoiGianXacNhan.Value.ToString("d...

332 │ Bartender = kho.Bartender ?? false,

333 │ TaiQuay = o.TaiQuay ?? false,

334 │ NoiBo = o.MaPhong != null ? (p.NoiBo ?? false) : false,

335 │ NguoiThanhToan = u.FullName

336 │ })

337 │ .ToList().OrderBy(x => x.ThoiGianDatHang).ToList();

338 │ return result;

339 │ }

340 │ public List<OrderItemModel> GetDanhSachMon(string MaDonHangChiTiet)

341 │ {

342 │ var listData = (from oi in dbContext.OpsOrderItems.Where(x => x.MaDonHangChiTiet == MaDonHangChiTiet)

343 │ join hh in dbContext.TblHangHoa.Where(x => !(x.Deleted ?? false)) on oi.MaHangHoa equals hh....

344 │ select new OrderItemModel

345 │ {

346 │ Ma = oi.Ma,

347 │ MaDonHang = oi.MaDonHang,

348 │ MaDonHangChiTiet = oi.MaDonHangChiTiet,

349 │ MaHangHoa = oi.MaHangHoa,

350 │ TenHangHoa = hh.Ten,

351 │ HinhAnh = hh.HinhAnh,

352 │ GiaBan = oi.GiaBan,

353 │ SoLuong = oi.SoLuong,

354 │ ThanhTien = oi.ThanhTien,

355 │ GhiChu = oi.GhiChu,

356 │

357 │ }).ToList();

358 │ return listData;

359 │ }

360 │ }

361 │ }

────────────────────────────────────────────────────────────

**📄 OrderModel.cs (CS, 113 lines)**

1 │ ﻿using System;

2 │ using iText.Commons.Actions.Contexts;

3 │ using Microsoft.EntityFrameworkCore;

4 │ using QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ namespace QuanLyBanHang.Models.ExModel.OPS

7 │ {

8 │ public class OrderModel : Base<OpsOrder>

9 │ {

10 │ public OrderModel(DbContext dbContext) : base(dbContext)

11 │ {

12 │ }

13 │ QuanLyBanHangContext dbContext

14 │ {

15 │ get { return \_dbContext as QuanLyBanHangContext; }

16 │ set { return; }

17 │ }

18 │ public OpsOrder GetItem(int Ma)

19 │ {

20 │ return First(t => t.Ma == Ma);

21 │ }

22 │ public List<OpsOrder> GetAll(Guid MaChiNhanh)

23 │ {

24 │ return Get(x => x.MaChiNhanh == MaChiNhanh).ToList();

25 │ }

26 │

27 │ public OpsOrder getMyOrders(int roomId, string uuid)

28 │ {

29 │ return First(t => t.MaPhong == roomId && t.Uuid == uuid && !(t.DaThucHien ?? false));

30 │ }

31 │

32 │ public int addOrder(int MaPhong, string UUID, string TenKH, string SDT, Guid MaChiNhanh)

33 │ {

34 │ var phongBenh = dbContext.TblPhongBenh.FirstOrDefault(t => t.Ma == MaPhong);

35 │ if (phongBenh == null)

36 │ {

37 │ throw new Exception($"Phòng bệnh không tồn tại: MaPhong={MaPhong}");

38 │ }

39 │ var br = dbContext.SysBranch.FirstOrDefault(x => x.Ma == MaChiNhanh);

40 │ if (br == null)

41 │ {

42 │ throw new Exception("Chi nhánh không tồn tại");

43 │ }

44 │ var order = dbContext.OpsOrder.FirstOrDefault(x => x.MaPhong == MaPhong && x.Uuid == UUID && !(x.DaThucHien ...

45 │ if (order == null )

46 │ {

47 │ var item = new OpsOrder

48 │ {

49 │ MaChiNhanh = MaChiNhanh,

50 │ MaChiNhanhInt = br.Nid ?? 0,

51 │ MaPhong = MaPhong,

52 │ CodePhong = phongBenh.Code,

53 │ TenKh = TenKH,

54 │ Uuid = UUID,

55 │ SoDienThoai = SDT,

56 │ };

57 │ dbContext.Add(item);

58 │ dbContext.SaveChanges();

59 │ return item.Ma;

60 │ }

61 │ else

62 │ {

63 │ order.TenKh = TenKH;

64 │ order.SoDienThoai = SDT;

65 │ dbContext.Update(order);

66 │ dbContext.SaveChanges();

67 │ return order.Ma;

68 │ }

69 │ }

70 │ public int addOrderTaiQuay(Guid MaChiNhanh, Guid MaNVThuNgan)

71 │ {

72 │ var nv = dbContext.SysUser.FirstOrDefault(x => x.Ma == MaNVThuNgan);

73 │ if (nv == null)

74 │ {

75 │ throw new Exception("User không tồn tại");

76 │ }

77 │ var br = dbContext.SysBranch.FirstOrDefault(x => x.Ma == MaChiNhanh);

78 │ if (br == null)

79 │ {

80 │ throw new Exception("Chi nhánh không tồn tại");

81 │ }

82 │ var item = new OpsOrder

83 │ {

84 │ MaChiNhanh = MaChiNhanh,

85 │ MaChiNhanhInt = br.Nid ?? 0,

86 │ TenKh = nv.FullName,

87 │ MaNvthuNgan = nv.Ma,

88 │ TaiQuay = true,

89 │ };

90 │ dbContext.Add(item);

91 │ dbContext.SaveChanges();

92 │ return item.Ma;

93 │ }

94 │ public void deleteOrder(int MaPhong, string UUID)

95 │ {

96 │ var order = getMyOrders(MaPhong, UUID);

97 │ if(order == null)

98 │ {

99 │ throw new Exception("Đơn hàng không tồn tại");

100 │ }

101 │ dbContext.Remove(order);

102 │ dbContext.SaveChanges();

103 │ }

104 │

105 │ public List<OpsOrder> getDonHangTheoKhoVaPhong(int MaKho, int MaPhong)

106 │ {

107 │ var orderList = dbContext.OpsOrder.Where(order => order.MaPhong == MaPhong

108 │ && dbContext.OpsOrderDetail.Any(detail => detail.MaDonHang == order.Ma && detail.MaKho == MaKho)).ToList...

109 │ return orderList;

110 │ }

111 │

112 │ }

113 │ }

────────────────────────────────────────────────────────────

**📄 OrderTaiQuayController.cs (CS, 548 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using Microsoft.AspNetCore.SignalR;

3 │ using Microsoft.EntityFrameworkCore;

4 │ using QuanLyBanHang.Common;

5 │ using QuanLyBanHang.Common.Attributes;

6 │ using QuanLyBanHang.Models.APIModels;

7 │ using QuanLyBanHang.Models.ExModel.CAT;

8 │ using QuanLyBanHang.Models.ExModel.OPS;

9 │ using QuanLyBanHang.Models.ExModel.StoredProcedureModel;

10 │ using QuanLyBanHang.Models.ExModel.SYS;

11 │ using QuanLyBanHang.Models.QLBHEntity;

12 │ using QuanLyBanHang.Models.VCEntity;

13 │ using QuanLyBanHang.Models.VCEntity.ExModel;

14 │ using QuanLyBanHang.Services;

15 │ using static QuanLyBanHang.Controllers.OPS.BanHangController;

16 │ using static QuanLyBanHang.Models.ExModel.OPS.OrderDetailModel;

17 │

18 │ namespace QuanLyBanHang.Controllers.OPS

19 │ {

20 │ public class OrderTaiQuayController : BaseController

21 │ {

22 │ private QuanLyVoucherContext vcContext { get; set; }

23 │ private IEInvoiceService \_service;

24 │ private readonly IHubContext<ChatHUB> \_chatHub;

25 │ public class TableList

26 │ {

27 │ public int Ma { get; set; }

28 │ public string? Ten { get; set; }

29 │ public string? TrangThai { get; set; }

30 │ public int SoDon { get; set; }

31 │ }

32 │ public class OrderItemClient

33 │ {

34 │ public int MaHangHoa { get; set; }

35 │ public int SoLuong { get; set; }

36 │ }

37 │ public OrderTaiQuayController(QuanLyBanHangContext dbcontext, QuanLyVoucherContext voucherContext, IEInvoiceServ...

38 │ {

39 │ m\_MenuID = MenuID.opsOrderTaiQuay.GetEnumGuid();

40 │ vcContext = voucherContext;

41 │ \_service = service;

42 │ \_chatHub = chatHub;

43 │

44 │ }

45 │ [TokenUserCheckHTTP]

46 │ public ActionResult Index(string MaKhoHang)

47 │ {

48 │ int MaKho = Cast.ToInt((MaKhoHang + "").Decrypt());

49 │ if (MaKho <= 0)

50 │ {

51 │ return RedirectToAction("ChonKho");

52 │ }

53 │ var user = new sys\_UserModel(dbcontext).GetItem(objToken.userid);

54 │ string[] strKhoHang = (user.KhoHang + "").Split(';');

55 │ if (!strKhoHang.Contains(Cast.ToInt(MaKho).ToString()))

56 │ {

57 │ return RedirectToAction("ChonKho");

58 │ }

59 │

60 │ DateTime NgayHD = new NgayKetCaModel(dbcontext).LayNgayKetCa(MaKho);

61 │ ViewData["MaKho"] = MaKho;

62 │ ViewData["TenKho"] = new DanhMucKhoModel(dbcontext).GetItem(MaKho).Ten;

63 │

64 │ return View();

65 │ }

66 │ [TokenUserCheckHTTP]

67 │ public ActionResult ChonKho()

68 │ {

69 │ var user = new sys\_UserModel(dbcontext).GetItem(objToken.userid);

70 │

71 │

72 │ string[] strKhoHang = (user.KhoHang + "").Split(';');

73 │ List<int> lstIDKhoHang = new List<int>();

74 │ foreach (string item in strKhoHang)

75 │ {

76 │ if (Cast.ToInt(item.Trim()) > 0)

77 │ {

78 │ lstIDKhoHang.Add(Cast.ToInt(item.Trim()));

79 │ }

80 │ }

81 │

82 │ ViewData["ListKhoHang"] = new DanhMucKhoModel(dbcontext).GetDataByIds(lstIDKhoHang, objToken.branch).Where(t...

83 │

84 │ return View("ChonKho");

85 │ }

86 │ [HttpPost]

87 │ [TokenUserCheckHTTP]

88 │ public ActionResult getMenuData(int MaKho)

89 │ {

90 │ DateTime NgayHD = new NgayKetCaModel(dbcontext).LayNgayKetCa(MaKho);

91 │ new TonKhoModel(dbcontext).TinhTonKhoBanHang(MaKho, string.Format("{0:yyyyMM}", NgayHD));

92 │

93 │ var listData = new GiaBanTheoKhoModel(dbcontext).DanhSachTonKhoHangHoaVaGiaBanTheoKho(Cast.ToInt(MaKho), str...

94 │ return Json(new ResponseModel(QLBHConst.OK, "", listData));

95 │ }

96 │ [HttpPost]

97 │ [TokenUserCheckHTTP]

98 │ public ActionResult getTableList(string MaKho)

99 │ {

100 │ var phong = dbcontext.TblPhongBenh.Where(p => !(p.Deleted ?? false) && p.MaChiNhanh == objToken.branch).ToLi...

101 │ var listData = phong.Select(p => new

102 │ {

103 │ p.Ma,

104 │ p.Ten,

105 │ SoDon = (from od in dbcontext.OpsOrderDetail.Where(x => !(x.DaThucHien ?? false) && x.MaKho == C...

106 │ join o in dbcontext.OpsOrder.Where(x => x.MaPhong == p.Ma && !(x.DaThucHien ?? false)) ...

107 │ where od.TrangThai < (int)OrderStatus.Paid

108 │ select new{ od.Ma, od.TrangThai }).ToList().Count()

109 │

110 │ })

111 │ .ToList()

112 │ .Select(x => new

113 │ {

114 │ x.Ma,

115 │ x.Ten,

116 │ x.SoDon,

117 │ TrangThai = x.SoDon == 0 ? "available" : "inused"

118 │ })

119 │ .ToList();

120 │

121 │

122 │ return Json(new ResponseModel(QLBHConst.OK, "", listData));

123 │

124 │ }

125 │ [HttpPost]

126 │ [TokenUserCheckHTTP]

127 │ public ActionResult getOrderList(string MaKho)

128 │ {

129 │ var orderList = (from o in dbcontext.OpsOrder.Where(x => !(x.DaThucHien ?? false) && x.MaPhong != null)

130 │ join od in dbcontext.OpsOrderDetail.Where(x => x.MaKho == Cast.ToInt(MaKho)) on o.Ma equals...

131 │ where od.TrangThai < (int)OrderStatus.Paid

132 │ select new

133 │ {

134 │ MaDonHang = o.Ma,

135 │ MaDonHangChiTiet = od.Ma,

136 │ MaPhong = o.MaPhong,

137 │ TenKH = o.TenKh,

138 │ TrangThai = od.TrangThai,

139 │ ThoiGianDatHang = od.ThoiGianDatMon == null ? null : od.ThoiGianDatMon.Value.ToString("...

140 │ ThoiGianXacNhan = od.ThoiGianXacNhan == null ? null : od.ThoiGianXacNhan.Value.ToString...

141 │ }).ToList();

142 │ var listData = orderList.Select(t => new

143 │ {

144 │ MaDonHang = t.MaDonHang,

145 │ MaDonHangChiTiet = t.MaDonHangChiTiet,

146 │ MaPhong = t.MaPhong,

147 │ TenKH = t.TenKH,

148 │ TrangThai = t.TrangThai,

149 │ DanhSachMon = getDanhSachMon(t.MaDonHangChiTiet),

150 │ TongTien = getDanhSachMon(t.MaDonHangChiTiet).Sum(x => (x.GiaBan ?? 0) \* (x.SoLuong ?? 0)),

151 │ ThoiGianDatHang = t.ThoiGianDatHang,

152 │ ThoiGianXacNhan = t.ThoiGianXacNhan,

153 │ }).ToList();

154 │ return Json(new ResponseModel(QLBHConst.OK, "", listData));

155 │ }

156 │ private List<OrderItemModel> getDanhSachMon(string MaDonHangChiTiet)

157 │ {

158 │ var listData = (from oi in dbcontext.OpsOrderItems.Where(x => x.MaDonHangChiTiet == MaDonHangChiTiet)

159 │ join hh in dbcontext.TblHangHoa.Where(x => !(x.Deleted ?? false)) on oi.MaHangHoa equals hh.Ma

160 │ select new OrderItemModel

161 │ {

162 │ Ma = oi.Ma,

163 │ MaDonHang = oi.MaDonHang,

164 │ MaDonHangChiTiet = oi.MaDonHangChiTiet,

165 │ MaHangHoa = oi.MaHangHoa,

166 │ TenHangHoa = hh.Ten,

167 │ HinhAnh = hh.HinhAnh,

168 │ GiaBan = oi.GiaBan,

169 │ SoLuong = oi.SoLuong,

170 │ ThanhTien = oi.ThanhTien,

171 │ GhiChu = oi.GhiChu,

172 │

173 │ }).ToList();

174 │ return listData;

175 │ }

176 │

177 │ [HttpPost]

178 │ [TokenUserCheckHTTP]

179 │ public ActionResult newOrder(string TenKH, int MaPhong, string MaKho, List<OrderItemClient> DanhSachMon)

180 │ {

181 │ var phong = dbcontext.TblPhongBenh.FirstOrDefault(x => x.Ma == MaPhong);

182 │ if(phong == null) return Json(new ResponseModelError("Phòng không tồn tại"));

183 │ var br = dbcontext.SysBranch.FirstOrDefault(x => x.Ma == objToken.branch);

184 │ if (br == null) return Json(new ResponseModelError("Chi nhánh không tồn tại"));

185 │ var kho = dbcontext.StockDanhMucKho.FirstOrDefault(x => x.Ma == Cast.ToInt(MaKho));

186 │ if (kho == null) return Json(new ResponseModelError("Kho không tồn tại"));

187 │ using (var transaction = dbcontext.Database.BeginTransaction())

188 │ {

189 │ try

190 │ {

191 │ var order = new OpsOrder

192 │ {

193 │ TenKh = TenKH,

194 │ MaPhong = MaPhong,

195 │ CodePhong = phong.Code,

196 │ TaiQuay = true,

197 │ MaNvthuNgan = objToken.userid,

198 │ MaChiNhanh = objToken.branch,

199 │ MaChiNhanhInt = br.Nid ?? 0,

200 │ };

201 │ dbcontext.Add(order);

202 │ dbcontext.SaveChanges();

203 │

204 │ var orderDetail = new OpsOrderDetail

205 │ {

206 │ Ma = "HD" + DateTime.Now.ToString("yyyyMMddHHmmssfff"),

207 │ MaChiNhanhInt = br.Nid ?? 0,

208 │ MaDonHang = order.Ma,

209 │ TrangThai = (int)OrderStatus.Confirmed,

210 │ GhiChu = "",

211 │ ThoiGianDatMon = DateTime.Now,

212 │ ThoiGianXacNhan = DateTime.Now,

213 │ MaKho = Cast.ToInt(MaKho),

214 │ MaNganh = kho.MaNganhVoucher,

215 │

216 │ };

217 │ dbcontext.Add(orderDetail);

218 │ dbcontext.SaveChanges();

219 │ var menuData = new GiaBanTheoKhoModel(dbcontext).DanhSachTonKhoHangHoaVaGiaBanTheoKho(Cast.ToInt(MaK...

220 │

221 │ var orderItems = DanhSachMon.Select(t => new OpsOrderItems

222 │ {

223 │ Ma = Utility.RandomString(7),

224 │ MaDonHang = order.Ma,

225 │ MaDonHangChiTiet = orderDetail.Ma,

226 │ MaHangHoa = t.MaHangHoa,

227 │ GiaBan = menuData.FirstOrDefault(x => x.MaHangHoa == t.MaHangHoa)?.GiaBan ?? 0,

228 │ SoLuong = t.SoLuong,

229 │ ThanhTien = (menuData.FirstOrDefault(x => x.MaHangHoa == t.MaHangHoa)?.GiaBan ?? 0) \* t.SoLuong,

230 │ GhiChu = ""

231 │

232 │ });

233 │ dbcontext.AddRange(orderItems);

234 │ dbcontext.SaveChanges();

235 │ transaction.Commit();

236 │ var orderItemsList = orderItems.Select(t => new

237 │ {

238 │ Ma = t.Ma,

239 │ MaDonHang = order.Ma,

240 │ MaDonHangChiTiet = orderDetail.Ma,

241 │ MaHangHoa = t.MaHangHoa,

242 │ TenHangHoa = menuData.FirstOrDefault(x => x.MaHangHoa == t.MaHangHoa)?.TenHang ?? "N/A",

243 │ GiaBan = menuData.FirstOrDefault(x => x.MaHangHoa == t.MaHangHoa)?.GiaBan ?? 0,

244 │ SoLuong = t.SoLuong,

245 │ ThanhTien = (menuData.FirstOrDefault(x => x.MaHangHoa == t.MaHangHoa)?.GiaBan ?? 0) \* t.SoLuong,

246 │ GhiChu = ""

247 │ });

248 │ var orderData = new

249 │ {

250 │ MaDonHang = order.Ma,

251 │ MaDonHangChiTiet = orderDetail.Ma,

252 │ MaPhong = order.MaPhong,

253 │ TenKH = order.TenKh,

254 │ TrangThai = orderDetail.TrangThai,

255 │ DanhSachMon = orderItemsList,

256 │ TongTien = orderItemsList.Sum(x => x.GiaBan \* x.SoLuong ),

257 │ ThoiGianDatHang = orderDetail.ThoiGianDatMon == null ? null : orderDetail.ThoiGianDatMon.Value.T...

258 │ ThoiGianXacNhan = orderDetail.ThoiGianXacNhan == null ? null : orderDetail.ThoiGianXacNhan.Value...

259 │ };

260 │

261 │ return Json(new ResponseModelSuccess(QLBHConst.AddSuccessfully, orderData));

262 │ }

263 │ catch (Exception ex)

264 │ {

265 │ transaction.Rollback();

266 │ return Json(new ResponseModelError(ex.Message));

267 │ }

268 │ }

269 │ }

270 │ [HttpPost]

271 │ [TokenUserCheckHTTP]

272 │ public ActionResult updateOrder(string MaDonHangChiTiet, string TenKH, string MaKho, List<OrderItemClient> DanhS...

273 │ {

274 │ var orderDetail = dbcontext.OpsOrderDetail.FirstOrDefault(x => x.Ma == MaDonHangChiTiet);

275 │ if (orderDetail == null) return Json(new ResponseModelError("Đơn hàng chi tiết không tồn tại"));

276 │ var order = dbcontext.OpsOrder.FirstOrDefault(x => x.Ma == orderDetail.MaDonHang);

277 │ if (order == null) return Json(new ResponseModelError("Đơn hàng không tồn tại"));

278 │

279 │ var orderItemList = dbcontext.OpsOrderItems.Where(x => x.MaDonHang == orderDetail.MaDonHang && x.MaDonHangCh...

280 │ using (var transaction = dbcontext.Database.BeginTransaction())

281 │ {

282 │ try

283 │ {

284 │ order.TenKh = TenKH;

285 │ dbcontext.Update(order);

286 │ dbcontext.RemoveRange(orderItemList);

287 │ dbcontext.SaveChanges();

288 │ var menuData = new GiaBanTheoKhoModel(dbcontext).DanhSachTonKhoHangHoaVaGiaBanTheoKho(Cast.ToInt(MaK...

289 │

290 │ var neworderItems = DanhSachMon.Select(t => new OpsOrderItems

291 │ {

292 │ Ma = Utility.RandomString(7),

293 │ MaDonHang = order.Ma,

294 │ MaDonHangChiTiet = orderDetail.Ma,

295 │ MaHangHoa = t.MaHangHoa,

296 │ GiaBan = menuData.FirstOrDefault(x => x.MaHangHoa == t.MaHangHoa)?.GiaBan ?? 0,

297 │ SoLuong = t.SoLuong,

298 │ ThanhTien = (menuData.FirstOrDefault(x => x.MaHangHoa == t.MaHangHoa)?.GiaBan ?? 0) \* t.SoLuong,

299 │ GhiChu = ""

300 │

301 │ }).ToList();

302 │ dbcontext.AddRange(neworderItems);

303 │ dbcontext.SaveChanges();

304 │ transaction.Commit();

305 │ var orderItemsList = neworderItems.Select(t => new

306 │ {

307 │ Ma = t.Ma,

308 │ MaDonHang = order.Ma,

309 │ MaDonHangChiTiet = orderDetail.Ma,

310 │ MaHangHoa = t.MaHangHoa,

311 │ TenHangHoa = menuData.FirstOrDefault(x => x.MaHangHoa == t.MaHangHoa)?.TenHang ?? "N/A",

312 │ GiaBan = menuData.FirstOrDefault(x => x.MaHangHoa == t.MaHangHoa)?.GiaBan ?? 0,

313 │ SoLuong = t.SoLuong,

314 │ ThanhTien = (menuData.FirstOrDefault(x => x.MaHangHoa == t.MaHangHoa)?.GiaBan ?? 0) \* t.SoLuong,

315 │ GhiChu = ""

316 │ });

317 │ var orderData = new

318 │ {

319 │ MaDonHang = order.Ma,

320 │ MaDonHangChiTiet = orderDetail.Ma,

321 │ MaPhong = order.MaPhong,

322 │ TenKH = order.TenKh,

323 │ TrangThai = orderDetail.TrangThai,

324 │ DanhSachMon = orderItemsList,

325 │ TongTien = neworderItems.Sum(x => x.GiaBan \* x.SoLuong),

326 │ ThoiGianDatHang = orderDetail.ThoiGianDatMon == null ? null : orderDetail.ThoiGianDatMon.Value.T...

327 │ ThoiGianXacNhan = orderDetail.ThoiGianXacNhan == null ? null : orderDetail.ThoiGianXacNhan.Value...

328 │ };

329 │ return Json(new ResponseModelSuccess(QLBHConst.UpdateSuccessfully, orderData));

330 │ }

331 │ catch (Exception ex)

332 │ {

333 │ transaction.Rollback();

334 │ return Json(new ResponseModelError(ex.Message));

335 │ }

336 │ }

337 │ }

338 │

339 │ [HttpPost]

340 │ [TokenUserCheckHTTP]

341 │ public ActionResult confirmOrder(string MaDonHangChiTiet)

342 │ {

343 │ var orderDetail = dbcontext.OpsOrderDetail.FirstOrDefault(x => x.Ma == MaDonHangChiTiet);

344 │ if (orderDetail == null) return Json(new ResponseModelError("Đơn hàng chi tiết không tồn tại"));

345 │ var order = dbcontext.OpsOrder.FirstOrDefault(x => x.Ma == orderDetail.MaDonHang);

346 │ orderDetail.TrangThai = (int)OrderStatus.Confirmed;

347 │ dbcontext.Update(orderDetail);

348 │ dbcontext.SaveChanges();

349 │ \_chatHub.Clients.Group(order.Uuid + "").SendAsync("confirmOrder", new

350 │ {

351 │ MaDonHang = orderDetail.Ma,

352 │ MaKho = orderDetail.MaKho,

353 │

354 │ });

355 │ \_chatHub.Clients.Group("OrderGroup").SendAsync("confirmOrder", new

356 │ {

357 │ MaDonHang = orderDetail.Ma,

358 │ MaKho = orderDetail.MaKho,

359 │

360 │ });

361 │ return Json(new ResponseModelSuccess("Xác nhận đơn hàng thành công"));

362 │ }

363 │ [HttpPost]

364 │ [TokenUserCheckHTTP]

365 │ public ActionResult cancelOrder(string MaDonHangChiTiet)

366 │ {

367 │ var orderDetail = dbcontext.OpsOrderDetail.FirstOrDefault(x => x.Ma == MaDonHangChiTiet);

368 │ if (orderDetail == null) return Json(new ResponseModelError("Đơn hàng chi tiết không tồn tại"));

369 │ var order = dbcontext.OpsOrder.FirstOrDefault(x => x.Ma == orderDetail.MaDonHang);

370 │ orderDetail.TrangThai = (int)OrderStatus.Cancel;

371 │ dbcontext.Update(orderDetail);

372 │ dbcontext.SaveChanges();

373 │ \_chatHub.Clients.Group(order.Uuid + "").SendAsync("cancelOrder", new

374 │ {

375 │ MaDonHang = orderDetail.Ma,

376 │ MaKho = orderDetail.MaKho,

377 │

378 │ });

379 │ \_chatHub.Clients.Group("OrderGroup").SendAsync("cancelOrder", new

380 │ {

381 │ MaDonHang = orderDetail.Ma,

382 │ MaKho = orderDetail.MaKho,

383 │

384 │ });

385 │ return Json(new ResponseModelSuccess("Hủy đơn hàng thành công"));

386 │ }

387 │ [HttpPost]

388 │ [TokenUserCheckHTTP]

389 │ public async Task<ActionResult> thanhtoan(string HinhThuc\_TT, int ThanhTien, int KhachDua, int HoanLai, string G...

390 │ {

391 │ var br = dbcontext.SysBranch.FirstOrDefault(x => x.Ma == objToken.branch);

392 │ if (br == null) return Json(new ResponseModelError("Chi nhánh không tồn tại"));

393 │ var kho = dbcontext.StockDanhMucKho.FirstOrDefault(x => x.Ma == Cast.ToInt(MaKho));

394 │ if (kho == null) return Json(new ResponseModelError("Kho không tồn tại"));

395 │

396 │ var menuData = new GiaBanTheoKhoModel(dbcontext).DanhSachTonKhoHangHoaVaGiaBanTheoKho(Cast.ToInt(MaKho), str...

397 │ var listChiTiet = DanhSachMon.Select(t => new cCT\_HoaDon

398 │ {

399 │ Ma = 0.Encrypt(),

400 │ MaHangHoa = t.MaHangHoa + "",

401 │ MaVach = menuData.FirstOrDefault(x => x.MaHangHoa == t.MaHangHoa)?.MaVach + "",

402 │ SoLuong = t.SoLuong + "",

403 │ DonGia = menuData.FirstOrDefault(x => x.MaHangHoa == t.MaHangHoa)?.GiaBan + "",

404 │ GiaGiam = "0",

405 │ ThanhTien = (t.SoLuong \* menuData.FirstOrDefault(x => x.MaHangHoa == t.MaHangHoa)?.GiaBan ?? 0) + "",

406 │ }).ToList();

407 │ if (!MaVachThe.IsEmpty())

408 │ {

409 │ var voucherRs = new ThanhToanVoucherModel(vcContext).GetVoucher(MaVachThe ?? "", br.MaChiNhanhVoucher ??...

410 │ var voucher = voucherRs.Voucher;

411 │ int? Result = voucherRs.Result;

412 │ if ((Result ?? 0) == 0)

413 │ {

414 │ return Json(new ResponseModel(QLBHConst.Error, "Thẻ không tồn tại"));

415 │ }

416 │ else if ((Result ?? 0) == -1)

417 │ {

418 │ return Json(new ResponseModel(QLBHConst.Error, "Thẻ không áp dụng tại chi nhánh này"));

419 │ }

420 │ else if ((Result ?? 0) == -2)

421 │ {

422 │ return Json(new ResponseModel(QLBHConst.Error, "Thẻ không áp dụng cho ngành này"));

423 │ }

424 │ else if ((Result ?? 0) == -3)

425 │ {

426 │ return Json(new ResponseModel(QLBHConst.Error, "Thẻ chưa kích hoạt"));

427 │ }

428 │ else if ((Result ?? 0) == -4)

429 │ {

430 │ return Json(new ResponseModel(QLBHConst.Error, "Thẻ chưa đến ngày sử dụng"));

431 │ }

432 │ else if ((Result ?? 0) == -5)

433 │ {

434 │ return Json(new ResponseModel(QLBHConst.Error, "Thẻ đã hết hạn sử dụng"));

435 │ }

436 │ }

437 │ if (Cast.ToInt(HinhThuc\_TT) == (int)HinhThucThanhToan.ChuyenKhoan && RequestID.IsEmpty())

438 │ {

439 │ return Json(new ResponseModel(QLBHConst.Error, "Mã thanh toán không hợp lệ"));

440 │ }

441 │ if (Cast.ToInt(HoanLai) < 0 && Cast.ToInt(HinhThuc\_TT) != (int)HinhThucThanhToan.AR && Cast.ToInt(HinhThuc\_T...

442 │ {

443 │ return Json(new ResponseModel(QLBHConst.Error, "Số tiền khách đưa không đủ"));

444 │ }

445 │ if (Cast.ToInt(HinhThuc\_TT) != (int)HinhThucThanhToan.TienMat && Cast.ToInt(HinhThuc\_TT) != (int)HinhThucTha...

446 │ && Cast.ToInt(HinhThuc\_TT) != (int)HinhThucThanhToan.The && Cast.ToInt(HinhThuc\_TT) != (int)HinhThucThan...

447 │ {

448 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng chọn hình thức thanh toán"));

449 │ }

450 │ cHoaDonBan res = new HoaDonBanModel(dbcontext, vcContext).Add(objToken.branch, Cast.ToInt(HinhThuc\_TT), 0, C...

451 │ if (res.Result == QLBHConst.OK)

452 │ {

453 │ List<int> hoaDons = new List<int>

454 │ {

455 │ Cast.ToInt(res.ResultText)

456 │ };

457 │ foreach(var MaDonHangChiTiet in ListMaDonHang)

458 │ {

459 │ var orderDetail = dbcontext.OpsOrderDetail.FirstOrDefault(x => x.Ma == MaDonHangChiTiet);

460 │ var order = dbcontext.OpsOrder.FirstOrDefault(x => x.Ma == orderDetail.MaDonHang);

461 │ orderDetail.TrangThai = (int)OrderStatus.Paid;

462 │ orderDetail.DaThucHien = true;

463 │ dbcontext.Update(orderDetail);

464 │ dbcontext.SaveChanges();

465 │ var orderDetailList = dbcontext.OpsOrderDetail.Where(x => x.MaDonHang == order.Ma).ToList();

466 │ var paidOrderList = orderDetailList.Where(x => x.TrangThai == (int)OrderStatus.Paid).ToList();

467 │ if(orderDetailList.Count == paidOrderList.Count)

468 │ {

469 │ order.DaThucHien = true;

470 │ dbcontext.Update(order);

471 │ dbcontext.SaveChanges();

472 │ }

473 │ await \_chatHub.Clients.Group(order.Uuid + "").SendAsync("payOrder", new

474 │ {

475 │ MaDonHang = orderDetail.Ma,

476 │ MaKho = orderDetail.MaKho,

477 │

478 │ });

479 │ await \_chatHub.Clients.Group("OrderGroup").SendAsync("payOrder", new

480 │ {

481 │ MaDonHang = orderDetail.Ma,

482 │ MaKho = orderDetail.MaKho,

483 │

484 │ });

485 │ }

486 │ TayNinhServiceBackgroundWorker.AddJob(new TayNinhServiceJob { ListMaHoaDon = hoaDons });

487 │ var MaHoaDon = Cast.ToInt(res.ResultText);

488 │ OpsHoaDon item = new HoaDonBanModel(dbcontext, vcContext).GetItem(MaHoaDon) ?? new OpsHoaDon();

489 │ List<sp\_ChiTietHoaDonResult> ListChitiet = new ChiTietHoaDonModel(dbcontext).GetFullData(item.Ma);

490 │ var usr = new sys\_UserModel(dbcontext).GetItem(objToken?.userid);

491 │

492 │ return Json(new ResponseModel(QLBHConst.OK, QLBHConst.AddSuccessfully, new

493 │ {

494 │ hoadon = item,

495 │ cthoadon = ListChitiet,

496 │ user = usr

497 │ }));

498 │ }

499 │ else

500 │ {

501 │ return Json(new ResponseModel(res.Result, res.ResultText));

502 │ }

503 │ }

504 │

505 │ // Thêm method này vào OrderTaiQuayController.cs

506 │

507 │ [HttpPost]

508 │ [TokenUserCheckHTTP]

509 │ public ActionResult ChonKhoApp()

510 │ {

511 │ try

512 │ {

513 │ var user = new sys\_UserModel(dbcontext).GetItem(objToken.userid);

514 │ if (user == null)

515 │ {

516 │ return Json(new ResponseModelError("Không tìm thấy thông tin người dùng"));

517 │ }

518 │

519 │ string[] strKhoHang = (user.KhoHang + "").Split(';');

520 │ List<int> lstIDKhoHang = new List<int>();

521 │

522 │ foreach (string item in strKhoHang)

523 │ {

524 │ if (Cast.ToInt(item.Trim()) > 0)

525 │ {

526 │ lstIDKhoHang.Add(Cast.ToInt(item.Trim()));

527 │ }

528 │ }

529 │

530 │ if (lstIDKhoHang.Count == 0)

531 │ {

532 │ return Json(new ResponseModel(QLBHConst.OK, "", new List<object>()));

533 │ }

534 │

535 │ var listKhoHang = new DanhMucKhoModel(dbcontext)

536 │ .GetDataByIds(lstIDKhoHang, objToken.branch)

537 │ .Where(t => (t.BanHang ?? false) && (t.Bartender ?? false))

538 │ .ToList();

539 │

540 │ return Json(new ResponseModel(QLBHConst.OK, "", listKhoHang));

541 │ }

542 │ catch (Exception ex)

543 │ {

544 │ return Json(new ResponseModelError($"Lỗi khi lấy danh sách kho hàng: {ex.Message}"));

545 │ }

546 │ }

547 │ }

548 │ }

────────────────────────────────────────────────────────────

**📄 PageNotFoundController.cs (CS, 28 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using QuanLyBanHang.Common.Attributes;

3 │

4 │ namespace QuanLyBanHang.Controllers.SYS

5 │ {

6 │ public class PageNotFoundController : Controller

7 │ {

8 │ public ActionResult Index()

9 │ {

10 │ return View();

11 │ }

12 │

13 │ public ActionResult BaoTri()

14 │ {

15 │ return View();

16 │ }

17 │ public ActionResult LoginAsDenied()

18 │ {

19 │ return View();

20 │ }

21 │

22 │ [TokenUserCheckHTTP()]

23 │ public ActionResult PermissionDenied()

24 │ {

25 │ return View();

26 │ }

27 │ }

28 │ }

────────────────────────────────────────────────────────────

**📄 PermissionController.cs (CS, 117 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using QuanLyBanHang.Common;

3 │ using QuanLyBanHang.Common.Attributes;

4 │ using QuanLyBanHang.Models.APIModels;

5 │ using QuanLyBanHang.Models.ExModel.SYS;

6 │ using QuanLyBanHang.Models.QLBHEntity;

7 │

8 │ namespace QuanLyBanHang.Controllers.SYS

9 │ {

10 │ public class PermissionController : BaseController

11 │ {

12 │

13 │ public PermissionController(QuanLyBanHangContext dbcontext, MemoryCacheManager memoryCache) : base(dbcontext, me...

14 │ {

15 │ m\_MenuID = MenuID.sysPermission.GetEnumGuid();

16 │ }

17 │

18 │ [TokenUserCheckHTTP()]

19 │ public IActionResult Index()

20 │ {

21 │ ViewData["ListData"] = new sys\_PermissionModel(dbcontext).GetData();

22 │ return View();

23 │ }

24 │

25 │ [TokenUserCheckHTTP(Permission.Add)]

26 │ public ActionResult Add()

27 │ {

28 │ return View("Form");

29 │ }

30 │

31 │ [TokenUserCheckHTTP(Permission.Edit)]

32 │ public ActionResult Edit(string ID)

33 │ {

34 │ ViewData["item"] = new sys\_PermissionModel(dbcontext).GetItem(ID.GetGuid());

35 │ return View("Form");

36 │ }

37 │

38 │ [HttpPost]

39 │ [TokenUserCheckHTTP(Permission.Edit)]

40 │ public ActionResult UpdateAPI(string ID, string Code, string Name, string Description, string FormAccess, string...

41 │ {

42 │ int result = new sys\_PermissionModel(dbcontext).Edit(ID.GetGuid(), Code, Name, Description, FormAccess == "1...

43 │ , Status == "1", objToken!.userid);

44 │ return Json(Utility.MakeResponseFromEditResult(Cast.ToString(result)));

45 │ }

46 │

47 │ [HttpPost]

48 │ [TokenUserCheckHTTP(Permission.Add)]

49 │ public ActionResult AddAPI(string Code, string Name, string Description, string FormAccess, string Status)

50 │ {

51 │ int result = new sys\_PermissionModel(dbcontext).Add(Code, Name, Description, FormAccess == "1", Status == "1...

52 │ return Json(Utility.MakeResponseFromAddResult(Cast.ToString(result)));

53 │ }

54 │

55 │ [HttpPost]

56 │ [TokenUserCheckHTTP(Permission.Delete)]

57 │ public ActionResult DeleteAPI(string ID)

58 │ {

59 │ string result = Cast.ToString(new sys\_PermissionModel(dbcontext).Delete(ID.GetGuid(), objToken!.userid));

60 │ return Json(Utility.MakeResponseFromDeleteResult(Cast.ToString(result)));

61 │ }

62 │

63 │ [HttpPost]

64 │ [TokenUserCheckHTTP(Permission.NotCheck)]

65 │ public ResponseModel GetQuyen()

66 │ {

67 │ List<dynamic> data = new List<dynamic>();

68 │ List<SysMenu> menu = new sys\_MenuModel(dbcontext).GetData();

69 │ List<VPermission> permissions = dbcontext.VPermission

70 │ .Where(t => t.UserId == objToken!.userid && t.BranchId == objToken.branch)

71 │ .ToList();

72 │

73 │ foreach (VPermission p in permissions.GroupBy(t => t.MenuId).Select(t => t.FirstOrDefault()).ToList())

74 │ {

75 │ List<Guid> permissionid = new List<Guid>();

76 │ List<VPermission> pp = permissions.Where(t => t.MenuId == p?.MenuId).ToList();

77 │ foreach (VPermission p2 in permissions.Where(t => t.MenuId == p?.MenuId))

78 │ {

79 │ permissionid.Add(p2?.PermissionId ?? Utility.defaultUID);

80 │ }

81 │ data.Add(new

82 │ {

83 │ MenuID = p?.MenuId,

84 │ MenuName = (menu.FirstOrDefault(t => t.Ma == p?.MenuId) ?? new SysMenu() { Name = "" }).Name,

85 │ PermissionList = permissionid

86 │ });

87 │ }

88 │ return new ResponseModel(QLBHConst.OK, "", data);

89 │ }

90 │

91 │ //[HttpPost]

92 │ //[TokenUserCheckHTTP(Permission.NotCheck)]

93 │ //public ResponseModel GetQuyenAPI()

94 │ //{

95 │ // List<dynamic> data = new List<dynamic>();

96 │ // List<SysMenu> menu = new sys\_MenuModel(dbcontext).GetData();

97 │ // List<VPermissionApi> permissions = dbcontext.VPermissionApi.Where(t => t.UserId == objToken!.userid).ToLis...

98 │ // foreach (VPermissionApi p in permissions.GroupBy(t => t.MenuId).Select(t => t.FirstOrDefault()).ToList())

99 │ // {

100 │ // List<int> permissionid = new List<int>();

101 │ // List<VPermissionApi> pp = permissions.Where(t => t.MenuId == p?.MenuId).ToList();

102 │ // foreach (VPermissionApi p2 in permissions.Where(t => t.MenuId == p?.MenuId))

103 │ // {

104 │ // permissionid.Add(p2?.PermissionId ?? 0);

105 │ // }

106 │ // data.Add(new

107 │ // {

108 │ // MenuID = p?.MenuId,

109 │ // ParentID = p?.ParentId,

110 │ // MenuName = (menu.FirstOrDefault(t => t.Id == p?.MenuId) ?? new SysMenu() { Name = "" }).Name,

111 │ // PermissionList = permissionid

112 │ // });

113 │ // }

114 │ // return new ResponseModel(QLBHConst.OK, "", data);

115 │ //}

116 │ }

117 │ }

────────────────────────────────────────────────────────────

**📄 PhongBanController.cs (CS, 72 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using QuanLyBanHang.Common;

3 │ using QuanLyBanHang.Common.Attributes;

4 │ using QuanLyBanHang.Models.APIModels;

5 │ using QuanLyBanHang.Models.QLBHEntity;

6 │

7 │ namespace QuanLyBanHang.Controllers.CAT

8 │ {

9 │ public class PhongBanController : BaseController

10 │ {

11 │ private readonly ILogger<PhongBanController> \_logger;

12 │

13 │ public PhongBanController(

14 │ QuanLyBanHangContext dbcontext,

15 │ MemoryCacheManager memoryCache,

16 │ ILogger<PhongBanController> logger) : base(dbcontext, memoryCache)

17 │ {

18 │ m\_MenuID = MenuID.tblPhongBan.GetEnumGuid();

19 │ \_logger = logger;

20 │ }

21 │

22 │ [TokenUserCheckHTTP(Permission.NotCheck)]

23 │ public IActionResult GetPhongBan()

24 │ {

25 │ \_logger.LogInformation("Người dùng {UserId} yêu cầu lấy danh sách Phòng ban.", objToken.userid);

26 │ try

27 │ {

28 │ Guid chinhanh = Utility.defaultUID;

29 │

30 │ var listData = (from pb in dbcontext.TblPhongBan

31 │ join cn in dbcontext.SysBranch on pb.MaChiNhanh equals cn.Ma

32 │ join \_ban in dbcontext.TblPhongBan on pb.MaBan equals \_ban.Ma into \_ban

33 │ from ban in \_ban.DefaultIfEmpty()

34 │ where !(pb.Deleted ?? false)

35 │ select new

36 │ {

37 │ ma = pb.Ma + "",

38 │ code = pb.Code + "",

39 │ ten = pb.Ten + "",

40 │ tenCN = cn.Ten + "",

41 │ maCN = cn.Ma,

42 │ maPB = pb.MaBan,

43 │ tenPB = ban.Ten + "",

44 │ ban = (pb.Ban ?? false) ? "1" : "0",

45 │ cosoquy = (pb.CoSoQuy ?? false) ? "1" : "0",

46 │ trangthai = ((pb.Status ?? false) ? "Hoạt Động" : "Không hoạt động"),

47 │ }).ToList();

48 │

49 │ // Logic phân quyền

50 │ if (objToken.branch != Branch.TCT.GetEnumGuid() && Utility.checkPermissionIDModel(objToken.userid, m\_Men...

51 │ {

52 │ \_logger.LogInformation("Lấy thành công {Count} bản ghi Phòng ban (ShowAll).", listData.Count)...

53 │ return Json(new ResponseModel(QLBHConst.OK, "", listData));

54 │ }

55 │ else if (objToken.branch != Branch.TCT.GetEnumGuid())

56 │ {

57 │ var filteredData = listData.Where(x => x.maCN == objToken.branch).ToList();

58 │ \_logger.LogInformation("Lấy thành công {Count} bản ghi Phòng ban cho chi nhánh {MaChiNhanh}....

59 │ return Json(new ResponseModel(QLBHConst.OK, "", filteredData));

60 │ }

61 │

62 │ \_logger.LogInformation("Lấy thành công {Count} bản ghi Phòng ban (TCT).", listData.Count);

63 │ return Json(new ResponseModel(QLBHConst.OK, "", listData));

64 │ }

65 │ catch (Exception ex)

66 │ {

67 │ \_logger.LogError(ex, "Lỗi khi lấy danh sách Phòng ban cho người dùng {UserId}.", objToken.user...

68 │ return Json(new ResponseModel(QLBHConst.Error, "Lỗi hệ thống khi lấy danh sách."));

69 │ }

70 │ }

71 │ }

72 │ }

────────────────────────────────────────────────────────────

**📄 PhongBenhController.cs (CS, 200 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using QuanLyBanHang.Common;

3 │ using QuanLyBanHang.Common.Attributes;

4 │ using QuanLyBanHang.Models.APIModels;

5 │ using QuanLyBanHang.Models.QLBHEntity;

6 │

7 │ namespace QuanLyBanHang.Controllers.CAT

8 │ {

9 │ public class PhongBenhController : BaseController

10 │ {

11 │ private readonly ILogger<PhongBenhController> \_logger;

12 │

13 │ public PhongBenhController(

14 │ QuanLyBanHangContext dbcontext,

15 │ MemoryCacheManager memoryCache,

16 │ ILogger<PhongBenhController> logger) : base(dbcontext, memoryCache)

17 │ {

18 │ m\_MenuID = MenuID.catPhongBenh.GetEnumGuid();

19 │ \_logger = logger;

20 │ }

21 │

22 │ [TokenUserCheckHTTP]

23 │ public IActionResult Index()

24 │ {

25 │ \_logger.LogInformation("Người dùng {UserId} truy cập trang danh sách Phòng bệnh.", objToken.userid...

26 │ ViewData["baseUrl"] = $"{Request.Scheme}://{Request.Host}";

27 │ return View();

28 │ }

29 │

30 │ [TokenUserCheckHTTP(Permission.Add)]

31 │ public ActionResult Add()

32 │ {

33 │ \_logger.LogInformation("Người dùng {UserId} truy cập trang thêm mới Phòng bệnh.", objToken.useri...

34 │ return View("Form");

35 │ }

36 │

37 │ [TokenUserCheckHTTP(Permission.Edit)]

38 │ public ActionResult Edit(string Ma)

39 │ {

40 │ \_logger.LogInformation("Người dùng {UserId} yêu cầu chỉnh sửa Phòng bệnh. Mã (đã mã hóa):...

41 │ try

42 │ {

43 │ Ma = Ma.Decrypt();

44 │ int id = Ma.ToInt();

45 │ ViewData["item"] = dbcontext.TblPhongBenh.FirstOrDefault(t => t.Ma == id && !(t.Deleted ?? false));

46 │ return View("Form");

47 │ }

48 │ catch (Exception ex)

49 │ {

50 │ \_logger.LogError(ex, "Lỗi khi tải dữ liệu để chỉnh sửa Phòng bệnh. Mã (đã mã hóa)...

51 │ return RedirectToAction("Index");

52 │ }

53 │ }

54 │

55 │ [HttpPost]

56 │ [TokenUserCheckHTTP(Permission.Add)]

57 │ public IActionResult AddAPI(string Code, string Ten, string TrangThai, string BanNoiBo)

58 │ {

59 │ \_logger.LogInformation("Người dùng {UserId} thêm mới Phòng bệnh: {TenPhongBenh}", objToken.userid,...

60 │ var result = CheckRequire(Code, Ten, TrangThai);

61 │ if (result != null)

62 │ return Json(result);

63 │

64 │ try

65 │ {

66 │ var item = new TblPhongBenh

67 │ {

68 │ Ten = Ten,

69 │ Code = Code,

70 │ Status = TrangThai == "1",

71 │ NoiBo = BanNoiBo.ToBoolean(),

72 │ UserCreated = objToken.userid,

73 │ CreatedDate = DateTime.Now,

74 │ MaChiNhanh = objToken.branch

75 │ };

76 │ dbcontext.TblPhongBenh.Add(item);

77 │ dbcontext.SaveChanges();

78 │ \_logger.LogInformation("Thêm mới Phòng bệnh ID {PhongBenhId} thành công.", item.Ma);

79 │ return Json(QLBHConst.ResponseAddSuccessfully);

80 │ }

81 │ catch (Exception ex)

82 │ {

83 │ \_logger.LogError(ex, "Lỗi khi thêm mới Phòng bệnh: {TenPhongBenh}", Ten);

84 │ return Json(new ResponseModel(QLBHConst.Error, "Lỗi hệ thống khi thêm mới."));

85 │ }

86 │ }

87 │

88 │ [HttpPost]

89 │ [TokenUserCheckHTTP(Permission.Edit)]

90 │ public ActionResult UpdateAPI(string Ma, string Code, string Ten, string TrangThai, string BanNoiBo)

91 │ {

92 │ \_logger.LogInformation("Người dùng {UserId} cập nhật Phòng bệnh. Mã (đã mã hóa): {MaPhongBen...

93 │ try

94 │ {

95 │ Ma = Ma.Decrypt();

96 │ int id = Ma.ToInt();

97 │ var result = CheckRequire(Code, Ten, TrangThai);

98 │ if (result != null)

99 │ return Json(result);

100 │

101 │ var item = dbcontext.TblPhongBenh.FirstOrDefault(x => x.Ma == id && !(x.Deleted ?? false));

102 │ if (item == null)

103 │ {

104 │ \_logger.LogWarning("Không tìm thấy Phòng bệnh ID: {PhongBenhId} để cập nhật.", id);

105 │ return Json(QLBHConst.ResponseNotExist);

106 │ }

107 │

108 │ item.Ten = Ten;

109 │ item.Status = TrangThai == "1";

110 │ item.Code = Code;

111 │ item.NoiBo = BanNoiBo.ToBoolean();

112 │ item.UserModified = objToken.userid;

113 │ item.ModifiedDate = DateTime.Now;

114 │

115 │ dbcontext.SaveChanges();

116 │ \_logger.LogInformation("Cập nhật Phòng bệnh ID {PhongBenhId} thành công.", id);

117 │ return Json(QLBHConst.ResponseUpdateSuccessfully);

118 │ }

119 │ catch (Exception ex)

120 │ {

121 │ \_logger.LogError(ex, "Lỗi khi cập nhật Phòng bệnh. Mã (đã mã hóa): {MaPhongBenh}", Ma);

122 │ return Json(new ResponseModel(QLBHConst.Error, "Lỗi hệ thống khi cập nhật."));

123 │ }

124 │ }

125 │

126 │ [HttpPost]

127 │ [TokenUserCheckHTTP(Permission.Delete)]

128 │ public IActionResult DeleteAPI(string Ma)

129 │ {

130 │ \_logger.LogInformation("Người dùng {UserId} yêu cầu xóa Phòng bệnh. Mã (đã mã hóa): {MaPhong...

131 │ try

132 │ {

133 │ Ma = Ma.Decrypt();

134 │ int id = Ma.ToInt();

135 │ var item = dbcontext.TblPhongBenh.FirstOrDefault(x => x.Ma == id);

136 │ if (item == null)

137 │ {

138 │ \_logger.LogWarning("Không tìm thấy Phòng bệnh ID: {PhongBenhId} để xóa.", id);

139 │ return Json(QLBHConst.ResponseNotExist);

140 │ }

141 │

142 │ item.Deleted = true;

143 │ item.UserDeleted = objToken.userid;

144 │ item.DeletedDate = DateTime.Now;

145 │ dbcontext.SaveChanges();

146 │

147 │ \_logger.LogInformation("Đã xóa (đánh dấu) Phòng bệnh ID {PhongBenhId} thành công.", id);

148 │ return Json(QLBHConst.ResponseDeleteSuccessfully);

149 │ }

150 │ catch (Exception ex)

151 │ {

152 │ \_logger.LogError(ex, "Lỗi khi xóa Phòng bệnh. Mã (đã mã hóa): {MaPhongBenh}", Ma);

153 │ return Json(new ResponseModel(QLBHConst.Error, "Lỗi hệ thống khi xóa."));

154 │ }

155 │ }

156 │

157 │ private static ResponseModel CheckRequire(String Code, String Ten, String Status)

158 │ {

159 │ if (Code.IsEmpty())

160 │ return new ResponseModel(QLBHConst.Error, "Vui lòng nhập code");

161 │ if (Ten.IsEmpty())

162 │ return new ResponseModel(QLBHConst.Error, "Vui lòng nhập tên");

163 │ if (Status.IsEmpty())

164 │ return new ResponseModel(QLBHConst.Error, "Vui lòng nhập trạng thái");

165 │ return null;

166 │ }

167 │

168 │ [TokenUserCheckHTTP(Permission.NotCheck)]

169 │ public IActionResult getListData()

170 │ {

171 │ \_logger.LogInformation("Người dùng {UserId} yêu cầu lấy danh sách Phòng bệnh.", objToken.userid...

172 │ try

173 │ {

174 │ var listData = dbcontext.TblPhongBenh.Where(x => !(x.Deleted ?? false) && x.MaChiNhanh == objToken.branc...

175 │ .Select(n => new

176 │ {

177 │ maEncrypt = n.Ma.Encrypt(),

178 │ n.Ma,

179 │ MaCu = n.MaCu ?? 0,

180 │ n.Code,

181 │ n.Ten,

182 │ NoiBo = n.NoiBo,

183 │ eNoiBo = (n.NoiBo ?? false).ToString().ToLower().Encrypt(),

184 │ eMa = n.Ma.ToString().Encrypt(),

185 │ eMaCu = (n.MaCu ?? 0).Encrypt(),

186 │ trangthai = n.Status ?? false,

187 │ tentrangthai = n.Status ?? false ? "Hoạt động" : "Hết hoạt động",

188 │ }).OrderBy(t => t.Ma).ToList();

189 │

190 │ \_logger.LogInformation("Lấy thành công {Count} bản ghi Phòng bệnh.", listData.Count);

191 │ return Json(new ResponseModel(QLBHConst.OK, "", listData));

192 │ }

193 │ catch (Exception ex)

194 │ {

195 │ \_logger.LogError(ex, "Lỗi khi lấy danh sách Phòng bệnh cho chi nhánh {MaChiNhanh}", objToken.br...

196 │ return Json(new ResponseModel(QLBHConst.Error, "Lỗi hệ thống khi lấy danh sách."));

197 │ }

198 │ }

199 │ }

200 │ }

────────────────────────────────────────────────────────────

**📄 PhongBenhModel.cs (CS, 21 lines)**

1 │ ﻿using Microsoft.EntityFrameworkCore;

2 │ using QuanLyBanHang.Models.QLBHEntity;

3 │

4 │ namespace QuanLyBanHang.Models.ExModel.CAT

5 │ {

6 │ public class PhongBenhModel : Base<TblPhongBenh>

7 │ {

8 │ public PhongBenhModel(DbContext dbContext) : base(dbContext)

9 │ {

10 │ }

11 │ public TblPhongBenh GetItem(int Ma)

12 │ {

13 │ return First(t => t.Ma == Ma && !(t.Deleted ?? false));

14 │ }

15 │

16 │ public List<TblPhongBenh> GetData(Guid MaChiNhanh)

17 │ {

18 │ return Get(t => !(t.Deleted ?? false) && t.MaChiNhanh == MaChiNhanh).OrderBy(t => t.Ten).ToList();

19 │ }

20 │ }

21 │ }

────────────────────────────────────────────────────────────

**📄 Product.cs (CS, 21 lines)**

1 │ namespace QuanLyBanHang.Models.EInvoiceService

2 │ {

3 │ public class Product

4 │ {

5 │ public string? ProdName { get; set; }

6 │ public string? ProdUnit { get; set; }

7 │ public string? ProdQuantity { get; set; }

8 │ public string? ProdPrice { get; set; }

9 │ public string? Amount { get; set; }

10 │ public string? Remark { get; set; }

11 │ public string? Total { get; set; }

12 │ public string? VATRate { get; set; }

13 │ public string? VATAmount { get; set; }

14 │ public string? Extra1 { get; set; }

15 │ public string? Extra2 { get; set; }

16 │ public string? Discount { get; set; }

17 │ public string? DiscountAmount { get; set; }

18 │ public string? DiscountNonTax { get; set; }

19 │ public string? IsSum { get; set; }

20 │ }

21 │ }

────────────────────────────────────────────────────────────

**📄 QLBHConst.cs (CS, 116 lines)**

1 │ ﻿using QuanLyBanHang.Models.APIModels;

2 │ using YamlDotNet.Core.Tokens;

3 │

4 │ namespace QuanLyBanHang.Common

5 │ {

6 │ public class QLBHConst

7 │ {

8 │ public static readonly string JSVersion = "?ver=20250704.02";

9 │ public static readonly List<string> AllowDocumentExt = new List<string> { ".DOC", ".DOCX", ".XLS", ".XLSX", ".XL...

10 │ public static readonly List<string> AllowImageExt = new List<string> { ".JPG", ".JPEG", ".BMP", ".GIF", ".PNG" }...

11 │ public static readonly List<string> AllowAudioExt = new List<string> { ".MP3", ".AAC", ".WMA", ".WAV" };

12 │ public static readonly List<string> AllowVideoExt = new List<string> { ".AVI", ".MP4", ".WMV", ".FLV", ".3GP" };

13 │ public const int MaxFileSize = int.MaxValue;//MB

14 │

15 │ public static string LogoPath = "/images/tanione-logo.jpg";

16 │ public static string TenCty = "Tanione Store";

17 │ public static string DiaChi = "187 Phạm Văn Đồng - Hoà Thành";

18 │ public static string DienThoai = "0941.696.939";

19 │

20 │ //Bảng kê thu mua hàng hóa, dịch vụ mua vào không hóa đơn

21 │ public static string TenDoanhNghiep = "CHI NHÁNH CÔNG TY CỔ PHẦN Y TẾ HÙNG DUY";

22 │ public static string DiaChiDoanhNghiep = "Số 187, Phạm Văn Đồng, Khu phố Hiệp Long, Phường Hiệp...

23 │ public static string MaSoThue = "3901250811-001";

24 │ public static string DiaChiToChucThuMua = "";

25 │ public static string NguoiPhuTrach = "";

26 │ public static string GiamDocDoanhNghiep = "";

27 │

28 │ public static string Note1 = "- Căn cứ vào số thực tế các mặt hàng trên mà đơn vị mua của ...

29 │ public static string Note2 = "- Đối với doanh nghiệp có tổ chức các trạm nơi thu mua ở nhiều...

30 │

31 │

32 │ public const string UploadPath = @"C:\Uploads\";

33 │ //public const string UploadPath = @"E:\Uploads\";

34 │ public const string DownloadPath = @"/File/DownloadFile?filename=";

35 │ public const string DocumentPath = UploadPath + "documents";

36 │ public const string ImagePath = UploadPath + "images";

37 │ public const string AudioPath = UploadPath + "audios";

38 │ public const string VideoPath = UploadPath + "videos";

39 │ public const string SecurePath = UploadPath + "secure";

40 │ public const string SoDienThoai1 = "02763 836 991";

41 │ public const string SoDienThoai2 = "0941 696939";

42 │ public const string HoSoBenhAnPath = @"C:\Uploads\hosobenhan\";

43 │ public const string risUrl = "http://172.16.28.99:7177/ws/rest/";

44 │ public const string risUrlViewer = risUrl + "viewer?accessionNumber=";

45 │

46 │ public const int smsLimit = 3;

47 │ public const int OTPLenght = 6;

48 │ public static string wwwRoot = "";

49 │

50 │ public const string UserNotFound = "UsernotFound";

51 │ public const string OK = "OK";

52 │ public const string Error = "Error";

53 │ public const string YES = "YES";

54 │ public const string NO = "NO";

55 │ public const string PerrmissionDenied = "PerrmissionDenied";

56 │ public const string TokenIsNotCorrect = "TokenIsNotCorrect";

57 │ public const string OTPRequired = "OTPRequired";

58 │ public const string SDTRequired = "SDTRequired";

59 │ public const string DEVLogin = "DEVLogin";

60 │ public const string Category = "Danh mục";

61 │ public const string System = "Hệ thống";

62 │ public const string Dashboard = "Dashboard";

63 │ public const string Operations = "Vận hành";

64 │ public const string Inactive = "Hết hoạt động";

65 │ public const string Active = "Hoạt động";

66 │

67 │ public const string AddSuccessfully = "Thêm thành công";

68 │ public const string AddFailure = "Thêm thất bại";

69 │ public const string ObjectExists = "Đối tượng đã tồn tại";

70 │ public const string UpdateSuccessfully = "Cập nhật thành công";

71 │ public const string UpdateFailure = "Cập nhật thất bại";

72 │ public const string NotExist = "Đối tượng không tồn tại";

73 │ public const string DeleteSuccessfully = "Xóa thành công";

74 │ public const string DeleteFailure = "Xóa không thành công";

75 │ public const string CannotDelete = "Không thể xóa";

76 │ public const string NoPermission = "Bạn không có quyền thực hiện thao tác này, vui lòng liên hệ ...

77 │ public const string HaveError = "Đã xảy ra lỗi";

78 │ public const string FileError = "Lỗi tập tin";

79 │ public const string LockedObject = "Đối tượng đã bị khoá";

80 │ public const string Successfully = "Thành công";

81 │ public const string NotSuccessfully = "Không thành công";

82 │ public const string CodeCannotEdit = "Mã đối tượng đã phát sinh giao dịch";

83 │ public const string DateFormatIsIncorrect = "Định dạng ngày không đúng";

84 │ public const string NoPermissionHTML = "Bạn không có quyền thực hiện thao tác này,<br/> vui lòng li...

85 │

86 │ public const string cacheKeyMenu = "menu-";

87 │

88 │ public const string URL = @"https://crm.honghunghospital.com.vn";

89 │

90 │ // Thêm các status khác nếu cần

91 │ public const string NotFound = "NotFound";

92 │ public const string BadRequest = "BadRequest";

93 │ public const string Unauthorized = "Unauthorized";

94 │ public const string Conflict = "Conflict";

95 │

96 │ public static ResponseModel ResponseAddSuccessfully = new ResponseModelSuccess(AddSuccessfully);

97 │ public static ResponseModel ResponseAddFailure = new ResponseModelError(AddFailure);

98 │ public static ResponseModel ResponseObjectExists = new ResponseModelError(ObjectExists);

99 │ public static ResponseModel ResponseUpdateSuccessfully = new ResponseModelSuccess(UpdateSuccessfully);

100 │ public static ResponseModel ResponseUpdateFailure = new ResponseModelError(UpdateFailure);

101 │ public static ResponseModel ResponseNotExist = new ResponseModelError(NotExist);

102 │ public static ResponseModel ResponseDeleteSuccessfully = new ResponseModelSuccess(DeleteSuccessfully);

103 │ public static ResponseModel ResponseDeleteFailure = new ResponseModelError(DeleteFailure);

104 │ public static ResponseModel ResponseCannotDelete = new ResponseModelError(CannotDelete);

105 │ public static ResponseModel ResponseNoPermission = new ResponseModelError(NoPermission);

106 │ public static ResponseModel ResponseHaveError = new ResponseModelError(HaveError);

107 │ public static ResponseModel ResponseFileError = new ResponseModelError(FileError);

108 │ public static ResponseModel ResponseLockedObject = new ResponseModelError(LockedObject);

109 │ public static ResponseModel ResponseSuccessfully = new ResponseModelSuccess(Successfully);

110 │ public static ResponseModel ResponseCodetCannotEdit = new ResponseModelError(CodeCannotEdit);

111 │ public static ResponseModel ResponseDateFormatIsIncorrect = new ResponseModelError(DateFormatIsIncorrect);

112 │ public static ResponseModel notFound = new ResponseModelError("Không tìm thấy !!");

113 │

114 │ public const int DISCORD\_COLOR = 5814783;

115 │ }

116 │ }

────────────────────────────────────────────────────────────

**📄 QRAddtionalBean.cs (CS, 27 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │ using System.Linq;

4 │ using System.Text;

5 │ using System.Threading.Tasks;

6 │

7 │ namespace QuanLyBanHang.Common.BIDV

8 │ {

9 │ public class QRAddtionalBean

10 │ {

11 │ public string billNumber { get; set; }

12 │ public string mobile { get; set; }

13 │ public string storeID { get; set; }

14 │ public string loyaltyNumber { get; set; }

15 │ public string referenceID { get; set; }

16 │ public string customerID { get; set; }

17 │ public string terminalID { get; set; }

18 │ public string purpose { get; set; }

19 │ public string expDate { get; set; }

20 │ public string checkSum { get; set; }

21 │ public string consumerAddress { get; set; }

22 │ public string consumerMobile { get; set; }

23 │ public string consumerEmail { get; set; }

24 │ public string consumerData { get; set; }

25 │

26 │ }

27 │ }

────────────────────────────────────────────────────────────

**📄 QRBean.cs (CS, 33 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │ using System.Linq;

4 │ using System.Text;

5 │ using System.Threading.Tasks;

6 │

7 │ namespace QuanLyBanHang.Common.BIDV

8 │ {

9 │ public class QRBean

10 │ {

11 │ public string payLoad { get; set; }

12 │ public string pointOIMethod { get; set; }

13 │ public string masterMerchant { get; set; } // to chuc dang ky like master merchant

14 │ public string merchantCode { get; set; }

15 │ public string merchantCC { get; set; }

16 │ public string ccy { get; set; }

17 │ public string amount { get; set; }

18 │ public string tipAndFee { get; set; }

19 │ public string fixedFee { get; set; }

20 │ public string percentFee { get; set; }

21 │ public string countryCode { get; set; }

22 │ public string merchantName { get; set; }

23 │ public string merchantCity { get; set; }

24 │ public string pinCode { get; set; }

25 │ public string addtionalData { get; set; }

26 │ public string term { get; set; }

27 │ public string crc16 { get; set; }

28 │ public QRAddtionalBean addtionalBean { get; set; }

29 │ public string visa { get; set; }

30 │ public string master { get; set; }

31 │ public string unionPay { get; set; }

32 │ }

33 │ }

────────────────────────────────────────────────────────────

**📄 QRPack.cs (CS, 264 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │ using System.Linq;

4 │ using System.Text;

5 │

6 │ namespace QuanLyBanHang.Common.BIDV

7 │ {

8 │ public static class QRPack

9 │ {

10 │ private static string EXPIRE\_DATE\_EMPTY = "0000000000";

11 │

12 │ public static QRPackBean pack(QRBean bean)

13 │ {

14 │ StringBuilder content = new StringBuilder();

15 │ if (!bean.payLoad.IsEmpty())

16 │ {

17 │ content.Append(QRTag.PAY\_LOAD)

18 │ .Append(padLeft(bean.payLoad.Length))

19 │ .Append(bean.payLoad);

20 │ }

21 │ if (!bean.pointOIMethod.IsEmpty())

22 │ {

23 │ content.Append(QRTag.POINT\_OI\_METHOD)

24 │ .Append(padLeft(bean.pointOIMethod.Length))

25 │ .Append(bean.pointOIMethod);

26 │ }

27 │ if (!bean.visa.IsEmpty())

28 │ {

29 │ content.Append(QRTag.VISA)

30 │ .Append(padLeft(bean.visa.Length))

31 │ .Append(bean.visa);

32 │ }

33 │ if (!bean.master.IsEmpty())

34 │ {

35 │ content.Append(QRTag.MASTERCARD)

36 │ .Append(padLeft(bean.master.Length))

37 │ .Append(bean.master);

38 │ }

39 │ if (!bean.unionPay.IsEmpty())

40 │ {

41 │ content.Append(QRTag.UNIONPAY)

42 │ .Append(padLeft(bean.unionPay.Length))

43 │ .Append(bean.unionPay);

44 │ }

45 │

46 │ string merchantAccount = packMerchant(bean);

47 │ if (!merchantAccount.IsEmpty())

48 │ {

49 │ content.Append(QRTag.MC\_ACCOUNT)

50 │ .Append(padLeft(merchantAccount.Length))

51 │ .Append(merchantAccount);

52 │ }

53 │ if (!bean.merchantCC.IsEmpty())

54 │ {

55 │ content.Append(QRTag.MC\_CATEGORY\_CODE)

56 │ .Append(padLeft(bean.merchantCC.Length))

57 │ .Append(bean.merchantCC);

58 │ }

59 │ if (!bean.ccy.IsEmpty())

60 │ {

61 │ content.Append(QRTag.CCY)

62 │ .Append(padLeft(bean.ccy.Length))

63 │ .Append(bean.ccy);

64 │ }

65 │ if (!bean.amount.IsEmpty())

66 │ {

67 │ content.Append(QRTag.AMOUNT)

68 │ .Append(padLeft(bean.amount.Length))

69 │ .Append(bean.amount);

70 │ }

71 │ if (!bean.tipAndFee.IsEmpty())

72 │ {

73 │ content.Append(QRTag.TIP\_AND\_FEE)

74 │ .Append(padLeft(bean.tipAndFee.Length))

75 │ .Append(bean.tipAndFee);

76 │ }

77 │ if (!bean.fixedFee.IsEmpty())

78 │ {

79 │ content.Append(QRTag.FIXED\_FEE)

80 │ .Append(padLeft(bean.fixedFee.Length))

81 │ .Append(bean.fixedFee);

82 │ }

83 │ if (!bean.percentFee.IsEmpty())

84 │ {

85 │ content.Append(QRTag.PERCENT\_FEE)

86 │ .Append(padLeft(bean.percentFee.Length))

87 │ .Append(bean.percentFee);

88 │ }

89 │ if (!bean.countryCode.IsEmpty())

90 │ {

91 │ content.Append(QRTag.COUNTRY\_CODE)

92 │ .Append(padLeft(bean.countryCode.Length))

93 │ .Append(bean.countryCode);

94 │ }

95 │ if (!bean.merchantName.IsEmpty())

96 │ {

97 │ content.Append(QRTag.MC\_NAME)

98 │ .Append(padLeft(bean.merchantName.Length))

99 │ .Append(bean.merchantName);

100 │ }

101 │ if (!bean.merchantCity.IsEmpty())

102 │ {

103 │ content.Append(QRTag.MC\_CITY)

104 │ .Append(padLeft(bean.merchantCity.Length))

105 │ .Append(bean.merchantCity);

106 │ }

107 │ if (!bean.pinCode.IsEmpty())

108 │ {

109 │ content.Append(QRTag.MC\_PIN\_CODE)

110 │ .Append(padLeft(bean.pinCode.Length))

111 │ .Append(bean.pinCode);

112 │ }

113 │ string addtionalData = packAddtional(bean.addtionalBean);

114 │ bean.addtionalData = addtionalData;

115 │

116 │ if (!addtionalData.IsEmpty())

117 │ {

118 │ content.Append(QRTag.ADDTIONAL\_DATA)

119 │ .Append(padLeft(addtionalData.Length))

120 │ .Append(addtionalData);

121 │ }

122 │

123 │ if (!bean.term.IsEmpty())

124 │ {

125 │ content.Append(QRTag.TERM)

126 │ .Append(padLeft(bean.term.Length))

127 │ .Append(bean.term);

128 │ }

129 │ string dataToCRC = content.Append(QRTag.CRC16).Append(QRTag.CRC\_LENGHT).ToString();

130 │ string crc16 = CRC16.caculate(dataToCRC);

131 │ bean.crc16 = crc16;

132 │ return new QRPackBean { qrData = dataToCRC + crc16, qrBean = bean };

133 │

134 │ }

135 │ public static string packAddtional(QRAddtionalBean bean)

136 │ {

137 │ StringBuilder content = new StringBuilder(100);

138 │ if (!bean.billNumber.IsEmpty())

139 │ {

140 │ content.Append(QRTag.BILL\_NUMBER)

141 │ .Append(padLeft(bean.billNumber.Length))

142 │ .Append(bean.billNumber);

143 │ }

144 │ if (!bean.storeID.IsEmpty())

145 │ {

146 │ content.Append(QRTag.STORE\_ID)

147 │ .Append(padLeft(bean.storeID.Length))

148 │ .Append(bean.storeID);

149 │ }

150 │

151 │ if (!bean.referenceID.IsEmpty())

152 │ {

153 │ string referenceID;

154 │ string clearReferenceID = bean.referenceID.Substring(2);

155 │ string prefix = bean.referenceID.Substring(0, 2);

156 │ if (!bean.expDate.IsEmpty())

157 │ {

158 │ referenceID = prefix + bean.expDate + clearReferenceID;

159 │ }

160 │ else

161 │ {

162 │ referenceID = prefix + EXPIRE\_DATE\_EMPTY + clearReferenceID;

163 │ }

164 │ content.Append(QRTag.REFERENCE\_NUMBER)

165 │ .Append(padLeft(referenceID.Length))

166 │ .Append(referenceID);

167 │ }

168 │ if (!bean.customerID.IsEmpty())

169 │ {

170 │ content.Append(QRTag.CUSTOMER\_ID)

171 │ .Append(padLeft(bean.customerID.Length))

172 │ .Append(bean.customerID);

173 │ }

174 │

175 │ if (!bean.terminalID.IsEmpty())

176 │ {

177 │ content.Append(QRTag.TERMINAL\_ID)

178 │ .Append(padLeft(bean.terminalID.Length))

179 │ .Append(bean.terminalID);

180 │ }

181 │

182 │ if (!bean.purpose.IsEmpty())

183 │ {

184 │ content.Append(QRTag.PERPOSE)

185 │ .Append(padLeft(bean.purpose.Length))

186 │ .Append(bean.purpose);

187 │ }

188 │

189 │ string consumerData = "";

190 │ if (!bean.consumerAddress.IsEmpty())

191 │ {

192 │ consumerData = bean.consumerAddress;

193 │ }

194 │ if (!bean.consumerMobile.IsEmpty())

195 │ {

196 │ consumerData = consumerData + bean.consumerMobile;

197 │ }

198 │ if (!bean.consumerEmail.IsEmpty())

199 │ {

200 │ consumerData = consumerData + bean.consumerEmail;

201 │ }

202 │ if (!consumerData.IsEmpty())

203 │ {

204 │ content.Append(QRTag.CONSUMER\_DATA)

205 │ .Append(padLeft(consumerData.Length))

206 │ .Append(consumerData);

207 │ }

208 │ return content.ToString();

209 │ }

210 │

211 │ private static string packMerchant(QRBean bean)

212 │ {

213 │ StringBuilder content = new StringBuilder(100);

214 │ if (!bean.masterMerchant.IsEmpty())

215 │ {

216 │ content.Append(QRTag.MC\_ACCOUNT\_GUID)

217 │ .Append(padLeft(bean.masterMerchant.Length))

218 │ .Append(bean.masterMerchant);

219 │ }

220 │ if (!bean.merchantCode.IsEmpty())

221 │ {

222 │ content.Append(QRTag.MC\_ACCOUNT\_MC\_ID)

223 │ .Append(padLeft(bean.merchantCode.Length))

224 │ .Append(bean.merchantCode);

225 │ }

226 │ return content.ToString();

227 │ }

228 │

229 │ private static string padLeft(int lenght)

230 │ {

231 │ if (lenght > 99) {

232 │ throw new Exception("Lenght field can not be greater than 99");

233 │ }

234 │ return LeftPad(lenght + "", 2, "0");

235 │ }

236 │

237 │

238 │ public static string LeftPad(string input, int length, string padChar)

239 │ {

240 │ if (input.Length >= length)

241 │ {

242 │ return input;

243 │ }

244 │

245 │ int padLength = length - input.Length;

246 │ StringBuilder paddedString = new StringBuilder(input);

247 │ paddedString.Insert(0, padChar, padLength);

248 │

249 │ return paddedString.ToString();

250 │ }

251 │ private static bool IsEmpty(this string str)

252 │ {

253 │

254 │ try

255 │ {

256 │ return str == null || str.Length == 0;

257 │ }

258 │ catch

259 │ {

260 │ return false;

261 │ }

262 │ }

263 │ }

264 │ }

────────────────────────────────────────────────────────────

**📄 QRPackBean.cs (CS, 8 lines)**

1 │ ﻿namespace QuanLyBanHang.Common.BIDV

2 │ {

3 │ public class QRPackBean

4 │ {

5 │ public QRBean qrBean { get; set; }

6 │ public string qrData { get; set; }

7 │ }

8 │ }

────────────────────────────────────────────────────────────

**📄 QRTag.cs (CS, 84 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │ using System.Linq;

4 │ using System.Text;

5 │ using System.Threading.Tasks;

6 │

7 │ namespace QuanLyBanHang.Common.BIDV

8 │ {

9 │ public static class QRTag

10 │ {

11 │ public static string PAY\_LOAD = "00";

12 │

13 │ public static string POINT\_OI\_METHOD = "01";

14 │

15 │ public static string MC\_ACCOUNT = "26";

16 │

17 │ public static string MC\_ACCOUNT\_GUID = "00";

18 │

19 │ public static string MC\_ACCOUNT\_MC\_ID = "01";

20 │

21 │ public static string MC\_CATEGORY\_CODE = "52";

22 │

23 │ public static string CCY = "53";

24 │

25 │ public static string AMOUNT = "54";

26 │

27 │ public static string TIP\_AND\_FEE = "55";

28 │

29 │ public static string FIXED\_FEE = "56";

30 │

31 │ public static string PERCENT\_FEE = "57";

32 │

33 │ public static string COUNTRY\_CODE = "58";

34 │

35 │ public static string MC\_NAME = "59";

36 │

37 │ public static string MC\_CITY = "60";

38 │

39 │ public static string MC\_PIN\_CODE = "61";

40 │

41 │ public static string ADDTIONAL\_DATA = "62";

42 │

43 │ public static string CRC16 = "63";

44 │

45 │ /\*\*

46 │ \* For addtional data of Qrcode

47 │ \*/

48 │ public static string BILL\_NUMBER = "01";

49 │

50 │ public static string STORE\_ID = "03"; // Terminal name

51 │

52 │ public static string LOYALTY\_NUMBER = "04";

53 │

54 │ public static string REFERENCE\_NUMBER = "05";

55 │

56 │ public static string CUSTOMER\_ID = "06";

57 │

58 │ public static string TERMINAL\_ID = "07";

59 │

60 │ public static string PERPOSE = "08";

61 │

62 │ public static string CONSUMER\_DATA = "09";

63 │

64 │ public static string ADDRESS = "A";

65 │

66 │ public static string MOBILE = "M";

67 │

68 │ public static string EMAIL = "E";

69 │

70 │ public static string EXPIRE\_DATE = "51";

71 │

72 │ public static string CHECK\_SUM\_ADDTIONAL = "52";

73 │

74 │ public static string TERM = "80";

75 │

76 │ public static string CRC\_LENGHT = "04";

77 │

78 │ public static string VISA = "02";

79 │

80 │ public static string MASTERCARD = "04";

81 │

82 │ public static string UNIONPAY = "15";

83 │ }

84 │ }

────────────────────────────────────────────────────────────

**📄 QuanLyBanHangContext.cs (CS, 2095 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │ using Microsoft.EntityFrameworkCore;

4 │

5 │ namespace QuanLyBanHang.Models.QLBHEntity;

6 │

7 │ public partial class QuanLyBanHangContext : DbContext

8 │ {

9 │ public QuanLyBanHangContext()

10 │ {

11 │ }

12 │

13 │ public QuanLyBanHangContext(DbContextOptions<QuanLyBanHangContext> options)

14 │ : base(options)

15 │ {

16 │ }

17 │

18 │ public virtual DbSet<BidvQr> BidvQr { get; set; }

19 │

20 │ public virtual DbSet<InvLogVnpt> InvLogVnpt { get; set; }

21 │

22 │ public virtual DbSet<KhayNlBep> KhayNlBep { get; set; }

23 │

24 │ public virtual DbSet<KhayThChti> KhayThChti { get; set; }

25 │

26 │ public virtual DbSet<KhayVtBep> KhayVtBep { get; set; }

27 │

28 │ public virtual DbSet<KhayVtCafe> KhayVtCafe { get; set; }

29 │

30 │ public virtual DbSet<KhoBep> KhoBep { get; set; }

31 │

32 │ public virtual DbSet<KhoVtthBep> KhoVtthBep { get; set; }

33 │

34 │ public virtual DbSet<LogVietQr> LogVietQr { get; set; }

35 │

36 │ public virtual DbSet<OldTonKho> OldTonKho { get; set; }

37 │

38 │ public virtual DbSet<OldTonKhoBanHang> OldTonKhoBanHang { get; set; }

39 │

40 │ public virtual DbSet<OpsChiTietHoaDon> OpsChiTietHoaDon { get; set; }

41 │

42 │ public virtual DbSet<OpsHoaDon> OpsHoaDon { get; set; }

43 │

44 │ public virtual DbSet<OpsOrder> OpsOrder { get; set; }

45 │

46 │ public virtual DbSet<OpsOrderDetail> OpsOrderDetail { get; set; }

47 │

48 │ public virtual DbSet<OpsOrderItems> OpsOrderItems { get; set; }

49 │

50 │ public virtual DbSet<SerChiTietDinhMuc> SerChiTietDinhMuc { get; set; }

51 │

52 │ public virtual DbSet<StockChiTietKiemKe> StockChiTietKiemKe { get; set; }

53 │

54 │ public virtual DbSet<StockChiTietNhapHang> StockChiTietNhapHang { get; set; }

55 │

56 │ public virtual DbSet<StockChiTietTonKho> StockChiTietTonKho { get; set; }

57 │

58 │ public virtual DbSet<StockChiTietXuatKho> StockChiTietXuatKho { get; set; }

59 │

60 │ public virtual DbSet<StockDanhMucKho> StockDanhMucKho { get; set; }

61 │

62 │ public virtual DbSet<StockKiemKe> StockKiemKe { get; set; }

63 │

64 │ public virtual DbSet<StockLoaiKho> StockLoaiKho { get; set; }

65 │

66 │ public virtual DbSet<StockLoaiNhapXuat> StockLoaiNhapXuat { get; set; }

67 │

68 │ public virtual DbSet<StockNhapHang> StockNhapHang { get; set; }

69 │

70 │ public virtual DbSet<StockNhapKho> StockNhapKho { get; set; }

71 │

72 │ public virtual DbSet<StockTonKho> StockTonKho { get; set; }

73 │

74 │ public virtual DbSet<StockTonKhoBanHang> StockTonKhoBanHang { get; set; }

75 │

76 │ public virtual DbSet<StockXuatKho> StockXuatKho { get; set; }

77 │

78 │ public virtual DbSet<SysBranch> SysBranch { get; set; }

79 │

80 │ public virtual DbSet<SysLogo> SysLogo { get; set; }

81 │

82 │ public virtual DbSet<SysMenu> SysMenu { get; set; }

83 │

84 │ public virtual DbSet<SysOtp> SysOtp { get; set; }

85 │

86 │ public virtual DbSet<SysPermission> SysPermission { get; set; }

87 │

88 │ public virtual DbSet<SysRole> SysRole { get; set; }

89 │

90 │ public virtual DbSet<SysRoleMenu> SysRoleMenu { get; set; }

91 │

92 │ public virtual DbSet<SysRolePermission> SysRolePermission { get; set; }

93 │

94 │ public virtual DbSet<SysRoleReport> SysRoleReport { get; set; }

95 │

96 │ public virtual DbSet<SysStt> SysStt { get; set; }

97 │

98 │ public virtual DbSet<SysThongTinShop> SysThongTinShop { get; set; }

99 │

100 │ public virtual DbSet<SysToken> SysToken { get; set; }

101 │

102 │ public virtual DbSet<SysTrustedDevice> SysTrustedDevice { get; set; }

103 │

104 │ public virtual DbSet<SysUser> SysUser { get; set; }

105 │

106 │ public virtual DbSet<SysUserRole> SysUserRole { get; set; }

107 │

108 │ public virtual DbSet<SysVariable> SysVariable { get; set; }

109 │

110 │ public virtual DbSet<TblBuongPhong> TblBuongPhong { get; set; }

111 │

112 │ public virtual DbSet<TblChiTietKetCa> TblChiTietKetCa { get; set; }

113 │

114 │ public virtual DbSet<TblChiTietLichSuGia> TblChiTietLichSuGia { get; set; }

115 │

116 │ public virtual DbSet<TblChucVu> TblChucVu { get; set; }

117 │

118 │ public virtual DbSet<TblDonViTinh> TblDonViTinh { get; set; }

119 │

120 │ public virtual DbSet<TblGiaBanTheoKho> TblGiaBanTheoKho { get; set; }

121 │

122 │ public virtual DbSet<TblGiamGia> TblGiamGia { get; set; }

123 │

124 │ public virtual DbSet<TblGiuongBenh> TblGiuongBenh { get; set; }

125 │

126 │ public virtual DbSet<TblHangHoa> TblHangHoa { get; set; }

127 │

128 │ public virtual DbSet<TblHinhThucThanhToan> TblHinhThucThanhToan { get; set; }

129 │

130 │ public virtual DbSet<TblKyKho> TblKyKho { get; set; }

131 │

132 │ public virtual DbSet<TblLichSuChinhGia> TblLichSuChinhGia { get; set; }

133 │

134 │ public virtual DbSet<TblLichSuGia> TblLichSuGia { get; set; }

135 │

136 │ public virtual DbSet<TblLoaiTieuDung> TblLoaiTieuDung { get; set; }

137 │

138 │ public virtual DbSet<TblLopHangHoa> TblLopHangHoa { get; set; }

139 │

140 │ public virtual DbSet<TblNgayKetCa> TblNgayKetCa { get; set; }

141 │

142 │ public virtual DbSet<TblNhaCc> TblNhaCc { get; set; }

143 │

144 │ public virtual DbSet<TblNhomChucVu> TblNhomChucVu { get; set; }

145 │

146 │ public virtual DbSet<TblNhomThongBao> TblNhomThongBao { get; set; }

147 │

148 │ public virtual DbSet<TblPhongBan> TblPhongBan { get; set; }

149 │

150 │ public virtual DbSet<TblPhongBenh> TblPhongBenh { get; set; }

151 │

152 │ public virtual DbSet<TblThongBao> TblThongBao { get; set; }

153 │

154 │ public virtual DbSet<TblThueVat> TblThueVat { get; set; }

155 │

156 │ public virtual DbSet<TblTrangThaiNhapKho> TblTrangThaiNhapKho { get; set; }

157 │

158 │ public virtual DbSet<TkCf> TkCf { get; set; }

159 │

160 │ public virtual DbSet<TkChti> TkChti { get; set; }

161 │

162 │ public virtual DbSet<TkMbhHh> TkMbhHh { get; set; }

163 │

164 │ public virtual DbSet<TkVtthCafe> TkVtthCafe { get; set; }

165 │

166 │ public virtual DbSet<TkVtthChti> TkVtthChti { get; set; }

167 │

168 │ public virtual DbSet<VChiTietKiemKe> VChiTietKiemKe { get; set; }

169 │

170 │ public virtual DbSet<VChiTietNhapHang> VChiTietNhapHang { get; set; }

171 │

172 │ public virtual DbSet<VChiTietXuatKho> VChiTietXuatKho { get; set; }

173 │

174 │ public virtual DbSet<VGiaBanTheoKho> VGiaBanTheoKho { get; set; }

175 │

176 │ public virtual DbSet<VHangHoa> VHangHoa { get; set; }

177 │

178 │ public virtual DbSet<VNhapHang> VNhapHang { get; set; }

179 │

180 │ public virtual DbSet<VPermission> VPermission { get; set; }

181 │

182 │ public virtual DbSet<VTonKho> VTonKho { get; set; }

183 │

184 │ public virtual DbSet<VTonKhoBanHang> VTonKhoBanHang { get; set; }

185 │

186 │ public virtual DbSet<VietQr> VietQr { get; set; }

187 │

188 │ public virtual DbSet<VtthBaoCaoVatTuSuDung> VtthBaoCaoVatTuSuDung { get; set; }

189 │

190 │ public virtual DbSet<VtthChiTietDeXuat> VtthChiTietDeXuat { get; set; }

191 │

192 │ public virtual DbSet<VtthChiTietDinhMuc> VtthChiTietDinhMuc { get; set; }

193 │

194 │ public virtual DbSet<VtthDeXuatVtksk> VtthDeXuatVtksk { get; set; }

195 │

196 │ public virtual DbSet<VtthDinhMuc> VtthDinhMuc { get; set; }

197 │

198 │ public virtual DbSet<VtthKhay> VtthKhay { get; set; }

199 │

200 │ public virtual DbSet<VtthLichSuXuatKhay> VtthLichSuXuatKhay { get; set; }

201 │

202 │ public virtual DbSet<VtthNhapXuatKhay> VtthNhapXuatKhay { get; set; }

203 │

204 │ protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

205 │ => optionsBuilder.UseSqlServer("Name=ConnectionStrings:QLBH");

206 │

207 │ protected override void OnModelCreating(ModelBuilder modelBuilder)

208 │ {

209 │ modelBuilder.Entity<BidvQr>(entity =>

210 │ {

211 │ entity.HasKey(e => e.Ma);

212 │

213 │ entity.ToTable("bidv\_QR");

214 │

215 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

216 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

217 │ entity.Property(e => e.MaQr)

218 │ .HasMaxLength(50)

219 │ .IsUnicode(false)

220 │ .HasColumnName("MaQR");

221 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

222 │ entity.Property(e => e.NoiDung).HasMaxLength(500);

223 │ entity.Property(e => e.Qrdata)

224 │ .HasMaxLength(500)

225 │ .HasColumnName("QRData");

226 │ entity.Property(e => e.SoHoaDon).HasMaxLength(100);

227 │ });

228 │

229 │ modelBuilder.Entity<InvLogVnpt>(entity =>

230 │ {

231 │ entity.HasKey(e => e.Ma);

232 │

233 │ entity.ToTable("inv\_LogVNPT");

234 │

235 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

236 │ entity.Property(e => e.InvKey)

237 │ .HasMaxLength(50)

238 │ .IsUnicode(false)

239 │ .HasColumnName("invKey");

240 │ entity.Property(e => e.ReasonPhrase).HasMaxLength(200);

241 │ entity.Property(e => e.Result)

242 │ .HasMaxLength(200)

243 │ .IsUnicode(false);

244 │ entity.Property(e => e.SoapEnvelope).HasColumnName("soapEnvelope");

245 │ entity.Property(e => e.StatusCode)

246 │ .HasMaxLength(20)

247 │ .IsUnicode(false);

248 │ });

249 │

250 │ modelBuilder.Entity<KhayNlBep>(entity =>

251 │ {

252 │ entity

253 │ .HasNoKey()

254 │ .ToTable("KHAY\_NL\_BEP$");

255 │

256 │ entity.Property(e => e.Khây).HasColumnName("KHÂY");

257 │ entity.Property(e => e.MãHàngHoá)

258 │ .HasMaxLength(255)

259 │ .HasColumnName("Mã hàng hoá");

260 │ entity.Property(e => e.TênHàngHoá)

261 │ .HasMaxLength(255)

262 │ .HasColumnName("Tên hàng hoá");

263 │ entity.Property(e => e.Đvt)

264 │ .HasMaxLength(255)

265 │ .HasColumnName("ĐVT");

266 │ });

267 │

268 │ modelBuilder.Entity<KhayThChti>(entity =>

269 │ {

270 │ entity

271 │ .HasNoKey()

272 │ .ToTable("KHAY\_TH\_CHTI$");

273 │

274 │ entity.Property(e => e.Khây).HasColumnName("KHÂY");

275 │ entity.Property(e => e.MãHàngHoá)

276 │ .HasMaxLength(255)

277 │ .HasColumnName("Mã hàng hoá");

278 │ entity.Property(e => e.TênHàngHoá)

279 │ .HasMaxLength(255)

280 │ .HasColumnName("Tên hàng hoá");

281 │ entity.Property(e => e.Đvt)

282 │ .HasMaxLength(255)

283 │ .HasColumnName("ĐVT");

284 │ });

285 │

286 │ modelBuilder.Entity<KhayVtBep>(entity =>

287 │ {

288 │ entity

289 │ .HasNoKey()

290 │ .ToTable("KHAY\_VT\_BEP$");

291 │

292 │ entity.Property(e => e.Khây).HasColumnName("KHÂY");

293 │ entity.Property(e => e.MãHàngHoá)

294 │ .HasMaxLength(255)

295 │ .HasColumnName("Mã hàng hoá");

296 │ entity.Property(e => e.TênHàngHoá)

297 │ .HasMaxLength(255)

298 │ .HasColumnName("Tên hàng hoá");

299 │ entity.Property(e => e.Đvt)

300 │ .HasMaxLength(255)

301 │ .HasColumnName("ĐVT");

302 │ });

303 │

304 │ modelBuilder.Entity<KhayVtCafe>(entity =>

305 │ {

306 │ entity

307 │ .HasNoKey()

308 │ .ToTable("KHAY\_VT\_CAFE$");

309 │

310 │ entity.Property(e => e.Khây).HasColumnName("KHÂY");

311 │ entity.Property(e => e.MãHàngHoá)

312 │ .HasMaxLength(255)

313 │ .HasColumnName("Mã hàng hoá");

314 │ entity.Property(e => e.TênHàngHoá)

315 │ .HasMaxLength(255)

316 │ .HasColumnName("Tên hàng hoá");

317 │ entity.Property(e => e.Đvt)

318 │ .HasMaxLength(255)

319 │ .HasColumnName("ĐVT");

320 │ });

321 │

322 │ modelBuilder.Entity<KhoBep>(entity =>

323 │ {

324 │ entity

325 │ .HasNoKey()

326 │ .ToTable("Kho\_Bep$");

327 │

328 │ entity.Property(e => e.GiáVốnChưaVat).HasColumnName("Giá vốn chưa VAT");

329 │ entity.Property(e => e.HạnSửDụngDdMmYyyy)

330 │ .HasMaxLength(255)

331 │ .HasColumnName("Hạn sử dụng\_(dd/MM/yyyy)");

332 │ entity.Property(e => e.MãHàngHoá)

333 │ .HasMaxLength(255)

334 │ .HasColumnName("Mã hàng hoá");

335 │ entity.Property(e => e.MãVạch)

336 │ .HasMaxLength(255)

337 │ .HasColumnName("Mã vạch");

338 │ entity.Property(e => e.SlTồn).HasColumnName("SL Tồn");

339 │ entity.Property(e => e.Stt).HasColumnName("STT");

340 │ entity.Property(e => e.ThuếVat).HasColumnName("Thuế VAT (%)");

341 │ entity.Property(e => e.TênHàngHoá)

342 │ .HasMaxLength(255)

343 │ .HasColumnName("Tên hàng hoá");

344 │ entity.Property(e => e.Đvt)

345 │ .HasMaxLength(255)

346 │ .HasColumnName("ĐVT");

347 │ });

348 │

349 │ modelBuilder.Entity<KhoVtthBep>(entity =>

350 │ {

351 │ entity

352 │ .HasNoKey()

353 │ .ToTable("Kho\_VTTH\_Bep$");

354 │

355 │ entity.Property(e => e.GiáVốnChưaVat).HasColumnName("Giá vốn chưa VAT");

356 │ entity.Property(e => e.HạnSửDụngDdMmYyyy)

357 │ .HasMaxLength(255)

358 │ .HasColumnName("Hạn sử dụng\_(dd/MM/yyyy)");

359 │ entity.Property(e => e.MãHàngHoá)

360 │ .HasMaxLength(255)

361 │ .HasColumnName("Mã hàng hoá");

362 │ entity.Property(e => e.MãVạch)

363 │ .HasMaxLength(255)

364 │ .HasColumnName("Mã vạch");

365 │ entity.Property(e => e.SlTồn).HasColumnName("SL Tồn");

366 │ entity.Property(e => e.Stt).HasColumnName("STT");

367 │ entity.Property(e => e.ThuếVat).HasColumnName("Thuế VAT (%)");

368 │ entity.Property(e => e.TênHàngHoá)

369 │ .HasMaxLength(255)

370 │ .HasColumnName("Tên hàng hoá");

371 │ entity.Property(e => e.Đvt)

372 │ .HasMaxLength(255)

373 │ .HasColumnName("ĐVT");

374 │ });

375 │

376 │ modelBuilder.Entity<LogVietQr>(entity =>

377 │ {

378 │ entity.HasKey(e => e.Ma);

379 │

380 │ entity.ToTable("log\_VietQR");

381 │

382 │ entity.Property(e => e.Ma).HasDefaultValueSql("(newid())");

383 │ entity.Property(e => e.CreateDate).HasColumnType("datetime");

384 │ entity.Property(e => e.Description).HasMaxLength(100);

385 │ entity.Property(e => e.RequestId).HasMaxLength(50);

386 │ });

387 │

388 │ modelBuilder.Entity<OldTonKho>(entity =>

389 │ {

390 │ entity.HasKey(e => e.Ma).HasName("PK\_\_old\_TonK\_\_3214CC9F1E8E6CA5");

391 │

392 │ entity.ToTable("old\_TonKho");

393 │

394 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

395 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

396 │ entity.Property(e => e.GiaVon).HasColumnType("decimal(18, 2)");

397 │ entity.Property(e => e.HanSd).HasColumnType("date");

398 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

399 │ entity.Property(e => e.NamThang)

400 │ .HasMaxLength(6)

401 │ .IsUnicode(false);

402 │ entity.Property(e => e.SlbanTrongKy).HasColumnType("decimal(18, 4)");

403 │ entity.Property(e => e.SlnhapTrongKy).HasColumnType("decimal(18, 4)");

404 │ entity.Property(e => e.SltonCuoiKy).HasColumnType("decimal(18, 4)");

405 │ entity.Property(e => e.SltonDauKy).HasColumnType("decimal(18, 4)");

406 │ entity.Property(e => e.SlxuatKhac).HasColumnType("decimal(18, 4)");

407 │ entity.Property(e => e.SlxuatTieuDung).HasColumnType("decimal(18, 4)");

408 │ entity.Property(e => e.SlxuatTra).HasColumnType("decimal(18, 4)");

409 │ });

410 │

411 │ modelBuilder.Entity<OldTonKhoBanHang>(entity =>

412 │ {

413 │ entity.HasKey(e => e.Ma).HasName("PK\_\_old\_TonK\_\_3214CC9F02868386");

414 │

415 │ entity.ToTable("old\_TonKhoBanHang");

416 │

417 │ entity.Property(e => e.NamThang)

418 │ .HasMaxLength(6)

419 │ .IsUnicode(false);

420 │ entity.Property(e => e.SlbanTrongKy).HasColumnType("decimal(18, 4)");

421 │ entity.Property(e => e.SlnhapTrongKy).HasColumnType("decimal(18, 4)");

422 │ entity.Property(e => e.SltonCuoiKy).HasColumnType("decimal(18, 4)");

423 │ entity.Property(e => e.SltonDauKy).HasColumnType("decimal(18, 4)");

424 │ entity.Property(e => e.SlxuatKhac).HasColumnType("decimal(18, 4)");

425 │ entity.Property(e => e.SlxuatTieuDung).HasColumnType("decimal(18, 4)");

426 │ entity.Property(e => e.SlxuatTra).HasColumnType("decimal(18, 4)");

427 │ });

428 │

429 │ modelBuilder.Entity<OpsChiTietHoaDon>(entity =>

430 │ {

431 │ entity.HasKey(e => new { e.Ma, e.MaChiNhanhInt }).HasName("PK\_stock\_ChiTietHoaDon\_new");

432 │

433 │ entity.ToTable("ops\_ChiTietHoaDon");

434 │

435 │ entity.Property(e => e.Ma).ValueGeneratedOnAdd();

436 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

437 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

438 │ entity.Property(e => e.DonGia).HasColumnType("decimal(18, 2)");

439 │ entity.Property(e => e.GiaGiam).HasColumnType("decimal(18, 2)");

440 │ entity.Property(e => e.MaDvt).HasColumnName("MaDVT");

441 │ entity.Property(e => e.MaTaiKhoanDoanhThu)

442 │ .HasMaxLength(50)

443 │ .IsUnicode(false);

444 │ entity.Property(e => e.MaTaiKhoanThue)

445 │ .HasMaxLength(50)

446 │ .IsUnicode(false);

447 │ entity.Property(e => e.MaVach)

448 │ .HasMaxLength(20)

449 │ .IsUnicode(false);

450 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

451 │ entity.Property(e => e.NgayTra).HasColumnType("datetime");

452 │ entity.Property(e => e.SoLuong).HasColumnType("decimal(20, 4)");

453 │ entity.Property(e => e.SoLuongTraLai).HasColumnType("decimal(20, 4)");

454 │ entity.Property(e => e.ThanhTien).HasColumnType("decimal(18, 2)");

455 │ entity.Property(e => e.ThanhTienTraLai).HasColumnType("decimal(18, 2)");

456 │ entity.Property(e => e.ThueVat).HasColumnName("ThueVAT");

457 │ entity.Property(e => e.TienThue).HasColumnType("decimal(18, 2)");

458 │ });

459 │

460 │ modelBuilder.Entity<OpsHoaDon>(entity =>

461 │ {

462 │ entity.HasKey(e => new { e.Ma, e.MaChiNhanhInt }).HasName("PK\_ops\_HoaDon\_new");

463 │

464 │ entity.ToTable("ops\_HoaDon", tb => tb.HasTrigger("trg\_HoaDonBan\_del"));

465 │

466 │ entity.Property(e => e.Ma).ValueGeneratedOnAdd();

467 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

468 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

469 │ entity.Property(e => e.GhiChu).HasMaxLength(100);

470 │ entity.Property(e => e.GiamGia)

471 │ .HasDefaultValueSql("((0))")

472 │ .HasColumnType("decimal(18, 2)");

473 │ entity.Property(e => e.HinhThucTt).HasColumnName("HinhThucTT");

474 │ entity.Property(e => e.KhoaSo).HasDefaultValueSql("((0))");

475 │ entity.Property(e => e.KhuyenMaiTienMat).HasColumnType("decimal(18, 2)");

476 │ entity.Property(e => e.MaVachThe).HasMaxLength(50);

477 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

478 │ entity.Property(e => e.NgayHoaDon).HasColumnType("datetime");

479 │ entity.Property(e => e.NgayThanhToan).HasColumnType("datetime");

480 │ entity.Property(e => e.PhaiThu).HasColumnType("decimal(18, 2)");

481 │ entity.Property(e => e.RequestId)

482 │ .HasMaxLength(50)

483 │ .IsUnicode(false)

484 │ .HasColumnName("RequestID");

485 │ entity.Property(e => e.ReservationCode).IsUnicode(false);

486 │ entity.Property(e => e.SoHoaDonThue)

487 │ .HasMaxLength(50)

488 │ .IsUnicode(false);

489 │ entity.Property(e => e.SoSeri)

490 │ .HasMaxLength(50)

491 │ .IsUnicode(false)

492 │ .HasColumnName("So\_Seri");

493 │ entity.Property(e => e.SupplierTaxCode).IsUnicode(false);

494 │ entity.Property(e => e.ThanhTien).HasColumnType("decimal(18, 2)");

495 │ entity.Property(e => e.TienThua).HasDefaultValueSql("((0))");

496 │ entity.Property(e => e.TienTraLaiKhach).HasColumnType("decimal(18, 2)");

497 │ entity.Property(e => e.TienTraLaiThe).HasColumnType("decimal(18, 2)");

498 │ entity.Property(e => e.TongTienTraLai).HasColumnType("decimal(18, 2)");

499 │ entity.Property(e => e.TransactionId).IsUnicode(false);

500 │ entity.Property(e => e.WebOrderId)

501 │ .HasMaxLength(50)

502 │ .IsUnicode(false);

503 │ });

504 │

505 │ modelBuilder.Entity<OpsOrder>(entity =>

506 │ {

507 │ entity.HasKey(e => new { e.Ma, e.MaChiNhanhInt }).HasName("PK\_ops\_Order\_new");

508 │

509 │ entity.ToTable("ops\_Order");

510 │

511 │ entity.Property(e => e.Ma).ValueGeneratedOnAdd();

512 │ entity.Property(e => e.CodePhong).HasMaxLength(50);

513 │ entity.Property(e => e.MaNvthuNgan).HasColumnName("MaNVThuNgan");

514 │ entity.Property(e => e.SoDienThoai)

515 │ .HasMaxLength(20)

516 │ .IsUnicode(false);

517 │ entity.Property(e => e.TenKh)

518 │ .HasMaxLength(50)

519 │ .HasColumnName("TenKH");

520 │ entity.Property(e => e.Uuid)

521 │ .HasMaxLength(50)

522 │ .HasColumnName("UUID");

523 │ });

524 │

525 │ modelBuilder.Entity<OpsOrderDetail>(entity =>

526 │ {

527 │ entity.HasKey(e => new { e.Ma, e.MaChiNhanhInt }).HasName("PK\_ops\_OrderDetail\_new");

528 │

529 │ entity.ToTable("ops\_OrderDetail");

530 │

531 │ entity.Property(e => e.Ma)

532 │ .HasMaxLength(50)

533 │ .IsUnicode(false);

534 │ entity.Property(e => e.GhiChu).HasMaxLength(50);

535 │ entity.Property(e => e.NvgiaoHang).HasColumnName("NVGiaoHang");

536 │ entity.Property(e => e.NvxacNhan).HasColumnName("NVXacNhan");

537 │ entity.Property(e => e.ThoiGianDatMon).HasColumnType("datetime");

538 │ entity.Property(e => e.ThoiGianXacNhan).HasColumnType("datetime");

539 │ });

540 │

541 │ modelBuilder.Entity<OpsOrderItems>(entity =>

542 │ {

543 │ entity.HasKey(e => new { e.Ma, e.MaDonHang, e.MaDonHangChiTiet }).HasName("PK\_ops\_OrderDanhSachMon");

544 │

545 │ entity.ToTable("ops\_OrderItems");

546 │

547 │ entity.Property(e => e.Ma)

548 │ .HasMaxLength(50)

549 │ .IsUnicode(false);

550 │ entity.Property(e => e.MaDonHangChiTiet)

551 │ .HasMaxLength(50)

552 │ .IsUnicode(false);

553 │ entity.Property(e => e.GhiChu).HasMaxLength(500);

554 │ entity.Property(e => e.GiaBan).HasColumnType("decimal(18, 2)");

555 │ entity.Property(e => e.ThanhTien).HasColumnType("decimal(18, 2)");

556 │ });

557 │

558 │ modelBuilder.Entity<SerChiTietDinhMuc>(entity =>

559 │ {

560 │ entity.HasKey(e => new { e.Ma, e.MaChiNhanhInt }).HasName("PK\_\_ser\_ChiT\_\_3214CC9FD1B04D86");

561 │

562 │ entity.ToTable("ser\_ChiTietDinhMuc");

563 │

564 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

565 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

566 │ entity.Property(e => e.DonGiaTb).HasColumnType("decimal(18, 2)");

567 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

568 │ entity.Property(e => e.SoLuong).HasColumnType("decimal(18, 4)");

569 │ entity.Property(e => e.ThanhTien).HasColumnType("decimal(18, 2)");

570 │ });

571 │

572 │ modelBuilder.Entity<StockChiTietKiemKe>(entity =>

573 │ {

574 │ entity.HasKey(e => new { e.Ma, e.MaChiNhanhInt }).HasName("PK\_stock\_ChiTietKiemKe\_new");

575 │

576 │ entity.ToTable("stock\_ChiTietKiemKe");

577 │

578 │ entity.Property(e => e.Ma).ValueGeneratedOnAdd();

579 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

580 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

581 │ entity.Property(e => e.GhiChu).HasMaxLength(200);

582 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

583 │ entity.Property(e => e.SoLuong).HasColumnType("decimal(20, 4)");

584 │ });

585 │

586 │ modelBuilder.Entity<StockChiTietNhapHang>(entity =>

587 │ {

588 │ entity.HasKey(e => new { e.Ma, e.MaChiNhanhInt }).HasName("PK\_stock\_ChiTietNhapHang\_new");

589 │

590 │ entity.ToTable("stock\_ChiTietNhapHang");

591 │

592 │ entity.Property(e => e.Ma).ValueGeneratedOnAdd();

593 │ entity.Property(e => e.ChietKhauPt)

594 │ .HasColumnType("decimal(20, 4)")

595 │ .HasColumnName("ChietKhauPT");

596 │ entity.Property(e => e.ChietKhauTm)

597 │ .HasColumnType("decimal(20, 4)")

598 │ .HasColumnName("ChietKhauTM");

599 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

600 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

601 │ entity.Property(e => e.GiaVon).HasColumnType("decimal(20, 4)");

602 │ entity.Property(e => e.HanSd)

603 │ .HasColumnType("datetime")

604 │ .HasColumnName("HanSD");

605 │ entity.Property(e => e.MaDvt).HasColumnName("MaDVT");

606 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

607 │ entity.Property(e => e.SoLuong).HasColumnType("decimal(20, 4)");

608 │ entity.Property(e => e.ThanhTien).HasColumnType("decimal(20, 4)");

609 │ entity.Property(e => e.ThanhTienSauCk)

610 │ .HasColumnType("decimal(20, 4)")

611 │ .HasColumnName("ThanhTienSauCK");

612 │ entity.Property(e => e.ThanhTienSauVat)

613 │ .HasColumnType("decimal(20, 4)")

614 │ .HasColumnName("ThanhTienSauVAT");

615 │ entity.Property(e => e.Vat).HasColumnName("VAT");

616 │ });

617 │

618 │ modelBuilder.Entity<StockChiTietTonKho>(entity =>

619 │ {

620 │ entity.HasKey(e => e.Ma);

621 │

622 │ entity.ToTable("stock\_ChiTietTonKho");

623 │

624 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

625 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

626 │ entity.Property(e => e.GhiChu).HasMaxLength(2000);

627 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

628 │ entity.Property(e => e.SoLuongTon).HasColumnType("decimal(20, 4)");

629 │ });

630 │

631 │ modelBuilder.Entity<StockChiTietXuatKho>(entity =>

632 │ {

633 │ entity.HasKey(e => new { e.Ma, e.MaChiNhanhInt }).HasName("PK\_stock\_ChiTietXuatKho\_new");

634 │

635 │ entity.ToTable("stock\_ChiTietXuatKho");

636 │

637 │ entity.Property(e => e.Ma).ValueGeneratedOnAdd();

638 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

639 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

640 │ entity.Property(e => e.GhiChu).HasMaxLength(500);

641 │ entity.Property(e => e.GiaVon).HasColumnType("decimal(20, 4)");

642 │ entity.Property(e => e.MaCtnhapKho).HasColumnName("MaCTNhapKho");

643 │ entity.Property(e => e.MaDvt).HasColumnName("MaDVT");

644 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

645 │ entity.Property(e => e.SoLuong).HasColumnType("decimal(20, 4)");

646 │ entity.Property(e => e.SoLuongThucXuat).HasColumnType("decimal(20, 4)");

647 │ });

648 │

649 │ modelBuilder.Entity<StockDanhMucKho>(entity =>

650 │ {

651 │ entity.HasKey(e => e.Ma);

652 │

653 │ entity.ToTable("stock\_DanhMucKho");

654 │

655 │ entity.HasIndex(e => e.Ma, "IX\_StockDanhMucKho\_Ma");

656 │

657 │ entity.HasIndex(e => e.MaChiNhanh, "IX\_StockDanhMucKho\_MaChiNhanh");

658 │

659 │ entity.Property(e => e.Code)

660 │ .HasMaxLength(50)

661 │ .IsUnicode(false);

662 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

663 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

664 │ entity.Property(e => e.DienGiai).HasMaxLength(200);

665 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

666 │ entity.Property(e => e.Status).HasDefaultValueSql("((1))");

667 │ entity.Property(e => e.TaiKhoanCo)

668 │ .HasMaxLength(20)

669 │ .IsUnicode(false);

670 │ entity.Property(e => e.TaiKhoanKho)

671 │ .HasMaxLength(20)

672 │ .IsUnicode(false);

673 │ entity.Property(e => e.Ten).HasMaxLength(150);

674 │ entity.Property(e => e.TenHienThi).HasMaxLength(150);

675 │ });

676 │

677 │ modelBuilder.Entity<StockKiemKe>(entity =>

678 │ {

679 │ entity.HasKey(e => new { e.Ma, e.MaChiNhanhInt }).HasName("PK\_stock\_KiemKe\_new");

680 │

681 │ entity.ToTable("stock\_KiemKe", tb =>

682 │ {

683 │ tb.HasTrigger("tg\_kk\_BienBanKiemKe\_ins");

684 │ tb.HasTrigger("trg\_kk\_BienBanKiemKe\_del");

685 │ });

686 │

687 │ entity.Property(e => e.Ma).ValueGeneratedOnAdd();

688 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

689 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

690 │ entity.Property(e => e.DienGiai).HasMaxLength(500);

691 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

692 │ entity.Property(e => e.NgayKiemKe).HasColumnType("datetime");

693 │ entity.Property(e => e.NgayLap).HasColumnType("datetime");

694 │ entity.Property(e => e.NguoiKiemKe).HasMaxLength(50);

695 │ entity.Property(e => e.SoKiemKe).HasMaxLength(50);

696 │ });

697 │

698 │ modelBuilder.Entity<StockLoaiKho>(entity =>

699 │ {

700 │ entity.HasKey(e => e.Ma);

701 │

702 │ entity.ToTable("stock\_LoaiKho");

703 │

704 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

705 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

706 │ entity.Property(e => e.MoTa)

707 │ .HasMaxLength(100)

708 │ .IsUnicode(false);

709 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

710 │ entity.Property(e => e.Ten).HasMaxLength(50);

711 │ });

712 │

713 │ modelBuilder.Entity<StockLoaiNhapXuat>(entity =>

714 │ {

715 │ entity.HasKey(e => e.Ma);

716 │

717 │ entity.ToTable("stock\_LoaiNhapXuat");

718 │

719 │ entity.Property(e => e.Code).HasMaxLength(20);

720 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

721 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

722 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

723 │ entity.Property(e => e.NhapKho).HasComment("1: Nhập kho, 0:Xuất kho");

724 │ entity.Property(e => e.Ten).HasMaxLength(200);

725 │ });

726 │

727 │ modelBuilder.Entity<StockNhapHang>(entity =>

728 │ {

729 │ entity.HasKey(e => new { e.Ma, e.MaChiNhanhInt }).HasName("PK\_stock\_NhapHang\_new");

730 │

731 │ entity.ToTable("stock\_NhapHang", tb =>

732 │ {

733 │ tb.HasTrigger("trg\_NhapHang\_del");

734 │ tb.HasTrigger("trg\_NhapHang\_ins");

735 │ });

736 │

737 │ entity.Property(e => e.Ma).ValueGeneratedOnAdd();

738 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

739 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

740 │ entity.Property(e => e.GhiChu).HasMaxLength(500);

741 │ entity.Property(e => e.MaLoaiNx).HasColumnName("MaLoaiNX");

742 │ entity.Property(e => e.MaNhaCc).HasColumnName("MaNhaCC");

743 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

744 │ entity.Property(e => e.NgayHoaDon).HasColumnType("datetime");

745 │ entity.Property(e => e.NgayNhapHang).HasColumnType("datetime");

746 │ entity.Property(e => e.SoHoaDon).HasMaxLength(50);

747 │ entity.Property(e => e.SoHoaDonNcc)

748 │ .HasMaxLength(50)

749 │ .HasColumnName("SoHoaDonNCC");

750 │ entity.Property(e => e.ThueVat).HasColumnName("ThueVAT");

751 │ entity.Property(e => e.TienChietKhau).HasColumnType("decimal(20, 4)");

752 │ });

753 │

754 │ modelBuilder.Entity<StockNhapKho>(entity =>

755 │ {

756 │ entity.HasKey(e => e.Ma).HasName("PK\_\_stock\_Nh\_\_3214CC9F0F897FDD");

757 │

758 │ entity.ToTable("stock\_NhapKho");

759 │

760 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

761 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

762 │ entity.Property(e => e.KyHieuHoaDon).HasMaxLength(50);

763 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

764 │ entity.Property(e => e.NgayHetHanTt).HasColumnType("date");

765 │ entity.Property(e => e.NgayNhapKho).HasColumnType("datetime");

766 │ entity.Property(e => e.NgayXacNhan).HasColumnType("datetime");

767 │ entity.Property(e => e.PhaiTra).HasColumnType("decimal(18, 2)");

768 │ entity.Property(e => e.SoChungTu).HasMaxLength(50);

769 │ entity.Property(e => e.SoPhieu).HasMaxLength(50);

770 │ entity.Property(e => e.ThanhTien).HasColumnType("decimal(18, 2)");

771 │ entity.Property(e => e.TienChietKhau).HasColumnType("decimal(18, 2)");

772 │ entity.Property(e => e.TienThueVat).HasColumnType("decimal(18, 2)");

773 │ });

774 │

775 │ modelBuilder.Entity<StockTonKho>(entity =>

776 │ {

777 │ entity.HasKey(e => new { e.Ma, e.MaChiNhanhInt }).HasName("PK\_stock\_TonKho\_new");

778 │

779 │ entity.ToTable("stock\_TonKho");

780 │

781 │ entity.Property(e => e.Ma).ValueGeneratedOnAdd();

782 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

783 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

784 │ entity.Property(e => e.GiaVon).HasColumnType("decimal(18, 4)");

785 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

786 │ entity.Property(e => e.NamThang)

787 │ .HasMaxLength(6)

788 │ .IsUnicode(false);

789 │ entity.Property(e => e.SlbanTrongKy)

790 │ .HasDefaultValueSql("((0))")

791 │ .HasColumnType("decimal(20, 4)")

792 │ .HasColumnName("SLBanTrongKy");

793 │ entity.Property(e => e.SlnhapTrongKy)

794 │ .HasDefaultValueSql("((0))")

795 │ .HasColumnType("decimal(20, 4)")

796 │ .HasColumnName("SLNhapTrongKy");

797 │ entity.Property(e => e.SltonCuoiKy)

798 │ .HasDefaultValueSql("((0))")

799 │ .HasColumnType("decimal(20, 4)")

800 │ .HasColumnName("SLTonCuoiKy");

801 │ entity.Property(e => e.SltonDauKy)

802 │ .HasDefaultValueSql("((0))")

803 │ .HasColumnType("decimal(20, 4)")

804 │ .HasColumnName("SLTonDauKy");

805 │ entity.Property(e => e.SlxuatKhac)

806 │ .HasColumnType("decimal(20, 4)")

807 │ .HasColumnName("SLXuatKhac");

808 │ entity.Property(e => e.SlxuatTieuDung)

809 │ .HasColumnType("decimal(20, 4)")

810 │ .HasColumnName("SLXuatTieuDung");

811 │ entity.Property(e => e.SlxuatTra)

812 │ .HasColumnType("decimal(20, 4)")

813 │ .HasColumnName("SLXuatTra");

814 │ });

815 │

816 │ modelBuilder.Entity<StockTonKhoBanHang>(entity =>

817 │ {

818 │ entity.HasKey(e => new { e.Ma, e.MaChiNhanhInt });

819 │

820 │ entity.ToTable("stock\_TonKhoBanHang");

821 │

822 │ entity.Property(e => e.Ma).ValueGeneratedOnAdd();

823 │ entity.Property(e => e.NamThang)

824 │ .HasMaxLength(6)

825 │ .IsUnicode(false);

826 │ entity.Property(e => e.SlbanTrongKy)

827 │ .HasColumnType("decimal(20, 4)")

828 │ .HasColumnName("SLBanTrongKy");

829 │ entity.Property(e => e.SlnhapTrongKy)

830 │ .HasColumnType("decimal(20, 4)")

831 │ .HasColumnName("SLNhapTrongKy");

832 │ entity.Property(e => e.SltonCuoiKy)

833 │ .HasColumnType("decimal(20, 4)")

834 │ .HasColumnName("SLTonCuoiKy");

835 │ entity.Property(e => e.SltonDauKy)

836 │ .HasColumnType("decimal(20, 4)")

837 │ .HasColumnName("SLTonDauKy");

838 │ entity.Property(e => e.SlxuatKhac)

839 │ .HasColumnType("decimal(20, 4)")

840 │ .HasColumnName("SLXuatKhac");

841 │ entity.Property(e => e.SlxuatTieuDung)

842 │ .HasColumnType("decimal(20, 4)")

843 │ .HasColumnName("SLXuatTieuDung");

844 │ entity.Property(e => e.SlxuatTra)

845 │ .HasColumnType("decimal(20, 4)")

846 │ .HasColumnName("SLXuatTra");

847 │ });

848 │

849 │ modelBuilder.Entity<StockXuatKho>(entity =>

850 │ {

851 │ entity.HasKey(e => new { e.Ma, e.MaChiNhanhInt }).HasName("PK\_stock\_XuatKho\_new");

852 │

853 │ entity.ToTable("stock\_XuatKho", tb =>

854 │ {

855 │ tb.HasTrigger("tg\_tbl\_XuatKho\_ins");

856 │ tb.HasTrigger("trg\_tbl\_XuatKho\_del");

857 │ });

858 │

859 │ entity.Property(e => e.Ma).ValueGeneratedOnAdd();

860 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

861 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

862 │ entity.Property(e => e.GhiChu).HasMaxLength(500);

863 │ entity.Property(e => e.KhoaSo).HasDefaultValueSql("((0))");

864 │ entity.Property(e => e.MaLoaiNx).HasColumnName("MaLoaiNX");

865 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

866 │ entity.Property(e => e.NgayDuyet).HasColumnType("datetime");

867 │ entity.Property(e => e.NgayLap).HasColumnType("datetime");

868 │ entity.Property(e => e.NgayXuat).HasColumnType("datetime");

869 │ entity.Property(e => e.SoPhieu)

870 │ .HasMaxLength(30)

871 │ .IsUnicode(false);

872 │ });

873 │

874 │ modelBuilder.Entity<SysBranch>(entity =>

875 │ {

876 │ entity.HasKey(e => e.Ma);

877 │

878 │ entity.ToTable("sys\_Branch");

879 │

880 │ entity.Property(e => e.Ma).HasDefaultValueSql("(newid())");

881 │ entity.Property(e => e.Code).HasMaxLength(50);

882 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

883 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

884 │ entity.Property(e => e.DiaChi).HasMaxLength(500);

885 │ entity.Property(e => e.Logo).HasMaxLength(500);

886 │ entity.Property(e => e.MaSoThue).HasMaxLength(50);

887 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

888 │ entity.Property(e => e.NgayKhoaSo).HasColumnType("datetime");

889 │ entity.Property(e => e.Nid).HasColumnName("NId");

890 │ entity.Property(e => e.PhapNhan).HasMaxLength(600);

891 │ entity.Property(e => e.SoTaiKhoan)

892 │ .HasMaxLength(10)

893 │ .IsUnicode(false);

894 │ entity.Property(e => e.Ten).HasMaxLength(500);

895 │ entity.Property(e => e.TenVietTat).HasMaxLength(200);

896 │ });

897 │

898 │ modelBuilder.Entity<SysLogo>(entity =>

899 │ {

900 │ entity

901 │ .HasNoKey()

902 │ .ToTable("sys\_Logo");

903 │

904 │ entity.Property(e => e.BranchId).HasColumnName("branchID");

905 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

906 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

907 │ entity.Property(e => e.Index).HasColumnName("index");

908 │ entity.Property(e => e.LogoImage)

909 │ .HasMaxLength(50)

910 │ .HasColumnName("logoImage");

911 │ entity.Property(e => e.LogoName)

912 │ .HasMaxLength(50)

913 │ .HasColumnName("logoName");

914 │ entity.Property(e => e.Ma).HasDefaultValueSql("(newid())");

915 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

916 │ });

917 │

918 │ modelBuilder.Entity<SysMenu>(entity =>

919 │ {

920 │ entity.HasKey(e => e.Ma);

921 │

922 │ entity.ToTable("sys\_Menu");

923 │

924 │ entity.Property(e => e.Ma).HasDefaultValueSql("(newid())");

925 │ entity.Property(e => e.ClassIcon)

926 │ .HasMaxLength(50)

927 │ .IsUnicode(false);

928 │ entity.Property(e => e.Code).HasMaxLength(50);

929 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

930 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

931 │ entity.Property(e => e.Icon)

932 │ .HasMaxLength(50)

933 │ .IsUnicode(false);

934 │ entity.Property(e => e.Link).HasMaxLength(200);

935 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

936 │ entity.Property(e => e.Name).HasMaxLength(100);

937 │ entity.Property(e => e.ParentId).HasColumnName("ParentID");

938 │ entity.Property(e => e.UsingFor).HasMaxLength(50);

939 │ });

940 │

941 │ modelBuilder.Entity<SysOtp>(entity =>

942 │ {

943 │ entity.HasKey(e => e.Ma);

944 │

945 │ entity.ToTable("sys\_OTP");

946 │

947 │ entity.Property(e => e.Ma).HasDefaultValueSql("(newid())");

948 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

949 │ entity.Property(e => e.Otp)

950 │ .HasMaxLength(500)

951 │ .HasColumnName("OTP");

952 │ });

953 │

954 │ modelBuilder.Entity<SysPermission>(entity =>

955 │ {

956 │ entity.HasKey(e => e.Ma);

957 │

958 │ entity.ToTable("sys\_Permission");

959 │

960 │ entity.Property(e => e.Ma).HasDefaultValueSql("(newid())");

961 │ entity.Property(e => e.Code).HasMaxLength(50);

962 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

963 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

964 │ entity.Property(e => e.Description).HasMaxLength(300);

965 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

966 │ entity.Property(e => e.Name).HasMaxLength(250);

967 │ });

968 │

969 │ modelBuilder.Entity<SysRole>(entity =>

970 │ {

971 │ entity.HasKey(e => e.Ma);

972 │

973 │ entity.ToTable("sys\_Role");

974 │

975 │ entity.Property(e => e.Ma).HasDefaultValueSql("(newid())");

976 │ entity.Property(e => e.Code).HasMaxLength(50);

977 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

978 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

979 │ entity.Property(e => e.Description).HasMaxLength(300);

980 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

981 │ entity.Property(e => e.Name).HasMaxLength(250);

982 │ });

983 │

984 │ modelBuilder.Entity<SysRoleMenu>(entity =>

985 │ {

986 │ entity.HasKey(e => e.Ma);

987 │

988 │ entity.ToTable("sys\_RoleMenu");

989 │

990 │ entity.Property(e => e.Ma).HasDefaultValueSql("(newid())");

991 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

992 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

993 │ entity.Property(e => e.MenuId).HasColumnName("MenuID");

994 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

995 │ entity.Property(e => e.RoleId).HasColumnName("RoleID");

996 │ });

997 │

998 │ modelBuilder.Entity<SysRolePermission>(entity =>

999 │ {

1000 │ entity.HasKey(e => e.Ma);

1001 │

1002 │ entity.ToTable("sys\_RolePermission");

1003 │

1004 │ entity.Property(e => e.Ma).HasDefaultValueSql("(newid())");

1005 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

1006 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

1007 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

1008 │ entity.Property(e => e.PermissionId).HasColumnName("PermissionID");

1009 │ entity.Property(e => e.RoleMenuId).HasColumnName("RoleMenuID");

1010 │ });

1011 │

1012 │ modelBuilder.Entity<SysRoleReport>(entity =>

1013 │ {

1014 │ entity.HasKey(e => e.Ma);

1015 │

1016 │ entity.ToTable("sys\_RoleReport");

1017 │

1018 │ entity.Property(e => e.Ma).ValueGeneratedNever();

1019 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

1020 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

1021 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

1022 │ });

1023 │

1024 │ modelBuilder.Entity<SysStt>(entity =>

1025 │ {

1026 │ entity.HasKey(e => e.Ma).HasName("PK\_sys\_STT");

1027 │

1028 │ entity.ToTable("sys\_stt");

1029 │

1030 │ entity.Property(e => e.Ma).HasDefaultValueSql("(newid())");

1031 │ entity.Property(e => e.Colname).HasMaxLength(200);

1032 │ entity.Property(e => e.CreateDate).HasColumnType("date");

1033 │ entity.Property(e => e.TableMax).HasColumnType("decimal(18, 0)");

1034 │ entity.Property(e => e.TableName).HasMaxLength(50);

1035 │ });

1036 │

1037 │ modelBuilder.Entity<SysThongTinShop>(entity =>

1038 │ {

1039 │ entity.HasKey(e => e.Ma);

1040 │

1041 │ entity.ToTable("sys\_ThongTinShop");

1042 │

1043 │ entity.Property(e => e.DiaChi).HasMaxLength(150);

1044 │ entity.Property(e => e.DienThoai).HasMaxLength(15);

1045 │ entity.Property(e => e.Logo).HasColumnType("image");

1046 │ entity.Property(e => e.Ten).HasMaxLength(50);

1047 │ });

1048 │

1049 │ modelBuilder.Entity<SysToken>(entity =>

1050 │ {

1051 │ entity.HasKey(e => e.Ma);

1052 │

1053 │ entity.ToTable("sys\_Token");

1054 │

1055 │ entity.Property(e => e.Ma).HasDefaultValueSql("(newid())");

1056 │ entity.Property(e => e.ConnectionId)

1057 │ .HasMaxLength(50)

1058 │ .HasColumnName("ConnectionID");

1059 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

1060 │ entity.Property(e => e.DeviceName).HasMaxLength(150);

1061 │ entity.Property(e => e.DevicesId)

1062 │ .HasMaxLength(50)

1063 │ .HasColumnName("DevicesID");

1064 │ entity.Property(e => e.Ip)

1065 │ .HasMaxLength(20)

1066 │ .HasColumnName("IP");

1067 │ entity.Property(e => e.LastChangeIp).HasColumnType("datetime");

1068 │ entity.Property(e => e.LastSeen).HasColumnType("datetime");

1069 │ entity.Property(e => e.RefreshToken).HasMaxLength(500);

1070 │ entity.Property(e => e.SignToken).HasMaxLength(500);

1071 │ entity.Property(e => e.Token).HasMaxLength(500);

1072 │ entity.Property(e => e.UserId).HasColumnName("UserID");

1073 │ });

1074 │

1075 │ modelBuilder.Entity<SysTrustedDevice>(entity =>

1076 │ {

1077 │ entity.HasKey(e => e.Ma);

1078 │

1079 │ entity.ToTable("sys\_TrustedDevice");

1080 │

1081 │ entity.Property(e => e.Ma).HasDefaultValueSql("(newid())");

1082 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

1083 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

1084 │ entity.Property(e => e.DeviceName).HasMaxLength(500);

1085 │ entity.Property(e => e.DevicesIdsha256)

1086 │ .HasMaxLength(200)

1087 │ .HasColumnName("DevicesIDSHA256");

1088 │ entity.Property(e => e.UserId).HasColumnName("UserID");

1089 │ });

1090 │

1091 │ modelBuilder.Entity<SysUser>(entity =>

1092 │ {

1093 │ entity.HasKey(e => new { e.Ma, e.NId });

1094 │

1095 │ entity.ToTable("sys\_User");

1096 │

1097 │ entity.Property(e => e.Ma).HasDefaultValueSql("(newid())");

1098 │ entity.Property(e => e.NId)

1099 │ .ValueGeneratedOnAdd()

1100 │ .HasColumnName("nID");

1101 │ entity.Property(e => e.BranchId).HasColumnName("BranchID");

1102 │ entity.Property(e => e.Cccd)

1103 │ .HasMaxLength(50)

1104 │ .IsUnicode(false)

1105 │ .HasColumnName("CCCD");

1106 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

1107 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

1108 │ entity.Property(e => e.DiaChi).HasMaxLength(500);

1109 │ entity.Property(e => e.Fcm)

1110 │ .HasMaxLength(200)

1111 │ .HasColumnName("FCM");

1112 │ entity.Property(e => e.FullName).HasMaxLength(300);

1113 │ entity.Property(e => e.HinhAnh)

1114 │ .HasMaxLength(200)

1115 │ .IsUnicode(false);

1116 │ entity.Property(e => e.KhoHang)

1117 │ .HasMaxLength(200)

1118 │ .IsUnicode(false);

1119 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

1120 │ entity.Property(e => e.Msnv)

1121 │ .HasMaxLength(50)

1122 │ .IsUnicode(false)

1123 │ .HasColumnName("MSNV");

1124 │ entity.Property(e => e.NgayKetThuc).HasColumnType("datetime");

1125 │ entity.Property(e => e.NgaySinh).HasColumnType("datetime");

1126 │ entity.Property(e => e.NgayVaoLam).HasColumnType("datetime");

1127 │ entity.Property(e => e.Password).HasMaxLength(300);

1128 │ entity.Property(e => e.SoDienThoai)

1129 │ .HasMaxLength(20)

1130 │ .IsUnicode(false);

1131 │ entity.Property(e => e.Username).HasMaxLength(100);

1132 │ });

1133 │

1134 │ modelBuilder.Entity<SysUserRole>(entity =>

1135 │ {

1136 │ entity.HasKey(e => e.Ma);

1137 │

1138 │ entity.ToTable("sys\_UserRole");

1139 │

1140 │ entity.Property(e => e.Ma).HasDefaultValueSql("(newid())");

1141 │ entity.Property(e => e.BranchId).HasColumnName("BranchID");

1142 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

1143 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

1144 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

1145 │ entity.Property(e => e.RoleId).HasColumnName("RoleID");

1146 │ entity.Property(e => e.UserId).HasColumnName("UserID");

1147 │ });

1148 │

1149 │ modelBuilder.Entity<SysVariable>(entity =>

1150 │ {

1151 │ entity.HasKey(e => e.Ma);

1152 │

1153 │ entity.ToTable("sys\_Variable");

1154 │

1155 │ entity.Property(e => e.Code).HasMaxLength(50);

1156 │ entity.Property(e => e.Value).HasMaxLength(500);

1157 │ });

1158 │

1159 │ modelBuilder.Entity<TblBuongPhong>(entity =>

1160 │ {

1161 │ entity.HasKey(e => e.Ma);

1162 │

1163 │ entity.ToTable("tbl\_BuongPhong");

1164 │

1165 │ entity.Property(e => e.Code)

1166 │ .HasMaxLength(20)

1167 │ .IsUnicode(false);

1168 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

1169 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

1170 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

1171 │ entity.Property(e => e.Status).HasDefaultValueSql("((1))");

1172 │ entity.Property(e => e.Ten).HasMaxLength(200);

1173 │ });

1174 │

1175 │ modelBuilder.Entity<TblChiTietKetCa>(entity =>

1176 │ {

1177 │ entity.HasKey(e => e.Ma);

1178 │

1179 │ entity.ToTable("tbl\_ChiTietKetCa");

1180 │

1181 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

1182 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

1183 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

1184 │ entity.Property(e => e.NgayBan).HasColumnType("datetime");

1185 │ entity.Property(e => e.NgayKetCa).HasColumnType("datetime");

1186 │ entity.Property(e => e.SoTien).HasColumnType("decimal(18, 0)");

1187 │ });

1188 │

1189 │ modelBuilder.Entity<TblChiTietLichSuGia>(entity =>

1190 │ {

1191 │ entity.HasKey(e => e.Ma);

1192 │

1193 │ entity.ToTable("tbl\_ChiTietLichSuGia");

1194 │

1195 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

1196 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

1197 │ entity.Property(e => e.GhiChu).HasMaxLength(100);

1198 │ entity.Property(e => e.MaKho).HasMaxLength(500);

1199 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

1200 │ });

1201 │

1202 │ modelBuilder.Entity<TblChucVu>(entity =>

1203 │ {

1204 │ entity.HasKey(e => e.Ma);

1205 │

1206 │ entity.ToTable("tbl\_ChucVu");

1207 │

1208 │ entity.Property(e => e.Ma).HasDefaultValueSql("(newid())");

1209 │ entity.Property(e => e.Code)

1210 │ .HasMaxLength(50)

1211 │ .IsUnicode(false);

1212 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

1213 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

1214 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

1215 │ entity.Property(e => e.Ten).HasMaxLength(500);

1216 │ });

1217 │

1218 │ modelBuilder.Entity<TblDonViTinh>(entity =>

1219 │ {

1220 │ entity.HasKey(e => e.Ma);

1221 │

1222 │ entity.ToTable("tbl\_DonViTinh");

1223 │

1224 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

1225 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

1226 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

1227 │ entity.Property(e => e.Status).HasDefaultValueSql("((1))");

1228 │ entity.Property(e => e.Ten).HasMaxLength(100);

1229 │ });

1230 │

1231 │ modelBuilder.Entity<TblGiaBanTheoKho>(entity =>

1232 │ {

1233 │ entity.HasKey(e => new { e.Ma, e.MaChiNhanhInt }).HasName("PK\_tbl\_GiaBanTheoKho\_new");

1234 │

1235 │ entity.ToTable("tbl\_GiaBanTheoKho", tb => tb.HasTrigger("trg\_tbl\_GiaBanTheoKho\_Ins"));

1236 │

1237 │ entity.Property(e => e.Ma).ValueGeneratedOnAdd();

1238 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

1239 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

1240 │ entity.Property(e => e.GhiChu).HasMaxLength(100);

1241 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

1242 │ entity.Property(e => e.Status).HasDefaultValueSql("((0))");

1243 │ });

1244 │

1245 │ modelBuilder.Entity<TblGiamGia>(entity =>

1246 │ {

1247 │ entity.HasKey(e => e.Ma).HasName("PK\_tbl\_GIamGia");

1248 │

1249 │ entity.ToTable("tbl\_GiamGia");

1250 │

1251 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

1252 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

1253 │ entity.Property(e => e.DenNgay).HasColumnType("datetime");

1254 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

1255 │ entity.Property(e => e.Status).HasDefaultValueSql("((1))");

1256 │ entity.Property(e => e.Ten).HasMaxLength(100);

1257 │ entity.Property(e => e.TuNgay).HasColumnType("datetime");

1258 │ });

1259 │

1260 │ modelBuilder.Entity<TblGiuongBenh>(entity =>

1261 │ {

1262 │ entity.HasKey(e => e.Ma);

1263 │

1264 │ entity.ToTable("tbl\_GiuongBenh");

1265 │

1266 │ entity.Property(e => e.Code).HasMaxLength(20);

1267 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

1268 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

1269 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

1270 │ entity.Property(e => e.Status).HasDefaultValueSql("((1))");

1271 │ entity.Property(e => e.Ten).HasMaxLength(200);

1272 │ });

1273 │

1274 │ modelBuilder.Entity<TblHangHoa>(entity =>

1275 │ {

1276 │ entity.HasKey(e => new { e.Ma, e.MaChiNhanhInt });

1277 │

1278 │ entity.ToTable("tbl\_HangHoa", tb => tb.HasTrigger("trg\_HangHoa\_ins"));

1279 │

1280 │ entity.Property(e => e.Ma).ValueGeneratedOnAdd();

1281 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

1282 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

1283 │ entity.Property(e => e.GhiChu).HasMaxLength(100);

1284 │ entity.Property(e => e.GiaBan).HasColumnType("decimal(20, 4)");

1285 │ entity.Property(e => e.GiaVon).HasColumnType("decimal(20, 4)");

1286 │ entity.Property(e => e.HinhAnh).HasMaxLength(100);

1287 │ entity.Property(e => e.MaDvt).HasColumnName("MaDVT");

1288 │ entity.Property(e => e.MaKt)

1289 │ .HasMaxLength(20)

1290 │ .IsUnicode(false)

1291 │ .HasColumnName("MaKT");

1292 │ entity.Property(e => e.MaVach)

1293 │ .HasMaxLength(50)

1294 │ .IsUnicode(false);

1295 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

1296 │ entity.Property(e => e.ParentId).HasColumnName("ParentID");

1297 │ entity.Property(e => e.Ten).HasMaxLength(200);

1298 │ });

1299 │

1300 │ modelBuilder.Entity<TblHinhThucThanhToan>(entity =>

1301 │ {

1302 │ entity.HasKey(e => e.Ma);

1303 │

1304 │ entity.ToTable("tbl\_HinhThucThanhToan");

1305 │

1306 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

1307 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

1308 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

1309 │ entity.Property(e => e.Status).HasDefaultValueSql("((1))");

1310 │ entity.Property(e => e.Ten).HasMaxLength(50);

1311 │ });

1312 │

1313 │ modelBuilder.Entity<TblKyKho>(entity =>

1314 │ {

1315 │ entity.HasKey(e => e.Ma);

1316 │

1317 │ entity.ToTable("tbl\_KyKho");

1318 │

1319 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

1320 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

1321 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

1322 │ });

1323 │

1324 │ modelBuilder.Entity<TblLichSuChinhGia>(entity =>

1325 │ {

1326 │ entity.HasKey(e => e.Ma).HasName("PK\_his\_LichSuChinhGia");

1327 │

1328 │ entity.ToTable("tbl\_LichSuChinhGia");

1329 │

1330 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

1331 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

1332 │ entity.Property(e => e.GhiChu).HasMaxLength(200);

1333 │ entity.Property(e => e.GiaCu).HasColumnType("decimal(18, 4)");

1334 │ entity.Property(e => e.GiaMoi).HasColumnType("decimal(18, 4)");

1335 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

1336 │ });

1337 │

1338 │ modelBuilder.Entity<TblLichSuGia>(entity =>

1339 │ {

1340 │ entity.HasKey(e => e.Ma);

1341 │

1342 │ entity.ToTable("tbl\_LichSuGia");

1343 │

1344 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

1345 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

1346 │ entity.Property(e => e.LyDo).HasMaxLength(250);

1347 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

1348 │ entity.Property(e => e.TieuDe).HasMaxLength(250);

1349 │ });

1350 │

1351 │ modelBuilder.Entity<TblLoaiTieuDung>(entity =>

1352 │ {

1353 │ entity.HasKey(e => e.Ma);

1354 │

1355 │ entity.ToTable("tbl\_LoaiTieuDung");

1356 │

1357 │ entity.Property(e => e.Ma).ValueGeneratedNever();

1358 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

1359 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

1360 │ entity.Property(e => e.LoaiTieuDung).HasMaxLength(50);

1361 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

1362 │ entity.Property(e => e.Status)

1363 │ .HasMaxLength(10)

1364 │ .IsFixedLength();

1365 │ });

1366 │

1367 │ modelBuilder.Entity<TblLopHangHoa>(entity =>

1368 │ {

1369 │ entity.HasKey(e => e.Ma);

1370 │

1371 │ entity.ToTable("tbl\_LopHangHoa");

1372 │

1373 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

1374 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

1375 │ entity.Property(e => e.MaTaiKhoanDoanhThu)

1376 │ .HasMaxLength(50)

1377 │ .IsUnicode(false);

1378 │ entity.Property(e => e.MaTaiKhoanThue)

1379 │ .HasMaxLength(50)

1380 │ .IsUnicode(false);

1381 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

1382 │ entity.Property(e => e.PrinterName)

1383 │ .HasMaxLength(200)

1384 │ .IsUnicode(false);

1385 │ entity.Property(e => e.QuanLyQuaNfc).HasColumnName("QuanLyQuaNFC");

1386 │ entity.Property(e => e.Status).HasDefaultValueSql("((1))");

1387 │ entity.Property(e => e.Ten).HasMaxLength(200);

1388 │ entity.Property(e => e.ThueVat).HasColumnName("ThueVAT");

1389 │ });

1390 │

1391 │ modelBuilder.Entity<TblNgayKetCa>(entity =>

1392 │ {

1393 │ entity.HasKey(e => e.Ma);

1394 │

1395 │ entity.ToTable("tbl\_NgayKetCa");

1396 │

1397 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

1398 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

1399 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

1400 │ entity.Property(e => e.NgayBanHang).HasColumnType("datetime");

1401 │ });

1402 │

1403 │ modelBuilder.Entity<TblNhaCc>(entity =>

1404 │ {

1405 │ entity.HasKey(e => new { e.Ma, e.MaChiNhanhInt }).HasName("PK\_tbl\_NhaCC\_new");

1406 │

1407 │ entity.ToTable("tbl\_NhaCC");

1408 │

1409 │ entity.Property(e => e.Ma).ValueGeneratedOnAdd();

1410 │ entity.Property(e => e.Cccd)

1411 │ .HasMaxLength(30)

1412 │ .IsUnicode(false)

1413 │ .HasColumnName("CCCD");

1414 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

1415 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

1416 │ entity.Property(e => e.DiaChi).HasMaxLength(100);

1417 │ entity.Property(e => e.DienThoai).HasMaxLength(20);

1418 │ entity.Property(e => e.Email).HasMaxLength(50);

1419 │ entity.Property(e => e.MaKt)

1420 │ .HasMaxLength(20)

1421 │ .HasColumnName("MaKT");

1422 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

1423 │ entity.Property(e => e.Mst)

1424 │ .HasMaxLength(30)

1425 │ .HasColumnName("MST");

1426 │ entity.Property(e => e.NguoiLienHe).HasMaxLength(80);

1427 │ entity.Property(e => e.Status).HasDefaultValueSql("((1))");

1428 │ entity.Property(e => e.TenNhaCc)

1429 │ .HasMaxLength(100)

1430 │ .HasColumnName("TenNhaCC");

1431 │ });

1432 │

1433 │ modelBuilder.Entity<TblNhomChucVu>(entity =>

1434 │ {

1435 │ entity.HasKey(e => e.Ma).HasName("PK\_\_TblNhomC\_\_3214CC9F7D45A2F8");

1436 │

1437 │ entity.ToTable("tbl\_NhomChucVu");

1438 │

1439 │ entity.Property(e => e.Ma).ValueGeneratedNever();

1440 │ entity.Property(e => e.Code).HasMaxLength(255);

1441 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

1442 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

1443 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

1444 │ entity.Property(e => e.Ten).HasMaxLength(255);

1445 │ });

1446 │

1447 │ modelBuilder.Entity<TblNhomThongBao>(entity =>

1448 │ {

1449 │ entity.HasKey(e => e.Ma).HasName("PK\_\_TblNhomT\_\_3214CC9F6FAA1739");

1450 │

1451 │ entity.ToTable("tbl\_NhomThongBao");

1452 │

1453 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

1454 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

1455 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

1456 │ });

1457 │

1458 │ modelBuilder.Entity<TblPhongBan>(entity =>

1459 │ {

1460 │ entity.HasKey(e => e.Ma);

1461 │

1462 │ entity.ToTable("tbl\_PhongBan");

1463 │

1464 │ entity.Property(e => e.Ma).HasDefaultValueSql("(newid())");

1465 │ entity.Property(e => e.Code)

1466 │ .HasMaxLength(50)

1467 │ .IsUnicode(false);

1468 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

1469 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

1470 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

1471 │ entity.Property(e => e.Ten).HasMaxLength(500);

1472 │ });

1473 │

1474 │ modelBuilder.Entity<TblPhongBenh>(entity =>

1475 │ {

1476 │ entity.HasKey(e => e.Ma);

1477 │

1478 │ entity.ToTable("tbl\_PhongBenh");

1479 │

1480 │ entity.Property(e => e.Code).HasMaxLength(20);

1481 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

1482 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

1483 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

1484 │ entity.Property(e => e.Status).HasDefaultValueSql("((1))");

1485 │ entity.Property(e => e.Ten).HasMaxLength(100);

1486 │ });

1487 │

1488 │ modelBuilder.Entity<TblThongBao>(entity =>

1489 │ {

1490 │ entity.HasKey(e => e.Ma).HasName("PK\_\_TblThong\_\_3214CC9FBEE08E51");

1491 │

1492 │ entity.ToTable("tbl\_ThongBao");

1493 │

1494 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

1495 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

1496 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

1497 │ entity.Property(e => e.TenNhom).HasMaxLength(255);

1498 │ });

1499 │

1500 │ modelBuilder.Entity<TblThueVat>(entity =>

1501 │ {

1502 │ entity.HasKey(e => e.Ma);

1503 │

1504 │ entity.ToTable("tbl\_ThueVAT");

1505 │

1506 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

1507 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

1508 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

1509 │ entity.Property(e => e.Status).HasDefaultValueSql("((1))");

1510 │ entity.Property(e => e.Ten).HasMaxLength(120);

1511 │ });

1512 │

1513 │ modelBuilder.Entity<TblTrangThaiNhapKho>(entity =>

1514 │ {

1515 │ entity.HasKey(e => e.Ma).HasName("PK\_\_TblTrang\_\_3214CC9FACCF2DCF");

1516 │

1517 │ entity.ToTable("tbl\_TrangThaiNhapKho");

1518 │

1519 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

1520 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

1521 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

1522 │ entity.Property(e => e.Ten).HasMaxLength(255);

1523 │ });

1524 │

1525 │ modelBuilder.Entity<TkCf>(entity =>

1526 │ {

1527 │ entity

1528 │ .HasNoKey()

1529 │ .ToTable("TK\_CF$");

1530 │

1531 │ entity.Property(e => e.GiáVốnChưaVat).HasColumnName("Giá vốn chưa VAT");

1532 │ entity.Property(e => e.HạnSửDụngDdMmYyyy)

1533 │ .HasMaxLength(255)

1534 │ .HasColumnName("Hạn sử dụng\_(dd/MM/yyyy)");

1535 │ entity.Property(e => e.MãHàngHoá)

1536 │ .HasMaxLength(255)

1537 │ .HasColumnName("Mã hàng hoá");

1538 │ entity.Property(e => e.MãVạch)

1539 │ .HasMaxLength(255)

1540 │ .HasColumnName("Mã vạch");

1541 │ entity.Property(e => e.SlTồn).HasColumnName("SL Tồn");

1542 │ entity.Property(e => e.SltonTt).HasColumnName("SLTonTT");

1543 │ entity.Property(e => e.Stt).HasColumnName("STT");

1544 │ entity.Property(e => e.ThuếVat)

1545 │ .HasMaxLength(255)

1546 │ .HasColumnName("Thuế VAT (%)");

1547 │ entity.Property(e => e.TênHàngHoá)

1548 │ .HasMaxLength(255)

1549 │ .HasColumnName("Tên hàng hoá");

1550 │ entity.Property(e => e.Đvt)

1551 │ .HasMaxLength(255)

1552 │ .HasColumnName("ĐVT");

1553 │ });

1554 │

1555 │ modelBuilder.Entity<TkChti>(entity =>

1556 │ {

1557 │ entity

1558 │ .HasNoKey()

1559 │ .ToTable("TK\_CHTI$");

1560 │

1561 │ entity.Property(e => e.GiáVốnChưaVat).HasColumnName("Giá vốn chưa VAT");

1562 │ entity.Property(e => e.HạnSửDụngDdMmYyyy)

1563 │ .HasMaxLength(255)

1564 │ .HasColumnName("Hạn sử dụng\_(dd/MM/yyyy)");

1565 │ entity.Property(e => e.MãHàngHoá)

1566 │ .HasMaxLength(255)

1567 │ .HasColumnName("Mã hàng hoá");

1568 │ entity.Property(e => e.MãVạch)

1569 │ .HasMaxLength(255)

1570 │ .HasColumnName("Mã vạch");

1571 │ entity.Property(e => e.SlTồn).HasColumnName("SL Tồn");

1572 │ entity.Property(e => e.Stt).HasColumnName("STT");

1573 │ entity.Property(e => e.ThuếVat).HasColumnName("Thuế VAT (%)");

1574 │ entity.Property(e => e.TênHàngHoá)

1575 │ .HasMaxLength(255)

1576 │ .HasColumnName("Tên hàng hoá");

1577 │ entity.Property(e => e.Đvt)

1578 │ .HasMaxLength(255)

1579 │ .HasColumnName("ĐVT");

1580 │ });

1581 │

1582 │ modelBuilder.Entity<TkMbhHh>(entity =>

1583 │ {

1584 │ entity

1585 │ .HasNoKey()

1586 │ .ToTable("TK\_MBH\_HH$");

1587 │

1588 │ entity.Property(e => e.GiáVốnChưaVat).HasColumnName("Giá vốn chưa VAT");

1589 │ entity.Property(e => e.HạnSửDụngDdMmYyyy)

1590 │ .HasMaxLength(255)

1591 │ .HasColumnName("Hạn sử dụng\_(dd/MM/yyyy)");

1592 │ entity.Property(e => e.MãHàngHoá)

1593 │ .HasMaxLength(255)

1594 │ .HasColumnName("Mã hàng hoá");

1595 │ entity.Property(e => e.MãVạch)

1596 │ .HasMaxLength(255)

1597 │ .HasColumnName("Mã vạch");

1598 │ entity.Property(e => e.SlTồn).HasColumnName("SL Tồn");

1599 │ entity.Property(e => e.Stt).HasColumnName("STT");

1600 │ entity.Property(e => e.ThuếVat)

1601 │ .HasMaxLength(255)

1602 │ .HasColumnName("Thuế VAT (%)");

1603 │ entity.Property(e => e.TênHàngHoá)

1604 │ .HasMaxLength(255)

1605 │ .HasColumnName("Tên hàng hoá");

1606 │ entity.Property(e => e.Đvt)

1607 │ .HasMaxLength(255)

1608 │ .HasColumnName("ĐVT");

1609 │ });

1610 │

1611 │ modelBuilder.Entity<TkVtthCafe>(entity =>

1612 │ {

1613 │ entity

1614 │ .HasNoKey()

1615 │ .ToTable("TK\_VTTH\_Cafe$");

1616 │

1617 │ entity.Property(e => e.F9).HasMaxLength(255);

1618 │ entity.Property(e => e.GiáVốnChưaVat).HasColumnName("Giá vốn chưa VAT");

1619 │ entity.Property(e => e.HạnSửDụngDdMmYyyy)

1620 │ .HasMaxLength(255)

1621 │ .HasColumnName("Hạn sử dụng\_(dd/MM/yyyy)");

1622 │ entity.Property(e => e.MãHàngHoá)

1623 │ .HasMaxLength(255)

1624 │ .HasColumnName("Mã hàng hoá");

1625 │ entity.Property(e => e.MãVạch)

1626 │ .HasMaxLength(255)

1627 │ .HasColumnName("Mã vạch");

1628 │ entity.Property(e => e.SlTồn).HasColumnName("SL Tồn");

1629 │ entity.Property(e => e.ThuếVat)

1630 │ .HasMaxLength(255)

1631 │ .HasColumnName("Thuế VAT (%)");

1632 │ entity.Property(e => e.TênHàngHoá)

1633 │ .HasMaxLength(255)

1634 │ .HasColumnName("Tên hàng hoá");

1635 │ entity.Property(e => e.Đvt)

1636 │ .HasMaxLength(255)

1637 │ .HasColumnName("ĐVT");

1638 │ });

1639 │

1640 │ modelBuilder.Entity<TkVtthChti>(entity =>

1641 │ {

1642 │ entity

1643 │ .HasNoKey()

1644 │ .ToTable("TK\_VTTH\_CHTI$");

1645 │

1646 │ entity.Property(e => e.GiáVốnChưaVat).HasColumnName("Giá vốn chưa VAT");

1647 │ entity.Property(e => e.HạnSửDụngDdMmYyyy)

1648 │ .HasMaxLength(255)

1649 │ .HasColumnName("Hạn sử dụng\_(dd/MM/yyyy)");

1650 │ entity.Property(e => e.MãHàngHoá)

1651 │ .HasMaxLength(255)

1652 │ .HasColumnName("Mã hàng hoá");

1653 │ entity.Property(e => e.MãVạch)

1654 │ .HasMaxLength(255)

1655 │ .HasColumnName("Mã vạch");

1656 │ entity.Property(e => e.SlTồn).HasColumnName("SL Tồn");

1657 │ entity.Property(e => e.Stt).HasColumnName("STT");

1658 │ entity.Property(e => e.ThuếVat)

1659 │ .HasMaxLength(255)

1660 │ .HasColumnName("Thuế VAT (%)");

1661 │ entity.Property(e => e.TênHàngHoá)

1662 │ .HasMaxLength(255)

1663 │ .HasColumnName("Tên hàng hoá");

1664 │ entity.Property(e => e.Đvt)

1665 │ .HasMaxLength(255)

1666 │ .HasColumnName("ĐVT");

1667 │ });

1668 │

1669 │ modelBuilder.Entity<VChiTietKiemKe>(entity =>

1670 │ {

1671 │ entity

1672 │ .HasNoKey()

1673 │ .ToView("vChiTietKiemKe");

1674 │

1675 │ entity.Property(e => e.DienGiai).HasMaxLength(500);

1676 │ entity.Property(e => e.GhiChu).HasMaxLength(200);

1677 │ entity.Property(e => e.MaKt)

1678 │ .HasMaxLength(20)

1679 │ .IsUnicode(false)

1680 │ .HasColumnName("MaKT");

1681 │ entity.Property(e => e.MaVach)

1682 │ .HasMaxLength(20)

1683 │ .IsUnicode(false);

1684 │ entity.Property(e => e.NgayKiemKe).HasColumnType("datetime");

1685 │ entity.Property(e => e.NguoiKiemKe).HasMaxLength(50);

1686 │ entity.Property(e => e.SoKiemKe).HasMaxLength(50);

1687 │ entity.Property(e => e.SoLuongKiemKe).HasColumnType("decimal(20, 4)");

1688 │ entity.Property(e => e.TenDvt)

1689 │ .HasMaxLength(100)

1690 │ .HasColumnName("TenDVT");

1691 │ entity.Property(e => e.TenHang).HasMaxLength(200);

1692 │ entity.Property(e => e.TenKho).HasMaxLength(150);

1693 │ entity.Property(e => e.TenLop).HasMaxLength(200);

1694 │ });

1695 │

1696 │ modelBuilder.Entity<VChiTietNhapHang>(entity =>

1697 │ {

1698 │ entity

1699 │ .HasNoKey()

1700 │ .ToView("vChiTietNhapHang");

1701 │

1702 │ entity.Property(e => e.ChietKhauPt)

1703 │ .HasColumnType("decimal(20, 4)")

1704 │ .HasColumnName("ChietKhauPT");

1705 │ entity.Property(e => e.ChietKhauTm)

1706 │ .HasColumnType("decimal(31, 15)")

1707 │ .HasColumnName("ChietKhauTM");

1708 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

1709 │ entity.Property(e => e.GhiChu).HasMaxLength(500);

1710 │ entity.Property(e => e.GiaVon).HasColumnType("decimal(20, 4)");

1711 │ entity.Property(e => e.HanSd)

1712 │ .HasColumnType("datetime")

1713 │ .HasColumnName("HanSD");

1714 │ entity.Property(e => e.MaDvt).HasColumnName("MaDVT");

1715 │ entity.Property(e => e.MaKt)

1716 │ .HasMaxLength(20)

1717 │ .IsUnicode(false)

1718 │ .HasColumnName("MaKT");

1719 │ entity.Property(e => e.MaKtncc)

1720 │ .HasMaxLength(20)

1721 │ .HasColumnName("MaKTNCC");

1722 │ entity.Property(e => e.MaNhaCc).HasColumnName("MaNhaCC");

1723 │ entity.Property(e => e.MaVach)

1724 │ .HasMaxLength(20)

1725 │ .IsUnicode(false);

1726 │ entity.Property(e => e.NgayHoaDon).HasColumnType("datetime");

1727 │ entity.Property(e => e.NgayNhapHang).HasColumnType("datetime");

1728 │ entity.Property(e => e.SoHoaDon).HasMaxLength(50);

1729 │ entity.Property(e => e.SoHoaDonNcc)

1730 │ .HasMaxLength(50)

1731 │ .HasColumnName("SoHoaDonNCC");

1732 │ entity.Property(e => e.SoLuong).HasColumnType("decimal(20, 4)");

1733 │ entity.Property(e => e.TenDvt)

1734 │ .HasMaxLength(100)

1735 │ .HasColumnName("TenDVT");

1736 │ entity.Property(e => e.TenHangHoa).HasMaxLength(200);

1737 │ entity.Property(e => e.TenKho).HasMaxLength(150);

1738 │ entity.Property(e => e.TenLopHang).HasMaxLength(200);

1739 │ entity.Property(e => e.TenNhaCc)

1740 │ .HasMaxLength(100)

1741 │ .HasColumnName("TenNhaCC");

1742 │ entity.Property(e => e.ThanhTien).HasColumnType("decimal(20, 4)");

1743 │ entity.Property(e => e.ThanhTienSauCk)

1744 │ .HasColumnType("decimal(20, 4)")

1745 │ .HasColumnName("ThanhTienSauCK");

1746 │ entity.Property(e => e.ThanhTienSauVat)

1747 │ .HasColumnType("decimal(20, 4)")

1748 │ .HasColumnName("ThanhTienSauVAT");

1749 │ entity.Property(e => e.Vat).HasColumnName("VAT");

1750 │ });

1751 │

1752 │ modelBuilder.Entity<VChiTietXuatKho>(entity =>

1753 │ {

1754 │ entity

1755 │ .HasNoKey()

1756 │ .ToView("vChiTietXuatKho");

1757 │

1758 │ entity.Property(e => e.GhiChu).HasMaxLength(500);

1759 │ entity.Property(e => e.GhiChuChiTiet).HasMaxLength(500);

1760 │ entity.Property(e => e.GiaVon).HasColumnType("decimal(20, 4)");

1761 │ entity.Property(e => e.KhoChuyenDen).HasMaxLength(150);

1762 │ entity.Property(e => e.MaDvt).HasColumnName("MaDVT");

1763 │ entity.Property(e => e.MaKt)

1764 │ .HasMaxLength(20)

1765 │ .IsUnicode(false)

1766 │ .HasColumnName("MaKT");

1767 │ entity.Property(e => e.MaVach)

1768 │ .HasMaxLength(20)

1769 │ .IsUnicode(false);

1770 │ entity.Property(e => e.NgayXuat).HasColumnType("datetime");

1771 │ entity.Property(e => e.SoLuong).HasColumnType("decimal(20, 4)");

1772 │ entity.Property(e => e.SoPhieu)

1773 │ .HasMaxLength(30)

1774 │ .IsUnicode(false);

1775 │ entity.Property(e => e.TenDvt)

1776 │ .HasMaxLength(100)

1777 │ .HasColumnName("TenDVT");

1778 │ entity.Property(e => e.TenHang).HasMaxLength(200);

1779 │ entity.Property(e => e.TenKho).HasMaxLength(150);

1780 │ entity.Property(e => e.TenLop).HasMaxLength(200);

1781 │ });

1782 │

1783 │ modelBuilder.Entity<VGiaBanTheoKho>(entity =>

1784 │ {

1785 │ entity

1786 │ .HasNoKey()

1787 │ .ToView("vGiaBanTheoKho");

1788 │

1789 │ entity.Property(e => e.MaDvt).HasColumnName("MaDVT");

1790 │ entity.Property(e => e.MaVach)

1791 │ .HasMaxLength(20)

1792 │ .IsUnicode(false);

1793 │ entity.Property(e => e.SltonCuoiKy)

1794 │ .HasColumnType("decimal(20, 4)")

1795 │ .HasColumnName("SLTonCuoiKy");

1796 │ entity.Property(e => e.TenDvt)

1797 │ .HasMaxLength(100)

1798 │ .HasColumnName("TenDVT");

1799 │ entity.Property(e => e.TenHang).HasMaxLength(200);

1800 │ entity.Property(e => e.TenKho).HasMaxLength(150);

1801 │ entity.Property(e => e.TenLop).HasMaxLength(200);

1802 │ });

1803 │

1804 │ modelBuilder.Entity<VHangHoa>(entity =>

1805 │ {

1806 │ entity

1807 │ .HasNoKey()

1808 │ .ToView("vHangHoa");

1809 │

1810 │ entity.Property(e => e.GhiChu).HasMaxLength(100);

1811 │ entity.Property(e => e.GiaBan).HasColumnType("decimal(20, 4)");

1812 │ entity.Property(e => e.GiaVon).HasColumnType("decimal(20, 4)");

1813 │ entity.Property(e => e.HinhAnh).HasMaxLength(100);

1814 │ entity.Property(e => e.MaDvt).HasColumnName("MaDVT");

1815 │ entity.Property(e => e.MaKt)

1816 │ .HasMaxLength(20)

1817 │ .IsUnicode(false)

1818 │ .HasColumnName("MaKT");

1819 │ entity.Property(e => e.MaVach)

1820 │ .HasMaxLength(20)

1821 │ .IsUnicode(false);

1822 │ entity.Property(e => e.ParentId).HasColumnName("ParentID");

1823 │ entity.Property(e => e.TenDvt)

1824 │ .HasMaxLength(100)

1825 │ .HasColumnName("TenDVT");

1826 │ entity.Property(e => e.TenHangHoa).HasMaxLength(200);

1827 │ entity.Property(e => e.TenLopHang).HasMaxLength(200);

1828 │ });

1829 │

1830 │ modelBuilder.Entity<VNhapHang>(entity =>

1831 │ {

1832 │ entity

1833 │ .HasNoKey()

1834 │ .ToView("vNhapHang");

1835 │

1836 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

1837 │ entity.Property(e => e.GhiChu).HasMaxLength(500);

1838 │ entity.Property(e => e.MaNhaCc).HasColumnName("MaNhaCC");

1839 │ entity.Property(e => e.NgayHoaDon).HasColumnType("datetime");

1840 │ entity.Property(e => e.NgayNhapHang).HasColumnType("datetime");

1841 │ entity.Property(e => e.SoHoaDon).HasMaxLength(50);

1842 │ entity.Property(e => e.SoHoaDonNcc)

1843 │ .HasMaxLength(50)

1844 │ .HasColumnName("SoHoaDonNCC");

1845 │ entity.Property(e => e.TenKho).HasMaxLength(150);

1846 │ entity.Property(e => e.TenLoaiNhapXuat).HasMaxLength(200);

1847 │ entity.Property(e => e.TenNhaCc)

1848 │ .HasMaxLength(100)

1849 │ .HasColumnName("TenNhaCC");

1850 │ entity.Property(e => e.ThanhTien).HasColumnType("numeric(38, 6)");

1851 │ entity.Property(e => e.ThueVat).HasColumnName("ThueVAT");

1852 │ entity.Property(e => e.TienChietKhau).HasColumnType("decimal(20, 4)");

1853 │ });

1854 │

1855 │ modelBuilder.Entity<VPermission>(entity =>

1856 │ {

1857 │ entity

1858 │ .HasNoKey()

1859 │ .ToView("vPermission");

1860 │

1861 │ entity.Property(e => e.BranchId).HasColumnName("BranchID");

1862 │ entity.Property(e => e.FullName).HasMaxLength(300);

1863 │ entity.Property(e => e.MenuId).HasColumnName("MenuID");

1864 │ entity.Property(e => e.PermissionId).HasColumnName("PermissionID");

1865 │ entity.Property(e => e.RoleId).HasColumnName("RoleID");

1866 │ entity.Property(e => e.UserId).HasColumnName("UserID");

1867 │ entity.Property(e => e.Username).HasMaxLength(100);

1868 │ });

1869 │

1870 │ modelBuilder.Entity<VTonKho>(entity =>

1871 │ {

1872 │ entity

1873 │ .HasNoKey()

1874 │ .ToView("vTonKho");

1875 │

1876 │ entity.Property(e => e.GiaVon).HasColumnType("decimal(20, 4)");

1877 │ entity.Property(e => e.NamThang)

1878 │ .HasMaxLength(6)

1879 │ .IsUnicode(false);

1880 │ entity.Property(e => e.NgayNhapHang).HasColumnType("datetime");

1881 │ entity.Property(e => e.SlbanTrongKy)

1882 │ .HasColumnType("decimal(20, 4)")

1883 │ .HasColumnName("SLBanTrongKy");

1884 │ entity.Property(e => e.SlnhapTrongKy)

1885 │ .HasColumnType("decimal(20, 4)")

1886 │ .HasColumnName("SLNhapTrongKy");

1887 │ entity.Property(e => e.SltonCuoiKy)

1888 │ .HasColumnType("decimal(20, 4)")

1889 │ .HasColumnName("SLTonCuoiKy");

1890 │ entity.Property(e => e.SltonDauKy)

1891 │ .HasColumnType("decimal(20, 4)")

1892 │ .HasColumnName("SLTonDauKy");

1893 │ entity.Property(e => e.SlxuatKhac)

1894 │ .HasColumnType("decimal(20, 4)")

1895 │ .HasColumnName("SLXuatKhac");

1896 │ entity.Property(e => e.SlxuatTieuDung)

1897 │ .HasColumnType("decimal(20, 4)")

1898 │ .HasColumnName("SLXuatTieuDung");

1899 │ entity.Property(e => e.SlxuatTra)

1900 │ .HasColumnType("decimal(20, 4)")

1901 │ .HasColumnName("SLXuatTra");

1902 │ entity.Property(e => e.SoHoaDon).HasMaxLength(50);

1903 │ entity.Property(e => e.TenHangHoa).HasMaxLength(200);

1904 │ entity.Property(e => e.TriGiaBan).HasColumnType("decimal(38, 6)");

1905 │ entity.Property(e => e.TriGiaCuoiKy).HasColumnType("decimal(38, 6)");

1906 │ entity.Property(e => e.TriGiaDauKy).HasColumnType("decimal(38, 6)");

1907 │ entity.Property(e => e.TriGiaNhap).HasColumnType("decimal(38, 6)");

1908 │ });

1909 │

1910 │ modelBuilder.Entity<VTonKhoBanHang>(entity =>

1911 │ {

1912 │ entity

1913 │ .HasNoKey()

1914 │ .ToView("vTonKhoBanHang");

1915 │

1916 │ entity.Property(e => e.LoGanNhat).HasColumnType("datetime");

1917 │ entity.Property(e => e.LopHang).HasMaxLength(200);

1918 │ entity.Property(e => e.MaKt)

1919 │ .HasMaxLength(20)

1920 │ .IsUnicode(false)

1921 │ .HasColumnName("MaKT");

1922 │ entity.Property(e => e.MaVach)

1923 │ .HasMaxLength(20)

1924 │ .IsUnicode(false);

1925 │ entity.Property(e => e.NamThang)

1926 │ .HasMaxLength(6)

1927 │ .IsUnicode(false);

1928 │ entity.Property(e => e.SlbanTrongKy)

1929 │ .HasColumnType("decimal(20, 4)")

1930 │ .HasColumnName("SLBanTrongKy");

1931 │ entity.Property(e => e.SlnhapTrongKy)

1932 │ .HasColumnType("decimal(20, 4)")

1933 │ .HasColumnName("SLNhapTrongKy");

1934 │ entity.Property(e => e.SltonCuoiKy)

1935 │ .HasColumnType("decimal(20, 4)")

1936 │ .HasColumnName("SLTonCuoiKy");

1937 │ entity.Property(e => e.SltonDauKy)

1938 │ .HasColumnType("decimal(20, 4)")

1939 │ .HasColumnName("SLTonDauKy");

1940 │ entity.Property(e => e.SlxuatKhac)

1941 │ .HasColumnType("decimal(20, 4)")

1942 │ .HasColumnName("SLXuatKhac");

1943 │ entity.Property(e => e.SlxuatTieuDung)

1944 │ .HasColumnType("decimal(20, 4)")

1945 │ .HasColumnName("SLXuatTieuDung");

1946 │ entity.Property(e => e.SlxuatTra)

1947 │ .HasColumnType("decimal(18, 4)")

1948 │ .HasColumnName("SLXuatTra");

1949 │ entity.Property(e => e.SoLuongGanNhat).HasColumnType("decimal(20, 4)");

1950 │ entity.Property(e => e.TenDv)

1951 │ .HasMaxLength(100)

1952 │ .HasColumnName("TenDV");

1953 │ entity.Property(e => e.TenHangHoa).HasMaxLength(200);

1954 │ entity.Property(e => e.TenLop).HasMaxLength(200);

1955 │ });

1956 │

1957 │ modelBuilder.Entity<VietQr>(entity =>

1958 │ {

1959 │ entity.HasKey(e => e.Ma);

1960 │

1961 │ entity.ToTable("viet\_QR");

1962 │

1963 │ entity.Property(e => e.Ma).HasDefaultValueSql("(newid())");

1964 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

1965 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

1966 │ entity.Property(e => e.MienGiam).HasColumnType("decimal(18, 2)");

1967 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

1968 │ entity.Property(e => e.NgayThu).HasColumnType("datetime");

1969 │ entity.Property(e => e.NoiDung).HasMaxLength(500);

1970 │ entity.Property(e => e.Qrdata)

1971 │ .HasMaxLength(500)

1972 │ .HasColumnName("QRData");

1973 │ entity.Property(e => e.SoHoaDon).HasMaxLength(100);

1974 │ });

1975 │

1976 │ modelBuilder.Entity<VtthBaoCaoVatTuSuDung>(entity =>

1977 │ {

1978 │ entity.HasKey(e => e.Ma).HasName("PK\_\_vtth\_Bao\_\_3214CC9FE0724F0A");

1979 │

1980 │ entity.ToTable("vtth\_BaoCaoVatTuSuDung");

1981 │

1982 │ entity.Property(e => e.Ma).ValueGeneratedNever();

1983 │ entity.Property(e => e.SoLuongBan).HasColumnType("decimal(18, 4)");

1984 │ entity.Property(e => e.SoLuongNvl).HasColumnType("decimal(18, 4)");

1985 │ });

1986 │

1987 │ modelBuilder.Entity<VtthChiTietDeXuat>(entity =>

1988 │ {

1989 │ entity.HasKey(e => e.Ma).HasName("PK\_\_vtth\_Chi\_\_3214CC9F58FD5802");

1990 │

1991 │ entity.ToTable("vtth\_ChiTietDeXuat");

1992 │

1993 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

1994 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

1995 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

1996 │ entity.Property(e => e.NgayTra).HasColumnType("datetime");

1997 │ entity.Property(e => e.SoLuong).HasColumnType("decimal(18, 4)");

1998 │ entity.Property(e => e.SoLuongKsk).HasColumnType("decimal(18, 4)");

1999 │ entity.Property(e => e.SoLuongSd).HasColumnType("decimal(18, 4)");

2000 │ entity.Property(e => e.SoLuongTra).HasColumnType("decimal(18, 4)");

2001 │ entity.Property(e => e.SoLuongXuat).HasColumnType("decimal(18, 4)");

2002 │ });

2003 │

2004 │ modelBuilder.Entity<VtthChiTietDinhMuc>(entity =>

2005 │ {

2006 │ entity.HasKey(e => e.Ma).HasName("PK\_\_vtth\_Chi\_\_3214CC9FB09C889C");

2007 │

2008 │ entity.ToTable("vtth\_ChiTietDinhMuc");

2009 │

2010 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

2011 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

2012 │ entity.Property(e => e.DonGiaTb).HasColumnType("decimal(18, 2)");

2013 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

2014 │ entity.Property(e => e.SoLuong).HasColumnType("decimal(18, 4)");

2015 │ entity.Property(e => e.ThanhTien).HasColumnType("decimal(18, 2)");

2016 │ });

2017 │

2018 │ modelBuilder.Entity<VtthDeXuatVtksk>(entity =>

2019 │ {

2020 │ entity.HasKey(e => e.Ma).HasName("PK\_\_vtth\_DeX\_\_3214CC9FC8130B1E");

2021 │

2022 │ entity.ToTable("vtth\_DeXuatVtksk");

2023 │

2024 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

2025 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

2026 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

2027 │ entity.Property(e => e.NgayDeNghi).HasColumnType("datetime");

2028 │ entity.Property(e => e.NgayHoanTat).HasColumnType("datetime");

2029 │ entity.Property(e => e.NgayNhan).HasColumnType("datetime");

2030 │ entity.Property(e => e.NgayXuat).HasColumnType("datetime");

2031 │ entity.Property(e => e.SoPhieu).HasMaxLength(50);

2032 │ });

2033 │

2034 │ modelBuilder.Entity<VtthDinhMuc>(entity =>

2035 │ {

2036 │ entity.HasKey(e => e.Ma).HasName("PK\_\_vtth\_Din\_\_3214CC9FDA512F62");

2037 │

2038 │ entity.ToTable("vtth\_DinhMuc");

2039 │

2040 │ entity.Property(e => e.Code).HasMaxLength(255);

2041 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

2042 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

2043 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

2044 │ entity.Property(e => e.Ten).HasMaxLength(255);

2045 │ });

2046 │

2047 │ modelBuilder.Entity<VtthKhay>(entity =>

2048 │ {

2049 │ entity.HasKey(e => e.Ma).HasName("PK\_\_vtth\_Kha\_\_3214CC9FD0BF9543");

2050 │

2051 │ entity.ToTable("vtth\_Khay");

2052 │

2053 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

2054 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

2055 │ entity.Property(e => e.DinhMucKhay).HasColumnType("decimal(18, 4)");

2056 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

2057 │ });

2058 │

2059 │ modelBuilder.Entity<VtthLichSuXuatKhay>(entity =>

2060 │ {

2061 │ entity.HasKey(e => e.Ma).HasName("PK\_\_vtth\_Lic\_\_3214CC9FD93A03B6");

2062 │

2063 │ entity.ToTable("vtth\_LichSuXuatKhay");

2064 │

2065 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

2066 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

2067 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

2068 │ entity.Property(e => e.NgayXuat).HasColumnType("datetime");

2069 │ entity.Property(e => e.SoLuongXuat).HasColumnType("decimal(18, 4)");

2070 │ });

2071 │

2072 │ modelBuilder.Entity<VtthNhapXuatKhay>(entity =>

2073 │ {

2074 │ entity.HasKey(e => e.Ma).HasName("PK\_\_vtth\_Nha\_\_3214CC9F924B0AFF");

2075 │

2076 │ entity.ToTable("vtth\_NhapXuatKhay");

2077 │

2078 │ entity.Property(e => e.ConLai).HasColumnType("decimal(18, 4)");

2079 │ entity.Property(e => e.CreatedDate).HasColumnType("datetime");

2080 │ entity.Property(e => e.DeletedDate).HasColumnType("datetime");

2081 │ entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

2082 │ entity.Property(e => e.NgayNhap).HasColumnType("datetime");

2083 │ entity.Property(e => e.NgayTra).HasColumnType("datetime");

2084 │ entity.Property(e => e.NgayXacNhanTra).HasColumnType("datetime");

2085 │ entity.Property(e => e.NgayXuat).HasColumnType("datetime");

2086 │ entity.Property(e => e.SoLuongNhap).HasColumnType("decimal(18, 4)");

2087 │ entity.Property(e => e.SoLuongTra).HasColumnType("decimal(18, 4)");

2088 │ entity.Property(e => e.SoLuongXuat).HasColumnType("decimal(18, 4)");

2089 │ });

2090 │

2091 │ OnModelCreatingPartial(modelBuilder);

2092 │ }

2093 │

2094 │ partial void OnModelCreatingPartial(ModelBuilder modelBuilder);

2095 │ }

────────────────────────────────────────────────────────────

**📄 QuanLyVoucherContext.cs (CS, 490 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │ using Microsoft.EntityFrameworkCore;

4 │

5 │ namespace QuanLyBanHang.Models.VCEntity;

6 │

7 │ public partial class QuanLyVoucherContext : DbContext

8 │ {

9 │ public QuanLyVoucherContext()

10 │ {

11 │ }

12 │

13 │ public QuanLyVoucherContext(DbContextOptions<QuanLyVoucherContext> options)

14 │ : base(options)

15 │ {

16 │ }

17 │

18 │ public virtual DbSet<SysMenu> SysMenu { get; set; }

19 │

20 │ public virtual DbSet<SysQuyen> SysQuyen { get; set; }

21 │

22 │ public virtual DbSet<SysRole> SysRole { get; set; }

23 │

24 │ public virtual DbSet<SysRoleMenu> SysRoleMenu { get; set; }

25 │

26 │ public virtual DbSet<SysRoleQuyen> SysRoleQuyen { get; set; }

27 │

28 │ public virtual DbSet<SysUser> SysUser { get; set; }

29 │

30 │ public virtual DbSet<SysUserRole> SysUserRole { get; set; }

31 │

32 │ public virtual DbSet<TblBanHang> TblBanHang { get; set; }

33 │

34 │ public virtual DbSet<TblChiNhanh> TblChiNhanh { get; set; }

35 │

36 │ public virtual DbSet<TblKhachHang> TblKhachHang { get; set; }

37 │

38 │ public virtual DbSet<TblLichSuThanhVien> TblLichSuThanhVien { get; set; }

39 │

40 │ public virtual DbSet<TblNganh> TblNganh { get; set; }

41 │

42 │ public virtual DbSet<TblNhomVoucher> TblNhomVoucher { get; set; }

43 │

44 │ public virtual DbSet<TblThanhToan> TblThanhToan { get; set; }

45 │

46 │ public virtual DbSet<TblThanhVien> TblThanhVien { get; set; }

47 │

48 │ public virtual DbSet<TblVoucher> TblVoucher { get; set; }

49 │

50 │ public virtual DbSet<TblVoucherRole> TblVoucherRole { get; set; }

51 │

52 │ public virtual DbSet<VBanHang> VBanHang { get; set; }

53 │

54 │ public virtual DbSet<VDoanhThu> VDoanhThu { get; set; }

55 │

56 │ public virtual DbSet<VQuyen> VQuyen { get; set; }

57 │

58 │ public virtual DbSet<VTaoThanhToan> VTaoThanhToan { get; set; }

59 │

60 │ public virtual DbSet<VThanhToan> VThanhToan { get; set; }

61 │

62 │ public virtual DbSet<VVoucher> VVoucher { get; set; }

63 │

64 │ protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

65 │ => optionsBuilder.UseSqlServer("Name=ConnectionStrings:QLVoucher");

66 │

67 │ protected override void OnModelCreating(ModelBuilder modelBuilder)

68 │ {

69 │ modelBuilder.Entity<SysMenu>(entity =>

70 │ {

71 │ entity.HasKey(e => e.Ma);

72 │

73 │ entity.ToTable("sys\_Menu");

74 │

75 │ entity.Property(e => e.Code).HasMaxLength(100);

76 │ entity.Property(e => e.DuongDan).HasMaxLength(300);

77 │ entity.Property(e => e.Icon).HasMaxLength(200);

78 │ entity.Property(e => e.NgaySua).HasColumnType("datetime");

79 │ entity.Property(e => e.NgayTao)

80 │ .HasDefaultValueSql("(getdate())")

81 │ .HasColumnType("datetime");

82 │ entity.Property(e => e.NgayXoa).HasColumnType("datetime");

83 │ entity.Property(e => e.Ten).HasMaxLength(300);

84 │ entity.Property(e => e.ThamSo).HasMaxLength(300);

85 │ entity.Property(e => e.TrangThai).HasDefaultValueSql("((1))");

86 │ });

87 │

88 │ modelBuilder.Entity<SysQuyen>(entity =>

89 │ {

90 │ entity.HasKey(e => e.Ma);

91 │

92 │ entity.ToTable("sys\_Quyen");

93 │

94 │ entity.Property(e => e.Code).HasMaxLength(50);

95 │ entity.Property(e => e.MoTa).HasMaxLength(300);

96 │ entity.Property(e => e.NgaySua).HasColumnType("datetime");

97 │ entity.Property(e => e.NgayTao)

98 │ .HasDefaultValueSql("(getdate())")

99 │ .HasColumnType("datetime");

100 │ entity.Property(e => e.NgayXoa).HasColumnType("datetime");

101 │ entity.Property(e => e.Ten).HasMaxLength(250);

102 │ entity.Property(e => e.TrangThai).HasDefaultValueSql("((1))");

103 │ });

104 │

105 │ modelBuilder.Entity<SysRole>(entity =>

106 │ {

107 │ entity.HasKey(e => e.Ma);

108 │

109 │ entity.ToTable("sys\_Role");

110 │

111 │ entity.Property(e => e.Code).HasMaxLength(50);

112 │ entity.Property(e => e.MoTa).HasMaxLength(300);

113 │ entity.Property(e => e.NgaySua).HasColumnType("datetime");

114 │ entity.Property(e => e.NgayTao)

115 │ .HasDefaultValueSql("(getdate())")

116 │ .HasColumnType("datetime");

117 │ entity.Property(e => e.NgayXoa).HasColumnType("datetime");

118 │ entity.Property(e => e.Ten).HasMaxLength(250);

119 │ entity.Property(e => e.TrangThai).HasDefaultValueSql("((1))");

120 │ });

121 │

122 │ modelBuilder.Entity<SysRoleMenu>(entity =>

123 │ {

124 │ entity.HasKey(e => e.Ma);

125 │

126 │ entity.ToTable("sys\_RoleMenu");

127 │

128 │ entity.Property(e => e.ChoPhepTruyCap).HasDefaultValueSql("((1))");

129 │ entity.Property(e => e.NgaySua).HasColumnType("datetime");

130 │ entity.Property(e => e.NgayTao)

131 │ .HasDefaultValueSql("(getdate())")

132 │ .HasColumnType("datetime");

133 │ entity.Property(e => e.NgayXoa).HasColumnType("datetime");

134 │ entity.Property(e => e.TrangThai).HasDefaultValueSql("((1))");

135 │ });

136 │

137 │ modelBuilder.Entity<SysRoleQuyen>(entity =>

138 │ {

139 │ entity.HasKey(e => e.Ma);

140 │

141 │ entity.ToTable("sys\_RoleQuyen");

142 │

143 │ entity.Property(e => e.NgaySua).HasColumnType("datetime");

144 │ entity.Property(e => e.NgayTao)

145 │ .HasDefaultValueSql("(getdate())")

146 │ .HasColumnType("datetime");

147 │ entity.Property(e => e.NgayXoa).HasColumnType("datetime");

148 │ entity.Property(e => e.TrangThai).HasDefaultValueSql("((1))");

149 │ });

150 │

151 │ modelBuilder.Entity<SysUser>(entity =>

152 │ {

153 │ entity.HasKey(e => e.Ma);

154 │

155 │ entity.ToTable("sys\_User");

156 │

157 │ entity.Property(e => e.Crmuser).HasColumnName("CRMUser");

158 │ entity.Property(e => e.Email).HasMaxLength(50);

159 │ entity.Property(e => e.HoTen).HasMaxLength(100);

160 │ entity.Property(e => e.Msnv)

161 │ .HasMaxLength(30)

162 │ .IsUnicode(false)

163 │ .HasColumnName("MSNV");

164 │ entity.Property(e => e.NgaySua).HasColumnType("datetime");

165 │ entity.Property(e => e.NgayTao)

166 │ .HasDefaultValueSql("(getdate())")

167 │ .HasColumnType("datetime");

168 │ entity.Property(e => e.NgayXoa).HasColumnType("datetime");

169 │ entity.Property(e => e.Password).HasMaxLength(300);

170 │ entity.Property(e => e.Photo).HasColumnType("image");

171 │ entity.Property(e => e.TenDangNhap).HasMaxLength(100);

172 │ entity.Property(e => e.TrangThai).HasDefaultValueSql("((1))");

173 │ });

174 │

175 │ modelBuilder.Entity<SysUserRole>(entity =>

176 │ {

177 │ entity.HasKey(e => e.Ma);

178 │

179 │ entity.ToTable("sys\_UserRole");

180 │

181 │ entity.Property(e => e.NgaySua).HasColumnType("datetime");

182 │ entity.Property(e => e.NgayTao)

183 │ .HasDefaultValueSql("(getdate())")

184 │ .HasColumnType("datetime");

185 │ entity.Property(e => e.NgayXoa).HasColumnType("datetime");

186 │ entity.Property(e => e.TrangThai).HasDefaultValueSql("((1))");

187 │ });

188 │

189 │ modelBuilder.Entity<TblBanHang>(entity =>

190 │ {

191 │ entity.HasKey(e => e.Ma);

192 │

193 │ entity.ToTable("tbl\_BanHang");

194 │

195 │ entity.Property(e => e.GhiChu).HasMaxLength(500);

196 │ entity.Property(e => e.MaKh).HasColumnName("MaKH");

197 │ entity.Property(e => e.MaThe).HasMaxLength(20);

198 │ entity.Property(e => e.MaVach).HasMaxLength(50);

199 │ entity.Property(e => e.NgayHetHan).HasColumnType("datetime");

200 │ entity.Property(e => e.NgayHieuLuc).HasColumnType("datetime");

201 │ entity.Property(e => e.NgayKichHoat).HasColumnType("datetime");

202 │ entity.Property(e => e.NgaySua).HasColumnType("datetime");

203 │ entity.Property(e => e.NgayTao)

204 │ .HasDefaultValueSql("(getdate())")

205 │ .HasColumnType("datetime");

206 │ entity.Property(e => e.NgayXoa).HasColumnType("datetime");

207 │ entity.Property(e => e.TrangThai).HasDefaultValueSql("((1))");

208 │ });

209 │

210 │ modelBuilder.Entity<TblChiNhanh>(entity =>

211 │ {

212 │ entity.HasKey(e => e.Ma);

213 │

214 │ entity.ToTable("tbl\_ChiNhanh");

215 │

216 │ entity.Property(e => e.Logo).HasColumnType("image");

217 │ entity.Property(e => e.NgaySua).HasColumnType("datetime");

218 │ entity.Property(e => e.NgayTao)

219 │ .HasDefaultValueSql("(getdate())")

220 │ .HasColumnType("datetime");

221 │ entity.Property(e => e.NgayXoa).HasColumnType("datetime");

222 │ entity.Property(e => e.Ten).HasMaxLength(100);

223 │ entity.Property(e => e.TrangThai).HasDefaultValueSql("((1))");

224 │ });

225 │

226 │ modelBuilder.Entity<TblKhachHang>(entity =>

227 │ {

228 │ entity.HasKey(e => e.Ma);

229 │

230 │ entity.ToTable("tbl\_KhachHang");

231 │

232 │ entity.Property(e => e.Cmnd)

233 │ .HasMaxLength(50)

234 │ .HasColumnName("CMND");

235 │ entity.Property(e => e.DiaChi).HasMaxLength(200);

236 │ entity.Property(e => e.DienThoai).HasMaxLength(50);

237 │ entity.Property(e => e.Email).HasMaxLength(50);

238 │ entity.Property(e => e.Ghichu).HasMaxLength(500);

239 │ entity.Property(e => e.Hinh).HasColumnType("image");

240 │ entity.Property(e => e.NgaySua).HasColumnType("datetime");

241 │ entity.Property(e => e.NgayTao)

242 │ .HasDefaultValueSql("(getdate())")

243 │ .HasColumnType("datetime");

244 │ entity.Property(e => e.NgayXoa).HasColumnType("datetime");

245 │ entity.Property(e => e.Ten).HasMaxLength(200);

246 │ entity.Property(e => e.TrangThai).HasDefaultValueSql("((1))");

247 │ });

248 │

249 │ modelBuilder.Entity<TblLichSuThanhVien>(entity =>

250 │ {

251 │ entity.HasKey(e => e.Ma);

252 │

253 │ entity.ToTable("tbl\_LichSuThanhVien");

254 │

255 │ entity.Property(e => e.Ma).ValueGeneratedNever();

256 │ entity.Property(e => e.GiaTriCu).HasMaxLength(100);

257 │ entity.Property(e => e.GiaTriMoi).HasMaxLength(100);

258 │ entity.Property(e => e.NgayXuLy)

259 │ .HasDefaultValueSql("(getdate())")

260 │ .HasColumnType("datetime");

261 │ entity.Property(e => e.NoiDung).HasMaxLength(100);

262 │ });

263 │

264 │ modelBuilder.Entity<TblNganh>(entity =>

265 │ {

266 │ entity.HasKey(e => e.Ma);

267 │

268 │ entity.ToTable("tbl\_Nganh");

269 │

270 │ entity.Property(e => e.NgaySua).HasColumnType("datetime");

271 │ entity.Property(e => e.NgayTao)

272 │ .HasDefaultValueSql("(getdate())")

273 │ .HasColumnType("datetime");

274 │ entity.Property(e => e.NgayXoa).HasColumnType("datetime");

275 │ entity.Property(e => e.Ten).HasMaxLength(100);

276 │ entity.Property(e => e.TrangThai).HasDefaultValueSql("((1))");

277 │ });

278 │

279 │ modelBuilder.Entity<TblNhomVoucher>(entity =>

280 │ {

281 │ entity.HasKey(e => e.Ma);

282 │

283 │ entity.ToTable("tbl\_NhomVoucher");

284 │

285 │ entity.Property(e => e.NgaySua).HasColumnType("datetime");

286 │ entity.Property(e => e.NgayTao)

287 │ .HasDefaultValueSql("(getdate())")

288 │ .HasColumnType("datetime");

289 │ entity.Property(e => e.NgayXoa).HasColumnType("datetime");

290 │ entity.Property(e => e.Ten).HasMaxLength(100);

291 │ entity.Property(e => e.TrangThai).HasDefaultValueSql("((1))");

292 │ });

293 │

294 │ modelBuilder.Entity<TblThanhToan>(entity =>

295 │ {

296 │ entity.HasKey(e => e.Ma);

297 │

298 │ entity.ToTable("tbl\_ThanhToan");

299 │

300 │ entity.Property(e => e.Code).HasMaxLength(50);

301 │ entity.Property(e => e.GhiChu).HasMaxLength(500);

302 │ entity.Property(e => e.NgaySua).HasColumnType("datetime");

303 │ entity.Property(e => e.NgayTao)

304 │ .HasDefaultValueSql("(getdate())")

305 │ .HasColumnType("datetime");

306 │ entity.Property(e => e.NgayXoa).HasColumnType("datetime");

307 │ entity.Property(e => e.PhuThu).HasDefaultValueSql("((0))");

308 │ entity.Property(e => e.SoDu).HasDefaultValueSql("((0))");

309 │ });

310 │

311 │ modelBuilder.Entity<TblThanhVien>(entity =>

312 │ {

313 │ entity.HasKey(e => e.Ma);

314 │

315 │ entity.ToTable("tbl\_ThanhVien");

316 │

317 │ entity.Property(e => e.MaKh).HasColumnName("MaKH");

318 │ entity.Property(e => e.MaVach).HasMaxLength(20);

319 │ entity.Property(e => e.NgayBatDau).HasColumnType("datetime");

320 │ entity.Property(e => e.NgayCapNhat).HasColumnType("datetime");

321 │ entity.Property(e => e.NgayDangKy)

322 │ .HasDefaultValueSql("(getdate())")

323 │ .HasColumnType("datetime");

324 │ entity.Property(e => e.NgayKetThuc).HasColumnType("datetime");

325 │ entity.Property(e => e.NgayXoa).HasColumnType("datetime");

326 │ });

327 │

328 │ modelBuilder.Entity<TblVoucher>(entity =>

329 │ {

330 │ entity.HasKey(e => e.Ma);

331 │

332 │ entity.ToTable("tbl\_Voucher");

333 │

334 │ entity.Property(e => e.LoaiThoiHan).HasMaxLength(50);

335 │ entity.Property(e => e.LoaiVoucher).HasMaxLength(50);

336 │ entity.Property(e => e.NgaySua).HasColumnType("datetime");

337 │ entity.Property(e => e.NgayTao)

338 │ .HasDefaultValueSql("(getdate())")

339 │ .HasColumnType("datetime");

340 │ entity.Property(e => e.NgayXoa).HasColumnType("datetime");

341 │ entity.Property(e => e.NhomVoucherId).HasColumnName("NhomVoucherID");

342 │ entity.Property(e => e.Ten).HasMaxLength(250);

343 │ entity.Property(e => e.TrangThai).HasDefaultValueSql("((1))");

344 │ });

345 │

346 │ modelBuilder.Entity<TblVoucherRole>(entity =>

347 │ {

348 │ entity.HasKey(e => e.Ma);

349 │

350 │ entity.ToTable("tbl\_VoucherRole");

351 │

352 │ entity.Property(e => e.NgaySua).HasColumnType("datetime");

353 │ entity.Property(e => e.NgayTao)

354 │ .HasDefaultValueSql("(getdate())")

355 │ .HasColumnType("datetime");

356 │ entity.Property(e => e.NgayXoa).HasColumnType("datetime");

357 │ entity.Property(e => e.TrangThai).HasDefaultValueSql("((1))");

358 │ });

359 │

360 │ modelBuilder.Entity<VBanHang>(entity =>

361 │ {

362 │ entity

363 │ .HasNoKey()

364 │ .ToView("vBanHang");

365 │

366 │ entity.Property(e => e.GhiChu).HasMaxLength(500);

367 │ entity.Property(e => e.LoaiThoiHan).HasMaxLength(50);

368 │ entity.Property(e => e.MaKh).HasColumnName("MaKH");

369 │ entity.Property(e => e.MaThe).HasMaxLength(20);

370 │ entity.Property(e => e.MaVach).HasMaxLength(50);

371 │ entity.Property(e => e.NgayHetHan).HasColumnType("datetime");

372 │ entity.Property(e => e.NgayHieuLuc).HasColumnType("datetime");

373 │ entity.Property(e => e.NgayKichHoat).HasColumnType("datetime");

374 │ entity.Property(e => e.NgayTao).HasColumnType("datetime");

375 │ entity.Property(e => e.NhomVoucherId).HasColumnName("NhomVoucherID");

376 │ entity.Property(e => e.TenVoucher).HasMaxLength(250);

377 │ });

378 │

379 │ modelBuilder.Entity<VDoanhThu>(entity =>

380 │ {

381 │ entity

382 │ .HasNoKey()

383 │ .ToView("vDoanhThu");

384 │

385 │ entity.Property(e => e.ChiNhanhBan).HasMaxLength(100);

386 │ entity.Property(e => e.ChiNhanhSuDung).HasMaxLength(100);

387 │ entity.Property(e => e.Cmnd)

388 │ .HasMaxLength(50)

389 │ .HasColumnName("CMND");

390 │ entity.Property(e => e.DiaChi).HasMaxLength(200);

391 │ entity.Property(e => e.DienThoai).HasMaxLength(50);

392 │ entity.Property(e => e.GhiChuBanHang).HasMaxLength(500);

393 │ entity.Property(e => e.LoaiVoucher).HasMaxLength(50);

394 │ entity.Property(e => e.MaBh).HasColumnName("MaBH");

395 │ entity.Property(e => e.MaKh).HasColumnName("MaKH");

396 │ entity.Property(e => e.MaThe).HasMaxLength(20);

397 │ entity.Property(e => e.NganhBan).HasMaxLength(100);

398 │ entity.Property(e => e.NganhSuDung).HasMaxLength(100);

399 │ entity.Property(e => e.NgayHetHan).HasColumnType("datetime");

400 │ entity.Property(e => e.NgayHieuLuc).HasColumnType("datetime");

401 │ entity.Property(e => e.NgayThanhToan).HasColumnType("datetime");

402 │ entity.Property(e => e.NhanVienThanhToan).HasMaxLength(100);

403 │ entity.Property(e => e.TenKh)

404 │ .HasMaxLength(200)

405 │ .HasColumnName("TenKH");

406 │ entity.Property(e => e.TenVoucher).HasMaxLength(250);

407 │ });

408 │

409 │ modelBuilder.Entity<VQuyen>(entity =>

410 │ {

411 │ entity

412 │ .HasNoKey()

413 │ .ToView("vQuyen");

414 │

415 │ entity.Property(e => e.Email).HasMaxLength(50);

416 │ entity.Property(e => e.HoTen).HasMaxLength(100);

417 │ entity.Property(e => e.TenDangNhap).HasMaxLength(100);

418 │ });

419 │

420 │ modelBuilder.Entity<VTaoThanhToan>(entity =>

421 │ {

422 │ entity

423 │ .HasNoKey()

424 │ .ToView("vTaoThanhToan");

425 │

426 │ entity.Property(e => e.LoaiThanhToan)

427 │ .HasMaxLength(13)

428 │ .IsUnicode(false);

429 │ entity.Property(e => e.MaKh).HasColumnName("MaKH");

430 │ entity.Property(e => e.MaThe).HasMaxLength(20);

431 │ entity.Property(e => e.MaVach).HasMaxLength(50);

432 │ entity.Property(e => e.NgayHetHan).HasColumnType("datetime");

433 │ entity.Property(e => e.NgayHieuLuc).HasColumnType("datetime");

434 │ entity.Property(e => e.NgayKichHoat).HasColumnType("datetime");

435 │ entity.Property(e => e.NgayTao).HasColumnType("datetime");

436 │ entity.Property(e => e.TenKh)

437 │ .HasMaxLength(200)

438 │ .HasColumnName("TenKH");

439 │ entity.Property(e => e.TenVoucher).HasMaxLength(250);

440 │ });

441 │

442 │ modelBuilder.Entity<VThanhToan>(entity =>

443 │ {

444 │ entity

445 │ .HasNoKey()

446 │ .ToView("vThanhToan");

447 │

448 │ entity.Property(e => e.Code).HasMaxLength(50);

449 │ entity.Property(e => e.GhiChu).HasMaxLength(500);

450 │ entity.Property(e => e.MaKh).HasColumnName("MaKH");

451 │ entity.Property(e => e.MaThe).HasMaxLength(20);

452 │ entity.Property(e => e.MaVach).HasMaxLength(50);

453 │ entity.Property(e => e.NgayTao).HasColumnType("datetime");

454 │ entity.Property(e => e.TenKh)

455 │ .HasMaxLength(200)

456 │ .HasColumnName("TenKH");

457 │ entity.Property(e => e.TenVoucher).HasMaxLength(250);

458 │ });

459 │

460 │ modelBuilder.Entity<VVoucher>(entity =>

461 │ {

462 │ entity

463 │ .HasNoKey()

464 │ .ToView("vVoucher");

465 │

466 │ entity.Property(e => e.ChiNhanhBan).HasMaxLength(100);

467 │ entity.Property(e => e.Cmnd)

468 │ .HasMaxLength(50)

469 │ .HasColumnName("CMND");

470 │ entity.Property(e => e.DiaChi).HasMaxLength(200);

471 │ entity.Property(e => e.DienThoai).HasMaxLength(50);

472 │ entity.Property(e => e.Email).HasMaxLength(50);

473 │ entity.Property(e => e.MaKh).HasColumnName("MaKH");

474 │ entity.Property(e => e.MaThe).HasMaxLength(20);

475 │ entity.Property(e => e.NganhBan).HasMaxLength(100);

476 │ entity.Property(e => e.NgayHetHan).HasColumnType("datetime");

477 │ entity.Property(e => e.NgayHieuLuc).HasColumnType("datetime");

478 │ entity.Property(e => e.NgayKichHoat).HasColumnType("datetime");

479 │ entity.Property(e => e.NhanVienBanHang).HasMaxLength(100);

480 │ entity.Property(e => e.TenKh)

481 │ .HasMaxLength(200)

482 │ .HasColumnName("TenKH");

483 │ entity.Property(e => e.TenVoucher).HasMaxLength(250);

484 │ });

485 │

486 │ OnModelCreatingPartial(modelBuilder);

487 │ }

488 │

489 │ partial void OnModelCreatingPartial(ModelBuilder modelBuilder);

490 │ }

────────────────────────────────────────────────────────────

**📄 ResponseData.cs (CS, 12 lines)**

1 │ ﻿namespace QuanLyBanHang.Models.VietinBank

2 │ {

3 │ public class ResponseData

4 │ {

5 │ public interface IResponseData : ISignData

6 │ {

7 │ public string RequestId { get; }

8 │ public StatusOutputBaseApi Status { get; }

9 │ public string Signature { get; }

10 │ }

11 │ }

12 │ }

────────────────────────────────────────────────────────────

**📄 ResponseModel.cs (CS, 51 lines)**

1 │ ﻿using Newtonsoft.Json;

2 │ using QuanLyBanHang.Common;

3 │

4 │ namespace QuanLyBanHang.Models.APIModels

5 │ {

6 │ public class ResponseModel

7 │ {

8 │ public string? Status { get; set; }

9 │ public string? Messenge { get; set; }

10 │ public string? Data { get; set; }

11 │ public ResponseModel() { }

12 │ public ResponseModel(string status, string messenge, Object data)

13 │ {

14 │ Status = status;

15 │ Messenge = messenge;

16 │ Data = JsonConvert.SerializeObject(data);

17 │ }

18 │ public ResponseModel(string status, string messenge)

19 │ {

20 │ Status = status;

21 │ Messenge = messenge;

22 │ Data = null;

23 │ }

24 │ }

25 │ public class ResponseModelSuccess : ResponseModel

26 │ {

27 │ public ResponseModelSuccess()

28 │ {

29 │ }

30 │

31 │ public ResponseModelSuccess(string messenge, Object? data = null)

32 │ {

33 │ Status = QLBHConst.OK;

34 │ Messenge = messenge;

35 │ Data = JsonConvert.SerializeObject(data);

36 │ }

37 │ }

38 │ public class ResponseModelError : ResponseModel

39 │ {

40 │ public ResponseModelError()

41 │ {

42 │ }

43 │

44 │ public ResponseModelError(string messenge, Object? data = null)

45 │ {

46 │ Status = QLBHConst.Error;

47 │ Messenge = messenge;

48 │ Data = JsonConvert.SerializeObject(data);

49 │ }

50 │ }

51 │ }

────────────────────────────────────────────────────────────

**📄 RestoreController.cs (CS, 561 lines)**

1 │ ﻿using EFCore.BulkExtensions;

2 │ using Microsoft.AspNetCore.Mvc;

3 │ using Microsoft.EntityFrameworkCore;

4 │ using QuanLyBanHang.Common;

5 │ using QuanLyBanHang.Models.APIModels;

6 │ using QuanLyBanHang.Models.HappyStoreEntity;

7 │ using QuanLyBanHang.Models.QLBHEntity;

8 │ using QuanLyBanHang.Models.VCEntity;

9 │ using QuanLyBanHang.Models.VCEntity.ExModel;

10 │ using QuanLyBanHang.Services;

11 │

12 │ namespace QuanLyBanHang.Controllers.SYS

13 │ {

14 │ public class RestoreController : BaseController

15 │ {

16 │ private HappyStoreContext storeContext { get; set; }

17 │ private QuanLyVoucherContext voucherContext { get; set; }

18 │

19 │ public RestoreController(QuanLyBanHangContext dbcontext, HappyStoreContext \_storecontext, QuanLyVoucherContext \_...

20 │ {

21 │ QLBHConst.wwwRoot = hostingEnvironment.WebRootPath;

22 │ m\_MenuID = Utility.defaultUID;

23 │ storeContext = \_storecontext;

24 │ voucherContext = \_voucherContext;

25 │ }

26 │

27 │ public IActionResult Index()

28 │ {

29 │ return View();

30 │ }

31 │ [HttpPost]

32 │ public async Task<IActionResult> RestoreTonKho(string namThang)

33 │ {

34 │ var hangHoaDict = dbcontext.TblHangHoa.Where(x => x.MaChiNhanh == Branch.BVDaKhoaHongHung.GetEnumGuid()).AsN...

35 │

36 │ var khoDict = dbcontext.StockDanhMucKho.Where(x => x.MaChiNhanh == Branch.BVDaKhoaHongHung.GetEnumGuid()).As...

37 │

38 │ var oldTonKhoList = (from tk in storeContext.TonKho.Where(x => x.NamThang == namThang && x.SltonCuoiKy > 0)

39 │ join hh in storeContext.HangHoa.Where(x => !(x.DaXoa ?? false)) on tk.MaHangHoa equals ...

40 │ join kho in storeContext.TblDanhMucKho.Where(x => !(x.DaXoa ?? false)) on tk.MaKho equa...

41 │ join ctnh in storeContext.CtNhapHang.Where(x => !(x.DaXoa ?? false)) on new { MaNhapHan...

42 │ from ctnh in nh0.DefaultIfEmpty()

43 │ select new

44 │ {

45 │ tk,

46 │ GiaVon = ctnh != null ? ctnh.GiaVon : 0,

47 │ HanSD = ctnh != null ? ctnh.HanSd : null,

48 │ MaVach = hh.MaVach,

49 │ MaKT = hh.MaKt,

50 │ CodeKho = kho.Code

51 │ }).ToList();

52 │

53 │ var newTonKhoList = new List<OldTonKho>();

54 │ var listKho = dbcontext.StockDanhMucKho.Where(x => x.MaChiNhanh == Branch.BVDaKhoaHongHung.GetEnumGuid()).To...

55 │ using (var transaction = dbcontext.Database.BeginTransaction())

56 │ {

57 │ try

58 │ {

59 │

60 │ foreach (var item in oldTonKhoList)

61 │ {

62 │ if (hangHoaDict.TryGetValue(item.MaKT, out var newMaHangHoa) &&

63 │ khoDict.TryGetValue(item.CodeKho, out var newMaKho))

64 │ {

65 │

66 │ newTonKhoList.Add(new OldTonKho

67 │ {

68 │ MaChiNhanh = Branch.BVDaKhoaHongHung.GetEnumGuid(),

69 │ MaChiNhanhInt = 2,

70 │ NamThang = namThang,

71 │ MaKho = newMaKho,

72 │ MaHangHoa = newMaHangHoa,

73 │ MaNhapHang = item.tk.MaNhapHang,

74 │ GiaVon = item.GiaVon,

75 │ HanSd = item.HanSD,

76 │ SltonDauKy = item.tk.SltonCuoiKy ?? 0,

77 │ SlnhapTrongKy = item.tk.SlnhapTrongKy ?? 0,

78 │ SlxuatKhac = item.tk.SlxuatKhac ?? 0,

79 │ SlxuatTieuDung = item.tk.SlxuatTieuDung ?? 0,

80 │ SlxuatTra = item.tk.SlxuatTra ?? 0,

81 │ SlbanTrongKy = item.tk.SlbanTrongKy ?? 0,

82 │ SltonCuoiKy = item.tk.SltonCuoiKy,

83 │

84 │ });

85 │ }

86 │ }

87 │ var delQuery = dbcontext.OldTonKho.Where(x => x.NamThang == namThang).ToList();

88 │ await dbcontext.BulkDeleteAsync(delQuery);

89 │ await dbcontext.BulkInsertAsync(newTonKhoList);

90 │ transaction.Commit();

91 │

92 │ return Json(new ResponseModelSuccess($"Đã restore Tồn kho từ db cũ, {oldTonKhoList.Count} dò...

93 │ }

94 │ catch (Exception ex)

95 │ {

96 │ transaction.Rollback();

97 │ return Json(new ResponseModelError(ex.Message));

98 │ }

99 │ }

100 │ }

101 │

102 │ public async Task<IActionResult> SyncTonKho(string namThang)

103 │ {

104 │ var brandId = Branch.BVDaKhoaHongHung.GetEnumGuid();

105 │ var listKho = dbcontext.StockDanhMucKho.Where(x => !(x.Deleted ?? false)).ToList();

106 │ var ngayTao = DateTime.Now;

107 │ var br = dbcontext.SysBranch.FirstOrDefault(x => x.Ma == brandId);

108 │

109 │

110 │ using (var transaction = dbcontext.Database.BeginTransaction())

111 │ {

112 │ try

113 │ {

114 │ var nhapHangDelete = dbcontext.StockNhapHang.Where(x => x.MaLoaiNx == (int)LoaiNhapXuat.PhieuNhapDon...

115 │ foreach(var nh in nhapHangDelete)

116 │ {

117 │ var ctnh = dbcontext.StockChiTietNhapHang.Where(x => x.MaNhapHang == nh.Ma).ToList();

118 │ await dbcontext.BulkDeleteAsync(ctnh);

119 │ }

120 │ await dbcontext.BulkDeleteAsync(nhapHangDelete);

121 │

122 │ foreach (var kho in listKho)

123 │ {

124 │ var listTonKhoByKho = dbcontext.OldTonKho.Where(x => x.NamThang == namThang && x.MaChiNhanh == b...

125 │ var nhapHang = new StockNhapHang

126 │ {

127 │ SoHoaDon = "",

128 │ MaLoaiNx = (int)LoaiNhapXuat.PhieuNhapDongBo,

129 │ NgayNhapHang = ngayTao,

130 │ NgayHoaDon = ngayTao,

131 │ MaNhaCc = 936,

132 │ MaKho = kho.Ma,

133 │ TienChietKhau = 0,

134 │ ThueVat = 0,

135 │ GhiChu = "Nhập hàng từ data cũ",

136 │ UserCreated = Utility.defaultUID,

137 │ CreatedDate = ngayTao,

138 │ KhoaSo = false,

139 │ MaChiNhanh = brandId,

140 │ MaChiNhanhInt = br.Nid ?? 0,

141 │ };

142 │

143 │ dbcontext.StockNhapHang.Add(nhapHang);

144 │ dbcontext.SaveChanges();

145 │

146 │ var ctNhapHang = listTonKhoByKho.Select(t => new StockChiTietNhapHang

147 │ {

148 │ MaNhapHang = nhapHang.Ma,

149 │ MaChiNhanhInt = br.Nid ?? 0,

150 │ MaHangHoa = t.MaHangHoa,

151 │ SoLuong = t.SltonCuoiKy,

152 │ GiaVon = t.GiaVon,

153 │ HanSd = t.HanSd,

154 │ //ChietKhauTm = (decimal?)ct.ChietKhauTM,

155 │ //ChietKhauPt = (decimal?)ct.ChietKhauPT,

156 │ //Vat = ct.VAT,

157 │ ThanhTien = t.SltonCuoiKy \* t.GiaVon,

158 │ //ThanhTienSauCk = thanhtiensauCK,

159 │ //ThanhTienSauVat = thanhtiensauVAT,

160 │ UserCreated = Utility.defaultUID,

161 │ CreatedDate = ngayTao

162 │ }).ToList();

163 │

164 │ await dbcontext.BulkInsertAsync(ctNhapHang);

165 │ TonKhoBackgroundWorker.AddJob(new TonKhoJob

166 │ {

167 │ MaChiNhanh = Branch.BVDaKhoaHongHung.GetEnumGuid(),

168 │ MaKho = kho.Ma,

169 │ NamThang = namThang,

170 │ UserId = Utility.defaultUID

171 │ });

172 │ }

173 │ transaction.Commit();

174 │

175 │ return Json(new ResponseModelSuccess($"Đã sync tồn kho"));

176 │ }

177 │ catch (Exception ex)

178 │ {

179 │ transaction.Rollback();

180 │ return Json(new ResponseModelError(ex.Message));

181 │ }

182 │ }

183 │

184 │ }

185 │

186 │ public async Task<IActionResult> SyncHangHoa()

187 │ {

188 │ var oldList = await storeContext.HangHoa.AsNoTracking().ToListAsync();

189 │

190 │ var currentHangHoaDict = await dbcontext.TblHangHoa.Where(x => x.MaChiNhanh == Branch.BVDaKhoaHongHung.GetEn...

191 │

192 │ var dvtDict = await dbcontext.TblDonViTinh.AsNoTracking().ToDictionaryAsync(x => x.Ten, x => x.Ma);

193 │ var lopHangDict = await dbcontext.TblLopHangHoa.Where(x => x.MaChiNhanh == Branch.BVDaKhoaHongHung.GetEnumGu...

194 │ var br = await dbcontext.SysBranch.FirstOrDefaultAsync(x => x.Ma == Branch.BVDaKhoaHongHung.GetEnumGuid());

195 │

196 │ var listToAdd = new List<TblHangHoa>();

197 │ var listToUpdate = new List<TblHangHoa>();

198 │

199 │ var oldDvtDict = await storeContext.DonViTinh.AsNoTracking().ToDictionaryAsync(d => d.Ma, d => d.Ten);

200 │ var oldLopHangDict = await storeContext.LopHang.AsNoTracking().ToDictionaryAsync(d => d.LopId, d => d.TenLop...

201 │ var currentTest = new HangHoa();

202 │

203 │ using (var transaction = await dbcontext.Database.BeginTransactionAsync())

204 │ {

205 │ try

206 │ {

207 │ foreach (var oldItem in oldList)

208 │ {

209 │ currentTest = oldItem;

210 │ if (!oldDvtDict.TryGetValue(oldItem.Dvt, out var tenDvt))

211 │ {

212 │ continue;

213 │ }

214 │

215 │ if (!dvtDict.TryGetValue(tenDvt, out var newMaDvt))

216 │ {

217 │ continue;

218 │ }

219 │ if (!oldLopHangDict.TryGetValue(oldItem.LopHang, out var tenLopHang))

220 │ {

221 │ continue;

222 │ }

223 │

224 │ if (!lopHangDict.TryGetValue(tenLopHang, out var newMaLopHang))

225 │ {

226 │ continue;

227 │ }

228 │

229 │ if (currentHangHoaDict.TryGetValue(oldItem.MaKt, out var existingItem))

230 │ {

231 │ // ---- NẾU ĐÃ TỒN TẠI -> CẬP NHẬT ----

232 │ existingItem.MaChiNhanhInt = br.Nid ?? 0;

233 │ existingItem.MaVach = oldItem.MaVach;

234 │ existingItem.MaLopHang = newMaLopHang;

235 │ existingItem.Ten = oldItem.TenHang;

236 │ existingItem.MaDvt = newMaDvt;

237 │ existingItem.GiaBan = oldItem.GiaBan;

238 │ existingItem.GiaVon = oldItem.GiaVon;

239 │ existingItem.HinhAnh = oldItem.HinhAnh;

240 │ existingItem.GhiChu = oldItem.GhiChu;

241 │ existingItem.BanNoiBo = oldItem.BanNoiBo;

242 │ existingItem.HetHang = oldItem.HetHang;

243 │ existingItem.Status = oldItem.TrangThai;

244 │

245 │ listToUpdate.Add(existingItem);

246 │ }

247 │ else

248 │ {

249 │ // ---- NẾU CHƯA TỒN TẠI -> THÊM MỚI ----

250 │ var newItem = new TblHangHoa

251 │ {

252 │ MaChiNhanhInt = br.Nid ?? 0,

253 │ MaChiNhanh = br.Ma,

254 │ MaKt = oldItem.MaKt,

255 │ MaVach = oldItem.MaVach,

256 │ MaLopHang = newMaLopHang,

257 │ Ten = oldItem.TenHang,

258 │ MaDvt = newMaDvt,

259 │ GiaBan = oldItem.GiaBan,

260 │ GiaVon = oldItem.GiaVon,

261 │ HinhAnh = oldItem.HinhAnh,

262 │ GhiChu = oldItem.GhiChu,

263 │ BanNoiBo = oldItem.BanNoiBo,

264 │ HetHang = oldItem.HetHang,

265 │ Status = oldItem.TrangThai,

266 │ };

267 │ listToAdd.Add(newItem);

268 │ }

269 │ }

270 │

271 │ if (listToAdd.Any())

272 │ {

273 │ await dbcontext.BulkInsertAsync(listToAdd);

274 │ }

275 │ if (listToUpdate.Any())

276 │ {

277 │ await dbcontext.BulkUpdateAsync(listToUpdate);

278 │ }

279 │

280 │ await transaction.CommitAsync();

281 │

282 │ string message = $"Đồng bộ hóa hàng hóa thành công. Thêm mới: {listToAdd.Count} dòng. ...

283 │ return Json(new ResponseModelSuccess(message));

284 │ }

285 │ catch (Exception ex)

286 │ {

287 │ await transaction.RollbackAsync();

288 │ return Json(new ResponseModelError(ex.Message, new { test = currentTest}));

289 │ }

290 │ }

291 │ }

292 │

293 │ public async Task<IActionResult> SyncLopHangHoa()

294 │ {

295 │ try

296 │ {

297 │ var oldLopHangList = await storeContext.LopHang.AsNoTracking().ToListAsync();

298 │ var br = await dbcontext.SysBranch.FirstOrDefaultAsync(x => x.Ma == Branch.BVDaKhoaHongHung.GetEnumGuid(...

299 │

300 │ var currentLopHangDict = await dbcontext.TblLopHangHoa

301 │ .Where(l => l.MaChiNhanh == br.Ma && l.Ten != null)

302 │ .AsNoTracking()

303 │ .ToDictionaryAsync(l => l.Ten!);

304 │

305 │ var listToAdd = new List<TblLopHangHoa>();

306 │ var listToUpdate = new List<TblLopHangHoa>();

307 │

308 │ foreach (var oldLop in oldLopHangList)

309 │ {

310 │ if (string.IsNullOrEmpty(oldLop.TenLop))

311 │ {

312 │ continue;

313 │ }

314 │

315 │ if (currentLopHangDict.TryGetValue(oldLop.TenLop, out var existingLop))

316 │ {

317 │ existingLop.ChoPhepBanAm = oldLop.ChoPhepBanAm;

318 │ existingLop.ThoiGianBatDau = oldLop.ThoiGianBatDau;

319 │ existingLop.ThoiGianKetThuc = oldLop.ThoiGianKetThuc;

320 │ existingLop.Status = oldLop.TrangThai;

321 │

322 │ listToUpdate.Add(existingLop);

323 │ }

324 │ else

325 │ {

326 │ // ---- NẾU CHƯA TỒN TẠI -> THÊM MỚI ----

327 │ var newLop = new TblLopHangHoa

328 │ {

329 │ MaChiNhanh = br.Ma,

330 │ Ten = oldLop.TenLop,

331 │ ChoPhepBanAm = oldLop.ChoPhepBanAm,

332 │ ThoiGianBatDau = oldLop.ThoiGianBatDau,

333 │ ThoiGianKetThuc = oldLop.ThoiGianKetThuc,

334 │ Status = oldLop.TrangThai

335 │ };

336 │ listToAdd.Add(newLop);

337 │ }

338 │ }

339 │

340 │ using (var transaction = await dbcontext.Database.BeginTransactionAsync())

341 │ {

342 │ if (listToAdd.Any())

343 │ {

344 │ await dbcontext.BulkInsertAsync(listToAdd);

345 │ }

346 │ if (listToUpdate.Any())

347 │ {

348 │ await dbcontext.BulkUpdateAsync(listToUpdate);

349 │ }

350 │

351 │ await transaction.CommitAsync();

352 │ }

353 │

354 │ string message = $"Đồng bộ lớp hàng hóa thành công. Thêm mới: {listToAdd.Count}. Cập nh�...

355 │ return Json(new ResponseModelSuccess(message));

356 │ }

357 │ catch (Exception ex)

358 │ {

359 │ // Nên ghi log lỗi chi tiết tại đây

360 │ return Json(new ResponseModelError($"Đã xảy ra lỗi khi đồng bộ lớp hàng hóa: {ex.Message}...

361 │ }

362 │ }

363 │ public async Task<IActionResult> SyncGiaBanTheoKho()

364 │ {

365 │ try

366 │ {

367 │ // === PHẦN CHUẨN BỊ DỮ LIỆU (giữ nguyên) ===

368 │ var maKtToNewHhIdMap = await dbcontext.TblHangHoa.Where(x => x.MaChiNhanh == Branch.BVDaKhoaHongHung.Get...

369 │ var tenKhoToNewKhoIdMap = await dbcontext.StockDanhMucKho.Where(x => x.MaChiNhanh == Branch.BVDaKhoaHong...

370 │

371 │ var oldPricesWithKeys = await (from gb in storeContext.TblGiaBanTheoKho

372 │ join hh in storeContext.HangHoa on gb.MaHangHoa equals hh.HangHoaId

373 │ join k in storeContext.TblDanhMucKho on gb.MaKho equals k.Ma

374 │ select new

375 │ {

376 │ HhMaKt = hh.MaKt,

377 │ KhoTen = k.Ten,

378 │ GiaBan = gb.GiaBan,

379 │ GhiChu = gb.GhiChu,

380 │ Status = gb.TrangThai,

381 │ }).AsNoTracking().ToListAsync();

382 │

383 │ var allNewPrices = await dbcontext.TblGiaBanTheoKho

384 │ .AsNoTracking()

385 │ .Where(gb => gb.MaHangHoa != null && gb.MaKho != null)

386 │ .ToListAsync();

387 │

388 │ var newPricesDict = allNewPrices

389 │ .GroupBy(gb => (gb.MaHangHoa.Value, gb.MaKho.Value))

390 │ .ToDictionary(g => g.Key, g => g.First());

391 │

392 │ var listToAdd = new List<Models.QLBHEntity.TblGiaBanTheoKho>();

393 │ var listToUpdate = new List<Models.QLBHEntity.TblGiaBanTheoKho>();

394 │ var br = await dbcontext.SysBranch.FirstOrDefaultAsync(x => x.Ma == Branch.BVDaKhoaHongHung.GetEnumGuid(...

395 │

396 │ var distinctOldPrices = oldPricesWithKeys

397 │ .GroupBy(p => (p.HhMaKt, p.KhoTen))

398 │ .Select(g => g.Last());

399 │ // ------------------------------------

400 │

401 │ foreach (var oldPrice in distinctOldPrices)

402 │ {

403 │ if (string.IsNullOrEmpty(oldPrice.HhMaKt) || string.IsNullOrEmpty(oldPrice.KhoTen))

404 │ {

405 │ continue;

406 │ }

407 │

408 │ if (!maKtToNewHhIdMap.TryGetValue(oldPrice.HhMaKt, out var newHhId) ||

409 │ !tenKhoToNewKhoIdMap.TryGetValue(oldPrice.KhoTen, out var newKhoId))

410 │ {

411 │ continue;

412 │ }

413 │

414 │ var compositeKey = (newHhId, newKhoId);

415 │

416 │ if (newPricesDict.TryGetValue(compositeKey, out var existingPrice))

417 │ {

418 │ // NẾU TỒN TẠI -> CẬP NHẬT

419 │ existingPrice.GiaBan = oldPrice.GiaBan;

420 │ existingPrice.GhiChu = oldPrice.GhiChu;

421 │ existingPrice.Status = oldPrice.Status;

422 │ existingPrice.ModifiedDate = DateTime.Now;

423 │

424 │ listToUpdate.Add(existingPrice);

425 │ }

426 │ else

427 │ {

428 │ // NẾU KHÔNG TỒN TẠI -> THÊM MỚI

429 │ var newPrice = new Models.QLBHEntity.TblGiaBanTheoKho

430 │ {

431 │ MaChiNhanhInt = br?.Nid ?? 0,

432 │ MaChiNhanh = br?.Ma,

433 │ MaHangHoa = newHhId,

434 │ MaKho = newKhoId,

435 │ GiaBan = oldPrice.GiaBan,

436 │ GhiChu = oldPrice.GhiChu,

437 │ Status = oldPrice.Status,

438 │ CreatedDate = DateTime.Now,

439 │ };

440 │ listToAdd.Add(newPrice);

441 │ }

442 │ }

443 │

444 │ // === PHẦN THỰC THI (giữ nguyên) ===

445 │ using (var transaction = await dbcontext.Database.BeginTransactionAsync())

446 │ {

447 │ if (listToAdd.Any())

448 │ {

449 │ await dbcontext.BulkInsertAsync(listToAdd);

450 │ }

451 │ if (listToUpdate.Any())

452 │ {

453 │ await dbcontext.BulkUpdateAsync(listToUpdate);

454 │ }

455 │ await transaction.CommitAsync();

456 │ }

457 │

458 │ string message = $"Đồng bộ giá bán thành công. Thêm mới: {listToAdd.Count}. Cập nhật: {l...

459 │ return Json(new ResponseModelSuccess(message));

460 │ }

461 │ catch (Exception ex)

462 │ {

463 │ return Json(new ResponseModelError($"Đã xảy ra lỗi: {ex.Message}"));

464 │ }

465 │ }

466 │ public async Task<IActionResult> AddThanhToanVoucher(string Serial, int maNganh)

467 │ {

468 │ var hoadon = await dbcontext.OpsHoaDon.FirstOrDefaultAsync(x => x.SoSeri == Serial);

469 │ if (hoadon == null) return Json(new ResponseModelError($"Không tìm thấy hóa đơn"));

470 │

471 │ var ctHoaDon = await dbcontext.OpsChiTietHoaDon.Where(x => x.MaHoaDon == hoadon.Ma).ToListAsync();

472 │ if (hoadon == null) return Json(new ResponseModelError($"Không tìm thấy ct hóa đơn"));

473 │

474 │ var ngayThanhToan = hoadon.NgayThanhToan ?? DateTime.Now;

475 │ var maChiNhanh = Branch.BVDaKhoaHongHung.GetEnumGuid();

476 │ var maChiNhanhVoucher = 3;

477 │ var maTheVoucher = hoadon.MaVachThe;

478 │ var voucherRs = new ThanhToanVoucherModel(voucherContext).GetVoucher(maTheVoucher ?? "", maChiNhanhVoucher, ...

479 │ var voucher = voucherRs.Voucher;

480 │ var Tienthe = 0;

481 │ var PhuThu = 0;

482 │ var ThanhTien = Cast.ToInt(hoadon.PhaiThu);

483 │ if ((voucher?.SoDu ?? 0) >= ThanhTien)

484 │ {

485 │ Tienthe = ThanhTien;

486 │

487 │ }

488 │ else

489 │ {

490 │ Tienthe = voucher?.SoDu ?? 0;

491 │ PhuThu = hoadon.TienKhachTra - hoadon.TienThua ?? 0;

492 │ }

493 │ using (var transaction = await voucherContext.Database.BeginTransactionAsync())

494 │ {

495 │ try

496 │ {

497 │ ThemThanhToan(ngayThanhToan.ToString("ddMMyyHHmmssfff"), maChiNhanhVoucher, maNganh, voucher.Ma ?? 0...

498 │

499 │ await transaction.CommitAsync();

500 │ return Json(new ResponseModelSuccess($"Thêm thành công. Mã voucher : {maTheVoucher}"));

501 │ }

502 │ catch (Exception ex)

503 │ {

504 │ await transaction.RollbackAsync();

505 │ return Json(new ResponseModelError($"Đã xảy ra lỗi: {ex.Message}"));

506 │ }

507 │

508 │ }

509 │ }

510 │ public int ThemThanhToan(

511 │ string code, int maChiNhanh, int maNganh, int maBanHang, int giaTriThanhToan,

512 │ int tienThe, int phuThu, string ghiChu, int nguoiTao, DateTime ngayTao, int hoaDon\_Id)

513 │ {

514 │ var banHang = voucherContext.TblBanHang.FirstOrDefault(bh => bh.Ma == maBanHang);

515 │ if (banHang == null) return -1;

516 │

517 │ int soDu = banHang.SoDu;

518 │ int conLai = soDu - tienThe;

519 │ if (conLai < 0) return -1;

520 │

521 │ var thanhToan = new TblThanhToan

522 │ {

523 │ Code = code,

524 │ MaChiNhanh = maChiNhanh,

525 │ MaNganh = maNganh,

526 │ MaBanHang = maBanHang,

527 │ GiaTriThanhToan = tienThe,

528 │ SoDu = conLai,

529 │ PhuThu = phuThu,

530 │ GhiChu = ghiChu,

531 │ NguoiTao = nguoiTao,

532 │ NgayTao = ngayTao

533 │ };

534 │

535 │ voucherContext.TblThanhToan.Add(thanhToan);

536 │ voucherContext.SaveChanges();

537 │

538 │ int maThanhToan = thanhToan.Ma;

539 │ if (maThanhToan > 0)

540 │ {

541 │ banHang.SoDu = conLai;

542 │ banHang.NguoiSua = nguoiTao;

543 │ banHang.NgaySua = ngayTao;

544 │

545 │ var hoaDon = dbcontext.OpsHoaDon.FirstOrDefault(h => h.Ma == hoaDon\_Id);

546 │ if (hoaDon != null)

547 │ {

548 │ hoaDon.MaThanhToan = maThanhToan;

549 │ hoaDon.NgayThanhToan = ngayTao;

550 │ hoaDon.TienThe = tienThe;

551 │ }

552 │

553 │ dbcontext.SaveChanges();

554 │ voucherContext.SaveChanges();

555 │ return maThanhToan;

556 │ }

557 │

558 │ return -1;

559 │ }

560 │ }

561 │ }

────────────────────────────────────────────────────────────

**📄 RfidHangHoa.cs (CS, 10 lines)**

1 │ ﻿namespace QuanLyBanHang.Models.DTO

2 │ {

3 │ public class RfidHangHoa

4 │ {

5 │ public string RfID { get; set; }

6 │ public string TenHangHoa { get; set; }

7 │ public string MaDVT { get; set; }

8 │ public string GiaVon { get; set; }

9 │ }

10 │ }

────────────────────────────────────────────────────────────

**📄 RoleController.cs (CS, 66 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using QuanLyBanHang.Common;

3 │ using QuanLyBanHang.Common.Attributes;

4 │ using QuanLyBanHang.Models.ExModel.SYS;

5 │ using QuanLyBanHang.Models.QLBHEntity;

6 │

7 │ namespace QuanLyBanHang.Controllers.SYS

8 │ {

9 │ public class RoleController : BaseController

10 │ {

11 │ public RoleController(QuanLyBanHangContext dbcontext, MemoryCacheManager memoryCache) : base(dbcontext, memoryCa...

12 │ {

13 │ m\_MenuID = MenuID.sysRole.GetEnumGuid();

14 │ }

15 │

16 │ [TokenUserCheckHTTP()]

17 │ public IActionResult Index()

18 │ {

19 │ ViewData["ListData"] = new sys\_RoleModel(dbcontext).GetDataVoiChucVu().OrderBy(t => t.CreatedDate).ToList();

20 │ return View();

21 │ }

22 │

23 │ [TokenUserCheckHTTP(Permission.Add)]

24 │ public ActionResult Add()

25 │ {

26 │ ViewData["ListChucVu"] = dbcontext.TblChucVu.Where(t => !(t.Deleted ?? false)).ToList();

27 │ return View("Form");

28 │ }

29 │

30 │ [TokenUserCheckHTTP(Permission.Edit)]

31 │ public ActionResult Edit(string ID)

32 │ {

33 │ ViewData["item"] = new sys\_RoleModel(dbcontext).GetItem(ID.GetGuid());

34 │ ViewData["ListChucVu"] = dbcontext.TblChucVu.Where(t => !(t.Deleted ?? false)).ToList();

35 │ return View("Form");

36 │ }

37 │ [HttpPost]

38 │ [TokenUserCheckHTTP(Permission.Edit)]

39 │ public ActionResult UpdateAPI(string ID, string Code, string Name, string MaChucVu, string Description

40 │ , string Status)

41 │ {

42 │ int result = new sys\_RoleModel(dbcontext).Edit(ID.GetGuid(), Code, Name, MaChucVu.GetGuid(), Description

43 │ , Status == "1", objToken!.userid);

44 │ return Json(Utility.MakeResponseFromEditResult(Cast.ToString(result)));

45 │ }

46 │

47 │ [HttpPost]

48 │ [TokenUserCheckHTTP(Permission.Add)]

49 │ public ActionResult AddAPI(string Code, string Name, string MaChucVu, string Description, string Status)

50 │ {

51 │ int result = new sys\_RoleModel(dbcontext).Add(Code, Name, MaChucVu.GetGuid(), Description, Status == "1", ob...

52 │ return Json(Utility.MakeResponseFromAddResult(Cast.ToString(result)));

53 │ }

54 │

55 │ [HttpPost]

56 │ [TokenUserCheckHTTP(Permission.Delete)]

57 │ public ActionResult DeleteAPI(string ID)

58 │ {

59 │ //

60 │ string result = Cast.ToString(new sys\_RoleModel(dbcontext).Delete(ID.GetGuid(), objToken!.userid));

61 │ return Json(Utility.MakeResponseFromDeleteResult(Cast.ToString(result)));

62 │ }

63 │

64 │

65 │ }

66 │ }

────────────────────────────────────────────────────────────

**📄 RoleMenuController.cs (CS, 74 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using QuanLyBanHang.Common;

3 │ using QuanLyBanHang.Common.Attributes;

4 │ using QuanLyBanHang.Models.ExModel.SYS;

5 │ using QuanLyBanHang.Models.QLBHEntity;

6 │

7 │ namespace QuanLyBanHang.Controllers.SYS

8 │ {

9 │ public class RoleMenuController : BaseController

10 │ {

11 │ public RoleMenuController(QuanLyBanHangContext dbcontext, MemoryCacheManager memoryCache) : base(dbcontext, memo...

12 │ {

13 │ m\_MenuID = MenuID.sysRoleMenu.GetEnumGuid();

14 │ }

15 │

16 │ [TokenUserCheckHTTP()]

17 │ public IActionResult Detail(string RoleID)

18 │ {

19 │ ViewData["ListData"] = new sys\_RoleMenuModel(dbcontext).GetDataFollowRoleID(RoleID.GetGuid());

20 │ ViewData["ListMenu"] = new sys\_MenuModel(dbcontext).GetData();

21 │ ViewData["RoleItem"] = new sys\_RoleModel(dbcontext).GetItem(RoleID.GetGuid());

22 │ return View("Index");

23 │ }

24 │

25 │

26 │ [TokenUserCheckHTTP(Permission.Add)]

27 │ public IActionResult Add(string RoleID)

28 │ {

29 │ ViewData["RoleItem"] = new sys\_RoleModel(dbcontext).GetItem(RoleID.GetGuid());

30 │ ViewData["ListMenu"] = new sys\_MenuModel(dbcontext).GetData().Where(t => (t.Status ?? false)).ToList();

31 │ return View("Form");

32 │ }

33 │

34 │ [TokenUserCheckHTTP(Permission.Edit)]

35 │ public IActionResult Edit(string ID)

36 │ {

37 │ SysRoleMenu drRM = new sys\_RoleMenuModel(dbcontext).GetItem(ID.GetGuid());

38 │ ViewData["item"] = drRM;

39 │ ViewData["ListMenu"] = new sys\_MenuModel(dbcontext).GetData().Where(t => (t.Status ?? false)).ToList();

40 │

41 │ if (drRM != null)

42 │ ViewData["RoleItem"] = new sys\_RoleModel(dbcontext).GetItem(drRM.RoleId);

43 │

44 │ return View("Form");

45 │ }

46 │ [HttpPost]

47 │ [TokenUserCheckHTTP(Permission.Edit)]

48 │ public IActionResult UpdateAPI(string ID, string Menu, string AllowAccess, string Status)

49 │ {

50 │ int result = new sys\_RoleMenuModel(dbcontext).Edit(ID.GetGuid(), Menu.GetGuid(), AllowAccess == "1"

51 │ , Status == "1", objToken!.userid);

52 │ memoryCache.Clear(t => t.Contains(QLBHConst.cacheKeyMenu));

53 │ return Json(Utility.MakeResponseFromEditResult(Cast.ToString(result)));

54 │

55 │ }

56 │ [HttpPost]

57 │ [TokenUserCheckHTTP(Permission.Add)]

58 │ public IActionResult AddAPI(string RoleID, string Menu, string AllowAccess, string Status)

59 │ {

60 │ int result = new sys\_RoleMenuModel(dbcontext).Add(RoleID.GetGuid(), Menu.GetGuid(), AllowAccess == "1"

61 │ , Status == "1", objToken!.userid);

62 │ return Json(Utility.MakeResponseFromAddResult(Cast.ToString(result)));

63 │ }

64 │

65 │ [HttpPost]

66 │ [TokenUserCheckHTTP(Permission.Delete)]

67 │ public IActionResult DeleteAPI(string ID)

68 │ {

69 │ int result = new sys\_RoleMenuModel(dbcontext).Delete(ID.GetGuid(), objToken!.userid);

70 │ memoryCache.Clear(t => t.Contains(QLBHConst.cacheKeyMenu));

71 │ return Json(Utility.MakeResponseFromDeleteResult(Cast.ToString(result)));

72 │ }

73 │ }

74 │ }

────────────────────────────────────────────────────────────

**📄 RolePermissionController.cs (CS, 161 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using QuanLyBanHang.Common;

3 │ using QuanLyBanHang.Common.Attributes;

4 │ using QuanLyBanHang.Models.APIModels;

5 │ using QuanLyBanHang.Models.ExModel.SYS;

6 │ using QuanLyBanHang.Models.QLBHEntity;

7 │

8 │ namespace QuanLyBanHang.Controllers.SYS

9 │ {

10 │ public class RolePermissionController : BaseController

11 │ {

12 │ public RolePermissionController(QuanLyBanHangContext dbcontext, MemoryCacheManager memoryCache) : base(dbcontext...

13 │ {

14 │ m\_MenuID = MenuID.sysRolePermission.GetEnumGuid();

15 │ }

16 │

17 │ [TokenUserCheckHTTP()]

18 │ public IActionResult Detail(string RoleMenuID)

19 │ {

20 │ SysRoleMenu drRM = new sys\_RoleMenuModel(dbcontext).GetItem(RoleMenuID.GetGuid());

21 │ ViewData["ListData"] = new sys\_RolePermissionModel(dbcontext).GetDataFollowRoleMenuID(RoleMenuID.GetGuid());

22 │ ViewData["ListPermission"] = new sys\_PermissionModel(dbcontext).GetData();

23 │ ViewData["RoleMenuItem"] = drRM;

24 │

25 │ if (drRM != null)

26 │ {

27 │ ViewData["RoleItem"] = new sys\_RoleModel(dbcontext).GetItem(drRM.RoleId);

28 │ ViewData["MenuItem"] = new sys\_MenuModel(dbcontext).GetItem(drRM.MenuId);

29 │ }

30 │ return View("index");

31 │ }

32 │

33 │ [TokenUserCheckHTTP(Permission.Add)]

34 │ public ActionResult Add(string RoleMenuID)

35 │ {

36 │ SysRoleMenu drRM = new sys\_RoleMenuModel(dbcontext).GetItem(RoleMenuID.GetGuid());

37 │ ViewData["RoleMenuItem"] = drRM;

38 │ ViewData["ListPermission"] = new sys\_PermissionModel(dbcontext).GetData().Where(t => (t.Status ?? false)).To...

39 │ if (drRM != null)

40 │ {

41 │ ViewData["RoleItem"] = new sys\_RoleModel(dbcontext).GetItem(drRM.RoleId);

42 │ ViewData["MenuItem"] = new sys\_MenuModel(dbcontext).GetItem(drRM.MenuId);

43 │ }

44 │ return View("Form");

45 │ }

46 │

47 │ [TokenUserCheckHTTP(Permission.Edit)]

48 │ public ActionResult Edit(string ID)

49 │ {

50 │ SysRolePermission drRP = new sys\_RolePermissionModel(dbcontext).GetItem(ID.GetGuid());

51 │ ViewData["item"] = drRP;

52 │ ViewData["ListPermission"] = new sys\_PermissionModel(dbcontext).GetData().Where(t => (t.Status ?? false)).To...

53 │

54 │ if (drRP != null)

55 │ {

56 │ SysRoleMenu drRM = new sys\_RoleMenuModel(dbcontext).GetItem(drRP.RoleMenuId);

57 │ ViewData["RoleMenuItem"] = drRM;

58 │ if (drRM != null)

59 │ {

60 │ ViewData["RoleItem"] = new sys\_RoleModel(dbcontext).GetItem(drRM.RoleId);

61 │ ViewData["MenuItem"] = new sys\_MenuModel(dbcontext).GetItem(drRM.MenuId);

62 │ }

63 │ }

64 │ return View("Form");

65 │ }

66 │

67 │ [HttpPost]

68 │ [TokenUserCheckHTTP(Permission.Delete)]

69 │ public ActionResult Delete(string ID)

70 │ {

71 │ string result = Cast.ToString(new sys\_RolePermissionModel(dbcontext).Delete(ID.GetGuid(), objToken!.userid))...

72 │ memoryCache.Clear(t => t.Contains(QLBHConst.cacheKeyMenu));

73 │ return Json(new ResponseModel(QLBHConst.OK, result == "1" ? "Thành công" : "Đã xảy ra lỗi " + result...

74 │ }

75 │ [HttpPost]

76 │ [TokenUserCheckHTTP(Permission.Add)]

77 │ public ActionResult AddAPI(string RoleMenuID, string PermissionID, string Status)

78 │ {

79 │ int result = new sys\_RolePermissionModel(dbcontext).Add(RoleMenuID.GetGuid()

80 │ , PermissionID.GetGuid(), Status == "1", objToken!.userid);

81 │ memoryCache.Clear(t => t.Contains(QLBHConst.cacheKeyMenu));

82 │ return Json(new ResponseModel(QLBHConst.OK, result == 1 ? "Thành công" : "Đã xảy ra lỗi " + result))...

83 │ }

84 │

85 │ [HttpPost]

86 │ [TokenUserCheckHTTP(Permission.Edit)]

87 │ public ActionResult UpdateAPI(string ID, string PermissionID, string Status)

88 │ {

89 │ int result = new sys\_RolePermissionModel(dbcontext).Edit(ID.GetGuid(), PermissionID.GetGuid()

90 │ , Status == "1", objToken!.userid);

91 │ memoryCache.Clear(t => t.Contains(QLBHConst.cacheKeyMenu));

92 │ return Json(new ResponseModel(QLBHConst.OK, result == 1 ? "Thành công" : "Đã xảy ra lỗi " + result))...

93 │ }

94 │

95 │ [TokenUserCheckHTTP()]

96 │ public ActionResult QuicklyPermission(string RoleID)

97 │ {

98 │ ViewData["RoleItem"] = new sys\_RoleModel(dbcontext).GetItem(RoleID.GetGuid());

99 │ ViewData["RoleID"] = RoleID;

100 │ ViewData["ListRoleMenu"] = new sys\_RoleMenuModel(dbcontext).GetRoleMenuFollowRoleID(RoleID.GetGuid());

101 │ ViewData["ListRolePermisson"] = new sys\_RolePermissionModel(dbcontext).GetRolePermissionFollowRoleID(RoleID....

102 │ return View();

103 │ }

104 │

105 │ [HttpPost]

106 │ [TokenUserCheckHTTP()]

107 │ public ActionResult UpdateQuicklyPermission(string RoleID, string RMAllowAccess, string RMStatus, string RPStatu...

108 │ {

109 │ bool kq = new sys\_RolePermissionModel(dbcontext).UpdateQuicklyPermission(RoleID.GetGuid(), RMAllowAccess + "...

110 │ , RPStatus + "", objToken!.userid);

111 │ memoryCache.Clear(t => t.Contains(QLBHConst.cacheKeyMenu));

112 │ return Json(new ResponseModel(QLBHConst.OK, kq ? "Thành công" : "Đã xảy ra lỗi "));

113 │ }

114 │

115 │ [HttpPost]

116 │ [TokenUserCheckHTTP(Permission.NotCheck)]

117 │ public IActionResult QuicklyPermissionGet(string RoleID)

118 │ {

119 │ var RoleItem = new sys\_RoleModel(dbcontext).GetItem(RoleID.GetGuid());

120 │ var ListRoleMenu = new sys\_RoleMenuModel(dbcontext).GetRoleMenuFollowRoleID(RoleID.GetGuid());

121 │ var ListRolePermisson = new sys\_RolePermissionModel(dbcontext).GetRolePermissionFollowRoleID(RoleID.GetGuid(...

122 │ var listData = new List<object>();

123 │

124 │ foreach (var RMItem in ListRoleMenu)

125 │ {

126 │ var permissions = ListRolePermisson.Where(t => t.MenuID == RMItem.MenuID).ToList();

127 │

128 │ if (permissions.Count == 0)

129 │ {

130 │ listData.Add(new

131 │ {

132 │ menu = RMItem.MenuName,

133 │ allowAccess = RMItem.AllowAccess == true ? "1" : "",

134 │ status = RMItem.Status == true ? "1" : "",

135 │ permission = "",

136 │ permissionStatus = ""

137 │ });

138 │ }

139 │ else

140 │ {

141 │ bool first = true;

142 │ foreach (var RPItem in permissions)

143 │ {

144 │ listData.Add(new

145 │ {

146 │ menu = first ? RMItem.MenuName : "",

147 │ allowAccess = RMItem.AllowAccess == true ? "1" : "",

148 │ status = RMItem.Status == true ? "1" : "",

149 │ permission = RPItem.PermissionName,

150 │ permissionStatus = RPItem.Status == true ? "1" : ""

151 │ });

152 │ first = false;

153 │ }

154 │ }

155 │ }

156 │

157 │ return Json(new ResponseModel(QLBHConst.OK, "", listData));

158 │ }

159 │ }

160 │

161 │ }

────────────────────────────────────────────────────────────

**📄 SMSController.cs (CS, 81 lines)**

1 │ ﻿using Newtonsoft.Json;

2 │ using Newtonsoft.Json.Linq;

3 │ using QuanLyBanHang.Common;

4 │ using System.Net.Http.Headers;

5 │ using System.Text;

6 │

7 │ namespace QuanLyBanHang.Controllers

8 │ {

9 │ public static class SMSController

10 │ {

11 │ private static string smsURLPublic = "http://123.31.36.151:8888/smsbn/api";

12 │ public static async Task<bool> sendOTP(string phone, string otp, string time) => await sendSMSVNPT(phone, "13772...

13 │ public static async Task<bool> sendThongBaoNo(string phone, string khachhang, string nganhhang, string sotienno,...

14 │ await sendSMSVNPT(phone, "1380291", khachhang, nganhhang,

15 │ sotienno + (!sotienthu.Contains(".") ? ".0" : ""),

16 │ sotienthu + (!sotienthu.Contains(".") ? ".0" : ""));

17 │

18 │ private static async Task<bool> sendSMSVNPT(string phone, string TEMPLATEID, params string[] arg)

19 │ {

20 │ if (phone.StartsWith("0"))

21 │ {

22 │ phone = "84" + phone.Substring(1);

23 │ }

24 │ if (phone.Length != 11)

25 │ {

26 │ return false;

27 │ }

28 │ JObject obj = (JObject)JsonConvert.DeserializeObject(await sms\_response(phone, TEMPLATEID, arg));

29 │ return (string)obj["RPLY"]["ERROR"] == "0";

30 │ }

31 │ public static async Task<string> sendThongBaoNoTest(string phone, string khachhang, string nganhhang, string sot...

32 │ {

33 │ return await sms\_response(phone, "1380291", khachhang, nganhhang, sotienno + (!sotienthu.Contains(".") ? ".0...

34 │ }

35 │

36 │ private static async Task<string> sms\_response(string phone, string TEMPLATEID, params string[] arg)

37 │ {

38 │ return await sms\_responses(new List<string>() { phone }, TEMPLATEID, arg);

39 │ }

40 │ private static async Task<string> sms\_responses(List<string> phone, string TEMPLATEID, params string[] arg)

41 │ {

42 │ var obj = new

43 │ {

44 │ RQST = new

45 │ {

46 │ name = "send\_sms\_list",

47 │ REQID = Utility.RandomString(5),

48 │ LABELID = "205840",

49 │ CONTRACTTYPEID = "1",

50 │ CONTRACTID = "16025",

51 │ TEMPLATEID,

52 │ PARAMS = arg.Select(t => new

53 │ {

54 │ NUM = (arg.ToList().IndexOf(t) + 1) + "",

55 │ CONTENT = Utility.BoDauTiengViet(t)

56 │ }),

57 │ SCHEDULETIME = "",

58 │ MOBILELIST = phone.Count == 1 ? phone.FirstOrDefault() : phone.Aggregate((a, b) => a + "," + b),

59 │ ISTELCOSUB = "0",

60 │ AGENTID = "181",

61 │ APIUSER = "Ctyhungduy",

62 │ APIPASS = "Vnpt@123",

63 │ USERNAME = "Ctyhungduy",

64 │ //DATACODING = "8"

65 │ }

66 │ };

67 │ HttpRequestMessage httpRequestMessage = new HttpRequestMessage

68 │ {

69 │ Method = HttpMethod.Post,

70 │ Content = new StringContent(JsonConvert.SerializeObject(obj), Encoding.UTF8, "application/json"),

71 │ };

72 │ httpRequestMessage.Headers.Accept.Add(new MediaTypeWithQualityHeaderValue("application/json"));

73 │ using (HttpClient client = new HttpClient() { BaseAddress = new Uri(smsURLPublic) })

74 │ {

75 │ var response = await client.SendAsync(httpRequestMessage);

76 │ return await response.Content.ReadAsStringAsync();

77 │ }

78 │ }

79 │ }

80 │ }

81 │

────────────────────────────────────────────────────────────

**📄 SerChiTietDinhMuc.cs (CS, 43 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class SerChiTietDinhMuc

7 │ {

8 │ public Guid Ma { get; set; }

9 │

10 │ public int MaChiNhanhInt { get; set; }

11 │

12 │ public Guid? MaDinhMuc { get; set; }

13 │

14 │ public Guid? MaHangHoa { get; set; }

15 │

16 │ public Guid? MaVatTu { get; set; }

17 │

18 │ public Guid? MaDvt { get; set; }

19 │

20 │ public decimal? SoLuong { get; set; }

21 │

22 │ public decimal? DonGiaTb { get; set; }

23 │

24 │ public decimal? ThanhTien { get; set; }

25 │

26 │ public string? GhiChu { get; set; }

27 │

28 │ public Guid? UserCreated { get; set; }

29 │

30 │ public DateTime? CreatedDate { get; set; }

31 │

32 │ public Guid? UserModified { get; set; }

33 │

34 │ public DateTime? ModifiedDate { get; set; }

35 │

36 │ public bool? Deleted { get; set; }

37 │

38 │ public Guid? UserDeleted { get; set; }

39 │

40 │ public DateTime? DeletedDate { get; set; }

41 │

42 │ public bool? KhongTruKho { get; set; }

43 │ }

────────────────────────────────────────────────────────────

**📄 SetUpMayInController.cs (CS, 37 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using QuanLyBanHang.Common;

3 │ using QuanLyBanHang.Common.Attributes;

4 │ using QuanLyBanHang.Controllers;

5 │ using QuanLyBanHang.Models.QLBHEntity;

6 │

7 │ namespace QuanLyBanHang.Models.ExModel.SYS

8 │ {

9 │ public class SetUpMayInController : BaseController

10 │

11 │ {

12 │ public SetUpMayInController(QuanLyBanHangContext dbcontext, MemoryCacheManager memoryCache) : base(dbcontext, me...

13 │ {

14 │ m\_MenuID = MenuID.sysSetUpMayIn.GetEnumGuid();

15 │ }

16 │

17 │ [TokenUserCheckHTTP()]

18 │ public ActionResult Index()

19 │ {

20 │ return View();

21 │ }

22 │

23 │ [TokenUserCheckHTTP()]

24 │ public ActionResult Add(string loai = "")

25 │ {

26 │ ViewData["loai"] = loai;

27 │ return View("Form");

28 │ }

29 │

30 │ [TokenUserCheckHTTP()]

31 │ public ActionResult Edit(string id)

32 │ {

33 │ ViewData["id"] = id;

34 │ return View("Form");

35 │ }

36 │ }

37 │ }

────────────────────────────────────────────────────────────

**📄 StockChiTietKiemKe.cs (CS, 33 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class StockChiTietKiemKe

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int MaChiNhanhInt { get; set; }

11 │

12 │ public int? MaKiemKe { get; set; }

13 │

14 │ public int? MaHangHoa { get; set; }

15 │

16 │ public decimal? SoLuong { get; set; }

17 │

18 │ public Guid? UserCreated { get; set; }

19 │

20 │ public DateTime? CreatedDate { get; set; }

21 │

22 │ public Guid? UserModified { get; set; }

23 │

24 │ public DateTime? ModifiedDate { get; set; }

25 │

26 │ public bool? Deleted { get; set; }

27 │

28 │ public Guid? UserDeleted { get; set; }

29 │

30 │ public DateTime? DeletedDate { get; set; }

31 │

32 │ public string? GhiChu { get; set; }

33 │ }

────────────────────────────────────────────────────────────

**📄 StockChiTietKiemKeNew.cs (CS, 33 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class StockChiTietKiemKeNew

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int MaChiNhanhInt { get; set; }

11 │

12 │ public int? MaKiemKe { get; set; }

13 │

14 │ public int? MaHangHoa { get; set; }

15 │

16 │ public decimal? SoLuong { get; set; }

17 │

18 │ public Guid? UserCreated { get; set; }

19 │

20 │ public DateTime? CreatedDate { get; set; }

21 │

22 │ public Guid? UserModified { get; set; }

23 │

24 │ public DateTime? ModifiedDate { get; set; }

25 │

26 │ public bool? Deleted { get; set; }

27 │

28 │ public Guid? UserDeleted { get; set; }

29 │

30 │ public DateTime? DeletedDate { get; set; }

31 │

32 │ public string? GhiChu { get; set; }

33 │ }

────────────────────────────────────────────────────────────

**📄 StockChiTietNhapHang.cs (CS, 49 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class StockChiTietNhapHang

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int MaChiNhanhInt { get; set; }

11 │

12 │ public int? MaNhapHang { get; set; }

13 │

14 │ public int? MaHangHoa { get; set; }

15 │

16 │ public int? MaDvt { get; set; }

17 │

18 │ public decimal? SoLuong { get; set; }

19 │

20 │ public decimal? GiaVon { get; set; }

21 │

22 │ public DateTime? HanSd { get; set; }

23 │

24 │ public decimal? ChietKhauTm { get; set; }

25 │

26 │ public decimal? ChietKhauPt { get; set; }

27 │

28 │ public int? Vat { get; set; }

29 │

30 │ public decimal? ThanhTien { get; set; }

31 │

32 │ public decimal? ThanhTienSauCk { get; set; }

33 │

34 │ public decimal? ThanhTienSauVat { get; set; }

35 │

36 │ public Guid? UserCreated { get; set; }

37 │

38 │ public DateTime? CreatedDate { get; set; }

39 │

40 │ public Guid? UserModified { get; set; }

41 │

42 │ public DateTime? ModifiedDate { get; set; }

43 │

44 │ public bool? Deleted { get; set; }

45 │

46 │ public Guid? UserDeleted { get; set; }

47 │

48 │ public DateTime? DeletedDate { get; set; }

49 │ }

────────────────────────────────────────────────────────────

**📄 StockChiTietNhapHangNew.cs (CS, 47 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class StockChiTietNhapHangNew

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int MaChiNhanhInt { get; set; }

11 │

12 │ public int? MaNhapHang { get; set; }

13 │

14 │ public int? MaHangHoa { get; set; }

15 │

16 │ public decimal? SoLuong { get; set; }

17 │

18 │ public decimal? GiaVon { get; set; }

19 │

20 │ public DateTime? HanSd { get; set; }

21 │

22 │ public decimal? ChietKhauTm { get; set; }

23 │

24 │ public decimal? ChietKhauPt { get; set; }

25 │

26 │ public int? Vat { get; set; }

27 │

28 │ public decimal? ThanhTien { get; set; }

29 │

30 │ public decimal? ThanhTienSauCk { get; set; }

31 │

32 │ public decimal? ThanhTienSauVat { get; set; }

33 │

34 │ public Guid? UserCreated { get; set; }

35 │

36 │ public DateTime? CreatedDate { get; set; }

37 │

38 │ public Guid? UserModified { get; set; }

39 │

40 │ public DateTime? ModifiedDate { get; set; }

41 │

42 │ public bool? Deleted { get; set; }

43 │

44 │ public Guid? UserDeleted { get; set; }

45 │

46 │ public DateTime? DeletedDate { get; set; }

47 │ }

────────────────────────────────────────────────────────────

**📄 StockChiTietNhapKho.cs (CS, 71 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class StockChiTietNhapKho

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int? MaNhapKho { get; set; }

11 │

12 │ public int? MaVatTu { get; set; }

13 │

14 │ public int? MaDvt { get; set; }

15 │

16 │ public string? SoLo { get; set; }

17 │

18 │ public DateTime? NgaySanXuat { get; set; }

19 │

20 │ public DateTime? HanSuDung { get; set; }

21 │

22 │ public decimal? SoLuong { get; set; }

23 │

24 │ public decimal? DonGia { get; set; }

25 │

26 │ public decimal? ChietKhauTm { get; set; }

27 │

28 │ public decimal? ChietKhauPer { get; set; }

29 │

30 │ public bool? ChietKhauSauVat { get; set; }

31 │

32 │ public decimal? TienChietKhau { get; set; }

33 │

34 │ public decimal? ThueVat { get; set; }

35 │

36 │ public decimal? TienThueVat { get; set; }

37 │

38 │ public decimal? ThanhTien { get; set; }

39 │

40 │ public decimal? TongThanhTien { get; set; }

41 │

42 │ public decimal? HanMucNo { get; set; }

43 │

44 │ public int? ThoiHanNo { get; set; }

45 │

46 │ public string? GhiChu { get; set; }

47 │

48 │ public Guid? UserCreated { get; set; }

49 │

50 │ public DateTime? CreatedDate { get; set; }

51 │

52 │ public Guid? UserModified { get; set; }

53 │

54 │ public DateTime? ModifiedDate { get; set; }

55 │

56 │ public bool? Deleted { get; set; }

57 │

58 │ public Guid? UserDeleted { get; set; }

59 │

60 │ public DateTime? DeletedDate { get; set; }

61 │

62 │ public int? MaCtnhap { get; set; }

63 │

64 │ public bool? Bh { get; set; }

65 │

66 │ public int? MaHopDong { get; set; }

67 │

68 │ public int? MaChiTietHopDong { get; set; }

69 │

70 │ public int? MaXuatKho { get; set; }

71 │ }

────────────────────────────────────────────────────────────

**📄 StockChiTietTonKho.cs (CS, 31 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class StockChiTietTonKho

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int? MaKiemKe { get; set; }

11 │

12 │ public int? MaHangHoa { get; set; }

13 │

14 │ public decimal? SoLuongTon { get; set; }

15 │

16 │ public string? GhiChu { get; set; }

17 │

18 │ public Guid? UserCreated { get; set; }

19 │

20 │ public DateTime? CreatedDate { get; set; }

21 │

22 │ public Guid? UserModified { get; set; }

23 │

24 │ public DateTime? ModifiedDate { get; set; }

25 │

26 │ public bool? Deleted { get; set; }

27 │

28 │ public Guid? UserDeleted { get; set; }

29 │

30 │ public DateTime? DeletedDate { get; set; }

31 │ }

────────────────────────────────────────────────────────────

**📄 StockChiTietXuatKho.cs (CS, 45 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class StockChiTietXuatKho

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int MaChiNhanhInt { get; set; }

11 │

12 │ public int? MaPhieuXuat { get; set; }

13 │

14 │ public int? MaHangHoa { get; set; }

15 │

16 │ public int? MaDvt { get; set; }

17 │

18 │ public decimal? SoLuong { get; set; }

19 │

20 │ public decimal? GiaVon { get; set; }

21 │

22 │ public string? GhiChu { get; set; }

23 │

24 │ public Guid? UserCreated { get; set; }

25 │

26 │ public DateTime? CreatedDate { get; set; }

27 │

28 │ public Guid? UserModified { get; set; }

29 │

30 │ public DateTime? ModifiedDate { get; set; }

31 │

32 │ public bool? Deleted { get; set; }

33 │

34 │ public Guid? UserDeleted { get; set; }

35 │

36 │ public DateTime? DeletedDate { get; set; }

37 │

38 │ public int? MaCtnhapKho { get; set; }

39 │

40 │ public int? MaVatTu { get; set; }

41 │

42 │ public decimal? SoLuongThucXuat { get; set; }

43 │

44 │ public int? MaXuatKho { get; set; }

45 │ }

────────────────────────────────────────────────────────────

**📄 StockChiTietXuatKhoNew.cs (CS, 37 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class StockChiTietXuatKhoNew

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int MaChiNhanhInt { get; set; }

11 │

12 │ public int? MaPhieuXuat { get; set; }

13 │

14 │ public int? MaHangHoa { get; set; }

15 │

16 │ public int? MaDvt { get; set; }

17 │

18 │ public decimal? SoLuong { get; set; }

19 │

20 │ public decimal? GiaVon { get; set; }

21 │

22 │ public string? GhiChu { get; set; }

23 │

24 │ public Guid? UserCreated { get; set; }

25 │

26 │ public DateTime? CreatedDate { get; set; }

27 │

28 │ public Guid? UserModified { get; set; }

29 │

30 │ public DateTime? ModifiedDate { get; set; }

31 │

32 │ public bool? Deleted { get; set; }

33 │

34 │ public Guid? UserDeleted { get; set; }

35 │

36 │ public DateTime? DeletedDate { get; set; }

37 │ }

────────────────────────────────────────────────────────────

**📄 StockDanhMucKho.cs (CS, 59 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class StockDanhMucKho

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public Guid? MaChiNhanh { get; set; }

11 │

12 │ public string? Code { get; set; }

13 │

14 │ public string? Ten { get; set; }

15 │

16 │ public string? DienGiai { get; set; }

17 │

18 │ public bool? MayBanHangTuDong { get; set; }

19 │

20 │ public bool? BanHang { get; set; }

21 │

22 │ public bool? NhapHang { get; set; }

23 │

24 │ public int? ThoiGianBatDau { get; set; }

25 │

26 │ public int? ThoiGianKetThuc { get; set; }

27 │

28 │ public string? TaiKhoanCo { get; set; }

29 │

30 │ public string? TaiKhoanKho { get; set; }

31 │

32 │ public int? MaNganhVoucher { get; set; }

33 │

34 │ public bool? HienThiTrongWebOrder { get; set; }

35 │

36 │ public string? TenHienThi { get; set; }

37 │

38 │ public bool? Bartender { get; set; }

39 │

40 │ public bool? Status { get; set; }

41 │

42 │ public Guid? UserCreated { get; set; }

43 │

44 │ public DateTime? CreatedDate { get; set; }

45 │

46 │ public Guid? UserModified { get; set; }

47 │

48 │ public DateTime? ModifiedDate { get; set; }

49 │

50 │ public bool? Deleted { get; set; }

51 │

52 │ public Guid? UserDeleted { get; set; }

53 │

54 │ public DateTime? DeletedDate { get; set; }

55 │

56 │ public int? MaLoaiKho { get; set; }

57 │

58 │ public Guid? MaPhongBan { get; set; }

59 │ }

────────────────────────────────────────────────────────────

**📄 StockKiemKe.cs (CS, 43 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class StockKiemKe

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int MaChiNhanhInt { get; set; }

11 │

12 │ public Guid? MaChiNhanh { get; set; }

13 │

14 │ public string? SoKiemKe { get; set; }

15 │

16 │ public int? MaKho { get; set; }

17 │

18 │ public int? Nam { get; set; }

19 │

20 │ public int? Thang { get; set; }

21 │

22 │ public DateTime? NgayLap { get; set; }

23 │

24 │ public DateTime? NgayKiemKe { get; set; }

25 │

26 │ public string? DienGiai { get; set; }

27 │

28 │ public Guid? UserCreated { get; set; }

29 │

30 │ public DateTime? CreatedDate { get; set; }

31 │

32 │ public Guid? UserModified { get; set; }

33 │

34 │ public DateTime? ModifiedDate { get; set; }

35 │

36 │ public bool? Deleted { get; set; }

37 │

38 │ public Guid? UserDeleted { get; set; }

39 │

40 │ public DateTime? DeletedDate { get; set; }

41 │

42 │ public string? NguoiKiemKe { get; set; }

43 │ }

────────────────────────────────────────────────────────────

**📄 StockKiemKeNew.cs (CS, 43 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class StockKiemKeNew

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int MaChiNhanhInt { get; set; }

11 │

12 │ public Guid? MaChiNhanh { get; set; }

13 │

14 │ public string? SoKiemKe { get; set; }

15 │

16 │ public int? MaKho { get; set; }

17 │

18 │ public int? Nam { get; set; }

19 │

20 │ public int? Thang { get; set; }

21 │

22 │ public DateTime? NgayLap { get; set; }

23 │

24 │ public DateTime? NgayKiemKe { get; set; }

25 │

26 │ public string? DienGiai { get; set; }

27 │

28 │ public Guid? UserCreated { get; set; }

29 │

30 │ public DateTime? CreatedDate { get; set; }

31 │

32 │ public Guid? UserModified { get; set; }

33 │

34 │ public DateTime? ModifiedDate { get; set; }

35 │

36 │ public bool? Deleted { get; set; }

37 │

38 │ public Guid? UserDeleted { get; set; }

39 │

40 │ public DateTime? DeletedDate { get; set; }

41 │

42 │ public string? NguoiKiemKe { get; set; }

43 │ }

────────────────────────────────────────────────────────────

**📄 StockLichSuVatTu.cs (CS, 49 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class StockLichSuVatTu

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int MaVatTu { get; set; }

11 │

12 │ public int? MaLoai { get; set; }

13 │

14 │ public int? MaNhom { get; set; }

15 │

16 │ public int? MaHangHoa { get; set; }

17 │

18 │ public string? Code { get; set; }

19 │

20 │ public string? Ten { get; set; }

21 │

22 │ public string? TenVietTat { get; set; }

23 │

24 │ public string? TenBh { get; set; }

25 │

26 │ public string? HamLuong { get; set; }

27 │

28 │ public int? MaDvt { get; set; }

29 │

30 │ public int? MaNhaCc { get; set; }

31 │

32 │ public bool? Status { get; set; }

33 │

34 │ public Guid? UserCreated { get; set; }

35 │

36 │ public DateTime? CreatedDate { get; set; }

37 │

38 │ public Guid? UserModified { get; set; }

39 │

40 │ public DateTime? ModifiedDate { get; set; }

41 │

42 │ public bool? Deleted { get; set; }

43 │

44 │ public Guid? UserDeleted { get; set; }

45 │

46 │ public DateTime? DeletedDate { get; set; }

47 │

48 │ public decimal? GiaNhap { get; set; }

49 │ }

────────────────────────────────────────────────────────────

**📄 StockLoaiKho.cs (CS, 29 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class StockLoaiKho

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string? Ten { get; set; }

11 │

12 │ public string? MoTa { get; set; }

13 │

14 │ public bool? Status { get; set; }

15 │

16 │ public Guid? UserCreated { get; set; }

17 │

18 │ public DateTime? CreatedDate { get; set; }

19 │

20 │ public Guid? UserModified { get; set; }

21 │

22 │ public DateTime? ModifiedDate { get; set; }

23 │

24 │ public bool? Deleted { get; set; }

25 │

26 │ public Guid? UserDeleted { get; set; }

27 │

28 │ public DateTime? DeletedDate { get; set; }

29 │ }

────────────────────────────────────────────────────────────

**📄 StockLoaiNhapKho.cs (CS, 29 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class StockLoaiNhapKho

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string? Code { get; set; }

11 │

12 │ public string? Ten { get; set; }

13 │

14 │ public bool? Status { get; set; }

15 │

16 │ public Guid? UserCreated { get; set; }

17 │

18 │ public DateTime? CreatedDate { get; set; }

19 │

20 │ public Guid? UserModified { get; set; }

21 │

22 │ public DateTime? ModifiedDate { get; set; }

23 │

24 │ public bool? Deleted { get; set; }

25 │

26 │ public Guid? UserDeleted { get; set; }

27 │

28 │ public DateTime? DeletedDate { get; set; }

29 │ }

────────────────────────────────────────────────────────────

**📄 StockLoaiNhapXuat.cs (CS, 34 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class StockLoaiNhapXuat

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string? Code { get; set; }

11 │

12 │ public string? Ten { get; set; }

13 │

14 │ /// <summary>

15 │ /// 1: Nhập kho, 0:Xuất kho

16 │ /// </summary>

17 │ public bool? NhapKho { get; set; }

18 │

19 │ public bool? Status { get; set; }

20 │

21 │ public Guid? UserCreated { get; set; }

22 │

23 │ public DateTime? CreatedDate { get; set; }

24 │

25 │ public Guid? UserModified { get; set; }

26 │

27 │ public DateTime? ModifiedDate { get; set; }

28 │

29 │ public bool? Deleted { get; set; }

30 │

31 │ public Guid? UserDeleted { get; set; }

32 │

33 │ public DateTime? DeletedDate { get; set; }

34 │ }

────────────────────────────────────────────────────────────

**📄 StockNhapHang.cs (CS, 51 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class StockNhapHang

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int MaChiNhanhInt { get; set; }

11 │

12 │ public Guid? MaChiNhanh { get; set; }

13 │

14 │ public int? MaKho { get; set; }

15 │

16 │ public int? MaLoaiNx { get; set; }

17 │

18 │ public int? MaNhaCc { get; set; }

19 │

20 │ public string? SoHoaDon { get; set; }

21 │

22 │ public string? SoHoaDonNcc { get; set; }

23 │

24 │ public DateTime? NgayHoaDon { get; set; }

25 │

26 │ public DateTime? NgayNhapHang { get; set; }

27 │

28 │ public decimal? TienChietKhau { get; set; }

29 │

30 │ public int? ThueVat { get; set; }

31 │

32 │ public string? GhiChu { get; set; }

33 │

34 │ public bool? KhongHoaDon { get; set; }

35 │

36 │ public Guid? UserCreated { get; set; }

37 │

38 │ public DateTime? CreatedDate { get; set; }

39 │

40 │ public Guid? UserModified { get; set; }

41 │

42 │ public DateTime? ModifiedDate { get; set; }

43 │

44 │ public bool? Deleted { get; set; }

45 │

46 │ public Guid? UserDeleted { get; set; }

47 │

48 │ public DateTime? DeletedDate { get; set; }

49 │

50 │ public bool? KhoaSo { get; set; }

51 │ }

────────────────────────────────────────────────────────────

**📄 StockNhapHangNew.cs (CS, 51 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class StockNhapHangNew

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int MaChiNhanhInt { get; set; }

11 │

12 │ public Guid? MaChiNhanh { get; set; }

13 │

14 │ public int? MaKho { get; set; }

15 │

16 │ public int? MaLoaiNx { get; set; }

17 │

18 │ public int? MaNhaCc { get; set; }

19 │

20 │ public string? SoHoaDon { get; set; }

21 │

22 │ public string? SoHoaDonNcc { get; set; }

23 │

24 │ public DateTime? NgayHoaDon { get; set; }

25 │

26 │ public DateTime? NgayNhapHang { get; set; }

27 │

28 │ public decimal? TienChietKhau { get; set; }

29 │

30 │ public int? ThueVat { get; set; }

31 │

32 │ public string? GhiChu { get; set; }

33 │

34 │ public bool? KhongHoaDon { get; set; }

35 │

36 │ public Guid? UserCreated { get; set; }

37 │

38 │ public DateTime? CreatedDate { get; set; }

39 │

40 │ public Guid? UserModified { get; set; }

41 │

42 │ public DateTime? ModifiedDate { get; set; }

43 │

44 │ public bool? Deleted { get; set; }

45 │

46 │ public Guid? UserDeleted { get; set; }

47 │

48 │ public DateTime? DeletedDate { get; set; }

49 │

50 │ public bool? KhoaSo { get; set; }

51 │ }

────────────────────────────────────────────────────────────

**📄 StockNhapKho.cs (CS, 75 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class StockNhapKho

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public Guid? MaPhongBan { get; set; }

11 │

12 │ public int? MaLoaiNhap { get; set; }

13 │

14 │ public string? SoPhieu { get; set; }

15 │

16 │ public string? KyHieuHoaDon { get; set; }

17 │

18 │ public int? MaNhaCc { get; set; }

19 │

20 │ public DateTime? NgayNhapKho { get; set; }

21 │

22 │ public DateTime? NgayXacNhan { get; set; }

23 │

24 │ public DateTime? NgayHetHanTt { get; set; }

25 │

26 │ public int? MaKhoXuat { get; set; }

27 │

28 │ public int? MaKhoNhap { get; set; }

29 │

30 │ public int? MaHopDong { get; set; }

31 │

32 │ public string? DienGiai { get; set; }

33 │

34 │ public decimal? ThanhTien { get; set; }

35 │

36 │ public decimal? TienThueVat { get; set; }

37 │

38 │ public decimal? TienChietKhau { get; set; }

39 │

40 │ public decimal? PhaiTra { get; set; }

41 │

42 │ public bool? DaChietKhau { get; set; }

43 │

44 │ public int? TrangThai { get; set; }

45 │

46 │ public int? SeqSoPhieu { get; set; }

47 │

48 │ public Guid? UserCreated { get; set; }

49 │

50 │ public DateTime? CreatedDate { get; set; }

51 │

52 │ public Guid? UserModified { get; set; }

53 │

54 │ public DateTime? ModifiedDate { get; set; }

55 │

56 │ public bool? Deleted { get; set; }

57 │

58 │ public Guid? UserDeleted { get; set; }

59 │

60 │ public DateTime? DeletedDate { get; set; }

61 │

62 │ public int? MaPhieuNhap { get; set; }

63 │

64 │ public string? SoChungTu { get; set; }

65 │

66 │ public bool? DaTongHopChietKhau { get; set; }

67 │

68 │ public int? MaPhieuTongHopCk { get; set; }

69 │

70 │ public int? MaDonDatHang { get; set; }

71 │

72 │ public Guid? NguoiXacNhan { get; set; }

73 │

74 │ public Guid? MaChiNhanh { get; set; }

75 │ }

────────────────────────────────────────────────────────────

**📄 StockTonKho.cs (CS, 51 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class StockTonKho

7 │ {

8 │ public long Ma { get; set; }

9 │

10 │ public int MaChiNhanhInt { get; set; }

11 │

12 │ public string? NamThang { get; set; }

13 │

14 │ public int? MaKho { get; set; }

15 │

16 │ public int? MaHangHoa { get; set; }

17 │

18 │ public int? MaNhapHang { get; set; }

19 │

20 │ public decimal? GiaVon { get; set; }

21 │

22 │ public decimal? SltonDauKy { get; set; }

23 │

24 │ public decimal? SlnhapTrongKy { get; set; }

25 │

26 │ public decimal? SlxuatKhac { get; set; }

27 │

28 │ public decimal? SlxuatTieuDung { get; set; }

29 │

30 │ public decimal? SlxuatTra { get; set; }

31 │

32 │ public decimal? SlbanTrongKy { get; set; }

33 │

34 │ public decimal? SltonCuoiKy { get; set; }

35 │

36 │ public Guid? UserCreated { get; set; }

37 │

38 │ public DateTime? CreatedDate { get; set; }

39 │

40 │ public Guid? UserModified { get; set; }

41 │

42 │ public DateTime? ModifiedDate { get; set; }

43 │

44 │ public bool? Deleted { get; set; }

45 │

46 │ public Guid? UserDeleted { get; set; }

47 │

48 │ public DateTime? DeletedDate { get; set; }

49 │

50 │ public Guid? MaChiNhanh { get; set; }

51 │ }

────────────────────────────────────────────────────────────

**📄 StockTonKhoBanHang.cs (CS, 33 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class StockTonKhoBanHang

7 │ {

8 │ public long Ma { get; set; }

9 │

10 │ public int MaChiNhanhInt { get; set; }

11 │

12 │ public string NamThang { get; set; } = null!;

13 │

14 │ public int MaKho { get; set; }

15 │

16 │ public int MaHangHoa { get; set; }

17 │

18 │ public decimal? SltonDauKy { get; set; }

19 │

20 │ public decimal? SlnhapTrongKy { get; set; }

21 │

22 │ public decimal? SlxuatKhac { get; set; }

23 │

24 │ public decimal? SlxuatTieuDung { get; set; }

25 │

26 │ public decimal? SlxuatTra { get; set; }

27 │

28 │ public decimal? SlbanTrongKy { get; set; }

29 │

30 │ public decimal? SltonCuoiKy { get; set; }

31 │

32 │ public Guid MaChiNhanh { get; set; }

33 │ }

────────────────────────────────────────────────────────────

**📄 StockTonKhoVatTu.cs (CS, 39 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class StockTonKhoVatTu

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string NamThang { get; set; } = null!;

11 │

12 │ public int? MaKho { get; set; }

13 │

14 │ public int? MaCtnhap { get; set; }

15 │

16 │ public int? MaVatTu { get; set; }

17 │

18 │ public decimal? GiaNhap { get; set; }

19 │

20 │ public decimal? SoLuongDauKy { get; set; }

21 │

22 │ public decimal? SoLuongNhap { get; set; }

23 │

24 │ public decimal? SoLuongXuat { get; set; }

25 │

26 │ public decimal? SoLuongXuatTam { get; set; }

27 │

28 │ public decimal? SoLuongTon { get; set; }

29 │

30 │ public bool? TrangThai { get; set; }

31 │

32 │ public bool? TreoNhapKho { get; set; }

33 │

34 │ public string? GhiChu { get; set; }

35 │

36 │ public DateTime? NgayNhapKho { get; set; }

37 │

38 │ public Guid? MaChiNhanh { get; set; }

39 │ }

────────────────────────────────────────────────────────────

**📄 StockVatTu.cs (CS, 55 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class StockVatTu

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int? MaHangHoa { get; set; }

11 │

12 │ public string? Code { get; set; }

13 │

14 │ public string? MaVach { get; set; }

15 │

16 │ public string? Ten { get; set; }

17 │

18 │ public string? TenVietTat { get; set; }

19 │

20 │ public decimal? GiaNhap { get; set; }

21 │

22 │ public bool? Status { get; set; }

23 │

24 │ public Guid? UserCreated { get; set; }

25 │

26 │ public DateTime? CreatedDate { get; set; }

27 │

28 │ public Guid? UserModified { get; set; }

29 │

30 │ public DateTime? ModifiedDate { get; set; }

31 │

32 │ public bool? Deleted { get; set; }

33 │

34 │ public Guid? UserDeleted { get; set; }

35 │

36 │ public DateTime? DeletedDate { get; set; }

37 │

38 │ public int? MaLoaiVatTu { get; set; }

39 │

40 │ public int? MaDvt { get; set; }

41 │

42 │ public int? MaNhaCc { get; set; }

43 │

44 │ public decimal? ChietKhauTm { get; set; }

45 │

46 │ public decimal? ChietKhauPer { get; set; }

47 │

48 │ public bool? CksauThue { get; set; }

49 │

50 │ public decimal? SoLuongMin { get; set; }

51 │

52 │ public decimal? SoLuongMax { get; set; }

53 │

54 │ public Guid? MaChiNhanh { get; set; }

55 │ }

────────────────────────────────────────────────────────────

**📄 StockXuatKho.cs (CS, 55 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class StockXuatKho

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int MaChiNhanhInt { get; set; }

11 │

12 │ public Guid? MaChiNhanh { get; set; }

13 │

14 │ public string? SoPhieu { get; set; }

15 │

16 │ public DateTime? NgayXuat { get; set; }

17 │

18 │ public int? MaKho { get; set; }

19 │

20 │ public int? MaMayBanHang { get; set; }

21 │

22 │ public int? MaLoaiNx { get; set; }

23 │

24 │ public string? GhiChu { get; set; }

25 │

26 │ public int? MaKhoChuyenDen { get; set; }

27 │

28 │ public bool? KhoaSo { get; set; }

29 │

30 │ public Guid? UserCreated { get; set; }

31 │

32 │ public DateTime? CreatedDate { get; set; }

33 │

34 │ public Guid? UserModified { get; set; }

35 │

36 │ public DateTime? ModifiedDate { get; set; }

37 │

38 │ public bool? Deleted { get; set; }

39 │

40 │ public Guid? UserDeleted { get; set; }

41 │

42 │ public DateTime? DeletedDate { get; set; }

43 │

44 │ public int? MaKhoDeNghi { get; set; }

45 │

46 │ public int? MaKhoXuat { get; set; }

47 │

48 │ public DateTime? NgayLap { get; set; }

49 │

50 │ public DateTime? NgayDuyet { get; set; }

51 │

52 │ public int? TrangThai { get; set; }

53 │

54 │ public int? MaLoaiXuat { get; set; }

55 │ }

────────────────────────────────────────────────────────────

**📄 StockXuatKhoNew.cs (CS, 43 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class StockXuatKhoNew

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int MaChiNhanhInt { get; set; }

11 │

12 │ public Guid? MaChiNhanh { get; set; }

13 │

14 │ public string? SoPhieu { get; set; }

15 │

16 │ public DateTime? NgayXuat { get; set; }

17 │

18 │ public int? MaKho { get; set; }

19 │

20 │ public int? MaMayBanHang { get; set; }

21 │

22 │ public int? MaLoaiNx { get; set; }

23 │

24 │ public string? GhiChu { get; set; }

25 │

26 │ public int? MaKhoChuyenDen { get; set; }

27 │

28 │ public bool? KhoaSo { get; set; }

29 │

30 │ public Guid? UserCreated { get; set; }

31 │

32 │ public DateTime? CreatedDate { get; set; }

33 │

34 │ public Guid? UserModified { get; set; }

35 │

36 │ public DateTime? ModifiedDate { get; set; }

37 │

38 │ public bool? Deleted { get; set; }

39 │

40 │ public Guid? UserDeleted { get; set; }

41 │

42 │ public DateTime? DeletedDate { get; set; }

43 │ }

────────────────────────────────────────────────────────────

**📄 SysBranch.cs (CS, 51 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class SysBranch

7 │ {

8 │ public Guid Ma { get; set; }

9 │

10 │ public string? Code { get; set; }

11 │

12 │ public string? MaSoThue { get; set; }

13 │

14 │ public string? Ten { get; set; }

15 │

16 │ public string? PhapNhan { get; set; }

17 │

18 │ public string? DiaChi { get; set; }

19 │

20 │ public string? Logo { get; set; }

21 │

22 │ public string? SoTaiKhoan { get; set; }

23 │

24 │ public bool? Status { get; set; }

25 │

26 │ public bool? CoSoQuy { get; set; }

27 │

28 │ public DateTime? NgayKhoaSo { get; set; }

29 │

30 │ public Guid? UserCreated { get; set; }

31 │

32 │ public DateTime? CreatedDate { get; set; }

33 │

34 │ public Guid? UserModified { get; set; }

35 │

36 │ public DateTime? ModifiedDate { get; set; }

37 │

38 │ public bool? Deleted { get; set; }

39 │

40 │ public Guid? UserDeleted { get; set; }

41 │

42 │ public DateTime? DeletedDate { get; set; }

43 │

44 │ public string? TenVietTat { get; set; }

45 │

46 │ public int? Nid { get; set; }

47 │

48 │ public int? MaChiNhanhVoucher { get; set; }

49 │

50 │ public bool? CoQuayPhaChe { get; set; }

51 │ }

────────────────────────────────────────────────────────────

**📄 SysDsbaoCao.cs (CS, 23 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class SysDsbaoCao

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string? TenBaoCao { get; set; }

11 │

12 │ public string? DienGiai { get; set; }

13 │

14 │ public bool? TrangThai { get; set; }

15 │

16 │ public int? Loai { get; set; }

17 │

18 │ public bool? DaXoa { get; set; }

19 │

20 │ public int? NguoiXoa { get; set; }

21 │

22 │ public DateTime? NgayXoa { get; set; }

23 │ }

────────────────────────────────────────────────────────────

**📄 SysDsdieuKienBaoCao.cs (CS, 23 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class SysDsdieuKienBaoCao

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int? MaBaoCao { get; set; }

11 │

12 │ public string? CotHienThi { get; set; }

13 │

14 │ public string? LoaiControl { get; set; }

15 │

16 │ public bool? TrangThai { get; set; }

17 │

18 │ public bool? DaXoa { get; set; }

19 │

20 │ public int? NguoiXoa { get; set; }

21 │

22 │ public DateTime? NgayXoa { get; set; }

23 │ }

────────────────────────────────────────────────────────────

**📄 SysLogo.cs (CS, 33 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class SysLogo

7 │ {

8 │ public Guid Ma { get; set; }

9 │

10 │ public Guid? MaChiNhanh { get; set; }

11 │

12 │ public string? LogoName { get; set; }

13 │

14 │ public string? LogoImage { get; set; }

15 │

16 │ public Guid? BranchId { get; set; }

17 │

18 │ public int? Index { get; set; }

19 │

20 │ public int? UserCreated { get; set; }

21 │

22 │ public DateTime? CreatedDate { get; set; }

23 │

24 │ public int? UserModified { get; set; }

25 │

26 │ public DateTime? ModifiedDate { get; set; }

27 │

28 │ public bool? Deleted { get; set; }

29 │

30 │ public int? UserDeleted { get; set; }

31 │

32 │ public DateTime? DeletedDate { get; set; }

33 │ }

────────────────────────────────────────────────────────────

**📄 SysMenu.cs (CS, 49 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class SysMenu

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string Code { get; set; } = null!;

11 │

12 │ public int? MenuCha { get; set; }

13 │

14 │ public string Ten { get; set; } = null!;

15 │

16 │ public string? DuongDan { get; set; }

17 │

18 │ public string? Url { get; set; }

19 │

20 │ public string? Icon { get; set; }

21 │

22 │ public string? IconWeb { get; set; }

23 │

24 │ public int SoThuTu { get; set; }

25 │

26 │ public string? ThamSo { get; set; }

27 │

28 │ public bool? HienMenu { get; set; }

29 │

30 │ public int? MenuCon { get; set; }

31 │

32 │ public bool? LuonLuonHien { get; set; }

33 │

34 │ public bool? TrangThai { get; set; }

35 │

36 │ public int? NguoiTao { get; set; }

37 │

38 │ public DateTime? NgayTao { get; set; }

39 │

40 │ public int? NguoiSua { get; set; }

41 │

42 │ public DateTime? NgaySua { get; set; }

43 │

44 │ public bool? DaXoa { get; set; }

45 │

46 │ public int? NguoiXoa { get; set; }

47 │

48 │ public DateTime? NgayXoa { get; set; }

49 │ }

────────────────────────────────────────────────────────────

**📄 SysMenu.cs (CS, 45 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.VCEntity;

5 │

6 │ public partial class SysMenu

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string Code { get; set; } = null!;

11 │

12 │ public int? MenuCha { get; set; }

13 │

14 │ public string Ten { get; set; } = null!;

15 │

16 │ public string? DuongDan { get; set; }

17 │

18 │ public string? Icon { get; set; }

19 │

20 │ public int SoThuTu { get; set; }

21 │

22 │ public string? ThamSo { get; set; }

23 │

24 │ public bool? HienMenu { get; set; }

25 │

26 │ public int? MenuCon { get; set; }

27 │

28 │ public bool? LuonLuonHien { get; set; }

29 │

30 │ public bool? TrangThai { get; set; }

31 │

32 │ public int? NguoiTao { get; set; }

33 │

34 │ public DateTime? NgayTao { get; set; }

35 │

36 │ public int? NguoiSua { get; set; }

37 │

38 │ public DateTime? NgaySua { get; set; }

39 │

40 │ public bool? DaXoa { get; set; }

41 │

42 │ public int? NguoiXoa { get; set; }

43 │

44 │ public DateTime? NgayXoa { get; set; }

45 │ }

────────────────────────────────────────────────────────────

**📄 SysMenu.cs (CS, 45 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class SysMenu

7 │ {

8 │ public Guid Ma { get; set; }

9 │

10 │ public string? Code { get; set; }

11 │

12 │ public Guid? ParentId { get; set; }

13 │

14 │ public string? Name { get; set; }

15 │

16 │ public int? Sequence { get; set; }

17 │

18 │ public string? Link { get; set; }

19 │

20 │ public string? Icon { get; set; }

21 │

22 │ public string? ClassIcon { get; set; }

23 │

24 │ public Guid? ChildOfMenu { get; set; }

25 │

26 │ public bool? Target { get; set; }

27 │

28 │ public bool? Status { get; set; }

29 │

30 │ public string? UsingFor { get; set; }

31 │

32 │ public Guid? UserCreated { get; set; }

33 │

34 │ public DateTime? CreatedDate { get; set; }

35 │

36 │ public Guid? UserModified { get; set; }

37 │

38 │ public DateTime? ModifiedDate { get; set; }

39 │

40 │ public bool? Deleted { get; set; }

41 │

42 │ public Guid? UserDeleted { get; set; }

43 │

44 │ public DateTime? DeletedDate { get; set; }

45 │ }

────────────────────────────────────────────────────────────

**📄 SysOtp.cs (CS, 17 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class SysOtp

7 │ {

8 │ public Guid Ma { get; set; }

9 │

10 │ public Guid? MaUser { get; set; }

11 │

12 │ public Guid? MaToken { get; set; }

13 │

14 │ public string? Otp { get; set; }

15 │

16 │ public DateTime? CreatedDate { get; set; }

17 │ }

────────────────────────────────────────────────────────────

**📄 SysPermission.cs (CS, 35 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class SysPermission

7 │ {

8 │ public Guid Ma { get; set; }

9 │

10 │ public string Code { get; set; } = null!;

11 │

12 │ public string Name { get; set; } = null!;

13 │

14 │ public string? Description { get; set; }

15 │

16 │ public bool? FormAccess { get; set; }

17 │

18 │ public bool? PhanQuyenKho { get; set; }

19 │

20 │ public bool? Status { get; set; }

21 │

22 │ public Guid? UserCreated { get; set; }

23 │

24 │ public DateTime? CreatedDate { get; set; }

25 │

26 │ public Guid? UserModified { get; set; }

27 │

28 │ public DateTime? ModifiedDate { get; set; }

29 │

30 │ public bool? Deleted { get; set; }

31 │

32 │ public Guid? UserDeleted { get; set; }

33 │

34 │ public DateTime? DeletedDate { get; set; }

35 │ }

────────────────────────────────────────────────────────────

**📄 SysPrinter.cs (CS, 15 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class SysPrinter

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string? TenMayTinh { get; set; }

11 │

12 │ public string? TenMayIn { get; set; }

13 │

14 │ public int? LanIn { get; set; }

15 │ }

────────────────────────────────────────────────────────────

**📄 SysQuyen.cs (CS, 33 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class SysQuyen

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string Code { get; set; } = null!;

11 │

12 │ public string Ten { get; set; } = null!;

13 │

14 │ public string? MoTa { get; set; }

15 │

16 │ public bool? TruyCapForm { get; set; }

17 │

18 │ public bool? TrangThai { get; set; }

19 │

20 │ public int? NguoiTao { get; set; }

21 │

22 │ public DateTime? NgayTao { get; set; }

23 │

24 │ public int? NguoiSua { get; set; }

25 │

26 │ public DateTime? NgaySua { get; set; }

27 │

28 │ public bool? DaXoa { get; set; }

29 │

30 │ public int? NguoiXoa { get; set; }

31 │

32 │ public DateTime? NgayXoa { get; set; }

33 │ }

────────────────────────────────────────────────────────────

**📄 SysQuyen.cs (CS, 33 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.VCEntity;

5 │

6 │ public partial class SysQuyen

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string Code { get; set; } = null!;

11 │

12 │ public string Ten { get; set; } = null!;

13 │

14 │ public string? MoTa { get; set; }

15 │

16 │ public bool? TruyCapForm { get; set; }

17 │

18 │ public bool? TrangThai { get; set; }

19 │

20 │ public int? NguoiTao { get; set; }

21 │

22 │ public DateTime? NgayTao { get; set; }

23 │

24 │ public int? NguoiSua { get; set; }

25 │

26 │ public DateTime? NgaySua { get; set; }

27 │

28 │ public bool? DaXoa { get; set; }

29 │

30 │ public int? NguoiXoa { get; set; }

31 │

32 │ public DateTime? NgayXoa { get; set; }

33 │ }

────────────────────────────────────────────────────────────

**📄 SysRole.cs (CS, 33 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class SysRole

7 │ {

8 │ public Guid Ma { get; set; }

9 │

10 │ public string Code { get; set; } = null!;

11 │

12 │ public string Name { get; set; } = null!;

13 │

14 │ public string? Description { get; set; }

15 │

16 │ public bool? Status { get; set; }

17 │

18 │ public Guid? UserCreated { get; set; }

19 │

20 │ public DateTime? CreatedDate { get; set; }

21 │

22 │ public Guid? UserModified { get; set; }

23 │

24 │ public DateTime? ModifiedDate { get; set; }

25 │

26 │ public bool? Deleted { get; set; }

27 │

28 │ public Guid? UserDeleted { get; set; }

29 │

30 │ public DateTime? DeletedDate { get; set; }

31 │

32 │ public Guid? MaChucVu { get; set; }

33 │ }

────────────────────────────────────────────────────────────

**📄 SysRole.cs (CS, 31 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class SysRole

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string Code { get; set; } = null!;

11 │

12 │ public string Ten { get; set; } = null!;

13 │

14 │ public string? MoTa { get; set; }

15 │

16 │ public bool? TrangThai { get; set; }

17 │

18 │ public int? NguoiTao { get; set; }

19 │

20 │ public DateTime? NgayTao { get; set; }

21 │

22 │ public int? NguoiSua { get; set; }

23 │

24 │ public DateTime? NgaySua { get; set; }

25 │

26 │ public bool? DaXoa { get; set; }

27 │

28 │ public int? NguoiXoa { get; set; }

29 │

30 │ public DateTime? NgayXoa { get; set; }

31 │ }

────────────────────────────────────────────────────────────

**📄 SysRole.cs (CS, 31 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.VCEntity;

5 │

6 │ public partial class SysRole

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string Code { get; set; } = null!;

11 │

12 │ public string Ten { get; set; } = null!;

13 │

14 │ public string? MoTa { get; set; }

15 │

16 │ public bool? TrangThai { get; set; }

17 │

18 │ public int? NguoiTao { get; set; }

19 │

20 │ public DateTime? NgayTao { get; set; }

21 │

22 │ public int? NguoiSua { get; set; }

23 │

24 │ public DateTime? NgaySua { get; set; }

25 │

26 │ public bool? DaXoa { get; set; }

27 │

28 │ public int? NguoiXoa { get; set; }

29 │

30 │ public DateTime? NgayXoa { get; set; }

31 │ }

────────────────────────────────────────────────────────────

**📄 SysRoleMenu.cs (CS, 31 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class SysRoleMenu

7 │ {

8 │ public Guid Ma { get; set; }

9 │

10 │ public Guid RoleId { get; set; }

11 │

12 │ public Guid MenuId { get; set; }

13 │

14 │ public bool? AllowAccess { get; set; }

15 │

16 │ public bool? Status { get; set; }

17 │

18 │ public Guid? UserCreated { get; set; }

19 │

20 │ public DateTime? CreatedDate { get; set; }

21 │

22 │ public Guid? UserModified { get; set; }

23 │

24 │ public DateTime? ModifiedDate { get; set; }

25 │

26 │ public bool? Deleted { get; set; }

27 │

28 │ public Guid? UserDeleted { get; set; }

29 │

30 │ public DateTime? DeletedDate { get; set; }

31 │ }

────────────────────────────────────────────────────────────

**📄 SysRoleMenu.cs (CS, 31 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class SysRoleMenu

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int MaRole { get; set; }

11 │

12 │ public int MaMenu { get; set; }

13 │

14 │ public bool? ChoPhepTruyCap { get; set; }

15 │

16 │ public bool? TrangThai { get; set; }

17 │

18 │ public int? NguoiTao { get; set; }

19 │

20 │ public DateTime? NgayTao { get; set; }

21 │

22 │ public int? NguoiSua { get; set; }

23 │

24 │ public DateTime? NgaySua { get; set; }

25 │

26 │ public bool? DaXoa { get; set; }

27 │

28 │ public int? NguoiXoa { get; set; }

29 │

30 │ public DateTime? NgayXoa { get; set; }

31 │ }

────────────────────────────────────────────────────────────

**📄 SysRoleMenu.cs (CS, 31 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.VCEntity;

5 │

6 │ public partial class SysRoleMenu

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int MaRole { get; set; }

11 │

12 │ public int MaMenu { get; set; }

13 │

14 │ public bool? ChoPhepTruyCap { get; set; }

15 │

16 │ public bool? TrangThai { get; set; }

17 │

18 │ public int? NguoiTao { get; set; }

19 │

20 │ public DateTime? NgayTao { get; set; }

21 │

22 │ public int? NguoiSua { get; set; }

23 │

24 │ public DateTime? NgaySua { get; set; }

25 │

26 │ public bool? DaXoa { get; set; }

27 │

28 │ public int? NguoiXoa { get; set; }

29 │

30 │ public DateTime? NgayXoa { get; set; }

31 │ }

────────────────────────────────────────────────────────────

**📄 SysRolePermission.cs (CS, 29 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class SysRolePermission

7 │ {

8 │ public Guid Ma { get; set; }

9 │

10 │ public Guid RoleMenuId { get; set; }

11 │

12 │ public Guid PermissionId { get; set; }

13 │

14 │ public bool? Status { get; set; }

15 │

16 │ public Guid? UserCreated { get; set; }

17 │

18 │ public DateTime? CreatedDate { get; set; }

19 │

20 │ public Guid? UserModified { get; set; }

21 │

22 │ public DateTime? ModifiedDate { get; set; }

23 │

24 │ public bool? Deleted { get; set; }

25 │

26 │ public Guid? UserDeleted { get; set; }

27 │

28 │ public DateTime? DeletedDate { get; set; }

29 │ }

────────────────────────────────────────────────────────────

**📄 SysRoleQuyen.cs (CS, 31 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class SysRoleQuyen

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int MaRole { get; set; }

11 │

12 │ public int MaMenu { get; set; }

13 │

14 │ public int MaQuyen { get; set; }

15 │

16 │ public bool? TrangThai { get; set; }

17 │

18 │ public int? NguoiTao { get; set; }

19 │

20 │ public DateTime? NgayTao { get; set; }

21 │

22 │ public int? NguoiSua { get; set; }

23 │

24 │ public DateTime? NgaySua { get; set; }

25 │

26 │ public bool? DaXoa { get; set; }

27 │

28 │ public int? NguoiXoa { get; set; }

29 │

30 │ public DateTime? NgayXoa { get; set; }

31 │ }

────────────────────────────────────────────────────────────

**📄 SysRoleQuyen.cs (CS, 31 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.VCEntity;

5 │

6 │ public partial class SysRoleQuyen

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int MaRole { get; set; }

11 │

12 │ public int MaMenu { get; set; }

13 │

14 │ public int MaQuyen { get; set; }

15 │

16 │ public bool? TrangThai { get; set; }

17 │

18 │ public int? NguoiTao { get; set; }

19 │

20 │ public DateTime? NgayTao { get; set; }

21 │

22 │ public int? NguoiSua { get; set; }

23 │

24 │ public DateTime? NgaySua { get; set; }

25 │

26 │ public bool? DaXoa { get; set; }

27 │

28 │ public int? NguoiXoa { get; set; }

29 │

30 │ public DateTime? NgayXoa { get; set; }

31 │ }

────────────────────────────────────────────────────────────

**📄 SysRoleReport.cs (CS, 27 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class SysRoleReport

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public Guid? MaRole { get; set; }

11 │

12 │ public int? MaBaoCao { get; set; }

13 │

14 │ public Guid? UserCreated { get; set; }

15 │

16 │ public DateTime? CreatedDate { get; set; }

17 │

18 │ public Guid? UserModified { get; set; }

19 │

20 │ public DateTime? ModifiedDate { get; set; }

21 │

22 │ public bool? Deleted { get; set; }

23 │

24 │ public Guid? UserDeleted { get; set; }

25 │

26 │ public DateTime? DeletedDate { get; set; }

27 │ }

────────────────────────────────────────────────────────────

**📄 SysStatus.cs (CS, 15 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class SysStatus

7 │ {

8 │ public int Id { get; set; }

9 │

10 │ public string? StatusMes { get; set; }

11 │

12 │ public int? Status { get; set; }

13 │

14 │ public string? Page { get; set; }

15 │ }

────────────────────────────────────────────────────────────

**📄 SysStt.cs (CS, 17 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class SysStt

7 │ {

8 │ public Guid Ma { get; set; }

9 │

10 │ public string? TableName { get; set; }

11 │

12 │ public string? Colname { get; set; }

13 │

14 │ public decimal? TableMax { get; set; }

15 │

16 │ public DateTime? CreateDate { get; set; }

17 │ }

────────────────────────────────────────────────────────────

**📄 SysThongTinShop.cs (CS, 19 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class SysThongTinShop

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public Guid? MaChiNhanh { get; set; }

11 │

12 │ public string? Ten { get; set; }

13 │

14 │ public string? DiaChi { get; set; }

15 │

16 │ public string? DienThoai { get; set; }

17 │

18 │ public byte[]? Logo { get; set; }

19 │ }

────────────────────────────────────────────────────────────

**📄 SysTrustedDevice.cs (CS, 21 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class SysTrustedDevice

7 │ {

8 │ public Guid Ma { get; set; }

9 │

10 │ public Guid? UserId { get; set; }

11 │

12 │ public string? DevicesIdsha256 { get; set; }

13 │

14 │ public string? DeviceName { get; set; }

15 │

16 │ public DateTime? CreatedDate { get; set; }

17 │

18 │ public bool? Deleted { get; set; }

19 │

20 │ public DateTime? DeletedDate { get; set; }

21 │ }

────────────────────────────────────────────────────────────

**📄 SysUser.cs (CS, 43 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.VCEntity;

5 │

6 │ public partial class SysUser

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string TenDangNhap { get; set; } = null!;

11 │

12 │ public string Password { get; set; } = null!;

13 │

14 │ public string HoTen { get; set; } = null!;

15 │

16 │ public bool? TuChoiNhanEmail { get; set; }

17 │

18 │ public bool? GioiTinh { get; set; }

19 │

20 │ public string? Email { get; set; }

21 │

22 │ public byte[]? Photo { get; set; }

23 │

24 │ public bool? TrangThai { get; set; }

25 │

26 │ public int? NguoiTao { get; set; }

27 │

28 │ public DateTime? NgayTao { get; set; }

29 │

30 │ public int? NguoiSua { get; set; }

31 │

32 │ public DateTime? NgaySua { get; set; }

33 │

34 │ public bool? DaXoa { get; set; }

35 │

36 │ public int? NguoiXoa { get; set; }

37 │

38 │ public DateTime? NgayXoa { get; set; }

39 │

40 │ public bool? Crmuser { get; set; }

41 │

42 │ public string? Msnv { get; set; }

43 │ }

────────────────────────────────────────────────────────────

**📄 SysUser.cs (CS, 67 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class SysUser

7 │ {

8 │ public Guid Ma { get; set; }

9 │

10 │ public int NId { get; set; }

11 │

12 │ public Guid? BranchId { get; set; }

13 │

14 │ public string? Username { get; set; }

15 │

16 │ public string? Password { get; set; }

17 │

18 │ public string? FullName { get; set; }

19 │

20 │ public Guid? MaPhongBan { get; set; }

21 │

22 │ public Guid? MaChucVu { get; set; }

23 │

24 │ public DateTime? NgaySinh { get; set; }

25 │

26 │ public string? DiaChi { get; set; }

27 │

28 │ public string? SoDienThoai { get; set; }

29 │

30 │ public int? GioiTinh { get; set; }

31 │

32 │ public string? HinhAnh { get; set; }

33 │

34 │ public string? Msnv { get; set; }

35 │

36 │ public string? KhoHang { get; set; }

37 │

38 │ public DateTime? NgayVaoLam { get; set; }

39 │

40 │ public DateTime? NgayKetThuc { get; set; }

41 │

42 │ public string? Cccd { get; set; }

43 │

44 │ public bool? EmailReceive { get; set; }

45 │

46 │ public string? Fcm { get; set; }

47 │

48 │ public bool? Reminder { get; set; }

49 │

50 │ public bool? Status { get; set; }

51 │

52 │ public bool? LdapLogin { get; set; }

53 │

54 │ public Guid? UserCreated { get; set; }

55 │

56 │ public DateTime? CreatedDate { get; set; }

57 │

58 │ public Guid? UserModified { get; set; }

59 │

60 │ public DateTime? ModifiedDate { get; set; }

61 │

62 │ public bool? Deleted { get; set; }

63 │

64 │ public Guid? UserDeleted { get; set; }

65 │

66 │ public DateTime? DeletedDate { get; set; }

67 │ }

────────────────────────────────────────────────────────────

**📄 SysUserRole.cs (CS, 29 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class SysUserRole

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int MaUser { get; set; }

11 │

12 │ public int MaRole { get; set; }

13 │

14 │ public bool? TrangThai { get; set; }

15 │

16 │ public int? NguoiTao { get; set; }

17 │

18 │ public DateTime? NgayTao { get; set; }

19 │

20 │ public int? NguoiSua { get; set; }

21 │

22 │ public DateTime? NgaySua { get; set; }

23 │

24 │ public bool? DaXoa { get; set; }

25 │

26 │ public int? NguoiXoa { get; set; }

27 │

28 │ public DateTime? NgayXoa { get; set; }

29 │ }

────────────────────────────────────────────────────────────

**📄 SysUserRole.cs (CS, 33 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.VCEntity;

5 │

6 │ public partial class SysUserRole

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int MaUser { get; set; }

11 │

12 │ public int MaRole { get; set; }

13 │

14 │ public int MaChiNhanh { get; set; }

15 │

16 │ public int MaNganh { get; set; }

17 │

18 │ public bool? TrangThai { get; set; }

19 │

20 │ public int? NguoiTao { get; set; }

21 │

22 │ public DateTime? NgayTao { get; set; }

23 │

24 │ public int? NguoiSua { get; set; }

25 │

26 │ public DateTime? NgaySua { get; set; }

27 │

28 │ public bool? DaXoa { get; set; }

29 │

30 │ public int? NguoiXoa { get; set; }

31 │

32 │ public DateTime? NgayXoa { get; set; }

33 │ }

────────────────────────────────────────────────────────────

**📄 SysUserRole.cs (CS, 33 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class SysUserRole

7 │ {

8 │ public Guid Ma { get; set; }

9 │

10 │ public Guid UserId { get; set; }

11 │

12 │ public Guid RoleId { get; set; }

13 │

14 │ public Guid? BranchId { get; set; }

15 │

16 │ public bool? Status { get; set; }

17 │

18 │ public Guid? UserCreated { get; set; }

19 │

20 │ public DateTime? CreatedDate { get; set; }

21 │

22 │ public Guid? UserModified { get; set; }

23 │

24 │ public DateTime? ModifiedDate { get; set; }

25 │

26 │ public bool? Deleted { get; set; }

27 │

28 │ public Guid? UserDeleted { get; set; }

29 │

30 │ public DateTime? DeletedDate { get; set; }

31 │

32 │ public Guid? MaChucVu { get; set; }

33 │ }

────────────────────────────────────────────────────────────

**📄 SysVariable.cs (CS, 13 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class SysVariable

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string? Code { get; set; }

11 │

12 │ public string Value { get; set; } = null!;

13 │ }

────────────────────────────────────────────────────────────

**📄 TayNinhServiceBackgroundWorker.cs (CS, 78 lines)**

1 │ ﻿using QuanLyBanHang.Models.ExModel.OPS;

2 │ using QuanLyBanHang.Models.QLBHEntity;

3 │ using Microsoft.Extensions.DependencyInjection;

4 │ using Microsoft.Extensions.Hosting;

5 │ using Microsoft.Extensions.Logging;

6 │ using System;

7 │ using System.Threading;

8 │ using System.Threading.Tasks;

9 │ using System.Collections.Concurrent;

10 │

11 │ namespace QuanLyBanHang.Services

12 │ {

13 │ public class TayNinhServiceBackgroundWorker : BackgroundService

14 │ {

15 │ private readonly IServiceScopeFactory \_serviceScopeFactory;

16 │ private readonly ILogger<TayNinhServiceBackgroundWorker> \_logger;

17 │ private static readonly ConcurrentQueue<TayNinhServiceJob> \_queue = new();

18 │

19 │ public TayNinhServiceBackgroundWorker(IServiceScopeFactory serviceScopeFactory, ILogger<TayNinhServiceBackground...

20 │ {

21 │ \_serviceScopeFactory = serviceScopeFactory;

22 │ \_logger = logger;

23 │

24 │ }

25 │

26 │ public static void AddJob(TayNinhServiceJob job)

27 │ {

28 │ \_queue.Enqueue(job);

29 │ Console.WriteLine($"[DEBUG] Enqueued job: TayNinhService");

30 │ }

31 │

32 │ public static int QueueLength => \_queue.Count;

33 │ public static List<TayNinhServiceJob> GetQueueSnapshot() => \_queue.ToList(); // dùng ConcurrentQueue nên gọi...

34 │

35 │ protected override async Task ExecuteAsync(CancellationToken stoppingToken)

36 │ {

37 │ \_logger.LogInformation("TayNinhServiceBackgroundWorker started.");

38 │

39 │ while (!stoppingToken.IsCancellationRequested)

40 │ {

41 │ if (\_queue.TryDequeue(out var job))

42 │ {

43 │ try

44 │ {

45 │ using var scope = \_serviceScopeFactory.CreateScope();

46 │ var context = scope.ServiceProvider.GetRequiredService<QuanLyBanHangContext>();

47 │ var tonKhoModel = new TonKhoModel(context);

48 │

49 │ \_logger.LogInformation($"⏳ Bắt đầu job TayNinhService");

50 │ var service = scope.ServiceProvider.GetRequiredService<ViettelIInvoiceService>();

51 │ //foreach(var hoadon in job.ListMaHoaDon)

52 │ //{

53 │ // await service.CreateInvoice(hoadon);

54 │

55 │ //} // có thông tư 32 cho phép sử dụng hóa đơn cũ, nên ko xài này

56 │ \_logger.LogInformation($"✅ Hoàn tất job TayNinhService");

57 │ }

58 │ catch (Exception ex)

59 │ {

60 │ \_logger.LogError(ex, $"❌ Lỗi job TayNinhService");

61 │ }

62 │ }

63 │ else

64 │ {

65 │ await Task.Delay(500, stoppingToken); // Chờ khi không có job

66 │ }

67 │ }

68 │

69 │ \_logger.LogInformation("TayNinhServiceBackgroundWorker is stopping.");

70 │ }

71 │ }

72 │ // Định nghĩa 1 Job tồn kho

73 │ public class TayNinhServiceJob

74 │ {

75 │ public List<int> ListMaHoaDon { get; set; }

76 │

77 │ }

78 │ }

────────────────────────────────────────────────────────────

**📄 TblBanHang.cs (CS, 55 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.VCEntity;

5 │

6 │ public partial class TblBanHang

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int MaChiNhanh { get; set; }

11 │

12 │ public int MaNganh { get; set; }

13 │

14 │ public int MaKh { get; set; }

15 │

16 │ public int MaVoucher { get; set; }

17 │

18 │ public string MaVach { get; set; } = null!;

19 │

20 │ public string? MaThe { get; set; }

21 │

22 │ public int GiaTri { get; set; }

23 │

24 │ public int SoDu { get; set; }

25 │

26 │ public int ThanhTien { get; set; }

27 │

28 │ public DateTime? NgayHieuLuc { get; set; }

29 │

30 │ public DateTime? NgayHetHan { get; set; }

31 │

32 │ public int? NguoiKichHoat { get; set; }

33 │

34 │ public DateTime? NgayKichHoat { get; set; }

35 │

36 │ public bool? TrangThai { get; set; }

37 │

38 │ public int? BanBoSungTu { get; set; }

39 │

40 │ public string? GhiChu { get; set; }

41 │

42 │ public int? NguoiTao { get; set; }

43 │

44 │ public DateTime? NgayTao { get; set; }

45 │

46 │ public int? NguoiSua { get; set; }

47 │

48 │ public DateTime? NgaySua { get; set; }

49 │

50 │ public bool? DaXoa { get; set; }

51 │

52 │ public int? NguoiXoa { get; set; }

53 │

54 │ public DateTime? NgayXoa { get; set; }

55 │ }

────────────────────────────────────────────────────────────

**📄 TblBuongPhong.cs (CS, 33 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class TblBuongPhong

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public Guid? MaChiNhanh { get; set; }

11 │

12 │ public string? Code { get; set; }

13 │

14 │ public string? Ten { get; set; }

15 │

16 │ public bool? NoiBo { get; set; }

17 │

18 │ public bool? Status { get; set; }

19 │

20 │ public Guid? UserCreated { get; set; }

21 │

22 │ public DateTime? CreatedDate { get; set; }

23 │

24 │ public Guid? UserModified { get; set; }

25 │

26 │ public DateTime? ModifiedDate { get; set; }

27 │

28 │ public bool? Deleted { get; set; }

29 │

30 │ public Guid? UserDeleted { get; set; }

31 │

32 │ public DateTime? DeletedDate { get; set; }

33 │ }

────────────────────────────────────────────────────────────

**📄 TblChiNhanh.cs (CS, 29 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.VCEntity;

5 │

6 │ public partial class TblChiNhanh

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string Ten { get; set; } = null!;

11 │

12 │ public byte[]? Logo { get; set; }

13 │

14 │ public bool? TrangThai { get; set; }

15 │

16 │ public int? NguoiTao { get; set; }

17 │

18 │ public DateTime? NgayTao { get; set; }

19 │

20 │ public int? NguoiSua { get; set; }

21 │

22 │ public DateTime? NgaySua { get; set; }

23 │

24 │ public bool? DaXoa { get; set; }

25 │

26 │ public int? NguoiXoa { get; set; }

27 │

28 │ public DateTime? NgayXoa { get; set; }

29 │ }

────────────────────────────────────────────────────────────

**📄 TblChiTietKetCa.cs (CS, 33 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class TblChiTietKetCa

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public DateTime? NgayBan { get; set; }

11 │

12 │ public DateTime? NgayKetCa { get; set; }

13 │

14 │ public Guid? MaUser { get; set; }

15 │

16 │ public int? MaKho { get; set; }

17 │

18 │ public decimal? SoTien { get; set; }

19 │

20 │ public Guid? UserCreated { get; set; }

21 │

22 │ public DateTime? CreatedDate { get; set; }

23 │

24 │ public Guid? UserModified { get; set; }

25 │

26 │ public DateTime? ModifiedDate { get; set; }

27 │

28 │ public bool? Deleted { get; set; }

29 │

30 │ public Guid? UserDeleted { get; set; }

31 │

32 │ public DateTime? DeletedDate { get; set; }

33 │ }

────────────────────────────────────────────────────────────

**📄 TblChiTietLichSuGia.cs (CS, 37 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class TblChiTietLichSuGia

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int? MaLichSu { get; set; }

11 │

12 │ public int? MaHangHoa { get; set; }

13 │

14 │ public string? MaKho { get; set; }

15 │

16 │ public int? GiaBanCu { get; set; }

17 │

18 │ public int? GiaBanMoi { get; set; }

19 │

20 │ public string? GiaBanCuTheoKho { get; set; }

21 │

22 │ public string? GhiChu { get; set; }

23 │

24 │ public Guid? UserCreated { get; set; }

25 │

26 │ public DateTime? CreatedDate { get; set; }

27 │

28 │ public Guid? UserModified { get; set; }

29 │

30 │ public DateTime? ModifiedDate { get; set; }

31 │

32 │ public bool? Deleted { get; set; }

33 │

34 │ public Guid? UserDeleted { get; set; }

35 │

36 │ public DateTime? DeletedDate { get; set; }

37 │ }

────────────────────────────────────────────────────────────

**📄 TblChiTietXuatKho.cs (CS, 35 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class TblChiTietXuatKho

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int? MaPhieuXuat { get; set; }

11 │

12 │ public int? MaHangHoa { get; set; }

13 │

14 │ public int? MaDvt { get; set; }

15 │

16 │ public decimal? SoLuong { get; set; }

17 │

18 │ public decimal? GiaVon { get; set; }

19 │

20 │ public string? GhiChu { get; set; }

21 │

22 │ public int? NguoiTao { get; set; }

23 │

24 │ public DateTime? NgayTao { get; set; }

25 │

26 │ public int? NguoiSua { get; set; }

27 │

28 │ public DateTime? NgaySua { get; set; }

29 │

30 │ public bool? DaXoa { get; set; }

31 │

32 │ public int? NguoiXoa { get; set; }

33 │

34 │ public DateTime? NgayXoa { get; set; }

35 │ }

────────────────────────────────────────────────────────────

**📄 TblChucVu.cs (CS, 31 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class TblChucVu

7 │ {

8 │ public Guid Ma { get; set; }

9 │

10 │ public string? Code { get; set; }

11 │

12 │ public string? Ten { get; set; }

13 │

14 │ public bool? Status { get; set; }

15 │

16 │ public Guid? MaNhomChucVu { get; set; }

17 │

18 │ public Guid? UserCreated { get; set; }

19 │

20 │ public DateTime? CreatedDate { get; set; }

21 │

22 │ public Guid? UserModified { get; set; }

23 │

24 │ public DateTime? ModifiedDate { get; set; }

25 │

26 │ public bool? Deleted { get; set; }

27 │

28 │ public Guid? UserDeleted { get; set; }

29 │

30 │ public DateTime? DeletedDate { get; set; }

31 │ }

────────────────────────────────────────────────────────────

**📄 TblCtKetCa.cs (CS, 19 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class TblCtKetCa

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public DateTime? NgayBan { get; set; }

11 │

12 │ public DateTime? NgayKetCa { get; set; }

13 │

14 │ public int? UserId { get; set; }

15 │

16 │ public int? MaKho { get; set; }

17 │

18 │ public decimal? SoTien { get; set; }

19 │ }

────────────────────────────────────────────────────────────

**📄 TblDanhMucKho.cs (CS, 47 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class TblDanhMucKho

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string? Code { get; set; }

11 │

12 │ public string? Ten { get; set; }

13 │

14 │ public string? DienGiai { get; set; }

15 │

16 │ public bool? MayBanHangTuDong { get; set; }

17 │

18 │ public bool? BanHang { get; set; }

19 │

20 │ public bool? NhapHang { get; set; }

21 │

22 │ public bool? TrangThai { get; set; }

23 │

24 │ public string? TaiKhoanCo { get; set; }

25 │

26 │ public string? TaiKhoanKho { get; set; }

27 │

28 │ public int? MaNganhVoucher { get; set; }

29 │

30 │ public int? NguoiTao { get; set; }

31 │

32 │ public DateTime? NgayTao { get; set; }

33 │

34 │ public int? NguoiSua { get; set; }

35 │

36 │ public DateTime? NgaySua { get; set; }

37 │

38 │ public bool? DaXoa { get; set; }

39 │

40 │ public int? NguoiXoa { get; set; }

41 │

42 │ public DateTime? NgayXoa { get; set; }

43 │

44 │ public int? ThoiGianBatDau { get; set; }

45 │

46 │ public int? ThoiGianKetThuc { get; set; }

47 │ }

────────────────────────────────────────────────────────────

**📄 TblDonViTinh.cs (CS, 27 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class TblDonViTinh

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string? Ten { get; set; }

11 │

12 │ public bool? Status { get; set; }

13 │

14 │ public Guid? UserCreated { get; set; }

15 │

16 │ public DateTime? CreatedDate { get; set; }

17 │

18 │ public Guid? UserModified { get; set; }

19 │

20 │ public DateTime? ModifiedDate { get; set; }

21 │

22 │ public bool? Deleted { get; set; }

23 │

24 │ public Guid? UserDeleted { get; set; }

25 │

26 │ public DateTime? DeletedDate { get; set; }

27 │ }

────────────────────────────────────────────────────────────

**📄 TblGiaBanTheoKho.cs (CS, 33 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class TblGiaBanTheoKho

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int? MaHangHoa { get; set; }

11 │

12 │ public int? MaKho { get; set; }

13 │

14 │ public int? GiaBan { get; set; }

15 │

16 │ public string? GhiChu { get; set; }

17 │

18 │ public DateTime? NgayTao { get; set; }

19 │

20 │ public int? NguoiTao { get; set; }

21 │

22 │ public bool? TrangThai { get; set; }

23 │

24 │ public DateTime? NgaySua { get; set; }

25 │

26 │ public int? NguoiSua { get; set; }

27 │

28 │ public bool? DaXoa { get; set; }

29 │

30 │ public int? NguoiXoa { get; set; }

31 │

32 │ public DateTime? NgayXoa { get; set; }

33 │ }

────────────────────────────────────────────────────────────

**📄 TblGiaBanTheoKho.cs (CS, 37 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class TblGiaBanTheoKho

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int MaChiNhanhInt { get; set; }

11 │

12 │ public Guid? MaChiNhanh { get; set; }

13 │

14 │ public int? MaKho { get; set; }

15 │

16 │ public int? MaHangHoa { get; set; }

17 │

18 │ public int? GiaBan { get; set; }

19 │

20 │ public string? GhiChu { get; set; }

21 │

22 │ public Guid? UserCreated { get; set; }

23 │

24 │ public DateTime? CreatedDate { get; set; }

25 │

26 │ public Guid? UserModified { get; set; }

27 │

28 │ public DateTime? ModifiedDate { get; set; }

29 │

30 │ public bool? Deleted { get; set; }

31 │

32 │ public Guid? UserDeleted { get; set; }

33 │

34 │ public DateTime? DeletedDate { get; set; }

35 │

36 │ public bool? Status { get; set; }

37 │ }

────────────────────────────────────────────────────────────

**📄 TblGiamGia.cs (CS, 37 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class TblGiamGia

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public Guid? MaChiNhanh { get; set; }

11 │

12 │ public int? MaKho { get; set; }

13 │

14 │ public int? Giatri { get; set; }

15 │

16 │ public string? Ten { get; set; }

17 │

18 │ public DateTime? TuNgay { get; set; }

19 │

20 │ public DateTime? DenNgay { get; set; }

21 │

22 │ public bool? Status { get; set; }

23 │

24 │ public Guid? UserCreated { get; set; }

25 │

26 │ public DateTime? CreatedDate { get; set; }

27 │

28 │ public Guid? UserModified { get; set; }

29 │

30 │ public DateTime? ModifiedDate { get; set; }

31 │

32 │ public bool? Deleted { get; set; }

33 │

34 │ public Guid? UserDeleted { get; set; }

35 │

36 │ public DateTime? DeletedDate { get; set; }

37 │ }

────────────────────────────────────────────────────────────

**📄 TblGiuongBenh.cs (CS, 33 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class TblGiuongBenh

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public Guid? MaChiNhanh { get; set; }

11 │

12 │ public string? Code { get; set; }

13 │

14 │ public int? MaBuongPhong { get; set; }

15 │

16 │ public string? Ten { get; set; }

17 │

18 │ public bool? Status { get; set; }

19 │

20 │ public Guid? UserCreated { get; set; }

21 │

22 │ public DateTime? CreatedDate { get; set; }

23 │

24 │ public Guid? UserModified { get; set; }

25 │

26 │ public DateTime? ModifiedDate { get; set; }

27 │

28 │ public bool? Deleted { get; set; }

29 │

30 │ public Guid? UserDeleted { get; set; }

31 │

32 │ public DateTime? DeletedDate { get; set; }

33 │ }

────────────────────────────────────────────────────────────

**📄 TblGiuongBenh.cs (CS, 31 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class TblGiuongBenh

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string? Code { get; set; }

11 │

12 │ public int? MaPhongBenh { get; set; }

13 │

14 │ public string? Ten { get; set; }

15 │

16 │ public bool? TrangThai { get; set; }

17 │

18 │ public int? NguoiTao { get; set; }

19 │

20 │ public DateTime? NgayTao { get; set; }

21 │

22 │ public int? NguoiSua { get; set; }

23 │

24 │ public DateTime? NgaySua { get; set; }

25 │

26 │ public bool? DaXoa { get; set; }

27 │

28 │ public int? NguoiXoa { get; set; }

29 │

30 │ public DateTime? NgayXoa { get; set; }

31 │ }

────────────────────────────────────────────────────────────

**📄 TblHangHoa.cs (CS, 61 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class TblHangHoa

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int MaChiNhanhInt { get; set; }

11 │

12 │ public Guid? MaChiNhanh { get; set; }

13 │

14 │ public string? MaKt { get; set; }

15 │

16 │ public string? MaVach { get; set; }

17 │

18 │ public int? MaLopHang { get; set; }

19 │

20 │ public string? Ten { get; set; }

21 │

22 │ public int? MaDinhMuc { get; set; }

23 │

24 │ public int? MaDvt { get; set; }

25 │

26 │ public decimal? GiaBan { get; set; }

27 │

28 │ public decimal? GiaVon { get; set; }

29 │

30 │ public string? HinhAnh { get; set; }

31 │

32 │ public string? GhiChu { get; set; }

33 │

34 │ public bool? BanNoiBo { get; set; }

35 │

36 │ public bool? HetHang { get; set; }

37 │

38 │ public bool? VatTu { get; set; }

39 │

40 │ public int? ParentId { get; set; }

41 │

42 │ public bool? Status { get; set; }

43 │

44 │ public Guid? UserCreated { get; set; }

45 │

46 │ public DateTime? CreatedDate { get; set; }

47 │

48 │ public Guid? UserModified { get; set; }

49 │

50 │ public DateTime? ModifiedDate { get; set; }

51 │

52 │ public bool? Deleted { get; set; }

53 │

54 │ public Guid? UserDeleted { get; set; }

55 │

56 │ public DateTime? DeletedDate { get; set; }

57 │

58 │ public int? MaLoai { get; set; }

59 │

60 │ public int? MaNhom { get; set; }

61 │ }

────────────────────────────────────────────────────────────

**📄 TblHinhThucThanhToan.cs (CS, 29 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class TblHinhThucThanhToan

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public Guid? MaChiNhanh { get; set; }

11 │

12 │ public string? Ten { get; set; }

13 │

14 │ public bool? Status { get; set; }

15 │

16 │ public Guid? UserCreated { get; set; }

17 │

18 │ public DateTime? CreatedDate { get; set; }

19 │

20 │ public Guid? UserModified { get; set; }

21 │

22 │ public DateTime? ModifiedDate { get; set; }

23 │

24 │ public bool? Deleted { get; set; }

25 │

26 │ public Guid? UserDeleted { get; set; }

27 │

28 │ public DateTime? DeletedDate { get; set; }

29 │ }

────────────────────────────────────────────────────────────

**📄 TblKhachHang.cs (CS, 41 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.VCEntity;

5 │

6 │ public partial class TblKhachHang

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string Ten { get; set; } = null!;

11 │

12 │ public string? Cmnd { get; set; }

13 │

14 │ public string? DienThoai { get; set; }

15 │

16 │ public bool? GioiTinh { get; set; }

17 │

18 │ public string? DiaChi { get; set; }

19 │

20 │ public string? Email { get; set; }

21 │

22 │ public string? Ghichu { get; set; }

23 │

24 │ public byte[]? Hinh { get; set; }

25 │

26 │ public bool? TrangThai { get; set; }

27 │

28 │ public int? NguoiTao { get; set; }

29 │

30 │ public DateTime? NgayTao { get; set; }

31 │

32 │ public int? NguoiSua { get; set; }

33 │

34 │ public DateTime? NgaySua { get; set; }

35 │

36 │ public bool? DaXoa { get; set; }

37 │

38 │ public int? NguoiXoa { get; set; }

39 │

40 │ public DateTime? NgayXoa { get; set; }

41 │ }

────────────────────────────────────────────────────────────

**📄 TblKyKho.cs (CS, 33 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class TblKyKho

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public Guid? MaChiNhanh { get; set; }

11 │

12 │ public int? MaKho { get; set; }

13 │

14 │ public int? Nam { get; set; }

15 │

16 │ public int? Thang { get; set; }

17 │

18 │ public bool? KhoaSo { get; set; }

19 │

20 │ public Guid? UserCreated { get; set; }

21 │

22 │ public DateTime? CreatedDate { get; set; }

23 │

24 │ public Guid? UserModified { get; set; }

25 │

26 │ public DateTime? ModifiedDate { get; set; }

27 │

28 │ public bool? Deleted { get; set; }

29 │

30 │ public Guid? UserDeleted { get; set; }

31 │

32 │ public DateTime? DeletedDate { get; set; }

33 │ }

────────────────────────────────────────────────────────────

**📄 TblKyKho.cs (CS, 31 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class TblKyKho

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int? MaKho { get; set; }

11 │

12 │ public int? Nam { get; set; }

13 │

14 │ public int? Thang { get; set; }

15 │

16 │ public bool? KhoaSo { get; set; }

17 │

18 │ public int? NguoiTao { get; set; }

19 │

20 │ public DateTime? NgayTao { get; set; }

21 │

22 │ public int? NguoiSua { get; set; }

23 │

24 │ public DateTime? NgaySua { get; set; }

25 │

26 │ public bool? DaXoa { get; set; }

27 │

28 │ public int? NguoiXoa { get; set; }

29 │

30 │ public DateTime? NgayXoa { get; set; }

31 │ }

────────────────────────────────────────────────────────────

**📄 TblLichSuChinhGia.cs (CS, 33 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class TblLichSuChinhGia

7 │ {

8 │ public long Ma { get; set; }

9 │

10 │ public int? MaHangHoa { get; set; }

11 │

12 │ public int? MaKho { get; set; }

13 │

14 │ public decimal? GiaCu { get; set; }

15 │

16 │ public decimal? GiaMoi { get; set; }

17 │

18 │ public string? GhiChu { get; set; }

19 │

20 │ public Guid? UserCreated { get; set; }

21 │

22 │ public DateTime? CreatedDate { get; set; }

23 │

24 │ public Guid? UserModified { get; set; }

25 │

26 │ public DateTime? ModifiedDate { get; set; }

27 │

28 │ public bool? Deleted { get; set; }

29 │

30 │ public Guid? UserDeleted { get; set; }

31 │

32 │ public DateTime? DeletedDate { get; set; }

33 │ }

────────────────────────────────────────────────────────────

**📄 TblLichSuGia.cs (CS, 29 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class TblLichSuGia

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string TieuDe { get; set; } = null!;

11 │

12 │ public string LyDo { get; set; } = null!;

13 │

14 │ public bool Status { get; set; }

15 │

16 │ public Guid? UserCreated { get; set; }

17 │

18 │ public DateTime? CreatedDate { get; set; }

19 │

20 │ public Guid? UserModified { get; set; }

21 │

22 │ public DateTime? ModifiedDate { get; set; }

23 │

24 │ public bool? Deleted { get; set; }

25 │

26 │ public Guid? UserDeleted { get; set; }

27 │

28 │ public DateTime? DeletedDate { get; set; }

29 │ }

────────────────────────────────────────────────────────────

**📄 TblLichSuThanhVien.cs (CS, 21 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.VCEntity;

5 │

6 │ public partial class TblLichSuThanhVien

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int? MaThanhVien { get; set; }

11 │

12 │ public int? NguoiXuLy { get; set; }

13 │

14 │ public DateTime? NgayXuLy { get; set; }

15 │

16 │ public string? NoiDung { get; set; }

17 │

18 │ public string? GiaTriCu { get; set; }

19 │

20 │ public string? GiaTriMoi { get; set; }

21 │ }

────────────────────────────────────────────────────────────

**📄 TblLoaiNhapXuat.cs (CS, 34 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class TblLoaiNhapXuat

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string? Code { get; set; }

11 │

12 │ public string? Ten { get; set; }

13 │

14 │ public bool? Status { get; set; }

15 │

16 │ /// <summary>

17 │ /// 1: Nhập kho, 0:Xuất kho

18 │ /// </summary>

19 │ public bool? NhapKho { get; set; }

20 │

21 │ public int? NguoiTao { get; set; }

22 │

23 │ public DateTime? NgayTao { get; set; }

24 │

25 │ public int? NguoiSua { get; set; }

26 │

27 │ public DateTime? NgaySua { get; set; }

28 │

29 │ public bool? DaXoa { get; set; }

30 │

31 │ public int? NguoiXoa { get; set; }

32 │

33 │ public DateTime? NgayXoa { get; set; }

34 │ }

────────────────────────────────────────────────────────────

**📄 TblLoaiTieuDung.cs (CS, 27 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class TblLoaiTieuDung

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string? LoaiTieuDung { get; set; }

11 │

12 │ public string? Status { get; set; }

13 │

14 │ public Guid? UserCreated { get; set; }

15 │

16 │ public DateTime? CreatedDate { get; set; }

17 │

18 │ public Guid? UserModified { get; set; }

19 │

20 │ public DateTime? ModifiedDate { get; set; }

21 │

22 │ public bool? Deleted { get; set; }

23 │

24 │ public Guid? UserDeleted { get; set; }

25 │

26 │ public DateTime? DeletedDate { get; set; }

27 │ }

────────────────────────────────────────────────────────────

**📄 TblLopHangHoa.cs (CS, 49 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class TblLopHangHoa

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public Guid? MaChiNhanh { get; set; }

11 │

12 │ public string? Ten { get; set; }

13 │

14 │ public bool? ChoPhepBanAm { get; set; }

15 │

16 │ public int? ThoiGianBatDau { get; set; }

17 │

18 │ public int? ThoiGianKetThuc { get; set; }

19 │

20 │ public bool? QuanLyQuaNfc { get; set; }

21 │

22 │ public bool? Status { get; set; }

23 │

24 │ public int? MaLoaiTieuDung { get; set; }

25 │

26 │ public string? PrinterName { get; set; }

27 │

28 │ public Guid? UserCreated { get; set; }

29 │

30 │ public DateTime? CreatedDate { get; set; }

31 │

32 │ public Guid? UserModified { get; set; }

33 │

34 │ public DateTime? ModifiedDate { get; set; }

35 │

36 │ public bool? Deleted { get; set; }

37 │

38 │ public Guid? UserDeleted { get; set; }

39 │

40 │ public DateTime? DeletedDate { get; set; }

41 │

42 │ public int? ThueVat { get; set; }

43 │

44 │ public string? MaTaiKhoanThue { get; set; }

45 │

46 │ public string? MaTaiKhoanDoanhThu { get; set; }

47 │

48 │ public bool? GiaChuaThue { get; set; }

49 │ }

────────────────────────────────────────────────────────────

**📄 TblNganh.cs (CS, 29 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.VCEntity;

5 │

6 │ public partial class TblNganh

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int MaChiNhanh { get; set; }

11 │

12 │ public string Ten { get; set; } = null!;

13 │

14 │ public bool? TrangThai { get; set; }

15 │

16 │ public int? NguoiTao { get; set; }

17 │

18 │ public DateTime? NgayTao { get; set; }

19 │

20 │ public int? NguoiSua { get; set; }

21 │

22 │ public DateTime? NgaySua { get; set; }

23 │

24 │ public bool? DaXoa { get; set; }

25 │

26 │ public int? NguoiXoa { get; set; }

27 │

28 │ public DateTime? NgayXoa { get; set; }

29 │ }

────────────────────────────────────────────────────────────

**📄 TblNgayKetCa.cs (CS, 27 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class TblNgayKetCa

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int? MaKho { get; set; }

11 │

12 │ public DateTime? NgayBanHang { get; set; }

13 │

14 │ public int? NguoiTao { get; set; }

15 │

16 │ public DateTime? NgayTao { get; set; }

17 │

18 │ public int? NguoiSua { get; set; }

19 │

20 │ public DateTime? NgaySua { get; set; }

21 │

22 │ public bool? DaXoa { get; set; }

23 │

24 │ public int? NguoiXoa { get; set; }

25 │

26 │ public DateTime? NgayXoa { get; set; }

27 │ }

────────────────────────────────────────────────────────────

**📄 TblNgayKetCa.cs (CS, 29 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class TblNgayKetCa

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public Guid? MaChiNhanh { get; set; }

11 │

12 │ public int? MaKho { get; set; }

13 │

14 │ public DateTime? NgayBanHang { get; set; }

15 │

16 │ public Guid? UserCreated { get; set; }

17 │

18 │ public DateTime? CreatedDate { get; set; }

19 │

20 │ public Guid? UserModified { get; set; }

21 │

22 │ public DateTime? ModifiedDate { get; set; }

23 │

24 │ public bool? Deleted { get; set; }

25 │

26 │ public Guid? UserDeleted { get; set; }

27 │

28 │ public DateTime? DeletedDate { get; set; }

29 │ }

────────────────────────────────────────────────────────────

**📄 TblNhaCc.cs (CS, 45 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class TblNhaCc

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int MaChiNhanhInt { get; set; }

11 │

12 │ public Guid? MaChiNhanh { get; set; }

13 │

14 │ public string? MaKt { get; set; }

15 │

16 │ public string TenNhaCc { get; set; } = null!;

17 │

18 │ public string? DiaChi { get; set; }

19 │

20 │ public string? DienThoai { get; set; }

21 │

22 │ public string? Mst { get; set; }

23 │

24 │ public string? Email { get; set; }

25 │

26 │ public string? NguoiLienHe { get; set; }

27 │

28 │ public string? Cccd { get; set; }

29 │

30 │ public bool? Status { get; set; }

31 │

32 │ public Guid? UserCreated { get; set; }

33 │

34 │ public DateTime? CreatedDate { get; set; }

35 │

36 │ public Guid? UserModified { get; set; }

37 │

38 │ public DateTime? ModifiedDate { get; set; }

39 │

40 │ public bool? Deleted { get; set; }

41 │

42 │ public Guid? UserDeleted { get; set; }

43 │

44 │ public DateTime? DeletedDate { get; set; }

45 │ }

────────────────────────────────────────────────────────────

**📄 TblNhomChucVu.cs (CS, 29 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class TblNhomChucVu

7 │ {

8 │ public Guid Ma { get; set; }

9 │

10 │ public string? Code { get; set; }

11 │

12 │ public string? Ten { get; set; }

13 │

14 │ public bool? Status { get; set; }

15 │

16 │ public Guid? UserCreated { get; set; }

17 │

18 │ public DateTime? CreatedDate { get; set; }

19 │

20 │ public Guid? UserModified { get; set; }

21 │

22 │ public DateTime? ModifiedDate { get; set; }

23 │

24 │ public bool? Deleted { get; set; }

25 │

26 │ public Guid? UserDeleted { get; set; }

27 │

28 │ public DateTime? DeletedDate { get; set; }

29 │ }

────────────────────────────────────────────────────────────

**📄 TblNhomThongBao.cs (CS, 27 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class TblNhomThongBao

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int? MaNhom { get; set; }

11 │

12 │ public Guid? MaUser { get; set; }

13 │

14 │ public Guid? UserCreated { get; set; }

15 │

16 │ public DateTime? CreatedDate { get; set; }

17 │

18 │ public Guid? UserModified { get; set; }

19 │

20 │ public DateTime? ModifiedDate { get; set; }

21 │

22 │ public bool? Deleted { get; set; }

23 │

24 │ public Guid? UserDeleted { get; set; }

25 │

26 │ public DateTime? DeletedDate { get; set; }

27 │ }

────────────────────────────────────────────────────────────

**📄 TblNhomVoucher.cs (CS, 27 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.VCEntity;

5 │

6 │ public partial class TblNhomVoucher

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string Ten { get; set; } = null!;

11 │

12 │ public bool? TrangThai { get; set; }

13 │

14 │ public int? NguoiTao { get; set; }

15 │

16 │ public DateTime? NgayTao { get; set; }

17 │

18 │ public int? NguoiSua { get; set; }

19 │

20 │ public DateTime? NgaySua { get; set; }

21 │

22 │ public bool? DaXoa { get; set; }

23 │

24 │ public int? NguoiXoa { get; set; }

25 │

26 │ public DateTime? NgayXoa { get; set; }

27 │ }

────────────────────────────────────────────────────────────

**📄 TblPhongBan.cs (CS, 37 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class TblPhongBan

7 │ {

8 │ public Guid Ma { get; set; }

9 │

10 │ public Guid? MaChiNhanh { get; set; }

11 │

12 │ public string? Code { get; set; }

13 │

14 │ public string? Ten { get; set; }

15 │

16 │ public Guid? MaBan { get; set; }

17 │

18 │ public bool? CoSoQuy { get; set; }

19 │

20 │ public bool? Ban { get; set; }

21 │

22 │ public bool? Status { get; set; }

23 │

24 │ public Guid? UserCreated { get; set; }

25 │

26 │ public DateTime? CreatedDate { get; set; }

27 │

28 │ public Guid? UserModified { get; set; }

29 │

30 │ public DateTime? ModifiedDate { get; set; }

31 │

32 │ public bool? Deleted { get; set; }

33 │

34 │ public Guid? UserDeleted { get; set; }

35 │

36 │ public DateTime? DeletedDate { get; set; }

37 │ }

────────────────────────────────────────────────────────────

**📄 TblPhongBan.cs (CS, 39 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class TblPhongBan

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string Code { get; set; } = null!;

11 │

12 │ public int? MaCn { get; set; }

13 │

14 │ public bool? KhoTong { get; set; }

15 │

16 │ public string Ten { get; set; } = null!;

17 │

18 │ public string? TenVietTat { get; set; }

19 │

20 │ public string? MoTa { get; set; }

21 │

22 │ public string? GroupZalo { get; set; }

23 │

24 │ public bool? TrangThai { get; set; }

25 │

26 │ public int? UserCreated { get; set; }

27 │

28 │ public DateTime? CreatedDate { get; set; }

29 │

30 │ public int? UserModified { get; set; }

31 │

32 │ public DateTime? ModifiedDate { get; set; }

33 │

34 │ public bool? Deleted { get; set; }

35 │

36 │ public int? UserDeleted { get; set; }

37 │

38 │ public DateTime? DeletedDate { get; set; }

39 │ }

────────────────────────────────────────────────────────────

**📄 TblPhongBenh.cs (CS, 31 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class TblPhongBenh

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string? Code { get; set; }

11 │

12 │ public string? Ten { get; set; }

13 │

14 │ public bool? TrangThai { get; set; }

15 │

16 │ public int? NguoiTao { get; set; }

17 │

18 │ public DateTime? NgayTao { get; set; }

19 │

20 │ public int? NguoiSua { get; set; }

21 │

22 │ public DateTime? NgaySua { get; set; }

23 │

24 │ public bool? DaXoa { get; set; }

25 │

26 │ public int? NguoiXoa { get; set; }

27 │

28 │ public DateTime? NgayXoa { get; set; }

29 │

30 │ public bool? NoiBo { get; set; }

31 │ }

────────────────────────────────────────────────────────────

**📄 TblPhongBenh.cs (CS, 35 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class TblPhongBenh

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string Ten { get; set; } = null!;

11 │

12 │ public bool? Status { get; set; }

13 │

14 │ public bool? NoiBo { get; set; }

15 │

16 │ public Guid? UserCreated { get; set; }

17 │

18 │ public DateTime? CreatedDate { get; set; }

19 │

20 │ public Guid? UserModified { get; set; }

21 │

22 │ public DateTime? ModifiedDate { get; set; }

23 │

24 │ public bool? Deleted { get; set; }

25 │

26 │ public Guid? UserDeleted { get; set; }

27 │

28 │ public DateTime? DeletedDate { get; set; }

29 │

30 │ public string? Code { get; set; }

31 │

32 │ public Guid? MaChiNhanh { get; set; }

33 │

34 │ public int? MaCu { get; set; }

35 │ }

────────────────────────────────────────────────────────────

**📄 TblThanhToan.cs (CS, 39 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.VCEntity;

5 │

6 │ public partial class TblThanhToan

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string Code { get; set; } = null!;

11 │

12 │ public int MaChiNhanh { get; set; }

13 │

14 │ public int MaNganh { get; set; }

15 │

16 │ public int MaBanHang { get; set; }

17 │

18 │ public int GiaTriThanhToan { get; set; }

19 │

20 │ public int? SoDu { get; set; }

21 │

22 │ public int? PhuThu { get; set; }

23 │

24 │ public string? GhiChu { get; set; }

25 │

26 │ public int? NguoiTao { get; set; }

27 │

28 │ public DateTime? NgayTao { get; set; }

29 │

30 │ public int? NguoiSua { get; set; }

31 │

32 │ public DateTime? NgaySua { get; set; }

33 │

34 │ public bool? DaXoa { get; set; }

35 │

36 │ public int? NguoiXoa { get; set; }

37 │

38 │ public DateTime? NgayXoa { get; set; }

39 │ }

────────────────────────────────────────────────────────────

**📄 TblThanhVien.cs (CS, 37 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.VCEntity;

5 │

6 │ public partial class TblThanhVien

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int? MaVoucher { get; set; }

11 │

12 │ public string? MaVach { get; set; }

13 │

14 │ public int? MaKh { get; set; }

15 │

16 │ public int? NguoiDangKy { get; set; }

17 │

18 │ public DateTime? NgayDangKy { get; set; }

19 │

20 │ public bool? DaXoa { get; set; }

21 │

22 │ public int? NguoiXoa { get; set; }

23 │

24 │ public DateTime? NgayXoa { get; set; }

25 │

26 │ public int? NguoiCapNhat { get; set; }

27 │

28 │ public DateTime? NgayCapNhat { get; set; }

29 │

30 │ public bool? TrangThai { get; set; }

31 │

32 │ public DateTime? NgayBatDau { get; set; }

33 │

34 │ public DateTime? NgayKetThuc { get; set; }

35 │

36 │ public int? DiemThuong { get; set; }

37 │ }

────────────────────────────────────────────────────────────

**📄 TblThongBao.cs (CS, 29 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class TblThongBao

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string? TenNhom { get; set; }

11 │

12 │ public Guid? MaChiNhanh { get; set; }

13 │

14 │ public bool? TrangThai { get; set; }

15 │

16 │ public Guid? UserCreated { get; set; }

17 │

18 │ public DateTime? CreatedDate { get; set; }

19 │

20 │ public Guid? UserModified { get; set; }

21 │

22 │ public DateTime? ModifiedDate { get; set; }

23 │

24 │ public bool? Deleted { get; set; }

25 │

26 │ public Guid? UserDeleted { get; set; }

27 │

28 │ public DateTime? DeletedDate { get; set; }

29 │ }

────────────────────────────────────────────────────────────

**📄 TblThueVat.cs (CS, 29 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class TblThueVat

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string? Ten { get; set; }

11 │

12 │ public int? GiaTri { get; set; }

13 │

14 │ public bool? Status { get; set; }

15 │

16 │ public Guid? UserCreated { get; set; }

17 │

18 │ public DateTime? CreatedDate { get; set; }

19 │

20 │ public Guid? UserModified { get; set; }

21 │

22 │ public DateTime? ModifiedDate { get; set; }

23 │

24 │ public bool? Deleted { get; set; }

25 │

26 │ public Guid? UserDeleted { get; set; }

27 │

28 │ public DateTime? DeletedDate { get; set; }

29 │ }

────────────────────────────────────────────────────────────

**📄 TblThueVat.cs (CS, 29 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class TblThueVat

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string? Ten { get; set; }

11 │

12 │ public int? GiaTri { get; set; }

13 │

14 │ public bool? TrangThai { get; set; }

15 │

16 │ public int? NguoiTao { get; set; }

17 │

18 │ public DateTime? NgayTao { get; set; }

19 │

20 │ public int? NguoiSua { get; set; }

21 │

22 │ public DateTime? NgaySua { get; set; }

23 │

24 │ public bool? DaXoa { get; set; }

25 │

26 │ public int? NguoiXoa { get; set; }

27 │

28 │ public DateTime? NgayXoa { get; set; }

29 │ }

────────────────────────────────────────────────────────────

**📄 TblTrangThaiNhapKho.cs (CS, 27 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class TblTrangThaiNhapKho

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string? Ten { get; set; }

11 │

12 │ public bool? Status { get; set; }

13 │

14 │ public Guid? UserCreated { get; set; }

15 │

16 │ public DateTime? CreatedDate { get; set; }

17 │

18 │ public Guid? UserModified { get; set; }

19 │

20 │ public DateTime? ModifiedDate { get; set; }

21 │

22 │ public bool? Deleted { get; set; }

23 │

24 │ public Guid? UserDeleted { get; set; }

25 │

26 │ public DateTime? DeletedDate { get; set; }

27 │ }

────────────────────────────────────────────────────────────

**📄 TblTransTrinm.cs (CS, 23 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class TblTransTrinm

7 │ {

8 │ public int Id { get; set; }

9 │

10 │ public int? No { get; set; }

11 │

12 │ public int? ProductId { get; set; }

13 │

14 │ public string? Pcode { get; set; }

15 │

16 │ public int? TransQty { get; set; }

17 │

18 │ public int? TransCost { get; set; }

19 │

20 │ public int? TransDisCount { get; set; }

21 │

22 │ public int? TransTotal { get; set; }

23 │ }

────────────────────────────────────────────────────────────

**📄 TblVoucher.cs (CS, 41 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.VCEntity;

5 │

6 │ public partial class TblVoucher

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int? MaChiNhanh { get; set; }

11 │

12 │ public string LoaiVoucher { get; set; } = null!;

13 │

14 │ public string Ten { get; set; } = null!;

15 │

16 │ public int? NhomVoucherId { get; set; }

17 │

18 │ public int GiaTri { get; set; }

19 │

20 │ public string LoaiThoiHan { get; set; } = null!;

21 │

22 │ public int GiaTriThoiHan { get; set; }

23 │

24 │ public int DonGia { get; set; }

25 │

26 │ public bool? TrangThai { get; set; }

27 │

28 │ public int? NguoiTao { get; set; }

29 │

30 │ public DateTime? NgayTao { get; set; }

31 │

32 │ public int? NguoiSua { get; set; }

33 │

34 │ public DateTime? NgaySua { get; set; }

35 │

36 │ public bool? DaXoa { get; set; }

37 │

38 │ public int? NguoiXoa { get; set; }

39 │

40 │ public DateTime? NgayXoa { get; set; }

41 │ }

────────────────────────────────────────────────────────────

**📄 TblVoucherRole.cs (CS, 31 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.VCEntity;

5 │

6 │ public partial class TblVoucherRole

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int MaVoucher { get; set; }

11 │

12 │ public int MaChiNhanh { get; set; }

13 │

14 │ public int MaNganh { get; set; }

15 │

16 │ public bool? TrangThai { get; set; }

17 │

18 │ public int? NguoiTao { get; set; }

19 │

20 │ public DateTime? NgayTao { get; set; }

21 │

22 │ public int? NguoiSua { get; set; }

23 │

24 │ public DateTime? NgaySua { get; set; }

25 │

26 │ public bool? DaXoa { get; set; }

27 │

28 │ public int? NguoiXoa { get; set; }

29 │

30 │ public DateTime? NgayXoa { get; set; }

31 │ }

────────────────────────────────────────────────────────────

**📄 TblXuatKho.cs (CS, 39 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class TblXuatKho

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string? SoPhieu { get; set; }

11 │

12 │ public DateTime? NgayXuat { get; set; }

13 │

14 │ public int? MaKho { get; set; }

15 │

16 │ public int? MaMayBanHang { get; set; }

17 │

18 │ public int? MaLoaiNx { get; set; }

19 │

20 │ public string? GhiChu { get; set; }

21 │

22 │ public int? MaKhoChuyenDen { get; set; }

23 │

24 │ public int? NguoiTao { get; set; }

25 │

26 │ public DateTime? NgayTao { get; set; }

27 │

28 │ public int? NguoiSua { get; set; }

29 │

30 │ public DateTime? NgaySua { get; set; }

31 │

32 │ public bool? KhoaSo { get; set; }

33 │

34 │ public bool? DaXoa { get; set; }

35 │

36 │ public int? NguoiXoa { get; set; }

37 │

38 │ public DateTime? NgayXoa { get; set; }

39 │ }

────────────────────────────────────────────────────────────

**📄 TestServiceController.cs (CS, 98 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using Microsoft.EntityFrameworkCore;

3 │ using QuanLyBanHang.Common;

4 │ using QuanLyBanHang.Models.EInvoiceService.Viettel;

5 │ using QuanLyBanHang.Models.QLBHEntity;

6 │ using QuanLyBanHang.Services;

7 │

8 │ namespace QuanLyBanHang.Controllers

9 │ {

10 │ public class TestServiceController : Controller

11 │ {

12 │ private readonly IEInvoiceService \_service;

13 │ private readonly ViettelIInvoiceService \_viettelIInvoiceService;

14 │ private readonly QuanLyBanHangContext \_dbcontext;

15 │ public TestServiceController(QuanLyBanHangContext dbcontext, IEInvoiceService service, ViettelIInvoiceService vi...

16 │ {

17 │ \_service = service;

18 │ \_viettelIInvoiceService = viettelIInvoiceService;

19 │ \_dbcontext = dbcontext;

20 │ }

21 │

22 │ public async Task<IActionResult> api1(string ids)

23 │ {

24 │ try

25 │ {

26 │ var list = ids?.Split(";").Select(i => i.ToInt()).ToList() ?? new List<int>();

27 │ var result = await \_service.ImportAndPublishAssignedNoAsync(list);

28 │ return Json(new

29 │ {

30 │ Mess = result,

31 │ });

32 │ }

33 │ catch (Exception ex)

34 │ {

35 │ return Json(new

36 │ {

37 │ Mess = ex.Message,

38 │ });

39 │ }

40 │

41 │ }

42 │

43 │ public async Task<IActionResult> api2(string fromDate, string toDate)

44 │ {

45 │ try

46 │ {

47 │ var dtpFromDate = fromDate.ToDateTime2() ?? DateTime.Now;

48 │ var dtpToDate = toDate.ToDateTime2() ?? DateTime.Now;

49 │ var result = await \_service.PublishInvByDateAsync(dtpFromDate, dtpToDate);

50 │ return Json(new

51 │ {

52 │ Mess = result,

53 │ });

54 │ }

55 │ catch (Exception ex)

56 │ {

57 │ return Json(new

58 │ {

59 │ Mess = ex.Message,

60 │ });

61 │ }

62 │ }

63 │

64 │ public async Task<IActionResult> api3(string id)

65 │ {

66 │ try

67 │ {

68 │ var hoaDon = \_dbcontext.OpsHoaDon.AsNoTracking().FirstOrDefault(t=>t.Ma == id.ToInt() && !(t.Deleted ?? ...

69 │ if(hoaDon == null)

70 │ {

71 │ return Json(new CreateInvoiceResponse

72 │ {

73 │ Description = "Hóa đơn không tồn tại"

74 │ });

75 │ }

76 │ if (hoaDon.SoHoaDonThue + "" != "")

77 │ {

78 │ return Json(new CreateInvoiceResponse

79 │ {

80 │ Description = "Hóa đơn đã phát hành"

81 │ });

82 │ }

83 │ var result = await \_viettelIInvoiceService.CreateInvoice(id.ToInt());

84 │ return Json(new

85 │ {

86 │ Mess = result,

87 │ });

88 │ }

89 │ catch (Exception ex)

90 │ {

91 │ return Json(new

92 │ {

93 │ Mess = ex.Message,

94 │ });

95 │ }

96 │ }

97 │ }

98 │ }

────────────────────────────────────────────────────────────

**📄 ThanhToanVoucherModel.cs (CS, 239 lines)**

1 │ ﻿using Microsoft.EntityFrameworkCore;

2 │ using QuanLyBanHang.Models.ExModel.StoredProcedureModel;

3 │ using QuanLyBanHang.Models.QLBHEntity;

4 │

5 │ namespace QuanLyBanHang.Models.VCEntity.ExModel

6 │ {

7 │ public class ThanhToanVoucherModel : Base<TblThanhToan>

8 │ {

9 │ public ThanhToanVoucherModel(DbContext dbContext) : base(dbContext)

10 │ {

11 │ }

12 │ QuanLyVoucherContext dbContext

13 │ {

14 │ get { return \_dbContext as QuanLyVoucherContext; }

15 │ set { return; }

16 │ }

17 │ public class VoucherResult

18 │ {

19 │ public int Result { get; set; }

20 │ public sp\_GetVoucherResult? Voucher { get; set; }

21 │ }

22 │

23 │ public int ThemThanhToan(

24 │ string code,

25 │ Guid maChiNhanh,

26 │ int maNganh,

27 │ int maBanHang,

28 │ int giaTriThanhToan,

29 │ int tienThe,

30 │ int phuThu,

31 │ string ghiChu,

32 │ Guid nguoiTao,

33 │ DateTime ngayTao,

34 │ int hoaDon\_Id

35 │ )

36 │ {

37 │ //// Lấy thông tin đơn hàng để kiểm tra số dư

38 │ //var banHang = await dbContext.tbl\_BanHang.Where(bh => bh.Ma == maBanHang).FirstOrDefaultAsync();

39 │

40 │ //if (banHang == null)

41 │ //{

42 │ // return -1; // Hoặc ném một exception nếu không tìm thấy đơn hàng

43 │ //}

44 │

45 │ //int soDu = banHang.SoDu ?? 0; // Sử dụng toán tử null-coalescing để tránh lỗi null

46 │ //int conLai = soDu - tienThe;

47 │

48 │ //if (conLai < 0)

49 │ //{

50 │ // return -1;

51 │ //}

52 │

53 │ //// Tạo đối tượng ThanhToan mới

54 │ //var thanhToanMoi = new tbl\_ThanhToan

55 │ //{

56 │ // Code = code,

57 │ // MaChiNhanh = maChiNhanh,

58 │ // MaNganh = maNganh,

59 │ // MaBanHang = maBanHang,

60 │ // GiaTriThanhToan = tienThe, // Sử dụng TienThe thay vì GiaTriThanhToan theo logic SQL

61 │ // SoDu = conLai,

62 │ // PhuThu = 0, // Giá trị cứng là 0 theo logic SQL

63 │ // GhiChu = ghiChu,

64 │ // NguoiTao = nguoiTao,

65 │ // NgayTao = ngayTao

66 │ //};

67 │

68 │ //// Thêm vào DbSet và lưu thay đổi để lấy ID

69 │ //dbContext.tbl\_ThanhToan.Add(thanhToanMoi);

70 │ //await dbContext.SaveChangesAsync();

71 │

72 │ //int maThanhToan = thanhToanMoi.Ma;

73 │

74 │ //if (maThanhToan > 0)

75 │ //{

76 │ // // Cập nhật thông tin đơn hàng

77 │ // banHang.SoDu = conLai;

78 │ // banHang.NguoiSua = nguoiTao;

79 │ // banHang.NgaySua = ngayTao;

80 │ // dbContext.tbl\_BanHang.Update(banHang);

81 │

82 │ // // Cập nhật thông tin hóa đơn

83 │ // var hoaDon = await dbContext.HoaDonBan.Where(hd => hd.HoaDon\_ID == hoaDon\_Id).FirstOrDefaultAsync();

84 │ // if (hoaDon != null)

85 │ // {

86 │ // hoaDon.MaThanhToan = maThanhToan;

87 │ // hoaDon.NgayThanhToan = ngayTao;

88 │ // dbContext.HoaDonBan.Update(hoaDon);

89 │ // }

90 │

91 │ // await dbContext.SaveChangesAsync();

92 │ // return maThanhToan;

93 │ //}

94 │

95 │ return -1; // Trong trường hợp không thể lấy được ID

96 │ }

97 │ public int SuaThanhToan(

98 │ int ma,

99 │ int maBanHang,

100 │ int giaTriThanhToan,

101 │ int tienThe,

102 │ int phuThu,

103 │ string ghiChu,

104 │ Guid nguoiSua,

105 │ DateTime ngaySua

106 │ )

107 │ {

108 │ //try

109 │ //{

110 │ // // Lấy thông tin đơn hàng để kiểm tra số dư

111 │ // var banHang = await \_context.tbl\_BanHang.Where(bh => bh.Ma == maBanHang).FirstOrDefaultAsync();

112 │

113 │ // if (banHang == null)

114 │ // {

115 │ // return 0; // Hoặc ném một exception nếu không tìm thấy đơn hàng

116 │ // }

117 │

118 │ // int soDuHienTai = banHang.SoDu ?? 0;

119 │ // // Cần tính toán số dư mới dựa trên sự thay đổi của TienThe

120 │ // // Để đơn giản, giả sử chúng ta muốn số dư mới sau khi thanh toán là SoDuHienTai ...

121 │ // int conLai = soDuHienTai - tienThe;

122 │

123 │ // if (conLai < 0)

124 │ // {

125 │ // return -1; // Số dư không đủ

126 │ // }

127 │

128 │ // // Lấy thông tin thanh toán cần sửa

129 │ // var thanhToanCanSua = await \_context.tbl\_ThanhToan.Where(tt => tt.Ma == ma).FirstOrDefaultAsync();

130 │

131 │ // if (thanhToanCanSua == null)

132 │ // {

133 │ // return 0; // Không tìm thấy thanh toán để sửa

134 │ // }

135 │

136 │ // // Cập nhật thông tin thanh toán

137 │ // thanhToanCanSua.GiaTriThanhToan = tienThe; // Sử dụng TienThe theo logic SQL

138 │ // thanhToanCanSua.SoDu = conLai;

139 │ // thanhToanCanSua.GhiChu = ghiChu;

140 │ // thanhToanCanSua.NguoiSua = nguoiSua;

141 │ // thanhToanCanSua.NgaySua = ngaySua;

142 │

143 │ // \_context.tbl\_ThanhToan.Update(thanhToanCanSua);

144 │

145 │ // // Cập nhật thông tin đơn hàng

146 │ // banHang.SoDu = conLai;

147 │ // banHang.NguoiSua = nguoiSua;

148 │ // banHang.NgaySua = ngaySua;

149 │

150 │ // \_context.tbl\_BanHang.Update(banHang);

151 │

152 │ // int affectedRows = await \_context.SaveChangesAsync();

153 │

154 │ // return affectedRows > 0 ? 1 : 0; // Trả về 1 nếu có bản ghi nào được cập nhật, 0 n��...

155 │ //}

156 │ //catch (Exception)

157 │ //{

158 │ // // Ghi log lỗi hoặc thực hiện xử lý lỗi khác

159 │ // return 0; // Trả về 0 trong trường hợp có lỗi

160 │ //}

161 │ return -1;

162 │ }

163 │ public VoucherResult GetVoucher(string maVach, int maChiNhanhApDung, int maNganhApDung)

164 │ {

165 │ var result = new VoucherResult { Result = 0 };

166 │

167 │ var allVouchers = (

168 │ from a in dbContext.TblBanHang

169 │ join b in dbContext.TblVoucherRole on a.MaVoucher equals b.MaVoucher

170 │ where a.MaVach == maVach

171 │ && (a.DaXoa ?? false) == false

172 │ && (b.DaXoa ?? false) == false

173 │ select new sp\_GetVoucherResult

174 │ {

175 │ Ma = a.Ma,

176 │ MaVoucher = a.MaVoucher,

177 │ MaVach = a.MaVach,

178 │ GiaTri = a.GiaTri,

179 │ SoDu = a.SoDu,

180 │ ThanhTien = a.ThanhTien,

181 │ NgayHieuLuc = a.NgayHieuLuc,

182 │ NgayHetHan = a.NgayHetHan,

183 │ TrangThai = a.TrangThai,

184 │ MaChiNhanhApDung = b.MaChiNhanh,

185 │ MaNganhApDung = b.MaNganh

186 │ }).ToList();

187 │

188 │ if (!allVouchers.Any())

189 │ {

190 │ result.Result = 0;

191 │ return result;

192 │ }

193 │

194 │ var chiNhanhMatch = allVouchers

195 │ .Where(x => x.MaChiNhanhApDung == maChiNhanhApDung)

196 │ .ToList();

197 │

198 │ if (!chiNhanhMatch.Any())

199 │ {

200 │ result.Result = -1;

201 │ return result;

202 │ }

203 │

204 │ var voucherList = chiNhanhMatch

205 │ .Where(x => x.MaNganhApDung == maNganhApDung)

206 │ .ToList();

207 │

208 │ if (!voucherList.Any())

209 │ {

210 │ result.Result = -2;

211 │ return result;

212 │ }

213 │

214 │ var voucher = voucherList.FirstOrDefault();

215 │ var now = DateTime.Now;

216 │

217 │ if ((voucher?.TrangThai ?? false) == false)

218 │ {

219 │ result.Result = -3;

220 │ }

221 │ else if ((voucher?.NgayHieuLuc ?? now) > now)

222 │ {

223 │ result.Result = -4;

224 │ }

225 │ else if ((voucher?.NgayHetHan ?? now) < now)

226 │ {

227 │ result.Result = -5;

228 │ }

229 │ else

230 │ {

231 │ result.Result = 1;

232 │ }

233 │

234 │ result.Voucher = voucher;

235 │ return result;

236 │ }

237 │

238 │ }

239 │ }

────────────────────────────────────────────────────────────

**📄 ThongTinShop.cs (CS, 17 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class ThongTinShop

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string? Ten { get; set; }

11 │

12 │ public string? DiaChi { get; set; }

13 │

14 │ public string? DienThoai { get; set; }

15 │

16 │ public byte[]? Logo { get; set; }

17 │ }

────────────────────────────────────────────────────────────

**📄 ThueVATModel.cs (CS, 22 lines)**

1 │ ﻿using Microsoft.EntityFrameworkCore;

2 │ using QuanLyBanHang.Models.QLBHEntity;

3 │

4 │ namespace QuanLyBanHang.Models.ExModel.CAT

5 │ {

6 │ public class ThueVATModel : Base<TblThueVat>

7 │ {

8 │ public ThueVATModel(DbContext dbContext) : base(dbContext)

9 │ {

10 │ }

11 │ QuanLyBanHangContext dbContext

12 │ {

13 │ get { return \_dbContext as QuanLyBanHangContext; }

14 │ set { return; }

15 │ }

16 │

17 │ public List<TblThueVat> GetData()

18 │ {

19 │ return Get(t => !(t.Deleted ?? false) && (t.Status??false)).OrderBy(t => t.GiaTri).ToList();

20 │ }

21 │ }

22 │ }

────────────────────────────────────────────────────────────

**📄 ThueVatController.cs (CS, 180 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using QuanLyBanHang.Common;

3 │ using QuanLyBanHang.Common.Attributes;

4 │ using QuanLyBanHang.Models.APIModels;

5 │ using QuanLyBanHang.Models.QLBHEntity;

6 │

7 │ namespace QuanLyBanHang.Controllers.CAT

8 │ {

9 │ public class ThueVatController : BaseController

10 │ {

11 │ private readonly ILogger<ThueVatController> \_logger;

12 │

13 │ public ThueVatController(

14 │ QuanLyBanHangContext dbcontext,

15 │ MemoryCacheManager memoryCache,

16 │ ILogger<ThueVatController> logger) : base(dbcontext, memoryCache)

17 │ {

18 │ m\_MenuID = MenuID.catThueVat.GetEnumGuid();

19 │ \_logger = logger;

20 │ }

21 │

22 │ [TokenUserCheckHTTP]

23 │ public IActionResult Index()

24 │ {

25 │ \_logger.LogInformation("Người dùng {UserId} truy cập trang danh sách Thuế VAT.", objToken.userid);

26 │ return View();

27 │ }

28 │

29 │ [TokenUserCheckHTTP(Permission.Add)]

30 │ public ActionResult Add()

31 │ {

32 │ \_logger.LogInformation("Người dùng {UserId} truy cập trang thêm mới Thuế VAT.", objToken.userid);

33 │ return View("Form");

34 │ }

35 │

36 │ [TokenUserCheckHTTP(Permission.Edit)]

37 │ public ActionResult Edit(string Ma)

38 │ {

39 │ \_logger.LogInformation("Người dùng {UserId} yêu cầu chỉnh sửa Thuế VAT. Mã (đã mã hóa): {M...

40 │ try

41 │ {

42 │ Ma = Ma.Decrypt();

43 │ int id = Ma.ToInt();

44 │ ViewData["item"] = dbcontext.TblThueVat.FirstOrDefault(t => t.Ma == id && !(t.Deleted ?? false));

45 │ return View("Form");

46 │ }

47 │ catch (Exception ex)

48 │ {

49 │ \_logger.LogError(ex, "Lỗi khi tải dữ liệu để chỉnh sửa Thuế VAT. Mã (đã mã hóa): {...

50 │ return RedirectToAction("Index");

51 │ }

52 │ }

53 │

54 │ [HttpPost]

55 │ [TokenUserCheckHTTP(Permission.Edit)] // Gộp cả Add và Edit

56 │ public ActionResult UpdateAPI(string Ma, string Ten, string TrangThai, string GiaTri)

57 │ {

58 │ \_logger.LogInformation("Người dùng {UserId} bắt đầu xử lý lưu (Thêm/Sửa) Thuế VAT. Mã (đ...

59 │ try

60 │ {

61 │ Ma = Ma.Decrypt();

62 │ var result = CheckRequire(Ten, TrangThai, GiaTri);

63 │ if (result != null)

64 │ return Json(result);

65 │

66 │ if (Ma.Equals("0"))

67 │ {

68 │ \_logger.LogInformation("Thêm mới Thuế VAT: {TenThueVat}", Ten);

69 │ var item = new TblThueVat

70 │ {

71 │ Ten = Ten,

72 │ Status = TrangThai == "1",

73 │ GiaTri = GiaTri.ToInt(),

74 │ UserCreated = objToken.userid,

75 │ CreatedDate = DateTime.Now,

76 │ };

77 │ dbcontext.TblThueVat.Add(item);

78 │ dbcontext.SaveChanges();

79 │ \_logger.LogInformation("Thêm mới Thuế VAT ID {ThueVatId} thành công.", item.Ma);

80 │ return Json(QLBHConst.ResponseAddSuccessfully);

81 │ }

82 │ else

83 │ {

84 │ int id = Ma.ToInt();

85 │ \_logger.LogInformation("Cập nhật Thuế VAT ID: {ThueVatId}", id);

86 │ var item = dbcontext.TblThueVat.FirstOrDefault(x => x.Ma == id && !(x.Deleted ?? false));

87 │ if (item == null)

88 │ {

89 │ \_logger.LogWarning("Không tìm thấy Thuế VAT ID: {ThueVatId} để cập nhật.", id);

90 │ return Json(QLBHConst.ResponseNotExist);

91 │ }

92 │

93 │ item.Ten = Ten;

94 │ item.Status = TrangThai == "1";

95 │ item.GiaTri = GiaTri.ToInt();

96 │ item.UserModified = objToken.userid;

97 │ item.ModifiedDate = DateTime.Now;

98 │

99 │ dbcontext.SaveChanges();

100 │ \_logger.LogInformation("Cập nhật Thuế VAT ID {ThueVatId} thành công.", id);

101 │ return Json(QLBHConst.ResponseUpdateSuccessfully);

102 │ }

103 │ }

104 │ catch (Exception ex)

105 │ {

106 │ \_logger.LogError(ex, "Lỗi khi xử lý lưu Thuế VAT. Mã (đã mã hóa): {MaThueVat}", Ma);

107 │ return Json(new ResponseModel(QLBHConst.Error, "Lỗi hệ thống khi lưu."));

108 │ }

109 │ }

110 │

111 │ [HttpPost]

112 │ [TokenUserCheckHTTP(Permission.Delete)]

113 │ public IActionResult DeleteAPI(string Ma)

114 │ {

115 │ \_logger.LogInformation("Người dùng {UserId} yêu cầu xóa Thuế VAT. Mã (đã mã hóa): {MaThueVat}...

116 │ try

117 │ {

118 │ Ma = Ma.Decrypt();

119 │ int id = Ma.ToInt();

120 │ var item = dbcontext.TblThueVat.FirstOrDefault(x => x.Ma == id);

121 │ if (item == null)

122 │ {

123 │ \_logger.LogWarning("Không tìm thấy Thuế VAT ID: {ThueVatId} để xóa.", id);

124 │ return Json(QLBHConst.ResponseNotExist);

125 │ }

126 │

127 │ item.Deleted = true;

128 │ item.UserDeleted = objToken.userid;

129 │ item.DeletedDate = DateTime.Now;

130 │ dbcontext.SaveChanges();

131 │

132 │ \_logger.LogInformation("Đã xóa (đánh dấu) Thuế VAT ID {ThueVatId} thành công.", id);

133 │ return Json(QLBHConst.ResponseDeleteSuccessfully);

134 │ }

135 │ catch (Exception ex)

136 │ {

137 │ \_logger.LogError(ex, "Lỗi khi xóa Thuế VAT. Mã (đã mã hóa): {MaThueVat}", Ma);

138 │ return Json(new ResponseModel(QLBHConst.Error, "Lỗi hệ thống khi xóa."));

139 │ }

140 │ }

141 │

142 │ private static ResponseModel CheckRequire(String Ten, String Status, String GiaTri)

143 │ {

144 │ if (Ten.IsEmpty())

145 │ return new ResponseModel(QLBHConst.Error, "Vui lòng nhập tên");

146 │ if (Status.IsEmpty())

147 │ return new ResponseModel(QLBHConst.Error, "Vui lòng nhập trạng thái");

148 │ if (GiaTri.IsEmpty())

149 │ return new ResponseModel(QLBHConst.Error, "Vui lòng nhập giá trị");

150 │ return null;

151 │ }

152 │

153 │ [TokenUserCheckHTTP(Permission.NotCheck)]

154 │ public IActionResult getListData()

155 │ {

156 │ \_logger.LogInformation("Người dùng {UserId} yêu cầu lấy danh sách Thuế VAT.", objToken.userid);

157 │ try

158 │ {

159 │ var listData = dbcontext.TblThueVat.Where(x => !(x.Deleted ?? false))

160 │ .Select(n => new

161 │ {

162 │ maEncrypt = n.Ma.Encrypt(),

163 │ Ma = n.Ma,

164 │ Ten = n.Ten,

165 │ GiaTri = n.GiaTri,

166 │ trangthai = n.Status ?? false,

167 │ tentrangthai = n.Status ?? false ? "Hoạt động" : "Hết hoạt động",

168 │ }).OrderBy(t => t.Ma).OrderBy(t => t.GiaTri).ToList();

169 │

170 │ \_logger.LogInformation("Lấy thành công {Count} bản ghi Thuế VAT.", listData.Count);

171 │ return Json(new ResponseModel(QLBHConst.OK, "", listData));

172 │ }

173 │ catch (Exception ex)

174 │ {

175 │ \_logger.LogError(ex, "Lỗi khi lấy danh sách Thuế VAT.");

176 │ return Json(new ResponseModel(QLBHConst.Error, "Lỗi hệ thống khi lấy danh sách."));

177 │ }

178 │ }

179 │ }

180 │ }

────────────────────────────────────────────────────────────

**📄 TkCf.cs (CS, 27 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class TkCf

7 │ {

8 │ public double? Stt { get; set; }

9 │

10 │ public string? MãVạch { get; set; }

11 │

12 │ public string? MãHàngHoá { get; set; }

13 │

14 │ public string? TênHàngHoá { get; set; }

15 │

16 │ public string? Đvt { get; set; }

17 │

18 │ public double? GiáVốnChưaVat { get; set; }

19 │

20 │ public double? SlTồn { get; set; }

21 │

22 │ public double? SltonTt { get; set; }

23 │

24 │ public string? ThuếVat { get; set; }

25 │

26 │ public string? HạnSửDụngDdMmYyyy { get; set; }

27 │ }

────────────────────────────────────────────────────────────

**📄 TkChti.cs (CS, 25 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class TkChti

7 │ {

8 │ public double? Stt { get; set; }

9 │

10 │ public string? MãVạch { get; set; }

11 │

12 │ public string? MãHàngHoá { get; set; }

13 │

14 │ public string? TênHàngHoá { get; set; }

15 │

16 │ public string? Đvt { get; set; }

17 │

18 │ public double? GiáVốnChưaVat { get; set; }

19 │

20 │ public double? SlTồn { get; set; }

21 │

22 │ public double? ThuếVat { get; set; }

23 │

24 │ public string? HạnSửDụngDdMmYyyy { get; set; }

25 │ }

────────────────────────────────────────────────────────────

**📄 TkMbhHh.cs (CS, 25 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class TkMbhHh

7 │ {

8 │ public double? Stt { get; set; }

9 │

10 │ public string? MãVạch { get; set; }

11 │

12 │ public string? MãHàngHoá { get; set; }

13 │

14 │ public string? TênHàngHoá { get; set; }

15 │

16 │ public string? Đvt { get; set; }

17 │

18 │ public double? GiáVốnChưaVat { get; set; }

19 │

20 │ public double? SlTồn { get; set; }

21 │

22 │ public string? ThuếVat { get; set; }

23 │

24 │ public string? HạnSửDụngDdMmYyyy { get; set; }

25 │ }

────────────────────────────────────────────────────────────

**📄 TkVtthCafe.cs (CS, 25 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class TkVtthCafe

7 │ {

8 │ public string? MãVạch { get; set; }

9 │

10 │ public string? MãHàngHoá { get; set; }

11 │

12 │ public string? TênHàngHoá { get; set; }

13 │

14 │ public string? Đvt { get; set; }

15 │

16 │ public double? GiáVốnChưaVat { get; set; }

17 │

18 │ public double? SlTồn { get; set; }

19 │

20 │ public string? ThuếVat { get; set; }

21 │

22 │ public string? HạnSửDụngDdMmYyyy { get; set; }

23 │

24 │ public string? F9 { get; set; }

25 │ }

────────────────────────────────────────────────────────────

**📄 TkVtthChti.cs (CS, 25 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class TkVtthChti

7 │ {

8 │ public double? Stt { get; set; }

9 │

10 │ public string? MãVạch { get; set; }

11 │

12 │ public string? MãHàngHoá { get; set; }

13 │

14 │ public string? TênHàngHoá { get; set; }

15 │

16 │ public string? Đvt { get; set; }

17 │

18 │ public double? GiáVốnChưaVat { get; set; }

19 │

20 │ public double? SlTồn { get; set; }

21 │

22 │ public string? ThuếVat { get; set; }

23 │

24 │ public string? HạnSửDụngDdMmYyyy { get; set; }

25 │ }

────────────────────────────────────────────────────────────

**📄 TmpOrder.cs (CS, 21 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class TmpOrder

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int? MaPhong { get; set; }

11 │

12 │ public string? CodePhong { get; set; }

13 │

14 │ public string? TenKh { get; set; }

15 │

16 │ public string? Uuid { get; set; }

17 │

18 │ public string? Order { get; set; }

19 │

20 │ public string? SoDienThoai { get; set; }

21 │ }

────────────────────────────────────────────────────────────

**📄 TmpOrderDetail.cs (CS, 25 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class TmpOrderDetail

7 │ {

8 │ public string Id { get; set; } = null!;

9 │

10 │ public int? MaHoaDon { get; set; }

11 │

12 │ public int? Status { get; set; }

13 │

14 │ public string? Note { get; set; }

15 │

16 │ public DateTime? OrderTime { get; set; }

17 │

18 │ public DateTime? ConfirmedTime { get; set; }

19 │

20 │ public int? MaKho { get; set; }

21 │

22 │ public int? MaNganh { get; set; }

23 │

24 │ public string? Danhsachmon { get; set; }

25 │ }

────────────────────────────────────────────────────────────

**📄 TmpUpdateTonKho.cs (CS, 21 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class TmpUpdateTonKho

7 │ {

8 │ public string? LopHang { get; set; }

9 │

10 │ public string? MaVach { get; set; }

11 │

12 │ public string? TenHangHoa { get; set; }

13 │

14 │ public string? Dvt { get; set; }

15 │

16 │ public int? GiaVon { get; set; }

17 │

18 │ public int? GiaBan { get; set; }

19 │

20 │ public int? SoLuong { get; set; }

21 │ }

────────────────────────────────────────────────────────────

**📄 TonDauKy1.cs (CS, 37 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class TonDauKy1

7 │ {

8 │ public double? Stt { get; set; }

9 │

10 │ public double? Mã { get; set; }

11 │

12 │ public string? MãBv { get; set; }

13 │

14 │ public string? MãCũ { get; set; }

15 │

16 │ public string? TênDượcVậtTư { get; set; }

17 │

18 │ public string? TênHoạtChất { get; set; }

19 │

20 │ public string? HàmLượng { get; set; }

21 │

22 │ public string? Dvt { get; set; }

23 │

24 │ public string? SốLô { get; set; }

25 │

26 │ public DateTime? NgàyHh { get; set; }

27 │

28 │ public double? ĐơnGiá { get; set; }

29 │

30 │ public double? TồnĐk { get; set; }

31 │

32 │ public double? TiềnTồnĐk { get; set; }

33 │

34 │ public double? TồnCk { get; set; }

35 │

36 │ public double? TiềnTồnCk { get; set; }

37 │ }

────────────────────────────────────────────────────────────

**📄 TonDayKy2.cs (CS, 37 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class TonDayKy2

7 │ {

8 │ public double? Stt { get; set; }

9 │

10 │ public double? Mã { get; set; }

11 │

12 │ public string? MãBv { get; set; }

13 │

14 │ public string? MãCũ { get; set; }

15 │

16 │ public string? TênDượcVậtTư { get; set; }

17 │

18 │ public string? TênHoạtChất { get; set; }

19 │

20 │ public string? HàmLượng { get; set; }

21 │

22 │ public string? Dvt { get; set; }

23 │

24 │ public string? SốLô { get; set; }

25 │

26 │ public DateTime? NgàyHh { get; set; }

27 │

28 │ public double? ĐơnGiá { get; set; }

29 │

30 │ public double? TồnĐk { get; set; }

31 │

32 │ public double? TiềnTồnĐk { get; set; }

33 │

34 │ public double? TồnCk { get; set; }

35 │

36 │ public double? TiềnTồnCk { get; set; }

37 │ }

────────────────────────────────────────────────────────────

**📄 TonKho.cs (CS, 39 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class TonKho

7 │ {

8 │ public string? NamThang { get; set; }

9 │

10 │ public int? MaKho { get; set; }

11 │

12 │ public int? MaHangHoa { get; set; }

13 │

14 │ public int? MaNhapHang { get; set; }

15 │

16 │ public decimal? GiaVon { get; set; }

17 │

18 │ public decimal? SltonDauKy { get; set; }

19 │

20 │ public decimal? SlnhapTrongKy { get; set; }

21 │

22 │ public decimal? SlxuatKhac { get; set; }

23 │

24 │ public decimal? SlxuatTieuDung { get; set; }

25 │

26 │ public decimal? SlxuatTra { get; set; }

27 │

28 │ public decimal? SlbanTrongKy { get; set; }

29 │

30 │ public decimal? SltonCuoiKy { get; set; }

31 │

32 │ public bool? DaXoa { get; set; }

33 │

34 │ public int? NguoiXoa { get; set; }

35 │

36 │ public DateTime? NgayXoa { get; set; }

37 │

38 │ public long Ma { get; set; }

39 │ }

────────────────────────────────────────────────────────────

**📄 TonKhoBackgroundWorker.cs (CS, 89 lines)**

1 │ ﻿using QuanLyBanHang.Models.ExModel.OPS;

2 │ using QuanLyBanHang.Models.QLBHEntity;

3 │ using Microsoft.Extensions.DependencyInjection;

4 │ using Microsoft.Extensions.Hosting;

5 │ using Microsoft.Extensions.Logging;

6 │ using System;

7 │ using System.Threading;

8 │ using System.Threading.Tasks;

9 │ using System.Collections.Concurrent;

10 │

11 │ namespace QuanLyBanHang.Services

12 │ {

13 │ public class TonKhoBackgroundWorker : BackgroundService

14 │ {

15 │ private readonly IServiceScopeFactory \_serviceScopeFactory;

16 │ private readonly ILogger<TonKhoBackgroundWorker> \_logger;

17 │

18 │ private static readonly ConcurrentQueue<TonKhoJob> \_queue = new();

19 │

20 │ public TonKhoBackgroundWorker(IServiceScopeFactory serviceScopeFactory, ILogger<TonKhoBackgroundWorker> logger)

21 │ {

22 │ \_serviceScopeFactory = serviceScopeFactory;

23 │ \_logger = logger;

24 │ }

25 │

26 │ public static void AddJob(TonKhoJob job)

27 │ {

28 │ \_queue.Enqueue(job);

29 │ Console.WriteLine($"[DEBUG] Enqueued job: {job.MaKho}-{job.NamThang}-{job.MaChiNhanh}");

30 │ }

31 │

32 │ public static int QueueLength => \_queue.Count;

33 │ public static List<TonKhoJob> GetQueueSnapshot() => \_queue.ToList(); // dùng ConcurrentQueue nên gọi đượ...

34 │

35 │ protected override async Task ExecuteAsync(CancellationToken stoppingToken)

36 │ {

37 │ \_logger.LogInformation("TonKhoBackgroundWorker started.");

38 │

39 │ while (!stoppingToken.IsCancellationRequested)

40 │ {

41 │ if (\_queue.TryDequeue(out var job))

42 │ {

43 │ try

44 │ {

45 │ using var scope = \_serviceScopeFactory.CreateScope();

46 │ var context = scope.ServiceProvider.GetRequiredService<QuanLyBanHangContext>();

47 │ var tonKhoModel = new TonKhoModel(context);

48 │

49 │ var start = job.NamThang;

50 │ var end = DateTime.Now.ToString("yyyyMM");

51 │

52 │ var from = new DateTime(int.Parse(start.Substring(0, 4)), int.Parse(start.Substring(4, 2)), 1);

53 │ var to = new DateTime(int.Parse(end.Substring(0, 4)), int.Parse(end.Substring(4, 2)), 1);

54 │

55 │ while (from <= to)

56 │ {

57 │ var namThang = from.ToString("yyyyMM");

58 │

59 │ \_logger.LogInformation($"⏳ Đang tính tồn kho: Kho={job.MaKho}, Tháng={namThang}, Chi ...

60 │ tonKhoModel.TinhTonKho(job.MaKho, namThang, job.UserId, false, job.MaChiNhanh);

61 │ tonKhoModel.TinhTonKhoBanHang(job.MaKho, namThang);

62 │ \_logger.LogInformation($"✅ Đã xong: Kho={job.MaKho}, Tháng={namThang}, Chi nhánh={job....

63 │

64 │ from = from.AddMonths(1);

65 │ }

66 │ }

67 │ catch (Exception ex)

68 │ {

69 │ \_logger.LogError(ex, $"❌ Lỗi job tồn kho: Kho={job.MaKho}, Tháng={job.NamThang}, Chi nhá...

70 │ }

71 │ }

72 │ else

73 │ {

74 │ await Task.Delay(500, stoppingToken);

75 │ }

76 │ }

77 │

78 │ \_logger.LogInformation("TonKhoBackgroundWorker is stopping.");

79 │ }

80 │ }

81 │ // Định nghĩa 1 Job tồn kho

82 │ public class TonKhoJob

83 │ {

84 │ public string NamThang { get; set; }

85 │ public int MaKho { get; set; } // THÊM vào

86 │ public Guid UserId { get; set; }

87 │ public Guid MaChiNhanh { get; set; }

88 │ }

89 │ }

────────────────────────────────────────────────────────────

**📄 TonKhoBanHang.cs (CS, 29 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class TonKhoBanHang

7 │ {

8 │ public string NamThang { get; set; } = null!;

9 │

10 │ public int MaKho { get; set; }

11 │

12 │ public int MaHangHoa { get; set; }

13 │

14 │ public decimal? SltonDauKy { get; set; }

15 │

16 │ public decimal? SlnhapTrongKy { get; set; }

17 │

18 │ public decimal? SlxuatKhac { get; set; }

19 │

20 │ public decimal? SlxuatTieuDung { get; set; }

21 │

22 │ public decimal? SlxuatTra { get; set; }

23 │

24 │ public decimal? SlbanTrongKy { get; set; }

25 │

26 │ public decimal? SltonCuoiKy { get; set; }

27 │

28 │ public long Ma { get; set; }

29 │ }

────────────────────────────────────────────────────────────

**📄 TonKhoBanHang.cs (CS, 31 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class TonKhoBanHang

7 │ {

8 │ public string NamThang { get; set; } = null!;

9 │

10 │ public int MaKho { get; set; }

11 │

12 │ public int MaHangHoa { get; set; }

13 │

14 │ public decimal? SltonDauKy { get; set; }

15 │

16 │ public decimal? SlnhapTrongKy { get; set; }

17 │

18 │ public decimal? SlxuatKhac { get; set; }

19 │

20 │ public decimal? SlxuatTieuDung { get; set; }

21 │

22 │ public decimal? SlxuatTra { get; set; }

23 │

24 │ public decimal? SlbanTrongKy { get; set; }

25 │

26 │ public decimal? SltonCuoiKy { get; set; }

27 │

28 │ public long Ma { get; set; }

29 │

30 │ public Guid MaChiNhanh { get; set; }

31 │ }

────────────────────────────────────────────────────────────

**📄 TonKhoController.cs (CS, 864 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using QuanLyBanHang.Common;

3 │ using QuanLyBanHang.Common.Attributes;

4 │ using QuanLyBanHang.Models.APIModels;

5 │ using QuanLyBanHang.Models.ExModel.CAT;

6 │ using QuanLyBanHang.Models.ExModel.OPS;

7 │ using QuanLyBanHang.Models.QLBHEntity;

8 │ using QuanLyBanHang.Services;

9 │ using System.Data;

10 │

11 │ namespace QuanLyBanHang.Controllers.OPS

12 │ {

13 │ public class TonKhoController : BaseController

14 │ {

15 │ private readonly IWebHostEnvironment \_hostingEnvironment;

16 │

17 │ public TonKhoController(QuanLyBanHangContext dbcontext, MemoryCacheManager memoryCache, IWebHostEnvironment host...

18 │ {

19 │ m\_MenuID = MenuID.opsTonKho.GetEnumGuid();

20 │ \_hostingEnvironment = hostingEnvironment;

21 │ }

22 │

23 │ [TokenUserCheckHTTP]

24 │ public IActionResult Index(string MaKho, string Nam, string Thang, string LoaiBaoCao, string TuNgay, string DenN...

25 │ {

26 │ DateTime dtpTuNgay = TuNgay.ToDateTime2(DateTime.Now) ?? DateTime.Now;

27 │ DateTime dtpDenNgayRaw = DenNgay.ToDateTime2(DateTime.Now) ?? DateTime.Now;

28 │ DateTime dtpDenNgay = dtpDenNgayRaw.Date.AddDays(1).AddTicks(-1);

29 │

30 │ if (Cast.ToInt(Thang) == 0 || Cast.ToInt(Nam) == 0)

31 │ {

32 │ Thang = DateTime.Now.Month.ToString();

33 │ Nam = DateTime.Now.Year.ToString();

34 │ }

35 │ Thang = "0" + Thang;

36 │ Thang = Thang.Substring(Thang.Length - 2, 2);

37 │ var test = Nam + Thang;

38 │ ViewData["ListKho"] = new DanhMucKhoModel(dbcontext).GetDataByUser(objToken.userid, objToken.branch);

39 │

40 │ // Lấy giá trị integer của LoaiBaoCao một lần để sử dụng trong switch

41 │ int loaiBaoCaoInt = Cast.ToInt(LoaiBaoCao);

42 │

43 │ switch (loaiBaoCaoInt)

44 │ {

45 │ case (int)ELoaiBaoCao.BaoCaoTonKhoTheoThang:

46 │ ViewData["ListData"] = new TonKhoModel(dbcontext).GetTonKhoBanHang(Cast.ToInt(MaKho), Nam + Thang )....

47 │ break;

48 │

49 │ case (int)ELoaiBaoCao.BaoCaoTonKhoTheoNgay:

50 │ ViewData["ListDataTheoNgay"] = new TonKhoModel(dbcontext).TonKhoTuNgayDenNgay(Cast.ToInt(MaKho), dtp...

51 │ break;

52 │

53 │ case (int)ELoaiBaoCao.BaoCaoNhapHangImportFast:

54 │ ViewData["ListDataNhapHangImportFast"] = new TonKhoModel(dbcontext).BaoCaoNhapHangImportFast(Cast.To...

55 │ break;

56 │

57 │ case (int)ELoaiBaoCao.BaoCaoTongHopNhapHang:

58 │ ViewData["ListDataTongHopNhapHang"] = new NhapHangModel(dbcontext).GetListChiTietNhapHang(dtpTuNgay,...

59 │ break;

60 │

61 │ case (int)ELoaiBaoCao.BaoCaoTongHopXuatHang:

62 │ ViewData["ListDataTongHopXuatHang"] = new TonKhoModel(dbcontext).BaoCaoTongHopXuatBan(Cast.ToInt(MaK...

63 │ break;

64 │

65 │ case (int)ELoaiBaoCao.BaoCaoTonKhoCoGiaVon:

66 │ ViewData["ListDataGiaVon"] = new TonKhoModel(dbcontext).BaoCaoTonKhoGiaVon(Cast.ToInt(MaKho), Nam + ...

67 │ break;

68 │ }

69 │ var kho = dbcontext.StockDanhMucKho.FirstOrDefault(x => x.Ma == Cast.ToInt(MaKho));

70 │ ViewData["Nam"] = Nam;

71 │ ViewData["Thang"] = Thang;

72 │ ViewData["MaKho"] = MaKho;

73 │ ViewData["LoaiBaoCao"] = LoaiBaoCao;

74 │ ViewData["LoaiKho"] = kho == null ? 0 : kho.MaLoaiKho;

75 │ ViewData["TuNgay"] = dtpTuNgay;

76 │ ViewData["DenNgay"] = dtpDenNgay;

77 │ return View();

78 │ }

79 │ [TokenUserCheckHTTP]

80 │ public ActionResult TinhTonKho(string MaKho, string Nam, string Thang, string KhoaSo)

81 │ {

82 │ List<VPermission> ListPermission = GetVPermissions();

83 │ if (Cast.ToInt(Thang) == 0 || Cast.ToInt(Nam) == 0)

84 │ {

85 │ Thang = Cast.ToString(DateTime.Now.Month);

86 │ Nam = Cast.ToString(DateTime.Now.Year);

87 │ }

88 │ Thang = "0" + Thang;

89 │ Thang = Thang.Substring(Thang.Length - 2, 2);

90 │

91 │ if (!new KyKhoModel(dbcontext).KiemTraKyKhoTruocKhiTinhTonKyHT(Cast.ToInt(Nam), Cast.ToInt(Thang), Cast.ToIn...

92 │ {

93 │ return Json(new ResponseModel(QLBHConst.Error, "Kỳ trước chưa khoá sổ, Vui lòng chạy tồn k...

94 │ }

95 │

96 │ if (ListPermission.Any(t => t.PermissionId == Permission.KhoaSo.GetEnumGuid()))

97 │ {

98 │ new TonKhoModel(dbcontext).TinhTonKhoBanHang(Cast.ToInt(MaKho), Nam + "" + Thang);

99 │

100 │ return Json(new ResponseModel(QLBHConst.OK, "Tính tồn kho thành công"));

101 │ }

102 │ else

103 │ {

104 │ return Json(QLBHConst.ResponseNoPermission);

105 │ }

106 │ }

107 │ [TokenUserCheckHTTP]

108 │ public ActionResult TinhTonKhoGiaVon(string MaKho, string Nam, string Thang, string KhoaSo)

109 │ {

110 │ List<VPermission> ListPermission = GetVPermissions();

111 │ if (Cast.ToInt(Thang) == 0 || Cast.ToInt(Nam) == 0)

112 │ {

113 │ Thang = Cast.ToString(DateTime.Now.Month);

114 │ Nam = Cast.ToString(DateTime.Now.Year);

115 │ }

116 │ Thang = "0" + Thang;

117 │ Thang = Thang.Substring(Thang.Length - 2, 2);

118 │

119 │ if (!new KyKhoModel(dbcontext).KiemTraKyKhoTruocKhiTinhTonKyHT(Cast.ToInt(Nam), Cast.ToInt(Thang), Cast.ToIn...

120 │ {

121 │ return Json(new ResponseModel(QLBHConst.Error, "Kỳ trước chưa khoá sổ, Vui lòng chạy tồn k...

122 │ }

123 │

124 │ if (ListPermission.Any(t => t.PermissionId == Permission.KhoaSo.GetEnumGuid()))

125 │ {

126 │ TonKhoBackgroundWorker.AddJob(new TonKhoJob

127 │ {

128 │ NamThang = Nam + "" + Thang,

129 │ MaKho = Cast.ToInt(MaKho),

130 │ UserId = objToken.userid,

131 │ MaChiNhanh = objToken.branch

132 │ });

133 │ return Json(new ResponseModel(QLBHConst.OK, "Tính tồn kho thành công"));

134 │ }

135 │ else

136 │ {

137 │ return Json(QLBHConst.ResponseNoPermission);

138 │ }

139 │ }

140 │ [TokenUserCheckHTTP]

141 │ public ActionResult UpdateKyKho(string MaKho, string Nam, string Thang, string TrangThai)

142 │ {

143 │ List<VPermission> ListPermission = GetVPermissions();

144 │

145 │

146 │ if (Cast.ToInt(Thang) == 0 || Cast.ToInt(Nam) == 0)

147 │ {

148 │ Thang = Cast.ToString(DateTime.Now.Month);

149 │ Nam = Cast.ToString(DateTime.Now.Year);

150 │ }

151 │

152 │ if (!new KyKhoModel(dbcontext).KiemTraKyKhoTruocKhiTinhTonKyHT(Cast.ToInt(Nam), Cast.ToInt(Thang), Cast.ToIn...

153 │ {

154 │ if (TrangThai == "")

155 │ {

156 │ return Json(new ResponseModel(QLBHConst.Error, "Kỳ trước chưa khoá sổ, Vui lòng chạy t��...

157 │ }

158 │ }

159 │

160 │ if (ListPermission.Any(t => t.PermissionId == Permission.MoKhoaSo.GetEnumGuid()))

161 │ {

162 │ string sThang = "0" + Thang;

163 │ sThang = sThang.Substring(sThang.Length - 2, 2);

164 │ new KyKhoModel(dbcontext).UpdatekyKho(Cast.ToInt(MaKho), Cast.ToInt(Nam), Cast.ToInt(Thang), TrangThai =...

165 │ TonKhoBackgroundWorker.AddJob(new TonKhoJob

166 │ {

167 │ NamThang = string.Format("{0:yyyyMM}", Nam + sThang),

168 │ MaKho = Cast.ToInt(MaKho),

169 │ UserId = objToken.userid,

170 │ MaChiNhanh = objToken.branch

171 │ });

172 │ return Json(new ResponseModel(QLBHConst.OK, TrangThai == "1" ? "Khóa sổ thành công !" : "Mở sổ ...

173 │ }

174 │ else

175 │ {

176 │ return Json(QLBHConst.ResponseNoPermission);

177 │ }

178 │ }

179 │

180 │ [HttpPost]

181 │ [Route("api/tonkho/getbythang")]

182 │ [TokenUserCheckHTTP]

183 │ public IActionResult GetTonKhoTheoThang(

184 │ [FromForm] string maKho,

185 │ [FromForm] string nam,

186 │ [FromForm] string thang,

187 │ [FromForm] string loaiBaoCao = "1",

188 │ [FromForm] string tinhToanLai = "0",

189 │ [FromForm] string tuNgay = "",

190 │ [FromForm] string denNgay = "")

191 │ {

192 │ try

193 │ {

194 │ // Kiểm tra và xử lý thông tin đầu vào

195 │ maKho = string.IsNullOrEmpty(maKho) ? "0" : maKho;

196 │ nam = string.IsNullOrEmpty(nam) ? DateTime.Now.Year.ToString() : nam;

197 │ thang = string.IsNullOrEmpty(thang) ? DateTime.Now.Month.ToString() : thang;

198 │ bool isTinhToanLai = tinhToanLai == "1";

199 │ int loaiBaoCaoInt = Cast.ToInt(loaiBaoCao);

200 │

201 │ // Định dạng lại tháng (thêm số 0 ở đầu nếu cần)

202 │ thang = "0" + thang;

203 │ thang = thang.Substring(thang.Length - 2, 2);

204 │

205 │ DateTime dtpTuNgay = DateTime.Now;

206 │ DateTime dtpDenNgay = DateTime.Now;

207 │

208 │ // Tính toán lại tồn kho nếu cần

209 │ if (isTinhToanLai && Cast.ToInt(maKho) > 0)

210 │ {

211 │ TonKhoBackgroundWorker.AddJob(new TonKhoJob

212 │ {

213 │ NamThang = string.Format("{0:yyyyMM}", nam + thang),

214 │ MaKho = Cast.ToInt(maKho),

215 │ UserId = objToken.userid,

216 │ MaChiNhanh = objToken.branch

217 │ });

218 │ }

219 │

220 │ // Kết quả dữ liệu phụ thuộc vào loại báo cáo

221 │ object data = null;

222 │

223 │ switch (loaiBaoCaoInt)

224 │ {

225 │ case (int)ELoaiBaoCao.BaoCaoTonKhoTheoThang:

226 │ // Lấy tất cả dữ liệu tồn kho theo tháng

227 │ var dataTheoThang = new TonKhoModel(dbcontext)

228 │ .GetTonKhoBanHang(Cast.ToInt(maKho), nam + thang);

229 │

230 │ // Chuyển đổi tất cả các trường thành chuỗi

231 │ data = dataTheoThang.Select(item => new

232 │ {

233 │ MaKho = item.MaKho.ToString() ?? "",

234 │ MaKt = item.MaKt ?? "",

235 │ TenLop = item.TenLop ?? "",

236 │ MaVach = item.MaVach ?? "",

237 │ MaHangHoa = item.MaHangHoa.ToString() ?? "",

238 │ TenHangHoa = item.TenHangHoa ?? "",

239 │ SltonDauKy = item.SltonDauKy?.ToString() ?? "",

240 │ SlnhapTrongKy = item.SlnhapTrongKy?.ToString() ?? "",

241 │ SlbanTrongKy = item.SlbanTrongKy?.ToString() ?? "",

242 │ SltonCuoiKy = item.SltonCuoiKy?.ToString() ?? "",

243 │ NamThang = item.NamThang ?? "",

244 │ SlxuatKhac = item.SlxuatKhac?.ToString() ?? "",

245 │ SlxuatTieuDung = item.SlxuatTieuDung?.ToString() ?? "",

246 │ TenDv = item.TenDv ?? "",

247 │ SlxuatTra = item.SlxuatTra?.ToString() ?? "",

248 │ LoGanNhat = item.LoGanNhat.HasValue ? item.LoGanNhat.Value.ToString("dd/MM/yyyy") : "",

249 │ SoLuongGanNhat = item.SoLuongGanNhat?.ToString() ?? ""

250 │ }).ToList();

251 │ break;

252 │

253 │ case (int)ELoaiBaoCao.BaoCaoTonKhoTheoNgay:

254 │ // Sử dụng tham số tuNgay, denNgay đã được truyền vào

255 │

256 │ dtpTuNgay = tuNgay.ToDateTime2(DateTime.Now)!.Value;

257 │ dtpDenNgay = denNgay.ToDateTime2(DateTime.Now)!.Value;

258 │

259 │ var dataTheoNgay = new TonKhoModel(dbcontext).TonKhoTuNgayDenNgay(Cast.ToInt(maKho), dtpTuNgay, ...

260 │ data = dataTheoNgay;

261 │ break;

262 │

263 │ case (int)ELoaiBaoCao.BaoCaoNhapHangImportFast:

264 │ // Tham số đã được truyền vào từ tham số phương thức

265 │

266 │ dtpTuNgay = tuNgay.ToDateTime2(DateTime.Now)!.Value;

267 │ dtpDenNgay = denNgay.ToDateTime2(DateTime.Now)!.Value;

268 │

269 │ var dataNhapHangImportFast = new TonKhoModel(dbcontext).BaoCaoNhapHangImportFast(Cast.ToInt(maKh...

270 │ data = dataNhapHangImportFast;

271 │ break;

272 │

273 │ case (int)ELoaiBaoCao.BaoCaoTongHopNhapHang:

274 │ // Tham số đã được truyền vào từ tham số phương thức

275 │

276 │ dtpTuNgay = tuNgay.ToDateTime2(DateTime.Now)!.Value;

277 │ dtpDenNgay = denNgay.ToDateTime2(DateTime.Now)!.Value;

278 │

279 │ var dataTongHopNhapHang = new NhapHangModel(dbcontext).GetListChiTietNhapHang(dtpTuNgay, dtpDenN...

280 │ data = dataTongHopNhapHang;

281 │ break;

282 │

283 │ case (int)ELoaiBaoCao.BaoCaoTongHopXuatHang:

284 │ // Tham số đã được truyền vào từ tham số phương thức

285 │

286 │ dtpTuNgay = tuNgay.ToDateTime2(DateTime.Now)!.Value;

287 │ dtpDenNgay = denNgay.ToDateTime2(DateTime.Now)!.Value;

288 │

289 │ var dataTongHopXuatBan = new TonKhoModel(dbcontext).BaoCaoTongHopXuatBan(Cast.ToInt(maKho), dtpT...

290 │ data = dataTongHopXuatBan;

291 │ break;

292 │

293 │ case (int)ELoaiBaoCao.BaoCaoTonKhoCoGiaVon:

294 │ var dataTonKhoGiaVon = new TonKhoModel(dbcontext).BaoCaoTonKhoGiaVon(Cast.ToInt(maKho), nam + th...

295 │ data = dataTonKhoGiaVon;

296 │ break;

297 │

298 │ default:

299 │ // Mặc định lấy dữ liệu tồn kho theo tháng nếu không phù hợp với bất k��...

300 │ var dataDefault = new TonKhoModel(dbcontext)

301 │ .GetTonKhoBanHang(Cast.ToInt(maKho), nam + thang);

302 │

303 │ data = dataDefault.Select(item => new

304 │ {

305 │ MaKho = item.MaKho.ToString() ?? "",

306 │ MaKt = item.MaKt ?? "",

307 │ TenLop = item.TenLop ?? "",

308 │ MaVach = item.MaVach ?? "",

309 │ MaHangHoa = item.MaHangHoa.ToString() ?? "",

310 │ TenHangHoa = item.TenHangHoa ?? "",

311 │ SltonDauKy = item.SltonDauKy?.ToString() ?? "",

312 │ SlnhapTrongKy = item.SlnhapTrongKy?.ToString() ?? "",

313 │ SlbanTrongKy = item.SlbanTrongKy?.ToString() ?? "",

314 │ SltonCuoiKy = item.SltonCuoiKy?.ToString() ?? "",

315 │ NamThang = item.NamThang ?? "",

316 │ SlxuatKhac = item.SlxuatKhac?.ToString() ?? "",

317 │ SlxuatTieuDung = item.SlxuatTieuDung?.ToString() ?? "",

318 │ TenDv = item.TenDv ?? "",

319 │ SlxuatTra = item.SlxuatTra?.ToString() ?? "",

320 │ LoGanNhat = item.LoGanNhat.HasValue ? item.LoGanNhat.Value.ToString("dd/MM/yyyy") : "",

321 │ SoLuongGanNhat = item.SoLuongGanNhat?.ToString() ?? ""

322 │ }).ToList();

323 │ break;

324 │ }

325 │

326 │ // Trả về dữ liệu

327 │ return Json(new ResponseModel(QLBHConst.OK, "Lấy dữ liệu thành công", data));

328 │ }

329 │ catch (Exception ex)

330 │ {

331 │ return Json(new ResponseModel(QLBHConst.Error, "Lỗi: " + ex.Message, null));

332 │ }

333 │ }

334 │

335 │ [TokenUserCheckHTTP]

336 │ public ActionResult ExportExcel(string MaKho, string Nam, string Thang, string LoaiBaoCao, string TuNgay, string...

337 │ {

338 │ Stream x = null;

339 │ string contentType = "";

340 │ string fileName = "";

341 │ int loaiBaoCaoInt = Cast.ToInt(LoaiBaoCao);

342 │ DateTime dtpTuNgay = TuNgay.ToDateTime2(DateTime.Now) ?? DateTime.Now;

343 │ DateTime dtpDenNgayRaw = DenNgay.ToDateTime2(DateTime.Now) ?? DateTime.Now;

344 │ DateTime dtpDenNgay = dtpDenNgayRaw.Date.AddDays(1).AddTicks(-1);

345 │ string FileType = "Excel";

346 │

347 │ if (Cast.ToInt(Thang) == 0 || Cast.ToInt(Nam) == 0)

348 │ {

349 │ Thang = Cast.ToString(DateTime.Now.Month);

350 │ Nam = Cast.ToString(DateTime.Now.Year);

351 │ }

352 │ Thang = "0" + Thang;

353 │ Thang = Thang.Substring(Thang.Length - 2, 2);

354 │ var kho = new DanhMucKhoModel(dbcontext).GetItem(MaKho.ToInt());

355 │

356 │ if (Cast.ToInt(MaKho) > 0)

357 │ {

358 │ TonKhoBackgroundWorker.AddJob(new TonKhoJob

359 │ {

360 │ NamThang = string.Format("{0:yyyyMM}", Nam + Thang),

361 │ MaKho = Cast.ToInt(MaKho),

362 │ UserId = objToken.userid,

363 │ MaChiNhanh = objToken.branch

364 │ });

365 │ }

366 │ switch (loaiBaoCaoInt)

367 │ {

368 │ case (int)ELoaiBaoCao.BaoCaoTonKhoTheoThang:

369 │ var listDataVTonKhoBanHang = new TonKhoModel(dbcontext).GetTonKhoBanHang(Cast.ToInt(MaKho), Nam + Th...

370 │

371 │ DataTable dtb = new();

372 │ dtb.Columns.Add("Lớp hàng");

373 │ dtb.Columns.Add("Mã hàng");

374 │ dtb.Columns.Add("Mã vạch");

375 │ dtb.Columns.Add("Tên hàng hóa");

376 │ dtb.Columns.Add("ĐVT");

377 │ dtb.Columns.Add("Tồn đầu kỳ", typeof(decimal));

378 │ dtb.Columns.Add("Nhập trong kỳ", typeof(decimal));

379 │ dtb.Columns.Add("Xuất khác", typeof(decimal));

380 │ dtb.Columns.Add("Xuất trả", typeof(decimal));

381 │ dtb.Columns.Add("Xuất tiêu dùng", typeof(decimal));

382 │ dtb.Columns.Add("Bán trong kỳ", typeof(decimal));

383 │ dtb.Columns.Add("Tồn cuối kỳ", typeof(decimal));

384 │ dtb.Columns.Add("Lô gần nhất");

385 │ dtb.Columns.Add("SL gần nhất", typeof(decimal));

386 │

387 │ int STT = 1;

388 │ foreach (var item in listDataVTonKhoBanHang)

389 │ {

390 │ dtb.Rows.Add(

391 │ item.TenLop,

392 │ item.MaKt,

393 │ item.MaVach,

394 │ item.TenHangHoa,

395 │ item.TenDv,

396 │ item.SltonDauKy,

397 │ item.SlnhapTrongKy,

398 │ item.SlxuatKhac,

399 │ item.SlxuatTra,

400 │ item.SlxuatTieuDung,

401 │ item.SlbanTrongKy,

402 │ item.SltonCuoiKy,

403 │ item.LoGanNhat.HasValue ? item.LoGanNhat.Value.ToString("dd/MM/yyyy") : "",

404 │ item.SoLuongGanNhat

405 │ );

406 │ STT++;

407 │ }

408 │ fileName = FileType == "Excel" ? "BaoCaoTonKho.xlsx" : "BaoCaoTonKho.pdf";

409 │ contentType = FileType == "Excel" ? "application/ms-excel" : "application/pdf";

410 │

411 │ string fileTemplate = Path.Combine(\_hostingEnvironment.ContentRootPath, "wwwroot", "templates", "Bao...

412 │ List<ExcelValue> ListGiaTriRiengLe = new()

413 │ {

414 │ new ExcelValue {

415 │ Range = "A4",

416 │ Value = $"Kho: {kho?.Ten ?? "Tất cả"}",

417 │ Format = new ExcelFormat {

418 │ IsBold = true

419 │ }

420 │ },

421 │ new ExcelValue {

422 │ Range = "B3",

423 │ Value = $"Tháng năm: {Thang}/{Nam}",

424 │ Format = new ExcelFormat {

425 │ IsBold = true

426 │ }

427 │ },

428 │

429 │ };

430 │ int DongBatDau = 6;

431 │ x = ExportToExcel.FromTemplate(fileTemplate, FileType, ListGiaTriRiengLe, dtb, true, DongBatDau, "A"...

432 │

433 │ break;

434 │

435 │ case (int)ELoaiBaoCao.BaoCaoTonKhoTheoNgay:

436 │ var listDataTonKhoTuNgayDenNgay = new TonKhoModel(dbcontext).TonKhoTuNgayDenNgay(Cast.ToInt(MaKho), ...

437 │ DataTable dtbTonKhoTuNgayDenNgay = new();

438 │ dtbTonKhoTuNgayDenNgay.Columns.Add("STT", typeof(int));

439 │ dtbTonKhoTuNgayDenNgay.Columns.Add("MaKT", typeof(string));

440 │ dtbTonKhoTuNgayDenNgay.Columns.Add("TenHang", typeof(string));

441 │ dtbTonKhoTuNgayDenNgay.Columns.Add("DVT", typeof(string));

442 │ dtbTonKhoTuNgayDenNgay.Columns.Add("SoLuongDauKy", typeof(decimal));

443 │ dtbTonKhoTuNgayDenNgay.Columns.Add("GiaTriDauKy", typeof(decimal));

444 │ dtbTonKhoTuNgayDenNgay.Columns.Add("SoLuongNhap", typeof(decimal));

445 │ dtbTonKhoTuNgayDenNgay.Columns.Add("GiaTriNhap", typeof(decimal));

446 │ dtbTonKhoTuNgayDenNgay.Columns.Add("SoLuongXuatKhac", typeof(decimal));

447 │ dtbTonKhoTuNgayDenNgay.Columns.Add("GiaTriXuatKhac", typeof(decimal));

448 │ dtbTonKhoTuNgayDenNgay.Columns.Add("SoLuongBan", typeof(decimal));

449 │ dtbTonKhoTuNgayDenNgay.Columns.Add("GiaTriBan", typeof(decimal));

450 │ dtbTonKhoTuNgayDenNgay.Columns.Add("SoLuongTon", typeof(decimal));

451 │ dtbTonKhoTuNgayDenNgay.Columns.Add("GiaTriTon", typeof(decimal));

452 │

453 │ int STTTonKhoTuNgayDenNgay = 0;

454 │

455 │ foreach (var item in listDataTonKhoTuNgayDenNgay)

456 │ {

457 │ STTTonKhoTuNgayDenNgay++;

458 │

459 │ dtbTonKhoTuNgayDenNgay.Rows.Add(

460 │ STTTonKhoTuNgayDenNgay,

461 │ item.MaKT,

462 │ item.TenHang,

463 │ item.DVT,

464 │ item.SoLuongDauKy,

465 │ item.SoLuongDauKy \* item.GiaVon,

466 │ item.SoLuongNhapTrongKy,

467 │ item.SoLuongNhapTrongKy \* item.GiaVon,

468 │ item.SoLuongXuatKhac,

469 │ item.SoLuongXuatKhac \* item.GiaVon,

470 │ item.SoLuongBanTrongKy,

471 │ item.SoLuongBanTrongKy \* item.GiaVon,

472 │ item.SoLuongTonCuoiKy,

473 │ item.SoLuongTonCuoiKy \* item.GiaVon

474 │ );

475 │ }

476 │

477 │ fileName = FileType == "Excel" ? "BaoCaoTonKhoTheoNgay.xlsx" : "BaoCaoTonKhoTheoNgay.pdf";

478 │ contentType = FileType == "Excel" ? "application/ms-excel" : "application/pdf";

479 │

480 │ fileTemplate = Path.Combine(\_hostingEnvironment.ContentRootPath, "wwwroot", "templates", "BaoCaoTonK...

481 │ List<ExcelValue> ListGiaTriRiengTonKhoTuNgayDenNgay = new()

482 │ {

483 │ new() { Range = $"A6:C6", Value = $"Từ ngày: {TuNgay} đến ngày: {DenNgay}",

484 │ Format = new ExcelFormat {

485 │ IsBold = true,

486 │ HorizontalAlignment = eHorizontalAlignment.Left

487 │

488 │ }

489 │ },

490 │ new() { Range = $"A7:C7", Value = $"Kho hàng: {kho.Ten}",

491 │ Format = new ExcelFormat {

492 │ IsBold = true,

493 │ HorizontalAlignment = eHorizontalAlignment.Left

494 │ } }

495 │ };

496 │ int DongBatDauTonKhoTuNgayDenNgay = 10;

497 │ x = ExportToExcel.FromTemplate(fileTemplate, FileType, ListGiaTriRiengTonKhoTuNgayDenNgay, dtbTonKho...

498 │ break;

499 │

500 │ case (int)ELoaiBaoCao.BaoCaoNhapHangImportFast:

501 │ var listDataBaoCaoNhapHangImportFast = new TonKhoModel(dbcontext).BaoCaoNhapHangImportFast(Cast.ToIn...

502 │

503 │ DataTable dtbImportFast = new();

504 │ dtbImportFast.Columns.Add("MaDVCS");

505 │ dtbImportFast.Columns.Add("MaGD");

506 │ dtbImportFast.Columns.Add("NgayLapChungTu");

507 │ dtbImportFast.Columns.Add("NgayChungTu");

508 │ dtbImportFast.Columns.Add("SoHD");

509 │ dtbImportFast.Columns.Add("MaNT");

510 │ dtbImportFast.Columns.Add("TyGia");

511 │ dtbImportFast.Columns.Add("MaNhaCC");

512 │ dtbImportFast.Columns.Add("NguoiGiaoHang");

513 │ dtbImportFast.Columns.Add("TaiKhoanCo");

514 │ dtbImportFast.Columns.Add("MaThanhToan");

515 │ dtbImportFast.Columns.Add("DienGiai");

516 │ dtbImportFast.Columns.Add("TrangThai");

517 │ dtbImportFast.Columns.Add("MaKT");

518 │ dtbImportFast.Columns.Add("DVT");

519 │ dtbImportFast.Columns.Add("MaKho");

520 │ dtbImportFast.Columns.Add("ViTri");

521 │ dtbImportFast.Columns.Add("Lo");

522 │ dtbImportFast.Columns.Add("SoLuong", typeof(decimal));

523 │ dtbImportFast.Columns.Add("GiaVon", typeof(decimal));

524 │ dtbImportFast.Columns.Add("ThanhTien", typeof(decimal));

525 │ dtbImportFast.Columns.Add("TyLeCK", typeof(decimal));

526 │ dtbImportFast.Columns.Add("ChietKhauNT", typeof(decimal));

527 │ dtbImportFast.Columns.Add("ThanhTienSauCK", typeof(decimal));

528 │ dtbImportFast.Columns.Add("TaiKhoanKho");

529 │ dtbImportFast.Columns.Add("GiaVon2", typeof(decimal));

530 │ dtbImportFast.Columns.Add("ThanhTien2", typeof(decimal));

531 │ dtbImportFast.Columns.Add("ChietKhauNT2", typeof(decimal));

532 │ dtbImportFast.Columns.Add("ThanhTienSauCK2", typeof(decimal));

533 │ dtbImportFast.Columns.Add("MauBC");

534 │ dtbImportFast.Columns.Add("Empty");

535 │ dtbImportFast.Columns.Add("SoHD2");

536 │ dtbImportFast.Columns.Add("SoSeri");

537 │ dtbImportFast.Columns.Add("NgayChungTu2");

538 │ dtbImportFast.Columns.Add("MaNhaCC2");

539 │ dtbImportFast.Columns.Add("TenNhaCC");

540 │ dtbImportFast.Columns.Add("DiaChi");

541 │ dtbImportFast.Columns.Add("MST");

542 │ dtbImportFast.Columns.Add("ThanhTien3", typeof(decimal));

543 │ dtbImportFast.Columns.Add("ThanhTienSauCK3", typeof(decimal));

544 │ dtbImportFast.Columns.Add("MaThue");

545 │ dtbImportFast.Columns.Add("ThueSuat");

546 │ dtbImportFast.Columns.Add("TienThue", typeof(decimal));

547 │ dtbImportFast.Columns.Add("TienThue2", typeof(decimal));

548 │ dtbImportFast.Columns.Add("TKThue");

549 │ dtbImportFast.Columns.Add("BoPhan");

550 │ dtbImportFast.Columns.Add("NoiBo");

551 │

552 │ foreach (var item in listDataBaoCaoNhapHangImportFast)

553 │ {

554 │ dtbImportFast.Rows.Add(

555 │ item.MaDVCS,

556 │ item.MaGD,

557 │ string.Format("{0:dd/MM/yyyy}", item.NgayLapChungTu),

558 │ string.Format("{0:dd/MM/yyyy}", item.NgayChungTu),

559 │ item.SoHD,

560 │ item.MaNT,

561 │ item.TyGia,

562 │ item.MaNhaCC,

563 │ item.NguoiGiaoHang,

564 │ item.TaiKhoanCo,

565 │ item.MaThanhToan,

566 │ item.DienGiai,

567 │ item.TrangThai,

568 │ item.MaKT,

569 │ Utility.BoDauTiengViet(item.DVT),

570 │ item.MaKho,

571 │ item.ViTri,

572 │ item.Lo,

573 │ item.SoLuong,

574 │ item.GiaVon,

575 │ item.ThanhTien,

576 │ item.TyLeCK,

577 │ item.ChietKhauNT,

578 │ item.ThanhTienSauCK,

579 │ item.TaiKhoanKho,

580 │ item.GiaVon,

581 │ item.ThanhTien,

582 │ item.ChietKhauNT,

583 │ item.ThanhTienSauCK,

584 │ item.MauBC,

585 │ "",

586 │ item.SoHD,

587 │ item.SoSeri,

588 │ string.Format("{0:dd/MM/yyyy}", item.NgayChungTu),

589 │ item.MaNhaCC,

590 │ item.TenNhaCC,

591 │ item.DiaChi,

592 │ item.MST,

593 │ item.ThanhTien,

594 │ item.ThanhTienSauCK,

595 │ item.MaThue,

596 │ item.ThueSuat,

597 │ item.TienThue,

598 │ item.TienThue,

599 │ item.TKThue,

600 │ item.BoPhan,

601 │ item.NoiBo

602 │ );

603 │ }

604 │ fileName = FileType == "Excel" ? "BaoCaoTonKhoImportFast.xlsx" : "BaoCaoTonKhoImportFast.pdf";

605 │ contentType = FileType == "Excel" ? "application/ms-excel" : "application/pdf";

606 │

607 │ fileTemplate = Path.Combine(\_hostingEnvironment.ContentRootPath, "wwwroot", "templates", "BaoCaoTonK...

608 │ List<ExcelValue> ListGiaTriRiengImportFast = new()

609 │ {

610 │ };

611 │ int DongBatDauImportFast = 2;

612 │ x = ExportToExcel.FromTemplate(fileTemplate, FileType, ListGiaTriRiengImportFast, dtbImportFast, tru...

613 │ break;

614 │

615 │ case (int)ELoaiBaoCao.BaoCaoTongHopNhapHang:

616 │ var listDataGetListChiTietNhapHang = new NhapHangModel(dbcontext).GetListChiTietNhapHang(dtpTuNgay, ...

617 │ DataTable dtbNhapHang = new();

618 │ dtbNhapHang.Columns.Add("STT");

619 │ dtbNhapHang.Columns.Add("Kho");

620 │ dtbNhapHang.Columns.Add("Ngày hóa đơn");

621 │ dtbNhapHang.Columns.Add("Số HĐ");

622 │ dtbNhapHang.Columns.Add("Kho xuất"); // Cột trống

623 │ dtbNhapHang.Columns.Add("Ngày xuất"); // Cột trống

624 │ dtbNhapHang.Columns.Add("Chứng từ xuất"); // Cột trống

625 │ dtbNhapHang.Columns.Add("Số hóa đơn NCC");

626 │ dtbNhapHang.Columns.Add("Ngày hóa đơn NCC");

627 │ dtbNhapHang.Columns.Add("Mã KT NCC");

628 │ dtbNhapHang.Columns.Add("Tên nhà cung cấp");

629 │ dtbNhapHang.Columns.Add("Số lô"); // Cột trống

630 │ dtbNhapHang.Columns.Add("Hạn sử dụng");

631 │ dtbNhapHang.Columns.Add("Ghi chú");

632 │ dtbNhapHang.Columns.Add("Mã KT");

633 │ dtbNhapHang.Columns.Add("Tên hàng");

634 │ dtbNhapHang.Columns.Add("ĐVT");

635 │ dtbNhapHang.Columns.Add("Số lượng", typeof(decimal));

636 │ dtbNhapHang.Columns.Add("VAT (%)", typeof(decimal));

637 │ dtbNhapHang.Columns.Add("Đơn giá", typeof(decimal));

638 │ dtbNhapHang.Columns.Add("Thành tiền chưa VAT", typeof(decimal));

639 │ dtbNhapHang.Columns.Add("CK (%)", typeof(decimal));

640 │ dtbNhapHang.Columns.Add("CK tiền mặt", typeof(decimal));

641 │ dtbNhapHang.Columns.Add("Tổng CK", typeof(decimal));

642 │ dtbNhapHang.Columns.Add("Giá trị VAT", typeof(decimal));

643 │ dtbNhapHang.Columns.Add("Đơn giá có VAT", typeof(decimal));

644 │ dtbNhapHang.Columns.Add("Tổng tiền", typeof(decimal));

645 │

646 │ decimal ThanhTienChuaVAT = 0;

647 │ decimal TongCK = 0;

648 │ decimal GiaTriVAT = 0;

649 │ decimal DonGiaVAT = 0;

650 │ decimal TongTien = 0;

651 │ decimal SumTTChuaVAT = 0;

652 │ decimal SumTLCK = 0;

653 │ decimal SumCKTM = 0;

654 │ decimal SumTongCK = 0;

655 │ decimal SumGiaTriVAT = 0;

656 │ decimal SumTongTien = 0;

657 │ int stt = 1;

658 │ foreach (var item in listDataGetListChiTietNhapHang)

659 │ {

660 │ ThanhTienChuaVAT = (item.SoLuong ?? 0) \* (item.GiaVon ?? 0);

661 │ TongCK = (item.ChietKhauTm ?? 0) + (ThanhTienChuaVAT \* (item.ChietKhauPt ?? 0) / 100.0M);

662 │ GiaTriVAT = (ThanhTienChuaVAT - TongCK) \* ((item.Vat ?? 0) / 100.0M);

663 │ DonGiaVAT = (item.GiaVon ?? 0) + ((item.GiaVon ?? 0) \* (item.Vat ?? 0) / 100.0M);

664 │ TongTien = (ThanhTienChuaVAT - TongCK) + GiaTriVAT;

665 │

666 │ // Cộng dồn tổng

667 │ SumTTChuaVAT += ThanhTienChuaVAT;

668 │ SumTLCK += item.ChietKhauPt ?? 0;

669 │ SumCKTM += item.ChietKhauTm ?? 0;

670 │ SumTongCK += TongCK;

671 │ SumGiaTriVAT += GiaTriVAT;

672 │ SumTongTien += TongTien;

673 │

674 │ // Thêm dòng dữ liệu

675 │ dtbNhapHang.Rows.Add(

676 │ stt++,

677 │ item.TenKho,

678 │ item.NgayHoaDon == null ? item.CreatedDate.Value.ToString("dd/MM/yyyy") : item.NgayNhapHang....

679 │ item.SoHoaDon,

680 │ "", // Cột trống

681 │ "", // Cột trống

682 │ "", // Cột trống

683 │ item.SoHoaDonNcc,

684 │ item.NgayHoaDon == null ? item.NgayNhapHang.Value.ToString("dd/MM/yyyy") : item.NgayHoaDon.V...

685 │ item.MaKtncc,

686 │ item.TenNhaCc,

687 │ "", // Cột trống

688 │ item.HanSd.HasValue ? item.HanSd.Value.ToString("dd/MM/yyyy") : "",

689 │ item.GhiChu,

690 │ item.MaKt,

691 │ item.TenHangHoa,

692 │ item.TenDvt,

693 │ item.SoLuong,

694 │ item.Vat ?? 0,

695 │ item.GiaVon,

696 │ ThanhTienChuaVAT,

697 │ item.ChietKhauPt ?? 0,

698 │ item.ChietKhauTm ?? 0,

699 │ TongCK,

700 │ GiaTriVAT,

701 │ DonGiaVAT,

702 │ TongTien

703 │ );

704 │ }

705 │

706 │ // Thêm dòng tổng cộng

707 │ DataRow totalRow = dtbNhapHang.NewRow();

708 │ for (int i = 0; i < dtbNhapHang.Columns.Count; i++)

709 │ {

710 │ totalRow[i] = DBNull.Value; // Đặt giá trị null cho tất cả các cột

711 │ }

712 │ totalRow["Thành tiền chưa VAT"] = string.Format("{0:#,##0.00}", SumTTChuaVAT);

713 │ totalRow["CK (%)"] = string.Format("{0:#,##0.00}", SumTLCK);

714 │ totalRow["CK tiền mặt"] = string.Format("{0:#,##0.00}", SumCKTM);

715 │ totalRow["Tổng CK"] = string.Format("{0:#,##0.00}", SumTongCK);

716 │ totalRow["Giá trị VAT"] = string.Format("{0:#,##0.00}", SumGiaTriVAT);

717 │ totalRow["Tổng tiền"] = string.Format("{0:#,##0.00}", SumTongTien);

718 │ dtbNhapHang.Rows.Add(totalRow);

719 │

720 │ fileName = FileType == "Excel" ? "BaoCaoTongHopNhapHang.xlsx" : "BaoCaoTongHopNhapHang.pdf";

721 │ contentType = FileType == "Excel" ? "application/ms-excel" : "application/pdf";

722 │

723 │ fileTemplate = Path.Combine(\_hostingEnvironment.ContentRootPath, "wwwroot", "templates", "BaoCaoTong...

724 │ List<ExcelValue> ListGiaTriRiengLeNhapHang = new List<ExcelValue>();

725 │ int totalRowIndex = dtbNhapHang.Rows.Count + 8;

726 │

727 │ List<ExcelValue> ListGiaTriRiengLeNhapHang2 = new List<ExcelValue>

728 │ {

729 │ new ExcelValue {

730 │ Range = $"A{totalRowIndex+1}:AA{totalRowIndex+1}",

731 │ KhongDienGiaTri = true,

732 │ KhongGopO = true,

733 │ Format = new ExcelFormat {

734 │ IsBold = true,

735 │ NumberFormat = "#,##0",

736 │ HorizontalAlignment = eHorizontalAlignment.Right

737 │ }

738 │ }

739 │ };

740 │ int DongBatDauNhapHang = 9;

741 │ x = ExportToExcel.FromTemplate(fileTemplate, FileType, ListGiaTriRiengLeNhapHang, dtbNhapHang, true,...

742 │

743 │ break;

744 │

745 │ case (int)ELoaiBaoCao.BaoCaoTongHopXuatHang:

746 │ var listDataBaoCaoTongHopXuatBan = new TonKhoModel(dbcontext).BaoCaoTongHopXuatBan(Cast.ToInt(MaKho)...

747 │

748 │ DataTable dtbXuatBan = new();

749 │ dtbXuatBan.Columns.Add("STT");

750 │ dtbXuatBan.Columns.Add("Ngày xuất / bán");

751 │ dtbXuatBan.Columns.Add("Mã KT");

752 │ dtbXuatBan.Columns.Add("Mã Vạch");

753 │ dtbXuatBan.Columns.Add("Tên hàng hóa");

754 │ dtbXuatBan.Columns.Add("ĐVT");

755 │ dtbXuatBan.Columns.Add("Số Lượng Xuất", typeof(decimal));

756 │ dtbXuatBan.Columns.Add("Số Lượng Bán", typeof(decimal));

757 │

758 │ int index = 0;

759 │

760 │ foreach (var item in listDataBaoCaoTongHopXuatBan)

761 │ {

762 │ index++;

763 │ dtbXuatBan.Rows.Add(new object[]

764 │ {

765 │ index,

766 │ string.Format("'{0:dd/MM/yyyy}", item.NgayXuatBan),

767 │ string.Format("'{0}", item.MaKT),

768 │ string.Format("'{0}", item.MaVach),

769 │ item.TenHang ?? "",

770 │ item.TenDV ?? "",

771 │ item.SoLuongXuat,

772 │ item.SoLuongBan,

773 │ });

774 │ }

775 │

776 │ fileName = FileType == "Excel" ? "BaoCaoTongHopXuatBan.xlsx" : "BaoCaoTongHopXuatBan.pdf";

777 │ contentType = FileType == "Excel" ? "application/ms-excel" : "application/pdf";

778 │

779 │ fileTemplate = Path.Combine(\_hostingEnvironment.ContentRootPath, "wwwroot", "templates", "BaoCaoTong...

780 │ List<ExcelValue> ListGiaTriRiengLeXuatBan = new()

781 │ {

782 │ };

783 │ int DongBatDauXuatBan = 8;

784 │ x = ExportToExcel.FromTemplate(fileTemplate, FileType, ListGiaTriRiengLeXuatBan, dtbXuatBan, true, D...

785 │ break;

786 │ case (int)ELoaiBaoCao.BaoCaoTonKhoCoGiaVon:

787 │ var listDataBaoCaoTonKhoGiaVon = new TonKhoModel(dbcontext).BaoCaoTonKhoGiaVon(Cast.ToInt(MaKho), Na...

788 │ DataTable dtbTonKhoGiaVon = new();

789 │ dtbTonKhoGiaVon.Columns.Add("STT");

790 │ dtbTonKhoGiaVon.Columns.Add("TenLop");

791 │ dtbTonKhoGiaVon.Columns.Add("MaVach");

792 │ dtbTonKhoGiaVon.Columns.Add("MaKT");

793 │ dtbTonKhoGiaVon.Columns.Add("TenHang");

794 │ dtbTonKhoGiaVon.Columns.Add("TenDVT");

795 │ dtbTonKhoGiaVon.Columns.Add("GiaVon", typeof(decimal));

796 │ dtbTonKhoGiaVon.Columns.Add("SLTonDauKy", typeof(decimal));

797 │ dtbTonKhoGiaVon.Columns.Add("SLNhapTrongKy", typeof(decimal));

798 │ dtbTonKhoGiaVon.Columns.Add("SLXuatKhac", typeof(decimal));

799 │ dtbTonKhoGiaVon.Columns.Add("SLXuatTra", typeof(decimal));

800 │ dtbTonKhoGiaVon.Columns.Add("SLXuatTieuDung", typeof(decimal));

801 │ dtbTonKhoGiaVon.Columns.Add("SLBanTrongKy", typeof(decimal));

802 │ dtbTonKhoGiaVon.Columns.Add("SLTonCuoiKy", typeof(decimal));

803 │

804 │ int indexTonKhoGiaVon = 0;

805 │

806 │ foreach (var item in listDataBaoCaoTonKhoGiaVon)

807 │ {

808 │ indexTonKhoGiaVon++;

809 │ dtbTonKhoGiaVon.Rows.Add(new object[]

810 │ {

811 │ indexTonKhoGiaVon,

812 │ item.TenLop ?? "",

813 │ item.MaVach?? "",

814 │ item.MaKT ?? "",

815 │ item.TenHang ?? "",

816 │ item.TenDVT ?? "",

817 │ item.GiaVon,

818 │ item.SLTonDauKy,

819 │ item.SLNhapTrongKy,

820 │ item.SLXuatKhac,

821 │ item.SLXuatTra,

822 │ item.SLXuatTieuDung,

823 │ item.SLBanTrongKy,

824 │ item.SLTonCuoiKy

825 │ });

826 │ }

827 │ fileName = FileType == "Excel" ? "BaoCaoTonKhoGiaVon.xlsx" : "BaoCaoTonKhoGiaVon.pdf";

828 │ contentType = FileType == "Excel" ? "application/ms-excel" : "application/pdf";

829 │

830 │ fileTemplate = Path.Combine(\_hostingEnvironment.ContentRootPath, "wwwroot", "templates", "BaoCaoTonK...

831 │ List<ExcelValue> ListGiaTriRiengLeTonKhoGiaVon = new()

832 │ {

833 │ new() { Range = $"A2:{ExportToExcel.GetExcelColumnName(dtbTonKhoGiaVon.Columns.Count)}2", Value ...

834 │ Format = new ExcelFormat {

835 │ IsBold = true,

836 │ HorizontalAlignment = eHorizontalAlignment.Center

837 │

838 │ }

839 │ },

840 │ new() { Range = $"A3:{ExportToExcel.GetExcelColumnName(dtbTonKhoGiaVon.Columns.Count)}3", Value ...

841 │ Format = new ExcelFormat {

842 │ IsBold = true,

843 │ HorizontalAlignment = eHorizontalAlignment.Center

844 │ } }

845 │

846 │ };

847 │ int DongBatDauTonKhoGiaVon = 6;

848 │ x = ExportToExcel.FromTemplate(fileTemplate, FileType, ListGiaTriRiengLeTonKhoGiaVon, dtbTonKhoGiaVo...

849 │ break;

850 │ }

851 │

852 │ ViewData["Nam"] = Nam;

853 │ ViewData["Thang"] = Thang;

854 │ ViewData["MaKho"] = MaKho;

855 │ ViewData["LoaiBaoCao"] = LoaiBaoCao;

856 │ ViewData["TuNgay"] = dtpTuNgay;

857 │ ViewData["DenNgay"] = dtpDenNgay;

858 │ return new FileStreamResult(x, contentType)

859 │ {

860 │ FileDownloadName = fileName

861 │ };

862 │ }

863 │ }

864 │ }

────────────────────────────────────────────────────────────

**📄 TonKhoK.cs (CS, 29 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class TonKhoK

7 │ {

8 │ public string? NamThang { get; set; }

9 │

10 │ public int? MaKho { get; set; }

11 │

12 │ public int? MaHangHoa { get; set; }

13 │

14 │ public int? MaNhapHang { get; set; }

15 │

16 │ public int? SltonDauKy { get; set; }

17 │

18 │ public int? SlnhapTrongKy { get; set; }

19 │

20 │ public int? SlxuatKhac { get; set; }

21 │

22 │ public int? SlxuatTieuDung { get; set; }

23 │

24 │ public int? SlbanTrongKy { get; set; }

25 │

26 │ public int? SltonCuoiKy { get; set; }

27 │

28 │ public int Ma { get; set; }

29 │ }

────────────────────────────────────────────────────────────

**📄 TonKhoModel.cs (CS, 1815 lines)**

1 │ ﻿using System;

2 │ using System.Diagnostics;

3 │ using System.Drawing;

4 │ using System.Globalization;

5 │ using System.Linq;

6 │ using System.Net.WebSockets;

7 │ using System.Reflection.PortableExecutable;

8 │ using EFCore.BulkExtensions;

9 │ using Microsoft.EntityFrameworkCore;

10 │ using NetTopologySuite.Index.HPRtree;

11 │ using QuanLyBanHang.Common;

12 │ using QuanLyBanHang.Models.ExModel.StoredProcedureModel;

13 │ using QuanLyBanHang.Models.QLBHEntity;

14 │ using QuanLyBanHang.Services;

15 │

16 │ namespace QuanLyBanHang.Models.ExModel.OPS

17 │ {

18 │ public class TonKhoModel : Base<StockTonKho>

19 │ {

20 │ private readonly QuanLyBanHangContext \_context;

21 │

22 │ private class HangHoaSoLuong

23 │ {

24 │ public int MaHangHoa { get; set; }

25 │ public decimal SoLuong { get; set; }

26 │ }

27 │ public TonKhoModel(QuanLyBanHangContext context) : base(context)

28 │ {

29 │ \_context = context;

30 │ }

31 │

32 │ public void TinhTonKho(int MaKho, string NamThang, Guid User, bool KhoaSo, Guid MaChiNhanh)

33 │ {

34 │

35 │ var kho = \_context.StockDanhMucKho.FirstOrDefault(x => x.Ma == MaKho);

36 │ var chiNhanh = \_context.SysBranch.AsNoTracking().FirstOrDefault(x => x.Ma == MaChiNhanh);

37 │ if (kho == null)

38 │ {

39 │ throw new Exception("Không tìm thấy kho");

40 │ }

41 │

42 │ if (chiNhanh == null)

43 │ {

44 │ throw new Exception("Không tìm thấy đơn vị");

45 │ }

46 │

47 │ switch (kho.MaLoaiKho)

48 │ {

49 │ case (int)LoaiKho.KhoTieuHao: TinhTonKho\_KhoTieuHao(NamThang, MaKho, MaChiNhanh, chiNhanh.Nid ?? 0); bre...

50 │ case (int)LoaiKho.KhoNguyenVatLieu: TinhTonKho\_KhoNguyenLieu(NamThang, MaKho, MaChiNhanh, chiNhanh.Nid ?...

51 │ case (int)LoaiKho.KhoBanHang: TinhTonKho\_KhoBanHang(NamThang, MaKho, MaChiNhanh, chiNhanh.Nid ?? 0); bre...

52 │ }

53 │ }

54 │ public void TinhTonKho\_KhoBanHang(string NamThang, int MaKho, Guid MaChiNhanh, int MaChiNhanhInt)

55 │ {

56 │ var br = \_context.SysBranch.FirstOrDefault(x => x.Ma == MaChiNhanh);

57 │ if (br == null)

58 │ {

59 │ throw new Exception("Không tìm thấy chi nhánh");

60 │ }

61 │ var currentDate = DateTime.ParseExact(NamThang + "01", "yyyyMMdd", CultureInfo.InvariantCulture);

62 │ string namThangTruoc = currentDate.AddMonths(-1).ToString("yyyyMM");

63 │

64 │ var tonKhoData = new List<StockTonKho>();

65 │ int year = currentDate.Year, month = currentDate.Month;

66 │

67 │ // 1. Tồn đầu kỳ từ tháng trước

68 │ var tonDau = \_context.StockTonKho.AsNoTracking()

69 │ .Where(t => t.NamThang == namThangTruoc && t.MaKho == MaKho && t.MaChiNhanhInt == MaChiNhanhInt && t.Slt...

70 │ .Select(t => new StockTonKho

71 │ {

72 │ NamThang = NamThang,

73 │ MaKho = t.MaKho,

74 │ MaHangHoa = t.MaHangHoa,

75 │ MaNhapHang = t.MaNhapHang,

76 │ GiaVon = t.GiaVon,

77 │ SltonDauKy = t.SltonCuoiKy ?? 0,

78 │ SlnhapTrongKy = 0,

79 │ SlbanTrongKy = 0,

80 │ SltonCuoiKy = t.SltonCuoiKy,

81 │ MaChiNhanh = t.MaChiNhanh,

82 │ MaChiNhanhInt = br.Nid ?? 0

83 │ }).ToList();

84 │ tonKhoData.AddRange(tonDau);

85 │

86 │ // 2. Nhập từ NCC

87 │ var nhapHang = \_context.StockNhapHang.AsNoTracking()

88 │ .Join(\_context.StockChiTietNhapHang.AsNoTracking(),

89 │ nh => nh.Ma, ctnh => ctnh.MaNhapHang, (nh, ctnh) => new { nh, ctnh })

90 │ .Where(x => x.nh.NgayNhapHang.HasValue && x.nh.NgayNhapHang.Value.Year == year &&

91 │ x.nh.NgayNhapHang.Value.Month == month &&

92 │ !(x.nh.Deleted ?? false) &&

93 │ !(x.ctnh.Deleted ?? false) &&

94 │ x.nh.MaKho == MaKho &&

95 │ x.nh.MaChiNhanhInt == MaChiNhanhInt)

96 │ .GroupBy(x => new { x.ctnh.MaHangHoa, x.nh.Ma, x.nh.MaKho, x.nh.MaChiNhanh })

97 │ .Select(g => new StockTonKho

98 │ {

99 │ NamThang = NamThang,

100 │ MaKho = g.Key.MaKho,

101 │ MaHangHoa = g.Key.MaHangHoa,

102 │ MaNhapHang = g.Key.Ma,

103 │ GiaVon = g.Max(x => x.ctnh.GiaVon),

104 │ SltonDauKy = 0,

105 │ SlnhapTrongKy = g.Sum(x => x.ctnh.SoLuong ?? 0),

106 │ SlbanTrongKy = 0,

107 │ SltonCuoiKy = g.Sum(x => x.ctnh.SoLuong ?? 0),

108 │ MaChiNhanh = g.Key.MaChiNhanh,

109 │ MaChiNhanhInt = br.Nid ?? 0

110 │ }).ToList();

111 │ tonKhoData.AddRange(nhapHang);

112 │

113 │ // 3. Nhập từ chuyển kho

114 │ var nhapChuyenKho = \_context.StockXuatKho.AsNoTracking()

115 │ .Join(\_context.StockChiTietXuatKho.AsNoTracking(),

116 │ xk => xk.Ma, ctxk => ctxk.MaPhieuXuat, (xk, ctxk) => new { xk, ctxk })

117 │ .Where(x => x.xk.NgayXuat.HasValue && x.xk.NgayXuat.Value.Year == year &&

118 │ x.xk.NgayXuat.Value.Month == month &&

119 │ !(x.xk.Deleted ?? false) &&

120 │ !(x.ctxk.Deleted ?? false) &&

121 │ x.xk.MaKhoChuyenDen == MaKho &&

122 │ x.xk.MaChiNhanhInt == MaChiNhanhInt)

123 │ .GroupBy(x => new { x.ctxk.MaHangHoa, x.xk.Ma, x.xk.MaKhoChuyenDen, x.xk.MaChiNhanh })

124 │ .Select(g => new StockTonKho

125 │ {

126 │ NamThang = NamThang,

127 │ MaKho = g.Key.MaKhoChuyenDen,

128 │ MaHangHoa = g.Key.MaHangHoa,

129 │ MaNhapHang = g.Key.Ma,

130 │ GiaVon = g.Max(x => x.ctxk.GiaVon),

131 │ SltonDauKy = 0,

132 │ SlnhapTrongKy = g.Sum(x => x.ctxk.SoLuong ?? 0),

133 │ SlbanTrongKy = 0,

134 │ SltonCuoiKy = g.Sum(x => x.ctxk.SoLuong ?? 0),

135 │ MaChiNhanh = g.Key.MaChiNhanh,

136 │ MaChiNhanhInt = br.Nid ?? 0

137 │ }).ToList();

138 │ tonKhoData.AddRange(nhapChuyenKho);

139 │ // Nhập trả hàng

140 │ var nhapTraHang = (from bh in \_context.OpsHoaDon.AsNoTracking().Where(x => x.MaChiNhanhInt == MaChiNhanhInt)

141 │ join ct in \_context.OpsChiTietHoaDon.AsNoTracking().Where(x => x.MaChiNhanhInt == MaChiNh...

142 │ join hh in \_context.TblHangHoa.AsNoTracking().Where(x=>x.MaChiNhanhInt == MaChiNhanhInt) ...

143 │ where bh.MaKho == MaKho && bh.NgayHoaDon.Year == year && bh.NgayHoaDon.Month == month

144 │ && !(bh.Deleted ?? false) && !(ct.Deleted ?? false) && ct.SoLuongTraLai != 0

145 │ group new { ct, hh } by new { ct.MaHangHoa } into g

146 │ select new StockTonKho

147 │ {

148 │ NamThang = NamThang,

149 │ MaKho = MaKho,

150 │ MaHangHoa = g.Key.MaHangHoa,

151 │ GiaVon = g.Select(x => x.hh.GiaVon).FirstOrDefault(),

152 │ SlnhapTrongKy = g.Sum(x => x.ct.SoLuongTraLai ?? 0),

153 │ SltonDauKy = 0,

154 │ SlxuatKhac = 0,

155 │ SlxuatTieuDung = 0,

156 │ SlxuatTra = 0,

157 │ SlbanTrongKy = 0,

158 │ SltonCuoiKy = 0,

159 │ MaChiNhanh = MaChiNhanh,

160 │ MaChiNhanhInt = br.Nid ?? 0,

161 │ }).ToList();

162 │ tonKhoData.AddRange(nhapTraHang);

163 │

164 │ // 4. Xuất kho

165 │ var xuatKho = \_context.StockXuatKho.AsNoTracking()

166 │ .Join(\_context.StockChiTietXuatKho.AsNoTracking(),

167 │ xk => xk.Ma, ctxk => ctxk.MaPhieuXuat, (xk, ctxk) => new { xk, ctxk })

168 │ .Where(x => x.xk.NgayXuat.HasValue && x.xk.NgayXuat.Value.Year == year &&

169 │ x.xk.NgayXuat.Value.Month == month &&

170 │ !(x.xk.Deleted ?? false) &&

171 │ !(x.ctxk.Deleted ?? false) &&

172 │ x.xk.MaKho == MaKho &&

173 │ x.xk.MaChiNhanhInt == MaChiNhanhInt)

174 │ .GroupBy(x => new { x.ctxk.MaHangHoa, x.xk.MaChiNhanh })

175 │ .Select(g => new

176 │ {

177 │ g.Key.MaHangHoa,

178 │ SLXuat = g.Sum(x => x.ctxk.SoLuong ?? 0),

179 │ g.Key.MaChiNhanh

180 │ }).ToList();

181 │

182 │ foreach (var xk in xuatKho)

183 │ {

184 │ decimal slXuat = xk.SLXuat;

185 │ while (slXuat > 0)

186 │ {

187 │ var tk = tonKhoData.Where(t => t.MaHangHoa == xk.MaHangHoa && (t.SltonCuoiKy ?? 0) > 0)

188 │ .OrderBy(t => t.MaNhapHang)

189 │ .FirstOrDefault();

190 │ if (tk != null)

191 │ {

192 │ var canTru = Math.Min(slXuat, tk.SltonCuoiKy ?? 0);

193 │ tk.SlxuatKhac = (tk.SlxuatKhac ?? 0) + canTru;

194 │ tk.SltonCuoiKy -= canTru;

195 │ slXuat -= canTru;

196 │ }

197 │ else

198 │ {

199 │ tonKhoData.Add(new StockTonKho

200 │ {

201 │ NamThang = NamThang,

202 │ MaKho = MaKho,

203 │ MaHangHoa = xk.MaHangHoa,

204 │ SlxuatKhac = slXuat,

205 │ SltonCuoiKy = -slXuat,

206 │ MaChiNhanh = xk.MaChiNhanh,

207 │ MaChiNhanhInt = br.Nid ?? 0

208 │ });

209 │ slXuat = 0;

210 │ }

211 │ }

212 │ }

213 │

214 │ // 5. Bán hàng

215 │ var banHang = \_context.OpsHoaDon.AsNoTracking()

216 │ .Join(\_context.OpsChiTietHoaDon.AsNoTracking(),

217 │ bh => bh.Ma, ct => ct.MaHoaDon, (bh, ct) => new { bh, ct })

218 │ .Where(x => x.bh.NgayHoaDon.Year == year &&

219 │ x.bh.NgayHoaDon.Month == month &&

220 │ !(x.bh.Deleted ?? false) &&

221 │ !(x.ct.Deleted ?? false) &&

222 │ x.bh.MaKho == MaKho &&

223 │ x.bh.MaChiNhanhInt == MaChiNhanhInt)

224 │ .GroupBy(x => new { x.ct.MaHangHoa, x.bh.MaChiNhanh })

225 │ .Select(g => new

226 │ {

227 │ g.Key.MaHangHoa,

228 │ SLBan = g.Sum(x => (x.ct.SoLuong ?? 0) - (x.ct.SoLuongTraLai ?? 0)),

229 │ g.Key.MaChiNhanh

230 │ }).ToList();

231 │

232 │ foreach (var bh in banHang)

233 │ {

234 │ decimal slBan = bh.SLBan;

235 │ while (slBan > 0)

236 │ {

237 │ var tk = tonKhoData.Where(t => t.MaHangHoa == bh.MaHangHoa && (t.SltonCuoiKy ?? 0) > 0)

238 │ .OrderBy(t => t.MaNhapHang)

239 │ .FirstOrDefault();

240 │ if (tk != null)

241 │ {

242 │ var canTru = Math.Min(slBan, tk.SltonCuoiKy ?? 0);

243 │ tk.SlbanTrongKy = (tk.SlbanTrongKy ?? 0) + canTru;

244 │ tk.SltonCuoiKy -= canTru;

245 │ slBan -= canTru;

246 │ }

247 │ else

248 │ {

249 │ tonKhoData.Add(new StockTonKho

250 │ {

251 │ NamThang = NamThang,

252 │ MaKho = MaKho,

253 │ MaHangHoa = bh.MaHangHoa,

254 │ SlbanTrongKy = slBan,

255 │ SltonCuoiKy = -slBan,

256 │ MaChiNhanh = bh.MaChiNhanh,

257 │ MaChiNhanhInt = br.Nid ?? 0

258 │ });

259 │ slBan = 0;

260 │ }

261 │ }

262 │ }

263 │ // Tính tồn cuối kỳ

264 │ foreach (var item in tonKhoData)

265 │ {

266 │ item.SltonCuoiKy = (item.SltonDauKy ?? 0) + (item.SlnhapTrongKy ?? 0)

267 │ - (item.SlbanTrongKy ?? 0) - (item.SlxuatKhac ?? 0)

268 │ - (item.SlxuatTieuDung ?? 0) - (item.SlxuatTra ?? 0);

269 │ }

270 │ using var transaction = \_context.Database.BeginTransaction();

271 │ try

272 │ {

273 │ var oldData = \_context.StockTonKho.Where(x => x.NamThang == NamThang && x.MaKho == MaKho).ToList();

274 │ \_context.BulkDelete(oldData);

275 │ \_context.SaveChanges();

276 │ \_context.BulkInsert(tonKhoData);

277 │ \_context.SaveChanges();

278 │ transaction.Commit();

279 │ }

280 │ catch (Exception ex)

281 │ {

282 │ transaction.Rollback();

283 │ throw new Exception($"❌ Lỗi khi cập nhật tồn kho: Tháng={NamThang}, Kho={MaKho} \n{ex.Message...

284 │ }

285 │ }

286 │ public void TinhTonKho\_KhoNguyenLieu(string namThang, int maKho, Guid maChiNhanh, int MaChiNhanhInt)

287 │ {

288 │ var currentMonth = DateTime.ParseExact(namThang + "01", "yyyyMMdd", CultureInfo.InvariantCulture);

289 │ var year = currentMonth.Year;

290 │ var month = currentMonth.Month;

291 │ var thangTruoc = currentMonth.AddMonths(-1).ToString("yyyyMM");

292 │

293 │ var branch = \_context.SysBranch.FirstOrDefault(x => x.Ma == maChiNhanh)

294 │ ?? throw new Exception("Không tìm thấy chi nhánh");

295 │

296 │ var tonKhoData = new List<StockTonKho>();

297 │

298 │ // 1. Tồn đầu kỳ

299 │ var tonTruoc = \_context.StockTonKho.AsNoTracking()

300 │ .Where(x => x.NamThang == thangTruoc && x.MaKho == maKho && x.MaChiNhanhInt == MaChiNhanhInt && (x.Slton...

301 │ .Select(x => new StockTonKho

302 │ {

303 │ NamThang = namThang,

304 │ MaKho = maKho,

305 │ MaNhapHang = x.MaNhapHang,

306 │ MaHangHoa = x.MaHangHoa,

307 │ GiaVon = x.GiaVon,

308 │ SltonDauKy = x.SltonCuoiKy ?? 0,

309 │ SlnhapTrongKy = 0,

310 │ SlxuatTieuDung = 0,

311 │ SltonCuoiKy = x.SltonCuoiKy ?? 0,

312 │ MaChiNhanh = maChiNhanh,

313 │ MaChiNhanhInt = branch.Nid ?? 0

314 │ }).ToList();

315 │

316 │ tonKhoData.AddRange(tonTruoc);

317 │

318 │ // 2. Nhập từ NCC

319 │ var nhapHang = \_context.StockNhapHang.AsNoTracking()

320 │ .Join(\_context.StockChiTietNhapHang.AsNoTracking(), nh => nh.Ma, ct => ct.MaNhapHang, (nh, ct) => new { ...

321 │ .Where(x => x.nh.NgayNhapHang.Value.Year == year && x.nh.NgayNhapHang.Value.Month == month &&

322 │ !(x.nh.Deleted ?? false) && !(x.ct.Deleted ?? false) &&

323 │ x.nh.MaKho == maKho && x.nh.MaChiNhanhInt == MaChiNhanhInt)

324 │ .GroupBy(x => new {x.ct.MaNhapHang, x.ct.MaHangHoa})

325 │ .Select(g => new StockTonKho{

326 │ NamThang = namThang,

327 │ MaKho = maKho,

328 │ MaNhapHang = g.Key.MaNhapHang,

329 │ MaHangHoa = g.Key.MaHangHoa,

330 │ GiaVon = g.Max(x => x.ct.GiaVon),

331 │ SlnhapTrongKy = g.Sum(x => x.ct.SoLuong ?? 0),

332 │ MaChiNhanh = maChiNhanh,

333 │ MaChiNhanhInt = branch.Nid ?? 0

334 │ }).ToList();

335 │ tonKhoData.AddRange(nhapHang);

336 │

337 │ // 4. Xuất kho nguyên liệu khi nhập khay (đã chuyển sang tiêu hao)

338 │ var xuatNguyenLieu = \_context.VtthNhapXuatKhay.AsNoTracking().Where(x => x.LoaiPhieu == (int)LoaiNhapXuatVTT...

339 │ .Join(\_context.VtthKhay.AsNoTracking().Where(x => x.MaChiNhanhInt == MaChiNhanhInt), nx => nx.MaKhay, k ...

340 │ .Join(\_context.TblHangHoa.AsNoTracking().Where(x=>x.MaChiNhanhInt == MaChiNhanhInt), x => x.k.MaVatTu, h...

341 │ .Where(x => x.nx.NgayNhap.HasValue &&

342 │ x.nx.NgayNhap.Value.Year == year && x.nx.NgayNhap.Value.Month == month &&

343 │ !(x.nx.Deleted ?? false) && x.k.MaKhoNguyenLieu == maKho)

344 │ .GroupBy(x => x.k.MaVatTu)

345 │ .Select(g => new

346 │ {

347 │ MaHangHoa = g.Key,

348 │ SoLuong = g.Sum(x => x.nx.SoLuongNhap ?? 0),

349 │ GiaVon = g.Select(x => x.hh.GiaVon ?? 0).FirstOrDefault()

350 │ })

351 │ .ToList();

352 │

353 │ foreach (var item in xuatNguyenLieu)

354 │ {

355 │

356 │ decimal slXuat = item.SoLuong;

357 │ while (slXuat > 0)

358 │ {

359 │ var tk = tonKhoData.Where(t => t.MaHangHoa == item.MaHangHoa && (t.SltonCuoiKy ?? 0) > 0)

360 │ .OrderBy(t => t.MaNhapHang)

361 │ .FirstOrDefault();

362 │ if (tk != null)

363 │ {

364 │ var canTru = Math.Min(slXuat, tk.SltonCuoiKy ?? 0);

365 │ tk.SlxuatKhac = (tk.SlxuatKhac ?? 0) + canTru;

366 │ tk.SltonCuoiKy -= canTru;

367 │ slXuat -= canTru;

368 │ }

369 │ else

370 │ {

371 │ tonKhoData.Add(new StockTonKho

372 │ {

373 │ NamThang = namThang,

374 │ MaKho = maKho,

375 │ GiaVon = item.GiaVon,

376 │ MaHangHoa = item.MaHangHoa ?? 0,

377 │ SltonDauKy = 0,

378 │ SlnhapTrongKy = 0,

379 │ SlxuatTieuDung = item.SoLuong,

380 │ SltonCuoiKy = -item.SoLuong,

381 │ MaChiNhanh = maChiNhanh,

382 │ MaChiNhanhInt = branch.Nid ?? 0

383 │ });

384 │ slXuat = 0;

385 │ }

386 │ }

387 │ }

388 │

389 │ // 5. Nhập trả khay đã xác nhận (từ tiêu hao -> về nguyên liệu)

390 │ var traKhay = \_context.VtthNhapXuatKhay.AsNoTracking().Where(x => x.LoaiPhieu == (int)LoaiNhapXuatVTTH.Tra &...

391 │ .Join(\_context.VtthKhay.AsNoTracking().Where(x => x.MaChiNhanhInt == MaChiNhanhInt), t => t.MaKhay, k =>...

392 │ .Join(\_context.TblHangHoa.AsNoTracking().Where(x => x.MaChiNhanhInt == MaChiNhanhInt), x => x.k.MaVatTu,...

393 │ .Where(x => x.t.XacNhanTra == true &&

394 │ x.t.NgayTra.HasValue &&

395 │ x.t.NgayTra.Value.Year == year && x.t.NgayTra.Value.Month == month &&

396 │ !(x.t.Deleted ?? false) && x.k.MaKhoNguyenLieu == maKho)

397 │ .GroupBy(x => x.k.MaVatTu)

398 │ .Select(g => new StockTonKho

399 │ {

400 │ NamThang = namThang,

401 │ MaKho = maKho,

402 │ MaHangHoa = g.Key,

403 │ GiaVon = g.Select(x => x.hh.GiaVon ?? 0).FirstOrDefault(),

404 │ SlnhapTrongKy = g.Sum(x => x.t.SoLuongTra ?? 0),

405 │ MaChiNhanh = maChiNhanh,

406 │ MaChiNhanhInt = branch.Nid ?? 0

407 │ })

408 │ .ToList();

409 │ tonKhoData.AddRange(traKhay);

410 │

411 │

412 │ // 6. Xuất kho

413 │ var xuatKho = \_context.StockXuatKho.AsNoTracking()

414 │ .Join(\_context.StockChiTietXuatKho.AsNoTracking().Where(x => x.MaChiNhanhInt == MaChiNhanhInt),

415 │ xk => xk.Ma, ctxk => ctxk.MaPhieuXuat, (xk, ctxk) => new { xk, ctxk })

416 │ .Where(x => x.xk.NgayXuat.HasValue && x.xk.NgayXuat.Value.Year == year &&

417 │ x.xk.NgayXuat.Value.Month == month &&

418 │ !(x.xk.Deleted ?? false) &&

419 │ !(x.ctxk.Deleted ?? false) &&

420 │ x.xk.MaLoaiNx == (int)LoaiNhapXuat.PhieuXuatThanhLy &&

421 │ x.xk.MaKho == maKho &&

422 │ x.xk.MaChiNhanhInt == MaChiNhanhInt)

423 │ .GroupBy(x => new { x.ctxk.MaHangHoa, x.xk.MaChiNhanh })

424 │ .Select(g => new

425 │ {

426 │ g.Key.MaHangHoa,

427 │ SLXuat = g.Sum(x => x.ctxk.SoLuong ?? 0),

428 │ g.Key.MaChiNhanh

429 │ }).ToList();

430 │

431 │ foreach (var xk in xuatKho)

432 │ {

433 │ decimal slXuat = xk.SLXuat;

434 │ while (slXuat > 0)

435 │ {

436 │ var tk = tonKhoData.Where(t => t.MaHangHoa == xk.MaHangHoa && (t.SltonCuoiKy ?? 0) > 0)

437 │ .OrderBy(t => t.MaNhapHang)

438 │ .FirstOrDefault();

439 │ if (tk != null)

440 │ {

441 │ var canTru = Math.Min(slXuat, tk.SltonCuoiKy ?? 0);

442 │ tk.SlxuatKhac = (tk.SlxuatKhac ?? 0) + canTru;

443 │ tk.SltonCuoiKy -= canTru;

444 │ slXuat -= canTru;

445 │ }

446 │ else

447 │ {

448 │ tonKhoData.Add(new StockTonKho

449 │ {

450 │ NamThang = namThang,

451 │ MaKho = maKho,

452 │ MaHangHoa = xk.MaHangHoa,

453 │ SlxuatKhac = slXuat,

454 │ SltonCuoiKy = -slXuat,

455 │ MaChiNhanh = xk.MaChiNhanh,

456 │ MaChiNhanhInt = branch.Nid ?? 0

457 │ });

458 │ slXuat = 0;

459 │ }

460 │ }

461 │ }

462 │ // Tính tồn cuối kỳ

463 │ foreach (var item in tonKhoData)

464 │ {

465 │ item.SltonCuoiKy = (item.SltonDauKy ?? 0) + (item.SlnhapTrongKy ?? 0)

466 │ - (item.SlbanTrongKy ?? 0) - (item.SlxuatKhac ?? 0)

467 │ - (item.SlxuatTieuDung ?? 0) - (item.SlxuatTra ?? 0);

468 │ }

469 │ // 7. Ghi dữ liệu

470 │ using var tran = \_context.Database.BeginTransaction();

471 │ try

472 │ {

473 │ var oldData = \_context.StockTonKho

474 │ .Where(x => x.NamThang == namThang && x.MaKho == maKho && x.MaChiNhanh == maChiNhanh)

475 │ .ToList();

476 │

477 │ \_context.BulkDelete(oldData);

478 │ \_context.SaveChanges();

479 │

480 │ \_context.BulkInsert(tonKhoData);

481 │ \_context.SaveChanges();

482 │

483 │ tran.Commit();

484 │ }

485 │ catch (Exception ex)

486 │ {

487 │ tran.Rollback();

488 │ throw new Exception($"Lỗi cập nhật tồn kho vật tư: {ex.Message}");

489 │ }

490 │ }

491 │ public void TinhTonKho\_KhoTieuHao(string namThang, int maKho, Guid maChiNhanh, int MaChiNhanhInt)

492 │ {

493 │ var currentMonth = DateTime.ParseExact(namThang + "01", "yyyyMMdd", CultureInfo.InvariantCulture);

494 │ var year = currentMonth.Year;

495 │ var month = currentMonth.Month;

496 │ var thangTruoc = currentMonth.AddMonths(-1).ToString("yyyyMM");

497 │

498 │ var branch = \_context.SysBranch.FirstOrDefault(x => x.Ma == maChiNhanh)

499 │ ?? throw new Exception("Không tìm thấy chi nhánh");

500 │

501 │ var tonKhoData = new List<StockTonKho>();

502 │

503 │ // 1. Tồn đầu kỳ

504 │ var tonTruoc = \_context.StockTonKho.AsNoTracking()

505 │ .Where(x => x.NamThang == thangTruoc && x.MaKho == maKho && x.MaChiNhanhInt == MaChiNhanhInt && (x.Slton...

506 │ .Select(x => new StockTonKho

507 │ {

508 │ NamThang = namThang,

509 │ MaKho = maKho,

510 │ MaHangHoa = x.MaHangHoa,

511 │ GiaVon = x.GiaVon,

512 │ SltonDauKy = x.SltonCuoiKy ?? 0,

513 │ SlnhapTrongKy = 0,

514 │ SlxuatTieuDung = 0,

515 │ SltonCuoiKy = x.SltonCuoiKy ?? 0,

516 │ MaChiNhanh = maChiNhanh,

517 │ MaChiNhanhInt = branch.Nid ?? 0

518 │ }).ToList();

519 │

520 │ tonKhoData.AddRange(tonTruoc);

521 │

522 │ // 2. Nhập từ khay (kho tiêu hao nhận hàng)

523 │ var nhapKhay = \_context.VtthNhapXuatKhay.AsNoTracking().Where(x => x.LoaiPhieu == (int)LoaiNhapXuatVTTH.Nhap...

524 │ .Join(\_context.VtthKhay.AsNoTracking().Where(x=>x.MaChiNhanhInt == MaChiNhanhInt), nx => nx.MaKhay, k =>...

525 │ .Join(\_context.TblHangHoa.AsNoTracking().Where(x => x.MaChiNhanhInt == MaChiNhanhInt), x => x.k.MaVatTu,...

526 │ .Where(x => x.nx.NgayNhap.HasValue &&

527 │ x.nx.NgayNhap.Value.Year == year && x.nx.NgayNhap.Value.Month == month &&

528 │ !(x.nx.Deleted ?? false) && x.k.MaKhoTieuHao == maKho)

529 │ .GroupBy(x => x.k.MaVatTu)

530 │ .Select(g => new StockTonKho{

531 │ NamThang = namThang,

532 │ MaKho = maKho,

533 │ MaHangHoa = g.Key,

534 │ GiaVon = g.Select(x => x.hh.GiaVon ?? 0).FirstOrDefault(),

535 │ SlnhapTrongKy = g.Sum(x => x.nx.SoLuongNhap ?? 0),

536 │ MaChiNhanh = maChiNhanh,

537 │ MaChiNhanhInt = branch.Nid ?? 0

538 │ })

539 │ .ToList();

540 │ tonKhoData.AddRange(nhapKhay);

541 │

542 │ // 3. Xuất từ khay (kho tiêu hao -> xuất tiêu hao)

543 │ var xuatKhay = \_context.VtthLichSuXuatKhay.AsNoTracking().Where(x => !(x.Deleted ?? false) && x.MaChiNhanhIn...

544 │ .Join(\_context.VtthKhay.AsNoTracking().Where(x=>x.MaChiNhanhInt == MaChiNhanhInt), nx => nx.MaKhay, k =>...

545 │ .Join(\_context.TblHangHoa.AsNoTracking().Where(x => x.MaChiNhanhInt == MaChiNhanhInt), x => x.k.MaVatTu,...

546 │ .Where(x => x.nx.NgayXuat.HasValue &&

547 │ x.nx.NgayXuat.Value.Year == year && x.nx.NgayXuat.Value.Month == month &&

548 │ !(x.nx.Deleted ?? false) && x.k.MaKhoTieuHao == maKho)

549 │ .GroupBy(x => x.k.MaVatTu)

550 │ .Select(g => new

551 │ {

552 │ MaHangHoa = g.Key,

553 │ SoLuong = g.Sum(x => x.nx.SoLuongXuat ?? 0),

554 │ GiaVon = g.Select(x => x.hh.GiaVon ?? 0).FirstOrDefault()

555 │ })

556 │ .ToList();

557 │

558 │ foreach (var item in xuatKhay)

559 │ {

560 │ decimal slXuat = item.SoLuong;

561 │ while (slXuat > 0)

562 │ {

563 │ var tk = tonKhoData.Where(t => t.MaHangHoa == item.MaHangHoa && (t.SltonCuoiKy ?? 0) > 0)

564 │ .OrderBy(t => t.MaNhapHang)

565 │ .FirstOrDefault();

566 │ if (tk != null)

567 │ {

568 │ var canTru = Math.Min(slXuat, tk.SltonCuoiKy ?? 0);

569 │ tk.SlxuatTieuDung = (tk.SlxuatTieuDung ?? 0) + canTru;

570 │ tk.SltonCuoiKy -= canTru;

571 │ slXuat -= canTru;

572 │ }

573 │ else

574 │ {

575 │ tonKhoData.Add(new StockTonKho

576 │ {

577 │ NamThang = namThang,

578 │ MaKho = maKho,

579 │ GiaVon = item.GiaVon,

580 │ MaHangHoa = item.MaHangHoa ?? 0,

581 │ SltonDauKy = 0,

582 │ SlnhapTrongKy = 0,

583 │ SlxuatTieuDung = item.SoLuong,

584 │ SltonCuoiKy = -item.SoLuong,

585 │ MaChiNhanh = maChiNhanh,

586 │ MaChiNhanhInt = branch.Nid ?? 0

587 │ });

588 │ slXuat = 0;

589 │ }

590 │ }

591 │ }

592 │

593 │ // 4. Trả khay đã xác nhận (giảm tồn kho ở kho tiêu hao)

594 │ var xuatTraKhay = \_context.VtthNhapXuatKhay.AsNoTracking().Where(x => x.LoaiPhieu == (int)LoaiNhapXuatVTTH.T...

595 │ .Join(\_context.VtthKhay.AsNoTracking().Where(x => x.MaChiNhanhInt == MaChiNhanhInt), t => t.MaKhay, k =>...

596 │ .Join(\_context.TblHangHoa.AsNoTracking().Where(x => x.MaChiNhanhInt == MaChiNhanhInt), x => x.k.MaVatTu,...

597 │ .Where(x => x.t.XacNhanTra == true &&

598 │ x.t.NgayTra.HasValue &&

599 │ x.t.NgayTra.Value.Year == year && x.t.NgayTra.Value.Month == month &&

600 │ !(x.t.Deleted ?? false) && x.k.MaKhoTieuHao == maKho)

601 │ .GroupBy(x => x.k.MaVatTu)

602 │ .Select(g => new

603 │ {

604 │ MaHangHoa = g.Key,

605 │ SoLuong = g.Sum(x => x.t.SoLuongTra ?? 0),

606 │ GiaVon = g.Select(x => x.hh.GiaVon ?? 0).FirstOrDefault()

607 │ })

608 │ .ToList();

609 │

610 │ foreach (var item in xuatTraKhay)

611 │ {

612 │ decimal slXuat = item.SoLuong;

613 │ while (slXuat > 0)

614 │ {

615 │ var tk = tonKhoData.Where(t => t.MaHangHoa == item.MaHangHoa && (t.SltonCuoiKy ?? 0) > 0)

616 │ .OrderBy(t => t.MaNhapHang)

617 │ .FirstOrDefault();

618 │ if (tk != null)

619 │ {

620 │ var canTru = Math.Min(slXuat, tk.SltonCuoiKy ?? 0);

621 │ tk.SlxuatTra = (tk.SlxuatTra ?? 0) + canTru;

622 │ tk.SltonCuoiKy -= canTru;

623 │ slXuat -= canTru;

624 │ }

625 │ else

626 │ {

627 │ tonKhoData.Add(new StockTonKho

628 │ {

629 │ NamThang = namThang,

630 │ MaKho = maKho,

631 │ GiaVon = item.GiaVon,

632 │ MaHangHoa = item.MaHangHoa ?? 0,

633 │ SltonDauKy = 0,

634 │ SlnhapTrongKy = 0,

635 │ SlxuatTra = item.SoLuong,

636 │ SltonCuoiKy = -item.SoLuong,

637 │ MaChiNhanh = maChiNhanh,

638 │ MaChiNhanhInt = branch.Nid ?? 0

639 │ });

640 │ slXuat = 0;

641 │ }

642 │ }

643 │ }

644 │ // Tính tồn cuối kỳ

645 │ foreach (var item in tonKhoData)

646 │ {

647 │ item.SltonCuoiKy = (item.SltonDauKy ?? 0) + (item.SlnhapTrongKy ?? 0)

648 │ - (item.SlbanTrongKy ?? 0) - (item.SlxuatKhac ?? 0)

649 │ - (item.SlxuatTieuDung ?? 0) - (item.SlxuatTra ?? 0);

650 │ }

651 │ // 5. Ghi dữ liệu

652 │ using var tran = \_context.Database.BeginTransaction();

653 │ try

654 │ {

655 │ var oldData = \_context.StockTonKho

656 │ .Where(x => x.NamThang == namThang && x.MaKho == maKho && x.MaChiNhanh == maChiNhanh)

657 │ .ToList();

658 │

659 │ \_context.BulkDelete(oldData);

660 │ \_context.SaveChanges();

661 │

662 │ \_context.BulkInsert(tonKhoData);

663 │ \_context.SaveChanges();

664 │

665 │ tran.Commit();

666 │ }

667 │ catch (Exception ex)

668 │ {

669 │ tran.Rollback();

670 │ throw new Exception($"Lỗi cập nhật tồn kho vật tư: {ex.Message}");

671 │ }

672 │ }

673 │

674 │ public void TinhTonKhoBanHang(int MaKho, string NamThang)

675 │ {

676 │ var kho = \_context.StockDanhMucKho.FirstOrDefault(x => x.Ma == MaKho);

677 │ if (kho == null)

678 │ {

679 │ throw new Exception("Không tìm thấy kho");

680 │ }

681 │ switch (kho.MaLoaiKho)

682 │ {

683 │ case (int)LoaiKho.KhoTieuHao: TinhTonKhoBanHang\_KhoTieuHao(NamThang, MaKho); break;

684 │ case (int)LoaiKho.KhoNguyenVatLieu: TinhTonKhoBanHang\_KhoNguyenLieu(NamThang, MaKho); break;

685 │ case (int)LoaiKho.KhoBanHang: TinhTonKhoBanHang\_KhoBanHang(NamThang, MaKho); break;

686 │ }

687 │

688 │ }

689 │ public void TinhTonKhoBanHang\_KhoBanHang(string namThang, int maKho)

690 │ {

691 │ var nam = int.Parse(namThang.Substring(0, 4));

692 │ var thang = int.Parse(namThang.Substring(4, 2));

693 │ var dtThang = new DateTime(nam, thang, 1);

694 │ var thangTruoc = dtThang.AddMonths(-1).ToString("yyyyMM");

695 │ var kho = \_context.StockDanhMucKho.FirstOrDefault(x => x.Ma == maKho);

696 │ var maChiNhanh = kho.MaChiNhanh ?? Utility.defaultUID;

697 │ var maChiNhanhInt = \_context.SysBranch.FirstOrDefault(x => x.Ma == maChiNhanh)?.Nid ?? 0;

698 │

699 │ var tonKhoMoi = new List<StockTonKhoBanHang>();

700 │ // Tạo số lượng đầu kỳ từ cuối kỳ tháng trước

701 │ var dauKyList = \_context.StockTonKhoBanHang.AsNoTracking()

702 │ .Where(x => x.NamThang == thangTruoc && x.MaKho == maKho && x.SltonCuoiKy != 0)

703 │ .Select(x => new StockTonKhoBanHang

704 │ {

705 │ MaChiNhanh = maChiNhanh,

706 │ MaChiNhanhInt = maChiNhanhInt,

707 │ NamThang = namThang,

708 │ MaKho = maKho,

709 │ MaHangHoa = x.MaHangHoa,

710 │ SltonDauKy = x.SltonCuoiKy ?? 0,

711 │ SlnhapTrongKy = 0,

712 │ SlxuatKhac = 0,

713 │ SlxuatTieuDung = 0,

714 │ SlxuatTra = 0,

715 │ SlbanTrongKy = 0,

716 │ SltonCuoiKy = x.SltonCuoiKy ?? 0

717 │ }).ToList();

718 │ tonKhoMoi.AddRange(dauKyList);

719 │

720 │ // Cập nhật hoặc thêm số lượng nhập trong kỳ từ phiếu nhập

721 │ var nhapHang = (from nh in \_context.StockNhapHang.AsNoTracking().Where(x => x.MaChiNhanhInt == maChiNhanhInt...

722 │ join ct in \_context.StockChiTietNhapHang.AsNoTracking().Where(x => x.MaChiNhanhInt == maChiN...

723 │ where nh.MaKho == maKho && nh.NgayNhapHang.HasValue && nh.NgayNhapHang.Value.Year == nam && ...

724 │ && !(nh.Deleted ?? false) && !(ct.Deleted ?? false) && ct.SoLuong != 0

725 │ group ct by ct.MaHangHoa into g

726 │ select new { MaHangHoa = g.Key, SLNhap = g.Sum(x => x.SoLuong ?? 0) }).ToList();

727 │

728 │ foreach (var item in nhapHang)

729 │ {

730 │ var tonKho = tonKhoMoi.FirstOrDefault(x => x.NamThang == namThang && x.MaKho == maKho && x.MaHangHoa == ...

731 │ if (tonKho != null)

732 │ tonKho.SlnhapTrongKy += item.SLNhap;

733 │ else

734 │ tonKhoMoi.Add(new StockTonKhoBanHang

735 │ {

736 │ NamThang = namThang,

737 │ MaKho = maKho,

738 │ MaHangHoa = item.MaHangHoa ?? 0,

739 │ SltonDauKy = 0,

740 │ SlxuatKhac = 0,

741 │ SlxuatTieuDung = 0,

742 │ SlxuatTra = 0,

743 │ SlbanTrongKy = 0,

744 │ SltonCuoiKy = 0,

745 │ SlnhapTrongKy = item.SLNhap,

746 │ MaChiNhanh = maChiNhanh,

747 │ MaChiNhanhInt = maChiNhanhInt,

748 │ });

749 │ }

750 │

751 │ // Nhập từ chuyển kho đến

752 │ var nhapChuyenKho = (from xk in \_context.StockXuatKho.AsNoTracking().Where(x=>x.MaChiNhanhInt == maChiNhanhI...

753 │ join ct in \_context.StockChiTietXuatKho.AsNoTracking().Where(x => x.MaChiNhanhInt == ma...

754 │ where xk.MaKhoChuyenDen == maKho && xk.NgayXuat.HasValue && xk.NgayXuat.Value.Year == n...

755 │ && !(xk.Deleted ?? false) && !(ct.Deleted ?? false) && ct.SoLuong != 0

756 │ group ct by ct.MaHangHoa into g

757 │ select new { MaHangHoa = g.Key, SLNhap = g.Sum(x => x.SoLuong ?? 0) }).ToList();

758 │

759 │ foreach (var item in nhapChuyenKho)

760 │ {

761 │ var tonKho = tonKhoMoi.FirstOrDefault(x => x.NamThang == namThang && x.MaKho == maKho && x.MaHangHoa == ...

762 │ if (tonKho != null)

763 │ tonKho.SlnhapTrongKy += item.SLNhap;

764 │ else

765 │ tonKhoMoi.Add(new StockTonKhoBanHang

766 │ {

767 │ NamThang = namThang,

768 │ MaKho = maKho,

769 │ MaHangHoa = item.MaHangHoa ?? 0,

770 │ SlnhapTrongKy = item.SLNhap,

771 │ SltonDauKy = 0,

772 │ SlxuatKhac = 0,

773 │ SlxuatTieuDung = 0,

774 │ SlxuatTra = 0,

775 │ SlbanTrongKy = 0,

776 │ SltonCuoiKy = 0,

777 │ MaChiNhanh = maChiNhanh,

778 │ MaChiNhanhInt = maChiNhanhInt,

779 │ });

780 │ }

781 │

782 │ // Nhập trả hàng

783 │ var nhapTraHang = (from bh in \_context.OpsHoaDon.AsNoTracking().Where(x => x.MaChiNhanhInt == maChiNhanhInt)

784 │ join ct in \_context.OpsChiTietHoaDon.AsNoTracking().Where(x => x.MaChiNhanhInt == maChiNh...

785 │ where bh.MaKho == maKho && bh.NgayHoaDon.Year == nam && bh.NgayHoaDon.Month == thang

786 │ && !(bh.Deleted ?? false) && !(ct.Deleted ?? false) && ct.SoLuongTraLai != 0

787 │ group ct by ct.MaHangHoa into g

788 │ select new { MaHangHoa = g.Key, SLTra = g.Sum(x => x.SoLuongTraLai ?? 0) }).ToList();

789 │

790 │ foreach (var item in nhapTraHang)

791 │ {

792 │ var tonKho = tonKhoMoi.FirstOrDefault(x => x.NamThang == namThang && x.MaKho == maKho && x.MaHangHoa == ...

793 │ if (tonKho != null)

794 │ tonKho.SlnhapTrongKy += item.SLTra;

795 │ else

796 │ tonKhoMoi.Add(new StockTonKhoBanHang

797 │ {

798 │ NamThang = namThang,

799 │ MaKho = maKho,

800 │ MaHangHoa = item.MaHangHoa ?? 0,

801 │ SlnhapTrongKy = item.SLTra,

802 │ SltonDauKy = 0,

803 │ SlxuatKhac = 0,

804 │ SlxuatTieuDung = 0,

805 │ SlxuatTra = 0,

806 │ SlbanTrongKy = 0,

807 │ SltonCuoiKy = 0,

808 │ MaChiNhanh = maChiNhanh,

809 │ MaChiNhanhInt = maChiNhanhInt,

810 │ });

811 │ }

812 │

813 │ // Bán trong kỳ

814 │ var banTrongKy = (from bh in \_context.OpsHoaDon.AsNoTracking().Where(x => x.MaChiNhanhInt == maChiNhanhInt)

815 │ join ct in \_context.OpsChiTietHoaDon.AsNoTracking().Where(x => x.MaChiNhanhInt == maChiNha...

816 │ where bh.MaKho == maKho && bh.NgayHoaDon.Year == nam && bh.NgayHoaDon.Month == thang

817 │ && !(bh.Deleted ?? false) && !(ct.Deleted ?? false) && ct.SoLuong != 0

818 │ group ct by ct.MaHangHoa into g

819 │ select new { MaHangHoa = g.Key, SLBan = g.Sum(x => x.SoLuong ?? 0) }).ToList();

820 │

821 │ foreach (var item in banTrongKy)

822 │ {

823 │ var tonKho = tonKhoMoi.FirstOrDefault(x => x.NamThang == namThang && x.MaKho == maKho && x.MaHangHoa == ...

824 │ if (tonKho != null)

825 │ tonKho.SlbanTrongKy += item.SLBan;

826 │ else

827 │ tonKhoMoi.Add(new StockTonKhoBanHang

828 │ {

829 │ NamThang = namThang,

830 │ MaKho = maKho,

831 │ MaHangHoa = item.MaHangHoa ?? 0,

832 │ SlbanTrongKy = item.SLBan,

833 │ SltonDauKy = 0,

834 │ SlnhapTrongKy = 0,

835 │ SlxuatKhac = 0,

836 │ SlxuatTieuDung = 0,

837 │ SlxuatTra = 0,

838 │ SltonCuoiKy = 0,

839 │ MaChiNhanh = maChiNhanh,

840 │ MaChiNhanhInt = maChiNhanhInt,

841 │ });

842 │ }

843 │

844 │ // Xuất tiêu dùng

845 │ var xuatTieuDung = (from xk in \_context.StockXuatKho.AsNoTracking().Where(x => x.MaChiNhanhInt == maChiNhanh...

846 │ join ct in \_context.StockChiTietXuatKho.AsNoTracking().Where(x => x.MaChiNhanhInt == maC...

847 │ where xk.MaKho == maKho && xk.NgayXuat.HasValue && xk.NgayXuat.Value.Year == nam && xk.N...

848 │ && !(xk.Deleted ?? false) && !(ct.Deleted ?? false) && ct.SoLuong != 0

849 │ && xk.MaKhoChuyenDen == null && xk.MaLoaiNx == 4

850 │ group ct by ct.MaHangHoa into g

851 │ select new { MaHangHoa = g.Key, SLXuat = g.Sum(x => x.SoLuong ?? 0) }).ToList();

852 │

853 │ foreach (var item in xuatTieuDung)

854 │ {

855 │ var tonKho = tonKhoMoi.FirstOrDefault(x => x.NamThang == namThang && x.MaKho == maKho && x.MaHangHoa == ...

856 │ if (tonKho != null)

857 │ tonKho.SlxuatTieuDung += item.SLXuat;

858 │ else

859 │ tonKhoMoi.Add(new StockTonKhoBanHang

860 │ {

861 │ NamThang = namThang,

862 │ MaKho = maKho,

863 │ MaHangHoa = item.MaHangHoa ?? 0,

864 │ SlxuatTieuDung = item.SLXuat,

865 │ SltonDauKy = 0,

866 │ SlnhapTrongKy = 0,

867 │ SlxuatKhac = 0,

868 │ SlxuatTra = 0,

869 │ SlbanTrongKy = 0,

870 │ SltonCuoiKy = 0,

871 │ MaChiNhanh = maChiNhanh,

872 │ MaChiNhanhInt = maChiNhanhInt,

873 │ });

874 │ }

875 │

876 │ // Xuất chuyển kho và kiểm kê

877 │ var xuatKhac = (from xk in \_context.StockXuatKho.AsNoTracking().Where(x => x.MaChiNhanhInt == maChiNhanhInt)

878 │ join ct in \_context.StockChiTietXuatKho.Where(x=>x.MaChiNhanhInt == maChiNhanhInt) on xk.Ma ...

879 │ where xk.MaKho == maKho && xk.NgayXuat.HasValue && xk.NgayXuat.Value.Year == nam && xk.NgayX...

880 │ && !(xk.Deleted ?? false) && !(ct.Deleted ?? false) && ct.SoLuong != 0

881 │ && (xk.MaLoaiNx == (int)LoaiNhapXuat.PhieuXuatKiemKe || xk.MaLoaiNx == (int)LoaiNhapXu...

882 │ group ct by ct.MaHangHoa into g

883 │ select new { MaHangHoa = g.Key, SLXuat = g.Sum(x => x.SoLuong ?? 0) }).ToList();

884 │

885 │ foreach (var item in xuatKhac)

886 │ {

887 │ var tonKho = tonKhoMoi.FirstOrDefault(x => x.NamThang == namThang && x.MaKho == maKho && x.MaHangHoa == ...

888 │ if (tonKho != null)

889 │ tonKho.SlxuatKhac += item.SLXuat;

890 │ else

891 │ tonKhoMoi.Add(new StockTonKhoBanHang

892 │ {

893 │ NamThang = namThang,

894 │ MaKho = maKho,

895 │ MaHangHoa = item.MaHangHoa ?? 0,

896 │ SlxuatKhac = item.SLXuat,

897 │ SltonDauKy = 0,

898 │ SlnhapTrongKy = 0,

899 │ SlxuatTieuDung = 0,

900 │ SlxuatTra = 0,

901 │ SlbanTrongKy = 0,

902 │ SltonCuoiKy = 0,

903 │ MaChiNhanh = maChiNhanh,

904 │ MaChiNhanhInt = maChiNhanhInt,

905 │ });

906 │ }

907 │

908 │ // Xuất trả nhà cung cấp

909 │ var xuatTra = (from xk in \_context.StockXuatKho.AsNoTracking().Where(x=>x.MaChiNhanhInt == maChiNhanhInt)

910 │ join ct in \_context.StockChiTietXuatKho.AsNoTracking().Where(x => x.MaChiNhanhInt == maChiNha...

911 │ where xk.MaKho == maKho && xk.NgayXuat.HasValue && xk.NgayXuat.Value.Year == nam && xk.NgayXu...

912 │ && !(xk.Deleted ?? false) && !(ct.Deleted ?? false) && ct.SoLuong != 0

913 │ && xk.MaKhoChuyenDen == null && xk.MaLoaiNx == (int)LoaiNhapXuat.PhieuXuatTraNhaCC

914 │ group ct by ct.MaHangHoa into g

915 │ select new { MaHangHoa = g.Key, SLXuat = g.Sum(x => x.SoLuong ?? 0) }).ToList();

916 │

917 │ foreach (var item in xuatTra)

918 │ {

919 │ var tonKho = tonKhoMoi.FirstOrDefault(x => x.NamThang == namThang && x.MaKho == maKho && x.MaHangHoa == ...

920 │ if (tonKho != null)

921 │ tonKho.SlxuatTra += item.SLXuat;

922 │ else

923 │ tonKhoMoi.Add(new StockTonKhoBanHang

924 │ {

925 │ NamThang = namThang,

926 │ MaKho = maKho,

927 │ MaHangHoa = item.MaHangHoa ?? 0,

928 │ SlxuatTra = item.SLXuat,

929 │ SltonDauKy = 0,

930 │ SlnhapTrongKy = 0,

931 │ SlxuatKhac = 0,

932 │ SlxuatTieuDung = 0,

933 │ SlbanTrongKy = 0,

934 │ SltonCuoiKy = 0,

935 │ MaChiNhanh = maChiNhanh,

936 │ MaChiNhanhInt = maChiNhanhInt,

937 │ });

938 │ }

939 │

940 │

941 │ // Tính tồn cuối kỳ

942 │ foreach (var item in tonKhoMoi)

943 │ {

944 │ item.SltonCuoiKy = (item.SltonDauKy ?? 0) + (item.SlnhapTrongKy ?? 0)

945 │ - (item.SlbanTrongKy ?? 0) - (item.SlxuatKhac ?? 0)

946 │ - (item.SlxuatTieuDung ?? 0) - (item.SlxuatTra ?? 0);

947 │ }

948 │

949 │ using var transaction = \_context.Database.BeginTransaction();

950 │ try

951 │ {

952 │ var tonKhoHienTai = \_context.StockTonKhoBanHang.Where(x => x.NamThang == namThang && x.MaKho == maKho).T...

953 │ \_context.BulkDelete(tonKhoHienTai);

954 │ \_context.SaveChanges();

955 │ \_context.BulkInsert(tonKhoMoi);

956 │ \_context.SaveChanges();

957 │ transaction.Commit();

958 │ }

959 │ catch (Exception ex)

960 │ {

961 │ transaction.Rollback();

962 │ throw new Exception($"❌ Lỗi khi cập nhật tồn kho: Tháng={namThang}, Kho={maKho}\n{ex.Message}...

963 │

964 │ }

965 │

966 │ }

967 │ public void TinhTonKhoBanHang\_KhoNguyenLieu(string namThang, int maKho )

968 │ {

969 │ var nam = int.Parse(namThang.Substring(0, 4));

970 │ var thang = int.Parse(namThang.Substring(4, 2));

971 │ var dtThang = new DateTime(nam, thang, 1);

972 │ var thangTruoc = dtThang.AddMonths(-1).ToString("yyyyMM");

973 │ var kho = \_context.StockDanhMucKho.FirstOrDefault(x => x.Ma == maKho);

974 │ var maChiNhanh = kho.MaChiNhanh ?? Utility.defaultUID;

975 │ var maChiNhanhInt = \_context.SysBranch.FirstOrDefault(x => x.Ma == maChiNhanh)?.Nid ?? 0;

976 │

977 │ var tonKhoData = new List<StockTonKhoBanHang>();

978 │ // Tạo số lượng đầu kỳ từ cuối kỳ tháng trước

979 │ var dauKyList = \_context.StockTonKhoBanHang.AsNoTracking()

980 │ .Where(x => x.MaChiNhanhInt == maChiNhanhInt && x.NamThang == thangTruoc && x.MaKho == maKho && x.SltonC...

981 │ .Select(x => new StockTonKhoBanHang

982 │ {

983 │ MaChiNhanh = maChiNhanh,

984 │ MaChiNhanhInt = maChiNhanhInt,

985 │ NamThang = namThang,

986 │ MaKho = maKho,

987 │ MaHangHoa = x.MaHangHoa,

988 │ SltonDauKy = x.SltonCuoiKy ?? 0,

989 │ SlnhapTrongKy = 0,

990 │ SlxuatKhac = 0,

991 │ SlxuatTieuDung = 0,

992 │ SlxuatTra = 0,

993 │ SlbanTrongKy = 0,

994 │ SltonCuoiKy = x.SltonCuoiKy ?? 0

995 │ }).ToList();

996 │ tonKhoData.AddRange(dauKyList);

997 │

998 │ // Cập nhật hoặc thêm số lượng nhập trong kỳ từ phiếu nhập

999 │ var nhapHang = (from nh in \_context.StockNhapHang.AsNoTracking().Where(x=>x.MaChiNhanhInt == maChiNhanhInt)

1000 │ join ct in \_context.StockChiTietNhapHang.AsNoTracking().Where(x => x.MaChiNhanhInt == maChiN...

1001 │ where nh.MaKho == maKho && nh.NgayNhapHang.HasValue && nh.NgayNhapHang.Value.Year == nam && ...

1002 │ && !(nh.Deleted ?? false) && !(ct.Deleted ?? false) && ct.SoLuong != 0

1003 │ group ct by ct.MaHangHoa into g

1004 │ select new { MaHangHoa = g.Key, SLNhap = g.Sum(x => x.SoLuong ?? 0) }).ToList();

1005 │

1006 │ foreach (var item in nhapHang)

1007 │ {

1008 │ var tonKho = tonKhoData.FirstOrDefault(x => x.NamThang == namThang && x.MaKho == maKho && x.MaHangHoa ==...

1009 │ if (tonKho != null)

1010 │ tonKho.SlnhapTrongKy += item.SLNhap;

1011 │ else

1012 │ tonKhoData.Add(new StockTonKhoBanHang

1013 │ {

1014 │ NamThang = namThang,

1015 │ MaKho = maKho,

1016 │ MaHangHoa = item.MaHangHoa ?? 0,

1017 │ SltonDauKy = 0,

1018 │ SlxuatKhac = 0,

1019 │ SlxuatTieuDung = 0,

1020 │ SlxuatTra = 0,

1021 │ SlbanTrongKy = 0,

1022 │ SltonCuoiKy = 0,

1023 │ SlnhapTrongKy = item.SLNhap,

1024 │ MaChiNhanh = maChiNhanh,

1025 │ MaChiNhanhInt = maChiNhanhInt,

1026 │ });

1027 │ }

1028 │

1029 │ // 4. Xuất kho nguyên liệu khi nhập khay (đã chuyển sang tiêu hao)

1030 │ var xuatNguyenLieu = \_context.VtthNhapXuatKhay.AsNoTracking().Where(x => x.MaChiNhanhInt == maChiNhanhInt &&...

1031 │ .Join(\_context.VtthKhay.AsNoTracking().Where(x => x.MaChiNhanhInt == maChiNhanhInt), nx => nx.MaKhay, k ...

1032 │ .Where(x => x.nx.NgayNhap.HasValue &&

1033 │ x.nx.NgayNhap.Value.Year == nam && x.nx.NgayNhap.Value.Month == thang &&

1034 │ !(x.nx.Deleted ?? false) && x.k.MaKhoNguyenLieu == maKho)

1035 │ .GroupBy(x => x.k.MaVatTu)

1036 │ .Select(g => new

1037 │ {

1038 │ MaHangHoa = g.Key,

1039 │ SoLuong = g.Sum(x => x.nx.SoLuongNhap ?? 0),

1040 │ }).ToList();

1041 │

1042 │ foreach (var item in xuatNguyenLieu)

1043 │ {

1044 │ var record = tonKhoData.FirstOrDefault(x => x.MaHangHoa == item.MaHangHoa);

1045 │ if (record != null)

1046 │ {

1047 │ record.SlxuatKhac = (record.SlxuatKhac ?? 0) + item.SoLuong;

1048 │ record.SltonCuoiKy = (record.SltonCuoiKy ?? 0) - item.SoLuong;

1049 │ }

1050 │ else

1051 │ {

1052 │ tonKhoData.Add(new StockTonKhoBanHang

1053 │ {

1054 │ NamThang = namThang,

1055 │ MaKho = maKho,

1056 │ MaHangHoa = item.MaHangHoa ?? 0,

1057 │ SltonDauKy = 0,

1058 │ SlnhapTrongKy = 0,

1059 │ SlxuatKhac = item.SoLuong,

1060 │ SltonCuoiKy = -item.SoLuong,

1061 │ MaChiNhanh = maChiNhanh,

1062 │ MaChiNhanhInt = maChiNhanhInt

1063 │ });

1064 │ }

1065 │ }

1066 │

1067 │ // 5. Nhập trả khay đã xác nhận (từ tiêu hao -> về nguyên liệu)

1068 │ var traKhay = \_context.VtthNhapXuatKhay.AsNoTracking().Where(x => x.MaChiNhanhInt == maChiNhanhInt && x.Loai...

1069 │ .Join(\_context.VtthKhay.AsNoTracking().Where(x=>x.MaChiNhanhInt == maChiNhanhInt), t => t.MaKhay, k => k...

1070 │ .Where(x => x.t.XacNhanTra == true &&

1071 │ x.t.NgayTra.HasValue &&

1072 │ x.t.NgayTra.Value.Year == nam && x.t.NgayTra.Value.Month == thang &&

1073 │ !(x.t.Deleted ?? false) && x.k.MaKhoNguyenLieu == maKho)

1074 │ .GroupBy(x => x.k.MaVatTu)

1075 │ .Select(g => new

1076 │ {

1077 │ MaHangHoa = g.Key,

1078 │ SoLuong = g.Sum(x => x.t.SoLuongTra ?? 0),

1079 │ }).ToList();

1080 │

1081 │ foreach (var item in traKhay)

1082 │ {

1083 │ var record = tonKhoData.FirstOrDefault(x => x.MaHangHoa == item.MaHangHoa);

1084 │ if (record != null)

1085 │ {

1086 │ record.SlnhapTrongKy = (record.SlnhapTrongKy ?? 0) + item.SoLuong;

1087 │ }

1088 │ else

1089 │ {

1090 │ tonKhoData.Add(new StockTonKhoBanHang

1091 │ {

1092 │ NamThang = namThang,

1093 │ MaKho = maKho,

1094 │ MaHangHoa = item.MaHangHoa ?? 0,

1095 │ SlnhapTrongKy = item.SoLuong,

1096 │ SltonCuoiKy = item.SoLuong,

1097 │ MaChiNhanh = maChiNhanh,

1098 │ MaChiNhanhInt = maChiNhanhInt

1099 │ });

1100 │ }

1101 │ }

1102 │ // 6. Xuất kho

1103 │ var xuatKhac = (from xk in \_context.StockXuatKho.Where(x => x.MaChiNhanhInt == maChiNhanhInt)

1104 │ join ct in \_context.StockChiTietXuatKho.Where(x => x.MaChiNhanhInt == maChiNhanhInt) on xk.M...

1105 │ where xk.MaKho == maKho && xk.NgayXuat.HasValue && xk.NgayXuat.Value.Year == nam && xk.NgayX...

1106 │ && !(xk.Deleted ?? false) && !(ct.Deleted ?? false) && ct.SoLuong != 0

1107 │ && xk.MaLoaiNx == (int)LoaiNhapXuat.PhieuXuatThanhLy

1108 │ group ct by ct.MaHangHoa into g

1109 │ select new { MaHangHoa = g.Key, SLXuat = g.Sum(x => x.SoLuong ?? 0) }).ToList();

1110 │

1111 │ foreach (var item in xuatKhac)

1112 │ {

1113 │ var tonKho = tonKhoData.FirstOrDefault(x => x.NamThang == namThang && x.MaKho == maKho && x.MaHangHoa ==...

1114 │ if (tonKho != null)

1115 │ tonKho.SlxuatKhac += item.SLXuat;

1116 │ else

1117 │ tonKhoData.Add(new StockTonKhoBanHang

1118 │ {

1119 │ NamThang = namThang,

1120 │ MaKho = maKho,

1121 │ MaHangHoa = item.MaHangHoa ?? 0,

1122 │ SlxuatKhac = item.SLXuat,

1123 │ SltonDauKy = 0,

1124 │ SlnhapTrongKy = 0,

1125 │ SlxuatTieuDung = 0,

1126 │ SlxuatTra = 0,

1127 │ SlbanTrongKy = 0,

1128 │ SltonCuoiKy = 0,

1129 │ MaChiNhanh = maChiNhanh,

1130 │ MaChiNhanhInt = maChiNhanhInt,

1131 │ });

1132 │ }

1133 │ // Tính tồn cuối kỳ

1134 │ foreach (var item in tonKhoData)

1135 │ {

1136 │ item.SltonCuoiKy = (item.SltonDauKy ?? 0) + (item.SlnhapTrongKy ?? 0)

1137 │ - (item.SlbanTrongKy ?? 0) - (item.SlxuatKhac ?? 0)

1138 │ - (item.SlxuatTieuDung ?? 0) - (item.SlxuatTra ?? 0);

1139 │ }

1140 │

1141 │ using var transaction = \_context.Database.BeginTransaction();

1142 │ try

1143 │ {

1144 │ var tonKhoHienTai = \_context.StockTonKhoBanHang.Where(x => x.NamThang == namThang && x.MaKho == maKho).T...

1145 │ \_context.BulkDelete(tonKhoHienTai);

1146 │ \_context.SaveChanges();

1147 │ \_context.BulkInsert(tonKhoData);

1148 │ \_context.SaveChanges();

1149 │ transaction.Commit();

1150 │ }

1151 │ catch (Exception ex)

1152 │ {

1153 │ transaction.Rollback();

1154 │ throw new Exception($" Lỗi khi cập nhật tồn kho: Tháng={namThang}, Kho={maKho}\n{ex.Message}");

1155 │

1156 │ }

1157 │

1158 │ }

1159 │ public void TinhTonKhoBanHang\_KhoTieuHao(string namThang, int maKho)

1160 │ {

1161 │ var nam = int.Parse(namThang.Substring(0, 4));

1162 │ var thang = int.Parse(namThang.Substring(4, 2));

1163 │ var dtThang = new DateTime(nam, thang, 1);

1164 │ var thangTruoc = dtThang.AddMonths(-1).ToString("yyyyMM");

1165 │ var kho = \_context.StockDanhMucKho.FirstOrDefault(x => x.Ma == maKho);

1166 │ var maChiNhanh = kho.MaChiNhanh ?? Utility.defaultUID;

1167 │ var maChiNhanhInt = \_context.SysBranch.FirstOrDefault(x => x.Ma == maChiNhanh)?.Nid ?? 0;

1168 │

1169 │ var tonKhoData = new List<StockTonKhoBanHang>();

1170 │ // Tạo số lượng đầu kỳ từ cuối kỳ tháng trước

1171 │ var dauKyList = \_context.StockTonKhoBanHang.AsNoTracking()

1172 │ .Where(x => x.MaChiNhanhInt == maChiNhanhInt && x.NamThang == thangTruoc && x.MaKho == maKho && x.SltonC...

1173 │ .Select(x => new StockTonKhoBanHang

1174 │ {

1175 │ MaChiNhanh = maChiNhanh,

1176 │ MaChiNhanhInt = maChiNhanhInt,

1177 │ NamThang = namThang,

1178 │ MaKho = maKho,

1179 │ MaHangHoa = x.MaHangHoa,

1180 │ SltonDauKy = x.SltonCuoiKy ?? 0,

1181 │ SlnhapTrongKy = 0,

1182 │ SlxuatKhac = 0,

1183 │ SlxuatTieuDung = 0,

1184 │ SlxuatTra = 0,

1185 │ SlbanTrongKy = 0,

1186 │ SltonCuoiKy = x.SltonCuoiKy ?? 0

1187 │ }).ToList();

1188 │ tonKhoData.AddRange(dauKyList);

1189 │

1190 │ // 2. Nhập từ khay (kho tiêu hao nhận hàng)

1191 │ var nhapKhay = \_context.VtthNhapXuatKhay.AsNoTracking().Where(x => x.LoaiPhieu == (int)LoaiNhapXuatVTTH.Nhap...

1192 │ .Join(\_context.VtthKhay.AsNoTracking().Where(x=> x.MaChiNhanhInt == maChiNhanhInt), nx => nx.MaKhay, k =...

1193 │ .Where(x => x.nx.NgayNhap.HasValue &&

1194 │ x.nx.NgayNhap.Value.Year == nam && x.nx.NgayNhap.Value.Month == thang &&

1195 │ !(x.nx.Deleted ?? false) && x.k.MaKhoTieuHao == maKho)

1196 │ .GroupBy(x => x.k.MaVatTu)

1197 │ .Select(g => new {

1198 │ MaHangHoa = g.Key,

1199 │ SoLuong = g.Sum(x => x.nx.SoLuongNhap ?? 0),

1200 │ })

1201 │ .ToList();

1202 │

1203 │ foreach (var item in nhapKhay)

1204 │ {

1205 │ var record = tonKhoData.FirstOrDefault(x => x.MaHangHoa == item.MaHangHoa);

1206 │ if (record != null)

1207 │ {

1208 │ record.SlnhapTrongKy = (record.SlnhapTrongKy ?? 0) + item.SoLuong;

1209 │ }

1210 │ else

1211 │ {

1212 │ tonKhoData.Add(new StockTonKhoBanHang

1213 │ {

1214 │ NamThang = namThang,

1215 │ MaKho = maKho,

1216 │ MaHangHoa = item.MaHangHoa ?? 0,

1217 │ SlxuatKhac = 0,

1218 │ SltonDauKy = 0,

1219 │ SlnhapTrongKy = item.SoLuong,

1220 │ SlxuatTieuDung = 0,

1221 │ SlxuatTra = 0,

1222 │ SlbanTrongKy = 0,

1223 │ SltonCuoiKy = 0,

1224 │ MaChiNhanh = maChiNhanh,

1225 │ MaChiNhanhInt = maChiNhanhInt,

1226 │ });

1227 │ }

1228 │ }

1229 │

1230 │ // 3. Xuất từ khay (kho tiêu hao -> xuất tiêu hao)

1231 │ var xuatKhay = \_context.VtthLichSuXuatKhay.AsNoTracking().Where(x => x.MaChiNhanhInt == maChiNhanhInt && !(x...

1232 │ .Join(\_context.VtthKhay.AsNoTracking().Where(x => x.MaChiNhanhInt == maChiNhanhInt), nx => nx.MaKhay, k ...

1233 │ .Where(x => x.nx.NgayXuat.HasValue &&

1234 │ x.nx.NgayXuat.Value.Year == nam && x.nx.NgayXuat.Value.Month == thang &&

1235 │ !(x.nx.Deleted ?? false) && x.k.MaKhoTieuHao == maKho)

1236 │ .GroupBy(x => x.k.MaVatTu)

1237 │ .Select(g => new

1238 │ {

1239 │ MaHangHoa = g.Key,

1240 │ SoLuong = g.Sum(x => x.nx.SoLuongXuat ?? 0),

1241 │ })

1242 │ .ToList();

1243 │

1244 │ foreach (var item in xuatKhay)

1245 │ {

1246 │ var tonKho = tonKhoData.FirstOrDefault(x => x.NamThang == namThang && x.MaKho == maKho && x.MaHangHoa ==...

1247 │ if (tonKho != null)

1248 │ tonKho.SlxuatTieuDung = (tonKho.SlxuatTieuDung ?? 0) + item.SoLuong;

1249 │ else

1250 │ tonKhoData.Add(new StockTonKhoBanHang

1251 │ {

1252 │ NamThang = namThang,

1253 │ MaKho = maKho,

1254 │ MaHangHoa = item.MaHangHoa ?? 0,

1255 │ SlxuatKhac = 0,

1256 │ SltonDauKy = 0,

1257 │ SlnhapTrongKy = 0,

1258 │ SlxuatTieuDung = item.SoLuong,

1259 │ SlxuatTra = 0,

1260 │ SlbanTrongKy = 0,

1261 │ SltonCuoiKy = 0,

1262 │ MaChiNhanh = maChiNhanh,

1263 │ MaChiNhanhInt = maChiNhanhInt,

1264 │ });

1265 │ }

1266 │

1267 │

1268 │ // 4. Trả khay đã xác nhận (giảm tồn kho ở kho tiêu hao)

1269 │ var xuatTraKhay = \_context.VtthNhapXuatKhay.AsNoTracking().Where(x => x.MaChiNhanhInt == maChiNhanhInt && x....

1270 │ .Join(\_context.VtthKhay.AsNoTracking().Where(x => x.MaChiNhanhInt == maChiNhanhInt), t => t.MaKhay, k =>...

1271 │ .Where(x => x.t.XacNhanTra == true &&

1272 │ x.t.NgayTra.HasValue &&

1273 │ x.t.NgayTra.Value.Year == nam && x.t.NgayTra.Value.Month == thang &&

1274 │ !(x.t.Deleted ?? false) && x.k.MaKhoTieuHao == maKho)

1275 │ .GroupBy(x => x.k.MaVatTu)

1276 │ .Select(g => new

1277 │ {

1278 │ MaHangHoa = g.Key,

1279 │ SoLuong = g.Sum(x => x.t.SoLuongTra ?? 0),

1280 │ })

1281 │ .ToList();

1282 │

1283 │ foreach (var item in xuatTraKhay)

1284 │ {

1285 │ var tonKho = tonKhoData.FirstOrDefault(x => x.NamThang == namThang && x.MaKho == maKho && x.MaHangHoa ==...

1286 │ if (tonKho != null)

1287 │ tonKho.SlxuatKhac += item.SoLuong;

1288 │ else

1289 │ tonKhoData.Add(new StockTonKhoBanHang

1290 │ {

1291 │ NamThang = namThang,

1292 │ MaKho = maKho,

1293 │ MaHangHoa = item.MaHangHoa ?? 0,

1294 │ SlxuatKhac = 0,

1295 │ SltonDauKy = 0,

1296 │ SlnhapTrongKy = 0,

1297 │ SlxuatTieuDung = 0,

1298 │ SlxuatTra = item.SoLuong,

1299 │ SlbanTrongKy = 0,

1300 │ SltonCuoiKy = 0,

1301 │ MaChiNhanh = maChiNhanh,

1302 │ MaChiNhanhInt = maChiNhanhInt,

1303 │ });

1304 │ }

1305 │

1306 │

1307 │ // Tính tồn cuối kỳ

1308 │ foreach (var item in tonKhoData)

1309 │ {

1310 │ item.SltonCuoiKy = (item.SltonDauKy ?? 0) + (item.SlnhapTrongKy ?? 0)

1311 │ - (item.SlbanTrongKy ?? 0) - (item.SlxuatKhac ?? 0)

1312 │ - (item.SlxuatTieuDung ?? 0) - (item.SlxuatTra ?? 0);

1313 │ }

1314 │

1315 │ using var transaction = \_context.Database.BeginTransaction();

1316 │ try

1317 │ {

1318 │ var tonKhoHienTai = \_context.StockTonKhoBanHang.Where(x => x.NamThang == namThang && x.MaKho == maKho).T...

1319 │ \_context.BulkDelete(tonKhoHienTai);

1320 │ \_context.SaveChanges();

1321 │ \_context.BulkInsert(tonKhoData);

1322 │ \_context.SaveChanges();

1323 │ transaction.Commit();

1324 │ }

1325 │ catch (Exception ex)

1326 │ {

1327 │ transaction.Rollback();

1328 │ throw new Exception($"❌ Lỗi khi cập nhật tồn kho: Tháng={namThang}, Kho={maKho}\n{ex.Message}...

1329 │

1330 │ }

1331 │

1332 │ }

1333 │

1334 │ public void MoKhoaSo(int Nam, int Thang, Guid User, Guid MaChiNhanh)

1335 │ {

1336 │ // Update NhapHang

1337 │ var nhapHangs = \_context.StockNhapHang

1338 │ .Where(n => n.NgayNhapHang.HasValue && n.NgayNhapHang.Value.Month == Thang &&

1339 │ n.NgayNhapHang.Value.Year == Nam &&

1340 │ n.MaChiNhanh == MaChiNhanh &&

1341 │ !(n.Deleted ?? false));

1342 │

1343 │ foreach (var nh in nhapHangs)

1344 │ {

1345 │ nh.KhoaSo = false;

1346 │ nh.UserModified = User;

1347 │ nh.ModifiedDate = DateTime.Now;

1348 │ }

1349 │

1350 │ // Update HoaDonBan

1351 │ var hoaDonBans = \_context.OpsHoaDon

1352 │ .Where(h => h.NgayHoaDon.Month == Thang &&

1353 │ h.NgayHoaDon.Year == Nam &&

1354 │ h.MaChiNhanh == MaChiNhanh &&

1355 │ !(h.Deleted ?? false));

1356 │

1357 │ foreach (var hd in hoaDonBans)

1358 │ {

1359 │ hd.KhoaSo = false;

1360 │ }

1361 │

1362 │ // Save changes to the database

1363 │ \_context.SaveChanges();

1364 │ }

1365 │ public StockTonKhoBanHang? GetTonKhoBanHangTheoHangHoa(int MaKho, int MaHangHoa, string NamThang)

1366 │ {

1367 │

1368 │ return \_context.StockTonKhoBanHang.FirstOrDefault(t => t.MaKho == MaKho && t.NamThang == NamThang && t.MaHan...

1369 │ }

1370 │ public decimal TonKhoTheoMaHang(int MaHangHoa, int Nam, int Thang, int MaKho, Guid MaChiNhanh)

1371 │ {

1372 │ var x = \_context.StockTonKho.Where(t => t.MaKho == MaKho && t.MaChiNhanh == MaChiNhanh && MaHangHoa == t.MaH...

1373 │ return x.Count == 0 ? 0 : x.Sum(t => t.SltonCuoiKy ?? 0);

1374 │ }

1375 │

1376 │ public List<VTonKhoBanHang> GetTonKhoBanHang(int MaKho, string NamThang)

1377 │ {

1378 │ var query = from tk in \_context.StockTonKhoBanHang.Where(x => x.MaKho == MaKho && x.NamThang == NamThang).As...

1379 │ join hh in \_context.TblHangHoa.AsNoTracking() on tk.MaHangHoa equals hh.Ma

1380 │ join dv in \_context.TblDonViTinh.AsNoTracking() on hh.MaDvt equals dv.Ma into dvGroup

1381 │ from dv in dvGroup.DefaultIfEmpty()

1382 │ join lh in \_context.TblLopHangHoa.AsNoTracking() on hh.MaLopHang equals lh.Ma into lhGroup

1383 │ from lh in lhGroup.DefaultIfEmpty()

1384 │ where (hh.Deleted ?? false) == false

1385 │ let latestNhap = (from ct in \_context.StockChiTietNhapHang.Where(x => !(x.Deleted ?? false)).AsN...

1386 │ join nh in \_context.StockNhapHang.Where(x => !(x.Deleted ?? false)).AsNoTracki...

1387 │ where ct.MaHangHoa == hh.Ma

1388 │ orderby nh.NgayNhapHang descending

1389 │ select new { ct.HanSd, ct.SoLuong }).FirstOrDefault()

1390 │ select new VTonKhoBanHang

1391 │ {

1392 │ MaKho = tk.MaKho,

1393 │ MaKt = hh.MaKt,

1394 │ TenLop = lh.Ten,

1395 │ MaVach = hh.MaVach,

1396 │ MaHangHoa = tk.MaHangHoa,

1397 │ TenHangHoa = hh.Ten,

1398 │ SltonDauKy = tk.SltonDauKy,

1399 │ SlnhapTrongKy = tk.SlnhapTrongKy,

1400 │ SlbanTrongKy = tk.SlbanTrongKy,

1401 │ SltonCuoiKy = tk.SltonCuoiKy,

1402 │ NamThang = tk.NamThang,

1403 │ SlxuatKhac = tk.SlxuatKhac,

1404 │ SlxuatTieuDung = tk.SlxuatTieuDung,

1405 │ TenDv = dv.Ten,

1406 │ SlxuatTra = tk.SlxuatTra,

1407 │ LoGanNhat = latestNhap != null ? latestNhap.HanSd : null,

1408 │ SoLuongGanNhat = latestNhap != null ? latestNhap.SoLuong : 0

1409 │ };

1410 │

1411 │

1412 │ return query.ToList();

1413 │ }

1414 │

1415 │ public List<sp\_BaoCaoTonKhoTheoNgayResult> TonKhoTuNgayDenNgay(int maKho, DateTime tuNgay, DateTime denNgay, Gui...

1416 │ {

1417 │ //return \_context.sp\_BaoCaoTonKhoTheoNgay

1418 │ // .FromSqlRaw("EXEC sp\_TonKho\_TuNgay\_DenNgay @MaKho = {0}, @TuNgay = {1}, @DenNgay = {2}", MaKho, TuNgay...

1419 │ // .ToList();

1420 │

1421 │ // Đảm bảo denNgay bao gồm cả ngày cuối cùng

1422 │ var denNgayEnd = denNgay.Date.AddDays(1).AddTicks(-1);

1423 │ var tuNgayStart = tuNgay.Date;

1424 │ // --- 1. Tính Tồn Đầu Kỳ tại `tuNgayStart` ---

1425 │

1426 │ // Lấy tháng trước của tuNgay để lấy tồn cuối tháng trước

1427 │ var ngayThangTruoc = tuNgayStart.AddMonths(-1);

1428 │ var namThangTruocStr = ngayThangTruoc.ToString("yyyyMM");

1429 │

1430 │ // Tồn cuối tháng trước (lấy từ StockTonKho hoặc StockTonKhoBanHang)

1431 │ var tonCuoiThangTruoc = \_context.StockTonKho // Hoặc StockTonKhoBanHang

1432 │ .Where(t => t.MaKho == maKho && t.NamThang == namThangTruocStr && t.MaChiNhanh == maChiNhanh) // Thêm M...

1433 │ .GroupBy(t => t.MaHangHoa)

1434 │ .Select(g => new HangHoaSoLuong

1435 │ {

1436 │ MaHangHoa = g.Key ?? 0,

1437 │ // Đặt tên chung là SoLuong

1438 │ SoLuong = g.Sum(t => t.SltonCuoiKy ?? 0)

1439 │ })

1440 │ .ToList(); // Lấy về memory để xử lý tiếp

1441 │

1442 │ // Tính ngày đầu tiên của tháng chứa tuNgay

1443 │ var ngayDauThang = new DateTime(tuNgayStart.Year, tuNgayStart.Month, 1);

1444 │

1445 │ // Phát sinh Nhập/Xuất/Bán từ đầu tháng đến trước tuNgay

1446 │ // Lấy dữ liệu Nhập hàng vào List

1447 │ var nhapHangList = \_context.StockNhapHang

1448 │ .Join(\_context.StockChiTietNhapHang, nh => nh.Ma, ctnh => ctnh.MaNhapHang, (nh, ctnh) => new { nh, ctnh ...

1449 │ .Where(x => x.nh.MaKho == maKho && x.nh.MaChiNhanh == maChiNhanh && !(x.nh.Deleted ?? false) && !(x.ctnh...

1450 │ x.nh.NgayNhapHang >= ngayDauThang && x.nh.NgayNhapHang < tuNgayStart)

1451 │ .Select(x => new { MaHangHoa = x.ctnh.MaHangHoa ?? 0, SLNhap = x.ctnh.SoLuong ?? 0m, SLXuat = 0m, SLBan ...

1452 │ .ToList(); // <-- Thực thi và lấy về List

1453 │

1454 │ // Lấy dữ liệu Nhập chuyển kho đến vào List

1455 │ var nhapChuyenKhoList = \_context.StockXuatKho

1456 │ .Join(\_context.StockChiTietXuatKho, xk => xk.Ma, ctxk => ctxk.MaPhieuXuat, (xk, ctxk) => new { xk, ctxk ...

1457 │ .Where(x => x.xk.MaKhoChuyenDen == maKho && x.xk.MaChiNhanh == maChiNhanh && !(x.xk.Deleted ?? false) &&...

1458 │ x.xk.NgayXuat >= ngayDauThang && x.xk.NgayXuat < tuNgayStart)

1459 │ .Select(x => new { MaHangHoa = x.ctxk.MaHangHoa ?? 0, SLNhap = x.ctxk.SoLuong ?? 0m, SLXuat = 0m, SLBan ...

1460 │ .ToList(); // <-- Thực thi và lấy về List

1461 │

1462 │ // Lấy dữ liệu Xuất kho đi vào List

1463 │ var xuatKhoList = \_context.StockXuatKho

1464 │ .Join(\_context.StockChiTietXuatKho, xk => xk.Ma, ctxk => ctxk.MaPhieuXuat, (xk, ctxk) => new { xk, ctxk ...

1465 │ .Where(x => x.xk.MaKho == maKho && x.xk.MaChiNhanh == maChiNhanh && !(x.xk.Deleted ?? false) && !(x.ctxk...

1466 │ x.xk.NgayXuat >= ngayDauThang && x.xk.NgayXuat < tuNgayStart)

1467 │ .Select(x => new { MaHangHoa = x.ctxk.MaHangHoa ?? 0, SLNhap = 0m, SLXuat = x.ctxk.SoLuong ?? 0m, SLBan ...

1468 │ .ToList(); // <-- Thực thi và lấy về List

1469 │

1470 │ // Lấy dữ liệu Bán hàng vào List

1471 │ var banHangList = \_context.OpsHoaDon

1472 │ .Join(\_context.OpsChiTietHoaDon, bh => bh.Ma, ctbh => ctbh.MaHoaDon, (bh, ctbh) => new { bh, ctbh })

1473 │ .Where(x => x.bh.MaKho == maKho && x.bh.MaChiNhanh == maChiNhanh && !(x.bh.Deleted ?? false) && !(x.ctbh...

1474 │ x.bh.NgayHoaDon >= ngayDauThang && x.bh.NgayHoaDon < tuNgayStart)

1475 │ .Select(x => new { MaHangHoa = x.ctbh.MaHangHoa ?? 0, SLNhap = 0m, SLXuat = 0m, SLBan = (x.ctbh.SoLuong ...

1476 │ .ToList(); // <-- Thực thi và lấy về List

1477 │

1478 │ // Kết hợp các List trong bộ nhớ bằng LINQ to Objects

1479 │ var combinedData = nhapHangList

1480 │ .Concat(nhapChuyenKhoList)

1481 │ .Concat(xuatKhoList)

1482 │ .Concat(banHangList);

1483 │

1484 │ // GroupBy và Select cuối cùng trên dữ liệu đã kết hợp trong bộ nhớ

1485 │ var phatSinhTruocTuNgay = combinedData

1486 │ .GroupBy(x => x.MaHangHoa)

1487 │ .Select(g => new HangHoaSoLuong // Sử dụng lớp cụ thể đã định nghĩa trước đó

1488 │ {

1489 │ MaHangHoa = g.Key,

1490 │ SoLuong = g.Sum(x => x.SLNhap - x.SLXuat - x.SLBan) // Tính phát sinh ròng

1491 │ })

1492 │ .ToList();// Lấy về memory

1493 │

1494 │ var combinedDauKyData = Enumerable.Concat(tonCuoiThangTruoc, phatSinhTruocTuNgay);

1495 │

1496 │

1497 │ // Tính tồn đầu kỳ thực tế tại tuNgayStart (LINQ to Objects)

1498 │ var tonDauKyTaiTuNgay = combinedDauKyData

1499 │ .GroupBy(x => x.MaHangHoa)

1500 │ .Select(g => new

1501 │ {

1502 │ MaHangHoa = g.Key,

1503 │ SoLuongDauKy = g.Sum(x => x.SoLuong) // Sum trực tiếp thuộc tính SoLuong chung

1504 │ })

1505 │ .Where(t => t.SoLuongDauKy != 0) // Chỉ lấy hàng hóa có tồn đầu

1506 │ .ToDictionary(t => t.MaHangHoa, t => t.SoLuongDauKy);

1507 │

1508 │ // --- 2. Tính Tổng Nhập/Xuất/Bán trong kỳ (từ `tuNgayStart` đến `denNgayEnd`) ---

1509 │ // Thực thi phần Nhập hàng

1510 │ var nhapHangTrongKyList = \_context.StockNhapHang

1511 │ .Join(\_context.StockChiTietNhapHang, nh => nh.Ma, ctnh => ctnh.MaNhapHang, (nh, ctnh) => new { nh, ctnh ...

1512 │ .Where(x => x.nh.MaKho == maKho && x.nh.MaChiNhanh == maChiNhanh && !(x.nh.Deleted ?? false) && !(x.ctnh...

1513 │ x.nh.NgayNhapHang >= tuNgayStart && x.nh.NgayNhapHang <= denNgayEnd)

1514 │ .Select(x => new { MaHangHoa = x.ctnh.MaHangHoa ?? 0, Loai = "Nhap", SoLuong = x.ctnh.SoLuong ?? 0 })

1515 │ .ToList(); // <-- Thực thi

1516 │

1517 │ // Thực thi phần Nhập chuyển kho đến

1518 │ var nhapChuyenKhoTrongKyList = \_context.StockXuatKho

1519 │ .Join(\_context.StockChiTietXuatKho, xk => xk.Ma, ctxk => ctxk.MaPhieuXuat, (xk, ctxk) => new { xk, ctxk ...

1520 │ .Where(x => x.xk.MaKhoChuyenDen == maKho && x.xk.MaChiNhanh == maChiNhanh && !(x.xk.Deleted ?? false) &&...

1521 │ x.xk.NgayXuat >= tuNgayStart && x.xk.NgayXuat <= denNgayEnd)

1522 │ .Select(x => new { MaHangHoa = x.ctxk.MaHangHoa ?? 0, Loai = "Nhap", SoLuong = x.ctxk.SoLuong ?? 0 })

1523 │ .ToList(); // <-- Thực thi

1524 │

1525 │ // Thực thi phần Xuất kho đi

1526 │ var xuatKhoTrongKyList = \_context.StockXuatKho

1527 │ .Join(\_context.StockChiTietXuatKho, xk => xk.Ma, ctxk => ctxk.MaPhieuXuat, (xk, ctxk) => new { xk, ctxk ...

1528 │ .Where(x => x.xk.MaKho == maKho && x.xk.MaChiNhanh == maChiNhanh && !(x.xk.Deleted ?? false) && !(x.ctxk...

1529 │ x.xk.NgayXuat >= tuNgayStart && x.xk.NgayXuat <= denNgayEnd)

1530 │ .Select(x => new { MaHangHoa = x.ctxk.MaHangHoa ?? 0, Loai = "XuatKhac", SoLuong = x.ctxk.SoLuong ?? 0 }...

1531 │ .ToList(); // <-- Thực thi

1532 │

1533 │ // Thực thi phần Bán hàng

1534 │ var banHangTrongKyList = \_context.OpsHoaDon

1535 │ .Join(\_context.OpsChiTietHoaDon, bh => bh.Ma, ctbh => ctbh.MaHoaDon, (bh, ctbh) => new { bh, ctbh })

1536 │ .Where(x => x.bh.MaKho == maKho && x.bh.MaChiNhanh == maChiNhanh && !(x.bh.Deleted ?? false) && !(x.ctbh...

1537 │ x.bh.NgayHoaDon >= tuNgayStart && x.bh.NgayHoaDon <= denNgayEnd)

1538 │ .Select(x => new { MaHangHoa = x.ctbh.MaHangHoa ?? 0, Loai = "Ban", SoLuong = (x.ctbh.SoLuong ?? 0) - (x...

1539 │ .ToList(); // <-- Thực thi

1540 │

1541 │ // Kết hợp các List trong bộ nhớ

1542 │ var combinedPhatSinhTrongKy = nhapHangTrongKyList

1543 │ .Concat(nhapChuyenKhoTrongKyList)

1544 │ .Concat(xuatKhoTrongKyList)

1545 │ .Concat(banHangTrongKyList);

1546 │

1547 │ // GroupBy và Select trên dữ liệu đã kết hợp trong bộ nhớ

1548 │ var phatSinhTrongKy = combinedPhatSinhTrongKy

1549 │ .GroupBy(x => new { x.MaHangHoa, x.Loai })

1550 │ .Select(g => new

1551 │ {

1552 │ g.Key.MaHangHoa,

1553 │ g.Key.Loai,

1554 │ TongSoLuong = g.Sum(x => x.SoLuong)

1555 │ })

1556 │ .ToList(); // <-- .ToList() cuối cùng để có kết quả phatSinhTrongKy

1557 │

1558 │ // --- 3. Kết hợp và tính toán ---

1559 │ var hangHoaTrongKy = phatSinhTrongKy.Select(p => p.MaHangHoa).Distinct().ToList();

1560 │ var tatCaHangHoa = tonDauKyTaiTuNgay.Keys.Union(hangHoaTrongKy).Distinct().ToList();

1561 │

1562 │ var result = (from hhId in tatCaHangHoa

1563 │ let dauKy = tonDauKyTaiTuNgay.ContainsKey(hhId) ? tonDauKyTaiTuNgay[hhId] : 0m

1564 │ let nhap = phatSinhTrongKy.Where(p => p.MaHangHoa == hhId && p.Loai == "Nhap").Sum(p => p.Tong...

1565 │ let xuatKhac = phatSinhTrongKy.Where(p => p.MaHangHoa == hhId && p.Loai == "XuatKhac").Sum(p =...

1566 │ let ban = phatSinhTrongKy.Where(p => p.MaHangHoa == hhId && p.Loai == "Ban").Sum(p => p.TongSo...

1567 │ let tonCuoi = dauKy + nhap - xuatKhac - ban

1568 │ // Chỉ lấy những hàng hóa có phát sinh hoặc có tồn đầu/cuối

1569 │ where dauKy != 0 || nhap != 0 || xuatKhac != 0 || ban != 0 || tonCuoi != 0

1570 │ select new // Tạo object tạm thời để join thông tin hàng hóa

1571 │ {

1572 │ MaHangHoa = hhId,

1573 │ SoLuongDauKy = dauKy,

1574 │ SoLuongNhapTrongKy = nhap,

1575 │ SoLuongXuatKhac = xuatKhac,

1576 │ SoLuongBanTrongKy = ban,

1577 │ SoLuongTonCuoiKy = tonCuoi

1578 │ })

1579 │ .Join(\_context.TblHangHoa, // Join với thông tin hàng hóa

1580 │ tk => tk.MaHangHoa,

1581 │ hh => hh.Ma,

1582 │ (tk, hh) => new { tk, hh })

1583 │ .GroupJoin(\_context.TblDonViTinh, // Left Join với Đơn vị tính

1584 │ th => th.hh.MaDvt,

1585 │ dvt => dvt.Ma,

1586 │ (th, dvts) => new { th.tk, th.hh, dvt = dvts.FirstOrDefault() })

1587 │ .Select(res => new sp\_BaoCaoTonKhoTheoNgayResult // Tạo kết quả cuối cùng

1588 │ {

1589 │ MaKT = res.hh.MaKt,

1590 │ TenHang = res.hh.Ten,

1591 │ DVT = res.dvt != null ? res.dvt.Ten : "",

1592 │ GiaVon = 0, // Tạm thời = 0, cần logic tính giá vốn phù hợp ở đây

1593 │ SoLuongDauKy = res.tk.SoLuongDauKy,

1594 │ SoLuongNhapTrongKy = res.tk.SoLuongNhapTrongKy,

1595 │ SoLuongXuatKhac = res.tk.SoLuongXuatKhac,

1596 │ SoLuongBanTrongKy = res.tk.SoLuongBanTrongKy,

1597 │ SoLuongTonCuoiKy = res.tk.SoLuongTonCuoiKy

1598 │ })

1599 │ .OrderBy(r => r.MaKT)

1600 │ .ToList();

1601 │

1602 │ return result;

1603 │ }

1604 │

1605 │ public List<sp\_NhapHangImportFASTResult> BaoCaoNhapHangImportFast(int MaKho, DateTime TuNgay, DateTime DenNgay, ...

1606 │ {

1607 │ return (

1608 │ from nk in \_context.StockNhapHang

1609 │ join ct in \_context.StockChiTietNhapHang on nk.Ma equals ct.MaNhapHang

1610 │ join hh in \_context.TblHangHoa on ct.MaHangHoa equals hh.Ma

1611 │ join ncc in \_context.TblNhaCc on nk.MaNhaCc equals ncc.Ma

1612 │ join k in \_context.StockDanhMucKho on nk.MaKho equals k.Ma

1613 │ join dv in \_context.TblDonViTinh on hh.MaDvt equals dv.Ma into dvGroup

1614 │ from dv in dvGroup.DefaultIfEmpty()

1615 │ where (nk.Deleted ?? false) == false

1616 │ && (ct.Deleted ?? false) == false

1617 │ && nk.MaKho == MaKho

1618 │ && nk.MaChiNhanh == MaChiNhanh

1619 │ && nk.NgayNhapHang >= TuNgay && nk.NgayNhapHang <= DenNgay

1620 │ orderby nk.Ma

1621 │ select new sp\_NhapHangImportFASTResult

1622 │ {

1623 │ Ma = nk.Ma,

1624 │ MaDVCS = "HHY1",

1625 │ MaGD = "2",

1626 │ NgayLapChungTu = nk.NgayNhapHang,

1627 │ NgayChungTu = nk.NgayNhapHang,

1628 │ MaNT = "VND",

1629 │ TyGia = 1,

1630 │ MaKT = ncc.MaKt + "",

1631 │ NguoiGiaoHang = "",

1632 │ TaiKhoanCo = k.TaiKhoanCo + "",

1633 │ MaThanhToan = "",

1634 │ DienGiai = "YT- Nhập " + k.Ten,

1635 │ TrangThai = "2",

1636 │ MaKT1 = hh.MaKt + "",

1637 │ DVT = dv.Ten + "",

1638 │ Code = k.Code + "",

1639 │ TenHang = hh.Ten + "",

1640 │ ViTri = "",

1641 │ Lo = "",

1642 │ SoLuong = ct.SoLuong ?? 0,

1643 │ GiaVon = ct.GiaVon ?? 0,

1644 │ ThanhTien = ct.ThanhTien,

1645 │ TyLeCK = (ct.ChietKhauPt ?? 0) > 0 ? (ct.ChietKhauPt ?? 0) / 100 : 0,

1646 │ ChietKhauNT = (ct.ChietKhauTm ?? 0) > 0 ? ct.ChietKhauTm ?? 0

1647 │ : (ct.ChietKhauPt ?? 0) > 0 ? ((ct.ChietKhauPt ?? 0) / 100) \* (ct.SoLuong ?? 0) \* (ct.Gia...

1648 │ : (nk.TienChietKhau ?? 0) > 0

1649 │ ? (nk.TienChietKhau ?? 0) /

1650 │ \_context.StockChiTietNhapHang.Count(x => x.MaNhapHang == nk.Ma)

1651 │ : 0,

1652 │ ThanhTienSauCK = ct.ThanhTienSauCk,

1653 │ TaiKhoanKho = k.TaiKhoanKho + "",

1654 │ MauBC = 3,

1655 │ MaTC = 1,

1656 │ SoHD = nk.SoHoaDon + "",

1657 │ SoSeri = "",

1658 │ NgayNhapHang = nk.NgayNhapHang,

1659 │ MaNhaCC = ncc.MaKt + "",

1660 │ TenNhaCC = ncc.TenNhaCc,

1661 │ DiaChi = ncc.DiaChi + "",

1662 │ MST = ncc.Mst + "",

1663 │ MaThue = "00",

1664 │ ThueSuat = (nk.ThueVat ?? 0) > 0 ? nk.ThueVat : (ct.Vat ?? 0),

1665 │ MaKho = k.Code + "",

1666 │ TienThue = ((nk.ThueVat ?? 0) > 0 ? nk.ThueVat : (ct.Vat ?? 0)) \* (ct.ThanhTien ?? 0),

1667 │ TKThue = "1331011",

1668 │ BoPhan = "BVHH001",

1669 │ NoiBo = 1

1670 │ }).ToList();

1671 │

1672 │ }

1673 │

1674 │ public List<sp\_BaoCaoTongHopXuatBanResult> BaoCaoTongHopXuatBan(int MaKho, DateTime TuNgay, DateTime DenNgay, Gu...

1675 │ {

1676 │

1677 │

1678 │ // Đảm bảo denNgay bao gồm cả ngày cuối cùng (đến cuối ngày)

1679 │ var denNgayEnd = DenNgay.Date.AddDays(1).AddTicks(-1);

1680 │ var tuNgayStart = TuNgay.Date; // Sử dụng tuNgayStart cho nhất quán

1681 │

1682 │ // --- Phần 1: Truy vấn số lượng xuất kho ---

1683 │ var xuatKhoList = (

1684 │ from xk in \_context.StockXuatKho

1685 │ join ct in \_context.StockChiTietXuatKho on xk.Ma equals ct.MaPhieuXuat

1686 │ where (ct.SoLuong ?? 0) != 0

1687 │ && xk.NgayXuat >= tuNgayStart && xk.NgayXuat <= denNgayEnd // Sử dụng biến đã chuẩn hóa

1688 │ && !(xk.Deleted ?? false)

1689 │ && xk.MaKho == MaKho

1690 │ && xk.MaChiNhanh == MaChiNhanh

1691 │ && !(ct.Deleted ?? false)

1692 │ group ct by new { xk.NgayXuat, xk.MaKho, ct.MaHangHoa } into g

1693 │ select new

1694 │ {

1695 │ NgayXuatBan = g.Key.NgayXuat ?? default(DateTime),

1696 │ MaKho = g.Key.MaKho,

1697 │ MaHangHoa = g.Key.MaHangHoa,

1698 │ SoLuongXuat = g.Sum(x => x.SoLuong ?? 0m),

1699 │ SoLuongBan = 0m

1700 │ }

1701 │ ).ToList(); // <-- Thực thi và lấy về List

1702 │

1703 │ // --- Phần 2: Truy vấn số lượng bán hàng ---

1704 │ var banHangList = (

1705 │ from bh in \_context.OpsHoaDon

1706 │ join ctbh in \_context.OpsChiTietHoaDon on bh.Ma equals ctbh.MaHoaDon

1707 │ where (ctbh.SoLuong ?? 0) != 0

1708 │ && bh.NgayHoaDon >= tuNgayStart && bh.NgayHoaDon <= denNgayEnd // Sử dụng biến đã chuẩn hóa

1709 │ && !(bh.Deleted ?? false)

1710 │ && bh.MaKho == MaKho

1711 │ && bh.MaChiNhanh == MaChiNhanh

1712 │ && !(ctbh.Deleted ?? false)

1713 │ group ctbh by new { bh.NgayHoaDon, bh.MaKho, ctbh.MaHangHoa } into g

1714 │ select new

1715 │ {

1716 │ NgayXuatBan = g.Key.NgayHoaDon, // Kiểu DateTime (khác với NgayXuat ở trên có thể là DateTi...

1717 │ MaKho = g.Key.MaKho,

1718 │ MaHangHoa = g.Key.MaHangHoa,

1719 │ SoLuongXuat = 0m,

1720 │ SoLuongBan = g.Sum(x => (x.SoLuong ?? 0m) - (x.SoLuongTraLai ?? 0m))

1721 │ }

1722 │ ).ToList(); // <-- Thực thi và lấy về List

1723 │

1724 │ // --- Kết hợp (UNION ALL) và Join với các bảng khác ---

1725 │ var combinedQuery = xuatKhoList

1726 │ .Select(x => new // Chiếu sang kiểu chung nếu cần (ví dụ xử lý DateTime...

1727 │ {

1728 │ x.NgayXuatBan, // Giữ nguyên nếu kiểu đã khớp

1729 │ x.MaKho,

1730 │ x.MaHangHoa,

1731 │ x.SoLuongXuat,

1732 │ x.SoLuongBan

1733 │ })

1734 │ .Concat(banHangList.Select(b => new // Chiếu sang kiểu chung nếu cần

1735 │ {

1736 │ b.NgayXuatBan, // Giữ nguyên nếu kiểu đã khớp

1737 │ b.MaKho,

1738 │ b.MaHangHoa,

1739 │ b.SoLuongXuat,

1740 │ b.SoLuongBan

1741 │ }));

1742 │

1743 │ var khoLookup = \_context.StockDanhMucKho.Where(k => k.Ma == MaKho).ToDictionary(k => k.Ma); // Tối ưu n��...

1744 │ var hangHoaIds = combinedQuery.Select(t => t.MaHangHoa).Distinct().ToList();

1745 │ var hangHoaLookup = \_context.TblHangHoa.Where(h => hangHoaIds.Contains(h.Ma)).ToDictionary(h => h.Ma);

1746 │ var dvtIds = hangHoaLookup.Values.Select(h => h.MaDvt).Distinct().ToList();

1747 │ var dvtLookup = \_context.TblDonViTinh.Where(d => dvtIds.Contains(d.Ma)).ToDictionary(d => d.Ma);

1748 │

1749 │

1750 │ var finalResultQuery =

1751 │ from t in combinedQuery

1752 │ // Lấy thông tin từ Dictionary đã tải về memory

1753 │ let k = khoLookup.ContainsKey(t.MaKho ?? 0) ? khoLookup[t.MaKho ?? 0] : null

1754 │ let hh = hangHoaLookup.ContainsKey(t.MaHangHoa ?? 0) ? hangHoaLookup[t.MaHangHoa ?? 0] : null

1755 │ let dv = (hh != null && dvtLookup.ContainsKey(hh.MaDvt ?? 0)) ? dvtLookup[hh.MaDvt ?? 0] : null

1756 │ where k != null && hh != null // Chỉ xử lý nếu có Kho và Hàng hóa

1757 │ group new { t, k, hh, dv } by new

1758 │ {

1759 │ Ngay = t.NgayXuatBan.Date, // Group theo ngày

1760 │ t.MaKho,

1761 │ TenKho = k.Ten,

1762 │ t.MaHangHoa,

1763 │ hh.MaKt,

1764 │ hh.MaVach,

1765 │ TenHang = hh.Ten,

1766 │ TenDV = dv?.Ten // Sử dụng null-conditional operator

1767 │ } into finalGroup

1768 │ select new sp\_BaoCaoTongHopXuatBanResult

1769 │ {

1770 │ NgayXuatBan = finalGroup.Key.Ngay,

1771 │ MaKho = finalGroup.Key.MaKho ?? 0,

1772 │ TenKho = finalGroup.Key.TenKho,

1773 │ MaHangHoa = finalGroup.Key.MaHangHoa ?? 0,

1774 │ MaKT = finalGroup.Key.MaKt,

1775 │ MaVach = finalGroup.Key.MaVach,

1776 │ TenHang = finalGroup.Key.TenHang,

1777 │ TenDV = finalGroup.Key.TenDV,

1778 │ SoLuongXuat = finalGroup.Sum(x => x.t.SoLuongXuat),

1779 │ SoLuongBan = finalGroup.Sum(x => x.t.SoLuongBan)

1780 │ };

1781 │

1782 │ // 3. Thực thi phần cuối (LINQ to Objects) và trả về kết quả

1783 │ return finalResultQuery.OrderBy(r => r.NgayXuatBan).ThenBy(r => r.TenHang).ToList();

1784 │ }

1785 │

1786 │ public List<sp\_BaoCaoTonKhoGiaVonResult> BaoCaoTonKhoGiaVon(int MaKho, string NamThang, Guid MaChiNhanh)

1787 │ {

1788 │ var result = (from tk in \_context.StockTonKho.Where(x => x.MaKho == MaKho && x.NamThang == NamThang && x.MaC...

1789 │ join hh in \_context.TblHangHoa on tk.MaHangHoa equals hh.Ma

1790 │ join dvt in \_context.TblDonViTinh on hh.MaDvt equals dvt.Ma into dvtGroup

1791 │ from dvt in dvtGroup.DefaultIfEmpty()

1792 │ join lh in \_context.TblLopHangHoa on hh.MaLopHang equals lh.Ma into lhGroup

1793 │ from lh in lhGroup.DefaultIfEmpty()

1794 │ select new sp\_BaoCaoTonKhoGiaVonResult

1795 │ {

1796 │ TenLop = lh.Ten,

1797 │ MaVach = hh.MaVach,

1798 │ MaKT = hh.MaKt,

1799 │ TenHang = hh.Ten,

1800 │ TenDVT = dvt.Ten,

1801 │ GiaVon = tk.GiaVon,

1802 │ SLTonDauKy = tk.SltonDauKy,

1803 │ SLNhapTrongKy = tk.SlnhapTrongKy,

1804 │ SLXuatKhac = tk.SlxuatKhac,

1805 │ SLXuatTra = tk.SlxuatTra,

1806 │ SLXuatTieuDung = tk.SlxuatTieuDung,

1807 │ SLBanTrongKy = tk.SlbanTrongKy,

1808 │ SLTonCuoiKy = tk.SltonCuoiKy

1809 │ }).ToList();

1810 │

1811 │

1812 │ return result;

1813 │ }

1814 │ }

1815 │ }

────────────────────────────────────────────────────────────

**📄 TonKhoTuNgayDenNgay.cs (CS, 29 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class TonKhoTuNgayDenNgay

7 │ {

8 │ public string? NamThang { get; set; }

9 │

10 │ public int? MaHangHoa { get; set; }

11 │

12 │ public int? MaNhapHang { get; set; }

13 │

14 │ public int? SltonDauKy1 { get; set; }

15 │

16 │ public int? SlnhapTrongKy1 { get; set; }

17 │

18 │ public int? SlbanTrongKy1 { get; set; }

19 │

20 │ public int? SltonCuoiKy1 { get; set; }

21 │

22 │ public int? SlnhapTrongKy2 { get; set; }

23 │

24 │ public int? SlbanTrongKy2 { get; set; }

25 │

26 │ public int? SltonCuoiKy2 { get; set; }

27 │

28 │ public int Ma { get; set; }

29 │ }

────────────────────────────────────────────────────────────

**📄 TonKhoVaGiaBanTheoKhoModel.cs (CS, 21 lines)**

1 │ ﻿namespace QuanLyBanHang.Models.ViewModels

2 │ {

3 │ public class TonKhoVaGiaBanTheoKhoModel

4 │ {

5 │ public int Ma { get; set; }

6 │ public string? MaVach { get; set; }

7 │ public string? TenHang { get; set; }

8 │ public decimal GiaBan { get; set; }

9 │ public int DVT { get; set; }

10 │ public int LopHang { get; set; }

11 │ public string? TenLop { get; set; }

12 │ public int MaKho { get; set; }

13 │ public string? TenKho { get; set; }

14 │ public string? TenDVT { get; set; }

15 │ public long SLTon { get; set; }

16 │ public string? HinhAnh { get; set; }

17 │ public bool BanNoiBo { get; set; }

18 │ public int MaHangHoa { get; set; }

19 │ public int LoaiTieuDung { get; set; }

20 │ }

21 │ }

────────────────────────────────────────────────────────────

**📄 TonKhoVaGiaBanTheoKhoModel\_AllStock.cs (CS, 19 lines)**

1 │ ﻿namespace QuanLyBanHang.Models.ViewModels

2 │ {

3 │ public class TonKhoVaGiaBanTheoKhoModel\_AllStock

4 │ {

5 │ public int Ma { get; set; }

6 │ public string? MaVach { get; set; }

7 │ public int? MaHangHoa { get; set; }

8 │ public string? TenHang { get; set; }

9 │ public int? GiaBan { get; set; }

10 │ public int LopHang { get; set; }

11 │ public string? TenLop { get; set; }

12 │ public int? MaKho { get; set; }

13 │ public string? TenKho { get; set; }

14 │ public int DVT { get; set; }

15 │ public string? TenDVT { get; set; }

16 │ public decimal SLTon { get; set; }

17 │ public string? HinhAnh { get; set; }

18 │ }

19 │ }

────────────────────────────────────────────────────────────

**📄 TonKhoVatTuModel.cs (CS, 278 lines)**

1 │ ﻿using Microsoft.EntityFrameworkCore;

2 │ using Newtonsoft.Json.Linq;

3 │ using QuanLyBanHang.Common;

4 │ using QuanLyBanHang.Models;

5 │ using QuanLyBanHang.Models.ExModel.OPS;

6 │ using QuanLyBanHang.Models.QLBHEntity;

7 │ using QuanLyBanHang.Services;

8 │ using System.Linq;

9 │

10 │ public class TonKhoVatTuModel : Base<StockTonKhoVatTu>

11 │ {

12 │ private readonly QuanLyBanHangContext \_context;

13 │

14 │ public TonKhoVatTuModel(QuanLyBanHangContext context) : base(context)

15 │ {

16 │ \_context = context;

17 │ }

18 │

19 │ public async Task<bool> KiemTraTonKhoTruocXuat(int maVatTu, int maKho, decimal soLuongCanXuat)

20 │ {

21 │

22 │ var tonKhoHienTai = GetSoLuongTonTheoVatTu(maVatTu, maKho);

23 │ return tonKhoHienTai >= soLuongCanXuat;

24 │ }

25 │

26 │ public decimal GetSoLuongTonTheoVatTu(int maVatTu, int maKho)

27 │ {

28 │ var namThang = DateTime.Now.ToString("yyyyMM");

29 │ var tonkho = \_context.StockTonKho.FirstOrDefault(x => x.NamThang == namThang && x.MaKho == maKho && x.MaHangHoa ...

30 │ if (tonkho == null)

31 │ {

32 │ return 0;

33 │ }

34 │ return tonkho.SltonCuoiKy ?? 0;

35 │ }

36 │

37 │ public async Task NhapKhay(int maKhay, decimal soLuongNhap, Guid? nguoiNhap, string ghiChu, Guid? branch)

38 │ {

39 │ var now = DateTime.Now;

40 │ var namThang = now.ToString("yyyyMM");

41 │ var khay = await \_context.VtthKhay.FirstOrDefaultAsync(x => x.Ma == maKhay && !(x.Deleted ?? false))

42 │ ?? throw new Exception("Không tìm thấy khay");

43 │

44 │ if (!(khay.Status ?? false)) throw new Exception("Khay không hoạt động");

45 │ if (khay.KhayTreo ?? false) throw new Exception("Khay đang treo");

46 │ if (khay.DaNhap ?? false) throw new Exception("Khay đã đầy");

47 │

48 │ if ((khay.DinhMucKhay ?? 0) != soLuongNhap) throw new Exception($"Số lượng nhập phải bằng định m��...

49 │ var tonkho = \_context.StockTonKhoBanHang.FirstOrDefault(x => x.MaKho == khay.MaKhoNguyenLieu && x.MaHangHoa == k...

50 │ if (tonkho == null) throw new Exception("Chưa tồn tại trong kho nguyên liệu");

51 │ if ((tonkho.SltonCuoiKy ?? 0) < (khay.DinhMucKhay ?? 0)) throw new Exception($"Số lượng tồn kho không đ...

52 │ var br = \_context.SysBranch.FirstOrDefault(x => x.Ma == branch);

53 │

54 │ var recordNhap = new VtthNhapXuatKhay

55 │ {

56 │ MaKhay = maKhay,

57 │ LoaiPhieu = (int)LoaiNhapXuatVTTH.Nhap,

58 │ SoLuongNhap = soLuongNhap,

59 │ NgayNhap = now,

60 │ CreatedDate = now,

61 │ UserCreated = nguoiNhap,

62 │ ConLai = soLuongNhap,

63 │ Deleted = false,

64 │ GhiChu = ghiChu,

65 │ MaChiNhanh = branch ?? Utility.defaultUID,

66 │ MaChiNhanhInt = br.Nid ?? 0

67 │ };

68 │ \_context.VtthNhapXuatKhay.Add(recordNhap);

69 │

70 │ khay.DaNhap = true;

71 │ khay.ModifiedDate = now;

72 │ khay.UserModified = nguoiNhap;

73 │ \_context.VtthKhay.Update(khay);

74 │

75 │ await \_context.SaveChangesAsync();

76 │ TonKhoBackgroundWorker.AddJob(new TonKhoJob

77 │ {

78 │ NamThang = DateTime.Now.ToString("yyyyMM"),

79 │ MaKho = khay.MaKhoNguyenLieu ?? 0,

80 │ UserId = nguoiNhap ?? Utility.defaultUID,

81 │ MaChiNhanh = branch ?? Utility.defaultUID

82 │ });

83 │ TonKhoBackgroundWorker.AddJob(new TonKhoJob

84 │ {

85 │ NamThang = DateTime.Now.ToString("yyyyMM"),

86 │ MaKho = khay.MaKhoTieuHao ?? 0,

87 │ UserId = nguoiNhap ?? Utility.defaultUID,

88 │ MaChiNhanh = branch ?? Utility.defaultUID

89 │ });

90 │ }

91 │

92 │ public async Task XuatKhay(int maKhay, decimal soLuongXuat, Guid? nguoiXuat, string? ghiChu, Guid? branch)

93 │ {

94 │ var now = DateTime.Now;

95 │ var namThang = now.ToString("yyyyMM");

96 │ var khay = await \_context.VtthKhay.FirstOrDefaultAsync(x => x.Ma == maKhay && !(x.Deleted ?? false));

97 │ if (khay == null) throw new Exception("Không tìm thấy khay");

98 │

99 │ var br = await \_context.SysBranch.FirstOrDefaultAsync(x => x.Ma == branch && !(x.Deleted ?? false));

100 │ if (br == null) throw new Exception("Không tìm thấy chi nhánh");

101 │

102 │ var lastRecord = await \_context.VtthNhapXuatKhay.Where(x => x.MaKhay == maKhay && !(x.Deleted ?? false)).OrderBy...

103 │ if(lastRecord == null)

104 │ {

105 │ throw new Exception("Không tìm thấy khay nhập");

106 │ }

107 │ if(lastRecord.LoaiPhieu == (int)LoaiNhapXuatVTTH.Xuat)

108 │ {

109 │ if (lastRecord.ConLai < soLuongXuat) throw new Exception($"Không đủ tồn trên khay\nHiện tại: {las...

110 │ lastRecord.SoLuongXuat += soLuongXuat;

111 │ lastRecord.ConLai -= soLuongXuat;

112 │ lastRecord.ModifiedDate = now;

113 │ lastRecord.UserModified = nguoiXuat;

114 │ \_context.VtthNhapXuatKhay.Update(lastRecord);

115 │

116 │ if (lastRecord.ConLai == 0)

117 │ {

118 │ khay.DaNhap = false;

119 │ \_context.Update(khay);

120 │ }

121 │

122 │ var lichSuXuat = new VtthLichSuXuatKhay

123 │ {

124 │ MaChiNhanhInt = br.Nid ?? 0,

125 │ MaChiNhanh = br.Ma,

126 │ MaNhapKhay = lastRecord.Ma,

127 │ MaKhoNhap = khay.MaKhoNguyenLieu,

128 │ MaKhoXuat = khay.MaKhoTieuHao,

129 │ MaKhoXuatTra = khay.MaKhoNguyenLieu,

130 │ MaKhay = khay.Ma,

131 │ MaVatTu = khay.MaVatTu,

132 │ NgayXuat = DateTime.Now,

133 │ SoLuongXuat = soLuongXuat,

134 │ NguoiXuat = nguoiXuat,

135 │ GhiChu = ghiChu,

136 │ };

137 │ \_context.VtthLichSuXuatKhay.Add(lichSuXuat);

138 │ \_context.SaveChanges();

139 │ }

140 │ else

141 │ {

142 │ var recordXuat = new VtthNhapXuatKhay

143 │ {

144 │ MaKhay = maKhay,

145 │ LoaiPhieu = (int)LoaiNhapXuatVTTH.Xuat,

146 │ SoLuongXuat = soLuongXuat,

147 │ NgayXuat = now,

148 │ NguoiXuat = nguoiXuat,

149 │ ConLai = khay.DinhMucKhay - soLuongXuat,

150 │ CreatedDate = now,

151 │ UserCreated = nguoiXuat,

152 │ Deleted = false,

153 │ GhiChu = ghiChu,

154 │ MaChiNhanh = branch ?? Utility.defaultUID,

155 │ MaChiNhanhInt = br.Nid ?? 0,

156 │ };

157 │ \_context.VtthNhapXuatKhay.Add(recordXuat);

158 │ if (recordXuat.ConLai == 0)

159 │ {

160 │ khay.DaNhap = false;

161 │ \_context.Update(khay);

162 │ }

163 │ var lichSuXuat = new VtthLichSuXuatKhay

164 │ {

165 │ MaChiNhanhInt = br.Nid ?? 0,

166 │ MaChiNhanh = br.Ma,

167 │ MaNhapKhay = recordXuat.Ma,

168 │ MaKhoNhap = khay.MaKhoNguyenLieu,

169 │ MaKhoXuat = khay.MaKhoTieuHao,

170 │ MaKhoXuatTra = khay.MaKhoNguyenLieu,

171 │ MaKhay = khay.Ma,

172 │ MaVatTu = khay.MaVatTu,

173 │ NgayXuat = DateTime.Now,

174 │ SoLuongXuat = soLuongXuat,

175 │ NguoiXuat = nguoiXuat,

176 │ GhiChu = ghiChu,

177 │ };

178 │ \_context.VtthLichSuXuatKhay.Add(lichSuXuat);

179 │ \_context.SaveChanges();

180 │

181 │ }

182 │

183 │ TonKhoBackgroundWorker.AddJob(new TonKhoJob

184 │ {

185 │ NamThang = DateTime.Now.ToString("yyyyMM"),

186 │ MaKho = khay.MaKhoNguyenLieu ?? 0,

187 │ UserId = nguoiXuat ?? Utility.defaultUID,

188 │ MaChiNhanh = branch ?? Utility.defaultUID

189 │ });

190 │ TonKhoBackgroundWorker.AddJob(new TonKhoJob

191 │ {

192 │ NamThang = DateTime.Now.ToString("yyyyMM"),

193 │ MaKho = khay.MaKhoTieuHao ?? 0,

194 │ UserId = nguoiXuat ?? Utility.defaultUID,

195 │ MaChiNhanh = branch ?? Utility.defaultUID

196 │ });

197 │

198 │ }

199 │

200 │ public async Task TraKhay(int maKhay, decimal soLuongTra, Guid? nguoiTra, string? ghiChu, Guid MaChiNhanh)

201 │ {

202 │ var now = DateTime.Now;

203 │

204 │ var khay = await \_context.VtthKhay.FirstOrDefaultAsync(x => x.Ma == maKhay && !(x.Deleted ?? false));

205 │ if (khay == null) throw new Exception("Không tìm thấy khay");

206 │

207 │ var br = await \_context.SysBranch.FirstOrDefaultAsync(x => x.Ma == MaChiNhanh && !(x.Deleted ?? false));

208 │ if (br == null) throw new Exception("Không tìm thấy chi nhánh");

209 │

210 │ var recordTra = new VtthNhapXuatKhay

211 │ {

212 │ MaKhay = maKhay,

213 │ LoaiPhieu = (int)LoaiNhapXuatVTTH.Tra,

214 │ SoLuongTra = soLuongTra,

215 │ NgayTra = now,

216 │ ConLai = 0,

217 │ NguoiTra = nguoiTra,

218 │ CreatedDate = now,

219 │ UserCreated = nguoiTra,

220 │ Deleted = false,

221 │ GhiChu = ghiChu,

222 │ MaChiNhanh = br.Ma,

223 │ MaChiNhanhInt = br.Nid ?? 0,

224 │ };

225 │ \_context.VtthNhapXuatKhay.Add(recordTra);

226 │ await \_context.SaveChangesAsync();

227 │

228 │ khay.KhayTreo = true;

229 │ khay.ModifiedDate = now;

230 │ khay.UserModified = nguoiTra;

231 │ await \_context.SaveChangesAsync();

232 │ }

233 │

234 │ public async Task XacNhanTraKhay(int maNhapXuatKhay, Guid? nguoiXacNhan, Guid? branch)

235 │ {

236 │ var now = DateTime.Now;

237 │

238 │ var recordTra = await \_context.VtthNhapXuatKhay.FirstOrDefaultAsync(x => x.Ma == maNhapXuatKhay && !(x.Deleted ?...

239 │ if (recordTra == null) throw new Exception("Không tìm thấy record trả");

240 │

241 │ if (recordTra.XacNhanTra == true || (recordTra.SoLuongTra ?? 0) == 0)

242 │ throw new Exception("Đã xác nhận hoặc không có số lượng trả");

243 │

244 │ var khay = await \_context.VtthKhay.FirstOrDefaultAsync(x => x.Ma == recordTra.MaKhay && !(x.Deleted ?? false));

245 │ if (khay == null) throw new Exception("Không tìm thấy khay");

246 │

247 │

248 │ recordTra.XacNhanTra = true;

249 │ recordTra.NgayXacNhanTra = now;

250 │ recordTra.ModifiedDate = now;

251 │ recordTra.UserModified = nguoiXacNhan;

252 │ \_context.VtthNhapXuatKhay.Update(recordTra);

253 │

254 │ khay.KhayTreo = false;

255 │ khay.DaNhap = false;

256 │ \_context.Update(khay);

257 │ await \_context.SaveChangesAsync();

258 │

259 │ TonKhoBackgroundWorker.AddJob(new TonKhoJob

260 │ {

261 │ NamThang = DateTime.Now.ToString("yyyyMM"),

262 │ MaKho = khay.MaKhoNguyenLieu ?? 0,

263 │ UserId = nguoiXacNhan ?? Utility.defaultUID,

264 │ MaChiNhanh = branch ?? Utility.defaultUID

265 │ });

266 │ TonKhoBackgroundWorker.AddJob(new TonKhoJob

267 │ {

268 │ NamThang = DateTime.Now.ToString("yyyyMM"),

269 │ MaKho = khay.MaKhoTieuHao ?? 0,

270 │ UserId = nguoiXacNhan ?? Utility.defaultUID,

271 │ MaChiNhanh = branch ?? Utility.defaultUID

272 │ });

273 │ }

274 │

275 │

276 │

277 │

278 │ }

────────────────────────────────────────────────────────────

**📄 UserController.cs (CS, 594 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using QuanLyBanHang.Common;

3 │ using QuanLyBanHang.Common.Attributes;

4 │ using QuanLyBanHang.Models.APIModels;

5 │ using QuanLyBanHang.Models.ExModel.SYS;

6 │ using QuanLyBanHang.Models.QLBHEntity;

7 │

8 │ namespace QuanLyBanHang.Controllers.SYS

9 │ {

10 │ public class UserController : BaseController

11 │ {

12 │ public UserController(QuanLyBanHangContext dbcontext, MemoryCacheManager memoryCache) : base(dbcontext, memoryCa...

13 │ {

14 │ m\_MenuID = MenuID.sysUser.GetEnumGuid();

15 │ }

16 │

17 │ [TokenUserCheckHTTP()]

18 │ public IActionResult Index()

19 │ {

20 │ ViewData["ListData"] = new sys\_UserModel(dbcontext).GetData().OrderByDescending(t => t.CreatedDate).ToList()...

21 │ ViewData["ListPhongBan"] = dbcontext.TblPhongBan.ToList();

22 │ ViewData["ListCN"] = dbcontext.SysBranch.ToList();

23 │ ViewData["ListChucVu"] = dbcontext.TblChucVu.Where(t => !(t.Deleted ?? false)).ToList();

24 │ return View();

25 │ }

26 │ [TokenUserCheckHTTP()]

27 │ public IActionResult Account()

28 │ {

29 │ ViewData["item"] = new sys\_UserModel(dbcontext).GetItem(objToken.userid);

30 │

31 │ return View();

32 │ }

33 │ [HttpPost]

34 │ [TokenUserCheckHTTP(Permission.NotCheck)]

35 │ public IActionResult UpdateMyAccount(string MSNV, string Ten, string CCCD, string NgaySinh, bool GioiTinh, strin...

36 │ {

37 │ try

38 │ {

39 │ var ur = dbcontext.SysUser.FirstOrDefault(x => x.Ma == objToken.userid);

40 │ ur.Msnv = MSNV;

41 │ ur.FullName = Ten;

42 │ ur.Cccd = CCCD;

43 │ ur.NgaySinh = NgaySinh.ToDateTime2();

44 │ ur.GioiTinh = GioiTinh ? 1 : 0;

45 │ ur.SoDienThoai = SDT;

46 │ ur.HinhAnh = HinhAnh != "" ? HinhAnh : "";

47 │ dbcontext.Update(ur);

48 │ dbcontext.SaveChanges();

49 │

50 │ }

51 │ catch(Exception ex)

52 │ {

53 │ return Json(new ResponseModelError(ex.Message));

54 │ }

55 │ return Json(new ResponseModelSuccess("Lưu thành công"));

56 │ }

57 │ [TokenUserCheckHTTP(Permission.Add)]

58 │ public IActionResult Add()

59 │ {

60 │ // ViewData["ListPhongBan"] = new tbl\_PhongBanModel(dbcontext).GetData().Where(t => t.Status ?? false).ToLis...

61 │ //ViewData["ListPhongBanKiem"] = new tbl\_PhongBanModel(dbcontext).GetData().Where(t => t.Status ?? false).To...

62 │ bool bShowAll = Utility.checkPermissionIDModel(objToken.userid, m\_MenuID, Permission.ShowAll.GetEnumGuid());

63 │ if (bShowAll)

64 │ ViewData["ListBranch"] = dbcontext.SysBranch.Where(t => !(t.Deleted ?? false) && (t.Status ?? false)).To...

65 │ else

66 │ ViewData["ListBranch"] = dbcontext.SysBranch.Where(t => !(t.Deleted ?? false) && (t.Status ?? false)

67 │ && t.Ma == objToken.branch).ToList();

68 │

69 │ ViewData["ListKhoHang"] = dbcontext.StockDanhMucKho.Where(t => !(t.Deleted ?? false) && t.MaChiNhanh == objT...

70 │

71 │ ViewData["ListChucVu"] = dbcontext.TblChucVu.Where(t => !(t.Deleted ?? false)).ToList();

72 │ return View("Edit");

73 │ }

74 │

75 │ [TokenUserCheckHTTP(Permission.Edit)]

76 │ public IActionResult Edit(string ID)

77 │ {

78 │ SysUser item = new sys\_UserModel(dbcontext).GetItem(ID.GetGuid());

79 │ ViewData["item"] = item;

80 │ bool bShowAll = Utility.checkPermissionIDModel(objToken.userid, m\_MenuID, Permission.ShowAll.GetEnumGuid());

81 │ if (bShowAll)

82 │ ViewData["ListBranch"] = dbcontext.SysBranch.Where(t => !(t.Deleted ?? false) && (t.Status ?? false)).To...

83 │ else

84 │ ViewData["ListBranch"] = dbcontext.SysBranch.Where(t => !(t.Deleted ?? false) && (t.Status ?? false)

85 │ && t.Ma == objToken.branch).ToList();

86 │

87 │ ViewData["ListKhoHang"] = dbcontext.StockDanhMucKho.Where(t => !(t.Deleted ?? false) && t.MaChiNhanh == objT...

88 │

89 │ //ViewData["ListPhongBan"] = new tbl\_PhongBanModel(dbcontext).GetData().Where(t => t.Status ?? false).ToList...

90 │

91 │ ViewData["ListChucVu"] = dbcontext.TblChucVu.Where(t => !(t.Deleted ?? false)).ToList();

92 │ return View();

93 │ }

94 │

95 │ [HttpPost]

96 │ [TokenUserCheckHTTP(Permission.NotCheck)]

97 │ public IActionResult UpdateAvatar(string HinhAnh)

98 │ {

99 │ (int, string) result = new sys\_UserModel(dbcontext).UpdateAvatar(objToken!.userid, HinhAnh);

100 │ if (result.Item1 == 1)

101 │ {

102 │ return Json(new ResponseModelSuccess("", result.Item2));

103 │ }

104 │ return Json(new ResponseModelError("Đã xảy ra lỗi"));

105 │ }

106 │

107 │ [HttpPost]

108 │ [TokenUserCheckHTTP(Permission.Add)]

109 │ public IActionResult AddAPI(string Username, string FullName, string GioiTinh, string NgaySinh, string SoDienTho...

110 │ , string MSNV, string MaPhongBan, string MaChucDanh, string MaChucVu, string CCCD, string MaDinhDanh, string...

111 │ string Photo, string Status, string Branch, string KhoHang)

112 │ {

113 │ var res = Utility.CheckParam(Username, FullName, GioiTinh, Status, Branch, NgaySinh);

114 │ if (res is not null)

115 │ {

116 │ return Json(res);

117 │ }

118 │ if (MSNV.IsEmpty())

119 │ {

120 │ return Json(new ResponseModelError("Vui lòng nhập Mã số nhân viên!"));

121 │ }

122 │ else if (MSNV.Trim().Length > 50)

123 │ {

124 │ return Json(new ResponseModelError("Mã số nhân viên không đúng. Yêu cầu MSNV không được ...

125 │ }

126 │ else if (dbcontext.SysUser.Any(x => x.Msnv == MSNV && !(x.Deleted ?? false) && (x.Status ?? false)))

127 │ {

128 │ return Json(new ResponseModelError("Mã số nhân viên đã tồn tại."));

129 │ }

130 │

131 │ DateTime dtpNgaySinh = Cast.ToDateTime(NgaySinh, "dd/MM/yyyy");

132 │ DateTime dtpNgayVaoLam = Cast.ToDateTime(NgayVaoLam, "dd/MM/yyyy");

133 │ DateTime? dtpNgayKetThuc = (NgayKetThuc != "" && NgayKetThuc != null) ? Cast.ToDateTime(NgayKetThuc, "dd/MM/...

134 │

135 │

136 │ string listKhoHang = "";

137 │ listKhoHang = Utility.SetItemMultiSelect(KhoHang);

138 │

139 │ string Password = string.Format("{0}@{1}", Username, DateTime.Now.Year);

140 │ int result = new sys\_UserModel(dbcontext).Add(Username, FullName, Password, Cast.ToInt(GioiTinh), dtpNgaySin...

141 │ MSNV.Trim(), MaPhongBan.GetGuidNull(), MaChucDanh.GetGuidNull(), MaChucVu.GetGuidNull(), CCCD, MaDinhDan...

142 │ Photo, Status == "1", objToken!.userid, new Guid(Branch), listKhoHang);

143 │

144 │ if (result == 1)

145 │ {

146 │ return Json(new ResponseModelSuccess("Thao tác thành công. Mật khẩu của bạn là " + Password)...

147 │ }

148 │ return Json(new ResponseModelError("Đã xảy ra lỗi"));

149 │ }

150 │

151 │ [HttpPost]

152 │ [TokenUserCheckHTTP(Permission.Edit)]

153 │ public IActionResult UpdateAPI(string ID, string Username, string FullName, string GioiTinh, string NgaySinh, st...

154 │ , string MSNV, string MaPhongBan, string MaChucDanh, string MaChucVu, string CCCD, string MaDinhDanh, string...

155 │ , string Photo, string Status, string Branch, string KhoHang)

156 │ {

157 │ var res = Utility.CheckParam(Username, FullName, GioiTinh, Status, Branch, NgaySinh);

158 │ if (res is not null)

159 │ {

160 │ return Json(res);

161 │ }

162 │

163 │ if (!SoDienThoai.IsEmpty())

164 │ {

165 │ if (SoDienThoai[0] == '0')

166 │ {

167 │ SoDienThoai = "84" + SoDienThoai.Substring(1);

168 │ }

169 │ }

170 │ if (MSNV == null || MSNV.IsEmpty())

171 │ {

172 │ return Json(new ResponseModelError("Vui lòng nhập Mã số nhân viên!"));

173 │ }

174 │ else if (MSNV.Trim().Length > 50)

175 │ {

176 │ return Json(new ResponseModelError("Mã số nhân viên không đúng. Yêu cầu MSNV không được ...

177 │ }

178 │ else if (dbcontext.SysUser.Any(x => x.Msnv == MSNV && !(x.Deleted ?? false) && (x.Status ?? false) && (x.Ma ...

179 │ {

180 │ return Json(new ResponseModelError("Mã số nhân viên đã tồn tại."));

181 │ }

182 │ DateTime dtpNgaySinh = Cast.ToDateTime(NgaySinh, "dd/MM/yyyy");

183 │ DateTime dtpNgayVaoLam = Cast.ToDateTime(NgayVaoLam, "dd/MM/yyyy");

184 │ DateTime? dtpNgayKetThuc = (NgayKetThuc != "" && NgayKetThuc != null) ? Cast.ToDateTime(NgayKetThuc, "dd/MM/...

185 │

186 │ string listKhoHang = "";

187 │ listKhoHang = Utility.SetItemMultiSelect(KhoHang);

188 │

189 │ int result = new sys\_UserModel(dbcontext).Edit(ID.GetGuid(), Username, FullName, Cast.ToInt(GioiTinh), dtpNg...

190 │ MSNV.Trim(), MaPhongBan.GetGuidNull(), MaChucDanh.GetGuidNull(), MaChucVu.GetGuidNull(), CCCD, MaDinhDan...

191 │ Photo, Status == "1", objToken!.userid, new Guid(Branch), listKhoHang, objToken.branch);

192 │

193 │ if (result == 1)

194 │ {

195 │ return Json(new ResponseModelSuccess("Thao tác thành công"));

196 │ }

197 │ return Json(new ResponseModelError("Đã xảy ra lỗi"));

198 │ }

199 │

200 │ [HttpPost]

201 │ [TokenUserCheckHTTP(Permission.Delete)]

202 │ public IActionResult DeleteAPI(string ID)

203 │ {

204 │ int result = new sys\_UserModel(dbcontext).Delete(ID.GetGuid(), objToken!.userid);

205 │ if (result == 1)

206 │ return Json(QLBHConst.ResponseDeleteSuccessfully);

207 │ else if (result == 0)

208 │ return Json(QLBHConst.ResponseDeleteFailure);

209 │ else if (result == -1)

210 │ return Json(QLBHConst.ResponseCannotDelete);

211 │ else if (result == -2)

212 │ return Json(QLBHConst.ResponseNotExist);

213 │

214 │ return Json(new ResponseModelError("Đã xảy ra lỗi"));

215 │ }

216 │

217 │ [HttpPost]

218 │ [TokenUserCheckHTTP()]

219 │ public IActionResult ResetPassword(string ID, string Password)

220 │ {

221 │ int result = new sys\_UserModel(dbcontext).ResetPassword(ID.GetGuid(), Password);

222 │

223 │ if (result == 1)

224 │ return Json(new ResponseModelSuccess("Đặt lại mật khẩu thành công"));

225 │ else if (result == 0)

226 │ return Json(new ResponseModelError("Đặt lại mật khẩu thất bại"));

227 │ else if (result == -1)

228 │ Json(QLBHConst.ResponseNotExist);

229 │ return Json(new ResponseModelError("Đã xảy ra lỗi"));

230 │ }

231 │ [HttpPost]

232 │ [TokenUserCheckHTTP(Permission.NotCheck)]

233 │ public ActionResult ChangePassword(string ID, string CurrentPassword, string NewPassword, string VerifyNewPasswo...

234 │ {

235 │ SysUser drU = new sys\_UserModel(dbcontext).GetItem(ID.GetGuid());

236 │

237 │ if (drU != null)

238 │ {

239 │ var res = Utility.CheckParam(CurrentPassword, NewPassword, VerifyNewPassword);

240 │ if (res is not null)

241 │ {

242 │ return Json(res);

243 │ }

244 │

245 │ if (CurrentPassword == "")

246 │ {

247 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng nhập mật khẩu hiện tại"));

248 │ }

249 │ else

250 │ {

251 │ if (drU.Password != Utility.GetSHA512(CurrentPassword))

252 │ {

253 │ return Json(new ResponseModel(QLBHConst.Error, "Mật khẩu hiện tại không đúng"));

254 │ }

255 │ }

256 │

257 │ if (NewPassword == "")

258 │ {

259 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng nhập mật khẩu mới"));

260 │ }

261 │

262 │ if (VerifyNewPassword == "")

263 │ {

264 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng nhập xác nhận mật khẩu"));

265 │ }

266 │ else

267 │ {

268 │ if (NewPassword != "" && VerifyNewPassword != "" && NewPassword != VerifyNewPassword)

269 │ {

270 │ return Json(new ResponseModel(QLBHConst.Error, "Mật khẩu mới và xác nhận mật khẩu ...

271 │ }

272 │ }

273 │

274 │

275 │ //..

276 │ string result = Cast.ToString(new sys\_UserModel(dbcontext).ChangePassword(ID.GetGuid(), CurrentPassword,...

277 │

278 │ if (result == "1")//Success

279 │ return Json(new ResponseModel(QLBHConst.OK, QLBHConst.UpdateSuccessfully));

280 │ else if (result == "0")//Failure

281 │ return Json(new ResponseModel(QLBHConst.Error, QLBHConst.UpdateFailure));

282 │ else if (result == "-1")//Not exist

283 │ return Json(new ResponseModel(QLBHConst.Error, QLBHConst.NotExist));

284 │ }

285 │

286 │ return Json(new ResponseModel(QLBHConst.Error, "Đã xảy ra lỗi"));

287 │ }

288 │

289 │ [HttpPost]

290 │ [TokenUserCheckHTTP(Permission.NotCheck)]

291 │ public ActionResult ChangePasswordAPI(string CurrentPassword, string NewPassword, string VerifyNewPassword)

292 │ {

293 │ return ChangePassword(objToken?.userid + "", CurrentPassword, NewPassword, VerifyNewPassword);

294 │ }

295 │

296 │ private class QuayTiepNhan

297 │ {

298 │ public string? MaKhuVuc { get; set; }

299 │ public string? Ten { get; set; }

300 │ public string? TrangThai { get; set; }

301 │ public string? Result { get; set; }

302 │ public string? ResultText { get; set; }

303 │ public string? url { get; set; }

304 │ }

305 │

306 │ /\* [HttpPost]

307 │ [TokenUserCheckHTTP()]

308 │ public string LayDanhSachQuayTheoKhuVuc(string MaKhuVuc, string QuayTiepNhanChon)

309 │ {

310 │ QuayTiepNhan pb = new QuayTiepNhan();

311 │ List<SttQuayTiepNhan> ListQuayTiepNhan = new List<SttQuayTiepNhan>();

312 │ ListQuayTiepNhan = new stt\_QuayTiepNhanModel(dbcontext).GetData().Where(t => t.MaKhuVuc == Cast.ToInt(MaK...

313 │ pb.Result = "1";

314 │ foreach (SttQuayTiepNhan item in ListQuayTiepNhan)

315 │ {

316 │ pb.ResultText += string.Format("<option value='{0}' {1}>{2}</option>", item.Ma

317 │ , Cast.ToString(QuayTiepNhanChon).Contains(Utility.SetItemMultiSelect(item.Ma)) ? "selected" : "",...

318 │ }

319 │ //return Json(new ResponseModel(QLBHConst.Error,"", pb));

320 │ return JsonConvert.SerializeObject(pb);

321 │ }

322 │ \*/

323 │ //[HttpPost]

324 │ //[TokenUserCheckHTTP(Permission.NotCheck)]

325 │ //public IActionResult UpdateReminder(string flag)

326 │ //{

327 │ // var ur = dbcontext.SysUser.FirstOrDefault(t => t.Ma == objToken!.userid && !(t.Deleted ?? false));

328 │ // if (ur == null)

329 │ // {

330 │ // return Json(new ResponseModelError("Không tìm thấy!"));

331 │ // }

332 │

333 │ // ur.Reminder = flag == "1";

334 │ // dbcontext.Update(ur);

335 │ // if (dbcontext.SaveChanges() > 0)

336 │ // {

337 │ // return Json(new ResponseModelSuccess("", new { flag }));

338 │ // }

339 │

340 │ // return Json(new ResponseModelError("Cập nhật thất bại"));

341 │ //}

342 │

343 │ [HttpPost]

344 │ [TokenUserCheckHTTP(Permission.NotCheck)]

345 │ public IActionResult UpdateFcm(string Fcm)

346 │ {

347 │ var ur = dbcontext.Set<SysUser>().Where(t => t.Ma == objToken!.userid && !(t.Deleted ?? false)).FirstOrDefau...

348 │ if (ur == null)

349 │ {

350 │ return Json(new ResponseModelError("Không tìm thấy user để cập nhật FCM!"));

351 │ }

352 │

353 │ ur.Fcm = Fcm;

354 │ dbcontext.Update(ur);

355 │ if (dbcontext.SaveChanges() > 0)

356 │ {

357 │ return Json(new ResponseModelSuccess("Thành công"));

358 │ }

359 │

360 │ return Json(new ResponseModelError("Cập nhật FCM thất bại"));

361 │ }

362 │

363 │ [HttpPost]

364 │ [TokenUserCheckHTTP(Permission.NotCheck)]

365 │ public IActionResult AccountInfo()

366 │ {

367 │ var ur = dbcontext.Set<SysUser>().Where(t => t.Ma == objToken!.userid && !(t.Deleted ?? false)).FirstOrDefau...

368 │ if (ur == null)

369 │ {

370 │ return Json(new ResponseModelError("Không tìm thấy thông tin tài khoản!"));

371 │ }

372 │ return Json(new ResponseModelSuccess("", new

373 │ {

374 │ ma = ur.Ma + "",

375 │ ten = ur.FullName + "",

376 │ maphongban = ur.MaPhongBan + "",

377 │ ngaysinh = ur.NgaySinh != null ? string.Format("{0:dd/MM/yyyy}", ur.NgaySinh!) : "",

378 │ diachi = ur.DiaChi + "",

379 │ sdt = ur.SoDienThoai + "",

380 │ }));

381 │ }

382 │

383 │ [HttpPost]

384 │ [TokenUserCheckHTTP(Permission.NotCheck)]

385 │ public ActionResult GetTrustedDevice()

386 │ {

387 │ List<SysTrustedDevice> trustedDevices = dbcontext.SysTrustedDevice.Where(t =>

388 │ !(t.Deleted ?? false) &&

389 │ t.UserId == objToken!.userid).ToList();

390 │ return Json(new ResponseModel(QLBHConst.OK, "", trustedDevices.Select(e => new

391 │ {

392 │ ma = e.Ma + "",

393 │ devicename = e.DeviceName + "",

394 │ createddate = string.Format("{0:dd/MM/yyyy hh:mm}", e.CreatedDate),

395 │ }).ToList()));

396 │ }

397 │

398 │ [HttpPost]

399 │ [TokenUserCheckHTTP(Permission.NotCheck)]

400 │ public ActionResult RevokeTrustedDevice(string Ma)

401 │ {

402 │ var ma = Ma.GetGuid();

403 │ SysTrustedDevice trustedDevice = dbcontext.SysTrustedDevice.Where(t => t.Ma == ma &&

404 │ t.UserId == objToken!.userid).FirstOrDefault();

405 │

406 │ if (trustedDevice != null)

407 │ {

408 │ trustedDevice.Deleted = true;

409 │ trustedDevice.DeletedDate = DateTime.Now;

410 │ dbcontext.SysTrustedDevice.Update(trustedDevice);

411 │ dbcontext.SaveChanges();

412 │ }

413 │ return Json(new ResponseModel(QLBHConst.OK, "Thành công"));

414 │ }

415 │

416 │ [HttpPost]

417 │ [TokenUserCheckHTTP(Permission.NotCheck)]

418 │ public IActionResult getListUsername()

419 │ {

420 │ var lst = (from u in dbcontext.SysUser

421 │ where !(u.Deleted ?? false)

422 │ select new

423 │ {

424 │ ma = u.Ma + "",

425 │ ten = u.FullName + "",

426 │ tenkhongdau = Utility.BoDauTiengViet(u.FullName).ToLower()

427 │ }).ToList();

428 │ return Json(new ResponseModelSuccess("", lst));

429 │ }

430 │ [TokenUserCheckHTTP(Permission.NotCheck)]

431 │ public IActionResult GetUserDeChon()

432 │ {

433 │ var listData = (from u in dbcontext.SysUser

434 │ join pb in dbcontext.TblPhongBan on u.MaPhongBan equals pb.Ma into pbGroup

435 │ from pb in pbGroup.DefaultIfEmpty()

436 │ where !(u.Deleted ?? false)

437 │ let isDuplicate = dbcontext.SysUser

438 │ .Where(su => !(su.Deleted ?? false))

439 │ .GroupBy(su => su.FullName)

440 │ .Any(g => g.Key == u.FullName && g.Count() > 1)

441 │ select new

442 │ {

443 │ ma = u.Ma,

444 │ maCN = u.BranchId,

445 │ maPB = u.MaPhongBan,

446 │ code = u.Msnv + "",

447 │ ten = u.FullName + (isDuplicate && pb != null ? $" ({pb.Code})" : ""),

448 │ trangthai = u.Status,

449 │ }).ToList();

450 │ return Json(new ResponseModel(QLBHConst.OK, "", listData));

451 │ }

452 │

453 │ [HttpPost]

454 │ [TokenUserCheckHTTP(Permission.NotCheck)]

455 │ public IActionResult getListNhanVienTheoChiNhanh(string MaChiNhanh)

456 │ {

457 │ var loadAll = MaChiNhanh.GetGuid().IsEmpty();

458 │ var lst = (from u in dbcontext.SysUser

459 │ where !(u.Deleted ?? false) &&(loadAll || u.BranchId == MaChiNhanh.GetGuid())

460 │ select new

461 │ {

462 │ ma = u.Ma + "",

463 │ msnv = u.Msnv + "",

464 │ ten = u.FullName + "",

465 │ tenkhongdau = Utility.BoDauTiengViet(u.FullName).ToLower()

466 │ }).ToList();

467 │ return Json(new ResponseModelSuccess("", lst));

468 │ }

469 │

470 │ [HttpPost]

471 │ [TokenUserCheckHTTP(Permission.NotCheck)]

472 │ public IActionResult getListNhanVienTheoRole(string MaChiNhanh)

473 │ {

474 │ var loadAll = MaChiNhanh.GetGuid().IsEmpty();

475 │ var lst = (from u in dbcontext.SysUser

476 │ join r in dbcontext.SysUserRole on u.Ma equals r.UserId

477 │ where !(u.Deleted ?? false) && !(r.Deleted ?? false) && (loadAll || r.BranchId == MaChiNhanh.GetG...

478 │ select new

479 │ {

480 │ ma = u.Ma + "",

481 │ msnv = u.Msnv + "",

482 │ ten = u.FullName + "",

483 │ tenkhongdau = Utility.BoDauTiengViet(u.FullName).ToLower()

484 │ }).ToList();

485 │ return Json(new ResponseModelSuccess("", lst.Distinct().ToList()));

486 │ }

487 │

488 │ [HttpPost]

489 │ [TokenUserCheckHTTP(Permission.NotCheck)]

490 │ public IActionResult getListNhanVienTheoRoleFull()

491 │ {

492 │ var lst = (from u in dbcontext.SysUser

493 │ join pb in dbcontext.TblPhongBan on u.MaPhongBan equals pb.Ma

494 │ join r in dbcontext.SysUserRole on u.Ma equals r.UserId

495 │ join cn in dbcontext.SysBranch on r.BranchId equals cn.Ma

496 │ where !(u.Deleted ?? false) && !(r.Deleted ?? false)

497 │ let isDuplicate = dbcontext.SysUser

498 │ .Where(su => !(su.Deleted ?? false))

499 │ .GroupBy(su => su.FullName)

500 │ .Any(g => g.Key == u.FullName && g.Count() > 1)

501 │ select new

502 │ {

503 │ ma = u.Ma,

504 │ maCN = cn.Ma,

505 │ maPB = u.MaPhongBan,

506 │ code = u.Msnv + "",

507 │ ten = u.FullName + (isDuplicate && pb != null ? $" ({pb.Code})" : ""),

508 │ trangthai = u.Status,

509 │ }).ToList();

510 │ return Json(new ResponseModelSuccess("", lst.Distinct().ToList()));

511 │ }

512 │

513 │ //---------------lấy thông tin người dùng cho app --------------//

514 │ [HttpPost]

515 │ [TokenUserCheckHTTP(Permission.NotCheck)]

516 │ public IActionResult GetCurrentUser()

517 │ {

518 │ var user = dbcontext.SysUser

519 │ .FirstOrDefault(t => t.Ma == objToken!.userid && !(t.Deleted ?? false));

520 │ if (user == null)

521 │ {

522 │ return Json(new ResponseModelError("Không tìm thấy thông tin người dùng!"));

523 │ }

524 │

525 │ // Lấy tên phòng ban

526 │ var phongBan = dbcontext.TblPhongBan

527 │ .FirstOrDefault(p => p.Ma == user.MaPhongBan && !(p.Deleted ?? false));

528 │

529 │ // Lấy chi nhánh từ objToken.branch thay vì user.BranchId

530 │ var chiNhanh = dbcontext.SysBranch

531 │ .FirstOrDefault(c => c.Ma == objToken.branch && !(c.Deleted ?? false));

532 │

533 │ // Xử lý danh sách kho hàng - đồng bộ mã và tên theo branch

534 │ string userKhoHang = user.KhoHang ?? "";

535 │ string khoHangMa = "";

536 │ string tenKhoHang = "";

537 │

538 │ if (!string.IsNullOrEmpty(userKhoHang))

539 │ {

540 │ // Lấy tất cả kho hàng từ chi nhánh hiện tại trước

541 │ var allKhoHangByBranch = dbcontext.StockDanhMucKho

542 │ .Where(k => !(k.Deleted ?? false) && k.MaChiNhanh == objToken.branch)

543 │ .ToDictionary(k => k.Ma, k => k.Ten);

544 │

545 │ // Tách chuỗi kho hàng của user thành danh sách mã kho

546 │ var userKhoHangCodes = userKhoHang.Split(';', StringSplitOptions.RemoveEmptyEntries);

547 │

548 │ // Filter chỉ lấy những mã kho có trong chi nhánh hiện tại

549 │ List<int> validKhoHangCodes = new List<int>();

550 │ List<string> validKhoHangNames = new List<string>();

551 │

552 │ foreach (var ma in userKhoHangCodes)

553 │ {

554 │ int khoMaInt = ma.ToInt();

555 │ if (allKhoHangByBranch.TryGetValue(khoMaInt, out string? ten))

556 │ {

557 │ validKhoHangCodes.Add(khoMaInt);

558 │ validKhoHangNames.Add(ten);

559 │ }

560 │ }

561 │

562 │ // Tạo chuỗi mã kho và tên kho đã được filter và đồng bộ

563 │ if (validKhoHangCodes.Any())

564 │ {

565 │ khoHangMa = ";" + string.Join(";", validKhoHangCodes) + ";";

566 │ tenKhoHang = ";" + string.Join(";", validKhoHangNames) + ";";

567 │ }

568 │ }

569 │

570 │ return Json(new ResponseModelSuccess("", new

571 │ {

572 │ ma = user.Ma.ToString(),

573 │ username = user.Username ?? "",

574 │ fullName = user.FullName ?? "",

575 │ msnv = user.Msnv ?? "",

576 │ birthDate = user.NgaySinh?.ToString("dd/MM/yyyy") ?? "",

577 │ gioiTinh = user.GioiTinh,

578 │ address = user.DiaChi ?? "",

579 │ phone = user.SoDienThoai ?? "",

580 │ avatarUrl = user.HinhAnh ?? "/assets/images/avatar.png",

581 │ khoHang = khoHangMa,

582 │ tenKhoHang = tenKhoHang,

583 │ ngayVaoLam = user.NgayVaoLam?.ToString("dd/MM/yyyy") ?? "",

584 │ phongBanId = user.MaPhongBan?.ToString() ?? "",

585 │ phongBanName = phongBan?.Ten ?? "",

586 │ branchId = objToken?.branch.ToString() ?? "", // Sử dụng objToken.branch

587 │ branchName = chiNhanh?.Ten ?? "",

588 │ emailReceive = user.EmailReceive ?? false,

589 │ reminder = user.Reminder ?? false

590 │ }));

591 │ }

592 │

593 │ }

594 │ }

────────────────────────────────────────────────────────────

**📄 UserRoleController.cs (CS, 87 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using QuanLyBanHang.Common;

3 │ using QuanLyBanHang.Common.Attributes;

4 │ using QuanLyBanHang.Models.ExModel.SYS;

5 │ using QuanLyBanHang.Models.QLBHEntity;

6 │

7 │ namespace QuanLyBanHang.Controllers.SYS

8 │ {

9 │ public class UserRoleController : BaseController

10 │ {

11 │ public UserRoleController(QuanLyBanHangContext dbcontext, MemoryCacheManager memoryCache) : base(dbcontext, memo...

12 │ {

13 │ m\_MenuID = MenuID.sysUserRole.GetEnumGuid();

14 │ }

15 │

16 │ [TokenUserCheckHTTP()]

17 │ public IActionResult Detail(string UserID)

18 │ {

19 │ ViewData["ListData"] = new sys\_UserRoleModel(dbcontext).GetDataFollowUserID(UserID.GetGuid());

20 │ ViewData["ListRole"] = new sys\_RoleModel(dbcontext).GetData();

21 │ ViewData["UserItem"] = new sys\_UserModel(dbcontext).GetItem(UserID.GetGuid());

22 │ ViewData["ListBranch"] = dbcontext.SysBranch.ToList();

23 │ return View("index");

24 │ }

25 │

26 │ [TokenUserCheckHTTP(Permission.Add)]

27 │ public IActionResult Add(string UserID)

28 │ {

29 │ ViewData["UserItem"] = new sys\_UserModel(dbcontext).GetItem(UserID.GetGuid());

30 │ ViewData["ListRole"] = new sys\_RoleModel(dbcontext).GetData().Where(t => t.Status ?? false).ToList();

31 │ bool bShowAll = Utility.checkPermissionIDModel(objToken.userid, m\_MenuID, Permission.ShowAll.GetEnumGuid());

32 │ if (bShowAll)

33 │ ViewData["ListBranch"] = dbcontext.SysBranch.Where(t => !(t.Deleted ?? false) && (t.Status ?? false)).To...

34 │ else

35 │ ViewData["ListBranch"] = dbcontext.SysBranch.Where(t => !(t.Deleted ?? false) && (t.Status ?? false)

36 │ && t.Ma == objToken.branch).ToList();

37 │ return View("Form");

38 │ }

39 │

40 │ [TokenUserCheckHTTP(Permission.Edit)]

41 │ public IActionResult Edit(string ID)

42 │ {

43 │ SysUserRole drUR = new sys\_UserRoleModel(dbcontext).GetItem(ID.GetGuid());

44 │ ViewData["item"] = drUR;

45 │ ViewData["ListRole"] = new sys\_RoleModel(dbcontext).GetData().Where(t => t.Status ?? false).ToList();

46 │

47 │ if (drUR != null)

48 │ ViewData["UserItem"] = new sys\_UserModel(dbcontext).GetItem(drUR.UserId);

49 │ bool bShowAll = Utility.checkPermissionIDModel(objToken.userid, m\_MenuID, Permission.ShowAll.GetEnumGuid());

50 │ if (bShowAll)

51 │ ViewData["ListBranch"] = dbcontext.SysBranch.Where(t => !(t.Deleted ?? false) && (t.Status ?? false)).To...

52 │ else

53 │ ViewData["ListBranch"] = dbcontext.SysBranch.Where(t => !(t.Deleted ?? false) && (t.Status ?? false)

54 │ && t.Ma == objToken.branch).ToList();

55 │

56 │ return View("Form");

57 │ }

58 │

59 │ [HttpPost]

60 │ [TokenUserCheckHTTP(Permission.Edit)]

61 │ public IActionResult UpdateAPI(string ID, string RoleID, string Status, string Branch)

62 │ {

63 │ int result = new sys\_UserRoleModel(dbcontext).Edit(ID.GetGuid(), RoleID.GetGuid(),

64 │ Status == "1",

65 │ objToken!.userid, Branch.GetGuid());

66 │ return Json(Utility.MakeResponseFromEditResult(Cast.ToString(result)));

67 │ }

68 │

69 │ [HttpPost]

70 │ [TokenUserCheckHTTP(Permission.Add)]

71 │ public IActionResult AddAPI(string UserID, string RoleID, string Status, string Branch)

72 │ {

73 │ int result = new sys\_UserRoleModel(dbcontext).Add(UserID.GetGuid(), RoleID.GetGuid(),

74 │ Status == "1",

75 │ objToken!.userid, Branch.GetGuid());

76 │ return Json(Utility.MakeResponseFromAddResult(Cast.ToString(result)));

77 │ }

78 │

79 │ [HttpPost]

80 │ [TokenUserCheckHTTP(Permission.Delete)]

81 │ public IActionResult DeleteAPI(string ID)

82 │ {

83 │ string result = Cast.ToString(new sys\_UserRoleModel(dbcontext).Delete(ID.GetGuid(), objToken!.userid));

84 │ return Json(Utility.MakeResponseFromDeleteResult(Cast.ToString(result)));

85 │ }

86 │ }

87 │ }

────────────────────────────────────────────────────────────

**📄 Utility.cs (CS, 975 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc.ViewFeatures;

2 │ using Microsoft.EntityFrameworkCore;

3 │ using Newtonsoft.Json;

4 │ using Newtonsoft.Json.Linq;

5 │ using QuanLyBanHang.Controllers;

6 │ using QuanLyBanHang.Models.APIModels;

7 │ using QuanLyBanHang.Models.QLBHEntity;

8 │ using System.Data;

9 │ using System.Diagnostics;

10 │ using System.Globalization;

11 │ using System.Net;

12 │ using System.Reflection;

13 │ using System.Security.Cryptography;

14 │ using System.Text;

15 │ using System.Text.RegularExpressions;

16 │ using System.Numerics;

17 │ using Microsoft.AspNetCore.SignalR;

18 │ using System.Security.Cryptography.X509Certificates;

19 │ using QuanLyBanHang.Models;

20 │

21 │ namespace QuanLyBanHang.Common

22 │ {

23 │ class HisHhhContextE : QuanLyBanHangContext

24 │ {

25 │ public string connection { get; set; } = "";

26 │ protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder) => optionsBuilder.UseSqlServer(con...

27 │ }

28 │ public class Utility

29 │ {

30 │ public static string Key = "Qu@nlyC0ngv13c";

31 │ private static readonly string KeyHash = "Qu@nlyC0ngv13c-Your-Super-Secret-And-Long-Passphrase-Here-Change-This!...

32 │ public static Random random = new Random();

33 │ public static QuanLyBanHangContext \_QLBHContext;

34 │ public static Guid defaultUID = new Guid();

35 │ private static readonly List<string> ImageExt = new List<string> { ".JPG", ".JPE", ".BMP", ".GIF", ".PNG" };

36 │ public static readonly List<string> AllowDocumentExt = new List<string> { ".JPG", ".JPE", ".BMP", ".GIF", ".PNG"

37 │ , ".DOC", ".DOCX", ".XLS", ".XLSX", ".PDF", ".PPT", ".PPTX" };

38 │ public static int MaxChunkSize = 1 \* 1024 \* 1024; // 1 MB;

39 │ public static string UploadHost = "https://file.hungduy.vn/file";

40 │ public static string UploadHostLocal = "http://172.16.10.4:5001/file";

41 │ private static readonly string base62 = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz";

42 │

43 │ private static IHubContext<ChatHUB> hubContext;

44 │

45 │ private const int KeyBitSize = 256; // AES-256

46 │ private const int NonceBitSize = 96; // 12 bytes, khuyến nghị cho AES-GCM

47 │ private const int TagBitSize = 128; // 16 bytes, khuyến nghị cho AES-GCM

48 │ private const int SaltBitSize = 128; // 16 bytes, cho PBKDF2

49 │ private const int Iterations = 100000; // Số lần lặp cho PBKDF2, càng cao càng tốt

50 │

51 │ public static bool TryDecryptToInt(string encrypted, out int result)

52 │ {

53 │ try

54 │ {

55 │ result = encrypted.Decrypt().ToInt();

56 │ return true;

57 │ }

58 │ catch

59 │ {

60 │ result = 0;

61 │ return false;

62 │ }

63 │ }

64 │ public static string RemoveSpecialCharacters(string input)

65 │ {

66 │ string pattern = @"[^a-zA-Z0-9\_áàảãạăắằẳẵặâấầẩẫậđéèẻẽẹêếềểễ��...

67 │ Regex r = new Regex(pattern);

68 │ return r.Replace(input + "", string.Empty);

69 │ }

70 │ public static string RandomStringNumber(int length)

71 │ {

72 │ const string chars = "0123456789";

73 │ return new string(Enumerable.Repeat(chars, length)

74 │ .Select(s => s[random.Next(s.Length)]).ToArray());

75 │ }

76 │ public static void initConnection(string connection)

77 │ {

78 │ \_QLBHContext = new HisHhhContextE()

79 │ {

80 │ connection = connection

81 │ };

82 │ }

83 │ public static void ThongBaoTelegram(string Message, string GroupName)

84 │ {

85 │ if (!(GroupName.Trim().IsEmpty() || Message.Trim().IsEmpty()))

86 │ {

87 │ hubContext.Clients.All.SendAsync("NotificationTele", Message, GroupName);

88 │ }

89 │ }

90 │ public static string RandomString(int length)

91 │ {

92 │ const string chars = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789";

93 │ return new string(Enumerable.Repeat(chars, length)

94 │ .Select(s => s[random.Next(s.Length)]).ToArray());

95 │ }

96 │ public static string RandomNumber(int length)

97 │ {

98 │ const string chars = "0123456789";

99 │ return new string(Enumerable.Repeat(chars, length)

100 │ .Select(s => s[random.Next(s.Length)]).ToArray());

101 │ }

102 │ public static string CheckPath(string path)

103 │ {

104 │ if (!Directory.Exists(path))

105 │ {

106 │ Directory.CreateDirectory(path);

107 │ }

108 │ return path;

109 │ }

110 │ public static T getViewData<T>(Microsoft.AspNetCore.Mvc.ViewFeatures.ViewDataDictionary<dynamic> d, string key)

111 │ {

112 │ try

113 │ {

114 │ return (T)(d[key] ?? Activator.CreateInstance<T>());

115 │ }

116 │ catch

117 │ {

118 │ return Activator.CreateInstance<T>();

119 │ }

120 │ }

121 │

122 │

123 │ public static string Encrypt(string toEncrypt)

124 │ {

125 │ try

126 │ {

127 │ byte[] toEncryptArray = Encoding.UTF8.GetBytes(toEncrypt);

128 │ using var hash = MD5.Create();

129 │ using var tripleDes = TripleDES.Create();

130 │ tripleDes.Key = hash.ComputeHash(Encoding.UTF8.GetBytes(Key));

131 │ tripleDes.Mode = CipherMode.ECB;

132 │ tripleDes.Padding = PaddingMode.PKCS7;

133 │

134 │ ICryptoTransform cTransform = tripleDes.CreateEncryptor();

135 │ byte[] resultArray = cTransform.TransformFinalBlock(toEncryptArray, 0, toEncryptArray.Length);

136 │ return ToBase62(resultArray);

137 │ }

138 │ catch

139 │ {

140 │ return "Error";

141 │ }

142 │ }

143 │

144 │ public static string Decrypt(string toDecrypt)

145 │ {

146 │ try

147 │ {

148 │ byte[] toEncryptArray = FromBase62(toDecrypt ?? "");

149 │

150 │ using var hash = MD5.Create();

151 │ using var tripleDes = TripleDES.Create();

152 │ tripleDes.Key = hash.ComputeHash(Encoding.UTF8.GetBytes(Key));

153 │ tripleDes.Mode = CipherMode.ECB;

154 │ tripleDes.Padding = PaddingMode.PKCS7;

155 │

156 │ ICryptoTransform cTransform = tripleDes.CreateDecryptor();

157 │ byte[] resultArray = cTransform.TransformFinalBlock(toEncryptArray, 0, toEncryptArray.Length);

158 │ var result = Encoding.UTF8.GetString(resultArray);

159 │ return result;

160 │ }

161 │ catch

162 │ {

163 │ return "Error";

164 │ }

165 │ }

166 │

167 │

168 │ private static string ToBase62(byte[] bytes)

169 │ {

170 │ var num = new BigInteger(bytes.Append((byte)0).ToArray());

171 │ var sb = new StringBuilder();

172 │ while (num > 0)

173 │ {

174 │ sb.Insert(0, base62[(int)(num % 62)]);

175 │ num /= 62;

176 │ }

177 │ return sb.ToString();

178 │ }

179 │ private static byte[] FromBase62(string str)

180 │ {

181 │ BigInteger num = 0;

182 │ foreach (var c in str)

183 │ num = num \* 62 + base62.IndexOf(c);

184 │

185 │ var bytes = num.ToByteArray();

186 │

187 │ if (bytes.Length >= 2 && bytes[^1] == 0)

188 │ bytes = bytes[..^1];

189 │

190 │ if (bytes.Length % 8 != 0)

191 │ {

192 │ var padded = new byte[(bytes.Length + 7) / 8 \* 8];

193 │ Array.Copy(bytes, padded, bytes.Length);

194 │ bytes = padded;

195 │ }

196 │

197 │ return bytes;

198 │ }

199 │

200 │

201 │

202 │

203 │ public static bool IsImage(string Extension)

204 │ {

205 │ try

206 │ {

207 │ if (ImageExt.Contains(Extension.ToUpper()))

208 │ return true;

209 │ return false;

210 │ }

211 │ catch

212 │ {

213 │ return false;

214 │ }

215 │ }

216 │ public static string SetItemMultiSelect(object obj)

217 │ {

218 │ return string.Format(";{0};", Cast.ToString(obj).Replace(",", ";"));

219 │ }

220 │ public static string GetSHA256(string str)

221 │ {

222 │ return GetSHA(str, SHA256.Create());

223 │ }

224 │ public static string GetSHA512(string str)

225 │ {

226 │ return GetSHA(str, SHA512.Create());

227 │ }

228 │ public static string GetMD5(string str)

229 │ {

230 │ return GetSHA(str, MD5.Create());

231 │ }

232 │ private static string GetSHA(string str, HashAlgorithm hash)

233 │ {

234 │ try

235 │ {

236 │ string[] value = (from x in hash.ComputeHash(Encoding.UTF8.GetBytes(str))

237 │ select x.ToString("X2")).ToArray<string>();

238 │ return string.Join("", value).ToUpper();

239 │ }

240 │ catch

241 │ {

242 │ }

243 │ return "";

244 │ }

245 │ public static bool IsNull(object o)

246 │ {

247 │ return o == null || Convert.IsDBNull(o);

248 │ }

249 │ public static string Base64Encode(string plainText)

250 │ {

251 │ var plainTextBytes = Encoding.UTF8.GetBytes(plainText);

252 │ return Convert.ToBase64String(plainTextBytes);

253 │ }

254 │ public static string Base64Decode(string base64EncodedData)

255 │ {

256 │ var base64EncodedBytes = Convert.FromBase64String(base64EncodedData);

257 │ return Encoding.UTF8.GetString(base64EncodedBytes);

258 │ }

259 │ public static bool checkTokenPermission(ViewDataDictionary<dynamic> ViewData, MenuID menu, Permission permission...

260 │ {

261 │ return checkTokenPermission(ViewData, menu.GetEnumGuid(), permission.GetEnumGuid());

262 │ }

263 │ public static bool checkTokenPermission(ViewDataDictionary<dynamic> ViewData, Guid menu, Guid permission)

264 │ {

265 │ try

266 │ {

267 │ SysUser? currentUser = ViewData["CurrentUser"] as SysUser;

268 │

269 │ if (currentUser == null)

270 │ {

271 │ Console.WriteLine("CurrentUser is null");

272 │ return false;

273 │ }

274 │

275 │ if (\_QLBHContext == null)

276 │ {

277 │ Console.WriteLine("\_QLBHContext is null");

278 │ return false;

279 │ }

280 │

281 │ if (\_QLBHContext.VPermission == null)

282 │ {

283 │ Console.WriteLine("VPermission is null");

284 │ return false;

285 │ }

286 │

287 │ if (currentUser.Ma == null)

288 │ {

289 │ Console.WriteLine("currentUser.Ma is null");

290 │ return false;

291 │ }

292 │

293 │ return \_QLBHContext.VPermission.Any(t =>

294 │ t.UserId == currentUser.Ma &&

295 │ t.PermissionId == permission &&

296 │ t.MenuId == menu

297 │ );

298 │ }

299 │ catch (Exception ex)

300 │ {

301 │ Console.WriteLine($"Exception: {ex.Message}");

302 │ return false;

303 │ }

304 │ }

305 │ public static bool checkTokenPermission(ViewDataDictionary<dynamic> ViewData, Guid permission)

306 │ {

307 │ if (!ViewData.ContainsKey("CurrentUser") || !ViewData.ContainsKey("currentMenu"))

308 │ return false;

309 │

310 │ var currentUser = ViewData["CurrentUser"] as SysUser;

311 │ if (currentUser == null)

312 │ return false;

313 │

314 │ Guid currentMenu;

315 │ try

316 │ {

317 │ currentMenu = ViewData["currentMenu"] is Guid guid

318 │ ? guid

319 │ : Guid.Parse(ViewData["currentMenu"].ToString());

320 │ }

321 │ catch

322 │ {

323 │ return false;

324 │ }

325 │

326 │ if (currentMenu == defaultUID) return false;

327 │

328 │ if (\_QLBHContext == null) return false; // make sure context is initialized

329 │

330 │ return \_QLBHContext.VPermission.Any(t =>

331 │ t.UserId == currentUser.Ma &&

332 │ t.PermissionId == permission &&

333 │ t.MenuId == currentMenu

334 │ );

335 │ }

336 │ public static bool checkTokenPermission(UserToken URtoken, Guid menu, Guid permission)

337 │ {

338 │ return \_QLBHContext.VPermission.Any(t => t.UserId == URtoken.userid && t.PermissionId == permission && t.Men...

339 │ }

340 │

341 │ public static bool CheckRole(ViewDataDictionary<dynamic> ViewData, Guid roleid)

342 │ {

343 │ try

344 │ {

345 │ if (!ViewData.ContainsKey("CurrentUser") || !ViewData.ContainsKey("currentMenu"))

346 │ return false;

347 │

348 │ SysUser? currentUser = ViewData["CurrentUser"] as SysUser;

349 │ if (currentUser == null)

350 │ return false;

351 │

352 │ Guid currentMenu;

353 │ try

354 │ {

355 │ currentMenu = ViewData["currentMenu"] is Guid g

356 │ ? g

357 │ : Guid.Parse(ViewData["currentMenu"].ToString());

358 │ }

359 │ catch

360 │ {

361 │ return false; // couldn't parse menu id

362 │ }

363 │

364 │ if (currentMenu == defaultUID)

365 │ return false;

366 │

367 │ if (\_QLBHContext == null || \_QLBHContext.VPermission == null)

368 │ return false;

369 │

370 │ return \_QLBHContext.VPermission.Any(t =>

371 │ t.RoleId == roleid &&

372 │ t.MenuId == currentMenu &&

373 │ t.UserId == currentUser.Ma

374 │ );

375 │ }

376 │ catch

377 │ {

378 │ return false;

379 │ }

380 │ }

381 │ public static ResponseModel? CheckParam(params object[] entities)

382 │ {

383 │ try

384 │ {

385 │ foreach (object x in entities)

386 │ {

387 │ if (x == null)

388 │ {

389 │ return new ResponseModelError("Vui lòng nhập đầy đủ dữ liệu");

390 │ }

391 │ if (x.GetType() == typeof(string) && ((string)x) == "")

392 │ {

393 │ return new ResponseModelError("Vui lòng nhập " + x.GetType().Name);

394 │ }

395 │ }

396 │ }

397 │ catch (Exception e)

398 │ {

399 │ return new ResponseModelError("Đã xảy ra lỗi khi kiểm tra biến " + e.Message);

400 │ }

401 │ return null;

402 │ }

403 │ public static ResponseModel MakeResponseFromAddResult(string result)

404 │ {

405 │ if (result == "1")//Success

406 │ {

407 │ return QLBHConst.ResponseAddSuccessfully;

408 │ }

409 │ else if (result == "0")//Failure

410 │ {

411 │ return QLBHConst.ResponseAddFailure;

412 │ }

413 │ else if (result == "-3")//Duplicate

414 │ {

415 │ return QLBHConst.ResponseObjectExists;

416 │ }

417 │ else if (result == "-1")//Duplicate

418 │ {

419 │ return QLBHConst.ResponseObjectExists;

420 │ }

421 │ else if (result == "-2")//Duplicate

422 │ {

423 │ return QLBHConst.ResponseObjectExists;

424 │ }

425 │ else if (result == "-4")//Duplicate

426 │ {

427 │ return QLBHConst.ResponseObjectExists;

428 │ }

429 │ else if (result == "-5")//Duplicate

430 │ {

431 │ return QLBHConst.ResponseObjectExists;

432 │ }

433 │ return QLBHConst.ResponseHaveError;

434 │ }

435 │ public static ResponseModel MakeResponseFromEditResult(string result)

436 │ {

437 │ if (result == "1")//Success

438 │ {

439 │ return QLBHConst.ResponseUpdateSuccessfully;

440 │ }

441 │

442 │ else if (result == "0")//Failure

443 │ {

444 │ return QLBHConst.ResponseUpdateFailure;

445 │ }

446 │

447 │ else if (result == "-2")//Not exist

448 │ {

449 │ return QLBHConst.ResponseNotExist;

450 │ }

451 │

452 │ else if (result == "-3")//Duplicate

453 │ {

454 │ return QLBHConst.ResponseObjectExists;

455 │ }

456 │

457 │ else if (result == "-4")//Duplicate

458 │ {

459 │ return QLBHConst.ResponseCodetCannotEdit;

460 │ }

461 │ return QLBHConst.ResponseHaveError;

462 │ }

463 │ public static ResponseModel MakeResponseFromDeleteResult(string result)

464 │ {

465 │ if (result == "1")//Success

466 │ {

467 │ return QLBHConst.ResponseDeleteSuccessfully;

468 │ }

469 │ else if (result == "0")//Failure

470 │ {

471 │ return QLBHConst.ResponseDeleteFailure;

472 │ }

473 │ else if (result == "-1")//Cannot delete

474 │ {

475 │ return QLBHConst.ResponseCannotDelete;

476 │ }

477 │ else if (result == "-2")//Not exist

478 │ {

479 │ return QLBHConst.ResponseNotExist;

480 │ }

481 │ else if (result == "-5")

482 │ {

483 │ return new ResponseModel(QLBHConst.Error, "Dữ liệu đang được sử dụng.");

484 │ }

485 │ return QLBHConst.ResponseHaveError;

486 │ }

487 │ public static string NumberToText(double inputNumber, bool suffix = true)

488 │ {

489 │ string[] unitNumbers = new string[] { "không", "một", "hai", "ba", "bốn", "năm", "sáu", "bảy", "tá...

490 │ string[] placeValues = new string[] { "", "nghìn", "triệu", "tỷ" };

491 │ bool isNegative = false;

492 │

493 │ // -12345678.3445435 => "-12345678"

494 │ string sNumber = inputNumber.ToString("#");

495 │ if (sNumber == "")

496 │ {

497 │ sNumber = "0";

498 │

499 │ }

500 │ double number = Convert.ToDouble(sNumber);

501 │ if (number < 0)

502 │ {

503 │ number = -number;

504 │ sNumber = number.ToString();

505 │ isNegative = true;

506 │ }

507 │

508 │

509 │ int ones, tens, hundreds;

510 │

511 │ int positionDigit = sNumber.Length; // last -> first

512 │

513 │ string result = " ";

514 │

515 │

516 │ if (positionDigit == 0)

517 │ result = unitNumbers[0] + result;

518 │ else

519 │ {

520 │ // 0: ###

521 │ // 1: nghìn ###,###

522 │ // 2: triệu ###,###,###

523 │ // 3: tỷ ###,###,###,###

524 │ int placeValue = 0;

525 │

526 │ while (positionDigit > 0)

527 │ {

528 │ // Check last 3 digits remain ### (hundreds tens ones)

529 │ tens = hundreds = -1;

530 │ ones = Convert.ToInt32(sNumber.Substring(positionDigit - 1, 1));

531 │ positionDigit--;

532 │ if (positionDigit > 0)

533 │ {

534 │ tens = Convert.ToInt32(sNumber.Substring(positionDigit - 1, 1));

535 │ positionDigit--;

536 │ if (positionDigit > 0)

537 │ {

538 │ hundreds = Convert.ToInt32(sNumber.Substring(positionDigit - 1, 1));

539 │ positionDigit--;

540 │ }

541 │ }

542 │

543 │ if ((ones > 0) || (tens > 0) || (hundreds > 0) || (placeValue == 3))

544 │ result = placeValues[placeValue] + result;

545 │

546 │ placeValue++;

547 │ if (placeValue > 3) placeValue = 1;

548 │

549 │ if ((ones == 1) && (tens > 1))

550 │ result = "một " + result;

551 │ else

552 │ {

553 │ if ((ones == 5) && (tens > 0))

554 │ result = "lăm " + result;

555 │ else if (ones > 0)

556 │ result = unitNumbers[ones] + " " + result;

557 │ }

558 │ if (tens < 0)

559 │ break;

560 │ else

561 │ {

562 │ if ((tens == 0) && (ones > 0)) result = "lẻ " + result;

563 │ if (tens == 1) result = "mười " + result;

564 │ if (tens > 1) result = unitNumbers[tens] + " mươi " + result;

565 │ }

566 │ if (hundreds < 0) break;

567 │ else

568 │ {

569 │ if ((hundreds > 0) || (tens > 0) || (ones > 0))

570 │ result = unitNumbers[hundreds] + " trăm " + result;

571 │ }

572 │ result = " " + result;

573 │ }

574 │ }

575 │ result = result.Trim();

576 │ if (isNegative) result = "Âm " + result;

577 │ return result;

578 │ }

579 │ public static string BoDauTiengViet(string s)

580 │ {

581 │ Regex regex = new Regex("\\p{IsCombiningDiacriticalMarks}+");

582 │ string temp = (s + "").Normalize(NormalizationForm.FormD);

583 │ return regex.Replace(temp, String.Empty).Replace('\u0111', 'd').Replace('\u0110', 'D');

584 │ }

585 │ public static string ByteArrayToString(byte[] ba)

586 │ {

587 │ StringBuilder hex = new StringBuilder(ba.Length \* 2);

588 │ foreach (byte b in ba)

589 │ hex.AppendFormat("{0:x2}", b);

590 │ return hex.ToString();

591 │ }

592 │ public static bool checkPermissionIDModel(Guid userid, Guid menu, Guid permission)

593 │ {

594 │ return \_QLBHContext.VPermission.Any(t => t.MenuId == menu && t.UserId == userid && t.PermissionId == permiss...

595 │ }

596 │ public static string ChuyenSoSangLaMa(int number)

597 │ {

598 │ if ((number < 0) || (number > 3999)) throw new ArgumentOutOfRangeException("insert value betwheen 1 and 3999...

599 │ if (number < 1) return string.Empty;

600 │ if (number >= 1000) return "M" + ChuyenSoSangLaMa(number - 1000);

601 │ if (number >= 900) return "CM" + ChuyenSoSangLaMa(number - 900);

602 │ if (number >= 500) return "D" + ChuyenSoSangLaMa(number - 500);

603 │ if (number >= 400) return "CD" + ChuyenSoSangLaMa(number - 400);

604 │ if (number >= 100) return "C" + ChuyenSoSangLaMa(number - 100);

605 │ if (number >= 90) return "XC" + ChuyenSoSangLaMa(number - 90);

606 │ if (number >= 50) return "L" + ChuyenSoSangLaMa(number - 50);

607 │ if (number >= 40) return "XL" + ChuyenSoSangLaMa(number - 40);

608 │ if (number >= 10) return "X" + ChuyenSoSangLaMa(number - 10);

609 │ if (number >= 9) return "IX" + ChuyenSoSangLaMa(number - 9);

610 │ if (number >= 5) return "V" + ChuyenSoSangLaMa(number - 5);

611 │ if (number >= 4) return "IV" + ChuyenSoSangLaMa(number - 4);

612 │ if (number >= 1) return "I" + ChuyenSoSangLaMa(number - 1);

613 │ throw new ArgumentOutOfRangeException("something bad happened");

614 │ }

615 │ public static async Task<bool> sendAsync(string log, string? controller = null, string? projectId = null)

616 │ {

617 │ StackTrace stackTrace = new StackTrace();

618 │ MethodBase method = stackTrace.GetFrame(1).GetMethod();

619 │ using HttpClient client = new HttpClient();

620 │ client.BaseAddress = new Uri("https://172.29.12.65/setLog");

621 │ client.DefaultRequestHeaders.Clear();

622 │ client.DefaultRequestHeaders.Host = "log.hungduy.vn";

623 │ return (await client.SendAsync(new HttpRequestMessage

624 │ {

625 │ Method = HttpMethod.Post,

626 │ Content = new StringContent(JsonConvert.SerializeObject(new

627 │ {

628 │ projectId = (projectId ?? method?.ReflectedType?.Namespace ?? string.Empty),

629 │ log = log,

630 │ controller = controller

631 │ }), Encoding.UTF8, "application/json")

632 │ })).IsSuccessStatusCode;

633 │ }

634 │ public static async Task<string> UploadFile(byte[] file, string filename = "")

635 │ {

636 │ if (file == null)

637 │ {

638 │ return "";

639 │ }

640 │ string obj = Base64Encode(JsonConvert.SerializeObject(

641 │ new UserToken(Guid.NewGuid(), "ServerRequest")));

642 │ string sign = GetSHA256(obj + AppController.passcode + DateTime.Now.ToString("yyyyMM"));

643 │ string token = obj + "." + sign;

644 │ if (filename == null || filename == string.Empty)

645 │ {

646 │ filename = RandomString(20) + ".pdf";

647 │ }

648 │ List<Task<bool>> tasks = new List<Task<bool>>();

649 │ int totalChunks = (int)Math.Ceiling((double)file.Length / MaxChunkSize);

650 │ byte[] buffer = new byte[MaxChunkSize];

651 │ List<byte[]> chunks = new();

652 │ int currentChunk = 1;

653 │ try

654 │ {

655 │ int currentChunkSize = Math.Min(MaxChunkSize, file.Length);

656 │ byte[] chunk = new byte[currentChunkSize];

657 │ Array.Copy(file, 0, chunk, 0, currentChunkSize);

658 │ await sendChunk(chunk, filename, "1", totalChunks + "", token);

659 │ currentChunk++;

660 │ }

661 │ catch (Exception e)

662 │ {

663 │ JObject objx = JsonConvert.DeserializeObject<JObject>(e.Message);

664 │ return (string)objx["message"];

665 │ }

666 │ if (totalChunks > 1)

667 │ {

668 │ for (int offset = MaxChunkSize; offset < file.Length; offset += MaxChunkSize)

669 │ {

670 │ int currentChunkSize = Math.Min(MaxChunkSize, (file.Length - offset));

671 │ byte[] chunk = new byte[currentChunkSize];

672 │ Array.Copy(file, offset, chunk, 0, currentChunkSize);

673 │ tasks.Add(sendChunk(chunk, filename, currentChunk + "", totalChunks + "", token));

674 │ currentChunk++;

675 │ }

676 │ // try

677 │ // {

678 │ await Task.WhenAll(tasks);

679 │ // }

680 │ // catch (Exception e)

681 │ // {

682 │ // JObject objx = JsonConvert.DeserializeObject<JObject>(e.Message);

683 │ // return (string)objx["message"];

684 │ // }

685 │ }

686 │ if (!tasks.Any(t => !t.Result))

687 │ {

688 │ using (var client = new HttpClient())

689 │ {

690 │ client.DefaultRequestHeaders.Add("token", token);

691 │ using (var form = new MultipartFormDataContent())

692 │ {

693 │ form.Add(new StringContent(filename), "filename");

694 │ form.Add(new StringContent(totalChunks + ""), "totalChunks");

695 │ HttpResponseMessage response = await client.PostAsync(UploadHostLocal + "/CompleteUpload", form)...

696 │ if (response.IsSuccessStatusCode)

697 │ {

698 │ var s = await response.Content.ReadAsStringAsync();

699 │ if (s.Contains("successfully"))

700 │ {

701 │ JObject objx = JsonConvert.DeserializeObject<JObject>(s);

702 │ return UploadHost + objx["filename"];

703 │ }

704 │ }

705 │ }

706 │ }

707 │ }

708 │ return "";

709 │ }

710 │ private static async Task<bool> sendChunk(byte[] chunk, string filename, string chunkNumber, string totalChunks,...

711 │ {

712 │ int count = 0;

713 │ resend:

714 │ try

715 │ {

716 │ using (var content = new ByteArrayContent(chunk))

717 │ {

718 │ using (var client = new HttpClient())

719 │ {

720 │ client.DefaultRequestHeaders.Add("token", token);

721 │ using (var form = new MultipartFormDataContent())

722 │ {

723 │ form.Add(content, "chunk", filename);

724 │ form.Add(new StringContent(chunkNumber), "chunkNumber");

725 │ form.Add(new StringContent(filename), "filename");

726 │ form.Add(new StringContent(totalChunks), "totalChunks");

727 │

728 │ HttpResponseMessage response = await client.PostAsync(UploadHostLocal + "/UploadChunk", form...

729 │ if (!response.IsSuccessStatusCode)

730 │ {

731 │ if (count > 3)

732 │ {

733 │ throw new Exception("sendChunk error");

734 │ }

735 │ count++;

736 │ goto resend;

737 │ //var s = await response.Content.ReadAsStringAsync();

738 │ //throw new Exception(s);

739 │ }

740 │ return true;

741 │ }

742 │ }

743 │ }

744 │ }

745 │ catch

746 │ {

747 │ count++;

748 │ goto resend;

749 │ }

750 │ return false;

751 │ }

752 │

753 │ public static string AddCommas(decimal value, int? numDecimal = null)

754 │ {

755 │ // Giới hạn số chữ số thập phân nếu có yêu cầu

756 │ if (numDecimal.HasValue)

757 │ {

758 │ value = Math.Round(value, numDecimal.Value);

759 │ }

760 │

761 │ // Tách phần nguyên và phần thập phân

762 │ var parts = value.ToString(CultureInfo.InvariantCulture).Split('.');

763 │ var integerPart = parts[0];

764 │ var decimalPart = parts.Length > 1 ? parts[1] : "";

765 │

766 │ // Format phần nguyên với dấu phẩy ngăn cách

767 │ string formattedInteger = string.Format("{0:N0}", long.Parse(integerPart));

768 │

769 │ // Giữ lại số chữ số thập phân nếu cần

770 │ if (numDecimal.HasValue && numDecimal > 0)

771 │ {

772 │ decimalPart = decimalPart.PadRight(numDecimal.Value, '0').Substring(0, numDecimal.Value);

773 │

774 │ // Nếu phần thập phân không phải toàn số 0, giữ lại

775 │ if (int.TryParse(decimalPart, out int decVal) && decVal != 0)

776 │ {

777 │ return $"{formattedInteger}.{decimalPart}";

778 │ }

779 │ }

780 │

781 │ return formattedInteger;

782 │ }

783 │

784 │ public static string GetSHA256WithRSA(string dataToSign, string privateKeyData)

785 │ {

786 │ try

787 │ {

788 │ using var rsa = new RSACryptoServiceProvider();

789 │ rsa.ImportFromPem(privateKeyData);

790 │ var dataBytes = Encoding.UTF8.GetBytes(dataToSign);

791 │ var signatureBytes = rsa.SignData(dataBytes, CryptoConfig.MapNameToOID("SHA256"));

792 │ return Convert.ToBase64String(signatureBytes);

793 │ }

794 │ catch

795 │ {

796 │ return null;

797 │ }

798 │ }

799 │

800 │ public static bool VerifySHA256WithRSA(string originalData, string signature, string publicKeyData)

801 │ {

802 │ try

803 │ {

804 │

805 │ using var rsa = new RSACryptoServiceProvider();

806 │ if (publicKeyData.Contains("BEGIN CERTIFICATE"))

807 │ {

808 │ var cert = new X509Certificate(Encoding.UTF8.GetBytes(publicKeyData));

809 │ var publicKey = cert.GetPublicKey();

810 │ rsa.ImportRSAPublicKey(publicKey, out \_);

811 │ }

812 │ else

813 │ {

814 │ rsa.ImportFromPem(publicKeyData);

815 │ }

816 │ var dataBytes = Encoding.UTF8.GetBytes(originalData);

817 │ var signatureBytes = Convert.FromBase64String(signature);

818 │ return rsa.VerifyData(dataBytes, CryptoConfig.MapNameToOID("SHA256"), signatureBytes);

819 │ }

820 │ catch

821 │ {

822 │ return false;

823 │ }

824 │ }

825 │

826 │ public static string Generate12CharCode()

827 │ {

828 │ DateTime now = DateTime.Now;

829 │ int year = now.Year % 100 % 26;

830 │ int month = now.Month;

831 │ char yearChar = (char)(year + 65);

832 │ char monthChar = (char)(month + 65);

833 │ string timePart = $"{now:ddHHmmssff}";

834 │ return $"{yearChar}{monthChar}{timePart}";

835 │ }

836 │ public static string GetCodeDiaChi(string ten)

837 │ {

838 │ ten = BoDauTiengViet(ten);

839 │ var Reg = new Regex(@"^[0-9a-zA-Z/-]");

840 │ try

841 │ {

842 │ string[] tmps = ten.Split(' ');

843 │ string code = "";

844 │ for (int i = 0; i < tmps.Length; i++)

845 │ {

846 │ if (tmps[i].Trim() != "" && Reg.IsMatch(tmps[i].Trim()))

847 │ {

848 │ code += tmps[i].Trim()[0];

849 │ }

850 │ }

851 │ return code;

852 │ }

853 │ catch

854 │ {

855 │ return "";

856 │ }

857 │ }

858 │ }

859 │

860 │ }

861 │ public class C2ID

862 │ {

863 │ public Guid? ID1 { get; set; }

864 │ public Guid? ID2 { get; set; }

865 │ public override bool Equals(object obj)

866 │ {

867 │ var item = obj as C2ID;

868 │

869 │ if (item == null)

870 │ {

871 │ return false;

872 │ }

873 │

874 │ return this.ID1.Equals(item.ID1) && this.ID2.Equals(item.ID2);

875 │ }

876 │

877 │ public override int GetHashCode()

878 │ {

879 │ return HashCode.Combine(ID1, ID2);

880 │ }

881 │ }

882 │

883 │ public class C2IDSTRING

884 │ {

885 │ public string ID1 { get; set; }

886 │ public string ID2 { get; set; }

887 │ public override bool Equals(object obj)

888 │ {

889 │ var item = obj as C2IDSTRING;

890 │

891 │ if (item == null)

892 │ {

893 │ return false;

894 │ }

895 │

896 │ return this.ID1.Equals(item.ID1) && this.ID2.Equals(item.ID2);

897 │ }

898 │

899 │ public override int GetHashCode()

900 │ {

901 │ return HashCode.Combine(ID1, ID2);

902 │ }

903 │ }

904 │

905 │ public class C2IDDate

906 │ {

907 │ public int ID1 { get; set; }

908 │ public DateTime ID2 { get; set; }

909 │ public override bool Equals(object obj)

910 │ {

911 │ var item = obj as C2IDDate;

912 │

913 │ if (item == null)

914 │ {

915 │ return false;

916 │ }

917 │

918 │ return this.ID1.Equals(item.ID1) && this.ID2.Date == item.ID2.Date;

919 │ }

920 │

921 │ public override int GetHashCode()

922 │ {

923 │ return HashCode.Combine(ID1, ID2);

924 │ }

925 │ }

926 │

927 │ public class C3IDDate

928 │ {

929 │ public Guid ID1 { get; set; }

930 │ public DateTime ID2 { get; set; }

931 │ public int ID3 { get; set; }

932 │ public override bool Equals(object obj)

933 │ {

934 │ var item = obj as C3IDDate;

935 │

936 │ if (item == null)

937 │ {

938 │ return false;

939 │ }

940 │

941 │ return this.ID1.Equals(item.ID1) && this.ID2.Date == item.ID2.Date && this.ID3 == item.ID3;

942 │ }

943 │

944 │ public override int GetHashCode()

945 │ {

946 │ return HashCode.Combine(ID1, ID2, ID3);

947 │ }

948 │ }

949 │

950 │ public class C4ID

951 │ {

952 │ public Guid ID1 { get; set; }

953 │ public Guid ID2 { get; set; }

954 │ public int ID3 { get; set; }

955 │ public Guid ID4 { get; set; }

956 │ public override bool Equals(object obj)

957 │ {

958 │ var item = obj as C4ID;

959 │

960 │ if (item == null)

961 │ {

962 │ return false;

963 │ }

964 │

965 │ return this.ID1.Equals(item.ID1) && this.ID2 == item.ID2 && this.ID3 == item.ID3 && this.ID4 == item.ID4;

966 │ }

967 │

968 │ public override int GetHashCode()

969 │ {

970 │ return HashCode.Combine(ID1, ID2, ID3, ID4);

971 │ }

972 │

973 │

974 │ }

975 │

────────────────────────────────────────────────────────────

**📄 VBanHang.cs (CS, 55 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.VCEntity;

5 │

6 │ public partial class VBanHang

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int MaKh { get; set; }

11 │

12 │ public int MaVoucher { get; set; }

13 │

14 │ public string MaVach { get; set; } = null!;

15 │

16 │ public int GiaTri { get; set; }

17 │

18 │ public int? SoDu { get; set; }

19 │

20 │ public int ThanhTien { get; set; }

21 │

22 │ public DateTime? NgayHieuLuc { get; set; }

23 │

24 │ public DateTime? NgayHetHan { get; set; }

25 │

26 │ public int? NguoiKichHoat { get; set; }

27 │

28 │ public DateTime? NgayKichHoat { get; set; }

29 │

30 │ public bool? TrangThai { get; set; }

31 │

32 │ public DateTime? NgayTao { get; set; }

33 │

34 │ public string TenVoucher { get; set; } = null!;

35 │

36 │ public string LoaiThoiHan { get; set; } = null!;

37 │

38 │ public int GiaTriThoiHan { get; set; }

39 │

40 │ public int DonGia { get; set; }

41 │

42 │ public int MaChiNhanh { get; set; }

43 │

44 │ public int MaNganh { get; set; }

45 │

46 │ public string? MaThe { get; set; }

47 │

48 │ public string? GhiChu { get; set; }

49 │

50 │ public int? BanBoSungTu { get; set; }

51 │

52 │ public int? NhomVoucherId { get; set; }

53 │

54 │ public int? NguoiTao { get; set; }

55 │ }

────────────────────────────────────────────────────────────

**📄 VChiTietKiemKe.cs (CS, 37 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class VChiTietKiemKe

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int? MaKiemKe { get; set; }

11 │

12 │ public int? MaHangHoa { get; set; }

13 │

14 │ public string MaVach { get; set; } = null!;

15 │

16 │ public string? MaKt { get; set; }

17 │

18 │ public string TenHang { get; set; } = null!;

19 │

20 │ public decimal? SoLuongKiemKe { get; set; }

21 │

22 │ public string TenDvt { get; set; } = null!;

23 │

24 │ public string TenLop { get; set; } = null!;

25 │

26 │ public DateTime? NgayKiemKe { get; set; }

27 │

28 │ public string? NguoiKiemKe { get; set; }

29 │

30 │ public string? GhiChu { get; set; }

31 │

32 │ public int? MaKho { get; set; }

33 │

34 │ public string? TenKho { get; set; }

35 │

36 │ public string? SoPhieu { get; set; }

37 │ }

────────────────────────────────────────────────────────────

**📄 VChiTietKiemKe.cs (CS, 47 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class VChiTietKiemKe

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int? MaKiemKe { get; set; }

11 │

12 │ public int? MaHangHoa { get; set; }

13 │

14 │ public string? MaVach { get; set; }

15 │

16 │ public string? MaKt { get; set; }

17 │

18 │ public string? TenHang { get; set; }

19 │

20 │ public Guid? MaChiNhanh { get; set; }

21 │

22 │ public decimal? SoLuongKiemKe { get; set; }

23 │

24 │ public string? TenDvt { get; set; }

25 │

26 │ public string? TenLop { get; set; }

27 │

28 │ public DateTime? NgayKiemKe { get; set; }

29 │

30 │ public int? MaKho { get; set; }

31 │

32 │ public string? TenKho { get; set; }

33 │

34 │ public string? SoKiemKe { get; set; }

35 │

36 │ public int MaDvt { get; set; }

37 │

38 │ public string? DienGiai { get; set; }

39 │

40 │ public int? Thang { get; set; }

41 │

42 │ public int? Nam { get; set; }

43 │

44 │ public string? GhiChu { get; set; }

45 │

46 │ public string? NguoiKiemKe { get; set; }

47 │ }

────────────────────────────────────────────────────────────

**📄 VChiTietNhapHang.cs (CS, 69 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class VChiTietNhapHang

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int MaNhapHang { get; set; }

11 │

12 │ public int MaHangHoa { get; set; }

13 │

14 │ public decimal SoLuong { get; set; }

15 │

16 │ public decimal GiaVon { get; set; }

17 │

18 │ public DateTime? HanSd { get; set; }

19 │

20 │ public int? NguoiTao { get; set; }

21 │

22 │ public DateTime? NgayTao { get; set; }

23 │

24 │ public decimal? ThanhTien { get; set; }

25 │

26 │ public decimal? ThanhTienSauCk { get; set; }

27 │

28 │ public decimal? ThanhTienSauVat { get; set; }

29 │

30 │ public string? SoHd { get; set; }

31 │

32 │ public DateTime NgayNhapHang { get; set; }

33 │

34 │ public int NhaCc { get; set; }

35 │

36 │ public string MaVach { get; set; } = null!;

37 │

38 │ public int LopHang { get; set; }

39 │

40 │ public string TenHang { get; set; } = null!;

41 │

42 │ public int MaDvt { get; set; }

43 │

44 │ public string TenDvt { get; set; } = null!;

45 │

46 │ public DateTime? NgayHoaDon { get; set; }

47 │

48 │ public string? SoHoaDonNcc { get; set; }

49 │

50 │ public decimal? ChietKhauTm { get; set; }

51 │

52 │ public decimal? ChietKhauPt { get; set; }

53 │

54 │ public int? Vat { get; set; }

55 │

56 │ public string TenNhaCc { get; set; } = null!;

57 │

58 │ public int? MaKho { get; set; }

59 │

60 │ public string? TenKho { get; set; }

61 │

62 │ public string TenLop { get; set; } = null!;

63 │

64 │ public string? MaKt { get; set; }

65 │

66 │ public string? MaKtncc { get; set; }

67 │

68 │ public string? GhiChu { get; set; }

69 │ }

────────────────────────────────────────────────────────────

**📄 VChiTietNhapHang.cs (CS, 71 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class VChiTietNhapHang

7 │ {

8 │ public Guid? MaChiNhanh { get; set; }

9 │

10 │ public int Ma { get; set; }

11 │

12 │ public int? MaNhapHang { get; set; }

13 │

14 │ public int? MaHangHoa { get; set; }

15 │

16 │ public decimal? SoLuong { get; set; }

17 │

18 │ public decimal? GiaVon { get; set; }

19 │

20 │ public DateTime? HanSd { get; set; }

21 │

22 │ public Guid? UserCreated { get; set; }

23 │

24 │ public DateTime? CreatedDate { get; set; }

25 │

26 │ public decimal? ThanhTien { get; set; }

27 │

28 │ public decimal? ThanhTienSauCk { get; set; }

29 │

30 │ public decimal? ThanhTienSauVat { get; set; }

31 │

32 │ public string? SoHoaDon { get; set; }

33 │

34 │ public DateTime? NgayNhapHang { get; set; }

35 │

36 │ public int? MaNhaCc { get; set; }

37 │

38 │ public string? MaVach { get; set; }

39 │

40 │ public int? MaLopHang { get; set; }

41 │

42 │ public string? TenHangHoa { get; set; }

43 │

44 │ public int? MaDvt { get; set; }

45 │

46 │ public string? TenDvt { get; set; }

47 │

48 │ public DateTime? NgayHoaDon { get; set; }

49 │

50 │ public string? SoHoaDonNcc { get; set; }

51 │

52 │ public decimal? ChietKhauTm { get; set; }

53 │

54 │ public decimal? ChietKhauPt { get; set; }

55 │

56 │ public int? Vat { get; set; }

57 │

58 │ public string TenNhaCc { get; set; } = null!;

59 │

60 │ public int? MaKho { get; set; }

61 │

62 │ public string? TenKho { get; set; }

63 │

64 │ public string? TenLopHang { get; set; }

65 │

66 │ public string? MaKt { get; set; }

67 │

68 │ public string? MaKtncc { get; set; }

69 │

70 │ public string? GhiChu { get; set; }

71 │ }

────────────────────────────────────────────────────────────

**📄 VChiTietXuatKho.cs (CS, 47 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class VChiTietXuatKho

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int? MaPhieuXuat { get; set; }

11 │

12 │ public int? MaHangHoa { get; set; }

13 │

14 │ public string MaVach { get; set; } = null!;

15 │

16 │ public string TenHang { get; set; } = null!;

17 │

18 │ public decimal? SoLuong { get; set; }

19 │

20 │ public decimal? GiaVon { get; set; }

21 │

22 │ public int? MaDvt { get; set; }

23 │

24 │ public string TenDvt { get; set; } = null!;

25 │

26 │ public string TenLop { get; set; } = null!;

27 │

28 │ public DateTime? NgayXuat { get; set; }

29 │

30 │ public string? SoPhieu { get; set; }

31 │

32 │ public bool? KhoaSo { get; set; }

33 │

34 │ public string? GhiChu { get; set; }

35 │

36 │ public int? MaKho { get; set; }

37 │

38 │ public string? TenKho { get; set; }

39 │

40 │ public string? GhiChuChiTiet { get; set; }

41 │

42 │ public string? MaKt { get; set; }

43 │

44 │ public int? MaKhoChuyenDen { get; set; }

45 │

46 │ public string? KhoChuyenDen { get; set; }

47 │ }

────────────────────────────────────────────────────────────

**📄 VChiTietXuatKho.cs (CS, 49 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class VChiTietXuatKho

7 │ {

8 │ public Guid? MaChiNhanh { get; set; }

9 │

10 │ public int Ma { get; set; }

11 │

12 │ public int? MaPhieuXuat { get; set; }

13 │

14 │ public int? MaHangHoa { get; set; }

15 │

16 │ public string? MaVach { get; set; }

17 │

18 │ public string? TenHang { get; set; }

19 │

20 │ public decimal? SoLuong { get; set; }

21 │

22 │ public decimal? GiaVon { get; set; }

23 │

24 │ public int? MaDvt { get; set; }

25 │

26 │ public string? TenDvt { get; set; }

27 │

28 │ public string? TenLop { get; set; }

29 │

30 │ public DateTime? NgayXuat { get; set; }

31 │

32 │ public string? SoPhieu { get; set; }

33 │

34 │ public bool? KhoaSo { get; set; }

35 │

36 │ public string? GhiChu { get; set; }

37 │

38 │ public int? MaKho { get; set; }

39 │

40 │ public string? TenKho { get; set; }

41 │

42 │ public string? GhiChuChiTiet { get; set; }

43 │

44 │ public string? MaKt { get; set; }

45 │

46 │ public int? MaKhoChuyenDen { get; set; }

47 │

48 │ public string? KhoChuyenDen { get; set; }

49 │ }

────────────────────────────────────────────────────────────

**📄 VDoanhThu.cs (CS, 63 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.VCEntity;

5 │

6 │ public partial class VDoanhThu

7 │ {

8 │ public int MaChiNhanhBan { get; set; }

9 │

10 │ public string ChiNhanhBan { get; set; } = null!;

11 │

12 │ public int MaNganhBan { get; set; }

13 │

14 │ public string NganhBan { get; set; } = null!;

15 │

16 │ public int? MaChiNhanhSuDung { get; set; }

17 │

18 │ public string? ChiNhanhSuDung { get; set; }

19 │

20 │ public int? MaNganhSuDung { get; set; }

21 │

22 │ public string? NganhSuDung { get; set; }

23 │

24 │ public int MaVoucher { get; set; }

25 │

26 │ public string TenVoucher { get; set; } = null!;

27 │

28 │ public string LoaiVoucher { get; set; } = null!;

29 │

30 │ public int MaKh { get; set; }

31 │

32 │ public string TenKh { get; set; } = null!;

33 │

34 │ public string? DienThoai { get; set; }

35 │

36 │ public string? DiaChi { get; set; }

37 │

38 │ public string? Cmnd { get; set; }

39 │

40 │ public int GiaTriVoucher { get; set; }

41 │

42 │ public int GiaTriThanhToan { get; set; }

43 │

44 │ public int SoDu { get; set; }

45 │

46 │ public DateTime? NgayHieuLuc { get; set; }

47 │

48 │ public DateTime? NgayHetHan { get; set; }

49 │

50 │ public DateTime? NgayThanhToan { get; set; }

51 │

52 │ public int? MaNhanVienThanhToan { get; set; }

53 │

54 │ public string? NhanVienThanhToan { get; set; }

55 │

56 │ public string? MaThe { get; set; }

57 │

58 │ public int MaBh { get; set; }

59 │

60 │ public int? PhuThu { get; set; }

61 │

62 │ public string? GhiChuBanHang { get; set; }

63 │ }

────────────────────────────────────────────────────────────

**📄 VGiaBanTheoKho.cs (CS, 31 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class VGiaBanTheoKho

7 │ {

8 │ public string MaVach { get; set; } = null!;

9 │

10 │ public int? MaHangHoa { get; set; }

11 │

12 │ public string TenHang { get; set; } = null!;

13 │

14 │ public int? GiaBan { get; set; }

15 │

16 │ public int LopHang { get; set; }

17 │

18 │ public string? TenLop { get; set; }

19 │

20 │ public int? MaKho { get; set; }

21 │

22 │ public string? TenKho { get; set; }

23 │

24 │ public int Dvt { get; set; }

25 │

26 │ public string? TenDvt { get; set; }

27 │

28 │ public decimal? SltonCuoiKy { get; set; }

29 │

30 │ public bool? ChoPhepBanAm { get; set; }

31 │ }

────────────────────────────────────────────────────────────

**📄 VGiaBanTheoKho.cs (CS, 33 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class VGiaBanTheoKho

7 │ {

8 │ public string? MaVach { get; set; }

9 │

10 │ public int? MaHangHoa { get; set; }

11 │

12 │ public string? TenHang { get; set; }

13 │

14 │ public int? GiaBan { get; set; }

15 │

16 │ public int? MaLopHang { get; set; }

17 │

18 │ public string? TenLop { get; set; }

19 │

20 │ public int? MaKho { get; set; }

21 │

22 │ public string? TenKho { get; set; }

23 │

24 │ public int? MaDvt { get; set; }

25 │

26 │ public string? TenDvt { get; set; }

27 │

28 │ public decimal? SltonCuoiKy { get; set; }

29 │

30 │ public bool? ChoPhepBanAm { get; set; }

31 │

32 │ public Guid? MaChiNhanh { get; set; }

33 │ }

────────────────────────────────────────────────────────────

**📄 VHangHoa.cs (CS, 39 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class VHangHoa

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string? MaKt { get; set; }

11 │

12 │ public string? MaVach { get; set; }

13 │

14 │ public int? MaLopHang { get; set; }

15 │

16 │ public Guid? MaChiNhanh { get; set; }

17 │

18 │ public string? TenLopHang { get; set; }

19 │

20 │ public string? TenHangHoa { get; set; }

21 │

22 │ public int? MaDvt { get; set; }

23 │

24 │ public string? TenDvt { get; set; }

25 │

26 │ public decimal? GiaBan { get; set; }

27 │

28 │ public decimal? GiaVon { get; set; }

29 │

30 │ public string? HinhAnh { get; set; }

31 │

32 │ public string? GhiChu { get; set; }

33 │

34 │ public bool? HetHang { get; set; }

35 │

36 │ public int? ParentId { get; set; }

37 │

38 │ public bool? Status { get; set; }

39 │ }

────────────────────────────────────────────────────────────

**📄 VHangHoa.cs (CS, 33 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class VHangHoa

7 │ {

8 │ public int HangHoaId { get; set; }

9 │

10 │ public string? MaKt { get; set; }

11 │

12 │ public string MaVach { get; set; } = null!;

13 │

14 │ public int LopHang { get; set; }

15 │

16 │ public string? TenLop { get; set; }

17 │

18 │ public string TenHang { get; set; } = null!;

19 │

20 │ public int Dvt { get; set; }

21 │

22 │ public string? TenDvt { get; set; }

23 │

24 │ public int? GiaBan { get; set; }

25 │

26 │ public decimal? GiaVon { get; set; }

27 │

28 │ public string? HinhAnh { get; set; }

29 │

30 │ public string? GhiChu { get; set; }

31 │

32 │ public bool TrangThai { get; set; }

33 │ }

────────────────────────────────────────────────────────────

**📄 VNhaCc.cs (CS, 37 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class VNhaCc

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string? MaKt { get; set; }

11 │

12 │ public string TenNhaCc { get; set; } = null!;

13 │

14 │ public string? DiaChi { get; set; }

15 │

16 │ public string? DienThoai { get; set; }

17 │

18 │ public string? Email { get; set; }

19 │

20 │ public string? NguoiLienHe { get; set; }

21 │

22 │ public bool TrangThai { get; set; }

23 │

24 │ public DateTime NgayTao { get; set; }

25 │

26 │ public int NguoiTao { get; set; }

27 │

28 │ public DateTime? NgaySua { get; set; }

29 │

30 │ public int? NguoiSua { get; set; }

31 │

32 │ public bool? DaXoa { get; set; }

33 │

34 │ public int? NguoiXoa { get; set; }

35 │

36 │ public DateTime? NgayXoa { get; set; }

37 │ }

────────────────────────────────────────────────────────────

**📄 VNhapHang.cs (CS, 45 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class VNhapHang

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string? SoHoaDon { get; set; }

11 │

12 │ public DateTime? NgayNhapHang { get; set; }

13 │

14 │ public int? MaNhaCc { get; set; }

15 │

16 │ public decimal? TienChietKhau { get; set; }

17 │

18 │ public int? ThueVat { get; set; }

19 │

20 │ public string? TenLoaiNhapXuat { get; set; }

21 │

22 │ public string? GhiChu { get; set; }

23 │

24 │ public bool? KhoaSo { get; set; }

25 │

26 │ public string? TenNhaCc { get; set; }

27 │

28 │ public Guid? UserCreated { get; set; }

29 │

30 │ public int? MaKho { get; set; }

31 │

32 │ public string? TenKho { get; set; }

33 │

34 │ public DateTime? CreatedDate { get; set; }

35 │

36 │ public decimal? ThanhTien { get; set; }

37 │

38 │ public DateTime? NgayHoaDon { get; set; }

39 │

40 │ public bool? KhongHoaDon { get; set; }

41 │

42 │ public string? SoHoaDonNcc { get; set; }

43 │

44 │ public Guid? MaChiNhanh { get; set; }

45 │ }

────────────────────────────────────────────────────────────

**📄 VNhapHang.cs (CS, 43 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class VNhapHang

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string? SoHd { get; set; }

11 │

12 │ public DateTime NgayNhapHang { get; set; }

13 │

14 │ public int NhaCc { get; set; }

15 │

16 │ public decimal? TienChietKhau { get; set; }

17 │

18 │ public int? ThueVat { get; set; }

19 │

20 │ public string? TenLoaiNhapXuat { get; set; }

21 │

22 │ public string? GhiChu { get; set; }

23 │

24 │ public bool? KhoaSo { get; set; }

25 │

26 │ public string? TenNhaCc { get; set; }

27 │

28 │ public int? NguoiTao { get; set; }

29 │

30 │ public int? MaKho { get; set; }

31 │

32 │ public string? TenKho { get; set; }

33 │

34 │ public DateTime? NgayTao { get; set; }

35 │

36 │ public decimal? ThanhTien { get; set; }

37 │

38 │ public DateTime? NgayHoaDon { get; set; }

39 │

40 │ public bool? KhongHoaDon { get; set; }

41 │

42 │ public string? SoHoaDonNcc { get; set; }

43 │ }

────────────────────────────────────────────────────────────

**📄 VPermission.cs (CS, 25 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class VPermission

7 │ {

8 │ public Guid RoleId { get; set; }

9 │

10 │ public Guid MenuId { get; set; }

11 │

12 │ public Guid PermissionId { get; set; }

13 │

14 │ public Guid UserId { get; set; }

15 │

16 │ public string? Username { get; set; }

17 │

18 │ public bool EmailReceive { get; set; }

19 │

20 │ public string? FullName { get; set; }

21 │

22 │ public bool? AllowAccess { get; set; }

23 │

24 │ public Guid? BranchId { get; set; }

25 │ }

────────────────────────────────────────────────────────────

**📄 VQuyen.cs (CS, 19 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class VQuyen

7 │ {

8 │ public int MaRole { get; set; }

9 │

10 │ public int MaMenu { get; set; }

11 │

12 │ public int? MaQuyen { get; set; }

13 │

14 │ public int MaUser { get; set; }

15 │

16 │ public string TenDangNhap { get; set; } = null!;

17 │

18 │ public string HoTen { get; set; } = null!;

19 │ }

────────────────────────────────────────────────────────────

**📄 VQuyen.cs (CS, 27 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.VCEntity;

5 │

6 │ public partial class VQuyen

7 │ {

8 │ public int MaRole { get; set; }

9 │

10 │ public int MaMenu { get; set; }

11 │

12 │ public int MaQuyen { get; set; }

13 │

14 │ public int MaUser { get; set; }

15 │

16 │ public string TenDangNhap { get; set; } = null!;

17 │

18 │ public int MaChiNhanh { get; set; }

19 │

20 │ public int MaNganh { get; set; }

21 │

22 │ public string? Email { get; set; }

23 │

24 │ public bool TuChoiNhanEmail { get; set; }

25 │

26 │ public string HoTen { get; set; } = null!;

27 │ }

────────────────────────────────────────────────────────────

**📄 VTaoThanhToan.cs (CS, 43 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.VCEntity;

5 │

6 │ public partial class VTaoThanhToan

7 │ {

8 │ public int MaBanHang { get; set; }

9 │

10 │ public int MaKh { get; set; }

11 │

12 │ public int MaVoucher { get; set; }

13 │

14 │ public string MaVach { get; set; } = null!;

15 │

16 │ public int GiaTri { get; set; }

17 │

18 │ public int? SoDu { get; set; }

19 │

20 │ public DateTime? NgayHieuLuc { get; set; }

21 │

22 │ public DateTime? NgayHetHan { get; set; }

23 │

24 │ public int? NguoiKichHoat { get; set; }

25 │

26 │ public DateTime? NgayKichHoat { get; set; }

27 │

28 │ public bool? TrangThai { get; set; }

29 │

30 │ public DateTime? NgayTao { get; set; }

31 │

32 │ public string TenVoucher { get; set; } = null!;

33 │

34 │ public int MaChiNhanh { get; set; }

35 │

36 │ public int MaNganh { get; set; }

37 │

38 │ public string TenKh { get; set; } = null!;

39 │

40 │ public string? MaThe { get; set; }

41 │

42 │ public string LoaiThanhToan { get; set; } = null!;

43 │ }

────────────────────────────────────────────────────────────

**📄 VThanhToan.cs (CS, 41 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.VCEntity;

5 │

6 │ public partial class VThanhToan

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int MaChiNhanh { get; set; }

11 │

12 │ public int MaNganh { get; set; }

13 │

14 │ public int GiaTriThanhToan { get; set; }

15 │

16 │ public int? SoDu { get; set; }

17 │

18 │ public string? GhiChu { get; set; }

19 │

20 │ public int? NguoiTao { get; set; }

21 │

22 │ public DateTime? NgayTao { get; set; }

23 │

24 │ public string Code { get; set; } = null!;

25 │

26 │ public int MaBanHang { get; set; }

27 │

28 │ public int MaKh { get; set; }

29 │

30 │ public int MaVoucher { get; set; }

31 │

32 │ public string MaVach { get; set; } = null!;

33 │

34 │ public string TenVoucher { get; set; } = null!;

35 │

36 │ public string TenKh { get; set; } = null!;

37 │

38 │ public string? MaThe { get; set; }

39 │

40 │ public int? PhuThu { get; set; }

41 │ }

────────────────────────────────────────────────────────────

**📄 VTonKho.cs (CS, 43 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class VTonKho

7 │ {

8 │ public int? MaKho { get; set; }

9 │

10 │ public string? SoHd { get; set; }

11 │

12 │ public DateTime? NgayNhapHang { get; set; }

13 │

14 │ public int? MaHangHoa { get; set; }

15 │

16 │ public string? TenHangHoa { get; set; }

17 │

18 │ public decimal? GiaVon { get; set; }

19 │

20 │ public decimal? SltonDauKy { get; set; }

21 │

22 │ public decimal? SlnhapTrongKy { get; set; }

23 │

24 │ public decimal? SlbanTrongKy { get; set; }

25 │

26 │ public decimal? SltonCuoiKy { get; set; }

27 │

28 │ public string? NamThang { get; set; }

29 │

30 │ public decimal? TriGiaDauKy { get; set; }

31 │

32 │ public decimal? TriGiaNhap { get; set; }

33 │

34 │ public decimal? TriGiaBan { get; set; }

35 │

36 │ public decimal? TriGiaCuoiKy { get; set; }

37 │

38 │ public decimal? SlxuatKhac { get; set; }

39 │

40 │ public decimal? SlxuatTieuDung { get; set; }

41 │

42 │ public decimal? SlxuatTra { get; set; }

43 │ }

────────────────────────────────────────────────────────────

**📄 VTonKho.cs (CS, 43 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class VTonKho

7 │ {

8 │ public int? MaKho { get; set; }

9 │

10 │ public string? SoHoaDon { get; set; }

11 │

12 │ public DateTime? NgayNhapHang { get; set; }

13 │

14 │ public int? MaHangHoa { get; set; }

15 │

16 │ public string? TenHangHoa { get; set; }

17 │

18 │ public decimal? GiaVon { get; set; }

19 │

20 │ public decimal? SltonDauKy { get; set; }

21 │

22 │ public decimal? SlnhapTrongKy { get; set; }

23 │

24 │ public decimal? SlbanTrongKy { get; set; }

25 │

26 │ public decimal? SltonCuoiKy { get; set; }

27 │

28 │ public string? NamThang { get; set; }

29 │

30 │ public decimal? TriGiaDauKy { get; set; }

31 │

32 │ public decimal? TriGiaNhap { get; set; }

33 │

34 │ public decimal? TriGiaBan { get; set; }

35 │

36 │ public decimal? TriGiaCuoiKy { get; set; }

37 │

38 │ public decimal? SlxuatKhac { get; set; }

39 │

40 │ public decimal? SlxuatTieuDung { get; set; }

41 │

42 │ public decimal? SlxuatTra { get; set; }

43 │ }

────────────────────────────────────────────────────────────

**📄 VTonKhoBanHang.cs (CS, 43 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class VTonKhoBanHang

7 │ {

8 │ public int MaKho { get; set; }

9 │

10 │ public string? MaKt { get; set; }

11 │

12 │ public string? TenLop { get; set; }

13 │

14 │ public string? MaVach { get; set; }

15 │

16 │ public int MaHangHoa { get; set; }

17 │

18 │ public string? TenHangHoa { get; set; }

19 │

20 │ public decimal? SltonDauKy { get; set; }

21 │

22 │ public decimal? SlnhapTrongKy { get; set; }

23 │

24 │ public decimal? SlbanTrongKy { get; set; }

25 │

26 │ public decimal? SltonCuoiKy { get; set; }

27 │

28 │ public string NamThang { get; set; } = null!;

29 │

30 │ public decimal? SlxuatKhac { get; set; }

31 │

32 │ public decimal? SlxuatTieuDung { get; set; }

33 │

34 │ public string? LopHang { get; set; }

35 │

36 │ public string? TenDv { get; set; }

37 │

38 │ public decimal? SlxuatTra { get; set; }

39 │

40 │ public DateTime? LoGanNhat { get; set; }

41 │

42 │ public decimal? SoLuongGanNhat { get; set; }

43 │ }

────────────────────────────────────────────────────────────

**📄 VTonKhoBanHang.cs (CS, 43 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class VTonKhoBanHang

7 │ {

8 │ public int? MaKho { get; set; }

9 │

10 │ public string? MaKt { get; set; }

11 │

12 │ public string? TenLop { get; set; }

13 │

14 │ public string MaVach { get; set; } = null!;

15 │

16 │ public int? MaHangHoa { get; set; }

17 │

18 │ public string TenHangHoa { get; set; } = null!;

19 │

20 │ public decimal? SltonDauKy { get; set; }

21 │

22 │ public decimal? SlnhapTrongKy { get; set; }

23 │

24 │ public decimal? SlbanTrongKy { get; set; }

25 │

26 │ public decimal? SltonCuoiKy { get; set; }

27 │

28 │ public string? NamThang { get; set; }

29 │

30 │ public decimal? SlxuatKhac { get; set; }

31 │

32 │ public decimal? SlxuatTieuDung { get; set; }

33 │

34 │ public int LopHang { get; set; }

35 │

36 │ public string? TenDv { get; set; }

37 │

38 │ public decimal? SlxuatTra { get; set; }

39 │

40 │ public DateTime? LoGanNhat { get; set; }

41 │

42 │ public decimal? SoLuongGanNhat { get; set; }

43 │ }

────────────────────────────────────────────────────────────

**📄 VTonKhoTuNgayDenNgay.cs (CS, 39 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.HappyStoreEntity;

5 │

6 │ public partial class VTonKhoTuNgayDenNgay

7 │ {

8 │ public string? NamThang { get; set; }

9 │

10 │ public int? MaNhapHang { get; set; }

11 │

12 │ public string? SoHd { get; set; }

13 │

14 │ public int? MaHangHoa { get; set; }

15 │

16 │ public string? Mavach { get; set; }

17 │

18 │ public string? TenHang { get; set; }

19 │

20 │ public string? Dvt { get; set; }

21 │

22 │ public int? GiaVon { get; set; }

23 │

24 │ public int? SltonDau { get; set; }

25 │

26 │ public int? TriGiaTonDau { get; set; }

27 │

28 │ public int? Slnhap { get; set; }

29 │

30 │ public int? TriGiaNhap { get; set; }

31 │

32 │ public int? Slban { get; set; }

33 │

34 │ public int? TriGiaBan { get; set; }

35 │

36 │ public int? SltonCuoi { get; set; }

37 │

38 │ public int? TriGiaTonCuoi { get; set; }

39 │ }

────────────────────────────────────────────────────────────

**📄 VTonKhoVatTu.cs (CS, 43 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class VTonKhoVatTu

7 │ {

8 │ public string NamThang { get; set; } = null!;

9 │

10 │ public int MaKho { get; set; }

11 │

12 │ public int MaCtnhap { get; set; }

13 │

14 │ public int? MaHangHoa { get; set; }

15 │

16 │ public int MaVatTu { get; set; }

17 │

18 │ public string? Code { get; set; }

19 │

20 │ public string? TenVatTu { get; set; }

21 │

22 │ public int? MaDvt { get; set; }

23 │

24 │ public string? TenDv { get; set; }

25 │

26 │ public decimal? GiaNhap { get; set; }

27 │

28 │ public decimal? SoLuongTon { get; set; }

29 │

30 │ public string? SoLo { get; set; }

31 │

32 │ public DateTime? HanSuDung { get; set; }

33 │

34 │ public bool? TrangThai { get; set; }

35 │

36 │ public bool? TreoNhapKho { get; set; }

37 │

38 │ public decimal? VatTuGiaNhap { get; set; }

39 │

40 │ public string? TenHangHoa { get; set; }

41 │

42 │ public string? MaKt { get; set; }

43 │ }

────────────────────────────────────────────────────────────

**📄 VVoucher.cs (CS, 51 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.VCEntity;

5 │

6 │ public partial class VVoucher

7 │ {

8 │ public int MaChiNhanhBan { get; set; }

9 │

10 │ public string ChiNhanhBan { get; set; } = null!;

11 │

12 │ public int MaNganhBan { get; set; }

13 │

14 │ public string NganhBan { get; set; } = null!;

15 │

16 │ public int MaVoucher { get; set; }

17 │

18 │ public string TenVoucher { get; set; } = null!;

19 │

20 │ public int MaKh { get; set; }

21 │

22 │ public string TenKh { get; set; } = null!;

23 │

24 │ public string? DienThoai { get; set; }

25 │

26 │ public string? DiaChi { get; set; }

27 │

28 │ public string? Cmnd { get; set; }

29 │

30 │ public string? Email { get; set; }

31 │

32 │ public int GiaTriVoucher { get; set; }

33 │

34 │ public int? SoDu { get; set; }

35 │

36 │ public DateTime? NgayHieuLuc { get; set; }

37 │

38 │ public DateTime? NgayHetHan { get; set; }

39 │

40 │ public bool? TrangThai { get; set; }

41 │

42 │ public int? NguoiKichHoat { get; set; }

43 │

44 │ public DateTime? NgayKichHoat { get; set; }

45 │

46 │ public int? MaNhanVienBanHang { get; set; }

47 │

48 │ public string NhanVienBanHang { get; set; } = null!;

49 │

50 │ public string? MaThe { get; set; }

51 │ }

────────────────────────────────────────────────────────────

**📄 VietQr.cs (CS, 47 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class VietQr

7 │ {

8 │ public Guid Ma { get; set; }

9 │

10 │ public int? SoTien { get; set; }

11 │

12 │ public string? NoiDung { get; set; }

13 │

14 │ public string? SoHoaDon { get; set; }

15 │

16 │ public int? LoaiHoaDon { get; set; }

17 │

18 │ public string? Qrdata { get; set; }

19 │

20 │ public int? TrangThai { get; set; }

21 │

22 │ public DateTime? CreatedDate { get; set; }

23 │

24 │ public Guid? UserModified { get; set; }

25 │

26 │ public DateTime? ModifiedDate { get; set; }

27 │

28 │ public bool? Deleted { get; set; }

29 │

30 │ public Guid? UserDeleted { get; set; }

31 │

32 │ public DateTime? DeletedDate { get; set; }

33 │

34 │ public Guid? MaTiepNhan { get; set; }

35 │

36 │ public Guid? MaBenhNhan { get; set; }

37 │

38 │ public DateTime? NgayThu { get; set; }

39 │

40 │ public decimal? MienGiam { get; set; }

41 │

42 │ public int? LoaiPhieuThu { get; set; }

43 │

44 │ public Guid? MaHopDong { get; set; }

45 │

46 │ public Guid? MaPhieuDeNghiTamUng { get; set; }

47 │ }

────────────────────────────────────────────────────────────

**📄 VtthBaoCaoVatTuSuDung.cs (CS, 23 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class VtthBaoCaoVatTuSuDung

7 │ {

8 │ public Guid Ma { get; set; }

9 │

10 │ public int MaChiNhanhInt { get; set; }

11 │

12 │ public int Nam { get; set; }

13 │

14 │ public int Thang { get; set; }

15 │

16 │ public int? MaKho { get; set; }

17 │

18 │ public int? MaHangHoa { get; set; }

19 │

20 │ public decimal? SoLuongBan { get; set; }

21 │

22 │ public decimal? SoLuongNvl { get; set; }

23 │ }

────────────────────────────────────────────────────────────

**📄 VtthChiTietDeXuat.cs (CS, 51 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class VtthChiTietDeXuat

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int? MaDeXuat { get; set; }

11 │

12 │ public int? MaVatTu { get; set; }

13 │

14 │ public int? MaCtnhapKho { get; set; }

15 │

16 │ public int? MaDvt { get; set; }

17 │

18 │ public decimal? SoLuong { get; set; }

19 │

20 │ public decimal? SoLuongXuat { get; set; }

21 │

22 │ public decimal? SoLuongKsk { get; set; }

23 │

24 │ public decimal? SoLuongSd { get; set; }

25 │

26 │ public decimal? SoLuongTra { get; set; }

27 │

28 │ public DateTime? NgayTra { get; set; }

29 │

30 │ public bool? DaXuat { get; set; }

31 │

32 │ public int? MaPhongBanXuat { get; set; }

33 │

34 │ public int? MaPhongBanSuDung { get; set; }

35 │

36 │ public string? GhiChu { get; set; }

37 │

38 │ public Guid? UserCreated { get; set; }

39 │

40 │ public DateTime? CreatedDate { get; set; }

41 │

42 │ public Guid? UserModified { get; set; }

43 │

44 │ public DateTime? ModifiedDate { get; set; }

45 │

46 │ public bool? Deleted { get; set; }

47 │

48 │ public Guid? UserDeleted { get; set; }

49 │

50 │ public DateTime? DeletedDate { get; set; }

51 │ }

────────────────────────────────────────────────────────────

**📄 VtthChiTietDinhMuc.cs (CS, 43 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class VtthChiTietDinhMuc

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int MaChiNhanhInt { get; set; }

11 │

12 │ public int? MaDinhMuc { get; set; }

13 │

14 │ public int? MaHangHoa { get; set; }

15 │

16 │ public int? MaVatTu { get; set; }

17 │

18 │ public int? MaDvt { get; set; }

19 │

20 │ public decimal? SoLuong { get; set; }

21 │

22 │ public decimal? DonGiaTb { get; set; }

23 │

24 │ public decimal? ThanhTien { get; set; }

25 │

26 │ public string? GhiChu { get; set; }

27 │

28 │ public Guid? UserCreated { get; set; }

29 │

30 │ public DateTime? CreatedDate { get; set; }

31 │

32 │ public Guid? UserModified { get; set; }

33 │

34 │ public DateTime? ModifiedDate { get; set; }

35 │

36 │ public bool? Deleted { get; set; }

37 │

38 │ public Guid? UserDeleted { get; set; }

39 │

40 │ public DateTime? DeletedDate { get; set; }

41 │

42 │ public bool? KhongTruKho { get; set; }

43 │ }

────────────────────────────────────────────────────────────

**📄 VtthDeXuatVtksk.cs (CS, 49 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class VtthDeXuatVtksk

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public Guid? MaPhongBan { get; set; }

11 │

12 │ public string? SoPhieu { get; set; }

13 │

14 │ public int? MaKhoDeNghi { get; set; }

15 │

16 │ public int? MaKhoXuat { get; set; }

17 │

18 │ public DateTime? NgayDeNghi { get; set; }

19 │

20 │ public DateTime? NgayXuat { get; set; }

21 │

22 │ public Guid? NguoiXuat { get; set; }

23 │

24 │ public DateTime? NgayNhan { get; set; }

25 │

26 │ public Guid? NguoiNhan { get; set; }

27 │

28 │ public DateTime? NgayHoanTat { get; set; }

29 │

30 │ public Guid? NguoiHoanTat { get; set; }

31 │

32 │ public int? TrangThai { get; set; }

33 │

34 │ public string? GhiChu { get; set; }

35 │

36 │ public Guid? UserCreated { get; set; }

37 │

38 │ public DateTime? CreatedDate { get; set; }

39 │

40 │ public Guid? UserModified { get; set; }

41 │

42 │ public DateTime? ModifiedDate { get; set; }

43 │

44 │ public bool? Deleted { get; set; }

45 │

46 │ public Guid? UserDeleted { get; set; }

47 │

48 │ public DateTime? DeletedDate { get; set; }

49 │ }

────────────────────────────────────────────────────────────

**📄 VtthDinhMuc.cs (CS, 43 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class VtthDinhMuc

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int MaChiNhanhInt { get; set; }

11 │

12 │ public Guid MaChiNhanh { get; set; }

13 │

14 │ public string? Code { get; set; }

15 │

16 │ public string? Ten { get; set; }

17 │

18 │ public string? QuyTrinhKt { get; set; }

19 │

20 │ public string? TaiLieuThamKhao { get; set; }

21 │

22 │ public string? GhiChu { get; set; }

23 │

24 │ public bool? Status { get; set; }

25 │

26 │ public Guid? UserCreated { get; set; }

27 │

28 │ public DateTime? CreatedDate { get; set; }

29 │

30 │ public Guid? UserModified { get; set; }

31 │

32 │ public DateTime? ModifiedDate { get; set; }

33 │

34 │ public bool? Deleted { get; set; }

35 │

36 │ public Guid? UserDeleted { get; set; }

37 │

38 │ public DateTime? DeletedDate { get; set; }

39 │

40 │ public int? MaNhomDvkt { get; set; }

41 │

42 │ public int? MaLoaiDvkt { get; set; }

43 │ }

────────────────────────────────────────────────────────────

**📄 VtthKhay.cs (CS, 47 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class VtthKhay

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int MaChiNhanhInt { get; set; }

11 │

12 │ public Guid MaChiNhanh { get; set; }

13 │

14 │ public Guid? MaPhongBan { get; set; }

15 │

16 │ public int? MaKhoTieuHao { get; set; }

17 │

18 │ public int? MaKhoNguyenLieu { get; set; }

19 │

20 │ public int? MaVatTu { get; set; }

21 │

22 │ public decimal? DinhMucKhay { get; set; }

23 │

24 │ public bool? DaNhap { get; set; }

25 │

26 │ public bool? KhayTreo { get; set; }

27 │

28 │ public string? MoTa { get; set; }

29 │

30 │ public bool? Status { get; set; }

31 │

32 │ public Guid? UserCreated { get; set; }

33 │

34 │ public DateTime? CreatedDate { get; set; }

35 │

36 │ public Guid? UserModified { get; set; }

37 │

38 │ public DateTime? ModifiedDate { get; set; }

39 │

40 │ public bool? Deleted { get; set; }

41 │

42 │ public Guid? UserDeleted { get; set; }

43 │

44 │ public DateTime? DeletedDate { get; set; }

45 │

46 │ public int? MaNhapKhay { get; set; }

47 │ }

────────────────────────────────────────────────────────────

**📄 VtthLichSuXuatKhay.cs (CS, 49 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class VtthLichSuXuatKhay

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int MaChiNhanhInt { get; set; }

11 │

12 │ public Guid MaChiNhanh { get; set; }

13 │

14 │ public int? MaNhapKhay { get; set; }

15 │

16 │ public int? MaKhoNhap { get; set; }

17 │

18 │ public int? MaKhoXuat { get; set; }

19 │

20 │ public int? MaKhoXuatTra { get; set; }

21 │

22 │ public int? MaKhay { get; set; }

23 │

24 │ public int? MaVatTu { get; set; }

25 │

26 │ public DateTime? NgayXuat { get; set; }

27 │

28 │ public decimal? SoLuongXuat { get; set; }

29 │

30 │ public Guid? NguoiXuat { get; set; }

31 │

32 │ public string? GhiChu { get; set; }

33 │

34 │ public Guid? UserCreated { get; set; }

35 │

36 │ public DateTime? CreatedDate { get; set; }

37 │

38 │ public Guid? UserModified { get; set; }

39 │

40 │ public DateTime? ModifiedDate { get; set; }

41 │

42 │ public bool? Deleted { get; set; }

43 │

44 │ public Guid? UserDeleted { get; set; }

45 │

46 │ public DateTime? DeletedDate { get; set; }

47 │

48 │ public Guid? MaPhongBan { get; set; }

49 │ }

────────────────────────────────────────────────────────────

**📄 VtthNhapXuatKhay.cs (CS, 55 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyBanHang.Models.QLBHEntity;

5 │

6 │ public partial class VtthNhapXuatKhay

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int MaChiNhanhInt { get; set; }

11 │

12 │ public Guid? MaChiNhanh { get; set; }

13 │

14 │ public int? LoaiPhieu { get; set; }

15 │

16 │ public DateTime? NgayNhap { get; set; }

17 │

18 │ public DateTime? NgayXuat { get; set; }

19 │

20 │ public DateTime? NgayTra { get; set; }

21 │

22 │ public int? MaKhay { get; set; }

23 │

24 │ public decimal? SoLuongNhap { get; set; }

25 │

26 │ public decimal? SoLuongXuat { get; set; }

27 │

28 │ public decimal? SoLuongTra { get; set; }

29 │

30 │ public decimal? ConLai { get; set; }

31 │

32 │ public Guid? NguoiXuat { get; set; }

33 │

34 │ public Guid? NguoiTra { get; set; }

35 │

36 │ public bool? XacNhanTra { get; set; }

37 │

38 │ public DateTime? NgayXacNhanTra { get; set; }

39 │

40 │ public string? GhiChu { get; set; }

41 │

42 │ public Guid? UserCreated { get; set; }

43 │

44 │ public DateTime? CreatedDate { get; set; }

45 │

46 │ public Guid? UserModified { get; set; }

47 │

48 │ public DateTime? ModifiedDate { get; set; }

49 │

50 │ public bool? Deleted { get; set; }

51 │

52 │ public Guid? UserDeleted { get; set; }

53 │

54 │ public DateTime? DeletedDate { get; set; }

55 │ }

────────────────────────────────────────────────────────────

**📄 XuatKhoController.cs (CS, 498 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using QuanLyBanHang.Common;

3 │ using QuanLyBanHang.Common.Attributes;

4 │ using QuanLyBanHang.Models.APIModels;

5 │ using QuanLyBanHang.Models.ExModel.CAT;

6 │ using QuanLyBanHang.Models.ExModel.OPS;

7 │ using QuanLyBanHang.Models.ExModel.SYS;

8 │ using QuanLyBanHang.Models.QLBHEntity;

9 │ using QuanLyBanHang.Models.ViewModels;

10 │ using QuanLyBanHang.Services;

11 │ using System.Data;

12 │

13 │ namespace QuanLyBanHang.Controllers.OPS

14 │ {

15 │ public class XuatKho

16 │ {

17 │ public string Result { get; set; }

18 │ public string ResultText { get; set; }

19 │ }

20 │ public class ChiTietXuatKho

21 │ {

22 │ public string Ma { get; set; }

23 │ public string MaHangHoa { get; set; }

24 │ public string MaDVT { get; set; }

25 │ public string SoLuong { get; set; }

26 │ public string GiaVon { get; set; }

27 │ public string GhiChu { get; set; }

28 │ }

29 │ public class XuatKhoController : BaseController

30 │ {

31 │ private readonly IWebHostEnvironment \_hostingEnvironment;

32 │

33 │ public XuatKhoController(QuanLyBanHangContext dbcontext, MemoryCacheManager memoryCache, IWebHostEnvironment hos...

34 │ {

35 │ m\_MenuID = MenuID.opsXuatHang.GetEnumGuid();

36 │ \_hostingEnvironment = hostingEnvironment;

37 │ }

38 │

39 │ [TokenUserCheckHTTP]

40 │ public ActionResult Index(string TuNgay, string DenNgay, string MaKho)

41 │ {

42 │ DateTime dtpTuNgay = TuNgay.ToDateTime2()?.Date ?? DateTime.Now.Date;

43 │ DateTime dtpDenNgay = DenNgay.ToDateTime2()?.Date.AddDays(1).AddTicks(-1) ?? DateTime.Now.Date.AddDays(1).Ad...

44 │

45 │ ViewData["ListData"] = new XuatKhoModel(dbcontext).GetData(dtpTuNgay, dtpDenNgay, Cast.ToInt(MaKho), objToke...

46 │ ViewData["ListLoaiPhieuXuat"] = new LoaiNhapXuatModel(dbcontext).GetData();

47 │ ViewData["ListKho"] = new DanhMucKhoModel(dbcontext).GetDataByUser(objToken.userid, objToken.branch);

48 │ ViewData["TuNgay"] = dtpTuNgay;

49 │ ViewData["DenNgay"] = dtpDenNgay;

50 │ ViewData["MaKho"] = MaKho;

51 │ return View();

52 │ }

53 │ [TokenUserCheckHTTP(Permission.Add)]

54 │ public ActionResult Add()

55 │ {

56 │ ViewData["ListLoaiNhapXuat"] = new LoaiNhapXuatModel(dbcontext).DanhSachLoaiXuatKho();

57 │ ViewData["ListKho"] = new DanhMucKhoModel(dbcontext).GetDataByUser(objToken.userid, objToken.branch);

58 │ ViewData["ListHangHoa"] = new TblHangHoaModel(dbcontext).LayDanhSachTblHangHoa(objToken.branch).ToList();

59 │ return View("Form");

60 │ }

61 │

62 │ [TokenUserCheckHTTP(Permission.Edit)]

63 │ public ActionResult Edit(string Ma)

64 │ {

65 │ StockXuatKho xk = new XuatKhoModel(dbcontext).GetItem(Cast.ToInt(Ma.Decrypt()), objToken.branch);

66 │ if (xk == null)

67 │ {

68 │ return Json(QLBHConst.ResponseNotExist);

69 │ }

70 │ ViewData["itemXuatKho"] = xk;

71 │ ViewData["ListChiTietXuatKho"] = new ChiTietXuatKhoModel(dbcontext).GetListChiTietXuatKho(xk.Ma, objToken.br...

72 │ ViewData["ListLoaiNhapXuat"] = new LoaiNhapXuatModel(dbcontext).DanhSachLoaiXuatKho();

73 │ ViewData["ListKho"] = new DanhMucKhoModel(dbcontext).GetDataByUser(objToken.userid, objToken.branch);

74 │ ViewData["ListHangHoa"] = new TblHangHoaModel(dbcontext).LayDanhSachTblHangHoa(objToken.branch).ToList();

75 │ return View("Form");

76 │ }

77 │ [TokenUserCheckHTTP]

78 │ public ActionResult Detail(string Ma)

79 │ {

80 │ StockXuatKho xk = new XuatKhoModel(dbcontext).GetItem(Cast.ToInt(Ma.Decrypt()), objToken.branch);

81 │ if (xk == null)

82 │ {

83 │ return Json(QLBHConst.ResponseNotExist);

84 │ }

85 │ ViewData["itemXuatKho"] = xk;

86 │ ViewData["ListChiTietXuatKho"] = new ChiTietXuatKhoModel(dbcontext).GetListChiTietXuatKho(xk.Ma, objToken.br...

87 │ ViewData["ListLoaiNhapXuat"] = new LoaiNhapXuatModel(dbcontext).DanhSachLoaiXuatKho();

88 │ ViewData["ListKho"] = new DanhMucKhoModel(dbcontext).GetDataByUser(objToken.userid, objToken.branch);

89 │ ViewData["ListHangHoa"] = new TblHangHoaModel(dbcontext).LayDanhSachTblHangHoa(objToken.branch).ToList();

90 │ ViewData["ViewOnly"] = true;

91 │ return View("Form");

92 │ }

93 │ [TokenUserCheckHTTP]

94 │ public ActionResult Update(string Ma, string NgayXuat, string MaKho, string MaKhoChuyenDen, string MaLoaiNX, str...

95 │ {

96 │ int \_Ma = Cast.ToInt(Ma.Decrypt());

97 │

98 │ if (ListChiTiet == null)

99 │ {

100 │ ListChiTiet = new List<ChiTietXuatKho>();

101 │ }

102 │

103 │ if (Cast.ToInt(MaKho) == 0)

104 │ {

105 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng chọn kho xuất hàng"));

106 │ }

107 │ if (Cast.ToInt(MaLoaiNX) == (int)LoaiNhapXuat.PhieuXuatChuyenKho)

108 │ {

109 │ if (Cast.ToInt(MaKho) == Cast.ToInt(MaKhoChuyenDen))

110 │ {

111 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng chọn kho xuất hàng và kho nhập hà...

112 │ }

113 │ }

114 │ else

115 │ {

116 │ MaKhoChuyenDen = "";

117 │ }

118 │ if (Cast.ToInt(MaLoaiNX) == 0)

119 │ {

120 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng chọn loại xuất kho"));

121 │ }

122 │ DateTime dtNgayXuat;

123 │ if (NgayXuat == "")

124 │ {

125 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng nhập ngày xuất kho"));

126 │ }

127 │ else

128 │ {

129 │ dtNgayXuat = Cast.ToDateTime(NgayXuat, "dd/MM/yyyy");

130 │ }

131 │ if (Cast.ToInt(MaLoaiNX) == (int)LoaiNhapXuat.PhieuXuatChuyenKho)

132 │ {

133 │ if (Cast.ToInt(MaKhoChuyenDen) == 0)

134 │ {

135 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng chọn kho đến"));

136 │ }

137 │ }

138 │ var drKhoaSo = new KyKhoModel(dbcontext).Get(t => t.Nam == dtNgayXuat.Year

139 │ && t.Thang == dtNgayXuat.Month

140 │ && t.MaKho == Cast.ToInt(MaKho)

141 │ && (t.KhoaSo ?? false)).FirstOrDefault();

142 │

143 │ if (drKhoaSo != null)

144 │ {

145 │ return Json(new ResponseModel(QLBHConst.Error, "Kỳ kho đã khoá sổ"));

146 │ }

147 │

148 │ int ThangTruoc = dtNgayXuat.Month - 1;

149 │ int Nam = dtNgayXuat.Year;

150 │ if (ThangTruoc == 0)

151 │ {

152 │ ThangTruoc = 12;

153 │ Nam -= 1;

154 │ }

155 │ TblKyKho? drKhoaSoThangTruoc = new KyKhoModel(dbcontext).Get(t => t.Nam == Nam && t.Thang == ThangTruoc && t...

156 │

157 │ if (drKhoaSoThangTruoc != null)

158 │ {

159 │ if (!(drKhoaSoThangTruoc.KhoaSo ?? false))

160 │ {

161 │ return Json(new ResponseModel(QLBHConst.Error, "Vui lòng khóa sổ kỳ trước trước khi thao...

162 │ }

163 │ }

164 │

165 │ List<VPermission> ListPermission = GetVPermissions();

166 │

167 │ if (\_Ma == 0 && ListPermission.Any(t => t.PermissionId == Permission.Add.GetEnumGuid()))//Add

168 │ {

169 │ XuatKho res = new XuatKhoModel(dbcontext).Add(Cast.ToInt(MaKho), Cast.ToInt(MaKhoChuyenDen), dtNgayXuat,...

170 │ TonKhoBackgroundWorker.AddJob(new TonKhoJob

171 │ {

172 │ NamThang = string.Format("{0:0000}{1:00}", dtNgayXuat.Year, dtNgayXuat.Month),

173 │ MaKho = Cast.ToInt(MaKho),

174 │ UserId = objToken.userid,

175 │ MaChiNhanh = objToken.branch

176 │ });

177 │ if (Cast.ToInt(MaLoaiNX) == (int)LoaiNhapXuat.PhieuXuatChuyenKho)

178 │ {

179 │ TonKhoBackgroundWorker.AddJob(new TonKhoJob

180 │ {

181 │ NamThang = string.Format("{0:0000}{1:00}", dtNgayXuat.Year, dtNgayXuat.Month),

182 │ MaKho = Cast.ToInt(MaKhoChuyenDen),

183 │ UserId = objToken.userid,

184 │ MaChiNhanh = objToken.branch

185 │ });

186 │ }

187 │ if (dtNgayXuat.Year != DateTime.Now.Year || dtNgayXuat.Month != DateTime.Now.Month)

188 │ {

189 │ TonKhoBackgroundWorker.AddJob(new TonKhoJob

190 │ {

191 │ NamThang = string.Format("{0:0000}{1:00}", DateTime.Now.Year, DateTime.Now.Month),

192 │ MaKho = Cast.ToInt(MaKho),

193 │ UserId = objToken.userid,

194 │ MaChiNhanh = objToken.branch

195 │ });

196 │ if (Cast.ToInt(MaLoaiNX) == (int)LoaiNhapXuat.PhieuXuatChuyenKho)

197 │ {

198 │ TonKhoBackgroundWorker.AddJob(new TonKhoJob

199 │ {

200 │ NamThang = string.Format("{0:0000}{1:00}", DateTime.Now.Year, DateTime.Now.Month),

201 │ MaKho = Cast.ToInt(MaKhoChuyenDen),

202 │ UserId = objToken.userid,

203 │ MaChiNhanh = objToken.branch

204 │ });

205 │ }

206 │ }

207 │

208 │ return Json(new ResponseModel(res.Result, res.ResultText));

209 │ }

210 │ else if (\_Ma != 0 && ListPermission.Any(t => t.PermissionId == Permission.Edit.GetEnumGuid()))//Edit

211 │ {

212 │ XuatKho res = new XuatKhoModel(dbcontext).Edit(\_Ma, Cast.ToInt(MaKho), Cast.ToInt(MaKhoChuyenDen), dtNga...

213 │ TonKhoBackgroundWorker.AddJob(new TonKhoJob

214 │ {

215 │ NamThang = string.Format("{0:0000}{1:00}", dtNgayXuat.Year, dtNgayXuat.Month),

216 │ MaKho = Cast.ToInt(MaKho),

217 │ UserId = objToken.userid,

218 │ MaChiNhanh = objToken.branch

219 │ });

220 │ if (Cast.ToInt(MaLoaiNX) == (int)LoaiNhapXuat.PhieuXuatChuyenKho)

221 │ {

222 │ TonKhoBackgroundWorker.AddJob(new TonKhoJob

223 │ {

224 │ NamThang = string.Format("{0:0000}{1:00}", dtNgayXuat.Year, dtNgayXuat.Month),

225 │ MaKho = Cast.ToInt(MaKhoChuyenDen),

226 │ UserId = objToken.userid,

227 │ MaChiNhanh = objToken.branch

228 │ });

229 │ }

230 │ if (dtNgayXuat.Year != DateTime.Now.Year || dtNgayXuat.Month != DateTime.Now.Month)

231 │ {

232 │ TonKhoBackgroundWorker.AddJob(new TonKhoJob

233 │ {

234 │ NamThang = string.Format("{0:0000}{1:00}", DateTime.Now.Year, DateTime.Now.Month),

235 │ MaKho = Cast.ToInt(MaKho),

236 │ UserId = objToken.userid,

237 │ MaChiNhanh = objToken.branch

238 │ });

239 │ if (Cast.ToInt(MaLoaiNX) == (int)LoaiNhapXuat.PhieuXuatChuyenKho)

240 │ {

241 │ TonKhoBackgroundWorker.AddJob(new TonKhoJob

242 │ {

243 │ NamThang = string.Format("{0:0000}{1:00}", DateTime.Now.Year, DateTime.Now.Month),

244 │ MaKho = Cast.ToInt(MaKhoChuyenDen),

245 │ UserId = objToken.userid,

246 │ MaChiNhanh = objToken.branch

247 │ });

248 │ }

249 │ }

250 │ return Json(new ResponseModel(res.Result, res.ResultText));

251 │ }

252 │ else

253 │ {

254 │ return Json(QLBHConst.ResponseNoPermission);

255 │ }

256 │

257 │

258 │ }

259 │

260 │ [HttpPost]

261 │ [TokenUserCheckHTTP(Permission.Delete)]

262 │ public ActionResult Delete(string Ma)

263 │ {

264 │ Ma = Ma.Decrypt();

265 │ int res = new XuatKhoModel(dbcontext).Delete(Cast.ToInt(Ma), objToken.userid, objToken.branch);

266 │ if (res == -2)

267 │ {

268 │ return Json(QLBHConst.ResponseNotExist);

269 │ }

270 │ else if (res == -3)

271 │ {

272 │ return Json(new ResponseModel(QLBHConst.Error, "Đã khóa sổ không thể xóa"));

273 │ }

274 │ else if (res == 1)

275 │ {

276 │ return Json(QLBHConst.ResponseDeleteSuccessfully);

277 │ }

278 │ else

279 │ {

280 │ return Json(QLBHConst.ResponseHaveError);

281 │ }

282 │ }

283 │

284 │ [TokenUserCheckHTTP]

285 │ public ActionResult In(string Ma)

286 │ {

287 │ Ma = Ma.Decrypt();

288 │ StockXuatKho xk = new XuatKhoModel(dbcontext).GetItem(Cast.ToInt(Ma), objToken.branch);

289 │ List<ChiTietXuatKhoViewModel> ct = new ChiTietXuatKhoModel(dbcontext).GetListChiTietXuatKho(xk == null ? 0 :...

290 │ ViewData["item"] = xk ?? new StockXuatKho();

291 │ ViewData["kho"] = new DanhMucKhoModel(dbcontext).GetItem(xk.MaKho ?? 0);

292 │ ViewData["khoden"] = new DanhMucKhoModel(dbcontext).GetItem(xk.MaKhoChuyenDen ?? 0);

293 │ ViewData["nguoixuat"] = new sys\_UserModel(dbcontext).GetItem(xk.UserCreated);

294 │ ViewData["ListChiTiet"] = ct ?? new List<ChiTietXuatKhoViewModel>();

295 │ return View();

296 │ }

297 │

298 │ [HttpPost]

299 │ [TokenUserCheckHTTP(Permission.KhoaSo)]

300 │ public ActionResult KhoaSo(string MaXuatKho, string Value)

301 │ {

302 │ List<VPermission> ListPermission = GetVPermissions();

303 │ if (Value == "1")

304 │ {

305 │ if (ListPermission.Any(t => t.PermissionId == Permission.KhoaSo.GetEnumGuid()))

306 │ {

307 │ int res = new XuatKhoModel(dbcontext).KhoaSo(Cast.ToInt(MaXuatKho.Decrypt()), objToken.userid, true,...

308 │ if (res == 1)

309 │ {

310 │ return Json(new ResponseModel(QLBHConst.OK, "Khóa sổ thành công"));

311 │ }

312 │ else if (res == -2)

313 │ {

314 │ return Json(QLBHConst.ResponseNotExist);

315 │ }

316 │ else if (res == -3)

317 │ {

318 │ return Json(new ResponseModel(QLBHConst.Error, "Đã khóa sổ bởi người khác"));

319 │ }

320 │ else

321 │ {

322 │ return Json(QLBHConst.ResponseHaveError);

323 │ }

324 │ }

325 │ else

326 │ {

327 │ return Json(QLBHConst.ResponseNoPermission);

328 │ }

329 │ }

330 │ else

331 │ {

332 │ if (ListPermission.Any(t => t.PermissionId == Permission.MoKhoaSo.GetEnumGuid()))

333 │ {

334 │ int res = new XuatKhoModel(dbcontext).KhoaSo(Cast.ToInt(MaXuatKho.Decrypt()), objToken.userid, false...

335 │ if (res == 1)

336 │ {

337 │ return Json(new ResponseModel(QLBHConst.OK, "Mở khóa sổ thành công"));

338 │ }

339 │ else if (res == -2)

340 │ {

341 │ return Json(QLBHConst.ResponseNotExist);

342 │ }

343 │ else if (res == -3)

344 │ {

345 │ return Json(new ResponseModel(QLBHConst.Error, "Đã mở khóa sổ bởi người khác"));

346 │ }

347 │ else

348 │ {

349 │ return Json(QLBHConst.ResponseHaveError);

350 │ }

351 │ }

352 │ else

353 │ {

354 │ return Json(QLBHConst.ResponseNoPermission);

355 │ }

356 │ }

357 │ }

358 │

359 │ [HttpPost]

360 │ [TokenUserCheckHTTP]

361 │ public ActionResult getTonKhoTheoMaHangHoa(string MaHangHoa, string MaKho, string Nam, string Thang)

362 │ {

363 │ return Json(new ResponseModel(QLBHConst.OK, "", new { soluongton = new TonKhoModel(dbcontext).TonKhoTheoMaHa...

364 │ }

365 │

366 │ [TokenUserCheckHTTP]

367 │ public ActionResult ExportExcel(string TuNgay, string DenNgay, string MaKho, string FileType)

368 │ {

369 │ DateTime TuNgayConvert = TuNgay.ToDateTime2(DateTime.Now).Value.Date;

370 │ DateTime DenNgayConvert = DenNgay.ToDateTime2(DateTime.Now).Value.Date;

371 │

372 │ var listData = new ChiTietXuatKhoModel(dbcontext).GetListChiTietXuatKho(TuNgayConvert, DenNgayConvert, MaKho...

373 │

374 │ DataTable dtb = new();

375 │ dtb.Columns.Add("Ngày xuất hàng");

376 │ dtb.Columns.Add("Số CT");

377 │ dtb.Columns.Add("Kho xuất");

378 │ dtb.Columns.Add("Kho nhận");

379 │ dtb.Columns.Add("Lớp hàng");

380 │ dtb.Columns.Add("Mã hàng");

381 │ dtb.Columns.Add("Mã vạch");

382 │ dtb.Columns.Add("Tên hàng hóa");

383 │ dtb.Columns.Add("ĐVT");

384 │ dtb.Columns.Add("Số lượng", typeof(decimal));

385 │ dtb.Columns.Add("Đơn giá", typeof(decimal));

386 │ dtb.Columns.Add("Ghi chú");

387 │

388 │

389 │ foreach (var item in listData)

390 │ {

391 │ dtb.Rows.Add(new object[]

392 │ {

393 │ item.NgayXuat?.ToString("dd/MM/yyyy"),

394 │ item.SoPhieu,

395 │ item.TenKho,

396 │ item.KhoChuyenDen,

397 │ item.TenLop,

398 │ item.MaKT, // Mã hàng

399 │ item.MaVach,

400 │ item.TenHang,

401 │ item.TenDVT,

402 │ item.SoLuong,

403 │ item.GiaVon,

404 │ item.GhiChuChiTiet

405 │ });

406 │

407 │ }

408 │ string fileName = FileType == "Excel" ? "BaoCaoXuatHang.xlsx" : "BaoCaoXuatHang.pdf";

409 │ string contentType = FileType == "Excel" ? "application/ms-excel" : "application/pdf";

410 │

411 │ string fileTemplate = Path.Combine(\_hostingEnvironment.ContentRootPath, "wwwroot", "templates", "BaoCaoXuatH...

412 │ List<ExcelValue> ListGiaTriRiengLe = new()

413 │ {

414 │ new() { Range = $"A2:{ExportToExcel.GetExcelColumnName(dtb.Columns.Count)}2", Value = $"Từ ngày {TuNg...

415 │ };

416 │ int DongBatDau = 5;

417 │ Stream x = ExportToExcel.FromTemplate(fileTemplate, FileType, ListGiaTriRiengLe, dtb, true, DongBatDau, "A",...

418 │ return new FileStreamResult(x, contentType)

419 │ {

420 │ FileDownloadName = fileName

421 │ };

422 │ }

423 │

424 │ [TokenUserCheckHTTP]

425 │ public ActionResult ExportExcelPhieuXuatKho(string Ma)

426 │ {

427 │ int MaPhieuXuat = Ma.Decrypt().ToInt();

428 │ string FileType = "Excel";

429 │ StockXuatKho xk = new XuatKhoModel(dbcontext).GetItem(MaPhieuXuat, objToken.branch);

430 │ var listData = new ChiTietXuatKhoModel(dbcontext).GetListChiTietXuatKho(MaPhieuXuat, objToken.branch);

431 │

432 │ DataTable dtb = new();

433 │ dtb.Columns.Add("NgayXuat");

434 │ dtb.Columns.Add("SoPhieu");

435 │ dtb.Columns.Add("TenKho");

436 │ dtb.Columns.Add("KhoChuyenDen");

437 │ dtb.Columns.Add("TenLop");

438 │ dtb.Columns.Add("MaKT");

439 │ dtb.Columns.Add("MaVach");

440 │ dtb.Columns.Add("TenHang");

441 │ dtb.Columns.Add("TenDVT");

442 │ dtb.Columns.Add("SoLuong", typeof(decimal));

443 │ dtb.Columns.Add("GiaVon", typeof(decimal));

444 │ dtb.Columns.Add("GhiChu");

445 │

446 │ int IndexRow = 9, i = 0, IndexRowStart = 9;

447 │ int iRows = listData.Count + 30;

448 │ int iColumns = 12;

449 │

450 │ foreach (var item in listData)

451 │ {

452 │ dtb.Rows.Add(new object[]

453 │ {

454 │ i+1,

455 │ item.MaKT,

456 │ item.TenHang,

457 │ item.TenDVT,

458 │ item.SoLuong,

459 │ item.GiaVon,

460 │ item.SoLuong \* item.GiaVon,

461 │ item.GhiChu,

462 │ });

463 │ }

464 │ string fileName = FileType == "Excel" ? "PhieuXuatKho.xlsx" : "PhieuXuatKho.pdf";

465 │ string contentType = FileType == "Excel" ? "application/ms-excel" : "application/pdf";

466 │

467 │ StockDanhMucKho KhoNguon = new DanhMucKhoModel(dbcontext).GetItem(xk.MaKho ?? 0) ?? new StockDanhMucKho();

468 │ StockDanhMucKho KhoDich = new DanhMucKhoModel(dbcontext).GetItem(xk.MaKhoChuyenDen ?? 0) ?? new StockDanhMuc...

469 │ SysUser NguoiXuat = new sys\_UserModel(dbcontext).GetItem(xk.UserCreated) ?? new SysUser();

470 │

471 │ string fileTemplate = Path.Combine(\_hostingEnvironment.ContentRootPath, "wwwroot", "templates", "PhieuXuatKh...

472 │ int listCount = listData.Count;

473 │

474 │ List<ExcelValue> ListGiaTriRiengLe = new()

475 │ {

476 │ // rowTong[6]

477 │ new() { Range = "F1", Value = xk.SoPhieu.ToString(), Format = new ExcelFormat { IsBold = true } },

478 │ new() { Range = "A3", Value = $"'{xk.CreatedDate:dd/MM/yyyy HH:mm:ss}" },

479 │ new() { Range = "A4", Value = $"Ngày xuất kho: {xk.NgayXuat:dd/MM/yyyy HH:mm:ss}" },

480 │ new() { Range = "A5", Value = $"Xuất kho tại: {KhoNguon.Ten}" },

481 │ new() { Range = "A6", Value = $"Ghi chú: {xk.GhiChu}" },

482 │ new() { Range = "D4", Value = $"Người xuất: {NguoiXuat.FullName}" },

483 │ new() { Range = "D5", Value = (xk.MaLoaiNx ?? 0) == (int)LoaiNhapXuat.PhieuXuatChuyenKho ? "" : $"Đến...

484 │ new() { Range = $"A{listCount + 9}", Value = $"Phiếu nhập này gồm: {listData.Count} khoản." },

485 │ new() { Range = $"E{listCount + 9}", Value = $"{DateTime.Now:HH:mm} ngày {DateTime.Now:dd} tháng {Date...

486 │ new() { Range = $"B{listCount + 1 + 9}", Value = "Người lập biểu" },

487 │ new() { Range = $"F{listCount + 1 + 9}", Value = "Thủ kho" },

488 │

489 │ };

490 │ int DongBatDau = 9;

491 │ Stream x = ExportToExcel.FromTemplate(fileTemplate, FileType, ListGiaTriRiengLe, dtb, true, DongBatDau, "A",...

492 │ return new FileStreamResult(x, contentType)

493 │ {

494 │ FileDownloadName = fileName

495 │ };

496 │ }

497 │ }

498 │ }

────────────────────────────────────────────────────────────

**📄 XuatKhoModel.cs (CS, 233 lines)**

1 │ ﻿using Microsoft.EntityFrameworkCore;

2 │ using NetTopologySuite.Index.HPRtree;

3 │ using Newtonsoft.Json.Linq;

4 │ using QuanLyBanHang.Common;

5 │ using QuanLyBanHang.Controllers.OPS;

6 │ using QuanLyBanHang.Models.QLBHEntity;

7 │ using QuanLyBanHang.Services;

8 │

9 │ namespace QuanLyBanHang.Models.ExModel.OPS

10 │ {

11 │ public class XuatKhoModel : Base<StockXuatKho>

12 │ {

13 │ public XuatKhoModel(DbContext dbContext) : base(dbContext)

14 │ {

15 │ }

16 │ QuanLyBanHangContext dbContext

17 │ {

18 │ get { return \_dbContext as QuanLyBanHangContext; }

19 │ set { return; }

20 │ }

21 │ public StockXuatKho GetItem(int Ma, Guid MaChiNhanh)

22 │ {

23 │ return First(t => t.Ma == Ma && !(t.Deleted ?? false) && t.MaChiNhanh == MaChiNhanh);

24 │ }

25 │ public List<StockXuatKho> GetData(DateTime TuNgay, DateTime DenNgay, Guid MaChiNhanh)

26 │ {

27 │ return Get(t => t.NgayXuat >= TuNgay && t.NgayXuat <= DenNgay && !(t.Deleted ?? false) && t.MaChiNhanh == Ma...

28 │ }

29 │ public List<StockXuatKho> GetData(DateTime TuNgay, DateTime DenNgay, int MaKho, Guid MaChiNhanh)

30 │ {

31 │ return Get(t => t.NgayXuat >= TuNgay && t.NgayXuat <= DenNgay && (MaKho == 0 || t.MaKho == MaKho) && !(t.Del...

32 │ }

33 │ public XuatKho Add(int MaKho, int MaKhoChuyenDen, DateTime NgayXuat, int MaLoaiNX, string GhiChu, Guid UserID, L...

34 │ {

35 │ DateTime dtNgayTao = DateTime.Now;

36 │ try

37 │ {

38 │

39 │ using (var transaction = dbContext.Database.BeginTransaction())

40 │ {

41 │ int row = -1;

42 │

43 │ try

44 │ {

45 │ var br = dbContext.SysBranch.FirstOrDefault(x => x.Ma == MaChiNhanh);

46 │ if (br == null)

47 │ {

48 │ throw new Exception("Không tìm thấy chi nhánh");

49 │ }

50 │ StockXuatKho kk = new StockXuatKho

51 │ {

52 │ MaChiNhanh = MaChiNhanh,

53 │ MaChiNhanhInt = br.Nid ?? 0,

54 │ MaKho = MaKho,

55 │ MaKhoChuyenDen = MaKhoChuyenDen,

56 │ NgayXuat = NgayXuat,

57 │ MaLoaiNx = MaLoaiNX,

58 │ GhiChu = GhiChu,

59 │ UserCreated = UserID,

60 │ CreatedDate = dtNgayTao

61 │ };

62 │

63 │ dbContext.StockXuatKho.Add(kk);

64 │ dbContext.SaveChanges();

65 │ row = 0;

66 │ int rowError = new ChiTietXuatKhoModel(dbContext).SaveListChiTiet(kk.Ma, ListChiTiet, UserID, dt...

67 │ if (rowError != 0)

68 │ {

69 │ row = rowError;

70 │ if (rowError == -1)

71 │ {

72 │ throw new Exception(QLBHConst.NotExist);

73 │ }

74 │ throw new Exception("");

75 │ }

76 │

77 │ transaction.Commit();

78 │ }

79 │ catch (Exception ex)

80 │ {

81 │ transaction.Rollback();

82 │ if (row == -1)

83 │ return new XuatKho { Result = QLBHConst.Error, ResultText = "Thêm thất bại." + ex.Messa...

84 │ else if (row == 0)

85 │ return new XuatKho { Result = QLBHConst.Error, ResultText = "Thêm thất bại. Kiểm tra ...

86 │ else if (row <= -1000)

87 │ return new XuatKho { Result = QLBHConst.Error, ResultText = "Thêm thất bại. Số lượ...

88 │ else

89 │ return new XuatKho { Result = QLBHConst.Error, ResultText = "Thêm thất bại. Kiểm tra ...

90 │ }

91 │ }

92 │ return new XuatKho { Result = QLBHConst.OK, ResultText = QLBHConst.AddSuccessfully };

93 │ }

94 │ catch (Exception ex)

95 │ {

96 │ return new XuatKho { Result = QLBHConst.Error, ResultText = "Có lỗi xảy ra. " + ex.Message };

97 │ }

98 │

99 │ }

100 │ public XuatKho Edit(int Ma, int MaKho, int MaKhoChuyenDen, DateTime NgayXuat, int MaLoaiNX, string GhiChu, Guid ...

101 │ {

102 │ DateTime dtNgayTao = DateTime.Now;

103 │ try

104 │ {

105 │ using (var transaction = dbContext.Database.BeginTransaction())

106 │ {

107 │ int row = -1;

108 │ try

109 │ {

110 │ var br = dbContext.SysBranch.FirstOrDefault(x => x.Ma == MaChiNhanh);

111 │ if (br == null)

112 │ {

113 │ return new XuatKho { Result = QLBHConst.Error, ResultText = "Không tìm thấy chi nhánh" ...

114 │ }

115 │ StockXuatKho kk = dbContext.StockXuatKho.FirstOrDefault(t => t.Ma == Ma && !(t.Deleted ?? false)...

116 │ if (kk == null)

117 │ {

118 │ return new XuatKho { Result = QLBHConst.Error, ResultText = QLBHConst.NotExist };

119 │ }

120 │ if ((kk.KhoaSo ?? false))

121 │ {

122 │ return new XuatKho { Result = QLBHConst.Error, ResultText = "Đã khóa sổ không thể s�...

123 │ }

124 │ kk.MaChiNhanh = MaChiNhanh;

125 │ kk.MaChiNhanhInt = br.Nid ?? 0;

126 │ kk.MaKho = MaKho;

127 │ kk.NgayXuat = NgayXuat;

128 │ kk.MaLoaiNx = MaLoaiNX;

129 │ kk.GhiChu = GhiChu;

130 │ kk.UserModified = UserID;

131 │ kk.ModifiedDate = dtNgayTao;

132 │ kk.MaKhoChuyenDen = MaKhoChuyenDen;

133 │ dbContext.Update(kk);

134 │ dbContext.SaveChanges();

135 │

136 │ row = 0;

137 │ int rowError = new ChiTietXuatKhoModel(dbContext).SaveListChiTiet(kk.Ma, ListChiTiet, UserID, dt...

138 │ if (rowError != 0)

139 │ {

140 │ row = rowError;

141 │ if (rowError == -1)

142 │ {

143 │ throw new Exception(QLBHConst.NotExist);

144 │ }

145 │ throw new Exception("");

146 │ }

147 │

148 │ transaction.Commit();

149 │ }

150 │ catch (Exception ex)

151 │ {

152 │ transaction.Rollback();

153 │ if (row == -1)

154 │ return new XuatKho { Result = QLBHConst.Error, ResultText = "Cập nhật thất bại." + e...

155 │ else if (row == 0)

156 │ return new XuatKho { Result = QLBHConst.Error, ResultText = "Cập nhật thất bại. Ki��...

157 │ else if (row <= -1000)

158 │ return new XuatKho { Result = QLBHConst.Error, ResultText = "Cập nhật thất bại. Số...

159 │ else

160 │ return new XuatKho { Result = QLBHConst.Error, ResultText = "Cập nhật thất bại. Ki��...

161 │ }

162 │ }

163 │

164 │ TonKhoBackgroundWorker.AddJob(new TonKhoJob

165 │ {

166 │ NamThang = $"{NgayXuat:yyyyMM}",

167 │ MaKho = MaKho,

168 │ UserId = UserID,

169 │ MaChiNhanh = MaChiNhanh

170 │ });

171 │

172 │ return new XuatKho { Result = QLBHConst.OK, ResultText = QLBHConst.UpdateSuccessfully };

173 │ }

174 │ catch (Exception ex)

175 │ {

176 │ return new XuatKho { Result = QLBHConst.Error, ResultText = "Có lỗi xảy ra. " + ex.Message };

177 │ }

178 │

179 │ }

180 │ public int Delete(int Ma, Guid UserID, Guid MaChiNhanh)

181 │ {

182 │ StockXuatKho drItem = GetItem(Ma, MaChiNhanh);

183 │

184 │ if (drItem != null)

185 │ {

186 │ if ((drItem.KhoaSo ?? false))

187 │ {

188 │ return -3;

189 │ }

190 │ DateTime dtNgayTao = DateTime.Now;

191 │ drItem.Deleted = true;

192 │ drItem.DeletedDate = dtNgayTao;

193 │ drItem.UserDeleted = UserID;

194 │

195 │ dbContext.Update(drItem);

196 │ dbContext.SaveChanges();

197 │

198 │ TonKhoBackgroundWorker.AddJob(new TonKhoJob

199 │ {

200 │ NamThang = string.Format("{0:yyyyMM}", drItem.NgayXuat),

201 │ MaKho = Cast.ToInt(drItem.MaKho),

202 │ UserId = UserID,

203 │ MaChiNhanh = MaChiNhanh

204 │ });

205 │ return 1;

206 │

207 │ }

208 │ return -2;

209 │ }

210 │

211 │ public int KhoaSo(int Ma, Guid UserID, bool value, Guid MaChiNhanh)

212 │ {

213 │ StockXuatKho drItem = GetItem(Ma, MaChiNhanh);

214 │

215 │ if (drItem != null)

216 │ {

217 │ if ((drItem.KhoaSo ?? false) == value)

218 │ {

219 │ return -3;

220 │ }

221 │ DateTime dtNgayTao = DateTime.Now;

222 │ drItem.KhoaSo = value;

223 │ drItem.ModifiedDate = dtNgayTao;

224 │ drItem.UserModified = UserID;

225 │ dbContext.Update(drItem);

226 │ dbContext.SaveChanges();

227 │ return 1;

228 │

229 │ }

230 │ return -2;

231 │ }

232 │ }

233 │ }

────────────────────────────────────────────────────────────

**📄 sp\_BaoCaoDoanhThuTheoNgay.cs (CS, 37 lines)**

1 │ ﻿namespace QuanLyBanHang.Models.ExModel.StoredProcedureModel

2 │ {

3 │ public class sp\_BaoCaoDoanhThuTheoNgay

4 │ {

5 │ public string TuNgay { get; set; } = string.Empty;

6 │

7 │ public string DenNgay { get; set; } = string.Empty;

8 │

9 │ public DateTime NgayHoaDon { get; set; }

10 │

11 │ public int? LopHang { get; set; }

12 │

13 │ public string TenLop { get; set; } = string.Empty;

14 │

15 │ public string SoSeri { get; set; } = string.Empty;

16 │

17 │ public string MaVach { get; set; } = string.Empty;

18 │

19 │ public string TenHang { get; set; } = string.Empty;

20 │

21 │ public int? Dvt { get; set; }

22 │

23 │ public string TenDVT { get; set; } = string.Empty;

24 │

25 │ public decimal SoLuong { get; set; }

26 │

27 │ public decimal DonGia { get; set; }

28 │

29 │ public decimal? ThanhTien { get; set; }

30 │

31 │ public decimal? GiamGia { get; set; }

32 │

33 │ public decimal? TongCong { get; set; }

34 │

35 │ public int HinhThucTT { get; set; }

36 │ }

37 │ }

────────────────────────────────────────────────────────────

**📄 sp\_BaoCaoDoanhThuTheoNguoiBanTheoCa.cs (CS, 40 lines)**

1 │ ﻿namespace QuanLyBanHang.Models.ExModel.StoredProcedureModel

2 │ {

3 │ public class sp\_BaoCaoDoanhThuTheoNguoiBanTheoCa

4 │ {

5 │ public DateTime TuNgay { get; set; }

6 │ public DateTime DenNgay { get; set; }

7 │ public DateTime NgayHoaDon { get; set; }

8 │ public Guid? NguoiBan { get; set; }

9 │ public string Username { get; set; } = string.Empty;

10 │ public string Fullname { get; set; } = string.Empty;

11 │ public string TenHang { get; set; } = string.Empty;

12 │ public int? LopHang { get; set; }

13 │ public string TenLop { get; set; } = string.Empty;

14 │ public string SoSeri { get; set; } = string.Empty;

15 │ public string MaVach { get; set; } = string.Empty;

16 │ public string MaKT { get; set; } = string.Empty;

17 │ public int Dvt { get; set; }

18 │ public string TenDVT { get; set; } = string.Empty;

19 │ public string TenKho { get; set; } = string.Empty;

20 │ public decimal? SoLuong { get; set; }

21 │ public decimal? DonGia { get; set; }

22 │ public decimal? SoLuongTra { get; set; }

23 │ public decimal ThanhTien { get; set; }

24 │ public decimal GiamGia { get; set; }

25 │ public decimal ThanhTienTra { get; set; }

26 │ public decimal TienChuyenKhoan { get; set; }

27 │ public decimal TongCong { get; set; }

28 │ public decimal? GiaGiam { get; set; }

29 │ public int MaHinhThuc { get; set; }

30 │ public int? MaHangHoa { get; set; }

31 │ public int? MaKho { get; set; }

32 │ public decimal? SoLuongCK { get; set; }

33 │ public int HinhThucTt { get; set; }

34 │ public string TenHinhThuc { get; set; } = string.Empty;

35 │ public decimal KhuyenMaiTienMat { get; set; }

36 │ public string GhiChu { get; set; } = string.Empty;

37 │ public string RequestID { get; set; } = string.Empty;

38 │ public DateTime? NgayTao { get; set; }

39 │ }

40 │ }

────────────────────────────────────────────────────────────

**📄 sp\_BaoCaoKiemKe.cs (CS, 18 lines)**

1 │ ﻿using Microsoft.EntityFrameworkCore;

2 │

3 │ namespace QuanLyBanHang.Models.ExModel.StoredProcedureModel

4 │ {

5 │ [Keyless]

6 │ public class sp\_BaoCaoKiemKe

7 │ {

8 │ public int? Ma { get; set; }

9 │ public string? TenLop { get; set; }

10 │ public string? MaKT { get; set; }

11 │ public string? MaVach { get; set; }

12 │ public string? TenHang { get; set; }

13 │ public decimal? SoLuongTon { get; set; }

14 │ public decimal? SoLuongKiemKe { get; set; }

15 │ public decimal? SoLuongChenhLech { get; set; }

16 │ public string? GhiChu { get; set; }

17 │ }

18 │ }

────────────────────────────────────────────────────────────

**📄 sp\_BaoCaoTonKhoGiaVonResult.cs (CS, 20 lines)**

1 │ ﻿namespace QuanLyBanHang.Models.ExModel.StoredProcedureModel

2 │ {

3 │ public class sp\_BaoCaoTonKhoGiaVonResult

4 │ {

5 │ public string? TenLop { get; set; }

6 │ public string? MaVach { get; set; }

7 │ public string? MaKT { get; set; }

8 │ public string? TenHang { get; set; }

9 │ public string? TenDVT { get; set; }

10 │ public decimal? GiaVon { get; set; }

11 │ public decimal? SLTonDauKy { get; set; }

12 │ public decimal? SLNhapTra { get; set; }

13 │ public decimal? SLNhapTrongKy { get; set; }

14 │ public decimal? SLXuatKhac { get; set; }

15 │ public decimal? SLXuatTra { get; set; }

16 │ public decimal? SLXuatTieuDung { get; set; }

17 │ public decimal? SLBanTrongKy { get; set; }

18 │ public decimal? SLTonCuoiKy { get; set; }

19 │ }

20 │ }

────────────────────────────────────────────────────────────

**📄 sp\_BaoCaoTonKhoTheoNgayResult.cs (CS, 23 lines)**

1 │ ﻿namespace QuanLyBanHang.Models.ExModel.StoredProcedureModel

2 │ {

3 │ public class sp\_BaoCaoTonKhoTheoNgayResult

4 │ {

5 │ public string MaKT { get; set; }

6 │

7 │ public string TenHang { get; set; }

8 │

9 │ public string DVT { get; set; }

10 │

11 │ public decimal? GiaVon { get; set; }

12 │

13 │ public decimal? SoLuongDauKy { get; set; }

14 │

15 │ public decimal SoLuongNhapTrongKy { get; set; }

16 │

17 │ public decimal? SoLuongXuatKhac { get; set; }

18 │

19 │ public decimal SoLuongBanTrongKy { get; set; }

20 │

21 │ public decimal SoLuongTonCuoiKy { get; set; }

22 │ }

23 │ }

────────────────────────────────────────────────────────────

**📄 sp\_BaoCaoTongHopXuatBanResult.cs (CS, 16 lines)**

1 │ ﻿namespace QuanLyBanHang.Models.ExModel.StoredProcedureModel

2 │ {

3 │ public class sp\_BaoCaoTongHopXuatBanResult

4 │ {

5 │ public DateTime NgayXuatBan { get; set; }

6 │ public int MaKho { get; set; }

7 │ public string? TenKho { get; set; }

8 │ public int MaHangHoa { get; set; }

9 │ public string? MaKT { get; set; }

10 │ public string? MaVach { get; set; }

11 │ public string? TenHang { get; set; }

12 │ public string? TenDV { get; set; }

13 │ public decimal SoLuongXuat { get; set; }

14 │ public decimal SoLuongBan { get; set; }

15 │ }

16 │ }

────────────────────────────────────────────────────────────

**📄 sp\_ChiTietHoaDonResult.cs (CS, 15 lines)**

1 │ ﻿namespace QuanLyBanHang.Models.ExModel.StoredProcedureModel

2 │ {

3 │ public class sp\_ChiTietHoaDonResult

4 │ {

5 │ public string? TenHang;

6 │

7 │ public decimal? SoLuong;

8 │

9 │ public decimal? DonGia;

10 │

11 │ public decimal? GiaGiam;

12 │

13 │ public decimal? ThanhTien;

14 │ }

15 │ }

────────────────────────────────────────────────────────────

**📄 sp\_ChiTietHoaDonTraResult.cs (CS, 17 lines)**

1 │ ﻿namespace QuanLyBanHang.Models.ExModel.StoredProcedureModel

2 │ {

3 │ public class sp\_ChiTietHoaDonTraResult

4 │ {

5 │ public string? TenHang;

6 │

7 │ public decimal? SoLuongTraLai;

8 │

9 │ public decimal? DonGia;

10 │

11 │ public decimal? GiaGiam;

12 │

13 │ public decimal? ThanhTienTraLai;

14 │

15 │ public DateTime? NgayTra;

16 │ }

17 │ }

────────────────────────────────────────────────────────────

**📄 sp\_GetVoucherResult.cs (CS, 27 lines)**

1 │ ﻿namespace QuanLyBanHang.Models.ExModel.StoredProcedureModel

2 │ {

3 │ public class sp\_GetVoucherResult

4 │ {

5 │ public int? Ma;

6 │

7 │ public int? MaVoucher;

8 │

9 │ public string? MaVach;

10 │

11 │ public int? GiaTri;

12 │

13 │ public int? SoDu;

14 │

15 │ public int? ThanhTien;

16 │

17 │ public DateTime? NgayHieuLuc;

18 │

19 │ public DateTime? NgayHetHan;

20 │

21 │ public bool? TrangThai;

22 │

23 │ public int? MaChiNhanhApDung;

24 │

25 │ public int? MaNganhApDung;

26 │ }

27 │ }

────────────────────────────────────────────────────────────

**📄 sp\_HoaDonTheoNgayResult.cs (CS, 37 lines)**

1 │ ﻿namespace QuanLyBanHang.Models.ExModel.StoredProcedureModel

2 │ {

3 │ public class sp\_HoaDonTheoNgayResult

4 │ {

5 │ public int Ma;

6 │

7 │ public DateTime NgayHoaDon;

8 │

9 │ public string? SoSeri;

10 │

11 │ public string? NguoiBan;

12 │

13 │ public int HinhThucTT;

14 │

15 │ public string? TenHinhThuc;

16 │

17 │ public decimal Trigia;

18 │

19 │ public decimal? GiaGiam;

20 │

21 │ public Guid NguoiThanhToan;

22 │

23 │ public bool? KhoaSo;

24 │

25 │ public int? MaKho;

26 │

27 │ public string? TenKho;

28 │

29 │ public bool? TraHang;

30 │

31 │ public decimal? KhuyenMaiTienMat;

32 │

33 │ public string? RequestID;

34 │

35 │ public string? WebOrderId;

36 │ }

37 │ }

────────────────────────────────────────────────────────────

**📄 sp\_NhapHangImportFASTResult.cs (CS, 89 lines)**

1 │ ﻿namespace QuanLyBanHang.Models.ExModel.StoredProcedureModel

2 │ {

3 │ public class sp\_NhapHangImportFASTResult

4 │ {

5 │ public int Ma;

6 │

7 │ public string MaDVCS;

8 │

9 │ public string MaGD;

10 │

11 │ public DateTime? NgayLapChungTu;

12 │

13 │ public DateTime? NgayChungTu;

14 │

15 │ public string MaNT;

16 │

17 │ public int TyGia;

18 │

19 │ public string MaKT;

20 │

21 │ public string NguoiGiaoHang;

22 │

23 │ public string TaiKhoanCo;

24 │

25 │ public string MaThanhToan;

26 │

27 │ public string DienGiai;

28 │

29 │ public string TrangThai;

30 │

31 │ public string MaKT1;

32 │

33 │ public string DVT;

34 │

35 │ public string Code;

36 │

37 │ public string TenHang;

38 │

39 │ public string ViTri;

40 │

41 │ public string Lo;

42 │

43 │ public decimal SoLuong;

44 │

45 │ public decimal GiaVon;

46 │

47 │ public System.Nullable<decimal> ThanhTien;

48 │

49 │ public System.Nullable<decimal> TyLeCK;

50 │

51 │ public System.Nullable<decimal> ChietKhauNT;

52 │

53 │ public System.Nullable<decimal> ThanhTienSauCK;

54 │

55 │ public string TaiKhoanKho;

56 │

57 │ public int MauBC;

58 │

59 │ public int MaTC;

60 │

61 │ public string SoHD;

62 │

63 │ public string SoSeri;

64 │

65 │ public System.DateTime? NgayNhapHang;

66 │

67 │ public string MaNhaCC;

68 │

69 │ public string TenNhaCC;

70 │

71 │ public string DiaChi;

72 │

73 │ public string MST;

74 │

75 │ public string MaThue;

76 │

77 │ public System.Nullable<int> ThueSuat;

78 │

79 │ public string MaKho;

80 │

81 │ public System.Nullable<decimal> TienThue;

82 │

83 │ public string TKThue;

84 │

85 │ public string BoPhan;

86 │

87 │ public int NoiBo;

88 │ }

89 │ }

────────────────────────────────────────────────────────────

**📄 sp\_RoleMenuResult.cs (CS, 13 lines)**

1 │ ﻿namespace QuanLyBanHang.Models.StoredProcedureModel

2 │ {

3 │ public class sp\_RoleMenuResult

4 │ {

5 │ public Guid? Ma { get; set; }

6 │ public Guid? MenuID { get; set; }

7 │ public bool? AllowAccess { get; set; }

8 │ public bool? Status { get; set; }

9 │ public int? Sequence { get; set; }

10 │ public string? TypeMenu { get; set; } = null!;

11 │ public string? MenuName { get; set; } = null!;

12 │ }

13 │ }

────────────────────────────────────────────────────────────

**📄 sp\_RolePermissionResult.cs (CS, 11 lines)**

1 │ ﻿namespace QuanLyBanHang.Models.StoredProcedureModel

2 │ {

3 │ public class sp\_RolePermissionResult

4 │ {

5 │ public Guid? Ma { get; set; }

6 │ public Guid? MenuID { get; set; }

7 │ public Guid? PermissionID { get; set; }

8 │ public bool? Status { get; set; }

9 │ public string? PermissionName { get; set; }

10 │ }

11 │ }

────────────────────────────────────────────────────────────

**📄 sys\_BranchModel.cs (CS, 23 lines)**

1 │ ﻿using QuanLyBanHang.Models.QLBHEntity;

2 │ using Microsoft.EntityFrameworkCore;

3 │

4 │ namespace QuanLyBanHang.Models.ExModel.SYS

5 │ {

6 │ public class sys\_BranchModel : Base<SysBranch>

7 │ {

8 │ public sys\_BranchModel(DbContext dbContext) : base(dbContext)

9 │ {

10 │ }

11 │

12 │ public List<SysBranch> getListChiNhanhTheoUserMenuID(Guid UserID, Guid MenuID)

13 │ {

14 │ List<SysBranch> listChiNhanh = (from rm in \_dbContext.Set<SysRoleMenu>().Where(t => (t.Status ?? false) && !...

15 │ join ur in \_dbContext.Set<SysUserRole>().Where(t => (t.Status ?? false) && !...

16 │ join cn in \_dbContext.Set<SysBranch>().Where(t => (t.Status ?? false) && !(t...

17 │ where ur.UserId == UserID && rm.MenuId == MenuID

18 │ orderby cn.Ten

19 │ select cn).Distinct().OrderBy(x => x.Ten).ToList();

20 │ return listChiNhanh;

21 │ }

22 │ }

23 │ }

────────────────────────────────────────────────────────────

**📄 sys\_MenuModel.cs (CS, 127 lines)**

1 │ ﻿using QuanLyBanHang.Common;

2 │ using QuanLyBanHang.Models.QLBHEntity;

3 │ using Microsoft.EntityFrameworkCore;

4 │

5 │ namespace QuanLyBanHang.Models.ExModel.SYS

6 │ {

7 │ public class sys\_MenuModel : Base<SysMenu>

8 │ {

9 │ public sys\_MenuModel(DbContext dbContext) : base(dbContext)

10 │ {

11 │ }

12 │ public List<SysMenu> GetParentMenu()

13 │ {

14 │ return Get(t => t.ParentId == null && (t.Status ?? false) && !(t.Deleted ?? false))

15 │ .OrderBy(t => t.Sequence).ToList();

16 │ }

17 │ public SysMenu GetItem(Guid MenuID)

18 │ {

19 │ return First(t => t.Ma == MenuID && !(t.Deleted ?? false));

20 │ }

21 │ public List<SysMenu> GetData()

22 │ {

23 │ return Get(t => !(t.Deleted ?? false)).OrderBy(t => t.Sequence).ToList();

24 │ }

25 │

26 │ public List<SysMenu> GetMenu(Guid? UserID)

27 │ {

28 │ var menu = Get(m => \_dbContext.Set<VPermission>().Any(p => p.MenuId == m.Ma && (p.AllowAccess ?? false)

29 │ && p.UserId == UserID && p.PermissionId == Permission.View.GetEnumGuid())

30 │ && (m.Status ?? false) && !(m.Deleted ?? false)).OrderBy(t => t.Sequence).ToList();

31 │

32 │ return menu;

33 │ }

34 │ private bool CheckExists(Guid Ma, string Code)

35 │ {

36 │ return Any(t => t.Ma != Ma && t.Code == Code && !(t.Deleted ?? false));

37 │ }

38 │ public (int row, Guid? id) Add(string Code, Guid? ParentID, string Name, int Sequence, string Link, string Icon,...

39 │ , Guid? ChildOfMenu, string UsingFor)

40 │ {

41 │ if (CheckExists(Utility.defaultUID, Code))

42 │ return (-3, null);

43 │

44 │ SysMenu menuAdd = new SysMenu

45 │ {

46 │ Code = Code,

47 │ ParentId = ParentID,

48 │ Name = Name,

49 │ Sequence = Sequence,

50 │ Link = Link,

51 │ Icon = Icon,

52 │ ChildOfMenu = ChildOfMenu,

53 │ Status = Status,

54 │ UsingFor = UsingFor,

55 │ UserCreated = UserID,

56 │ CreatedDate = DateTime.Now,

57 │ };

58 │

59 │ \_dbContext.Set<SysMenu>().Add(menuAdd);

60 │ return (\_dbContext.SaveChanges(), menuAdd.Ma);

61 │ }

62 │ // Summary:

63 │ // Edit menu.

64 │ // Returns:

65 │ // Case when Result = 1 then successful

66 │ // when Result = 0 then failure

67 │ // when Result = -2 then not exist

68 │ // when Result = -3 then object is duplicate

69 │ public int Edit(Guid ID, string Code, Guid? ParentID, string Name, int Sequence, string Link, string Icon, bool ...

70 │ , Guid? UserID, Guid? ChildOfMenu, string UsingFor)

71 │ {

72 │ if (CheckExists(ID, Code))

73 │ return -3;

74 │

75 │ SysMenu menuUpdate = GetItem(ID);

76 │

77 │ if (menuUpdate != null)

78 │ {

79 │ menuUpdate.Code = Code;

80 │ menuUpdate.ParentId = ParentID;

81 │ menuUpdate.Name = Name;

82 │ menuUpdate.Sequence = Sequence;

83 │ menuUpdate.Link = Link;

84 │ menuUpdate.Icon = Icon;

85 │ menuUpdate.ChildOfMenu = ChildOfMenu;

86 │ menuUpdate.Status = Status;

87 │ menuUpdate.UsingFor = UsingFor;

88 │ menuUpdate.UserModified = UserID;

89 │ menuUpdate.ModifiedDate = DateTime.Now;

90 │ return Update(menuUpdate);

91 │ }

92 │

93 │ return -2;

94 │ }

95 │

96 │ // Summary:

97 │ // Delete menu.

98 │ // Returns:

99 │ // Case when Result = 1 then successful

100 │ // when Result = 0 then failure

101 │ // when Result = -1 then cannot delete

102 │ // when Result = -2 then not exist

103 │ public int Delete(Guid ID, Guid UserID)

104 │ {

105 │ SysMenu drM = GetItem(ID);

106 │

107 │ if (drM != null)

108 │ {

109 │ //Check children

110 │ SysMenu? drPM = First(t => t.ParentId == ID && !(t.Deleted ?? false));

111 │ SysRoleMenu? drRM = \_dbContext.Set<SysRoleMenu>().FirstOrDefault(t => t.MenuId == ID && !(t.Deleted ?? f...

112 │

113 │ if (drPM != null || drRM != null)

114 │ return -1;

115 │ else

116 │ {

117 │ drM.Deleted = true;

118 │ drM.DeletedDate = DateTime.Now;

119 │ drM.UserDeleted = UserID;

120 │ return Update(drM);

121 │ }

122 │ }

123 │

124 │ return -2;

125 │ }

126 │ }

127 │ }

────────────────────────────────────────────────────────────

**📄 sys\_PermissionModel.cs (CS, 93 lines)**

1 │ ﻿using QuanLyBanHang.Common;

2 │ using QuanLyBanHang.Models.QLBHEntity;

3 │ using Microsoft.EntityFrameworkCore;

4 │

5 │ namespace QuanLyBanHang.Models.ExModel.SYS

6 │ {

7 │ public class sys\_PermissionModel : Base<SysPermission>

8 │ {

9 │

10 │ public sys\_PermissionModel(DbContext dbContext) : base(dbContext)

11 │ {

12 │ }

13 │

14 │ public SysPermission GetItem(Guid Ma)

15 │ {

16 │ return First(t => t.Ma == Ma && !(t.Deleted ?? false));

17 │ }

18 │

19 │ public List<SysPermission> GetData()

20 │ {

21 │ return Get(t => !(t.Deleted ?? false)).OrderByDescending(t => t.Ma).ToList();

22 │ }

23 │ public List<SysPermission> GetDataPhanQuyenKho()

24 │ {

25 │ return Get(t => !(t.Deleted ?? false) && (t.PhanQuyenKho ?? false)).OrderBy(t => t.Ma).ToList();

26 │ }

27 │

28 │ public int Add(string Code, string Name, string Description, bool FormAccess, bool Status, Guid UserID)

29 │ {

30 │ if (CheckExists(Utility.defaultUID, Code))

31 │ return -3;

32 │

33 │ SysPermission roleAdd = new SysPermission

34 │ {

35 │ Code = Code,

36 │ Name = Name,

37 │ Description = Description,

38 │ FormAccess = FormAccess,

39 │ Status = Status,

40 │ UserCreated = UserID,

41 │ CreatedDate = DateTime.Now

42 │ };

43 │ return Add(roleAdd);

44 │ }

45 │ public int Edit(Guid ID, string Code, string Name, string Description, bool FormAccess, bool Status, Guid UserID...

46 │ {

47 │ if (CheckExists(ID, Code))

48 │ return -3;

49 │

50 │ SysPermission roleUpdate = GetItem(ID);

51 │

52 │ if (roleUpdate != null)

53 │ {

54 │ roleUpdate.Code = Code;

55 │ roleUpdate.Name = Name;

56 │ roleUpdate.Description = Description;

57 │ roleUpdate.FormAccess = FormAccess;

58 │ roleUpdate.Status = Status;

59 │ roleUpdate.UserModified = UserID;

60 │ roleUpdate.ModifiedDate = DateTime.Now;

61 │ return Update(roleUpdate);

62 │ }

63 │

64 │ return -2;

65 │ }

66 │ public int Delete(Guid ID, Guid UserID)

67 │ {

68 │ SysPermission drR = GetItem(ID);

69 │

70 │ if (drR != null)

71 │ {

72 │ //Check children

73 │ SysRolePermission? drRolePermission = \_dbContext.Set<SysRolePermission>().FirstOrDefault(t => t.Permissi...

74 │

75 │ if (drRolePermission != null)

76 │ return -1;

77 │ else

78 │ {

79 │ drR.Deleted = true;

80 │ drR.DeletedDate = DateTime.Now;

81 │ drR.UserDeleted = UserID;

82 │ return Update(drR);

83 │ }

84 │ }

85 │

86 │ return -2;

87 │ }

88 │ private bool CheckExists(Guid ID, string Code)

89 │ {

90 │ return Any(t => t.Ma != ID && t.Code == Code && !(t.Deleted ?? false));

91 │ }

92 │ }

93 │ }

────────────────────────────────────────────────────────────

**📄 sys\_RoleMenuModel.cs (CS, 95 lines)**

1 │ ﻿using Microsoft.EntityFrameworkCore;

2 │ using QuanLyBanHang.Common;

3 │ using QuanLyBanHang.Models.QLBHEntity;

4 │ using QuanLyBanHang.Models.StoredProcedureModel;

5 │ using System.Data;

6 │

7 │ namespace QuanLyBanHang.Models.ExModel.SYS

8 │ {

9 │ public class sys\_RoleMenuModel : Base<SysRoleMenu>

10 │ {

11 │ public sys\_RoleMenuModel(DbContext dbContext) : base(dbContext)

12 │ {

13 │ }

14 │

15 │ public List<sp\_RoleMenuResult> GetRoleMenuFollowRoleID(Guid RoleID)

16 │ {

17 │ return \_dbContext.Set<sp\_RoleMenuResult>().FromSqlRaw($"sp\_RoleMenu @iRole='{RoleID}'").ToList();

18 │ }

19 │ public SysRoleMenu GetItem(Guid RoleMenuID)

20 │ {

21 │ return First(t => t.Ma == RoleMenuID && !(t.Deleted ?? false));

22 │ }

23 │ public List<SysRoleMenu> GetDataFollowRoleID(Guid RoleID)

24 │ {

25 │ return Get(t => t.RoleId == RoleID && !(t.Deleted ?? false)).OrderByDescending(t => t.CreatedDate).ToList();

26 │ }

27 │ public SysRoleMenu GetItem(Guid RoleID, Guid MenuID)

28 │ {

29 │ return First(t => t.RoleId == RoleID && t.MenuId == MenuID && !(t.Deleted ?? false));

30 │ }

31 │ public int Add(Guid RoleID, Guid MenuID, bool AllowAccess, bool Status, Guid UserID)

32 │ {

33 │ if (CheckExists(Utility.defaultUID, RoleID, MenuID))

34 │ return -3;

35 │

36 │ SysRoleMenu roleMenuAdd = new SysRoleMenu

37 │ {

38 │ RoleId = RoleID,

39 │ MenuId = MenuID,

40 │ AllowAccess = AllowAccess,

41 │ Status = Status,

42 │ UserCreated = UserID,

43 │ CreatedDate = DateTime.Now,

44 │ };

45 │ return Add(roleMenuAdd);

46 │ }

47 │ public int Edit(Guid ID, Guid MenuID, bool AllowAccess, bool Status, Guid UserID)

48 │ {

49 │ SysRoleMenu roleMenuUpdate = GetItem(ID);

50 │

51 │ if (roleMenuUpdate != null)

52 │ {

53 │ if (CheckExists(ID, roleMenuUpdate.RoleId, MenuID))

54 │ return -3;

55 │

56 │ roleMenuUpdate.MenuId = MenuID;

57 │ roleMenuUpdate.AllowAccess = AllowAccess;

58 │ roleMenuUpdate.Status = Status;

59 │ roleMenuUpdate.UserModified = UserID;

60 │ roleMenuUpdate.ModifiedDate = DateTime.Now;

61 │

62 │ return Update(roleMenuUpdate);

63 │ }

64 │

65 │ return -2;

66 │ }

67 │ public int Delete(Guid ID, Guid UserID)

68 │ {

69 │ SysRoleMenu drRM = GetItem(ID);

70 │

71 │ if (drRM != null)

72 │ {

73 │ //Check children

74 │ SysRolePermission? drRP = \_dbContext.Set<SysRolePermission>().FirstOrDefault(t => t.RoleMenuId == ID && ...

75 │

76 │ if (drRP != null)

77 │ return -1;

78 │ else

79 │ {

80 │ drRM.Deleted = true;

81 │ drRM.DeletedDate = DateTime.Now;

82 │ drRM.UserDeleted = UserID;

83 │

84 │ return Update(drRM);

85 │ }

86 │ }

87 │

88 │ return -2;

89 │ }

90 │ private bool CheckExists(Guid ID, Guid RoleID, Guid MenuID)

91 │ {

92 │ return Any(t => t.Ma != ID && t.RoleId == RoleID && t.MenuId == MenuID && !(t.Deleted ?? false));

93 │ }

94 │ }

95 │ }

────────────────────────────────────────────────────────────

**📄 sys\_RoleModel.cs (CS, 121 lines)**

1 │ ﻿using QuanLyBanHang.Common;

2 │ using QuanLyBanHang.Models.QLBHEntity;

3 │ using Microsoft.EntityFrameworkCore;

4 │ using System.Net.WebSockets;

5 │

6 │ namespace QuanLyBanHang.Models.ExModel.SYS

7 │ {

8 │ public class sys\_RoleModel : Base<SysRole>

9 │ {

10 │ public class RoleChucVu

11 │ {

12 │ public Guid Ma { get; set; }

13 │ public string? Code { get; set; }

14 │ public string? Name { get; set; }

15 │ public string? TenChucVu { get; set; }

16 │ public Guid? MaChucVu { get; set; }

17 │ public string? Description { get; set; }

18 │ public bool? Status { get; set; }

19 │ public DateTime? CreatedDate { get; set; }

20 │ }

21 │ public sys\_RoleModel(DbContext dbContext) : base(dbContext)

22 │ {

23 │ }

24 │ QuanLyBanHangContext dbContext

25 │ {

26 │ get { return \_dbContext as QuanLyBanHangContext; }

27 │ set { return; }

28 │ }

29 │ public List<SysRole> GetData()

30 │ {

31 │ return Get(t => !(t.Deleted ?? false)).OrderByDescending(t => t.Ma).ToList();

32 │ }

33 │ public List<RoleChucVu> GetDataVoiChucVu()

34 │ {

35 │ var listData = (from role in dbContext.SysRole.AsNoTracking().Where(x => !(x.Deleted ?? false))

36 │ join cv in dbContext.TblChucVu on role.MaChucVu equals cv.Ma into cvGroup

37 │ from cv in cvGroup.DefaultIfEmpty()

38 │ select new RoleChucVu

39 │ {

40 │ Ma = role.Ma,

41 │ Code = role.Code,

42 │ Name = role.Name,

43 │ TenChucVu = cv.Ten ?? "",

44 │ MaChucVu = role.MaChucVu,

45 │ Description = role.Description,

46 │ Status = role.Status,

47 │ CreatedDate = role.CreatedDate

48 │ }).ToList();

49 │ return listData;

50 │ }

51 │ public SysRole GetItem(Guid RoleID)

52 │ {

53 │ return First(t => t.Ma == RoleID && !(t.Deleted ?? false));

54 │ }

55 │ public int Add(string Code, string Name, Guid MaChucVu, string Description, bool Status, Guid UserID)

56 │ {

57 │ if (CheckExists(Utility.defaultUID, Code))

58 │ return -3;

59 │

60 │ SysRole roleAdd = new SysRole

61 │ {

62 │ Code = Code,

63 │ Name = Name,

64 │ MaChucVu = MaChucVu,

65 │ Description = Description,

66 │ Status = Status,

67 │ UserCreated = UserID,

68 │ CreatedDate = DateTime.Now

69 │ };

70 │ return Add(roleAdd);

71 │ }

72 │ public int Edit(Guid ID, string Code, string Name, Guid MaChucVu, string Description, bool Status, Guid UserID)

73 │ {

74 │ if (CheckExists(ID, Code))

75 │ return -3;

76 │

77 │ SysRole roleUpdate = GetItem(ID);

78 │

79 │ if (roleUpdate != null)

80 │ {

81 │ roleUpdate.Code = Code;

82 │ roleUpdate.Name = Name;

83 │ roleUpdate.MaChucVu = MaChucVu;

84 │ roleUpdate.Description = Description;

85 │ roleUpdate.Status = Status;

86 │ roleUpdate.UserModified = UserID;

87 │ roleUpdate.ModifiedDate = DateTime.Now;

88 │ return Update(roleUpdate);

89 │ }

90 │

91 │ return -2;

92 │ }

93 │ public int Delete(Guid ID, Guid UserID)

94 │ {

95 │ SysRole drR = GetItem(ID);

96 │

97 │ if (drR != null)

98 │ {

99 │ //Check children

100 │ SysRoleMenu? drRM = \_dbContext.Set<SysRoleMenu>().FirstOrDefault(t => t.RoleId == ID && !(t.Deleted ?? f...

101 │ SysUserRole? drUR = \_dbContext.Set<SysUserRole>().FirstOrDefault(t => t.RoleId == ID && !(t.Deleted ?? f...

102 │

103 │ if (drUR != null || drRM != null)

104 │ return -1;

105 │ else

106 │ {

107 │ drR.Deleted = true;

108 │ drR.DeletedDate = DateTime.Now;

109 │ drR.UserDeleted = UserID;

110 │ return Update(drR);

111 │ }

112 │ }

113 │

114 │ return -2;

115 │ }

116 │ private bool CheckExists(Guid ID, string Code)

117 │ {

118 │ return Any(t => t.Ma != ID && t.Code == Code && !(t.Deleted ?? false));

119 │ }

120 │ }

121 │ }

────────────────────────────────────────────────────────────

**📄 sys\_RolePermissionModel.cs (CS, 243 lines)**

1 │ ﻿

2 │ //using QuanLyBanHang.Models.StoredProcedureModel;

3 │ using QuanLyBanHang.Common;

4 │ using QuanLyBanHang.Models.QLBHEntity;

5 │ using QuanLyBanHang.Models.StoredProcedureModel;

6 │ using Microsoft.EntityFrameworkCore;

7 │

8 │ namespace QuanLyBanHang.Models.ExModel.SYS

9 │ {

10 │ public class sys\_RolePermissionModel : Base<SysRolePermission>

11 │ {

12 │ private class RoleMenu

13 │ {

14 │ public Guid ID { get; set; }

15 │ public Guid MenuID { get; set; }

16 │ public bool AllowAccess { get; set; }

17 │ public bool Status { get; set; }

18 │ }

19 │

20 │ private class RolePermission

21 │ {

22 │ public Guid ID { get; set; }

23 │ public Guid MenuID { get; set; }

24 │ public Guid PermissionID { get; set; }

25 │ public bool Status { get; set; }

26 │ }

27 │

28 │ public sys\_RolePermissionModel(DbContext dbContext) : base(dbContext)

29 │ {

30 │ }

31 │

32 │ public SysRolePermission GetItem(Guid RolePermissionID)

33 │ {

34 │ return First(t => t.Ma == RolePermissionID && !(t.Deleted ?? false));

35 │ }

36 │ public List<SysRolePermission> GetDataFollowRoleMenuID(Guid RoleMenuID)

37 │ {

38 │ return Get(t => t.RoleMenuId == RoleMenuID && !(t.Deleted ?? false)).OrderByDescending(t => t.Ma).ToList();

39 │ }

40 │ public List<sp\_RolePermissionResult> GetRolePermissionFollowRoleID(Guid RoleID)

41 │ {

42 │ return \_dbContext.Set<sp\_RolePermissionResult>().FromSqlRaw($"exec sp\_RolePermission @iRole='{RoleID}'").ToL...

43 │ }

44 │

45 │ public int Delete(Guid ID, Guid UserID)

46 │ {

47 │ SysRolePermission drRP = GetItem(ID);

48 │ if (drRP != null)

49 │ {

50 │ drRP.Deleted = true;

51 │ drRP.DeletedDate = DateTime.Now;

52 │ drRP.UserDeleted = UserID;

53 │ return Update(drRP);

54 │ }

55 │ return -2;

56 │ }

57 │ public int Add(Guid RoleMenuID, Guid PermissionID, bool Status, Guid UserID)

58 │ {

59 │ if (CheckExists(Utility.defaultUID, RoleMenuID, PermissionID))

60 │ return -3;

61 │

62 │ SysRolePermission rolePermissionAdd = new SysRolePermission

63 │ {

64 │ RoleMenuId = RoleMenuID,

65 │ PermissionId = PermissionID,

66 │ Status = Status,

67 │ UserCreated = UserID,

68 │ CreatedDate = DateTime.Now

69 │ };

70 │

71 │ return Add(rolePermissionAdd);

72 │ }

73 │

74 │ public int Edit(Guid ID, Guid PermissionID, bool Status, Guid UserID)

75 │ {

76 │ SysRolePermission rolePermissionUpdate = GetItem(ID);

77 │ if (rolePermissionUpdate != null)

78 │ {

79 │ rolePermissionUpdate.PermissionId = PermissionID;

80 │ rolePermissionUpdate.Status = Status;

81 │ rolePermissionUpdate.UserModified = UserID;

82 │ rolePermissionUpdate.ModifiedDate = DateTime.Now;

83 │ return Update(rolePermissionUpdate);

84 │ }

85 │ return -2;

86 │ }

87 │ public bool UpdateQuicklyPermission(Guid RoleID, string RMAllowAccess, string RMStatus, string RPStatus, Guid Us...

88 │ {

89 │ //..

90 │ List<RoleMenu>? ListRoleMenu = new List<RoleMenu>();

91 │ List<RolePermission> ListRolePermisson = new List<RolePermission>();

92 │ string[]? ArrayItem = null;

93 │

94 │ //Xu ly RMAllowAccess

95 │ string[] Array = RMAllowAccess.Split(',');

96 │

97 │ if (Array != null && Array.Count() > 0)

98 │ {

99 │ for (int i = 0; i < Array.Count(); i++)

100 │ {

101 │ ArrayItem = Array[i].Split('.');

102 │

103 │ if (ArrayItem != null && ArrayItem.Count() == 2)

104 │ {

105 │ RoleMenu roleMenuAdd = new RoleMenu

106 │ {

107 │ ID = ArrayItem[0].GetGuid(),

108 │ MenuID = ArrayItem[1].GetGuid(),

109 │ AllowAccess = true,

110 │ Status = false

111 │ };

112 │ ListRoleMenu.Add(roleMenuAdd);

113 │ }

114 │ }

115 │ ListRoleMenu = ListRoleMenu.GroupBy(t => new { t.ID, t.MenuID }).Select(t => t.FirstOrDefault()).OrderBy...

116 │ }

117 │

118 │ //Xu ly RMStatus

119 │ Array = RMStatus.Split(',');

120 │ Guid itemRoleMenuID; Guid itemMenuID;

121 │ RoleMenu? itemRoleMenu = null;

122 │

123 │ if (Array != null && Array.Count() > 0)

124 │ {

125 │ for (int i = 0; i < Array.Count(); i++)

126 │ {

127 │ ArrayItem = Array[i].Split('.');

128 │

129 │ if (ArrayItem != null && ArrayItem.Count() == 2)

130 │ {

131 │ itemRoleMenuID = ArrayItem[0].GetGuid();

132 │ itemMenuID = ArrayItem[1].GetGuid();

133 │ itemRoleMenu = ListRoleMenu.FirstOrDefault(t => t.ID == itemRoleMenuID && t.MenuID == itemMenuID...

134 │

135 │ if (itemRoleMenu == null)

136 │ {

137 │ RoleMenu roleMenuAdd = new RoleMenu

138 │ {

139 │ ID = itemRoleMenuID,

140 │ MenuID = itemMenuID,

141 │ AllowAccess = false,

142 │ Status = true

143 │ };

144 │ ListRoleMenu.Add(roleMenuAdd);

145 │ }

146 │ else

147 │ itemRoleMenu.Status = true;

148 │ }

149 │ }

150 │ }

151 │

152 │ //Xu ly RPStatus

153 │ Array = RPStatus.Split(',');

154 │

155 │ if (Array != null && Array.Count() > 0)

156 │ {

157 │ for (int i = 0; i < Array.Count(); i++)

158 │ {

159 │ ArrayItem = Array[i].Split('.');

160 │

161 │ if (ArrayItem != null && ArrayItem.Count() == 3)

162 │ {

163 │ RolePermission rolePermissionAdd = new RolePermission

164 │ {

165 │ ID = ArrayItem[0].GetGuid(),

166 │ MenuID = ArrayItem[1].GetGuid(),

167 │ PermissionID = ArrayItem[2].GetGuid(),

168 │ Status = true

169 │ };

170 │ ListRolePermisson.Add(rolePermissionAdd);

171 │ }

172 │ }

173 │ }

174 │

175 │ //..

176 │ if (ListRoleMenu != null && ListRolePermisson != null)

177 │ {

178 │ //Update false role menu, permission exist

179 │ bool bUpdateRMP = UpdateRoleMenuPermission(RoleID);

180 │

181 │ if (bUpdateRMP)

182 │ {

183 │ int iCountUpdateRoleMenu = 0, iCountUpdateRolePermission = 0;

184 │

185 │ //Update Role Menu

186 │ sys\_RoleMenuModel sys\_RoleMenuM = new sys\_RoleMenuModel(\_dbContext);

187 │ foreach (RoleMenu item in ListRoleMenu)

188 │ {

189 │ if (item.ID == Utility.defaultUID)//Add

190 │ iCountUpdateRoleMenu += sys\_RoleMenuM.Add(RoleID, item.MenuID, item.AllowAccess, item.Status...

191 │ else//Edit

192 │ {

193 │ SysRoleMenu drRM = sys\_RoleMenuM.GetItem(item.ID);

194 │

195 │ if (drRM != null)

196 │ iCountUpdateRoleMenu += sys\_RoleMenuM.Edit(drRM.Ma, drRM.MenuId, item.AllowAccess, item....

197 │ }

198 │ }

199 │

200 │ //Update Role Permission

201 │ sys\_RolePermissionModel sys\_RolePermissionM = new sys\_RolePermissionModel(\_dbContext);

202 │ foreach (RolePermission item in ListRolePermisson)

203 │ {

204 │ if (item.ID == Utility.defaultUID)//Add

205 │ {

206 │ SysRoleMenu drRM = sys\_RoleMenuM.GetItem(RoleID, item.MenuID);

207 │

208 │ if (drRM != null)

209 │ iCountUpdateRolePermission += sys\_RolePermissionM.Add(drRM.Ma, item.PermissionID, item.S...

210 │ }

211 │ else//Edit

212 │ {

213 │ SysRolePermission drRP = sys\_RolePermissionM.GetItem(item.ID);

214 │

215 │ if (drRP != null)

216 │ iCountUpdateRolePermission += sys\_RolePermissionM.Edit(drRP.Ma, drRP.PermissionId, item....

217 │ }

218 │ }

219 │

220 │ //Verify

221 │ if (ListRoleMenu.Count == iCountUpdateRoleMenu && ListRolePermisson.Count == iCountUpdateRolePermiss...

222 │ return true;

223 │ return false;

224 │ }

225 │

226 │ return false;

227 │ }

228 │

229 │ return false;

230 │ }

231 │

232 │ private bool UpdateRoleMenuPermission(Guid RoleID)

233 │ {

234 │ if (\_dbContext.Database.ExecuteSqlRaw($"sp\_UpdateRoleMenuPermission @iRole='{RoleID}'") >= 0)

235 │ return true;

236 │ return false;

237 │ }

238 │ private bool CheckExists(Guid ID, Guid RoleMenuID, Guid PermissionID)

239 │ {

240 │ return Any(t => t.Ma != ID && t.RoleMenuId == RoleMenuID && t.PermissionId== PermissionID && !(t.Deleted ?? ...

241 │ }

242 │ }

243 │ }

────────────────────────────────────────────────────────────

**📄 sys\_UserModel.cs (CS, 233 lines)**

1 │ ﻿using QuanLyBanHang.Common;

2 │ using QuanLyBanHang.Models.QLBHEntity;

3 │ using Microsoft.EntityFrameworkCore;

4 │ using System.Net.WebSockets;

5 │

6 │ namespace QuanLyBanHang.Models.ExModel.SYS

7 │ {

8 │ public class sys\_UserModel : Base<SysUser>

9 │ {

10 │ public sys\_UserModel(DbContext dbContext) : base(dbContext)

11 │ {

12 │ }

13 │ QuanLyBanHangContext dbContext

14 │ {

15 │ get { return \_dbContext as QuanLyBanHangContext; }

16 │ set { return; }

17 │ }

18 │ public SysUser GetItem(Guid? UserID)

19 │ {

20 │ return First(t => t.Ma == UserID && !(t.Deleted ?? false));

21 │ }

22 │ public (int, SysUser?) Authenticate(string Username, string Password, Guid BranchID)

23 │ {

24 │ SysUser OutUser = First(t => t.Username == Username && !(t.Deleted ?? false));

25 │ bool bUsingLdap = false, bLdapLoginIsSuccess = false;

26 │ int iLdapLoginStatus = -2;

27 │ if (OutUser != null)

28 │ {

29 │ if (!(OutUser.Status ?? false))

30 │ {

31 │ return (0, null);

32 │ }

33 │

34 │ bUsingLdap = OutUser.LdapLogin ?? false;

35 │ if (bUsingLdap)

36 │ {

37 │ iLdapLoginStatus = 0;//LDAP.IsAuthenticated(Username, Password);

38 │ bLdapLoginIsSuccess = iLdapLoginStatus == 1;

39 │ if (bLdapLoginIsSuccess)

40 │ {

41 │ return (iLdapLoginStatus, OutUser);

42 │ }

43 │ }

44 │

45 │ OutUser = First(u => \_dbContext.Set<SysUserRole>().Any(ur => ur.UserId == u.Ma

46 │ && (ur.Status ?? false) && !(ur.Deleted ?? false) && ur.BranchId == BranchID)

47 │ && (u.Status ?? false) && !(u.Deleted ?? false) && u.Username == Username

48 │ && (bUsingLdap ? bLdapLoginIsSuccess : u.Password == Utility.GetSHA512(Password)));

49 │ return (OutUser != null ? 1 : -1, OutUser);

50 │ }

51 │ return (-1, null);

52 │ }

53 │ public List<SysUser> GetData()

54 │ {

55 │ return Get(t => !(t.Deleted ?? false)).OrderByDescending(t => t.Ma).ToList();

56 │ }

57 │ private bool CheckExists(Guid ID, string Username)

58 │ {

59 │ return Any(t => t.Ma != ID && t.Username == Username && !(t.Deleted ?? false));

60 │ }

61 │ public int Edit(Guid ID, string Username, string FullName, int GioiTinh, DateTime NgaySinh, string SoDienThoai, ...

62 │ string MSNV, Guid? MaPhongBan, Guid? MaChucDanh, Guid? MaChucVu, string CCCD, string MaDinhDanh, DateTime Ng...

63 │ string Photo, bool Status, Guid? UserID, Guid? BranchID, string KhoHang, Guid MaChiNhanhDangNhap)

64 │ {

65 │ if (CheckExists(ID, Username))

66 │ {

67 │ return -3;

68 │ }

69 │

70 │ SysUser userUpdate = GetItem(ID);

71 │

72 │ // 1. Parse danh sách kho cũ từ DB (chuỗi) thành List<int>

73 │ var oldKhoHang = (userUpdate.KhoHang + "")

74 │ .Split(';', StringSplitOptions.RemoveEmptyEntries)

75 │ .Select(x => int.Parse(x.Trim()))

76 │ .Distinct()

77 │ .ToList();

78 │

79 │ // 2. Lấy danh sách mã kho thuộc chi nhánh hiện tại

80 │ var currentChiNhanhMaKhoList = dbContext.StockDanhMucKho

81 │ .Where(x => x.MaChiNhanh == MaChiNhanhDangNhap)

82 │ .Select(k => k.Ma)

83 │ .ToList();

84 │

85 │ // 3. Loại bỏ các kho thuộc chi nhánh hiện tại khỏi danh sách kho cũ

86 │ oldKhoHang = oldKhoHang

87 │ .Where(ma => !currentChiNhanhMaKhoList.Contains(ma))

88 │ .ToList();

89 │

90 │ // 4. Parse danh sách mã kho mới được chọn từ form (chuỗi "5;6;7") thành List<int>

91 │ var selectedKhoHang = (KhoHang + "")

92 │ .Split(';', StringSplitOptions.RemoveEmptyEntries)

93 │ .Select(x => int.Parse(x.Trim()))

94 │ .Distinct()

95 │ .ToList();

96 │

97 │ // 5. Thêm các kho mới nếu chưa có (chỉ thuộc chi nhánh hiện tại)

98 │ oldKhoHang.AddRange(selectedKhoHang

99 │ .Where(ma => currentChiNhanhMaKhoList.Contains(ma) && !oldKhoHang.Contains(ma)));

100 │

101 │ if (userUpdate != null)

102 │ {

103 │ userUpdate.Username = Username;

104 │ userUpdate.FullName = FullName;

105 │ userUpdate.GioiTinh = GioiTinh;

106 │ userUpdate.NgaySinh = NgaySinh;

107 │ userUpdate.MaPhongBan = MaPhongBan;

108 │ userUpdate.SoDienThoai = SoDienThoai;

109 │ userUpdate.DiaChi = DiaChi;

110 │ userUpdate.Msnv = MSNV;

111 │ userUpdate.Cccd = CCCD;

112 │ userUpdate.Msnv = MSNV;

113 │ userUpdate.NgayVaoLam = NgayVaoLam;

114 │ userUpdate.NgayKetThuc = NgayKetThuc;

115 │ userUpdate.HinhAnh = Photo != "" ? Photo : userUpdate.HinhAnh;

116 │ userUpdate.Status = Status;

117 │ userUpdate.UserModified = UserID;

118 │ userUpdate.ModifiedDate = DateTime.Now;

119 │ userUpdate.BranchId = BranchID;

120 │ userUpdate.MaChucVu = MaChucVu;

121 │ userUpdate.KhoHang = Utility.SetItemMultiSelect(string.Join(";", oldKhoHang));

122 │ try

123 │ {

124 │ return Update(userUpdate);

125 │ }

126 │ catch (Exception ex)

127 │ {

128 │ string ss = ex.Message;

129 │ return 0;

130 │ }

131 │

132 │ }

133 │ return -2;

134 │ }

135 │

136 │ public (int, string) UpdateAvatar(Guid ID, string HinhAnh)

137 │ {

138 │ SysUser userUpdate = GetItem(ID);

139 │

140 │ if (userUpdate != null)

141 │ {

142 │ userUpdate.HinhAnh = HinhAnh != "" ? HinhAnh : userUpdate.HinhAnh;

143 │ return (Update(userUpdate), userUpdate.HinhAnh??"");

144 │ }

145 │ return (-2, "");

146 │ }

147 │

148 │ public int Add(string Username, string FullName, string Password, int GioiTinh, DateTime NgaySinh, string SoDien...

149 │ string MSNV, Guid? MaPhongBan, Guid? MaChucDanh, Guid? MaChucVu, string CCCD, string MaDinhDanh, DateTime Ng...

150 │ string Photo, bool Status, Guid? UserID, Guid? BranchID, string KhoHang)

151 │ {

152 │ if (CheckExists(Utility.defaultUID, Username))

153 │ {

154 │ return -3;

155 │ }

156 │

157 │ SysUser userAdd = new SysUser

158 │ {

159 │ Username = Username,

160 │ Password = Utility.GetSHA512(Password),

161 │ FullName = FullName,

162 │ GioiTinh = GioiTinh,

163 │ NgaySinh = NgaySinh,

164 │ SoDienThoai = SoDienThoai,

165 │ DiaChi = DiaChi,

166 │ Msnv = MSNV,

167 │ MaPhongBan = MaPhongBan,

168 │ MaChucVu = MaChucVu,

169 │ Cccd = CCCD,

170 │ NgayVaoLam = NgayVaoLam,

171 │ NgayKetThuc = NgayKetThuc,

172 │ Status = Status,

173 │ UserCreated = UserID,

174 │ CreatedDate = DateTime.Now,

175 │ HinhAnh = Photo ?? "/images/avatar.png",

176 │ BranchId = BranchID,

177 │ KhoHang = KhoHang,

178 │ };

179 │ return Add(userAdd);

180 │ }

181 │

182 │ public int Delete(Guid ID, Guid UserID)

183 │ {

184 │ SysUser drU = GetItem(ID);

185 │

186 │ if (drU != null)

187 │ {

188 │ //Check children

189 │ SysUserRole? drUR = \_dbContext.Set<SysUserRole>().FirstOrDefault(t => t.UserId == ID && !(t.Deleted ?? f...

190 │

191 │ if (drUR != null)

192 │ {

193 │ return -1;

194 │ }

195 │ else

196 │ {

197 │ drU.Deleted = true;

198 │ drU.DeletedDate = DateTime.Now;

199 │ drU.UserDeleted = UserID;

200 │ return Update(drU);

201 │ }

202 │ }

203 │

204 │ return -2;

205 │ }

206 │

207 │ public int ResetPassword(Guid ID, string Password)

208 │ {

209 │ SysUser drU = GetItem(ID);

210 │ if (drU != null)

211 │ {

212 │ drU.Password = Utility.GetSHA512(Password);

213 │ return Update(drU);

214 │ }

215 │

216 │ return -1;

217 │ }

218 │

219 │ public int ChangePassword(Guid ID, string OldPassword, string NewPassword)

220 │ {

221 │ SysUser drU = GetItem(ID);

222 │

223 │ if (drU != null)

224 │ {

225 │ if (drU.Password == Utility.GetSHA512(OldPassword))

226 │ return ResetPassword(ID, NewPassword);

227 │ return -2;

228 │ }

229 │

230 │ return -1;

231 │ }

232 │ }

233 │ }

────────────────────────────────────────────────────────────

**📄 sys\_UserRoleModel.cs (CS, 79 lines)**

1 │ ﻿using QuanLyBanHang.Common;

2 │ using QuanLyBanHang.Models.QLBHEntity;

3 │ using Microsoft.EntityFrameworkCore;

4 │

5 │ namespace QuanLyBanHang.Models.ExModel.SYS

6 │ {

7 │ public class sys\_UserRoleModel : Base<SysUserRole>

8 │ {

9 │ public sys\_UserRoleModel(DbContext dbContext) : base(dbContext)

10 │ {

11 │ }

12 │ public SysUserRole GetItem(Guid UserRoleID)

13 │ {

14 │ return First(t => t.Ma == UserRoleID && !(t.Deleted ?? false));

15 │ }

16 │

17 │ public List<SysUserRole> GetData()

18 │ {

19 │ return Get(t => !(t.Deleted ?? false)).OrderByDescending(t => t.CreatedDate).ToList();

20 │ }

21 │

22 │ public List<SysUserRole> GetDataFollowUserID(Guid UserID)

23 │ {

24 │ return Get(t => t.UserId == UserID && !(t.Deleted ?? false)).OrderByDescending(t => t.CreatedDate).ToList();

25 │ }

26 │ public int Edit(Guid ID, Guid RoleID, bool Status, Guid UserID, Guid Branch)

27 │ {

28 │ SysUserRole userRoleUpdate = GetItem(ID);

29 │ if (userRoleUpdate != null)

30 │ {

31 │ if (CheckExists(ID, userRoleUpdate.UserId, RoleID, Branch))

32 │ return -3;

33 │

34 │ userRoleUpdate.RoleId = RoleID;

35 │ userRoleUpdate.Status = Status;

36 │ userRoleUpdate.UserModified = UserID;

37 │ userRoleUpdate.BranchId = Branch;

38 │ userRoleUpdate.ModifiedDate = DateTime.Now;

39 │ return Update(userRoleUpdate);

40 │ }

41 │

42 │ return -2;

43 │ }

44 │ public int Add(Guid UserID, Guid RoleID, bool Status, Guid UserCreatedID, Guid Branch)

45 │ {

46 │ if (CheckExists(Utility.defaultUID, UserID, RoleID, Branch))

47 │ return -3;

48 │

49 │ SysUserRole userRoleAdd = new SysUserRole

50 │ {

51 │ UserId = UserID,

52 │ RoleId = RoleID,

53 │ Status = Status,

54 │ BranchId = Branch,

55 │ UserCreated = UserCreatedID,

56 │ CreatedDate = DateTime.Now

57 │ };

58 │ return Add(userRoleAdd);

59 │ }

60 │ public int Delete(Guid ID, Guid UserID)

61 │ {

62 │ SysUserRole drUR = GetItem(ID);

63 │ if (drUR != null)

64 │ {

65 │ drUR.Deleted = true;

66 │ drUR.DeletedDate = DateTime.Now;

67 │ drUR.UserDeleted = UserID;

68 │ return Update(drUR);

69 │ }

70 │ return -2;

71 │ }

72 │

73 │ private bool CheckExists(Guid ID, Guid UserID, Guid RoleID, Guid MaChiNhanh)

74 │ {

75 │ return Any(t => t.Ma != ID && t.UserId == UserID && t.RoleId == RoleID && t.BranchId == MaChiNhanh

76 │ && !(t.Deleted ?? false));

77 │ }

78 │ }

79 │ }

────────────────────────────────────────────────────────────

**📄 tbl\_ChucVuModel.cs (CS, 94 lines)**

1 │ ﻿using QuanLyBanHang.Common;

2 │ using QuanLyBanHang.Models.QLBHEntity;

3 │ using Microsoft.EntityFrameworkCore;

4 │

5 │ namespace QuanLyBanHang.Models.ExModel.SYS

6 │ {

7 │ public class tbl\_ChucVuModel : Base<TblChucVu>

8 │ {

9 │ public tbl\_ChucVuModel(DbContext dbContext) : base(dbContext)

10 │ {

11 │ }

12 │

13 │ public TblChucVu GetItem(Guid MenuID)

14 │ {

15 │ return First(t => t.Ma == MenuID && !(t.Deleted ?? false));

16 │ }

17 │ public List<TblChucVu> GetData()

18 │ {

19 │ return Get(t => !(t.Deleted ?? false)).OrderByDescending(x => x.CreatedDate).ToList();

20 │ }

21 │

22 │ public bool CheckExists(Guid Ma, string Code)

23 │ {

24 │ return Any(t => t.Ma != Ma && t.Code == Code && !(t.Deleted ?? false));

25 │ }

26 │

27 │ public (int row, Guid? id) Add(string Code, string Ten, bool Status, Guid? MaNhomChucVu, Guid UserId)

28 │ {

29 │ if (CheckExists(Utility.defaultUID, Code))

30 │ return (-3, null);

31 │

32 │ TblChucVu newItem = new TblChucVu

33 │ {

34 │ Code = Code,

35 │ Ten = Ten,

36 │ Status = Status,

37 │ CreatedDate = DateTime.Now,

38 │ UserCreated = UserId,

39 │ MaNhomChucVu = MaNhomChucVu ?? Utility.defaultUID,

40 │ };

41 │

42 │ \_dbContext.Set<TblChucVu>().Add(newItem);

43 │ return (\_dbContext.SaveChanges(), newItem.Ma);

44 │ }

45 │ // Summary:

46 │ // Edit menu.

47 │ // Returns:

48 │ // Case when Result = 1 then successful

49 │ // when Result = 0 then failure

50 │ // when Result = -2 then not exist

51 │ // when Result = -3 then object is duplicate

52 │ public int Edit(Guid Ma, string Code, string Ten, bool Status, Guid? MaNhomChucVu, Guid UserId)

53 │ {

54 │ if (CheckExists(Ma, Code))

55 │ return -3;

56 │

57 │ TblChucVu itemCurrent = GetItem(Ma);

58 │

59 │ if (itemCurrent != null)

60 │ {

61 │ itemCurrent.Code = Code;

62 │ itemCurrent.Ten = Ten;

63 │ itemCurrent.Status = Status;

64 │ itemCurrent.ModifiedDate = DateTime.Now;

65 │ itemCurrent.UserModified = UserId;

66 │ itemCurrent.MaNhomChucVu = MaNhomChucVu ?? Utility.defaultUID;

67 │ return Update(itemCurrent);

68 │ }

69 │ return -2;

70 │ }

71 │

72 │ // Summary:

73 │ // Delete menu.

74 │ // Returns:

75 │ // Case when Result = 1 then successful

76 │ // when Result = 0 then failure

77 │ // when Result = -1 then cannot delete

78 │ // when Result = -2 then not exist

79 │ public int Delete(Guid ID, Guid UserID)

80 │ {

81 │ TblChucVu drM = GetItem(ID);

82 │

83 │ if (drM != null)

84 │ {

85 │ drM.Deleted = true;

86 │ drM.DeletedDate = DateTime.Now;

87 │ drM.UserDeleted = UserID;

88 │ return Update(drM);

89 │ }

90 │

91 │ return -2;

92 │ }

93 │ }

94 │ }

────────────────────────────────────────────────────────────

**📄 tbl\_NhomChucVuModel.cs (CS, 92 lines)**

1 │ ﻿using QuanLyBanHang.Common;

2 │ using QuanLyBanHang.Models.QLBHEntity;

3 │ using Microsoft.EntityFrameworkCore;

4 │

5 │ namespace QuanLyBanHang.Models.ExModel.SYS

6 │ {

7 │ public class tbl\_NhomChucVuModel : Base<TblNhomChucVu>

8 │ {

9 │ public tbl\_NhomChucVuModel(DbContext dbContext) : base(dbContext)

10 │ {

11 │ }

12 │

13 │ public TblNhomChucVu GetItem(Guid MenuID)

14 │ {

15 │ return First(t => t.Ma == MenuID && !(t.Deleted ?? false));

16 │ }

17 │ public List<TblNhomChucVu> GetData()

18 │ {

19 │ return Get(t => !(t.Deleted ?? false)).OrderByDescending(x => x.CreatedDate).ToList();

20 │ }

21 │

22 │ public bool CheckExists(Guid Ma, string Code)

23 │ {

24 │ return Any(t => t.Ma != Ma && t.Code == Code && !(t.Deleted ?? false));

25 │ }

26 │

27 │ public (int row, Guid? id) Add(string Code, string Ten, bool Status, Guid UserId)

28 │ {

29 │ if (CheckExists(Utility.defaultUID, Code))

30 │ return (-3, null);

31 │

32 │ TblNhomChucVu newItem = new TblNhomChucVu

33 │ {

34 │ Code = Code,

35 │ Ten = Ten,

36 │ Status = Status,

37 │ CreatedDate = DateTime.Now,

38 │ UserCreated = UserId,

39 │ };

40 │

41 │ \_dbContext.Set<TblNhomChucVu>().Add(newItem);

42 │ return (\_dbContext.SaveChanges(), newItem.Ma);

43 │ }

44 │ // Summary:

45 │ // Edit menu.

46 │ // Returns:

47 │ // Case when Result = 1 then successful

48 │ // when Result = 0 then failure

49 │ // when Result = -2 then not exist

50 │ // when Result = -3 then object is duplicate

51 │ public int Edit(Guid Ma, string Code, string Ten, bool Status, Guid UserId)

52 │ {

53 │ if (CheckExists(Ma, Code))

54 │ return -3;

55 │

56 │ TblNhomChucVu itemCurrent = GetItem(Ma);

57 │

58 │ if (itemCurrent != null)

59 │ {

60 │ itemCurrent.Code = Code;

61 │ itemCurrent.Ten = Ten;

62 │ itemCurrent.Status = Status;

63 │ itemCurrent.ModifiedDate = DateTime.Now;

64 │ itemCurrent.UserModified = UserId;

65 │ return Update(itemCurrent);

66 │ }

67 │ return -2;

68 │ }

69 │

70 │ // Summary:

71 │ // Delete menu.

72 │ // Returns:

73 │ // Case when Result = 1 then successful

74 │ // when Result = 0 then failure

75 │ // when Result = -1 then cannot delete

76 │ // when Result = -2 then not exist

77 │ public int Delete(Guid ID, Guid UserID)

78 │ {

79 │ TblNhomChucVu drM = GetItem(ID);

80 │

81 │ if (drM != null)

82 │ {

83 │ drM.Deleted = true;

84 │ drM.DeletedDate = DateTime.Now;

85 │ drM.UserDeleted = UserID;

86 │ return Update(drM);

87 │ }

88 │

89 │ return -2;

90 │ }

91 │ }

92 │ }

────────────────────────────────────────────────────────────

**📄 vChiTietNhapHangModel.cs (CS, 29 lines)**

1 │ ﻿using Microsoft.EntityFrameworkCore;

2 │ using QuanLyBanHang.Models.QLBHEntity;

3 │

4 │ namespace QuanLyBanHang.Models.ExModel.OPS

5 │ {

6 │ public class vChiTietNhapHangModel : Base<VChiTietNhapHang>

7 │ {

8 │ public vChiTietNhapHangModel(DbContext dbContext) : base(dbContext)

9 │ {

10 │ }

11 │

12 │ QuanLyBanHangContext dbContext

13 │ {

14 │ get { return \_dbContext as QuanLyBanHangContext; }

15 │ set { return; }

16 │ }

17 │

18 │

19 │ public VChiTietNhapHang GetItem(int Ma)

20 │ {

21 │ return First(t => t.Ma == Ma);

22 │ }

23 │

24 │ public List<VChiTietNhapHang> GetData(int MaNhapHang)

25 │ {

26 │ return Get(t => t.MaNhapHang == MaNhapHang).ToList();

27 │ }

28 │ }

29 │ }

────────────────────────────────────────────────────────────

**📄 vKhay.cs (CS, 25 lines)**

1 │ ﻿namespace QuanLyBanHang.Models.ExModel.VTTH

2 │ {

3 │ public class vKhay

4 │ {

5 │ public int Ma { get; set; }

6 │

7 │ public int? MaKhoTieuHao { get; set; }

8 │

9 │ public int? MaKhoNguyenLieu { get; set; }

10 │

11 │ public int? MaVatTu { get; set; }

12 │

13 │ public decimal? DinhMucKhay { get; set; }

14 │

15 │ public bool? DaNhap { get; set; }

16 │

17 │ public bool? KhayTreo { get; set; }

18 │

19 │ public string? MoTa { get; set; }

20 │

21 │ public decimal? ConLai { get; set; }

22 │

23 │ public bool? Status { get; set; }

24 │ }

25 │ }