**📄 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"),

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 │ {

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 │ }