**PHẦN MỀM QUẢN LÝ ĐĂNG KÝ HỌC PHẦN**

1914775 – Đinh Trọng Đạt

**Các chức năng đã xây dựng:**

1. **Chức năng thuộc quyền Sinh viên:**

Sinh viên đăng nhập bằng tài khoản của sinh viên đã được cấp.

* 1. **Đăng ký/Điều chỉnh học phần.**

**Trường hợp sinh viên chưa thanh toán học phí của kỳ trước sẽ không được đăng ký học phần của kỳ này.**

private void DKHPForm\_Load(object sender, EventArgs e){

string preYear = (DateTime.Now.Year - 1).ToString() + " - " + DateTime.Now.Year.ToString();

LoadHPToLV();

if (HocPhiBL.getInstance().KiemTraHocPhi(user, preYear) == 1)

{

MessageBox.Show("Sinh viên vui lòng hoàn thành học phí còn nợ trước khi đăng ký học phần");

btnKeHoach.Enabled = false;

btnCaiThien.Enabled = false;

btnDCHP.Enabled = false;

}

}

**Danh sách học phần mà sinh viên chưa tích luỹ sẽ hiển thị trên danh sách.**

private void LoadHPToLV()

{

HocPhanBL hpBL = HocPhanBL.getInstance();

listHP = hpBL.GetHPChuaTL(int.Parse(user));

lvHP.Items.Clear();

foreach (var hp in listHP)

{

ListViewItem item = lvHP.Items.Add(hp.MaHP.ToString());

item.SubItems.Add(hp.TenHP);

item.SubItems.Add(hp.LoaiHP);

item.SubItems.Add(hp.TongSoTC.ToString());

}

}

**Chọn đăng ký theo kế hoạch:** sẽ hiển thị những môn học được giảng dạy trọng học kỳ này.

/// <summary>

/// Hàm lấy danh sách các môn học đã đăng ký trong học kỳ này

/// </summary>

private void LoadCurrent\_DSKQ()

{

int soLuongDangKy = 0;

HocPhanBL hpBL = HocPhanBL.getInstance();

listKQ = hpBL.GetCurrentKQHP(currentSV.MSSV, int.Parse(cbbHK.Text == "" ? "0" : cbbHK.Text), currentYear);

lvKQDK.Items.Clear();

foreach (var hp in listKQ)

{

soLuongDangKy += hp.TongSoTC;

txtQuantity.Text = soLuongDangKy.ToString();

ListViewItem item = lvKQDK.Items.Add(hp.MaHP.ToString());

item.SubItems.Add(hp.TenHP);

item.SubItems.Add(hp.LoaiHP);

item.SubItems.Add(hp.TongSoTC.ToString());

item.SubItems.Add(hp.TCLT.ToString());

item.SubItems.Add(hp.TCTH.ToString());

}

}

/// <summary>

/// Hàm tính năm học của sinh viên: vd

/// 1914775 -> sinh viên năm 3

/// </summary>

/// <param name="user"></param>

/// <returns></returns>

private int GetNam(string user)

{

int curYear = DateTime.Now.Year;

currentSV = svBL.GetByMSSV(int.Parse(user));

return curYear - 2000 - int.Parse(user.Substring(0, 2)) + 1;

}

/// <summary>

/// Xuất danh sách học phần đã được lên kế hoạch giảng dạy cho từng học kì

/// Tương ứng với học kỳ và năm học của sinh viên đăng nhập

/// sẽ xuất ra danh sách học phần tương ứng.

/// </summary>

private void LoadDSHP()

{

HocPhanBL hpBL = HocPhanBL.getInstance();

listHP = hpBL.GetHPTheoKeHoach(currentSV.MSSV, nam, int.Parse(cbbHK.Text == "" ? "0" : cbbHK.Text), currentSV.Khoa);

lvHP.Items.Clear();

foreach (var hp in listHP)

{

ListViewItem item = lvHP.Items.Add(hp.MaHP.ToString());

item.SubItems.Add(hp.TenHP);

item.SubItems.Add(hp.LoaiHP);

item.SubItems.Add(hp.TongSoTC.ToString());

item.SubItems.Add(hp.TCLT.ToString());

item.SubItems.Add(hp.TCTH.ToString());

}

}

/// <summary>

/// Load danh sách kết quả đăng ký học phần

/// </summary>

/// <param name="hocphans"></param>

private void LoadKQHP(List<HocPhan> hocphans)

{

int soLuongDangKy = 0;

txtQuantity.Text = soLuongDangKy.ToString();

lvKQDK.Items.Clear();

foreach (var hp in hocphans)

{

soLuongDangKy += hp.TongSoTC;

txtQuantity.Text = soLuongDangKy.ToString();

if (soLuongDangKy <= 25)

{

txtWanning.Visible = false;

btnDangKy.Enabled = true;

ListViewItem item = lvKQDK.Items.Add(hp.MaHP);

item.SubItems.Add(hp.TenHP);

item.SubItems.Add(hp.LoaiHP);

item.SubItems.Add(hp.TongSoTC.ToString());

item.SubItems.Add(hp.TCLT.ToString());

item.SubItems.Add(hp.TCTH.ToString());

}

else if (soLuongDangKy > 25)

{

txtWanning.Visible = true;

btnDangKy.Enabled = false;

}

}

}

private void cbbHK\_SelectedIndexChanged(object sender, EventArgs e)

{

if (cbbHK.Text == "") return;

LoadDSHP();

LoadCurrent\_DSKQ();

if (listKQ.Count > 0)

{

if (MessageBox.Show("Bạn đã đăng ký học phần của học kỳ này, Chọn chức năng \"Điều chỉnh học phần\" để điều chỉnh?", "Hệ thống", MessageBoxButtons.YesNo) == DialogResult.Yes)

{

DCKQDKForm frm = new DCKQDKForm(currentSV.MSSV.ToString());

frm.ShowDialog(this);

}

}

}

**Chọn học phần đăng ký, sau khi sinh viên nhấn đăng ký, những học phần đã được đăng ký sẽ được lưu lại trên cơ sở dữ liệu.**

/// <summary>

/// Lấy thông tin của học phần trên listview

/// </summary>

/// <param name="item"></param>

/// <returns></returns>

private HocPhan GetHPLV(ListViewItem item)

{

HocPhan hp = new HocPhan();

hp.MaHP = item.SubItems[0].Text;

hp.TenHP = item.SubItems[1].Text;

hp.LoaiHP = item.SubItems[2].Text;

hp.TongSoTC = int.Parse(item.SubItems[3].Text);

hp.TCLT = int.Parse(item.SubItems[4].Text);

hp.TCTH = int.Parse(item.SubItems[5].Text);

return hp;

}

/// <summary>

/// Đăng ký học phần

/// </summary>

/// <returns></returns>

private int InsertChiTietDK()

{

HocPhanBL hpBL = HocPhanBL.getInstance();

ChiTietDangKy ct = new ChiTietDangKy();

if (lvKQDK.Items.Count < 0)

MessageBox.Show("Chưa chọn học phần để đăng ký, vui lòng chọn học phần");

else

{

listKQ = hpBL.GetCurrentKQHP(currentSV.MSSV, int.Parse(cbbHK.Text == "" ? "0" : cbbHK.Text), currentYear);

int i = this.lvHP.CheckedItems.Count - 1;

while (i >= 0)

{

listKQ.Add(GetHPLV(this.lvHP.CheckedItems[i]));

i--;

}

foreach (var item in listKQ)

{

ct.MSSV = currentSV.MSSV;

ct.MaHP = item.MaHP;

ct.NgayDangKy = DateTime.Now.ToShortDateString();

ct.HocKy = int.Parse(cbbHK.Text);

ct.NamHoc = DateTime.Now.Year.ToString() + " - " + (DateTime.Now.Year + 1).ToString();

ChiTietDKBL ctBL = ChiTietDKBL.getInstance();

ctBL.Insert(ct);

}

return 1;

}

return -1;

}

private void btnDangKy\_Click(object sender, EventArgs e)

{

int result = InsertChiTietDK();

if (result > 0)

{

AddHocPhi();

MessageBox.Show("Đăng ký học phần thành công !!");

LoadDSHP();

cbbHK.Enabled = false;

}

else MessageBox.Show("Thêm dữ liệu không thành công. Vui lòng kiểm tra lại dữ liệu nhập");

}

**Trường hợp sinh viên vừa đăng ký học phần** mà lỡ tay chọn lại sẽ xuất thông báo hỏi có cho phép chuyển sang màn hình chức năng điều chỉnh học phần.

private void cbbHK\_SelectedIndexChanged(object sender, EventArgs e)

{

if (cbbHK.Text == "") return;

LoadDSHP();

LoadCurrent\_DSKQ();

if (listKQ.Count > 0)

{

if (MessageBox.Show("Bạn đã đăng ký học phần của học kỳ này, Chọn chức năng \"Điều chỉnh học phần\" để điều chỉnh?", "Hệ thống", MessageBoxButtons.YesNo) == DialogResult.Yes)

{

DCKQDKForm frm = new DCKQDKForm(currentSV.MSSV.ToString());

frm.ShowDialog(this);

}

}

}

**Chức năng xuất kết quả đăng ký:**

/// <summary>

/// Xuất kêt quả đăng ký

/// </summary>

/// <param name="lv"></param>

/// <param name="listKQDK"></param>

/// <param name="path"></param>

private void WriteToExcel(ListView lv, List<HocPhan> listKQDK, string path)

{

ExcelPackage.LicenseContext = OfficeOpenXml.LicenseContext.NonCommercial;

using (ExcelPackage p = new ExcelPackage())

{

p.Workbook.Worksheets.Add("sheet 1");

ExcelWorksheet ws = p.Workbook.Worksheets[0];

ws.Name = "sheet 1";

ws.Cells.Style.Font.Size = 11;

ws.Cells.Style.Font.Name = "Calibri";

List<string> arrCollumHeader = new List<string>();

foreach (ColumnHeader item in lv.Columns)

{

arrCollumHeader.Add(item.Text);

}

var countColHeader = arrCollumHeader.Count();

int colIndex = 1;

int rowIndex = 1;

//Tao cac Header

foreach (var item in arrCollumHeader)

{

var cell = ws.Cells[rowIndex, colIndex];

var style = cell.Style.Font;

style.Bold = true;

cell.Value = item;

colIndex++;

}

//Lay danh sach sinh vien

foreach (var item in listKQDK)

{

colIndex = 1;

rowIndex++;

ws.Cells[rowIndex, colIndex++].Value = item.MaHP;

ws.Cells[rowIndex, colIndex++].Value = item.TenHP;

ws.Cells[rowIndex, colIndex++].Value = item.LoaiHP;

ws.Cells[rowIndex, colIndex++].Value = item.TongSoTC;

ws.Cells[rowIndex, colIndex++].Value = item.TCLT;

ws.Cells[rowIndex, colIndex++].Value = item.TCTH;

}

//save file

Byte[] bin = p.GetAsByteArray();

File.WriteAllBytes(path, bin);

}

}

private void btnXuat\_Click(object sender, EventArgs e)

{

saveFileDialog1.FileName = string.Format("DangKyHP SV{0} Hoc Ky {1} {2}", currentSV.MSSV.ToString(), cbbHK.Text, DateTime.Now.Year.ToString() + " - " + (DateTime.Now.Year + 1).ToString());

saveFileDialog1.InitialDirectory = @"E:\";

saveFileDialog1.DefaultExt = "xlsx";

saveFileDialog1.Filter = "Excel 2007 files(xlsx) (\*.xlsx)|\*.xlsx";

List<HocPhan> kqhp = new List<HocPhan>();

int i = this.lvKQDK.Items.Count - 1;

while (i >= 0)

{

kqhp.Add(GetHPLV(this.lvKQDK.Items[i]));

i--;

}

DialogResult dlg = saveFileDialog1.ShowDialog();

if (dlg == DialogResult.OK)

{

string path = string.Format(@"{0}", saveFileDialog1.FileName);

WriteToExcel(lvKQDK, kqhp, path);

MessageBox.Show("Xuất phiếu đăng ký thành công!");

}

}

**Chức năng điều chỉnh học phần:** cho phép sinh viên điều chỉnh(thêm, xoá) những học phần vừa đăng ký.

#region

/// <summary>

/// Hàm tính năm học của sinh viên: vd

/// 1914775 -> sinh viên năm 3

/// </summary>

/// <param name="user"></param>

/// <returns></returns>

private int GetNam(string user)

{

int curYear = DateTime.Now.Year;

currentSV = svBL.GetByMSSV(int.Parse(user));

return curYear - 2000 - int.Parse(user.Substring(0, 2)) + 1;

}

/// <summary>

/// Hàm lấy danh sách các môn học đã đăng ký trong học kỳ này

/// </summary>

private void LoadCurrent\_DSKQ()

{

HocPhanBL hpBL = HocPhanBL.getInstance();

listKQ = hpBL.GetCurrentKQHP(currentSV.MSSV, int.Parse(cbbHK.Text == "" ? "0" : cbbHK.Text), currentYear);

lvKQDK.Items.Clear();

foreach (var hp in listKQ)

{

lbTitle.Text = String.Format("Danh sách học phần sinh viên {0} đã đăng ký trong học kỳ {1}",

currentSV.HoLot + " " + currentSV.Ten, hp.HocKy);

soLuongDangKy += hp.TongSoTC;

txtQuantity.Text = soLuongDangKy.ToString();

ListViewItem item = lvKQDK.Items.Add(hp.MaHP.ToString());

item.SubItems.Add(hp.TenHP);

item.SubItems.Add(hp.LoaiHP);

item.SubItems.Add(hp.TongSoTC.ToString());

item.SubItems.Add(hp.TCLT.ToString());

item.SubItems.Add(hp.TCTH.ToString());

}

}

/// <summary>

/// Xuất kết quả đăng ký học phần

/// </summary>

/// <param name="hocphans"></param>

private void LoadKQHP(List<HocPhan> hocphans)

{

int soLuongDangKy = 0;

txtQuantity.Text = soLuongDangKy.ToString();

lvKQDK.Items.Clear();

foreach (var hp in hocphans)

{

soLuongDangKy += hp.TongSoTC;

txtQuantity.Text = soLuongDangKy.ToString();

if (soLuongDangKy <= 25)

{

txtWanning.Visible = false;

btnDangKy.Enabled = true;

ListViewItem item = lvKQDK.Items.Add(hp.MaHP);

item.SubItems.Add(hp.TenHP);

item.SubItems.Add(hp.LoaiHP);

item.SubItems.Add(hp.TongSoTC.ToString());

item.SubItems.Add(hp.TCLT.ToString());

item.SubItems.Add(hp.TCTH.ToString());

}

else if (soLuongDangKy > 25)

{

txtWanning.Visible = true;

btnDangKy.Enabled = false;

}

}

}

/// <summary>

/// Xuất danh sách học phần đã được lên kế hoạch giảng dạy cho từng học kì

/// Tương ứng với học kỳ và năm học của sinh viên đăng nhập

/// sẽ xuất ra danh sách học phần tương ứng.

/// </summary>

private void LoadDSHP()

{

HocPhanBL hpBL = HocPhanBL.getInstance();

listHP = hpBL.GetHPTheoKeHoach(currentSV.MSSV, nam, int.Parse(cbbHK.Text == "" ? "0" : cbbHK.Text), currentSV.Khoa);

lvHP.Items.Clear();

foreach (var hp in listHP)

{

ListViewItem item = lvHP.Items.Add(hp.MaHP.ToString());

item.SubItems.Add(hp.TenHP);

item.SubItems.Add(hp.LoaiHP);

item.SubItems.Add(hp.TongSoTC.ToString());

item.SubItems.Add(hp.TCLT.ToString());

item.SubItems.Add(hp.TCTH.ToString());

}

}

/// <summary>

/// Lấy thông tin của học phần trên listview

/// </summary>

/// <param name="item"></param>

/// <returns></returns>

private HocPhan GetHPLV(ListViewItem item)

{

HocPhan hp = new HocPhan();

hp.MaHP = item.SubItems[0].Text;

hp.TenHP = item.SubItems[1].Text;

hp.LoaiHP = item.SubItems[2].Text;

hp.TongSoTC = int.Parse(item.SubItems[3].Text);

hp.TCLT = int.Parse(item.SubItems[4].Text);

hp.TCTH = int.Parse(item.SubItems[5].Text);

return hp;

}

private int DieuChinhKQHP()

{

ChiTietDangKy ct = new ChiTietDangKy();

if (lvKQDK.Items.Count < 0)

MessageBox.Show("Chưa chọn học phần để đăng ký, vui lòng chọn học phần");

else

{

foreach (var item in listKQ)

{

ct.MSSV = currentSV.MSSV;

ct.MaHP = item.MaHP;

ct.NgayDangKy = DateTime.Now.ToShortDateString();

ct.HocKy = int.Parse(cbbHK.Text);

ct.NamHoc = DateTime.Now.Year.ToString() + " - " + (DateTime.Now.Year + 1).ToString();

ChiTietDKBL ctBL = ChiTietDKBL.getInstance();

ctBL.DieuChinhHocPhan(ct);

}

return 1;

}

return -1;

}

/// <summary>

/// Xoá học phần vừa đăng ký ra khỏi phiếu học phần

/// </summary>

/// <param name="mssv"></param>

/// <param name="mahp"></param>

/// <returns></returns>

private int Delete(int mssv, string mahp)

{

ChiTietDKBL ctdkBL = ChiTietDKBL.getInstance();

return ctdkBL.DeleteByKey(mssv, mahp);

}

#endregion

private void cbbHK\_SelectedIndexChanged(object sender, EventArgs e)

{

if (cbbHK.Text == "") return;

LoadCurrent\_DSKQ();

LoadDSHP();

}

private void DCKQDKForm\_Load(object sender, EventArgs e)

{

LoadDSHP();

}

private void btnDangKy\_Click(object sender, EventArgs e)

{

int result = DieuChinhKQHP();

if (result > 0)

{

UpdateHocPhi();

MessageBox.Show("Điều chỉnh học phần thành công !!");

LoadDSHP();

cbbHK.Enabled = false;

}

else MessageBox.Show("Cập nhật dữ liệu không thành công. Vui lòng kiểm tra lại dữ liệu nhập");

}

private void lvHP\_ItemChecked(object sender, ItemCheckedEventArgs e)

{

HocPhanBL hpBL = HocPhanBL.getInstance();

listKQ = hpBL.GetCurrentKQHP(currentSV.MSSV, int.Parse(cbbHK.Text == "" ? "0" : cbbHK.Text), currentYear);

int i = this.lvHP.CheckedItems.Count - 1;

while (i >= 0)

{

listKQ.Add(GetHPLV(this.lvHP.CheckedItems[i]));

i--;

}

LoadKQHP(listKQ);

}

private void tsmDelete\_Click(object sender, EventArgs e)

{

ListViewItem item;

int count = lvKQDK.Items.Count - 1;

for (int i = count; i >= 0; i--)

{

item = lvKQDK.Items[i];

if (item.Selected)

{

int result = Delete(currentSV.MSSV, item.SubItems[0].Text.ToString().Trim());

if (result > 0)

{

UpdateHocPhi();

lvKQDK.Items.Remove(item);

LoadDSHP();

}

else MessageBox.Show("Cập nhật dữ liệu không thành công. Vui lòng kiểm tra lại dữ liệu nhập");

}

}

}

Học phí sẽ được tự động tính sau khi sinh viên đăng ký.

private int TongTien()

{

int TongTien = 0;

int i = this.lvKQDK.Items.Count - 1;

while (i >= 0)

{

var hocPhan = GetHPLV(this.lvKQDK.Items[i]);

TongTien += hocPhan.TCLT \* 300000 + hocPhan.TCTH \* 380000;

i--;

}

return TongTien;

}

/// <summary>

/// Hàm thêm học phí cho sinh viên

/// </summary>

/// <param name="hp"></param>

/// <returns></returns>

private int AddHocPhi()

{

HocPhi hp = new HocPhi();

HocPhiBL hocPhiBL = HocPhiBL.getInstance();

hp.MSSV = currentSV.MSSV;

hp.HocKy = int.Parse(cbbHK.Text);

hp.NamHoc = DateTime.Now.Year.ToString() + " - " + (DateTime.Now.Year + 1).ToString();

hp.SoTien = TongTien();

hp.CapNhat = DateTime.Now.ToShortDateString();

hp.TinhTrang = false;

return hocPhiBL.Insert(hp);

}

**Chức năng đăng ký học phần ngoài kế hoạch** cũng tương tự chức năng đăng ký theo kế hoạch, sinh viên được quyền đăng ký học cải thiện, học lại được mở trong học kỳ này của các khoá.

#region

/// <summary>

/// Hàm lấy danh sách các môn học đã đăng ký trong học kỳ này

/// </summary>

private void LoadCurrent\_DSKQ()

{

int soLuongDangKy = 0;

HocPhanBL hpBL = HocPhanBL.getInstance();

listKQ = hpBL.GetCurrentKQHP(currentSV.MSSV, int.Parse(cbbHK.Text == "" ? "0" : cbbHK.Text), currentYear);

lvKQDK.Items.Clear();

foreach (var hp in listKQ)

{

soLuongDangKy += hp.TongSoTC;

txtQuantity.Text = soLuongDangKy.ToString();

ListViewItem item = lvKQDK.Items.Add(hp.MaHP.ToString());

item.SubItems.Add(hp.TenHP);

item.SubItems.Add(hp.LoaiHP);

item.SubItems.Add(hp.TongSoTC.ToString());

item.SubItems.Add(hp.TCLT.ToString());

item.SubItems.Add(hp.TCTH.ToString());

}

}

/// <summary>

/// Xuất danh sách học phần Ngoài kế hoạch

/// Tương ứng với khoa của sinh viên đăng nhập

/// sẽ xuất ra danh sách học phần tương ứng. Dành cho sinh viên học cải thiện.

/// </summary>

private void LoadDSHP()

{

HocPhanBL hpBL = HocPhanBL.getInstance();

listHP = hpBL.GetHPNgoaiKeHoach(currentSV.MSSV, int.Parse(cbbHK.Text == "" ? "0" : cbbHK.Text), currentSV.Khoa);

lvHP.Items.Clear();

foreach (var hp in listHP)

{

ListViewItem item = lvHP.Items.Add(hp.MaHP.ToString());

item.SubItems.Add(hp.TenHP);

item.SubItems.Add(hp.LoaiHP);

item.SubItems.Add(hp.TongSoTC.ToString());

item.SubItems.Add(hp.TCLT.ToString());

item.SubItems.Add(hp.TCTH.ToString());

}

}

/// <summary>

/// Load danh sách kết quả đăng ký học phần

/// </summary>

/// <param name="hocphans"></param>

private void LoadKQHP(List<HocPhan> hocphans)

{

int soLuongDangKy = 0;

txtQuantity.Text = soLuongDangKy.ToString();

lvKQDK.Items.Clear();

foreach (var hp in hocphans)

{

soLuongDangKy += hp.TongSoTC;

txtQuantity.Text = soLuongDangKy.ToString();

if (soLuongDangKy <= 25)

{

txtWanning.Visible = false;

btnDangKy.Enabled = true;

ListViewItem item = lvKQDK.Items.Add(hp.MaHP);

item.SubItems.Add(hp.TenHP);

item.SubItems.Add(hp.LoaiHP);

item.SubItems.Add(hp.TongSoTC.ToString());

item.SubItems.Add(hp.TCLT.ToString());

item.SubItems.Add(hp.TCTH.ToString());

}

else if (soLuongDangKy > 25)

{

txtWanning.Visible = true;

btnDangKy.Enabled = false;

}

}

}

/// <summary>

/// Lấy thông tin của học phần trên listview

/// </summary>

/// <param name="item"></param>

/// <returns></returns>

private HocPhan GetHPLV(ListViewItem item)

{

HocPhan hp = new HocPhan();

hp.MaHP = item.SubItems[0].Text;

hp.TenHP = item.SubItems[1].Text;

hp.LoaiHP = item.SubItems[2].Text;

hp.TongSoTC = int.Parse(item.SubItems[3].Text);

hp.TCLT = int.Parse(item.SubItems[4].Text);

hp.TCTH = int.Parse(item.SubItems[5].Text);

return hp;

}

/// <summary>

/// Đăng ký học phần

/// </summary>

/// <returns></returns>

private int InsertChiTietDK()

{

ChiTietDangKy ct = new ChiTietDangKy();

if (lvKQDK.Items.Count < 0)

MessageBox.Show("Chưa chọn học phần để đăng ký, vui lòng chọn học phần");

else

{

listKQ = HocPhanBL.getInstance().GetCurrentKQHP(currentSV.MSSV, int.Parse(cbbHK.Text == "" ? "0" : cbbHK.Text), currentYear);

int i = this.lvHP.CheckedItems.Count - 1;

while (i >= 0)

{

listKQ.Add(GetHPLV(this.lvHP.CheckedItems[i]));

i--;

}

foreach (var item in listKQ)

{

ct.MSSV = currentSV.MSSV;

ct.MaHP = item.MaHP;

ct.NgayDangKy = DateTime.Now.ToShortDateString();

ct.HocKy = int.Parse(cbbHK.Text);

ct.NamHoc = DateTime.Now.Year.ToString() + " - " + (DateTime.Now.Year + 1).ToString();

ChiTietDKBL ctBL = ChiTietDKBL.getInstance();

ctBL.Insert(ct);

}

return 1;

}

return -1;

}

/// <summary>

/// Hàm tính học phí cho sinh viên

/// </summary>

/// <returns></returns>

private int TongTien()

{

int TongTien = 0;

int i = this.lvKQDK.Items.Count - 1;

while (i >= 0)

{

var hocPhan = GetHPLV(this.lvKQDK.Items[i]);

TongTien += hocPhan.TCLT \* 300000 + hocPhan.TCTH \* 380000;

i--;

}

return TongTien;

}

/// <summary>

/// Hàm thêm học phí cho sinh viên

/// </summary>

/// <param name="hp"></param>

/// <returns></returns>

private int AddHocPhi()

{

HocPhi hp = new HocPhi();

HocPhiBL hocPhiBL = HocPhiBL.getInstance();

hp.MSSV = currentSV.MSSV;

hp.HocKy = int.Parse(cbbHK.Text);

hp.NamHoc = DateTime.Now.Year.ToString() + " - " + (DateTime.Now.Year + 1).ToString();

hp.SoTien = TongTien();

hp.CapNhat = DateTime.Now.ToShortDateString();

hp.TinhTrang = false;

return hocPhiBL.Insert(hp);

}

#endregion

private void btnDangKy\_Click(object sender, EventArgs e)

{

int result = InsertChiTietDK();

if (result > 0)

{

AddHocPhi();

MessageBox.Show("Đăng ký học phần thành công !!");

LoadDSHP();

cbbHK.Enabled = false;

}

else MessageBox.Show("Thêm dữ liệu không thành công. Vui lòng kiểm tra lại dữ liệu nhập");

}

private void cbbHK\_SelectedIndexChanged(object sender, EventArgs e)

{

if (cbbHK.Text == "") return;

LoadDSHP();

LoadCurrent\_DSKQ();

if (listKQ.Count > 0)

{

if (MessageBox.Show("Bạn đã đăng ký học phần của học kỳ này, Chọn chức năng \"Điều chỉnh học phần\" để điều chỉnh?", "Hệ thống", MessageBoxButtons.YesNo) == DialogResult.Yes)

{

DCKQDKForm frm = new DCKQDKForm(currentSV.MSSV.ToString());

frm.ShowDialog(this);

}

}

}

private void DKNgoaiKHForm\_Load(object sender, EventArgs e)

{

LoadDSHP();

}

private void lvHP\_ItemChecked(object sender, ItemCheckedEventArgs e)

{

listKQ = HocPhanBL.getInstance().GetCurrentKQHP(currentSV.MSSV, int.Parse(cbbHK.Text == "" ? "0" : cbbHK.Text), currentYear);

int i = this.lvHP.CheckedItems.Count - 1;

while (i >= 0)

{

listKQ.Add(GetHPLV(this.lvHP.CheckedItems[i]));

i--;

}

LoadKQHP(listKQ);

}

private void btnXuat\_Click(object sender, EventArgs e)

{

saveFileDialog1.FileName = string.Format("DangKyHP SV{0} Hoc Ky {1} {2}", currentSV.MSSV.ToString(), cbbHK.Text, DateTime.Now.Year.ToString() + " - " + (DateTime.Now.Year + 1).ToString());

saveFileDialog1.InitialDirectory = @"E:\";

saveFileDialog1.DefaultExt = "xlsx";

saveFileDialog1.Filter = "Excel 2007 files(xlsx) (\*.xlsx)|\*.xlsx";

List<HocPhan> kqhp = new List<HocPhan>();

int i = this.lvKQDK.Items.Count - 1;

while (i >= 0)

{

kqhp.Add(GetHPLV(this.lvKQDK.Items[i]));

i--;

}

DialogResult dlg = saveFileDialog1.ShowDialog();

if (dlg == DialogResult.OK)

{

string path = string.Format(@"{0}", saveFileDialog1.FileName);

WriteToExcel(lvKQDK, kqhp, path);

MessageBox.Show("Xuất phiếu đăng ký thành công!");

}

}

**Chức năng xem kết quả đăng ký học phần.**

#region

private void LoadDSHP()

{

HocPhanBL hpBL = HocPhanBL.getInstance();

listHP = hpBL.GetKQDK(currentSV.MSSV);

lvHP.Items.Clear();

foreach (var hp in listHP)

{

ListViewItem item = lvHP.Items.Add(hp.MaHP.ToString());

item.SubItems.Add(hp.TenHP);

item.SubItems.Add(hp.LoaiHP);

item.SubItems.Add(hp.TongSoTC.ToString());

}

}

private void LoadDetail()

{

int SoLuongTC = 0;

lbName.Text = currentSV.HoLot +" "+ currentSV.Ten + " Lớp: " + currentSV.TenLop + " Khoa: "+currentSV.Khoa;

foreach (var item in listHP)

{

SoLuongTC += item.TongSoTC;

}

txtQuantity.Text = SoLuongTC.ToString();

}

/// <summary>

/// Xuất kêt quả đăng ký

/// </summary>

/// <param name="lv"></param>

/// <param name="listKQDK"></param>

/// <param name="path"></param>

private void WriteToExcel(ListView lv, List<HocPhan> listKQDK, string path)

{

ExcelPackage.LicenseContext = OfficeOpenXml.LicenseContext.NonCommercial;

using (ExcelPackage p = new ExcelPackage())

{

p.Workbook.Worksheets.Add("sheet 1");

ExcelWorksheet ws = p.Workbook.Worksheets[0];

ws.Name = "sheet 1";

ws.Cells.Style.Font.Size = 11;

ws.Cells.Style.Font.Name = "Calibri";

List<string> arrCollumHeader = new List<string>();

foreach (ColumnHeader item in lv.Columns)

{

arrCollumHeader.Add(item.Text);

}

var countColHeader = arrCollumHeader.Count();

int colIndex = 1;

int rowIndex = 1;

//Tao cac Header

foreach (var item in arrCollumHeader)

{

var cell = ws.Cells[rowIndex, colIndex];

var style = cell.Style.Font;

style.Bold = true;

cell.Value = item;

colIndex++;

}

//Lay danh sach sinh vien

foreach (var item in listKQDK)

{

colIndex = 1;

rowIndex++;

ws.Cells[rowIndex, colIndex++].Value = item.MaHP;

ws.Cells[rowIndex, colIndex++].Value = item.TenHP;

ws.Cells[rowIndex, colIndex++].Value = item.LoaiHP;

ws.Cells[rowIndex, colIndex++].Value = item.TongSoTC;

ws.Cells[rowIndex, colIndex++].Value = item.TCLT;

ws.Cells[rowIndex, colIndex++].Value = item.TCTH;

}

//save file

Byte[] bin = p.GetAsByteArray();

File.WriteAllBytes(path, bin);

}

}

#endregion

private void btnXuat\_Click(object sender, EventArgs e)

{

saveFileDialog1.FileName = string.Format("KQDangKyHP SV{0}", currentSV.MSSV.ToString());

saveFileDialog1.InitialDirectory = @"E:\";

saveFileDialog1.DefaultExt = "xlsx";

saveFileDialog1.Filter = "Excel 2007 files(xlsx) (\*.xlsx)|\*.xlsx";

DialogResult dlg = saveFileDialog1.ShowDialog();

if (dlg == DialogResult.OK)

{

string path = string.Format(@"{0}", saveFileDialog1.FileName);

WriteToExcel(lvHP, listHP, path);

MessageBox.Show("Xuất phiếu đăng ký thành công!");

}

}

* 1. **Xem lịch thi.**

Sinh viên được xem lịch thi của những học phần đã đăng ky (trong trường hợp giáo vụ đã lập lịch thi cho những môn đó).

SinhVien currentSV;

List<LichThi> lichThiList;

string currentYear = DateTime.Now.Year.ToString() + " - " + (DateTime.Now.Year + 1).ToString();

public LichThiForm(string userName)

{

currentSV = SinhVienBL.getInstance().GetByMSSV(int.Parse(userName));

InitializeComponent();

}

#region Các hàm xử lý

private void GetItem()

{

HocPhanBL hpBL = HocPhanBL.getInstance();

LichThiBL ltBL = LichThiBL.Instance();

lichThiList = ltBL.GetLichThiSV(currentSV.MSSV, int.Parse(cbbHocky.Text == ""? "1" : cbbHocky.Text));

lvLichThi.Items.Clear();

foreach (var lt in lichThiList)

{

string tenHP = hpBL.GetByMaHP(lt.MaHP).TenHP;

int STC = hpBL.GetByMaHP(lt.MaHP).TongSoTC;

lbLichThi.Text = String.Format("Lịch thi học kỳ {0} năm học {1} của sinh viên {2}, lớp {3}, khoa {4}",

cbbHocky.Text, currentYear, currentSV.HoLot + " " + currentSV.Ten, currentSV.TenLop, currentSV.Khoa);

ListViewItem item = lvLichThi.Items.Add(lt.MaHP);

item.SubItems.Add(tenHP);

item.SubItems.Add(STC.ToString());

item.SubItems.Add(lt.NgayThi);

item.SubItems.Add(lt.GioThi);

item.SubItems.Add(lt.ThoiLuong.ToString());

item.SubItems.Add(lt.PhongThi);

item.SubItems.Add(lt.DiaDiem);

item.SubItems.Add(lt.GhiChu);

}

}

#endregion

private void LichThiForm\_Load(object sender, EventArgs e)

{

GetItem();

}

private void cbbHocky\_SelectedIndexChanged(object sender, EventArgs e)

{

GetItem();

}

* 1. **Xem học phí.**

Sinh viên được xem học phí của mình, học phí được tính tự động sau khi đăng ký học phần.

List<HocPhi> listHocPhi;

SinhVien currentSV;

public HocPhiForm(string userName)

{

currentSV = SinhVienBL.getInstance().GetByMSSV(int.Parse(userName));

InitializeComponent();

}

#region

private void GetItem()

{

int TongTien = 0;

HocPhiBL hpBL = HocPhiBL.getInstance();

listHocPhi = hpBL.GetByMSSV(currentSV.MSSV);

lvHP.Items.Clear();

foreach (var hp in listHocPhi)

{

ListViewItem item = lvHP.Items.Add(hp.id.ToString());

item.SubItems.Add(String.Format("Học phí học kỳ {0} năm học {1}", hp.HocKy, hp.NamHoc));

item.SubItems.Add(hp.SoTien.ToString());

item.SubItems.Add(DateTime.Now.ToShortDateString());

TongTien += hp.SoTien;

}

txtHocPhi.Text = String.Format("{0} đồng", TongTien);

if(TongTien == 0)

txtSoTienBangChu.Text = "Sinh viên đã thanh toán học phí";

else

txtSoTienBangChu.Text = Utils.NumberToText(TongTien);

}

#endregion

private void HocPhiForm\_Load(object sender, EventArgs e)

{

GetItem();

}

* 1. **Xem/Sửa thông tin tài khoản.**

Sinh viên được xem thông tin của mình, đổi mật khẩu đăng nhập:

SinhVien currentSV;

SinhVienBL svBL = SinhVienBL.getInstance();

public ThongTinSVForm(string user)

{

currentSV = svBL.GetByMSSV(int.Parse(user));

InitializeComponent();

}

private void btnDoiMK\_Click(object sender, EventArgs e)

{

DoiMKForm frm = new DoiMKForm(currentSV.MSSV.ToString());

frm.ShowDialog();

}

private void ThongTínVForm\_Load(object sender, EventArgs e)

{

lbHT.Text = currentSV.HoLot + " " + currentSV.Ten;

lbMSSV.Text = currentSV.MSSV.ToString();

lbLop.Text = currentSV.TenLop;

lbKhoa.Text = currentSV.Khoa;

}

Form Đổi mật khẩu:

string userName;

TaiKhoanBL tkBL = TaiKhoanBL.getInstance();

public DoiMKForm(string userName)

{

this.userName = userName;

InitializeComponent();

}

private void Validation()

{

if (tkBL.isCorrectPass(txtMK.Text))

{

if (txtNewMK.Text == txtReNewMK.Text)

{

tkBL.UpdatePass(userName, txtNewMK.Text);

MessageBox.Show("Đổi mật khẩu thành công");

this.Close();

}

else

MessageBox.Show("Mật khẩu nhập lại không khớp");

}

else

MessageBox.Show("Sai mật khẩu");

}

private void btnXacNhan\_Click(object sender, EventArgs e)

{

Validation();

}

private void btnHuy\_Click(object sender, EventArgs e)

{

this.Close();

}

private void cbShowPass\_CheckedChanged(object sender, EventArgs e)

{

if(cbShowPass.Checked)

txtMK.UseSystemPasswordChar = false;

else

txtMK.UseSystemPasswordChar = true;

}

private void cbShowNewPas\_CheckedChanged(object sender, EventArgs e)

{

if (cbShowNewPas.Checked)

txtNewMK.UseSystemPasswordChar = false;

else

txtNewMK.UseSystemPasswordChar = true;

}

private void cbShowRePass\_CheckedChanged(object sender, EventArgs e)

{

if (cbShowRePass.Checked)

txtReNewMK.UseSystemPasswordChar = false;

else

txtReNewMK.UseSystemPasswordChar = true;

}

private void DoiMKForm\_Load(object sender, EventArgs e)

{

txtMK.UseSystemPasswordChar = true;

txtNewMK.UseSystemPasswordChar = true;

txtReNewMK.UseSystemPasswordChar = true;

}

1. **Chức năng thuộc quyền Quản lý:**
   1. **Quản lý đăng ký học phần.**

Giáo vụ có thể xem danh sách chi tiết đăng ký học phần của sinh viên. Các chức năng Lọc theo năm học, học kỳ, khoa, lớp, khoá, tìm theo từ khoá, thống kế, xuất chi tiết đăng ký.

Mã nguồn:

#region

/// <summary>

/// Hàm xuất thông tin học phần đăng ký

/// </summary>

private void LoadItem()

{

lvQL.Items.Clear();

SqlDataReader reader = ctdkBL.GetChiTietDKHP();

while (reader.Read())

{

ListViewItem item = lvQL.Items.Add(reader["MSSV"].ToString());

item.SubItems.Add(reader["MaHP"].ToString());

item.SubItems.Add(reader["HoLot"].ToString());

item.SubItems.Add(reader["Ten"].ToString());

item.SubItems.Add(reader["TenHP"].ToString());

item.SubItems.Add(reader["TenLop"].ToString());

item.SubItems.Add(reader["LoaiHP"].ToString());

item.SubItems.Add(reader["Khoa"].ToString());

item.SubItems.Add(reader["STC"].ToString());

item.SubItems.Add(reader["SLDK"].ToString());

if (!years.Contains(reader["NamHoc"].ToString()))

AddYears(reader["NamHoc"].ToString());

if (!courses.Contains(reader["HocKy"].ToString()))

AddCourses(reader["HocKy"].ToString());

}

}

/// <summary>

/// Lọc danh sách đăng ký học phần theo các tiêu chí

/// </summary>

/// <param name="hocky"></param>

/// <param name="nam"></param>

/// <param name="khoa"></param>

/// <param name="lop"></param>

private void LoadItemWithCourse(int hocky, string nam)

{

lvQL.Items.Clear();

SqlDataReader reader = ctdkBL.GetChiTietDKHPTheoHocKy(hocky, nam);

while (reader.Read())

{

ListViewItem item = lvQL.Items.Add(reader["MSSV"].ToString());

item.SubItems.Add(reader["MaHP"].ToString());

item.SubItems.Add(reader["HoLot"].ToString());

item.SubItems.Add(reader["Ten"].ToString());

item.SubItems.Add(reader["TenHP"].ToString());

item.SubItems.Add(reader["TenLop"].ToString());

item.SubItems.Add(reader["LoaiHP"].ToString());

item.SubItems.Add(reader["Khoa"].ToString());

item.SubItems.Add(reader["STC"].ToString());

item.SubItems.Add(reader["SLDK"].ToString());

}

}

/// <summary>

/// Hàm tìm kiếm

/// </summary>

/// <param name="key"></param>

private void SearchByKey(string key)

{

var items = this.lvQL.Items.Cast<ListViewItem>()

.Where(x => (

x.SubItems[3].Text.Contains(key) ||

x.SubItems[4].Text.Contains(key) ||

x.SubItems[5].Text.Contains(key) ||

x.SubItems[7].Text.Contains(key)))

.ToList();

lvQL.Items.Clear();

foreach (var ct in items)

{

ListViewItem item = lvQL.Items.Add(ct.SubItems[0].Text);

item.SubItems.Add(ct.SubItems[1].Text);

item.SubItems.Add(ct.SubItems[2].Text);

item.SubItems.Add(ct.SubItems[3].Text);

item.SubItems.Add(ct.SubItems[4].Text);

item.SubItems.Add(ct.SubItems[5].Text);

item.SubItems.Add(ct.SubItems[6].Text);

item.SubItems.Add(ct.SubItems[7].Text);

item.SubItems.Add(ct.SubItems[8].Text);

item.SubItems.Add(ct.SubItems[9].Text);

}

}

/// <summary>

/// Lấy danh sách 6 khoá còn học trong năm nay

/// </summary>

/// <returns></returns>

private List<int> GetCourse()

{

List<int> result = new List<int>();

int i = 0;

int yearOld = 1976;

while (i < 6)

{

result.Add(DateTime.Now.Year - yearOld);

yearOld++;

i++;

}

return result;

}

/// <summary>

/// Xuất danh sách theo khoa

/// </summary>

/// <param name="khoa"></param>

private void GetByKhoa(string khoa)

{

LoadItem();

var items = this.lvQL.Items.Cast<ListViewItem>()

.Where(x => (

x.SubItems[7].Text.Equals(khoa.Trim())

))

.ToList();

lvQL.Items.Clear();

foreach (var ct in items)

{

ListViewItem item = lvQL.Items.Add(ct.SubItems[0].Text);

item.SubItems.Add(ct.SubItems[1].Text);

item.SubItems.Add(ct.SubItems[2].Text);

item.SubItems.Add(ct.SubItems[3].Text);

item.SubItems.Add(ct.SubItems[4].Text);

item.SubItems.Add(ct.SubItems[5].Text);

item.SubItems.Add(ct.SubItems[6].Text);

item.SubItems.Add(ct.SubItems[7].Text);

item.SubItems.Add(ct.SubItems[8].Text);

item.SubItems.Add(ct.SubItems[9].Text);

}

}

/// <summary>

/// Xuất danh sách theo lớp

/// </summary>

/// <param name="lop"></param>

private void GetByLop(string lop)

{

LoadItem();

var items = this.lvQL.Items.Cast<ListViewItem>()

.Where(x => (

x.SubItems[5].Text.Equals(lop.Trim())

))

.ToList();

lvQL.Items.Clear();

foreach (var ct in items)

{

ListViewItem item = lvQL.Items.Add(ct.SubItems[0].Text);

item.SubItems.Add(ct.SubItems[1].Text);

item.SubItems.Add(ct.SubItems[2].Text);

item.SubItems.Add(ct.SubItems[3].Text);

item.SubItems.Add(ct.SubItems[4].Text);

item.SubItems.Add(ct.SubItems[5].Text);

item.SubItems.Add(ct.SubItems[6].Text);

item.SubItems.Add(ct.SubItems[7].Text);

item.SubItems.Add(ct.SubItems[8].Text);

item.SubItems.Add(ct.SubItems[9].Text);

}

}

/// <summary>

/// Xuất danh sách theo khoá học

/// </summary>

/// <param name="course"></param>

private void GetByCourse(int course)

{

LoadItem();

var items = this.lvQL.Items.Cast<ListViewItem>()

.Where(x => (

(2000 + int.Parse(x.SubItems[0].Text.Substring(0, 2)) - 1976).Equals(course)

))

.ToList();

lvQL.Items.Clear();

foreach (var ct in items)

{

ListViewItem item = lvQL.Items.Add(ct.SubItems[0].Text);

item.SubItems.Add(ct.SubItems[1].Text);

item.SubItems.Add(ct.SubItems[2].Text);

item.SubItems.Add(ct.SubItems[3].Text);

item.SubItems.Add(ct.SubItems[4].Text);

item.SubItems.Add(ct.SubItems[5].Text);

item.SubItems.Add(ct.SubItems[6].Text);

item.SubItems.Add(ct.SubItems[7].Text);

item.SubItems.Add(ct.SubItems[8].Text);

item.SubItems.Add(ct.SubItems[9].Text);

}

}

/// <summary>

/// Lấy năm học

/// </summary>

/// <param name="year"></param>

private void AddYears(string year)

{

years.Add(year);

}

/// <summary>

/// Lấy học kỳ

/// </summary>

/// <param name="course"></param>

private void AddCourses(string course)

{

courses.Add(course);

}

/// <summary>

/// Xuất kêt quả đăng ký

/// </summary>

/// <param name="lv"></param>

/// <param name="listKQDK"></param>

/// <param name="path"></param>

private void WriteToExcel(ListView lv, string path)

{

ExcelPackage.LicenseContext = OfficeOpenXml.LicenseContext.NonCommercial;

using (ExcelPackage p = new ExcelPackage())

{

p.Workbook.Worksheets.Add("sheet 1");

ExcelWorksheet ws = p.Workbook.Worksheets[0];

ws.Name = "sheet 1";

ws.Cells.Style.Font.Size = 11;

ws.Cells.Style.Font.Name = "Calibri";

List<string> arrCollumHeader = new List<string>();

foreach (ColumnHeader item in lv.Columns)

{

arrCollumHeader.Add(item.Text);

}

var countColHeader = arrCollumHeader.Count();

int colIndex = 1;

int rowIndex = 1;

//Tao cac Header

foreach (var item in arrCollumHeader)

{

var cell = ws.Cells[rowIndex, colIndex];

var style = cell.Style.Font;

style.Bold = true;

cell.Value = item;

colIndex++;

}

//Lay danh sach sinh vien

for (int i = 0; i < lvQL.Items.Count; i++)

{

ListViewItem item = lvQL.Items[i];

colIndex = 1;

rowIndex++;

ws.Cells[rowIndex, colIndex++].Value = item.SubItems[0].Text;

ws.Cells[rowIndex, colIndex++].Value = item.SubItems[1].Text;

ws.Cells[rowIndex, colIndex++].Value = item.SubItems[2].Text;

ws.Cells[rowIndex, colIndex++].Value = item.SubItems[3].Text;

ws.Cells[rowIndex, colIndex++].Value = item.SubItems[4].Text;

ws.Cells[rowIndex, colIndex++].Value = item.SubItems[5].Text;

ws.Cells[rowIndex, colIndex++].Value = item.SubItems[6].Text;

ws.Cells[rowIndex, colIndex++].Value = item.SubItems[7].Text;

ws.Cells[rowIndex, colIndex++].Value = item.SubItems[8].Text;

ws.Cells[rowIndex, colIndex++].Value = item.SubItems[9].Text;

}

//save file

Byte[] bin = p.GetAsByteArray();

File.WriteAllBytes(path, bin);

}

}

#endregion

private void btnXuat\_Click(object sender, EventArgs e)

{

saveFileDialog1.FileName = string.Format("Danh Sach Dang Ky Hoc Phan Hoc ky {0} Nam hoc {1}", cbbHK.Text, DateTime.Now.Year.ToString() + " - " + (DateTime.Now.Year + 1).ToString());

saveFileDialog1.InitialDirectory = @"E:\";

saveFileDialog1.DefaultExt = "xlsx";

saveFileDialog1.Filter = "Excel 2007 files(xlsx) (\*.xlsx)|\*.xlsx";

DialogResult dlg = saveFileDialog1.ShowDialog();

if (dlg == DialogResult.OK)

{

string path = string.Format(@"{0}", saveFileDialog1.FileName);

WriteToExcel(lvQL, path);

MessageBox.Show("Xuất phiếu đăng ký thành công!");

}

}

private void btnFilter\_Click(object sender, EventArgs e)

{

int hocKy = int.Parse(cbbHK.Text == "" ? "0" : cbbHK.Text);

string nam = cbbNamHoc.Text;

LoadItemWithCourse(hocKy, nam);

}

private void btnSearch\_Click(object sender, EventArgs e)

{

if (txtSearch.Text == "")

LoadItem();

SearchByKey(txtSearch.Text);

}

private void txtSearch\_TextChanged(object sender, EventArgs e)

{

if (txtSearch.Text == "")

LoadItem();

}

private void cbbKhoa\_TextChanged(object sender, EventArgs e)

{

LoadcomboboxLop(cbbKhoa.Text);

GetByKhoa(cbbKhoa.Text);

}

private void cbbLop\_TextChanged(object sender, EventArgs e)

{

GetByLop(cbbLop.Text);

}

private void cbbCourse\_TextChanged(object sender, EventArgs e)

{

GetByCourse(int.Parse(cbbCourse.Text));

}

private void btnThongKe\_Click(object sender, EventArgs e)

{

SqlDataReader reader = LichThiBL.Instance().LoadLichThi(cbbNamHoc.Text, int.Parse(cbbHK.Text));

ThongKeFrm frm = new ThongKeFrm();

frm.LoadChart(reader, cbbNamHoc.Text, cbbHK.Text);

frm.ShowDialog();

}

--HẾT--