**Lab3.1 Thiết Kế MDI Application**

**Yêu cầu:**

- Kết Nối CSDL với SQL Server, tạo CSDL với file script *QNHASACH.sql*

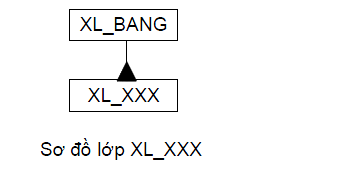
- Thiết kế frmMain, frmDangNhap, frmDoiMatKhau

- Sử dụng các control: Menustrip , Ribbon contro (Devexpress), TabControl

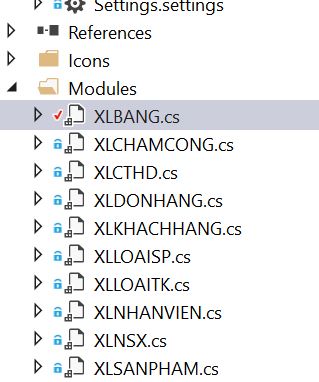
- Xử lý xự kiện cho các nút lệnh.

- Chia sẽ quản lý source code với github cho các thành viên trong nhóm (nhóm 2 người), nộp link github.

**Yêu cầu 1: Cài đặt các class xử lý dữ liệu, kế thừa từ Data Table theo cấu trúc sau:**



**- Tạo folder Modules chứa các class**

****

**- Class XL\_BANG gồm fields, constructors, methods, event xử lý dữ liệu chung cho tất cả các bảng.**

**Code:**

class XLBANG: DataTable

{

#region Fields

public static string cnnStr = "Data Source=.;Initial Catalog=QLNHASACH;Integrated Security=True";

private static SqlConnection \_cnn;

private SqlDataAdapter \_da = new SqlDataAdapter();

private string \_query; // Cau truy van

private string \_name; // Ten table

#endregion

#region Properties

public string query

{

get { return \_query; }

set { \_query = value; }

}

public string name

{

get { return \_name; }

set { \_name = value; }

}

public int getCount // Lay so record

{

get { return this.DefaultView.Count; }

}

#endregion

#region Consructors

public XLBANG() : base() { }

public XLBANG(string pName)

{

\_name = pName;

docBang();

}

public XLBANG(string pName, string pQuery)

{

\_name = pName;

\_query = pQuery;

docBang();

}

#endregion

#region Methods

//Doc du lieu

public void docBang()

{

if (\_query == null)

\_query = "Select \* from " + \_name;

if (\_cnn == null)

\_cnn = new SqlConnection(cnnStr);

try

{

\_da = new SqlDataAdapter(\_query, \_cnn);

\_da.FillSchema(this, SchemaType.Mapped);

\_da.Fill(this);

\_da.RowUpdated += new SqlRowUpdatedEventHandler(da\_RowUpdated);

SqlCommandBuilder cmmb = new SqlCommandBuilder(\_da);

}catch(SqlException ex)

{

throw ex;

}

}

//Ghi du lieu

public Boolean ghi()

{

Boolean result = true;

try

{

\_da.Update(this);

this.AcceptChanges();

}catch(SqlException ex)

{

this.RejectChanges();

result = false;

throw ex;

}

return result;

}

//Loc du lieu

public void locDuLieu(string pDieuKien)

{

try

{

this.DefaultView.RowFilter = pDieuKien;

}catch(SqlException ex)

{

throw ex;

}

}

//Thực hiện câu truy vấn cập nhật dữ liệu (Insert, Update, Delete)

public static int Thuc\_hien\_lenh(String pQuery)

{

if (\_cnn == null)

\_cnn = new SqlConnection(cnnStr);

try

{

SqlCommand cmd = new SqlCommand(pQuery, \_cnn);

\_cnn.Open();

int result = cmd.ExecuteNonQuery();

\_cnn.Close();

return result;

}

catch

{

return -1;

}

}

//Thực hiện câu truy vấn trả về một giá trị

public static Object Thuc\_hien\_lenh\_tinh\_toan(String pQuery)

{

if (\_cnn == null)

\_cnn = new SqlConnection(cnnStr);

try

{

SqlCommand cmd = new SqlCommand(pQuery, \_cnn);

\_cnn.Open();

Object result = cmd.ExecuteScalar();

\_cnn.Close();

return result;

}

catch

{

return null;

}

}

//Goi Store Procedure

public static Object Thuc\_hien\_StoreProcedure(String nameProc, SqlParameter[] p )

{

if (\_cnn == null)

\_cnn = new SqlConnection(cnnStr);

try

{

SqlCommand cmd = new SqlCommand();

cmd.CommandType = CommandType.StoredProcedure;

cmd.CommandText = nameProc;

//Add Parameters

cmd.Parameters.AddRange(p);

\_cnn.Open();

Object result = cmd.ExecuteScalar();

\_cnn.Close();

return result;

}

catch

{

return null;

}

}

#endregion

#region Events

//Xử lý sự kiện Rowupdated đối với trường khóa chính có kiểu Autonumber

private void da\_RowUpdated(Object sender, SqlRowUpdatedEventArgs e)

{

if (this.PrimaryKey[0].AutoIncrement)

{

if ((e.Status == UpdateStatus.Continue) && (e.StatementType == StatementType.Insert))

{

SqlCommand cmd = new SqlCommand("Select @@IDENTITY ", \_cnn);

e.Row.ItemArray[0] = cmd.ExecuteScalar();

e.Row.AcceptChanges();

}

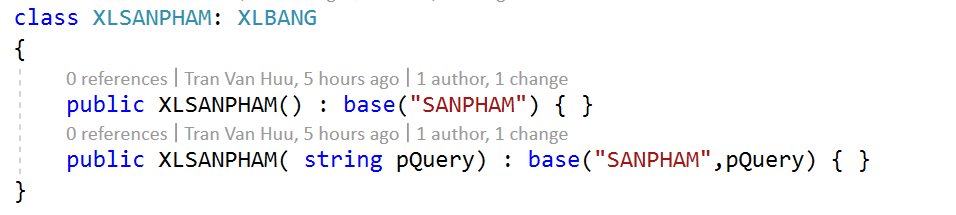
}

}

#endregion

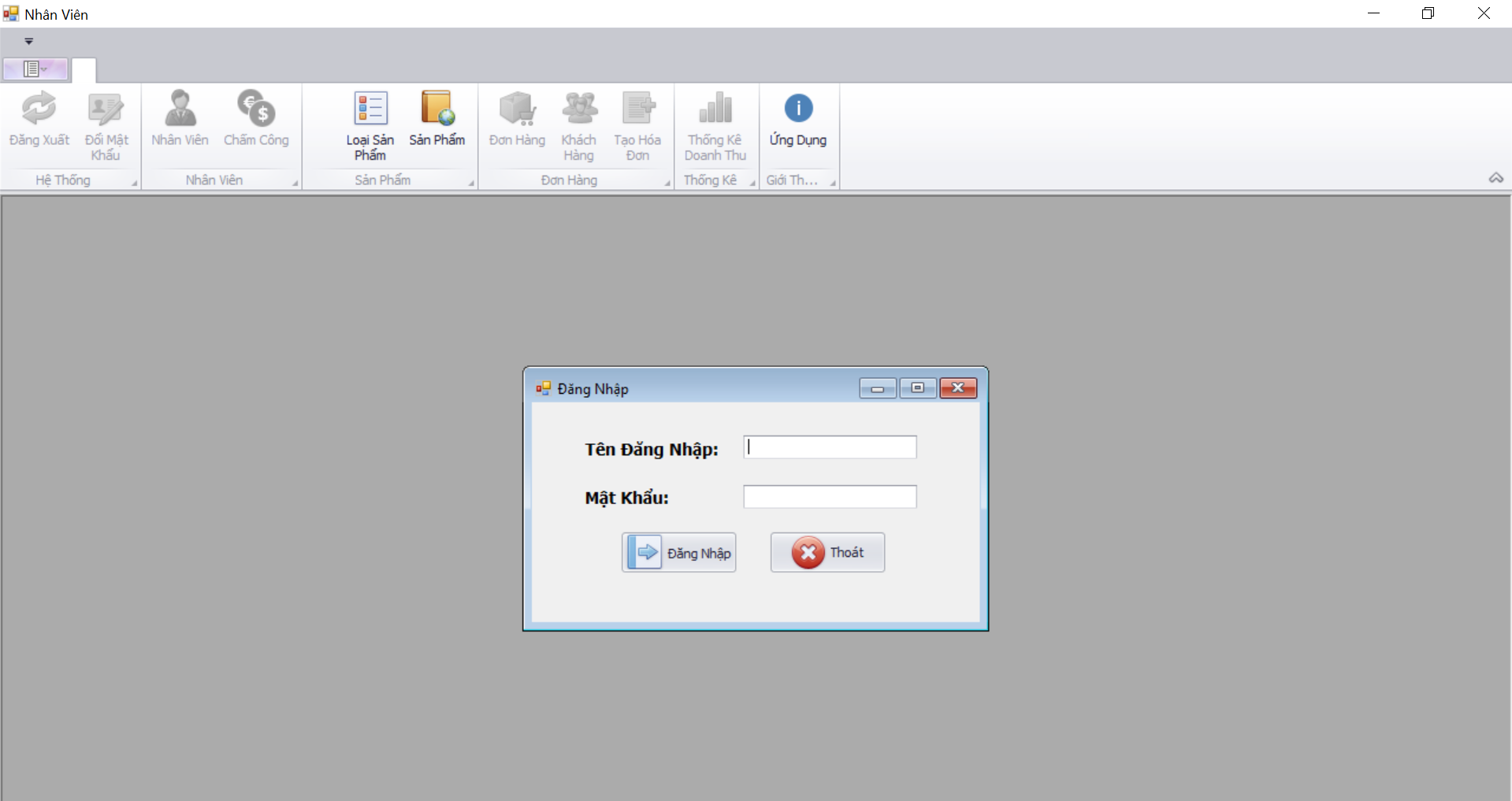
}

**- Class XL\_XXX gồm các fields, constructors, methos cho từng bảng riêng kế thừa từ XL\_BANG**

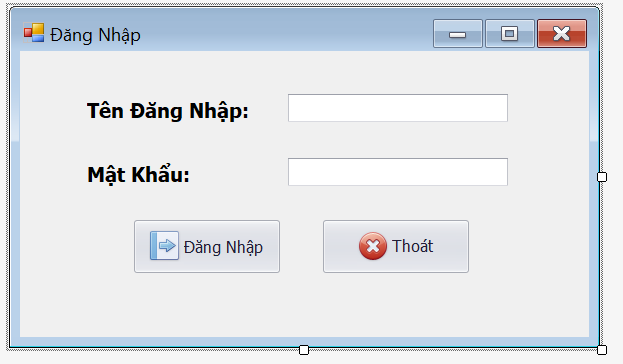


**Xử lý tương tự cho các bảng còn lại.**

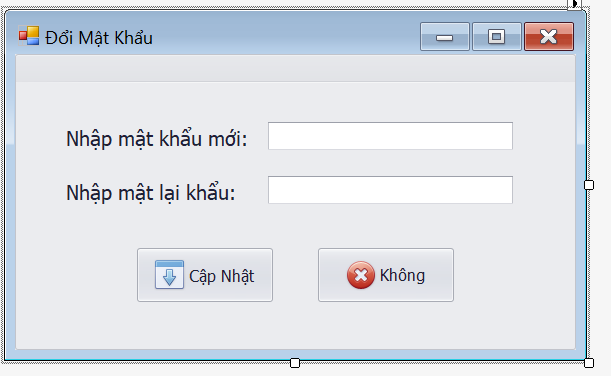
**Yêu cầu 2: Thiết kế màn hình chính frmMain:**



**Yêu cầu 3: Thiết kế màn hình chính frmDangNhap:**



**Yêu cầu 4: Thiết kế màn hình chính frmDoiMatKhau:**



**Yêu cầu 5: Xử lý code cho các form, có phân quyền.**