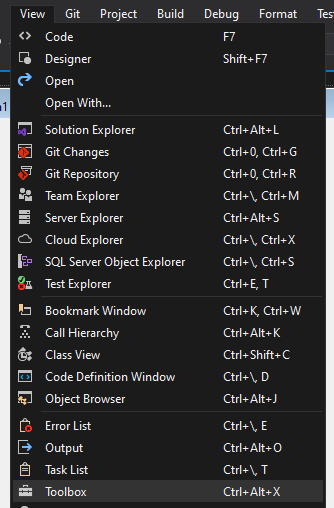
LAB1

1) Tạo project

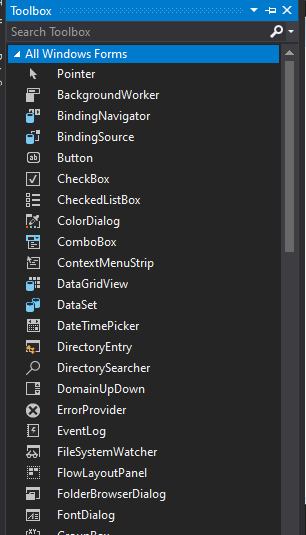
Mở Visual Studio → Create a new project → chọn Windows Forms App (.NET Framework)

1. Thiết kế Form bằng Designer

Trên tab View mở ToolBox:



Trong All Windows Forms tiến hành kéo thả các control sau lên Form1 và đặt các Name tương ứng (dùng khi tham chiếu trong code):

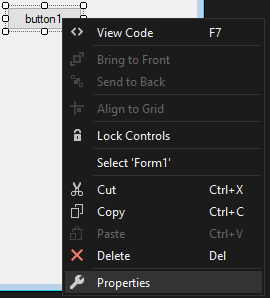


PictureBox (góc trên trái) — để logo. (Name không bắt buộc)

Label lớn ở trên (Tiêu đề) — lblTitle, Text = "Quán ăn nhanh Hưng Thịnh".

Trong groupBoxMenu tạo nhiều Button cho từng món — ví dụ btnComChien, btnBanhMy, btnCoca, btnLipton, …

Chuột phải vào box button chọn Properties:



Trong Properties → Tag gõ giá trị menu cho tất cả các Button món (dùng để tự gắn event handler chung).

Text của mỗi button là tên món (vd: Cơm chiên trứng, Bánh mỳ ốp la, Coca, Lipton, Ốc rang muối, Khoai tây chiên, 7 up, Cam, Mỳ xào hải sản, Cá viên chiên, Pepsi, Cafe, Burger bò nướng, Đùi gà rán, Bún bò Huế).

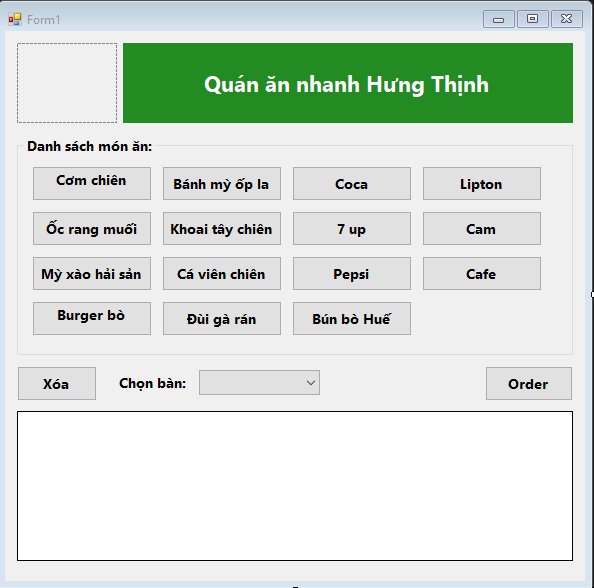
Label nhỏ: "Danh sách món ăn:"

Button btnDelete (Text = Xóa) — xóa món được chọn hoặc xóa tất cả.

ComboBox comboBoxTables — danh sách chọn bàn.

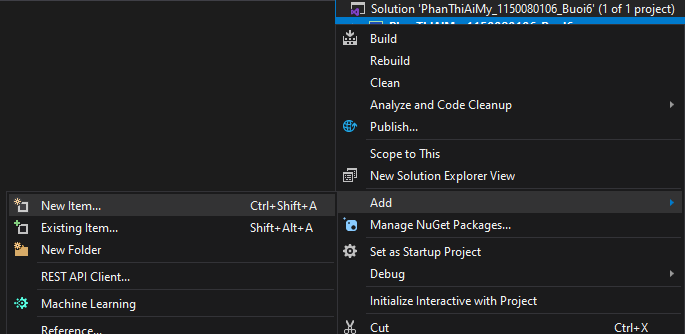
Button btnOrder (Text = Order) — ghi đơn xuống tệp.

DataGridView dgvOrders — hiển thị danh sách món và số lượng (đặt dưới cùng).

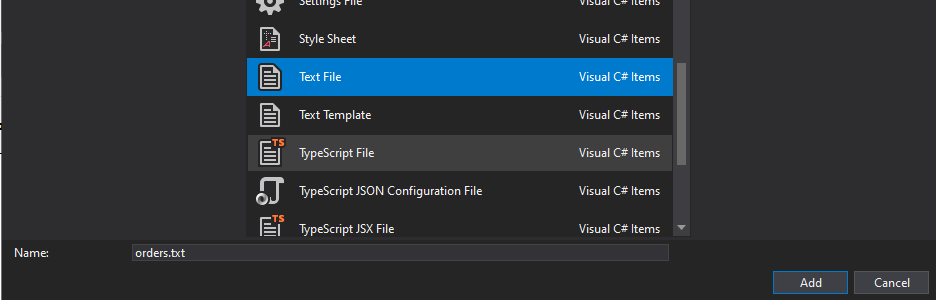


1. Lập trình (Form1.cs) — mã nguồn chính

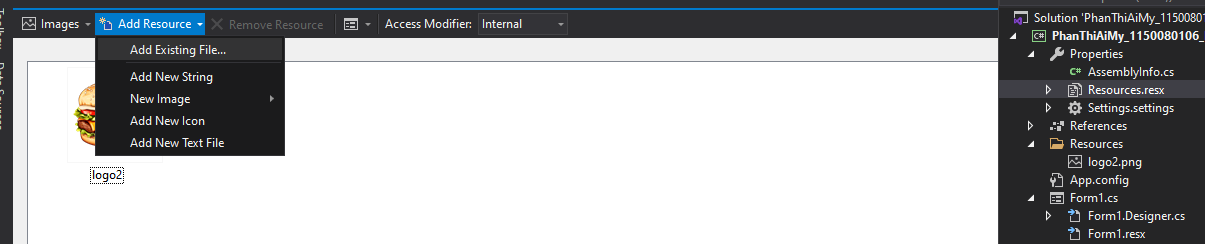
Tại Project -> Chuột phải chọn Add -> Chọn New Item



Tiếp tục chọn Add một TextFile và đặt tên là orders.txt:



Vào Properties -> Chọn Resource.resx -> Trong Add Resource chọn Add Existing File -> add ảnh logo



Trong Form1.cs:

using System;

using System.Collections.Generic;

using System.Drawing;

using System.IO;

using System.Text;

using System.Windows.Forms;

namespace PhanThiAiMy\_1150080106\_Buoi6

{

public partial class Form1 : Form

{

// Lưu tạm đơn hiện tại: key = tên món, value = số lượng

private Dictionary<string, int> orderItems = new Dictionary<string, int>(StringComparer.OrdinalIgnoreCase);

public Form1()

{

InitializeComponent();

this.Load += Form1\_Load;

}

private void Form1\_Load(object sender, EventArgs e)

{

pictureBoxLogo.Image = Properties.Resources.logo2;

// 1) Khởi tạo danh sách bàn (ví dụ Bàn 1..10)

comboBoxTables.Items.Clear();

for (int i = 1; i <= 10; i++)

comboBoxTables.Items.Add("Bàn " + i);

// 2) Thiết lập DataGridView

dgvOrders.Columns.Clear();

dgvOrders.Columns.Add("colDish", "Món");

dgvOrders.Columns.Add("colQty", "Số lượng");

dgvOrders.Columns["colQty"].Width = 80;

dgvOrders.ReadOnly = true;

dgvOrders.SelectionMode = DataGridViewSelectionMode.FullRowSelect;

dgvOrders.AllowUserToAddRows = false;

// 3) Tự động gán event Click cho tất cả Button trong groupBoxMenu có Tag = "menu"

foreach (Control c in groupBoxMenu.Controls)

{

if (c is Button btn && (btn.Tag?.ToString() ?? "") == "menu")

{

btn.Click += MenuButton\_Click;

}

}

// Tùy chọn: gán event cho btnDelete, btnOrder

btnDelete.Click += BtnDelete\_Click;

btnOrder.Click += BtnOrder\_Click;

// (Tùy) double-click trên dgv để giảm 1 món

dgvOrders.DoubleClick += DgvOrders\_DoubleClick;

}

// Khi bấm 1 button món ăn -> tăng số lượng lên 1

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

{

if (sender is Button btn)

{

string dish = btn.Text.Trim();

AddToOrder(dish, 1);

RefreshGrid();

}

}

// Thêm / tăng số lượng món

private void AddToOrder(string dish, int qty)

{

if (string.IsNullOrWhiteSpace(dish) || qty <= 0) return;

if (orderItems.ContainsKey(dish))

orderItems[dish] += qty;

else

orderItems[dish] = qty;

}

// Cập nhật DataGridView từ orderItems

private void RefreshGrid()

{

dgvOrders.Rows.Clear();

foreach (var kv in orderItems)

{

dgvOrders.Rows.Add(kv.Key, kv.Value);

}

}

// Nút Xóa: nếu có hàng chọn thì xóa món đó, nếu không có hàng chọn -> hỏi xóa tất cả

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

{

if (dgvOrders.SelectedRows.Count > 0)

{

var r = dgvOrders.SelectedRows[0];

string dish = r.Cells[0].Value?.ToString();

if (!string.IsNullOrEmpty(dish) && orderItems.ContainsKey(dish))

{

orderItems.Remove(dish);

RefreshGrid();

}

}

else

{

var dr = MessageBox.Show("Bạn có muốn xóa tất cả món trong đơn?", "Xóa tất cả", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (dr == DialogResult.Yes)

{

orderItems.Clear();

RefreshGrid();

}

}

}

// Nút Order: ghi đơn ra tệp văn bản (append)

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

{

if (comboBoxTables.SelectedItem == null)

{

MessageBox.Show("Vui lòng chọn bàn trước khi Order.", "Thiếu thông tin", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

if (orderItems.Count == 0)

{

MessageBox.Show("Danh sách món rỗng.", "Không có món", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

// Nội dung order

StringBuilder sb = new StringBuilder();

sb.AppendLine("============= NEW ORDER =============");

sb.AppendLine($"Thời gian: {DateTime.Now:yyyy-MM-dd HH:mm:ss}");

sb.AppendLine($"Bàn: {comboBoxTables.SelectedItem}");

sb.AppendLine("-------------------------------------");

sb.AppendLine("Món\tSố lượng");

foreach (var kv in orderItems)

{

sb.AppendLine($"{kv.Key}\t{kv.Value}");

}

sb.AppendLine("-------------------------------------");

sb.AppendLine();

// Đường dẫn file — đặt trong thư mục chạy ứng dụng

string filePath = Path.Combine(Application.StartupPath, "orders.txt");

try

{

File.AppendAllText(filePath, sb.ToString(), Encoding.UTF8);

MessageBox.Show($"Đã ghi order xuống file:\n{filePath}", "Ghi file thành công", MessageBoxButtons.OK, MessageBoxIcon.Information);

// Sau khi order xong -> xóa đơn hiện tại (tùy chọn)

orderItems.Clear();

RefreshGrid();

comboBoxTables.SelectedIndex = -1;

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi ghi file: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

// Double-click 1 hàng trên dgv -> giảm 1 số lượng (nếu giảm về 0 thì xóa)

private void DgvOrders\_DoubleClick(object sender, EventArgs e)

{

if (dgvOrders.SelectedRows.Count == 0) return;

string dish = dgvOrders.SelectedRows[0].Cells[0].Value?.ToString();

if (string.IsNullOrEmpty(dish)) return;

if (orderItems.ContainsKey(dish))

{

orderItems[dish]--;

if (orderItems[dish] <= 0) orderItems.Remove(dish);

RefreshGrid();

}

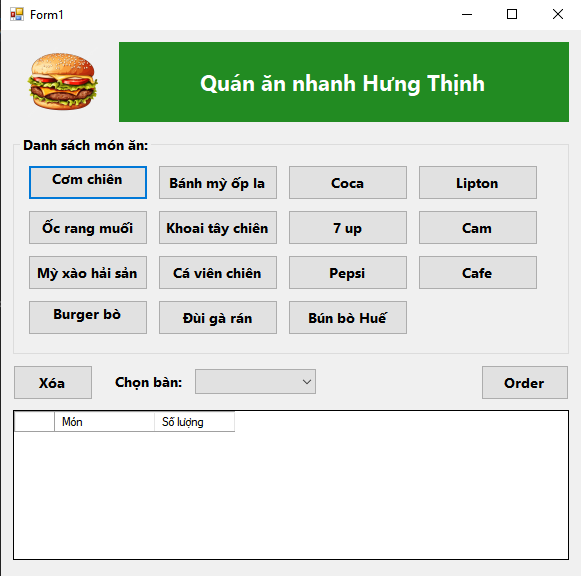
}

}

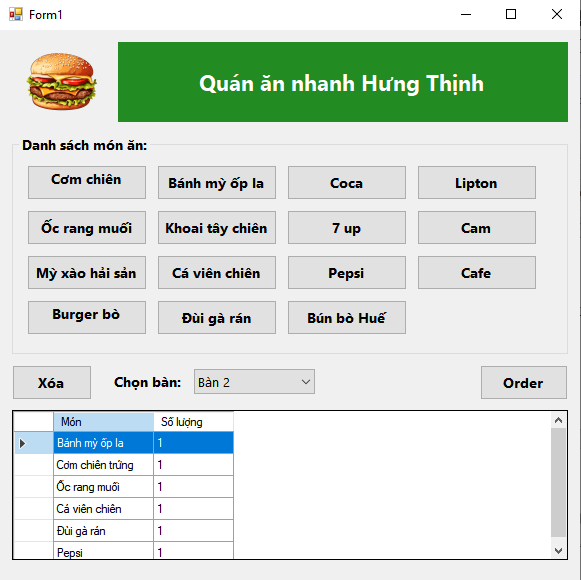
}

4) Thử nghiệm

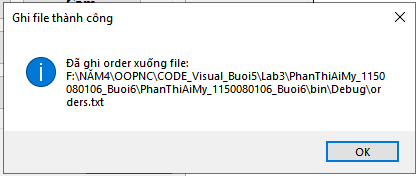
Chạy ứng dụng (F5).

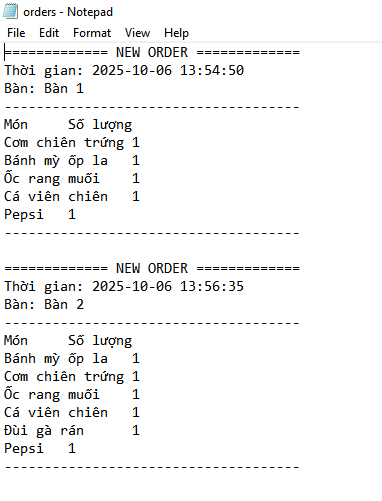


Đặt món và nhấn Order:



Sau khi nhấn Order:





Thực hành 2:

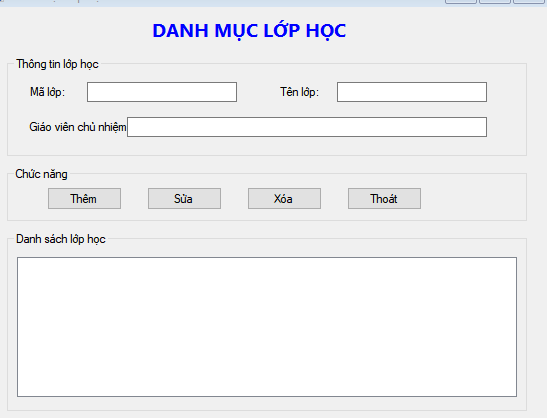
1) Tạo project

Mở Visual Studio → Create a new project → chọn Windows Forms App (.NET Framework)

1. Thiết kế Form bằng Designer

Trên tab View mở ToolBox và vào Properties thiết kế như sau:

|  |  |  |
| --- | --- | --- |
| **Loại Control** | **Name** | **Text hiển thị** |
| Label | lblTitle | **DANH MỤC SINH VIÊN** |
| Label | lblHoTen | Họ tên: |
| Label | lblNgaySinh | Ngày sinh: |
| Label | lblLop | Lớp: |
| Label | lblDiaChi | Địa chỉ: |
| TextBox | txtHoTen | (trống) |
| TextBox | txtLop | (trống) |
| TextBox | txtDiaChi | (trống) |
| DateTimePicker | dtpNgaySinh | (mặc định) |
| Button | btnThem | Thêm |
| Button | btnSua | Sửa |
| Button | btnXoa | Xóa |
| Button | btnThoat | Thoát |
| ListView | lvSinhVien | (để hiển thị danh sách sinh viên) |



1. Lập trình (Form1.cs) — mã nguồn chính

using System;

using System.Windows.Forms;

namespace PhanThiAiMy\_1150080106\_Buoi6

{

public partial class Form2 : Form

{

public Form2()

{

InitializeComponent();

lvLop.View = View.Details;

lvLop.FullRowSelect = true;

lvLop.GridLines = true;

lvLop.Columns.Add("Mã lớp", 100);

lvLop.Columns.Add("Tên lớp", 150);

lvLop.Columns.Add("Giáo viên CN", 150);

}

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

{

if (string.IsNullOrWhiteSpace(txtMaLop.Text) || string.IsNullOrWhiteSpace(txtTenLop.Text))

{

MessageBox.Show("Mã lớp và tên lớp không được để trống!", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

ListViewItem item = new ListViewItem(txtMaLop.Text);

item.SubItems.Add(txtTenLop.Text);

item.SubItems.Add(txtGV.Text);

lvLop.Items.Add(item);

ClearInput();

}

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

{

if (lvLop.SelectedItems.Count == 0)

{

MessageBox.Show("Vui lòng chọn dòng cần xóa!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

return;

}

lvLop.Items.Remove(lvLop.SelectedItems[0]);

}

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

{

if (lvLop.SelectedItems.Count == 0)

{

MessageBox.Show("Vui lòng chọn dòng cần sửa!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

return;

}

ListViewItem item = lvLop.SelectedItems[0];

item.SubItems[0].Text = txtMaLop.Text;

item.SubItems[1].Text = txtTenLop.Text;

item.SubItems[2].Text = txtGV.Text;

}

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

{

this.Close();

}

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

{

if (lvLop.SelectedItems.Count > 0)

{

ListViewItem item = lvLop.SelectedItems[0];

txtMaLop.Text = item.SubItems[0].Text;

txtTenLop.Text = item.SubItems[1].Text;

txtGV.Text = item.SubItems[2].Text;

}

}

private void ClearInput()

{

txtMaLop.Clear();

txtTenLop.Clear();

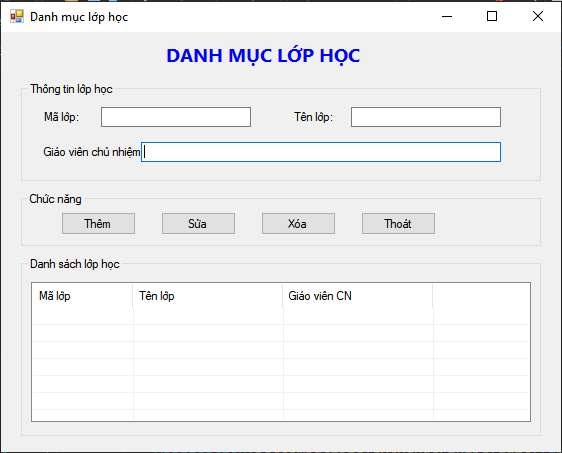
txtGV.Clear();

}

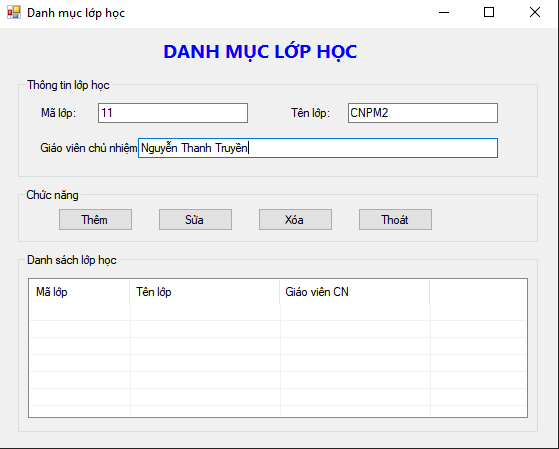
}

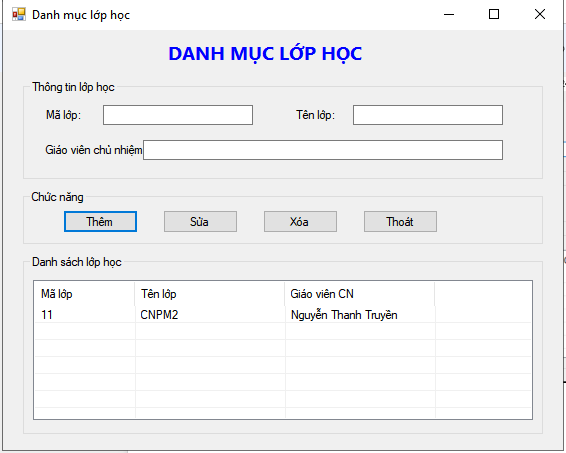
}

1. Chạy thử nghiệm

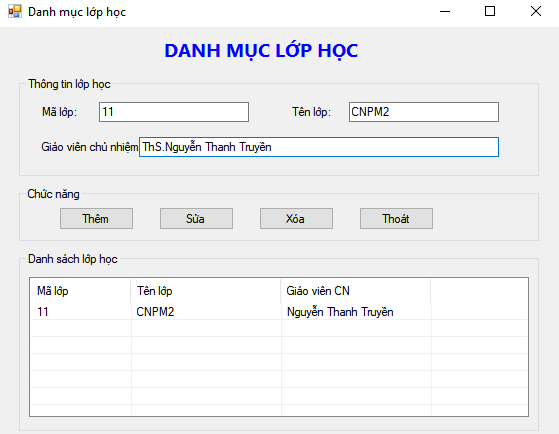


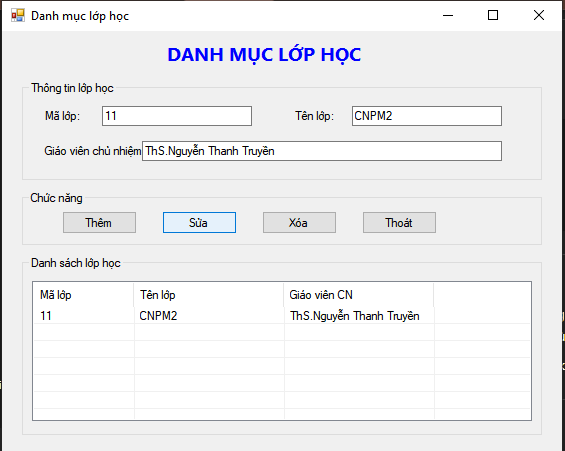
Tính năng thêm:



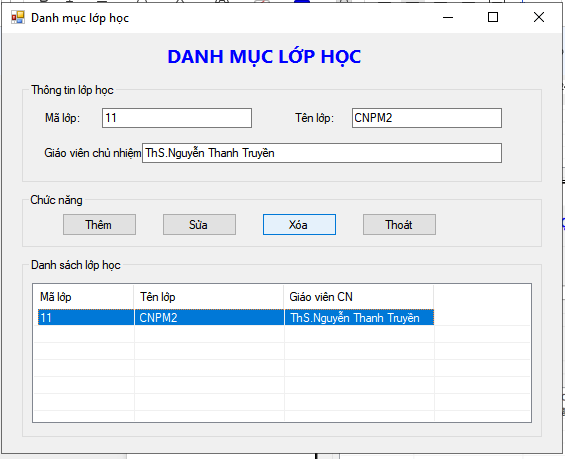


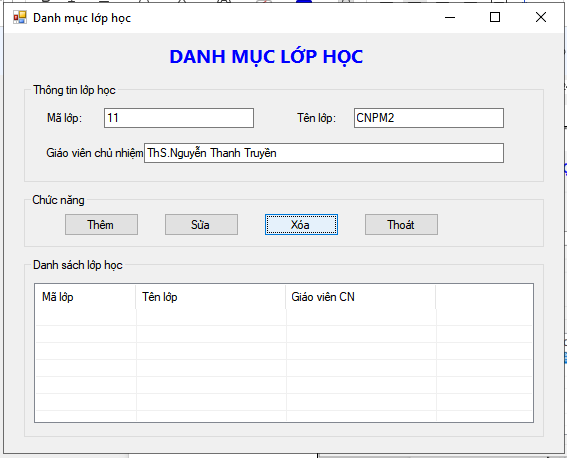
Tính năng sửa:





Tính năng xoá:





LAB4

Thực hành 1: Tạo kết nối giữa Winform và CSDL.

