Họ tên: Phạm Gia Bảo

Lớp: 11\_DH\_CNPM1

MSSV: 1150080043

LAB 3

**Thực hành 1:**

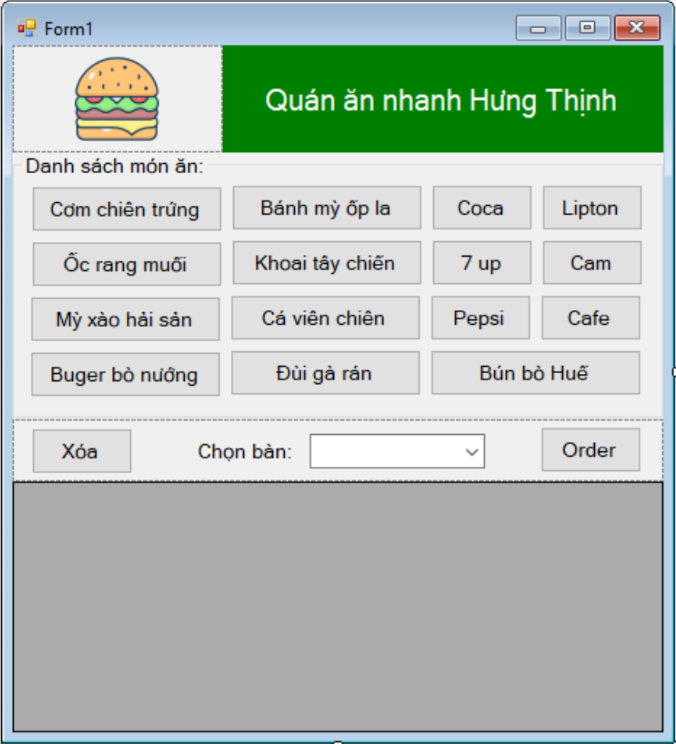
Bước 1: Mở visual studio 2022 và chọn Windows Forms App (.Net Framework)

Bước 2: Nhấn toolbox sau đây: Button, Label, PictureBox, ComboBox, DataGridView, GroupBox

Đặt tên như sau: BtnComchientrung, BtnOcrangmuoi, BtnMyxaohaisan,…

BtnXoa, btnOrder,…

Bước 3: Làm design theo yêu cầu



Bước 4: Code Form1.cs

using System;

using System.Data;

using System.Diagnostics;

using System.IO;

using System.Text;

using System.Windows.Forms;

namespace Lab3\_TH1

{

public partial class Form1 : Form

{

private DataTable dtOrder;

public Form1()

{

InitializeComponent();

SetupGrid();

SeedTables();

WireFoodButtons();

btnXoa.Click += btnXoa\_Click;

btnOrder.Click += btnOrder\_Click;

}

private void SetupGrid()

{

dtOrder = new DataTable();

dtOrder.Columns.Add("Món", typeof(string));

dtOrder.Columns.Add("Số lượng", typeof(int));

dgv1.AutoGenerateColumns = true;

dgv1.DataSource = dtOrder;

dgv1.AllowUserToAddRows = false;

dgv1.SelectionMode = DataGridViewSelectionMode.FullRowSelect;

dgv1.MultiSelect = false;

dgv1.ReadOnly = true;

}

private void SeedTables()

{

if (cboBan.Items.Count == 0)

{

cboBan.Items.AddRange(new object[] { "Bàn 1", "Bàn 2", "Bàn 3", "Bàn 4" });

}

if (cboBan.SelectedIndex < 0 && cboBan.Items.Count > 0)

cboBan.SelectedIndex = 0;

}

private void WireFoodButtons()

{

var foodButtons = new Button[]

{

btnComchientrung, btnOcrangmuoi, btnMyxaohaisan,

btnBanhmyopla, btnKhoaitaychien, btnCavienchien,

btnBugerbonuong, btnDuiga, btnBunbohue,

btnCoca, btnPepsi, btn7up, btnLipton, btnCam, btnCafe

};

foreach (var b in foodButtons)

b.Click += FoodButton\_Click;

}

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

{

if (sender is Button btn)

{

string mon = btn.Text.Trim();

AddOrIncrement(mon);

}

}

private void AddOrIncrement(string mon)

{

string safe = mon.Replace("'", "''");

var rows = dtOrder.Select($"[Món] = '{safe}'");

if (rows.Length > 0)

{

int sl = (int)rows[0]["Số lượng"];

rows[0]["Số lượng"] = sl + 1;

}

else

{

dtOrder.Rows.Add(mon, 1);

}

}

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

{

if (dgv1.SelectedRows.Count > 0)

{

foreach (DataGridViewRow r in dgv1.SelectedRows)

if (!r.IsNewRow) dgv1.Rows.Remove(r);

return;

}

if (dgv1.CurrentRow != null && !dgv1.CurrentRow.IsNewRow)

{

dgv1.Rows.Remove(dgv1.CurrentRow);

return;

}

MessageBox.Show("Hãy chọn một dòng rồi bấm Xóa.");

}

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

{

if (cboBan.SelectedItem == null || string.IsNullOrWhiteSpace(cboBan.Text))

{

MessageBox.Show("Hãy chọn tên bàn trước khi Order.");

return;

}

if (dtOrder.Rows.Count == 0)

{

MessageBox.Show("Chưa có món nào được chọn.");

return;

}

try

{

string ban = cboBan.Text.Trim();

var sb = new StringBuilder();

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

sb.AppendLine($"Bàn: {ban}");

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

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

int tongSoMon = 0;

foreach (DataRow row in dtOrder.Rows)

{

string tenMon = row["Món"].ToString();

int soLuong = (int)row["Số lượng"];

tongSoMon += soLuong;

sb.AppendLine($"{tenMon.PadRight(25)} x {soLuong}");

}

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

sb.AppendLine($"Tổng số lượng: {tongSoMon}");

MessageBox.Show(sb.ToString(), "Xác nhận Order",

MessageBoxButtons.OK, MessageBoxIcon.Information);

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

using (var sw = new StreamWriter(path, append: true, Encoding.UTF8))

{

sw.WriteLine(sb.ToString());

sw.WriteLine();

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi ghi tệp: " + ex.Message);

}

}

private void groupBox1\_Enter(object sender, EventArgs e) { }

private void pictureBox1\_Click(object sender, EventArgs e) { }

private void label1\_Click(object sender, EventArgs e) { }

private void button16\_Click(object sender, EventArgs e) { }

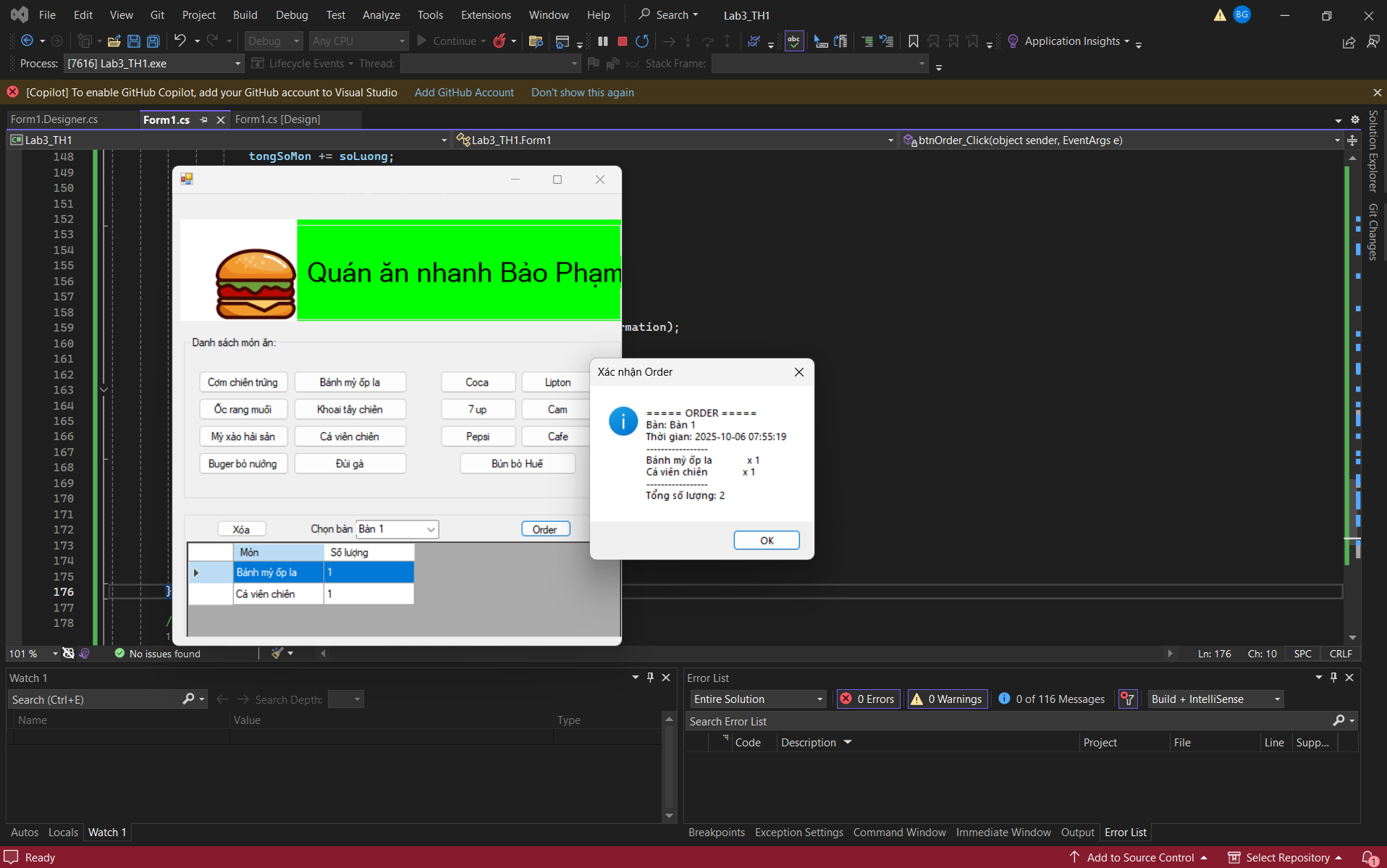
private void listBox1\_SelectedIndexChanged(object sender, EventArgs e) { }

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e) { }

}

}

Kết quả:



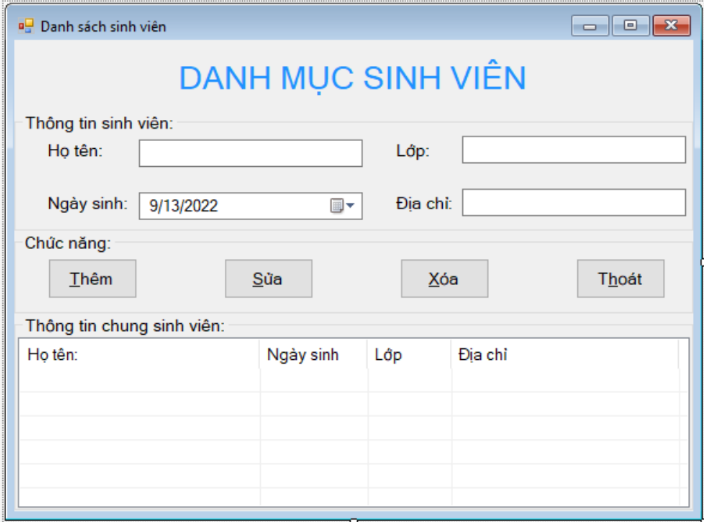
**Thực hành 2:**

Bước 1: Mở visual studio 2022 và chọn Windows Forms App (.Net Framework)

Bước 2: Nhấn toolbox sau đây: Button, TextBox, DateTimePicker, Label, ListView, GroupBox

Đặt tên như sau: BtnThem, BtnSua, BtnXoa, BthThoat, txtHoten, txtLop, txtDiaChi, LvSinhvien

Bước 3: Làm design theo yêu cầu



Bước 4: Code trong form1.cs

using System;

using System.Globalization;

using System.Windows.Forms;

namespace Lab3\_TH2

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

SetupListView();

btnThem.Click += btnThem\_Click;

btnXoa.Click += btnXoa\_Click;

btnSua.Click += bthSua\_Click;

btnThoat.Click += btnThoat\_Click;

LvSinhvien.SelectedIndexChanged += lvSinhVien\_SelectedIndexChanged;

}

private void SetupListView()

{

LvSinhvien.View = View.Details;

LvSinhvien.FullRowSelect = true;

LvSinhvien.GridLines = true;

LvSinhvien.HideSelection = false;

LvSinhvien.Columns.Clear();

LvSinhvien.Columns.Add("Họ tên", 180);

LvSinhvien.Columns.Add("Ngày sinh", 110);

LvSinhvien.Columns.Add("Lớp", 100);

LvSinhvien.Columns.Add("Địa chỉ", 220);

}

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

{

string hoten = txtHoten.Text.Trim();

if (string.IsNullOrEmpty(hoten))

{

MessageBox.Show("Họ tên không được rỗng.");

txtHoten.Focus();

return;

}

string ngay = date.Value.ToString("dd/MM/yyyy");

string lop = txtLop.Text.Trim();

string diachi = txtDiachi.Text.Trim();

var item = new ListViewItem(new[] { hoten, ngay, lop, diachi });

LvSinhvien.Items.Add(item);

ClearInputs();

}

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

{

if (LvSinhvien.SelectedItems.Count == 0)

{

MessageBox.Show("Hãy chọn một dòng trên danh sách để xóa.");

return;

}

while (LvSinhvien.SelectedItems.Count > 0)

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

}

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

{

if (LvSinhvien.SelectedItems.Count == 0)

{

MessageBox.Show("Hãy chọn một dòng để sửa.");

return;

}

string hoten = txtHoten.Text.Trim();

if (string.IsNullOrEmpty(hoten))

{

MessageBox.Show("Họ tên không được rỗng.");

txtHoten.Focus();

return;

}

var it = LvSinhvien.SelectedItems[0];

it.SubItems[0].Text = hoten;

it.SubItems[1].Text = date.Value.ToString("dd/MM/yyyy");

it.SubItems[2].Text = txtLop.Text.Trim();

it.SubItems[3].Text = txtDiachi.Text.Trim();

}

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

{

Close();

}

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

{

if (LvSinhvien.SelectedItems.Count == 0) return;

var it = LvSinhvien.SelectedItems[0];

txtHoten.Text = it.SubItems[0].Text;

if (DateTime.TryParseExact(

it.SubItems[1].Text, "dd/MM/yyyy",

CultureInfo.InvariantCulture, DateTimeStyles.None, out var d))

{

date.Value = d;

}

txtLop.Text = it.SubItems[2].Text;

txtDiachi.Text = it.SubItems[3].Text;

}

private void ClearInputs()

{

txtHoten.Clear();

txtLop.Clear();

txtDiachi.Clear();

date.Value = DateTime.Today;

txtHoten.Focus();

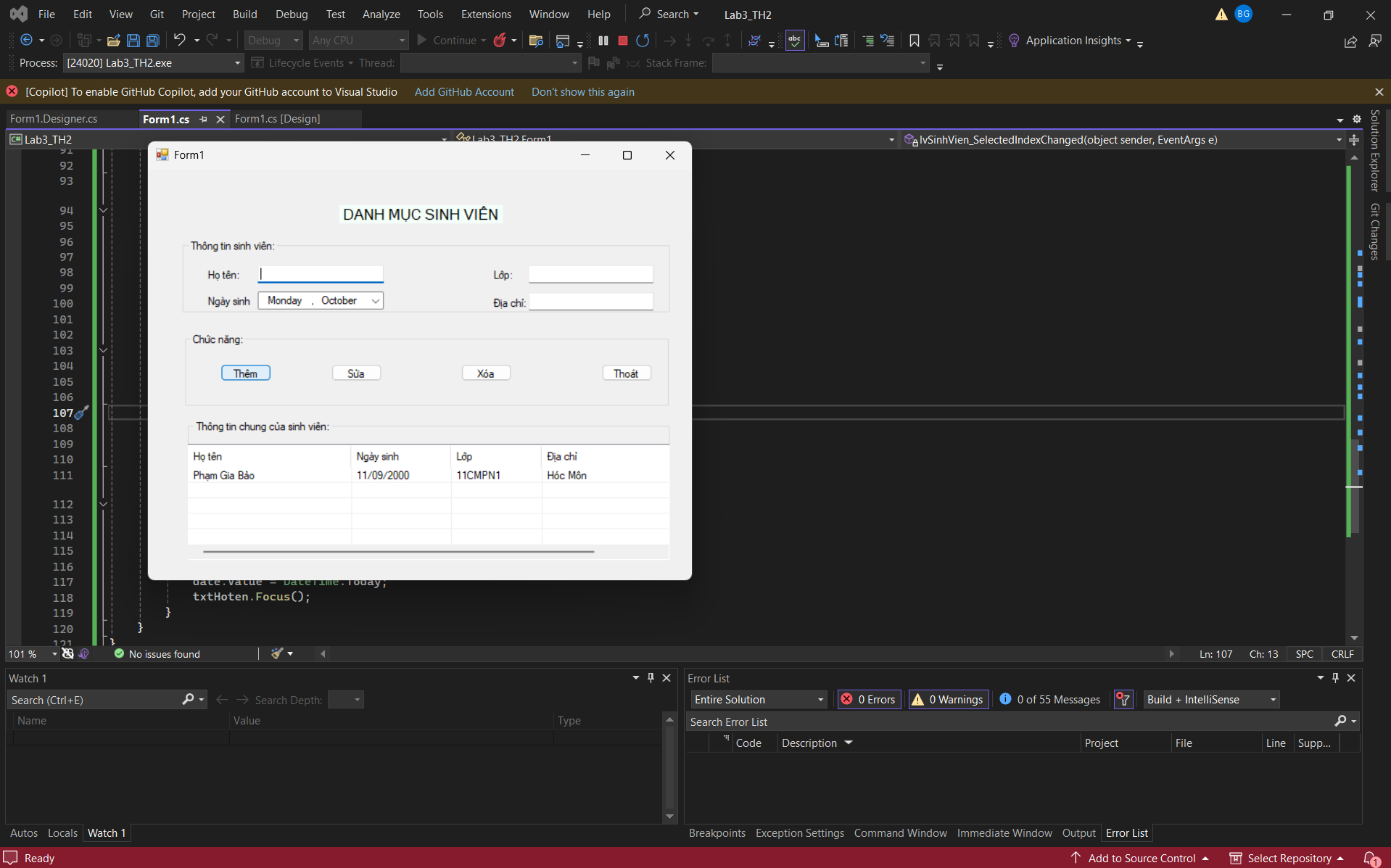
}

}

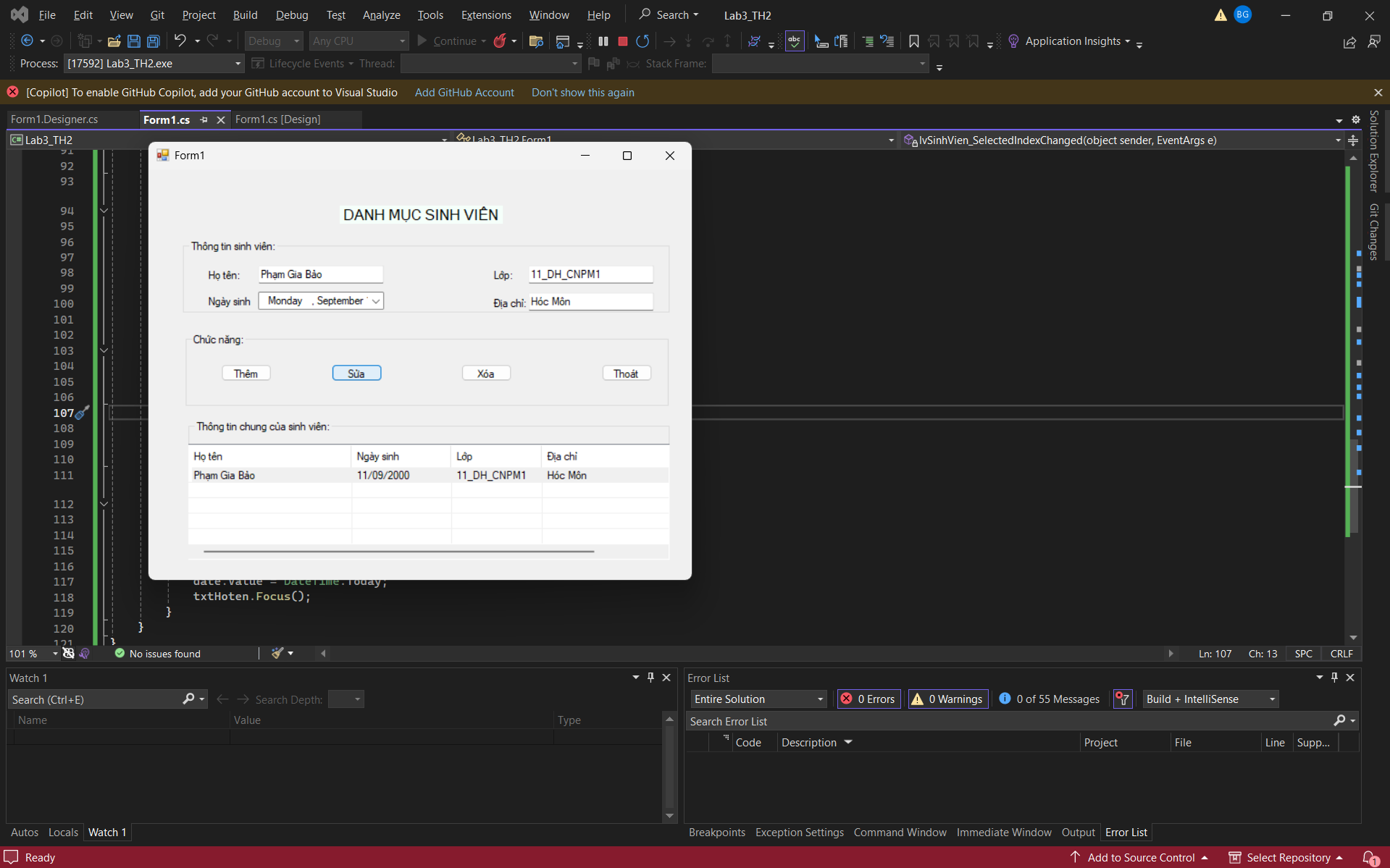
}

Kết quả:

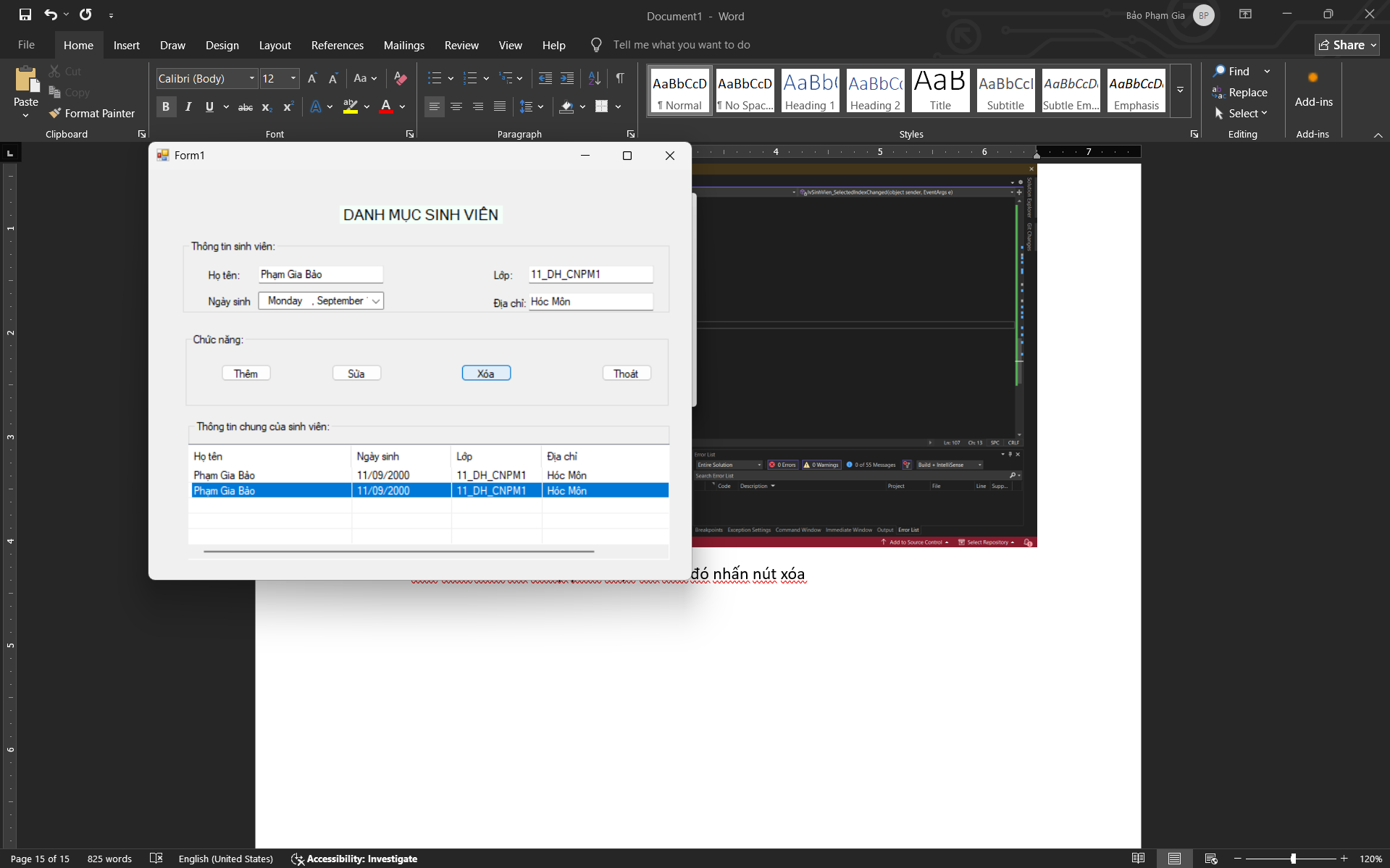
* Thêm sinh viên

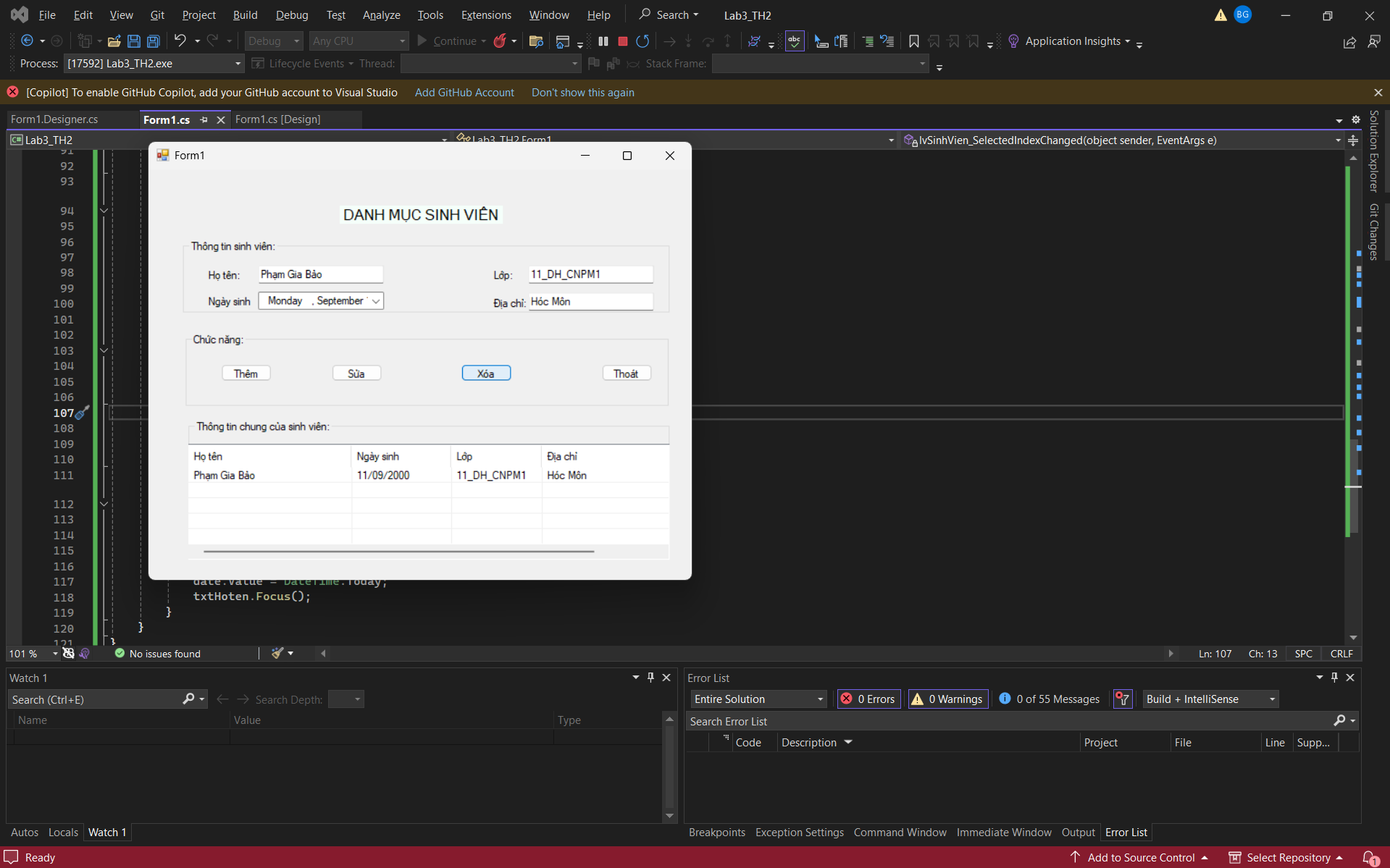


* Sửa sinh viên thì nhấp phải mục đó và thay đổi thông tin sau đó nhấn nút sửa

****

* Xóa sinh viên thì nhấp phải mục đó sau đó nhấn nút xóa

****



LAB 4

