**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**



**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**NGÀNH CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN**

**CẤU TRÚC DỮ LIỆU & GIẢI THUẬT**

**Đề tài: Danh sách liên kết và chương trình danh sách sinh viên bằng c#**

**SVTH : NGUYỄN LÊ MINH VŨ**

**MSSV : 16110272**

**SVTH : TRẦN PHƯƠNG NAM**

**MSSV : 16110163**

**TP. Hồ Chí Minh, tháng 12 năm 2017**

**MỤC LỤC**

NỘI DUNG…………………………………………………………………………………………………………………………….. 1

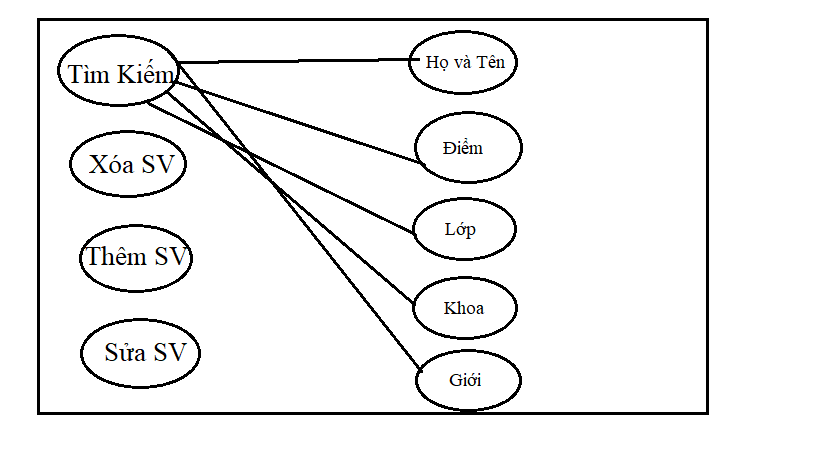
1. Mô tả Project………………………………………………………………………………………………………………………… 1
   1. Chương trình dùng làm gì………………………………………………………………………………………………. 1
   2. Use case diagram của chương trình……………………………………………………………………………….. 1
   3. Bảng mô tả ý nghĩa của usr case……………………………………………………………………………………… 1
2. Mô tả quá trình làm………………………………………………………………………………………………………………. 2
   1. Thiết kế giao diện……………………………………………………………………………………………………………. 2
      1. Bảng mô tả thiết kế…………………………………………………………………………………………….. 3
      2. Nội dung thiết kế………………………………………………………………………………………………… 3
3. Màn hình đăng nhập…………………………………………………………………………………….. 3
4. Màn hình chính…………………………………………………………………………………………….. 3
   1. Thiết kế lớp……………………………………………………………………………………………………………………… 4
      1. Class diagram………………………………………………………………………………………………………. 4
      2. Class chính…………………………………………………………………………………………………………… 5
5. Class Node…………………………………………………………………………………………………….. 5
6. Class Program……………………………………………………………………………………………….. 12
   * 1. Sequence diagram………………………………………………………………………………………………. .14
   1. Thiết kế các Code chính……………………………………………………………………………………………………. 15
7. Code bảng đăng nhập………………………………………………………………………………………………… 15
8. Code bảng datagridview…………………………………………………………………………………………….. 15
9. Code thêm sinh viên………………………………………………………………………………………………..... 16
10. Code sửa sinh viên…………………………………………………………………………………………………….. 16
11. Code lưu sinh viên……………………………………………………………………………………………………… 17
12. Code loadform, unlockcontrol và lockcontrol…………………………………………………………….. 18
13. Code kiểm tra textbox……………………………………………………………………………………………….. 20
14. Code xóa sinh viên…………………………………………………………………………………………………….. 22
15. Code filter………………………………………………………………………………………………………………….. 23
    1. Cài đặt……………………………………………………………………………………………………………………………… 31
16. Mô tả công việc………………………………………………………………………………………………………………………. 31
17. Kết luận………………………………………………………………………………………………………………………………….. 32
18. Kết quả…………………………………………………………………………………………………………………………….. 32
19. Khó khắn………………………………………………………………………………………………………………………….. 32
20. Ý tưởng……………………………………………………………………………………………………………………………. 33
21. Ưu điểm…………………………………………………………………………………………………………………………… 33
22. Nhược điểm……………………………………………………………………………………………………………………… 33

**Nội dung**

1. **Mô tả Project**
   1. **Chương trình dùng làm gì?**

Trong các trường học nói chung và từng lớp học nói riêng, việc sắp xếp và tìm kiếm các thông tin của từng sinh viên hoặc học sinh sao cho hợp lí, dễ hiểu và tìm kiếm được nhanh chóng nhất. Vì thế chương trình này sẽ giúp người thực hiện những điều đó.

* 1. **Use case diagram của chương trình**



* 1. **Bảng mô tả ý nghĩa của use case**

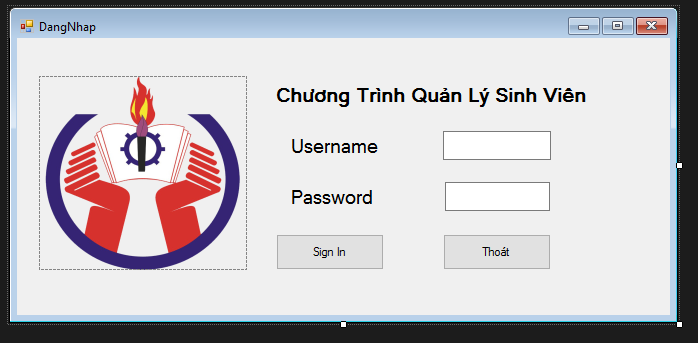
|  |  |
| --- | --- |
| Tên bảng | Chức Năng |
| Giao diện chính | * Tìm kiếm SV * Chỉnh Sửa SV * Xóa SV * Thêm SV |
| Bảng thêm | * Tên sinh viên * MSSV * Lớp * Khoa * Giới * Điểm môn 1 * Điểm môn 2 * Điểm môn 3 * Điểm môn 4 |
| Bảng xóa | * Tên sinh viên * MSSV * Lớp * Khoa * Giới * Điểm môn 1 * Điểm môn 2 * Điểm môn 3 * Điểm môn 4 |
| Bảng tìm kiếm | * Tìm theo tên SV * Tìm theo MSSV * Tìm theo giới |

1. **Mô tả quá trình làm**
   1. **Thiết kế giao diện**
      1. **Bảng mô tả thiết kế**

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên Class** | **Công dụng** |
| 1 | Danh sách Liên kết đơn | Dùng để lưu trữ, liên kết các Node lại với nhau tạo thành một danh sách liên kết đơn |
| 2 | Node | Dùng để lưu trữ các đối tượng từ File |

* + 1. **Nội dung thiết kế**

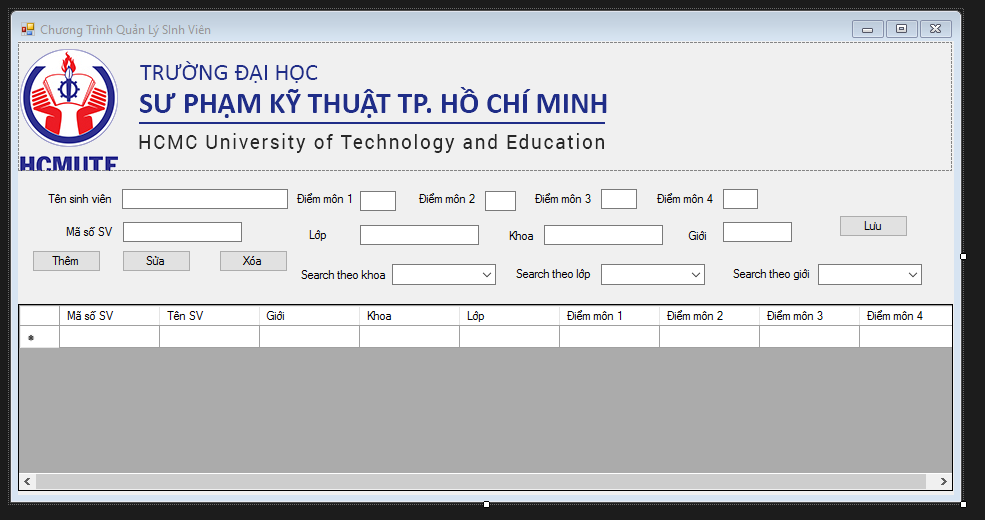
1. **Màn hình đăng nhập**



* Yêu cầu người dùng đăng nhập để sử dụng
* Nhập Username và Password từ bàn phím

Thực hiện chức năng đăng nhập cho người quản lý khi cần đăng nhập vào form quản lý để thực hiện thêm, sửa, xóa sinh viên.

1. **Màn hình chính**



* Ba Combobox để tìm kiếm sinh viên theo từng yếu tố
* Chín Textbox để Thêm, Xóa hoặc sửa SV
* Chín Textbox để hiển thị
  1. **Thiết kế lớp**
     1. **Class diagram**

|  |
| --- |
| Sinh Viên |
| +MSSV: Text  +HoVaTen: Text  +NgaySinh: Datetime  +DienThoai: Text  +NoiSinh: Text |
| +Nhap ( )  +Sua ( )  +Xoa ( )  +TimKiem ( ) |

* + 1. **Class chính**

1. **Class Node**

public class SV

{

public string maSV { get; set; }

public string hoTen { get; set; }

public double diemNhapMon { get; set; }

public double diemToanA1 { get; set; }

public double diemToanA2 { get; set; }

public double diemVatLy { get; set; }

public double diemAnhVan { get; set; }

public double diemTB { get; set; }

public string xepLoai { get; set; }

public SV(string maSV, string hoTen, double diemNhapMon, double diemToanA1, double diemToanA2,

double diemVatLy, double diemAnhVan)

{

this.maSV = maSV;

this.hoTen = hoTen;

this.diemAnhVan = diemAnhVan;

this.diemNhapMon = diemNhapMon;

this.diemToanA1 = diemToanA1;

this.diemToanA2 = diemToanA2;

this.diemVatLy = diemVatLy;

this.diemTB = TinhDienTB();

this.xepLoai = XepLoai();

}

public double TinhDienTB()

{

return (diemAnhVan + diemNhapMon + diemToanA1 + diemToanA2 + diemVatLy) / 5;

}

public string XepLoai()

{

if (diemTB >= 8.0)

return "Gioi";

else if (diemTB >= 7.0)

return "Kha";

else if (diemTB >= 5.0)

return "TB";

else return "Yeu";

}

}

public class Node

{

// Data

public SV data { get; set; }

public Node nextLink { get; set; }

// Constructor

public Node(SV data)

{

this.data = data;

}

1. }Class DSLK

public class LinkedList

{

// Data

private Node head;

private Node tail;

//Constructor

public LinkedList()

{

head = tail = null;

}

// Methods

// Kiê?m tra rô~ng

public bool IsEmpty

{

get

{

return head == null;

}

}

#region "Insert Methods"

// Thêm phâ`n tu? va`o dâ`u

public void InsertFirst(SV data)

{

Node node = new Node(data);

if (head == null)

head = tail = node;

else

{

node.nextLink = head;

head = node;

}

}

// Thêm phâ`n tu? va`o cuô´i

public void InsertLast(SV data)

{

Node node = new Node(data);

if (head == null)

head = tail = node;

else

{

tail.nextLink = node;

tail = node;

}

}

#endregion

#region "Remove Methods"

// Xo´a phâ`n tu? dâ`u

public void RemoveFirst()

{

if (head == null)

return;

else

{

Node tam = head;

head = tam.nextLink;

tam = null;

GC.Collect(); // Gia?i pho´ng vu`ng nho´

}

}

// Xo´a phâ`n tu? cuô´i

public void RemoveLast()

{

if (head == null)

return;

else

{

Node tam = tail;

tail.nextLink = null;

tam = null;

GC.Collect();

}

}

// Xo´a mô?t node co´ gia´ tri? bâ´t ki`

public void RemoveNode(string value)

{

if (head.data.maSV == value)

{

RemoveFirst();

return;

}

Node p = head;

int t = 0;

while (p != null)

{

t++;

if (p.data.maSV == value)

{

int h = 0;

for (Node q = head; q != p; q = q.nextLink)

{

h++;

if (h == t - 1)

{

q.nextLink = p.nextLink;

p = null;

GC.Collect();

return;

}

}

//Node a = p.previousLink, b = p.nextLink;

//a.nextLink = b;

//b.previousLink = a;

//p.nextLink = p.previousLink = null;

//p = null;

//GC.Collect();

//return;

}

p = p.nextLink;

}

}

#endregion

// Sort

public void InterchangeSort()

{

for (Node p = head; p != tail; p = p.nextLink)

for (Node q = p.nextLink; q != null; q = q.nextLink)

if (int.Parse(p.data.maSV) > int.Parse(q.data.maSV))

{

SV t = p.data;

p.data = q.data;

q.data = t;

}

}

// Print List

public void PrintList(ref int n)

{

Node node = head;

int i = 0;

while (node != null)

{

Console.Write("Ma so sinh vien: {0}\n", node.data.maSV);

Console.Write("Ho va ten: {0}\n", node.data.hoTen);

Console.Write("Diem Anh Van: {0}\n", node.data.diemAnhVan);

Console.Write("Diem Nhap Mon: {0}\n", node.data.diemNhapMon);

Console.Write("Diem Toan A1: {0}\n", node.data.diemToanA1);

Console.Write("Diem Toan A2: {0}\n", node.data.diemToanA2);

Console.Write("Diem Vat Ly: {0}\n", node.data.diemVatLy);

Console.Write("Diem trung binh: {0}\n", node.data.diemTB);

Console.Write("Xep loai: {0}\n", node.data.xepLoai);

Console.WriteLine("==========================================");

i++;

if (i >= n) break;

node = node.nextLink;

}

}

}

1. **Class Program**

class Program

{

public static LinkedList listSV = new LinkedList();

//Tao Danh Sach SV

public static void taoListSV()

{

listSV.InsertLast(new SV("1", "SV A", 6.7, 7.6, 3.4, 5.6, 6.6));

listSV.InsertLast(new SV("2", "SV B", 7.4, 3.7, 5.2, 3.5, 6.4));

listSV.InsertLast(new SV("3", "SV C", 7.1, 6.7, 4.3, 7.5, 8.4));

listSV.InsertLast(new SV("4", "SV D", 5.6, 5.7, 5.3, 4.5, 7.4));

listSV.InsertLast(new SV("5", "SV E", 9.1, 8.7, 8.3, 9.5, 8.4));

listSV.InsertLast(new SV("6", "SV F", 6.1, 8.7, 5.3, 6.5, 5.4));

listSV.InsertLast(new SV("7", "SV G", 6.2, 8.2, 5.2, 6.2, 5.2));

listSV.InsertLast(new SV("8", "SV H", 6.3, 8.3, 5.3, 6.3, 5.3));

listSV.InsertLast(new SV("9", "SV I", 6.4, 8.4, 5.4, 6.4, 5.4));

listSV.InsertLast(new SV("10", "SV J", 6.5, 8.5, 5.5, 6.5, 5.5));

listSV.InsertLast(new SV("11", "SV K", 6.5, 8.5, 5.5, 6.5, 5.6));

listSV.InsertLast(new SV("12", "SV L", 6, 8, 5, 6, 5));

listSV.InsertLast(new SV("13", "SV M", 6.6, 8.6, 5.6, 6.6, 5.46));

listSV.InsertLast(new SV("14", "SV N", 6.9, 8.9, 5.9, 6.9, 5.9));

listSV.InsertLast(new SV("15", "SV O", 6.1, 8.7, 5.3, 8.0, 5.9));

listSV.InsertLast(new SV("16", "SV P", 6.5, 8.7, 5.3, 6.5, 5.4));

listSV.InsertLast(new SV("17", "SV Q", 7, 8.7, 5.3, 6.5, 5.4));

listSV.InsertLast(new SV("18", "SV R", 8.9, 8.6, 5.3, 6.5, 5.4));

listSV.InsertLast(new SV("19", "SV S", 5.2, 7, 5.3, 6.5, 5.4));

listSV.InsertLast(new SV("20", "SV T", 1.1, 3, 2, 5, 5));

}

public static void PrinListSV()

{

Console.Write("Chao mung cac ban den voi cuong trinh tim kiem Sinh Vien");

Console.Write("\nNhap so luong sinh vien muon in ra man hinh: ");

int n = int.Parse(Console.ReadLine());

Console.WriteLine("\n==========================================");

listSV.PrintList( n);

}

static void Main(string[] args)

{

taoListSV();

Console.WriteLine("Nhap vao so luong sinh vien muon in ra man hinh:");

int n = int.Parse(Console.ReadLine());

listSV.PrintList(n);

//Bang lua chon

Console.WriteLine("Chon 1 de xoa/Chon 2 de them");

Console.WriteLine("Chon: ");

int chon = int.Parse(Console.ReadLine());

if (chon == 1)

{

//Xoa sinh vien theo MaSV

Console.WriteLine("Nhap STT can xoa: ");

string canXoa = Console.ReadLine();

listSV.RemoveNode(canXoa);

n = n - 1;

listSV.PrintList( n);

}

if (chon == 2)

{

//Them sinh vien

Console.WriteLine("Nhap so Sv can them: ");

string canThem = Console.ReadLine();

SV svThem = new SV("10", "Ouip", 9.0, 9.0, 9.0, 9.0, 9.0);

listSV.InsertLast(svThem);

n = n + 1;

listSV.PrintList( n);

}

//Sort

listSV.InterchangeSort();

PrinListSV();

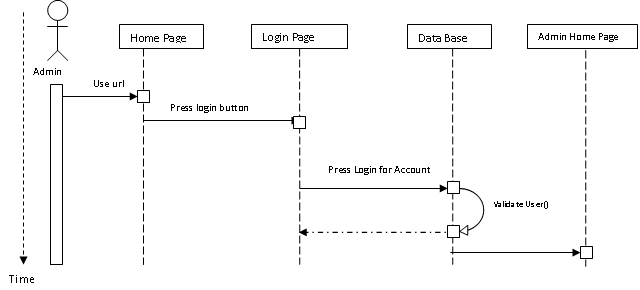
Console.ReadKey();

}

}

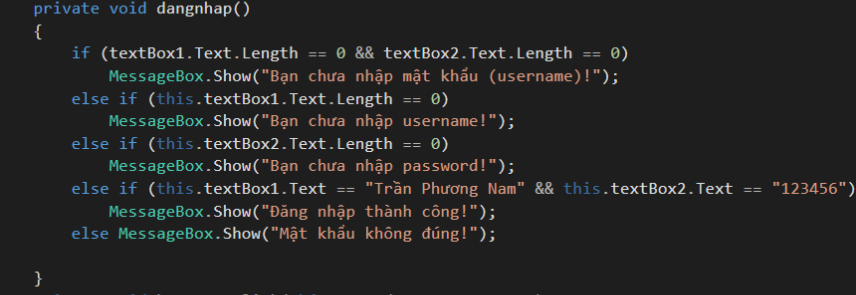
}

* + 1. **Sequence diagram**

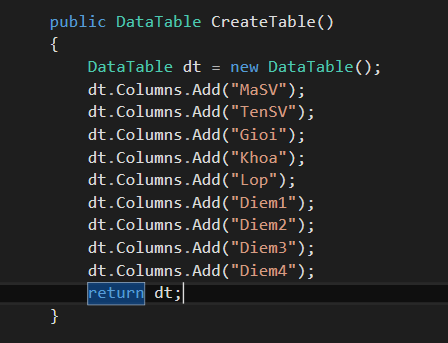


* 1. **Thiết kế các code chính**

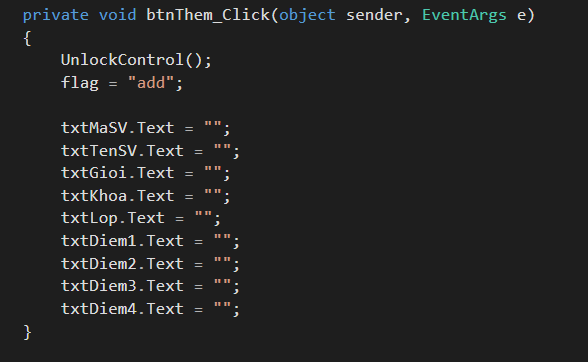
1. **Code bảng đăng nhập**



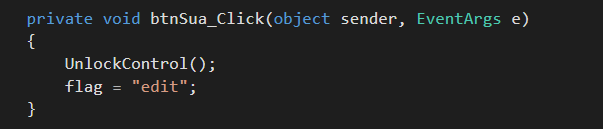
1. **Code tạo bảng datagridview**



1. **Code thêm sinh viên**



1. **Code sửa sinh viên**



1. **Code lưu sinh viên**

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

{

UnlockControl();

flag = "edit";

}

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

{

if (flag == "add")

{

if (checkData())

{

dtSV.Rows.Add(txtMaSV.Text, txtTenSV.Text, txtGioi.Text, txtKhoa.Text, txtLop.Text, txtDiem1.Text, txtDiem2.Text, txtDiem3.Text, txtDTB.Text);

dataGridSinhVien.DataSource = dtSV;

dataGridSinhVien.RefreshEdit();

}

}

else if (flag == "edit")

{

dtSV.Rows[index][0] = txtMaSV.Text;

dtSV.Rows[index][1] = txtTenSV.Text;

dtSV.Rows[index][2] = txtGioi.Text;

dtSV.Rows[index][3] = txtKhoa.Text;

dtSV.Rows[index][4] = txtLop.Text;

dtSV.Rows[index][5] = txtDiem1.Text;

dtSV.Rows[index][6] = txtDiem2.Text;

dtSV.Rows[index][7] = txtDiem3.Text;

dtSV.Rows[index][8] = txtDTB.Text;

dataGridSinhVien.DataSource = dtSV;

dataGridSinhVien.RefreshEdit();

}

LockControl();

}

1. **Code load form, unlockcontrol và lockcontrol**

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

{

LockControl();

dtSV = CreateTable();

}

private void label2\_Click(object sender, EventArgs e)

{

}

public void LockControl()

{

btnThem.Enabled = true;

btnSua.Enabled = true;

btnXoa.Enabled = true;

btnLuu.Enabled = false;

btnHuy.Enabled = false;

txtTenSV.ReadOnly = true;

txtMaSV.ReadOnly = true;

txtDiem1.ReadOnly = true;

txtDiem2.ReadOnly = true;

txtLop.ReadOnly = true;

txtGioi.ReadOnly = true;

txtDTB.ReadOnly = true;

btnThem.Focus();

}

public void UnlockControl()

{

btnThem.Enabled = false;

btnSua.Enabled = false;

btnXoa.Enabled = false;

btnLuu.Enabled = true;

btnHuy.Enabled = true;

txtMaSV.ReadOnly = false;

txtTenSV.ReadOnly = false;

txtGioi.ReadOnly = false;

txtKhoa.ReadOnly = false;

txtLop.ReadOnly = false;

txtDiem1.ReadOnly = false;

txtDiem2.ReadOnly = false;

txtDiem3.ReadOnly = false;

txtDTB.ReadOnly = false;

txtMaSV.Focus();

}

1. **Code check textbox**

public bool checkData()

{

if (string.IsNullOrWhiteSpace(txtMaSV.Text))

{

MessageBox.Show("Bạn chưa nhập mã sinh viên", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtMaSV.Focus();

return false;

}

if (string.IsNullOrWhiteSpace(txtTenSV.Text))

{

MessageBox.Show("Bạn chưa nhập tên sinh viên", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtTenSV.Focus();

return false;

}

if (string.IsNullOrWhiteSpace(txtGioi.Text))

{

MessageBox.Show("Bạn chưa nhập giới tính", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtGioi.Focus();

return false;

}

if (string.IsNullOrWhiteSpace(txtDTB.Text))

{

MessageBox.Show("Bạn chưa nhập tên khoa", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtDTB.Focus();

return false;

}

if (string.IsNullOrWhiteSpace(txtLop.Text))

{

MessageBox.Show("Bạn chưa nhập tên lớp", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtLop.Focus();

return false;

}

if (string.IsNullOrWhiteSpace(txtDiem1.Text))

{

MessageBox.Show("Bạn chưa nhập điểm môn 1", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtDiem1.Focus();

return false;

}

if (string.IsNullOrWhiteSpace(txtDiem2.Text))

{

MessageBox.Show("Bạn chưa nhập điểm môn 2", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtDiem2.Focus();

return false;

}

if (string.IsNullOrWhiteSpace(txtDiem2.Text))

{

MessageBox.Show("Bạn chưa nhập điểm môn 3", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtDiem3.Focus();

return false;

}

if (string.IsNullOrWhiteSpace(txtDiem2.Text))

{

MessageBox.Show("Bạn chưa nhập điểm trung bình", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtDTB.Focus();

return false;

}

return true;

}

1. **Code xóa SV**

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

{

if (MessageBox.Show("Bạn có muốn xóa sinh viên này không?","Cảnh báo",MessageBoxButtons.YesNo,MessageBoxIcon.Warning)==DialogResult.Yes)

{

dtSV.Rows.RemoveAt(index);

dataGridSinhVien.DataSource = dtSV;

dataGridSinhVien.RefreshEdit();

}

}

1. **Code Filter**

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

{

if (CBoxTTSort.SelectedItem == "a->z" && CBoxSortTheo.SelectedItem == "Tên SV")

{

dataGridSinhVien.Sort(dataGridSinhVien.Columns[1], ListSortDirection.Ascending);

}

else if (CBoxTTSort.SelectedItem == "z->a" && CBoxSortTheo.SelectedItem == "Tên SV")

{

dataGridSinhVien.Sort(dataGridSinhVien.Columns[1], ListSortDirection.Descending);

}

else if (CBoxTTSort.SelectedItem == "123" && CBoxSortTheo.SelectedItem == "Mã SV")

{

dataGridSinhVien.Sort(dataGridSinhVien.Columns[0], ListSortDirection.Ascending);

}

else if (CBoxTTSort.SelectedItem == "321" && CBoxSortTheo.SelectedItem == "Mã SV")

{

dataGridSinhVien.Sort(dataGridSinhVien.Columns[0], ListSortDirection.Descending);

}

else if (CBoxTTSort.SelectedItem == "123" && CBoxSortTheo.SelectedItem == "Lớp")

{

dataGridSinhVien.Sort(dataGridSinhVien.Columns[4], ListSortDirection.Ascending);

}

else if (CBoxTTSort.SelectedItem == "321" && CBoxSortTheo.SelectedItem == "Lớp")

{

dataGridSinhVien.Sort(dataGridSinhVien.Columns[4], ListSortDirection.Descending);

}

else if (CBoxTTSort.SelectedItem == "a->z" && CBoxSortTheo.SelectedItem == "Khoa")

{

dataGridSinhVien.Sort(dataGridSinhVien.Columns[3], ListSortDirection.Ascending);

}

else if (CBoxTTSort.SelectedItem == "123" && CBoxSortTheo.SelectedItem == "Điểm môn 1")

{

dataGridSinhVien.Sort(dataGridSinhVien.Columns[5], ListSortDirection.Ascending);

}

else if (CBoxTTSort.SelectedItem == "321" && CBoxSortTheo.SelectedItem == "Điểm môn 1")

{

dataGridSinhVien.Sort(dataGridSinhVien.Columns[5], ListSortDirection.Descending);

}

else if (CBoxTTSort.SelectedItem == "123" && CBoxSortTheo.SelectedItem == "Điểm môn 2")

{

dataGridSinhVien.Sort(dataGridSinhVien.Columns[6], ListSortDirection.Ascending);

}

else if (CBoxTTSort.SelectedItem == "321" && CBoxSortTheo.SelectedItem == "Điểm môn 2")

{

dataGridSinhVien.Sort(dataGridSinhVien.Columns[6], ListSortDirection.Descending);

}

else if (CBoxTTSort.SelectedItem == "123" && CBoxSortTheo.SelectedItem == "Điểm môn 3")

{

dataGridSinhVien.Sort(dataGridSinhVien.Columns[7], ListSortDirection.Ascending);

}

else if (CBoxTTSort.SelectedItem == "321" && CBoxSortTheo.SelectedItem == "Điểm môn 3")

{

dataGridSinhVien.Sort(dataGridSinhVien.Columns[7], ListSortDirection.Descending);

}

else if (CBoxTTSort.SelectedItem == "123" && CBoxSortTheo.SelectedItem == "Điểm TB")

{

dataGridSinhVien.Sort(dataGridSinhVien.Columns[8], ListSortDirection.Ascending);

}

else if (CBoxTTSort.SelectedItem == "321" && CBoxSortTheo.SelectedItem == "Điểm TB")

{

dataGridSinhVien.Sort(dataGridSinhVien.Columns[8], ListSortDirection.Descending);

}

}

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

{

if (CBoxTTSort.SelectedItem == "a->z" && CBoxSortTheo.SelectedItem == "Tên SV")

{

dataGridSinhVien.Sort(dataGridSinhVien.Columns[1], ListSortDirection.Ascending);

}

else if (CBoxTTSort.SelectedItem == "z->a" && CBoxSortTheo.SelectedItem == "Tên SV")

{

dataGridSinhVien.Sort(dataGridSinhVien.Columns[1], ListSortDirection.Descending);

}

else if (CBoxTTSort.SelectedItem == "123" && CBoxSortTheo.SelectedItem == "Mã SV")

{

dataGridSinhVien.Sort(dataGridSinhVien.Columns[0], ListSortDirection.Ascending);

}

else if (CBoxTTSort.SelectedItem == "321" && CBoxSortTheo.SelectedItem == "Mã SV")

{

dataGridSinhVien.Sort(dataGridSinhVien.Columns[0], ListSortDirection.Descending);

}

else if (CBoxTTSort.SelectedItem == "123" && CBoxSortTheo.SelectedItem == "Lớp")

{

dataGridSinhVien.Sort(dataGridSinhVien.Columns[4], ListSortDirection.Ascending);

}

else if (CBoxTTSort.SelectedItem == "321" && CBoxSortTheo.SelectedItem == "Lớp")

{

dataGridSinhVien.Sort(dataGridSinhVien.Columns[4], ListSortDirection.Descending);

}

else if (CBoxTTSort.SelectedItem == "a->z" && CBoxSortTheo.SelectedItem == "Khoa")

{

dataGridSinhVien.Sort(dataGridSinhVien.Columns[3], ListSortDirection.Ascending);

}

else if (CBoxTTSort.SelectedItem == "123" && CBoxSortTheo.SelectedItem == "Điểm môn 1")

{

dataGridSinhVien.Sort(dataGridSinhVien.Columns[5], ListSortDirection.Ascending);

}

else if (CBoxTTSort.SelectedItem == "321" && CBoxSortTheo.SelectedItem == "Điểm môn 1")

{

dataGridSinhVien.Sort(dataGridSinhVien.Columns[5], ListSortDirection.Descending);

}

else if (CBoxTTSort.SelectedItem == "123" && CBoxSortTheo.SelectedItem == "Điểm môn 2")

{

dataGridSinhVien.Sort(dataGridSinhVien.Columns[6], ListSortDirection.Ascending);

}

else if (CBoxTTSort.SelectedItem == "321" && CBoxSortTheo.SelectedItem == "Điểm môn 2")

{

dataGridSinhVien.Sort(dataGridSinhVien.Columns[6], ListSortDirection.Descending);

}

else if (CBoxTTSort.SelectedItem == "123" && CBoxSortTheo.SelectedItem == "Điểm môn 3")

{

dataGridSinhVien.Sort(dataGridSinhVien.Columns[7], ListSortDirection.Ascending);

}

else if (CBoxTTSort.SelectedItem == "321" && CBoxSortTheo.SelectedItem == "Điểm môn 3")

{

dataGridSinhVien.Sort(dataGridSinhVien.Columns[7], ListSortDirection.Descending);

}

else if (CBoxTTSort.SelectedItem == "123" && CBoxSortTheo.SelectedItem == "Điểm TB")

{

dataGridSinhVien.Sort(dataGridSinhVien.Columns[8], ListSortDirection.Ascending);

}

else if (CBoxTTSort.SelectedItem == "321" && CBoxSortTheo.SelectedItem == "Điểm TB")

{

dataGridSinhVien.Sort(dataGridSinhVien.Columns[8], ListSortDirection.Descending);

}

}

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

{

CBoxTTSort.Items.Clear();

if (CBoxSortTheo.SelectedItem.Equals("Mã SV") || CBoxSortTheo.SelectedItem.Equals("Lop"))

{

CBoxTTSort.Items.Add("123");

CBoxTTSort.Items.Add("321");

}

else if (CBoxSortTheo.SelectedItem.Equals("Tên SV") || CBoxSortTheo.SelectedItem.Equals("Khoa"))

{

CBoxTTSort.Items.Add("a->z");

CBoxTTSort.Items.Add("z->a");

}

else if (CBoxSortTheo.SelectedItem.Equals("Điểm môn 1") || CBoxSortTheo.SelectedItem.Equals("Điểm môn 2") || CBoxSortTheo.SelectedItem.Equals("Điểm môn 3") || CBoxTTSort.SelectedItem.Equals("Điểm TB"))

{

CBoxTTSort.Items.Add("max->min");

CBoxTTSort.Items.Add("min->max");

}

else

{

CBoxTTSort.Items.Add("Nam");

CBoxTTSort.Items.Add("Nữ");

}

}

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

{

string rowFilter = string.Format("[{0}] = '{1}'", dtSV.Columns[3], CBoxSTK.SelectedItem);

(dataGridSinhVien.DataSource as DataTable).DefaultView.RowFilter = rowFilter;

}

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

{

string rowFilter = string.Format("[{0}] = '{1}'", dtSV.Columns[4], CBoxSTL.SelectedItem);

(dataGridSinhVien.DataSource as DataTable).DefaultView.RowFilter = rowFilter;

}

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

{

string rowFilter = string.Format("[{0}] = '{1}'", dtSV.Columns[2], CBoxSTG.SelectedItem);

(dataGridSinhVien.DataSource as DataTable).DefaultView.RowFilter = rowFilter;

}

}

}

* 1. **Cài đặt và thử nghiệm**

- Chương trình được cài đặt trên nền Visual Studio 2017, sử dụng ngôn ngữ C# và Pro

1. **Mô tả phần công việc**

|  |  |  |  |
| --- | --- | --- | --- |
| **Công Việc** | **Nguyễn Lê Minh Vũ** | **Trần Phương Nam** | **Ghi Chú** |
| Thiết kế giao diện Form | X | X | 50%-50% |
| Thiết kế Form main screen | X | X | 50%-50% |
| Thiết kế Form Add, Edit, Delete | X | X | 50%-50% |
| Làm phần hình ảnh cho các Form | X | X | 50%-50% |
| Thiết kế main screen | X | X | 50%-50% |
| Thực hiện hàm menu lựa chọn | X | X | 50%-50% |
| Thực hiện hàm thêm, xóa, sửa | X | X | 50%-50% |
| Báo cáo đồ án | X | X | 50%-50% |

1. **Kết Luận**
2. **Kết quả**

Sau khi trải qua nửa học kì nghiên cứu cũng như thực hiện với sự giúp đỡ của giáo viên hướng dẫn, nhóm chúng em đã thực hiện được một chương trình Danh sách sinh viên ứng dụng kiến thức môn Cấu trúc dữ liệu và Giải thuật là danh sách liên kết.

1. **Khó khăn**

* Do tụi em gặp khá nhiều khó khăn trong việc viết code và lấy điểm bài tập nên chưa tập trung 100% sức lực vào đồ án.
* Vì tụi em chưa biết sử dụng Github nên phải gửi bài cho nhau qua Facebook và lịch học 2 đứa không trùng nhau nên việc học nhóm khá là khó khăn.
* Có một vài thiếu sót trong việc sử dụng danh sách liên kết trong winform nhưng nhờ giảng viên hướng dẫn mà tụi em đã khắc phục và giải quyết tốt hơn.

1. **Ý tưởng**

* Nếu có thêm thời gian nghiên cứu thì tụi em đã có thể sử dụng cơ sở dữ liệu (database) để lưu danh sách sinh viên thì code sẽ ngắn gọn hơn và thao tác sẽ dễ dàng hơn.

1. **Ưu điểm**

Phần mềm có giao diện khá đẹp, gần gũi với người dùng, tốc độ nhanh thuận tiện cho việc tra

cứu sinh viên cho phpws người dung thêm, xóa và sửa những thông tin cần thiết của các sinh viên trong danh sách

1. **Nhược điểm**

Khi sử dụng khá là mất thời gian khi tìm kiếm sinh viên vì phải nhập đầy đủ thông tin như Họ và tên và MSSV nên mới chạy ra được thông tin của sinh viên đó.