

TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG HÀ NỘI
KHOA CÔNG NGHỆ THÔNG TIN



BÀI BÁO CÁO MÔN CÔNG NGHỆ.NET

**Đề tài bài tập lớn : XÂY DỰNG PHẦN MỀM QUẢN LÝ
SINH VIÊN TRÊN NỀN WINDOWS**

NHÓM 9

Họ và tên sinh viên : 1. Đồng Tiến Hiệp

2. An Vũ Sơn

3. Bùi Văn Phương

4. Nguyễn Tuyết Nhung

5. Lê Văn Hiếu

Lớp : DH11C4

Giảng viên giảng dạy : Thầy Vũ Văn Huân & Bùi Tá Hậu

Hà Nội – 2024

Mục lục

Không tìm thấy mục nào của mục lục.

1. CHƯƠNG 1: MỞ ĐẦU

1.1. Giới thiệu

1.1.1. Lý do chọn đề tài

Lý do chúng tôi chọn đề tài "tài quản lý sinh viên" không chỉ đến từ nhu cầu thực tiễn mà còn từ sự hiểu biết sâu sắc về tầm quan trọng của việc này. Dưới đây là một số lý do mà chúng tôi tin rằng đề tài này mang lại giá trị lớn đối với cộng đồng giáo dục:

Tính Thực Tiễn: Sinh viên là nhân tố chính trong môi trường giáo dục. Việc quản lý thông tin của họ không chỉ giúp cho việc tổ chức hệ thống học tập mà còn đảm bảo rằng họ nhận được sự hỗ trợ và tư vấn đầy đủ.

Tăng Cường Hiệu Quả: Bằng cách sử dụng một hệ thống quản lý sinh viên, trường học có thể tối ưu hóa các quy trình quản lý và tạo điều kiện thuận lợi cho sinh viên cũng như cán bộ quản lý.

Tiện Ích và Linh Hoạt: Đề tài này tập trung vào việc xây dựng một trang web, mang lại tính tiện ích và linh hoạt cao cho người dùng. Khả năng truy cập từ xa và tương tác dễ dàng là những yếu tố quan trọng được xem xét.

Sự Thú Vị và Thách Thức: Phát triển một hệ thống quản lý sinh viên không chỉ là một dự án kỹ thuật mà còn là một thách thức sáng tạo. Nó đòi hỏi sự kết hợp giữa kiến thức về công nghệ thông tin và sự hiểu biết sâu sắc về quản lý giáo dục.

1.1.2. Tầm quan trọng của đề tài

Tối Ưu Hóa Quy Trình Quản Lý: Trang web quản lý sinh viên giúp tối ưu hóa và tự động hóa các quy trình quản lý, từ việc nhập liệu thông tin sinh viên đến việc quản lý điểm số và thông báo. Điều này giúp giảm thiểu các công việc thủ công, tăng cường hiệu suất và giảm thiểu sai sót trong quản lý.

Tăng Cường Tương Tác Sinh Viên - Cán Bộ Quản Lý: Trang web này tạo điều kiện thuận lợi cho việc tương tác giữa sinh viên và cán bộ quản lý thông qua các tính năng như tạo hồ sơ sinh viên, xem lịch học, gửi thông

báo, và tra cứu thông tin. Điều này giúp tạo ra một môi trường học tập tích cực và tạo sự thuận tiện cho cả hai bên.

Tiện Ích và Linh Hoạt: Sự linh hoạt của trang web quản lý sinh viên cho phép truy cập từ xa thông qua mạng Internet, giúp cho sinh viên và cán bộ quản lý có thể truy cập thông tin mọi lúc, mọi nơi. Điều này tạo ra sự thuận tiện và tiết kiệm thời gian đáng kể.

Quản Lý Thông Tin Hiệu Quả: Trang web này cung cấp một nền tảng để quản lý tất cả các thông tin liên quan đến sinh viên như thông tin cá nhân, kết quả học tập, lịch học, và thông tin liên lạc. Việc tổ chức thông tin một cách cụ thể và dễ dàng truy cập giúp cho quản lý trở nên hiệu quả hơn. Nghiên cứu sơ bộ

1.2. Báo cáo tính khả thi của sản phẩm

1.2.1. Tính khả thi về kỹ thuật

Yêu cầu chức năng: Đảm bảo rằng trang web có đầy đủ các tính năng cần thiết như quản lý thông tin sinh viên, điểm số, lịch học, thông báo, và tương tác trực tuyến. Xác định các yêu cầu cụ thể của người dùng để đảm bảo tính khả thi của sản phẩm.

Tài nguyên: Đánh giá tài nguyên như nhân lực, ngân sách và thời gian để phát triển sản phẩm. Xác định liệu có đủ tài nguyên để hoàn thành dự án theo kế hoạch và trong phạm vi ngân sách được giao hay không.

Phân tích rủi ro: Đưa ra đánh giá về các rủi ro có thể ảnh hưởng đến tính khả thi của dự án như rủi ro về kỹ thuật, rủi ro về thị trường, và rủi ro về tài chính. Phát triển các biện pháp để giảm thiểu và quản lý các rủi ro này.

Phản hồi từ người dùng: Tiến hành khảo sát và thu thập phản hồi từ người dùng tiềm năng để đánh giá sự quan tâm và tiếp nhận của thị trường đối với sản phẩm. Phản hồi này sẽ giúp đánh giá tính khả thi của dự án trong môi trường thực tế. Tính khả thi về mặt kinh tế

2 . CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

2.2. Tổng quan ứng dụng

2.2.1. Đối tượng sử dụng ứng dụng

Sinh viên: Sinh viên là nhóm đối tượng chính sử dụng ứng dụng này để quản lý thông tin cá nhân, theo dõi lịch học, xem kết quả học tập, và nhận thông báo từ trường. Họ sử dụng ứng dụng để tra cứu thông tin về các khoá học, đăng ký lớp học, và tương tác với giảng viên và cán bộ quản lý.

Giáo viên và giảng viên: Giáo viên và giảng viên sử dụng ứng dụng để quản lý thông tin về sinh viên trong lớp học, ghi chú điểm, và cung cấp thông báo liên quan đến hoạt động giảng dạy và học tập. Họ cũng sử dụng ứng dụng để tương tác với sinh viên qua các chức năng như gửi thông điệp hoặc lên lịch hẹn.

Cán bộ quản lý và nhân viên hành chính: Cán bộ quản lý và nhân viên hành chính của trường sử dụng ứng dụng để quản lý thông tin sinh viên toàn trường, bao gồm việc xem xét và cập nhật hồ sơ sinh viên, điều chỉnh lịch học, và phát thông báo chung. Họ cũng sử dụng ứng dụng để tạo và quản lý các sự kiện và hoạt động ngoại khóa cho sinh viên.

Ban quản lý trường: Ban quản lý trường sử dụng ứng dụng để xem tổng quan về các hoạt động học tập và quản lý tài nguyên của trường, từ thông tin sinh viên đến lịch học và cán bộ giáo viên. Họ cũng sử dụng ứng dụng để đưa ra các quyết định chiến lược và phát triển chính sách liên quan đến quản lý sinh viên và hoạt động giáo dục của trường.

2.2.2. Các chức năng chính của phần mềm

Hệ thống cần có chức năng hỗ trợ và giải quyết nhu cầu của Admin như sau:

Đăng nhập, đăng xuất

Khi muốn sử dụng phần mềm, admin và Người dùng cần đăng nhập bằng tài khoản có sẵn, từ đó có thể sử dụng các chức năng của phần mềm. Khi thực hiện chức năng xong, admin và khách hàng có thể thoát bằng chức năng đăng xuất.

Quản lý thông tin sinh viên: Tạo và quản lý hồ sơ cá nhân của sinh viên, bao gồm thông tin cơ bản, học vấn, và thông tin liên lạc. Cập nhật thông tin sinh viên khi cần thiết, như thay đổi địa chỉ hoặc thông tin liên hệ.

Quản lý lớp học và học phần: Tạo và quản lý các lớp học và học phần, bao gồm thông tin về giảng viên, lịch học, và địa điểm. Cho phép đăng ký và rút khỏi các lớp học và học phần theo yêu cầu của sinh viên.

Quản lý điểm số: Ghi chép và tính toán điểm số của sinh viên cho từng khoá học và học phần. Cung cấp phản hồi về điểm số cho sinh viên và giáo viên.

Quản lý lịch học: Hiển thị lịch học cho sinh viên và giáo viên, bao gồm thông tin về thời gian, địa điểm, và nội dung của từng buổi học. Cập nhật thông tin lịch học khi có thay đổi hoặc hủy bỏ.

Tra cứu thông tin và thống kê: Cho phép tra cứu thông tin sinh viên và học phần dựa trên các tiêu chí như tên, mã số sinh viên, hoặc khoá học. Tạo báo cáo và thống kê về tiến trình học tập, điểm số, và các chỉ số khác cho sinh viên và cán bộ quản lý.

2.2.3 Xác định Usecase

Hệ thống có các Usecase sau:

Đăng nhập (ADMIN/Giảng viên , sinh viên)

Quản lý tài khoản (ADMIN)

Quản lý sinh viên(ADMIN)

Quản lý giảng viên(ADMIN)

Quản lý lớp học phần(ADMIN)

Quản lý môn học (ADMIN)

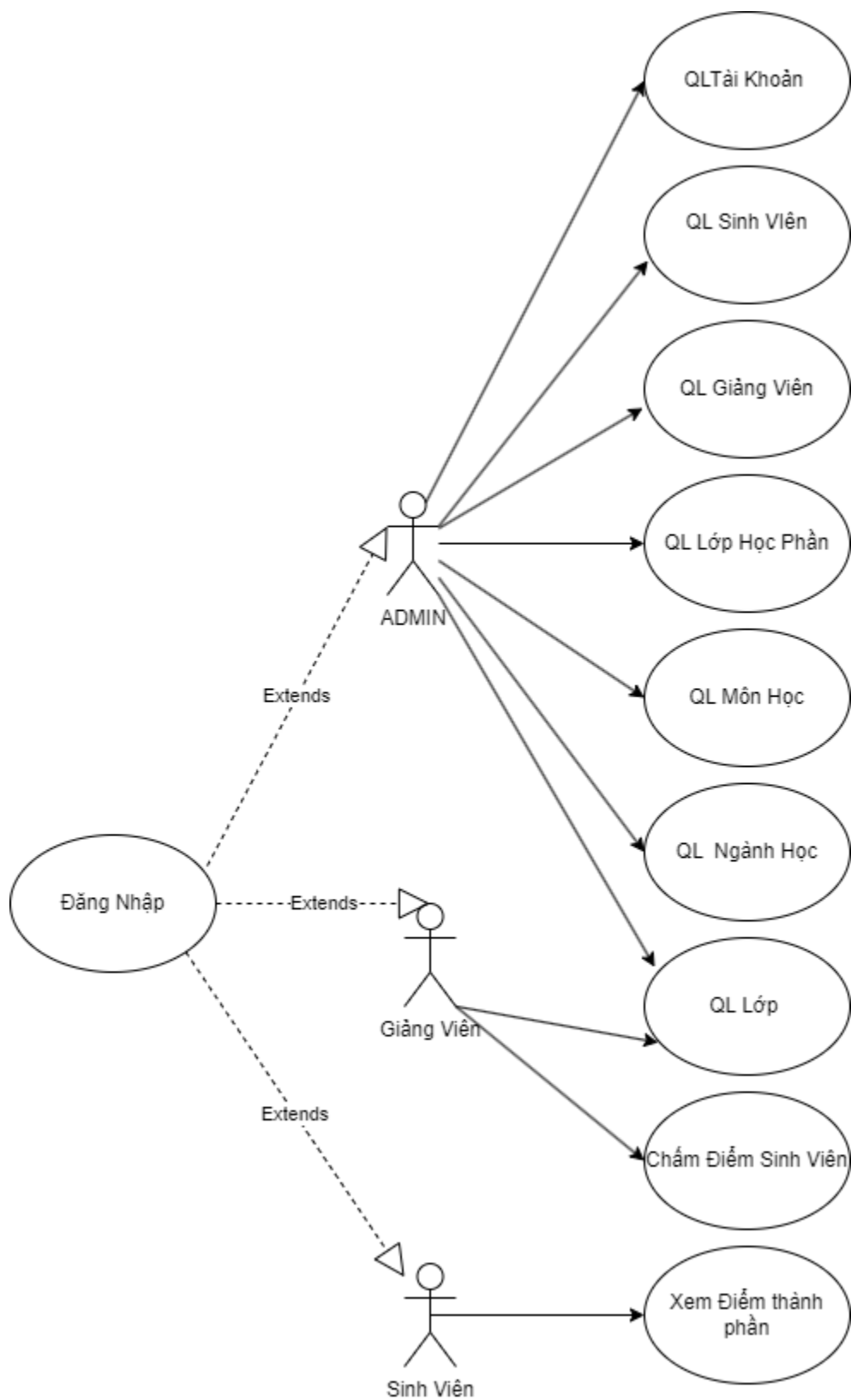
Quản lý ngành học(ADMIN)

Quản lý lớp(ADMIN)

Quản lý kết Quả Học tập (Giảng viên/Sinh viên)

Quản lý thông tin cá nhân (Sinh viên)

2.3. Biểu đồ Usecase tổng quát

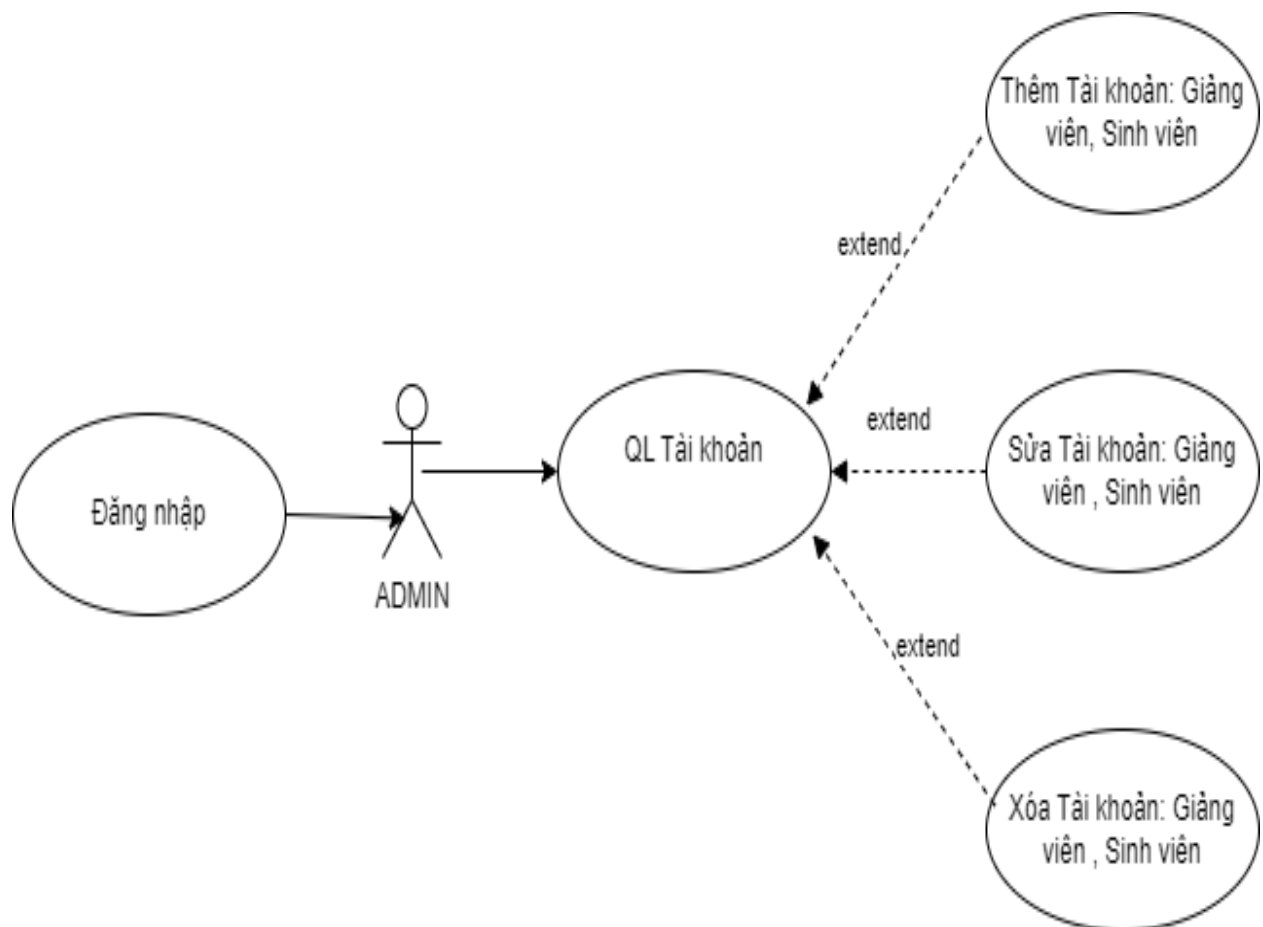


2.4. Biểu đồ Usecase phân rã

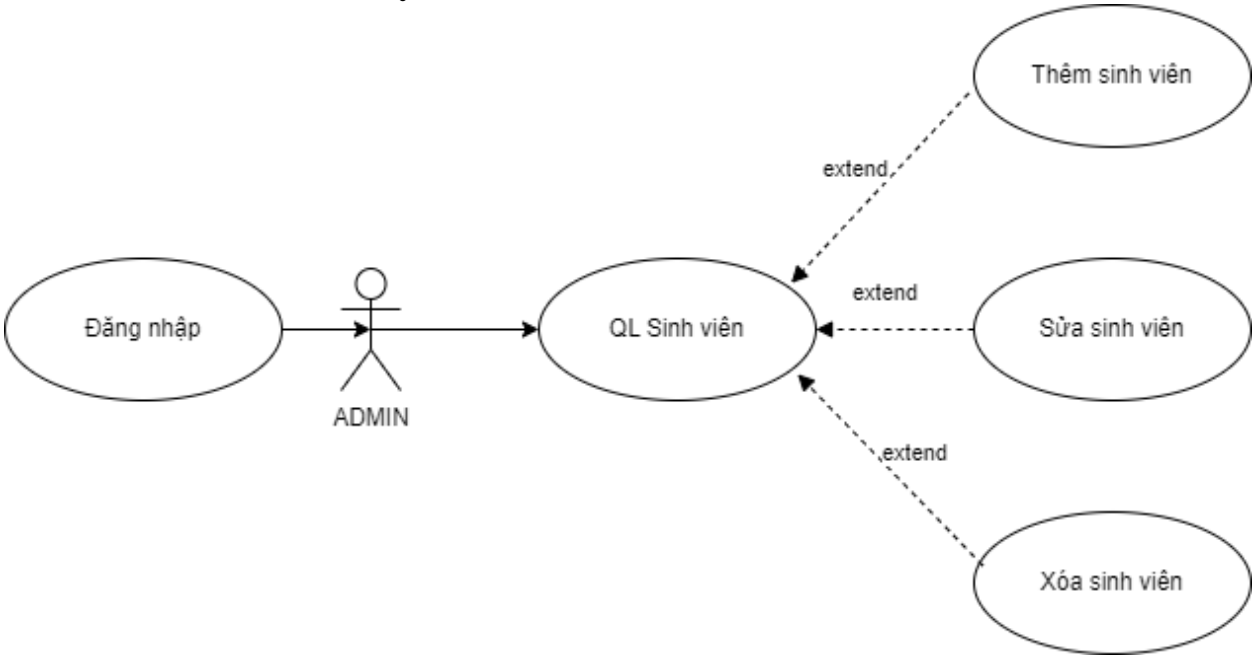
2.4.1. Đăng Nhập



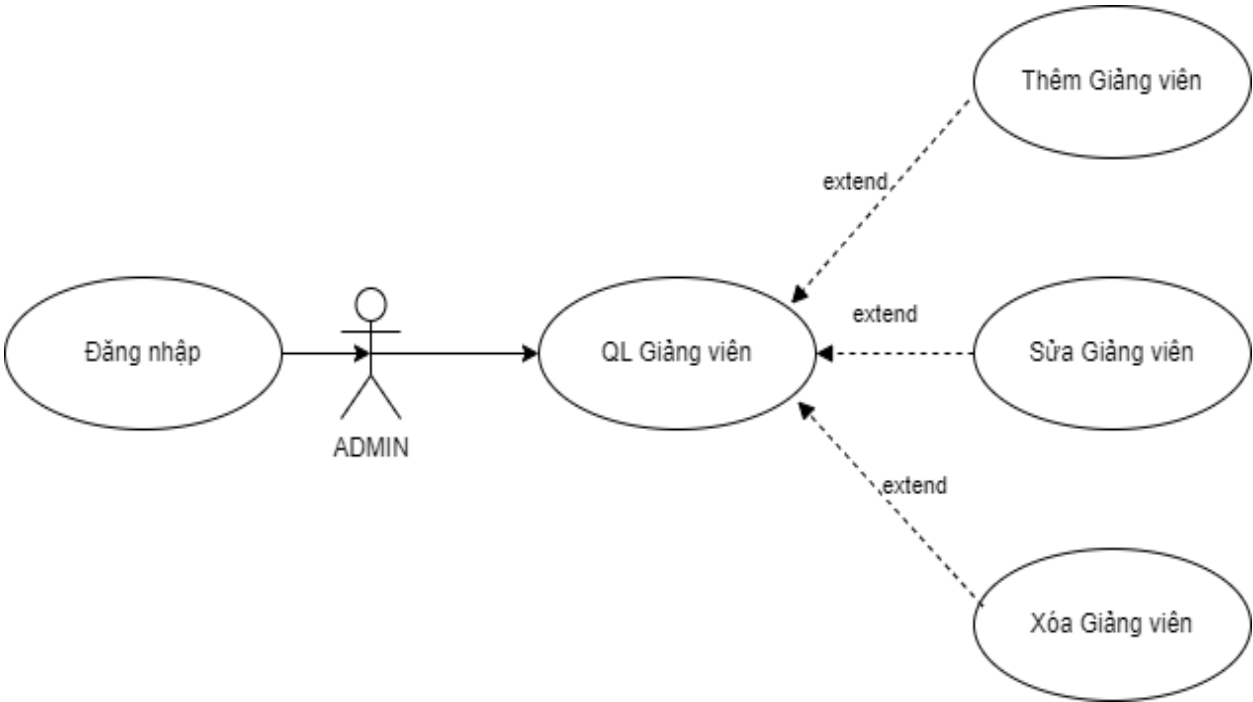
2.4.2. Quản lý tài khoản



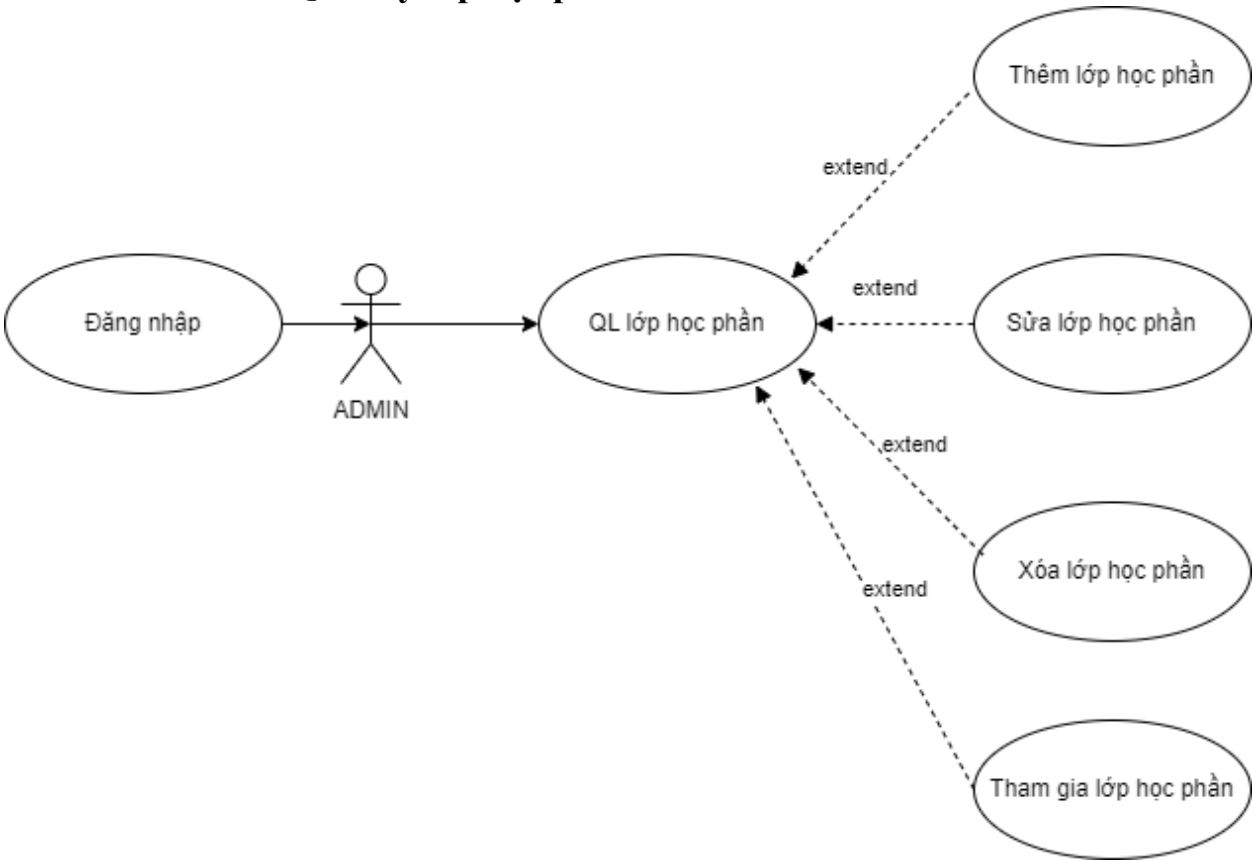
2.4.3. Quản lý sinh viên



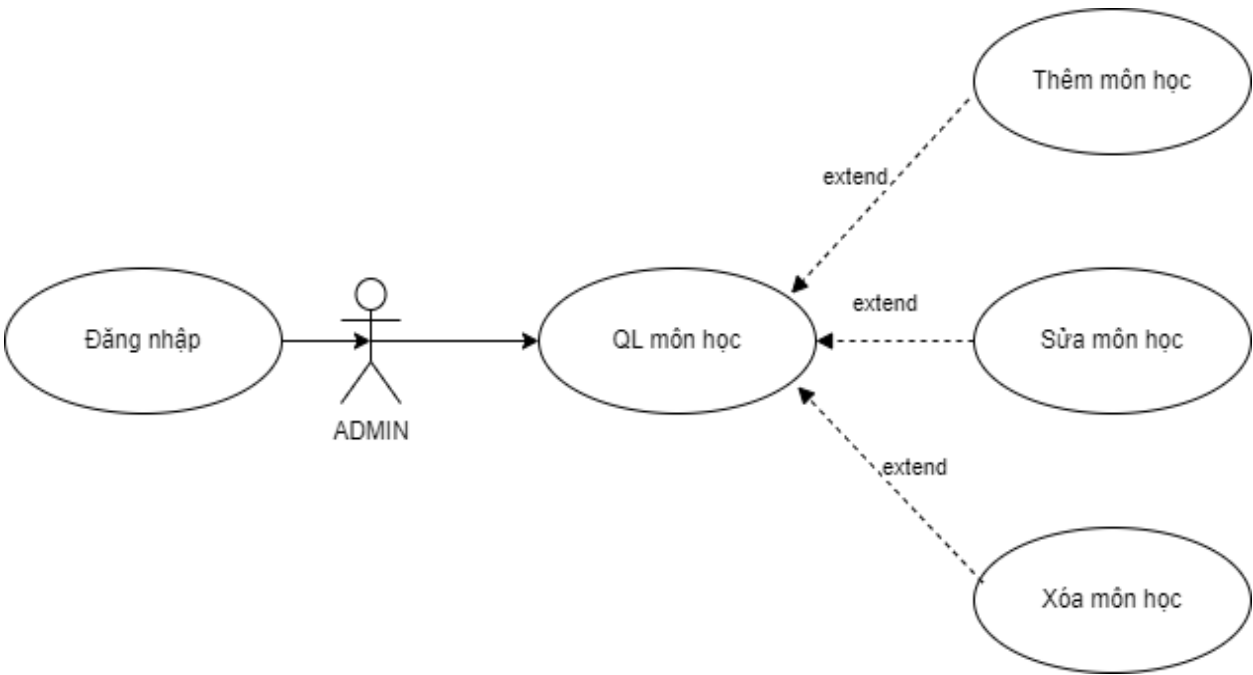
2.4.4. Quản lý giảng viên



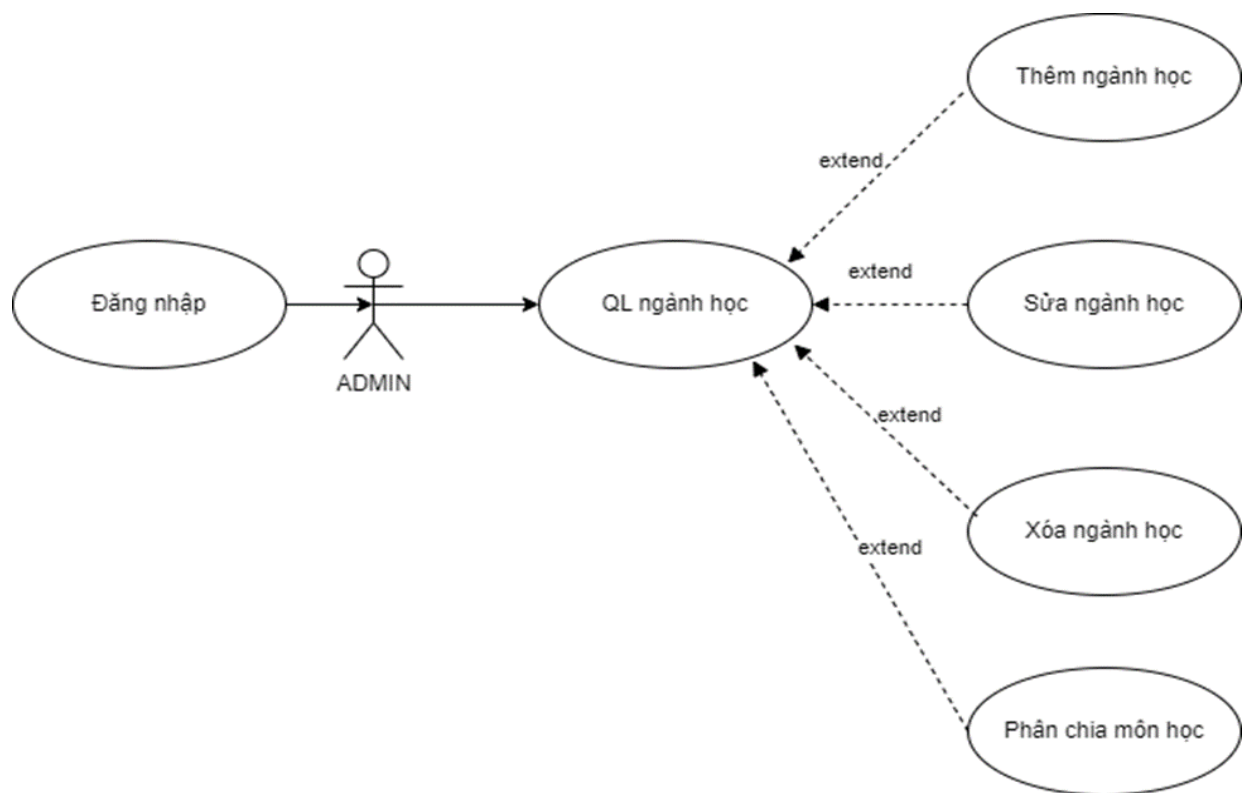
2.4.5. Quản lý lớp học phần



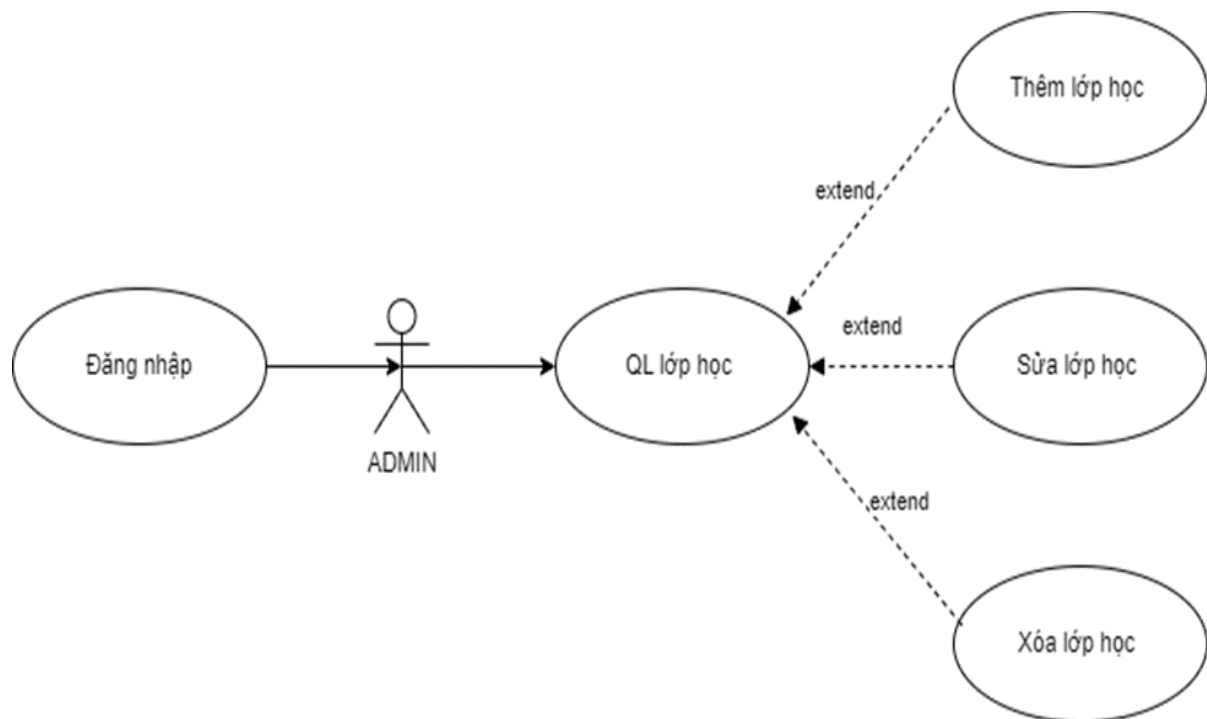
2.4.6. Quản lý môn học



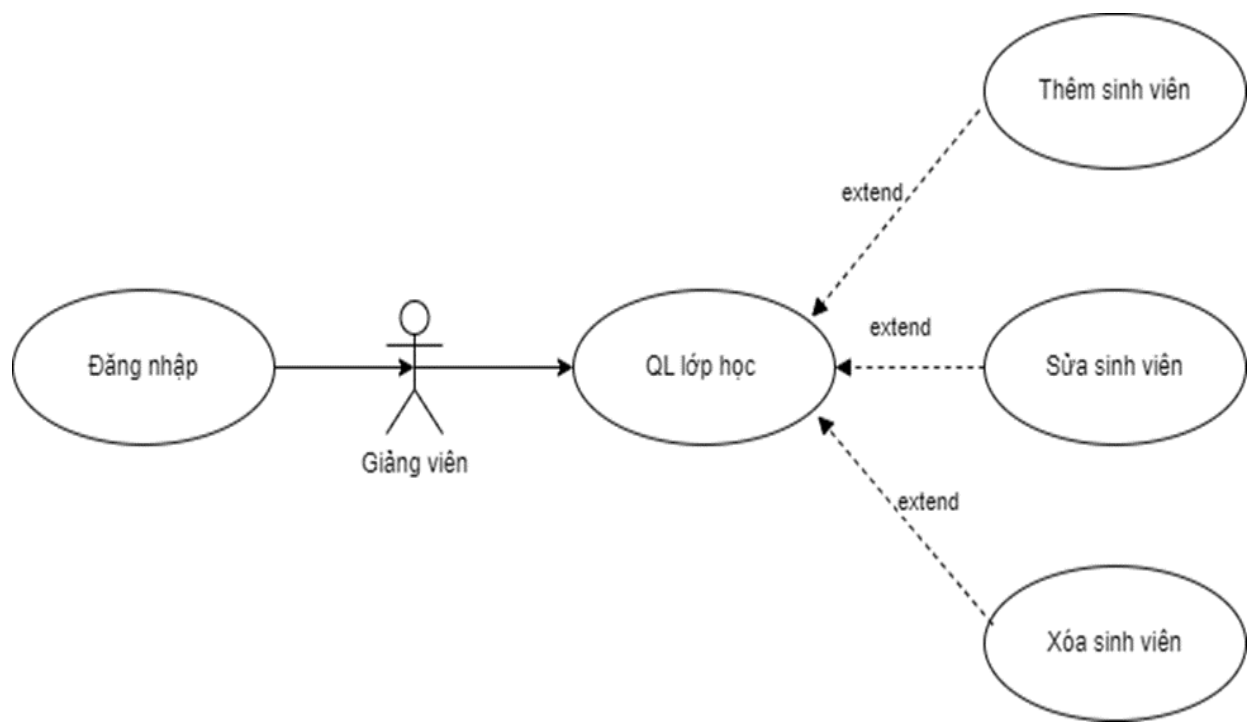
2.4.7. Quản lý ngành học



2.4.8. Quản lý lớp



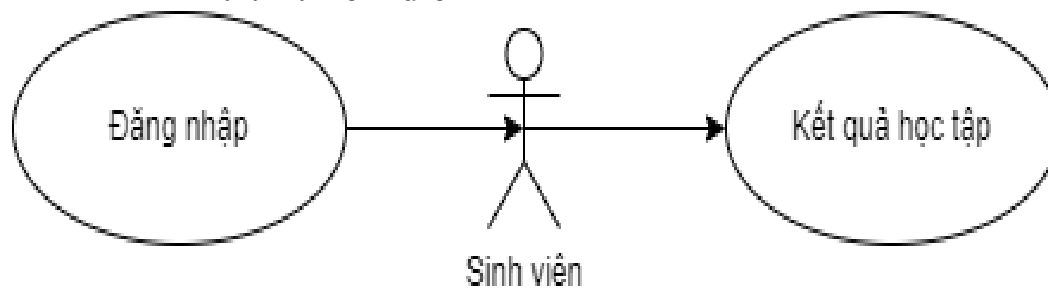
2.4.9. Giảng viên quản lý lớp



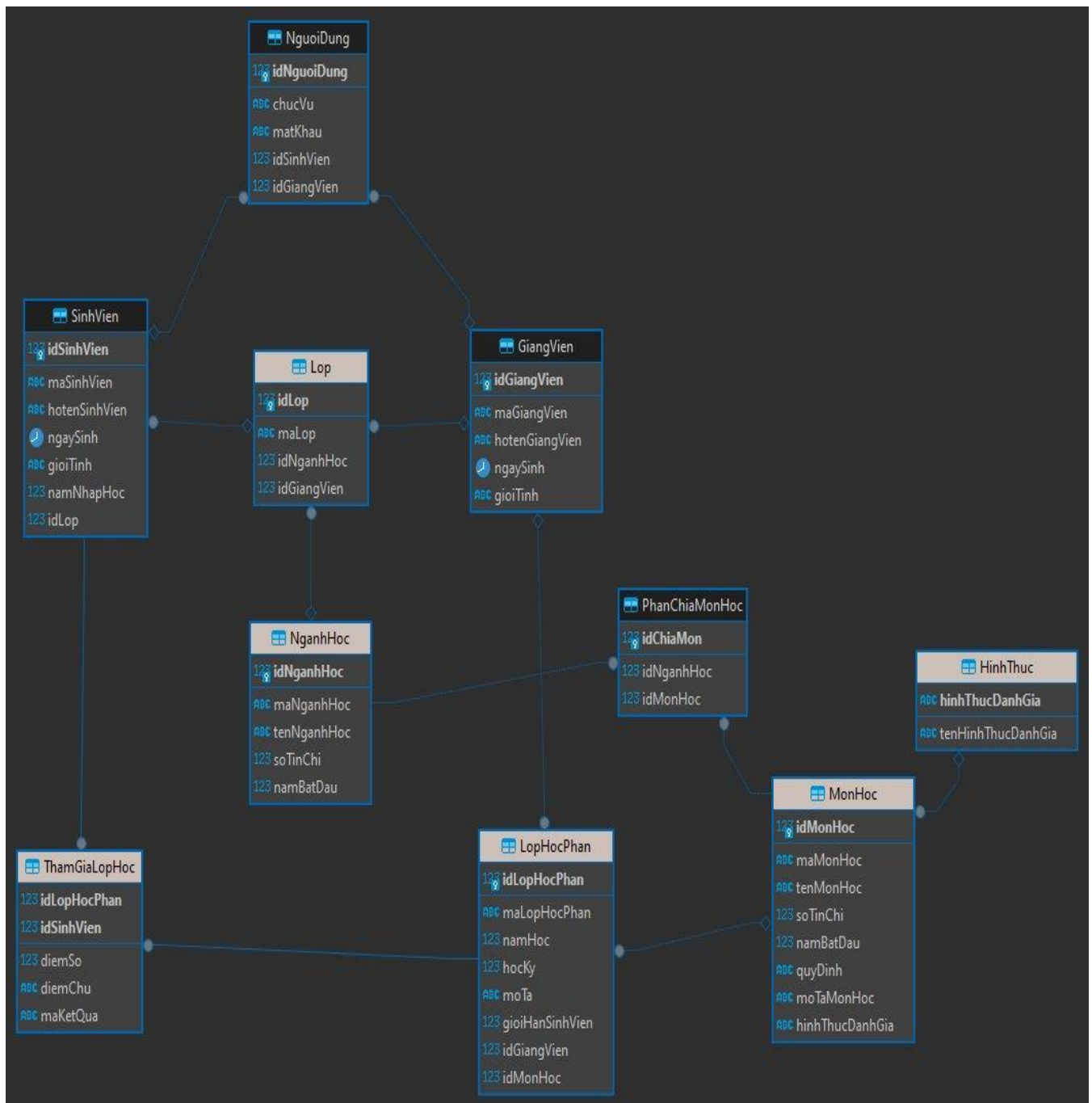
2.4.10. Quản lý điểm



2.4.11. Xem điểm



2.5. Mối Quan Hệ Giữa Các Bảng



2.6. Thiết Kế Cơ Sở Dữ Liệu

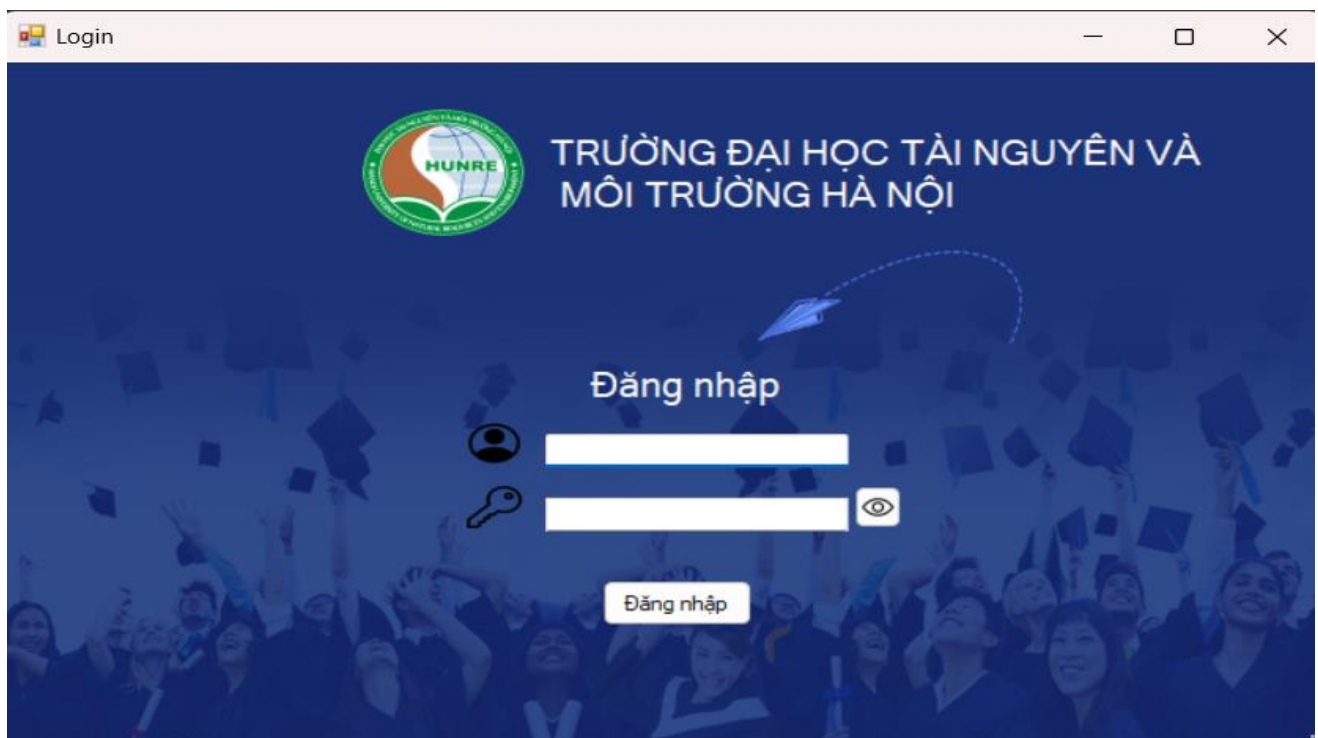
- Phân thiết kế cơ sở dữ liệu gồm những bảng
 - + Bảng dữ liệu *dbo.GiangVien*
 - + Bảng dữ liệu *dbo.HinhThuc*
 - + Bảng dữ liệu *dbo.Lop*
 - + Bảng dữ liệu *dbo.LopHocPhan*
 - + Bảng dữ liệu *dbo.MonHoc*

- + Bảng dữ liệu *dbo.NganhHoc*
- + Bảng dữ liệu *dbo.NguoiDung*
- + Bảng dữ liệu *dbo.PhanChiaMonHoc*
- + Bảng dữ liệu *dbo.SinhVien*
- + Bảng dữ liệu *dbo.ThamGiaLopHoc*

3. Chương 3: XÂY DỰNG GIAO DIỆN CHƯƠNG TRÌNH

3.1. Giao Diện Form Đăng Nhập

- Người dùng có thể đăng nhập vào hệ thống .
- Người dùng có thể đăng nhập với lựa chọn là Sinh Viên, Admin hoặc Giảng Viên sẽ hiện ra Form theo lựa chọn của người dùng.
- Giao diện đăng nhập gồm các chức năng như : Đăng Nhập , Hiển Thị Mật Khẩu



Hình 3.1.1. Form Đăng Nhập

```

if (db.NguoiDung.Where(tk => tk.SinhVien.maSinhVien == tbUserName.Text && tk.matKhau == tbPassWord.Text).SingleOrDefault() != null)
{
    MessageBox.Show("Đăng nhập thành công");
    // Nếu thông tin đăng nhập chính xác, đóng form đăng nhập và mở form menu chính
    this.Hide();
    MenuSinhVien menuSV = new MenuSinhVien(tbUserName.Text);
    menuSV.ShowDialog();
    this.Close();
}
else if (db.NguoiDung.Where(tk => tk.GiangVien.maGiangVien == tbUserName.Text && tk.matKhau == tbPassWord.Text).SingleOrDefault() != null)
{
    MessageBox.Show("Đăng nhập thành công");
    // Nếu thông tin đăng nhập chính xác, đóng form đăng nhập và mở form menu chính
    this.Hide();
    KetQuaHocPhanForm menuGV = new KetQuaHocPhanForm(tbUserName.Text);
    menuGV.ShowDialog();
    this.Close();
}
else if (db.NguoiDung.Where(tk => tk.matKhau == tbPassWord.Text && tk.chucVu == tbUserName.Text).SingleOrDefault() != null && tbUserName.Text == "ADMIN")
{
    MessageBox.Show("Đăng nhập thành công");
    // Nếu thông tin đăng nhập chính xác, đóng form đăng nhập và mở form menu chính
    this.Hide();
    MenuADMIN menuADMIN = new MenuADMIN();
    menuADMIN.ShowDialog();
    this.Close();
}
else if (tbPassWord == null || tbUserName == null)
{
    MessageBox.Show("Vui lòng điền đầy đủ thông tin");
    tbUserName.Clear();
    tbPassWord.Clear();
}
else
{
    MessageBox.Show("Tài khoản hoặc mật khẩu không chính xác");
    tbUserName.Clear();
    tbPassWord.Clear();
}

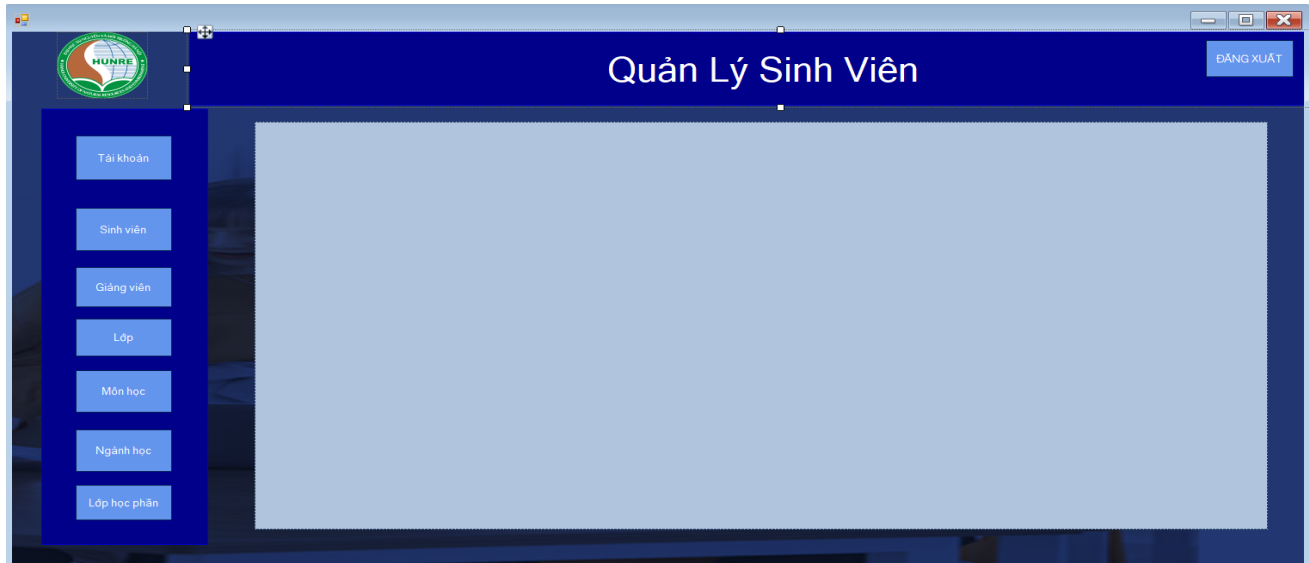
private void button1_Click(object sender, EventArgs e)
{
    tbPassWord.UseSystemPasswordChar = !tbPassWord.UseSystemPasswordChar;
}

```

3.2. Giao Diện Form MENU ADMIN

- Giao diện bao 7 giao diện con với chức năng tương ứng của giao diện đó:
 - Tài khoản: Quản lý tài khoản Đăng nhập

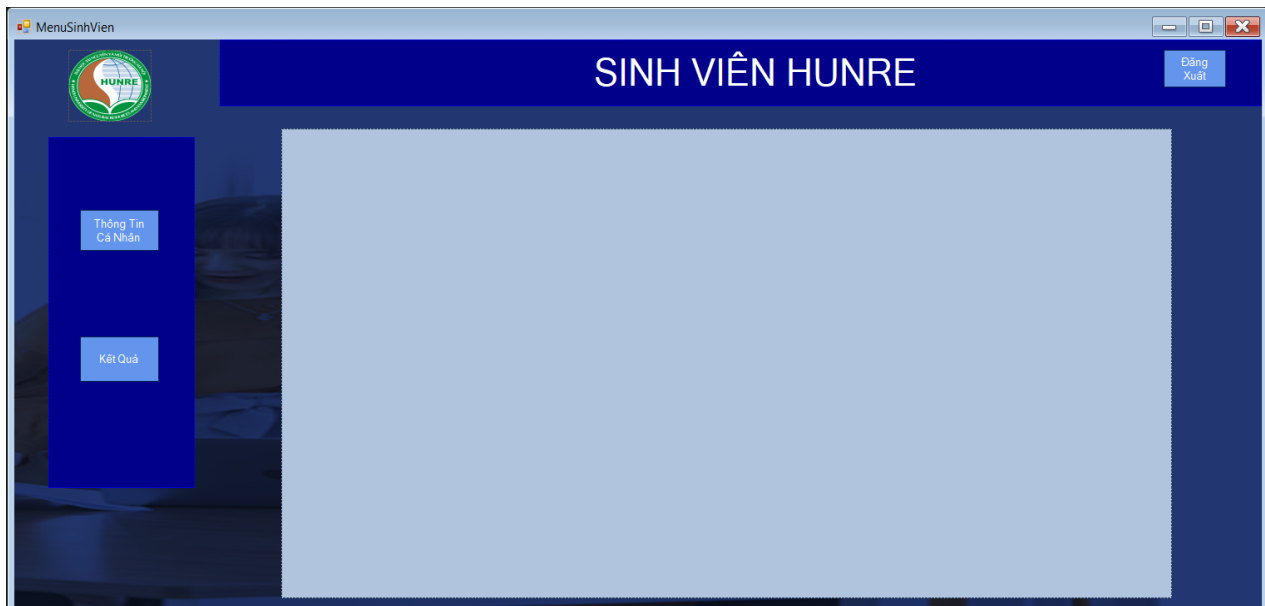
- Sinh Viên/Giảng Viên: Quản lý thông tin Sinh Viên/Giảng Viên
- Lớp: Quản lý những Lớp hành chính
- Môn học: Quản lý Môn học
- Ngành học: Quản lý ngành học
- Lớp học phần: Quản lý Lớp học phần



Hình 3.2.2. Form MENU ADMIN

3.3. Giao Diện Form MENU Sinh Viên

- Giao diện bao gồm 2 form con bao gồm Thông tin cá nhân và Kết Quả (Hiện thị kết quả học tập).



Hình 3.3.3. Form Sinh Viên

3.4. Giao Diện Form MENU Giảng Viên

- Giao diện có chức năng chấm điểm kết quả học tập Sinh Viên.

	maLopHocPhan	maSinhVien	hotenSinhVien	ngaySinh	tenMonHoc	diemSo	diemCh
▶	1	SV2019	Hiệp	7/25/2003	Toán rời rạc	10	A
	1	SV2025	Hiệp	4/15/2024	Toán rời rạc	9	A
	1	SV2026	Hiệp	4/15/2024	Toán rời rạc	8	B
	1	SV3019	Hiệp	4/16/2024	Toán rời rạc	7.5	B
	1	SV3023	Phượng	4/16/2024	Toán rời rạc	3	F
	1	SV3026	Sơn	4/16/2024	Toán rời rạc		
	1	SV3029	Hiếu	4/16/2024	Toán rời rạc		
	1	SV3030	An Vũ Sơn	7/25/2003	Toán rời rạc		

Hình 3.4.4. Form MENU Giảng Viên


```

void AddBi ndi ng()
{
    if ( dGVSi nhVi en. DataSou rce == null ) r eturn;
    el se
    {
        tbMLHP. DataBi ndi ngs. Add( " Text", dGVSi nhVi en. DataSou rce, " mǎLopHocPhan",
true, DataSou rceUpdat eMode. Never);
        tbMSV. DataBi ndi ngs. Add( " Text", dGVSi nhVi en. DataSou rce, " mǎSi nhVi en", true,
DataSou rceUpdat eMode. Never);
        tbDi ẽmSo. DataBi ndi ngs. Add( " Text", dGVSi nhVi en. DataSou rce, " di ẽmSo", true,
DataSou rceUpdat eMode. Never);
        tbDi ẽmChu. DataBi ndi ngs. Add( " Text", dGVSi nhVi en. DataSou rce, " di ẽmChu", true,
DataSou rceUpdat eMode. Never);
    }
}
void Cl ear Bi ndi ng()
{
    tbMLHP. DataBi ndi ngs. Cl ear();
    tbMSV. DataBi ndi ngs. Cl ear();
    tbDi ẽmSo. DataBi ndi ngs. Cl ear();
    tbDi ẽmChu. DataBi ndi ngs. Cl ear();
}
void LoadDat aDgvSi nhVi en()
{
    var resul t = from c in db. ThamGi aLopHoc
                  where c. LopHocPhan. Gi angVi en. mǎGi angVi en == mǎGV
                  select new
                  {
                      mǎLopHocPhan = c. LopHocPhan. mǎLopHocPhan,
                      mǎSi nhVi en = c. Si nhVi en. mǎSi nhVi en,
                      hot ẽnSi nhVi en = c. Si nhVi en. hot ẽnSi nhVi en,
                      ngaySi nh = c. Si nhVi en. ngaySi nh,
                      t ẽnMbnHoc = c. LopHocPhan. MbnHoc. t ẽnMbnHoc,
                      di ẽmSo = c. di ẽmSo,
                      di ẽmChu = c. di ẽmChu
                  };
    dGVSi nhVi en. DataSou rce = resul t. ToLi st();
}
Si nhVi en Sel ectIDSv(String mǎSV)
{
    Si nhVi en si nhVi en = db. Si nhVi en. Where(p => p. mǎSi nhVi en ==
mǎSV). Si ngl eOrDef aul t();
    r eturn si nhVi en;
}
LopHocPhan Sel ectIDHP(String mǎHocPhan)
{
    LopHocPhan l opHocPhan = db. LopHocPhan. Where(p => p. mǎLopHocPhan ==
mǎHocPhan). Si ngl eOrDef aul t();
    r eturn l opHocPhan;
}
void Edi tData(String mǎl hp, String mǎsv)
{
    i nt i dl hp = Sel ectIDHP(mǎl hp). i dLopHocPhan;
    i nt i dsv = Sel ectIDSv(mǎsv). i dSi nhVi en;
    ThamGi aLopHoc thamGi aLopHoc = db. ThamGi aLopHoc. Fi nd(i dl hp, i dsv);
    thamGi aLopHoc. di ẽmChu = tbDi ẽmChu. Text;
    thamGi aLopHoc. di ẽmSo = Convert. ToDoubl e(tbDi ẽmSo. Text);
}

```

3.5 Giao Diện Form Tài Khoản

	idNguoiDung	maSinhVien	matKhau	hotenSinhVien	ngaySinh	gioiTinh
▶	2	SV2019	2018SV2018	Hiệp	7/25/2003	Nam
	3	SV2025	2025SV2025	Hiệp	4/15/2024	Nam
	4	SV2026	2026SV2026	Hiệp	4/15/2024	Nam
	1003	SV3019	3019SV3019	Hiệp	4/16/2024	Nam
	1006	SV3023	3023SV3023	Phượng	4/16/2024	Nam
	1009	SV3026	hip	Sơn	4/16/2024	Nam
	1011	SV3029	hieune	Hiếu	4/16/2024	Nam
	2014	SV3030	3030SV3030	An Vũ Sơn	7/25/2003	Nam
	2015	SV3031	3031SV3031	An Vũ Sơn	7/25/2003	Nam

Hình 3.5.5. Form Tài Khoản

```
private void cbChucVu_SelectedValueChanged(object sender, EventArgs e)
{
    LoadData();
}

private void dgvThongTin_CellClick(object sender, DataGridViewCellEventArgs e)
{
    AddBi ndi ng();
    ClearBi ndi ng();
}

private void buttonUpdate_Click(object sender, EventArgs e)
{
    if (string.IsNullOrEmpty(tbiD.Text))
    {
        MessageBox.Show("Vui lòng chọn người dùng để chỉnh sửa!");
    }
    else if (string.IsNullOrEmpty(tbMatKhau.Text))
    {
        MessageBox.Show("Vui lòng điền mật khẩu muốn thay đổi!");
    }
    else
    {
        EditData(tbiD.Text);
        LoadData();
        ClearBi ndi ng();
        AddBi ndi ng();
    }
}
```

3.6. Giao Diện Form Sinh Viên

Quản lý sinh viên

Thông tin chi tiết

Mã sinh viên

Họ và tên

Ngày sinh

5/ 5/2024

Giới tính

☐ Nam ☐ Nữ

Năm nhập học

2024

Chức Năng

Thêm mới

Cập nhật

Xóa

Reset

Danh sách sinh viên

Lớp

C4

	maSinhVien	hotenSinhVien	maLop	ngaySinh	gioiTinh
▶	SV2019	Hiệp	C4	7/25/2003	Nam
	SV2026	Hiệp	C4	4/15/2024	Nam
	SV3019	Hiệp	C4	4/16/2024	Nam
	SV3023	Phường	C4	4/16/2024	Nam
	SV3026	Sơn	C4	4/16/2024	Nam
	SV3029	Hiếu	C4	4/16/2024	Nam
	SV3030	An Vũ Sơn	C4	7/25/2003	Nam

Hình 3.6.6. Form Sinh Viên

```

void AddData()
{
    string selectedValue = null;
    foreach (RadioButton radioButton in panel Gi oi Ti nh. Controls)
    {
        if (radioButton.Checked)
        {
            selectedValue = radioButton.Text;
            break;
        }
    }
    Lop lop = db. Lops. Where(p => p.mãLop == cbLop. SelectedValue). SingleOrDefault();
    SinhVien sinhVien = new SinhVien()
    {
        hotenSinhVien = tbHoTen.Text,
        ngaySinh = dTPNgaySinh.Value.Date,
        gi oi Ti nh = selectedValue,
        namNhapHoc = dTPNam.Value.Year,
        idLop = Convert.ToInt32(lop.idLop)
    };

    try
    {
        db. SinhVien. Add(sinhVien);
        db. SaveChanges();

        SinhVien sinhVienUpdate = db. SinhVien. Find(sinhVien.idSinhVien);
        sinhVienUpdate.mãSinhVien = ("SV" + sinhVien.idSinhVien). ToString();
        db. SaveChanges();

        AddAcc(sinhVien);
    }
    catch (Exception ex)
    {
        MessageBox.Show("Thêm sinh viên thất bại! Lỗi: " + ex.Message);
    }
}

void EditData(String mãSinhVien)
{
    SinhVien sinhVien = db. SinhVien. Find(SelectData(mãSinhVien). idSinhVien);
    if (sinhVien != null)
    {
        string selectedValue = null;
        foreach (RadioButton radioButton in panel Gi oi Ti nh. Controls)
        {
            if (radioButton.Checked)
            {
                selectedValue = radioButton.Text;
                break;
            }
        }
        sinhVien.hotenSinhVien = tbHoTen.Text;
        sinhVien.ngaySinh = dTPNgaySinh.Value.Date;
        sinhVien.gi oi Ti nh = selectedValue.ToString();
        sinhVien.namNhapHoc = dTPNam.Value.Year;
        try
        {

```

3.7. Giao Diện Form Giảng Viên

Thông tin chi tiết

Mã giảng viên

Họ và tên

Ngày sinh

05/05/2024

Giới tính

☐ Nam

☐ Nữ

Chức Năng

Thêm mới

Cập nhật

Xóa

Reset

Quản lý giảng viên

Danh sách giảng viên

	maGiangVien	hotenGiangVien	ngaySinh	gioiTinh
▶	GV1	San	2/3/1991	Nam
	GV2	Khải	4/4/1996	Nam
	GV3	Thuận	7/23/1990	Nam
	GV4	Anh	6/12/1989	Nữ
	GV5	Liêm	1/17/2000	Nữ
	GV6	Sang	4/16/2024	Nam
	GV9	Trần	4/17/2024	Nữ

Hình 3.7.7. Form Quản Lý Giảng Viên

```

void AddData()
{
    string selectedValue = null;

    // Lặp qua tất cả các control trong Panel
    foreach (RadioButton radioButton in panel Gi oi Ti nh. Controls)
    {
        // Kiểm tra nếu control là RadioButton
        if (radioButton.Checked)
        {
            // Lấy giá trị của RadioButton được chọn
            selectedValue = radioButton.Text;
            break; // Thoát khỏi vòng lặp sau khi đã tìm thấy RadioButton được chọn
        }
    }

    Gi angVi en gi angVi en = new Gi angVi en()
    {
        hotenGi angVi en = tbHoTen.Text,
        ngaySi nh = dTPNgaySi nh.Value.Date,
        gi oi Ti nh = selectedValue.ToString()
    };

    try
    {
        db.Gi angVi en.Add(gi angVi en);
        db.SaveChanges();

        Gi angVi en gi angVi enUpdate = db.Gi angVi en.Find(gi angVi en.i dGi angVi en);
        gi angVi enUpdate.maGi angVi en = ("GV" + gi angVi en.i dGi angVi en).ToString();
        db.SaveChanges();

        AddAcc(gi angVi en);
    }
    catch (Exception ex)
    {
        MessageBox.Show("Thêm gi ả ng vi ên thất bại! Lỗi: " + ex.Message);
    }
}

Gi angVi en SelectData(String maGi angVi en)
{
    Gi angVi en gi angVi en = db.Gi angVi en.Where(p => p.maGi angVi en ==
maGi angVi en).Si ngl eOrDefaul t();
    return gi angVi en;
}

void EditData(String maGi angVi en)
{
    Gi angVi en gi angVi en = db.Gi angVi en.Find(SelectData(maGi angVi en).i dGi angVi en);
    if (gi angVi en != null)
    {
        string selectedValue = null;
        foreach (RadioButton radioButton in panel Gi oi Ti nh. Controls)
        {
            if (radioButton.Checked)
            {
                selectedValue = radioButton.Text;
            }
        }
    }
}

```

3.8. Giao Diện Form Lớp

Thông tin

Mã lớp

Mã ngành

Mã giảng viên

CNTT

Thêm

Sửa

Xóa

Reset

Danh sách giảng viên

	maGiangVien	hotenGiangVien	ni
▶	GV1	San	2/
	GV2	Khải	4/
	GV3	Thuận	7/
	GV4	Anh	6/
	GV5	Liêm	1/
	GV6	Sang	4/
	GV9	Trần	4/

Danh sách lớp

	maLop	tenNganhHoc	maGiangVien	hotenGiangVien
▶	CNTT2003	Công Nghệ Thôn...	GV9	Trần

Hình 3.8.8. Form Lớp

```

void AddData()
{
    try
    {
        Lop lop = new Lop()
        {
            idNganhHoc = SelectIdNganh(),
            idGiangVien = SelectIdGiangVien()
        };
        db.Lop.Add(lop);
        db.SaveChanges();

        Lop lopUpdate = db.Lop.Find(lop.idLop);
        lopUpdate.maLop = cbMaNganh.SelectedVa lue.ToString() + lopUpdate.idLop;
        db.SaveChanges();
        MessageBox.Show("Thêm lớp thành công.");
    }
    catch (System.Data.Entity.Infrastructure.DbUpdateException ex)
    {
        MessageBox.Show("Lỗi khi thêm lớp: " + ex.Message);
    }
}

void EditData()
{
    try
    {
        Lop lop = db.Lop.Find(SelectIdLop());
        lop.idGiangVien = SelectIdGiangVien();
        lop.idNganhHoc = SelectIdNganh();
        db.SaveChanges();
        MessageBox.Show("Sửa thông tin lớp thành công.");
    }
    catch (System.Data.Entity.Infrastructure.DbUpdateException ex)
    {
        MessageBox.Show("Lỗi khi thêm lớp: " + ex.Message);
    }
}

void DeleteData()
{
    try
    {
        if (!db.SinhVien.Any(p => p.idLop == SelectIdLop()))
        {
            Lop lopDel = db.Lop.Find(SelectIdLop());
            db.Lop.Remove(lopDel);
            db.SaveChanges();
            MessageBox.Show("Xóa lớp thành công.");
        }
        else
        {
            MessageBox.Show("Không thể xóa lớp có sinh viên");
        }
    }
    catch (System.Data.Entity.Infrastructure.DbUpdateException ex)
    {

```


3.9. Giao Diện Form Môn Học

Quản lý Môn Học

Thông tin chi tiết

Mã Môn Học

2

Tên Môn Học

Toán rời rạc

Số Tín Chi

3

Năm Bắt Đầu

2024

Quy Định

☐ Bắt buộc ☐ Tự chọn

Mô Tả Môn Học

"Chưa có mô tả"

Hình Thức Đánh Giá

Bài tập lớn

Chức Năng

Thêm mới

Cập nhật

Xóa

Reset

Danh Sách Môn Học

	maMonHoc	tenMonHoc	soTinChi	namBatDau	quyDinh
▶	2	Toán rời rạc	3	2020	BB
	3	Kinh tế vĩ mô	3	2020	BB
	4	Lập trình Android	3	2020	BB
	5	Lập trình WEB	3	2020	BB
	CNNET	Công Nghệ NET	3	2024	Bắt buộc
	KT	Kiểm Thử	3	2024	Bắt buộc
	QLD	Quản Lý Dự án	3	2019	Bắt buộc

Hình 3.9.9. Form Môn Học

```

void AddData()
{
    string selectedValue = null;

    foreach (RadioButton radioButton in panel QuyDnh.Controls)
    {
        if (radioButton.Checked)
        {
            selectedValue = radioButton.Text;
            break;
        }
    }

    MonHoc monHoc = new MonHoc()
    {
        maMonHoc = new string(tbTenMH.Text.Where(char.IsUpper).ToArray()),
        tenMonHoc = tbTenMH.Text,
        soTinChi = Convert.ToInt32(tbSoTC.Text),
        namBatDau = dTPNanBD.Value.Year,
        quyDinh = selectedValue,
        moTaMonHoc = tbMoTaMH.Text,
        hinHThucDanhGiã = db.HinHThuc.Where(p => p.tenHinHThucDanhGiã ==
cbHTDG.SelectedVal ue).Select(p => p.hinHThucDanhGiã).SingleOrDefault()
    };

    try
    {
        if (db.MonHoc.Where(p => p.maMonHoc == monHoc.maMonHoc).Any())
        {
            MessageBox.Show("Mã môn học này đã tồn tại");
            ResetData();
        }
        else
        {
            db.MonHoc.Add(monHoc);
            db.SaveChanges();

            MessageBox.Show("Thêm Môn Học Thành Công!");
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("Thêm Môn Học thất bại! Lỗi: " + ex.Message);
    }
}

void EditData(string maMonHoc)
{
    MonHoc monHoc = db.MonHoc.Find(SelectData(maMonHoc).idMonHoc);

    string selectedValue = null;

    foreach (RadioButton radioButton in panel QuyDnh.Controls)
    {
        if (radioButton.Checked)
        {
            selectedValue = radioButton.Text;
            break;
        }
    }
}

```

3.10. Giao Diện Form Ngành Học

Quản Lý Ngành Học

Thông tin

Mã ngành học

Tên ngành học

Số tín chỉ

Năm bắt đầu 2024

Chức năng

Thêm Sửa

Xoá Reset

Phân chia môn học

Danh sách

	maNganhHoc	tenNganhHoc	soTinChi	namBatDau
▶	KT	Công Nghệ T...	9	2020
	QLDD	Quản lý đất đai	9	2020
	QLKD	Quản lý kinh ...	18	2020
	CNTT	Công Nghệ T...	21	2020
	QTDL	QTDL		2024

Hình 3.10.10. Form Quản Lý Ngành Học

```

void AddData()
{
    NganhHoc nganhHoc = new NganhHoc()
    {
        maNganhHoc = new string(tbTenNH.Text.Where(char.IsUpper).ToArray()),
        tenNganhHoc = tbTenNH.Text,
        namBatDau = dTPNam.Value.Date.Year,
    };
    if (db.NganhHoc.Any(p => p.maNganhHoc == nganhHoc.maNganhHoc))
    {
        MessageBox.Show("Mã ngành này đã bị trùng, vui lòng đổi tên ngành!");
    }
    else
    {
        try
        {
            db.NganhHoc.Add(nganhHoc);
            db.SaveChanges();
            MessageBox.Show("Thêm ngành học thành công!");
        }
        catch (Exception ex)
        {
            MessageBox.Show("Thêm ngành học thất bại! Lỗi: " + ex.Message);
        }
    }
}

void EditData(String maNganhHoc)
{
    NganhHoc nganhHoc = db.NganhHoc.Find(SelectData(maNganhHoc).idNganhHoc);
    nganhHoc.tenNganhHoc = tbTenNH.Text;
    nganhHoc.maNganhHoc = new
string(tbTenNH.Text.Where(char.IsUpper).ToArray());
    nganhHoc.namBatDau = dTPNam.Value.Date.Year;
    if (db.NganhHoc.Any(p => p.maNganhHoc == nganhHoc.maNganhHoc))
        MessageBox.Show("Mã ngành này đã bị trùng, vui lòng đổi tên ngành!");
    else
    {
        try
        {
            db.SaveChanges();
            MessageBox.Show("Sửa thông tin ngành học thành công!");
        }
        catch (Exception ex)
        {
            MessageBox.Show("Sửa thông tin ngành học thất bại! Lỗi: " +
ex.Message);
        }
    }
}

void DeleteData(String maNganhHoc)
{
    if (db.Lop.Any(p => p.NganhHoc.maNganhHoc == maNganhHoc))
        MessageBox.Show("Ngành học " + maNganhHoc + " đã có lớp học, không thể
xóa!");
    else if (db.PhanChi aMbnHoc.Any(p => p.NganhHoc.maNganhHoc == maNganhHoc))

```

3.11. Giao Diện Form Lớp Học Phần

Thông tin chi tiết

Mã Lớp Học Phần

Năm Học

Học Kỳ

Mô tả

Giới hạn sinh viên

Tên môn học

Tên giáo viên

Thêm mới

Cập nhật

Xóa

Reset

Tham gia học phần

Danh Sách Lớp Học Phần

	maLopHocPhan	namHoc	moTa	hocKy	tenMonHoc
▶	1	2020	"Chưa có mô tả"	1	Toán rời rạc
	2	2020	"Chưa có mô tả"	2	Kinh tế vĩ mô
	3	2021	"Chưa có mô tả"	1	Lập trình Android

Hình 3.11.11. Form Lớp Học Phần

```

void AddData()
{
    int idgv = SelectIdGiangVien(cbTenGV. SelectedValue. ToString());
    int idmh = SelectIdMonHoc(cbTenMH. SelectedValue. ToString());
    LopHocPhan lopHocPhan = new LopHocPhan()
    {
        nanHoc = Convert.ToInt32(tbNanHoc. Text),
        maTa = tbMaTa. Text,
        hocky = Convert.ToInt32(tbHocky. Text),
        gioidanhSinhVien = Convert.ToInt32(tbGioidanhSV. Text),
        idGiangVien = idgv,
        idMonHoc = idmh
    };

    try
    {
        db.LopHocPhan. Add(lopHocPhan);
        db.SaveChanges();

        String manh = db.MonHoc. Where(p => p.idMonHoc == idmh). Select(p =>
p.maMonHoc). FirstOrDefault();
        LopHocPhan lopHocPhanUpdate = db.LopHocPhan. Find(lopHocPhan.idLopHocPhan);
        lopHocPhanUpdate.maLopHocPhan = lopHocPhan.nanHoc. ToString() + manh;
        db.SaveChanges();

        MessageBox.Show("Thêm Lớp Học Phần Thành Công!");
    }
    catch (Exception ex)
    {
        MessageBox.Show("Thêm Lớp Học Phần thất bại! Lỗi: " + ex.Message);
    }
}

```

```

void EditData(String maLopHocPhan)
{
    int idlhp = SelectIdLopHocPhan(maLopHocPhan);
    int idgv = SelectIdGiangVien(cbTenGV. SelectedValue. ToString());
    int idmh = SelectIdMonHoc(cbTenMH. SelectedValue. ToString());

    LopHocPhan lopHocPhan = db.LopHocPhan. Find(idlhp);

    if (lopHocPhan != null)
    {
        lopHocPhan.maTa = tbMaTa. Text;
        lopHocPhan.hocky = Convert.ToInt32(tbHocky. Text);
        lopHocPhan.nanHoc = Convert.ToInt32(tbNanHoc. Text);
        lopHocPhan.gioidanhSinhVien = Convert.ToInt32(tbGioidanhSV. Text);
        lopHocPhan.idMonHoc = idmh;
        lopHocPhan.idGiangVien = idgv;
        try
        {
            db.SaveChanges();
            MessageBox.Show("Sửa thông tin Lớp Học Phần thành công!");
        }
        catch (Exception ex)
        {

```

3.12. Giao Diện Form Kết Quả Sinh Viên

Thông Tin Cá Nhân

Mã SV:

SV2025

Họ Và Tên:

Hiệp

Ngày Sinh:

2024-04-15

Ngành Học:

Công Nghệ Thông Tin

Lớp Học:

KT1001

Giới Tính:

Nam

Tổng Số Tín Chỉ

Tổng Số Tín:

9

Tín Chỉ Tích Lũy:

3

Bảng Điểm

	maLopHocPhan	tenMonHoc	soTinChi	hotenGiangVien	hinhThucDanhGia	diemSo
▶	1	Toán rời rạc	3	San	TN	9

Hình 3.12.12. Form Kết Quả Sinh Viên

```

void LoadDataDGV()
{
    var result = from c in db.ThamGiaLopHoc
                  where c.SinhVien.maSinhVien == idSV
                  select new
                  {
                      maLopHocPhan = c.LopHocPhan.maLopHocPhan,
                      tenMonHoc = c.LopHocPhan.MonHoc.tenMonHoc,
                      soTinChi = c.LopHocPhan.MonHoc.soTinChi,
                      hotenGiangVien = c.LopHocPhan.GiangVien.hotenGiangVien,
                      hinhtucDanhGia = c.LopHocPhan.MonHoc.hinhtucDanhGia,
                      diemSo = c.diemSo,
                      diemChu = c.diemChu,
                      moTa = c.LopHocPhan.moTa

                  };
    DGVKetQua.DataSource = result.ToList();

    int tongSoTinChi = (int) result.Sum(c => c.soTinChi);
    tbTSTL.Text = tongSoTinChi.ToString();

    int tongSoTinChiTichLuy = ((int) db.SinhVien.Where(p => p.maSinhVien ==
idSV).Select(p => p.Lop.NganhHoc.soTinChi).SingleOrDefault());
    tbTST.Text = tongSoTinChiTichLuy.ToString();

}

void LoadGroupBoxTTCN()
{
    String maSinhVien = db.SinhVien.Where(p => p.maSinhVien == idSV).Select(p =>
p.maSinhVien).SingleOrDefault();
    tbMaSV.Text = maSinhVien;
    String hotenSinhVien = db.SinhVien.Where(p => p.maSinhVien == idSV).Select(p =>
p.hotenSinhVien).SingleOrDefault();
    tbHoTen.Text = hotenSinhVien;
    String ngaysinh = db.SinhVien.Where(p => p.maSinhVien == idSV).Select(p =>
p.ngaysinh.Value.ToString()).SingleOrDefault();
    tbNgaySinh.Text = ngaysinh;
    String gioitinh = db.SinhVien.Where(p => p.maSinhVien == idSV).Select(p =>
p.gioitinh).SingleOrDefault();
    tbGioiTinh.Text = gioitinh;
    String lop = db.SinhVien.Where(p => p.maSinhVien == idSV).Select(p =>
p.Lop.maLop).SingleOrDefault();
    tbLopHoc.Text = lop;
    String nganhhoc = db.SinhVien.Where(p => p.maSinhVien == idSV).Select(p =>
p.Lop.NganhHoc.tenNganhHoc).SingleOrDefault();
    tbNganhHoc.Text = nganhhoc;
}

private void dataGridView1_CellContentClick_1(object sender,
DataGridViewCellEventArgs e)
{
    LoadDataDGV();
}

```


3.13. Code thêm, sửa, xóa

- Phần mềm sử dụng framework EntityFramework với những câu lệnh LINQ để có thể linh hoạt chuyển đổi dữ liệu từ cơ sở dữ liệu thành model

```
void AddData()
{
    string selectedValue = null;
    foreach (RadioButton radioButton in panel Gi oi Ti nh. Controls)
    {
        if (radioButton.Checked)
        {
            selectedValue = radioButton.Text;
            break;
        }
    }
    Lop lop = db.Lop.Where(p => p.maLop == cbLop.SelectedValue).SingleOrDefault();
    SinhVien sinhVien = new SinhVien()
    {
        hotenSinhVien = tbHoTen.Text,
        ngaySinh = dTPNgaySinh.Value.Date,
        gi oi Ti nh = selectedValue,
        namNhapHoc = dTPNam.Value.Year,
        idLop = Convert.ToInt32(lop.idLop)
    };

    try
    {
        db.SinhVien.Add(sinhVien);
        db.SaveChanges();

        SinhVien sinhVienUpdate = db.SinhVien.Find(sinhVien.idSinhVien);
        sinhVienUpdate.maSinhVien = ("SV" + sinhVien.idSinhVien).ToString();
        db.SaveChanges();

        AddAcc(sinhVien);
    }
    catch (Exception ex)
    {
        MessageBox.Show("Thêm sinh viên thất bại! Lỗi: " + ex.Message);
    }
}

void EditData(String maSinhVien)
{
    SinhVien sinhVien = db.SinhVien.Find(SelectData(maSinhVien).idSinhVien);
    if (sinhVien != null)
    {
        string selectedValue = null;
        foreach (RadioButton radioButton in panel Gi oi Ti nh. Controls)
        {
            if (radioButton.Checked)
            {
                selectedValue = radioButton.Text;
                break;
            }
        }
        sinhVien.hotenSinhVien = tbHoTen.Text;
        sinhVien.ngaySinh = dTPNgaySinh.Value.Date;
        sinhVien.gi oi Ti nh = selectedValue.ToString();
        sinhVien.namNhapHoc = dTPNam.Value.Year;
        try
        {
            db.SinhVien.Update(sinhVien);
            db.SaveChanges();
        }
        catch (Exception ex)
        {
            MessageBox.Show("Sửa sinh viên thất bại! Lỗi: " + ex.Message);
        }
    }
}
```

Code đăng nhập FormLogin

```

if (db.NguoiDung.Where(tk => tk.SinhVien.maSinhVien == tbUserName.Text && tk.matKhau == tbPassWord.Text).SingleOrDefault() != null)
{
    MessageBox.Show("Đăng nhập thành công");
    // Nếu thông tin đăng nhập chính xác, đóng form đăng nhập và mở form menu chính
    this.Hide();
    MenuSinhVien menuSV = new MenuSinhVien(tbUserName.Text);
    menuSV.ShowDialog();
    this.Close();
}
else if (db.NguoiDung.Where(tk => tk.GiangVien.maGiangVien == tbUserName.Text && tk.matKhau == tbPassWord.Text).SingleOrDefault() != null)
{
    MessageBox.Show("Đăng nhập thành công");
    // Nếu thông tin đăng nhập chính xác, đóng form đăng nhập và mở form menu chính
    this.Hide();
    KetQuaHocPhanForm menuGV = new KetQuaHocPhanForm(tbUserName.Text);
    menuGV.ShowDialog();
    this.Close();
}
else if (db.NguoiDung.Where(tk => tk.matKhau == tbPassWord.Text && tk.chucVu == tbUserName.Text).SingleOrDefault() != null && tbUserName.Text == "ADM N")
{
    MessageBox.Show("Đăng nhập thành công");
    // Nếu thông tin đăng nhập chính xác, đóng form đăng nhập và mở form menu chính
    this.Hide();
    MenuADM N menuADM N = new MenuADM N();
    menuADM N.ShowDialog();
    this.Close();
}
else if (tbPassWord == null || tbUserName == null)
{
    MessageBox.Show("Vui lòng điền đầy đủ thông tin");
    tbUserName.Clear();
    tbPassWord.Clear();
}
else
{
    MessageBox.Show("Tài khoản hoặc mật khẩu không chính xác");
    tbUserName.Clear();
    tbPassWord.Clear();
}

private void button1_Click(object sender, EventArgs e)
{
    tbPassWord.UseSystemPasswordChar = !tbPassWord.UseSystemPasswordChar;
}

```

4. CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

4.1. Kết luận

Kết luận Sau quá trình nghiên cứu tìm hiểu xây dựng website em đã đạt được các kết quả như sau:

- Nghiên cứu và xây dựng thành công phần mềm quản lý Sinh Viên
- Đáp ứng đầy đủ mọi chức năng đã phân tích
- Hoàn thành báo cáo nghiên cứu của đề tài khóa luận. Báo cáo đã nêu đầy đủ các nội dung về công cụ, ngôn ngữ phục vụ cho bài toán, phân tích và thiết kế các biểu đồ, sơ đồ phù hợp và phần mềm quản lý phần mềm quản lý Sinh Viên
- Phần mềm đang trong quá trình thử nghiệm, phát triển và bản thân em cũng chưa có nhiều kinh nghiệm nên vẫn có nhiều thiếu sót rất mong thầy cô góp ý để em có thể hoàn thiện sản phẩm một cách hoàn chỉnh hơn

4.2. Hướng phát triển

Trong thời gian tới, em sẽ nghiên cứu sâu hơn về hệ thống để hoàn thiện và sửa chữa các chức năng đồng thời cũng tìm ra các nhược điểm của hệ thống nhằm khắc phục các khó khăn của hệ thống :

5. CHƯƠNG 5: TÀI LIỆU THAM KHẢO

5.1. Tiếng Việt

[1] Đâu Quang Tuấn, Thiết kế trang web bằng Visual studio(winform 2022), NXB Giao thông vận tải

[2] Nguyễn Trường Sinh, Thiết Kế Trang Web Tương Tác Bằng C# (2001),
NXB Giáo Dục

[3] <https://hoanghamobile.com/gioi-thieu-cong-ty>

5.2. Tiếng Anh

[4] <https://expressjs.com/>

- [5] <https://nodejs.org/en>
- [6] <https://code.visualstudio.com/>
- [7] <https://vi.m.wikipedia.org/wiki>