

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

**KHOA CÔNG NGHỆ THÔNG TIN**

**-----🙞🙞🙞🙞🙞-----**



**ĐỀ TÀI CUỐI KỲ MÔN HỆ QUẢN TRỊ CSDL**

**TÊN ĐỀ TÀI: XÂY DỰNG HỆ THỐNG QUẢN LÝ ĐIỂM**

**GVHD:** TS. Nguyễn Thành Sơn

**Lớp HP:** DBMS330284\_01

**SV thực hiện:** 23110210 – Ninh Thị Mỹ Hạnh - Nhóm 5

**Học kỳ:** 1

**Năm học:** 2025 - 2026

*Thành phố Hồ Chí Minh, tháng 9, năm 2025*

**SINH VIÊN THỰC HIỆN**

HỌC KÌ 1 NĂM HỌC 2025 - 2026

**Nhóm 5**

**Đề tài:** Quản lý điểm

|  |  |
| --- | --- |
| **Họ tên sinh viên** | **Mã số sinh viên** |
| Ninh Thị Mỹ Hạnh | 23110210 |

**NHẬN XÉT CỦA GV**

*Tp. Hồ Chí Minh, ngày …., tháng..., năm 2025*

**Giảng viên chấm điểm**

**MỤC LỤC**

[CHƯƠNG 1: ĐẶC TẢ HỆ THỐNG QUẢN LÝ ĐIỂM 3](#_Toc209969645)

[1.Đặc tả hệ thống 3](#_Toc209969646)

[1.1. Mô tả bài toán 3](#_Toc209969647)

[1.2 Mô tả dữ liệu 3](#_Toc209969648)

[1.3 Mô tả chức năng 4](#_Toc209969649)

[2. Giao diện hệ thống 4](#_Toc209969650)

[2.1. Giao diện đăng nhập (giao diện chung) 4](#_Toc209969651)

[2.2. Giao diện đối với giảng viên 4](#_Toc209969652)

[2.3. Giao diện đối với quản trị viên 5](#_Toc209969653)

[3. Công thức tính điểm 5](#_Toc209969654)

[CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 7](#_Toc209969655)

[1. Thiết kế cơ sở dữ liệu mức quan niệm 7](#_Toc209969656)

[2. Thiết kế cơ sở dữ liệu mức logic 10](#_Toc209969657)

[3. Các ràng buộc 10](#_Toc209969658)

[4. Cài đặt CSDL và các ràng buộc 12](#_Toc209969659)

[5. Lược đồ quan hệ sau khi cài đặt SQL 16](#_Toc209969660)

[6. Các triggers 18](#_Toc209969661)

[7. Các hàm 19](#_Toc209969662)

[CHƯƠNG 3: THIẾT KẾ CÁC CHỨC NĂNG 23](#_Toc209969663)

[1. Kết nối cơ sở dữ liệu 23](#_Toc209969664)

[2. Chức năng đăng nhập 28](#_Toc209969665)

[3. Chức năng hiển thị 31](#_Toc209969666)

[4. Chức năng quản lý 44](#_Toc209969667)

[5. Thống kê 63](#_Toc209969668)

[CHƯƠNG 4: TẠO USER VÀ PHÂN QUYỀN 70](#_Toc209969669)

[1. Tạo Role và phân quyền 70](#_Toc209969670)

[1.1. Giảng viên 70](#_Toc209969671)

[1.2. Quản trị viên (Admin) 70](#_Toc209969672)

[CHƯƠNG 5: THIẾT KẾ GIAO DIỆN HỆ THỐNG 71](#_Toc209969673)

[1.Kiến trúc hệ thống 71](#_Toc209969674)

[2. Giao diện của chương trình: 71](#_Toc209969675)

[2.1. Form đăng nhập 71](#_Toc209969676)

[2.2. Form admin 72](#_Toc209969677)

[2.3 Form giảng viên 76](#_Toc209969678)

[CHƯƠNG 6: KẾT LUẬN 80](#_Toc209969679)

# CHƯƠNG 1: ĐẶC TẢ HỆ THỐNG QUẢN LÝ ĐIỂM

## 1.Đặc tả hệ thống

### 1.1. Mô tả bài toán

Hệ thống quản lý điểm học sinh được xây dựng nhằm giải quyết vấn đề tổ chức, lưu trữ và xử lý dữ liệu điểm số một cách nhanh chóng, minh bạch và hiệu quả. Trong hệ thống có hai nhóm đối tượng chính.

Quản trị viên (Admin) là người có quyền hạn cao nhất, chịu trách nhiệm quản lý toàn bộ hoạt động và người dùng. Admin có thể cấu hình hệ thống, phân quyền cho giảng viên, thêm, xóa, sửa thông tin giảng viên và sinh viên, theo dõi thông tin lớp học phần, đăng ký học phần cũng như trực tiếp quản lý điểm sinh viên.

Giảng viên là nhóm người dùng trực tiếp thực hiện các nghiệp vụ liên quan đến điểm số. Họ có nhiệm vụ quản lý điểm của các lớp học phần mình giảng dạy, nhập và cập nhật điểm thành phần, điểm thi cuối kỳ, thống kê kết quả giảng dạy và có thể điều chỉnh sinh viên giữa các lớp học phần khi cần thiết.

Mục tiêu của hệ thống là số hóa và tự động hóa toàn bộ quy trình quản lý điểm, từ khâu quản lý thông tin học sinh, hỗ trợ nhập và xử lý điểm, cho đến báo cáo và phân tích số liệu. Việc triển khai hệ thống sẽ góp phần giảm thiểu sai sót do thao tác thủ công, nâng cao hiệu quả công việc của giáo viên, đồng thời đảm bảo tính minh bạch, chính xác trong công tác quản lý kết quả học tập.

### 1.2 Mô tả dữ liệu

Dữ liệu trong hệ thống quản lý điểm tập trung vào ba nhóm chính: thông tin người dùng, thông tin học tập và dữ liệu điểm số.

Thông tin sinh viên bao gồm mã số sinh viên, họ tên, ngày sinh, nơi sinh, giới tính, lớp, trạng thái. Đây là cơ sở dữ liệu nền tảng phục vụ cho việc quản lý kết quả học tập và liên kết với các học phần.

Thông tin giảng viên bao gồm mã giảng viên, họ tên, học vị, khoa, thông tin liên hệ và danh sách lớp học phần giảng dạy. Việc lưu trữ dữ liệu giảng viên giúp phân công giảng dạy chính xác và thuận tiện trong việc quản lý điểm theo từng lớp học phần.

Dữ liệu môn học và lớp học phần bao gồm mã môn học, tên môn học, số tín chỉ, hình thức đánh giá và thông tin lớp học phần cụ thể trong từng học kỳ. Lớp học phần được liên kết với cả giảng viên giảng dạy và danh sách sinh viên đăng ký, từ đó bảo đảm sự thống nhất trong quản lý đào tạo.

Cuối cùng là dữ liệu điểm số, bao gồm điểm giữa kỳ, điểm thi cuối kỳ và điểm tổng kết. Dữ liệu này được nhập và cập nhật trực tiếp bởi giáo viên, được liên kết chặt chẽ với thông tin sinh viên và lớp học phần. Nhờ đó, hệ thống có thể hỗ trợ việc tra cứu, thống kê và báo cáo kết quả học tập một cách nhanh chóng, chính xác và minh bạch.

### 1.3 Mô tả chức năng

*Chức năng Đăng nhập:* Mỗi người dùng có tài khoản riêng để đăng nhập và truy cập các chức năng tương ứng với quyền hạn của mình

*Chức năng quản lý sinh viên:* cho phép thêm mới, chỉnh sửa và cập nhật thông tin sinh viên, đồng thời hỗ trợ theo dõi tiến độ học tập của từng sinh viên trong suốt quá trình đào tạo.

*chức năng quản lý giảng viên:* Tương tự như chức năng sinh viên cho phép lưu trữ và điều chỉnh thông tin giảng viên.

*Chức năng theo dõi danh sách lớp học phần và đăng ký của sinh viên:* Chức năng theo dõi kết quả đăng ký và danh sách lớp học từ phần mềm đăng ký môn học (thoả điều kiện của phần mềm quản lý điểm).

*Chức năng nhập điểm và xử lý điểm:* là chức năng nghiệp vụ cốt lõi. Giảng viên nhập điểm giữa kỳ và điểm thi cuối kỳ, sau đó hệ thống sẽ tự động tính toán điểm tổng kết. Các kết quả này được lưu trữ và đồng bộ, cho phép xuất báo cáo thống kê hoặc phân tích kết quả học tập.

*Hệ thống còn có chức năng báo cáo – thống kê*: cho phép quản trị viên và giảng viên tạo các báo cáo tổng hợp về tình hình học tập của sinh viên, tỷ lệ đạt, tỷ lệ trượt, phân bố điểm. Những dữ liệu này giúp đưa ra quyết định kịp thời để nâng cao chất lượng giảng dạy.

*Chức năng Quản lý Người dùng:* Đây là phần chức năng thuộc về quyền quản trị, chỉ có người dùng được cấp quyền quản trị mới có quyền thao tác, quản lý mục này. Phần này sẽ lưu thông tin về tài khoản, mật khẩu cũng như các thông tin liên quan của người dùng, phục vụ quản lý người dùng như: tạo mới hoặc đổi mật khẩu cho người dùng nếu quên.

*Chức năng Cấu hình hệ thống:* Tương tự như quản lý người dùng, chức năng này chỉ có quản trị viên mới có quyền truy cập. Nơi mà quản trị viên có thể cấu hình các thông số cơ bản về số tín chỉ hay thông số trọng số thành phần điểm, ngoài ra còn có danh sách Nhật ký hệ thống nhằm kiểm tra, kiểm soát các hành vi đăng nhập người dùng.

## 2. Giao diện hệ thống

### 2.1. Giao diện đăng nhập (giao diện chung)

Giao diện đăng nhập là formDangNhap.

Người dùng chọn quyền rồi đăng nhập đúng với tài khoản quyền của mình.

### 2.2. Giao diện đối với giảng viên

Giao diện giảng viên là formGiangVien chứa các danh mục sau:

*Danh mục trang cá nhân:* gồmcác thông tin cơ bản của giảng, thông tin về mã giảng viên, họ tên, học vị, khoa, email, số điện thoại. Giảng viên chỉ có thể chỉnh sửa thông tin liên lạc.

*Danh mục danh sách giảng dạy:* có thể xem danh sách giảng của mình dạy qua các năm

*Danh mục danh sách sinh viên:* Xem danh sách sinh viên theo lớp học phần của năm hiện tại. Có thể chuyển giao sinh viên giữa hai lớp cùng môn học mình giảng dạy để dễ theo dõi quản lý điểm sinh viên khi có nhu cầu chuyển lớp.

*Danh mục danh quản lý điểm:* Nhập và cập nhập điểm giữa kỳ và cuối kỳ của sinh viên theo lớp học. Thống kê biểu đồ tỉ lệ đạt rớt cũng như theo phổ điểm của lớp học phần đó.

### 2.3. Giao diện đối với quản trị viên

Giao diện giảng viên là formAdmin chứa các danh mục sau:

*Danh mục trang cá nhân:* hiển thị thông tin cá nhân.

*Danh mục quản lý sinh viên:* thêm, xoá, sửa thông tin sinh viên.

*Danh mục quản lý giảng viên:* thêm, xoá, sửa thông tin giảng viên.

*Danh mục danh sách sinh viên đăng ký:* xem danh sách sinh viên đăng ký môn học của toàn trường qua các năm

*Danh mục quản lý kết quả học tập:* xem danh sách điểm sinh viên qua các năm. Chỉ được cập nhập điểm ở năm học hiện tại. Thống kê đạt rớt của từng năm học.

*Danh mục điểm chi tiết sinh viên:* theo dõi quá trình học tập sinh viên. Kết quả của từng môn học đã đăng ký, điểm trung bình tích luỹ, điểm rèn luyện….

*Danh mục thiết lập số tín chỉ:* cập nhập số tín chỉ nếu có thay đổi (ví dụ môn DBMS có số tín chỉ là 3 nhưng chương trình đào tạo thay đổi số tín chỉ thành 2).

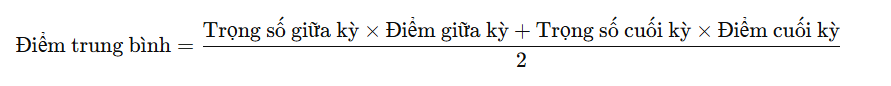
*Danh mục thiết lập trọng số:* thiết lập trọng số phần trăm điểm giữa kỳ, cuối kỳ.

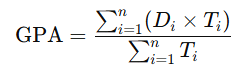
*Danh mục quản lý người dùng:* quản lý tài khoản có thể thêm xoá sửa tài khoản người dùng. Cập nhập trạng thái khoá tài khoản.

*Danh mục lịch sử đăng nhập:*theo dõi lịch sử đăng nhập để khoá hoặc xoá tài khoản kịp thời.

## 3. Công thức tính điểm

Điểm tổng kết từng môn:



Điểm trung bình tích luỹ:

Trong đó:

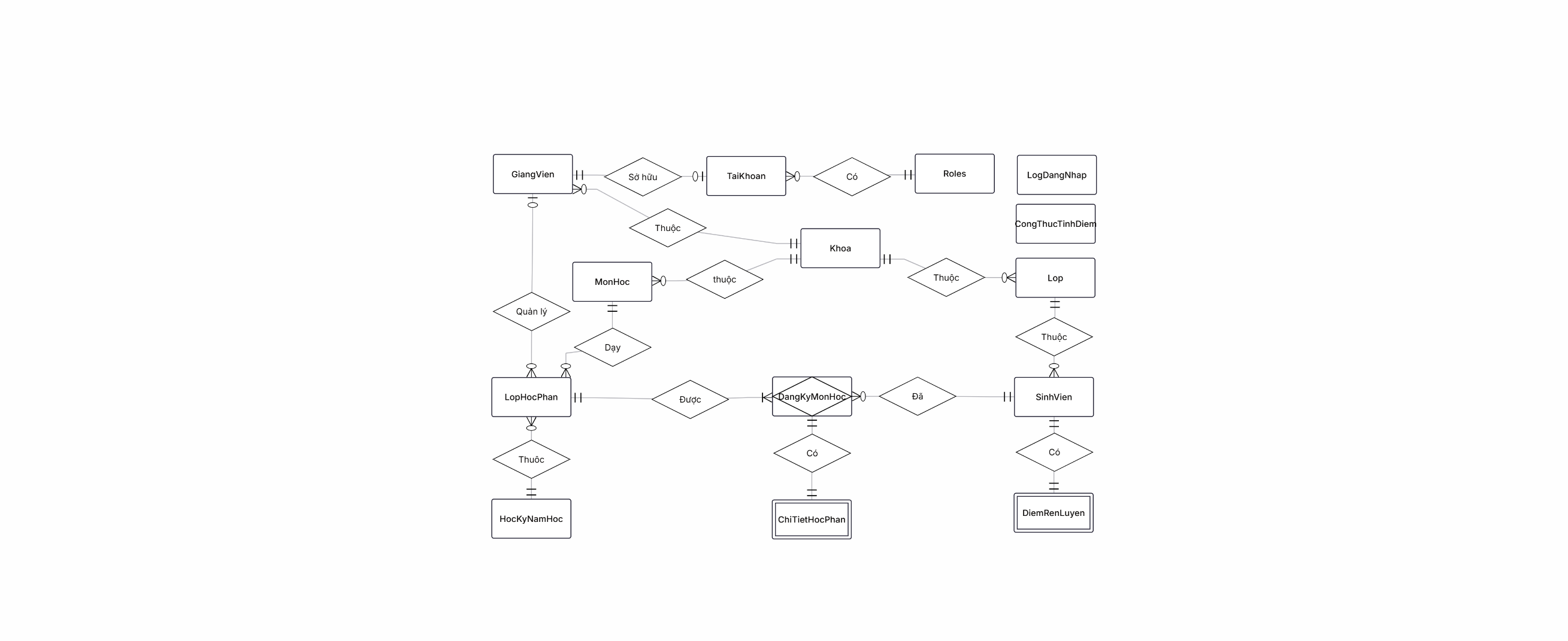
* Di​: điểm của học phần thứ I (tính theo hệ 10 hoặc hệ 4).
* Ti: số tín chỉ của học phần thứ i.
* n: tổng số học phần đã tích lũy (Môn học đạt).

Quy đổi điểm

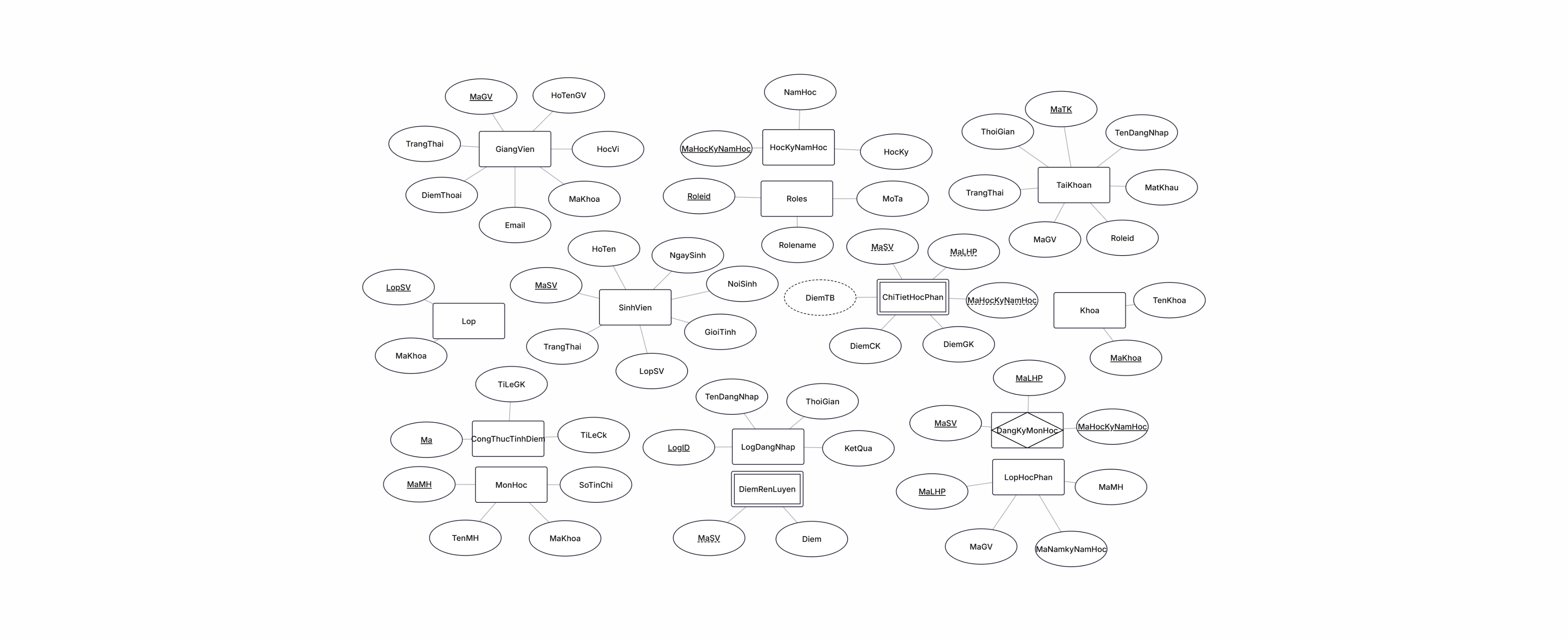
|  |  |  |
| --- | --- | --- |
| **Hệ 10 (thang 10)** | **Hệ 4 (thang 4)** | **Hệ chữ (Letter grade)** |
| Từ 9.0 – 10 | 4.0 | A+ |
| Từ 8.5 – 9.0 | 3.7 | A |
| Từ 8.0 – 8.5 | 3.5 | B+ |
| Từ 7.0 – 8.0 | 3.0 | B |
| Từ 6.5 – 7.0 | 2.5 | C+ |
| Từ 5.5 – 6.5 | 2.0 | C |
| Từ 5.0 – 5.5 | 1.5 | D+ |
| Từ 4.0 – 5.0 | 1.0 | D |
| Dưới 4.0 | 0.0 | F |

# CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## 1. Thiết kế cơ sở dữ liệu mức quan niệm

****

*Hình 1. Sơ đồ mô hình thực thể kết hợp*



*Hình 2. Chi tiết mô hình thực thể kết hợp*

## 2. Thiết kế cơ sở dữ liệu mức logic

Từ sơ đồ thực thể kết hợp (ERD), ta có các lược đồ quan hệ:

Khoa(MaKhoa, TenKhoa)

Lop(LopSV, MaKhoa)

SinhVien(MaSV, HoTen, NgaySinh, NoiSinh, GioiTinh, LopSV, TrangThai)

DiemRenLuyen(MaSV, Diem)

MonHoc(MaMH, TenMH, SoTinChi,MaKhoa)

GiangVien(MaGV, HoTenGV, HocVi, MaKhoa, Email, DienThoai, TrangThai)

HocKyNamHoc(MaHocKyNamHoc, HocKy, NamHoc)

LopHocPhan(MaLHP, MaHocKyNamHoc, MaMH, MaGV)

DangKyMonHoc(MaSV, MaLHP, MaHocKyNamHoc)

ChiTietHocPhan(MaSV, MaLHP, MaHocKyNamHoc, DiemGK, DiemCK, DiemTB)

CongThucTinhDiem(Ma, TiLeGK, TiLeCK)

Roles(Roleid, Rolename, MoTa)

TaiKhoan(MaTK, TenDangNhap, MatKhau, Roleid, MaGV, TrangThai, ThoiGian)

LogDangNhap(LogID, TenDangNhap, ThoiGian, KetQua)

## 3. Các ràng buộc

|  |  |  |
| --- | --- | --- |
| **STT** | **BẢNG** | **RÀNG BUỘC** |
| **1** | **Khoa** | [MaKhoa] [VARCHAR(10)] PRIMARY KEY,  [TenKhoa] [NVARCHAR(100)] NOT NULL |
| **2** | **Lop** | [LopSV] [VARCHAR(20)] PRIMARY KEY,  [MaKhoa] [VARCHAR(10)] NOT NULL FOREIGN KEY REFERENCES Khoa(MaKhoa) |
| **3** | **SinhVien** | [MaSV] [VARCHAR(10)] PRIMARY KEY,  [HoTen] [NVARCHAR(100)] NOT NULL,  [NgaySinh] [DATE],  [NoiSinh] [NVARCHAR(100)],  [GioiTinh] [NVARCHAR(10)] CHECK(GioiTinh IN (N'Nam', N'Nữ')),  [LopSV] [VARCHAR(20)] FOREIGN KEY REFERENCES Lop(LopSV),  [TrangThai] [BIT] DEFAULT 0 |
| **4** | **DiemRenLuyen** | [MaSV] [VARCHAR(10)] PRIMARY KEY FOREIGN KEY REFERENCES SinhVien(MaSV),  [Diem] [DECIMAL(5,2)] CHECK(Diem >= 0 AND Diem <= 100) |
| **5** | **MonHoc** | [MaMH] [VARCHAR(20)] PRIMARY KEY,  [TenMH] [NVARCHAR(100)] NOT NULL,  [SoTinChi] [INT] NOT NULL CHECK(SoTinChi > 0),  [ MaKhoa][ VARCHAR(10) ]NOT NULL REFERENCES Khoa(MaKhoa) |
| **6** | **GiangVien** | [MaGV] [VARCHAR(10)] PRIMARY KEY,  [HoTenGV] [NVARCHAR(100)] NOT NULL,  [HocVi] [NVARCHAR(50)],  [MaKhoa] [NVARCHAR(100)] REFERENCES Khoa(MaKhoa),  [Email] [VARCHAR(100)],  [DienThoai] [VARCHAR(15)],  [TrangThai] [BIT] DEFAULT 0 |
| **7** | **HocKyNamHoc** | [MaHocKyNamHoc] [INT] PRIMARY KEY IDENTITY(1,1),  [HocKy] [INT] NOT NULL CHECK(HocKy IN (1,2)),  [NamHoc] [VARCHAR(9)] NOT NULL |
| **8** | **LopHocPhan** | [MaLHP] [VARCHAR(20)] NOT NULL,  [MaMH] [VARCHAR(20)] NOT NULL FOREIGN KEY REFERENCES MonHoc(MaMH),  [MaGV] [VARCHAR(10)] FOREIGN KEY REFERENCES GiangVien(MaGV),  [MaHocKyNamHoc] [INT] NOT NULL FOREIGN KEY REFERENCES HocKyNamHoc(MaHocKyNamHoc),  PRIMARY KEY(MaLHP, MaHocKyNamHoc) |
| **9** | **DangKyMonHoc** | [MaSV] [VARCHAR(10)] NOT NULL FOREIGN KEY REFERENCES SinhVien(MaSV),  [MaLHP] [VARCHAR(20)] NOT NULL,  [MaHocKyNamHoc] [INT] NOT NULL,  PRIMARY KEY(MaSV, MaLHP, MaHocKyNamHoc),  FOREIGN KEY(MaLHP, MaHocKyNamHoc) REFERENCES LopHocPhan(MaLHP, MaHocKyNamHoc) |
| **10** | **ChiTietHocPhan** | [MaSV] [VARCHAR(10)] NOT NULL,  [MaLHP] [VARCHAR(20)] NOT NULL,  [MaHocKyNamHoc] [INT] NOT  [DiemGK] [DECIMAL(4,2)] CHECK(DiemGK BETWEEN 0 AND 10),  [DiemCK] [DECIMAL(4,2)] CHECK(DiemCK BETWEEN 0 AND 10),  [DiemTB] [DECIMAL(4,2)],  PRIMARY KEY(MaSV, MaLHP, MaHocKyNamHoc),  FOREIGN KEY(MaSV, MaLHP, MaHocKyNamHoc) REFERENCES DangKyMonHoc (MaSV,MaLHP, MaHocKyNamHoc)  ON UPDATE CASCADE |
| **11** | **CongThucTinhDiem** | [Ma] [INT] PRIMARY KEY IDENTITY(1,1),  [TiLeGK] [DECIMAL(4,2)] NOT NULL CHECK(TiLeGK >= 0 AND TiLeGK <= 1),  [TiLeCK] [DECIMAL(4,2)] NOT NULL CHECK(TiLeCK >= 0 AND TiLeCK <= 1),  CONSTRAINT CK\_TiLeTong CHECK(TiLeGK + TiLeCK = 1) |
| **12** | **Roles** | [Roleid] [INT] PRIMARY KEY,  [Rolename] [NVARCHAR(100)],  [MoTa] [NVARCHAR(1000)] |
| **13** | **TaiKhoan** | [MaTK] [INT] PRIMARY KEY IDENTITY(1,1),  [TenDangNhap] [NVARCHAR(50)] NOT NULL UNIQUE COLLATE SQL\_Latin1\_General\_CP1\_CS\_AS,  [MatKhau] [NVARCHAR(255)] NOT NULL COLLATE SQL\_Latin1\_General\_CP1\_CS\_AS,  [Roleid] [INT] FOREIGN KEY REFERENCES Roles(Roleid),  [MaGV] [VARCHAR(10)] FOREIGN KEY REFERENCES GiangVien(MaGV),  [TrangThai] [BIT] DEFAULT 1,  [ThoiGian] [DATETIME] DEFAULT GETDATE() |
| **14** | **LogDangNhap** | [LogID] [INT] PRIMARY KEY IDENTITY(1,1),  [TenDangNhap] [NVARCHAR(50)],  [ThoiGian] [DATETIME] DEFAULT GETDATE(),  [KetQua] [NVARCHAR(100)] |

## 4. Cài đặt CSDL và các ràng buộc

***Bảng Khoa***

|  |
| --- |
| CREATE TABLE Khoa (  MaKhoa VARCHAR(10) PRIMARY KEY,  TenKhoa NVARCHAR(100) NOT NULL  ); |

***Bảng Lớp***

|  |
| --- |
| CREATE TABLE Lop (  LopSV VARCHAR(20) PRIMARY KEY,  MaKhoa VARCHAR(10) NOT NULL REFERENCES Khoa(MaKhoa)  ); |

***Bảng Sinh viên***

|  |
| --- |
| CREATE TABLE SinhVien (  MaSV VARCHAR(10) PRIMARY KEY,  HoTen NVARCHAR(100) NOT NULL,  NgaySinh DATE,  NoiSinh NVARCHAR(100),  GioiTinh NVARCHAR(10) CHECK (GioiTinh IN (N'Nam', N'Nữ')),  LopSV VARCHAR(20) REFERENCES Lop(LopSV),  TrangThai BIT DEFAULT 0 -- 0 là còn hoạt động  ); |

***Bảng Điểm rèn luyện***

|  |
| --- |
| CREATE TABLE DiemRenLuyen (  MaSV VARCHAR(10) PRIMARY KEY REFERENCES SinhVien(MaSV),  Diem DECIMAL(5,2) CHECK (Diem >= 0 AND Diem <= 100)  ); |

***Bảng Môn học***

|  |
| --- |
| CREATE TABLE MonHoc (  MaMH VARCHAR(20) PRIMARY KEY,  TenMH NVARCHAR(100) NOT NULL,  SoTinChi INT NOT NULL CHECK (SoTinChi > 0),  MaKhoa VARCHAR(10) NOT NULL REFERENCES Khoa(MaKhoa)  ); |

***Bảng Giảng viên***

|  |
| --- |
| CREATE TABLE GiangVien (  MaGV VARCHAR(10) PRIMARY KEY,  HoTenGV NVARCHAR(100) NOT NULL,  HocVi NVARCHAR(50),  MaKhoa VARCHAR(10) NOT NULL REFERENCES Khoa(MaKhoa),  Email VARCHAR(100),  DienThoai VARCHAR(15),  TrangThai BIT DEFAULT 0 -- 0 là còn hoạt động  ); |

***Bảng Học kỳ năm học***

|  |
| --- |
| CREATE TABLE HocKyNamHoc (  MaHocKyNamHoc INT IDENTITY(1,1) PRIMARY KEY,  HocKy INT NOT NULL CHECK (HocKy IN (1,2)),  NamHoc VARCHAR(9) NOT NULL  ); |

***Bảng Lớp học phần***

|  |
| --- |
| CREATE TABLE LopHocPhan (  MaLHP VARCHAR(20) NOT NULL,  MaMH VARCHAR(20) NOT NULL REFERENCES MonHoc(MaMH),  MaGV VARCHAR(10) NOT NULL REFERENCES GiangVien(MaGV),  MaHocKyNamHoc INT NOT NULL REFERENCES HocKyNamHoc(MaHocKyNamHoc),  PRIMARY KEY (MaLHP,MaHocKyNamHoc)  ); |

***Bảng*** ***Đăng ký môn học***

|  |
| --- |
| CREATE TABLE DangKyMonHoc (  MaSV VARCHAR(10) NOT NULL REFERENCES SinhVien(MaSV),  MaLHP VARCHAR(20) NOT NULL,  MaHocKyNamHoc INT NOT NULL,  PRIMARY KEY (MaSV, MaLHP,MaHocKyNamHoc),  FOREIGN KEY (MaLHP,MaHocKyNamHoc) REFERENCES LopHocPhan(MaLHP,MaHocKyNamHoc)  ); |

***Bảng Chi tiết học phần***

|  |
| --- |
| CREATE TABLE ChiTietHocPhan (  MaSV VARCHAR(10) NOT NULL,  MaLHP VARCHAR(20) NOT NULL,  MaHocKyNamHoc INT NOT NULL,  DiemGK DECIMAL(4,2) CHECK (DiemGK BETWEEN 0 AND 10),  DiemCK DECIMAL(4,2) CHECK (DiemCK BETWEEN 0 AND 10),  DiemTB DECIMAL(4,2),  PRIMARY KEY (MaSV, MaLHP, MaHocKyNamHoc),  FOREIGN KEY (MaSV, MaLHP, MaHocKyNamHoc)  REFERENCES DangKyMonHoc(MaSV, MaLHP, MaHocKyNamHoc)  ON UPDATE CASCADE  ); |

***Bảng Công thức tính điểm***

|  |
| --- |
| CREATE TABLE CongThucTinhDiem(  Ma INT IDENTITY(1,1) PRIMARY KEY,  TiLeGK DECIMAL(4,2) NOT NULL CHECK (TiLeGK >= 0 AND TiLeGK <= 1),  TiLeCK DECIMAL(4,2) NOT NULL CHECK (TiLeCK >= 0 AND TiLeCK <= 1),  CONSTRAINT CK\_TiLeTong CHECK (TiLeGK + TiLeCK = 1)  ); |

***Bảng Vai trò (Roles)***

|  |
| --- |
| CREATE TABLE Roles (  Roleid int PRIMARY KEY,  Rolename NVARCHAR(100),  MoTa NVARCHAR(1000)  ); |

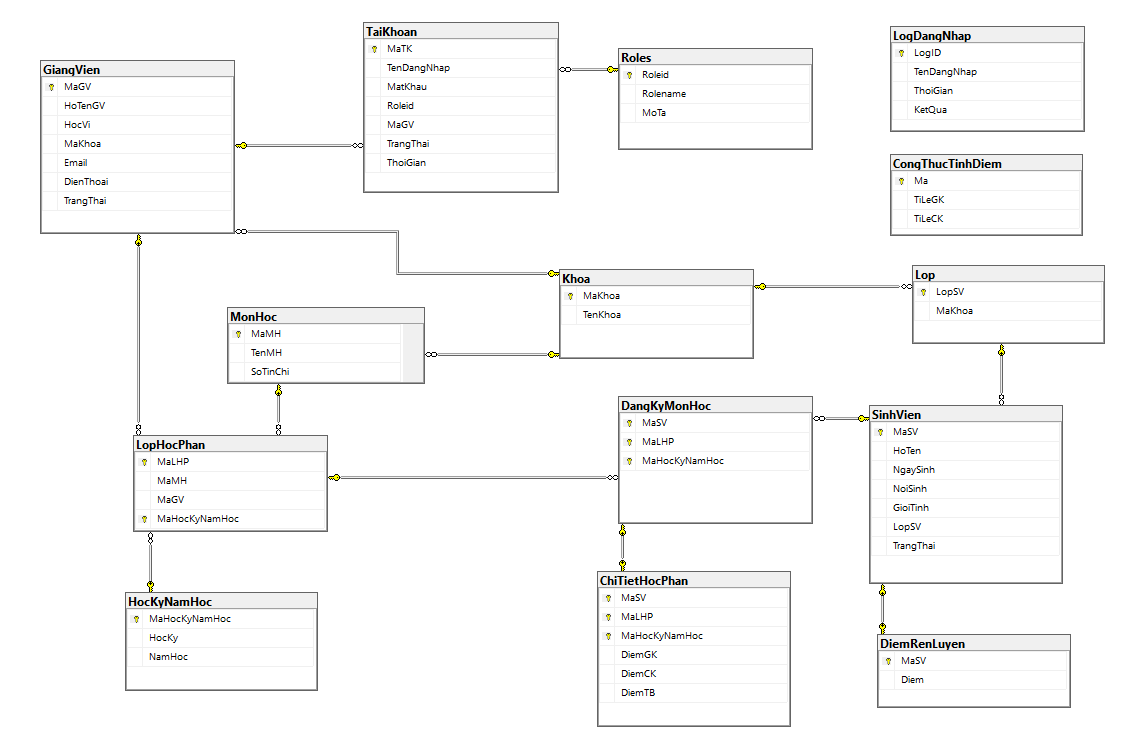
***Bảng Tài khoản***

|  |
| --- |
| CREATE TABLE TaiKhoan (  MaTK INT IDENTITY(1,1) PRIMARY KEY,  TenDangNhap NVARCHAR(50) COLLATE SQL\_Latin1\_General\_CP1\_CS\_AS NOT NULL UNIQUE,  MatKhau NVARCHAR(255) COLLATE SQL\_Latin1\_General\_CP1\_CS\_AS NOT NULL,  Roleid int ,  MaGV VARCHAR(10),  TrangThai BIT DEFAULT 1,  ThoiGian DATETIME DEFAULT GETDATE(),  FOREIGN KEY (Roleid) REFERENCES Roles(Roleid),  FOREIGN KEY (MaGV) REFERENCES GiangVien(MaGV)  ); |

***Bảng Log/Nhật ký hệ thống***

|  |
| --- |
| CREATE TABLE LogDangNhap  (  LogID INT IDENTITY(1,1) PRIMARY KEY,  TenDangNhap NVARCHAR(50),  ThoiGian DATETIME DEFAULT GETDATE(),  KetQua NVARCHAR(100)  ); |

## 5. Lược đồ quan hệ sau khi cài đặt SQL

****

*Hình 3. Lược đồ quan hệ cơ sở dữ liệu*

## 6. Các triggers

**a. Trigger khoá tài khoản khi giảng viên ngừng hoạt động**

|  |
| --- |
| CREATE TRIGGER trg\_KhoaTaiKhoanVaCapNhatLopHocPhan  ON GiangVien  AFTER UPDATE  AS  BEGIN  SET NOCOUNT ON;  -- Chỉ những giảng viên vừa nghỉ (trạng thái từ 0 -> 1)  -- 1. Khóa tài khoản  UPDATE tk  SET tk.TrangThai = 0  FROM TaiKhoan tk  INNER JOIN inserted i ON tk.MaGV = i.MaGV  INNER JOIN deleted d ON d.MaGV = i.MaGV  WHERE i.TrangThai = 1 AND d.TrangThai = 0;  -- 2. Cập nhật lớp học phần: MaGV = NULL  UPDATE lhp  SET lhp.MaGV = NULL  FROM LopHocPhan lhp  INNER JOIN inserted i ON lhp.MaGV = i.MaGV  INNER JOIN deleted d ON d.MaGV = i.MaGV  WHERE i.TrangThai = 1 AND d.TrangThai = 0;  END  GO |

**b. Trigger tạo lịch sử đăng nhập**

|  |
| --- |
| CREATE TRIGGER trg\_LogDangNhap  ON LogDangNhap  AFTER INSERT  AS  BEGIN  SET NOCOUNT ON;  SELECT  'Người dùng: ' + ISNULL(TenDangNhap, 'NULL') +  ' | Kết quả: ' + ISNULL(KetQua, 'NULL') AS ThongBao  FROM inserted;  END; |

**c. Trigger thêm sinh viên đã đăng ký môn học vào bảng chi tiết học phần**

|  |
| --- |
| CREATE TRIGGER trg\_AfterInsert\_DangKyMonHoc  ON DangKyMonHoc  AFTER INSERT  AS  BEGIN  INSERT INTO ChiTietHocPhan (MaSV, MaLHP, DiemGK, DiemCK,DiemTB, MaHocKyNamHoc)  SELECT  i.MaSV,  i.MaLHP,  NULL,  NULL,  NULL,  i.MaHocKyNamHoc  FROM inserted i  END; |

**d. Trigger cập nhập điểm còn thiếu của sinh viên bằng 0 nếu ngừng hoạt động**

|  |
| --- |
| CREATE TRIGGER dbo.trg\_UpdateDiem\_KhiSVNgungHoatDong  ON SinhVien  AFTER INSERT, UPDATE  AS  BEGIN  SET NOCOUNT ON;  UPDATE CTHP  SET  DiemGK = ISNULL(CTHP.DiemGK, 0),  DiemCK = ISNULL(CTHP.DiemCK, 0),  DiemTB = 0  FROM ChiTietHocPhan CTHP  INNER JOIN inserted i ON CTHP.MaSV = i.MaSV  WHERE i.TrangThai = 1;  END;  GO |

**e. Trigger tính điểm trung bình khi có đủ điểm giữa kỳ và cuối kỳ**

|  |
| --- |
| CREATE TRIGGER trg\_TinhDiemTB  ON ChiTietHocPhan  AFTER INSERT, UPDATE  AS  BEGIN  SET NOCOUNT ON;  UPDATE CTHP  SET CTHP.DiemTB = dbo.fn\_TinhDiemTrungBinh(CTHP.DiemGK, CTHP.DiemCK)  FROM ChiTietHocPhan CTHP  INNER JOIN inserted i  ON CTHP.MaSV = i.MaSV  AND CTHP.MaLHP = i.MaLHP  AND CTHP.MaHocKyNamHoc = i.MaHocKyNamHoc  WHERE i.DiemGK IS NOT NULL  AND i.DiemCK IS NOT NULL;  END; |

## 7. Các hàm

**a. Hàm trả về giá trị trạng thái môn học**

|  |
| --- |
| CREATE FUNCTION fn\_TrangThaiDiemTB  (  @DiemTB FLOAT  )  RETURNS NVARCHAR(20)  AS  BEGIN  DECLARE @TrangThai NVARCHAR(20);  IF @DiemTB IS NULL  SET @TrangThai = NULL;  ELSE IF @DiemTB >= 5  SET @TrangThai = N'Đạt';  ELSE  SET @TrangThai = N'Không đạt';  RETURN @TrangThai;  END; |

**b. Hàm quy đổi đúng định dạng dd/mm/yyyy**

|  |
| --- |
| CREATE FUNCTION dbo.fn\_FormattedDate (@Ngay DATE)  RETURNS NVARCHAR(10)  AS  BEGIN  IF @Ngay IS NULL  RETURN NULL;  RETURN RIGHT('0' + CAST(DAY(@Ngay) AS VARCHAR(2)), 2) + '/' +  RIGHT('0' + CAST(MONTH(@Ngay) AS VARCHAR(2)), 2) + '/' +  CAST(YEAR(@Ngay) AS VARCHAR(4));  END;  GO |

**c. Hàm quy đổi điểm hệ 10 sang hệ 4**

|  |
| --- |
| CREATE FUNCTION fn\_QuyDoiDiemHe4 (@DiemHe10 DECIMAL(4,2))  RETURNS DECIMAL(3,2)  AS  BEGIN  DECLARE @DiemHe4 DECIMAL(3,2);  IF @DiemHe10 IS NULL  RETURN NULL;  IF @DiemHe10 >= 9.0 SET @DiemHe4 = 4.0;  ELSE IF @DiemHe10 >= 8.5 SET @DiemHe4 = 3.7;  ELSE IF @DiemHe10 >= 8.0 SET @DiemHe4 = 3.5;  ELSE IF @DiemHe10 >= 7.0 SET @DiemHe4 = 3.0;  ELSE IF @DiemHe10 >= 6.5 SET @DiemHe4 = 2.5;  ELSE IF @DiemHe10 >= 5.5 SET @DiemHe4 = 2.0;  ELSE IF @DiemHe10 >= 5.0 SET @DiemHe4 = 1.5;  ELSE IF @DiemHe10 >= 4.0 SET @DiemHe4 = 1.0;  ELSE SET @DiemHe4 = 0.0;  RETURN @DiemHe4;  END  GO |

**d. Hàm quy đổi điểm hệ 10 sang hệ chữ**

|  |
| --- |
| CREATE FUNCTION fn\_QuyDoiDiemChu (@DiemHe10 DECIMAL(4,2))  RETURNS NVARCHAR(2)  AS  BEGIN  DECLARE @Chu NVARCHAR(2);  IF @DiemHe10 IS NULL  RETURN NULL;  IF @DiemHe10 >= 9.0 SET @Chu = N'A+';  ELSE IF @DiemHe10 >= 8.5 SET @Chu = N'A';  ELSE IF @DiemHe10 >= 8.0 SET @Chu = N'B+';  ELSE IF @DiemHe10 >= 7.0 SET @Chu = N'B';  ELSE IF @DiemHe10 >= 6.5 SET @Chu = N'C+';  ELSE IF @DiemHe10 >= 5.5 SET @Chu = N'C';  ELSE IF @DiemHe10 >= 5.0 SET @Chu = N'D+';  ELSE IF @DiemHe10 >= 4.0 SET @Chu = N'D';  ELSE SET @Chu = N'F';  RETURN @Chu;  END  GO |

**e. Hàm tính điểm trung bình**

|  |
| --- |
| CREATE FUNCTION dbo.fn\_TinhDiemTrungBinh  (  @DiemGK DECIMAL(4,2),  @DiemCK DECIMAL(4,2)  )  RETURNS DECIMAL(4,2)  AS  BEGIN  DECLARE @DiemTB DECIMAL(4,2);  DECLARE @TiLeGK DECIMAL(4,2);  DECLARE @TiLeCK DECIMAL(4,2);  SELECT TOP 1  @TiLeGK = TiLeGK,  @TiLeCK = TiLeCK  FROM CongThucTinhDiem  ORDER BY Ma DESC;  IF @TiLeGK IS NULL OR @TiLeCK IS NULL  BEGIN  SET @TiLeGK = 0.5;  SET @TiLeCK = 0.5;  END  IF @DiemGK IS NULL OR @DiemCK IS NULL  BEGIN  RETURN NULL;  END  IF @DiemCK < 3  BEGIN  SET @DiemGK=0;  END    SET @DiemTB = ROUND(@DiemGK \* @TiLeGK + @DiemCK \* @TiLeCK, 2);  RETURN @DiemTB;  END  GO |

# CHƯƠNG 3: THIẾT KẾ CÁC CHỨC NĂNG

## Kết nối cơ sở dữ liệu

|  |
| --- |
| using DoAnCK.UI\_Admin;  using DoAnCK.UI\_Dangnhap;  using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  namespace DoAnCK  {  public interface IRefreshable  {  void RefreshData();  }  public partial class FormMain : Form  {  public static string ConnString { get; private set; }  public static int CurrentRoleID { get; private set; }  uc\_Dangnhap ucDangnhap;  uc\_Chonquyen ucChonquyen;  public FormMain()  {  InitializeComponent();  ConnString = @"Server=.;Database=QL\_SinhVien;Integrated Security=true;Encrypt=false;TrustServerCertificate=true;Connection Timeout=30;";  CurrentRoleID = 0;  }  // Phương thức để cập nhật ConnString với username và password sau khi đăng nhập thành công  public static void UpdateConnString(string username, string password)  {  ConnString = $@"Server=.;Database=QL\_SinhVien;User ID={username};Password={password};Encrypt=false;TrustServerCertificate=true;Connection Timeout=30;";  }  public static void SetCurrentRole(int roleID)  {  CurrentRoleID = roleID;  }  public static void ResetConnection()  {  ConnString = @"Server=.;Database=QL\_SinhVien;Integrated Security=true;Encrypt=false;TrustServerCertificate=true;Connection Timeout=30;";  CurrentRoleID = 0;  }  private void FormMain\_Load(object sender, EventArgs e)  {  if (ucChonquyen == null)  {  ucChonquyen = new uc\_Chonquyen();  ucChonquyen.Dock = DockStyle.Fill;  ucChonquyen.AutoSize = true;  ucChonquyen.OnChonQuyen += UcChonquyen\_OnChonQuyen;  panelMain.Controls.Add(ucChonquyen);  ucChonquyen.BringToFront();  }  else  {  ucChonquyen.BringToFront();  }  }  private void UcChonquyen\_OnChonQuyen(object sender, int role)  {  SetCurrentRole(role);  if (ucDangnhap != null)  {  panelMain.Controls.Remove(ucDangnhap);  ucDangnhap.Dispose();  }  ucDangnhap = new uc\_Dangnhap(role);  ucDangnhap.Dock = DockStyle.Fill;  ucDangnhap.AutoSize = true;  ucDangnhap.OnExit += UcDangnhap\_OnExit;  panelMain.Controls.Add(ucDangnhap);  ucDangnhap.BringToFront();  }  private void UcDangnhap\_OnExit(object sender, EventArgs e)  {  ResetConnection();  ucChonquyen.BringToFront();  }  }  } |

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Drawing;  using System.Data;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  namespace DoAnCK.UI\_Dangnhap  {  public partial class uc\_Chonquyen : UserControl  {  public event EventHandler<int> OnChonQuyen;  public uc\_Chonquyen()  {  InitializeComponent();    }  private void btn\_GV\_Click(object sender, EventArgs e)  {  OnChonQuyen?.Invoke(this, 2);  }  private void btnAdmin\_Click(object sender, EventArgs e)  {  OnChonQuyen?.Invoke(this, 1);  }  }  } |

Chọn roles tạm thời

|  |
| --- |
| using Microsoft.EntityFrameworkCore;  using QuanLyThuVien.DAL.Entities;  using System;  namespace QuanLyThuVien.BLL.Services  {  public static class ContextFactory  {  public static QuanLyThuVienContext CreateContext()  {  var connectionString = ConnectionStringProvider.GetConnectionString();    var optionsBuilder = new DbContextOptionsBuilder<QuanLyThuVienContext>();  optionsBuilder.UseSqlServer(connectionString);    return new QuanLyThuVienContext(optionsBuilder.Options);  }  public static QuanLyThuVienContext CreateContext(string connectionString)  {  var optionsBuilder = new DbContextOptionsBuilder<QuanLyThuVienContext>();  optionsBuilder.UseSqlServer(connectionString);    return new QuanLyThuVienContext(optionsBuilder.Options);  }  }  } |

Xử lý kết nối cơ sở dữ liệu và đăng nhập vào hệ thống thông qua form Đăng nhập

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Drawing;  using System.Data;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  using System.Data.SqlClient;  namespace DoAnCK.UI\_Dangnhap  {  public partial class uc\_Dangnhap : UserControl  {  public event EventHandler OnExit;  private string connStr = FormMain.ConnString;  private int roleID;  public static string TenDangNhap { get; private set; }  public static string MatKhau { get; private set; }  public uc\_Dangnhap(int roleID)  {  InitializeComponent();  this.roleID = roleID;  }  private void uc\_Dangnhap\_Load(object sender, EventArgs e)  {  }  private void txtThoat\_Click(object sender, EventArgs e)  {  OnExit?.Invoke(this, EventArgs.Empty);  }  private void btnDangNhap\_Click(object sender, EventArgs e)  {  string tenDangNhap = txtTaiKhoan.Text.Trim();  string matKhau = txtMatKhau.Text.Trim();  if (string.IsNullOrEmpty(tenDangNhap) || string.IsNullOrEmpty(matKhau))  {  MessageBox.Show("Vui lòng nhập đầy đủ tài khoản và mật khẩu!", "Thông báo",  MessageBoxButtons.OK, MessageBoxIcon.Warning);  return;  }  try  {  using (SqlConnection conn = new SqlConnection(FormMain.ConnString))  using (SqlCommand cmd = new SqlCommand("sp\_DangNhap", conn))  {  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@TenDangNhap", tenDangNhap);  cmd.Parameters.AddWithValue("@MatKhau", matKhau);  cmd.Parameters.AddWithValue("@RoleIDtam", roleID);  var pRoleID = new SqlParameter("@RoleID", SqlDbType.Int) { Direction = ParameterDirection.Output };  var pTrangThai = new SqlParameter("@TrangThai", SqlDbType.Bit) { Direction = ParameterDirection.Output };  var pMaGV = new SqlParameter("@MaGV", SqlDbType.VarChar, 10) { Direction = ParameterDirection.Output };  var pKetQua = new SqlParameter("@KetQua", SqlDbType.NVarChar, 100) { Direction = ParameterDirection.Output };  cmd.Parameters.Add(pRoleID);  cmd.Parameters.Add(pTrangThai);  cmd.Parameters.Add(pMaGV);  cmd.Parameters.Add(pKetQua);  conn.Open();  cmd.ExecuteNonQuery();  string ketQua = pKetQua.Value?.ToString();  int? roleThuc = pRoleID.Value != DBNull.Value ? (int?)pRoleID.Value : null;  bool? trangThai = pTrangThai.Value != DBNull.Value ? (bool?)pTrangThai.Value : null;  string maGV = pMaGV.Value?.ToString();  if (roleThuc == null)  {  MessageBox.Show($"{ketQua}",  "Đăng nhập thất bại", MessageBoxButtons.OK, MessageBoxIcon.Error);  return;  }  if (trangThai == false)  {  MessageBox.Show($"{ketQua}",  "Tài khoản bị khóa", MessageBoxButtons.OK, MessageBoxIcon.Warning);  return;  }  FormMain.UpdateConnString(tenDangNhap, matKhau);  TenDangNhap = tenDangNhap;  MatKhau = matKhau;  if (roleThuc == 1) // Admin  {  MessageBox.Show($"{ketQua}",  "Thành công", MessageBoxButtons.OK, MessageBoxIcon.Information);  // Đảm bảo frmAdmin sử dụng connection string mới  frmAdmin frm = new frmAdmin();  Form parent = this.FindForm();  parent.Hide();  frm.ShowDialog();    }  else if (roleThuc == 2) // GiangVien  {  MessageBox.Show($"{ketQua}",  "Thành công", MessageBoxButtons.OK, MessageBoxIcon.Information);  frmGiangVien frm = new frmGiangVien(maGV);  Form parent = this.FindForm();  parent.Hide();  frm.ShowDialog();    }  else  {  MessageBox.Show("Quyền không hợp lệ!", "Lỗi",  MessageBoxButtons.OK, MessageBoxIcon.Error);  }  }  }  catch (SqlException ex)  {    MessageBox.Show($"Lỗi SQL với quyền {GetRoleName(roleID)}: {ex.Message}",  "Lỗi đăng nhập", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  catch (Exception ex)  {  MessageBox.Show($"Lỗi kết nối: {ex.Message}", "Lỗi",  MessageBoxButtons.OK, MessageBoxIcon.Error);  }  }  private string GetRoleName(int roleID)  {  return roleID == 1 ? "Admin" : roleID == 2 ? "Giảng viên" : "Không xác định";  }  }  } |

## Chức năng đăng nhập

|  |
| --- |
| CREATE PROCEDURE dbo.sp\_DangNhap  @TenDangNhap NVARCHAR(50),  @MatKhau NVARCHAR(255),  @RoleIDtam INT, -- 1 = Admin, 2 = GiangVien (quyền người dùng chọn)  @RoleID INT OUTPUT, -- quyền thực của tài khoản (1 = Admin, 2 = GiangVien)  @TrangThai BIT OUTPUT,  @MaGV VARCHAR(10) OUTPUT,  @KetQua NVARCHAR(100) OUTPUT  AS  BEGIN  SET NOCOUNT ON;  DECLARE @RoleThuc NVARCHAR(20); -- role thực từ database  BEGIN TRY  -- Lấy thông tin tài khoản theo TenDangNhap  SELECT  @RoleThuc = r.RoleName,  @TrangThai = tk.TrangThai,  @MaGV = tk.MaGV,  @RoleID = r.Roleid  FROM TaiKhoan tk  LEFT JOIN Roles r ON tk.Roleid = r.Roleid  WHERE tk.TenDangNhap = @TenDangNhap  AND tk.MatKhau = @MatKhau;  -- Kiểm tra tài khoản tồn tại  IF @RoleThuc IS NULL  BEGIN  SET @KetQua = N'Tên đăng nhập hoặc mật khẩu không chính xác!';  INSERT INTO LogDangNhap(TenDangNhap, KetQua) VALUES (@TenDangNhap, @KetQua);  RETURN;  END  -- Kiểm tra trạng thái tài khoản  IF @TrangThai = 0  BEGIN  SET @KetQua = N'Tài khoản đã bị khóa!';  INSERT INTO LogDangNhap(TenDangNhap, KetQua) VALUES (@TenDangNhap, @KetQua);  RETURN;  END  -- Kiểm tra quyền đăng nhập  IF (@RoleIDtam = 1 AND @RoleThuc = N'Admin') OR (@RoleIDtam = 2 AND @RoleThuc = N'GiangVien')  BEGIN  SET @KetQua = N'Đăng nhập thành công với quyền ' + @RoleThuc;  INSERT INTO LogDangNhap(TenDangNhap, KetQua) VALUES (@TenDangNhap, @KetQua);  END  ELSE  BEGIN  SET @KetQua = N'Tài khoản không có quyền này!';  INSERT INTO LogDangNhap(TenDangNhap, KetQua) VALUES (@TenDangNhap, @KetQua);  -- Reset thông tin đầu ra nếu không phù hợp quyền  SET @RoleID = NULL;  SET @TrangThai = NULL;  SET @MaGV = NULL;  END  END TRY  BEGIN CATCH  SET @KetQua = N'Lỗi trong quá trình đăng nhập: ' + ERROR\_MESSAGE();  SET @RoleID = NULL;  SET @TrangThai = NULL;  SET @MaGV = NULL;  END CATCH  END |

C#

|  |
| --- |
| private void btnDangNhap\_Click(object sender, EventArgs e)  {  string tenDangNhap = txtTaiKhoan.Text.Trim();  string matKhau = txtMatKhau.Text.Trim();  if (string.IsNullOrEmpty(tenDangNhap) || string.IsNullOrEmpty(matKhau))  {  MessageBox.Show("Vui lòng nhập đầy đủ tài khoản và mật khẩu!", "Thông báo",  MessageBoxButtons.OK, MessageBoxIcon.Warning);  return;  }  try  {  using (SqlConnection conn = new SqlConnection(FormMain.ConnString))  using (SqlCommand cmd = new SqlCommand("sp\_DangNhap", conn))  {  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@TenDangNhap", tenDangNhap);  cmd.Parameters.AddWithValue("@MatKhau", matKhau);  cmd.Parameters.AddWithValue("@RoleIDtam", roleID);  var pRoleID = new SqlParameter("@RoleID", SqlDbType.Int) { Direction = ParameterDirection.Output };  var pTrangThai = new SqlParameter("@TrangThai", SqlDbType.Bit) { Direction = ParameterDirection.Output };  var pMaGV = new SqlParameter("@MaGV", SqlDbType.VarChar, 10) { Direction = ParameterDirection.Output };  var pKetQua = new SqlParameter("@KetQua", SqlDbType.NVarChar, 100) { Direction = ParameterDirection.Output };  cmd.Parameters.Add(pRoleID);  cmd.Parameters.Add(pTrangThai);  cmd.Parameters.Add(pMaGV);  cmd.Parameters.Add(pKetQua);  conn.Open();  cmd.ExecuteNonQuery();  string ketQua = pKetQua.Value?.ToString();  int? roleThuc = pRoleID.Value != DBNull.Value ? (int?)pRoleID.Value : null;  bool? trangThai = pTrangThai.Value != DBNull.Value ? (bool?)pTrangThai.Value : null;  string maGV = pMaGV.Value?.ToString();  if (roleThuc == null)  {  MessageBox.Show($"{ketQua}",  "Đăng nhập thất bại", MessageBoxButtons.OK, MessageBoxIcon.Error);  return;  }  if (trangThai == false)  {  MessageBox.Show($"{ketQua}",  "Tài khoản bị khóa", MessageBoxButtons.OK, MessageBoxIcon.Warning);  return;  }  FormMain.UpdateConnString(tenDangNhap, matKhau);  TenDangNhap = tenDangNhap;  MatKhau = matKhau;  if (roleThuc == 1) // Admin  {  MessageBox.Show($"{ketQua}",  "Thành công", MessageBoxButtons.OK, MessageBoxIcon.Information);  // Đảm bảo frmAdmin sử dụng connection string mới  frmAdmin frm = new frmAdmin(maGV);  Form parent = this.FindForm();  parent.Hide();  frm.ShowDialog();    }  else if (roleThuc == 2) // GiangVien  {  MessageBox.Show($"{ketQua}",  "Thành công", MessageBoxButtons.OK, MessageBoxIcon.Information);  frmGiangVien frm = new frmGiangVien(maGV);  Form parent = this.FindForm();  parent.Hide();  frm.ShowDialog();    }  else  {  MessageBox.Show("Quyền không hợp lệ!", "Lỗi",  MessageBoxButtons.OK, MessageBoxIcon.Error);  }  }  }  catch (SqlException ex)  {  MessageBox.Show($"Lỗi SQL với quyền {GetRoleName(roleID)}: {ex.Message}",  "Lỗi đăng nhập", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  catch (Exception ex)  {  MessageBox.Show($"Lỗi kết nối: {ex.Message}", "Lỗi",  MessageBoxButtons.OK, MessageBoxIcon.Error);  } |

## Chức năng hiển thị

**Thông tin giảng viên**

|  |
| --- |
| CREATE FUNCTION dbo.fn\_GetThongTinGV(@MaGV VARCHAR(10))  RETURNS TABLE  AS  RETURN  (  SELECT  gv.MaGV,  gv.HoTenGV,  gv.HocVi,  k.TenKhoa,  gv.Email,  gv.DienThoai  FROM GiangVien gv  LEFT JOIN Khoa k  ON gv.MaKhoa = k.MaKhoa  WHERE gv.MaGV = @MaGV  );  GO |

C#

|  |
| --- |
| private void LoadThongTinGiangVien()  {  try  {  string queryGV = $"SELECT \* FROM dbo.fn\_GetThongTinGV('{MaGV}')";  DataTable dtGV = frmAdmin.getData(queryGV);  if (dtGV != null && dtGV.Rows.Count > 0)  {  DataRow row = dtGV.Rows[0];  txtMa.Text = row["MaGV"].ToString();  txtHoten.Text = row["HoTenGV"].ToString();  txtHocvi.Text = row["HocVi"]?.ToString() ?? "";  txtKhoa.Text = row["Khoa"]?.ToString() ?? "";  txtEmail.Text = row["Email"]?.ToString() ?? "";  txtSDT.Text = row["DienThoai"]?.ToString() ?? "";  }  else  {  MessageBox.Show("Không tìm thấy thông tin giảng viên.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  }  catch (Exception ex)  {  MessageBox.Show("Lỗi khi tải thông tin: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

**Danh sách giảng viên**

|  |
| --- |
| CREATE VIEW dbo.v\_GiangVien\_Detail  AS  SELECT  GV.MaGV,  GV.HoTenGV,  GV.HocVi,  GV.MaKhoa,  GV.Email,  GV.DienThoai,  GV.TrangThai  FROM GiangVien GV;  GO |

C#:

|  |
| --- |
| public void RefreshData()  {  btnLuu.Enabled = false;  btnHuy.Enabled = false;  btnSua.Enabled = true;  isAdding = false;  gvDanhSachSV.OptionsBehavior.Editable = false;  btnThem.Enabled = true;    dtGiangVien = frmAdmin.getData("SELECT \* FROM v\_GiangVien\_Detail;");  if (dtGiangVien != null)  {  gcDanhSachSV.DataSource = dtGiangVien;  gvDanhSachSV.OptionsSelection.MultiSelect = true;  gvDanhSachSV.OptionsSelection.MultiSelectMode = GridMultiSelectMode.RowSelect;  }  loadKhoa();  gvDanhSachSV.RefreshData();    } |

**Danh sách sinh viên**

|  |
| --- |
| CREATE VIEW dbo.v\_SinhVien\_Detail  AS  SELECT  SV.MaSV,  SV.HoTen,  SV.NgaySinh,  SV.NoiSinh,  SV.GioiTinh,  SV.LopSV,  SV.TrangThai  FROM SinhVien SV;  GO |

C#:

|  |
| --- |
| public void RefreshData()  {  btnLuu.Enabled = false;  btnHuy.Enabled = false;  btnSua.Enabled = true;  isAdding = false;  gvDanhSachSV.OptionsBehavior.Editable = false;  btnThem.Enabled = true;    dtSinhVien = frmAdmin.getData("SELECT \* FROM v\_SinhVien\_Detail;");  if (dtSinhVien != null)  {  gcDanhSachSV.DataSource = dtSinhVien;  gvDanhSachSV.OptionsSelection.MultiSelect = true;  gvDanhSachSV.OptionsSelection.MultiSelectMode = GridMultiSelectMode.RowSelect;  loadGioiTinh();  LoadLop();  }  gvDanhSachSV.RefreshData();  } |

**Danh sách năm học**

|  |
| --- |
| CREATE PROCEDURE sp\_DanhSachHocKyNamHoc  AS  BEGIN  SELECT  MaHocKyNamHoc,  HocKy,  NamHoc  FROM HocKyNamHoc  ORDER BY MaHocKyNamHoc DESC;  END;  GO |

C#:

|  |
| --- |
| string queryNamHoc = "EXEC sp\_DanhSachHocKyNamHoc;";  DataTable dtNamHoc = frmAdmin.getData(queryNamHoc);  if (dtNamHoc != null && dtNamHoc.Rows.Count > 0)  {  dtNamHoc.Columns.Add("HK\_NamHoc", typeof(string));  foreach (DataRow row in dtNamHoc.Rows)  {  row["HK\_NamHoc"] = $"HK{row["HocKy"]} - {row["NamHoc"]}";  }  cbbNamHoc.DataSource = dtNamHoc;  cbbNamHoc.DisplayMember = "HK\_NamHoc";  cbbNamHoc.ValueMember = "MaHocKyNamHoc"; |

**Năm học hiện tại**

|  |
| --- |
| CREATE PROCEDURE sp\_GetMaHocKyMoiNhat  AS  BEGIN  SELECT TOP 1 MaHocKyNamHoc  FROM HocKyNamHoc  ORDER BY MaHocKyNamHoc DESC;  END;  GO |

C#:

|  |
| --- |
| string queryMaHK = "EXEC sp\_GetMaHocKyMoiNhat";  DataTable dtHK = frmGiangVien.getData(queryMaHK);  if (dtHK != null && dtHK.Rows.Count > 0)  maHocKyNamHoc = Convert.ToInt32(dtHK.Rows[0]["MaHocKyNamHoc"]);  else  MessageBox.Show("Không tìm thấy học kỳ/năm học hiện tại.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning); |

**Danh sách đăng ký môn học theo học kỳ năm học**

|  |
| --- |
| CREATE FUNCTION dbo.fn\_DangKyMonHocTheoNamHoc  (  @MaHocKyNamHoc INT  )  RETURNS TABLE  AS  RETURN  (  SELECT  LHP.MaMH,  MH.TenMH,  DKMH.MaLHP,  DKMH.MaSV,  SV.HoTen  FROM DangKyMonHoc DKMH  INNER JOIN SinhVien SV ON DKMH.MaSV = SV.MaSV  INNER JOIN LopHocPhan LHP ON DKMH.MaLHP = LHP.MaLHP AND LHP.MaHocKyNamHoc = DKMH.MaHocKyNamHoc  INNER JOIN MonHoc MH ON LHP.MaMH = MH.MaMH  WHERE DKMH.MaHocKyNamHoc = @MaHocKyNamHoc  );  GO |

C#;

|  |
| --- |
| int maHocKyNamHoc = Convert.ToInt32(cbbNamHoc.SelectedValue);  string queryLopHocPhan = $"SELECT \* FROM dbo.fn\_DangKyMonHocTheoNamHoc({maHocKyNamHoc})";  DataTable dt = frmAdmin.getData(queryLopHocPhan);  gcDanhSachSV.DataSource = dt; |

**Danh sách giảng dạy theo theo học kỳ năm học**

|  |
| --- |
| CREATE FUNCTION dbo.fn\_LopHocPhanTheoNamHoc  (  @MaHocKyNamHoc INT  )  RETURNS TABLE  AS  RETURN  (  SELECT  LHP.MaMH,  MH.TenMH,  LHP.MaLHP,  GV.MaGV,  GV.HoTenGV,  MH.MaKhoa,  GV.Email  FROM LopHocPhan LHP  INNER JOIN MonHoc MH ON LHP.MaMH = MH.MaMH  INNER JOIN GiangVien GV ON LHP.MaGV = GV.MaGV  INNER JOIN HocKyNamHoc HKNH ON LHP.MaHocKyNamHoc = HKNH.MaHocKyNamHoc  WHERE LHP.MaHocKyNamHoc = @MaHocKyNamHoc AND GV.TrangThai =0    ); |

C#:

|  |
| --- |
| int maHocKyNamHoc = Convert.ToInt32(cbbNamHoc.SelectedValue);  string queryLopHocPhan = $"SELECT \* FROM dbo.fn\_LopHocPhanTheoNamHoc({maHocKyNamHoc})";  DataTable dt = frmAdmin.getData(queryLopHocPhan);    gcDanhSachSV.DataSource = dt; |

**Bảng điểm theo theo học kỳ năm học**

|  |
| --- |
| CREATE FUNCTION dbo.fn\_ChiTietHocPhan (@MaHocKyNamHoc INT)  RETURNS TABLE  AS  RETURN  (  SELECT  CTHP.MaSV,  SV.HoTen,  LHP.MaMH,  MH.TenMH,  CTHP.DiemGK,  CTHP.DiemCK,  CTHP.DiemTB,  dbo.fn\_TrangThaiDiemTB(CTHP.DiemTB) AS KetQua  FROM ChiTietHocPhan CTHP  INNER JOIN SinhVien SV  ON CTHP.MaSV = SV.MaSV  INNER JOIN LopHocPhan LHP  ON CTHP.MaLHP = LHP.MaLHP  AND CTHP.MaHocKyNamHoc = LHP.MaHocKyNamHoc  INNER JOIN MonHoc MH  ON LHP.MaMH = MH.MaMH  WHERE CTHP.MaHocKyNamHoc = @MaHocKyNamHoc  );  GO |

C#:

|  |
| --- |
| private void LoadBangDiem()  {  if (isLoading || cbbNamHoc.SelectedValue == null) return;  DataRowView drv = cbbNamHoc.SelectedItem as DataRowView;  if (drv == null)  return;  maHocKyNamHoc = Convert.ToInt32(drv["MaHocKyNamHoc"]);  string query = $"SELECT \* FROM fn\_ChiTietHocPhan({maHocKyNamHoc})";  dt = frmAdmin.getData(query);  if (dt != null && dt.Rows.Count > 0)  gcDanhSachSV.DataSource = dt;  else  {  gcDanhSachSV.DataSource = null;  MessageBox.Show("Không có dữ liệu sinh viên.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  } |

**Bảng điểm theo lớp học phần**

|  |
| --- |
| CREATE FUNCTION fn\_SinhVienVaDiemTheoLopHocPhan  (  @MaLHP VARCHAR(20),  @MaHocKyNamHoc INT  )  RETURNS TABLE  AS  RETURN  (  SELECT  SV.MaSV,  SV.HoTen,  dbo.fn\_FormattedDate(SV.NgaySinh) AS NgaySinh,  SV.GioiTinh,  CTHP.DiemGK,  CTHP.DiemCK,  CTHP.DiemTB,  dbo.fn\_TrangThaiDiemTB(CTHP.DiemTB) AS TrangThai  FROM DangKyMonHoc DKMH  INNER JOIN SinhVien SV  ON DKMH.MaSV = SV.MaSV  INNER JOIN LopHocPhan LHP  ON DKMH.MaLHP = LHP.MaLHP  AND DKMH.MaHocKyNamHoc = LHP.MaHocKyNamHoc  LEFT JOIN ChiTietHocPhan CTHP  ON CTHP.MaSV = SV.MaSV  AND CTHP.MaLHP = LHP.MaLHP  AND CTHP.MaHocKyNamHoc = LHP.MaHocKyNamHoc  WHERE LHP.MaLHP = @MaLHP  AND LHP.MaHocKyNamHoc = @MaHocKyNamHoc  AND SV.TrangThai = 0  );  GO |

C#:

|  |
| --- |
| private void LoadBangDiemTheoLop()  {  if (isLoading || cbbMa.SelectedValue == null) return;  string maLHP = cbbMa.SelectedValue.ToString();  string query = $"SELECT \* FROM fn\_SinhVienVaDiemTheoLopHocPhan('{maLHP}', {maHocKyNamHoc})";  dt = frmGiangVien.getData(query);  if (dt != null && dt.Rows.Count > 0)  gcDanhSach.DataSource = dt;  else  {  gcDanhSach.DataSource = null;  MessageBox.Show("Không có dữ liệu sinh viên cho lớp học phần này.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  } |

**Danh sách môn học**

|  |
| --- |
| CREATE VIEW dbo.v\_MonHoc  AS  SELECT  MaMH,  TenMH,  SoTinChi  FROM MonHoc; |

C#:

|  |
| --- |
| public void RefreshData()  {    dt = frmAdmin.getData("SELECT \* FROM v\_MonHoc;");  if (dt != null)  {  gcDanhSachSV.DataSource = dt;  btnLuu.Enabled = false;  btnHuy.Enabled = false;  gvDanhSachSV.OptionsBehavior.Editable = false;  }  gvDanhSachSV.RefreshData();  } |

**Danh sách môn học lớp học phần theo giảng viên**

|  |
| --- |
| CREATE FUNCTION fn\_DanhSachMonHoc\_GiangVien (  @MaGV VARCHAR(10),  @MaHocKyNamHoc INT  )  RETURNS TABLE  AS  RETURN  (  SELECT  MH.MaMH,  MH.TenMH,  LHP.MaLHP  FROM LopHocPhan LHP  INNER JOIN MonHoc MH ON LHP.MaMH = MH.MaMH  INNER JOIN GiangVien GV ON LHP.MaGV = GV.MaGV  WHERE GV.MaGV = @MaGV  AND LHP.MaHocKyNamHoc = @MaHocKyNamHoc  );  GO |

C#:

|  |
| --- |
| int maHocKyNamHoc = Convert.ToInt32(cbbNamHoc.SelectedValue);  string queryMonHoc = $"SELECT \* FROM fn\_DanhSachMonHoc\_GiangVien('{MaGV}', {maHocKyNamHoc})";  DataTable dt = frmGiangVien.getData(queryMonHoc);  gcDanhSach.DataSource = dt; |

**Danh sách sinh viên theo lớp học phần**

|  |
| --- |
| CREATE FUNCTION fn\_SinhVienTheoLopHocPhan(  @MaLHP VARCHAR(20),  @MaHocKyNamHoc INT  )  RETURNS TABLE  AS  RETURN  (  SELECT  SV.MaSV,  SV.HoTen,  dbo.fn\_FormattedDate(SV.NgaySinh) AS NgaySinh,  SV.GioiTinh  FROM DangKyMonHoc DKMH  INNER JOIN SinhVien SV ON DKMH.MaSV = SV.MaSV  INNER JOIN LopHocPhan LHP ON DKMH.MaLHP = LHP.MaLHP  WHERE LHP.MaLHP = @MaLHP  AND LHP.MaHocKyNamHoc = @MaHocKyNamHoc  AND SV.TrangThai = 0  );  GO |

C#:

|  |
| --- |
| private void LoadSinhVien()  {  if (isLoading || cbbMa.SelectedValue == null) return;  string maLHP = cbbMa.SelectedValue.ToString();  string query = $"SELECT \* FROM fn\_SinhVienTheoLopHocPhan('{maLHP}', {maHocKyNamHoc})";  dt = frmGiangVien.getData(query);  if (dt != null && dt.Rows.Count > 0)  gcDanhSach.DataSource = dt;  else  {  gcDanhSach.DataSource = null;  MessageBox.Show("Không có dữ liệu sinh viên cho lớp học phần này.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  } |

**Lớp học phần theo giảng viên**

|  |
| --- |
| CREATE PROCEDURE sp\_GetLopHocPhanByGV  @MaHocKyNamHoc INT,  @MaGV VARCHAR(10)  AS  BEGIN  SET NOCOUNT ON;  SELECT DISTINCT MaLHP  FROM LopHocPhan  WHERE MaHocKyNamHoc = @MaHocKyNamHoc  AND MaGV = @MaGV  ORDER BY MaLHP;  END;  GO |

C#:

|  |
| --- |
| private void LoadMaLop()  {  string query = $"EXEC sp\_GetLopHocPhanByGV @MaHocKyNamHoc = {maHocKyNamHoc}, @MaGV = '{MaGV}'";  DataTable dtMa = frmGiangVien.getData(query);  if (dtMa != null && dtMa.Rows.Count > 0)  {  cbbMa.DataSource = dtMa;  cbbMa.DisplayMember = "MaLHP";  cbbMa.ValueMember = "MaLHP";  }  else  {  cbbMa.DataSource = null;  MessageBox.Show("Giảng viên này chưa phụ trách lớp học phần nào trong học kỳ hiện tại.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  } |

**Giảng viên theo khoa khác**

|  |
| --- |
| CREATE PROCEDURE sp\_GetGVTheoKhoa  @MaKhoa VARCHAR(10),  @MaGVHienTai VARCHAR(10)  AS  BEGIN  SET NOCOUNT ON;  SELECT gv.MaGV, gv.HoTenGV  FROM GiangVien gv  LEFT JOIN Khoa k  ON gv.MaKhoa = k.MaKhoa  WHERE gv.MaKhoa = @MaKhoa  AND gv.TrangThai = 0  AND gv.MaGV <> @MaGVHienTai;  END;  GO |

C#:

|  |
| --- |
| void LoadGV(string maKhoa, string maGVcu)  {  string query = $"EXEC sp\_GetGVTheoKhoa @MaKhoa = '{maKhoa}', @MaGVHienTai = '{maGVcu}'";  DataTable dtGV = frmAdmin.getData(query);  repoLookUpGiangVien.DataSource = null;  repoLookUpGiangVien.DataSource = dtGV;  repoLookUpGiangVien.DisplayMember = "HoTenGV";  repoLookUpGiangVien.ValueMember = "MaGV";  repoLookUpGiangVien.NullText = "Chọn giảng viên thay thế";  repoLookUpGiangVien.CustomDisplayText += (s, e) =>  {  e.DisplayText = "Chọn giảng viên thay thế";  };  } |

**Công thức tính điểm mới nhất**

|  |
| --- |
| CREATE PROCEDURE dbo.sp\_LayCongThucTinhDiemMoiNhat  AS  BEGIN  SET NOCOUNT ON;  SELECT TOP 1  Ma,  TiLeGK,  TiLeCK  FROM CongThucTinhDiem  ORDER BY Ma DESC;  END  GO |

C#:

|  |
| --- |
| public void RefreshData()  {  try  {  string sql = "EXEC dbo.sp\_LayCongThucTinhDiemMoiNhat";  DataTable dt = frmAdmin.getData(sql);  if (dt != null && dt.Rows.Count > 0)  {  DataRow row = dt.Rows[0];  txtTiLeGK.Text = row["TiLeGK"].ToString();  txtTiLeCK.Text = row["TiLeCK"].ToString();  }  else  {  txtTiLeGK.Text = "";  txtTiLeCK.Text = "";  }  }  catch (Exception ex)  {  MessageBox.Show("Lỗi khi tải dữ liệu: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  txtTiLeGK.Text = "";  txtTiLeCK.Text = "";  }  }  } |

**Lớp học khác cùng môn cùng giảng viên**

|  |
| --- |
| CREATE PROCEDURE sp\_LayLopHocPhanKhac  @MaHocKyNamHoc INT,  @MaGV VARCHAR(10),  @MaLHPHienTai VARCHAR(20)  AS  BEGIN  SET NOCOUNT ON;  SELECT MaLHP  FROM LopHocPhan  WHERE MaHocKyNamHoc = @MaHocKyNamHoc  AND MaGV = @MaGV  AND MaMH = (SELECT MaMH FROM LopHocPhan WHERE MaLHP = @MaLHPHienTai AND MaHocKyNamHoc = @MaHocKyNamHoc)  AND MaLHP <> @MaLHPHienTai  ORDER BY MaLHP;  END; |

C#:

|  |
| --- |
| void LoadLopChuyen(string maLHPHienTai, string maGV)  {  string query = $"EXEC sp\_LayLopHocPhanKhac @MaHocKyNamHoc = {maHocKyNamHoc}, @MaGV = '{maGV}', @MaLHPHienTai = '{maLHPHienTai}'";  DataTable dtLop = frmGiangVien.getData(query);  barListItem.Strings.Clear();  if (dtLop != null && dtLop.Rows.Count > 0)  {  foreach (DataRow row in dtLop.Rows)  {  barListItem.Strings.Add(row["MaLHP"].ToString());  }  }    } |

**Thông tin giảng dạy theo giảng viên**

|  |
| --- |
| CREATE FUNCTION fn\_DanhSachMonHoc\_GiangVien (  @MaGV VARCHAR(10),  @MaHocKyNamHoc INT  )  RETURNS TABLE  AS  RETURN  (  SELECT  MH.MaMH,  MH.TenMH,  LHP.MaLHP  FROM LopHocPhan LHP  INNER JOIN MonHoc MH ON LHP.MaMH = MH.MaMH  INNER JOIN GiangVien GV ON LHP.MaGV = GV.MaGV  WHERE GV.MaGV = @MaGV  AND LHP.MaHocKyNamHoc = @MaHocKyNamHoc  );  GO |

C#:

|  |
| --- |
| int maHocKyNamHoc = Convert.ToInt32(cbbNamHoc.SelectedValue);  string queryMonHoc = $"SELECT \* FROM fn\_DanhSachMonHoc\_GiangVien('{MaGV}', {maHocKyNamHoc})";  DataTable dt = frmGiangVien.getData(queryMonHoc);  gcDanhSach.DataSource = dt; |

**Danh sách sinh viên lớp đại học**

|  |
| --- |
| CREATE FUNCTION fn\_SinhVienTheoLop (@LopSV VARCHAR(20))  RETURNS TABLE  AS  RETURN  (  SELECT MaSV,HoTen  FROM SinhVien  WHERE LopSV = @LopSV  );  GO |

C#:

|  |
| --- |
| private void LoadTreeView()  {  if (isLoading || cbbMa.SelectedValue == null) return;  treeViewTen.Nodes.Clear();  string selectedLop = cbbMa.SelectedValue.ToString();  string query = $"SELECT MaSV, HoTen FROM dbo.fn\_SinhVienTheoLop('{selectedLop}')";  DataTable dtSV = frmAdmin.getData(query);  foreach (DataRow row in dtSV.Rows)  {  string nodeText = $"{row["MaSV"]} - {row["HoTen"]}";  treeViewTen.Nodes.Add(nodeText);  }  } |

**Thông tin chi tiết sinh viên có điểm rèn luyện**

|  |
| --- |
| CREATE VIEW vw\_ThongTinChiTietSV AS  SELECT  sv.MaSV,  sv.HoTen,  dbo.fn\_FormattedDate( sv.NgaySinh) as NgaySinh,  sv.NoiSinh,  sv.GioiTinh,  ISNULL(drl.Diem, 0) AS DiemRenLuyen  FROM SinhVien sv  LEFT JOIN DiemRenLuyen drl ON sv.MaSV = drl.MaSV;  GO |

C#:

|  |
| --- |
| string queryInfo = $"SELECT \* FROM vw\_ThongTinChiTietSV WHERE MaSV = '{maSV}'";  DataTable dtInfo = frmAdmin.getData(queryInfo);  if (dtInfo != null && dtInfo.Rows.Count > 0)  {  DataRow row = dtInfo.Rows[0];  lblMa.Text = row["MaSV"].ToString();  lblTen.Text = row["HoTen"].ToString();  lblNS.Text = row["NgaySinh"].ToString();  lblDRL.Text = row["DiemRenLuyen"] == DBNull.Value ? "0" :  row["DiemRenLuyen"].ToString();  lblNoiSinh.Text = row["NoiSinh"].ToString();  lblGioiTinh.Text = row["GioiTinh"].ToString();  } |

**Chi tiết điểm sinh viên**

|  |
| --- |
| CREATE FUNCTION dbo.fn\_ChiTietDiemSV(@MaSV VARCHAR(10))  RETURNS TABLE  AS  RETURN  (  SELECT  mh.MaMH,  mh.TenMH,  mh.SoTinChi,  cthp.DiemTB AS DiemHe10,  dbo.fn\_QuyDoiDiemHe4(cthp.DiemTB) AS DiemHe4,  dbo.fn\_QuyDoiDiemChu(cthp.DiemTB) AS DiemChu,  dbo.fn\_TrangThaiDiemTB(cthp.DiemTB) AS TrangThai  FROM DangKyMonHoc DKMH  INNER JOIN LopHocPhan LHP  ON DKMH.MaLHP = LHP.MaLHP  AND DKMH.MaHocKyNamHoc = LHP.MaHocKyNamHoc  INNER JOIN MonHoc mh  ON LHP.MaMH = mh.MaMH  OUTER APPLY  (  SELECT TOP 1 CTHP.DiemTB  FROM ChiTietHocPhan CTHP  WHERE CTHP.MaSV = @MaSV  AND CTHP.MaLHP = LHP.MaLHP  AND CTHP.MaHocKyNamHoc = LHP.MaHocKyNamHoc  ORDER BY CTHP.DiemTB DESC  ) AS cthp  WHERE DKMH.MaSV = @MaSV  );  GO |

C#:

|  |
| --- |
| string query = $"SELECT \* FROM fn\_ChiTietDiemSV('{maSV}')";  DataTable dtChiTiet = frmAdmin.getData(query);  if (isLoading || cbbMa.SelectedValue == null) return;  string selectedLop = cbbMa.SelectedValue.ToString();  gcDanhSach.DataSource = dtChiTiet; |

**Danh sách tài khoản**

|  |
| --- |
| CREATE VIEW vw\_ThongTinTaiKhoan  AS  SELECT  tk.MaTK,  tk.TenDangNhap,  tk.MatKhau,  tk.TrangThai,  tk.ThoiGian,  r.Roleid,  tk.MaGV  FROM TaiKhoan tk  LEFT JOIN Roles r ON tk.Roleid = r.Roleid  LEFT JOIN GiangVien gv ON tk.MaGV = gv.MaGV;  GO |

C#:

|  |
| --- |
| private void LoadData()  {  try  {  dtTaiKhoan = frmAdmin.getData("SELECT \* FROM vw\_ThongTinTaiKhoan");  if (dtTaiKhoan != null && dtTaiKhoan.Rows.Count > 0)  {  gcDanhSach.DataSource = dtTaiKhoan;  }  else  {  MessageBox.Show("không có dữ liệu hoặc rỗng!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  }  catch (Exception ex)  {  MessageBox.Show($"Lỗi khi load dữ liệu: {ex.Message} StackTrace: {ex.StackTrace}", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

## Chức năng quản lý

**Chức năng thêm sinh viên**

|  |
| --- |
| CREATE PROCEDURE dbo.sp\_ThemSinhVien  @MaSV VARCHAR(10),  @HoTen NVARCHAR(100),  @LopSV VARCHAR(20),  @NgaySinh DATE = NULL,  @NoiSinh NVARCHAR(100) = NULL,  @GioiTinh NVARCHAR(10) = NULL  AS  BEGIN  SET NOCOUNT ON  BEGIN TRY  BEGIN TRANSACTION  IF EXISTS (SELECT 1 FROM SinhVien WHERE MaSV = @MaSV)  BEGIN  RAISERROR('Mã sinh viên ''%s'' đã tồn tại!', 16, 1, @MaSV)  RETURN  END  INSERT INTO SinhVien (MaSV, HoTen, LopSV, NgaySinh, NoiSinh, GioiTinh)  VALUES (@MaSV, @HoTen, @LopSV, @NgaySinh, @NoiSinh, @GioiTinh)  COMMIT TRANSACTION  END TRY  BEGIN CATCH  IF @@TRANCOUNT > 0  ROLLBACK TRANSACTION  DECLARE @ErrMsg NVARCHAR(4000) = ERROR\_MESSAGE();  RAISERROR(@ErrMsg, 16, 1);  END CATCH  END  GO |

C#;

|  |
| --- |
| foreach (DataRow row in changes.Rows)  {  if (row.RowState == DataRowState.Added)  {  string maSV = row["MaSV"].ToString().Trim();  string hoTen = row["HoTen"].ToString().Trim();  string lopSV = row["LopSV"].ToString().Trim();  string ngaySinh = row["NgaySinh"] == DBNull.Value ? "NULL" :  $"'{Convert.ToDateTime(row["NgaySinh"]).ToString("yyyy-MM-dd")}'";  string noiSinh = row["NoiSinh"]?.ToString() ?? "";  string gioiTinh = row["GioiTinh"]?.ToString() ?? "";  if (string.IsNullOrWhiteSpace(maSV) || string.IsNullOrWhiteSpace(hoTen) || string.IsNullOrWhiteSpace(lopSV))  {  MessageBox.Show("Mã sinh viên, Họ tên, Lớp SV không được để trống!");  return;  }  string query = $@"EXEC sp\_ThemSinhVien  @MaSV = '{maSV}',  @HoTen = N'{hoTen}',  @LopSV = '{lopSV}',  @NgaySinh = {ngaySinh},  @NoiSinh = {(string.IsNullOrWhiteSpace(noiSinh) ? "NULL" : $"N'{noiSinh}'")},  @GioiTinh = {(string.IsNullOrWhiteSpace(gioiTinh) ? "NULL" : $"N'{gioiTinh}'")}";  frmAdmin.executeQuery(query);  } |

**Chức năng cập nhập sinh viên**

|  |
| --- |
| CREATE PROCEDURE dbo.sp\_CapNhatSinhVien  @MaSV VARCHAR(10),  @HoTen NVARCHAR(100),  @LopSV VARCHAR(20),  @NgaySinh DATE = NULL,  @NoiSinh NVARCHAR(100) = NULL,  @GioiTinh NVARCHAR(10) = NULL,  @TrangThai BIT  AS  BEGIN  SET NOCOUNT ON;  BEGIN TRY  BEGIN TRANSACTION;  UPDATE SinhVien  SET HoTen = @HoTen,  LopSV = @LopSV,  NgaySinh = @NgaySinh,  NoiSinh = @NoiSinh,  GioiTinh = @GioiTinh,  TrangThai = @TrangThai  WHERE MaSV = @MaSV;  IF @@ROWCOUNT = 0  BEGIN  RAISERROR('Không tìm thấy sinh viên có mã ''%s''', 16, 1, @MaSV);  RETURN;  END  COMMIT TRANSACTION;  END TRY  BEGIN CATCH  IF @@TRANCOUNT > 0  ROLLBACK TRANSACTION;  DECLARE @ErrMsg NVARCHAR(4000) = ERROR\_MESSAGE();  RAISERROR(@ErrMsg, 16, 1);;  END CATCH  END |

C#:

|  |
| --- |
| if (row.RowState == DataRowState.Modified)  {  string maSV = row["MaSV"].ToString().Trim();  string hoTen = row["HoTen"].ToString().Trim();  string lopSV = row["LopSV"].ToString().Trim();  string ngaySinh = row["NgaySinh"] == DBNull.Value ? "NULL" :  $"'{Convert.ToDateTime(row["NgaySinh"]).ToString("yyyy-MM-dd")}'";  string noiSinh = row["NoiSinh"]?.ToString() ?? "";  string gioiTinh = row["GioiTinh"]?.ToString() ?? "";  int trangThai = row["TrangThai"] != DBNull.Value ? Convert.ToInt32(row["TrangThai"]) : 0;  if (string.IsNullOrWhiteSpace(maSV) || string.IsNullOrWhiteSpace(hoTen) || string.IsNullOrWhiteSpace(lopSV))  {  MessageBox.Show("Mã sinh viên, Họ tên, Lớp SV không được để trống!");  return;  }  string query = $@"EXEC sp\_CapNhatSinhVien  @MaSV = '{maSV}',  @HoTen = N'{hoTen}',  @LopSV = '{lopSV}',  @NgaySinh = {ngaySinh},  @NoiSinh = {(string.IsNullOrWhiteSpace(noiSinh) ? "NULL" : $"N'{noiSinh}'")},  @GioiTinh = {(string.IsNullOrWhiteSpace(gioiTinh) ? "NULL" : $"N'{gioiTinh}'")},  @TrangThai = {trangThai} ";  frmAdmin.executeQuery(query);    } } |

**Chức năng xoá sinh viên**

|  |
| --- |
| CREATE PROCEDURE dbo.sp\_XoaSinhVien  @MaSV VARCHAR(10)  AS  BEGIN  SET NOCOUNT ON;  BEGIN TRY  BEGIN TRANSACTION;  DELETE FROM SinhVien  WHERE MaSV = @MaSV;  COMMIT TRANSACTION;  END TRY  BEGIN CATCH  IF @@TRANCOUNT > 0  ROLLBACK TRANSACTION;  THROW;  END CATCH  END; |

C#:

|  |
| --- |
| for (int i = selectedRows.Length - 1; i >= 0; i--)  {  DataRow row = gvDanhSachSV.GetDataRow(selectedRows[i]);  if (row != null && row.RowState == DataRowState.Added)  {  row.Delete();  gvDanhSachSV.DeleteRow(selectedRows[i]);  }  else  {  string maSV = gvDanhSachSV.GetRowCellValue(selectedRows[i], "MaSV").ToString();  string query = $"EXEC sp\_XoaSinhVien @MaSV = '{maSV}'";  frmAdmin.executeQuery(query);  gvDanhSachSV.DeleteRow(selectedRows[i]);  }  } |

**Chức năng thêm giảng viên**

|  |
| --- |
| CREATE PROCEDURE dbo.sp\_ThemGiangVien  @MaGV VARCHAR(10),  @HoTenGV NVARCHAR(100),  @HocVi NVARCHAR(50) = NULL,  @MaKhoa NVARCHAR(10) = NULL,  @Email NVARCHAR(100) = NULL,  @DienThoai NVARCHAR(15) = NULL  AS  BEGIN  SET NOCOUNT ON;  BEGIN TRY  BEGIN TRANSACTION;  IF EXISTS (SELECT 1 FROM GiangVien WHERE MaGV = @MaGV)  BEGIN  RAISERROR('Mã giảng viên ''%s'' đã tồn tại!', 16, 1, @MaGV);  RETURN;  END  INSERT INTO GiangVien (MaGV, HoTenGV, HocVi, MaKhoa, Email, DienThoai)  VALUES (@MaGV, @HoTenGV, @HocVi, @MaKhoa, @Email, @DienThoai);  COMMIT TRANSACTION;  END TRY  BEGIN CATCH  IF @@TRANCOUNT > 0  ROLLBACK TRANSACTION;  DECLARE @ErrMsg NVARCHAR(4000) = ERROR\_MESSAGE();  RAISERROR(@ErrMsg, 16, 1);  END CATCH  END  GO |

C#;

|  |
| --- |
| try  {  foreach (DataRow row in changes.Rows)  {  if (row.RowState == DataRowState.Added)  {  string maGV = row["MaGV"].ToString().Trim();  string hoTenGV = row["HoTenGV"].ToString().Trim();  string hocVi = row["HocVi"]?.ToString() ?? "";  string makhoa = row["MaKhoa"]?.ToString() ?? "";  string email = row["Email"]?.ToString() ?? "";  string dienThoai = row["DienThoai"]?.ToString() ?? "";  if (string.IsNullOrWhiteSpace(maGV) || string.IsNullOrWhiteSpace(hoTenGV) || string.IsNullOrWhiteSpace(makhoa))  {  MessageBox.Show("Mã giảng viên, Họ tên giảng viên và khoa không được để trống!");  return;  }  // Gọi stored procedure để thêm giảng viên  string query = $@"EXEC sp\_ThemGiangVien  @MaGV = '{maGV}',  @HoTenGV = N'{hoTenGV}',  @HocVi = {(string.IsNullOrWhiteSpace(hocVi) ? "NULL" : $"N'{hocVi}'")},  @MaKhoa = {(string.IsNullOrWhiteSpace(makhoa) ? "NULL" : $"N'{makhoa}'")},  @Email = {(string.IsNullOrWhiteSpace(email) ? "NULL" : $"N'{email}'")},  @DienThoai = {(string.IsNullOrWhiteSpace(dienThoai) ? "NULL" : $"N'{dienThoai}'")}";  frmAdmin.executeQuery(query);  } |

**Chức năng cập nhập giảng viên**

|  |
| --- |
| CREATE PROCEDURE dbo.sp\_CapNhatGiangVien  @MaGV VARCHAR(10),  @HoTenGV NVARCHAR(100),  @HocVi NVARCHAR(50) = NULL,  @MaKhoa NVARCHAR(10) = NULL,  @Email NVARCHAR(100) = NULL,  @DienThoai NVARCHAR(15) = NULL,  @TrangThai BIT  AS  BEGIN  SET NOCOUNT ON;  BEGIN TRY  BEGIN TRANSACTION;  UPDATE GiangVien  SET HoTenGV = @HoTenGV,  HocVi = @HocVi,  MaKhoa = @MaKhoa,  Email = @Email,  DienThoai = @DienThoai,  TrangThai = @TrangThai  WHERE MaGV = @MaGV;  IF @@ROWCOUNT = 0  BEGIN  RAISERROR('Không tìm thấy giảng viên có mã ''%s''', 16, 1, @MaGV);  RETURN;  END  COMMIT TRANSACTION;  END TRY  BEGIN CATCH  IF @@TRANCOUNT > 0  ROLLBACK TRANSACTION;  DECLARE @ErrMsg NVARCHAR(4000) = ERROR\_MESSAGE();  RAISERROR(@ErrMsg, 16, 1);  END CATCH  END  GO |

C#:

|  |
| --- |
| if (row.RowState == DataRowState.Modified)  {  string maGV = row["MaGV"].ToString().Trim();  string hoTenGV = row["HoTenGV"].ToString().Trim();  string hocVi = row["HocVi"]?.ToString() ?? "";  string makhoa = row["MaKhoa"]?.ToString() ?? "";  string email = row["Email"]?.ToString() ?? "";  string dienThoai = row["DienThoai"]?.ToString() ?? "";  int trangThai = row["TrangThai"] != DBNull.Value ? Convert.ToInt32(row["TrangThai"]) : 0;  // Kiểm tra dữ liệu bắt buộc  if (string.IsNullOrWhiteSpace(maGV) || string.IsNullOrWhiteSpace(hoTenGV))  {  MessageBox.Show("Mã giảng viên và Họ tên giảng viên không được để trống!");  return;  }  string query = $@"EXEC sp\_CapNhatGiangVien  @MaGV = '{maGV}',  @HoTenGV = N'{hoTenGV}',  @HocVi = {(string.IsNullOrWhiteSpace(hocVi) ? "NULL" : $"N'{hocVi}'")},  @MaKhoa = {(string.IsNullOrWhiteSpace(makhoa) ? "NULL" : $"N'{makhoa}'")},  @Email = {(string.IsNullOrWhiteSpace(email) ? "NULL" : $"N'{email}'")},  @DienThoai = {(string.IsNullOrWhiteSpace(dienThoai) ? "NULL" : $"N'{dienThoai}'")},  @TrangThai = { trangThai} ";  frmAdmin.executeQuery(query);  } |

**Chức năng xoá giảng viên**

|  |
| --- |
| CREATE PROCEDURE dbo.sp\_XoaGiangVien  @MaGV VARCHAR(10)  AS  BEGIN  SET NOCOUNT ON;  BEGIN TRY  BEGIN TRANSACTION;  -- Đặt MaGV về NULL trong LopHocPhan thay vì xóa  UPDATE LopHocPhan  SET MaGV = NULL  WHERE MaGV = @MaGV;    -- Xóa bản ghi trong TaiKhoan  DELETE FROM TaiKhoan  WHERE MaGV = @MaGV;    -- Xóa bản ghi trong GiangVien  DELETE FROM GiangVien  WHERE MaGV = @MaGV;    COMMIT TRANSACTION;  END TRY  BEGIN CATCH  IF @@TRANCOUNT > 0  ROLLBACK TRANSACTION;  THROW;  END CATCH  END  GO |

C#:

|  |
| --- |
| for (int i = selectedRows.Length - 1; i >= 0; i--)  {  string maGV = gvDanhSachSV.GetRowCellValue(selectedRows[i], "MaGV").ToString();  string query = $"EXEC sp\_XoaGiangVien @MaGV = '{maGV}'";  frmAdmin.executeQuery(query);  gvDanhSachSV.DeleteRow(selectedRows[i]);  } |

**Cập nhập điểm sinh viên**

|  |
| --- |
| CREATE PROCEDURE dbo.sp\_CapNhatDiemHocPhan  @MaSV VARCHAR(10),  @MaMH VARCHAR(20),  @MaHocKyNamHoc INT,  @DiemGK DECIMAL(4,2),  @DiemCK DECIMAL(4,2)  AS  BEGIN  SET NOCOUNT ON;  BEGIN TRY  BEGIN TRAN;  UPDATE CTHP  SET DiemGK = @DiemGK,  DiemCK = @DiemCK  FROM ChiTietHocPhan CTHP  INNER JOIN LopHocPhan LHP  ON CTHP.MaLHP = LHP.MaLHP  AND CTHP.MaHocKyNamHoc = LHP.MaHocKyNamHoc  WHERE CTHP.MaSV = @MaSV  AND LHP.MaMH = @MaMH  AND LHP.MaHocKyNamHoc = @MaHocKyNamHoc;  COMMIT TRAN;  END TRY  BEGIN CATCH  IF @@TRANCOUNT > 0  ROLLBACK TRAN;  THROW;  END CATCH  END  GO |

C#:

|  |
| --- |
| foreach (DataRow row in changes.Rows)  {  string maSV = row["MaSV"].ToString();  string maMH = row["MaMH"].ToString();  decimal? diemGK = row["DiemGK"] != DBNull.Value ? Convert.ToDecimal(row["DiemGK"]) : (decimal?)null;  decimal? diemCK = row["DiemCK"] != DBNull.Value ? Convert.ToDecimal(row["DiemCK"]) : (decimal?)null;  if ((diemGK.HasValue && (diemGK < 0 || diemGK > 10)) ||  (diemCK.HasValue && (diemCK < 0 || diemCK > 10)))  {  MessageBox.Show($"Điểm của sinh viên {maSV} không hợp lệ (0-10).", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Warning);  return;  }  string diemGKValue = diemGK.HasValue ? diemGK.Value.ToString() : "NULL";  string diemCKValue = diemCK.HasValue ? diemCK.Value.ToString() : "NULL";  string query = $@"EXEC sp\_CapNhatDiemHocPhan  @MaSV = '{maSV}',  @MaMH = '{maMH}',  @MaHocKyNamHoc = {maHocKyNamHoc},  @DiemGK = {diemGKValue},  @DiemCK = {diemCKValue}";  frmAdmin.executeQuery(query); |

**Xoá đăng ký môn học**

|  |
| --- |
| CREATE PROCEDURE sp\_XoaDangKyMonHoc  @MaSV VARCHAR(10),  @MaLHP VARCHAR(20),  @MaHocKyNamHoc INT  AS  BEGIN  SET NOCOUNT ON;  DELETE FROM ChiTietHocPhan  WHERE MaSV = @MaSV  AND MaLHP = @MaLHP  AND MaHocKyNamHoc = @MaHocKyNamHoc;  DELETE FROM DangKyMonHoc  WHERE MaSV = @MaSV  AND MaLHP = @MaLHP  AND MaHocKyNamHoc = @MaHocKyNamHoc;  END;  GO |

C#:

|  |
| --- |
| if (result == DialogResult.Yes)  {  string query = $"EXEC sp\_XoaDangKyMonHoc @MaSV = '{maSV}', @MaLHP = '{maLHP}', @MaHocKyNamHoc = {maHocKyNamHoc}";  try  {  frmGiangVien.executeQuery(query);  dt.Rows.Remove(row);  gvDanhSach.RefreshData();  MessageBox.Show("Xóa sinh viên thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  catch (Exception ex)  {  MessageBox.Show("Xóa thất bại: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

**Chuyển lớp học phần**

|  |
| --- |
| CREATE PROCEDURE sp\_ChuyenLopHocPhan  @MaSV VARCHAR(10),  @MaLHPNguon VARCHAR(20),  @MaLHPDich VARCHAR(20),  @MaHocKyNamHoc INT  AS  BEGIN  SET NOCOUNT ON;  BEGIN TRANSACTION;  BEGIN TRY  UPDATE DangKyMonHoc  SET MaLHP = @MaLHPDich  WHERE MaSV = @MaSV  AND MaLHP = @MaLHPNguon  AND MaHocKyNamHoc = @MaHocKyNamHoc;  COMMIT TRANSACTION;  END TRY  BEGIN CATCH  ROLLBACK TRANSACTION;  THROW;  END CATCH  END;  GO |

C#:

|  |
| --- |
| if (result == DialogResult.Yes)  {  string query = $"EXEC sp\_ChuyenLopHocPhan @MaSV = '{maSV}', @MaLHPNguon = '{maLHPNguon}', @MaLHPDich = '{maLHPDich}', @MaHocKyNamHoc = {maHocKyNamHoc}";  try  {  frmGiangVien.executeQuery(query);  MessageBox.Show($"Đã chuyển sinh viên {maSV} từ lớp {maLHPNguon} sang lớp {maLHPDich}.", "Thành công");  LoadSinhVien();  }  catch (Exception ex)  {  MessageBox.Show("Lỗi khi chuyển lớp: " + ex.Message, "Lỗi");  }  } |

**Đổi giảng viên lớp học phần**

|  |
| --- |
| CREATE PROCEDURE sp\_ChuyenGiangVien  @MaLHP VARCHAR(20),  @MaHocKyNamHoc INT,  @MaGVNguon VARCHAR(10),  @MaGVDich VARCHAR(10)  AS  BEGIN  SET NOCOUNT ON;  BEGIN TRY  BEGIN TRANSACTION;  UPDATE LopHocPhan  SET MaGV = @MaGVDich  WHERE MaLHP = @MaLHP  AND MaHocKyNamHoc = @MaHocKyNamHoc  AND (MaGV = @MaGVNguon OR (@MaGVNguon IS NULL AND MaGV IS NULL) OR MaGV IS NULL);  COMMIT TRANSACTION;  END TRY  BEGIN CATCH  IF @@TRANCOUNT > 0  ROLLBACK TRANSACTION;  DECLARE @ErrorMessage NVARCHAR(4000) = ERROR\_MESSAGE();  RAISERROR(@ErrorMessage, 16, 1);  END CATCH  END;  GO |

C#:

|  |
| --- |
| if (result == DialogResult.Yes)  {  string maGVCu = row["MaGV"]?.ToString();  string query = $"EXEC sp\_ChuyenGiangVien @MaLHP = '{maLHP}', @MaHocKyNamHoc = {maHocKyNamHoc}, @MaGVNguon = '{maGVCu}', @MaGVDich = '{maGV}'";  try  {  frmAdmin.executeQuery(query);  MessageBox.Show($"Lớp {maLHP} đã đổi giảng viên thành công.", "Thành công");  LoadLHP();  }  catch (Exception ex)  {  MessageBox.Show("Lỗi khi đổi: " + ex.Message, "Lỗi");  }  } |

**Huỷ lớp học phần**

|  |
| --- |
| CREATE PROCEDURE sp\_HuyLopHocPhan  @MaLHP VARCHAR(20),  @MaHocKyNamHoc INT  AS  BEGIN  SET NOCOUNT ON;  BEGIN TRY  BEGIN TRANSACTION;  -- 1. Xóa chi tiết học phần của sinh viên  DELETE FROM ChiTietHocPhan  WHERE MaLHP = @MaLHP AND MaHocKyNamHoc = @MaHocKyNamHoc;  -- 2. Xóa đăng ký môn học của sinh viên  DELETE FROM DangKyMonHoc  WHERE MaLHP = @MaLHP AND MaHocKyNamHoc = @MaHocKyNamHoc;  -- 3. Xóa lớp học phần  DELETE FROM LopHocPhan  WHERE MaLHP = @MaLHP AND MaHocKyNamHoc = @MaHocKyNamHoc;  COMMIT TRANSACTION;  END TRY  BEGIN CATCH  IF @@TRANCOUNT > 0  ROLLBACK TRANSACTION;  DECLARE @ErrorMessage NVARCHAR(4000) = ERROR\_MESSAGE();  RAISERROR(@ErrorMessage, 16, 1);  END CATCH  END;  GO |

C#:

|  |
| --- |
| private void btnHuy\_ItemClick(object sender, ItemClickEventArgs e)  {  int[] selectedRows = gvDanhSachSV.GetSelectedRows();  if (selectedRows.Length > 0)  {  var result = MessageBox.Show($"Bạn có chắc chắn muốn huỷ {selectedRows.Length} lớp học phần đã chọn?",  "Xác nhận huỷ",  MessageBoxButtons.YesNo,  MessageBoxIcon.Question);  if (result == DialogResult.Yes)  {  try  {  for (int i = selectedRows.Length - 1; i >= 0; i--)  {  DataRow row = gvDanhSachSV.GetDataRow(selectedRows[i]);  if (row != null && row.RowState == DataRowState.Added)  {  row.Delete();  gvDanhSachSV.DeleteRow(selectedRows[i]);  }  else  {  string maLHP = gvDanhSachSV.GetRowCellValue(selectedRows[i], "MaLHP").ToString();  string query = $"EXEC sp\_HuyLopHocPhan @MaLHP = '{maLHP}', @MaHocKyNamHoc={maHocKyNamHoc}";  frmAdmin.executeQuery(query);  gvDanhSachSV.DeleteRow(selectedRows[i]);  }  }  MessageBox.Show("Huỷ thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  catch (Exception ex)  {  MessageBox.Show("Lỗi khi huỷ lớp: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  }  }  else  {  MessageBox.Show("Vui lòng chọn ít nhất một lớp để huỷ!",  "Thông báo",  MessageBoxButtons.OK,  MessageBoxIcon.Information);  }  }  } |

**Thêm công thức tính điểm**

|  |
| --- |
| CREATE PROCEDURE sp\_ThemCongThucTinhDiem  @TiLeGK DECIMAL(3,2),  @TiLeCK DECIMAL(3,2)  AS  BEGIN  SET NOCOUNT ON;  INSERT INTO CongThucTinhDiem(TiLeGK, TiLeCK)  VALUES(@TiLeGK, @TiLeCK);  END;  GO |

C#:

|  |
| --- |
| if (decimal.TryParse(txtTiLeGK.Text, out decimal tGK) &&  decimal.TryParse(txtTiLeCK.Text, out decimal tCK) &&  tGK >= 0 && tGK <= 1 && tCK >= 0 && tCK <= 1)  {  try  {  string query = $"EXEC sp\_ThemCongThucTinhDiem @TiLeGK = {tGK}, @TiLeCK = {tCK}";  frmAdmin.executeQuery(query);  MessageBox.Show("Lưu thành công!");  }  catch (Exception ex)  {  MessageBox.Show("Lỗi khi lưu: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  }  else  {  MessageBox.Show("Vui lòng nhập giá trị hợp lệ (0–1).");  }  } |

**Xoá tài khoản**

|  |
| --- |
| CREATE PROCEDURE dbo.sp\_XoaTaiKhoanForce  @MaTK INT  AS  BEGIN  SET NOCOUNT ON;  DECLARE @TenDangNhap NVARCHAR(50);  DECLARE @spid INT;  DECLARE @sql NVARCHAR(MAX);  -- Lấy tên login trước  SELECT @TenDangNhap = TenDangNhap FROM TaiKhoan WHERE MaTK = @MaTK;  IF @TenDangNhap IS NULL  BEGIN  RAISERROR('Không tìm thấy tài khoản có MaTK = %d', 16, 1, @MaTK);  RETURN;  END  -- =====================  -- Kill tất cả session của login (bên ngoài transaction)  -- =====================  DECLARE cur\_sessions CURSOR FOR  SELECT session\_id  FROM sys.dm\_exec\_sessions  WHERE login\_name = @TenDangNhap;  OPEN cur\_sessions;  FETCH NEXT FROM cur\_sessions INTO @spid;  WHILE @@FETCH\_STATUS = 0  BEGIN  SET @sql = 'KILL ' + CAST(@spid AS NVARCHAR(10));  EXEC(@sql);  FETCH NEXT FROM cur\_sessions INTO @spid;  END  CLOSE cur\_sessions;  DEALLOCATE cur\_sessions;  -- =====================  -- Xóa user/login/TaiKhoan trong transaction  -- =====================  BEGIN TRY  BEGIN TRANSACTION;  -- Xóa khỏi tất cả role database  DECLARE @sqlRemove NVARCHAR(MAX) = '';  SELECT @sqlRemove = @sqlRemove + 'ALTER ROLE [' + r.name + '] DROP MEMBER [' + @TenDangNhap + ']; '  FROM sys.database\_role\_members drm  JOIN sys.database\_principals r ON drm.role\_principal\_id = r.principal\_id  JOIN sys.database\_principals u ON drm.member\_principal\_id = u.principal\_id  WHERE u.name = @TenDangNhap;  IF @sqlRemove <> ''  BEGIN  EXEC(@sqlRemove);  END  -- Xóa user database  IF EXISTS (SELECT \* FROM sys.database\_principals WHERE name = @TenDangNhap)  BEGIN  SET @sql = 'DROP USER [' + @TenDangNhap + ']';  EXEC(@sql);  END  -- Xóa login server  IF EXISTS (SELECT \* FROM sys.server\_principals WHERE name = @TenDangNhap)  BEGIN  SET @sql = 'DROP LOGIN [' + @TenDangNhap + ']';  EXEC(@sql);  END  DELETE FROM TaiKhoan WHERE MaTK = @MaTK;  COMMIT TRANSACTION;  END TRY  BEGIN CATCH  IF @@TRANCOUNT > 0  ROLLBACK TRANSACTION;  DECLARE @ErrMsg NVARCHAR(4000) = ERROR\_MESSAGE();  RAISERROR(@ErrMsg, 16, 1);  END CATCH  END  GO |

C#:

|  |
| --- |
| for (int i = selectedRows.Length - 1; i >= 0; i--)  {  DataRow row = gvDanhSach.GetDataRow(selectedRows[i]);  if (row != null && row.RowState == DataRowState.Added)  {  gvDanhSach.DeleteRow(selectedRows[i]);  }  else  {  object maTKValue = gvDanhSach.GetRowCellValue(selectedRows[i], "MaTK");  if (maTKValue == null || maTKValue == DBNull.Value)  {  MessageBox.Show($"Không tìm thấy MaTK cho dòng {selectedRows[i]}", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  continue;  }  int maTK = Convert.ToInt32(maTKValue);  MessageBox.Show($"MaTK = {maTK}");  frmAdmin.executeQuery($"EXEC sp\_XoaTaiKhoanForce @MaTK = {maTK}");  gvDanhSach.DeleteRow(selectedRows[i]);  } } |

**Thêm tài khoản**

|  |
| --- |
| CREATE PROCEDURE dbo.sp\_ThemTaiKhoan  @TenDangNhap NVARCHAR(50),  @MatKhau NVARCHAR(255),  @Roleid INT = NULL,  @MaGV VARCHAR(10) = NULL,  @TrangThai BIT = 1  AS  BEGIN  SET NOCOUNT ON;  BEGIN TRY  BEGIN TRANSACTION;  -- Kiểm tra trùng TenDangNhap  IF EXISTS (SELECT 1 FROM TaiKhoan WHERE TenDangNhap = @TenDangNhap)  BEGIN  RAISERROR('Tên đăng nhập ''%s'' đã tồn tại!', 16, 1, @TenDangNhap);  RETURN;  END  -- Kiểm tra giảng viên đã có tài khoản với vai trò này chưa  IF EXISTS (SELECT 1 FROM TaiKhoan WHERE MaGV = @MaGV AND Roleid = @Roleid)  BEGIN  RAISERROR('Mã giảng viên ''%s'' đã có tài khoản với vai trò này!', 16, 1, @MaGV);  RETURN;  END  -- Thêm bản ghi vào bảng TaiKhoan  INSERT INTO TaiKhoan (TenDangNhap, MatKhau, Roleid, MaGV, TrangThai)  VALUES (@TenDangNhap, @MatKhau, @Roleid, @MaGV, @TrangThai);  DECLARE @sql NVARCHAR(MAX);  -- Tạo LOGIN cho cả Admin và GiangVien  SET @sql = '  IF NOT EXISTS (SELECT \* FROM sys.server\_principals WHERE name = ''' + @TenDangNhap + ''')  CREATE LOGIN [' + @TenDangNhap + '] WITH PASSWORD = ''' + REPLACE(@MatKhau,'''','''''') + ''';  ';  EXEC(@sql);  SET @sql = 'ALTER LOGIN [' + @TenDangNhap + '] WITH DEFAULT\_DATABASE = [QL\_SinhVien];';  EXEC(@sql);  -- Tạo USER cho cả Admin và GiangVien  SET @sql = '  IF NOT EXISTS (SELECT \* FROM sys.database\_principals WHERE name = ''' + @TenDangNhap + ''')  CREATE USER [' + @TenDangNhap + '] FOR LOGIN [' + @TenDangNhap + '];  ALTER LOGIN [' + @TenDangNhap + '] ENABLE;  ';    EXEC(@sql);  SET @sql = 'USE QL\_SinhVien; GRANT CONNECT TO [' + @TenDangNhap + '];';  EXEC(@sql);  IF @Roleid = 1 -- Admin  BEGIN  -- Gán sysadmin  SET @sql = 'ALTER SERVER ROLE [sysadmin] ADD MEMBER [' + @TenDangNhap + '];';  EXEC(@sql);  -- Thêm vào AdminRole  SET @sql = 'ALTER ROLE [AdminRole] ADD MEMBER [' + @TenDangNhap + '];';  EXEC(@sql);  END  ELSE IF @Roleid = 2 -- GiangVien  BEGIN  -- Thêm vào GiangVienRole  SET @sql = 'ALTER ROLE [GiangVienRole] ADD MEMBER [' + @TenDangNhap + '];';  EXEC(@sql);  END  COMMIT TRANSACTION;  END TRY  BEGIN CATCH  IF @@TRANCOUNT > 0  ROLLBACK TRANSACTION  DECLARE @ErrMsg NVARCHAR(4000) = ERROR\_MESSAGE();  RAISERROR(@ErrMsg, 16, 1);  END CATCH  END  GO |

C#:

|  |
| --- |
| if (row.RowState == DataRowState.Added)  {  string tenDangNhap = row["TenDangNhap"]?.ToString() ?? "";  string matKhau = row["MatKhau"]?.ToString() ?? "";  string roleId = row["Roleid"]?.ToString() ?? "";  string maGV = row["MaGV"]?.ToString() ?? "";  int trangThai = row["TrangThai"] == DBNull.Value ? 1: (Convert.ToBoolean(row["TrangThai"]) ? 1 : 0);  if (string.IsNullOrWhiteSpace(tenDangNhap) || string.IsNullOrWhiteSpace(matKhau) || string.IsNullOrWhiteSpace(roleId))  {  MessageBox.Show("Tên đăng nhập, mật khẩu và Role không được để trống!");  return;  }    if (string.IsNullOrWhiteSpace(maGV))  {  MessageBox.Show("Phải có mã giảng viên");  return;  }    string query = $@"EXEC sp\_ThemTaiKhoan  @TenDangNhap = N'{tenDangNhap}',  @MatKhau = N'{matKhau}',  @Roleid = {roleId},  @MaGV = {(string.IsNullOrWhiteSpace(maGV) ? "NULL" : $"'{maGV}'")},  @TrangThai = {trangThai}";    frmAdmin.executeQuery(query);    } |

**Cập nhập tài khoản**

|  |
| --- |
| CREATE PROCEDURE dbo.sp\_CapNhatTaiKhoan  @MaTK INT,  @TenDangNhap NVARCHAR(50) = NULL,  @MatKhau NVARCHAR(255) = NULL,  @Roleid INT = NULL,  @MaGV VARCHAR(10) = NULL,  @TrangThai BIT = NULL  AS  BEGIN  SET NOCOUNT ON;  DECLARE @TenDangNhapCu NVARCHAR(50);  DECLARE @RoleidCu INT;  DECLARE @LoginName NVARCHAR(50);  DECLARE @sql NVARCHAR(MAX);  DECLARE @PwdSafe NVARCHAR(255);  BEGIN TRY  BEGIN TRANSACTION;  -- Lấy thông tin cũ  SELECT @TenDangNhapCu = TenDangNhap, @RoleidCu = Roleid  FROM TaiKhoan  WHERE MaTK = @MaTK;  IF @TenDangNhapCu IS NULL  BEGIN  RAISERROR('Không tìm thấy tài khoản có MaTK = %d', 16, 1, @MaTK);  RETURN;  END  -- Kiểm tra MaGV hợp lệ nếu được cung cấp  IF @MaGV IS NOT NULL AND NOT EXISTS (SELECT 1 FROM GiangVien WHERE MaGV = @MaGV)  BEGIN  RAISERROR('Mã giảng viên ''%s'' không tồn tại!', 16, 1, @MaGV);  RETURN;  END  -- Cập nhật bảng TaiKhoan  UPDATE TaiKhoan  SET TenDangNhap = ISNULL(@TenDangNhap, TenDangNhap),  MatKhau = ISNULL(@MatKhau, MatKhau),  Roleid = ISNULL(@Roleid, Roleid),  MaGV = ISNULL(@MaGV, MaGV),  TrangThai = ISNULL(@TrangThai, TrangThai)  WHERE MaTK = @MaTK;  SET @LoginName = ISNULL(@TenDangNhap, @TenDangNhapCu);  -- Nếu TenDangNhap thay đổi, đổi tên login và user  IF @TenDangNhap IS NOT NULL AND @TenDangNhap <> @TenDangNhapCu  BEGIN  IF EXISTS (SELECT \* FROM sys.server\_principals WHERE name = @TenDangNhapCu)  BEGIN  SET @sql = 'ALTER LOGIN ' + QUOTENAME(@TenDangNhapCu) + ' WITH NAME = ' + QUOTENAME(@LoginName) + ';';  PRINT @sql; -- In lệnh SQL để kiểm tra  EXEC(@sql);  END  IF EXISTS (SELECT \* FROM sys.database\_principals WHERE name = @TenDangNhapCu)  BEGIN  SET @sql = 'ALTER USER ' + QUOTENAME(@TenDangNhapCu) + ' WITH NAME = ' + QUOTENAME(@LoginName) + ';';  PRINT @sql; -- In lệnh SQL để kiểm tra  EXEC(@sql);  END  END  -- Nếu MatKhau thay đổi, cập nhật login  IF @MatKhau IS NOT NULL  BEGIN  SET @PwdSafe = REPLACE(@MatKhau, '''', ''''''); -- escape dấu nháy đơn  IF EXISTS (SELECT \* FROM sys.server\_principals WHERE name = @LoginName)  BEGIN  SET @sql = 'ALTER LOGIN ' + QUOTENAME(@LoginName) + ' WITH PASSWORD = ''' + @PwdSafe + ''';';  PRINT @sql; -- In lệnh SQL để kiểm tra  EXEC(@sql);  END  END  -- Nếu Roleid thay đổi, cập nhật role  IF @Roleid IS NOT NULL AND @Roleid <> @RoleidCu  BEGIN  -- Xóa khỏi tất cả role database  DECLARE @sqlRemove NVARCHAR(MAX) = '';  SELECT @sqlRemove = @sqlRemove + 'ALTER ROLE ' + QUOTENAME(r.name) + ' DROP MEMBER ' + QUOTENAME(@LoginName) + '; '  FROM sys.database\_role\_members drm  JOIN sys.database\_principals r ON drm.role\_principal\_id = r.principal\_id  JOIN sys.database\_principals u ON drm.member\_principal\_id = u.principal\_id  WHERE u.name = @LoginName;  IF @sqlRemove <> ''  BEGIN  PRINT @sqlRemove; -- In lệnh SQL để kiểm tra  EXEC(@sqlRemove);  END  -- Cập nhật role mới  IF @Roleid = 1 -- Admin  BEGIN  -- Gán sysadmin nếu chưa có  IF NOT EXISTS (  SELECT \* FROM sys.server\_role\_members srm  JOIN sys.server\_principals sp ON srm.member\_principal\_id = sp.principal\_id  WHERE sp.name = @LoginName AND srm.role\_principal\_id = SUSER\_ID('sysadmin')  )  BEGIN  SET @sql = 'ALTER SERVER ROLE [sysadmin] ADD MEMBER ' + QUOTENAME(@LoginName) + ';';  PRINT @sql; -- In lệnh SQL để kiểm tra  EXEC(@sql);  END  SET @sql = 'ALTER ROLE [AdminRole] ADD MEMBER ' + QUOTENAME(@LoginName) + ';';  PRINT @sql; -- In lệnh SQL để kiểm tra  EXEC(@sql);  END  ELSE IF @Roleid = 2 -- GiangVien  BEGIN  -- Thu hồi sysadmin nếu trước đó là Admin  IF @RoleidCu = 1  BEGIN  SET @sql = 'ALTER SERVER ROLE [sysadmin] DROP MEMBER ' + QUOTENAME(@LoginName) + ';';  PRINT @sql; -- In lệnh SQL để kiểm tra  EXEC(@sql);  END  SET @sql = 'ALTER ROLE [GiangVienRole] ADD MEMBER ' + QUOTENAME(@LoginName) + ';';  PRINT @sql; -- In lệnh SQL để kiểm tra  EXEC(@sql);  END  END  COMMIT TRANSACTION;  END TRY  BEGIN CATCH  IF @@TRANCOUNT > 0  ROLLBACK TRANSACTION;  SELECT  ERROR\_NUMBER() AS ErrorNumber,  ERROR\_MESSAGE() AS ErrorMessage;  END CATCH  END  GO |

C#:

|  |
| --- |
| if (row.RowState == DataRowState.Modified)  {  int maTK = Convert.ToInt32(row["MaTK"]);  string tenDangNhap = row["TenDangNhap"]?.ToString() ?? "";  string matKhau = row["MatKhau"]?.ToString() ?? "";  string roleId = row["Roleid"]?.ToString() ?? "";  string maGV = row["MaGV"]?.ToString() ?? "";  string trangThai = row["TrangThai"]?.ToString() ?? "";  if (int.Parse(roleId) == 2 && string.IsNullOrWhiteSpace(maGV))  {  MessageBox.Show("Phải có mã giảng viên");  return;  }  string query = $@"EXEC sp\_CapNhatTaiKhoan  @MaTK = {maTK},  @TenDangNhap = {(string.IsNullOrWhiteSpace(tenDangNhap) ? "NULL" : $"N'{tenDangNhap}'")},  @MatKhau = {(string.IsNullOrWhiteSpace(matKhau) ? "NULL" : $"N'{matKhau}'")},  @Roleid = {(string.IsNullOrWhiteSpace(roleId) ? "NULL" : roleId)},  @MaGV = {(string.IsNullOrWhiteSpace(maGV) ? "NULL" : $"'{maGV}'")},  @TrangThai = {(string.IsNullOrWhiteSpace(trangThai) ? "NULL" : trangThai)}";  frmAdmin.executeQuery(query);  } |

## Thống kê

**Thống kê môn học theo học kỳ năm học**

|  |
| --- |
| CREATE PROCEDURE sp\_TrungBinhMonHoc  @MaHocKyNamHoc INT  AS  BEGIN  SET NOCOUNT ON;  SELECT  MH.MaMH,  MH.TenMH,  COUNT(\*) AS SoSV\_Tong,  ROUND(AVG(CTHP.DiemTB), 2) AS DiemTB,  SUM(CASE WHEN dbo.fn\_TrangThaiDiemTB(CTHP.DiemTB) = N'Đạt' THEN 1 ELSE 0 END) AS SoSV\_Dat,  SUM(CASE WHEN dbo.fn\_TrangThaiDiemTB(CTHP.DiemTB) = N'Không đạt' THEN 1 ELSE 0 END) AS SoSV\_Rot,  SUM(CASE WHEN CTHP.DiemTB IS NULL THEN 1 ELSE 0 END) AS SoSV\_Chuacham  FROM ChiTietHocPhan CTHP  INNER JOIN LopHocPhan LHP  ON CTHP.MaLHP = LHP.MaLHP  AND CTHP.MaHocKyNamHoc = LHP.MaHocKyNamHoc  INNER JOIN MonHoc MH  ON LHP.MaMH = MH.MaMH  WHERE CTHP.MaHocKyNamHoc = @MaHocKyNamHoc  GROUP BY MH.MaMH, MH.TenMH  ORDER BY DiemTB DESC;  END |

C#:

|  |
| --- |
| try  {  string query = $"EXEC sp\_TrungBinhMonHoc @MaHocKyNamHoc = {currentMaHocKyNamHoc}";  DataTable dt = frmAdmin.getData(query);  if (dt != null && dt.Rows.Count > 0)  {  gridControl1.DataSource = dt;  ConfigureGridView();  string queryHocKy = $"SELECT HocKy, NamHoc FROM HocKyNamHoc WHERE MaHocKyNamHoc = {currentMaHocKyNamHoc}";  DataTable dtHocKy = frmAdmin.getData(queryHocKy);  if (dtHocKy != null && dtHocKy.Rows.Count > 0)  {  DataRow rowHK = dtHocKy.Rows[0];  label2.Text = $"Học kỳ {rowHK["HocKy"]} - Năm học {rowHK["NamHoc"]}";  }    }  else  {  gridControl1.DataSource = null;  label2.Text = "Không có dữ liệu";  MessageBox.Show("Không có dữ liệu thống kê cho học kỳ này.", "Thông báo",  MessageBoxButtons.OK, MessageBoxIcon.Information);  }  }    catch (Exception ex)  {  MessageBox.Show("Lỗi khi tải dữ liệu thống kê: " + ex.Message, "Lỗi",  MessageBoxButtons.OK, MessageBoxIcon.Error);  gridControl1.DataSource = null;  label2.Text = "Lỗi tải dữ liệu";  } |

**Thống kê trạng thái môn học theo lớp học phần**

|  |
| --- |
| CREATE PROCEDURE dbo.sp\_ThongKeDiemTheoKhoangNho  @MaLHP VARCHAR(20),  @MaHocKyNamHoc INT  AS  BEGIN  SET NOCOUNT ON;  SELECT  COUNT(\*) AS TongSoSinhVien,  ISNULL(SUM(CASE WHEN DiemTB >= 0 AND DiemTB < 1 THEN 1 ELSE 0 END), 0) AS Khoang0\_1,  ISNULL(SUM(CASE WHEN DiemTB >= 1 AND DiemTB < 2 THEN 1 ELSE 0 END), 0) AS Khoang1\_2,  ISNULL(SUM(CASE WHEN DiemTB >= 2 AND DiemTB < 3 THEN 1 ELSE 0 END), 0) AS Khoang2\_3,  ISNULL(SUM(CASE WHEN DiemTB >= 3 AND DiemTB < 4 THEN 1 ELSE 0 END), 0) AS Khoang3\_4,  ISNULL(SUM(CASE WHEN DiemTB >= 4 AND DiemTB < 5 THEN 1 ELSE 0 END), 0) AS Khoang4\_5,  ISNULL(SUM(CASE WHEN DiemTB >= 5 AND DiemTB < 6 THEN 1 ELSE 0 END), 0) AS Khoang5\_6,  ISNULL(SUM(CASE WHEN DiemTB >= 6 AND DiemTB < 7 THEN 1 ELSE 0 END), 0) AS Khoang6\_7,  ISNULL(SUM(CASE WHEN DiemTB >= 7 AND DiemTB < 8 THEN 1 ELSE 0 END), 0) AS Khoang7\_8,  ISNULL(SUM(CASE WHEN DiemTB >= 8 AND DiemTB < 9 THEN 1 ELSE 0 END), 0) AS Khoang8\_9,  ISNULL(SUM(CASE WHEN DiemTB >= 9 AND DiemTB <= 10 THEN 1 ELSE 0 END), 0) AS Khoang9\_10  FROM fn\_SinhVienVaDiemTheoLopHocPhan(@MaLHP, @MaHocKyNamHoc)  WHERE DiemTB IS NOT NULL;  END  GO |

C#:

|  |
| --- |
| public void HienThiBieuDoThongKe(string maLHP, int maHocKyNamHoc)  {  if (string.IsNullOrEmpty(maLHP) || maHocKyNamHoc <= 0)  {  MessageBox.Show("Chưa có dữ liệu để hiển thị biểu đồ.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  return;  }  // Lấy dữ liệu thống kê từ stored procedure bằng getData  string query = $"EXEC sp\_ThongKeDiemLopHocPhan @MaLHP = '{maLHP}', @MaHocKyNamHoc = {maHocKyNamHoc}";  DataTable dtThongKe = frmGiangVien.getData(query);  if (dtThongKe != null && dtThongKe.Rows.Count > 0)  {  DataRow row = dtThongKe.Rows[0];  // Xóa dữ liệu cũ  chartThongKe.Series.Clear();  // Tạo series mới  Series series = new Series("Thống kê", ViewType.Pie);  series.Points.Add(new SeriesPoint("Đạt", Convert.ToInt32(row["SoSinhVienDat"])));  series.Points.Add(new SeriesPoint("Rớt", Convert.ToInt32(row["SoSinhVienRớt"])));  series.Points.Add(new SeriesPoint("Chưa chấm", Convert.ToInt32(row["SoSinhVienChuaCham"])));  // Đặt màu sắc cho từng phần  series.Points[0].Color = Color.FromArgb(76, 175, 80); // Xanh lá  series.Points[1].Color = Color.FromArgb(244, 67, 54); // Đỏ  series.Points[2].Color = Color.FromArgb(255, 193, 7); // Vàng  // Hiển thị nhãn và phần trăm  series.Label.TextPattern = "{A}: {V} ({VP:P1})";  series.Label.Font = new Font("Segoe UI", 12, FontStyle.Bold);  // Hiệu ứng 3D (nếu muốn)  chartThongKe.Legend.Visibility = DevExpress.Utils.DefaultBoolean.True;  chartThongKe.Series.Add(series);  chartThongKe.Titles.Clear();  chartThongKe.Titles.Add(new ChartTitle() { Text = "Thống kê kết quả lớp học phần", Font = new Font("Segoe UI", 14, FontStyle.Bold) });  chartThongKe.Dock = DockStyle.Fill;  chartThongKe.BackColor = Color.White;  chartThongKe.BorderOptions.Visibility = DevExpress.Utils.DefaultBoolean.False;  }  else  {  MessageBox.Show("Không có dữ liệu thống kê.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  } |

**Thống kê phổ điểm môn học theo lớp học phần**

|  |
| --- |
| CREATE PROCEDURE dbo.sp\_ThongKeDiemLopHocPhan  @MaLHP VARCHAR(20),  @MaHocKyNamHoc INT  AS  BEGIN  SET NOCOUNT ON;  SELECT  COUNT(\*) AS TongSoSinhVien,  ISNULL(SUM(CASE WHEN TrangThai = N'Đạt' THEN 1 ELSE 0 END), 0) AS SoSinhVienDat,  ISNULL(SUM(CASE WHEN TrangThai = N'Không đạt' THEN 1 ELSE 0 END), 0) AS SoSinhVienRớt,  ISNULL(SUM(CASE WHEN TrangThai IS NULL THEN 1 ELSE 0 END), 0) AS SoSinhVienChuaCham  FROM fn\_SinhVienVaDiemTheoLopHocPhan(@MaLHP, @MaHocKyNamHoc);  END  GO |

C#:

|  |
| --- |
| public void HienThiBieuDoPhoDiem(string maLHP, int maHocKyNamHoc)  {  // Kiểm tra tham số đầu vào  if (string.IsNullOrEmpty(maLHP) || maHocKyNamHoc <= 0)  {  MessageBox.Show("Chưa có dữ liệu để hiển thị biểu đồ.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  return;  }  // Lấy dữ liệu phổ điểm từ stored procedure bằng getData  string query = $"EXEC sp\_ThongKeDiemTheoKhoangNho @MaLHP = '{maLHP}', @MaHocKyNamHoc = {maHocKyNamHoc}";  DataTable dtPhoDiem = frmGiangVien.getData(query);  if (dtPhoDiem != null && dtPhoDiem.Rows.Count > 0)  {  DataRow row = dtPhoDiem.Rows[0];    // Xóa dữ liệu cũ  chartThongKe.Series.Clear();  // Tạo series mới cho biểu đồ cột  Series series = new Series("Phổ điểm", ViewType.Bar);  // Thêm các điểm dữ liệu cho từng khoảng điểm  series.Points.Add(new SeriesPoint("0-1", Convert.ToInt32(row["Khoang0\_1"])));  series.Points.Add(new SeriesPoint("1-2", Convert.ToInt32(row["Khoang1\_2"])));  series.Points.Add(new SeriesPoint("2-3", Convert.ToInt32(row["Khoang2\_3"])));  series.Points.Add(new SeriesPoint("3-4", Convert.ToInt32(row["Khoang3\_4"])));  series.Points.Add(new SeriesPoint("4-5", Convert.ToInt32(row["Khoang4\_5"])));  series.Points.Add(new SeriesPoint("5-6", Convert.ToInt32(row["Khoang5\_6"])));  series.Points.Add(new SeriesPoint("6-7", Convert.ToInt32(row["Khoang6\_7"])));  series.Points.Add(new SeriesPoint("7-8", Convert.ToInt32(row["Khoang7\_8"])));  series.Points.Add(new SeriesPoint("8-9", Convert.ToInt32(row["Khoang8\_9"])));  series.Points.Add(new SeriesPoint("9-10", Convert.ToInt32(row["Khoang9\_10"])));  // Đặt màu sắc gradient cho các cột  Color[] colors = {  Color.FromArgb(244, 67, 54), // Đỏ đậm  Color.FromArgb(255, 87, 34), // Đỏ cam  Color.FromArgb(255, 152, 0), // Cam  Color.FromArgb(255, 193, 7), // Vàng  Color.FromArgb(205, 220, 57), // Vàng xanh  Color.FromArgb(139, 195, 74), // Xanh lá nhạt  Color.FromArgb(76, 175, 80), // Xanh lá  Color.FromArgb(0, 150, 136), // Xanh lá đậm  Color.FromArgb(33, 150, 243), // Xanh dương  Color.FromArgb(63, 81, 181) // Xanh tím  };  for (int i = 0; i < series.Points.Count && i < colors.Length; i++)  {  series.Points[i].Color = colors[i];  }  // Cấu hình hiển thị  series.Label.TextPattern = "{V}";  series.Label.Font = new Font("Segoe UI", 10, FontStyle.Bold);  series.LabelsVisibility = DevExpress.Utils.DefaultBoolean.True;  // Thêm series vào chart  chartThongKe.Series.Add(series);  // Cấu hình chart  chartThongKe.Legend.Visibility = DevExpress.Utils.DefaultBoolean.False;  chartThongKe.Titles.Clear();  chartThongKe.Titles.Add(new ChartTitle()  {  Text = "Phổ điểm lớp học phần",  Font = new Font("Segoe UI", 14, FontStyle.Bold)  });  // Cấu hình trục X và Y  if (chartThongKe.Diagram is XYDiagram diagram)  {  diagram.AxisX.Title.Text = "Khoảng điểm";  diagram.AxisX.Title.Font = new Font("Segoe UI", 12, FontStyle.Bold);  diagram.AxisY.Title.Text = "Số sinh viên";  diagram.AxisY.Title.Font = new Font("Segoe UI", 12, FontStyle.Bold);  diagram.AxisY.WholeRange.SetMinMaxValues(0, Convert.ToInt32(row["TongSoSinhVien"]) + 1);  }  chartThongKe.Dock = DockStyle.Fill;  chartThongKe.BackColor = Color.White;  chartThongKe.BorderOptions.Visibility = DevExpress.Utils.DefaultBoolean.False;  }  else  {  MessageBox.Show("Không có dữ liệu phổ điểm.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  } |

**Tính điểm trung bình và số tín chỉ đạt**

|  |
| --- |
| CREATE PROCEDURE dbo.sp\_TinhTBVaTinChiDat  @MaSV VARCHAR(10)  AS  BEGIN  SET NOCOUNT ON;  SELECT  AVG(CTHP.DiemTB) AS DiemTB\_He10,  AVG(dbo.fn\_QuyDoiDiemHe4(CTHP.DiemTB)) AS DiemTB\_He4,  SUM(MH.SoTinChi) AS TinChiDat  FROM ChiTietHocPhan CTHP  INNER JOIN LopHocPhan LHP  ON CTHP.MaLHP = LHP.MaLHP  AND CTHP.MaHocKyNamHoc = LHP.MaHocKyNamHoc  INNER JOIN MonHoc MH  ON LHP.MaMH = MH.MaMH  WHERE CTHP.MaSV = @MaSV  AND dbo.fn\_TrangThaiDiemTB(CTHP.DiemTB) = N'Đạt';  END  GO |

C#:

|  |
| --- |
| private void TinhDiemTBVaTinChiDat(string maSV)  {  try  {  string query = $"EXEC sp\_TinhTBVaTinChiDat @MaSV = '{maSV}'";  DataTable dtResult = frmAdmin.getData(query);  if (dtResult != null && dtResult.Rows.Count > 0)  {  DataRow row = dtResult.Rows[0];  double diemTB\_He10 = row["DiemTB\_He10"] != DBNull.Value ?  Convert.ToDouble(row["DiemTB\_He10"]) : 0;  lblDiemhe10.Text = $"{diemTB\_He10:F2}";  double diemTB\_He4 = row["DiemTB\_He4"] != DBNull.Value ?  Convert.ToDouble(row["DiemTB\_He4"]) : 0;  lblDiemhe4.Text = $"{diemTB\_He4:F2}";  int tinChiDat = row["TinChiDat"] != DBNull.Value ?  Convert.ToInt32(row["TinChiDat"]) : 0;  lblSoTin.Text = $"{tinChiDat}";  }  else  {  lblDiemhe4.Text = "0.00";  lblDiemhe10.Text = "0.00";  lblSoTin.Text = "0";  }  }  catch (Exception ex)  {  MessageBox.Show("Lỗi khi tính điểm TB và tín chỉ: " + ex.Message,  "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  lblDiemhe4.Text = "0.00";  lblDiemhe10.Text = "0.00";  lblSoTin.Text = "0";  }  } |

# CHƯƠNG 4: TẠO USER VÀ PHÂN QUYỀN

Hệ thống có 2 nhóm người dùng là: Thủ thư và Quản trị viên (Admin)

## 1. Tạo Role và phân quyền

### 1.1. Giảng viên

|  |
| --- |
| IF NOT EXISTS (SELECT \* FROM sys.database\_principals WHERE name = 'GiangVienRole')  CREATE ROLE GiangVienRole;  GO  -- Quyền SELECT trên các bảng cần thiết  GRANT SELECT ON dbo.SinhVien TO GiangVienRole;  GRANT SELECT ON dbo.GiangVien TO GiangVienRole;  GRANT SELECT ON dbo.MonHoc TO GiangVienRole;  GRANT SELECT ON dbo.LopHocPhan TO GiangVienRole;  GRANT SELECT ON dbo.HocKyNamHoc TO GiangVienRole;  -- Quyền SELECT, UPDATE trên ChiTietHocPhan  GRANT SELECT, UPDATE ON dbo.ChiTietHocPhan TO GiangVienRole;  -- Quyền EXECUTE trên các stored procedure cần thiết  GRANT EXECUTE ON dbo.sp\_GetMaHocKyMoiNhat TO GiangVienRole;  GRANT EXECUTE ON dbo.sp\_DanhSachHocKyNamHoc TO GiangVienRole;  GRANT EXECUTE ON dbo.sp\_CapNhatDiemHocPhan TO GiangVienRole;  GRANT EXECUTE ON dbo.sp\_ThongKeDiemLopHocPhan TO GiangVienRole;  GRANT EXECUTE ON dbo.sp\_ThongKeDiemTheoKhoangNho TO GiangVienRole;  GRANT EXECUTE ON dbo.sp\_GetLopHocPhanByGV TO GiangVienRole;  GRANT EXECUTE ON dbo.sp\_LayLopHocPhanKhac TO GiangVienRole;  GRANT EXECUTE ON dbo.sp\_ChuyenLopHocPhan TO GiangVienRole;  GRANT EXECUTE ON dbo.sp\_CapNhatGiangVien TO GiangVienRole;  GRANT EXECUTE ON dbo.sp\_XoaDangKyMonHoc TO GiangVienRole;  -- Quyền SELECT trên view  GRANT SELECT ON dbo.vw\_ThongTinChiTietSV TO GiangVienRole;  -- Quyền SELECT trên Table-Valued Function  GRANT SELECT ON dbo.fn\_SinhVienVaDiemTheoLopHocPhan TO GiangVienRole;  GRANT SELECT ON dbo.fn\_GetThongTinGV TO GiangVienRole;  GRANT SELECT ON dbo.fn\_SinhVienTheoLopHocPhan TO GiangVienRole;  GRANT SELECT ON dbo.fn\_ChiTietDiemSV TO GiangVienRole;  GRANT SELECT ON dbo.fn\_DanhSachMonHoc\_GiangVien TO GiangVienRole;  GO |

### 1.2. Quản trị viên (Admin)

|  |
| --- |
| -- =====================================================  -- PHÂN QUYỀN CHO ADMINROLE - FULL QUYỀN TRÊN TOÀN BỘ OBJECTS  -- =====================================================  -- Admin có full quyền trên toàn bộ database (bao gồm DML, DDL, EXECUTE trên tất cả tables, views, procs, functions, types, triggers)  IF NOT EXISTS (SELECT \* FROM sys.database\_principals WHERE name = 'AdminRole')  CREATE ROLE AdminRole;  GRANT SELECT, INSERT, UPDATE, DELETE, EXECUTE ON DATABASE::QL\_SinhVien TO [AdminRole];  GRANT CONTROL ON SCHEMA::dbo TO [AdminRole]; |

# CHƯƠNG 5: THIẾT KẾ GIAO DIỆN HỆ THỐNG

## Kiến trúc hệ thống

Môi trường cài đặt và công nghệ sử dụng

- Hệ cơ sở dữ liệu Microsoft SQL Server Management Studio 19.

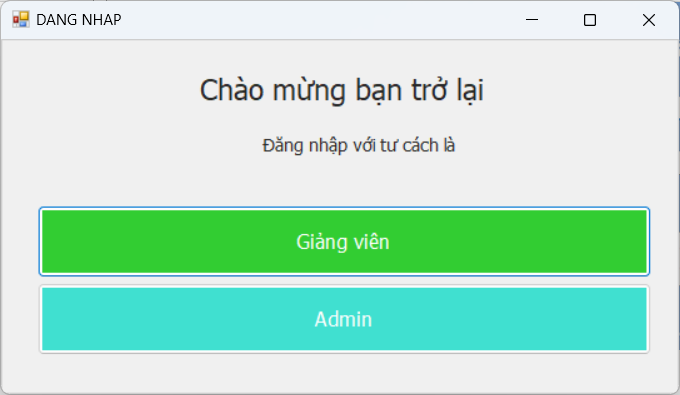
- Chương trình Winform được viết bằng (.NET Framework) C#.

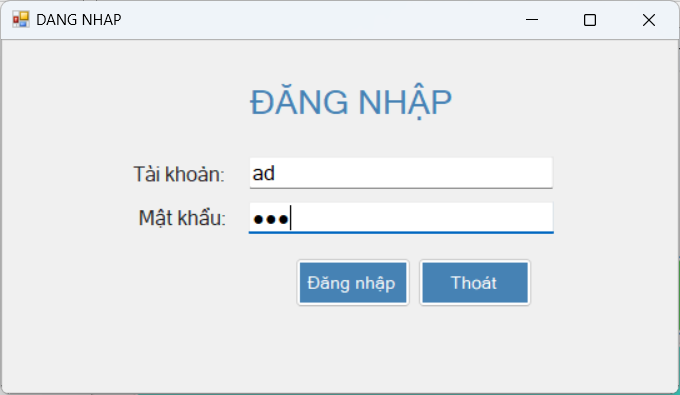
- Công cụ giao diện người dùng: DevExpress 22

- Công cụ lập trình: Microsoft Visual Studio 2022

## 2. Giao diện của chương trình:

### 2.1. Form đăng nhập



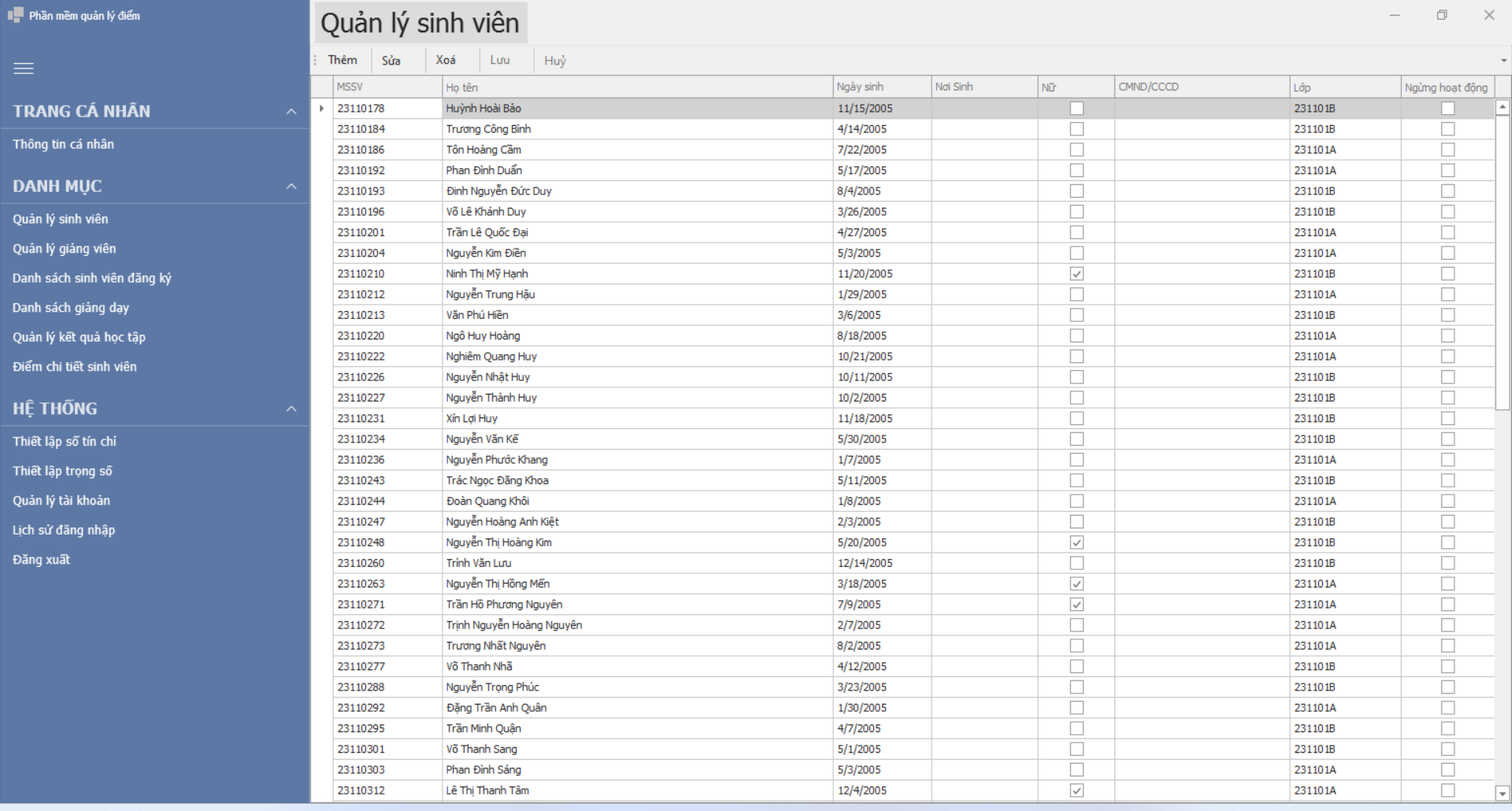


### 2.2. Form admin

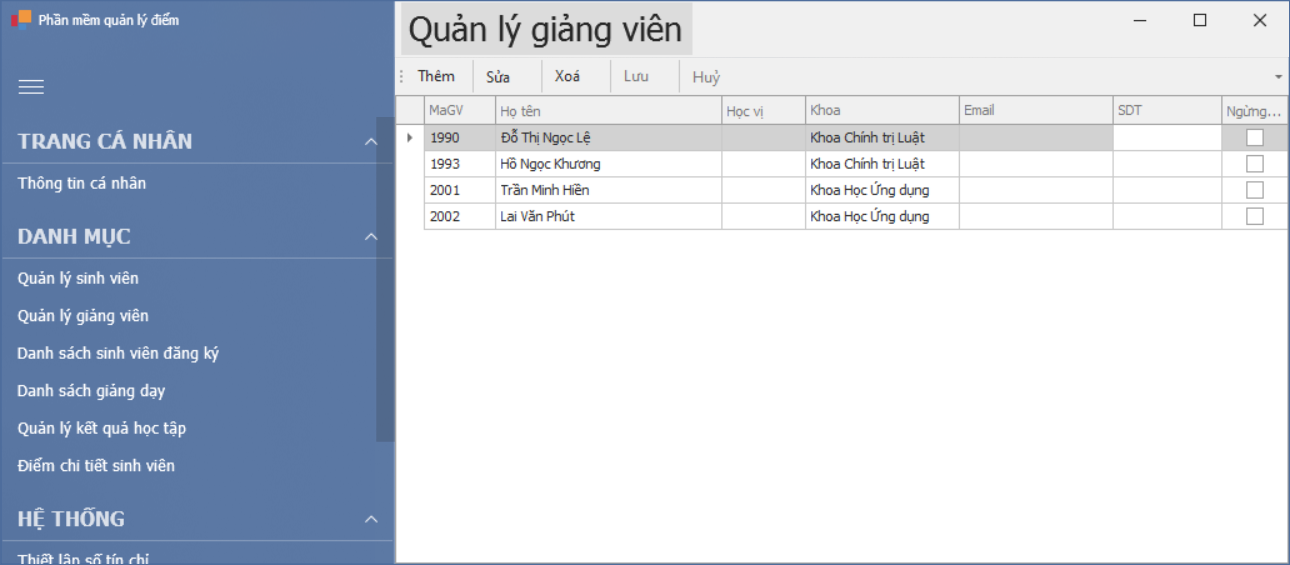
**Thông tin cá nhân**



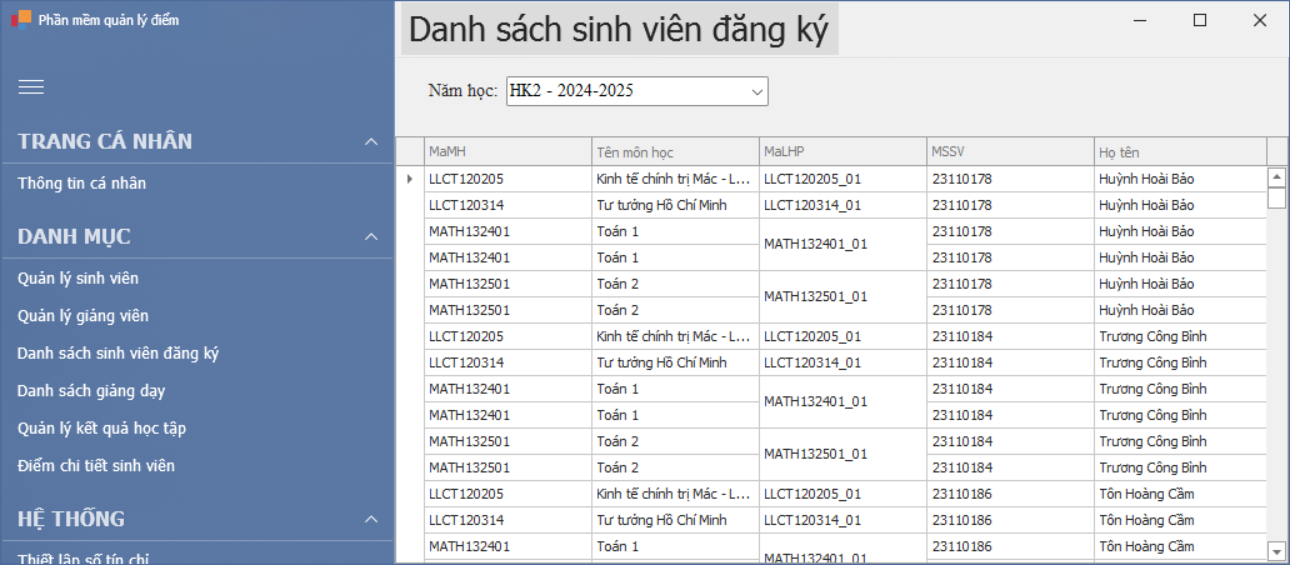
**Quản lý sinh viên**



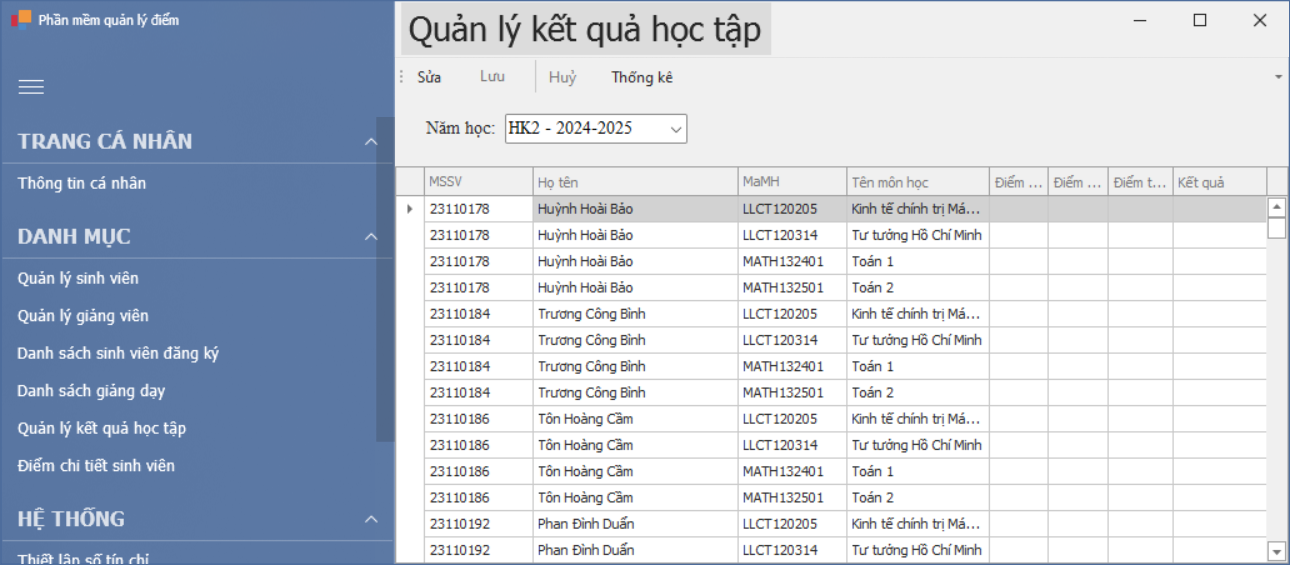
**Quản lý giảng viên**

****

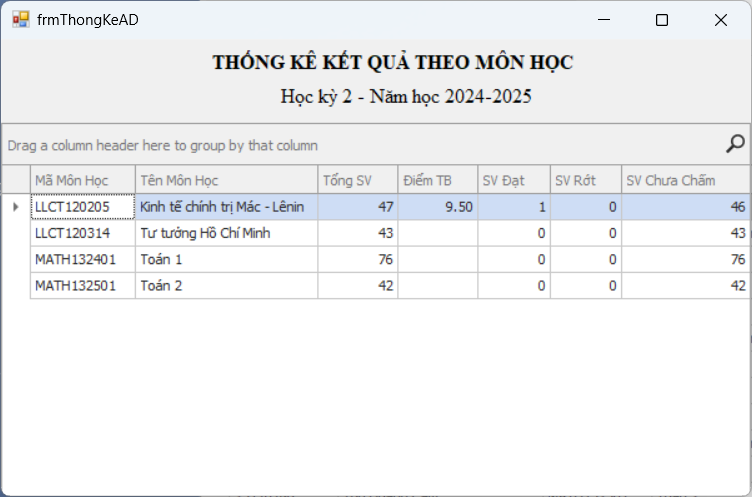
**Danh sách sinh viên đăng ký**

****

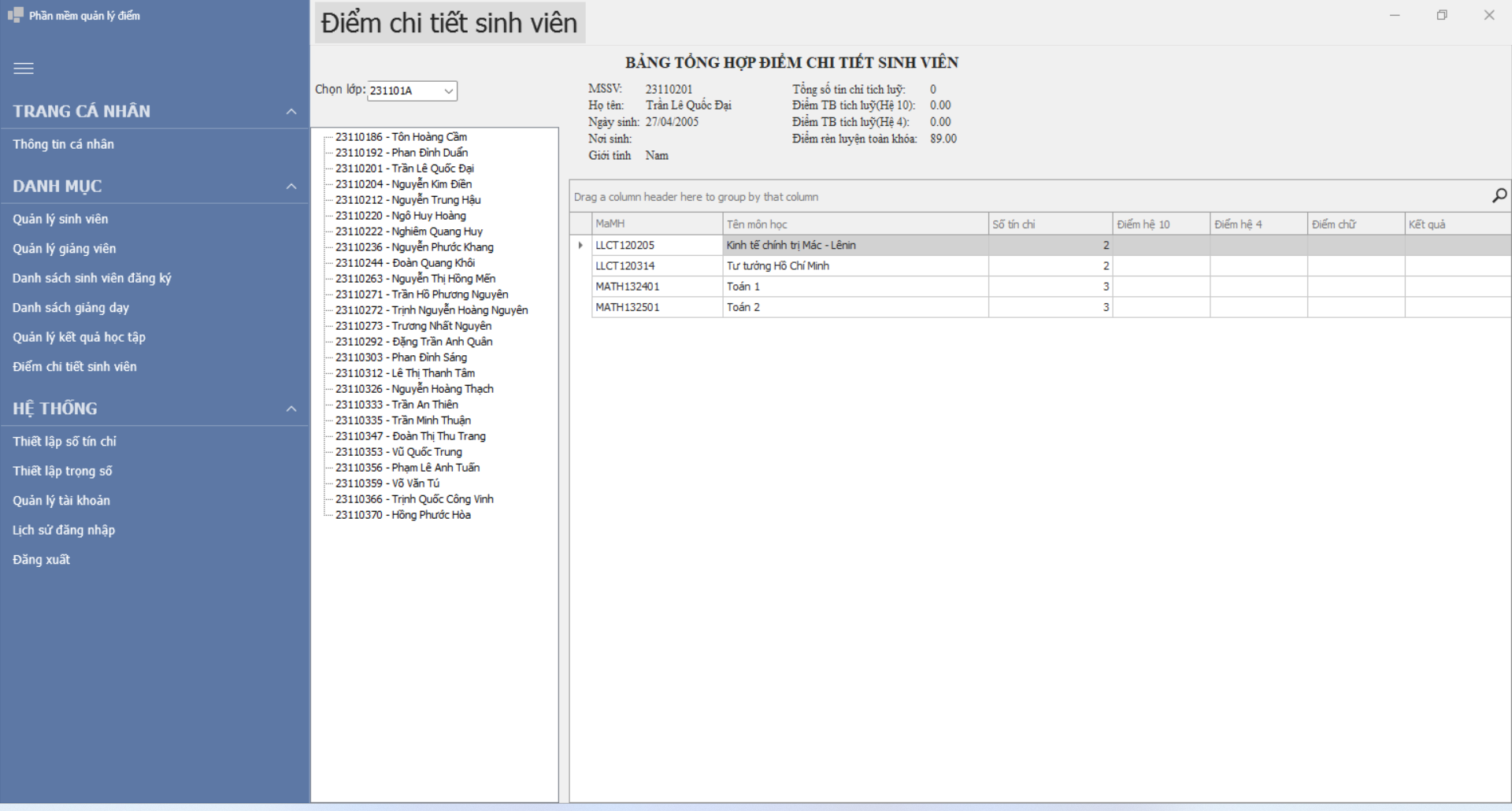
**Quản lý kết quả học tập**

****

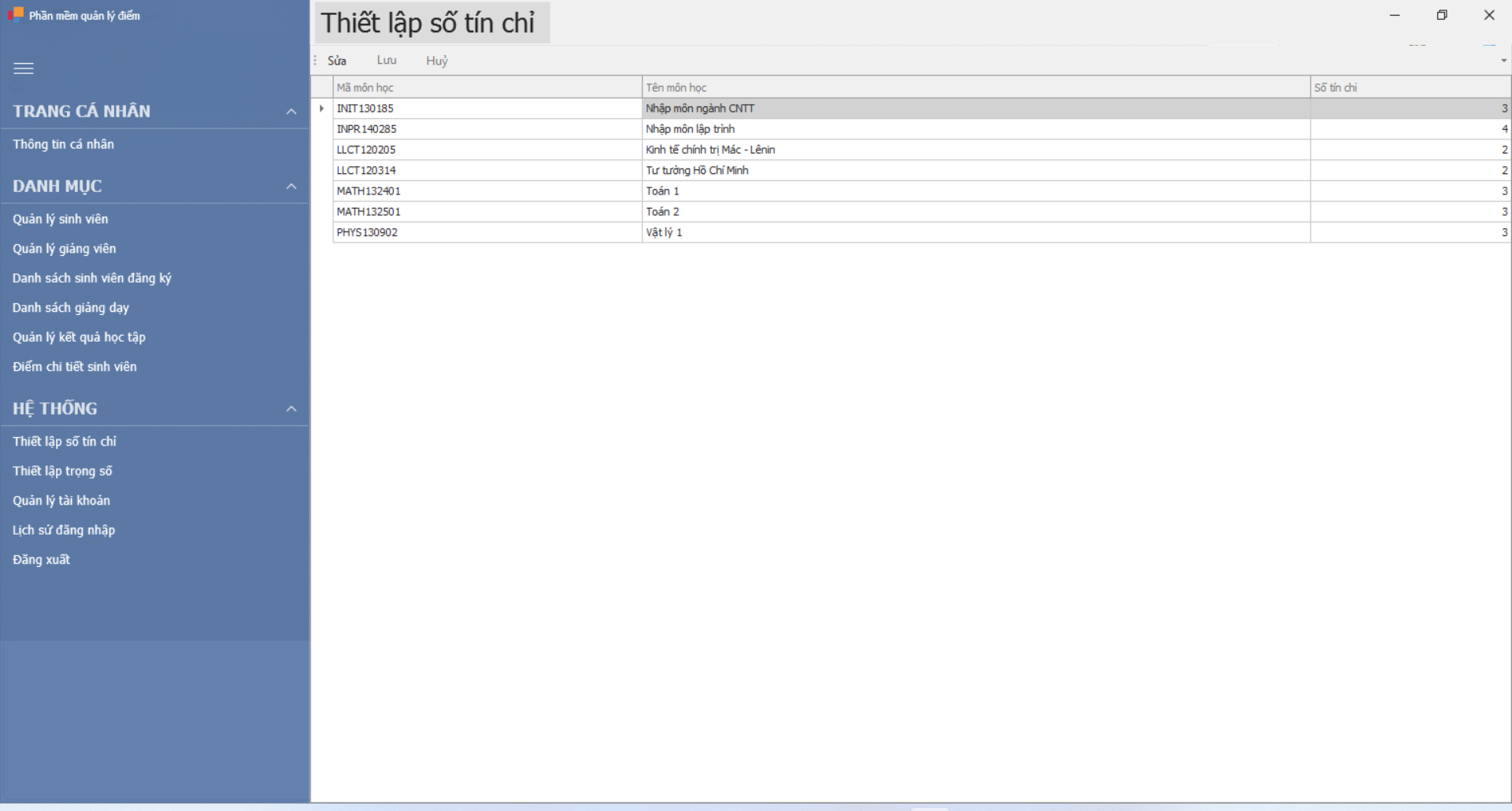
**Thống kê môn theo năm học**

****

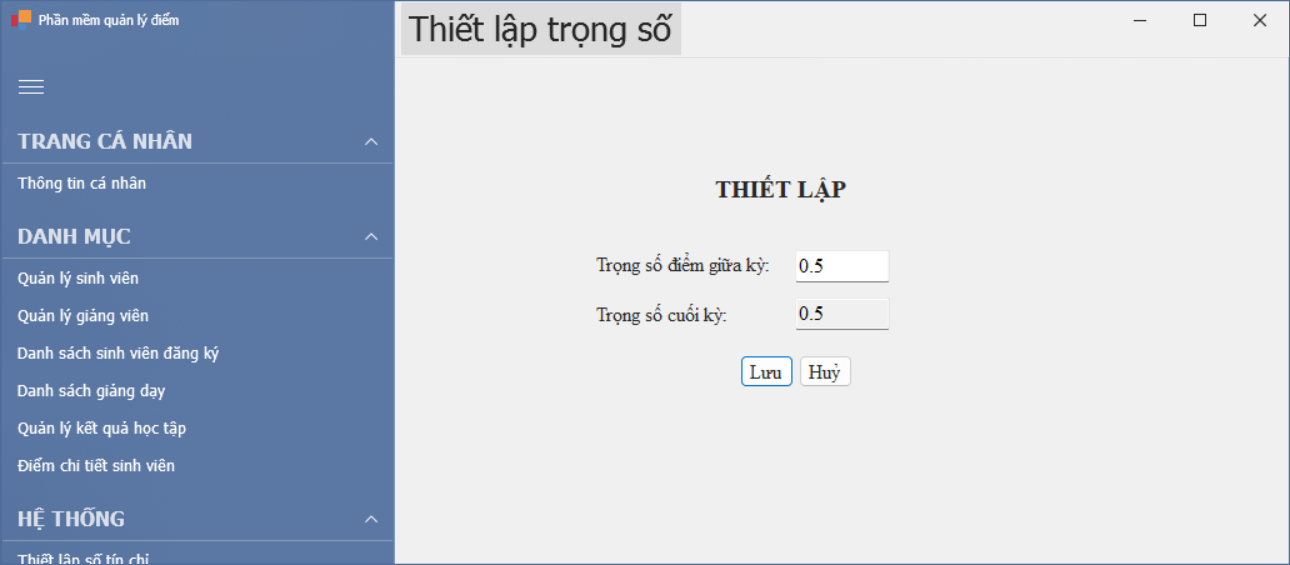
**Điểm chi tiết sinh viên theo lớp**

****

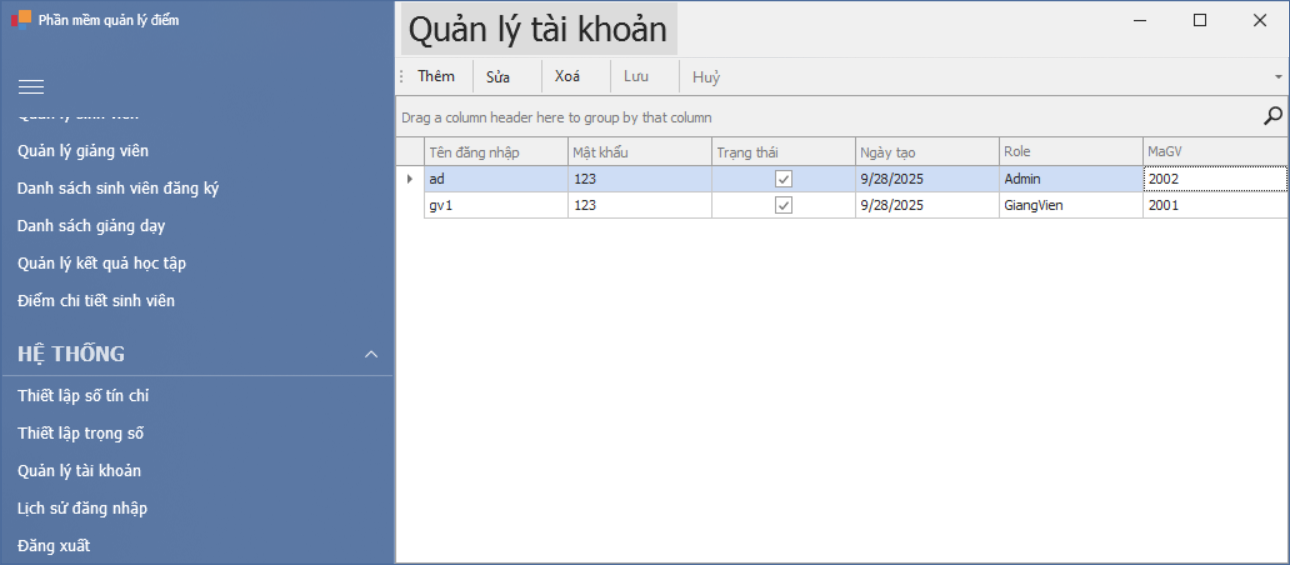
**Thiết lập số tín chỉ**

****

**Thiết lập trọng số**

****

**Quản lý tài khoản**

****

**Lịch sử đăng nhập**

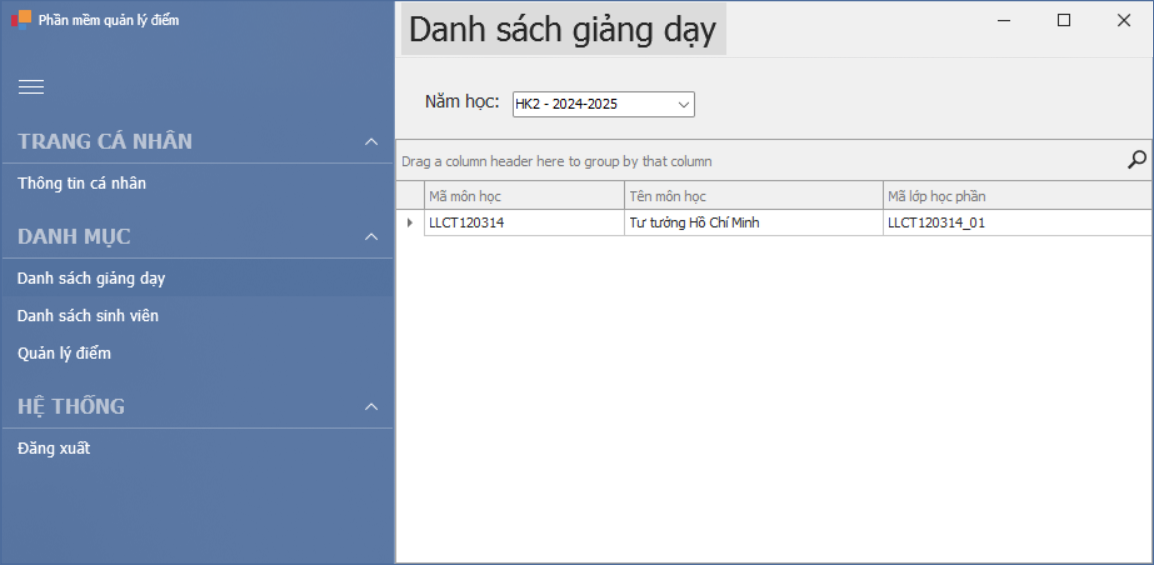
****

### 2.3 Form giảng viên

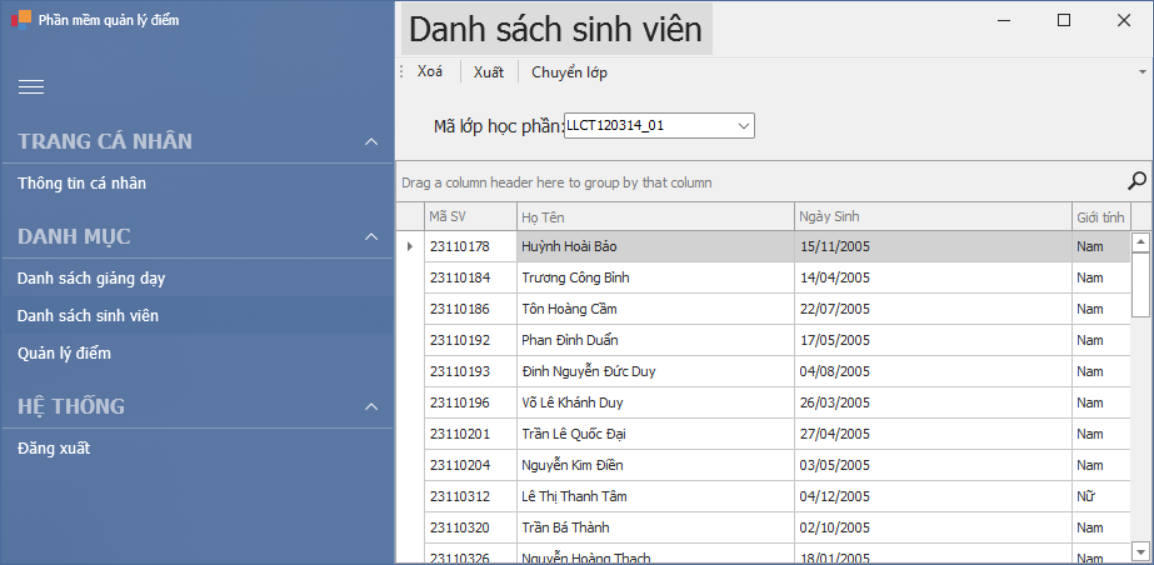
**Thông tin cá nhân**

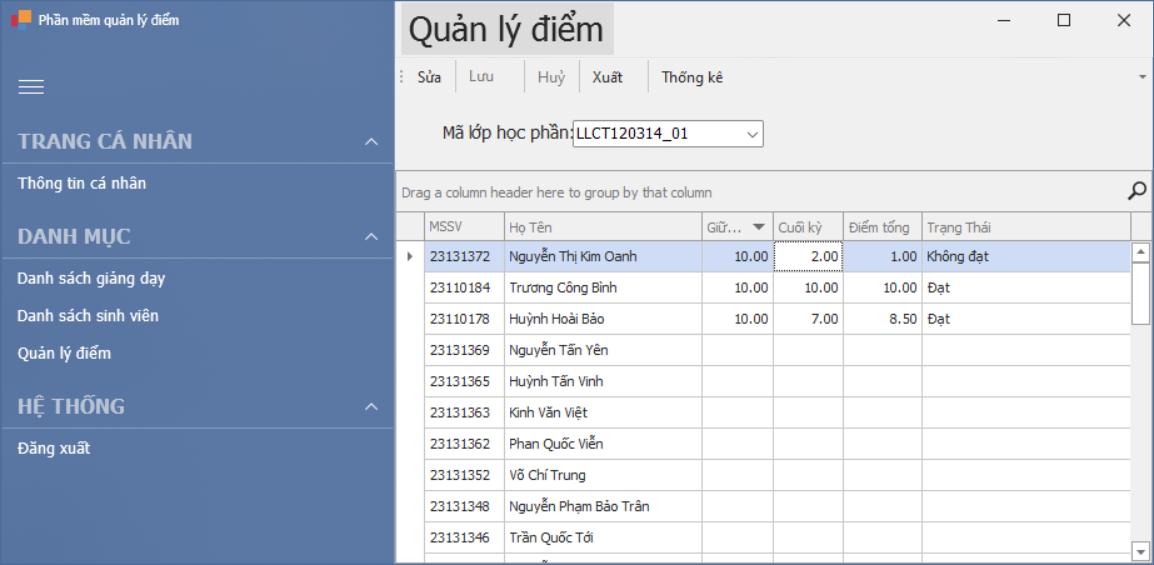
****

**Danh sách giảng dạy theo năm học**

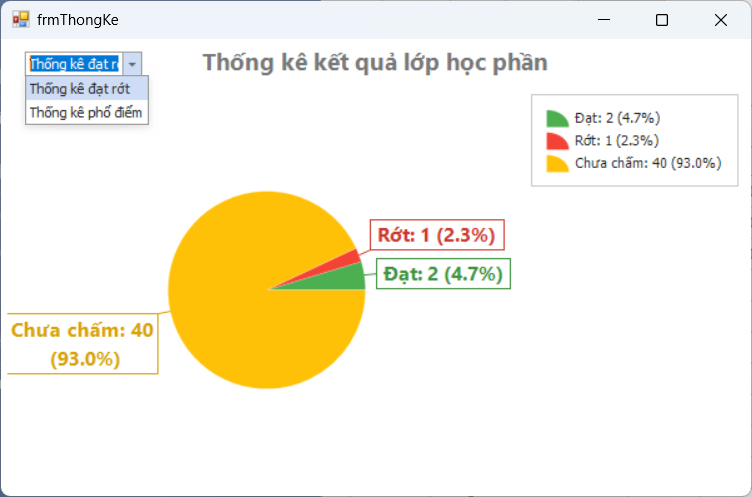
****

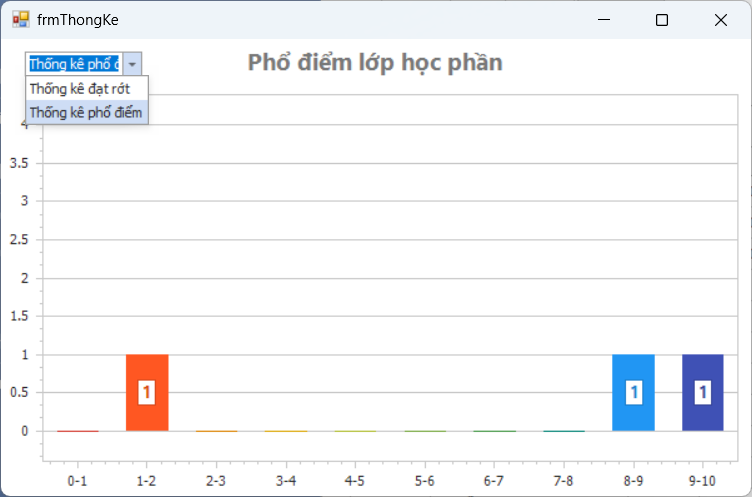
**Danh sách sinh viên theo lớp học phần**

**Quản lý điểm theo lớp học phần**

****

**Thống kê kết quả lớp học phần**

****

****

# CHƯƠNG 6: KẾT LUẬN

Dự án “Hệ thống Quản lý Điểm” giúp em hiểu rõ hơn về cách xây dựng một ứng dụng quản lý dữ liệu thực tế, từ khâu thiết kế cơ sở dữ liệu đến lập trình và triển khai chức năng. Việc kết hợp C# và SQL Server mang lại quy trình làm việc rõ ràng và hiệu quả, đặc biệt trong các thao tác như nhập điểm, chỉnh sửa điểm và thống kê kết quả học tập.

Trong quá trình thực hiện, em đã vận dụng các kỹ thuật như Stored Procedure, Function, Trigger và View để tối ưu hiệu năng và đảm bảo tính chính xác của dữ liệu. Đồng thời, em cũng nắm rõ hơn về quy trình quản lý đào tạo trong nhà trường, từ phân công lớp học phần đến xử lý điểm số theo từng học kỳ.

Dù hệ thống còn có thể mở rộng thêm, nhưng phiên bản hiện tại đã đáp ứng được các chức năng cốt lõi của một công cụ hỗ trợ giáo vụ và giảng viên.

Dự án này là bước thực hành quan trọng giúp em tự tin hơn khi làm việc với hệ quản trị cơ sở dữ liệu và phần mềm quản lý trong các dự án sau này.