

**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 THƯ VIỆN**

**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 thư viện

|  |  |
| --- | --- |
| **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 THƯ VIỆN 4](#_Toc209600440)

[**1. Đặc tả đề tài 4**](#_Toc209600441)

[**1.1. Mô tả bài toán 4**](#_Toc209600442)

[**1.2 Mô tả dữ liệu 4**](#_Toc209600443)

[**1.3 Mô tả chức năng 5**](#_Toc209600444)

[**2. Giao diện hệ thống 6**](#_Toc209600445)

[**2.1. Giao diện đối với thủ thư 6**](#_Toc209600446)

[**2.2. Giao diện đối với quản trị viên 8**](#_Toc209600447)

[**3. Tính phí phạt theo từng loại 8**](#_Toc209600448)

[CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 9](#_Toc209600449)

[**1. Thiết kế cơ sở dữ liệu mức quan niệm 9**](#_Toc209600450)

[**2. Thiết kế cơ sở dữ liệu mức logic 11**](#_Toc209600451)

[**3. Các ràng buộc 12**](#_Toc209600452)

[**4. Cài đặt CSDL và các ràng buộc 14**](#_Toc209600453)

[**5. Lược đồ quan hệ sau khi cài đặt SQL 18**](#_Toc209600454)

[**6. Các triggers 19**](#_Toc209600455)

[**7. Các views 22**](#_Toc209600456)

[**8. Các hàm 24**](#_Toc209600457)

[CHƯƠNG 3: THIẾT KẾ CÁC CHỨC NĂNG 26](#_Toc209600458)

[**1. Kết nối cơ sở dữ liệu 26**](#_Toc209600459)

[**2. Danh mục Trang chủ 32**](#_Toc209600460)

[**3. Danh mục Sách 35**](#_Toc209600461)

[4. Danh mục Thành viên 46](#_Toc209600462)

[**5. Danh mục Mượn/Trả 49**](#_Toc209600463)

[**6. Danh mục Phạt – Lịch sử thanh toán 54**](#_Toc209600464)

[**7. Danh mục Thống kê - Báo cáo 55**](#_Toc209600465)

[**8. Danh mục Người dùng 60**](#_Toc209600466)

[**9. Danh mục Cấu hình hệ thống – Nhật ký hệ thống 63**](#_Toc209600467)

[CHƯƠNG 4: TẠO USER VÀ PHÂN QUYỀN 66](#_Toc209600468)

[**1. Tạo Role và phân quyền 66**](#_Toc209600469)

[**1.1. Thủ thư (Librarian) 66**](#_Toc209600470)

[**1.2. Quản trị viên (Admin) 68**](#_Toc209600471)

[**2. Đồng bộ giữa CRUD Người dùng (Users) và UserLogin của database 68**](#_Toc209600472)

[CHƯƠNG 5: THIẾT KẾ GIAO DIỆN HỆ THỐNG 72](#_Toc209600473)

[1. Kiến trúc hệ thống 72](#_Toc209600474)

[**1.1 Cấu trúc thư mục hệ thống 72**](#_Toc209600475)

[**1.2. Mô hình 3 lớp (3 layers) 74**](#_Toc209600476)

[**1.2.1. Lớp giao diện UI (User Interface) 74**](#_Toc209600477)

[**1.2.2. Lớp nghiệp vụ BLL (Business Logic Layer) 74**](#_Toc209600478)

[**1.2.3. Lớp truy cập dữ liệu DAL (Data Acess Láyer) 74**](#_Toc209600479)

[2. Giao diện của chương trình 75](#_Toc209600480)

[**2.1. Form đăng nhập 75**](#_Toc209600481)

[**2.2. Form chính – Danh mục Trang chủ 75**](#_Toc209600482)

[**2.3. Danh mục Quản lý Sách 76**](#_Toc209600483)

[**2.3.1 Tab Sách 76**](#_Toc209600484)

[**2.3.2 Tab Bản sao sách 77**](#_Toc209600485)

[**2.3.3 Tab Tác giả 77**](#_Toc209600486)

[**2.3.4 Tab Nhà xuất bản 78**](#_Toc209600487)

[**2.3.5 Tab Thể loại 78**](#_Toc209600488)

[**2.5 Danh mục Quản lý Thành viên 79**](#_Toc209600489)

[**2.6 Danh mục Quản lý Mượn/Trả 79**](#_Toc209600490)

[**2.6.1 Tab Danh sách mượn 79**](#_Toc209600491)

[**2.6.2 Tab Chi tiết 80**](#_Toc209600492)

[**2.7 Danh mục Quản lý Phạt – Lịch sử thanh toán 80**](#_Toc209600493)

[**2.7.1 Tab Quản lý Phạt 80**](#_Toc209600494)

[**2.7.2 Tab Lịch sử thanh toán 81**](#_Toc209600495)

[**2.8 Danh mục Thống kê – Báo cáo 81**](#_Toc209600496)

[**2.8.1 Top sách được mượn 81**](#_Toc209600497)

[**2.8.2 Tổng tiền phạt theo tuần/tháng/quý/năm 82**](#_Toc209600498)

[**2.8.3 Thống kê tình trạng sách 82**](#_Toc209600499)

[**2.8.4 Thống kê sách mượn quá hạn 83**](#_Toc209600500)

[**2.8.5 Thống kê sách mượn theo loại thành viên (Sinh viên/Giảng viên) 83**](#_Toc209600501)

[**2.9 Danh mục Quản lý người dùng 84**](#_Toc209600502)

[**2.10 Danh mục Cấu hình hệ thống 84**](#_Toc209600503)

[CHƯƠNG 6: KẾT LUẬN 85](#_Toc209600504)

# CHƯƠNG 1: ĐẶC TẢ HỆ THỐNG QUẢN LÝ THƯ VIỆN

## 1. Đặc tả đề tài

### 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).

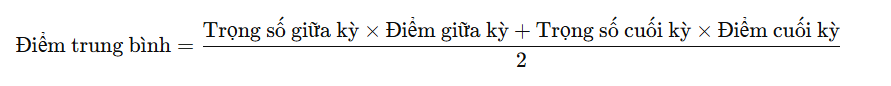
*Danh mục thiết lập trọng số:* thiết lập trọng số phần trăm điểm giữa kỳ, cuối kỳ.

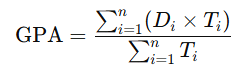
*Danh mục quản lý người dùng:* quản lý tài khoản có thể thêm xoá sửa tài khoản người dùng. Cập nhập trạng thái khoá tài khoản.

*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:

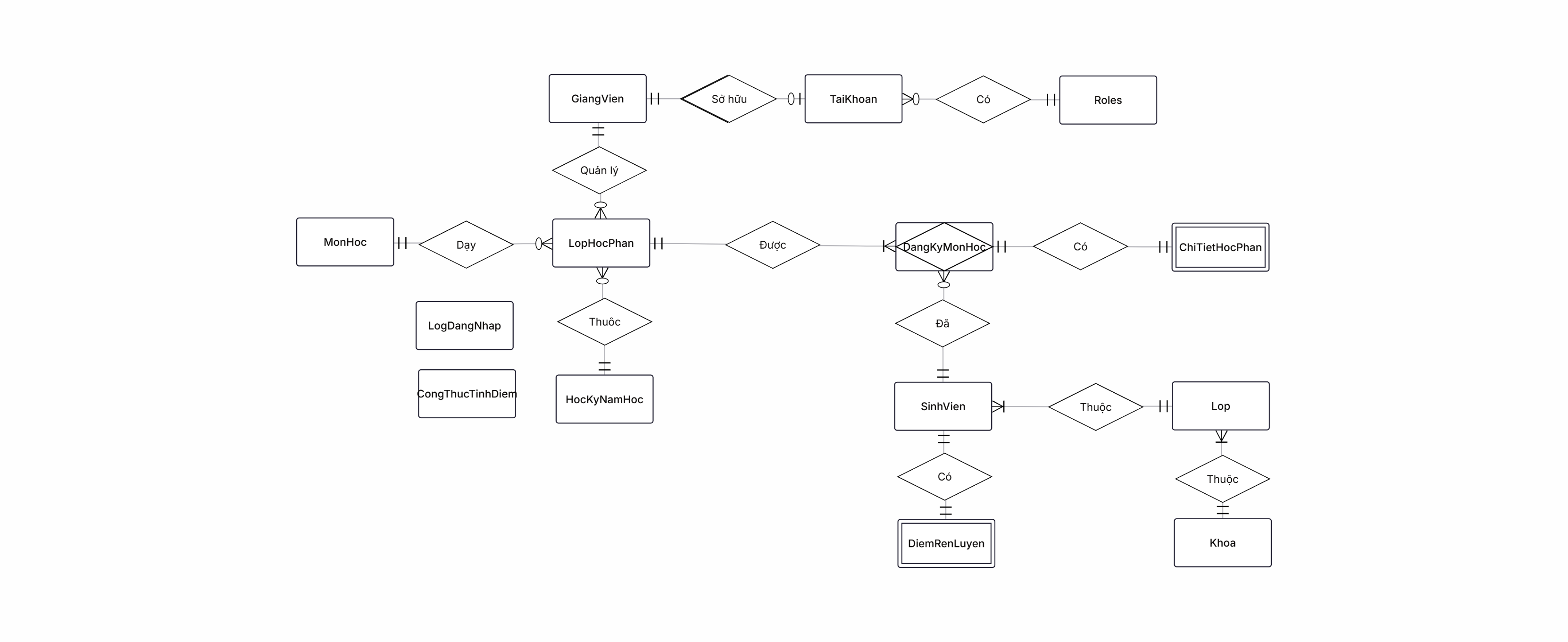
* Di​: điểm của học phần thứ ii (tính theo hệ 10 hoặc hệ 4).
* Ti: số tín chỉ của học phần thứ ii.
* n: tổng số học phần đã tích lũy (Môn học đạt).

Quy đổi điểm

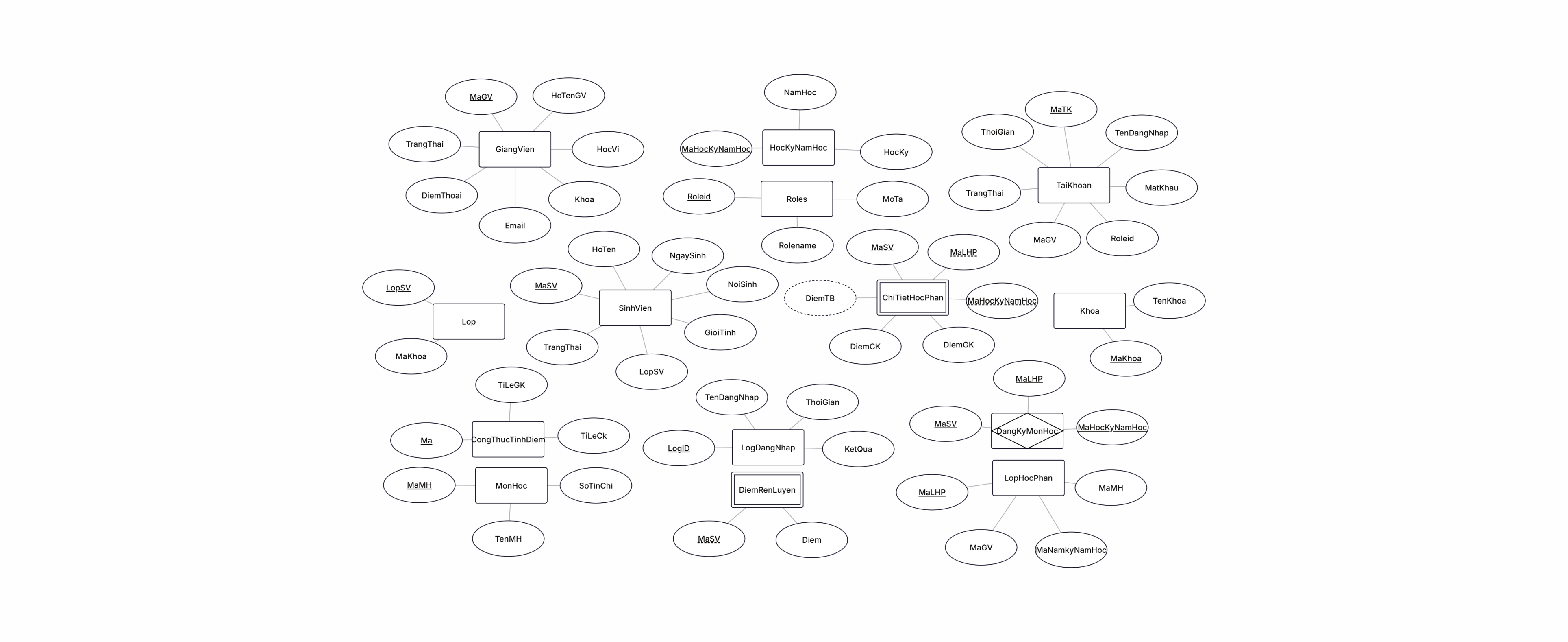
|  |  |  |
| --- | --- | --- |
| **Hệ 10 (thang 10)** | **Hệ 4 (thang 4)** | **Hệ chữ (Letter grade)** |
| Từ 9.0 – 10 | 4.0 | A+ |
| Từ 8.5 – 9.0 | 3.7 | A |
| Từ 8.0 – 8.5 | 3.5 | B+ |
| Từ 7.0 – 8.0 | 3.0 | B |
| Từ 6.5 – 7.0 | 2.5 | C+ |
| Từ 5.5 – 6.5 | 2.0 | C |
| Từ 5.0 – 5.5 | 1.5 | D+ |
| Từ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

GiangVien(MaGV, HoTenGV, HocVi, Khoa, 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) |
| **6** | **GiangVien** | [MaGV] [VARCHAR(10)] PRIMARY KEY,  [HoTenGV] [NVARCHAR(100)] NOT NULL,  [HocVi] [NVARCHAR(50)],  [Khoa] [NVARCHAR(100)],  [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)] NOT NULL 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) |
| **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)  ); |

***Bảng Giảng viên***

|  |
| --- |
| CREATE TABLE GiangVien (  MaGV VARCHAR(10) PRIMARY KEY,  HoTenGV NVARCHAR(100) NOT NULL,  HocVi NVARCHAR(50),  Khoa NVARCHAR(100),  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)  ); |

***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\_KhoaTaiKhoan\_KhiGVBNgungHoatDong  ON GiangVien  AFTER UPDATE  AS  BEGIN  SET NOCOUNT ON;  UPDATE tk  SET tk.TrangThai = 0  FROM TaiKhoan tk  INNER JOIN inserted i ON tk.MaGV = i.MaGV  WHERE i.TrangThai = 1;  END; |

**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 views

**a. View hiển thị 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; |

**b. View hiển thị danh sách giảng viên**

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

**c. View hiển thị danh sách môn học**

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

**d. View hiển thị thông tin chi tiết sinh viên**

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

**e. View danh sách các thành viên bị khoá**

|  |
| --- |
| CREATE VIEW dbo.ThanhVienBiKhoa AS  SELECT tv.MaThanhVien,  tv.TenThanhVien,  SUM(ISNULL(p.SoTien,0)) AS TongNo,  MAX(DATEDIFF(DAY, c.NgayTraDuKien, GETDATE())) AS MaxNgayQuaHan  FROM dbo.ThanhVien tv  LEFT JOIN dbo.PhieuMuon pm ON pm.MaThanhVien = tv.MaThanhVien  LEFT JOIN dbo.ChiTietPhieuMuon c ON pm.MaPhieuMuon = c.MaPhieuMuon AND c.NgayTraThucTe IS NULL  LEFT JOIN dbo.Phat p ON c.MaChiTiet = p.MaMuonSach AND p.TrangThai = N'ChuaThanhToan'  GROUP BY tv.MaThanhVien, tv.TenThanhVien  HAVING SUM(ISNULL(p.SoTien,0)) > 0 OR MAX(DATEDIFF(DAY, c.NgayTraDuKien, GETDATE())) > 7; |

**f. View danh sách top sách được mượn nhiều**

|  |
| --- |
| CREATE VIEW dbo.vw\_TopBooks AS  SELECT s.TenSach, COUNT(\*) AS SoLanMuon, pm.NgayMuon  FROM dbo.ChiTietPhieuMuon ct  JOIN dbo.PhieuMuon pm ON ct.MaPhieuMuon = pm.MaPhieuMuon  JOIN dbo.BanSaoSach bs ON ct.MaBanSao = bs.MaBanSao  JOIN dbo.Sach s ON bs.MaSach = s.MaSach  GROUP BY s.TenSach, pm.NgayMuon; |

**g. View danh sách tổng tiền phạt**

|  |
| --- |
| CREATE VIEW dbo.vw\_TongTienPhat AS  SELECT ph.PaymentDate, ph.Amount  FROM dbo.PaymentHistory ph; |

**h. View danh sách phiếu mượn chi tiết**

|  |
| --- |
| CREATE VIEW dbo.vw\_PhieuMuonChiTiet AS  SELECT pm.MaPhieuMuon,  pm.NgayMuon,  pm.NgayTraDuKien,  ct.NgayTraThucTe,  tv.TenThanhVien,  s.TenSach  FROM dbo.ChiTietPhieuMuon ct  JOIN dbo.PhieuMuon pm ON ct.MaPhieuMuon = pm.MaPhieuMuon  JOIN dbo.ThanhVien tv ON pm.MaThanhVien = tv.MaThanhVien  JOIN dbo.BanSaoSach bs ON ct.MaBanSao = bs.MaBanSao  JOIN dbo.Sach s ON bs.MaSach = s.MaSach; |

**i. View tình trạng sách**

|  |
| --- |
| CREATE VIEW dbo.vw\_TinhTrangSach AS  SELECT TinhTrang  FROM dbo.BanSaoSach; |

**k. View danh sách sách quá hạn theo thành viên**

|  |
| --- |
| CREATE VIEW dbo.SachQuaHanTheoThanhVien AS  SELECT  tv.MaThanhVien,  tv.TenThanhVien,  COUNT(o.MaChiTiet) AS TongSoSachQuaHan,  SUM(o.SoNgayQuaHan) AS TongNgayQuaHan,  ISNULL(SUM(CASE  WHEN ph.MaPhat IS NOT NULL THEN ph.SoTien  ELSE dbo.fn\_TinhPhat(o.NgayTraDuKien, GETDATE())  END), 0) AS TongPhat,  STRING\_AGG(  CONCAT(  o.TenSach,  N' (Quá hạn: ', o.SoNgayQuaHan, N' ngày',  CASE  WHEN ph.MaPhat IS NOT NULL THEN CONCAT(N', Phí: ', CAST(ph.SoTien AS NVARCHAR), N' VND)')  ELSE CONCAT(N', Dự kiến: ', CAST(dbo.fn\_TinhPhat(o.NgayTraDuKien, GETDATE()) AS NVARCHAR), N' VND)')  END  ), '; '  ) AS DanhSachSachQuaHan  FROM dbo.ThanhVien tv  CROSS APPLY dbo.RTO\_GetOverdueBooksByMember(tv.MaThanhVien) o  INNER JOIN dbo.PhieuMuon p ON p.MaThanhVien = tv.MaThanhVien  INNER JOIN dbo.ChiTietPhieuMuon c ON c.MaPhieuMuon = p.MaPhieuMuon AND c.MaChiTiet = o.MaChiTiet  LEFT JOIN dbo.Phat ph ON ph.MaMuonSach = o.MaChiTiet AND ph.TrangThai = N'ChuaThanhToan'  GROUP BY tv.MaThanhVien, tv.TenThanhVien; |

## 8. 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 trả về giá tính phạt quá hạn**

|  |
| --- |
| CREATE FUNCTION dbo.TinhPhatQuaHan (@NgayTraDuKien DATE, @NgayTraThucTe DATE)  RETURNS DECIMAL(10,2)  AS  BEGIN  DECLARE @SoNgay INT = DATEDIFF(DAY, @NgayTraDuKien, @NgayTraThucTe);  DECLARE @PhiPhatNgay DECIMAL(10,2);  SELECT @PhiPhatNgay = cht.GiaTri  FROM dbo.CauHinhHeThong cht  WHERE cht.TenCauHinh = N'PhiPhatTreHan';  SET @PhiPhatNgay = ISNULL(@PhiPhatNgay, 5000.00);  RETURN IIF(@SoNgay > 0, @SoNgay \* @PhiPhatNgay, 0);  END; |

**c. Hàm trả về bảng danh sách sinh viên và 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 |

**d. Hàm trả về danh sách sinh viên theo lớp đại học**

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

## 

1. **Chức năng hiển thị**

**Thông tin giảng viên**

|  |
| --- |
| CREATE FUNCTION dbo.fn\_GetThongTinGV(@MaGV VARCHAR(10))  RETURNS TABLE  AS  RETURN  (  SELECT MaGV, HoTenGV, HocVi, Khoa, Email, DienThoai  FROM GiangVien  WHERE 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 sinh viên**

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

C#:

|  |
| --- |
| 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 giảng viên**

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

C#:

|  |
| --- |
| 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 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); |

**4. Chức năng quản lý sinh viên**

**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]);  }  } |

1. **Chức năng quản lý giảng viên**

**Chức năng thêm giảng viên**

|  |
| --- |
| CREATE PROCEDURE dbo.sp\_ThemGiangVien  @MaGV VARCHAR(10),  @HoTenGV NVARCHAR(100),  @HocVi NVARCHAR(50) = NULL,  @Khoa NVARCHAR(50) = 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, Khoa, Email, DienThoai)  VALUES (@MaGV, @HoTenGV, @HocVi, @Khoa, @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 |

C#;

|  |
| --- |
| 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 khoa = row["Khoa"]?.ToString() ?? "";  string email = row["Email"]?.ToString() ?? "";  string dienThoai = row["DienThoai"]?.ToString() ?? "";  if (string.IsNullOrWhiteSpace(maGV) || string.IsNullOrWhiteSpace(hoTenGV))  {  MessageBox.Show("Mã giảng viên và Họ tên giảng viên 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}'")},  @Khoa = {(string.IsNullOrWhiteSpace(khoa) ? "NULL" : $"N'{khoa}'")},  @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,  @Khoa NVARCHAR(50) = 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,  Khoa = @Khoa,  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 |

C#:

|  |
| --- |
| if (row.RowState == DataRowState.Modified)  {  string maGV = row["MaGV"].ToString().Trim();  string hoTenGV = row["HoTenGV"].ToString().Trim();  string hocVi = row["HocVi"]?.ToString() ?? "";  string khoa = row["Khoa"]?.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}'")},  @Khoa = {(string.IsNullOrWhiteSpace(khoa) ? "NULL" : $"N'{khoa}'")},  @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;  DELETE DKMH  FROM DangKyMonHoc DKMH  INNER JOIN LopHocPhan LHP ON DKMH.MaLHP = LHP.MaLHP  WHERE LHP.MaGV = @MaGV;  DELETE FROM LopHocPhan WHERE MaGV = @MaGV;  DELETE FROM TaiKhoan WHERE MaGV = @MaGV;  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]);  } |

1. **Danh sách đăng ký môn học theo kỳ**

|  |
| --- |
| 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  INNER JOIN MonHoc MH ON LHP.MaMH = MH.MaMH  WHERE DKMH.MaHocKyNamHoc = @MaHocKyNamHoc AND SV.TrangThai = 0  );  GO |

C#;

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

1. Danh sách giảng dạy theo năm học

|  |
| --- |
| CREATE FUNCTION dbo.fn\_LopHocPhanTheoNamHoc  (  @MaHocKyNamHoc INT  )  RETURNS TABLE  AS  RETURN  (  SELECT  LHP.MaMH,  MH.TenMH,  LHP.MaLHP,  GV.MaGV,  GV.HoTenGV,  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; |

8. Quản lý kết quả học tập

Hiển thị bảng điểm theo 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  ); |

C#:

|  |
| --- |
| maHocKyNamHoc = Convert.ToInt32(drv["MaHocKyNamHoc"]);  string query = $"SELECT \* FROM fn\_ChiTietHocPhan({maHocKyNamHoc})";  dt = frmAdmin.getData(query); |

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); |

Thống kê kết quả theo môn 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";  } |

9. Điểm chi tiết sinh viên

2.6.1 Lấy 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);  }  } |

* + 1. **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";  }  } |

2.6.3 Hiển thị thông tin chi tiết sinh viên

|  |
| --- |
| 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();  } |

2.6.4 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; |

**Thiết lập số tín chỉ**

|  |
| --- |
| 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();  } |

Quản lý tài khoản

2.8.1 hiển thị 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);  }  } |

2.8.2 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]);  } } |

2.8.2 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);  -- 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 + '];  ';  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);    } |

2.8.3 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);  } |

* 1. lịch sử đăng nhập

1. danh sách môn học 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; command.Parameters.Add("@SoDienThoai", SqlDbType.NVarChar, 20).Value =  string.IsNullOrWhiteSpace(soDienThoai) ? (object)DBNull.Value : soDienThoai;  command.Parameters.Add("@DiaChi", SqlDbType.NVarChar, 200).Value =  string.IsNullOrWhiteSpace(diaChi) ? (object)DBNull.Value : diaChi;  command.Parameters.Add("@LoaiThanhVien", SqlDbType.NVarChar, 20).Value = loaiThanhVien;  command.Parameters.Add("@NgayDangKy", SqlDbType.Date).Value =  ngayDangKy.HasValue ? (object)ngayDangKy.Value : DBNull.Value;  connection.Open();  command.ExecuteNonQuery();  }  }  } |

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);  }  } |

Lấy 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);  }  } |

* 1. **Xoá sinh viên khỏi lớp học phần (xoá đăng kí môn học )**

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

C#:

|  |
| --- |
| private void btnXoa\_ItemClick(object sender, DevExpress.XtraBars.ItemClickEventArgs e)  {  int rowHandle = gvDanhSach.FocusedRowHandle;  if (rowHandle < 0) return;  DataRow row = gvDanhSach.GetDataRow(rowHandle);  if (row == null) return;  string maSV = row["MaSV"].ToString();  string maLHP = cbbMa.SelectedValue.ToString();  DialogResult result = MessageBox.Show(  $"Bạn có chắc chắn muốn xóa sinh viên {maSV} khỏi lớp học phần {maLHP} không?",  "Xác nhận xóa",  MessageBoxButtons.YesNo,  MessageBoxIcon.Question  );  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);  }  }  } |

* 1. **Chức năng trả sách**

|  |
| --- |
| CREATE PROCEDURE [dbo].[TraSachProc]  @MaBanSao INT,  @UserID INT,  @TinhTrangSach NVARCHAR(20) = N'BinhThuong', -- 'BinhThuong' | 'HuHong' | 'Mat'  @GhiChu NVARCHAR(255) = NULL  AS  BEGIN  SET NOCOUNT ON;  SET XACT\_ABORT ON;  BEGIN TRY  BEGIN TRAN;  DECLARE  @MaChiTiet INT,  @MaPhieuMuon INT,  @NgayTraDuKien DATE,  @SoNgayTre INT,  @SoTienPhat DECIMAL(18,0),  @MaSach INT,  @GiaSach DECIMAL(18,0),  @PhiPhatNgay DECIMAL(10,2),  @PhiSachHuHong DECIMAL(10,2);  SELECT  @PhiPhatNgay = ISNULL(c1.GiaTri, 5000.00),  @PhiSachHuHong = ISNULL(c2.GiaTri, 50000.00)  FROM [dbo].[CauHinhHeThong] c1  LEFT JOIN [dbo].[CauHinhHeThong] c2 ON c2.TenCauHinh = N'PhiSachHuHong'  WHERE c1.TenCauHinh = N'PhiPhatTreHan';  -- Lấy chi tiết mượn còn đang mở  SELECT TOP 1  @MaChiTiet = ct.MaChiTiet,  @MaPhieuMuon = ct.MaPhieuMuon,  @NgayTraDuKien = ct.NgayTraDuKien,  @MaSach = bs.MaSach  FROM ChiTietPhieuMuon ct  JOIN BanSaoSach bs ON ct.MaBanSao = bs.MaBanSao  WHERE ct.MaBanSao = @MaBanSao  AND ct.NgayTraThucTe IS NULL;  IF @MaChiTiet IS NULL  BEGIN  RAISERROR(N'Không tìm thấy phiếu mượn đang mở cho bản sao này.',16,1);  ROLLBACK TRAN;  RETURN;  END  -- Lấy giá sách (nếu có cột Gia trong bảng Sach)  SELECT @GiaSach = s.Gia  FROM Sach s  WHERE s.MaSach = @MaSach;  -- Cập nhật chi tiết mượn: Ngày trả + Ghi chú  UPDATE ChiTietPhieuMuon  SET NgayTraThucTe = GETDATE(),  TrangThai = N'Đã trả',  GhiChu = @GhiChu  WHERE MaChiTiet = @MaChiTiet;  -- Xử lý tình trạng sách  IF @TinhTrangSach = N'BinhThuong'  BEGIN  UPDATE BanSaoSach  SET TinhTrang = N'Sẵn sàng'  WHERE MaBanSao = @MaBanSao;  END  ELSE IF @TinhTrangSach = N'HuHong'  BEGIN  UPDATE BanSaoSach  SET TinhTrang = N'Hư hỏng'  WHERE MaBanSao = @MaBanSao;  INSERT INTO Phat(MaMuonSach, SoTien, LyDo, TrangThai)  VALUES(@MaChiTiet, @PhiSachHuHong, N'Sách bị hư hỏng', N'Chưa thanh toán');  END  ELSE IF @TinhTrangSach = N'Mat'  BEGIN  UPDATE BanSaoSach  SET TinhTrang = N'Mất'  WHERE MaBanSao = @MaBanSao;  INSERT INTO Phat(MaMuonSach, SoTien, LyDo, TrangThai)  VALUES(@MaChiTiet, ISNULL(@GiaSach, 100000), N'Mất sách', N'Chưa thanh toán');  END    -- Nếu tất cả chi tiết của phiếu đã trả → cập nhật phiếu  IF NOT EXISTS (SELECT 1 FROM ChiTietPhieuMuon WHERE MaPhieuMuon = @MaPhieuMuon AND NgayTraThucTe IS NULL)  BEGIN  UPDATE PhieuMuon  SET TrangThai = N'Đã trả hết'  WHERE MaPhieuMuon = @MaPhieuMuon;  END  COMMIT TRAN;  END TRY  BEGIN CATCH  IF XACT\_STATE()<>0 ROLLBACK TRAN;  THROW;  END CATCH  END; |

C#:

|  |
| --- |
| public void ExecuteTraNhieuSachProc(List<int> listMaBanSao, int userId, string tinhTrangSach, string ghiChu = null)  {  if (listMaBanSao == null || listMaBanSao.Count == 0)  throw new ArgumentException("listMaBanSao is empty.");  var tinhTrang = (tinhTrangSach ?? "BinhThuong").Trim();  using (var connection = new SqlConnection(\_connectionString))  {  var errors = new StringBuilder();  var successCount = 0;  connection.InfoMessage += (s, e) => { };  connection.Open();  foreach (var maBanSao in listMaBanSao)  {  try  {  using (var command = new SqlCommand("[dbo].[TraSachProc]", connection))  {  command.CommandType = CommandType.StoredProcedure;  command.CommandTimeout = 60;  command.Parameters.Add("@MaBanSao", SqlDbType.Int).Value = maBanSao;  command.Parameters.Add("@UserID", SqlDbType.Int).Value = userId;  command.Parameters.Add("@TinhTrangSach", SqlDbType.NVarChar, 20).Value = tinhTrang;  command.Parameters.Add("@GhiChu", SqlDbType.NVarChar, 255).Value =  string.IsNullOrWhiteSpace(ghiChu) ? (object)DBNull.Value : ghiChu.Trim();  command.ExecuteNonQuery();  successCount++;  }  }  catch (Exception ex)  {  errors.AppendLine($"Error processing MaBanSao={maBanSao}: {ex.Message}");  }  }  if (errors.Length > 0)  {  if (successCount == 0)  throw new Exception($"Failed to process any books. Errors: {errors}");  throw new Exception($"Processed {successCount}/{listMaBanSao.Count}. Errors: {errors}");  }  }  } |

## Danh mục Phạt – Lịch sử thanh toán

Chức năng thanh toán phạt

|  |
| --- |
| CREATE PROCEDURE [dbo].[ThanhToanPhat]  @MaPhat INT,  @Method NVARCHAR(50) = 'Cash',  @Note NVARCHAR(200) = NULL  AS  BEGIN  SET NOCOUNT ON;  SET XACT\_ABORT ON;  BEGIN TRY  BEGIN TRAN;  DECLARE @MaChiTiet INT, @MaThanhVien INT, @SoTien DECIMAL(18,2);  -- Lấy chi tiết phạt  SELECT @MaChiTiet = MaMuonSach, @SoTien = SoTien  FROM dbo.Phat  WHERE MaPhat = @MaPhat;  IF @MaChiTiet IS NULL  THROW 55001, N'MaPhat không hợp lệ', 1;  -- Lấy thành viên  SELECT @MaThanhVien = p.MaThanhVien  FROM dbo.ChiTietPhieuMuon c  JOIN dbo.PhieuMuon p ON c.MaPhieuMuon = p.MaPhieuMuon  WHERE c.MaChiTiet = @MaChiTiet;  -- Insert lịch sử thanh toán (Amount lấy từ bảng Phat)  INSERT INTO dbo.PaymentHistory (MaPhat, MaThanhVien, Amount, PaymentDate, Method, Note)  VALUES (@MaPhat, @MaThanhVien, @SoTien, GETDATE(), @Method,@Note);  -- Update trạng thái phạt  UPDATE dbo.Phat SET TrangThai = N'Đã thanh toán'  WHERE MaPhat = @MaPhat;  -- Audit log  INSERT INTO dbo.AuditLog (UserName, Action, EntityName, EntityID, Details)  VALUES (  ORIGINAL\_LOGIN(),  'ThanhToanPhat',  'Phat',  CAST(@MaPhat AS NVARCHAR(50)),  'Amount=' + CAST(@SoTien AS NVARCHAR(50)) + '; Method=' + ISNULL(@Method,'')  );  COMMIT TRAN;  END TRY  BEGIN CATCH  IF XACT\_STATE()<>0 ROLLBACK TRAN;  THROW;  END CATCH  END; |

C#:

|  |
| --- |
| public void ExecuteThanhToanPhatProc(int maPhat, string note, string method = "Tiền mặt")  {  using (var connection = new SqlConnection(\_connectionString))  {  using (var command = new SqlCommand("ThanhToanPhat", connection))  {  command.CommandType = CommandType.StoredProcedure;  command.CommandTimeout = 60;  command.Parameters.Add("@MaPhat", SqlDbType.Int).Value = maPhat;  command.Parameters.Add("@Method", SqlDbType.NVarChar, 50).Value = method ?? "Tiền mặt";  command.Parameters.Add("@Note", SqlDbType.NVarChar, 200).Value = note ?? (object)DBNull.Value;  connection.Open();  command.ExecuteNonQuery();  }  }  } |

## Danh mục Thống kê - Báo cáo

* 1. **Biểu đồ Top sách mượn theo ngày**

|  |
| --- |
| CREATE PROCEDURE [dbo].[sp\_TopSachMuonTheoNgay]  @FromDate DATE,  @ToDate DATE  AS  BEGIN  SELECT TOP 10 TenSach, COUNT(\*) AS SoLanMuon  FROM vw\_TopBooks  WHERE NgayMuon BETWEEN @FromDate AND @ToDate  GROUP BY TenSach  ORDER BY SoLanMuon DESC;  END |

C#:

|  |
| --- |
| public DataTable GetTopSachMuon(DateTime fromDate, DateTime toDate)  {  using (var conn = new SqlConnection(\_connectionString))  using (var cmd = new SqlCommand("sp\_TopSachMuonTheoNgay", conn))  {  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@FromDate", fromDate);  cmd.Parameters.AddWithValue("@ToDate", toDate);  var adapter = new SqlDataAdapter(cmd);  var dt = new DataTable();  adapter.Fill(dt);  return dt;  }  } |

* 1. **Biểu đồ Tổng tiền phạt theo tuần**

|  |
| --- |
| CREATE PROCEDURE [dbo].[sp\_TongTienPhatTheoTuan]  @Year INT  AS  BEGIN  SELECT DATEPART(WEEK, PaymentDate) AS Tuan,  SUM(Amount) AS TongTien  FROM vw\_TongTienPhat  WHERE YEAR(PaymentDate) = @Year  GROUP BY DATEPART(WEEK, PaymentDate)  ORDER BY Tuan;  END |

C#:

|  |
| --- |
| public DataTable GetTongTienPhatTheoTuan(int year)  {  using (var conn = new SqlConnection(\_connectionString))  using (var cmd = new SqlCommand("sp\_TongTienPhatTheoTuan", conn))  {  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@Year", year);  var adapter = new SqlDataAdapter(cmd);  var dt = new DataTable();  adapter.Fill(dt);  return dt;  }  } |

* 1. **Biểu đồ Tổng tiền phạt theo tháng**

|  |
| --- |
| CREATE PROCEDURE [dbo].[sp\_TongTienPhatTheoThang]  @Year INT  AS  BEGIN  SELECT MONTH(PaymentDate) AS Thang, SUM(Amount) AS TongTien  FROM vw\_TongTienPhat  WHERE YEAR(PaymentDate) = @Year  GROUP BY MONTH(PaymentDate)  ORDER BY Thang;  END |

C#:

|  |
| --- |
| public DataTable GetTongTienPhatTheoThang(int year)  {  using (var conn = new SqlConnection(\_connectionString))  using (var cmd = new SqlCommand("sp\_TongTienPhatTheoThang", conn))  {  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@Year", year);  var adapter = new SqlDataAdapter(cmd);  var dt = new DataTable();  adapter.Fill(dt);  return dt; } } |

* 1. **Biểu đồ Tổng tiền phạt theo quý**

|  |
| --- |
| CREATE PROCEDURE [dbo].[sp\_TongTienPhatTheoQuy]  @Year INT  AS  BEGIN  SELECT DATEPART(QUARTER, PaymentDate) AS Quy,  SUM(Amount) AS TongTien  FROM vw\_TongTienPhat  WHERE YEAR(PaymentDate) = @Year  GROUP BY DATEPART(QUARTER, PaymentDate)  ORDER BY Quy;  END |

C#:

|  |
| --- |
| public DataTable GetTongTienPhatTheoQuy(int year)  {  using (var conn = new SqlConnection(\_connectionString))  using (var cmd = new SqlCommand("sp\_TongTienPhatTheoQuy", conn))  {  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@Year", year);  var adapter = new SqlDataAdapter(cmd);  var dt = new DataTable();  adapter.Fill(dt);  return dt;  }  } |

* 1. **Biểu đồ Tổng tiền phạt theo năm**

|  |
| --- |
| CREATE PROCEDURE [dbo].[sp\_TongTienPhatTheoNam]  AS  BEGIN  SELECT YEAR(PaymentDate) AS Nam,  SUM(Amount) AS TongTien  FROM vw\_TongTienPhat  GROUP BY YEAR(PaymentDate)  ORDER BY Nam;  END |

C#:

|  |
| --- |
| public DataTable GetTongTienPhatTheoNam()  {  using (var conn = new SqlConnection(\_connectionString))  using (var cmd = new SqlCommand("sp\_TongTienPhatTheoNam", conn))  {  cmd.CommandType = CommandType.StoredProcedure;  var adapter = new SqlDataAdapter(cmd);  var dt = new DataTable();  adapter.Fill(dt);  return dt;  }  } |

* 1. **Biểu đồ Thống kê tình trạng sách**

|  |
| --- |
| CREATE PROCEDURE [dbo].[sp\_ThongKeTinhTrangSach]  AS  BEGIN  SELECT TinhTrang, COUNT(\*) AS SoLuong  FROM vw\_TinhTrangSach  GROUP BY TinhTrang;  END  GO |

C#:

|  |
| --- |
| public DataTable GetThongKeTinhTrangSach()  {  using (var conn = new SqlConnection(\_connectionString))  using (var cmd = new SqlCommand("sp\_ThongKeTinhTrangSach", conn))  {  cmd.CommandType = CommandType.StoredProcedure;  var adapter = new SqlDataAdapter(cmd);  var dt = new DataTable();  adapter.Fill(dt);  return dt;  }  } |

* 1. **Biểu đồ Thống kê sách quá hạn**

|  |
| --- |
| CREATE VIEW dbo.SachQuaHan AS  SELECT c.MaChiTiet AS MaMuonSach,  s.TenSach,  tv.TenThanhVien,  p.NgayMuon,  c.NgayTraDuKien,  c.TrangThai  FROM dbo.ChiTietPhieuMuon c  INNER JOIN dbo.PhieuMuon p ON c.MaPhieuMuon = p.MaPhieuMuon  INNER JOIN dbo.Sach s ON c.MaSach = s.MaSach  INNER JOIN dbo.ThanhVien tv ON p.MaThanhVien = tv.MaThanhVien  WHERE c.NgayTraThucTe IS NULL  AND c.NgayTraDuKien < GETDATE(); |

C#:

|  |
| --- |
| public DataTable GetSachQuaHan()  {  using (var conn = new SqlConnection(\_connectionString))  using (var cmd = new SqlCommand("SELECT \* FROM SachQuaHan", conn))  {  var adapter = new SqlDataAdapter(cmd);  var dt = new DataTable();  adapter.Fill(dt);  return dt;  }  } |

* 1. **Biểu đồ Thống kê sách mượn theo loại thành viên**

|  |
| --- |
| CREATE FUNCTION dbo.RTM\_GetBorrowingStatsByMemberType(  @LoaiThanhVien NVARCHAR(20)  )  RETURNS @Stats TABLE  (  LoaiThanhVien NVARCHAR(20),  SoThanhVien INT,  TongSachMuon INT,  TyLeQuaHan DECIMAL(5,2),  TongPhat DECIMAL(18,2)  )  AS  BEGIN  INSERT INTO @Stats (LoaiThanhVien, SoThanhVien)  SELECT @LoaiThanhVien, COUNT(\*) FROM dbo.ThanhVien WHERE LoaiThanhVien = @LoaiThanhVien;  DECLARE @TongMuon INT, @QuaHan INT;  SELECT @TongMuon = COUNT(c.MaChiTiet),  @QuaHan = COUNT(CASE WHEN c.NgayTraDuKien < GETDATE() AND c.NgayTraThucTe IS NULL THEN 1 END)  FROM dbo.ChiTietPhieuMuon c  INNER JOIN dbo.PhieuMuon p ON c.MaPhieuMuon = p.MaPhieuMuon  INNER JOIN dbo.ThanhVien tv ON p.MaThanhVien = tv.MaThanhVien  WHERE tv.LoaiThanhVien = @LoaiThanhVien;  UPDATE @Stats  SET TongSachMuon = @TongMuon,  TyLeQuaHan = CASE WHEN @TongMuon > 0 THEN (@QuaHan \* 100.0 / @TongMuon) ELSE 0 END;  DECLARE @TongPhat DECIMAL(18,2) = (SELECT ISNULL(SUM(ph.SoTien), 0)  FROM dbo.Phat ph  INNER JOIN dbo.ChiTietPhieuMuon c ON ph.MaMuonSach = c.MaChiTiet  INNER JOIN dbo.PhieuMuon p ON c.MaPhieuMuon = p.MaPhieuMuon  INNER JOIN dbo.ThanhVien tv ON p.MaThanhVien = tv.MaThanhVien  WHERE tv.LoaiThanhVien = @LoaiThanhVien AND ph.TrangThai = N'ChuaThanhToan');  UPDATE @Stats SET TongPhat = @TongPhat;  RETURN;  END; |

C#:

|  |
| --- |
| public DataTable GetMuonSachByType(string loaiThanhVien)  {  using (var conn = new SqlConnection(\_connectionString))  using (var cmd = new SqlCommand("SELECT \* FROM RTM\_GetBorrowingStatsByMemberType(@LoaiThanhVien)", conn))  {  cmd.Parameters.AddWithValue("@LoaiThanhVien", loaiThanhVien);  var adapter = new SqlDataAdapter(cmd);  var dt = new DataTable();  adapter.Fill(dt);  return dt;  }  } |

## Danh mục Người dùng

Thêm Người dùng

|  |
| --- |
| CREATE PROCEDURE [dbo].[ThemUsers]  @UserName NVARCHAR(100),  @FullName NVARCHAR(200) = NULL,  @PasswordHash NVARCHAR(300),  @Email NVARCHAR(200) = NULL,  @RoleID INT = NULL,  @IsActive BIT = 1,  @CreatedAt DATETIME = NULL  AS  BEGIN  SET NOCOUNT ON;  BEGIN TRY  -- Kiểm tra định dạng Email nếu có  IF @Email IS NOT NULL AND @Email NOT LIKE '%@%.%'  BEGIN  RAISERROR (N'Email không hợp lệ.', 16, 1);  RETURN;  END  -- Kiểm tra RoleID hợp lệ nếu có  IF @RoleID IS NOT NULL AND NOT EXISTS (SELECT 1 FROM [dbo].[Roles] WHERE RoleID = @RoleID)  BEGIN  RAISERROR (N'RoleID không tồn tại.', 16, 1);  RETURN;  END  -- Gán giá trị mặc định cho CreatedAt nếu NULL  SET @CreatedAt = COALESCE(@CreatedAt, GETDATE());  INSERT INTO [dbo].[Users] (UserName, FullName, PasswordHash, Email, RoleID, IsActive, CreatedAt)  VALUES (@UserName, @FullName, @PasswordHash, @Email, @RoleID, @IsActive, @CreatedAt);  END TRY  BEGIN CATCH  DECLARE @ErrorMessage NVARCHAR(4000) = ERROR\_MESSAGE();  RAISERROR (@ErrorMessage, 16, 1);  END CATCH  END; |

C#:

|  |
| --- |
| public void AddUser(string userName, string fullName, string passwordHash, string email = null,  int? roleId = null, bool isActive = true, DateTime? createdAt = null)  {  if (string.IsNullOrWhiteSpace(userName))  throw new ArgumentException("Tên người dùng không được để trống", nameof(userName));  if (string.IsNullOrWhiteSpace(passwordHash))  throw new ArgumentException("Mật khẩu không được để trống", nameof(passwordHash));  var currentConnectionString = ConnectionStringProvider.GetConnectionString();    using (var connection = new SqlConnection(currentConnectionString))  {  using (var command = new SqlCommand("[dbo].[ThemUsers]", connection))  {  command.CommandType = CommandType.StoredProcedure;  command.CommandTimeout = 60;  command.Parameters.Add("@UserName", SqlDbType.NVarChar, 100).Value = userName;  command.Parameters.Add("@FullName", SqlDbType.NVarChar, 200).Value =  string.IsNullOrWhiteSpace(fullName) ? (object)DBNull.Value : fullName;  command.Parameters.Add("@PasswordHash", SqlDbType.NVarChar, 300).Value = passwordHash;  command.Parameters.Add("@Email", SqlDbType.NVarChar, 200).Value =  string.IsNullOrWhiteSpace(email) ? (object)DBNull.Value : email;  command.Parameters.Add("@RoleID", SqlDbType.Int).Value =  roleId.HasValue ? (object)roleId.Value : DBNull.Value;  command.Parameters.Add("@IsActive", SqlDbType.Bit).Value = isActive;  command.Parameters.Add("@CreatedAt", SqlDbType.DateTime).Value =  createdAt.HasValue ? (object)createdAt.Value : DBNull.Value;  connection.Open();  command.ExecuteNonQuery();  }  }  } |

Sửa Người dùng

|  |
| --- |
| CREATE PROCEDURE [dbo].[SuaUsers]  @UserID INT,  @UserName NVARCHAR(100),  @FullName NVARCHAR(200) = NULL,  @PasswordHash NVARCHAR(300),  @Email NVARCHAR(200) = NULL,  @RoleID INT = NULL,  @IsActive BIT  AS  BEGIN  SET NOCOUNT ON;  BEGIN TRY  -- Kiểm tra định dạng Email nếu có  IF @Email IS NOT NULL AND @Email NOT LIKE '%@%.%'  BEGIN  RAISERROR (N'Email không hợp lệ.', 16, 1);  RETURN;  END  -- Kiểm tra RoleID hợp lệ nếu có  IF @RoleID IS NOT NULL AND NOT EXISTS (SELECT 1 FROM [dbo].[Roles] WHERE RoleID = @RoleID)  BEGIN  RAISERROR (N'RoleID không tồn tại.', 16, 1);  RETURN;  END  UPDATE [dbo].[Users]  SET UserName = @UserName,  FullName = @FullName,  PasswordHash = @PasswordHash,  Email = @Email,  RoleID = @RoleID,  IsActive = @IsActive  WHERE UserID = @UserID;  END TRY  BEGIN CATCH  DECLARE @ErrorMessage NVARCHAR(4000) = ERROR\_MESSAGE();  RAISERROR (@ErrorMessage, 16, 1);  END CATCH  END; |

C#:

|  |
| --- |
| public void UpdateUser(int userId, string userName, string fullName, string passwordHash, string email = null,  int? roleId = null, bool isActive = true)  {  if (userId <= 0)  throw new ArgumentException("ID người dùng không hợp lệ", nameof(userId));  if (string.IsNullOrWhiteSpace(userName))  throw new ArgumentException("Tên người dùng không được để trống", nameof(userName));  if (string.IsNullOrWhiteSpace(passwordHash))  throw new ArgumentException("Mật khẩu không được để trống", nameof(passwordHash));  var currentConnectionString = ConnectionStringProvider.GetConnectionString();  using (var connection = new SqlConnection(currentConnectionString))  {  using (var command = new SqlCommand("[dbo].[SuaUsers]", connection))  {  command.CommandType = CommandType.StoredProcedure;  command.CommandTimeout = 60;  command.Parameters.Add("@UserID", SqlDbType.Int).Value = userId;  command.Parameters.Add("@UserName", SqlDbType.NVarChar, 100).Value = userName;  command.Parameters.Add("@FullName", SqlDbType.NVarChar, 200).Value =  string.IsNullOrWhiteSpace(fullName) ? (object)DBNull.Value : fullName;  command.Parameters.Add("@PasswordHash", SqlDbType.NVarChar, 300).Value = passwordHash;  command.Parameters.Add("@Email", SqlDbType.NVarChar, 200).Value =  string.IsNullOrWhiteSpace(email) ? (object)DBNull.Value : email;  command.Parameters.Add("@RoleID", SqlDbType.Int).Value =  roleId.HasValue ? (object)roleId.Value : DBNull.Value;  command.Parameters.Add("@IsActive", SqlDbType.Bit).Value = isActive;  connection.Open();  command.ExecuteNonQuery();  }  }  } |

Xoá Người dùng

|  |
| --- |
| CREATE PROCEDURE [dbo].[XoaUsers]  @UserID INT  AS  BEGIN  SET NOCOUNT ON;  BEGIN TRY  -- Kiểm tra xem User có được tham chiếu trong bảng PhieuMuon không  IF EXISTS (SELECT 1 FROM [dbo].[PhieuMuon] WHERE UserID = @UserID)  BEGIN  RAISERROR (N'Không thể xóa người dùng vì vẫn còn phiếu mượn liên quan.', 16, 1);  RETURN;  END  DELETE FROM [dbo].[Users] WHERE UserID = @UserID;  END TRY  BEGIN CATCH  DECLARE @ErrorMessage NVARCHAR(4000) = ERROR\_MESSAGE();  RAISERROR (@ErrorMessage, 16, 1);  END CATCH  END; |

C#:

|  |
| --- |
| public void DeleteUser(int userId)  {  if (userId <= 0)  throw new ArgumentException("ID người dùng không hợp lệ", nameof(userId));  var currentConnectionString = ConnectionStringProvider.GetConnectionString();  using (var connection = new SqlConnection(currentConnectionString))  {  using (var command = new SqlCommand("[dbo].[XoaUsers]", connection))  {  command.CommandType = CommandType.StoredProcedure;  command.CommandTimeout = 60;  command.Parameters.Add("@UserID", SqlDbType.Int).Value = userId;  connection.Open();  command.ExecuteNonQuery();  }  }  } |

## Danh mục Cấu hình hệ thống – Nhật ký hệ thống

Lấy danh sách cấu hình

|  |
| --- |
| CREATE PROCEDURE [dbo].[sp\_LayDanhSachCauHinh]  AS  BEGIN  SELECT TenCauHinh, GiaTri, MoTa, NgayCapNhat  FROM [dbo].[CauHinhHeThong]  ORDER BY TenCauHinh;  END; |

C#:

|  |
| --- |
| public DataTable LoadCauHinh()  {    using (var conn = new SqlConnection(\_connectionString))  using (var cmd = new SqlCommand("sp\_LayDanhSachCauHinh", conn))  {  cmd.CommandType = CommandType.StoredProcedure;  var adapter = new SqlDataAdapter(cmd);  var dt = new DataTable();  adapter.Fill(dt);  return dt;  }  } |

Cập nhật cấu hình hệ thống

|  |
| --- |
| CREATE PROCEDURE [dbo].[sp\_CapNhatCauHinhHeThong]  @TenCauHinh NVARCHAR(100),  @GiaTriMoi DECIMAL(10,2),  @MoTa NVARCHAR(200) = NULL  AS  BEGIN  SET NOCOUNT ON;  SET XACT\_ABORT ON;    BEGIN TRY  BEGIN TRAN;    IF EXISTS (SELECT 1 FROM [dbo].[CauHinhHeThong] WHERE TenCauHinh = @TenCauHinh)  BEGIN  UPDATE [dbo].[CauHinhHeThong]  SET GiaTri = @GiaTriMoi,  MoTa = ISNULL(@MoTa, MoTa),  NgayCapNhat = GETDATE()  WHERE TenCauHinh = @TenCauHinh;    INSERT INTO [dbo].[AuditLog] (UserName, Action, EntityName, EntityID, Details)  VALUES (ORIGINAL\_LOGIN(), 'CapNhatCauHinh', 'CauHinhHeThong', @TenCauHinh,  N'Cập nhật: ' + CAST(@GiaTriMoi AS NVARCHAR(20)));  END  ELSE  BEGIN  INSERT INTO [dbo].[CauHinhHeThong] (TenCauHinh, GiaTri, MoTa)  VALUES (@TenCauHinh, @GiaTriMoi, @MoTa);    INSERT INTO [dbo].[AuditLog] (UserName, Action, EntityName, EntityID, Details)  VALUES (ORIGINAL\_LOGIN(), 'ThemCauHinh', 'CauHinhHeThong', @TenCauHinh,  N'Thêm mới: ' + CAST(@GiaTriMoi AS NVARCHAR(20)));  END    COMMIT TRAN;    SELECT N'Thành công! Đã cập nhật ' + @TenCauHinh + N' = ' + CAST(@GiaTriMoi AS NVARCHAR(20)) + N' VND' AS ThongBao;    END TRY  BEGIN CATCH  IF XACT\_STATE() <> 0 ROLLBACK TRAN;  THROW;  END CATCH  END; |

C#:

|  |
| --- |
| public void UpdateCauHinh(string tenCauHinh, decimal giaTri, string moTa)  {  using (var conn = new SqlConnection(\_connectionString))  using (var cmd = new SqlCommand("sp\_CapNhatCauHinhHeThong", conn))  {  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@TenCauHinh", tenCauHinh);  cmd.Parameters.AddWithValue("@GiaTriMoi", giaTri);  cmd.Parameters.AddWithValue("@MoTa", moTa);  conn.Open();  cmd.ExecuteNonQuery();  }  } |

# 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\_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

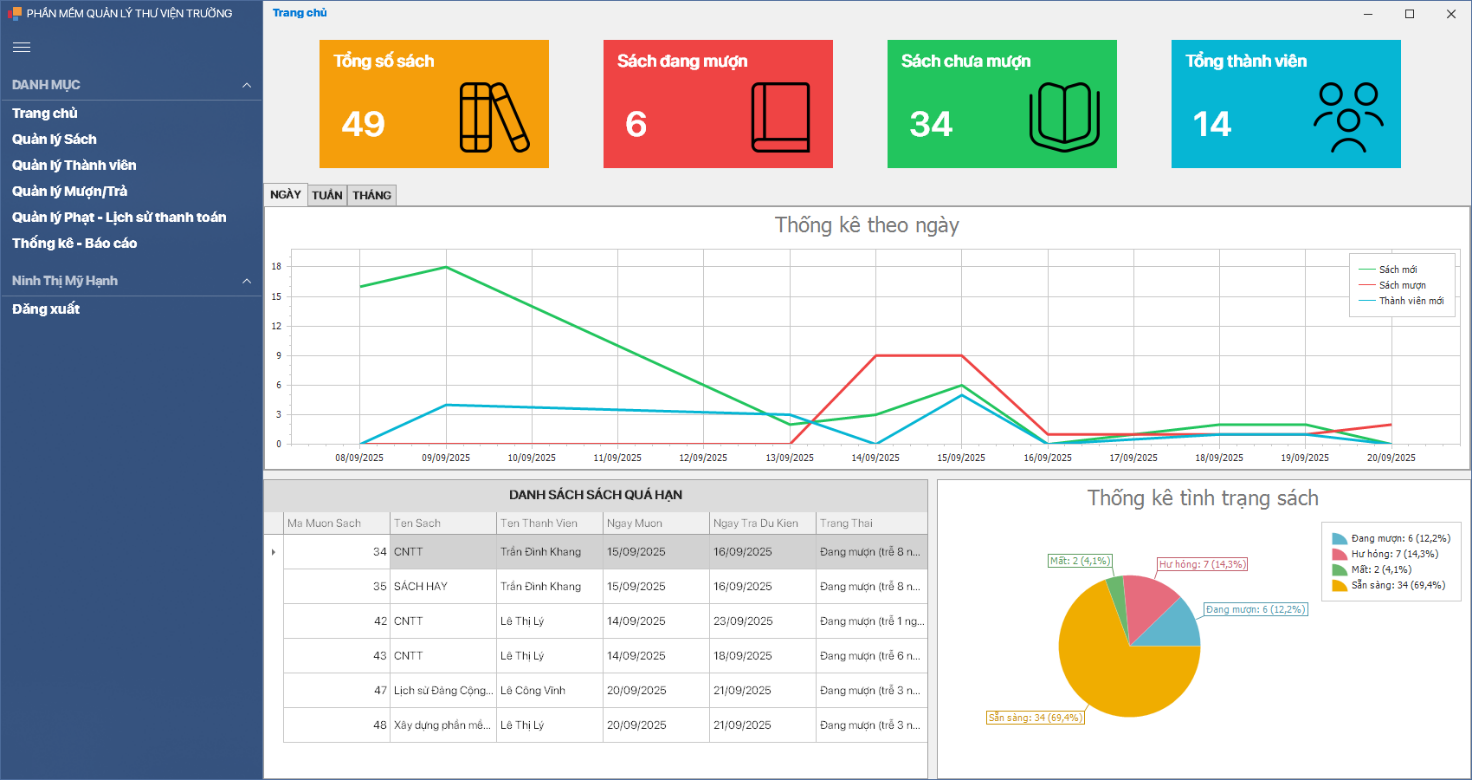
# 2. Giao diện của chương trình:

## 2.1. Form đăng nhập

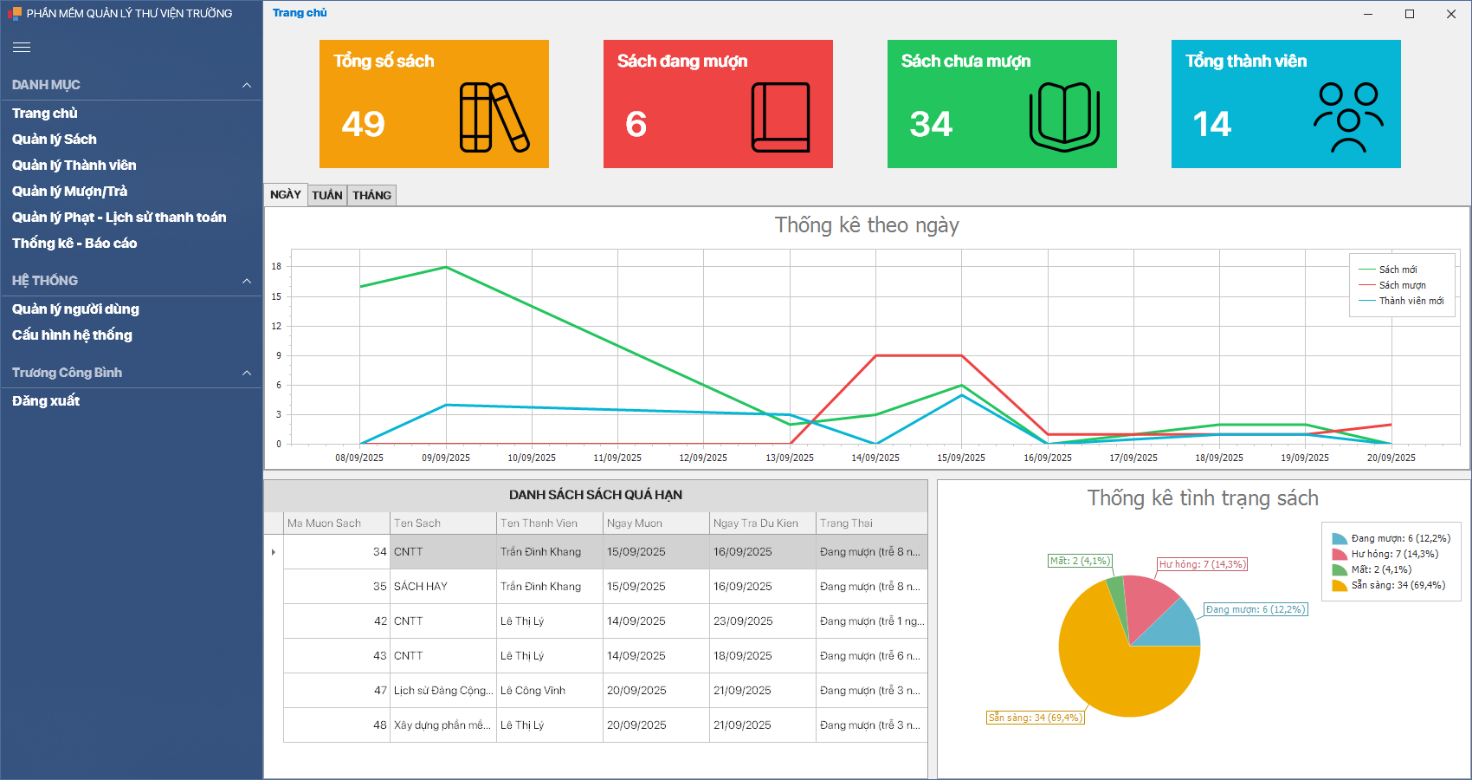


## 2.2. Form chính – Danh mục Trang chủ

- Đối với Thủ thư

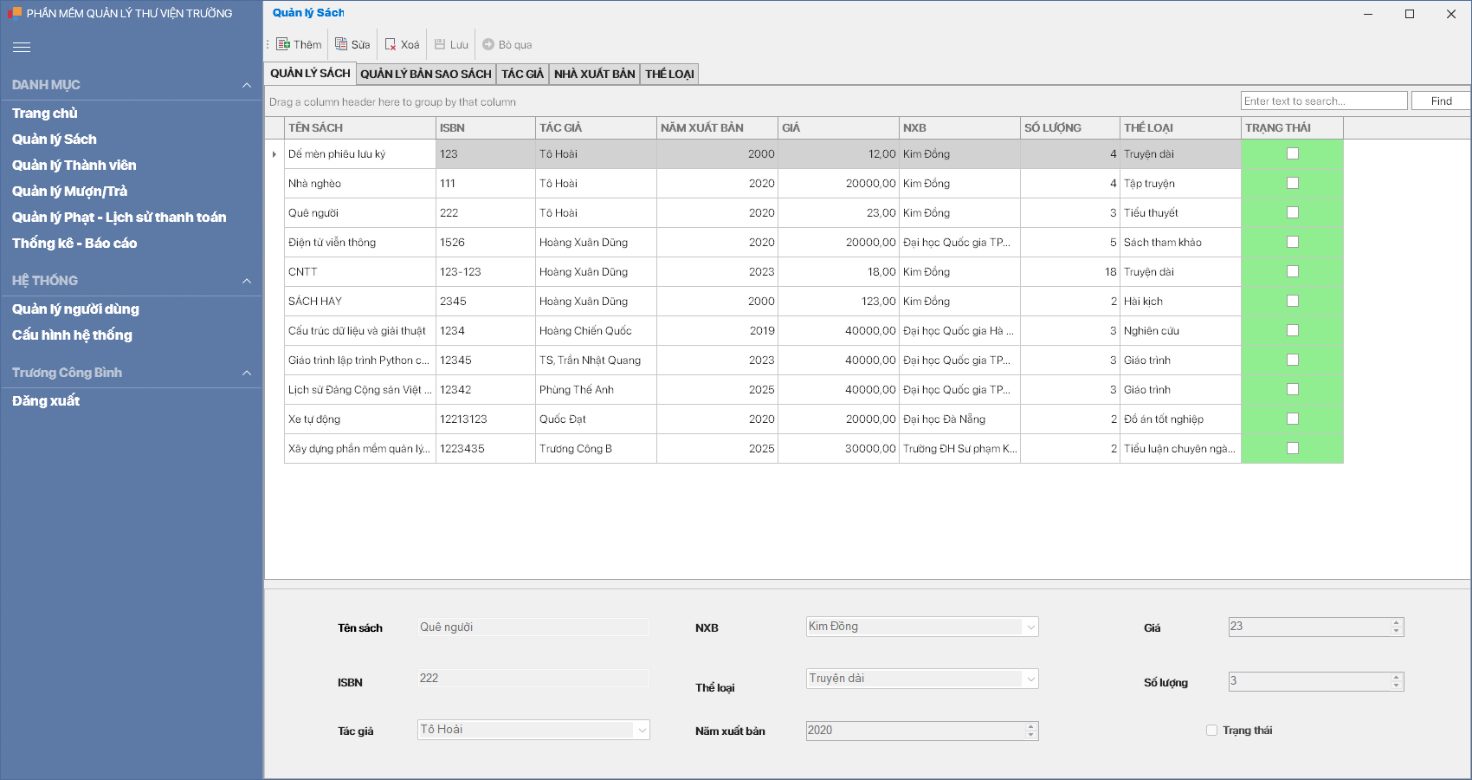


- Đối với Quản trị viên

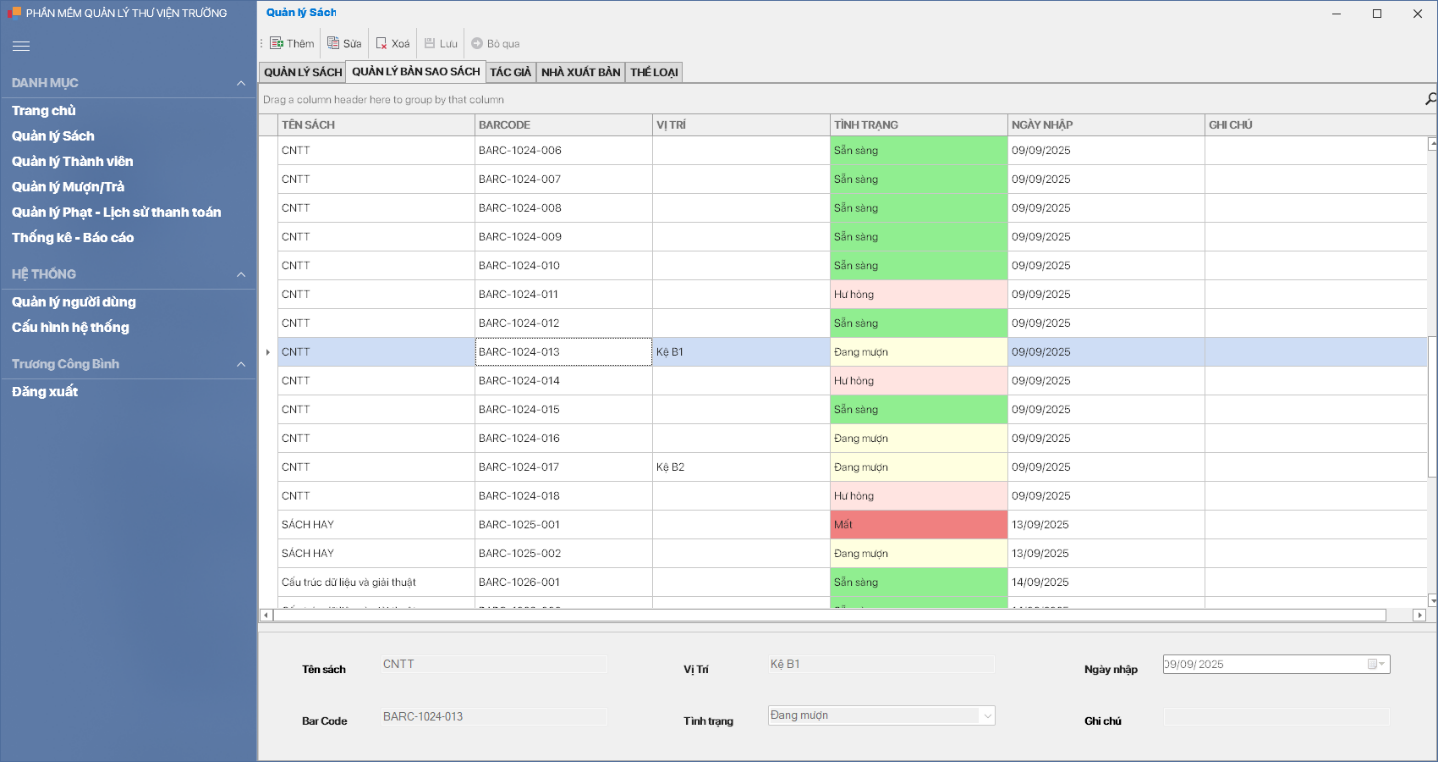


## 2.3. Danh mục Quản lý Sách

### 2.3.1 Tab Sách



### 2.3.2 Tab Bản sao sách



### 2.3.3 Tab Tác giả

A screenshot of a computer

AI-generated content may be incorrect.

### 2.3.4 Tab Nhà xuất bản

A screenshot of a computer

AI-generated content may be incorrect.

### 2.3.5 Tab Thể loại

A screenshot of a computer

AI-generated content may be incorrect.

## 2.5 Danh mục Quản lý Thành viên

A screenshot of a computer

AI-generated content may be incorrect.

## 2.6 Danh mục Quản lý Mượn/Trả

### 2.6.1 Tab Danh sách mượn

A screenshot of a computer

AI-generated content may be incorrect.

### 2.6.2 Tab Chi tiết

A screenshot of a computer

AI-generated content may be incorrect.

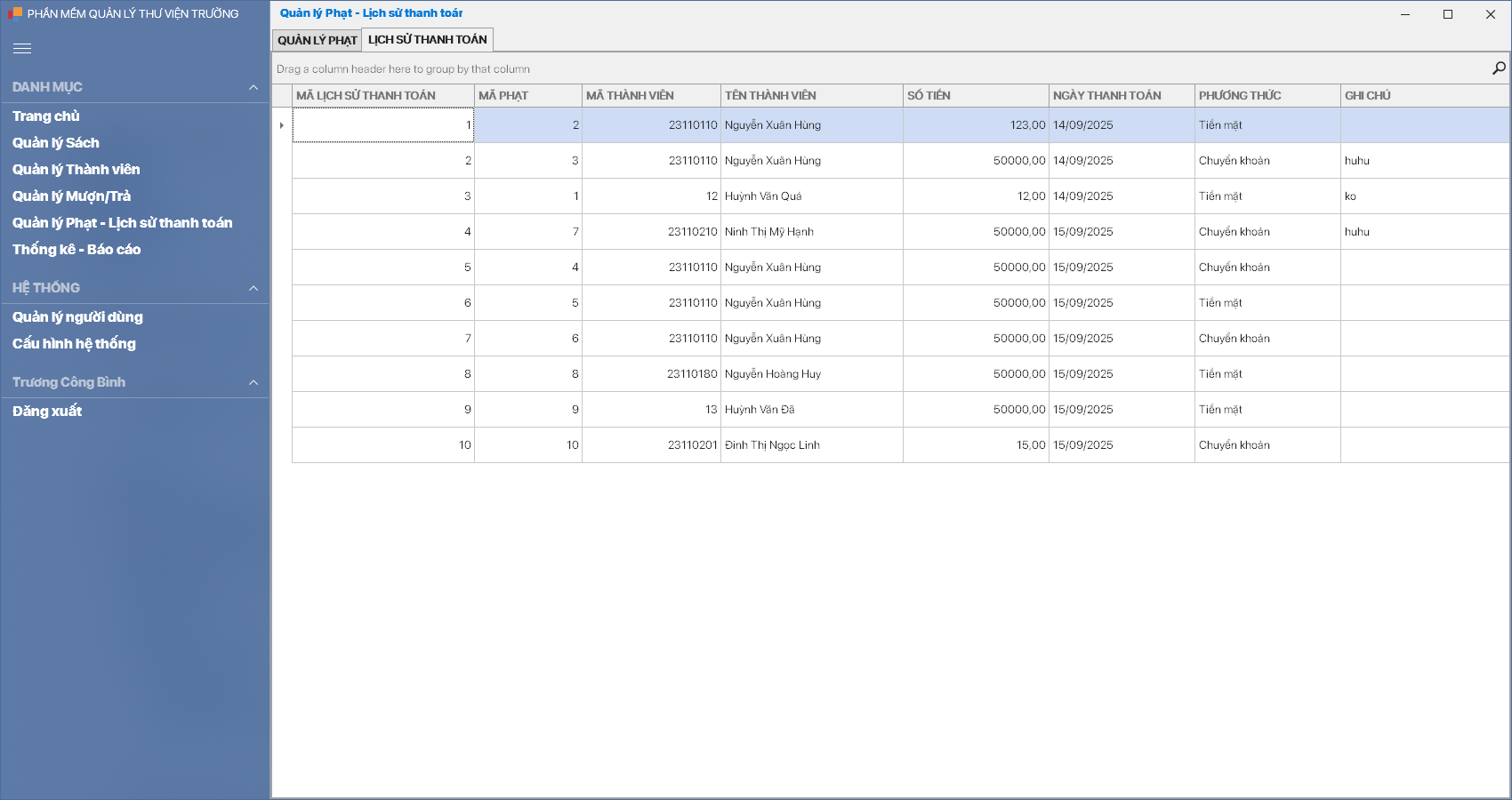
## Danh mục Quản lý Phạt – Lịch sử thanh toán

### 2.7.1 Tab Quản lý Phạt

A screenshot of a computer

AI-generated content may be incorrect.

### 2.7.2 Tab Lịch sử thanh toán



## 2.8 Danh mục Thống kê – Báo cáo

### 2.8.1 Top sách được mượn

A screenshot of a computer

AI-generated content may be incorrect.

### 2.8.2 Tổng tiền phạt theo tuần/tháng/quý/năm

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

### 2.8.3 Thống kê tình trạng sách

A screenshot of a computer

AI-generated content may be incorrect.

### 2.8.4 Thống kê sách mượn quá hạn

A screenshot of a computer

AI-generated content may be incorrect.

### 2.8.5 Thống kê sách mượn theo loại thành viên (Sinh viên/Giảng viên)

A screenshot of a graph

AI-generated content may be incorrect.

## 2.9 Danh mục Quản lý người dùng

A screenshot of a computer

AI-generated content may be incorrect.

## 2.10 Danh mục Cấu hình hệ thống

A screenshot of a computer

AI-generated content may be incorrect.

# CHƯƠNG 6: KẾT LUẬN

Dự án “Xây dựng hệ thống Quản lý thư viện” là một hành trình học tập đầy nỗ lực trong lĩnh vực hệ quản trị cơ sở dữ liệu (DBMS). Trong suốt quá trình thiết kế và phát triển, em đã vận dụng các kiến thức lý thuyết vào thực tiễn, từ đó hiểu rõ hơn về quy trình xây dựng một hệ thống quản lý dữ liệu hoàn chỉnh.

Trong lĩnh vực quản lý cơ sở dữ liệu, việc lựa chọn công nghệ phù hợp đóng vai trò quan trọng. Em đã sử dụng **C#** cho phần lập trình ứng dụng và **SQL Server** cho phần cơ sở dữ liệu. Sự kết hợp này không chỉ giúp tạo ra một hệ thống mạnh mẽ, hiệu quả mà còn thuận tiện để phát triển ứng dụng theo mô hình **3 lớp (UI, BLL, DAL),** đảm bảo mã nguồn rõ ràng, dễ bảo trì và dễ mở rộng.

Việc triển khai cơ sở dữ liệu trên SQL Server đã giúp em tiếp cận và hiểu sâu hơn về cách tổ chức dữ liệu trong một hệ thống quản lý thư viện hiện đại. SQL Server mang lại khả năng lưu trữ, truy xuất và xử lý dữ liệu nhanh chóng – yếu tố then chốt trong việc phục vụ độc giả, giúp họ tra cứu, mượn trả sách một cách tiện lợi. Em đã xây dựng nhiều truy vấn SQL tối ưu, cùng với việc triển khai **Stored Procedures, Functions, Triggers và Views** để tự động hóa, bảo mật và nâng cao hiệu năng hệ thống. Các thủ tục lưu trữ giúp đóng gói logic nghiệp vụ, hàm mang đến sự linh hoạt trong xử lý dữ liệu, trigger đảm bảo tính toàn vẹn dữ liệu khi có thay đổi, còn view hỗ trợ tổng hợp thông tin phục vụ báo cáo và thống kê.

Bên cạnh đó, dự án còn mở ra cơ hội để em tìm hiểu thêm về đặc thù trong công tác quản lý thư viện, bao gồm hoạt động mượn – trả sách, xử lý vi phạm quá hạn, quản lý thành viên và thống kê báo cáo. Việc nghiên cứu nhu cầu sử dụng thư viện của sinh viên và giảng viên giúp chúng em định hướng xây dựng một sản phẩm đáp ứng đúng thực tiễn quản lý trong nhà trường.

Dự án “Xây dựng hệ thống Quản lý thư viện” không chỉ đem đến cho em một hệ thống quản lý dữ liệu hoàn chỉnh mà còn là một trải nghiệm học tập bổ ích, giúp em rèn luyện kỹ năng từ thiết kế giao diện, lập trình ứng dụng đến quản trị và bảo mật dữ liệu. Đây sẽ là nền tảng quan trọng để em tự tin hơn trong sự nghiệp tương lai, đồng thời mở ra nhiều cơ hội tiếp tục khám phá và phát triển trong lĩnh vực công nghệ thông tin, đặc biệt là trong quản lý và khai thác hệ quản trị cơ sở dữ liệu.