

**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).

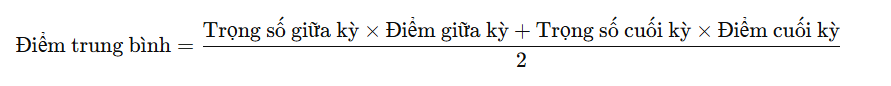
*Danh mục thiết lập trọng số:* thiết lập trọng số phần trăm điểm giữa kỳ, cuối kỳ.

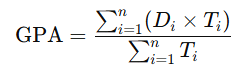
*Danh mục quản lý người dùng:* quản lý tài khoản có thể thêm xoá sửa tài khoản người dùng. Cập nhập trạng thái khoá tài khoản.

*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:

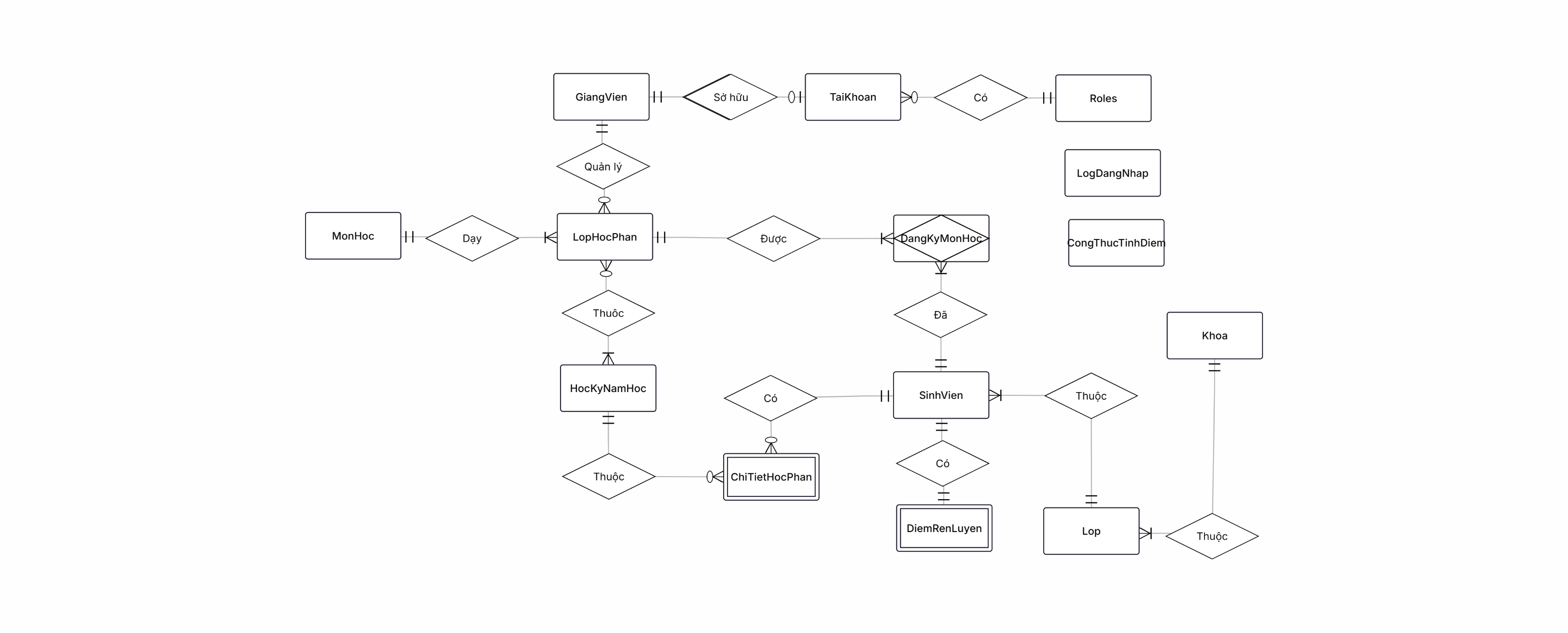
* Di​: điểm của học phần thứ ii (tính theo hệ 10 hoặc hệ 4).
* Ti: số tín chỉ của học phần thứ ii.
* n: tổng số học phần đã tích lũy (Môn học đạt).

Quy đổi điểm

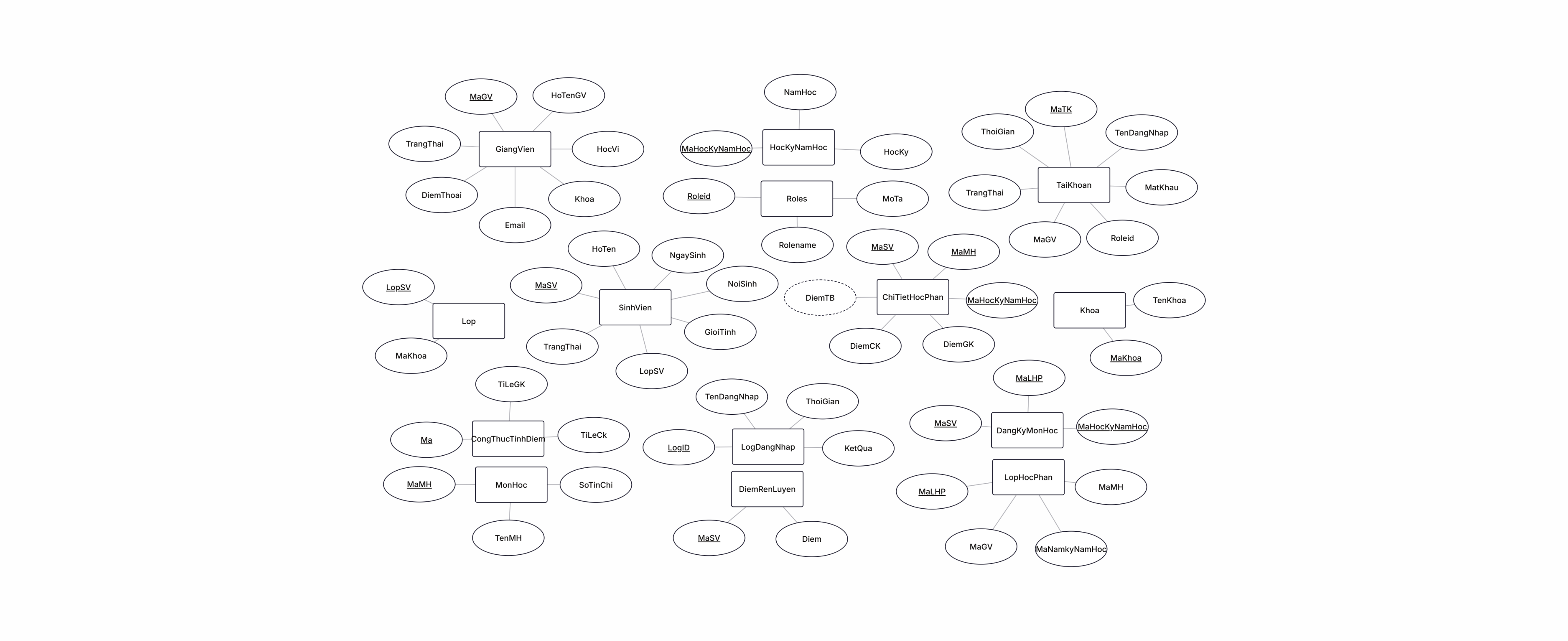
|  |  |  |
| --- | --- | --- |
| **Hệ 10 (thang 10)** | **Hệ 4 (thang 4)** | **Hệ chữ (Letter grade)** |
| Từ 9.0 – 10 | 4.0 | A+ |
| Từ 8.5 – 9.0 | 3.7 | A |
| Từ 8.0 – 8.5 | 3.5 | B+ |
| Từ 7.0 – 8.0 | 3.0 | B |
| Từ 6.5 – 7.0 | 2.5 | C+ |
| Từ 5.5 – 6.5 | 2.0 | C |
| Từ 5.0 – 5.5 | 1.5 | D+ |
| Từ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

 **TheLoai**(**MaTheLoai**, TenTheLoai, MoTa)

 **TacGia**(**MaTacGia**, TenTacGia, QuocTich, NamSinh)

 **NhaXuatBan**(**MaNhaXuatBan**, TenNhaXuatBan, DiaChi, SoDienThoai)

 **Sach**(**MaSach**, TenSach, ISBN, MaTacGia → TacGia, MaNhaXuatBan → NhaXuatBan, MaTheLoai → TheLoai, NamXuatBan, Gia, SoLuong, TrangThai, RowVersion)

 **BanSaoSach**(**MaBanSao**, MaSach → Sach, Barcode, ViTri, TinhTrang, NgayNhap, GhiChu, RowVersion)

 **ThanhVien**(**MaThanhVien**, TenThanhVien, Email, SoDienThoai, DiaChi, LoaiThanhVien, NgayDangKy, RowVersion)

 **Users**(**UserID**, UserName, FullName, PasswordHash, Email, RoleID → Roles, IsActive, CreatedAt)

 **Roles**(**RoleID**, RoleName, Description)

 **PhieuMuon**(**MaPhieuMuon**, MaThanhVien → ThanhVien, UserID → Users, NgayMuon, NgayTraDuKien, TrangThai, GhiChu, RowVersion)

 **ChiTietPhieuMuon**(**MaChiTiet**, MaPhieuMuon → PhieuMuon, MaSach → Sach, MaBanSao → BanSaoSach, NgayTraDuKien, NgayTraThucTe, TrangThai, GhiChu, RowVersion)

 **Phat**(**MaPhat**, MaMuonSach → ChiTietPhieuMuon, SoTien, LyDo, NgayPhat, TrangThai)

 **PaymentHistory**(**MaPayment**, MaPhat → Phat, MaThanhVien → ThanhVien, Amount, PaymentDate, Method, Note)

 **CauHinhHeThong**(**MaCauHinh**, TenCauHinh, GiaTri, MoTa, NgayCapNhat)

 **AuditLog**(**LogID**, UserName, Action, EntityName, EntityID, Details, CreatedAt)

 **TaskScheduler**(**TaskID**, TaskName, LastRun, CreatedAt)

## 3. Các ràng buộc

|  |  |  |
| --- | --- | --- |
| **STT** | **BẢNG** | **RÀNG BUỘC** |
| **1** | **Sach** | [MaSach] [int] PRIMARY KEY  [TenSach] [nvarchar](200) NOT NULL,  [ISBN] [nvarchar](13) NOT NULL,  [MaTacGia] [int] FOREIGN KEY TacGia,  [MaNhaXuatBan] [int] FOREIGN KEY NhaXuatBan,  [MaTheLoai] [int] FOREIGN KEY TheLoai,  [NamXuatBan] [int] NOT NULL,  [Gia] [decimal](10, 2) NOT NULL,  [SoLuong] [int] NOT NULL,  [TrangThai] [bit] NOT NULL  CHECK(Gia > 0)  CHECK(([NamXuatBan]>(1900) AND [NamXuatBan]<=datepart(year,getdate())  CHECK(SoLuong > 0) |
| **2** | **NhaXuatBan** | [MaNhaXuatBan] [int] PRIMARY KEY,  [TenNhaXuatBan] [nvarchar](100) NOT NULL |
| **3** | **TheLoai** | [MaTheLoai] [int] PRIMARY KEY,  [TenTheLoai] [nvarchar](50) NOT NULL |
| **4** | **TacGia** | [MaTacGia] [int] PRIMARY KEY,  [TenTacGia] [nvarchar](100) NOT NULL,  CHECK(NamSinh > 1800 and NamSinh < datepart(year,getdate()) |
| **5** | **BanSaoSach** | [MaBanSao] [int] PRIMARY KEY,  [MaSach] [int] FOREIGN KEY Sach  [Barcode] [nvarchar](100) NOT NULL,  [TinhTrang] [nvarchar](50) NOT NULL,  [NgayNhap] [date] NOT NULL |
| **6** | **PhieuMuon** | [MaPhieuMuon] [int] PRIMARY KEY,  [MaThanhVien] [int] FOREIGN KEY ThanhVien,  [UserID] [int] FOREIGN KEY Users,  [NgayMuon] [datetime] NOT NULL,  [TrangThai] [nvarchar](50) NOT NULL, |
| **7** | **PhieuMuonChiTiet** | [MaChiTiet] [int] PRIMARY KEY,  [MaPhieuMuon] [int] FOREIGN KEY PhieuMuon,  [MaSach] [int] FOREIGN KEY Sach,  [MaBanSao] [int] FOREIGN KEY BanSaoSach,  [TrangThai] [nvarchar](50) NOT NULL, |
| **8** | **ThanhVien** | [MaThanhVien] [int] PRIMARY KEY,  [TenThanhVien] [nvarchar](100) NOT NULL,  [Email] [nvarchar](100) NOT NULL,  [LoaiThanhVien] [nvarchar](20) NOT NULL,  [NgayDangKy] [date] NOT NULL,  CHECK(Email like '%@%.%') |
| **9** | **Phat** | [MaPhat] [int] PRIMARY KEY,  [MaMuonSach] [int] FOREIGN KEY ChiTietPhieuMuon,  [SoTien] [decimal](10, 2) NOT NULL,  [LyDo] [nvarchar](200) NOT NULL,  [NgayPhat] [date] NOT NULL,  [TrangThai] [nvarchar](20) NOT NULL,  CHECK(SoTien > 0) |
| **10** | **PaymentHistory** | [MaPayment] [int] PRIMARY KEY,  [MaPhat] [int] FOREIGN KEY Phat,  [MaThanhVien] [int] FOREIGN KEY ThanhVien,  [Amount] [decimal](18, 2) NOT NULL,  [PaymentDate] [datetime] NOT NULL, |
| **11** | **Users** | [UserID] [int] PRIMARY KEY,  [UserName] [nvarchar](100) NOT NULL,  [PasswordHash] [nvarchar](300) NOT NULL,  [RoleID] [int] FOREIGN KEY Roles,  [IsActive] [bit] NOT NULL, |
| **12** | **Roles** | [RoleID] [int] PRIMARY KEY,  [RoleName] [nvarchar](100) NOT NULL, |
| **13** | **CauHinhHeThong** | [MaCauHinh] [int] PRIMARY KEY,  [TenCauHinh] [nvarchar](100) NOT NULL,  [GiaTri] [decimal](10, 2) NOT NULL, |
| **14** | **AuditLog** | [LogID] [bigint] PRIMARY KEY,  [Action] [nvarchar](200) NOT NULL,  [CreatedAt] [datetime] NOT NULL, |
| **15** | **TaskScheduler** | [TaskID] [int] PRIMARY KEY |

## 4. Cài đặt CSDL và các ràng buộc

***Bảng Sách***

|  |
| --- |
| CREATE TABLE Sach (  MaSach INT IDENTITY(1,1) PRIMARY KEY,  TenSach NVARCHAR(200) NOT NULL,  ISBN NVARCHAR(13) NOT NULL UNIQUE,  MaTacGia INT NOT NULL,  MaNhaXuatBan INT NOT NULL,  MaTheLoai INT NOT NULL,  NamXuatBan INT NOT NULL CHECK (NamXuatBan > 1900 AND NamXuatBan <= YEAR(GETDATE())),  Gia DECIMAL(10,2) NOT NULL CHECK (Gia > 0),  SoLuong INT NOT NULL DEFAULT 1 CHECK (SoLuong >= 0),  TrangThai BIT NOT NULL DEFAULT 1,  RowVersion ROWVERSION,  CONSTRAINT FK\_Sach\_TacGia FOREIGN KEY (MaTacGia) REFERENCES TacGia(MaTacGia),  CONSTRAINT FK\_Sach\_NhaXuatBan FOREIGN KEY (MaNhaXuatBan) REFERENCES NhaXuatBan(MaNhaXuatBan),  CONSTRAINT FK\_Sach\_TheLoai FOREIGN KEY (MaTheLoai) REFERENCES TheLoai(MaTheLoai)  ); |

***Bảng Nhà xuất bản***

|  |
| --- |
| CREATE TABLE NhaXuatBan (  MaNhaXuatBan INT IDENTITY(1,1) PRIMARY KEY,  TenNhaXuatBan NVARCHAR(100) NOT NULL,  DiaChi NVARCHAR(200) NULL,  SoDienThoai NVARCHAR(20) NULL  ); |

***Bảng Thể loại***

|  |
| --- |
| CREATE TABLE TheLoai (  MaTheLoai INT IDENTITY(1,1) PRIMARY KEY,  TenTheLoai NVARCHAR(50) NOT NULL,  MoTa NVARCHAR(200) NULL  ); |

***Bảng Tác giả***

|  |
| --- |
| CREATE TABLE TacGia (  MaTacGia INT IDENTITY(1,1) PRIMARY KEY,  TenTacGia NVARCHAR(100) NOT NULL,  QuocTich NVARCHAR(50) NULL,  NamSinh INT NULL CHECK (NamSinh > 1800 AND NamSinh < YEAR(GETDATE()))  ); |

***Bảng Bản sao sách***

|  |
| --- |
| CREATE TABLE BanSaoSach (  MaBanSao INT IDENTITY(1,1) PRIMARY KEY,  MaSach INT NOT NULL,  Barcode NVARCHAR(100) NOT NULL UNIQUE,  ViTri NVARCHAR(100) NULL,  TinhTrang NVARCHAR(50) NOT NULL DEFAULT N'Sẵn sàng',  NgayNhap DATE NOT NULL DEFAULT GETDATE(),  GhiChu NVARCHAR(500) NULL,  RowVersion ROWVERSION,  CONSTRAINT FK\_BanSaoSach\_Sach FOREIGN KEY (MaSach) REFERENCES Sach(MaSach)); |

***Bảng Phiếu mượn***

|  |
| --- |
| CREATE TABLE PhieuMuon (  MaPhieuMuon INT IDENTITY(1,1) PRIMARY KEY,  MaThanhVien INT NOT NULL,  UserID INT NOT NULL,  NgayMuon DATETIME NOT NULL,  NgayTraDuKien DATETIME NULL,  TrangThai NVARCHAR(50) NOT NULL,  GhiChu NVARCHAR(400) NULL,  RowVersion ROWVERSION,  CONSTRAINT FK\_PhieuMuon\_ThanhVien FOREIGN KEY (MaThanhVien) REFERENCES ThanhVien(MaThanhVien),  CONSTRAINT FK\_PhieuMuon\_Users FOREIGN KEY (UserID) REFERENCES Users(UserID)  ); |

***Bảng Phiếu mượn chi tiết***

|  |
| --- |
| CREATE TABLE ChiTietPhieuMuon (  MaChiTiet INT IDENTITY(1,1) PRIMARY KEY,  MaPhieuMuon INT NOT NULL,  MaSach INT NOT NULL,  MaBanSao INT NULL,  NgayTraDuKien DATETIME NULL,  NgayTraThucTe DATETIME NULL,  TrangThai NVARCHAR(50) NOT NULL DEFAULT N'Đang mượn',  GhiChu NVARCHAR(400) NULL,  RowVersion ROWVERSION,  CONSTRAINT FK\_ChiTietPhieuMuon\_PhieuMuon FOREIGN KEY (MaPhieuMuon) REFERENCES PhieuMuon(MaPhieuMuon),  CONSTRAINT FK\_ChiTietPhieuMuon\_Sach FOREIGN KEY (MaSach) REFERENCES Sach(MaSach),  CONSTRAINT FK\_ChiTietPhieuMuon\_BanSao FOREIGN KEY (MaBanSao) REFERENCES BanSaoSach(MaBanSao)  ); |

***Bảng Thành viên***

|  |
| --- |
| CREATE TABLE ThanhVien (  MaThanhVien INT IDENTITY(1,1) PRIMARY KEY,  TenThanhVien NVARCHAR(100) NOT NULL,  Email NVARCHAR(100) NOT NULL UNIQUE,  SoDienThoai NVARCHAR(20) NULL,  DiaChi NVARCHAR(200) NULL,  LoaiThanhVien NVARCHAR(20) NOT NULL,  NgayDangKy DATE NOT NULL DEFAULT GETDATE(),  RowVersion ROWVERSION,  CONSTRAINT CHK\_Email\_ThanhVien CHECK (Email LIKE '%@%.%')  ); |

***Bảng*** ***Phạt***

|  |
| --- |
| CREATE TABLE Phat (  MaPhat INT IDENTITY(1,1) PRIMARY KEY,  MaMuonSach INT NOT NULL,  SoTien DECIMAL(10,2) NOT NULL CHECK (SoTien >= 0),  LyDo NVARCHAR(200) NOT NULL,  NgayPhat DATETIME NOT NULL DEFAULT GETDATE(),  TrangThai NVARCHAR(20) NOT NULL DEFAULT N'ChuaThanhToan',  CONSTRAINT FK\_Phat\_ChiTiet FOREIGN KEY (MaMuonSach) REFERENCES ChiTietPhieuMuon(MaChiTiet)); |

***Bảng Lịch sử thanh toán (PaymentHistory)***

|  |
| --- |
| CREATE TABLE PaymentHistory (  MaPayment INT IDENTITY(1,1) PRIMARY KEY,  MaPhat INT NULL,  MaThanhVien INT NULL,  Amount DECIMAL(18,2) NOT NULL,  PaymentDate DATETIME NOT NULL DEFAULT GETDATE(),  Method NVARCHAR(50) NULL,  Note NVARCHAR(500) NULL,  CONSTRAINT FK\_PaymentHistory\_Phat FOREIGN KEY (MaPhat) REFERENCES Phat(MaPhat),  CONSTRAINT FK\_PaymentHistory\_ThanhVien FOREIGN KEY (MaThanhVien) REFERENCES ThanhVien(MaThanhVien)  ); |

***Bảng Người dùng (Users)***

|  |
| --- |
| CREATE TABLE Users (  UserID INT IDENTITY(1,1) PRIMARY KEY,  UserName NVARCHAR(100) NOT NULL UNIQUE,  FullName NVARCHAR(200) NULL,  PasswordHash NVARCHAR(300) NOT NULL,  Email NVARCHAR(200) NULL,  RoleID INT NULL,  IsActive BIT NOT NULL DEFAULT 1,  CreatedAt DATETIME NOT NULL DEFAULT GETDATE(),  CONSTRAINT FK\_Users\_Roles FOREIGN KEY (RoleID) REFERENCES Roles(RoleID)  ); |

***Bảng Vai trò (Roles)***

|  |
| --- |
| CREATE TABLE Roles (  RoleID INT IDENTITY(1,1) PRIMARY KEY,  RoleName NVARCHAR(100) NOT NULL UNIQUE,  Description NVARCHAR(300) NULL  ); |

***Bảng Cấu hình hệ thống***

|  |
| --- |
| CREATE TABLE CauHinhHeThong (  MaCauHinh INT IDENTITY(1,1) PRIMARY KEY,  TenCauHinh NVARCHAR(100) NOT NULL UNIQUE,  GiaTri DECIMAL(10,2) NOT NULL,  MoTa NVARCHAR(200) NULL,  NgayCapNhat DATETIME NULL DEFAULT GETDATE()  ); |

***Bảng Log/Nhật ký hệ thống (AuditLog)***

|  |
| --- |
| CREATE TABLE AuditLog (  LogID BIGINT IDENTITY(1,1) PRIMARY KEY,  UserName NVARCHAR(100) NULL,  Action NVARCHAR(200) NOT NULL,  EntityName NVARCHAR(100) NULL,  EntityID NVARCHAR(100) NULL,  Details NVARCHAR(MAX) NULL,  CreatedAt DATETIME NOT NULL DEFAULT GETDATE()  ); |

***Bảng Lên lịch (TaskScheduler)***

|  |
| --- |
| CREATE TABLE TaskScheduler (  TaskID INT IDENTITY(1,1) PRIMARY KEY,  TaskName NVARCHAR(100) NULL,  LastRun DATETIME NULL,  CreatedAt DATETIME NOT NULL DEFAULT GETDATE()  ); |

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

**A screenshot of a computer screen

AI-generated content may be incorrect.**

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

## 

## 6. Các triggers

**a. Trigger tự động tạo phạt khi có sách mượn quá hạn**

|  |
| --- |
| CREATE TRIGGER dbo.TRG\_CTPM\_TaoPhatQuaHan  ON dbo.ChiTietPhieuMuon  AFTER UPDATE  AS  BEGIN  SET NOCOUNT ON;  IF UPDATE(NgayTraThucTe)  BEGIN  DECLARE @PhiPhatNgay DECIMAL(10,2);  SELECT @PhiPhatNgay = cht.GiaTri  FROM dbo.CauHinhHeThong cht  WHERE cht.TenCauHinh = N'PhiPhatTreHan';  SET @PhiPhatNgay = ISNULL(@PhiPhatNgay, 5000.00);  INSERT INTO dbo.Phat (MaMuonSach, SoTien, LyDo, NgayPhat, TrangThai)  SELECT i.MaChiTiet,  CASE WHEN DATEDIFF(DAY, i.NgayTraDuKien, i.NgayTraThucTe) > 0  THEN DATEDIFF(DAY, i.NgayTraDuKien, i.NgayTraThucTe) \* @PhiPhatNgay  ELSE 0 END AS SoTien,  CASE WHEN DATEDIFF(DAY, i.NgayTraDuKien, i.NgayTraThucTe) > 0  THEN N'Quá hạn ' + CAST(DATEDIFF(DAY, i.NgayTraDuKien, i.NgayTraThucTe) AS NVARCHAR) + N' ngày'  ELSE N'' END AS LyDo,  GETDATE(),  CASE WHEN DATEDIFF(DAY, i.NgayTraDuKien, i.NgayTraThucTe) > 0 THEN N'ChuaThanhToan' ELSE N'Da thanh toan' END  FROM inserted i  WHERE i.NgayTraThucTe IS NOT NULL  AND DATEDIFF(DAY, i.NgayTraDuKien, i.NgayTraThucTe) > 0  AND NOT EXISTS (SELECT 1 FROM dbo.Phat p WHERE p.MaMuonSach = i.MaChiTiet);  END  END; |

**b. Trigger cập nhật trạng thái của bản sao sách trong chi tiết phiếu mượn khi có sách mượn/trả**

|  |
| --- |
| CREATE TRIGGER dbo.TRG\_CTPM\_UpdateBanSaoStatus  ON dbo.ChiTietPhieuMuon  AFTER INSERT, UPDATE  AS  BEGIN  SET NOCOUNT ON;  UPDATE bs  SET bs.TinhTrang = N'Đang mượn'  FROM dbo.BanSaoSach bs  INNER JOIN inserted i ON bs.MaBanSao = i.MaBanSao  WHERE i.TrangThai = N'Đang mượn';  UPDATE bs2  SET bs2.TinhTrang = N'Sẵn sàng'  FROM dbo.BanSaoSach bs2  INNER JOIN inserted i2 ON bs2.MaBanSao = i2.MaBanSao  WHERE i2.NgayTraThucTe IS NOT NULL;  END; |

**c. Trigger tự động thêm bản sao khi số lượng sách được nhập**

|  |
| --- |
| CREATE TRIGGER dbo.TRG\_AutoInsertBanSaoSach  ON dbo.Sach  AFTER INSERT  AS  BEGIN  SET NOCOUNT ON;  DECLARE @MaSach INT, @SoLuong INT, @i INT;  DECLARE cur CURSOR FOR SELECT MaSach, SoLuong FROM inserted;  OPEN cur;  FETCH NEXT FROM cur INTO @MaSach, @SoLuong;  WHILE @@FETCH\_STATUS = 0  BEGIN  SET @i = 1;  WHILE @i <= @SoLuong  BEGIN  INSERT INTO dbo.BanSaoSach (MaSach, Barcode, ViTri, TinhTrang, NgayNhap, GhiChu)  VALUES (  @MaSach,  'BARC-' + CAST(@MaSach AS NVARCHAR(20)) + '-' + RIGHT('000' + CAST(@i AS NVARCHAR(10)), 3),  NULL,  N'Sẵn sàng',  GETDATE(),  NULL  );  SET @i += 1;  END  FETCH NEXT FROM cur INTO @MaSach, @SoLuong;  END  CLOSE cur;  DEALLOCATE cur;  END; |

**d. Trigger kiểm soát số lượng sách**

|  |
| --- |
| CREATE TRIGGER dbo.TRG\_HandleSoLuongSach  ON dbo.Sach  AFTER UPDATE  AS  BEGIN  SET NOCOUNT ON;  IF UPDATE(SoLuong)  BEGIN  DECLARE @MaSach INT, @OldQty INT, @NewQty INT;  DECLARE cur CURSOR FOR  SELECT i.MaSach, d.SoLuong AS OldQty, i.SoLuong AS NewQty  FROM inserted i  JOIN deleted d ON i.MaSach = d.MaSach;  OPEN cur;  FETCH NEXT FROM cur INTO @MaSach, @OldQty, @NewQty;  WHILE @@FETCH\_STATUS = 0  BEGIN  IF @NewQty < @OldQty  BEGIN  ROLLBACK TRAN;  THROW 60001, N'Khong duoc giam SoLuong. Vui long cap nhat TinhTrang cua BanSaoSach (Mất/Hỏng/Thanh lý)', 1;  END  IF @NewQty > @OldQty  BEGIN  DECLARE @i INT = 1, @NeedAdd INT = @NewQty - @OldQty;  WHILE @i <= @NeedAdd  BEGIN  INSERT INTO dbo.BanSaoSach (MaSach, Barcode, TinhTrang, NgayNhap)  VALUES (@MaSach,  'BARC-' + CAST(@MaSach AS NVARCHAR) + '-' + RIGHT('000' + CAST(@OldQty + @i AS NVARCHAR), 3),  N'Sẵn sàng',  GETDATE());  SET @i = @i + 1;  END  END  FETCH NEXT FROM cur INTO @MaSach, @OldQty, @NewQty;  END  CLOSE cur;  DEALLOCATE cur;  END  END; |

**e. Trigger cập nhật trạng thái ẩn cho bản sao sách khi tình trạng sách là không sẵn sàng**

|  |
| --- |
| CREATE TRIGGER dbo.TRG\_HideBanSaoWhenSachChecked  ON dbo.Sach  AFTER UPDATE  AS  BEGIN  SET NOCOUNT ON;  UPDATE bs  SET bs.TinhTrang = N'Bị ẩn'  FROM dbo.BanSaoSach bs  INNER JOIN inserted i ON bs.MaSach = i.MaSach  WHERE i.TrangThai = 1;  UPDATE bs  SET bs.TinhTrang = N'Sẵn sàng'  FROM dbo.BanSaoSach bs  INNER JOIN inserted i ON bs.MaSach = i.MaSach  WHERE i.TrangThai = 0  AND bs.TinhTrang = N'Bị ẩn';  END; |

## 7. Các views

**a. View hiển thị danh sách sách đang mượn**

|  |
| --- |
| CREATE VIEW dbo.DanhSachSachDangMuon AS  SELECT c.MaChiTiet AS MaMuonSach,  COALESCE(bs.Barcode, CAST(c.MaSach AS NVARCHAR(50))) AS BarcodeOrSachID,  s.TenSach,  tv.TenThanhVien,  p.NgayMuon,  c.NgayTraDuKien,  c.MaBanSao  FROM dbo.ChiTietPhieuMuon c  LEFT JOIN dbo.BanSaoSach bs ON c.MaBanSao = bs.MaBanSao  LEFT JOIN dbo.Sach s ON c.MaSach = s.MaSach  LEFT JOIN dbo.PhieuMuon p ON c.MaPhieuMuon = p.MaPhieuMuon  LEFT JOIN dbo.ThanhVien tv ON p.MaThanhVien = tv.MaThanhVien  WHERE c.NgayTraThucTe IS NULL; |

**b. View hiển thị top sách được mượn nhiều**

|  |
| --- |
| CREATE VIEW dbo.TopSachMuon AS  SELECT s.MaSach, s.TenSach, COUNT(c.MaChiTiet) AS SoLanMuon  FROM dbo.ChiTietPhieuMuon c  INNER JOIN dbo.Sach s ON c.MaSach = s.MaSach  GROUP BY s.MaSach, s.TenSach; |

**c. View hiển thị 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(); |

**d. View báo cáo phạt**

|  |
| --- |
| CREATE VIEW dbo.BaoCaoPhat AS  SELECT ph.MaPhat,  ph.MaMuonSach AS MaChiTiet,  tv.TenThanhVien,  ph.SoTien,  ph.LyDo  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; |

**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 tính số lượng sách đang mượn theo thành viên**

|  |
| --- |
| CREATE FUNCTION dbo.fn\_CountOnLoanByMember(@MaThanhVien INT)  RETURNS INT  AS  BEGIN  DECLARE @cnt INT;  SELECT @cnt = COUNT(\*)  FROM dbo.ChiTietPhieuMuon c  JOIN dbo.PhieuMuon p ON c.MaPhieuMuon = p.MaPhieuMuon  WHERE p.MaThanhVien = @MaThanhVien  AND c.NgayTraThucTe IS NULL; -- still on loan  RETURN ISNULL(@cnt, 0);  END; |

**b. Hàm 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 thống kê số lượng 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; |

**d. Hàm trả về danh sách sách quá hạn theo thành viên**

|  |
| --- |
| CREATE FUNCTION dbo.RTO\_GetOverdueBooksByMember(  @MaThanhVien INT  )  RETURNS TABLE  AS  RETURN  (  SELECT c.MaChiTiet, s.TenSach, c.NgayTraDuKien, DATEDIFF(DAY, c.NgayTraDuKien, GETDATE()) AS SoNgayQuaHan  FROM dbo.ChiTietPhieuMuon c  INNER JOIN dbo.PhieuMuon p ON c.MaPhieuMuon = p.MaPhieuMuon  INNER JOIN dbo.Sach s ON c.MaSach = s.MaSach  WHERE p.MaThanhVien = @MaThanhVien  AND c.NgayTraThucTe IS NULL  AND c.NgayTraDuKien < GETDATE()); |

# CHƯƠNG 3: THIẾT KẾ CÁC CHỨC NĂNG

## Kết nối cơ sở dữ liệu

Chuỗi kết nối được lưu trong App.config để đảm bảo bảo mật. Chuỗi kết nối chính là QuanLyThuVienConnectionString, còn chuỗi QuanLyThuVien\_Integrated để backup.

|  |
| --- |
| <connectionStrings>  <add name="QuanLyThuVien\_Integrated" connectionString="Server=.;Database=QuanLyThuVien;Trusted\_Connection=True;TrustServerCertificate=True;" providerName="System.Data.SqlClient" />  <add name="QuanLyThuVienConnectionString" connectionString="Server=.;Database=QuanLyThuVien;User ID={0};Password={1};Encrypt=false;TrustServerCertificate=true;Connection Timeout=30;"  providerName="System.Data.SqlClient" />  </connectionStrings> |

Lớp cung cấp chuỗi kết nối

|  |
| --- |
| using System;  using System.Configuration;  namespace QuanLyThuVien.BLL.Services  {  public static class ConnectionStringProvider  {  private static string \_currentConnectionString;  public static void SetConnectionString(string connectionString)  {  \_currentConnectionString = connectionString;  }  public static string GetConnectionString()  {  if (!string.IsNullOrEmpty(\_currentConnectionString))  {  return \_currentConnectionString;  }  return GetIntegratedConnectionString() ?? GetDefaultConnectionString();  }  public static string GetIntegratedConnectionString()  {  return ConfigurationManager.ConnectionStrings["QuanLyThuVien\_Integrated"]?.ConnectionString;  }  public static string GetDefaultConnectionString()  {  return ConfigurationManager.ConnectionStrings["QuanLyThuVienConnectionString"]?.ConnectionString;  }  public static void Clear()  {  \_currentConnectionString = null;  } }} |

Lớp tạo context để thực thi các procedure

|  |
| --- |
| 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 DevExpress.XtraEditors;  using QuanLyThuVien.BLL.Services;  using QuanLyThuVien.DAL.Entities;  using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Configuration;  using System.Data;  using System.Data.SqlClient;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  namespace QuanLyThuVien.UI  {  public partial class frmLogin : DevExpress.XtraEditors.XtraForm  {  private NguoiDungService \_nguoiDungService;  private readonly string \_integratedConnString;  private readonly string \_sqlAuthConnStringTemplate;  public class UserSession  {  public static string CurrentUserName { get; set; }  public static string CurrentUserPassword { get; set; }  public static Users CurrentUser { get; set; }  public static string ConnectionString { get; set; }  public static void Clear()  {  CurrentUserName = null;  CurrentUserPassword = null;  CurrentUser = null;  ConnectionString = null;  }  }  public frmLogin()  {  InitializeComponent();  \_integratedConnString = ConfigurationManager.ConnectionStrings["QuanLyThuVien\_Integrated"]?.ConnectionString;  \_sqlAuthConnStringTemplate = ConfigurationManager.ConnectionStrings["QuanLyThuVienConnectionString"]?.ConnectionString;  if (string.IsNullOrEmpty(\_integratedConnString))  {  throw new Exception("Không tìm thấy connection string 'QuanLyThuVien\_Integrated' trong app.config!");  }  }  private void frmLogin\_Load(object sender, EventArgs e)  {  ConnectionStringProvider.SetConnectionString(\_integratedConnString);    var dbContext = ContextFactory.CreateContext();  var userRepository = new GenericRepository<Users>(dbContext);  \_nguoiDungService = new NguoiDungService(userRepository);  this.AcceptButton = btnXacNhan;  this.CancelButton = btnThoat;  txtUsername.Focus();  }  private void btnXacNhan\_Click(object sender, EventArgs e)  {  string username = txtUsername.Text.Trim();  string password = txtPassword.Text.Trim();  if (string.IsNullOrEmpty(username))  {  XtraMessageBox.Show("Tên đăng nhập không được để trống!", "Thông báo",  MessageBoxButtons.OK, MessageBoxIcon.Information);  txtUsername.Focus();  return;  }  if (string.IsNullOrEmpty(password))  {  XtraMessageBox.Show("Mật khẩu không được để trống!", "Thông báo",  MessageBoxButtons.OK, MessageBoxIcon.Information);  txtPassword.Focus();  return;  }  if (!ValidateUser(username, password))  {  return;  }  string sqlAuthConnString = BuildSqlAuthConnectionString(username, password);  if (string.IsNullOrEmpty(sqlAuthConnString))  {  XtraMessageBox.Show("Không thể tạo connection string SQL Authentication!", "Lỗi",  MessageBoxButtons.OK, MessageBoxIcon.Error);  return;  }  if (!TestSqlAuthConnection(sqlAuthConnString))  {  XtraMessageBox.Show("Lỗi kết nối database với tài khoản này! Vui lòng liên hệ admin.", "Lỗi",  MessageBoxButtons.OK, MessageBoxIcon.Error);  return;  }  SaveUserSession(username, password, sqlAuthConnString);  var user = \_nguoiDungService.GetAllUsers().FirstOrDefault(u => u.UserName == username);  frmMain frm = new frmMain(user, sqlAuthConnString);  frm.FormClosed += FrmMain\_FormClosed;  frm.Show();  this.Hide();  }  private bool ValidateUser(string username, string password)  {  try  {  bool userExists = \_nguoiDungService.checkUserExist(username);  if (!userExists)  {  XtraMessageBox.Show("Tên đăng nhập không tồn tại!", "Thông báo",  MessageBoxButtons.OK, MessageBoxIcon.Information);  txtUsername.Focus();  return false;  }  var user = \_nguoiDungService.GetAllUsers().FirstOrDefault(u => u.UserName == username);  if (user == null || !user.PasswordHash.Equals(password))  {  XtraMessageBox.Show("Mật khẩu không đúng!", "Thông báo",  MessageBoxButtons.OK, MessageBoxIcon.Information);  txtPassword.Focus();  txtPassword.SelectAll();  return false;  }  if ((bool)!user.IsActive)  {  XtraMessageBox.Show("Tài khoản này đã bị khóa! Vui lòng liên hệ admin.", "Thông báo",  MessageBoxButtons.OK, MessageBoxIcon.Warning);  return false;  }  return true;  }  catch (Exception ex)  {  XtraMessageBox.Show($"Lỗi kiểm tra thông tin đăng nhập: {ex.Message}", "Lỗi",  MessageBoxButtons.OK, MessageBoxIcon.Error);  return false;  }  }  private string BuildSqlAuthConnectionString(string username, string password)  {  try  {  if (string.IsNullOrEmpty(\_sqlAuthConnStringTemplate))  {  XtraMessageBox.Show("Không tìm thấy template SQL Auth trong app.config!", "Lỗi cấu hình",  MessageBoxButtons.OK, MessageBoxIcon.Error);  return null;  }  string connString = string.Format(\_sqlAuthConnStringTemplate, username, password);  System.Diagnostics.Debug.WriteLine($"SQL Auth Conn: {connString}");  return connString;  }  catch (Exception ex)  {  XtraMessageBox.Show($"Lỗi tạo connection string: {ex.Message}", "Lỗi",  MessageBoxButtons.OK, MessageBoxIcon.Error);  return null;  }  }  private bool TestSqlAuthConnection(string connectionString)  {  try  {  using (SqlConnection conn = new SqlConnection(connectionString))  {  conn.Open();  System.Diagnostics.Debug.WriteLine($"Kết nối thành công với user: {UserSession.CurrentUserName}");  using (SqlCommand cmd = new SqlCommand("SELECT COUNT(\*) FROM [dbo].[Sach]", conn))  {  int bookCount = (int)cmd.ExecuteScalar();  System.Diagnostics.Debug.WriteLine($"Số sách trong DB: {bookCount}");  }  try  {  using (SqlCommand cmd = new SqlCommand("EXEC [dbo].[TimKiemSach]", conn))  {  cmd.CommandType = CommandType.StoredProcedure;  using (SqlDataReader reader = cmd.ExecuteReader())  {  System.Diagnostics.Debug.WriteLine("Proc TimKiemSach chạy OK");  }  }  }  catch (SqlException procEx)  {  System.Diagnostics.Debug.WriteLine($"Proc test warning: {procEx.Message}");  }  return true;  }  }  catch (SqlException sqlEx)  {  string errorDetails = $"SQL Error #{sqlEx.Number}: {sqlEx.Message}\n" +  $"State: {sqlEx.State}, Class: {sqlEx.Class}\n" +  $"Server: {sqlEx.Server}\n" +  $"User: {UserSession.CurrentUserName}";  System.Diagnostics.Debug.WriteLine(errorDetails);  if (sqlEx.Number == 18456)  {  XtraMessageBox.Show("Tài khoản hoặc mật khẩu SQL Server không đúng! Vui lòng liên hệ admin.", "Lỗi đăng nhập",  MessageBoxButtons.OK, MessageBoxIcon.Error);  }  else if (sqlEx.Number == 4060)  {  XtraMessageBox.Show("Không thể kết nối đến database QuanLyThuVien!", "Lỗi database",  MessageBoxButtons.OK, MessageBoxIcon.Error);  }  else  {  XtraMessageBox.Show($"Lỗi kết nối database: {sqlEx.Message}", "Lỗi",  MessageBoxButtons.OK, MessageBoxIcon.Error);  }  return false;  }  catch (Exception ex)  {  XtraMessageBox.Show($"Lỗi không xác định: {ex.Message}", "Lỗi",  MessageBoxButtons.OK, MessageBoxIcon.Error);  return false;  }  }  private void SaveUserSession(string username, string password, string connectionString)  {  UserSession.CurrentUserName = username;  UserSession.CurrentUserPassword = password;  UserSession.ConnectionString = connectionString;  ConnectionStringProvider.SetConnectionString(connectionString);  System.Diagnostics.Debug.WriteLine($"Session lưu thành công - User: {username}");  }  private void FrmMain\_FormClosed(object sender, FormClosedEventArgs e)  {  UserSession.Clear();  ConnectionStringProvider.Clear();    ConnectionStringProvider.SetConnectionString(\_integratedConnString);    this.Show();  txtUsername.Focus();  txtUsername.Text = "";  txtPassword.Text = "";  }  private void btnThoat\_Click(object sender, EventArgs e)  {  UserSession.Clear();  Application.Exit();  }  }  } |

## Danh mục Trang chủ

* 1. **Chức năng hiển thị một số thông tin cơ bản về thư viện (sách, thành viên)**

|  |
| --- |
| CREATE PROCEDURE [dbo].[sp\_DashboardSummary]  AS  BEGIN  -- Tổng số bản sao sách  SELECT COUNT(\*) AS TongSoSach FROM BanSaoSach;  -- Số bản sao sẵn sàng  SELECT COUNT(\*) AS SachChuaMuon  FROM BanSaoSach  WHERE TinhTrang = N'Sẵn sàng';  -- Số bản sao đang mượn  SELECT COUNT(\*) AS SachDangMuon  FROM BanSaoSach  WHERE TinhTrang = N'Đang mượn';  -- Tổng số thành viên  SELECT COUNT(\*) AS TongThanhVien FROM ThanhVien;  END; |

C#

|  |
| --- |
| public DataSet GetSummary()  {  using (var conn = new SqlConnection(\_connectionString))  using (var cmd = new SqlCommand("sp\_DashboardSummary", conn))  {  cmd.CommandType = CommandType.StoredProcedure;  var adapter = new SqlDataAdapter(cmd);  var ds = new DataSet();  adapter.Fill(ds);  return ds; }} |

**2.2 Chức năng biểu đồ đường thể hiện số lượng mượn theo ngày/tháng/năm**

|  |
| --- |
| CREATE PROCEDURE [dbo].[sp\_DashboardTrends]  @FromDate DATE,  @ToDate DATE,  @GroupBy NVARCHAR(10) = 'DAY' -- 'DAY' or 'MONTH' or 'YEAR'  AS  BEGIN  IF @GroupBy = 'DAY'  BEGIN  SELECT CAST(pm.NgayMuon AS DATE) AS Ngay, COUNT(\*) AS SoLuongMuon  FROM vw\_PhieuMuonChiTiet pm  WHERE pm.NgayMuon BETWEEN @FromDate AND @ToDate  GROUP BY CAST(pm.NgayMuon AS DATE)  ORDER BY Ngay;  END  ELSE IF @GroupBy = 'MONTH'  BEGIN  SELECT YEAR(pm.NgayMuon) AS Nam, MONTH(pm.NgayMuon) AS Thang, COUNT(\*) AS SoLuongMuon  FROM vw\_PhieuMuonChiTiet pm  WHERE pm.NgayMuon BETWEEN @FromDate AND @ToDate  GROUP BY YEAR(pm.NgayMuon), MONTH(pm.NgayMuon)  ORDER BY Nam, Thang;  END  ELSE IF @GroupBy = 'YEAR'  BEGIN  SELECT YEAR(pm.NgayMuon) AS Nam, COUNT(\*) AS SoLuongMuon  FROM vw\_PhieuMuonChiTiet pm  WHERE pm.NgayMuon BETWEEN @FromDate AND @ToDate  GROUP BY YEAR(pm.NgayMuon)  ORDER BY Nam;  END  END; |

C#:

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

* 1. **Chức năng lấy thông tin sách quá hạn từ View SachQuaHan**

|  |
| --- |
| 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. **Chức năng biểu đồ tròn thể hiện 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 GetBookStatus()  {  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;  }  } |

## Danh mục Sách

* 1. **Tab Sách**

Thêm sách

|  |
| --- |
| CREATE PROCEDURE [dbo].[ThemSach]  @TenSach NVARCHAR(200),  @ISBN NVARCHAR(13),  @MaTacGia INT,  @MaNhaXuatBan INT,  @MaTheLoai INT,  @NamXuatBan INT,  @Gia DECIMAL(10,2),  @SoLuong INT,  @TrangThai BIT = 1  AS  BEGIN  SET NOCOUNT ON;  BEGIN TRY  INSERT INTO [dbo].[Sach] (TenSach, ISBN, MaTacGia, MaNhaXuatBan, MaTheLoai, NamXuatBan, Gia, SoLuong, TrangThai)  VALUES (@TenSach, @ISBN, @MaTacGia, @MaNhaXuatBan, @MaTheLoai, @NamXuatBan, @Gia, @SoLuong, @TrangThai);  END TRY  BEGIN CATCH  DECLARE @ErrorMessage NVARCHAR(4000) = ERROR\_MESSAGE();  RAISERROR (@ErrorMessage, 16, 1);  END CATCH  END; |

C#;

|  |
| --- |
| public void AddBook(string tenSach, string isbn, int maTacGia, int maNhaXuatBan, int maTheLoai,  int namXuatBan, decimal gia, int soLuong, bool trangThai = true)  {  if (string.IsNullOrWhiteSpace(tenSach))  throw new ArgumentException("Tên sách không được để trống.", nameof(tenSach));  if (string.IsNullOrWhiteSpace(isbn))  throw new ArgumentException("ISBN không được để trống.", nameof(isbn));  if (maTacGia <= 0)  throw new ArgumentException("Mã tác giả không hợp lệ.", nameof(maTacGia));  if (maNhaXuatBan <= 0)  throw new ArgumentException("Mã nhà xuất bản không hợp lệ.", nameof(maNhaXuatBan));  if (maTheLoai <= 0)  throw new ArgumentException("Mã thể loại không hợp lệ.", nameof(maTheLoai));  using (var connection = new SqlConnection(\_connectionString))  {  using (var command = new SqlCommand("[dbo].[ThemSach]", connection))  {  command.CommandType = CommandType.StoredProcedure;  command.CommandTimeout = 60;  command.Parameters.Add("@TenSach", SqlDbType.NVarChar, 200).Value = tenSach;  command.Parameters.Add("@ISBN", SqlDbType.NVarChar, 13).Value = isbn;  command.Parameters.Add("@MaTacGia", SqlDbType.Int).Value = maTacGia;  command.Parameters.Add("@MaNhaXuatBan", SqlDbType.Int).Value = maNhaXuatBan;  command.Parameters.Add("@MaTheLoai", SqlDbType.Int).Value = maTheLoai;  command.Parameters.Add("@NamXuatBan", SqlDbType.Int).Value = namXuatBan;  command.Parameters.Add("@Gia", SqlDbType.Decimal).Value = gia;  command.Parameters.Add("@SoLuong", SqlDbType.Int).Value = soLuong;  command.Parameters.Add("@TrangThai", SqlDbType.Bit).Value = trangThai;  connection.Open();  command.ExecuteNonQuery();  }  }  } |

Sửa sách

|  |
| --- |
| CREATE PROCEDURE [dbo].[SuaSach]  @MaSach INT,  @TenSach NVARCHAR(200),  @ISBN NVARCHAR(13),  @MaTacGia INT,  @MaNhaXuatBan INT,  @MaTheLoai INT,  @NamXuatBan INT,  @Gia DECIMAL(10,2),  @SoLuong INT,  @TrangThai BIT  AS  BEGIN  SET NOCOUNT ON;  BEGIN TRY  UPDATE [dbo].[Sach]  SET TenSach = @TenSach,  ISBN = @ISBN,  MaTacGia = @MaTacGia,  MaNhaXuatBan = @MaNhaXuatBan,  MaTheLoai = @MaTheLoai,  NamXuatBan = @NamXuatBan,  Gia = @Gia,  SoLuong = @SoLuong,  TrangThai = @TrangThai  WHERE MaSach = @MaSach;  END TRY  BEGIN CATCH  DECLARE @ErrorMessage NVARCHAR(4000) = ERROR\_MESSAGE();  RAISERROR (@ErrorMessage, 16, 1);  END CATCH  END; |

C#:

|  |
| --- |
| public void UpdateBook(int maSach, string tenSach, string isbn, int maTacGia, int maNhaXuatBan, int maTheLoai,  int namXuatBan, decimal gia, int soLuong, bool trangThai)  {  if (maSach <= 0)  throw new ArgumentException("Mã sách không hợp lệ.", nameof(maSach));  if (string.IsNullOrWhiteSpace(tenSach))  throw new ArgumentException("Tên sách không được để trống.", nameof(tenSach));  if (string.IsNullOrWhiteSpace(isbn))  throw new ArgumentException("ISBN không được để trống.", nameof(isbn));  if (maTacGia <= 0)  throw new ArgumentException("Mã tác giả không hợp lệ.", nameof(maTacGia));  if (maNhaXuatBan <= 0)  throw new ArgumentException("Mã nhà xuất bản không hợp lệ.", nameof(maNhaXuatBan));  if (maTheLoai <= 0)  throw new ArgumentException("Mã thể loại không hợp lệ.", nameof(maTheLoai));  using (var connection = new SqlConnection(\_connectionString))  {  using (var command = new SqlCommand("[dbo].[SuaSach]", connection))  {  command.CommandType = CommandType.StoredProcedure;  command.CommandTimeout = 60;  command.Parameters.Add("@MaSach", SqlDbType.Int).Value = maSach;  command.Parameters.Add("@TenSach", SqlDbType.NVarChar, 200).Value = tenSach;  command.Parameters.Add("@ISBN", SqlDbType.NVarChar, 13).Value = isbn;  command.Parameters.Add("@MaTacGia", SqlDbType.Int).Value = maTacGia;  command.Parameters.Add("@MaNhaXuatBan", SqlDbType.Int).Value = maNhaXuatBan;  command.Parameters.Add("@MaTheLoai", SqlDbType.Int).Value = maTheLoai;  command.Parameters.Add("@NamXuatBan", SqlDbType.Int).Value = namXuatBan;  command.Parameters.Add("@Gia", SqlDbType.Decimal).Value = gia;  command.Parameters.Add("@SoLuong", SqlDbType.Int).Value = soLuong;  command.Parameters.Add("@TrangThai", SqlDbType.Bit).Value = trangThai;  connection.Open();  command.ExecuteNonQuery();  }  }  } |

Xoá sách

|  |
| --- |
| CREATE PROCEDURE [dbo].[XoaSach]  @MaSach INT  AS  BEGIN  SET NOCOUNT ON;  BEGIN TRY  DELETE FROM [dbo].[Sach] WHERE MaSach = @MaSach;  END TRY  BEGIN CATCH  DECLARE @ErrorMessage NVARCHAR(4000) = ERROR\_MESSAGE();  RAISERROR (@ErrorMessage, 16, 1);  END CATCH  END; |

C#:

|  |
| --- |
| public void DeleteBook(int maSach)  {  if (maSach <= 0)  throw new ArgumentException("Mã sách không hợp lệ.", nameof(maSach));  using (var connection = new SqlConnection(\_connectionString))  {  using (var command = new SqlCommand("[dbo].[XoaSach]", connection))  {  command.CommandType = CommandType.StoredProcedure;  command.CommandTimeout = 60;  command.Parameters.Add("@MaSach", SqlDbType.Int).Value = maSach;  connection.Open();  command.ExecuteNonQuery();  }  }  } |

* 1. **Tab Nhà xuất bản**

Thêm Nhà xuất bản

|  |
| --- |
| CREATE PROCEDURE [dbo].[ThemNhaXuatBan]  @TenNhaXuatBan NVARCHAR(100),  @DiaChi NVARCHAR(200) = NULL,  @SoDienThoai NVARCHAR(20) = NULL  AS  BEGIN  SET NOCOUNT ON;  BEGIN TRY  INSERT INTO [dbo].[NhaXuatBan] (TenNhaXuatBan, DiaChi, SoDienThoai)  VALUES (@TenNhaXuatBan, @DiaChi, @SoDienThoai);  END TRY  BEGIN CATCH  DECLARE @ErrorMessage NVARCHAR(4000) = ERROR\_MESSAGE();  RAISERROR (@ErrorMessage, 16, 1);  END CATCH  END; |

C#:

|  |
| --- |
| public void AddPublisher(string tenNhaXuatBan, string diaChi = null, string soDienThoai = null)  {  if (string.IsNullOrWhiteSpace(tenNhaXuatBan))  throw new ArgumentException("Tên nhà xuất bản không được để trống.", nameof(tenNhaXuatBan));  using (var connection = new SqlConnection(\_connectionString))  {  using (var command = new SqlCommand("[dbo].[ThemNhaXuatBan]", connection))  {  command.CommandType = CommandType.StoredProcedure;  command.CommandTimeout = 60;  command.Parameters.Add("@TenNhaXuatBan", SqlDbType.NVarChar, 100).Value = tenNhaXuatBan;  command.Parameters.Add("@DiaChi", SqlDbType.NVarChar, 200).Value =  string.IsNullOrWhiteSpace(diaChi) ? (object)DBNull.Value : diaChi;  command.Parameters.Add("@SoDienThoai", SqlDbType.NVarChar, 20).Value =  string.IsNullOrWhiteSpace(soDienThoai) ? (object)DBNull.Value : soDienThoai;  connection.Open();  command.ExecuteNonQuery();  }  }  } |

Sửa Nhà xuất bản

|  |
| --- |
| CREATE PROCEDURE [dbo].[SuaNhaXuatBan]  @MaNhaXuatBan INT,  @TenNhaXuatBan NVARCHAR(100),  @DiaChi NVARCHAR(200) = NULL,  @SoDienThoai NVARCHAR(20) = NULL  AS  BEGIN  SET NOCOUNT ON;  BEGIN TRY  UPDATE [dbo].[NhaXuatBan]  SET TenNhaXuatBan = @TenNhaXuatBan,  DiaChi = @DiaChi,  SoDienThoai = @SoDienThoai  WHERE MaNhaXuatBan = @MaNhaXuatBan;  END TRY  BEGIN CATCH  DECLARE @ErrorMessage NVARCHAR(4000) = ERROR\_MESSAGE();  RAISERROR (@ErrorMessage, 16, 1);  END CATCH  END; |

C#:

|  |
| --- |
| public void UpdatePublisher(int maNhaXuatBan, string tenNhaXuatBan, string diaChi = null, string soDienThoai = null)  {  if (maNhaXuatBan <= 0)  throw new ArgumentException("Mã nhà xuất bản không hợp lệ.", nameof(maNhaXuatBan));    if (string.IsNullOrWhiteSpace(tenNhaXuatBan))  throw new ArgumentException("Tên nhà xuất bản không được để trống.", nameof(tenNhaXuatBan));  using (var connection = new SqlConnection(\_connectionString))  {  using (var command = new SqlCommand("[dbo].[SuaNhaXuatBan]", connection))  {  command.CommandType = CommandType.StoredProcedure;  command.CommandTimeout = 60;  command.Parameters.Add("@MaNhaXuatBan", SqlDbType.Int).Value = maNhaXuatBan;  command.Parameters.Add("@TenNhaXuatBan", SqlDbType.NVarChar, 100).Value = tenNhaXuatBan;  command.Parameters.Add("@DiaChi", SqlDbType.NVarChar, 200).Value =  string.IsNullOrWhiteSpace(diaChi) ? (object)DBNull.Value : diaChi;  command.Parameters.Add("@SoDienThoai", SqlDbType.NVarChar, 20).Value =  string.IsNullOrWhiteSpace(soDienThoai) ? (object)DBNull.Value : soDienThoai;  connection.Open();  command.ExecuteNonQuery();  }  }  } |

Xoá Nhà xuất bản

|  |
| --- |
| CREATE PROCEDURE [dbo].[XoaNhaXuatBan]  @MaNhaXuatBan INT  AS  BEGIN  SET NOCOUNT ON;  BEGIN TRY  -- Kiểm tra xem nhà xuất bản có được tham chiếu trong bảng Sach không  IF EXISTS (SELECT 1 FROM [dbo].[Sach] WHERE MaNhaXuatBan = @MaNhaXuatBan)  BEGIN  RAISERROR (N'Không thể xóa nhà xuất bản vì vẫn còn sách liên quan.', 16, 1);  RETURN;  END  DELETE FROM [dbo].[NhaXuatBan] WHERE MaNhaXuatBan = @MaNhaXuatBan;  END TRY  BEGIN CATCH  DECLARE @ErrorMessage NVARCHAR(4000) = ERROR\_MESSAGE();  RAISERROR (@ErrorMessage, 16, 1);  END CATCH  END; |

C#:

|  |
| --- |
| public void DeletePublisher(int maNXB)  {  if (maNXB <= 0)  throw new ArgumentException("Mã nhà xuất bản không hợp lệ.", nameof(maNXB));  using (var connection = new SqlConnection(\_connectionString))  {  using (var command = new SqlCommand("[dbo].[XoaNhaXuatBan]", connection))  {  command.CommandType = CommandType.StoredProcedure;  command.CommandTimeout = 60;  command.Parameters.Add("@MaNhaXuatBan", SqlDbType.Int).Value = maNXB;  connection.Open();  command.ExecuteNonQuery();  }  }  } |

* 1. **Tab Tác giả**

Thêm Tác giả

|  |
| --- |
| CREATE PROCEDURE [dbo].[ThemTacGia]  @TenTacGia NVARCHAR(100),  @QuocTich NVARCHAR(50) = NULL,  @NamSinh INT = NULL  AS  BEGIN  SET NOCOUNT ON;  BEGIN TRY  -- Kiểm tra NamSinh hợp lệ  IF @NamSinh IS NOT NULL AND (@NamSinh <= 1800 OR @NamSinh > YEAR(GETDATE()))  BEGIN  RAISERROR (N'Năm sinh phải lớn hơn 1800 và nhỏ hơn hoặc bằng năm hiện tại.', 16, 1);  RETURN;  END  INSERT INTO [dbo].[TacGia] (TenTacGia, QuocTich, NamSinh)  VALUES (@TenTacGia, @QuocTich, @NamSinh);  END TRY  BEGIN CATCH  DECLARE @ErrorMessage NVARCHAR(4000) = ERROR\_MESSAGE();  RAISERROR (@ErrorMessage, 16, 1);  END CATCH  END; |

C#:

|  |
| --- |
| public void AddAuthor(string tenTacGia, string quocTich = null, int? namSinh = null)  {  if (string.IsNullOrWhiteSpace(tenTacGia))  throw new ArgumentException("Tên tác giả không được để trống.", nameof(tenTacGia));  using (var connection = new SqlConnection(\_connectionString))  {  using (var command = new SqlCommand("[dbo].[ThemTacGia]", connection))  {  command.CommandType = CommandType.StoredProcedure;  command.CommandTimeout = 60;  command.Parameters.Add("@TenTacGia", SqlDbType.NVarChar, 100).Value = tenTacGia;  command.Parameters.Add("@QuocTich", SqlDbType.NVarChar, 50).Value =  string.IsNullOrWhiteSpace(quocTich) ? (object)DBNull.Value : quocTich;  command.Parameters.Add("@NamSinh", SqlDbType.Int).Value =  namSinh.HasValue ? (object)namSinh.Value : DBNull.Value;  connection.Open();  command.ExecuteNonQuery();  }  }  } |

Sửa Tác giả

|  |
| --- |
| CREATE PROCEDURE [dbo].[SuaTacGia]  @MaTacGia INT,  @TenTacGia NVARCHAR(100),  @QuocTich NVARCHAR(50) = NULL,  @NamSinh INT = NULL  AS  BEGIN  SET NOCOUNT ON;  BEGIN TRY  -- Kiểm tra NamSinh hợp lệ  IF @NamSinh IS NOT NULL AND (@NamSinh <= 1800 OR @NamSinh > YEAR(GETDATE()))  BEGIN  RAISERROR (N'Năm sinh phải lớn hơn 1800 và nhỏ hơn hoặc bằng năm hiện tại.', 16, 1);  RETURN;  END  UPDATE [dbo].[TacGia]  SET TenTacGia = @TenTacGia,  QuocTich = @QuocTich,  NamSinh = @NamSinh  WHERE MaTacGia = @MaTacGia;  END TRY  BEGIN CATCH  DECLARE @ErrorMessage NVARCHAR(4000) = ERROR\_MESSAGE();  RAISERROR (@ErrorMessage, 16, 1);  END CATCH  END; |

C#:

|  |
| --- |
| public void UpdateAuthor(int maTacGia, string tenTacGia, string quocTich = null, int? namSinh = null)  {  if (maTacGia <= 0)  throw new ArgumentException("Mã tác giả không hợp lệ.", nameof(maTacGia));    if (string.IsNullOrWhiteSpace(tenTacGia))  throw new ArgumentException("Tên tác giả không được để trống.", nameof(tenTacGia));  using (var connection = new SqlConnection(\_connectionString))  {  using (var command = new SqlCommand("[dbo].[SuaTacGia]", connection))  {  command.CommandType = CommandType.StoredProcedure;  command.CommandTimeout = 60;  command.Parameters.Add("@MaTacGia", SqlDbType.Int).Value = maTacGia;  command.Parameters.Add("@TenTacGia", SqlDbType.NVarChar, 100).Value = tenTacGia;  command.Parameters.Add("@QuocTich", SqlDbType.NVarChar, 50).Value =  string.IsNullOrWhiteSpace(quocTich) ? (object)DBNull.Value : quocTich;  command.Parameters.Add("@NamSinh", SqlDbType.Int).Value =  namSinh.HasValue ? (object)namSinh.Value : DBNull.Value;  connection.Open();  command.ExecuteNonQuery();  }  }} |

Xoá tác giả

|  |
| --- |
| CREATE PROCEDURE [dbo].[XoaTacGia]  @MaTacGia INT  AS  BEGIN  SET NOCOUNT ON;  BEGIN TRY  -- Kiểm tra xem tác giả có được tham chiếu trong bảng Sach không  IF EXISTS (SELECT 1 FROM [dbo].[Sach] WHERE MaTacGia = @MaTacGia)  BEGIN  RAISERROR (N'Không thể xóa tác giả vì vẫn còn sách liên quan.', 16, 1);  RETURN;  END  DELETE FROM [dbo].[TacGia] WHERE MaTacGia = @MaTacGia;  END TRY  BEGIN CATCH  DECLARE @ErrorMessage NVARCHAR(4000) = ERROR\_MESSAGE();  RAISERROR (@ErrorMessage, 16, 1);  END CATCH  END; |

C#:

|  |
| --- |
| public void DeleteAuthor(int maTacGia)  {  if (maTacGia <= 0)  throw new ArgumentException("Mã tác giả không hợp lệ.", nameof(maTacGia));  using (var connection = new SqlConnection(\_connectionString))  {  using (var command = new SqlCommand("[dbo].[XoaTacGia]", connection))  {  command.CommandType = CommandType.StoredProcedure;  command.CommandTimeout = 60;  command.Parameters.Add("@MaTacGia", SqlDbType.Int).Value = maTacGia;  connection.Open();  command.ExecuteNonQuery();  }  }  } |

* 1. **Tab Thể loại**

Thêm Thể loại

|  |
| --- |
| CREATE PROCEDURE [dbo].[ThemTheLoai]  @TenTheLoai NVARCHAR(50),  @MoTa NVARCHAR(200) = NULL  AS  BEGIN  SET NOCOUNT ON;  BEGIN TRY  INSERT INTO [dbo].[TheLoai] (TenTheLoai, MoTa)  VALUES (@TenTheLoai, @MoTa);  END TRY  BEGIN CATCH  DECLARE @ErrorMessage NVARCHAR(4000) = ERROR\_MESSAGE();  RAISERROR (@ErrorMessage, 16, 1);  END CATCH  END; |

C#:

|  |
| --- |
| public void AddTheLoai(string tenTheLoai, string moTa = null)  {  if (string.IsNullOrWhiteSpace(tenTheLoai))  throw new ArgumentException("Tên thể loại không được để trống.", nameof(tenTheLoai));  using (var connection = new SqlConnection(\_connectionString))  {  using (var command = new SqlCommand("[dbo].[ThemTheLoai]", connection))  {  command.CommandType = CommandType.StoredProcedure;  command.CommandTimeout = 60;  command.Parameters.Add("@TenTheLoai", SqlDbType.NVarChar, 50).Value = tenTheLoai;  command.Parameters.Add("@MoTa", SqlDbType.NVarChar, 200).Value =  string.IsNullOrWhiteSpace(moTa) ? (object)DBNull.Value : moTa;  connection.Open();  command.ExecuteNonQuery();  } } |

Sửa Thể loại

|  |
| --- |
| CREATE PROCEDURE [dbo].[SuaTheLoai]  @MaTheLoai INT,  @TenTheLoai NVARCHAR(50),  @MoTa NVARCHAR(200) = NULL  AS  BEGIN  SET NOCOUNT ON;  BEGIN TRY  UPDATE [dbo].[TheLoai]  SET TenTheLoai = @TenTheLoai,  MoTa = @MoTa  WHERE MaTheLoai = @MaTheLoai;  END TRY  BEGIN CATCH  DECLARE @ErrorMessage NVARCHAR(4000) = ERROR\_MESSAGE();  RAISERROR (@ErrorMessage, 16, 1);  END CATCH  END; |

C#:

|  |
| --- |
| public void UpdateTheLoai(int maTheLoai, string tenTheLoai, string moTa = null)  {  if (maTheLoai <= 0)  throw new ArgumentException("Mã thể loại không hợp lệ.", nameof(maTheLoai));  if (string.IsNullOrWhiteSpace(tenTheLoai))  throw new ArgumentException("Tên thể loại không được để trống.", nameof(tenTheLoai));  using (var connection = new SqlConnection(\_connectionString))  {  using (var command = new SqlCommand("[dbo].[SuaTheLoai]", connection))  {  command.CommandType = CommandType.StoredProcedure;  command.CommandTimeout = 60;  command.Parameters.Add("@MaTheLoai", SqlDbType.Int).Value = maTheLoai;  command.Parameters.Add("@TenTheLoai", SqlDbType.NVarChar, 50).Value = tenTheLoai;  command.Parameters.Add("@MoTa", SqlDbType.NVarChar, 200).Value =  string.IsNullOrWhiteSpace(moTa) ? (object)DBNull.Value : moTa;  connection.Open();  command.ExecuteNonQuery();  }  }  } |

Xoá Thể loại

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

C#:

|  |
| --- |
| public void DeleteTheLoai(int maTheLoai)  {  if (maTheLoai <= 0)  throw new ArgumentException("Mã thể loại không hợp lệ.", nameof(maTheLoai));  using (var connection = new SqlConnection(\_connectionString))  {  using (var command = new SqlCommand("[dbo].[XoaTheLoai]", connection))  {  command.CommandType = CommandType.StoredProcedure;  command.CommandTimeout = 60;  command.Parameters.Add("@MaTheLoai", SqlDbType.Int).Value = maTheLoai;  connection.Open();  command.ExecuteNonQuery();  }  }  } |

# Danh mục Thành viên

Thêm Thành viên

|  |
| --- |
| CREATE PROCEDURE [dbo].[ThemThanhVien]  @MaThanhVien INT,  @TenThanhVien NVARCHAR(100),  @Email NVARCHAR(100),  @SoDienThoai NVARCHAR(20) = NULL,  @DiaChi NVARCHAR(200) = NULL,  @LoaiThanhVien NVARCHAR(20) = N'SinhVien',  @NgayDangKy DATE = NULL  AS  BEGIN  SET NOCOUNT ON;  BEGIN TRY  -- Kiểm tra định dạng Email  IF @Email NOT LIKE '%@%.%'  BEGIN  RAISERROR (N'Email không hợp lệ.', 16, 1);  RETURN;  END  -- Kiểm tra LoaiThanhVien hợp lệ (nếu cần, có thể thêm các giá trị hợp lệ cụ thể)  IF @LoaiThanhVien NOT IN (N'SinhVien', N'GiaoVien') -- Giả sử các loại thành viên  BEGIN  RAISERROR (N'Loại thành viên không hợp lệ.', 16, 1);  RETURN;  END  -- Gán giá trị mặc định cho NgayDangKy nếu NULL  SET @NgayDangKy = COALESCE(@NgayDangKy, GETDATE());  INSERT INTO [dbo].[ThanhVien] (MaThanhVien, TenThanhVien, Email, SoDienThoai, DiaChi, LoaiThanhVien, NgayDangKy)  VALUES (@MaThanhVien, @TenThanhVien, @Email, @SoDienThoai, @DiaChi, @LoaiThanhVien, @NgayDangKy);  END TRY  BEGIN CATCH  DECLARE @ErrorMessage NVARCHAR(4000) = ERROR\_MESSAGE();  RAISERROR (@ErrorMessage, 16, 1);  END CATCH  END; |

C#:

|  |
| --- |
| public void AddMember(int maThanhVien, string tenThanhVien, string email, string soDienThoai = null,  string diaChi = null, string loaiThanhVien = "SinhVien", DateTime? ngayDangKy = null)  {  if (maThanhVien <= 0)  throw new ArgumentException("Mã thành viên phải lớn hơn 0", nameof(maThanhVien));  if (string.IsNullOrWhiteSpace(tenThanhVien))  throw new ArgumentException("Tên thành viên không được để trống", nameof(tenThanhVien));  if (string.IsNullOrWhiteSpace(email))  throw new ArgumentException("Email không được để trống", nameof(email));  if (string.IsNullOrWhiteSpace(loaiThanhVien))  throw new ArgumentException("Loại thành viên không được để trống", nameof(loaiThanhVien));  using (var connection = new SqlConnection(\_connectionString))  {  using (var command = new SqlCommand("[dbo].[ThemThanhVien]", connection))  {  command.CommandType = CommandType.StoredProcedure;  command.CommandTimeout = 60;  command.Parameters.Add("@MaThanhVien", SqlDbType.Int).Value = maThanhVien;  command.Parameters.Add("@TenThanhVien", SqlDbType.NVarChar, 100).Value = tenThanhVien;  command.Parameters.Add("@Email", SqlDbType.NVarChar, 100).Value = email;  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();  }  }  } |

Sửa Thành viên

|  |
| --- |
| CREATE PROCEDURE [dbo].[SuaThanhVien]  @MaThanhVien INT,  @TenThanhVien NVARCHAR(100),  @Email NVARCHAR(100),  @SoDienThoai NVARCHAR(20) = NULL,  @DiaChi NVARCHAR(200) = NULL,  @LoaiThanhVien NVARCHAR(20),  @NgayDangKy DATE  AS  BEGIN  SET NOCOUNT ON;  BEGIN TRY  -- Kiểm tra định dạng Email  IF @Email NOT LIKE '%@%.%'  BEGIN  RAISERROR (N'Email không hợp lệ.', 16, 1);  RETURN;  END  -- Kiểm tra LoaiThanhVien hợp lệ  IF @LoaiThanhVien NOT IN (N'SinhVien', N'GiaoVien') -- Giả sử các loại thành viên  BEGIN  RAISERROR (N'Loại thành viên không hợp lệ.', 16, 1);  RETURN;  END  UPDATE [dbo].[ThanhVien]  SET TenThanhVien = @TenThanhVien,  Email = @Email,  SoDienThoai = @SoDienThoai,  DiaChi = @DiaChi,  LoaiThanhVien = @LoaiThanhVien,  NgayDangKy = @NgayDangKy  WHERE MaThanhVien = @MaThanhVien;  END TRY  BEGIN CATCH  DECLARE @ErrorMessage NVARCHAR(4000) = ERROR\_MESSAGE();  RAISERROR (@ErrorMessage, 16, 1);  END CATCH  END; |

C#:

|  |
| --- |
| public void UpdateMember(int maThanhVien, string tenThanhVien, string email, string soDienThoai = null,  string diaChi = null, string loaiThanhVien = "SinhVien", DateTime? ngayDangKy = null)  {  if (maThanhVien <= 0)  throw new ArgumentException("Mã thành viên không hợp lệ", nameof(maThanhVien));  if (string.IsNullOrWhiteSpace(tenThanhVien))  throw new ArgumentException("Tên thành viên không được để trống", nameof(tenThanhVien));  if (string.IsNullOrWhiteSpace(email))  throw new ArgumentException("Email không được để trống", nameof(email));  if (string.IsNullOrWhiteSpace(loaiThanhVien))  throw new ArgumentException("Loại thành viên không được để trống", nameof(loaiThanhVien));  using (var connection = new SqlConnection(\_connectionString))  {  using (var command = new SqlCommand("[dbo].[SuaThanhVien]", connection))  {  command.CommandType = CommandType.StoredProcedure;  command.CommandTimeout = 60;  command.Parameters.Add("@MaThanhVien", SqlDbType.Int).Value = maThanhVien;  command.Parameters.Add("@TenThanhVien", SqlDbType.NVarChar, 100).Value = tenThanhVien;  command.Parameters.Add("@Email", SqlDbType.NVarChar, 100).Value = email;  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 ?? DateTime.Now;  connection.Open();  command.ExecuteNonQuery();  }  }  } |

Xoá Thành viên

|  |
| --- |
| CREATE PROCEDURE [dbo].[XoaThanhVien]  @MaThanhVien INT  AS  BEGIN  SET NOCOUNT ON;  BEGIN TRY  -- Kiểm tra xem thành viên có liên quan đến PhieuMuon hoặc Reservation không  IF EXISTS (SELECT 1 FROM [dbo].[PhieuMuon] WHERE MaThanhVien = @MaThanhVien)  OR EXISTS (SELECT 1 FROM [dbo].[Reservation] WHERE MaThanhVien = @MaThanhVien)  BEGIN  RAISERROR (N'Không thể xóa thành viên vì vẫn còn phiếu mượn hoặc đặt trước liên quan.', 16, 1);  RETURN;  END  DELETE FROM [dbo].[ThanhVien] WHERE MaThanhVien = @MaThanhVien;  END TRY  BEGIN CATCH  DECLARE @ErrorMessage NVARCHAR(4000) = ERROR\_MESSAGE();  RAISERROR (@ErrorMessage, 16, 1);  END CATCH  END; |

C#:

|  |
| --- |
| public void DeleteMember(int maThanhVien)  {  if (maThanhVien <= 0)  throw new ArgumentException("Mã thành viên không hợp lệ", nameof(maThanhVien));  using (var connection = new SqlConnection(\_connectionString))  {  using (var command = new SqlCommand("[dbo].[XoaThanhVien]", connection))  {  command.CommandType = CommandType.StoredProcedure;  command.CommandTimeout = 60;  command.Parameters.Add("@MaThanhVien", SqlDbType.Int).Value = maThanhVien;  connection.Open();  command.ExecuteNonQuery();  }  }  } |

## Danh mục Mượn/Trả

* 1. **Chức năng mượn sách**

|  |
| --- |
| CREATE PROCEDURE [dbo].[MuonSachProc]  @MaThanhVien INT,  @UserID INT,  @NgayTraDuKien DATE,  @ListBanSao dbo.ListBanSao READONLY,  @MaPhieuMuon INT = NULL,  @GhiChu NVARCHAR(200)  AS  BEGIN  SET NOCOUNT ON;  SET XACT\_ABORT ON;  BEGIN TRY  BEGIN TRAN;  -- Nếu chưa có phiếu thì tạo mới  IF @MaPhieuMuon IS NULL  BEGIN  INSERT INTO dbo.PhieuMuon (MaThanhVien, UserID, NgayMuon, NgayTraDuKien, TrangThai, GhiChu)  VALUES (@MaThanhVien, @UserID, GETDATE(), @NgayTraDuKien, N'Đang mượn',@GhiChu);  SET @MaPhieuMuon = SCOPE\_IDENTITY();  END  -- Thêm nhiều bản sao sách vào ChiTietPhieuMuon  INSERT INTO dbo.ChiTietPhieuMuon (MaPhieuMuon, MaSach, MaBanSao, NgayTraDuKien, TrangThai, GhiChu)  SELECT  @MaPhieuMuon,  b.MaSach,  l.MaBanSao,  @NgayTraDuKien,  N'Đang mượn',  @GhiChu  FROM @ListBanSao l  JOIN dbo.BanSaoSach b ON l.MaBanSao = b.MaBanSao  WHERE b.TinhTrang = N'Sẵn sàng';  -- Update trạng thái của các bản sao thành "Đang mượn"  UPDATE bs  SET bs.TinhTrang = N'Đang mượn'  FROM dbo.BanSaoSach bs  JOIN @ListBanSao l ON bs.MaBanSao = l.MaBanSao;  SELECT @MaPhieuMuon AS MaPhieuMuon;  COMMIT TRAN;  END TRY  BEGIN CATCH  IF XACT\_STATE()<>0 ROLLBACK TRAN;  THROW;  END CATCH  END; |

C#:

|  |
| --- |
| public int ExecuteMuonSachProc(int maThanhVien, int userId, DateTime ngayTraDuKien, List<int> listMaBanSao, string ghichu, int? maPhieuMuon = null)  {  using (var connection = new SqlConnection(\_connectionString))  {  using (var command = new SqlCommand("MuonSachProc", connection))  {  command.CommandType = CommandType.StoredProcedure;  var listBanSaoTable = new DataTable();  listBanSaoTable.Columns.Add("MaBanSao", typeof(int));  foreach (var maBanSao in listMaBanSao)  {  listBanSaoTable.Rows.Add(maBanSao);  }  command.Parameters.Add("@MaThanhVien", SqlDbType.Int).Value = maThanhVien;  command.Parameters.Add("@UserID", SqlDbType.Int).Value = userId;  command.Parameters.Add("@NgayTraDuKien", SqlDbType.Date).Value = ngayTraDuKien;  command.Parameters.Add("@GhiChu", SqlDbType.NVarChar, 200).Value = ghichu ?? (object)DBNull.Value;  var tableParam = command.Parameters.AddWithValue("@ListBanSao", listBanSaoTable);  tableParam.SqlDbType = SqlDbType.Structured;  tableParam.TypeName = "dbo.ListBanSao";  if (maPhieuMuon.HasValue)  {  command.Parameters.Add("@MaPhieuMuon", SqlDbType.Int).Value = maPhieuMuon.Value;  }  else  {  command.Parameters.Add("@MaPhieuMuon", SqlDbType.Int).Value = DBNull.Value;  }  connection.Open();  var result = command.ExecuteScalar();  return maPhieuMuon ?? Convert.ToInt32(result);  }  }  } |

* 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. Thủ thư (Librarian)

|  |
| --- |
| CREATE ROLE [LibrarianRole];  GO  -- Bảng quản lý sách và bản sao (full quyền: SELECT, INSERT, UPDATE, DELETE)  GRANT SELECT, INSERT, UPDATE, DELETE ON [dbo].[Sach] TO [LibrarianRole];  GRANT SELECT, INSERT, UPDATE, DELETE ON [dbo].[BanSaoSach] TO [LibrarianRole];  -- Bảng quản lý tác giả, nhà xuất bản, thể loại (full quyền)  GRANT SELECT, INSERT, UPDATE, DELETE ON [dbo].[TacGia] TO [LibrarianRole];  GRANT SELECT, INSERT, UPDATE, DELETE ON [dbo].[NhaXuatBan] TO [LibrarianRole];  GRANT SELECT, INSERT, UPDATE, DELETE ON [dbo].[TheLoai] TO [LibrarianRole];  -- Bảng quản lý thành viên (full quyền)  GRANT SELECT, INSERT, UPDATE, DELETE ON [dbo].[ThanhVien] TO [LibrarianRole];  -- Bảng quản lý mượn/trả (full quyền)  GRANT SELECT, INSERT, UPDATE, DELETE ON [dbo].[PhieuMuon] TO [LibrarianRole];  GRANT SELECT, INSERT, UPDATE, DELETE ON [dbo].[ChiTietPhieuMuon] TO [LibrarianRole];  -- Bảng quản lý phạt (full quyền)  GRANT SELECT, INSERT, UPDATE, DELETE ON [dbo].[Phat] TO [LibrarianRole];  -- Bảng thanh toán lịch sử (full quyền)  GRANT SELECT, INSERT, UPDATE, DELETE ON [dbo].[PaymentHistory] TO [LibrarianRole];  -- Bảng task scheduler (full quyền - giả sử librarian có thể quản lý task)  GRANT SELECT, INSERT, UPDATE, DELETE ON [dbo].[TaskScheduler] TO [LibrarianRole];  -- Bảng roles (chỉ SELECT - không cho phép sửa roles, vì liên quan security)  GRANT SELECT ON [dbo].[Roles] TO [LibrarianRole];  DENY INSERT, UPDATE, DELETE ON [dbo].[Roles] TO [LibrarianRole];  -- =====================================================  -- 2. KHÔNG CHO QUYỀN TRÊN BẢNG NHẠY CẢM (Users, CauHinhHeThong, AuditLog)  -- =====================================================  -- Bảng Users – Chỉ SELECT  GRANT SELECT ON ON [dbo].[Users] TO [LibrarianRole];  -- Bảng CauHinhHeThong - CHỈ SELECT (xem cấu hình, không sửa/xóa/insert)  GRANT SELECT ON [dbo].[CauHinhHeThong] TO [LibrarianRole];  DENY INSERT, UPDATE, DELETE ON [dbo].[CauHinhHeThong] TO [LibrarianRole];  -- Bảng AuditLog - CHỈ SELECT (xem log, không sửa/xóa/insert)  GRANT SELECT ON [dbo].[AuditLog] TO [LibrarianRole];  DENY INSERT, UPDATE, DELETE ON [dbo].[AuditLog] TO [LibrarianRole];  GO  -- =====================================================  -- 3. QUYỀN TRÊN CÁC VIEWS (CHỈ SELECT - đọc báo cáo)  -- =====================================================  -- Tất cả views đều chỉ được SELECT  GRANT SELECT ON [dbo].[SachQuaHanTheoThanhVien] TO [LibrarianRole];  GRANT SELECT ON [dbo].[DanhSachSachDangMuon] TO [LibrarianRole];  GRANT SELECT ON [dbo].[TopSachMuon] TO [LibrarianRole];  GRANT SELECT ON [dbo].[SachQuaHan] TO [LibrarianRole];  GRANT SELECT ON [dbo].[BaoCaoPhat] TO [LibrarianRole];  GRANT SELECT ON [dbo].[ThanhVienBiKhoa] TO [LibrarianRole];  GRANT SELECT ON [dbo].[vw\_TopBooks] TO [LibrarianRole];  GRANT SELECT ON [dbo].[vw\_TongTienPhat] TO [LibrarianRole];  GRANT SELECT ON [dbo].[vw\_PhieuMuonChiTiet] TO [LibrarianRole];  GRANT SELECT ON [dbo].[vw\_TinhTrangSach] TO [LibrarianRole];  GO  -- =====================================================  -- 4. QUYỀN EXECUTE TRÊN STORED PROCEDURES (Full EXECUTE trừ proc liên quan Users/CauHinhHeThong)  -- =====================================================  -- Proc quản lý sách, tác giả, nhà xuất bản, thể loại (full EXECUTE)  GRANT EXECUTE ON [dbo].[ThemSach] TO [LibrarianRole];  GRANT EXECUTE ON [dbo].[SuaSach] TO [LibrarianRole];  GRANT EXECUTE ON [dbo].[XoaSach] TO [LibrarianRole];  GRANT EXECUTE ON [dbo].[ThemTacGia] TO [LibrarianRole];  GRANT EXECUTE ON [dbo].[SuaTacGia] TO [LibrarianRole];  GRANT EXECUTE ON [dbo].[XoaTacGia] TO [LibrarianRole];  GRANT EXECUTE ON [dbo].[ThemNhaXuatBan] TO [LibrarianRole];  GRANT EXECUTE ON [dbo].[SuaNhaXuatBan] TO [LibrarianRole];  GRANT EXECUTE ON [dbo].[XoaNhaXuatBan] TO [LibrarianRole];  GRANT EXECUTE ON [dbo].[ThemTheLoai] TO [LibrarianRole];  GRANT EXECUTE ON [dbo].[SuaTheLoai] TO [LibrarianRole];  GRANT EXECUTE ON [dbo].[XoaTheLoai] TO [LibrarianRole];  -- Proc quản lý thành viên (full EXECUTE)  GRANT EXECUTE ON [dbo].[ThemThanhVien] TO [LibrarianRole];  GRANT EXECUTE ON [dbo].[SuaThanhVien] TO [LibrarianRole];  GRANT EXECUTE ON [dbo].[XoaThanhVien] TO [LibrarianRole];  -- Proc quản lý mượn/trả/đặt trước (full EXECUTE)  GRANT EXECUTE ON [dbo].[MuonSachProc] TO [LibrarianRole];  GRANT EXECUTE ON [dbo].[TraSachProc] TO [LibrarianRole];  GRANT EXECUTE ON [dbo].[GiaHanMuon] TO [LibrarianRole];  GRANT EXECUTE ON [dbo].[DatTruoc] TO [LibrarianRole];  GRANT EXECUTE ON [dbo].[ImportSachBatch] TO [LibrarianRole];  GRANT EXECUTE ON [dbo].[TimKiemSach] TO [LibrarianRole];  GRANT EXECUTE ON [dbo].[UpdateOverdueStatusProc] TO [LibrarianRole];  -- Proc quản lý phạt/thanh toán (full EXECUTE)  GRANT EXECUTE ON [dbo].[ThanhToanPhat] TO [LibrarianRole];  -- Proc báo cáo/thống kê/dashboard (full EXECUTE)  GRANT EXECUTE ON [dbo].[sp\_DashboardSummary] TO [LibrarianRole];  GRANT EXECUTE ON [dbo].[sp\_DashboardOverdue] TO [LibrarianRole];  GRANT EXECUTE ON [dbo].[sp\_DashboardTrends] TO [LibrarianRole];  GRANT EXECUTE ON [dbo].[sp\_GetThongTinThanhVienMuonTra] TO [LibrarianRole];  GRANT EXECUTE ON [dbo].[sp\_ThongKeTinhTrangSach] TO [LibrarianRole];  GRANT EXECUTE ON [dbo].[sp\_TongTienPhatTheoNam] TO [LibrarianRole];  GRANT EXECUTE ON [dbo].[sp\_TongTienPhatTheoQuy] TO [LibrarianRole];  GRANT EXECUTE ON [dbo].[sp\_TongTienPhatTheoThang] TO [LibrarianRole];  GRANT EXECUTE ON [dbo].[sp\_TongTienPhatTheoTuan] TO [LibrarianRole];  GRANT EXECUTE ON [dbo].[sp\_TopSachMuonTheoNgay] TO [LibrarianRole];  -- =====================================================  -- 5. DENY EXECUTE TRÊN PROC LIÊN QUAN USERS/CAUHINH/ROLES  -- =====================================================  -- Proc quản lý Users - KHÔNG EXECUTE  DENY EXECUTE ON [dbo].[ThemUsers] TO [LibrarianRole];  DENY EXECUTE ON [dbo].[SuaUsers] TO [LibrarianRole];  DENY EXECUTE ON [dbo].[XoaUsers] TO [LibrarianRole];  -- Proc quản lý Cấu hình - KHÔNG EXECUTE (chỉ SELECT bảng, không update)  DENY EXECUTE ON [dbo].[sp\_CapNhatCauHinhHeThong] TO [LibrarianRole];  -- Proc sync security (nếu có) - KHÔNG EXECUTE  DENY EXECUTE ON [dbo].[SyncUserToDBSecurity] TO [LibrarianRole];  GO  -- =====================================================  -- 6. QUYỀN TRÊN USER-DEFINED FUNCTIONS (EXECUTE - tương đương gọi)  -- =====================================================  GRANT EXECUTE ON [dbo].[fn\_CountOnLoanByMember] TO [LibrarianRole];  GRANT EXECUTE ON [dbo].[fn\_TinhPhat] TO [LibrarianRole];  GRANT EXECUTE ON [dbo].[RTM\_GetBorrowingStatsByMemberType] TO [LibrarianRole];  GRANT EXECUTE ON [dbo].[TinhPhatQuaHan] TO [LibrarianRole];  GRANT EXECUTE ON [dbo].[RTO\_GetOverdueBooksByMember] TO [LibrarianRole];  GO  -- =====================================================  -- 7. QUYỀN TRÊN USER-DEFINED TABLE TYPES (EXECUTE - sử dụng trong proc)  -- =====================================================  GRANT EXECUTE ON TYPE::[dbo].[ListBanSao] TO [LibrarianRole];  GO |

### 1.2. Quản trị viên (Admin)

|  |
| --- |
| CREATE ROLE [AdminRole];  GO  -- =====================================================  -- 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)  GRANT SELECT, INSERT, UPDATE, DELETE, EXECUTE ON DATABASE::QuanLyThuVien TO [AdminRole];  GRANT CONTROL ON SCHEMA::dbo TO [AdminRole];  GO |

## 2. Đồng bộ giữa CRUD Người dùng (Users) và UserLogin của database

Stored Procedure đồng bộ giữa thao tác CRUD Người dùng và UsersLogin trong database, được thực hiện sau các Stored Procedure CRUD

|  |
| --- |
| CREATE PROCEDURE [dbo].[SyncUserToDBSecurity]  @UserName NVARCHAR(100),  @PlainPassword NVARCHAR(300),  @RoleID INT,  @Action NVARCHAR(20) = 'INSERT'  AS  BEGIN  SET NOCOUNT ON;  DECLARE @SQL NVARCHAR(MAX);    BEGIN TRY  BEGIN TRANSACTION;    IF @RoleID NOT IN (1, 2)  BEGIN  RAISERROR(N'RoleID phải là 1 (Admin) hoặc 2 (Librarian)', 16, 1);  RETURN;  END    IF @Action IN ('INSERT', 'UPDATE')  BEGIN  IF NOT EXISTS (SELECT 1 FROM master.sys.sql\_logins WHERE name = @UserName)  BEGIN  SET @SQL = N'USE master; CREATE LOGIN [' + @UserName + N'] WITH PASSWORD = ''' + REPLACE(@PlainPassword, '''', '''''') + N''', DEFAULT\_DATABASE = [QuanLyThuVien], CHECK\_EXPIRATION = OFF, CHECK\_POLICY = OFF';  EXEC sp\_executesql @SQL;  END  ELSE  BEGIN  SET @SQL = N'USE master; ALTER LOGIN [' + @UserName + N'] WITH PASSWORD = ''' + REPLACE(@PlainPassword, '''', '''''') + N'''';  EXEC sp\_executesql @SQL;  END    SET @SQL = N'USE [QuanLyThuVien];  IF EXISTS (SELECT 1 FROM sys.database\_principals WHERE name = ''' + @UserName + N''')  DROP USER [' + @UserName + N'];  CREATE USER [' + @UserName + N'] FOR LOGIN [' + @UserName + N'];';  EXEC sp\_executesql @SQL;    SET @SQL = N'USE [QuanLyThuVien]; IF EXISTS (SELECT 1 FROM sys.database\_role\_members rm JOIN sys.database\_principals dp ON rm.member\_principal\_id = dp.principal\_id WHERE dp.name = ''' + @UserName + N''') ALTER ROLE [AdminRole] DROP MEMBER [' + @UserName + N']';  EXEC sp\_executesql @SQL;    SET @SQL = N'USE [QuanLyThuVien]; IF EXISTS (SELECT 1 FROM sys.database\_role\_members rm JOIN sys.database\_principals dp ON rm.member\_principal\_id = dp.principal\_id WHERE dp.name = ''' + @UserName + N''') ALTER ROLE [LibrarianRole] DROP MEMBER [' + @UserName + N']';  EXEC sp\_executesql @SQL;    IF @RoleID = 1  SET @SQL = N'USE [QuanLyThuVien]; ALTER ROLE [AdminRole] ADD MEMBER [' + @UserName + N']';  ELSE  SET @SQL = N'USE [QuanLyThuVien]; ALTER ROLE [LibrarianRole] ADD MEMBER [' + @UserName + N']';    EXEC sp\_executesql @SQL;  END  ELSE IF @Action = 'DELETE'  BEGIN  DECLARE @CurrentSessionID INT = @@SPID;  DECLARE @SessionID INT;  DECLARE @KillSQL NVARCHAR(100);    CREATE TABLE #SessionsToKill (SessionID INT);    INSERT INTO #SessionsToKill (SessionID)  SELECT session\_id  FROM sys.dm\_exec\_sessions  WHERE login\_name = @UserName  AND is\_user\_process = 1  AND session\_id != @CurrentSessionID;    DECLARE kill\_cursor CURSOR FOR  SELECT SessionID FROM #SessionsToKill;    OPEN kill\_cursor;  FETCH NEXT FROM kill\_cursor INTO @SessionID;    WHILE @@FETCH\_STATUS = 0  BEGIN  SET @KillSQL = N'KILL ' + CAST(@SessionID AS NVARCHAR(10));  EXEC sp\_executesql @KillSQL;  FETCH NEXT FROM kill\_cursor INTO @SessionID;  END;    CLOSE kill\_cursor;  DEALLOCATE kill\_cursor;    DROP TABLE #SessionsToKill;    WAITFOR DELAY '00:00:02';    SET @SQL = N'USE [QuanLyThuVien]; IF EXISTS (SELECT 1 FROM sys.database\_role\_members rm JOIN sys.database\_principals dp ON rm.member\_principal\_id = dp.principal\_id WHERE dp.name = ''' + @UserName + N''') ALTER ROLE [AdminRole] DROP MEMBER [' + @UserName + N']';  EXEC sp\_executesql @SQL;    SET @SQL = N'USE [QuanLyThuVien]; IF EXISTS (SELECT 1 FROM sys.database\_role\_members rm JOIN sys.database\_principals dp ON rm.member\_principal\_id = dp.principal\_id WHERE dp.name = ''' + @UserName + N''') ALTER ROLE [LibrarianRole] DROP MEMBER [' + @UserName + N']';  EXEC sp\_executesql @SQL;    SET @SQL = N'USE [QuanLyThuVien]; IF EXISTS (SELECT 1 FROM sys.database\_principals WHERE name = ''' + @UserName + N''') DROP USER [' + @UserName + N']';  EXEC sp\_executesql @SQL;      SET @SQL = N'USE master; IF EXISTS (SELECT 1 FROM sys.sql\_logins WHERE name = ''' + @UserName + N''') DROP LOGIN [' + @UserName + N']';  EXEC sp\_executesql @SQL;  END    DECLARE @AuditDetails NVARCHAR(500);  SET @AuditDetails = N'Action=' + @Action + N'; RoleID=' + CAST(@RoleID AS NVARCHAR(10));    INSERT INTO [dbo].[AuditLog] (UserName, Action, EntityName, EntityID, Details, CreatedAt)  VALUES (  ISNULL(SUSER\_NAME(), 'SYSTEM'),  'SyncUserToDBSecurity',  'UserSecurity',  @UserName,  @AuditDetails,  GETDATE()  );    COMMIT TRANSACTION;    END TRY  BEGIN CATCH  IF @@TRANCOUNT > 0  ROLLBACK TRANSACTION;  DECLARE @ErrorMsg NVARCHAR(4000) = ERROR\_MESSAGE();  RAISERROR(@ErrorMsg, 16, 1);  END CATCH  END; |

C#:

|  |
| --- |
| public bool SyncUser(string userName, string plainPassword, int roleId, string action)  {  try  {  using (var connection = new SqlConnection(\_sysAdminConStr))  {  using (var command = new SqlCommand("[dbo].[SyncUserToDBSecurity]", connection))  {  command.CommandType = CommandType.StoredProcedure;  command.CommandTimeout = 30;  command.Parameters.AddWithValue("@UserName", userName);  command.Parameters.AddWithValue("@RoleID", roleId);  command.Parameters.AddWithValue("@Action", action);  if (action == "INSERT" || action == "UPDATE")  {  if (string.IsNullOrEmpty(plainPassword))  throw new ArgumentException("Plain password bắt buộc cho INSERT/UPDATE");  command.Parameters.AddWithValue("@PlainPassword", plainPassword);  }  else  {  command.Parameters.AddWithValue("@PlainPassword", (object)DBNull.Value);  }  connection.Open();  int result = command.ExecuteNonQuery();  return result > 0;  }  }  }  catch (SqlException sqlEx)  {  string errorMsg = $"SQL Error #{sqlEx.Number}: {sqlEx.Message} | User: {userName} | Action: {action}";  System.Diagnostics.Debug.WriteLine(errorMsg);  return false;  }  catch (Exception ex)  {  System.Diagnostics.Debug.WriteLine($"Sync Error: {ex.Message} | User: {userName}");  return false;  }  } |

# 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#.

- Framework hỗ trợ data access: Entity Framework 6.

- Công cụ giao diện người dùng: DevExpress 22

- Công cụ lập trình: Microsoft Visual Studio 2022

## 1.1 Cấu trúc thư mục hệ thống

Thư mục hệ thống tổng thể của dự án bao gồm mô hình 3 lớp UI, BLL, DAL.

A screenshot of a computer

AI-generated content may be incorrect.

Thư mục lớp UI

A screenshot of a computer program

AI-generated content may be incorrect.

Thư mục lớp BLL

A screenshot of a computer program

AI-generated content may be incorrect.

Thư mục lớp DAL

A screenshot of a computer

AI-generated content may be incorrect.

## 1.2. Mô hình 3 lớp (3 layers)

Mô hình 3 lớp (Three-Layer Architecture) là một kiến trúc phần mềm phổ biến được sử dụng để phát triển các ứng dụng, giúp tách biệt các thành phần của ứng dụng thành ba lớp riêng biệt, mỗi lớp có trách nhiệm riêng, giúp mã nguồn dễ bảo trì, mở rộng và kiểm thử.

**A diagram of layers of data

AI-generated content may be incorrect.**

Lợi ích của mô hình 3 lớp:

* Tách biệt trách nhiệm: Mỗi lớp có một trách nhiệm riêng, giúp mã nguồn dễ bảo trì và mở rộng.
* Tái sử dụng mã: Các lớp có thể được tái sử dụng trong các dự án khác.
* Dễ kiểm thử: Các lớp có thể được kiểm thử độc lập, giúp phát hiện và sửa lỗi dễ dàng hơn.

Mô hình 3 lớp là một kiến trúc mạnh mẽ và linh hoạt, giúp phát triển các ứng dụng WinForms một cách hiệu quả và dễ dàng quản lý.

### 1.2.1. Lớp giao diện UI (User Interface)

Lớp này chịu trách nhiệm hiển thị giao diện người dùng và xử lý các tương tác của người dùng. Trong ứng dụng WinForms, lớp này bao gồm các form, control và các thành phần giao diện khác.

Chức năng chính: Hiển thị dữ liệu cho người dùng và nhận đầu vào từ người dùng.

### 1.2.2. Lớp nghiệp vụ BLL (Business Logic Layer)

Lớp này chứa các logic nghiệp vụ của ứng dụng. Nó thực hiện các thao tác CRUD (Create, Read, Update, Delete). Nó xử lý các quy tắc nghiệp vụ, tính toán và các thao tác liên quan đến dữ liệu mà không liên quan trực tiếp đến giao diện người dùng hoặc cơ sở dữ liệu.

Chức năng chính: Xử lý logic nghiệp vụ và các quy tắc nghiệp vụ.

### 1.2.3. Lớp truy cập dữ liệu DAL (Data Acess Láyer)

Lớp này chịu trách nhiệm tương tác và quản lý kết nối với cơ sở dữ liệu.

Chức năng chính: Truy xuất và lưu trữ dữ liệu từ/đến cơ sở dữ liệu.

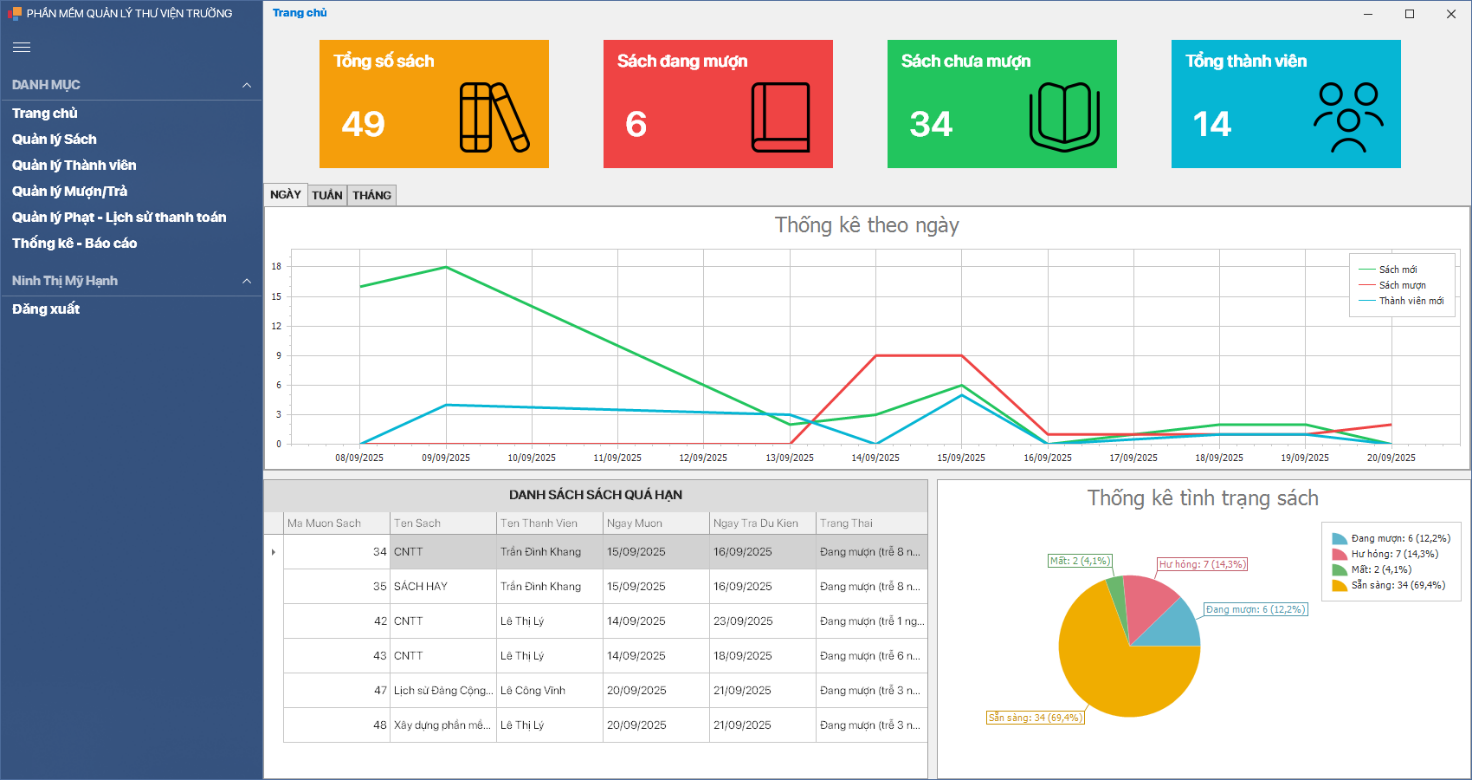
# 2. Giao diện của chương trình:

## 2.1. Form đăng nhập

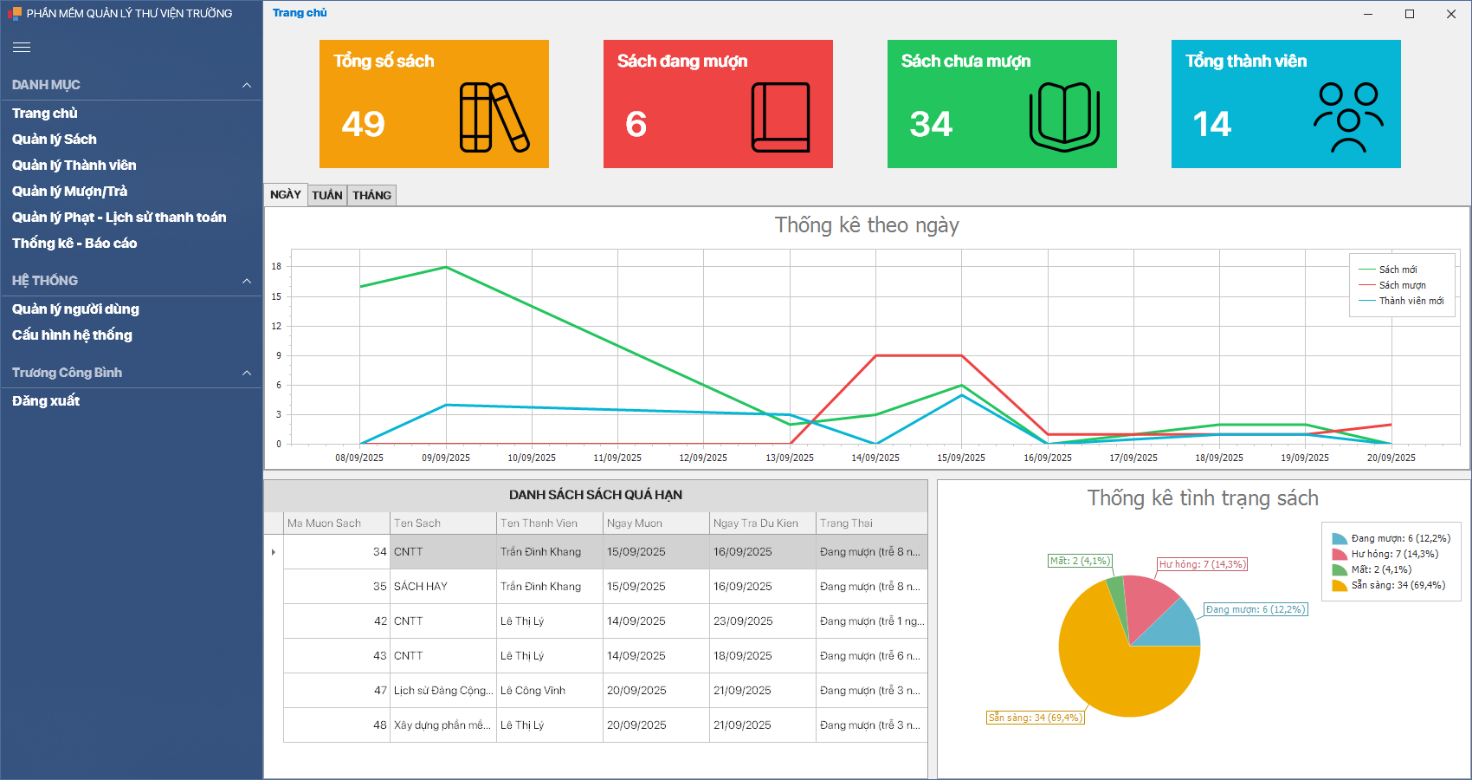


## 2.2. Form chính – Danh mục Trang chủ

- Đối với Thủ thư

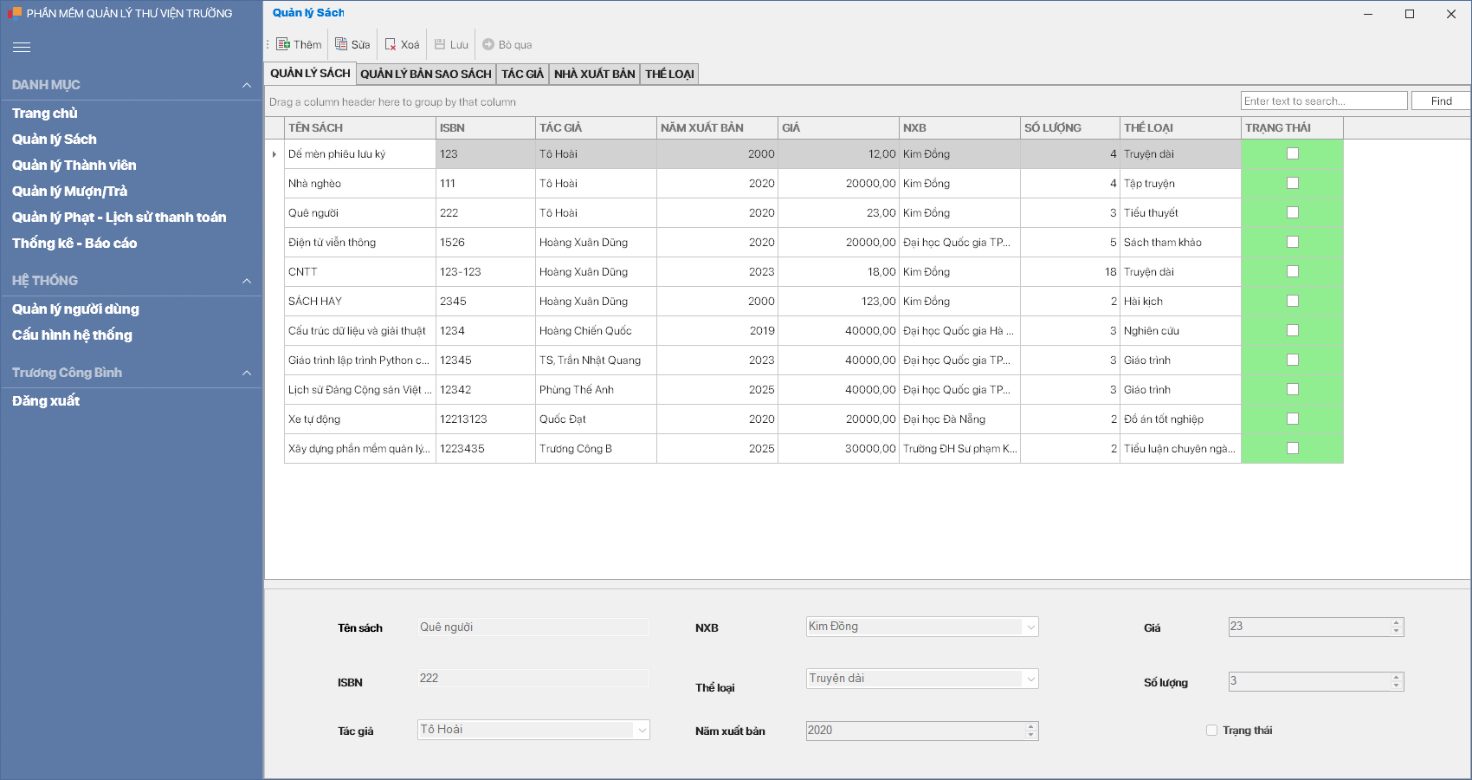


- Đối với Quản trị viên

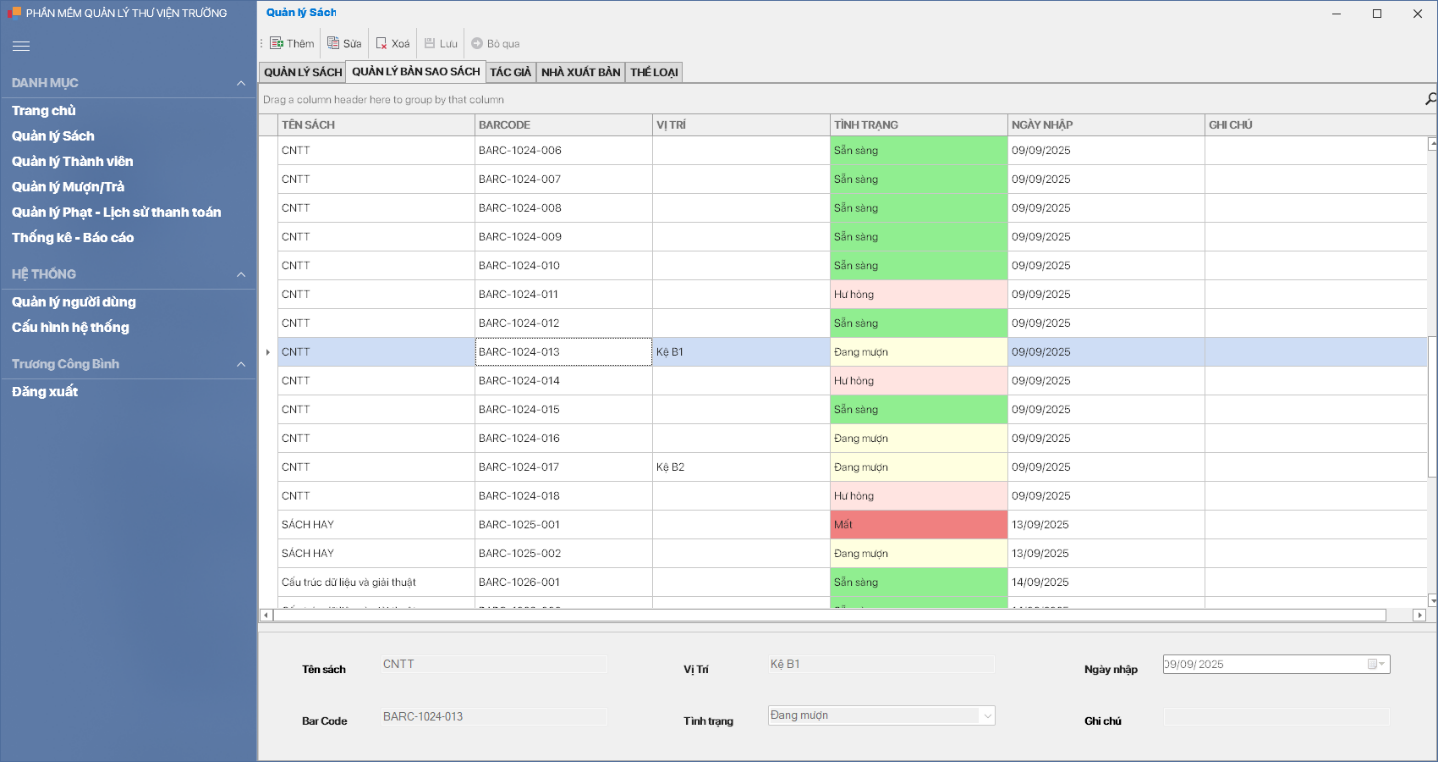


## 2.3. Danh mục Quản lý Sách

### 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.

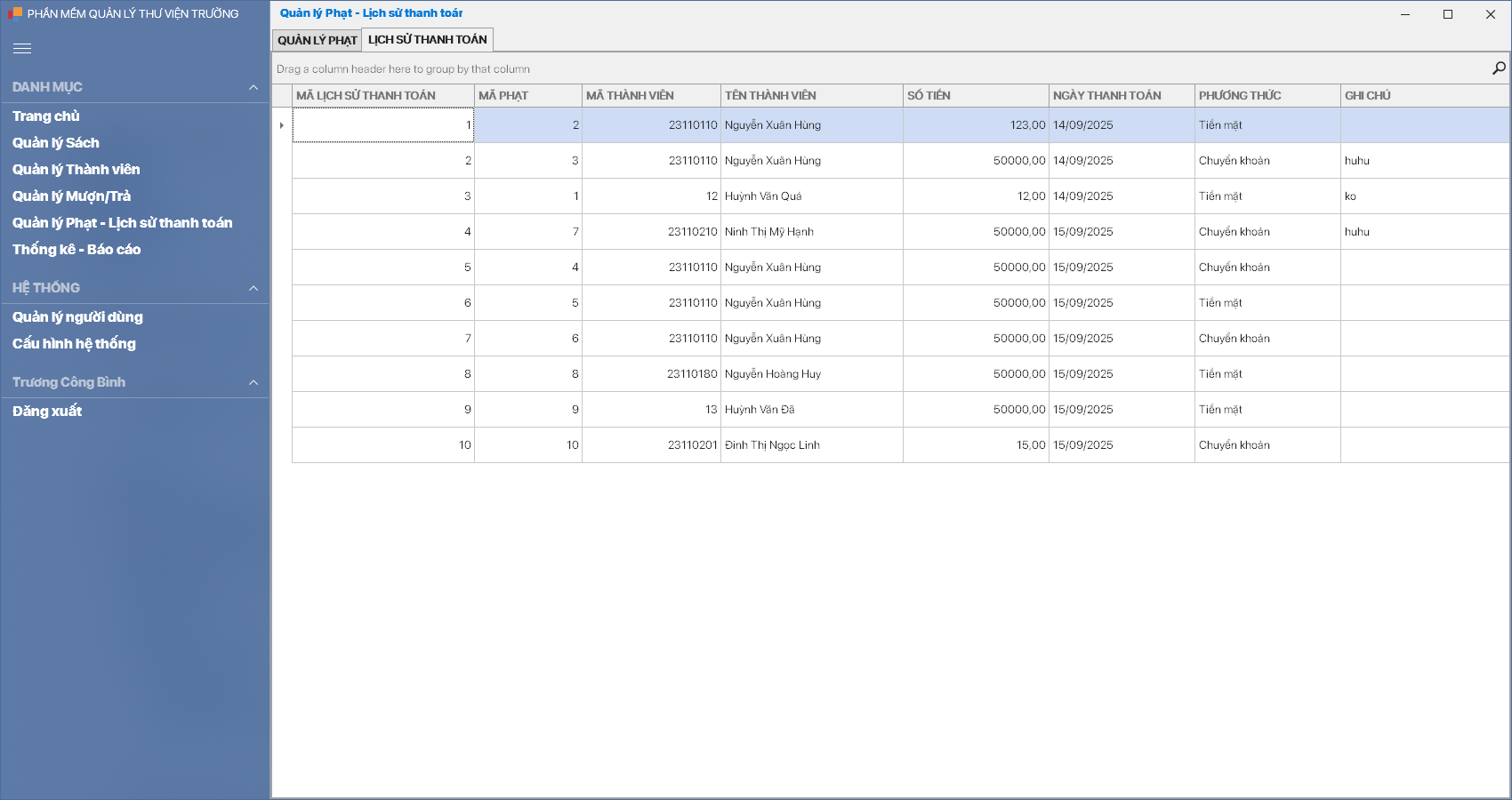
## Danh mục Quản lý Phạt – Lịch sử thanh toán

### 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.