|  |
| --- |
| FPT POLYTECHNIC |
| **DỰ ÁN 1** |
| NGÀNH CÔNG NGHỆ THÔNG TIN (ỨNG DỤNG PHẦN MỀM) |



|  |
| --- |
| TPHCM 2025 |

|  |
| --- |
| GIẢNG VIÊN HƯỚNG DẪN: THẦY LẠI MINH ĐĂNG |
| SINH VIÊN THỰC HIỆN |
| 1. Nguyễn Anh Tú (PS12345), Trưởng nhóm 2. Nguyễn Duy Phương 3. Phạm Thành Khôi 4. Hồ Nhật Hạ 5. Nguyễn Lê Quốc Thắng |

MỤC LỤC

[1 Phân tích 2](#_Toc520132211)

[1.1 Hiện trạng 2](#_Toc520132212)

[1.2 Yêu cầu hệ thống 2](#_Toc520132213)

[1.3 Use case 2](#_Toc520132214)

[2 Thiết kế 2](#_Toc520132215)

[2.1 Mô hình triển khai 2](#_Toc520132216)

[2.2 Thiết kế CSDL 2](#_Toc520132217)

[2.2.1 Sơ đồ quan hệ thực thể 2](#_Toc520132218)

[2.2.2 Thiết kế chi tiết các thực thể 2](#_Toc520132219)

[2.3 Thiết kế giao diện 2](#_Toc520132220)

[2.3.1 Sơ đồ tổ chức giao diện 2](#_Toc520132221)

[2.3.2 Thiết kế giao diện cho các chức năng nghiệp vụ 2](#_Toc520132222)

[3 Thực hiện viết mã 2](#_Toc520132223)

[3.1 Viết mã tạo CSDL 2](#_Toc520132224)

[3.1.1 Tạo CSDL 2](#_Toc520132225)

[3.1.2 SQL truy vấn và thao tác 2](#_Toc520132226)

[3.1.3 Các thủ tục lưu tổng hợp thống kê 2](#_Toc520132227)

[3.2 Lập trình JDBC 3](#_Toc520132228)

[3.2.1 Lớp hỗ trợ 3](#_Toc520132229)

[3.2.2 Model class - Các lớp mô tả dữ liệu 3](#_Toc520132230)

[3.2.3 DAO Class - Các lớp truy xuất dữ liệu 3](#_Toc520132231)

[3.3 Viết mã cho ứng dụng 3](#_Toc520132232)

[3.3.1 Xử lý Form X 3](#_Toc520132233)

[3.3.2 Xử lý Form Y 3](#_Toc520132234)

[4 Kiểm thử 3](#_Toc520132235)

[4.1 Kiểm thử form X 3](#_Toc520132236)

[4.2 Kiểm thử form Y 3](#_Toc520132237)

[5 Đóng gói và triển khai 3](#_Toc520132238)

[5.1 Hướng dẫn chuyển đổi jar thành exe 3](#_Toc520132239)

[5.2 Hướng dẫn cài đặt triển khai 3](#_Toc520132240)

[5.3 Hướng dẫn sử dụng phần mềm 3](#_Toc520132241)

# Phân tích

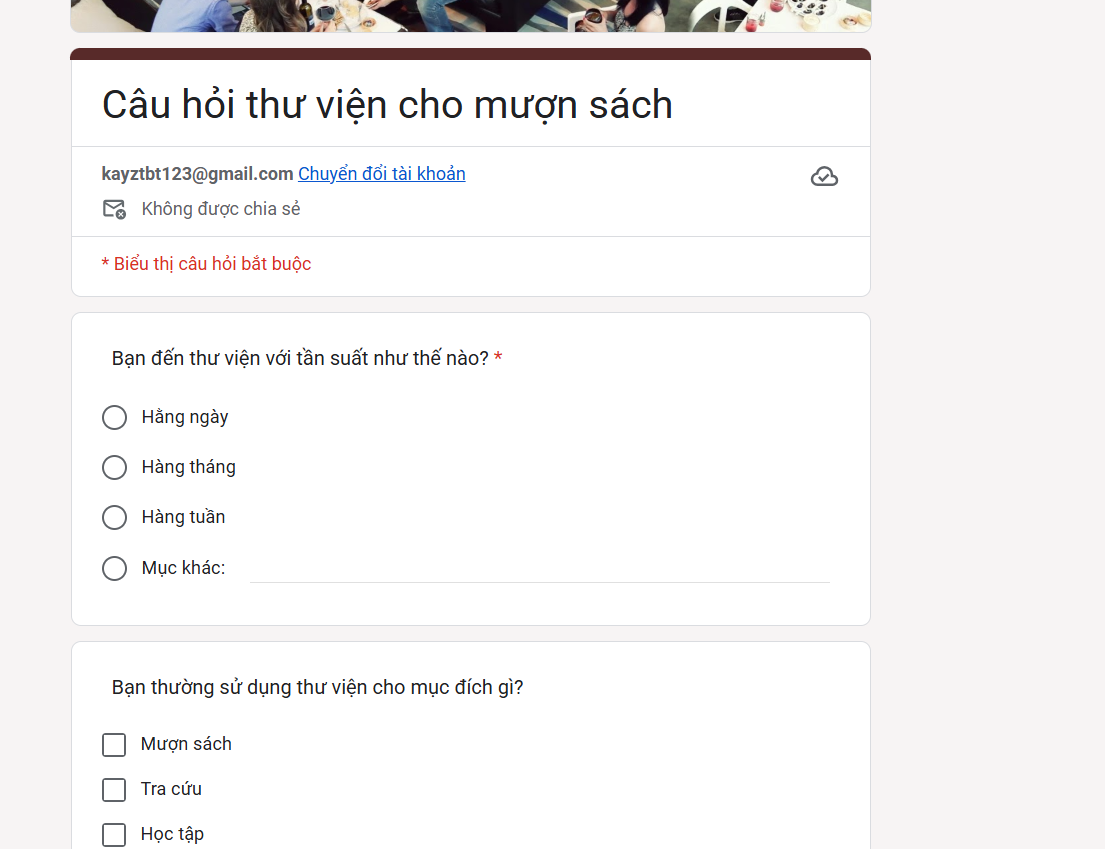
## Hiện trạng

Hiện tại, công tác quản lý thư viện tại trường học được thực hiện chủ yếu bằng phương pháp **thủ công (sổ sách)** kết hợp với các công cụ tin học cơ bản (như Microsoft Excel), dẫn đến nhiều hạn chế và sai sót trong quá trình vận hành.

* **1. Quản lý Danh mục Sách và Kho**
* **Sử dụng File Excel và Sổ sách:** Danh mục đầu sách, số lượng tồn kho, và thông tin vị trí sách (trên kệ) được lưu trữ rải rác trong nhiều file Excel và sổ ghi chép.
  + **Vấn đề:** Dữ liệu **không đồng bộ**, nhân viên thư viện phải cập nhật thủ công khi có sách mới hoặc sách bị mất/hỏng. Điều này dễ dẫn đến **sai lệch lớn** giữa số liệu trong hồ sơ và số lượng sách vật lý thực tế.
* **Tìm kiếm Chậm:** Việc tìm kiếm sách theo Tác giả hoặc Thể loại phải thông qua thao tác lọc thủ công trong file Excel hoặc tra cứu mục lục giấy, gây **mất thời gian** cho cả độc giả và nhân viên, không đáp ứng được nhu cầu phục vụ nhanh chóng.
* **Kiểm kê Tốn kém:** Hoạt động kiểm kê sách định kỳ (cuối kỳ/cuối năm) là một quy trình **tốn thời gian và công sức**, do phải so sánh thủ công từng bản ghi với từng cuốn sách trên kệ.
* **2. Quy trình Mượn và Trả Sách**
* **Ghi nhận Thủ công:** Mọi giao dịch mượn/trả sách đều được **ghi chép bằng tay** vào thẻ mượn của độc giả và sổ theo dõi chung.
  + **Vấn đề:** Dễ gây nhầm lẫn khi ghi chép thông tin (Mã sách, Ngày trả), và khó khăn trong việc **đọc lại chữ viết tay** của các nhân viên khác nhau.
* **Kiểm soát Giới hạn và Trễ hạn yếu:** Việc kiểm tra xem một độc giả có đang nợ sách hoặc đã đạt **giới hạn mượn tối đa** hay chưa đòi hỏi nhân viên phải tra cứu lại hồ sơ hoặc sổ sách, làm chậm quá trình mượn.
* **Tính Phí phạt:** Việc tính toán **phí phạt** cho sách trả trễ hoàn toàn được thực hiện bằng tay theo quy định, dễ xảy ra sai sót trong tính toán số ngày và số tiền phải nộp.
* **3. Quản lý Độc giả và Báo cáo**
* **Thiếu Hồ sơ Tập trung:** Thông tin độc giả (Học sinh/Giáo viên) thường được quản lý ở một hệ thống khác (hoặc một file Excel riêng), việc đối chiếu và cập nhật trạng thái độc giả (ví dụ: cấm mượn do nợ sách) là **không kịp thời**.
* **Khó khăn trong Báo cáo:** Hiện tại, thư viện **không có khả năng tạo các báo cáo tổng hợp nhanh chóng** và chính xác, như:
  + Thống kê độc giả nợ sách quá hạn.
  + Báo cáo chi tiết về tổng số lượng sách đang lưu thông.
  + Thống kê sách bị mất, hỏng.
  + Các báo cáo này thường phải tổng hợp thủ công, gây khó khăn cho việc đánh giá hiệu quả hoạt động và ra quyết định.
* **4. Vấn đề về Bảo mật và Khả năng truy cập**
* **Rủi ro Mất Dữ liệu:** Toàn bộ dữ liệu (sổ sách, file Excel) đều có nguy cơ bị mất, hư hỏng do tác động vật lý hoặc lỗi kỹ thuật, và **khó khôi phục** nếu không có quy trình sao lưu chuẩn.
* **Không có Phân quyền:** Dữ liệu có thể bị truy cập hoặc chỉnh sửa bởi bất kỳ ai có quyền tiếp cận file Excel, **thiếu tính bảo mật và trách nhiệm**.

## Yêu cầu hệ thống

* **I. Yêu cầu Chức năng (Functional Requirements)**
* **Form khảo sát:**



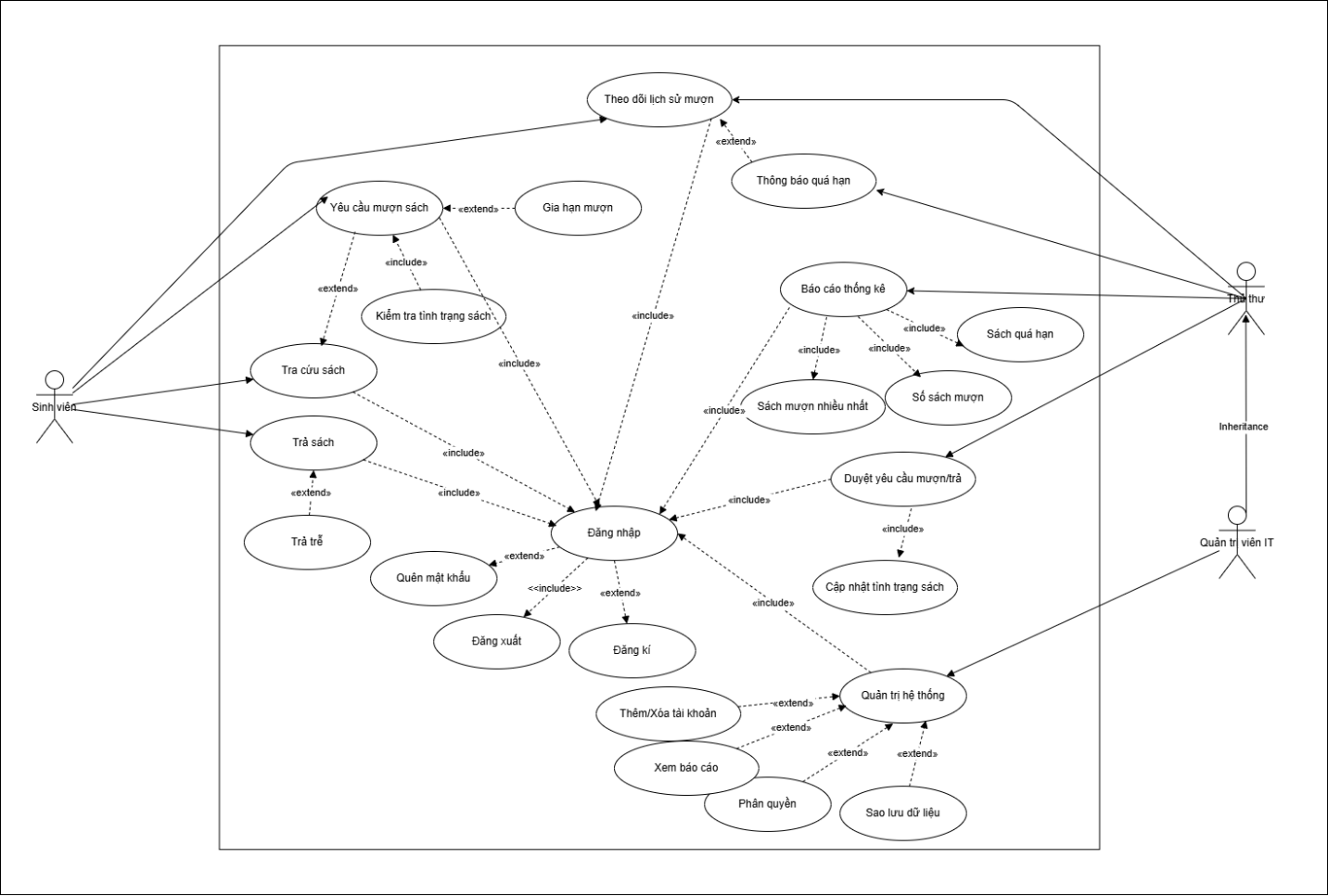
Đây là các tính năng mà phần mềm **phải làm được**.

* **1. Quản lý Danh mục (Catalog Management)**
* **FR1.1 (Thêm/Sửa/Xóa Sách):** Hệ thống phải cho phép nhân viên thư viện thêm mới, sửa đổi thông tin (tiêu đề, tác giả, ISBN, nhà xuất bản, số lượng, thể loại), và xóa bỏ đầu sách khỏi danh mục.
* **FR1.2 (Tìm kiếm):** Hệ thống phải hỗ trợ tìm kiếm sách theo Tiêu đề, Tác giả, ISBN, và Thể loại với tốc độ phản hồi dưới 2 giây.
* **FR1.3 (Quản lý Đầu mục):** Hệ thống phải cho phép quản lý các danh mục khác (Tác giả, Nhà xuất bản, Thể loại).
* **2. Quản lý Độc giả (Patron Management)**
* **FR2.1 (Thêm/Sửa/Xóa Độc giả):** Hệ thống phải cho phép thêm, sửa, xóa thông tin độc giả (Học sinh/Giáo viên) bao gồm: Mã số, Tên, Lớp/Bộ môn, Địa chỉ, Số điện thoại.
* **FR2.2 (Trạng thái Độc giả):** Hệ thống phải theo dõi trạng thái độc giả (Đang hoạt động/Bị cấm) và lý do bị cấm (ví dụ: quá hạn trả sách).
* **3. Quản lý Mượn/Trả (Circulation Management)**
* **FR3.1 (Mượn Sách):** Hệ thống phải cho phép nhân viên quét mã độc giả và mã sách để tạo giao dịch mượn. Hệ thống phải kiểm tra và cảnh báo nếu độc giả đạt đến giới hạn mượn tối đa (ví dụ: 3 cuốn) hoặc sách đã hết.
* **FR3.2 (Trả Sách):** Hệ thống phải cho phép ghi nhận trả sách và tự động xóa giao dịch mượn.
* **FR3.3 (Tính Phạt):** Hệ thống phải tự động tính phí phạt dựa trên số ngày trễ hẹn so với ngày trả dự kiến. Mức phạt phải được cấu hình.
* **FR3.4 (Gia hạn):** Hệ thống phải cho phép gia hạn thời gian mượn nếu sách chưa quá hạn (có thể giới hạn số lần gia hạn).
* **4. Báo cáo & Thống kê (Reporting)**
* **FR4.1 (Thống kê Sách):** Hệ thống phải cung cấp báo cáo về số lượng sách hiện có, sách đang được mượn, sách bị mất/hỏng.
* **FR4.2 (Thống kê Độc giả):** Hệ thống phải cung cấp danh sách độc giả đang nợ sách (quá hạn) và tổng số tiền phạt chưa thanh toán.
* **FR4.3 (Lịch sử Giao dịch):** Hệ thống phải lưu trữ và cho phép tìm kiếm lịch sử mượn/trả của một độc giả cụ thể.
* **5. Quản lý Người dùng & Bảo mật (User Management & Security)**
* **FR5.1 (Đăng nhập):** Hệ thống phải yêu cầu nhân viên thư viện đăng nhập bằng Tên người dùng và Mật khẩu.
* **FR5.2 (Phân quyền):** Hệ thống phải phân quyền giữa **Quản trị viên** (có thể quản lý tài khoản người dùng và cấu hình) và **Nhân viên Thư viện** (chỉ thực hiện các thao tác mượn/trả và quản lý danh mục).
* **II. Yêu cầu Phi chức năng (Non-functional Requirements)**

Đây là các yêu cầu về **chất lượng** của phần mềm.

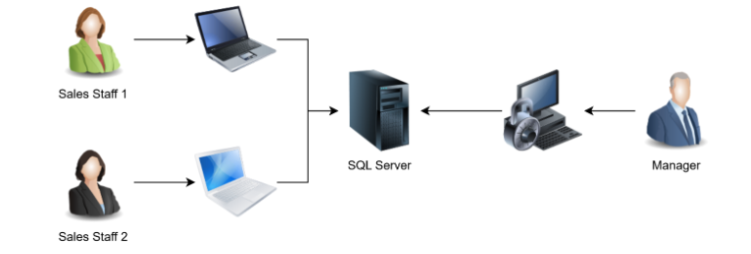
* **1. Yêu cầu Hiệu suất (Performance)**
* **NFR1.1 (Tốc độ Xử lý):** Thời gian thực hiện các giao dịch cơ bản (Mượn/Trả sách) không được vượt quá **3 giây**.
* **NFR1.2 (Khởi động):** Thời gian tải ứng dụng không quá **5 giây**.
* **2. Yêu cầu Bảo mật (Security)**
* **NFR2.1 (Mã hóa):** Mật khẩu của người dùng phải được lưu trữ dưới dạng mã hóa một chiều (ví dụ: sử dụng thuật toán Hash như SHA-256).
* **NFR2.2 (Kiểm soát Truy cập):** Chỉ người dùng đã đăng nhập mới có thể truy cập vào các chức năng của hệ thống (xem FR5.2).
* **3. Yêu cầu Tính khả dụng (Availability)**
* **NFR3.1 (Thời gian Hoạt động):** Hệ thống phải có khả năng hoạt động liên tục trong thời gian làm việc của thư viện (ví dụ: từ 7:00 sáng đến 5:00 chiều) với độ ổn định **99%**.
* **NFR3.2 (Sao lưu):** Dữ liệu phải được sao lưu tự động hàng ngày.
* **4. Yêu cầu Khả năng Mở rộng & Dễ bảo trì (Maintainability)**
* **NFR4.1 (Cấu trúc):** Phần mềm phải được xây dựng theo kiến trúc MVC để dễ dàng bảo trì và phát triển tính năng mới.
* **NFR4.2 (Giao diện):** Giao diện người dùng phải trực quan, dễ sử dụng, giảm thiểu các lỗi nhập liệu.

## Use case



# Thiết kế

## Mô hình triển khai



## Thiết kế CSDL

---------------------------------------------------------

-- TẠO DATABASE

---------------------------------------------------------

IF DB\_ID('LibraryDB') IS NOT NULL

DROP DATABASE LibraryDB;

GO

CREATE DATABASE LibraryDB;

GO

USE LibraryDB;

GO

---------------------------------------------------------

-- 1. ROLE – USERS – PERMISSIONS

---------------------------------------------------------

CREATE TABLE UserRoles (

RoleID INT IDENTITY(1,1) PRIMARY KEY,

RoleName NVARCHAR(50) UNIQUE NOT NULL,

Description NVARCHAR(200)

);

CREATE TABLE Users (

UserID INT IDENTITY(1,1) PRIMARY KEY,

Username NVARCHAR(50) UNIQUE NOT NULL,

Password NVARCHAR(100) NOT NULL,

FullName NVARCHAR(100) NOT NULL,

Email NVARCHAR(100),

RoleID INT NOT NULL,

CONSTRAINT FK\_Users\_Role FOREIGN KEY (RoleID)

REFERENCES UserRoles(RoleID)

);

CREATE TABLE SystemPermissions (

PermissionID INT IDENTITY(1,1) PRIMARY KEY,

RoleID INT NOT NULL,

PermissionName NVARCHAR(200) NOT NULL,

CONSTRAINT FK\_Permissions\_Role FOREIGN KEY (RoleID)

REFERENCES UserRoles(RoleID)

);

---------------------------------------------------------

-- 2. TÁC GIẢ – THỂ LOẠI – SÁCH

---------------------------------------------------------

CREATE TABLE Authors (

AuthorID INT IDENTITY(1,1) PRIMARY KEY,

AuthorName NVARCHAR(100) NOT NULL

);

CREATE TABLE Categories (

CategoryID INT IDENTITY(1,1) PRIMARY KEY,

CategoryName NVARCHAR(50) NOT NULL

);

CREATE TABLE Books (

BookID INT IDENTITY(1,1) PRIMARY KEY,

Title NVARCHAR(200) NOT NULL,

AuthorID INT NOT NULL,

CategoryID INT NOT NULL,

PublishedYear INT CHECK (PublishedYear >= 0),

Quantity INT NOT NULL CHECK (Quantity >= 0),

Img NVARCHAR(200),

CONSTRAINT FK\_Book\_Author FOREIGN KEY (AuthorID)

REFERENCES Authors(AuthorID),

CONSTRAINT FK\_Book\_Category FOREIGN KEY (CategoryID)

REFERENCES Categories(CategoryID)

);

---------------------------------------------------------

-- 3. MƯỢN – TRẢ – TRỄ

---------------------------------------------------------

CREATE TABLE BorrowRequest (

RequestID INT IDENTITY(1,1) PRIMARY KEY,

UserID INT NOT NULL,

BookID INT NOT NULL,

RequestDate DATETIME NOT NULL DEFAULT GETDATE(),

Status NVARCHAR(20) DEFAULT 'Pending',

Note NVARCHAR(200),

CONSTRAINT FK\_Request\_User FOREIGN KEY (UserID) REFERENCES Users(UserID),

CONSTRAINT FK\_Request\_Book FOREIGN KEY (BookID) REFERENCES Books(BookID)

);

CREATE TABLE Borrow (

BorrowID INT IDENTITY(1,1) PRIMARY KEY,

UserID INT NOT NULL,

BookID INT NOT NULL,

BorrowDate DATE NOT NULL DEFAULT GETDATE(),

ReturnDate DATE NULL,

Status NVARCHAR(20) NOT NULL DEFAULT 'Borrowed',

CONSTRAINT FK\_Borrow\_User FOREIGN KEY (UserID) REFERENCES Users(UserID),

CONSTRAINT FK\_Borrow\_Book FOREIGN KEY (BookID) REFERENCES Books(BookID)

);

CREATE TABLE ReturnLate (

LateID INT IDENTITY(1,1) PRIMARY KEY,

BorrowID INT NOT NULL,

DaysLate INT NOT NULL CHECK (DaysLate > 0),

FineAmount DECIMAL(10,2) NOT NULL CHECK (FineAmount >= 0),

CreatedAt DATETIME DEFAULT GETDATE(),

CONSTRAINT FK\_Late\_Borrow FOREIGN KEY (BorrowID) REFERENCES Borrow(BorrowID)

);

---------------------------------------------------------

-- 4. TRẠNG THÁI – THỐNG KÊ – BACKUP – LOG

---------------------------------------------------------

CREATE TABLE BookStatusHistory (

StatusID INT IDENTITY(1,1) PRIMARY KEY,

BookID INT NOT NULL,

Status NVARCHAR(50) NOT NULL,

UpdatedBy INT NOT NULL,

UpdatedAt DATETIME DEFAULT GETDATE(),

Notes NVARCHAR(200),

CONSTRAINT FK\_Status\_Book FOREIGN KEY (BookID) REFERENCES Books(BookID),

CONSTRAINT FK\_Status\_User FOREIGN KEY (UpdatedBy) REFERENCES Users(UserID)

);

CREATE TABLE Reports (

ReportID INT IDENTITY(1,1) PRIMARY KEY,

ReportType NVARCHAR(50) NOT NULL,

CreatedAt DATETIME DEFAULT GETDATE(),

CreatedBy INT NOT NULL,

Data NVARCHAR(MAX),

CONSTRAINT FK\_Report\_User FOREIGN KEY (CreatedBy) REFERENCES Users(UserID)

);

CREATE TABLE SystemBackup (

BackupID INT IDENTITY(1,1) PRIMARY KEY,

FilePath NVARCHAR(300) NOT NULL,

BackupDate DATETIME DEFAULT GETDATE(),

PerformedBy INT NOT NULL,

CONSTRAINT FK\_Backup\_User FOREIGN KEY (PerformedBy) REFERENCES Users(UserID)

);

CREATE TABLE AuditLogs (

LogID INT IDENTITY(1,1) PRIMARY KEY,

UserID INT NOT NULL,

Action NVARCHAR(200),

LogTime DATETIME DEFAULT GETDATE(),

Details NVARCHAR(MAX),

CONSTRAINT FK\_Audit\_User FOREIGN KEY (UserID) REFERENCES Users(UserID)

);

---------------------------------------------------------

-- 5. LOGIN HISTORY – QUÊN MẬT KHẨU – NOTI TRỄ

---------------------------------------------------------

CREATE TABLE LoginHistory (

LogID INT IDENTITY(1,1) PRIMARY KEY,

UserID INT NOT NULL,

LoginTime DATETIME DEFAULT GETDATE(),

IP NVARCHAR(50),

Device NVARCHAR(100),

CONSTRAINT FK\_Login\_User FOREIGN KEY (UserID) REFERENCES Users(UserID)

);

CREATE TABLE PasswordReset (

ResetID INT IDENTITY(1,1) PRIMARY KEY,

UserID INT NOT NULL,

Token NVARCHAR(200) NOT NULL,

ExpiredAt DATETIME NOT NULL,

Used BIT NOT NULL DEFAULT 0,

CONSTRAINT FK\_Reset\_User FOREIGN KEY (UserID) REFERENCES Users(UserID)

);

CREATE TABLE OverdueNotifications (

NotificationID INT IDENTITY(1,1) PRIMARY KEY,

BorrowID INT NOT NULL,

UserID INT NOT NULL,

SendDate DATETIME DEFAULT GETDATE(),

IsRead BIT DEFAULT 0,

CONSTRAINT FK\_Noti\_Borrow FOREIGN KEY (BorrowID) REFERENCES Borrow(BorrowID),

CONSTRAINT FK\_Noti\_User FOREIGN KEY (UserID) REFERENCES Users(UserID)

);

---------------------------------------------------------

-- DỮ LIỆU MẪU

---------------------------------------------------------

-- ROLES

INSERT INTO UserRoles (RoleName, Description) VALUES

('Admin', 'Quản trị hệ thống'),

('Thủ Thư', 'Quản lý mượn trả'),

('Member', 'Sinh viên mượn sách');

-- USERS

INSERT INTO Users (Username, Password, FullName, Email, RoleID) VALUES

('admin', '123456', 'Quản Trị Viên', 'admin@lib.com', 1),

('ngoc', '123456', 'Phạm Ngọc', 'ngoc@example.com', 3),

('khoi', '123456', 'Phạm Thành Khôi', 'khoi@example.com', 3),

('thuthu', '123456', 'Nguyễn Thư', 'thu@example.com', 2);

-- PERMISSIONS

INSERT INTO SystemPermissions (RoleID, PermissionName) VALUES

(1, 'Quản trị hệ thống'),

(1, 'Xem báo cáo'),

(1, 'Sao lưu dữ liệu'),

(2, 'Duyệt yêu cầu mượn'),

(2, 'Cập nhật tình trạng sách'),

(3, 'Mượn sách'),

(3, 'Trả sách');

-- AUTHORS

INSERT INTO Authors (AuthorName) VALUES

('J.K. Rowling'),

('George Orwell'),

('Dan Brown'),

('Tô Hoài'),

('Nguyên Hồng');

-- CATEGORIES

INSERT INTO Categories (CategoryName) VALUES

('Fantasy'),

('Science Fiction'),

('Thriller'),

('Văn học Việt Nam');

-- BOOKS

INSERT INTO Books (Title, AuthorID, CategoryID, PublishedYear, Quantity, Img) VALUES

('Harry Potter và Hòn Đá Phù Thủy', 1, 1, 1997, 10, 'images/harry.jpg'),

('1984', 2, 2, 1949, 5, 'images/1984.jpg'),

('Mật Mã Da Vinci', 3, 3, 2003, 7, 'images/davinci.jpg'),

('Dế Mèn Phiêu Lưu Ký', 4, 4, 1941, 12, 'images/demen.jpg'),

('Bỉ Vỏ', 5, 4, 1937, 6, 'images/bivo.jpg'),

('Harry Potter và Phòng Chứa Bí Mật', 1, 1, 1998, 8, 'images/hp2.jpg'),

('Harry Potter và Tên Tù Nhân Ngục Azkaban', 1, 1, 1999, 7, 'images/hp3.jpg'),

('Harry Potter và Chiếc Cốc Lửa', 1, 1, 2000, 10, 'images/hp4.jpg'),

('Harry Potter và Hội Phượng Hoàng', 1, 1, 2003, 12, 'images/hp5.jpg'),

('Harry Potter và Hoàng Tử Lai', 1, 1, 2005, 9, 'images/hp6.jpg'),

('Harry Potter và Bảo Bối Tử Thần', 1, 1, 2007, 11, 'images/hp7.jpg'),

('Trại Súc Vật', 2, 2, 1945, 10, 'images/traisucvat.jpg'),

('Chúa Tể Những Chiếc Nhẫn: Hiệp Hội Nhẫn', 2, 1, 1954, 6, 'images/lotr1.jpg'),

('Chúa Tể Những Chiếc Nhẫn: Hai Tòa Tháp', 2, 1, 1954, 6, 'images/lotr2.jpg'),

('Chúa Tể Những Chiếc Nhẫn: Nhà Vua Trở Lại', 2, 1, 1955, 5, 'images/lotr3.jpg'),

('Biểu Tượng Thất Truyền', 3, 3, 2009, 7, 'images/bieutuong.jpg'),

('Pháo Đài Số', 3, 3, 1998, 6, 'images/phaodaiso.jpg'),

('Thiên Thần Và Ác Quỷ', 3, 3, 2000, 9, 'images/angelsdemons.jpg'),

('Vợ Nhặt', 5, 4, 1954, 10, 'images/vonhat.jpg'),

('Lão Hạc', 5, 4, 1943, 9, 'images/laohac.jpg');

-- BORROW REQUEST

INSERT INTO BorrowRequest (UserID, BookID, Status) VALUES

(2, 1, 'Approved'),

(3, 3, 'Pending');

-- BORROW

INSERT INTO Borrow (UserID, BookID, BorrowDate, Status) VALUES

(2, 1, '2024-11-01', 'Borrowed'),

(3, 3, '2024-10-20', 'Returned');

-- LATE RETURN

INSERT INTO ReturnLate (BorrowID, DaysLate, FineAmount) VALUES

(1, 5, 25000);

-- STATUS HISTORY

INSERT INTO BookStatusHistory (BookID, Status, UpdatedBy, Notes) VALUES

(1, 'Tốt', 4, 'Kiểm tra đầu kỳ'),

(3, 'Rách nhẹ', 4, 'Cần bọc lại');

-- REPORTS

INSERT INTO Reports (ReportType, CreatedBy, Data) VALUES

('TopBorrow', 1, '{"Harry Potter":10,"1984":7}');

-- BACKUP

INSERT INTO SystemBackup (FilePath, PerformedBy)

VALUES ('backup/2024\_11\_10.bak', 1);

-- AUDIT LOGS

INSERT INTO AuditLogs (UserID, Action, Details) VALUES

(1, 'Thêm sách mới', 'Admin thêm Mật Mã Da Vinci');

-- LOGIN HISTORY

INSERT INTO LoginHistory (UserID, IP, Device) VALUES

(2, '192.168.1.5', 'Laptop Windows'),

(3, '192.168.1.10', 'Android Phone');

-- PASSWORD RESET

INSERT INTO PasswordReset (UserID, Token, ExpiredAt) VALUES

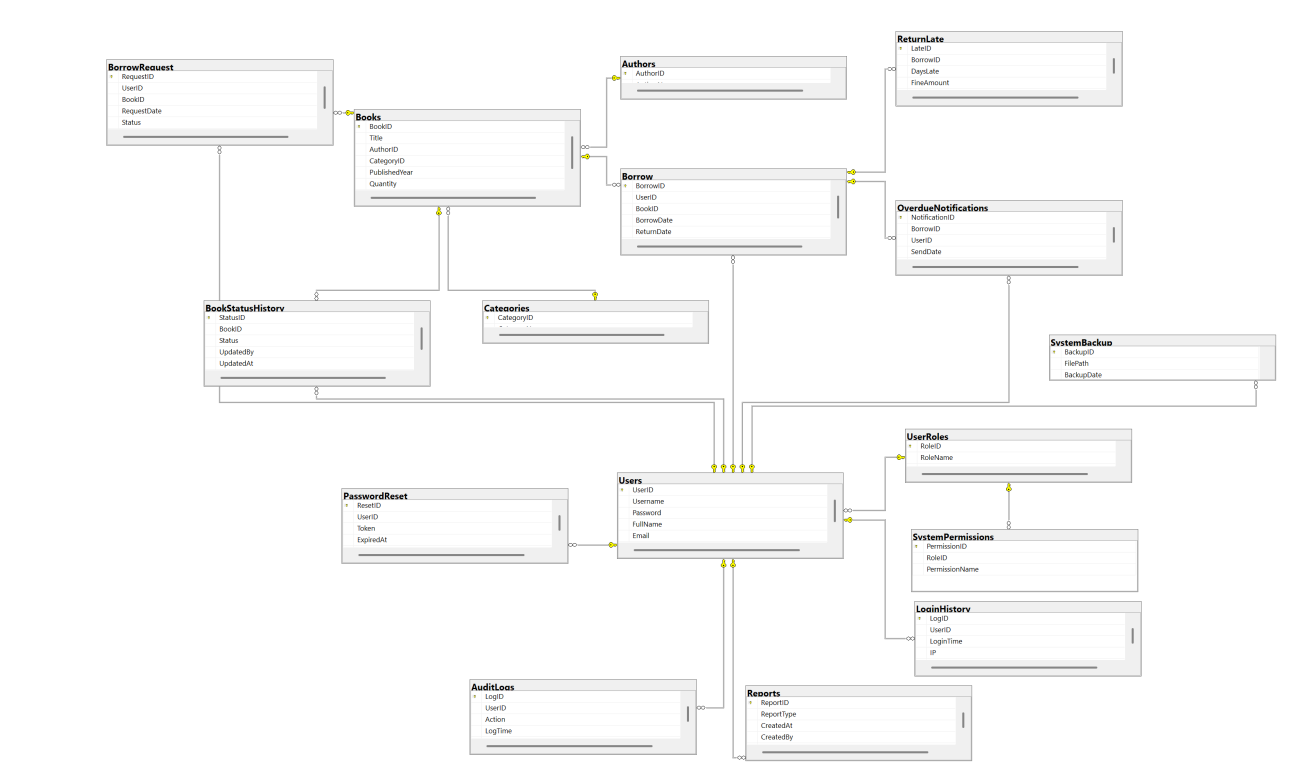
(2, 'abc123', '2024-12-01 23:59');

-- OVERDUE NOTIFICATION

INSERT INTO OverdueNotifications (BorrowID, UserID)

VALUES (1, 2);

### Sơ đồ quan hệ thực thể



### Thiết kế chi tiết các thực thể

USERS:

| **Thuộc tính** | **Kiểu Dữ liệu (Java)** | **Ràng buộc (SQL)** | **Mô tả** |
| --- | --- | --- | --- |
| **UserID** | int | PRIMARY KEY, IDENTITY | Mã định danh duy nhất của người dùng. |
| **Username** | String | UNIQUE, NOT NULL | Tên đăng nhập. |
| **Password** | String | NOT NULL | Mật khẩu (cần mã hóa khi lưu trữ). |
| **FullName** | String | NOT NULL | Họ và tên đầy đủ. |
| **Email** | String | Tùy chọn | Địa chỉ email. |
| **RoleID** | int | FOREIGN KEY | Mã vai trò (tham chiếu tới UserRoles). |

BOOKS:

| **Thuộc tính** | **Kiểu Dữ liệu (Java)** | **Ràng buộc (SQL)** | **Mô tả** |
| --- | --- | --- | --- |
| **BookID** | int | PRIMARY KEY, IDENTITY | Mã định danh duy nhất của sách. |
| **Title** | String | NOT NULL | Tiêu đề sách. |
| **AuthorID** | int | FOREIGN KEY, NOT NULL | Mã tác giả (tham chiếu tới Authors). |
| **CategoryID** | int | FOREIGN KEY, NOT NULL | Mã thể loại (tham chiếu tới Categories). |
| **PublishedYear** | int | CHECK (>= 0) | Năm xuất bản. |
| **Quantity** | int | NOT NULL, CHECK (>= 0) | Tổng số lượng bản sách hiện có. |
| **Img** | String | Tùy chọn | Đường dẫn ảnh bìa sách. |

BORROW:

| **Thuộc tính** | **Kiểu Dữ liệu (Java)** | **Ràng buộc (SQL)** | **Mô tả** |
| --- | --- | --- | --- |
| **BorrowID** | int | PRIMARY KEY, IDENTITY | Mã định danh duy nhất của giao dịch. |
| **UserID** | int | FOREIGN KEY, NOT NULL | Mã người mượn (tham chiếu tới Users). |
| **BookID** | int | FOREIGN KEY, NOT NULL | Mã sách được mượn (tham chiếu tới Books). |
| **BorrowDate** | Date/LocalDateTime | NOT NULL, DEFAULT GETDATE() | Ngày mượn. |
| **ReturnDate** | Date/LocalDateTime | NULL | Ngày trả thực tế (NULL nếu chưa trả). |
| **Status** | String | NOT NULL | Trạng thái mượn ("Borrowed", "Returned", "Overdue"). |

ReturnLate:

| **Thuộc tính** | **Kiểu Dữ liệu (Java)** | **Ràng buộc (SQL)** | **Mô tả** |
| --- | --- | --- | --- |
| **LateID** | int | PRIMARY KEY, IDENTITY | Mã định danh duy nhất. |
| **BorrowID** | int | FOREIGN KEY, NOT NULL | Mã giao dịch mượn liên quan (tham chiếu tới Borrow). |
| **DaysLate** | int | NOT NULL, CHECK (> 0) | Số ngày trễ hạn. |
| **FineAmount** | double/BigDecimal | NOT NULL, CHECK (>= 0) | Số tiền phạt. |
| **CreatedAt** | Date/LocalDateTime | DEFAULT GETDATE() | Thời điểm ghi nhận phạt. |

UserRole:

| **Thuộc tính** | **Kiểu Dữ liệu (Java)** | **Ràng buộc (SQL)** | **Mô tả** |
| --- | --- | --- | --- |
| **RoleID** | int | PRIMARY KEY, IDENTITY | Mã vai trò. |
| **RoleName** | String | UNIQUE, NOT NULL | Tên vai trò ("Admin", "Thủ Thư", "Member"). |
| **Description** | String | Tùy chọn | Mô tả vai trò. |

Author:

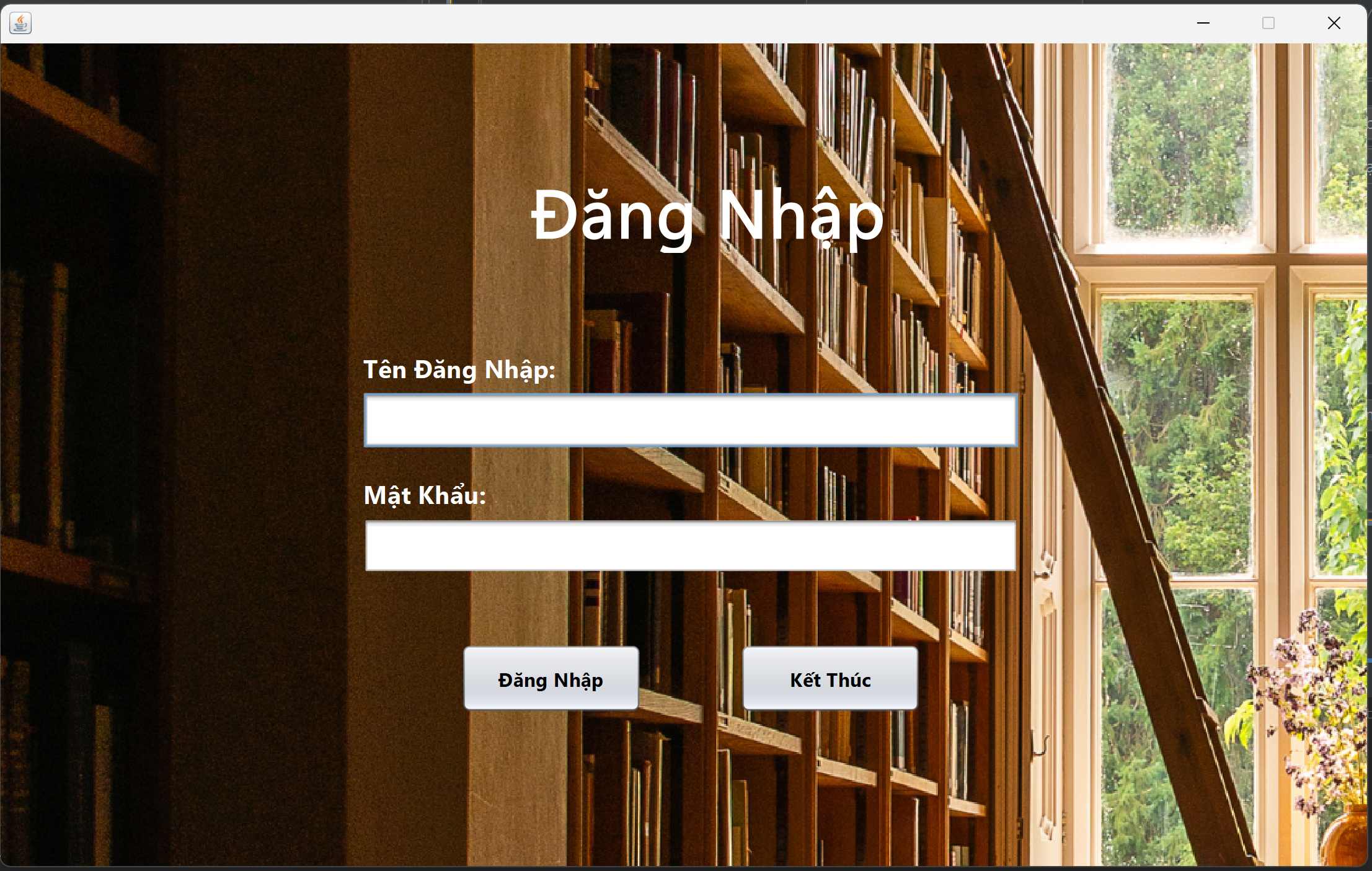
| **Thuộc tính** | **Kiểu Dữ liệu (Java)** | **Ràng buộc (SQL)** | **Mô tả** |
| --- | --- | --- | --- |
| **AuthorID** | int | PRIMARY KEY, IDENTITY | Mã tác giả. |
| **AuthorName** | String | NOT NULL | Tên tác giả. |

Category:

| **Thuộc tính** | **Kiểu Dữ liệu (Java)** | **Ràng buộc (SQL)** | **Mô tả** |
| --- | --- | --- | --- |
| **CategoryID** | int | PRIMARY KEY, IDENTITY | Mã thể loại. |
| **CategoryName** | String | NOT NULL | Tên thể loại. |

## Thiết kế giao diện

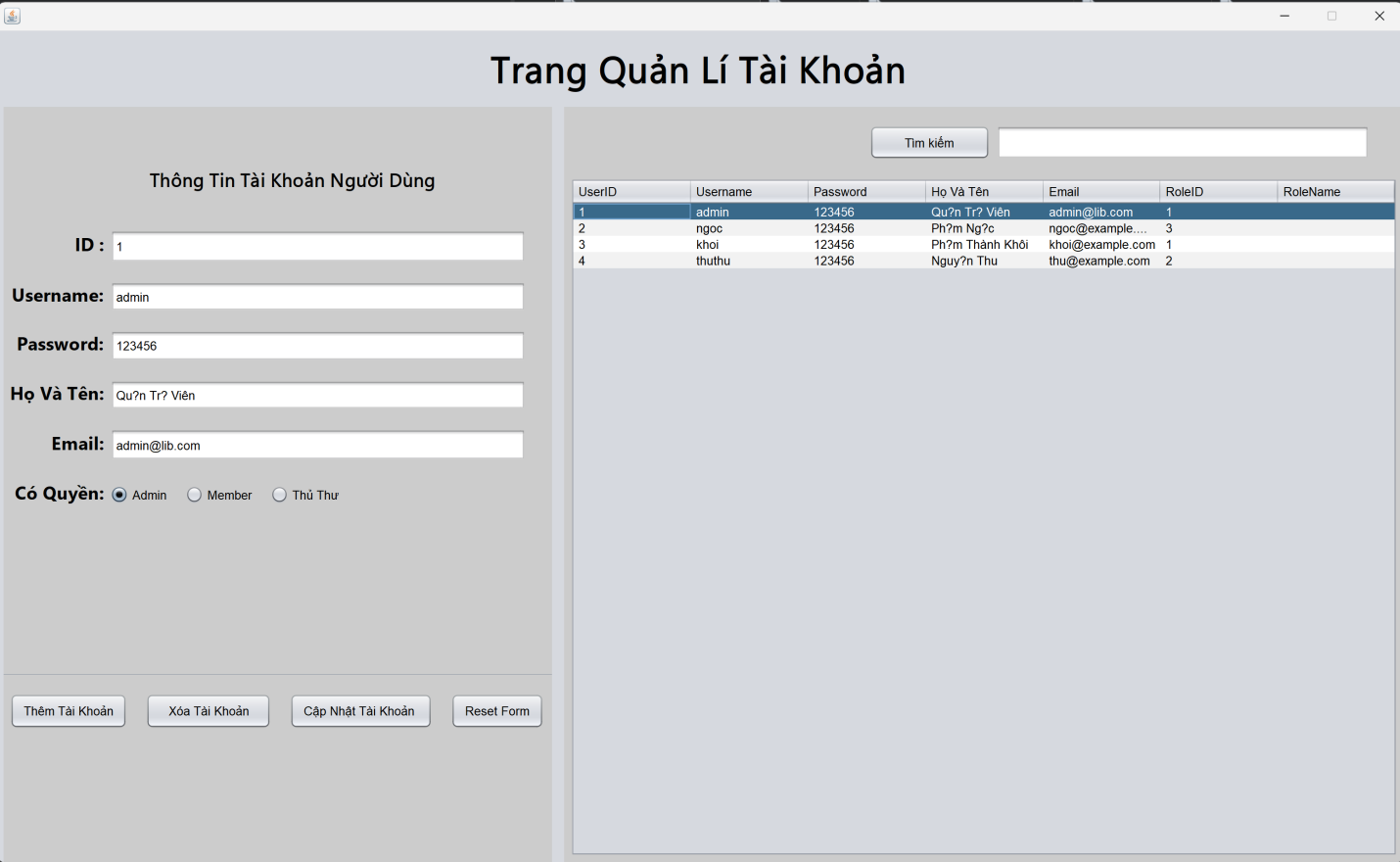
FORM ĐĂNG NHẬP (LOGIN FORM):



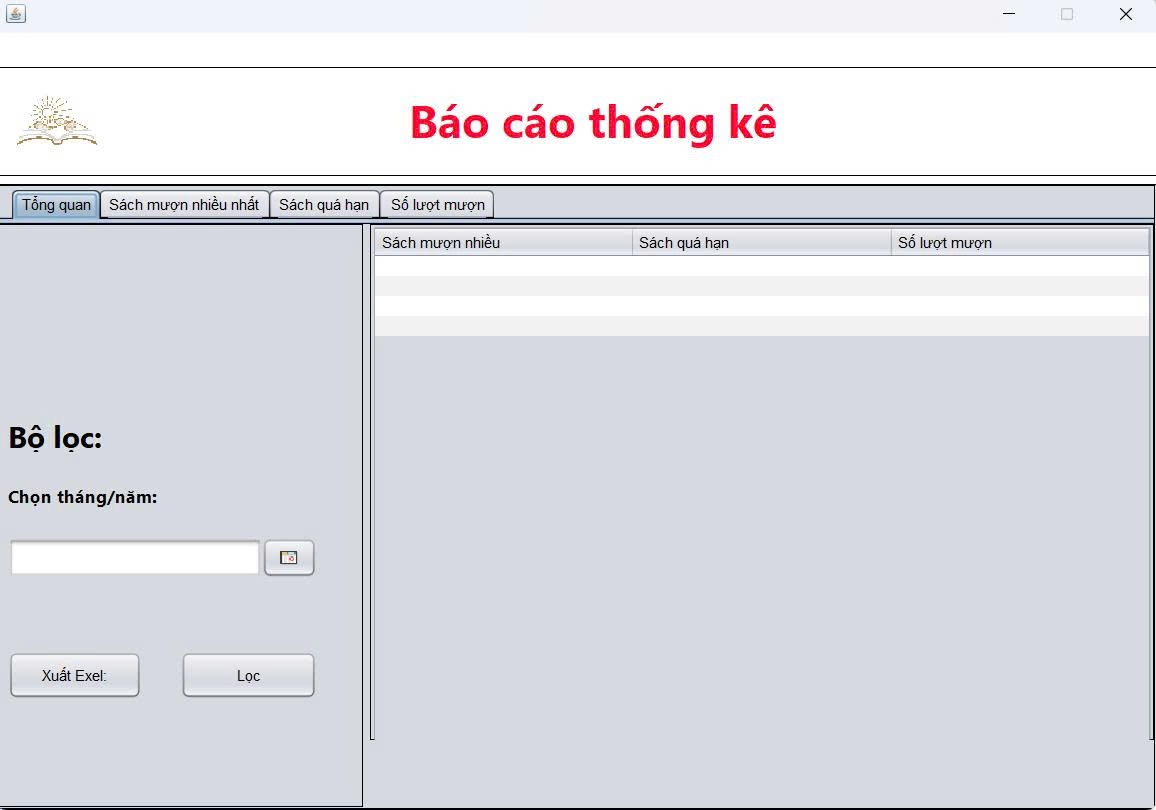
FORM CHÍNH CỦA ADMIN (QUẢN TRỊ VIÊN IT):



FORM QUẢN LÝ TÀI KHOẢN NGƯỜI DÙNG:

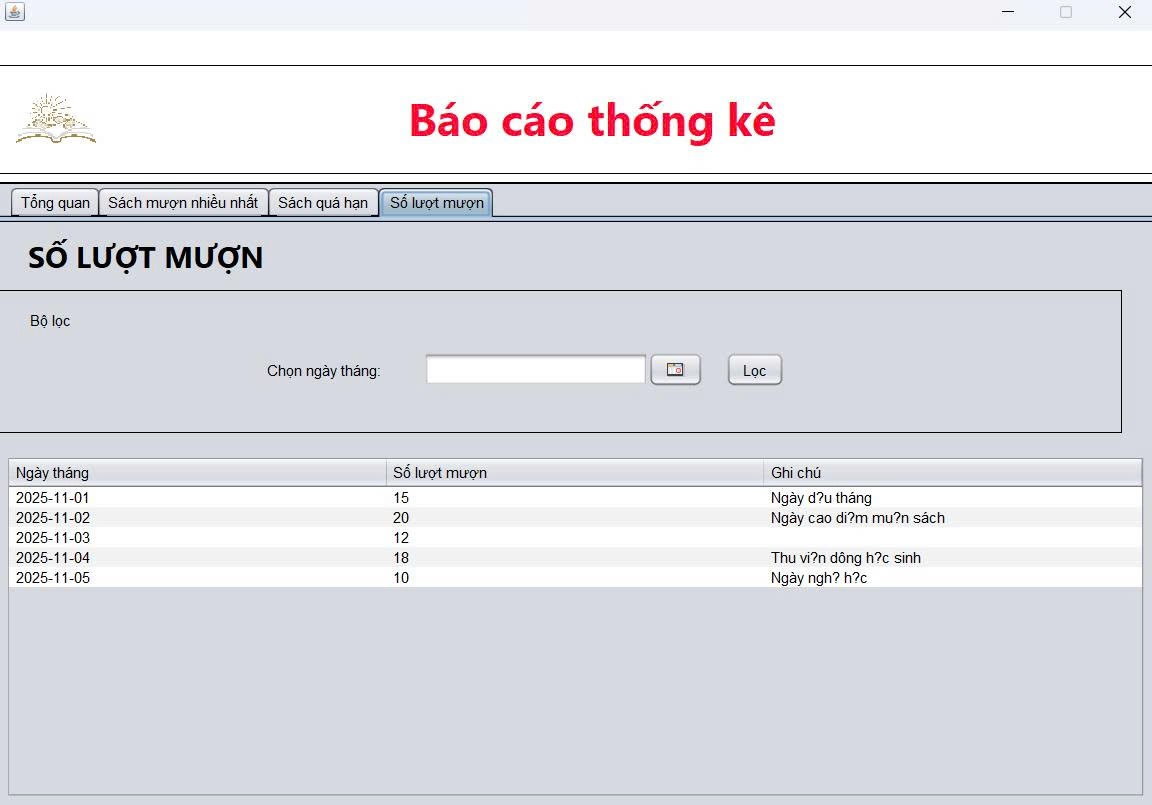


FORM BÁO CÁO THỐNG KÊ:





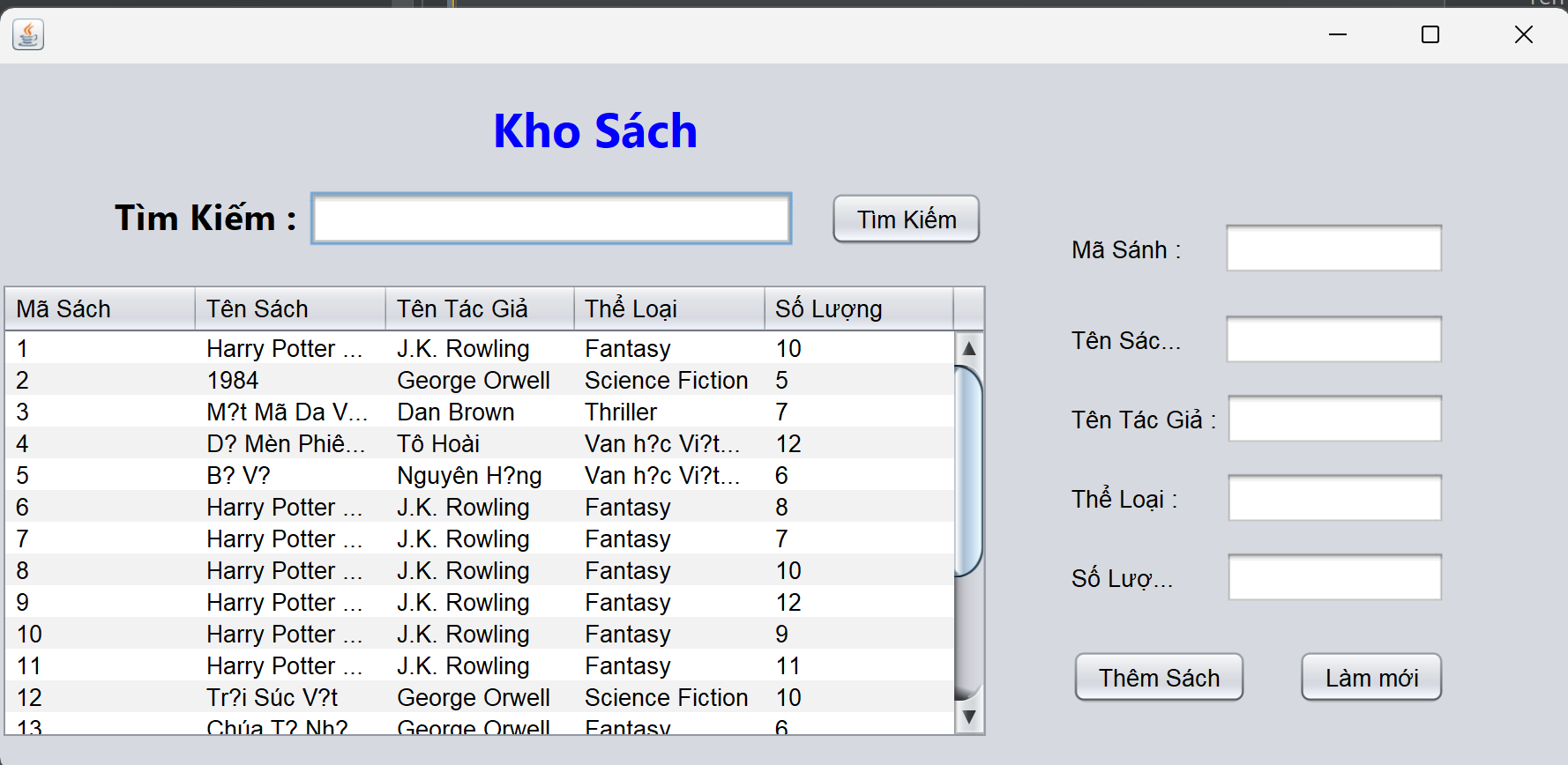




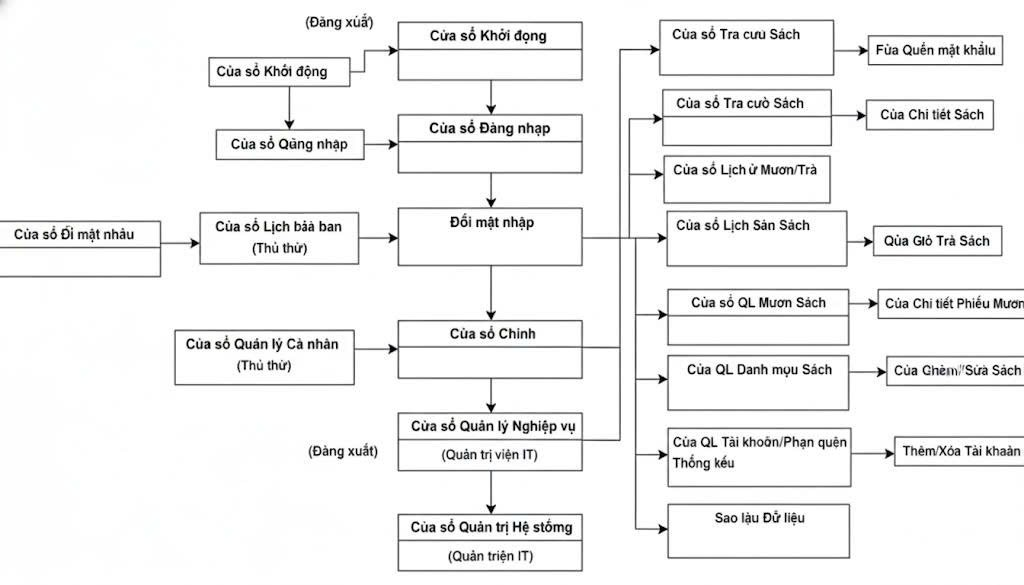
FORM THÔNG BÁO QUÁ HẠN:



FORM QUẢN LÝ KHO SÁCH:



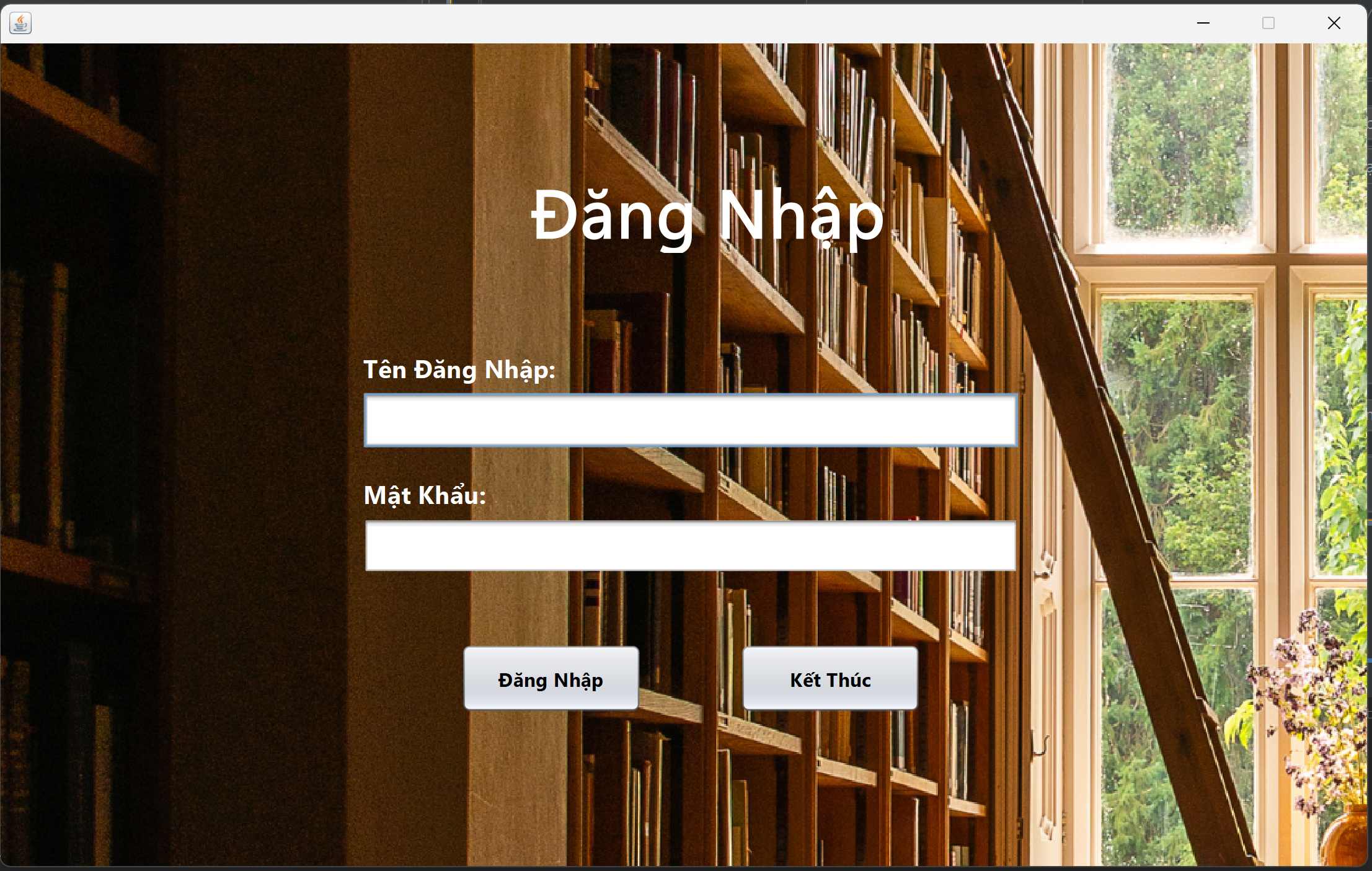
### Sơ đồ tổ chức giao diện



### Thiết kế giao diện cho các chức năng nghiệp vụ



#### Thiết kế Form Đăng nhập



#### Thiết kế Form Thống kê



# Thực hiện viết mã

## Viết mã tạo CSDL

### Tạo CSDL

### SQL truy vấn và thao tác

#### SQL đối với Bảng 1

#### SQL đối với Bảng 2

### Các thủ tục lưu tổng hợp thống kê

#### Procedure 1

#### Procedure 2

## Lập trình JDBC

### Lớp hỗ trợ

#### Lớp tiện ích X

#### Lớp tiện ích Y

### Model class - Các lớp mô tả dữ liệu

#### Model 1

#### Model 2

### DAO Class - Các lớp truy xuất dữ liệu

#### Lớp DAO làm việc với Bảng 1

#### Lớp DAO làm việc với Bảng 2

## Viết mã cho ứng dụng

### Xử lý Form X

### Xử lý Form Y

# Kiểm thử

## Kiểm thử form X

## Kiểm thử form Y

# Đóng gói và triển khai

## Hướng dẫn chuyển đổi jar thành exe

## Hướng dẫn cài đặt triển khai

## Hướng dẫn sử dụng phần mềm