

# **HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**



## **BÀI TẬP LỚN**

### **HỌC PHẦN: LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

#### **ĐỀ TÀI: QUẢN LÝ THƯ VIỆN**

**NHÓM 07 / LỚP D23CQCE04-B**

**Họ và tên: Nguyễn Chí Ngọc**

**Mã sinh viên: B23DCCE073**

**Họ và tên: Vũ Chiến Thắng**

**Mã sinh viên: B23DCVT393**

**Họ và tên: Phạm Đào Thanh Tùng**

**Mã sinh viên: B23DCVT449**

**Họ và tên: Nguyễn Trọng Nam Khánh**

**Mã sinh viên: B23DCCE052**

# MỤC LỤC

<b>Lời nói đầu</b>	<b>4</b>
<b>I. Bảng phân công nhiệm vụ</b>	<b>5</b>
<b>II. Tổng quan đề tài</b>	<b>5</b>
2.1. Sự cần thiết của đề tài	5
2.2. Hiện trạng của đối tượng cần quản lý	6
2.3. Các chức năng chính của dự án	6
2.3.1. Giới thiệu về kỹ thuật và công nghệ	6
2.3.1.1. Ngôn ngữ Java	6
2.3.1.2. Java Swing & Giao diện Người dùng (GUI)	7
2.3.1.3. SQL Server & Kiến trúc Cơ sở Dữ liệu	7
2.3.1.4. Kiến trúc 3 Lớp (Three-Tier Architecture)	7
2.3.2. Giới thiệu về dự án	8
2.3.2.1. Mục tiêu	8
2.3.2.2. Phạm vi	8
2.3.3. Phân tích yêu cầu	9
2.3.3.1. Các tác nhân (Actors)	9
2.3.3.2. Yêu cầu chức năng (Functional Requirements)	9
2.3.3.3. Yêu cầu phi chức năng (Non-Functional Requirements)	10
<b>III. Phân tích, thiết kế ra lớp</b>	<b>11</b>
3.1. Xác định các lớp	11
3.1.1. Entity Classes (Lớp thực thể)	11
3.1.2. Database Utility Class	11
3.1.3. GUI Classes (Lớp giao diện)	12
3.2. Thuộc tính & phương thức của các lớp	12
3.2.1. Class: Users (Abstract)	12
3.2.2. Class: Staff (extends Users)	13
3.2.3. Class: Librarian (extends Staff)	13
3.2.4. Class: Clerk (extends Staff)	13
3.2.5. Class: Borrower (extends Users)	13
3.2.6. Class: Books	14
3.2.7. Class: Loan	15
3.2.8. Class: Library	16
3.3. Quan hệ giữa các lớp	16
3.3.1. Quan hệ Kế thừa (Inheritance)	16
3.3.2. Quan hệ Kết hợp (Association)	17
3.3.3. Quan hệ Thành phần (Composition)	17
3.3.4. Quan hệ Sử dụng (Dependency)	17
3.3.5. Quan hệ Đa hình (Polymorphism)	17
3.3.6. Quan hệ Điều phối (Delegation)	17
<b>IV. Kiến trúc hệ thống</b>	<b>17</b>
4.1. Kiến trúc tổng quan	17
4.2. Kiến trúc phân tầng	18
4.2.1. Presentation Layer (Lớp trình bày)	18

4.2.2. Business Logic Layer (Lớp nghiệp vụ)	18
4.2.3. Data Access Layer (Lớp truy cập dữ liệu)	18
4.3. Luồng dữ liệu (Data Flow)	18
4.3.1. Luồng đăng nhập	18
4.3.2. Luồng mượn sách	19
4.3.3. Luồng tìm kiếm sách	19
4.4. Kiến trúc cơ sở dữ liệu	19
4.4.1. Mô hình quan hệ	19
4.4.2. Các bảng dữ liệu	19
4.5. Tính năng bảo mật	19
4.5.1. Authentication	19
4.5.2. Authorization	19
4.5.3. Data Protection	20
<b>V. Class diagram</b>	20
<b>VI. Hình ảnh giao diện</b>	21

## ***Lời nói đầu***

Trong kỷ nguyên số, khi công nghệ thông tin đang phát triển với tốc độ chóng mặt và thẩm thấu vào mọi ngóc ngách của đời sống xã hội, việc ứng dụng các phần mềm quản lý vào các hoạt động thường ngày đã trở thành một xu thế tất yếu, mang tính đột phá. Thay vì duy trì những phương pháp thủ công, truyền thống vốn tốn nhiều thời gian, công sức và dễ phát sinh sai sót, các tổ chức, doanh nghiệp nói chung và các thư viện nói riêng đang dần chuyển mình sang sử dụng các hệ thống phần mềm quản lý chuyên nghiệp. Những hệ thống này được xây dựng dựa trên nền tảng cơ sở dữ liệu vững chắc và các nguyên lý lập trình hiện đại, nhằm nâng cao hiệu suất công việc, đảm bảo độ chính xác và tạo ra khả năng mở rộng linh hoạt cho tương lai.

Lĩnh vực thư viện – kho tàng tri thức của nhân loại – cũng không nằm ngoài xu hướng đó. Trước đây, công tác quản lý thư viện thường gắn liền với những cuốn sổ sách ghi chép đồ sộ, các thẻ mục lục bằng giấy, và một quy trình mượn/trả sách phức tạp, thủ công. Cách thức vận hành này không chỉ đòi hỏi một lượng lớn nhân lực mà còn tiềm ẩn nhiều rủi ro như thất lạc thông tin, nhầm lẫn trong tra cứu, khó khăn trong việc thống kê báo cáo, và đặc biệt là trải nghiệm không mấy thuận tiện cho cả người quản lý lẫn người đọc. Khi nhu cầu tiếp cận thông tin và tri thức ngày càng gia tăng, những hạn chế của mô hình cũ càng trở nên lộ rõ, đòi hỏi một giải pháp công nghệ toàn diện và hiệu quả hơn.

Xuất phát từ thực tế đó, việc nghiên cứu và phát triển một hệ thống Quản lý Thư viện là một hướng đi cấp thiết và đầy triển vọng. Hệ thống này không chỉ đơn thuần số hóa các nghiệp vụ cốt lõi như quản lý đầu sách, quản lý thành viên, mượn/trả và tra cứu tài liệu, mà còn hướng tới việc tự động hóa toàn bộ quy trình, hỗ trợ thủ thư trong công tác thống kê, báo cáo doanh thu (nếu có), theo dõi tình trạng sách và tối ưu hóa trải nghiệm người dùng. Bằng việc vận dụng nhuần nhuyễn các nguyên lý cốt lõi của OOP như tính đóng gói, kế thừa, đa hình và trừu tượng, phần mềm sẽ có được một kiến trúc mã nguồn rõ ràng, dễ dàng bảo trì, nâng cấp và thích ứng với những yêu cầu thay đổi trong tương lai.

Nhận thức được tầm quan trọng của việc ứng dụng công nghệ vào quản lý, nhóm chúng em đã lựa chọn đề tài "Phần mềm Quản lý Thư viện" làm dự án thực hành cho môn học Lập trình Hướng Đối tượng. Thông qua dự án này, chúng em mong muốn được vận dụng những kiến thức lý thuyết đã được học – từ phân tích yêu cầu, thiết kế các lớp đối tượng và mối quan hệ giữa chúng, cho đến việc triển khai thành một ứng dụng hoàn chỉnh – để hiểu sâu hơn về sức mạnh và tính thực tiễn của mô hình lập trình hướng đối tượng trong việc giải quyết các bài toán phức tạp.

Nhóm chúng em xin chân thành gửi lời cảm ơn sâu sắc đến cô Đỗ Thị Liên, người đã tận tình giảng dạy, hướng dẫn và tạo mọi điều kiện thuận lợi để chúng em có thể hoàn thành đề tài báo cáo này một cách tốt nhất

**I. Bảng phân công nhiệm vụ**

<b>MSV</b>	<b>Họ và tên</b>	<b>Nhiệm vụ</b>	<b>Đánh giá</b>	<b>Ghi chú</b>
<b>B23DCCE073</b>	<b>Nguyễn Chí Ngọc</b>	Viết code Borrower, User, Library, StudentMenuUI, dbConnectivity Viết báo cáo Chạy demo, test chức năng, góp ý	Hoàn thành	
<b>B23DCVT393</b>	<b>Vũ Chiến Thắng</b>	Viết code Staff, User, Clerk, Library dbConnectivity, ClerkMenuUI Update tiến độ lên Github Chạy demo, test chức năng, góp ý Viết báo cáo	Hoàn thành	
<b>B23DCCE052</b>	<b>Nguyễn Trọng Nam Khánh (TN)</b>	Viết code LoginUI, dbConnectivity, LibrarianMenu, Staff, User, Phân công nhiệm vụ Giám sát tiến độ Tổ chức các cuộc họp và tổng kết chức năng hàng tuần	Hoàn thành	
<b>B23DCVT449</b>	<b>Phạm Đào Thanh Tùng</b>	Viết code Book, Library, Loan, dbConnectivity Phụ giúp, đóng góp các chức năng cần thiết	Hoàn thành	

		Chạy demo, test chức năng, góp ý Viết báo cáo		
--	--	--	--	--

## II. Tổng quan đề tài

### 2.1. Sự cần thiết của đề tài

Đối với thư viện, ứng dụng cho phép ta kiểm soát lượng dữ liệu sách, người đọc theo thời gian mà không bị nguy cơ mất mát dữ liệu. Trong thư viện, có rất nhiều loại sách với nhiều thể loại khác nhau, nhiều tác giả và có số lượng còn lại trong kho khác nhau. Lượng sinh viên cũng vậy, có nhiều sinh viên với nhiều mã sinh viên khác nhau, lớp và các ngành học khác nhau. Cho nên chương trình này cho phép kiểm soát thông tin của thư viện để quản lý và phục vụ cho nhu cầu đọc mượn của người đọc.

### 2.2. Hiện trạng của đối tượng cần quản lý

Hiện nay, thư viện là một hệ thống khá phổ biến trong các trường đại học cũng như các trường THPT. Đi cùng với sự phát triển của thư viện là nhu cầu của độc giả ngày một tăng lên, vì vậy số lượng sách nhập vào trong thư viện cũng tăng lên rất nhiều mà hệ thống cũ trước đây khó có thể quản lý và kiểm soát tốt được do cơ sở dữ liệu ngày một quá lớn và việc quản lý bằng thủ công chiếm phần lớn trong việc quản lý thư viện. Thư viện được xây dựng nhằm phục vụ, cung cấp tài nguyên thông tin và dịch vụ thông tin cho tất cả các bạn đọc là sinh viên, giảng viên và cán bộ nhân viên của trường. Hằng năm số lượng sách nhập vào thư viện ngày càng nhiều, để đáp ứng nhu cầu xem và mượn sách của độc giả, vì thế việc đọc sách và ghi chép lại các quá trình mượn đọc ngày càng nhiều gây khó khăn cho cả độc giả lẫn thủ thư mà nguyên nhân chủ yếu là do thư viện vẫn còn áp dụng lối quản lý thủ công bằng giấy tờ nên như việc tìm kiếm khó khăn, không nắm được tình trạng đọc sách, khó khăn đòi sách quá hạn, quản lý sách ra vào kho khó khăn và làm hồ sơ phức tạp,... Ban quản lý thư viện đang gặp nhiều khó khăn, bất cập trong việc quản lý thư viện. Vì vậy, Ban quản lý thư viện muốn công nghệ hoá các quy trình quản lý của thư viện, nhằm mục tiêu thúc đẩy và nâng cao hiệu quả trong công tác nghiệp vụ quản lý thư viện một cách tối ưu hơn với đầy đủ các tính năng cho các hoạt động nghiệp vụ quản lý. Ban quản lý thư viện yêu cầu muốn xây dựng cơ sở dữ liệu để quản lý hệ thống hoá được quy trình nghiệp vụ của thư viện, đáp ứng được những nhu cầu mượn trả sách của bạn đọc nhằm giải quyết những khó khăn, bất tiện của thư viện.

### 2.3. Giới thiệu về dự án

#### 2.3.1. Giới thiệu về kỹ thuật và công nghệ

##### 2.3.1.1. Ngôn ngữ Java

- Máy ảo Java (JVM): JVM đóng vai trò trung gian, thực thi mã bytecode được biên dịch từ mã nguồn Java. Điều này đảm bảo tính độc lập với hệ điều hành và phần cứng nền tảng.

- Hướng Đối tượng (OOP): Java là ngôn ngữ thuần hướng đối tượng, cung cấp nền tảng vững chắc để triển khai các nguyên lý như Kế thừa, Đóng gói, Đa hình và Trừu tượng – những nguyên lý đã được áp dụng triệt để trong kiến trúc của toàn bộ hệ thống (thể hiện qua các lớp Users, Staff, Borrower, Books, Loan).

- Tính Đa nền tảng: Ứng dụng có thể chạy trên bất kỳ hệ điều hành nào (Windows, Linux, macOS) mà có cài đặt JVM, nhờ vào khẩu hiệu "Viết một lần, chạy mọi nơi".

- Hỗ trợ Đa luồng: Java cung cấp các API mạnh mẽ cho lập trình đa luồng, hữu ích cho việc phát triển các ứng dụng có nhiều người dùng đồng thời trong tương lai.

- Tính Bảo mật: Mô hình bảo mật của Java giúp xây dựng các ứng dụng an toàn, hạn chế các truy cập trái phép vào tài nguyên hệ thống.

- Thư viện Phong phú: Hệ thống thư viện chuẩn (Standard Library) đồ sộ, đặc biệt là các gói java.sql cho kết nối cơ sở dữ liệu và javax.swing để xây dựng Giao diện Người dùng Đồ họa (GUI), đã được tận dụng triệt để trong dự án.

#### **2.3.1.2. Java Swing & Giao diện Người dùng (GUI)**

- Thành phần Độc lập Nền tảng: Các thành phần GUI của Swing (như JFrame, JButton, JTextField, JTable, JTabbedPane) được vẽ trực tiếp bằng Java, đảm bảo giao diện hiển thị nhất quán trên mọi hệ điều hành.

- Kiến trúc MVC (Model-View-Controller): Các thành phần như JTable sử dụng DefaultTableModel để tách biệt dữ liệu (Model) và giao diện (View), giúp code dễ bảo trì và mở rộng.

- Ứng dụng trong đề tài: Toàn bộ giao diện của hệ thống, từ màn hình đăng nhập (LoginUI), menu chính cho Sinh viên (StudentMenuUI), Nhân viên (ClerkMenuUI) đến Thủ thư (LibrarianMenu), đều được xây dựng dựa trên Swing. Điều này mang lại trải nghiệm đồ họa hoàn chỉnh, thay thế cho giao diện console ban đầu.

#### **2.3.1.3. SQL Server & Kiến trúc Cơ sở Dữ liệu**

- Hệ thống sử dụng Microsoft SQL Server làm Hệ quản trị Cơ sở dữ liệu (DBMS) quan hệ, thông qua lớp kết nối dbConnectivity.java.

- Kết nối JDBC: Ứng dụng sử dụng JDBC (Java Database Connectivity) API để thiết lập kết nối, thực thi truy vấn và cập nhật dữ liệu lên SQL Server. Chuỗi kết nối trong code (jdbc:sqlserver://localhost;databaseName=Library...) xác định máy chủ, cơ sở dữ liệu và thông tin xác thực.

- Quản lý Giao dịch (ACID): Các thao tác phức tạp, đặc biệt là nghiệp vụ Mượn/Trả sách, được đảm bảo tính toàn vẹn thông qua cơ chế giao dịch của cơ sở dữ liệu, tuân thủ các tiêu chuẩn ACID (Tính nguyên tử, Nhất quán, Cô lập, Bền vững).

- Truy vấn SQL: Hệ thống sử dụng rộng rãi các câu lệnh SQL (SELECT, INSERT, UPDATE, DELETE) để thao tác với các bảng như Books, Borrower, Staff, Loan, đáp ứng mọi yêu cầu nghiệp vụ từ tra cứu đến báo cáo.

- Bảo mật Cơ sở dữ liệu: Quyền truy cập được kiểm soát thông qua xác thực bằng tài khoản và mật khẩu SQL Server. Việc sử dụng PreparedStatement (có thể được áp dụng) giúp phòng chống các lỗ hổng bảo mật như SQL Injection.

#### **2.3.1.4. Kiến trúc 3 Lớp (Three-Tier Architecture)**

- Hệ thống được thiết kế theo mô hình kiến trúc 3 lớp điển hình, tách biệt rõ ràng các thành phần:

- Lớp trình bày (Presentation Layer): Bao gồm tất cả các lớp GUI (LoginUI, StudentMenuUI, ClerkMenuUI, LibrarianMenu), chịu trách nhiệm tương tác với người dùng cuối.

- Lớp nghiệp vụ (Business Logic Layer): Được đại diện bởi các lớp thực thể (Users, Staff, Books, Loan,...) và các phương thức nghiệp vụ bên trong chúng. Lớp này đóng gói toàn bộ logic xử lý của hệ thống, như đăng nhập, tìm sách, mượn/trả, tính phí.

- Lớp truy cập dữ liệu (Data Access Layer): Lớp dbConnectivity đảm nhiệm việc tương tác trực tiếp với cơ sở dữ liệu SQL Server, thực hiện các thao tác CRUD (Create, Read, Update, Delete) một cách trong suốt với các lớp trên.

#### **2.3.2. Giới thiệu về dự án**

**2.3.2.1. Mục tiêu:** Bằng cách vận dụng kiến thức về lập trình hướng đối tượng đã học, nhóm chúng em xây dựng Phần mềm Quản lý Thư viện nhằm hướng đến những mục tiêu cụ thể sau:

- Tối ưu hóa và tự động hóa quy trình nghiệp vụ: Thay thế hoàn toàn phương pháp quản lý thủ công bằng sổ sách, giúp các thao tác như mượn, trả, tra cứu sách, tính phí phát trở nên nhanh chóng, chính xác và giảm thiểu sai sót.

- Cung cấp giao diện trực quan và thân thiện: Phát triển một ứng dụng với giao diện console rõ ràng (và có thể mở rộng sang GUI trong tương lai), giúp nhân viên thư viện (Thủ thư, Nhân viên quầy) và bạn đọc dễ dàng thao tác, từ việc tìm kiếm tài liệu đến quản lý các giao dịch mượn/trả.

- Hỗ trợ công tác quản trị toàn diện: Cung cấp cho Thủ thư đầy đủ công cụ để quản lý kho tài liệu (thêm, xóa, sửa thông tin sách), quản lý hồ sơ bạn đọc, theo dõi tình trạng mượn/trả, và xử lý các khoản phí phạt một cách hệ thống, từ đó nâng cao hiệu quả vận hành của thư viện.

- Áp dụng triệt để các nguyên lý OOP: Dự án được xây dựng dựa trên nền tảng vững chắc của bốn tính chất cốt lõi trong Lập trình Hướng Đối tượng: Tính Đóng gói (Encapsulation), Tính Kế thừa (Inheritance), Tính Đa hình (Polymorphism) và Tính Trừu tượng (Abstraction). Cấu trúc hệ thống được thiết kế rõ ràng thông qua biểu đồ lớp UML, đảm bảo tính mạch lạc, dễ bảo trì và mở rộng khi cần thiết.

#### **2.3.2.2. Phạm vi:** Phạm vi cụ thể của dự án phần mềm này bao gồm:

- Quản lý dữ liệu cốt lõi của thư viện: Bao gồm thông tin về Sách (Books), Người dùng (Users) - được phân loại thành Bạn đọc (Borrower), Nhân viên (Staff), Thủ thư (Librarian) và Nhân viên quầy (Clerk), cùng với dữ liệu về các Giao dịch mượn/trả (Loan).



- Triển khai các chức năng nghiệp vụ chính:
  - + Cho Người dùng: Đăng nhập, Tìm kiếm sách (theo Tên, Tác giả, Chủ đề).
  - + Cho người đọc: Xem thông tin cá nhân, theo dõi lịch sử mượn sách.
  - + Cho Nhân viên (Clerk): Thực hiện nghiệp vụ mượn/trả sách (Check-out/Check-in), gia hạn sách (Renew), thêm và cập nhật thông tin bạn đọc.
  - + Cho Thủ thư (Librarian): Toàn quyền quản lý kho sách (thêm, xóa, sửa sách).
- Phát triển giao diện: Xây dựng một ứng dụng với menu tương tác, giúp minh họa đầy đủ và rõ ràng các luồng nghiệp vụ của hệ thống.
- Thiết kế hướng đối tượng bài bản: Xây dựng biểu đồ lớp UML chi tiết và triển khai mã nguồn bằng Java, qua đó áp dụng các tính chất của OOP, xử lý ngoại lệ (Exception Handling) và các kiến thức nền tảng khác.

### 2.3.3. Phân tích yêu cầu

#### 2.3.3.1. Các tác nhân (Actors)

Actor	Vai trò	Quyền hạn
Sinh viên (Student)	Người dùng thuộc đối tượng bạn đọc	Tìm kiếm sách, xem thông tin cá nhân, xem lịch sử mượn sách
Nhân viên (Clerk)	Nhân viên thư viện	Thực hiện nghiệp vụ mượn/trả sách, quản lý thông tin bạn đọc, tra cứu sách
Thủ thư (Librarian)	Quản trị viên thư viện	Toàn quyền quản lý hệ thống, bao gồm tất cả chức năng của Clerk và quản lý kho sách
Hệ thống (System)	Xử lý tự động	Tính toán phí phạt, kiểm tra tính khả dụng của sách

#### 2.3.3.2. Yêu cầu chức năng (Functional Requirements)

- YC1: Quản lý Đăng nhập & Phân quyền
  - + Đăng nhập hệ thống: Người dùng đăng nhập bằng User ID và lựa chọn vai trò (Student/Clerk/Librarian)
  - + Phân quyền tự động: Hệ thống tự động xác định và chuyển hướng đến giao diện phù hợp dựa trên vai trò

- + Kiểm tra tính hợp lệ: Xác thực User ID có tồn tại trong cơ sở dữ liệu và thuộc vai trò được chọn
- YC2: Quản lý Kho Sách:
  - + Tìm kiếm sách linh hoạt:
    - \* Tìm theo tựa đề (Title)
    - \* Tìm theo tác giả (Author)
    - \* Tìm theo chủ đề (Subject)
  - + Quản lý thông tin sách (Chỉ Librarian):
    - \* Thêm sách mới vào kho
    - \* Xóa sách khỏi hệ thống
    - \* Cập nhật thông tin sách (tựa đề, tác giả, chủ đề, số lượng)
  - + Theo dõi số lượng: Kiểm tra tính khả dụng của sách (CheckAvailability) và số lượng hiện có
- YC3: Quản lý Người dùng
  - + Phân loại người dùng: Student (Borrower), Clerk, Librarian
  - + Quản lý thông tin bạn đọc (Clerk/Librarian):
    - \* Thêm bạn đọc mới (AddBorrower)
    - \* Cập nhật thông tin cá nhân (tên, giới tính)
  - + Lưu trữ thông tin: ID, tên, giới tính, địa chỉ, số điện thoại, trạng thái phạt
- YC4: Quản lý Mượn/Trả Sách (Circulation)
  - + Cho mượn sách (CheckOutItem):
    - \* Kiểm tra trạng thái phạt của bạn đọc
    - \* Giảm số lượng sách có sẵn
    - \* Tạo bản ghi mượn sách với thông tin: ngày mượn, hạn trả (7 ngày)
  - + Nhận trả sách (CheckInItem):
    - \* Kiểm tra và cập nhật trạng thái trả sách
    - \* Tăng số lượng sách có sẵn
    - \* Tự động tính phí phạt nếu trả quá hạn
  - + Gia hạn sách (RenewItem): Cập nhật số lượng sách khi có thay đổi
- YC5: Quản lý Phí Phạt (Fine Management)
  - + Tính toán phí tự động (CalculateFine):
    - \*  $\text{Phí} = 30.000 \text{ VND/ngày}$  cho mỗi ngày trả trễ
    - \* Tự động kích hoạt khi sách được trả sau hạn
  - + Theo dõi trạng thái nợ: Đánh dấu bạn đọc là "fine\_defaulter" khi có phí chưa thanh toán
  - + Ghi nhận thanh toán (PayFine):
    - \* Xóa trạng thái nợ khi bạn đọc thanh toán
    - \* Reset số tiền phạt về 0
- YC6: Quản lý Lịch sử & Tra cứu
  - + Xem lịch sử mượn sách: Hiển thị toàn bộ lịch sử mượn/trả của bạn đọc
  - + Tra cứu thông tin: Cho phép nhân viên tra cứu lịch sử mượn sách của bất kỳ bạn đọc nào
  - + Theo dõi trạng thái: Hiển thị chi tiết các giao dịch đang mượn, đã trả, và các khoản phạt

### 2.3.3.3. Yêu cầu phi chức năng (Non-Functional Requirements)

Yêu cầu	Mô tả	Tiêu chí đánh giá
Hiệu năng	Thời gian phản hồi nhanh	< 3 giây cho thao tác tra cứu
Độ tin cậy	Hoạt động ổn định	Uptime > 99%, ít crash
Bảo mật	Phân quyền rõ ràng	Chỉ truy cập đúng chức năng được phân quyền
Khả năng bảo trì	Code module hóa	Dễ dàng sửa đổi từng thành phần
Khả năng sử dụng	Giao diện trực quan	Người dùng không cần đào tạo nhiều
Khả năng mở rộng	Dễ thêm tính năng mới	Kiến trúc hỗ trợ mở rộng

## III. Phân tích, thiết kế ra lớp

### 3.1. Xác định các lớp

- Hệ thống được thiết kế với kiến trúc hướng đối tượng rõ ràng, bao gồm 8 entity classes + 1 database utility class + 5 GUI classes.

#### 3.1.1. Entity Classes (Lớp thực thể)

STT	Tên lớp	Quan hệ kế thừa	Nhiệm vụ
1	Users	Abstract class	Lớp cơ sở trừu tượng quản lý thông tin chung của tất cả người dùng
2	Staff	extends Users	Lớp nhân viên thư viện, xử lý các nghiệp vụ cơ bản
3	Librarian	extends Staff	Thủ thư, có quyền quản lý toàn bộ kho sách
4	Clerk	extends Staff	Nhân viên quầy, thực hiện nghiệp vụ mượn/trả
5	Borrower	extends Users	Bạn đọc (sinh viên), quản lý thông tin mượn sách cá nhân

6	Books		Quản lý thông tin sách trong thư viện
7	Loan		Quản lý giao dịch mượn/trả sách
8	Library		Lớp điều phối chính, quản lý toàn bộ dữ liệu hệ thống

### 3.1.2. Database Utility Class

STT	Tên lớp	Nhiệm vụ
1	dbConnectivity	Quản lý kết nối database, thực thi truy vấn SQL, ánh xạ dữ liệu

### 3.1.3. GUI Classes (Lớp giao diện)

STT	Tên lớp	Kế thừa	Nhiệm vụ
1	LoginUI	JFrame	Giao diện đăng nhập hệ thống
2	StudentMenuUI	JFrame	Menu chính cho sinh viên
3	ClerkMenuUI	JFrame	Menu chính cho nhân viên thư viện
4	LibrarianMenu	JFrame	Menu chính cho thủ thư
5	ClerkMenuUI (base)	JFrame	Lớp cơ sở cho menu nhân viên

## 3.2. Thuộc tính & phương thức của các lớp

### 3.2.1. Class: Users (Abstract)

- Attributes:
  - + user\_id: int - Mã định danh người dùng (private)
  - + user\_name: String - Tên người dùng (private)
  - + gender: char - Giới tính (private)
- Methods:
  - + Users() - Constructor mặc định
  - + Users(int, String, char) - Constructor có tham số
  - + SearchBookbyTitle(String title): ArrayList<Books> - Tìm sách theo tựa đề
  - + SearchBookbyAuthor(String author): ArrayList<Books> - Tìm sách theo tác giả

- + SearchBookbySubject (String subject): ArrayList<Books> - Tìm sách theo chủ đề
- + GetId: int - Getter ID người dùng
- + GetName: String - Getter tên người dùng
- + GetGender: char - Getter giới tính
- + SetId: void - Setter ID
- + SetName: void - Setter tên
- + SetGender: void - Setter giới tính
- + PrintInformation: String - Hiện thị thông tin cơ bản
- + SetFineAmount: void - Thiết lập số tiền phạt
- + SetFineStatus: boolean - Thiết lập trạng thái phạt
- + UpdateLoanInfo: void - Cập nhật thông tin mượn
- + GetFineStatus: boolean - Lấy trạng thái phạt
- + AddLoanInfo: boolean - Thêm thông tin mượn
- + abstract String ViewInformation(ArrayList<Loan> LoanLoanList, int user\_id) - Phương thức trừu tượng xem thông tin
- Relationships:
  - + Lớp cha của Staff, Borrower
  - + Sử dụng Books thông qua phương thức tìm kiếm
  - + Liên kết với Loan thông qua user\_id

### 3.2.2. Class: Staff (extends Users)

- Attributes: (Kế thừa tất cả từ Users)
- Methods:
  - + Staff – Constructor
  - + CheckOutItem: boolean - Xử lý nghiệp vụ mượn sách
  - + CheckInItem: boolean - Xử lý nghiệp vụ trả sách
  - + ViewInformation: String-Override: Xem thông tin mượn sách của user
  - + AddBorrower:boolean - Thêm bạn đọc mới
  - + UpdatePersonalInformation: boolean - Cập nhật thông tin cá nhân bạn đọc
  - + RenewItem: void - Cập nhật số lượng sách (tăng/giảm)
  - + PayFine: void - Xử lý thanh toán phạt
- Relationships:
  - + Kế thừa từ Users
  - + Lớp cha của Librarian, Clerk
  - + Sử dụng Books, Loan, Borrower trong nghiệp vụ
  - + Tương tác với dbConnectivity để lưu trữ dữ liệu

### 3.2.3. Class: Librarian (extends Staff)

- Attributes: (Kế thừa tất cả từ Staff)
- Methods:
  - + Librarian(int user\_id, String user\_name, char gender) – Constructor
  - + AddBook: void - Thêm sách mới vào kho
  - + DeleteBook: boolean - Xóa sách khỏi hệ thống

- + ChangeInfo: void - Thay đổi thông tin sách
- Relationships:
  - + Kế thừa từ Staff
  - + Có toàn quyền quản lý Books
  - + Sử dụng dbConnectivity cho thao tác thêm/xóa/sửa sách

### 3.2.4. Class: Clerk (extends Staff)

- Attributes: (Kế thừa tất cả từ Staff)
- Methods:
  - + Clerk(int user\_id, String user\_name, char gender) – Constructor
- Relationships:
  - + Kế thừa từ Staff
  - + Chuyên xử lý nghiệp vụ với Borrower và Loan

### 3.2.5. Class: Borrower (extends Users)

- Attributes:
  - + address: String - Địa chỉ (private)
  - + telephone: String - Số điện thoại (private)
  - + fine\_defaulter: boolean - Trạng thái nợ phạt (private)
  - + fine: double - Số tiền phạt (private)
  - + BookLoans: ArrayList<Loan> - Danh sách sách đang mượn (private)
- Methods:
  - + Borrower() - Constructor mặc định
  - + Borrower(int, String, char, String, String, boolean, double) - Constructor có tham số
  - + GetFineStatus(): boolean - Override: Lấy trạng thái phạt
  - + GetFineAmount(): double - Lấy số tiền phạt
  - + GetAddress(): String - Lấy địa chỉ
  - + GetTelephone(): String - Lấy số điện thoại
  - + SetFineStatus: boolean - Override: Thiết lập trạng thái phạt
  - + SetTelephone: boolean - Cập nhật số điện thoại
  - + SetAddress: boolean - Cập nhật địa chỉ
  - + SetFineAmount: void - Override: Thiết lập số tiền phạt
  - + SetName: void - Override: Cập nhật tên
  - + SetGender: void - Override: Cập nhật giới tính
  - + AddLoanInfo: boolean - Override: Thêm thông tin mượn sách
  - + AllLoansofUser: void - Tải tất cả giao dịch mượn của user
  - + UpdateLoanInfo: void - Override: Cập nhật thông tin mượn
  - + ViewInformation: String - Override: Hiện thị thông tin chi tiết + lịch sử mượn
- Relationships:
  - + Kế thừa từ Users
  - + Có quan hệ 1-N với Loan (một borrower có nhiều loan)
  - + Sử dụng Books thông qua Loan
  - + Tương tác với dbConnectivity để cập nhật thông tin cá nhân

### 3.2.6. Class: Books

- Attributes:
  - + book\_id: int - Mã sách (private)
  - + author: String - Tác giả (private)
  - + title: String - Tựa đề (private)
  - + subject: String - Chủ đề (private)
  - + quantity: int - Số lượng (private)
- Methods:
  - + Books() - Constructor mặc định
  - + Books(int, String, String, String, int) - Constructor có tham số
  - + SetBookId: void - Setter mã sách
  - + SetAuthor: void - Setter tác giả
  - + SetTitle: void - Setter tựa đề
  - + SetQuantity: void - Setter số lượng
  - + SetSubject: void - Setter chủ đề
  - + GetTitle(): String - Getter tựa đề
  - + GetAuthor(): String - Getter tác giả
  - + GetBookId(): int - Getter mã sách
  - + GetSubject(): String - Getter chủ đề
  - + GetaBook(): Books - Lấy đối tượng sách
  - + GetQuantity(): int - Getter số lượng
  - + CheckAvailability: boolean - Kiểm tra tính khả dụng của sách
  - + DecreaseQuantity(): void - Giảm số lượng (khi mượn)
  - + IncreaseQuantity(): void - Tăng số lượng (khi trả)
  - + PrintInformation(): String - Hiển thị thông tin đầy đủ
- Relationships:
  - + Có quan hệ 1-N với Loan (một sách có thể được mượn nhiều lần)
  - + Được sử dụng bởi Users thông qua tìm kiếm
  - + Được quản lý bởi Librarian
  - + Tương tác với dbConnectivity để đồng bộ dữ liệu

### 3.2.7. Class: Loan

- Attributes:
  - + loanId: int - Mã giao dịch mượn (private)
  - + issue\_date: Date - Ngày mượn (private)
  - + due\_date: Date - Hạn trả (private)
  - + return\_date: Date - Ngày trả thực tế (private)
  - + borrower: Users - Người mượn (private)
  - + borrowed\_book: Books - Sách được mượn (private)
  - + fine\_status: String - Trạng thái phạt (private)
  - + returned\_status: boolean - Trạng thái đã trả (private)
- Methods:
  - + Loan() - Constructor mặc định
  - + GetaBookId(): int - Lấy mã sách được mượn

- + GetFineStatus(): String - Lấy trạng thái phạt
- + GetLoanId(): int - Lấy mã giao dịch
- + getReturnDate(): Date - Lấy ngày trả
- + getDueDate(): Date - Lấy hạn trả
- + getIssueDate(): Date - Lấy ngày mượn
- + GetaBook(): Books - Lấy đối tượng sách
- + GetaBorrowerId(): int - Lấy mã người mượn
- + Getborrower(): Users - Lấy đối tượng người mượn
- + GetStatus(): boolean - Lấy trạng thái đã trả
- + SetReturnedDate: void - Setter ngày trả
- + SetaBook: void - Setter sách được mượn
- + SetaBorrower: void - Setter người mượn
- + SetReturnStatus: void - Setter trạng thái đã trả
- + SetLoan: void - Cập nhật toàn bộ thông tin giao dịch
- + SetFineStatus: void - Setter trạng thái phạt
- + GetLoan: boolean - Xử lý quy trình mượn sách
- + CalculateFine(): double - Tính toán phí phạt nếu trả trễ
- + PrintLoanInfo(): String - Hiện thị thông tin giao dịch
- Relationships:
  - + Có quan hệ N-1 với Borrower (nhiều loan thuộc về một borrower)
  - + Có quan hệ N-1 với Books (nhiều loan liên quan đến một sách)
  - + Được xử lý bởi Staff
  - + Tương tác với dbConnectivity để lưu trữ giao dịch

### 3.2.8. Class: Library

- Attributes:
  - + UsersList: ArrayList<Users> - Danh sách người dùng (public static)
  - + BooksList: ArrayList<Books> - Danh sách sách (public static)
  - + LibrarianList: ArrayList<Librarian> - Danh sách thủ thư (public static)
  - + ClerkList: ArrayList<Clerk> - Danh sách nhân viên (public static)
  - + LoanList: ArrayList<Loan> - Danh sách giao dịch (public static)
- Methods:
  - + Library() - Constructor - khởi tạo và tải toàn bộ dữ liệu
  - + IsBorrowerPresent(int id): boolean - Kiểm tra tồn tại borrower
  - + IsClerkPresent(int id): boolean - Kiểm tra tồn tại clerk
  - + IsLibrarianPresent(int id): boolean - Kiểm tra tồn tại librarian
  - + CheckLoanofUser: String - Kiểm tra lịch sử mượn của user
  - + ClerkSearchBookbyTitle - Clerk tìm sách theo tựa đề
  - + ClerkSearchBookbyAuthor - Clerk tìm sách theo tác giả
  - + ClerkSearchBookbySubject - Clerk tìm sách theo chủ đề
  - + AddNewBorrower: boolean - Thêm borrower mới
  - + ClerkUpdatingInfo: Boolean - Clerk cập nhật thông tin
  - + ClerkRecordFine: void - Clerk ghi nhận thanh toán phạt
  - + ClerkCheckInItem: void - Clerk xử lý trả sách
  - + ClerkCheckOutItem: void - Clerk xử lý mượn sách



- + ClerkRenewItem: void - Clerk cập nhật số lượng sách
- + LibrarianAddNewBook: void - Librarian thêm sách mới
- + LibrarianDeleteBook: boolean - Librarian xóa sách
- + LibrarianUpdateBookInfo: void - Librarian cập nhật thông tin sách
- + main: void - Hàm main chính
- Relationships:
  - + Chứa và quản lý tất cả các danh sách dữ liệu
  - + Sử dụng dbConnectivity để tải dữ liệu khởi tạo
  - + Cung cấp interface cho các GUI classes
  - + Điều phối hoạt động giữa các entity classes

### **3.3. Quan hệ giữa các lớp**

#### **3.3.1. Quan hệ Kế thừa (Inheritance)**

- Staff → Users: Mọi nhân viên đều là người dùng hệ thống, kế thừa thông tin cơ bản và khả năng tìm kiếm sách
- Librarian → Staff: Thủ thư là một loại nhân viên, có toàn bộ quyền của nhân viên cộng thêm quyền quản lý kho sách
- Clerk → Staff: Nhân viên quầy là một loại nhân viên, kế thừa toàn bộ chức năng nghiệp vụ
- Borrower → Users: Bạn đọc là một loại người dùng, kế thừa thông tin cơ bản và khả năng tìm kiếm

#### **3.3.2. Quan hệ Kết hợp (Association)**

- Library → Users, Books, Loan: Thư viện chứa danh sách người dùng, sách và giao dịch, nhưng các đối tượng này có thể tồn tại độc lập
- Borrower → Loan: Mỗi bạn đọc có nhiều giao dịch mượn sách trong lịch sử
- Staff → Books, Loan: Nhân viên làm việc với sách và giao dịch trong quá trình xử lý nghiệp vụ

#### **3.3.3. Quan hệ Thành phần (Composition)**

- Loan → Books & Borrower: Mỗi giao dịch mượn sách bao gồm thông tin về sách được mượn và người mượn. Nếu giao dịch bị xóa, thông tin về mối quan hệ cụ thể này cũng mất đi
- Borrower → Personal Information: Thông tin cá nhân như địa chỉ, số điện thoại là thành phần không thể tách rời của bạn đọc

#### **3.3.4. Quan hệ Sử dụng (Dependency)**

- Tất cả Entity Classes → dbConnectivity: Các lớp thực thể phụ thuộc vào lớp kết nối cơ sở dữ liệu để lưu trữ và truy xuất thông tin
- GUI Classes → Library & Entity Classes: Các lớp giao diện phụ thuộc vào lớp thư viện và các lớp thực thể để thực hiện các chức năng nghiệp vụ
- Users → Books: Người dùng sử dụng thông tin sách để thực hiện tìm kiếm

#### **3.3.5. Quan hệ Đa hình (Polymorphism)**

- ViewInformation(): Được định nghĩa trừu tượng trong Users và được triển khai cụ thể trong Staff và Borrower, mỗi lớp có cách hiển thị thông tin khác nhau
- SetFineStatus(), SetFineAmount(), AddLoanInfo(): Các phương thức được ghi đè trong Borrower để cung cấp triển khai cụ thể cho bạn đọc

### 3.3.6. Quan hệ Điều phối (Delegation)

- GUI Classes → Library → Các Entity Classes: Lớp Library đóng vai trò điều phối, cung cấp interface đơn giản cho GUI trong khi ủy quyền công việc cho các lớp entity thích hợp
- Library → Staff/Librarian: Ủy quyền các thao tác nghiệp vụ cho đối tượng nhân viên/thủ thư cụ thể dựa trên ID

## IV. Kiến trúc hệ thống

**4.1. Kiến trúc tổng quan:** Hệ thống Quản lý Thư viện được xây dựng theo kiến trúc 3 lớp (3-Tier Architecture) kết hợp với mô hình MVC (Model-View-Controller), đảm bảo sự tách biệt rõ ràng giữa các thành phần và dễ dàng bảo trì, mở rộng.

### 4.2. Kiến trúc phân tầng

#### 4.2.1. Presentation Layer (Lớp trình bày)

- Thành phần:
  - + GUI Classes: LoginUI, StudentMenuUI, ClerkMenuUI, LibrarianMenu
  - + Công nghệ: Java Swing
- Nhiệm vụ:
  - + Hiển thị giao diện người dùng trực quan
  - + Tiếp nhận input từ người dùng
  - + Hiển thị kết quả và thông báo
  - + Điều hướng giữa các màn hình chức năng
- Đặc điểm:
  - + Hoàn toàn độc lập với logic nghiệp vụ
  - + Sử dụng các component Swing: JFrame, JButton, JTable, JtabbedPane
  - + Áp dụng mô hình Observer cho các sự kiện người dùng

#### 4.2.2. Business Logic Layer (Lớp nghiệp vụ)

- Thành phần:
  - + Entity Classes: Users, Staff, Librarian, Clerk, Borrower, Books, Loan
  - + Điều phối: Library
- Nhiệm vụ:
  - + Đóng gói toàn bộ logic nghiệp vụ thư viện
  - + Thực hiện các quy tắc nghiệp vụ (validation, tính phí, kiểm tra điều kiện)
  - + Điều phối luồng xử lý giữa các entity
  - + Quản lý trạng thái hệ thống
- Đặc điểm:
  - + Áp dụng triệt để các nguyên lý OOP
  - + Sử dụng design pattern: Factory Method (qua kế thừa), Facade (Library)
  - + Đảm bảo tính toàn vẹn dữ liệu nghiệp vụ

### 4.2.3. Data Access Layer (Lớp truy cập dữ liệu)

- Thành phần:
  - + dbConnectivity: Lớp kết nối và thao tác cơ sở dữ liệu
- Trách nhiệm:
  - + Quản lý kết nối đến SQL Server
  - + Thực thi các truy vấn SQL (CRUD operations)
  - + Ánh xạ dữ liệu quan hệ thành đối tượng Java
  - + Xử lý transaction và exception
- Đặc điểm:
  - + Sử dụng JDBC API
  - + Implement Data Mapper pattern
  - + Đóng gói hoàn toàn việc tương tác với database

## 4.3. Luồng dữ liệu (Data Flow)

### 4.3.1. Luồng đăng nhập

- LoginUI → Nhận input userID và role
- LoginUI → Library → Kiểm tra tính hợp lệ
- Library → dbConnectivity → Truy vấn database
- dbConnectivity → Trả về kết quả xác thực
- Library → LoginUI → Chuyển hướng đến menu tương ứng

### 4.3.2. Luồng mượn sách

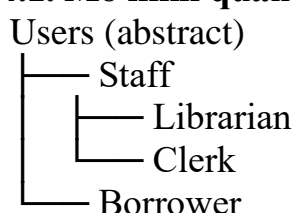
- ClerkMenuUI → Nhập userID, bookID
- ClerkMenuUI → Library.ClerkCheckOutItem()
- Library → Tìm Staff object tương ứng
- Staff → Kiểm tra điều kiện mượn (fine status, availability)
- Staff → Tạo Loan object mới
- Staff → dbConnectivity → Lưu thông tin loan
- Books → dbConnectivity → Cập nhật số lượng
- Kết quả → Hiển thị trên ClerkMenuUI

### 4.3.3. Luồng tìm kiếm sách

- StudentMenuUI → Nhập từ khóa, chọn loại tìm kiếm
- StudentMenuUI → Library → Chuyển đến User object
- User → dbConnectivity → Thực hiện truy vấn SQL
- dbConnectivity → Trả về ArrayList<Books>
- Kết quả → Hiển thị trong Jtable

## 4.4. Kiến trúc cơ sở dữ liệu

### 4.4.1. Mô hình quan hệ



- Books (1)  $\longleftrightarrow$  (N) Loan (N)  $\longleftrightarrow$  (1) Borrower

#### 4.4.2. Các bảng dữ liệu

- Books: book\_id, title, author, subject, quantity
- Borrower: borrower\_id, borrower\_name, borrower\_gender, borrower\_address, borrower\_number, fine\_defaulter, fine
- Staff: staff\_id, staff\_name, staff\_gender, rank
- Loan: loanid, issue\_date, due\_date, return\_date, fine\_status, returned\_status, borrower\_id, book\_id

#### 4.5. Tính năng bảo mật

**4.5.1. Authentication:** Xác thực qua UserID và phân quyền dựa trên role

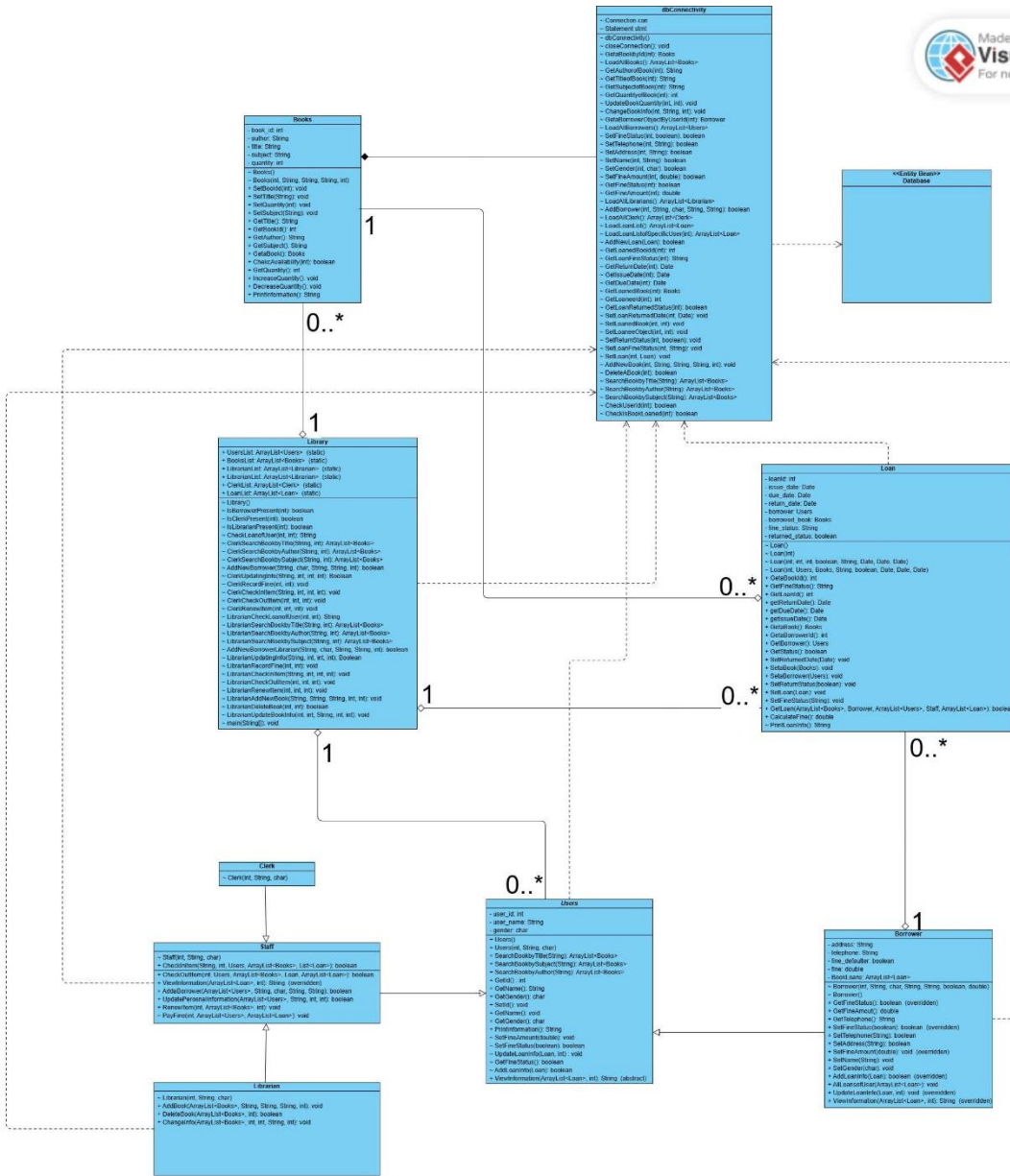
#### 4.5.2. Authorization

- Menu và chức năng khác nhau cho mỗi role
- Librarian: Full quyền
- Clerk: Chỉ nghiệp vụ mượn/trả
- Student: Chỉ xem và tìm kiếm

#### 4.5.3. Data Protection

- Input validation tại GUI
- Business rules validation tại Business Layer
- SQL injection protection qua parameterized queries

### V. Class diagram



## VI. Hình ảnh giao diện



# HỆ THỐNG QUẢN LÝ THƯ VIỆN

☒ Student (Sinh viên)

☐ Clerk (Nhân viên)

☐ Librarian (Thủ thư)

User ID: 101

**ĐĂNG NHẬP**

## Cập nhật Người mượn

User ID cần cập nhật: 103

Thông tin mới: Dom Dom Chua

Cập nhật Tên

Cập nhật Giới tính

## Log hoạt động

```
>> Thành công: Đã cho mượn sách 9 cho User 101.  
>> Thành công: Đã nhận trả sách 9 từ User 101.  
>> Thành công: Đã ghi nhận trả phạt cho User 101.  
>> Thành công: Đã Tăng số lượng cho Book ID 9.  
>> Thành công: Đã thêm người mượn mới: Vua dom dom  
>> --- Lịch sử User ID 103 ---
```

--- Kết thúc lịch sử ---

```
>> Thành công: Cập nhật thông tin cho User ID 103.
```

Librarian Menu (ID: 301)

Chào mừng, Le Van C (Librarian)

Đăng xuất

Trà cứu Sách & User

Giao dịch Mượn/Trả

Quản lý Người mượn

Quản lý Kho Sách (Admin)

Thêm Sách Mới

Tựa đề: Cam nang cac loai dom dom

Tác giả: Dach

Chủ đề: Khoa hoc

Số lượng: 5000000

Thêm Sách Mới

Xóa Sách

Book ID cần xóa:

Xóa Sách

Cập nhật Thông tin Sách

Book ID cần cập nhật:

Cập nhật thông tin: Tựa đề

Thông tin mới:

Cập nhật Sách

Log hoạt động

>> Hệ thống sẵn sàng.

>> Thành công: Đã thêm sách mới: Cam nang cac loai dom dom.

Clerk Menu (ID: 201)

Chào mừng, Tran Thi B (Clerk)

Đăng xuất

Trà cứu Sách & User

Giao dịch Mượn/Trả

Quản lý Người mượn

Thêm Người mượn Mới

Tên: Vua dom dom

Giới tính (M/F): M

Số điện thoại: 55000000

Địa chỉ: Cən Tho

Thêm Người mượn

Cập nhật Người mượn

User ID cần cập nhật:

Thông tin mới:

Cập nhật Tên

Cập nhật Giới tính

Log hoạt động

Id: 101	Name: Nguyen Van A (Student)	Gender: M	9	Roger S. Bjarne Stroustrup	The C++ Programming Language	Lập trình C++	43	available
Id: 101	Name: Nguyen Van A (Student)	Gender: M	9	Roger S. Bjarne Stroustrup	The C++ Programming Language	Lập trình C++	43	available

--- Kết thúc Lịch sử ---

>> Thành công: Đã cho mượn sách 9 cho User 101.

>> Thành công: Đã nhận trả sách 9 từ User 101.

>> Thành công: Đã ghi nhận Trả phạt cho User 101.

>> Thành công: Đã Tăng số lượng cho Book ID 9.

>> Thành công: Đã thêm người mượn mới: Vua dom dom

Công cụ cắt

Đã sao chép ảnh chụp màn hình vào bảng tạm

Tự động lưu vào thư mục ảnh chụp màn hình.

Đánh dấu và Chia sẻ

**Chào mừng, Nguyen Van A (Student)****Đăng xuất**

Tìm kiếm Sách Lịch sử Mượn sách

**Tải/Làm mới Lịch sử Mượn sách**

Thông tin cá nhân và lịch sử mượn sách:

Id: 101 Name: Nguyen Van A (Student) Gender: M Address :123 Duong Ba Dinh  
Telephone: 0909123456

Loan Info

loanId:1 issue date:2025-11-10 due date:2025-11-17 return\_date:2025-11-11  
borrower id 101 borrowed\_book :9 fine status :no fine returned status true

loanId:2 issue date:2025-11-10 due date:2025-11-17 return\_date:2025-11-11  
borrower id 101 borrowed\_book :9 fine status :no fine returned status true

**Chào mừng, Nguyen Van A (Student)****Đăng xuất**

Tìm kiếm Sách Lịch sử Mượn sách

Nhập từ khóa: Giao trình C++

Tìm theo Từ đề

Tìm theo Tác giả

Tìm theo Chủ đề

Book ID	Tựa đề	Tác giả	Chủ đề	Số lượng
10	Giao trình C++	Bộ Giáo dục và Đào tạo	Lập trình C++	25



Student Menu (ID: 101)

Chào mừng, Nguyen Van A (Student)

Đăng xuất

Tim kiếm SáchLịch sử Mượn sách

Nhập từ khóa:

Tim theo Từ khóa

Tim theo Tác giả

Tim theo Chủ đề

Book ID	Tựa đề	Tác giả	Chủ đề	Số lượng
6	Giáo trình Mạng máy tính	Trương Minh Nhật Quan	Mạng máy tính	60
7	Computer Networking: A Top-Down Approach	Kurose	Mạng máy tính	15

Student Menu (ID: 101)

Chào mừng, Nguyen Van A (Student)

Đăng xuất

Tim kiếm SáchLịch sử Mượn sách

Tải/Làm mới Lịch sử Mượn sách

Thông tin cá nhân và lịch sử mượn sách:

Id: 101   Name: Nguyen Van A (Student)   Gender: M   Address: 123 Duong Ba Dinh   Telephone: 0909123456

Loan Info

loanId:1	issue date:2025-11-10	due date:2025-11-17	return_date:2025-11-11	borrower id 101	borrowed_book :9	fine status :no fine	returned status: true
loanId:2	issue date:2025-11-10	due date:2025-11-17	return_date:2025-11-11	borrower id 101	borrowed_book :9	fine status :no fine	returned status: true

Công cụ cắt

Đã sao chép ảnh chụp màn hình vào bảng tạm

Tự động lưu vào thư mục ảnh chụp màn hình.

Đánh dấu và Chia sẻ

## Chào mừng, Tran Thi B (Clerk)

Đăng xuất

Tra cứu Sách &amp; User | Giao dịch Mượn/Trả | Quản lý Người mượn

Nhập từ khóa/ID: 101

Tìm sách theo Tựa đề

Tìm sách theo Tác giả

Tìm sách theo Chủ đề

Xem lịch sử User

Book ID	Tựa đề	Tác giả	Chủ đề	Số lượng
6	Giao trình Mạng máy tính	Trương Minh Nhật Quan	Mạng máy tính	60
7	Computer Networking: A Top-Down Approach	Kurose	Mạng máy tính	15

## Log hoạt động

```
Id: 101  Name:Nguyen Van A (Student)  Gender: M  9  Roger S. Bjarne Stroustrup  The C++ Programming Language  Lập trình C++  43  available
Id: 101  Name:Nguyen Van A (Student)  Gender: M  9  Roger S. Bjarne Stroustrup  The C++ Programming Language  Lập trình C++  43  available

--- Kết thúc Lịch sử ---
>> --- Lịch sử User ID 101 ---
Id: 101  Name:Nguyen Van A (Student)  Gender: M  9  Roger S. Bjarne Stroustrup  The C++ Programming Language  Lập trình C++  43  available
Id: 101  Name:Nguyen Van A (Student)  Gender: M  9  Roger S. Bjarne Stroustrup  The C++ Programming Language  Lập trình C++  43  available

--- Kết thúc Lịch sử ---
```