**TRƯỜNG ĐẠI HỌC PHENIKAA  
KHOA CÔNG NGHỆ THÔNG TIN**

**A logo with blue and orange circle

Description automatically generated**------**🙠🕮🙢**-----

**BÀI TẬP LỚN**

**KĨ THUẬT PHẦN MỀM**

**Tên đề tài: Quản lý sách trong một thư viện**

**Giáo viên hướng dẫn :** **Vũ Quang Dũng**

**Lớp : CSE702025-1-3-24 ( nhóm 5 )**

**Sinh viên thực hiện:**

|  |  |  |
| --- | --- | --- |
| Mã sinh viên | Họ và tên | Điểm |
| 23010244 | Trần Văn Dương |  |
| 23010220 | Trần Phương Nam |  |
| 23010211 | Trần Đình Danh |  |

**Hà Nội – 2025**

**LỜI MỞ ĐẦU**

Trên thế giới cũng như ở Việt Nam, công nghệ thông tin đã trở thành một

công nghệ mũi nhọn, nó là ngành khoa học kỹ thuật không thể thiếu trong việc

áp dụng vào các hoạt động xã hội như: quản lý hệ thống thư viện sách, kinh tế,

thông tin, kinh doanh, và mọi hoạt động trong lĩnh vực nghiên cứu...

Ở nước ta hiện nay, việc áp dụng vi tính hóa trong quản lý tại các cơ quan,

xí nghiệp, tổ chức đang rất phổ biến và trở nên cấp thiết, bởi ngành nghề nào

cũng đòi hỏi con người phải xử lý khối lượng công việc khổng lồ, và những kiến

thức, những suy nghĩ, những đào tạo chuyên sâu. Một vấn đề cấp thiết đặt ra

trong quản lý là làm thế nào để chuẩn hóa cách xử lý dữ liệu ở trường học.

Mặc dù đã rất cố gắng để hoàn thành công việc, nhưng do thời gian có hạn

và thiếu kinh nghiệm cũng như kỹ năng chưa cao nên việc phân tích và thiết kế

hệ thống quản lý còn nhiều thiếu sót, kính mong quý thầy cô góp ý,

bổ sung để chúng em hoàn thiện cho bài tập tốt hơn nữa.

Chúng em xin chân thành cảm ơn!

Sinh viên thực hiện !

**Mục lục**

**LỜI MỞ ĐẦU**

**Chương 1. Giới thiệu**

* 1. Giới thiệu đề tài

1.2. Mục tiêu

1.3 Phạm vi sử dụng

1.4 Công nghệ hỗ trợ

1.5 Bảng phân chia công việc

**Chương 2: Phân tích thiết kế thư viện sách**

2.1 Giới thiệu

2.2 Tài liệu đặc tả yêu cầu người dùng

2.3 Phân tích hệ thống

2.4 Thiết kế hướng đối tượng

2.5 Data Flow ( Luồng dữ liệu )

2.6 UI Flow ( Luông giao diện )

2.7 Sequence Diagram ( Sơ đồ trình tự )

2.8 State Diagram ( Trạng thái của sách )

2.9 Activity Diagram ( Quy trình mượn sách )

2.10 CSDL ( MySQL )

**Chương 3: Thiết kế giao diện**

**Chương 4: Triển khai phần mềm**

4.1 Cách chạy ứng dụng

4.2 Kiểm tra chức năng

4.3 Testcase và kiểm thử

4.4 Đa luông và tiến trình xử lý

4.5 Định hướng phát triển dịch vụ API

**Chương 5 : Kết luận**

5.1 Kết luận

5.2 Đánh giá kết quả đạt được

5.3 Hướng phát triển trong tương lai

**Tài liệu tham khảo**

**Chương 1: Giới thiệu về đề tài quản lý thư viện sách**

**1. Giới thiệu đề tài**

**1.1 Tên đề tài:**

Library Management System (LMS) – Phần mềm quản lý thư viện sách

Lý do chọn đề tài:

Trong thời đại công nghệ số hiện nay, nhu cầu quản lý thư viện sách theo cách truyền thống (sổ ghi tay, thẻ mượn trả) không còn đáp ứng được sự phát triển nhanh chóng của các thư viện hiện đại. Với lượng đầu sách ngày càng lớn, người dùng ngày càng đa dạng (sinh viên, giảng viên, nhân viên...), thư viện cần một hệ thống giúp quản lý thông tin sách, người mượn và quá trình mượn trả một cách hiệu quả, chính xác và tiết kiệm thời gian.

Bên cạnh đó, nhiều thư viện ở các trường học, trung tâm giáo dục hoặc khu dân cư vẫn chưa được trang bị hệ thống phần mềm chuyên dụng. Điều này dẫn đến tình trạng thất lạc sách, thống kê không đầy đủ, hoặc việc tra cứu mất nhiều thời gian và công sức.

Chính vì những lý do trên, nhóm quyết định chọn đề tài “Library Management System (LMS)” nhằm xây dựng một phần mềm có giao diện thân thiện, dễ sử dụng và phục vụ tốt nhu cầu quản lý thư viện sách cho các đơn vị giáo dục và cộng đồng.

**1.2. Mục tiêu**

Mục tiêu của hệ thống LMS là tạo ra một công cụ phần mềm hiệu quả, hỗ trợ công tác quản lý thư viện sách một cách khoa học và hiện đại. Cụ thể:

Quản lý thông tin sách trong thư viện: tên sách, tác giả, nhà xuất bản, thể loại, năm xuất bản, vị trí lưu trữ...

Hỗ trợ thủ thư và người dùng trong việc tra cứu nhanh sách theo từ khóa hoặc danh mục.

Quản lý quá trình mượn – trả sách: theo dõi người mượn, ngày mượn, ngày trả, tình trạng quá hạn.

Hạn chế tối đa tình trạng mất sách hoặc trả chậm nhờ hệ thống nhắc nhở tự động.

Cung cấp báo cáo thống kê định kỳ như: số sách đang có, sách đang được mượn, sách mượn nhiều nhất...

Tăng tính chuyên nghiệp và hiện đại trong công tác thư viện.

**1.3. Phạm vi sử dụng**

Phần mềm LMS được thiết kế phù hợp với quy mô thư viện vừa và nhỏ, áp dụng chủ yếu cho các trường học, trung tâm đào tạo, hoặc thư viện cộng đồng. Các chức năng chính bao gồm:

Quản lý sách: Thêm mới, chỉnh sửa, xóa sách; tìm kiếm sách theo tên, tác giả, thể loại...

Quản lý người dùng (thủ thư và người mượn): Lưu thông tin người mượn, tạo tài khoản người dùng.

Quản lý mượn – trả: Cập nhật lịch sử mượn/trả sách, trạng thái sách (đang mượn / có sẵn).

Hệ thống nhắc nhở: Gửi thông báo khi đến hạn trả sách hoặc sách quá hạn.

Thống kê – báo cáo: Tổng hợp các báo cáo như sách mượn nhiều nhất, sách chưa trả, lịch sử mượn...

Đăng nhập bảo mật: Người dùng và thủ thư đăng nhập vào hệ thống bằng tài khoản được cấp.

Hệ thống hiện hỗ trợ một quản trị viên thư viện và nhiều người mượn, chưa hỗ trợ phân quyền nâng cao.

**1.4. Công nghệ hỗ trợ**

+ Apache NetBeans

Apache NetBeans là một môi trường phát triển tích hợp (IDE) mã nguồn mở, chủ yếu dùng để lập trình Java, đồng thời hỗ trợ các ngôn ngữ khác như PHP, C/C++, HTML, JavaScript. Được phát triển bởi Apache Software Foundation, NetBeans có giao diện thân thiện, dễ sử dụng, phù hợp cho cả sinh viên và lập trình viên chuyên nghiệp.

NetBeans tích hợp nhiều công cụ tiện ích như trình biên dịch, trình gỡ lỗi, thiết kế giao diện đồ họa (GUI Builder), hỗ trợ Maven, Gradle và Git. IDE này cũng hỗ trợ phát triển ứng dụng web và desktop với Java EE, JavaFX. Với hệ thống plugin mở rộng và cộng đồng hỗ trợ mạnh mẽ, Apache NetBeans là lựa chọn phổ biến trong giáo dục và phát triển phần mềm Java.

+ XAMPP

XAMPP là một phần mềm mã nguồn mở dùng để xây dựng môi trường máy chủ web cục bộ. Tên gọi XAMPP là viết tắt của X (cross-platform), Apache, MySQL, PHP, và Perl. Phần mềm này giúp người dùng dễ dàng cài đặt và chạy máy chủ web trên các hệ điều hành như Windows, Linux, và macOS.

XAMPP được sử dụng phổ biến trong phát triển web, đặc biệt là với các ứng dụng sử dụng PHP và MySQL. Ưu điểm của XAMPP là cài đặt đơn giản, giao diện thân thiện, dễ sử dụng cho cả người mới học và lập trình viên chuyên nghiệp. Ngoài ra, XAMPP còn tích hợp sẵn phpMyAdmin để quản lý cơ sở dữ liệu MySQL, giúp thao tác với dữ liệu trở nên thuận tiện

+Java I/O: Đọc và Ghi Tập Tin Văn Bản

Trong phát triển phần mềm, khả năng xử lý dữ liệu từ tập tin là một yếu tố quan trọng, đặc biệt là trong các ứng dụng yêu cầu lưu trữ và truy xuất thông tin. Trong ngôn ngữ lập trình Java, việc làm việc với các tập tin văn bản có thể được thực hiện thông qua hai gói chính: java.io và java.nio.+ MySQL

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mã nguồn mở, được sử dụng rộng rãi để lưu trữ và quản lý dữ liệu trong các ứng dụng web và phần mềm. MySQL sử dụng ngôn ngữ truy vấn SQL (Structured Query Language) để thao tác dữ liệu.

MySQL nổi bật với khả năng xử lý nhanh, đáng tin cậy và dễ tích hợp với các ngôn ngữ lập trình phổ biến như PHP, Java, Python. Đây là một thành phần quan trọng trong nhiều hệ thống như XAMPP, LAMP, được dùng để xây dựng các ứng dụng như hệ thống quản lý thư viện, website bán hàng, quản lý sinh viên,...

+ GUI Builder (Matisse)

GUI Builder, còn được gọi là Matisse, là công cụ thiết kế giao diện đồ họa được tích hợp sẵn trong NetBeans IDE. Công cụ này cho phép người dùng tạo giao diện Java Swing một cách trực quan bằng cách kéo và thả các thành phần như nút bấm (JButton), nhãn (JLabel), bảng (JTable), ô nhập liệu (JTextField)...

Với GUI Builder, lập trình viên có thể dễ dàng thiết kế giao diện người dùng mà không cần viết mã thủ công, giúp tiết kiệm thời gian và giảm lỗi khi xây dựng các ứng dụng desktop. Giao diện được tạo sẽ tự động sinh mã Java tương ứng ở phần thiết kế

+ **Java SE**

(Java Platform, Standard Edition) là nền tảng chính để phát triển ứng dụng desktop hoặc console bằng Java, với các thư viện phong phú như Swing/SWT để xây dựng giao diện người dùng .

**+ Mô hình MVC**

**(Model–View–Controller)** là một **kiến trúc thiết kế phần mềm**, giúp phân tách rõ ràng giữa **dữ liệu - phần xử lý** (Model), **giao diện** (View) và **điều khiển luồng** (Controller), làm code trở nên dễ bảo trì và mở rộng hơn

**1.5 Bảng phân chia công việc:**

**- Bảng Phân Công Chi Tiết Theo Tuần**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tuần** | **Trần Văn Dương** | **Trần Phương Nam** | **Trần Đình Danh** |
| 1 | Thiết kế DB, DBConnection | UI Design, LoginPage | Phân tích yêu cầu, UseCase |
| 2-3 | ManageBooks, ManageStudents | SignupPage, HomePage | Sequence Diagram, State Diagram |
| 4-5 | IssueBook, ReturnBook | UI Testing, Refinement | Documentation, UML |
| 6 | Reports, Statistics | Final UI Polish | Testing, Bug Fixes |
| 7-8 | API Design, Threading | Performance Optimization | User Manual, Deployment |

* **Road map**

**Phase 1**: Cơ sở hạ tầng – Setup công cụ (IDE, CSDL) & giao diện sơ khởi

**Phase 2**: Quản lý người dùng – Đăng nhập/đăng ký & dashboard

**Phase 3**: Quản lý dữ liệu – CRUD sách/sinh viên + tìm kiếm/lọc

**Phase 4**: Mượn/trả sách – Xử lý logic + phạt trễ hạn

**Phase 5**: Báo cáo/thống kê – Lịch sử, thống kê, biểu đồ

**Phase 6**: Hoàn thiện & triển khai – Testing, tối ưu, hướng dẫn

**Phase 7**: Tài liệu & UML – Vẽ diagram, viết spec

**Phase 8**: Hiệu năng & API –tích hợp dịch vụ

**Phase 9**: Test tổng, hoàn thiện báo cáo và nộp dự án

**Chương 2: Phân tích thiết kế thư viện sách**

**2.1 Giới thiệu**

Quản lý sách trong thư viện bằng Java là một ứng dụng giúp tổ chức và theo dõi các hoạt động liên quan đến sách, chẳng hạn như thêm sách mới, cập nhật thông tin, tìm kiếm và quản lý trạng thái mượn trả. Bằng cách sử dụng các lớp và cấu trúc dữ liệu trong Java, ứng dụng này hỗ trợ quản lý hiệu quả danh mục sách trong thư viện, giúp người dùng dễ dàng truy cập thông tin và duy trì hệ thống sách một cách khoa học.

* 1. **Phân tích các chức năng**

- Người sử dụng phần mềm gồm: Quản lý, nhân viên.

- Phân quyền chi tiết vai trò, quyền hạn của admin như thêm/xóa/sửa nhân viên, nhập sách từ nhà cung cấp, quản lý toàn bộ hệ thốngNhân viênThống kê sách, xem thông tin sinh viên, quản lý mượn/trả, lọc sáchSinh viênMượn sách (tối đa 7 ngày), xem thông tin sách có sẵn

-Hoạt động của hệ thống quản lý thư viện sách:

* + Quản lý có thể thêm, xóa và sửa thông tin nhân viên và các chức năng của nhân viên
  + Quản lý có thể nhập sách từ các nhà cung cấp khác nhau
  + Nhân viên có thể thống kê sách xem mỗi đầu sách còn bao nhiêu quyển, và những sinh viên quá hạn mà chưa trả sách
  + Thủ thư có thể xem thông tin sinh viên và sách đang mượn.
  + Thủ thư có thể quản lý xem sinh viên mượn sách gì, thời gian mượn.
  + Thủ thư có thể xem, sửa thông tin sách. Lọc theo tác giả, ngày nhập kho, mã số sách, tên sách, loại sách, tình trạng và trạngthái mượn
  + Sinh viên có thể mượn tối đa 7 ngày, nếu mượn quá sẽ cảnh báo không được mượn nữa
  + Không thể mượn sách nếu không đủ số lượng trong thư viện, và những sách không có quyền mượn về
  + Xử lý các trường hợp quá hạn trả sách: Hệ thống sẽ tự động nhận diện các trường hợp mượn sách quá hạn, hiển thị thông báo cho người quản trị để xử lý, và có thể ghi nhận các khoản phạt nếu cần.

-Yêu cầu chức năng:

+ Người quản lý, nhân viên:

* Quản lý truy cập. sách
* Đăng ký/Đăng nhập
* Phân quyền truy cập
* Nhập sách mới, thống kê số lượng sách
* Quản lý mượn trả sách, sinh viên mượn sách
* Quản lý thông tin sinh viên

-Yêu cầu phi chức năng:

* Giao diện đồ họa thân thiện
* Tương thích Windows
* Hiệu năng ổn định
* Bảo mật đăng nhập

**2.3 Phân tích hệ thống**

* **Mô hình hóa chức năng hệ thống USECASE**

A diagram of a diagram

Description automatically generated with medium confidence

* **Mô tả vắn tắt các UseCase:**

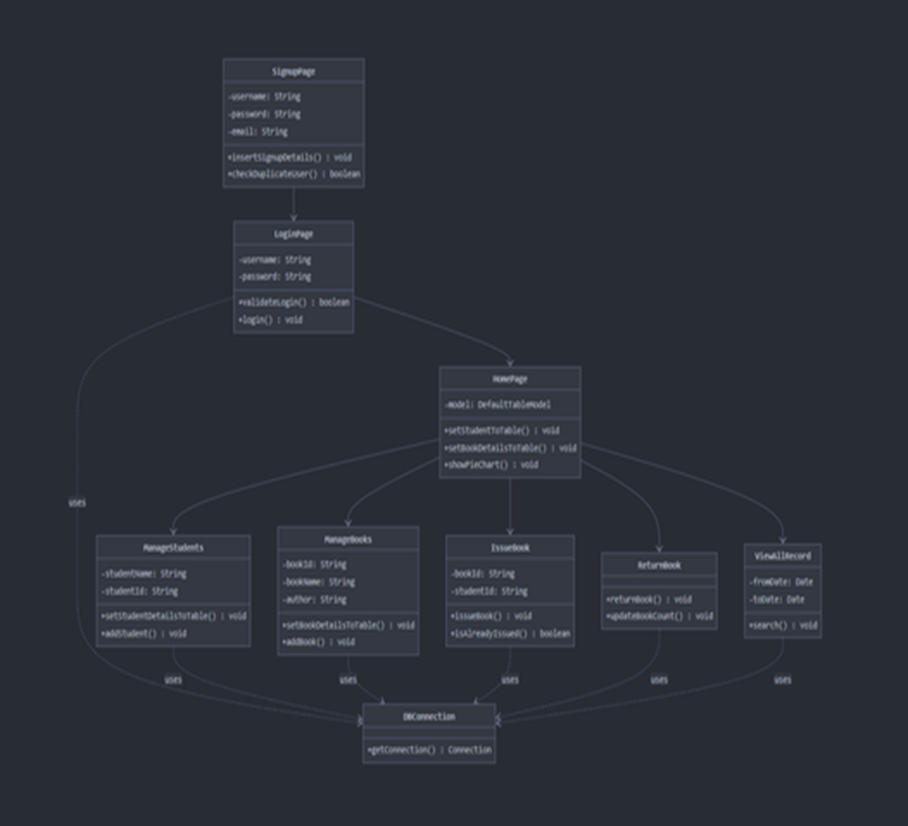
**A white sheet with black text

Description automatically generated**

**Ghi chú:**

* Admin: Quản trị viên có đầy đủ quyền quản lý hệ thống
* Người dùng: Tài khoản chung và đã được cấp thì mới có có thể xem thông tin cơ bản
* Các use case con (4-7, 9-13) được bao gồm trong use case cha tương ứng

**2.4Thiết kế hướng đối tượng:**

* **Sơ đồ lớp**

**2.4.1 Lớp kết nối cơ sở dữ liệu**

| **Lớp** | **Thuộc tính / Phương thức** | **Kiểu** | **Chức năng** |
| --- | --- | --- | --- |
| DBConnection | getConnection() | Connection | Tạo kết nối tới CSDL |
|  | closeConnection() | void | Đóng kết nối CSDL |

**2.4.2 Lớp người dùng**

| **Lớp** | **Thuộc tính / Phương thức** | **Kiểu** | **Chức năng** |
| --- | --- | --- | --- |
| User | username, password, email, role | String | Lưu thông tin người dùng |
|  | validateLogin() | boolean | Kiểm tra thông tin đăng nhập |
|  | authenticate() | boolean | Xác thực với CSDL |
| LoginPage | bt\_username, bt\_password | JTextField, JPasswordField | Trường nhập đăng nhập |
| (extends User) | validateLogin(), login() | boolean, void | Kiểm tra và xử lý đăng nhập |
| SignupPage | bt\_contact | JTextField | Nhập số điện thoại |
| (extends User) | insertSignupDetails(), checkDuplicateUser() | void, boolean | Đăng ký người dùng mới |

**2.4.3 Lớp quản lý sách**

| **Lớp** | **Thuộc tính / Phương thức** | **Kiểu** | **Chức năng** |
| --- | --- | --- | --- |
| Book | bookId, bookName, author, quantity, category, status | int, String | Thông tin sách |
| ManageBooks | setBookDetailsToTable() | void | Hiển thị danh sách sách |
|  | addBook(), updateBook(), deleteBook() | void | Thêm/sửa/xóa sách |
|  | searchBook() | List<Book> | Tìm kiếm sách |

**2.4.4 Lớp quản lý sinh viên**

| **Lớp** | **Thuộc tính / Phương thức** | **Kiểu** | **Chức năng** |
| --- | --- | --- | --- |
| Student | studentId, studentName, course, branch, contact | String | Thông tin sinh viên |
| ManageStudents | setStudentDetailsToTable() | void | Hiển thị danh sách sinh viên |
|  | addStudent(), updateStudent(), deleteStudent() | void | Thêm/sửa/xóa sinh viên |

**2.4.5 Lớp mượn sách**

| **Lớp** | **Thuộc tính / Phương thức** | **Kiểu** | **Chức năng** |
| --- | --- | --- | --- |
| IssueBook | bookId, studentId, issueDate, dueDate | int, String, Date | Thông tin mượn sách |
|  | issueBook() | void | Thực hiện mượn sách |
|  | isAlreadyIssued() | boolean | Kiểm tra đã mượn chưa |
|  | getIssueBookDetails() | void | Hiển thị thông tin mượn |

**2.4.6 Lớp trả sách**

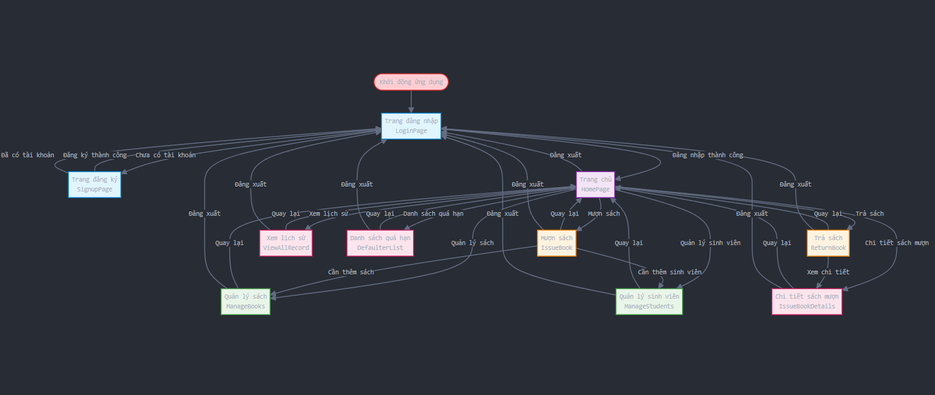
| **Lớp** | **Thuộc tính / Phương thức** | **Kiểu** | **Chức năng** |
| --- | --- | --- | --- |
| ReturnBook | bookId, studentId, returnDate | int, String, Date | Thông tin trả sách |
|  | returnBook() | void | Thực hiện trả sách |
|  | updateBookCount() | void | Cập nhật số lượng sách sau khi trả |
|  | calculateFine() | double | Tính tiền phạt nếu trả trễ |
|  | checkOverdue() | boolean | Kiểm tra sách có bị trễ hạn không |

**2.5 Data Flow ( Luồng dữ liệu )**

**A computer screen shot of a diagram

Description automatically generated**

| **Thực thể** | **Quy trình xử lý** | **Dữ liệu** | **Kho dữ liệu** |
| --- | --- | --- | --- |
| Nhân viên | Nhập mã sách + mã sinh viên | Thông tin mượn |  |
| → Hệ thống | IssueBook.getIssueBookDetails() | → Truy vấn sách | → Bảng sách |
| → Hệ thống | Kiểm tra số lượng sách |  |  |
| → Hệ thống | Kiểm tra sinh viên | → Truy vấn thông tin | → Bảng sinh viên |
| → Hệ thống | Kiểm tra số lượng sách mượn trước đó | → Truy vấn lịch sử | → Bảng mượn trả |
| → Hệ thống | Cập nhật mượn sách | Thêm bản ghi mới | → Ghi vào bảng mượn trả |
| → Hệ thống | Giảm số lượng sách | Cập nhật | → Bảng sách |

**2.6 UI Flow ( Luồng giao diện )**

**2.6.1 Sơ Đồ Navigation**

[Splash Screen]

↓

[Login Page] ←→ [Signup Page]

↓ (Login thành công)

[Home Page / Dashboard]

├── [Manage Books]

├── [Manage Students]

├── [Issue Books]

├── [Return Books]

├── [View All Records]

├── [Issue Book Details]

└── [Defaulter List]

**2.6.2 Workflow Chính**

Bắt đầu →

[Splash Screen] →

[Login Page] ←→ [Signup Page] →

(Đăng nhập thành công) →

[Dashboard] →

├─ Quản lý sách → Thêm/Sửa/Xóa sách

├─ Quản lý sinh viên → Thêm/Sửa/Xóa sinh viên

├─ Mượn sách → Nhập mã sách + mã sinh viên → Xác nhận

├─ Trả sách → Nhập thông tin trả → Cập nhật

├─ Thống kê → Xem báo cáo → Biểu đồ mượn – trả

└─ Danh sách trễ hạn → Hiển thị sinh viên mượn quá hạn

**2.7 Sequence Diagram ( Sơ đồ trình tự )**

User -> LoginPage: nhập tài khoản & mật khẩu

LoginPage -> DBConnection: getConnection()

DBConnection -> Database: SELECT \* FROM users WHERE ...

Database --> DBConnection: trả về kết quả

DBConnection -> LoginPage: trả về trạng thái đăng nhập

LoginPage -> HomePage: mở giao diện chính

User -> HomePage: chọn chức năng "Mượn sách"

HomePage -> IssueBook: mở giao diện mượn sách

User -> IssueBook: nhập mã sách + mã sinh viên

IssueBook -> Book: kiểm tra sách có sẵn

Book --> IssueBook: kết quả có/không

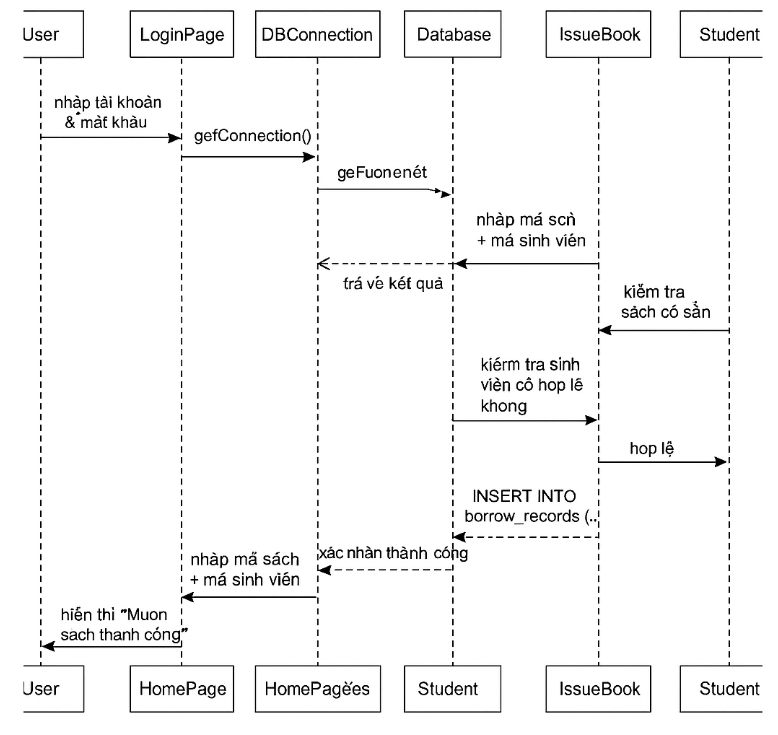
IssueBook -> Student: kiểm tra sinh viên có hợp lệ không

Student --> IssueBook: hợp lệ

IssueBook -> Database: INSERT INTO borrow\_records (...)

Database --> IssueBook: xác nhận thành công

IssueBook -> User: hiển thị "Mượn sách thành công"



**Mô hình Sequence Diagram**

**2.8 State Diagram (Trạng thái của Sách)**

[Start]

↓

[ Có sẵn ]

├── Mượn sách ───────→ [ Đang được mượn ]

├── Báo hư ──────────→ [ Không khả dụng ]

[ Đang được mượn ]

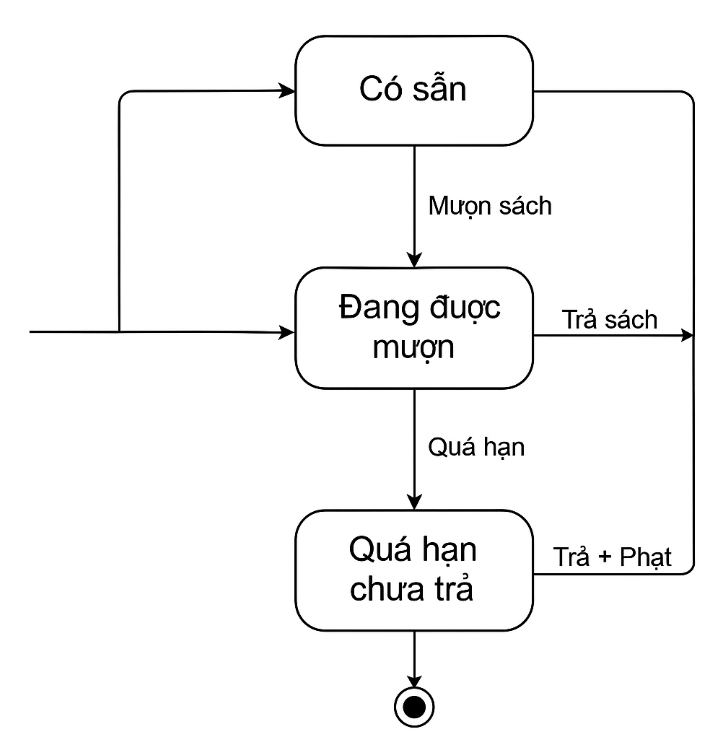
├── Trả sách ────────→ [ Có sẵn ]

├── Quá hạn ─────────→ [ Quá hạn chưa trả ]

[ Quá hạn chưa trả ]

└── Trả + Phạt ──────→ [ Có sẵn ]

[ Không khả dụng ] (kết thúc hoặc chờ sửa chữa)



Có sẵn

**Mô hình State Diagram**

**2.9 Activity Diagrams ( Quy trình mượn sách )**

Bắt đầu

→ Nhập thông tin: mã sách, mã sinh viên

→ Kiểm tra tình trạng sách

⮕ Nếu không có sách → Thông báo "Hết sách" → Kết thúc

→ Kiểm tra sinh viên hợp lệ

⮕ Nếu không hợp lệ → Thông báo "Không hợp lệ" → Kết thúc

→ Kiểm tra số sách đã mượn < 5

 ⮕ Nếu vượt quá giới hạn → Thông báo "Vượt số lượng cho phép" → Kết thúc

→ Tạo bản ghi mượn sách

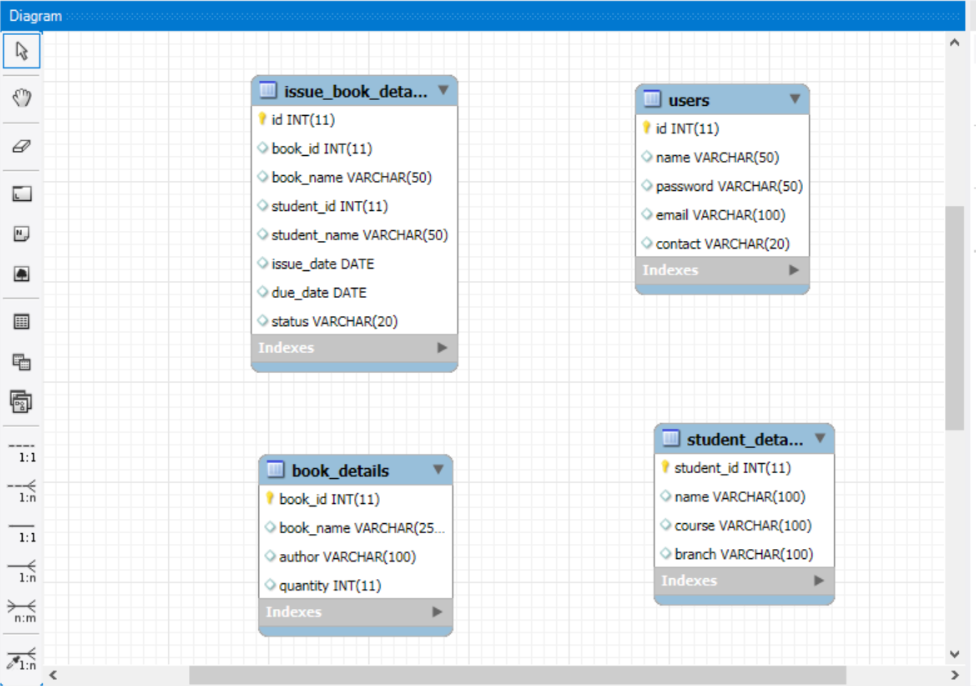
→ Trừ số lượng sách trong kho

A diagram of a flowchart

Description automatically generated→ Thông báo "Mượn sách thành công" - >Kết thúc

**Mô hình State Diagram**

**2.10 CSDL ( MySQL )**

- **Mô hình EER:**

**Ghi chú:**

users

* Lưu thông tin người dùng hệ thống (quản lý/thủ thư).
* Gồm: id, name, password, email, contact.

book\_details

* Quản lý sách: book\_id, book\_name, author, quantity.

student\_details

* Thông tin sinh viên: student\_id, name, course, branch.

issue\_book\_details

* + Ghi nhận mượn sách.
* Gồm: book\_id, student\_id, issue\_date, due\_date, status.

**Chương 3: Thiết kế giao diện**

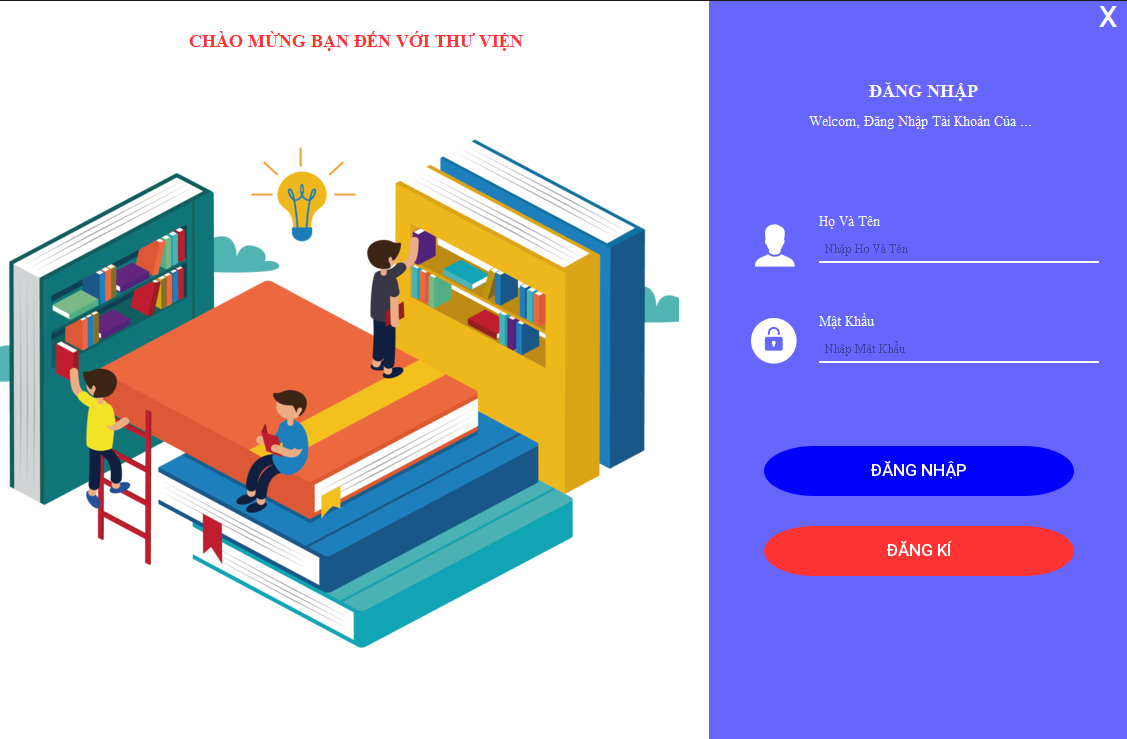
**3.1 Thiết kế hệ thống**

**- Thiết kế giao diện :**

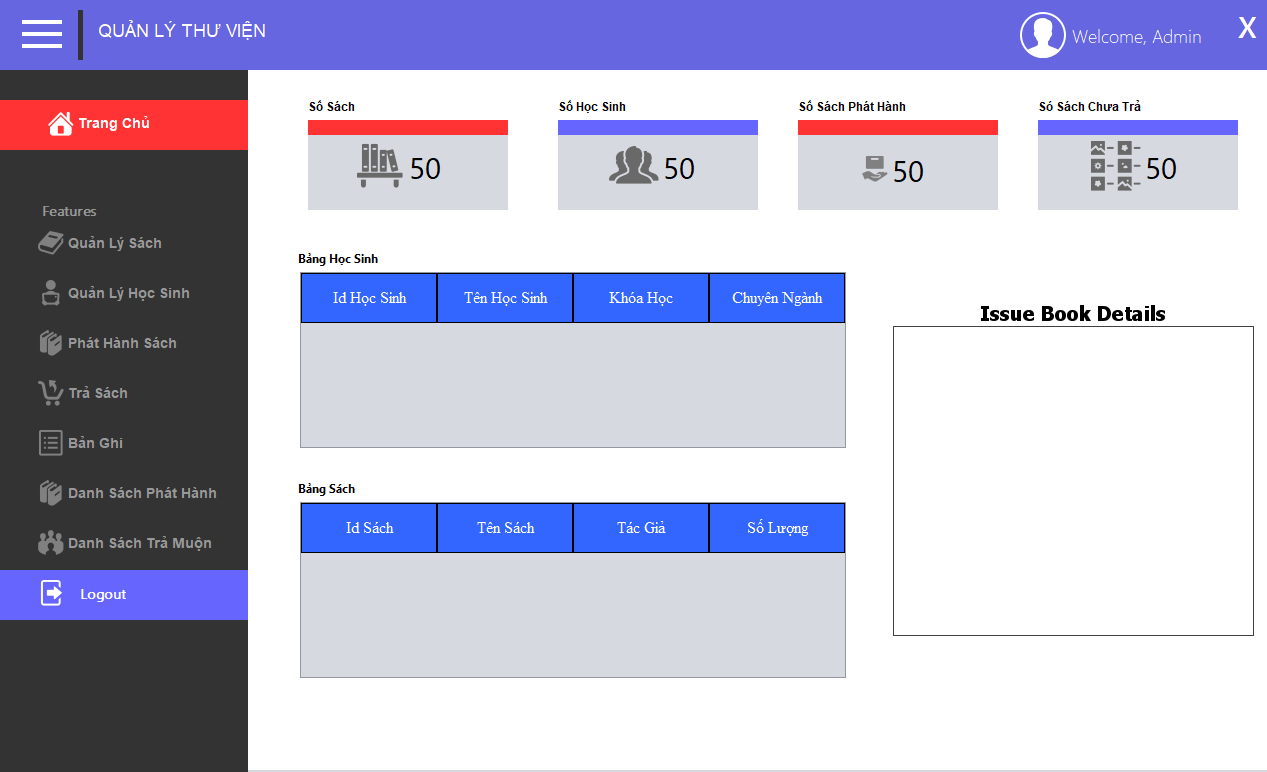
**A group of people sitting on books

Description automatically generated+ Phần đăng ký:**

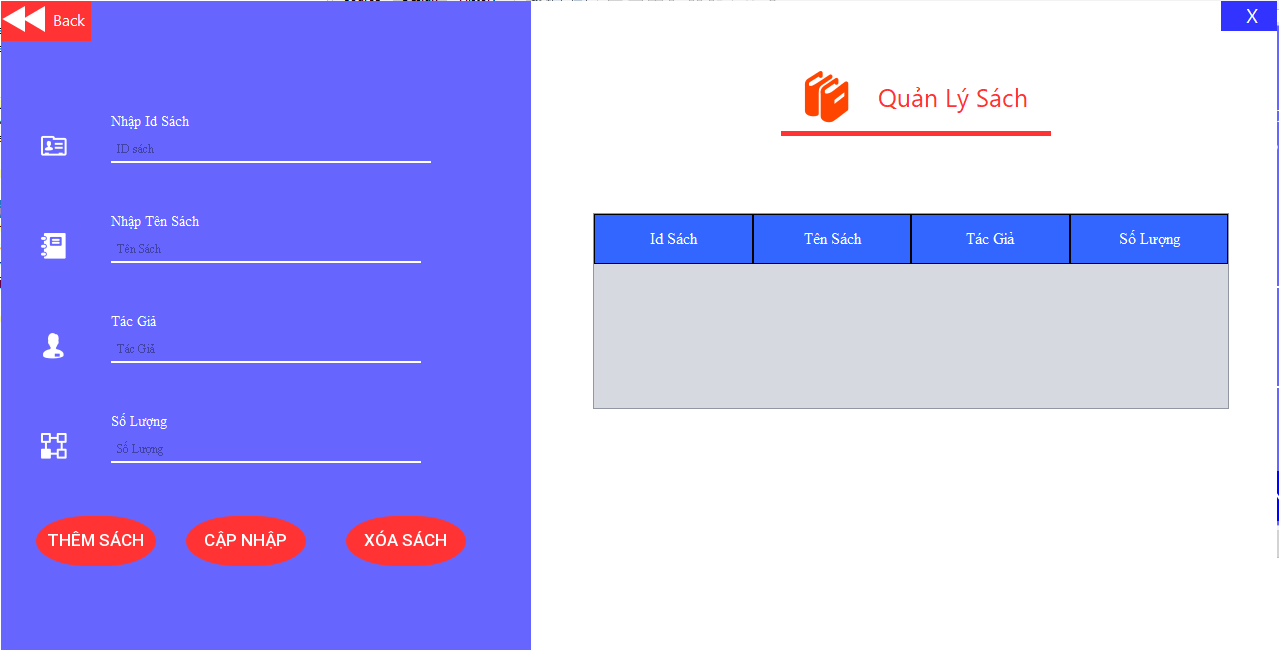
**+ Phần đăng nhập**

****

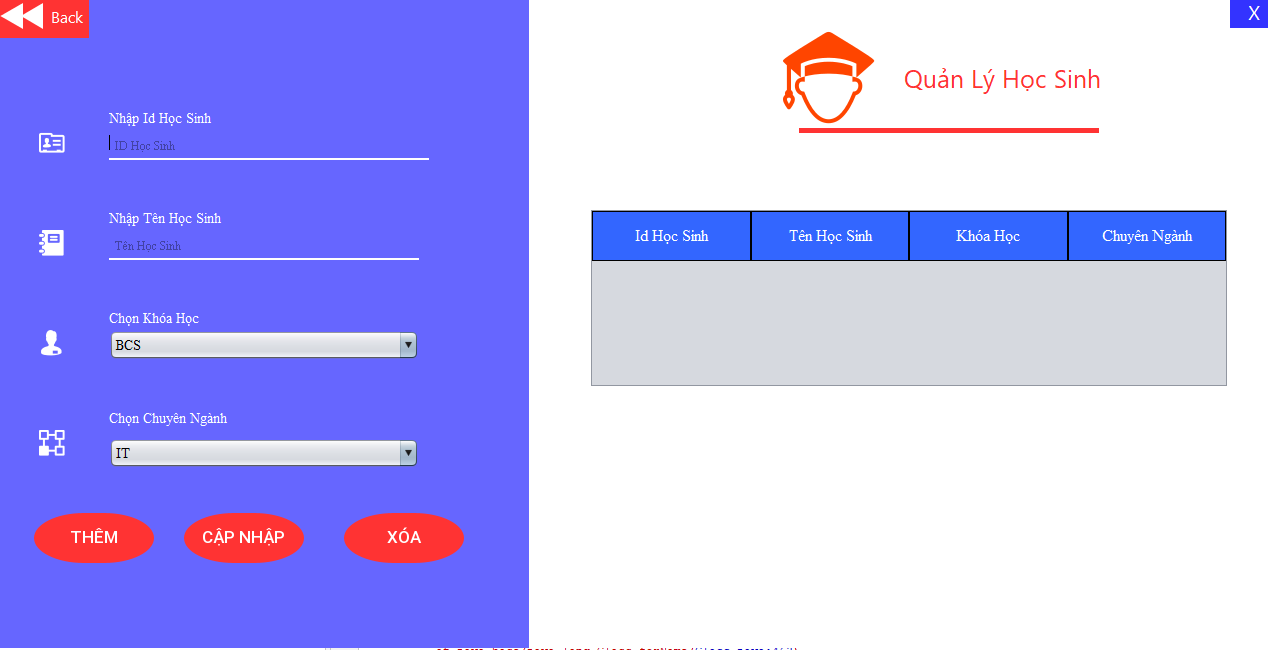
**+ Trang chủ của phần mềm:**

****

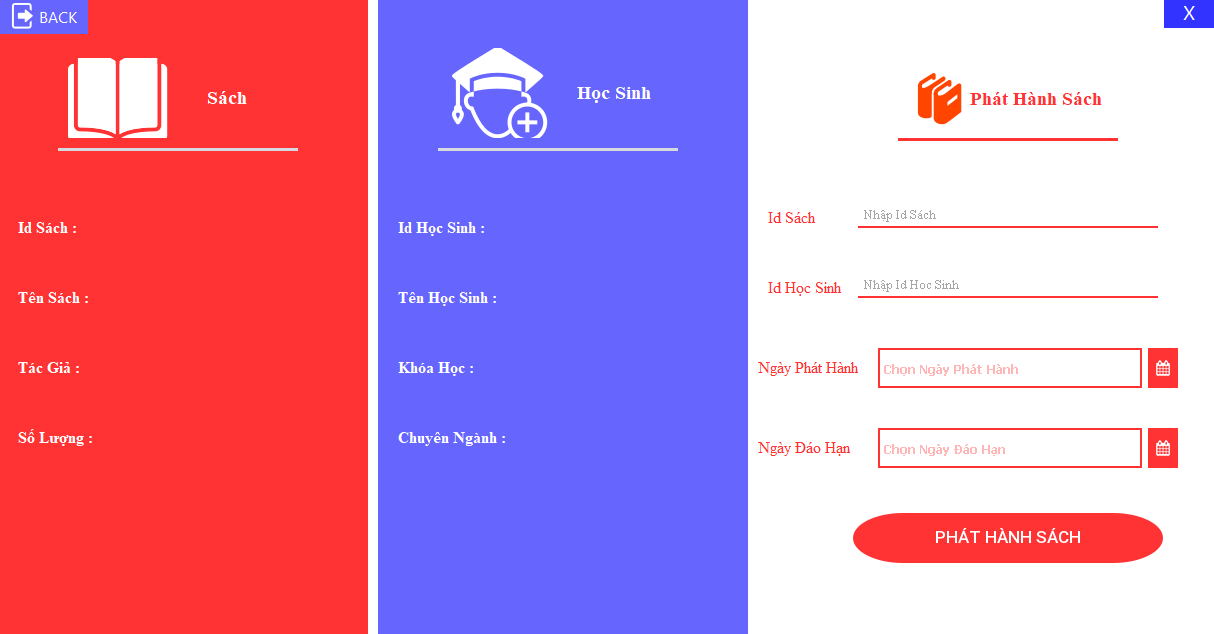
**+ Phần quản lý sách:**



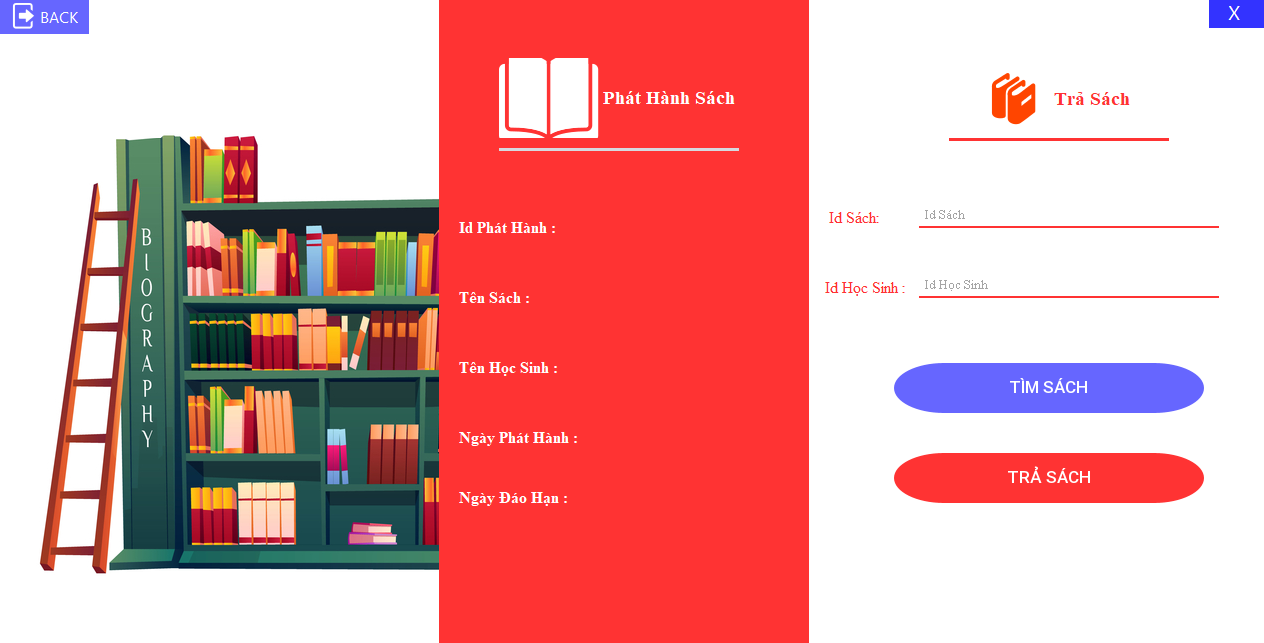
**+ Phần quản lý độc giả:**

****

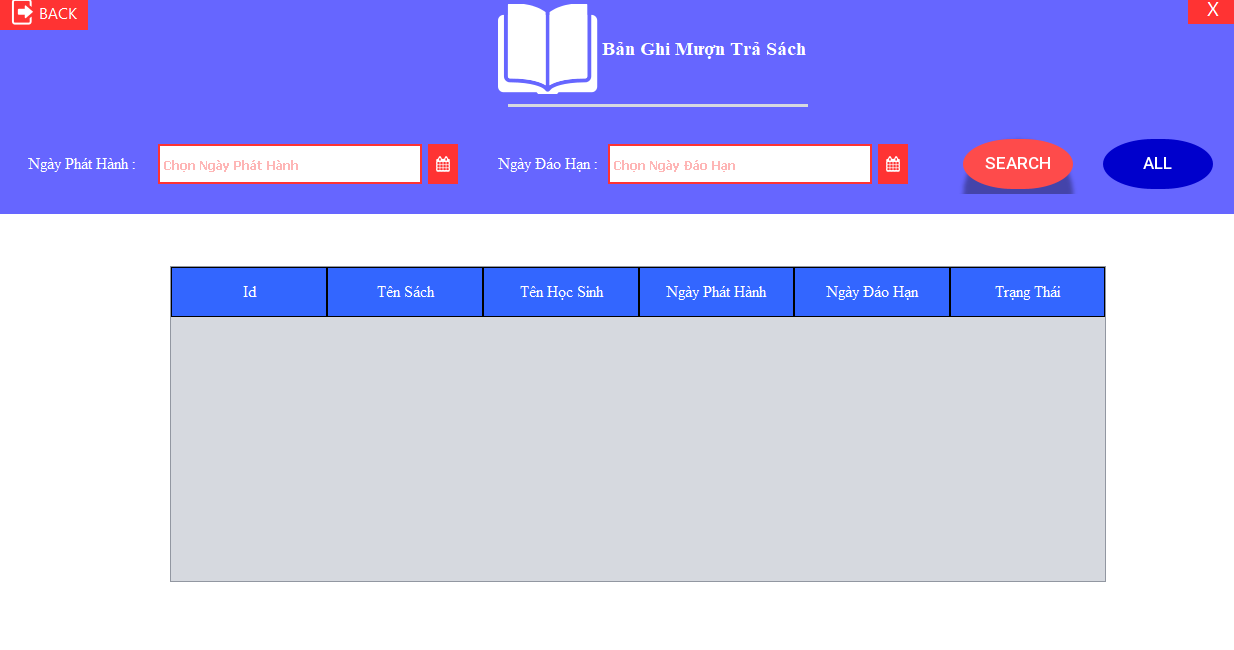
**+ Phần phát hành sách :**



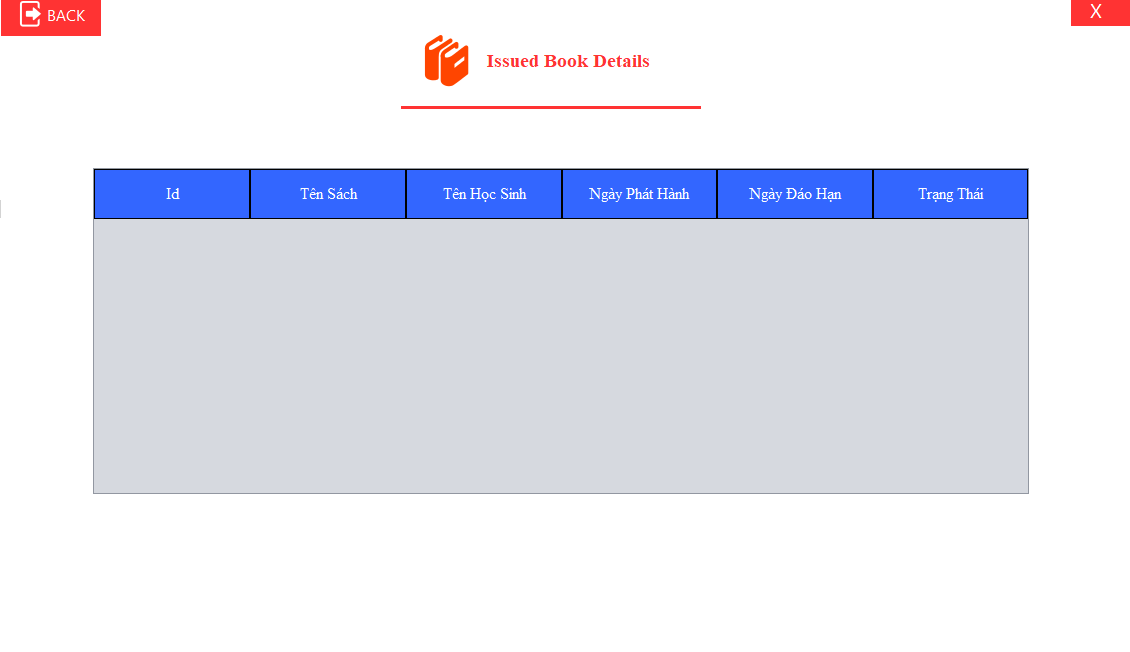
**+ Phần Quản lý mượn trả :**



**+ Phần tìm kiếm:**

****

**+ Phần hiển thị chi tiết sách đã mượn :**

****

**+ Phần danh sách trả muộn :**

**A screenshot of a computer

Description automatically generated**

**Chương 4: Triển khai phần mềm**

**Link code** : <https://github.com/CSE702025-Nhom-5/projectNhom5>

**4.1 Cách chạy ứng dụng:**

* 1. Mở XAMPP → Start MySQL hoặc truy cập : http://localhost/phpmyadmin/index.php?route=/sql&pos=0&db=library\_ms&table=book\_details
  2. Mở NetBeans → Import hoặc mở project Java của bạn.
  3. Thêm thư viện MySQL JDBC (mysql-connector-java.jar) vào project.
  4. Kiểm tra file DBConnection.java để đảm bảo đúng thông tin:
  5. URL kết nối DB
  6. Tên người dùng/password, hoặc đăng kí nếu chưa có tài khoản
  7. Run file chính LoginPage.java

**4.2 Kiểm thử chức năng**

**- Sử dụng các Test Case để xác minh:**

* Thêm sách thành công, Mượn trả đúng hạn ,Tìm kiếm theo tên/tác giả
* Xử lý lỗi khi kết nối sai hoặc nhập thiếu thông tin

**- Một số lỗi thường gặp khi triển khai**

**A screenshot of a chat

Description automatically generated**

**4.3 Test case và kịch bản kiểm thử**

**4.3.1 Kiểm thử giao diện (UI test cases)**

* Kịch bản đăng nhập, đăng ký, thêm/sửa/xóa sách và sinh viên, mượn/trả sách,...

**4.3.2 Kiểm thử API (API test cases)**

* Mục tiêu: đảm bảo các endpoint REST hoạt động đúng chức năng, bảo mật và xử lý dữ liệu chính xác.
* Cấu trúc test case (theo chuẩn):
* Test ID (ví dụ: TC\_API\_001)
* Tiêu đề tóm tắt chức năng
* Mô tả mục đích test
* Điều kiện tiên quyết
* Bước thực hiện (method, URL, header, payload
* Kết quả mong đợi (status code, structure)
* Kết quả thực tế
* Kết luận Pass/Fail

**4.3.3 Kiểm thử tích hợp & chạy tự động**

Kết hợp test API và UI tests, kiểm tra luồng end‑to‑end**.**

**4.4 Đa luồng và Tiến trình xử lý**

* Nhóm dự kiến áp dụng đa luồng (SwingWorker, Thread) cho một số chức năng trong giao diện đồ họa, như:
* Tải danh sách sinh viên/sách
* Hiển thị biểu đồ thống kê
* Mượn/trả sách có thời gian xử lý

**4.5 Định hướng phát triển dịch vụ API**

* Trong tương lai, hệ thống có thể được mở rộng thành ứng dụng web hoặc mobile. Khi đó, việc triển khai API là cần thiết để đồng bộ dữ liệu và mở rộng hệ thống.
* Các API cơ bản có thể bao gồm:
* GET /api/books – Lấy danh sách sách
* POST /api/borrow – Mượn sách
* POST /api/return – Trả sách
* GET /app/students/{id}/history – Lịch sử mượn

**Chương 5: Kết luận**

**5.1. Kết luận**

* Phần mềm Quản lý thư viện sách được xây dựng nhằm hỗ trợ thủ thư, cán bộ thư viện và sinh viên trong công tác quản lý, tra cứu, mượn – trả sách một cách thuận tiện và hiệu quả.
* Hệ thống được lập trình bằng ngôn ngữ Java, sử dụng giao diện đồ họa Swing và kết nối cơ sở dữ liệu MySQL. Giao diện thân thiện, dễ sử dụng; các chức năng cơ bản như quản lý sách, sinh viên, mượn – trả, thống kê đã được triển khai đầy đủ và hoạt động ổn định.
* Thông qua việc triển khai hệ thống, nhóm đã củng cố được kiến thức về kỹ thuật phần mềm, thiết kế giao diện, mô hình hóa hệ thống và tổ chức dữ liệu. Đồng thời cũng hiểu rõ hơn về quy trình xây dựng một phần mềm từ khâu phân tích đến cài đặt và kiểm thử.

**5.2. Đánh giá kết quả đạt được**

* Giao diện phần mềm được thiết kế trực quan, dễ thao tác, thân thiện với người dùng.
* Các chức năng thêm, sửa, xóa và tìm kiếm sách, sinh viên đều thực hiện được đúng yêu cầu.
* Chức năng mượn và trả sách xử lý chính xác, đảm bảo tính toàn vẹn dữ liệu.
* Phần thống kê cơ bản giúp quản lý dễ theo dõi tình trạng mượn – trả sách và số lượng sách còn trong kho.
* Hệ thống tương đối ổn định, dễ triển khai trên các máy có cài đặt môi trường Java và MySQL.

**5.3. Hướng phát triển trong tương lai**

Để phần mềm hoàn thiện hơn và đáp ứng được nhu cầu sử dụng thực tế ở quy mô lớn hơn, nhóm đề xuất một số hướng phát triển như sau:

* Phát triển thêm phiên bản web hoặc mobile:
* Cho phép người dùng tra cứu sách, đăng nhập và theo dõi lịch sử mượn – trả qua trình duyệt hoặc điện thoại. Có thể sử dụng các công nghệ như Spring Boot (cho web), hoặc Flutter (cho ứng dụng di động).
* Tăng cường bảo mật tài khoản:
* Bổ sung tính năng mã hóa mật khẩu, xác thực hai bước (2FA), hoặc đăng nhập qua email.
* Tích hợp thông báo nhắc hạn:
* Gửi email hoặc thông báo tại giao diện khi sách gần đến hạn hoặc đã quá hạn trả.
* Kết nối với nền tảng lưu trữ đám mây:
* Cho phép quản lý và sao lưu dữ liệu trực tuyến, hỗ trợ việc sử dụng phần mềm ở nhiều máy khác nhau hoặc nhiều chi nhánh thư viện.
* Cải tiến giao diện người dùng:
* Sử dụng thêm các thư viện giao diện hiện đại hơn hoặc chuyển sang JavaFX để tăng trải nghiệm người dùng.

Thông qua các định hướng trên, nhóm mong muốn phần mềm không chỉ dừng lại ở mức quản lý thư viện nội bộ, mà có thể trở thành nền tảng có thể mở rộng và triển khai ở nhiều đơn vị giáo dục khác nhau.

**TÀI LIỆU THAM KHẢO**

Sommerville, I. (2016). Software Engineering (10th Edition). Pearson Education.

Pressman, R. S., & Maxim, B. R. (2014). Software Engineering: A Practitioner’s Approach (8th Edition). McGraw-Hill Education.

Freeman, E., Robson, E., Bates, B., & Sierra, K. (2008). Head First Design Patterns. O'Reilly Media.

Bell, I. (2020). Engineering Software Products – An Introduction to Modern Software Engineering. Pearson Education.

Cũng như được chắt lọc và tổng hợp từ nhiều kênh youtube khác nhau