ĐẠI HỌC BÁCH KHOA HÀ NỘI

**Trường Công nghệ thông tin và truyền thông**

----- 🙡 🕮 🙣 -----



**Báo Cáo Lập Trình Hướng Đối Tượng**

***Đề tài:***

**Thiết kế hướng đối tượng cho bài toán**

**Quản lý thư viện**

Giảng viên: **Lê Đức Hậu**

**Nhóm 2:**

|  |  |
| --- | --- |
| Họ và tên | MSSV |
| Trần Đình Nhật Minh | 20225746 |
| Nguyễn Duy Khánh | 20225865 |
| Nguyễn Huy Hoàng | 20225845 |
| Phan Hồng Minh | 20225888 |

Trần Công Minh 20225745

**Hà Nội, năm 2024**

Table of Contents

[Table of Figures 3](#_Toc186064364)

[I. Phân công thành viên 4](#_Toc186064365)

[II. Mô tả dự án 4](#_Toc186064366)

[ *Các thực thể chính***:** 4](#_Toc186064367)

[ *Các chức năng chính***:** 4](#_Toc186064368)

[1. Quản Lý Sách 4](#_Toc186064369)

[2. Quản Lý Độc Giả 4](#_Toc186064370)

[3. Quản Lý Phiếu Mượn/Trả 5](#_Toc186064371)

[4. Thống Kê 5](#_Toc186064372)

[III. Thiết kế hệ thống 5](#_Toc186064373)

[*a.* *Tạo package và class* 5](#_Toc186064374)

[1. Package code 5](#_Toc186064375)

[2. Package Function 6](#_Toc186064376)

[3. Package GUI 11](#_Toc186064377)

[4. Package DB 13](#_Toc186064378)

[*b.* *Class Diagram* 14](#_Toc186064379)

[ *Package UI* 14](#_Toc186064380)

[ *Package Code* 15](#_Toc186064381)

[ *Package Java và JavaFX* 15](#_Toc186064382)

[*c.* *Usecase Diagram* 17](#_Toc186064383)

[IV. Database 18](#_Toc186064384)

[V. Các tính chất cơ bản 18](#_Toc186064385)

[VI. Demo 19](#_Toc186064386)

Table of Figures

[Hình 1: PackageUI ClassDiagram 13](#_Toc185973278)

[Hình 2: Package Code ClassDiagram 14](#_Toc185973279)

[Hình 3*:* Package Java và JavaFX ClassDiagram 14](#_Toc185973280)

[Hình 4: Usecase Diagram 15](#_Toc185973281)

[Hình 5: Database 16](#_Toc185973283)

[Hình 6 : Màn hình đăng nhập 17](#_Toc185973285)

[Hình 7: Màn hình chính 17](#_Toc185973286)

[Hình 8: Quản lý sách 18](#_Toc185973287)

[Hình 9: Quản lý người mượn 19](#_Toc185973288)

[Hình 10: Quản lý phiếu mượn 20](#_Toc185973289)

[Hình 11: Thống kê sách chưa trả 20](#_Toc185973290)

[Hình 12: Sách hỏng 21](#_Toc185973291)

[Hình 13: Sách trả muộn 21](#_Toc185973292)

1. **Phân công thành viên**

1. Trần Đình Nhật Minh 20225746 :Tạo package UI, package code, tạo giao diện,tạo class Diagram

2. Nguyễn Duy Khánh 20225845 :Thiết kế database, tạo class Login, tạo usecase Diagram, thuyết trình

3. Nguyễn Huy Hoàng 20225845 :Làm Slide

4. Phan Hồng Minh 20225888 :

5. Trần Công Minh 20225745 :

1. **Mô tả dự án**

* Xây dựng một hệ thống quản lý thông tin thư viện cơ bản, bao gồm các chức năng như: quản lý sách, quản lý độc giả, quản lý mượn trả sách,...
* Công cụ sử dụng: Java, SQL, JavaFx
* *Các thực thể chính***:**
  + **Sách:** Thông tin về sách (m. sách, tên sách, tác giả, nhà xuất bản, số lượng,...).
  + **Độc giả:** Thông tin về độc giả (m. độc giả, tên độc giả, ngày sinh, địa chỉ,...).
  + **Mượn trả:** Thông tin về việc mượn trả sách (m. mượn, ngày mượn, ngày trả dự kiến,ngày trả thực tế,...).
  + **Thể loại:** Phân loại sách theo thể loại (viễn tưởng, kinh dị, lịch sử,...).
* *Các chức năng chính***:**

### 1. Quản Lý Sách

Hệ thống cho phép thư viện quản lý danh mục sách, bao gồm:

* Thêm mới sách: Nhập thông tin sách bao gồm tên sách, tác giả, nhà xuất bản, năm xuất bản, thể loại, số lượng.
* Tìm kiếm sách: Tìm kiếm sách theo nhiều tiêu chí.
* Xóa sách: Loại bỏ những sách không còn trong danh mục.

### 2. Quản Lý Độc Giả

Hệ thống cung cấp các chức năng quản lý thông tin độc giả:

* Thêm mới độc giả: Lưu trữ thông tin họ tên, địa chỉ, số điện thoại, email.
* Xóa độc giả: Loại bỏ độc giả không còn tham gia thư viện.
* Tìm kiếm độc giả: Theo tên, số điện thoại hoặc email.

### 3. Quản Lý Phiếu Mượn/Trả

* Thêm phiếu mượn: Nhập thông tin phiếu bao gồm độc giả, sách mượn, ngày mượn, và ngày trả dự kiến.
* Cập nhật phiếu trả: Lưu trữ thông tin khi độc giả trả sách.
* Theo dõi sách trễ hẹn: Tự động danh sách các phiếu quá hạn.
* Xóa phiếu : Xóa phiếu khỏi bảng.
* Tìm kiếm phiếu theo nhiều tiêu chí.
* Cập nhật ngày trả và tình trạng hư hỏng của sách

### 4. Thống Kê

Hệ thống cung cấp các báo cáo thống kê quan trọng:

* Số người mượn: Thống kê tổng số độc giả.
* Tổng số sách: Thống kê tổng số sách trong kho.
* Số phiếu mượn: Tổng số phiếu mượn sách.
* Số phiếu quá hạn: Thống kê tổng số các phiếu trễ hẹn và hiển thị ra bảng.
* Số sách hỏng: Danh sách và tình trạng sách bị hư hỏng và hiển thị ra bảng.
* Số sách chưa trả: Thống kê tất cả các phiếu mượn chưa trả và hiển thị ra bảng.

1. **Thiết kế hệ thống**
   1. *Tạo package và class*

Hệ thống gồm 2 package chính: code và UI

1. Package code

* Đây là package chứa tất cả các lớp cơ bản phục vụ cho hệ thống quản lý bao gồm:
  + - * **Sách (Book):**
        + Thuộc tính: Mã sách, tên sách, tác giả, nhà xuất bản, số lượng, thể loại.
      * **Độc giả (Reader):**
        + Thuộc tính: Mã độc giả, tên độc giả, số điện thoại, địa chỉ.
      * **Mượn trả (Borrow):**
        + Thuộc tính: Mã mượn, ngày mượn, ngày trả dự kiến, ngày trả thực tế, sách,độc giả, tình trạng
      * **Thể loại (Category):**
        + Thuộc tính: Mã thể loại, tên thể loại
* Quan hệ giữa các lớp

• Một sách thuộc một thể loại.

• Một độc giả có thể mượn nhiều sách.

• Một sách có thể được nhiều độc giả mượn.

• Một mượn trả liên kết với một sách và một độc giả.

1. Package Function

Đây là package chứa tất cả lớp thực hiện chức năng chính của hệ thống

* BaseUI
  + Lớp BaseUI kế thừa lớp Scene của JavaFX, cung cấp một giao diện cơ bản để thiết kế các màn hình khác trong ứng dụng.
  + Đây là lớp được thiết kế để cung cấp các phương thức tiện ích và bố cục giao diện chung cho các lớp con khác như:
    - Phươn thức khởi tạo của BaseUI: thiết lập giao diện cơ bản và lưu tham chiếu đến Stage chính của ứng dụng để sử dụng.
    - setting: Tạo khoảng cách giữa các lable
    - setBtn: Định dạng kích thước và kiểu hiển thị của một nút.
    - returnMenu: Chuyển từ giao diện hiện tại về giao diện menu chính.
    - layout2, layout3: các phương thức để quản lý bố cục của các phần tử hiển thị trên giao diện như
* BookManagement (quản lý sách)
  + Lớp BookManagement kế thừa lớp BaseUI dùng để quản lý các chức năng liên quan đến sách trong thư viện.
    - Các thuộc tính:
      * **TableView<Book> tableView**: Hiển thị danh sách sách trong giao diện.
      * **ObservableList<Book> bookList**: Danh sách các sách được quan sát và hiển thị trong TableView.
      * **Connection connection**: Kết nối tới cơ sở dữ liệu.
      * **TextField searchFieldByID, searchFieldByName,** **searchFieldByAuthor,** **searchFieldByCategory,** **searchFieldByQuantity,** **searchFieldByPublisher**: Các trường nhập liệu dùng để tìm kiếm sách theo các tiêu chí khác nhau.
      * **BorderPane root**: Layout chính của giao diện người dùng, chứa các phần tử giao diện.
    - Các phương thức chính:

Phương thức khởi tạo BookManagement(Stage primaryStage): Khởi tạo giao diện người dùng, các trường tìm kiếm và các nút hành động, đồng thời thiết lập kết nối cơ sở dữ liệu và tải danh sách sách từ cơ sở dữ liệu.

Layout1(int height, Node node1, Node node2, Node node3, Node node4): Định dạng vị trí các nút

searchBook(): Lọc các sách trong danh sách theo các tiêu chí tìm kiếm được nhập vào các trường tìm kiếm, hiển thị kết quả ra bảng

creatBook():Tạo một đối tượng Book mới từ các giá trị trong các trường tìm kiếm

fetchBooksFromDatabase(): Thực hiện câu lệnh SQL để lấy thông tin sách cùng với thông tin tác giả, nhà xuất bản, và thể loại từ các bảng liên quan trong cơ sở dữ liệu sau đó tạo đối tượng Book từ các kết quả trả về và thêm vào bookList

addBookToDatabase(Book book):

Kiểm tra xem tác giả đã tồn tại trong cơ sở dữ liệu chưa. Nếu chưa, thêm tác giả vào bảng author.

Thêm sách vào bảng book với các thông tin liên quan.

Cập nhật TableView bằng cách thêm sách vào bookList và làm mới bảng.

deleteBooks(): Xóa sách khỏi bookList và cơ sở dữ liệu

creatTableView():Tạo và hiển thị bảng (TableView) với thông tin sách.

BorrowManagement (Quản lý phiếu mượn)

* + Lớp BorrowManagement kế thừa từ BaseUI để quản lý các chức năng liên quan mượn trả sách trong thư viện.
    - Các thuộc tính:

**TableView<Borrow> table**: Bảng hiển thị thông tin mượn sách.

**TextField txtSearchLoanId**, **txtSearchReaderId**, **txtSearchBookId**, **txtSearchBorrowDate**, **txtSearchReturnDate**, **txtSearchReturnDateReturned**: Các ô nhập liệu cho người dùng tìm kiếm theo nhiều tiêu chí

**ComboBox<String> cbDamagedStatus**: Hộp chọn trạng thái hư hỏng của sách.

**Connection connection**: Đối tượng kết nối với cơ sở dữ liệu.

**ObservableList<Borrow> data**: Danh sách các giao dịch mượn sách.

* + - Các phương thức chính:

Phương thức khởi tạo BookManagement(Stage primaryStage): Khởi tạo giao diện người dùng, các trường tìm kiếm và các nút hành động, đồng thời thiết lập kết nối cơ sở dữ liệu và tải danh sách sách từ cơ sở dữ liệu.

Layout,layout4:Định dạng vị trí các nút và một số textField

fetchBorrowDataFromDatabase(): Thực hiện câu lệnh SQL để lấy thông tin mã mượn, mã người mượn,ngày mượn, hạn trả, ngày trả thực tế và tình trạng hư hỏng sau đó thêm vào bookList

searchBorrows ():Lọc các phiếu mượn trong danh sách theo các tiêu chí tìm kiếm được nhập vào các trường tìm kiếm, hiển thị kết quả ra bảng

createBorrow():Tạo phiếu mượn từ các thông tin được nhập vào từ các trường tìm kiếm rồi thêm vào danh sách và cơ sở dữ liệu

updateIsDamaged():Cập nhật tình trạng hư hỏng của sách khi được trả

updateReturndates():Cập nhật ngày trả vào danh sách và cơ sở dữ liệu

addBorrowToDatabase(Borrow borrow):

Thêm phiếu mượn vào bảng borrow với các thông tin liên quan.

Cập nhật TableView bằng cách thêm sách vào bookList và làm mới bảng.

deleteBorrows(): Xóa phiếu mượn khỏi borrowList và cơ sở dữ liệu

createBorrowTableView():Tạo bảng hiển thị các thông tin liên quan đến mượn trả sách

ReaderManagement(Quản lý người mượn)

* + Lớp ReaderManagement kế thừa lớp BaseUI dùng để quản lý các chức năng liên quan đến sách trong thư viện.
    - Các thuộc tính:
      * **TableView<Reader> tableView**: Hiển thị danh sách sách trong giao diện.
      * **ObservableList<Reader> bookList**: Danh sách các người mượn được quan sát và hiển thị trong TableView.
      * **Connection connection**: Kết nối tới cơ sở dữ liệu.
      * **TextField searchFieldByID, searchFieldByName,** **searchFieldByAddress,** **searchFieldByPhone**: Các trường nhập liệu dùng để tìm kiếm sách theo các tiêu chí khác nhau.
      * **BorderPane root**: Layout chính của giao diện người dùng, chứa các phần tử giao diện.
    - Các phương thức chính:

Phương thức khởi tạo readerManagement(Stage primaryStage): Khởi tạo giao diện người dùng, các trường nhập liệu và các nút hành động, đồng thời thiết lập kết nối cơ sở dữ liệu và tải danh sách sách từ cơ sở dữ liệu.

searchReaders():Tìm kiếm thông tìn người mượn theo dữ liệu được nhập vào và hiển thị ra bảng

deleteReaders():Xóa người mượn được nhập vào khỏi danh sách và cơ sở dữ liệu

fetchReadersFromDatabase():Thực hiện câu lệnh SQL để lấy thông tin liên quan đến người mượn sau đó thêm vào readerList

createReader():Tạo trường thông tin người mượn mới từ các dữ liệu được nhập vào

addReaderToDatabase(Reader reader): Thêm thông tin người mượn mới tạo vào danh sách và cơ sở dữ liệu

createReaderTableView():Tạo bảng hiện thị thông tin người mượn

* Statistic (Thống Kê)

Phương thức khởi tạo Statistic(Stage primaryStage): Khởi tạo giao diện người dùng, hiển thị các thống kê, tạo các nút hành động, đồng thời thiết lập kết nối cơ sở dữ liệu và tải danh sách sách từ cơ sở dữ liệu.

* + Thống kê số người mượn
    - countTotalBorrowers(): đếm tổng số người mượn
  + Thống kê tổng số sách trong thư viện
    - countTotalBookInStock():Đếm tổng số sách có trong thư viện
  + Thống kê số phiếu mượn
    - countTotalBooksBorrows: Đếm số phiếu mượn
  + Thống kê số sách hỏng và hiển thị ra bảng
    - CountDamagedBooks(): Đếm số sách hỏng

fetchDamagedBooksDataFromDatabase():Thực hiện câu lệnh SQL để lấy thông tin liên quan đến sách bị hỏng và thêm vào danh sách

* + - showTableDamaged(ObservableList<Borrow> borrowList)
  + Thống kê số phiếu trả quá hạn và hiển thị ra bảng
    - countOverdueLoans():Đếm số phiếu trả quá hạn
  + fetchOverdueLoanDataFromDatabase ():Thực hiện câu lệnh SQL để lấy thông tin liên quan đến sách trả sau thời hạn và thêm vào danh sách Thống kê số phiếu chưa trả sách và hiển thị ra bảng
    - countUnreturnedBooks ():Đếm số sách chưa được trả

fetch UnreturnedBooksDataFromDatabase():Thực hiện câu lệnh SQL để lấy thông tin liên quan đến sách chưa được trả và thêm vào danh sách

* + Phương thức tạo bảng cho sách trả quá hạn và chưa trả
    - showTableUnreturnAndOverDue(ObservableList<Borrow> borrowList)()

1. Package GUI

* BaseScene
  + Đây là một lớp trừu tượng kế thừa từ lớp Scene trong JavaFX cung cấp các tính năng và phương thức chung cho lớp Login và Menu sử dụng
    - Phương thức khởi tạo BaseScene(Stage primaryStage, double width, double height)
    - Phương thức trừu tượng setupUI():Đây là một phương thức trừu tượng, yêu cầu các lớp con (như Login, Menu) phải thực hiện và định nghĩa cách thiết lập giao diện người dùng của cảnh đó.
    - Phương thức setBtnStyle(Button btn, double width, double height): dùng để định dạng thiết kế cho nút bấm
* Login
  + Lớp Login kế thừa lớp Scene của Javafx để tạo giao diện đăng nhập và xử lý xác thực tài khoản
    - Các thuộc tính:

**txtUsername**: Là một đối tượng TextField dùng để nhập tên đăng nhập của người dùng.

**txtPassword**: Là một đối tượng PasswordField dùng để nhập mật khẩu, với mục đích bảo mật (ẩn mật khẩu khi nhập).

**root**: Là đối tượng BorderPane, là phần tử chứa toàn bộ giao diện của lớp này.

* + - Các phương thức:
      * + Login(Stage primaryStage): Phương thức này dùng để khởi tạo giao diện với các trường đăng nhập và nút đăng nhập
        + layout(int height, Node node1, Node node2, Node node3): Định dạng các ô đăng nhập và nút đăng nhập
        + setting(String label1, Node label2): sắp xếp ô đăng nhập và nhãn một cách hợp lý
        + authenticate (String username, String password): Kiểm tra thông tin đăng nhập trong cơ sở dữ liệu.
* Menu
  + Lớp Menu kế thừa từ lớp Scene của Javafx,cung cấp giao diện chính của ứng dụng, cho phép người dùng truy cập vào các chức năng quản lý bao gồm:
    - Quản lý sách.
    - Quản lý người mượn.
    - Quản lý phiếu mượn.
    - Thống kê.
  + Gồm 2 phương thức:
    - Phương thức khởi tạo của lớp Menu dùng để:

Tạo giao diện chính

Tiêu đề giao diện

Tạo các nút chức năng

Thêm phần tử vào giao diện

Định dạng các nút

Liên kết hành động cho nút

* UI
  + Lớp UI kế thừa từ lớp Application của JavaFX, cung cấp các phương thức chính để:
    - Khởi tạo giao diện ứng dụng.
    - Thiết lập giao diện đăng nhập.
    - Xử lý các lỗi phát sinh khi ứng dụng khởi chạy.
  + Gồm 2 phương thức:
    - Phương thức start(Stage primaryStage): Phương thức này được gọi khi ứng dụng JavaFX khởi chạy. Nó thiết lập giao diện chính và hiển thị cửa sổ đăng nhập.
    - Phương thức main(String[] args): Đây là phương thức khởi đầu của chương trình, được JVM gọi đầu tiên khi chạy ứng dụng.

1. Package DB

Chứa duy nhất lớp Dbconnect dùng để kết nối với cơ sở dữ liệu

* Phương thức connect(): kết nối đến cơ sở dữ liệu
  1. *Class Diagram*
* *Package UI*

*Ảnh có chứa văn bản, biểu đồ, Song song, Kế hoạch

Mô tả được tạo tự động*

*Ảnh có chứa văn bản, biểu đồ, Song song, Kế hoạch

Mô tả được tạo tự động*

*A close-up of a yellow paper

Description automatically generated*

Hình 1: PackageUI ClassDiagram

* *Package Code*

*Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, Song song

Mô tả được tạo tự động*

Hình 2: Package Code ClassDiagram

* *Package Java và JavaFX*

*Ảnh có chứa biểu đồ, ảnh chụp màn hình, hàng, Hình chữ nhật

Mô tả được tạo tự động*

*Ảnh có chứa hàng, ảnh chụp màn hình, biểu đồ, Hình chữ nhật

Mô tả được tạo tự động*

Hình 3*:* Package Java và JavaFX ClassDiagram

* 1. *Usecase Diagram*

*Ảnh có chứa hàng, biểu đồ, văn bản, ảnh chụp màn hình

Mô tả được tạo tự động*

Hình 4: Usecase Diagram

# **Database**

A screenshot of a computer

Description automatically generated

Hình 5: Database

# **Các tính chất cơ bản**

* + - 1. Tính đóng gói: sử dụng các thuộc tính private để bảo vệ dữ liệu và cung cấp các phương thức getter, setter để truy cập hoặc thay đổi giá trị một cách an toàn, giúp đảm bảo tính toàn vẹn dữ liệu và kiểm soát quyền truy cập.
      2. Tính kế thừa: Các lớp ReaderManagement, Statistic

BorrowManagement, BookManagement kế thừa lớp BaseUI hay các lớp Login và Menu kế thừa lớp BaseScene để tái sử dụng các phương thức đã được định nghĩa sẵn giúp mã nguồn ngắn gọn, dễ bảo trì và mở rộng

* + - 1. Tính đa hình: Phương thức setupUI() và setBtnStyle() trong lớp BaseScene được các lớp con Login và Menu định nghĩa lại để có thể thiết lập giao diện một cách phù hợp
      2. Tính trừu tượng: Lớp trừu tượng BaseScene định nghĩa phương thức trừu tượng setupUI() buộc các lớp con (Login, Menu) phải định nghĩa cụ thể cách thiết lập giao diện.

1. **Demo**
   * + 1. Màn hình đăng nhập

A screenshot of a computer

Description automatically generated

Hình 6 : Màn hình đăng nhập

* + - 1. Màn hình chính

A screenshot of a computer

Description automatically generated

Hình 7: Màn hình chính

* + - 1. Quản lý sách

A screenshot of a computer

Description automatically generated

Hình 8: Quản lý sách

* + - 1. Quản lý người mượn

A screenshot of a computer

Description automatically generated

Hình 9: Quản lý người mượn

* + - 1. Quản lý phiếu mượn

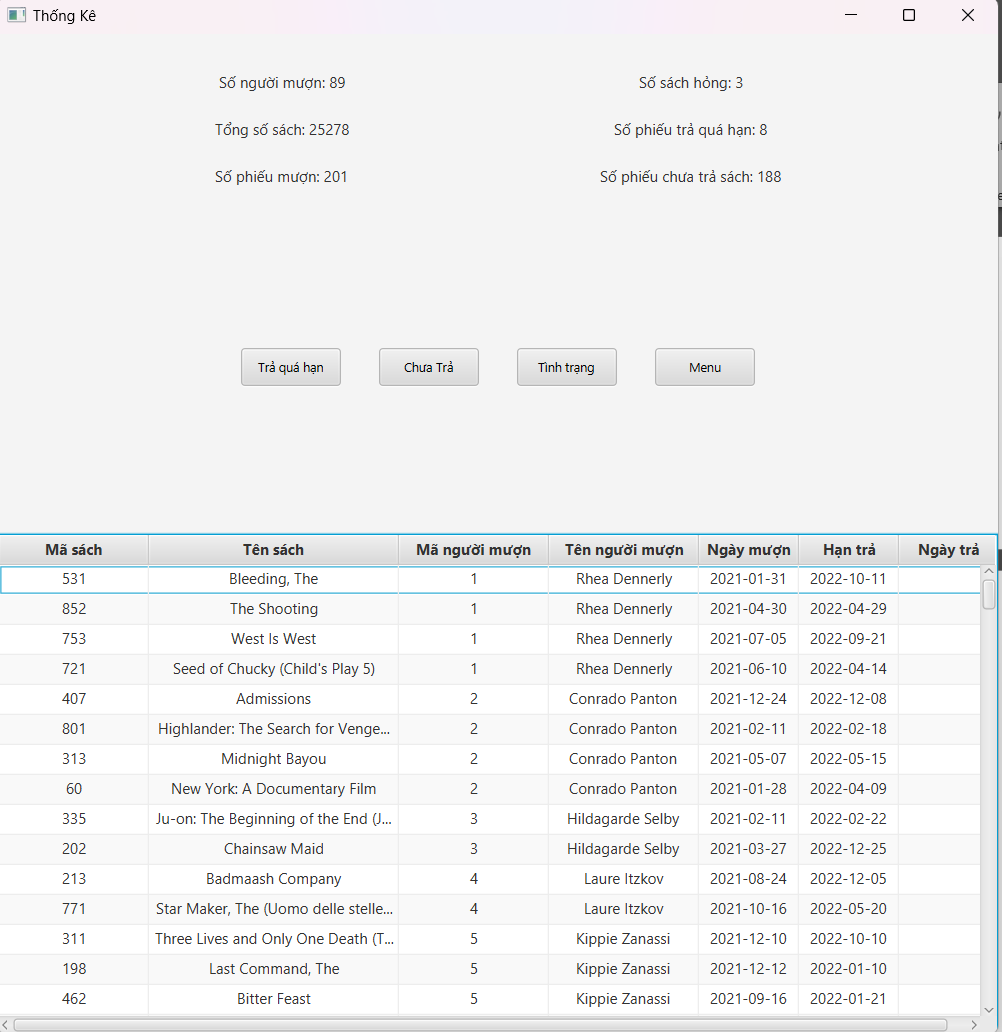
A screenshot of a computer

Description automatically generated

Hình 10: Quản lý phiếu mượn

* + - 1. Thống kê

6.1 Sách chưa trả



Hình 11: Thống kê sách chưa trả

6.2 Sách hỏng

A screenshot of a computer

Description automatically generated

Hình 12: Sách hỏng

6.3 Sách trả muộn

A screenshot of a computer

Description automatically generated

Hình 13: Sách trả muộn