**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

======\*\*\*======



BÁO CÁO THÍ NGHIỆM/THỰC NGHIỆM THUỘC HỌC PHẦN:

LẬP TRÌNH JAVA

**XÂY DỰNG PHẦN MỀM QUẢN LÝ SÁCH**

| Giảng viên: | ThS. Nguyễn Thái Cường |
| --- | --- |
| Nhóm – Lớp: | 9 |
| Thành viên: | Nguyễn Thị Anh Phương-2022601040 |
|  | Phạm Thị Quyên-2022600278 |
|  | Bùi Thị Kiều Trang-2022607581 |
|  | Nguyễn Công Thành-2022600390 |
| Lớp: | 20232IT6019009 |

Hà Nội, Năm 2024

**MỤC LỤC**

[**LỜI MỞ ĐẦU 4**](#_heading=h.gjdgxs)

[**PHẦN 1: GIỚI THIỆU 6**](#_heading=h.30j0zll)

[**1.**](#_heading=h.1fob9te) **Tên đề tài 6**

[**2.**](#_heading=h.3znysh7) **Lý do chọn đề tài 6**

[**3.**](#_heading=h.2et92p0) **Xác định nội dung học tập và các kỹ năng, kiến thức then chốt 6**

[**4.**](#_heading=h.tyjcwt) **Công cụ thực hiện đề tài 7**

[**PHẦN 2: KẾT QUẢ NGHIÊN CỨU 10**](#_heading=h.3dy6vkm)

[**2.1.**](#_heading=h.1t3h5sf) **Giới thiệu 10**

[**2.2.**](#_heading=h.4d34og8) **Khảo sát hệ thống 10**

[2.2.1.](#_heading=h.2s8eyo1) Khảo sát sơ bộ 10

[2.2.2.](#_heading=h.17dp8vu) Tài liệu đặc tả yêu cầu người dùng 11

[2.2.2.1.](#_heading=h.3rdcrjn) Yêu cầu chức năng 11

[2.2.2.2.](#_heading=h.26in1rg) Yêu cầu phi chức năng 12

[**2.3.**](#_heading=h.lnxbz9) **Phân tích hệ thống 13**

[2.3.1. Mô hình hóa chức năng hệ thống 13](#_heading=h.35nkun2)

[2.3.1.1.](#_heading=h.1ksv4uv) Biểu đồ use case tổng quát 13

[2.3.1.2.](#_heading=h.44sinio) Biểu đồ use case thứ cấp 14

[2.3.1.3.](#_heading=h.2jxsxqh) Mô tả chi tiết các use case 15

[2.3.2. Mô hình hóa dữ liệu và cài đặt hệ thống 47](#_heading=h.z337ya)

[2.3.2.1.](#_heading=h.3j2qqm3) Các yêu cầu dữ liệu 47

[2.3.2.2.](#_heading=h.1y810tw) Biểu đồ thực thể liên kế mức logic 48

[2.3.2.3.](#_heading=h.4i7ojhp) Sơ đồ lớp dữ liệu 48

[2.3.3.](#_heading=h.2xcytpi) Giao diện hệ thống 51

[2.3.3.1.](#_heading=h.1ci93xb) Đối với người dùng 51

[2.3.3.2.](#_heading=h.3whwml4) Đối với thủ thư 54

[**2.4.**](#_heading=h.2bn6wsx) **Thực hiện bài toán 59**

[2.4.1.](#_heading=h.qsh70q) Đối với người dùng 59

[2.4.1.1.](#_heading=h.3as4poj) Đăng ký (Phạm Thị Quyên) 59

[2.4.1.2.](#_heading=h.1pxezwc) Đăng nhập (Phạm Thị Quyên) 62

[2.4.1.3.](#_heading=h.49x2ik5) Tìm kiếm (Phạm Thị Quyên) 64

[2.4.1.4.](#_heading=h.2p2csry) Mượn sách (Bùi Thị Kiều Trang) 65

[2.4.1.5.](#_heading=h.147n2zr) Quản lý tài khoản (Bùi Thị Kiều Trang) 68

[2.4.1.6.](#_heading=h.3o7alnk) Xem lịch sử mượn sách (Bùi Thị Kiều Trang) 70

[2.4.2.](#_heading=h.23ckvvd) Đối với thủ thư 72

[2.4.2.1.](#_heading=h.ihv636) Đăng nhập (Nguyễn Thị Anh Phương) 72

[2.4.2.2.](#_heading=h.32hioqz) Thêm sách (Nguyễn Thị Anh Phương) 75

[2.4.2.3.](#_heading=h.1hmsyys) Sửa thông tin sách (Nguyễn Thị Anh Phương) 77

[2.4.2.4.](#_heading=h.41mghml) Xóa thông tin sách (Nguyễn Công Thành) 80

[2.4.2.5.](#_heading=h.2grqrue) Đổi mật khẩu (Nguyễn Thị Anh Phương) 82

[2.4.2.6.](#_heading=h.vx1227) Khóa/mở tài khoản (Nguyễn Công Thành) 85

[2.4.2.7.](#_heading=h.3fwokq0) Phiếu vi phạm (Nguyễn Công Thành) 86

[2.4.2.8.](#_heading=h.1v1yuxt) Phê duyệt mượn/trả sách (Nguyễn Công Thành) 89

[2.4.2.9.](#_heading=h.4f1mdlm) Tìm kiếm sách (Nguyễn Công Thành) 91

[**PHẦN 3: KẾT LUẬN VÀ BÀI HỌC KINH NGHIỆM 93**](#_heading=h.2u6wntf)

[**3.1. Nội dung đã thực hiện 93**](#_heading=h.19c6y18)

[3.1.1. Kiến thức và kỹ năng đã học được 93](#_heading=h.3tbugp1)

[3.1.2. Chuẩn đầu ra đạt được 93](#_heading=h.28h4qwu)

[3.1.3. Bài học kinh nghiệm 93](#_heading=h.nmf14n)

[**3.2. Hướng phát triển 93**](#_heading=h.37m2jsg)

[**TÀI LIỆU THAM KHẢO 95**](#_heading=h.1mrcu09)

# LỜI MỞ ĐẦU

Trong bối cảnh phát triển mạnh mẽ của khoa học và công nghệ hiện nay, Công Nghệ Thông Tin (CNTT) đang có những bước tiến vượt bậc, làm thay đổi cách con người sống và làm việc. Một trong những xu hướng quan trọng là việc áp dụng CNTT vào nhiều lĩnh vực khác nhau nhằm hướng tới xây dựng một hành tinh thông minh. Sự kết hợp giữa CNTT và Internet đã tạo nên một mạng lưới kết nối toàn cầu, hỗ trợ đắc lực cho nhiều ngành nghề như giao thông, quân sự, y học, và đặc biệt là trong công tác quản lý, bao gồm quản lý sách.

Qua khảo sát một số quán sách và thư viện tại Hà Nội, nhóm chúng tôi nhận thấy sự cần thiết của việc ứng dụng CNTT vào quản lý sách. Từ đó, nhóm quyết định chọn đề tài “Xây dựng phần mềm quản lý sách” cho bài tập lớn môn Lập trình Java. Đề tài này không chỉ giúp tối ưu hóa việc quản lý sách mà còn đáp ứng nhu cầu tìm kiếm và mượn sách của người dùng một cách nhanh chóng và hiệu quả.

Nhờ sự hướng dẫn tận tình của thầy Nguyễn Thái Cường, nhóm chúng tôi đã nghiên cứu và áp dụng các kiến thức đã học để xây dựng chương trình quản lý sách. Trong quá trình khảo sát, nhóm đã tìm hiểu kỹ về nhu cầu và khó khăn trong việc quản lý sách tại các quán sách và thư viện. Dựa trên những yêu cầu thu thập được, nhóm tiến hành phân tích và thiết kế hệ thống với các chức năng quản lý sách, quản lý người mượn, tra cứu sách, và quản lý mượn trả sách. Sau đó, nhóm sử dụng ngôn ngữ lập trình Java để phát triển các chức năng đã thiết kế, đồng thời tiến hành kiểm thử để đảm bảo phần mềm hoạt động hiệu quả.

Do thời gian và kiến thức còn hạn chế, phần mềm của nhóm vẫn còn một số thiếu sót. Nhóm rất mong nhận được ý kiến đóng góp của thầy cô và các bạn để hoàn thiện phần mềm hơn.

Nhóm hy vọng rằng phần mềm quản lý thư viện này sẽ là công cụ hữu ích cho việc quản lý sách và phục vụ tốt hơn nhu cầu của người dùng. Chúng em rất mong nhận được phản hồi để tiếp tục cải tiến và phát triển phần mềm.

# PHẦN 1: GIỚI THIỆU

## Tên đề tài

**Xây dựng phần mềm quản lý sách**

## Lý do chọn đề tài

Ngành Công nghệ thông tin (CNTT) đang trải qua một sự phát triển nhanh chóng và có ảnh hưởng sâu sắc đến nhiều lĩnh vực. Điều này đã làm cho việc thực hiện các tác vụ trở nên dễ dàng, thuận tiện và nhanh chóng hơn bao giờ hết. CNTT đã đơn giản hóa các công việc điều này giúp tiết kiệm một lượng lớn thời gian và công sức của con người, tăng độ chính xác và hiệu quả trong công việc.

Nhận thấy được, quản lý sách là một yêu cầu cấp thiết trong các tổ chức như thư viện, trường học, doanh nghiệp hoặc các trung tâm cho thuê/mượn sách khác Nhóm 9 quyết định chọn đề tài: “***Xây dựng phần mềm quản lý sách***.” Một phần mềm có thể tận dụng những lợi ích của CNTT để tối ưu hóa quá trình quản lý sách. Phần mềm quản lý sách giúp tạo ra sự thuận lợi và nhanh chóng trong việc tìm kiếm, mượn và trả sách.  Nó giúp tổ chức và cập nhật thông tin sách một cách hiệu quả, nhanh chóng, chính xác.

## Xác định nội dung học tập và các kỹ năng, kiến thức then chốt

**Nội dung học tập:**

* Cấu trúc của một chương trình java
* Các kiểu dữ liệu và chuyển kiểu dữ liệu
* Các toán tử
* Các cấu trúc điều khiển
* Mảng và xử lý mảng
* Lớp và đối tượng trong java
* Các hàm khởi tạo
* Phương thức tĩnh static
* Mảng đối tượng
* Nạp chồng phương thức
* Kết tập, kế thừa trong java
* Tính trừu tượng, đa hình và interface
* Ghi đè phương thức
* Xử lý ngoại lệ, I/O theo luồng và thao tác với tệp
* Cấu trúc Collection
* Giao diện Java Swing

**Kiến thức trang bị:**

* Lập trình java cơ sở
* Lập trình java hướng đối tượng
* Xử lý ngoại lệ và thao tác với tệp
* Lập trình java với cấu trúc Collection
* Lập trình giao diện Java

**Kĩ năng then chốt:**

* Kĩ năng làm việc nhóm
* Kĩ năng phân tích và xử lý tình huống
* Kĩ năng thu thập và chuẩn hóa thông tin
* Kĩ năng xây dựng ý tưởng đề tài

## Công cụ thực hiện đề tài

* Cài đặt và triển khai phần mềm dựa trên công cụ viết mã nguồn như:
* Rational Rose:
* Rational Rose là một công cụ lập mô hình trực quan mạnh trợ giúp bạn phân tích và thiết kế các hệ thống phần mềm hướng đối tượng. Nó được dùng để lập mô hình hệ thống trước khi bạn viết mã (code). Dùng mô hình, bạn có thể bắt kịp những thiếu sót về thiết kế, trong khi việc chỉnh sửa chúng vẫn chưa tốn kém.
* Dùng để phân tích thiết kế hệ thống.
* Eclipse:
* Eclipse là một môi trường phát triển tích hợp (IDE) mã nguồn mở, chủ yếu được sử dụng để phát triển ứng dụng Java.
* Nó cung cấp nhiều tính năng hữu ích cho lập trình như trình soạn thảo mã, trình gỡ lỗi, hỗ trợ refactoring, và khả năng mở rộng thông qua các plugins.
* Eclipse được thiết kế để tăng tính năng suất và hiệu quả của quá trình phát triển phần mềm. Nó hỗ trợ các quy trình phát triển như Agile và DevOps.
* Intell:
* Intell là một công cụ hỗ trợ lập trình và phát triển phần mềm. Nó cung cấp nhiều tính năng như gợi ý mã, tự động hoàn thiện mã, quản lý dự án, và tích hợp với các công cụ phát triển khác.
* Intell được thiết kế để tăng tính năng suất và tính hiệu quả của quá trình lập trình bằng cách cung cấp các tính năng thông minh và tự động hóa các nhiệm vụ lặp đi lặp lại.
* Intell hỗ trợ nhiều ngôn ngữ lập trình phổ biến như Java, C++, Python, JavaScript, và hơn thế nữa. Nó có thể được tích hợp và sử dụng trong các IDE như Eclipse, IntelliJ IDEA, và Visual Studio Code.
* JDK 17:
* JDK 17 là một bản tham chiếu của Java 17, được ra mắt vào ngày 15 tháng 9. JDK 17 trong đó là bản phát hành hỗ trợ được Oracle hỗ trợ trong sáu tháng. Oracle triển khai JDK 17 như một điểm khởi đầu để chuyển sang JDK 17, với người dùng có thể thử nghiệm trên JDK 17 trước khi triển khai lên JDK 19. Các bản phát hành LTS được xuất bản ba năm một lần.
* JDK 17 (Java Development Kit 17) là một bản phát hành của Java Development Kit, bao gồm môi trường phát triển và các công cụ cần thiết để phát triển ứng dụng Java. Phiên bản này thường đi kèm với một số cải tiến, sửa lỗi và tính năng mới so với các phiên bản JDK trước đó, cũng như các bản cập nhật cho ngôn ngữ Java. JDK 17 có thể được sử dụng để phát triển và triển khai các ứng dụng Java.

# PHẦN 2: KẾT QUẢ NGHIÊN CỨU

## Giới thiệu

Trong bối cảnh số hóa và quản lý thông tin ngày càng quan trọng, nhóm chúng tôi quyết định lựa chọn đề tài "Xây dựng phần mềm quản lý sách" nhằm tạo ra một công cụ hữu ích cho việc quản lý và tra cứu sách một cách hiệu quả. Sản phẩm của chúng tôi sẽ là một ứng dụng phần mềm chạy được, cung cấp các chức năng quản lý sách như thêm, xóa, sửa, tìm kiếm sách, quản lý người dùng và thống kê.

Để phát triển phần mềm này, chúng tôi sẽ áp dụng quy trình Agile Scrum, một phương pháp phát triển phần mềm hiện đại giúp tăng tính linh hoạt và khả năng phản hồi nhanh chóng đối với các yêu cầu và thay đổi mới. Quy trình bao gồm các giai đoạn chính: khởi tạo, lập kế hoạch và thiết kế, phát triển, kiểm thử tích hợp và triển khai, cùng bảo trì và hỗ trợ. Mỗi giai đoạn sẽ có các bước cụ thể nhằm đảm bảo chất lượng và tiến độ của dự án.

Nhóm sẽ sử dụng các công cụ hỗ trợ viết mã nguồn như NetBeans, GitHub, Rational Rose, JDK 17. Các công cụ này không chỉ hỗ trợ mạnh mẽ cho việc phát triển mà còn giúp tối ưu hóa quy trình làm việc nhóm, đảm bảo tính toàn vẹn và khả năng mở rộng của phần mềm.

Với mục tiêu tạo ra một sản phẩm hoàn thiện và tiện dụng, chúng tôi hy vọng phần mềm quản lý sách này sẽ đáp ứng tốt nhu cầu của người dùng, đồng thời giúp cải thiện hiệu quả quản lý và tra cứu thông tin sách.

## Khảo sát hệ thống

### Khảo sát sơ bộ

Mục tiêu chính của xây dựng phần mềm quản lý sách là nhằm tối ưu hóa các quy trình liên quan đến sách, từ lưu trữ thông tin sách đến theo dõi quá trình mượn/trả.

Về phạm vi, hệ thống cần đáp ứng các yêu cầu cơ bản như quản lý thông tin sách, theo dõi tồn kho, ghi nhận các giao dịch mượn/trả sách, quản lý thông tin độc giả. Ngoài ra, hệ thống cũng cung cấp các tính năng nâng cao như tìm kiếm/tra cứu sách, tạo các báo cáo thống kê. Qua đó, phần mềm quản lý sách sẽ giúp các tổ chức quản lý tài nguyên hiệu quả hơn, đáp ứng nhu cầu ngày càng cao của độc giả.

### Tài liệu đặc tả yêu cầu người dùng

#### Yêu cầu chức năng

* *Đối với người dùng:*
* Đăng ký: Cho phép người dùng tạo tài khoản mới trong hệ thống bằng cách cung cấp thông tin cá nhân cần thiết.
* Đăng nhập: Cho phép người dùng đăng nhập vào tài khoản đã đăng ký trước đó bằng cách cung cấp thông tin đăng nhập như tên tài khoản và mật khẩu để thực hiện các chức năng tương ứng với quyền của mình.
* Xem thông tin sách: cho phép bản đọc xem thông tin chi tiết về các cuốn sách trong hệ thống.
* Tìm kiếm sách:  Cho phép người dùng tìm kiếm sách môn mong muốn một cách nhanh chóng.
* Chính sửa thông tin cá nhân: Cung cấp cho người dùng khả năng chỉnh sửa thông tin cá nhân trong tài khoản.
* Đăng xuất: Cho phép người dùng đăng xuất khỏi tài khoản hiện tại, kết thúc phiên làm việc và trở về trạng thái người dùng chưa đăng nhập.
* *Đối* với người quản lý/thủ thư:
* Đăng nhập: Hệ thống cho phép người quản lý đăng nhập để thực hiện các chức năng tương ứng với quyền của mình.
* Quản lý tài khoản: Quản lý và cập nhật thông tin tài khoản các thủ thư, khoá và mở khoá tài khoản người dùng.
* Quản lý sách: Lưu trữ và cập nhật thông tin sách.
* Quản lý mượn / trả sách: Lưu trữ, xử lý các yêu cầu mượn/ trả sách của người dùng.
* Tra cứu và tìm kiếm: Cho phép tài khoản của thủ thư có thể tìm kiếm và truy cập vào các thông tin chi tiết về sách hay người mượn sách một cách dễ dàng và hiệu quả.
* Quản lý vi phạm: Kiểm tra và xử lý các vi phạm của người dùng khi cần thiết.

#### Yêu cầu phi chức năng

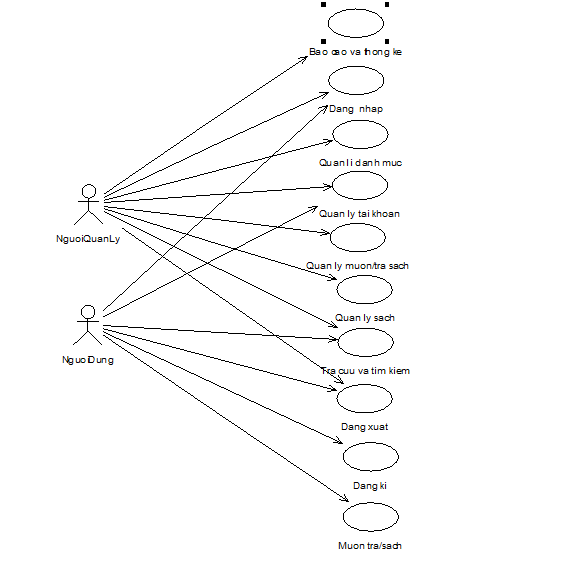
* Hệ thống xử lý nhanh gọn, chính xác và thuận tiện.
* Giao diện hài hòa, thân thiện và dễ sử dụng ( có thể thêm chức năng cần thiết sau này).
* Quy trình phát triển phần mềm phù hợp để dễ dàng bảo trì và nâng cấp.
* Có sự ràng buộc chặt chẽ về chất lượng, môi trường và chuẩn sử dụng.
* Đảm bảo về mặt thời gian, bản quyền.
* Chi phí chấp nhận được.
* Ngôn ngữ lập trình: Java - Swing.
* Lưu trữ cơ sở dữ liệu: File Text.
* Hệ điều hành: Windows

## Phân tích hệ thống

### ***2.3.1. Mô*** hình ***hóa*** chức ***năng hệ thống***

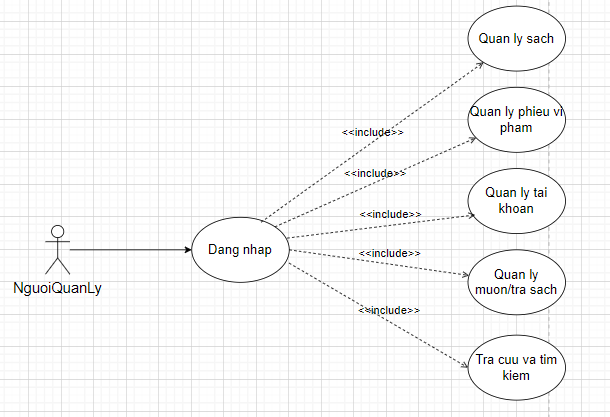
#### Biểu đồ use case tổng quát

* Các tác nhân của hệ thống: Theo phần đặc tả yêu cầu người dùng, có thể xác định các actor như sau:
* Người quản lý/ thủ thư.
* Người dùng.
* Các use case chính: Dựa theo việc xác định actor ở trên, ta xác định được các use case tương ứng:
* Đăng nhập  (người quản lý, độc giả)
* Đăng ký (độc giả)
* Mượn/trả sách (độc giả)
* Quản lý sách (người quản lý)
* Đổi mật khẩu (người quản lý, độc giả)
* Quản lý tài khoản (người quản lý, độc giả)
* Quản lý mượn/trả sách (người quản lý)
* Quản lý phiếu vi phạm (người quản lý)
* Tra cứu và tìm kiếm (người quản lý, độc giả)



*Hình2.1. Biểu đồ use case tổng quát*

#### Biểu đồ use case thứ cấp



*Hình 2.2. Biểu đồ phân rã use case*

#### Mô tả chi tiết các use case

1. **Use case Đăng nhập**

| Tên use case | Đăng nhập |
| --- | --- |
| Tác nhân | Người quản lý/thủ thư |
| Mô tả | Use case này cho phép người quản lý/thủ thư đăng nhập tài khoản vào hệ thống. Để thực hiện các chức năng quản lý. |
| Sự kiện kích hoạt chức năng | Use case bắt đầu khi người quản lý kích vào nút “Đăng nhập (thủ thư)” trên giao diện chính. Hệ thống hiển thị lên màn hình một giao diện đăng nhập. |
| Tiền điều kiện | Thông tin người quản lý/thủ thư đã có trong bảng TAI\_KHOAN. |
| Luồng sự kiện chính | | # | Thực hiện bởi | Hành động | | --- | --- | --- | | 1 | Người quản lý, hệ thống | * 1. Người quản lý/thủ thư nhập đầy đủ các thông tin như tên tài khoản, mật khẩu vào các trường tương ứng.   2. Người quản lý/thủ thư kích vào nút “Đăng nhập”. Hệ thống kiểm tra chi tiết người dùng trong bảng TAI\_KHOAN và hiển thị trang chủ đầy đủ các chức năng tương ứng với quyền của người quản lý. | | 2 | Hệ thống | Use case kết thúc. | |
| Luồng sự kiện thay thế | | # | Thực hiện bởi | Hành động | | --- | --- | --- | | 1 | Người quản lý, hệ thống | Tại mục 1a trong luồng sự kiện chính nếu người quản lý nhập không đâỳ đủ thông tin đã nêu trên thì hệ thống sẽ hiển thị lên màn hình thông báo lỗi “Vui lòng nhập đầy đủ thông tin”. Use case kết thúc. | | 2 | Người quản lý, hệ thống | Tại mục 1a trong luồng sự kiện chính, nếu người dùng nhập sai tên tài khoản hay mật khẩu thì hệ thống sẽ thông báo lỗi. Người dùng có thể nhập lại và tiếp tục bước 2 hoặc chọn các chức năng khác để kết thúc use case.. | | 3 | Người quản lý, hệ thống | Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc. | |
| Hậu điều kiện | Không có |
| Điểm mở rộng | Không có |

1. **Quản lý sách**

| Tên  use case | Quản lý sách |
| --- | --- |
| Tác nhân | Người quản lý |
| Mô tả | Use case này cho phép quản lý thực hiện các chức năng thêm, sửa, xóa sách. |
| Sự kiện kích hoạt chức năng | Use case này bắt đầu khi người quản lý chọn các chức năng thêm sách sách, sửa sách, xóa sách tại mục”Sách” trên menu tại giao diện trang chủ của người quản lý. |
| Tiền điều kiện | Cần phải đăng nhập thành công |
| Luồng sự kiện chính | | # | Thực hiện bởi | Hành động | | --- | --- | --- | | 1 | Người quản lý, hệ thống | Người quản lý chọn “Thêm sách”.   * 1. Hệ thống sẽ hiển thị giao diện cho phép người quản lý nhập các thông tin: mã sách, tên sách, nhà xuất bản, ngày phát hành, thế loại, số lượng, giá.   2. Người quản lý nhập đầy đủ thông tin đã nêu trên.   3. Người quản lý nhấn vào nút “Thêm sách”. Hệ thống sẽ hiển thị thông báo “Thêm sách thành công” lên màn hình. Use case kết thúc. | | 2 | Người quản lý, hệ thống | Người quản lý chọn “Xóa sách”.   * 1. Hệ thống sẽ hiển thị các thông tin gồm mã sách, tên sách, nhà xuất bản, ngày phát hành, thế loại, số lượng, giá.   2. Người quản trị nhấn vào một dòng sách cần xóa. Hệ thống sẽ hiển thị thông báo “Bạn có chắc chắn muốn xóa” lên màn hình.   3. Người quản trị kích vào nút “Yes”. Hệ thống sẽ hiển thị thông báo “Xóa thành công” lên màn hình. Use case kết thúc. | | 3 | Người quản lý, hệ thống | Người quản lý chọn “Sửa sách”.   * 1. Hệ thống sẽ hiển thị các thông tin gồm mã sách, tên sách, nhà xuất bản, ngày phát hành, thế loại, số lượng, giá.   2. Người quản lý chọn một dòng sách. Hệ thống sẽ hiển thị lên màn hình giao diện sửa gồm các thông tin  mã sách, tên sách, nhà xuất bản, ngày phát hành, thế loại, số lượng, giá.   3. Người quản lý sẽ nhập những thông tin cần chỉnh sửa và kích vào nút “Lưu”. Hệ thống sẽ hiển thị lên màn hình thông báo “Sửa sách thành công”. Use case kết thúc. | |
| Luồng sự kiện thay thế | | # | Thực hiện bởi | Hành động | | --- | --- | --- | | 1 | Người quản lý, hệ thống | Tại mục 1b, 3c trong luồng sự kiện chính nếu người quản lý nhập không đủ thông tin thì hệ thống sẽ hiển thị thông báo lỗi lên màn hình. Use case kết thúc. | | 2 | Người quản lý, hệ thống | Tại mục 1c trong luồng sự kiện chính, nếu người quản lý kích vào nút “Hủy” thì toàn bộ thông tin vừa nhập sẽ bị xóa. Use case kết thúc | | 3 | Người quản lý, hệ thống | Sau khi hoàn thành bước 1c trong luồng sự kiện chính nếu kích vào nút “Quay lại” hệ thống sẽ quay lại giao diện chính. Use case kết thúc. | | 4 | Người quản lý, hệ thống | Trước khi thực hiện mục 2b, 3b tại luồng sự kiện chính người quản lý có thể  nhập tên sách vào ô tìm kiếm và kích vào nút “Tìm kiếm”.Hệ thống sẽ hiển thị thông tin sách liên quan đến từ khóa. Người dùng tiếp tục thực hiện từ bước 2b, 3b trong luồng sự kiện chính. Use case kết thúc. | | 5 | Người quản lý, hệ thống | Tại mục 2c tại luồng sự kiện chính nếu người quản lý kích vào nút “No” hoặc “Cancel”. Hệ thống sẽ không ghi nhận hành động xóa thông tin sách vừa chọn. Use case kết thúc. | | 6 | Người quản lý, hệ thống | Tại bước 3c trong luống sự kiện chính nếu người quản lý nhấp vào nút “Hủy” thì hệ thống sẽ quay lại thông tin ban đầu của sách.. Use case kết thúc. | | 7 | Người quản lý, hệ thống | Tại bước 3a, 3b trong luồng sự kiện chính khi kích vào nút “Quay lại” hệ thống sẽ quay lại mục 3a. Use case kết thúc. | | 8 | Người quản lý, hệ thống | Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc. | |
| Hậu điều kiện | Thông tin được cập nhật cơ sở dữ liệu |
| Điểm mở rộng | Không có |

1. **Đổi mật khẩu (người quản lý)**

| Tên use case | Đổi mật khẩu |
| --- | --- |
| Tác nhân | Người quản lý |
| Mô tả | Use case cho phép quản lý thay đổi mật khẩu. |
| Sự kiện kích hoạt chức năng | Use case này bắt đầu khi người quản lý kích chọn chức năng “Đổi mật khẩu”  trên menu tại giao diện trang chủ của người quản. |
| Tiền điều kiện | Phải đăng nhập thành công |
| Luồng sự kiện chính | | # | Thực hiện bởi | Hành động | | --- | --- | --- | | 1 | Người quản lý, hệ thống | Hệ thống sẽ hiển thị giao diện đổi mật khẩu gốm nhập mật khẩu cũ, nhập mật khẩu mới, xác nhận mật khẩu. Người quản lý nhập đầy đủ các thông tin được nêu trên | | 2 | Người quản lý, hệ thống | Người quản lý kích vào nút “Đổi mật khẩu”. Hệ thống hiển thị thông báo “Đổi mật khẩu thành công”. | | 3 | Hệ thống | Use case kết thúc. | |
| Luồng sự kiện thay thế | | # | Thực hiện bởi | Hành động | | --- | --- | --- | | 1 | Người quản lý, hệ thống | Tại bước 1, 2 trong luồng sự kiện chính nếu người dùng nhấn vào nút “Quay lại”. Hệ thống sẽ hiển thị giao diện trang chủ chính. Use case kết thúc. | | 2 | Người quản lý, hệ thống | Tại bước 1 trong luồng sự kiện chính nếu người quản lý không nhập đầy đủ thông tin hệ thống hiển thị lên màn hình thông báo lỗi “Vui lòng nhập đầy đủ thông tin”. Use case kết thúc.. | | 3 |  | Tại bước 1 trong luồng sự kiện chính nếu người quản lý nhập tại mục “Nhập mật khẩu cũ” không trùng khớp với dữ liệu được lưu trước đó thì hệ thống hiển thị thông báo lỗi “Mật khẩu không chính xác” lên màn hình. Use case kết thúc. | | 4 |  | Tại bước 1 trong luồng sự kiện chính nếu người quản lý nhập tại mục “Nhập mật khẩu mới” và “Xác nhận mật khẩu” dữ liệu không giống nhau hệ thống sẽ hiển thị thông báo “Mật khẩu không trùng khớp” lên màn hình. | | 5 | Người quản lý, hệ thống | Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc. | |
| Hậu điều kiện | Thông tin mật khẩu mới được cập nhật vào cơ sở dữ liệu. |
| Điểm mở rộng | Không có |

1. **Quản lý tài khoản (người quản lý)**

| Tên use case | Quản lý tài khoản |
| --- | --- |
| Tác nhân | Người quản lý |
| Mô tả | Use case này cho phép người quản lý khóa tài khoản người dùng nếu phát hiện có vi phạm các quy định hoặc chính sách của hệ thống. |
| Sự kiện kích hoạt chức năng | Use case bắt đầu khi  người quản lý chọn các chức năng khóa tài khoản tại mục”Người dùng” trên menu giao diện trang chủ của người quản lý. |
| Tiền điều kiện | Cần phải đăng nhập thành công |
| Luồng sự kiện chính | | # | Thực hiện bởi | Hành động | | --- | --- | --- | | 1 | Người quản lý, hệ thống. | Hệ thống hiển thị thông tin người đọc bao gồm mã tài khoản, tên người dùng, địa chỉ, số điện thoại, tên đăng nhập. mật khẩu, trạng thái, số lần vi phạm. Người quản lý kích vào dòng người dùng muốn khóa tài khoản. Hệ thống sẽ hiển thị lên thông báo “Bạn có chắc chắn muốn khóa tài khoản của người dùng” | | 2 | Người quản lý, hệ thống | Người quản lý kích vào nút “Yes”. Hệ thống sẽ hiển thị lên thông báo “Đã khóa tài khoản người dùng này”. | | 3 | Hệ thống | Use case kết thúc. | |
| Luồng sự kiện thay thế | | # | Thực hiện bởi | Hành động | | --- | --- | --- | | 1 | Người quản lý, hệ thống | Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu người quản lý kích vào nút quay lại.Hệ thống sẽ không công nhận các hành động trước đó và quay lại màn hình trang chủ chính. Use case kết thúc. | | 2 | Người quản lý, hệ thống | Tại bước 2 của luồng sự kiện chính nếu người quản lý kích vào nút “No” hoặc “Cancel”. Hệ thống sẽ không ghi nhận thao tác khóa tài khoản. Use case kết thúc. | | 3 | Hệ thống | Tại bước 1 trong luồng sự kiện chính người quản lý nhập tên đăng nhập vào khung tìm kiếm. Người quản lý kích vào nút tìm kiếm hệ thống sẽ hiển thị những người dùng có tên đăng nhập bắt đầu bằng từ khóa. Use case kết thúc. | |
| Hậu điều kiện | Dữ liệu được cập nhật vào cơ sở dữ liệu. |
| Điểm mở rộng | Không có |

1. **Quản lý mượn/trả sách**

| Tên use case | Quản lý mượn/trả sách |
| --- | --- |
| Tác nhân | Người quản lý |
| Mô tả | Use case cho phép quản lý lưu trữ, cập nhật, sửa, xóa thông tin mượn và trả sách của người dùng. |
| Sự kiện kích hoạt chức năng | Use case này bắt đầu khi thủ thư click vào “Yêu cầu mượn/trả sách” thuộc người dùng. Hệ thống sẽ hiển thị lên màn hình các phiếu mượn chưa được phê duyệt và các phiếu mượn chưa trả. |
| Tiền điều kiện | Cần phải đăng nhập thành công |
| Luồng sự kiện chính | | # | Thực hiện bởi | Hành động | | --- | --- | --- | | 1 | Người quản lý, hệ thống | a. Người quản lý kích vào nút “Chấp thuận” ở phiếu mượn chưa được phê duyệt hệ thống sẽ hiện lên màn hình thông báo “Xác nhận đồng ý”.  b. Người quản lý kích vào nút “Đồng ý” hệ thống sẽ  hiện lên màn hình thông báo “Phê duyệt thành công vui lòng tải lại trang”. | | 2 | Người quản lý, hệ thống | a. Người quản lý chọn hành động trả sách ở phiếu mượn chưa được trả hệ thống sẽ hiện lên màn hình thông báo “Xác nhận đồng ý”.  b. Người quản lý kích vào nút “Đồng ý” hệ thống sẽ  hiện lên màn hình thông báo “Trả sách thành công vui lòng tải lại trang”. | | 3 | Hệ thống | Use case kết thúc. | |
| Luồng sự kiện thay thế | | # | Thực hiện bởi | Hành động | | --- | --- | --- | | 1 | Người quản lý, hệ thống | Tại bước 1a trong luồng sự kiện chính khi người quản lý kích vào nút “Quay lại” hệ thống sẽ hiển thị lên màn hình trang chủ chính của hệ thống. Use case kết thúc. | | 2 | Người quản lý, hệ thống | Tại bước 1a trong luồng sự kiện chính khi người quản lý kích vào nút “Không chấp thuận” hệ thống sẽ hiện thông báo “Xác nhận không đồng ý” lên màn hình. Người dùng kích vào nút “Đồng ý” hệ thống sẽ hiển thị lên màn hình thông báo” Không phê duyệt thành công”. Use case kết thúc. | | 3 | Người quản lý, hệ thống | Tại bước 2a trong luồng sự kiện chính khi người quản lý kích vào nút “Mất sách” hệ thống sẽ hiện thông báo “Xác nhận mất sách” lên màn hình. Người dùng kích vào nút “Yes” hệ thống sẽ hiển thị lên màn hình thông báo” Cập nhật thành công”. Use case kết thúc. | | 3 | Người quản lý, hệ thống. | Người quản lý kích vào nút Reload hệ thống sẽ cập nhật lại và hiển thị danh sách phiếu mượn chưa được phê duyệt. | | 4 | Hệ thống | Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc. | |
| Hậu điều kiện | . Dữ liệu được cập nhật vào cơ sở dữ liệu |
| Điểm mở rộng | Không có |

1. **Tra cứu và tìm kiếm (thủ thư)**

| Tên use case | Tra cứu và tìm kiếm |
| --- | --- |
| Tác nhân | Người quản lý |
| Mô tả | Use case này cho phép người quản lý tìm kiếm thông tin về các cuốn sách hoặc user. |
| Sự kiện kích hoạt chức năng | Use case bắt đầu khi người quản lý nhập vào từ khóa vào khung tìm kiếm  trên trang chủ chính. Hệ thống hiển thị 2 lựa chọn “Người dùng” và “Sách”. |
| Tiền điều kiện | Cần phải đăng nhập thành công. |
| Luồng sự kiện chính | | # | Thực hiện bởi | Hành động | | --- | --- | --- | | 1 | Người quản lý, hệ thống | Người quản lý kích chọn 1 trong 2 lựa chọn trên và kích vào nút “Tìm kiếm” | | 2 | Hệ  thống | .Hệ thống hiển thị thông tin sách hoặc người dùng có tên đăng nhập hoặc tên sách bắt đầu bằng từ khóa tìm kiếm. | | 3 | Hệ thống | Use case kết thúc. | |
| Luồng sự kiện thay thế | | # | Thực hiện bởi | Hành động | | --- | --- | --- | | 1 | Người quản lý, hệ thống | Tại bước 2 trong luồng sự kiện chính nếu người quản lý kích vào nút “Quay lại”. Hệ thống hiển thị lên màn hình giao diện trang chủ chính của hệ thống. Use case kết thúc. | | 2 | Người quản lý, hệ thống | Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc. | |
| Hậu điều kiện | Không có |
| Điểm mở rộng | Không có |

1. **Quản lý phiếu vi phạm**

| Tên use case | Quản lý phiếu vi phạm |
| --- | --- |
| Tác nhân | Người quản lý |
| Mô tả | Use case này cho phép người quản lý/thủ thư xem thông tin phiếu vi phạm và tạo phiếu vi phạm. |
| Sự kiện kích hoạt chức năng | Use case bắt đầu khi người quản lý kích vào mục “Phiếu vi phạm” trên trang chủ người quản lý. Hệ thống hiển thị thông tin danh sách phiếu mượn vi phạm lên màn hình gồm mã phiếu mượn, ngày mượn, ngày trả, mã tài khoản, mã sách, trạng thái lên màn hình. |
| Tiền điều kiện | Cần phải đăng nhập thành công. |
| Luồng sự kiện chính | | # | Thực hiện bởi | Hành động | | --- | --- | --- | | 1 | Người quản lý, hệ thống. | Người quản lý kích vào phiếu mượn cần lập phiếu vi phạm hệ thống hiển thị form tạo phiếu vi phạm lên màn hình gồm mã phiếu mượn, mã tài khoản, số tiền phạt, số ngày vi phạm, lý do lên màn hình. | | 2 | Người quản lý, hệ thống. | Người quản lý kích vào nút “Tạo” hệ thống hiển thị thông báo “Tạo phiếu vi phạm thành công” lên màn hình. | | 3 | Hệ thống | Use case kết thúc. | |
| Luồng sự kiện thay thế | | # | Thực hiện bởi | Hành động | | --- | --- | --- | | 1 | Người quản lý, hệ thống | Tại bước 1  luồng sự kiện chính người quản lý kích vào nút “Quay lại” hệ thống sẽ quay lại trang chủ hệ thống. Use case kết thúc. | | 2 | Người quản lý, hệ thống | Tại bước 1, bước 2 trong  luồng sự kiện chính người quản lý kích vào nút “Danh sách phiếu vi phạm” hệ thống hiển thị lên màn hình danh sách các phiếu vi phạm. Use case kết thúc. | | 3 | Người quản lý, hệ thống | Tại bước 2 trong luồng sự kiện chính người quản quản lý kích vào nút “Hủy” hệ thống sẽ không xử lý hoạt động vừa thao tác. Use case kết thúc. | | 4 | Người quản lý, hệ thống | Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use | |
| Hậu điều kiện | Dữ liệu được cập nhật vào cơ sở dữ liệu |
| Điểm mở rộng | Không có |

1. **Đăng nhập (người dùng)**

| Tên use case | Đăng nhập |
| --- | --- |
| Tác nhân | Người dùng/độc giả |
| Mô tả | Use case này cho phép người dùng/độc giả đăng nhập tài khoản vào hệ thống. Để thực hiện các chức năng của người dùng.. |
| Sự kiện kích hoạt chức năng | Use case bắt đầu khi độc giả kích vào nút “Đăng nhập” trên giao diện chính. Hệ thống hiển thị lên màn hình một giao diện đăng nhập. |
| Tiền điều kiện | Thông tin người dùng/độc giả đã có trong bảng TAI\_KHOAN. |
| Luồng sự kiện chính | | # | Thực hiện bởi | Hành động | | --- | --- | --- | | 1 | Người dùng/độc giả, hệ thống | * 1. Người dùng/độc giả nhập đầy đủ các thông tin như tên tài khoản, mật khẩu vào các trường tương ứng.   2. Người dùng/độc giả kích vào nút “Đăng nhập”. Hệ thống kiểm tra chi tiết người dùng trong bảng TAI\_KHOAN và hiển thị trang chủ đầy đủ các chức năng tương ứng với quyền của người dùng. | | 2 | Hệ thống | Use case kết thúc | |
| Luồng sự kiện thay thế | | # | Thực hiện bởi | Hành động | | --- | --- | --- | | 1 | Người dùng, hệ thống | Tại mục 1a trong luồng sự kiện chính nếu người dùng nhập không đâỳ đủ thông tin đã nêu trên thì hệ thống sẽ hiển thị lên màn hình thông báo lỗi “Vui lòng nhập đầy đủ thông tin”. Use case kết thúc. | | 2 | Người dùng, hệ thống | Tại mục 1a trong luồng sự kiện chính, nếu người dùng nhập sai tên tài khoản hay mật khẩu thì hệ thống sẽ thông báo lỗi. Người dùng có thể nhập lại và tiếp tục bước 2 hoặc chọn các chức năng khác để kết thúc use case. | | 3 | Người dùng, hệ thống | Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc. | |
| Hậu điều kiện | Không có |
| Điểm mở rộng | Không có |

1. **Đăng ký**

| Tên use case | Đăng ký |
| --- | --- |
| Tác nhân | Người dùng |
| Mô tả | Use case này cho phép người dùng đăng ký tài khoản để thực hiện các chức năng của người dùng |
| Sự kiện kích hoạt chức năng | Use case bắt đầu khi người dùng kích vào nút “Đăng ký” trên giao diện chính. Hệ thống hiển thị lên màn hình một giao diện đăng ký. |
| Tiền điều kiện | Không có |
| Luồng sự kiện chính | | # | Thực hiện bởi | Hành động | | --- | --- | --- | | 1 | Người dùng/độc giả, hệ thống | * 1. Người dùng/độc giả nhập đầy đủ thông tin như tên người dùng, địa chỉ, số điện thoại, tên đăng nhập, mật khẩu vào các trường tương ứng   2. Người dùng kích vào nút “Đăng ký”nếu các thông tin đều đúng yêu cầu thì hệ thống sẽ hiện thông báo đăng ký tài khoản thành công   3. Người dùng click vào nút “Quay lại” hệ thống sẽ trở về giao diện ban đầu. | | 2 | Hệ thống | Use case kết thúc | |
| Luồng sự kiện thay thế | | # | Thực hiện bởi | Hành động | | --- | --- | --- | | 1 | Người dùng, hệ thống | Tại mục 1a trong luồng sự kiện chính nếu người dùng nhập không đâỳ đủ thông tin đã nêu trên thì hệ thống sẽ hiển thị lên màn hình thông báo lỗi “Vui lòng nhập đầy đủ thông tin”. Use case kết thúc. | | 2 | Người dùng, hệ thống | Tại mục 1a nếu người dùng nhập sai yêu cầu mật khẩu “mật khẩu phải có ít nhất 5 ký tự gồm 1 chữ in hoa, 1 chữ số” thì yêu cầu này sẽ hiện màu đỏ và k thể đăng ký được.Use  case kết thúc | | 3 | Người dùng, hệ thống | Tại mục 1a nếu người dùng nhập thông tin ở trường “xác nhận mật khẩu” không trùng khớp với thông tin ban đầu ở trường”mật khẩu” thì khi click vào nút “đăng ký” hệ thống sẽ hiện thông báo “mật khẩu không trùng khớp”.Use case kết thúc. | | 4 | Người dùng, hệ thống | Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc. | |
| Hậu điều kiện | Thông tin người dùng được thêm vào bảng TAI\_KHOAN |
| Điểm mở rộng | Không có |

1. **Đăng xuất (người dùng)**

| Tên use case | Đăng xuất |
| --- | --- |
| Tác nhân | Người dùng |
| Mô tả | Use case này cho phép người dùng thoát ra khỏi tài khoản đang đăng nhập. |
| Sự kiện kích hoạt chức năng | Use case này bắt đầu khi người dùng click vào nút “đăng xuất” |
| Tiền điều kiện | Đăng nhập thành công |
| Luồng sự kiện chính | | **#** | Thực hiện bởi | Hành động | | --- | --- | --- | | 1 | Người dùng, hệ thống | Người dùng click vào nút “đăng xuấ” hệ thống sẽ trở về giao diện đăng nhập ban đâu | | 2 | Hệ thống | Use case kết thúc | |
| Luồng sự kiện thay thế | | **#** | Thực hiện bởi | Hành động | | --- | --- | --- | | 1 | Người dùng, hệ thống | Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc. | |
| Hậu điều kiện | Không có |
| Điểm mở rộng | Không có |

1. **Sửa đổi thông tin tài khoản(Người dùng)**

| Tên use case | Sửa đổi thông tin tài khoản |
| --- | --- |
| Tác nhân | Người dùng/độc giả |
| Mô tả | Use case này cho phép người dùng thay đổi và cập nhật lại thông tin cá nhân. |
| Sự kiện kích hoạt chức năng | Người dùng kích vào nút “Chỉnh sửa thông tin cá nhân” trong mục thông tin cá nhân. Hệ thống hiển thị lên màn hình thông tin cá nhân gồm tên tài khoản, tên đầy đủ, địa chỉ, số điện thoại và các nút có chức năng “Lưu”, “Đổi mật khẩu”, “Quay lại”. |
| Tiền điều kiện | Đăng nhập thành công |
| Luồng sự kiện chính | | # | Thực hiện bởi | Người dùng | | --- | --- | --- | | 1 | Người dùng, hệ thống | * 1. Khi người dùng muốn thay đổi thông tin về tên tài khoản, tên đầy đủ, địa chỉ, số điện thoại. Người dùng xóa thông tin cũ và nhập thông tin mới cần thay đổi.   2. Người dùng nhấn nút “Lưu”. Hệ thống sẽ hiển thị lên màn hình thông báo “Thông tin đã được cập nhật”. Người dùng nhấn vào nút “OK” hệ thống sẽ quay lại trang chủ chính bên phía người dùng. | | 2 | Người dùng, hệ thống | * + - 1. Khi người dùng muốn thay đổi mật khẩu. Người dùng kích vào nút “Đổi mật khẩu” . Hệ thống sẽ hiển thị lên giao diện đổi mật khẩu gồm mật khẩu hiện tại, mật khẩu mới, xác nhận mật khẩu. Người dùng nhập đầy đủ thông tin, đúng định dạng.       2. Người dùng kích vào nút “Lưu”. Hệ thống hiển thị lên màn hình thông báo “Mật khẩu đã được cập nhật”.  Người dùng nhấn vào nút “OK” hệ thống sẽ quay lại trang chủ chính bên phía người dùng. | | 3 | Hệ thống | Use case kết thúc | |
| Luồng sự kiện thay thế | | # | Thực hiện bởi | Hành động | | --- | --- | --- | | 1 | Người dùng, hệ thống | Tại mục 1a, 2a của luồng sự kiện chính nếu người dùng kích vào nút “Quay lại” thì hệ thống sẽ quay lại trang chủ chính bên phía người dùng. Use case kết thúc. | |  | Người dùng, hệ thống | Tại bước  1b của luồng sự kiện chính nếu người dùng nhập thiếu thông tin hệ thống sẽ hiển thị lên màn hình thông báo “Vui lòng nhập đầy đủ thông tin”. Use case kết thúc. | | 2 | Người dùng, hệ thống | Tại mục  2b của luồng sự kiện chính nếu người dùng không nhập đầy đủ thông tin thì hệ thống sẽ hiển thị lên màn hình thông báo “Mật khẩu mới không đáp ứng yêu cầu”. Use case kết thúc. | | 3 | Người dùng, hệ thống | Tại mục 2b nếu người dùng nhập mật khẩu mới và xác nhận mật khẩu không khớp thì hệ thống sẽ hiển thị lên màn hình thông báo lỗi. Use case kết thúc. | | 4 | Người dùng, hệ thống | Tại mục 2b nếu người dùng nhập mật khẩu hiện tại không chính xác thì hệ thống sẽ hiển thị lên màn hình thông báo lỗi. Use case kết thúc. | | 5 | Người dùng, hệ thống | Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use | |
| Hậu điều kiện | Thông tin mới tài khoản được cập nhật. |
| Điểm mở rộng | Không có |

1. **Tra cứu và tìm kiếm (người dùng)**

| Tên use case | Tra cứu và tìm kiếm |
| --- | --- |
| Tác nhân | Người dùng |
| Mô tả | Use case cho phép người người dùng tra cứu và tìm kiếm thông tin sách bằng tên hoặc mã sách |
| Sự kiện kích hoạt chức năng | Use case này bắt đầu khi người dùng click vào thanh “tra cứu và tìm kiếm” ở trên màn hình trang chủ.Lúc này người dùng bắt buộc phải điền thông tin cần tìm kiếm. |
| Tiền điều kiện | Cần phải đăng nhập thành công |
| Luồng sự kiện chính | | # | Thực hiện bởi | Hành động | | --- | --- | --- | | 1 | Người dùng, hệ thống | * 1. Sau khi điền thông tin cần tìm kiếm vào thanh tìm kiếm trên trang chủ   2. Người dùng click vào nút “Tìm kiếm”. Hệ thống kiểm tra thông tin cần tìm kiếm trong mục Quan\_Ly\_Sach. Hệ thống sẽ hiển thị thông tin sách mà người quản lý vừa tìm kiếm | | 2 | Hệ thống | Use case kết thúc | |
| Luồng sự kiện thay thế | | **#** | Thực hiện bởi | Hành động | | --- | --- | --- | | 1 | Người dùng, hệ thống | Tại mục 1a) nếu dùng không điền thông tin vào thanh tìm kiếm, hệ thống sẽ hiện thông báo “mời nhập thông tin cần tìm kiếm” | | 2 | Người dùng, hệ thống | Nếu thông tin tìm kiếm không có trong bảng Quan\_Ly\_Sach thì màn hình tìm kiếm sẽ không hiển thị thông tin của bất kỳ cuốn sách nào | | 3 | Người dùng, hệ thống | Nếu người dùng click vào nút “Quay lại” thì màn hình sẽ quay trở lại màn hình trang chủ | | 4 | Người dùng,hệ thống | Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use | |
| Hậu điều kiện | Không có |
| Điểm mở rộng | Không có |

1. **Mượn sách**

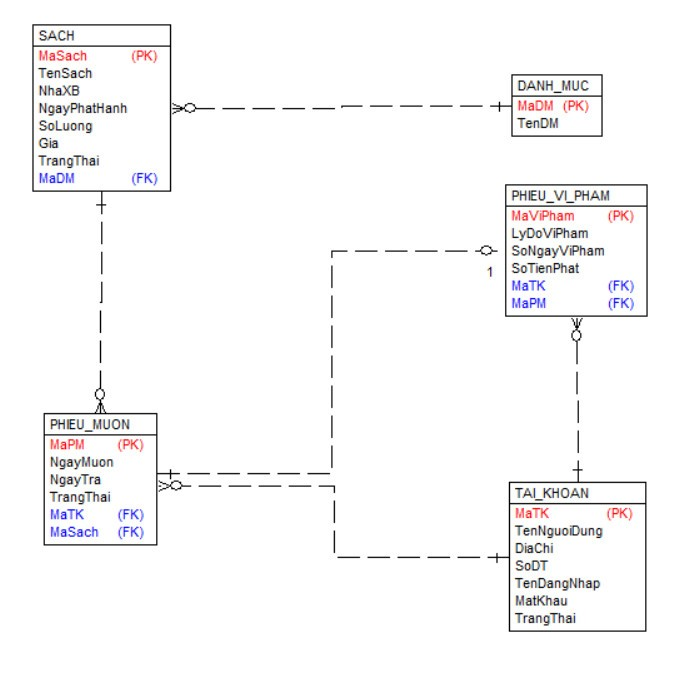
| Tên use case | Mượn trả sách |
| --- | --- |
| Tác nhân | Người dùng |
| Mô tả | Use case này cho phép người dùng mượn sách qua trang phần mềm trực tuyến. |
| Sự kiện kích hoạt chức năng | Use case bắt đầu khi người dùng kích vào nút “Xem” tại dòng sách muốn mượn trong mục “Tìm kiếm sách”. Hệ thống sẽ hiển thị lên màn hình thông tin chi tiết sách gồm mã sách, tên sách, nhà xuất bản, ngày phát hành, thể loại, số lượng, giá từ bảng SACH và các chức năng “Đặt mượn sách”, “Quay lại”. |
| Tiền điều kiện | Người dùng phải đăng nhập thành công |
| Luồng sự kiện chính | | # | Thực hiện bởi | Hành động | | --- | --- | --- | | 1 | Người dùng, hệ thống | Người dùng kích vào nút “Đặt mượn sách”. Hệ thống sẽ hiển thị lên màn hình thông tin tóm tắt sách gồm mã sách, tên sách, thể loại từ bảng SACH. | | 2 | Người dùng, hệ thống | Người dùng kích vào nút “Xác nhận”  hệ thống hiển thị thông báo “Chờ xác nhận của thủ thư” lên màn hình. | | 3 | Người dùng, hệ thống | Người dùng nhấn vào nút “OK” hệ thống sẽ gửi yêu cầu đến thủ thư. | | 4 | Hệ thống | Use case kết thúc. | |
| Luồng sự kiện thay thế | | # | Thực hiện bởi | Hành động | | --- | --- | --- | | 1 | Người dùng, hệ thống | Tại bước 1 trong luồng sự kiện chính nếu người dùng kích vào nút “Quay lại” hệ thống sẽ thoát khỏi form chi tiết sách. Use case kết thúc. | | 2 | Người dùng, hệ thống | Tại bước 2 trong luồng sự kiện chính nếu người dùng kích vào nút “Quay lại” hệ thống sẽ thoát khỏi form thông tin chi tiết sách. Use case kết thúc. | | 3 | Hệ thống | Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc. | |
| Hậu điều kiện | Thông tin mượn sách được lưu vào bảng PHIEU\_MUON |
| Điểm mở rộng | Không có |

### 2.3.2. Mô hình hóa dữ liệu và cài đặt hệ thống

#### Các yêu cầu dữ liệu

* Danh mục gồm mã danh mục, tên danh mục. Mỗi danh mục có thể chứa 0, 1 hoặc nhiều đầu sách. Mỗi danh mục có thể ở trong 0,1 hoặc nhiều đầu sách.
* Sách gồm mã sách, tên sách, nhà xuất bản, ngày phát hành, số lượng, giá, trạng thái. Mỗi phiếu mượn áp dụng cho một và chỉ một đầu sách. Mỗi đầu sách chỉ thuộc về một và chỉ một danh mục.
* Phiếu mượn gồm mã phiếu mượn, ngày mượn, ngày trả, trạng thái. Mỗi phiếu mượn có thể mượn 1 và chỉ 1 đầu sách. Mỗi phiếu mượn chỉ có thể có 0 hoặc 1 phiếu vi phạm. Mỗi phiếu vi phạm chỉ áp dụng cho một và chỉ một phiếu mượn.
* Phiếu vi phạm gồm mã vi phạm, lý do vi phạm, số ngày vi phạm. số tiền phạt. Mỗi tài khoản có thể có 0, 1 hoặc nhiều phiếu vi phạm. Một phiếu vi phạm thuộc về một và chỉ một tài khoản.
* Tài khoản gồm có mã tài khoản, tên người dùng, địa chỉ, số điện thoại, tên đăng nhập, mật khẩu. Một tài khoản có 0, 1 hoặc nhiều phiếu mượn. Một phiếu mượn thuộc về một và chỉ một tài khoản.

#### Biểu đồ thực thể liên kế mức logic



*Hình 2.3. Biểu đồ thực thể liên kết*

#### Sơ đồ lớp dữ liệu

**Bảng sách:**

| Thuộc tính | Kiểu dữ liệu | Mô tả |
| --- | --- | --- |
| MaSach | Nvarchar | Mã sách |
| TenSach | Nvarchar | Tên sách |
| NhaXB | Nvarchar | Nhà xuất bản |
| NgayPhatHang | Date time | Ngày phát hành |
| SoLuong | Integer | Số lượng |
| Gia | Double | Giá |
| TrangThai | Bit | Trạng Thái |
| MaDM | Nvarchar | Mã danh mục |

**Bảng danh mục:**

| Thuộc tính | Kiểu dữ liệu | Mô tả |
| --- | --- | --- |
| MaDM | Nvarchar | Mã danh mục |
| TenDM | Nvarchar | Tên danh mục |

**Bảng phiếu mượn:**

| Thuộc tính | Kiểu dữ liệu | Mô tả |
| --- | --- | --- |
| MaPM | Nvarchar | Mã phiếu mượn |
| NgayMuon | Date time | Ngày mượn |
| NgayTra | Date time | Ngày trả |
| TrangThai | Nvarchar | Trạng thái |
| MaTK | Nvarchar | Mã tài khoản |
| MaSach | Nvarchar | Mã sách |

**Bảng tài khoản:**

| Thuộc tính | Kiểu dữ liệu | Mô tả |
| --- | --- | --- |
| MaTK | Nvarchar | Mã tài khoản |
| TenNguoiDung | Nvarchar | Tên người dùng |
| DiaChi | Nvarchar | Địa chỉ |
| SoDT | Char | Số điện thoại |
| TenDangNhap | Nvarchar | Tên đăng nhập |
| MatKhau | Nvarchar | Mật khẩu |
| TrangThai | Bit | Trạng thái |

**Bảng phiếu vi phạm:**

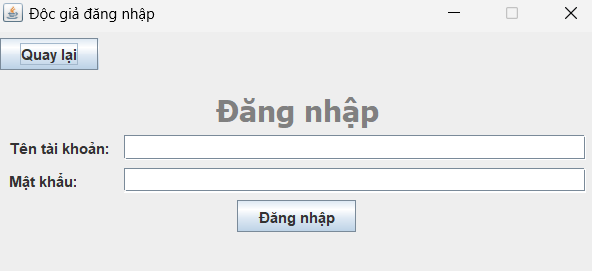
| Thuộc tính | Kiểu dữ liệu | Mô tả |
| --- | --- | --- |
| MaViPham | Nvarchar | Mã vi phạm |
| LyDoViPham | Nvarchar | Lý do vi phạm |
| SoNgayViPham | Integer | Số ngày vi phạm |
| SoTienPhat | Double | Số tiền phạt |
| MaTK | Nvarchar | Mã tài khoản |
| MaPM | Nvarchar | Mã phiếu mượn |

### Giao diện hệ thống



*Hình 2.4. Đăng ký /Đăng nhập*

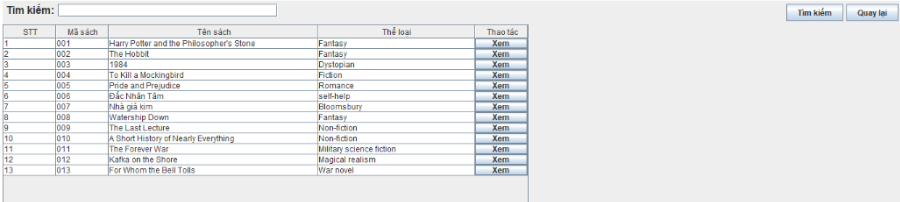
#### Đối với người dùng



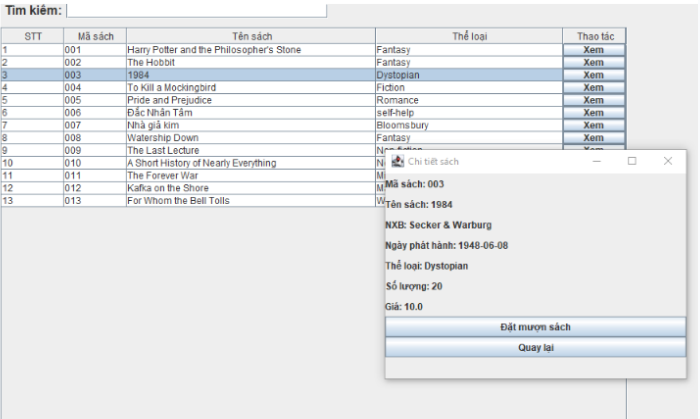
*Hình 2.5. Form đăng nhập (user)*



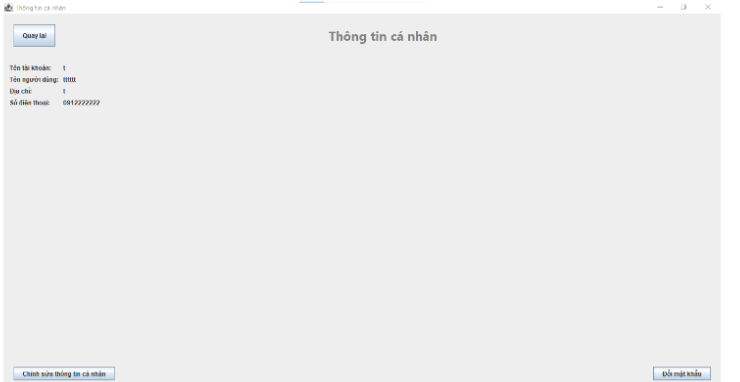
*Hình 2.6. Form đăng ký*



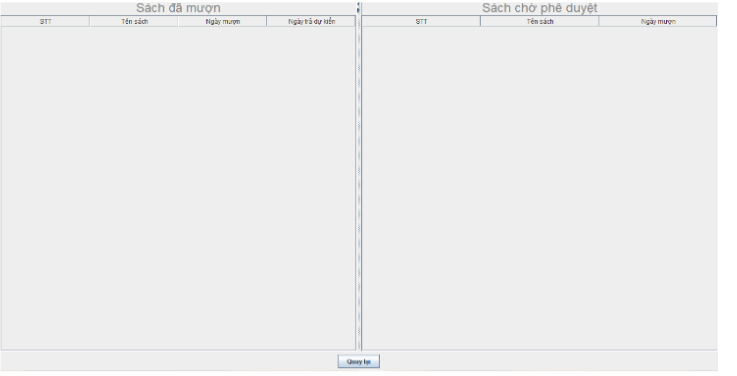
*Hình 2.7. Tìm kiếm sách*



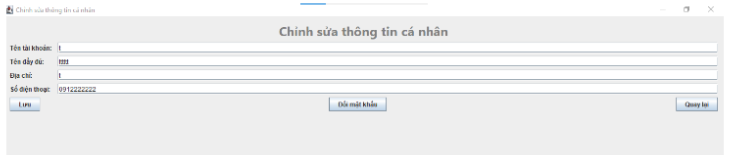
*Hình 2.8. Mượn sách*

**

*Hình 2.9. Thông tin cá nhân*

**

*Hình 2.10. Lịch sử mượn sách*

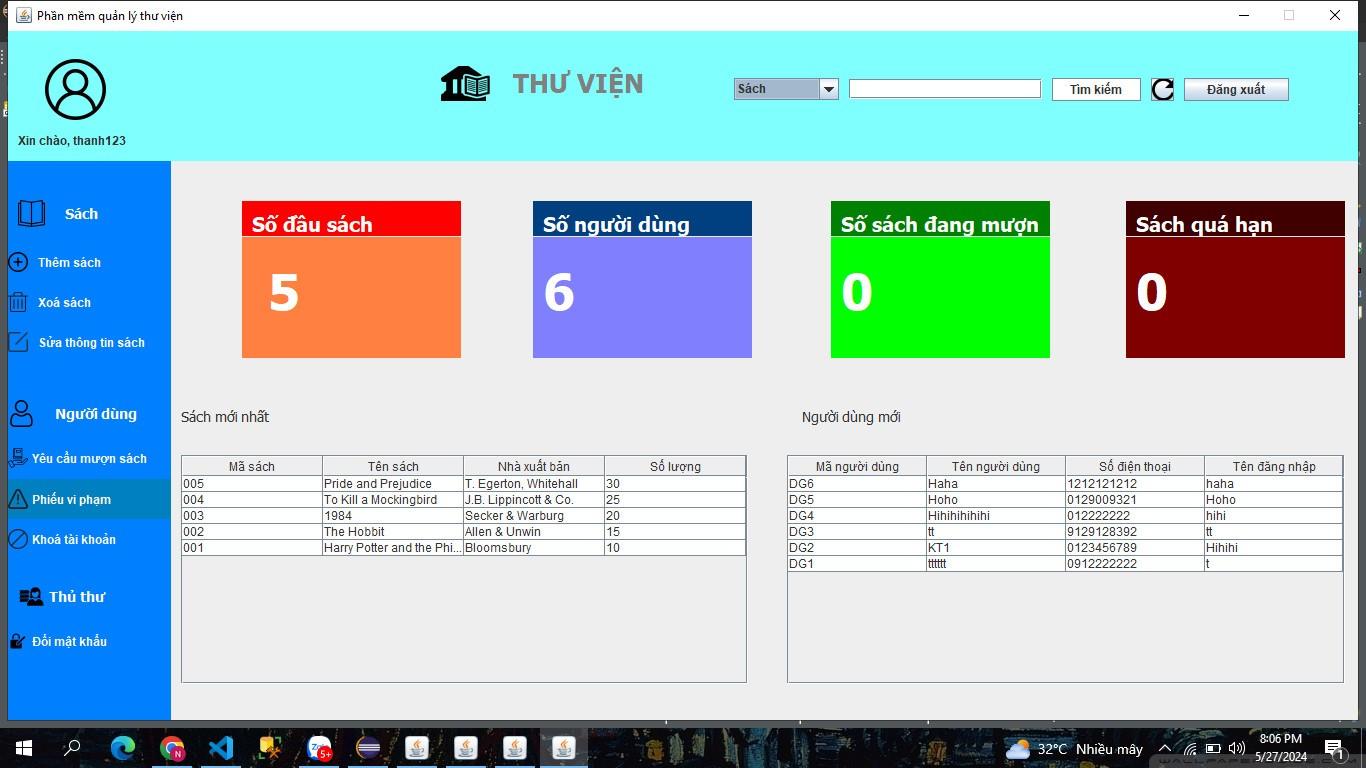
**

*Hình 2.11. Sửa tài khoản*

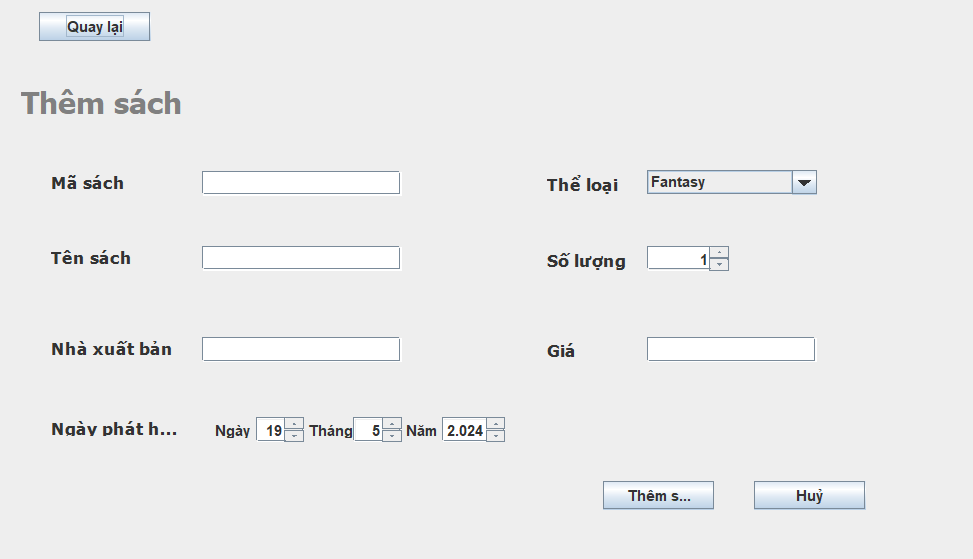
#### Đối với thủ thư



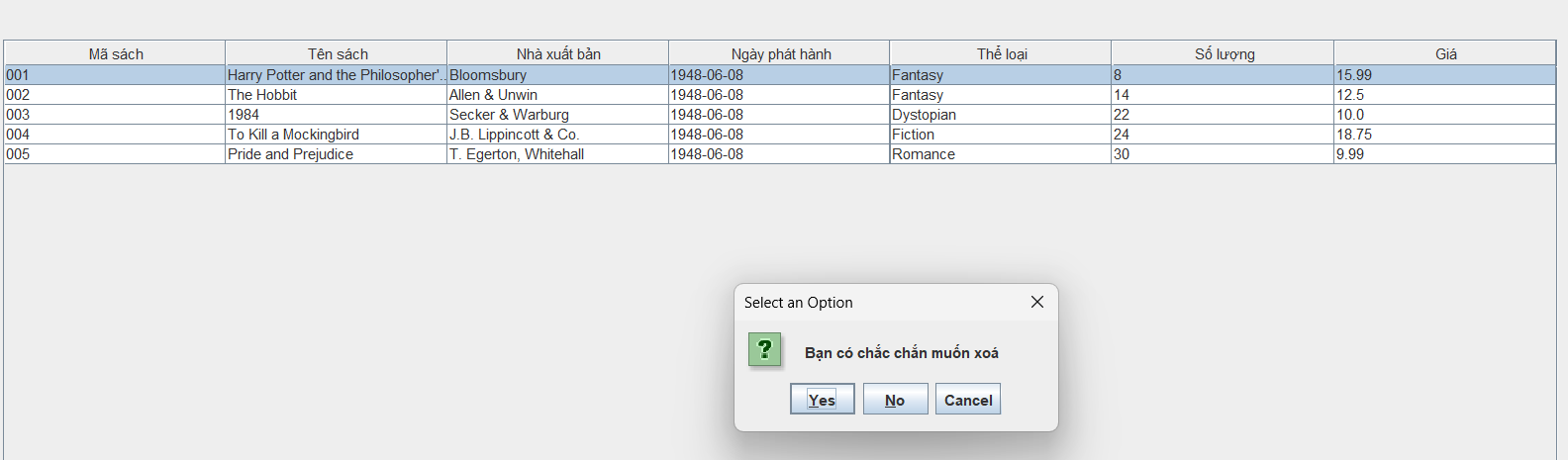
*Hình 2.12. Form đăng nhập (thủ thư)*



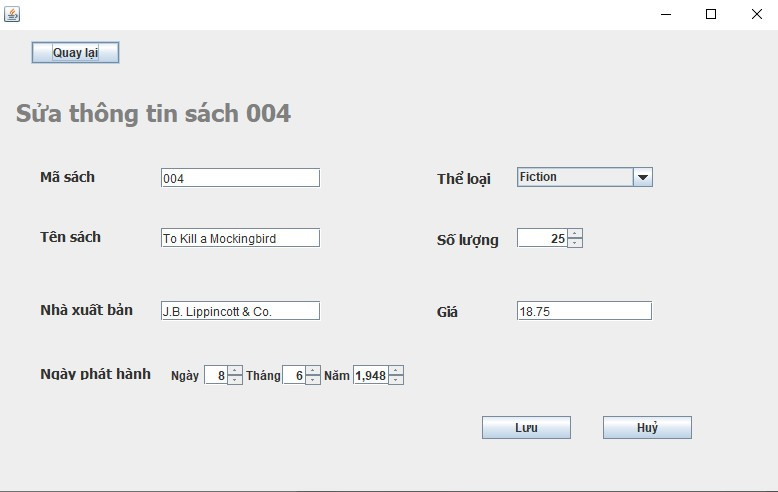
*Hình 2.13. Trang chủ (quản lý)*



*Hình 2.14. Thêm sách*



*Hình 2.15. Xóa sách*



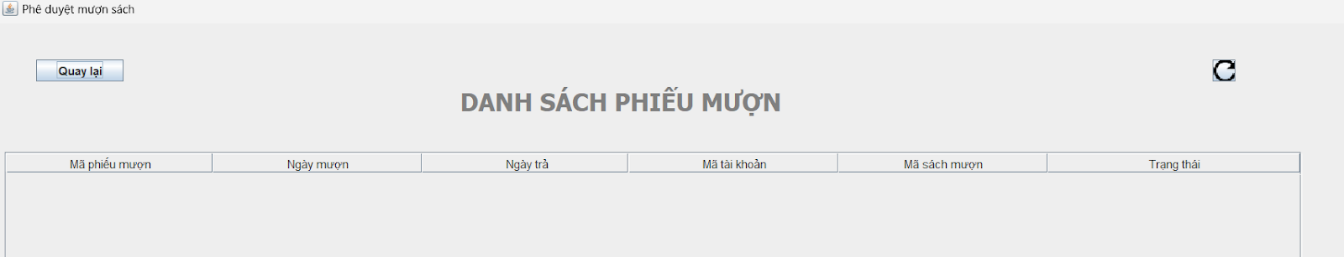
*Hình 2.16. Sửa thông tin sách*



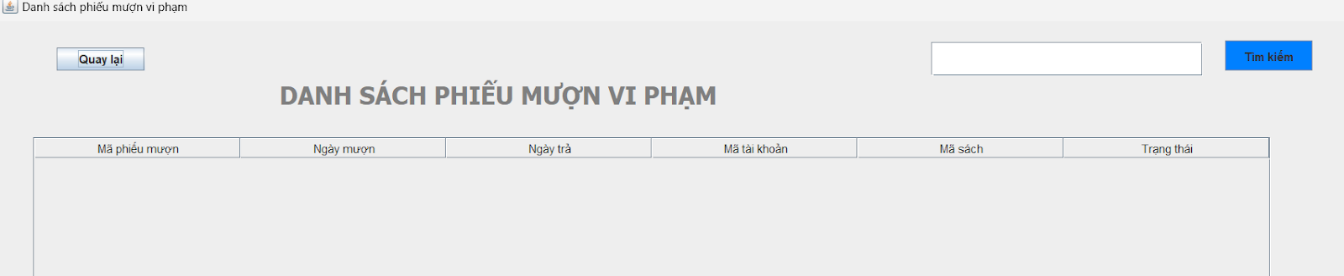
*Hình 2.17. Tìm kiếm sách*



*Hình 2.18. Tìm kiếm người dùng*



*Hình 2.19. Yêu cầu mượn sách*



*Hình 2.20. Phiếu vi phạm*



*Hình 2.21. Khóa tài khoản*



*Hình 2.22. Đổi mật khẩu*

## Thực hiện bài toán

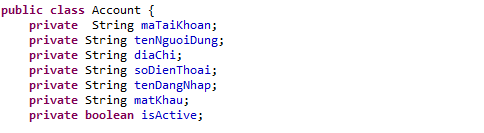
### Đối với người dùng

#### Đăng ký (Phạm Thị Quyên)



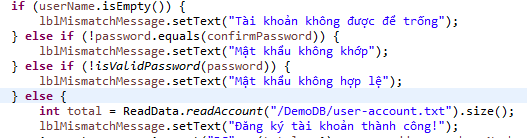
*Hình 2.23. Giao diện đăng ký*

* **Nội dung nhập/xuất:**Người dùng nhập đẩy đủ thông tin về tên người dùng,địa chỉ, số điện thoại, tên đăng nhập, mật khẩu và xác nhận mật khẩu. Mật khẩu của người dùng phải có ít nhất 5 ký tự gốm 1 chữ in hoa, 1 chữ số. Đăng ký thành công người dùng có thể dùng tên tài khoản và mật khẩu vừa đăng ký để đăng nhập.
* **Hướng đối tượng:** Định nghĩa lớp Account với các thuộc tính mã tài khoản, tên người dùng, địa chỉ, số điện thoại, tên đăng nhập, mật khẩu. Lớp này có các phương thức khởi tạo, các phương thức truy cập (getters) và thiết lập (setters) cho các thuộc tính, và một phương thức toString() được ghi đè để in ra thông tin của đối tượng Account.



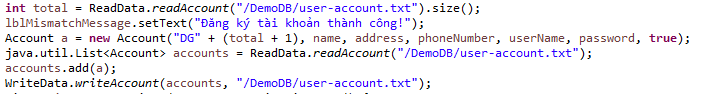
*Hình 2.24. Các thuộc tính của Account*

* **Bắt lỗi và gom rác:**
  + Đoạn code sẽ kiểm tra xem người dùng đã nhập đầy đủ các thông tin yêu cầu chưa. Nếu người dùng còn để trống hệ thống sẽ hiện thông báo “Tài khoản không được để trống” bằng lệnh IblMismatchMessage.setText
  + Đoạn code này cũng sẽ kiểm tra xem mật khẩu và xác nhận mật khẩu của người dùng có giống nhau không bằng lệnh !passwork.equals(confirmPasswork). Nếu nó khác nhau hệ thống sẽ hiện thông báo “Mật khẩu không khớp”.
  + Nếu người dùng nhập mật khẩu không đúng với yêu cầu hệ thống sẽ hiện thị thông báo “mật khẩu không hợp lệ” bằng mã lệnh !isValidPassword



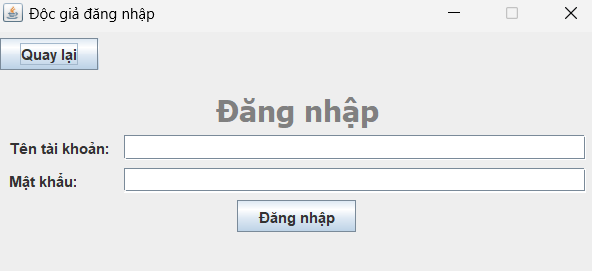
*Hình 2.25. Bắt lỗi khi thực hiện đăng ký*

* **Thao tác file:** Các hàm readAccout() và writeAccount() dùng để đọc và ghi danh sách Account.

****

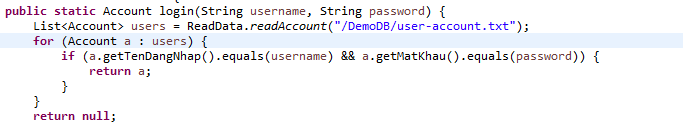
*Hình 2.26. Thao tác đọc ghi dữ liệu với file user-account.txt*

#### Đăng nhập (Phạm Thị Quyên)



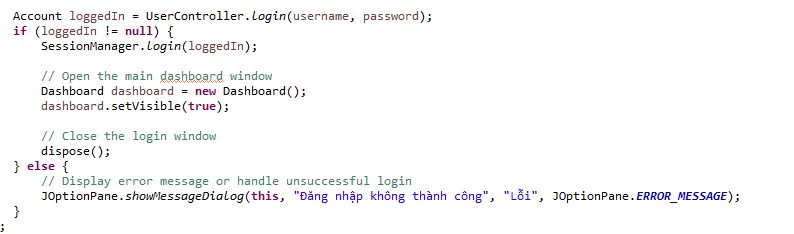
*Hình 2.27. Giao diện đăng nhập( Người dùng)*

* **Nội dung nhập/xuất:** Để đăng nhập người dùng nhập vào tên tài khoản và mật khẩu. Tên tài khoản và mật khẩu của người dùng là tên và mật khẩu mà người dùng đã đăng ký trước đó. Sau khi đăng nhập thành công sẽ hiển thị lên màn hình giao diện trang chủ chính.
* **Hướng đối tượng:** Lớp Account được sử dụng để thể hiện thông tin của người dùng. Phương thức Account login thể hiện việc đăng nhập của người dùng



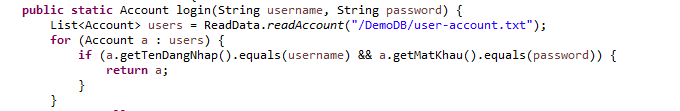
*Hình 2.28. Phương thức đăng nhập người dùng*

* **Bắt lỗi và gom rác:**
  + Nếu đăng nhập thành công một phiên làm việc mới được bắt đầu bằng cách gọi SessionManager.login(loggedIn) để lưu trữ thông tin về tài khoản đăng nhập. Sau đó, một cửa sổ bảng điều khiển chính (Dashboard) được tạo và hiển thị cho người dùng.
  + Nếu đăng nhập không thành công. Hệ thống sẽ hiện thị thông báo “Đăng nhập không thành công”, “Lỗi” cho người dùng thông qua JOptionPane.showMessageDialog(). Thông báo này có thể thông báo cho người dùng biết rằng thông tin đăng nhập không hợp lệ hoặc có lỗi xảy ra trong quá trình xác minh đăng nhập.

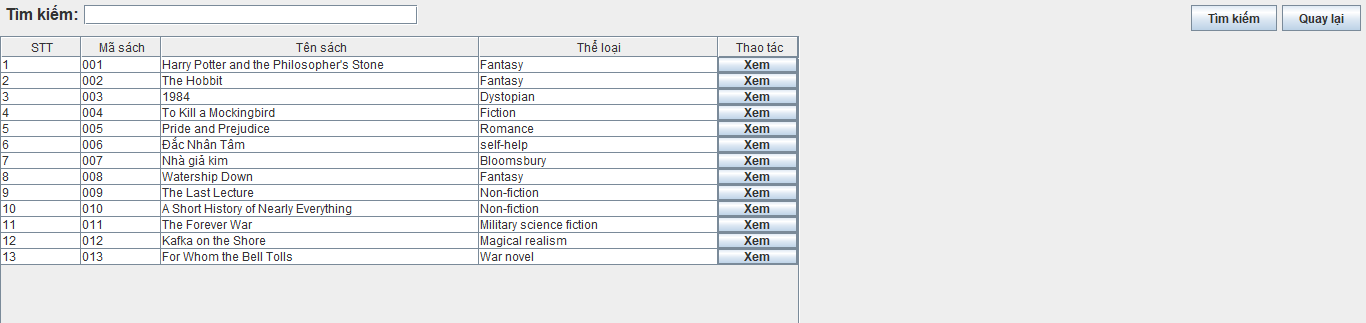


*Hình 2.29. Bắt lỗi khi thực hiện đăng nhập*

* **Thao tác file:** **Sử** dụng phương thức ReadData.readAccount() để đọc dữ liệu từ tệp /DemoDB/user-account.txt và sau đó kiểm tra thông tin đăng nhập trong danh sách các tài khoản từ tệp .

*Hình 2.30. Thao tác đọc/ghi file user-account.txt khi Đăng nhập*

#### Tìm kiếm (Phạm Thị Quyên)

**

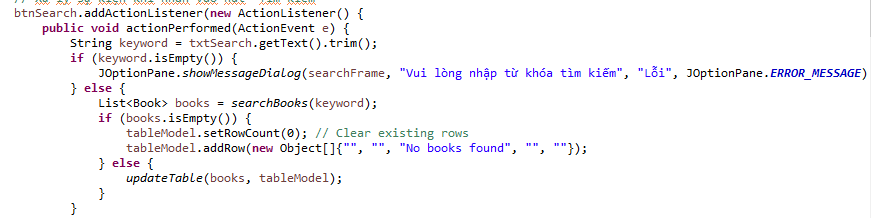
*Hình 2.31.Giao diện tìm kiếm ( Người dùng)*

* **Nội dung nhập/xuất:** Người dùng nhập từ khóa cần tìm kiếm vào thanh tìm kiếm. Sau khi nhấn click nút “Tìm kiếm” hệ thống sẽ hiện thị những thông tin liên quan về từ khóa lên màn hình.
* **Hướng đối tượng:** Lớp Book được sử dụng để biểu diễn thông tin về một cuốn sách. Phương thúc searchBooks dùng để thể hiện hành vi của việc tìm kiếm sách của người dùng.

**https://lh7-us.googleusercontent.com/docsz/AD_4nXf0nNnrI4wWsO6T76kz2-nUUrKzXxH3r93at_ck_ja6zH4nNQ1Sme-_HKJ7c4IxzK3jd2nnrc-qYA1cD4JR2iIyYJuGFPKgliGfDooqPQwQH1Bo0x2pbI0foXhMVdKGvr2Qb3FfWNTrFAHkdCXRlVFVI5eR?key=mNimE1Z9oMGM36aSsqhnkQ**

*Hình 2.32.  Gọi phương thức searchBooks để tìm kiếm toàn bộ sách*

* **Bắt lỗi và gom rác:**
  + Lấy từ khóa tìm kiếm từ trường văn bản txtSearch và loại bỏ các khoảng trắng ở đầu và cuối chuỗi bằng cách sử dụng phương thức trim().
  + Nếu người dùng không nhập gì, một hộp thoại thông báo lỗi sẽ xuất hiện để yêu cầu người dùng nhập từ khóa tìm kiếm.
  + Nếu người dùng đã nhập, hàm searchBooks(keyword) sẽ được gọi để tìm kiếm các cuốn sách phù hợp với từ khóa. Kết quả trả về là một danh sách các cuốn sách.
  + Nếu danh sách kết quả rỗng, bảng hiển thị kết quả table sẽ được xóa hết các hàng hiện có và thêm một hàng thông báo "No books found".
  + Nếu danh sách kết quả không rỗng, hàm updateTable(books, tableModel) được gọi để cập nhật bảng hiển thị với danh sách kết quả mới.



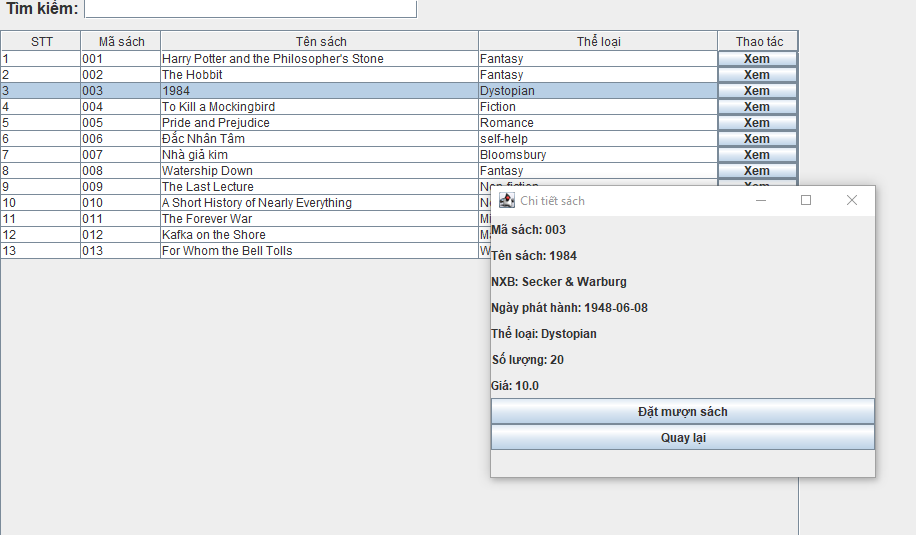
*2.33. Bắt lỗi với tìm kiếm sách*

* **Thao tác file:**
  + Đoạn mã này thực hiện thao tác đọc tệp văn bản để tìm kiếm thông tin về sách trong tệp dữ liệu**.**Sử dụng một đối tượng BufferedReader để đọc từng dòng trong tệp văn bản.

https://lh7-us.googleusercontent.com/docsz/AD_4nXfyTl6ZXCZ3eSbNHzIiOavSQmjMbdwidklaRwyFrEACLbYcZ70tyzdtP02mmMmGFm3ZHB17c7qIIIF-VjZNeT2n-S68bpR6gkPk0rl00hZ4cFAufXvpXSjguearYHYT1g0IyqGeaQT0hYEVS187T_7HHtgb?key=mNimE1Z9oMGM36aSsqhnkQ

*Hình 2.34. Đọc dữ liệu sách từ Book.txt*

#### Mượn sách (Bùi Thị Kiều Trang)

**

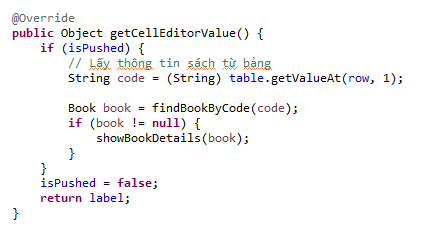
*Hình 2.35. Giao diện Mượn sách*

* **Nội dung nhập/xuất:** Sau khi thao tác “xem” cuốn sách cần đặt mượn, hệ thống sẽ hiển thị giao diện thông tin về giá thành và số lượng của cuốn sách để người dùng lựa chọn đặt mượn sách.
* **Hướng đối tượng:** Lớp Book được sử dụng để biểu diễn thông tin về một cuốn sách.

**https://lh7-us.googleusercontent.com/docsz/AD_4nXfGIR-_qxX4CWtQgVIIoBw2B0_9Es8batHqq0byMijIHYJrjQwxWSqpeT9XSCw23fobc_SstEP4V2DZxqDMlIpZeNaaIoUIHdTJxYatPItSuZA-yhi-n2L451tIzbYAIfWRyjxKZ3lexJDvC-hOX4ilzBTb?key=mNimE1Z9oMGM36aSsqhnkQ**

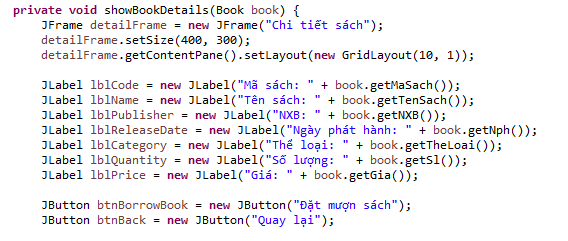
*Hình 2.36. Cập nhật kết quả tìm kiếm sách*

* **Bắt lỗi và gom rác:**
  + Trong phương thức getCellEditorValue, khi người dùng nhấn vào nút "Xem", mã sẽ lấy mã sách từ ô tương ứng trong JTable, sau đó tìm kiếm cuốn sách tương ứng trong danh sách sách. Nếu sách được tìm thấy, phương thức showBookDetails sẽ được gọi để hiển thị chi tiết của cuốn sách đó trong một cửa sổ riêng biệt.



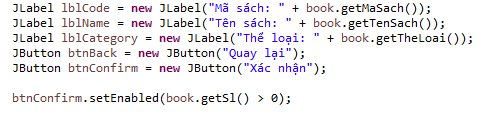
*Hình 2.37. Lấy thông tin sách*

* + Trong showBookDetails, một cửa sổ mới sẽ được tạo để hiển thị thông tin chi tiết của cuốn sách. Nó bao gồm các chi tiết như mã sách, tên sách, nhà xuất bản, ngày phát hành, thể loại, số lượng và giá. Cửa sổ này cũng bao gồm các nút như "Đặt mượn sách" và "Quay lại".



*Hình 2.38. Thông tin chi tiết cúa sách*

* + Khi người dùng nhấn vào nút "Đặt mượn sách", phương thức showDetailedBookInfo được gọi để hiển thị thông tin chi tiết về việc mượn sách, bao gồm nút "Xác nhận" để xác nhận việc mượn sách.

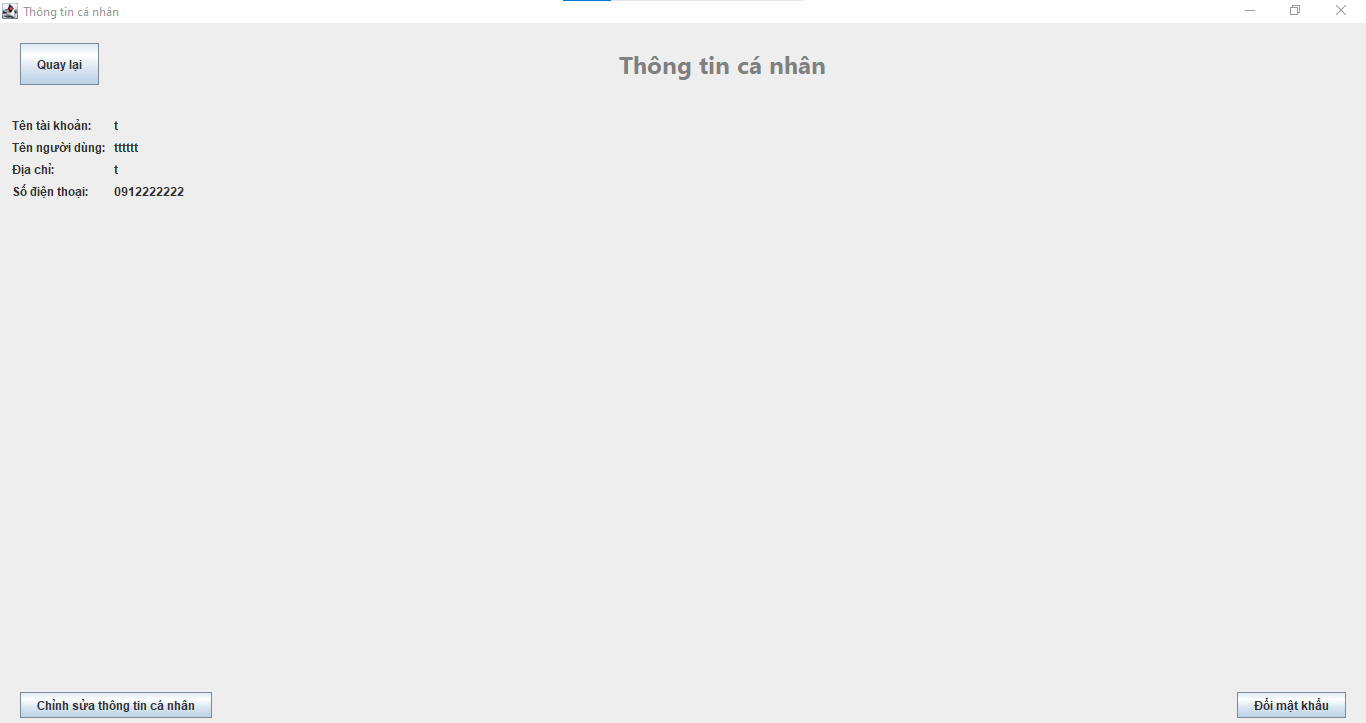


*Hình 2.39. Xác nhận mượn sách*

* **Thao tác file:** Đoạn code sử dụng phương thức ReadData.readBook để đọc danh sách các cuốn sách từ tệp tin "/DemoDB/Book.txt".**https://lh7-us.googleusercontent.com/docsz/AD_4nXfyTl6ZXCZ3eSbNHzIiOavSQmjMbdwidklaRwyFrEACLbYcZ70tyzdtP02mmMmGFm3ZHB17c7qIIIF-VjZNeT2n-S68bpR6gkPk0rl00hZ4cFAufXvpXSjguearYHYT1g0IyqGeaQT0hYEVS187T_7HHtgb?key=mNimE1Z9oMGM36aSsqhnkQ**

*Hình 2.40. Thao tác đọc/ghi với file Book.txt*

#### Quản lý tài khoản (Bùi Thị Kiều Trang)

**

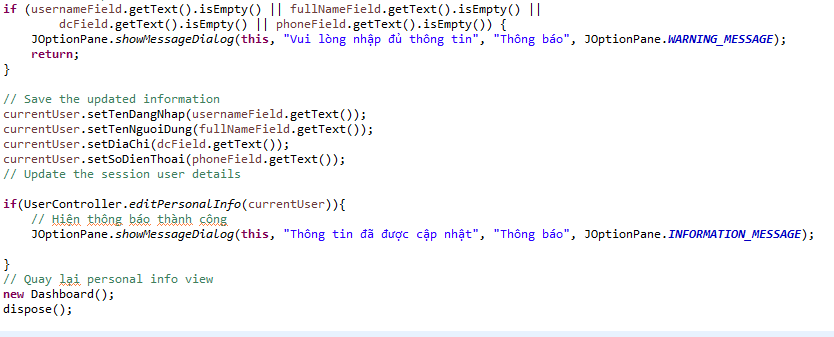
*Hình 2.41. Giao diện thông tin cá nhân*

* **Nội dung nhập/xuất:** Nhập thông tin cá nhân cần chỉnh sửa, mật khẩu cũ và mật khẩu cần thay đổi. Sau khi thay đổi click nút “Lưu” ngay lập tức thông tin sẽ được cập nhật vào hệ thống. Nếu click nút “Quay lại” thì hệ thông vẫn lưu những thông tin đã đc lưu trước đó.
* **Hướng đối tượng:** Lớp Account được sử dụng để biểu diễn thông tin của một người dùng. Phương thức EditPersonalInfo dùng để thể hiện hành vi của việc thay đổi thông tin cá nhân của người dùng.

https://lh7-us.googleusercontent.com/docsz/AD_4nXeiEKTWH7Yznd9BaAS-LQEWVWhyzY4df8AaN_rt6dIxpnlurDRqMQ8WBrXSLnsQ9E-kK6rXeM_SzfgQLGWbAynWTVIhAr-6FLWQsPmyaDU4l7DlpLzObT3KNRdII6bMHrq8IkZBJEhWhVIuSTnHBts_1wsI?key=mNimE1Z9oMGM36aSsqhnkQ

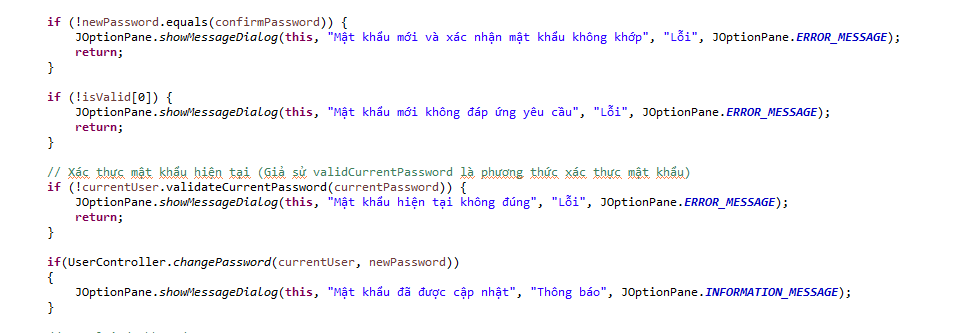
*Hình 2.42. Phương thức chỉnh sửa thông tin cá nhân*

* **Bắt lỗi và gom rác:**
  + Đầu tiên, mã kiểm tra xem có trường nào trong các trường nhập liệu (usernameField, fullNameField, dcField, phoneField) bị bỏ trống hay không.
  + Nếu có bất kỳ trường nào bị bỏ trống, mã sẽ hiển thị một hộp thoại cảnh báo (JOptionPane.showMessageDialog) với thông báo "Vui lòng nhập đủ thông tin" và dừng việc thực hiện bằng cách sử dụng return;.
  + Nếu tất cả các trường đều đã được nhập, mã sẽ tiến hành cập nhật đối tượng currentUser với các giá trị mới từ các trường nhập liệu:Tên đăng nhập, tên người dùng, địa chỉ, số điện thoại.

****

*Hình 2.43. Bắt lỗi khi sửa thông tin tài khoản*

* So sánh newPassword và confirmPassword. Nếu không khớp, hệ thống hiển thị thông báo lỗi “Mật khẩu mới và xác nhận mật khẩu không khớp” và dừng quá trình (return)**.**
* isValid[0] là một biến kiểm tra độ mạnh của mật khẩu. Nếu isValid[0] là false, mật khẩu mới không đáp ứng yêu cầu về độ mạnh.Hệ thống hiển thị thông báo lỗi “Mật khẩu không đáp ứng yêu cầu”  và dừng quá trình (return)
* ValidateCurrentPassword là một phương thức của currentUser để kiểm tra mật khẩu hiện tại. Nếu mật khẩu hiện tại không đúng, hiển thị thông báo lỗi và dừng quá trình (return).

****

*Hình 2.44. Bắt lỗi khi đổi mật khẩu*

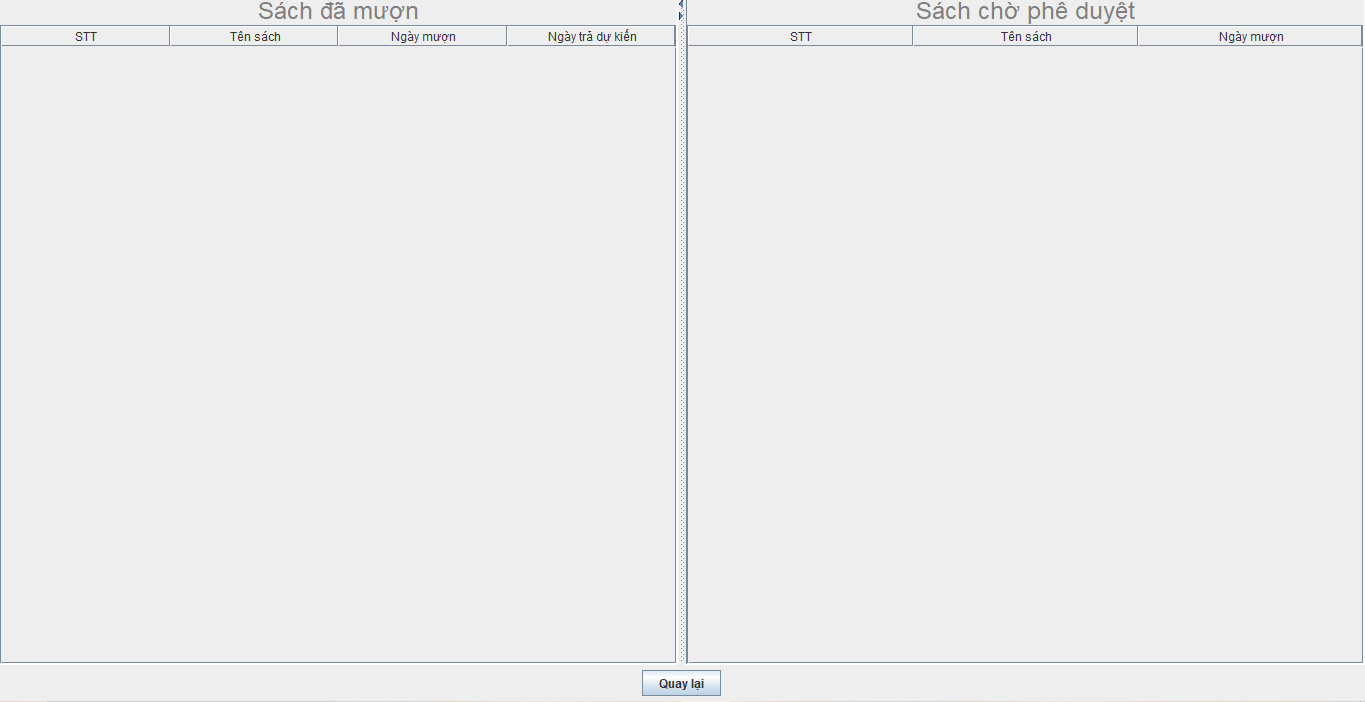
* **Thao tác file:** Đoạn code sử dụng phương thức ReadData.readAccount và WriteData.writeAccount để đọc/ghi danh sách các thông tin về người dùng "/DemoDB/user-account.txt".

**https://lh7-us.googleusercontent.com/docsz/AD_4nXcREN1Qu-4KtywL9jvfmBVpl-VVKIrjnKoPiKEI749Fn2HbuhA12uNw-iXZ1FEyjNPXNU_Eeo0BVr9y2p2INIGaptM4hqirF64q5CINsCZ-XTR-_qZCu1ZJnqfAo9Sb9_81ZCn1rTNr-BXbBXMkUF-M1hfS?key=mNimE1Z9oMGM36aSsqhnkQ**

**https://lh7-us.googleusercontent.com/docsz/AD_4nXfiL6dvPX7r5Y6YsbJ53l7Ro5VvOdSuk3jces2QQFcQR2HVrndIJ0iHFXGfl2w0hclbcomZTvyTn1HW9LiaLPGaSVBXqvuZiAoceyNY_W6cQ4jNV34s2pGBTtsZXA9V4CeOA17pre2h0WtHCXtR7mNvE7xm?key=mNimE1Z9oMGM36aSsqhnkQ**

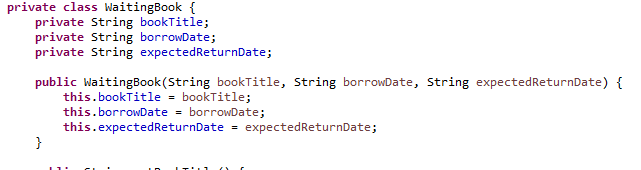
*Hình 2.45. Thao tác đọc/ ghi dữ liệu file user-account*

#### Xem lịch sử mượn sách (Bùi Thị Kiều Trang)

**

*Hình 2.46. Giao diện xem lịch sử mượn sách*

* **Nội dung nhập xuất:** Hệ thống sẽ hiện thị những thông tin về tên sách, ngày mượn, ngày trả dự kiến của những cuốn sách mà người dùng đã mượn và đang chờ phê duyệt để thuận lợi cho người dùng trong việc theo dõi mượn sách
* **Hướng đối tượng:** Lớp WaitingBook với các thuộc tính booktitle(tên sách), borrowdate(ngày mượn), expextedReturnDate(ngày trả dự kiến)

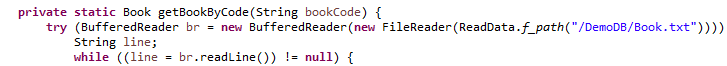
****

*Hình 2.47. Các thuộc tính lớp WaitingBook*

* **Thao tác file:**
  + Phương thức getWaitingBooks đọc tệp tin "borrow-slip.txt" để lấy danh sách các cuốn sách đang chờ mượn của người dùng hiện tại. Sử dụng BufferedReader và FileReader để mở tệp "borrow-slip.txt". https://lh7-us.googleusercontent.com/docsz/AD_4nXfl4mKybEsEP21l_peL1Zp3e5B5NVQ0xBiC72Tcn4QZFq0XRkhbuWGCwjcWLjTBhdLaE9kPKZxIavm4kMpyLIg5hisIwTh_GyE9ScL3Gmj1d6qmReEorYfGk9t34SRoN1LJZ-sDH2qfSACETU-jaV4LDknJ?key=mNimE1Z9oMGM36aSsqhnkQ

*Hình 2.48. Thao tác đọc/ghi dữ liệu từ file borrow-slip.txt*

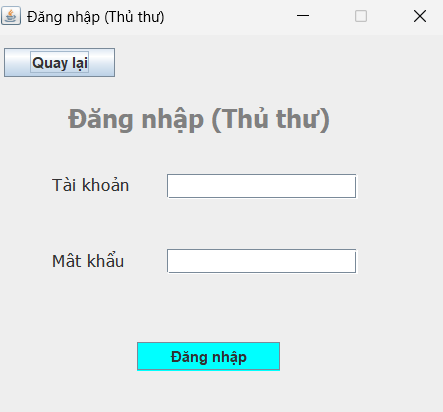
* + Phương thức getBookByCode đọc tệp tin "Book.txt" để lấy thông tin về sách dựa trên mã sách. Sử dụng BufferedReader và FileReader để mở tệp "Book.txt". Sử dụng br.readLine() để đọc từng dòng của tệp.



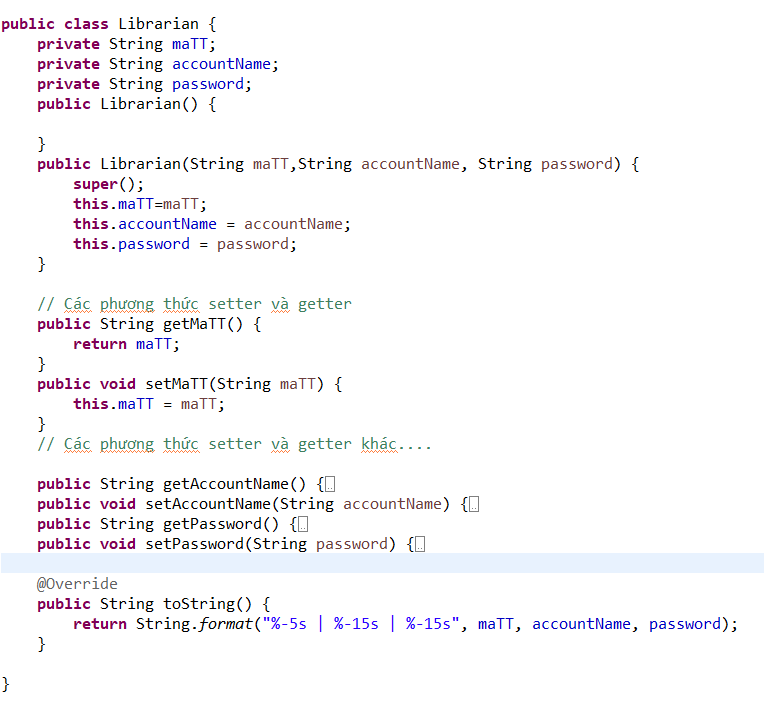
*Hình 2.49. Lấy đối tượng sách từ mã sách*

### Đối với thủ thư

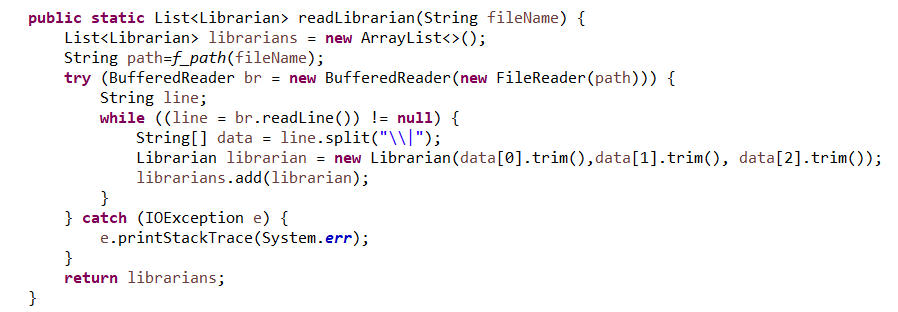
#### Đăng nhập (Nguyễn Thị Anh Phương)

**

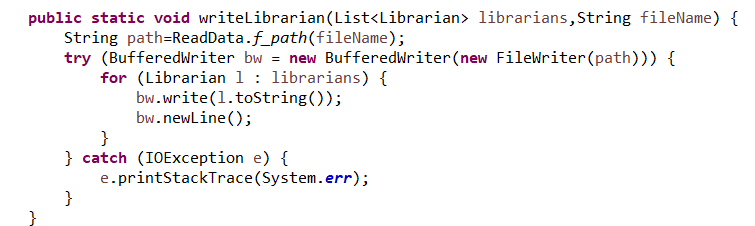
*Hình 2.50. Giao diện đăng nhập( Thủ thư)*

* **Nội dung nhập/xuất:** Để đăng nhập người dùng nhập vào tên tài khoản và mật khẩu. Tên tài khoản và mật khẩu của thủ thư được tạo sẵn và đưa lại cho thủ thư. Sau khi đăng nhập thành công sẽ hiển thị lên màn hình giao diện trang chủ chính.
* **Hướng đối tượng:**  Trong đoạn code, định nghĩa một lớp Librarian với các thuộc tính maTT, accountName, và password. Lớp này có các phương thức khởi tạo, các phương thức truy cập (getters) và thiết lập (setters) cho các thuộc tính, và một phương thức toString() được ghi đè để in ra thông tin của đối tượng Librarian.

*Hình 2.51. Định nghĩa lớp Librarian*

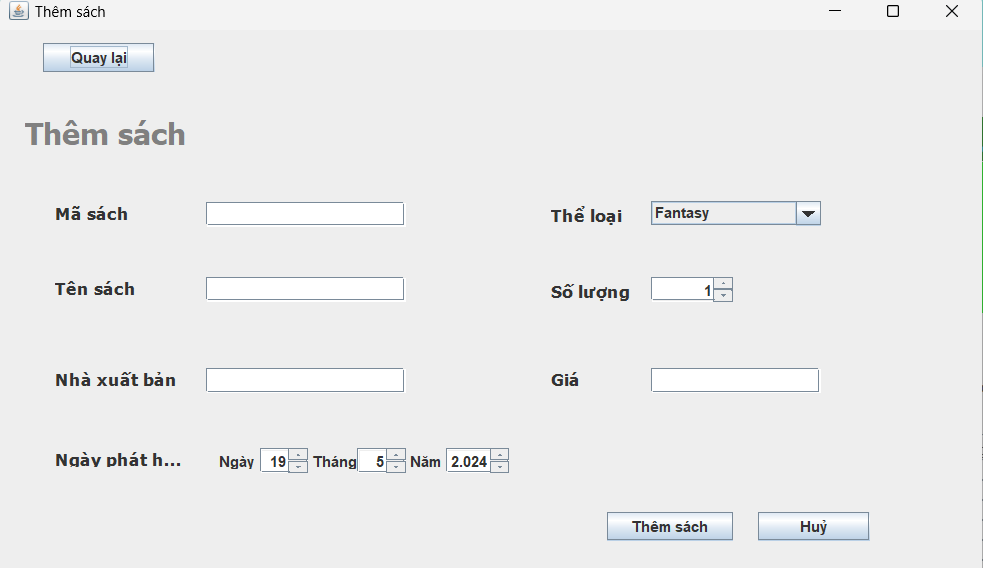
* **Bắt lỗi và gom rác:**
  + Trong hàm readLibrarian() và writeLibrarian() đều sử dụng bắt lỗi (exception handling) để xử lý các lỗi có thể xảy ra trong quá trình đọc và ghi dữ liệu.
  + Cơ chế bắt lỗi: nếu có bất kỳ lỗi nào xảy ra trong quá trình đọc/ghi tệp (VD: không tìm thấy tệp, không đọc/ghi được dữ liệu,...) ngoại lệ IOException sẽ được bắt và xử lý. Trong khối catch, ngoại lệ IOException được bắt và in ra thông tin lỗi sử dụng printStackTrace().
* **Thao tác file:** Các hàm readLibrarian() và writeLibrarian() được sử dụng đề đọc và ghi danh sách librarians từ/vào tệp tin. Các hàm này sử dụng BufferedReader và BufferedWriter để đọc và ghi tệp tin.

*Hình 2.52. Đọc dữ liệu từ tệp tin*



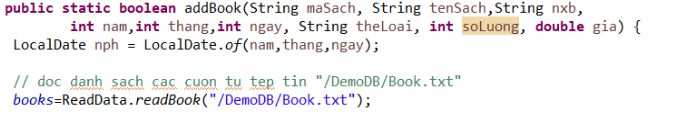
*Hình 2.53. Ghi dữ liệu vào tệp tin*

#### Thêm sách (Nguyễn Thị Anh Phương)



*Hình 2.54. Giao diện thêm sách*

* **Nội dung nhập/xuất:** Để thêm sách người dùng nhập đầy đủ các thông tin sách. Nếu nhập thiếu thông tin hệ thống sẽ hiển thị lên thông báo “vui lòng nhập đầy đủ thông tin”. Nếu nhập sai định giá hệ thống sẽ hiển thị lên màn hình “Giá không hợp lệ”. Khi nhập xong và đúng thông tin hệ thống sẽ đối chiếu với list mã sách trong hệ cơ sở dữ liệu. Nếu có mã sách trùng thì số lượng sách tăng lên tương ứng với số lượng nhập. Nếu không có mã sách trùng thì thông tin sách mới được cập nhật.
* **Hướng đối tượng:**
  + Lớp book được sử dụng để biểu diễn thông tin về một cuốn sách. Phương thức addBook thể hiện hành vi (logic) của việc thêm một cuốn sách vào danh sách.



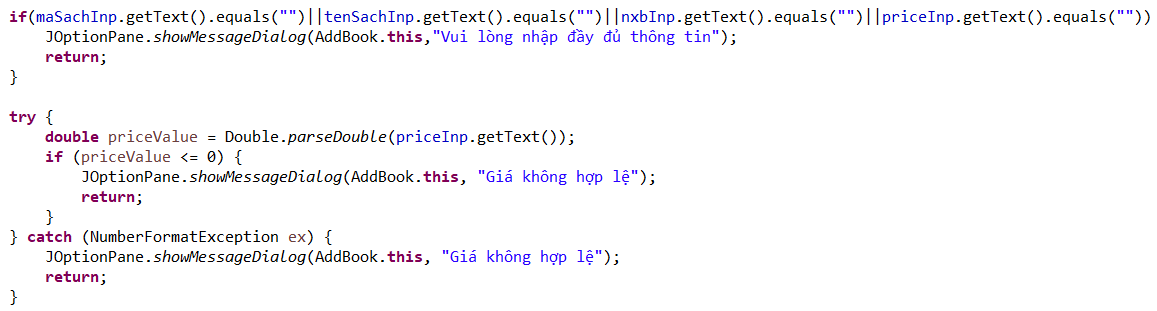
*Hình 2.55. Thêm sách vào cơ sở dữ liệu*

* + Sử dụng lớp JButton và ActionListener để tạo và xử lý sự kiện click vào nút “Thêm sách”.

https://lh7-us.googleusercontent.com/docsz/AD_4nXcK6zcZUAgfHVlJL5slRbcUsOk8m07Y1eAgVVN3RmLJK5vUr9aGKtAZ7jPp34-n_zkWUp2p7IPXnS10Lhu23XFjczLYbDwhZKof_MIHoGxPqO3yWQZGM1gRGLwNsVgt8FESRe3glRVmqIcy9Q2_-dV0PEM?key=mNimE1Z9oMGM36aSsqhnkQ

*Hình 2.56. Tạo và xử lý sự kiện click vào nút “Thêm sách”.*

* **Bắt lỗi và gom rác:**
  + Phần code kiểm tra xem người dùng có nhập đầy đủ thông tin hay không, nếu không hệ thống hiển thị thông báo lỗi lên màn hình bằng JOptionPane.showMessageDialog.
  + Phần code cũng kiểm tra xem giá sách có hợp lệ (lớn hơn 0) hay không, nếu không hợp lệ thì hiển thị thông báo lỗi.
  + Sử dụng try-catch để bắt lỗi khi chuyển đổi giá trị từ chuỗi sang số thực.  Nếu giá trị không hợp lệ, nó sẽ hiển thị thông báo lỗi.



*Hình 2.57. Bắt lỗi khi thêm sách*

* **Tập hợp**: Danh sách books được sử dụng để lưu trữ các đối tượng Book.

https://lh7-us.googleusercontent.com/docsz/AD_4nXesR5GsT4q4FICicWbQwYXfJ4wmjoK8GjZSqeZFGETmMAgDVmH9Pr1iZses1DNy_3d2k1FPcjWtJGrpc22p8pxxHK_3_Z7oCRUgIkr1N_otGqI4FKIFTNv4t84dzEcC3vRuWcS9OMxc29eYIxg0alGBuQM?key=mNimE1Z9oMGM36aSsqhnkQ

*Hình 2.58. Đọc sách từ tệp tin “DemoDB/Book.txt”*

* **Thao tác file:**
  + Đoạn code sử dụng phương thức ReadData.readBook để đọc danh sách các cuốn sách từ tệp tin "/DemoDB/Book.txt".

https://lh7-us.googleusercontent.com/docsz/AD_4nXddzUCT_n041_-lmHuBOC9Xk5K-ilbwXwzkATJLXfgpyGkHYM5-iRHTniszKppACT4__NoxrslLGvwvEcbiidjM_cK28NFJec0Mm0CpxP3Kcpdsi9ODOVQS0IqCBFOyYURFXbios38GDhDV_Uitp2dFAto?key=mNimE1Z9oMGM36aSsqhnkQ

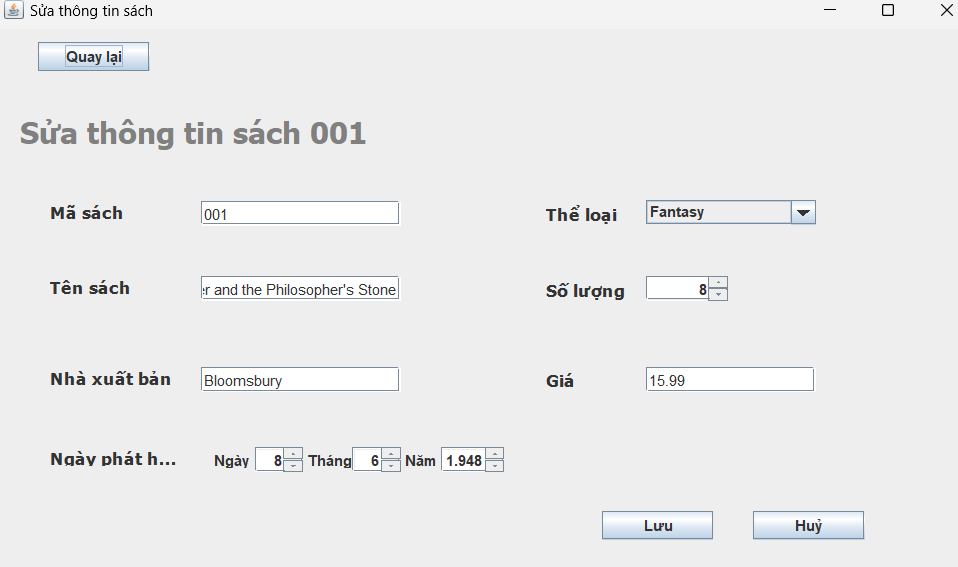
*Hình 2.59. Đọc sách từ tệp tin “DemoDB/Book.txt”*

* + Phương thức WriteData.writeBook được sử dụng để ghi danh sách các cuốn sách vào tệp tin "/DemoDB/Book.txt".

https://lh7-us.googleusercontent.com/docsz/AD_4nXfXhVkcw-pHtYZU8M9VtZp7GAmeQKt76PWOmpavhxcNXdqzwwR1ALKD3ekquX6apTMkbtItdXsHKqlac6jy8wqCR4Fq9MANNP7ErckHkZ8O00tXi4klnZfLG-j18xBXskayDc-xwLTzCjIZ9bPBH41PNFo?key=mNimE1Z9oMGM36aSsqhnkQ

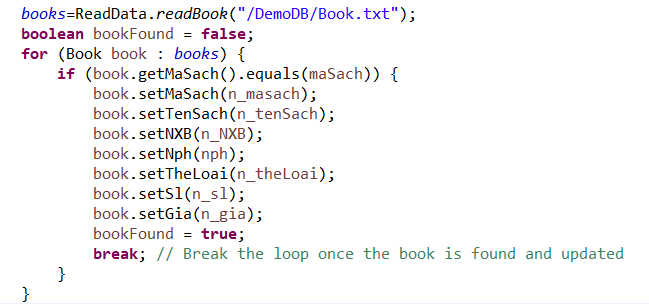
*Hình 2.60. Ghi sách vào tệp tin “DemoDB/Book.txt”*

#### Sửa thông tin sách (Nguyễn Thị Anh Phương)



*Hình 2.61. Giao diện sửa thông tin sách*

* **Nội dung nhập/xuất:** Để sửa thông tin sách thủ thư cần nhập đầy đủ thông tin sách. Nếu nhập thiếu thông tin hệ thống sẽ hiển thị lên thông báo “Vui lòng nhập đầy đủ thông tin”. Nếu nhập không đúng định dạng giá hệ thống hiển thị lên thông báo “Giá không hợp lệ”. Nếu nhập đầy đủ và đúng thông tin dữ liệu sẽ được ghi vào file “Book.txt”.
* **Hướng đối tượng:**
  + Sử dụng lớp Book để biểu diễn thông tin của một cuốn sách với các thuộc tính như mã sách, tên sách, NXB, ngày phát hành, thể loại, số lượng và giá. Sử dụng các phương thức getter và setter để truy cập và cập nhật các thuộc tính của đối tượng Book.



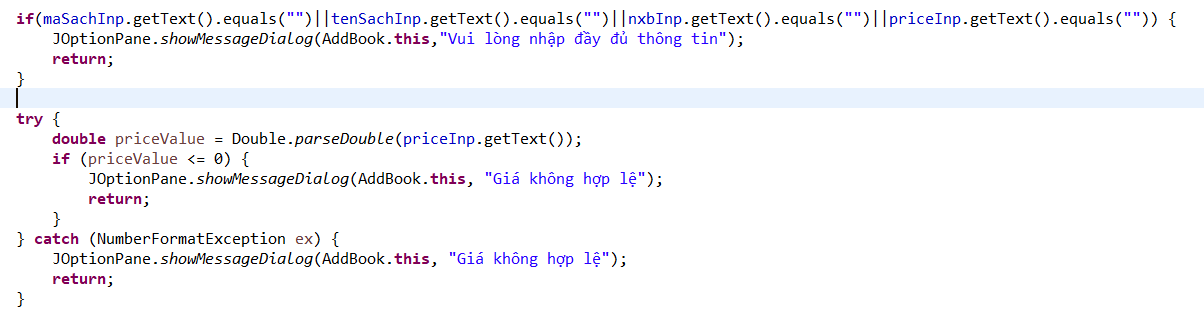
*Hình 2.62. Cập nhật thông tin sách*

* + Sử dụng lớp JButton và ActionListener để tạo nút "Lưu" và xử lý sự kiện khi người dùng nhấn nút.

https://lh7-us.googleusercontent.com/docsz/AD_4nXdh27oGe0-SYNA0AD1lZrob6iJqRybm45Z0cG_UkBrPzTEmyh4VUSZqC-beD38V_cvldJ4SlfC8y7xz0UeO7mEnWZpwOeYfKl9_UF7fwLKtPFwQHHQg3Bg87PfrJKYWgVfqM9yCdNOc1-kSRvGjmLRsFTBA?key=mNimE1Z9oMGM36aSsqhnkQ

*Hình 2.63. Tạo nút submit và thêm ActionListener*

* **Bắt lỗi và gom rác:**
  + Phần code cũng kiểm tra xem giá sách có hợp lệ (lớn hơn 0) hay không, nếu không hợp lệ thì hiển thị thông báo lỗi.
  + Sử dụng try-catch để bắt lỗi khi chuyển đổi giá trị từ chuỗi sang số thực.  Nếu giá trị không hợp lệ, nó sẽ hiển thị thông báo lỗi.
  + Kiểm tra các trường nhập liệu có trống hay không. Nếu có trường trống, hệ thống sẽ hiển thị thông báo yêu cầu nhập đầy đủ thông tin.



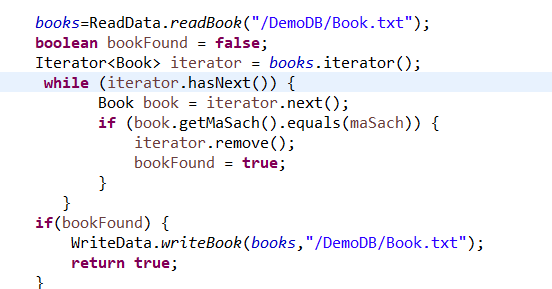
*Hình 2.6.4. Bắt lỗi khi sửa thông tin sách*

* **Tập hợp:** Gán vào biến books một danh sách chứa thông tin về các cuốn sách.

https://lh7-us.googleusercontent.com/docsz/AD_4nXfd2mNCKgsKDQ2Yl5d5-NsvsrGGWNKUlPW7FEXXjBYjMqQvcI20GrYnxBrNndZlLHorjHKHMJrpN3VR4OlQlhlGSI0eXK6YobLhO15dndd0mOVu5JB7cCF6TCMHE7AsJj1AG6LDf-pGK4joc7C3dXLQ7rsT?key=mNimE1Z9oMGM36aSsqhnkQ

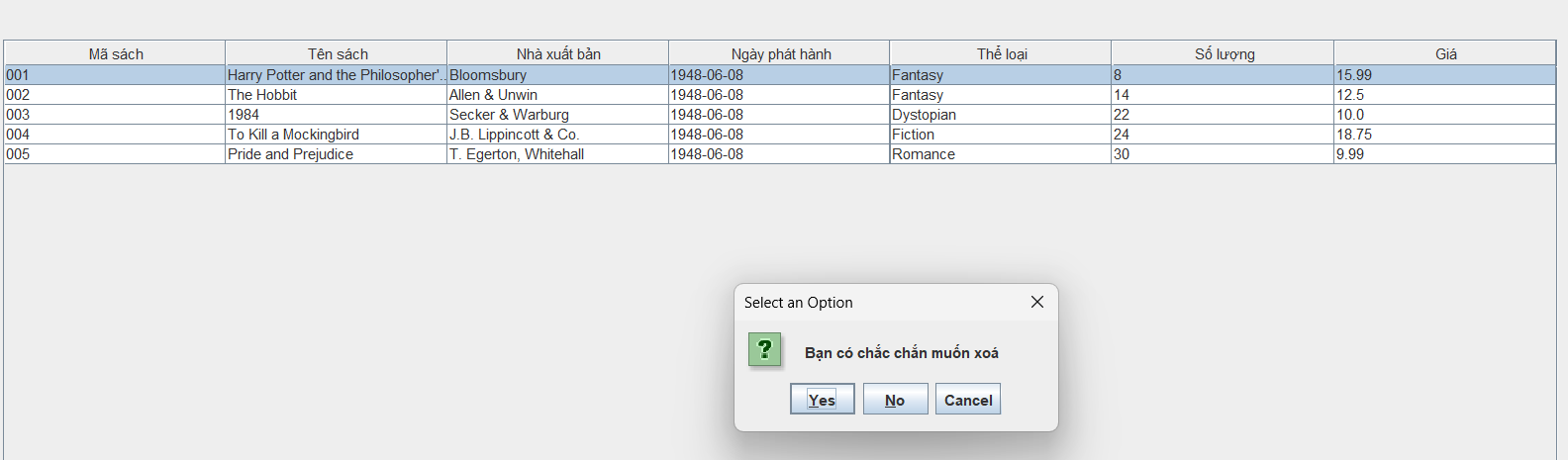
*Hình 2.65. Đọc dữ liệu sách từ tệp*

* **Thao tác file:**
  + Sử dụng phương thức readBook() của lớp ReadData để đọc dữ liệu về các cuốn sách từ tệp tin “/DemoDB/Book.txt” và lưu vào danh sách books.
  + Sử dụng phương thức writeBook() của lớp WriteData để ghi lại danh sách các cuốn sách (bao gồm cả cuốn sách đã được cập nhật) vào tệp tin “/DemoDB/Book.txt.”



*Hình 2.66. Đọc và ghi dữ liệu sách từ/vào tệp tin*

#### Xóa thông tin sách (Nguyễn Công Thành)



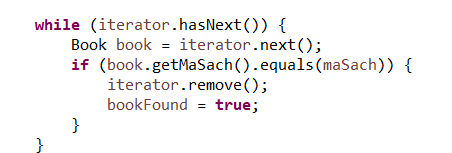
*Hình 2.67. Giao diện xóa thông tin sách*

* **Nội dung nhập/xuất:** Nhận mã sách cần xóa, đầu ra là true hoặc false để thông báo đã xóa thành công hay chưa. Nếu xóa thành công, danh sách sách sẽ được cập nhật lại.
* **Hướng đối tượng:**
  + Sử dụng lớp Book để biểu diễn thực thể sách.

https://lh7-us.googleusercontent.com/docsz/AD_4nXdSBwhTroej18W5e6F0ZV09Be3yM6j-w9vfp4s7lvbfz9devW8p0vF52wHiFI6lHvJtAiolWXEFGnDBFJ8nWRdK4ZFpiKNkztmpfFtRBp-Aw-J_5wHbpyjD99zRYi5dYIWGgICy-ZZIkjKqpQe7DjTFKkmn?key=mNimE1Z9oMGM36aSsqhnkQ

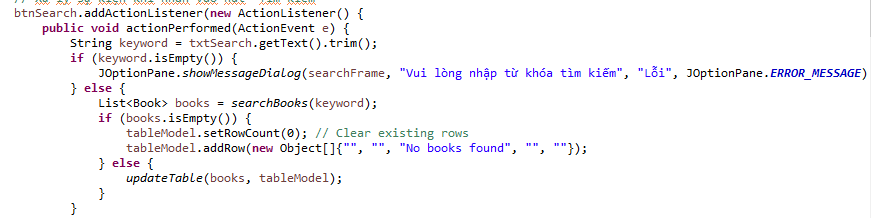
*Hình 2.68. Cập nhật kết quả tìm kiếm sách*

* + Sử dụng phương thức getMaSach() của lớp Book để lấy mã sách của đối tượng.



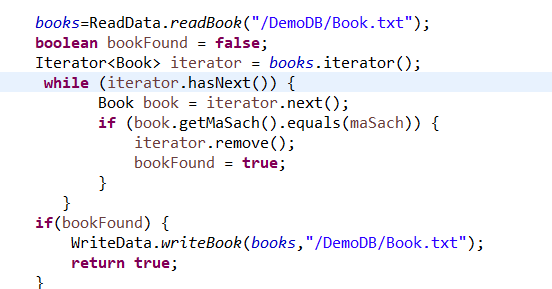
*Hình 2.69.  Lấy thông tin sách*

* **Bắt lỗi và gom rác:**
  + Lấy từ khóa tìm kiếm từ trường văn bản txtSearch và loại bỏ các khoảng trắng ở đầu và cuối chuỗi bằng cách sử dụng phương thức trim().
  + Nếu người dùng không nhập gì, một hộp thoại thông báo lỗi sẽ xuất hiện để yêu cầu người dùng nhập từ khóa tìm kiếm.
  + Nếu người dùng đã nhập, hàm searchBooks(keyword) sẽ được gọi để tìm kiếm các cuốn sách phù hợp với từ khóa. Kết quả trả về là một danh sách các cuốn sách.
  + Nếu danh sách kết quả rỗng, bảng hiển thị kết quả table sẽ được xóa hết các hàng hiện có và thêm một hàng thông báo "No books found".
  + Nếu danh sách kết quả không rỗng, hàm updateTable(books, tableModel) được gọi để cập nhật bảng hiển thị với danh sách kết quả mới.



*Hình 2.70. Bắt lỗi với tìm kiếm sách*

* **Thao tác file:**
  + Đọc dữ liệu từ file "/DemoDB/Book.txt" và lưu vào danh sách books.
  + Nếu bookFound = true sẽ ghi lại danh sách books (với cuốn sách đã bị xóa) vào file "/DemoDB/Book.txt".



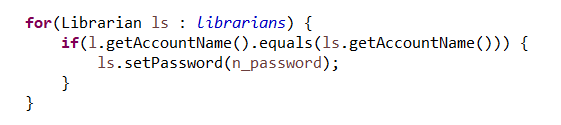
*Hình 2.71. Đọc/ghi dữ liệu sách từ/vào tệp tin*

#### Đổi mật khẩu (Nguyễn Thị Anh Phương)



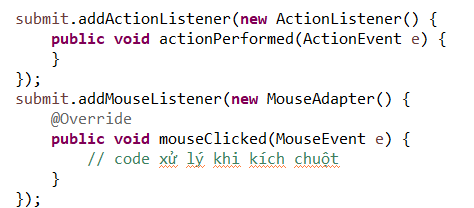
*Hình 2.72. Giao diện đổi mật khẩu (Thủ thư)*

* **Nội dung nhập/xuất:** Cung cấp thông tin về mật khẩu cũ, mật khẩu mới và xác nhận mật khẩu mới. Nếu như không nhập đầy đủ thông tin, mật khẩu cũ không chính xác, xác nhận mật khẩu không khớp thì sẽ hiển thị thông báo lỗi. Đổi mật khẩu thành công thông tin sẽ được ghi vào file "/DemoDB/Librarian.txt".
* **Hướng đối tượng:**
  + Sử dụng lớp user-account để đại diện cho người dùng và các phương thức của lớp này, như getAccountName() và setPassword().

****

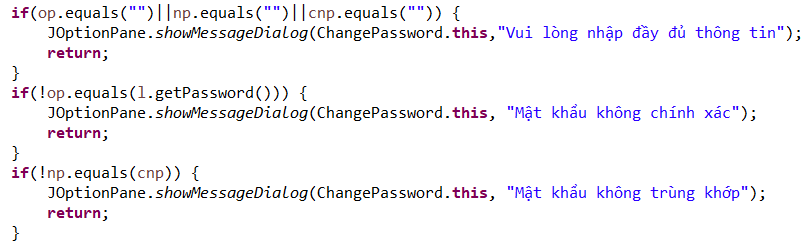
*Hình 2.73. Cập nhật mật khẩu của thủ thư*

* + Sử dụng các khái niệm hướng đối tượng như lớp, giao diện và hướng đối tượng ẩn danh để xử lý các sự kiện liên quan đến nút “Đổi mật khẩu”



*Hình 2.74. Xử lý sự kiện khi nhấn nút Đổi mật khẩu*

* **Bắt lỗi và gom rác:** Nếu có lỗi xảy ra, chương trình sẽ hiển thị thông báo lỗi thông qua JOptionPane.showMessageDialog() và dừng thực thi bằng lệnh return. Cơ chế bắt lỗi nhằm kiểm tra các trường hợp như người dùng không  nhập đủ thông tin, mật khẩu cũ không chính xác hoặc mật khẩu mới không trùng khớp.



*Hình 2.75. Bắt lỗi khi đổi mật khẩu*

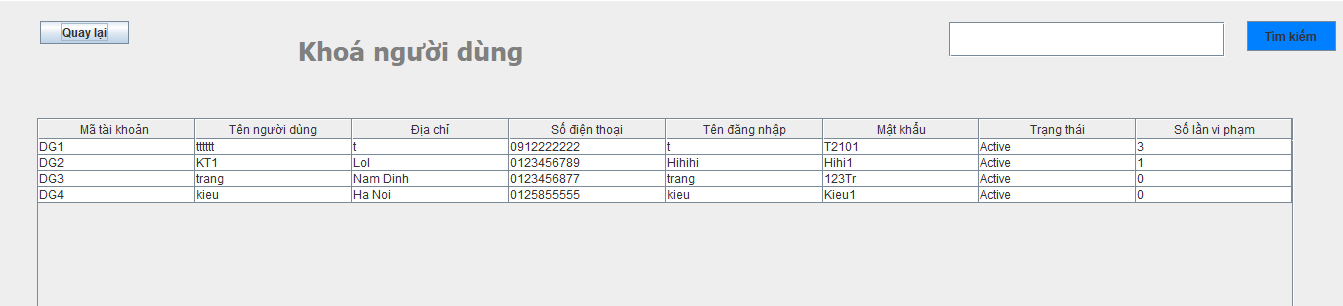
* **Tập hợp:**
  + Tập hợp: Sử dụng danh sách Librarians để lưu trữ các đối tượng Librarian
* **Thao tác file:**
  + Đoạn code đọc dữ liệu từ file "/DemoDB/Librarian.txt" và ghi dữ liệu vào file "/DemoDB/Librarian.txt".

https://lh7-us.googleusercontent.com/docsz/AD_4nXfTDwRS_ySwYAIh2oBP4PKYQr4IsAE90j5ON5i1DIsCbiKYQnWh8kJkmIkbnvECBmfEnCNsl9AK9pqhsag91hx4HoVWFP-fPLBXM1n1tA4pM6C5WMDnY20nsFFjK1vWFhNdCxmnHgJ_sScYE0EPOmKUrrty?key=mNimE1Z9oMGM36aSsqhnkQ

https://lh7-us.googleusercontent.com/docsz/AD_4nXf9xRY2ddFYTzKfqmS2eXwjy8GDFAteXuI-BKfni2_aLUkLwsH_b_Kz27EhuJHQqMbo65EtK_s7bgHTxmeL4yq4Q9T1iOFJtnm8jIMzrWtDhpVQfJcSuzceJyAd8ZSPpvGm90DvUGVXmZhbLPEY7iPUOlY?key=mNimE1Z9oMGM36aSsqhnkQ

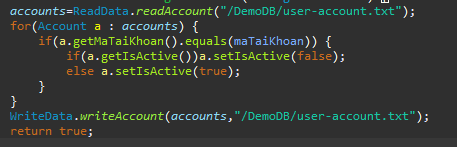
*Hình 2.76. Đọc/ghi dữ liệu thủ thư từ/vào tệp tin*

#### Khóa/mở tài khoản (Nguyễn Công Thành)



*Hình 2.77. Giao diện khóa/mở tài khoản*

* **Nội dung nhập/xuất:** Cung cấp thông tin về mã tài khoản cần khoá hoặc mở khoá. Khoá hoặc mở khoá tài khoản người dùng thành công, trạng thái của người dùng sẽ được thay đổi và ghi vào file user-account.txt.
* **Hướng đối tượng:**
  + Sử dụng lớp Librarian để đại diện cho thủ thư và các phương thức của lớp này, như getMaTaiKhoan() và setIsActive().

****

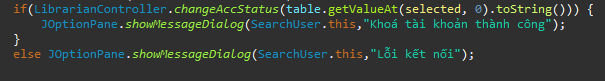
*Hình 2.78. Phuơng thức Khóa tài khoản*

* + Sử dụng các khái niệm hướng đối tượng như lớp, giao diện và hướng đối tượng ẩn danh để xử lý các sự kiện liên quan đến nhấn chuột vào bảng,

https://lh7-us.googleusercontent.com/docsz/AD_4nXe9GP6hRFK751wfIof-1PXPqg_1-XWg5UZm7Lu9UrYwxi_AsQ82aaTg6bGt3rqLlrtxT4JLDXEZiz66lvYpO6kDHEKzjwjgIWHZQ6UKvEqdLW4Py0sKGtaFl9TGXYb0VORdezqD5kv8QlG8hWN_fePFcyNS?key=mNimE1Z9oMGM36aSsqhnkQ

*Hình 2.79. Sự kiện kích chuột*

* **Bắt lỗi và gom rác:** Nếu có lỗi xảy ra, chương trình sẽ hiển thị thông báo lỗi thông qua JOptionPane.showMessageDialog() và dừng thực thi bằng lệnh return. Cơ chế bắt lỗi nhằm kiểm tra trường hợp lỗi kết nối.



*Hình 2.80. Bắt lỗi khi khóa tài khoản*

* **Tập hợp:**
  + Sử dụng danh sách accounts để lưu trữ các đối tượng Account
* **Thao tác file:**
  + Đoạn code đọc dữ liệu từ file "/DemoDB/user-account.txt" và ghi dữ liệu vào file "/DemoDB/account.txt".

https://lh7-us.googleusercontent.com/docsz/AD_4nXd7OU905sDU-G-SWr-i607sGbFNh0-XjOG-CUjZvvMecyPxR-9rl_r36i8LM6fMMSZSXosMPtRneq2zXOXJayPhTbFGuYfDK_jnCsJlLBsQUfrT8h9fymh7KqtyYN4sJl3ao6WqjoG4tFrRVDcA16OGm2xP?key=mNimE1Z9oMGM36aSsqhnkQ

https://lh7-us.googleusercontent.com/docsz/AD_4nXd83b6Zoc_lfR0El8XCoTmKmHUj5b1vnekfoHY1m3MDxMg5OFUbelamFKDxfTK9p84BkV6fUwYfXLQh1LOeQ7B7a0_qZwukzf0KktXX6vs51tqtpiQD3lE_gM5EY-rxEQZGbmtlqBss6NOGMYyEiYabatYz?key=mNimE1Z9oMGM36aSsqhnkQ

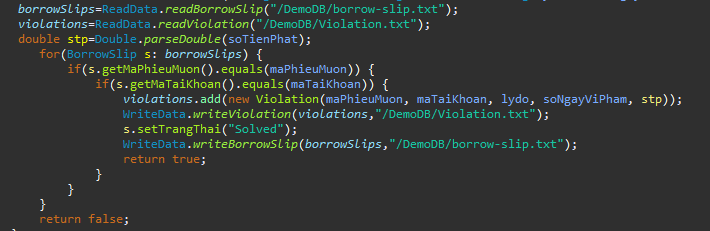
*Hình 2.81. Đọc/ghi dữ liệu từ tệp user-account*

#### Phiếu vi phạm (Nguyễn Công Thành)



*Hình 2.82. Giao diện tạo phiếu vi phạm*

* **Nội dung nhập/xuất:** Cung cấp thông tin về mã phiêu mượn, mã tài khoản vi phạm, số ngày vi phạm, số tiên phạt, lý do phạt. Tạo phiếu vi phạm thành công, một bản ghi mới về phiếu vi phạm sẽ được thêm vào file Violation.txt. Trạng thái của phiếu mượn sẽ chuyển về “Solved”.
* **Hướng đối tượng:**
  + Sử dụng lớp Violation để đại diện cho phiếu vi phạm, lớp BorrowSlip đại diện cho phiếu mượn và các phương thức của lớp BorrowSlip như getMaPhieuMuon() và getMaTaiKhoan() setTrangThai().

****

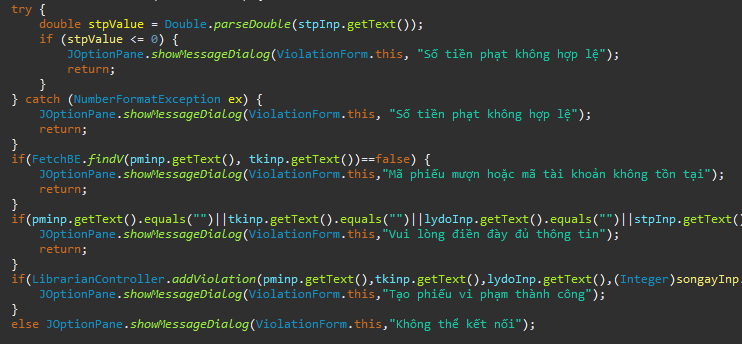
*Hình 2.83. Cập nhật trạng thái*

* + Sử dụng các khái niệm hướng đối tượng như lớp, giao diện và hướng đối tượng ẩn danh để xử lý các sự kiện liên quan đến nút “Submit”.

https://lh7-us.googleusercontent.com/docsz/AD_4nXfw4KoMbSz3fOBeblTvaZveBmwc61XMGaN5cLmnOsHa08L37-9inZM480qnyk7jS2pro_krDN_xnXVkPou3dmCzntfr3wV3mIQubCmtwxjMLghFFIwTNQCo3KtZB9-fHYcX2W3gppe0UKHafgSJ6kb25eY?key=mNimE1Z9oMGM36aSsqhnkQ

*Hình 2.84. Xử lý sự kiện khi kích vào nút “Submit”*

* **Bắt lỗi và gom rác:** 
  + Nếu có lỗi xảy ra, chương trình sẽ hiển thị thông báo lỗi thông qua JOptionPane.showMessageDialog() và dừng thực thi bằng lệnh return. Cơ chế bắt lỗi nhằm kiểm tra các trường hợp như dữ liệu không phù hợp, mã tài khoản không khớp với mã phiếu mượn, thông tin không đầy đủ, không thể kết nối.



*Hình 2.85. Bắt lỗi khi tạo phiếu mượn*

* **Tập hợp:**
  + Tập hợp: Sử dụng danh sách violations để lưu trữ các đối tượng Violation
* **Thao tác file:**
  + Đoạn code đọc dữ liệu từ file "/DemoDB/Violation.txt" và ghi dữ liệu vào file "/DemoDB/violation.txt".Đọc, ghi  file “/DemoDB/borrow-slip.txt”.

https://lh7-us.googleusercontent.com/docsz/AD_4nXd5afi0IlAPqsY6zrPIlrznJGPYmjF_6YJJXHYH84FmgDpgClKvpH61Kj-lG0ti2lvOjuYC6ZArFjRA-o0zaHlBGNQJ81oTTIGZL8lvc1GroHLGfwVniFspwTGYPinnrG9kYcnkOe2emLsOEpcd-jmMML_1?key=mNimE1Z9oMGM36aSsqhnkQ

https://lh7-us.googleusercontent.com/docsz/AD_4nXes4j6EDN0QqN7JA10z9vhwqhNR17cqXzVvYlgrypxG7B7z9HeKk5BkOuDu8vNxS5g9cC9WLMUdUxYht9504syGsW-gVEu6kYAPfO2nZ-fSoKpuaqM3PRhgQxfT2Q90toYIVNhKIpwkQty1Y8xy2f7cl5u5?key=mNimE1Z9oMGM36aSsqhnkQ

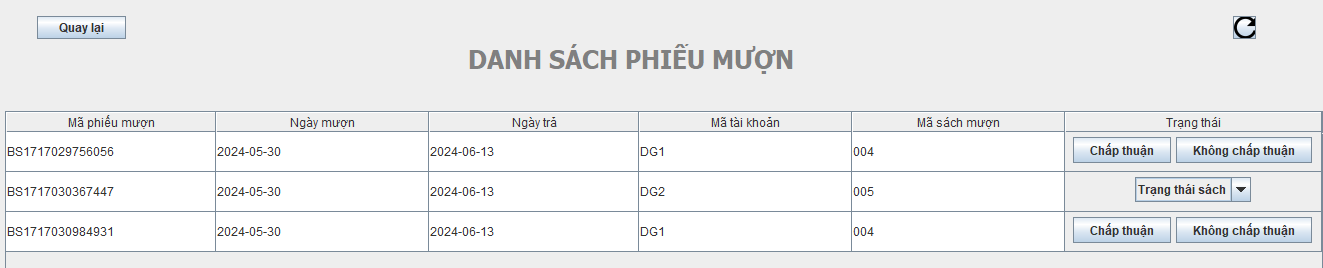
*Hình 2.86. Đọc/ghi dữ liệu từ file "/DemoDB/Violation.txt"*

https://lh7-us.googleusercontent.com/docsz/AD_4nXeLCjfeuB4sy9NcBFmWqcS4q1NM3bbvv9-KSYsWKPOgwzIApDIYygb3XtUqHKttGMzXFbxMcyFqO7lKHAl0KmrfqmVuiznqKVUkCSDAcasM1fSv4Rz2UX0cmVVbM4d9wGZdEdgeKmI0Cey_zDVUJQz-TBl7?key=mNimE1Z9oMGM36aSsqhnkQ

https://lh7-us.googleusercontent.com/docsz/AD_4nXcO7lv4KkyPYRb0Wme2EInqzSOQ8D5Lx8p_TZbzF_D1nkOjQtrNcXV4iN0ME7VeYZLC5Snt6QHpbPOT_m5leQAuABreLpVHLfYFmc_AgdlI6TFutuRPwL8NTqe6b224hBRH6LEKRirlfF_qTVXSguycqp4?key=mNimE1Z9oMGM36aSsqhnkQ

*Hình 2.87. .Đọc/ghi file “/DemoDB/borrow-slip.txt”.*

#### Phê duyệt mượn/trả sách (Nguyễn Công Thành)



*Hình 2.88. Giao diện phê duyệt mượn/trả sách*

* **Nội dung nhập/xuất:** 
  + Cung cấp thông tin về mã phiếu mượn, trạng thái của mã phiếu mượn. Mỗi phiếu mượn có thể có 1 trong các trạng thái “Pending” (chờ phê duyêt), “Dissaprove” (không đồng ý cho mượn), “Approved” (đồng ý mượn).Khi phiếu mượn ở trạng thái “Approved”, người dùng có thể chọn hành động để trả sách hoặc báo với thủ thư là làm mất sách.
  + Nếu trả sách đúng hạn trạng thái sẽ được chuyển thành onTime, quá hạn sẽ là Expired. Mất sách sẽ là Lost. Trạng thái mới của phiếu mượn sẽ được cập nhật vào file “/DemoDB/borrow-slip.txt”.Khi đồng ý mượn sách số lượng sách của đầu sách đó sẽ giảm đi 1.Khi trả số lượng sẽ được khôi phục.Số lượng mới sẽ được cập nhật vào “/DemoDB/Book.txt”.Trạng thái có thể mượn của sách có thể được thay đổi và cập nhật.
* **Hướng đối tượng:**
  + Sử dụng lớp BorrowSlip đại diện cho phiếu mượn và các phương thức của lớp BorrowSlip như getMaPhieuMuon() và getTrangThai() setTrangThai(), getMaSach(). Lớp Book đại diện cho sách và các phương thưc của lớp Book như getMaSach(), reduceQuantity() và getSl(), setAvaiable().
  + Sử dụng các khái niệm hướng đối tượng như lớp, giao diện và hướng đối tượng ẩn danh để xử lý các sự kiện liên quan đến nút “Approve”,”Reject” và “Return”.



*Hình 2.89. Bắt lỗi khi tạo phiếu mượn*

* **Bắt lỗi và gom rác:** Không có lỗi đáng kể.
* **Tập hợp:** 
  + Sử dụng danh sách borrowSlips để lưu trữ các đối tượng BorrowSlip.
* **Thao tác file:**
  + Đoạn code đọc/ ghi dữ liệu trong file "/DemoDB/borrow-slip.txt" .Đọc, ghi dữ liệu trong  file “/DemoDB/Book.txt”.

https://lh7-us.googleusercontent.com/docsz/AD_4nXe9WrpZ-LOiyTU4nuoFEBK3m_zldRBNjfp2q_kiZxP7S5Rr0Munyh6gbarqZb4qD2COh1LnsEmFSm4hr0g_HJ3Zp4xmCqM9d2_SGmjWABVJgHPR7BipVwb6PbQAU-S7uOCp25nK7-vFZYjWssoUp2YjOKEx?key=mNimE1Z9oMGM36aSsqhnkQ

https://lh7-us.googleusercontent.com/docsz/AD_4nXdexIS0E6BYsOhpUmWZQlx8QSkiwb_nHDULaSBJw9SUjTxmIKmgDQ8zi4mct7ntpTS8Bvjpl2PN_ztkOnCM6bWYX20dprPx45BjMjl9eykVhUUOdoXd1FAMikkRvesZ2Ack8Hr1zmCL2PTJqT9_k-TOdABX?key=mNimE1Z9oMGM36aSsqhnkQ

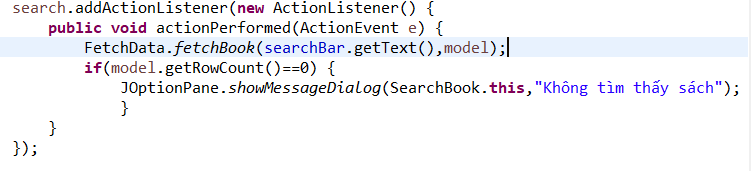
Hình 2.90. Đọc/ghi dữ liệu từ file "/DemoDB/borrow-slip.txt "

#### Tìm kiếm sách (Nguyễn Công Thành)



*Hình 2.91. Giao diện tìm kiếm sách*

* + - * **Nội dung nhập/xuất:** Nhập từ khóa về sách cần tìm kiếm. Hệ thống lấy dữ liệu trong file và hiển thị lên màn hình thông tin liên quan.
      * **Hướng đối tượng:**
  + Sử dụng lớp ActionListener để xử lý sự kiện khi người dùng nhấn nút tìm kiếm và các lớp FetchData và SearchBook.



*Hình 2.92. Xử lý sự kiện tìm kiếm và hiển thị thông báo*

* + Tạo đối tượng book mới

*Hình 2.93. Tạo đối tượng book mới*

* + Tạo đối tượng EditBook



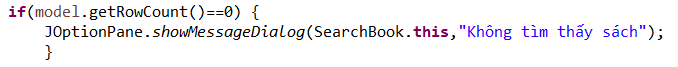
*Hình 2.94. Tạo đối tượng EditBook mới*

* + Gọi phương thức delBook() của LibrarianController

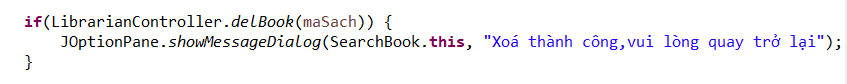


*Hình 2.95. Gọi phương thức delBook()*

* **Bắt lỗi và gom rác:**
  + Kiểm tra model.getRowCount()==0 để xác định xem có kết quả tìm kiếm hay không. Nếu không có kết quả, nó hiển thị một thông báo lỗi sử dụng JOptionPane.showMessageDialog()**.**
  + Sử dụng JOptionPane.showConfirmDialog() để hiển thị hộp thoại xác nhận xóa. Điều này giúp tránh các lỗi do người dùng vô tình xóa sách.

****





*Hình 2.96. Bắt lỗi khi tìm kiếm sách*

* **Tập hợp:** Sử dụng model để lưu trữ kết quả tìm kiếm.



*Hình 2.97. Lấy thông tin tìm kiếm*

# 

# PHẦN 3: KẾT LUẬN VÀ BÀI HỌC KINH NGHIỆM

## 3.**1. Nội dung đã thực hiện**

### 3.1.1. Kiến thức và kỹ năng đã học được

* Hiểu và vận dụng các kỹ thuật phân tích và thiết kế hệ thống như mô hình hóa chức năng (use case), mô hình hóa dữ liệu (biểu đồ thực thể liên kết).
* Thiết kế và xây dựng giao diện người dùng, thủ thư.
* Lập trình các chức năng chính của hệ thống như đăng nhập, quản lý sách, …
* Áp dụng các kỹ năng làm việc nhóm, phân công công việc và lập kế hoạch.

### 3.1.2. Chuẩn đầu ra đạt được

* Lập trình và triển khai được các chức năng chính của hệ thống.
* Thiết kế được giao diện hệ thống của người dùng và thủ thư
* Thực hành kỹ năng phân tích, thiết kế, lập trình và quản lý tiến độ.

### 3.1.3. Bài học kinh nghiệm

* Tầm quan trọng của lên kế hoạch và phối hợp nhóm hiệu quả.
* Cách tiếp cận và giải quyết các vấn đề phát sinh trong quá trình triển khai, thực hiện bài tập lớn.

## 3.2. Hướng phát triển

* Tính khả thi: Chủ đề về ***“Xây dựng phần mềm quản lý sách”*** có tính khả thi cao vì đây là một vấn đề thực tế, cụ thể.
* Yêu cầu khi thực hiện:
  + Đề tài này yêu cầu người thực hiện phải có kiến thức về phân tích thiết kế hệ thống và một số kiến thức cơ bản về quản lý sách.
  + Các kỹ năng cần có bao gồm: lập trình, thiết kế giao diện, phân tích yêu cầu, mô hình hóa dữ liệu,...
* Thuận lợi và khó khăn
  + Thuận lợi: Đề tài ***“Xây dựng phần mềm quản lý sách”*** mang tính thực tế, dễ tiếp cận, dễ tìm hiểu.
  + Khó khăn: Thu thập và chọn lọc tài liệu về yêu cầu từ người dùng, cũng như thiết kế giao diện người dùng thân thiện.

# TÀI LIỆU THAM KHẢO

* + - * Nguyễn Bá Nghiễn (Chủ biên), Ngô Văn Bình,Vương Quốc Dũng, Đỗ Sinh Trường (2000), *Giáo trình lập trình hướng đỗi tượng với Java*, Nhà xuất bản thống kê.