**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**PHÂN HIỆU TẠI TP. HỒ CHÍ MINH**

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO BÀI TẬP LỚN**

**CÔNG NGHỆ JAVA**

**ĐỀ TÀI: QUẢN LÝ THƯ VIỆN**

Giảng viên hướng dẫn: TS. NGUYỄN VĂN DŨ

Sinh viên thực hiện: NGUYỄN LÊ GIA MỸ

Lớp : CÔNG NGHỆ THÔNG TIN

Khoá :63

MSSV : 6351071046

Tp. Hồ Chí Minh, năm 2024

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**PHÂN HIỆU TẠI TP. HỒ CHÍ MINH**

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO BÀI TẬP LỚN**

**CÔNG NGHỆ JAVA**

**ĐỀ TÀI: QUẢN LÝ THƯ VIỆN**

Giảng viên hướng dẫn: TS. NGUYỄN VĂN DŨ

Nhóm thực hiện: 2

Thành viên: NGUYỄN LÊ GIA MỸ

ĐINH KIM YẾN NHI

Tp. Hồ Chí Minh, năm 2024

# MỤC LỤC

[MỤC LỤC i](#_Toc165630225)

[LỜI CẢM ƠN iv](#_Toc165630226)

[DANH MỤC CHỮ VIẾT TẮT...... v](#_Toc165630227)

[DANH MỤC HÌNH ẢNH vi](#_Toc165630228)

[CHƯƠNG 1. GIỚI THIỆU 1](#_Toc165630229)

[1.1. Tổng quan về đề tài 1](#_Toc165630230)

[1.2. Lý do chọn đề tài 1](#_Toc165630231)

[1.3. Mục tiêu và phạm vi của đề tài 2](#_Toc165630232)

[CHƯƠNG 2. BÀI TOÁN 4](#_Toc165630234)

[2.1. Giới thiệu bài toán 4](#_Toc165630235)

[2.2. Các chức năng của bài toán 5](#_Toc165630236)

[2.3. Class diagram 7](#_Toc165630237)

[2.4. Lược đồ cơ sở dữ liệu 8](#_Toc165630238)

[2.5. Các công nghệ/API sử dụng 8](#_Toc165630239)

[CHƯƠNG 3. KẾT QUẢ 9](#_Toc165630240)

[3.1. Tổng quan 9](#_Toc165630241)

[3.2. Màn hình chính 10](#_Toc165630242)

[3.3. Màn hình quản lý độc giả 12](#_Toc165630243)

3.4. Màn hình quản lý sách [17](#_Toc165630244)

[3.5. Màn hình quản lý mượn trả 25](#_Toc165630245)

[3.6. Màn hình thống kê – báo cáo 29](#_Toc165630245)

[CHƯƠNG 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 31](#_Toc165630246)

[5.1. Kết luận 31](#_Toc165630247)

[5.2. Hướng phát triển 31](#_Toc165630248)

[PHỤ LỤC 33](#_Toc165630249)

[TÀI LIỆU THAM KHẢO 34](#_Toc165630250)

**BẢN PHÂN CHIA CÔNG VIỆC**

|  |  |  |
| --- | --- | --- |
| **Thành viên** | **Khối lượng công việc** | **Các phần được phân công** |
| Nguyễn Lê Gia Mỹ | 50% | Giao diện và những phần còn lại. |
| Đinh Kim Yến Nhi | 50% | Báo cáo và những phần còn lại. |

(\*) Trên đây là bảng phân chia công việc, trong quá trình làm bài, mọi thành viên  
đều hỗ trợ cho nhau để bài tập lớn được hoàn thiện nhất có thể nên số liệu chỉ minh  
họa không chính xác hoàn toàn.

# LỜI CẢM ƠN

Kính thưa Thầy,

Em xin gửi lời cảm ơn chân thành đến Thầy về việc hướng dẫn và hỗ trợ em trong quá trình thực hiện báo cáo bài tập lớn trong môn công nghệ Java.

Nhờ có sự chỉ dẫn từ Thầy, em đã có cơ hội học hỏi và áp dụng những kiến thức và kỹ năng mới trong việc phát triển dự án này. Những lời khuyên và góp ý của Thầy đã giúp em nắm vững kiến thức, cũng như phát triển khả năng giải quyết vấn đề và sáng tạo trong quá trình làm việc.

Em rất biết ơn sự kiên nhẫn và sự nhiệt tình của Thầy trong việc truyền đạt kiến thức và hỗ trợ em vượt qua các khó khăn trong quá trình học tập và thực hành. Sự động viên và khích lệ từ Thầy là động lực lớn giúp em hoàn thành dự án một cách thành công.

Một lần nữa, em xin chân thành cảm ơn Thầy về sự hỗ trợ và sự cống hiến của mình. Mong rằng em sẽ tiếp tục nhận được sự hướng dẫn và cố vấn từ Thầy trong những bước tiếp theo của hành trình học tập và phát triển sự nghiệp.

Trân trọng,

Nguyễn Lê Gia Mỹ

# DANH MỤC CHỮ VIẾT TẮT

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Mô tả** | **Ý nghĩa** | **Ghi chú** |
| 1 | CSDL | Cơ sở dữ liệu |  |
| 2 | MVC | Model-View-Controller |  |
| 3 | AWT | Abstract Window Toolkit |  |
| 4 | GUI | Graphical User Interface |  |
| 5 | API | Application Programming Interface |  |

# DANH MỤC HÌNH ẢNH

Hình 1: Biểu đồ use case tổng quát................................................................................................4

Hình 2: Biểu đồ lớp........................................................................................................................7

Hình 3: Lược đồ cơ sở dữ liệu.......................................................................................................8

Hình 4: Mô hình MVC.................................................................................................................10

Hình 5: Màn hình chính...............................................................................................................11

Hình 6: Màn hình quản lý độc giả................................................................................................13

Hình 7: Màn hình quản lý sách....................................................................................................18

Hình 8: Màn hình quản lý mượn trả.............................................................................................26

Hình 9: Màn hình thống kê – báo cáo..........................................................................................31

# CHƯƠNG 1. GIỚI THIỆU

## Tổng quan về đề tài

Mô tả đề tài: Quản lí các thông tin về sách, độc giả, hoạt động mượn trả.

Đối tượng sử dụng: Cán bộ quản lý thư viện (thủ thư).

Các chức năng chính: cập nhật thông tin, tìm kiếm thông tin, thống kê, báo cáo.

Các đối tượng dữ liệu:

+ Sách: Mã sách, tên sách, nhà xuất bản, tác giả, thể loại, đơn giá, số lượng.

+ Độc giả: Mã đọc giả, tên độc giả, hạn sử dụng, ngày sinh, địa chỉ, số điện thoại, giới tính.

+ Mượn trả: Mã mượn, mã độc giả mượn, tên độc giả mượn, ngày mượn, hạn trả, ngày trả, trạng thái, tiền phạt.

+ Chi tiết mượn trả: Mã mượn, mã sách mượn.

Công nghệ và công cụ: Eclipse, SQL Server, IDE, Java Swing,…

## Lý do chọn đề tài

Tính thực tiễn: Thư viện là một phần không thể thiếu của nhiều cộng đồng, tổ chức và trường học. Việc phát triển một ứng dụng quản lý thư viện không chỉ mang lại giá trị thực tiễn mà còn giúp cải thiện quản lý tài nguyên và dịch vụ thư viện.

Ứng dụng thực hành: Phát triển một ứng dụng quản lý thư viện đòi hỏi kỹ năng thiết kế, lập trình và quản lý dự án. Điều này cung cấp cho sinh viên cơ hội áp dụng kiến thức học được trong lớp học vào một dự án thực tế.

Học hỏi kỹ thuật: Quản lý thư viện đòi hỏi một loạt các chức năng như quản lý sách, quản lý độc giả, mượn/trả sách, tìm kiếm sách và thống kê. Phát triển một ứng dụng quản lý thư viện cung cấp cơ hội để nắm vững các kỹ thuật lập trình phức tạp như xử lý cơ sở dữ liệu, giao diện người dùng và xử lý sự kiện.

Tính đa dạng của chức năng: Dự án quản lý thư viện cung cấp nhiều khía cạnh khác nhau cho sinh viên khám phá, từ giao diện người dùng đến lưu trữ dữ liệu và xử lý logic. Điều này có thể giúp họ phát triển kỹ năng đa năng và tự tin trong việc làm việc với các công nghệ khác nhau.

Khả năng mở rộng: Một ứng dụng quản lý thư viện có thể được mở rộng và cải thiện theo thời gian, tùy thuộc vào nhu cầu và yêu cầu của người dùng. Điều này tạo ra một cơ hội để sinh viên tiếp tục phát triển và cải thiện dự án sau khi hoàn thành.

Tóm lại, việc chọn đề tài "Quản lý thư viện" cho báo cáo mang lại nhiều lợi ích học thuật và thực tiễn cho sinh viên, đồng thời cung cấp một cơ hội để áp dụng kiến thức và kỹ năng lập trình vào một dự án thực tế.

## Mục tiêu và phạm vi của đề tài

Mục tiêu:

Phát triển một ứng dụng phần mềm hoàn chỉnh để quản lý các hoạt động trong một thư viện.

Cung cấp một giao diện người dùng thân thiện và dễ sử dụng để cho phép người quản trị thư viện thực hiện các nhiệm vụ quản lý.

Tối ưu hóa hiệu suất và độ tin cậy của hệ thống để đáp ứng được nhu cầu thực tế của một thư viện.

Phạm vi:

Quản lý sách: Cho phép thêm, sửa đổi, xóa sách từ cơ sở dữ liệu. Mỗi sách sẽ có thông tin như tên, tác giả, số lượng tồn kho, vị trí.

Quản lý độc giả: Cho phép thêm, sửa đổi, xóa thông tin độc giả từ cơ sở dữ liệu. Mỗi độc giả có thông tin như tên, địa chỉ, số điện thoại.

Tìm kiếm sách: Cho phép người dùng tìm kiếm sách theo tiêu chí như tên sách, tác giả, thể loại.

Mượn/trả sách: Hỗ trợ quản lý quá trình mượn và trả sách, bao gồm việc ghi lại thông tin về ngày mượn, ngày trả và thông tin độc giả mượn sách.

Thống kê và báo cáo: Cung cấp các báo cáo về số lượng sách còn lại, sách đã mượn, sách quá hạn, độc giả mượn sách, v.v.

Tính linh hoạt và mở rộng:

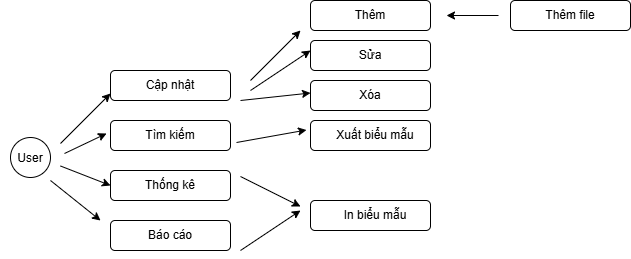
Thiết kế hệ thống sao cho có thể dễ dàng mở rộng và thêm tính năng mới trong tương lai.

Hỗ trợ cơ chế mở rộng giao diện và logic xử lý để dễ dàng tích hợp các tính năng mới mà không làm ảnh hưởng đến hiệu suất và tính ổn định của hệ thống.

Với mục tiêu và phạm vi này, dự án sẽ tập trung vào việc xây dựng một ứng dụng quản lý thư viện toàn diện, đồng thời đảm bảo tính linh hoạt và mở rộng để có thể đáp ứng được các yêu cầu và nhu cầu mới trong tương lai.

# CHƯƠNG 2: BÀI TOÁN

## 2.1. Giới thiệu bài toán



Hình 1: Biểu đồ use case tổng quát

- Trình tự thêm:

+ User chọn chức năng thêm

+ Hệ thống yêu cầu nhập thông tin

+ User nhập thông tin

+ Nếu thông tin đúng, hệ thống sẽ lưu thông tin vừa nhập vào cơ sở dữ liệu và xử lý

+ Nếu thông tin sai, hệ thống yêu cầu nhập lại.

- Trình tự xóa:

+ User chọn đối tượng cần xóa từ danh sách hiện có bằng cách click vào hoặc nhập mã.

+ Chọn xóa

+ Hệ thống hiển thị thông báo xác nhận trước khi xóa để tránh xóa nhầm.

+ Sau khi xác nhận, hệ thống xóa khỏi cơ sở dữ liệu.

- Trình tự sửa:

+ User chọn cần chỉnh sửa từ danh sách hiện có bằng cách click vào.

+ Hệ thống hiển thị thông tin chi tiết để người quản trị chỉnh sửa.

+ Sau khi chỉnh sửa, hệ thống cập nhật thông tin mới vào cơ sở dữ liệu.

- Quy trình mượn sách:

+ User nhập thông tin độc giả và sách cần mượn.

+ Hệ thống kiểm tra tình trạng sẵn có của sách và thông tin độc giả.

+ Nếu hợp lệ, hệ thống ghi nhận thông tin về ngày mượn, ngày dự kiến trả, và cập nhật số lượng sách trong kho.

+ Hệ thống lưu trữ giao dịch mượn sách vào cơ sở dữ liệu.

- Quy trình trả sách:

+ User nhập thông tin độc giả và sách cần trả.

+ Hệ thống kiểm tra thông tin và ghi nhận ngày trả sách.

+ Hệ thống cập nhật lại số lượng sách trong kho và tình trạng mượn của độc giả.

+ Nếu có sách quá hạn, hệ thống sẽ tính phí phạt theo từng ngày.

- Báo cáo và thống kê

+ Thống kê sách:

Hệ thống tạo báo cáo về số lượng sách hiện có, sách đã mượn và xuất.

+ Thống kê độc giả:

Hệ thống tạo báo cáo về số lượng độc giả, số lượng mượn sách của từng độc giả, và các độc giả có sách quá hạn và xuất.

+ Báo cáo mượn/trả:

Hệ thống tổng hợp và báo cáo chi tiết về hoạt động mượn và trả sách trong khoảng thời gian xác định và xuất.

## 2.2. Các chức năng của bài toán

- Quản lý sách:

Thêm sách mới: Cho phép user thêm sách mới vào hệ thống. User có thể nhập các thông tin như tên sách, tác giả, nhà xuất bản, thể loại, và số lượng sách.

Sửa thông tin sách: Cho phép user chỉnh sửa thông tin của sách đã có trong hệ thống. Điều này bao gồm cập nhật thông tin về sách như thay đổi tên, tác giả, hoặc số lượng sách hiện có.

Xóa sách: Cho phép user xóa sách khỏi hệ thống. Việc này bao gồm cả việc kiểm tra và xác nhận trước khi xóa để tránh xóa nhầm dữ liệu quan trọng.

Tìm kiếm sách: Cho phép user tìm kiếm sách trong hệ thống dựa trên các tiêu chí như tên sách, tác giả, thể loại, hoặc nhà xuất bản.

- Quản lý độc giả:

Thêm độc giả mới: Cho phép user thêm độc giả mới vào hệ thống với các thông tin như tên, địa chỉ, số điện thoại, và mã độc giả.

Sửa thông tin độc giả: Cho phép user cập nhật thông tin của độc giả đã đăng ký trong hệ thống.

Xóa độc giả: Cho phép user xóa độc giả khỏi hệ thống, bao gồm cả việc kiểm tra và xác nhận trước khi xóa.

Tìm kiếm độc giả: Cho phép user tìm kiếm độc giả dựa trên các tiêu chí như tên, mã độc giả, hoặc số điện thoại.

- Quản lý mượn/trả sách:

Quy trình mượn sách:

Kiểm tra thông tin sách và độc giả trước khi mượn.

Ghi nhận thông tin về ngày mượn, ngày trả, và cập nhật số lượng sách trong kho.

Quy trình trả sách:

Ghi nhận ngày trả sách.

Cập nhật lại số lượng sách trong kho và tình trạng mượn của độc giả.

Quản lý quá hạn: Theo dõi các sách đã quá hạn và tính tiền phạt theo số ngày quá hạn.

- Báo cáo và thống kê

Thống kê sách: Cung cấp các báo cáo về số lượng sách hiện có, sách đã mượn.

Thống kê độc giả: Cung cấp các báo cáo về số lượng độc giả, số lượng mượn sách của độc giả, và các độc giả có sách quá hạn.

Báo cáo mượn/trả: Tổng hợp và báo cáo chi tiết về hoạt động mượn và trả sách.

- Giao diện người dùng

Trang chủ: Cung cấp thông tin tổng quan và các chức năng chính của hệ thống.

Giao diện quản lý sách: Cung cấp các chức năng liên quan đến quản lý sách như thêm, sửa, xóa, tìm kiếm sách, xuất file, thêm file.

Giao diện quản lý độc giả: Cung cấp các chức năng liên quan đến quản lý độc giả như thêm, sửa, xóa, và tìm kiếm độc giả.

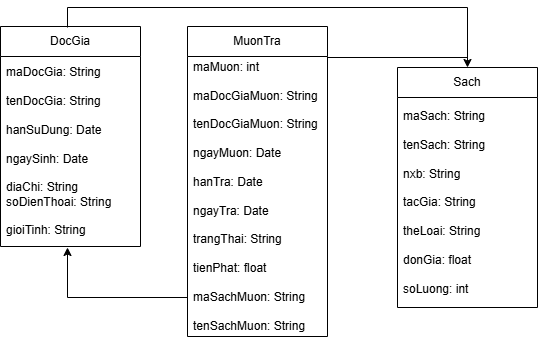
Giao diện mượn/trả sách: Hỗ trợ các quy trình mượn và trả sách, bao gồm cả việc quản lý quá hạn.

Giao diện báo cáo: Hiển thị các báo cáo và thống kê chi tiết về sách và độc giả.

## 2.3. Class diagram

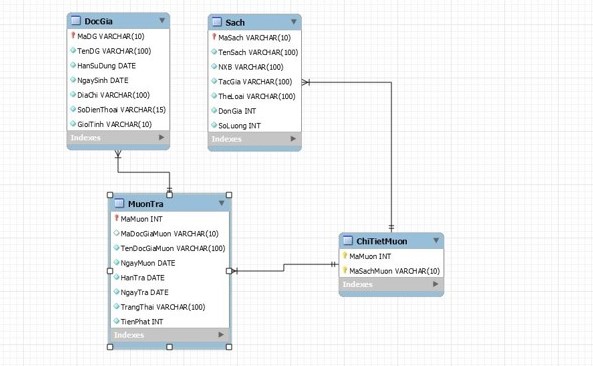
Class diagram mô tả kiểu của các đối tượng trong hệ thống và các loại quan hệ khác nhau tồn tại giữa chúng.

Là một kỹ thuật mô hình hóa tồn tại ở tất cả các phương pháp phát triển hướng đối tượng.



Hình 2: Biểu đồ lớp

## 2.4. Lược đồ cơ sở dữ liệu



Hình 3: Lược đồ cơ sở dữ liệu

## 2.5. Các công nghệ/API sử dụng

Ngôn ngữ lập trình: Java

Hệ QTCSDL: SQL Server

Thiết kế giao diện: Java Swing, AWT

Gói thư viện: jxl.jar, sqljdbc4.jar, rs2xml.jar, jcalendar.jar, iText.jar

Thư viện cung cấp các hàm cho phép đọc file .xls từ ứng dụng Java.

sqljdbc4.jar: kết nối cơ sở dữ liệu với hệ QTCSDL SQL Server

iText.jar : ghi file .pdf

Jcalendar.jar : hỗ trợ chọn dữ liệu liên quan đến ngày tháng

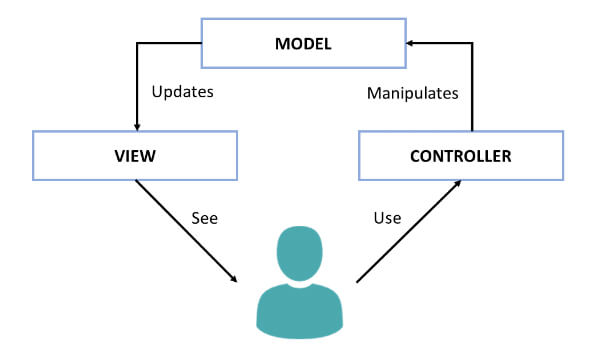
# CHƯƠNG 3. KẾT QUẢ

## 3.1. Tổng quan

- User có những cửa sổ để nhìn và điều khiển thông qua chuột, phím gọi là view và controller.

- Khi đó, controller tác động đến model – là những class bên dưới.

- Sau khi xử lý xong, hệ thống hiển thị ra view. Lúc này, view sẽ bị thay đổi.



Hình 4: Mô hình MVC

## 3.2. Màn hình chính



Hình 5: Màn hình chính

* Khai báo các biến thành phần GUI:

jPanel1: Một JPanel chứa tiêu đề của giao diện.

jPanel3: Một JPanel chứa các nút chức năng chính.

panelChinh: Một JPanel hiển thị nội dung chính tương ứng với các nút chức năng.

btnQLDG, btnQLS, btnQLMT, btnTKBC: Các nút bấm để chuyển đổi giữa các chức năng quản lý.

* Constructor GUIMain:

Thiết lập một số thuộc tính của JFrame như biểu tượng (setIconImage), không thay đổi kích thước (setResizable), và đóng ứng dụng khi nhấn nút thoát (setDefaultCloseOperation).

Khởi tạo các thành phần GUI con như GUIQuanLyDocGia, GUIQuanLySach, GUIQuanLyMuonTra, và GUIThongKeBaoCao.

* Phương thức initComponents:

Thiết lập bố cục cho các thành phần GUI, bao gồm jPanel1, panelChinh, và jPanel3.

Thiết lập các thuộc tính như màu nền, phông chữ, và biểu tượng cho các nhãn và nút.

Thêm các thành phần vào JFrame sử dụng GroupLayout và BoxLayout.

* Các phương thức xử lý sự kiện cho các nút:

btnQLDGActionPerformed: Xử lý khi người dùng nhấn nút "Quản lý độc giả". Nó sẽ cập nhật panelChinh để hiển thị giao diện quản lý độc giả.

btnQLSActionPerformed: Tương tự, xử lý khi nhấn nút "Quản lý sách".

btnQLMTActionPerformed: Xử lý khi nhấn nút "Quản lý mượn trả".

btnTKBCActionPerformed: Xử lý khi nhấn nút "Thống kê - báo cáo".

* Phương thức main:

Thiết lập giao diện "Nimbus" (nếu có sẵn).

Tạo và hiển thị cửa sổ GUIMain.

## 3.3. Màn hình quản lý độc giả



Hình 6: Màn hình quản lý độc giả

* View:

Imports: Mã nhập các lớp và thư viện Java cần thiết cho các hoạt động cơ sở dữ liệu, thành phần giao diện người dùng (UI), ghi nhật ký, xử lý tệp, v.v.

Biến và Đối tượng:

DefaultTableModel dm: Sử dụng để quản lý dữ liệu trong một bảng.

CDocGia cdocGia: Một thể hiện của một lớp chịu trách nhiệm quản lý các thành viên của thư viện.

Connection con, Statement st, ResultSet rs, PreparedStatement ps: Các đối tượng liên quan đến cơ sở dữ liệu.

SimpleDateFormat sdf: Được sử dụng để định dạng ngày.

List<DocGia> docGiaList: Một danh sách để chứa các đối tượng thành viên của thư viện.

Các thành phần GUI:

Các thành phần Swing khác nhau như nhãn, trường văn bản, nút, nút radio, hộp combo, lựa chọn ngày và bảng được khởi tạo và được sắp xếp bằng GroupLayout.

Constructor: Khởi tạo các thành phần giao diện người dùng, thiết lập kết nối cơ sở dữ liệu, tạo một đối tượng lệnh, và tải dữ liệu vào bảng.

Bộ lắng nghe Sự kiện:

Lắng nghe hành động cho các nút như Làm mới, Xóa, Chỉnh sửa, Thêm, Tìm kiếm và Xuất ra PDF.

Lắng nghe hành động nút radio để chọn giới tính.

Lắng nghe sự kiện cho tìm kiếm.

Phương thức:

loadDataInTable(): Lấy dữ liệu từ cơ sở dữ liệu và điền vào bảng.

Các phương thức khác để xử lý các hoạt động cơ sở dữ liệu như chèn, xóa, cập nhật, v.v.

Comments: Một số bình luận được cung cấp để giải thích một số phần của mã hoặc để nhắc nhở nhà phát triển về các nhiệm vụ cần thực hiện.

jPanel2Layout: Định nghĩa cách sắp xếp các thành phần trong jPanel2, bao gồm nhãn, trường văn bản, nút radio, và các nút chức năng như Thêm, Sửa, Xóa, Làm mới.

jPanel3Layout: Xác định cách sắp xếp các thành phần trong jPanel3, bao gồm trường tìm kiếm, combo box để chọn loại tìm kiếm, nút tìm kiếm và nút xuất PDF.

jPanel4Layout: Xác định cách sắp xếp jScrollPane1, là nơi hiển thị danh sách độc giả trong jPanel4.

Trong đó:

Các thành phần như jLabel, jTextField, JDateChooser, JRadioButton, JButton đều được đặt vào jPanel2 và jPanel3.

jScrollPane1 chứa tbDocGia, là một bảng hiển thị danh sách độc giả.

Mỗi layout (jPanel2Layout, jPanel3Layout, jPanel4Layout) định nghĩa cách các thành phần sẽ được sắp xếp, điều này giúp tổ chức giao diện một cách dễ dàng và linh hoạt.

loadDataInTable(): Hàm này tải dữ liệu từ cơ sở dữ liệu và hiển thị vào bảng. Nó tạo một DefaultTableModel mới, thêm các cột vào và sau đó lấy dữ liệu từ danh sách docGiaList và thêm vào bảng.

loadDateSearch(): Hàm này thực hiện tìm kiếm dữ liệu dựa trên thông tin được nhập từ trường tìm kiếm. Nó xây dựng một DefaultTableModel mới sau đó thực hiện tìm kiếm dữ liệu dựa trên các tiêu chí được chọn từ combo box cbbTimKiem.

clear(): Hàm này xóa dữ liệu trong các trường văn bản sau khi thêm, sửa hoặc xóa.

themFile1(): Hàm này cho phép người dùng chọn một tệp Excel từ hộp thoại JFileChooser, sau đó nó đọc dữ liệu từ tệp Excel và thêm dữ liệu vào cơ sở dữ liệu. Dữ liệu cũng được thêm vào DefaultTableModel để hiển thị trên bảng.

Trong đó:

Hàm DBConnect.insert() được gọi để thêm dữ liệu vào cơ sở dữ liệu từ mỗi dòng trong tệp Excel.

Dữ liệu từ mỗi dòng trong tệp Excel được lấy ra từ sheet.getCell() và sau đó được thêm vào DefaultTableModel dm.

writeFileExcel(): Hàm này tạo một tệp Excel mới và ghi dữ liệu từ bảng tbDocGia vào đó. Dữ liệu được lấy từ DefaultTableModel của bảng và được ghi vào tệp Excel theo định dạng cột.

btnRefreshActionPerformed(): Hàm này được gọi khi người dùng nhấn nút "Làm mới". Nó gọi hàm loadDataInTable() để làm mới dữ liệu trong bảng.

btnTimKiemActionPerformed(): Hàm này được gọi khi người dùng nhấn nút "Tìm kiếm". Nó gọi hàm loadDateSearch() để thực hiện tìm kiếm dựa trên thông tin được nhập.

btnXuatPDFActionPerformed(): Hàm này được gọi khi người dùng nhấn nút "Xuất PDF". Nó tạo một tệp PDF chứa dữ liệu tìm kiếm từ bảng và ghi vào tệp PDF.

tbDocGiaMouseClicked(): Hàm này được gọi khi người dùng nhấn chuột vào bảng tbDocGia. Nó lấy dữ liệu từ hàng được chọn trong bảng và hiển thị lên các trường văn bản và lựa chọn ngày.

jScrollPane1MouseClicked(): Hàm này được gọi khi người dùng nhấn chuột vào jScrollPane1, không có hành động cụ thể được thực hiện trong hàm này.

btnSuaActionPerformed(): Hàm này được gọi khi người dùng nhấn nút "Sửa". Nó kiểm tra các trường văn bản và lựa chọn ngày để đảm bảo rằng dữ liệu đã nhập đầy đủ, sau đó cập nhật thông tin của độc giả được chọn trong bảng.

btnThemActionPerformed(): Hàm này được gọi khi người dùng nhấn nút "Thêm". Nó kiểm tra các trường văn bản và lựa chọn ngày để đảm bảo rằng dữ liệu đã nhập đầy đủ, sau đó thêm một độc giả mới vào cơ sở dữ liệu.

btnXoaActionPerformed(): Hàm này được gọi khi người dùng nhấn nút "Xóa". Nó xóa thông tin của độc giả được chọn trong bảng khỏi cơ sở dữ liệu sau khi hiển thị một hộp thoại xác nhận.

* Controller:

+ Constructor:

Khởi tạo đối tượng CDocGia.

Thiết lập kết nối đến cơ sở dữ liệu thông qua lớp DBConnect.

Khởi tạo một Statement để thực thi các truy vấn SQL.

+ selectAll():

Truy vấn tất cả các dòng từ bảng DocGia.

Chuyển kết quả của truy vấn thành danh sách các đối tượng DocGia.

Mỗi dòng từ kết quả truy vấn được chuyển thành một đối tượng DocGia và được thêm vào danh sách.

+ search(DocGia docGia, String columnName):

Tìm kiếm các đối tượng DocGia dựa trên các thông tin được cung cấp trong đối tượng docGia và tên cột (columnName) tương ứng.

Xây dựng câu truy vấn SQL dựa trên thông tin tìm kiếm được cung cấp.

Thực thi truy vấn và trả về danh sách các đối tượng DocGia thỏa mãn điều kiện tìm kiếm.

+ insert(DocGia docGia):

Thêm một đối tượng DocGia mới vào bảng DocGia.

Sử dụng một câu truy vấn INSERT để thêm dữ liệu mới.

Chuyển đổi ngày từ Java sang định dạng ngày trong SQL trước khi thêm vào cơ sở dữ liệu.

+ update(DocGia docGia):

Cập nhật thông tin của một đối tượng DocGia đã tồn tại trong bảng DocGia.

Sử dụng một câu truy vấn UPDATE để cập nhật dữ liệu.

Tương tự như insert(), các giá trị ngày được chuyển đổi từ Java sang định dạng ngày trong SQL trước khi cập nhật.

+ delete(DocGia docGia):

Xóa một đối tượng DocGia khỏi bảng DocGia.

Sử dụng một câu truy vấn DELETE để xóa dữ liệu.

Đảm bảo rằng kết nối đến cơ sở dữ liệu được đóng sau khi thực hiện xong thao tác xóa.

Code đã được cấu trúc để đảm bảo thực thi các truy vấn SQL một cách an toàn và đáng tin cậy, đồng thời đảm bảo việc xử lý kết nối đến cơ sở dữ liệu sau mỗi thao tác.

## 3.4. Màn hình quản lý sách



Hình 7: Màn hình quản lý sách

* View:

Constructor (GUIQuanLySach()):

Khởi tạo các thành phần giao diện bằng cách gọi initComponents().

Thiết lập kết nối đến cơ sở dữ liệu bằng cách sử dụng DBConnect.getConnection().

Tạo một đối tượng Statement để thực hiện các câu lệnh SQL.

Gọi phương thức loadDataInTable() để tải dữ liệu sách vào bảng.

initComponents():

Phương thức này khởi tạo tất cả các thành phần giao diện như nhãn, ô nhập liệu, nút bấm, panel và bảng.

Nó cũng thiết lập bố cục và thêm các bộ lắng nghe sự kiện (action listeners) cho các nút bấm.

btnThemActionPerformed(java.awt.event.ActionEvent evt):

Phương thức này được gọi khi nút "Thêm" được nhấn.

Nó xử lý logic để thêm một bản ghi sách mới vào cơ sở dữ liệu.

btnXoaActionPerformed(java.awt.event.ActionEvent evt):

Phương thức này được gọi khi nút "Xóa" được nhấn.

Nó xử lý logic để xóa một bản ghi sách đã chọn từ cơ sở dữ liệu.

btnSuaActionPerformed(java.awt.event.ActionEvent evt):

Phương thức này được gọi khi nút "Sửa" được nhấn.

Nó xử lý logic để cập nhật một bản ghi sách đã chọn trong cơ sở dữ liệu.

btnRefreshActionPerformed(java.awt.event.ActionEvent evt):

Phương thức này được gọi khi nút "Refresh" được nhấn.

Nó làm mới form bằng cách xóa các ô nhập liệu và tải lại dữ liệu bảng.

btnThemTuFileActionPerformed(java.awt.event.ActionEvent evt):

Phương thức này được gọi khi nút "Thêm file" được nhấn.

Nó xử lý logic để thêm các bản ghi sách từ một file Excel.

btnTimKiemActionPerformed(java.awt.event.ActionEvent evt):

Phương thức này được gọi khi nút "Tìm kiếm" được nhấn.

Nó xử lý logic để tìm kiếm các bản ghi sách dựa trên tiêu chí và từ khóa đã chọn.

XuatPDFActionPerformed(java.awt.event.ActionEvent evt):

Phương thức này được gọi khi nút "Xuất pdf" được nhấn.

Nó xử lý logic để xuất các bản ghi sách ra file PDF.

tbSachMouseClicked(java.awt.event.MouseEvent evt):

Phương thức này được gọi khi một dòng trong bảng (tbSach) được nhấn.

Nó xử lý logic để điền dữ liệu từ dòng đã chọn vào các ô nhập liệu.

Các phương thức tiện ích (không được hiển thị đầy đủ trong mã nguồn nhưng được tham chiếu):

loadDataInTable():

Phương thức này tải dữ liệu sách từ cơ sở dữ liệu và hiển thị chúng trong bảng.

refreshForm():

Phương thức này làm mới form, xóa sạch các ô nhập liệu và thiết lập lại trạng thái ban đầu.

searchBook():

Phương thức này thực hiện tìm kiếm sách dựa trên từ khóa và tiêu chí đã chọn.

exportToPDF():

Phương thức này xuất dữ liệu sách ra file PDF.

importFromExcel():

Phương thức này đọc dữ liệu sách từ file Excel và thêm chúng vào cơ sở dữ liệu.

loadDataInTable()

Phương thức này dùng để tải dữ liệu từ cơ sở dữ liệu vào JTable.

Giải thích từng bước:

Tạo một DefaultTableModel mới.

Định nghĩa các cột cho bảng và thêm vào mô hình (dm).

Sử dụng đối tượng CSach (cSach) để lấy danh sách các cuốn sách từ cơ sở dữ liệu bằng cách gọi phương thức selectAll().

Với mỗi cuốn sách (Sach) trong danh sách, tạo một Vector mới và điền các thuộc tính của cuốn sách vào đó.

Mỗi Vector được thêm vào mô hình dữ liệu (dm).

Cuối cùng, thiết lập mô hình dữ liệu cho JTable (tbSach) để hiển thị các dữ liệu đã tải.

loadDateSearch()

Phương thức này dùng để tìm kiếm dữ liệu dựa trên tiêu chí tìm kiếm và hiển thị kết quả lên JTable.

Giải thích từng bước:

Tạo một DefaultTableModel mới và xóa tất cả các cột và dòng cũ.

Định nghĩa các cột cho bảng và thêm vào mô hình (dm).

Tạo đối tượng CSach để lấy dữ liệu từ cơ sở dữ liệu.

Khởi tạo một đối tượng Sach và thiết lập các thuộc tính của nó bằng giá trị tìm kiếm từ tfTimKiem.

Dựa trên tiêu chí tìm kiếm được chọn từ cbbTimKiem, gọi phương thức search() của cSach để lấy danh sách sách phù hợp.

Với mỗi cuốn sách trong danh sách kết quả, tạo một Vector mới và điền các thuộc tính của cuốn sách vào đó.

Mỗi Vector được thêm vào mô hình dữ liệu (dm).

Cuối cùng, thiết lập mô hình dữ liệu cho JTable (tbSach) để hiển thị kết quả tìm kiếm.

clear()

Phương thức này dùng để xóa nội dung trong các trường nhập liệu.

Giải thích từng bước:

Đặt nội dung của tất cả các trường nhập liệu (tfMaSach, tfTenSach, tfNhaXuatBan, tfTacGia, tfTheLoai, tfDonGia, tfSoLuong) thành chuỗi rỗng.

themFile()

Phương thức này dùng để thêm dữ liệu từ một tập tin vào cơ sở dữ liệu và JTable.

Giải thích từng bước:

Mở một hộp thoại chọn tập tin (JFileChooser) và chờ người dùng chọn tập tin.

Nếu người dùng chọn một tập tin, đọc nội dung của tập tin đó.

Tạo một DefaultTableModel mới và định nghĩa các cột.

Đọc dữ liệu từ tập tin Excel bằng thư viện jxl và lưu vào cơ sở dữ liệu.

Thêm dữ liệu từ tập tin vào mô hình dữ liệu (dm).

Thiết lập mô hình dữ liệu cho JTable (tbSach) để hiển thị dữ liệu.

btnXoaActionPerformed(java.awt.event.ActionEvent evt)

Phương thức này xử lý sự kiện khi người dùng nhấn nút "Xóa".

Giải thích từng bước:

Hiển thị một hộp thoại xác nhận xóa.

Nếu người dùng chọn "Yes", kiểm tra xem cuốn sách có đang được mượn hay không.

Nếu sách không được mượn, xóa sách khỏi cơ sở dữ liệu.

Làm mới dữ liệu trên bảng và xóa nội dung các trường nhập liệu.

btnThemActionPerformed(java.awt.event.ActionEvent evt)

Phương thức này xử lý sự kiện khi người dùng nhấn nút "Thêm".

Giải thích từng bước:

Hiển thị hộp thoại xác nhận thêm dữ liệu.

Kiểm tra các trường nhập liệu xem có bị bỏ trống hay không.

Kiểm tra xem mã sách có tồn tại trong cơ sở dữ liệu hay không.

Nếu mã sách không tồn tại, thêm sách mới vào cơ sở dữ liệu.

Làm mới dữ liệu trên bảng.

btnSuaActionPerformed(java.awt.event.ActionEvent evt)

Phương thức này xử lý sự kiện khi người dùng nhấn nút "Sửa".

Giải thích từng bước:

Hiển thị hộp thoại xác nhận sửa dữ liệu.

Kiểm tra trường mã sách xem có bị bỏ trống hay không.

Tạo đối tượng sách mới với các thông tin đã sửa và cập nhật vào cơ sở dữ liệu.

Làm mới dữ liệu trên bảng.

btnRefreshActionPerformed(java.awt.event.ActionEvent evt)

Phương thức này xử lý sự kiện khi người dùng nhấn nút "Làm mới".

Giải thích từng bước:

Gọi phương thức loadDataInTable() để làm mới dữ liệu trên bảng.

btnThemTuFileActionPerformed(java.awt.event.ActionEvent evt)

Phương thức này xử lý sự kiện khi người dùng nhấn nút "Thêm từ file".

Giải thích từng bước:

Gọi phương thức themFile() để thêm dữ liệu từ file vào cơ sở dữ liệu và JTable.

tbSachMouseClicked(java.awt.event.MouseEvent evt)

Phương thức này xử lý sự kiện khi người dùng nhấn vào một hàng trong JTable.

Giải thích từng bước:

Lấy chỉ số của hàng được chọn.

Lấy dữ liệu của hàng được chọn và điền vào các trường nhập liệu tương ứng.

12. btnTimKiemActionPerformed(java.awt.event.ActionEvent evt)

Phương thức này xử lý sự kiện khi người dùng nhấn nút "Tìm kiếm".

Giải thích từng bước:

Gọi phương thức loadDateSearch() để tìm kiếm dữ liệu và hiển thị kết quả trên bảng.

XuatPDFActionPerformed(java.awt.event.ActionEvent evt)

Phương thức này xử lý sự kiện khi người dùng nhấn nút "Xuất PDF".

Giải thích từng bước:

Tạo tiêu đề và dữ liệu cho báo cáo PDF từ kết quả tìm kiếm.

Sử dụng lớp PrintPDF để xuất báo cáo PDF và lưu vào file do người dùng chọn.

* Controller:

khởi tạo (CSach() Constructor)

Phương thức khởi tạo dùng để thiết lập kết nối với cơ sở dữ liệu và khởi tạo đối tượng Statement. Sử dụng DBConnect.getConnection() để lấy kết nối tới cơ sở dữ liệu. Tạo đối tượng Statement từ kết nối để thực hiện các câu lệnh SQL.

lấy tất cả sách (selectAll())

Phương thức này dùng để lấy toàn bộ danh sách sách từ cơ sở dữ liệu. Kết nối tới cơ sở dữ liệu và thực hiện câu lệnh SQL SELECT \* FROM Sach. Tạo danh sách các đối tượng Sach và thêm các sách vào danh sách này. Trả về danh sách sách.

tìm kiếm sách (search(Sach sach, String columnName))

Phương thức này dùng để tìm kiếm sách dựa trên thuộc tính của đối tượng Sach và tên cột được chỉ định. Xây dựng câu lệnh SQL dựa trên giá trị của columnName. Thực hiện câu lệnh SQL và lấy kết quả. Tạo danh sách các đối tượng Sach từ kết quả truy vấn và trả về danh sách này.

thêm sách (insert(Sach sach))

Phương thức này dùng để thêm một sách mới vào cơ sở dữ liệu. Chuẩn bị câu lệnh SQL INSERT INTO Sach. Thiết lập các tham số của câu lệnh SQL từ đối tượng Sach. Thực hiện câu lệnh SQL để thêm sách vào cơ sở dữ liệu.

cập nhật sách (update(Sach sach))

Phương thức này dùng để cập nhật thông tin của một sách đã tồn tại trong cơ sở dữ liệu. Chuẩn bị câu lệnh SQL UPDATE Sach. Thiết lập các tham số của câu lệnh SQL từ đối tượng Sach. Thực hiện câu lệnh SQL để cập nhật sách trong cơ sở dữ liệu.

xóa sách (delete(Sach sach))

Phương thức này dùng để xóa một sách khỏi cơ sở dữ liệu. Chuẩn bị câu lệnh SQL DELETE FROM Sach. Thiết lập tham số của câu lệnh SQL từ đối tượng Sach. Thực hiện câu lệnh SQL để xóa sách khỏi cơ sở dữ liệu. Đóng kết nối tới cơ sở dữ liệu trong khối finally.

## 3.5. Màn hình quản lý mượn trả



Hình 8: Màn hình quản lý mượn trả

- View:

Phương thức khởi tạo GUIQuanLyMuonTra()

Phương thức này được gọi khi một đối tượng GUIQuanLyMuonTra được tạo ra. Trong phương thức này, các thành phần giao diện người dùng được khởi tạo và cập nhật.

Phương thức updateBangMuon()

Phương thức này được sử dụng để cập nhật bảng hiển thị thông tin về các phiếu mượn sách trong cơ sở dữ liệu.

Phương thức updateChiTietMuon()

Phương thức này cập nhật bảng hiển thị thông tin chi tiết về việc mượn sách của mỗi độc giả.

Phương thức updateChiTietPhieuMuon()

Phương thức này cập nhật bảng hiển thị chi tiết về các phiếu mượn sách, bao gồm mã mượn, mã sách, tên sách, mã độc giả, ngày mượn và hạn trả.

Phương thức loadDLSearch()

Phương thức này được sử dụng để tải dữ liệu tìm kiếm từ cơ sở dữ liệu dựa trên mã mượn sách được nhập vào.

Phương thức PhieuMuon(String maMuon)

Phương thức này trả về một vector chứa thông tin chi tiết về phiếu mượn sách dựa trên mã mượn sách được nhập vào.

Sự kiện btnTimKiemSachActionPerformed()

Sự kiện này được kích hoạt khi nút "Tìm kiếm" cho sách được nhấn. Nó tìm kiếm sách trong cơ sở dữ liệu dựa trên mã sách được nhập vào.

Sự kiện btnTimKiemDocGiaActionPerformed()

Sự kiện này được kích hoạt khi nút "Tìm kiếm" cho độc giả được nhấn. Nó tìm kiếm độc giả trong cơ sở dữ liệu dựa trên mã độc giả được nhập vào.

Sự kiện btnMuonSachActionPerformed()

Sự kiện này được kích hoạt khi nút "Mượn sách" được nhấn. Nó thực hiện việc ghi thông tin mượn sách vào cơ sở dữ liệu.

Sự kiện btnLayMaMuonActionPerformed()

Sự kiện này được kích hoạt khi nút "Lấy mã mượn" được nhấn. Nó thực hiện việc tự động tạo mã mượn sách mới.

Thiết lập GroupLayout cho jPanel8:

Đoạn mã này khởi tạo và cấu hình GroupLayout cho jPanel8, một trình quản lý bố trí được sử dụng trong Java Swing để sắp xếp các thành phần. GroupLayout cho phép bạn tạo các bố cục linh hoạt và động.

Phương thức jPanel8Layout.setHorizontalGroup() chỉ định bố cục theo chiều ngang của các thành phần trong jPanel8. Nó xác định cách các thành phần được sắp xếp từ trái sang phải.

Phương thức jPanel8Layout.setVerticalGroup() chỉ định bố cục theo chiều dọc của các thành phần trong jPanel8. Nó xác định cách các thành phần được sắp xếp từ trên xuống dưới.

Căn chỉnh Thành phần:

Enumeration javax.swing.GroupLayout.Alignment được sử dụng để chỉ định cách các thành phần nên được căn chỉnh trong container của chúng. Ví dụ, javax.swing.GroupLayout.Alignment.LEADING căn chỉnh các thành phần vào cạnh dẫn đầu của container, trong khi javax.swing.GroupLayout.Alignment.TRAILING căn chỉnh các thành phần vào cạnh cuối của container.

Khởi tạo Thành phần:

Một số JLabels (ví dụ: jLabel4, jLabel5, ...) và biến JLabel (ví dụ: lbTenDocGia, lbMaDocGia, ...) được khởi tạo. JLabels được sử dụng để hiển thị văn bản hoặc hình ảnh.

Biến JButton (ví dụ: btnLayMaMuon, btnMuonSach) cũng được khởi tạo. JButton được sử dụng để tạo các nút có thể nhấp trong ứng dụng Swing.

Cấu hình Thành phần:

Văn bản, biểu tượng và trình nghe hành động được cấu hình cho một số nút bằng cách sử dụng các phương thức như setText(), setIcon(), và addActionListener().

Bố cục Thành phần:

Các thành phần được thêm vào jPanel8Layout bằng các phương thức như addGroup() và addComponent(). Các phương thức này chỉ định cách các thành phần nên được sắp xếp liên quan đến nhau.

Căn chỉnh và kích thước của thành phần được cấu hình bằng cách sử dụng các phương thức như setAlignment() và setSize().

Liên kết theo Chiều ngang và Chiều dọc:

Phương thức linkSize() được sử dụng để đảm bảo kích thước nhất quán cho một nhóm thành phần cả trong chiều ngang và chiều dọc.

Tab "Chi tiết phiếu mượn" (jTabbedPane1):

Tab này chứa một jPanel (jPanel18) để hiển thị chi tiết về phiếu mượn sách.

jPanel18 chứa jPanel19, một container để chứa các thành phần hiển thị thông tin về phiếu mượn và tìm kiếm thông tin.

jPanel19 có hai phần chính:

jPanel20: Chứa các thành phần để tìm kiếm thông tin phiếu mượn, bao gồm một nhãn (jLabel15) và một trường văn bản để nhập mã mượn (tfMaMuon2), cùng với một nút "Tìm kiếm" (btnTimKiem5).

jPanel21: Chứa một bảng (jtablehiTietPhieuMuon) để hiển thị chi tiết các phiếu mượn đã được tìm thấy. Bên dưới bảng là hai nút "Refresh" (btnRefresh3) và "In phiếu" (btnInPhieu) để làm mới dữ liệu và in phiếu mượn.

Xử lý sự kiện:

Có các phương thức xử lý sự kiện cho các nút:

btnInPhieuActionPerformed(): Xử lý sự kiện khi nút "In phiếu" được nhấp. Phương thức này tạo và in một phiếu mượn dưới dạng PDF.

btnRefresh3ActionPerformed(): Xử lý sự kiện khi nút "Refresh" được nhấp. Phương thức này cập nhật lại chi tiết phiếu mượn.

btnTimKiem5ActionPerformed(): Xử lý sự kiện khi nút "Tìm kiếm" được nhấp. Phương thức này thực hiện tìm kiếm thông tin phiếu mượn dựa trên mã mượn nhập vào.

Khởi tạo và cấu hình thành phần:

Đoạn mã cũng chứa các phương thức để khởi tạo và cấu hình các thành phần, như nhãn, trường văn bản, bảng và nút.

Các thành phần được đặt tên và được sử dụng để hiển thị thông tin và tương tác với người dùng.

- Controller:

Biến và Phương Thức Khởi Tạo: Class CMuonTra có một số biến instance để làm việc với cơ sở dữ liệu như Connection, Statement, ResultSet, PreparedStatement. Constructor của class này khởi tạo kết nối đến cơ sở dữ liệu và tạo một đối tượng Statement để thực thi các truy vấn SQL.

Phương Thức Xóa Dữ Liệu:

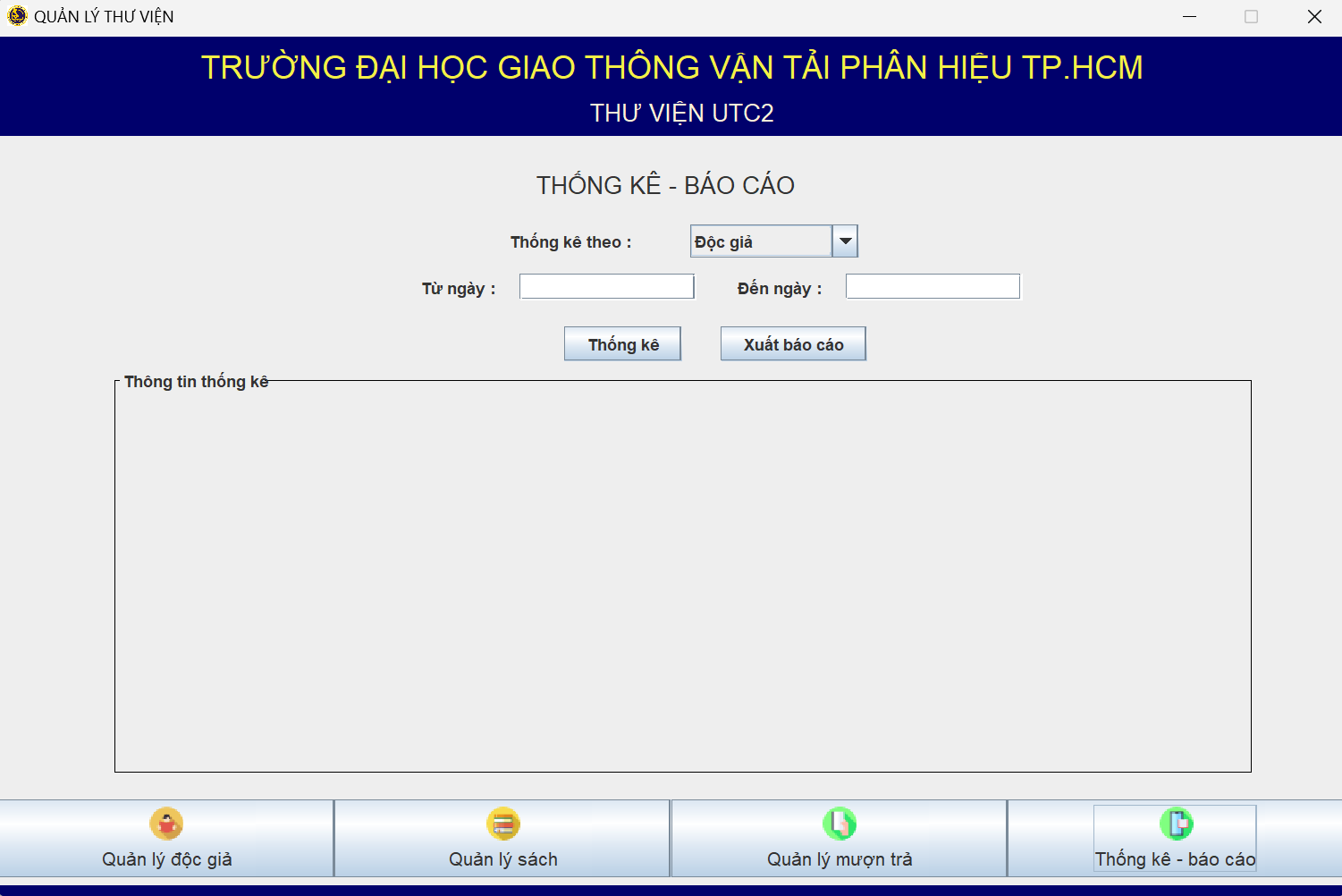
Phương thức deleteMuonTra xóa dữ liệu trong bảng MuonTra dựa trên mã mượn được truyền vào.

Phương thức deleteChiTietMuon xóa dữ liệu trong bảng ChiTietMuon dựa trên mã mượn được truyền vào.

Cả hai phương thức trên trả về một giá trị boolean để chỉ ra thành công hoặc không thành công của việc thực thi truy vấn xóa.

Xử Lý Ngoại Lệ: Các ngoại lệ SQLException được xử lý bằng cách ghi log và không được xử lý một cách cụ thể trong mã này.

## 3.6. Màn hình thống kê – báo cáo



Hình 9: Màn hình thống kê – báo cáo

Phương thức btnThongKeActionPerformed:

Được gọi khi người dùng nhấn vào nút "Thống kê".

Dựa vào lựa chọn của người dùng, nó hiển thị giao diện tương ứng cho việc thống kê thông tin về độc giả, sách hoặc mượn trả.

Phương thức btnXuatBaoCaoActionPerformed:

Được gọi khi người dùng nhấn vào nút "Xuất báo cáo".

Tùy thuộc vào lựa chọn của người dùng, nó thực hiện thống kê thông tin và xuất báo cáo dưới dạng tệp PDF.

Dữ liệu được trích xuất từ cơ sở dữ liệu thông qua các truy vấn SQL và được sắp xếp, định dạng và in ra tệp PDF.

Các phương thức hỗ trợ (NameAndSoLuong, dsDGMuonSach, baTenSachMuonnhieuNhat):

Các phương thức này được sử dụng để truy vấn cơ sở dữ liệu và trả về dữ liệu cần thiết cho việc thống kê và báo cáo.

Chúng thực hiện các truy vấn SQL và xử lý kết quả trả về từ cơ sở dữ liệu.

Các thành phần giao diện (jcbThongKe, tfTuNgay, tfDenNgay):

jcbThongKe: ComboBox cho phép người dùng chọn loại thông kê (độc giả, sách hoặc mượn trả).

tfTuNgay và tfDenNgay: Các trường văn bản cho phép người dùng nhập khoảng thời gian muốn thống kê.

Phần thống kê (pThongKe):

Là một JPanel được sử dụng để hiển thị các thành phần giao diện tương ứng với kết quả thống kê.

Cả lớp GUIThongKeBaoCao được thiết kế để hiển thị giao diện người dùng cho việc thực hiện và xuất báo cáo thống kê từ cơ sở dữ liệu.

# CHƯƠNG 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## 5.1. Kết luận

Trong quá trình thực hiện bài tập lớn Quản lý Thư viện Java, em đã có một số trải nghiệm đáng giá và nhận được nhiều bài học quý báu về lập trình và quản lý dự án phần mềm.

Đầu tiên, em đã có cơ hội hiểu rõ hơn về ngôn ngữ lập trình Java và áp dụng được nhiều khái niệm OOP (Object-Oriented Programming). Việc này giúp em xây dựng ứng dụng một cách có cấu trúc và dễ bảo trì hơn. Em đã sử dụng các nguyên tắc như kế thừa, đa hình và đóng gói để tạo ra mã nguồn linh hoạt và tái sử dụng được.

Thứ hai, em đã học được cách thiết kế và phát triển một ứng dụng Java từ đầu. Từ việc xác định các chức năng cần thiết cho thư viện đến việc triển khai chúng trong mã nguồn, em đã có cơ hội thực hành kỹ năng phát triển phần mềm thực tế.

Cuối cùng, em cũng đã nắm vững hơn về quản lý dự án phần mềm. Quản lý dự án không chỉ đơn giản là việc viết mã, mà còn bao gồm việc lập kế hoạch, phân bổ tài nguyên và kiểm soát chất lượng. Qua bài tập lớn này, em đã học được cách tổ chức dự án một cách hiệu quả, từ việc quản lý thời gian đến việc giải quyết các vấn đề phát sinh trong quá trình phát triển.

Tóm lại, bài tập lớn Quản lý Thư viện Java không chỉ giúp em rèn luyện và phát triển kỹ năng lập trình, mà còn cung cấp cho em một cái nhìn tổng quan về quy trình phát triển phần mềm và quản lý dự án.của bài tập lớn.

## 5.2. Hướng phát triển

Hướng phát triển:

* Giới hạn số lượng sách mượn của mỗi độc giả và từng quyển sách sẽ có hạn trả khác nhau.
* Phân quyền khi sử dụng phần mềm.
* Độc giả có thể đăng kí mượn sách online.

# PHỤ LỤC

Java Development Kit (JDK):

Truy cập trang web của Oracle để tải xuống và cài đặt JDK phù hợp với hệ điều hành của bạn.

Đảm bảo biến môi trường JAVA\_HOME được thiết lập và được định vị đến thư mục JDK của bạn.

Integrated Development Environment (IDE):

Sử dụng một IDE Java như IntelliJ IDEA, Eclipse hoặc NetBeans.

Tải xuống và cài đặt IDE mà bạn chọn từ trang web chính thức của nhà phát triển.

Cài Đặt Cơ Sở Dữ Liệu (Nếu Cần):

Nếu dự án của bạn sử dụng cơ sở dữ liệu, hãy cài đặt và cấu hình một hệ quản trị cơ sở dữ liệu như MySQL, PostgreSQL hoặc SQLite.

Tạo cơ sở dữ liệu và bảng cho ứng dụng của bạn, nếu cần.

Tải Về Mã Nguồn và Tập Tin Dự Án:

Tải xuống mã nguồn và tập tin dự án từ nguồn được cung cấp, chẳng hạn như GitHub hoặc một kho lưu trữ tương tự.

Giải nén tập tin nếu cần và mở dự án trong IDE của bạn.

Cấu Hình Dự Án:

Đảm bảo rằng tất cả các phụ thuộc của dự án được cài đặt và cấu hình đúng cách trong IDE của bạn.

Nếu cần, cấu hình kết nối đến cơ sở dữ liệu của bạn trong tập tin cấu hình của ứng dụng.

Xây Dựng và Chạy Ứng Dụng:

Sử dụng tính năng xây dựng và chạy trong IDE của bạn để biên dịch và chạy ứng dụng của bạn.

Kiểm tra xem ứng dụng có chạy đúng như mong đợi không và xử lý các lỗi nếu cần.

# TÀI LIỆU THAM KHẢO

1. Sách và Tài Liệu Học Tập:

Sierra, Kathy, và Bert Bates. "Head First Java." O'Reilly Media, 2005.

Bloch, Joshua. "Effective Java." Addison-Wesley Professional, 2017.

Schildt, Herbert. "Java: The Complete Reference." McGraw-Hill Education, 2018.

Flanagan, David. "Java in a Nutshell." O'Reilly Media, 2018.

Freeman, Eric, và Elisabeth Robson. "Head First Design Patterns." O'Reilly Media, 2004.

2. Hướng Dẫn Trực Tuyến và Khóa Học:

Lập trình Java – TITV

Lập trình Java với CSDL – TITV

3. Repositories và Dự Án Mẫu:

GitHub Java Repository