**PHẦN 1: MỞ ĐẦU**

**1.1 Giới thiệu đề tài**

Trong những năm gần đây, cùng với sự phát triển mạnh mẽ của nền kinh tế và sự phát triển vượt bậc của ngành Công nghệ thông tin (CNTT), tin học đang và sẽ tiếp tục phát triển không ngừng. Các ứng dụng của CNTT được áp dụng phổ biến trong nhiều lĩnh vực như Quản lý nhân sự, Quản trị kinh doanh, Y tế, Giáo dục, v.v. Việc ứng dụng tin học vào các công việc quen thuộc trong cuộc sống hàng ngày đã giúp xử lý lượng lớn các công việc khổng lồ một cách nhanh gọn và chính xác, đặc biệt phải kể đến là lĩnh vực quản lý. Việc sử dụng các phương pháp thủ công để quản lý các tác vụ như ghi chép bằng tay hoặc ghi chú bằng các ứng dụng đơn giản trở nên rất khó khăn và tốn công sức trong việc tổng hợp thông tin, dẫn đến kết quả không được như mong đợi.

Với thời đại số hóa đang đi lên như hiện nay, mọi hoạt động đều đòi hỏi sự tiện lợi, nhanh chóng và đạt được kết quả chính xác. Trong bối cảnh đó, việc quản lý các thư viện sách mini cũng không nằm ngoài xu thế này. Thay vì các phương pháp truyền thống như ghi chép sổ sách thủ công, việc xây dựng một chương trình quản lý thư viện sách mini là một nhu cầu cấp thiết. Sự đa dạng và thay đổi liên tục của các đầu sách, số lượng người mượn và trả sách hàng ngày, đòi hỏi việc quản lý cần linh hoạt và hiệu quả hơn bao giờ hết. Chính vì thế, chương trình quản lý thư viện sách mini ra đời là một ví dụ điển hình cho việc tối ưu hóa toàn bộ quy trình, từ quản lý danh mục sách, thông tin độc giả, đến việc theo dõi tình trạng mượn/trả, giúp tiết kiệm thời gian, giảm thiểu sai sót và nâng cao chất lượng dịch vụ cho người đọc.

**1.2 Tính cấp thiết của đề tài**

Việc quản lý các thông tin cần thiết tại các thư viện sách mini có vai trò rất quan trọng trong việc nâng cao hiệu quả hoạt động và tạo dựng uy tín. Giống như các quầy, quán kinh doanh, một thư viện sách mini cũng cần được tổ chức và vận hành một cách chuyên nghiệp để đáp ứng tốt nhất nhu cầu của độc giả và duy trì nguồn tài liệu.

Quản lý các tác vụ bằng phần mềm giúp công việc đạt tính chính xác cao, giúp việc theo dõi trở nên dễ dàng, khoa học, nhanh chóng hơn.

**1.3 Lý do chọn đề tài**

Từ tính cấp thiết của việc điều phối các luồng hoạt động trong quản lý thông tin và tài liệu, nhóm chúng em quyết định lựa chọn đề tài **“Xây dựng chương trình quản lý thư viện sách mini”**. Đề tài này giúp chúng em áp dụng những kiến thức đã học để xây dựng một chương trình hỗ trợ quản lý thư viện, giúp cho mọi hoạt động liên quan đến sách và độc giả trở nên dễ dàng hơn, thuận tiện và nhanh chóng hơn. Cụ thể, việc lựa chọn đề tài này xuất phát từ:

* **Nhu cầu thực tiễn:** Nhận thấy sự thiếu hụt các công cụ quản lý hiệu quả tại các thư viện sách mini, nơi thường xuyên diễn ra các hoạt động mượn, trả, và quản lý sách thủ công, dẫn đến nhiều bất cập.
* **Áp dụng kiến thức:** Đề tài là cơ hội tuyệt vời để thực hành và củng cố các kiến thức về lập trình, cơ sở dữ liệu, và thiết kế giao diện người dùng vào một dự án thực tế.
* **Góp phần tối ưu hóa:** Chương trình được kỳ vọng sẽ giúp tối ưu hóa các quy trình quản lý sách và độc giả, giảm thiểu sai sót, tiết kiệm thời gian và nguồn lực cho người quản lý thư viện.
* **Nâng cao trải nghiệm:** Mục tiêu cuối cùng là mang lại trải nghiệm tốt hơn cho độc giả, giúp họ dễ dàng tìm kiếm sách và theo dõi thông tin cá nhân.

**1.4 Công cụ thực hiện đề tài**

* NetBeans IDE: NetBeans là một môi trường phát triển tích hợp (IDE) mạnh mẽ, được thiết kế chủ yếu cho ngôn ngữ lập trình Java. Nó cho phép các ứng dụng được phát triển từ một tập hợp các thành phần phần mềm được gọi là modules. NetBeans hoạt động linh hoạt trên nhiều nền tảng phổ biến như Windows, macOS và Linux. Đối với đề tài này, NetBeans sẽ cung cấp một giao diện trực quan và các công cụ hỗ trợ toàn diện cho việc thiết kế giao diện người dùng (GUI) dạng Swing, quản lý mã nguồn, biên dịch, và gỡ lỗi (debug) chương trình, từ đó giúp đẩy nhanh đáng kể quá trình phát triển ứng dụng quản lý thư viện.
* Java Development Kit (JDK) 21: JDK 21 là phiên bản hỗ trợ dài hạn (LTS) mới nhất của Java Development Kit, bao gồm toàn bộ môi trường phát triển và các công cụ cần thiết để xây dựng và triển khai ứng dụng Java. Phiên bản này được lựa chọn dựa trên tính ổn định cao, các cải tiến về hiệu suất, và các tính năng mới được tích hợp. JDK 21 sẽ là nền tảng cốt lõi để viết và chạy toàn bộ mã nguồn của chương trình quản lý thư viện, đảm bảo ứng dụng hoạt động hiệu quả, bảo mật và tương thích với các tiêu chuẩn công nghệ hiện đại.
* SQL Server: SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mạnh mẽ do Microsoft phát triển, chuyên dùng để lưu trữ, quản lý và truy xuất dữ liệu một cách hiệu quả. Trong khuôn khổ đề tài này, SQL Server sẽ đóng vai trò là nơi lưu trữ tập trung cho toàn bộ thông tin của thư viện, bao gồm danh mục các đầu sách, thông tin chi tiết của người mượn, và các giao dịch mượn/trả sách. Việc sử dụng SQL Server giúp đảm bảo tính toàn vẹn, bảo mật cao và khả năng mở rộng của dữ liệu, là yếu tố then chốt cho một hệ thống quản lý thư viện bền vững và hiệu quả.
* JDBC Driver cho SQL Server: Để thiết lập kênh giao tiếp giữa ứng dụng Java và cơ sở dữ liệu SQL Server, chúng em sẽ sử dụng JDBC (Java Database Connectivity) Driver chuyên biệt cho SQL Server (thường là file mssql-jdbc.jar). JDBC là một giao diện lập trình ứng dụng (API) chuẩn của Java, cho phép các ứng dụng Java kết nối và tương tác với các hệ quản trị cơ sở dữ liệu khác nhau. Driver này đóng vai trò cầu nối, giúp chương trình có thể thực hiện linh hoạt các thao tác cơ bản như thêm mới, chỉnh sửa, xóa và truy vấn dữ liệu từ SQL Server.
* CASE Studio / draw.io / StarUML (tùy chọn): CASE Studio là một công cụ chuyên nghiệp để thiết kế mô hình dữ liệu, hỗ trợ vẽ Entity Relationship Diagrams (ERD) và Data Flow Diagrams (DFD), đồng thời có khả năng tự động tạo scripts SQL cho các cơ sở dữ liệu khác nhau. Tuy nhiên, để linh hoạt và tận dụng các công cụ trực quan hóa phổ biến hơn, nhóm em có thể cân nhắc sử dụng draw.io hoặc StarUML. Các công cụ này sẽ giúp chúng em thiết kế mô hình cơ sở dữ liệu một cách rõ ràng và xây dựng sơ đồ lớp (UML Class Diagram), từ đó hình dung cấu trúc dữ liệu và mối quan hệ giữa các thành phần phần mềm trước khi tiến hành lập trình.

**PHẦN 2: KẾT QUẢ NGHIÊN CỨU**

**2.1 Giới thiệu**

Trong khuôn khổ Bài tập lớn môn học, nhóm chúng em đã tiến hành phân tích, thiết kế và xây dựng một ứng dụng phần mềm hoàn chỉnh. Nhiệm vụ trọng tâm là áp dụng các kiến thức về lập trình hướng đối tượng, kiến trúc phần mềm và các công nghệ liên quan để tạo ra một sản phẩm có tính ứng dụng thực tế. Kết quả của quá trình này là dự án **"Phần mềm Quản lý Thư viện mini"**, một ứng dụng desktop được xây dựng bằng ngôn ngữ Java, có khả năng thực hiện các nghiệp vụ cơ bản của một thư viện mini.

Dự án được xây dựng nhằm mục đích tin học hóa các công tác quản lý tại một thư viện, bao gồm các chức năng chính:

* **Quản lý Sách:** Thêm, sửa, xóa (ẩn) và tìm kiếm thông tin các đầu sách.
* **Quản lý Người mượn:** Quản lý thông tin độc giả của thư viện.
* **Quản lý Mượn/Trả sách:** Ghi nhận các phiếu mượn, xử lý nghiệp vụ trả sách và gia hạn.
* **Thống kê và Sao lưu:** Thống kê số lượng sách được mượn theo thời gian và cung cấp chức năng sao lưu, khôi phục cơ sở dữ liệu để đảm bảo an toàn dữ liệu.

Để đảm bảo dự án có cấu trúc rõ ràng, dễ bảo trì và mở rộng, nhóm đã thống nhất lựa chọn quy trình xây dựng phần mềm dựa trên kiến trúc phân lớp. Cụ thể, mã nguồn của dự án được tổ chức thành các tầng riêng biệt:

* **Tầng giao diện (GUI):** Chịu trách nhiệm về giao diện và tương tác người dùng.
* **Tầng đối tượng (Object):** Định nghĩa các đối tượng dữ liệu như Sach, NguoiMuon, PhieuMuon.
* **Tầng logic nghiệp vụ (Logic):** Xử lý các quy tắc nghiệp vụ và logic của ứng dụng.
* **Tầng truy cập dữ liệu (CRUD):** Chịu trách nhiệm giao tiếp và thao tác với cơ sở dữ liệu.

Sản phẩm được cài đặt và triển khai hoàn toàn bằng NetBeans IDE, một môi trường phát triển tích hợp (IDE) mạnh mẽ cho Java. Lý do chính nhóm lựa chọn NetBeans là vì công cụ này cung cấp trình thiết kế giao diện GUI Buildertheo cơ chế kéo-thả rất hiệu quả. Điều này giúp đẩy nhanh đáng kể quá trình xây dựng giao diện người dùng cho ứng dụng desktop (Java Swing), cho phép nhóm tập trung nhiều hơn vào việc xử lý logic nghiệp vụ.

Ngoài NetBeans và bộ công cụ phát triển Java (JDK), dự án còn sử dụng các công cụ và thư viện hỗ trợ sau:

* **Hệ quản trị CSDL:** Microsoft SQL Server để lưu trữ dữ liệu.
* **Thư viện giao diện:** **FlatLaf** được tích hợp để tạo ra một giao diện người dùng hiện đại, phẳng và sạch sẽ, cải thiện trải nghiệm người dùng so với giao diện Swing mặc định.
* **Thư viện biểu đồ:** **JFreeChart** được sử dụng cho chức năng thống kê, cho phép trực quan hóa dữ liệu mượn sách một cách sinh động.
* **Thư viện lịch:** **JCalendar** cung cấp thành phần JDateChooser để người dùng có thể chọn ngày tháng một cách thuận tiện.

**2.2 Khảo sát hệ thống**

**2.2.1 Khảo sát sơ bộ**

Mục tiêu chính của việc xây dựng chương trình quản lý thư viện sách mini nhằm tối ưu hóa các quy trình liên quan đến hoạt động của thư viện, từ việc lưu trữ thông tin sách, thông tin người mượn cho đến theo dõi quá trình mượn, trả sách và quản lý tình trạng tài liệu.

Về phạm vi, hệ thống cần đáp ứng được các yêu cầu cơ bản như quản lý được danh mục sách (bao gồm thông tin về tên sách, tác giả, thể loại, tình trạng), quản lý thông tin người mượn, và ghi nhận được các giao dịch mượn/trả sách xảy ra. Ngoài ra, chương trình cũng cần cung cấp các tính năng nâng cao như xem thống kê số lượng sách theo thể loại, thống kê số lượt mượn/trả theo ngày, tháng, hoặc năm, và danh sách sách quá hạn. Qua đó, chương trình giúp quản lý tài nguyên sách hiệu quả hơn, nâng cao năng suất làm việc của thủ thư và cải thiện trải nghiệm cho độc giả.

**2.2.2.1 Yêu cầu chức năng**

* Quản lý Sách: Cho phép quản trị viên thực hiện các thao tác thêm mới, cập nhật thông tin (tên sách, tác giả, thể loại, nhà xuất bản, năm xuất bản, số lượng, tình trạng), và xóa thông tin sách khỏi hệ thống (SACH).
* Quản lý Người mượn: Cho phép quản trị viên quản lý thông tin của người mượn, bao gồm tạo mới, chỉnh sửa thông tin, xoá người mượn khỏi hệ thống(NGUOIMUON).
* Quản lý mượn/trả sách: Cho phép quản trị viên tạo phiếu mượn sách mới, cập nhật trạng thái phiếu mượn (ví dụ: đang mượn, đã trả, quá hạn), ghi nhận ngày trả, số lần gia hạn cho phiếu mượn, và xem danh sách tất cả các phiếu mượn/trả (PHIEUMUON).
* Thống kê mượn sách: Cung cấp cho quản trị viên công cụ để xem thống kê tổng quan về hoạt động mượn sách của thư viện theo các chu kỳ thời gian (ngày, tháng, năm), với khả năng trực quan hóa dữ liệu qua biểu đồ, được truy cập trên một tab riêng biệt.

**2.2.2.2 Yêu cầu phi chức năng**

* Độ khả dụng: Giao diện người dùng của hệ thống phải được thiết kế một cách trực quan, dễ hiểu và dễ thao tác, giúp người quản lý và độc giả có thể sử dụng các chức năng một cách hiệu quả mà không cần nhiều hướng dẫn.
* Độ tin cậy: Hệ thống cần đảm bảo hoạt động ổn định, chính xác trong mọi tác vụ nghiệp vụ (thêm/sửa/xóa dữ liệu, mượn/trả sách). Dữ liệu được lưu trữ phải chính xác và không bị mất mát. Hệ thống cũng cần có khả năng xử lý lỗi một cách mềm dẻo.
* Bảo mật: Hệ thống phải có khả năng bảo mật dữ liệu quan trọng của thư viện (thông tin sách, phiếu mượn, thông tin người mượn).
* Khả năng bảo trì: Mã nguồn của hệ thống cần được viết rõ ràng, có cấu trúc dễ hiểu và dễ dàng cho việc bảo trì, sửa lỗi, nâng cấp cũng như bổ sung các tính năng mới trong tương lai.
* Khả năng sử dụng: Ứng dụng phải cung cấp thông tin một cách rõ ràng, dễ tiếp cận, và có luồng nghiệp vụ hợp lý, giúp người dùng dễ dàng tìm thấy chức năng và dữ liệu mà họ đang tìm kiếm.
* Khả năng tương tác: Các yếu tố và thành phần trên ứng dụng phải phản hồi nhanh chóng với thao tác của người dùng, đảm bảo trải nghiệm tương tác mượt mà và không bị gián đoạn.
* Lưu trữ cơ sở dữ liệu: Dữ liệu của hệ thống sẽ được lưu trữ tập trung và an toàn trong Cơ sở dữ liệu SQL Server.
* Ngôn ngữ lập trình: Hệ thống sẽ được phát triển bằng ngôn ngữ Java và sử dụng thư viện Swing để xây dựng giao diện người dùng.
* Môi trường cài đặt: Ứng dụng được thiết kế để chạy và cài đặt trên hệ điều hành Microsoft Windows.

**2.3 Phân tích hệ thống**

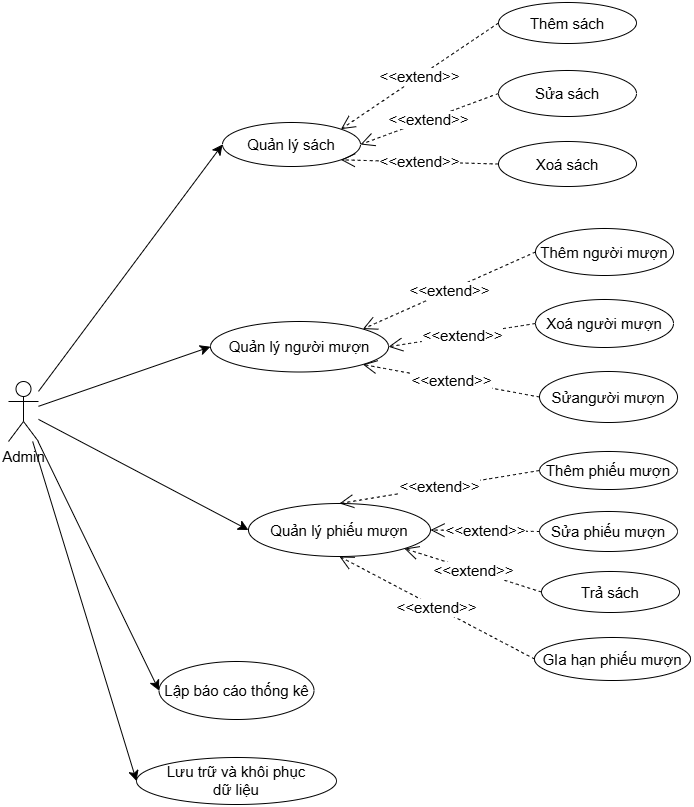
**2.3.1 Mô hình hóa chức năng hệ thống**

- Actor chính sử dụng chương trình và vai trò:

* Quản trị viên: Quản lý Sách, Quản lý Người dùng , Quản lý Phiếu Mượn/Trả, Thống kê mượn sách.

- Mô tả sơ bộ các Use Case:

* Quản lý sách: Cho phép quản trị viên thực hiện các thao tác thêm mới, cập nhật thông tin, và xóa thông tin sách khỏi hệ thống.
* Quản lý người dùng : Cho phép quản trị viên quản lý thông tin các tài khoản độc giả, bao gồm tạo mới, chỉnh sửa thông tin, xem lịch sử mượn của độc giả, cũng như kích hoạt/khóa tài khoản độc giả.
* Quản lý mượn/trả sách: Cho phép quản trị viên tạo phiếu mượn sách mới, cập nhật trạng thái phiếu mượn, ghi nhận ngày trả, số lần gia hạn, và xem danh sách tất cả các phiếu mượn/trả.
* Thống kê mượn sách: Cung cấp chức năng cho quản trị viên xem và trực quan hóa các báo cáo thống kê về hoạt động mượn sách của thư viện theo các chu kỳ thời gian.



*Hình 2.1 Biểu đồ use case*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Id | Tên use case | Mô tả ngắn gọn | Chức năng | Ghi chú |
| UC\_001 | Quan ly sach | Cho phép quản trị viên quản lý thông tin về các đầu sách | Thực hiện các thao tác thêm, hiển thị, tìm kiếm, sửa và xóa thông tin sách | Quản trị viên |
| UC\_002 | Quan ly nguoi dung | Cho phép quản trị viên quản lý thông tin của các độc giả. | Thực hiện các thao tác thêm, hiển thị, sửa và xóa thông tin người dùng (độc giả | Quản trị viên |
| UC\_003 | Quan ly muon/tra sach | Cho phép quản trị viên quản lý các giao dịch mượn và trả sách | Thực hiện các thao tác liên quan đến việc cho mượn, trả sách và kiểm tra trạng thái mượn | Quản trị viên |
| UC\_004 | Thong ke muon sach | Cho phép quản trị viên xem và trực quan hóa các báo cáo thống kê về hoạt động mượn sách của thư viện theo các chu kỳ thời gian. | Cung cấp chức năng xem và trực quan hóa các báo cáo thống kê mượn sách theo thời gian. | Quản trị viên |

**2.3.2 Mô hình hóa dữ liệu**

2.3.2.1 Thông tin lưu trữ

Thuộc tính của các lớp:

* Lớp SACH gồm có các thuộc tính:
* MaSach: Thuộc tính khóa để xác định mã định danh duy nhất của mỗi cuốn sách.
* TenSach: Thuộc tính xác định tên đầy đủ của cuốn sách.
* TacGia: Thuộc tính xác định tên tác giả của cuốn sách.
* TheLoai: Thuộc tính xác định thể loại của cuốn sách.
* NhaXuatBan: Thuộc tính xác định tên nhà xuất bản của cuốn sách.
* NamXuatBan: Thuộc tính xác định năm cuốn sách được xuất bản.
* SoLuong: Thuộc tính xác định tổng số lượng bản sao hiện có của cuốn sách trong thư viện.
* TinhTrang: Thuộc tính xác định tình trạng của sách.
* Lớp NGUOIMUON gồm có các thuộc tính:
* MaNguoiMuon: Thuộc tính khóa để xác định mã định danh duy nhất của mỗi người mượn.
* HoTen: Thuộc tính xác định họ và tên đầy đủ của người mượn.
* GioiTinh: Thuộc tính xác định giới tính của người mượn.
* DiaChi: Thuộc tính xác định địa chỉ liên hệ của người mượn.
* SoDienThoai: Thuộc tính xác định số điện thoại liên hệ của người mượn.
* Email: Thuộc tính xác định địa chỉ email của người mượn.
* Lớp PHIEUMUON gồm có các thuộc tính:
* MaPhieuMuon: Thuộc tính khóa để xác định mã định danh duy nhất của mỗi phiếu mượn.
* MaSach: Thuộc tính xác định cuốn sách được mượn trong phiếu này.
* MaNguoiMuon: Thuộc tính xác định người mượn đã thực hiện việc mượn sách.
* SoLuong: Thuộc tính xác định số lượng bản sao của cuốn sách được mượn trong phiếu này.
* NgayMuon: Thuộc tính xác định ngày mà sách được mượn.
* NgayTra: Thuộc tính xác định ngày dự kiến hoặc ngày thực tế mà sách phải/đã được trả lại.
* TinhTrang: Thuộc tính xác định trạng thái hiện tại của phiếu mượn.
* SoLanGiaHan: Thuộc tính xác định số lần mà phiếu mượn này đã được gia hạn.

**2.3.2.2 Biểu đồ lớp**

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

Nội dung do AI tạo ra có thể không chính xác.**

*Hình 2.2 Biểu đồ lớp*

**2.4 Thực hiện bài toán**

**2.4.1 Quản lý sách**

**2.4.2 Quản lý người mượn**

**Giao diện chức năng**

Giao diện chức năng Quản lý Người Mượn đã được xây dựng hoàn thành với Java Swing. Giao diện này được thiết kế bằng cách sử dụng công cụ kéo thả của NetBeans IDE, đảm bảo tính trực quan và dễ sử dụng.

**Thành phần và chức năng:**

Bảng hiển thị: Một JTable được sử dụng để hiển thị danh sách người mượn với các thông tin chi tiết như Mã người mượn, Họ tên, Giới tính, Địa chỉ, Số điện thoại và Email. Để đảm bảo tính toàn vẹn của dữ liệu, bảng này được cấu hình để không cho phép chỉnh sửa trực tiếp trên các ô.

Khu vực nhập liệu: Các JTextField để nhập liệu cho Họ tên, Địa chỉ, Số điện thoại và Email. Các JComboBox để chọn Mã người mượn (khi cần tra cứu hoặc sửa/xóa) và Giới tính.

**Các nút chức năng:**

* + Thêm người mượn: Cho phép thêm một người mượn mới vào cơ sở dữ liệu dựa trên thông tin đã nhập.
  + Sửa người mượn: Cập nhật thông tin của một người mượn đã được chọn trong bảng.
  + Xoá người mượn: Xoá một người mượn ra khỏi hệ thống. Chức năng này có kiểm tra ràng buộc, không cho phép xóa nếu người đó đang mượn sách.
  + Làm mới: Tải lại toàn bộ dữ liệu từ cơ sở dữ liệu lên bảng và đặt lại các trường nhập liệu về trạng thái ban đầu.
  + Thoát: Đóng ứng dụng.

Tìm kiếm: Cung cấp một JComboBox cho phép người dùng chọn tiêu chí tìm kiếm (theo mã hoặc theo tên) và một nút "Tìm kiếm" để thực hiện lọc dữ liệu trên JTable bằng cách sử dụng TableRowSorter.

**Mã lệnh**

* **Hướng đối tượng** Quản lý người mượn đã áp dụng và thể hiện rõ ràng cả 4 tính chất nền tảng của Lập trình Hướng đối tượng (OOP), bao gồm: Tính Đóng Gói, Tính Kế Thừa, Tính Trừu Tượng và Tính Đa Hình. Việc áp dụng các tính chất này giúp mã nguồn trở nên có cấu trúc, linh hoạt và dễ bảo trì.
  + **Tính đóng gói:** Được thể hiện ở lớp NguoiMuon.java, nơi tất cả các thuộc tính như maNguoiMuon, hoTen, diaChi,... đều được khai báo với mức truy cập private. Việc truy xuất hoặc thay đổi giá trị của các thuộc tính này phải được thực hiện thông qua các phương thức public getter và setter, đảm bảo dữ liệu được kiểm soát chặt chẽ.

A close up of words

AI-generated content may be incorrect.

A close-up of a text

AI-generated content may be incorrect.

* + **Tính kế thừa:** Lớp LogicNguoiMuon sử dụng lớp ngoại lệ tùy chỉnh InvalidDataException, vốn được tạo ra bằng cách kế thừa từ lớp Exception của Java. Ngoài ra, lớp giao diện tabQuanLyNguoiMuon kế thừa từ lớp javax.swing.JPanel để tái sử dụng các chức năng của một panel trong Swing.

A white background with black text

AI-generated content may be incorrect.

* + **Tính trừu tượng:** Được áp dụng tại tầng kết nối giữa LogicNguoiMuon và NguoiMuonCRUD. Cụ thể, interface INguoiMuonCRUD được tạo ra để định nghĩa một bộ các hành vi bắt buộc cho việc xử lý dữ liệu người mượn (ví dụ: getDsNguoiMuon, themNguoiMuon, suaNguoiMuon, xoaNguoiMuon) mà không cần quan tâm đến việc chúng được cài đặt chi tiết bằng câu lệnh SQL nào.

A screenshot of a computer code

AI-generated content may be incorrect.

* + **Tính đa hình:** Được thể hiện rõ trong lớp LogicNguoiMuon. Lớp này làm việc với một biến có kiểu là interface INguoiMuonCRUD (nguoiMuon), nhưng biến đó lại tham chiếu đến một đối tượng của lớp NguoiMuonCRUD. Khi một phương thức được gọi thông qua biến interface, phương thức tương ứng của đối tượng NguoiMuonCRUD sẽ được thực thi tại thời điểm chạy.



* **Mô hình phân tầng:** Dự án được xây dựng theo mô hình 3 tầng rõ rệt, giúp tách biệt giao diện, logic nghiệp vụ và truy cập dữ liệu:
  + **Tầng Giao diện (GUI):** Lớp tabQuanLyNguoiMuon.java chịu trách nhiệm hiển thị dữ liệu và nhận tương tác từ người dùng.
  + **Tầng Logic nghiệp vụ (Logic):** Lớp LogicNguoiMuon.java có nhiệm vụ xử lý các quy tắc nghiệp vụ như kiểm tra dữ liệu đầu vào có hợp lệ không, hoặc kiểm tra xem người mượn có đang mượn sách không trước khi xóa.
  + **Tầng Truy cập dữ liệu (CRUD):** Lớp NguoiMuonCRUD.java chịu trách nhiệm thực thi các câu lệnh SQL để thao tác trực tiếp với bảng NGUOIMUON trong cơ sở dữ liệu.
* **Sử dụng Interface:** Tầng truy cập dữ liệu được thiết kế thông qua interface INguoiMuonCRUD. Lớp NguoiMuonCRUD sẽ implement interface này. Cách làm này giúp giảm sự phụ thuộc giữa các tầng, giúp việc bảo trì hoặc thay đổi phương thức truy cập dữ liệu trong tương lai dễ dàng hơn mà không ảnh hưởng đến tầng Logic và GUI.



* **Lớp kiểm tra dữ liệu hợp lệ:** Lớp LogicNguoiMuon có phương thức checked(NguoiMuon nm) để kiểm tra các điều kiện như số điện thoại có đúng 10 ký tự hay không và các trường thông tin quan trọng có bị bỏ trống không.

A screen shot of a computer code

AI-generated content may be incorrect.

* **Các thống nhất trong bài tập lớn:**
  + **Tính đóng gói:** Được áp dụng triệt để cho các lớp thực thể trong gói Object (NguoiMuon, Sach, PhieuMuon). Tất cả thuộc tính đều là private và được truy cập qua public getters/setters.
  + **Tính kế thừa:** Được sử dụng để tái sử dụng code từ thư viện (ví dụ extends javax.swing.JPanel) và để tạo các kiểu dữ liệu chuyên biệt (ví dụ extends Exception).
  + **Interface:** Được dùng nhất quán để kết nối giữa tầng Logic và tầng CRUD cho tất cả các đối tượng, giúp hệ thống linh hoạt và giảm phụ thuộc.
  + **Phân tầng theo gói:** Cấu trúc các gói GUI, Logic, CRUD, Object, Exception phân chia rõ ràng trách nhiệm của từng phần trong hệ thống, giúp mã nguồn có tổ chức và dễ quản lý.

**Bắt lỗi và gom rác**

* **Cơ chế bẫy lỗi:**
  + Chương trình không bắt lỗi trong các phương thức set của lớp thực thể. Thay vào đó, việc kiểm tra tính hợp lệ của dữ liệu được tập trung thực hiện ở tầng Logic (LogicNguoiMuon.java).
  + Khi phát hiện lỗi nghiệp vụ (ví dụ: thông tin chưa nhập đầy đủ, số điện thoại sai định dạng, hoặc người dùng đang mượn sách không thể xóa), tầng Logic sẽ ném ra một ngoại lệ tự định nghĩa là InvalidDataException kèm theo thông báo lỗi cụ thể.
  + Tầng GUI (tabQuanLyNguoiMuon.java) đặt các lời gọi đến tầng Logic trong một khối try-catch. Khi bắt được InvalidDataException, nó sẽ hiển thị thông báo lỗi cho người dùng thông qua JOptionPane, giúp người dùng hiểu rõ vấn đề và cách khắc phục.

A screen shot of a computer code

AI-generated content may be incorrect.

* **Lớp xử lý lỗi tự tạo:**
  + Nhóm đã tạo lớp ngoại lệ riêng là InvalidDataException kế thừa từ lớp Exception của Java.
  + **Tác dụng:** Lớp này được dùng để đóng gói các thông báo lỗi liên quan đến nghiệp vụ của ứng dụng. Điều này giúp mã nguồn ở tầng GUI trở nên sạch sẽ và dễ đọc hơn. Thay vì phải kiểm tra nhiều điều kiện và hiển thị các thông báo khác nhau, tầng GUI chỉ cần catch một loại Exception và hiển thị thông điệp chứa trong đó.

**Tập hợp**

* **Loại tập hợp được chọn:** ArrayList<NguoiMuon> được sử dụng trong lớp NguoiMuonCRUD để lưu trữ danh sách các người mượn đọc từ cơ sở dữ liệu.
* **Lý do lựa chọn:**
  + ArrayList rất phù hợp với nhu cầu đọc toàn bộ dữ liệu từ ResultSet của cơ sở dữ liệu và lưu trữ lại trong bộ nhớ.
  + Nó cho phép truy cập phần tử theo chỉ số một cách nhanh chóng, rất hữu ích khi cần lấy dữ liệu từ một hàng cụ thể trong JTable mà người dùng đã chọn.
  + Thao tác lặp trên ArrayList rất hiệu quả, phù hợp cho việc duyệt qua danh sách để hiển thị lên giao diện hoặc xử lý logic.
* **Các phép toán sử dụng:**
  + **Thêm phần tử:** Phương thức add() được dùng trong phương thức getDsNguoiMuon() để thêm đối tượng NguoiMuon mới vào danh sách sau khi đọc từ ResultSet.

A computer screen shot of a code

AI-generated content may be incorrect.

* + **Duyệt tập hợp:** Vòng lặp for-each được sử dụng trong phương thức layDuLieu() của tabQuanLyNguoiMuon để duyệt qua các phần tử trong ArrayList và hiển thị chúng lên JTable.

A screen shot of a computer code

AI-generated content may be incorrect.

**Thao tác file: Không có**

**2.4.3 Quản lý mượn/trả sách**

**Giao diện chức năng**

Giao diện chức năng quản lý mượn-trả sách đã xây dựng hoàn thành với Java Swing. Giao diện được thiết kế bằng cách sử dụng công cụ kéo thả của NetBeans IDE.

**Thành phần và chức năng:**

Bảng hiển thị: Một JTable để hiển thị danh sách các phiếu mượn với các thông tin chi tiết như mã phiếu, mã sách, mã người mượn, số lượng, ngày mượn, ngày trả, tình trạng và số lần gia hạn. Bảng này không cho phép sửa trực tiếp để đảm bảo tính toàn vẹn dữ liệu.

Khu vực nhập liệu: Gồm các JComboBox để chọn mã phiếu mượn, mã sách, mã người mượn, JSpinner để nhập số lượng và JDateChooser từ thư viện jcalendar để chọn ngày trả.

Các nút chức năng:

* + Mượn sách: Cho phép tạo phiếu mượn mới.
  + Trả sách: Cập nhật trạng thái phiếu mượn thành "Đã trả" và cập nhật lại số lượng sách trong kho.
  + Sửa phiếu: Sửa thông tin của một phiếu mượn đã chọn.
  + Gia hạn: Cho phép gia hạn ngày trả sách thêm 3 ngày.
  + Làm mới: Tải lại toàn bộ dữ liệu từ cơ sở dữ liệu và đặt lại các trường nhập liệu về trạng thái ban đầu.
  + Kiểm tra: Kiểm tra số lượng sách còn lại trong kho trước khi cho mượn.
  + Thoát: Đóng ứng dụng.

Tìm kiếm: Cung cấp JComboBox cho phép người dùng chọn tiêu chí tìm kiếm (theo mã phiếu, mã sách, mã người mượn, hoặc các phiếu đang mượn/quá hạn) và một nút “Tìm kiếm” để thực hiện lọc dữ liệu trên bảng.

Xem chi tiết: Người dùng có thể nhấp đúp chuột vào một hàng trong bảng để mở một cửa sổ mới hiển thị thông tin chi tiết về người mượn và sách tương ứng với phiếu mượn đó.

**Mã lệnh:**

* **Hướng đối tượng:** Quản lý mượn/trả sách đã áp dụng và thể hiện rõ ràng cả 4 tính chất nền tảng của Lập trình Hướng đối tượng (OOP), bao gồm: Tính Đóng Gói, Tính Kế Thừa, Tính Trừu Tượng và Tính Đa Hình. Việc áp dụng các tính chất này giúp mã nguồn trở nên có cấu trúc, linh hoạt và dễ bảo trì.
  + Tính đóng gói được thể hiện ở lớp PhieuMuon.java tất cả các thuộc tính đều được khai báo với mức truy cập private. Việc truy xuất hoặc thay đổi giá trị của thuộc tính phải được thực hiện thông qua các phương thức getter và setter.

A white background with black and pink text

AI-generated content may be incorrect.

A white background with black and pink text

AI-generated content may be incorrect.

* + Tính kế thừa: Lớp LogicPhieuMuon sử dụng lớp ngoại lệ tùy chỉnh InvalidDataException, vốn được tạo ra bằng cách kế thừa từ lớp Exception của Java.

A computer screen shot of a code

AI-generated content may be incorrect.

* + Tính trừu tượng được áp dụng tại tầng kết nối giữa LogicPhieuMuon và PhieuMuonCRUD. Cụ thể, interface IPhieuMuonCRUD được tạo ra để định nghĩa một bộ các hành vi bắt buộc cho việc xử lý dữ liệu phiếu mượn (ví dụ: themPhieuMuon, traSach, giaHanSach) mà không cần quan tâm đến việc chúng được cài đặt chi tiết bằng câu lệnh SQL nào.

A screenshot of a computer code

AI-generated content may be incorrect.

* + Tính đa hình được thể hiện rõ nhất trong lớp LogicPhieuMuon. Lớp này làm việc với một biến có kiểu là interface IPhieuMuonCRUD, nhưng biến đó lại tham chiếu đến một đối tượng của lớp PhieuMuonCRUD. Khi gọi một phương thức thông qua biến interface, phương thức tương ứng của đối tượng PhieuMuonCRUD sẽ được thực thi.



* **Mô hình phân tầng:** Dự án được xây dựng theo mô hình 3 tầng rõ rệt, giúp tách biệt giao diện, logic nghiệp vụ và truy cập dữ liệu:
  + Tầng Giao diện (GUI): Gồm các lớp trong gói GUI, ví dụ tabQuanLyPhieuMuon.java, chịu trách nhiệm hiển thị dữ liệu và nhận tương tác từ người dùng.
  + Tầng Logic nghiệp vụ (Logic): Gồm các lớp trong gói Logic, ví dụ LogicPhieuMuon.java, có nhiệm vụ xử lý các quy tắc nghiệp vụ như kiểm tra số lượng sách, giới hạn số lần mượn, v.v. trước khi gọi đến tầng truy cập dữ liệu.
  + Tầng Truy cập dữ liệu (CRUD): Gồm các lớp trong gói CRUD, ví dụ PhieuMuonCRUD.java, chịu trách nhiệm thực thi các câu lệnh SQL để thao tác trực tiếp với cơ sở dữ liệu.
* **Sử dụng Interface:** Tầng truy cập dữ liệu được thiết kế thông qua các interface IPhieuMuonCRUD. Lớp PhieuMuonCRUD sẽ implement interface này. Cách làm này giúp giảm sự phụ thuộc giữa các tầng, dễ dàng thay đổi hoặc bảo trì cơ sở dữ liệu trong tương lai mà không ảnh hưởng đến tầng Logic và GUI.



* **Lớp kiểm tra dữ liệu hợp lệ:** Các lớp trong gói Logic đóng vai trò kiểm tra dữ liệu. Ví dụ, lớp LogicPhieuMuon có phương thức checked(PhieuMuon phieu) để kiểm tra các điều kiện như ngày trả phải sau ngày hiện tại, số lượng mượn phải lớn hơn 0, và các mã phải được chọn.

**A computer screen shot of a code

AI-generated content may be incorrect.**

* **Các thống nhất trong bài tập lớn:**
* Tính đóng gói: Được áp dụng triệt để cho các lớp thực thể trong gói Object. Tất cả thuộc tính đều là private và được truy cập qua public getters/setters.
* Tính kế thừa: Được sử dụng để tái sử dụng code từ thư viện (ví dụ extends javax.swing.JPanel) và để tạo các kiểu dữ liệu chuyên biệt (ví dụ extends Exception).
* Interface: Được dùng để kết nối giữa tầng Logic và tầng CRUD, giúp hệ thống linh hoạt và giảm phụ thuộc.
* Phân tầng theo gói: Cấu trúc các gói GUI, Logic, CRUD, Object, Exception phân chia rõ ràng trách nhiệm của từng phần trong hệ thống, giúp mã nguồn có tổ chức và dễ quản lý.
* **Sơ đồ lớp**

A diagram of a computer program

AI-generated content may be incorrect.

**Bắt lỗi và gom rác**

* Cơ chế bẫy lỗi:
  + Chương trình không bắt lỗi trong các phương thức set của lớp thực thể. Thay vào đó, việc kiểm tra tính hợp lệ của dữ liệu được thực hiện ở tầng Logic (LogicPhieuMuon.java).
  + Khi phát hiện lỗi nghiệp vụ (ví dụ: số lượng sách không đủ, người mượn đã mượn quá 3 cuốn, ngày trả không hợp lệ), tầng Logic sẽ ném ra một ngoại lệ tự định nghĩa là InvalidDataException.
  + Tầng GUI (tabQuanLyPhieuMuon.java) sẽ đặt các lời gọi đến tầng Logic trong một khối try-catch. Khi bắt được InvalidDataException, nó sẽ hiển thị thông báo lỗi cho người dùng thông qua JOptionPane, giúp người dùng hiểu rõ lỗi và cách khắc phục.
* Lớp xử lý lỗi tự tạo:
  + Nhóm đã tạo lớp ngoại lệ riêng là InvalidDataException kế thừa từ lớp Exception của Java.
  + Tác dụng: Lớp này được dùng để đóng gói các thông báo lỗi liên quan đến nghiệp vụ của ứng dụng, giúp mã nguồn ở tầng GUI trở nên sạch sẽ hơn. Thay vì kiểm tra nhiều điều kiện và hiển thị các thông báo khác nhau, tầng GUI chỉ cần catch một loại Exception và hiển thị thông điệp chứa trong đó.

A computer screen with text

AI-generated content may be incorrect.

**Tập hợp**

* Loại tập hợp được chọn: ArrayList được sử dụng để lưu trữ danh sách các đối tượng như PhieuMuon.
* Lý do lựa chọn:
  + ArrayList phù hợp với nhu cầu lưu trữ và truy xuất dữ liệu từ cơ sở dữ liệu. Dữ liệu được đọc một lần từ ResultSet và nạp vào ArrayList.
  + Nó cho phép truy cập phần tử theo chỉ số một cách nhanh chóng, rất hữu ích khi cần lấy dữ liệu từ một hàng cụ thể trong JTable.
  + Thao tác lặp trên ArrayList rất hiệu quả, phù hợp cho việc duyệt qua danh sách để hiển thị lên giao diện hoặc xử lý logic.
* Các phép toán sử dụng:
  + Thêm phần tử: Phương thức add() được dùng để thêm đối tượng mới vào danh sách sau khi đọc từ ResultSet.

A computer code with many text

AI-generated content may be incorrect.

* + Duyệt tập hợp: Vòng lặp for-each được sử dụng để duyệt qua các phần tử trong ArrayList và hiển thị chúng lên JTable.

Ảnh có chứa văn bản, Phông chữ, hàng, ảnh chụp màn hình

Nội dung do AI tạo ra có thể không chính xác.

**Thao tác file: Không có**

**2.4.4 Thống kế mượn sách**

**Giao diện chức năng**

Giao diện chức năng Thống kê mượn sách được xây dựng trên một tab riêng biệt (tabThongKeMuonSach.java), cung cấp cho người dùng một công cụ mạnh mẽ để phân tích và trực quan hóa dữ liệu mượn sách theo thời gian.

**Thành phần và chức năng:**

* **Bảng hiển thị:** Một JTable (tableBaoCaoSoLuong) được dùng để trình bày dữ liệu thống kê dưới dạng bảng. Các cột bao gồm "Thời gian" (có thể là ngày, tháng/năm, hoặc năm) và "Số lượng phiếu mượn" tương ứng.
* **Khu vực tùy chọn thống kê:**
  + Một JComboBox (cbDate) cho phép người dùng chọn chu kỳ thống kê: theo Ngày, Tháng, hoặc Năm.
  + Hai thành phần JDateChooser (dateStart và dateEnd) từ thư viện JCalendar cho phép người dùng chọn khoảng thời gian bắt đầu và kết thúc để giới hạn phạm vi thống kê.
* **Các nút chức năng:**
  + **Thống kê vào bảng:** Sau khi người dùng chọn chu kỳ và khoảng thời gian, nút này sẽ thực hiện truy vấn cơ sở dữ liệu và hiển thị kết quả tổng hợp lên JTable.
  + **Hiển thị biểu đồ:** Nút này lấy các tùy chọn tương tự, truy vấn dữ liệu và sau đó sử dụng thư viện **JFreeChart** để tạo một biểu đồ cột trực quan. Biểu đồ này được hiển thị trong một cửa sổ ChartFrame mới, giúp người dùng dễ dàng so sánh và nắm bắt xu hướng.
  + **Thoát:** Đóng ứng dụng.

**Mã lệnh**

* **Tương tác với thư viện ngoài:** Chức năng này tích hợp thành công với hai thư viện bên ngoài quan trọng:
  + **JCalendar:** Để cung cấp giao diện chọn ngày tháng thân thiện với người dùng.
  + **JFreeChart:** Để tạo biểu đồ thống kê chuyên nghiệp, trực quan hóa dữ liệu số lượng sách mượn theo thời gian.
* **Mô hình phân tầng và Trừu tượng hóa:**
  + Chức năng này tuân thủ mô hình phân tầng. Lớp giao diện tabThongKeMuonSach chịu trách nhiệm nhận yêu cầu từ người dùng.
  + Nó gọi các phương thức được định nghĩa trong interface IPhieuMuonCRUD (thongKeMuonSachTable và thongKeMuonSachChart).

A screenshot of a computer code

AI-generated content may be incorrect.

* + Lớp PhieuMuonCRUD thực thi các phương thức này, đóng gói logic truy vấn SQL phức tạp. Điều này thể hiện rõ Tính Trừu Tượng, vì tầng GUI không cần biết chi tiết về cách câu lệnh SQL được viết hay thực thi.
* **Xử lý dữ liệu tại tầng CRUD:**
  + Lớp PhieuMuonCRUD chứa logic để tạo các câu lệnh SQL động dựa trên chu kỳ (Ngày, Tháng, Năm) người dùng chọn.

A white background with text

AI-generated content may be incorrect.

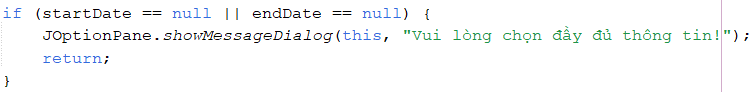
* + Các câu lệnh SQL sử dụng các hàm tổng hợp (SUM(SoLuong)) và nhóm dữ liệu (GROUP BY) để tính toán số lượng sách được mượn trong mỗi khoảng thời gian. Ví dụ, để thống kê theo tháng, câu lệnh format(NgayMuon, 'MM/yyyy') được sử dụng để nhóm các phiếu mượn theo tháng và năm.
  + Kết quả từ ResultSet được xử lý và đưa trực tiếp vào các mô hình dữ liệu (DefaultTableModel cho bảng và DefaultCategoryDataset cho biểu đồ).

A screen shot of a computer code

AI-generated content may be incorrect.

**Bắt lỗi và gom rác**

* **Cơ chế bẫy lỗi:**
  + Tại tầng GUI (tabThongKeMuonSach), chương trình kiểm tra các giá trị đầu vào trước khi xử lý, ví dụ như kiểm tra xem người dùng đã chọn ngày bắt đầu và kết thúc chưa. Nếu chưa, một JOptionPane sẽ được hiển thị để thông báo.



* + Tại tầng CRUD (PhieuMuonCRUD), toàn bộ khối mã tương tác với cơ sở dữ liệu được đặt trong một khối try-catch để bắt SQLException, đảm bảo ứng dụng không bị dừng đột ngột nếu có lỗi kết nối hoặc truy vấn.

A screenshot of a computer code

AI-generated content may be incorrect.

**Lưu trữ và Khôi phục dữ liệu**

**Giao diện chức năng**

Chức năng sao lưu và phục hồi dữ liệu được đặt trong tab "Nạp dữ liệu" (tabNapDuLieu.java), cung cấp một cơ chế an toàn để bảo vệ và khôi phục toàn bộ cơ sở dữ liệu của ứng dụng.

**Thành phần và chức năng:**

* **Tạo file backup:**
  + Một nút "Tạo file backup" (btnWriteFile) cho phép người dùng kích hoạt quá trình sao lưu.
  + Khi nhấn nút, một JFileChooser sẽ hiện ra, cho phép người dùng chọn thư mục để lưu tệp sao lưu (.bak).
  + Đường dẫn thư mục đã chọn sẽ được hiển thị trên một JTextField không cho phép chỉnh sửa (lbDuongDan) để xác nhận.
* **Khôi phục dữ liệu:**
  + Một nút "Khôi phục dữ liệu" (btnReadFile) dùng để bắt đầu quá trình phục hồi.
  + JFileChooser cũng được sử dụng để người dùng có thể duyệt và chọn một tệp .bak đã tồn tại.
  + Tên tệp và đường dẫn đầy đủ sẽ được hiển thị trên JTextField (lbFile).

**Mã lệnh**

* **Mô hình phân tầng:** Chức năng này cũng được cấu trúc theo mô hình phân tầng:
  + **Tầng GUI (tabNapDuLieu.java):** Tương tác với người dùng thông qua JFileChooser và các nút bấm.
  + **Tầng Logic (LogicBackupRestore.java):** Đóng vai trò trung gian, nhận đường dẫn từ GUI, kiểm tra tính hợp lệ cơ bản (ví dụ: đường dẫn không được rỗng), và gọi đến tầng xử lý sao lưu.
  + **Tầng Backup (Backup\_Restore.java):** Đây là tầng thấp nhất, chịu trách nhiệm xây dựng và thực thi các câu lệnh Transact-SQL (T-SQL) đặc thù của SQL Server để thực hiện sao lưu và phục hồi.
* **Thực thi lệnh SQL Server:**
  + Thay vì sử dụng các thao tác đọc/ghi tệp của Java, chương trình tận dụng tính năng sao lưu và phục hồi gốc của SQL Server để đảm bảo tính toàn vẹn và hiệu quả.
  + **Sao lưu:** Lớp Backup\_Restore thực thi câu lệnh BACKUP DATABASE QLTV TO DISK = N'...'.
  + **Phục hồi:** Lớp này thực thi một chuỗi lệnh: USE MASTER để chuyển ngữ cảnh cơ sở dữ liệu, sau đó là RESTORE DATABASE QLTV FROM DISK = N'...' WITH REPLACE để ghi đè cơ sở dữ liệu hiện tại bằng dữ liệu từ tệp sao lưu.

**Bắt lỗi và gom rác**

* **Cơ chế bẫy lỗi:**
  + Tầng Logic (LogicBackupRestore) sẽ ném ra một InvalidDataException nếu đường dẫn tệp hoặc thư mục do người dùng cung cấp là rỗng.

A black and white text

AI-generated content may be incorrect.

* + Tầng GUI (tabNapDuLieu) sử dụng khối try-catch để bắt InvalidDataException này và hiển thị thông báo lỗi cho người dùng.

A screen shot of a computer code

AI-generated content may be incorrect.

* + Tầng Backup (Backup\_Restore) cũng đặt việc thực thi các câu lệnh SQL trong khối try-catch để xử lý các ngoại lệ SQLException có thể xảy ra trong quá trình sao lưu/phục hồi, chẳng hạn như không có quyền ghi vào thư mục hoặc tệp sao lưu bị lỗi.

A screen shot of a computer code

AI-generated content may be incorrect.

**Thao tác file**

* Sử dụng JFileChooser để lấy đường dẫn tệp từ người dùng và sau đó chuyển đường dẫn này vào câu lệnh SQL để SQL Server tự thực hiện thao tác trên tệp .bak. Cách tiếp cận này hiệu quả và an toàn hơn cho việc sao lưu cơ sở dữ liệu.

A white background with black text

AI-generated content may be incorrect.

**PHẦN 3: KẾT QUẢ VÀ BÀI HỌC KINH NGHIỆM**

**3.1 Nội dung đã thực hiện**

**3.1.1 Những kiến thức và kĩ năng học được trong quá trình thực hiện đề tài**

Trong quá trình thực hiện đề tài “Xây dựng chương trình quản lý thư viện sách mini”, chúng em đã được học và áp dụng những kiến thức như là:

* Kiến thức về Java Swing: Để xây dựng giao diện người dùng (GUI) cho chương trình quản lý thư viện, nhóm chúng em đã được học và thực hành về các thành phần cơ bản của Java Swing như JPanel, JFrame, JButton, JTable, JTextField, JComboBox, và cách sử dụng chúng để tạo ra các thành phần giao diện tương tác. Chúng em cũng đã áp dụng cách xử lý sự kiện cho các thành phần này, ví dụ như xử lý sự kiện nhấn nút "Thêm" hay "Sửa".
* Kỹ năng lập trình Java: Đồ án này giúp chúng em cải thiện đáng kể kỹ năng lập trình Java. Chúng em đã học cách sử dụng các cú pháp, các lớp, các phương thức để xây dựng chương trình quản lý thư viện. Ngoài ra, kỹ năng debug và kỹ năng tìm kiếm các tài liệu liên quan cũng được cải thiện hơn trong quá trình lập trình các chức năng như thêm, sửa, xóa dữ liệu.
* Kiến thức về cấu trúc dữ liệu và cơ sở dữ liệu: Trong khi xây dựng chương trình, chúng em đã học cách sử dụng các cấu trúc dữ liệu phù hợp để lưu trữ và quản lý thông tin sách, người mượn và phiếu mượn. Đặc biệt, chúng em đã áp dụng kiến thức về cơ sở dữ liệu quan hệ (SQL Server) để thiết kế các bảng SACH, NGUOIMUON, PHIEUMUON và quản lý mối quan hệ giữa chúng, cũng như thực hiện các thao tác truy vấn (select), thêm (insert), dữ liệu thông qua JDBC. Điều đó giúp chúng em biết cách lựa chọn cấu trúc dữ liệu và mô hình cơ sở dữ liệu sao cho phù hợp với từng bài toán đề ra.
* Kỹ năng thiết kế giao diện (UI/UX): Để tạo ra một giao diện người dùng dễ sử dụng và mang tính thẩm mỹ cho chương trình quản lý thư viện, chúng em đã tìm hiểu cách thiết kế giao diện bằng việc sắp xếp khéo léo vị trí của các thành phần như nút bấm, trường nhập dữ liệu và bảng hiển thị thông tin. Việc sử dụng các JTable để hiển thị danh sách và JTextField để nhập dữ liệu là một ví dụ cụ thể.
* Kỹ năng làm việc nhóm: Để xây dựng một chương trình quản lý thư viện hoàn chỉnh, chúng em đã áp dụng các kỹ năng quản lý dự án như lập kế hoạch, phân chia công việc hợp lý và theo dõi tiến độ. Điều đó giúp chúng em tạo ra được chương trình một cách nhanh chóng và hoàn thành đúng hạn.

Qua quá trình thực hiện đề tài “Xây dựng chương trình quản lý thư viện sách mini”, chúng em đã được học và biết thêm được nhiều kỹ năng mới như kỹ năng sử dụng Java Swing, kết nối và thao tác với cơ sở dữ liệu SQL Server, sử dụng cấu trúc dữ liệu hợp lý, cách phân tích yêu cầu các bài toán nghiệp vụ của thư viện, hiểu sâu hơn về lập trình Java và cách quản lý dự án hiệu quả. Những kiến thức và kỹ năng này giúp cho chúng em có một cái nhìn sâu và vững chắc để xây dựng và phát triển các ứng dụng quản lý và cơ sở dữ liệu liên quan sau này.

**3.1.2 Chuẩn đầu ra đã đạt được trong quá trình thực hiện đề tài**

* Xây dựng thành công chương trình quản lý thư viện sử dụng Java Swing: Chương trình đã được phát triển với giao diện đồ họa thân thiện, cung cấp các tính năng quản lý nghiệp vụ cơ bản và nâng cao của thư viện.
* Quản lý dữ liệu: Chương trình cung cấp các chức năng cơ bản để thêm, sửa, xóa, và hiển thị thông tin về sách, người mượn, và phiếu mượn/trả.
* Tìm kiếm và báo cáo: Hỗ trợ tìm kiếm sách và xem các báo cáo thống kê hoạt động của thư viện.
* Giao diện thân thiện: Giao diện được thiết kế trực quan, dễ sử dụng cho người dùng.
* Áp dụng kỹ năng: Đã vận dụng hiệu quả các kỹ năng phân tích hệ thống, quản lý dự án, thiết kế và lập trình Java trong suốt quá trình thực hiện đề tài.

**3.1.3 Bài học kinh nghiệm sau khi kết thúc đề tài**

* Kỹ năng lập trình và thiết kế giao diện: Chúng em nhận thấy tầm quan trọng của việc trang bị kiến thức lập trình căn bản vững chắc, đồng thời cần tìm hiểu sâu hơn về cách thiết kế giao diện người dùng sao cho bố cục hài hòa và thân thiện với người dùng. Việc thực hành thường xuyên với các thành phần của Java Swing và việc xem xét các mẫu thiết kế UI/UX là cần thiết để tạo ra sản phẩm chất lượng.
* Quy trình phân tích và quản lý: Việc phân tích kỹ lưỡng yêu cầu của bài toán (quản lý thư viện) và quản lý dự án một cách hiệu quả là cực kỳ quan trọng để xây dựng chương trình đạt chất lượng. Chúng em đã hiểu rõ hơn tầm quan trọng của việc lên kế hoạch cụ thể, phân công công việc rõ ràng trước khi bắt đầu viết mã, điều này giúp quá trình phát triển chương trình diễn ra suôn sẻ và có hệ thống hơn.
* Kỹ năng giao tiếp và làm việc nhóm: Trong quá trình thực hiện đề tài, chúng em nhận thấy rằng khả năng giao tiếp tốt giữa các thành viên là yếu tố then chốt giúp cho quá trình làm việc nhóm hiệu quả hơn. Hơn nữa, chúng em còn học được cách lắng nghe ý kiến đóng góp, học cách trao đổi thông tin một cách rõ ràng, dễ hiểu để từ đó có thể cùng nhau đưa ra các hướng đi và giải pháp phù hợp cho các vấn đề phát sinh.

**3.2 Hướng phát triển**

Sau khi hoàn thành đề tài “Xây dựng chương trình quản lý thư viện sách mini”, chúng em nhận thấy có một số hướng có thể được phát triển trong tương lai để nâng cao chất lượng và tính năng của hệ thống:

* Cải thiện giao diện người dùng: Dù giao diện hiện tại đã đáp ứng được các yêu cầu cơ bản về thiết kế và chức năng, vẫn còn không gian để cải tiến. Chúng em có thể tùy chỉnh bố cục, thêm các hiệu ứng đồ họa hoặc sử dụng các thư viện UI/UX tiên tiến hơn để làm tăng trải nghiệm người dùng, giúp chương trình trở nên hiện đại và hấp dẫn hơn.
* Bổ sung tính năng thông báo: Tính năng thông báo có thể được thêm vào chương trình nhằm hiển thị thông tin về các sự kiện quan trọng trong thư viện, ví dụ như thông báo sách sắp đến hạn trả, sách đã quá hạn, hoặc thông báo về các cuốn sách mới về thư viện cho độc giả.
* Tích hợp cơ sở dữ liệu nâng cao và tối ưu hóa: Hiện tại, chương trình đã tích hợp cơ sở dữ liệu SQL Server. Hướng phát triển tiếp theo có thể là tối ưu hóa các truy vấn, đảm bảo tính toàn vẹn và bảo mật dữ liệu cao hơn, cũng như xây dựng các tính năng sao lưu và phục hồi dữ liệu tự động hiệu quả hơn.
* Chuyển hóa thành ứng dụng di động hoặc nền tảng web: Không chỉ giới hạn trên phiên bản Windows, một hướng phát triển khác là phát triển chương trình trên nền tảng di động (Mobile) hoặc nền tảng web. Điều này sẽ mở rộng khả năng tiếp cận của độc giả, cho phép họ tra cứu sách, xem lịch sử mượn trả từ bất cứ đâu, và có thể tích hợp thêm tính năng yêu cầu giữ sách trực tuyến.