khung3khung4khung1

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM ĐÀ NẴNG**

**KHOA TIN HỌC**

**🙟🕮🙝**

**Môn: Lập trình Java**

**Đề Tài:**



**Giảng viên hướng dẫn : Hồ Ngọc Tú**

**Sinh viên thực hiện : Ngô Đình Anh Quân**

**Nguyễn Đức Hà Nha**

**Lớp : 23CNTT2**

**Khoa : Tin học**

**Ngành : Công Nghệ Thông Tin**

**Đà Nẵng, 8 tháng 11 năm 2024**

khung2

**LỜI MỞ ĐẦU**

Ngày nay, công nghệ thông tin đã và đang làm thay đổi cách thức con người vận hành và quản lý công việc trong nhiều lĩnh vực. Đặc biệt, trong lĩnh vực thư viện, sự xuất hiện của các phần mềm quản lý đã thay thế dần phương pháp quản lý truyền thống, vốn tồn tại nhiều hạn chế như ghi chép thủ công, dễ sai sót, và tốn thời gian. Với mong muốn góp phần nâng cao hiệu quả quản lý trong hoạt động thư viện, đồ án này tập trung phát triển một phần mềm quản lý thư viện nhằm tối ưu hóa quy trình làm việc, hỗ trợ người dùng trong việc tra cứu và quản lý thông tin một cách chính xác, nhanh chóng.

Đồ án này không chỉ là một bước thực hành ứng dụng công nghệ vào thực tiễn mà còn là cơ hội để học hỏi và nâng cao kỹ năng lập trình, thiết kế hệ thống, cũng như giải quyết các vấn đề thực tế. Qua đây, nhóm thực hiện hy vọng phần mềm sẽ mang lại giá trị hữu ích và khả năng ứng dụng cao trong thực tế.

**Phần I : Khái quát chung**

**1.1/ Bối cảnh và lý do thực hiện :**

- Trong môi trường thư viện hiện đại, việc quản lý số lượng lớn tài liệu, sách, và người dùng trở thành một thách thức nếu chỉ dựa vào các phương pháp truyền thống. Các vấn đề phổ biến như mất mát thông tin, trùng lặp dữ liệu, và quá tải công việc quản lý đều cần có giải pháp công nghệ để khắc phục. Chính vì vậy, việc xây dựng một phần mềm quản lý thư viện là cần thiết, không chỉ giúp giảm thiểu sai sót mà còn cải thiện hiệu quả làm việc.

**1.2/ Mục tiêu của đồ án :**

- Mục tiêu chính của đồ án là phát triển một phần mềm quản lý thư viện với các chức năng cơ bản như:

* Quản lý thông tin sách, tài liệu.
* Quản lý tất cả thông tin sinh viên và sinh viên mượn sách
* Kiểm tra số sinh viên mượn sách quá hạn.
* Hỗ trợ mượn/trả sách và theo dõi tình trạng sách.
* Thêm/xóa sinh viên
* Thêm/xóa sách

**1.3/ Phạm vi thực hiện :**

- Phần mềm tập trung giải quyết các bài toán cơ bản liên quan đến quản lý sách và người dùng trong thư viện cỡ nhỏ đến vừa. Phần mềm được phát triển với giao diện thân thiện, dễ sử dụng, và có khả năng mở rộng trong tương lai.

**Phần II : Nội dung nghiên cứu**

**2.1/ Công nghệ sử dụng**

- Phần mềm quản lý thư viện được phát triển dựa trên các công nghệ sau**:**

1. **Ngôn ngữ lập trình :**

• Java : Được sử dụng để xây dựng toàn bộ chương trình hỗ trợ cho tính phổ biến, ổn định và khả năng xử lý hiệu quả dữ liệu.

1. **Hiển thị giao diện :**

• Console : Lập trình hoạt động trên lệnh giao diện, đảm bảo tính đơn giản và tối ưu để phát triển các chức năng quản lý cơ bản.

1. **Cơ sở dữ liệu :**

**•** Dữ liệu được lưu trữ trong các tệp hoặc sử dụng cơ sở dữ liệu cấu trúc như danh sách, mảng (mảng) hoặc bản đồ (hashmap) trong Java.

1. **Công cụ phát triển :**

• VS Code : Được sử dụng để phát triển và quản lý mã nguồn.

**2.2/ Tổng quan về các chức năng**

- Phần mềm quản lý thư viện trên console bao gồm các chức năng chính sau:

1. **Xem tất cả thông tin có trong thư viện** :

• Hiển thị danh sách đầy đủ các hiện có trong thư viện, bao gồm thông tin như mã sách, tên sách, tác giả và trạng thái (có thể vay hoặc có sẵn).

1. **Xem thông tin của tất cả sinh viên** :

**•** Hiển thị danh sách các sinh viên đã đăng ký trong hệ thống, bao gồm các sinh viên mã hóa, tên sinh viên và số mượn (nếu có).

1. **Xem thông tin mượn của sinh viên** :

**•** Nghiên cứu chi tiết về sách mượn của từng thành viên, bao gồm mã sách, tên sách, ngày mượn và hạn mức trả.

1. **Kiểm tra số lượng sinh viên mượn đã quá giới hạn** :

• Hỗ trợ kiểm tra và liệt kê danh sách sinh viên hiện đang được xác định theo thời hạn, bao gồm thông tin chi tiết về hạn ngạch và ngày quá hạn.

1. **Thêm/xóa sinh viên** :

**•** Chức năng thêm mới thành viên sinh viên vào hệ thống hoặc xóa thông tin sinh viên khỏi danh sách quản lý.

1. **Thêm/xóa sách** :

**•** Hỗ trợ thêm danh sách mới vào thư viện hoặc xóa danh sách còn lại trong hệ thống.

1. **Cho sinh viên mượn sách** :

**•** Đăng ký mượn sách cho sinh viên, cập nhật trạng thái và ghi nhận ngày vay, hạn mức.

1. **Sinh viên trả lương** :

• Xác minh danh sách thanh toán của sinh viên, cập nhật lại trạng thái trong thư viện và xóa thông tin vay tương ứng.

1. **Thoát** :

• Kết thúc chương trình và lưu trữ dữ liệu (nếu có yêu cầu).

**2.3/ Thiết kế hệ thống**

**A screenshot of a computer

Description automatically generated**

**1. Model (Mô hình dữ liệu)**

- Bao dữ liệu chính của mô hình bao gồm:

**a) Account(Tài khoản)**

- Mục tiêu : Quản lý thông tin đăng nhập tài khoản của người dùng (thư viện nhân viên).

- Thuộc tính :

+ username: Tên đăng nhập.

+ password: Mật khẩu.

- Phương pháp : Kiểm tra thông tin đăng nhập.

**b) Book(Sách)**

- Mục tiêu : Quản lý danh sách thông tin trong thư viện.

- Thuộc tính :

+ id: Mã duy nhất.

+ title: Tên sách.

+ author: Tên tác giả.

+ publisher: Nhà xuất bản.

+ publicationYear: Năm xuất bản.

+ language: Ngôn ngữ của sách.

+ genre: Thể loại sách.

+ status: Trạng thái (Có sẵn, Vay mượn).

- Phương pháp :

+ Thêm sách.

+ Update information.

**c) Student(Sinh viên)**

- Mục đích : Quản lý thông tin sinh viên.

- Thuộc tính :

+ id: Mã sinh viên.

+ name: Tên sinh viên.

+ birthday: Ngày sinh.

+ gender: Giới tính.

+ phone: Số điện thoại liên lạc.

+ department: Khoa/Bộ phận quản lý sinh viên.

+ email: Địa chỉ email của sinh viên.

+ borrowBooks: Danh sách các sách mà sinh viên mượn (ID list).

+ borrowLimit: Số lượng tối đa mà sinh viên có thể vay.

- Phương pháp :

+ addBorrowedBook(String bookId): Add list vào danh sách mượn. Kiểm tra hạn mức vay trước khi bổ sung.

+ toString(): Hiển thị đầy đủ thông tin sinh viên dưới dạng chuỗi.

**2. DAO (Đối tượng truy cập dữ liệu)**

- Các lớp DAO cam chịu tương tác với dữ liệu, bao gồm việc đọc, ghi và cập nhật dữ liệu từ/đến nguồn lưu trữ.

**a) Lớp StudentDao**

- Mục đích: Quản lý dữ liệu sinh viên.

- Phương thức chính

+ add(Student student): Thêm một sinh viên mới vào danh sách.

+ update(List<Student> updatedStudents): Cập nhật toàn bộ danh sách sinh viên với danh sách mới.

+ read(): Trả về danh sách sinh viên hiện tại.

**b) Lớp BookDao**

- Mục đích: Quản lý dữ liệu sách.

- Phương thức chính

+ add(Book book): Thêm sách mới vào danh sách.

+ update(List<Book> updatedBooks): Cập nhật danh sách sách mới.

+ read(): Trả về danh sách sách hiện tại.

**3. Views (Giao diện người dùng)**

- Phần view hiển thị giao diện cơ bản trên console, giúp người dùng tương tác với hệ thống.

**a) LoginView**

- getAccount(): Lấy thông tin tài khoản từ người dùng (tên đăng nhập và mật khẩu).

- checkEmpty(): Kiểm tra xem chuỗi nhập vào có trống hay không (trả về true nếu trống).

**b) HomeView**

- showBook(List<Book> books): Hiển thị danh sách sách từ tham số đầu vào.

- showStudent(List<Student> students): Hiển thị danh sách sinh viên từ tham số đầu vào.

- getBook(): Lấy thông tin sách từ người dùng thông qua nhập liệu.

- getStudent(): Lấy thông tin sinh viên từ người dùng thông qua nhập liệu.

- searchStudent(): Yêu cầu người dùng nhập mã sinh viên để tìm kiếm.

- searchBook(): Yêu cầu người dùng nhập ID sách để tìm kiếm.

**4. Control (Điều khiển)**

- Các lớp điều khiển xử lý logic và luồng dữ liệu giữa các lớp Model, DAO, và View.

**a) LoginControl**

- checkAccount(Account account): Xác thực tài khoản dựa trên tên đăng nhập và mật khẩu. Giả định có một tài khoản hợp lệ (username: admin, password: 12345).

**2. HomeControl**

- Danh sách thuộc tính:

+ books: Danh sách các sách trong thư viện.

+ students: Danh sách sinh viên.

- Phương thức chính:

+ updateBook(List<Book>): Cập nhật danh sách sách từ dữ liệu mới.

+ updateStudent(List<Student>): Cập nhật danh sách sinh viên từ dữ liệu mới.

+ addBook(Book): Thêm sách mới vào danh sách.

+ addStudent(Student): Thêm sinh viên mới vào danh sách.

+ searchBook(Book): Tìm kiếm sách dựa trên ID.

+ searchStudent(Student): Tìm kiếm sinh viên dựa trên ID.

+ checkOverTime(): Kiểm tra sinh viên mượn sách quá hạn (so sánh borrowLimit với số sách đã mượn).

**2.4/ Phân tích chức năng**

**1. Componet ( Chức năng quản lý hệ thống )**