|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| BAN CƠ YẾU CHÍNH PHỦ  **HỌC VIỆN KỸ THUẬT MẬT MÃ**  -----**🙦🕮🙤**-----  A red circle with white text and a book and key  Description automatically generated    **ĐỒ ÁN LẬP TRÌNH WEB**  ĐỀ TÀI:  **XÂY DỰNG WEB BÁN SÁCH**   |  |  | | --- | --- | | Giảng viên hướng dẫn | : **Trần Đắc Tốt** | | Học viên thực hiện | : **Hà Thị Thanh Dung**  **Trần Thanh Hải**  **Dương Chí Duy** | | Lớp | : CT06N | | Khóa | : 6 | | Chuyên ngành | : Công nghệ thông tin |   **TP. Hồ Chí Minh, 2024** |

**LỜI CẢM ƠN**

Trước tiên với tình cảm sâu sắc và chân thành nhất, cho phép chúng em được bày tỏ lòng biết ơn đến tất cả các cá nhân và tổ chức đã tạo điều kiện hỗ trợ, giúp đỡ chúng em trong suốt quá trình học tập và nghiên cứu đề tài này. Trong suốt thời gian từ khi bắt đầu học tập tại trường đến nay,chúng em đã nhân được rất nhiều sự quan tâm, giúp đỡ của quý Thầy Cô và bạn bè. Với lòng biết ơn sâu sắc nhất,chúng em xin gửi đến quý Thầy Cô ở Khoa Công Nghệ Thông Tin đã truyền đạt vốn kiến thức quý báu cho chúng em trong suốt thời gian học tập tại trường. Nhờ có những lời hướng dẫn, dạy bảo của các thấy cô nên đề tài nghiên cứu của chúng em mới có thể hoàn thiện tốt đẹp.

Một lần nữa, chúng em xin chân thành cảm ơn Thầy Trần Đắc Tốt - người đã trực tiếp giúp đỡ, quan tâm, hướng dẫn chúng em hoàn thành tốt bài báo cáo này trong thời gian qua

Bản báo cáo đồ án thực hiên trong khoảng thời gian vừa qua. Bước đầu đi vào thực tế của chúng em còn hạn chế và còn nhiều bỡ ngỡ nên không tránh khỏi những thiếu sót, chúng em rất mong nhận được những ý kiến đóng góp quý báu của quý Thầy Cô để kiến thức của chúng em trong lĩnh vực này được hoàn thiện hơn đồng thời có điều kiên bổ sung, nâng cao ý thức của mình.

Em xin chân thành cảm ơn!

**LỜI MỞ ĐẦU**

Ngày nay với sự bùng nổ và phát triển mạnh mẽ của lĩnh vực Công nghệ thông tin đặc biệt với sự ra đời của Internet, nó đang dần trở thành một phần không thể thiếu của cuộc sống con người. Công nghệ thông tin đang được phát triển và ứng dụng mạnh mẽ trong các lĩnh vực mua bán, y tế, học tập, trí thức, cùng với lẽ đó chúng em phát triển ra đồ án này.

Trong bối cảnh phát triển nhanh chóng của công nghệ thông tin, việc xây dựng website bán sách không chỉ mang lại sự thuận tiện cho người tiêu dùng mà còn tối ưu hóa quy trình kinh doanh cho các nhà xuất bản và nhà sách. Website bán sách là một ứng dụng thực tế giúp kết nối người đọc với kho tàng tri thức phong phú, đồng thời tạo ra trải nghiệm mua sắm trực tuyến tối ưu. Qua việc delving vào xu hướng tiêu dùng hiện đại, đồ án này sẽ khám phá các công nghệ và giải pháp giúp nâng cao hiệu quả kinh doanh trong ngành sách

Đáp ứng những nhu cầu đó,chúng em đã chọn đề tài “Xây dựng web bán sách” để xây dựng và phát triển. Web được thiết kế dựa trên thực tế nghiệp vụ của cửa hàng để có khả năng đáp ứng và hỗ trợ tốt trong công việc quản lý và kinh doanh của cửa hàng, tránh nhầm lẫn sai sót, đồng thời hỗ trợ cho việc tìm kiếm thông tin nhanh chóng, chính xác.

**MỤC LỤC**

**[LỜI CẢM ƠN](#_Toc178789492)** [2](#_Toc178789492)

**[DANH MỤC BẢNG BIỂU](#_Toc178789493)** [6](#_Toc178789493)

**[CHƯƠNG 1: TỔNG QUAN](#_Toc178789494)** [7](#_Toc178789494)

**[1.1](#_Toc178789495)****[Lý do chọn đề tài:](#_Toc178789495)** [7](#_Toc178789495)

**[1.2](#_Toc178789496)****[Giới thiệu chung:](#_Toc178789496)** [7](#_Toc178789496)

**[1.3](#_Toc178789497)****[Phạm vi hệ thống:](#_Toc178789497)** [10](#_Toc178789497)

**[1.4](#_Toc178789498)****[Giới thiệu ngôn ngữ và công cụ:](#_Toc178789498)** [10](#_Toc178789498)

**[1.4.1 Ngôn ngữ java](#_Toc178789499)** [10](#_Toc178789499)

**[1.4.2 MySQL Workbench](#_Toc178789500)** [10](#_Toc178789500)

**[1.4.3 Visual Studio](#_Toc178789501)** [11](#_Toc178789501)

**[1.4.4 Spring Boot Framework:](#_Toc178789502)** [12](#_Toc178789502)

**[1.4.5 Thymeleaf](#_Toc178789503)** [12](#_Toc178789503)

**[CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG](#_Toc178789504)** [13](#_Toc178789504)

**[2.1 Danh sách tác nhân](#_Toc178789505)** [13](#_Toc178789505)

**[2.2 Danh sách Use case](#_Toc178789506)** [15](#_Toc178789506)

**[2.3 Mô hình use case:](#_Toc178789507)** [18](#_Toc178789507)

**[2.3.1 Sơ đồ use-case chính:](#_Toc178789508)** [18](#_Toc178789508)

**[2.3.2 Sơ đồ UseCase Admin](#_Toc178789509)** [19](#_Toc178789509)

**[2.3.3 Sơ đồ UseCase Khách hàng](#_Toc178789510)** [20](#_Toc178789510)

**[2.4 Đặc tả chi tiết các Usecase(UC)](#_Toc178789511)** [20](#_Toc178789511)

[2.4.1 Use Case Đăng nhập hệ thống 20](#_Toc178789512)

[2.4.2 Use Case Đăng xuất hệ thống 22](#_Toc178789513)

[2.4.3 Use Case Đăng kí tài khoản 22](#_Toc178789514)

[2.4.4 Use Case Quản lý sách 24](#_Toc178789515)

[2.4.5 Use Case Thêm, Sửa, Thanh toán giỏ hàng 30](#_Toc178789516)

**[CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ CƠ SỞ DỮ LIỆU](#_Toc178789517)** [36](#_Toc178789517)

**[3.1](#_Toc178789518)****[Biểu đồ diagram:](#_Toc178789518)** [36](#_Toc178789518)

**[3.2](#_Toc178789519)****[Bảng book:](#_Toc178789519)** [39](#_Toc178789519)

**[3.3](#_Toc178789520)****[Bảng authors:](#_Toc178789520)** [40](#_Toc178789520)

**[3.4](#_Toc178789521)****[Bảng categories:](#_Toc178789521)** [41](#_Toc178789521)

**[3.5](#_Toc178789522)****[Bảng comments:](#_Toc178789522)** [41](#_Toc178789522)

**[3.6](#_Toc178789523)****[Bảng companies:](#_Toc178789523)** [42](#_Toc178789523)

**[3.7](#_Toc178789524)****[Bảng customers:](#_Toc178789524)** [42](#_Toc178789524)

**[3.8](#_Toc178789525)****[Bảng hibernate\_sequence:](#_Toc178789525)** [43](#_Toc178789525)

**[3.9](#_Toc178789526)****[Bảng migrations:](#_Toc178789526)** [43](#_Toc178789526)

**[3.10](#_Toc178789527)****[Bảng order\_details:](#_Toc178789527)** [44](#_Toc178789527)

**[3.11](#_Toc178789528)****[Bảng orders:](#_Toc178789528)** [44](#_Toc178789528)

**[3.12](#_Toc178789529)****[Bảng password\_resets:](#_Toc178789529)** [44](#_Toc178789529)

**[3.13](#_Toc178789530)****[Bảng payments:](#_Toc178789530)** [45](#_Toc178789530)

**[3.14](#_Toc178789531)****[Bảng roles:](#_Toc178789531)** [45](#_Toc178789531)

**[3.15](#_Toc178789532)****[Bảng saves:](#_Toc178789532)** [45](#_Toc178789532)

**[3.16](#_Toc178789533)****[Bảng verificationtoken:](#_Toc178789533)** [46](#_Toc178789533)

**[CHƯƠNG 4: CÀI ĐẶT VÀ THỰC NGHIỆM](#_Toc178789534)** [46](#_Toc178789534)

**[KẾT LUẬN](#_Toc178789535)** [48](#_Toc178789535)

**[TÀI LIỆU THAM KHẢO](#_Toc178789536)** [49](#_Toc178789536)

# **DANH MỤC BẢNG BIỂU**

**CHƯƠNG 1: TỔNG QUAN**

* 1. **Lý do chọn đề tài:**

Lời Mở Đầu

Trong thời đại số hóa ngày nay, công nghệ thông tin đã trở thành một phần thiết yếu trong mọi lĩnh vực, đặc biệt là trong ngành bán lẻ. Ngành sách, với sứ mệnh truyền đạt tri thức và văn hóa, cũng không nằm ngoài xu hướng này. Việc xây dựng một website bán sách không chỉ đáp ứng nhu cầu mua sắm tiện lợi của người tiêu dùng mà còn giúp các nhà xuất bản và nhà sách tối ưu hóa quy trình quản lý và phân phối sản phẩm.

Đồ án này nhằm mục tiêu thiết kế và phát triển một website bán sách với các tính năng hiện đại, dễ sử dụng, và thân thiện với người dùng. Chúng tôi sẽ tập trung vào việc cải thiện trải nghiệm khách hàng thông qua giao diện trực quan, hệ thống tìm kiếm thông minh, và các phương thức thanh toán đa dạng. Đồng thời, đồ án cũng sẽ đề cập đến những thách thức và giải pháp trong việc vận hành và quản lý một nền tảng thương mại điện tử hiệu quả.

Hy vọng rằng sản phẩm cuối cùng sẽ không chỉ là một công cụ mua sắm mà còn là cầu nối giữa độc giả và kho tàng tri thức vô hạn.

* 1. **Giới thiệu chung:**

Định nghĩa: Quản lý bán hàng có thể được định nghĩa là hoạt động quản trị của cá nhân hoặc một nhóm người thuộc lĩnh vực bán hàng hoặc những người hỗ trợ trực tiếp cho lực lượng bán hàng nhằm triển khai việc cung cấp các sản phẩm hoặc dịch vụ của doanh nghiệp trên cơ sở chiến lược kinh doanh, nguồn lực của doanh nghiệp và môi trường kinh doanh có liên quan.

Quản lý bán hàng có nhiệm vụ phân phối hàng hóa đến những kênh phân phối theo mục tiêu đã đề ra nhằm tối ưu việc hiện diện của hàng hóa, dịch vụ với chi phí hợp lý, đạt hiệu quả tối đa và đạt được mục tiêu marketing của nhà hàng về ngắn hạn cũng như dài hạn.

Chức năng của hệ thống quản lý bán hàng:

Trong thời đại bùng nổ công nghệ 4.0 như hiện nay, một hệ thống giúp quản lý việc bán hàng là không thể thiếu trong hoạt động kinh doanh của doanh nghiệp. Hệ thống quản lý chính là những phần mềm quản lý bán hàng, giúp cho người dùng đặc biệt là các doanh nghiệp kiểm soát quản lý được hàng hóa một cách chặt chẽ hơn. Việc quản lý của hệ thống sẽ được kiểm soát trong tất cả các khâu: Nhập hàng, quản lý đơn hàng, quản lý kho hàng, quản lý nhân viên, chăm sóc khách hàng....

Có một hệ thống giúp quản lý bán hàng sẽ giúp doanh nghiệp tối ưu được hoạt động kinh doanh của mình, cụ thể thông qua những chức năng của hệ thống quản lý bán hàng.

Chiến lược bán hàng:

Đây là kế hoạch tổng thể nhằm đạt được mục tiêu doanh thu và mở rộng thị trường. Chiến lược bao gồm các phương pháp tiếp cận khách hàng, định giá sản phẩm, và các hoạt động khuyến mãi.

Quản lý đội ngũ bán hàng:

Điều này liên quan đến việc tuyển dụng, đào tạo, động viên và đánh giá hiệu suất của nhân viên bán hàng. Một đội ngũ bán hàng hiệu quả có thể tạo ra sự khác biệt lớn trong việc đạt được mục tiêu doanh thu.

Dự báo bán hàng:

Quá trình dự đoán nhu cầu sản phẩm hoặc dịch vụ trong tương lai dựa trên các dữ liệu lịch sử và xu hướng thị trường. Dự báo chính xác giúp lập kế hoạch sản xuất và quản lý hàng tồn kho hiệu quả.

Quản lý quan hệ khách hàng (CRM):

Hệ thống và quy trình giúp quản lý và phân tích tương tác với khách hàng, từ đó cải thiện sự hài lòng và tăng cường mối quan hệ lâu dài với khách hàng.

Kênh phân phối:

Các phương tiện hoặc cách thức mà sản phẩm hoặc dịch vụ được đưa đến tay khách hàng. Quản lý kênh phân phối bao gồm việc lựa chọn và hợp tác với các nhà phân phối, đại lý và bán lẻ.

Phân tích dữ liệu bán hàng:

Sử dụng dữ liệu để phân tích xu hướng bán hàng, hành vi của khách hàng, và hiệu suất của đội ngũ bán hàng. Phân tích này giúp đưa ra quyết định dựa trên dữ liệu thay vì cảm tính.

Chiến lược giá cả:

Xác định giá bán của sản phẩm hoặc dịch vụ dựa trên các yếu tố như chi phí sản xuất, giá thị trường, và giá của đối thủ cạnh tranh. Giá cả là yếu tố quan trọng ảnh hưởng đến doanh thu và lợi nhuận.

Quản lý hàng tồn kho:

Quá trình theo dõi và quản lý lượng hàng tồn kho để đảm bảo rằng có đủ sản phẩm để đáp ứng nhu cầu của khách hàng mà không gây ra tình trạng thừa hoặc thiếu hàng.

Khuyến mãi và chiết khấu:

Các chương trình khuyến mãi và chiết khấu giúp thúc đẩy doanh số bán hàng và thu hút khách hàng mới. Chúng bao gồm giảm giá, tặng quà, và các chiến lược khuyến mãi khác.

Ứng dụng quản lý quán ăn cần thực hiện các chức năng cơ bản sau:

• Quản lý thực đơn: Thêm, sửa, xóa , tìm kiếm các món ăn và đồ uống.

• Quản lý đơn hàng: Tạo, chỉnh sửa và theo dõi các đơn hàng.

• Quản lý nhân viên: Thêm, sửa, xóa thông tin nhân viên và quản lý phân quyền.

• Quản lý hóa đơn: Xem và quản lý hóa đơn thanh toán của khách hàng.

• Quản lý bàn ăn: Cập nhật tình trạng bàn ăn (trống, đã đặt, đang phục vụ). Chuyển bàn

• Báo cáo doanh thu: Xem các báo cáo về doanh thu hàng ngày, hàng tuần, và hàng tháng.

• Thanh toán bằng Paypal

* 1. **Phạm vi hệ thống:**

Phạm vi nghiên cứu chỉ áp dụng tại địa bàn một cửa hàng nhất định, chưa mở rộng đến chuỗi cửa hàng, do đó các kết quả tìm được có thể chỉ hữu ích trong một phạm vi nhất định, không có ý nghĩa trên chuỗi cửa hàng và không bao quát rộng, bởi những cửa hàng khác nhau sẽ có mức thu nhập và mua hàng khác nhau. Nếu có cơ hội và điều kiện để mở rộng và nghiên cứu dự án thì dự án sẽ mang lại nhiều lợi ích cho doanh nghiệp và đặc biệt trong lĩnh vực thương mại điện tử, mang lại những trải nghiệm mới và tiện ích cho khách hàng.

* 1. **Giới thiệu ngôn ngữ và công cụ:**

### **1.4.1 Ngôn ngữ java**

Java là một ngôn ngữ lập trình hướng đối tượng, được phát triển bởi Sun Microsystems vào giữa những năm 1990. Với cú pháp đơn giản và khả năng tương thích cao, Java cho phép lập trình viên viết ứng dụng một lần và chạy trên nhiều nền tảng khác nhau nhờ vào công nghệ Java Virtual Machine (JVM). Ngôn ngữ này nổi bật với tính an toàn, quản lý bộ nhớ tự động và thư viện phong phú, làm cho nó trở thành lựa chọn phổ biến trong phát triển ứng dụng web, di động và doanh nghiệp. Java cũng hỗ trợ lập trình đa luồng, giúp tối ưu hóa hiệu suất cho các ứng dụng phức tạp.

### **1.4.2 MySQL Workbench**

MySQL Workbench là một công cụ phát triển tích hợp cho quản lý cơ sở dữ liệu MySQL, cung cấp giao diện đồ họa thân thiện cho người dùng. Nó cho phép thiết kế, mô hình hóa, và quản lý cơ sở dữ liệu dễ dàng thông qua các tính năng như tạo và chỉnh sửa bảng, thực hiện truy vấn SQL, và quản lý người dùng. MySQL Workbench cũng hỗ trợ các chức năng sao lưu, phục hồi và tối ưu hóa hiệu suất cơ sở dữ liệu. Ngoài ra, nó còn cung cấp các công cụ để phân tích và trực quan hóa dữ liệu, giúp lập trình viên và quản trị viên dễ dàng theo dõi và quản lý hệ thống cơ sở dữ liệu.

### **1.4.3 Visual Studio**

Visual Studio là một môi trường phát triển tích hợp (IDE) do Microsoft phát triển. Nó cung cấp một không gian làm việc toàn diện cho các lập trình viên, giúp họ viết, kiểm tra và triển khai các ứng dụng. Visual Studio hỗ trợ nhiều ngôn ngữ lập trình khác nhau như C++, C#, Visual Basic, Python, JavaScript,... và được sử dụng rộng rãi để phát triển các ứng dụng Windows, ứng dụng web, ứng dụng di động và các dịch vụ đám mây.

Lợi ích việc sử dụng Visual Studio:

* Giao diện trực quan: Visual Studio có giao diện người dùng thân thiện, dễ sử dụng, giúp các lập trình viên tập trung vào việc viết mã.
* Tính năng phong phú:
* IntelliSense: Tự động hoàn thành mã, cung cấp gợi ý về các từ khóa, hàm, biến,... giúp tăng tốc độ viết mã.
* Debugger: Công cụ gỡ lỗi mạnh mẽ, giúp tìm và sửa lỗi trong mã một cách hiệu quả.
* Refactoring: Tổ chức lại mã một cách tự động, giúp cải thiện chất lượng mã.
* Git Integration: Tích hợp với hệ thống quản lý phiên bản Git, giúp quản lý mã nguồn một cách hiệu quả.
* Hỗ trợ đa nền tảng: Visual Studio có thể được sử dụng để phát triển ứng dụng cho nhiều nền tảng khác nhau như Windows, Android, iOS, web,...
* Cộng đồng lớn: Có một cộng đồng người dùng lớn và sôi động, giúp bạn dễ dàng tìm kiếm tài liệu, giải đáp thắc mắc và chia sẻ kinh nghiệm.

### **1.4.4 Spring Boot Framework:**

Spring Boot là một framework được phát triển dựa trên Spring Framework, nhằm giúp xây dựng các ứng dụng Java một cách nhanh chóng và dễ dàng hơn. Spring Boot giúp giảm bớt các cấu hình phức tạp và rườm rà mà các lập trình viên thường gặp phải khi sử dụng Spring Framework thông thường. Một số điểm nổi bật của Spring Boot bao gồm:

1. **Tự động cấu hình (Auto-Configuration):** Spring Boot tự động cấu hình các thành phần của ứng dụng dựa trên các dependency có trong project, giảm thiểu công việc cấu hình thủ công cho lập trình viên.
2. **Embedded Server (Máy chủ tích hợp):** Spring Boot tích hợp các máy chủ như Tomcat, Jetty, hoặc Undertow trực tiếp vào ứng dụng. Điều này cho phép ứng dụng có thể chạy ngay mà không cần phải cấu hình riêng biệt các máy chủ web.
3. **Starter POMs:** Spring Boot cung cấp các gói "starter" để dễ dàng tích hợp các tính năng thường dùng (như Spring Web, Spring Data, Spring Security...) mà không cần phải cấu hình từng dependency riêng lẻ.
4. **Spring Boot CLI:** Công cụ dòng lệnh cho phép bạn phát triển ứng dụng Spring Boot bằng cách sử dụng Groovy, giúp tạo ứng dụng một cách nhanh chóng và tiện lợi.
5. **Spring Initializr:** Một công cụ trực tuyến cho phép bạn tạo cấu trúc ứng dụng Spring Boot cơ bản với các dependency cần thiết mà không cần viết nhiều code.
6. **Actuator:** Spring Boot cung cấp một bộ công cụ giúp bạn giám sát và quản lý ứng dụng, chẳng hạn như kiểm tra trạng thái, theo dõi số liệu, và quản lý cấu hình.
7. **Microservices Support:** Spring Boot được thiết kế để hỗ trợ phát triển các ứng dụng microservices, với khả năng mở rộng linh hoạt và dễ dàng tích hợp các công cụ hỗ trợ như Docker, Kubernetes.

### **1.4.5 Thymeleaf**

Thymeleaf là một thư viện mẫu (template engine) cho Java, thường được sử dụng trong các ứng dụng web. Nó cho phép tạo ra các trang HTML động bằng cách kết hợp dữ liệu từ server với các mẫu HTML. Dưới đây là một số điểm nổi bật về Thymeleaf:

**Tính năng chính**

* **Mẫu HTML tự nhiên**: Thymeleaf cho phép bạn tạo các mẫu HTML có thể mở và chỉnh sửa dễ dàng trong các trình soạn thảo HTML mà không cần phải xử lý các phần tử đặc biệt.
* **Biểu thức EL (Expression Language)**: Hỗ trợ các biểu thức để truy cập và hiển thị dữ liệu từ model.
* **Hỗ trợ đa dạng các loại tài liệu**: Ngoài HTML, Thymeleaf còn hỗ trợ các định dạng như XML, XHTML và thậm chí là văn bản thuần túy.

**Tính năng nâng cao**

Conditional rendering: Có thể hiển thị hoặc ẩn các phần tử HTML dựa trên các điều kiện.

Iteration: Có thể lặp qua các danh sách và tạo các phần tử HTML động.

Fragment: Hỗ trợ tái sử dụng các phần của mẫu HTML, giúp giảm thiểu mã lặp.

**Tài liệu và cộng đồng**

Thymeleaf có tài liệu chi tiết và cộng đồng người dùng rộng rãi, giúp người dùng dễ dàng tìm kiếm thông tin và hỗ trợ.

# **CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG**

## **2.1 Danh sách tác nhân**

|  |  |
| --- | --- |
| **Tác nhân** | **Mô tả** |
| Nguời dùng (User) | Người truy cập trang web có thể là khách hàng hoặc người dùng ẩn danh.   * ***Khách hàng đã đăng ký (Registered Customer):*** Người dùng có tài khoản, có thể đặt mua sách, xem lịch sử mua hàng, quản lý giỏ hàng, và cập nhật thông tin cá nhân. * ***Khách hàng không đăng ký (Guest):*** Người dùng không có tài khoản, chỉ có thể tìm kiếm và xem sách mà không cần đăng nhập. |
| Quản trị viên (Admin) | Admin (Quản trị viên) đóng vai trò quan trọng trong việc giám sát và quản lý toàn bộ hệ thống |
| Hệ thống thanh toán (Payment Gateway): | Đây là tác nhân trung gian giữa website bán sách và ngân hàng. Tác nhân này chịu trách nhiệm xử lý các giao dịch thanh toán trực tuyến (bằng thẻ tín dụng, ví điện tử, chuyển khoản ngân hàng, v.v.). |

**1. Người dùng (Customer)**

* **Đăng ký tài khoản**: Người dùng có thể tạo một tài khoản mới để mua sách và theo dõi đơn hàng.
* **Đăng nhập**: Người dùng có thể đăng nhập vào hệ thống bằng tài khoản đã đăng ký.
* **Đăng xuất**: Người dùng có thể đăng xuất khỏi hệ thống
* **Tìm kiếm sách**: Người dùng có thể tìm kiếm sách theo tên, tác giả, thể loại hoặc từ khóa.
* **Xem chi tiết sách**: Người dùng có thể xem thông tin chi tiết về một cuốn sách, bao gồm mô tả, đánh giá và giá cả.
* **Thêm sách vào giỏ hàng**: Người dùng có thể thêm sách vào giỏ hàng để mua sau.
* **Xem giỏ hàng**: Người dùng có thể xem các sách đã thêm vào giỏ hàng.
* **Xóa sản phẩm trong giỏ hàng:** Người dùng có thể xóa sản phẩm đã thêm trong giỏ hàng
* **Thanh toán đơn hàng**: Người dùng có thể tiến hành thanh toán cho các sách trong giỏ hàng.
* **Xem chi tiết đơn hàng:** Người dùngcó thể xem chi tiết của một đơn hàng đã đặt, bao gồm thông tin sản phẩm, trạng thái đơn hàng, và các chi tiết thanh toán.
* **Xem lịch sử đơn hàng**: Người dùng có thể xem các đơn hàng đã đặt trong quá khứ.

**2. Quản trị viên (Administrator)**

* **Đăng nhập/Đăng xuất**: Admin có thể đăng nhập và đăng xuất khỏi hệ thống.
* **Quản lý sách**: Admin có thể thêm, chỉnh sửa hoặc xóa sách trong hệ thống.
* **Quản lý thể loại sách**: Quản lí các thể loại sách trong hệ thống gồm xóa, sửa, thêm danh mục
* **Quản lý tác giả**: Quản lí các tác giả trong hệ thống
* **Quản lý đơn hàng**: Admin có thể xem và cập nhật trạng thái của các đơn hàng.

**3. Hệ thống thanh toán (Payment Gateway)**

* **Xử lý giao dịch thanh toán**: Hệ thống thanh toán thực hiện xử lý giao dịch khi người dùng thanh toán đơn hàng.

.

## **2.2 Danh sách Use case**

|  |  |  |
| --- | --- | --- |
| Mã số | Tên Usecase | Mô tả |
| UC1 | Đăng nhập | Đăng nhập vào hệ thống |
| UC2 | Đăng xuất | Đăng xuất tài khoản |
| UC3 | Đăng ký tài khoản | Đăng kí tài khoản mới |
| UC4 | Quản lí sách | Quản lý thông tin sách trong hệ thống, bao gồm thêm mới, sửa đổi, xóa và tìm kiếm sách. |
| UC4.1 | Thêm sách | Người quản lý (admin) có thể thêm sách mới vào hệ thống, bao gồm các thông tin như tên sách, tác giả, thể loại, năm xuất bản, |
| UC4.2 | Sửa thông tin sách | Người quản lý (admin) có thể sửa thông sách trong hệ thống, bao gồm các thông tin như tên sách, tác giả, thể loại, năm xuất bản, |
| UC4.3 | Xóa sách | Người quản lý (admin) có thể xóa đã không còn kinh doanh trên hệ thống |
| UC5 | Quản lí thể loại sách | Quản lí các thể loại sách trong hệ thống gồm xóa, sửa, thêm danh mục |
| UC5.1 | Thêm thể loại sách | Người quản lý (admin) có thể thêm thể loại sách mới vào hệ thống |
| UC5.2 | Sửa thông tin thể loại sách | Người quản lý (admin) có thể sửa thông thể loại sách trong hệ thống |
| UC5.3 | Xóa thể loại sách | Người quản lý (admin) có thể xóa thể loại sách ra khỏi hệ thống |
| UC6 | Quản lí tác giả | Quản lí các tác giả trong hệ thống gồm xóa, sửa, thêm tác giả |
| UC6.1 | Thêm tác giả | Người quản lý (admin) có thể thêm tác giả mới vào hệ thống |
| UC6.2 | Sửa thông tin tác giả | Người quản lý (admin) có thể sửa thông tác giả trong hệ thống |
| UC6.3 | Xóa tác giả | Người quản lý (admin) có thể xóa tác giả ra khỏi hệ thống |
| UC7 | Quản lí dơn hàng | Quản lý các đơn hàng trong hệ thống |
| UC7.1 | Xem danh sách đơn đặt hàng | Người quản lý (admin) có thể xem danh sách các đơn đặt hàng |
| UC8 | Tìm kiếm sách | Người dùng có thể tìm kiếm sách theo tên, tác giả, thể loại hoặc từ khóa. |
| UC9 | Xem chi tiết sách | Người dùng có thể xem thông tin chi tiết về một cuốn sách, bao gồm mô tả, đánh giá và giá cả. |
| UC10 | Thêm sách vào giỏ hàng | Người dùng có thể thêm sách vào giỏ hàng để mua sau. |
| UC11 | Xem giỏ hàng | Người dùng có thể xem các sách đã thêm vào giỏ hàng. |
| UC12 | Xóa sản phẩm trong giỏ hàng | Người dùng có thể xóa sản phẩm đã thêm trong giỏ hàng |
| UC12 | Thanh toán đơn hàng | Người dùng có thể tiến hành thanh toán cho các sách trong giỏ hàng. |
| UC13 | Xem chi tiết đơn hàng | Người dùngcó thể xem chi tiết của một đơn hàng đã đặt, bao gồm thông tin sản phẩm, trạng thái đơn hàng, và các chi tiết thanh toán. |
| UC14 | Xem lịch sử đơn hàng | Người dùng có thể xem các đơn hàng đã đặt trong quá khứ. |
| UC15 | Xử lý giao dịch thanh toán | Hệ thống thanh toán thực hiện xử lý giao dịch khi người dùng thanh toán đơn hàng |

Bảng 2.1 Danh sách các use case

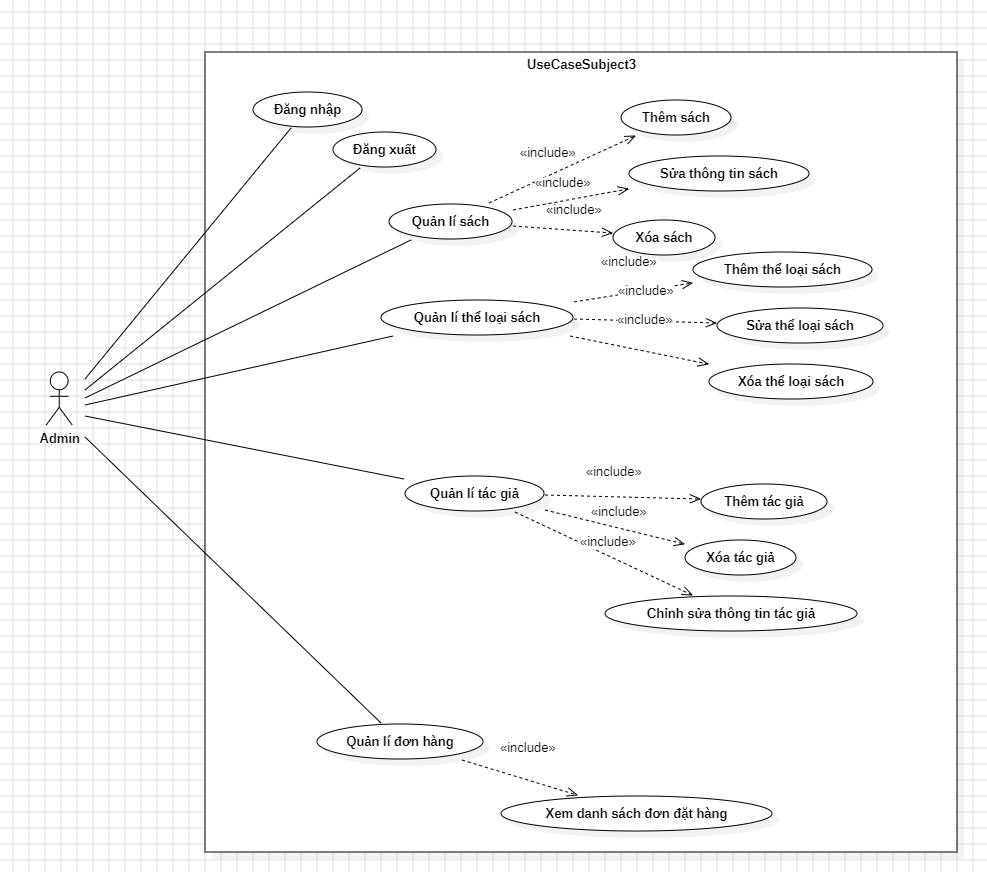
## **2.3 Mô hình use case:**

### **2.3.1 Sơ đồ use-case chính:**

A diagram of a diagram

Description automatically generated with medium confidence

### **2.3.2 Sơ đồ UseCase Admin**



### **2.3.3 Sơ đồ UseCase Khách hàng**

A diagram of a person with text

Description automatically generated

## **2.4 Đặc tả chi tiết các Usecase(UC)**

***2.4.1 Use Case Đăng nhập hệ thống***

|  |  |
| --- | --- |
| Use Case | Đăng nhập hệ thống |
| Actor | Khách hàng, Admin |
| Brief Description | Use case mô tả các bước đăng nhập của người dùng vào hệ thống |
| Pre-Condition | Không có |
| Basic Flows | 1. Hệ thống yêu cầu người dùng nhập tên đăng nhập và mật khẩu 2. Người dùng nhập tên đăng nhập và mật khẩu của mình và nhấn nút đăng nhập 3. Hệ thống kiểm tra thông tin đăng nhập 4. Hệ thống thông báo đăng nhập thành công và cho người dùng vào hệ thống, đồng thời phân quyền tùy theo vai trò |
| Alternative Flows | 1 Hệ thống kiểm tra thông tin đăng nhập sai và hiển thị thông báo ra màn hình |
| Post-conditions | Cho phép người dùng đăng nhập vào hệ thống |

Bảng 2.0 Use case đăng nhập hệ thống

***2.4.2 Use Case Đăng xuất hệ thống***

|  |  |
| --- | --- |
| Use Case | Đăng xuất hệ thống |
| Actor | Khách hàng, Admin |
| Brief Description | Cho phép người dùng đăng xuất khỏi hệ thống |
| Pre-Condition | Actor phải đăng nhập thành công vào hệ thống |
| Basic Flows | 1. Bắt đầu: Người dùng đang ở trong trạng thái đăng nhập và muốn kết thúc phiên làm việc. 2. Người dùng chọn chức năng đăng xuất: Người dùng nhấn vào nút hoặc liên kết "Đăng xuất" trên giao diện hệ thống. 3. Hệ thống xử lý yêu cầu đăng xuất: 4. Hệ thống chuyển hướng người dùng  * Hệ thống chuyển người dùng đến trang đăng nhập hoặc trang chính của ứng dụng, thông báo rằng họ đã đăng xuất thành công.  1. Kết thúc:   Người dùng đã hoàn toàn thoát khỏi hệ thống. |
| Alternative Flows | Không có |
| Post-conditions | Không có |

Bảng 2.1 Use case đăng xuất hệ thống

***2.4.3 Use Case Đăng kí tài khoản***

|  |  |
| --- | --- |
| Use Case | Đăng kí tài khoản |
| Actor | Khách hàng |
| Brief Description | Cho phép người dùng đăng kí tài khoản mới |
| Pre-Condition | Không |
| Basic Flows | 1. Hệ thống hiển thị màn hình để người dùng nhập các thông tin về khách hàng (thông tin tài khoản, thông tin cá nhân)  2. Người dùng nhập thông tin để đăng ký  3. Hệ thống tạo tài khoản theo các thông tin mà người dùng đã cung cấp. Sau đó phát sinh một mã kích hoạt và gửi link kích hoạtqua email cho người dùng.  4. Người dùng kiểm tra email và click vào link để kích hoạt tài khoản.  5. Kết thúc Use-Case. |
| Alternative Flows | 1. Chưa nhập đủ thông tin. Tất cả các thông tin về khách hàng là bắt buộc trừ thông tin về địa chỉ.  2. Tên đăng nhập đã tồn tại.  3. Email không hợp lệ.  4. Ngày tháng năm sinh không hợp lệ.  5. Mật khẩu không khớp.  6. Linh kích hoạt không hợp lệ. |
| Post-conditions | 1. Tài khoản của khách hàng được tạo thành công.  2. Mail kích hoạt được gửi thành công cho người dùng. |

Bảng 2.2 Usse case đăng kí tài khoản

***2.4.4 Use Case Quản lý sách***

A diagram with green and blue circles

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã UC** | UC-4.1 | **Tên Use Case** | Thêm sách |
| **Độ ưu tiên** | Cao | **Tác nhân** | Admin |
| **Mô tả** | Admin có khả năng thêm mới thông tin về sách vào hệ thống để cập nhật. | | |
| **Luồng chạy** | 1. Admin Truy Cập Trang Quản Lý Sách:  * Admin đăng nhập vào hệ thống và truy cập trang quản lý sách.  1. Admin Chọn Tùy Chọn "Thêm Mới Sách":  * Admin chọn tùy chọn để bắt đầu quá trình thêm sách mới.  1. Admin Nhập Thông Tin Sách:  * Admin điền thông tin chi tiết về sản phẩm như tên, mô tả, giá, số lượng, loại, hình ảnh và các thông tin khác liên quan.  1. Admin Chọn "Thêm":  * Admin chọn tùy chọn để lưu thông tin sản phẩm vào cơ sở dữ liệu.  1. Hệ Thống Xác Nhận Thêm Sản Phẩm:  * Hệ thống xác nhận rằng thông tin sản phẩm đã được thêm thành công và cập nhật danh sách sản phẩm.  1. Admin Kiểm Tra và Điều Chỉnh (Nếu Cần):  * Admin kiểm tra thông tin sản phẩm trên danh sách để đảm bảo tính chính xác và có thể điều chỉnh nếu cần thiết. | | |
| **Lưu ý** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã UC** | UC-4.2 | **Tên Use Case** | Sửa sách |
| **Độ ưu tiên** | Cao | **Tác nhân** | Admin |
| **Mô tả** | Admin có khả năng sửa đổi thông tin về sách trong hệ thống để cập nhật và điều chỉnh thông tin liên quan. | | |
| **Luồng chạy** | 1. Admin Truy Cập Trang Quản Lý Sách:  * Admin đăng nhập vào hệ thống và truy cập trang quản lý sách.  1. Admin Chọn Tùy Chọn "Sửa Sách":  * Admin chọn tùy chọn vào icon sửa sách để bắt đầu quá trình sửa sách mới.  1. Admin Điều Chỉnh Thông Tin Sách:  * Admin điền thông tin chi tiết về sản phẩm như tên, mô tả, giá, số lượng, loại, hình ảnh và các thông tin khác liên quan.  1. Admin Chọn "Đồng ý":  * Admin chọn tùy chọn để lưu thông tin sản phẩm vừa sửa vào cơ sở dữ liệu.  1. Hệ Thống Xác Nhận Sửa Sản Phẩm:  * Hệ thống xác nhận rằng thông tin sản phẩm đã được thêm thành công và cập nhật danh sách sản phẩm.  1. Admin Kiểm Tra và Điều Chỉnh (Nếu Cần):  * Admin kiểm tra thông tin sản phẩm trên danh sách để đảm bảo tính chính xác và có thể điều chỉnh nếu cần thiết. | | |
| **Lưu ý** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã UC** | UC-4.3 | **Tên Use Case** | Xóa sách |
| **Độ ưu tiên** | Cao | **Tác nhân** | Admin |
| **Mô tả** | Admin có khả năng xóa thông tin về sách khỏi hệ thống khi sản phẩm không còn khả dụng hoặc không còn được bán. | | |
| **Luồng chạy** | 1. Admin Truy Cập Trang Quản Lý Sách:  * Admin đăng nhập vào hệ thống và truy cập trang quản lý sách.  1. Admin Chọn Sách Cần Xóa:  * Admin chọn sách cụ thể từ danh sách sách để bắt đầu quá trình xóa.  1. Admin Xác Nhận Yêu Cầu Xóa:  * Admin xác nhận rằng họ muốn xóa sách khỏi hệ thống và xác nhận yêu cầu xóa.  1. Hệ Thống Xác Nhận Xóa Sách:  * Hệ thống xác nhận rằng sách đã được xóa khỏi cơ sở dữ liệu và cập nhật danh sách sách.  1. Kiểm Tra và Đảm Bảo Tính Chính Xác (Nếu Cần):   Admin kiểm tra lại danh sách sản phẩm để đảm bảo rằng sản phẩm đã được xóa và không còn xuất hiện trên trang web. | | |
| **Lưu ý** |  | | |

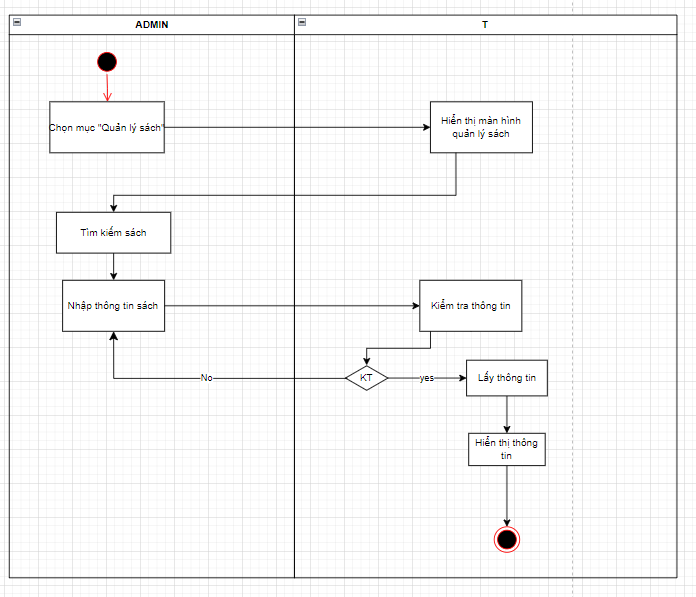
**Sơ đồ hoạt động thêm, sửa, xóa sách**

**A diagram of a flowchart

Description automatically generated**

|  |  |  |  |
| --- | --- | --- | --- |
| Mã UC | UC-4.4 | Tên Use Case | Tìm kiếm sách |
| Độ ưu tiên | Cao | Tác nhân | Admin |
| Mô tả | Admin có khả năng tìm kiếm thông tin về sản phẩm để dễ dàng quản lý và kiểm tra trạng thái của sách trong hệ thống. | | |
| Luồng chạy | 1. Admin Truy Cập Trang Quản Lý Sách:  * Admin đăng nhập vào hệ thống và truy cập trang quản lý sách.  1. Admin Nhập Thông Tin Tìm Kiếm:  * Admin nhập thông tin tìm kiếm như tên sách, thương hiệu, hoặc mô tả,.. vào ô tìm kiếm.  1. Admin Chọn "Tìm Kiếm":  * Admin nhấn nút "Tìm Kiếm" để bắt đầu quá trình tìm kiếm sách/  1. Hệ Thống Hiển Thị Kết Quả Tìm Kiếm:  * Hệ thống hiển thị danh sách các sách phù hợp với thông tin tìm kiếm của admin.  1. Admin Xem Chi Tiết Sách (Nếu Cần):  * Admin có thể chọn xem chi tiết thông tin về sácg từ danh sách kết quả tìm kiếm. | | |
| Lưu ý | Admin cần có quyền truy cập và quyền hạn để thực hiện chức năng này.  Hệ thống cần cung cấp thông báo rõ ràng nếu không tìm thấy kết quả phù hợp với thông tin tìm kiếm. | | |

**Sơ đồ hoạt động tìm kiếm sách**

****

***2.4.5 Use Case Thêm, Sửa, Thanh toán giỏ hàng***

A diagram of a company

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã UC** | UC-10.1,10.2 | **Tên Use Case** | Thêm , sửa sản phẩm giỏ hàng |
| **Độ ưu tiên** | Trung bình | **Tác nhân** | Khách hàng |
| **Mô tả** | Người dùng có khả năng thêm và sửa sản phẩm trong giỏ hàng của mình. | | |
| **Luồng chạy** | 1. Người Dùng Truy Cập Giỏ Hàng:  * Người dùng truy cập trang giỏ hàng trong hệ thống sau khi đã đăng nhập vào tài khoản của mình.  1. Người Dùng Thêm Sản Phẩm Vào Giỏ Hàng Qua Web  * Người dùng chọn tùy chọn "Thêm vào Giỏ Hàng" từ trang sản phẩm và chọn số lượng sản phẩm cần thêm.  1. Hệ Thống Cập Nhật Giỏ Hàng:  * Hệ thống cập nhật giỏ hàng của người dùng với thông tin về sản phẩm mới và số lượng.  1. Người Dùng Sửa Số Lượng Sản Phẩm Trong Giỏ Hàng:  * Nếu cần, người dùng có thể sửa số lượng sản phẩm trong giỏ hàng bằng cách cập nhật thông tin trên trang giỏ hàng.  1. Hệ Thống Cập Nhật Thông Tin Sản Phẩm:  * Hệ thống cập nhật thông tin giỏ hàng với số lượng sản phẩm được chỉnh sửa (nếu có). | | |
| **Lưu ý** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã UC** | UC-16.2 | **Tên Use Case** | Xóa sản phẩm giỏ hàng |
| **Độ ưu tiên** | Trung bình | **Tác nhân** | Người dùng |
| **Mô tả** | Người dùng có khả năng xóa sản phẩm khỏi giỏ hàng của mình. | | |
| **Luồng chạy** | 1. Người Dùng Truy Cập Giỏ Hàng:  * Người dùng truy cập trang giỏ hàng trong hệ thống sau khi đã đăng nhập vào tài khoản của mình.  1. Người Dùng Chọn Sản Phẩm Cần Xóa:  * Người dùng chọn sản phẩm cần xóa khỏi giỏ hàng từ danh sách các sản phẩm hiển thị trên trang giỏ hàng.  1. Người Dùng Chọn Tùy Chọn "Xóa":  * Sau khi chọn sản phẩm, người dùng chọn tùy chọn "Xóa" để loại bỏ sản phẩm khỏi giỏ hàng.  1. Hệ Thống Cập Nhật Giỏ Hàng:  * Hệ thống cập nhật giỏ hàng của người dùng bằng cách loại bỏ sản phẩm đã chọn. | | |
| **Lưu ý** | Giao diện người dùng cần cung cấp tùy chọn rõ ràng để xóa sản phẩm và cần xác nhận trước khi thực hiện thao tác này.  Cần kiểm tra và xác nhận tính khả dụng của sản phẩm trước khi xóa khỏi giỏ hàng.  Thông báo cần được hiển thị để thông báo cho người dùng về việc xóa sản phẩm thành công. | | |

**Sơ đồ hoạt động xóa, cập nhật sản phẩm trong giỏ hàng**

A diagram of a flowchart

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã UC** | UC-12 | **Tên Use Case** | Thanh Toán |
| **Độ ưu tiên** | Cao | **Tác nhân** | Người dùng |
| **Mô tả** | Người dùng có khả năng thanh toán khi thực hiện giao dịch, bao gồm cả việc thanh toán online và tiền mặt. | | |
| **Luồng chạy** | 1. Người Dùng Truy Cập Trang Thanh Toán:  * Người dùng truy cập vào giỏ hàng và chọn trang thanh toán.  1. Người Dùng Nhập Địa Chỉ Nhận Hàng:  * Nhập các thông tin địa chỉ, SĐT vào ô điền thông tin.  1. Người Dùng Thực Hiện Thanh Toán Online Hoặc Chuẩn Bị Tiền Mặt:  * Nếu thanh toán online, người dùng thực hiện quy trình thanh toán PAYPAL. * Nếu thanh toán tiền mặt, người dùng chuẩn bị tiền mặt cho quá trình thanh toán khi nhận hàng.  1. Hệ thống tạo hóa đơn và gửi về email của khách hàng.  * Khi thanh toán thành công thì hệ thống sẽ tạo hóa đơn cho khách hàng. * Hệ Thống sẽ auto gửi mail xác nhận cho khách hàng. | | |
| **Lưu ý** |  | | |

**Sơ đồ hoạt động thanh toán**

A diagram of a work flow

Description automatically generated

***2.4.6 UseCase quản lí thể loại sách***

UC 5.1 Use case thêm thể loại sách

|  |  |
| --- | --- |
| Use Case | Thêm thể loại sách |
| Actor | Admin |
| Brief Description | Admin có thể thêm mới một thể loại sách vào hệ thống, giúp tổ chức và phân loại các đầu sách dễ dàng hơn. |
| Pre-Condition | Actor phải đăng nhập thành công vào hệ thống và có quyền truy cập là admin |
| Basic Flows | 1. Admin truy cập vào phần quản lý thể loại sách trên giao diện hệ thống. 2. Admin chọn tùy chọn "Thêm thể loại mới." 3. Hệ thống hiển thị biểu mẫu yêu cầu nhập thông tin về thể loại mới, bao gồm: 4. Tên thể loại (bắt buộc). 5. Admin nhập thông tin về thể loại sách mới và xác nhận. 6. Hệ thống kiểm tra thông tin hợp lệ (ví dụ: tên thể loại chưa tồn tại). 7. Nếu thông tin hợp lệ, hệ thống lưu thể loại sách mới vào cơ sở dữ liệu. 8. Hệ thống thông báo thành công và cập nhật danh sách các thể loại. |
| Alternative Flows | 6.1 Tên thể loại đã tồn tại: Hệ thống sẽ thông báo lỗi và yêu cầu admin nhập tên khác.  6.2Thông tin không hợp lệ: Nếu admin không điền đầy đủ hoặc sai thông tin (ví dụ bỏ trống tên thể loại), hệ thống sẽ yêu cầu chỉnh sửa |
| Post-conditions | Thể loại sách mới được thêm vào cơ sở dữ liệu và hiển thị trong danh sách các thể loại sách hiện có. |

UC 5.2 Use case Sửa thông tin thể loại sách

|  |  |
| --- | --- |
| Use Case | Sửa thông tin thể loại sách |
| Actor | Admin |
| Brief Description | Admin có thể chỉnh sửa thông tin của một thể loại sách trong hệ thống (như tên thể loại hoặc mô tả) để cập nhật hoặc thay đổi khi cần. |
| Pre-Condition | Actor phải đăng nhập thành công vào hệ thống và có quyền truy cập là admin |
| Basic Flows | 1. Admin truy cập vào phần quản lý thể loại sách trong giao diện hệ thống. 2. Admin chọn thể loại sách cần chỉnh sửa từ danh sách hiện có. 3. Admin nhấp vào tùy chọn "Chỉnh sửa" hoặc biểu tượng tương ứng để chỉnh sửa thể loại. 4. Hệ thống hiển thị biểu mẫu với thông tin hiện tại của thể loại, bao gồm:  * Tên thể loại.  1. Admin cập nhật thông tin mới (tên thể loại, mô tả...). 2. Hệ thống kiểm tra tính hợp lệ của thông tin mới:  * Tên thể loại phải không trùng lặp với các thể loại hiện có. * Thông tin cần điền đầy đủ và hợp lệ.  1. Nếu thông tin hợp lệ, hệ thống lưu thay đổi vào cơ sở dữ liệu. 2. Hệ thống thông báo thành công và cập nhật danh sách các thể loại với thông tin đã sửa. |
| Alternative Flows | 7.1 Tên thể loại trùng lặp: Nếu tên thể loại mới trùng với một thể loại hiện có, hệ thống sẽ thông báo lỗi và yêu cầu admin chọn tên khác.  7.2 Thông tin không hợp lệ: Nếu admin không điền đúng hoặc đầy đủ thông tin (ví dụ bỏ trống tên), hệ thống sẽ yêu cầu chỉnh sửa. |
| Post-conditions | Thông tin của thể loại sách được cập nhật trong cơ sở dữ liệu và hiển thị trong danh sách các thể loại. |

UC 5.3 Use case xóa thể loại sách

|  |  |
| --- | --- |
| Use Case | Thêm thể loại sách |
| Actor | Admin |
| Brief Description | Admin có thể xóa một thể loại sách khỏi hệ thống khi thể loại đó không còn cần thiết hoặc không sử dụng nữa. |
| Pre-Condition | Actor phải đăng nhập thành công vào hệ thống và có quyền truy cập là admin |
| Basic Flows | 1. Admin truy cập vào phần quản lý thể loại sách trong giao diện hệ thống. 2. Admin chọn thể loại sách cần xóa từ danh sách hiện có. 3. Admin nhấp vào nút "Xóa" hoặc tùy chọn xóa tương ứng với thể loại sách đó. 4. Hệ thống hiển thị thông báo xác nhận yêu cầu xóa, đảm bảo admin biết rằng xóa thể loại có thể ảnh hưởng đến các sách liên quan. 5. Admin xác nhận việc xóa thể loại. 6. Hệ thống kiểm tra các ràng buộc và thực hiện việc xóa thể loại: 7. Hệ thống thông báo thành công và cập nhật danh sách các thể loại sách. |
| Alternative Flows | 5.1 Nếu Admin không muốn xóa thể loại sách, họ có thể nhấn nút "Hủy" để thoát giao diện chỉnh sửa mà không thay đổi dữ liệu. |
| Post-conditions | Thể loại sách được xóa khỏi cơ sở dữ liệu và không còn hiển thị trong danh sách các thể loại. |

***2.4.7 Use case Quản lý đơn hàng***

|  |  |
| --- | --- |
| Use Case | Quản lý đơn hàng |
| Actor | Admin |
| Brief Description | Admin có thể theo dõi, xem, chỉnh sửa, và xóa các đơn hàng trong hệ thống, đảm bảo quá trình xử lý đơn hàng diễn ra suôn sẻ và hiệu quả. |
| Pre-Condition | Actor phải đăng nhập thành công vào hệ thống và có quyền truy cập là admin |
| Basic Flows | 1. Admin truy cập vào phần quản lý đơn hàng trong giao diện hệ thống. 2. Hệ thống hiển thị danh sách các đơn hàng hiện có, bao gồm thông tin như:  * Mã đơn hàng * Tên khách hàng * Trạng thái đơn hàng (đang xử lý, đã giao, hủy, v.v.) * Ngày đặt hàng  1. Admin có thể thực hiện các thao tác sau:  * Xem chi tiết đơn hàng: Admin chọn một đơn hàng cụ thể để xem thông tin chi tiết, bao gồm các mặt hàng, số lượng, địa chỉ giao hàng, và thông tin thanh toán. * Chỉnh sửa đơn hàng: Admin có thể chỉnh sửa thông tin đơn hàng, như thay đổi trạng thái, cập nhật thông tin khách hàng, hoặc thêm bớt mặt hàng. * Hủy đơn hàng: Admin có thể hủy một đơn hàng nếu khách hàng yêu cầu hoặc nếu có lý do hợp lệ. * Tìm kiếm và lọc đơn hàng: Admin có thể tìm kiếm đơn hàng theo mã đơn hàng, tên khách hàng, hoặc trạng thái để dễ dàng quản lý.  1. Sau khi thực hiện thao tác, hệ thống cập nhật danh sách đơn hàng và thông báo thành công. |
| Alternative Flows | · Đơn hàng không tồn tại: Nếu admin cố gắng xem hoặc chỉnh sửa một đơn hàng không tồn tại, hệ thống sẽ thông báo lỗi.  · Lỗi hệ thống: Nếu có vấn đề trong quá trình xử lý (ví dụ: mất kết nối cơ sở dữ liệu), hệ thống sẽ thông báo lỗi và không thực hiện thao tác. |
| Post-conditions | · Thông tin đơn hàng được cập nhật trong cơ sở dữ liệu (nếu có chỉnh sửa).  · Danh sách đơn hàng hiện có được cập nhật để phản ánh các thay đổi đã thực hiện. |

***2.4.8 Use Case Xem chi tiết sách***

|  |  |
| --- | --- |
| Use Case | Xem chi tiết sách |
| Actor | Người dùng, Admin |
| Brief Description | Người dùng và admin có thể xem chi tiết thông tin về một cuốn sách trong hệ thống, bao gồm thông tin mô tả, tác giả, thể loại, đánh giá, và các thông tin khác liên quan. |
| Pre-Condition | · Người dùng hoặc admin đã đăng nhập vào hệ thống.  · Hệ thống có thông tin về các cuốn sách và danh sách sách hiển thị trên giao diện. |
| Basic Flows | 1. Người dùng hoặc admin truy cập vào hệ thống và tìm kiếm cuốn sách cần xem chi tiết. 2. Hệ thống hiển thị danh sách các sách theo kết quả tìm kiếm hoặc từ danh mục sách. 3. Người dùng hoặc admin chọn một cuốn sách từ danh sách để xem chi tiết. 4. Hệ thống hiển thị trang chi tiết của cuốn sách, bao gồm các thông tin như:  * Tên sách. * Tác giả. * Thể loại. * Mô tả ngắn về sách. * Năm xuất bản. * Nhà xuất bản. * Giá bán  1. Người dùng có thể thực hiện các thao tác khác như:  * Thêm sách vào giỏ hàng (nếu là hệ thống bán hàng). * Thêm sách vào danh sách yêu thích.  1. Admin có thể kiểm tra và chỉnh sửa thông tin nếu cần (nếu admin có quyền chỉnh sửa). |
| Alternative Flows | · Sách không tồn tại: Nếu cuốn sách đã bị xóa hoặc không còn trong hệ thống, hệ thống sẽ hiển thị thông báo lỗi "Sách không tồn tại" và quay lại danh sách sách.  · Lỗi hệ thống: Nếu xảy ra sự cố khi hiển thị chi tiết sách (ví dụ mất kết nối cơ sở dữ liệu), hệ thống sẽ thông báo lỗi và không thể hiển thị chi tiết sách. |
| Post-conditions | Người dùng hoặc admin có thể xem thông tin chi tiết của cuốn sách đã chọn. |

***2.4.9 Use Case Xem chi tiết đơn hàng***

|  |  |
| --- | --- |
| Use Case | Xem chi tiết đơn hàng |
| Actor | Người dùng |
| Brief Description | Khách hàng có thể xem chi tiết thông tin về một đơn hàng đã đặt, bao gồm sản phẩm đã mua, tổng giá trị đơn hàng, trạng thái đơn hàng, phương thức thanh toán, và thông tin giao hàng. |
| Pre-Condition | * Khách hàng đã đăng nhập vào tài khoản của mình trên hệ thống. * Khách hàng đã đặt một hoặc nhiều đơn hàng trước đó và các đơn hàng này được lưu trong cơ sở dữ liệu. |
| Basic Flows | 1. Khách hàng đăng nhập vào hệ thống và truy cập vào phần **Tài khoản của tôi** hoặc **Lịch sử đơn hàng**. 2. Hệ thống hiển thị danh sách các đơn hàng đã đặt của khách hàng, bao gồm thông tin cơ bản như mã đơn hàng, ngày đặt hàng, và trạng thái hiện tại (đang xử lý, đã giao, đã hủy...). 3. Khách hàng chọn một đơn hàng cụ thể để xem chi tiết. 4. Hệ thống hiển thị trang chi tiết đơn hàng với các thông tin sau:  * **Ngày đặt hàng**. * **Sản phẩm đã mua** (bao gồm tên sản phẩm, số lượng, và giá mỗi sản phẩm). * **Tổng giá trị đơn hàng** (bao gồm chi phí sản phẩm, thuế, và phí vận chuyển nếu có). * **Phương thức thanh toán** (thẻ tín dụng, chuyển khoản, thanh toán khi nhận hàng...). * **Thông tin giao hàng** (tên người nhận, địa chỉ giao hàng, số điện thoại liên lạc). * **Trạng thái đơn hàng** (đang xử lý, đã giao, đang vận chuyển, đã hủy...).  1. · Khách hàng có thể theo dõi trạng thái đơn hàng và thông tin giao hàng nếu đơn hàng đang trong quá trình xử lý hoặc vận chuyển. 2. · Nếu có bất kỳ vấn đề gì, khách hàng có thể chọn liên hệ với bộ phận hỗ trợ qua tùy chọn được cung cấp trên giao diện. |
| Alternative Flows | · Đơn hàng không tồn tại: Nếu khách hàng cố gắng xem chi tiết đơn hàng đã bị xóa hoặc không còn tồn tại trong hệ thống, hệ thống sẽ thông báo lỗi "Đơn hàng không tồn tại".  · Lỗi hệ thống: Nếu có vấn đề trong việc truy xuất thông tin đơn hàng (ví dụ lỗi kết nối cơ sở dữ liệu), hệ thống sẽ thông báo và không thể hiển thị chi tiết đơn hàng. |
| Post-conditions | · Khách hàng có thể xem thông tin chi tiết đầy đủ của đơn hàng đã chọn.  · Trạng thái đơn hàng và thông tin giao hàng hiển thị chính xác theo dữ liệu cập nhật từ hệ thống. |

***2.4.10 Use case Xem lịch sử đơn hàng***

|  |  |
| --- | --- |
| Use Case | Xem lịch sử đơn hàng |
| Actor | Người dùng |
| Brief Description | Khách hàng có thể xem lịch sử các đơn hàng đã đặt trước đây, bao gồm thông tin về từng đơn hàng như mã đơn hàng, ngày đặt hàng, trạng thái, và tổng giá trị. |
| Pre-Condition | · Khách hàng đã đăng nhập vào tài khoản của mình trên hệ thống.  · Khách hàng đã đặt một hoặc nhiều đơn hàng trước đó và các đơn hàng này được lưu trong cơ sở dữ liệu. |
| Basic Flows | 1. Khách hàng đăng nhập vào hệ thống và truy cập vào phần **Tài khoản của tôi** hoặc **Lịch sử đơn hàng**. 2. Hệ thống hiển thị danh sách các đơn hàng đã đặt của khách hàng, bao gồm thông tin cơ bản như mã đơn hàng, ngày đặt hàng, và trạng thái hiện tại (đang xử lý, đã giao, đã hủy...). 3. Khách hàng có thể thực hiện các thao tác sau:  * Xem chi tiết từng đơn hàng: Khi khách hàng chọn một đơn hàng cụ thể, hệ thống sẽ hiển thị thông tin chi tiết của đơn hàng đó (bao gồm các sản phẩm, địa chỉ giao hàng, phương thức thanh toán...). * Lọc và tìm kiếm đơn hàng: Khách hàng có thể lọc đơn hàng theo thời gian, trạng thái (đã giao, đã hủy, đang xử lý), hoặc tìm kiếm theo mã đơn hàng.  1. Hệ thống sẽ hiển thị kết quả theo yêu cầu tìm kiếm hoặc lọc của khách hàng. |
| Alternative Flows | · **Không có đơn hàng nào:** Nếu khách hàng chưa từng đặt hàng, hệ thống sẽ thông báo "Chưa có đơn hàng nào" và không hiển thị danh sách lịch sử.  · **Lỗi hệ thống:** Nếu xảy ra sự cố trong quá trình truy xuất thông tin (ví dụ lỗi cơ sở dữ liệu), hệ thống sẽ hiển thị thông báo lỗi và không thể hiển thị lịch sử đơn hàng. |
| Post-conditions | Khách hàng có thể xem toàn bộ lịch sử đơn hàng đã đặt và có thể truy cập chi tiết thông tin của từng đơn hàng. |

# **CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ CƠ SỞ DỮ LIỆU, GIAO DIỆN**

* 1. **Biểu đồ diagram:**

A computer screen shot of a computer

Description automatically generated

* + 1. **Biểu đồ bảng books, authors, categories và companies:**

**A screenshot of a computer

Description automatically generated**

* + 1. **Biểu đồ bảng orders, order\_details, payments:**

**A screenshot of a computer

Description automatically generated**

* + 1. **Biểu đồ bảng customers:**

**A screenshot of a computer

Description automatically generated**

* 1. **Bảng book:**

|  |  |  |
| --- | --- | --- |
| Tên trường thông tin | Cụ thể | Ghi chú |
| id | Mã sách | Khóa chính |
| book\_name | Tên sách |  |
| description | Mô tả về sách |  |
| publish\_date | Ngày phát hành |  |
| suggest | Mục danh sách được khuyến nghị |  |
| author\_id | Mã tác giả | Khóa ngoại |
| company\_id | Mã nhà phát hành | Khóa ngoại |
| category\_id | Mã thể loại | Khóa ngoại |
| publishing\_house | Tên nhà phát hành |  |
| translator | Người dịch |  |
| number\_of\_pages | Số trang của cuốn sách |  |
| quality | Số lượng sách |  |
| price | Giá sách |  |
| cover\_price | Gía giảm |  |
| book\_image | Hình minh họa chính của cuốn sách |  |
| images | Hình ảnh minh họa thêm |  |
| created\_at | Ngày tạo |  |
| updated\_at | Ngày cập nhật |  |
| favorite | Yêu thích |  |

* 1. **Bảng authors:**

|  |  |  |
| --- | --- | --- |
| Tên trường thông tin | Cụ thể | Ghi chú |
| id | Mã tác giả | Khóa chính |
| author\_name | Tên tác giả |  |
| author\_info | Thông tin tác giả |  |
| author\_image | Hình ảnh tác giả |  |

* 1. **Bảng categories:**

|  |  |  |
| --- | --- | --- |
| Tên trường thông tin | Cụ thể | Ghi chú |
| id | Mã thể loại | Khóa chính |
| category\_name | Tên thể loại |  |
| parent\_id | Thời gian ra khỏi quán |  |
| order | Mã bàn |  |

* 1. **Bảng comments:**

|  |  |  |
| --- | --- | --- |
| Tên trường thông tin | Cụ thể | Ghi chú |
| id | Mã bình luận | Khóa chính |
| content | Nội dung bình luận |  |
| createdAt | Ngày bình luận |  |
| rate | Đánh giá |  |
| updatedAt | Ngày cập nhật |  |
| bookId | Id sách được bình luận |  |
| userId | Id khách bình luận |  |

* 1. **Bảng companies:**

|  |  |  |
| --- | --- | --- |
| Tên trường thông tin | Cụ thể | Ghi chú |
| id | Mã nhà phát hành | Khóa chính |
| company\_name | Tên nhà phát hành |  |
| company\_info | Thông tin nhà phát hành |  |
| company\_image | Hình ảnh minh họa nhà phát hành |  |
| companyImage |  |  |

* 1. **Bảng customers:**

|  |  |  |
| --- | --- | --- |
| Tên trường thông tin | Cụ thể | Ghi chú |
| customerId | Mã khách hàng | Khóa chính |
| email | Gmail đăng kí của khách hàng |  |
| enabled | Tình trạng của tài khoản khách hàng |  |
| fullname | Họ và tên khách hàng |  |
| isUsing2FA | Có sử dụng bảo mật 2 lớp |  |
| password | Mật khẩu của tài khoản | Mã hóa |

* 1. **Bảng hibernate\_sequence:**

|  |  |  |
| --- | --- | --- |
| Tên trường thông tin | Cụ thể | Ghi chú |
| next\_val | Chứa giá trị tiếp theo sẽ được sử dụng cho khóa chính | Là giá trị hiện tại của trình tự (sequence). Mỗi lần Hibernate cần tạo ra một giá trị mới cho khóa chính, nó sẽ lấy giá trị từ next\_val, sau đó tăng giá trị này lên để sử dụng cho lần tiếp theo. |

* 1. **Bảng migrations:**

|  |  |  |
| --- | --- | --- |
| Tên trường thông tin | Cụ thể | Ghi chú |
| id | Mã phiên bản | Khóa chính |
| batch | Cột này đại diện cho một nhóm các lần di trú được thực hiện cùng nhau. | Ví dụ, nếu bạn chạy nhiều di trú cùng một lúc, tất cả các di trú đó sẽ được gán cùng một số batch. |
| migration | Cột này lưu trữ tên của tệp di trú (migration file) hoặc định danh của di trú đã được thực hiện. | Tên này thường bao gồm một dấu thời gian hoặc số thứ tự cùng với mô tả về thay đổi được thực hiện, ví dụ: 20231001123000\_create\_users\_table. |

* 1. **Bảng order\_details:**

|  |  |  |
| --- | --- | --- |
| Tên trường thông tin | Cụ thể | Ghi chú |
| id | Mã thông tin hóa đơn chi tiết | Khóa chính |
| price | Tổng giá của orders |  |
| quality |  |  |
| book\_id | Chứa các mã sách | Khóa ngoại |
| order\_id | Mã hóa đơn | Khóa ngoại |

* 1. **Bảng orders:**

|  |  |  |
| --- | --- | --- |
| Tên trường thông tin | Cụ thể | Ghi chú |
| id | Mã hóa đơn | Khóa chính |
| created\_at | Ngày mua |  |
| nameReceiver | Tên người nhận |  |
| order\_status | Tình trạng hóa đơn |  |
| phoneReceiver | Sdt người nhận |  |
| shipping\_address | Địa chỉ ng nhận |  |
| shipping\_fee | Phí giao hàng |  |

* 1. **Bảng password\_resets:**

|  |  |  |
| --- | --- | --- |
| Tên trường thông tin | Cụ thể | Ghi chú |
| email | Thông tin email | Khóa chính |
| token | Token user gửi request |  |
| created\_at | Ngày khởi tạo request |  |

* 1. **Bảng payments:**

|  |  |  |
| --- | --- | --- |
| Tên trường thông tin | Cụ thể | Ghi chú |
| id | Mã thanh toán | Khóa chính |
| createdAT | Ngày thanh toán |  |
| paymentStatus | Tình trạng thanh toán |  |
| paymentType | Phương thức thanh toán |  |
| rememberToken | Ghi lại Token |  |

* 1. **Bảng roles:**

|  |  |  |
| --- | --- | --- |
| Tên trường thông tin | Cụ thể | Ghi chú |
| id | Mã loại tài khoản | Khóa chính |
| roleName | Loại tài khoản |  |
| customerId | Mã tài khoản khách hàng | Khóa ngoại |

* 1. **Bảng saves:**

|  |  |  |
| --- | --- | --- |
| Tên trường thông tin | Cụ thể | Ghi chú |
| id | Mã favorite | Khóa chính |
| created\_at | Ngày tạo |  |
| updated\_at | Ngày cập nhật |  |
| book\_id | ID sách được yêu thích |  |
| customerId | ID tài khoản khách hàng yêu thích sách đó |  |

* 1. **Bảng verificationtoken:**

|  |  |  |
| --- | --- | --- |
| Tên trường thông tin | Cụ thể | Ghi chú |
| id | Mã khóa **verificationtoken** | Khóa chính |
| expiryDate | Ngày hết hạn |  |
| token | Thời gian ra khỏi quán | Khóa ngoại |
| customerId | ID khách hàng | Khóa ngoại |

* 1. **7 Giao diện**

**3.17.1 Sơ đồ giao diện người dùng:**

**A diagram with blue rectangles

Description automatically generated**

**3.17.2 Sơ đồ giao diện admin:**

**A diagram of a company

Description automatically generated**

# **CHƯƠNG 4: CÀI ĐẶT VÀ THỰC NGHIỆM**

* 1. **Cấu trúc thư mục:**

**A screenshot of a computer program

Description automatically generated**

**src/main/java/com/java**: Đây là thư mục chính chứa mã nguồn Java của ứng dụng.

**config/**: Chứa các lớp cấu hình ứng dụng. Các lớp này thường sử dụng các chú thích như @Configuration và chứa các bean cấu hình cho Spring.

**controller/**: Chứa các lớp điều khiển (Controller) xử lý các yêu cầu từ người dùng. Mỗi lớp điều khiển tương ứng với một bộ chức năng nhất định (ví dụ: quản lý sách, giỏ hàng, thanh toán).

**entity/**: Chứa các lớp thực thể (Entity) mà ứng dụng tương tác với cơ sở dữ liệu. Mỗi lớp tương ứng với một bảng trong cơ sở dữ liệu.

**repository/**: Chứa các lớp truy cập dữ liệu (Repository). Các lớp này thường mở rộng JpaRepository hoặc CrudRepository để tương tác với cơ sở dữ liệu.

**service/**: Chứa các lớp dịch vụ (Service) chứa logic nghiệp vụ của ứng dụng. Các lớp này thường sử dụng các lớp repository để thực hiện các thao tác dữ liệu.

**src/main/resources/**: Thư mục này chứa các tài nguyên cần thiết cho ứng dụng.

**static/**: Chứa các tài nguyên tĩnh như CSS, JavaScript, hình ảnh, v.v. Những tài nguyên này được phục vụ trực tiếp cho trình duyệt.

**templates/**: Chứa các tệp mẫu Thymeleaf (.html) được sử dụng để tạo giao diện người dùng.

**application.properties**: Tệp cấu hình chính cho ứng dụng Spring Boot. Tệp này chứa các thuộc tính cấu hình như kết nối cơ sở dữ liệu, cài đặt Hibernate, v.v.

* 1. **Cài đặt thanh toán:**
     1. **Thanh toán bằng VN Pay:**

VNPay là một cổng thanh toán trực tuyến phổ biến tại Việt Nam, cho phép người dùng thực hiện giao dịch thanh toán qua thẻ ngân hàng, ví điện tử, và các phương thức thanh toán khác. VNPay cung cấp API để các ứng dụng web và di động có thể tích hợp vào quy trình thanh toán một cách dễ dàng và an toàn.

**Các bước tích hợp API VNPay**

**Đăng ký tài khoản VNPay**: Truy cập trang web VNPay và đăng ký tài khoản thử nghiệm (merchant account). Sau khi đăng ký, sẽ nhận được mã đơn vị (Merchant Code) và khóa bí mật (Hash Secret).

**Cấu hình môi trường phát triển**: Sử dụng URL môi trường phát triển (sandbox).

**Tạo yêu cầu thanh toán**: Tạo một yêu cầu thanh toán bằng cách gửi các thông tin cần thiết đến VNPay API, bao gồm mã đơn vị, số tiền, mô tả giao dịch, và các thông tin khác.

**Xử lý phản hồi từ VNPay**: Sau khi người dùng hoàn tất thanh toán, VNPay sẽ gửi phản hồi về trạng thái giao dịch. Xử lý phản hồi này để cập nhật trạng thái giao dịch trong hệ thống.

Đoạn Code Xử Lý Thanh toán bằng VN Pay:

if (StringUtils.equals(checkOut, "vnpay")) {

            // VNPAY payment process

            String baseUrl = request.getScheme() + "://" + request.getServerName() + ":" + request.getServerPort();

            try {

                String vnpayUrl = vnPayService.createOrder(request, (int) totalPrice, "Thanh Toan Bang VN PAY", baseUrl);

                return "redirect:" + vnpayUrl;

            } catch (Exception e) {

                log.error("Error while processing VNPAY payment: " + e.getMessage());

                // Handle VNPAY error

            }

        }

Thực hiện sau khi thanh toán thành công:

@GetMapping("/vnpay-payment-return")

    public String vnpayPaymentReturn(HttpServletRequest request, Model model, Customer customer)

            throws MessagingException {

        // Process VNPAY return request

        int paymentStatus = vnPayService.orderReturn(request);

        Collection<CartItem> cartItems = shoppingCartService.getCartItems();

        model.addAttribute("cartItems", cartItems);

        model.addAttribute("total", shoppingCartService.getAmount());

        double totalPrice = 0;

        double fee = orderFinal.getShippingFee();;

        for (CartItem cartItem : cartItems) {

            double price = cartItem.getQuantity() \* cartItem.getBook().getPrice();

            totalPrice += price;

        }

        totalPrice += fee;

        model.addAttribute("totalPrice", totalPrice);

        model.addAttribute("totalCartItems", shoppingCartService.getCount());

        if (paymentStatus == 1) {

            // Handle successful payment, update order, and send confirmation email

            session = request.getSession();

                Date date = new Date();

                orderFinal.setCreatedAt(date);

                orderFinal.setOrderStatus("ĐÃ THANH TOÁN BẰNG VNPAY");

                orderFinal.getId();

                orderFinal.setCustomer(customer);

                orderRepository.save(orderFinal);

                for (CartItem cartItem : cartItems) {

                    OrderDetail orderDetail = new OrderDetail();

                    orderDetail.setQuality(cartItem.getQuantity());

                    orderDetail.setOrder(orderFinal);

                    orderDetail.setBook(cartItem.getBook());

                    double unitPrice = cartItem.getBook().getPrice();

                    orderDetail.setPrice(unitPrice);

                    orderDetailRepository.save(orderDetail);

                }

                serviceCommon.sendSimpleEmail(customer.getEmail(), "Book Store Xác Nhận Đơn hàng", "aaaa", cartItems,

                        fee, totalPrice, orderFinal);

                shoppingCartService.clear();

                session.removeAttribute("cartItems");

                model.addAttribute("orderId", orderFinal.getId());

                orderFinal = new Order();

            return "redirect:/success";  // Redirect to success page

        } else {

            log.error("VNPAY payment failed.");

            return "redirect:/";  // Redirect to checkout page in case of failure

        }

    }

* + 1. **Thanh toán bằng PayPal:**

PayPal là một trong những cổng thanh toán trực tuyến phổ biến nhất trên thế giới, cho phép người dùng thực hiện giao dịch thanh toán một cách an toàn và nhanh chóng. PayPal cung cấp API cho các nhà phát triển để tích hợp thanh toán vào ứng dụng hoặc trang web của họ, cho phép người dùng thanh toán bằng thẻ tín dụng, thẻ ghi nợ, và tài khoản PayPal.

**Các bước tích hợp API PayPal**

1. **Đăng ký tài khoản PayPal Developer**: Truy cập trang PayPal Developer để đăng ký tài khoản.
2. **Tạo ứng dụng**: Sau khi đăng ký, cần tạo một ứng dụng trong phần "Dashboard" để lấy Client ID và Client Secret sẽ sử dụng để xác thực với API PayPal.
3. **Chọn môi trường**: PayPal cung cấp hai môi trường cho việc phát triển và thử nghiệm:

**Sandbox**: Môi trường thử nghiệm, nơi có thể kiểm tra giao dịch mà không cần sử dụng tiền thật.

1. **Cài đặt thư viện SDK**: PayPal cung cấp SDK cho nhiều ngôn ngữ lập trình. Cài đặt SDK cho Java bằng cách thêm dependency vào file pom.xml

<dependency>

<groupId>com.paypal.sdk</groupId>

<artifactId>paypal-sdk</artifactId>

<version>1.7.1</version>

</dependency>

1. **Tạo yêu cầu thanh toán**: Sử dụng API PayPal để tạo yêu cầu thanh toán và chuyển hướng người dùng đến trang thanh toán của PayPal.
2. **Xử lý phản hồi từ PayPal**: Sau khi thanh toán hoàn tất, PayPal sẽ chuyển hướng người dùng về ứng dụng với thông tin về giao dịch. Cần xử lý phản hồi này để cập nhật trạng thái giao dịch trong hệ thống của mình.
   1. **Cài đặt tính phí giao hành bằng API Giao Hàng Nhanh**

Giao Hàng Nhanh (GHN) là một trong những dịch vụ giao hàng phổ biến tại Việt Nam, cung cấp API cho phép các ứng dụng và trang web tích hợp để tạo đơn hàng, kiểm tra trạng thái đơn hàng, và nhiều chức năng khác liên quan đến giao hàng.

**Các bước tích hợp API Giao Hàng Nhanh**

1. **Đăng ký tài khoản GHN**: Truy cập trang web Giao Hàng Nhanh để đăng ký tài khoản. Sau khi đăng ký, Nhận được mã API Key và Secret Key, dùng để xác thực các yêu cầu gửi đến API.
2. **Tài liệu API**: Truy cập tài liệu API Giao Hàng Nhanh để nắm rõ các endpoint và phương thức HTTP được hỗ trợ.
3. **Cài đặt thư viện HTTP**: Cần một thư viện để thực hiện các yêu cầu HTTP. Nếu bạn sử dụng Spring Boot, bạn có thể sử dụng RestTemplate hoặc WebClient.
4. **Gọi API Giao Hàng Nhanh**: Sử dụng API Key và Secret Key để gọi các endpoint của GHN.

Cài đặt lớp GHNLocationService để truy cập tới các phương thức lấy danh sách địa chỉ theo mức và tính phí giao hàng:

**Lấy danh sách tỉnh/thành phố**

public String getProvinces() {

HttpHeaders headers = new HttpHeaders();

headers.set("Token", apiKey);

HttpEntity<String> entity = new HttpEntity<>(headers);

ResponseEntity<String> response = restTemplate.exchange(provinceUrl, HttpMethod.GET, entity, String.class);

return response.getBody(); // Trả về danh sách tỉnh/thành phố dưới dạng JSON

}

* **getProvinces()**: Phương thức này tạo một HttpHeaders để thêm token xác thực, tạo một HttpEntity với các header, và gửi yêu cầu GET đến provinceUrl. Phản hồi được trả về dưới dạng JSON.

**Lấy danh sách quận/huyện theo tỉnh/thành phố**

public String getDistricts(int provinceId) {

HttpHeaders headers = new HttpHeaders();

headers.set("Token", apiKey);

HttpEntity<String> entity = new HttpEntity<>(headers);

String url = districtUrl + "?province\_id=" + provinceId;

ResponseEntity<String> response = restTemplate.exchange(url, HttpMethod.GET, entity, String.class);

return response.getBody(); // Trả về danh sách quận/huyện dưới dạng JSON

}

getDistricts(int provinceId): Phương thức này nhận provinceId làm tham số và gọi endpoint để lấy danh sách quận/huyện tương ứng với tỉnh/thành phố đó.

**Lấy danh sách phường/xã theo quận/huyện**

public String getWards(int districtId) {

HttpHeaders headers = new HttpHeaders();

headers.set("Token", apiKey);

HttpEntity<String> entity = new HttpEntity<>(headers);

String url = wardUrl + "?district\_id=" + districtId;

ResponseEntity<String> response = restTemplate.exchange(url, HttpMethod.GET, entity, String.class);

return response.getBody(); // Trả về danh sách phường/xã dưới dạng JSON

}

* **getWards(int districtId)**: Tương tự như các phương thức trên, phương thức này gọi endpoint để lấy danh sách phường/xã tương ứng với districtId.

**Tính phí giao hàng**

public String calculateShippingFee(int fromDistrictId, int toDistrictId, String toWardCode, int weight) {

HttpHeaders headers = new HttpHeaders();

headers.set("Token", apiKey);

headers.set("Content-Type", "application/json");

String requestBody = "{"

+ "\"from\_district\_id\": " + fromDistrictId + ","

+ "\"to\_district\_id\": " + toDistrictId + ","

+ "\"to\_ward\_code\": \"" + toWardCode + "\","

+ "\"weight\": " + weight + ","

+ "\"service\_type\_id\": 2"

+ "}";

HttpEntity<String> entity = new HttpEntity<>(requestBody, headers);

ResponseEntity<String> response = restTemplate.exchange(calculateFeeUrl, HttpMethod.POST, entity, String.class);

return response.getBody(); // Trả về phí giao hàng dưới dạng JSON

}

* **calculateShippingFee(...)**: Phương thức này gửi yêu cầu POST tới API để tính phí giao hàng. Nó tạo một JSON request body với các thông tin cần thiết như ID quận/huyện xuất phát, ID quận/huyện đích, mã phường/xã đích, trọng lượng và loại dịch vụ. Phản hồi từ API sẽ chứa thông tin về phí giao hàng.

Test API GHN:

BODY REQUEST:

{

    "from\_district\_id":1452,

    "to\_district\_id":2063,

    "to\_ward\_code":"800256",

    "weight":1000,

    "service\_type\_id":2

}

BODY RESPONSE:

{

    "code": 200,

    "message": "Success",

    "data": {

        "total": 44000,

        "service\_fee": 44000,

        "insurance\_fee": 0,

        "pick\_station\_fee": 0,

        "coupon\_value": 0,

        "r2s\_fee": 0,

        "return\_again": 0,

        "document\_return": 0,

        "double\_check": 0,

        "cod\_fee": 0,

        "pick\_remote\_areas\_fee": 0,

        "deliver\_remote\_areas\_fee": 0,

        "cod\_failed\_fee": 0

    }

}

* 1. **Cài đặt Xác Thực Hai Yếu Tố (2FA) với Google Authenticator**

Google Authenticator là một ứng dụng cho phép người dùng tạo mã xác thực tạm thời dựa trên thời gian (TOTP) hoặc số thứ tự (HOTP). Mã này được sử dụng để xác thực danh tính người dùng khi đăng nhập.

**Cách hoạt động**

* **Tạo mã bí mật (Secret Key)**: Sử dụng AuthenticatorService để tạo một mã bí mật cho người dùng.
* **Tạo URL cho Google Authenticator**: URL này sẽ được sử dụng để tạo mã QR mà người dùng có thể quét bằng ứng dụng Google Authenticator.
* **Tạo mã QR**: Sử dụng QRCodeService để tạo mã QR từ URL đã tạo.
* **Trả về mã QR**: Trả về mã QR dưới dạng hình ảnh (base64) để hiển thị trên giao diện người dùng.

# **KẾT LUẬN**

Ứng dụng quản lý web bán sách được phát triển bằng Java đã đáp ứng được các yêu cầu cơ bản của một hệ thống quản lý. Với các tính năng quản lý số lượng, thể loại sách, đơn hàng, thanh toán và báo cáo doanh thu,.. ứng dụng giúp cải thiện hiệu quả quản lý và điều web bán sách.

Đề xuất:

Mở rộng tính năng: Có thể mở rộng thêm các tính năng như thanh toán bằng thẻ ngân hang thường và notification box để đáp ứng nhu cầu của các cửa hang lớn hơn.

Cải thiện giao diện: Tiếp tục cải thiện giao diện người dùng để tăng cường trải nghiệm người dùng và làm cho ứng dụng trở nên thân thiện hơn.

Tích hợp các công nghệ mới: Xem xét tích hợp với các công nghệ mới như mobile app hoặc cloud-based solutions để mở rộng khả năng và tiện ích của ứng dụng.

Nhìn chung, dự án đã hoàn thành thành công và đáp ứng được các mục tiêu đề ra, mang lại giải pháp hiệu quả cho việc quản lý cửa hàng.

# **TÀI LIỆU THAM KHẢO**