**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP.HCM**



**Website giới thiệu, bán thiết bị máy tính.**

***Nhóm 01 - Sinh viên thực hiện***

1. Đỗ Chí Tường - 21129041

[MỤC LỤC 1](#_Toc172594149)

[DANH MỤC CÁC HÌNH VẼ 2](#_Toc172594150)

[DANH MỤC CÁC BẢNG BIỂU 3](#_Toc172594151)

[CHƯƠNG 1 : GIỚI THIỆU 4](#_Toc172594152)

[1.1 Tổng quan 4](#_Toc172594153)

[1.2 Mục tiêu đề tài 4](#_Toc172594154)

[1.3 Phạm vi đề tài 4](#_Toc172594155)

[1.4 Mô tả yêu cầu chức năng 5](#_Toc172594156)

[1.4.1 Chức năng cho người dùng không có tài khoản (Guest): 5](#_Toc172594157)

[1.4.2 Chức năng cho người dùng có tài khoản (Customer): 5](#_Toc172594158)

[1.4.3 Chức năng cho người quản trị hệ thống (Admin): 5](#_Toc172594159)

[CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT 7](#_Toc172594160)

[2.1 Kiến trúc Layer (3 tier ) 7](#_Toc172594161)

[2.2 Cloud Computing với Cloundinary 8](#_Toc172594162)

[2.3 Spring boot 9](#_Toc172594163)

[2.4 Maria DB 10](#_Toc172594164)

[CHƯƠNG 3 : PHÂN TÍCH VÀ THIẾT KẾ 11](#_Toc172594165)

[3.1 Phân tích yêu cầu bằng UML 11](#_Toc172594166)

[3.1.1 Usecase tổng quát 11](#_Toc172594167)

[3.1.2 Danh sách tác nhân và mô tả 11](#_Toc172594168)

[3.1.3 Danh sách các tình huống hoạt động (Use cases) 12](#_Toc172594169)

[3.1.4 Tình huống hoạt động 14](#_Toc172594170)

[3.2 Class diagram 26](#_Toc172594171)

[3.3 Deployment diagram 26](#_Toc172594172)

[CHƯƠNG 4 : HIỆN THỰC 28](#_Toc172594173)

[4.1 Cấu hình phần cứng, phần mềm 28](#_Toc172594174)

[4.1.1 Máy chủ (Server) 28](#_Toc172594175)

[4.1.2 Máy trạm (Client) 28](#_Toc172594176)

[4.1.3 Cấu hình phần mềm 28](#_Toc172594177)

[4.2 Giao diện của hệ thống 29](#_Toc172594178)

[4.2.1 Đăng nhập 29](#_Toc172594179)

[4.2.2 Đăng ký 30](#_Toc172594180)

[4.2.3 Danh sách sản phẩm 30](#_Toc172594181)

[4.2.4 Chi tiết sản phẩm 31](#_Toc172594182)

[4.2.5 Danh sách sản phẩm theo loại 31](#_Toc172594183)

[4.2.6 Giỏ hảng 32](#_Toc172594184)

[4.2.7 Quản lí danh mục sản phẩm 32](#_Toc172594185)

[4.2.8 Thêm danh mục 33](#_Toc172594186)

[4.2.9 Sửa danh mục 33](#_Toc172594187)

[4.2.10 Danh sách sản phẩm 34](#_Toc172594188)

[4.2.11 Thêm sản phẩm 34](#_Toc172594189)

[4.2.12 Sửa sản phẩm 35](#_Toc172594190)

[4.2.13 Danh sách người dùng 35](#_Toc172594191)

[4.2.14 Sửa người dùng 36](#_Toc172594192)

[4.2.15 Danh sách đơn hàng 36](#_Toc172594193)

[4.2.16 Chi tiết đơn hàng 37](#_Toc172594194)

[4.3 Kế hoạch và hiện thực kiểm thử hệ thống 37](#_Toc172594195)

[4.3.1 Kế hoạch kiểm thử 37](#_Toc172594196)

[4.3.2 Kiểm thử hệ thống 38](#_Toc172594197)

[CHƯƠNG 5 : KẾT LUẬN 40](#_Toc172594198)

[5.1 Kết quả đạt được 40](#_Toc172594199)

[5.2 Hạn chế của đồ án 40](#_Toc172594200)

[5.3 Hướng phát triển 41](#_Toc172594201)

[TÀI LIỆU THAM KHẢO 42](#_Toc172594202)

# : GIỚI THIỆU

## Tổng quan

Trong thời đại số hóa hiện nay, việc xây dựng và quản lý các hệ thống thương mại điện tử trở nên quan trọng hơn bao giờ hết. Một hệ thống thương mại điện tử hiệu quả không chỉ giúp người dùng mua sắm thuận tiện mà còn hỗ trợ các nhà quản trị trong việc quản lý sản phẩm, đơn hàng và người dùng. Đề tài này tập trung vào việc phát triển một ứng dụng web thương mại điện tử, cung cấp các chức năng từ cơ bản đến nâng cao, bao gồm quản lý sản phẩm, giỏ hàng, thanh toán và quản lý tài khoản người dùng.

Hệ thống được xây dựng với các công nghệ hiện đại, bao gồm Spring Boot cho phía server, Thymeleaf cho việc render giao diện người dùng, và Gradle cho quản lý phụ thuộc. Mục tiêu của hệ thống là cung cấp một nền tảng vững chắc cho cả người dùng không có tài khoản và người dùng đã đăng ký, đồng thời tạo điều kiện thuận lợi cho các quản trị viên trong việc quản lý hệ thống.

## Mục tiêu đề tài

Mục tiêu chính của đề tài này là:

* **Phát triển một hệ thống thương mại điện tử toàn diện:** Cung cấp các chức năng cơ bản như xem danh sách sản phẩm, chi tiết sản phẩm, giỏ hàng, và thanh toán cho người dùng. Đối với người quản trị, hệ thống cũng cung cấp các chức năng quản lý sản phẩm, tài khoản người dùng, và đơn hàng.
* **Tạo trải nghiệm người dùng mượt mà và thân thiện:** Đảm bảo người dùng có thể dễ dàng thực hiện các thao tác từ xem sản phẩm, thêm vào giỏ hàng, đến thanh toán và quản lý tài khoản.
* **Cung cấp công cụ quản lý hiệu quả:** Cho phép quản trị viên theo dõi và quản lý thông tin sản phẩm, tài khoản người dùng và đơn hàng một cách dễ dàng.

## Phạm vi đề tài

Đề tài này tập trung vào phát triển một hệ thống thương mại điện tử với các chức năng chính sau:

* **Chức năng cho người dùng không có tài khoản (Guest):**
  + Xem danh sách sản phẩm và chi tiết sản phẩm.
  + Chọn mua sản phẩm và xem giỏ hàng.
  + Đăng ký tài khoản mới.
* **Chức năng cho người dùng có tài khoản (Customer):**
  + Thực hiện các chức năng của Guest.
  + Xử lý thanh toán và quản lý đơn hàng.
* **Chức năng cho người quản trị hệ thống (Admin):**
  + Thực hiện các chức năng của Customer.
  + Quản lý thông tin sản phẩm, tài khoản người dùng, và đơn hàng.

## Mô tả yêu cầu chức năng

### Chức năng cho người dùng không có tài khoản (Guest):

* **Xem danh sách sản phẩm:**
  + Người dùng có thể duyệt qua danh sách sản phẩm, lọc theo loại sản phẩm, và xem thông tin cơ bản về từng sản phẩm.
* **Xem chi tiết sản phẩm:**
  + Người dùng có thể xem thông tin chi tiết của từng sản phẩm khi nhấp vào sản phẩm từ danh sách.
* **Chọn mua sản phẩm:**
  + Cho phép người dùng chọn số lượng sản phẩm và thêm vào giỏ hàng.
* **Xem giỏ hàng:**
  + Hiển thị danh sách sản phẩm đã chọn mua và cho phép chỉnh sửa số lượng hoặc xóa sản phẩm khỏi giỏ hàng.
* **Đăng ký tài khoản:**
  + Cung cấp form đăng ký tài khoản, kiểm tra tính hợp lệ và lưu thông tin vào cơ sở dữ liệu. Gửi email xác nhận và thông báo.

### Chức năng cho người dùng có tài khoản (Customer):

* **Tất cả chức năng của Guest:**
  + Cung cấp các chức năng của người dùng không có tài khoản.
* **Xử lý thanh toán:**
  + Khi người dùng thực hiện thanh toán, cập nhật thông tin vào cơ sở dữ liệu, gửi email xác nhận đơn hàng và thông báo kết quả cho người dùng. Xóa giỏ hàng trong Session sau khi thanh toán thành công.

### Chức năng cho người quản trị hệ thống (Admin):

* **Tất cả chức năng của Customer:**
  + Cung cấp các chức năng của người dùng có tài khoản.
* **Quản lý thông tin sản phẩm/loại sản phẩm:**
  + Xem danh sách và chi tiết sản phẩm/loại sản phẩm.
  + Thêm mới, cập nhật, và xóa sản phẩm/loại sản phẩm.
* **Quản lý thông tin tài khoản người dùng:**
  + Xem danh sách và chi tiết tài khoản người dùng.
  + Cập nhật và xóa tài khoản người dùng nếu chưa thực hiện đặt hàng.
* **Quản lý thông tin đơn hàng trực tuyến:**
  + Xem danh sách và chi tiết đơn hàng.
  + Cập nhật số lượng mặt hàng trong đơn hàng.

# : CƠ SỞ LÝ THUYẾT

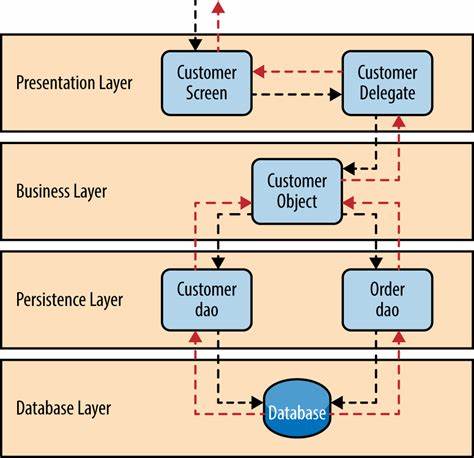
## Kiến trúc Layer (3 tier )

Kiến trúc Layer, hay còn gọi là kiến trúc phân tầng, là một mô hình thiết kế phần mềm phổ biến được sử dụng để tổ chức mã nguồn thành các tầng riêng biệt. Mỗi tầng trong kiến trúc này có một vai trò và trách nhiệm cụ thể, giúp dễ dàng quản lý, bảo trì và mở rộng hệ thống. Kiến trúc Layer thường được sử dụng trong phát triển ứng dụng doanh nghiệp và web.

#### Các Tầng Trong Kiến Trúc Layer

Một hệ thống theo kiến trúc Layer thường được chia thành các tầng chính sau:

1. **Presentation Layer (Tầng Trình Diễn):**
   * Đây là tầng tương tác với người dùng, bao gồm giao diện người dùng và các thành phần hiển thị.
   * Các công nghệ thường được sử dụng: HTML, CSS, JavaScript, Thymeleaf, JSP, React, Angular.
2. **Business Logic Layer (Tầng Logic Kinh Doanh):**
   * Tầng này chứa các logic xử lý nghiệp vụ của ứng dụng. Đây là nơi các quy tắc kinh doanh được thực thi.
   * Các công nghệ thường được sử dụng: Java, C#, Spring, .NET, EJB.
3. **Data Access Layer (Tầng Truy Cập Dữ Liệu):**
   * Tầng này chịu trách nhiệm tương tác với cơ sở dữ liệu, bao gồm các thao tác như thêm, sửa, xóa, và truy vấn dữ liệu.
   * Các công nghệ thường được sử dụng: JDBC, Hibernate, JPA, Entity Framework.
4. **Database Layer (Tầng Cơ Sở Dữ Liệu):**
   * Đây là tầng lưu trữ dữ liệu, bao gồm các hệ quản trị cơ sở dữ liệu như MySQL, SQL Server, Oracle, MongoDB.



Layer

## Cloud Computing với Cloundinary

**Cloud Computing** là mô hình cung cấp các dịch vụ máy tính qua internet, bao gồm máy chủ, lưu trữ, cơ sở dữ liệu, phân tích, mạng, phần mềm, và trí tuệ nhân tạo. Mô hình này giúp doanh nghiệp giảm chi phí đầu tư cơ sở hạ tầng và tăng khả năng mở rộng linh hoạt.

**Cloudinary** là một dịch vụ lưu trữ và quản lý nội dung đám mây chuyên biệt, đặc biệt là cho các tệp đa phương tiện như hình ảnh và video. Cloudinary cung cấp các tính năng mạnh mẽ bao gồm:

* **Lưu trữ và phân phối hình ảnh và video:** Cloudinary cho phép lưu trữ và phân phối các tệp hình ảnh và video một cách hiệu quả. Các tệp này được lưu trữ trên đám mây và có thể được truy cập từ bất kỳ đâu qua URL.
* **Xử lý hình ảnh và video:** Cloudinary cung cấp các công cụ để xử lý hình ảnh và video, bao gồm thay đổi kích thước, cắt, nén, và áp dụng các bộ lọc. Điều này giúp tối ưu hóa các tệp đa phương tiện cho các thiết bị và kích thước màn hình khác nhau.
* **Tích hợp dễ dàng:** Cloudinary dễ dàng tích hợp vào các ứng dụng web và di động, hỗ trợ nhiều ngôn ngữ lập trình và framework.
* **Bảo mật và phân quyền:** Cloudinary cung cấp các tính năng bảo mật như xác thực và phân quyền truy cập, đảm bảo rằng chỉ những người có quyền mới có thể truy cập và quản lý các tệp đa phương tiện.

**Ứng dụng trong dự án:** Trong hệ thống thương mại điện tử của bạn, Cloudinary có thể được sử dụng để quản lý hình ảnh sản phẩm. Khi người dùng tải lên hình ảnh sản phẩm mới, hình ảnh này sẽ được lưu trữ và xử lý trên Cloudinary, giúp giảm tải cho máy chủ của bạn và cải thiện hiệu suất tải trang.

## Spring boot



Hình 2‑1 Spring Boot development

**Spring Boot** là một framework dựa trên Spring Framework, được thiết kế để đơn giản hóa việc phát triển ứng dụng Java. Nó giúp bạn tạo ra các ứng dụng độc lập, sẵn sàng để triển khai với ít cấu hình hơn.

**Các tính năng chính của Spring Boot:**

* **Khởi tạo dự án nhanh chóng:** Spring Boot cung cấp các cấu hình mặc định và các bộ khởi tạo dự án (Spring Initializr) giúp bạn nhanh chóng bắt đầu với các ứng dụng Spring.
* **Tự động cấu hình:** Spring Boot tự động cấu hình các thành phần của ứng dụng dựa trên các phụ thuộc được khai báo trong dự án. Điều này giảm thiểu việc cần phải cấu hình thủ công.
* **Quản lý phụ thuộc:** Spring Boot hỗ trợ Maven và Gradle cho việc quản lý phụ thuộc. Bạn có thể dễ dàng thêm hoặc cập nhật các thư viện cần thiết cho dự án.
* **Tích hợp các dịch vụ phổ biến:** Spring Boot tích hợp dễ dàng với nhiều dịch vụ và công nghệ phổ biến như cơ sở dữ liệu, bảo mật, và các dịch vụ web.
* **Chạy ứng dụng dễ dàng:** Spring Boot cung cấp khả năng chạy ứng dụng như một ứng dụng độc lập với server tích hợp (Tomcat, Jetty, Undertow), không cần phải triển khai lên server ứng dụng bên ngoài.

**Ứng dụng trong dự án:** Spring Boot sẽ là nền tảng chính cho việc phát triển các chức năng của hệ thống thương mại điện tử. Nó sẽ giúp quản lý các dịch vụ web, tích hợp với cơ sở dữ liệu, và cung cấp các API cho giao diện người dùng.

## Maria DB

**MariaDB** là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mã nguồn mở, được phát triển từ MySQL. Nó cung cấp các tính năng mạnh mẽ và hiệu suất cao cho việc quản lý và truy xuất dữ liệu.

**Các tính năng chính của MariaDB:**

* **Hiệu suất cao:** MariaDB tối ưu hóa việc xử lý các truy vấn và hỗ trợ các tính năng như indexing và caching để cải thiện hiệu suất.
* **Tính tương thích cao:** MariaDB tương thích với MySQL, cho phép chuyển đổi dễ dàng từ MySQL sang MariaDB mà không cần phải thay đổi nhiều trong mã nguồn.
* **Bảo mật:** MariaDB cung cấp các tính năng bảo mật như mã hóa dữ liệu, xác thực người dùng, và phân quyền truy cập.
* **Mở rộng và mở rộng:** MariaDB hỗ trợ việc mở rộng quy mô và tích hợp với các công cụ phân tích và báo cáo.

**Ứng dụng trong dự án:** MariaDB sẽ được sử dụng làm cơ sở dữ liệu chính cho hệ thống thương mại điện tử của bạn, lưu trữ thông tin về sản phẩm, đơn hàng, người dùng và các dữ liệu khác. Việc lựa chọn MariaDB giúp đảm bảo tính ổn định và hiệu suất cao cho hệ thống.

# : PHÂN TÍCH VÀ THIẾT KẾ

## Phân tích yêu cầu bằng UML

### Usecase tổng quát

A diagram of a user flow

Description automatically generated

### Danh sách tác nhân và mô tả

Danh sách tác nhân và mô tả xem ở Bảng 3-1.

|  |  |
| --- | --- |
| Tác nhân | Mô tả tác nhân |
| Guest | Người dùng không có tài khoản. Tác nhân này có thể thực hiện các chức năng cơ bản như xem danh sách sản phẩm, xem chi tiết sản phẩm, chọn mua sản phẩm, và xem giỏ hàng. Họ cũng có khả năng đăng ký tài khoản mới. |
| User | Người dùng đã đăng ký tài khoản. Tác nhân này có quyền thực hiện tất cả các chức năng của người dùng không có tài khoản (User), cộng thêm khả năng xử lý thanh toán khi có sản phẩm trong giỏ hàng và đăng nhập vào hệ thống. Sau khi thanh toán, thông tin đơn hàng sẽ được cập nhật và giỏ hàng sẽ được xóa trong session. |
| Admin | Quản trị viên hệ thống. Tác nhân này có quyền truy cập và thực hiện các chức năng quản lý hệ thống, bao gồm quản lý sản phẩm, loại sản phẩm, tài khoản người dùng, và đơn hàng. Admin có thể xem, thêm mới, cập nhật, và xóa thông tin sản phẩm, loại sản phẩm, cũng như xem và quản lý các tài khoản người dùng và đơn hàng. |

Bảng 3‑1 Danh sách tác nhân và mô tả

### Danh sách các tình huống hoạt động (Use cases)

|  |  |  |
| --- | --- | --- |
| **ID** | **Tên Use case** | **Mô tả** |
| **UC01** | Xem danh sách sản phẩm | Người dùng (User) hoặc người dùng đã đăng ký (Member) có thể duyệt qua danh sách sản phẩm, bao gồm các loại như thiết bị máy tính, mỹ phẩm, quần áo, v.v. Danh sách này được lấy từ cơ sở dữ liệu (CSDL). |
| **UC02** | Xem chi tiết sản phẩm | Người dùng có thể xem thông tin chi tiết của từng sản phẩm từ danh sách sản phẩm. Thông tin chi tiết bao gồm mô tả, giá cả, và hình ảnh sản phẩm. |
| **UC03** | Thêm sản phẩm vào giỏ hàng | Người dùng có thể chọn mua sản phẩm từ trang danh sách sản phẩm hoặc trang chi tiết sản phẩm. Sản phẩm được chọn sẽ được thêm vào giỏ hàng, thông tin giỏ hàng được lưu trong biến Session. |
| **UC04** | Xem giỏ hàng | Người dùng có thể xem danh sách các sản phẩm đã chọn mua trong giỏ hàng, bao gồm thông tin về số lượng và tổng số tiền. Thông tin giỏ hàng được lưu trong Session và không cập nhật CSDL. |
| **UC05** | Chỉnh sửa số lượng sản phẩm trong giỏ hàng | Khi xem giỏ hàng, người dùng có thể thay đổi số lượng của từng sản phẩm. Nếu số lượng chỉnh sửa là 0, sản phẩm đó sẽ bị xóa khỏi giỏ hàng. |
| **UC06** | Đăng ký tài khoản | Người dùng có thể đăng ký tài khoản mới với thông tin cần thiết. Email phải là duy nhất trong hệ thống. Sau khi đăng ký thành công, thông tin được lưu vào CSDL và hệ thống gửi email xác nhận cũng như thông báo đăng ký thành công. |
| **UC07** | Xử lý thanh toán | Người dùng đã đăng ký (Member) có thể thực hiện thanh toán cho sản phẩm trong giỏ hàng. Hệ thống sẽ cập nhật thông tin đơn hàng vào CSDL, gửi email xác nhận, và thông báo đơn hàng thành công. Sau khi thanh toán thành công, thông tin giỏ hàng trong Session được xóa về null. |
| **UC08** | Quản lý loại sản phẩm | Admin có thể xem danh sách, xem chi tiết, thêm mới, cập nhật, và xóa sản phẩm |
| **UC09** | Quản lý thông tin sản phẩm | Admin có thể xem danh sách, xem chi tiết, thêm mới, cập nhật, và xóa loại sản phẩm. |
| **UC10** | Quản lý thông tin tài khoản người dùng | Admin có thể xem danh sách tài khoản người dùng, xem chi tiết tài khoản (không bao gồm mật khẩu), xóa tài khoản nếu chưa thực hiện đơn hàng online, và cập nhật thông tin tài khoản người dùng. |
| **UC11** | Quản lý thông tin đơn hàng trực tuyến | Admin có thể xem danh sách đơn hàng theo ngày mua, xem chi tiết từng đơn hàng, và cập nhật số lượng mặt hàng trong đơn hàng trực tuyến. |

Bảng 3‑2 Danh sách các tình huống trong hệ thống

### Tình huống hoạt động

Đặc tả use case.

#### UC07: Xử lý thanh toán hóa đơn

|  |  |
| --- | --- |
| **Tên use case:** Xử lý thanh toán hóa đơn. | |
| **Actor: User và Admin** | |
| **Mô tả:** Use case thực hiện việc thanh toán giỏ hàng. | |
| **Precondition:** Đăng nhập và đã thêm sản phẩm thành công. | |
| **Postcondition:** Thông tinhóa đơn được gửi về mail và lưu ở csdl. | |
| **Basic flow:** | |
| **user** | **Hệ thống** |
| 1 Ấn vào giỏ hàng  3. Thêm, xóa sản phẩm trong giỏ hàng  4. Nhấn ‘Thanh toán’. | 2. Hệ thống hiển thị giao diện ‘giỏ hàng’.  5. Hệ thống Lưu vào cơ sở dữ liệu  6. Hệ thống gửi email hóa đơn về gmail của tài khoản  7. Hệ thống hiển thị hóa đơn |
| **Alternate flow:** | |
|  |  |

Activity diagram.

A diagram of a user

Description automatically generated

Sequence diagram

A diagram of a diagram

Description automatically generated

#### UC08: Quản lý loại sản phẩm

|  |  |
| --- | --- |
| **Tên use case:** Quản lý loại sản phẩm | |
| **Actor: Admin** | |
| **Mô tả:** Use case thực hiện chức năng quản lí loại sản phẩm | |
| **Precondition:** Đăng nhập vào với tài khoản admin | |
| **Postcondition:** Thêm, xóa, Sửa loại sản phẩm | |
| **Basic flow:** | |
| **Nhân viên** | **Hệ thống** |
| 1 Chọn Quản lý loại sản phẩm  2. Chọn “thêm loại sản phẩm”  4. Nhập dữ liệu vào form  5. Nhấn “thêm” | 3. Hệ thống hiển thị giao diện Thêm loại sản phẩm  6. Hệ thống Lưu vào cơ sở dữ liệu  7. Hiển thị trang quản lý loại sản phẩm |
| **Alternate flow:** | |
| 2.1 Chọn “Xóa loại sản phẩm ”  2.1.1 Chọn “Sửa loại sản phẩm”  2.1.3 Nhập thông tin cần Sửa.  2.1.4: Ấn “Sửa” | 2.2 Hệ thống kiểm tra xem đã có sản phẩm nào thuộc loại sản phẩm này chưa  2.3 Hiển thị xóa thành công và xóa ở csdl  2.3.1 Thông báo xóa không thành công  2.1.2 Hệ thống hiển thị giao diện sửa loại sản phẩm  2.1.5. Hệ thống Lưu vào cơ sở dữ liệu  2.1.6 Hiển thị trang quản lý loại sản phẩm |

Activity diagram.

A diagram of a chat

Description automatically generated with medium confidence

Sequence diagram

A diagram of a diagram

Description automatically generated

#### UC09: Quản lý sản phẩm

|  |  |
| --- | --- |
| **Tên use case:** Quản lý sản phẩm | |
| **Actor: Admin** | |
| **Mô tả:** Use case thực hiện chức năng quản lí sản phẩm | |
| **Precondition:** Đăng nhập vào với tài khoản admin | |
| **Postcondition:** Thêm, xóa, Sửa loại sản phẩm | |
| **Basic flow:** | |
| **Nhân viên** | **Hệ thống** |
| 1 Chọn Quản lý sản phẩm  2. Chọn “thêm sản phẩm”  4. Nhập dữ liệu vào form  5. Nhấn “thêm” | 3. Hệ thống hiển thị giao diện Thêm sản phẩm  6. Hệ thống Lưu vào cơ sở dữ liệu  7. Hiển thị trang quản lý sản phẩm |
| **Alternate flow:** | |
| 2.1 Chọn “Xóa sản phẩm ”  2.1.1 Chọn “Sửa sản phẩm”  2.1.3 Nhập thông tin cần Sửa.  2.1.4: Ấn “Sửa” | 2.2 Hệ thống kiểm tra xem đã có hóa đơn nào chứa sản phẩm này chứ  2.3 Hiển thị xóa thành công và xóa ở csdl  2.3.1 Thông báo xóa không thành công  2.1.2 Hệ thống hiển thị giao diện sửa sản phẩm  2.1.5. Hệ thống Lưu vào cơ sở dữ liệu |

Activity diagram.

A diagram of a chat

Description automatically generated with medium confidence

Sequence diagram

A diagram of a diagram

Description automatically generated

#### UC10: Quản lý thông tin tài khoản người dùng.

|  |  |
| --- | --- |
| **Tên use case:** Quản lý thông tin tài khoản người dùng | |
| **Actor: Admin** | |
| **Mô tả:** Use case thực hiện chức năng quản lí loại người dùng | |
| **Precondition:** Đăng nhập vào với tài khoản admin | |
| **Postcondition:** xóa, sửa người dùng | |
| **Basic flow:** | |
| **Admin** | **Hệ thống** |
| 1 Chọn Quản lý người dùng.  2. Chọn “sửa người dùng”.  4. Nhập thông tin cần sữa.  5. Nhấn “sửa”. | 3. Hệ thống hiển thị giao diện “sửa thông tin người dùng”.  6. Hệ thống Lưu vào cơ sở dữ liệu.  7. Hiển thị trang quản lý người dùng. |
| **Alternate flow:** | |
| 2.1 Chọn “Xóa người dùng” | 2.2 Hệ thống kiểm tra xem đã có hóa đơn nào chứa người dùng này chứa  2.3 Hiển thị xóa thành công và xóa ở csdl  2.3.1 Thông báo xóa không thành công |

Activity diagram.

A diagram of a flowchart

Description automatically generated

Sequence diagram

A diagram of a project

Description automatically generated

#### UC11: Quản lý thông tin hóa đơn.

|  |  |
| --- | --- |
| **Tên use case:** Quản lý thông tin hóa đơn. | |
| **Actor: Admin** | |
| **Mô tả:** Use case thực hiện chức năng quản lí hóa đơn | |
| **Precondition:** Đăng nhập vào với tài khoản admin | |
| **Postcondition:** xóa, sửa người dùng | |
| **Basic flow:** | |
| **Admin** | **Hệ thống** |
| 1 Chọn Quản lý hóa đơn.  3. Nhấn chi tiết đơn hàng | 1. Hệ thống hiển thị giao diện danh sách người dùng. 2. Hệ thống hiển thị giao diện “Chi tiết đơn hàng”. |
| **Alternate flow:** | |
|  |  |

Activity diagram.

A diagram of a computer program

Description automatically generated with medium confidence

Sequence diagram

A diagram of a diagram

Description automatically generated

## Class diagram

Mô hình lớp (Mô hình lớp đầy đủ + Mô hình lớp bao gồm các lớp Entity)

Trong sơ đồ trên thì Cart và CartItem là lớp entity

A diagram of a computer

Description automatically generated with medium confidence

## Deployment diagram

Mô hình triển khai hệ thống.

A diagram of a computer server

Description automatically generated

# : HIỆN THỰC

## Cấu hình phần cứng, phần mềm

### ****Máy chủ (Server)****

* + **CPU:** Intel Xeon hoặc AMD EPYC, tối thiểu 4 cores.
  + **RAM:** Tối thiểu 16GB, khuyến nghị 32GB hoặc cao hơn.
  + **Ổ cứng:** SSD với dung lượng từ 500GB trở lên.
  + **Mạng:** Kết nối mạng tốc độ cao, tối thiểu 1 Gbps.

### ****Máy trạm (Client)****

* + **CPU:** Intel Core i5 hoặc AMD Ryzen 5 trở lên.
  + **RAM:** Tối thiểu 8GB.
  + **Ổ cứng:** SSD với dung lượng tối thiểu 256GB.
  + **Màn hình:** Độ phân giải tối thiểu Full HD (1920x1080).

### Cấu hình phần mềm

1. **Hệ điều hành**
   * **Server:** Ubuntu Server 20.04 hoặc CentOS 8.
   * **Client:** Windows 10/11 hoặc macOS Catalina trở lên.
2. **Phần mềm máy chủ**
   * **Web Server:** Apache Tomcat 9 hoặc Nginx.
   * **Cơ sở dữ liệu:** MariaDB
   * **Java:** OpenJDK 17 hoặc cao hơn.
   * **Thư viện và Frameworks:**
     + Spring Boot 3.3.0
     + Thymeleaf
     + Hibernate/JPA
3. **Phần mềm phát triển**
   * **IDE:** IntelliJ IDEA
   * **Build Tools:** Gradle.
4. **Công cụ kiểm thử**
   * **JUnit 5:** Dùng cho kiểm thử đơn vị.
5. **Các công cụ hỗ trợ khác**
   * **Thư viện xử lý ảnh:** commons-io, cloudinary-http44, cloudinary-taglib
   * **Email:** spring-boot-starter-mail
   * **Thymeleaf:** spring-boot-starter-thymeleaf
   * **Lombok:** Dùng để giảm boilerplate code.
   * **Devtools:** spring-boot-devtools để hỗ trợ phát triển.
   * **Cloundinarry:** H**ổ trợ lưu ảnh**

## Giao diện của hệ thống

Màn hình chức năng đã thực hiện.

### Đăng nhập

A screenshot of a computer

Description automatically generated

### Đăng ký

A screenshot of a computer

Description automatically generated

### Danh sách sản phẩm

A screenshot of a computer

Description automatically generated

### Chi tiết sản phẩm

A computer screen shot of a computer

Description automatically generated

### Danh sách sản phẩm theo loại

A screenshot of a computer

Description automatically generated

### Giỏ hảng

A screenshot of a computer

Description automatically generated

### Quản lí danh mục sản phẩm

A screenshot of a computer

Description automatically generated

### Thêm danh mục

A screenshot of a computer

Description automatically generated

### Sửa danh mục

A screenshot of a computer

Description automatically generated

### Danh sách sản phẩm

A screenshot of a computer

Description automatically generated

### Thêm sản phẩm

A screenshot of a computer

Description automatically generated

### Sửa sản phẩm

A screenshot of a computer

Description automatically generated

### Danh sách người dùng

A screenshot of a computer

Description automatically generated

### Sửa người dùng

A screenshot of a computer

Description automatically generated

### Danh sách đơn hàng

A screenshot of a computer

Description automatically generated

### Chi tiết đơn hàng

A screenshot of a computer

Description automatically generated

## Kế hoạch và hiện thực kiểm thử hệ thống

### Kế hoạch kiểm thử

**Mục tiêu kiểm thử:**

* Đảm bảo hệ thống hoạt động đúng theo yêu cầu và không có lỗi nghiêm trọng.
* Đảm bảo tất cả các chức năng được triển khai hoạt động chính xác và hiệu quả.
* Đảm bảo hệ thống có hiệu năng tốt và đáp ứng được tải công việc mong đợi.

**Phạm vi kiểm thử:**

* Kiểm thử chức năng cho các loại người dùng: guest, customer, admin.
* Kiểm thử giao diện người dùng.
* Kiểm thử hiệu năng và tải của hệ thống.
* Kiểm thử bảo mật hệ thống.

**Các loại kiểm thử:**

1. **Kiểm thử đơn vị (Unit Testing):** Kiểm thử các thành phần đơn lẻ của hệ thống, như các phương thức và class.
2. **Kiểm thử tích hợp (Integration Testing):** Kiểm thử sự tương tác giữa các thành phần và module trong hệ thống.
3. **Kiểm thử hệ thống (System Testing):** Kiểm thử toàn bộ hệ thống để đảm bảo các yêu cầu đã được đáp ứng.
4. **Kiểm thử chấp nhận (Acceptance Testing):** Kiểm thử hệ thống từ góc nhìn của người dùng cuối để đảm bảo hệ thống đáp ứng yêu cầu kinh doanh.

**Công cụ và môi trường kiểm thử:**

* **Công cụ kiểm thử:** JUnit, Selenium, JMeter.
* **Môi trường kiểm thử:** Môi trường kiểm thử giống với môi trường sản xuất bao gồm hệ điều hành, máy chủ ứng dụng, và cơ sở dữ liệu.

**Lịch trình kiểm thử:**

* **Kiểm thử đơn vị:** 1 tuần.
* **Kiểm thử tích hợp:** 2 tuần.
* **Kiểm thử hệ thống:** 2 tuần.
* **Kiểm thử chấp nhận:** 1 tuần.

**Người chịu trách nhiệm:**

* **Kiểm thử đơn vị:** Các lập trình viên.
* **Kiểm thử tích hợp:** Nhóm kiểm thử tích hợp.
* **Kiểm thử hệ thống:** Nhóm kiểm thử hệ thống.
* **Kiểm thử chấp nhận:** Người dùng cuối và nhóm kiểm thử chấp nhận.

### Kiểm thử hệ thống

**Kiểm thử chức năng:**

1. **Chức năng xem danh sách sản phẩm:**
   * **Test Case:** Người dùng không có tài khoản (guest) truy cập trang danh sách sản phẩm.
   * **Kết quả mong đợi:** Hiển thị danh sách sản phẩm từ cơ sở dữ liệu.
2. **Chức năng xem chi tiết sản phẩm:**
   * **Test Case:** Người dùng không có tài khoản (guest) truy cập trang chi tiết của một sản phẩm.
   * **Kết quả mong đợi:** Hiển thị thông tin chi tiết của sản phẩm từ cơ sở dữ liệu.
3. **Chức năng thêm sản phẩm vào giỏ hàng:**
   * **Test Case:** Người dùng không có tài khoản (guest) thêm sản phẩm vào giỏ hàng từ trang danh sách sản phẩm và trang chi tiết sản phẩm.
   * **Kết quả mong đợi:** Sản phẩm được thêm vào giỏ hàng và thông tin giỏ hàng được lưu trong session.
4. **Chức năng xem giỏ hàng:**
   * **Test Case:** Người dùng không có tài khoản (guest) truy cập trang giỏ hàng.
   * **Kết quả mong đợi:** Hiển thị danh sách sản phẩm đã thêm vào giỏ hàng và cho phép chỉnh sửa số lượng hoặc bỏ sản phẩm ra khỏi giỏ hàng.
5. **Chức năng đăng ký tài khoản:**
   * **Test Case:** Người dùng không có tài khoản (guest) đăng ký tài khoản mới với thông tin hợp lệ.
   * **Kết quả mong đợi:** Tài khoản được tạo thành công, lưu vào cơ sở dữ liệu, và gửi email xác nhận.

**Kiểm thử hiệu năng:**

* **Test Case:** Sử dụng JMeter để kiểm tra khả năng xử lý tải của hệ thống với 1000 người dùng đồng thời truy cập và thực hiện các chức năng khác nhau.
* **Kết quả mong đợi:** Hệ thống hoạt động ổn định và thời gian phản hồi nằm trong giới hạn cho phép.

**Kiểm thử bảo mật:**

* **Test Case:** Thực hiện các kiểm thử bảo mật như SQL Injection, Cross-Site Scripting (XSS), và kiểm tra tính bảo mật của các phiên làm việc.
* **Kết quả mong đợi:** Hệ thống không bị xâm nhập hoặc bị ảnh hưởng bởi các cuộc tấn công bảo mật.

**Kiểm thử chấp nhận:**

* **Test Case:** Người dùng cuối thực hiện các chức năng chính trên hệ thống và xác nhận rằng hệ thống đáp ứng yêu cầu kinh doanh.
* **Kết quả mong đợi:** Người dùng cuối chấp nhận hệ thống và không có lỗi nghiêm trọng nào được tìm thấy.

# : KẾT LUẬN

## Kết quả đạt được

Trong quá trình thực hiện đồ án, chúng tôi đã đạt được những kết quả sau:

1. **Xây dựng hệ thống web quản lý sản phẩm:**
   * Hoàn thiện các chức năng cơ bản cho người dùng không có tài khoản (guest), người dùng có tài khoản (customer), và người quản trị hệ thống (admin).
   * Người dùng có thể xem danh sách sản phẩm, xem chi tiết sản phẩm, thêm sản phẩm vào giỏ hàng, và quản lý giỏ hàng.
   * Người dùng có thể đăng ký tài khoản, xử lý thanh toán, và nhận email xác nhận đơn hàng.
2. **Tích hợp giao diện thân thiện với người dùng:**
   * Sử dụng Thymeleaf để xây dựng giao diện động.
   * Thiết kế giao diện đẹp mắt, dễ sử dụng và thân thiện với người dùng.
3. **Đảm bảo hiệu năng và bảo mật:**
   * Kiểm thử hiệu năng hệ thống với lượng người dùng lớn và đảm bảo hệ thống hoạt động ổn định.
   * Thực hiện các kiểm thử bảo mật để bảo vệ hệ thống khỏi các cuộc tấn công.
4. **Quản lý sản phẩm và người dùng:**
   * Người quản trị có thể quản lý thông tin sản phẩm, loại sản phẩm, và tài khoản người dùng.
   * Hỗ trợ tìm kiếm, xem chi tiết, thêm mới, chỉnh sửa, và xóa sản phẩm và tài khoản người dùng.
5. **Quản lý đơn hàng trực tuyến:**
   * Hệ thống cho phép xem danh sách đơn hàng, xem chi tiết đơn hàng, và quản lý trạng thái đơn hàng.

## Hạn chế của đồ án

Mặc dù đạt được nhiều kết quả tích cực, đồ án vẫn còn một số hạn chế:

1. **Chưa hoàn thiện tất cả các chức năng nâng cao:**
   * Một số chức năng nâng cao như hệ thống gợi ý sản phẩm, phân tích hành vi người dùng, và báo cáo thống kê chi tiết chưa được triển khai.
2. **Hiệu năng hệ thống chưa được tối ưu hóa hoàn toàn:**
   * Hệ thống hoạt động ổn định với số lượng người dùng vừa phải, nhưng cần tối ưu hóa thêm để đảm bảo hiệu năng tốt hơn khi lượng người dùng tăng đột biến.
3. **Giao diện người dùng còn một số điểm chưa tối ưu:**
   * Mặc dù giao diện đã thân thiện và dễ sử dụng, nhưng vẫn cần cải thiện để đáp ứng tốt hơn nhu cầu của người dùng.
4. **Bảo mật hệ thống cần được nâng cao hơn:**
   * Mặc dù đã thực hiện kiểm thử bảo mật, nhưng cần liên tục cập nhật và cải thiện để chống lại các mối đe dọa bảo mật mới.

## Hướng phát triển

Để hoàn thiện và nâng cao hệ thống, các hướng phát triển sau đây cần được thực hiện:

1. **Bổ sung và hoàn thiện các chức năng nâng cao:**
   * Triển khai hệ thống gợi ý sản phẩm dựa trên hành vi người dùng.
   * Xây dựng các báo cáo thống kê chi tiết và phân tích dữ liệu để hỗ trợ quản lý và ra quyết định.
2. **Tối ưu hóa hiệu năng hệ thống:**
   * Sử dụng các kỹ thuật và công nghệ mới để cải thiện hiệu năng và khả năng mở rộng của hệ thống.
   * Tối ưu hóa cơ sở dữ liệu và mã nguồn để giảm thời gian phản hồi và tăng tốc độ xử lý.
3. **Cải thiện giao diện người dùng:**
   * Thu thập phản hồi từ người dùng để cải thiện giao diện và trải nghiệm người dùng.
   * Nâng cấp giao diện để tương thích tốt hơn với các thiết bị di động và máy tính bảng.
4. **Nâng cao bảo mật hệ thống:**
   * Cập nhật và triển khai các biện pháp bảo mật mới để bảo vệ hệ thống khỏi các mối đe dọa mới.
   * Thực hiện kiểm tra và đánh giá bảo mật định kỳ để phát hiện và khắc phục các lỗ hổng bảo mật.
5. **Đào tạo và hỗ trợ người dùng:**
   * Tổ chức các buổi đào tạo và cung cấp tài liệu hướng dẫn sử dụng hệ thống cho người dùng.
   * Cung cấp hỗ trợ kỹ thuật và giải đáp thắc mắc cho người dùng trong quá trình sử dụng hệ thống.

Chương này tổng kết lại những kết quả đạt được, hạn chế và hướng phát triển của đồ án. Mặc dù đã đạt được nhiều kết quả tích cực, nhưng vẫn còn nhiều hạn chế cần được khắc phục. Các hướng phát triển được đề xuất nhằm hoàn thiện và nâng cao chất lượng hệ thống trong tương lai.

# TÀI LIỆU THAM KHẢO

Các tài liệu Tiếng Anh

1. Kevin Faaborg & Sandro Pasquali - Mastering Node.js Second Edition – December 2017
2. David Herron, Diogo Resende & Valentin Bojinov - Node.js Complete Reference Guide – December 2018

Các tài liệu từ Internet

1. [Spring Boot](https://spring.io/projects/spring-boot)