# MỤC LỤC

Contents

[MỤC LỤC 1](#_heading=h.gjdgxs)

[DANH MỤC BẢNG BIỂU 5](#_heading=h.1qoc8b1)

[MỞ ĐẦU 7](#_heading=h.30j0zll)

[CHƯƠNG 1: TỔNG QUAN 1](#_heading=h.1fob9te)

[1.1](#_heading=h.3znysh7) Mô tả đề tài 1

[Nội dung tìm hiểu: 1](#_heading=h.2et92p0)

[Công nghệ và công cụ: 1](#_heading=h.tyjcwt)

[1.2](#_heading=h.3dy6vkm) Yêu cầu đề tài 1

[Tìm hiểu và xây dựng web API: 1](#_heading=h.1t3h5sf)

[Phát triển Frontend: 1](#_heading=h.4d34og8)

[Chức năng : 1](#_heading=h.2s8eyo1)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 2](#_heading=h.17dp8vu)

[2.1](#_heading=h.3rdcrjn) Quy trình hoạt động cơ bản 2

[2.2](#_heading=h.26in1rg) Kiến trúc phát triển ứng dụng Controller-Service-Repository 2

[1.](#_heading=h.lnxbz9) Repository (lớp kết nối cơ sở dữ liệu): 2

[2.](#_heading=h.35nkun2) Service (lớp logic): 3

[3.](#_heading=h.1ksv4uv) Controller (lớp điều khiển và xử lý yêu cầu): 3

[2.3](#_heading=h.44sinio) Kiến trúc phát triển ứng dụng Controller-Service-Repository 3

[2.3.1](#_heading=h.2jxsxqh) JPA(Java Persistence API) 3

[2.3.2](#_heading=h.z337ya) Công cụ quản lý dự án Maven 3

[2.3.3](#_heading=h.3j2qqm3) Spring Boot framework 4

[Cấu trúc dự án Spring Boot: 5](#_heading=h.1y810tw)

[Chi tiết cấu trúc: 5](#_heading=h.2xcytpi)

[2.3.4](#_heading=h.1ci93xb) Thư viện React 6

[CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 7](#_heading=h.3whwml4)

[3.1](#_heading=h.2bn6wsx) Mô tả bài toán: 7

[3.2](#_heading=h.qsh70q) Sơ đồ use case: 8

[3.2.1](#_heading=h.1pxezwc) Danh sách các actor: 8

[3.2.2](#_heading=h.2p2csry) Danh sách các use-case 9

[3.3](#_heading=h.3o7alnk) Sơ đồ lớp: 10

[Hình 4. Sơ đồ lớp website bán giày online 10](#_heading=h.23ckvvd)

[3.3.2](#_heading=h.ihv636) Danh sách các lớp đối tượng và quan hệ 11

[3.2.3 Mô tả chi tiết từng lớp đối tượng Lớp “Account” 12](#_heading=h.1hmsyys)

[Lớp”ProductType” 12](#_heading=h.2grqrue)

[Lớp “Order” 13](#_heading=h.vx1227)

[Lớp “Sale” 13](#_heading=h.3fwokq0)

[Lớp “Product” 14](#_heading=h.4f1mdlm)

[Lớp “Feedback” 14](#_heading=h.19c6y18)

[Lớp “ProductDetail” 15](#_heading=h.28h4qwu)

[Quan hệ “OrderItem” 15](#_heading=h.37m2jsg)

[3.4](#_heading=h.46r0co2) Mô hình cấu trúc cơ sở dữ liệu 16

[3.4.1](#_heading=h.2lwamvv) Mô hình cơ sở dữ liệu 16

[3.4.2](#_heading=h.3l18frh) Mô hình dữ liệu mức logic 16

[3.5](#_heading=h.206ipza) Thiết kế API theo các chức năng: 17

[3.5.1](#_heading=h.4k668n3) Chức năng đăng ký tài khoản 17

[3.5.2](#_heading=h.1egqt2p) Quản lý giỏ hàng 17

[3.5.3](#_heading=h.sqyw64) Xem giỏ hàng 18

[3.5.4](#_heading=h.1rvwp1q) Xem tình trạng đơn hàng 19

[3.5.5](#_heading=h.2r0uhxc) Đặt hàng và hủy đơn hàng 19

[3.5.6](#_heading=h.25b2l0r) Đánh giá sản phẩm sau khi mua 20

[3.5.7](#_heading=h.34g0dwd) Quản lý tài khoản cá nhân 21

[3.5.8](#_heading=h.2iq8gzs) Đăng nhập 22

[3.5.9](#_heading=h.3hv69ve) Quản lý tình trạng đơn hàng 22

[3.5.10](#_heading=h.4h042r0) Quản lý tài khoản 23

[3.5.11](#_heading=h.3vac5uf) Quản lý các phương thức thanh toán 24

[3.5.12](#_heading=h.48pi1tg) Quản lý đơn đặt hàng 26

[3.5.15](#_heading=h.1302m92) Quản lý sản phẩm 26

[Quản lý sản phẩm: 28](#_heading=h.1gf8i83)

[Quản lý chi tiết sản phẩm: 30](#_heading=h.3ep43zb)

[3.5.16](#_heading=h.3s49zyc) Quản lý khuyến mãi: 32

[3.6](#_heading=h.45jfvxd) Kiến trúc hệ thống 34

[CHƯƠNG 4: KẾT QUẢ 35](#_heading=h.zu0gcz)

[CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 38](#_heading=h.2ce457m)

[5.1](#_heading=h.rjefff) Kết quả đạt được: 38

[5.2](#_heading=h.3bj1y38) Hướng phát triển: 38

[DANH MỤC TÀI LIỆU THAM KHẢO 39](#_heading=h.4anzqyu)

# MỞ ĐẦU

**Lý do chọn đề tài:**

Hiện nay, công nghệ ngày càng phát triển, các nhà doanh nghiệp dần chuyển sang tin học hóa các công việc quản lý, cũng như nhu cầu mua hàng online của người dùng ngày càng cao do sự tiện lợi và đa dạng lựa chọn.

Từ đó lượng khách hàng cần thuê lập trình website cũng ngày càng tăng lên, việc lập trình cũng trở nên đa dạng và phức tạp hơn. Web API (Application Programming Interface) với cách lập trình và hoạt động mới có lợi thế về các mặt như thuận tiện cho việc tích hợp các ứng dụng và dịch vụ, dễ dàng thêm mới hoặc cập nhật các tính năng,…trở nên ngày càng phổ biến. Đó là lí do nhóm chúng em thực hiện đề tài “Tìm hiểu và xây dựng hệ thống web mua bán giày”.

# CHƯƠNG 1: TỔNG QUAN

# Mô tả đề tài

## Nội dung tìm hiểu:

Nghiên cứu và hiểu rõ về khái niệm, nguyên tắc và cơ chế hoạt động của Web API trong phát triển ứng dụng web.

Xây dựng web bán giày

## Công nghệ và công cụ:

*Backend:* Sử dụng ngôn ngữ lập trình Java và Spring Boot để xây dựng và triển khai Web API.

*Database:* Sử dụng cơ sở dữ liệu như MySQL để lưu trữ và quản lý dữ liệu liên quan đến sản phẩm, đơn hàng, người dùng, và các thông tin khác.

*Frontend:* Sử dụng Reactjs để xây dựng giao diện người dùng và tương tác với Web API

# Yêu cầu đề tài

## Tìm hiểu và xây dựng web API:

Thiết kế và triển khai các API cho việc quản lý và tương tác với sản phẩm, đơn hàng, người dùng, giỏ hàng, và các tính năng khác.

Kết nối và tương tác với cơ sở dữ liệu để lưu trữ và truy xuất dữ liệu một cách hiệu quả và an toàn.

## Phát triển Frontend:

Xây dựng giao diện người dùng thân thiện, cho phép người thực hiện các tính năng cơ bản. Tích hợp và tương tác với Web API để lấy và cập nhật dữ liệu.

## Chức năng :

Cung cấp các chức năng cơ bản như xem danh sách sản phẩm, tìm kiếm, thêm vào giỏ hàng, xử lý đơn hàng, và quản lý người dùng,….Áp dụng các tiêu chuẩn bảo mật, xác thực, và phân quyền để bảo vệ dữ liệu và đảm bảo an toàn cho ứng dụng.

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

# Quy trình hoạt động cơ bản

* + - * 1. *Yêu cầu (Request):* Ứng dụng hoặc dịch vụ gửi một yêu cầu đến API thông qua endpoint và phương thức HTTP.
        2. *Xử lý yêu cầu*: API nhận yêu cầu, kiểm tra quyền truy cập, xác thực người dùng, và xử lý yêu cầu dựa trên endpoint và phương thức tương ứng.
        3. *Trả về phản hồi (Response):* Sau khi xử lý yêu cầu, API trả về một phản hồi dưới dạng dữ liệu (JSON, XML) chứa thông tin hoặc kết quả tương ứng với yêu cầu.
        4. *Xử lý phản hồi:* Ứng dụng hoặc dịch vụ nhận phản hồi từ API và xử lý dữ

## Kiến trúc phát triển ứng dụng Controller-Service-Repository

Mô hình 3 lớp với các lớp Repository, Controller, và Service là một kiến trúc phổ biến trong phát triển ứng dụng, đặc biệt là trong các ứng dụng Spring Boot và Java.

## Repository (lớp kết nối cơ sở dữ liệu):

*Mô tả:* Lớp Repository chịu trách nhiệm về việc truy cập và tương tác với cơ sở dữ liệu.

*Chức Năng:* Thực hiện các truy vấn dữ liệu đến cơ sở dữ liệu. Định nghĩa các phương thức để thêm, xóa, cập nhật, và truy vấn dữ liệu. Sử dụng các công nghệ như JPA, Hibernate, JDBC để kết nối và tương tác với cơ sở dữ liệu.

## Service (lớp logic):

*Mô tả:* Lớp Service chứa logic và quy tắc ứng dụng.

*Chức Năng:* Thực hiện các nghiệp vụ cụ thể và logic ứng dụng. Gọi các phương thức từ Repository để truy vấn hoặc cập nhật dữ liệu. Xử lý các logic phức tạp, tính toán, và quy trình kinh doanh. Cung cấp API hoặc phương thức cho Controller để xử lý yêu cầu từ giao diện người dùng.

## Controller (lớp điều khiển và xử lý yêu cầu):

*Mô tả:* Lớp Controller chịu trách nhiệm về việc nhận yêu cầu từ người dùng và phản hồi lại kết quả.

*Chức Năng:* Nhận và xử lý yêu cầu HTTP từ giao diện người dùng hoặc các nguồn khác. Gọi các phương thức từ Service để xử lý logic kinh doanh và trả về kết quả. Định nghĩa các endpoint API, xử lý định tuyến, và chuyển hướng yêu cầu tới các Service tương ứng. Trả về phản hồi cho người dùng dưới dạng dữ liệu hoặc giao diện tương tác.

## Kiến trúc phát triển ứng dụng Controller-Service-Repository

## JPA(Java Persistence API)

JPA là viết tắt của "Java Persistence API," là một phần của Java EE và Java SE, được sử dụng để quản lý truy cập cơ sở dữ liệu quan hệ thông qua các đối tượng Java. JPA cung cấp một cách tiếp cận đơn giản và đối tượng hóa cho việc tương tác với cơ sở dữ liệu, giúp giảm thiểu mã lặp và tăng cường hiệu suất và tính linh hoạt của ứng dụng.

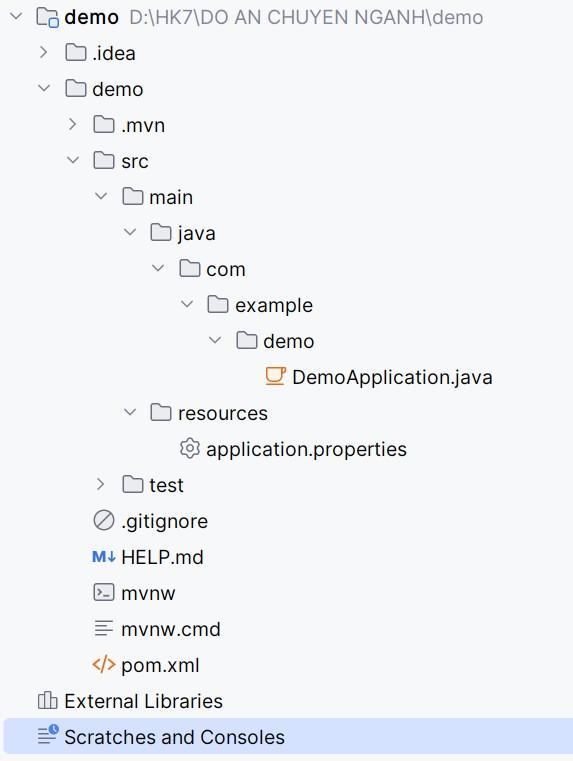
## Công cụ quản lý dự án Maven

Maven là một công cụ quản lý dự án và cấu trúc dự án dùng để xây dựng và quản lý các dự án Java. Được phát triển bởi Apache Software Foundation, Maven cung cấp một cách tiếp cận tự động hóa quá trình xây dựng, biên dịch, và quản lý phụ thuộc của ứng dụng Java.

## Spring Boot framework

Spring Boot là một trong số các module của Spring framework chuyên cung cấp các tính năng RAD (Rapid Application Development) cho phép tạo ra và phát triển các ứng dụng độc lập dựa trên Spring một cách nhanh chóng. Spring Boot ra đời với mục đích loại bỏ những cấu hình phức tạp của Spring, nó không yêu cầu cấu hình XML và nâng cao năng suất cho các nhà phát triển. …

## Cấu trúc dự án Spring Boot:



**Hình 2. Cấu trúc tổ chức thư mục của dự án Spring Boot**

## Chi tiết cấu trúc:

*src/main/java:* Đây là nơi chứa mã nguồn của ứng dụng Java.

*com.example.demo:* Một package demo ví dụ, người dùng có thể đặt tên khác tùy thuộc vào tên gói của dự án.

*src/main/resources:* Đây là nơi chứa tài nguyên không phải mã nguồn, ví dụ như tệp cấu hình, template, tệp tĩnh (như hình ảnh, CSS).

*application.properties:* Tệp cấu hình chung cho ứng dụng,ví dụ như cấu hình cơ sở dữ liệu, cổng máy chủ, v.v.

*pom.xml:* Là tệp cấu hình Maven (hoặc Gradle) cho dự án Spring Boot, chứa thông tin về các phụ thuộc, plugin, và cấu hình khác của dự án.

## Thư viện React

React là thư viện JavaScript phổ biến nhất để xây dựng giao diện người dùng (UI). Nó cho tốc độ phản hồi tuyệt vời khi user nhập liệu bằng cách sử dụng phương pháp mới để render trang web. Components của công cụ này được phát triển bởi Facebook. Nó được ra mắt như một công cụ JavaScript mã nguồn mở vào năm 2013. Hiện tại, nó đã đi trước các đối thủ chính như Angular và Bootstrap, hai thư viện JavaScript bán chạy nhất thời bấy giờ.

React là một thư viện GUI nguồn mở JavaScript tập trung vào một điều cụ thể; hoàn thành nhiệm vụ UI hiệu quả. Là lập trình viên JavaScript, người dùng sẽ dễ dàng hiểu được những điều cơ bản về React.

React cho phép người dùng sử dụng lại components đã được phát triển thành các ứng dụng khác có cùng chức năng. Tính năng tái sử dụng component là một lợi thế khác biệt cho các lập trình viên.

React component dễ viết hơn vì nó sử dụng JSX, mở rộng cú pháp tùy chọn cho JavaScript cho phép người dùng kết hợp HTML với JavaScript.

# CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

# Mô tả bài toán:

Website bán giày online cần lưu trữ những thông tin như: Lưu trữ thông tin của khách hàng khi mua hàng bao gồm: họ tên, địa chỉ, email, ngày sinh..Khách hàng có thể xem sản phẩm mà không cần đăng nhập.

Khách hàng có thể thêm sản phẩm vào giỏ hàng của bản thân và quản lí giỏ hàng đó. Khách hàng cũng có thể chọn phương thức thanh toán khi đặt hàng. Khách hàng cũng có thể đánh giá sản phẩm sau khi mua. Khách hàng có thể thay đổi địa chỉ giao hàng lúc đặt hàng.

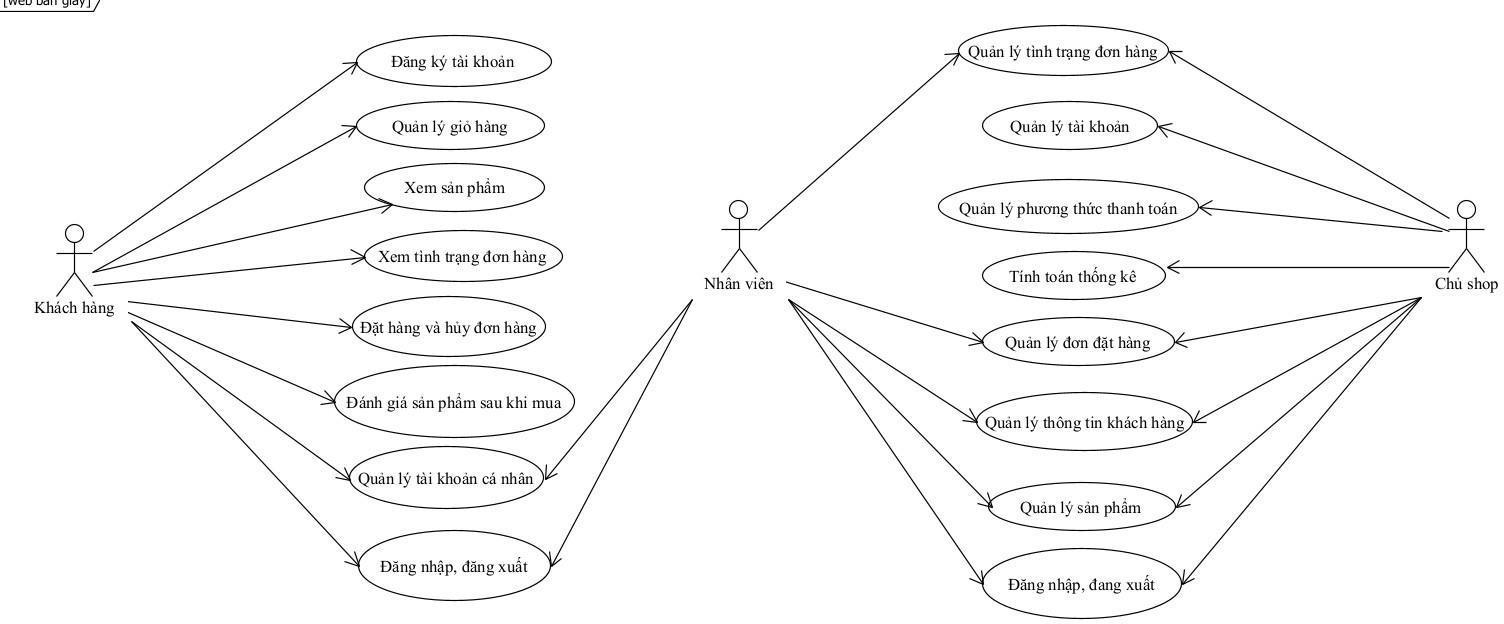
Nhân viên có nhiệm vụ quản lý, nhập thông tin các sản phẩm, sửa thông tin cũng như quản lí về chi tiết sản phẩm và xác nhận đơn hàng. . Khi khách hàng đặt hàng, nhân viên sẽ kiểm tra và chuyển trạng thái đơn hàng để xác nhận giao đơn hàng đó.

Mỗi sản phẩm thuộc một loại sản phẩm, và có thể được áp mã khuyến mãi. Vì sản phẩm kinh doanh là giày nên yêu cầu lưu trữ về màu sắc và kích cỡ của sản phẩm.

Chủ cửa hàng có thể quản lý các tài khoản của nhân viên và có thể xem các tài khoản khách hàng. Chủ cửa hàng có thể áp các mã khuyến mãi lên các sản phẩm.

Ngoài ra, chủ cửa hàng có thể thực hiện các tính toán thống kê như doanh thu trong tháng, số lượng sản phẩm bán ra trong tháng, sản phẩm bán chạy nhất.

# Sơ đồ use case:



**Hình 3. Sơ đồ use-case**

## Danh sách các actor:

| **STT** | **Tên Actor** | **Ý nghĩa/Ghi chú** |
| --- | --- | --- |
| 1 | Khách hàng | Khách hàng mua hàng online trên trang web |
| 2 | Nhân viên | Nhân viên của cửa hàng |
| 3 | Admin | Chủ cửa hàng |

**Bảng 1. Bảng danh sách các Actor**

## Danh sách các use-case

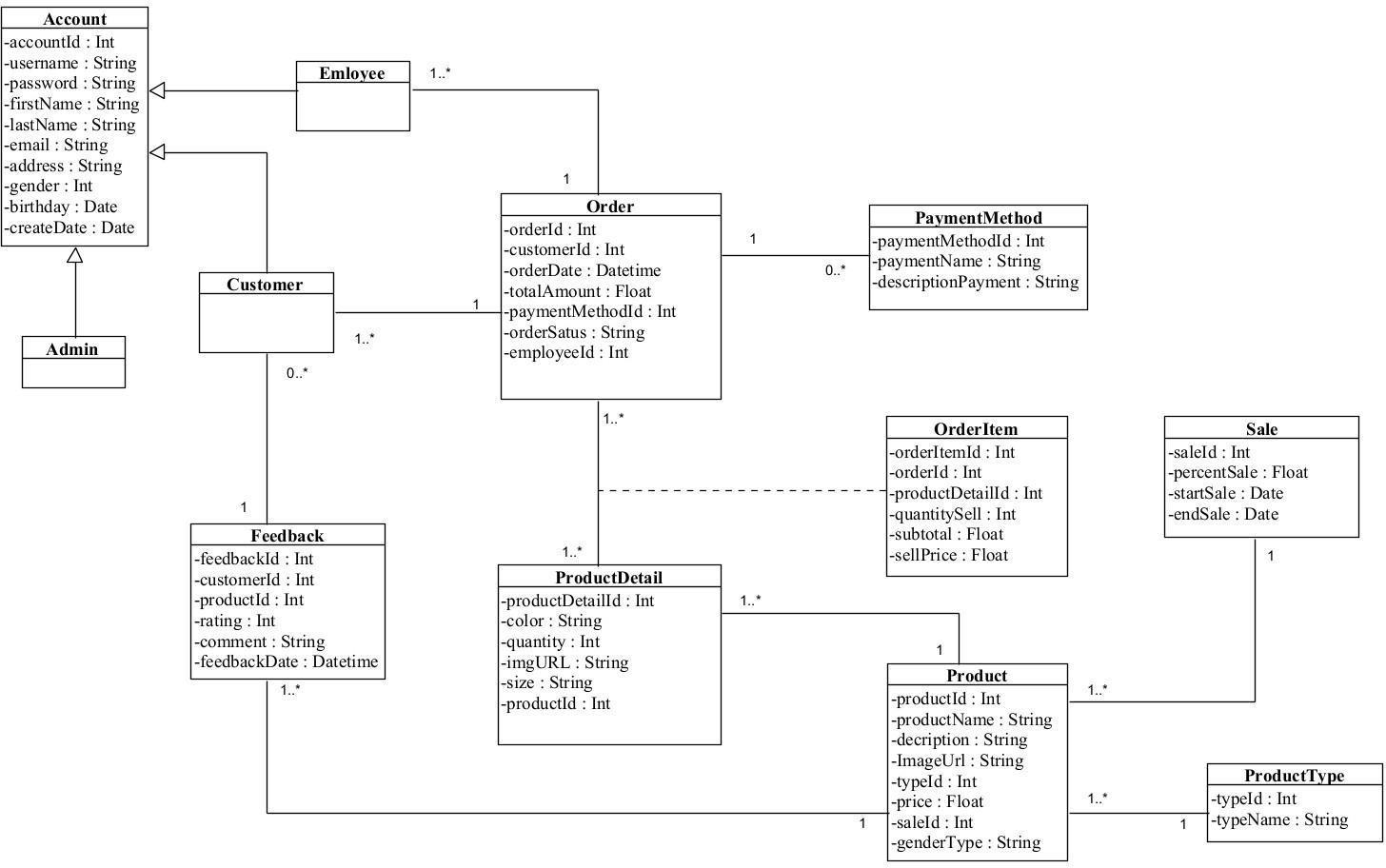
| **STT** | **Tên Use-case** | **Ý nghĩa/Ghi chú** |
| --- | --- | --- |
| 1 | Đăng ký tài khoản | Khách hàng có thể đăng ký tài khoản khi chưa có tài khoản trên trang web |
| 2 | Quản lý giỏ hàng | Khách hàng có thể thêm sản phẩm vào giỏ hàng, xóa sản phẩm khỏi giỏ hàng, chọn sản phẩm trong giỏ tiến hành đặt hàng |
| 3 | Xem giỏ hàng | Khách hàng có thể xem giỏ hàng cảu bản thân. |
| 4 | Xem tình trạng đơn hàng | Khách hàng có thể theo dõi tình trạng đơn hàng của bản thân |
| 5 | Đặt hàng và hủy đơn hàng | Khách hàng có thể đặt hàng và hủy đơn khi đơn hàng chưa được xác nhận. |
| 6 | Đánh giá sản phẩm sau khi mua | Khách hàng có thể đánh giá sản phẩm đã mua |
| 7 | Quản lý tài khoản cá nhân | Khách hàng có thể thay đổi thông tin cá nhân |
| 8 | Đăng nhập, đăng xuất | Khách hàng, nhân viên và chủ cửa hàng có thể đăng nhập, đăng xuất tài khoản |
| 9 | Quản lý tình trạng đơn hàng | Nhân viên có thể quản lý, cập nhật tình trạng đơn hàng |
| 10 | Quản lý tài khoản | Chủ cửa hàng có thể xem tất cả các tài khoản trên hệ thống và thêm sửa xóa các tài khoản nhân viên |
| 11 | Quản lý các phương thức thanh toán | Chủ cửa hàng có thể thêm, sửa xóa các phương thức thanh toán |
| 12 | Tính toán thống kê | Chủ cửa hàng có thể tính toán thống kê doanh thu |

| 13 | Quản lý đơn đặt hàng | Nhân viên và chủ cửa hàng có thể xem, update tình trạng các đơn đặt hàng |
| --- | --- | --- |
| 14 | Quản lý thông tin khách hàng | Chủ cửa hàng có thể quản lý thông tin của khách hàng |
| 15 | Quản lý sản phẩm | Nhân viên và chủ cửa hàng có thể thêm sửa xóa các sản phẩm trong cửa hàng, các loại sản phẩm và các chi tiết sản phẩm |

**Bảng 2.Bảng danh sách các use-case**

## Sơ đồ lớp:

* + 1. ***Sơ đồ lớp:***



# Hình 4. Sơ đồ lớp website bán giày online

## Danh sách các lớp đối tượng và quan hệ

| **STT** | **Tên lớp/quan hệ** | **Ý nghĩa/Ghi chú** |
| --- | --- | --- |
| 1 | Account | Lưu thông tin các tài khoản |
| 2 | Admin | Lưu thông tin các tài khoản chủ cửa hàng |
| 3 | Customer | Lưu thông tin các tài khoản khách hàng |
| 4 | Employee | Lưu thông tin các tài khoản nhân viên |
| 5 | Order | Lưu thông tin các đơn hàng và giỏ hàng của khách hàng |
| 6 | Feedback | Lưu thông tin các đánh giá của khách hàng |
| 7 | PaymentMethod | Lưu thông tin các phương thức thanh toán |
| 8 | Product | Lưu thông tin các sản phẩm |
| 9 | ProductDetail | Lưu thông tin các chi tiết sản phẩm |
| 10 | Sale | Lưu thông tin các khuyến mãi áp dụng lên sản phẩm |
| 11 | ProductType | Lưu thông tin các loại sản phẩm |
| 12 | OrderItem | Lưu thông tin chi tiết đơn hàng và giỏ hàng của khách hàng |

**Bảng 3. Bảng danh sách các lớp đối tượng và quan hệ**

## 3.2.3 Mô tả chi tiết từng lớp đối tượng Lớp “Account”

| **STT** | **Tên thuộc tính** | **Kiểu giá trị** | **Loại** | **Ý nghĩa/ghi chú** |
| --- | --- | --- | --- | --- |
| 1 | accountID | int | Private | Lưu trữ mã của mỗi tài khoản |
| 2 | username | String | Private | Lưu trữ tên đăng nhập của mỗi tài khoản |
| 3 | password | String | private | Lưu trữ mật khẩu |
| 4 | firstName | String | Private | Lưu trữ họ của chủ tài khoản |
| 5 | lastName | String | Private | Lưu trữ tên của chủ tài khoản |
| 6 | email | String | private | Lưu trữ email của chủ tài khoản |
| 7 | address | String | Private | Lưu trữ địa chỉ của chủ tài khoản |
| 8 | gender | enum(“female”, “male”, “unknow”) | Private | Lưu trữ giới tính của chủ tài khoản |
| 9 | birthday | Date | Private | Lưu trữ ngày sinh của chủ tài khoản |
| 10 | createDate | Date | Private | Lưu trữ ngày tạo tài khoản |

**Bảng 4. Bảng mô tả chi tiết lớp Account**

***Lớp “Admin”, Lớp “Customer”, Lớp “Employee” :*** 3 lớp này đều kế thừa từ lớp Account, thuộc tính đều kế thừa từ lớp Account.

## Lớp”ProductType”

| **STT** | **Tên thuộc tính** | **Kiểu giá trị** | **Loại** | **Ý nghĩa/ghi chú** |
| --- | --- | --- | --- | --- |
| 1 | typeId | Int | Private | Lưu trữ mã loại sản phẩm |
| 2 | typeName | String | Private | Lưu trữ tên loại sản phẩm |

**Bảng 5 Bảng mô tả chi tiết lớp ProductType**

# Lớp “Order”

| **STT** | **Tên thuộc tính** | **Kiểu giá trị** | **Loại** | **Ý nghĩa/ghi chú** |
| --- | --- | --- | --- | --- |
| 1 | orderID | Int | Private | Lưu trữ mã đơn hàng/ giỏ hàng |
| 2 | customerID | Int | Private | Lưu trữ mã khách hàng |
| 3 | orderDate | Date | private | Lưu trữ ngày tạo giỏ/ đặt hàng |
| 4 | PaymentMethodId | Int | Private | Lưu trữ mã phương thức thanh toán |
| 5 | orderStatus | enum('added\_to\_card', 'to\_pay', 'to\_receive', 'completed', 'canceled') | Private | Lưu trữ tình trạng đơn hàng |
| 6 | employeeId | int | Private | Lưu trữ mã nhân viên xác nhận đơn |

**Bảng 6 Bảng mô tả chi tiết lớp Order**

# Lớp “Sale”

| **STT** | **Tên thuộc tính** | **Kiểu giá trị** | **Loại** | **Ý nghĩa/ghi chú** |
| --- | --- | --- | --- | --- |
| 1 | saleId | Int | Private | Lưu trữ mã khuyến mãi |
| 2 | percentSale | Float | Private | Lưu trữ phần trăm khuyến mãi |

**Bảng 5 Bảng mô tả chi tiết lớp Sale**

# Lớp “Product”

| **STT** | **Tên thuộc tính** | **Kiểu giá trị** | **Loại** | **Ý nghĩa/ghi chú** |
| --- | --- | --- | --- | --- |
| 1 | productId | Int | Private | Lưu trữ mã sản phẩm |
| 2 | productName | String | Private | Lưu trữ tên sản phẩm |
| 3 | decription | String | Private | Lưu trữ mô tả sản phẩm |
| 4 | imageUrl | String | Private | Lưu trữ đường dẫn hình ảnh |
| 5 | typeId | Int | Private | Lưu trữ mã loại |
| 6 | price | Float | Private | Lưu trữ giá gốc của sản phẩm |
| 7 | saleId | Int | Private | Lưu trữ mã khuyến mãi(nếu có) |
| 8 | genderType | enum('male','female',' unisex') | Private | Lưu trữ thông tin sản phẩm phù hợp với giới tính người dùng nào |

**Bảng 6 Bảng mô tả chi tiết lớp Product**

# Lớp “Feedback”

| **STT** | **Tên thuộc tính** | **Kiểu gái trị** | **Loại** | **Ý nghĩa/ghi chú** |
| --- | --- | --- | --- | --- |
| 1 | feedbackId | Int | Private | Lưu trữ mã đánh giá |
| 2 | customerId | Int | Private | Lưu trữ mã khách hàng |
| 3 | productId | Int | Private | Lưu trữ mã sản phẩm |
| 4 | rating | Enum( 'very\_bad', 'bad',  'average', 'good', 'excellent') | Private | Lưu trữ mức độ hài lòng |
| 5 | comment | String | Private | Lưu trữ bình luận đánh giá |
| 6 | feedbackDate | Date | Private | Lưu trữ ngày đánh giá |

**Bảng 7 Bảng mô tả chi tiết quan hệ Feedback**

# Lớp “ProductDetail”

| **STT** | **Tên thuộc tính** | **Kiểu giá trị** | **Loại** | **Ý nghĩa/ghi chú** |
| --- | --- | --- | --- | --- |
| 1 | productDetailId | Int | Private | Lưu trữ mã chi tiết sản phẩm |
| 2 | color | String | Private | Lưu trữ màu chi tiết sản phẩm |
| 3 | quantity | Int | Private | Lưu trữ số lượng chi tiết sản phẩm |
| 4 | ImgURL | String | Private | Lưu trữ đường dẫn hình ảnh chi tiết sản phẩm |
| 5 | size | String | Private | Lưu trữ đường dẫn hình ảnh chi tiết sản phẩm |
| 6 | productId | Int | Private | Lưu trữ mã sản phẩm |

**Bảng 8. Bảng mô tả chi tiết quan hệ ProductDetail**

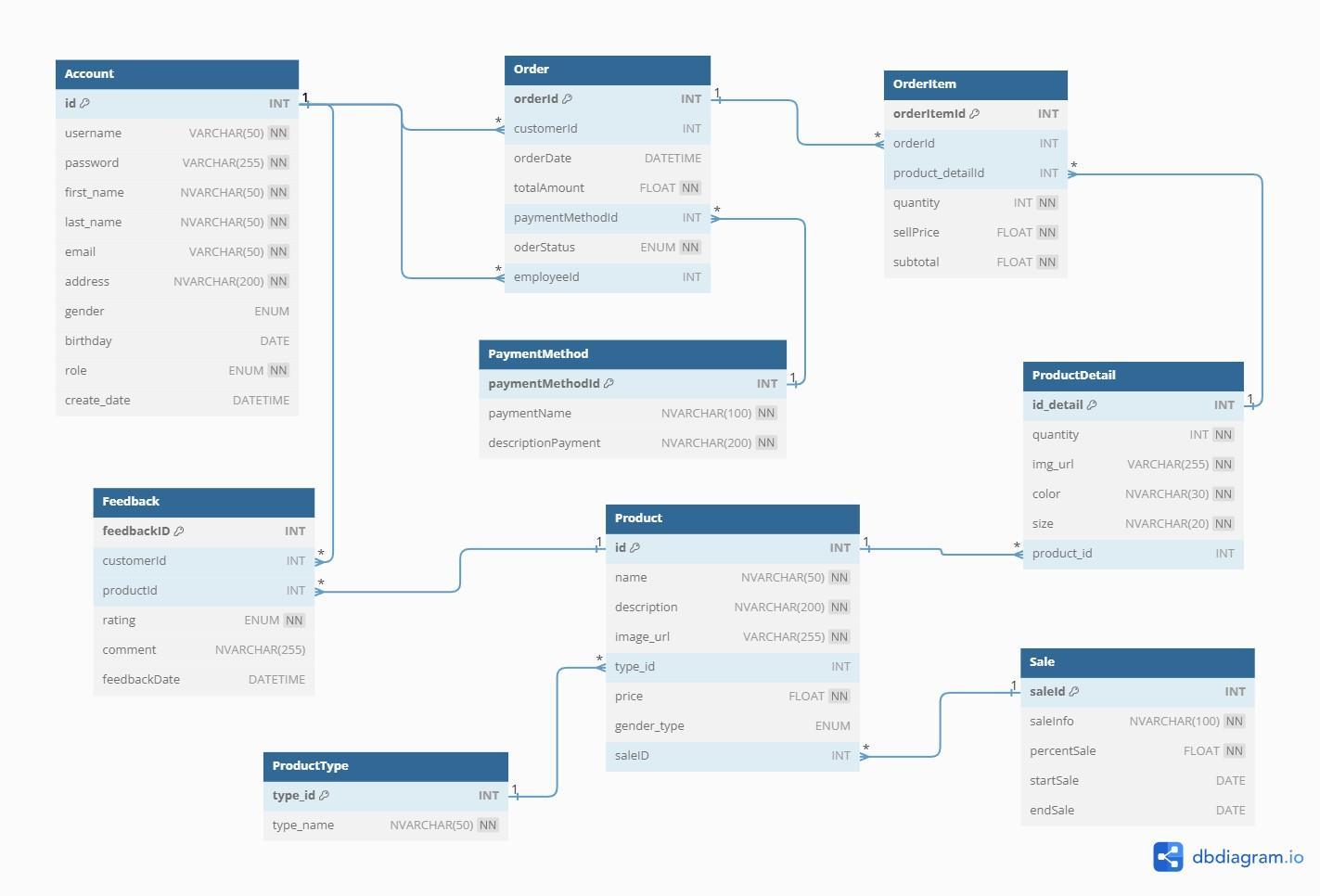
# Quan hệ “OrderItem”

| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Loại** | **Ý nghĩa/ghi chú** |
| --- | --- | --- | --- | --- |
| 1 | orderItemId | Int | Private | Lưu trữ mã chi tiết đơn hàng/ giỏ hàng |
| 2 | orderId | Int | Private | Lưu trữ mã đơn hàng |
| 3 | productDetailId | Int | Private | Lưu trữ mã chi tiết sản phẩm |
| 4 | quantitySell | Int | Private | Lưu trữ số lượng chi tiết bán |
| 5 | subtotal | Float | Private | Lưu trữ thành tiền của chi tiết |
| 6 | sellPrice | Float | Private | Lưu trữ giá bán của sản phẩm tại thời điểm đặt hàng |

**Bảng 9. Bảng mô tả chi tiết quan hệ OrderItem**

## Mô hình cấu trúc cơ sở dữ liệu

## Mô hình cơ sở dữ liệu



**Hình 5. Mô hình cấu trúc dữ liệu**

## Mô hình dữ liệu mức logic

Account ( **id,** username, password, first\_name, last\_name, email, address, gender, birthday, role, create\_date)

ProductType ( **type\_id,** type\_name) Sale ( **saleId**, saleInfo, percentSale)

Product( **id,** name, description, image\_url, price, gender\_type, *saleID, type\_id*) ProductDetail( **id\_detail**, quantity, img\_url, color, size, *product\_id* ) PaymentMethod( **paymentMethodId,** paymentName, descriptionPayment)

Order ( **orderId**, orderDate, totalAmount, *paymentMethodId, employeeId, customerId*) OrderItem ( **orderItemId**, quantity, sellPrice, subtotal, *orderId, product\_detailId*) Feedback (**feedbackID**, rating, comment, feedbackDate, *customerId, productId*)

# Thiết kế API theo các chức năng:

## Chức năng đăng ký tài khoản

| Phân quyền: | Tất cả người dùng |
| --- | --- |
| Url: | http://localhost:8080/api/v1/customers/register |
| Input:(Body) | {  "username": "dinhdinh", "password": "123456", "firstName": "pham quyen", "lastName": "dinh", "address": "p5, tp Trà Vinh", "birthday" : "2002-02-10",  "email": ["pdin](mailto:pdinh0823@gmail.com)h[0823@gmail.com](mailto:pdinh0823@gmail.com)", "role": "CUSTOMER",  "gender":"FEMALE"  } |
| Output: | Create successfully |
| Phương thức | POST |
| Authorization | Không có |

**Bảng 10. Diễn giải API đăng ký tài khoản khách hàng**

## Quản lý giỏ hàng

*Xóa sản phẩm khỏi giỏ hàng:*

| Phân quyền: | Khách hàng |
| --- | --- |
| Url: | http://localhost:8080/api/v1/orderItems/delete/{id} |
| Input:(Head) | Id (id chi tiết giỏ hàng) |
| Output: | Delete successfull! |
| Phương thức | DELETE |
| Authorization | Basic Auth (username, password) |

**Bảng 11. Diễn giải API xóa sản phẩm khỏi giỏ hàng**

*Thêm sản phẩm vào giỏ hàng:*

| Phân quyền: | Khách hàng |
| --- | --- |
| Url: | http://localhost:8080/api/v1/orderItems/create |
| Input: | No |
| Output: | Delete successfull! |
| Phương thức | POST |
| Authorization | Basic Auth (username, password) |

**Bảng 12 Diễn giải API thêm sản phẩm vào giỏ hàng**

## Xem giỏ hàng

| Phân quyền: | Khách hàng |
| --- | --- |
| Url: | http://localhost:8080/api/v1/orders/getCartByCustomer/{id} |
| Input:(Head) | Id (id account khách hàng) |
| Output: | [  {  "orderItem\_id": 12,  "quantity": 1,  "subtotal": 315000.0,  "price": 315000.0,  "size": "40",  "name": "Giày thể thao thời trang", "brand": "Sneakers",  "color": "Black",  "imgSrc": "https://firebasestorage.googleapis.com/v0/b... "  },……  ] |
| Phương thức | GET |
| Authorization | Không có |

**Bảng 13 Diễn giải API xem giỏ hàng**

## Xem tình trạng đơn hàng

| Phân quyền: | Khách hàng |
| --- | --- |
| Url: | http://localhost:8080/api/v1/orders/status |
| Input | Không |
| Output: | { "content": [ {  "total\_amount": 350000,  "oder\_date": "2024-01-03T15:37:05.000+00:00",  "oderStatus": "TO PAY", "customer\_id": 4,  "employee\_id": 3,  "payment\_method\_id": 1  },….] } |
| Phương thức | GET |
| Authorization | Basic Auth (username, password) |

**Bảng 14 Diễn giải API xem tình trạng đơn hàng**

## Đặt hàng và hủy đơn hàng

*Đặt hàng:*

| Phân quyền: | Khách hàng |
| --- | --- |
| Url: | http://localhost:8080/api/v1/orders/create?ids=4,5 |
| Input (Head, Body) | Ids (danh sách các id của chi tiết giỏ hàng) Body{  “recipientAddress”: “phường 5, thành phố Trà Vinh”  } |
| Output: | Create succesfully |
| Phương thức | POST |
| Authorization | Basic Auth (username, password) |

**Bảng 15. Diễn giải API đặt hàng**

*Hủy đơn hàng:*

| Phân quyền: | Khách hàng |
| --- | --- |
| Url: | http://localhost:8080/api/v1/orders/cancel/{id} |
| Input (Head) | Id (id đơn hàng) |
| Output: | Update succesfully |
| Phương thức | PUT |
| Authorization | Basic Authen(username, password) |

**Bảng 16. Diễn giải API hủy đơn hàng**

## Đánh giá sản phẩm sau khi mua

| Phân quyền: | Khách hàng |
| --- | --- |
| Url: | http://localhost:8080/api/v1/feedbacks |
| Input:(Body) | {  "comment": "It's very beatifull", "rating": "GOOD",  "customer\_id": "3",  "product\_id": "2"  } |
| Output: | Feedback succesfully! |
| Phương thức | POST |
| Authorization | Basic Auth (username, password) |

**Bảng 17. Diễn giải API đánh giá sản phẩm**

## Quản lý tài khoản cá nhân

*Xem thông tin cá nhân:*

| Phân quyền: | Khách hàng, Nhân viên, Chủ cửa hàng |
| --- | --- |
| Url: | http://localhost:8080/api/v1/customers/{id} |
| Input:(Head) | Id (id account khách hàng) |
| Output: | { "id": 4,  "username": "yenphuong", "address": "Phường 1, TP Trà Vinh", "birthday": "2002-10-20",  "email": ["yenphuong](mailto:yenphuong@gmail.com)@[gmail.com",](mailto:yenphuong@gmail.com) "gender": "FEMALE",  "createdDate": "2024-01-03T15:37:05.000+00:00"} |
| Phương thức | GET |
| Authorization | Basic Auth (username, password) |

**Bảng 18. Diễn giải API xem thông tin cá nhân**

*Sửa thông tin cá nhân:*

| Phân quyền: | Khách hàng |
| --- | --- |
| Url: | http://localhost:8080/api/v1/customers/update/{id} |
| Input:(Head & Body) | Head: id account khách hàng Body:{ "password": "12345",  "firstName": "nguyen", "lastName": "kha", "address": "trà vinh", "birthday" : "2002-06-15",  "email": "pdinh",  "gender":"FEMALE"} |
| Output: | Update successfully! |
| Phương thức | PUT |
| Authorization | Basic Auth (username, password) |

**Bảng 19. Diễn giải API sửa thông tin cá nhân**

## Đăng nhập

| Phân quyền: | Tất cả người dùng |
| --- | --- |
| Url: | http://localhost:8080/api/v1/login |
| Input: | Không |
| Output: | {  "id": 4,  "fullName": "Yến Phượng", "role": "CUSTOMER"  } |
| Phương thức | GET |
| Authorization | Basic Authen(username, password) |

**Bảng 20. Diễn giải API đăng nhập**

## Quản lý tình trạng đơn hàng

*Cập nhật tình trạng đơn hàng:*

| Phân quyền: | Employee (Nhân viên) |
| --- | --- |
| Url: | http://localhost:8080/api/v1/orders/updateStatus/{id} |
| Input:(Head & Body) | Head: Id (id đơn hàng) Body:  {  "oderStatus": "TO\_RECEIVE", "employee\_id": "2"  } |
| Output: | Update successfully! |
| Phương thức | PUT |
| Authorization | Basic Authen(username, password) |

**Bảng 21. Diễn giải API cập nhật tình trạng đơn hàng**

## Quản lý tài khoản

*Xem tất cả tài khoản:*

| Phân quyền: | Admin (Chủ cửa hàng) |
| --- | --- |
| Url: | http://localhost:8080/api/v1/accounts |
| Input | Không |
| Output: | {  "content": [  {  "username": "admin", "createdDate": "03-01-2024", "role": "ADMIN",  "fullName": "User"  },….]} |
| Phương thức | GET |
| Authorization | Basic Authen(username, password) |

**Bảng 22. Diễn giải API xem tất cả tài khoản**

*Tạo tài khoản nhân viên:*

| Phân quyền: | Admin (Chủ cửa hàng) |
| --- | --- |
| Url: | http://localhost:8080/api/v1/employees/create |
| Input(Body) | { "username": "khanguyen", "password": "12345", "firstName": "nguyen", "lastName": "kha", "address": "trà vinh", "birthday" : "1995-05-15",  "email": "[nguyenkha@gmail.com](mailto:nguyenkha@gmail.com)", "gender":"MALE"} |
| Output: | Create successfully! |
| Phương thức | POST |
| Authorization | Basic Authen(username, password) |

**Bảng 23. Diễn giải API tạo tài khoản nhân viên**

## Quản lý các phương thức thanh toán

*Xem các phương thức thanh toán:*

| Phân quyền: | Admin(Chủ cửa hàng) |
| --- | --- |
| Url: | http://localhost:8080/api/v1/paymentMethods |
| Input | Không |
| Output: | {  "content": [  {  "id": 1,  "name": "Credit Card",  "description\_payment": "Pay with your credit card"  },….]} |
| Phương thức | GET |
| Authorization | Basic Authen(username, password) |

**Bảng 24. Diễn giải API xem các phương thức thanh toán**

*Thêm phương thức thanh toán:*

| Phân quyền: | Admin(Chủ cửa hàng) |
| --- | --- |
| Url: | http://localhost:8080/api/v1/paymentMethods |
| Input (Body) | {  "name": "test payment", "description\_payment" : "thanh toan qua the"  } |
| Output: | Create successfully! |
| Phương thức | POST |
| Authorization | Basic Authen(username, password) |

**Bảng 25. Diễn giải API thêm phương thức thanh toán**

*Sửa phương thức thanh toán:*

| Phân quyền: | Admin(Chủ cửa hàng) |
| --- | --- |
| Url: | http://localhost:8080/api/v1/paymentMethods |
| Input (Body) | {  "name": "test payment", "description\_payment" : "thanh toan qua the"  } |
| Output: | Update successfully! |
| Phương thức | PUT |
| Authorization | Basic Authen(username, password) |

**Bảng 26. Diễn giải API sửa phương thức thanh toán**

*Xóa phương thức thanh toán: (Tất cả các đơn hàng đã chứa phương thức này sẽ null tại giá trị mã phương thức*)

| Phân quyền: | Admin(Chủ cửa hàng) |
| --- | --- |
| Url: | http://localhost:8080/api/v1/paymentMethods/{id} |
| Input (Head) | Id phương thức |
| Output: | Delete successfully! |
| Phương thức | DELETE |
| Authorization | Basic Authen(username, password) |

**Bảng 27. Diễn giải API xóa phương thức thanh toán**

## Quản lý đơn đặt hàng

*Xem tất cả đơn đặt hàng:*

| Phân quyền: | Admin(Chủ cửa hàng), Nhân viên(Customer) |
| --- | --- |
| Url: | http://localhost:8080/api/v1/orders/all |
| Input | Không |
| Output: | {  "content": [  {  "total\_amount": 0.0,  "oder\_date": "2024-01-03T15:37:05.000+00:00",  "oderStatus": "TO\_PAY", "customer\_id": 4,  "employee\_id": 0,  "payment\_method\_id": 0  },..]} |
| Phương thức | GET |
| Authorization | Basic Authen(username, password) |

**Bảng 28. Diễn giải API xem tất cả đơn hàng**

## Quản lý sản phẩm

***Quản lý loại sản phẩm:***

*Xem tất cả loại sản phẩm:*

| Phân quyền: | Tất cả người dùng |
| --- | --- |
| Url: | http://localhost:8080/api/v1/productTypes |
| Input | Không |
| Output: | [{ "id": 3,  "name": "Boots"},…..] |
| Phương thức | GET |
| Authorization | Không |

**Bảng 29. Diễn giải API xem tất cả loại sản phẩm**

*Thêm loại sản phẩm:*

| Phân quyền: | Chủ cửa hàng, Nhân viên |
| --- | --- |
| Url: | http://localhost:8080/api/v1/productTypes |
| Input:(Body) | { "name": "bata" } |
| Output: | Create successfully! |
| Phương thức | POST |
| Authorization | Basic Authen(username, password) |

**Bảng 30. Diễn giải API thêm loại sản phẩm**

*Sửa loại sản phẩm:*

| Phân quyền: | Chủ cửa hàng, Nhân viên |
| --- | --- |
| Url: | http://localhost:8080/api/v1/productTypes/{id} |
| Input:(Head, Body) | Head: id loại Body:  {  "name": "bata"  } |
| Output: | Update successfully! |
| Phương thức | PUT |
| Authorization | Basic Authen(username, password) |

**Bảng 31. Diễn giải API sửa loại sản phẩm**

*Xóa loại sản phẩm: (Khi xóa, nếu sản phẩm nào có mã loại này sẽ được gán mã loại null)*

| Phân quyền: | Chủ cửa hàng, Nhân viên |
| --- | --- |
| Url: | http://localhost:8080/api/v1/productTypes/{id} |
| Input:(Head) | Id loại |
| Output: | Delete successfully! |
| Phương thức | DELETE |
| Authorization | Basic Authen(username, password) |

**Bảng 32. Diễn giải API xóa loại sản phẩm**

## Quản lý sản phẩm:

*Xem tất cả sản phẩm:*

| Phân quyền: | Tất cả người dùng |
| --- | --- |
| Url: | http://localhost:8080/api/v1/products/full |
| Input | Không |
| Output: | [ {  "id": 1,  "name": "Giày thể thao thời trang", "description": "Giày thể thao nam nữ,….”  "image\_url": "https://firebasestorage.googleapis. ",  "price": 315000.0,  "type\_name": "Sneakers", "type\_id": 0,  "sale\_percent": 0.0,  "sale\_id": 0.0, "gender\_for": "UNISEX"  },…] |
| Phương thức | GET |
| Authorization | Không |

**Bảng 33. Diễn giải API xem tất cả sản phẩm**

*Thêm sản phẩm:*

| Phân quyền: | Chủ cửa hàng, Nhân viên |
| --- | --- |
| Url: | http://localhost:8080/api/v1/products/create |
| Input:(Body) | {"name": "T-shirt",  "description": "Description for Product1", "image\_url": "product1.jpg",  "price": 29.99,  "type\_id": 1,  "genderType" :"FEMALE", "sale\_id" : "0"} |
| Output: | Create successfully! |
| Phương thức | POST |
| Authorization | Basic Authen(username, password) |

**Bảng 34. Diễn giải API thêm sản phẩm**

*Sửa sản phẩm:*

| Phân quyền: | Chủ cửa hàng, Nhân viên |
| --- | --- |
| Url: | http://localhost:8080/api/v1/products/update/{id} |
| Input:(Head, Body) | Head: id sản phẩm Body  {"name": "T-shirt",  "description": "Description for Product1", "image\_url": "product1.jpg",  "price": 29.99,  "type\_id": 1,  "genderType" :"FEMALE", "sale\_id" : "0"} |
| Output: | Update successfully! |
| Phương thức | PUT |
| Authorization | Basic Authen(username, password) |

**Bảng 35. Diễn giải API sửa sản phẩm**

## Quản lý chi tiết sản phẩm:

*Xem chi tiết sản phẩm theo sản phẩm:*

| Phân quyền: | Tất cả người dùng |
| --- | --- |
| Url: | http://localhost:8080/api/v1/products/productDetail/{id} |
| Input(Head) | Id sản phẩm |
| Output: | [ {"id": 1,  "quantity": 2,  "img\_url": "https://firebasestorage. ",  "color": "White",  "size": "38",  "product\_id": 1},..] |
| Phương thức | GET |
| Authorization | Không |

**Bảng 36. Diễn giải API xem chi tiết sản phẩm theo sản phẩm**

*Thêm chi tiết sản phẩm:*

| Phân quyền: | Chủ cửa hàng, Nhân viên |
| --- | --- |
| Url: | http://localhost:8080/api/v1/productDetails/create |
| Input:(Body) | {"quantity": 5,  "img\_url": "running\_shoes\_blue\_42.jpg", "color": "red",  "size": "41",  "product\_id": 1} |
| Output: | Create successfully! |
| Phương thức | POST |
| Authorization | Basic Authen(username, password) |

**Bảng 37. Diễn giải API thêm chi tiết sản phẩm**

*Sửa chi tiết sản phẩm:*

| Phân quyền: | Chủ cửa hàng, Nhân viên |
| --- | --- |
| Url: | http://localhost:8080/api/v1/products/update/{id} |
| Input:(Head, Body) | Head: id chi tiết sản phẩm Body  {  "quantity": 10,  "img\_url": "running\_shoes\_42.jpg", "color": "red",  "size": "41"  } |
| Output: | Update successfully! |
| Phương thức | PUT |
| Authorization | Basic Authen(username, password) |

**Bảng 38. Diễn giải API sửa chi tiết sản phẩm**

*Xóa chi tiết sản phẩm: (Khi xóa, nếu đơn hàng nào có mã sản phẩmnày sẽ được gán mã sản phẩm null, nếu tồn tại trong giỏ hàng thì xóa khỏi giỏ hàng)*

| Phân quyền: | Chủ cửa hàng, Nhân viên |
| --- | --- |
| Url: | http://localhost:8080/api/v1/productDetails/{id} |
| Input:(Head) | Id chi tiết sản phẩm |
| Output: | Delete successfully! |
| Phương thức | DELETE |
| Authorization | Basic Authen(username, password) |

**Bảng 39. Diễn giải API xóa chi tiết sản phẩm**

## Quản lý khuyến mãi:

*Xem tất cả khuyến mãi:*

| Phân quyền: | Chủ cửa hàng |
| --- | --- |
| Url: | http://localhost:8080/api/v1/sales |
| Input | Không |
| Output: | [{  "id": 1,  "sale\_info": "Summer Sale", "percent\_sale": 15.0  },…] |
| Phương thức | GET |
| Authorization | Basic Authen(username, password) |

**Bảng 40. Diễn giải API xem tất cả khuyến mãi**

*Thêm khuyến mãi:*

| Phân quyền: | Chủ cửa hàng |
| --- | --- |
| Url: | http://localhost:8080/api/v1/sales/create |
| Input:(Body) | {  "sale\_info": " Sale", "percent\_sale": 15.0  } |
| Output: | Create successfully! |
| Phương thức | POST |
| Authorization | Basic Authen(username, password) |

**Bảng 41. Diễn giải API thêm khuyến mãi**

*Sửa khuyến mãi:*

| Phân quyền: | Chủ cửa hàng |
| --- | --- |
| Url: | http://localhost:8080/api/v1/sales/update/{id} |
| Input:(Head, Body) | Head: id khuyến mãi Body  {  "sale\_info": " Sale", "percent\_sale": 15.0 } |
| Output: | Update successfully! |
| Phương thức | PUT |
| Authorization | Basic Authen(username, password) |

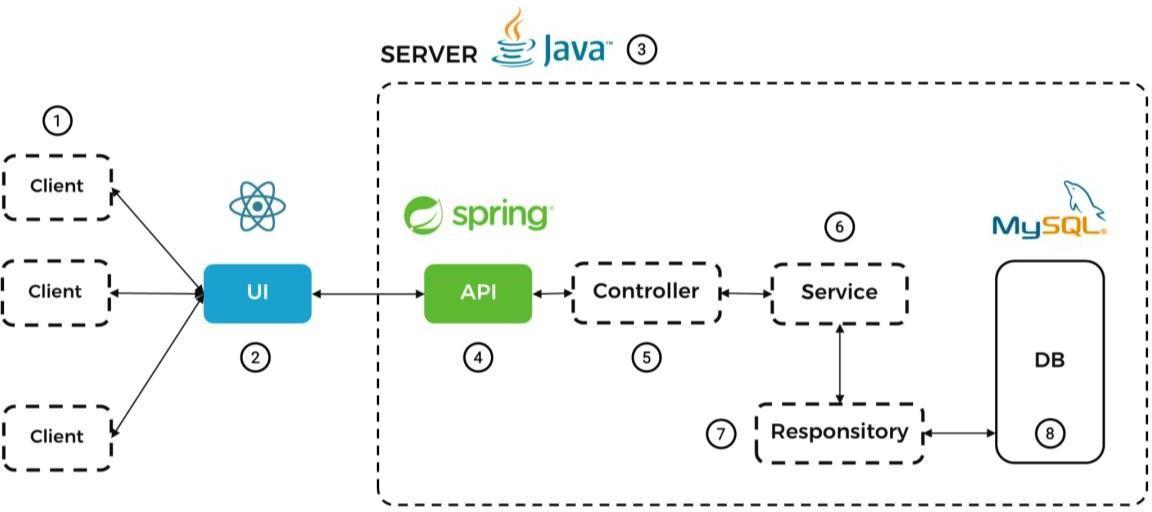
**Bảng 42. Diễn giải API sửa khuyến mãi**

*Xóa khuyến mãi: (Khi xóa, nếu sản phẩm nào có mã khuyến mãi này sẽ được gán null)*

| Phân quyền: | Chủ cửa hàng |
| --- | --- |
| Url: | http://localhost:8080/api/v1/sales/delete/{id} |
| Input:(Head) | Id khuyến mãi |
| Output: | Delete successfully! |
| Phương thức | DELETE |
| Authorization | Basic Authen(username, password) |

**Bảng 43. Diễn giải API xóa khuyến mãi**

# Kiến trúc hệ thống



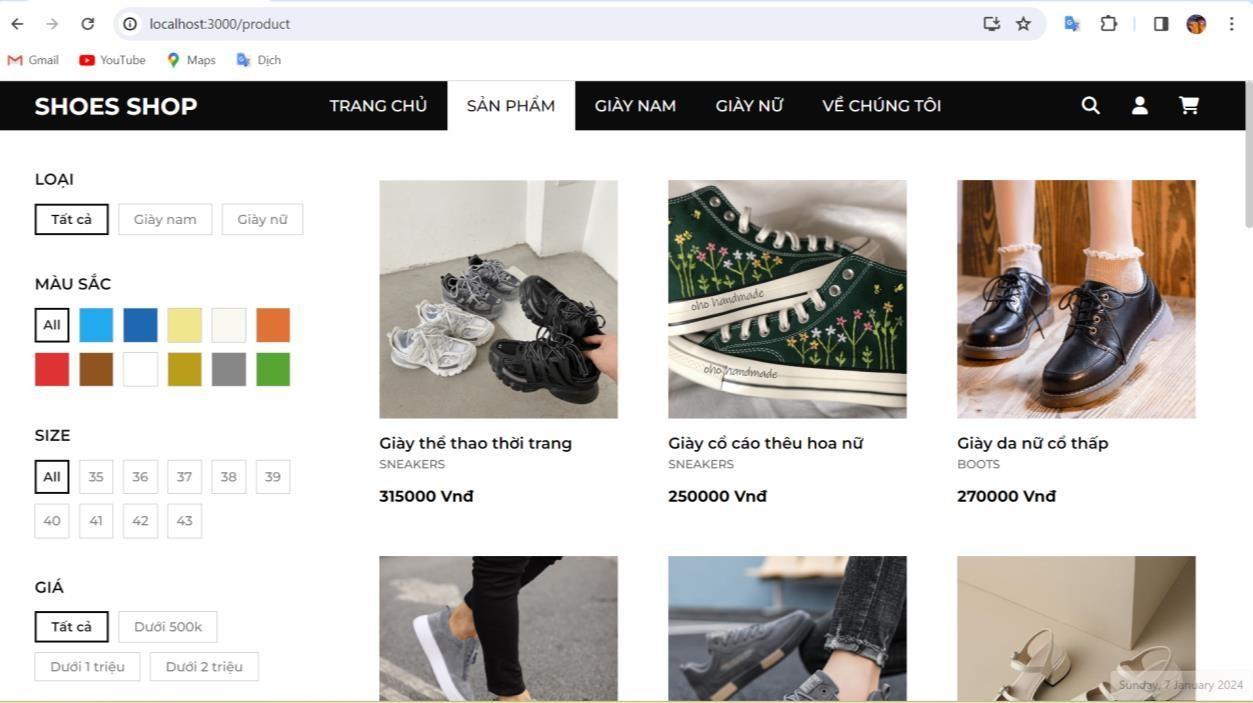
**Hình 6. Thiết kế kiến trúc hệ thống**

Thiết kế kiến trúc hệ thống bao gồm:

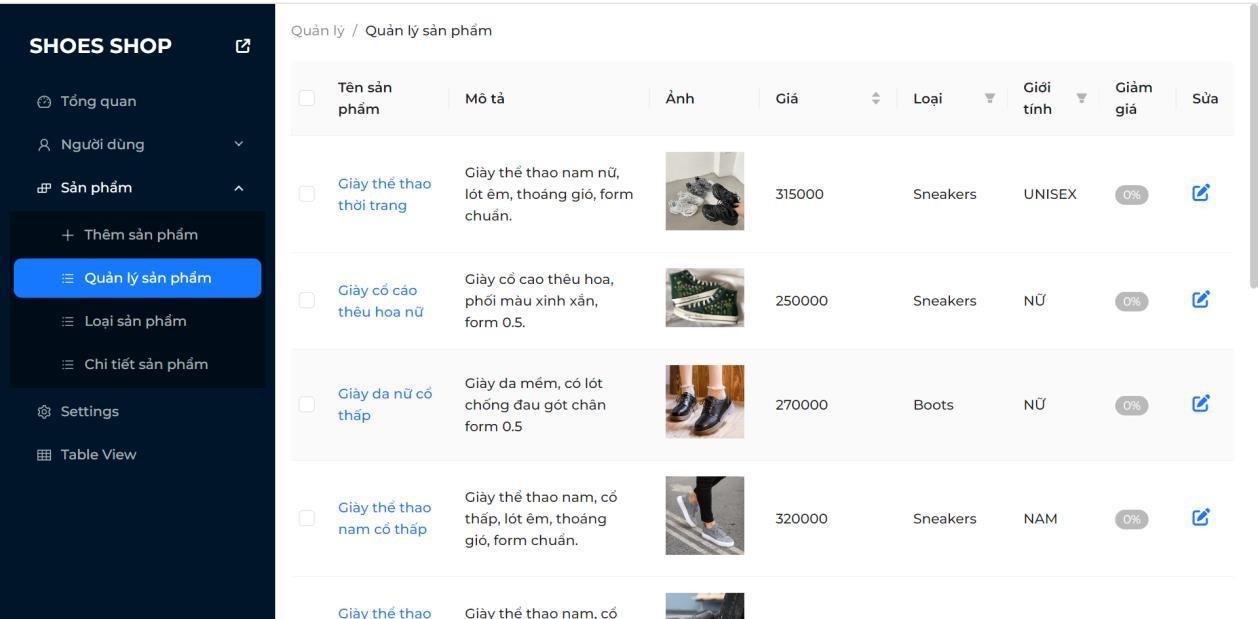
1. Client : máy khách
2. Giao diện tương tác người dùng: React
3. Server máy chủ Java: Tomcat
4. Gọi API qua URL từ Spring Boot
5. Controller nhận yêu cầu, gửi đến Service xử lý logic
6. Service xử lý, gọi repository lấy dữ liệu
7. Repository giao tiếp lấy dữ liệu và gửi dữ liệu về Database
8. Database MySql tiếp nhận câu lệnh và trả về dữ liệu

# CHƯƠNG 4: KẾT QUẢ

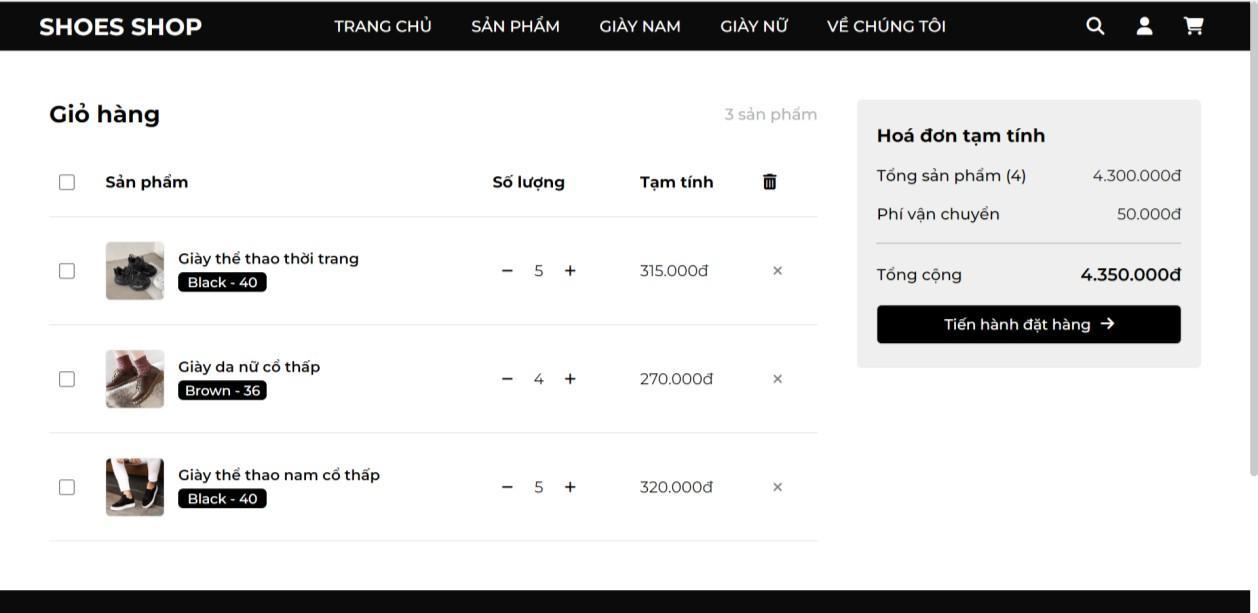
Một số giao diện các chức năng:



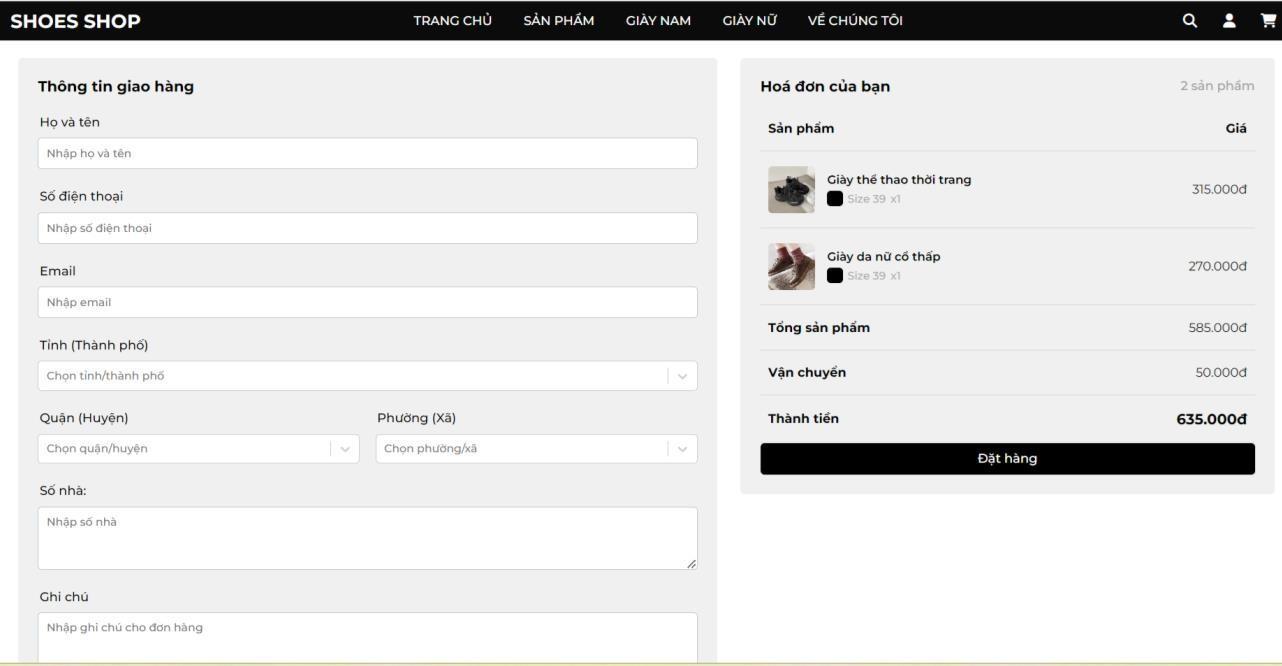
**Hình 7. Giao diện trang xem sản phẩm**



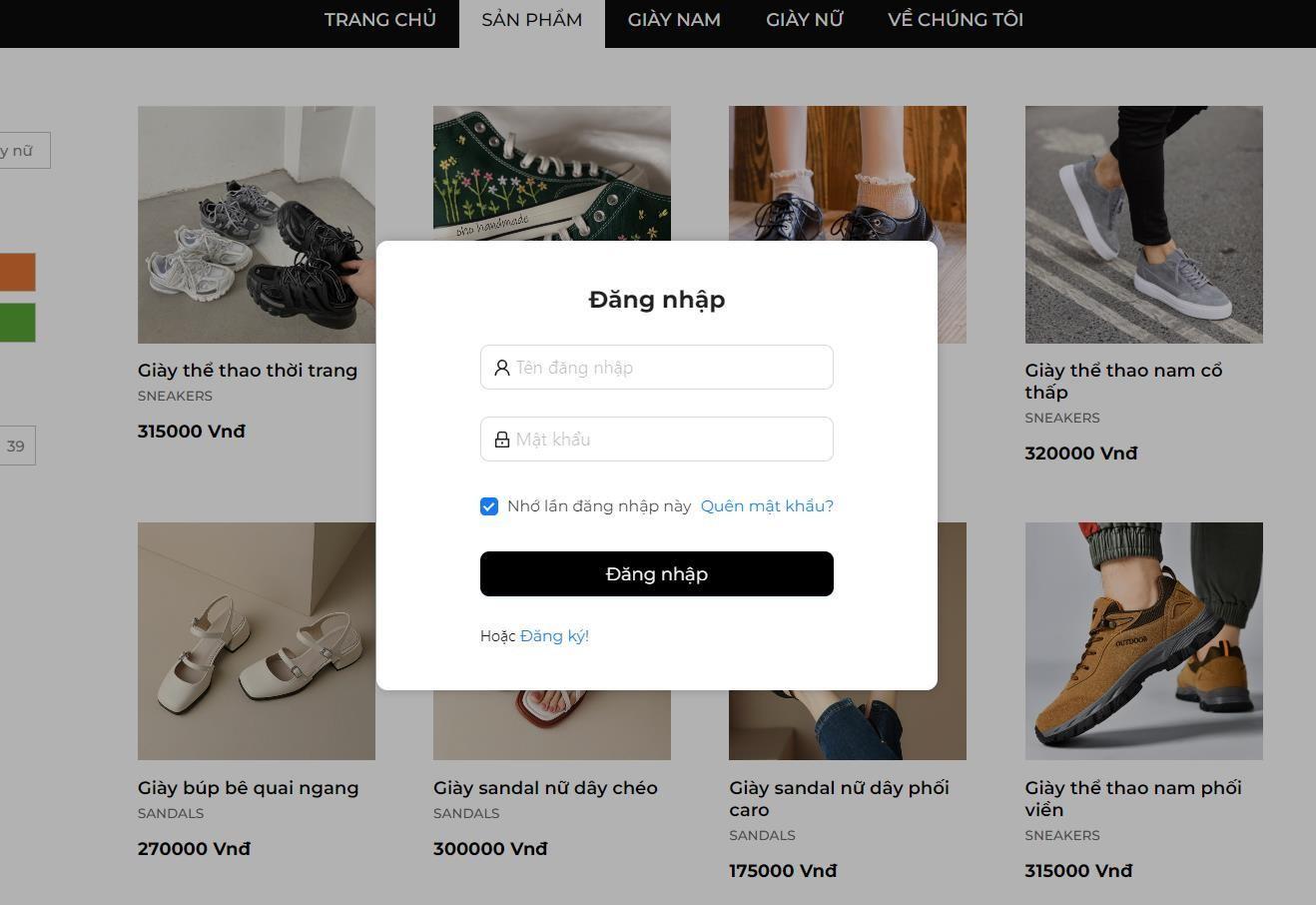
**Hình 8 Giao diện trang admin quản lý**



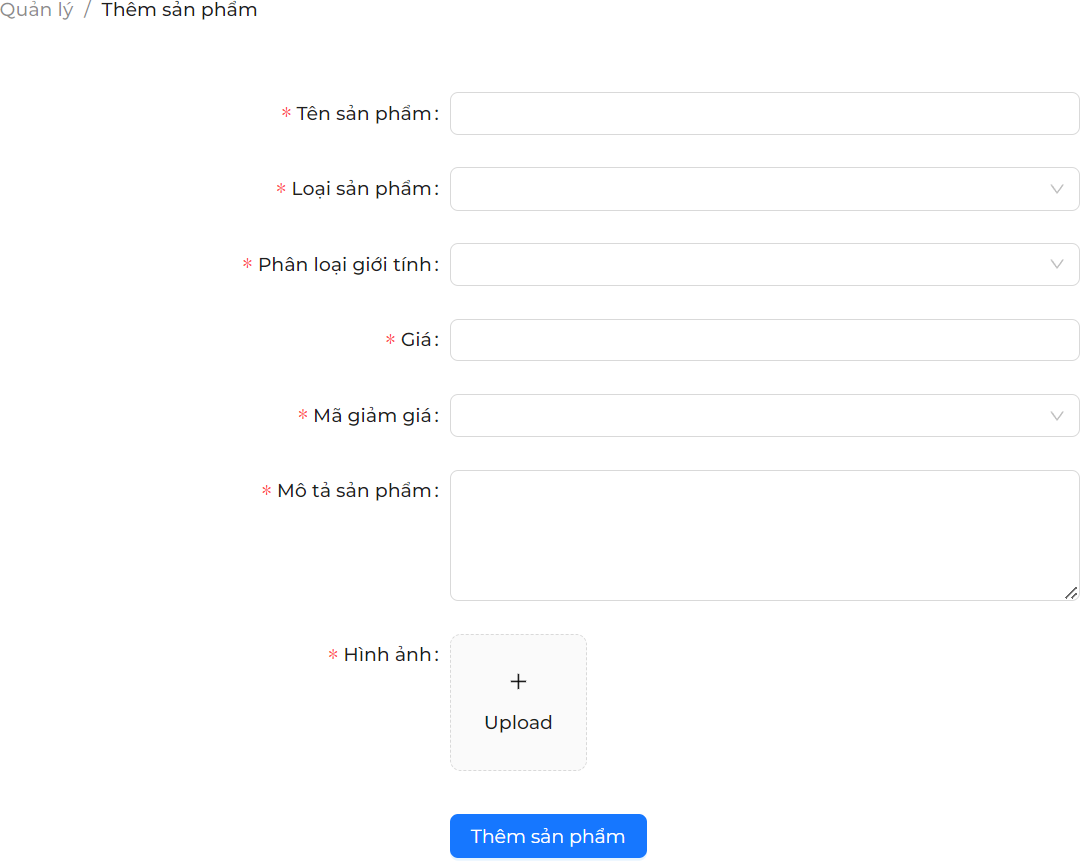
**Hình 9. Giao diện giỏ hàng**



**Hình 10. Giao diện đặt hàng**



**Hình 11.Giao diện đăng nhập**



**Hình 12.Giao diện thêm sản phẩm**

# CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

# Kết quả đạt được:

Nắm vững khái niệm, nguyên tắc, và cơ chế hoạt động của Web API. Áp dụng các công nghệ và tiêu chuẩn để phát triển hệ thống web.

# Hướng phát triển:

Cải thiện, phát triển về phân quyền và xác thực phân quyền bằng token.

Phát triển các tính năng cho web bán hàng minh họa như: quản lý đơn vị vận chuyển, đổi mật khẩu, quên mật khẩu, đánh giá bằng hình ảnh, khách hàng có thể nhắn tin cho nhân viên của shop,…