

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN**



BÁO CÁO CUỐI KỲ MÔN CÔNG NGHỆ JAVA

PHÁT TRIỂN WEBSITE QUẢN LÝ VÀ BÁN THUỐC PHARMACY

Người hướng dẫn: **ThS NGUYỄN THANH PHONG**

Người thực hiện: **LÝ TUẤN AN - 52000620**

NGUYỄN THÀNH AN – 52000621

PHAN THỊ DIỄM THÚY - 52000149

HUỲNH NGUYỄN TƯỜNG VY - 52000170

Lớp : 20050201

Khoá: 24

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2022

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN**



BÁO CÁO CUỐI KỲ MÔN CÔNG NGHỆ JAVA

PHÁT TRIỂN WEBSITE QUẢN LÝ VÀ BÁN THUỐC PHARMACY

Người hướng dẫn: **ThS NGUYỄN THANH PHONG**

Người thực hiện: **LÝ TUẤN AN - 52000620**

NGUYỄN THÀNH AN – 52000621

PHAN THỊ DIỄM THÚY - 52000149

HUỲNH NGUYỄN TƯỜNG VY - 52000170

Lớp : 20050201

Khoá: 24

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2022

LỜI CẢM ƠN

Trong suốt thời gian qua, nhờ sự giảng dạy tận tâm của quý Thầy Cô khoa Công Nghệ Thông Tin, trường Đại học Tôn Đức Thắng, em đã học hỏi được rất nhiều điều bổ ích và tích lũy cho mình một số kiến thức để hoàn thành bài báo cáo này. Em xin chân thành cảm ơn.

Em xin cảm ơn thầy Nguyễn Thanh Phong đã tận tình chỉ bảo chúng em qua những buổi học tại lớp, thầy đã chỉ em cách thức làm bài, chỉ điểm những chỗ còn sai sót chưa phù hợp cũng như phải làm sao để trình bày bố cục đẹp. Nếu không có những lời hướng dẫn, dạy bảo của thầy thì bài thu hoạch của em cũng rất khó để hoàn thiện. Một lần nữa em xin chân thành cảm ơn thầy.

Bước đầu đi vào thực tế với nền kiến thức mở rộng, kiến thức của em còn hạn chế và nhiều bất ngờ. Vì thế, trong quá trình biên soạn khó tránh những sai sót, em rất mong nhận được những ý kiến đóng góp quý báu của thầy và các bạn để bài báo cáo hoàn thiện hơn.

ĐỒ ÁN ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Tôi xin cam đoan đây là sản phẩm đồ án của riêng tôi và được sự hướng dẫn của ThS Nguyễn Thanh Phong. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình. Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

TP. Hồ Chí Minh, ngày tháng năm

Tác giả

(ký tên và ghi rõ họ tên)

An

Lý Tuấn An

An

Nguyễn Thành An

Thúy

Phan Thị Diễm Thúy

Vy

Huỳnh Nguyễn Tường Vy

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

Phần xác nhận của GV hướng dẫn

Tp. Hồ Chí Minh, ngày tháng năm
(kí và ghi họ tên)

Phần đánh giá của GV chấm bài

Tp. Hồ Chí Minh, ngày tháng năm
(kí và ghi họ tên)

MỤC LỤC

LỜI CẢM ƠN	1
PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN	3
MỤC LỤC.....	4
DANH MỤC BẢNG BIỂU	7
DANH MỤC HÌNH ẢNH.....	8
CHƯƠNG I: TỔNG QUAN	10
1.1 Mô tả nội dung đề tài website quản lý và bán thuốc	10
1.1.1 Phân quyền hệ thống và các chức năng chung	10
1.1.2 Các chức năng của khách hàng	10
1.1.3 Các chức năng của Admin.....	11
1.2 Cơ sở lý thuyết	12
1.2.1 Spring MVC	12
1.2.1.1 Spring MVC là gì?	12
1.2.1.2 Mô hình Spring MVC	12
1.2.1.3 Ứng dụng của Spring MVC trong Web	12
1.2.1.4 Flow trong Spring MVC	13
1.2.2 JPA Hibernate.....	14
1.2.2.1 JPA Hibernate là gì?	14
1.2.2.2 Một số Annotation phổ biến trong JPA	14
1.2.2.3 Kiến trúc JPA.....	15
1.2.2.4 Lợi ích	15
1.2.2.5 Các tính năng của JPA Hibernate	15
1.2.3 Spring Security	16
1.2.3.1 Spring Secutiry là gì?.....	16

1.2.3.2	Các tính năng của Spring Security	17
1.2.3.3	Flow trong Spring Security	18
1.2.4	Bcrypt Library	19
1.2.4.1	Bcrypt là gì?	19
1.2.4.2	Cách sử dụng Bcrypt	19
1.2.4.3	Lợi ích của Bcrypt	20
1.3	Đặc tả Use Case	21
1.3.1	Đăng ký	21
1.3.2	Đăng nhập	21
1.3.3	Quên mật khẩu	22
1.3.4	Đăng xuất	23
1.3.5	Tìm kiếm sản phẩm	23
1.3.6	Quản lý sản phẩm	24
1.3.7	Quản lý giỏ hàng	26
1.3.8	Quản lý đơn hàng	29
1.3.9	Quản lý user	31
1.3.10	Xem chi tiết đơn hàng	32
1.4	Thiết kế sơ đồ	33
1.4.1	Use Case	33
1.4.2	Sequence	34
1.4.2.1	Đăng kí	34
1.4.2.2	Đăng nhập	35
1.4.2.3	Đăng xuất	35
1.4.2.4	Quên mật khẩu	36
1.4.2.5	Sửa thông tin	37
1.4.2.6	Tìm kiếm sản phẩm	38
1.4.2.7	Thêm sản phẩm vào giỏ hàng	38
1.4.2.8	Đặt hàng	39
1.4.2.9	Quản lý User	40

1.4.2.10 Quản lý sản phẩm	40
1.4.2.11 Quản lý giỏ hàng.....	41
1.4.2.12 Quản lý đơn hàng.....	41
1.4.2.13 Sử dụng các chức năng	42
1.4.3 ERD	43
CHƯƠNG II: PHÁT TRIỂN WEBSITE VÀ DEMO	44
2.1 Cấu trúc project	44
2.2 Demo	45
2.2.1 Giao diện đăng ký	45
2.2.2 Giao diện đăng nhập	46
2.2.3 Giao diện trang chủ.....	46
2.2.4 Giao diện trang chi tiết sản phẩm	47
2.2.5 Giao diện trang giỏ hàng.....	47
2.2.6 Giao diện trang tạo hóa đơn.....	48
2.2.7 Giao diện trang tạo đơn hàng thành công	48
2.2.8 Giao diện trang lịch sử đơn hàng	49
2.2.9 Giao diện trang chi tiết lịch sử đơn hàng.....	49
2.2.10 Giao diện trang chủ admin.....	50
2.2.11 Giao diện trang quản lý danh mục sản phẩm	50
2.2.12 Giao diện trang quản lý tính năng của sản phẩm	51
2.2.13 Giao diện trang quản lý sản phẩm	51
2.2.14 Giao diện trang quản lý tài khoản.....	52
2.2.15 Giao diện trang quản lý đơn hàng.....	52
2.3 Các bước thực hiện để chạy project	53
CHƯƠNG III: KẾT LUẬN	54
3.1 Ưu điểm của trang web	54
3.2 Nhược điểm của trang web	54
3.3 Phân công công việc và đánh giá của thành viên	55
TÀI LIỆU THAM KHẢO.....	57

DANH MỤC BẢNG BIỂU

Bảng 1. 1 Đặc tả use case đăng ký	21
Bảng 1. 2 Đặc tả use case đăng nhập	22
Bảng 1. 3 Đặc tả use case quên mật khẩu	23
Bảng 1. 4 Đặc tả use case đăng xuất	23
Bảng 1. 5 Đặc tả use case tìm kiếm	24
Bảng 1. 6 Đặc tả use case thêm sản phẩm	24
Bảng 1. 7 Đặc tả use case cập nhật sản phẩm	25
Bảng 1. 8 Đặc tả use case xóa sản phẩm.....	26
Bảng 1. 9 Đặc tả use case thêm sản phẩm vào giỏ hàng.....	26
Bảng 1. 10 Đặc tả use case cập nhật giỏ hàng	27
Bảng 1. 11 Đặc tả use case xóa sản phẩm ra khỏi giỏ hàng	28
Bảng 1. 12 Đặc tả use case tạo đơn hàng	29
Bảng 1. 13 Đặc tả use case cập nhật thông tin đơn hàng	30
Bảng 1. 14 Đặc tả use case hủy đơn hàng.....	30
Bảng 1. 15 Đặc tả use case cập nhật thông tin user	31
Bảng 1. 16 Đặc tả use case xóa user	32
Bảng 1. 17 Đặc tả use case xem chi tiết đơn hàng.....	32

DANH MỤC HÌNH ẢNH

Hình 1. 1 Flow trong Spring MVC	13
Hình 1. 2 Kiến trúc JPA	15
Hình 1. 3 Flow trong Spring Security	18
Hình 1. 4 Use case tổng quát.....	33
Hình 1. 5 Sequence đăng kí.....	34
Hình 1. 6 Sequence đăng nhập.....	35
Hình 1. 7 Sequence đăng xuất.....	35
Hình 1. 8 Sequence quên mật khẩu.....	36
Hình 1. 9 Sequence sửa thông tin.....	37
Hình 1. 10 Sequence tìm kiếm sản phẩm.....	38
Hình 1. 11 Sequence thêm sản phẩm vào giỏ hàng	38
Hình 1. 12 Sequence đặt hàng.....	39
Hình 1. 13 Sequence quản lý user.....	40
Hình 1. 14 Sequence quản lý sản phẩm	40
Hình 1. 15 Sequence quản lý giỏ hàng.....	41
Hình 1. 16 Sequence quản lý đơn hàng.....	41
Hình 1. 17 Sequence sử dụng các chức năng CRUD.....	42
Hình 1. 18 Mô hình ERD	43
Hình 2. 1 Cấu trúc project.....	44
Hình 2. 2 Trang đăng ký	45
Hình 2. 3 Trang đăng nhập.....	46
Hình 2. 4 Trang chủ	46
Hình 2. 5 Trang chi tiết sản phẩm.....	47

Hình 2. 6 Trang giỏ hàng	47
Hình 2. 7 Trang tạo hóa đơn	48
Hình 2. 8 Trang tạo hóa đơn thành công.....	48
Hình 2. 9 Trang lịch sử đặt hàng.....	49
Hình 2. 10 Trang chi tiết lịch sử đơn hàng	49
Hình 2. 11 Trang chủ admin	50
Hình 2. 12 Trang quản lý danh mục sản phẩm	50
Hình 2. 13 Trang quản lý tính năng sản phẩm.....	51
Hình 2. 14 Trang quản lý sản phẩm	51
Hình 2. 15 Trang quản lý tài khoản	52
Hình 2. 16 Trang quản lý đơn hàng	52

CHƯƠNG I: TỔNG QUAN

1.1 Mô tả nội dung đề tài website quản lý và bán thuốc

1.1.1 Phân quyền hệ thống và các chức năng chung

- Website được chia ra làm 2 cấp phân quyền chính đó là khách hàng và admin.
- Với mỗi cấp phân quyền thì sẽ có quyền hạn khác nhau để thực hiện các tác vụ khác nhau trong hệ thống.

1.1.2 Các chức năng của khách hàng

- Tìm kiếm và lọc các sản phẩm
 - o Cho phép người dùng tìm kiếm các sản phẩm mong muốn theo theo danh mục, tính năng của thuốc, sắp xếp theo tên và giá tiền.
 - o Hệ thống sẽ hiển thị kết quả tìm kiếm phù hợp với yêu cầu của người dùng. Điều này sẽ giúp người dùng tìm kiếm sản phẩm một cách dễ dàng và nhanh chóng.
- Xem chi tiết sản phẩm: Người dùng có thể xem chi tiết về mô tả sản phẩm.
- Quản lý tài khoản
 - o Người dùng có thể quản lý tài khoản của mình bao gồm xem và cập nhật thông tin tài khoản.
 - o Điều kiện: Người dùng phải thực hiện đăng nhập thì mới có thể thực hiện các chức năng quản lý tài khoản.
- Quản lý giỏ hàng:
 - o Thêm: Người dùng có thể thêm sản phẩm mong muốn của mình vào giỏ hàng hiện tại của mình. Hệ thống sẽ cập nhật lại số lượng và tổng tiền ở trang xem giỏ hàng.
 - o Xóa: Người dùng có thể xóa các sản phẩm ra khỏi giỏ hàng hiện tại của mình. Hệ thống sẽ cập nhật lại số lượng và tổng tiền ở trang xem giỏ hàng.

- Xem: Người dùng có thể xem lại các sản phẩm trong giỏ hàng trước đi tiến hành thanh toán tại trang xem giỏ hàng.
- Điều kiện: Người dùng phải thực hiện đăng nhập thì mới có thể thực hiện các chức năng quản lý giỏ hàng.
- Đặt hàng (Thanh toán trực tiếp): Người dùng có thể thực hiện đặt các sản phẩm đang có trong giỏ hàng bằng cách điền đầy đủ các thông tin giao hàng. Đơn hàng sau đó sẽ được lưu vào lịch sử đặt hàng.
- Xem lịch sử đặt hàng: Các đơn hàng sẽ có các trạng thái theo các cột mốc khác nhau như: “Chờ duyệt”, “Đang giao hàng”, “Đã giao hàng”, ... tùy thuộc vào sự phê duyệt từ phía admin.

1.1.3 Các chức năng của Admin

- Quản lý danh mục sản phẩm: Admin có thể thêm, xóa, sửa danh mục sản phẩm.
- Quản lý danh mục tính năng: Admin có thể thêm, xóa, sửa danh mục tính năng.
- Quản lý sản phẩm: Admin có thể thêm, xóa, sửa sản phẩm.
- Quản lý đơn hàng: Admin có thể xem, sửa và phê duyệt các đơn đặt hàng từ khách hàng.
- Quản lý tài khoản khách hàng: Admin có thể xem, chỉnh sửa quyền truy cập giỏ hàng hoặc đặt hàng của khách hàng.

1.2 Cơ sở lý thuyết

1.2.1 Spring MVC

1.2.1.1 Spring MVC là gì?

- Spring MVC là một framework được sử dụng để phát triển các ứng dụng web Java. Nó được xây dựng dựa trên mô hình MVC (Model-View-Controller), cung cấp cho các nhà phát triển một cách tiếp cận tổ chức và phân tách các thành phần trong ứng dụng.

1.2.1.2 Mô hình Spring MVC

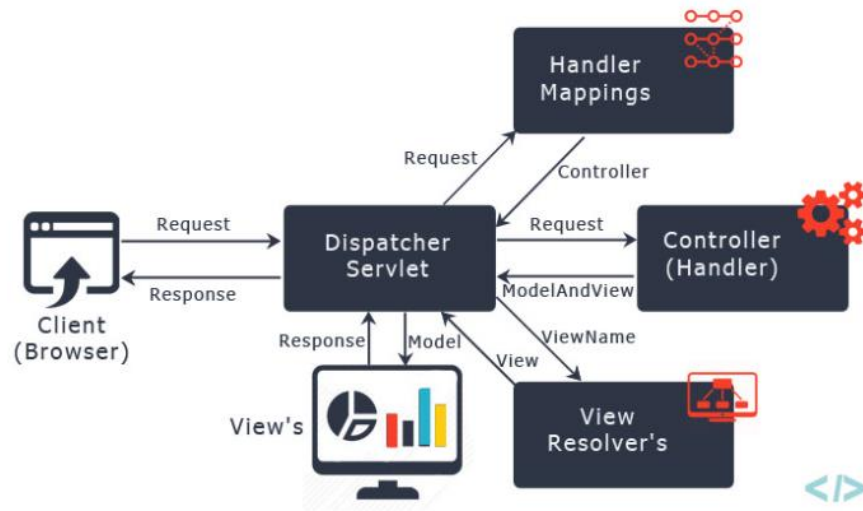
- Mô hình Spring MVC được tổ chức theo các thành phần sau:
 - *Model*: Là thành phần chịu trách nhiệm xử lý dữ liệu và tương tác với cơ sở dữ liệu. Model được sử dụng để truy xuất và thao tác dữ liệu, sau đó trả về kết quả cho Controller.
 - *View*: Là thành phần chịu trách nhiệm hiển thị dữ liệu cho người dùng cuối. View thường là các trang HTML được render bằng cách sử dụng các framework như Thymeleaf hoặc JSP.
 - *Controller*: Là thành phần chịu trách nhiệm điều hướng và quản lý luồng điều khiển trong ứng dụng. Controller xử lý các yêu cầu từ người dùng và gọi các phương thức trong Model để lấy dữ liệu, sau đó chuyển tiếp kết quả đó đến View để hiển thị.

1.2.1.3 Ứng dụng của Spring MVC trong Web

- Spring MVC cung cấp một số class và interface để hỗ trợ phát triển ứng dụng web. Các class và interface quan trọng nhất bao gồm:
 - *DispatcherServlet*: Là một Servlet chính được sử dụng để điều khiển và quản lý các yêu cầu tới các Controller trong ứng dụng. Nó là thành phần trung tâm của Spring MVC.
 - *HandlerMapping*: Là một interface định nghĩa cách mà các yêu cầu sẽ được ánh xạ đến các Controller tương ứng.

- *ModelAndView*: Là một class chứa dữ liệu và thông tin để hiển thị kết quả cho View.
- *ViewResolver*: Là một interface định nghĩa cách mà các ModelAndView sẽ được render ra các trang HTML.

1.2.1.4 Flow trong Spring MVC



Hình 1. 1 Flow trong Spring MVC

- Người dùng điền vào tên website mình cần truy cập trên browser sau đó bấm enter. Lúc này mình gửi 1 request lên server nơi mà mình triển khai ứng dụng Spring.
- Thành phần DispatcherServlet của Spring MVC sẽ nhận được request (yêu cầu) của người dùng ở bước 1. Dispatcher là thành phần quan trọng nhất trong springmvc. Nó sẽ là nơi đầu tiên nhận request từ client sau đó sẽ chuyển request đó tới các controller tương ứng, đồng thời sẽ là chốt chặn cuối cùng trả về kết quả cho client.
- Sau khi nhận được request (yêu cầu) DispatcherServlet sẽ chuyển yêu cầu đó tới Controller bằng các cơ chế mapping mà ta khai báo trong Handler Mapping.

1.2.2 JPA Hibernate

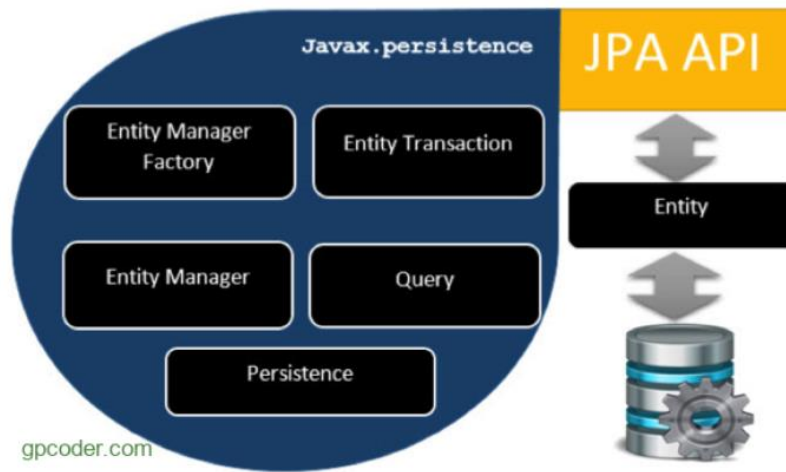
1.2.2.1 JPA Hibernate là gì?

- *JPA (Java Persistence API)* là một tiêu chuẩn Java để lập trình ORM (Object-Relational Mapping) cho các ứng dụng Java EE và Java SE. JPA được sử dụng để ánh xạ các đối tượng Java vào cơ sở dữ liệu quan hệ, giúp giảm thiểu việc viết mã SQL thủ công và đơn giản hóa việc tương tác với cơ sở dữ liệu.
- *Hibernate* là một framework ORM phổ biến cho Java, được sử dụng để ánh xạ các đối tượng Java vào cơ sở dữ liệu quan hệ. Hibernate cung cấp một số tính năng như quản lý phiên, định tuyến và giao dịch, và là một trong những cách phổ biến nhất để triển khai JPA.
- Các đối tượng Java được ánh xạ vào các bảng trong cơ sở dữ liệu thông qua các Annotation được định nghĩa trong JPA. Những Annotation này được sử dụng để xác định các quan hệ giữa các đối tượng, các thuộc tính và cột tương ứng trong cơ sở dữ liệu.

1.2.2.2 Một số Annotation phổ biến trong JPA

- `@Entity`: Xác định rằng một class là một đối tượng được ánh xạ vào cơ sở dữ liệu.
- `@Id`: Xác định rằng một thuộc tính trong đối tượng là khóa chính của bảng tương ứng trong cơ sở dữ liệu.
- `@GeneratedValue`: Xác định rằng giá trị của khóa chính được tự động tạo ra bởi cơ sở dữ liệu.
- `@Column`: Xác định tên của cột trong bảng tương ứng với một thuộc tính trong đối tượng.

1.2.2.3 Kiến trúc JPA



Hình 1. 2 Kiến trúc JPA

- JPA bao gồm ba thành phần chính:
 - o Entity
 - o Entity Manager
 - o Entity ManagerFactory
- Ngoài ra còn có: EntityTransaction, Persistence, Query.

1.2.2.4 Lợi ích

- Đơn giản hóa công nghệ cho tầng persistence (tầng dữ liệu).
- Không phụ thuộc vào các framework ORM.
- Có nhiều nhà cung cấp hỗ trợ cài đặt JPA.
- Dữ liệu có thể được lưu trữ thông qua việc ORM.

1.2.2.5 Các tính năng của JPA Hibernate

Hibernate là một framework ORM (Object-Relational Mapping) cho phép lập trình viên ánh xạ các đối tượng Java vào cơ sở dữ liệu quan hệ. Hibernate hỗ trợ triển khai tiêu chuẩn JPA (Java Persistence API), cung cấp các tính năng sau:

- Mapping đối tượng và quan hệ dữ liệu: Hibernate cho phép lập trình viên ánh xạ đối tượng Java vào các bảng trong cơ sở dữ liệu quan hệ. Các quan hệ giữa các đối tượng và các thuộc tính của chúng có thể được định nghĩa bằng các Annotation hoặc XML mapping file.
- Hỗ trợ JPA API: Hibernate cung cấp các API và interface cho phép lập trình viên sử dụng các tính năng của JPA, bao gồm EntityManager, Query và Transaction.
- Caching: Hibernate hỗ trợ các cơ chế cache để tăng tốc độ truy cập cơ sở dữ liệu, giảm thiểu số lượng truy vấn và tối ưu hiệu suất.
- Fetching strategies: Hibernate cho phép lập trình viên định nghĩa cách tải các đối tượng từ cơ sở dữ liệu, bao gồm fetching strategies cho việc tải dữ liệu theo chiều sâu hoặc theo chiều rộng.
- Quản lý phiên: hibernate hỗ trợ quản lý phiên để đảm bảo tính toàn vẹn của dữ liệu và giảm thiểu số lượng truy vấn cơ sở dữ liệu.
- Hỗ trợ tương thích với các cơ sở dữ liệu quan hệ khác nhau: Hibernate hỗ trợ tương thích với nhiều cơ sở dữ liệu quan hệ khác nhau bao gồm MySQL, Oracle, MicrosoftSQL, Server và PostgreSQL.
- Tính năng truy vấn động: Hibernate cho phép lập trình viên tạo truy vấn động để truy vấn dữ liệu từ cơ sở dữ liệu, giúp giảm thiểu việc viết mã SQL thủ công.

1.2.3 Spring Security

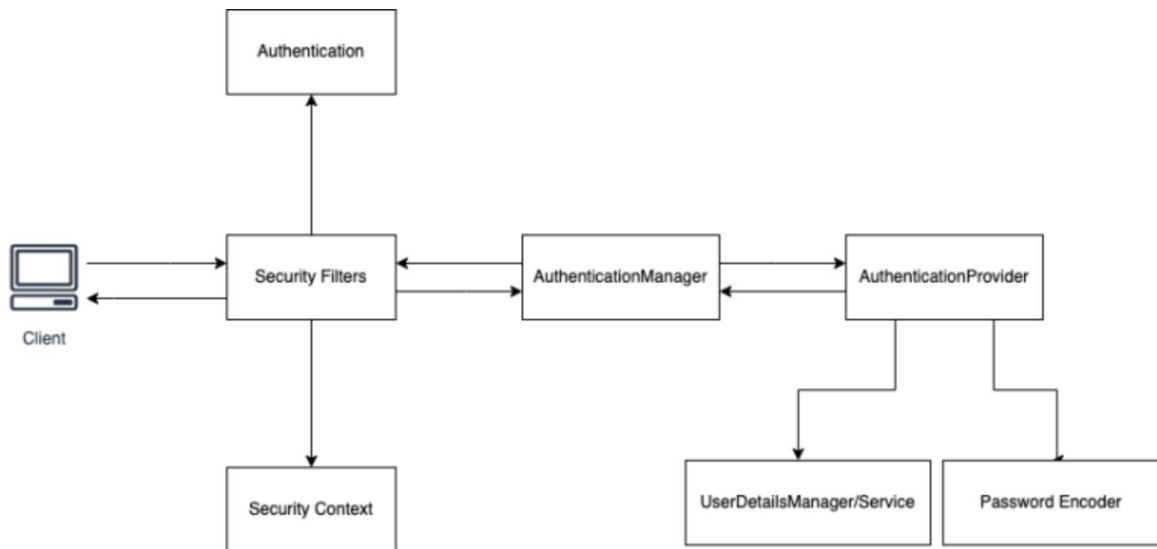
1.2.3.1 Spring Security là gì?

- Spring Security là một framework giúp ta phân quyền và xác thực người dùng trước khi cho phép họ truy cập vào các tài nguyên của chúng ta.
- Spring Security cung cấp các tính năng như xác thực, ủy quyền và bảo vệ chống lại các cuộc tấn công thông thường.

1.2.3.2 Các tính năng của Spring Security

- **Authentication:** Là một cách mà ta xác minh danh tính của người dùng đang cố truy cập vào một tài nguyên cụ thể. Một cách phổ biến để xác thực người dùng là yêu cầu họ nhập username và password. Sau khi xác thực được thực hiện thì ta có thể thực hiện việc ủy quyền.
- **Authorization:** Spring Security cho phép phân quyền truy cập với các tài nguyên của ứng dụng. Nó có thể được phân quyền để cấu hình dựa trên vai trò (Role-based) hoặc dựa trên các điều kiện khác.
- **Session Management:** Spring Security hỗ trợ quản lý phiên và ngăn chặn tấn công CSRF (Cross-Site Request Forgery).
- **Remember Me:** Giúp người dùng đăng nhập một lần và duy trì phiên đăng nhập trên nhiều lần truy cập sau.
- **Security Headers:** Cơ chế cấu hình các header bảo mật cho ứng dụng web như Content Security Policy (CSP), HTTP Strict Transport Security (HSTS) và X-XSS Protection.
- **LDAP và Active Directory:** Spring Security hỗ trợ tích hợp với LDAP và Active Directory để xác thực và phân quyền truy cập.
- **Auditing:** Khả năng ghi lại các hoạt động của người dùng và các sự kiện bảo mật quan trọng trong ứng dụng.
- **Multi-factor Authentication:** Cho phép người dùng xác thực bằng nhiều cách thức khác nhau như mật khẩu, mã OTP, thẻ thông minh, ...
- **Tích hợp:** Spring Security có thể được tích hợp với các framework khác nhau như Spring Boot, Spring Web MVC và các framework khác để cung cấp tính năng bảo mật.

1.2.3.3 Flow trong Spring Security



Hình 1. 3 Flow trong Spring Security

- **Security Filters:** Là những filters sẽ nằm giữa client request với server. Khi nhận được request, các filter sẽ tách lọc những thông tin từ request thành các authentication details (username, password, roles, ...). Default Spring Security sẽ sử dụng class UsernamePasswordAuthenticationFilter.
- **Authentication:** là một base object làm nhiệm vụ validate user credentials nhận được từ phía client. Ở behavior mặc định, Authentication object sẽ là class UsernamePasswordAuthenticationToken.
- **AuthenticationManager:** là một interface với method authenticate() làm nhiệm vụ xác định những Authentication providers phù hợp nhất để xử lý Authentication object nhận được từ filters. AuthenticationManager sẽ nhận kết quả authenticate từ Provider (Success hoặc Not success). Nếu không success, nó sẽ thử một provider phù hợp khác.
- **AuthenticationProvider:** Là những classes implement interface Authentication Provider với method authenticate() làm nhiệm vụ xử lý các logic liên quan đến authentication.

- **UserDetailsService:** Là interface chứa thông tin, schema của user details. Ở behavior mặc định, Spring Security sẽ sử dụng InMemoryUserDetailsManager, với method loadUserByUsername() để lấy ra thông tin của user trong hệ thống.
- **PasswordEncoder:** là interface có nhiệm vụ encode, encrypt và decrypt password của user, validate và trả về kết quả valid/invalid cho Authentication Provider xử lý.
- **Spring Context:** Sau khi Spring Security đã validate đủ, user details sẽ được lưu vào Security context. Ở lần truy cập tới, thông tin user sẽ được filter retrieve ở đây thay vì thực hiện đầy đủ các bước flow như ở trên.

1.2.4 Bcrypt Library

1.2.4.1 Bcrypt là gì?

- Bcrypt là một thuật toán mã hóa mật khẩu dựa trên chức năng băm (hash function) Blowfish. Nó được sử dụng phổ biến trong các ứng dụng web để lưu trữ mật khẩu của người dùng một cách an toàn và không thể giải mã.
- Sử dụng Bcrypt giúp đảm bảo tính bảo mật cho mật khẩu của người dùng và giúp tránh được các cuộc tấn công tìm kiếm ngược.
- Để sử dụng Bcrypt trong các ứng dụng Java, có thể sử dụng các thư viện như Spring Security hoặc JBCrypt, đều hỗ trợ việc sử dụng Bcrypt để mã hóa mật khẩu của người dùng.

1.2.4.2 Cách sử dụng Bcrypt

- Khi người dùng đăng ký hoặc thay đổi mật khẩu, mật khẩu của họ sẽ được mã hóa bằng Bcrypt và lưu trữ vào database.
- Các bước thực hiện của thuật toán Bcrypt như sau:

- Bcrypt tạo ngẫu nhiên một chuỗi salt, độ dài salt được xác định bởi tham số cost factor. Salt được thêm vào mật khẩu của người dùng để tạo thành một chuỗi mới, gọi là hash.
- Hash được mã hóa thành Bcrypt nhiều lần (với $2^{\text{cost factor}}$ lần), với mỗi lần mã hóa sử dụng hash của lần trước đó như là input. Việc mã hóa nhiều lần giúp tăng độ khó của việc tìm kiếm ngược và chống lại các cuộc tấn công khác.
- Hash được lưu trữ trong database cùng với salt.
- Khi người dùng đăng nhập, mật khẩu được nhập vào và mã hóa lại bằng cùng một chuỗi salt như lần đăng ký hoặc thay đổi mật khẩu.
- Bcrypt so sánh hash được lưu trữ trong database với hash mới được tạo ra từ mật khẩu nhập vào. Nếu hai hash này giống nhau, người dùng sẽ được đăng nhập thành công.

1.2.4.3 Lợi ích của Bcrypt

- Bảo mật mật khẩu: Mã hóa mật khẩu với độ bảo mật cao, giúp bảo vệ mật khẩu của người dùng trước các cuộc tấn công tìm kiếm ngược và các hình thức tấn công khác.
- Độ tin cậy cao: Bcrypt được sử dụng rộng rãi trong cộng đồng an ninh mạng và được coi là một trong những thuật toán mã hóa mật khẩu mạnh mẽ và tin cậy nhất.
- Khả năng điều chỉnh độ khó: Bằng cách sử dụng tham số cost factor, giúp tăng độ bảo mật của mật khẩu và chống lại các cuộc tấn công brute force.
- Dễ sử dụng: Bcrypt có thể được sử dụng trong các ứng dụng web và dự án mã nguồn mở bằng các sử dụng các thư viện hỗ trợ như bcrypt hay Spring Security.

1.3 Đặc tả Use Case

1.3.1 Đăng ký

USECASE NAME	Đăng ký	
TRIGGERING EVENT	Người dùng cần tạo một tài khoản mới.	
BRIEF DESCRIPTION	Người dùng đăng ký vào hệ thống khi muốn có tài khoản mới.	
ACTORS	User, Admin	
PRECONDITION	Tên đăng nhập không bị trùng lặp với các tài khoản đã đăng ký thành công trong hệ thống.	
POSTCONDITION	Quay về trang chủ của hệ thống, yêu cầu đăng nhập.	
FLOW OF EVENTS	Actor	System
	1. Người dùng chọn nút “Đăng ký”. 2. Người dùng nhập tên người dùng, số điện thoại/email, mật khẩu. 3. Người dùng nhấn nút “OK”.	1.1 Hệ thống hiển thị form đăng ký. 3.1 Hệ thống gửi yêu cầu xác thực đến cơ sở dữ liệu tài khoản. 3.2 Cơ sở dữ liệu kiểm tra thông tin đăng ký. 3.3 Hệ thống lưu thông tin người dùng và hiển thị đăng ký thành công, chuyển đến giao diện đăng nhập.
EXCEPTION CONDITION	3.2.1 Nếu tên người dùng, sđt/email đã tồn tại trong cơ sở dữ liệu thì thông báo tài khoản đã tồn tại → Đăng ký thất bại. 3.2.2 Nếu chưa tồn tại → Đăng ký thành công.	

Bảng 1. 1 Đặc tả use case đăng ký

1.3.2 Đăng nhập

USECASE NAME	Đăng nhập	
TRIGGERING EVENT	Người dùng cần đăng nhập vào ứng dụng.	
BRIEF DESCRIPTION	Người dùng đăng nhập vào hệ thống khi muốn sử dụng.	
ACTORS	User, Admin	
STAKEHOLDER	User, Admin	
PRECONDITION	Hệ thống chưa được đăng nhập	
POSTCONDITION	Cho phép đăng nhập vào hệ thống nếu đúng thông tin tài khoản	
FLOW OF EVENTS	Actor	System
	1. Chọn ứng dụng hiển thị trên desktop.	1.1 Hiển thị form đăng nhập. 2.1 Hệ thống gửi yêu cầu xác thực đến cơ sở dữ liệu tài khoản.

	2. Người dùng nhập tên đăng nhập, mật khẩu và nhấn nút “Đăng nhập”.	2.2 CSDL kiểm tra thông tin đăng nhập. 2.3 Hệ thống trả kết quả đăng nhập thành công và chuyển đến màn hình chính của ứng dụng.
EXCEPTION CONDITION	2.2.1 Nếu tồn tại cấp quyền tương ứng. 2.2.2 Nếu không trả về thông báo đăng nhập thất bại.	

Bảng 1. 2 Đặc tả use case đăng nhập

1.3.3 Quên mật khẩu

USECASE NAME	Quên mật khẩu	
TRIGGERING EVENT	Người quên mật khẩu cần đổi sang mật khẩu mới	
BRIEF DESCRIPTION	Người dùng xác nhận lại email khi muốn lấy lại mật khẩu	
ACTORS	User, Admin	
STAKEHOLDER	User, Admin	
PRECONDITION	Xác thực đúng thông qua email	
POSTCONDITION	Quay về trang chủ của hệ thống và đăng nhập với mật khẩu mới.	
FLOW OF EVENTS	Actor	System
	1. Người dùng chọn nút “Quên mật khẩu”. 2. Người dùng nhập email. 3. Người dùng nhập mã xác thực được gửi tới email. 4. Người dùng nhập mật khẩu mới và nhắc lại mật khẩu mới.	1.1 Hệ thống hiển thị form điền email hoặc số điện thoại. 2.1 Hệ thống kiểm tra thông tin email. 3.1 Hệ thống kiểm tra mã xác thực mà người dùng nhập. 4.1 Hệ thống cập nhật lại mật khẩu mới. Và quay về trang đăng nhập, yêu cầu đăng nhập lại.
EXCEPTION CONDITION	2.2.1 Nếu email có trong cơ sở dữ liệu thì cho phép người dùng nhập mã xác thực được gửi tới email. 2.2.2 Nếu không thì hệ thống gửi thông báo là email không hợp lệ. 3.1.1 Nếu mã xác thực đúng thì cho phép người dùng nhập mật khẩu mới	

	<p>3.1.2 Nếu mã xác thực sai thì thông báo xác thực thất bại.</p> <p>4.1.1 Nếu mật khẩu chưa tồn tại thì hệ thống lưu lại mật khẩu mới và hiển thị thành công, chuyển đến giao diện đăng nhập.</p> <p>4.1.2 Nếu mật khẩu đã tồn tại, hệ thống thông báo cho người dùng, yêu cầu nhập mật khẩu khác.</p>
--	---

Bảng 1. 3 Đặc tả use case quên mật khẩu

1.3.4 Đăng xuất

USECASE NAME	Đăng xuất	
TRIGGERING EVENT	Người dùng muốn đăng xuất khỏi ứng dụng.	
BRIEF DESCRIPTION	Người dùng đăng xuất khỏi hệ thống khi không muốn sử dụng nữa	
ACTORS	User, Admin	
STAKEHOLDER	User, Admin	
PRECONDITION	Đăng nhập thành công.	
POSTCONDITION	Quay về trang chủ của hệ thống khi chưa đăng nhập.	
FLOW OF EVENTS	Actor	System
	1. Người dùng chọn chức năng “Đăng xuất”. 2. Người dùng chọn YES/NO	2. Hiển thị thông báo xác nhận đăng xuất. 2.1 Đăng xuất khỏi ứng dụng và hiển thị trang đăng nhập
EXCEPTION CONDITION	2.1.1 Nếu chọn YES, đăng xuất tài khoản khỏi ứng dụng 2.1.2 Nếu chọn NO, giữ nguyên phiên bản đăng nhập.	

Bảng 1. 4 Đặc tả use case đăng xuất

1.3.5 Tìm kiếm sản phẩm

USECASE NAME	Tìm kiếm sản phẩm	
TRIGGERING EVENT	Người dùng cần tìm một sản phẩm nào đó.	
BRIEF DESCRIPTION	Người dùng tìm kiếm 1 sản phẩm để thêm vào đơn hoặc kiểm tra	
ACTORS	User, Admin	
STAKEHOLDER	User, Admin	
PRECONDITION	Không có	
POSTCONDITION	Hiển thị tất cả các sản phẩm tìm được dựa trên từ khóa mà người dùng nhập trên thanh tìm kiếm	
FLOW OF EVENTS	Actor	System

	1. Người dùng nhập từ khóa cần tìm và nhấn nút “Tìm kiếm”	1.1 Hiện thị các sản phẩm có liên quan đến từ khóa mà người dùng nhập
EXCEPTION CONDITION	1.1.1 Nếu từ người dùng nhập vào không có hoặc không hợp lệ thì hiển thị trống	

Bảng 1. 5 Đặc tả use case tìm kiếm

1.3.6 Quản lý sản phẩm

- Thêm sản phẩm

USECASE NAME	Thêm sản phẩm	
TRIGGERING EVENT	Người dùng muốn thêm 1 sản phẩm mới vào cửa hàng	
BRIEF DESCRIPTION	Khi người dùng muốn thêm 1 sản phẩm mới vào danh sách các sản phẩm	
ACTORS	Admin	
STAKEHOLDER	Admin	
PRECONDITION	Người dùng phải đăng nhập thành công vào ứng dụng	
POSTCONDITION	Nếu đăng nhập thành công thì có thể thêm sản phẩm mới vào cửa hàng	
FLOW OF EVENTS	Actor	System
	1. Người dùng bấm chọn tab sản phẩm	1.1 Hệ thống hiển thị các loại sản phẩm hiện có và được bày bán.
	2. Admin bấm chọn nút “+” để thêm sản phẩm.	2.1 Hiện thị form nhập thông tin của sản phẩm như mã, tên, đơn vị
	3. Nhấn chọn nút “OK” để tiến hành thêm sản phẩm	3.1 Hiện thị thông báo thêm sản phẩm thành công
EXCEPTION CONDITION	3.1.1 Nếu nhập thiếu thông tin thì yêu cầu nhập đầy đủ.	

Bảng 1. 6 Đặc tả use case thêm sản phẩm

- Cập nhật thông tin sản phẩm

USECASE NAME	Cập nhật thông tin sản phẩm
TRIGGERING EVENT	Người dùng muốn thay đổi thông tin của sản phẩm
BRIEF DESCRIPTION	Khi người dùng muốn thay đổi thông tin 1 sản phẩm nào đó có trong danh sách các sản phẩm được bày bán
ACTORS	Admin

STAKEHOLDER	Admin	
PRECONDITION	Người dùng phải đăng nhập thành công vào ứng dụng và sản phẩm phải có trong danh sách được bày bán	
POSTCONDITION	Nếu đăng nhập thành công thì có thể cập nhật thông tin sản phẩm có trong danh sách bày bán.	
FLOW OF EVENTS	Actor	System
	1. Admin bấm chọn tab sản phẩm 2. Admin bấm chọn “Sửa” để thay đổi thông tin sản phẩm. 3. Cập nhật thông tin cần thay đổi và nhấn nút “OK”	1.1 Hệ thống hiển thị các loại sản phẩm hiện có và được bày bán. 2.1 Hiện thị form thông tin của sản phẩm như mã, tên, đơn vị 3.1 Hiện thị thông báo cập nhật thành công
EXCEPTION CONDITION	Không có.	

Bảng 1. 7 Đặc tả use case cập nhật sản phẩm

- Xóa sản phẩm

USECASE NAME	Xóa sản phẩm	
TRIGGERING EVENT	Người dùng muốn xóa 1 sản phẩm trong cửa hàng	
BRIEF DESCRIPTION	Khi người dùng muốn xóa 1 sản phẩm ra khỏi danh sách các sản phẩm	
ACTORS	Admin	
STAKEHOLDER	Admin	
PRECONDITION	Người dùng phải đăng nhập thành công vào ứng dụng	
POSTCONDITION	Nếu đăng nhập thành công thì có thể xóa sản phẩm cần xóa và sản phẩm phải có trong danh sách được bày bán	
FLOW OF EVENTS	Actor	System
	1. Admin bấm chọn tab sản phẩm 2. Admin bấm chọn nút “xóa” để xóa sản phẩm. 3. Nhấn chọn nút “OK” để tiến hành xóa sản phẩm	1.1 Hệ thống hiển thị các loại sản phẩm hiện có và được bày bán. 2.1 Hiện thị thông báo bạn có chắc muốn xóa sản phẩm này. 3.1 Hiện thị thông báo xóa sản phẩm thành công

EXCEPTION CONDITION	3.1.1 Nếu chọn OK thì sản phẩm sẽ bị xóa, nếu chọn Cancel thì hủy thao tác xóa sản phẩm.
----------------------------	--

Bảng 1. 8 Đặc tả use case xóa sản phẩm

1.3.7 Quản lý giỏ hàng

- Thêm sản phẩm vào giỏ hàng

USECASE NAME	Thêm sản phẩm vào giỏ hàng	
TRIGGERING EVENT	Nhấn thêm sản phẩm	
BRIEF DESCRIPTION	Người dùng có thể thêm sản phẩm vào giỏ hàng của mình trong khi đang mua sắm trên trang web của cửa hàng.	
ACTORS	User, Admin	
STAKEHOLDER	User, Admin	
PRECONDITION	Nhấn thêm sản phẩm vào giỏ hàng	
POSTCONDITION	Quay về trang chủ của hệ thống.	
FLOW OF EVENTS	Actor	System
	1. Người dùng truy cập trang web của cửa hàng và chọn sản phẩm mà họ muốn mua. 2. Khi người dùng chọn sản phẩm, họ có thể nhấp vào nút "Thêm vào giỏ hàng" để thêm sản phẩm vào giỏ hàng của mình.	2.1 Hệ thống kiểm tra xem sản phẩm đã được thêm vào giỏ hàng chưa. Nếu sản phẩm đã có trong giỏ hàng, hệ thống cập nhật số lượng sản phẩm. 2.2 Nếu sản phẩm chưa có trong giỏ hàng, hệ thống thêm sản phẩm vào giỏ hàng với số lượng là 1. 2.3 Hệ thống hiển thị thông báo cho người dùng về việc thêm sản phẩm vào giỏ hàng thành công.
EXCEPTION CONDITION	Người dùng không thể thêm sản phẩm vào giỏ hàng nếu sản phẩm đã hết hàng hoặc không có sẵn. Nếu người dùng nhập số lượng sản phẩm lớn hơn số lượng sản phẩm có sẵn trong kho, họ sẽ nhận được thông báo lỗi. Nếu người dùng chưa đăng nhập, họ sẽ được yêu cầu đăng nhập hoặc đăng ký trước khi thêm sản phẩm vào giỏ hàng.	

Bảng 1. 9 Đặc tả use case thêm sản phẩm vào giỏ hàng

- Cập nhật thông tin giỏ hàng

USECASE NAME	Sửa sản phẩm trong giỏ hàng	
TRIGGERING EVENT	Nhấn sửa sản phẩm	
BRIEF DESCRIPTION	Người dùng có thể sửa số lượng sản phẩm trong giỏ hàng của mình trong khi đang mua sắm trên trang web của cửa hàng.	
ACTORS	User, Admin	
STAKEHOLDER	User, Admin	
PRECONDITION	Nhấn sửa sản phẩm trong giỏ hàng	
POSTCONDITION	Quay về trang chủ của hệ thống.	
FLOW OF EVENTS	Actor	System
	1. Người dùng truy cập trang web của cửa hàng và xem giỏ hàng của mình. 2. Khi người dùng xem giỏ hàng, họ có thể sửa số lượng sản phẩm bằng cách nhập số lượng mới vào ô số lượng và nhấp vào nút "Cập nhật" bên cạnh sản phẩm mà họ muốn sửa.	2.1 Hệ thống kiểm tra xem số lượng sản phẩm mới có khả dụng hay không. Nếu số lượng sản phẩm mới lớn hơn số lượng sản phẩm có sẵn trong kho, hệ thống sẽ hiển thị thông báo lỗi. 2.2 Nếu số lượng sản phẩm mới hợp lệ, hệ thống cập nhật lại số lượng sản phẩm trong giỏ hàng của người dùng. 2.3 Hệ thống cập nhật lại giỏ hàng và hiển thị thông báo cho người dùng về việc sửa sản phẩm trong giỏ hàng thành công.
EXCEPTION CONDITION	Người dùng không thể sửa số lượng sản phẩm trong giỏ hàng nếu sản phẩm đã hết hàng hoặc không có sẵn. Nếu người dùng nhập số lượng sản phẩm là số âm hoặc không hợp lệ, hệ thống sẽ hiển thị thông báo lỗi. Nếu người dùng chưa đăng nhập, họ sẽ được yêu cầu đăng nhập hoặc đăng ký trước khi sửa sản phẩm trong giỏ hàng.	

Bảng 1. 10 Đặc tả use case cập nhật giỏ hàng

- Xóa sản phẩm giỏ hàng

USECASE NAME	Xóa sản phẩm ra khỏi giỏ hàng	
TRIGGERING EVENT	Nhấn xóa sản phẩm	
BRIEF DESCRIPTION	Người dùng có thể xóa sản phẩm khỏi giỏ hàng của mình trong khi đang mua sắm trên trang web của cửa hàng.	
ACTORS	User, Admin	
STAKEHOLDER	User, Admin	
PRECONDITION	Nhấn xóa sản phẩm ra khỏi giỏ hàng	
POSTCONDITION	Quay về trang chủ của hệ thống.	
FLOW OF EVENTS	Actor	System
	1. Người dùng truy cập trang web của cửa hàng và xem giỏ hàng của mình. 2. Khi người dùng xem giỏ hàng, họ có thể nhấp vào nút "Xóa" hoặc "Bỏ sản phẩm" bên cạnh sản phẩm mà họ muốn xóa khỏi giỏ hàng.	2.1 Hệ thống xác nhận lại việc xóa sản phẩm khỏi giỏ hàng bằng cách hiển thị thông báo yêu cầu xác nhận của người dùng. 2.2 Nếu người dùng xác nhận muốn xóa sản phẩm khỏi giỏ hàng, hệ thống xóa sản phẩm khỏi giỏ hàng của người dùng. 2.3 Hệ thống cập nhật lại giỏ hàng và hiển thị thông báo cho người dùng về việc xóa sản phẩm ra khỏi giỏ hàng thành công.
EXCEPTION CONDITION	<p>Người dùng không thể xóa sản phẩm ra khỏi giỏ hàng nếu sản phẩm đã được thanh toán hoặc đã được chuyển đến quá trình xử lý đơn hàng.</p> <p>Người dùng không thể xóa sản phẩm ra khỏi giỏ hàng nếu sản phẩm đã được thanh toán hoặc đã được chuyển đến quá trình xử lý đơn hàng.</p> <p>Người dùng không thể xóa sản phẩm ra khỏi giỏ hàng nếu sản phẩm đã được thanh toán hoặc đã được chuyển đến quá trình xử lý đơn hàng.</p>	

Bảng 1. 11 Đặc tả use case xóa sản phẩm ra khỏi giỏ hàng

1.3.8 Quản lý đơn hàng

- Tạo đơn hàng

USECASE NAME	Tạo đơn hàng	
TRIGGERING EVENT	Người dùng muốn tạo một đơn hàng mới	
BRIEF DESCRIPTION	Khi người tạo một đơn hàng cho khách hàng để tiến hành thanh toán	
ACTORS	User, admin	
STAKEHOLDER	User, admin	
PRECONDITION	Người dùng phải đăng nhập thành công vào ứng dụng	
POSTCONDITION	Nếu đăng nhập thành công thì có thể tiến hành tạo đơn hàng	
FLOW OF EVENTS	Actor	System
	1. Người dùng bấm chọn tab đơn hàng. 2. Người dùng nhập các số liệu cần thiết như mặt hàng, số lượng mua, mã giảm, ... rồi bấm “Thanh toán”	1.1 Hệ thống hiển thị form để tạo đơn hàng 2.1 Tiến hành in hóa đơn và gửi cho khách
EXCEPTION CONDITION	2.1.1 Nếu nhập thiếu thông tin cần thiết thì yêu cầu nhập đủ	

Bảng 1. 12 Đặc tả use case tạo đơn hàng

- Cập nhật thông tin đơn hàng

USECASE NAME	Cập nhật thông tin đơn hàng	
TRIGGERING EVENT	Người dùng muốn cập nhật thông tin đơn hàng	
BRIEF DESCRIPTION	Người dùng muốn cập nhật thông tin đơn hàng thì có sự thiếu sót hoặc là thay đổi về sản phẩm mà khách muốn mua	
ACTORS	User, admin	
STAKEHOLDER	User, admin	
PRECONDITION	Người dùng phải đăng nhập thành công vào ứng dụng và đơn hàng đã được tạo trước đó.	
POSTCONDITION	Nếu đăng nhập và tạo đơn thành công trước đó, người dùng có thể sửa thông tin của đơn hàng	
FLOW OF EVENTS	Actor	System
	1. Người dùng bấm chọn tab đơn hàng. 2. Người dùng bấm chọn “Sửa”.	1.1 Hệ thống hiển thị các đơn hàng đã tạo.

	3. Cập nhật lại thông tin của đơn hàng đó và nhấn “OK”	2.1 Hiện thị form của đơn hàng đó và cho phép người dùng edit vào đơn hàng. 3.1 Thông báo cập nhật đơn hàng thành công.
EXCEPTION CONDITION	Không có	

Bảng 1. 13 Đặc tả use case cập nhật thông tin đơn hàng

- Hủy đơn hàng

USECASE NAME	Hủy đơn hàng	
TRIGGERING EVENT	Khi người dùng muốn hủy đơn hàng đã tạo	
BRIEF DESCRIPTION	Người dùng muốn hủy đơn hàng đã tạo khi khách không mua nữa.	
ACTORS	User, admin	
STAKEHOLDER	User, admin	
PRECONDITION	Người dùng phải đăng nhập thành công vào ứng dụng và đơn hàng phải được tạo thành công trước đó	
POSTCONDITION	Nếu đăng nhập và tạo đơn thành công thì người dùng có thể hủy đơn hàng	
FLOW OF EVENTS	Actor	System
	1. Người dùng bấm chọn tab đơn hàng. 2. Người dùng bấm chọn vào check box của đơn hàng mà mình muốn hủy. 3. Bấm chọn “Ok” và hủy đơn hàng.	1.1 Hệ thống hiển thị các đơn hàng đã tạo. 2.1 Hiện thị form thông báo xác nhận hủy Ok / Cancel. 3.1 Hiện thị thông báo hủy đơn hàng thành công.
EXCEPTION CONDITION	2.2.1 Nếu chọn OK thì hủy đơn thành công 2.2.2 Nếu chọn Cancel thì đơn hàng vẫn tiếp tục được thực hiện.	

Bảng 1. 14 Đặc tả use case hủy đơn hàng

1.3.9 Quản lý user

- Cập nhật thông tin user

USECASE NAME	Cập nhật thông tin user	
TRIGGERING EVENT	Admin muốn thay đổi thông tin của user khi được yêu cầu hoặc phát hiện sai sót	
BRIEF DESCRIPTION	Admin muốn thay đổi thông tin của user khi được yêu cầu hoặc phát hiện sai sót	
ACTORS	Admin	
STAKEHOLDER	Admin	
PRECONDITION	Người dùng phải đăng nhập thành công với vai trò là admin và user phải có trong list user	
POSTCONDITION	Nếu đăng nhập thành công thì có thể cập nhật thông tin của khách hàng	
FLOW OF EVENTS	Actor	System
	1. Admin bấm chọn tab thiết lập. 2. Admin bấm chọn mục quản lý user 3. Bấm chọn “Sửa” để tiến hành thay đổi thông tin cho user 4. Bấm “OK” để thay đổi thông tin user	1.1 Hiện thị mục quản lý 2.1 Hiện thị danh sách thông tin của tất cả các user. 3.1 Hiện thị chi tiết những thông tin của khách hàng. 4.1 Hiện thông báo cập nhật thành công.
EXCEPTION CONDITION	4.1.1 Nếu bấm “OK” thì hiện thông báo thành công 4.1.2 Nếu bấm “Cancel” thì quay trở lại thông tin của user	

Bảng 1. 15 Đặc tả use case cập nhật thông tin user

- Xóa user

USECASE NAME	Xóa user
TRIGGERING EVENT	Admin muốn xóa user
BRIEF DESCRIPTION	Admin muốn xóa user trong danh sách user
ACTORS	Admin
STAKEHOLDER	Admin

PRECONDITION	Người dùng phải đăng nhập thành công với vai trò là admin và user cần xóa phải có trong danh mục quản lý user	
POSTCONDITION	Nếu đăng nhập thành công thì có thể xóa user cần xóa	
FLOW OF EVENTS	Actor	System
	1. Admin bấm chọn tab thiết lập 2. Admin bấm chọn mục quản lý user 3. Admin bấm chọn nút “xóa” để xóa user cần xóa. 3. Nhấn chọn nút “OK” để tiến hành xóa	1.1 Hệ thống hiển thị mục quản lý 2.1 Hiện thị danh sách thông tin của tất cả các user 3.1 Hiện thị thông báo bạn có chắc muốn xóa user này. 4.1 Hiện thị thông báo xóa thành công.
EXCEPTION CONDITION	3.1.1 Nếu chọn OK thì nhân viên được chọn sẽ bị xóa 3.1.2 Nếu chọn Cancel thì hủy thao tác xóa	

Bảng 1. 16 Đặc tả use case xóa user

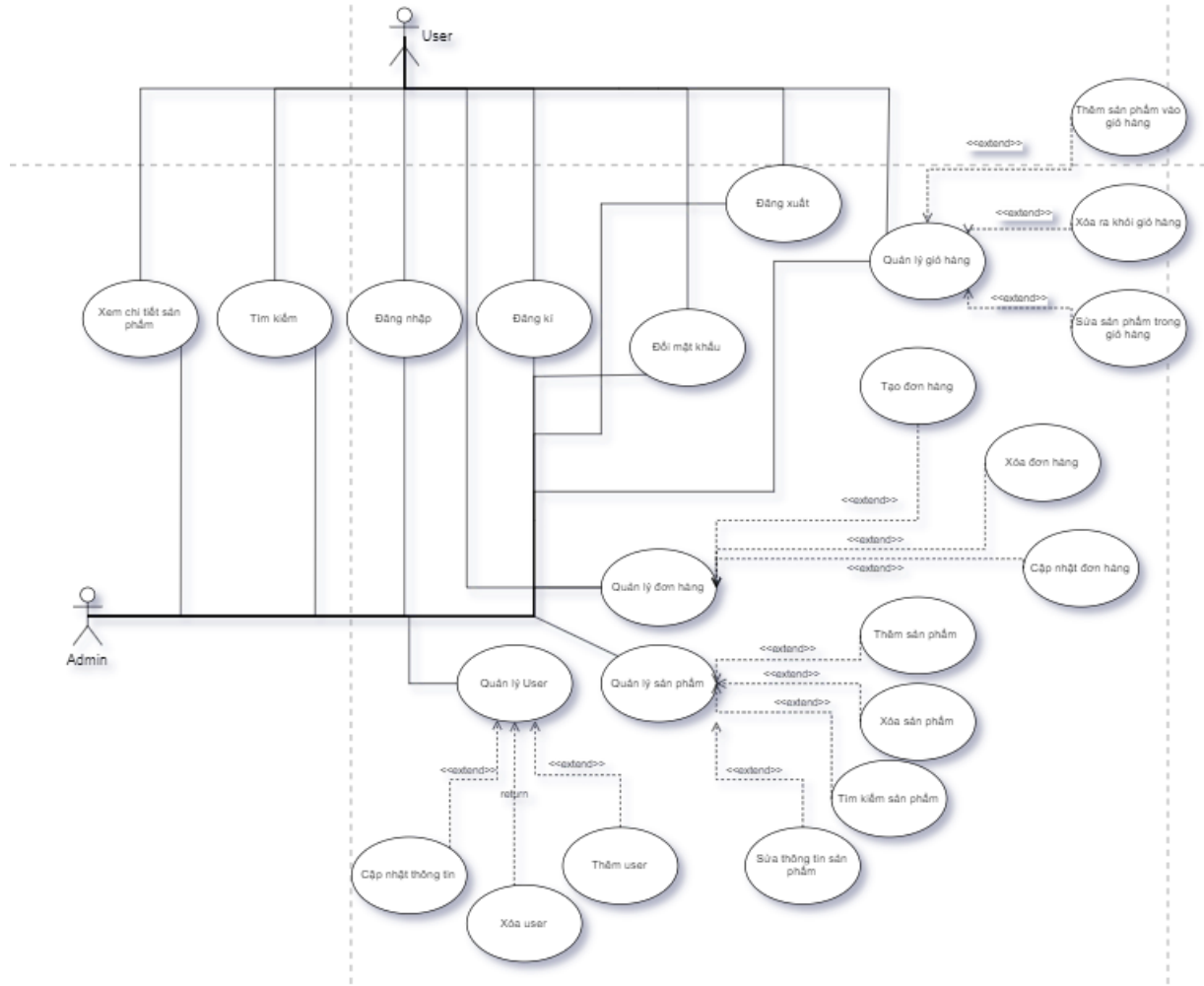
1.3.10 Xem chi tiết đơn hàng

USECASE NAME	Xem chi tiết đơn hàng	
TRIGGERING EVENT	Người dùng muốn biết thêm chi tiết về hóa đơn mua hàng	
BRIEF DESCRIPTION	Người dùng xem chi tiết thông tin hóa đơn mua hàng để kiểm tra	
ACTORS	User, Admin	
STAKEHOLDER	User, Admin	
PRECONDITION	Không có	
POSTCONDITION	Quay về trang chủ của hệ thống, yêu cầu đăng nhập.	
FLOW OF EVENTS	Actor	FLOW OF EVENTS
	1. Người dùng bấm chọn hóa đơn muốn kiểm tra 2. Người dùng tiến hành xem thông tin của hóa đơn	2. Hiện thị thông tin của hóa đơn
EXCEPTION CONDITION	Không có.	

Bảng 1. 17 Đặc tả use case xem chi tiết đơn hàng

1.4 Thiết kế sơ đồ

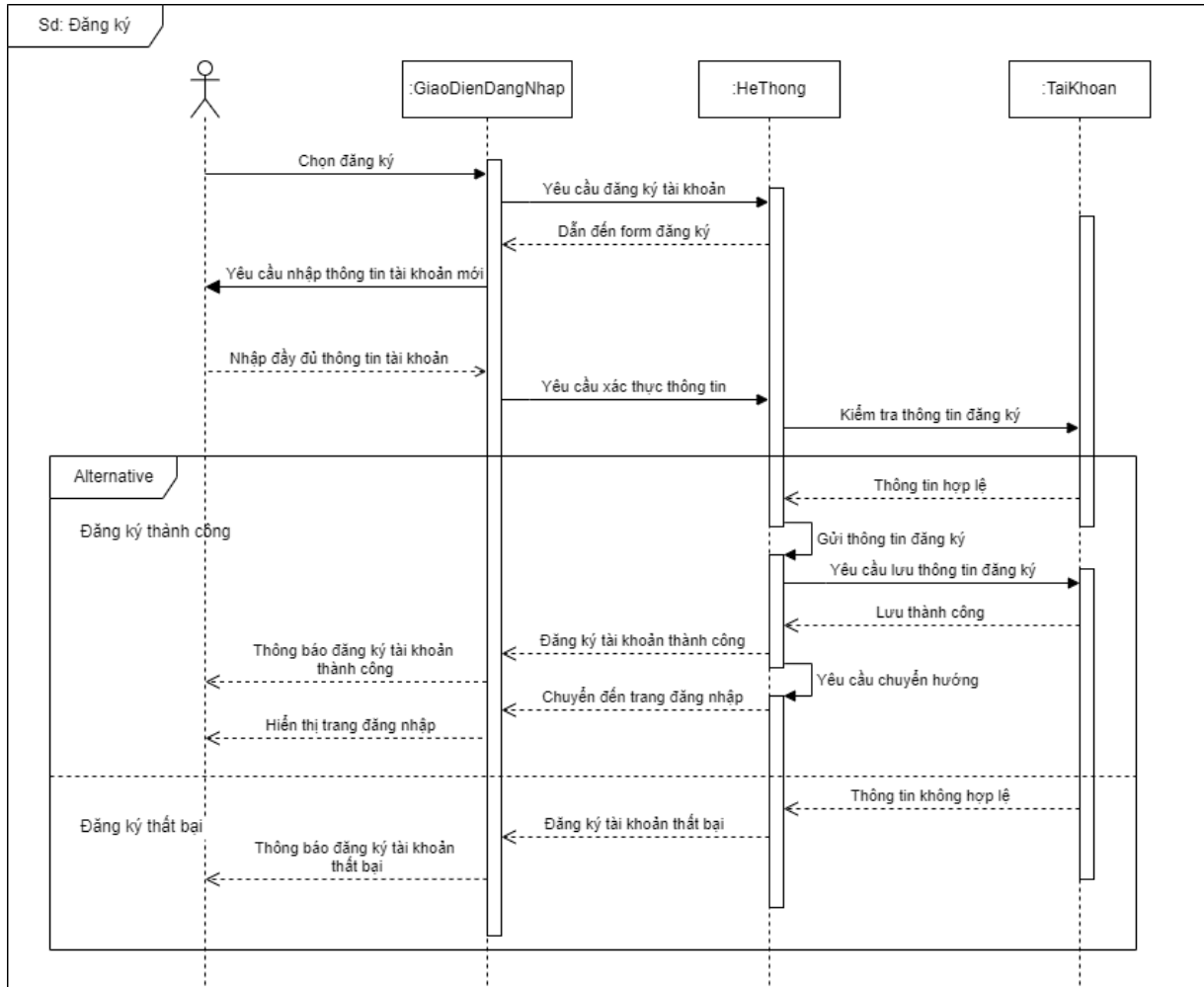
1.4.1 Use Case



Hình 1. 4 Use case tổng quát

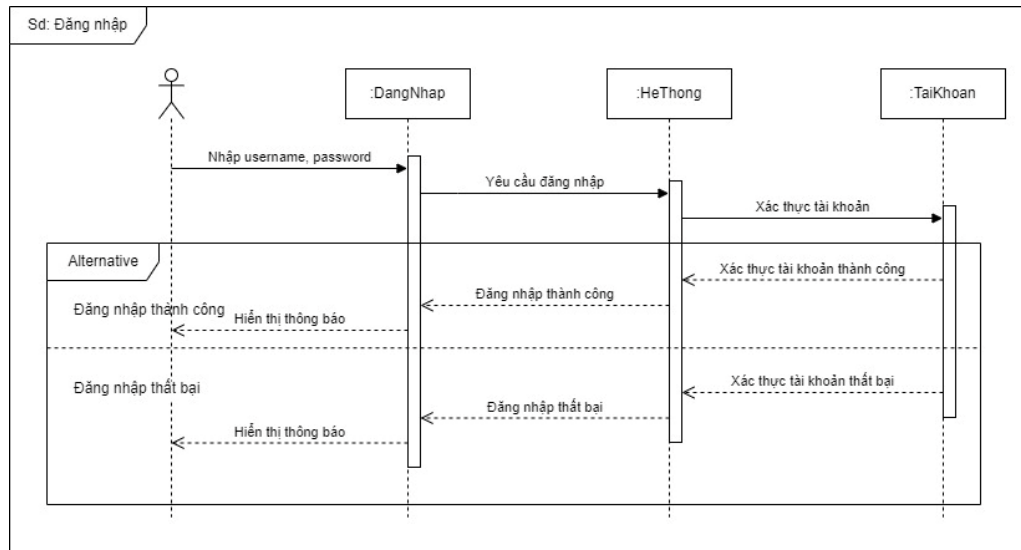
1.4.2 Sequence

1.4.2.1 Đăng kí



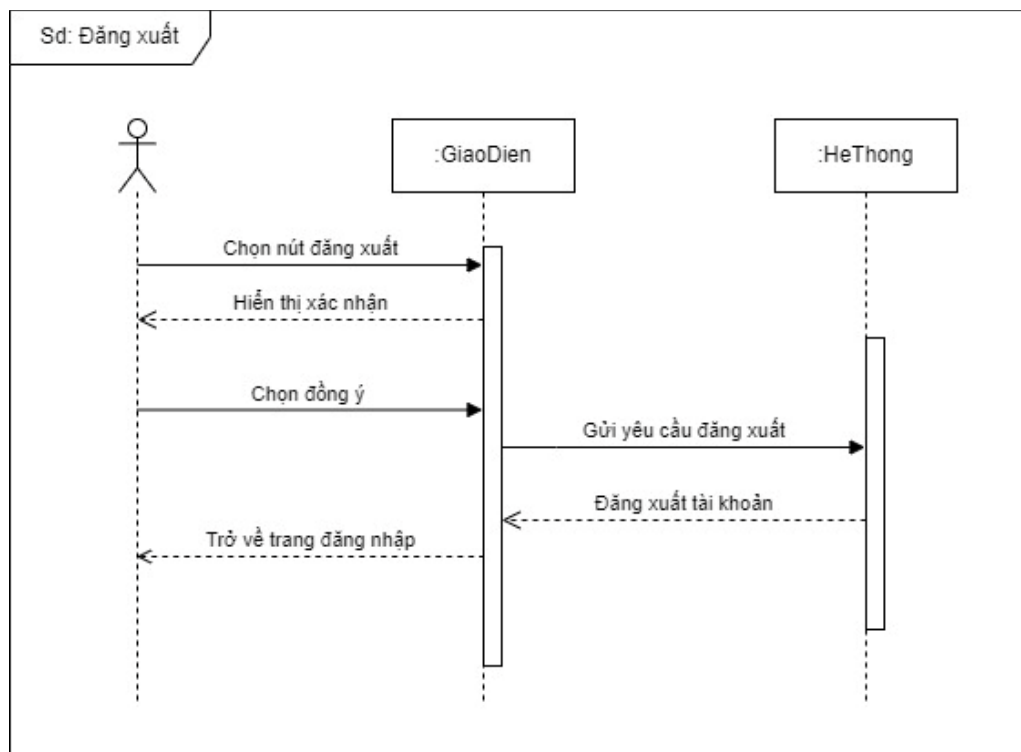
Hình 1. 5 Sequence đăng kí

1.4.2.2 Đăng nhập



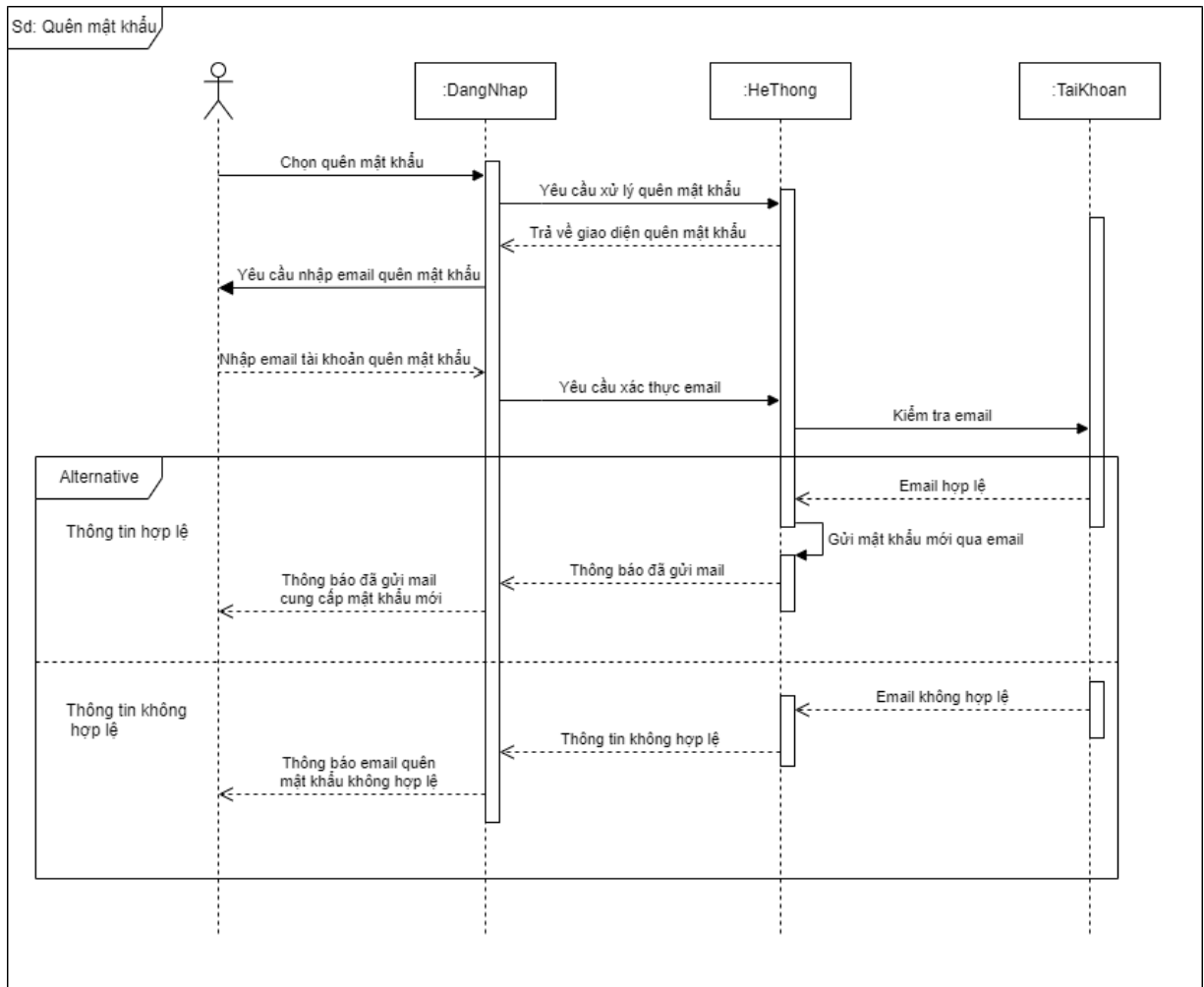
Hình 1. 6 Sequence đăng nhập

1.4.2.3 Đăng xuất



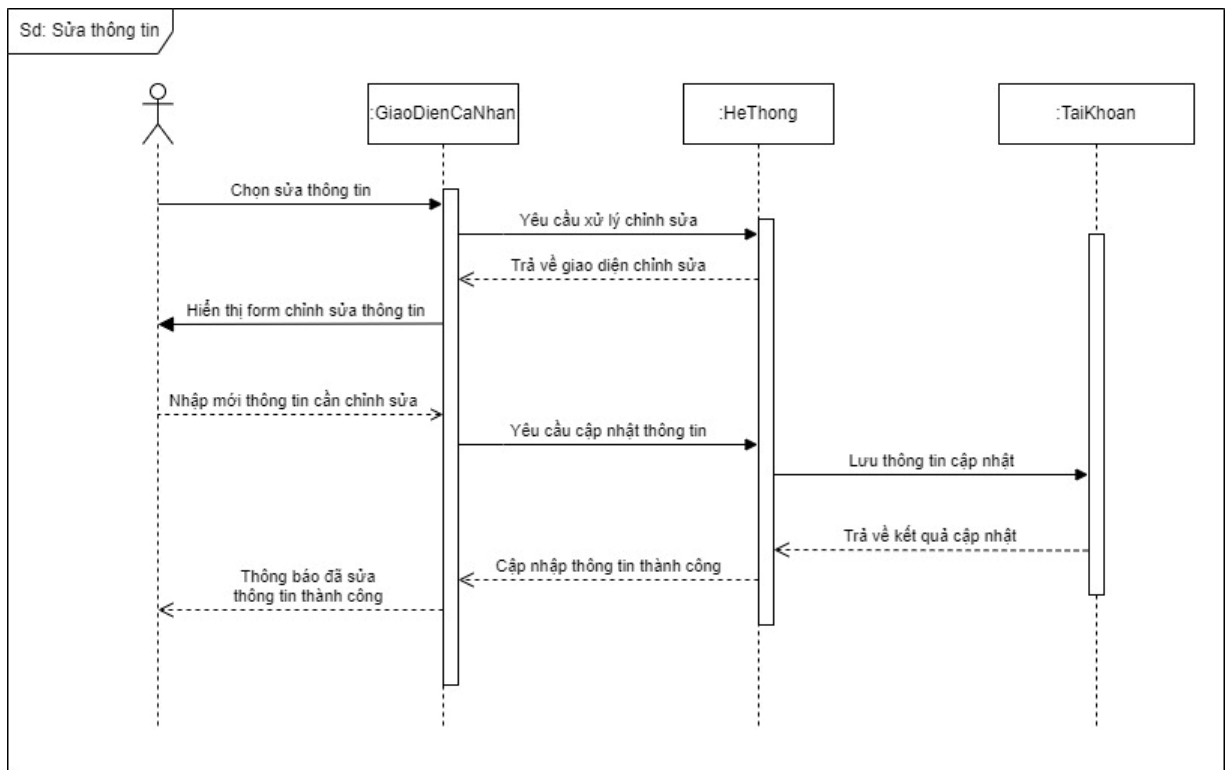
Hình 1. 7 Sequence đăng xuất

1.4.2.4 Quên mật khẩu



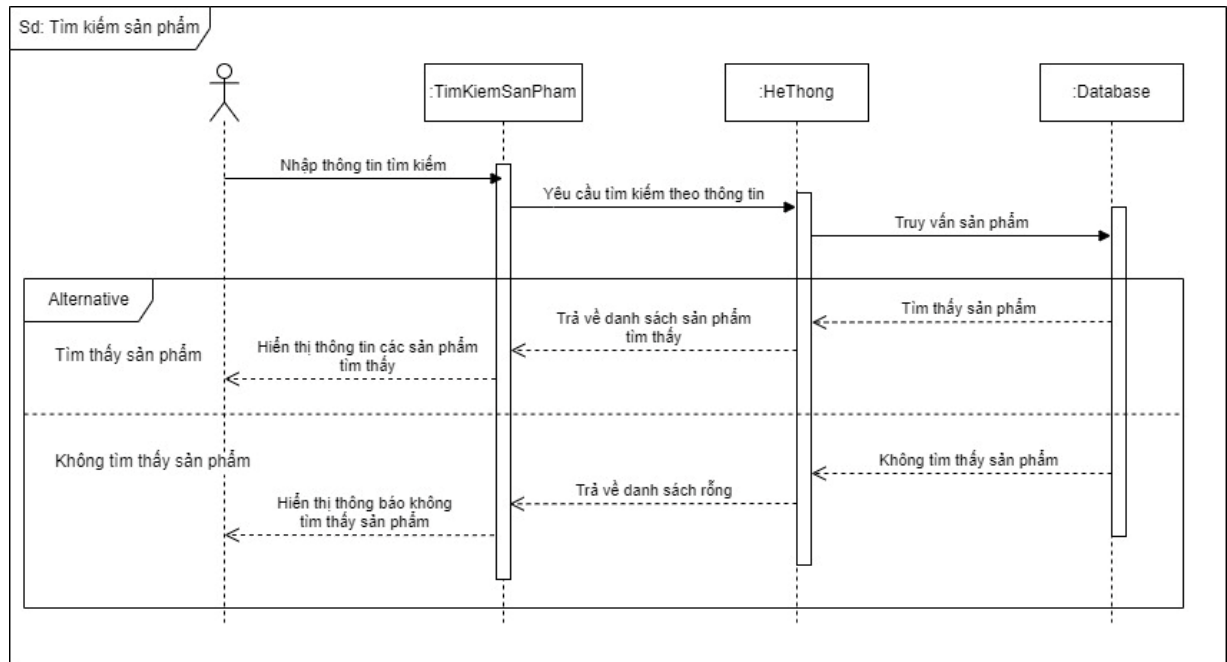
Hình 1. 8 Sequence quên mật khẩu

1.4.2.5 Sửa thông tin



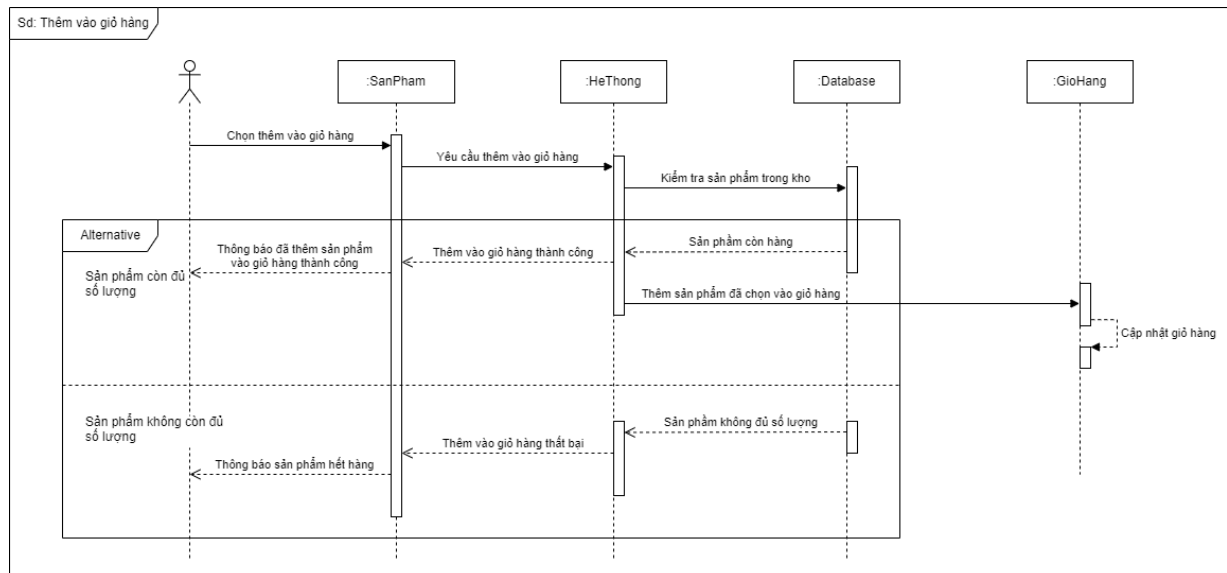
Hình 1. 9 Sequence sửa thông tin

1.4.2.6 Tìm kiếm sản phẩm



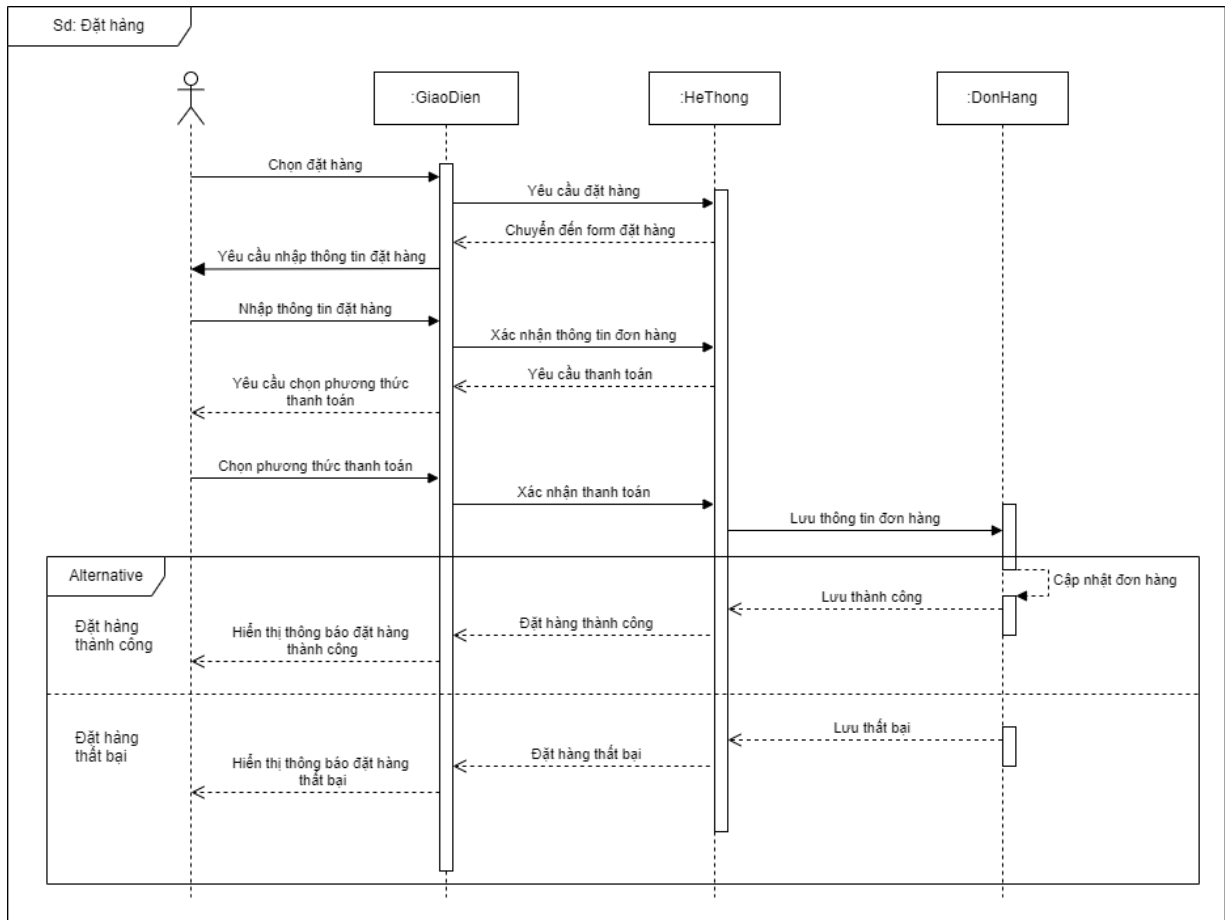
Hình 1. 10 Sequence tìm kiếm sản phẩm

1.4.2.7 Thêm sản phẩm vào giỏ hàng



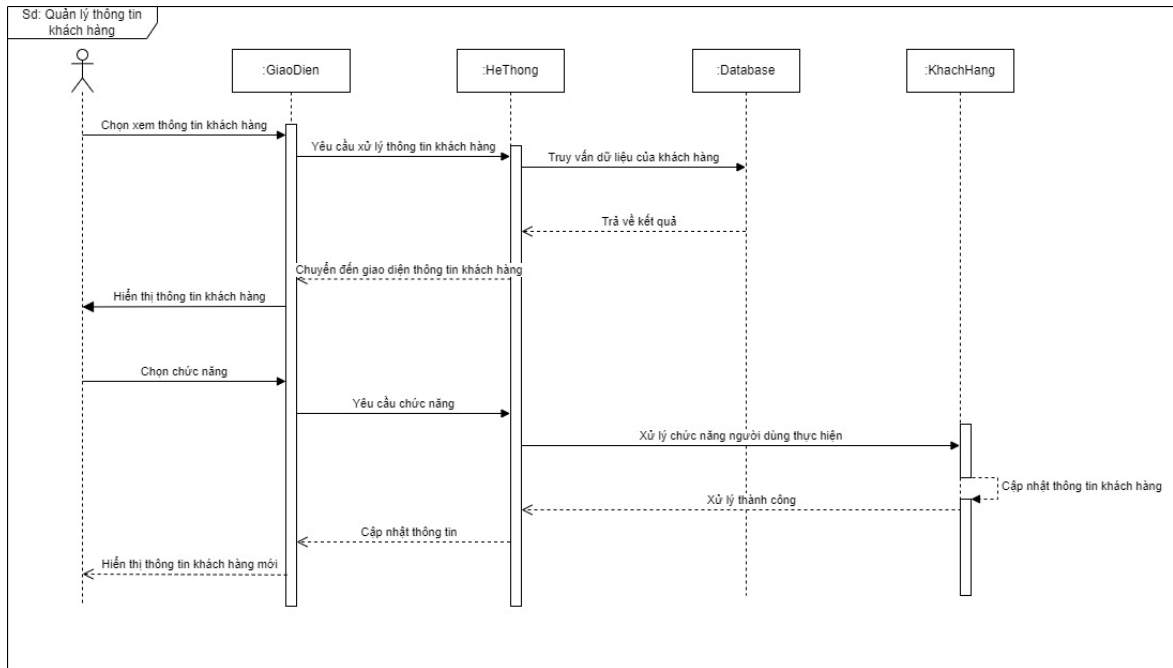
Hình 1. 11 Sequence thêm sản phẩm vào giỏ hàng

1.4.2.8 Đặt hàng



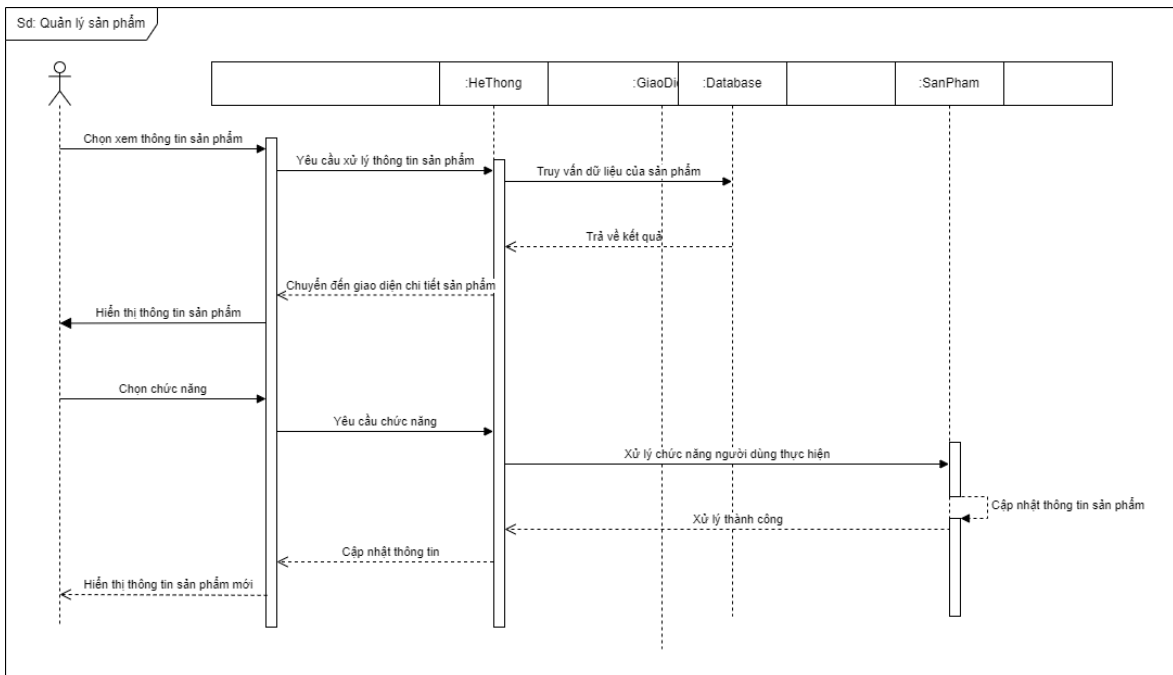
Hình 1. 12 Sequence đặt hàng

1.4.2.9 Quản lý User



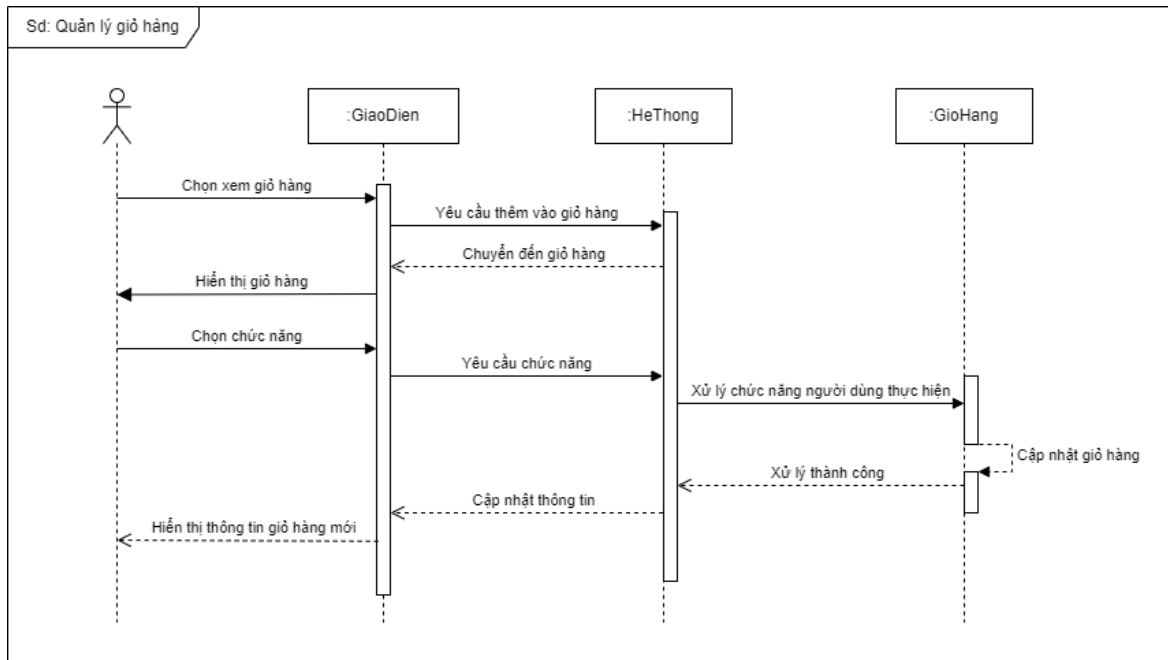
Hình 1. 13 Sequence quản lý user

1.4.2.10 Quản lý sản phẩm



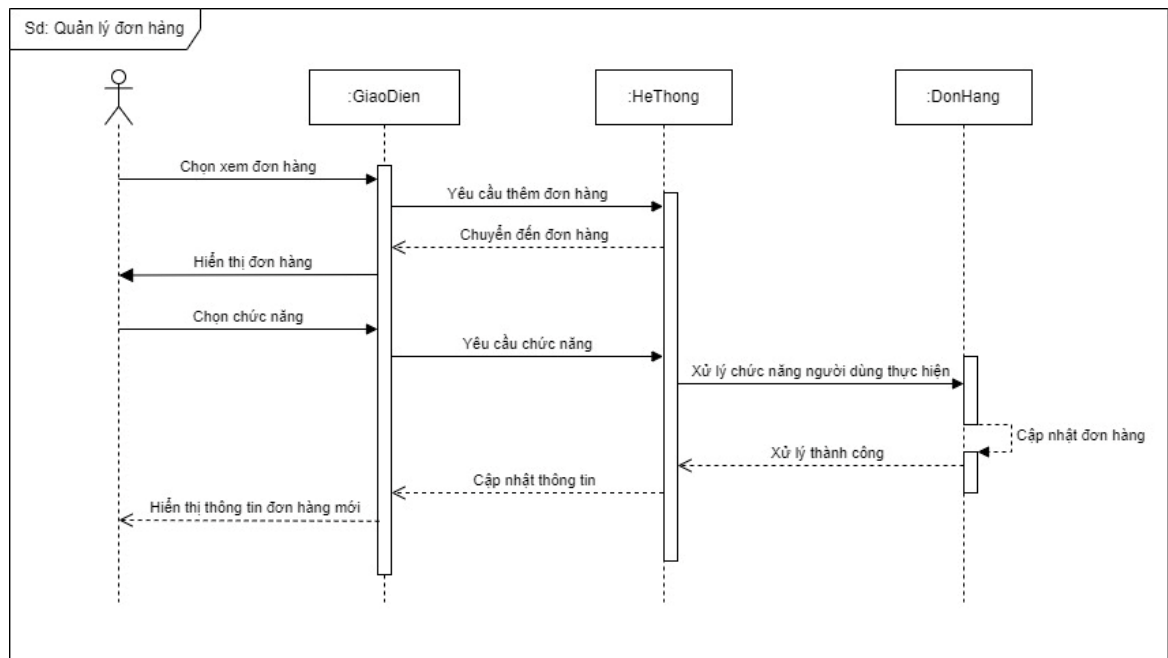
Hình 1. 14 Sequence quản lý sản phẩm

1.4.2.11 Quản lý giỏ hàng



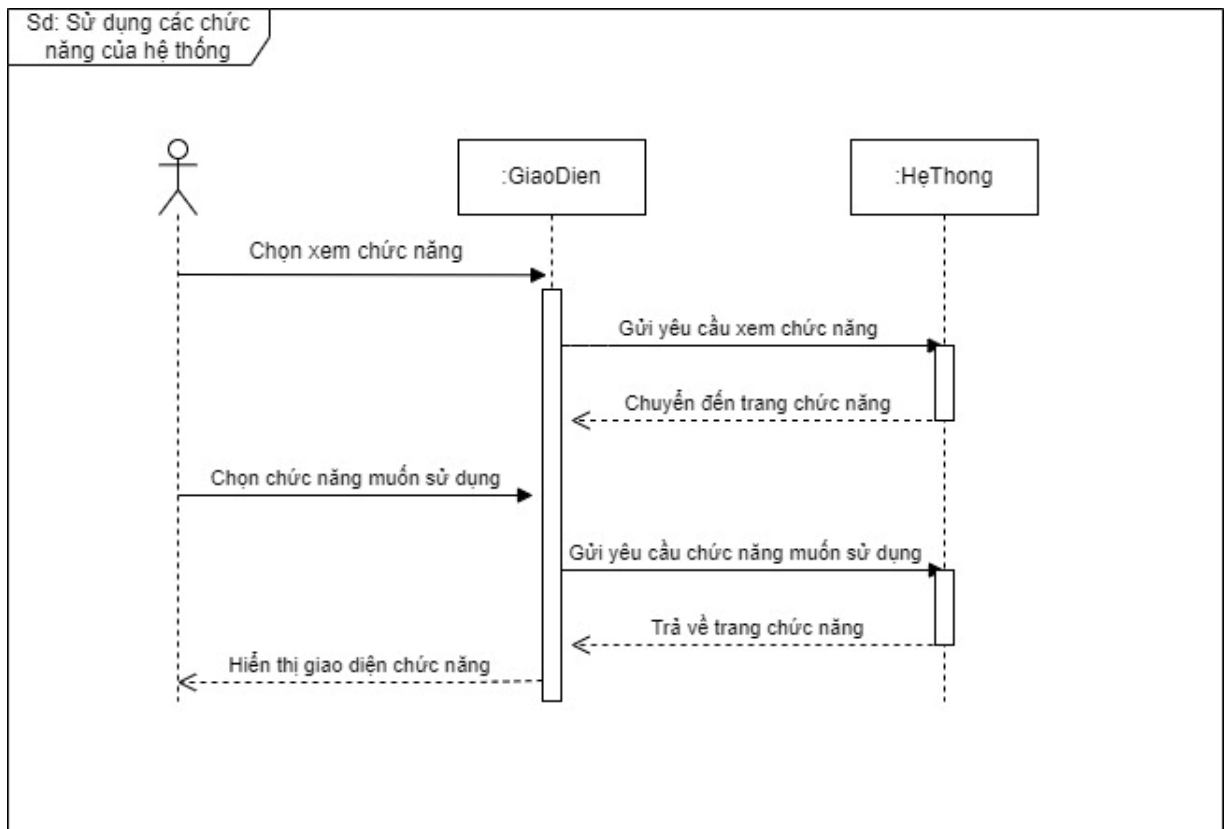
Hình 1. 15 Sequence quản lý giỏ hàng

1.4.2.12 Quản lý đơn hàng



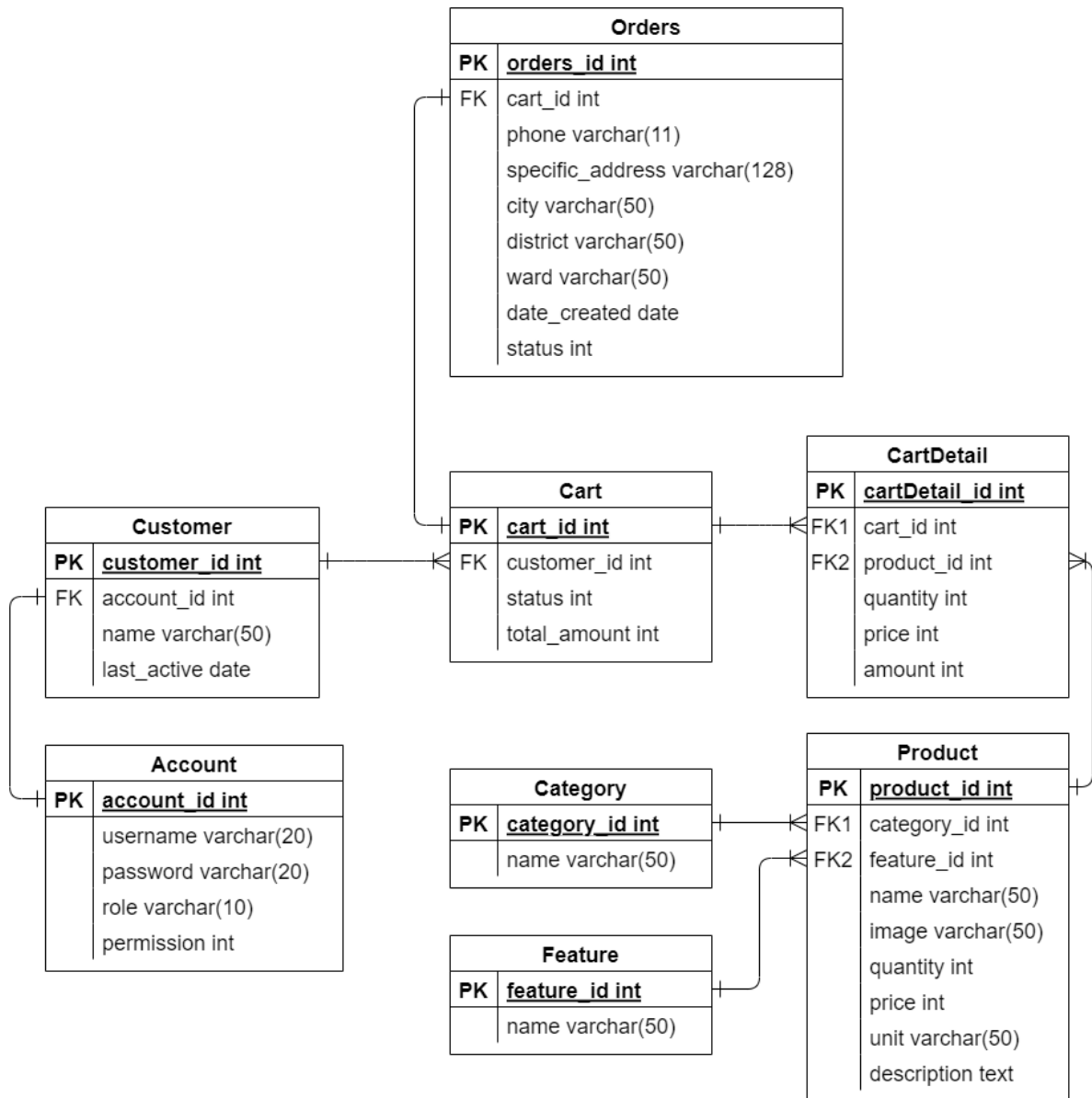
Hình 1. 16 Sequence quản lý đơn hàng

1.4.2.13 Sử dụng các chức năng



Hình 1. 17 Sequence sử dụng các chức năng CRUD

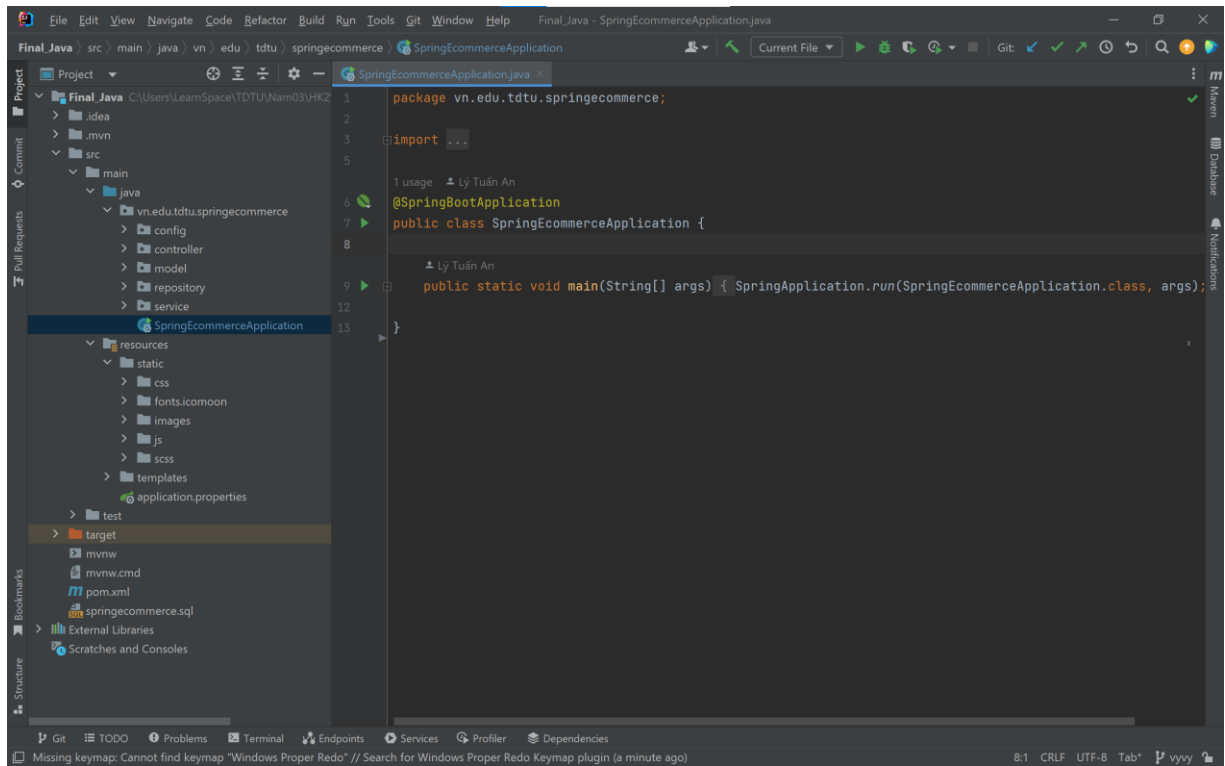
1.4.3 ERD



Hình 1. 18 Mô hình ERD

CHƯƠNG II: PHÁT TRIỂN WEBSITE VÀ DEMO

2.1 Cấu trúc project



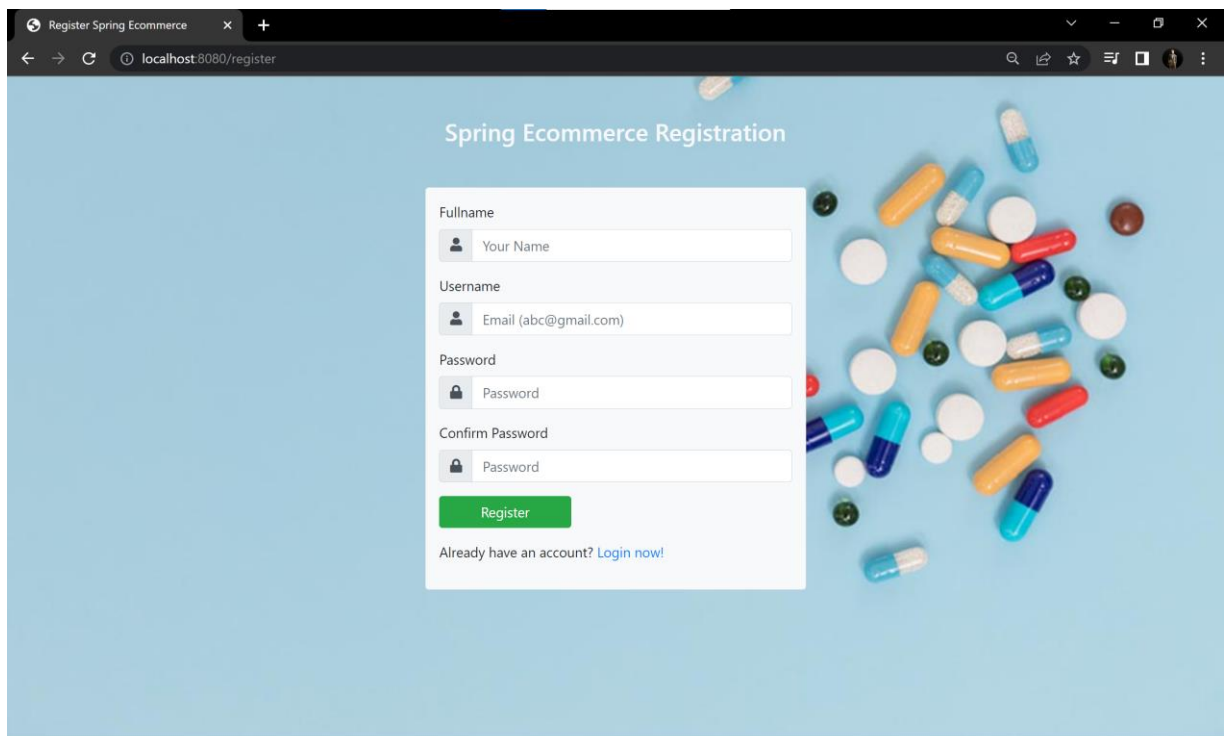
Hình 2. 1 Cấu trúc project

- Gồm có 2 folder chính gồm có:
 - o Java – package vn.edu.tdtu.springecommerce: Chứa các folder để thực hiện xử lý chức năng.
 - o Resources: Chứa templates giao diện của website.
- Folder java:
 - o Folder config: Chứa các class configuration dùng cấu hình các bean.
 - o Folder controller: Chứa các class xử lý request và response.
 - o Folder model: Chứa các class mô phỏng các table trong database.
 - o Folder repository: Chứa các interface kế thừa JPA thực hiện các xử lý đến database.

- Folder service: Xử lý các yêu cầu từ controller.
- Folder resources:
 - Folder static: Chứa các file js, css và image để thực hiện style các giao diện trong folder templates.
 - Folder templates: Chứa các file html làm layout cho website.

2.2 Demo

2.2.1 Giao diện đăng ký



Register Spring Ecommerce

localhost:8080/register

Spring Ecommerce Registration

Fullname
Your Name

Username
Email (abc@gmail.com)

Password
Password

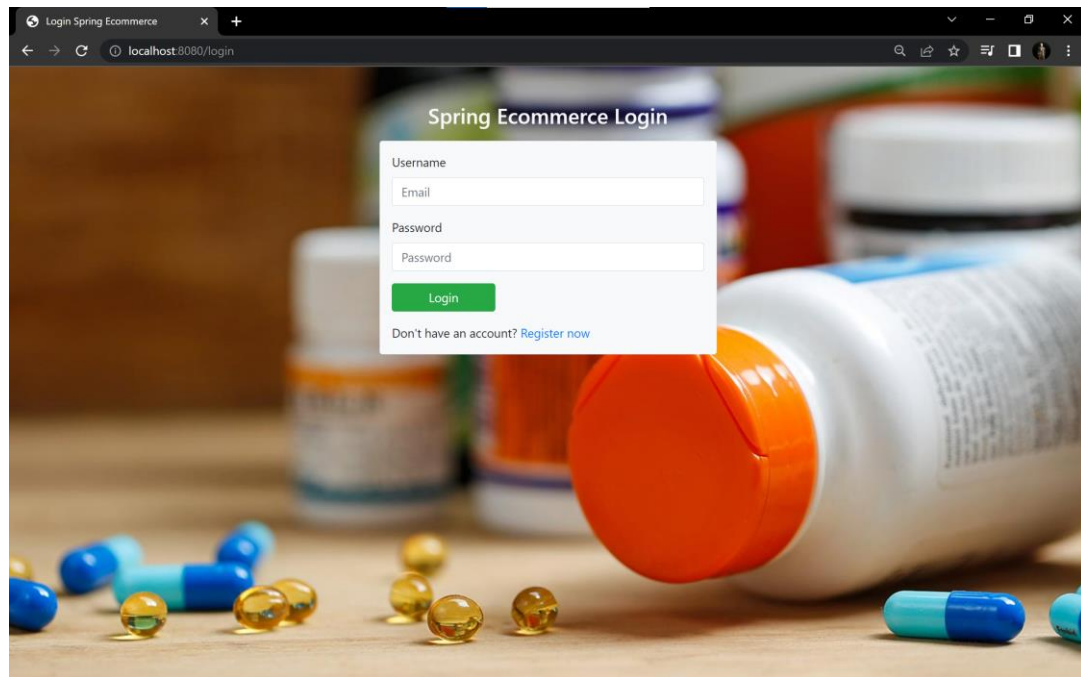
Confirm Password
Password

Register

Already have an account? [Login now!](#)

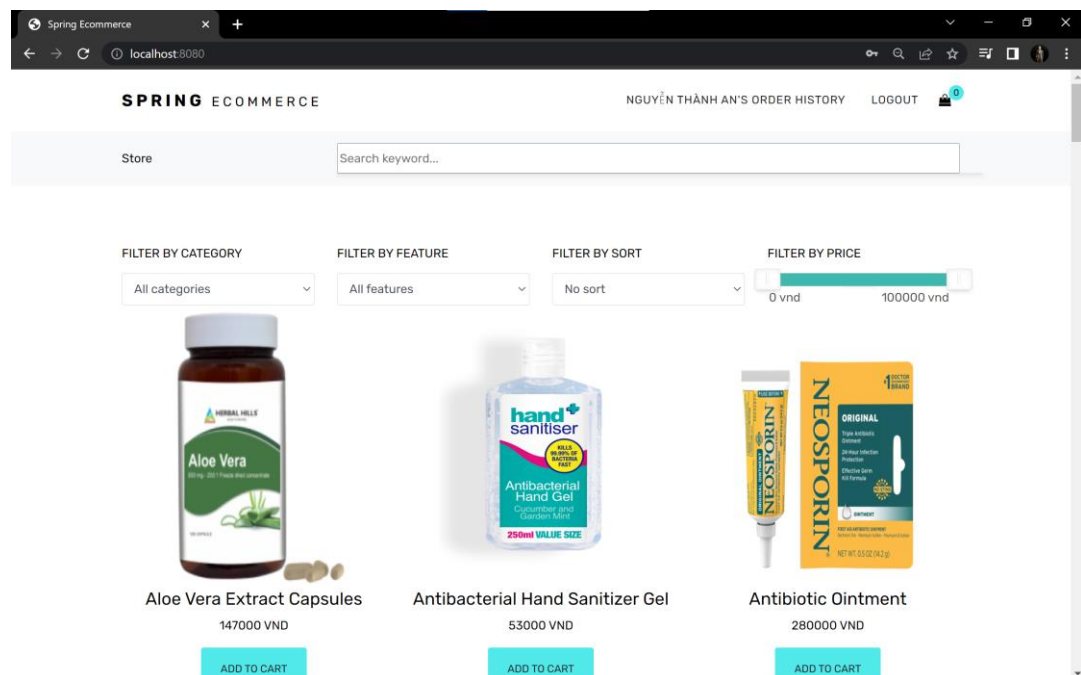
Hình 2. 2 Trang đăng ký

2.2.2 Giao diện đăng nhập



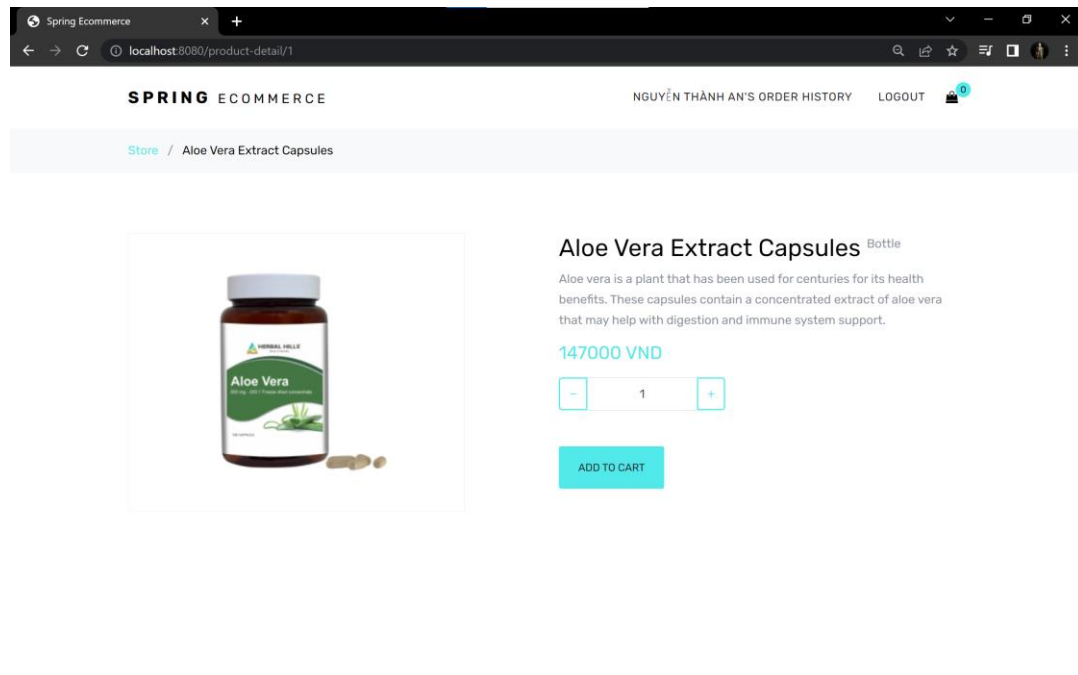
Hình 2. 3 Trang đăng nhập

2.2.3 Giao diện trang chủ



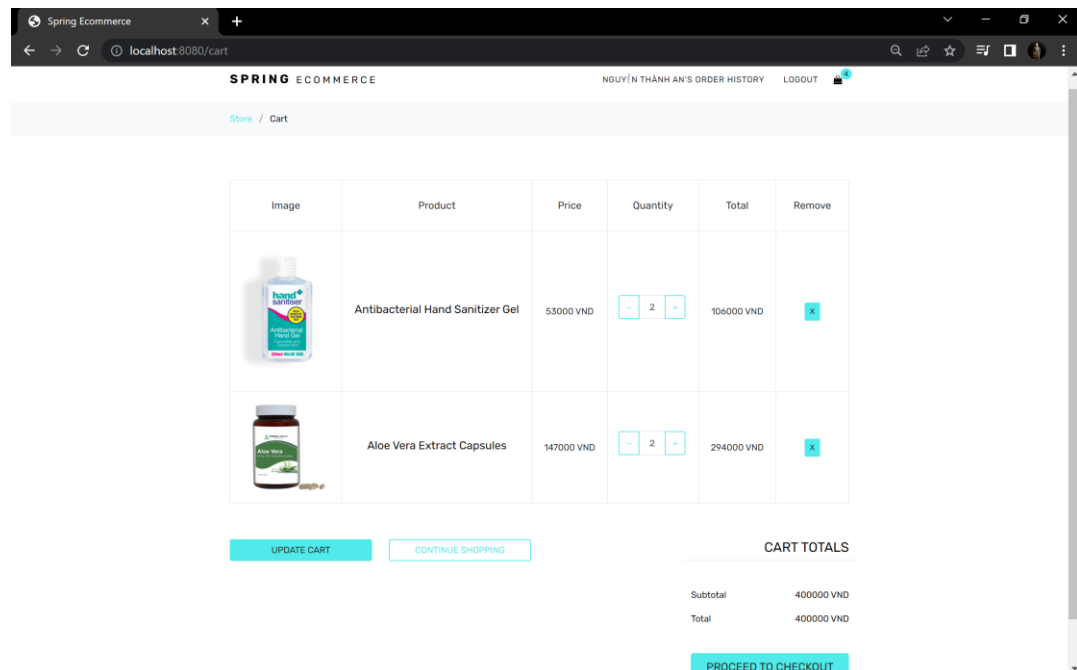
Hình 2. 4 Trang chủ

2.2.4 Giao diện trang chi tiết sản phẩm



Hình 2. 5 Trang chi tiết sản phẩm

2.2.5 Giao diện trang giỏ hàng



Hình 2. 6 Trang giỏ hàng

2.2.6 Giao diện trang tạo hóa đơn

Spring Ecommerce

localhost:8080/checkout

SPRING ECOMMERCE

NGUYỄN THÀNH AN'S ORDER HISTORY LOGOUT

Store / Cart / Checkout

Order Details

Specific Address *

Street address

Ward* District*

Ward District

City* Phone*

City Phone Number

Your Order

Product	Total
Antibacterial Hand Sanitizer Gel x 2	106000 VND
Aloe Vera Extract Capsules x 2	294000 VND
Cart Subtotal	400000 VND
Order Total	400000 VND

PLACE ORDER

Hình 2. 7 Trang tạo hóa đơn

2.2.7 Giao diện trang tạo đơn hàng thành công

Spring Ecommerce

localhost:8080/thankyou

SPRING ECOMMERCE

NGUYỄN THÀNH AN'S ORDER HISTORY LOGOUT

Store / Thank You

✓

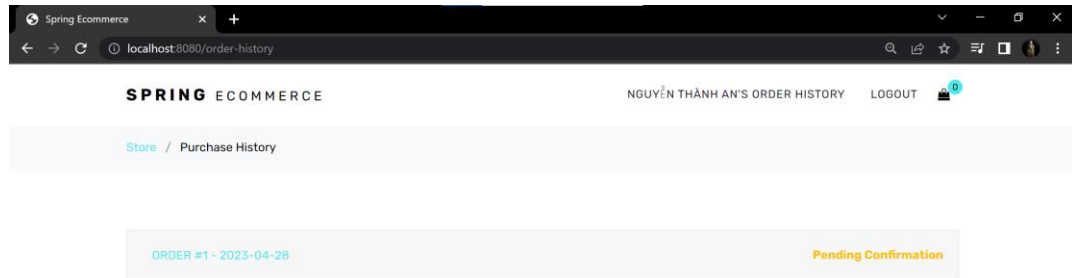
Thank you!

You order was successfully completed.

BACK TO STORE

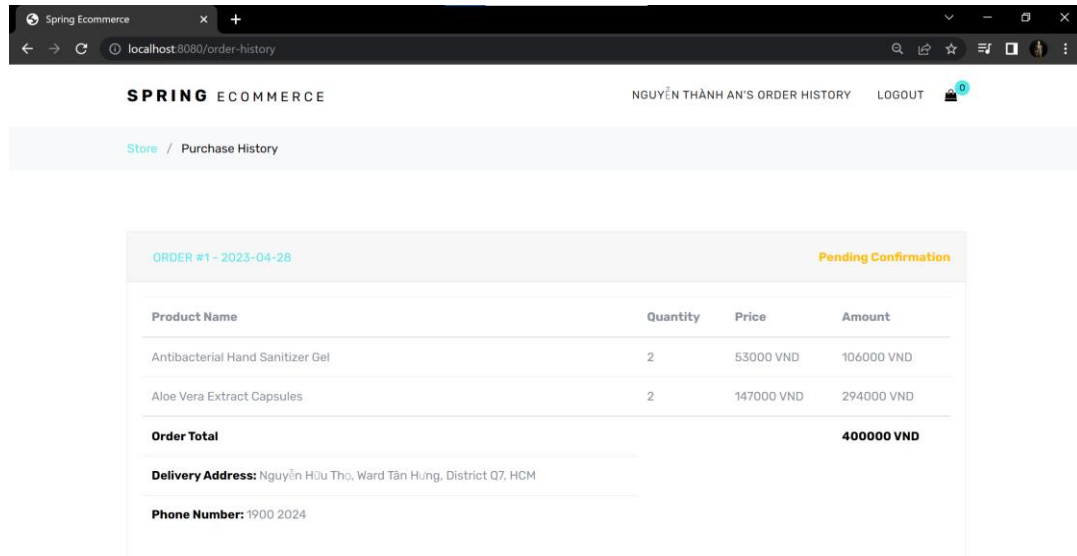
Hình 2. 8 Trang tạo hóa đơn thành công

2.2.8 Giao diện trang lịch sử đơn hàng



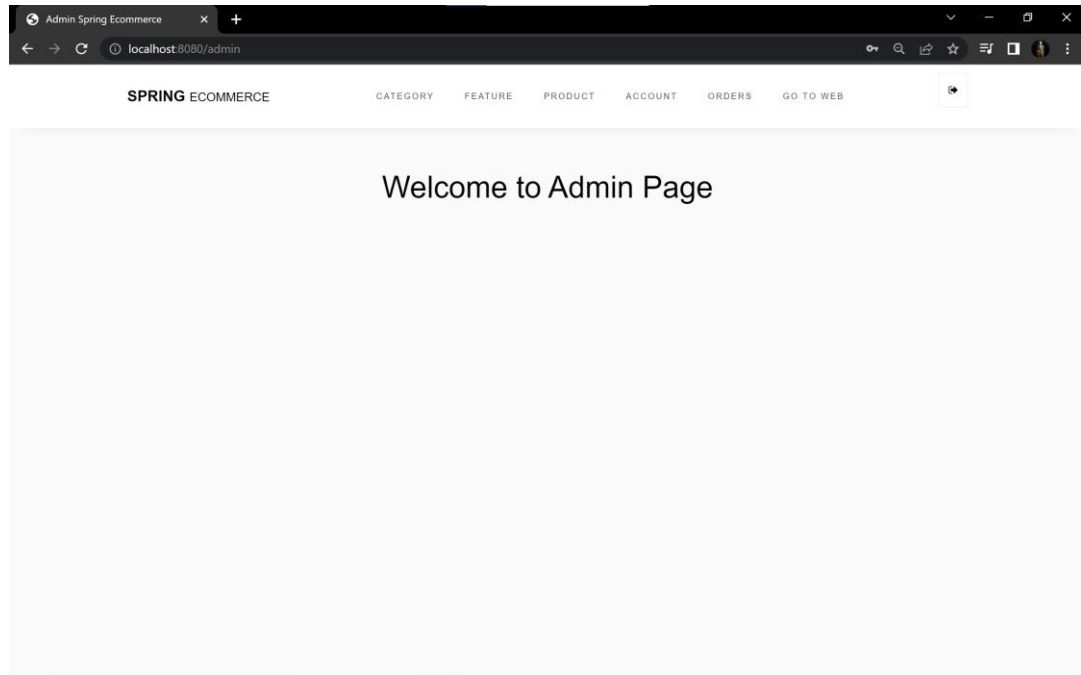
Hình 2. 9 Trang lịch sử đặt hàng

2.2.9 Giao diện trang chi tiết lịch sử đơn hàng



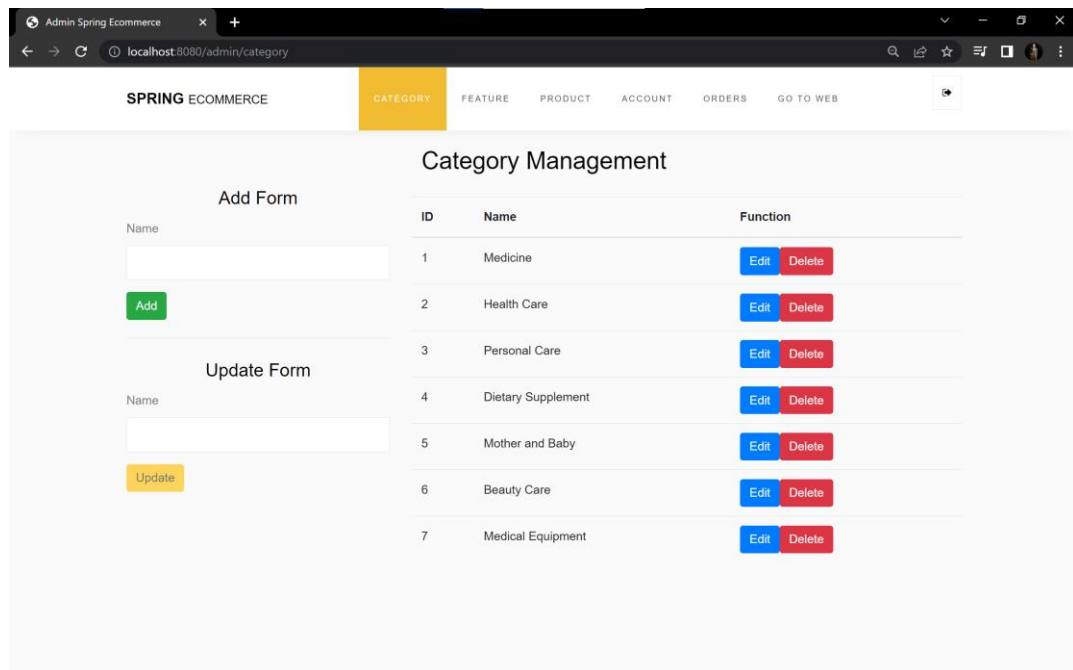
Hình 2. 10 Trang chi tiết lịch sử đơn hàng

2.2.10 Giao diện trang chủ admin



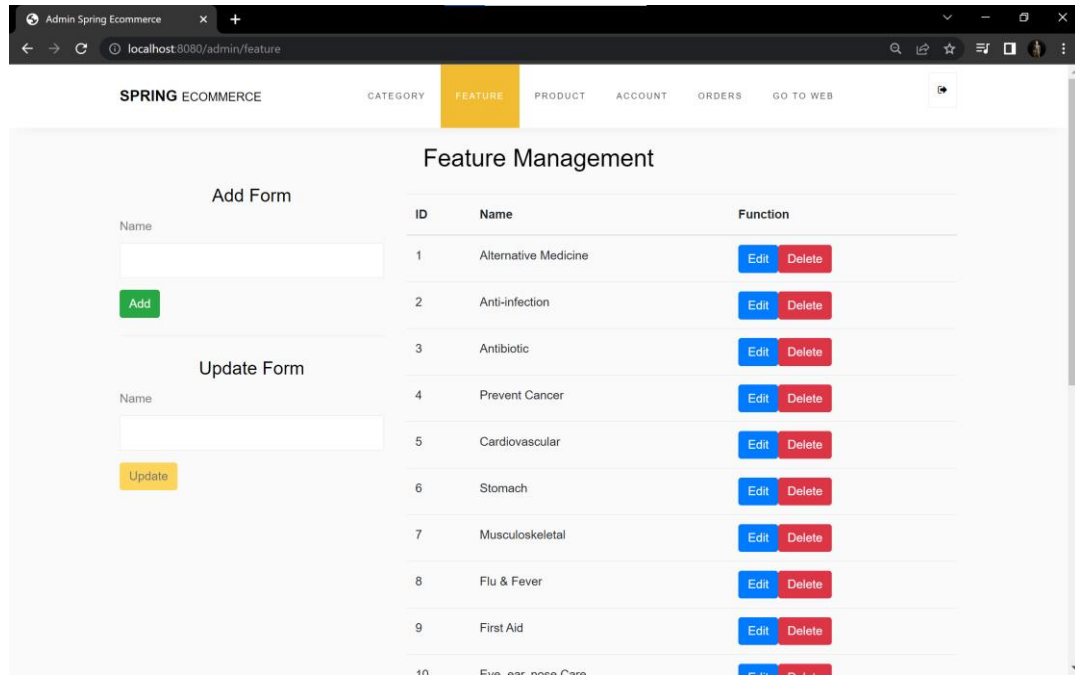
Hình 2. 11 Trang chủ admin

2.2.11 Giao diện trang quản lý danh mục sản phẩm



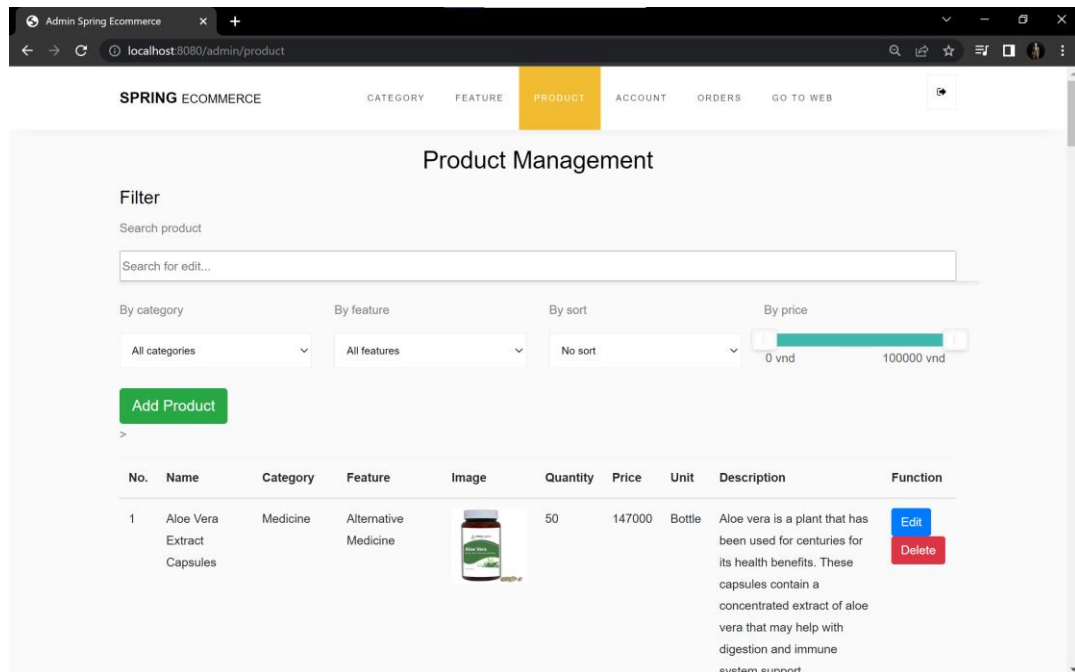
Hình 2. 12 Trang quản lý danh mục sản phẩm

2.2.12 Giao diện trang quản lý tính năng của sản phẩm



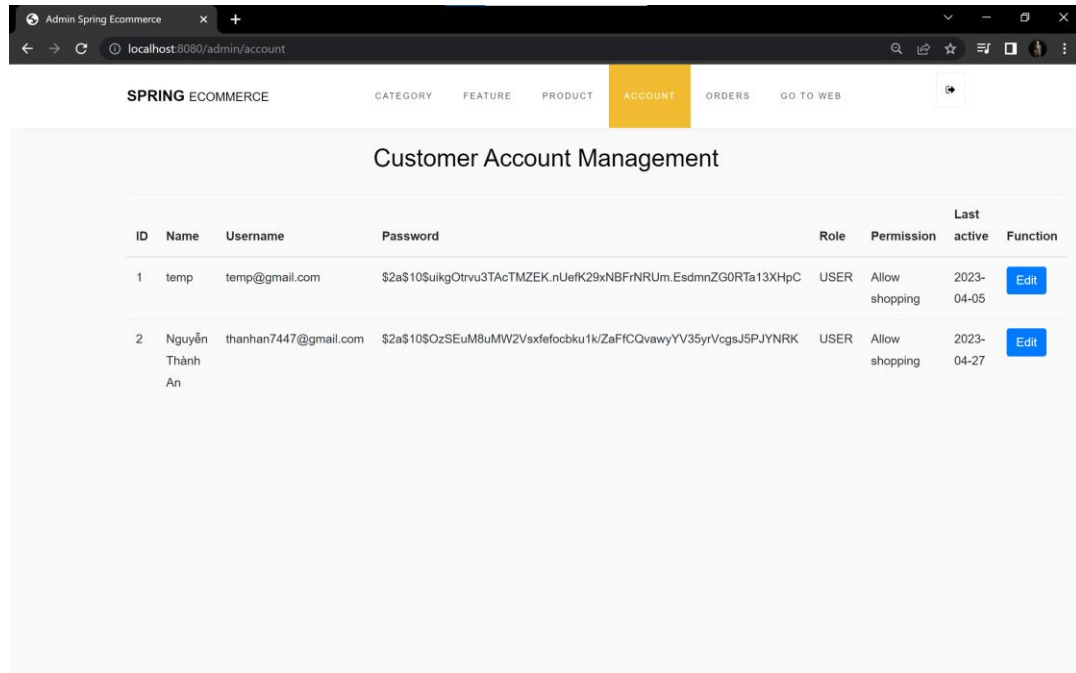
Hình 2. 13 Trang quản lý tính năng sản phẩm

2.2.13 Giao diện trang quản lý sản phẩm



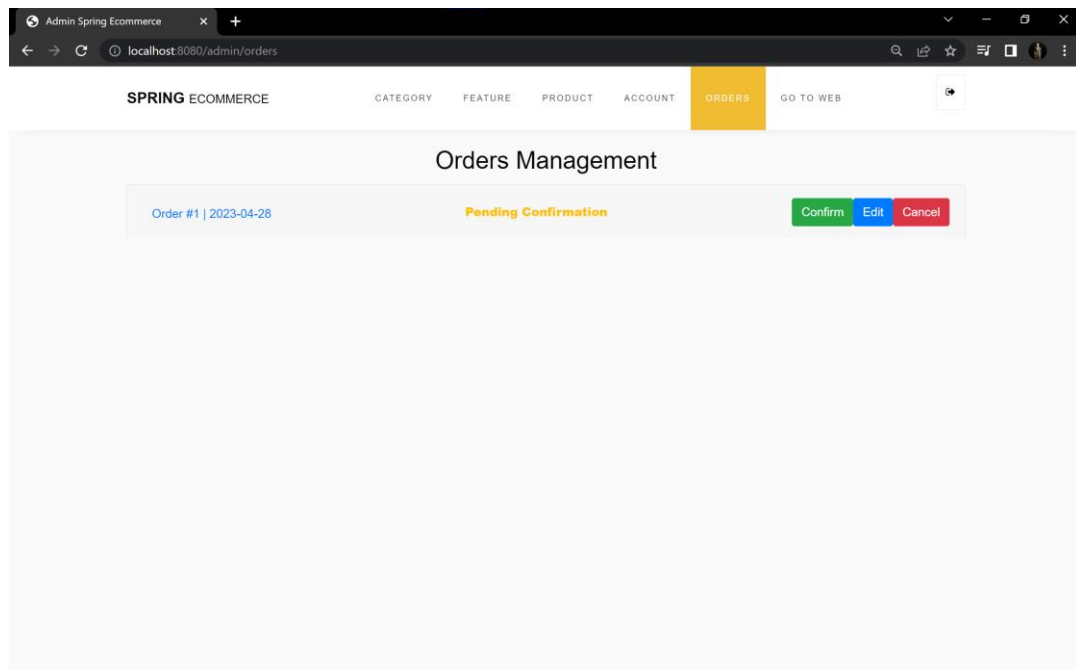
Hình 2. 14 Trang quản lý sản phẩm

2.2.14 Giao diện trang quản lý tài khoản



Hình 2. 15 Trang quản lý tài khoản

2.2.15 Giao diện trang quản lý đơn hàng



Hình 2. 16 Trang quản lý đơn hàng

2.3 Các bước thực hiện để chạy project

- Sau khi clone project trên GitHub hoặc download từ drive nộp bài thì tiến hành khởi động project lên trên IntelliJ hoặc Eclipse.

Link Git: https://github.com/devilkun1/Final_Java.git

Chạy lệnh sau: `git clone -b vyvy (link git)`

- Trong project sẽ chứa 1 file `springcommerce.sql`, ta sẽ khởi động XAMPP lên và import file `springcommerce.sql` để sử dụng database.
- Cuối cùng thì chỉ cần run project lên và có thể thực hiện các chức năng như phần demo.

CHƯƠNG III: KẾT LUẬN

3.1 Ưu điểm của trang web

- *Tính bảo mật cao*: Spring Java cung cấp nhiều tính năng bảo mật để bảo vệ thông tin khách hàng và dữ liệu giao dịch. Điều này đảm bảo rằng người dùng có thể tin tưởng vào việc mua sắm và thanh toán trên trang web mà không phải lo lắng về sự an toàn của thông tin cá nhân và tài khoản của họ.
- *Tính linh hoạt và mở rộng*: trang web có đa dạng các tính năng và chức năng dễ dàng sử dụng, website có thể được cập nhật và nâng cấp một cách nhanh chóng và linh hoạt. Ngoài ra, trang web cũng hỗ trợ kết nối với các dịch vụ khác như thanh toán trực tuyến, dịch vụ vận chuyển, hoặc các dịch vụ bổ sung khác.
- *Tốc độ và hiệu suất*: trang web có tốc độ tải nhanh hơn, giảm thiểu thời gian chờ đợi và tăng cường trải nghiệm người dùng.
- *Khả năng quản lý và xử lý dữ liệu*: Trang web có nhiều tính năng hỗ trợ quản lý và xử lý dữ liệu, giúp cho việc quản lý thông tin sản phẩm, khách hàng, đơn hàng và giao dịch trở nên dễ dàng và tiện lợi hơn.
- *Thông tin sản phẩm và đánh giá*: Trang web cung cấp thông tin sản phẩm chi tiết và đánh giá của khách hàng về sản phẩm, giúp người mua hàng có thể tìm hiểu và đánh giá sản phẩm trước khi quyết định mua hàng.

3.2 Nhược điểm của trang web

- *Chi phí phát triển và bảo trì*: Việc phát triển và bảo trì trang web có thể đòi hỏi chi phí khá lớn, đặc biệt là khi cần phải xây dựng và bảo trì trang web.
- *Sự cạnh tranh của thị trường*: Thị trường đang trở nên cạnh tranh hơn bao giờ hết, vì vậy, trang web cần phải cố gắng để đáp ứng nhu cầu của khách hàng và tạo ra sự khác biệt trong cách thức kinh doanh để có thể tồn tại và phát triển trong thời gian dài.

- *Sự phụ thuộc vào công nghệ*: Trang web phải luôn cập nhật và sử dụng các công nghệ mới để đáp ứng nhu cầu của khách hàng. Nếu trang web không thích nghi được với các xu hướng công nghệ mới, nó có thể bị lạc hậu và mất khách hàng.

3.3 Phân công công việc và đánh giá của thành viên

BẢNG PHÂN CÔNG CÔNG VIỆC

Thành viên	Công việc	Mức độ hoàn thành
Lý Tuấn An	<ul style="list-style-type: none"> - Thiết kế giao diện - Thiết kế model - Thực hiện chức năng: <ul style="list-style-type: none"> + Thêm sản phẩm vào giỏ hàng + Cập nhật sản phẩm + Xóa sản phẩm + Cập nhật giỏ hàng + Xóa sản phẩm ra khỏi giỏ hàng + Lịch sử đặt hàng + Quản lý tính năng sản phẩm -Tìm hiểu lý thuyết: JPA hibernate - Đặc tả usecase: Thêm sản phẩm vào giỏ hàng, cập nhật giỏ hàng, xóa sản phẩm ra khỏi giỏ hàng, tạo đơn hàng. - Vẽ sơ đồ sequence: quên mật khẩu, sửa thông tin, thêm sản phẩm vào giỏ hàng, đặt hàng. - Vẽ sơ đồ ERD 	25%
Nguyễn Thành An	<ul style="list-style-type: none"> - Thiết kế giao diện - Phân tích yêu cầu hệ thống 	25%

	<ul style="list-style-type: none"> - Thực hiện chức năng: <ul style="list-style-type: none"> + Đăng nhập, + Đăng kí + Đăng xuất + Quên mật khẩu. + Quản lý tài khoản - Viết báo cáo - Tìm hiểu lý thuyết: Bcrypt Library - Đặc tả usecase: đăng nhập, đăng kí, quên mật khẩu, đăng xuất. - Vẽ sơ đồ sequence: đăng nhập, đăng kí, đăng xuất. 	
Huỳnh Nguyễn Tường Vy	<ul style="list-style-type: none"> - Thiết kế giao diện - Thiết kế model -Thực hiện chức năng: <ul style="list-style-type: none"> + Xem chi tiết đơn hàng + Cập nhật thông tin user + Hủy đơn hàng + Tạo đơn hàng. + Xóa user. + Quản lý đơn hàng -Tìm hiểu lý thuyết: Spring Security - Đặc tả usecase: Xem chi tiết đơn hàng, xóa user, cập nhật thông tin user, hủy đơn hàng, cập nhật thông tin đơn hàng. - Vẽ sơ đồ sequence: Quản lý user, quản lý sản phẩm, quản lý giỏ hàng. 	25%

Phan Thị Diễm Thúy	<ul style="list-style-type: none"> - Thiết kế giao diện -Thực hiện chức năng: <ul style="list-style-type: none"> + Tìm kiếm + Thêm sản phẩm + Cập nhật sản phẩm + Quản lý danh mục sản phẩm - Viết báo cáo - Tìm hiểu lý thuyết: Spring MVC - Đặc tả usecase: tìm kiếm, thêm sản phẩm, cập nhật sản phẩm, xóa sản phẩm - Vẽ sơ đồ sequence: Tìm kiếm, Quản lý đơn hàng, Các chức năng CRUD 	25%
Tổng kết: 100% hoàn thành		

TÀI LIỆU THAM KHẢO

- [1]: <https://docs.spring.io/spring-framework/docs/3.2.x/spring-framework-reference/html/mvc.html>
- [2]: <https://viblo.asia/p/mot-so-khai-niem-co-ban-ve-spring-mvc-6J3Zg0PWlmB>
- [3]: <https://www.baeldung.com/learn-jpa-hibernate>

- [4]: <https://techmaster.vn/posts/36274/huong-dan-su-dung-jpa-va-hibernate-bang-spring-boot-data-jpa-phan-1>
- [5]: <https://viblo.asia/p/dung-hibernate-da-lau-the-ban-co-biet-jpa-la-gi-eW65G1zxZDO>
- [6]: <https://magz.techover.io/2023/01/02/spring-security-tim-hieu-ve-internal-flow/>
- [7]: <https://auth0.com/blog/hashing-in-action-understanding-bcrypt/>