

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP

Xây dựng hệ thống bán hàng trực tuyến dành cho các cửa hàng đồ uống

VƯƠNG NGỌC QUÂN

quan.vn166630@sis.hust.edu.vn

Ngành Công nghệ thông tin

Giảng viên hướng dẫn: ThS. Nguyễn Hồng Phương

Chữ ký của GVHD

Bộ môn: Hệ thống thông tin

Viện: Công nghệ thông tin và truyền thông

HÀ NỘI, 6/2020

ĐỀ TÀI TỐT NGHIỆP

1. Thông tin sinh viên

Họ và tên sinh viên: Vương Ngọc Quân

Điện thoại liên lạc: 0355111616

Email: quanvuongngoc98@gmail.com

Lớp: CN.CNTT2-K61

Hệ đào tạo: Cử nhân

Đề án tốt nghiệp được thực hiện tại: Trường Đại học Bách khoa Hà Nội

Thời gian làm đề án tốt nghiệp: Từ ngày 13/03/2020 đến ngày 26/06/2020

2. Mục đích nội dung của đề án tốt nghiệp

Xây dựng trang web bán hàng trực tuyến cho các cửa hàng đồ uống quy mô nhỏ sử dụng ngôn ngữ Java, giúp cho người bán dễ dàng quản lý cửa hàng, người mua thuận tiện trong việc mua hàng.

3. Các nhiệm vụ cụ thể của đề án tốt nghiệp

- Tìm hiểu các công nghệ: ngôn ngữ lập trình Java, cách thức hoạt động của website, Spring framework, cơ sở dữ liệu quan hệ SQL...
- Khảo sát nhu cầu của khách hàng, phân tích và thiết kế hệ thống
- Xây dựng hệ thống với đầy đủ các chức năng đã phân tích và thiết kế, đáp ứng được nhu cầu của khách hàng

4. Lời cam đoan của sinh viên

Tôi – Vương Ngọc Quân – cam kết đề án tốt nghiệp là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của ThS. Nguyễn Hồng Phương.

Các kết quả nêu trong đề án tốt nghiệp là trung thực, không phải sao chép toàn văn của bất kỳ công trình nào khác.

Sinh viên thực hiện

5. Xác nhận của giáo viên hướng dẫn về mức độ hoàn thành của đề án tốt nghiệp và cho phép bảo vệ

Giáo viên hướng dẫn

Lời cảm ơn

Em xin gửi lời cảm ơn đặc biệt đến thầy giáo ThS. Nguyễn Hồng Phương đã tận tình hướng dẫn em về mọi mặt trong quá trình làm đồ án tốt nghiệp. Em cũng xin được cảm ơn tất cả các thầy cô trường Đại học Bách khoa Hà Nội, đặc biệt là các thầy cô Viện Công nghệ thông tin và truyền thông đã dạy dỗ, cung cấp cho em rất nhiều kiến thức bổ ích áp dụng vào trong đồ án tốt nghiệp. Cùng với đó, em xin cảm ơn chân thành đến bố mẹ, gia đình đã tạo điều kiện tốt nhất để em có thể học tập và gắn bó với ngôi trường Bách khoa. Và cuối cùng là lời cảm ơn đến bạn gái và tất cả những người bạn trong nhóm Ocs đã tạo động lực, giúp đỡ rất nhiều để em có thể hoàn thành tốt đồ án tốt nghiệp này.

Em xin chân thành cảm ơn!

Tóm tắt nội dung đồ án

Đồ án tốt nghiệp với đề tài “Xây dựng hệ thống bán hàng trực tuyến dành cho các cửa hàng đồ uống” của em mang mục tiêu chủ chốt là tạo ra phương tiện mua bán hàng hóa trực tuyến thông qua mạng Internet dành cho các quán đồ uống với quy mô nhỏ. Phương pháp thực hiện là xây dựng một trang web có thể truy cập từ máy tính và điện thoại thông minh giúp cho người bán quản lý việc bán hàng, và người mua có thể đặt hàng trực tuyến. Công nghệ sử dụng để thực hiện đồ án là ngôn ngữ lập trình Java với sự hỗ trợ của framework Spring và dữ liệu được lưu trữ bởi cơ sở dữ liệu quan hệ SQL. Về kết quả, đồ án sau khi hoàn thành sẽ đáp ứng đầy đủ nhu cầu mua bán và thanh toán trực tuyến như yêu cầu đã đặt ra, đem lại sự tiện lợi trong công việc kinh doanh của cửa hàng. Cùng với đó, quá trình làm đồ án tốt nghiệp cũng giúp em có thêm nhiều kiến thức chuyên ngành công nghệ thông tin và các kỹ năng để độc lập hoàn thành một dự án.

MỤC LỤC

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI.....	1
1.1 Đặt vấn đề.....	1
1.2 Mục tiêu và phạm vi đề tài.....	1
1.3 Định hướng giải pháp	1
1.4 Bố cục đồ án.....	2
CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU	3
2.1 Khảo sát hiện trạng	3
2.2 Tổng quan chức năng.....	3
2.2.1 Biểu đồ use case tổng quan.....	3
2.2.2 Biểu đồ use case phân rã	4
2.3 Đặc tả chức năng	7
2.4 Yêu cầu phi chức năng.....	10
CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG.....	12
3.1 Java	12
3.2 HTTP.....	14
3.2.1 HTTP Request.....	14
3.2.2 HTTP Response	15
3.3 Apache Maven.....	15
3.4 Spring	17
3.4.1 Spring Framework.....	18
3.4.2 Spring Boot	20
3.4.3 Spring Security.....	21
3.4.4 Spring Data JPA.....	22
3.5 Thymeleaf.....	23
CHƯƠNG 4. PHÁT TRIỂN VÀ TRIỂN KHAI ỨNG DỤNG	24
4.1 Thiết kế kiến trúc	24
4.1.1 Lựa chọn kiến trúc phần mềm	24
4.1.2 Mô hình hóa các lớp thực thể	26
4.1.3 Xác định lớp tham gia từng use case.....	26
4.2 Thiết kế chi tiết.....	32
4.2.1 Thiết kế lớp.....	32
4.2.2 Biểu đồ trình tự cho từng use case	38

4.2.3	Thiết kế giao diện.....	50
4.2.4	Thiết kế cơ sở dữ liệu	54
4.3	Xây dựng ứng dụng	55
4.3.1	Thư viện và công cụ sử dụng.....	55
4.3.2	Kết quả đạt được	56
4.3.3	Minh họa chức năng.....	57
4.4	Kiểm thử.....	60
CHƯƠNG 5. KẾT LUẬN		64
5.1	Kết luận	64
5.2	Hướng phát triển của đề án trong tương lai	64
TÀI LIỆU THAM KHẢO.....		65

DANH MỤC HÌNH VẼ

Hình 2.1 Biểu đồ use case tổng quan	4
Hình 2.2 Biểu đồ use case phân rã Quản lý giỏ hàng	4
Hình 2.3 Biểu đồ use case phân rã Thay đổi thông tin	4
Hình 2.4 Biểu đồ use case phân rã Quản lý tài khoản.....	5
Hình 2.5 Biểu đồ use case phân rã Quản lý bài viết	5
Hình 2.6 Biểu đồ use case phân rã Quản lý danh mục.....	5
Hình 2.7 Biểu đồ use case phân rã Quản lý sản phẩm	6
Hình 2.8 Biểu đồ use case phân rã Quản lý khuyến mãi.....	6
Hình 2.9 Biểu đồ use case phân rã Quản lý đơn hàng	6
Hình 2.10 Biểu đồ use case phân rã Quản lý tùy chọn	7
Hình 3.1 Java Programming Language	12
Hình 3.2 JVM, JRE, JDK	13
Hình 3.3 HTTP.....	14
Hình 3.4 HTTP Request	14
Hình 3.5 HTTP Response	15
Hình 3.6 Apache Maven	15
Hình 3.7 Sử dụng pom.xml.....	17
Hình 3.8 Spring	17
Hình 3.9 Spring Framework	19
Hình 3.10 Spring Initializr	20
Hình 3.11 Spring Security	21
Hình 3.12 Spring Data	22
Hình 3.13 Thymeleaf.....	23
Hình 4.1 Mô hình MVC	24
Hình 4.2 Mô hình hóa các lớp thực thể	26
Hình 4.3 Lớp tham gia use case Đăng ký	26
Hình 4.4 Lớp tham gia use case Tìm kiếm sản phẩm	27
Hình 4.5 Lớp tham gia use case Xem bài viết	27
Hình 4.6 Lớp tham gia use case Đăng nhập	27
Hình 4.7 Lớp tham gia use case Đăng xuất.....	28
Hình 4.8 Lớp tham gia use case Nhận thông báo đơn hàng	28
Hình 4.9 Lớp tham gia use case Đổi mật khẩu	28
Hình 4.10 Lớp tham gia use case Lấy lại mật khẩu	28
Hình 4.11 Lớp tham gia use case Thay đổi thông tin	29
Hình 4.12 Lớp tham gia use case Quản lý giỏ hàng	29
Hình 4.13 Lớp tham gia use case Tạo đơn hàng.....	29

Hình 4.14 Lớp tham gia use case Thanh toán đơn hàng	30
Hình 4.15 Lớp tham gia use case Xem lịch sử mua hàng	30
Hình 4.16 Lớp tham gia use case Chat với quản trị viên	30
Hình 4.17 Lớp tham gia use case Chat với người dùng	30
Hình 4.18 Lớp tham gia use case Cài đặt hệ thống.....	30
Hình 4.19 Lớp tham gia use case Quản lý tài khoản	31
Hình 4.20 Lớp tham gia use case Quản lý bài viết	31
Hình 4.21 Lớp tham gia use case Quản lý danh mục.....	31
Hình 4.22 Lớp tham gia use case Quản lý sản phẩm	31
Hình 4.23 Lớp tham gia use case Quản lý tùy chọn	32
Hình 4.24 Lớp tham gia use case Quản lý khuyến mãi.....	32
Hình 4.25 Lớp tham gia use case Quản lý đơn hàng	32
Hình 4.26 Biểu đồ trình tự use case Đăng ký	38
Hình 4.27 Biểu đồ trình tự use case Tìm kiếm sản phẩm	38
Hình 4.28 Biểu đồ trình tự use case Xem bài viết	39
Hình 4.29 Biểu đồ trình tự use case Đăng nhập	39
Hình 4.30 Biểu đồ trình tự use case Đăng xuất	40
Hình 4.31 Biểu đồ trình tự use case Nhận thông báo đơn hàng	40
Hình 4.32 Biểu đồ trình tự use case Đổi mật khẩu	40
Hình 4.33 Biểu đồ trình tự use case Lấy lại mật khẩu	41
Hình 4.34 Biểu đồ trình tự use case Sửa thông tin cá nhân.....	41
Hình 4.35 Biểu đồ trình tự use case Thêm địa chỉ nhận hàng.....	42
Hình 4.36 Biểu đồ trình tự use case Chat với quản trị viên	42
Hình 4.37 Biểu đồ trình tự use case Xem lịch sử mua hàng	43
Hình 4.38 Biểu đồ trình tự use case Thêm sản phẩm vào giỏ	43
Hình 4.39 Biểu đồ trình tự use case Tạo đơn hàng.....	44
Hình 4.40 Biểu đồ trình tự use case Thanh toán đơn hàng	44
Hình 4.41 Biểu đồ trình tự use case Cài đặt hệ thống.....	45
Hình 4.42 Biểu đồ trình tự use case Chat với người dùng	46
Hình 4.43 Biểu đồ trình tự use case Xem thông tin tài khoản.....	46
Hình 4.44 Biểu đồ trình tự use case Thêm bài viết.....	47
Hình 4.45 Biểu đồ trình tự use case Thêm danh mục	47
Hình 4.46 Biểu đồ trình tự use case Thêm sản phẩm.....	48
Hình 4.47 Biểu đồ trình tự use case Thêm tùy chọn.....	48
Hình 4.48 Biểu đồ trình tự use case Thêm lựa chọn.....	49
Hình 4.49 Biểu đồ trình tự use case Thêm khuyến mãi	49
Hình 4.50 Biểu đồ trình tự use case Thêm đơn hàng.....	50

Hình 4.51 Thiết kế giao diện (màn hình máy tính).....	51
Hình 4.52 Thiết kế giao diện (màn hình điện thoại)	52
Hình 4.53 Thiết kế giao diện (vị trí hộp thoại)	53
Hình 4.54 Thiết kế giao diện (vị trí thông điệp)	53
Hình 4.55 Thiết kế giao diện (Datatable)	53
Hình 4.56 Thiết kế giao diện (Bootstrap Modal).....	54
Hình 4.57 Thiết kế cơ sở dữ liệu.....	54
Hình 4.58 Cấu trúc thư mục dự án	56
Hình 4.59 Cấu trúc thư mục src	57
Hình 4.60 Chức năng Đăng nhập.....	58
Hình 4.61 Chức năng Tìm kiếm sản phẩm.....	58
Hình 4.62 Chức năng Chat với quản trị viên.....	59
Hình 4.63 Chức năng Thanh toán	59
Hình 4.64 Chức năng Thêm bài viết	60
Hình 4.65 Chức năng Quản lý tùy chọn.....	60
Hình 4.66 Kiểm thử đăng nhập thất bại	61
Hình 4.67 Kiểm thử phân quyền	61
Hình 4.68 Kiểm thử đăng ký thất bại	62
Hình 4.69 Kiểm thử Thêm địa chỉ nhận hàng vượt mức.....	62
Hình 4.70 Kiểm thử Cập nhật thông tin sai định dạng ảnh	63
Hình 4.71 Kiểm thử Tạo đơn hàng trống	63

DANH MỤC BẢNG BIỂU

Bảng 2.1 Đặc tả use case Đăng nhập	7
Bảng 2.2 Đặc tả use case Lấy lại mật khẩu	7
Bảng 2.3 Đặc tả use case Quản lý giỏ hàng/Thêm sản phẩm.....	8
Bảng 2.4 Đặc tả use case Thanh toán đơn hàng	9
Bảng 2.5 Đặc tả use case Quản lý đơn hàng/Cập nhật trạng thái	9
Bảng 2.6 Đặc tả use case Chat với người dùng	10
Bảng 4.1 Thiết kế lớp NotificationController	32
Bảng 4.2 Thiết kế lớp AccountController	33
Bảng 4.3 Thiết kế lớp ChatController	35
Bảng 4.4 Thiết kế lớp AccountManagementController	36
Bảng 4.5 Thiết kế lớp OptionManagementController	36
Bảng 4.6 Thiết kế lớp SettingController	37
Bảng 4.7 Thư viện và công cụ sử dụng	55

DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT

ĐATN	Đồ án tốt nghiệp
MVC	Model – View – Controller
JVM	Java Virtual Machine
JIT	Just In Time Compiler
JRE	Java Runtime Environment
JDK	Java Development Kit
HTTP	Hypertext Transfer Protocol
HTML	Hypertext Markup Language [1]
XML	Extensible Markup Language
CSS	Cascading Style Sheets [2]
ES6	ECMAScript 2015 [3]
JSP	Java Server Pages
POM	Project Object Model
IDE	Integrated Development Environment
API	Application Programming Interface
JDBC	Java Database Connectivity
ORM	Object Relational Mapping
OXM	Object XML Mapping
JMS	Java Message Service
POJO	Plain Old Java Object
AOP	Aspect Oriented Programming
CSRF	Cross Site Request Forgery
JPA	Java Persistent API
CRUD	Create, Read, Update, Delete
UI	User Interface

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1 Đặt vấn đề

Xuất phát từ tình hình thực tế Internet đã được phổ biến rộng rãi khắp thế giới cũng như toàn lãnh thổ Việt Nam, cùng với đó là sự hiện diện của điện thoại thông minh, máy tính bảng, máy tính xách tay ngày một nhiều lên, thương mại điện tử, mua bán online xuất hiện và trở thành một cách thức kinh doanh được sử dụng ở khắp mọi nơi, mọi quy mô kinh doanh, chủ thể kinh doanh. Từ tình hình thực tế đó, vấn đề được đặt ra là mỗi cửa hàng kinh doanh đều cần triển khai cho mình mô hình kinh doanh trực tuyến, nhằm bắt kịp với yêu cầu của thị trường và cải thiện hiệu suất kinh doanh. Với đề tài là xây dựng hệ thống kinh doanh đồ uống trực tuyến, hệ thống không hề gây ảnh hưởng đến công việc kinh doanh đồ uống truyền thống, mà nó song hành, chia sẻ đáng kể lượng công việc và thời gian mà mỗi nhân viên bán hàng, nhân viên phục vụ, nhân viên thanh toán và chủ cửa hàng phải thực hiện. Ngoài ra, người mua cũng là những đối tượng được hưởng sự tiện lợi, minh bạch trong quá trình mua hàng, dẫn đến những quyết định mở hầu bao một cách chính xác và nhanh gọn hơn.

1.2 Mục tiêu và phạm vi đề tài

Qua trải nghiệm thực tế của em ở một số cửa hàng cà phê, giải khát, đồ uống tại nơi đang sinh sống, có thể thấy được rằng, các cửa hàng quy mô trung bình nhỏ như vậy (chỉ có 1 cơ sở tại 1 địa phương duy nhất) cũng đã áp dụng công nghệ thông tin vào trong nghiệp vụ bán hàng. Tuy nhiên, đó chỉ là những phần mềm, công cụ hỗ trợ quản lý bàn uống, tính toán hóa đơn, in hóa đơn trong nghiệp vụ bán hàng tại chỗ. Tiếp theo, hiện nay trên thị trường đã tồn tại những nền tảng kinh doanh đồ ăn trực tuyến rất nổi tiếng, có thể kể đến là Now.vn, đây là những hệ thống đã làm rất tốt trong ngành bán đồ ăn uống trực tuyến, nhưng nó có một số hạn chế đó là triển khai rất mạnh mẽ ở những khu vực phát triển, trung tâm thành phố nhưng chưa dành cho những cửa hàng có vị trí địa lý xa nội thành. Như vậy có thể thấy rõ rằng, nhu cầu của người dùng đối với sản phẩm (trang web bán hàng) trong đề tài đồ án tốt nghiệp là vẫn còn và chưa được đáp ứng, và khả năng áp dụng trong thực tiễn của đề tài là rất cao. Từ đó, em đã đặt ra mục tiêu và phạm vi cho đề tài đồ án tốt nghiệp của mình, đó là tạo ra một phần mềm với nhiệm vụ tối thiểu là giúp cho các cửa hàng như trên có thể triển khai mô hình bán hàng online cho riêng mình, mang bản sắc, nghiệp vụ riêng của cửa hàng khi không muốn, hoặc, không được hỗ trợ bởi những nền tảng chung như đã đề cập tới phía trên.

1.3 Định hướng giải pháp

Theo như mục tiêu của đề tài đồ án tốt nghiệp đã nêu ở mục 1.2, chắc chắn định hướng đưa ra để giải quyết vấn đề đó là tạo ra một trang web, thứ mà có thể truy cập và sử dụng ở khắp mọi nơi, miễn là nơi đó có Internet. Giải pháp và công nghệ được em lựa chọn trong việc thực hiện đồ án tốt nghiệp lần này là sử dụng ngôn ngữ lập trình hướng đối tượng Java – ngôn ngữ lâu đời, mạnh mẽ, có cộng đồng lập trình viên lớn bậc nhất thế giới, với sự hỗ trợ của Spring

framework, một “hệ sinh thái” gồm nhiều dự án dành cho ngôn ngữ java, như cách nó tự mô tả: “Spring makes Java simple, modern, productive, reactive, cloud-ready”. Với mong muốn tạo ra sản phẩm cuối cùng là 1 dự án web mang mô hình MVC (Model – View – Controller).

1.4 Bố cục đồ án

Phần còn lại của báo cáo đồ án tốt nghiệp này được tổ chức như sau.

Chương 2: Khảo sát và phân tích yêu cầu là tài liệu trình bày về 2 giai đoạn đầu tiên của quy trình phát triển phần mềm, đó là Requirement Gathering và Analysis. Mục 1, khảo sát hiện trạng đưa ra kết quả khảo sát nhu cầu, mong muốn của khách hàng (chủ của các cửa hàng kinh doanh) về các chức năng cần có của sản phẩm. Từ đó mục 2 vẽ lên biểu đồ use case tổng quan về các chức năng đó, đồng thời phân ra các use case phức tạp. Mục 3 sẽ bao gồm các đặc tả làm rõ cho từng use case và cuối cùng là các yêu cầu phi chức năng của hệ thống như hiệu năng, giao diện người dùng, trải nghiệm người dùng...

Chương 3: Công nghệ sử dụng là nơi giới thiệu chi tiết và kỹ càng về các công nghệ, nền tảng hoặc cơ sở lý thuyết sử dụng trong đồ án tốt nghiệp và cách thức sử dụng các công nghệ/nền tảng/ lý thuyết đó để giải quyết các vấn đề và yêu cầu cụ thể tồn tại trong đồ án.

Chương 4: Phát triển và triển khai ứng dụng sẽ bao gồm 4 giai đoạn tiếp theo trong quy trình phát triển phần mềm, đó là Design, Coding, Testing và Implementation. Mục thiết kế kiến trúc làm rõ kiến trúc lựa chọn, đưa ra cái nhìn tổng quan về toàn bộ dự án. Tiếp theo là mục thiết kế chi tiết sẽ đi sâu vào phân tích từng phần, từng tầng của mô hình kiến trúc đó. Và 3 mục cuối cùng chính là tài liệu về cách thức xây dựng ứng dụng dựa trên bản thiết kế, kiểm thử các trường hợp gây ra lỗi, và triển khai ứng dụng sau khi đã hoàn thành.

Chương 5: Kết luận gồm 2 nội dung chính, đó là mục đưa ra kết luận về sản phẩm mà đồ án tốt nghiệp của mình đã xây dựng, và mục hướng phát triển trình bày các công việc cần thiết để hoàn thiện, cải thiện chức năng đã làm, hướng nâng cấp các chức đó và mở rộng thêm các chức năng mới.

CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

2.1 Khảo sát hiện trạng

Thông qua việc khảo sát hiện trạng về các sản phẩm tương tự, và đặc biệt quan trọng nhất là việc thu thập yêu cầu từ khách hàng, trang web của em được mong đợi sẽ có các chức năng chính được mô tả như sau:

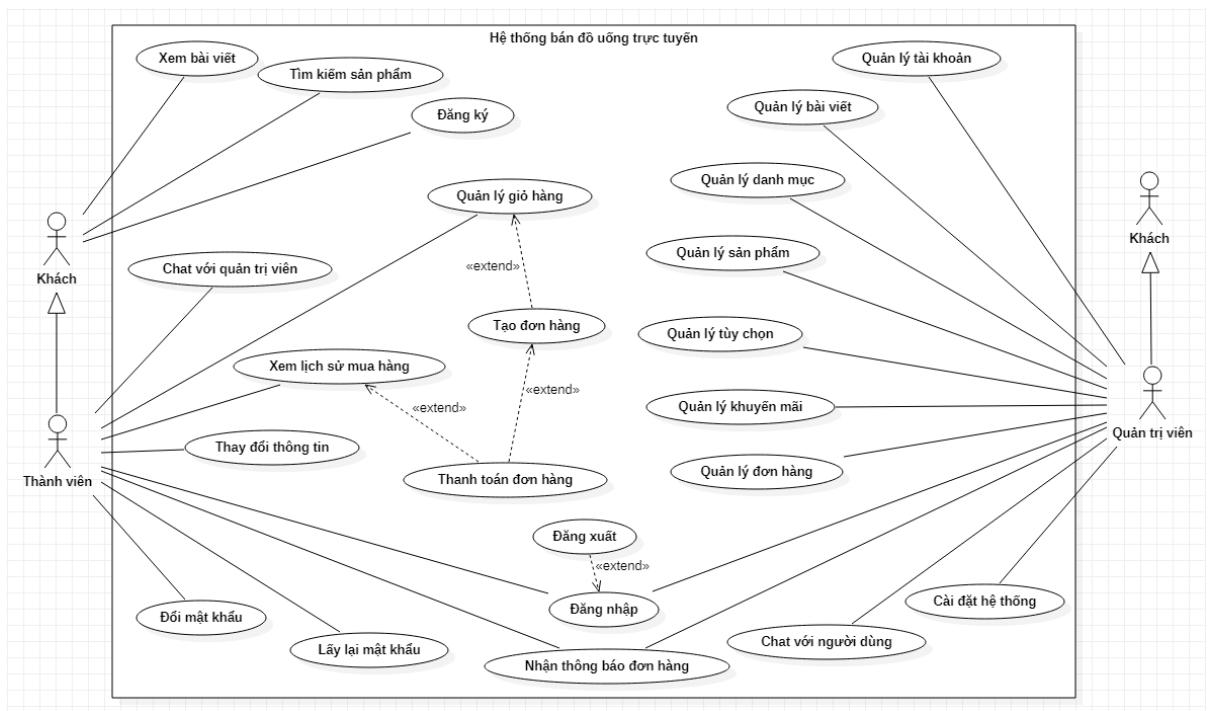
- Authentication & Authorization (Xác thực và phân quyền): Sản phẩm tạo ra cho phép người dùng đăng ký tài khoản hệ thống, đăng nhập bằng tài khoản đã tạo, đăng xuất, và chức năng quên mật khẩu
- Quản lý thông tin tài khoản: Cho phép người dùng đổi mật khẩu, thay đổi thông tin tài khoản, thông tin địa chỉ nhận hàng
- Viết bài tin tức: Người dùng được phép xem những bài viết mà quản trị viên đã đăng lên hệ thống
- Quản lý các chương trình khuyến mãi: Áp dụng các chương trình khuyến mãi, giảm giá sản phẩm theo cài đặt của quản trị viên
- Tìm kiếm đồ uống: Người dùng có thể tìm kiếm sản phẩm muốn mua qua bộ lọc sản phẩm
- Chức năng giỏ hàng: Thêm sửa xóa các sản phẩm trong giỏ hàng
- Mua hàng trực tuyến: Đặt đơn hàng từ các sản phẩm đã có trong giỏ hàng
- Thanh toán trực tuyến: Thanh toán trực tuyến thông qua thẻ ngân hàng cho đơn hàng đã mua
- Xem lịch sử và trạng thái đơn hàng: Người dùng được thông báo mỗi khi đơn hàng của họ được thay đổi trạng thái, có thể xem lại các đơn hàng mình đã tạo trước đó
- Chat thời gian thực: Sản phẩm cho phép trò chuyện trực tuyến thời gian thực giữa người mua và người bán

2.2 Tổng quan chức năng

2.2.1 Biểu đồ use case tổng quan

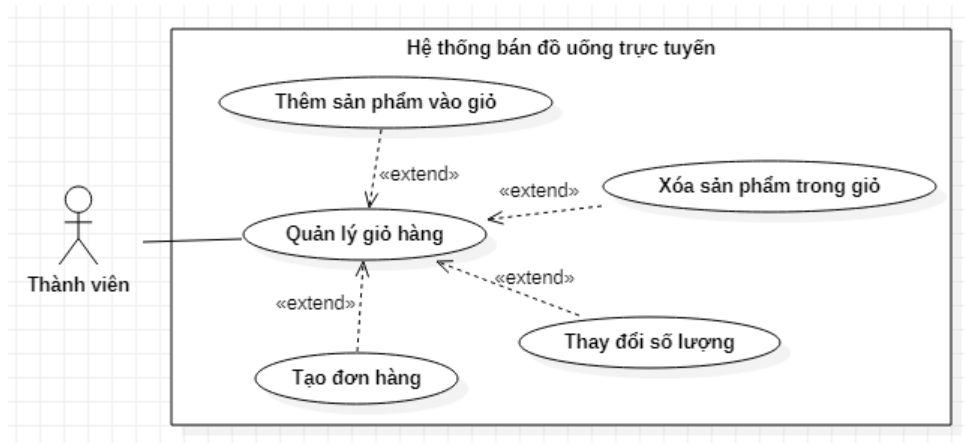
Dựa vào tài liệu khảo sát chức năng sản phẩm bên trên, em xác định hệ thống gồm có 3 tác nhân chính: Guest (Khách), Member (Thành viên), Admin (Quản trị viên)

- Khách: những cá nhân chưa có tài khoản trên hệ thống
- Thành viên: những cá nhân đã có tài khoản trên hệ thống với quyền USER
- Quản trị viên: cá nhân đã có tài khoản trên hệ thống với quyền ADMIN, có khả năng đăng nhập và quản lý toàn bộ hệ thống

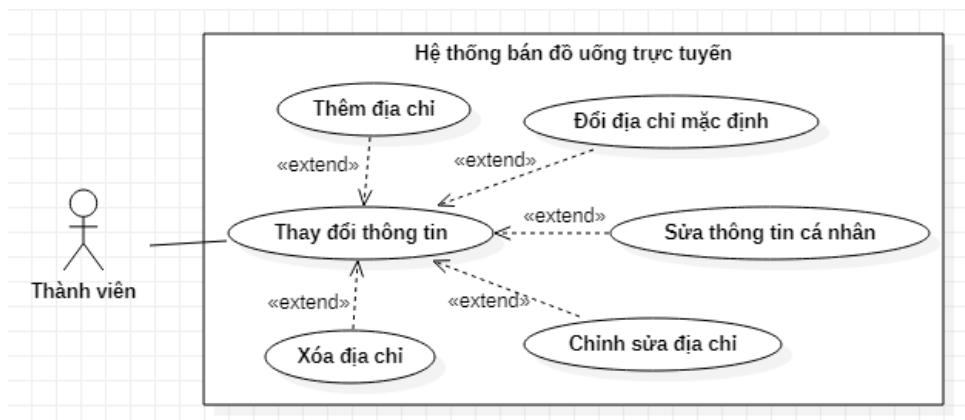


Hình 2.1 Biểu đồ use case tổng quan

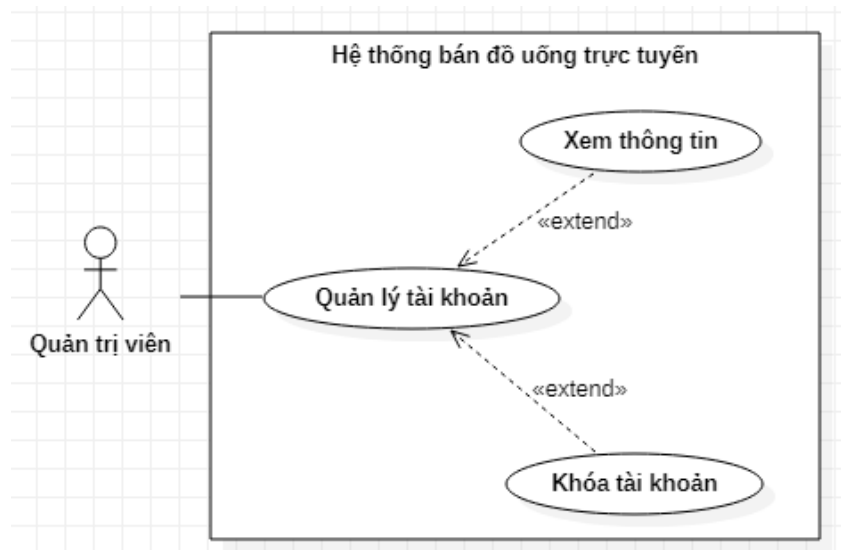
2.2.2 Biểu đồ use case phân rã



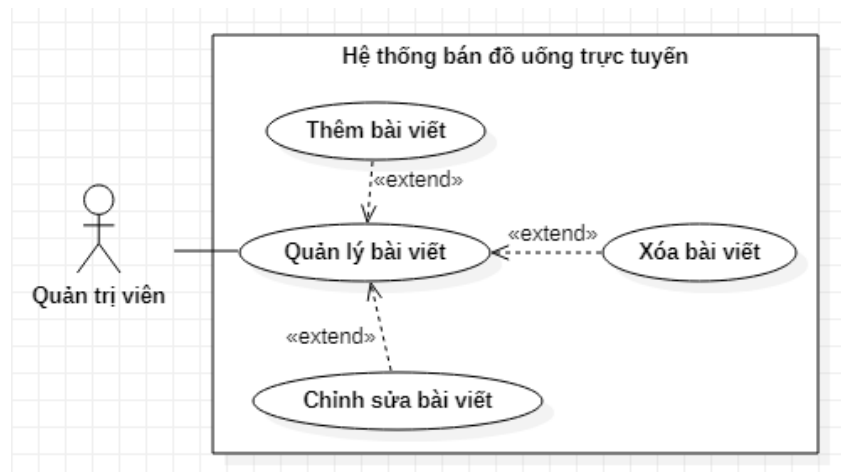
Hình 2.2 Biểu đồ use case phân rã Quản lý giỏ hàng



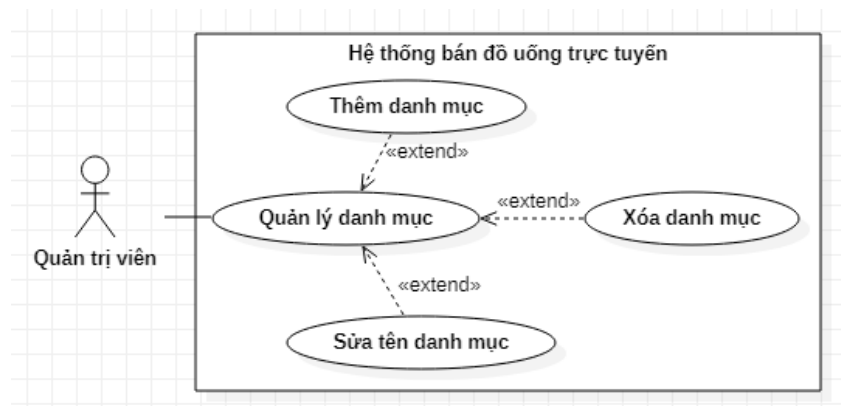
Hình 2.3 Biểu đồ use case phân rã Thay đổi thông tin



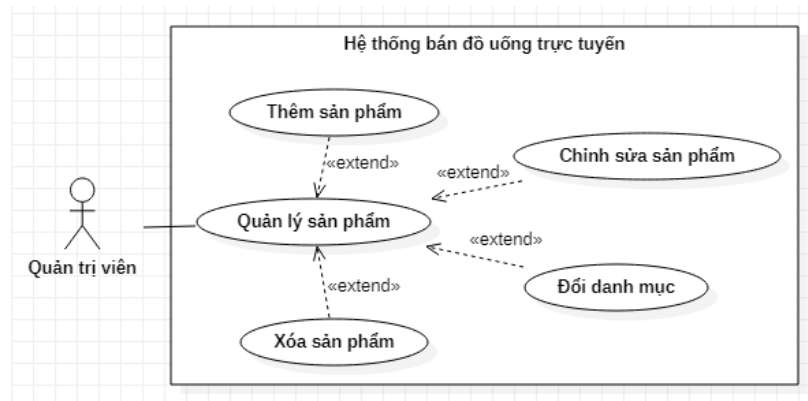
Hình 2.4 Biểu đồ use case phân rã Quản lý tài khoản



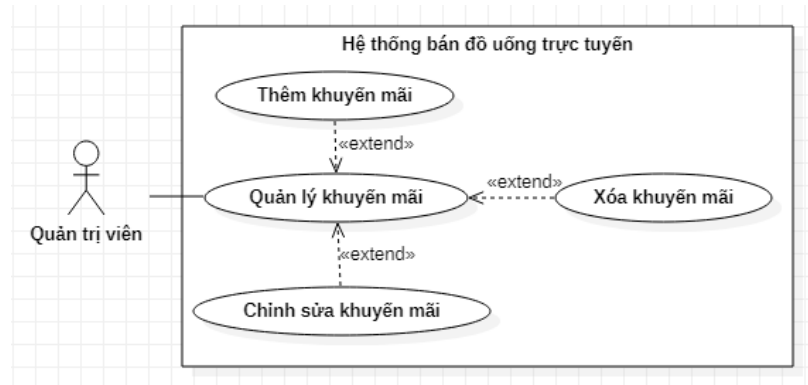
Hình 2.5 Biểu đồ use case phân rã Quản lý bài viết



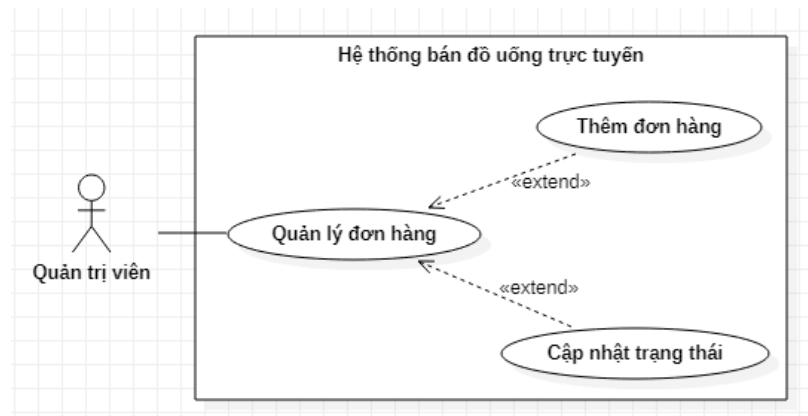
Hình 2.6 Biểu đồ use case phân rã Quản lý danh mục



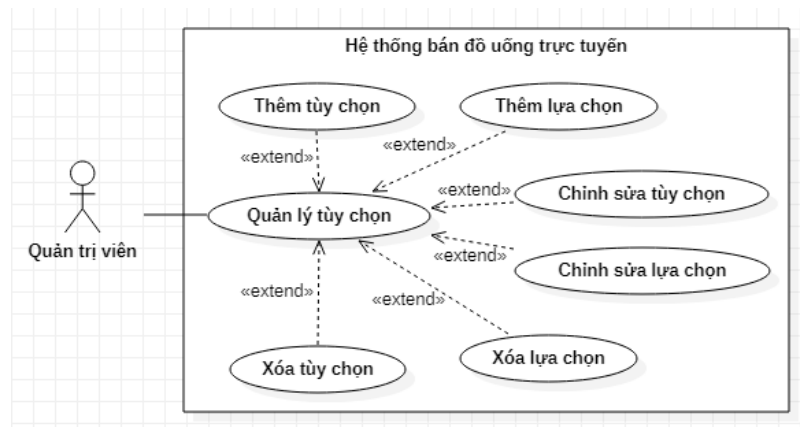
Hình 2.7 Biểu đồ use case phân rã Quản lý sản phẩm



Hình 2.8 Biểu đồ use case phân rã Quản lý khuyến mãi



Hình 2.9 Biểu đồ use case phân rã Quản lý đơn hàng



Hình 2.10 Biểu đồ use case phân rã Quản lý tùy chọn

2.3 Đặc tả chức năng

Sau đây là đặc tả của một số use case quan trọng nhất trong hệ thống:

Bảng 2.1 Đặc tả use case Đăng nhập

Tên use case	Đăng nhập
Mô tả	Cho phép người dùng đăng nhập vào hệ thống
Tác nhân	Thành viên và Quản trị viên
Tiền điều kiện	Người dùng đã có tài khoản trên hệ thống
Hậu điều kiện	Hệ thống đưa người dùng đến trang chủ đã đăng nhập
Luồng sự kiện	<ul style="list-style-type: none"> • Người dùng truy cập trang chủ • Hệ thống đưa ra giao diện khi chưa đăng nhập • Người dùng chọn “Đăng nhập” • Hệ thống đưa ra giao diện đăng nhập • Người dùng nhập tài khoản và mật khẩu • Hệ thống kiểm tra tính hợp lệ của tài khoản và mật khẩu • Hệ thống đưa ra giao diện khi đã đăng nhập thành công

Bảng 2.2 Đặc tả use case Lấy lại mật khẩu

Tên use case	Lấy lại mật khẩu
Mô tả	Cho phép người dùng lấy lại mật khẩu khi vô tình quên
Tác nhân	Thành viên

Tiền điều kiện	Người dùng đã có tài khoản trên hệ thống
Hậu điều kiện	Hệ thống đổi mật khẩu mới cho tài khoản người dùng
Luồng sự kiện	<ul style="list-style-type: none"> • Người dùng truy cập trang chủ • Hệ thống đưa ra giao diện chưa đăng nhập • Người dùng chọn “Đăng nhập” • Hệ thống đưa ra giao diện đăng nhập • Người dùng chọn “Quên mật khẩu” • Hệ thống đưa ra giao diện quên mật khẩu • Người dùng nhập email và nộp • Hệ thống gửi link lấy lại mật khẩu vào email đã nhập • Người dùng truy cập đường link trong email • Hệ thống đưa ra giao diện lấy lại mật khẩu • Người dùng nhập mật khẩu mới và nhập lại mật khẩu • Hệ thống kiểm tra trùng khớp giữa 2 mật khẩu • Hệ thống đổi mật khẩu mới và thông báo kết quả thành công

Bảng 2.3 Đặc tả use case Quản lý giỏ hàng/Thêm sản phẩm

Tên use case	Quản lý giỏ hàng/Thêm sản phẩm
Mô tả	Cho phép người dùng thêm sản phẩm muốn mua vào giỏ hàng
Tác nhân	Thành viên
Tiền điều kiện	Người dùng có tài khoản trên hệ thống, đã đăng nhập thành công và tìm kiếm được sản phẩm muốn mua
Hậu điều kiện	Sản phẩm được thêm vào giỏ hàng của người dùng
Luồng sự kiện	<ul style="list-style-type: none"> • Người dùng chọn “Thêm vào giỏ” • Hệ thống đưa ra giao diện để chọn số lượng và các tùy chọn thêm dành cho sản phẩm • Người dùng nhập số lượng và các tùy chọn

	<ul style="list-style-type: none"> • Hệ thống thực hiện kiểm tra nếu sản phẩm đã tồn tại trong giỏ hàng • Hệ thống thêm chi tiết sản phẩm vào trong giỏ hàng của người dùng và thông báo thành công
--	---

Bảng 2.4 Đặc tả use case Thanh toán đơn hàng

Tên use case	Thanh toán đơn hàng
Mô tả	Cho phép người dùng thanh toán trực tuyến cho đơn hàng đã đặt
Tác nhân	Thành viên
Tiền điều kiện	Người dùng đăng nhập thành công, đã tạo thành công đơn hàng trên hệ thống
Hậu điều kiện	Hệ thống cập nhật trạng thái đơn hàng đã thanh toán
Luồng sự kiện	<ul style="list-style-type: none"> • Người dùng vừa mới tạo thành công đơn hàng • Hệ thống đưa ra giao diện thanh toán • Người dùng nhập thông tin và yêu cầu thanh toán • Hệ thống thu thập thông tin và yêu cầu lên Stripe API • Hệ thống kiểm tra mã phản hồi từ Stripe API là hợp lệ • Hệ thống thực hiện cập nhật trạng thái đơn hàng và đưa ra thông báo

Bảng 2.5 Đặc tả use case Quản lý đơn hàng/Cập nhật trạng thái

Tên use case	Quản lý đơn hàng/Cập nhật trạng thái
Mô tả	Cho phép quản trị viên cập nhật trạng thái của đơn hàng trên hệ thống
Tác nhân	Quản trị viên
Tiền điều kiện	Quản trị viên đăng nhập thành công
Hậu điều kiện	Hệ thống cập nhật trạng thái đơn hàng
Luồng sự kiện	<ul style="list-style-type: none"> • Quản trị viên truy cập trang “Quản lý đơn hàng” • Hệ thống đưa ra giao diện danh sách các đơn hàng và trạng thái của từng đơn • Quản trị viên thực hiện chọn

	trạng thái cho đơn hàng muốn cập nhật <ul style="list-style-type: none"> Hệ thống thay đổi trạng thái của đơn hàng và thông báo cho người dùng sở hữu đơn hàng đó
--	--

Bảng 2.6 Đặc tả use case Chat với người dùng

Tên use case	Chat với người dùng
Mô tả	Cho phép quản trị viên trò chuyện trực tiếp với người dùng
Tác nhân	Quản trị viên
Tiền điều kiện	Quản trị viên đăng nhập thành công
Hậu điều kiện	Hệ thống gửi tin nhắn thành công tới người nhận
Luồng sự kiện	<ul style="list-style-type: none"> Quản trị viên truy cập trang “Quản lý tin nhắn” Hệ thống đưa ra giao diện danh sách các hộp thoại với người dùng Quản trị viên thực hiện chọn hộp thoại của người dùng mà muốn gửi tin nhắn tới họ Hệ thống đưa ra giao diện lịch sử trò chuyện giữa 2 người Quản trị viên nhập tin nhắn và gửi đi Hệ thống lưu lại tin nhắn vào cơ sở dữ liệu và thực hiện thông báo ngay lập tức đến người nhận

2.4 Yêu cầu phi chức năng

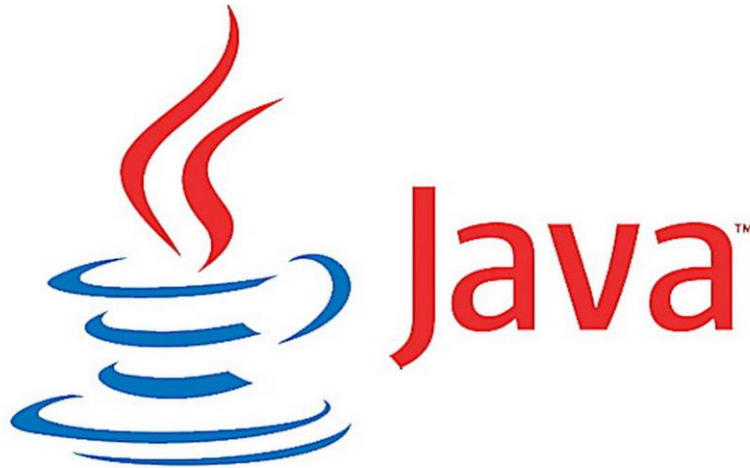
Ngoài những yêu cầu về các chức năng chính của sản phẩm đã được khảo sát, nêu ra và phân tích phía trên, hệ thống còn cần phải đáp ứng các yêu cầu phi chức năng như sau:

- Hiệu năng: thời gian phản hồi của trang web được yêu cầu không vượt quá 3 giây, đạt trung bình 1 giây
- Tính bảo mật: đảm bảo chức năng xác thực và phân quyền hoạt động tốt, không gặp lỗi bảo mật
- Tính dễ dùng: phần mềm phải bố trí các điều khiển, các phím chức năng ở nơi thuận tiện, dễ phát hiện, dễ thao tác
- Giao diện: đẹp, bố cục rõ ràng, mang màu sắc sáng chủ đạo, căn chỉnh thẳng đều các khoảng cách, đối với các trang dành cho Khách và Người dùng, cần hiển thị tốt cả trên điện thoại thông minh

- Trải nghiệm người dùng: tất cả các hành động sau khi thực hiện đều phải thông báo đã thành công hoặc thất bại, sử dụng bảng để hiển thị danh sách dữ liệu
- Yêu cầu khác: sản phẩm không vi phạm pháp luật, không vi phạm bản quyền, không dùng các công cụ hỗ trợ không rõ nguồn gốc và không đảm bảo tính an toàn

CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG

3.1 Java



Hình 3.1 Java Programming Language

Java [4] là một ngôn ngữ lập trình hướng đối tượng. Khác với các ngôn ngữ lập trình khác, mã nguồn java không được biên dịch trực tiếp thành mã máy hoặc thông dịch khi chạy, mà nó được biên dịch thành bytecode và bytecode sau đó được thực thi bởi máy ảo java JVM biến java thành ngôn ngữ lập trình viết một lần chạy mọi nơi (Write Once, Run Anywhere)

5 nguyên tắc tạo ra ngôn ngữ java:

- Đơn giản, hướng đối tượng và quen thuộc
- Mạnh mẽ và bảo mật
- Kiến trúc trung lập và di động
- Hiệu suất cao
- Phân luồng và khả biến

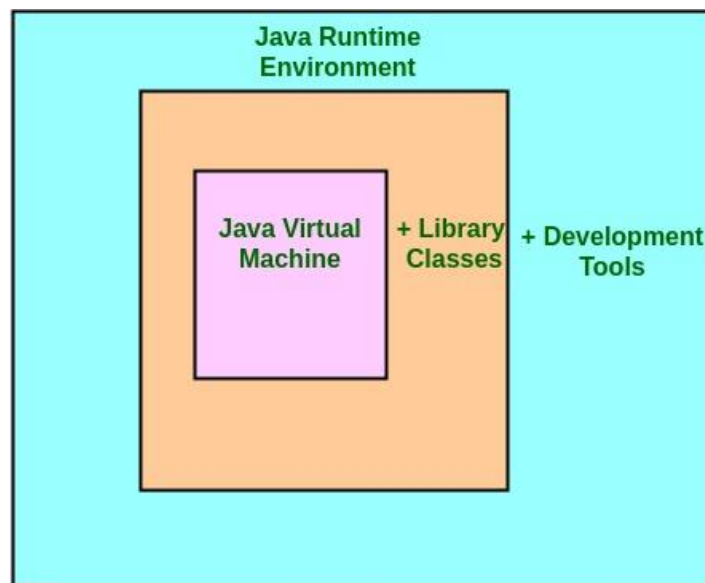
Các phiên bản java:

- JDK 1.0 (January 23, 1996)
- JDK 1.1 (February 19, 1997)
- J2SE 1.2 (December 8, 1998)
- J2SE 1.3 (May 8, 2000)
- J2SE 1.4 (February 6, 2002)
- J2SE 5.0 (September 30, 2004)
- Java SE 6 (December 11, 2006)
- Java SE 7 (July 28, 2011)
- Java SE 8 (March 18, 2014)
- Java SE 9 (September 21, 2017)
- Java SE 10 (March 20, 2018)
- Java SE 11 (September 25, 2018)
- Java SE 12 (March 19, 2019)

- Java SE 13 (September 17, 2019)
- Java SE 14 (March 17, 2020)

Một số khái niệm cốt lõi:

- JIT Compiler: Just-In-Time Compiler – công nghệ biên dịch tại chỗ, với JIT compiler, mã nguồn không còn được biên dịch toàn bộ trước khi chương trình được chạy mà quá trình biên dịch vẫn tiếp tục hoạt động trong thời gian chạy chương trình, tạo ra những điều chỉnh làm tăng hiệu suất thực thi.
- JVM: Java Virtual Machine – máy ảo java, trái tim của ngôn ngữ lập trình java giúp nó trở thành ngôn ngữ write once run anywhere. Trong quá trình chạy chương trình java, máy ảo java có trách nhiệm biến mã bytecode thành mã máy và thực thi chúng mà không phụ thuộc vào bất kỳ hệ điều hành hay phần cứng.
- JRE: Java Runtime Environment – môi trường thực thi java, cung cấp những yêu cầu cần thiết để chạy một chương trình java, nó bao gồm JVM, các class cơ bản và các file hỗ trợ.
- JDK: Java Development Kit – bộ công cụ phát triển java, cung cấp những yêu cầu cần thiết để phát triển ứng dụng java, nó bao gồm JRE, Java, Javac, Jar, Javadoc và nhiều công cụ phát triển khác.



JDK = JRE + Development Tool
JRE = JVM + Library Classes

Hình 3.2 JVM, JRE, JDK

Những tính năng mới nổi bật có trong phiên bản java 8:

- Lambda Expressions
- Stream API
- Interface Default Method
- Optional Class

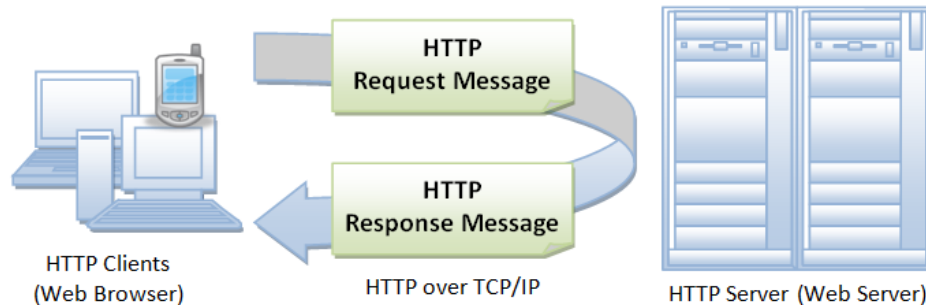
- Standard Encoding and Decoding Base64

3.2 HTTP

HTTP [5] là một giao thức thuộc tầng ứng dụng, nằm trên giao thức tầng giao vận và tầng mạng là TCP/IP, được dùng để truyền tải các tài liệu đa phương tiện như HTML, CSS. Nó được thiết kế để truyền thông giữa các trình duyệt web và máy chủ web. HTTP tuân theo mô hình client – server truyền thống, với client mở kết nối để tạo ra các yêu cầu và chờ nhận phản hồi từ server.

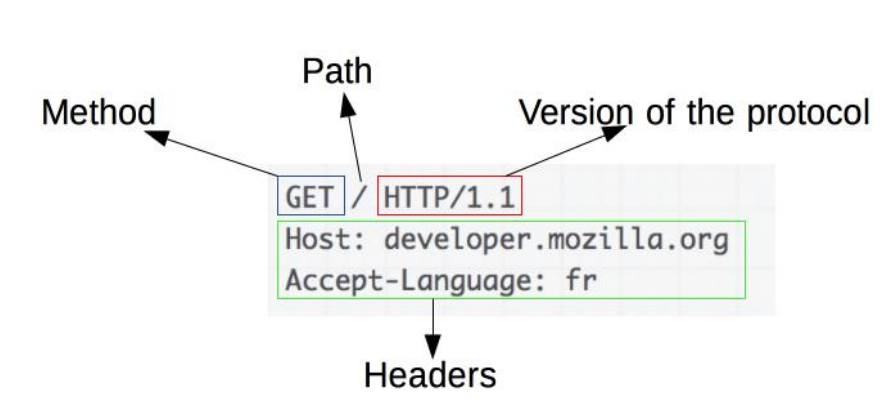
Các phiên bản HTTP:

- HTTP 0.9 (1991)
- HTTP 1.0 (1996)
- HTTP 1.1 (1997)
- HTTP 2.0 (2015)
- HTTP 3.0 (2018)



Hình 3.3 HTTP

3.2.1 HTTP Request



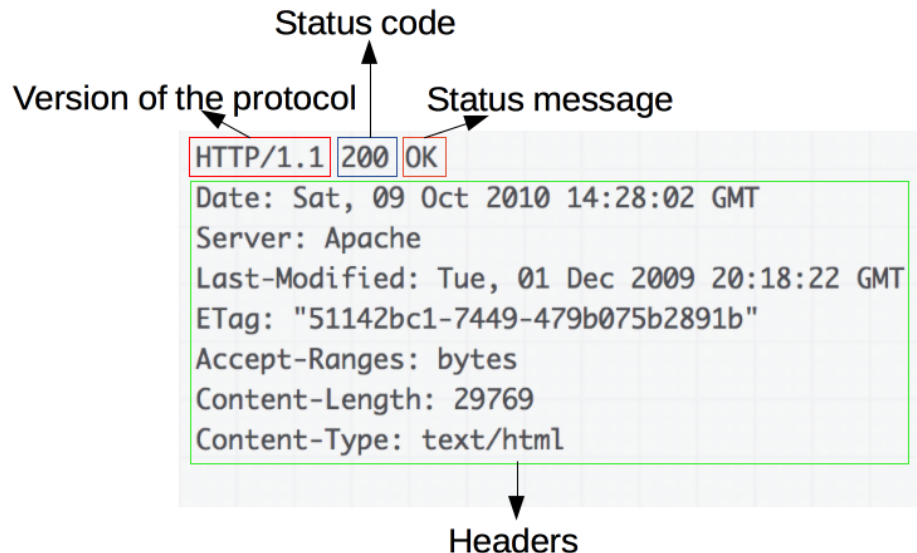
Hình 3.4 HTTP Request

HTTP Request bao gồm:

- Request Line: gồm 3 phần ngăn cách nhau bởi dấu cách: HTTP Method, Path và Protocol version. HTTP Method là một trong các phương thức sau: GET, HEAD, POST, PUT, DELETE, TRACE, OPTIONS, CONNECT, PATCH.

- Request Headers: chứa các thông tin bổ sung
- Empty Line: đánh dấu sự kết thúc của phần request headers
- Request Body: chứa phần thông tin muốn truyền tải qua request, thường được dùng trong phương thức POST

3.2.2 HTTP Response



Hình 3.5 HTTP Response

HTTP Response bao gồm:

- Status Line: gồm phiên bản của giao thức, mã trạng thái và thông điệp trạng thái. Các mã trạng thái được chia ra làm 5 nhóm với các ý nghĩa khác nhau: 1XX – Informational, 2XX – Successful, 3XX – Redirection, 4XX – Client Error, 5XX – Server Error.
- Response Headers: chứa các thông tin bổ sung
- Empty Line: đánh dấu sự kết thúc của phần response headers
- Response Body: chứa các tài nguyên muốn trả về cho client thông qua response

3.3 Apache Maven



Hình 3.6 Apache Maven

Apache Maven là một công cụ quản lý và thiết lập dự án phần mềm. Maven dùng khái niệm Project Object Model (POM) để mô tả việc build project,

các thành phần phụ thuộc và các module. Nó định nghĩa trước các target cho việc khai báo task, trình biên dịch, đóng gói và thứ tự hoạt động để mọi việc diễn ra tốt nhất.

Mục tiêu chính của Maven:

- Làm cho quá trình xây dựng phần mềm dễ dàng
- Cung cấp một hệ thống xây dựng phần mềm thống nhất
- Cung cấp những thông tin dự án quan trọng
- Khuyến khích việc thực hành phát triển dự án tốt hơn

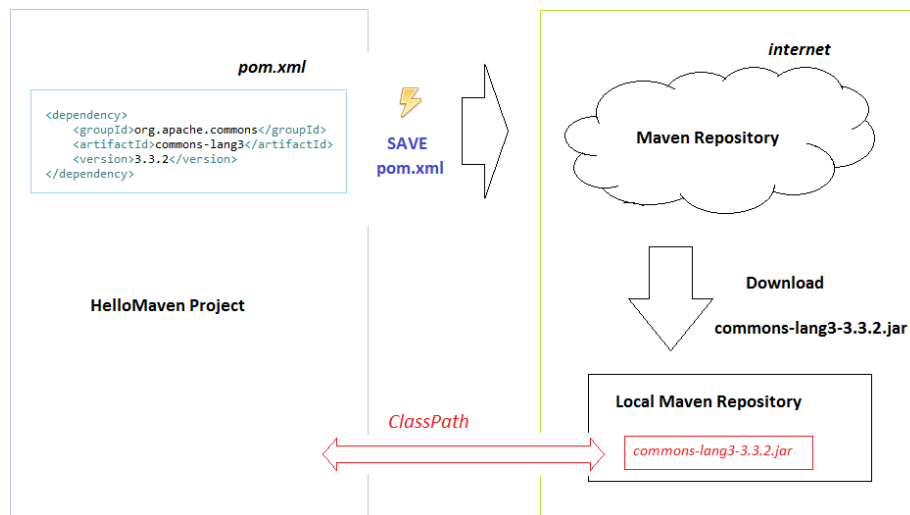
Các tính năng:

- Thiết lập dự án một cách đơn giản
- Cách sử dụng thống nhất trên tất cả dự án
- Tự động cập nhật phụ thuộc, phụ thuộc bắc cầu
- Tạo khả năng làm việc với nhiều dự án cùng lúc
- Khối lượng repository và thư viện khổng lồ có sẵn
- Có khả năng mở rộng
- Ít cấu hình cho các tính năng mới
- Tác vụ Ant để quản lý phụ thuộc và triển khai
- Xây dựng dựa trên Model
- Quản lý thông tin dự án
- Quản lý các phiên bản phát hành
- Quản lý phụ thuộc

Cấu trúc thư mục dự án Maven (với `${basedirectory}` là thư mục gốc dự án):

- `${basedir}/src/main/java`: thư mục mã nguồn java
- `${basedir}/src/main/resources`: thư mục tài nguyên
- `${basedir}/src/test`: thư mục chứa mã kiểm thử
- `${basedir}/target`: thư mục chứa file jar
- `${basedir}/target/classes`: thư mục chứa bytecode đã biên dịch
- `${basedir}/pom.xml`: file cấu hình dự án

Quản lý thư viện trong Maven:



Hình 3.7 Sử dụng pom.xml

Đối với các dự án truyền thống không sử dụng Apache Maven, khi cần dùng đến 1 thư viện bên ngoài không nằm trong bộ thư viện chuẩn của JDK, ví dụ như thư viện common-lang3 trong hình trên, công việc mà các lập trình viên cần làm là tìm kiếm trên internet file jar của thư viện đó (common-lang3.jar) sau đó download về copy vào trong dự án và khai báo classpath đến thư viện đó. Tuy nhiên đối với dự án sử dụng Maven, việc cần làm là khai báo common-lang3 trong 1 thẻ dependency trong file pom.xml và lưu lại, mọi công việc còn lại sẽ do Maven đảm nhận. Ngay sau khi lưu file pom.xml, Maven sẽ tự động kiểm tra trong Local Repository xem đã tồn tại thư viện đó chưa, nếu chưa, nó sẽ tìm kiếm và download thư viện về từ Maven Repository xuống nhờ Internet và lưu trong Local Repository, sau đó tự động khai báo classpath đến file jar vừa được download về. Như vậy, việc quản lý, tìm kiếm và sử dụng thư viện đã dễ dàng hơn rất nhiều, bên cạnh đó còn rất nhiều tính năng khác được hỗ trợ khi dự án java có sử dụng Apache Maven. Chính vì thế hiện tại Eclipse IDE cũng đã tích hợp sẵn Maven trong đó và được sử dụng dễ dàng chỉ với vài thao tác chuột.

3.4 Spring



Hình 3.8 Spring

Spring là một framework mã nguồn mở được dùng để phát triển các ứng dụng Java, nó giúp cho việc lập trình Java trở nên nhanh hơn, dễ dàng hơn, an toàn hơn cho tất cả lập trình viên. Spring tập trung vào tốc độ, sự đơn giản và hiệu quả, điều khiến nó trở thành Java framework phổ biến bậc nhất trên thế giới.

Lí do nên sử dụng Spring:

- Spring có mặt ở khắp mọi nơi: những công cụ của Spring được tin tưởng bởi các lập trình viên trên khắp thế giới. Nó mang đến những sản phẩm được sử dụng bởi hàng triệu người mỗi ngày. Spring cũng có sự đóng góp của rất nhiều ông lớn công nghệ như Alibaba, Amazon, Google, Microsoft...
- Spring linh hoạt: nó chứa hàng loạt các extension và thư viện bên thứ ba cho phép lập trình viên xây dựng hầu hết các ứng dụng mà họ có thể tưởng tượng ra. Bất kể là các trang web bảo mật, reactive hay microservices, hoặc xây dựng các luồng truyền dữ liệu phức tạp, đều có các công cụ được cung cấp để thực hiện việc đó.
- Spring hiệu quả: Spring Boot sẽ thay đổi cách các lập trình viên tiếp cận với nhiệm vụ, hợp lý hóa triệt để các trải nghiệm lập trình.
- Spring nhanh: hiệu năng cũng là vấn đề được quan tâm và cải thiện rất nhiều khi dùng Spring. Khởi tạo nhanh, thực thi nhanh, shutdown nhanh, hay ngay cả việc dựng nên một dự án Spring cũng chỉ mất vài giây với công cụ Spring Initializr.
- Spring bảo mật: sự an toàn khi sử dụng Spring cho dự án Java được đảm bảo. Hơn nữa, với Spring Security, việc tích hợp xác thực và phân quyền cho ứng dụng Java là rất nhanh chóng và dễ dàng.
- Spring hỗ trợ tốt: với cộng đồng lớn, rộng khắp toàn cầu, đa dạng, mọi lứa tuổi, mọi trình độ, từ người mới bắt đầu đến chuyên gia dày kinh nghiệm, sự hỗ trợ và tài nguyên là rất nhiều.

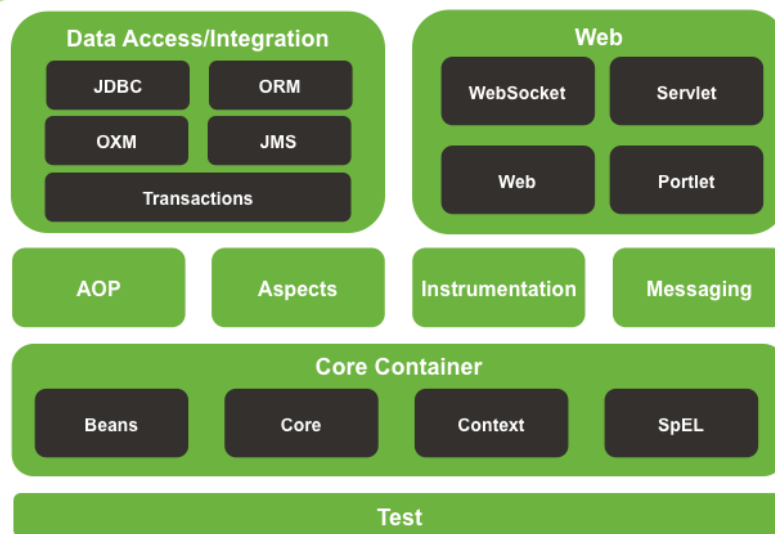
Spring được thiết kế theo các module, các lập trình viên có thể lựa chọn những module cần cho dự án và không cần quan tâm đến phần còn lại. Sau đây là các dự án con của Spring: Spring Boot, Spring Framework, Spring Data, Spring Cloud, Spring Security, Spring Cloud Data Flow, Spring Session, Spring Integration, Spring HATEOAS, Spring REST Docs, Spring Batch, Spring AMQP, Spring Android, Spring CredHub, Spring Flo, Spring Kafka, Spring LDAP, Spring Mobile, Spring Roo, Spring Shell, Spring Vault, Spring Statemachine, Spring Web Flow, Spring Web Services

3.4.1 Spring Framework

Đây là dự án đầu tiên cũng là dự án nền tảng mang tính cốt lõi của Spring. Sau này sự phát triển của các dự án khác cũng đều dựa trên những công nghệ mà Spring Framework đã cung cấp.



Spring Framework Runtime



Hình 3.9 Spring Framework

Spring Framework được chia ra làm nhiều module, ứng dụng có thể lựa chọn sử dụng những module nó cần.

- Core Container: chứa Core, Beans, Context và Expression Language. Module Core cung cấp phần cơ bản của framework bao gồm tính năng Inversion of Control và Dependency Injection. Module Beans cung cấp BeanFactory. Module Context được xây dựng dựa trên Core và Beans, nó là phương tiện để truy cập các đối tượng đã được định nghĩa và cấu hình, với interface ApplicationContext là trung tâm của module này. Module SpEL cung cấp một Expression Language mạnh mẽ để truy vấn và thao tác với các đối tượng trong quá trình runtime.
- Data Access/Integration: chứa JDBC, ORM, OXM, JMS và Transactions. Module JDBC cung cấp tầng trừu tượng hỗ trợ các yêu cầu sử dụng JDBC. Module ORM cung cấp khả năng tích hợp đến các ORM (Object Relational Mapping) API như JPA, JDO, Hibernate, iBatis. Module OXM cung cấp tầng trừu tượng hỗ trợ OXM (Object XML Mapping) như JAXB, Castor, XMLBeans, JiBX, Xstream. Module JMS chứa các tính năng tạo và nhận thông điệp. Và cuối cùng module Transactions hỗ trợ quản lý giao dịch cho những class nhất định và tất cả các POJO
- Web: chứa WebSocket, Web, Servlet, Portlet. Module Web và Servlet cung cấp các tính năng cho phát triển web như upload file, khởi tạo IoC trên servlet và application context dành cho web. Module WebSocket cung cấp khả năng hỗ trợ giao tiếp 2 chiều giữa web client và web server trong một ứng dụng web. Module Portlet cung cấp các chức năng tương tự như trong Servlet nhưng sử dụng trong môi trường portlet.
- Test: hỗ trợ kiểm thử với các framework Junit hoặc TestNG

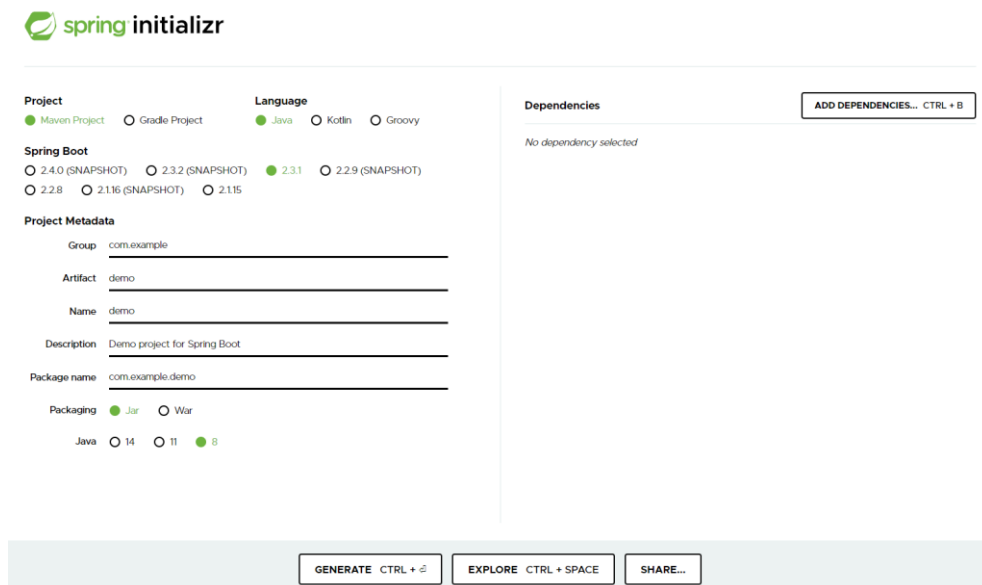
- AOP: cung cấp các chức năng liên quan tới lập trình hướng khía cạnh (Aspect Oriented Programming)
- Aspects: hỗ trợ tích hợp AspectJ – một framework mạnh mẽ của AOP
- Instrumentation: hỗ trợ class instrumentation và class loader được sử dụng tại các máy chủ ứng dụng
- Messaging: cung cấp sự hỗ trợ cho STOMP

3.4.2 Spring Boot

Spring Boot là một dự án thuộc hệ sinh thái Spring, nó giúp cho việc khởi tạo dự án Java Spring trở nên dễ dàng hơn. So với trước đây khi chưa có Spring Boot, những dự án Java Spring thường gặp phải rắc rối ở bước cấu hình rất phức tạp, nhưng với Spring Boot, lượng công việc để cấu hình và khởi chạy một dự án được giảm đi rất nhiều, và không cần đến bất kỳ dòng xml nào (ngoại trừ file pom.xml)

Mục tiêu chính của Spring Boot:

- Cung cấp khả năng tiếp cận và bắt đầu dự án Spring nhanh hơn
- Hỗ trợ tự động cấu hình
- Cung cấp các tính năng được sử dụng nhiều ở các dự án (ví dụ embedded server, security, health checks...)
- Hoàn toàn không tự sinh mã, không yêu cầu cấu hình xml



Hình 3.10 Spring Initializr

Ưu điểm của Spring Boot:

- Các dự án Spring Boot có thể được khởi tạo hoàn toàn tự động với công cụ Spring Initializr được Spring cung cấp
- Xây dựng trên nền tảng Spring Framework nên nó chứa đầy đủ các tính năng của Spring Framework

- Hỗ trợ tạo ra các ứng dụng stand-alone (chạy bằng file jar), ngay cả khi đó là các ứng dụng web vì server đã được nhúng trực tiếp vào dự án (Tomcat, Jetty)
- Giảm lượng code, tăng tốc độ phát triển dự án nhờ việc tự động cấu hình rất nhiều và không yêu cầu cấu hình bất kỳ dòng xml nào
- Có sẵn nhiều bộ thư viện hỗ trợ đặc biệt (các thư viện starter: spring-boot-starter, spring-boot-starter-web, spring-boot-starter-test, spring-boot-starter-websocket...)

3.4.3 Spring Security

Spring Security là một dự án trong hệ sinh thái Spring, nó cung cấp các tính năng xác thực, phân quyền và chống lại các cuộc tấn công. Spring Security là tiêu chuẩn chung trong việc bảo vệ an toàn cho các dự án Java Spring.



Hình 3.11 Spring Security

Spring Security cung cấp hỗ trợ toàn diện trong việc xác thực người dùng. Xác thực là cách phần mềm xác minh danh tính của những ai đang cố gắng truy cập vào tài nguyên hệ thống, cách xác thực người dùng thường gặp nhất là yêu cầu người dùng nhập vào tài khoản và mật khẩu. Xác thực thành công chính là cơ sở để ứng dụng thực hiện chức năng phân quyền.

Để đảm bảo bảo mật trong khi xác thực, Spring Security thực hiện lưu trữ mật khẩu người dùng đã được mã hóa. Các cơ chế mã hóa mật khẩu được sử dụng đều là mã hóa một chiều, bao gồm:

- DelegatingPasswordEncoder
- BcryptPasswordEncoder
- Argon2PasswordEncoder
- Pbkdf2PasswordEncoder
- ScryptPasswordEncoder

Các tính năng phân quyền nâng cao cũng chính là một trong những lí do lớn nhất cho sự phổ biến của Spring Security. Bất kể cách ứng dụng thực hiện xác thực là

gì, do Spring Security cung cấp hay là do một nhà cung cấp nào khác, thì đều có những dịch vụ phù hợp được tạo ra phục vụ việc phân quyền cho dự án một cách đơn giản và nhất quán.

Ngoài ra, Spring Security còn cung cấp sự bảo vệ ứng dụng khỏi sự tấn công phổ biến hiện nay: chống lại tấn công CSRF, sử dụng Security Response Headers, hỗ trợ cấu hình điều hướng cho ứng dụng sử dụng HTTPS, bảo vệ HSTS, hỗ trợ cấu hình cho Proxy Server.

3.4.4 Spring Data JPA

Spring Data JPA là một phần trong dự án Spring Data của hệ sinh thái Spring, nó giúp dễ dàng hơn trong việc xây dựng các JPA Repository, hỗ trợ nghiệp vụ truy xuất dữ liệu. Trước đây khi chưa có Spring Data JPA, việc triển khai tầng truy xuất dữ liệu (data access layer) là khá phức tạp, rất nhiều các đoạn mã thừa, lặp đi lặp lại chỉ để thực hiện những câu query đơn giản dùng cho phân trang, lọc dữ liệu. Spring Data JPA được sinh ra với mục đích cải thiện data access layer, giúp cho lập trình viên có thể thực hiện các thao tác với dữ liệu một cách tốt nhất.



Hình 3.12 Spring Data

JPA (Java Persistence API) là một đặc tả định nghĩa chung cho các API dành cho ORM framework và việc quản lý các đối tượng phục vụ ORM trong Java. Có rất nhiều framework ORM thực thi đặc tả trên, trong đó có thể kể tới là Hibernate và EclipseLink. Và Spring Data JPA đã được xây dựng dựa trên JPA, với Persistent Provider mặc định được lựa chọn là Hibernate. Nó sử dụng đầy đủ các tính năng của JPA, ngoài ra còn hỗ trợ thêm nhiều tính năng khác giúp loại bỏ hoàn toàn

các class repository như: tự động tạo câu truy vấn theo tên phương thức, hỗ trợ phân trang, sắp xếp, CRUD, thực hiện truy vấn động, ánh xạ entity bằng XML...

3.5 Thymeleaf

Thymeleaf là một Java Template Engine phía server dành cho các ứng dụng Java trên môi trường web và ứng dụng độc lập, với khả năng xử lý HTML, CSS, XML, Javascript và Text. Mục tiêu chính của Thymeleaf là tạo ra các view template dễ dàng bảo trì. Để đạt được điều này, Thymeleaf được xây dựng dựa trên ý tưởng giữ nguyên template và tiêm logic vào các file template đó sao cho không làm thay đổi cấu trúc như trong mẫu thiết kế. Điều này làm tăng sự dễ dàng trong hợp tác giữa đội thiết kế giao diện và đội phát triển phần mềm.



Hình 3.13 Thymeleaf

Thymeleaf hỗ trợ 6 loại template:

- HTML
- CSS
- Javascript
- XML
- Text
- Raw

Ưu điểm khi sử dụng Thymeleaf:

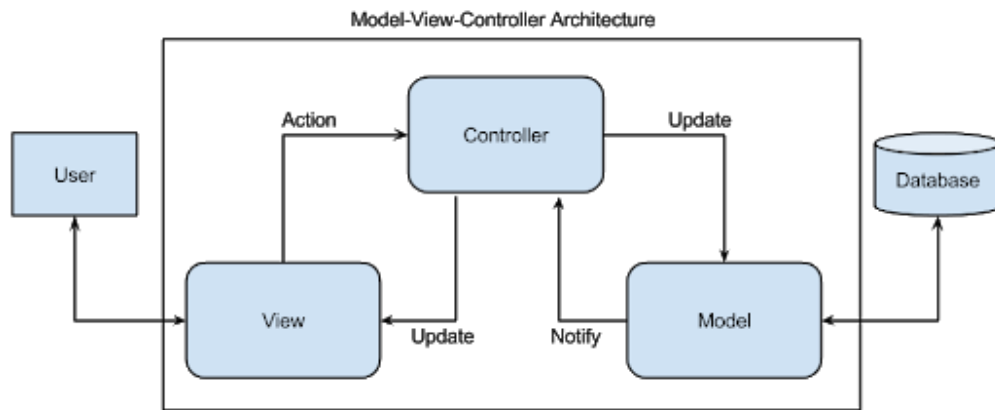
- Dễ dàng tích hợp vào dự án Spring Boot: Thymeleaf được Spring đề cử làm View Technology cho các dự án của mình. Ngoài ra, Spring Boot còn cung cấp thư viện spring-boot-starter-thymeleaf để hỗ trợ việc cấu hình nhanh để sử dụng Thymeleaf trong dự án.
- Thao tác trực tiếp với file html: đối với các dự án web sử dụng JSP, lập trình viên cần thao tác với các file jsp tạo ra nhiều phiền toái
- Không làm thay đổi file template: Thymeleaf hầu như giữ nguyên file thiết kế, không sử dụng thêm các thẻ non-html, mà dùng các attribute để render view (đối với JSP phải import taglib và dùng các thẻ được định nghĩa trong taglib)

CHƯƠNG 4. PHÁT TRIỂN VÀ TRIỂN KHAI ỨNG DỤNG

4.1 Thiết kế kiến trúc

4.1.1 Lựa chọn kiến trúc phần mềm

Qua các bước khảo sát và phân tích hệ thống, em quyết định sử dụng mô hình MVC (Model – View – Controller) làm kiến trúc phần mềm cho sản phẩm của mình. Model – View – Controller là một mẫu thiết kế phần mềm, kiến trúc phần mềm được sử dụng rất phổ biến trong ngành lập trình phần mềm, không chỉ là những ứng dụng web mà còn có thể áp dụng cho ứng dụng desktop, ứng dụng trên điện thoại... Sau đây là sơ đồ của mô hình MVC:



Hình 4.1 Mô hình MVC

Giống như cái tên của nó, mô hình này được chia ra làm 3 phần:

- Model: nơi chứa các lớp quy định cấu trúc dữ liệu của các đối tượng hoạt động trong hệ thống (ví dụ Product, Account, Order...) và các liên kết ràng buộc giữa các đối tượng đó. Cùng với đó là các nghiệp vụ tương tác với dữ liệu như truy vấn dữ liệu, thêm sửa xóa dữ liệu vào trong cơ sở dữ liệu đã được kết nối.
- View: nơi chứa toàn bộ giao diện mà hệ thống dùng để trả về cho trình duyệt và hiển thị cho người dùng, các giao diện này cung cấp phương thức giúp người dùng có thể tương tác với hệ thống (thông qua các input, button, checkbox...)
- Controller: nằm giữa Model và View, tạo ra mối liên kết giữa người dùng hệ thống và dữ liệu trong hệ thống. Nó chứa các lớp điều khiển, nhận về request từ người dùng thông qua View, thực hiện các request đó, sử dụng các nghiệp vụ mà tầng Model đã cung cấp để giao tiếp với cơ sở dữ liệu và trả về dữ liệu mới, cập nhật View rồi hiển thị lên cho người dùng.

Ưu điểm của mô hình MVC:

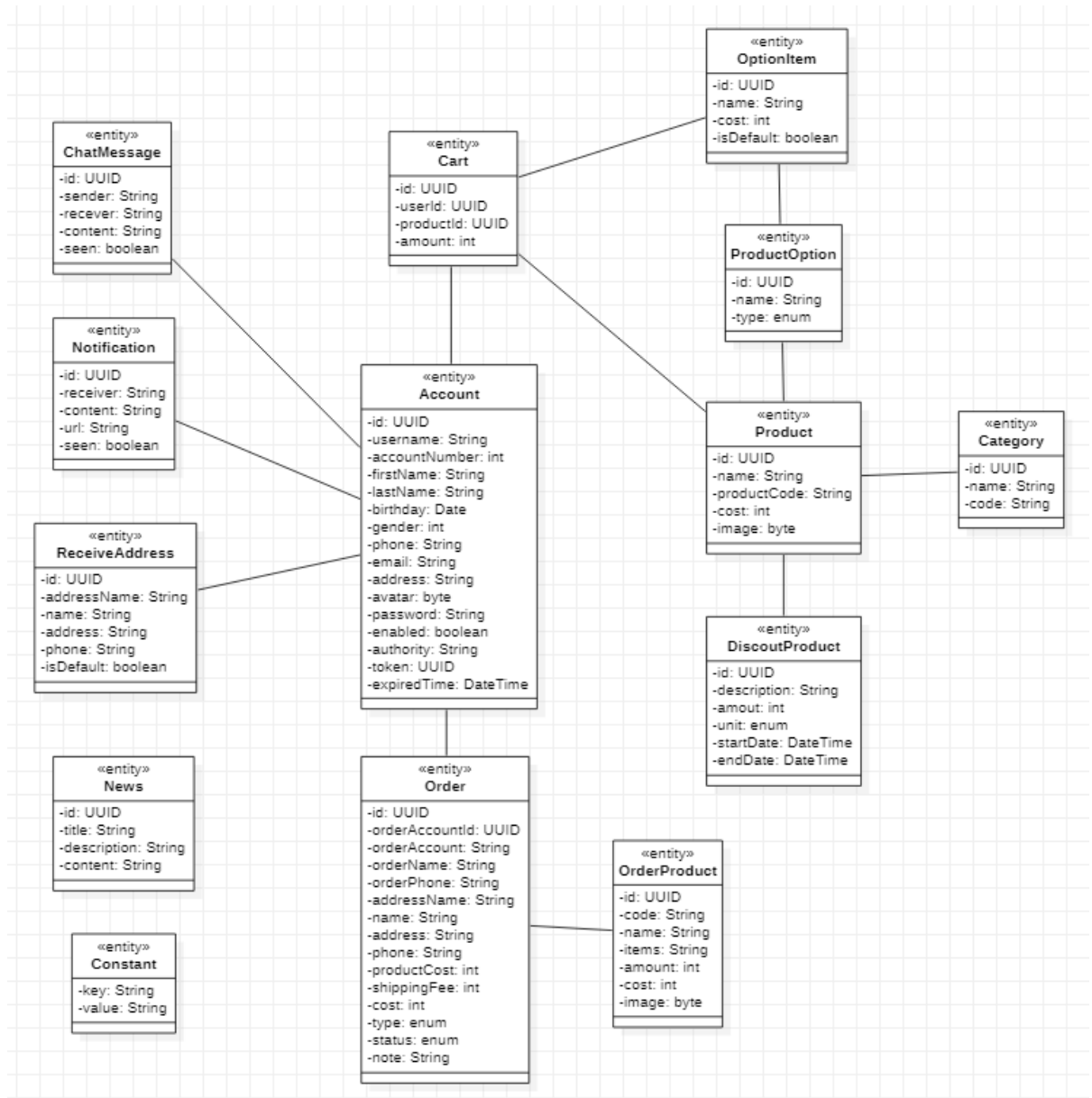
- Phát triển song song: lập trình viên có thể làm việc đồng thời trên cả 3 phần Model, View và Controller mà không cần ràng buộc lẫn nhau

- Trình tự xử lý rõ ràng: luồng xử lý nghiệp vụ và truyền dữ liệu rất rành mạch dễ hiểu, ngay cả đối với những thành viên mới cũng có thể tiếp cận dự án dễ dàng
- Dễ sửa đổi: do chia làm 3 phần nên việc phát triển và sửa đổi hệ thống sau này trở nên dễ dàng hơn
- Dễ kiểm thử: các phần có thể được kiểm thử mà không phụ thuộc lẫn nhau

Nhược điểm:

- Tổ ra kém hiệu quả đối với các hệ thống lớn có độ phức tạp cao
- Các tính năng phần mềm nếu thay đổi sẽ yêu cầu sự thay đổi lên tất cả 3 phần mới đảm bảo được tính chính xác của tính năng
- Các lập trình viên phải có khả năng hiểu và kỹ năng làm việc tốt trên toàn bộ 3 phần của dự án
- Để đảm bảo được mô hình MVC, có thể tồn tại nhiều đoạn mã thừa

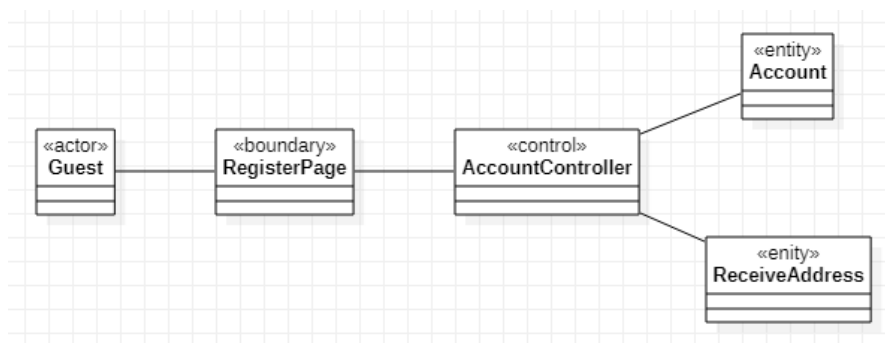
4.1.2 Mô hình hóa các lớp thực thể



Hình 4.2 Mô hình hóa các lớp thực thể

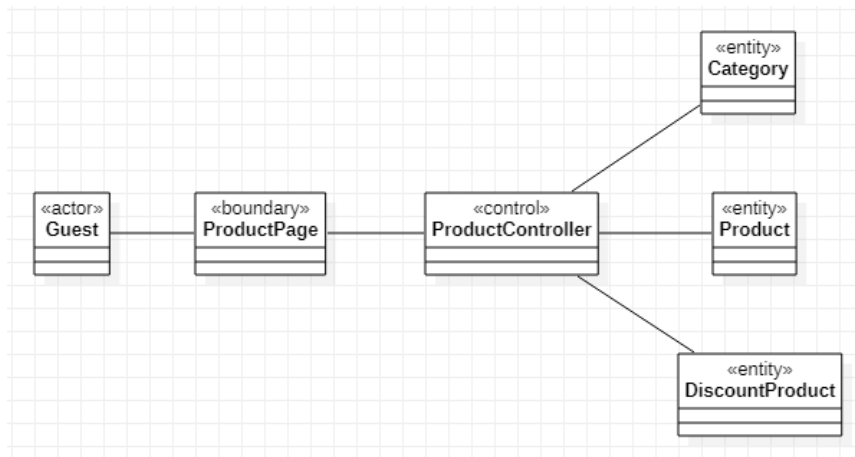
4.1.3 Xác định lớp tham gia từng use case

4.1.3.1. Lớp tham gia use case Đăng ký



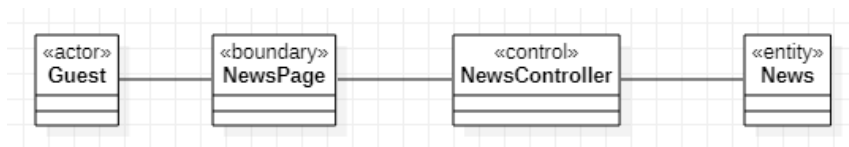
Hình 4.3 Lớp tham gia use case Đăng ký

4.1.3.2. Lớp tham gia use case Tìm kiếm sản phẩm



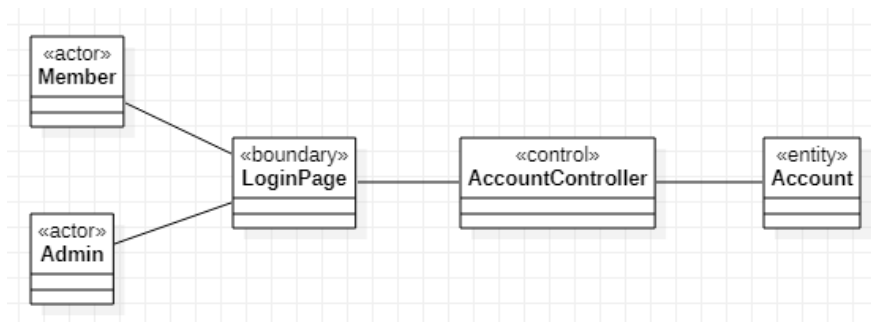
Hình 4.4 Lớp tham gia use case Tìm kiếm sản phẩm

4.1.3.3. Lớp tham gia use case Xem bài viết



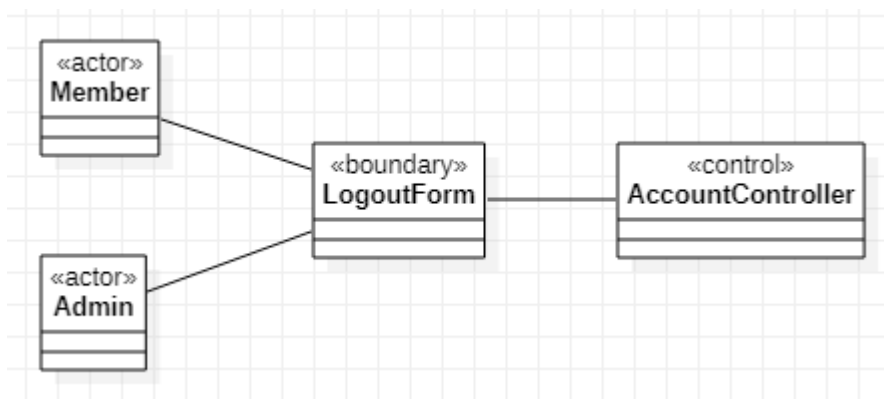
Hình 4.5 Lớp tham gia use case Xem bài viết

4.1.3.4. Lớp tham gia use case Đăng nhập



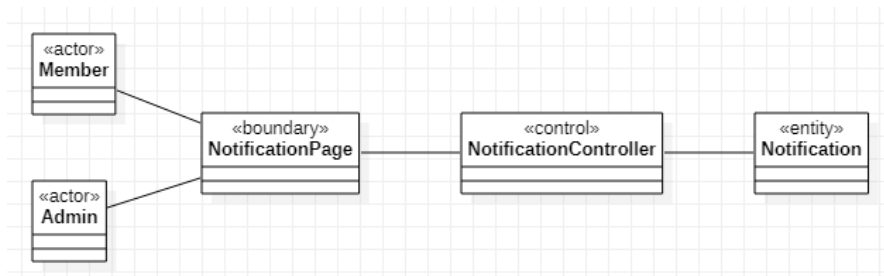
Hình 4.6 Lớp tham gia use case Đăng nhập

4.1.3.5. Lớp tham gia use case Đăng xuất



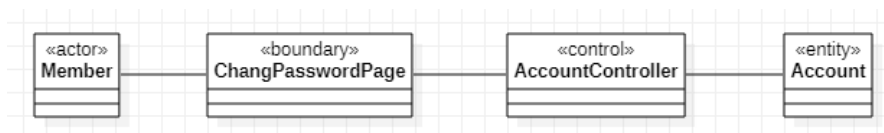
Hình 4.7 Lớp tham gia use case Đăng xuất

4.1.3.6. Lớp tham gia use case Nhận thông báo đơn hàng



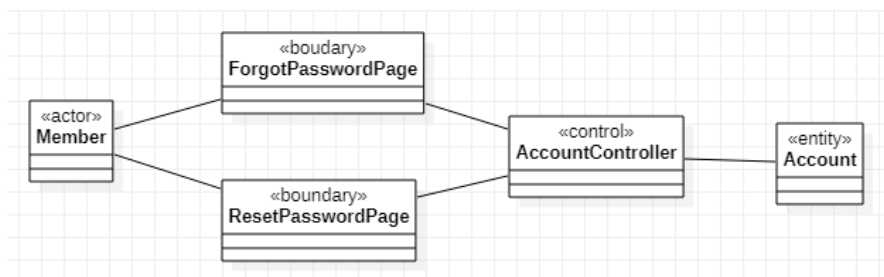
Hình 4.8 Lớp tham gia use case Nhận thông báo đơn hàng

4.1.3.7. Lớp tham gia use case Đổi mật khẩu



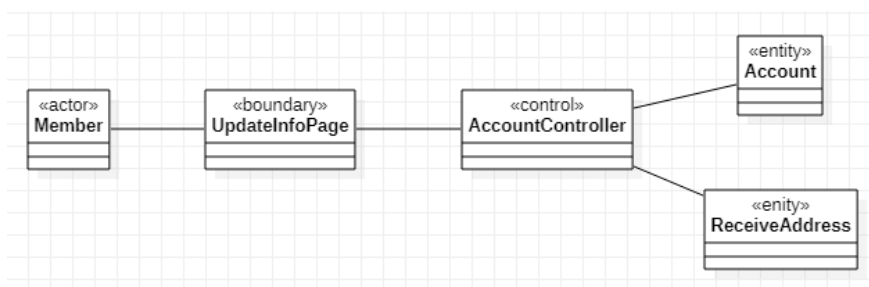
Hình 4.9 Lớp tham gia use case Đổi mật khẩu

4.1.3.8. Lớp tham gia use case Lấy lại mật khẩu



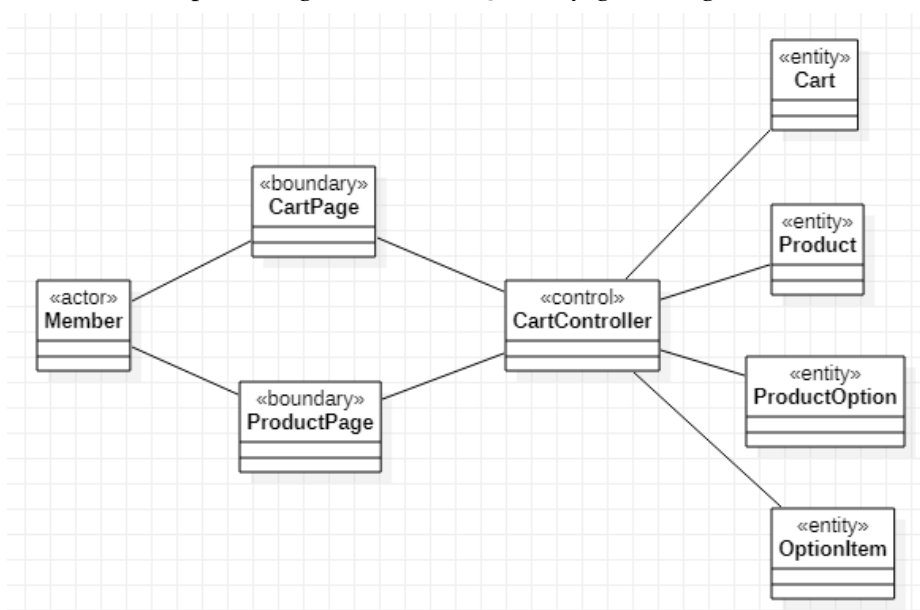
Hình 4.10 Lớp tham gia use case Lấy lại mật khẩu

4.1.3.9. Lớp tham gia use case Thay đổi thông tin



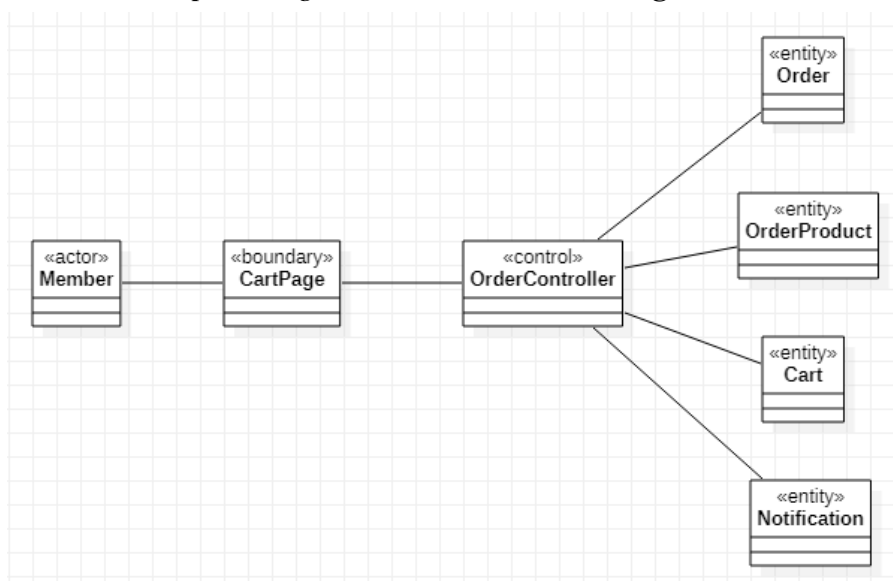
Hình 4.11 Lớp tham gia use case Thay đổi thông tin

4.1.3.10. Lớp tham gia use case Quản lý giỏ hàng



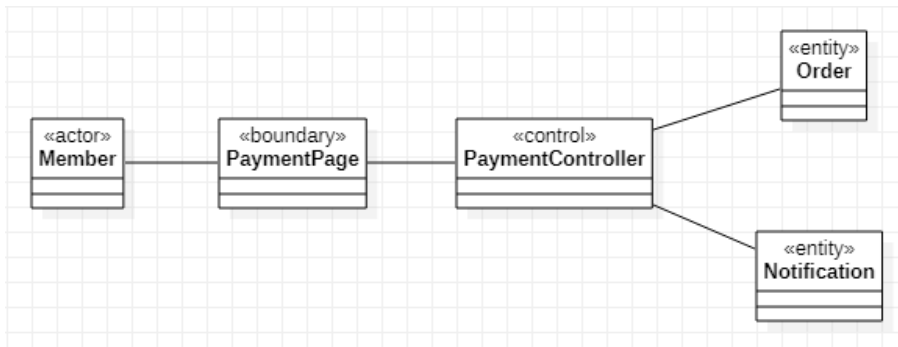
Hình 4.12 Lớp tham gia use case Quản lý giỏ hàng

4.1.3.11. Lớp tham gia use case Tạo đơn hàng



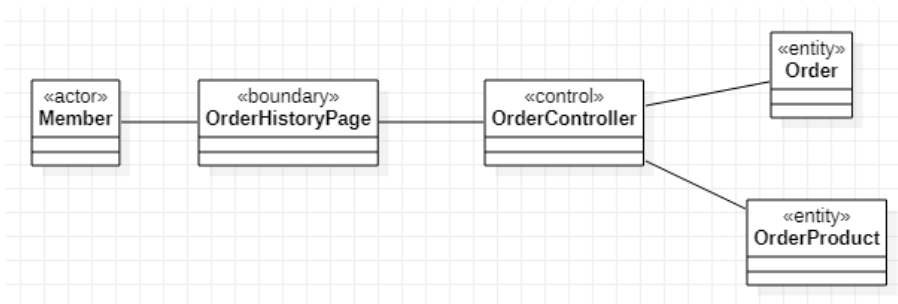
Hình 4.13 Lớp tham gia use case Tạo đơn hàng

4.1.3.12. Lớp tham gia use case Thanh toán đơn hàng



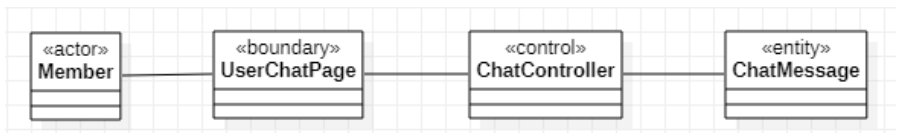
Hình 4.14 Lớp tham gia use case Thanh toán đơn hàng

4.1.3.13. Lớp tham gia use case Xem lịch sử mua hàng



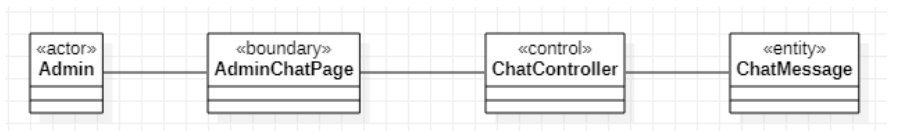
Hình 4.15 Lớp tham gia use case Xem lịch sử mua hàng

4.1.3.14. Lớp tham gia use case Chat với quản trị viên



Hình 4.16 Lớp tham gia use case Chat với quản trị viên

4.1.3.15. Lớp tham gia use case Chat với người dùng



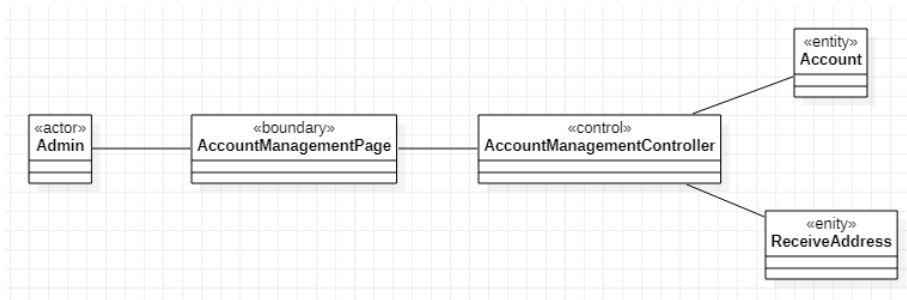
Hình 4.17 Lớp tham gia use case Chat với người dùng

4.1.3.16. Lớp tham gia use case Cài đặt hệ thống



Hình 4.18 Lớp tham gia use case Cài đặt hệ thống

4.1.3.17. Lớp tham gia use case Quản lý tài khoản



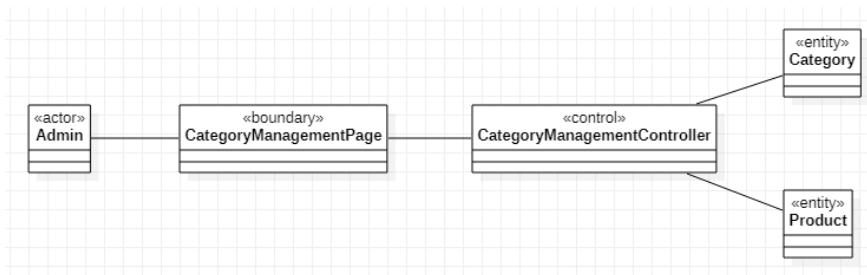
Hình 4.19 Lớp tham gia use case Quản lý tài khoản

4.1.3.18. Lớp tham gia use case Quản lý bài viết



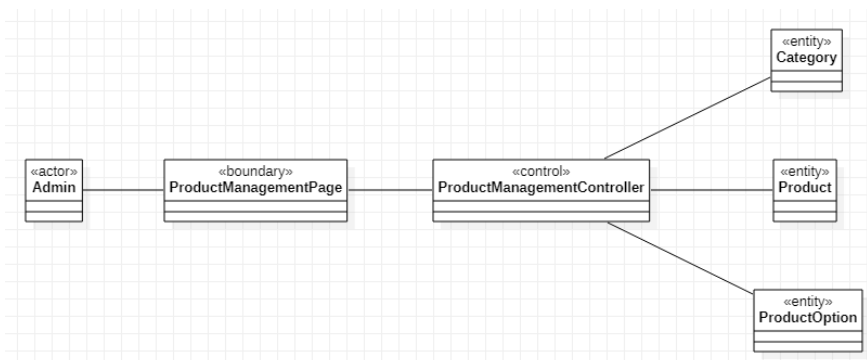
Hình 4.20 Lớp tham gia use case Quản lý bài viết

4.1.3.19. Lớp tham gia use case Quản lý danh mục



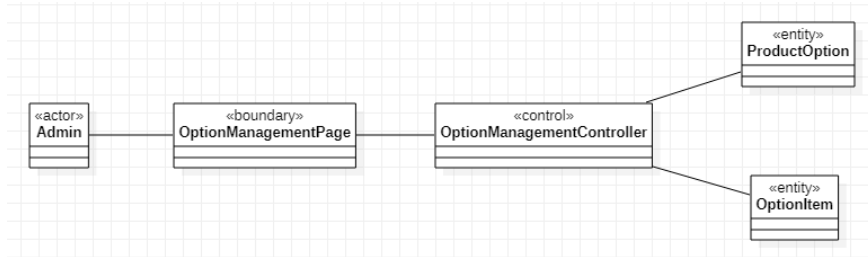
Hình 4.21 Lớp tham gia use case Quản lý danh mục

4.1.3.20. Lớp tham gia use case Quản lý sản phẩm



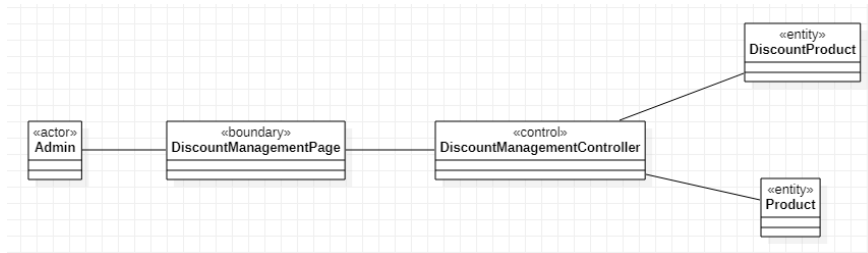
Hình 4.22 Lớp tham gia use case Quản lý sản phẩm

4.1.3.21. Lớp tham gia use case Quản lý tùy chọn



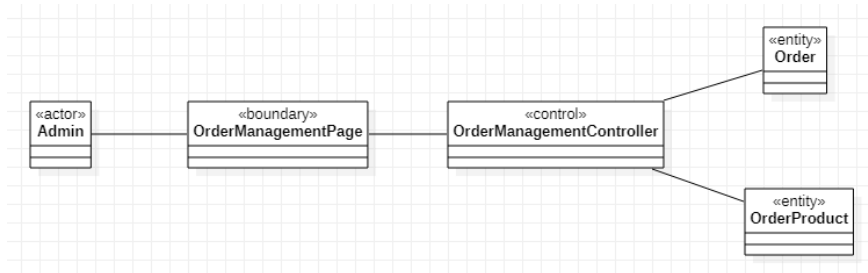
Hình 4.23 Lớp tham gia use case Quản lý tùy chọn

4.1.3.22. Lớp tham gia use case Quản lý khuyến mãi



Hình 4.24 Lớp tham gia use case Quản lý khuyến mãi

4.1.3.23. Lớp tham gia use case Quản lý đơn hàng



Hình 4.25 Lớp tham gia use case Quản lý đơn hàng

4.2 Thiết kế chi tiết

4.2.1 Thiết kế lớp

Sau đây là thiết kế chi tiết của 1 số lớp quan trọng:

4.2.1.1. Thiết kế lớp NotificationController

Bảng 4.1 Thiết kế lớp NotificationController

Phương thức	Đầu vào	Đầu ra	Mô tả
getNotifications		Danh sách thông báo	Sử dụng username từ phiên đăng nhập hệ thống, lấy và trả về tất cả các thông báo của username đó

seenNotification	notificationId	boolean	Nhận vào id của thông báo, cập nhật trạng thái của thông báo đó là “Đã xem” và trả về kết quả thực hiện
------------------	----------------	---------	---

4.2.1.2. Thiết kế lớp AccountController

Bảng 4.2 Thiết kế lớp AccountController

Phương thức	Đầu vào	Đầu ra	Mô tả
register	Account	boolean	Nhận vào thông tin tài khoản, thực hiện kiểm tra tính hợp lệ của thông tin và thêm tài khoản đó vào hệ thống cùng với 1 địa chỉ nhận hàng mặc định được tạo ra từ các thông tin đăng ký nhận vào rồi trả về kết quả thực hiện
validateRegister	Account	boolean	Kiểm tra sự tồn tại của username, tính hợp lệ của thông tin tài khoản nhận vào rồi trả về kết quả
login	username, password	String	Kiểm tra username có tồn tại hay không, nếu có, kiểm tra tính chính xác của mật khẩu tương ứng với username đó và trả về kết quả
logout		boolean	Xóa phiên đăng nhập hiện tại của người dùng trên hệ thống và trả về kết quả thực hiện
changePassword	oldPassword, newPassword	String	Kiểm tra tính hợp lệ của mật khẩu cũ và mật khẩu mới, nếu đúng, thực hiện đổi sang mật khẩu mới cho tài khoản đó và trả về kết quả

validatePassword	oldPassword, newPassword	boolean	Kiểm tra tính chính xác của mật khẩu cũ đối với tài khoản đang đăng nhập, sự khác nhau giữa mật khẩu cũ và mới, và tính hợp lệ của mật khẩu mới (độ dài, kí tự cho phép) rồi trả về kết quả
forgotPassword	username	boolean	Cập nhật token lấy lại mật khẩu và ngày hết hạn thao tác lấy lại mật khẩu sau đó gửi email lấy lại mật khẩu đến địa chỉ mail của tài khoản đầu vào rồi trả về kết quả thực hiện
sendEmail	email, token	boolean	Thực hiện gửi email đến địa chỉ đầu vào với nội dung lấy lại mật khẩu
resetPassword	token, newPassword	boolean	Thực hiện tìm kiếm tài khoản tương ứng với token đầu vào trong database, kiểm tra ngày hết hạn thao tác, và đổi mật khẩu của tài khoản đó sang mật khẩu mới rồi trả về kết quả
updateInfo	command	boolean	Lấy những thông tin muốn cập nhật trong đối tượng command, kiểm tra tính hợp lệ của những thông tin đó và thực hiện thay đổi cho tài khoản đang đăng nhập hệ thống rồi trả về kết quả thực hiện
validateInfo	command	boolean	Kiểm tra tính hợp lệ của thông tin tài khoản nhận vào rồi trả về kết quả

addAddress	ReceiveAddress	boolean	Nhận về và thực hiện thêm mới địa chỉ nhận hàng cho tài khoản đang đăng nhập rồi trả về kết quả thực hiện
editAddress	ReceiveAddress	boolean	Lấy id của địa chỉ đầu vào, tìm kiếm trong cơ sở dữ liệu địa chỉ nhận hàng tương ứng với id đó và thực hiện thay đổi các thông tin còn lại của địa chỉ nhận hàng rồi trả về kết quả thực hiện
deleteAddress	id	boolean	Xóa 1 địa chỉ nhận hàng theo id truyền vào rồi trả về kết quả
getAccountInfo		Thông tin tài khoản	Lấy ra và trả về thông tin của tài khoản đang đăng nhập và những địa chỉ nhận hàng của tài khoản đó

4.2.1.3. Thiết kế lớp ChatController

Bảng 4.3 Thiết kế lớp ChatController

Phương thức	Đầu vào	Đầu ra	Mô tả
getChatMessages	partner	Danh sách tin nhắn	Lấy ra tất cả các tin nhắn giữa người thứ nhất là username đang đăng nhập vào hệ thống với người thứ 2 là partner, chuẩn dữ liệu liên quan đến tin nhắn đó và trả về danh sách tin nhắn
getListPartner		Danh sách tài khoản	Lấy tất cả các tài khoản có tin nhắn liên hệ với username đang đăng nhập trong hệ thống và trả về kết quả
addChatMessage	partner, message	boolean	Thêm tin nhắn vào cơ sở dữ liệu với người gửi là username đang đăng

			nhập hệ thống và người nhận là partner đầu vào rồi trả về kết quả thực hiện
seenMessage	partner	boolean	Cập nhật tất cả tin nhắn với người gửi là partner và người nhận là username đang đăng nhập vào hệ thống thành trạng thái “Đã xem” và trả về kết quả thực hiện

4.2.1.4. Thiết kế lớp AccountManagementController

Bảng 4.4 Thiết kế lớp AccountManagementController

Phương thức	Đầu vào	Đầu ra	Mô tả
datatable	datatableParameter	Danh sách tài khoản	Nhận vào các tham số được gửi từ thư viện Datatable và lấy tất cả các tài khoản thỏa mãn tham số đó cùng các thông tin liên quan và trả về kết quả
updateStatus	accountId, status	boolean	Cập nhật tài khoản có id là accountId đầu vào thành trạng thái là status rồi trả về kết quả thực hiện
getAccountInfo	accountId	Thông tin tài khoản	Lấy và trả về thông tin chi tiết và các địa chỉ nhận hàng của tài khoản có id là accountId đầu vào

4.2.1.5. Thiết kế lớp OptionManagementController

Bảng 4.5 Thiết kế lớp OptionManagementController

Phương thức	Đầu vào	Đầu ra	Mô tả
optionDatatable	datatableParameter	Danh sách tùy chọn	Nhận vào các tham số được gửi từ thư viện Datatable và lấy tất cả các tùy chọn thỏa mãn tham số đó và trả về kết quả
itemDatatable	datatableParameter	Danh sách lựa chọn	Nhận vào các tham số được gửi từ thư viện

			Datatable và lấy tất cả các lựa chọn thỏa mãn tham số đó và trả về kết quả
addOption	ProductOption	boolean	Thêm 1 tùy chọn vào cơ sở dữ liệu và trả về kết quả thực hiện
editOption	ProductOption	boolean	Lấy id trong tham số đầu vào, tìm kiếm tùy chọn trong cơ sở dữ liệu theo id đó và cập nhật những thông tin còn lại rồi trả về kết quả thực hiện
deleteOption	optionId	boolean	Thực hiện xóa tùy chọn theo id đầu vào và tất cả các lựa chọn của tùy chọn đó trong cơ sở dữ liệu rồi trả về kết quả thực hiện
addItem	OptionItem	boolean	Thêm 1 lựa chọn vào cơ sở dữ liệu và trả về kết quả thực hiện
updateItem	OptionItem	boolean	Lấy id trong tham số đầu vào, tìm kiếm lựa chọn trong cơ sở dữ liệu theo id đó và cập nhật những thông tin còn lại rồi trả về kết quả thực hiện
deleteItem	itemId	boolean	Thực hiện xóa 1 lựa chọn theo id đầu vào và trả về kết quả thực hiện

4.2.1.6. Thiết kế lớp SettingController

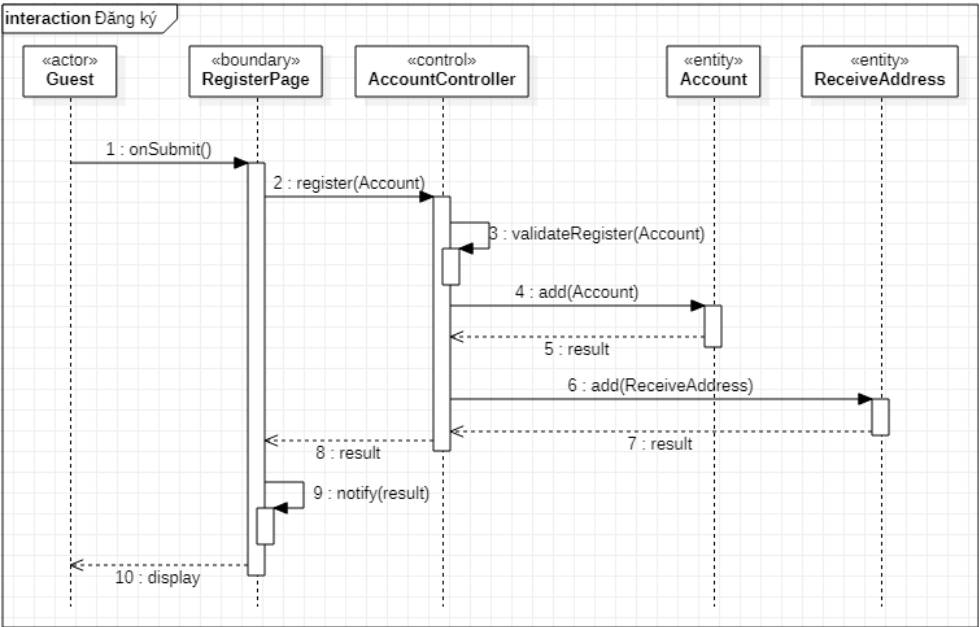
Bảng 4.6 Thiết kế lớp SettingController

Phương thức	Đầu vào	Đầu ra	Mô tả
loadSetting		Danh sách cài đặt	Lấy tất cả các cài đặt hiện tại trong cơ sở dữ liệu và đặt giá trị mặc định cho những cài đặt chưa được lưu trước đây rồi trả về kết quả
updateSetting	Danh sách		Nhận về danh sách các

	cài đặt		cài đặt và cập nhật tương ứng vào trong cơ sở dữ liệu rồi trả về kết quả thực hiện
--	---------	--	--

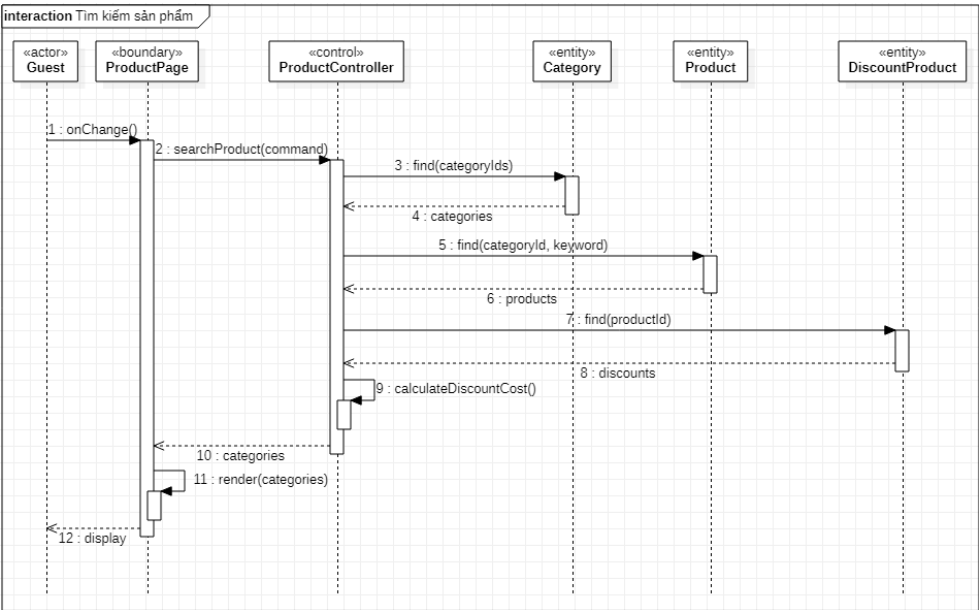
4.2.2 Biểu đồ trình tự cho từng use case

4.2.2.7. Biểu đồ trình tự use case Đăng ký



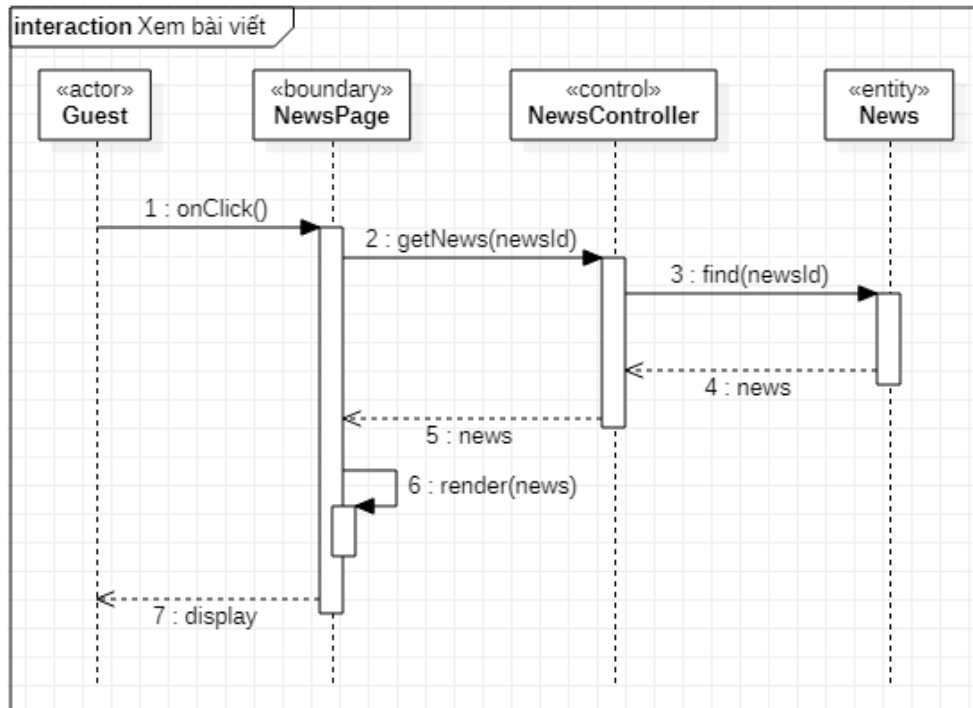
Hình 4.26 Biểu đồ trình tự use case Đăng ký

4.2.2.8. Biểu đồ trình tự use case Tìm kiếm sản phẩm



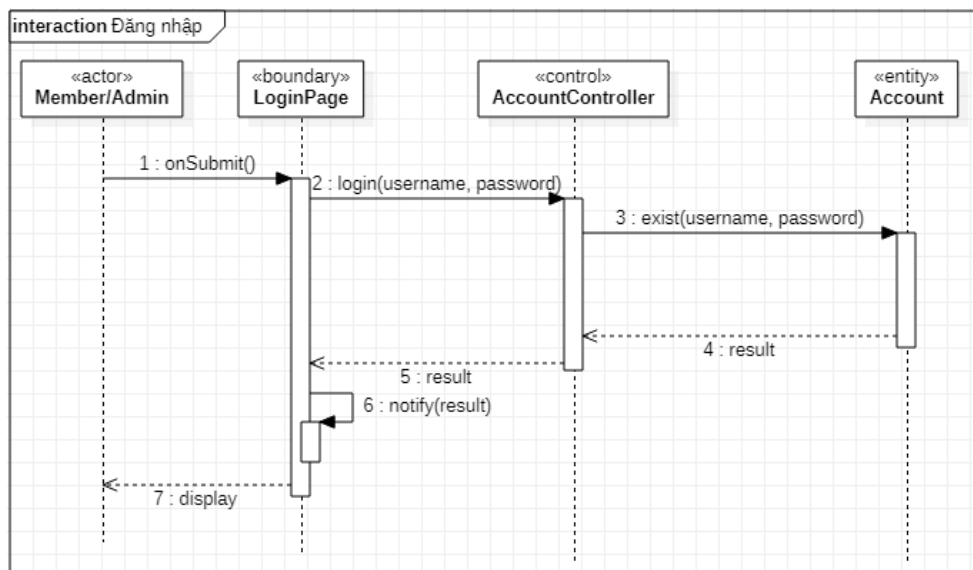
Hình 4.27 Biểu đồ trình tự use case Tìm kiếm sản phẩm

4.2.2.9. Biểu đồ trình tự use case Xem bài viết



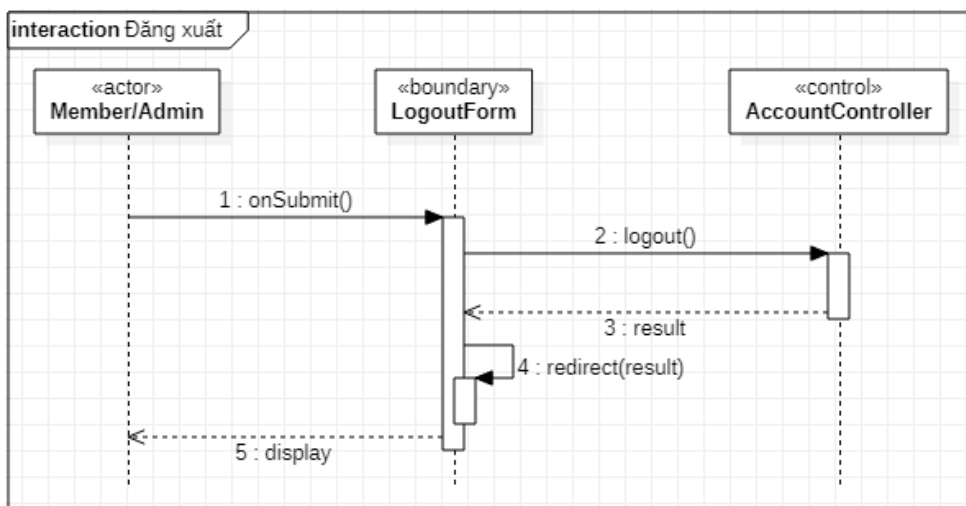
Hình 4.28 Biểu đồ trình tự use case Xem bài viết

4.2.2.10. Biểu đồ trình tự use case Đăng nhập



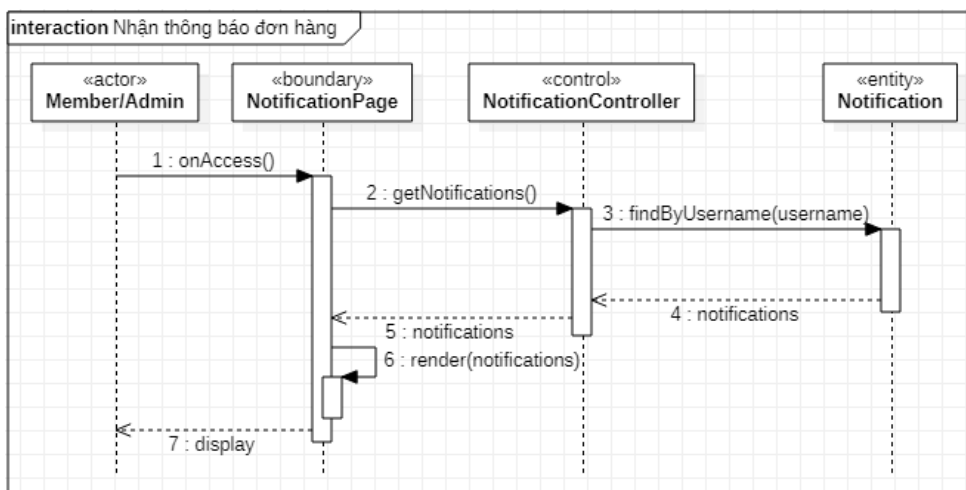
Hình 4.29 Biểu đồ trình tự use case Đăng nhập

4.2.2.11. Biểu đồ trình tự use case Đăng xuất



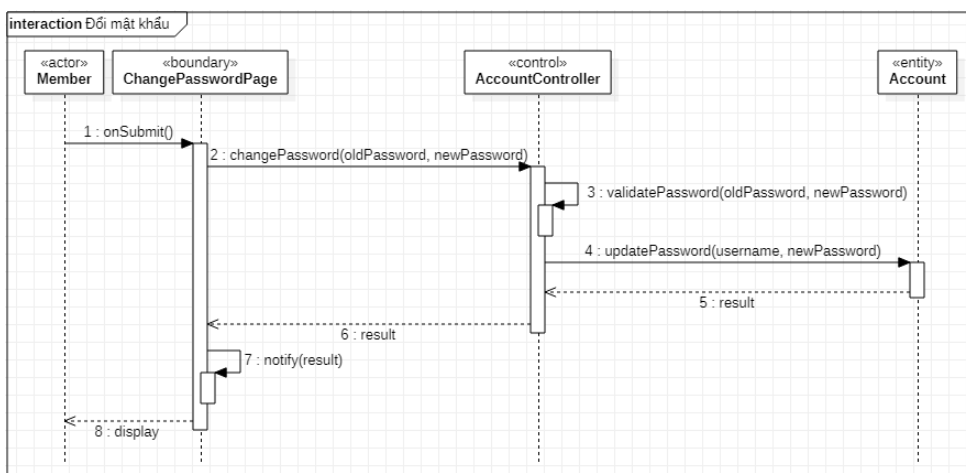
Hình 4.30 Biểu đồ trình tự use case Đăng xuất

4.2.2.12. Biểu đồ trình tự use case Nhận thông báo đơn hàng



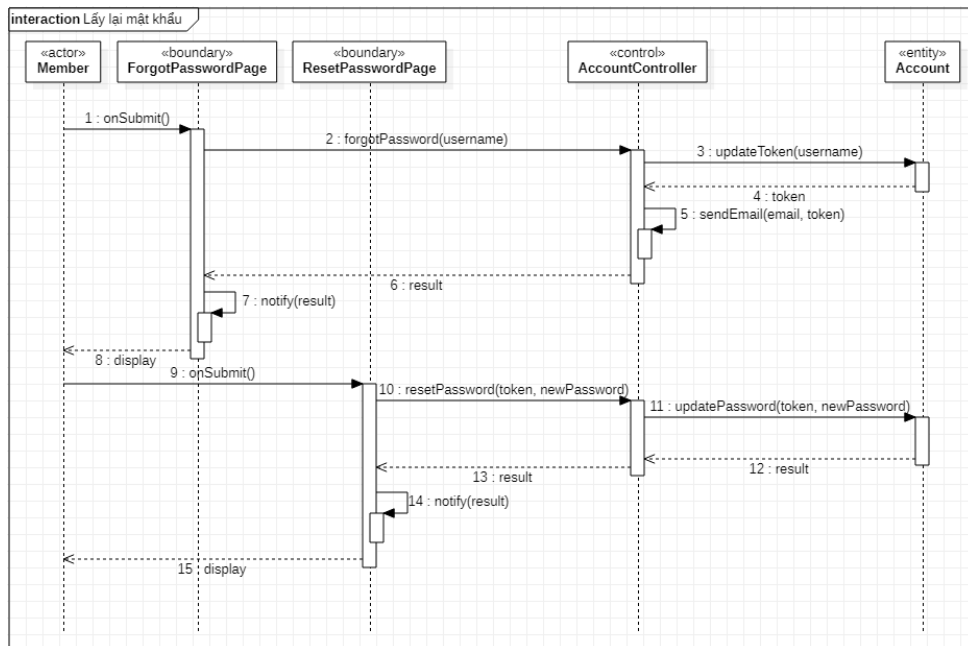
Hình 4.31 Biểu đồ trình tự use case Nhận thông báo đơn hàng

4.2.2.13. Biểu đồ trình tự use case Đổi mật khẩu



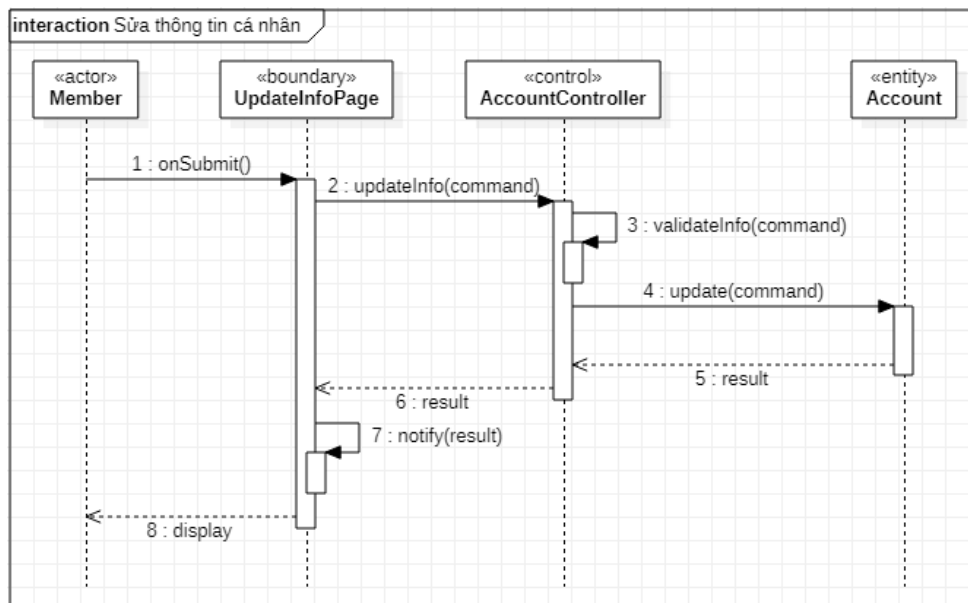
Hình 4.32 Biểu đồ trình tự use case Đổi mật khẩu

4.2.2.14. Biểu đồ trình tự use case Lấy lại mật khẩu



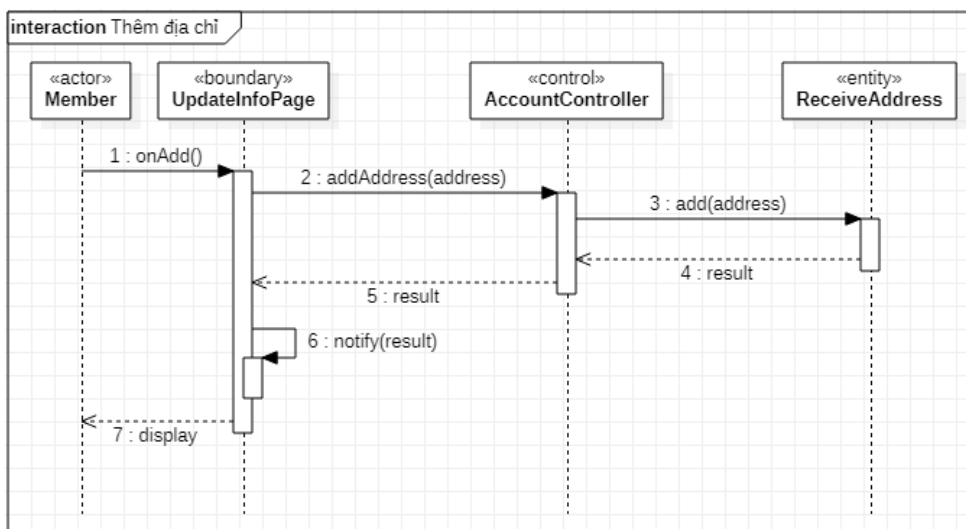
Hình 4.33 Biểu đồ trình tự use case Lấy lại mật khẩu

4.2.2.15. Biểu đồ trình tự use case Sửa thông tin cá nhân



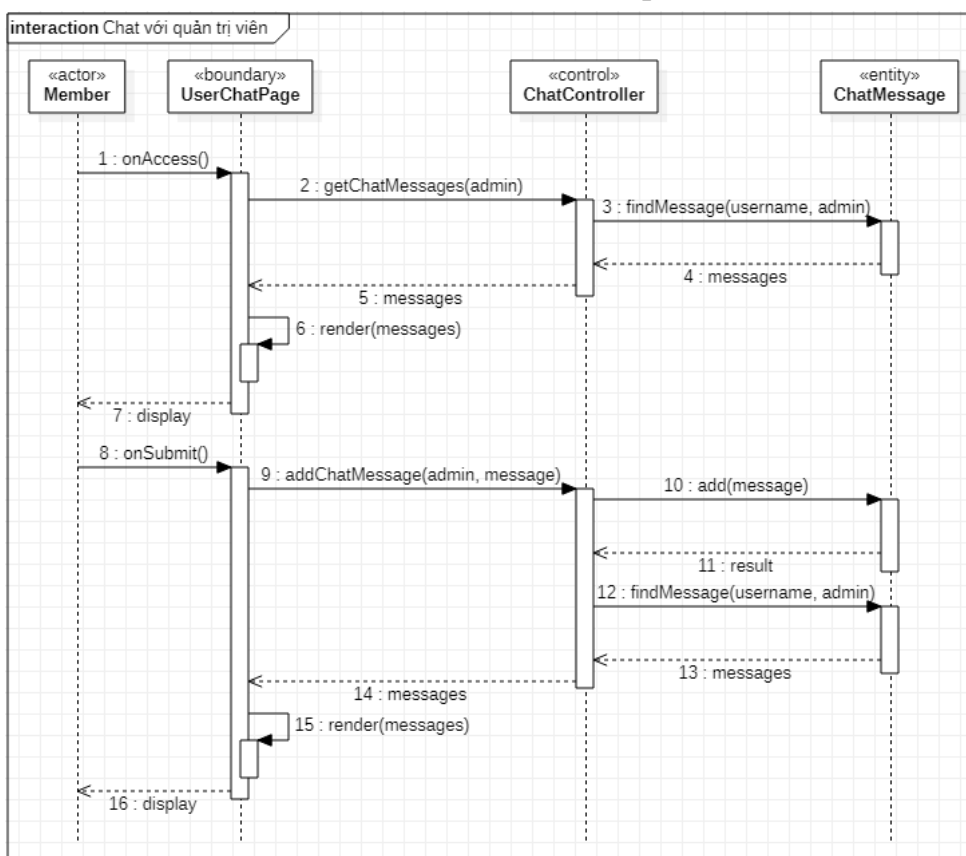
Hình 4.34 Biểu đồ trình tự use case Sửa thông tin cá nhân

4.2.2.16. Biểu đồ trình tự use case Thêm địa chỉ nhận hàng



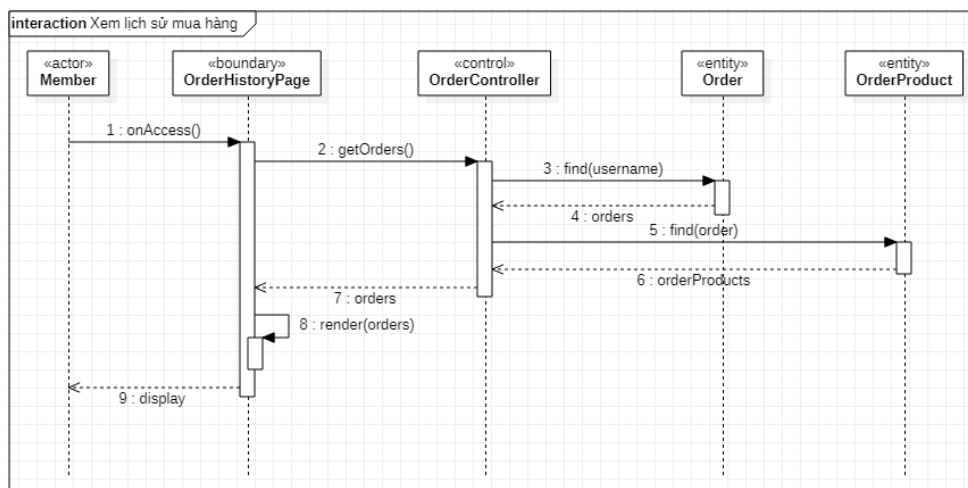
Hình 4.35 Biểu đồ trình tự use case Thêm địa chỉ nhận hàng

4.2.2.17. Biểu đồ trình tự use case Chat với quản trị viên



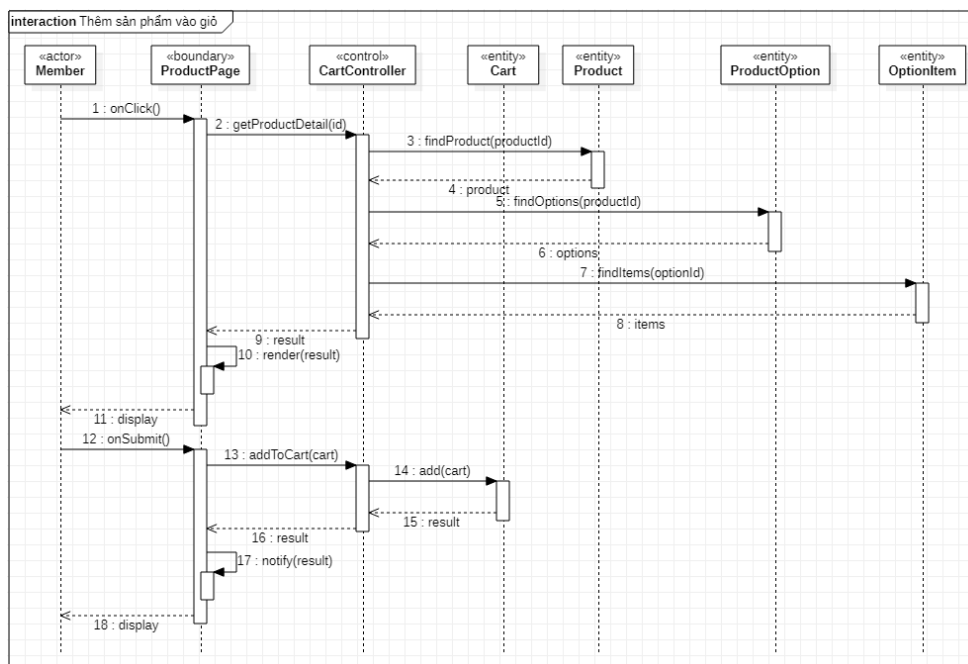
Hình 4.36 Biểu đồ trình tự use case Chat với quản trị viên

4.2.2.18. Biểu đồ trình tự use case Xem lịch sử mua hàng



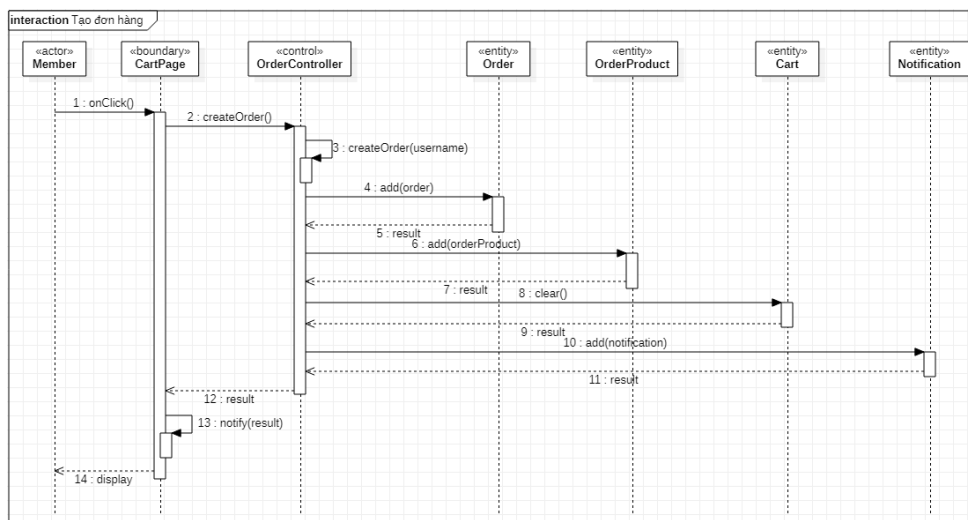
Hình 4.37 Biểu đồ trình tự use case Xem lịch sử mua hàng

4.2.2.19. Biểu đồ trình tự use case Thêm sản phẩm vào giỏ



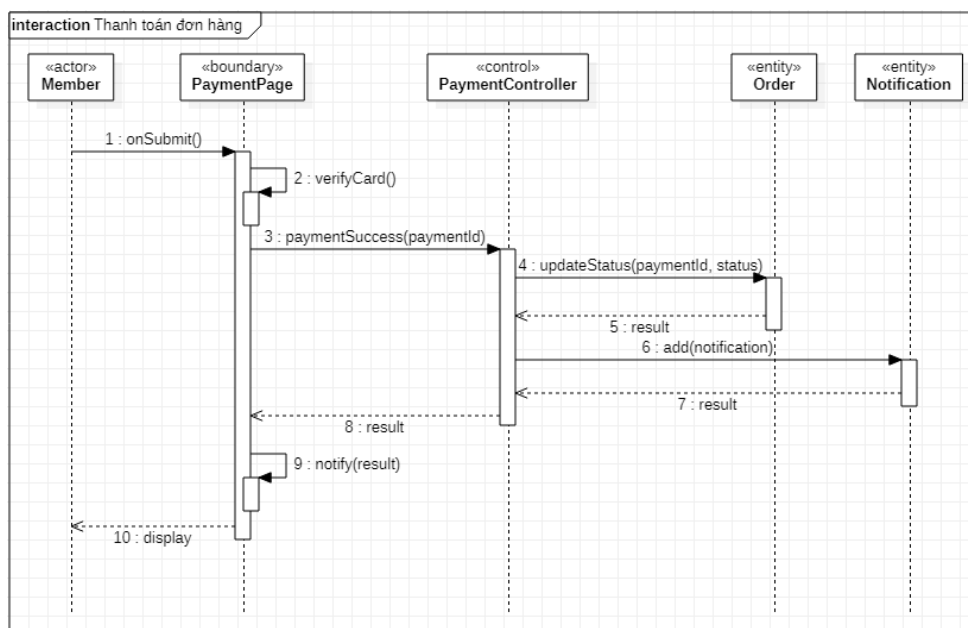
Hình 4.38 Biểu đồ trình tự use case Thêm sản phẩm vào giỏ

4.2.2.20. Biểu đồ trình tự use case Tạo đơn hàng



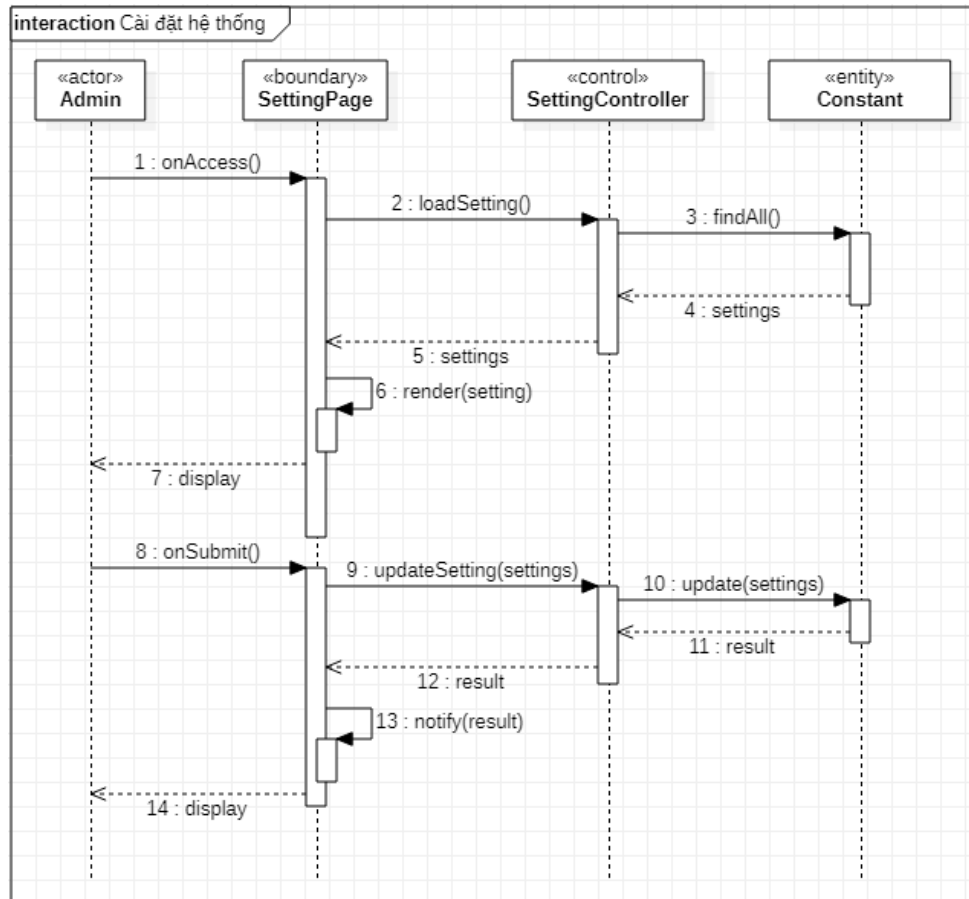
Hình 4.39 Biểu đồ trình tự use case Tạo đơn hàng

4.2.2.21. Biểu đồ trình tự use case Thanh toán đơn hàng



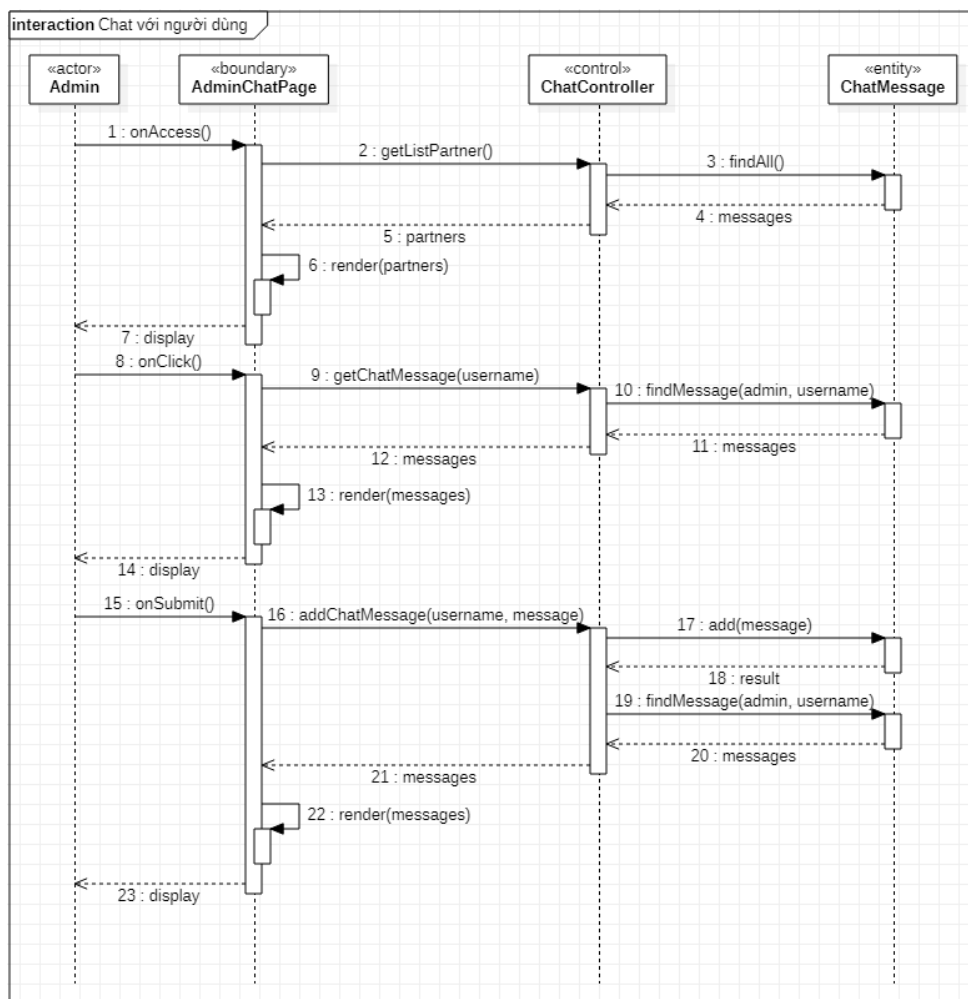
Hình 4.40 Biểu đồ trình tự use case Thanh toán đơn hàng

4.2.2.22. Biểu đồ trình tự use case Cài đặt hệ thống



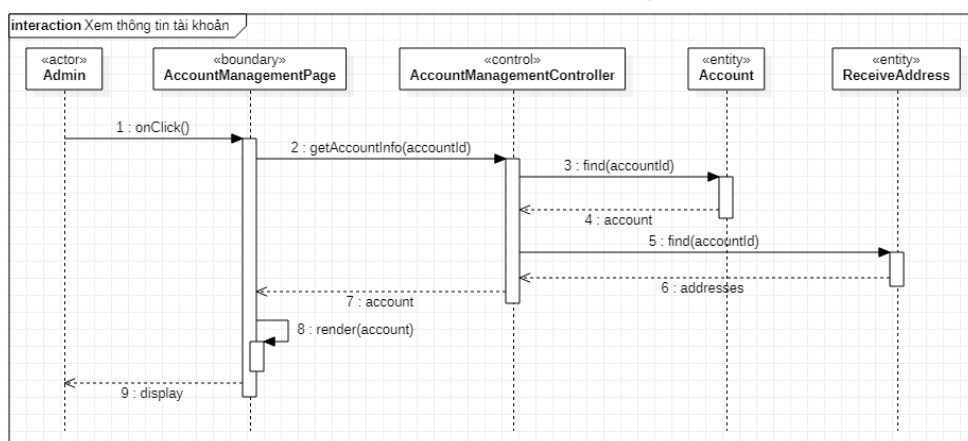
Hình 4.41 Biểu đồ trình tự use case Cài đặt hệ thống

4.2.2.23. Biểu đồ trình tự use case Chat với người dùng



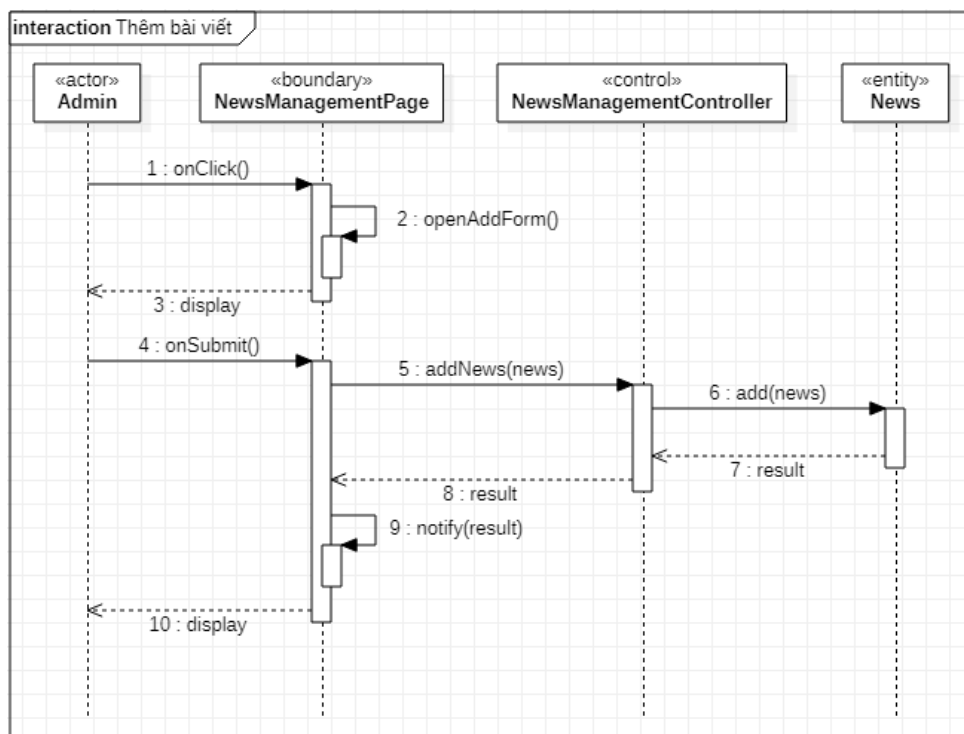
Hình 4.42 Biểu đồ trình tự use case Chat với người dùng

4.2.2.24. Biểu đồ trình tự use case Xem thông tin tài khoản



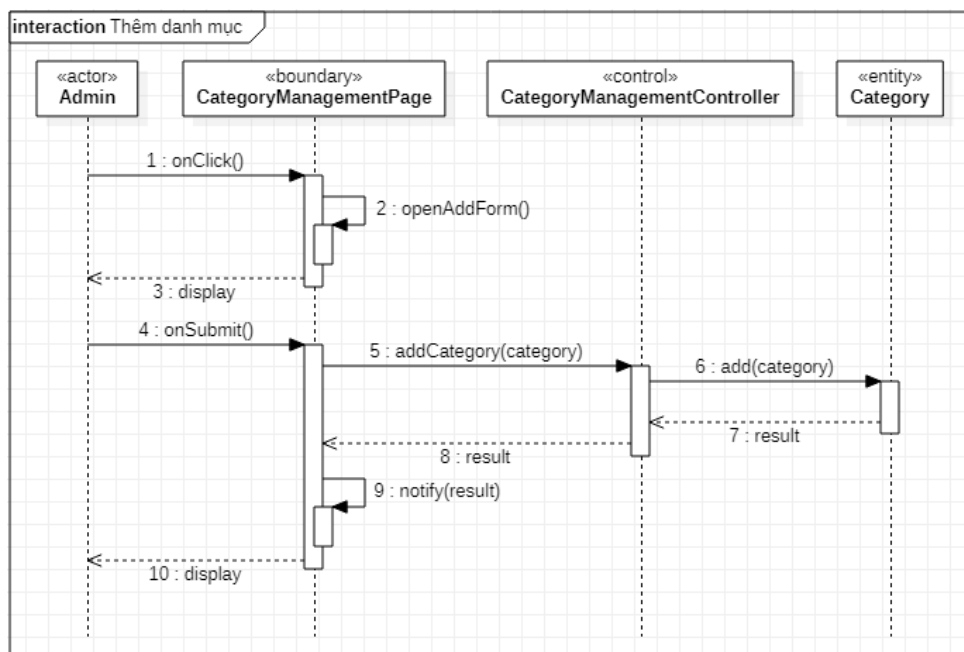
Hình 4.43 Biểu đồ trình tự use case Xem thông tin tài khoản

4.2.2.25. Biểu đồ trình tự use case Thêm bài viết



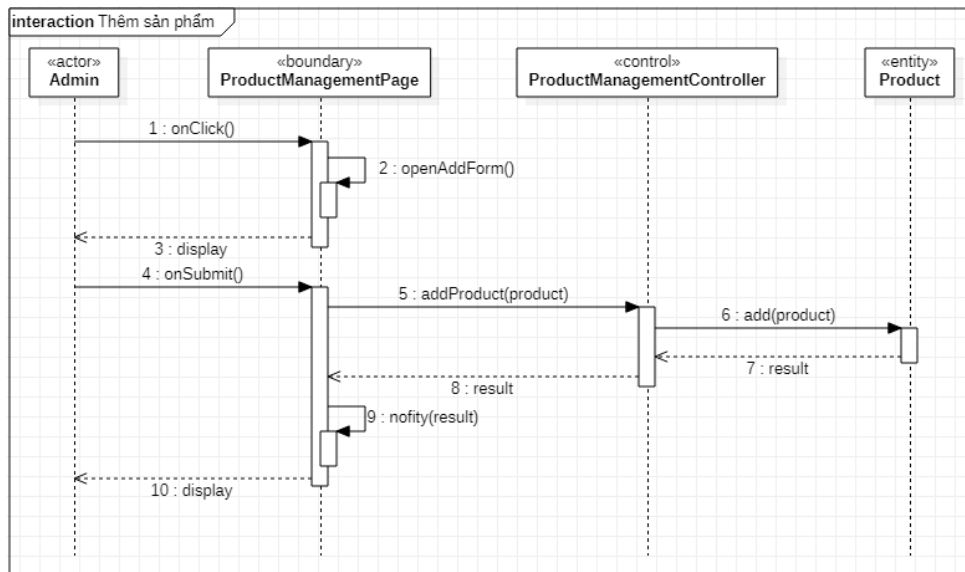
Hình 4.44 Biểu đồ trình tự use case Thêm bài viết

4.2.2.26. Biểu đồ trình tự use case Thêm danh mục



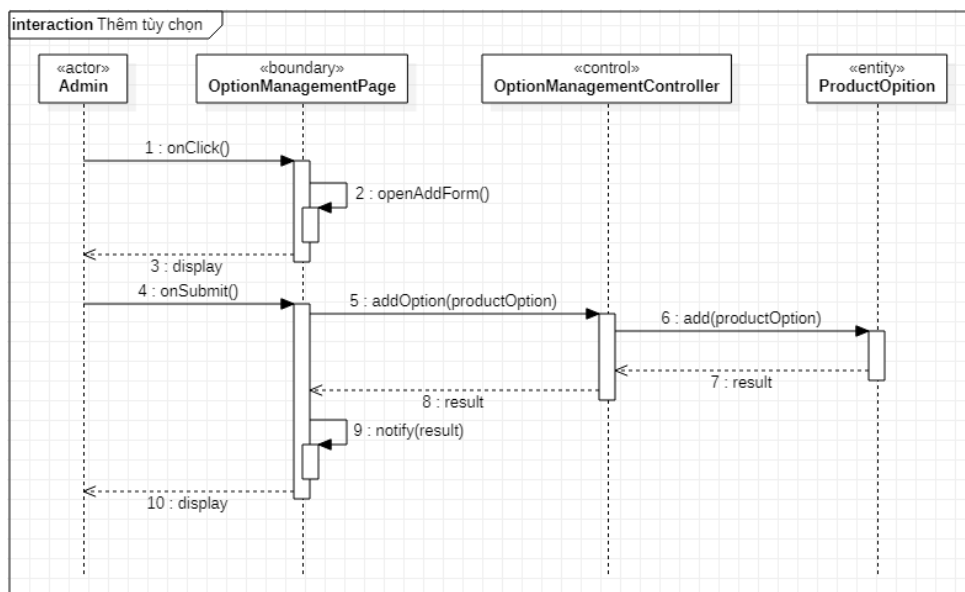
Hình 4.45 Biểu đồ trình tự use case Thêm danh mục

4.2.2.27. Biểu đồ trình tự use case Thêm sản phẩm



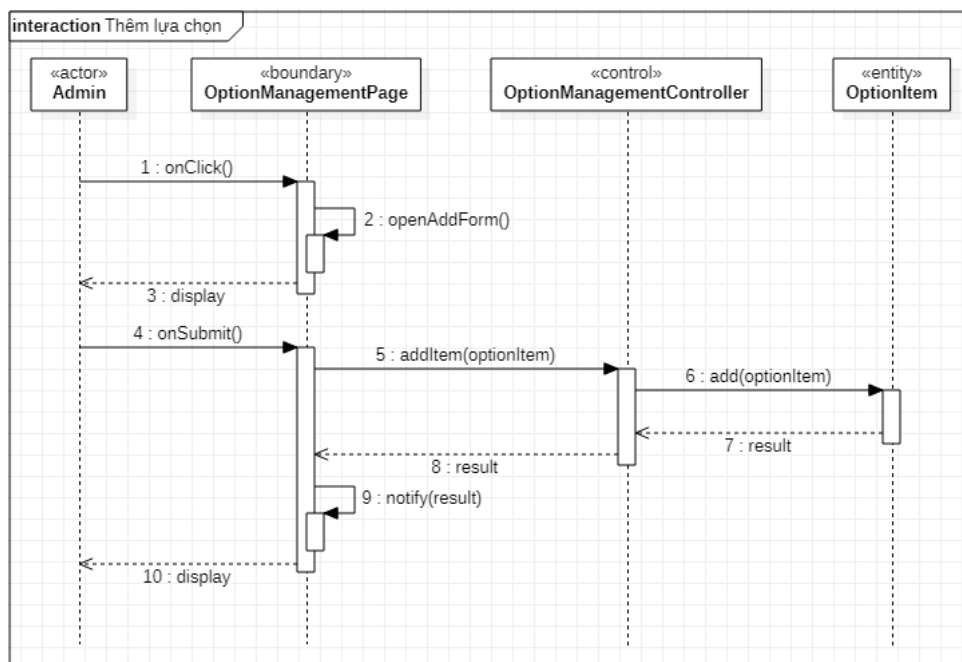
Hình 4.46 Biểu đồ trình tự use case Thêm sản phẩm

4.2.2.28. Biểu đồ trình tự use case Thêm tùy chọn



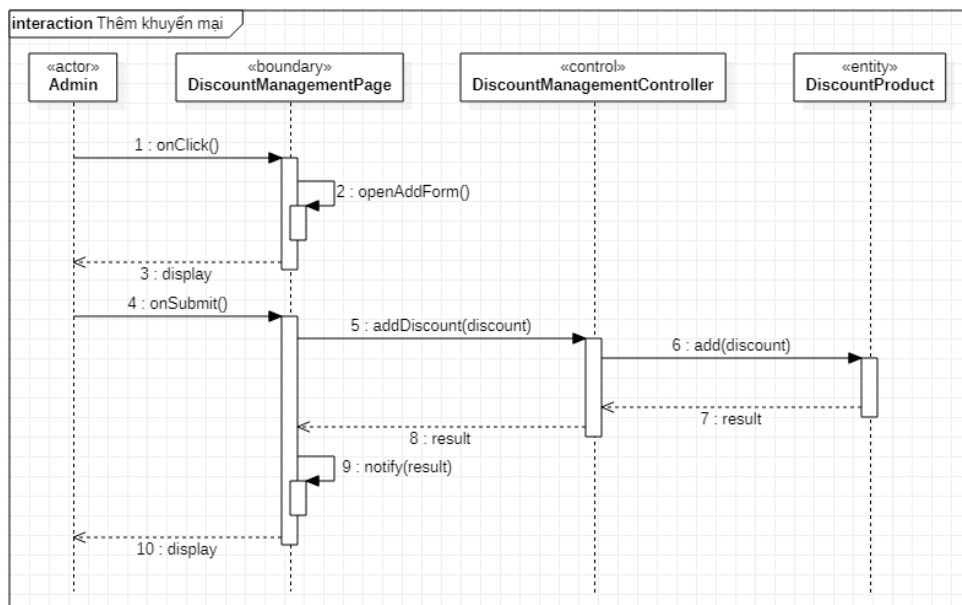
Hình 4.47 Biểu đồ trình tự use case Thêm tùy chọn

4.2.2.29. Biểu đồ trình tự use case Thêm lựa chọn



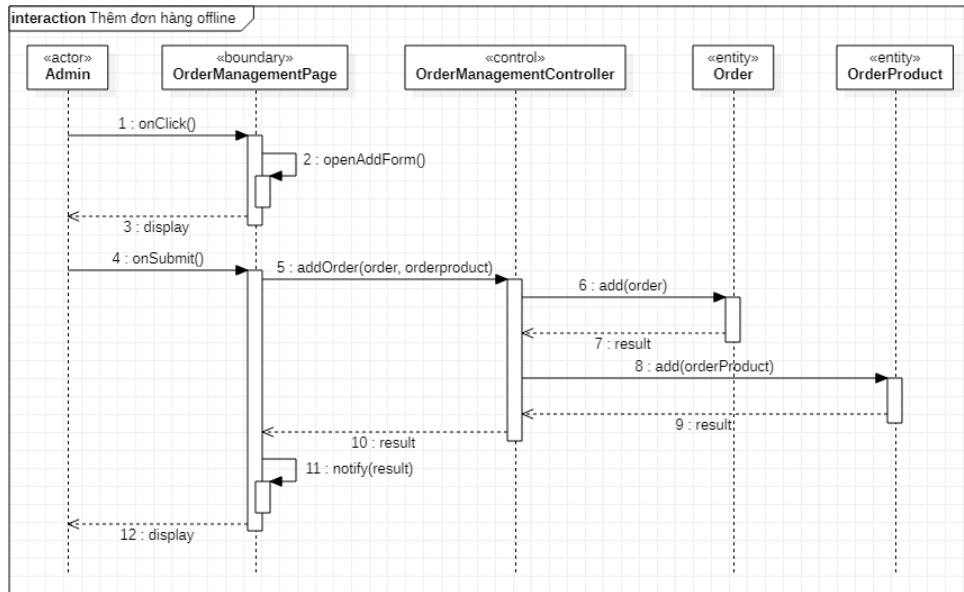
Hình 4.48 Biểu đồ trình tự use case Thêm lựa chọn

4.2.2.30. Biểu đồ trình tự use case Thêm khuyến mãi



Hình 4.49 Biểu đồ trình tự use case Thêm khuyến mãi

4.2.2.31. Biểu đồ trình tự use case Thêm đơn hàng



Hình 4.50 Biểu đồ trình tự use case Thêm đơn hàng

4.2.3 Thiết kế giao diện

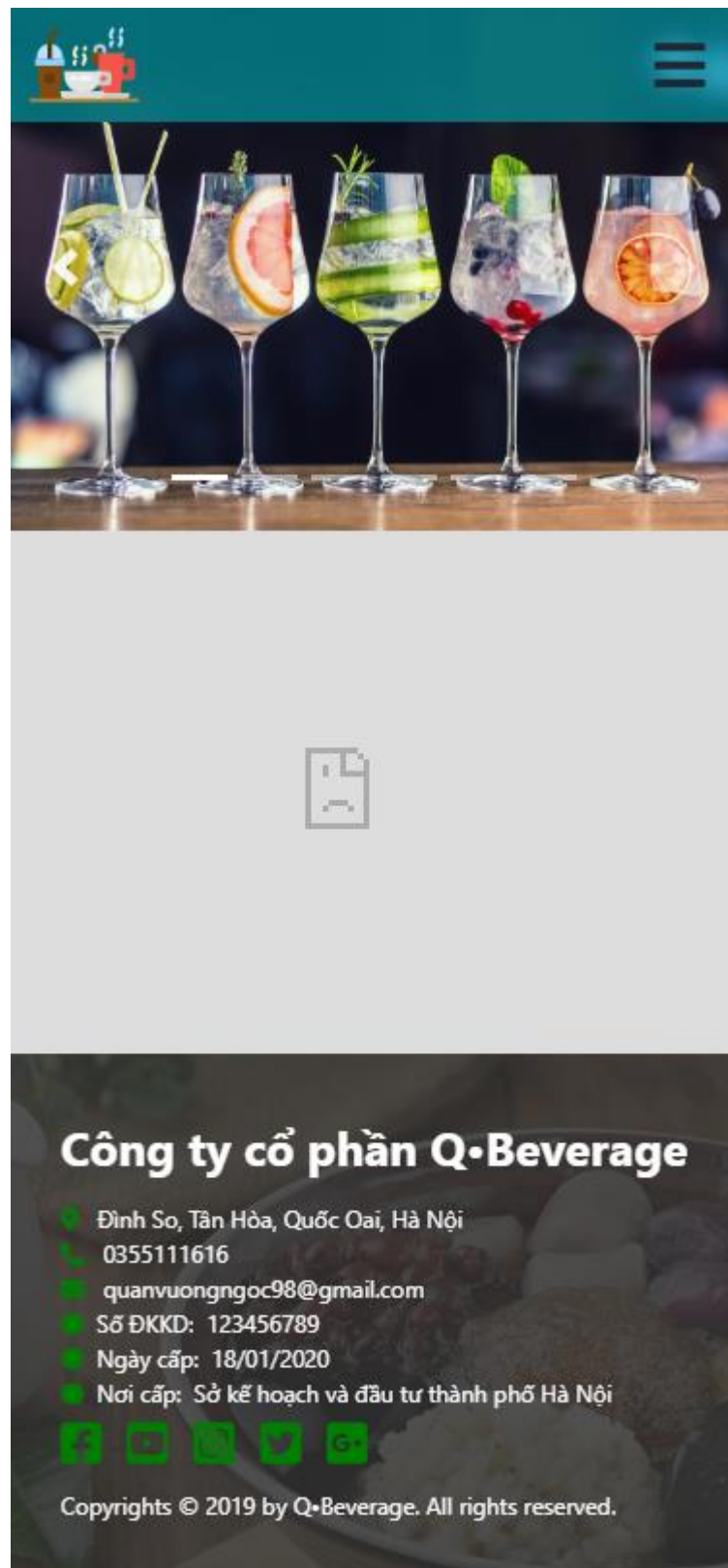
Sản phẩm của đồ án tốt nghiệp tuân thủ các thiết kế về giao diện như sau:

- Màn hình hỗ trợ: hiển thị tốt trên màn hình máy tính laptop và máy tính để bàn (độ rộng $\geq 992\text{px}$), đối với các trang dành cho Khách và Thành viên, hỗ trợ thêm cho màn hình điện thoại thông minh chế độ xoay dọc (độ rộng $< 576\text{px}$)
- Các nút điều khiển bên trong các form: sử dụng thống nhất button primary của Bootstrap framework
- Các nút điều khiển trong các component UI: sử dụng 1 icon đại diện và kèm theo title, không có chữ
- Vị trí hộp thoại tin nhắn của Người dùng: góc dưới bên phải màn hình
- Vị trí thông điệp phản hồi: góc trên bên phải màn hình
- Các trang quản lý hệ thống dành cho Quản trị viên: hiển thị dữ liệu sử dụng thư viện Datatable
- Các hành động thêm sửa xóa dữ liệu: mở và thực hiện trên Bootstrap Modal
- Màu sắc cho trang web: sử dụng font sáng chữ tối, mang phong cách outline style

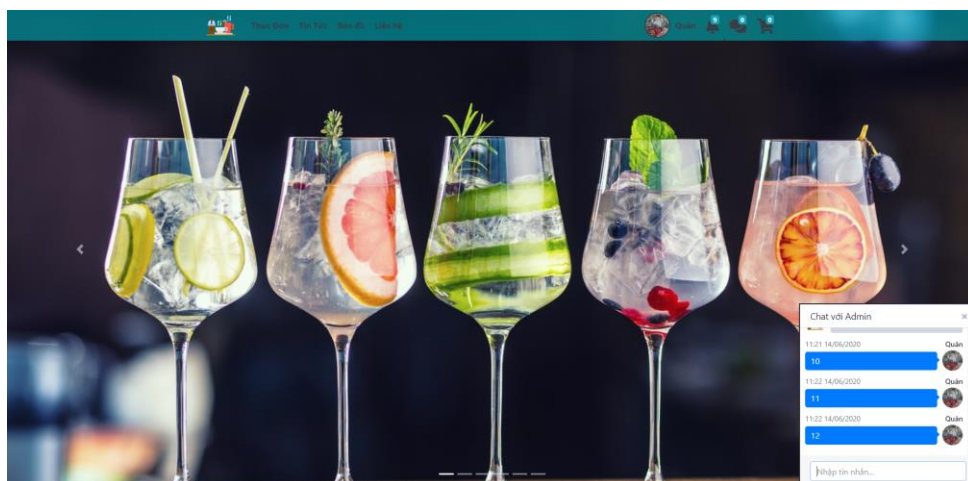
Sau đây là một số hình ảnh giao diện thiết kế dựa theo tài liệu thiết kế trên:



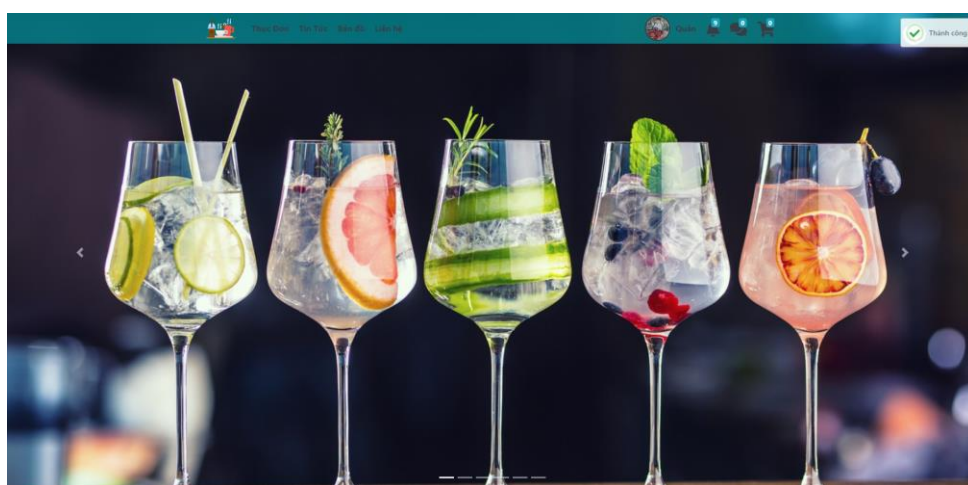
Hình 4.51 Thiết kế giao diện (màn hình máy tính)



Hình 4.52 Thiết kế giao diện (màn hình điện thoại)



Hình 4.53 Thiết kế giao diện (vị trí hộp thoại)



Hình 4.54 Thiết kế giao diện (vị trí thông điệp)

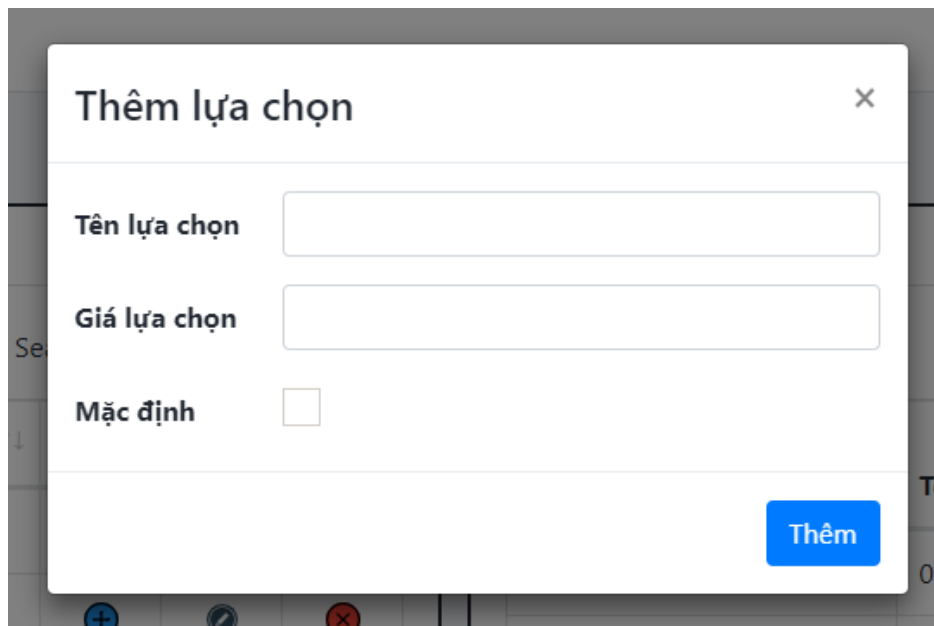
Tùy chọn

Show
10
entries
Search:

Tên tùy chọn	Loại tùy chọn			
Đá	SINGLE	+	⌕	×
Đường	SINGLE	+	⌕	×
Topping	MULTIPLE	+	⌕	×

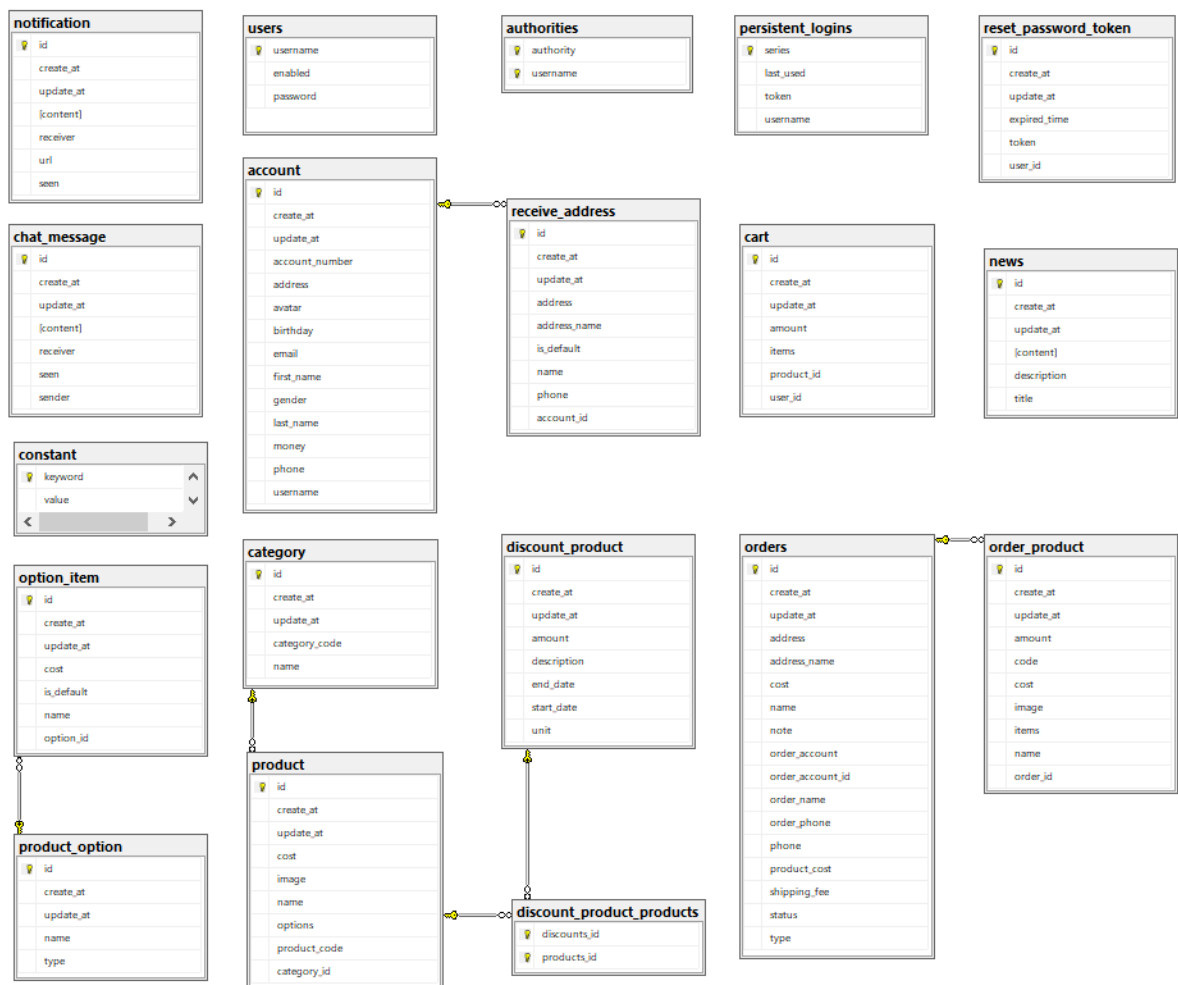
Showing 1 to 3 of 3 entries
Previous
1
Next

Hình 4.55 Thiết kế giao diện (Datatable)



Hình 4.56 Thiết kế giao diện (Bootstrap Modal)

4.2.4 Thiết kế cơ sở dữ liệu



Hình 4.57 Thiết kế cơ sở dữ liệu

4.3 Xây dựng ứng dụng

4.3.1 Thư viện và công cụ sử dụng

Bảng 4.7 Thư viện và công cụ sử dụng

Thư viện & công cụ	Mục đích
Java 8	Ngôn ngữ lập trình
HTML 5	Ngôn ngữ lập trình
CSS 3	Ngôn ngữ lập trình
Javascript ES6	Ngôn ngữ lập trình
Apache Maven 3.6.3	Công cụ quản lý dự án
Git 2.25.0	Công cụ quản lý phiên bản mã nguồn
Spring Framework 5.2.7	Làm cốt lõi, nền tảng cho một dự án Spring
Spring Boot 2.2.5	Giúp khởi tạo, cấu hình dự án Spring dễ dàng
Spring Security 5.3.2	Giúp xác thực và phân quyền cho dự án Spring
Spring Data JPA 2.3.1	Hỗ trợ truy vấn dữ liệu cho dự án Spring
Hibernate 5.4	Framework sử dụng làm Persistence Provider cho Spring Data JPA
Thymeleaf 3.0	View Technology cho dự án Spring
Jquery 3.4.1	Thư viện javascript
Jquery UI 1.12.1	Plugin jquery
Popper.js 1.16.0	Thư viện hỗ trợ Bootstrap
Switchery 0.8.2	Thư viện javascript
Bootstrap 4	Framework cho HTML CSS Javascript
SweetAlert2 8.19.0	Thư viện javascript
OverlayScrollbar 1.10.0	Thư viện javascript
MomentJS	Thư viện javascript
DateRangePicker 3.0.5	Thư viện javascript
Jquery InputMask 4.0.9	Thư viện javascript
Jquery Datatable	Thư viện javascript
Summernote 0.8.12	Thư viện javascript
BsCustomFileInput 1.4.2	Thư viện javascript
SortableJs 1.10.2	Thư viện javascript
SockJs 1.4.0	Thư viện javascript

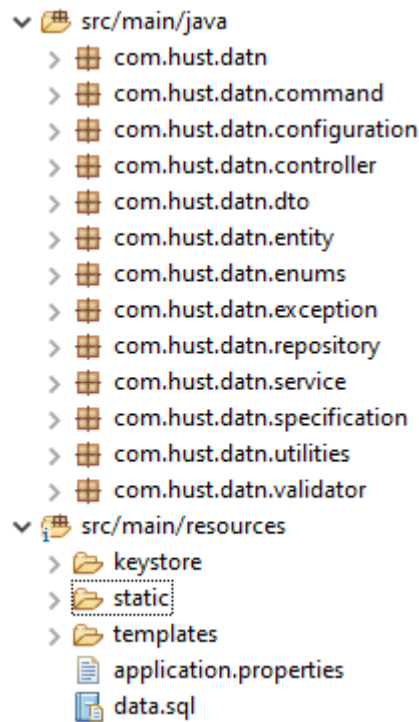
StompJs 5.4.4	Thư viện javascript
Eclipse IDE 2020	IDE lập trình java
Visual Studio Code 1.46.0	Công cụ lập trình
Microsoft SQL Server Management Studio 18	Phần mềm quản lý cơ sở dữ liệu SQL
SourceTree 3.3.8	Phần mềm quản lý phiên bản mã nguồn dựa trên nền Git
Stripe API 19.23.0	API Stripe giúp thanh toán trực tuyến

4.3.2 Kết quả đạt được

.git	2020-06-15 23:46	File folder	
.mvn	2020-03-14 15:22	File folder	
.settings	2020-03-14 22:23	File folder	
src	2020-03-18 19:51	File folder	
target	2020-06-09 17:32	File folder	
.classpath	2020-03-14 22:23	CLASSPATH File	2 KB
.gitignore	2020-03-18 19:51	Text Document	1 KB
.project	2020-03-14 22:23	PROJECT File	1 KB
HELP.md	2020-03-14 15:22	MD File	1 KB
mvnw	2020-03-14 15:22	File	10 KB
mvnw.cmd	2020-03-14 15:22	Windows Comma...	7 KB
pom.xml	2020-06-14 13:51	XML Document	3 KB
README.md	2020-03-15 23:31	MD File	1 KB

Hình 4.58 Cấu trúc thư mục dự án

Thư mục dự án bao gồm các thư mục con và tệp giúp quản lý dự án của Git, Maven, Eclipse, file README, HELP và quan trọng nhất là thư mục chứa mã nguồn dự án: src



Hình 4.59 Cấu trúc thư mục src

Thư mục src/main/java: chứa toàn bộ mã nguồn java dự án, bao gồm 12 package, 113 file java (4 enum, 3 anotation, 15 interface, 91 class dung lượng 188 KB)

Thư mục src/main/resources/keystore: chứa file cấu hình https cho dự án

Thư mục src/main/resources/static: chứa mã nguồn CSS, javascript, ảnh dùng trong dự án gồm cả từ thư viện và mã tự viết (20 file CSS dung lượng 26.6KB, 26 file javascript dung lượng 53.9KB)

Thư mục src/main/resources/templates: chứa mã nguồn html của dự án (54 file dung lượng 206KB)

Tệp application.properties: file cấu hình dự án Spring Boot


Tệp data.sql: chứa mã sql khởi tạo dữ liệu cho dự án Hibernate


4.3.3 Minh họa chức năng

Sau đây là một vài chức năng của sản phẩm được minh họa bằng hình ảnh:

Q•Beverage

Quý khách vui lòng đăng nhập





☐ Ghi nhớ đăng nhập

Đăng nhập

[Quên mật khẩu](#)

[Đăng ký tài khoản](#)

Hình 4.60 Chức năng Đăng nhập

Danh mục sản phẩm

Bộ lọc

Tìm kiếm

Danh mục

☒ Trà sữa



☒ Nước hoa quả



Khuyến mãi

☒ Tất cả sản phẩm



☐ Sản phẩm khuyến mãi



Trà sữa

 Hồng trà - PRD01
20000 đ
15000 đ

 Trà sữa - PRD02
30000 đ
25000 đ

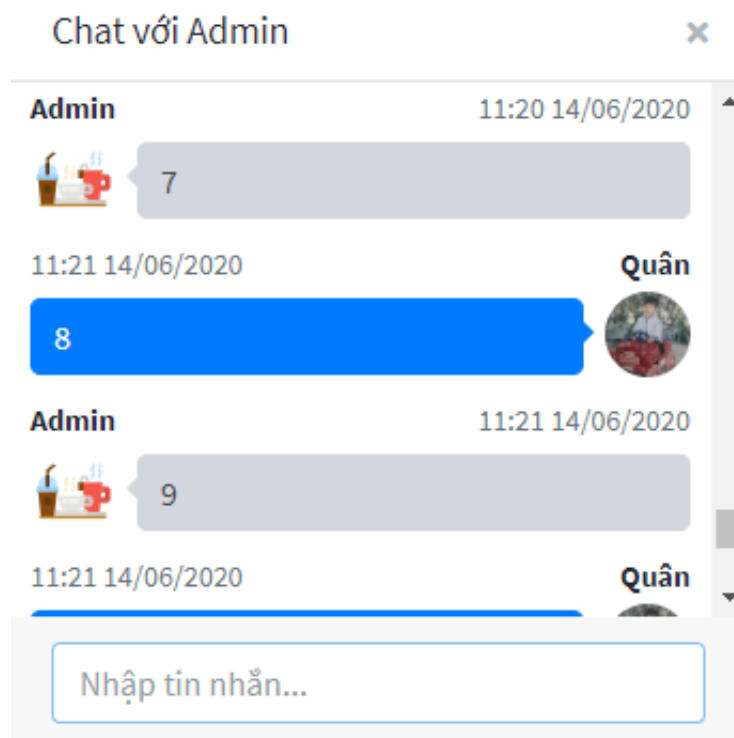
Nước hoa quả

 Nước dừa - PRD04
15000 đ
10000 đ

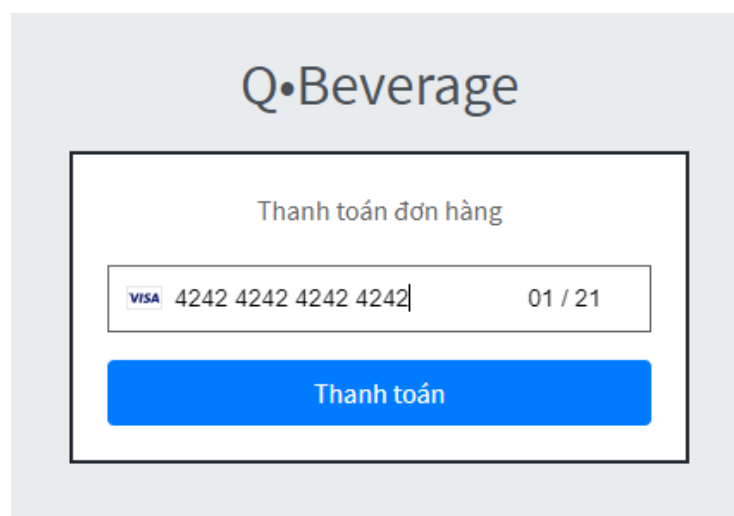
 Nước cam - PRD03
25000 đ
20000 đ

Hình 4.61 Chức năng Tìm kiếm sản phẩm

58



Hình 4.62 Chức năng Chat với quản trị viên



Hình 4.63 Chức năng Thanh toán

Thêm tin tức


Tiêu đề

Mô tả ngắn

Nội dung

Source Sans Pro

Đây là hình ảnh buổi lễ



Buổi lễ đã thành công tốt đẹp!!!

[Thêm](#)

Hình 4.64 Chức năng Thêm bài viết

Quản lý tùy chọn

Tùy chọn

Show 20 entries

Search:

Tên tùy chọn	Loại tùy chọn			
Đá	SINGLE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Đường	SINGLE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Topping	MULTIPLE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Showing 1 to 3 of 3 entries

Previous [1](#) Next

Lựa chọn

Show 20 entries

Search:

Tên tùy chọn	Tên lựa chọn	Giá lựa chọn	Mặc định			
Đá	0%	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Đá	75%	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Đá	50%	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Đá	25%	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Đá	100%	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Đường	100%	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Đường	0%	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Đường	75%	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Đường	50%	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Đường	25%	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Topping	Chân trâu	5000		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Topping	Thạch	5000		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Topping	Rau câu	5000		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Showing 1 to 13 of 13 entries

Previous [1](#) Next

Hình 4.65 Chức năng Quản lý tùy chọn


4.4 Kiểm thử


Sau đây là một vài trường hợp kiểm thử được thử nghiệm trên sản phẩm và minh họa bằng hình ảnh:

Q•Beverage

Quý khách vui lòng đăng nhập

Tài khoản hoặc mật khẩu không đúng





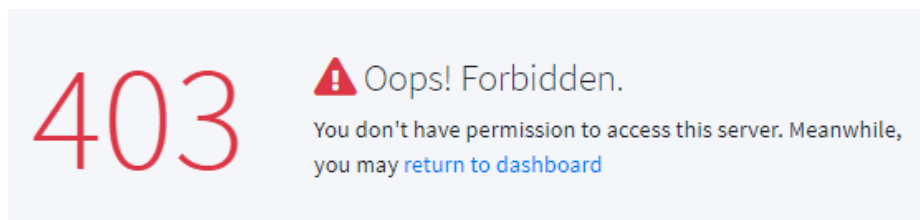
☐ **Ghi nhớ đăng nhập**

Đăng nhập

[Quên mật khẩu](#)

[Đăng ký tài khoản](#)

Hình 4.66 Kiểm thử đăng nhập thất bại



Hình 4.67 Kiểm thử phân quyền

❌ Tài khoản ít nhất 8 ký tự (chữ, số, gạch dưới)

Q•Beverage

Đăng ký tài khoản

👤

👤

📅

☒ Nam ☐ Nữ

📞

✉️

📍

👤

🔒

🔒

Đăng ký

[Tối trang đăng nhập](#)

Hình 4.68 Kiểm thử đăng ký thất bại

❌ Số lượng địa chỉ đã đạt tối đa

Địa chỉ nhận hàng +

Nhà Quấn
✓
✎
✕

Tên	Anh Quấn
SĐT	0355111616
Địa chỉ	Hà Nội

Test
✓
✎
✕

Tên	Test
SĐT	0123456789
Địa chỉ	Test


Nhà Vĩ
✓
✎
✕

Tên	Chị Vĩ
SĐT	0911632292
Địa chỉ	Hà Nam

Test 1
✓
✎
✕

Tên	Test
SĐT	0123456789
Địa chỉ	Test

Hình 4.69 Kiểm thử Thêm địa chỉ nhận hàng vượt mức



Sai định dạng (jpg hoặc png)

Vương Ngọc Quân

quanvn98 - 209062020

Giới tính	Nam
------------------	-----

Ngày sinh	18/01/1998
------------------	------------

Địa chỉ	Hà Nội
----------------	--------

SĐT	0355111616
------------	------------


Email	quanvuongngoc98@gmail.com
--------------	---------------------------

Hình 4.70 Kiểm thử Cập nhật thông tin sai định dạng ảnh

Không có sản phẩm nào trong giỏ hàng

Tổng tiền: 0 đ

Xem trước đơn hàng

 Không có sản phẩm trong giỏ hàng

Hình 4.71 Kiểm thử Tạo đơn hàng trống

CHƯƠNG 5. KẾT LUẬN

5.1 Kết luận

Như vậy quá trình thực hiện đồ án “Xây dựng hệ thống bán hàng trực tuyến dành cho các cửa hàng đồ uống” của em đã kết thúc. Sản phẩm được tạo ra là một trang web bán hàng online dựa trên ngôn ngữ Java và Spring framework. Trang web đã đáp ứng đầy đủ được các yêu cầu của đề tài, đó là cung cấp khả năng bán hàng và thanh toán dành cho các cửa hàng quy mô nhỏ có nhu cầu tích hợp mô hình kinh doanh trực tuyến với giao diện đẹp mắt, bao gồm đầy đủ các chức năng cơ bản về mặt quản lý cửa hàng, hỗ trợ nghiệp vụ mua bán, trò chuyện thời gian thực, thông báo đẩy, thanh toán trực tuyến và nhiều chức năng khác... Ngoài ra hệ thống cũng đảm bảo được tính chính xác trong nghiệp vụ, nơi mà mô hình bán hàng truyền thống có thể xảy ra một số lỗi thủ công. Từ những kết quả đạt được như trên, sản phẩm đã có thể đủ điều kiện và độ tin cậy để áp dụng vào trong thực tế, mang tính khả thi cao và chắc chắn sẽ mang lại được những lợi ích cho các khách hàng có nhu cầu sử dụng sản phẩm, bao gồm cả các đối tượng bên bán và bên mua.

Ngoài những lợi ích đem lại cho khách hàng như trên, đồ án còn giúp em học hỏi được rất nhiều trong quá trình thực hiện. Hiểu rõ hơn về cách thức hoạt động của một trang web, luồng truyền dữ liệu request response, nắm được mô hình MVC trong kiến trúc xây dựng phần mềm. Cùng với đó là các kiến thức về ngôn ngữ Java, Javascript, framework Spring, Hibernate, Apache Maven, SQL database, và cả các ngôn ngữ lập trình giao diện người dùng (HTML, CSS), quản lý phiên bản mã nguồn (Git), kỹ năng debug và kỹ năng phân tích thiết kế hệ thống hướng đối tượng... Tiếp theo là nâng cao khả năng đọc hiểu Tiếng Anh qua các tài liệu nước ngoài, kỹ năng tìm kiếm tài liệu, chọn lọc thông tin, viết báo cáo, quản lý thời gian cá nhân... Đồ án tốt nghiệp của em sau khi hoàn thành chắc chắn vẫn còn tồn tại nhiều điểm hạn chế cần khắc phục, những chức năng cần được cải thiện, và những dự định đặt ra nhưng chưa thể hoàn thành được, đó cũng chính là yếu tố để em nhìn nhận được điểm yếu, điểm chưa được đạt được trong kỹ năng lập trình, kỹ năng mềm của bản thân và có thêm những kinh nghiệm quý báu trong quá trình học tập và làm việc sau này.

5.2 Hướng phát triển của đồ án trong tương lai

Trong tương lai gần, em sẽ vẫn còn tiếp tục nghiên cứu và phát triển đề tài trong đồ án. Mục tiêu cuối cùng là làm cho sản phẩm được đưa vào thực tế sử dụng, hoạt động tốt với những chức năng hiện tại mà nó cung cấp và làm hài lòng tất cả các khách hàng. Cùng với đó là từng bước nâng cấp các tính năng còn sơ sài như: chức năng thông báo đẩy, chức năng trò chuyện thời gian thực (có thêm icon và gắn thẻ), tạo các mẫu báo cáo thống kê và hóa đơn để in ra giấy, cung cấp các chương trình khuyến mãi đơn hàng... Sau đó, mở rộng quy mô và các tính năng để trở thành một hệ thống tốt hơn, không chỉ phục vụ những cửa hàng quy mô nhỏ mà còn quy mô lớn, hoặc thậm chí là các cửa hàng với sản phẩm kinh doanh khác.

TÀI LIỆU THAM KHẢO

- [1] "HTML 5.3," World Wide Web Consortium, [Online]. Available: <https://www.w3.org/TR/html53/>.
- [2] "CSS Snapshot 2018," World Wide Web Consortium, [Online]. Available: <https://www.w3.org/TR/css-2018/>.
- [3] "ECMAScript® 2015 Language Specification," Ecma International, [Online]. Available: <http://www.ecma-international.org/ecma-262/6.0/>.
- [4] "Java™ Platform, Standard Edition 8," Oracle, [Online]. Available: <https://docs.oracle.com/javase/8/docs/api/>.
- [5] "RFC 2616 - Hypertext Transfer Protocol -- HTTP/1.1," IETF, [Online]. Available: <https://tools.ietf.org/html/rfc2616>.