

ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN TỐT NGHIỆP

NGÀNH: CÔNG NGHỆ THÔNG TIN
CHUYÊN NGÀNH: CÔNG NGHỆ PHẦN MỀM

ĐỀ TÀI:

XÂY DỰNG WEBSITE THƯƠNG MẠI ĐIỆN TỬ THEO KIẾN TRÚC MICROSERVICES

Người hướng dẫn: **TS. NGUYỄN NĂNG HÙNG VÂN**

Sinh viên thực hiện: **LÊ THANH VIỆT**

Số thẻ sinh viên: 102210287

Lớp: **21TCLC_DT4**

Đà Nẵng, 06/2025 ĐẠI HỌC ĐÀ NẴNG

**ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN**

ĐỒ ÁN TỐT NGHIỆP

**NGÀNH: CÔNG NGHỆ THÔNG TIN
CHUYÊN NGÀNH: CÔNG NGHỆ PHẦN MỀM**

ĐỀ TÀI:

XÂY DỰNG WEBSITE THƯƠNG MẠI ĐIỆN TỬ THEO KIẾN TRÚC MICROSERVICES

Người hướng dẫn: TS. NGUYỄN NĂNG HÙNG VÂN

Sinh viên thực hiện: LÊ THANH VIỆT

Số thẻ sinh viên: 102210287

Lớp: 21TCLC_DT4

Đà Nẵng, 06/2025

NHẬN XÉT ĐỒ ÁN TỐT NGHIỆP

I. Thông tin chung:

- Họ và tên sinh viên: Lê Thanh Việt
- Lớp: 21TCLC_DT4 Số thẻ SV: 102210287
- Tên đề tài: Xây dựng website thương mại điện tử theo kiến trúc Microservices
- Người hướng dẫn: Nguyễn Năng Hùng Vân. Học hàm/ học vị: Tiến sĩ

II. Nhận xét, đánh giá đồ án tốt nghiệp:

- Về tính cấp thiết, tính mới, khả năng ứng dụng của đề tài:

Đề tài "Xây dựng website thương mại điện tử theo kiến trúc Microservices" là rất cấp thiết trong bối cảnh thương mại điện tử phát triển mạnh mẽ và nhu cầu về các hệ thống có khả năng mở rộng, linh hoạt và chịu tải cao ngày càng tăng. Kiến trúc Microservices là một xu hướng kiến trúc hiện đại, giải quyết các thách thức của các hệ thống nguyên khối (monolithic) truyền thống. Việc áp dụng kiến trúc Microservices cho website thương mại điện tử là một hướng tiếp cận mới mẻ và tiên tiến so với các giải pháp truyền thống tại nhiều doanh nghiệp vừa và nhỏ. Đặc biệt, việc tích hợp các công nghệ AI như Chatbot dựa trên RAG và hệ thống gợi ý sản phẩm sử dụng embedding và RRF thể hiện tính đón đầu công nghệ và sáng tạo trong giải pháp. Hệ thống có khả năng ứng dụng cao trong thực tế, không chỉ cho các doanh nghiệp thương mại điện tử mà còn là một mô hình tham khảo cho việc phát triển các ứng dụng phân tán khác. Các module như chatbot và gợi ý sản phẩm trực tiếp nâng cao trải nghiệm người dùng, mang lại giá trị kinh doanh rõ ràng.

- Về kết quả giải quyết các nội dung nhiệm vụ yêu cầu của đồ án:

Các nhiệm vụ được đưa ra đa phần đều đã được thực hiện đầy đủ và đạt yêu cầu, cụ thể kết quả đạt được:

- Về mặt lí thuyết, sinh viên đã thành công trong việc áp dụng rõ ràng kiến trúc Microservices thông qua việc chia tách các dịch vụ và nghiên cứu sử dụng các công nghệ liên quan như Spring Cloud, Kafka cho giao tiếp và quản lý dịch vụ. Về mặt AI, đồ án đã nghiên cứu và tích hợp Chatbot AI, xây dựng hệ thống gợi ý sản phẩm sử dụng kiến trúc RAG với các mô hình và công cụ như Gemini AI, ChromaDB và Sentence Transformers.

- Về mặt thực tiễn, sinh viên đã thiết kế và triển khai thành công ứng dụng với giao diện trực quan, xây dựng theo kiến trúc hiện đại, đầy đủ các chức năng, nâng cao trải nghiệm của người dùng với chatbot và hệ thống gợi ý sản phẩm.

3. Về hình thức, cấu trúc, bố cục của đồ án tốt nghiệp:

Hợp lí, đảm bảo đúng yêu cầu

4. Đề tài có giá trị khoa học/ có bài báo/ giải quyết vấn đề đặt ra của doanh nghiệp hoặc nhà trường:

Đề tài đã giải quyết vấn đề cấp thiết của các website thương mại điện tử hiện đại về khả năng mở rộng, linh hoạt và cung cấp trải nghiệm cá nhân hóa, tự động hóa thông qua chatbot và hệ thống gợi ý.

5. Các tồn tại, thiếu sót cần bổ sung, chỉnh sửa:

Thời gian phản hồi chậm cho các truy vấn phức tạp, ảnh hưởng đến trải nghiệm người dùng trong một số trường hợp. Cần tối ưu tốc độ xử lý và cần bổ sung thêm dữ liệu huấn luyện lớn hơn và phong phú hơn .

III. Tinh thần, thái độ làm việc của sinh viên:

Sinh viên có thái độ làm việc nghiêm túc, biết tìm tòi tài liệu chuyên ngành. Trong quá trình thực hiện, sinh viên luôn bám sát tiến độ, thường xuyên trao đổi với giáo viên hướng dẫn. Đề tài được hoàn thành đúng thời hạn.

IV. Đánh giá:

1. Điểm đánh giá:/10 (lấy đến 1 số lẻ thập phân)

2. Đề nghị: Được bảo vệ đồ án Bổ sung để bảo vệ Không được bảo vệ

Đà Nẵng, ngày 18 tháng 06 năm 2025

Người hướng dẫn

TS. Nguyễn Năng Hùng Vân

TÓM TẮT

Tên đề tài: Xây dựng website bán hàng theo kiến trúc Microservices

Sinh viên thực hiện: Lê Thanh Việt

Số thẻ SV: 102210287

Lớp: 21TCLC-DT4

Trong bối cảnh thương mại điện tử ngày càng phát triển mạnh mẽ, việc xây dựng các hệ thống website bán hàng hiện đại, linh hoạt và dễ mở rộng trở nên vô cùng cần thiết. Đề tài “Xây dựng website bán hàng theo kiến trúc Microservices” nhằm mục tiêu thiết kế và triển khai một hệ thống thương mại điện tử áp dụng kiến trúc Microservices, thay vì kiến trúc đơn thể truyền thống.

Mục tiêu cốt lõi của đề tài là thiết kế và triển khai một nền tảng thương mại điện tử theo nguyên lý Microservices, trong đó hệ thống được phân tách một cách rõ ràng thành tập hợp các dịch vụ nhỏ, độc lập, có thể triển khai và quản lý riêng biệt. Cụ thể, hệ thống dự kiến bao gồm các dịch vụ chuyên biệt như: product service chịu trách nhiệm về thông tin sản phẩm; identity service quản lý tài khoản, thông tin cá nhân; cart service xử lý việc thêm/xóa sản phẩm vào giỏ; order service quản lý quy trình đặt hàng; payment service tích hợp các cổng thanh toán; AI service giúp hỗ trợ trả lời câu hỏi của khách hàng đồng thời gợi ý sản phẩm thích hợp cho người dùng.

Việc phân tách này mang lại nhiều lợi ích. Mỗi service có thể sử dụng công nghệ và ngôn ngữ phù hợp nhất cho từng tác vụ. Khả năng triển khai độc lập cho phép cập nhật, nâng cấp hoặc bảo trì từng dịch vụ mà không gây gián đoạn cho toàn bộ hệ thống, từ đó giảm thiểu rủi ro và tăng tốc độ phát triển. Hơn nữa, việc mở rộng quy mô trở nên hiệu quả hơn khi chỉ những dịch vụ có nhu cầu tài nguyên cao mới cần được tăng cường, tối ưu hóa việc sử dụng cơ sở hạ tầng. Tính linh hoạt này cũng góp phần tăng cường khả năng phục hồi của hệ thống; nếu một dịch vụ gặp lỗi, các dịch vụ khác vẫn có thể tiếp tục hoạt động, giảm thiểu tác động đến trải nghiệm người dùng.

Đề tài cũng tích hợp các công nghệ hiện đại như Docker và Kubernetes để triển khai và quản lý các container, Kafka để truyền thông giữa các dịch vụ, cùng với chatbot AI nhằm nâng cao trải nghiệm người dùng. Ngoài ra, hệ thống còn áp dụng cơ chế bảo mật với xác thực người dùng qua JWT, phân quyền truy cập theo vai trò.

Kết quả của đề tài là một nền tảng bán hàng trực tuyến có tính linh hoạt cao, dễ mở rộng, dễ bảo trì và có khả năng đáp ứng tốt với sự phát triển lâu dài của doanh nghiệp.

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Họ tên sinh viên: Lê Thanh Việt

Số thẻ sinh viên: 102210287

Lớp: 21TCLC_DT4 Khoa: Công Nghệ Thông Tin Ngành: Công Nghệ Phần Mềm

1. Tên đề tài đồ án:

Xây dựng website thương mại điện tử theo kiến trúc Microservices

2. Đề tài thuộc diện: Có ký kết thỏa thuận sở hữu trí tuệ đối với kết quả thực hiện

3. Các số liệu và dữ liệu ban đầu:

4. Nội dung các phần thuyết minh và tính toán:

MỞ ĐẦU: Trình bày lí do thực hiện đề tài; mục tiêu đề tài; phạm vi và đối tượng nghiên cứu; phương pháp nghiên cứu; cấu trúc đồ án.

Chương 1 - CƠ SỞ LÝ THUYẾT: Trình bày các khái niệm nền tảng như phương pháp, công nghệ và thuật toán được áp dụng để triển khai hệ thống.

Chương 2 - PHƯƠNG PHÁP ĐỀ XUẤT XÂY DỰNG CHATBOT VÀ HỆ THỐNG GỢI Ý SẢN PHẨM: Trình bày, mô tả phương pháp sử dụng trong việc xây dựng chatbot và hệ thống gợi ý sản phẩm cho người dùng.

Chương 3 - PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG: Trình bày phân tích và thiết kế trong xây dựng hệ thống và luồng hoạt động hệ thống

Chương 4 – XÂY DỰNG WEBSITE THƯƠNG MẠI ĐIỆN TỬ THEO KIẾN TRÚC MICROSERVICES: Trình bày cách thức cài đặt, phát triển chương trình và đánh giá kết quả đạt được.

KẾT LUẬN: Rút ra kết quả, những bài học đạt được qua quá trình xây dựng hệ thống, những điểm hạn chế và phương hướng phát triển trong tương lai.

TÀI LIỆU THAM KHẢO: Giới thiệu về chi tiết thông tin tài liệu đã được sử dụng.

5. Các bản vẽ, đồ thị (ghi rõ các loại và kích thước bản vẽ):

6. Họ tên người hướng dẫn: TS.Nguyễn Năng Hùng Vân

- Sơ đồ tuần tự

- Sơ đồ hoạt động

- Biểu đồ Use Case
- Sơ đồ ERD

7. Ngày giao nhiệm vụ đồ án: 21/03/2025

8. Ngày hoàn thành đồ án: 15/06./2025.

Đà Nẵng, ngày 15 tháng 06 năm 2025

Trưởng Bộ môn

Người hướng dẫn

LỜI NÓI ĐẦU

Với sự phát triển không ngừng của khoa học công nghệ, đặc biệt là trong lĩnh vực Công nghệ Thông tin, việc tiếp thu và vận dụng kiến thức chuyên sâu vào thực tiễn đã trở thành một yêu cầu thiết yếu đối với mỗi kỹ sư tương lai. Đồ án tốt nghiệp này là thành quả của một quá trình học tập, nghiên cứu và ứng dụng những tri thức đã được trang bị, đồng thời cũng là cơ hội để em tổng kết, củng cố và phát triển năng lực chuyên môn của bản thân.

Trước tiên, em xin gửi lời cảm ơn chân thành và sâu sắc đến quý thầy cô Khoa Công nghệ Thông tin, Trường Đại học Bách khoa – Đại học Đà Nẵng, những người đã tận tình giảng dạy, truyền đạt cho em những kiến thức chuyên môn quý báu, cũng như những kinh nghiệm thực tiễn trong suốt quá trình học tập và rèn luyện tại trường. Những kiến thức đó là nền tảng vững chắc để em có thể thực hiện và hoàn thành đồ án tốt nghiệp này.

Đặc biệt, em xin bày tỏ lòng biết ơn sâu sắc đến thầy Nguyễn Năng Hùng Vân – người đã trực tiếp hướng dẫn, tận tình giúp đỡ, định hướng và góp ý cho em trong suốt quá trình thực hiện đồ án. Những chỉ dẫn quý báu và sự hỗ trợ của thầy đã giúp em vượt qua những khó khăn, phát hiện và khắc phục những thiếu sót để hoàn thiện sản phẩm.

Mặc dù đã nỗ lực hoàn thành đồ án với tinh thần trách nhiệm cao nhất, nhưng do kiến thức và kinh nghiệm thực tế còn hạn chế, nên khó tránh khỏi những thiếu sót và hạn chế nhất định. Em rất mong nhận được sự góp ý của quý thầy cô để em có thể hoàn thiện sản phẩm một cách tốt hơn. Những nhận xét này sẽ là nguồn tư liệu vô cùng quý giá, giúp em tiếp tục học hỏi, cải thiện và hoàn thiện hơn nữa kiến thức cũng như kỹ năng của mình trong tương lai.

Một lần nữa, em xin chân thành cảm ơn !

CAM ĐOAN

Em xin cam đoan:

1. Những nội dung trong luận văn này là do em thực hiện với sự hướng dẫn của thầy Nguyễn Năng Hùng Vân
2. Những nội dung được trình bày trong luận văn này đều là những phần em đã thực hiện trong vai trò chính và đóng góp trực tiếp trong dự án.
3. Mọi tham khảo dùng trong đề án đều được trích dẫn rõ ràng tên tác giả, tên công trình, thời gian, địa điểm công bố.

Nếu có những sao chép không hợp lệ, vi phạm, em xin chịu hoàn toàn trách nhiệm.

Đà Nẵng, ngày 15 tháng 06 năm 2025

Sinh viên thực hiện

Lê Thanh Việt

MỤC LỤC

TÓM TẮT	
NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP	
LỜI NÓI ĐẦU	I
CAM ĐOAN	II
MỤC LỤC	III
DANH SÁCH BẢNG	VII
DANH SÁCH HÌNH VẼ	IX
DANH SÁCH CÁC CHỮ VIẾT TẮT	XIII
MỞ ĐẦU	1
CHƯƠNG 1: CƠ SỞ LÝ THUYẾT	5
1.1 KIẾN TRÚC MICROSERVICES	5
1.1.1 Định nghĩa.....	5
1.1.2 Vai trò.....	5
1.1.3 So sánh giữa kiến trúc Monolithic và Microservices.....	7
1.2 RETRIEVAL-AUGMENTED GENERATION (RAG)	9
1.2.1 Định nghĩa.....	9
1.2.2 Cách hoạt động của RAG.....	9
1.2.3 Các thành phần của RAG.....	10
1.2.4 Lợi ích của RAG.....	12
1.2.5 Ứng dụng của RAG	13
1.3 SENTENCE TRANSFORMERS	14
1.3.1 Định nghĩa.....	14
1.3.2 Vai trò của Sentence Transformers trong RAG	14
1.3.3 Kết luận	16
1.4 RECIPROCAL RANK FUSION (RRF).....	16
1.4.1 Định nghĩa.....	16
1.4.2 Công thức	16
1.4.3 Ưu điểm của RRF	17

1.4.4	<i>Kết luận</i>	18
CHƯƠNG 2: PHƯƠNG PHÁP ĐỀ XUẤT XÂY DỰNG CHATBOT VÀ HỆ THỐNG GỢI Ý SẢN PHẨM.....19		
2.1	CHATBOT	19
2.1.1	<i>Giới thiệu</i>	19
2.1.2	<i>Mô tả hệ thống</i>	21
2.1.3	<i>Thực nghiệm và đánh giá kết quả</i>	23
2.1.4	<i>Kết luận</i>	27
2.2	HỆ THỐNG GỢI Ý SẢN PHẨM	28
2.2.1	<i>Mô tả hệ thống</i>	28
2.2.2	<i>Thực nghiệm và đánh giá kết quả</i>	28
2.2.3	<i>Kết luận</i>	31
CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG32		
3.1	TÁC NHÂN	32
3.2	SƠ ĐỒ USE CASE HỆ THỐNG	34
3.2.1	<i>Admin xem thông tin cá nhân</i>	35
3.2.2	<i>Admin quản lí sản phẩm</i>	36
3.2.3	<i>Admin quản lí danh mục</i>	39
3.2.4	<i>Admin quản lí đơn hàng</i>	41
3.2.5	<i>Admin quản lí tồn kho</i>	43
3.2.6	<i>Admin quản lí nhân hàng</i>	45
3.2.7	<i>Admin quản lí user</i>	47
3.2.8	<i>User quản lí profile</i>	49
3.2.9	<i>User quản lý địa chỉ nhận hàng</i>	50
3.2.10	<i>User quản lý giỏ hàng</i>	51
3.2.11	<i>User quản lí đơn hàng</i>	53
3.2.12	<i>User chatbot</i>	54
3.2.13	<i>User hệ thống gợi ý sản phẩm</i>	55
3.3	BIỂU ĐỒ HOẠT ĐỘNG.....	56
3.3.1	<i>Chức năng đăng nhập</i>	56
3.3.2	<i>Chức năng đăng kí</i>	57
3.3.3	<i>Chức năng thêm sản phẩm vào giỏ hàng</i>	57
3.3.4	<i>Chức năng thêm địa chỉ cho người dùng</i>	58

3.3.5	Chức năng cập nhật thông tin địa chỉ.....	58
3.3.6	Chức năng quản lí thông tin cá nhân của người dùng.....	59
3.3.7	Chức năng xem danh sách sản phẩm.....	59
3.3.8	Chức năng xem chi tiết sản phẩm.....	60
3.3.9	Chức năng xem danh sách đơn hàng.....	60
3.3.10	Chức năng xem chi tiết đơn hàng.....	61
3.3.11	Chức năng huỷ đơn hàng.....	61
3.3.12	Chức năng lọc các sản phẩm.....	62
3.3.13	Chức năng tìm kiếm sản phẩm.....	62
3.3.14	Chức năng chatbot.....	63
3.3.15	Chức năng gợi ý sản phẩm cho người dùng.....	64
3.3.16	Chức năng quản lí giỏ hàng.....	65
3.3.17	Chức năng admin quản lí user.....	65
3.3.18	Chức năng admin thêm danh mục mới.....	66
3.3.19	Chức năng sửa thông tin danh mục.....	66
3.3.20	Chức năng admin xem danh sách đơn hàng.....	67
3.3.21	Chức năng admin cập nhật trạng thái đơn hàng.....	67
3.3.22	Chức năng admin thêm sản phẩm mới.....	68
3.3.23	Chức năng cập nhật thông tin sản phẩm.....	68
3.4	BIỂU ĐỒ TUẦN TỰ.....	69
3.4.1	Chức năng đăng nhập.....	69
3.4.2	Chức năng đăng kí.....	69
3.4.3	Chức năng lọc sản phẩm.....	70
3.4.4	Chức năng tìm kiếm sản phẩm.....	70
3.4.5	Chức năng user thêm sản phẩm vào giỏ hàng.....	71
3.4.6	Chức năng admin xoá tài khoản người dùng.....	71
3.4.7	Chức năng thanh toán.....	72
3.4.8	Chức năng user xem danh sách đơn hàng.....	73
3.4.9	Chức năng user xem chi tiết đơn hàng.....	73
3.4.10	Chức năng user thêm địa chỉ.....	74
3.4.11	Chức năng user sửa thông tin địa chỉ.....	74
3.4.12	Chức năng xoá địa chỉ.....	75
3.4.13	Chức năng admin thêm danh mục mới.....	75
3.4.14	Chức năng admin cập nhật thông tin danh mục.....	76

3.4.15	Chức năng admin thêm mới sản phẩm.....	76
3.4.16	Chức năng cập nhật thông tin sản phẩm.....	77
3.4.17	Chức năng admin hiển thị toàn bộ đơn hàng.....	77
3.4.18	Chức năng admin cập nhật trạng thái đơn hàng	78
3.4.19	Chức năng admin hiển thị danh sách người dùng	78
3.5	LƯỢC ĐỒ CƠ SỞ DỮ LIỆU	79
3.5.1	Sơ đồ Entity Relationship Diagram – ERD.....	79
3.6	MÔ TẢ CƠ SỞ DỮ LIỆU	81
3.6.1	Mô tả bảng users	81
3.6.2	Mô tả bảng role.....	82
3.6.3	Mô tả bảng verification_code	82
3.6.4	Mô tả bảng cart.....	82
3.6.5	Mô tả bảng cart_items	83
3.6.6	Mô tả bảng orders.....	84
3.6.7	Mô tả bảng order_items.....	84
3.6.8	Mô tả bảng home_category.....	85
3.6.9	Mô tả bảng products	85
3.6.10	Mô tả bảng addresses.....	86
3.6.11	Mô tả bảng payment_order.....	87
3.6.12	Mô tả bảng transaction	87
3.7	SƠ ĐỒ TỔNG QUAN HỆ THỐNG	88
3.7.1	Kết luận	88

CHƯƠNG 4: XÂY DỰNG WEBSITE THƯƠNG MẠI ĐIỆN TỬ THEO KIẾN TRÚC MICROSERVICES.....89

4.1	TỔNG QUAN VỀ CÔNG NGHỆ SỬ DỤNG.....	89
4.1.1	Apache Kafka	89
4.1.2	Docker	89
4.1.3	Kubernetes.....	89
4.1.4	ReactJS và Redux Toolkit.....	89
4.1.5	Cơ sở dữ liệu quan hệ MySQL	90
4.1.6	Cơ sở dữ liệu phi quan hệ MongoDB.....	90
4.1.7	Cơ sở dữ liệu Vector ChromaDB	90
4.1.8	Công cụ hỗ trợ.....	90
4.2	GIAO DIỆN HỆ THỐNG	91

4.2.1	Trang chủ người dùng.....	91
4.2.2	Giao diện chatbot.....	93
4.2.3	Giao diện phân gợi ý sản phẩm.....	94
4.2.4	Trang đăng nhập.....	95
4.2.5	Trang đăng kí.....	96
4.2.6	Trang lấy lại mật khẩu.....	97
4.2.7	Trang danh sách sản phẩm.....	99
4.2.8	Trang chi tiết sản phẩm.....	100
4.2.9	Trang danh sách đơn hàng của người dùng.....	101
4.2.10	Trang cá nhân người dùng.....	102
4.2.11	Trang danh sách các địa chỉ.....	103
4.2.12	Giao diện thêm địa chỉ mới.....	104
4.2.13	Giao diện sửa thông tin địa chỉ.....	106
4.2.14	Trang giỏ hàng của người dùng.....	107
4.2.15	Trang thanh toán.....	109
4.2.16	Trang chủ giao diện admin.....	110
4.2.17	Trang danh sách các đơn hàng.....	111
4.2.18	Trang danh sách sản phẩm.....	112
4.2.19	Trang thêm sản phẩm mới.....	113
4.2.20	Trang cập nhật sản phẩm.....	114
4.2.21	Trang danh sách người dùng.....	116
4.2.22	Trang danh sách danh mục.....	117
4.2.23	Trang thêm danh mục mới.....	118
4.2.24	Trang cập nhật danh mục.....	119
	KẾT LUẬN.....	120
	TÀI LIỆU THAM KHẢO.....	123

DANH SÁCH BẢNG

Bảng 3.1 Các tác nhân.....	32
Bảng 3.2 Đặc tả usecase xem thông tin cá nhân	35
Bảng 3.3 Đặc tả usecase admin quản lí sản phẩm	37
Bảng 3.4 Đặc tả usecase admin quản lí danh mục	39
Bảng 3.5 Đặc tả usecase admin quản lí đơn hàng.....	42
Bảng 3.6 Đặc tả usecase admin quản lí tồn kho.....	43
Bảng 3.7 Đặc tả usecase quản lí nhân hàng	45
Bảng 3.8 Đặc tả usecase quản lí người dùng	48
Bảng 3.9 Đặc tả usecase quản lí profile	49
Bảng 3.10 Đặc tả usecase quản lí địa chỉ.....	50
Bảng 3.11 Đặc tả usecase quản lí giỏ hàng.....	52
Bảng 3.12 Đặc tả usecase quản lí đơn hàng.....	53
Bảng 3.13 Đặc tả usecase chatbot.....	55
Bảng 3.14 Đặc tả usecase gợi ý sản phẩm	56
Bảng 3.15 Mô tả bảng users.....	81
Bảng 3.16 Mô tả bảng role.....	82
Bảng 3.17 Mô tả bảng verification_code	82
Bảng 3.18 Mô tả bảng cart.....	82
Bảng 3.19 Mô tả bảng cart_items	83
Bảng 3.20 Mô tả bảng orders	84
Bảng 3.21 Mô tả bảng order_items.....	84
Bảng 3.22 Mô tả bảng home_category	85
Bảng 3.23 Mô tả bảng products	85
Bảng 3.24 Mô tả bảng addresses.....	86
Bảng 3.25 Mô tả bảng payment_order.....	87
Bảng 3.26 Mô tả bảng transaction	87
Bảng 4.1 Đặc tả giao diện trang chủ	91
Bảng 4.2 Đặc tả giao diện chatbot	93
Bảng 4.3 Đặc tả giao diện gợi ý sản phẩm.....	94
Bảng 4.4 Đặc tả giao diện đăng nhập.....	95
Bảng 4.5 Đặc tả giao diện đăng kí	96
Bảng 4.6 Đặc tả giao diện lấy lại mật khẩu	97
Bảng 4.7 Đặc tả giao diện danh sách sản phẩm	99
Bảng 4.8 Đặc tả giao diện chi tiết sản phẩm.....	100

Bảng 4.9 Đặc tả giao diện danh sách đơn hàng	101
Bảng 4.10 Đặc tả giao diện cá nhân.....	102
Bảng 4.11 Đặc tả giao diện danh sách địa chỉ	103
Bảng 4.12 Đặc tả giao diện thêm địa chỉ mới	104
Bảng 4.13 Đặc tả giao diện sửa thông tin địa chỉ	106
Bảng 4.14 Đặc tả giao diện giỏ hàng	107
Bảng 4.15 Đặc tả giao diện thanh toán	109
Bảng 4.16 Đặc tả giao diện trang chủ admin	110
Bảng 4.17 Đặc tả giao diện danh sách đơn hàng	111
Bảng 4.18 Đặc tả giao diện quản lí sản phẩm.....	112
Bảng 4.19 Đặc tả giao diện thêm sản phẩm mới.....	113
Bảng 4.20 Đặc tả giao diện cập nhật thông tin sản phẩm	115
Bảng 4.21 Đặc tả giao diện danh sách người dùng.....	116
Bảng 4.22 Đặc tả giao diện danh sách danh mục.....	117
Bảng 4.23 Đặc tả giao diện thêm danh mục	118
Bảng 4.24 Đặc tả giao diện cập nhật danh mục	119

DANH SÁCH HÌNH VẼ

Hình 1.1 Kiến trúc Monolithic.....	7
Hình 1.2 Kiến trúc Microservices.....	8
Hình 1.3 Cách hoạt động của RAG.....	10
Hình 2.1 Sơ đồ hoạt động của Chatbot.....	21
Hình 2.2 Testcase tìm kiếm sản phẩm theo giá.....	24
Hình 2.3 Testcase hỏi về sản phẩm.....	25
Hình 2.4 Testcase hỏi về sản phẩm.....	25
Hình 2.5 Testcase chitchat.....	26
Hình 2.6 Testcase chichat.....	26
Hình 2.7 Kết quả thực nghiệm.....	29
Hình 2.8 Kết quả thực nghiệm.....	30
Hình 2.9 Kết quả thực nghiệm.....	30
Hình 3.1 Sơ đồ usecase tổng quát hệ thống.....	34
Hình 3.2 Sơ đồ usecase xem thông tin cá nhân.....	35
Hình 3.3 Sơ đồ usecase quản lí sản phẩm.....	36
Hình 3.4 Sơ đồ usecase admin quản lí danh mục.....	39
Hình 3.5 Sơ đồ usecase admin quản lí đơn hàng.....	41
Hình 3.6 Sơ đồ usecase admin quản lí tồn kho.....	43
Hình 3.7 Sơ đồ usecase admin quản lí nhãn hàng.....	45
Hình 3.8 Sơ đồ usecase admin quản lí user.....	47
Hình 3.9 Sơ đồ usecase quản lí profile.....	49
Hình 3.10 Sơ đồ usecase quản lí địa chỉ nhận hàng.....	50
Hình 3.11 Sơ đồ usecase quản lí giỏ hàng.....	51
Hình 3.12 Sơ đồ usecase quản lí đơn hàng.....	53
Hình 3.13 Sơ đồ usecase chatbot.....	54
Hình 3.14 Sơ đồ usecase gợi ý sản phẩm.....	55
Hình 3.15 Sơ đồ hoạt động đăng nhập.....	56
Hình 3.17 Sơ đồ hoạt động đăng kí.....	57
Hình 3.18 Sơ đồ hoạt động thêm sản phẩm vào giỏ.....	57
Hình 3.19 Sơ đồ hoạt động thêm địa chỉ.....	58
Hình 3.20 Sơ đồ hoạt động cập nhật địa chỉ.....	58
Hình 3.21 Sơ đồ hoạt động quản lí thông tin cá nhân.....	59
Hình 3.22 Sơ đồ hoạt động xem danh sách sản phẩm.....	59
Hình 3.23 Sơ đồ hoạt động xem chi tiết sản phẩm.....	60

Hình 3.24 Sơ đồ hoạt động xem danh sách đơn hàng.....	60
Hình 3.25 Sơ đồ hoạt động xem chi tiết đơn hàng.....	61
Hình 3.26 Sơ đồ hoạt động huỷ đơn hàng.....	61
Hình 3.27 Sơ đồ hoạt động lọc sản phẩm	62
Hình 3.28 Sơ đồ hoạt động tìm kiếm sản phẩm.....	62
Hình 3.29 Sơ đồ hoạt động của chatbot	63
Hình 3.30 Sơ đồ hoạt động gợi ý sản phẩm.....	64
Hình 3.31 Sơ đồ hoạt động quản lí giỏ hàng.....	65
Hình 3.32 Sơ đồ hoạt động quản lí user.....	65
Hình 3.33 Sơ đồ hoạt động thêm danh mục.....	66
Hình 3.34 Sơ đồ hoạt động sửa thông tin danh mục	66
Hình 3.35 Sơ đồ hoạt động admin xem danh sách đơn hàng.....	67
Hình 3.36 Sơ đồ hoạt động cập nhật trạng thái đơn hàng.....	67
Hình 3.37 Sơ đồ hoạt động thêm sản phẩm	68
Hình 3.38 Sơ đồ hoạt động cập nhật thông tin sản phẩm	68
Hình 3.39 Sơ đồ tuần tự chức năng đăng nhập	69
Hình 3.40 Sơ đồ tuần tự chức năng đăng kí.....	69
Hình 3.41 Sơ đồ tuần tự lọc sản phẩm	70
Hình 3.42 Sơ đồ tuần tự tìm kiếm sản phẩm.....	70
Hình 3.43 Sơ đồ tuần tự thêm sản phẩm vào giỏ hàng	71
Hình 3.44 Sơ đồ tuần tự xoá tài khoản người dùng	71
Hình 3.45 Sơ đồ tuần tự chức năng thanh toán.....	72
Hình 3.46 Sơ đồ tuần tự xem danh sách đơn hàng	73
Hình 3.47 Sơ đồ tuần tự xem chi tiết đơn hàng	73
Hình 3.48 Sơ đồ tuần tự thêm địa chỉ	74
Hình 3.49 Sơ đồ tuần tự sửa địa chỉ.....	74
Hình 3.50 Sơ đồ tuần tự xoá địa chỉ.....	75
Hình 3.51 Sơ đồ tuần tự thêm danh mục mới	75
Hình 3.52 Sơ đồ tuần tự cập nhật danh mục	76
Hình 3.53 Sơ đồ tuần tự thêm mới sản phẩm.....	76
Hình 3.54 Sơ đồ tuần tự cập nhật thông tin sản phẩm	77
Hình 3.55 Sơ đồ tuần tự hiển thị danh sách đơn hàng	77
Hình 3.56 Sơ đồ tuần tự cập nhật trạng thái đơn hàng.....	78
Hình 3.57 Sơ đồ tuần tự hiển thị danh sách user	78

Hình 3.58 Sơ đồ ERD	79
Hình 3.59 Sơ đồ tổng quan hệ thống.....	88
Hình 4.1 Trang chủ người dùng.....	91
Hình 4.2 Giao diện chatbot	93
Hình 4.3 Giao diện gợi ý sản phẩm.....	94
Hình 4.4 Trang đăng nhập.....	95
Hình 4.5 Trang đăng kí	96
Hình 4.6 Trang lấy lại mật khẩu người dùng	97
Hình 4.7 Trang danh sách sản phẩm.....	99
Hình 4.8 Trang chi tiết sản phẩm.....	100
Hình 4.9 Trang danh sách đơn hàng của người dùng	101
Hình 4.10 Trang cá nhân người dùng	102
Hình 4.11 Trang danh sách địa chỉ	103
Hình 4.12 Giao diện thêm địa chỉ mới.....	104
Hình 4.13 Giao diện sửa thông tin địa chỉ	106
Hình 4.14 Trang giỏ hàng của người dùng	107
Hình 4.15 Trang thanh toán	109
Hình 4.16 Giao diện trang chủ admin.....	110
Hình 4.17 Trang danh sách các đơn hàng.....	111
Hình 4.18 Trang danh sách sản phẩm.....	112
Hình 4.19 Trang thêm sản phẩm mới.....	113
Hình 4.20 Trang cập nhật thông tin của sản phẩm.....	114
Hình 4.21 Trang danh sách người dùng.....	116
Hình 4.22 Trang danh sách danh mục.....	117
Hình 4.23 Trang thêm danh mục mới	118
Hình 4.24 Trang cập nhật thông tin danh mục.....	119

DANH SÁCH CÁC CHỮ VIẾT TẮT

STT	Chữ viết tắt	Diễn giải
1	RAG	Retrieval-Augmented Generation
2	RRF	Reciprocal Rank Fusion
3	API	Application Programming Interface
4	LLM	Large Language Model
5	ACID	Atomic, Consistent, Isolated, Durable
6	RESTful	Representational State Transfer
7	AI	Artificial intelligence
8	DB	Database
9	RDBMS	Relational Database Management System
10	NoSQL	Not Only SQL
11	NLP	Natural Language Processing
12	ERD	Entity-Relationship Diagram
13	NNS	Nearest Neighbor Search
14	ANN	Approximate Nearest Neighbor

MỞ ĐẦU

1. Lí do chọn đề tài

- Trong thời đại công nghệ số phát triển mạnh mẽ, nhu cầu mua sắm trực tuyến ngày càng gia tăng. Các hệ thống website bán hàng truyền thống (monolithic) thường gặp khó khăn trong việc mở rộng, bảo trì và đáp ứng lượng truy cập lớn.
- Kiến trúc Microservices ra đời như một giải pháp tối ưu, cho phép chia nhỏ hệ thống thành các dịch vụ độc lập, dễ dàng phát triển và triển khai. Kết hợp với Apache Kafka để xử lý dữ liệu thời gian thực, Docker để đóng gói ứng dụng và Kubernetes để triển khai.
- Đề tài này không chỉ có tính ứng dụng cao, giải quyết nhiều vấn đề mà hệ thống Monolithic gặp phải, mà còn giúp bản thân em tiếp cận các công nghệ tiên tiến đang được ứng dụng rộng rãi trong ngành công nghệ thông tin. Mặc dù còn tồn tại một số hạn chế, nhưng với các phương pháp khắc phục phù hợp, hệ thống có thể đạt được hiệu suất cao, khả năng mở rộng tốt và dễ bảo trì.

2. Mục tiêu đề tài

- Xây dựng một hệ thống website bán hàng hiện đại với giao diện thân thiện, hiệu năng cao và khả năng phản hồi nhanh, nhằm hỗ trợ người dùng mua sắm thuận tiện, nhanh chóng và hiệu quả.
- Tích hợp chatbot AI hỗ trợ khách hàng với khả năng hiểu và phản hồi ngôn ngữ tự nhiên, giúp tự động hóa việc tư vấn, giải đáp thắc mắc của người dùng.
- Hỗ trợ các phương thức thanh toán điện tử như VNPay nhằm mang lại sự tiện lợi và đa dạng trong quá trình thanh toán.
- Quản trị viên dễ dàng kiểm soát các đơn hàng, sản phẩm được bán, theo dõi doanh thu.
- Đem lại cho người dùng nhiều lựa chọn mua hàng, thanh toán, giải đáp mọi thắc mắc cho người dùng 24/7, giúp người dùng quản lí và theo dõi các đơn hàng.
- Việc xây dựng hệ thống theo kiến trúc Microservices giúp cho việc mở rộng, thêm các tính năng mới cũng như việc sửa chữa, bảo trì sau này có thể được thực hiện một cách dễ dàng.

3. Phạm vi và đối tượng nghiên cứu

3.1. Đối tượng nghiên cứu

- Kiến trúc Microservices và các phương pháp triển khai trong hệ thống thương mại điện tử.
- Hệ thống thương mại điện tử: tập trung vào việc phân tích và xây dựng một hệ thống thương mại điện tử hoàn chỉnh, bao gồm các chức năng cốt lõi như quản lý sản phẩm, giỏ hàng, thanh toán trực tuyến, quản lý đơn hàng, và quản lý người dùng.
- Công nghệ và các công cụ hỗ trợ:
 - + Docker và Kubernetes để đóng gói và triển khai hệ thống.
 - + Hệ thống cơ sở dữ liệu
 - + Apache Kafka trong việc xử lý dữ liệu bất đồng bộ, đảm bảo hệ thống có thể mở rộng linh hoạt.
 - + AI chatbot để hỗ trợ khách hàng
 - + Hệ thống gợi ý sản phẩm cho khách hàng
 - + Tích hợp VNPAY việc thanh toán.
- Người dùng và hành vi: nghiên cứu hành vi và nhu cầu của người dùng cuối (end-user) trong hệ thống thương mại điện tử, từ đó thiết kế các microservices đáp ứng tốt các yêu cầu về hiệu suất, độ tin cậy và trải nghiệm người dùng.

3.2. Phạm vi

- Phạm vi hệ thống:
 - + Xây dựng website bán hàng trực tuyến theo kiến trúc Microservices, với các microservices độc lập cho từng chức năng.
 - + Sử dụng Apache Kafka để quản lý giao tiếp bất đồng bộ giữa các microservices.
 - + Tích hợp cổng thanh toán VNPAY cho chức năng thanh toán trực tuyến.
 - + Triển khai chatbot AI để hỗ trợ khách hàng tự động.
 - + Xây dựng hệ thống gợi ý sản phẩm dựa trên hành vi người dùng.
 - + Thiết kế và triển khai các cơ sở dữ liệu riêng biệt cho từng microservice để đảm bảo tính độc lập và khả năng mở rộng.
- Phạm vi ứng dụng:
 - + Hệ thống được triển khai dưới dạng ứng dụng web, hỗ trợ truy cập qua trình duyệt trên các thiết bị khác nhau.

- + Giao diện người dùng được phát triển bằng ReactJS và quản lý trạng thái bằng Redux Toolkit.
- Phạm vi người dùng:
 - + Khách hàng: Người mua hàng có thể
 - Đăng nhập, đăng kí và lấy lại mật khẩu
 - Xem danh sách và chi tiết sản phẩm
 - Tìm kiếm và lọc sản phẩm
 - Thêm và cập nhật số lượng sản phẩm trong giỏ hàng
 - Nhấn với chatbot
 - Quản lí profile: xem chi tiết và danh sách đơn hàng; thêm và sửa địa chỉ mới
 - + Quản trị viên: Người quản lý hệ thống
 - Xem thông kê doanh thu
 - Quản lý sản phẩm
 - Cập nhật trạng thái đơn hàng
 - Quản lí người dùng
 - Quản lí danh mục

4. Phương pháp nghiên cứu

- Tổng quan tài liệu:
 - + Tiến hành xem xét toàn diện các tài liệu hiện có về hệ thống thương mại điện tử, kiến trúc microservices và các công nghệ liên quan để thiết lập nền tảng lý thuyết và xác định các thực tiễn tốt nhất.
- Thu thập và phân tích yêu cầu:
 - + Tương tác với các bên liên quan để thu thập các yêu cầu chi tiết cho nền tảng thương mại điện tử.
 - + Phân tích các yêu cầu này để xác định rõ ràng các đặc tả chức năng và phi chức năng.
- Thiết kế kiến trúc hệ thống:
 - + Thiết kế kiến trúc dựa trên microservices, xác định các dịch vụ riêng lẻ như quản lý sản phẩm, xử lý đơn hàng và xác thực người dùng.
 - + Lập kế hoạch tích hợp Apache Kafka để xử lý giao tiếp bất đồng bộ giữa các dịch vụ.

- Phát triển microservices và giao diện:
 - + Phát triển từng microservice bằng cách sử dụng các công nghệ phù hợp, đảm bảo chúng có thể mở rộng và bảo trì được.
 - + Xây dựng giao diện người dùng bằng ReactJS và quản lý trạng thái với Redux Toolkit để cung cấp trải nghiệm người dùng mượt mà.
- Tích hợp các dịch vụ bên ngoài:
 - + Tích hợp VNPAY để xử lý thanh toán trực tuyến an toàn.
 - + Triển khai chatbot AI để hỗ trợ khách hàng và hệ thống gợi ý để cá nhân hóa các đề xuất sản phẩm.
- Containerization và orchestration:
 - + Sử dụng Docker để containerize từng microservice, tạo điều kiện cho việc triển khai nhất quán trên các môi trường khác nhau.
 - + Triển khai các dịch vụ containerized trên cụm Kubernetes để quản lý việc mở rộng, cân bằng tải và failover.
- Kiểm thử và đảm bảo chất lượng:
 - + Thực hiện kiểm thử nghiêm ngặt, bao gồm kiểm thử đơn vị cho từng dịch vụ, kiểm thử tích hợp cho các tương tác dịch vụ và kiểm thử end-to-end cho toàn bộ hệ thống.
 - + Đảm bảo hệ thống đáp ứng các tiêu chuẩn về hiệu suất, bảo mật và khả năng sử dụng.

5. Cấu trúc dự án

- Mở đầu
- Chương 1: Cơ sở lý thuyết
- Chương 2: Phương pháp đề xuất
- Chương 3: Phân tích và thiết kế hệ thống
- Chương 4: Triển khai hệ thống
- Kết luận
- Tài liệu tham khảo

Chương 1: CƠ SỞ LÝ THUYẾT

1.1 Kiến trúc Microservices

1.1.1 Định nghĩa

Kiến trúc Microservices, một paradigm kiến trúc phần mềm hiện đại, tiếp cận việc xây dựng các ứng dụng phức tạp bằng cách cấu trúc chúng thành một tập hợp các service nhỏ, và được triển khai độc lập. Các service này thường giao tiếp với nhau thông qua các giao thức mạng nhẹ, phổ biến nhất là Representational State Transfer (RESTful) API hoặc các hệ thống hàng đợi như RabbitMQ hoặc Kafka.

1.1.2 Vai trò

Việc áp dụng kiến trúc Microservices vào dự án đem lại nhiều lợi ích và tích cực trên nhiều khía cạnh:

- Khả năng mở rộng linh hoạt và hiệu quả:
 - + Trong một hệ thống thương mại điện tử truyền thống (monolithic), việc mở rộng thường đòi hỏi nhiều công sức, ngay cả khi chỉ có một vài thành phần cụ thể đang chịu tải cao. Với Microservices, mỗi dịch vụ, chẳng hạn như service quản lý người dùng, quản lý danh mục sản phẩm, xử lý đơn hàng, hay hệ thống đề xuất có thể được mở rộng một cách độc lập.
 - + Mở rộng theo chiều ngang (Horizontal Scaling): Khi lưu lượng truy cập vào một dịch vụ cụ thể tăng đột biến như là đang có chương trình khuyến mãi, chỉ cần nhân bản các instance của dịch vụ đó mà không cần tác động đến các dịch vụ khác. Điều này giúp tối ưu hóa việc sử dụng tài nguyên, giảm chi phí vận hành so với việc mở rộng toàn bộ ứng dụng.
 - + Mở rộng theo chiều dọc (Vertical Scaling): Đối với các dịch vụ đòi hỏi nhiều tài nguyên hơn về CPU hoặc bộ nhớ, có thể nâng cấp phần cứng cho các máy chủ chạy dịch vụ đó một cách độc lập. Khả năng granular scaling này đảm bảo hệ thống có thể đáp ứng một cách linh hoạt với sự biến động của nhu cầu, từ đó cải thiện trải nghiệm người dùng và hiệu suất tổng thể của nền tảng thương mại điện tử.

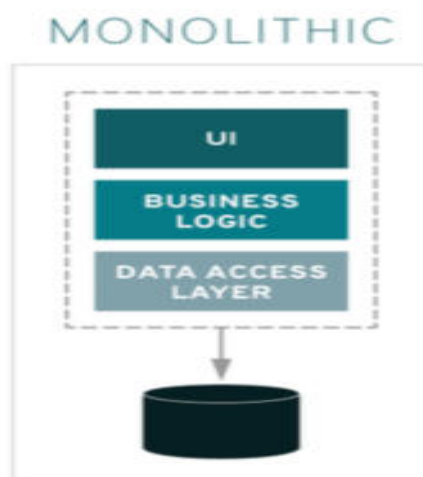
- Tối ưu hóa quá trình bảo trì và triển khai:
 - + Kiến trúc Microservices phân rã một ứng dụng lớn thành các thành phần nhỏ hơn, mỗi service có nhiệm vụ rõ ràng. Điều này mang lại những lợi ích đáng kể trong quá trình bảo trì và triển khai:
 - Giảm thiểu rủi ro khi triển khai: Việc cập nhật, vá lỗi, hoặc triển khai một phiên bản mới của một microservice chỉ ảnh hưởng đến dịch vụ đó và các dịch vụ phụ thuộc trực tiếp, hạn chế tối đa nguy cơ gây ra sự cố cho toàn bộ hệ thống.
 - Thời gian hoàn thiện sản phẩm được cải thiện: Các nhóm phát triển nhỏ hơn, tập trung vào một dịch vụ cụ thể, có thể làm việc và phát hành các tính năng mới hoặc bản vá lỗi một cách nhanh chóng và độc lập. Điều này giúp rút ngắn chu kỳ phát triển và mang lại giá trị cho người dùng một cách liên tục.
 - Đơn giản hóa việc bảo trì: Với codebase nhỏ hơn và tính độc lập cao, việc xác định và sửa lỗi trong một microservice trở nên dễ dàng hơn so với việc gỡ rối một ứng dụng nguyên khối phức tạp. Điều này giảm thiểu thời gian chết (downtime) và chi phí bảo trì.
- Nâng cao tính linh hoạt và đổi mới trong phát triển:
 - + Kiến trúc Microservices không áp đặt một khuôn mẫu công nghệ duy nhất cho toàn bộ hệ thống. Điều này tạo ra sự linh hoạt lớn trong quá trình phát triển:
 - Đa dạng hóa công nghệ: Mỗi microservice có thể được xây dựng bằng ngôn ngữ lập trình, framework, và cơ sở dữ liệu phù hợp nhất với yêu cầu cụ thể của nó. Ví dụ, một dịch vụ xử lý ảnh có thể sử dụng các thư viện tối ưu cho việc này, trong khi một dịch vụ quản lý giao dịch có thể ưu tiên một cơ sở dữ liệu ACID-compliant. Sự linh hoạt này cho phép các nhóm phát triển tận dụng các công nghệ mới nhất và phù hợp nhất, thúc đẩy sự đổi mới.
 - Tổ chức đội ngũ hiệu quả: Các nhóm phát triển có thể được tổ chức xung quanh các nghiệp vụ cụ thể. Mỗi nhóm có quyền tự chủ trong việc lựa chọn công nghệ và quy trình làm việc cho dịch vụ của mình, tăng tính hiệu quả làm việc.

- Tăng cường khả năng chịu lỗi và tính ổn định:
 - + Trong kiến trúc Microservices, các dịch vụ hoạt động độc lập. Nếu một microservice gặp sự cố, nó không làm ảnh hưởng trực tiếp đến các dịch vụ khác. Cơ chế cách ly lỗi này giúp hệ thống tổng thể trở nên ổn định và có khả năng phục hồi tốt hơn.
 - Degraded Functionality: Trong trường hợp một service bị lỗi, các phần khác của hệ thống vẫn có thể tiếp tục hoạt động (có thể với một số chức năng bị hạn chế). Ví dụ, nếu dịch vụ đề xuất sản phẩm gặp sự cố, người dùng vẫn có thể duyệt và mua hàng.
 - Resilience Patterns: Kỹ thuật Fallback có thể được triển khai trên từng microservice để tăng cường khả năng chịu lỗi và tự phục hồi.

Việc áp dụng kiến trúc Microservices mang lại một nền tảng vững chắc và linh hoạt cho hệ thống thương mại điện tử, cho phép nó phát triển, mở rộng và thích ứng với những thay đổi của thị trường và nhu cầu người dùng một cách hiệu quả. Tuy nhiên, việc triển khai và quản lý một hệ thống Microservices cũng đặt ra những thách thức mới về quản lý độ phức tạp, giao tiếp giữa các service, và quản lý dữ liệu phân tán.

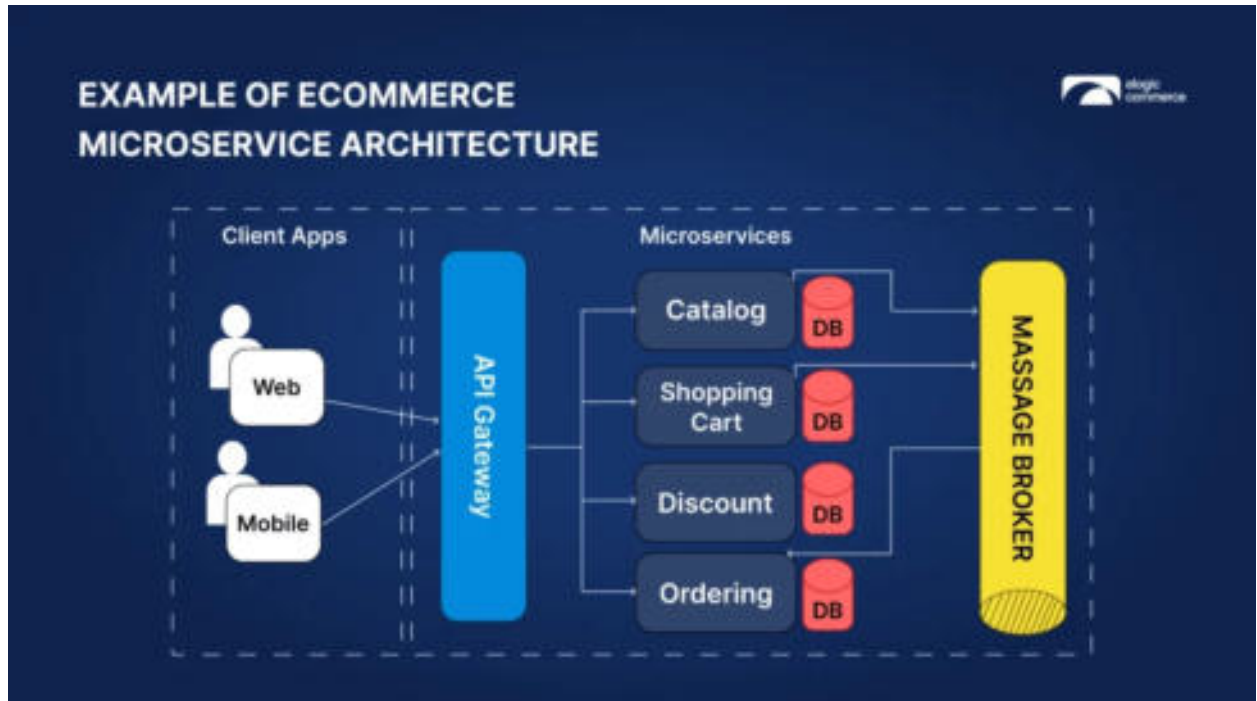
1.1.3 So sánh giữa kiến trúc Monolithic và Microservices

- Kiến trúc Monolithic là mô hình truyền thống trong phát triển phần mềm, nơi toàn bộ ứng dụng được đóng gói trong một khối mã lệnh duy nhất. Tất cả các chức năng từ hiển thị giao diện, xử lý nghiệp vụ đến quản lý dữ liệu – đều nằm trong một đơn vị triển khai. Mặc dù cách tiếp cận này đơn giản trong giai đoạn ban đầu, nhưng nó bộc lộ nhiều hạn chế khi hệ thống phát triển về quy mô và độ phức tạp.



Hình 1.1 Kiến trúc Monolithic

- Trong khi đó, kiến trúc Microservices chia hệ thống thành nhiều dịch vụ nhỏ, độc lập, mỗi dịch vụ đảm nhận một chức năng riêng biệt và giao tiếp thông qua giao thức mạng như HTTP (RESTful) hoặc hệ thống hàng đợi (Kafka, RabbitMQ). Cách tiếp cận này mang lại nhiều ưu điểm vượt trội, đặc biệt là trong lĩnh vực thương mại điện tử, như đã được nhấn mạnh trong nghiên cứu của Elogic (2024) về lợi ích của Microservices trong xây dựng các hệ thống eCommerce hiện đại .



Hình 1.2 Kiến trúc Microservices

- Một số điểm so sánh nổi bật giữa hai kiến trúc bao gồm:
 - + **Phân tách chức năng:** Trong kiến trúc Monolithic, toàn bộ nghiệp vụ được tích hợp trong cùng một khối, khiến việc phát triển và bảo trì trở nên khó khăn khi mở rộng. Ngược lại, Microservices phân chia rõ ràng theo nghiệp vụ như người dùng, sản phẩm, đơn hàng,... giúp tổ chức nhóm phát triển dễ dàng hơn và giảm thiểu độ phức tạp trong bảo trì.
 - + **Khả năng mở rộng:** Monolithic thường đòi hỏi mở rộng toàn bộ hệ thống ngay cả khi chỉ một phần cần thêm tài nguyên, gây lãng phí và khó kiểm soát. Microservices hỗ trợ mở rộng theo chiều ngang (tăng số lượng instance) hoặc chiều dọc (nâng cấp phần cứng riêng biệt) cho từng dịch vụ cụ thể – từ đó tối ưu hóa chi phí và tài nguyên.

- + Tốc độ triển khai và thời gian đưa ra thị trường (time-to-market): Do sự độc lập giữa các dịch vụ, các nhóm phát triển có thể phát hành tính năng mới, sửa lỗi nhanh chóng mà không ảnh hưởng đến các phần còn lại của hệ thống – điều mà kiến trúc Monolithic không thể đáp ứng hiệu quả.
- + Khả năng tương thích với nhiều nền tảng: Microservices hỗ trợ các mô hình triển khai hiện đại như Headless Commerce – cho phép backend và frontend hoạt động tách biệt, linh hoạt phục vụ nhiều nền tảng (Web, Mobile, IoT...).
- + Tính ổn định và khả năng chịu lỗi cao: Khi một thành phần trong hệ thống monolithic gặp lỗi, toàn bộ ứng dụng có thể bị ảnh hưởng. Trong Microservices, sự cố của một dịch vụ có thể được cô lập, và các phần khác vẫn tiếp tục hoạt động (ví dụ như chức năng mua hàng vẫn sử dụng được ngay cả khi dịch vụ gợi ý sản phẩm tạm thời không khả dụng).
- Tóm lại, mặc dù kiến trúc Microservices đi kèm với các thách thức như quản lý độ phức tạp, đồng bộ dữ liệu và giao tiếp liên dịch vụ, nhưng với những ưu điểm nổi bật đã trình bày, đây là lựa chọn phù hợp để xây dựng các hệ thống thương mại điện tử hiện đại, có yêu cầu cao về khả năng mở rộng, tính linh hoạt, và thời gian triển khai nhanh chóng.

1.2 Retrieval-Augmented Generation (RAG)

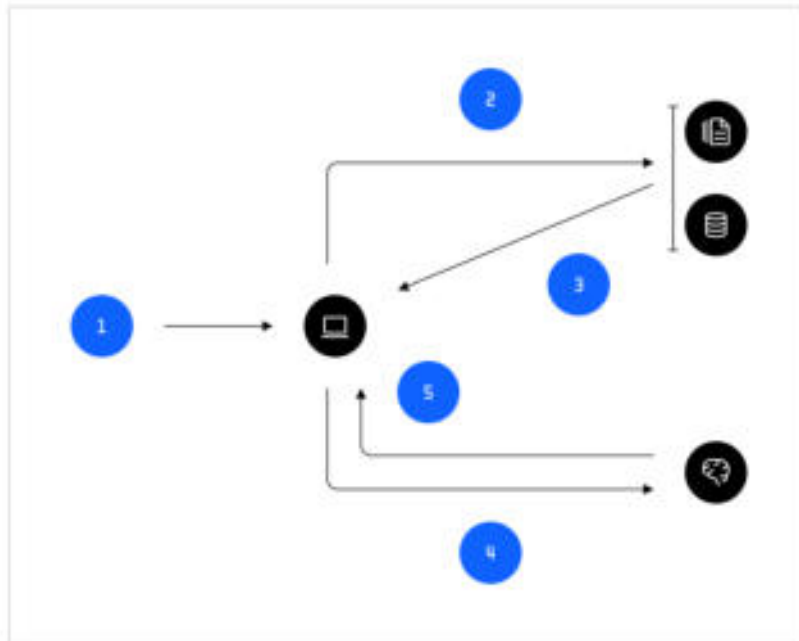
1.2.1 Định nghĩa

- Retrieval-Augmented Generation (RAG) là một kỹ thuật giúp tăng độ chính xác và độ tin cậy của các mô hình AI sinh ngữ (generative AI) bằng cách kết hợp thông tin truy xuất từ các nguồn dữ liệu cụ thể, liên quan. Thay vì chỉ dựa vào kiến thức đã được huấn luyện sẵn trong mô hình ngôn ngữ lớn, RAG cho phép mô hình truy cập vào các kho dữ liệu bên ngoài để lấy thông tin mới nhất, chính xác nhất phục vụ cho mục đích trả lời truy vấn cho người dùng và đảm bảo câu trả lời chính xác.

1.2.2 Cách hoạt động của RAG

- RAG hoạt động bằng cách kết hợp các mô hình truy xuất thông tin với các mô hình AI sinh văn bản để tạo ra nội dung có tính xác thực cao hơn. Hệ thống RAG sẽ truy vấn một cơ sở tri thức và thêm ngữ cảnh vào câu hỏi của người dùng trước khi tạo ra phản hồi.
- Thông thường, các mô hình ngôn ngữ lớn (LLM) chỉ lấy thông tin từ tập dữ liệu huấn luyện. RAG bổ sung một thành phần truy xuất thông tin vào quy trình làm việc

của AI, thu thập dữ liệu liên quan và cung cấp nó cho mô hình sinh văn bản nhằm nâng cao chất lượng và tính hữu ích của câu trả lời.



Hình 1.3 Cách hoạt động của RAG

- Hệ thống RAG hoạt động qua 5 giai đoạn:
 1. Người dùng gửi một câu hỏi (prompt).
 2. Mô hình truy xuất thông tin thực hiện truy vấn đến cơ sở tri thức để tìm dữ liệu liên quan.
 3. Thông tin liên quan được trả về từ cơ sở tri thức đến tầng tích hợp (integration layer).
 4. Hệ thống RAG tạo một câu hỏi bổ sung (augmented prompt) cho LLM, có thêm ngữ cảnh từ dữ liệu truy xuất.
 5. LLM sinh ra phản hồi và gửi lại kết quả cho người dùng.
- Quá trình này cho thấy nguồn gốc của tên gọi RAG. Hệ thống RAG thực hiện việc truy xuất dữ liệu (retrieves) từ cơ sở tri thức, bổ sung ngữ cảnh (augments) vào câu hỏi, và tạo ra phản hồi (generates).

1.2.3 Các thành phần của RAG

- Hệ thống RAG bao gồm bốn thành phần chính:
 1. Cơ sở tri thức: Kho dữ liệu bên ngoài mà hệ thống có thể truy xuất.

2. Bộ truy xuất: Tìm kiếm và truy xuất các dữ liệu liên quan dựa trên truy vấn của người dùng, thường sử dụng tìm kiếm dựa trên vector hoặc từ khóa từ cơ sở tri thức (cơ sở dữ liệu vector ChromaDB).
 3. Lớp tích hợp: Thành phần điều phối hoạt động tổng thể của hệ thống RAG.
 4. Bộ sinh: Mô hình ngôn ngữ sử dụng các tài liệu được truy xuất cùng với truy vấn gốc để tạo ra câu trả lời mạch lạc và phù hợp ngữ cảnh.
- Cơ sở tri thức
 - + Giai đoạn đầu tiên trong việc xây dựng một hệ thống RAG là thiết lập một cơ sở tri thức có thể truy vấn. Kho dữ liệu bên ngoài này có thể bao gồm thông tin từ nhiều nguồn khác nhau như tệp PDF, tài liệu, hướng dẫn sử dụng, trang web, tệp âm thanh, v.v. Phần lớn các dữ liệu này là dữ liệu phi cấu trúc, tức là chưa được gán nhãn hoặc tổ chức một cách rõ ràng.
 - + Để có thể sử dụng hiệu quả nguồn dữ liệu này, hệ thống RAG áp dụng một kỹ thuật gọi là embedding, nhằm chuyển đổi dữ liệu thành các biểu diễn số học gọi là vector. Mô hình embedding thực hiện quá trình vector hóa dữ liệu trong một không gian toán học đa chiều, trong đó các điểm dữ liệu được sắp xếp dựa trên mức độ tương đồng ngữ nghĩa. Các điểm dữ liệu có nội dung liên quan mật thiết sẽ được đặt gần nhau trong không gian vector.
 - + Để đảm bảo chất lượng và tính cập nhật của hệ thống RAG, cơ sở tri thức cần được duy trì và cập nhật thường xuyên.
 - + Một hạn chế quan trọng của các mô hình ngôn ngữ lớn là giới hạn cửa sổ ngữ cảnh (context window) - tức là lượng thông tin tối đa mà mô hình có thể xử lý mà không làm mất đi ngữ cảnh. Do đó, tài liệu đầu vào cần được chia thành các đoạn nhỏ (chunking) nhằm đảm bảo rằng quá trình embedding không vượt quá giới hạn của LLM trong hệ thống RAG.
 - + Kích thước đoạn (chunk size) là một siêu tham số quan trọng cần được điều chỉnh hợp lý. Nếu các đoạn quá lớn, thông tin được biểu diễn có thể trở nên quá tổng quát, làm giảm khả năng đáp ứng chính xác đối với truy vấn người dùng. Ngược lại, nếu đoạn quá nhỏ, các biểu diễn có thể bị mất tính nhất quán về mặt ngữ nghĩa, dẫn đến giảm hiệu quả truy xuất thông tin.
 - Bộ truy xuất
 - + Quá trình vector hóa dữ liệu là bước chuẩn bị cần thiết để cơ sở tri thức có thể hỗ trợ tìm kiếm vector ngữ nghĩa (semantic vector search) - một kỹ thuật

cho phép xác định các điểm dữ liệu trong cơ sở tri thức có mức độ tương đồng cao với truy vấn của người dùng. So với các phương pháp tìm kiếm theo từ khóa truyền thống, thuật toán học máy tìm kiếm ngữ nghĩa có khả năng truy vấn trên các cơ sở dữ liệu lớn và nhanh chóng xác định thông tin liên quan, từ đó giảm độ trễ trong quá trình xử lý.

- + Mô hình truy xuất thông tin thực hiện việc chuyển đổi truy vấn của người dùng thành một vector embedding, sau đó thực hiện tìm kiếm các vector tương đồng trong cơ sở tri thức. Các kết quả tìm được sẽ được trả về từ cơ sở tri thức để sử dụng trong giai đoạn tiếp theo của hệ thống RAG.
- Lớp tích hợp
 - + Lớp tích hợp đóng vai trò trung tâm trong kiến trúc của hệ thống RAG, chịu trách nhiệm điều phối các quy trình và truyền tải dữ liệu giữa các thành phần trong mạng. Sau khi dữ liệu từ cơ sở tri thức được truy xuất, hệ thống RAG sử dụng dữ liệu này để tạo một prompt mới cho mô hình ngôn ngữ lớn. Prompt này bao gồm truy vấn gốc của người dùng kết hợp với ngữ cảnh bổ sung do mô hình truy xuất cung cấp.
 - + Các hệ thống RAG thường áp dụng nhiều kỹ thuật thiết kế prompt nhằm tự động hóa quá trình tạo prompt hiệu quả, giúp LLM tạo ra phản hồi chất lượng cao. Bên cạnh đó, các framework điều phối LLM như LangChain, LlamaIndex (mã nguồn mở) đảm nhiệm vai trò quản lý hoạt động tổng thể của hệ thống AI.
- Bộ sinh
 - + Bộ sinh là thành phần tạo ra phản hồi dựa trên prompt đã được tăng cường (augmented prompt) do lớp tích hợp cung cấp. Prompt này đóng vai trò tổng hợp thông tin giữa truy vấn đầu vào của người dùng và dữ liệu đã truy xuất, đồng thời hướng dẫn bộ sinh sử dụng các thông tin này trong quá trình sinh phản hồi.
 - + Các mô hình được sử dụng làm bộ sinh thường là các mô hình ngôn ngữ lớn đã được huấn luyện sẵn, chẳng hạn như GPT, Claude, hoặc Gemini.

1.2.4 Lợi ích của RAG

- + Tăng độ tin cậy: Câu trả lời có thể trích dẫn nguồn, giúp người dùng kiểm chứng thông tin.

- + Giảm hiện tượng "hallucination": Hạn chế việc mô hình tạo ra thông tin sai lệch hoặc không có thật.
- + Dễ triển khai: Có thể tích hợp vào các mô hình hiện tại mà không cần huấn luyện lại toàn bộ.
- + Linh hoạt: Dễ dàng thay đổi, cập nhật nguồn dữ liệu bên ngoài mà không ảnh hưởng đến mô hình gốc.

1.2.5 Ứng dụng của RAG

Trong project này, Retrieval-Augmented Generation (RAG) được ứng dụng để xây dựng một hệ thống gợi ý và tìm kiếm sản phẩm thông minh, kết hợp giữa khả năng truy xuất dữ liệu từ kho sản phẩm và khả năng sinh ngữ của mô hình AI. Cụ thể:

- Tìm kiếm sản phẩm thông minh:
 - + Khi người dùng nhập truy vấn (ví dụ: tìm kiếm sản phẩm):
 - Chuyển truy vấn thành vector (embedding) và thực hiện tìm kiếm ngữ nghĩa trong cơ sở dữ liệu sản phẩm (vector search).
 - Đồng thời thực hiện tìm kiếm theo từ khóa truyền thống (keyword search).
 - Kết hợp kết quả từ cả hai phương pháp này bằng thuật toán RRF để đảm bảo kết quả trả về vừa chính xác, vừa đa dạng.
- Tăng độ chính xác và độ tin cậy của kết quả:
 - + Nhờ việc kết hợp nhiều phương pháp truy xuất và sử dụng thuật toán RRF để xếp hạng, hệ thống có thể trả về các sản phẩm phù hợp nhất với nhu cầu thực tế của người dùng, giảm thiểu việc bỏ sót các sản phẩm quan trọng.
- Tạo nội dung gợi ý sản phẩm:
 - + Dựa trên các sản phẩm đã truy xuất, hệ thống sử dụng khả năng sinh ngữ để tạo ra các đoạn mô tả, gợi ý sản phẩm hấp dẫn, dễ hiểu, giúp người dùng dễ dàng lựa chọn.
- Cá nhân hóa gợi ý:
 - + Hệ thống có thể kết hợp thông tin về sở thích, lịch sử tìm kiếm của người dùng để đưa ra các gợi ý sản phẩm phù hợp, nâng cao trải nghiệm cá nhân hóa.
- Bổ sung sản phẩm ngẫu nhiên:

- + Trong trường hợp kết quả truy xuất chưa đủ, hệ thống sẽ tự động bổ sung thêm các sản phẩm ngẫu nhiên từ kho dữ liệu, đảm bảo luôn cung cấp đủ số lượng gợi ý cho người dùng.

Việc ứng dụng RAG trong project giúp hệ thống tìm kiếm và gợi ý sản phẩm trở nên thông minh, chính xác, đa dạng và cá nhân hóa hơn, mang lại trải nghiệm tốt hơn cho người dùng cuối.

1.3 Sentence Transformers

1.3.1 Định nghĩa

- Sentence Transformers là một họ các mô hình mạng nơ-ron chuyên biệt, được xây dựng dựa trên kiến trúc Transformer (chẳng hạn như BERT, RoBERTa, XLM-RoBERTa, hay MPNet). Điểm khác biệt và cũng là ưu điểm nổi bật của Sentence Transformers so với các mô hình Transformer "cơ bản" là chúng được huấn luyện đặc biệt để tạo ra các biểu diễn vector dày đặc cho toàn bộ câu, đoạn văn, hoặc thậm chí là tài liệu dài. Các embedding này phản ánh ý nghĩa ngữ nghĩa của văn bản, giúp máy tính hiểu và so sánh các câu dựa trên nội dung thực sự, thay vì chỉ dựa vào từ khóa. Mục tiêu chính của quá trình huấn luyện này là đảm bảo rằng các câu hoặc đoạn văn có ý nghĩa ngữ nghĩa tương tự sẽ có các vector nhúng (embedding vectors) gần nhau trong không gian vector đa chiều.
- Trong khi các mô hình Transformer truyền thống như BERT tạo ra các embedding cho từng token (từ/từ vựng) và yêu cầu một bước gộp riêng biệt để có được một embedding cấp câu, Sentence Transformers được tinh chỉnh để tạo ra các embedding chất lượng cao ngay lập tức. Điều này thường được thực hiện thông qua các kỹ thuật huấn luyện Siamese và Triplet Network, nơi mô hình học cách phân biệt giữa các cặp câu giống nhau và khác nhau. Kết quả là một vector duy nhất, phản ánh một cách cô đọng và chính xác ý nghĩa ngữ nghĩa toàn diện của văn bản gốc, cho phép máy tính hiểu, so sánh và xử lý các câu dựa trên nội dung ý nghĩa thực sự của chúng, vượt xa khả năng phân tích dựa trên từ khóa đơn thuần.

1.3.2 Vai trò của Sentence Transformers trong RAG

- Kiến trúc RAG là một phương pháp mạnh mẽ để kết hợp khả năng tạo sinh của các mô hình ngôn ngữ lớn với khả năng truy xuất thông tin từ kho tri thức bên ngoài. Trong đó, Sentence Transformers đóng vai trò trung tâm và không thể thiếu, đảm

bảo quá trình truy xuất thông tin diễn ra hiệu quả và mang lại các kết quả có liên quan ngữ nghĩa cao. Vai trò chính của Sentence Transformers bao gồm:

- Mã hóa truy vấn:
 - + Khi người dùng đưa ra một truy vấn (câu hỏi hoặc yêu cầu tìm kiếm), Sentence Transformers sẽ đảm nhận nhiệm vụ chuyển đổi truy vấn đó thành một vector ngữ nghĩa (semantic vector) trong không gian vector. Quá trình này đảm bảo rằng ý định và ngữ cảnh của truy vấn được captured một cách chính xác vào dạng số học. Vector truy vấn này sau đó được sử dụng làm "điểm neo" để tìm kiếm các tài liệu hoặc sản phẩm tương tự trong cơ sở dữ liệu vector. Sự mã hóa ngữ nghĩa này cho phép hệ thống tìm kiếm hiệu quả ngay cả khi truy vấn không chứa các từ khóa chính xác nhưng có ý nghĩa tương đồng với nội dung trong kho dữ liệu.
- Mã hóa tài liệu:
 - + Trước khi có thể thực hiện tìm kiếm, toàn bộ kho tài liệu, cơ sở kiến thức hoặc thông tin sản phẩm cần được xử lý tiền kỳ. Mỗi tài liệu, đoạn văn hoặc mô tả sản phẩm sẽ được Sentence Transformers chuyển đổi thành một vector nhúng tương ứng. Các vector này sau đó được lưu trữ trong một cơ sở dữ liệu vector (ChromaDB) cùng với ID và metadata của tài liệu gốc. Việc mã hóa trước này giúp chuẩn bị dữ liệu một cách hiệu quả, cho phép quá trình truy xuất sau này diễn ra nhanh chóng thông qua các phép tính khoảng cách hoặc độ tương đồng giữa các vector.
- Tìm kiếm ngữ nghĩa:
 - + Đây là vai trò then chốt nhất của Sentence Transformers trong RAG. Sau khi truy vấn của người dùng được mã hóa thành vector, hệ thống sẽ thực hiện tìm kiếm NNS hoặc ANN trong cơ sở dữ liệu vector. Bằng cách tính toán độ tương đồng giữa vector truy vấn và tất cả các vector tài liệu đã được mã hóa, Sentence Transformers giúp hệ thống xác định và truy xuất các tài liệu có ý nghĩa ngữ nghĩa gần nhất với truy vấn ban đầu. Khác với tìm kiếm dựa trên từ khóa truyền thống chỉ khớp các từ chính xác, tìm kiếm ngữ nghĩa cho phép hệ thống hiểu được ý nghĩa ẩn sâu của câu hỏi, từ đó trả về các kết quả có tính liên quan cao hơn, ngay cả khi chúng không chứa các từ khóa trùng lặp. Điều này cải thiện đáng kể chất lượng của thông tin được cung cấp cho LLM để tạo ra phản hồi cuối cùng.

1.3.3 Kết luận

Tóm lại, việc tích hợp Sentence Transformers vào kiến trúc RAG là một bước tiến quan trọng trong việc xây dựng các hệ thống truy xuất thông tin và tạo sinh ngôn ngữ thông minh. Chúng không chỉ nâng cao độ chính xác và hiệu quả của quá trình truy xuất thông tin bằng cách cho phép tìm kiếm dựa trên ngữ nghĩa, mà còn đảm bảo rằng các kết quả được truy xuất thực sự liên quan đến ý định và ngữ cảnh của truy vấn người dùng, từ đó cung cấp dữ liệu đầu vào chất lượng cao cho các mô hình ngôn ngữ lớn, giúp chúng tạo ra các phản hồi chính xác, toàn diện và có tính tương thích ngữ cảnh cao hơn.

1.4 Reciprocal Rank Fusion (RRF)

1.4.1 Định nghĩa

- Reciprocal Rank Fusion là một thuật toán kết hợp xếp hạng mạnh mẽ và hiệu quả, được thiết kế để tổng hợp các danh sách kết quả xếp hạng từ nhiều nguồn hoặc hệ thống tìm kiếm riêng biệt thành một danh sách xếp hạng tổng hợp duy nhất. Khác với các phương pháp kết hợp tuyến tính đơn giản hoặc yêu cầu huấn luyện phức tạp, RRF hoạt động dựa trên nguyên tắc đơn giản là ưu tiên các mục (tài liệu, sản phẩm) xuất hiện ở vị trí cao trong nhiều danh sách xếp hạng khác nhau.

1.4.2 Công thức

- Cơ chế hoạt động của RRF tập trung vào việc gán điểm cho mỗi mục dựa trên thứ hạng nghịch đảo của nó trong từng danh sách xếp hạng riêng lẻ. Điểm số cuối cùng của một mục là tổng các điểm số này trên tất cả các nguồn tìm kiếm. Công thức toán học của thuật toán RRF được định nghĩa như sau:

$$RRF(d) = \sum_{i=1}^n \frac{1}{k + r_i(d)}$$

Công thức 1.1 Thuật toán RRF

Trong đó:

- + d : Là một tài liệu /sản phẩm cụ thể
- + n : Số lượng danh sách xếp hạng (các nguồn tìm kiếm khác nhau)
- + $r_i(d)$: Là thứ hạng của tài liệu/sản phẩm d trong danh sách xếp hạng thứ i

- + k là một hằng số dương (thường lấy giá trị 60). Hằng số này có vai trò làm giảm ảnh hưởng của các tài liệu xuất hiện ở vị trí rất thấp (thứ hạng cao) và làm cho đóng góp của thứ hạng cao (vị trí thấp) trở nên quan trọng hơn..
- + $RRF(d)$: Là tổng điểm của tài liệu/sản phẩm.
- Quá trình tổng hợp và xếp hạng:
 - + Tính điểm cá nhân: Với mỗi sản phẩm /tài liệu được trả về từ ít nhất một trong các phương pháp tìm kiếm (từ khóa hoặc ngữ nghĩa), hệ thống sẽ tính toán điểm RRF của nó bằng cách áp dụng công thức trên.
 - + Tổng hợp điểm: Các điểm số từ tất cả các phương pháp tìm kiếm được cộng lại để tạo thành tổng điểm RRF cho mỗi sản phẩm.
 - + Xếp hạng cuối cùng: Các sản phẩm sau đó được sắp xếp theo thứ tự giảm dần của tổng điểm RRF. Sản phẩm nào có tổng điểm RRF cao nhất sẽ được ưu tiên hiển thị ở vị trí đầu tiên trong danh sách kết quả cuối cùng. Điều này đảm bảo rằng các sản phẩm liên quan nhất, xuất hiện ở vị trí cao trong nhiều phương pháp tìm kiếm khác nhau, sẽ được ưu tiên hiển thị cho người dùng.

1.4.3 Ưu điểm của RRF

- RRF đã chứng minh hiệu quả vượt trội trong nhiều ứng dụng tìm kiếm do những ưu điểm nổi bật sau:
 - + Đơn giản và dễ triển khai: Một trong những lợi thế lớn nhất của RRF là không yêu cầu quá trình huấn luyện phức tạp, dữ liệu huấn luyện đặc biệt, hay điều chỉnh nhiều siêu tham số. Điều này giúp giảm đáng kể thời gian và công sức triển khai so với các phương pháp kết hợp dựa trên học máy.
 - + Hiệu quả cao: Nghiên cứu thực nghiệm, đặc biệt là trong các cuộc thi TREC (Text REtrieval Conference), đã chỉ ra rằng RRF thường đạt được hiệu suất vượt trội hoặc tương đương với các phương pháp kết hợp xếp hạng phức tạp hơn, bao gồm cả các mô hình học máy được huấn luyện kỹ lưỡng. Điều này được thể hiện qua các số liệu như Mean Average Precision (MAP) và Normalized Discounted Cumulative Gain (NDCG).
 - + Tăng cường độ đa dạng và độ bao phủ: Bằng cách kết hợp các danh sách từ nhiều nguồn khác nhau (tìm kiếm ngữ nghĩa nắm bắt ý nghĩa, tìm kiếm từ khóa nắm bắt sự chính xác), RRF tận dụng sức mạnh bổ trợ của từng phương pháp. Điều này giúp giảm nguy cơ bỏ sót các tài liệu quan trọng mà chỉ một

trong các phương pháp có thể nhận diện được, từ đó tăng cường độ bao phủ và đa dạng của kết quả tìm kiếm.

- + Chống nhiễu: RRF ít bị ảnh hưởng bởi lỗi hoặc sự thiếu chính xác của một nguồn tìm kiếm đơn lẻ. Nếu một nguồn cung cấp kết quả kém chất lượng, điểm số thấp từ nguồn đó sẽ bị làm yếu đi bởi các điểm số cao từ các nguồn khác, giúp duy trì chất lượng tổng thể của danh sách xếp hạng cuối cùng.
- + Đảm bảo các ưu tiên liên quan nhất: Công thức của RRF tự động ưu tiên các tài liệu xuất hiện ở vị trí cao trong nhiều danh sách xếp hạng. Điều này đồng nghĩa với việc các sản phẩm thực sự liên quan đến truy vấn của người dùng (được xác nhận bởi nhiều phương pháp) sẽ có khả năng cao xuất hiện ở những vị trí đầu tiên, nâng cao chất lượng trải nghiệm tìm kiếm.
- + Giảm thiểu rủi ro bỏ sót: RRF giảm thiểu rủi ro bỏ sót các sản phẩm quan trọng chỉ vì một phương pháp tìm kiếm đơn lẻ không nhận diện được chúng (ví dụ: một truy vấn ngữ nghĩa không khớp hoàn hảo với từ khóa, hoặc ngược lại). Điều này đặc biệt quan trọng trong thương mại điện tử, nơi việc tìm thấy sản phẩm phù hợp là yếu tố rất quan trọng.

1.4.4 Kết luận

- Reciprocal Rank Fusion nổi bật như một thuật toán kết hợp xếp hạng mạnh mẽ và thực dụng, đóng vai trò không thể thiếu trong việc xây dựng các hệ thống tìm kiếm lai (hybrid search) hiện đại. Khác biệt với các phương pháp phức tạp yêu cầu huấn luyện hoặc điều chỉnh tham số tinh vi, RRF mang lại một giải pháp đơn giản nhưng cực kỳ hiệu quả để tổng hợp kết quả từ nhiều nguồn tìm kiếm khác nhau.
- Thông qua cơ chế gán điểm dựa trên thứ hạng nghịch đảo, RRF ưu tiên một cách thông minh các tài liệu hoặc sản phẩm xuất hiện ở vị trí cao trong nhiều danh sách xếp hạng riêng lẻ. Điều này không chỉ đảm bảo rằng những kết quả thực sự liên quan và được xác nhận bởi nhiều phương pháp sẽ được ưu tiên hiển thị, mà còn khắc phục những hạn chế cố hữu của việc chỉ dựa vào một phương pháp tìm kiếm duy nhất.
- Đối với website bán hàng, việc áp dụng RRF không chỉ tối ưu hóa trải nghiệm của người dùng bằng cách cung cấp các kết quả chính xác và toàn diện hơn, mà còn giảm thiểu rủi ro bỏ sót những sản phẩm quan trọng.

Chương 2: PHƯƠNG PHÁP ĐỀ XUẤT XÂY DỰNG CHATBOT VÀ HỆ THỐNG GỌI Ý SẢN PHẨM

2.1 Chatbot

2.1.1 Giới thiệu

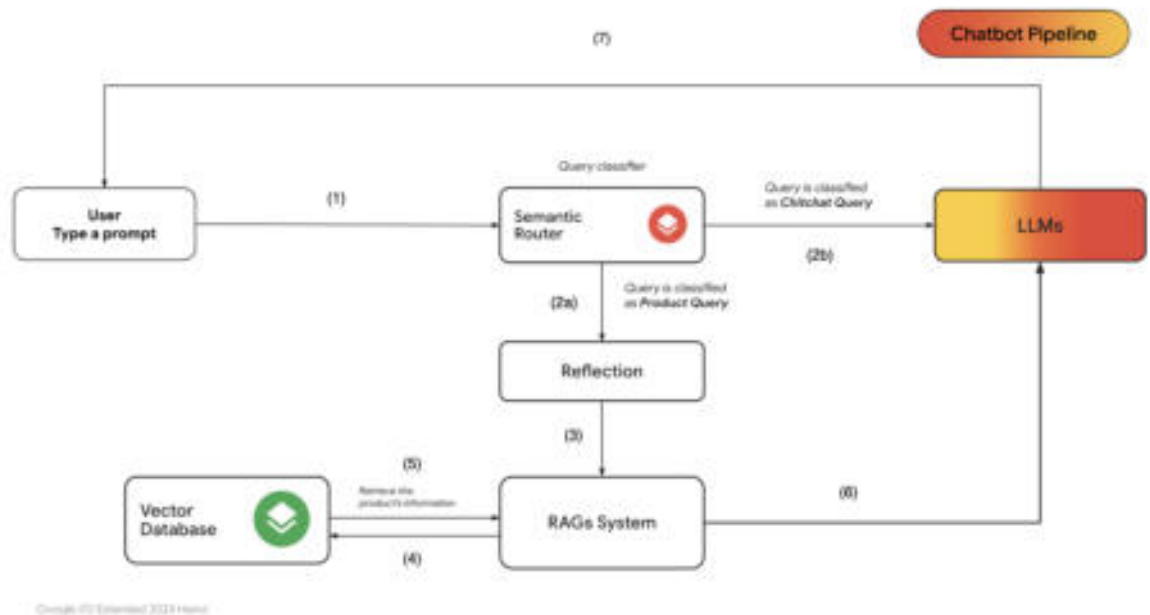
- Trong bối cảnh thương mại điện tử ngày càng đòi hỏi trải nghiệm khách hàng được cá nhân hóa và tức thì, việc tích hợp các hệ thống chatbot thông minh trở nên vô cùng quan trọng. Hệ thống chatbot được đề xuất này không chỉ là một công cụ giao tiếp đơn thuần mà là một giải pháp tương tác ngôn ngữ tự nhiên tiên tiến, được xây dựng dựa trên kiến trúc RAG. Sự lựa chọn kiến trúc này cùng với sự kết hợp của nhiều công nghệ hiện đại nhằm đảm bảo khả năng truy vấn tri thức hiệu quả, phản hồi nhanh chóng và đặc biệt là khả năng thích ứng linh hoạt với các loại truy vấn khác nhau, từ hội thoại thông thường đến các yêu cầu thông tin sản phẩm chuyên sâu. Dưới đây là phân tích chi tiết về các thành phần công nghệ và vai trò của chúng trong quá trình vận hành hệ thống:
 - + Thư viện Genmini AI: Dùng để gọi API của Genimini và nhận phản hồi từ mô hình LLM (Genmini-1.5-flash). Model cung cấp sức mạnh xử lý ngôn ngữ cho chatbot, giúp nó tạo ra các phản hồi tự nhiên, phong phú và đa dạng.
 - + Vector Database - ChromaDB: Được sử dụng để lưu trữ các siêu dữ liệu (metadata) liên quan, ChromaDB hỗ trợ việc tạo và tìm kiếm các embeddings từ các văn bản. Điều này giúp chatbot có thể tham chiếu lại dữ liệu một cách hiệu quả khi trả lời các câu hỏi có tính chất ngữ nghĩa phức tạp.
 - + Flask API: Được dùng để triển khai API backend, phục vụ việc xử lý và truyền tải các phản hồi từ chatbot. Flask hỗ trợ giao tiếp giữa các thành phần của ứng dụng, đảm bảo dữ liệu và phản hồi từ mô hình được truyền tải chính xác đến giao diện người dùng.
 - + Mô hình tạo vector embedding – Sentence Transformers: Để có thể truy xuất thông tin ngữ nghĩa, hệ thống sử dụng mô hình SentenceTransformer (thuộc thư viện HuggingFace Transformers) để chuyển đổi văn bản (từ truy vấn người dùng và tài liệu) thành các vector số có tính biểu diễn ngữ nghĩa cao. Những embedding này là đầu vào cho quá trình tìm kiếm văn bản tương tự

trong vector database. Đây là bước quan trọng trong pipeline RAG, đảm bảo rằng chatbot có thể hiểu ý định và ngữ cảnh của câu hỏi một cách chính xác.

- + Retrieval-Augmented Generation: RAG là kiến trúc kết hợp giữa quá trình truy xuất và sinh phản hồi (generation). Trong hệ thống này, truy vấn từ người dùng trước tiên được embedding và tìm kiếm trong ChromaDB để lấy các tài liệu liên quan, sau đó những tài liệu này được kết hợp với truy vấn gốc để đưa vào mô hình Gemini AI nhằm sinh ra phản hồi đầy đủ và ngữ nghĩa hơn. Kiến trúc này giúp chatbot không chỉ trả lời dựa trên kiến thức có định trong mô hình mà còn linh hoạt tham chiếu đến dữ liệu riêng của hệ thống.
- + Semantic Router: là thành phần có chức năng phân loại truy vấn người dùng. Việc phân loại giúp hệ thống xác định hướng xử lý phù hợp: nếu là chit-chat thì gửi thẳng đến Gemini AI, nếu là truy vấn sản phẩm thì sử dụng RAG để truy xuất tri thức rồi sinh ra phản hồi.
 - Chitchat: Đối với các truy vấn giao tiếp tự nhiên, mang tính xã hội hoặc trò chuyện thông thường (không yêu cầu truy xuất kiến thức cụ thể từ cơ sở dữ liệu sản phẩm), Semantic Router sẽ điều hướng trực tiếp truy vấn đó đến mô hình Gemini AI để tạo phản hồi nhanh chóng và tự nhiên.
 - Product (truy vấn sản phẩm): Khi truy vấn có ý định tìm kiếm thông tin cụ thể từ dữ liệu đã lưu trữ, Semantic Router sẽ chuyển hướng truy vấn đến luồng RAG, nơi nó sẽ được embedding, truy xuất tài liệu liên quan từ ChromaDB, và sau đó được sử dụng bởi Gemini AI để tạo ra câu trả lời chi tiết.
- + Reflection là module quản lý lịch sử hội thoại, có khả năng lưu trữ các truy vấn trước đó cũng như phản hồi đã sinh. Điều này giúp cải thiện trải nghiệm người dùng bằng cách:
 - Giảm thời gian phản hồi cho những câu hỏi lặp lại.
 - Ghi nhớ ngữ cảnh trong một chuỗi hội thoại.
 - Tăng khả năng phản hồi cá nhân hóa.

2.1.2 Mô tả hệ thống

- Hệ thống chatbot được triển khai thông qua một API Flask, đóng vai trò là điểm cuối cho giao tiếp với người dùng. Các thành phần được sắp xếp theo một pipeline xử lý logic như sau:



Hình 2.1 Sơ đồ hoạt động của Chatbot

- Luồng hoạt động của chatbot:
 - + (1) Người dùng gửi truy vấn: Người dùng nhập câu hỏi hoặc yêu cầu tương tác với chatbot thông qua giao diện người dùng.
 - + (2) Phân loại truy vấn bởi Semantic Router: API Flask chuyển truy vấn đến Semantic Router. Semantic Router phân tích ý định ngữ nghĩa của truy vấn và phân loại nó thành một trong hai luồng chính: “chitchat” hoặc “product”:
 - (2a) Nếu là truy vấn "product": Truy vấn được chuyển hướng đến Module Reflection lưu lại vào MongoDB.
 - (2b) Nếu là truy vấn "chitchat": Truy vấn được gửi thẳng đến LLM (Gemini AI) để tạo phản hồi trực tiếp, vì nó không yêu cầu truy xuất kiến thức chuyên sâu.
 - + (3) Quản lý ngữ cảnh bởi Reflection (cho truy vấn “product”): Module Reflection tiếp nhận truy vấn. Hệ thống sẽ lưu lại lịch sử trò chuyện của người dùng (có thể trong bộ nhớ tạm thời hoặc cơ sở dữ liệu) và nếu cần, sử

dụng ngữ cảnh từ lịch sử để bổ sung cho truy vấn hiện tại trước khi chuyển đến hệ thống RAG.

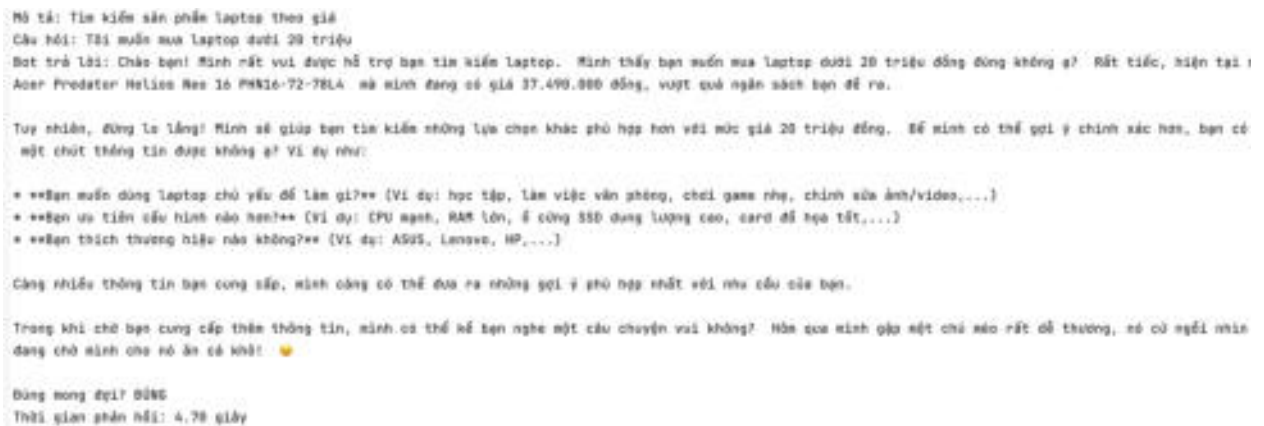
- + (4) Truy vấn Vector Database từ RAGs System: Hệ thống RAG tiếp nhận truy vấn. Nó sử dụng mô hình tạo vector embedding Sentence Transformers để chuyển đổi truy vấn thành một vector nhúng. Vector này sau đó được dùng để truy vấn ChromaDB.
- + (5) Trả về thông tin từ Vector Database: ChromaDB thực hiện tìm kiếm ngữ nghĩa dựa trên vector truy vấn và trả về các tài liệu/thông tin sản phẩm có liên quan nhất cho hệ thống RAG.
- + (6) Tạo ra phản hồi bởi LLMs từ RAGs System: Hệ thống RAG tổng hợp thông tin được truy xuất từ ChromaDB và truy vấn gốc của người dùng để tạo thành một prompt đầu vào cho LLM (Gemini AI). LLM sử dụng prompt này để sinh ra câu trả lời đầy đủ, có ngữ cảnh và ngữ nghĩa.
- + (7) LLMs trả về kết quả cho Người dùng: LLM gửi phản hồi đã được tạo sinh trở lại qua API Flask, và API Flask truyền tải kết quả cuối cùng đến giao diện người dùng để hiển thị cho người dùng.
- Vai trò cụ thể của từng phần:
 - + Genmini AI và Embedding Model:
 - Sử dụng thư viện GenminiAI để tạo các câu trả lời thông qua mô hình LLM.
 - Mô hình nhúng Sentence Transformer được tích hợp thông qua ChromaDB để tạo các nhúng văn bản cho truy vấn, phục vụ truy vấn ngữ nghĩa.
 - + ChromaDB - Vector Database:
 - ChromaDB lưu trữ các vector của dữ liệu để phục vụ cho việc tìm kiếm và truy vấn ngữ nghĩa. Lưu trữ dưới dạng embedding vector, documents và metadatas, hỗ trợ tìm kiếm hiệu quả dựa trên độ tương đồng về ngữ nghĩa.
 - + Retrieval-Augmented Generation :
 - Xây dựng `hybrid_vector_serach()` để query dữ liệu trong vector database embed của ChromaDB.
 - RAG truy vấn nhúng của câu hỏi và lấy thông tin có liên quan, đảm bảo câu trả lời có độ chính xác cao và có ngữ cảnh.

- + Semantic Router:
 - Semantic Router sử dụng các mẫu (sample) để phân loại truy vấn của người dùng thành hai loại chính: chitchat và product.
 - Các truy vấn chitchat sẽ được xử lý trực tiếp qua LLM, trong khi các truy vấn product sẽ được chuyển tới RAG để lấy thông tin sản phẩm.
- + Reflection:
 - Bộ phản hồi đảm bảo chatbot lưu trữ lịch sử hội thoại vào bộ nhớ và có thể lưu cache cho các truy vấn sản phẩm, cải thiện khả năng phản hồi nhanh cho các câu hỏi tương tự trong tương lai.
- + API Endpoint:
 - Flask API nhận truy vấn từ người dùng qua endpoint `/api/v1/chatting`. Dựa vào phân loại truy vấn từ Semantic Router, API sẽ chọn phương pháp phản hồi phù hợp (sử dụng RAG cho câu hỏi về sản phẩm, hoặc phản hồi trực tiếp qua LLM cho câu hỏi chitchat).

2.1.3 Thực nghiệm và đánh giá kết quả

- Thực nghiệm:
 - + Mục tiêu: là kiểm tra khả năng hoạt động thực tế của chatbot AI trong hệ thống thương mại điện tử, đánh giá chất lượng phản hồi, tốc độ xử lý, khả năng hiểu và duy trì ngữ cảnh.
 - + Các kịch bản kiểm thử:
 - Truy vấn sản phẩm: Người dùng hỏi về thông tin sản phẩm, so sánh sản phẩm, hỏi về giá, tính năng, hình ảnh, v.v.
 - Chitchat: Người dùng có thể trò chuyện với chatbot một cách tự do, thoải mái về các chủ đề như thời tiết, phim ảnh, ...
 - Truy vấn hỗn hợp: Người dùng hỏi về sản phẩm được lồng ghép trong đoạn hội thoại tự nhiên.
 - Truy vấn lặp lại: Người dùng hỏi lại các câu hỏi đã từng hỏi để kiểm tra khả năng phản hồi từ cache/lịch sử.
 - Truy vấn phức tạp: Người dùng hỏi các câu hỏi yêu cầu tổng hợp thông tin từ nhiều nguồn.

- Các tiêu chí đánh giá:
 - + Độ chính xác của phản hồi: Đánh giá chat bot có trả lời đúng và đầy đủ thông tin sản phẩm hay không.
 - + Khả năng hiểu ngữ cảnh: Chatbot có nhận diện đúng loại truy vấn (chitchat hay product) và chọn pipeline xử lý phù hợp không.
 - + Tốc độ phản hồi: Đo thời gian từ khi người dùng gửi truy vấn đến khi nhận được phản hồi.
 - + Khả năng lưu trữ và sử dụng lịch sử hội thoại: Chatbot có nhớ và sử dụng được thông tin từ các lượt chat trước không.
- Thực nghiệm:



Hình 2.2 Testcase tìm kiếm sản phẩm theo giá

- + Tìm kiếm sản phẩm theo giá:
 - Câu hỏi: "Tôi muốn mua laptop dưới 20 triệu"
 - Thời gian phản hồi: 6.23 giây
 - Phản hồi của Chatbot: "Chào bạn! Để tôi giúp bạn tìm laptop dưới 20 triệu. Bạn có thể cho tôi biết thêm về mục đích sử dụng (học tập, làm việc, gaming), ưu tiên cấu hình (CPU, RAM), hay kích thước màn hình mong muốn không? Điều này sẽ giúp tôi đề xuất chính xác hơn."
 - Phân tích: Phản hồi phù hợp và thông minh. Chatbot không chỉ xác nhận yêu cầu mà còn chủ động thu thập thêm thông tin để cá nhân hóa gợi ý, thể hiện khả năng hiểu ý định và quy trình tư vấn. Thời gian phản hồi 6.23 giây là chấp nhận được cho một truy vấn RAG phức tạp.

```
Mô tả: Hỏi về sản phẩm mới
Câu hỏi: Có điện thoại nào mới ra không?
Bot trả lời: Chào bạn! Mình rất vui được hỗ trợ bạn. Hiện tại mình chưa có thông tin về điện thoại mới ra mắt trong hệ thống của cửa hàng. D
Laptop, và dường như có sự nhầm lẫn về giá cả trong dữ liệu đó (ví dụ: Laptop ASUS ExpertBook có giá 14.600.000 đồng và 100.000 đồng cùng lúc)
liệu để cung cấp thông tin chính xác nhất cho bạn.

Tuy nhiên, để giúp bạn, mình có thể cung cấp thông tin về các Laptop đang có sẵn:

* **Laptop ASUS ExpertBook B1 81402CVA N61280:** (Giá cả đang cần được kiểm tra lại, xin lỗi về sự bất tiện này). Thông tin chi tiết về cấu h
ou khi mình kiểm tra lại dữ liệu.

* **Laptop Lenovo IdeaPad Slim 3 15A8RR 82XN00RCVN:** (Giá cả đang cần được kiểm tra lại, xin lỗi về sự bất tiện này). Thông tin chi tiết về cấ
p sau khi mình kiểm tra lại dữ liệu.

Bạn có muốn mình tìm hiểu thêm về các mẫu laptop khác, hoặc có thể cho mình biết bạn đang tìm kiếm loại điện thoại nào không? Ví dụ, bạn thích
ng bao nhiêu? Biết thêm thông tin sẽ giúp mình hỗ trợ bạn tốt hơn.

Trong khi chờ cập nhật thông tin về điện thoại và giá cả laptop, bạn có muốn mình kể cho bạn nghe một câu chuyện vui không? Hoặc bạn muốn mình
? Mình thấy hôm nay trời đẹp lắm đó!

Đừng mong đợi? SỬNG
Thời gian phản hồi: 3.09 giây
```

Hình 2.3 Testcase hỏi về sản phẩm

- + Hỏi về sản phẩm mới:
 - Câu hỏi: "Có điện thoại nào mới ra không?"
 - Thời gian phản hồi: 2.19 giây
 - Phản hồi của Chatbot: "Rất tiếc, hiện tại tôi chưa có thông tin cụ thể về các mẫu điện thoại mới ra mắt trong kho dữ liệu của mình. Bạn có muốn tìm hiểu về các dòng điện thoại hiện có hoặc các ưu đãi khác không?"
 - Phân tích: Phản hồi trung thực và có tính hỗ trợ. Chatbot thừa nhận không có thông tin trong kho dữ liệu, đồng thời đề xuất các lựa chọn thay thế. Tốc độ phản hồi nhanh cho thấy hiệu quả trong việc xử lý truy vấn ngoài phạm vi trực tiếp của RAG hoặc việc LLM nhanh chóng nhận diện tình huống.

```
Mô tả: Hỏi về sản phẩm tốt nhất
Câu hỏi: Máy tính bảng nào tốt nhất?
Bot trả lời: Chào bạn! Mình rất vui được hỗ trợ bạn tìm hiểu về máy tính bảng. Tuy nhiên, thông tin mình có hiện tại chỉ bao gồm hai chiếc laptop ASUS Zenbook Duo và Lenovo I
deaPad Slim 3. Vì vậy, mình không thể trả lời chính xác câu hỏi "Máy tính bảng nào tốt nhất?". Dữ liệu của mình chưa đầy đủ về máy tính bảng.

Để giúp bạn tìm được máy tính bảng tốt nhất, mình cần biết thêm về nhu cầu sử dụng của bạn. Bạn muốn dùng máy tính bảng để làm gì? Ví dụ: xem phim, chơi game, học tập, hay làm
việc? Ngân sách của bạn là bao nhiêu? Bạn thích thương hiệu nào? Cung cấp thêm thông tin bạn cung cấp, mình càng có thể giúp bạn tìm ra lựa chọn phù hợp nhất.

Trong khi đó, mình có thể cung cấp thêm thông tin về hai chiếc laptop mình đang có dữ liệu:

* **Laptop ASUS Zenbook Duo 14 UX4202C 6A1647:** Đây là một chiếc laptop cao cấp của ASUS với màn hình kép độc đáo, rất phù hợp cho những ai cần đa nhiệm mạnh mẽ. Giá bán là
12.000.000 VNĐ. (Mau sắc hiện chưa có trong dữ liệu).

* **Laptop Lenovo IdeaPad Slim 3 15A8RR 82XN00RCVN:** Đây là một chiếc laptop Lenovo thuộc phân khúc tầm trung, phù hợp cho công việc văn phòng và giải trí cơ bản. Giá bán là
100.000 VNĐ (Giá này có vẻ không chính xác, bạn có thể kiểm tra lại thông tin trên website của chúng tôi). (Mau sắc hiện chưa có trong dữ liệu).

Nếu bạn muốn biết thêm thông tin về nhu cầu của bạn để mình có thể tư vấn tốt hơn nhé! Ngoài ra, nếu bạn muốn biết thêm về các sản phẩm khác như điện thoại hay tai nghe, mình
cũng rất sẵn lòng hỗ trợ! Chúc bạn một ngày tốt lành!

Thời gian phản hồi: 3.70 giây
```

Hình 2.4 Testcase hỏi về sản phẩm

- + Yêu cầu tư vấn sản phẩm tốt nhất:
 - Câu hỏi: "Máy tính bảng nào tốt nhất?"
 - Thời gian phản hồi: 3.52 giây
 - Phản hồi của Chatbot: "Rất xin lỗi, hiện tại kho dữ liệu của tôi chủ yếu tập trung vào laptop. Tuy nhiên, nếu bạn quan tâm đến hiệu năng cao và tính di động, các dòng laptop ultrabook như [Tên một số mẫu laptop thay thế] có thể là lựa chọn tốt. Bạn có muốn tôi tìm kiếm thông tin về các mẫu đó không?"
 - Phân tích: Phản hồi hữu ích và chuyên hướng thông minh. Chatbot nhận diện được sản phẩm không có trong kho dữ liệu chính xác (máy tính bảng) nhưng đã khéo léo chuyển hướng sang sản phẩm tương tự (laptop) và đề xuất các lựa chọn thay thế phù hợp với các tiêu chí gián tiếp (hiệu năng cao, di động).

Loại test: chitchat

Mô tả: Câu chào hỏi cơ bản

Câu hỏi: Bạn khỏe không?

Bot trả lời: Cảm ơn bạn đã chia sẻ! Mình luôn sẵn sàng lắng nghe.

Thời gian phản hồi: 0.06 giây

Hình 2.5 Testcase chitchat

- + Chitchat - câu chào hỏi cơ bản:
 - Câu hỏi: "Bạn khỏe không?"
 - Thời gian phản hồi: 0.06 giây
 - Phản hồi của Chatbot: "Cảm ơn bạn đã chia sẻ! Mình luôn sẵn sàng lắng nghe?"
 - Phân tích: Chưa trả lời đúng trọng tâm.

Loại test: chitchat

Mô tả: Câu cảm ơn

Câu hỏi: Cảm ơn bạn đã giúp đỡ

Bot trả lời: Rất vui được phục vụ bạn! Nếu cần hỗ trợ gì thêm, cứ hỏi mình nhé.

Thời gian phản hồi: 0.10 giây

Hình 2.6 Testcase chichat

- + Chitchat - lời cảm ơn:
 - Câu hỏi: "Cảm ơn bạn đã giúp đỡ"
 - Thời gian phản hồi: 0.1 giây
 - Phản hồi của Chatbot: "Rất vui được phục vụ bạn! Nếu cần hỗ trợ gì thêm" cứ hỏi mình nhé."
 - Phân tích: Phản hồi phù hợp. Chatbot nhận diện được ý định "cảm ơn" để đưa ra một lời hỏi đáp xã giao tương ứng.
- Đánh giá tổng thể:
 - + Chất lượng phản hồi của chatbot: Nhìn chung, chất lượng phản hồi cho truy vấn sản phẩm là rất tốt, chi tiết, thân thiện và chuyên nghiệp, đặc biệt là khả năng chủ động thu thập thêm thông tin. Các phản hồi chitchat cũng tự nhiên và khéo léo chuyển hướng. Tuy nhiên, có những trường hợp như lời cảm ơn, khả năng hiểu sắc thái hội thoại cần cải thiện.
 - + Tốc độ phản hồi: Thời gian phản hồi cho truy vấn liên quan đến sản phẩm (trung bình 3.05 giây) là khá nhanh, cho thấy hiệu quả của pipeline RAG. Thời gian phản hồi cho câu hỏi đơn giản (trung bình 1.23 giây) là rất nhanh và phù hợp với kỳ vọng cho tương tác tức thì.
 - + Khả năng hiểu ngữ cảnh: Khả năng nhận diện loại truy vấn của Semantic Router hoạt động hiệu quả trong hầu hết các trường hợp. Module Reflection cũng cho thấy tiềm năng trong việc duy trì ngữ cảnh.
 - + Mức độ hài lòng: Dựa trên các phản hồi mẫu, chatbot mang lại trải nghiệm tích cực trong việc tìm kiếm thông tin sản phẩm.

2.1.4 Kết luận

Chức năng của chatbot AI đáp ứng tốt các yêu cầu về độ chính xác, tốc độ, khả năng hiểu ngữ cảnh và lưu trữ hội thoại. Việc tích hợp ChromaDB giúp tăng hiệu quả truy vấn ngữ nghĩa, còn Gemini AI đảm bảo phản hồi tự nhiên, linh hoạt. Tuy nhiên thời gian để phản hồi các câu hỏi phức tạp còn chậm. Hệ thống có thể mở rộng, dễ bảo trì và sẵn sàng tích hợp thêm các tính năng nâng cao trong tương lai.

2.2 Hệ thống gợi ý sản phẩm

2.2.1 Mô tả hệ thống

- Hệ thống gợi ý sản phẩm đóng vai trò tối quan trọng trong việc cá nhân hóa trải nghiệm mua sắm, giúp người dùng dễ dàng khám phá các sản phẩm phù hợp với sở thích và nhu cầu của họ. Hệ thống được xây dựng dựa trên sự kết hợp của các kỹ thuật truy xuất thông tin tiên tiến và thuật toán hợp nhất xếp hạng, đảm bảo độ chính xác và đa dạng của các gợi ý. Quy trình hoạt động của hệ thống được thiết kế theo các bước sau:
 - + Phân tích sở thích người dùng: Hệ thống thu thập thông tin về lịch sử mua hàng hoặc các tương tác trước đó của người dùng để tạo truy vấn tổng hợp.
 - + Mã hóa truy vấn: Sử dụng Sentence Transformers để chuyển truy vấn tổng hợp thành vector ngữ nghĩa. Vector này là biểu diễn số học của sở thích người dùng trong không gian ngữ nghĩa, cho phép thực hiện các phép tính độ tương đồng với các vector sản phẩm.
 - + Hybrid Search: Thực hiện đồng thời hai phương pháp tìm kiếm trên ChromaDB:
 - Semantic Search: So sánh embedding của truy vấn với embedding của các sản phẩm để tìm ra sản phẩm có ý nghĩa gần nhất với sở thích người dùng.
 - Keyword Search: Tìm các sản phẩm chứa từ khóa liên quan đến sở thích người dùng.
 - + Kết hợp kết quả bằng RRF: Hợp nhất và xếp hạng kết quả từ hai phương pháp tìm kiếm bằng thuật toán Reciprocal Rank Fusion, ưu tiên các sản phẩm xuất hiện ở vị trí cao trong cả hai danh sách.
 - + Bổ sung sản phẩm ngẫu nhiên: Nếu số lượng sản phẩm gợi ý chưa đủ, hệ thống sẽ bổ sung thêm các sản phẩm ngẫu nhiên để đảm bảo người dùng luôn có đủ sự lựa chọn mua hàng.
 - + Trả về danh sách sản phẩm gợi ý: Hệ thống gửi danh sách sản phẩm phù hợp nhất cho người dùng thông qua API.

2.2.2 Thực nghiệm và đánh giá kết quả

- Thực nghiệm:

- + Mục tiêu: Đánh giá khả năng gợi ý sản phẩm phù hợp với sở thích và nhu cầu thực tế của người dùng, kiểm tra tốc độ xử lý và mức độ hài lòng của người dùng với các gợi ý nhận được.
- + Các kịch bản kiểm thử:
 - Người dùng có lịch sử mua hàng: Hệ thống gợi ý các sản phẩm cùng danh mục hoặc liên quan đến sở thích trước đó.
 - Người dùng mới (chưa có lịch sử): Hệ thống gợi ý các sản phẩm ngẫu nhiên.
 - Người dùng thay đổi sở thích: Hệ thống cập nhật gợi ý dựa trên các truy vấn mới nhất.
- Các tiêu chí đánh giá:
 - + Độ phù hợp của gợi ý: Sản phẩm gợi ý có sát với sở thích, nhu cầu của người dùng không.
 - + Tốc độ phản hồi: Thời gian từ khi gửi yêu cầu đến khi nhận được danh sách gợi ý.
 - + Độ đa dạng của sản phẩm: Danh sách gợi ý có đa dạng về chủng loại, thương hiệu, giá cả không.
 - + Mức độ hài lòng của người dùng: Đánh giá chủ quan hoặc khảo sát nhanh từ người dùng thử nghiệm.
- Kết quả thực nghiệm:

```
DEBUG - interests: ['laptop', 'gaming', 'dell']
[DEBUG] ChromaDB collection count: 23
DEBUG - recs from hybrid_search: 2 [{'_id': '10', 'title': 'Laptop ASUS Zenbook S 14 OLED UX5406SA PV140WS', 'de
DEBUG - recommendations: 2
DEBUG - CALLING get_random_products
DEBUG - Tổng số sản phẩm trong ChromaDB: 23
Kịch bản: Người dùng có lịch sử mua hàng rõ ràng
```

Hình 2.7 Kết quả thực nghiệm

- + Sở thích: ['laptop', 'gaming', 'dell']
- + Số lượng sản phẩm trong ChromaDB: 23
- + Kết quả từ hybrid_search: 2 sản phẩm liên quan:
 - Laptop ASUS Zenbook S 14 OLED UX5406SA PV140WS
 - Laptop Lenovo IdeaPad Slim 3 15ABR8 82XM00MCVN
- + Hệ thống bổ sung thêm 3 sản phẩm ngẫu nhiên để đủ 5 gợi ý.
- + Đánh giá:

- Độ phù hợp: 2/5 sản phẩm đầu tiên liên quan trực tiếp đến sở thích (laptop, gaming, dell), đúng với kỳ vọng.
- Độ đa dạng: 3 sản phẩm còn lại là random, giúp tăng sự đa dạng nhưng có thể không sát sở thích.

```
PASSED [100%]
DEBUG - get_recommendations CALLED
DEBUG - interests: []
DEBUG - recommendations: 0
DEBUG - CALLING get_random_products
DEBUG - Tổng số sản phẩm trong ChromaDB: 23
Kịch bản: Người dùng mới (chưa có lịch sử)
{'id': '7', 'title': 'Laptop ASUS Zenbook 14 UX3405CA PZ284WS', 'description': '', 'selling_price': 36990000, 'images': [{"ht
{'id': '14', 'title': 'Laptop ASUS Zenbook 14 OLED UX3402VA WM283W', 'description': '', 'selling_price': 23890000, 'images': '
{'id': '15', 'title': 'Laptop gaming Acer Predator Helios Neo 16 PHN16 72 78L4', 'description': '', 'selling_price': 37490000,
{'id': '17', 'title': 'Laptop gaming Gigabyte G5 MF5 H2VN353KH', 'description': '', 'selling_price': 21290000, 'images': [{"h
{'id': '18', 'title': 'Laptop Dell XPS 9350 XPS9350-USIA165W11GR-PP', 'description': '', 'selling_price': 54990000, 'images':
```

Hình 2.8 Kết quả thực nghiệm

- + Sở thích: []
- + Không có sản phẩm nào từ hybrid_search.
- + Hệ thống trả về 5 sản phẩm random
- + Đánh giá:
 - Độ phù hợp: Đúng với kỳ vọng cho user mới, hệ thống gợi ý các sản phẩm phổ biến/ngẫu nhiên.
 - Độ đa dạng: Sản phẩm đa dạng về thương hiệu và giá cả.

```
recommendation_test.py::test_user_changed_interest PASSED [100%]
DEBUG - interests: ['smartphone', 'gaming_laptops', 'tai nghe']
[DEBUG] ChromaDB collection count: 23
DEBUG - recs from hybrid_search: 2 [{'_id': '20', 'title': 'Tai nghe Logitech G733 LIGHTSPEED Wireless Black', 'desc
DEBUG - recommendations: 2
DEBUG - CALLING get_random_products
DEBUG - Tổng số sản phẩm trong ChromaDB: 23
Kịch bản: Người dùng thay đổi sở thích
```

Hình 2.9 Kết quả thực nghiệm

- + Sở thích: ['smartphone', 'gaming_laptops', 'tai nghe']
- + Số lượng sản phẩm trong ChromaDB: 23
- + Kết quả từ hybrid_search: 2 sản phẩm liên quan
 - Tai nghe Logitech G733 LIGHTSPEED Wireless Black (tai nghe, đúng sở thích)
 - Laptop Lenovo IdeaPad Slim 3 15ABR8 82XM00MCVN

- + Hệ thống bổ sung thêm 3 sản phẩm ngẫu nhiên để đủ 5 gợi ý.
- + Đánh giá:
 - Độ phù hợp: sản phẩm đầu tiên liên quan trực tiếp đến sở thích mới của người dùng.
 - Độ đa dạng: Danh sách gợi ý bao gồm cả sản phẩm liên quan đến sở thích mới và các sản phẩm ngẫu nhiên, đảm bảo không bị lặp lại và phù hợp với nhiều nhu cầu khác nhau.

2.2.3 Kết luận

Hệ thống gợi ý sản phẩm đã đáp ứng tốt khi người dùng thay đổi sở thích, ưu tiên các sản phẩm liên quan và bổ sung thêm sản phẩm ngẫu nhiên để đảm bảo danh sách gợi ý luôn đầy đủ và đa dạng.

Chương 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

3.1 Tác nhân

Bảng dưới đây mô tả tất cả các tác nhân của hệ thống bán hàng trực tuyến. Một tác nhân đại diện cho một vai trò; hiện tại hệ thống có các vai trò chính gồm: Admin và User.

Mỗi người dùng có những thao tác riêng khi truy cập vào hệ thống, tùy thuộc vào quyền hạn của mình:

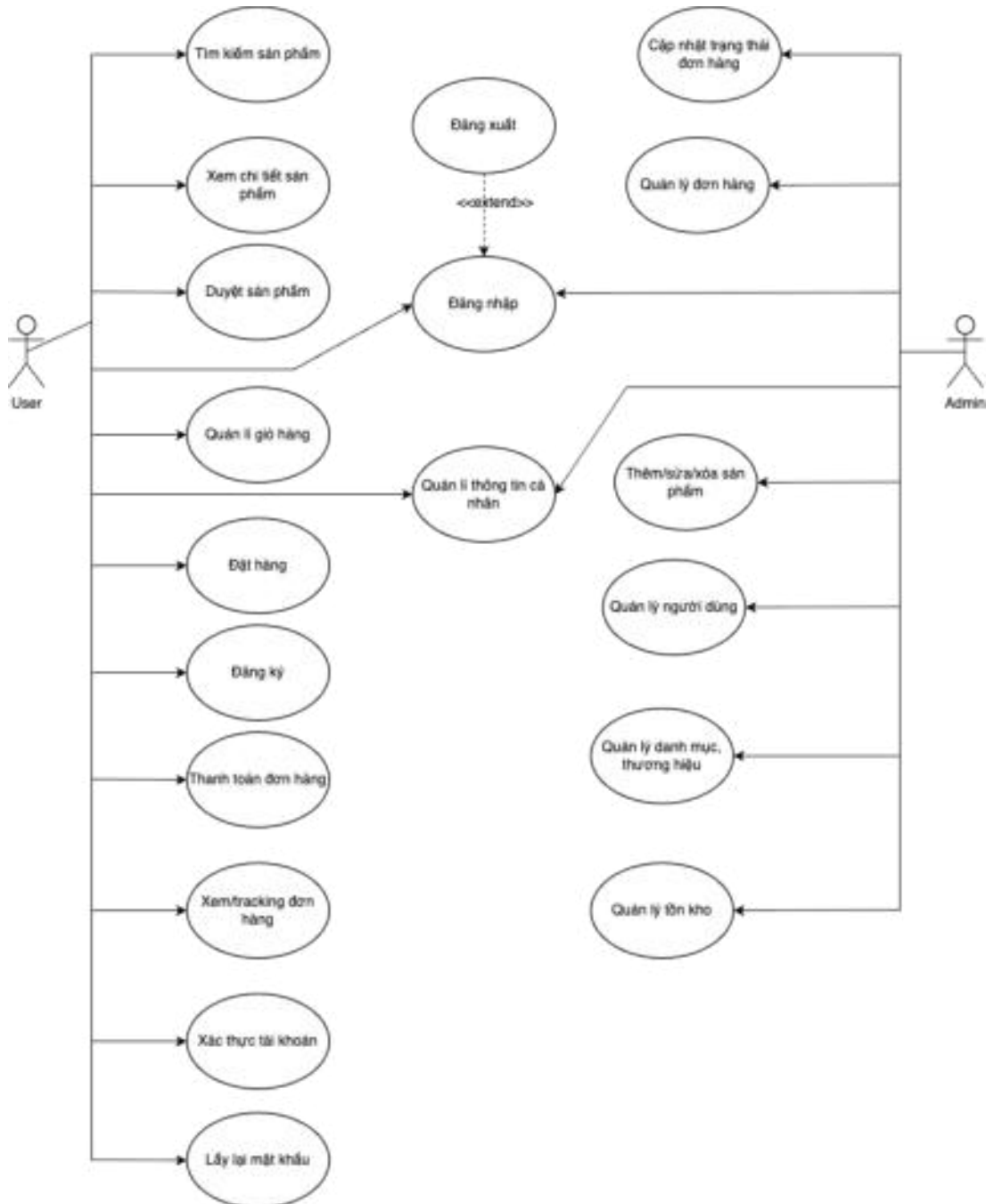
Bảng 3.1 Các tác nhân

Tác nhân	Mô tả
Admin	<ul style="list-style-type: none">- Quản lý sản phẩm: Admin có khả năng thêm mới, chỉnh sửa thông tin chi tiết.- Quản lý danh mục sản phẩm: Tạo, chỉnh sửa và sắp xếp các danh mục sản phẩm để tối ưu hóa khả năng tìm kiếm và duyệt sản phẩm của người dùng.- Quản lý đơn hàng: Theo dõi và xử lý tất cả các đơn hàng từ khi được đặt cho đến khi hoàn tất. Admin có thể xem chi tiết đơn hàng, cập nhật trạng thái đơn hàng.- Quản lý Người dùng: Xem, chỉnh sửa và quản lý thông tin tài khoản của tất cả người dùng trong hệ thống. Admin có thể kích hoạt hoặc vô hiệu hóa tài khoản, đặt lại mật khẩu và xử lý các vấn đề liên quan đến tài khoản người dùng.- Xem thống kê doanh thu và số đơn hàng theo các tháng trong một năm.
User	<ul style="list-style-type: none">- Đăng ký- Đăng nhập- Đăng xuất

	<ul style="list-style-type: none">- Quản lý giỏ hàng: Thêm sản phẩm vào giỏ hàng, cập nhật số lượng sản phẩm, xóa sản phẩm khỏi giỏ hàng. Người dùng có thể xem tổng giá trị các sản phẩm trong giỏ hàng trước khi tiến hành thanh toán.- Quản lý đơn hàng:<ul style="list-style-type: none">+ Mua hàng: Hoàn tất quá trình thanh toán để tạo một đơn hàng mới.+ Xem tình trạng đơn hàng: Theo dõi trạng thái hiện tại của các đơn hàng đã đặt.+ Xem chi tiết đơn hàng: Truy cập thông tin chi tiết của từng đơn hàng đã đặt, bao gồm danh sách sản phẩm, số lượng, giá cả, địa chỉ giao hàng và phương thức thanh toán.- Quản lý profile<ul style="list-style-type: none">+ Update thông tin cá nhân+ Đổi mật khẩu- Xem tình trạng đơn hàng- Quản lý địa chỉ giao hàng:<ul style="list-style-type: none">+ Thêm địa chỉ mới+ Cập nhật thông tin địa chỉ- Trang danh sách sản phẩm:<ul style="list-style-type: none">+ Sắp xếp theo từng thuộc tính sản phẩm+ Lọc theo thuộc tính sản phẩm, nhãn hàng+ Tìm kiếm sản phẩm- Trang chi tiết sản phẩm:<ul style="list-style-type: none">+ Hiển thị thông tin chi tiết+ Ảnh hiển thị theo slider + hover zoom effect+ Thêm sản phẩm vào giỏ hàng- Thanh toán:<ul style="list-style-type: none">+ Thanh toán tiền mặt+ Thanh toán thông qua VNPay
--	---

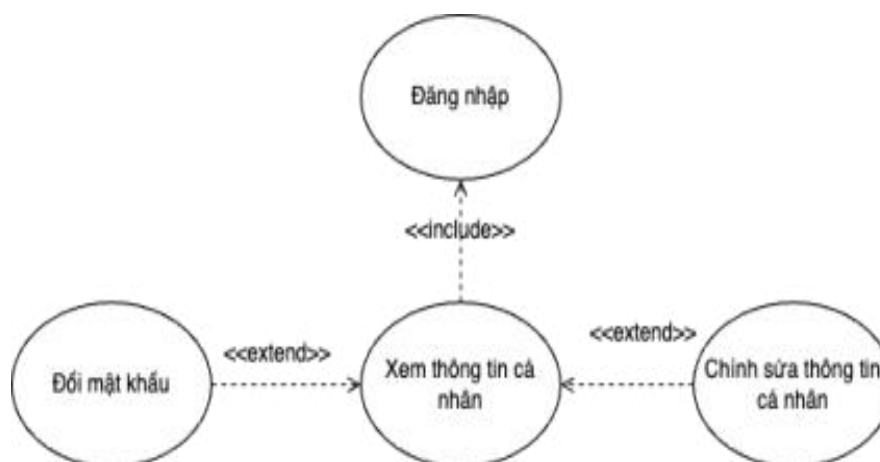
3.2 Sơ đồ use case hệ thống

Sơ đồ này cung cấp một mô hình tổng thể về các nhóm chức năng tương ứng với các tác nhân có trong hệ thống. Các chức năng được chia làm các Usecase sau đây:



Hình 3.1 Sơ đồ usecase tổng quát hệ thống

3.2.1 Admin xem thông tin cá nhân



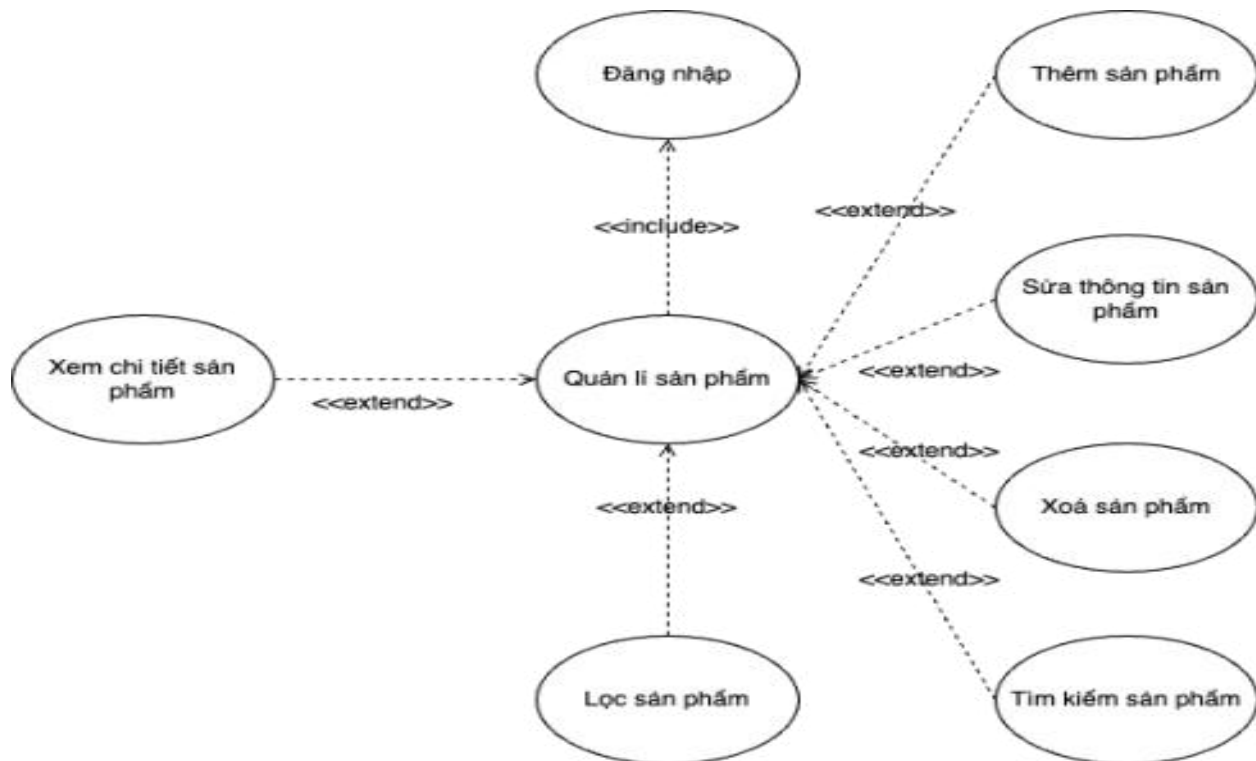
Hình 3.2 Sơ đồ usecase xem thông tin cá nhân

Bảng 3.2 Đặc tả usecase xem thông tin cá nhân

Tên usecase	Admin xem thông tin cá nhân
Tác nhân chính	Admin
Mô tả Usecase	Cho phép admin có thể xem, cập nhật thông tin cá nhân và đổi mật khẩu tài khoản
Tiền điều kiện	Đăng nhập vào hệ thống với quyền admin
Hậu điều kiện	Hệ thống hiện thông báo thành công và hiển thị lại dữ liệu đã cập nhật
Kích hoạt	Khi admin vào ‘Tài khoản’ thì nó sẽ dẫn đến trang cá nhân
	<ul style="list-style-type: none"> - Chuỗi sự kiện: <ul style="list-style-type: none"> + Giao diện bao gồm các chức năng : Cập nhật thông tin cá nhân và đổi mật khẩu - Cập nhật thông tin cá nhân: <ul style="list-style-type: none"> + Hệ thống hiển thị giao diện cá nhân bao gồm 2 nút cập nhật thông tin cá nhân và đổi mật khẩu, mặc định nút cập nhật thông tin cá nhân sẽ được nhấn và mở ra giao diện tương ứng

<ul style="list-style-type: none">+ Admin nhập vào dữ liệu tương ứng và nhấn nút lưu+ Hệ thống xác thực thông tin đầu vào+ Nếu hợp lệ hệ thống sẽ hiển thị thông báo cập nhật thành công+ Nếu không hợp lệ, hệ thống sẽ hiển thị thông báo lỗi+ Trong trường hợp admin nhấn nút cancel+ Hệ thống sẽ reset lại dữ liệu về ban đầu <ul style="list-style-type: none">- Đổi mật khẩu<ul style="list-style-type: none">+ Admin click vào nút đổi mật khẩu+ Hệ thống hiển thị giao diện đổi mật khẩu+ Admin nhập vào dữ liệu tương ứng và nhấn lưu+ Hệ thống xác thực thông tin đầu vào+ Nếu hợp lệ hệ thống sẽ hiển thị thông báo đổi mật khẩu thành công+ Nếu không hợp lệ hệ thống sẽ hiển thị thông báo lỗi	
Ngoại lệ	Hệ thống hủy việc cập nhật khi admin nhấn nút điều hướng trên sidebar

3.2.2 Admin quản lý sản phẩm



Hình 3.3 Sơ đồ usecase quản lý sản phẩm

Bảng 3.3 Đặc tả usecase admin quản lý sản phẩm

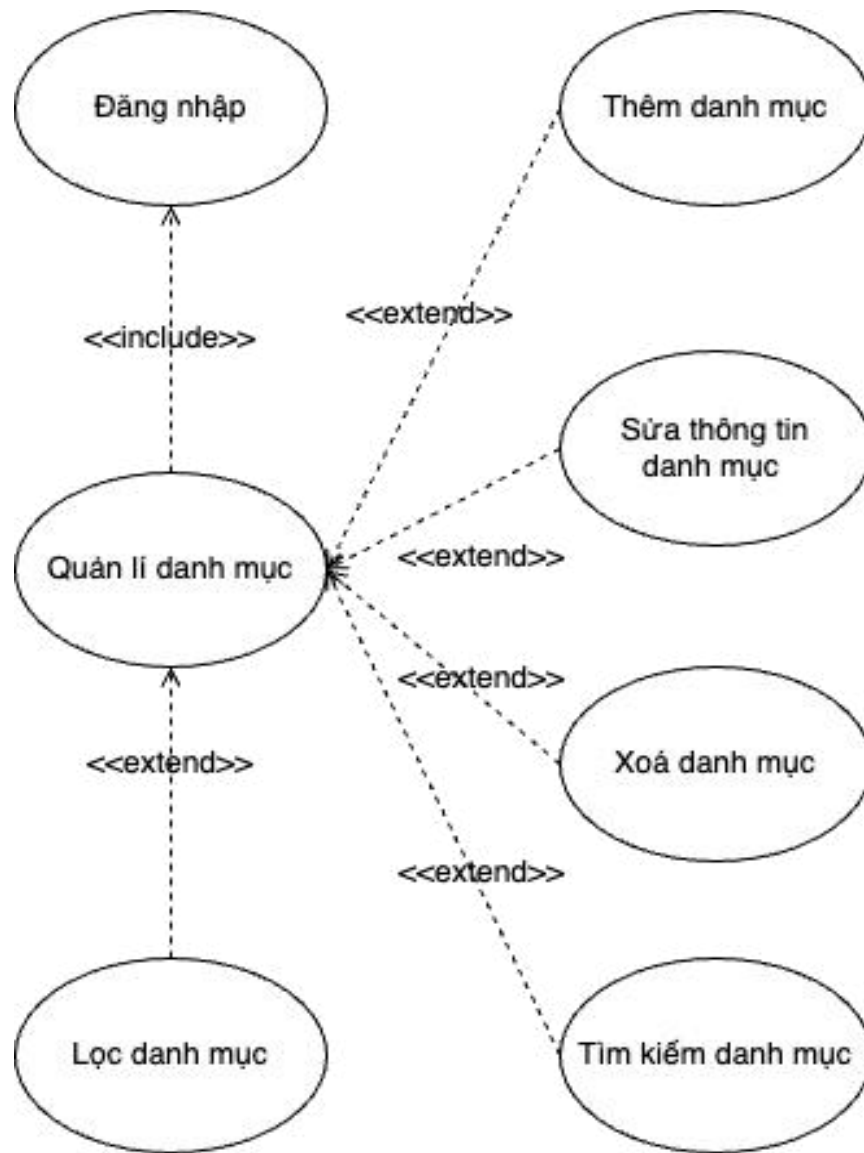
Tên Usecase	Admin quản lý sản phẩm
Tác nhân chính	Admin
Mô tả Usecase	Cho phép admin có thể thêm, sửa, xóa, tìm kiếm, lọc, xem chi tiết sản phẩm
Tiền điều kiện	Đăng nhập vào hệ thống với quyền admin
Hậu điều kiện	Hệ thống cập nhật lại bảng dữ liệu
Kích hoạt	Khi admin chọn quản lý sản phẩm
<ul style="list-style-type: none">- Chuỗi sự kiện:<ul style="list-style-type: none">+ Giao diện bao gồm các chức năng : Thêm, xóa, sửa, tìm kiếm, lọc sản phẩm- Thêm sản phẩm:<ul style="list-style-type: none">+ Hệ thống hiển thị giao diện quản lý sản phẩm+ Admin click vào nút thêm mới sản phẩm+ Hệ thống hiển thị giao diện thêm mới sản phẩm+ Admin nhập vào dữ liệu tương ứng và nhấn nút lưu+ Hệ thống xác thực thông tin đầu vào+ Nếu hợp lệ hệ thống sẽ hiển thị thông báo thêm sản phẩm thành công nếu không hợp lệ sẽ hiển thị thông báo lỗi+ Hệ thống quay lại giao diện quản lý sản phẩm và cập nhật lại bảng dữ liệu+ Trong trường hợp admin nhấn nút cancel+ Hệ thống sẽ hủy bỏ việc thêm mới và quay lại trang quản lý sản phẩm- Sửa sản phẩm:<ul style="list-style-type: none">+ Admin click vào nút action của một record bất kì trong bảng+ Hệ thống hiển thị dropdown gồm chức năng xem, sửa, xóa+ Admin click sửa+ Hệ thống hiển thị giao diện cập nhật sản phẩm+ Admin nhập vào dữ liệu tương ứng và nhấn nút lưu+ Hệ thống xác thực thông tin đầu vào	

- + Nếu hợp lệ hệ thống sẽ hiển thị thông báo cập nhật sản phẩm thành công nếu không hợp lệ sẽ hiển thị thông báo lỗi
- + Hệ thống quay lại giao diện quản lí sản phẩm và cập nhật lại bảng dữ liệu
- + Trong trường hợp admin nhấn nút cancel
- + Hệ thống sẽ hủy bỏ việc sửa mới và quay lại trang quản lí sản phẩm
- Xóa sản phẩm:
 - + Admin click vào nút action của một record bất kì trong bảng
 - + Hệ thống hiển thị dropdown gồm chức năng xem, sửa, xóa
 - + Admin click xóa
 - + Hệ thống hiển thị thông báo bạn có chắc chắn muốn xóa record này
 - + Admin click Có,
 - + Hệ thống hiển thị thông báo xóa thành công và cập nhật lại bảng
 - + Trường hợp admin click cancel
 - + Hệ thống hủy bỏ việc xóa và tắt thông báo
- Xem sản phẩm:
 - + Admin click xem trong dropdown của action bất kì
 - + Hệ thống hiển thị giao diện gồm các thông tin chi tiết của sản phẩm
- Tìm kiếm, lọc sản phẩm:
 - + Admin gõ vào tên sản phẩm cần tìm vào ô search và đồng thời chọn thêm các dữ liệu khác như nhãn hàng, danh mục
 - + Hệ thống sẽ thực hiện tìm kiếm và cập nhật lại bảng với dữ liệu tìm kiếm được

Ngoại lệ

Hệ thống hủy việc cập nhật khi admin nhấn nút điều hướng trên sidebar

3.2.3 Admin quản lí danh mục



Hình 3.4 Sơ đồ usecase admin quản lí danh mục

Bảng 3.4 Đặc tả usecase admin quản lí danh mục

Tên Usecase	Admin quản lí danh mục
Tác nhân chính	Admin
Mô tả Usecase	Cho phép admin có thể thêm, sửa, xóa, tìm kiếm danh mục
Tiền điều kiện	Đăng nhập vào hệ thống với quyền admin

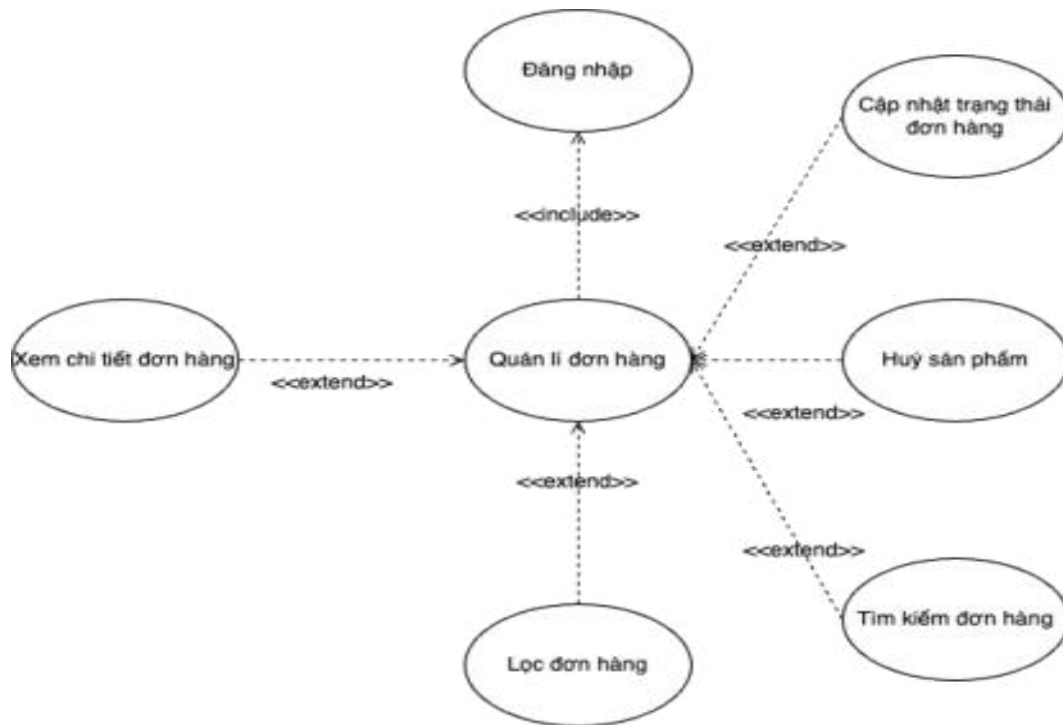
Hậu điều kiện	Admin thêm, xóa, sửa, tìm kiếm danh mục thành công Hệ thống cập nhật lại bảng dữ liệu
Kích hoạt	Khi admin chọn quản lí danh mục
<ul style="list-style-type: none">- Chuỗi sự kiện:<ul style="list-style-type: none">+ Giao diện bao gồm các chức năng : Thêm, xóa, sửa, tìm kiếm danh mục- Thêm danh mục:<ul style="list-style-type: none">+ Hệ thống hiển thị giao diện quản lí danh mục+ Admin click vào nút thêm mới danh mục+ Hệ thống hiển thị giao diện thêm mới danh mục+ Admin nhập vào dữ liệu tương ứng và nhấn nút lưu+ Hệ thống xác thực thông tin đầu vào+ Nếu hợp lệ hệ thống sẽ hiển thị thông báo thêm sản phẩm thành công nếu không hợp lệ sẽ hiển thị thông báo lỗi+ Hệ thống quay lại giao diện quản lí danh mục và cập nhật lại bảng dữ liệu+ Trong trường hợp admin nhấn nút cancel+ Hệ thống sẽ hủy bỏ việc thêm mới và quay lại trang quản lí danh mục- Sửa danh mục:<ul style="list-style-type: none">+ Admin click vào nút action của một record bất kì trong bảng+ Hệ thống hiển thị dropdown gồm chức năng xem, sửa, xóa+ Admin click sửa+ Hệ thống hiển thị giao diện cập nhật danh mục+ Admin nhập vào dữ liệu tương ứng và nhấn nút lưu+ Hệ thống xác thực thông tin đầu vào+ Nếu hợp lệ hệ thống sẽ hiển thị thông báo cập nhật danh mục thành công nếu không hợp lệ sẽ hiển thị thông báo lỗi+ Hệ thống quay lại giao diện quản lí danh mục và cập nhật lại bảng dữ liệu+ Trong trường hợp admin nhấn nút cancel+ Hệ thống sẽ hủy bỏ việc sửa mới và quay lại trang quản lí danh mục- Xóa danh mục:<ul style="list-style-type: none">+ Admin click vào nút action của một record bất kì trong bảng	

- + Hệ thống hiển thị dropdown gồm chức năng xem, sửa, xóa
- + Admin click xóa
- + Hệ thống hiển thị thông báo bạn có chắc chắn muốn xóa record này
- + Admin click Có
- + Hệ thống hiển thị thông báo xóa thành công và cập nhật lại bảng
- + Trường hợp admin click cancel
- + Hệ thống hủy bỏ việc xóa và tắt thông báo
- Tìm kiếm danh mục:
 - + Admin gõ vào tên danh mục cần tìm vào ô search
 - + Hệ thống sẽ thực hiện tìm kiếm và cập nhật lại bảng với dữ liệu tìm kiếm được

Ngoại lệ

Hệ thống hủy việc cập nhật khi admin nhấn nút điều hướng trên sidebar

3.2.4 Admin quản lý đơn hàng

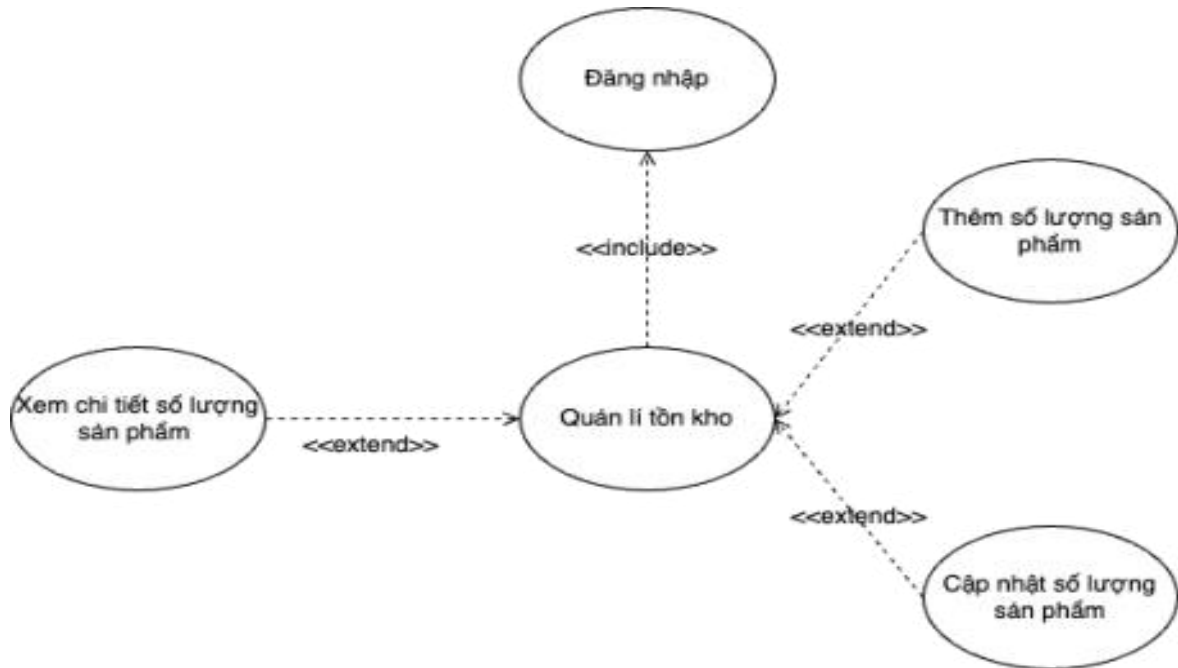


Hình 3.5 Sơ đồ usecase admin quản lý đơn hàng

Bảng 3.5 Đặc tả usecase admin quản lý đơn hàng

Tên Usecase	Admin quản lý đơn hàng
Tác nhân chính	Admin
Mô tả Usecase	Cho phép admin có thể xem chi tiết, cập nhật, lọc và tìm kiếm các đơn hàng
Tiền điều kiện	Đăng nhập vào hệ thống với quyền admin
Hậu điều kiện	Hệ thống cập nhật lại bảng dữ liệu
Kích hoạt	Khi admin chọn quản lý đơn hàng
<ul style="list-style-type: none">- Chuỗi sự kiện:<ul style="list-style-type: none">Giao diện bao gồm các chức năng : xem chi tiết, cập nhật, lọc và tìm kiếm đơn hàng- Cập nhật trạng thái đơn hàng:<ul style="list-style-type: none">+ Hệ thống hiển thị giao diện quản lý đơn hàng+ Admin chọn vào một record bất kì, nhấn action và chọn cập nhật trạng thái đơn hàng+ Hệ thống hiển thị thông báo trạng thái đơn hàng sau khi cập nhật, nếu thành công dữ liệu bảng sẽ được cập nhật lại, ngược lại hiển thị thông báo lỗi- Tìm kiếm, lọc thông tin đơn hàng:<ul style="list-style-type: none">+ Admin gõ vào tên người dùng, id order và lọc ra các order với trạng thái đơn hàng, trạng thái thanh toán, phương thức thanh toán, phương thức giao hàng và khoảng thời gian giao dịch+ Hệ thống sẽ thực hiện tìm kiếm và cập nhật lại bảng với dữ liệu tìm kiếm được- Xem chi tiết đơn hàng:<ul style="list-style-type: none">+ Admin click xem chi tiết đơn hàng trong dropdown của action bất kì+ Hệ thống hiển thị giao diện bao gồm các thông tin có trong đơn hàng- Huỷ đơn hàng:<ul style="list-style-type: none">+ Admin click vào button Huỷ đơn+ Hệ thống hiển thị thông báo huỷ đơn nếu thành công còn nếu không thì sẽ thông báo lỗi	

3.2.5 Admin quản lý tồn kho



Hình 3.6 Sơ đồ usecase admin quản lý tồn kho

Bảng 3.6 Đặc tả usecase admin quản lý tồn kho

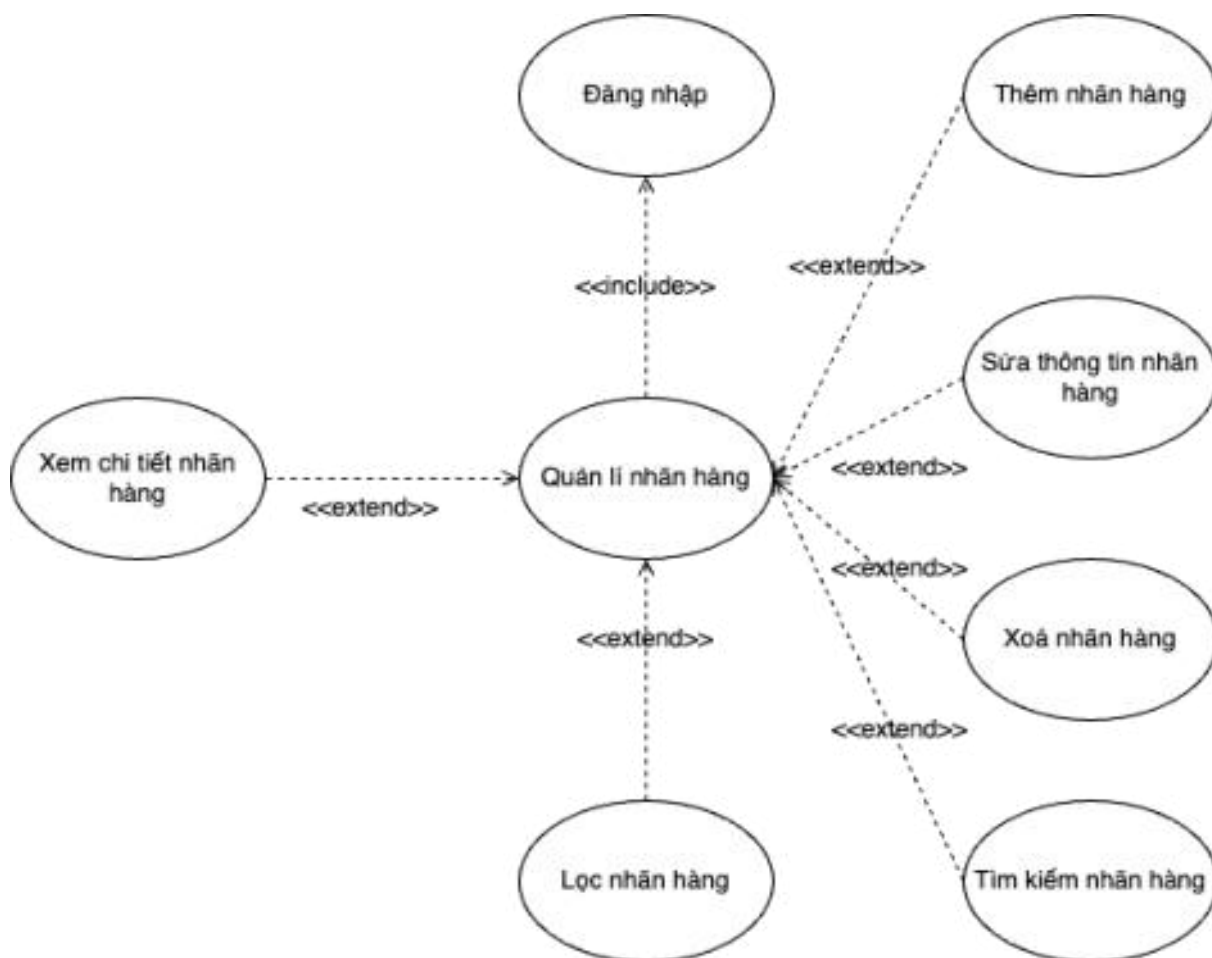
Tên Usecase	Admin quản lý tồn kho
Tác nhân chính	Admin
Mô tả Usecase	Cho phép admin có thể thêm, sửa số lượng sản phẩm
Tiền điều kiện	Đăng nhập vào hệ thống với quyền admin
Hậu điều kiện	Admin thêm, sửa thành công Hệ thống cập nhật lại dữ liệu cho bảng dữ liệu và hiển thị thông báo thành công
Kích hoạt	Khi admin chọn quản lý sản phẩm

- Chuỗi sự kiện: Giao diện bao gồm các chức năng : Thêm, sửa
- Thêm số lượng sản phẩm:
 - + Hệ thống hiển thị giao diện quản lí sản phẩm
 - + Admin click vào nút thêm mới sản phẩm
 - + Hệ thống hiển thị giao diện thêm sản phẩm
 - + Admin nhập vào trường dữ liệu số lượng sản phẩm nhấn nút lưu
 - + Hệ thống xác thực thông tin đầu vào
 - + Nếu hợp lệ hệ thống sẽ hiển thị thông báo thêm thành công nếu không hợp lệ sẽ hiển thị thông báo lỗi
 - + Hệ thống quay lại giao diện quản lí nhập kho và cập nhật lại bảng dữ liệu
 - + Trong trường hợp admin nhấn nút cancel
 - + Hệ thống sẽ hủy bỏ việc thêm mới và quay lại trang quản lí nhập kho
- Sửa số lượng sản phẩm:
 - + Admin click vào button Sửa trong phần Quản lí sản phẩm
 - + Hệ thống hiển thị giao diện gồm các thông tin chi tiết của sản phẩm bao gồm số lượng sản phẩm còn lại, thay đổi dữ liệu của trường đó
 - + Nếu hợp lệ hệ thống sẽ hiển thị thông báo thêm thành công nếu không hợp lệ sẽ hiển thị thông báo lỗi

Ngoại lệ

Sẽ hiển thị thông báo lỗi

3.2.6 Admin quản lý nhân hàng



Hình 3.7 Sơ đồ usecase admin quản lý nhân hàng

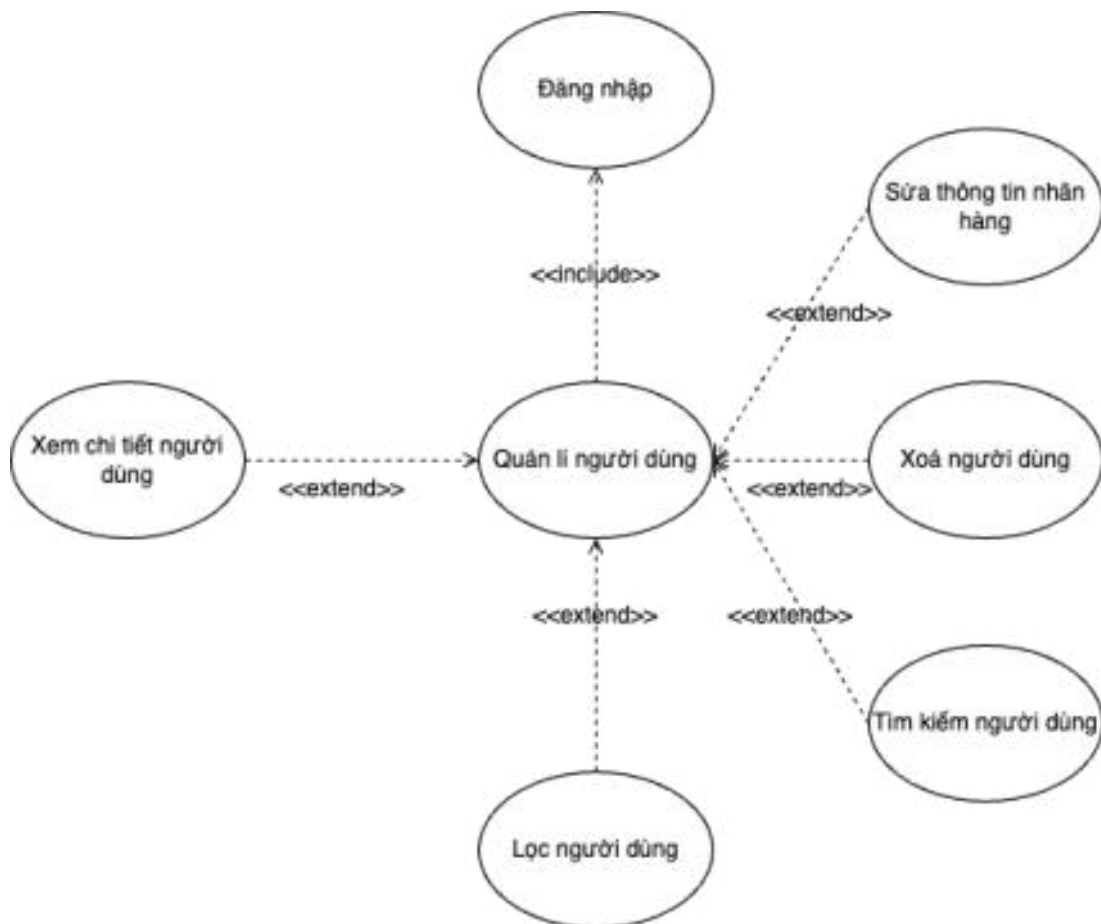
Bảng 3.7 Đặc tả usecase quản lý nhân hàng

Tên Usecase	Admin quản lý nhân hàng
Tác nhân chính	Admin
Mô tả Usecase	Cho phép admin có thể thêm, sửa, xóa, tìm kiếm nhân hàng
Tiền điều kiện	Đăng nhập vào hệ thống với quyền admin
Hậu điều kiện	Admin thêm, xóa, sửa, tìm kiếm nhân hàng thành công Hệ thống cập nhật lại bảng dữ liệu
Kích hoạt	Khi admin chọn quản lý nhân hàng

- Chuỗi sự kiện:
 - + Giao diện bao gồm các chức năng: Thêm, xóa, sửa, tìm kiếm, lọc nhãn hàng
- Thêm nhãn hàng:
 - + Hệ thống hiển thị giao diện quản lí nhãn hàng
 - + Admin click vào nút thêm mới nhãn hàng
 - + Hệ thống hiển thị giao diện thêm mới nhãn hàng
 - + Admin nhập vào dữ liệu tương ứng và nhấn nút lưu
 - + Hệ thống xác thực thông tin đầu vào
 - + Nếu hợp lệ hệ thống sẽ hiển thị thông báo thêm nhãn hàng thành công nếu không hợp lệ sẽ hiển thị thông báo lỗi
 - + Hệ thống quay lại giao diện quản lí nhãn hàng và cập nhật lại bảng dữ liệu
 - + Trong trường hợp admin nhấn nút cancel
 - + Hệ thống sẽ hủy bỏ việc thêm mới và quay lại trang quản lí nhãn hàng
- Sửa thông tin nhãn hàng:
 - + Admin click vào nút action của một record bất kì trong bảng
 - + Hệ thống hiển thị dropdown gồm chức năng xem, sửa, xóa
 - + Admin click sửa
 - + Hệ thống hiển thị giao diện cập nhật nhãn hàng
 - + Admin nhập vào dữ liệu tương ứng và nhấn nút lưu
 - + Hệ thống xác thực thông tin đầu vào
 - + Nếu hợp lệ hệ thống sẽ hiển thị thông báo cập nhật nhãn hàng thành công nếu không hợp lệ sẽ hiển thị thông báo lỗi
 - + Hệ thống quay lại giao diện quản lí nhãn hàng và cập nhật lại bảng dữ liệu
 - + Trong trường hợp admin nhấn nút cancel
 - + Hệ thống sẽ hủy bỏ việc sửa mới và quay lại trang quản lí nhãn hàng
- Xóa nhãn hàng:
 - + Admin click vào nút action của một record bất kì trong bảng
 - + Hệ thống hiển thị dropdown gồm chức năng xem, sửa, xóa
 - + Admin click xóa
 - + Hệ thống hiển thị thông báo bạn có chắc chắn muốn xóa record này

<ul style="list-style-type: none">+ Admin click Có+ Hệ thống hiển thị thông báo xóa thành công và cập nhật lại bảng+ Trường hợp admin click cancel+ Hệ thống hủy bỏ việc xóa và tắt thông báo- Tìm kiếm, lọc nhãn hàng:<ul style="list-style-type: none">+ Admin gõ vào tên nhãn hàng cần tìm vào ô search+ Hệ thống sẽ thực hiện tìm kiếm và cập nhật lại bảng với dữ liệu tìm kiếm được.	
Ngoại lệ	Hệ thống hủy việc cập nhật khi admin nhấn nút điều hướng trên sidebar

3.2.7 Admin quản lý user

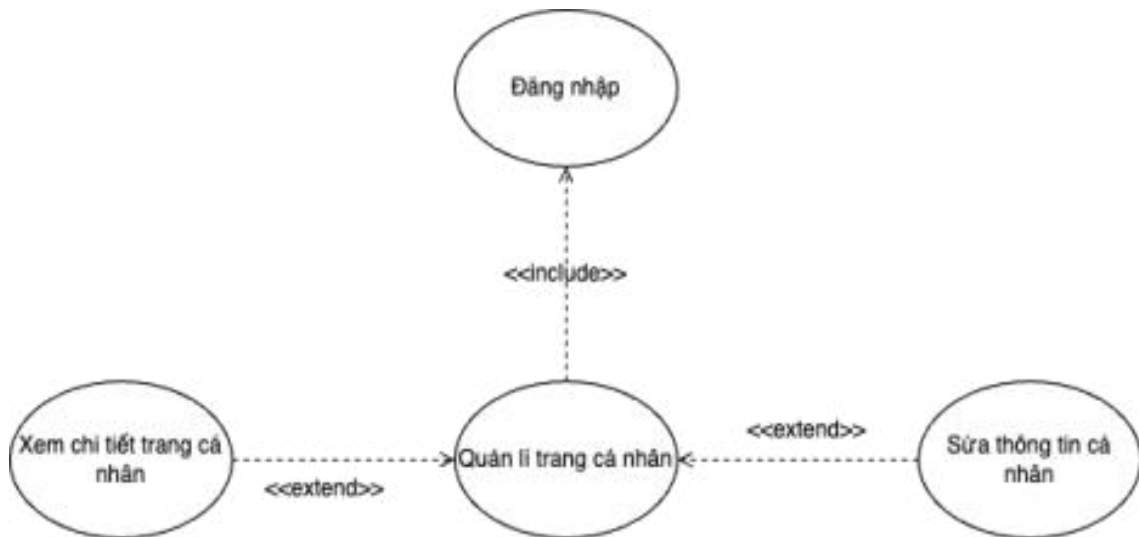


Hình 3.8 Sơ đồ usecase admin quản lý user

Bảng 3.8 Đặc tả usecase quản lí người dùng

Tên Usecase	Admin quản lí user
Tác nhân chính	Admin
Mô tả Usecase	Cho phép admin có thể tìm kiếm, lọc, xóa user
Tiền điều kiện	Đăng nhập vào hệ thống với quyền admin
Hậu điều kiện	Admin tìm kiếm, lọc, chặn user Hệ thống cập nhật lại bảng dữ liệu
Kích hoạt	Khi admin chọn quản lí user
	<ul style="list-style-type: none">- Chuỗi sự kiện:<ul style="list-style-type: none">+ Giao diện bao gồm các chức năng : tìm kiếm, lọc, chặn, xóa user- Tìm kiếm và lọc user:<ul style="list-style-type: none">+ Admin gõ vào tên, email, số điện thoại của user cần tìm và có thể lọc user trạng thái hoạt động, trạng thái bị chặn và ngày user tham gia hệ thống+ Hệ thống sẽ thực hiện tìm kiếm và cập nhật lại bảng với dữ liệu tìm kiếm được.- Chặn/bỏ chặn hoặc xóa/phục hồi user:<ul style="list-style-type: none">+ Admin chọn vào record user bất kì, chọn vào xóa/phục hồi, chặn/ bỏ chặn user+ Hệ thống sẽ hiển thị thông báo tương ứng
Ngoại lệ	Hệ thống hủy việc cập nhật khi admin nhấn nút điều hướng trên sidebar ng hủy việc cập nhật khi admin nhấn nút điều hướng trên sidebar

3.2.8 User quản lí profile



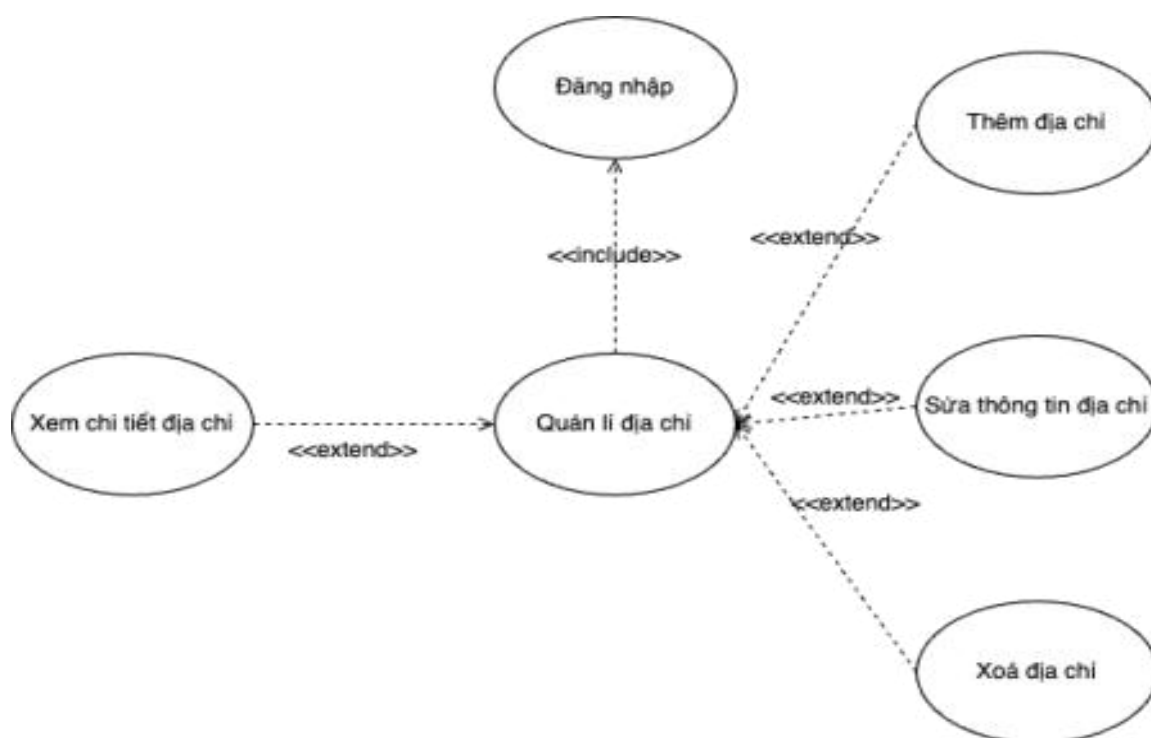
Hình 3.9 Sơ đồ usecase quản lí profile

Bảng 3.9 Đặc tả usecase quản lí profile

Tên Usecase	Quản lí profile
Tác nhân chính	Người dùng (Khách hàng)
Mô tả Usecase	Người dùng thực hiện các thao tác quản lí profile bao gồm xem thông tin profile, chỉnh sửa thông tin.
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống. Profile người dùng đã được tạo trong cơ sở dữ liệu.
Hậu điều kiện	Thông tin profile được cập nhật thành công (nếu chỉnh sửa). Mật khẩu được thay đổi và xác nhận thành công (nếu thay đổi).
Kích hoạt	<ul style="list-style-type: none"> - Người dùng nhấn vào "Quản lí profile" trong giao diện tài khoản. - Thực hiện một trong các hành động sau:

	<ul style="list-style-type: none"> + Xem thông tin profile: Hệ thống hiển thị thông tin hiện tại của người dùng. + Chỉnh sửa thông tin: Nhập thông tin mới và nhấn "Lưu". - Hệ thống xác nhận và thực hiện yêu cầu.
Ngoại lệ	Hiển thị thông báo lỗi

3.2.9 User quản lý địa chỉ nhận hàng



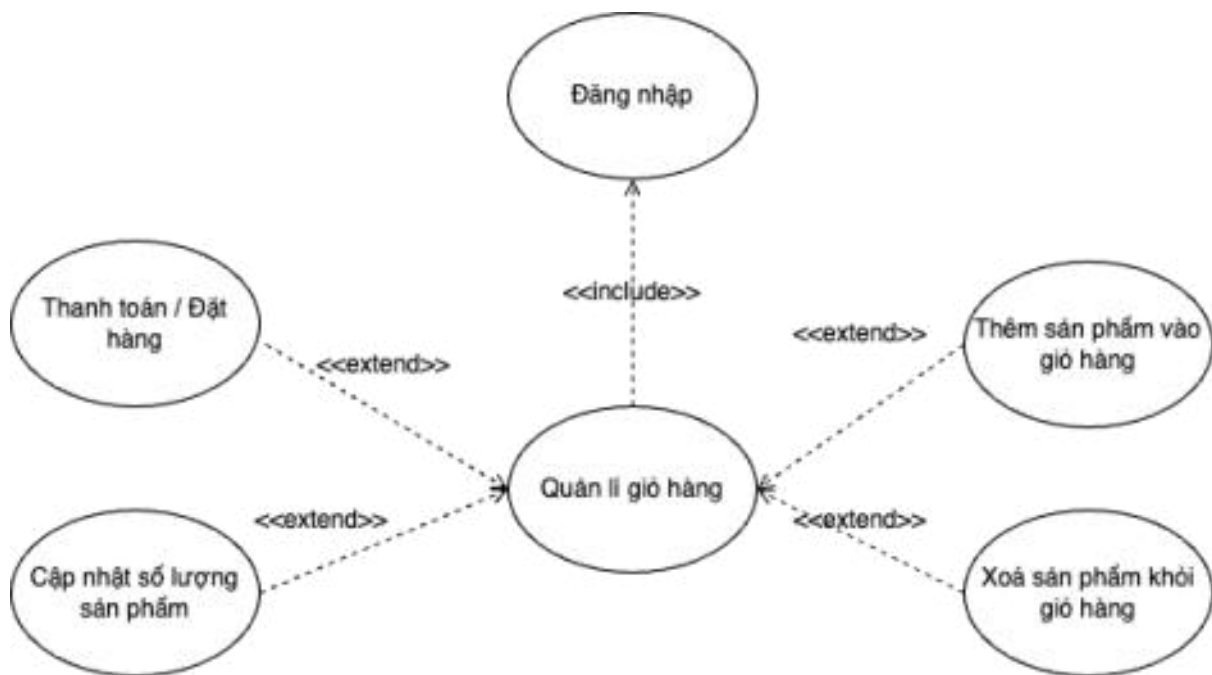
Hình 3.10 Sơ đồ usecase quản lý địa chỉ nhận hàng

Bảng 3.10 Đặc tả usecase quản lý địa chỉ

Tên Usecase	Quản lý địa chỉ nhận hàng
Tác nhân chính	Người dùng (Khách hàng)
Mô tả Usecase	Người dùng thêm mới, sửa, xóa hoặc chọn địa chỉ nhận hàng mặc định trong danh sách các địa chỉ đã lưu trữ.
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống.

Hậu điều kiện	Danh sách địa chỉ nhận hàng được cập nhật thành công (thêm, sửa, xóa hoặc chọn mặc định). Thông tin địa chỉ nhận hàng hiển thị đúng trên giao diện.
Kích hoạt	<ul style="list-style-type: none">- Người dùng nhấn vào "Địa chỉ giao hàng" trong giao diện tài khoản.- Thực hiện một trong các hành động sau:<ul style="list-style-type: none">+ Thêm địa chỉ mới: Nhập thông tin đầy đủ của địa chỉ mới và nhấn "Lưu".+ Sửa địa chỉ: Nhấn vào nút 'Sửa' trong danh sách, cập nhật các thông tin mới và nhấn nút "Lưu".+ Xóa địa chỉ: Nhấn nút "Xóa" kế bên địa chỉ mong muốn.- Hệ thống xác nhận và lưu thay đổi.
Ngoại lệ	Hiển thị thông báo lỗi

3.2.10 User quản lý giỏ hàng

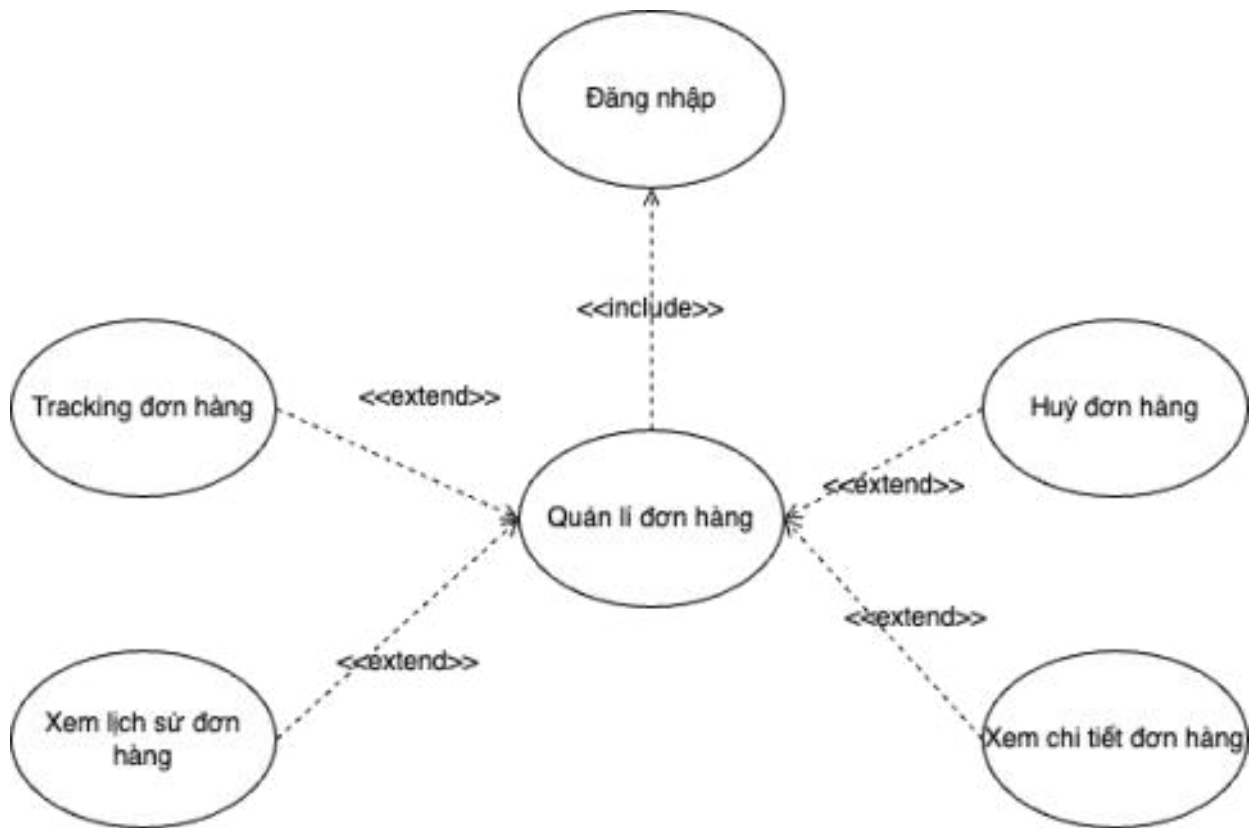


Hình 3.11 Sơ đồ usecase quản lý giỏ hàng

Bảng 3.11 Đặc tả usecase quản lí giỏ hàng

Tên Usecase	Quản lí giỏ hàng
Tác nhân chính	Người dùng (Khách hàng)
Mô tả Usecase	Người dùng thêm, sửa, xóa sản phẩm trong giỏ hàng hoặc cập nhật số lượng sản phẩm.
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống. Hệ thống đã có danh sách sản phẩm hiển thị cho người dùng.
Hậu điều kiện	Giỏ hàng được cập nhật thành công (thêm, sửa, xóa sản phẩm). Thông tin giỏ hàng được lưu trữ và hiển thị đúng trên giao diện.
Kích hoạt	<ul style="list-style-type: none">- Người dùng truy cập giao diện sản phẩm.- Nhấn "Thêm vào giỏ hàng".- Vào mục "Giỏ hàng" và xem danh sách các sản phẩm đã thêm.- Thực hiện một trong các tác vụ:<ul style="list-style-type: none">+ Thay đổi số lượng sản phẩm có trong giỏ hàng+ Xóa sản phẩm: Nhấn nút "Xóa" kế bên sản phẩm để xóa sản phẩm ra khỏi giỏ hàng.
Ngoại lệ	

3.2.11 User quản lý đơn hàng



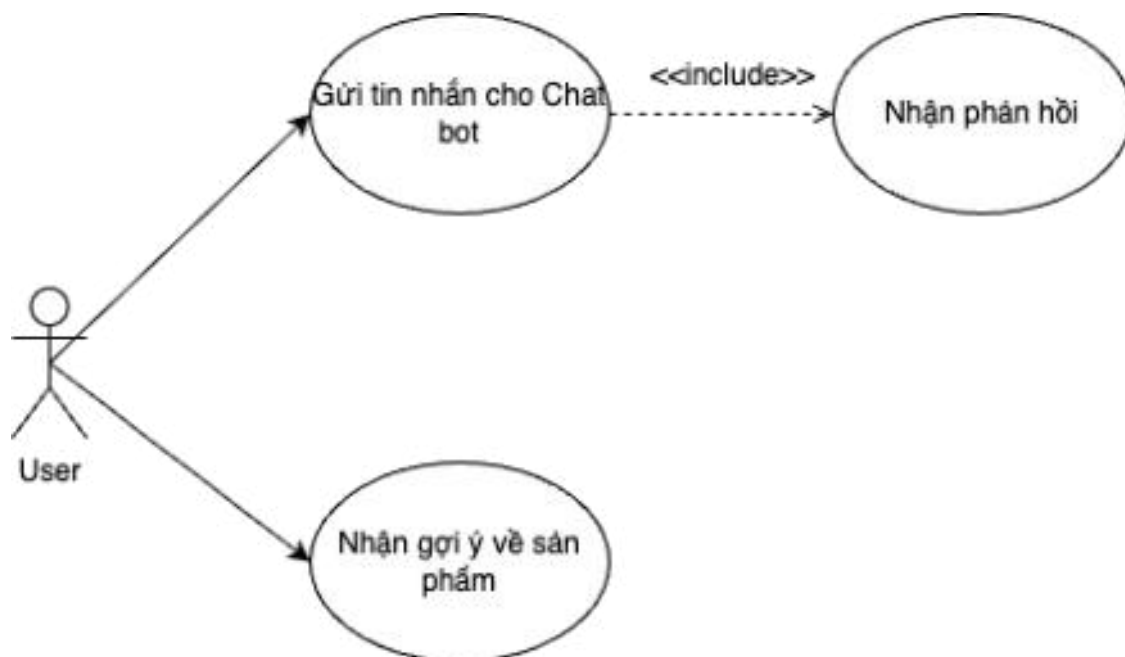
Hình 3.12 Sơ đồ usecase quản lý đơn hàng

Bảng 3.12 Đặc tả usecase quản lý đơn hàng

Tên Usecase	Quản lý đơn hàng
Tác nhân chính	Người dùng (Khách hàng)
Mô tả Usecase	Người dùng thực hiện các thao tác quản lý đơn hàng bao gồm xem lịch sử đơn hàng, hủy đơn hàng, xem chi tiết đơn hàng, và theo dõi tình trạng đơn hàng.
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống. Đã có đơn hàng được tạo trong hệ thống.
Hậu điều kiện	Lịch sử đơn hàng được hiển thị (nếu xem). Đơn hàng đã hủy được cập nhật tình trạng (nếu hủy). Thông tin chi tiết đơn hàng được hiển thị (nếu xem).

	Tình trạng đơn hàng được cập nhật trong giao diện (nếu theo dõi).
Kích hoạt	<ul style="list-style-type: none">- Người dùng nhấn vào "Quản lý đơn hàng" trong giao diện tài khoản.- Thực hiện một trong các hành động sau:<ul style="list-style-type: none">+ Xem lịch sử đơn hàng: Hệ thống hiển thị danh sách các đơn hàng.+ Hủy đơn hàng: Chọn đơn hàng, nhấn "Hủy", nhập lý do (nếu có), và xác nhận.+ Xem chi tiết đơn hàng: Nhấn vào đơn hàng mong muốn, xem thông tin chi tiết.+ Theo dõi đơn hàng: Nhấn vào tình trạng đơn hàng để xem các cập nhật.- Hệ thống xác nhận và thực hiện yêu cầu.

3.2.12 User chatbot

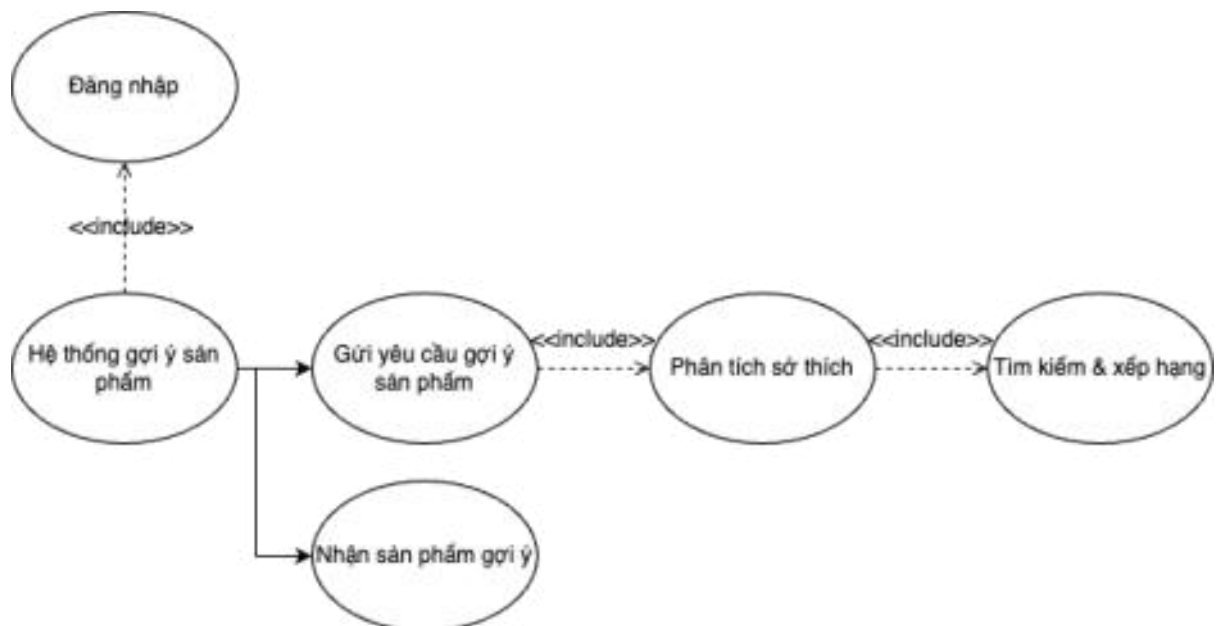


Hình 3.13 Sơ đồ usecase chatbot

Bảng 3.13 Đặc tả usecase chatbot

Tên Usecase	Chatbot
Tác nhân chính	Người dùng
Mô tả Usecase	Người dùng có thể nhập câu hỏi về sản phẩm của cửa hàng hoặc các câu hỏi khác
Tiền điều kiện	Người dùng phải nhập câu hỏi
Hậu điều kiện	Chatbot sẽ trả lời câu hỏi cho người dùng, nếu liên quan đến sản phẩm thì nó sẽ hiển thị các sản phẩm gợi ý
Kích hoạt	Người dùng nhấn vào biểu tượng chatbot để bắt đầu sử dụng Hệ thống nhận câu hỏi và render câu trả lời cho người dùng.

3.2.13 User hệ thống gợi ý sản phẩm



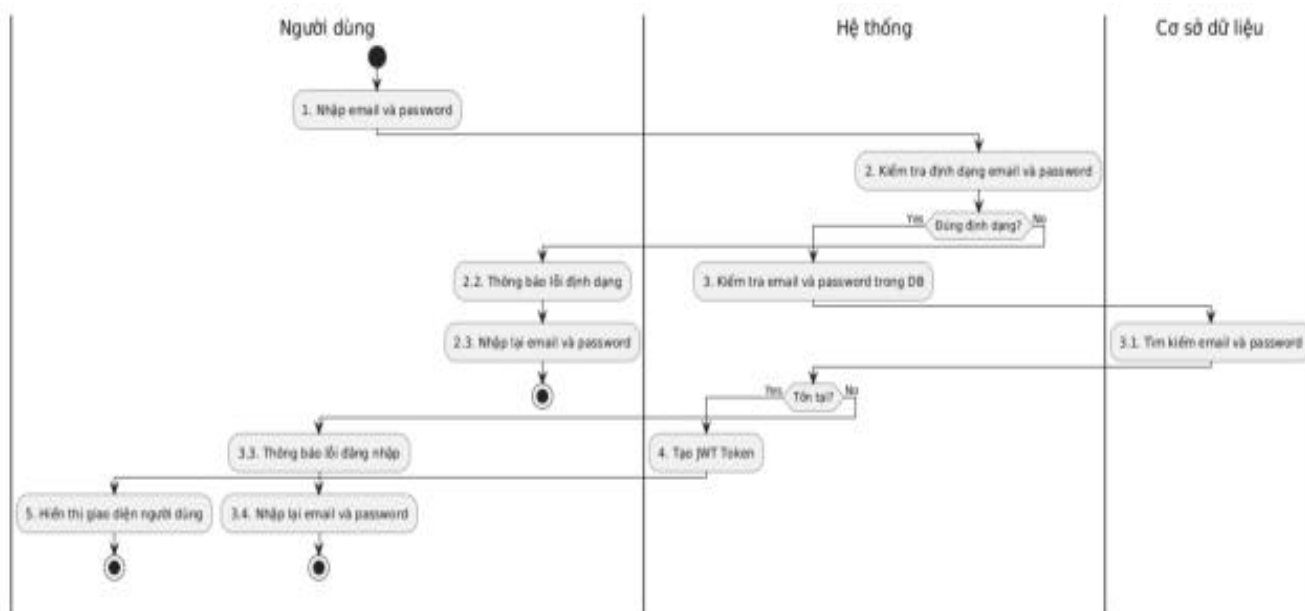
Hình 3.14 Sơ đồ usecase gợi ý sản phẩm

Bảng 3.14 Đặc tả usecase gợi ý sản phẩm

Tên Usecase	Hệ thống gợi ý sản phẩm
Tác nhân chính	Người dùng
Mô tả Usecase	Khi người dùng đăng nhập thì dựa vào sở thích của người dùng thì hệ thống sẽ gợi ý các sản phẩm phù hợp với người dùng đó
Tiền điều kiện	Người dùng phải đăng nhập
Hậu điều kiện	Hệ thống sẽ hiển thị các sản phẩm phù hợp cho người dùng
Kích hoạt	Sau khi người dùng đăng nhập thì hệ thống sẽ phân tích, tổng hợp và hiển thị các sản phẩm theo nhu cầu mua sắm của người dùng

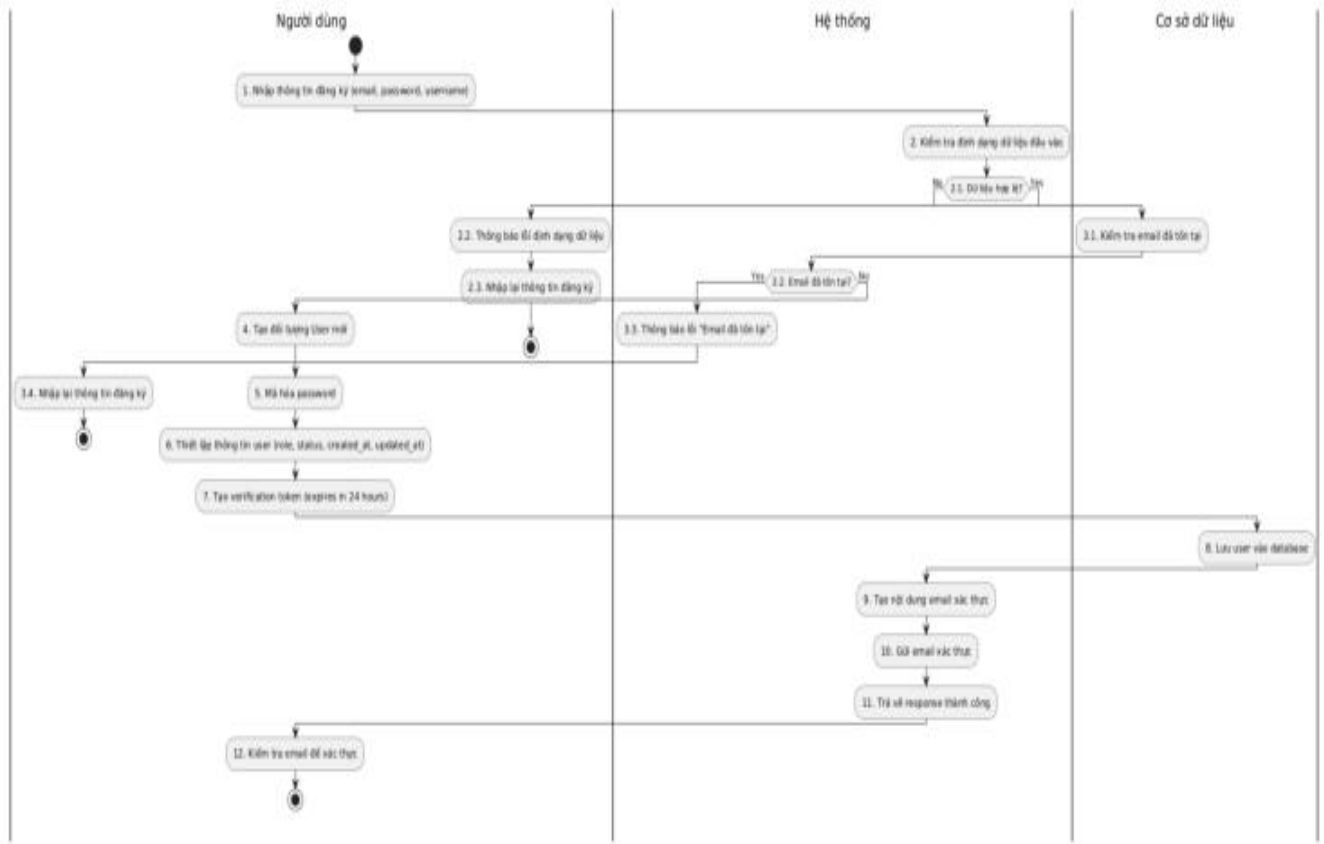
3.3 Biểu đồ hoạt động

3.3.1 Chức năng đăng nhập



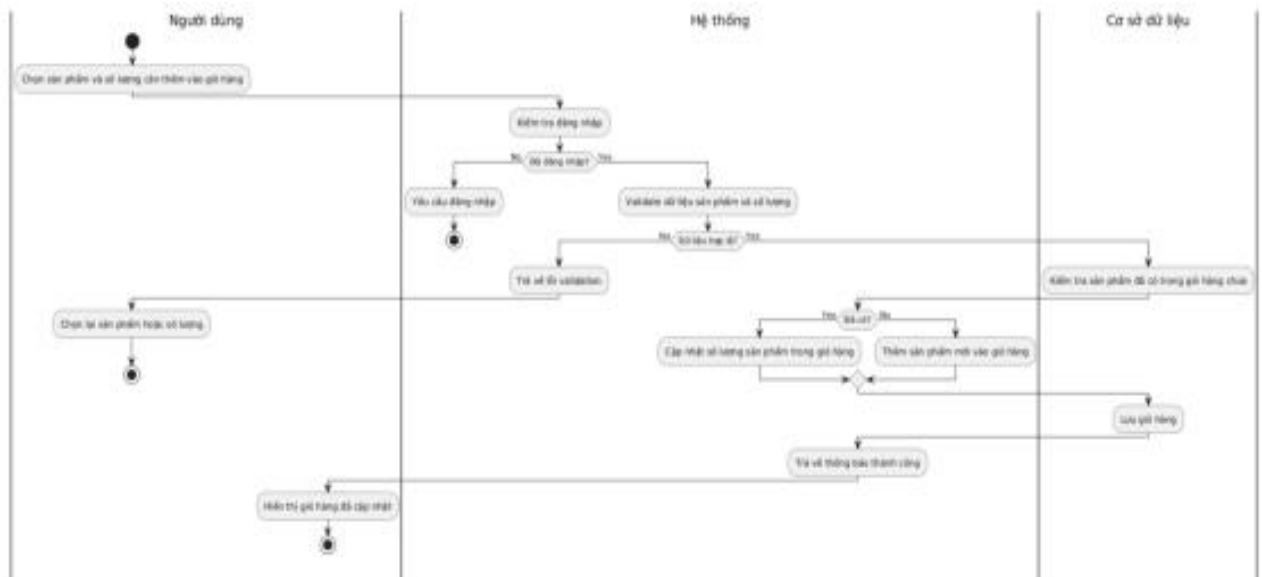
Hình 3.15 Sơ đồ hoạt động đăng nhập

3.3.2 Chức năng đăng kí



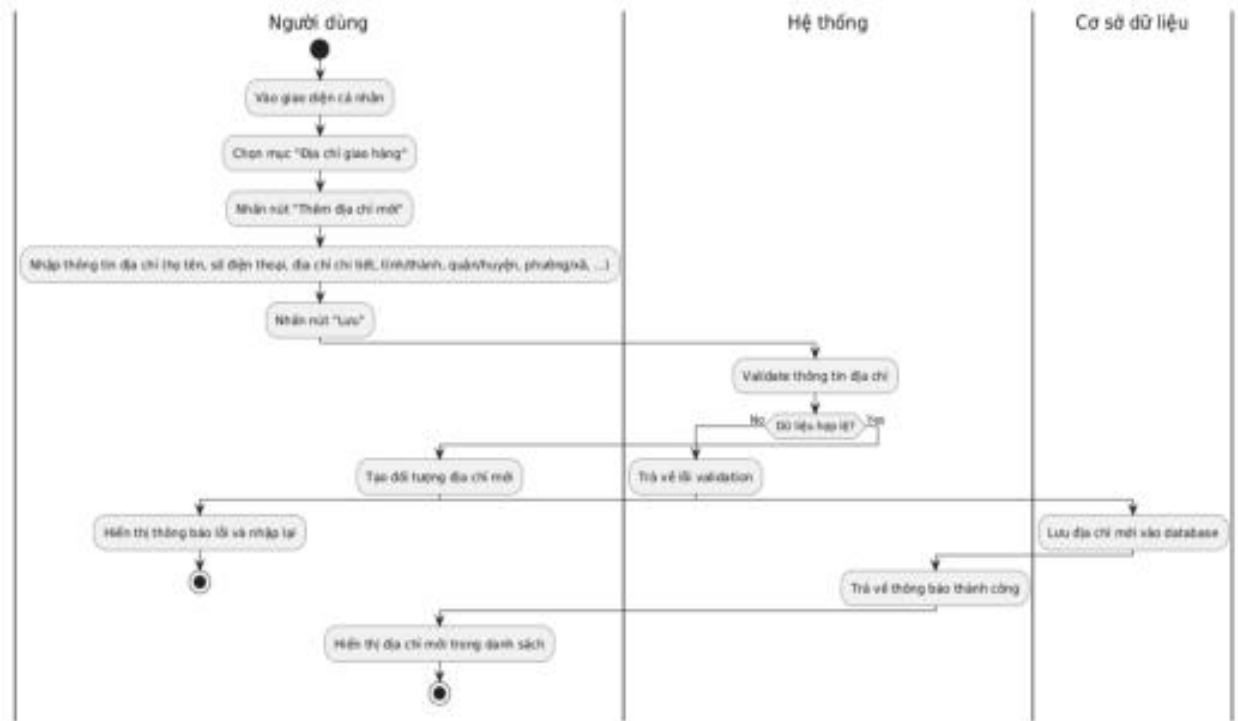
Hình 3.16 Sơ đồ hoạt động đăng kí

3.3.3 Chức năng thêm sản phẩm vào giỏ hàng



Hình 3.17 Sơ đồ hoạt động thêm sản phẩm vào giỏ

3.3.4 Chức năng thêm địa chỉ cho người dùng



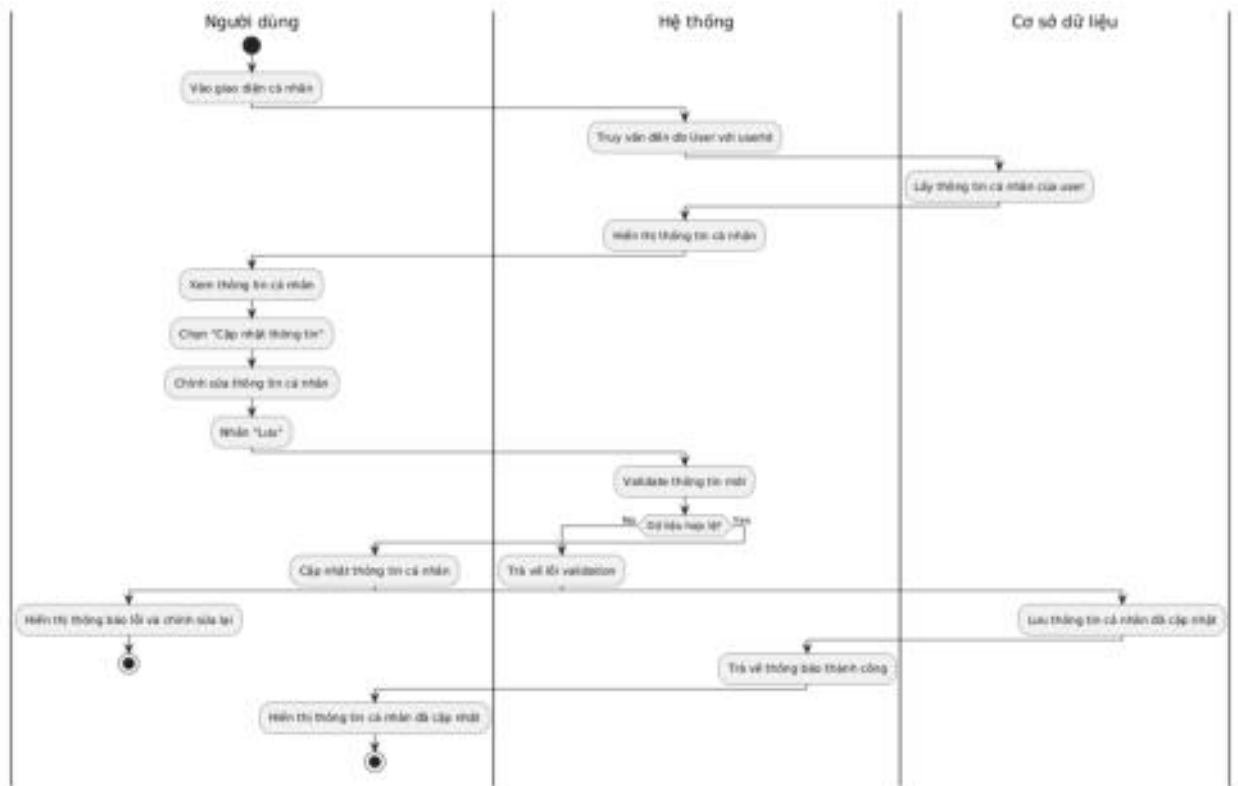
Hình 3.18 Sơ đồ hoạt động thêm địa chỉ

3.3.5 Chức năng cập nhật thông tin địa chỉ



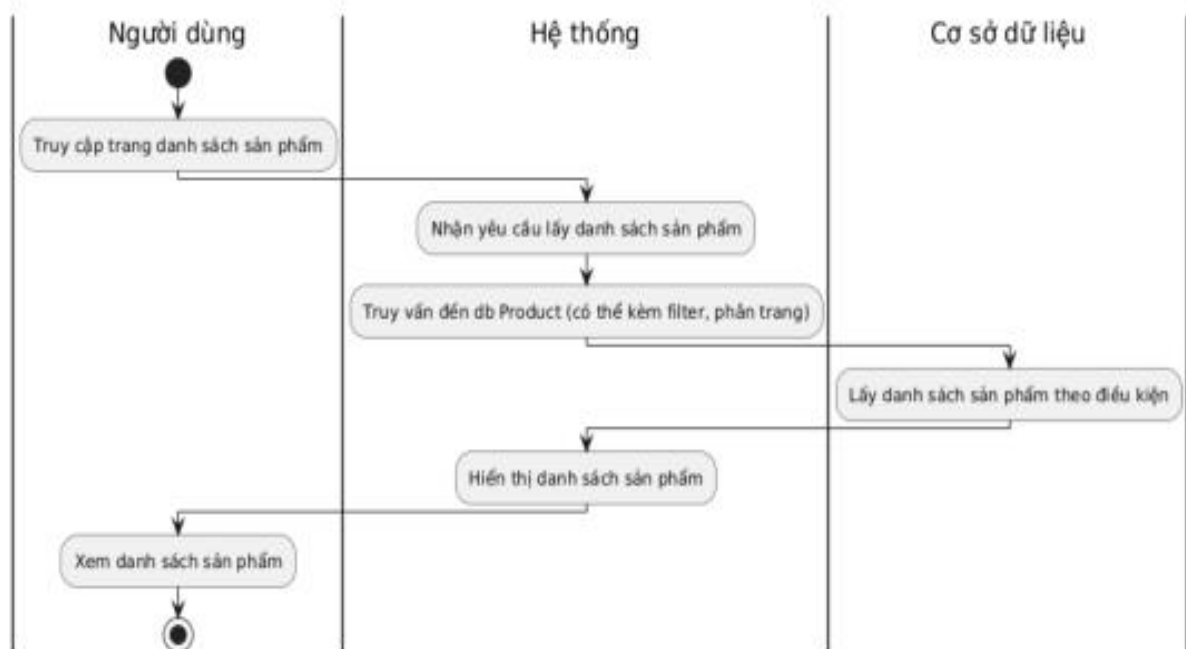
Hình 3.19 Sơ đồ hoạt động cập nhật địa chỉ

3.3.6 Chức năng quản lý thông tin cá nhân của người dùng



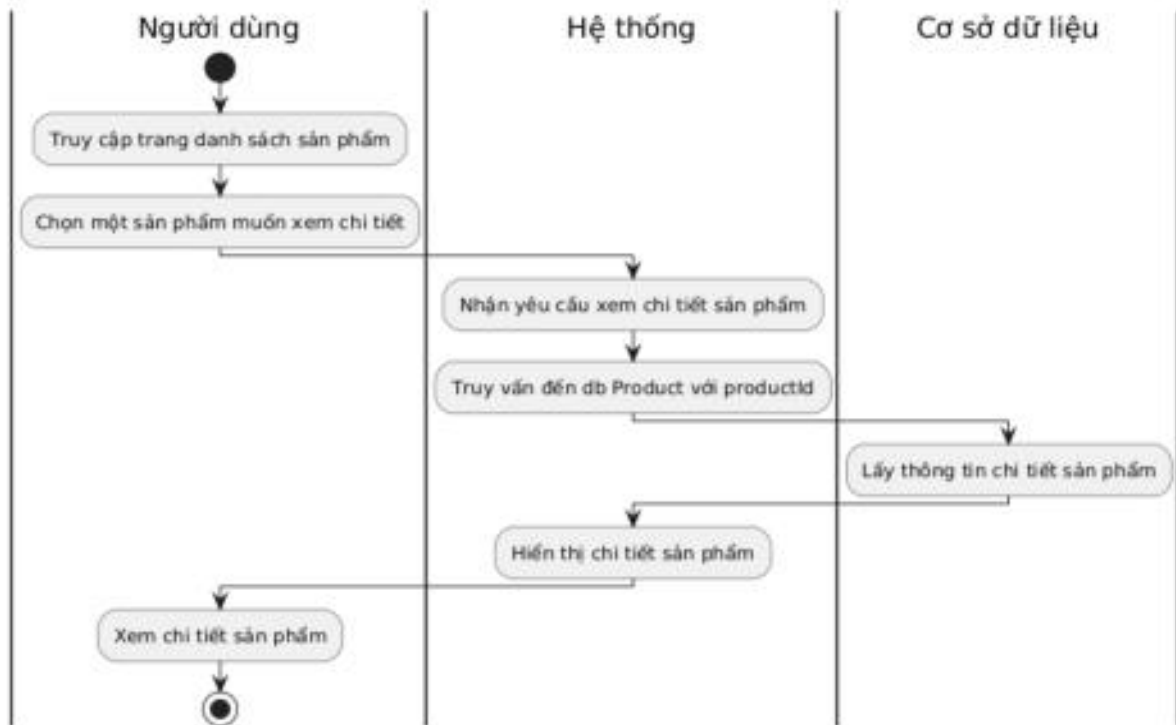
Hình 3.20 Sơ đồ hoạt động quản lý thông tin cá nhân

3.3.7 Chức năng xem danh sách sản phẩm



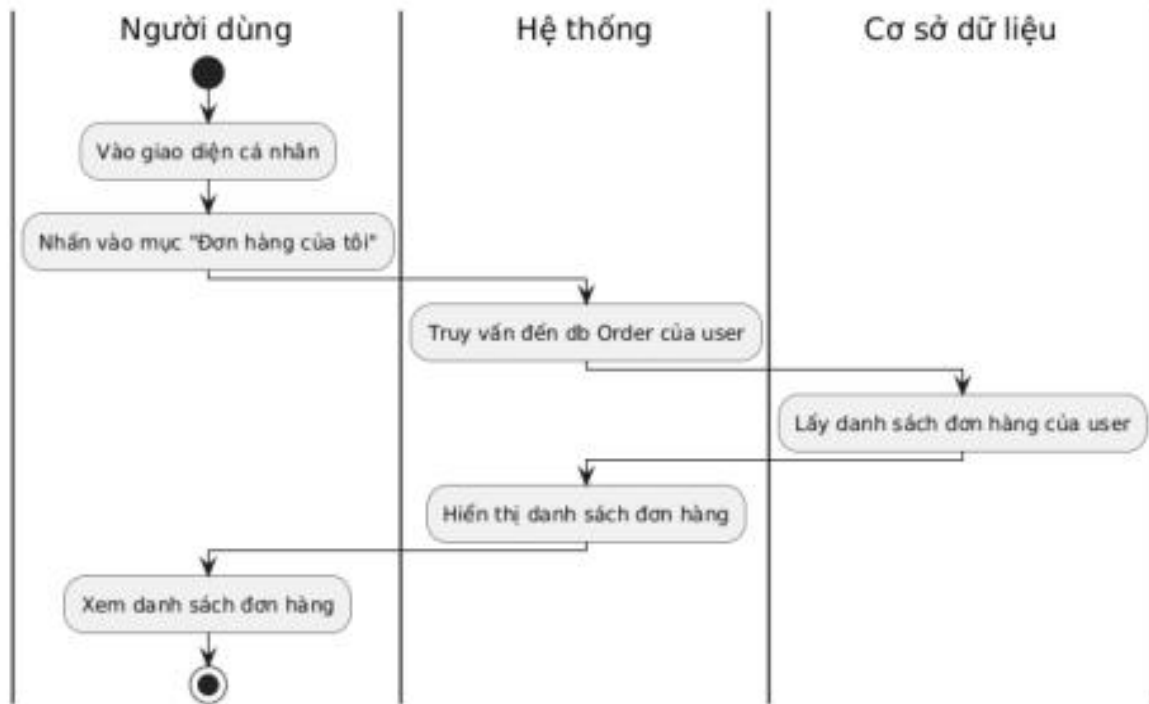
Hình 3.21 Sơ đồ hoạt động xem danh sách sản phẩm

3.3.8 Chức năng xem chi tiết sản phẩm



Hình 3.22 Sơ đồ hoạt động xem chi tiết sản phẩm

3.3.9 Chức năng xem danh sách đơn hàng



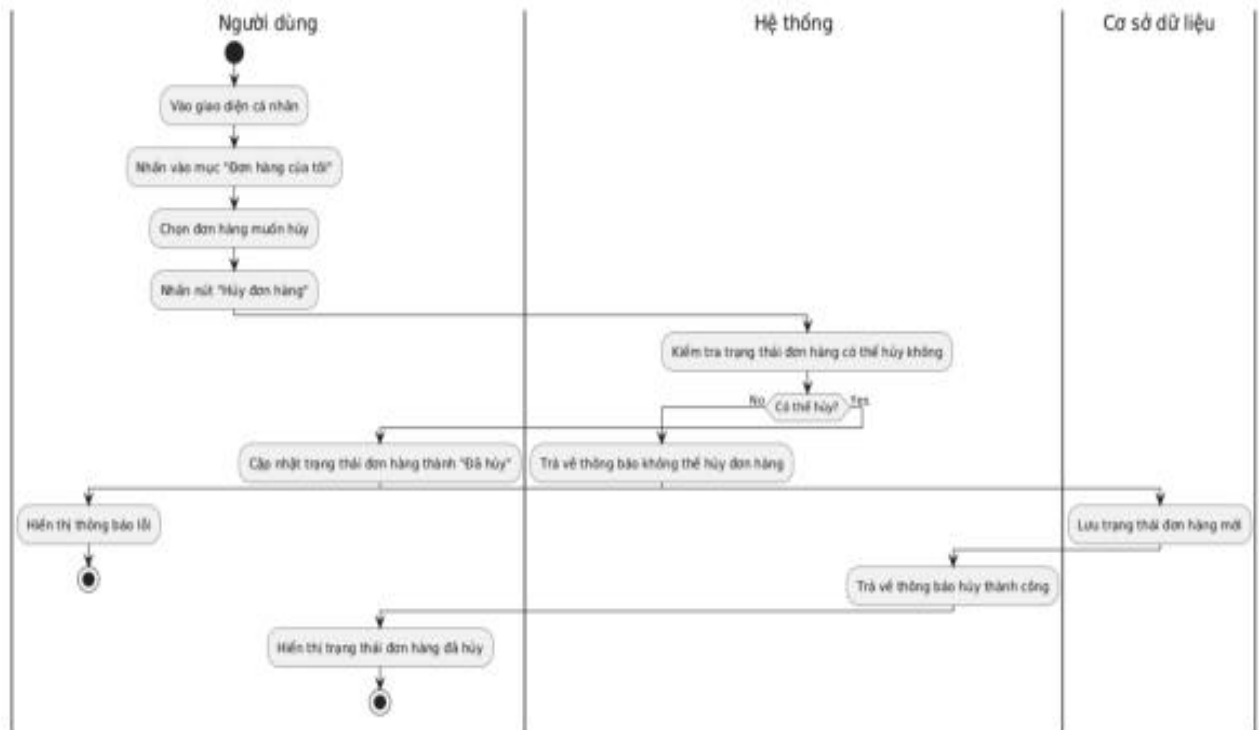
Hình 3.23 Sơ đồ hoạt động xem danh sách đơn hàng

3.3.10 Chức năng xem chi tiết đơn hàng



Hình 3.24 Sơ đồ hoạt động xem chi tiết đơn hàng

3.3.11 Chức năng huỷ đơn hàng



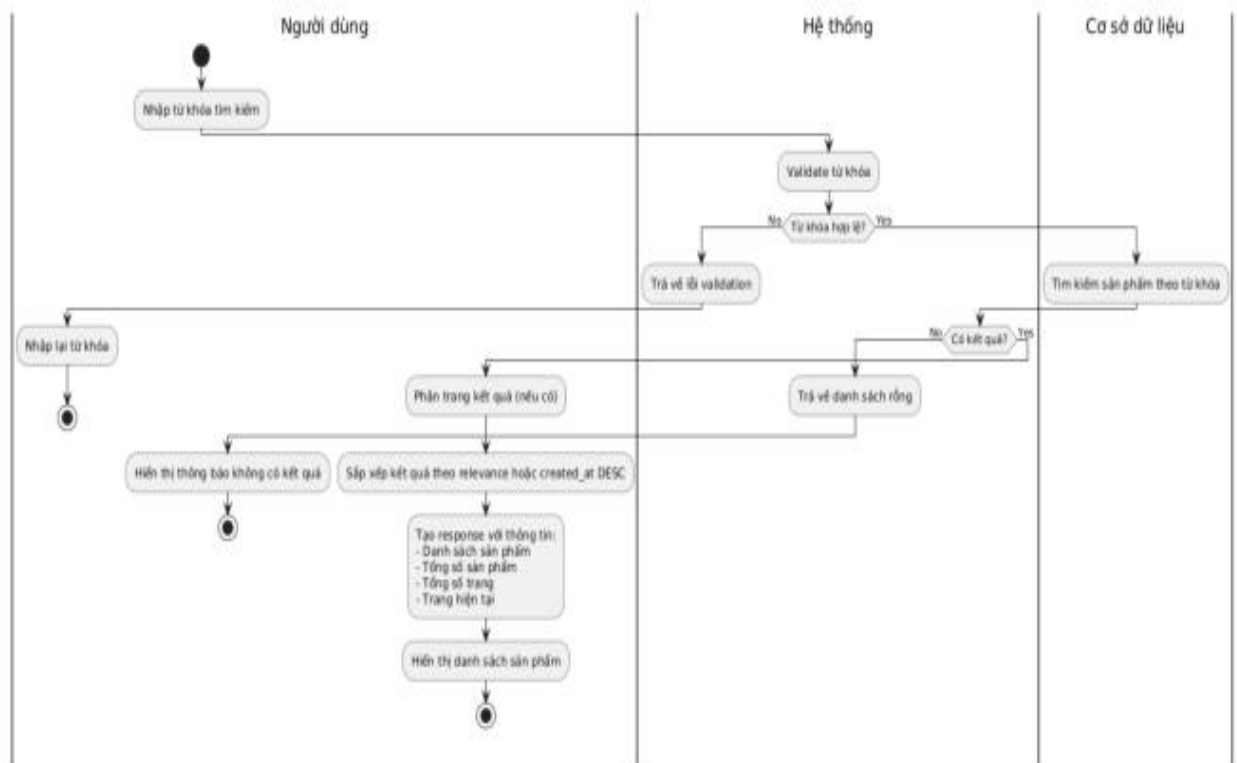
Hình 3.25 Sơ đồ hoạt động huỷ đơn hàng

3.3.12 Chức năng lọc các sản phẩm



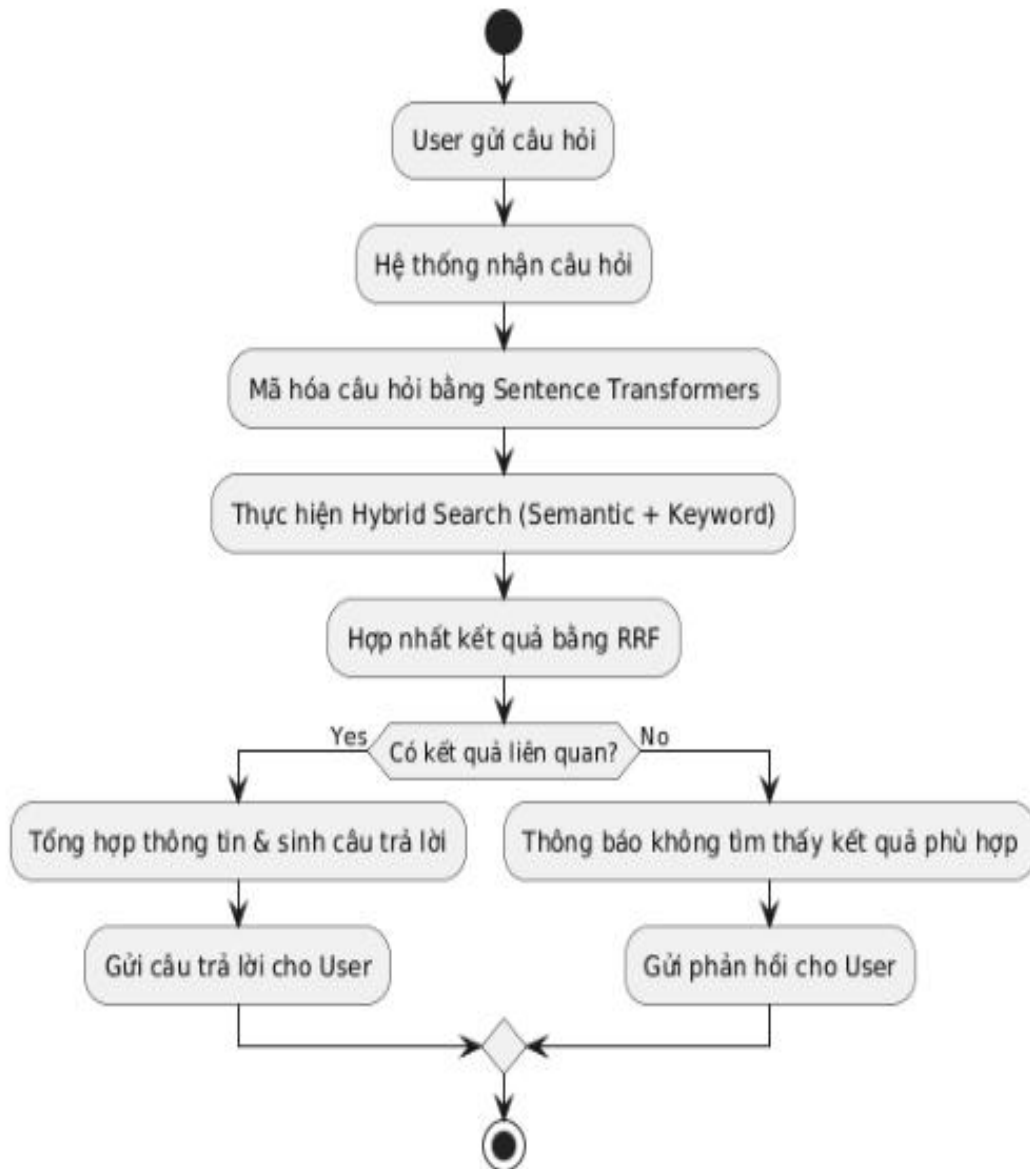
Hình 3.26 Sơ đồ hoạt động lọc sản phẩm

3.3.13 Chức năng tìm kiếm sản phẩm



Hình 3.27 Sơ đồ hoạt động tìm kiếm sản phẩm

3.3.14 Chức năng chatbot



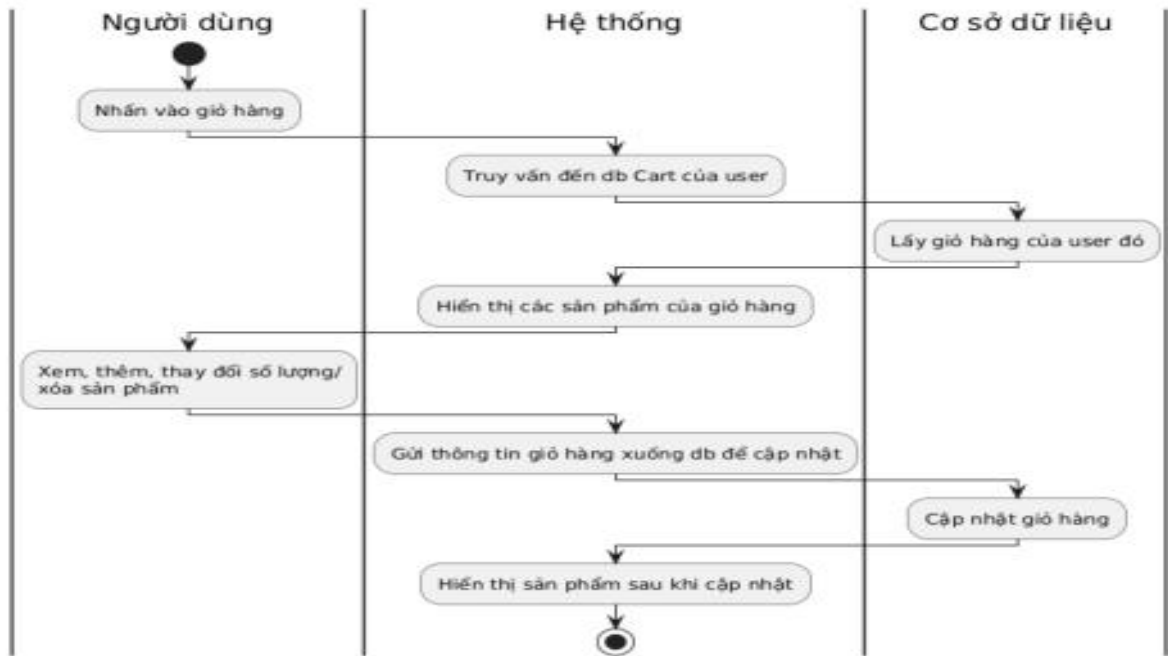
Hình 3.28 Sơ đồ hoạt động của chatbot

3.3.15 Chức năng gợi ý sản phẩm cho người dùng



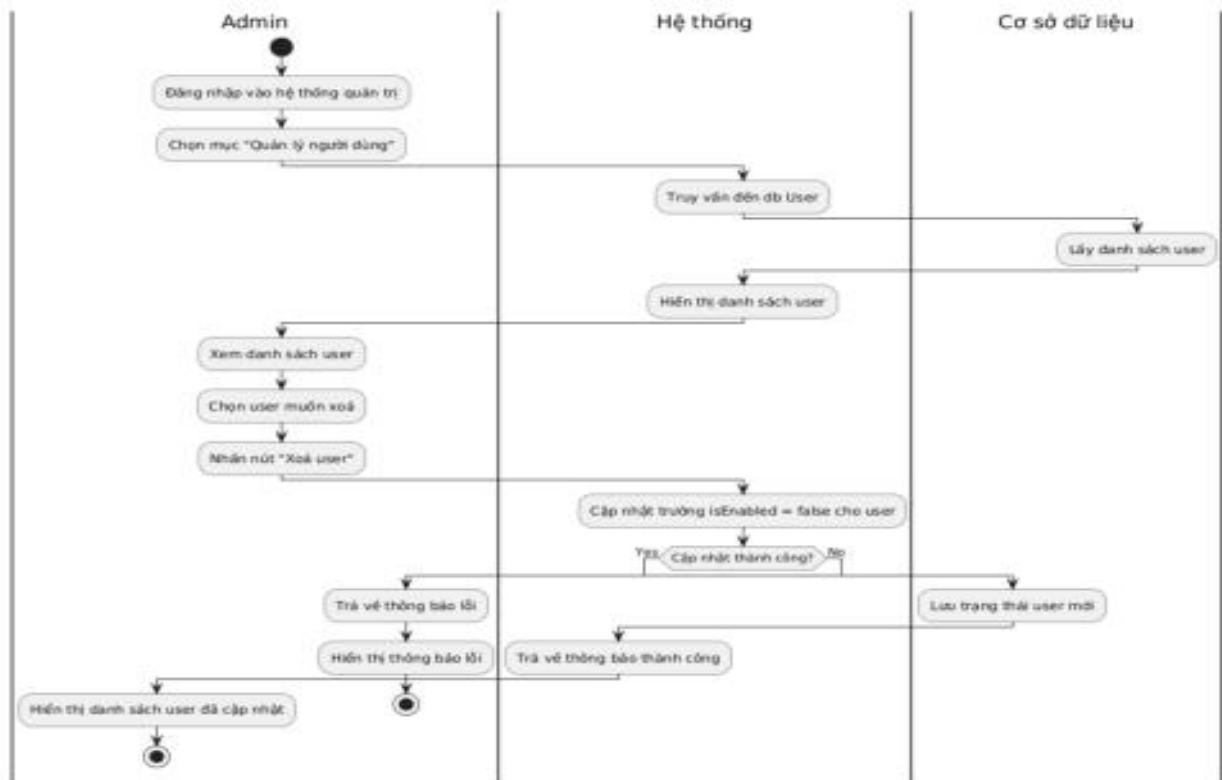
Hình 3.29 Sơ đồ hoạt động gợi ý sản phẩm

3.3.16 Chức năng quản lý giỏ hàng



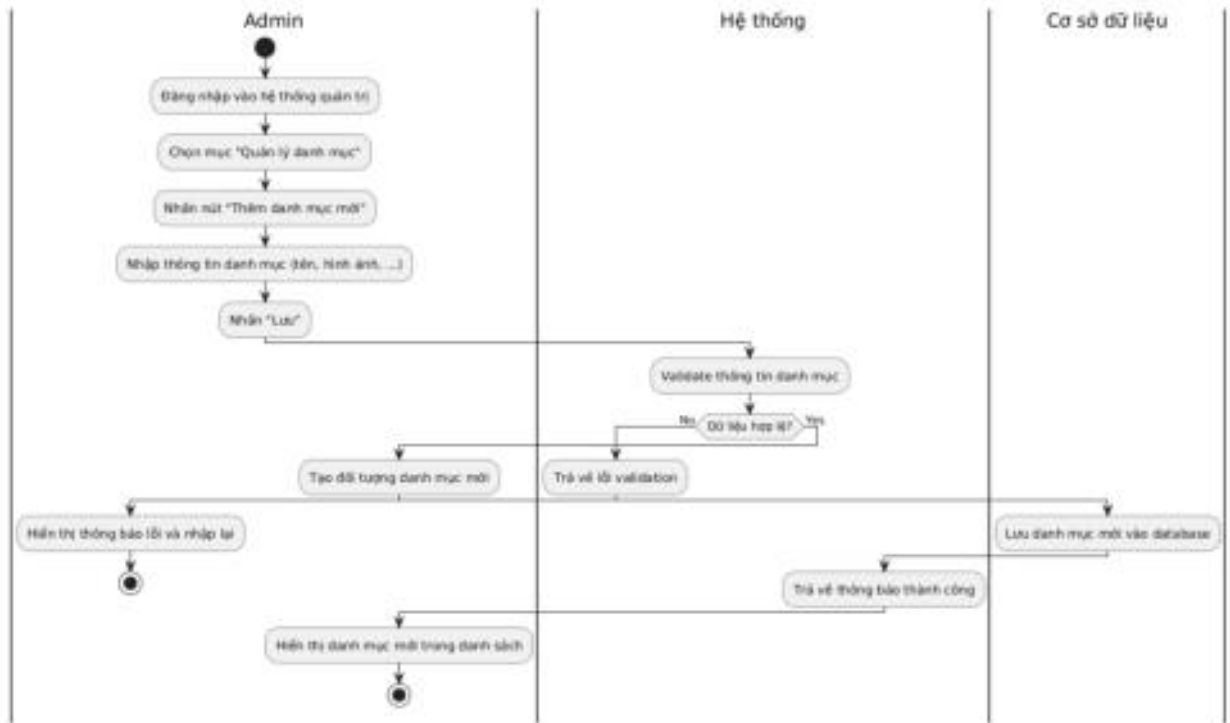
Hình 3.30 Sơ đồ hoạt động quản lý giỏ hàng

3.3.17 Chức năng admin quản lý user



Hình 3.31 Sơ đồ hoạt động quản lý user

3.3.18 Chức năng admin thêm danh mục mới



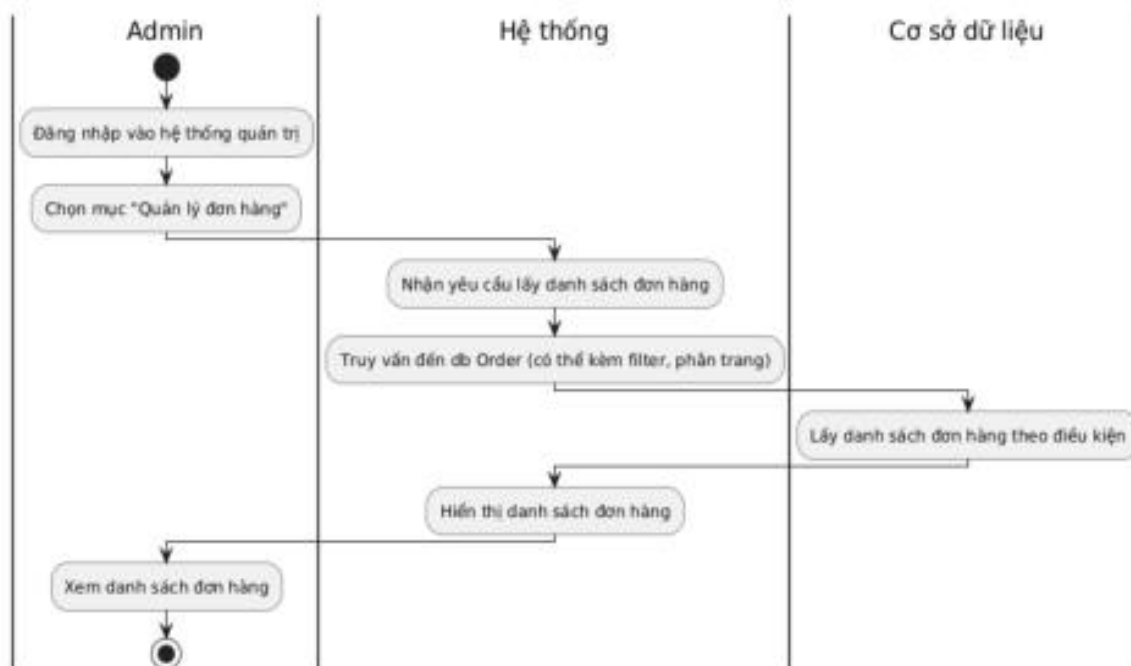
Hình 3.32 Sơ đồ hoạt động thêm danh mục

3.3.19 Chức năng sửa thông tin danh mục



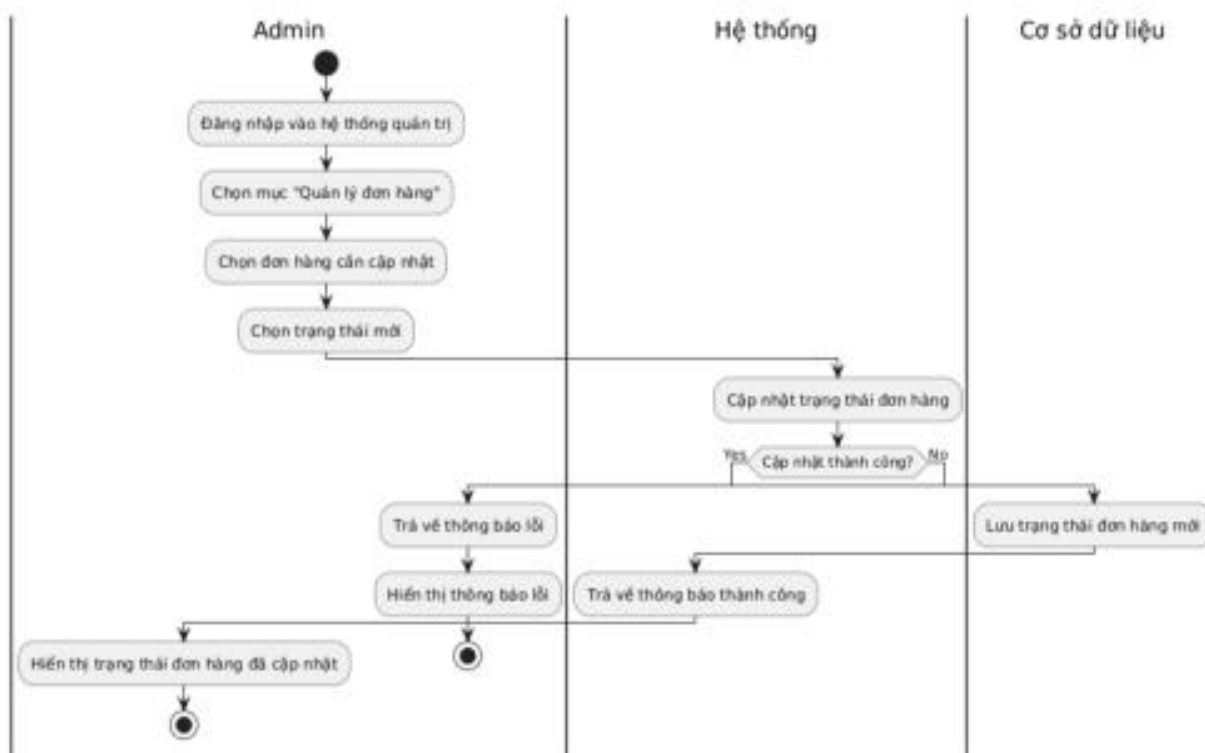
Hình 3.33 Sơ đồ hoạt động sửa thông tin danh mục

3.3.20 Chức năng admin xem danh sách đơn hàng



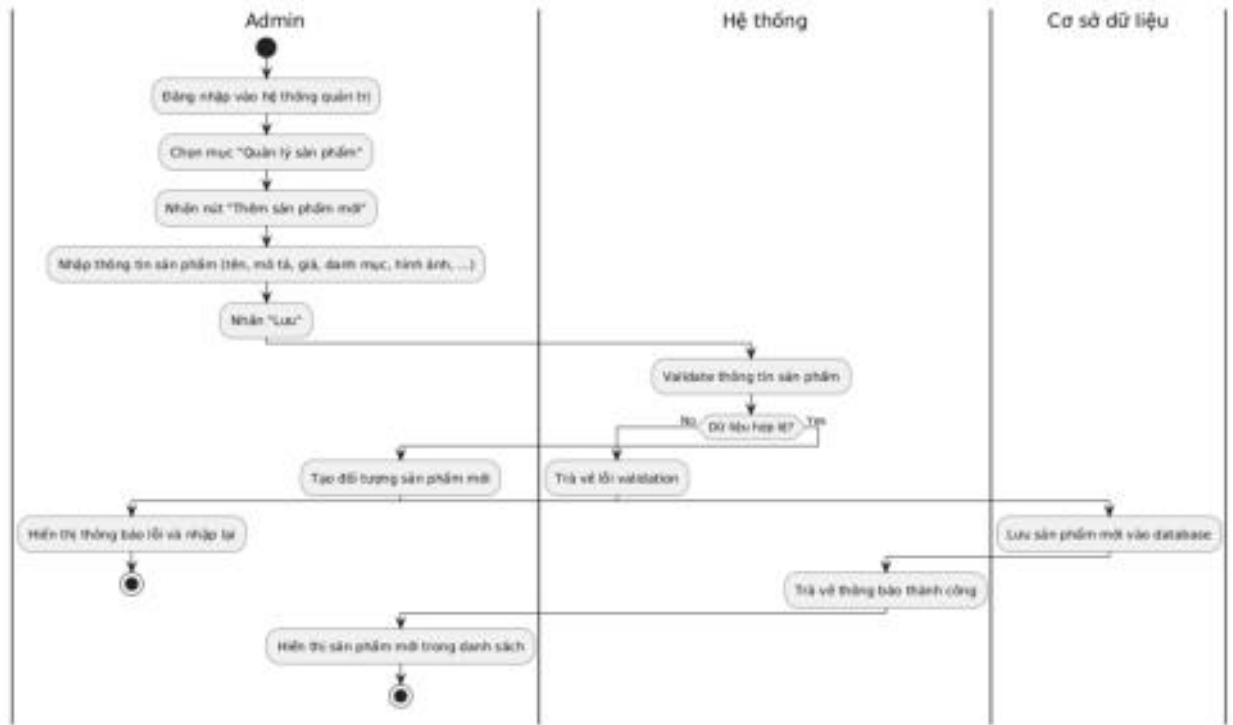
Hình 3.34 Sơ đồ hoạt động admin xem danh sách đơn hàng

3.3.21 Chức năng admin cập nhật trạng thái đơn hàng



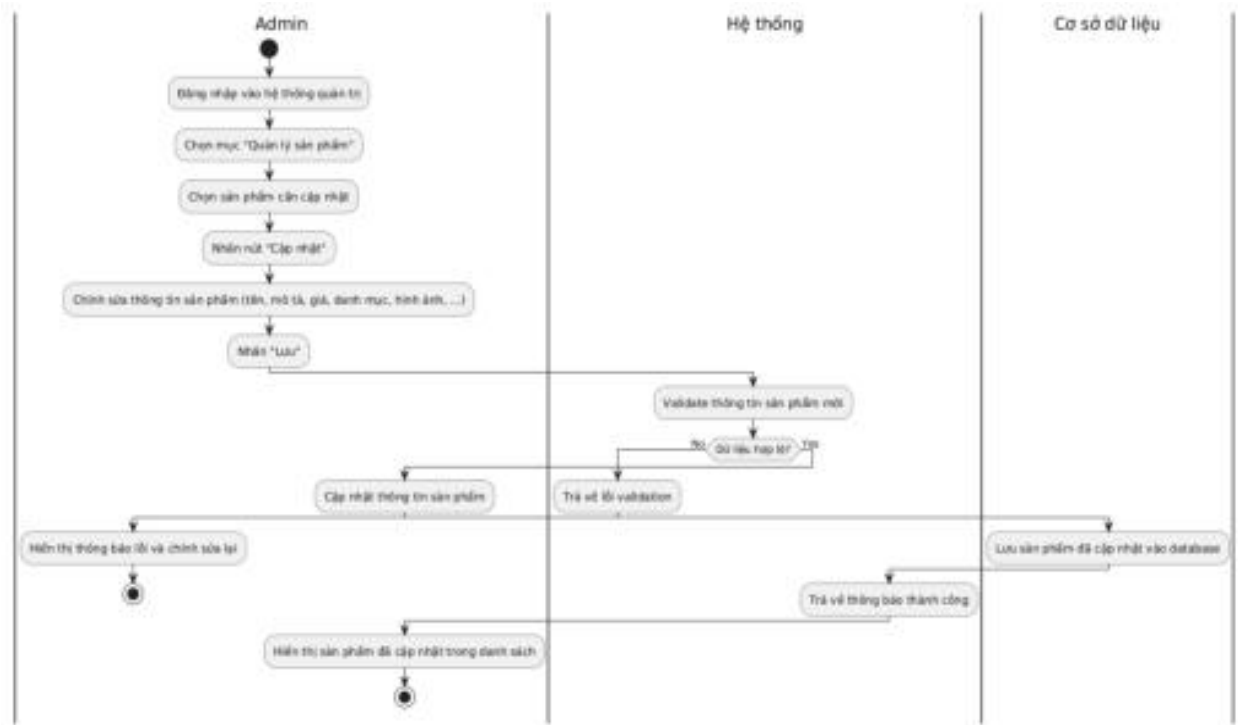
Hình 3.35 Sơ đồ hoạt động cập nhật trạng thái đơn hàng

3.3.22 Chức năng admin thêm sản phẩm mới



Hình 3.36 Sơ đồ hoạt động thêm sản phẩm

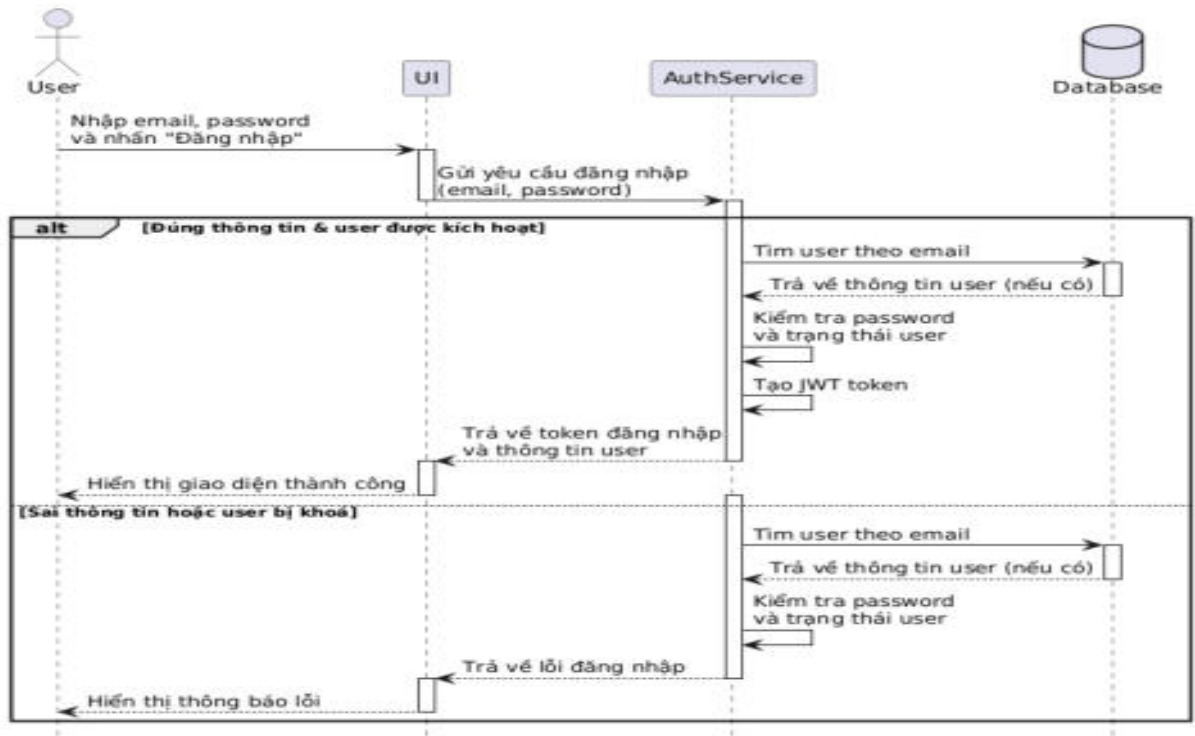
3.3.23 Chức năng cập nhật thông tin sản phẩm



Hình 3.37 Sơ đồ hoạt động cập nhật thông tin sản phẩm

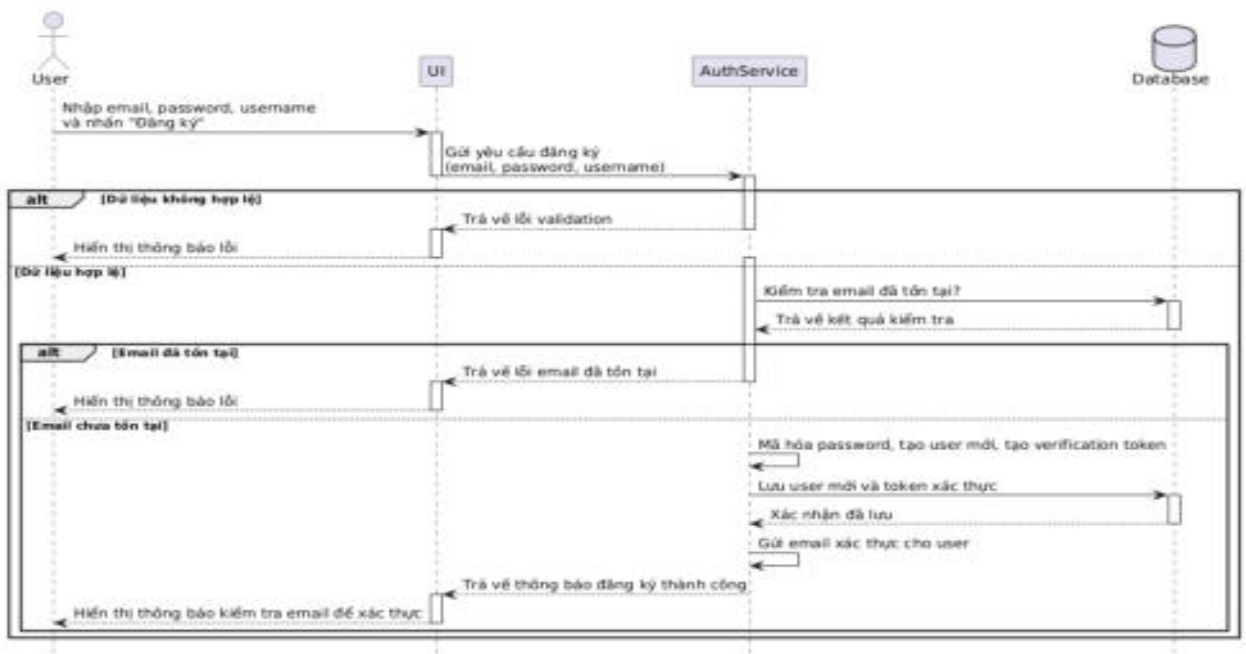
3.4 Biểu đồ tuần tự

3.4.1 Chức năng đăng nhập



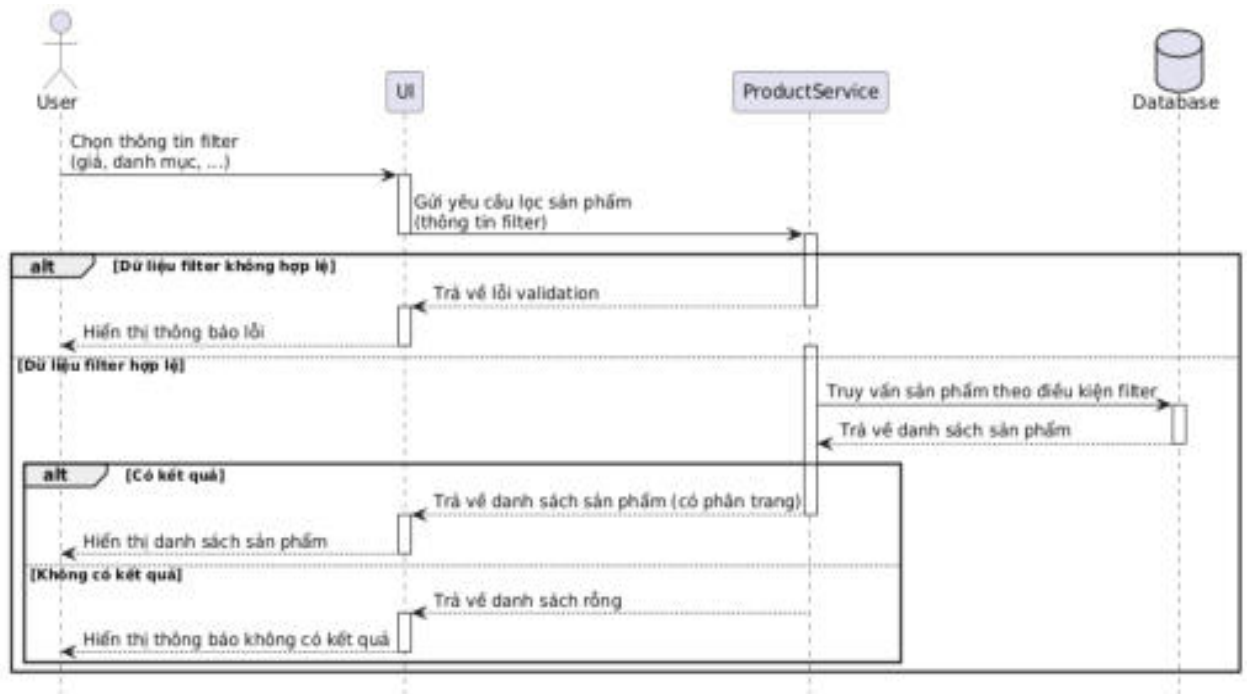
Hình 3.38 Sơ đồ tuần tự chức năng đăng nhập

3.4.2 Chức năng đăng kí



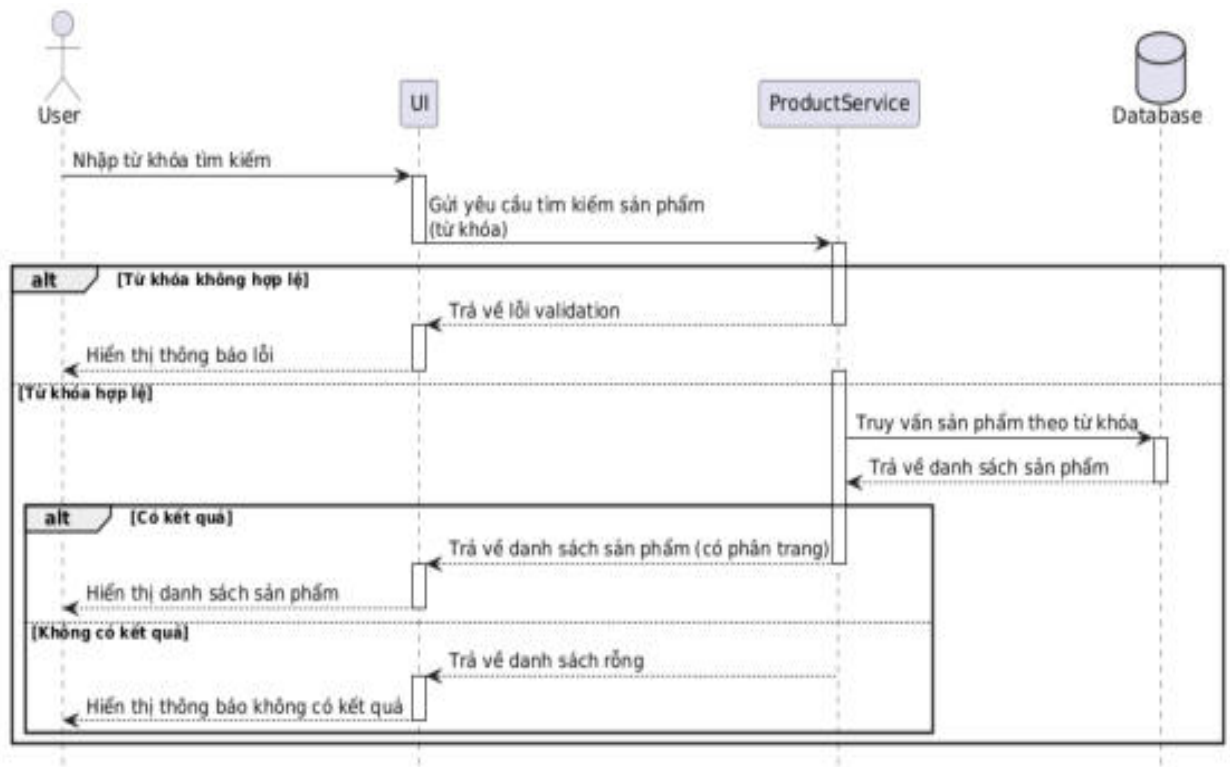
Hình 3.39 Sơ đồ tuần tự chức năng đăng kí

3.4.3 Chức năng lọc sản phẩm



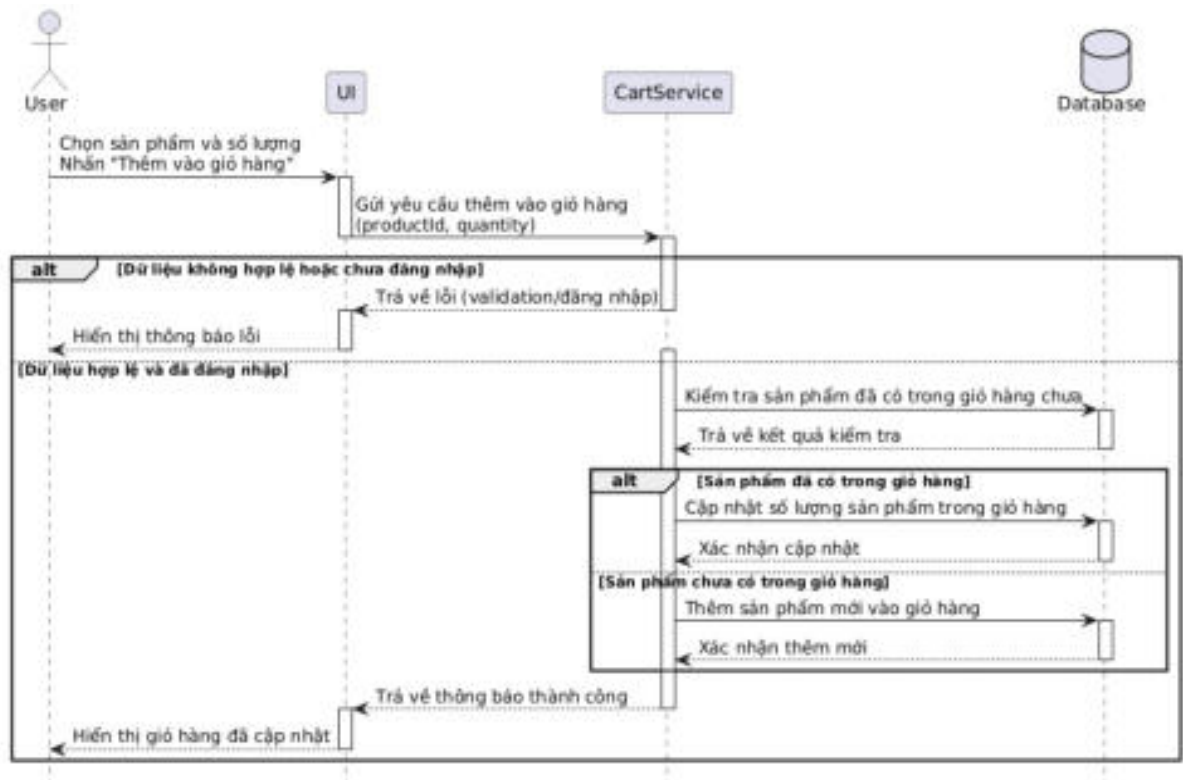
Hình 3.40 Sơ đồ tuần tự lọc sản phẩm

3.4.4 Chức năng tìm kiếm sản phẩm



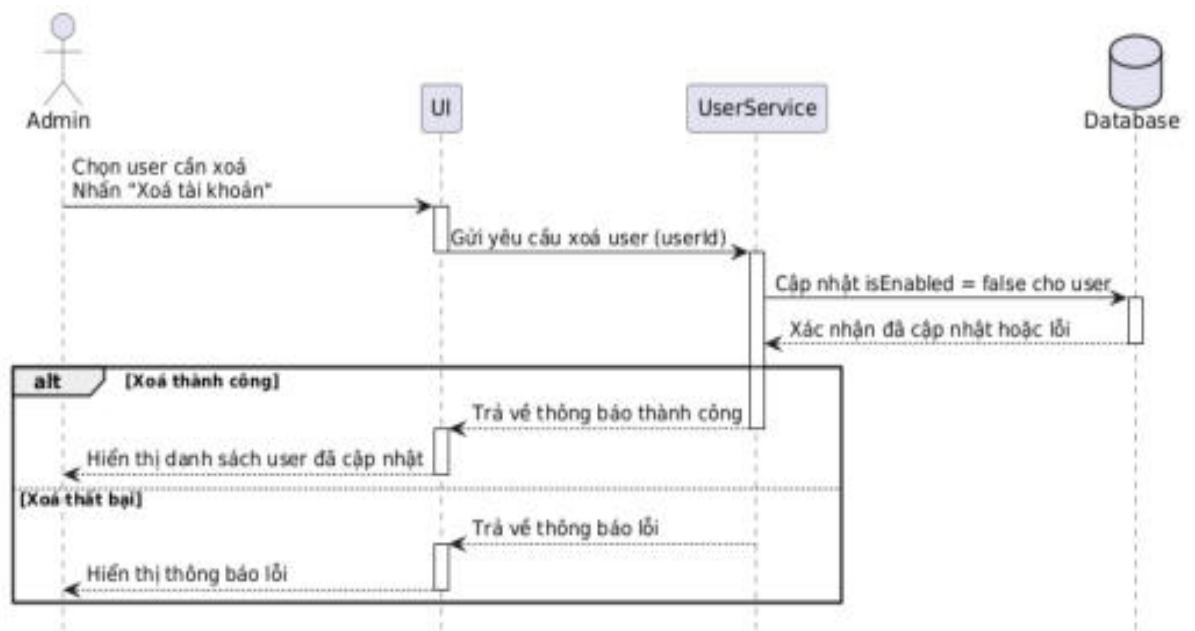
Hình 3.41 Sơ đồ tuần tự tìm kiếm sản phẩm

3.4.5 Chức năng user thêm sản phẩm vào giỏ hàng



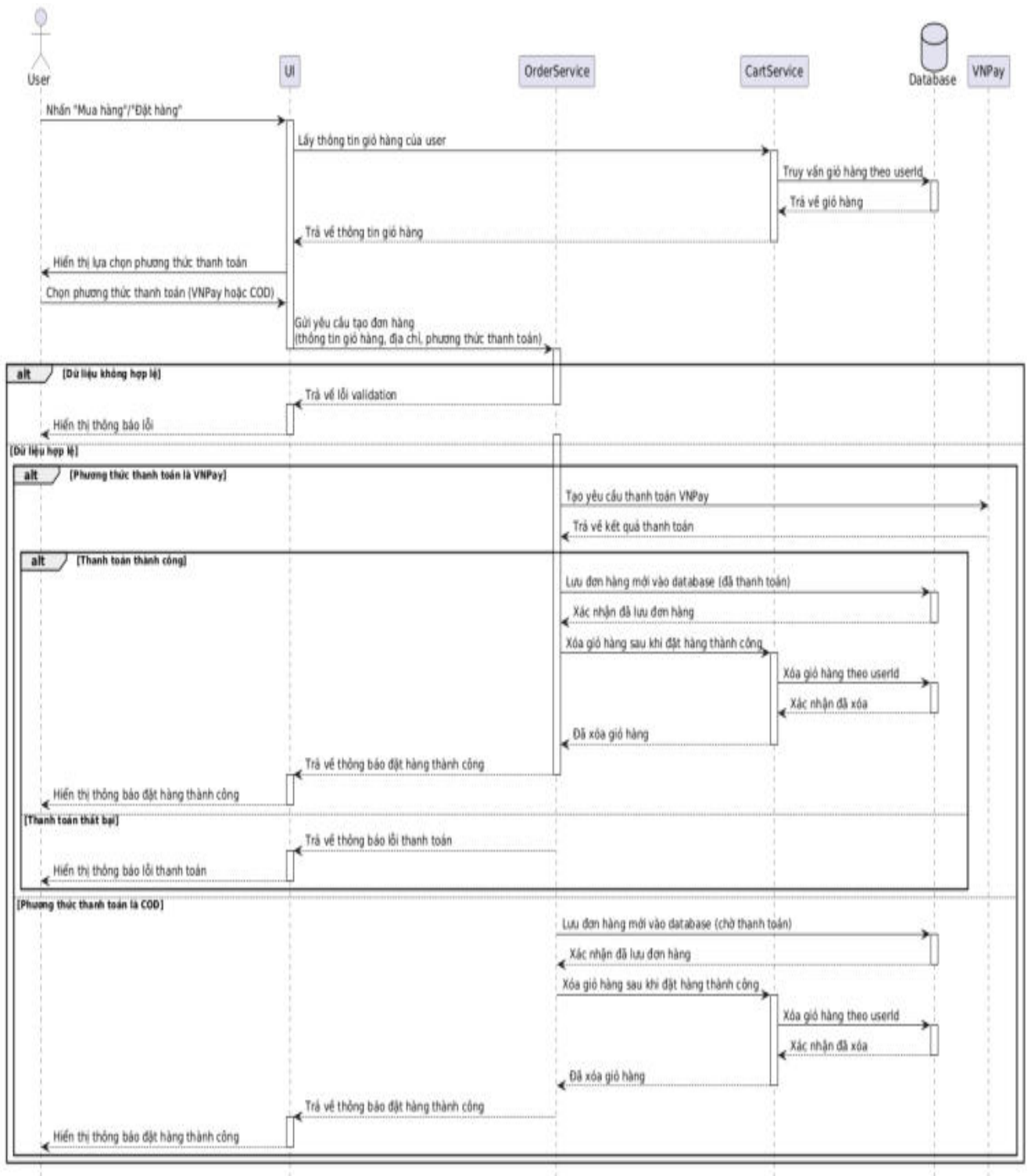
Hình 3.42 Sơ đồ tuần tự thêm sản phẩm vào giỏ hàng

3.4.6 Chức năng admin xoá tài khoản người dùng



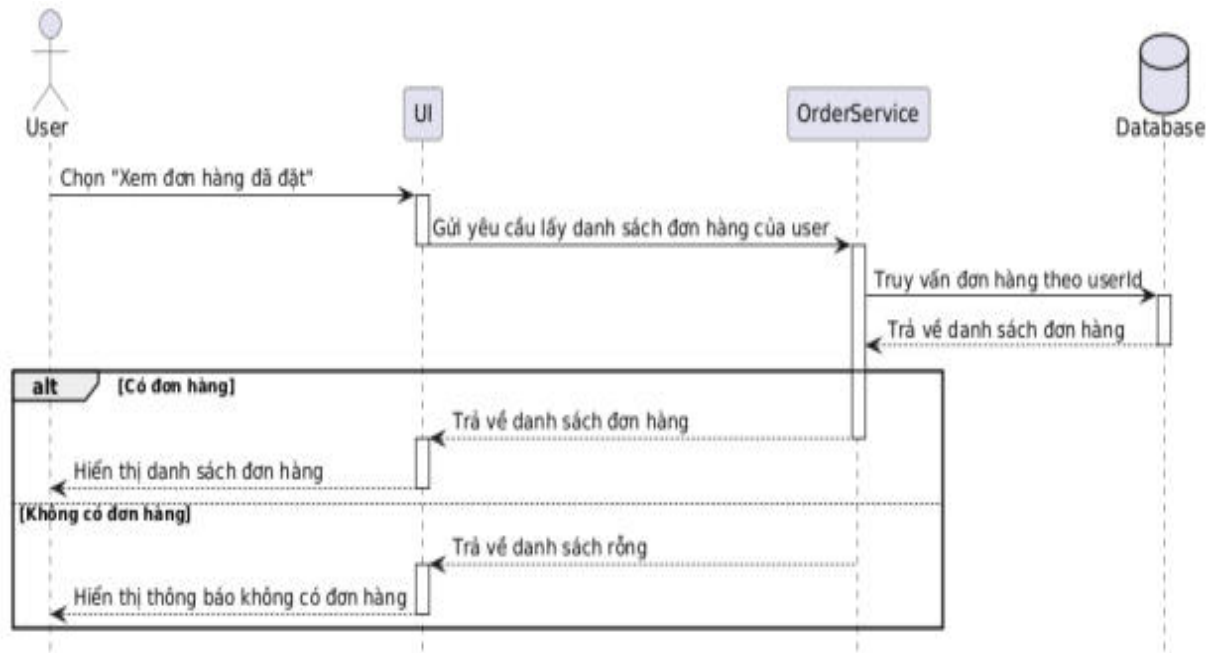
Hình 3.43 Sơ đồ tuần tự xoá tài khoản người dùng

3.4.7 Chức năng thanh toán



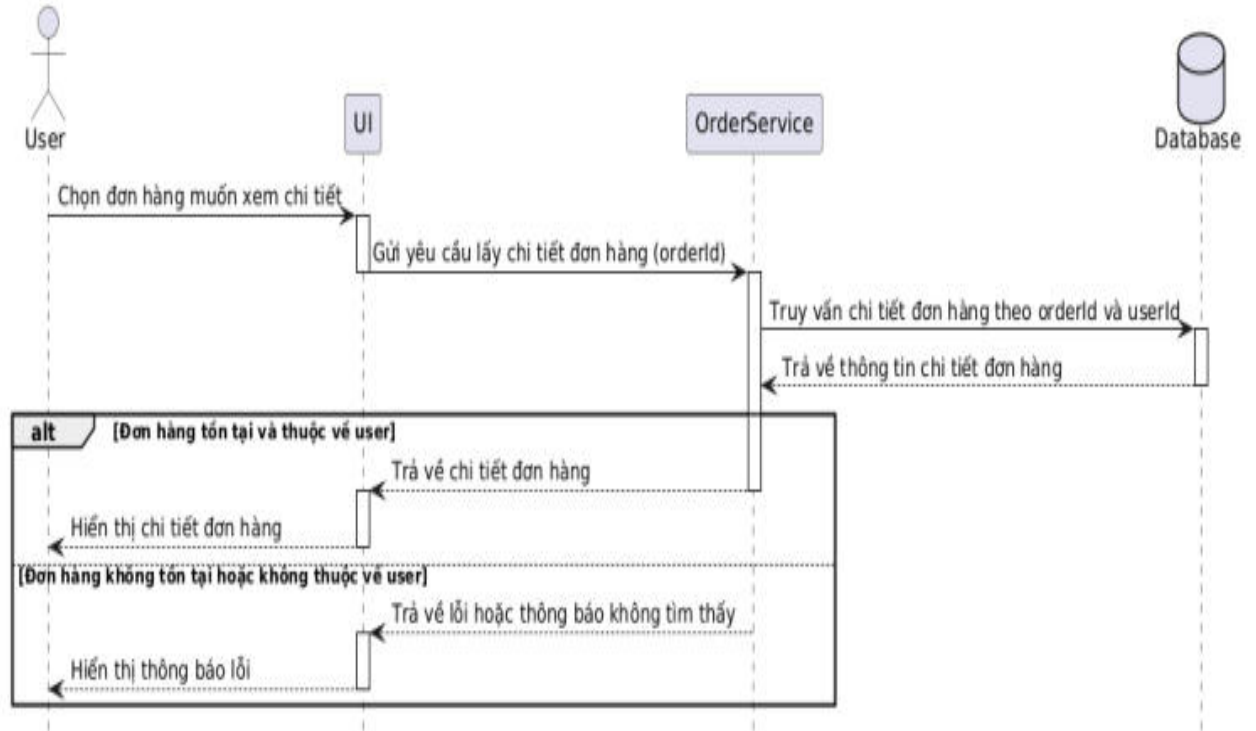
Hình 3.44 Sơ đồ tuần tự chức năng thanh toán

3.4.8 Chức năng user xem danh sách đơn hàng



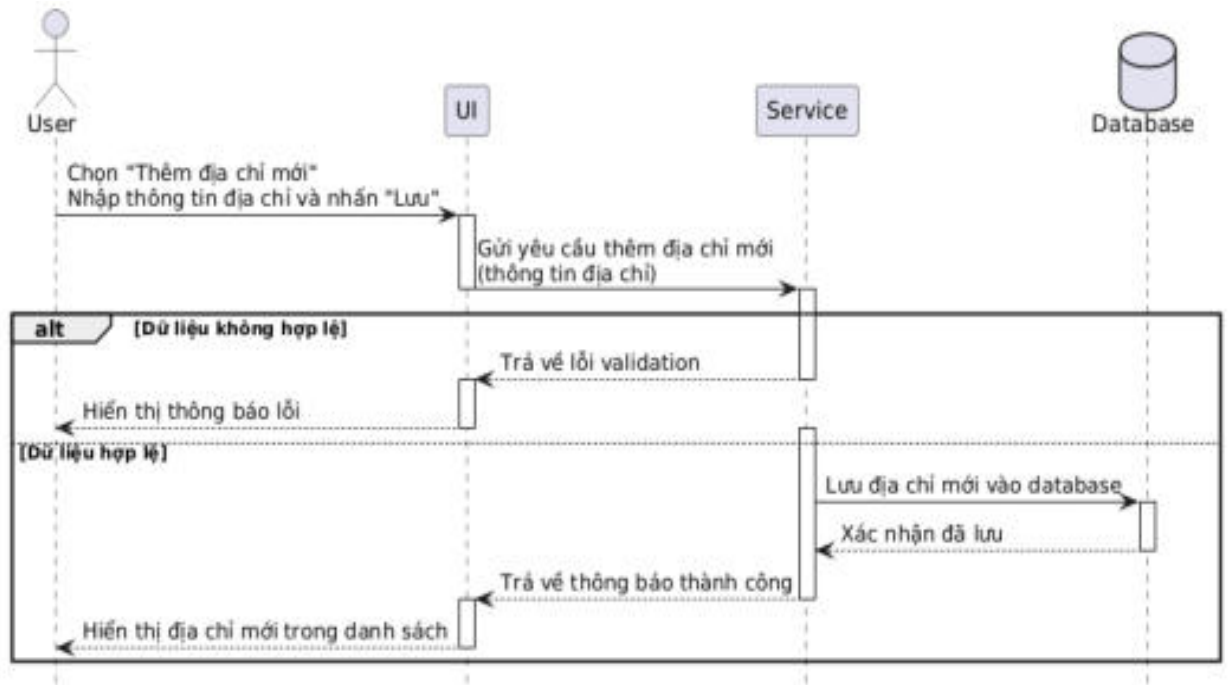
Hình 3.45 Sơ đồ tuần tự xem danh sách đơn hàng

3.4.9 Chức năng user xem chi tiết đơn hàng



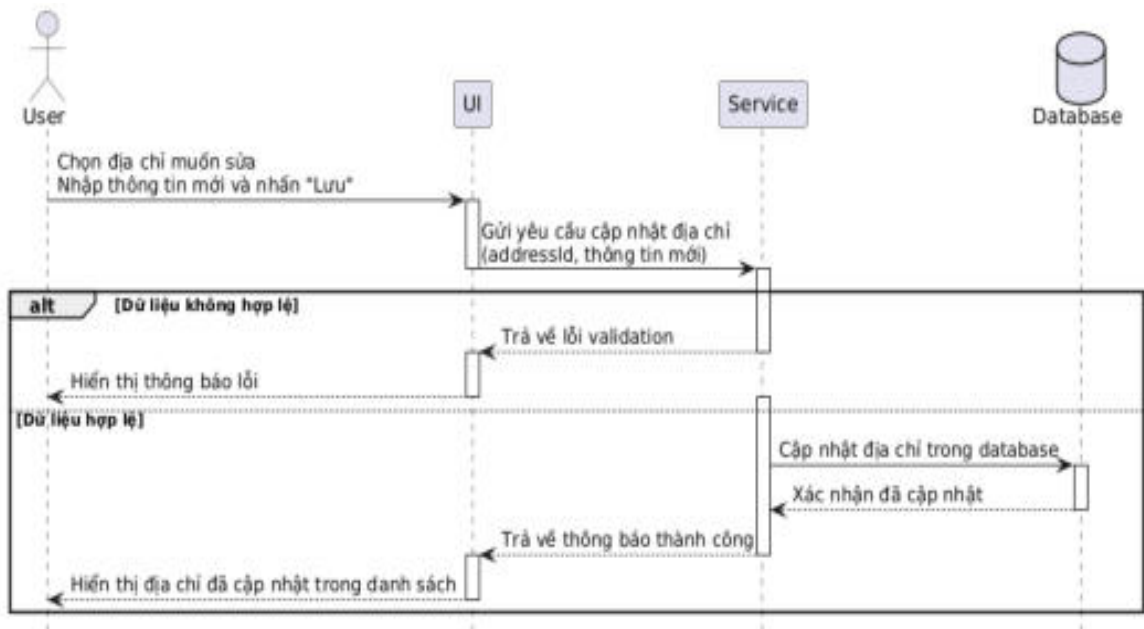
Hình 3.46 Sơ đồ tuần tự xem chi tiết đơn hàng

3.4.10 Chức năng user thêm địa chỉ



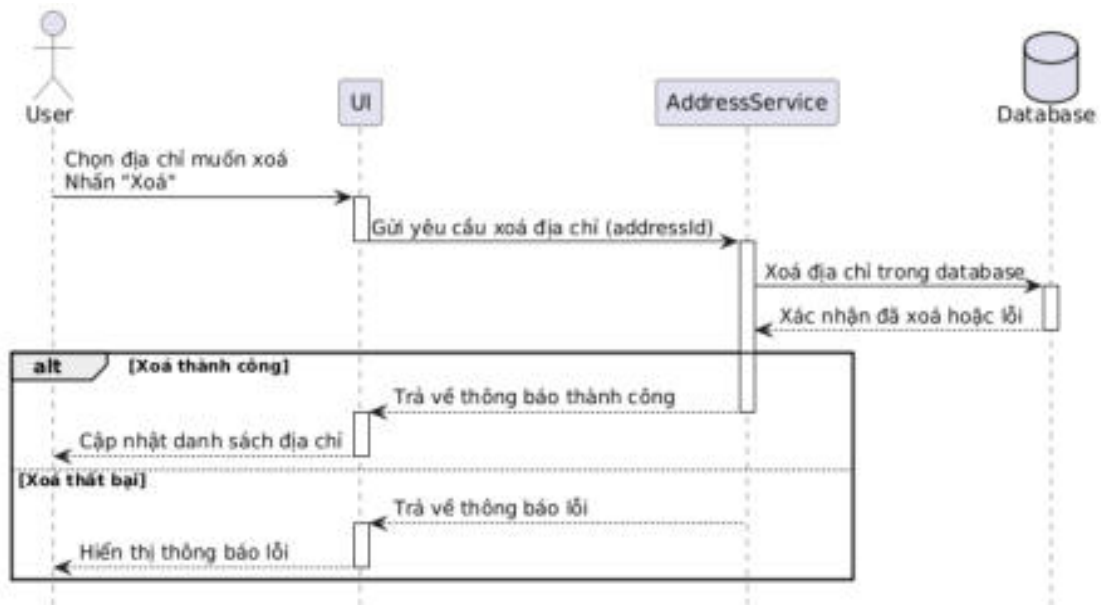
Hình 3.47 Sơ đồ tuần tự thêm địa chỉ

3.4.11 Chức năng user sửa thông tin địa chỉ



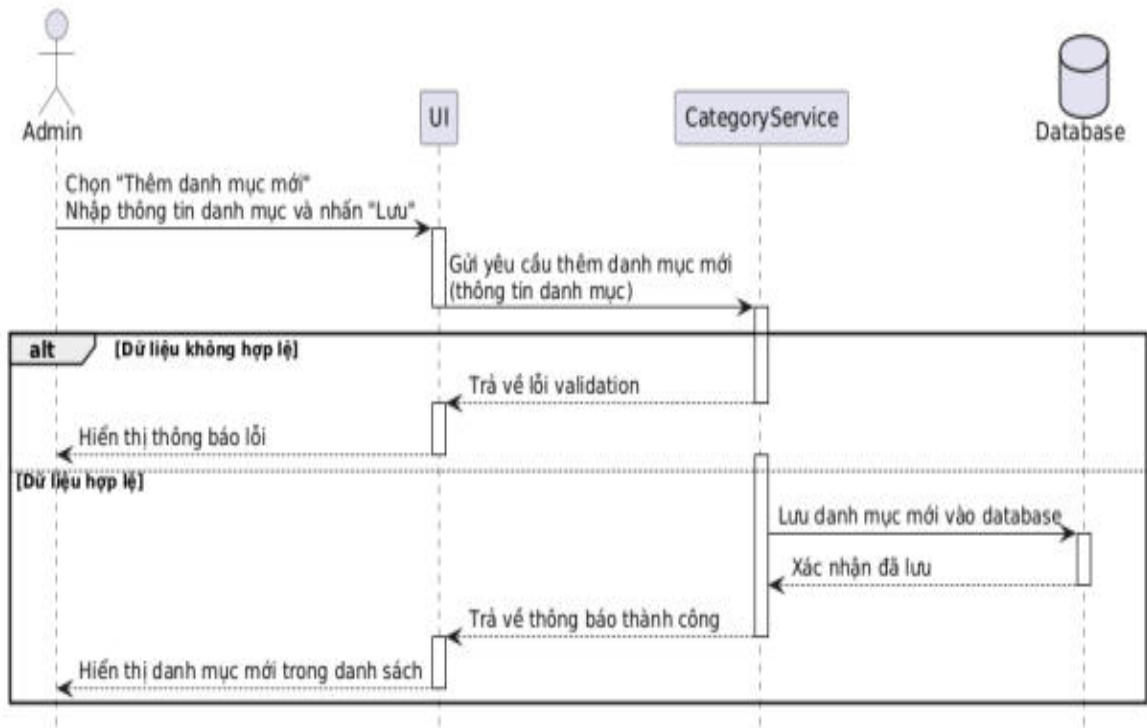
Hình 3.48 Sơ đồ tuần tự sửa địa chỉ

3.4.12 Chức năng xoá địa chỉ



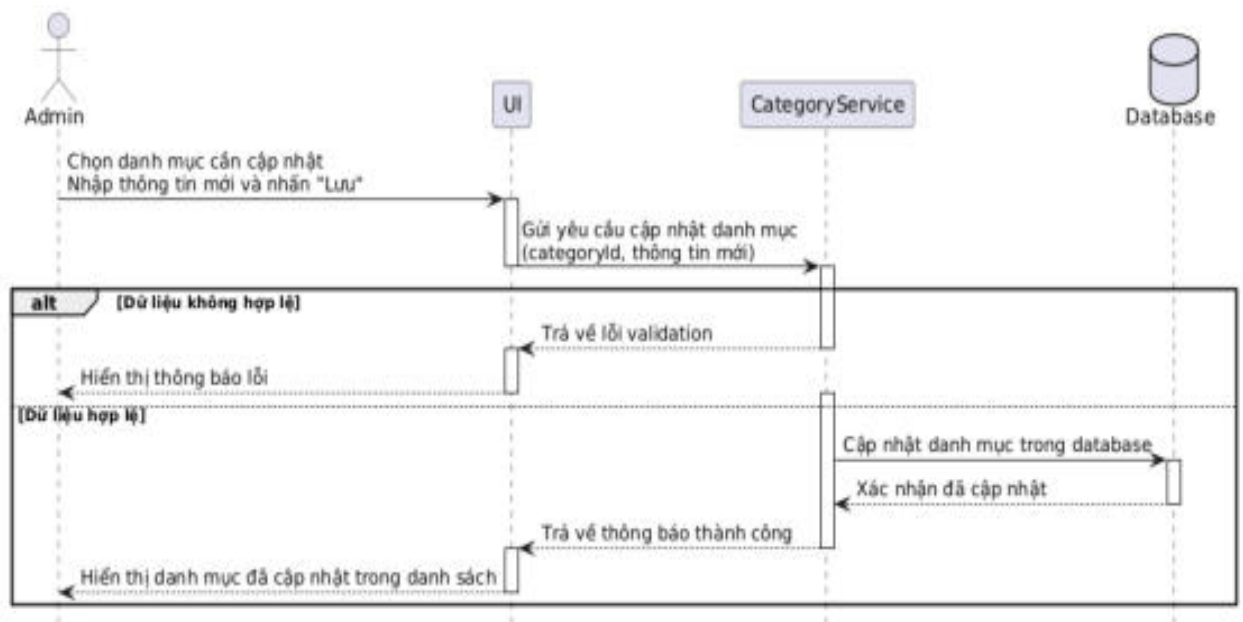
Hình 3.49 Sơ đồ tuần tự xoá địa chỉ

3.4.13 Chức năng admin thêm danh mục mới



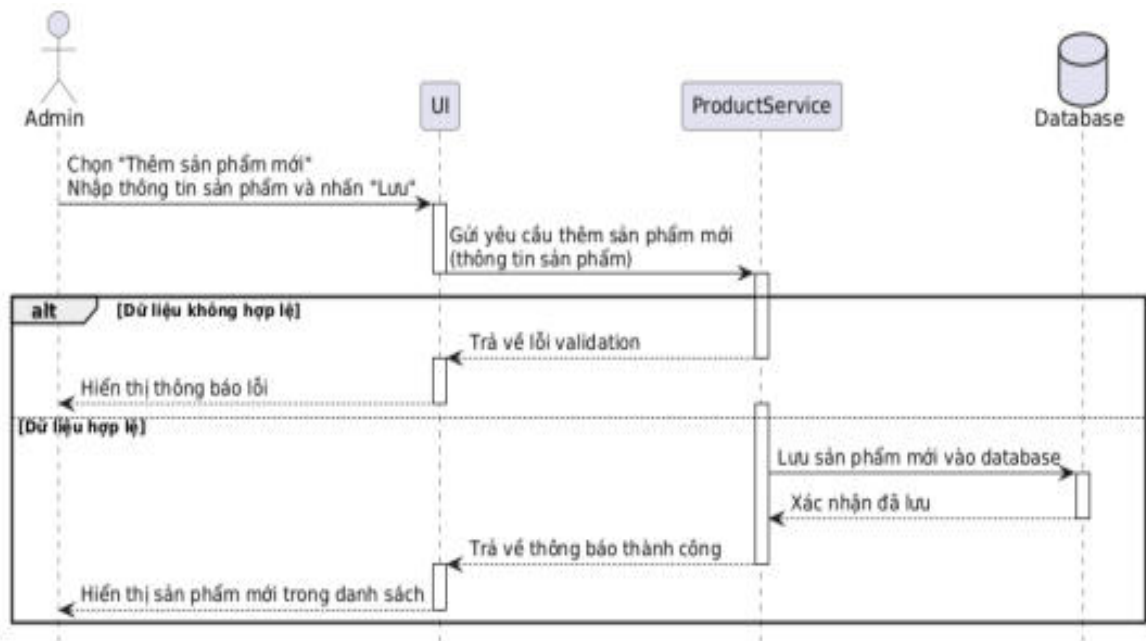
Hình 3.50 Sơ đồ tuần tự thêm danh mục mới

3.4.14 Chức năng admin cập nhật thông tin danh mục



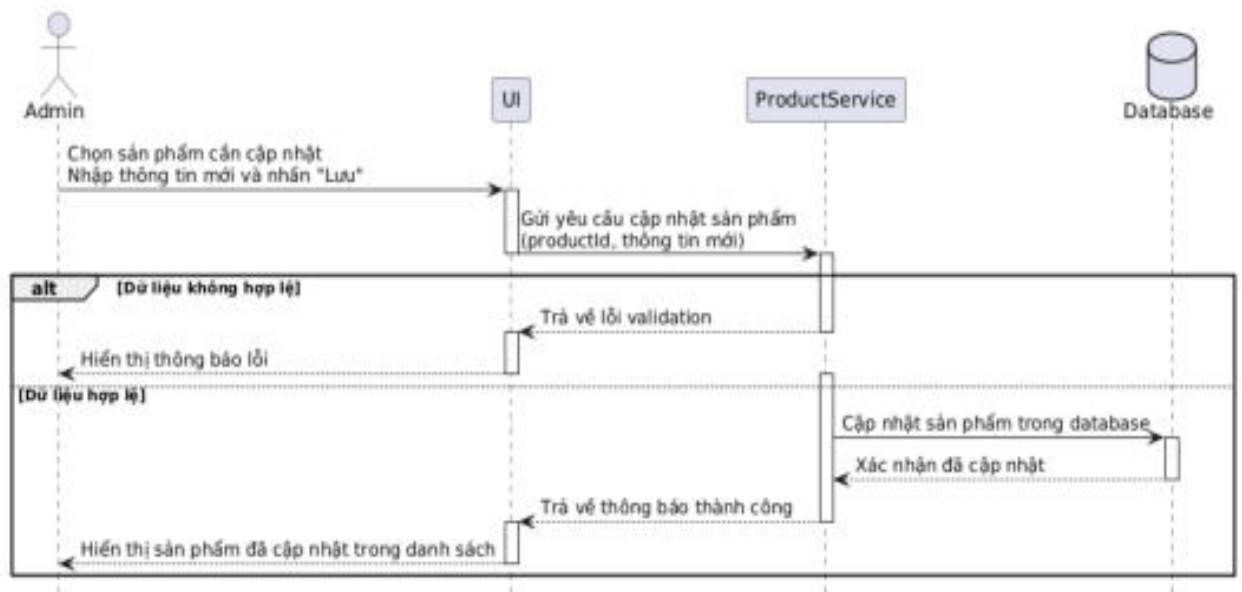
Hình 3.51 Sơ đồ tuần tự cập nhật danh mục

3.4.15 Chức năng admin thêm mới sản phẩm



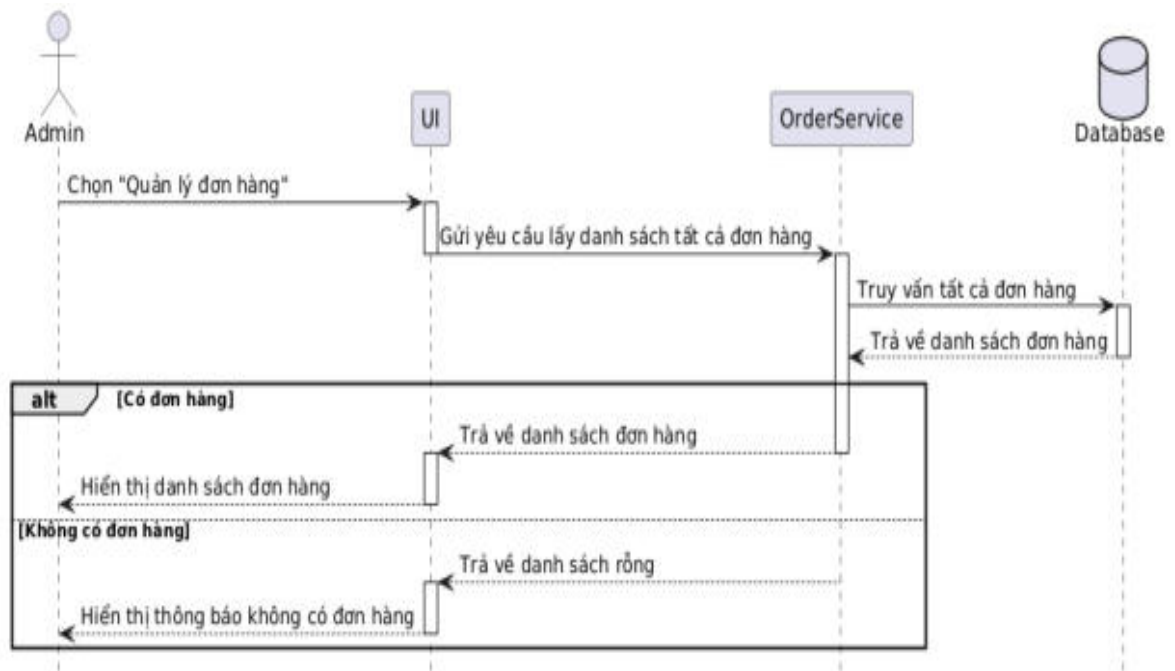
Hình 3.52 Sơ đồ tuần tự thêm mới sản phẩm

3.4.16 Chức năng cập nhật thông tin sản phẩm



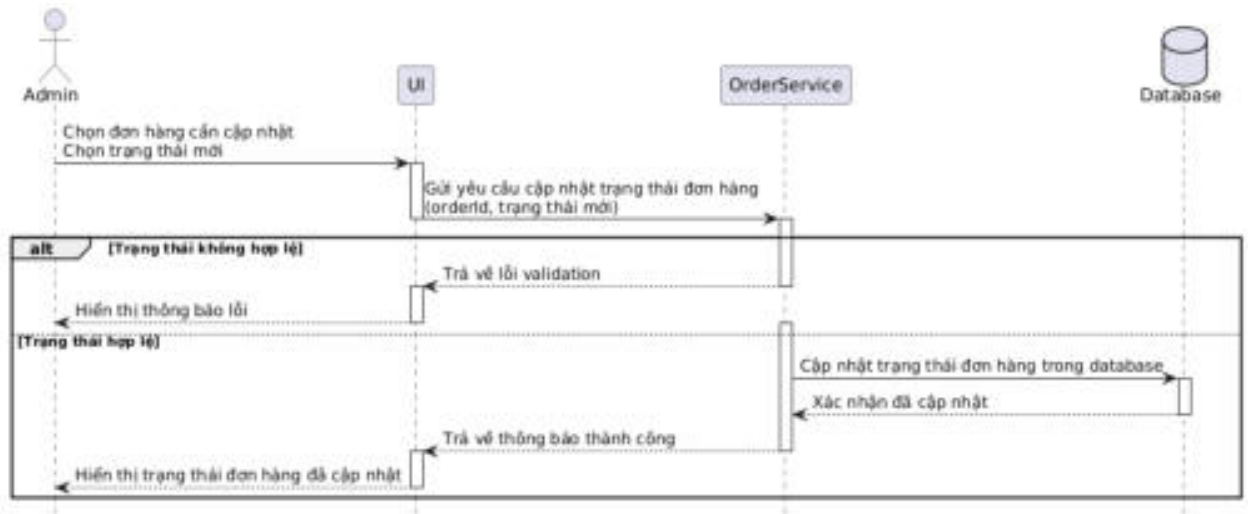
Hình 3.53 Sơ đồ tuần tự cập nhật thông tin sản phẩm

3.4.17 Chức năng admin hiển thị toàn bộ đơn hàng



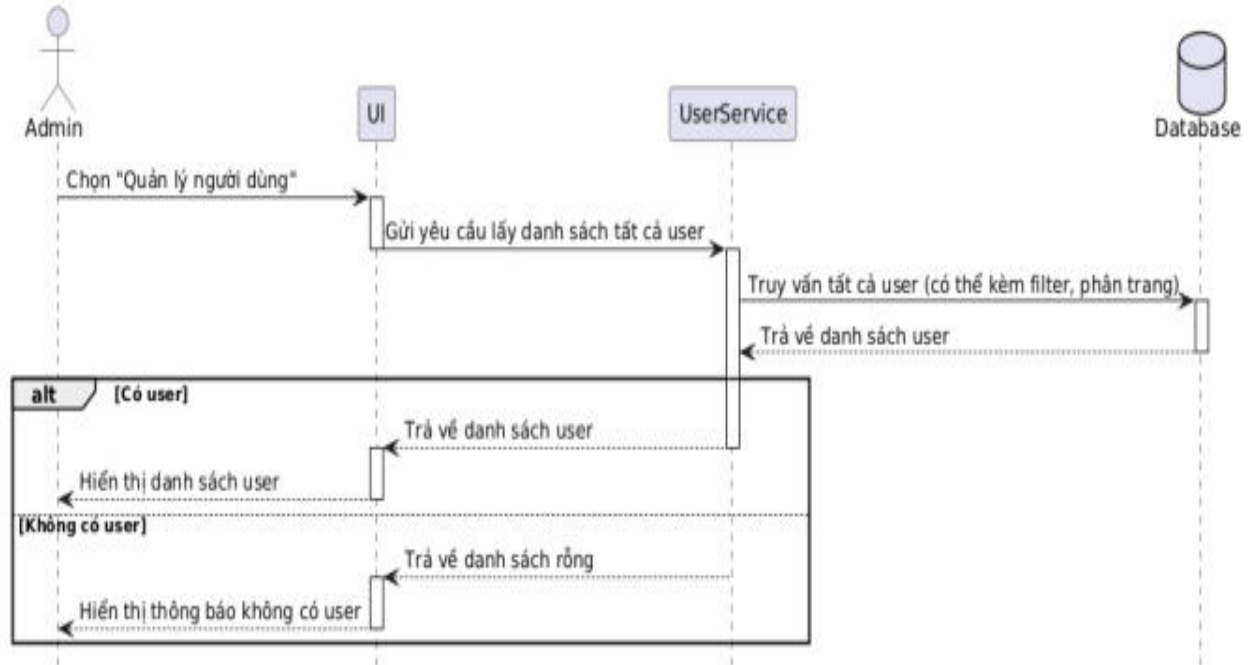
Hình 3.54 Sơ đồ tuần tự hiển thị danh sách đơn hàng

3.4.18 Chức năng admin cập nhật trạng thái đơn hàng



Hình 3.55 Sơ đồ tuần tự cập nhật trạng thái đơn hàng

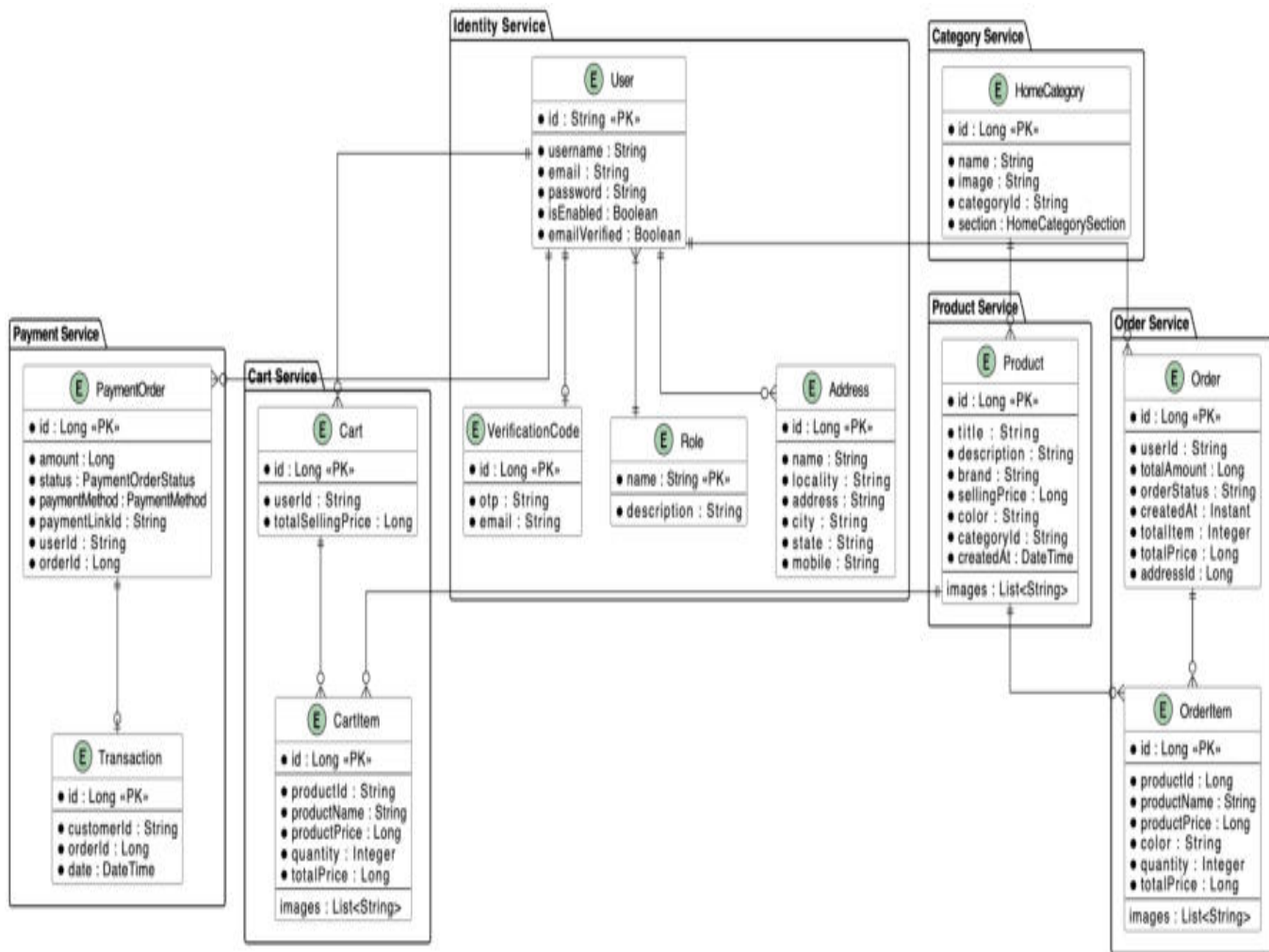
3.4.19 Chức năng admin hiển thị danh sách người dùng



Hình 3.56 Sơ đồ tuần tự hiển thị danh sách user

3.5 Lược đồ cơ sở dữ liệu

3.5.1 Sơ đồ Entity Relationship Diagram – ERD



Hình 3.57 Sơ đồ ERD

- Phân quyền hệ thống:
 - + Entity:
 - Role: Vai trò của người dùng
 - + Relationship:
 - Một người dùng có thể có nhiều role, một role có thể gắn cho nhiều người (n– n)
 - Hệ thống sử dụng Role để kiểm soát quyền truy cập và chức năng của từng người dùng.

- Quản lý sản phẩm, thương hiệu và danh mục hiển thị
 - + Entity:
 - Product: Thông tin sản phẩm.
 - Brand: Thương hiệu sản phẩm.
 - HomeCategory: Danh mục hiển thị trên trang chủ.
 - + Relationship:
 - Một Brand có thể có nhiều Product (1-n).
 - Một HomeCategory có thể có nhiều Product (1-n).
 - Mỗi Product thuộc về một HomeCategory và một Brand.
- Quản lý giỏ hàng
 - + Entity:
 - Cart: Giỏ hàng của người dùng.
 - CartItem: Sản phẩm trong giỏ hàng.
 - + Relationship:
 - Một Cart có thể chứa nhiều CartItem (1-n).
 - Mỗi CartItem liên kết với một Product.
- Quản lý đơn hàng và chi tiết đơn hàng
 - + Entity:
 - Order: Đơn hàng của khách hàng.
 - OrderItem: Chi tiết sản phẩm trong đơn hàng.
 - + Relationship:
 - Một Order có thể có nhiều OrderItem (1-n).
 - Mỗi OrderItem liên kết với một Product.
- Quản lý người dùng và địa chỉ nhận hàng
 - + Entity:
 - User: Người dùng.
 - Address: Địa chỉ nhận hàng của khách hàng.
 - + Relationship:
 - Một User có thể có nhiều Address (1-n).
 - Mỗi Order liên kết với một User và một Address (1-1).
- Quản lý thanh toán và giao hàng
 - + Entity:
 - PaymentOrder: Thông tin thanh toán cho đơn hàng.

- PaymentMethod: Phương thức thanh toán.
 - Transaction: Giao dịch thanh toán.
 - Delivery: Thông tin giao hàng.
 - DeliveryMethod: Phương thức giao hàng.
- + Relationship:
- Mỗi Order có một PaymentOrder và một Delivery (1-1).
 - PaymentOrder liên kết với một PaymentMethod (n-1).
 - Delivery liên kết với một DeliveryMethod (n-1).
 - PaymentOrder có thể có nhiều Transaction (1-n).

3.6 Mô tả cơ sở dữ liệu

3.6.1 Mô tả bảng users

Bảng 3.15 Mô tả bảng users

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	String (UUID)	Định danh duy nhất cho người dùng (tự động sinh ra, dạng UUID)
2	username	String	Tên hiển thị của người dùng.
3	email	String	Địa chỉ email của người dùng.
4	password	String	Mật khẩu của người dùng.
5	shipAddress	Set<Address>	Danh sách các địa chỉ của người dùng.
6	isEnabled	boolean	Trạng thái kích hoạt của tài khoản
7	emailVerified	boolean	Trạng thái xác thực của tài khoản
8	orderIds	List<Long>	Danh sách ID các đơn hàng của người dùng

3.6.2 Mô tả bảng role

Bảng 3.16 Mô tả bảng role

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	name	String	Tên vai trò (ADMIN, USER)
2	description	String	Mô tả vai trò

3.6.3 Mô tả bảng verification_code

Bảng 3.17 Mô tả bảng verification_code

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	Long	Định danh duy nhất cho mã xác thực.
2	otp	String	Mã OTP để xác thực
3	email	String	Email nhận mã xác thực.
4	user	User	Tham chiếu đến người dùng nhận mã xác nhận.

3.6.4 Mô tả bảng cart

Bảng 3.18 Mô tả bảng cart

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	Long	Định danh duy nhất cho giỏ hàng của người dùng
2	userId	String	ID của người dùng tương ứng với giỏ hàng này

3	totalSellingPrice	Long	Tổng giá trị các sản phẩm của giỏ hàng
4	items	List<CartItem>	Danh sách các sản phẩm

3.6.5 Mô tả bảng cart_items

Bảng 3.19 Mô tả bảng cart_items

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	Long	Định danh duy nhất cho sản phẩm trong giỏ hàng.
2	cart	Cart	Tham chiếu đến giỏ hàng chứa sản phẩm
3	productId	String	ID của sản phẩm.
4	productName	String	Tên sản phẩm.
5	productPrice	Long	Giá của sản phẩm.
6	quantity	Integer	Số lượng của sản phẩm của mục.
7	totalPrice	Long	Tổng giá trị cho mục sản phẩm này
8	images	List<String>	Danh sách hình ảnh sản phẩm.

3.6.6 Mô tả bảng orders

Bảng 3.20 Mô tả bảng orders

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	Long	Định danh duy nhất của đơn hàng
2	userId	String	Định danh của người đặt đơn.
3	orderStatus	String	Trạng thái của đơn hàng.
4	createAt	Instant	Thời điểm tạo đơn hàng.
5	items	List<OrderItems>	Danh sách chi tiết các sản phẩm trong đơn hàng.
6	totalPrice	Long	Tổng giá trị đơn hàng.
7	addressId	Long	Địa chỉ để vận chuyển đơn hàng.

3.6.7 Mô tả bảng order_items

Bảng 3.21 Mô tả bảng order_items

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	Long	Định danh duy nhất cho mỗi mục sản phẩm trong đơn hàng.
2	productId	Long	ID của sản phẩm.
3	productName	String	Tên sản phẩm tại thời điểm đặt hàng.
4	productPrice	Long	Giá của sản phẩm tại thời điểm đặt hàng.
5	color	String	Màu của sản phẩm.

6	quantity	Integer	Số lượng của sản phẩm của mục.
7	totalPrice	Long	Tổng giá trị cho mục sản phẩm này
8	images	List<String>	Danh sách hình ảnh sản phẩm tại thời điểm đặt hàng.

3.6.8 Mô tả bảng *home_category*

Bảng 3.22 Mô tả bảng *home_category*

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	Long	Định danh duy nhất cho mỗi danh mục.
2	name	String	Tên hiển thị của danh mục.
3	image	String	Đường dẫn ảnh đại diện cho danh mục.
4	categoryId	String	Mã định danh danh mục.

3.6.9 Mô tả bảng *products*

Bảng 3.23 Mô tả bảng *products*

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	Long	Định danh duy nhất sản phẩm.
2	title	String	Tên của sản phẩm.
3	description	String	Mô tả của sản phẩm.

4	brand	String	Nhãn hàng của sản phẩm.
5	sellingPrice	Long	Giá bán của sản phẩm.
6	color	boolean	Màu sắc của danh mục.
7	categoryId	String	Mã danh mục.
8	images	List<String>	Danh sách ảnh của sản phẩm.
9	createAt	LocalDateTime	Thời điểm tạo ra sản phẩm

3.6.10 Mô tả bảng addresses

Bảng 3.24 Mô tả bảng addresses

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	Long	Định danh duy nhất cho địa chỉ
2	name	String	Tên của người dùng.
3	locality	String	Phường/xã.
4	address	String	Địa chỉ chi tiết.
5	city	String	Tỉnh/thành phố.
6	state	String	Quận
7	mobile	String	Số điện thoại nhận hàng

3.6.11 Mô tả bảng *payment_order*

Bảng 3.25 Mô tả bảng *payment_order*

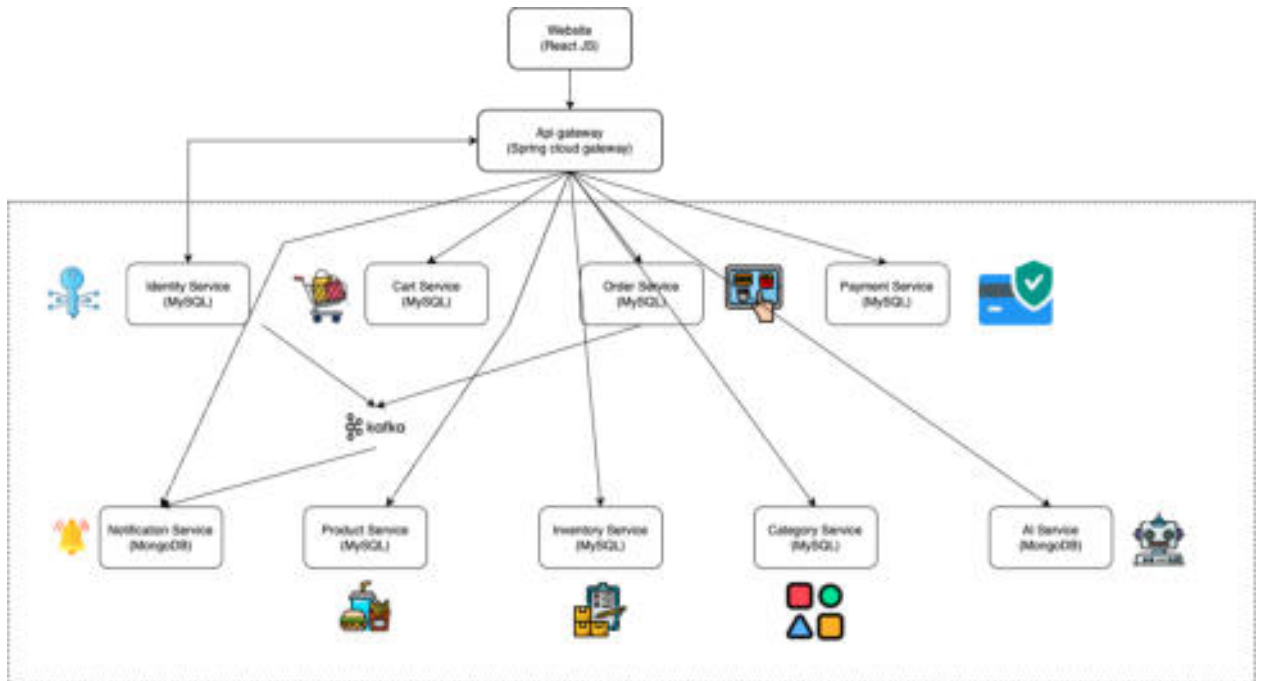
STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	Long	Định danh duy nhất cho đơn thanh toán.
2	amount	Long	Số tiền của đơn hàng.
3	status	PaymentOrderStatus	Trạng thái thanh toán của đơn hàng (SUCCESS, FAILED, PENDING,...).
4	paymentMethod	PaymentMethod	Cách thức thanh toán (COD, VNPay).
5	userId	String	ID của người dùng thanh toán.
6	orderId	Long	ID đơn hàng phải thanh toán.

3.6.12 Mô tả bảng *transaction*

Bảng 3.26 Mô tả bảng *transaction*

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	Long	Định danh duy nhất cho giao dịch.
2	customerId	String	ID của khách hàng thực hiện giao dịch.
3	orderId	Long	ID đơn hàng liên quan đến giao dịch.
4	date	LocalDateTime	Thời điểm thực hiện giao dịch.

3.7 Sơ đồ tổng quan hệ thống



Hình 3.58 Sơ đồ tổng quan hệ thống

3.7.1 Kết luận

- Chương này đã trình bày một cái nhìn tổng quan và chi tiết về quá trình phân tích, thiết kế hệ thống thương mại điện tử theo kiến trúc Microservices. Bắt đầu với việc xác định rõ ràng các tác nhân chính là Admin và User, cùng với các vai trò và quyền hạn cụ thể của mỗi tác nhân, đặt nền móng cho việc hiểu rõ các yêu cầu chức năng của hệ thống.
- Dựa trên việc nhận diện các tác nhân và các tương tác, chương này tiếp tục đi sâu vào mô tả use case. Các use case này không chỉ định nghĩa các chức năng mà hệ thống phải cung cấp từ góc nhìn của người dùng và quản trị viên, mà còn là cơ sở để phát triển các yêu cầu chi tiết cho từng dịch vụ Microservices.
- Tổng thể, Chương 3 cung cấp một bản thiết kế toàn diện và vững chắc cho hệ thống thương mại điện tử, từ việc xác định các thực thể bên ngoài cho đến việc chi tiết hóa các tương tác nội bộ và luồng nghiệp vụ.

Chương 4: XÂY DỰNG WEBSITE THƯƠNG MẠI ĐIỆN TỬ THEO KIẾN TRÚC MICROSERVICES

4.1 Tổng quan về công nghệ sử dụng

4.1.1 Apache Kafka

- Apache Kafka là một nền tảng truyền thông điệp phân tán, có khả năng mở rộng cao, chịu lỗi và được thiết kế để xử lý các luồng dữ liệu thời gian thực lớn. Kafka hoạt động như một hệ thống publish-subscribe messaging cho phép các ứng dụng (producers) gửi dữ liệu vào các topics (chủ đề), và các ứng dụng khác (consumers) có thể đăng ký để nhận dữ liệu từ các topics này.

4.1.2 Docker

- Docker là một nền tảng containerization cho phép đóng gói một ứng dụng và tất cả các phụ thuộc của nó (thư viện, framework, runtime, công cụ hệ thống, cấu hình) vào một đơn vị di động gọi là container. Container là một môi trường chạy độc lập và nhất quán, không phụ thuộc vào hệ điều hành host.

4.1.3 Kubernetes

- Kubernetes (thường được gọi là K8s) là một hệ thống orchestration container mã nguồn mở, được thiết kế để tự động hóa việc triển khai, mở rộng và quản lý các ứng dụng containerized. Kubernetes cung cấp một nền tảng mạnh mẽ để quản lý vòng đời của các container trên một cụm các máy chủ.

4.1.4 ReactJS và Redux Toolkit

- ReactJS: Là một thư viện JavaScript mã nguồn mở được sử dụng để xây dựng giao diện người dùng (UI) động và tương tác. React tiếp cận việc xây dựng UI theo kiểu component-based, nơi giao diện được chia thành các thành phần độc lập, có khả năng tái sử dụng và quản lý trạng thái cục bộ. React sử dụng một Virtual DOM (Document Object Model ảo) để tối ưu hóa quá trình cập nhật giao diện, chỉ thực hiện thay đổi thực tế trên DOM khi cần thiết, giúp tăng hiệu suất ứng dụng.
- Redux Toolkit: Là một bộ công cụ (opinionated toolset) được xây dựng dựa trên Redux, nhằm đơn giản hóa quá trình phát triển ứng dụng React với Redux. Redux

là một thư viện quản lý trạng thái toàn cục (global state management) cho các ứng dụng JavaScript lớn và phức tạp. Redux Toolkit cung cấp các API tiện lợi để cấu hình Redux store, định nghĩa reducers, actions và selectors một cách dễ dàng hơn so với Redux thuần túy, giúp giảm thiểu boilerplate code và tăng tốc độ phát triển.

4.1.5 Cơ sở dữ liệu quan hệ MySQL

- Là một hệ thống quản lý cơ sở dữ liệu quan hệ mã nguồn mở rất phổ biến. Nó được sử dụng rộng rãi trong các ứng dụng và các ứng dụng doanh nghiệp khác. MySQL tuân thủ mô hình cơ sở dữ liệu quan hệ, nơi dữ liệu được tổ chức thành các bảng (tables) có cấu trúc cố định, với các mối quan hệ được xác định rõ ràng giữa các bảng.

4.1.6 Cơ sở dữ liệu phi quan hệ MongoDB

- MongoDB là một cơ sở dữ liệu NoSQL dạng tài liệu (document-oriented database). Thay vì sử dụng bảng, hàng và cột như RDBMS, MongoDB lưu trữ dữ liệu dưới dạng các tài liệu JSON (hoặc BSON - Binary JSON, một phiên bản nhị phân của JSON), cho phép cấu trúc dữ liệu linh hoạt hơn.

4.1.7 Cơ sở dữ liệu Vector ChromaDB

- ChromaDB là một cơ sở dữ liệu vector mã nguồn mở được thiết kế đặc biệt để lưu trữ, quản lý và tìm kiếm các vector nhúng. Vector nhúng là các biểu diễn số học của dữ liệu (văn bản, hình ảnh, âm thanh, v.v.) trong một không gian đa chiều, nơi các mục có ý nghĩa ngữ nghĩa tương tự sẽ nằm gần nhau. ChromaDB được sử dụng rộng rãi trong các ứng dụng AI/ML, đặc biệt là Xử lý ngôn ngữ tự nhiên (NLP) và các hệ thống tìm kiếm ngữ nghĩa.

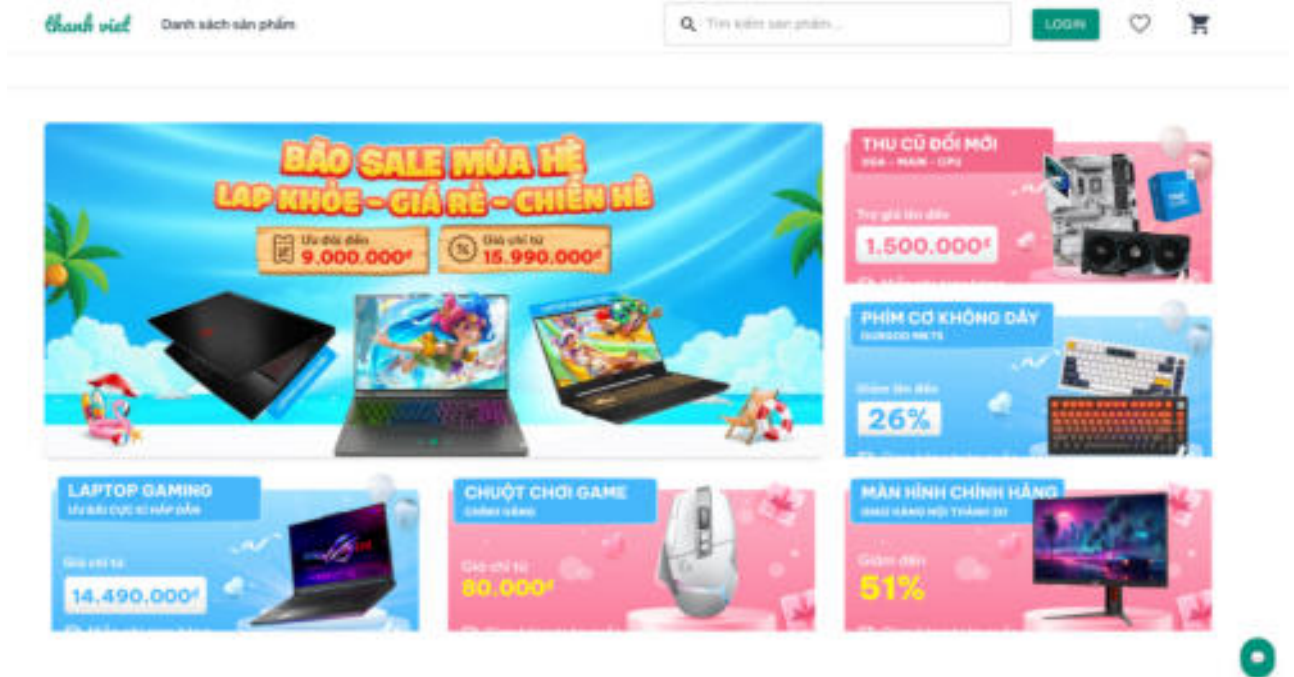
4.1.8 Công cụ hỗ trợ

- Trong quá trình phát triển và kiểm thử hệ thống, em đã sử dụng nhiều công cụ hỗ trợ nhằm tăng hiệu quả làm việc và đảm bảo chất lượng phần mềm:
 - + Git: Sử dụng để quản lý mã nguồn, hỗ trợ làm việc nhóm, theo dõi lịch sử thay đổi và triển khai CI/CD.
 - + Postman: Hỗ trợ kiểm thử các API RESTful một cách nhanh chóng, trực quan và dễ cấu hình.

- + IntelliJ IDEA: Môi trường phát triển tích hợp (IDE) mạnh mẽ, hỗ trợ lập trình Java và Spring Boot, giúp tăng năng suất viết mã và dễ dàng quản lý dự án microservices.
- + PyCharm: IDE chuyên dụng cho Python, được sử dụng trong việc phát triển các thành phần xử lý AI, chatbot hoặc tích hợp RAG-based systems.
- + WebStorm: IDE hỗ trợ phát triển giao diện người dùng (frontend) bằng JavaScript, React và Redux, giúp tăng tốc quá trình thiết kế và kiểm thử giao diện.

4.2 Giao diện hệ thống

4.2.1 Trang chủ người dùng



Hình 4.1 Trang chủ người dùng

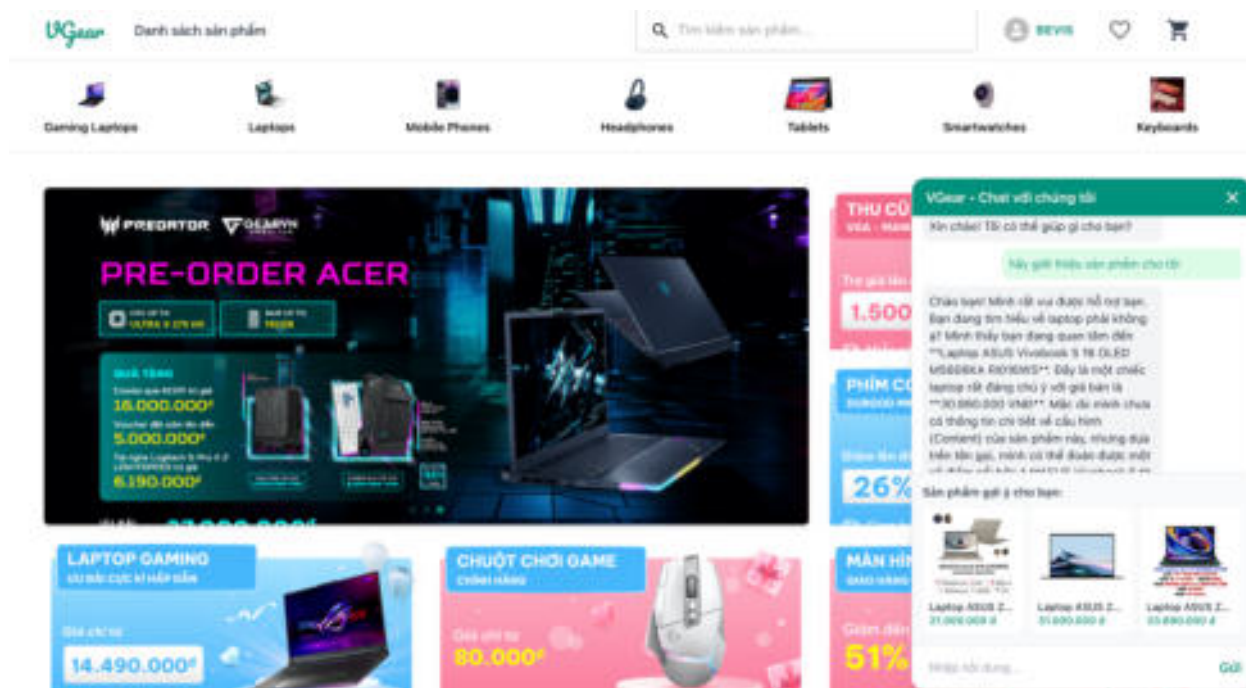
Bảng 4.1 Đặc tả giao diện trang chủ

NO	Component Name	Required	Format	Reference	Note
1	Header	Có	Section	Thanh trên cùng của trang chủ	Chứa logo, menu, thanh tìm kiếm,

Xây dựng website bán hàng theo kiến trúc Microservices

					nút đăng nhập, giỏ hàng
2	Logo	Có	Thẻ h1	Biểu tượng	Nhấn vào để về trang chủ
3	Danh sách sản phẩm	Có	Thẻ h1	Dùng để điều hướng	Chuyển đến trang danh sách sản phẩm
4	Thanh tìm kiếm	Có	Input	Tìm kiếm sản phẩm của cửa hàng	Tìm kiếm theo từ khoá
5	Login	Có	Button	Đến trang đăng nhập	Nhấn để chuyển đến trang đăng nhập
6	Biểu tượng yêu thích	Có	Icon	Đến trang danh sách sản phẩm yêu thích	Chuyển đến trang danh sách sản phẩm yêu thích
7	Biểu tượng giỏ hàng	Có	Icon	Hiện thị số sản phẩm trong giỏ hàng và dùng để chuyển trang giỏ hàng	Nhấn vào để chuyển đến giỏ hàng
8	Banner chính	Có	Image	Hình ảnh lớn, nổi bật	
9	Banner phụ	Có	Image	Các hình ảnh nhỏ	

4.2.2 Giao diện chatbot

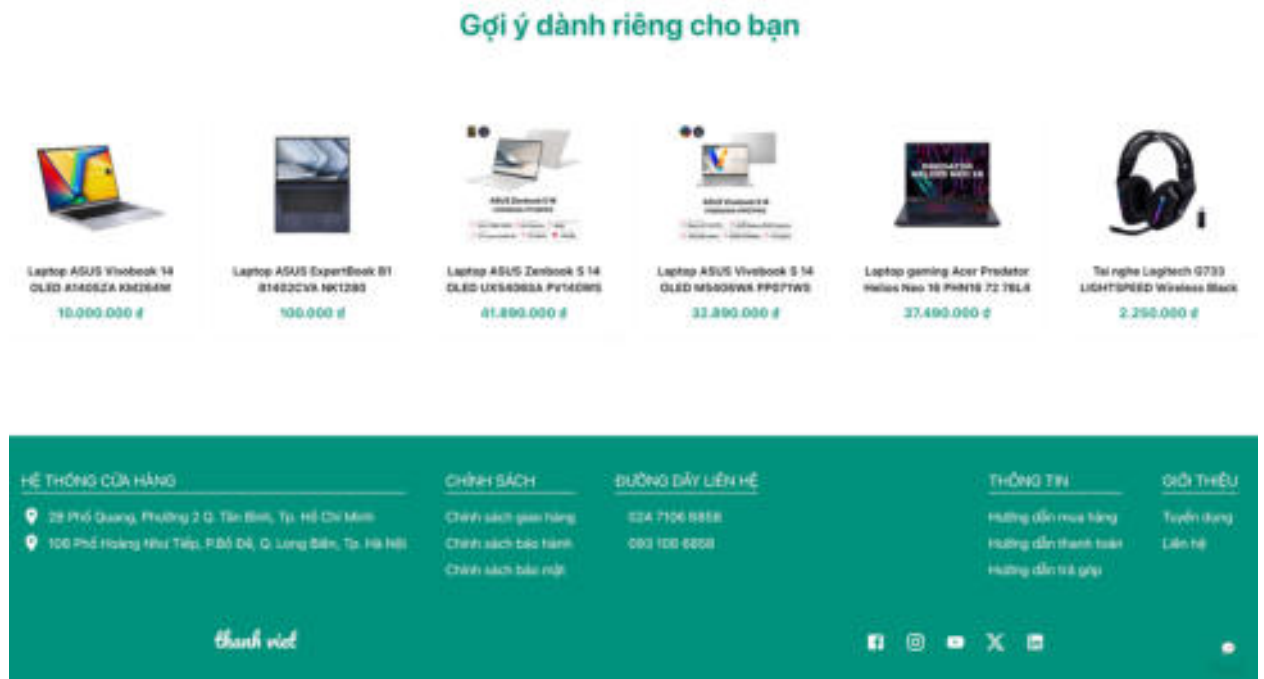


Hình 4.2 Giao diện chatbot

Bảng 4.2 Đặc tả giao diện chatbot

NO	Component Name	Required	Format	Reference	Note
1	Header	Có	Section	Tên và nút đóng chatbot	Chứa label và nút đóng chatbot
2	Khu vực hội thoại	Có	Textarea	Hiện thị nội dung trò chuyện giữa chatbot và người dùng	
3	Thanh nhập liệu	Có	Section	Dùng để nhập nội dung và gửi đi query của người dùng	Bao gồm phần input và nút gửi
4	Gợi ý sản phẩm	Có	Section	Hiện thị các sản phẩm gợi ý cho người dùng	Chỉ xuất hiện khi người dùng chat với bot

4.2.3 Giao diện phần gợi ý sản phẩm

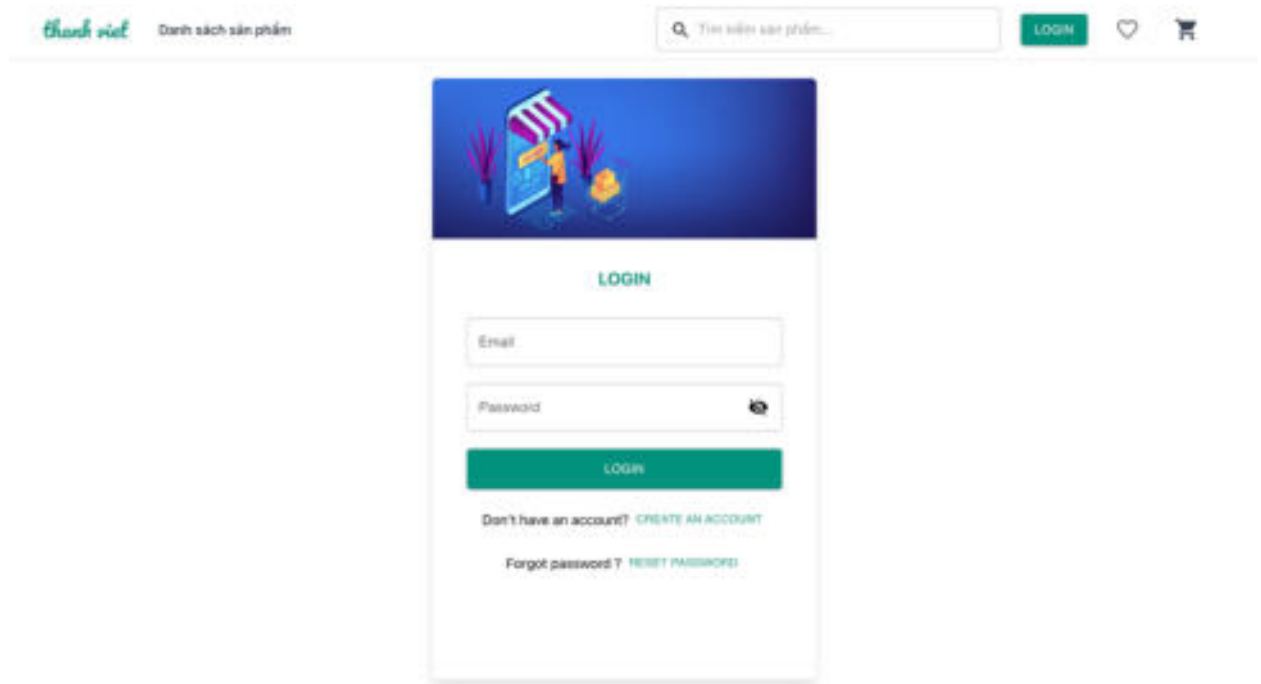


Hình 4.3 Giao diện gợi ý sản phẩm

Bảng 4.3 Đặc tả giao diện gợi ý sản phẩm

NO	Component Name	Required	Format	Reference	Note
1	Ảnh sản phẩm	Có	Image	Hiển thị hình ảnh của sản phẩm	
2	Tên sản phẩm	Có	Text	Hiển thị tên của sản phẩm được gợi ý cho người dùng	
3	Giá sản phẩm	Có	Text	Hiển thị giá của sản phẩm được gợi ý	
4	Footer	Có	Section	Hiển thị các thông tin của cửa hàng	

4.2.4 Trang đăng nhập



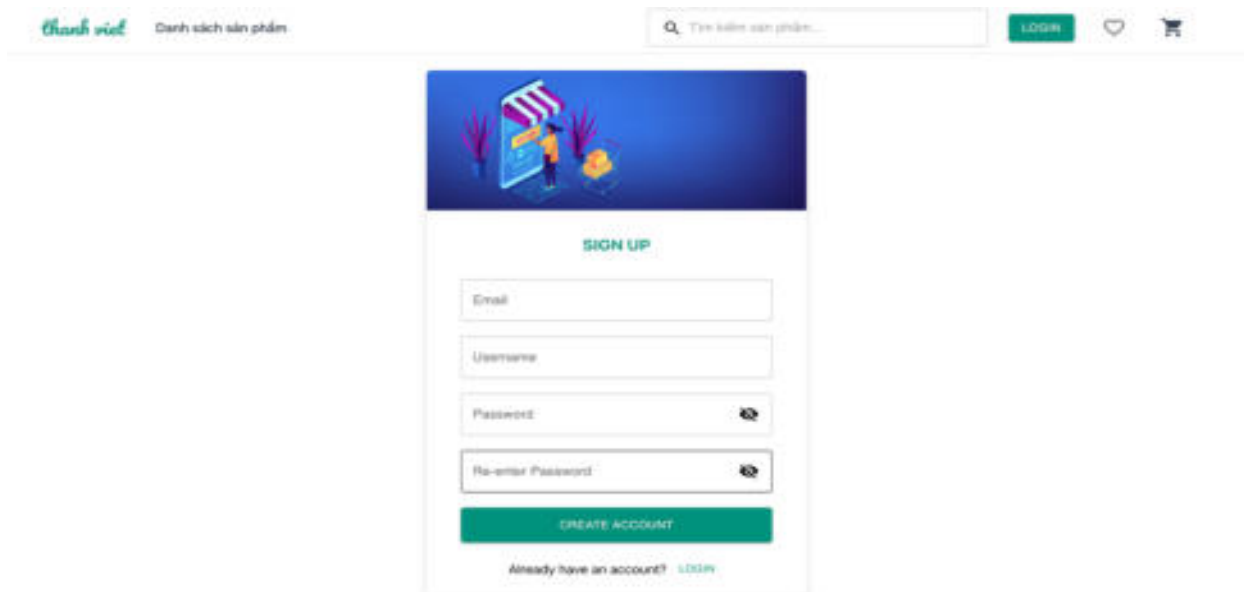
Hình 4.4 Trang đăng nhập

Bảng 4.4 Đặc tả giao diện đăng nhập

NO	Component Name	Required	Format	Reference	Note
1	Hình ảnh	Có	Image	Ảnh phía trên cùng màn hình	
2	LOGIN	Có	Trường văn bản	Hiển thị tên màn hình	Đặt ở dưới hình ảnh
3	Email	Có	Input	Nhập email để đăng nhập	Chuyển đến trang danh sách sản phẩm
4	Password	Có	Input	Nhập mật khẩu để đăng nhập	Có nút ẩn/hiện mật khẩu
5	Nút Login	Có	Button	Gửi thông tin người dùng lên server	Nhấn để chuyển đến trang đăng nhập
6	CREATE AN ACCOUNT	Có	Link	Chuyển đến trang đăng kí	

7	RESET PASSWORD	Có	Link	Chuyển đến trang lấy lại mật khẩu
---	-------------------	----	------	--------------------------------------

4.2.5 Trang đăng kí



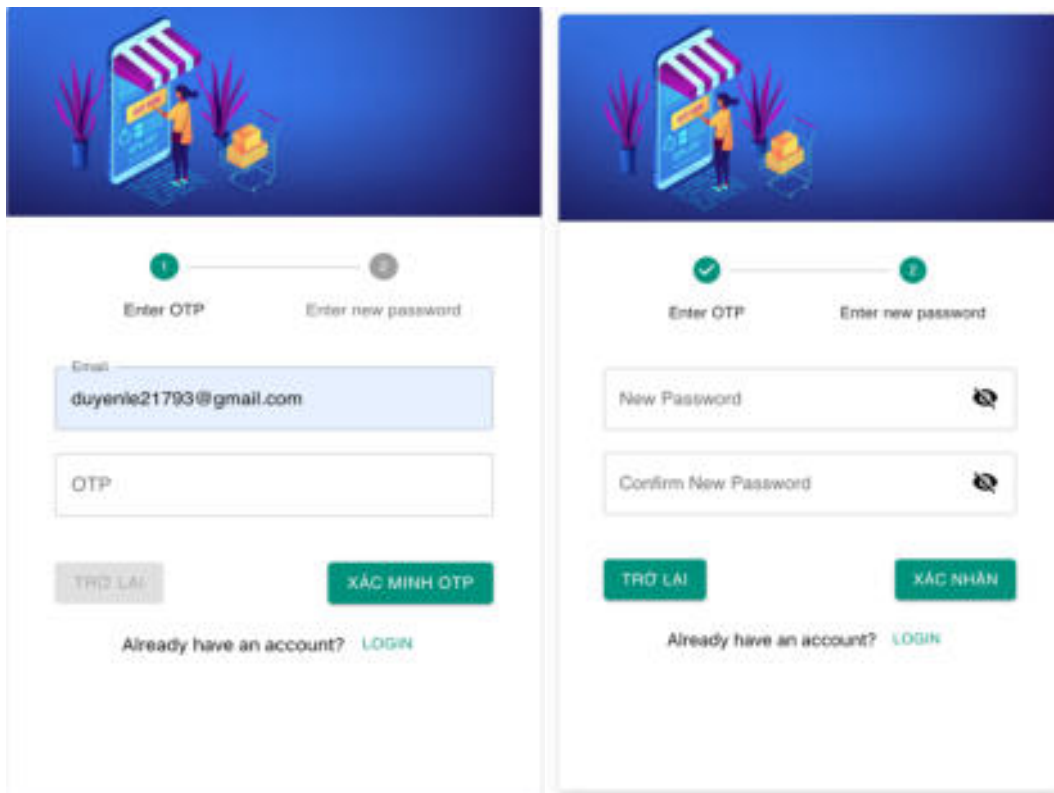
Hình 4.5 Trang đăng kí

Bảng 4.5 Đặc tả giao diện đăng kí

NO	Component Name	Required	Format	Reference	Note
1	Hình ảnh	Có	Image	Ảnh phía trên cùng màn hình	
2	SIGN UP	Có	Trường văn bản	Hiển thị tên màn hình	Đặt ở dưới hình ảnh
3	Email	Có	Input	Nhập email để đăng nhập	Kiểm tra xem email đã tồn tại
4	Username	Có	Input	Nhập vào tên người dùng	Phải nhiều hơn bốn kí tự
5	Password	Có	Input	Nhập mật khẩu để đăng nhập	Có nút ẩn/hiện mật khẩu
6	Re-enter password	Có	Input	Nhập vào lại mật khẩu	Có nút ẩn/hiện mật khẩu, kiểm

					tra trùng khớp với mật khẩu
7	CREATE ACCOUNT	Có	Button	Dùng để gửi thông tin đăng nhập lên server	
8	Nút Login	Có	Link	Chuyển đến trang đăng nhập	Nhấn để chuyển đến trang đăng nhập

4.2.6 Trang lấy lại mật khẩu



Hình 4.6 Trang lấy lại mật khẩu người dùng

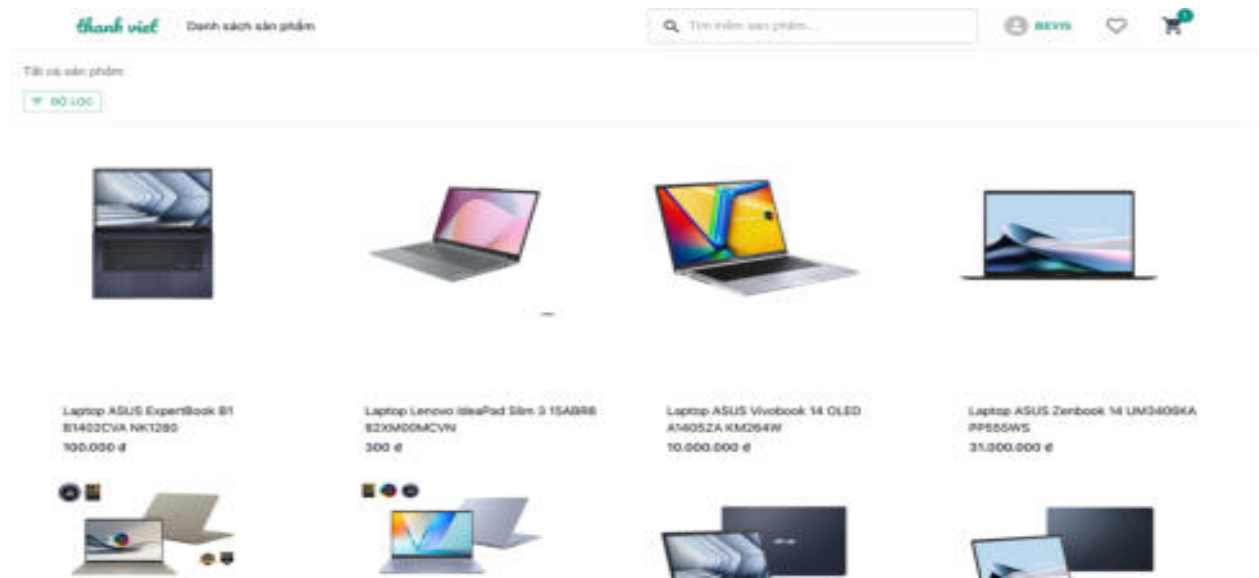
Bảng 4.6 Đặc tả giao diện lấy lại mật khẩu

NO	Component Name	Required	Format	Reference	Note
----	----------------	----------	--------	-----------	------

Xây dựng website bán hàng theo kiến trúc Microservices

1	Hình ảnh	Có	Image	Ảnh phía trên cùng màn hình	
2	Thanh tiến trình	Có	Stepper	Hiển thị 2 bước: nhập OTP và nhập mật khẩu mới	Đánh dấu bước hiện tại
3	Email	Có	Input	Nhập email để hệ thống sẽ gửi mã OTP	Kiểm tra có email tồn tại không
4	OTP	Có	Input	Nhập vào mã OTP đã được gửi tới thông qua email	Kiểm tra xem có trùng với OTP đã gửi hay không
5	New Password	Có	Input	Nhập mật khẩu mới	Có nút ẩn/hiện mật khẩu
6	Confirm new password	Có	Input	Nhập vào lại mật khẩu mới	Có nút ẩn/hiện mật khẩu, kiểm tra trùng khớp với mật khẩu
7	Xác minh OTP	Có	Button	Dùng để gửi thông tin lên server	
8	Xác nhận	Có	Button	Gửi thông tin lên server để đổi mật khẩu	
9	Trở lại	Có	Button	Quay trở lại bước trước	
10	Nút Login	Có	Link	Chuyển đến trang đăng nhập	Nhấn để chuyển đến trang đăng nhập

4.2.7 Trang danh sách sản phẩm

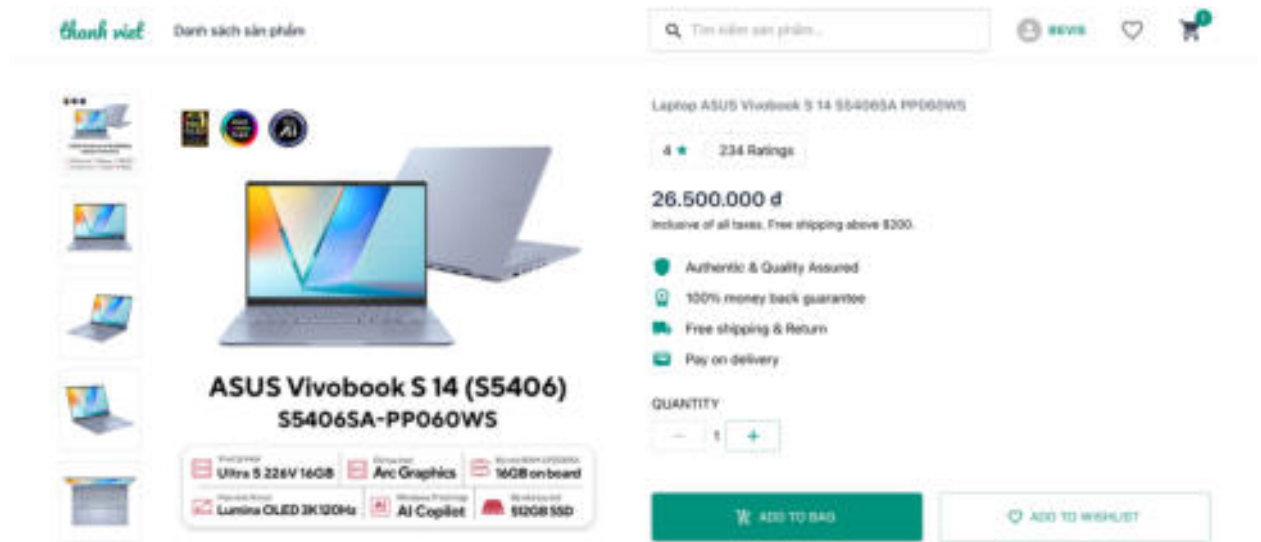


Hình 4.7 Trang danh sách sản phẩm

Bảng 4.7 Đặc tả giao diện danh sách sản phẩm

NO	Component Name	Required	Format	Reference	Note
1	Bộ lọc sản phẩm	Có	Button	Hiện thị các trường để lọc sản phẩm	
2	Danh sách sản phẩm	Có	Grid	Hiện thị danh sách sản phẩm	Danh sách sẽ thay đổi nếu người dùng lọc
3	Ảnh sản phẩm	Có	Image	Hiện thị ảnh sản phẩm	Nếu người dùng trở vào thì nó sẽ hiện các ảnh khác, nhấn vào để đến trang chi tiết sản phẩm
4	Tên sản phẩm	Có	Text	Hiện thị tên sản phẩm	
5	Giá sản phẩm	Có	Text	Hiện thị giá sản phẩm	

4.2.8 Trang chi tiết sản phẩm



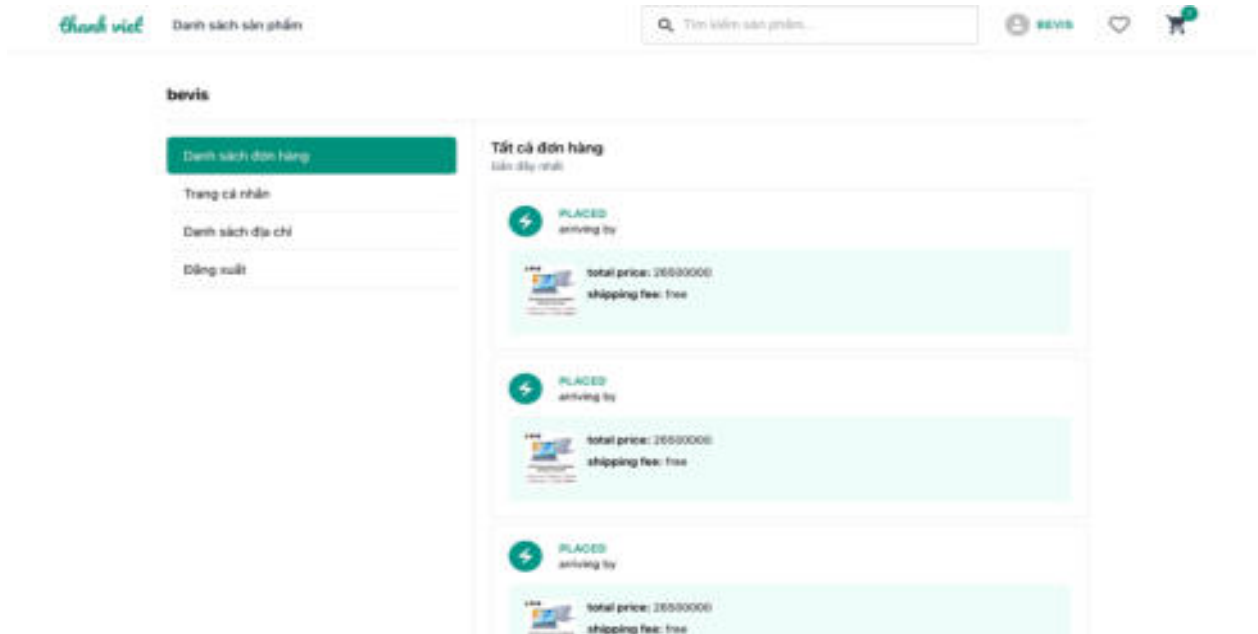
Hình 4.8 Trang chi tiết sản phẩm

Bảng 4.8 Đặc tả giao diện chi tiết sản phẩm

NO	Component Name	Required	Format	Reference	Note
1	Hình ảnh chính	Có	Image	Hiển thị hình ảnh của sản phẩm	Ảnh lớn ở trung tâm
2	Danh sách ảnh phụ	Có	Image List	Danh sách các hình ảnh khác của sản phẩm	Nhấn vào để xem
3	Tên sản phẩm	Có	Text	Hiển thị tên sản phẩm	
4	Giá sản phẩm	Có	Text	Hiển thị giá sản phẩm	
5	Quantity	Có	Button	Nút tăng/giảm số lượng sản phẩm	Thêm vào giỏ hàng sản phẩm với số lượng tương ứng
6	Add to bag	Có	Button	Thêm sản phẩm vào giỏ hàng	

7	Add to wishlist	Có	Button	Thêm vào danh sách yêu thích	
---	-----------------	----	--------	------------------------------	--

4.2.9 Trang danh sách đơn hàng của người dùng



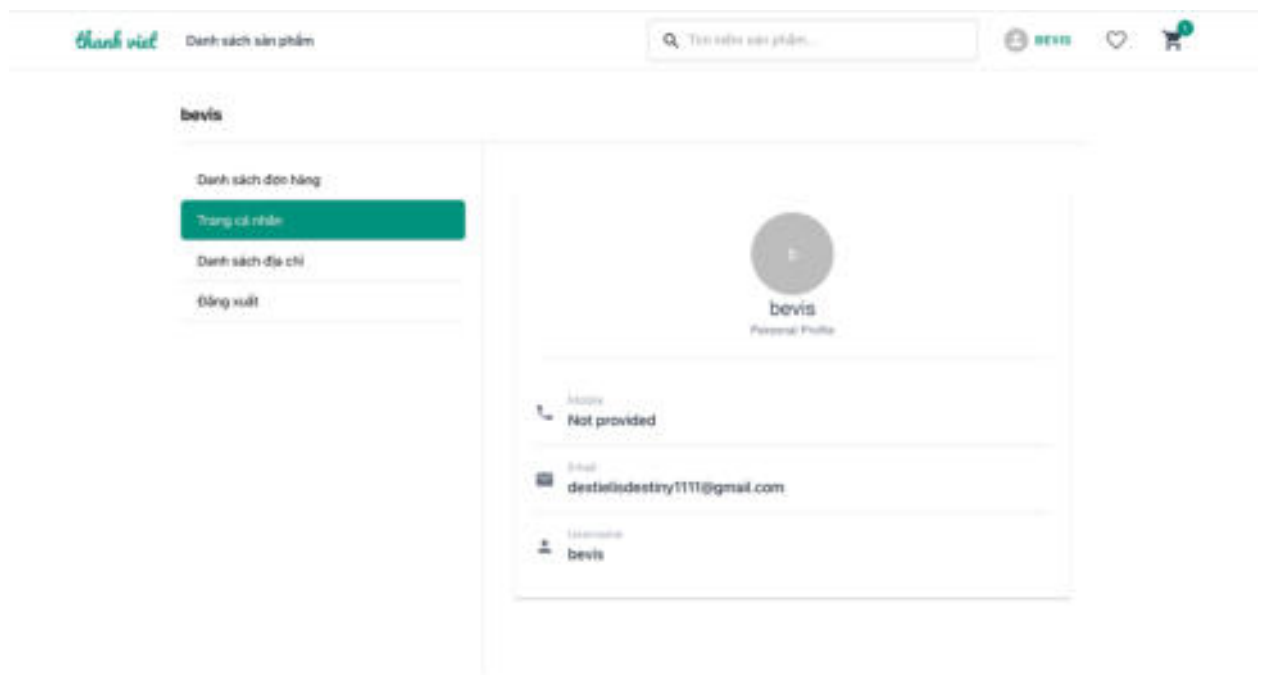
Hình 4.9 Trang danh sách đơn hàng của người dùng

Bảng 4.9 Đặc tả giao diện danh sách đơn hàng

NO	Component Name	Required	Format	Reference	Note
1	Side bar	Có	List	Bao gồm các mục	Người dùng nhấn vào để đi sang các trang khác
2	Danh sách đơn hàng	Có	List	Danh sách các đơn hàng của người dùng	Nhấn vào để xem chi tiết đơn hàng
3	Thông tin đơn hàng	Có	Card	Hiển thị thông tin của đơn hàng	Bao gồm trạng thái, tổng tiền, hình ảnh sản phẩm
4	Trạng thái đơn hàng	Có	Text	Hiển thị trạng thái đơn hàng	

5	Hình ảnh sản phẩm	Có	Image	Hiển thị hình ảnh sản phẩm	
6	Tổng tiền đơn hàng	Có	Text	Hiển thị tổng số tiền của đơn hàng này	
7	Phí vận chuyển	Có	Text	Hiển thị phí vận chuyển	

4.2.10 Trang cá nhân người dùng



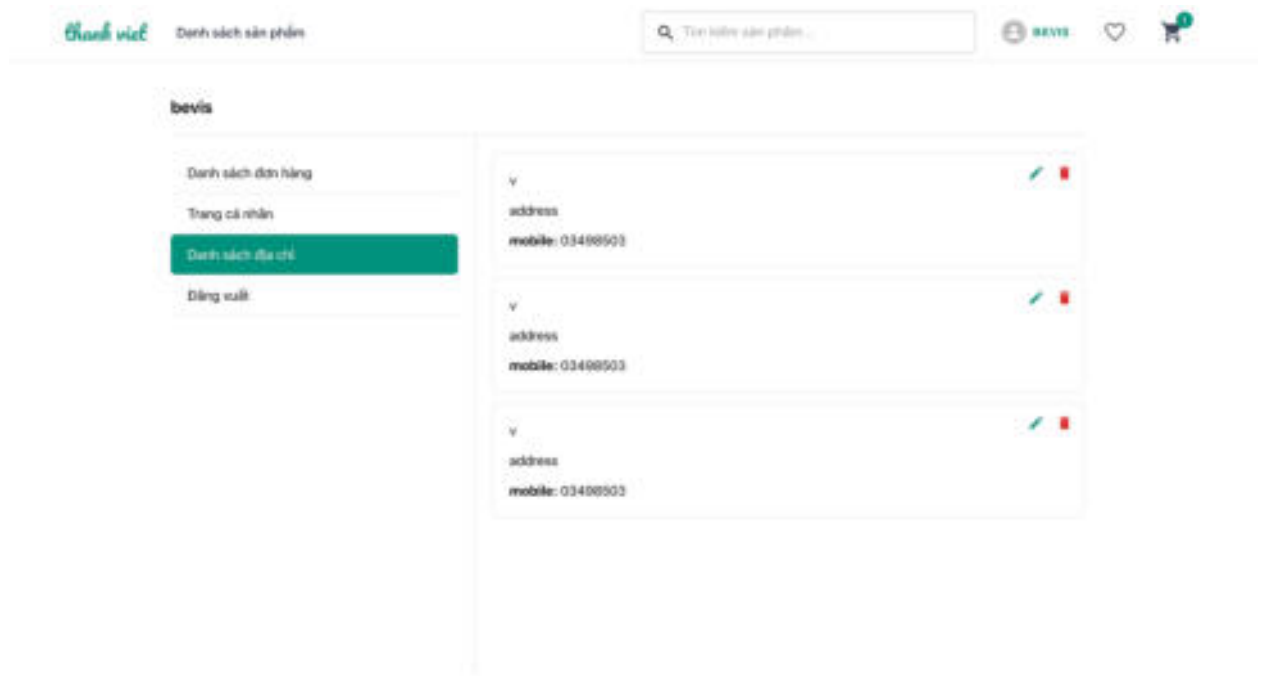
Hình 4.10 Trang cá nhân người dùng

Bảng 4.10 Đặc tả giao diện cá nhân

NO	Component Name	Required	Format	Reference	Note
1	Ảnh đại diện	Có	Image	Hiển thị hình ảnh của người dùng	
2	Tên người dùng	Có	Text	Hiển thị tên của người dùng	
3	Email	Có	Text	Hiển thị email của người dùng	

4	Username	Có	Text	Hiển thị tên người dùng	
---	----------	----	------	-------------------------	--

4.2.11 Trang danh sách các địa chỉ



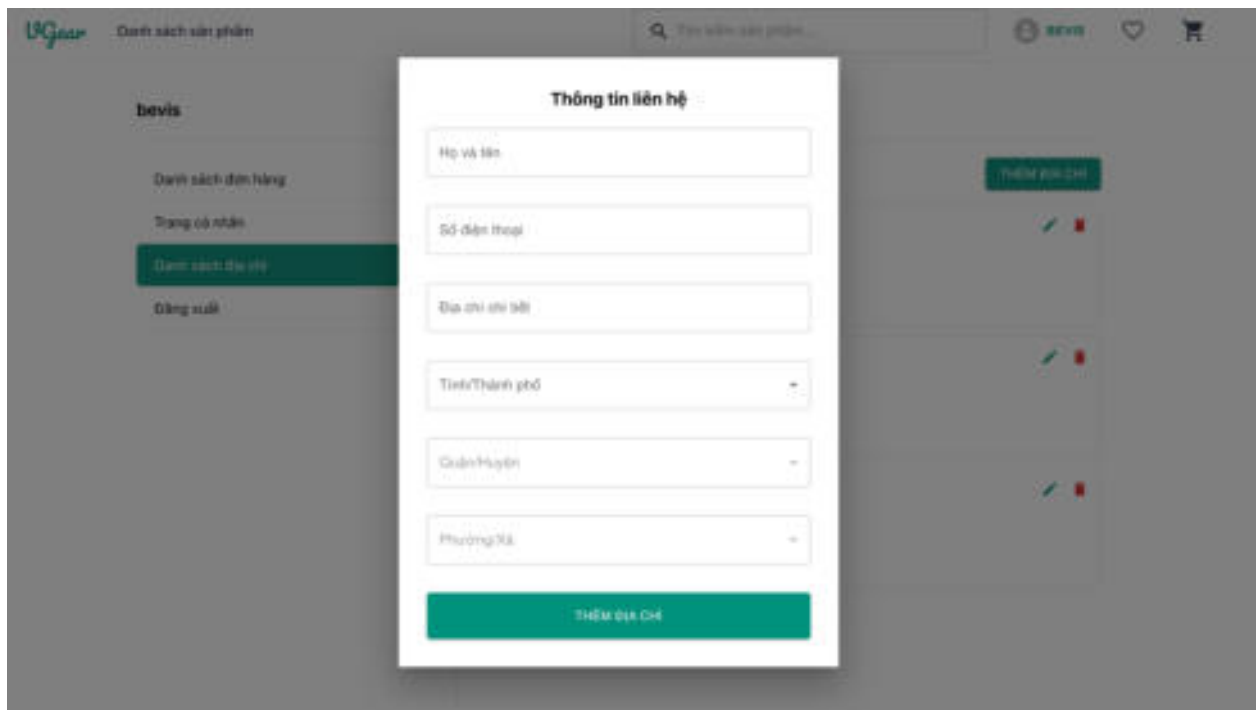
Hình 4.11 Trang danh sách địa chỉ

Bảng 4.11 Đặc tả giao diện danh sách địa chỉ

NO	Component Name	Required	Format	Reference	Note
1	Danh sách các địa chỉ	Có	List	Danh sách các địa chỉ đã lưu của người dùng	
2	Địa chỉ	Có	Card	Hiển thị thông tin của địa chỉ	Người dùng có thể chỉnh sửa hoặc xóa
3	Tên	Có	Text	Hiển thị tên của người dùng	
4	Địa chỉ chi tiết	Có	Text	Hiển thị địa chỉ chi tiết của người dùng	
5	Mobile	Có	Text	Hiển thị số điện thoại	

6	Thêm địa chỉ mới	Có	Button	Mở modal thêm địa chỉ mới cho người dùng	
7	Sửa	Có	Button	Mở modal để người dùng sửa thông tin địa chỉ	
8	Xoá	Có	Button	Xoá địa chỉ	Trước khi xoá thì sẽ hiện cảnh báo cho người dùng

4.2.12 Giao diện thêm địa chỉ mới



Hình 4.12 Giao diện thêm địa chỉ mới

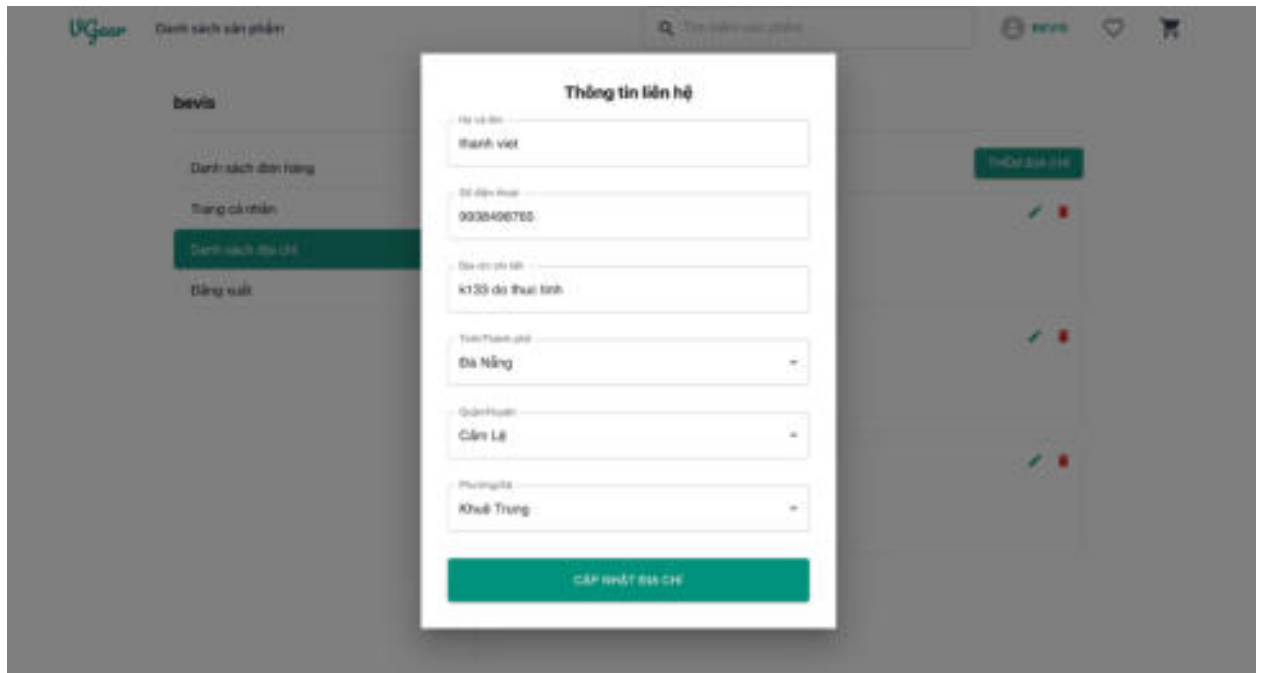
Bảng 4.12 Đặc tả giao diện thêm địa chỉ mới

NO	Component Name	Required	Format	Reference	Note
1	Thông tin liên hệ	Có	Label	Hiển thị tên của giao diện	

Xây dựng website bán hàng theo kiến trúc Microservices

2	Họ và tên	Có	Input	Nhập vào tên người nhận hàng	
3	Số điện thoại	Có	Input	Nhập vào số điện thoại của người nhận hàng	
4	Địa chỉ chi tiết	Có	Input	Nhập vào địa chỉ chi tiết của người dùng	
5	Tỉnh/Thành phố	Có	Select	Chọn tỉnh/thành phố của người dùng	Chỉ khi chọn tỉnh/thành phố thì mới chọn được quận
6	Quận	Có	Select	Chọn quận của người dùng	Chỉ khi chọn quận thì mới chọn được phường/xã
7	Phường/Xã	Có	Select	Chọn phường/xã của người dùng	
8	Thêm địa chỉ	Có	Button	Gửi thông tin lên server để thêm địa chỉ mới cho người dùng	Hiển thị thông báo nếu thành công/lỗi

4.2.13 Giao diện sửa thông tin địa chỉ



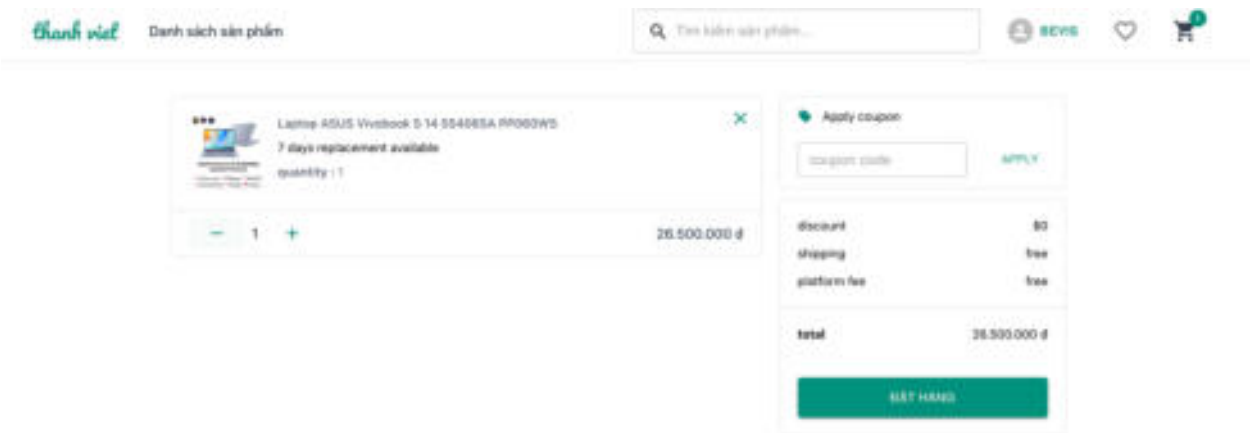
Hình 4.13 Giao diện sửa thông tin địa chỉ

Bảng 4.13 Đặc tả giao diện sửa thông tin địa chỉ

NO	Component Name	Required	Format	Reference	Note
1	Thông tin liên hệ	Có	Label	Hiển thị tên của giao diện	
2	Họ và tên	Có	Input	Hiển thị tên của người nhận hàng	
3	Số điện thoại	Có	Input	Hiển thị số điện thoại của người nhận hàng	
4	Địa chỉ chi tiết	Có	Input	Hiển thị địa chỉ chi tiết của người dùng	
5	Tỉnh/Thành phố	Có	Select	Hiển thị tỉnh/thành phố của người dùng	Chỉ khi chọn tỉnh/thành phố thì mới chọn được quận

6	Quận	Có	Select	Hiển thị quận của người dùng	Chỉ khi chọn quận thì mới chọn được phường/xã
7	Phường/Xã	Có	Select	Hiển thị phường/xã của người dùng	
8	Cập nhật địa chỉ	Có	Button	Gửi thông tin lên server để cập nhật thông tin địa chỉ cho người dùng	Hiển thị thông báo nếu thành công/lỗi

4.2.14 Trang giỏ hàng của người dùng



Hình 4.14 Trang giỏ hàng của người dùng

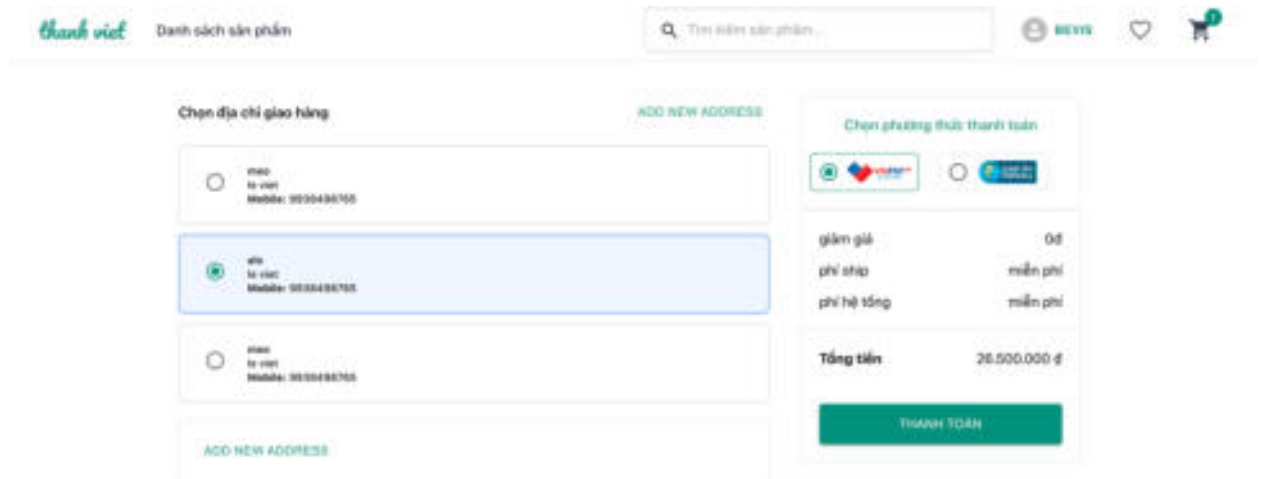
Bảng 4.14 Đặc tả giao diện giỏ hàng

NO	Component Name	Required	Format	Reference	Note
1	Danh sách sản phẩm trong giỏ hàng	Có	List	Danh sách các sản phẩm trong giỏ hàng	

Xây dựng website bán hàng theo kiến trúc Microservices

2	Sản phẩm	Có	Card	Hiển thị thông tin của sản phẩm trong giỏ hàng	Nếu người dùng thêm sản phẩm giống thì số lượng sẽ tăng lên
3	Tên	Có	Text	Hiển thị tên của sản phẩm	
4	Ảnh sản phẩm	Có	Image	Hiển thị ảnh của sản phẩm	
5	Số lượng	Có	Text	Hiển thị số lượng sản phẩm trong giỏ	
6	Nút thêm/giảm	Có	Button	Dùng để tăng/giảm số lượng của sản phẩm có sẵn trong giỏ	
7	Đặt hàng	Có	Button	Di chuyển đến trang chọn phương thức thanh toán	
8	Nút xoá sản phẩm	Có	Button	Xoá sản phẩm khỏi giỏ hàng	

4.2.15 Trang thanh toán

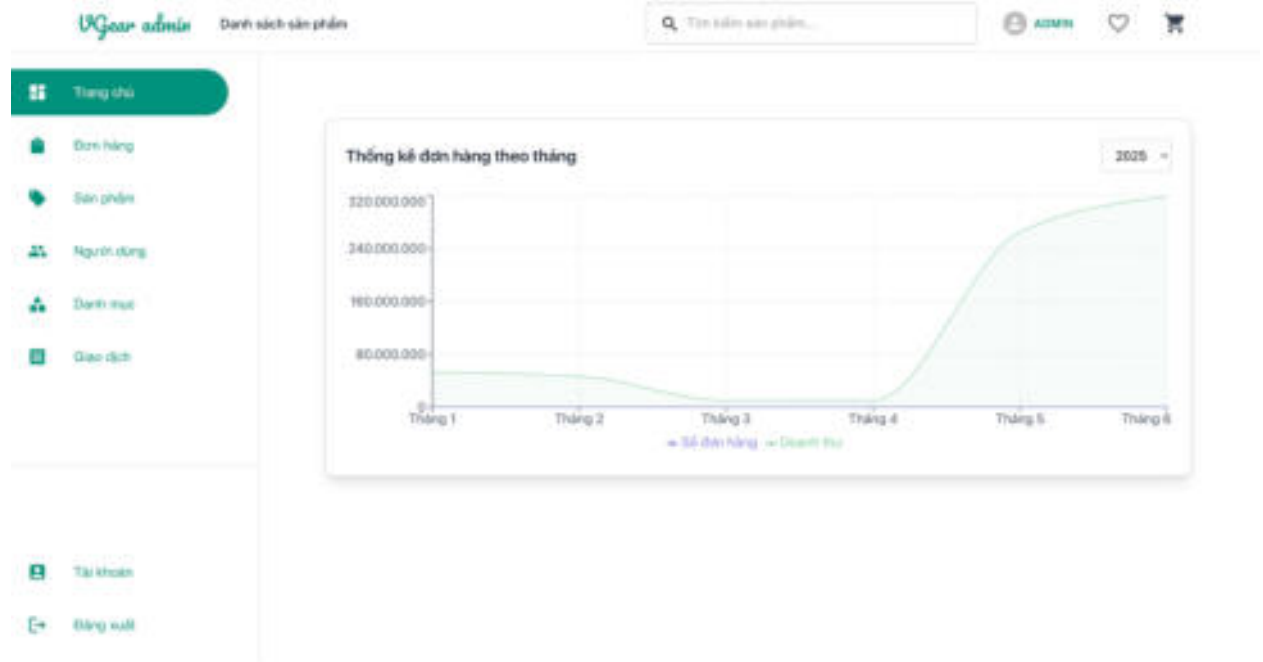


Hình 4.15 Trang thanh toán

Bảng 4.15 Đặc tả giao diện thanh toán

NO	Component Name	Required	Format	Reference	Note
1	Danh sách các địa chỉ giao hàng	Có	List	Hiển thị danh sách các địa chỉ của người dùng	Người dùng phải chọn một địa chỉ để tiếp tục
2	Phương thức thanh toán	Có	Card	Hiển thị các phương thức thanh toán	Phải chọn một trong hai
3	Giảm giá	Có	Text	Số tiền được giảm giá	
4	Phí ship	Có	Text	Hiển thị phí giao hàng	
5	Tổng tiền	Có	Text	Hiển thị số tiền của đơn hàng cần phải thanh toán	
6	Thanh toán	Có	Button	Tiến hành đặt hàng	

4.2.16 Trang chủ giao diện admin

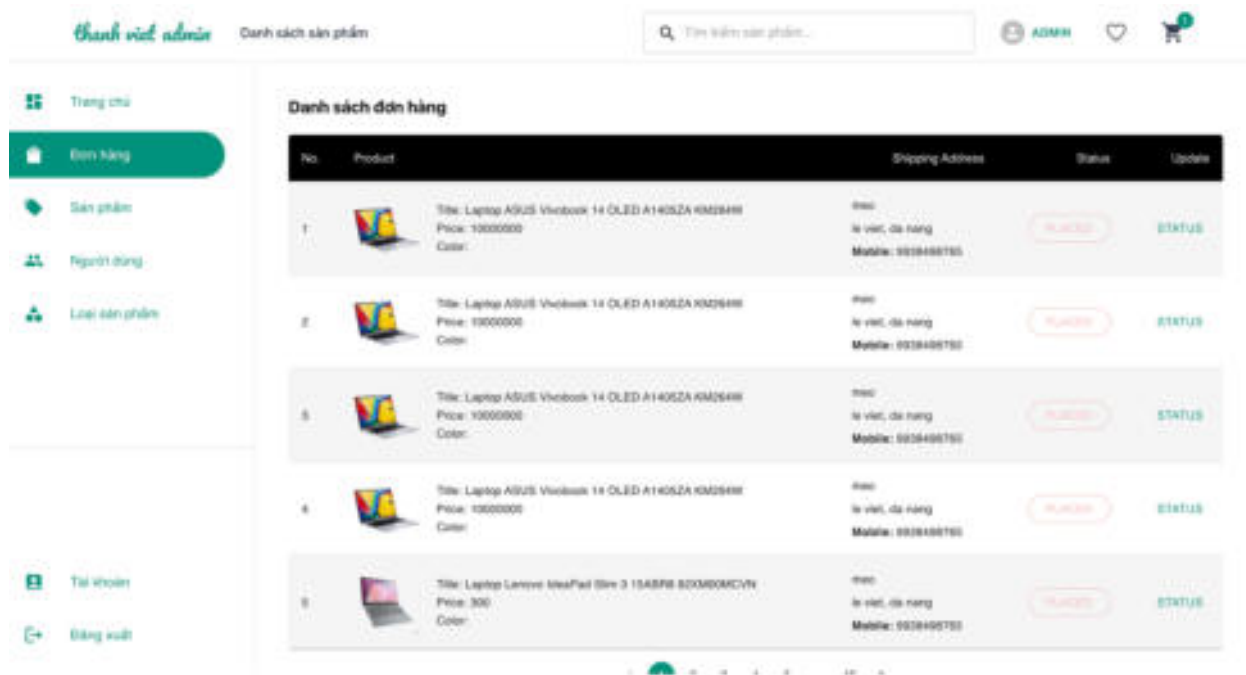


Hình 4.16 Giao diện trang chủ admin

Bảng 4.16 Đặc tả giao diện trang chủ admin

NO	Component Name	Required	Format	Reference	Note
1	Card	Có	Card	Hiển thị thông tin đơn hàng của 12 tháng	
2	Tiêu đề	Có	Text	Hiển thị tên của biểu đồ	
3	2025	Có	Dropdown	Chọn năm muốn hiển thị dữ liệu	
4	Biểu đồ đường	Có	Line chart	Bao gồm 2 đường: số đơn hàng và doanh thu	

4.2.17 Trang danh sách các đơn hàng

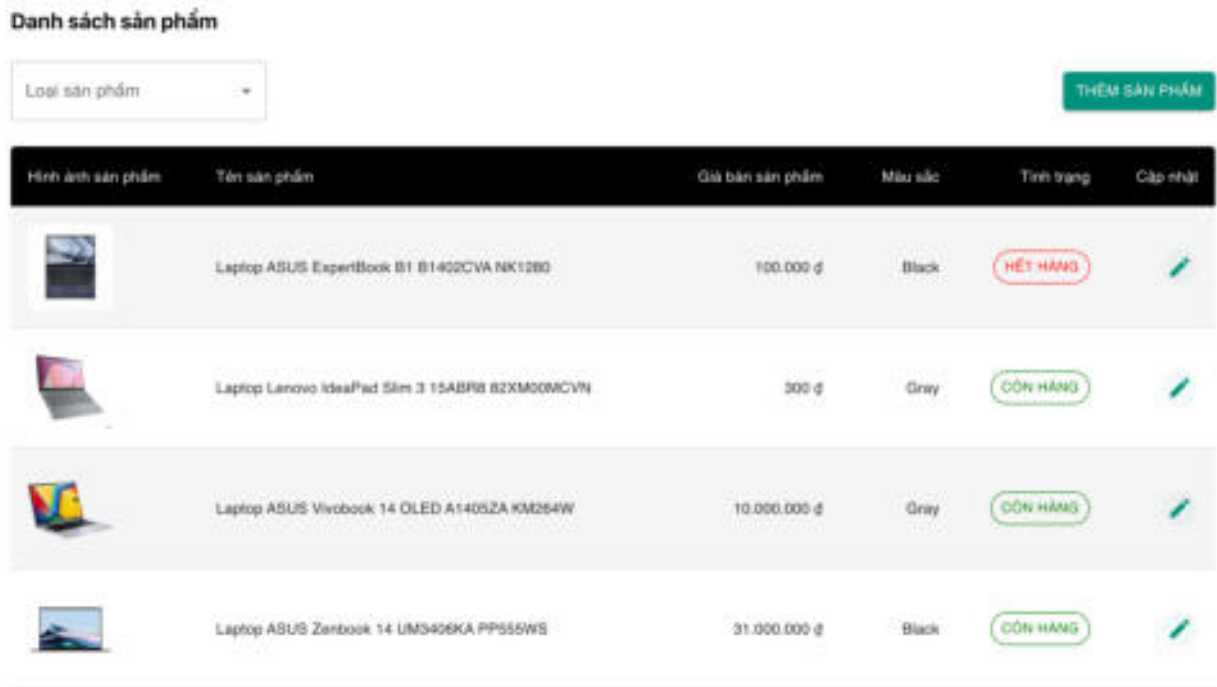


Hình 4.17 Trang danh sách các đơn hàng

Bảng 4.17 Đặc tả giao diện danh sách đơn hàng

NO	Component Name	Required	Format	Reference	Note
1	Bảng dữ liệu	Có	Table	Hiện thị thông tin các đơn hàng	
2	Product	Có	Text	Hiện thị các thông tin của sản phẩm	
3	Shipping address	Có	Text	Hiện thị thông tin địa chỉ	
4	Status	Có	Text	Hiện thị trạng thái đơn hàng	
5	Cập nhật trạng thái đơn hàng	Có	Button	Cập nhật trạng thái đơn hàng	Hiện thị thông báo
6	Phân trang	Có	Pagination	Dùng để phân trang sản phẩm	

4.2.18 Trang danh sách sản phẩm



Hình 4.18 Trang danh sách sản phẩm

Bảng 4.18 Đặc tả giao diện quản lý sản phẩm

NO	Component Name	Required	Format	Reference	Note
1	Bảng dữ liệu	Có	Table	Hiển thị danh sách sản phẩm	
2	Hình ảnh	Có	Image	Hiển thị hình ảnh của sản phẩm	
3	Tên sản phẩm	Có	Text	Hiển thị tên của sản phẩm	
4	Giá sản phẩm	Có	Text	Hiển thị giá bán của sản phẩm	
5	Màu sắc	Có	Text	Màu sắc của sản phẩm	
6	Số lượng trong kho	Có	Text	Cho biết sản phẩm còn hay hết trong kho	
7	Thêm sản phẩm	Có	Button	Chuyển đến trang thêm sản phẩm mới	

8	Cập nhật sản phẩm	Có	Button	Chuyển đến trang cập nhật sản phẩm	
9	Loại sản phẩm	Có	Select	Dùng để lọc sản phẩm theo danh mục	
10	Phân trang	Có	Pagination	Dùng để phân trang	Mỗi trang chứa tối đa 8 sản phẩm

4.2.19 Trang thêm sản phẩm mới

Hình 4.19 Trang thêm sản phẩm mới

Bảng 4.19 Đặc tả giao diện thêm sản phẩm mới

NO	Component Name	Required	Format	Reference	Note
1	Ảnh	Có	Image	Tải lên hình ảnh cho sản phẩm	Có thể tải nhiều ảnh
2	Tên sản phẩm	Có	Input	Nhập vào tên sản phẩm	

Xây dựng website bán hàng theo kiến trúc Microservices

3	Nhãn hàng	Có	Select	Chọn nhãn hàng cho sản phẩm	
4	Giá sản phẩm	Có	Input	Nhập vào giá bán của sản phẩm	
5	Số lượng	Có	Input	Nhập vào số lượng sản phẩm có trong kho	
6	Loại sản phẩm	Có	Select	Chọn danh mục cho sản phẩm	
7	Màu sản phẩm	Có	Select	Chọn màu sắc cho sản phẩm	
8	Mô tả sản phẩm	Có	Textarea	Nhập vào mô tả cho sản phẩm	
9	Thêm sản phẩm	Có	Button	Gửi thông tin sản phẩm lên server để thêm sản phẩm mới	Hiện thị thông báo

4.2.20 Trang cập nhật sản phẩm

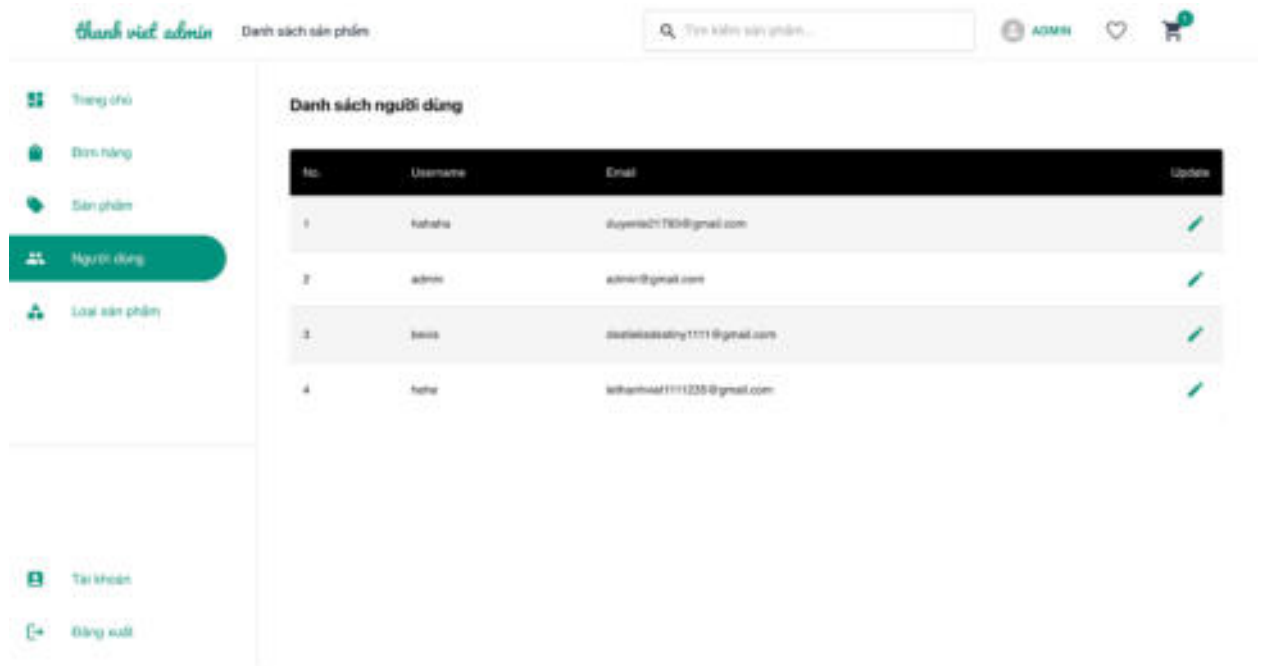
The screenshot shows a web form for updating a product. At the top, there are three image thumbnails: one with a plus sign and two with red 'X' marks. Below the images is a 'Title' field containing 'Laptop Lenovo IdeaPad Slim 3 15ABFB 82XM00MCVN'. Below the title are four input fields: 'Selling Price' with the value '300', 'Quantity' with the value '1', 'Color' with a dropdown menu showing 'Gray', and 'Category' with a dropdown menu showing 'Laptop'. Below these fields is a large 'Description' text area containing the text 'aaa'. At the bottom of the form is a prominent green button with the text 'CẬP NHẬT SẢN PHẨM'.

Hình 4.20 Trang cập nhật thông tin của sản phẩm

Bảng 4.20 Đặc tả giao diện cập nhật thông tin sản phẩm

NO	Component Name	Required	Format	Reference	Note
1	Ảnh	Có	Image List	Hiển thị tất cả hình ảnh của sản phẩm	
2	Tên sản phẩm	Có	Input	Hiển thị tên sản phẩm	
3	Nhãn hàng	Có	Select	Hiển thị nhãn hàng của sản phẩm	
4	Giá sản phẩm	Có	Input	Hiển thị giá bán của sản phẩm	
5	Số lượng	Có	Input	Hiển thị số lượng sản phẩm còn trong kho	
6	Loại sản phẩm	Có	Select	Hiển thị danh mục của sản phẩm	
7	Màu sản phẩm	Có	Select	Hiển thị màu sắc của sản phẩm	
8	Mô tả sản phẩm	Có	Textarea	Hiển thị mô tả của sản phẩm	
9	Cập nhật sản phẩm	Có	Button	Gửi thông tin sản phẩm lên server để cập nhật thông tin sản phẩm	Hiển thị thông báo

4.2.21 Trang danh sách người dùng

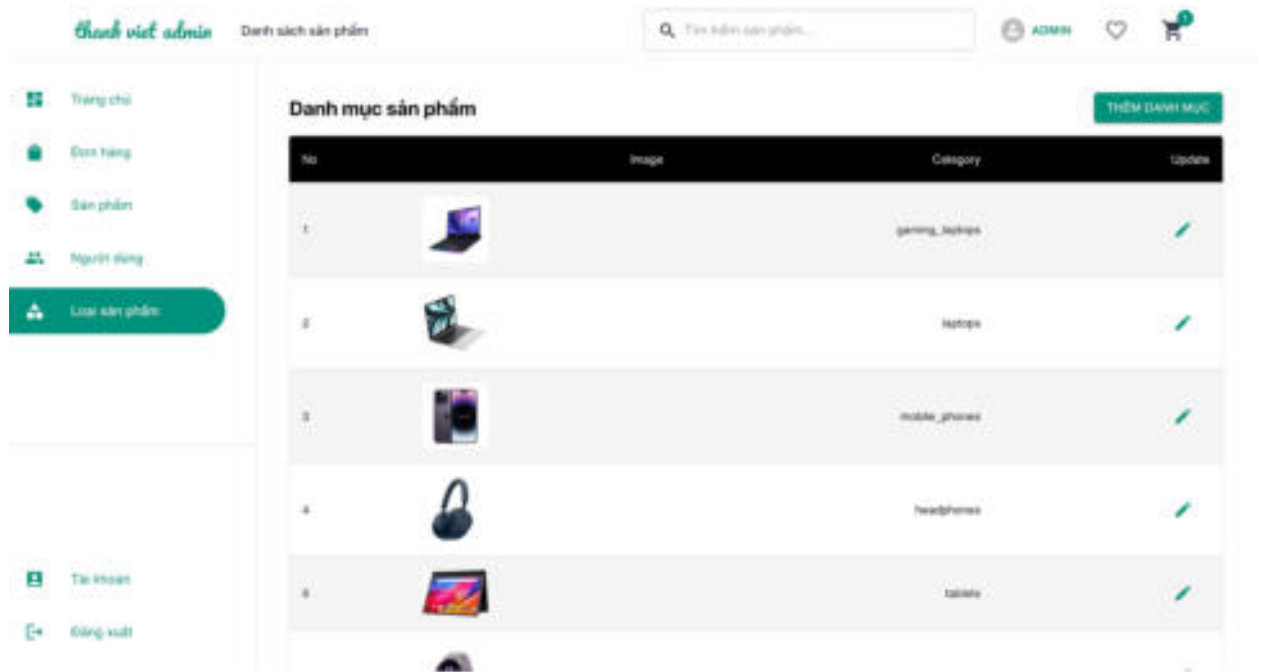


Hình 4.21 Trang danh sách người dùng

Bảng 4.21 Đặc tả giao diện danh sách người dùng

NO	Component Name	Required	Format	Reference	Note
1	Bảng dữ liệu	Có	Table	Hiển thị danh sách của người dùng	
2	Username	Có	Text	Hiển thị tên của người dùng	
3	Email	Có	Text	Hiển thị email của người dùng	
4	Cập nhật	Có	Button	Cập nhật thông tin người dùng	

4.2.22 Trang danh sách danh mục

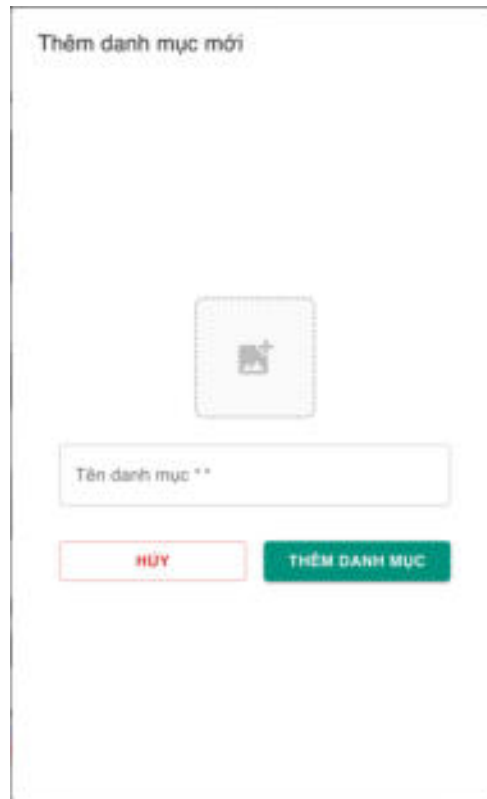


Hình 4.22 Trang danh sách danh mục

Bảng 4.22 Đặc tả giao diện danh sách danh mục

NO	Component Name	Required	Format	Reference	Note
1	Bảng dữ liệu	Có	Table	Hiển thị danh sách danh mục	
2	Hình ảnh	Có	Image	Hiển thị hình ảnh của danh mục	
3	Tên danh mục	Có	Text	Hiển thị tên của danh mục	
4	Thêm danh mục	Có	Button	Chuyển đến trang thêm danh mục mới	
5	Cập nhật danh mục	Có	Button	Chuyển đến trang cập nhật danh mục	

4.2.23 Trang thêm danh mục mới



Hình 4.23 Trang thêm danh mục mới

Bảng 4.23 Đặc tả giao diện thêm danh mục

NO	Component Name	Required	Format	Reference	Note
1	Ảnh	Có	Image	Tải lên hình ảnh cho danh mục	
2	Tên danh mục	Có	Input	Nhập vào tên danh mục	
3	Hủy	Có	Button	Trở về trang danh sách danh mục	
4	Thêm danh mục	Có	Button	Gửi thông tin sản phẩm lên server để thêm danh mục	Hiển thị thông báo

4.2.24 Trang cập nhật danh mục



Hình 4.24 Trang cập nhật thông tin danh mục

Bảng 4.24 Đặc tả giao diện cập nhật danh mục

NO	Component Name	Required	Format	Reference	Note
1	Ảnh	Có	Image	Hiển thị hình ảnh của danh mục	
2	Tên danh mục	Có	Input	Hiển thị tên của danh mục	
3	Hủy	Có	Button	Trở về trang danh sách danh mục	
4	Cập nhật	Có	Button	Gửi thông tin sản phẩm lên server để cập nhật thông tin danh mục	Hiển thị thông báo

KẾT LUẬN

1. Kết quả đạt được

- Về mặt lí thuyết:
 - + Nghiên cứu các công nghệ sử dụng trong đồ án:
 - Microservices: Là một kiến trúc phần mềm chia ứng dụng thành các dịch vụ nhỏ, độc lập, mỗi dịch vụ thực hiện một chức năng cụ thể như quản lý sản phẩm hoặc thanh toán. Điều này giúp tăng tính linh hoạt và dễ dàng bảo trì (Microservices Architecture).
 - Apache Kafka: Một nền tảng xử lý dữ liệu luồng phân tán, hỗ trợ giao tiếp bất đồng bộ giữa các dịch vụ, tối ưu hóa hiệu suất và khả năng chịu tải.
 - Docker: Công cụ đóng gói ứng dụng vào container, đảm bảo môi trường triển khai nhất quán.
 - Kubernetes: Hệ thống quản lý container, cung cấp khả năng mở rộng tự động và cân bằng tải.
 - + Nghiên cứu về một trang bán hàng trực tuyến
 - Dự án đã phân tích các nền tảng thương mại điện tử hiện có để xác định các tính năng cần thiết như quản lý sản phẩm, giỏ hàng, thanh toán, và hỗ trợ khách hàng. Điều này giúp định hình hệ thống đáp ứng nhu cầu thực tế của người dùng.
 - + Tìm hiểu kiến trúc Microservices
 - Kiến trúc Microservices được nghiên cứu kỹ lưỡng vì phù hợp với hệ thống thương mại điện tử, nơi cần xử lý nhiều chức năng độc lập. Mỗi dịch vụ có thể được phát triển và triển khai riêng biệt, giảm sự phụ thuộc và tăng khả năng mở rộng so với kiến trúc monolith truyền thống.
- Về mặt ứng dụng:
 - + Hệ thống được chia thành các microservices độc lập, bao gồm quản lý sản phẩm, quản lý người dùng, giỏ hàng, đơn hàng, thanh toán, chatbot hỗ trợ khách hàng và hệ thống gợi ý sản phẩm. Mỗi dịch vụ có thể được triển khai, nâng cấp hoặc bảo trì riêng biệt, đảm bảo tính linh hoạt và ổn định.

Service	Chức năng
Product Service	Quản lý sản phẩm
Identity Service	Quản lý người dùng
Cart Service	Quản lý giỏ hàng
Category Service	Quản lý danh mục
Payment Service	Tích hợp cổng thanh toán trực tuyến VNPAY
Order Service	Quản lý đơn hàng
Notification Service	Tiếp nhận các sự kiện từ các service khác
AI Service	Chatbot AI tự động trả lời thắc mắc của người dùng. Đề xuất các sản phẩm dựa trên lịch sử mua hàng của người dùng.

Các service của hệ thống

- + Ứng dụng Apache Kafka để xử lý các tác vụ bất đồng bộ giữa các dịch vụ, giúp tối ưu hiệu suất và tăng khả năng chịu tải của hệ thống.
- + Đóng gói các dịch vụ bằng Docker nhằm tạo ra một môi trường triển khai nhất quán, giảm thiểu sự khác biệt giữa môi trường phát triển và môi trường vận hành.
- + Triển khai hệ thống trên Kubernetes để tận dụng khả năng mở rộng tự động, cân bằng tải và quản lý tài nguyên hiệu quả.

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)
eureka	ClusterIP	None	<none>	8761/TCP
eureka-lb	NodePort	10.98.97.27	<none>	80:30243/TCP
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP
mongodb-service	ClusterIP	None	<none>	27017/TCP
mysql	ClusterIP	10.108.112.76	<none>	3306/TCP
pbl7-api-gateway-svc	LoadBalancer	10.97.113.166	<pending>	80:32294/TCP
pbl7-cart-service-svc	ClusterIP	10.107.12.170	<none>	80/TCP
pbl7-category-service-svc	ClusterIP	10.97.167.66	<none>	80/TCP
pbl7-identity-service-svc	ClusterIP	10.103.45.109	<none>	80/TCP
pbl7-inventory-service-svc	ClusterIP	10.106.20.165	<none>	80/TCP
pbl7-notification-service-svc	ClusterIP	10.105.79.202	<none>	80/TCP
pbl7-order-service-svc	ClusterIP	10.111.82.21	<none>	80/TCP
pbl7-payment-service-svc	ClusterIP	10.108.101.149	<none>	80/TCP
pbl7-product-service-svc	ClusterIP	10.102.142.116	<none>	80/TCP

Triển khai hệ thống lên Kubernetes

- + Phát triển chatbot AI hỗ trợ khách hàng với khả năng hiểu và phản hồi ngôn ngữ tự nhiên, giúp tự động hóa việc tư vấn, giải đáp thắc mắc của người dùng.
- + Tích hợp các phương thức thanh toán điện tử như VNPay nhằm mang lại sự tiện lợi và đa dạng trong quá trình thanh toán.
- Tuy nhiên, vẫn còn hạn chế được ghi nhận:
 - + Thời gian phản hồi chậm cho các truy vấn chatbot phức tạp, ảnh hưởng đến trải nghiệm người dùng trong một số trường hợp.

2. Hướng phát triển trong tương lai

- Dựa trên kết quả đạt được, các cải tiến trong tương lai có thể tập trung vào:
 - + Tối ưu hóa thời gian phản hồi của chatbot, đặc biệt cho các truy vấn phức tạp, nhằm nâng cao hiệu suất.
 - + Giải quyết các thách thức trong quản lý dữ liệu phân tán và tương tác giữa các dịch vụ để tăng cường tính ổn định.
 - + Mở rộng hệ thống với các tính năng bổ sung, như thêm chức năng đánh giá sản phẩm hoặc cải thiện hệ thống đề xuất dựa trên dữ liệu lớn.

TÀI LIỆU THAM KHẢO

- [1] Rick Merritt, *What Is Retrieval-Augmented Generation, aka RAG?*, 2025.
- [2] Ivan Belcic, *What is RAG (retrieval augmented generation)?*, 2024.
- [3] Gordon V. Cormack, Charles L. A. Clarke, Nick Craswell, *Efficient and Effective Best-Match Document Retrieval Cormack Clarke Craswell SIGIR*, 2009
- [4] James Lewis & Martin Fowler, *Microservices: a definition of this new architectural term*, 2014.
- [5] Patrick Lewis, *Retrieval-augmented generation for knowledge-intensive NLP tasks*, 2020.
- [6] Khanal, *Building Scalable Microservices with Apache Kafka and Docker Compose*, 2024.
- [7] Sonani, F. *Microservices Architecture with Docker and Kubernetes: A Complete Guide*, 2024.
- [8] Newman, *Building Microservices: Designing Fine-Grained Systems (2nd ed.)*. O'Reilly Media, 2021.
- [9] Confluent, *Spring Microservices into Production with Kubernetes and GitOps*, 2020.
- [10] Richardson, C. *Microservices Patterns: With examples in Java*. Manning Publications, 2021.
- [11] David, *Microservices-based architecture in e-commerce*, 2023.
- [12] Tatva Soft, *E-commerce Microservices Architecture Explained*.
- [13] Pavlo Okhrem, *Ecommerce Microservices Architecture: Building a Flexible and Scalable Platform*, 2024.
- [14] Chroma Documentation. Chroma: the open-source embedding database.
- [15] Cormack, G. V., Clarke, C. L. A., & Buettcher, *Reciprocal Rank Fusion Outperforms Condorcet and Individual Rank Learning Methods*. In SIGIR, 2009.
- [16] <https://www.sbert.net>.
- [17] <https://huggingface.co/sentence-transformers>.
- [18] <https://www.geeksforgeeks.org/nlp/sentence-transformer/>.
- [19] What is Retrieval-Augmented Generation RAG AWS