

BỘ KHOA HỌC VÀ CÔNG NGHỆ
HỌC VIỆN CÔNG NGHỆ BUUTURE CHÍNH VIỄN THÔNG

Khoa Công Nghệ thông Tin 2



BÁO CÁO ĐỒ ÁN

Môn: Phát triển các hệ thống thông minh

Đề tài: Xây dựng hệ thống bán giày, tích hợp chatbox

Người hướng dẫn : Nguyễn Ngọc Duy

Sinh viên thực hiện : Lưu Hoài Nhân – N21DCCN157

Lớp : D21CQCNPM02-N

Khóa : 2021 - 2026

TP.HCM, tháng 10/2025

BỘ KHOA HỌC VÀ CÔNG NGHỆ
HỌC VIỆN CÔNG NGHỆ BUUTURE CHÍNH VIỄN THÔNG

Khoa Công Nghệ thông Tin 2



BÁO CÁO ĐỒ ÁN

Môn: Phát triển các hệ thống thông minh

Đề tài: Xây dựng hệ thống bán giày, tích hợp chatbox

Người hướng dẫn : Nguyễn Ngọc Duy

Sinh viên thực hiện : Lưu Hoài Nhân – N21DCCN157

Lớp : D21CQCNPM02-N

Khóa : 2021 - 2026

TP.HCM, tháng 10/2025

Mục lục

LỜI MỞ ĐẦU	4
LỜI CẢM ƠN.....	5
DANH SÁCH HÌNH.....	6
DANH SÁCH BẢNG.....	9
CHƯƠNG 1: TỔNG QUAN	10
1.1. Giới thiệu đề tài.....	10
1.1.1. Lý do chọn đề tài	10
1.1.2. Mục tiêu đề tài	10
1.1.3. Đối tượng và phạm vi nghiên cứu	11
1.2. Cơ sở lý thuyết	12
1.2.1. Công nghệ sử dụng	12
1.2.2. Mô hình MVC.....	15
1.2.3. Hệ quản trị cơ sở dữ liệu SQL Server.....	17
1.3. Thiết kế thông minh	19
1.3.1. Chatbot AI tích hợp xử lý ngôn ngữ tự nhiên (NLP).....	19
1.3.2. Hệ thống gợi ý sản phẩm thông minh (Recommendation System)	20
1.3.3. Tìm kiếm thông minh (Intelligent Search)	20
1.3.4. Kiến trúc web thông minh	21
1.4. Kiến trúc hệ thống Client-Server	21
CHƯƠNG 2: Phân tích và thiết kế hệ thống	23
2.1. Mô tả bài toán.....	23
2.2. Yêu cầu xây dựng hệ thống.....	24
2.3. Phân tích thiết kế.....	26
2.3.1. Đối với khách hàng (user).....	26
2.3.2. Đối với quản trị viên (admin)	40
CHƯƠNG 3: Xây dựng hệ thống	61
3.1. Xây dựng cơ sở dữ liệu	61
3.1.1. Các lớp và mô tả hệ thống	61
3.1.2. Sơ đồ ERD	68

3.2. Xây dựng giao diện và chức năng hệ thống	68
3.2.1. Giao diện người dùng	68
3.2.2. Giao diện quản trị viên.....	68
TÀI LIỆU THAM KHẢO	69

LỜI MỞ ĐẦU

Trong bối cảnh thương mại điện tử đang phát triển mạnh mẽ tại Việt Nam, việc xây dựng một hệ thống bán hàng trực tuyến hiệu quả là nhu cầu cấp thiết của nhiều doanh nghiệp. Đặc biệt, ngành công nghiệp giày dép với thị trường rộng lớn và đa dạng về sản phẩm, đòi hỏi những giải pháp công nghệ thông minh để nâng cao trải nghiệm khách hàng.

Đồ án "Xây dựng hệ thống bán giày tích hợp chatbot" được thực hiện nhằm ứng dụng các kiến thức đã học về công nghệ thông tin vào việc phát triển một hệ thống thương mại điện tử hoàn chỉnh. Hệ thống không chỉ cung cấp các chức năng cơ bản của một website bán hàng mà còn tích hợp chatbot AI thông minh, hỗ trợ khách hàng trong việc tìm kiếm và lựa chọn sản phẩm phù hợp.

Báo cáo này trình bày chi tiết quá trình phân tích, thiết kế và xây dựng hệ thống, từ việc nghiên cứu cơ sở lý thuyết, phân tích yêu cầu, thiết kế cơ sở dữ liệu đến việc hiện thực hóa các chức năng của hệ thống. Thông qua đồ án này, em mong muốn góp phần ứng dụng công nghệ thông tin vào thực tiễn kinh doanh, đồng thời nâng cao kiến thức và kỹ năng chuyên môn trong lĩnh vực phát triển phần mềm.

LỜI CẢM ƠN

Em xin bày tỏ lòng biết ơn sâu sắc đến:

Thầy “Nguyễn Ngọc Duy” - giảng viên môn “Phát triển các hệ thống thông minh”, đã tận tình chỉ bảo, định hướng và đưa ra những góp ý quý báu trong suốt quá trình thực hiện đồ án. Những kiến thức chuyên môn và kinh nghiệm thực tiễn của thầy đã giúp em hoàn thành đồ án một cách tốt nhất.

Ban lãnh đạo và các thầy cô trong Khoa Công nghệ Thông tin, Học viện Công nghệ Bưu chính Viễn thông cơ sở thành phố Hồ Chí Minh đã tạo điều kiện thuận lợi về cơ sở vật chất, tài liệu tham khảo và môi trường học tập để em có thể nghiên cứu và thực hiện đồ án.

Gia đình và bạn bè đã luôn động viên, hỗ trợ tinh thần trong suốt quá trình học tập và thực hiện đồ án.

Mặc dù đã rất cố gắng, nhưng do hạn chế về thời gian và kinh nghiệm, đồ án không tránh khỏi những thiếu sót. Em rất mong nhận được sự góp ý, chỉ bảo của các thầy cô và bạn đọc để đồ án được hoàn thiện hơn.

Xin chân thành cảm ơn!

DANH SÁCH HÌNH

Hình 2.1: Use Case tổng quan cho User.....	27
Hình 2.2: Use Case chi tiết - Đăng ký tài khoản	27
Hình 2.3: Use Case chi tiết - Xem và tìm kiếm sản phẩm.....	27
Hình 2.4: Use Case chi tiết - Quản lý giỏ hàng	28
Hình 2.5: Use Case chi tiết - Đặt hàng	28
Hình 2.6: Use Case chi tiết - Quản lý tài khoản cá nhân.....	28
Hình 2.7: Use Case chi tiết - Tương tác với Chatbot	29
Hình 2.8: Use Case chi tiết - Đánh giá sản phẩm	29
Hình 2.9: Use Case với các ràng buộc và điều kiện	29
Hình 2.10: Biểu đồ hoạt động - Đăng ký tài khoản mới	30
Hình 2.11: Biểu đồ hoạt động - Xem danh sách sản phẩm	30
Hình 2.12: Biểu đồ hoạt động - Tìm kiếm sản phẩm	30
Hình 2.13: Biểu đồ hoạt động - Xem đánh giá sản phẩm.....	31
Hình 2.14: Biểu đồ hoạt động - Thêm/xóa sản phẩm giỏ hàng	31
Hình 2.15: Biểu đồ hoạt động - Đặt hàng.....	31
Hình 2.16: Biểu đồ hoạt động - Quản lý thông tin cá nhân.....	32
Hình 2.17: Biểu đồ hoạt động - Xem lịch sử mua hàng	32
Hình 2.18: Biểu đồ hoạt động - Đánh giá sản phẩm đã mua.....	32
Hình 2.19: Biểu đồ hoạt động - Chat với Chatbot.....	33
Hình 2.20: Biểu đồ trình tự - Đăng ký tài khoản mới	33
Hình 2.21: Biểu đồ trình tự - Xem danh sách sản phẩm	34
Hình 2.22: Biểu đồ trình tự - Tìm kiếm sản phẩm	34
Hình 2.23: Biểu đồ trình tự - Thêm sản phẩm vào giỏ hàng	35
Hình 2.24: Biểu đồ trình tự - Đặt hàng.....	36
Hình 2.25: Biểu đồ trình tự - Xem lịch sử mua hàng	37
Hình 2.26: Biểu đồ trình tự - Đánh giá sản phẩm	38
Hình 2.27: Biểu đồ trình tự - Chat với Chatbot.....	39
Hình 2.28: Biểu đồ trình tự - Quản lý thông tin cá nhân.....	40
Hình 2.29: Use Case tổng quan - Hệ thống Admin.....	43

Hình 2.30: Use Case chi tiết - Quản lý sản phẩm (Super Admin).....	43
Hình 2.31: Use Case chi tiết - Quản lý đơn hàng (Super Admin).....	44
Hình 2.32: Use Case chi tiết - Quản lý khách hàng (Super Admin)	44
Hình 2.33: Use Case chi tiết - Quản lý nhân viên (Super Admin)	45
Hình 2.34: Use Case chi tiết - Báo cáo và thống kê (Super Admin).....	45
Hình 2.35: Use Case chi tiết - Xử lý đơn hàng (Staff)	46
Hình 2.36: Use Case - Workflow quản lý đơn hàng hoàn chỉnh.....	46
Hình 2.38: Use Case - Tương tác giữa các Actor.....	47
Hình 2.39: Biểu đồ hoạt động - Đăng nhập Admin.....	47
Hình 2.40: Biểu đồ hoạt động - Thêm sản phẩm.....	47
Hình 2.41: Biểu đồ hoạt động - Sửa sản phẩm.....	48
Hình 2.42: Biểu đồ hoạt động - Quản lý kho	48
Hình 2.43: Biểu đồ hoạt động - Xử lý đơn hàng	48
Hình 2.44: Biểu đồ hoạt động - Quản lý khách hàng	49
Hình 2.45: Biểu đồ hoạt động - Quản lý nhân viên.....	49
Hình 2.46: Biểu đồ hoạt động - Xem báo cáo	49
Hình 2.47: Biểu đồ hoạt động - Xem đơn hàng (Staff)	50
Hình 2.48: Biểu đồ hoạt động - Xử lý đơn hàng (Staff).....	50
Hình 2.49: Biểu đồ hoạt động - Cập nhật trạng thái.....	50
Hình 2.50: Biểu đồ hoạt động - Hủy đơn hàng	51
Hình 2.51: Biểu đồ trình tự - Đăng nhập hệ thống Admin.....	51
Hình 2.52: Biểu đồ trình tự - Thêm sản phẩm mới	52
Hình 2.53: Biểu đồ trình tự - Sửa thông tin sản phẩm.....	53
Hình 2.54: Biểu đồ trình tự - Quản lý kho hàng.....	54
Hình 2.55: Biểu đồ trình tự - Xử lý đơn hàng (Super Admin)	55
Hình 2.56: Biểu đồ trình tự - Quản lý khách hàng	56
Hình 2.57: Biểu đồ trình tự - Quản lý nhân viên.....	57
Hình 2.58: Biểu đồ trình tự - Xem báo cáo thống kê	58
Hình 2.59: Biểu đồ trình tự - Xem đơn hàng được phân công (Staff).....	59
Hình 2.60: Biểu đồ trình tự - Xử lý đơn hàng (Staff).....	60

Hình 2.61: Biểu đồ trình tự - Cập nhật trạng thái đơn hàng (Staff)60

Hình 3.1: Sơ đồ ERD.....68

DANH SÁCH BẢNG

Bảng 2.1: Bảng phân quyền.....	42
Bảng 3.1: Lớp User (Người dùng cơ bản).....	62
Bảng 3.2: Lớp Customer (Khách hàng).....	62
Bảng 3.3: Lớp Admin (Quản trị viên)	63
Bảng 3.4: Lớp Staff (Nhân viên)	63
Bảng 3.5: Lớp Product (Sản phẩm)	64
Bảng 3.6: Lớp Order (Đơn hàng)	65
Bảng 3.7: Lớp Cart (Giỏ hàng).....	66
Bảng 3.8: Lớp ChatSession (Phiên chat).....	67

CHƯƠNG 1: TỔNG QUAN

1.1. Giới thiệu đề tài

1.1.1. Lý do chọn đề tài

Trong thời đại công nghệ số 4.0, thương mại điện tử đã trở thành xu hướng tất yếu và phát triển mạnh mẽ trên toàn thế giới. Đặc biệt, ngành công nghiệp thời trang và phụ kiện, trong đó có giày dép, chiếm một phần đáng kể trong tổng doanh thu thương mại điện tử. Việc lựa chọn đề tài "Xây dựng hệ thống bán giày tích hợp chatbot" xuất phát từ những lý do sau:

- Thứ nhất, nhu cầu thực tiễn trong kinh doanh: Thị trường giày dép tại Việt Nam có quy mô lớn với nhiều thương hiệu đa dạng từ Nike, Adidas đến các thương hiệu địa phương. Việc xây dựng một hệ thống bán hàng trực tuyến chuyên về giày sẽ giúp doanh nghiệp tiếp cận khách hàng rộng rãi hơn và tăng doanh thu.
- Thứ hai, xu hướng ứng dụng AI trong thương mại điện tử: Chatbot AI đang được ứng dụng rộng rãi trong các hệ thống bán hàng trực tuyến để cải thiện trải nghiệm khách hàng. Việc tích hợp chatbot không chỉ giúp tự động hóa quy trình tư vấn mà còn có thể hoạt động 24/7, giảm chi phí nhân sự cho doanh nghiệp.
- Thứ ba, cơ hội ứng dụng kiến thức đã học: Đề tài cho phép tác giả ứng dụng các kiến thức về phát triển web (NodeJS, ReactJS), cơ sở dữ liệu (SQL Server), và trí tuệ nhân tạo (NLP, Machine Learning) vào một dự án thực tế.
- Thứ tư, tính khả thi và phù hợp với thời gian thực hiện: Đề tài có phạm vi rõ ràng, tập trung vào một ngành hàng cụ thể (giày), giúp việc phân tích và thiết kế hệ thống được chi tiết và hiệu quả hơn.

1.1.2. Mục tiêu đề tài

Đề tài nhằm đạt được các mục tiêu sau:

Mục tiêu tổng quát: Xây dựng một hệ thống thương mại điện tử hoàn chỉnh chuyên về bán giày, tích hợp chatbot AI để nâng cao trải nghiệm người dùng và hỗ trợ quá trình mua sắm trực tuyến.

Mục tiêu cụ thể:

1. Về chức năng hệ thống:

- Xây dựng hệ thống quản lý sản phẩm giày với đầy đủ thông tin: tên sản phẩm, thương hiệu, màu sắc, kích thước, giá cả, hình ảnh và mô tả chi tiết.
- Phát triển chức năng quản lý người dùng với hai vai trò: khách hàng (user) và quản trị viên (admin).
- Xây dựng hệ thống gio hàng và thanh toán trực tuyến.
- Tạo giao diện quản trị để admin có thể quản lý sản phẩm, đơn hàng và khách hàng.

2. Về chatbot AI:

- Phát triển chatbot có khả năng hiểu ngôn ngữ tự nhiên (NLP) để tương tác với khách hàng.

- Xây dựng chức năng tư vấn sản phẩm dựa trên nhu cầu của khách hàng (kích cỡ, giới tính, mục đích sử dụng).
- Tích hợp chức năng gợi ý sản phẩm thông minh dựa trên giá tiền, thương hiệu và kiểu dáng.
- Phát triển tính năng tìm kiếm nhanh thông qua chatbot.

3. Về kỹ thuật:

- Ứng dụng kiến trúc MVC để tổ chức code một cách khoa học và dễ bảo trì.
- Sử dụng NodeJS để xây dựng backend API hiệu quả.
- Phát triển frontend với ReactJS để tạo giao diện người dùng thân thiện.
- Thiết kế cơ sở dữ liệu SQL Server tối ưu cho hệ thống.

4. Về nghiên cứu và học tập:

- Nghiên cứu và ứng dụng các công nghệ mới trong phát triển web.
- Tìm hiểu về xử lý ngôn ngữ tự nhiên và machine learning trong thương mại điện tử.
- Rèn luyện kỹ năng phân tích, thiết kế và triển khai hệ thống thông tin.

1.1.3. Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu:

1. Đối tượng người dùng:

- Khách hàng (User): Người tiêu dùng có nhu cầu mua giày trực tuyến, bao gồm:
 - o Nam và nữ ở các độ tuổi khác nhau
 - o Có nhu cầu mua giày cho các mục đích: chạy bộ, đi học, đi làm, chơi thể thao, dạo phố
 - o Ưa thích mua sắm trực tuyến và sử dụng công nghệ
- Quản trị viên (Admin): Người quản lý hệ thống, bao gồm:
 - o Nhân viên bán hàng
 - o Quản lý cửa hàng
 - o Nhân viên kỹ thuật

2. Đối tượng kỹ thuật:

- Hệ thống thương mại điện tử web-based
- Chatbot AI tích hợp xử lý ngôn ngữ tự nhiên
- Cơ sở dữ liệu quản lý thông tin sản phẩm và giao dịch
- API RESTful cho việc trao đổi dữ liệu

Phạm vi nghiên cứu:

1. Về sản phẩm:

- Chỉ tập trung vào giày: sneaker, giày thể thao, giày boot, sandal
- Thông tin sản phẩm: tên, thương hiệu (Nike, Adidas, Vans, v.v.), màu sắc, kích thước (từ 35-45), giá cả, hình ảnh, mô tả

- Không bao gồm: quần áo, phụ kiện khác ngoài giày

2. Về chức năng:

- Chức năng cơ bản: đăng ký, đăng nhập, xem sản phẩm, tìm kiếm, giỏ hàng, thanh toán
- Chức năng quản trị: quản lý sản phẩm, đơn hàng, khách hàng
- Chức năng chatbot: tư vấn sản phẩm, gợi ý, tìm kiếm bằng ngôn ngữ tự nhiên

3. Về giới hạn:

- Hệ thống được phát triển ở mức độ demo/prototype
- Chatbot sử dụng rule-based và NLP cơ bản, chưa áp dụng deep learning phức tạp
- Chức năng thanh toán mô phỏng, chưa tích hợp với cổng thanh toán thực tế
- Hỗ trợ tiếng Việt chủ yếu, có thể bổ sung tiếng Anh cơ bản
- Số lượng sản phẩm demo khoảng 50-100 sản phẩm để test hệ thống

1.2. Cơ sở lý thuyết

1.2.1. Công nghệ sử dụng

1.2.1.a. NodeJS

1. Giới thiệu về NodeJS



NodeJS là một nền tảng được xây dựng dựa trên thời gian chạy JavaScript của Chrome để dễ dàng xây dựng các ứng dụng mạng nhanh và có thể mở rộng. NodeJS sử dụng kiến trúc hướng sự kiện event-driven, không chặn (non-blocking) làm cho nó nhẹ nhàng hơn, hoàn hảo cho các ứng dụng real-time sử dụng nhiều dữ liệu chạy trên các thiết bị phân tán.

NodeJS là môi trường chạy mã JavaScript phía máy chủ. Nó được xây dựng dựa trên JavaScript engine V8 của Google Chrome, cho phép chạy mã JavaScript nhanh và hiệu quả trên máy chủ.

Với NodeJS bạn có thể xây dựng ứng dụng web và backend bằng JavaScript, một ngôn ngữ phổ biến và linh hoạt. NodeJS cho phép bạn xử lý các ứng dụng khách yêu cầu và tạo API, kết nối với cơ sở dữ liệu, thao tác với các tệp và thực hiện các tác vụ máy chủ khác một cách đơn giản và hiệu quả.

NodeJS cũng chú trọng đến việc xử lý không đồng bộ (bất đồng bộ), cho phép ứng dụng xử lý nhiều yêu cầu cùng một lúc và không bị chặn bởi các tác vụ đang chờ đợi. Điều này làm

cho NodeJS trở thành một lựa chọn phù hợp cho việc xây dựng ứng dụng chạy thời gian thực và có khả năng xử lý tải cao.

NodeJS cũng đi kèm với module thư viện phong phú, cho phép lập trình viên sử dụng các chức năng có sẵn và không cần viết lại từ đầu. Nó còn có một cộng đồng rộng lớn và hỗ trợ mạnh mẽ từ cộng đồng lập trình viên, chắc chắn rằng bạn có thể tìm thấy các tài liệu hướng dẫn và các module hữu ích để phát triển ứng dụng NodeJS của riêng bạn.

2. Lợi ích của NodeJS

NodeJS có nhiều ưu điểm khi sử dụng trong việc phát triển ứng dụng web và back-end. Dưới đây là một số điểm ưu tiên của NodeJS:

- Hiệu suất cao: NodeJS được xây dựng trên JavaScript engine V8, đảm bảo hiệu suất cao với việc xử lý mã JavaScript nhanh chóng. Nó cũng hỗ trợ xử lý không đồng bộ, cho phép xử lý nhiều yêu cầu cùng một lúc mà không gây gián đoạn.
- Xử lý không đồng bộ: NodeJS được xây dựng với kiến trúc không đồng bộ, cho phép xử lý nhiều yêu cầu cùng một lúc mà không bị chặn bởi các nhiệm vụ chờ đợi. Điều này làm cho ứng dụng trở nên rất nhạy bén và có thể đáp ứng tải cao.
- Sử dụng cùng một ngôn ngữ: Với NodeJS, bạn có thể sử dụng cùng một ngôn ngữ JavaScript cho phía máy chủ và phía khách hàng. Điều này giúp giảm mức độ phức tạp trong quá trình phát triển và duy trì ứng dụng.
- Hệ sinh thái module phong phú: NodeJS có một module hệ sinh thái phong phú, với hàng ngàn module có sẵn để tái sử dụng. Bạn có thể tìm thấy các module hữu ích trong cộng đồng NodeJS và sử dụng chúng để giảm thời gian phát triển và tăng tính linh hoạt của ứng dụng.
- Cộng đồng lập trình viên mạnh mẽ: NodeJS có một cộng đồng lập trình thành viên lớn và nổi. Điều này đảm bảo rằng bạn có thể tìm thấy các tài nguyên, hướng dẫn và hỗ trợ từ những người khác trong cộng đồng để giải quyết vấn đề và nâng cao kỹ năng lập trình của mình.
- Thích hợp cho ứng dụng thời gian thực: Với việc hỗ trợ xử lý không đồng bộ và hiệu suất cao. NodeJS là một lựa chọn phù hợp cho việc xây dựng ứng dụng thời gian thực như ứng dụng trò chuyện trực tuyến, máy chủ trò chơi và ứng dụng theo thời gian thực khác.

3. Nhược điểm của NodeJS

Mặc dù NodeJS có nhiều ưu điểm, nhưng cũng có một số nhược điểm bạn cần lưu ý:

- Đơn luồng: NodeJS dựa trên mô hình đơn luồng, điều này có nghĩa là có một luồng xử lý trong NodeJS. Điều này có thể gây ra vấn đề khi có các tác vụ tốn nhiều thời gian xử lý, làm cho ứng dụng hoạt động chậm.
- Khả năng xử lý CPU hạn chế: NodeJS phù hợp cho việc xử lý các tác vụ I/O (ví dụ: yêu cầu mạng, truy vấn cơ sở dữ liệu), nhưng không phải là lựa chọn tốt cho các tác

vụ tính toán mất nhiều thời gian. Do đó, nếu ứng dụng của bạn có yêu cầu xử lý CPU nặng, NodeJS có thể không phải là lựa chọn tốt.

- Thư viện thứ ba không ổn định: Mặc dù NodeJS có một hệ sinh thái module phong phú, nhưng không phải tất cả các module bên thứ ba đều đảm bảo ổn định và chất lượng. Có thể gặp phải các vấn đề như sự không tương thích, lỗi hoặc sự thiếu hỗ trợ từ các module bên thứ ba.
- Quản lý phiên bản module: Khi sử dụng NodeJS, việc quản lý phiên bản module có thể trở nên phức tạp. Vì NodeJS sử dụng mô hình xử lý module phiên bản, việc kiểm soát và đồng bộ các phiên bản module có thể trở nên rắc rối, đặc biệt khi có nhiều module sử dụng các phiên bản khác nhau.

Tính mở rộng: là một thách thức khi sử dụng NodeJS trong một môi trường có tải cao hoặc cần xây dựng ứng dụng lớn. Điều này đòi hỏi việc kiến trúc ứng dụng và các giải pháp phân tán được quản lý cẩn thận để đảm bảo tính mở rộng và hiệu suất ổn định

1.2.1.b. ReactJS

1. Giới thiệu về ReactJS



ReactJS là một thư viện JavaScript phổ biến và mạnh mẽ được sử dụng để xây dựng giao diện người dùng (UI) cho các ứng dụng web hiện đại. ReactJS được phát triển bởi Facebook và đã nhanh chóng trở thành một công cụ phổ biến trong cộng đồng phát triển web.

ReactJS sử dụng một khái niệm gọi là "component" để xây dựng giao diện người dùng. Mỗi component đại diện cho một phần của UI và chứa các logic, phương thức và trạng thái riêng. Các component này có thể được sắp xếp, kết hợp và tái sử dụng để tạo nên giao diện người dùng phức tạp.

Một trong những ưu điểm quan trọng của ReactJS là Virtual DOM (Document Object Model ảo). ReactJS sử dụng Virtual DOM để giảm thiểu việc tương tác trực tiếp với DOM thực tế, điều này giúp cải thiện hiệu suất và tốc độ của ứng dụng. ReactJS sẽ cập nhật và tái render chỉ những thành phần cần thiết khi trạng thái của ứng dụng thay đổi, giúp tiết kiệm tài nguyên và tăng tốc độ render.

ReactJS cũng hỗ trợ mô hình dữ liệu một chiều (one-way data flow), trong đó dữ liệu di chuyển theo một hướng duy nhất từ component cha đến các component con. Điều này giúp dễ

dàng quản lý và dự đoán luồng dữ liệu trong ứng dụng và làm cho quá trình phát triển và bảo trì dễ dàng hơn.

2. Ưu điểm của ReactJs

- Hiệu suất cao: ReactJS sử dụng Virtual DOM và cơ chế tái render thông minh để tối ưu hóa hiệu suất. Nó chỉ cập nhật những phần thay đổi thực sự trên giao diện người dùng, giúp giảm tải và cải thiện tốc độ render.

- Quản lý dễ dàng: ReactJS sử dụng mô hình dữ liệu một chiều, giúp quản lý và dự đoán luồng dữ liệu trong ứng dụng một cách rõ ràng. Điều này làm cho việc phát triển và bảo trì ứng dụng dễ dàng hơn, đặc biệt khi ứng dụng có quy mô lớn.

- Tái sử dụng thành phần: ReactJS cho phép tái sử dụng các thành phần UI. Bằng cách chia UI thành các component nhỏ hơn, chúng có thể được sử dụng lại trong nhiều phần của ứng dụng, giúp tiết kiệm thời gian và công sức phát triển.

- Cộng đồng phát triển mạnh mẽ: ReactJS có cộng đồng lớn và nhiều nguồn tài nguyên hữu ích. Có hàng ngàn gói npm có sẵn để mở rộng chức năng và khả năng của ứng dụng ReactJS. Cộng đồng cũng rất năng động và hỗ trợ lẫn nhau thông qua tài liệu, diễn đàn và các dự án mã nguồn mở.

3. Nhược điểm của ReactJs

- Khả năng học ban đầu: ReactJS có một học phần khá dày đặc và đòi hỏi hiểu biết về JavaScript và các khái niệm front-end. Điều này có thể tạo ra một thời gian học ban đầu đối với những người mới bắt đầu hoặc người không quen với các công nghệ tương tự.

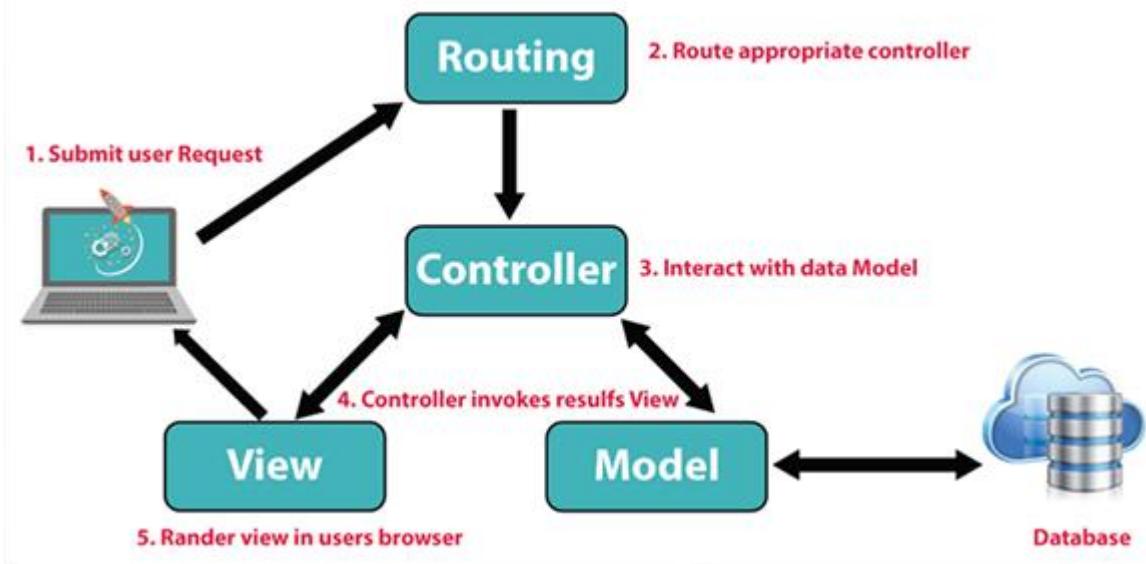
- Hạn chế của UI: ReactJS tập trung chủ yếu vào việc xây dựng giao diện người dùng (UI). Nếu ứng dụng của bạn đòi hỏi nhiều tính năng khác như định tuyến (routing) hoặc quản lý trạng thái phức tạp, bạn có thể cần phải sử dụng các thư viện hoặc khung công việc bổ sung.

- Khả năng phức tạp hóa: Khi ứng dụng ReactJS phát triển và mở rộng, có thể xảy ra tình trạng phức tạp hóa codebase. Việc quản lý các thành phần, props, state và logic có thể trở nên phức tạp và khó hiểu, đặc biệt đối với các ứng dụng lớn và phức tạp.

- Reactjs chỉ phục vụ cho tầng View: React chỉ là View Library nó không phải là một MVC framework như những framework khác. Đây chỉ là thư viện của Facebook giúp render ra phần view. Vì thế React sẽ không có phần Model và Controller, mà phải kết hợp với các thư viện khác

1.2.2. Mô hình MVC

1. Khái niệm mô hình MVC



MVC là viết tắt của cụm từ “Model-View-Controller”. Đây là mô hình thiết kế sử dụng trong kỹ thuật phần mềm. MVC là một mẫu kiến trúc phần mềm để tạo lập giao diện người dùng trên máy tính. MVC chia thành ba phần được kết nối với nhau như tên gọi: Model (dữ liệu), View (giao diện) và Controller (bộ điều khiển).

Đơn giản hơn là mô hình này được chia thành 3 phần trong source code. Và mỗi phần đảm nhận vai trò và nhiệm vụ riêng biệt nhau và độc lập.

Mô hình MVC gồm 3 loại chính là thành phần bên trong không thể thiếu khi áp dụng mô hình này:

- Model: Là bộ phận có chức năng lưu trữ toàn bộ dữ liệu của ứng dụng và là cầu nối giữa 2 thành phần bên dưới là View và Controller. Một model là dữ liệu được sử dụng bởi chương trình. Đây có thể là cơ sở dữ liệu, hoặc file XML bình thường hay một đối tượng đơn giản. Chẳng hạn như biểu tượng hay là một nhân vật trong game.
- View: Đây là phần giao diện (theme) dành cho người sử dụng. View là phương tiện hiển thị các đối tượng trong một ứng dụng. Chẳng hạn như hiển thị một cửa sổ, nút hay văn bản trong một cửa sổ khác. Nó bao gồm bất cứ thứ gì mà người dùng có thể nhìn thấy được.
- Controller: Là bộ phận có nhiệm vụ xử lý các yêu cầu người dùng đưa đến thông qua View. Một controller bao gồm cả Model lẫn View. Nó nhận input và thực hiện các update tương ứng.

2. Ưu điểm của mô hình MVC

Đầu tiên, nhắc tới ưu điểm mô hình MVC thì đó là băng thông (Bandwidth) nhẹ vì không sử dụng viewstate nên khá tiết kiệm băng thông. Việc giảm băng thông giúp website hoạt động ổn định hơn.

Kiểm tra đơn giản và dễ dàng, kiểm tra lỗi phần mềm trước khi bàn giao lại cho người dùng.

Một lợi thế chính của MVC là nó tách biệt các phần Model, Controller và View với nhau.

Sử dụng mô hình MVC chức năng Controller có vai trò quan trọng và tối ưu trên các nền tảng ngôn ngữ khác nhau

Ta có thể dễ dàng duy trì ứng dụng vì chúng được tách biệt với nhau.

Có thể chia nhiều developer làm việc cùng một lúc. Công việc của các developer sẽ không ảnh hưởng đến nhau.

Phiên bản mới nhất của MVC hỗ trợ thiết kế responsive website mặc định và các mẫu cho mobile. Chúng ta có thể tạo công cụ View của riêng mình với cú pháp đơn giản hơn nhiều so với công cụ truyền thống.

3. Nhược điểm của mô hình MVC

Bên cạnh những ưu điểm MVC mang lại thì nó cũng có một số nhược điểm cần khắc phục.

MVC đa phần phù hợp với công ty chuyên về website hoặc các dự án lớn thì mô hình này phù hợp hơn so với các dự án nhỏ, lẽ vì khá là cồng kềnh và mất thời gian.

- Không thể Preview các trang như ASP.NET.
- Khó triển khai.

1.2.3. Hệ quản trị cơ sở dữ liệu SQL Server

SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) được phát triển và thương mại hóa bởi Microsoft. Nó được thiết kế để lưu trữ, quản lý và truy xuất dữ liệu hiệu quả, đáp ứng nhu cầu xử lý dữ liệu của các ứng dụng từ nhỏ đến lớn trong doanh nghiệp. Với lịch sử phát triển từ năm 1989, SQL Server đã trải qua nhiều phiên bản cải tiến, từ SQL Server 1.0 hợp tác với Sybase cho đến các phiên bản hiện đại như SQL Server 2019, 2022 và SQL Server 2025 Preview với khả năng tích hợp AI.

Đặc điểm chính

Quan hệ và đa người dùng: SQL Server hỗ trợ lưu trữ dữ liệu dưới dạng bảng quan hệ, cho phép nhiều người dùng cùng truy cập và thao tác dữ liệu đồng thời thông qua các giao dịch tuân thủ ACID (Atomicity, Consistency, Isolation, Durability). Hệ thống sử dụng cơ chế khóa tinh vi và quản lý concurrency để đảm bảo tính toàn vẹn dữ liệu trong môi trường đa người dùng.

Ngôn ngữ truy vấn: SQL Server sử dụng Transact-SQL (T-SQL), một phần mở rộng mạnh mẽ của SQL chuẩn, cho phép viết các truy vấn phức tạp, thủ tục lưu trữ, triggers và các tập lệnh với khả năng lập trình procedural. T-SQL bao gồm các cấu trúc điều khiển, xử lý ngoại lệ, biến và tham số, tạo nên một ngôn ngữ lập trình hoàn chỉnh trong môi trường cơ sở dữ liệu.

Đa nền tảng và khả năng triển khai: Từ SQL Server 2017, Microsoft đã mở rộng khả năng triển khai sang các hệ điều hành Linux và các môi trường container Docker, không chỉ

giới hạn trên Windows Server. Điều này cho phép SQL Server tích hợp vào các kiến trúc hybrid cloud và multi-platform, đáp ứng nhu cầu đa dạng của doanh nghiệp hiện đại.

Khả năng mở rộng và bảo mật: SQL Server cung cấp các tính năng mở rộng từ phân vùng dữ liệu (partitioning), Always On Availability Groups cho high availability, đến các cơ chế bảo mật tiên tiến như Transparent Data Encryption (TDE), Always Encrypted, Row-Level Security và xác thực đa yếu tố. Hệ thống hỗ trợ mở rộng theo chiều ngang và chiều dọc, từ database nhỏ vài GB đến các hệ thống enterprise với hàng terabyte dữ liệu.

Tính năng tích hợp nâng cao

SQL Server không chỉ là một hệ quản trị cơ sở dữ liệu đơn thuần mà còn là một nền tảng dữ liệu toàn diện với nhiều dịch vụ tích hợp. Machine Learning Services cho phép tích hợp và chạy các mô hình học máy với R và Python trực tiếp trong cơ sở dữ liệu, mang lại khả năng phân tích dữ liệu in-database mạnh mẽ. Integration Services (SSIS) cung cấp nền tảng ETL (Extract, Transform, Load) cho việc xây dựng các giải pháp tích hợp dữ liệu phức tạp. Analysis Services (SSAS) hỗ trợ phân tích đa chiều OLAP và mô hình hóa dữ liệu cho business intelligence. Reporting Services (SSRS) cung cấp khả năng xây dựng, quản lý và xuất báo cáo doanh nghiệp với nhiều định dạng khác nhau.

Vai trò trong hệ thống thông tin

Trong kiến trúc hệ thống thông tin hiện đại, SQL Server đóng vai trò trung tâm như tầng dữ liệu (data tier) trong mô hình n-tier architecture. Nó không chỉ lưu trữ dữ liệu mà còn thực thi business logic thông qua stored procedures, triggers và constraints, đảm bảo tính toàn vẹn và nhất quán của dữ liệu. SQL Server được sử dụng làm backbone cho các ứng dụng doanh nghiệp như hệ thống ERP, CRM, quản lý nhân sự, e-commerce và các ứng dụng web scale-out. Khả năng tích hợp sâu với Microsoft technology stack (.NET, Azure, Office 365) cùng với khả năng cross-platform mới làm cho SQL Server trở thành lựa chọn linh hoạt cho nhiều môi trường công nghệ khác nhau.

Vai trò trong phát triển web

SQL Server đóng vai trò quan trọng như hệ quản trị cơ sở dữ liệu backend cho các ứng dụng web, từ website nhỏ đến các platform e-commerce lớn. Trong kiến trúc web development, SQL Server hoạt động như data tier, nhận và xử lý các request từ application tier thông qua các giao thức kết nối như ADO.NET, ODBC, JDBC. Hệ thống hỗ trợ connection pooling để tối ưu hóa hiệu suất, xử lý hàng nghìn concurrent connections và cung cấp các tính năng như caching, indexing strategies, và query optimization để đảm bảo response time nhanh cho web applications. SQL Server tích hợp tốt với các web development frameworks như ASP.NET, PHP, Python, Node.js thông qua các driver và ORM tools, cho phép developers xây dựng các ứng dụng web robust và scalable.

Nhờ tính ổn định, bảo mật cao, khả năng xử lý dữ liệu lớn và hệ sinh thái công cụ phong phú, SQL Server là lựa chọn phổ biến và đáng tin cậy trong nhiều tổ chức và doanh nghiệp trên toàn thế giới, từ các startup công nghệ đến các tập đoàn Fortune 500.

1.3. Thiết kế thông minh

Trong bối cảnh thương mại điện tử hiện đại, việc ứng dụng trí tuệ nhân tạo (AI) và các công nghệ thông minh đã trở thành yếu tố then chốt để nâng cao trải nghiệm khách hàng và tối ưu hóa hiệu quả kinh doanh. Hệ thống bán giày web-based được xây dựng trong đồ án này tích hợp nhiều giải pháp thiết kế thông minh nhằm tạo ra một trải nghiệm mua sắm trực tuyến độc đáo và hiệu quả.

1.3.1. Chatbot AI tích hợp xử lý ngôn ngữ tự nhiên (NLP)

Kiến trúc Chatbot thông minh:

Chatbot của hệ thống được thiết kế dựa trên kiến trúc hybrid kết hợp giữa rule-based system và natural language processing cơ bản, cho phép xử lý các truy vấn phức tạp từ người dùng một cách tự nhiên và chính xác.

Các thành phần chính:

1. Natural Language Understanding (NLU)

- Intent Recognition: Nhận diện ý định của người dùng từ câu nói tự nhiên
 - Ví dụ: "Tôi muốn mua giày chạy bộ" → Intent: find_running_shoes
 - "Có giày Nike nào dưới 2 triệu không?" → Intent: search_by_brand_price
- Entity Extraction: Trích xuất thông tin cụ thể từ câu nói
 - Thương hiệu: Nike, Adidas, Vans, Converse
 - Loại giày: chạy bộ, thể thao, sneaker, boot, sandal
 - Kích thước: 35, 36, 37... 45
 - Màu sắc: đen, trắng, đỏ, xanh, vàng
 - Giá tiền: dưới 1 triệu, từ 1-2 triệu, trên 2 triệu
 - Giới tính: nam, nữ, unisex

2. Dialog Management

- Quản lý luồng hội thoại đa bước trong giao diện web
- Ghi nhớ context của cuộc hội thoại trong session
- Xử lý các trường hợp người dùng thay đổi yêu cầu giữa chừng

Ví dụ luồng hội thoại thông minh trên web:

User: "Tôi muốn mua giày chạy bộ"

Bot: "Bạn muốn tìm giày chạy bộ cho nam hay nữ? Và bạn thường đi size bao nhiêu ạ?"

User: "Nam size 42"

Bot: "Tuyệt! Bạn có ngân sách dự kiến không? Và có thích thương hiệu nào đặc biệt không?"

User: "Khoảng 2 triệu, thích Nike"

Bot: "Tôi tìm thấy 3 mẫu giày chạy bộ Nike nam size 42 trong tầm giá 2 triệu phù hợp với bạn:

[Hiển thị carousel sản phẩm với hình ảnh và link trong chatbox]"

1.3.2. Hệ thống gợi ý sản phẩm thông minh (Recommendation System)

Thuật toán gợi ý được tối ưu cho web:

1. Content-Based Filtering

- Phân tích đặc điểm sản phẩm: thương hiệu, loại giày, màu sắc, giá cả
- Gợi ý dựa trên lịch sử duyệt web của người dùng (cookies, session storage)
- Tính điểm tương đồng giữa các sản phẩm

2. Behavioral Analysis

- Phân tích click-through rate trên từng sản phẩm
- Tracking thời gian xem sản phẩm
- Phân tích search patterns của người dùng

3. Real-time Recommendations

- Gợi ý ngay lập tức khi người dùng xem sản phẩm
- "Sản phẩm tương tự" hiển thị dynamic
- "Người khác cũng xem" dựa trên session data

1.3.3. Tìm kiếm thông minh (Intelligent Search)

Search Engine tối ưu cho web interface:

1. Smart Autocomplete

- Gợi ý từ khóa real-time khi người dùng gõ
- Hiển thị trending searches
- Lưu search history trong localStorage

2. Advanced Filtering

- Filter sidebar với multiple selections
- Price range slider dynamic
- Color picker visual interface

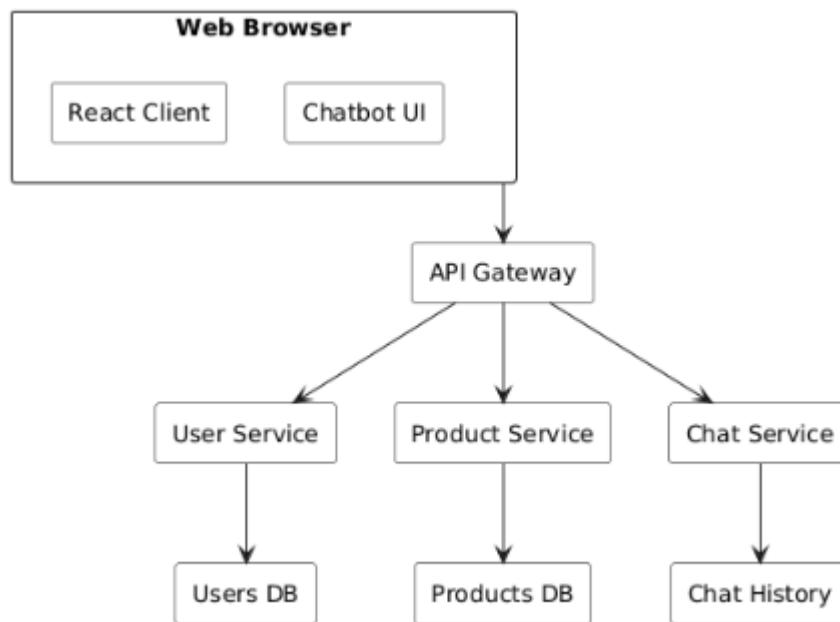
- Size selector với size chart popup

3. Semantic Search

- "giày đi mưa" → tìm giày chống nước, boot
- "giày đi học" → tìm giày thể thao, sneaker đơn giản
- "giày công sở" → tìm giày tây, giày da

1.3.4. Kiến trúc web thông minh

Web Architecture tối ưu:



Thiết kế thông minh trong hệ thống web không chỉ nâng cao trải nghiệm người dùng mà còn tối ưu hóa hiệu suất, bảo mật và khả năng mở rộng của ứng dụng web, tạo ra một nền tảng vững chắc cho việc kinh doanh trực tuyến hiệu quả.

1.4. Kiến trúc hệ thống Client-Server

1. Định nghĩa

Kiến trúc Client-Server (khách – chủ) là một mô hình kiến trúc mạng trong đó các thiết bị hoặc chương trình gọi là client (khách) gửi yêu cầu đến một hoặc nhiều server (máy chủ) để yêu cầu dịch vụ hoặc tài nguyên. Server sẽ xử lý yêu cầu và trả về kết quả cho client.

2. Các thành phần chính

- Client (Khách): Là thành phần khởi tạo yêu cầu, thường là ứng dụng giao diện người dùng, ví dụ như trình duyệt web, ứng dụng di động, phần mềm desktop. Client xử lý các tương tác người dùng và gửi yêu cầu tới server.

- Server (Máy chủ): Là thành phần cung cấp dịch vụ hoặc dữ liệu cho client. Server có thể lưu trữ dữ liệu, thực thi các xử lý nghiệp vụ và quản lý tài nguyên chung. Ví dụ: máy chủ web, máy chủ cơ sở dữ liệu.

3. Cách thức hoạt động

- Client gửi một yêu cầu (request) đến server theo một giao thức xác định (ví dụ: HTTP, FTP).
- Server nhận yêu cầu, xử lý và truy xuất dữ liệu hoặc thực hiện các chức năng nghiệp vụ.
- Server trả lại kết quả (response) cho client.
- Client nhận dữ liệu và hiển thị hoặc xử lý tiếp theo.

4. Đặc điểm của kiến trúc Client-Server

- Phân tách rõ ràng giữa client và server, mỗi bên đảm nhiệm một vai trò riêng biệt.
- Giao tiếp qua mạng: client và server thường chạy trên các máy khác nhau và giao tiếp qua mạng (LAN hoặc Internet).
- Tính tập trung: Server tập trung quản lý dữ liệu và xử lý nghiệp vụ, giúp quản lý và bảo mật dữ liệu hiệu quả.
- Mở rộng dễ dàng: Có thể bổ sung thêm nhiều client hoặc server tùy theo nhu cầu hệ thống.

5. Ưu điểm

- Tối ưu tài nguyên, phân chia tải hợp lý giữa client và server.
- Dễ dàng quản lý tập trung trên server (dữ liệu, bảo mật).
- Các client tương tác đơn giản, nhẹ nhàng.
- Phù hợp cho các hệ thống đa người dùng.

6. Nhược điểm

- Phụ thuộc vào server: nếu server bị lỗi hoặc ngưng hoạt động thì toàn bộ client không thể truy cập dịch vụ.
- Yêu cầu kết nối mạng ổn định.
- Server có thể bị quá tải nếu số lượng client quá lớn.

7. Ví dụ thực tế

- Trình duyệt web (client) gửi yêu cầu tải trang web đến web server (server).
- Ứng dụng email (client) kết nối đến mail server để gửi và nhận email.
- Ứng dụng mobile kết nối API server để lấy dữ liệu cập nhật.

CHƯƠNG 2: Phân tích và thiết kế hệ thống

2.1. Mô tả bài toán

Hệ thống website bán giày được xây dựng với mục tiêu giúp khách hàng dễ dàng tìm kiếm và lựa chọn sản phẩm giày phù hợp với nhu cầu sử dụng, tiết kiệm thời gian và mang lại trải nghiệm mua sắm thông minh hơn. Đồng thời, hệ thống cũng giúp người quản trị dễ dàng trong việc quản lý sản phẩm giày, đơn hàng cũng như các nội dung liên quan trên website. Hệ thống tập trung vào hai nhóm người dùng chính là khách hàng và quản trị viên.

Khi khách hàng truy cập vào website mà chưa đăng nhập, họ chỉ có thể xem thông tin về sản phẩm giày, tìm kiếm theo thương hiệu, loại giày hoặc các bài viết giới thiệu. Để có thể đặt hàng, khách hàng cần đăng ký và đăng nhập tài khoản. Sau khi đăng nhập, khách hàng có thể thêm sản phẩm vào giỏ hàng, chọn kích cỡ phù hợp, lựa chọn phương thức thanh toán và tiến hành đặt hàng. Ngoài ra, người dùng còn có thể theo dõi trạng thái đơn hàng, chỉnh sửa thông tin tài khoản và đánh giá sản phẩm sau khi mua.

Hệ thống có tích hợp một chatbot thông minh sử dụng xử lý ngôn ngữ tự nhiên (NLP) để hỗ trợ khách hàng trong việc tư vấn và lựa chọn giày. Chatbot sẽ hoạt động tự động 24/7, có thể trả lời các câu hỏi về sản phẩm như "Tôi muốn tìm giày chạy bộ nam size 42", "Có giày Nike nào dưới 2 triệu không?", hoặc "Giày nào phù hợp để đi làm?". Bên cạnh đó, chatbot còn hỗ trợ tư vấn chọn giày dựa trên mục đích sử dụng (chạy bộ, đi học, đi làm, chơi thể thao), giới tính, kích cỡ và ngân sách của khách hàng. Chatbot có thể gợi ý sản phẩm phù hợp và hiển thị trực tiếp hình ảnh, giá cả cùng link sản phẩm ngay trong khung chat, giúp khách hàng có thể tìm được đôi giày mong muốn nhanh chóng mà không cần phải duyệt qua hàng trăm sản phẩm.

Đặc biệt, chatbot còn được trang bị khả năng hiểu ngôn ngữ tự nhiên, có thể nhận diện các thông tin như loại sản phẩm, thương hiệu, màu sắc, kích thước từ câu nói của khách hàng. Ví dụ khi khách hàng nói "Tôi muốn mua đôi giày thể thao Adidas màu trắng size 40 cho nữ", chatbot sẽ tự động tách thông tin và đưa ra danh sách sản phẩm phù hợp. Hệ thống cũng có chức năng tìm kiếm nhanh thông qua chatbot, người dùng chỉ cần gõ "giày Nike dưới 2 triệu" và chatbot sẽ trả về danh sách phù hợp ngay lập tức.

Khi khách hàng hoàn tất đặt hàng, hệ thống sẽ thông báo cho quản trị viên. Quản trị viên sẽ xem xét và xác nhận đơn hàng, kiểm tra tình trạng kho hàng theo size và màu sắc cụ thể, đồng thời có thể thay đổi trạng thái đơn hàng tùy theo tiến độ xử lý như: chờ xác nhận, đang chuẩn bị hàng, đang giao hàng, đã hoàn tất hoặc hủy đơn. Ngoài ra, quản trị viên còn có thể quản lý danh mục sản phẩm giày (sneaker, giày thể thao, boot, sandal), quản lý thông tin chi tiết từng sản phẩm bao gồm thương hiệu, màu sắc, các size có sẵn, số lượng tồn kho, hình ảnh và mô tả sản phẩm.

Quản trị viên cũng có thể tạo và quản lý các chương trình khuyến mãi đặc biệt cho từng dòng sản phẩm như giảm giá giày thể thao, khuyến mãi combo giày + phụ kiện, hoặc ưu đãi theo thương hiệu nhằm thu hút khách hàng. Hệ thống cũng hỗ trợ quản lý kho hàng chi tiết theo từng size và màu sắc của mỗi mẫu giày, giúp admin dễ dàng cập nhật số lượng tồn kho, theo dõi sản phẩm bán chạy và sản phẩm ế ẩm để có chiến lược kinh doanh phù hợp.

Ngoài ra, hệ thống còn tích hợp chức năng phân tích dữ liệu từ chatbot để hiểu rõ xu hướng tìm kiếm của khách hàng, các câu hỏi thường gặp, từ đó admin có thể cải thiện chatbot và bổ sung thêm sản phẩm theo nhu cầu thị trường. Chatbot cũng học hỏi từ các tương tác trước đó để đưa ra gợi ý ngày càng chính xác và phù hợp hơn với từng khách hàng.

2.2. Yêu cầu xây dựng hệ thống

❖ Trang chủ

- Trình bày thân thiện với giao diện hiện đại
- Hiển thị các sản phẩm giày nổi bật và bán chạy
- Banner quảng cáo các chương trình khuyến mãi hiện tại
- Thanh tìm kiếm thông minh với gợi ý tự động
- Menu danh mục sản phẩm theo loại giày (Sneaker, Giày thể thao, Boot, Sandal)
- Menu thương hiệu (Nike, Adidas, Vans, Converse, v.v.)
- Tích hợp chatbot widget ở góc màn hình
- Các chức năng cơ bản: đăng nhập, đăng ký, gio hàng
- Hiển thị thông tin liên hệ và chính sách bảo hành

❖ Trang chatbot

- Giao diện chat thân thiện, dễ sử dụng
- Hỗ trợ nhập liệu bằng text và các button nhanh
- Hiển thị sản phẩm với hình ảnh, giá cả trong chat
- Chức năng tư vấn theo nhu cầu (giới tính, mục đích sử dụng, ngân sách)
- Tìm kiếm nhanh bằng ngôn ngữ tự nhiên
- Lưu lịch sử chat của người dùng
- Kết nối trực tiếp đến trang sản phẩm từ chat

❖ Trang danh sách sản phẩm

- Trình bày dạng lưới với hình ảnh sản phẩm rõ nét
- Bộ lọc thông minh: thương hiệu, giá, màu sắc, size, loại giày
- Sắp xếp theo: giá tăng/giảm dần, mới nhất, bán chạy nhất
- Hiển thị số lượng sản phẩm tìm thấy
- Phân trang để tối ưu tốc độ tải
- Chức năng so sánh sản phẩm
- Thêm nhanh vào gio hàng từ trang danh sách

❖ Trang chi tiết sản phẩm

- Hiển thị hình ảnh sản phẩm từ nhiều góc độ
- Thông tin chi tiết: tên, thương hiệu, mô tả, chất liệu
- Bảng size chi tiết với hướng dẫn chọn size
- Lựa chọn màu sắc với preview hình ảnh
- Hiển thị tình trạng còn hàng theo từng size
- Chức năng thêm vào giỏ hàng và mua ngay
- Đánh giá và nhận xét từ khách hàng đã mua
- Sản phẩm tương tự và gợi ý kết hợp

❖ Trang giỏ hàng và thanh toán

- Hiển thị danh sách sản phẩm đã chọn với hình ảnh, size, màu
- Thay đổi số lượng hoặc xóa sản phẩm khỏi giỏ
- Áp dụng mã giảm giá và voucher
- Tính toán chi phí vận chuyển
- Hiển thị tổng tiền rõ ràng
- Form nhập thông tin giao hàng
- Lựa chọn phương thức thanh toán
- Xác nhận đặt hàng và gửi email xác nhận

❖ Trang quản trị

- **Dashboard:** Thống kê doanh thu, đơn hàng, sản phẩm bán chạy
- **Quản lý sản phẩm:** Thêm, sửa, xóa sản phẩm với đầy đủ thông tin
- **Quản lý danh mục:** Tạo và quản lý các loại giày, thương hiệu
- **Quản lý kho hàng:** Cập nhật số lượng theo size và màu sắc
- **Quản lý đơn hàng:** Xem, xác nhận, cập nhật trạng thái đơn hàng
- **Quản lý khách hàng:** Xem thông tin, lịch sử mua hàng
- **Quản lý chatbot:** Xem lịch sử chat, câu hỏi thường gặp
- **Báo cáo:** Thống kê bán hàng
- **Phân quyền:** Tạo tài khoản nhân viên với các quyền khác nhau

2.3. Phân tích thiết kế

2.3.1. Đối với khách hàng (user)

Phân loại người dùng:

- Guest User (Khách vãng lai): Có thể xem sản phẩm, tìm kiếm, chat với bot nhưng không thể mua hàng
- Registered User (Khách hàng đăng ký): Có đầy đủ quyền mua sắm và quản lý tài khoản

Chức năng chính của User:

A. Đăng ký và xác thực:

- Đăng ký tài khoản với email, mật khẩu, họ tên, số điện thoại
- Đăng nhập/đăng xuất hệ thống
- Quên mật khẩu và khôi phục tài khoản

B. Duyệt và tìm kiếm sản phẩm:

- Xem danh sách sản phẩm theo danh mục
- Sử dụng bộ lọc: thương hiệu, giá, size, màu sắc
- Tìm kiếm sản phẩm bằng từ khóa
- Xem chi tiết sản phẩm với hình ảnh, thông số kỹ thuật
- Đọc đánh giá và nhận xét từ khách hàng khác

C. Tương tác với Chatbot:

- Chat với AI để được tư vấn sản phẩm
- Nhận gợi ý sản phẩm phù hợp với nhu cầu
- Hỏi đáp về thông tin sản phẩm, size, giá cả

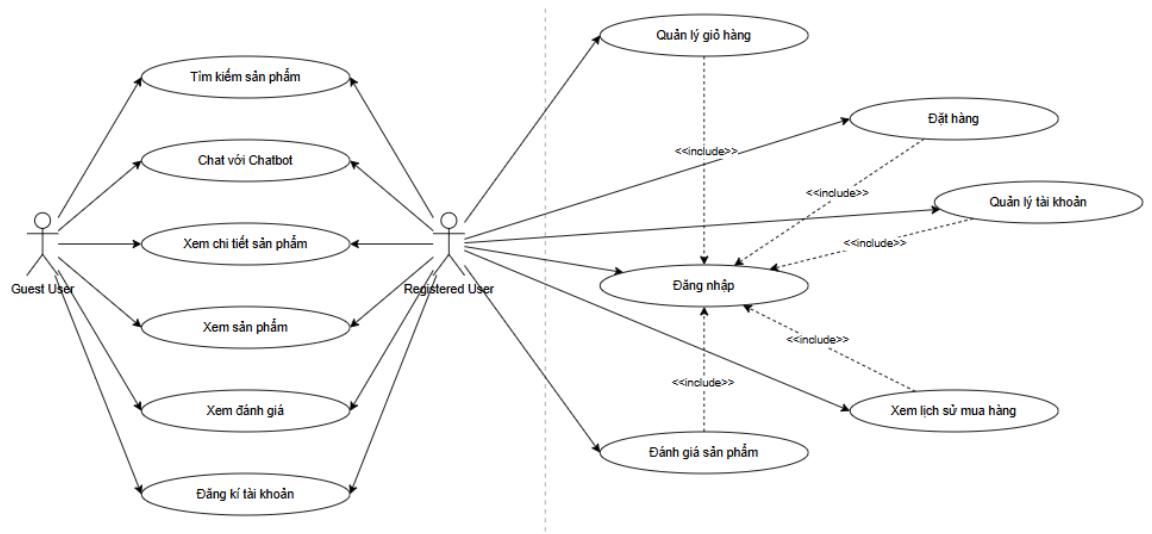
D. Quản lý giỏ hàng và mua sắm:

- Thêm sản phẩm vào giỏ hàng (yêu cầu đăng nhập)
- Điều chỉnh số lượng hoặc xóa sản phẩm
- Đặt hàng với thông tin giao hàng
- Thanh toán bằng tiền mặt (COD)
- Theo dõi trạng thái đơn hàng

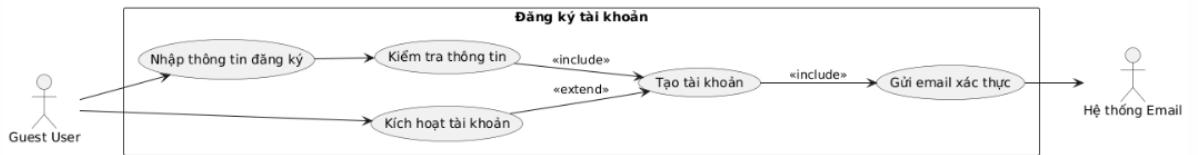
E. Quản lý tài khoản cá nhân:

- Cập nhật thông tin cá nhân
- Xem lịch sử mua hàng
- Đánh giá sản phẩm đã mua

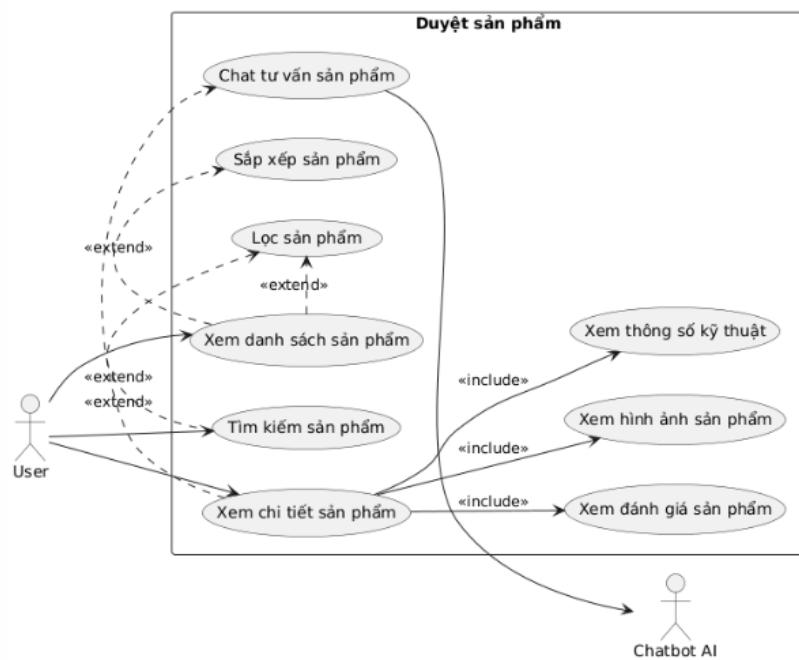
2.3.1.a. Đối với khách hàng (User) – Sơ đồ use case



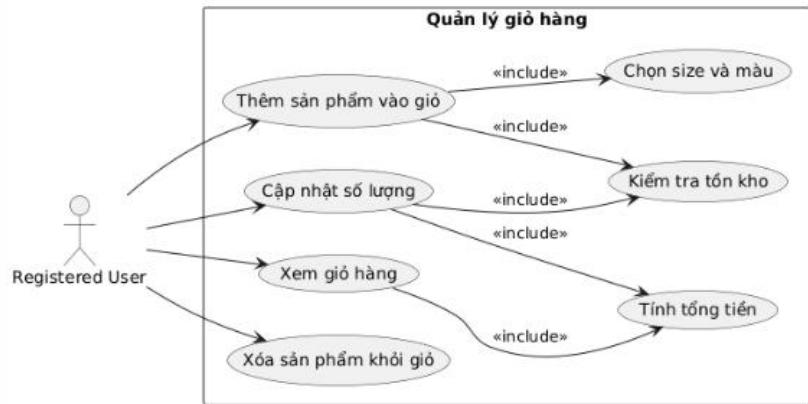
Hình 2.1: Use Case tổng quan cho User



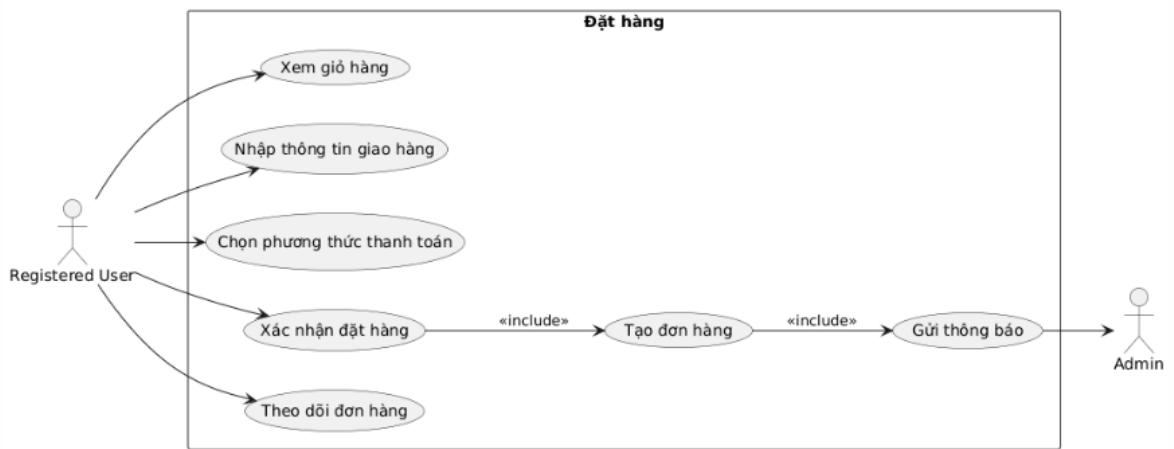
Hình 2.2: Use Case chi tiết - Đăng ký tài khoản



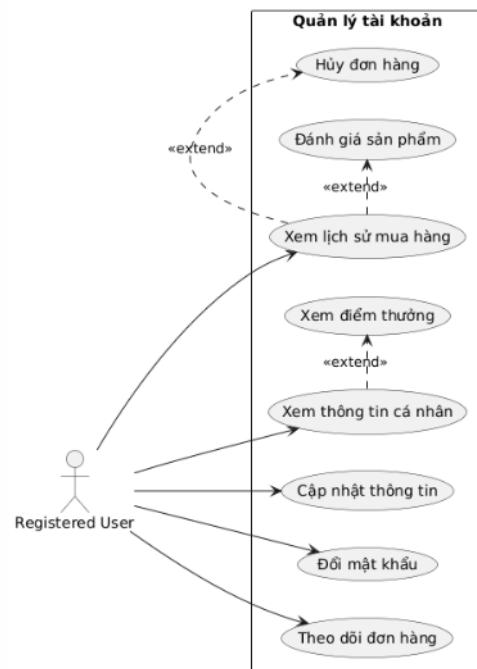
Hình 2.3: Use Case chi tiết - Xem và tìm kiếm sản phẩm



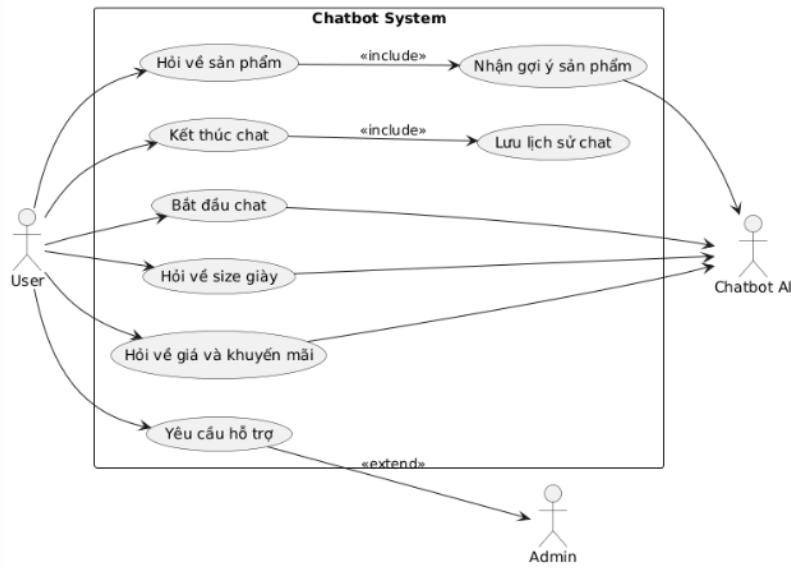
Hình 2.4: Use Case chi tiết - Quản lý giỏ hàng



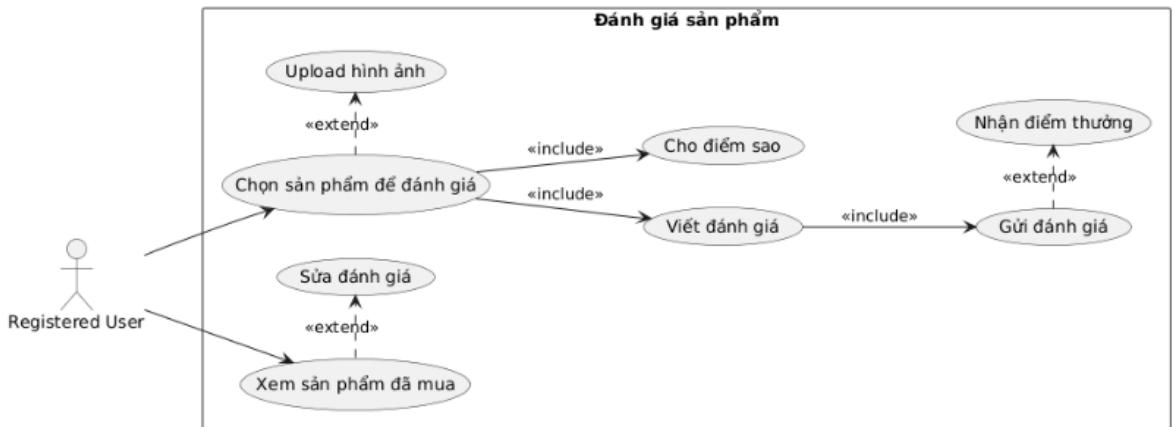
Hình 2.5: Use Case chi tiết - Đặt hàng



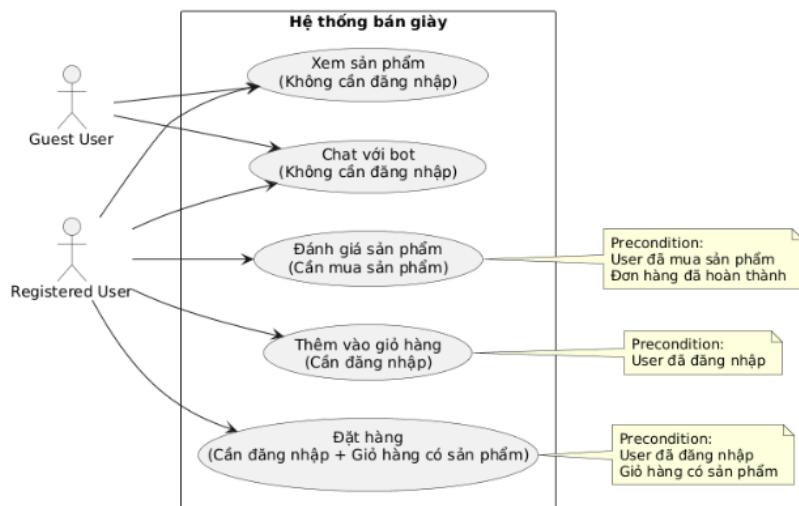
Hình 2.6: Use Case chi tiết - Quản lý tài khoản cá nhân



Hình 2.7: Use Case chi tiết - Tương tác với Chatbot

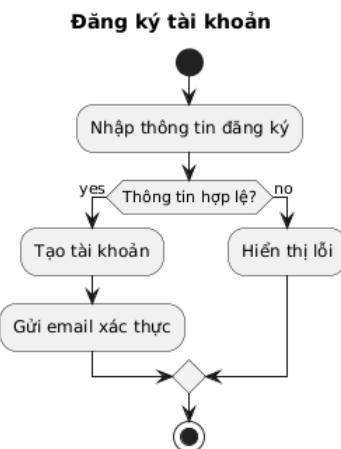


Hình 2.8: Use Case chi tiết - Đánh giá sản phẩm

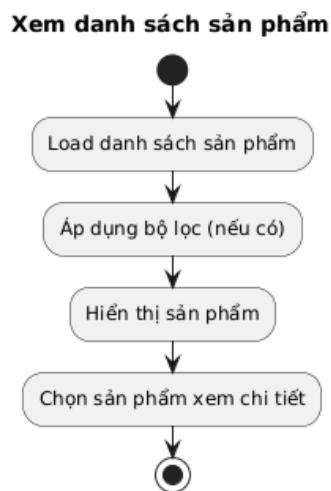


Hình 2.9: Use Case với các ràng buộc và điều kiện

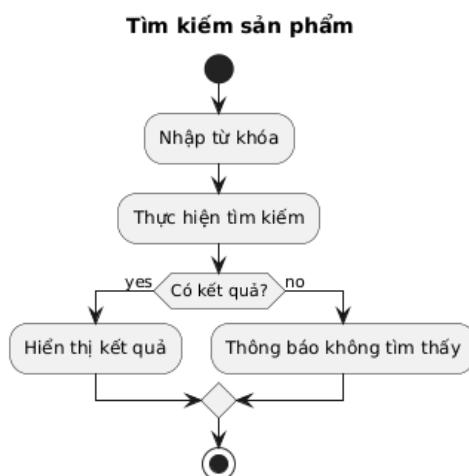
2.3.1.b. Đối với khách hàng (User) - Biểu đồ hoạt động



Hình 2.10: Biểu đồ hoạt động - Đăng ký tài khoản mới

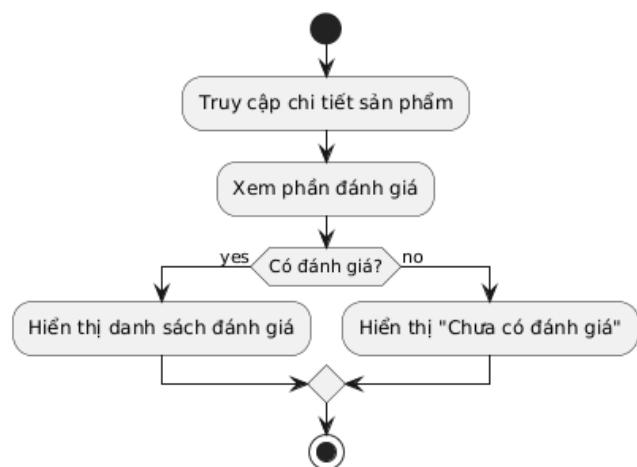


Hình 2.11: Biểu đồ hoạt động - Xem danh sách sản phẩm



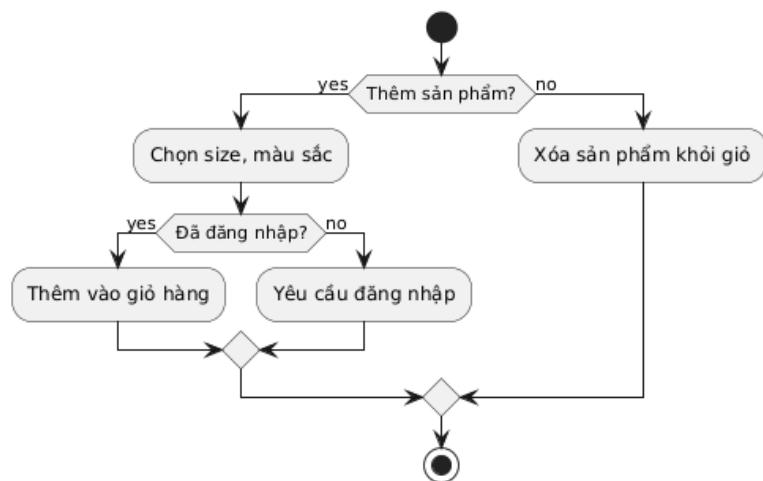
Hình 2.12: Biểu đồ hoạt động - Tìm kiếm sản phẩm

Xem đánh giá sản phẩm



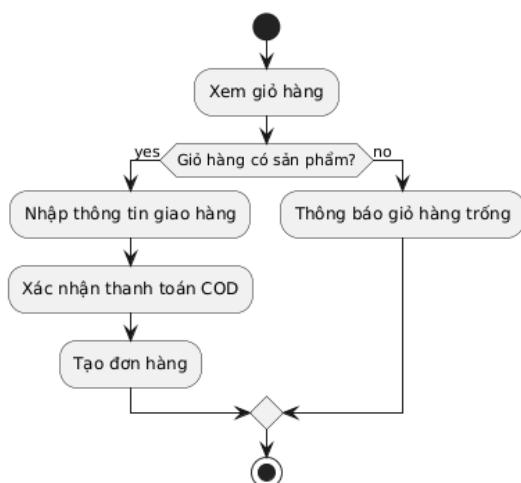
Hình 2.13: Biểu đồ hoạt động - Xem đánh giá sản phẩm

Quản lý giỏ hàng



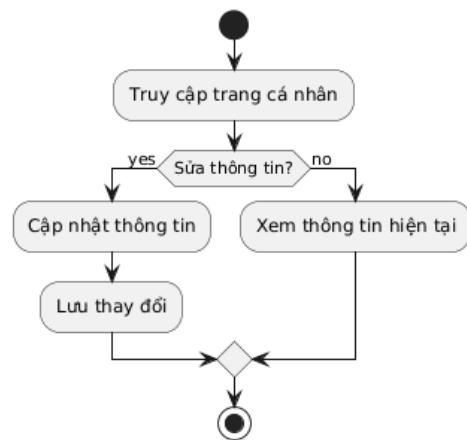
Hình 2.14: Biểu đồ hoạt động - Thêm/xóa sản phẩm giỏ hàng

Đặt hàng



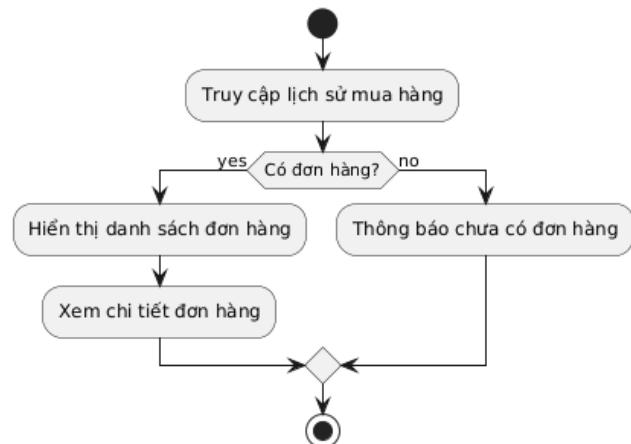
Hình 2.15: Biểu đồ hoạt động - Đặt hàng

Quản lý thông tin cá nhân



Hình 2.16: Biểu đồ hoạt động - Quản lý thông tin cá nhân

Xem lịch sử mua hàng



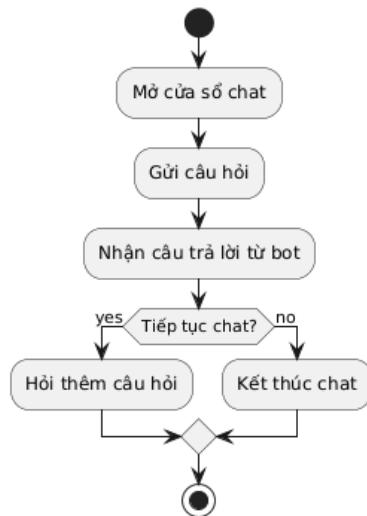
Hình 2.17: Biểu đồ hoạt động - Xem lịch sử mua hàng

Đánh giá sản phẩm



Hình 2.18: Biểu đồ hoạt động - Đánh giá sản phẩm đã mua

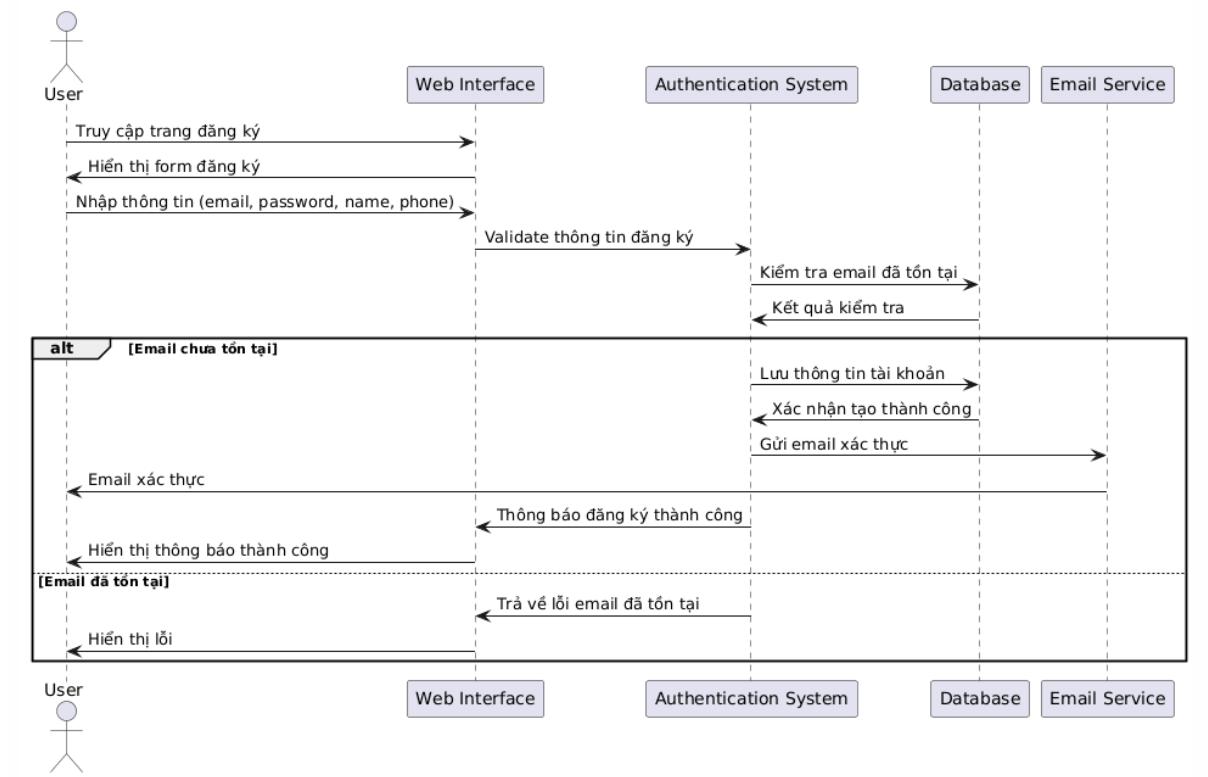
Chat với Chatbot



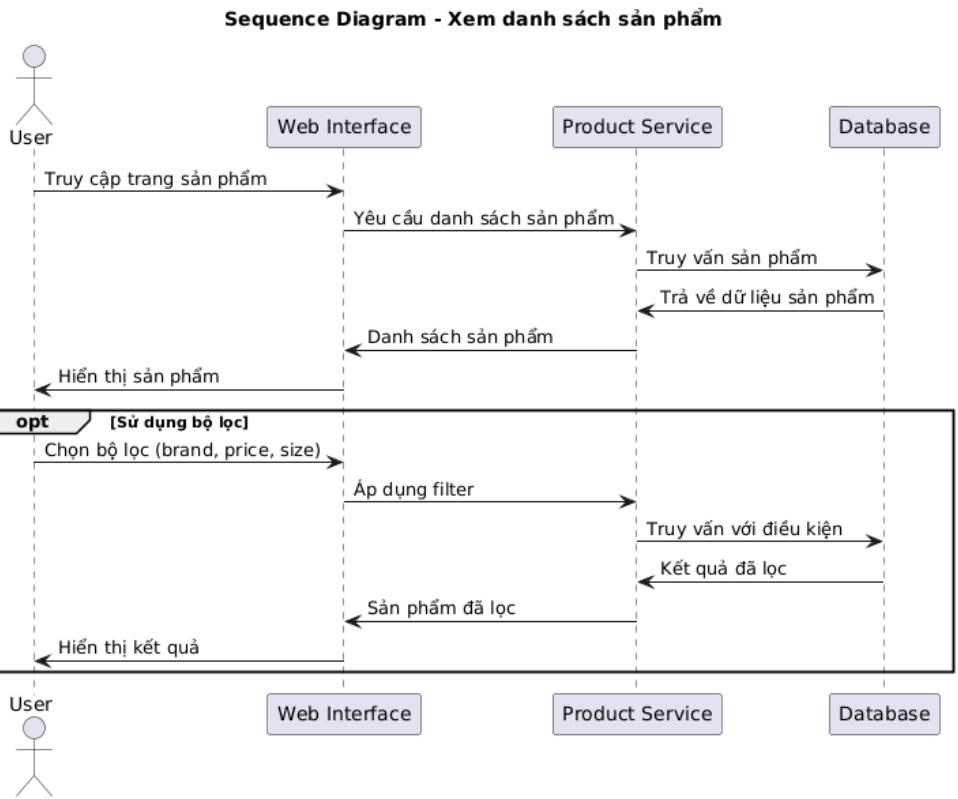
Hình 2.19: Biểu đồ hoạt động - Chat với Chatbot

2.3.1.c. Đối với khách hàng (User) - Biểu đồ trình tự

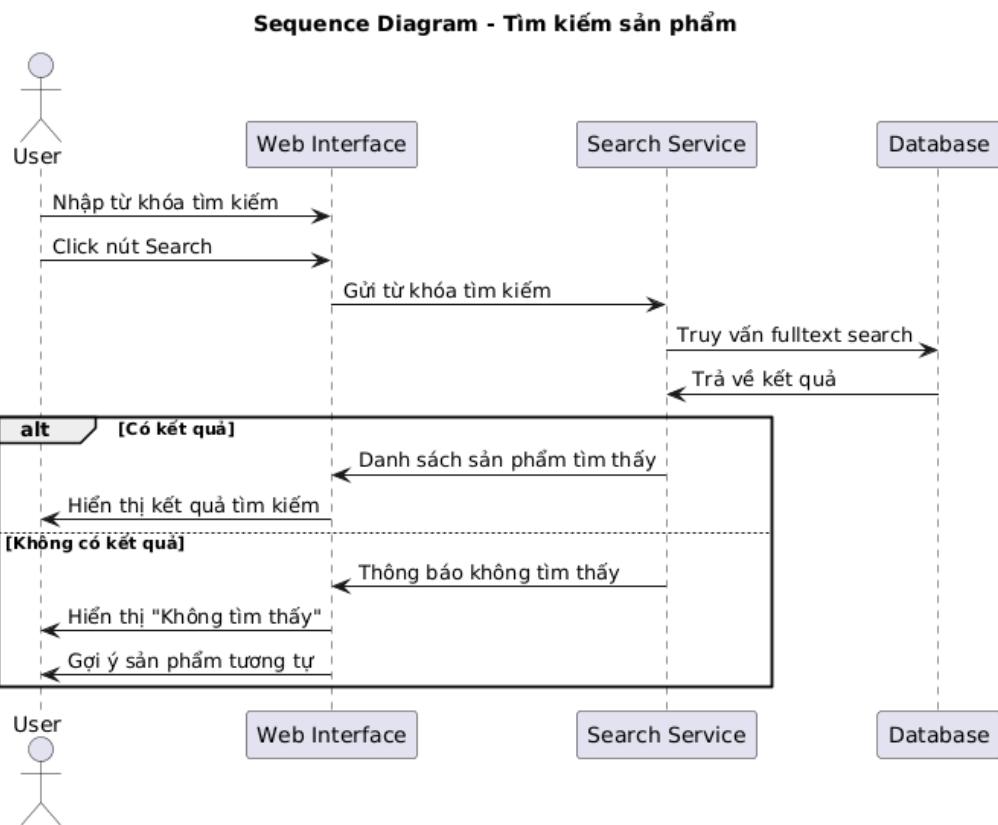
Sequence Diagram - Đăng ký tài khoản



Hình 2.20: Biểu đồ trình tự - Đăng ký tài khoản mới

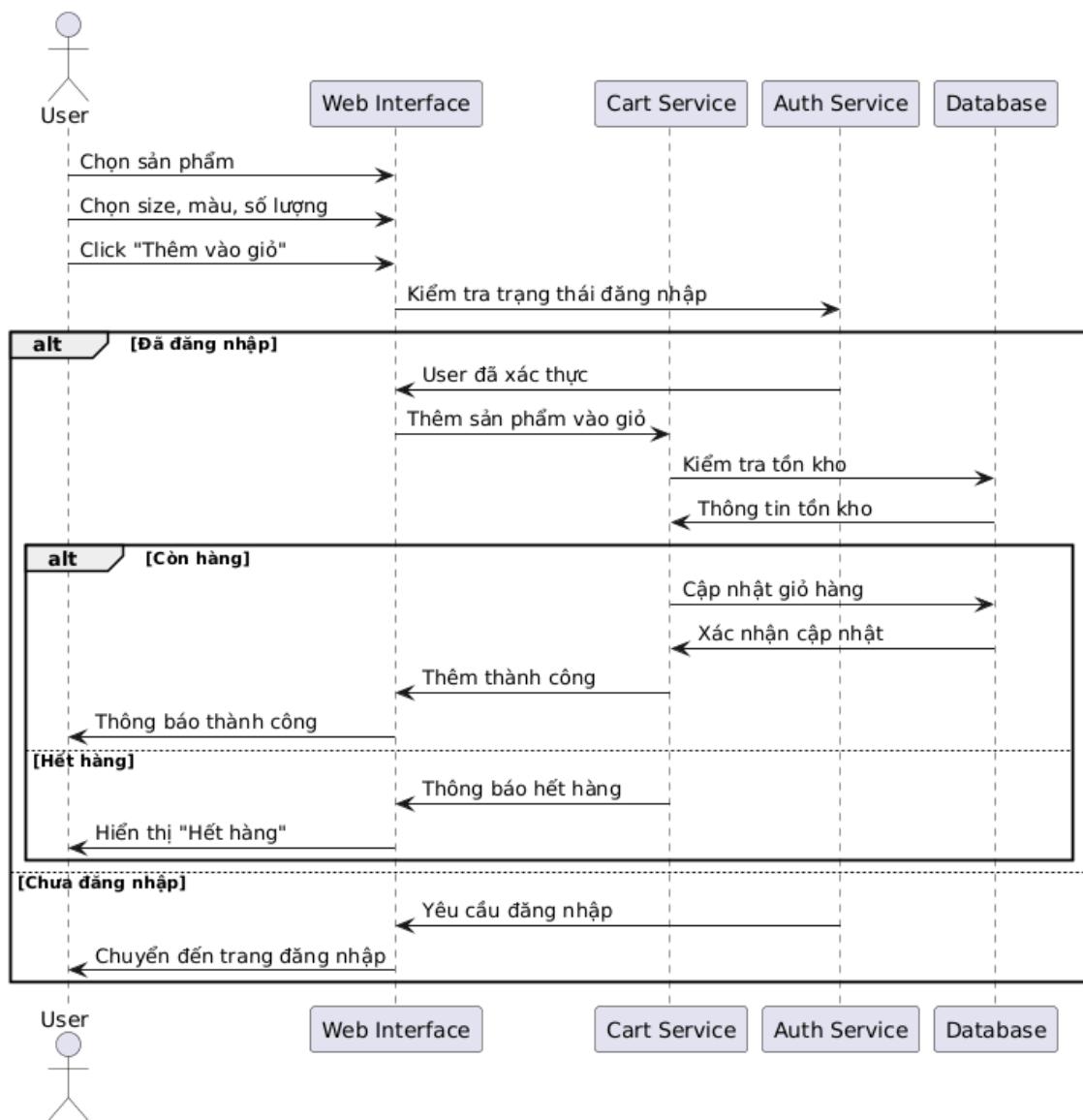


Hình 2.21: Biểu đồ trình tự - Xem danh sách sản phẩm

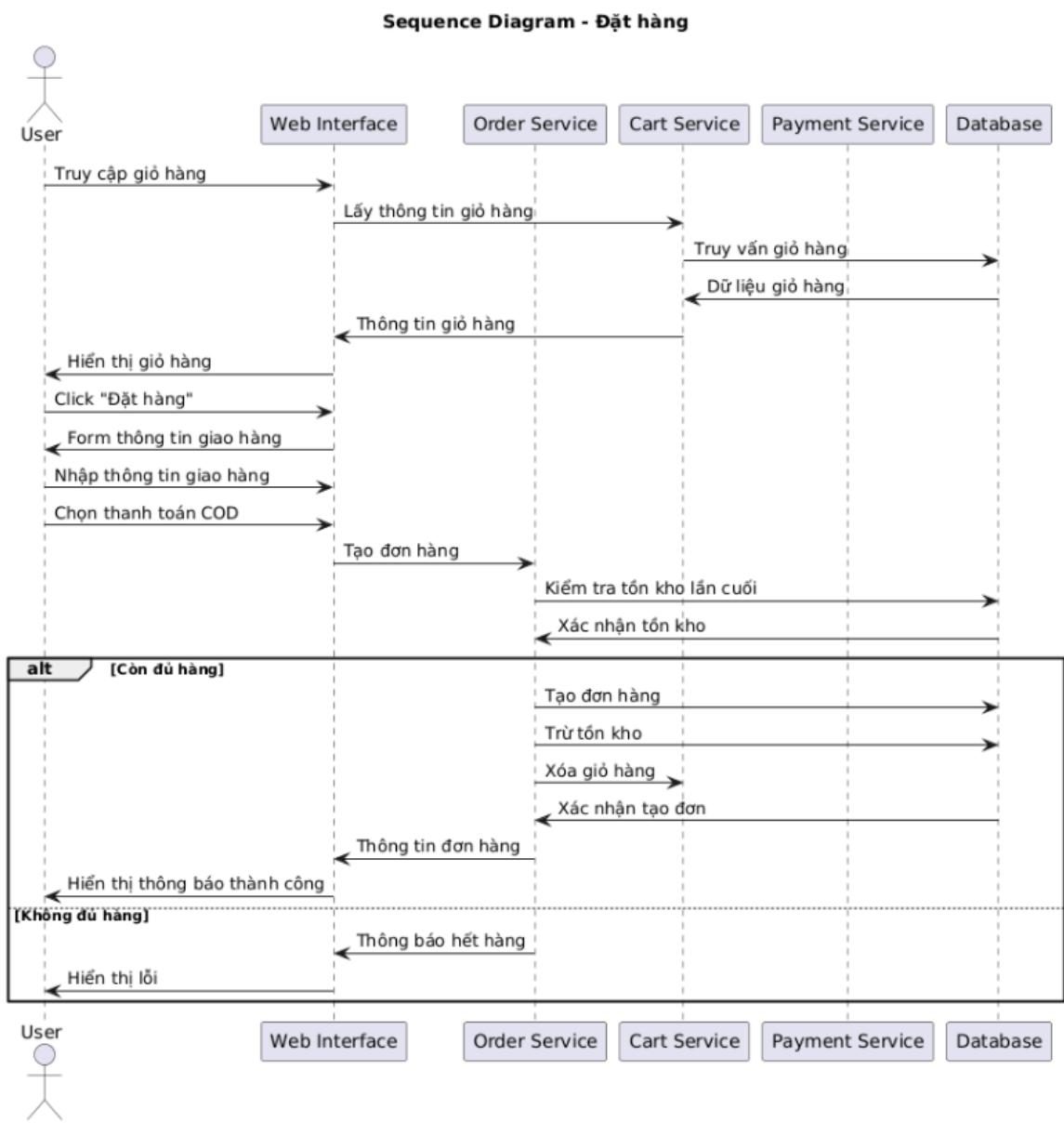


Hình 2.22: Biểu đồ trình tự - Tìm kiếm sản phẩm

Sequence Diagram - Thêm sản phẩm vào giỏ hàng

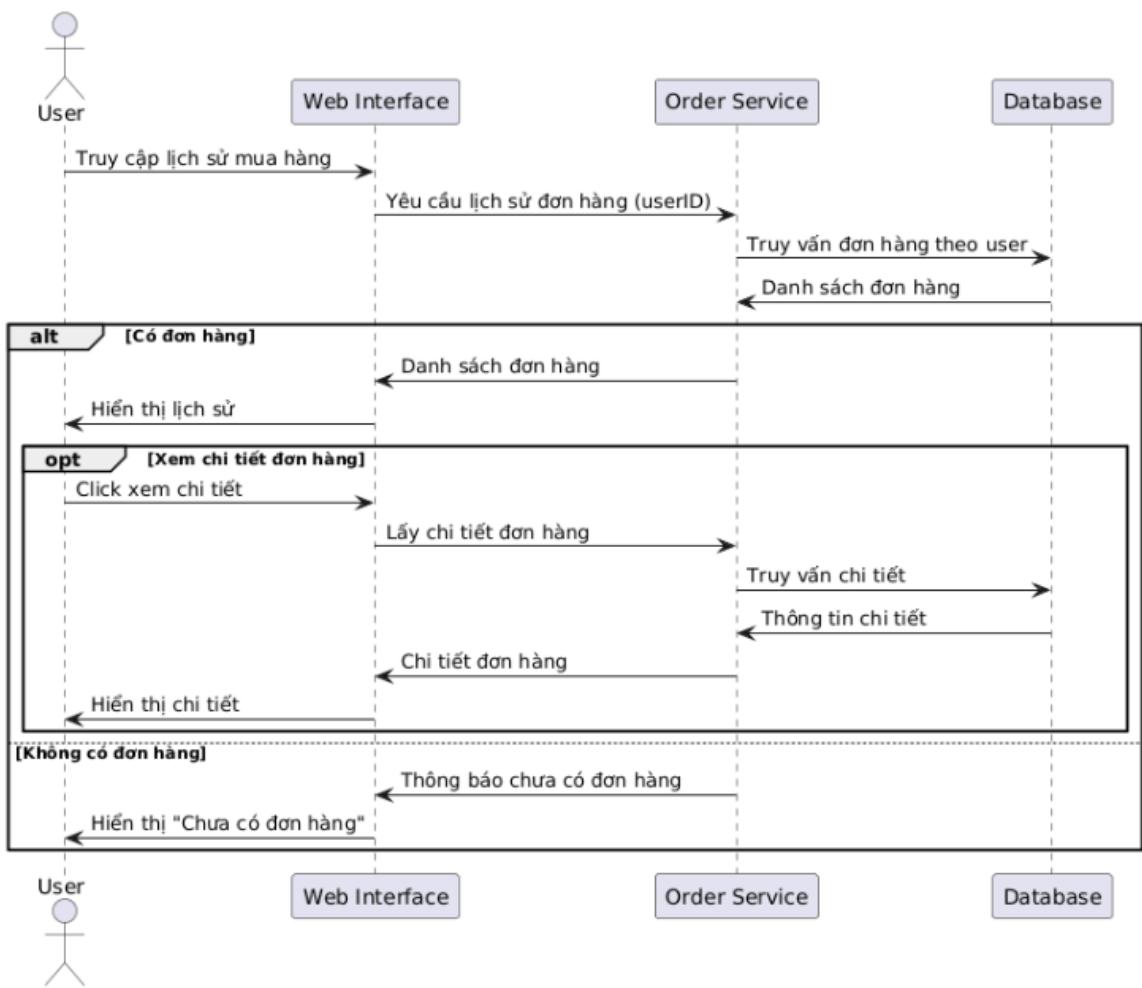


Hình 2.23: Biểu đồ trình tự - Thêm sản phẩm vào giỏ hàng

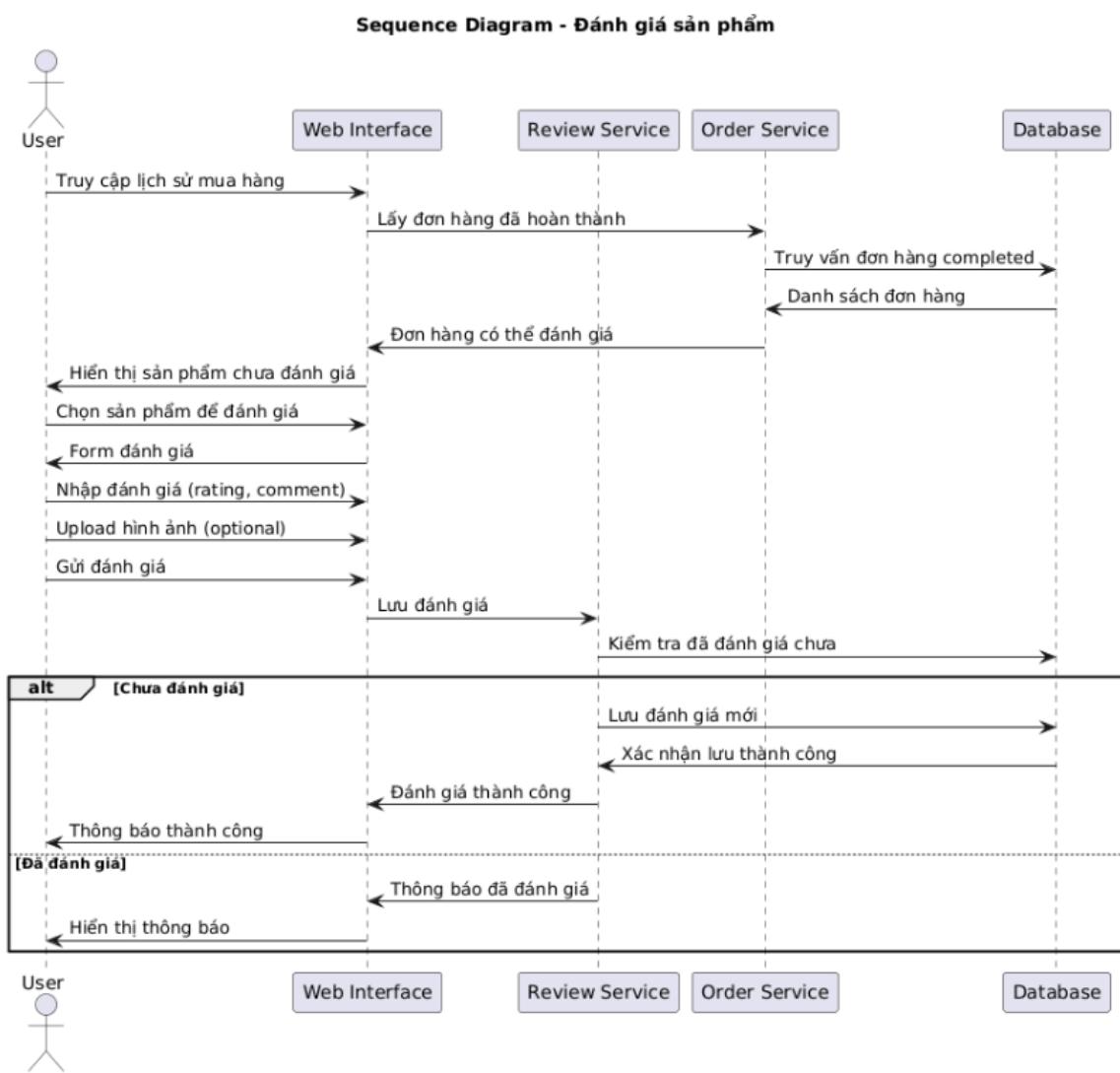


Hình 2.24: Biểu đồ trình tự - Đặt hàng

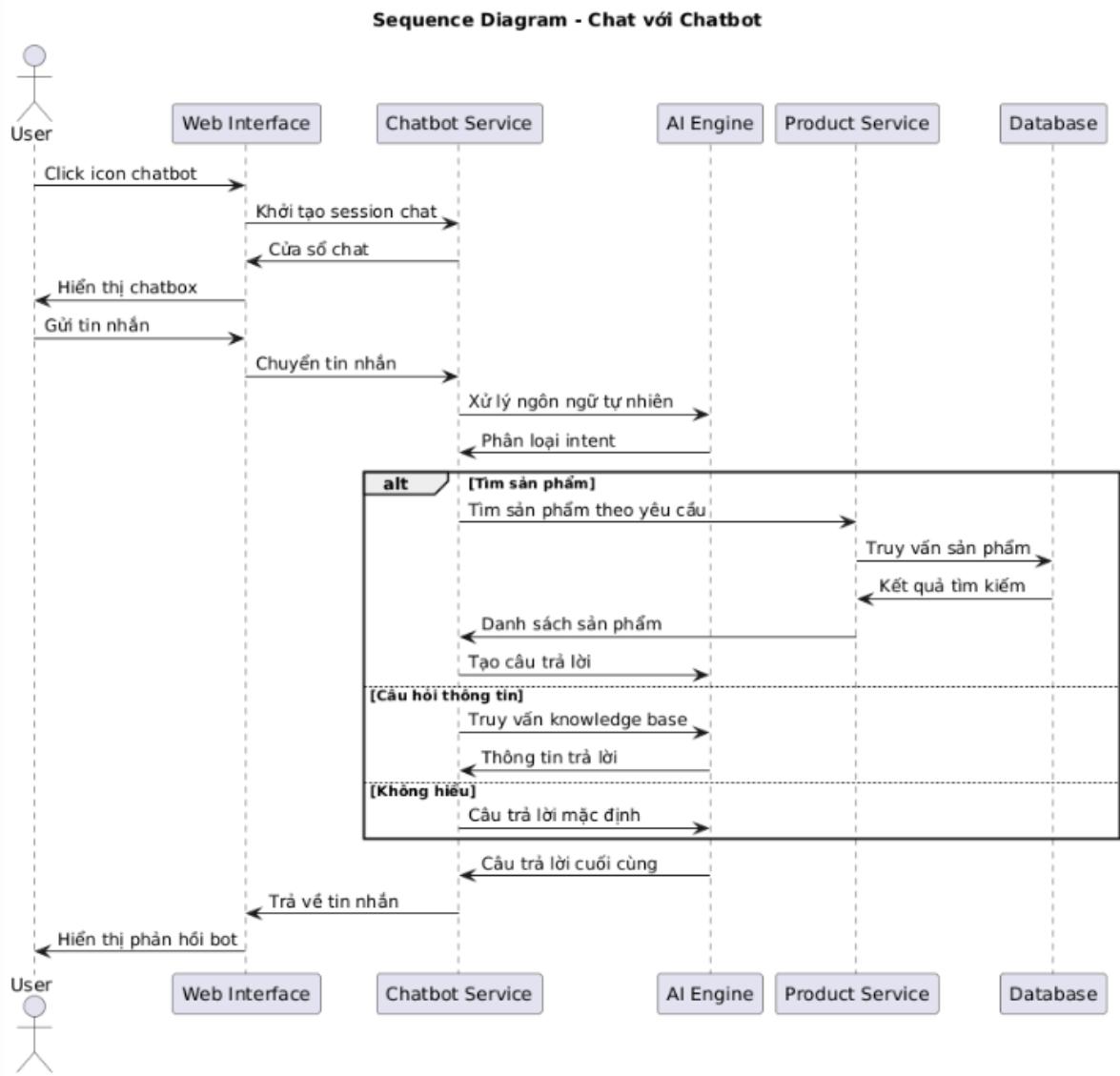
Sequence Diagram - Xem lịch sử mua hàng



Hình 2.25: Biểu đồ trình tự - Xem lịch sử mua hàng

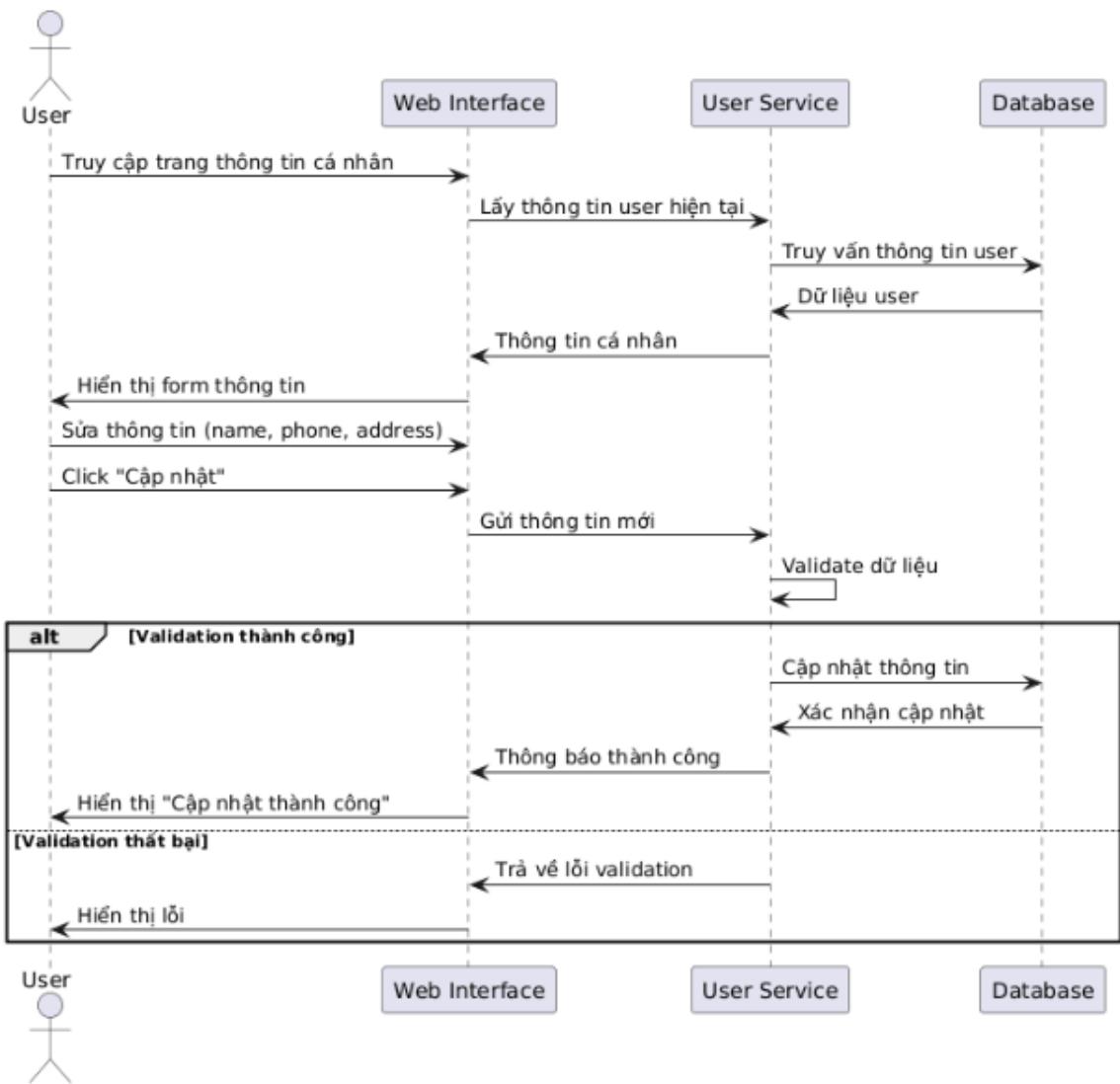


Hình 2.26: Biểu đồ trình tự - Đánh giá sản phẩm



Hình 2.27: Biểu đồ trình tự - Chat với Chatbot

Sequence Diagram - Cập nhật thông tin cá nhân



Hình 2.28: Biểu đồ trình tự - Quản lý thông tin cá nhân

2.3.2. Đối với quản trị viên (admin)

Phân loại Admin:

A. Super Admin (Quản lý toàn hệ thống):

- Quyền cao nhất, quản lý tất cả chức năng
- Tạo và phân quyền cho nhân viên
- Báo cáo tổng thể

B. Staff (Nhân viên):

- Chỉ được xem và xử lý đơn hàng

- Có thể hủy đơn hàng khi cần thiết
- Không thể thay đổi sản phẩm hoặc cấu hình hệ thống

Chức năng Super Admin:

A. Quản lý sản phẩm:

- Thêm, sửa, xóa sản phẩm
- Upload và quản lý hình ảnh sản phẩm
- Cập nhật giá, mô tả, thông số kỹ thuật
- Quản lý kho hàng theo size và màu sắc
- Quản lý danh mục và thương hiệu

B. Quản lý đơn hàng:

- Xem tất cả đơn hàng trong hệ thống
- Xác nhận, cập nhật trạng thái đơn hàng
- Hủy đơn hàng và hoàn trả kho
- In hóa đơn và quản lý giao hàng
- Liên hệ khách hàng

C. Quản lý khách hàng:

- Xem danh sách và thông tin khách hàng
- Xem lịch sử mua hàng của từng khách
- Khóa/mở khóa tài khoản khi cần
- Hỗ trợ khách hàng qua chat hoặc email

D. Quản lý nhân viên:

- Tạo tài khoản nhân viên
- Phân quyền truy cập
- Theo dõi hoạt động của nhân viên
- Khóa/mở tài khoản nhân viên

E. Báo cáo và thống kê:

- Dashboard tổng quan: doanh thu, đơn hàng, khách hàng
- Báo cáo bán hàng theo thời gian
- Thông kê sản phẩm bán chạy
- Phân tích hành vi khách hàng

- Báo cáo hoạt động chatbot

Chức năng Staff (Nhân viên):

- Xem danh sách đơn hàng được phân công
- Xem chi tiết thông tin đơn hàng
- Cập nhật trạng thái đơn hàng
- Hủy đơn hàng với lý do cụ thể
- Liên hệ khách hàng qua điện thoại
- Ghi chú xử lý đơn hàng

Phân quyền rõ ràng:

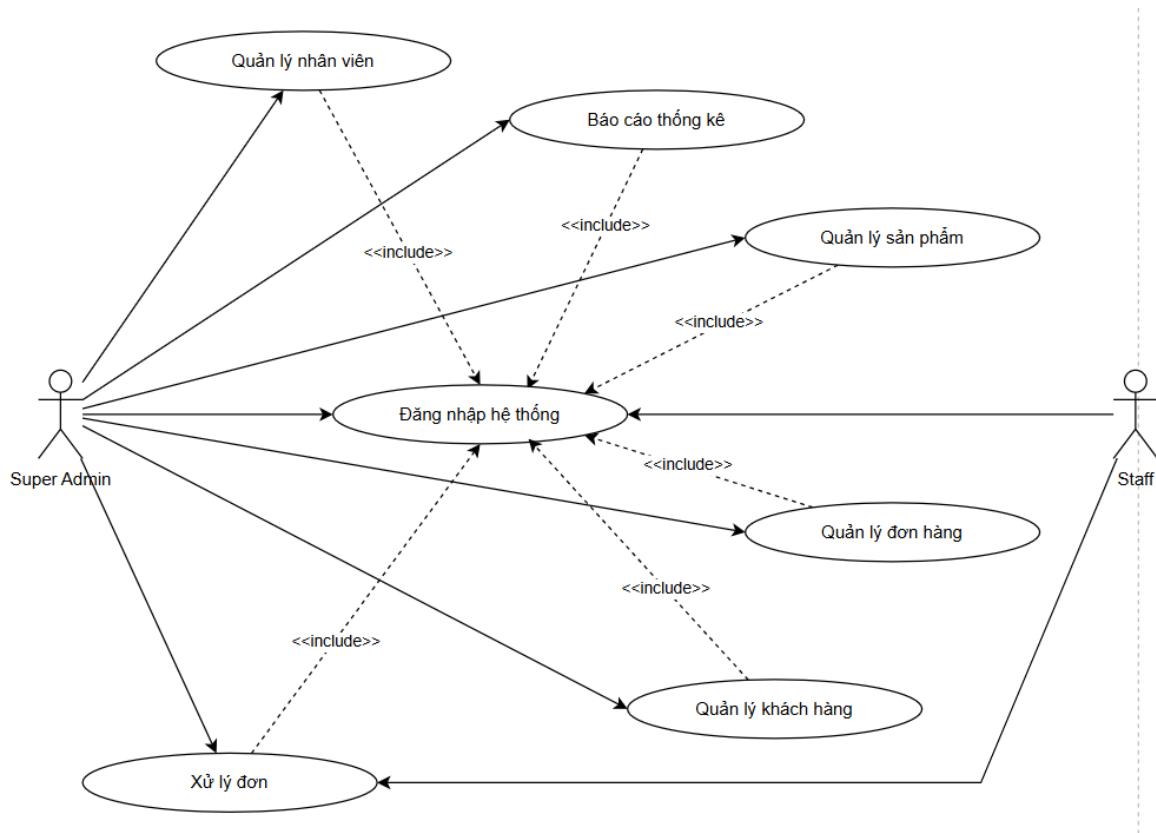
Chức năng	Super Admin	Staff
Quản lý sản phẩm	✓	✗
Xem đơn hàng	✓	✓
Xử lý đơn hàng	✓	✓
Hủy đơn hàng	✓	✓
Quản lý khách hàng	✓	✗
Quản lý nhân viên	✓	✗
Xem báo cáo	✓	✗
Cấu hình hệ thống	✓	✗

Bảng 2.1: Bảng phân quyền

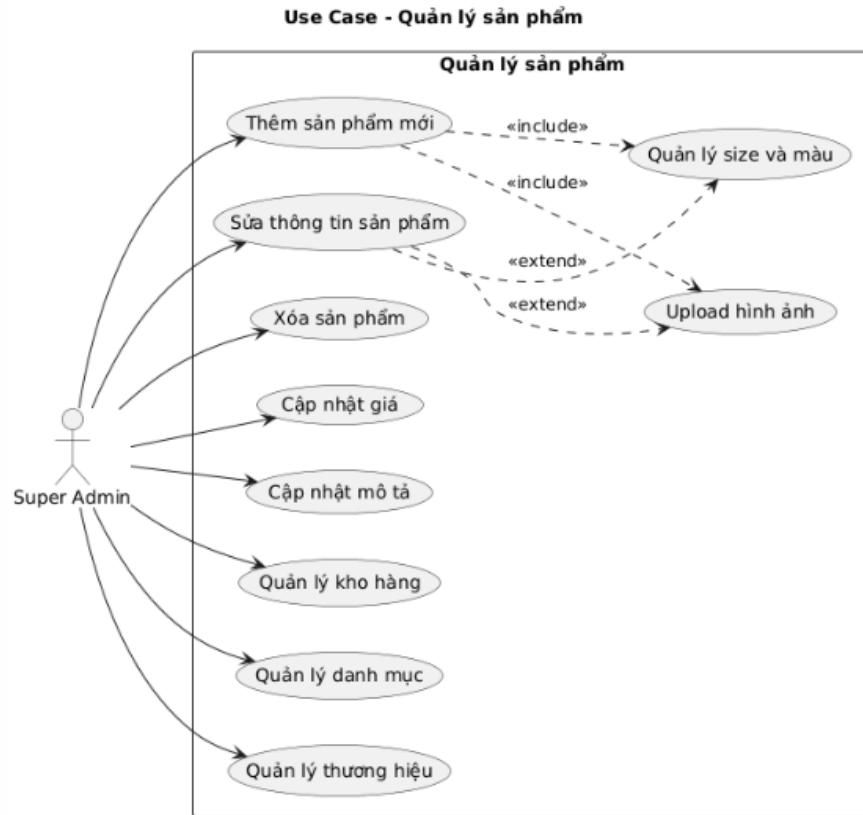
Workflow xử lý đơn hàng:

Khách đặt hàng → Thông báo cho Admin/Staff → Xác nhận đơn hàng
→ Chuẩn bị hàng → Giao hàng → Cập nhật trạng thái → Hoàn thành

2.3.2.a Đối với quản trị viên (admin) – Sơ đồ use case

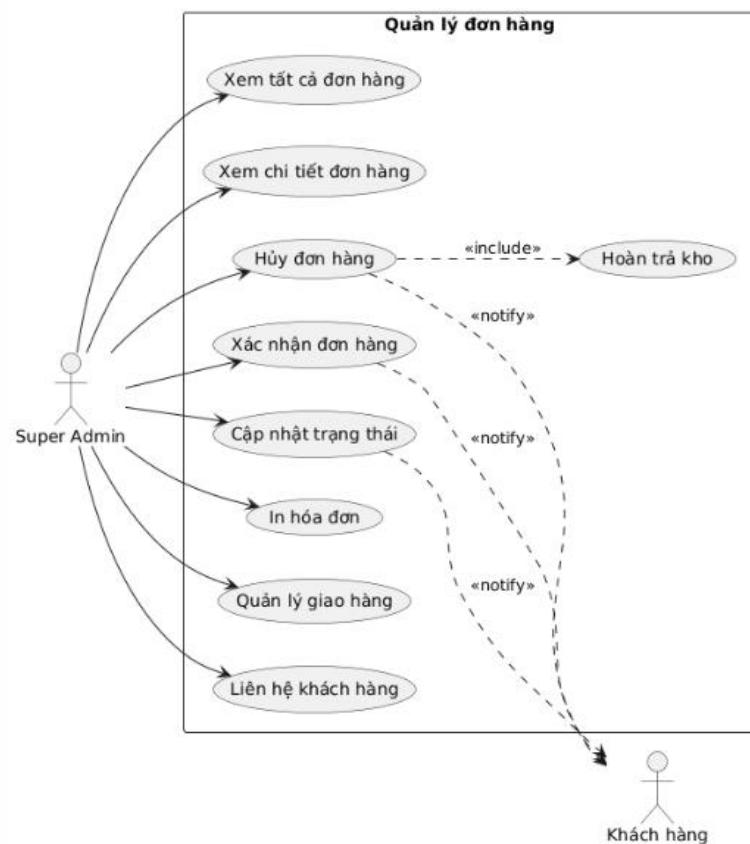


Hình 2.29: Use Case tổng quan - Hệ thống Admin



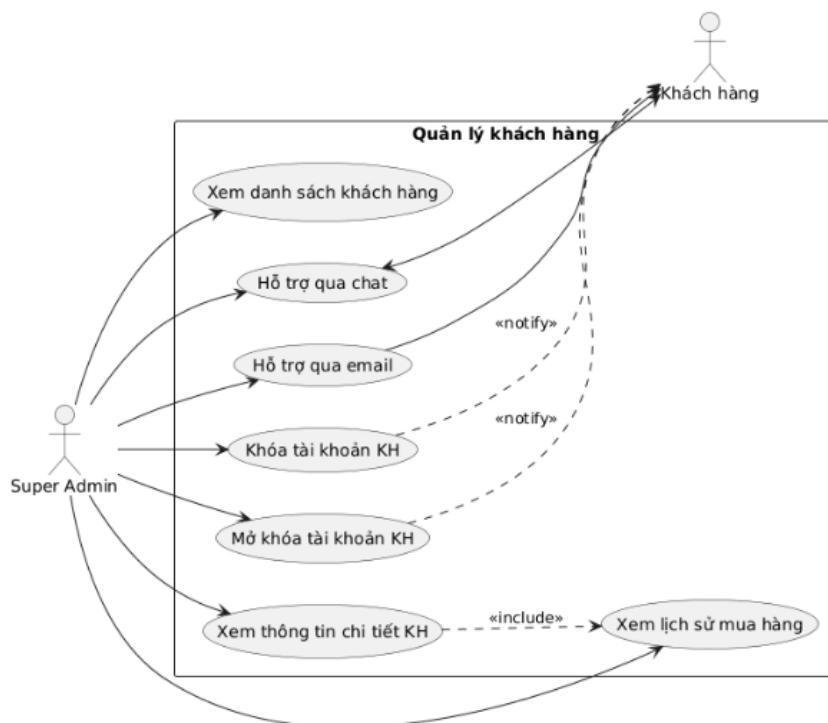
Hình 2.30: Use Case chi tiết - Quản lý sản phẩm (Super Admin)

Use Case - Quản lý đơn hàng (Super Admin)

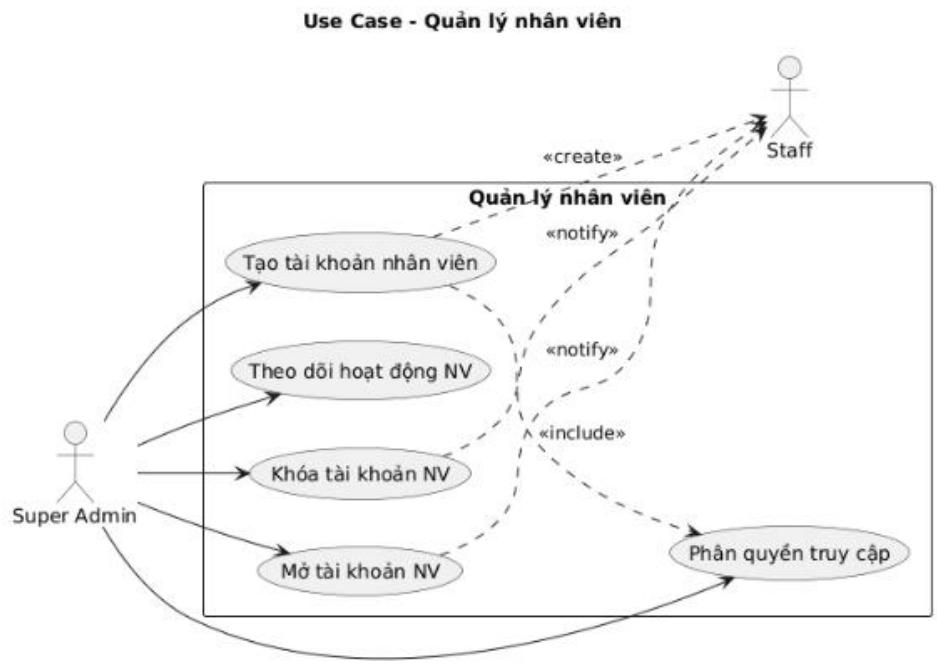


Hình 2.31: Use Case chi tiết - Quản lý đơn hàng (Super Admin)

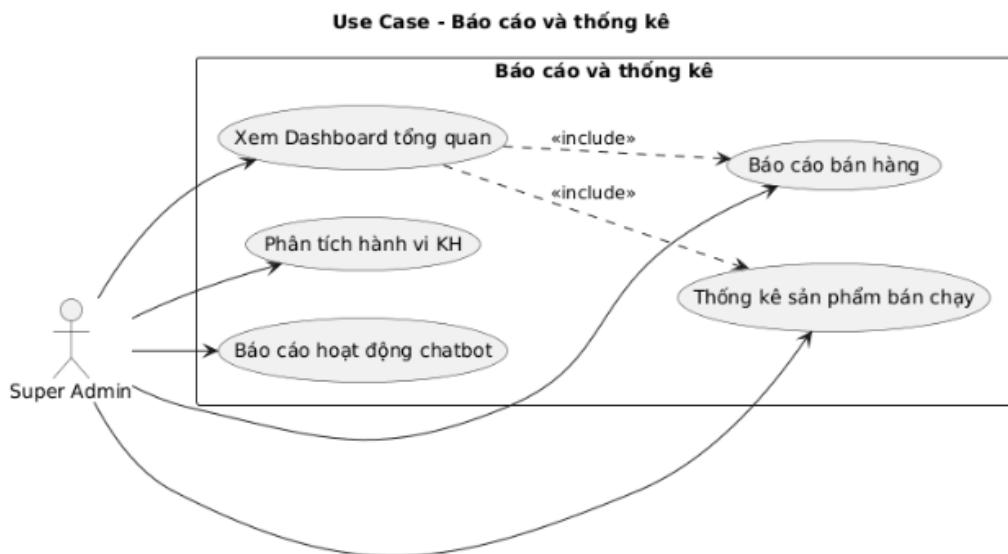
Use Case - Quản lý khách hàng



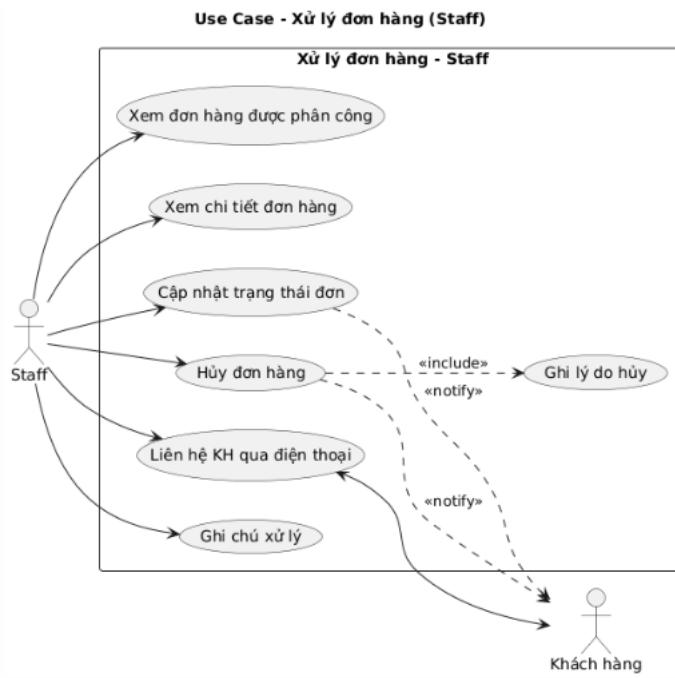
Hình 2.32: Use Case chi tiết - Quản lý khách hàng (Super Admin)



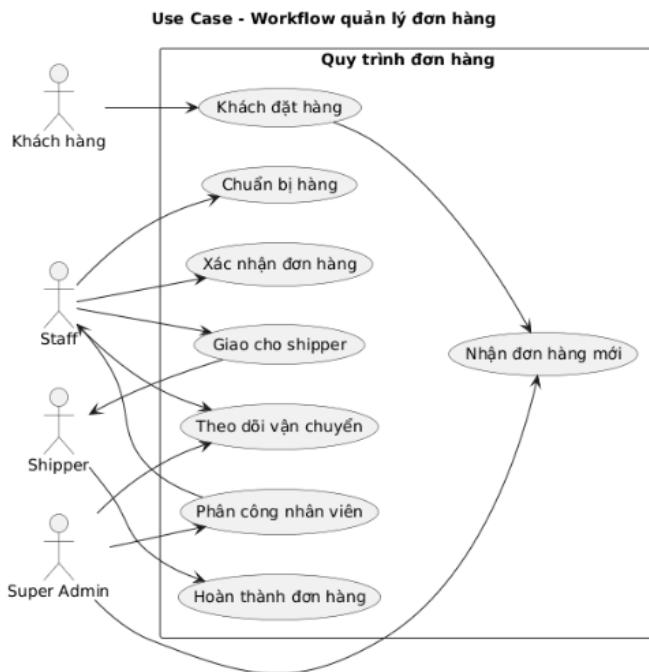
Hình 2.33: Use Case chi tiết - Quản lý nhân viên (Super Admin)



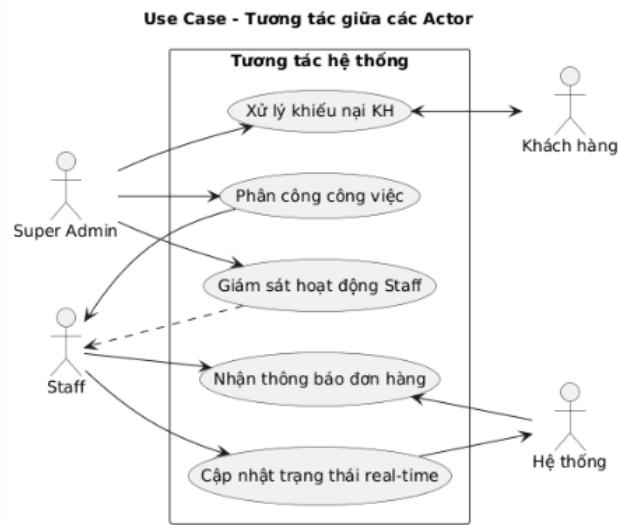
Hình 2.34: Use Case chi tiết - Báo cáo và thống kê (Super Admin)



Hình 2.35: Use Case chi tiết - Xử lý đơn hàng (Staff)



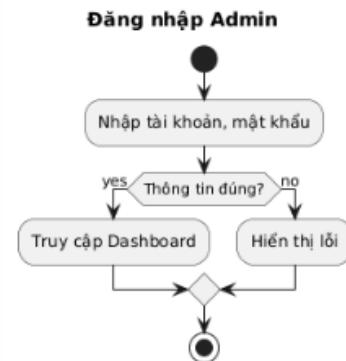
Hình 2.36: Use Case - Workflow quản lý đơn hàng hoàn chỉnh



Hình 2.37: Use Case - Tương tác giữa các Actor

2.3.2.b Đối với quản trị viên (admin) – Biểu đồ hoạt động

A. Super Admin

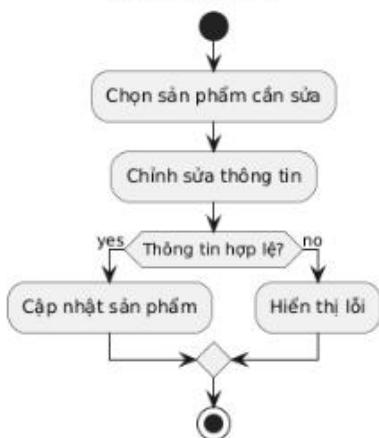


Hình 2.38: Biểu đồ hoạt động - Đăng nhập Admin



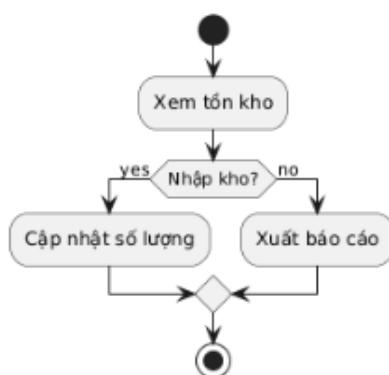
Hình 2.39: Biểu đồ hoạt động - Thêm sản phẩm

Sửa sản phẩm



Hình 2.40: Biểu đồ hoạt động - Sửa sản phẩm

Quản lý kho



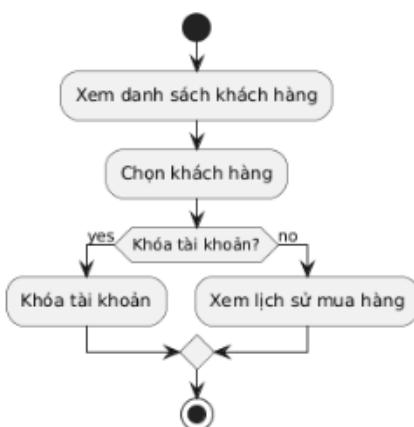
Hình 2.41: Biểu đồ hoạt động - Quản lý kho

Xử lý đơn hàng



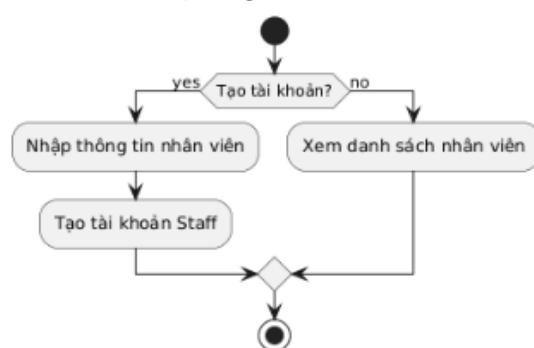
Hình 2.42: Biểu đồ hoạt động - Xử lý đơn hàng

Quản lý khách hàng



Hình 2.43: Biểu đồ hoạt động - Quản lý khách hàng

Quản lý nhân viên



Hình 2.44: Biểu đồ hoạt động - Quản lý nhân viên

Xem báo cáo



Hình 2.45: Biểu đồ hoạt động - Xem báo cáo

B. Staff

Xem đơn hàng Staff



Hình 2.46: Biểu đồ hoạt động - Xem đơn hàng (Staff)

Xử lý đơn hàng Staff



Hình 2.47: Biểu đồ hoạt động - Xử lý đơn hàng (Staff)

Cập nhật trạng thái



Hình 2.48: Biểu đồ hoạt động - Cập nhật trạng thái

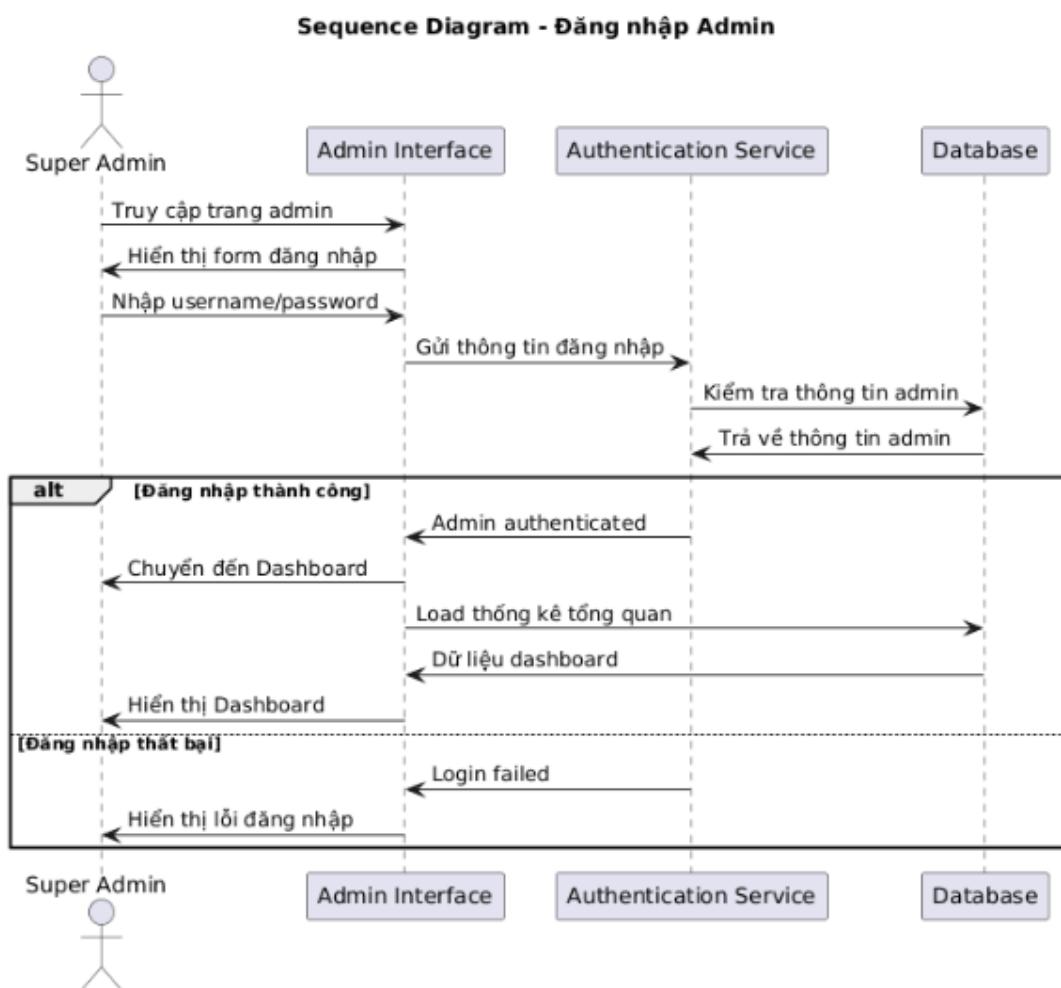
Hủy đơn hàng



Hình 2.49: Biểu đồ hoạt động - Hủy đơn hàng

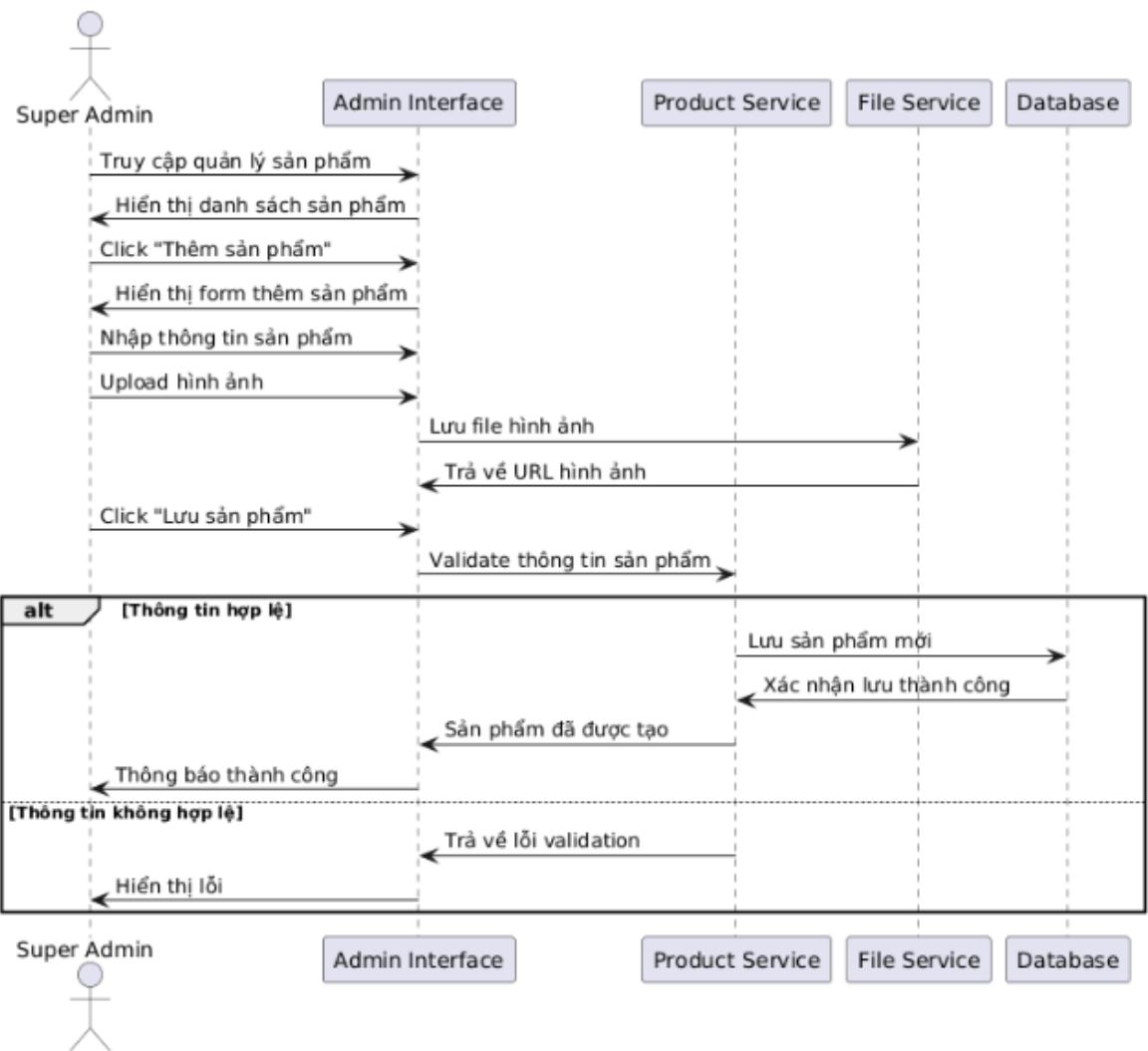
2.3.2.c Đối với quản trị viên (admin) – Biểu đồ trình tự

A. Super Admin



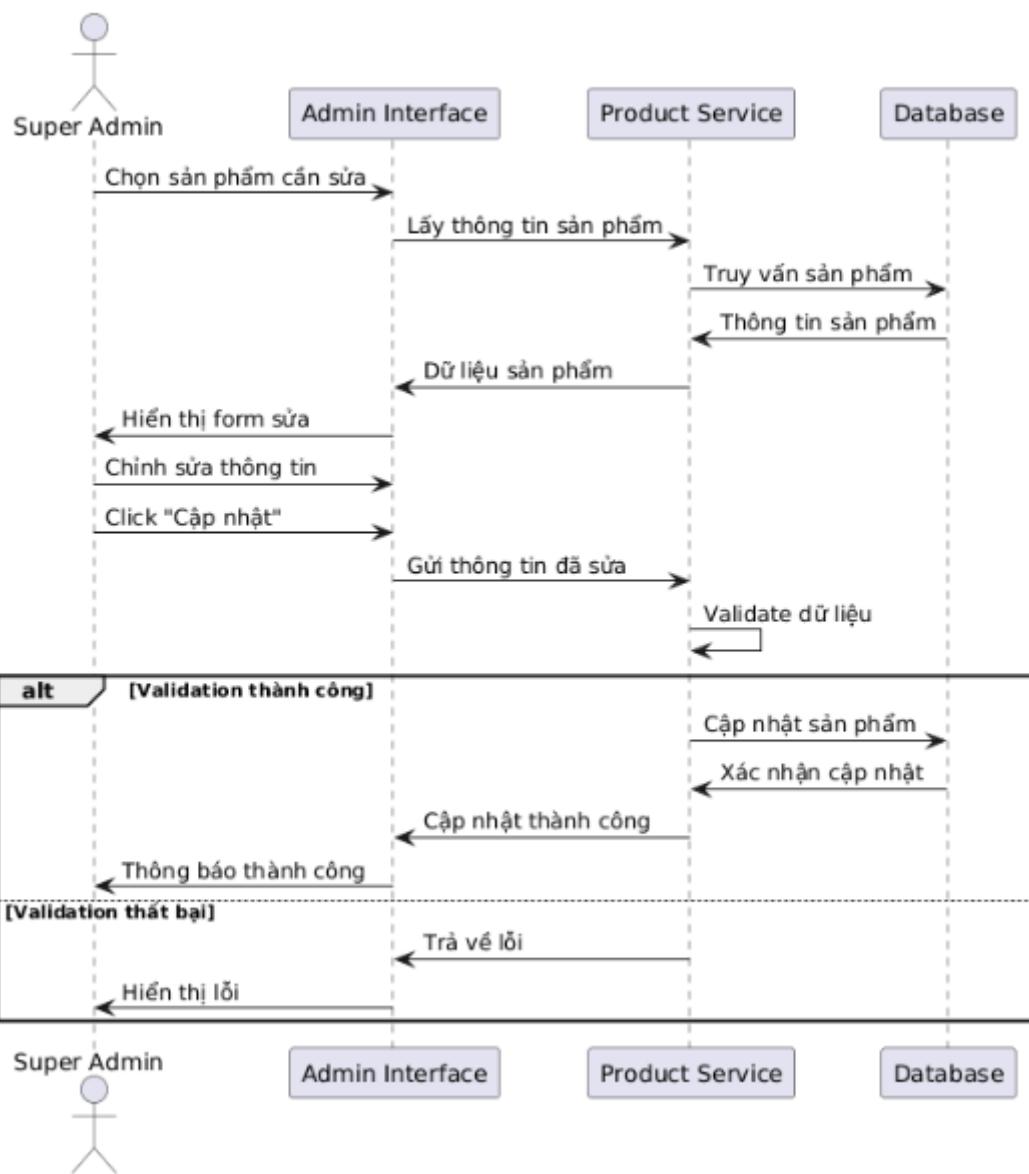
Hình 2.50: Biểu đồ trình tự - Đăng nhập hệ thống Admin

Sequence Diagram - Thêm sản phẩm



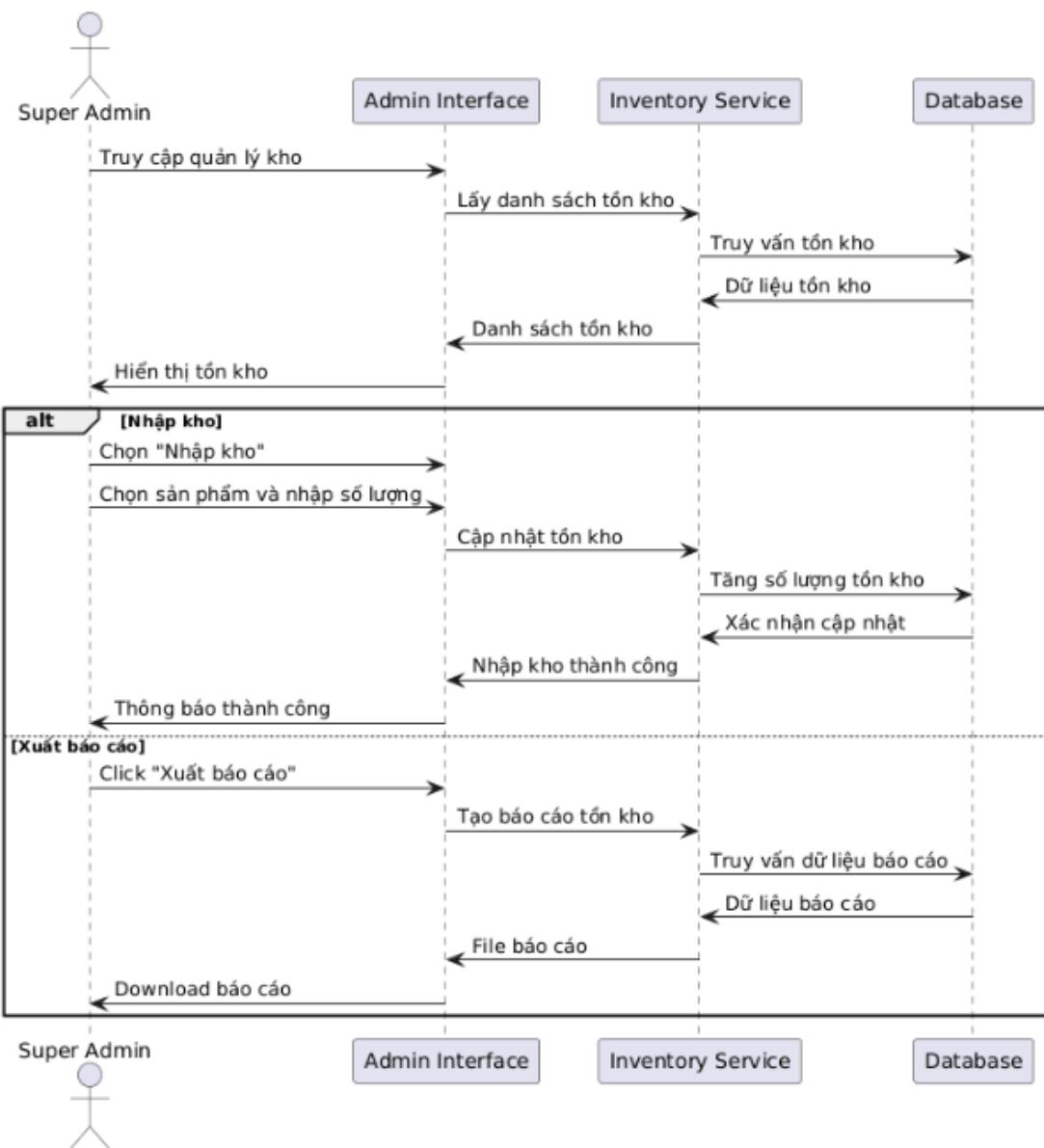
Hình 2.51: Biểu đồ trình tự - Thêm sản phẩm mới

Sequence Diagram - Sửa sản phẩm

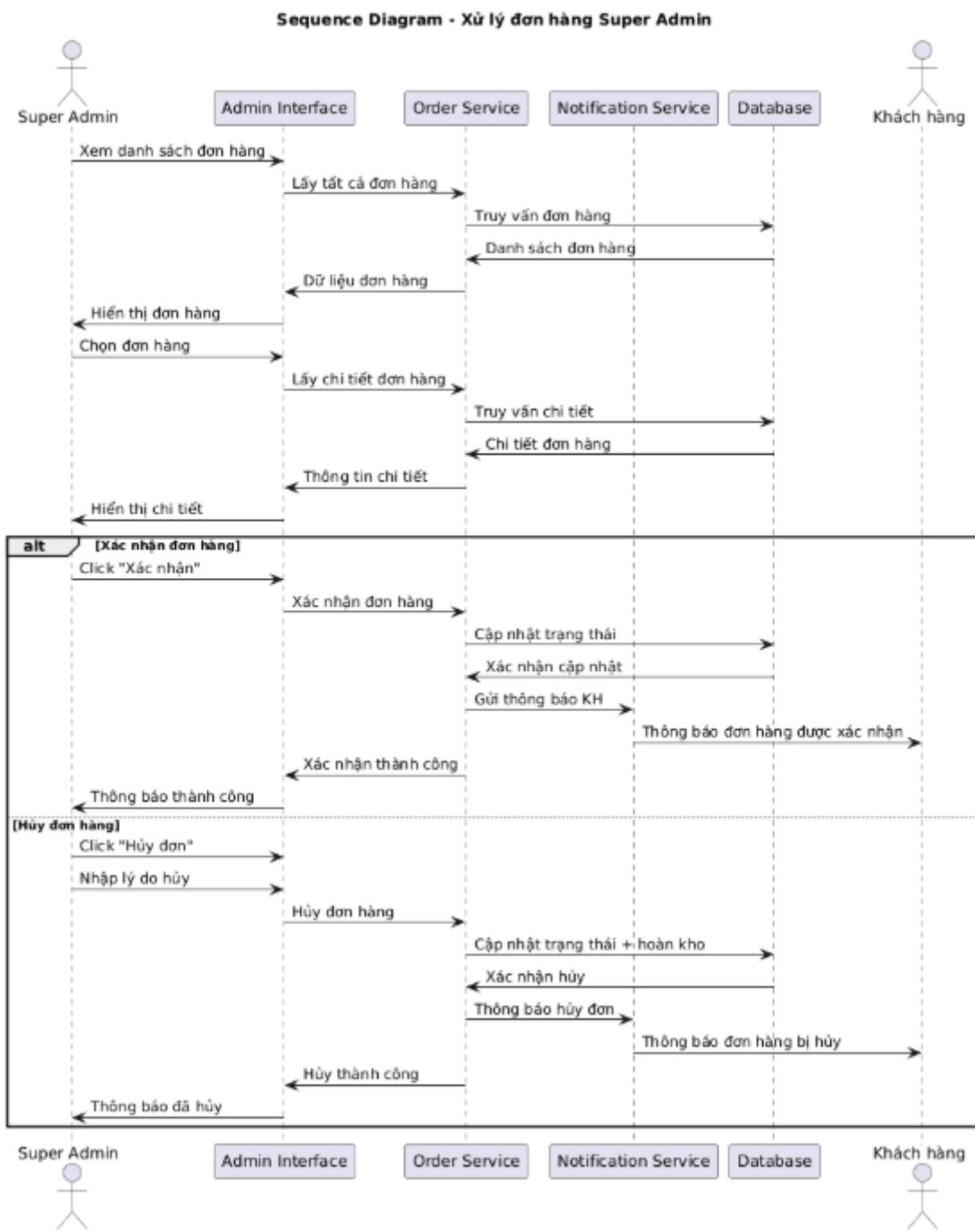


Hình 2.52: Biểu đồ trình tự - Sửa thông tin sản phẩm

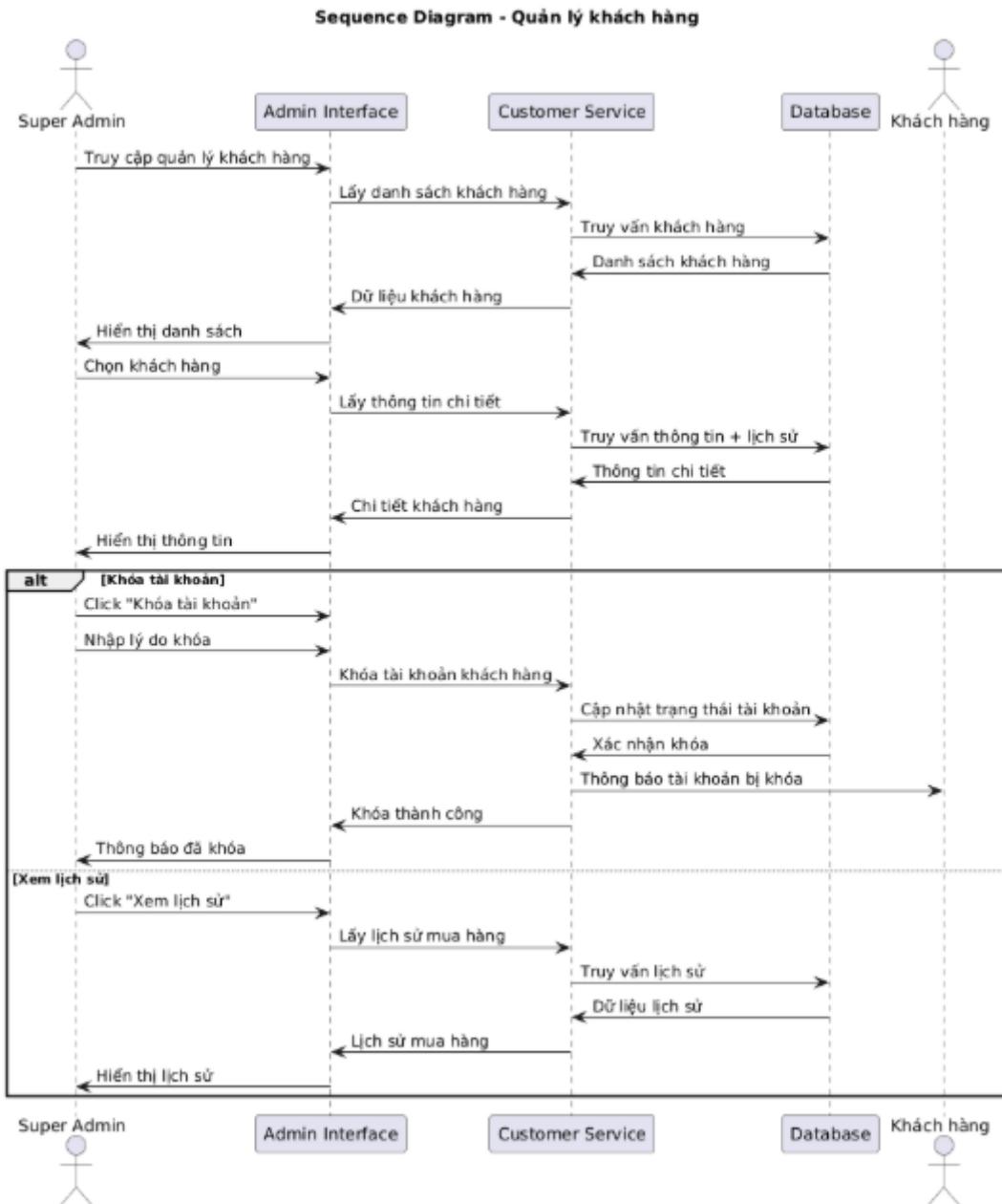
Sequence Diagram - Quản lý kho



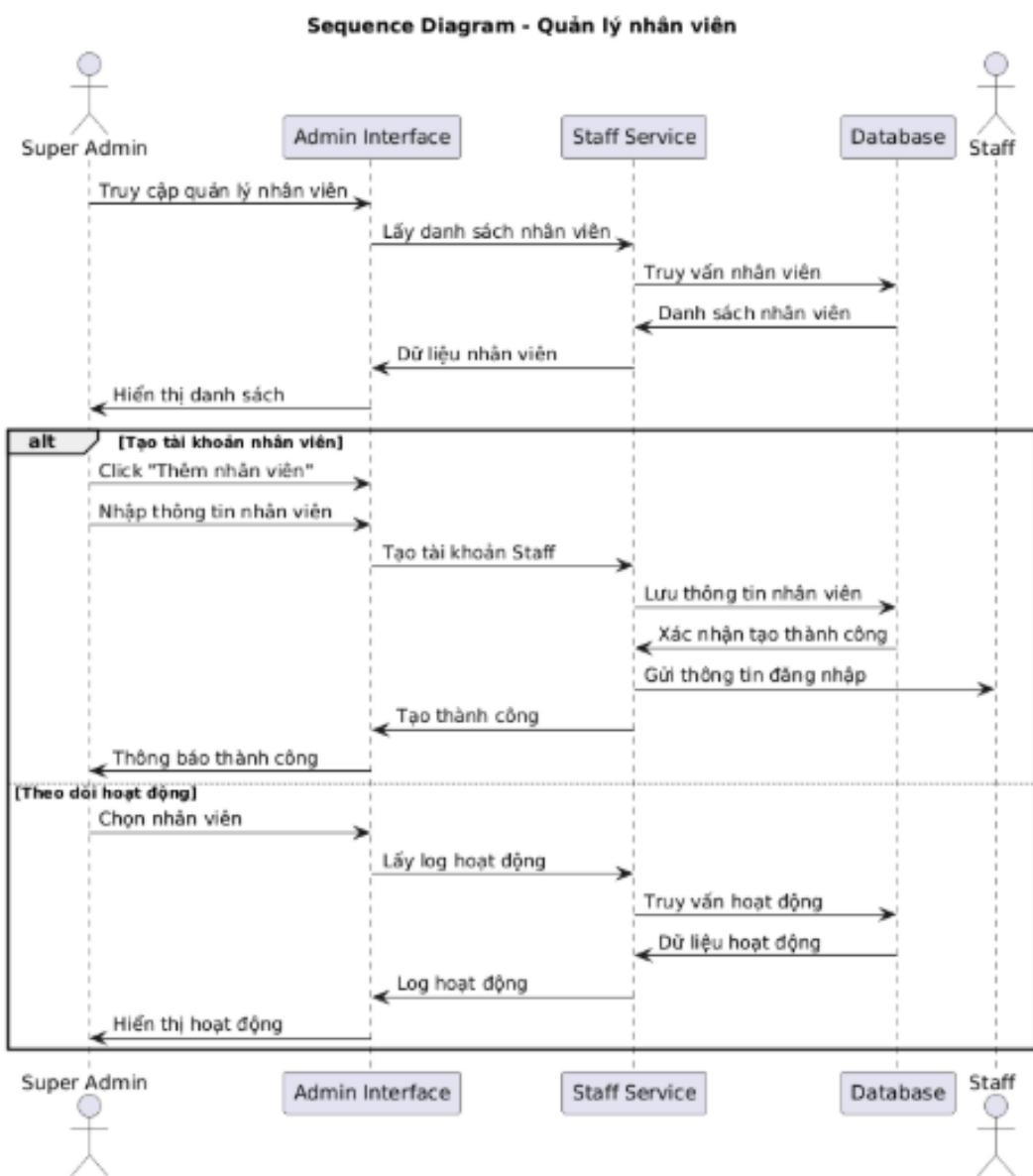
Hình 2.53: Biểu đồ trình tự - Quản lý kho hàng



Hình 2.54: Biểu đồ trình tự - Xử lý đơn hàng (Super Admin)

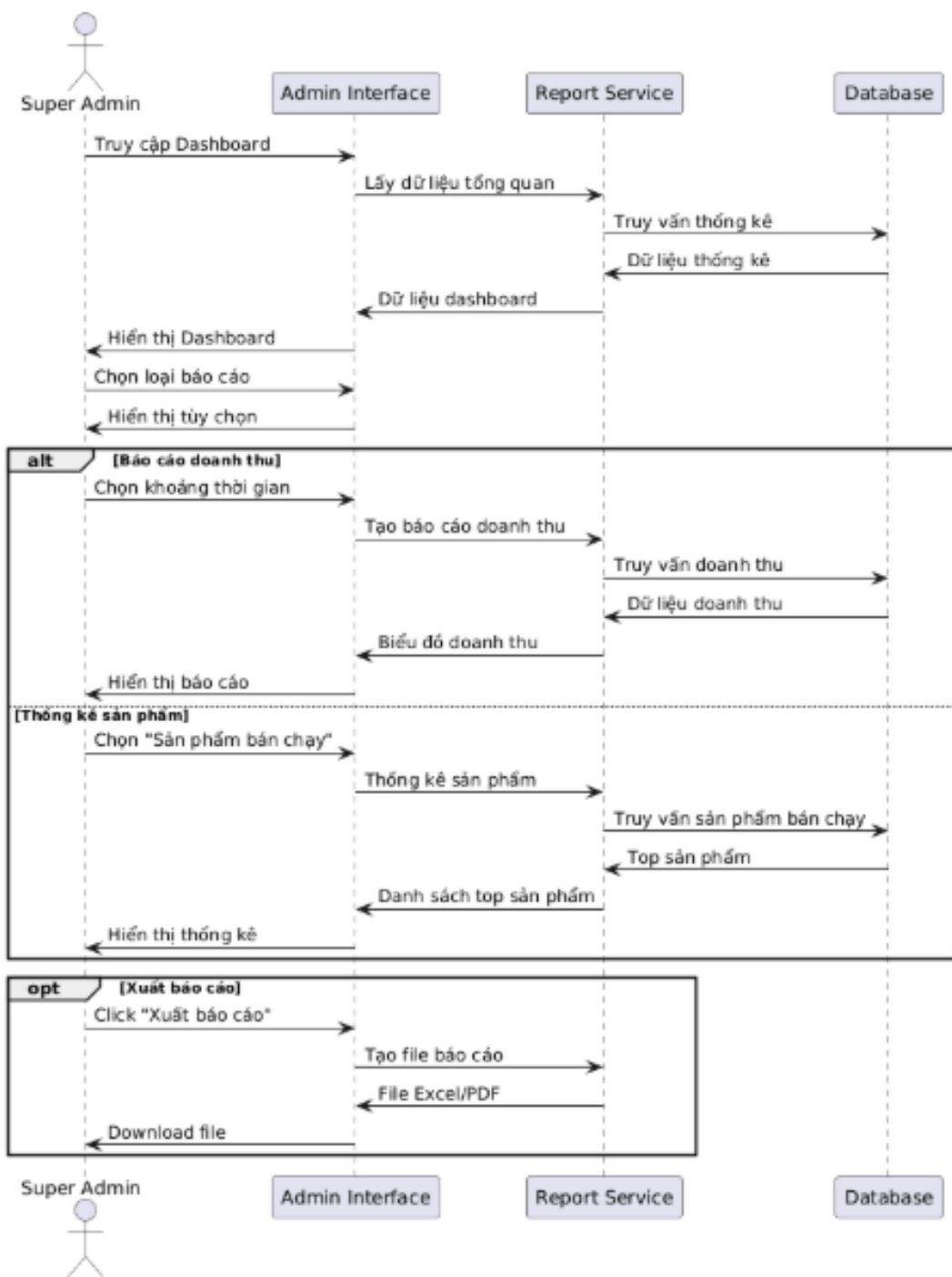


Hình 2.55: Biểu đồ trình tự - Quản lý khách hàng



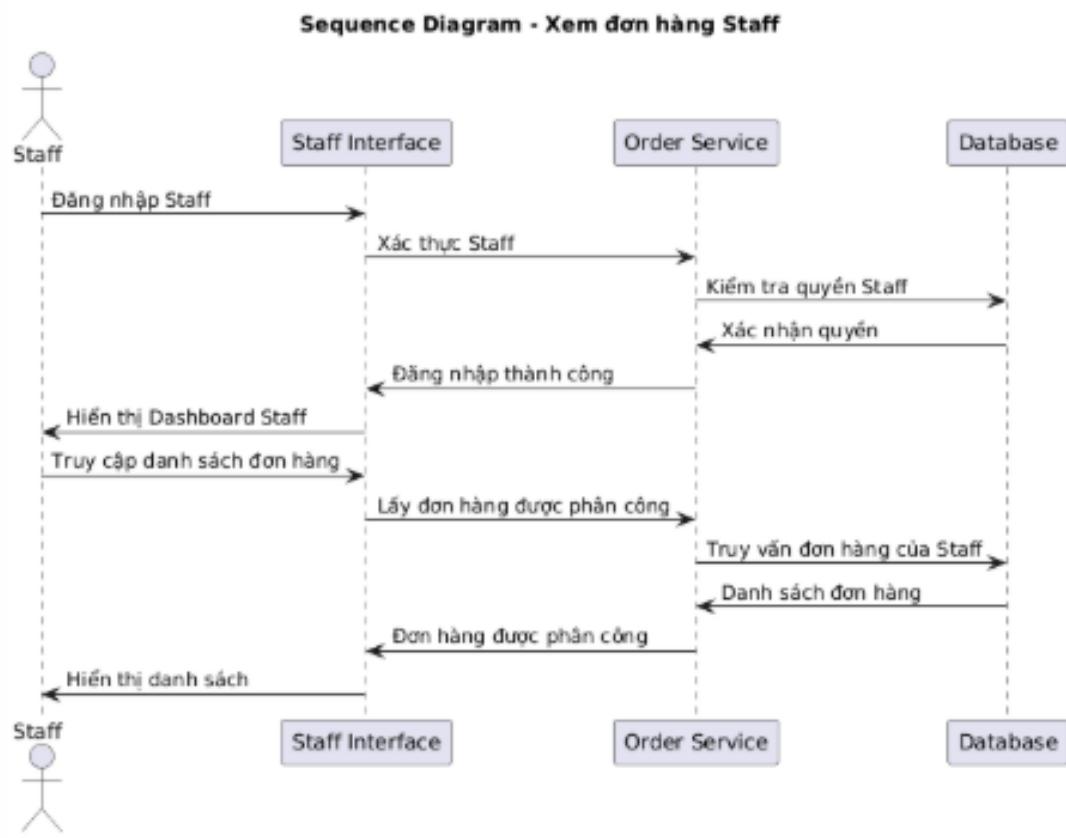
Hình 2.56: Biểu đồ trình tự - Quản lý nhân viên

Sequence Diagram - Báo cáo thống kê

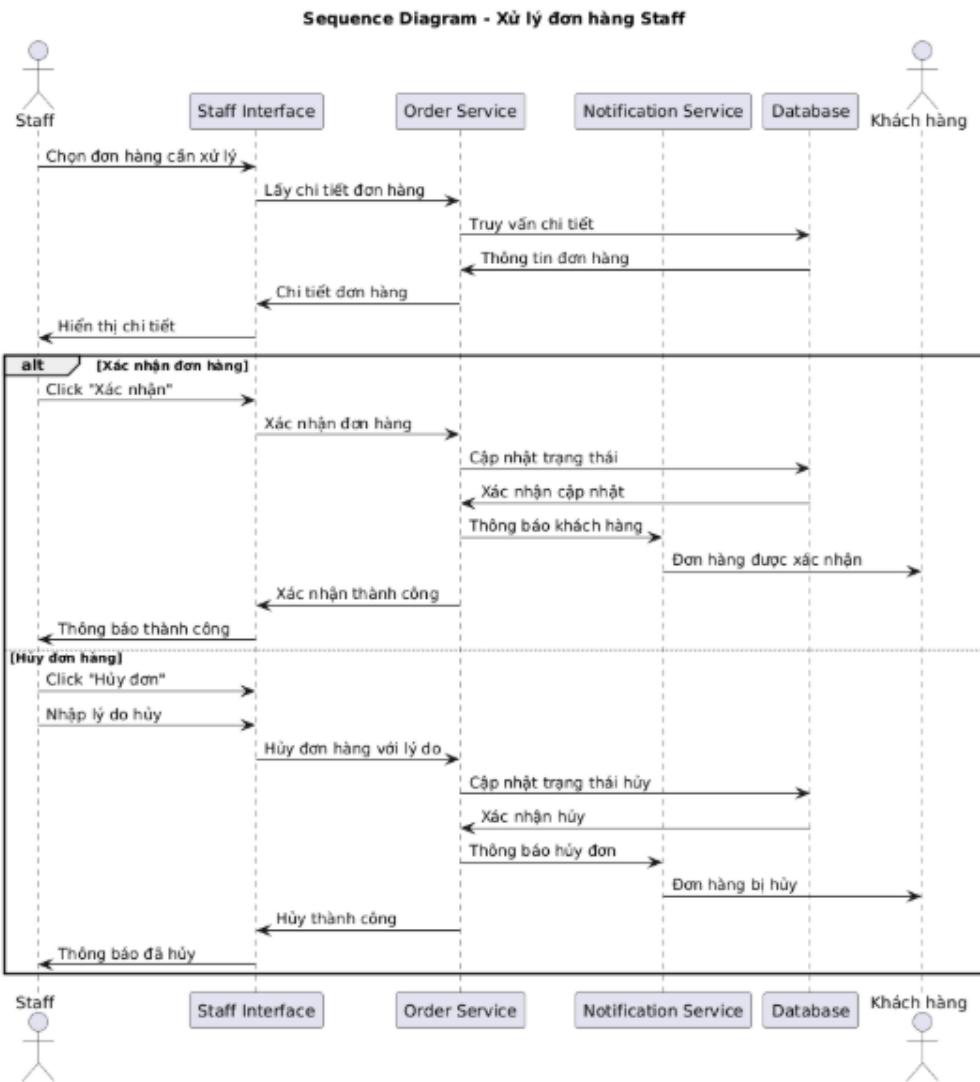


Hình 2.57: Biểu đồ trình tự - Xem báo cáo thống kê

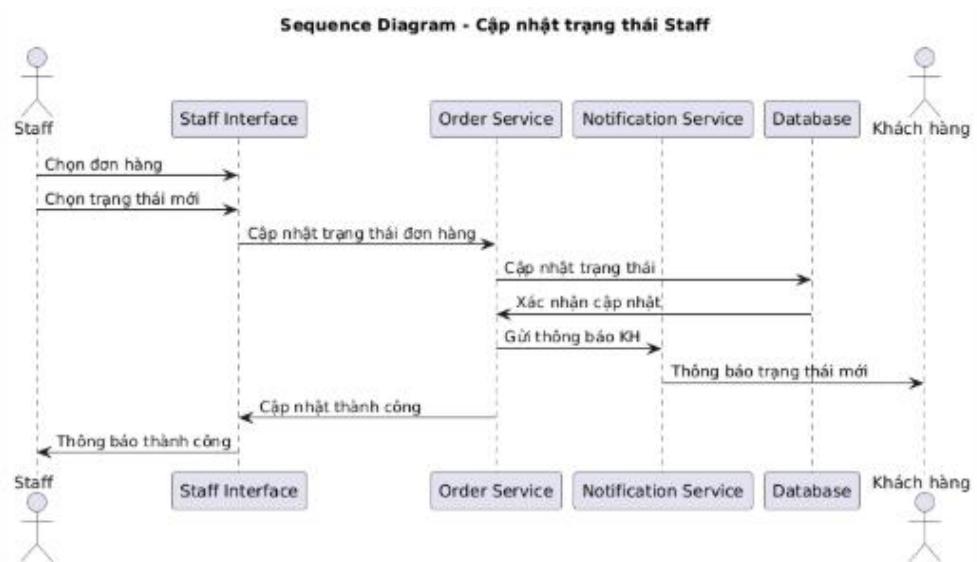
B. Staff



Hình 2.58: Biểu đồ trình tự - Xem đơn hàng được phân công (Staff)



Hình 2.59: Biểu đồ trình tự - Xử lý đơn hàng (Staff)

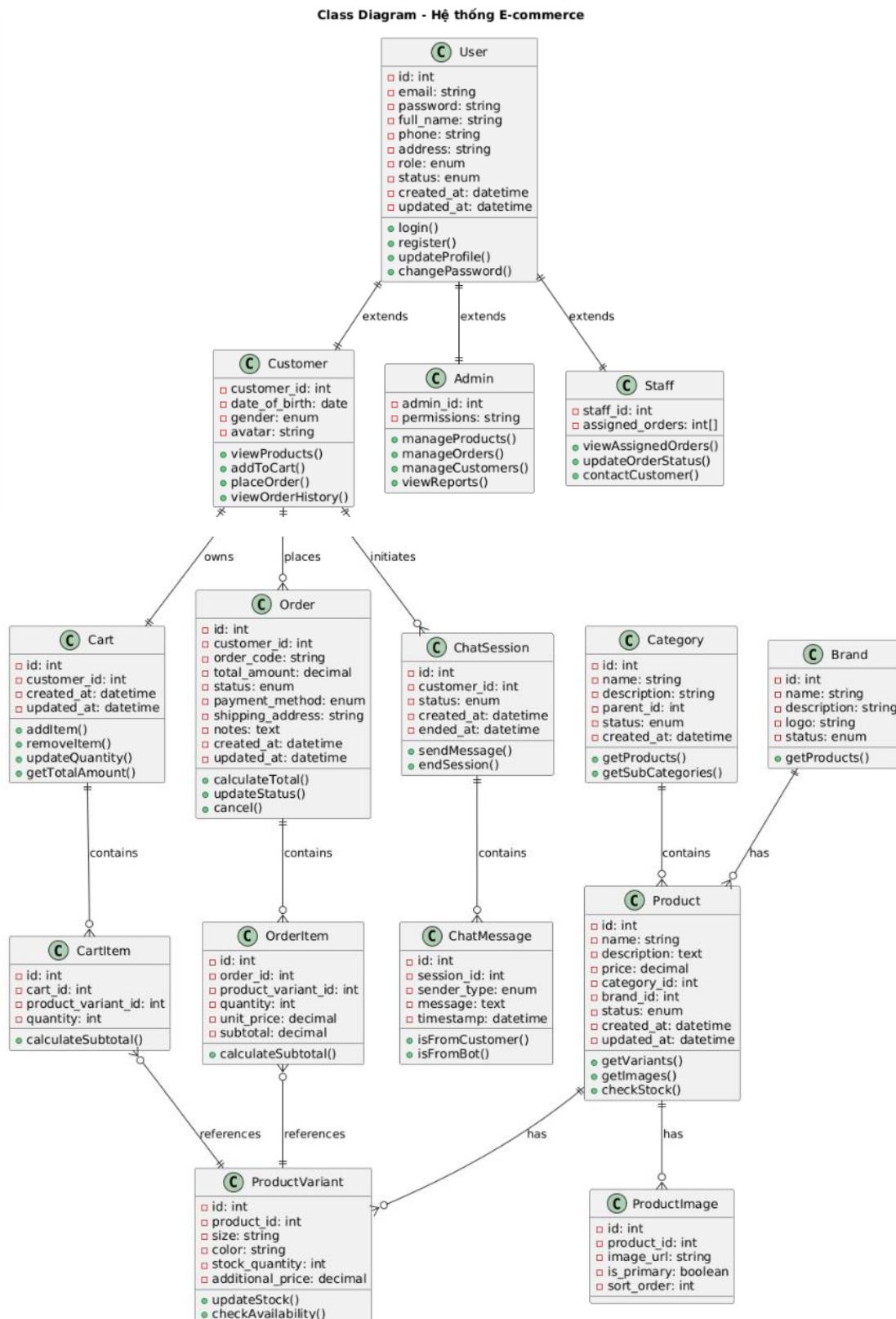


Hình 2.60: Biểu đồ trình tự - Cập nhật trạng thái đơn hàng (Staff)

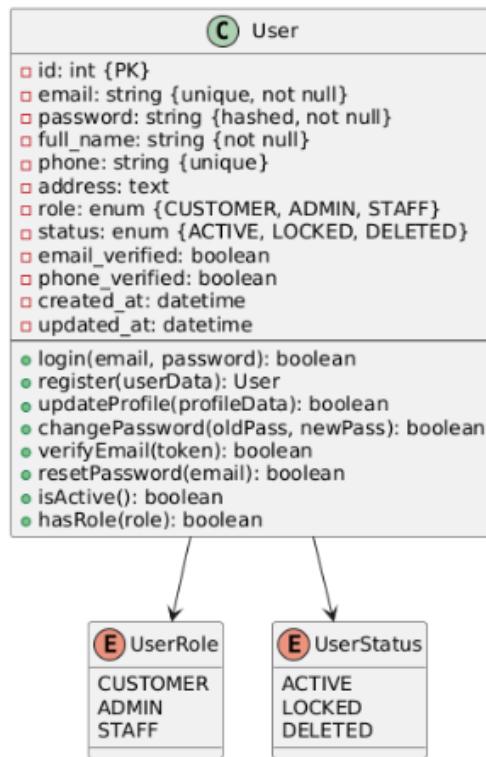
CHƯƠNG 3: Xây dựng hệ thống

3.1. Xây dựng cơ sở dữ liệu

3.1.1. Các lớp và mô tả hệ thống

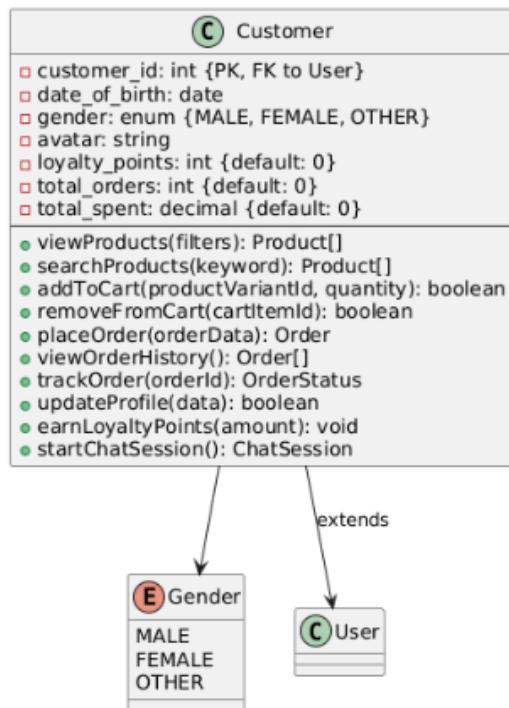


Class User - Chi tiết



Bảng 3.1: Lớp User (Người dùng cơ bản)

Class Customer - Chi tiết



Bảng 3.2: Lớp Customer (Khách hàng)

Class Admin - Chi tiết



Bảng 3.3: Lớp Admin (Quản trị viên)

Class Staff - Chi tiết



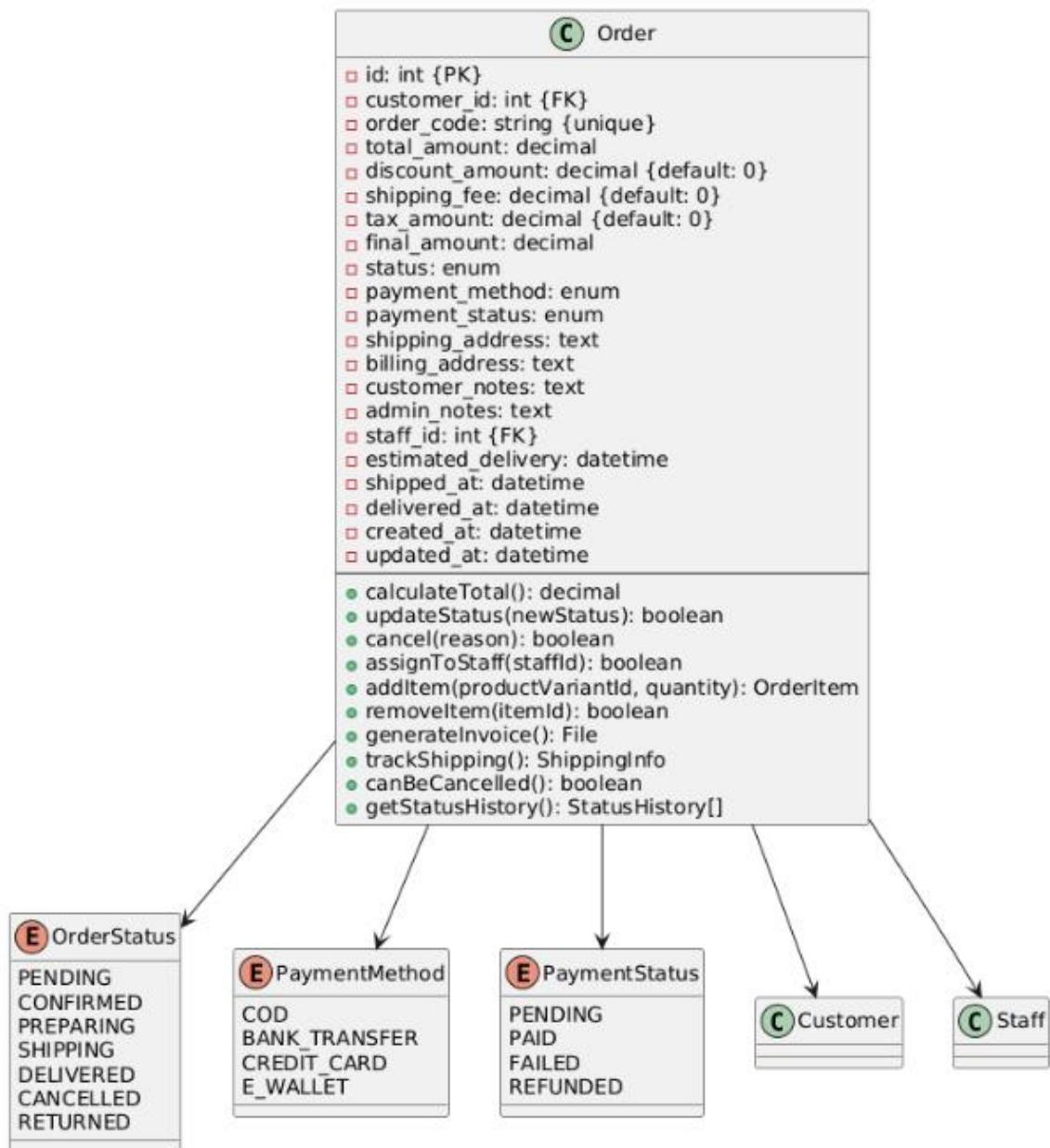
Bảng 3.4: Lớp Staff (Nhân viên)

Class Product - Chi tiết



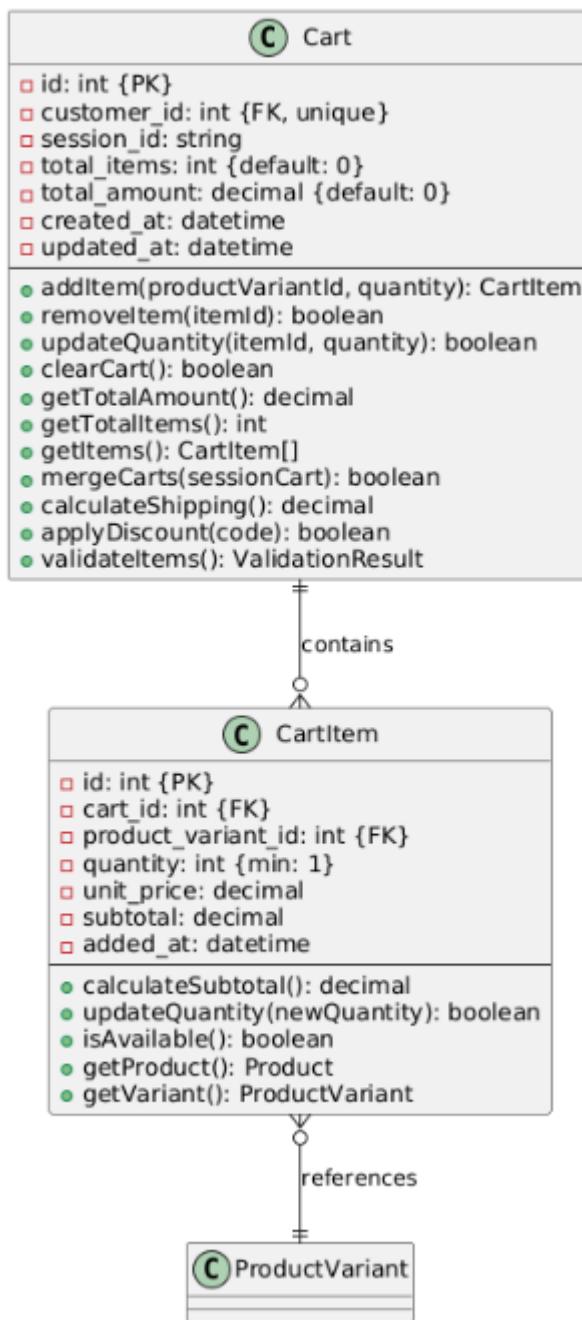
Bảng 3.5: Lớp Product (Sản phẩm)

Class Order - Chi tiết



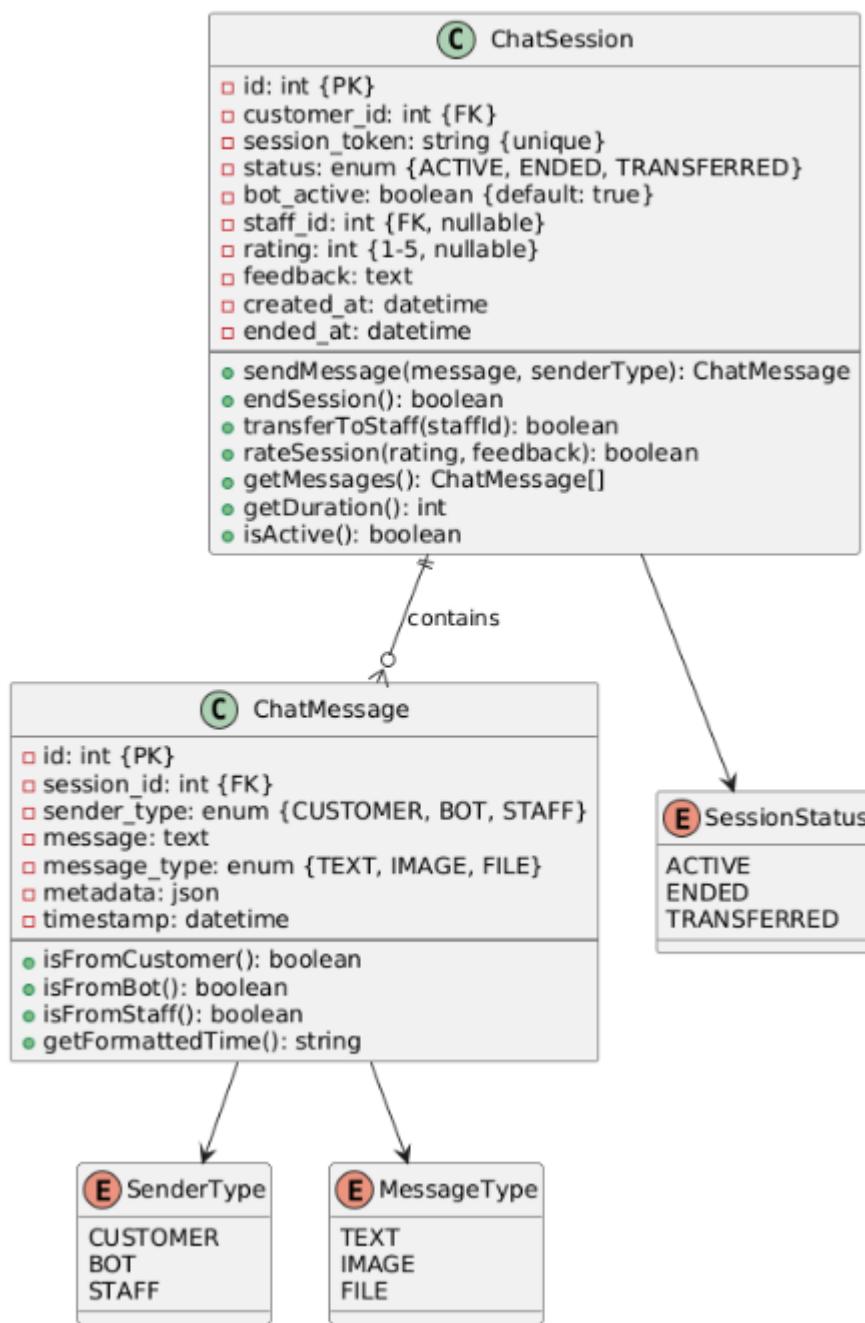
Bảng 3.6: Lớp Order (Đơn hàng)

Class Cart - Chi tiết



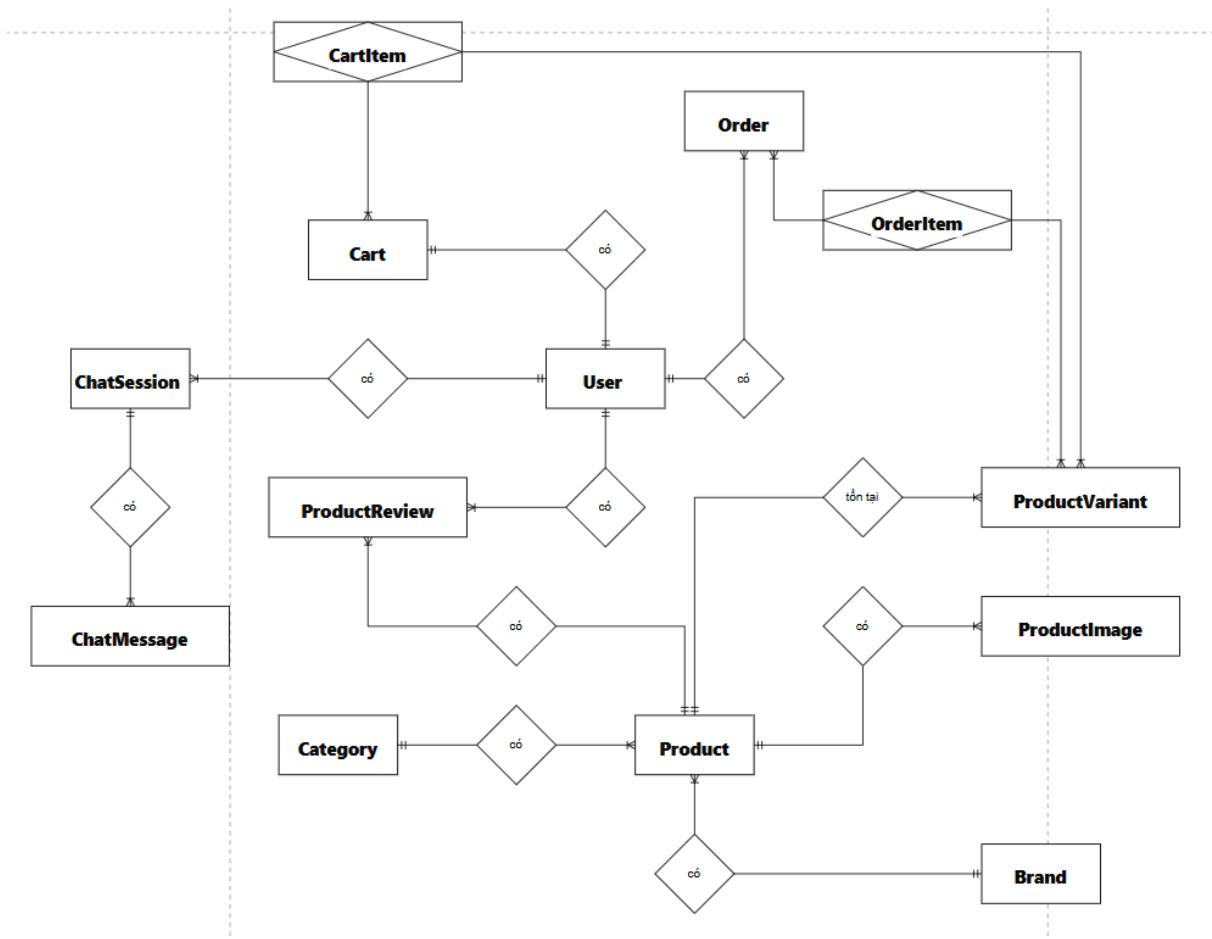
Bảng 3.7: Lớp Cart (Giỏ hàng)

Class ChatSession - Chi tiết



Bảng 3.8: Lớp ChatSession (Phiên chat)

3.1.2. Sơ đồ ERD



Hình 3.1: Sơ đồ ERD

3.2. Xây dựng giao diện và chức năng hệ thống

3.2.1. Giao diện người dùng

3.2.2. Giao diện quản trị viên

TÀI LIỆU THAM KHẢO

- [1] NodeJS: <https://nodejs.org/fr/docs>
- [2] ReactJS: <https://react.dev/learn>
- [3] SQL Server: <https://learn.microsoft.com/en-us/sql/sql-server>
- [4] Client – Server: https://en.wikipedia.org/wiki/Client–server_model