



ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA TOÁN – TIN HỌC

Chuyên ngành Toán Tin Ứng dụng

LUẬN VĂN TỐT NGHIỆP

SMARTSHOP: TRANG WEB THƯƠNG MẠI ĐIỆN TỬ THÔNG MINH

Giảng viên hướng dẫn:

TS. Trần Anh Tuấn

Sinh viên thực hiện:

Vũ Thụy Bích Ngọc - 20110254

Phan Hữu Tình - 20110326

TP. Hồ Chí Minh, tháng 07 năm 2024

LỜI CẢM ƠN

Chúng em xin gửi lời cảm ơn chân thành đến khoa Toán-Tin học, trường Đại học Khoa học tự nhiên vì sự hỗ trợ, tạo mọi điều kiện tốt nhất trong suốt quá trình thực hiện luận văn tốt nghiệp này. Hệ thống thư viện đa dạng sách và tài liệu đã cung cấp cho chúng em một nguồn tài nguyên quý giá để nghiên cứu và tìm kiếm thông tin.

Chúng em cũng xin tỏ lòng kính trọng và lời cảm ơn sâu sắc đến giảng viên hướng dẫn - Thầy Trần Anh Tuấn đã dành thời gian và tận tâm chỉ dẫn cho chúng em hàng tuần. Kiến thức mà chúng em đã được học từ Thầy đã giúp chúng em nắm vững kiến thức và áp dụng chúng vào bài báo cáo này.

Chúng em nhận thức rằng, trong một khoảng thời gian tương đối ngắn với kiến thức và kinh nghiệm của chúng em thì bài luận văn này có thể không tránh khỏi những thiếu sót, hạn chế về mặt kiến thức và khả năng diễn đạt. Chúng em rất mong nhận được sự nhận xét, ý kiến đóng góp và phê bình từ các Thầy/Cô để luận văn tốt nghiệp của chúng em được hoàn thiện hơn nữa.

Cuối cùng, chúng em xin cảm ơn gia đình, người thân, thầy cô và bạn bè vì đã luôn đồng hành với chúng em trong suốt hành trình 4 năm học tập vừa qua. Chúng em xin gửi lời chúc sức khỏe đến tất cả mọi người.

Chúng em xin chân thành cảm ơn!

LỜI MỞ ĐẦU

Internet đã trở thành một phần không thể thiếu trong cuộc sống hiện đại, và tầm ảnh hưởng của nó lan rộng trong mọi lĩnh vực. Đối với giáo dục, kinh doanh, và cuộc sống hàng ngày, Internet đã tạo ra những thay đổi đáng kể như giúp quản lý, tương tác với học sinh trở nên hiệu quả hơn, mà còn mở ra cánh cửa cho học sinh tiếp cận đa dạng nguồn tài liệu trực tuyến. Đối với doanh nghiệp, Internet mang lại nhiều lợi ích to lớn như tiếp cận thị trường rộng lớn, tiết kiệm chi phí quảng cáo, tăng cường tương tác và giao tiếp với khách hàng, ... Nó đã đưa chúng ta vào một kỷ nguyên mới - kỷ nguyên của công nghệ số.

Ở Việt Nam, sự quan tâm đến kinh doanh trực tuyến đã tăng lên đáng kể từ cả người dân thông thường đến các doanh nghiệp lớn. Đặc biệt, thương mại điện tử được xem là một giải pháp kinh tế hiệu quả và là một phần không thể thiếu trong hoạt động kinh doanh hiện nay. Có thể thấy, thương mại điện tử đã thay đổi cách mọi người mua sắm và thực hiện các giao dịch so với trước đây. Điều này mang lại nhiều lợi ích cho xã hội.

Hiện nay, có nhiều lý do khiến các doanh nghiệp và công ty quyết định tập trung vào việc sử dụng thương mại điện tử để cạnh tranh, tồn tại và phát triển bền vững. Ở Việt Nam, số lượng các công ty sử dụng thương mại điện tử để kinh doanh cả trong và ngoài nước đang gia tăng hằng năm.

Tuy nhiên, các doanh nghiệp sẽ gặp phải nhiều khó khăn và thách thức. Cho đến nay, việc doanh nghiệp hay người bán thông thường sử dụng các công nghệ thương mại điện tử cho việc kinh doanh của cá nhân, tổ chức chưa đủ kiến thức và kỹ năng để xây dựng và vận hành website đúng cách. Vì thế, chúng em quyết định chọn đề tài “Xây dựng website thương mại điện tử” để tập trung vào việc thiết kế và biết cách vận hành một website. Kết quả, chúng em sẽ hiểu được cách mà thương mại điện tử hoạt động cũng như các phương pháp kinh doanh trực tiếp và công nghệ được sử dụng trong website.

DANH SÁCH CHỮ VIẾT TẮT

	English	Tiếng Việt
TMĐT	E-commerce	Thương mại điện tử
AWS	Amazon Web Services	Tên nhà cung cấp dịch vụ đám mây
B2B	Business to Business	Doanh nghiệp với Doanh nghiệp
B2C	Business to Customer	Doanh nghiệp với Khách hàng
C2C	Customer to Customer	Khách hàng với Khách hàng
REST	Representational State Transfer	Tiêu chuẩn thiết kế API
UI	User Interface	Giao diện người dùng
API	Application Programmer Interface	Giao diện lập trình ứng dụng
ERD	Entity Relationship Diagram	Mô hình thực thể kết hợp
BFD	Business Function Diagram	Mô hình phân rã chức năng
DFD	Data Flow Diagram	Mô hình dòng dữ liệu
JSON	JavaScript Object Notation	Một kiểu dữ liệu trong JavaScript
SPA	Single Page Applications	Ứng dụng web đơn trang
JWT	JSON Web Token	Mã chứng thực
WS	Web Service	Dịch vụ Web
SMS	Short Message Service	Dịch vụ tin nhắn ngắn
HTML	HyperText Markup Language	Ngôn ngữ đánh dấu siêu văn bản
CSS	Cascading Style Sheets	Bảng kiểu lệnh nối tiếp
JSON	JavaScript Object Notation	Định dạng đối tượng JavaScript
XML	eXtensible Markup Language	Ngôn ngữ đánh dấu mở rộng
URI	Uniform Resource Identifier	Định danh tài nguyên thống nhất
RDBMS	Relational Database Management System	Hệ thống quản lý cơ sở dữ liệu quan hệ
UX	User Experience	Trải nghiệm người dùng

MỤC LỤC

LỜI CẢM ƠN	1
LỜI MỞ ĐẦU	2
DANH SÁCH CHỮ VIẾT TẮT	3
CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI	10
1.1. Giới thiệu bài toán.....	10
1.2. Mục tiêu và nhiệm vụ	11
1.3. Tính thực tiễn của đề tài.....	11
1.4. Hướng giải quyết	12
1.5. Bố cục luận văn.....	13
CHƯƠNG 2: CÔNG NGHỆ ÁP DỤNG, KỸ THUẬT LIÊN QUAN	15
2.1. Giới thiệu về thương mại điện tử	15
2.1.1. Định nghĩa	15
2.1.2. Lợi ích và khó khăn của thương mại điện tử	15
2.1.3. Các loại hình giao dịch trong thương mại điện tử	16
2.2. Kiến thức hệ thống.....	19
2.2.1. Mô hình RESTful Web Service.....	19
2.2.2. Kiến trúc Microservice và Monolithic	21
2.3. Các công nghệ tiên tiến cho phát triển và quản lý ứng dụng.....	23
2.3.1. Spring Boot 3.....	23
2.3.2. Bootstrap 5.....	24
2.3.3. PostgreSQL.....	25
2.3.4. DBeaver	26
2.3.5. Docker	26
2.3.6. IntelliJ IDEA.....	28
2.3.7. Nginx	30
2.4. Các kỹ thuật và công cụ liên quan	31
2.4.1. Thanh toán trực tuyến VNPay	31
2.4.2. Spring Security	33
2.4.3. Spring Data JPA	35
2.4.4. Thymeleaf	36
2.5. Các công cụ hỗ trợ kiểm thử	36
2.5.1. Postman	36
2.5.2. Apache JMeter	37
CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG.....	39
3.1. Đặc tả hệ thống	39
3.1.1. Nhiệm vụ	39
3.1.2. Quy trình nghiệp vụ	39
3.1.3. Tổng quan hệ thống (Big Picture)	40
3.2. Phân tích hệ thống.....	42
3.2.1. Yêu cầu ứng dụng.....	42

3.2.2. Yêu cầu người dùng.....	42
3.3. Hạ tầng triển khai ứng dụng.....	43
3.3.1. Docker compose	43
3.3.2. VPS (Virtual Private Server)	43
3.4. Thiết kế hệ thống	45
3.4.1. Mô hình cơ sở dữ liệu.....	45
3.4.2. Sơ đồ Use Case (Use Case diagram)	49
3.4.3. Sơ đồ tuần tự (Sequence diagram)	58
3.4.4. Mô hình phân rã chức năng (BFD).....	71
3.4.5. Mô hình dòng dữ liệu (DFD).....	74
CHƯƠNG 4: KẾT QUẢ CHƯƠNG TRÌNH	83
4.1. Quy trình đăng nhập và đăng ký	83
4.2. Quy trình quản lý và hiển thị sản phẩm	84
4.3. Quy trình đặt hàng và thanh toán	88
4.4. Quy trình quản lý cửa hàng.....	90
4.5. Quá trình kiểm thử	92
CHƯƠNG 5: TỔNG KẾT VÀ PHÁT TRIỂN ĐỀ TÀI.....	97
5.1. Những đóng góp của đề tài	97
5.2. Hướng phát triển	97
TÀI LIỆU THAM KHẢO	99

DANH SÁCH BẢNG

Bảng 3.3.1.2.1. CartItem	45
Bảng 3.3.1.2.2. Category	46
Bảng 3.3.1.2.3. Color	46
Bảng 3.3.1.2.4. User.....	46
Bảng 3.3.1.2.5. Image	46
Bảng 3.3.1.2.6. Product.....	46
Bảng 3.3.1.2.7. Order.....	47
Bảng 3.3.1.2.8. Size	47
Bảng 3.3.1.2.9. ShoppingCart.....	47
Bảng 3.3.2.1. Đặc tả Use Case tổng quát hệ thống.....	50
Bảng 3.3.2.2. Đặc tả Use Case Quản lý sản phẩm của quản trị viên.....	52
Bảng 3.3.2.3. Đặc tả Use Case Quản lý đơn hàng của quản trị viên	52
Bảng 3.3.2.4. Đặc tả Use Case Quản lý tài khoản của quản trị viên	53
Bảng 3.3.2.5. Đặc tả Use Case Quản lý vận chuyển của quản trị viên.....	54
Bảng 3.3.2.6. Đặc tả Use Case Tìm kiếm sản phẩm của khách hàng.....	55
Bảng 3.3.2.7. Đặc tả Use Case Quản lý hồ sơ cá nhân của khách hàng.....	56
Bảng 3.3.2.8. Đặc tả Use Case Đơn hàng của khách hàng	57
Bảng 3.3.2.9. Đặc tả Use Case Giờ hàng của khách hàng	58

DANH SÁCH HÌNH ẢNH

Hình 2.2.1.1. Mô hình Web service.....	19
Hình 2.2.1.2. RESTful API.....	20
Hình 2.2.2.1. Kiến trúc Microservice	21
Hình 2.2.2.2. Kiến trúc Monolithic	23
Hình 2.3.5.1. Kiến trúc Docker	27
Hình 2.3.6.1. Những tính năng nổi bật của IntelliJ IDEA	28
Hình 2.3.7.1. Minh họa cách Nginx hoạt động.....	30
Hình 2.4.1. Sơ đồ tuần tự các bước Thanh toán trực tuyến VNPay	32
Hình 2.4.2.1.1. Chuyển hướng đến trang Login.....	33
Hình 2.4.2.1.2. Xử lý xác thực bởi username và password	34
Hình 2.4.2.1.3. Quy trình sử lý DaoAuthenticationProvider.....	34
Hình 2.5.2.1.1. Môi trường thử nghiệm phân tán (Distributed test environment)	37
Hình 3.1.1. Tổng quan hệ thống	40
Hình 3.1.2. Cách thức hoạt động của Docker Compose.....	43
Hình 3.1.3. Triển khai VPS với Docker	44
Hình 3.3.1.3.1. Sơ đồ ERD thể hiện các mối quan hệ trong hệ thống.....	48
Hình 3.3.2.1. Sơ đồ Use Case tổng quát hệ thống	50
Hình 3.3.2.2. Sơ đồ Use Case tổng quát của quản trị viên	51
Hình 3.3.2.3. Sơ đồ Use Case Quản lý sản phẩm của quản trị viên	51
Hình 3.3.2.4. Sơ đồ Use Case Quản lý đơn hàng của quản trị viên	52
Hình 3.3.2.5. Sơ đồ Use Case Quản lý tài khoản của quản trị viên	53
Hình 3.3.2.6. Sơ đồ Use Case Quản lý vận chuyển của quản trị viên.....	54
Hình 3.3.2.7. Sơ đồ Use Case Tổng quát của khách hàng.....	55
Hình 3.3.2.8. Sơ đồ Use Case Tìm kiếm sản phẩm của khách hàng	55
Hình 3.3.2.9. Sơ đồ Use Case Quản lý hồ sơ cá nhân của khách hàng	56
Hình 3.3.2.10. Sơ đồ Use Case Đơn hàng của khách hàng	57
Hình 3.3.2.11. Sơ đồ Use Case Giỏ hàng của khách hàng	58
Hình 3.3.3.1. Biểu đồ tuần tự Đăng ký	59
Hình 3.3.3.2. Biểu đồ tuần tự Đăng nhập	59
Hình 3.3.3.3. Biểu đồ tuần tự Thanh toán	60
Hình 3.3.3.4. Biểu đồ tuần tự Xem thông tin sản phẩm	60
Hình 3.3.3.5. Biểu đồ tuần tự Tìm kiếm sản phẩm.....	61
Hình 3.3.3.6. Biểu đồ tuần tự Giỏ hàng (Xóa sản phẩm)	61
Hình 3.3.3.7. Biểu đồ tuần tự Giỏ hàng (Cập nhật số lượng).....	62

Hình 3.3.3.8. Biểu đồ tuần tự Quản lý hồ sơ cá nhân	62
Hình 3.3.3.9. Biểu đồ tuần tự Đơn hàng	63
Hình 3.3.3.10. Biểu đồ tuần tự Đăng nhập trang quản trị viên	64
Hình 3.3.3.11. Biểu đồ tuần tự Quản lý sản phẩm (Xóa sản phẩm)	64
Hình 3.3.3.12. Biểu đồ tuần tự Quản lý sản phẩm (Thêm sản phẩm)	65
Hình 3.3.3.13. Biểu đồ tuần tự Quản lý đơn hàng (Cập nhật sản phẩm)	66
Hình 3.3.3.14. Biểu đồ tuần tự Quản lý đơn hàng (Xóa đơn hàng)	67
Hình 3.3.3.15. Biểu đồ tuần tự Quản lý đơn hàng (Cập nhật đơn hàng)	68
Hình 3.3.3.16. Biểu đồ tuần tự Quản lý tài khoản (Xóa tài khoản)	69
Hình 3.3.3.17. Biểu đồ tuần tự Quản lý tài khoản (Cập nhật tài khoản)	70
Hình 3.3.3.18. Biểu đồ tuần tự Quản lý tài khoản (Thêm tài khoản)	70
Hình 3.3.4.1. Sơ đồ phân rã chức năng Hệ thống quản lý bán sản phẩm	71
Hình 3.3.4.2. Sơ đồ phân rã chức năng chi tiết của Quản trị viên (phần 1)	71
Hình 3.3.4.3. Sơ đồ phân rã chức năng chi tiết của Quản trị viên (phần 2)	72
Hình 3.3.4.4. Sơ đồ phân rã chức năng chi tiết của Khách hàng có tài khoản (phần 1)	73
Hình 3.3.4.5. Sơ đồ phân rã chức năng chi tiết của Khách hàng có tài khoản (phần 2)	73
Hình 3.3.4.6. Sơ đồ phân rã chức năng chi tiết của Khách hàng không có tài khoản	74
Hình 3.3.5.1. Mô hình hóa xử lý ở mức ngữ cảnh	74
Hình 3.3.5.2. Mô hình hóa xử lý ở mức định	75
Hình 3.3.5.3. Sơ đồ luồng dữ liệu chi tiết của Đăng nhập	76
Hình 3.3.5.4. Sơ đồ luồng dữ liệu chi tiết của Quản lý vận chuyển	76
Hình 3.3.5.5. Sơ đồ luồng dữ liệu chi tiết của Quản lý kho	77
Hình 3.3.5.6. Sơ đồ luồng dữ liệu chi tiết của Quản lý sản phẩm	77
Hình 3.3.5.7. Sơ đồ luồng dữ liệu chi tiết của Quản lý tài khoản	78
Hình 3.3.5.8. Sơ đồ luồng dữ liệu chi tiết của Quản lý đơn hàng	78
Hình 3.3.5.9. Sơ đồ luồng dữ liệu chi tiết của Đăng nhập	79
Hình 3.3.5.10. Sơ đồ luồng dữ liệu chi tiết của Quản lý hồ sơ cá nhân	79
Hình 3.3.5.11. Sơ đồ luồng dữ liệu chi tiết của Xem thông tin sản phẩm	79
Hình 3.3.5.12. Sơ đồ luồng dữ liệu chi tiết của Tìm kiếm sản phẩm	80
Hình 3.3.5.13. Sơ đồ luồng dữ liệu chi tiết của Giỏ hàng	80
Hình 3.3.5.14. Sơ đồ luồng dữ liệu chi tiết của Thanh toán	81
Hình 3.3.5.15. Sơ đồ luồng dữ liệu chi tiết của Đơn hàng	81
Hình 3.3.5.16. Sơ đồ luồng dữ liệu chi tiết của Đăng ký	82
Hình 3.3.5.17. Sơ đồ luồng dữ liệu chi tiết của Tìm kiếm sản phẩm	82
Hình 3.3.5.18. Sơ đồ luồng dữ liệu chi tiết của Xem thông tin sản phẩm	82
Hình 4.1.1. Trang đăng nhập của User	83

Hình 4.1.2. Trang đăng ký của User	83
Hình 4.1.3. Trang khảo sát sở thích của khách hàng	84
Hình 4.2.1. Trang chủ	86
Hình 4.2.2. Tìm kiếm sản phẩm	86
Hình 4.2.3. Trang hiển thị sản phẩm	87
Hình 4.2.4. Trang hiển thị sản phẩm sau khi áp dụng bộ lọc	87
Hình 4.2.5. Trang chi tiết sản phẩm (phản chi tiết)	87
Hình 4.2.6. Trang chi tiết sản phẩm (phản sản phẩm đề xuất)	88
Hình 4.3.1. Trang giỏ hàng	88
Hình 4.3.2. Trang đơn hàng	89
Hình 4.3.3. Trang thanh toán VNPAY	89
Hình 4.3.4. Trang thanh toán thành công	90
Hình 4.4.1. Trang chủ cửa hàng	90
Hình 4.4.2. Trang thêm sản phẩm sản phẩm	91
Hình 4.4.3. Trang quản lý sản phẩm	91
Hình 4.4.4. Trang quản lý tài khoản (danh sách khách hàng)	91
Hình 4.4.5. Trang quản lý tài khoản (danh sách quản trị viên)	92
Hình 4.4.6. Trang quản lý đơn hàng	92
Hình 4.4.7. Trang quản lý đơn hàng chi tiết	92
Hình 4.5.1. Api thêm danh sách sản phẩm	93
Hình 4.5.2. Api xóa sản phẩm	93
Hình 4.5.3. Api lấy tất cả sản phẩm	93
Hình 4.5.4. Api lấy sản phẩm cụ thể	94
Hình 4.5.5. Api cập nhật sản phẩm	94
Hình 4.5.6. Kết quả cập nhật sản phẩm với id = 10	94
Hình 4.5.7. Tạo Thread Group	94
Hình 4.5.8. Truy cập đến trang chủ	95
Hình 4.5.9. Kết quả đạt được với View Results Tree	95
Hình 4.5.10. Kết quả đạt được với Graph Results	96

CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

1.1. Giới thiệu bài toán

Trong bối cảnh hiện nay, sự phát triển mạnh mẽ của thương mại điện tử (TMĐT) và việc chuyển đổi sang giao dịch không sử dụng tiền mặt đang trở thành xu hướng không thể phủ nhận. Kết hợp điều này với sự phổ biến của Internet đã mở ra nhiều cơ hội mới cho TMĐT trong những năm gần đây, không chỉ ở Việt Nam mà còn lan rộng trên toàn thế giới.

Thị trường TMĐT đang ngày càng mở rộng cùng với sự đổi mới liên tục trong các mô hình kinh doanh, chuỗi cung ứng thông minh, hiệu quả hơn, nhờ vào sự lan tỏa của công nghệ thông tin. Shopee - một trong những cột mốc đáng kể của thị trường TMĐT, là một nền tảng TMĐT hàng đầu tại Việt Nam. Shopee đã trở thành điểm đến không thể thiếu, nơi mà cung cấp cho khách hàng trải nghiệm mua sắm trực tuyến toàn diện.

Trang web này sẽ giúp cho khách hàng có thể mua sắm dễ dàng và nhanh chóng bất cứ đâu chỉ cần có Internet. Hơn thế nữa, khách hàng có thể dễ dàng tìm thấy được đa dạng các sản phẩm từ nhiều ngành hàng khác nhau, từ đó so sánh giá cả và lựa chọn sản phẩm với giá cả hợp lý, điều này giúp tiết kiệm nhiều thời gian và đáp ứng được nhu cầu mua sắm tức thì. Hệ thống tìm kiếm linh hoạt, dễ sử dụng và giao diện thân thiện giúp người dùng dễ dàng thao tác trên trang web, chỉ cần một vài thao tác đơn giản là khách hàng có thể tự do mua sắm và đặt hàng một cách tự tin.

Một điểm đặc biệt của hệ thống chúng em đó là khả năng thu thập sở thích người dùng ngay từ ban đầu để lọc những sản phẩm phù hợp cho khách hàng.

Thay vì sử dụng AI có nhiều ưu điểm như khả năng phân tích dữ liệu lớn và học từ hành vi người dùng để đưa ra các đề xuất. Nhiều trang web thương mại điện tử hiện nay sử dụng AI để cải thiện trải nghiệm người dùng. Nhưng vấn đề ở đây là một khi AI gặp sai sót trong việc phân tích dữ liệu hoặc đưa ra các đề xuất không phù hợp, việc điều chỉnh sẽ phức tạp và tốn nhiều thời gian. Điều này có thể ảnh hưởng nghiêm trọng đến trải nghiệm của người dùng và trong trường hợp xấu nhất, khách hàng có thể rời bỏ chúng ta trước khi chúng ta kịp nhận ra và khắc phục vấn đề. Do đó, phương pháp thu thập dữ liệu ngay từ đầu là một giải pháp đầy triển vọng với các ưu điểm sau:

- **Đơn giản và hiệu quả**

Bằng cách thu thập sở thích người dùng ngay từ đầu, chúng em có thể cung cấp các đề xuất chính xác và phù hợp ngay lập tức. Người dùng nhanh chóng tìm thấy sản phẩm họ quan tâm mà không cần phải "dạy" AI qua thời gian dài.

- **Tránh rủi ro sai sót**

Phương pháp này giảm thiểu rủi ro sai sót trong quá trình phân tích dữ liệu, giúp hệ thống hoạt động ổn định và đáng tin cậy hơn.

- *Trải nghiệm cá nhân hóa*

Người dùng cảm thấy được tôn trọng và quan tâm khi hệ thống hiểu rõ và đáp ứng đúng sở thích của họ ngay từ lần đầu tiên truy cập.

1.2. Mục tiêu và nhiệm vụ

Trong quá trình nghiên cứu dự án này, chúng em đã đặt ra các mục tiêu như sau:

- Hiểu rõ về bản chất và ứng dụng của thương mại điện tử.
- Xây dựng một trang web TMĐT hoàn chỉnh sử dụng các công nghệ như PostgreSQL cho cơ sở dữ liệu, Java cho phần back-end, và HTML, CSS, JavaScript cho phần front-end.
- Tạo ra một giao diện người dùng thân thiện và thú vị, tối ưu hóa trải nghiệm mua sắm trực tuyến cho người dùng.
- Áp dụng kiến trúc Monolithic trong việc phát triển hệ thống giúp đơn giản hóa việc quản lý và triển khai ứng dụng.
- Nghiên cứu và triển khai các giải pháp thanh toán trực tuyến để cung cấp cho người dùng phương thức thanh toán an toàn, nhanh chóng và tiện lợi.
- Nâng vững cơ chế bảo mật của website bằng cách sử dụng các phương pháp mã hóa dữ liệu và quản lý quyền truy cập của người dùng.
- Khám phá và sử dụng các dịch vụ điện toán đám mây như AWS hoặc Google Cloud Platform để lưu trữ dữ liệu và triển khai ứng dụng.

1.3. Tính thực tiễn của đề tài

Trong thực tế hiện nay, người dân vẫn chọn cách mua hàng hóa hay thực phẩm trực tiếp bởi khi đó họ có thể được trải nghiệm thực tế với sản phẩm đó và chắc chắn với sự lựa chọn của bản thân. Tuy nhiên, điều đó khiến các cửa hàng mua sắm trực tiếp phải đối mặt với sự cạnh tranh gay gắt từ các hệ thống đã áp dụng mua sắm trực tuyến và các nền tảng kinh doanh hiện đại, cũng như áp lực từ các loại thuế và chi phí thuê mặt bằng có thể tăng lên chóng mặt. Điều này dẫn đến nhiều thách thức trong hoạt động kinh doanh, bao gồm:

- Khách hàng cần phải dành thời gian đến các cửa hàng trực tiếp để mua hàng, tốn kém và không tiện lợi. Chưa kể tới mặt hàng cần mua đó thì lại có địa chỉ cửa hàng ở khá xa nơi khách hàng đang sinh sống.
- Sự thiếu hụt về thanh toán điện tử, việc lưu trữ thông tin khách hàng còn nhiều bất cập.
- Nhà đất ngày càng tăng dẫn tới chi phí thuê mặt bằng ngày một tăng. Do đó, khó khăn trong việc mở rộng quy mô kinh doanh và thuê thêm nhân lực.
- Khả năng quảng cáo và giới thiệu sản phẩm đến tay khách hàng là hạn chế.

- Quản lý việc giao hàng và thống kê số lượng sản phẩm tồn kho còn nhiều khó khăn, không đảm bảo tính chính xác và chặt chẽ.

Vì vậy, việc các cửa hàng kinh doanh trực tiếp áp dụng TMĐT trở nên cực kỳ cần thiết. Điều này giúp tăng cường quảng bá thương hiệu sản phẩm, giảm chi phí hoạt động và gia tăng tương tác với khách hàng.

1.4. Hướng giải quyết

Sau quá trình tìm hiểu và nghiên cứu, chúng em nhận thấy rằng hầu hết các trang web TMĐT hiện nay đều có đầy đủ các chức năng cơ bản như giới thiệu chi tiết về sản phẩm, tìm kiếm sản phẩm, thêm vào giỏ hàng, mua hàng, thanh toán tiền mặt và trực tuyến. Tuy nhiên, bên cạnh những chức năng trên những chức năng khác vẫn còn nhiều hạn chế như:

- Sự phức tạp trong mong muốn của khách hàng dẫn tới việc trang web không thỏa mãn được nhu cầu cá nhân của khách hàng, từ đó gây cho khách hàng một trải nghiệm không tích cực đối với trang web.
 - Hướng giải quyết:* khi khách hàng truy cập vào trang web và đăng ký tài khoản thì họ sẽ được điền một form để trả lời một vài câu hỏi liên quan đến giới tính, sở thích hay liệu họ đang quan tâm đến mặt hàng nào để trang web có thể lấy các câu trả lời đó làm mục tiêu quảng bá, hiển thị sản phẩm phù hợp nhất với khách hàng.
- Sự thiếu tin tưởng từ khách hàng với các sản phẩm trực tuyến vẫn còn đang là một thách thức lớn, khiến họ cảm thấy không hoàn toàn yên tâm khi mua hàng qua internet.
 - Hướng giải quyết:* trang web cần cung cấp thông tin rõ ràng và minh bạch về sản phẩm, chính sách đổi trả và bảo mật thông tin khách hàng. Ngoài ra, việc có những đánh giá và nhận xét tích cực từ khách hàng trước đó cũng giúp xây dựng lòng tin từ phía khách hàng mới.
- Sự phụ thuộc quá mức vào các đơn vị vận chuyển vì nếu đơn vị vận chuyển nghỉ thì sẽ phải tạm ngưng giao hàng cho khách do không có nhân lực để giao hàng.
 - Hướng giải quyết:* quản trị viên cần tìm kiếm và hợp tác với nhiều đối tác vận chuyển khác nhau. Điều này giúp tăng tính linh hoạt và giảm thiểu rủi ro khi một đơn vị vận chuyển gặp vấn đề.
- Việc kiểm soát số lượng sản phẩm bán ra và quản lý tồn kho gấp nhiều khó khăn, có thể dẫn đến thất thoát sản phẩm và tiền bạc đáng kể.
 - Hướng giải quyết:* thêm chức năng quản lý số lượng sản phẩm bán ra và tồn kho cho trang web.
- Phương thức thanh toán trực tuyến chưa thực sự phổ biến ở mọi độ tuổi.
 - Hướng giải quyết:* trang web cần có hướng dẫn chi tiết và tư vấn cho khách hàng về lợi ích và đảm bảo tính an toàn của việc thanh toán trực tuyến.

Các bước cơ bản trong quá trình thực hiện bài toán TMĐT bao gồm:

- **Bước 1:** Khách hàng đăng nhập bằng tài khoản hiện có hoặc có thẻ đăng ký mới nếu cần.
- **Bước 2:** Khách hàng có thể tìm kiếm sản phẩm theo nhiều cách khác nhau, bao gồm tìm kiếm bằng từ khóa, theo danh mục, hoặc theo các tiêu chí khác nhau như sản phẩm mới nhất, giảm giá, bán chạy, hoặc phổ biến.
- **Bước 3:** Sau khi tìm được sản phẩm mong muốn, khách hàng thêm sản phẩm vào giỏ hàng sau khi kiểm tra thông tin chi tiết và lựa chọn số lượng.
- **Bước 4:** Khách hàng kiểm tra giỏ hàng, cập nhật số lượng, số điện thoại và địa chỉ giao hàng trước khi tiến hành đặt hàng.
- **Bước 5:** Khách hàng chọn phương thức thanh toán phù hợp và kiểm tra lại thông tin đặt hàng trước khi hoàn tất.
- **Bước 6:** Sau khi đặt hàng thành công, hệ thống gửi mã số đơn hàng và thông báo thời gian giao hàng dự kiến cho khách hàng qua email.
- **Bước 7:** Khách hàng có thể theo dõi tiến trình xử lý và vận chuyển của đơn hàng trên website, sử dụng mã vận đơn được cung cấp.

1.5. Bố cục luận văn

Cấu trúc của bài báo cáo Luận văn này được tổ chức theo các phần chính tương ứng với nội dung của từng chương, như sau:

✚ *Chương 1: Tổng quan đề tài*

Chương này sẽ tổng quan về định nghĩa, lợi ích, và yêu cầu của thương mại điện tử. Mục đích, giúp người đọc có cái nhìn tổng quan ban đầu về lĩnh vực này, và chuẩn bị tinh thần cho việc tìm hiểu sâu hơn về các khía cạnh trong thương mại điện tử. Ngoài ra, chương này cũng nêu rõ mục tiêu, nhiệm vụ cần làm, từ đó đưa ra các phương pháp giải quyết vấn đề, và ý nghĩa thực tiễn của đề tài. Cuối cùng, là trình bày về cấu trúc của luận văn, đưa ra tóm tắt về nội dung của từng chương sẽ được giới thiệu phía sau.

✚ *Chương 2: Cơ sở lý thuyết và kỹ thuật liên quan*

Chương này cung cấp cái nhìn tổng quát về lý thuyết và kỹ thuật được áp dụng trong dự án, bao gồm các khía cạnh của thương mại điện tử và mô hình liên quan như B2C. Nó bao gồm các công nghệ được sử dụng trong quá trình phát triển dự án như PostgreSQL cho cơ sở dữ liệu, Java cho phần backend và HTML, CSS và JavaScript cho frontend và việc tổ chức hệ thống theo mô hình Monolithic, sử dụng token để bảo mật và VNPay để thanh toán trực tuyến.

✚ *Chương 3: Phân tích và thiết kế hệ thống*

Chương này tập trung vào quá trình phân tích và thiết kế hệ thống cho dự án. Trong phần này, chúng em sẽ đi sâu vào việc phân tích các yêu cầu chức năng và phi chức năng của hệ thống, đồng thời tạo ra các thiết kế logic và cấu trúc phần mềm. Bằng cách này, chúng

em sẽ xác định cách triển khai và hoạt động cho dự án, từ đó đảm bảo sự hiệu quả và đáp ứng đúng đắn với mục tiêu đề ra và thỏa nhu cầu của người dùng.

 **Chương 4: Kết quả chương trình**

Chương này sẽ trình bày kết quả của việc áp dụng những lý thuyết được thảo luận trong Chương 2 vào quá trình phân tích và thiết kế hệ thống trong Chương 3.

 **Chương 5: Tổng kết và hướng phát triển đề tài**

Chương này sẽ chứa phần tổng hợp và tóm tắt lại toàn bộ nội dung và kết quả đạt được trong quá trình nghiên cứu của luận văn. Ngoài ra, chương cũng sẽ bao gồm các ý kiến đóng góp cho đề tài và đề xuất hướng phát triển tiềm năng cho dự án.

 **Tài liệu tham khảo**

CHƯƠNG 2: CÔNG NGHỆ ÁP DỤNG, KỸ THUẬT LIÊN QUAN

2.1. Giới thiệu về thương mại điện tử

2.1.1. Định nghĩa

- **Thương mại điện tử (E-commerce/electronic commerce)**

Là hoạt động mua bán hàng hóa và dịch vụ dựa trên Internet thông qua các phương tiện điện tử. Đây là một lĩnh vực đa dạng, bao gồm nhiều hình thức như mua bán hàng hóa và dịch vụ trực tuyến, giao nhận thông tin qua mạng, thanh toán trực tuyến và các hoạt động đấu giá. Nó đang tạo ra một cuộc cách mạng trong cách mua sắm, ảnh hưởng đến các khía cạnh từ sản xuất, quảng cáo, bán hàng, phân phối sản phẩm và thanh toán, cũng như việc số hóa thông tin qua Internet. Đây là nơi các cửa hàng truyền thống mở rộng không gian kinh doanh từ phố xá nhộn nhịp của thế giới thực sang không gian kỹ thuật số.

- **Website thương mại điện tử**

Là một trang web động, được xây dựng để hỗ trợ quy trình mua bán hàng hóa, từ trình bày và giới thiệu sản phẩm hoặc dịch vụ đến việc ký kết hợp đồng, cung cấp dịch vụ, thanh toán và hỗ trợ sau bán hàng. Trang web này cung cấp các tính năng từ cơ bản đến nâng cao, sử dụng các công nghệ hiện đại trong lĩnh vực Công nghệ Thông tin để tạo ra một trải nghiệm trực tuyến linh hoạt và thân thiện với người dùng. Điều này giúp người bán có thể quảng bá sản phẩm của họ tới người tiêu dùng và người dùng có thể dễ dàng tìm kiếm và lựa chọn sản phẩm một cách thuận tiện và nhanh chóng.

Một trang web TMĐT thường đi kèm với các tính năng cơ bản như đăng ký, đăng nhập, gio hàng, đặt hàng, trình bày và quản lý sản phẩm, cũng như quản lý đơn hàng. Đồng thời, nó cũng tích hợp các phương thức thanh toán đa dạng như VNPay, Banking, Momo, ZaloPay, ... để người dùng có thể lựa chọn phương thức thanh toán phù hợp và thuận tiện nhất cho họ.

- **Những đặc trưng nổi bật của một trang web TMĐT:**

- Thao tác đơn giản, thuận tiện và nhanh chóng.
- Cung cấp thông tin rõ ràng, chính xác và nhanh chóng.
- Dễ dàng tương tác với người bán.
- Hình thức thanh toán đa dạng, có thể là tiền mặt hay thanh toán bằng thẻ.
- Tính năng gio hàng thông minh.
- Hình ảnh và video sinh động, chân thực, chất lượng.
- Tăng sự tương tác, khả năng mua hàng của khách hàng.

2.1.2. Lợi ích và khó khăn của thương mại điện tử

2.1.2.1. Lợi ích

TMĐT ngày càng trở nên cần thiết trong cuộc sống, không thể phủ nhận nó có vai trò quan trọng vào việc cung cấp các phương thức giao dịch hiệu quả, nhanh chóng và an toàn. TMĐT tận dụng tối đa mọi nguồn lực có sẵn và là một sự kết hợp đặc biệt trong lĩnh vực khoa học và công nghệ áp dụng vào hoạt động kinh doanh. Rõ hơn về sự hữu ích đó:

- ***Đối với khách hàng***

Khách hàng không phải di chuyển đến cửa hàng để thực hiện mua hàng. Thay vào đó ở bất kỳ đâu chỉ cần có kết nối Internet sẽ có thể thực hiện mua sắm bất kỳ lúc nào họ muốn. Khi đó họ tiết kiệm được rất nhiều thời gian và công sức đi lại.

- ***Đối với doanh nghiệp***

Việc kinh doanh trên mạng cho phép họ dễ dàng triển khai các ý tưởng về các dịch vụ hỗ trợ và chiến lược tiếp thị. Sự sáng tạo trở thành yếu tố quyết định khi tất cả các đối thủ đều áp dụng TMĐT. Vì thế mang lại sự phát triển cho thị trường TMĐT nói chung và nền kinh tế đất nước nói riêng.

Do TMĐT được thực hiện trên mạng, nên không bị hạn chế bởi khoảng cách địa lý. Điều này cho phép cả những nhà cung cấp nhỏ và lớn đều có thể được nhiều người biết đến, nhờ vào tính toàn cầu của mạng. Hạn chế tối đa cho chi phí thuê nhân viên quản lý, bán hàng và mặt tiền thuê mặt bằng đất đỏ.

Trong kinh doanh truyền thống, doanh nghiệp cần xem xét nhiều yếu tố như mặt bằng, đội ngũ nhân sự, chi phí, kho chứa, và phải liên kết một cách hợp lý để tạo ra lợi nhuận. Trái lại, với các phần mềm TMĐT, doanh nghiệp có thể dễ dàng linh hoạt mở rộng quy mô và phát triển sản phẩm. Bằng cách nâng cấp hệ thống hoặc tăng chi phí để tiếp cận nhiều khu vực khác nhau, doanh nghiệp có thể tiếp cận được nhiều khách hàng hơn một cách hiệu quả.

TMĐT cũng tạo điều kiện thuận lợi cho việc tương tác với khách hàng, giúp cho việc hỗ trợ và tư vấn cho họ diễn ra một cách nhanh chóng và thuận tiện hơn. Quan trọng hơn, việc cá nhân hóa chăm sóc khách hàng giúp tạo ra một môi trường gần gũi hơn, từ đó nâng cao hiệu quả của quy trình chăm sóc khách hàng. Tạo điều kiện cho việc phát triển một cộng đồng khách hàng trung thành. Một yếu tố nhỏ mà TMĐT đã góp phần đó là bảo vệ môi trường giảm thiểu tắc giao thông và hạn chế khí thải ô nhiễm.

2.1.2.2. Khó khăn

Hiện nay, tình trạng gian lận trong TMĐT ngày càng phổ biến, tội phạm thực hiện nhiều hành vi ngày càng tinh vi để có thể xâm nhập vào cửa hàng trực tuyến nhằm mục đích đánh cắp thông tin quan trọng hay giả danh nhà cung cấp dịch vụ để lừa gạt chiếm đoạt tài sản. Chính vì thế đã làm vấy lên sự lo ngại cho khách hàng khi sử dụng các dịch vụ TMĐT.

2.1.3. Các loại hình giao dịch trong thương mại điện tử

Trong TMĐT, ba chủ thể chính là: Doanh nghiệp (B), Người tiêu dùng (C), và Chính phủ (G). Doanh nghiệp đóng vai trò là động lực chính trong việc thúc đẩy sự phát triển của

TMĐT, trong khi Người tiêu dùng giữ vai trò quyết định đến sự thành công của hệ thống này. Về phía chính phủ có trách nhiệm định hướng, điều tiết và quản lý hoạt động TMĐT.

Dựa vào mối quan hệ giữa các chủ thể trên, chúng ta có các loại hình giao dịch TMĐT như: B2B (Business-to-Business), B2C (Business-to-Consumer), B2G (Business-to-Government), C2G (Consumer-to-Government), C2C (Consumer-to-Consumer), ... trong đó, B2B, B2C và C2C được coi là ba loại hình giao dịch TMĐT quan trọng nhất.

2.1.3.1. Mô hình B2B (Business-to-Business)

B2B tức là kinh doanh giữa các doanh nghiệp. Trong mô hình này, các doanh nghiệp thực hiện giao dịch và hợp tác với nhau. Mô hình áp dụng cho hình thức kinh doanh có chứng từ giữa các công ty, tổ chức, giữa công ty mẹ và các công ty con, giữa các công ty trong cùng hiệp hội, ... Điện hình là khi một doanh nghiệp cung cấp sản phẩm, dịch vụ hoặc hàng hóa cho một công ty khác. Bên cạnh đó, B2B đóng vai trò không thể thiếu trong chuỗi cung ứng.

- ***Ưu điểm***

Trong mô hình B2B, các giao dịch thường có quy mô lớn hơn so với B2C, đặc biệt là khi các doanh nghiệp lớn hợp tác với nhau. Điều này mang lại sự ổn định và đáng tin cậy cho cả hai bên, bởi các doanh nghiệp thường có kinh nghiệm và sự chuyên môn trong lĩnh vực của mình. Mỗi giao dịch B2B có thể bao gồm hàng nghìn mặt hàng và giá trị của mỗi đơn hàng trong mô hình này vì vậy cũng rất lớn. Từ góc nhìn của bên mua hàng, họ có được giá tốt hơn do các doanh nghiệp thực hiện giao dịch B2B thường có mối quan hệ tốt với nhau. Cũng chính vì quan hệ tốt với nhau nên đôi bên có thể hợp tác, tận dụng tối đa tài nguyên của nhau. Ví dụ, họ có thể hợp tác quảng cáo, truyền thông, trong vận chuyển hàng hóa hoặc trong vận hành...

- ***Nhược điểm***

Cơ sở thị trường hạn chế, đây cũng là điều hiển nhiên vì số lượng doanh nghiệp luôn ít hơn số khách hàng cá nhân. Tốn nhiều nguồn lực và thời gian vào nghiên cứu để quảng cáo, tiếp cận những khách hàng tiềm năng, ...

Đây r้าย các quy trình và quy định phức tạp, mỗi giao dịch B2B sẽ cần hai doanh nghiệp tham gia, mà mỗi doanh nghiệp lại có nhiều phòng ban, nhiều người tham gia vào quyết định chấp nhận giao dịch. Có thể gặp phải rủi ro tài chính lớn, đặc biệt là khi liên quan đến việc thanh toán, vận chuyển và hợp đồng dài hạn. Việc tranh gay gắt và khốc liệt là điều không thể tránh khỏi từ các đối thủ cùng ngành, đòi hỏi các doanh nghiệp phải nỗ lực để giữ vững vị thế của mình.

2.1.3.2. Mô hình B2C (Business-to-Customer)

Mô hình B2C là kinh doanh giữa các doanh nghiệp hướng tới người tiêu dùng cá nhân. Nói cách khác B2C chính là mô hình chuyên về lĩnh vực bán lẻ, doanh nghiệp trực tiếp bán hàng hóa, cung cấp dịch vụ cho khách hàng tiêu dùng. Các mặt hàng thường là máy tính, đồ

điện tử, dụng cụ thể thao, đồ dùng văn phòng, sách và âm nhạc, đồ chơi, sức khoẻ và mỹ phẩm, giải trí.

- ***Ưu điểm***

Mô hình này cho phép doanh nghiệp có nhiều cơ hội tiếp cận khách hàng trên phạm vi rộng hơn. Chi phí phải bỏ ra cho các dịch vụ quảng cáo trên các website hay mạng xã hội sẽ rẻ hơn và tối ưu lượt tiếp cận hơn so với các chi phí biển quảng cáo ngoài trời. Đồng thời tiết kiệm nhiều hơn chi phí liên quan tới cơ sở hạ tầng, điện nước, phí nhân viên so với mô hình kinh doanh bán lẻ truyền thống. Bên cạnh đó, doanh nghiệp dễ dàng quản lý hàng tồn kho bằng các phần mềm quản lý và số lượng nhân công cho việc đó chắc chắn sẽ ít hơn.

Việc chăm sóc khách hàng là một trong những nhiệm vụ đầy thách thức và khó khăn lớn nhất trong việc kinh doanh. Để giải quyết vấn đề này B2C có thể cho phép chúng ta tương tác trực tiếp với người mua qua email, SMS, thông báo, ... Bằng cách này, doanh nghiệp sẽ tối ưu hoá được lượng người truy cập vào website cũng như chăm sóc được khách hàng nếu họ có nhu cầu quan tâm thêm về sản phẩm của mình.

- ***Nhược điểm***

Thị trường B2C thường rất cạnh tranh, đặc biệt là trên internet, khiến cho việc thu hút và giữ chân khách hàng trở nên khó khăn. Do đặc điểm hàng hoá được giao bán trên các trang TMĐT, người mua không thể tự tay kiểm tra chất lượng trước khi cân nhắc có nên mua sản phẩm đó hay không. Bên trung gian thứ ba khó có thể kiểm soát chặt chẽ chất lượng sản phẩm từ nhà cung cấp trước khi nó được đưa đến tay của người tiêu dùng. Vì thế dẫn đến việc khó đảm bảo chất lượng.

Tâm lý của mọi khách hàng ngoài việc chất lượng sản phẩm tốt và giá cả hợp lý thì họ còn chú trọng việc bảo mật thông tin cá nhân. Chính vì thế ngành công nghiệp TMĐT luôn đặt ra một nhiệm vụ lâu dài và khó khăn đó là thuyết phục người tiêu dùng tin tưởng và tính bảo mật của những giao dịch trực tuyến của họ.

2.1.3.3. Mô hình C2C (Customer-to-Customer)

Mô hình C2C là một mô hình kinh doanh trong đó các giao dịch được thực hiện trực tiếp giữa các cá nhân, thay vì thông qua doanh nghiệp trung gian. Trong mô hình này, các cá nhân sẽ đóng vai trò cả người bán và người mua, tiến hành mua bán hàng hoá hoặc cung cấp dịch vụ cho nhau. Một ví dụ phổ biến của mô hình C2C là các trang web TMĐT dạng chợ trực tuyến, nơi người dùng có thể đăng thông tin về sản phẩm của mình và tiến hành giao dịch với nhau.

- ***Ưu điểm***

Người bán và người mua có thể tương tác trực tiếp mà không cần qua bất kỳ sự can thiệp nào từ doanh nghiệp trung gian, giúp tăng tính linh hoạt trong quá trình giao dịch. Do loại bỏ bước trung gian, mô hình C2C có thể giảm chi phí cho cả người mua lẫn người bán.

Tận dụng được tối đa giá trị của sản phẩm thông qua việc rao bán các sản phẩm không có nhu cầu sử dụng hay sản phẩm đã qua sử dụng nhưng người tiêu dùng không còn nhu cầu sử dụng nữa. Nhờ vậy, giá trị sản phẩm được khai thác tối đa, không bị vứt đi lãng phí.

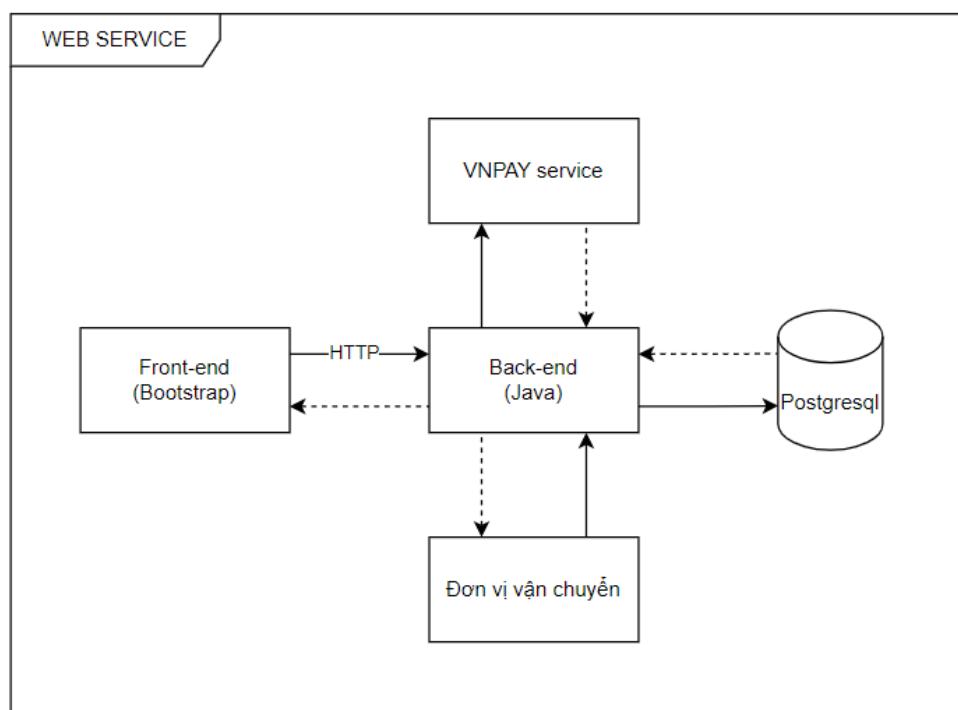
- **Nhược điểm**

Không có sự kiểm soát chặt chẽ về chất lượng sản phẩm và dịch vụ, người mua có thể gặp phải rủi ro khi mua hàng từ các cá nhân không đáng tin cậy. Việc quản lý giao dịch và giải quyết các tranh chấp có thể phức tạp do sự trực tiếp giữa các cá nhân, đặc biệt khi có tranh chấp phát sinh trong quá trình giao dịch.

2.2. Kiến thức hệ thống

2.2.1. Mô hình RESTful Web Service

2.2.1.1. Web service



Hình 2.2.1.1. Mô hình Web service

Thuật ngữ Web Service (WS) có thể được hiểu theo hai nghĩa:

- WS là một dịch vụ mà một thiết bị điện tử cung cấp cho một thiết bị khác. Hai máy chủ này sẽ giao tiếp với nhau thông qua World Wide Web. Hoặc:
- WS là một server chạy trên một thiết bị máy tính, nghe các request tại một port nhất định qua mạng, phục vụ các web document (HTML, JSON, XML, image...).

Các Web Service thường cung cấp dữ liệu dưới dạng XML hoặc JSON để máy tính có thể đọc và xử lý thông tin. Điều này giúp các ứng dụng có thể trao đổi dữ liệu với nhau một cách hiệu quả, nhưng đôi khi làm cho dữ liệu trở nên khó hiểu đối với người dùng thông thường.

2.2.1.2. RESTful web service

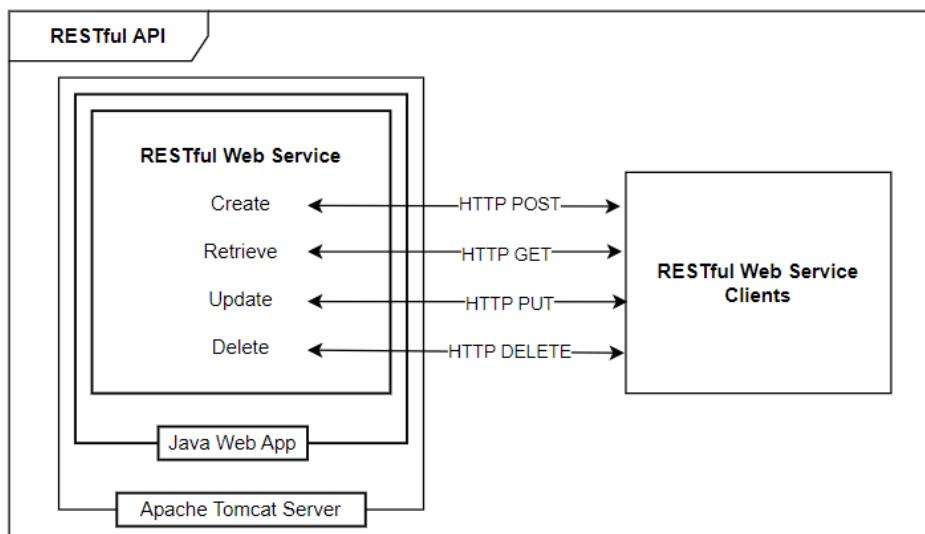
RESTful web service là các dịch vụ web được xây dựng dựa trên cấu trúc REST (REpresentational State Transfer). Tức là nó giống như một kiến trúc, nguyên tắc cần tuân theo để thiết kế, xây dựng một web service.

Hiện nay, REST đã được chọn để sử dụng rộng rãi thay thế cho các web service dựa trên SOAP (Simple Object Access Protocol) và WSDL (Web Services Description Language). RESTful web service nhẹ, có khả năng dễ mở rộng và bảo trì. Trong kiến trúc REST mọi thứ đều được coi là tài nguyên, chúng có thể là: tệp văn bản, ảnh, trang html, video, hoặc dữ liệu động... REST là một bộ quy tắc để tạo ra một ứng dụng Web Service, mà nó tuân thủ 4 nguyên tắc thiết kế cơ bản sau:

- Yêu cầu các nhà phát triển sử dụng một cách rõ ràng các phương thức HTTP và nhất quán với cách mà chúng được định nghĩa.

Theo đó sẽ có:

- Để tạo một tài nguyên trên server ta dùng phương thức POST
- Để lấy(đọc) tài nguyên trên server ta dùng phương thức GET
- Để update tài nguyên trên server ta dùng phương thức PUT
- Để xóa tài nguyên trên server ta dùng phương thức DELETE



Hình 2.2.1.2. RESTful API

- Hiển thị cấu trúc thư mục như URI.
- Chuyển đổi XML, JSON hoặc cả hai.
- Phi trạng thái: máy chủ sẽ không lưu giữ thông tin của client mà nó giao tiếp, thông tin hoặc được giữ trên client hoặc được chuyển thành trạng thái của tài nguyên. Mỗi request lên server thì client phải đóng gói thông tin đầy đủ để server hiểu được. Lợi ích mà nó mang lại là giúp tách biệt client ra khỏi sự thay đổi của server. Giúp hệ thống của chúng ta dễ phát triển, bảo trì, mở rộng vì không cần tốn công CRUD trạng thái của client.

2.2.2. Kiến trúc Microservice và Monolithic

2.2.2.1. Microservice

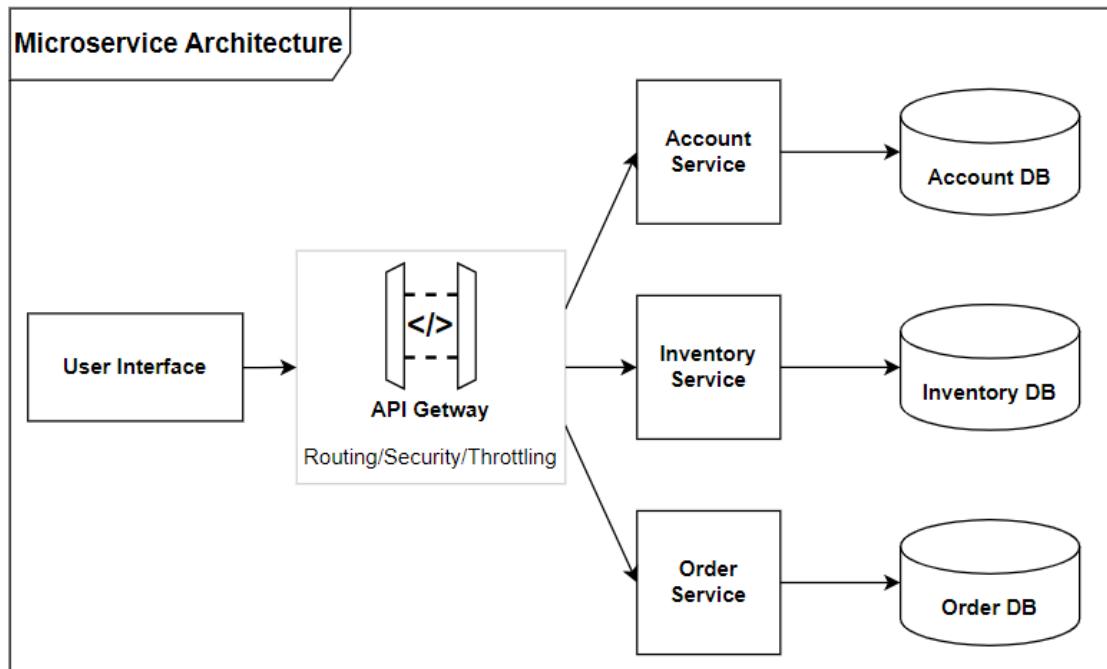
2.2.2.1.1. Giới thiệu

Microservice là một kiến trúc phần mềm mà ở đó các ứng dụng chúng ta được chia nhỏ thành những ứng dụng nhỏ hơn. Mà trong đó các ứng dụng nhỏ hơn này đảm bảo được một số tiêu chí sau:

Được deploy độc lập với nhau: nghĩa là trong những ứng dụng nhỏ này sẽ được deploy hoàn toàn độc lập với nhau mà không ảnh hưởng đến các ứng dụng khác hay nói cách khác nó sẽ không ảnh hưởng đến các microservice khác.

Các microservice có liên kết lỏng với nhau: tức là nó không thể có liên kết quá chặt với nhau và đây cũng là một nguyên tắc rất quan trọng trong việc thiết kế mô hình microservice. Điều đó sẽ góp phần giúp cho các microservice có thể hoạt động độc lập với nhau mà không cần phụ thuộc quá nhiều vào các microservice khác.

Được xây dựng và tổ chức dựa trên các tính năng của hệ thống (business capabilities): Nói cách khác nó sẽ được xuất phát từ những yêu cầu của hệ thống. Ví dụ: khi dựng một hệ thống về TMĐT thì đâu đó chúng ta sẽ cần thiết kế microservice để phục vụ cho trang TMĐT như microservice repository, microservice account, microservice order, và nhiều các microservice khác nữa. Để các dịch vụ trong kiến trúc microservice hiểu nhau, chúng cần định rõ cách giao tiếp và trao đổi thông tin. Điều này có thể được thông qua việc đồng thuận về các giao thức giao tiếp như RESTful API hoặc message queue, định dạng dữ liệu như JSON hoặc XML, và cấu trúc dữ liệu...



Hình 2.2.2.1. Kiến trúc Microservice

2.2.2.1.2. Ưu điểm

Mỗi dịch vụ (service) được quản lý và bảo trì độc lập, không ảnh hưởng đến các dịch vụ khác trong hệ thống. Việc bảo trì một dịch vụ không gây ra sự gián đoạn hoạt động của các phần còn lại của hệ thống.

Giúp làm giảm sự phức tạp của mỗi dịch vụ và làm cho mã nguồn của chúng dễ hiểu hơn. Chúng ta có thể tập trung vào cụ thể của hệ thống mà không cần phải lo lắng về các phần khác. Hơn nữa, vì mỗi dịch vụ có thể được triển khai và quản lý độc lập, việc thực hiện các bản vá trở nên đơn giản và dễ dàng hơn.

Mỗi dịch vụ (service) được coi như một dự án riêng, vì vậy mà chúng có thể sử dụng các công nghệ (như ngôn ngữ lập trình, database) khác nhau. Chúng chỉ cần đảm bảo tuân theo chuẩn giao tiếp được đặt ra từ trước, còn bên trong chúng thực hiện ra sao và sử dụng công nghệ gì không ảnh hưởng tới các service còn lại.

2.2.2.1.3. Nhược điểm

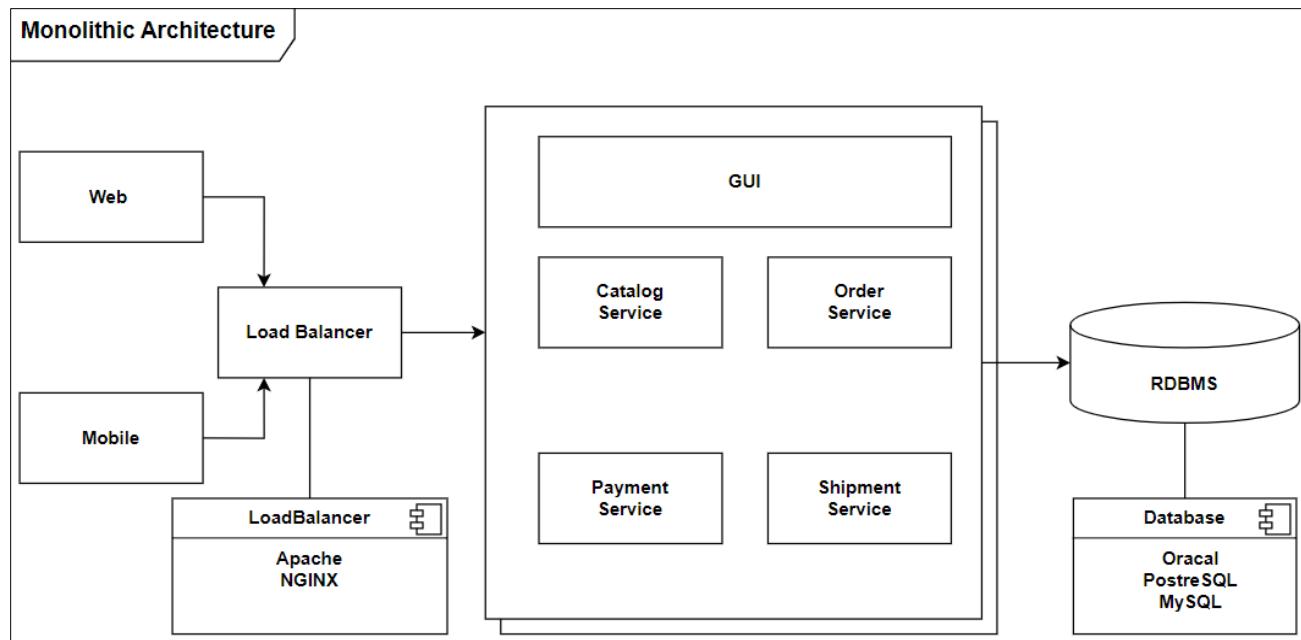
- **Phân mảnh dữ liệu:** Trong kiến trúc microservices, dữ liệu thường được phân tán và lưu trữ trong nhiều dịch vụ khác nhau, làm cho việc đảm bảo tính toàn vẹn dữ liệu trở nên phức tạp.
- **Vấn đề về thứ tự xử lý:** Ví dụ, khi tạo đơn hàng, thông tin khách hàng phải được tạo trước khi tạo thông tin đơn hàng. Trong kiến trúc monolithic, cơ sở dữ liệu được lưu trữ tập trung và thường được ràng buộc bằng khóa ngoại để đảm bảo thứ tự này. Tuy nhiên, với microservices, không có gì đảm bảo rằng dữ liệu được tạo ra theo đúng thứ tự khi "khách hàng" và "đơn hàng" là hai dịch vụ riêng biệt.
- **Giao tiếp qua mạng:** Trong kiến trúc microservices, các dịch vụ giao tiếp với nhau thông qua mạng, dẫn đến độ trễ cao hơn so với các phương thức nội bộ trong một hệ thống monolithic.

2.2.2.2. Monolithic

2.2.2.2.1. Giới thiệu

Kiến trúc Monolithic là một phương pháp phát triển phần mềm truyền thống trong đó toàn bộ chức năng của hệ thống dựa trên một ứng dụng duy nhất, tương tự như một khối đá nguyên khối khổng lồ. Tất cả các thành phần của ứng dụng từ giao diện người dùng, logic nghiệp vụ cho đến truy cập cơ sở dữ liệu đều được tích hợp vào một nền tảng duy nhất và hoạt động như một đơn vị tự trị.

Hãy tưởng tượng một tảng đá khổng lồ, vững chắc và liền mạch. Mọi phần của tảng đá này đều kết nối chặt chẽ với nhau, không có chỗ cho sự tách rời hay phân chia. Tảng đá này chính là hình ảnh tượng trưng cho một ứng dụng monolithic: tất cả các phần của ứng dụng đều nằm trong một khối duy nhất, được xây dựng, kiểm thử, triển khai và mở rộng như một thể thống nhất.



Hình 2.2.2.2. Kiến trúc Monolithic

2.2.2.2. Ưu điểm

- **Đơn giản trong triển khai:** chỉ cần một tệp triển khai duy nhất, dễ dàng hơn trong việc quản lý phiên bản và triển khai.
- **Hiệu suất cao:** do các thành phần liên kết chặt chẽ và giao tiếp nội bộ không cần qua mạng, giúp giảm thiểu độ trễ.
- **Phát triển nhanh cho các ứng dụng nhỏ:** với các ứng dụng nhỏ, việc phát triển và bảo trì có thể nhanh chóng và ít phức tạp hơn.
- **Hoạt động mượt mà:** Vì không sử dụng API để giao tiếp giữa các thành phần nên ứng dụng monolithic hoạt động tốt hơn ứng dụng microservice.

2.2.2.3. Nhược điểm

- **Phức tạp khi mở rộng:** trái với kiến trúc microservice thì monolithic các khớp nối sẽ liên kết chặt chẽ với nhau. Và cũng vì thế nên mã của chúng có xu hướng trở nên phức tạp khi mở rộng quy mô, khiến việc tách biệt các dịch vụ để hỗ trợ mã và mở rộng quy mô độc lập trở nên khó khăn hơn.
- **Tính linh hoạt thấp:** vì một phần phụ thuộc vào phần khác nên bất kỳ thay đổi nào cũng tác động đến toàn bộ mạng. Kết quả là, những thay đổi cần có thời gian và phải được phối hợp cẩn thận. Ngoài ra, mỗi bản cập nhật nhỏ đều cần phải triển khai lại.
- **Giới hạn mở rộng quy mô:** ứng dụng sử dụng kiến trúc monolithic chỉ có thể mở rộng quy mô tổng thể. Việc mở rộng quy mô các thành phần riêng lẻ là không thể.

2.3. Các công nghệ tiên tiến cho phát triển và quản lý ứng dụng

2.3.1. Spring Boot 3

2.3.1.1. Giới thiệu:

Spring Boot là một extension của Spring framework chuyên cung cấp các tính năng RAD (Rapid Application Development) cho phép tạo ra và phát triển các ứng dụng độc lập dựa trên Spring một cách nhanh chóng.

Spring Boot ra đời với mục đích loại bỏ những cấu hình phức tạp của Spring, việc yêu cầu cấu hình XML là không cần thiết và nâng cao năng suất cho các nhà phát triển. Với sự góp mặt của Spring Boot, hệ sinh thái Spring đã trở nên mạnh mẽ, phổ biến và hiệu quả hơn bao giờ hết.

2.3.1.2. Ưu điểm

- Cũng như các đề cập trên thì Spring Boot đã khắc phục được những hạn chế về cấu hình của Spring. Hội tụ đầy đủ các tính năng của Spring framework.
- Tích hợp tốt với các thư viện và framework phổ biến khác như Spring Data, Spring Security, Hibernate, và Thymeleaf...
- Sử dụng Spring Boot, chúng ta có thể quản lý các dependency của ứng dụng một cách dễ dàng thông qua Maven hoặc Gradle. Khởi tạo ứng dụng nhanh chóng và dễ dàng hơn.
- Đặc tính cơ bản có thể kể đến đó là cung cấp sẵn cho chúng ta một lớp có chứa hàm main và được hoạch định làm điểm mở đầu cho toàn bộ chương trình. Lớp này được đặt tên là Spring Application, nó giúp khởi chạy các ứng dụng từ hàm main, khi chạy nó chúng ta chỉ cần gọi phương thức run.

2.3.1.3. Nhược điểm

- Đôi khi, quá trình tự động cấu hình của Spring Boot có thể làm cho việc kiểm soát ứng dụng trở nên khó khăn, đặc biệt là đối với các ứng dụng lớn và phức tạp.
- Việc tùy chỉnh cấu hình hoặc thay đổi mặc định của Spring Boot có thể gặp khó khăn, đặc biệt là đối với những người mới bắt đầu.

2.3.2. Bootstrap 5

2.3.2.1. Giới thiệu

Bootstrap được phát triển bởi Mark Otto và Jacob Thornton với mục đích ban đầu là sử dụng bootstrap như một framework hỗ trợ nội bộ tại công ty Twitter và xuất bản như một mã nguồn mở vào tháng 8/2011 trên website GitHub.

Là một framework HTML, CSS, và JavaScript phổ biến để phát triển các trang web và ứng dụng web đáp ứng nhanh chóng và tiện lợi hơn. Được phát triển bởi Twitter, Bootstrap cung cấp một bộ công cụ mạnh mẽ và dễ sử dụng để xây dựng các giao diện người dùng (UI) hiện đại, tương thích với nhiều thiết bị và trình duyệt.

2.3.2.2. Lợi thế

- Dễ học, dễ sử dụng: cộng đồng đông đúc và tài liệu tham khảo rõ ràng chính là sức mạnh của Bootstrap. Mặc định hỗ trợ responsive và viết theo xu hướng mobile first,

nghĩa là ưu tiên giao diện mobile trước. Điều này cải thiện đáng kể hiệu suất trang web khi có người dùng truy cập bằng mobile. Từ đó không còn nỗi lo trang web đó có thể chạy trên nền tảng di động hay không.

- Bootstrap đảm bảo tính tương thích cao với hầu hết các trình duyệt web phổ biến. Đồng thời, tích hợp nhiều thành phần UI với nhiều thành phần giao diện như nút, bảng, form, modal, navbars, v.v., giúp nhanh chóng tạo ra các phần tử giao diện nhất quán và chuyên nghiệp.

2.3.2.3. Nhược điểm

- Mặc dù Bootstrap rất linh hoạt, nhưng việc sử dụng quá nhiều các thành phần mặc định có thể dẫn đến các trang web trông giống nhau. Điều này có thể làm giảm tính độc đáo và thương hiệu của trang web.
- Bootstrap bao gồm rất nhiều thành phần và chức năng, các tệp CSS và JavaScript của nó có thể khá lớn, làm tăng thời gian tải trang nếu không được tối ưu hóa.
- Đối với các dự án yêu cầu tính cá nhân hóa cao hoặc các giao diện người dùng phức tạp, Bootstrap có thể không phải là lựa chọn tốt nhất.

2.3.3. PostgreSQL

2.3.3.1. Giới thiệu

PostgreSQL là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mã nguồn mở mạnh mẽ, được sử dụng rộng rãi bởi các doanh nghiệp và phát triển phần mềm để lưu trữ và quản lý dữ liệu. PostgreSQL nổi tiếng với tính năng mạnh mẽ, tuân thủ chặt chẽ các tiêu chuẩn SQL và khả năng mở rộng cao.

2.3.3.2. Ưu điểm

- PostgreSQL tuân thủ các tiêu chuẩn SQL chặt chẽ, giúp đảm bảo tính nhất quán và tính toàn vẹn của dữ liệu. Điều này rất quan trọng trong một hệ thống Ecommerce, nơi dữ liệu sản phẩm, đơn hàng, khách hàng, và thanh toán phải luôn chính xác và đáng tin cậy.
- PostgreSQL có khả năng mở rộng cả theo chiều dọc (vertical scaling) và chiều ngang (horizontal scaling) nhờ vào các kỹ thuật như partitioning, replication, và sharding.
- Bên cạnh đó một lợi thế để chọn sử dụng PostgreSQL đó là sự đồng bộ. Khi phải chuyển từ hệ quản trị cơ sở dữ liệu quan hệ này sang một hệ quản trị cơ sở dữ liệu khác như Oracle, MySQL, SQLServer, ...

2.3.3.3. Nhược điểm

- Do tính năng phong phú và khả năng tùy biến cao, việc cấu hình và quản lý PostgreSQL có thể phức tạp hơn so với các hệ quản trị cơ sở dữ liệu khác.
- PostgreSQL có thể yêu cầu nhiều tài nguyên hệ thống hơn để chạy hiệu quả, đặc biệt khi xử lý các tác vụ phức tạp và khối lượng dữ liệu lớn.

- Đối với người mới, việc học và sử dụng PostgreSQL có thể tốn nhiều thời gian và công sức do phải nắm vững nhiều khái niệm và công cụ khác nhau.

2.3.4. DBeaver

2.3.4.1. DBeaver là công cụ quản lý cơ sở dữ liệu toàn diện

DBeaver là một công cụ quản lý cơ sở dữ liệu mã nguồn mở mạnh mẽ và đa năng, được thiết kế để hỗ trợ các nhà phát triển, quản trị viên cơ sở dữ liệu và các chuyên gia dữ liệu trong việc quản lý và tương tác với nhiều hệ quản trị cơ sở dữ liệu khác nhau. Với giao diện người dùng thân thiện và tính năng phong phú, DBeaver trở thành lựa chọn hàng đầu cho các chuyên gia trong lĩnh vực dữ liệu.

2.3.4.2. Điểm nổi bật của DBeaver trong quản lý dữ liệu

- Hỗ trợ đa dạng các hệ quản trị cơ sở dữ liệu:** DBeaver hỗ trợ một loạt các hệ quản trị cơ sở dữ liệu như MySQL, PostgreSQL, Oracle, SQL Server, SQLite, DB2, và nhiều hệ quản trị khác. Điều này giúp người dùng có thể quản lý và tương tác với nhiều cơ sở dữ liệu từ một công cụ duy nhất, tiết kiệm thời gian và công sức.
- Giao diện người dùng thân thiện:** giao diện đồ họa của DBeaver được thiết kế tối ưu, giúp người dùng dễ dàng điều hướng và thao tác. Tính năng kéo và thả, cùng với các tùy chọn cấu hình linh hoạt, giúp việc quản lý dữ liệu trở nên dễ dàng hơn bao giờ hết.
- Tích hợp với nhiều công cụ khác:** DBeaver tích hợp tốt với nhiều công cụ khác nhau, từ các hệ thống quản lý dự án đến các công cụ phân tích dữ liệu. Điều này giúp tạo ra một hệ sinh thái làm việc liền mạch, tăng cường hiệu quả và năng suất.

2.3.4.3. Những thách thức cần lưu ý

- Tài nguyên hệ thống:** DBeaver, do tính năng phong phú và giao diện đồ họa, có thể tiêu tốn khá nhiều tài nguyên hệ thống, đặc biệt khi làm việc với các cơ sở dữ liệu lớn. Điều này có thể ảnh hưởng đến hiệu suất làm việc nếu không có một hệ thống máy tính đủ mạnh.
- Độ phức tạp cho người mới bắt đầu:** dù có giao diện người dùng thân thiện, nhưng đối với những người mới bắt đầu, việc nắm bắt toàn bộ các tính năng của DBeaver có thể là một thách thức. Cần thời gian và sự kiên nhẫn để làm quen và sử dụng thành thạo công cụ này.

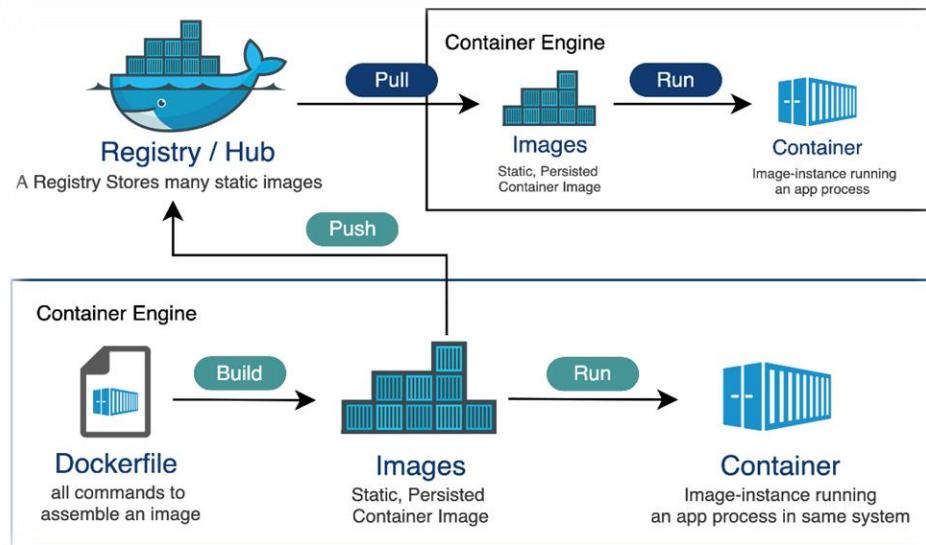
2.3.5. Docker

2.3.5.1. Giới thiệu

Trong thế giới công nghệ phát triển nhanh chóng, Docker đã nổi lên như một công cụ không thể thiếu, mở ra một cuộc cách mạng hóa trong việc quản lý và triển khai ứng dụng.

Docker là một nền tảng containerization mã nguồn mở, giúp các nhà phát triển tạo, triển khai và chạy các ứng dụng một cách nhất quán trong mọi môi trường. Ra mắt vào năm

2013, Docker đã nhanh chóng trở thành biểu tượng của sự đổi mới, giúp chuẩn hóa môi trường phần mềm, giảm thiểu sự không nhất quán và tối ưu hóa quy trình phát triển.



Hình 2.3.5.1. Kiến trúc Docker

Có năm thành phần chính trong kiến trúc Docker:

- **Docker Daemon:** tuân theo các yêu cầu API Docker và quản lý các đối tượng Docker như images, containers, networks and volumes.
- **Docker Clients:** với sự trợ giúp của Docker Client, người dùng có thể tương tác với Docker. Máy khách Docker cung cấp giao diện dòng lệnh (CLI) cho phép người dùng chạy và dừng các lệnh ứng dụng đối với daemon Docker.
- **Docker Host:** cung cấp một môi trường hoàn chỉnh để thực thi và chạy các ứng dụng. Nó bao gồm daemon Docker, Images, Containers, Networks, and Storage.
- **Docker Registry:** là nơi lưu trữ các Docker images. Docker Hub là một registry công cộng mà bất kỳ ai cũng có thể sử dụng, và Docker được cấu hình để sử dụng các images trên Docker Hub theo mặc định. Chúng ta cũng có thể thiết lập và lưu trữ repository riêng của mình trên Docker, tạo ra một Docker Registry cá nhân.
- **Docker Images:** là các mẫu chỉ đọc, được xây dựng dựa trên các hướng dẫn chi tiết trong Dockerfile. Những hình ảnh này không chỉ xác định cách thức đóng gói ứng dụng của chúng ta và các phần phụ thuộc mà còn mô tả các quy trình sẽ chạy khi ứng dụng được khởi động. Hãy hình dung Docker image như một bản thiết kế hoàn hảo của ứng dụng, chứa đựng tất cả từ mã nguồn, thư viện cho đến các cài đặt hệ thống cần thiết, sẵn sàng để triển khai bất cứ khi nào và ở bất cứ đâu.

2.3.5.2. Tại sao lại nói Docker đưa công nghệ lên tầm cao mới

- **Khả năng di chuyển cao:** Docker đảm bảo rằng các ứng dụng chạy mượt mà trên mọi hệ thống, từ máy tính cá nhân đến các máy chủ đám mây, nhờ vào sự nhất quán trong môi trường container. Điều này giúp giảm thiểu các vấn đề phát sinh do sự khác biệt giữa các môi trường phát triển và sản xuất.

- **Tốc độ và hiệu suất:** việc khởi động các container Docker nhanh chóng và nhẹ nhàng hơn nhiều so với các máy ảo truyền thống, giúp tăng cường hiệu suất và giảm thời gian triển khai. Docker giúp tối ưu hóa tài nguyên, cho phép triển khai nhiều container trên cùng một máy chủ mà không cần lo lắng về hiệu suất.
- **Môi Trường cách ly và bảo mật:** mỗi container Docker hoạt động trong một môi trường cách ly, đảm bảo rằng các ứng dụng không gây ảnh hưởng lẫn nhau. Điều này không chỉ nâng cao độ bảo mật mà còn giúp dễ dàng kiểm soát và quản lý các phiên bản ứng dụng.

2.3.5.3. Nhũng thách thức cần vượt qua

- **Quản lý phức tạp:** quản lý hàng trăm hoặc hàng ngàn container Docker có thể trở nên phức tạp và đòi hỏi một hệ thống quản lý container mạnh mẽ như Kubernetes. Sự phức tạp này đòi hỏi đội ngũ phát triển phải có kỹ năng và kiến thức chuyên sâu.
- **Hiệu năng hạn chế trên một số ứng dụng:** Docker không phải lúc nào cũng là giải pháp tối ưu cho mọi loại ứng dụng, đặc biệt là những ứng dụng yêu cầu hiệu năng cực cao hoặc có tính toán phức tạp về mặt tài nguyên.

2.3.6. IntelliJ IDEA

2.3.6.1. Giới thiệu

IntelliJ IDEA là một môi trường phát triển tích hợp (IDE) đầy sức mạnh, được thiết kế bởi JetBrains để hỗ trợ lập trình viên phát triển ứng dụng một cách hiệu quả và nhanh chóng. Đây là một trong những công cụ được ưa chuộng nhất trong giới phát triển phần mềm, đặc biệt là đối với các dự án Java.



Hình 2.3.6.1. Nhũng tính năng nổi bật của IntelliJ IDEA

2.3.6.2. Những điều kỳ diệu mà IntelliJ mang lại

IntelliJ IDEA không chỉ đơn thuần là một môi trường phát triển tích hợp (IDE), nó là một người bạn đồng hành thông minh và đáng tin cậy của lập trình viên.

- **Hoàn thành mã thông minh:** IntelliJ IDEA mang đến một trải nghiệm viết mã nhanh chóng và chính xác với tính năng hoàn thành mã thông minh. Không chỉ dừng lại ở việc gợi ý các từ khóa, nó hiểu rõ ngữ cảnh và cung cấp các gợi ý hợp lý dựa trên mã chúng ta đang viết. Điều này giúp tiết kiệm thời gian và giảm thiểu lỗi.
- **Điều hướng mã siêu tốc:** Với IntelliJ IDEA, chúng ta có thể di chuyển qua các tệp, lớp, phương thức và biến trong dự án của mình một cách dễ dàng và nhanh chóng. Tính năng điều hướng mã giúp chúng ta tìm thấy ngay những gì chúng ta cần mà không cần phải lục lọi qua hàng tá tệp tin.
- **Phân tích mã tự động:** Một trong những tính năng kỳ diệu của IntelliJ IDEA là khả năng phân tích mã tự động. Nó liên tục kiểm tra mã của chúng ta để phát hiện lỗi và cung cấp các gợi ý cải tiến ngay lập tức. Nhờ đó, chúng ta có thể duy trì chất lượng mã cao và khắc phục lỗi nhanh chóng.
- **Refactoring mạnh mẽ:** Tính năng refactoring của IntelliJ IDEA giúp chúng ta tái cấu trúc mã một cách an toàn và hiệu quả. Chúng ta có thể đổi tên biến, phương thức, lớp, trích xuất mã thành phương thức mới và nhiều hơn nữa chỉ với vài cú nhấp chuột. Mã của chúng ta sẽ luôn sạch sẽ và dễ bảo trì.
- **Hỗ trợ đa ngôn ngữ:** Không chỉ dành riêng cho Java, IntelliJ IDEA hỗ trợ nhiều ngôn ngữ lập trình khác như Kotlin, Scala, Groovy, và các ngôn ngữ phát triển web. Điều này làm cho IntelliJ IDEA trở thành một công cụ phát triển đa năng, phù hợp với nhiều dự án khác nhau.
- **Tích hợp công cụ phát triển hiện đại:** IntelliJ IDEA tích hợp mượt mà với các công cụ phát triển hiện đại như Git, Docker, và Kubernetes. Chúng ta có thể quản lý mã nguồn, xây dựng và triển khai ứng dụng ngay trong IDE mà không cần chuyển đổi giữa nhiều công cụ khác nhau.

2.3.7.3. Những hạn chế mà chúng ta cần biết

Dù IntelliJ IDEA được nhiều lập trình viên ca ngợi như một công cụ "thần thánh" với vô số tính năng mạnh mẽ, nhưng không phải là không có những điểm trừ.

- **Yêu cầu tài nguyên cao:** IntelliJ IDEA không phải là lựa chọn tốt nhất cho những chiếc máy tính "cổ". Với yêu cầu tài nguyên hệ thống cao, IDE này có thể biến máy tính của chúng ta thành "rùa bò" nếu cấu hình không đủ mạnh. Điều này đặc biệt khó chịu khi làm việc với các dự án lớn, nơi mà mỗi thao tác đều phải chờ đợi.
- **Giá cả đắt đỏ:** Phiên bản Ultimate của IntelliJ IDEA, mặc dù đầy đủ tính năng, lại có mức giá không hề dễ chịu. Với các lập trình viên tự do, sinh viên, hoặc những startup

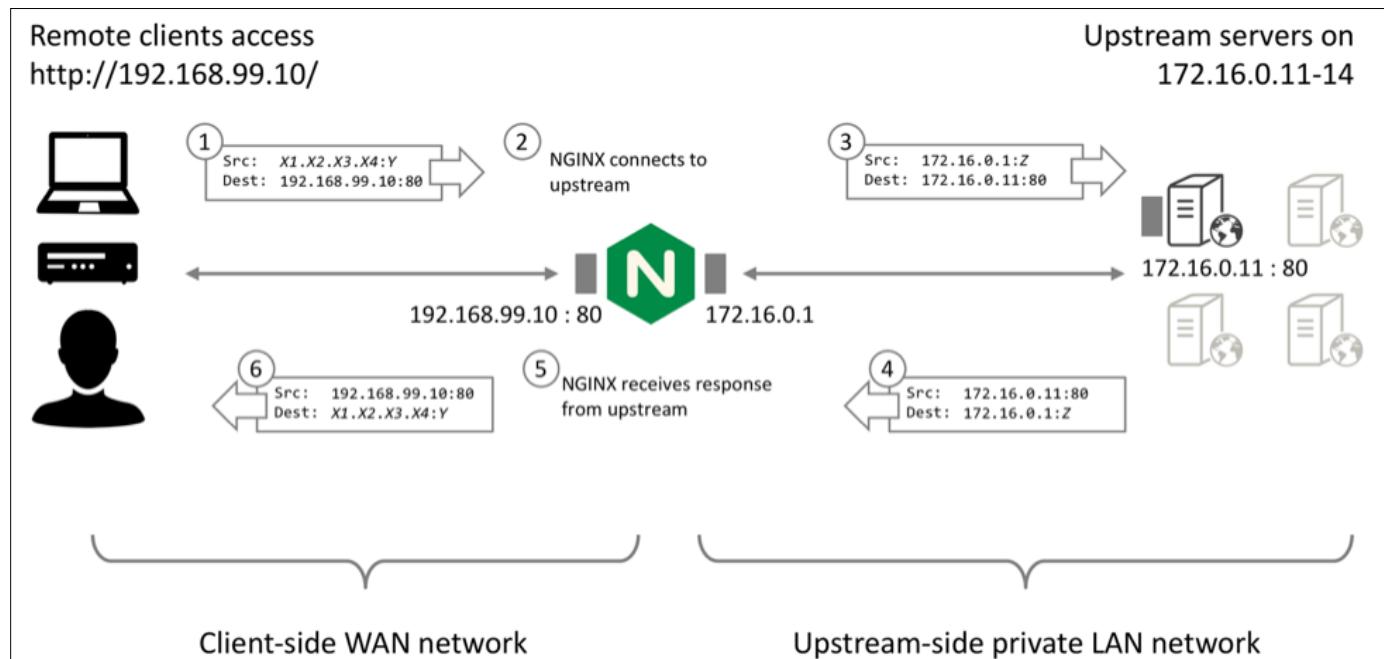
với ngân sách eo hẹp, việc đầu tư vào một IDE đắt đỏ như vậy có thể là một bài toán khó.

- **Độ phức tạp khi sử dụng:** IntelliJ IDEA như một "mê cung tính năng" mà người mới bắt đầu dễ dàng lạc lối. Để khai thác hết tiềm năng của IDE này, chúng ta cần một thời gian dài để học và làm quen. Đôi khi, sự đa dạng và phức tạp lại trở thành rào cản, khiến người dùng cảm thấy choáng ngợp.

2.3.7. Nginx

2.3.7.1. Giới thiệu

Nginx là một phần mềm máy chủ web mã nguồn mở được thiết kế để xử lý một lượng lớn yêu cầu một cách hiệu quả và ổn định. Ra mắt lần đầu tiên vào năm 2004 bởi lập trình viên người Nga Igor Sysoev, Nginx đã nhanh chóng trở thành một trong những máy chủ web phổ biến nhất trên thế giới. Được biết đến với hiệu suất cao, khả năng xử lý đồng thời hàng nghìn kết nối và tính năng cân bằng tải mạnh mẽ, Nginx không chỉ giới hạn trong vai trò máy chủ web mà còn được sử dụng rộng rãi làm proxy ngược (reverse proxy), cân bằng tải (load balancer), và cache HTTP.



Hình 2.3.7.1. Minh họa cách Nginx hoạt động

2.3.7.2. Lợi ích của Nginx mang lại

Nginx không chỉ đơn thuần là một máy chủ web và reverse proxy, mà còn là một công cụ mạnh mẽ giúp bạn xây dựng và quản lý các ứng dụng web hiện đại. Trong đó:

- **Hiệu suất cao**
 - Xử lý nhiều kết nối đồng thời: nginx được thiết kế để xử lý hàng nghìn kết nối đồng thời, làm cho nó phù hợp cho các trang web có lượng truy cập lớn.

- o Tối ưu bộ nhớ và CPU: sử dụng kiến trúc sự kiện phi đồng bộ (asynchronous event-driven), Nginx tiêu tốn ít tài nguyên hệ thống hơn so với các máy chủ web truyền thống như Apache.

- **Cân bằng tải (Load Balancing)**

- o Phân phối yêu cầu: nginx có khả năng phân phối yêu cầu đến nhiều máy chủ backend, giúp tối ưu hóa hiệu suất và độ tin cậy của hệ thống.
- o Thuật toán linh hoạt: Hỗ trợ nhiều thuật toán cân bằng tải như round-robin, least connections, và IP hash, giúp điều chỉnh linh hoạt theo nhu cầu thực tế.

- **Reverse Proxy**

- o Bảo vệ Backend: nginx đóng vai trò như một lớp bảo vệ giữa client và các máy chủ backend, giúp tăng cường bảo mật và giảm tải cho các máy chủ backend.
- o Tăng tốc độ phản hồi: giảm độ trễ và tăng tốc độ phản hồi bằng cách caching các tài nguyên tĩnh và proxy cho các yêu cầu động.

2.3.7.3. Những hạn chế có thể gặp phải

- **Yêu cầu kiến thức kỹ thuật**

- o Cấu Hình Ban Đầu Phức Tạp: Cấu hình ban đầu của Nginx có thể phức tạp đối với người mới bắt đầu, đặc biệt là khi cần tối ưu hóa cho hiệu suất và bảo mật.
- o Đòi Hỏi Hiểu Biết Về Hệ Thống: Việc tối ưu hóa và khai thác toàn bộ khả năng của Nginx yêu cầu hiểu biết sâu rộng về hệ thống và mạng.

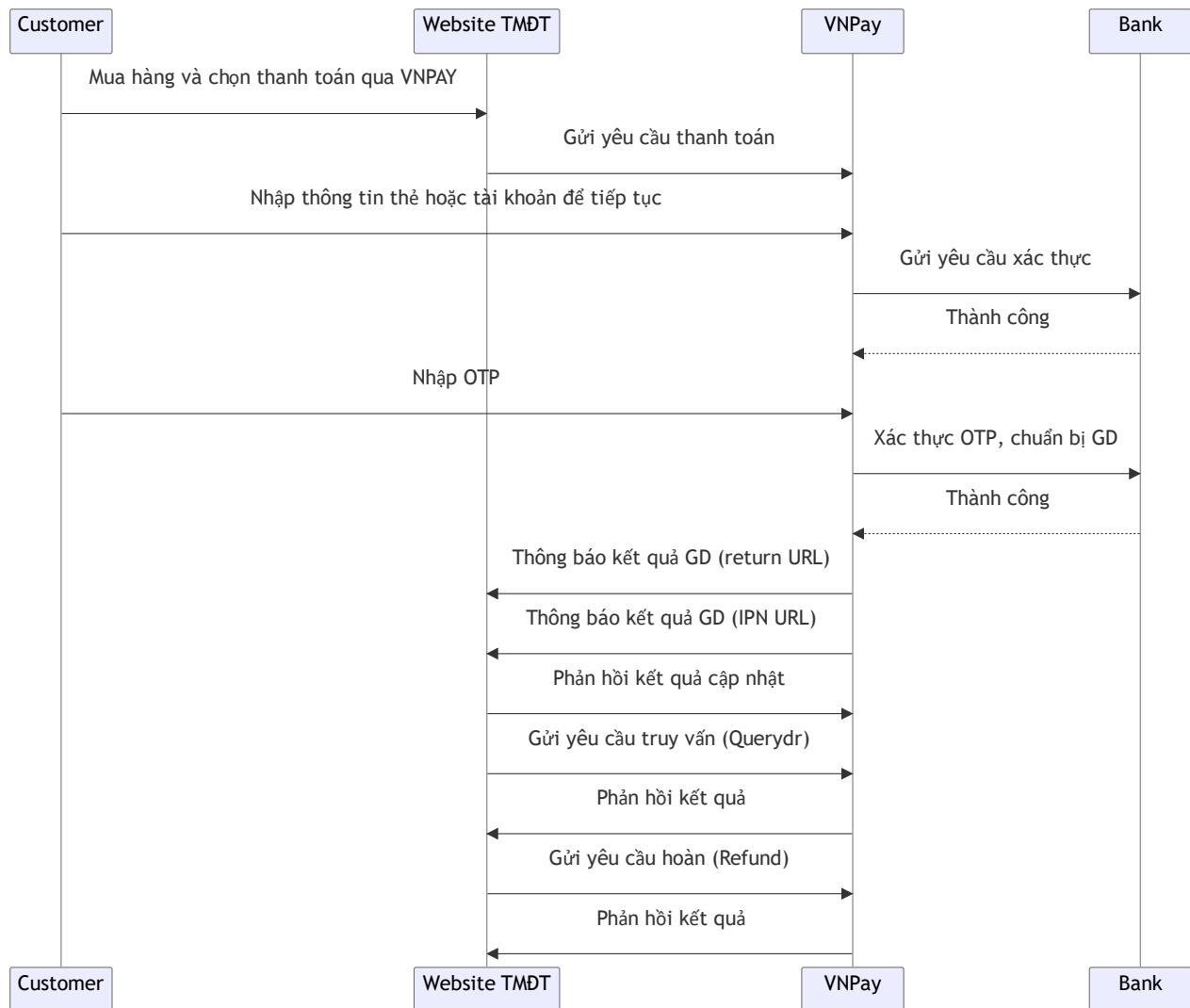
- **Khả năng hỗ trợ**

- o Cộng Đồng và Tài Liệu: Mặc dù cộng đồng Nginx rất lớn và tài liệu phong phú, nhưng một số vấn đề cụ thể có thể yêu cầu tìm kiếm kỹ hoặc hỏi đáp từ cộng đồng.

2.4. Các kỹ thuật và công cụ liên quan

2.4.1. Thanh toán trực tuyến VNPay

Trong các website TMĐT và nền tảng mua bán trực tuyến, việc tích hợp các phương thức thanh toán trực tuyến là yếu tố quan trọng hàng đầu. Điều này không chỉ tạo sự tiện lợi cho người dùng mà còn nâng cao trải nghiệm mua sắm, gia tăng độ tin cậy và khuyến khích khách hàng quay lại sử dụng dịch vụ. Một hệ thống thanh toán trực tuyến an toàn, đa dạng và linh hoạt là chìa khóa để thu hút và giữ chân khách hàng trong môi trường cạnh tranh khốc liệt của thị trường TMĐT.



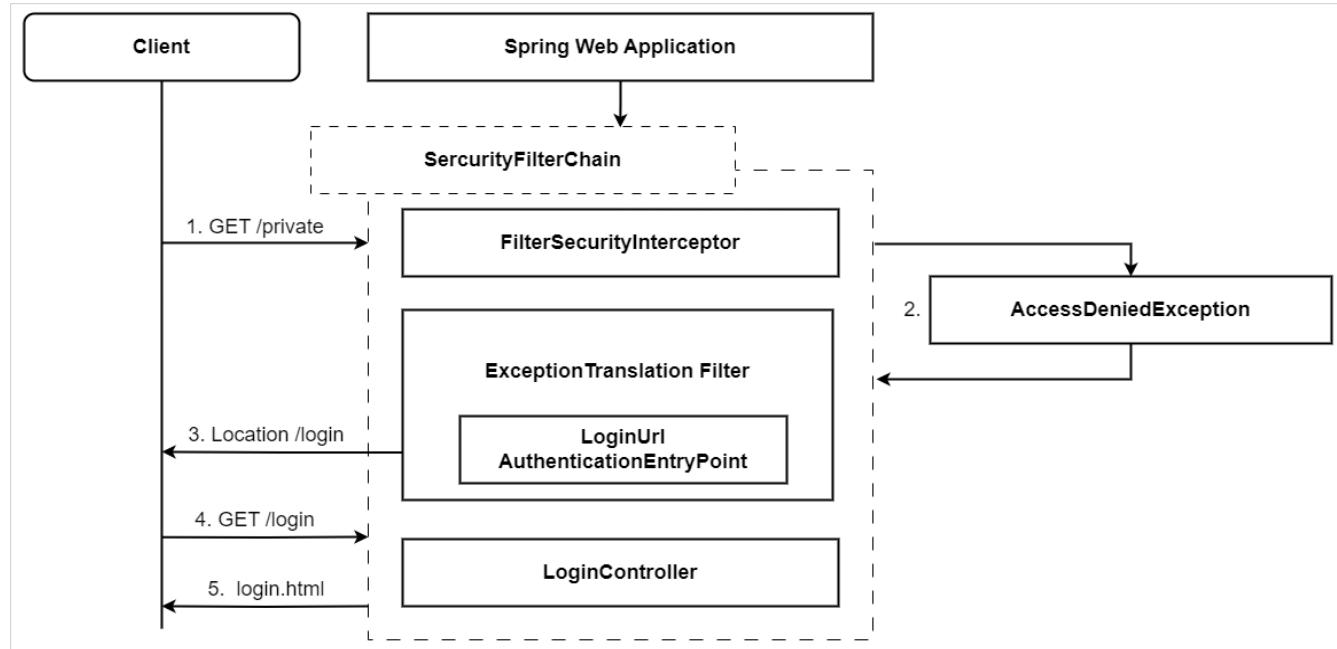
Hình 2.4.1. Sơ đồ tuần tự các bước Thanh toán trực tuyến VNPAY
Các bước thực hiện (hình 2.4.1):

- **Bước 1:** Khách hàng tạo đơn hàng trên Website và chọn thanh toán qua VNPAY
- **Bước 2:** Website thành lập một yêu cầu thanh toán dưới dạng URL chứa thông tin thanh toán và chuyển hướng khách hàng sang URL đó (Cổng thanh toán VNPAY). Cổng thanh toán VNPAY xử lý yêu cầu Thanh toán từ Website.
- **Bước 3:** Khách hàng nhập thông tin để xác minh tài khoản ngân hàng.
- **Bước 4:** Nhập OTP để xác thực giao dịch.
- **Bước 5:** Giao dịch hoàn tất (thất bại hoặc thành công), VNPAY tiến hành chuyển hướng khách hàng về Website (vnp_ReturnUrl), thông báo cho Website kết quả thanh toán của khách hàng thông qua IPN URL.
- **Bước 6:** Website hiển thị kết quả giao dịch tới khách hàng.

2.4.2. Spring Security

2.4.2.1. Tổng quan

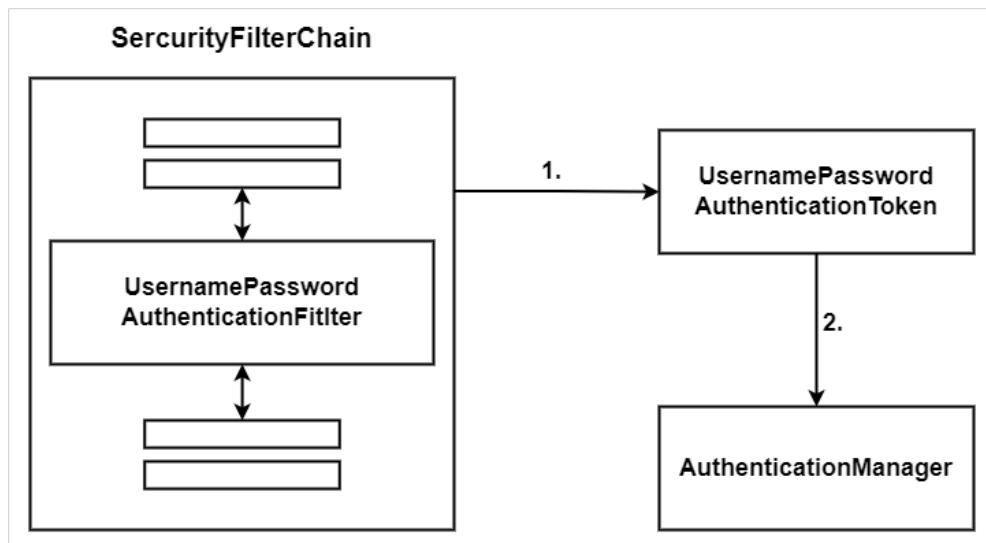
Spring Security là một framework mạnh mẽ trong hệ sinh thái của Spring được sử dụng để xác thực và ủy quyền người dùng trong ứng dụng Java. Nó hoạt động như một người bảo vệ thận trọng cho các ứng dụng web của chúng ta, giám sát ai có thể truy cập trang web của chúng ta, xác minh danh tính người dùng và thực thi các quy tắc để bảo vệ dữ liệu của chúng ta khỏi bị truy cập trái phép.



Hình 2.4.2.1.1. Chuyển hướng đến trang Login

Quy trình SecurityFilterChain khi người dùng yêu cầu truy cập tài nguyên bảo vệ (hình 2.4.2.1.1):

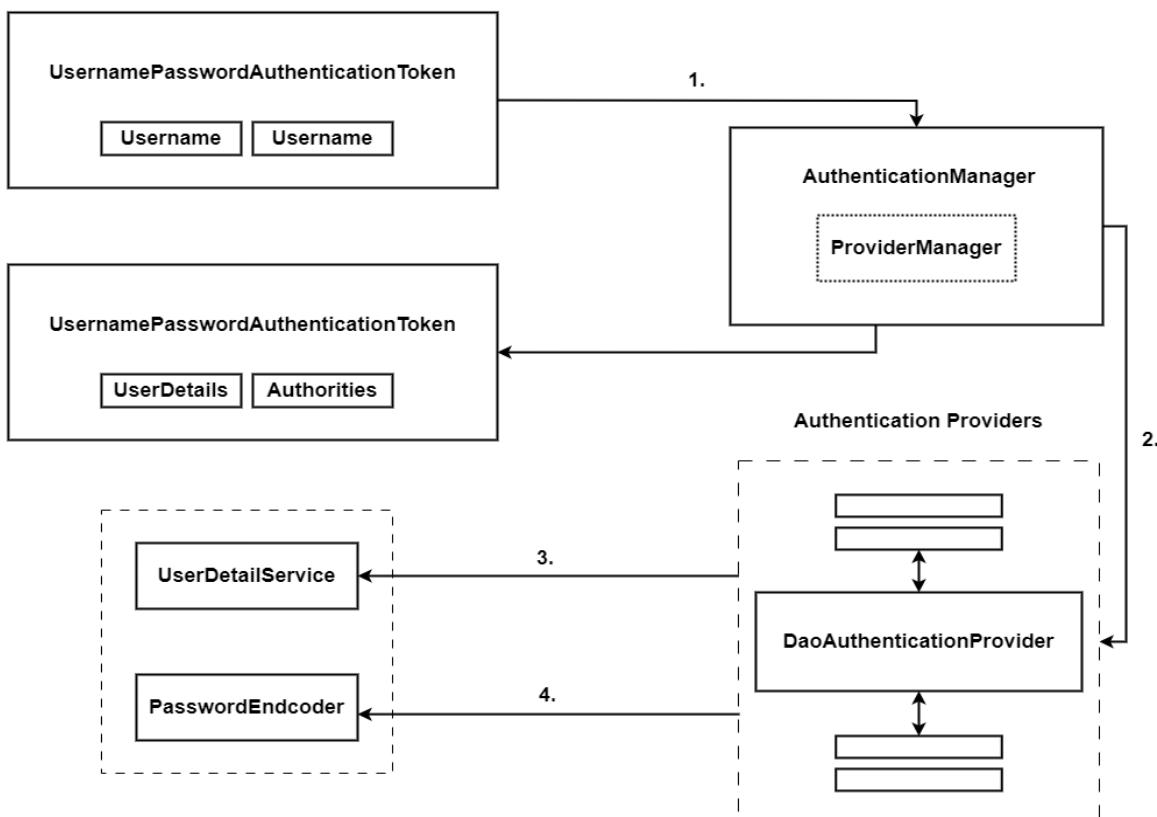
- (1) Đầu tiên người dùng gửi yêu cầu tới một tài nguyên bảo vệ (ví dụ /private), nhưng yêu cầu này không có thông tin xác thực.
- (2) AuthorizationFilter trong chuỗi bảo mật của Spring Security sẽ kiểm tra và nhận thấy rằng yêu cầu không được xác thực. Do đó, yêu cầu bị từ chối và AccessDeniedException được ném ra.
- (3) Vì người dùng chưa được xác thực, ExceptionTranslationFilter can thiệp và bắt đầu quá trình xác thực. Bộ lọc này sẽ gửi chuyển hướng (redirect) đến trang đăng nhập bằng cách sử dụng AuthenticationEntryPoint đã được cấu hình trước. Thông thường, Authentication EntryPoint là một phiên bản của LoginUrlAuthenticationEntryPoint, được thiết lập để chuyển hướng người dùng đến trang đăng nhập.
- (4) Trình duyệt của người dùng sẽ theo chuyển hướng và gửi một yêu cầu đến trang đăng nhập.
- (5) Ứng dụng sẽ xử lý yêu cầu trang đăng nhập và hiển thị giao diện đăng nhập cho người dùng. Người dùng sẽ nhập tên đăng nhập và mật khẩu của mình trên trang này.



Hình 2.4.2.1.2. Xử lý xác thực bởi username và password

Quy trình xác thực sử dụng UsernamePasswordAuthenticationFilter (Hình 2.4.2.1.2)

- (1) Khi người dùng gửi thông tin đăng nhập thì UsernamePasswordAuthenticationFilter trích xuất thông tin từ HttpServletRequest và tạo UsernamePasswordAuthentication Token.
- (2) UsernamePasswordAuthenticationToken được chuyển tới AuthenticationManager để thực hiện xác thực.



Hình 2.4.2.1.3. Quy trình xử lý DaoAuthenticationProvider

AuthenticationManager là giao diện trung tâm trong Spring Security chịu trách nhiệm quản lý quy trình xác thực.

- (1) AuthenticationManager nhận các yêu cầu xác thực (chứa thông tin đăng nhập của người dùng) từ UsernamePasswordAuthenticationToken.
- (2) AuthenticationManager sử dụng DaoAuthenticationProvider để xử lý xác thực.
- (3) DaoAuthenticationProvider tra cứu thông tin người dùng từ UserDetailsService.
- (4) DaoAuthenticationProvider sử dụng PasswordEncoder để so sánh mật khẩu được cung cấp với mật khẩu được lưu trữ. AuthenticationManager nhận đối tượng Authentication đã xác thực từ Dao AuthenticationProvider và chuyển nó lại cho UsernamePassword AuthenticationFilter. UsernamePasswordAuthenticationFilter sau đó lưu trữ đối tượng Authentication vào SecurityContext.

2.4.2.2. Ưu điểm

- Spring Security hỗ trợ nhiều cơ chế xác thực và ủy quyền khác nhau như xác thực dựa trên Form, Basic Authentication, OAuth2, JWT, và nhiều hơn nữa. Để dàng mở rộng và tùy chỉnh các thành phần bảo mật để phù hợp với nhu cầu cụ thể của ứng dụng.
- Spring Security sử dụng các thuật toán mã hóa mạnh mẽ như BCrypt, SCrypt và Argon2 để mã hóa mật khẩu. Những thuật toán này được thiết kế để chống lại các cuộc tấn công brute-force và rainbow table, giúp bảo vệ mật khẩu người dùng một cách hiệu quả.

2.4.2.3. Nhược điểm

- Các lỗ hổng bảo mật mới liên tục được phát hiện. Do đó, cần phải cập nhật thường xuyên các phiên bản mới nhất của Spring Security để đảm bảo an toàn. Việc duy trì và cập nhật các cấu hình bảo mật đòi hỏi sự chú ý và nỗ lực liên tục từ phía các nhà phát triển.
- Mặc dù Spring Boot giúp đơn giản hóa nhiều cấu hình, nhưng đối với các yêu cầu bảo mật phức tạp, việc cấu hình Spring Security vẫn có thể đòi hỏi nhiều nỗ lực và kiến thức chuyên sâu. Việc tùy chỉnh sâu có thể dẫn đến các cấu hình dài và phức tạp.

2.4.3. Spring Data JPA

2.4.3.1. Giới thiệu

Spring Data JPA (Java Persistence API) là một phần của dự án Spring Data, cung cấp một lớp trừu tượng để làm việc với cơ sở dữ liệu quan hệ. Spring Data JPA giúp đơn giản hóa và tăng hiệu quả công việc với các thao tác cơ bản như lưu trữ, cập nhật, xóa và truy vấn dữ liệu. Một điều cần lưu ý: JPA là một Interface trong khi Hibernate là một Implementation.

2.4.3.2. Ưu điểm

- Giảm thiểu mã nguồn phải viết cho các thao tác CRUD thông qua các repository mặc định và các phương thức truy vấn tự động. Tích hợp sâu với các thành phần khác của Spring Framework như Spring Boot, Spring.
- Security, Spring MVC, giúp xây dựng các ứng dụng web và microservices một cách nhanh chóng và hiệu quả. Hỗ trợ nhiều cơ sở dữ liệu quan hệ như PostgreSQL, H2, MySQL, Oracle, SQLServer, và nhiều cơ sở dữ liệu khác.

2.4.3.3. Nhược điểm

- Việc sử dụng các phương thức truy vấn tự động và các lớp trừu tượng có thể gây ra một số vấn đề về hiệu suất, đặc biệt khi làm việc với lượng dữ liệu lớn hoặc các truy vấn phức tạp.
- Cần tối ưu hóa và kiểm tra kỹ lưỡng các truy vấn để đảm bảo hiệu suất tốt nhất.

2.4.4. Thymeleaf

2.4.6.1. Giới thiệu

Thymeleaf là một công cụ mẫu (template engine) dành cho các ứng dụng Java, đặc biệt là các ứng dụng web. Nó cho phép phát triển giao diện web bằng cách kết hợp mã HTML với mã Java để tạo ra các trang web động.

2.4.6.2. Lý do sử dụng Thymeleaf

• *Tích hợp tuyệt vời với Spring Framework:*

Thymeleaf được thiết kế để tích hợp dễ dàng với Spring Framework, đặc biệt là Spring Boot. Sự tích hợp này giúp việc cấu hình và triển khai ứng dụng trở nên đơn giản và nhanh chóng. Chúng ta có thể sử dụng các tính năng của Spring như Dependency Injection, Spring Security, và Spring Data một cách mượt mà với Thymeleaf.

• *Hỗ trợ tự nhiên cho HTML5:*

Thymeleaf cho phép chúng ta viết các mẫu HTML5 một cách tự nhiên và rõ ràng. Các mẫu này có thể được kiểm tra và hiển thị như các tài liệu HTML tĩnh, giúp cho việc phát triển giao diện người dùng dễ dàng và trực quan hơn. Chúng ta không cần phải sử dụng các tag hoặc cú pháp phức tạp, chỉ cần sử dụng các thuộc tính của HTML.

2.5. Các công cụ hỗ trợ kiểm thử

2.5.1. Postman

2.5.1.1. Giới thiệu

Postman hiện là một trong những công cụ phổ biến nhất được sử dụng trong thử nghiệm các API. Với Postman, ta có thể gọi Rest API mà không cần viết dòng code nào.

Một số chức năng cơ bản:

- Cho phép gửi HTTP Request với các method GET, POST, PUT, DELETE.
- Cho phép post dữ liệu dưới dạng form (key-value), text, json.
- Hiện kết quả trả về dạng text, hình ảnh, XML, JSON.

- Cho phép thay đổi header của các request.

2.5.1.2. Lợi thế

Quản lý tốt các collections và environments giúp tổ chức và tái sử dụng các yêu cầu API một cách hiệu quả. Tích hợp dễ dàng với các công cụ phát triển và kiểm thử khác, giúp quy trình làm việc trở nên liên tục và hiệu quả. Hỗ trợ CI/CD giúp dễ dàng tự động hóa quy trình kiểm thử.

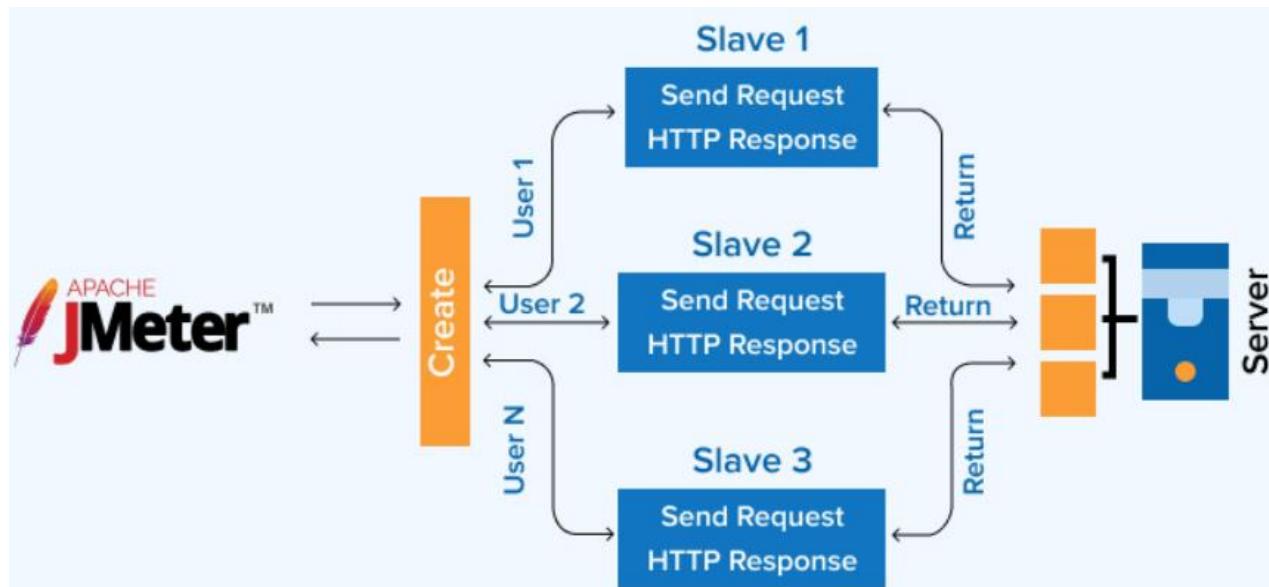
2.5.1.3. Bất lợi

Khi làm việc với các collections lớn hoặc nhiều tab mở cùng lúc, Postman có thể gặp vấn đề về hiệu suất, gây chậm trễ hoặc treo ứng dụng.

2.5.2. Apache JMeter

2.5.2.1. Giới thiệu

Apache JMeter là một công cụ kiểm thử mã nguồn mở mạnh mẽ và phổ biến, được đông đảo nhiều chuyên gia sử dụng để đánh giá hiệu năng và khả năng chịu tải của các ứng dụng. Được phát triển bởi Apache Software Foundation, JMeter ban đầu được tạo ra để kiểm thử các ứng dụng web. Tuy nhiên, qua thời gian, JMeter đã mở rộng khả năng của mình, cho phép kiểm thử hiệu năng của nhiều loại dịch vụ khác nhau, bao gồm cơ sở dữ liệu, dịch vụ web, giao thức TCP, FTP và nhiều loại dịch vụ khác nữa.



Hình 2.5.2.1.1. Môi trường thử nghiệm phân tán (Distributed test environment)

2.5.2.2. Lợi thế

Apache JMeter là một công cụ mã nguồn mở được thiết kế để kiểm thử hiệu năng và tải của các ứng dụng, đặc biệt là các ứng dụng web. Được phát triển bởi Apache Software Foundation, JMeter giúp các nhà phát triển và kiểm thử viên đánh giá khả năng chịu tải, hiệu suất và độ tin cậy của các ứng dụng dưới nhiều điều kiện khác nhau.

Các tính năng của JMeter:

- **Kiểm thử tải (Load Testing):** mô phỏng nhiều người dùng ảo truy cập vào hệ thống đồng thời để đánh giá hiệu suất.
- **Kiểm thử áp lực (Stress Testing):** tăng dần tải để xác định điểm phá vỡ của hệ thống.
- **Kiểm thử khả năng chịu đựng (Endurance Testing):** kiểm tra khả năng hệ thống hoạt động dưới tải trong một khoảng thời gian dài.
- **Kiểm thử chức năng (Functional Testing):** đảm bảo các chức năng cụ thể của ứng dụng hoạt động đúng.
- **Hỗ trợ đa giao thức:** HTTP, HTTPS, FTP, JDBC, LDAP, JMS, POP3, IMAP, SMTP.

2.5.2.3 Bất lợi

Mặc dù Apache JMeter là một công cụ kiểm thử tải và hiệu năng mạnh mẽ và phổ biến, nhưng nó cũng có một số hạn chế và bất lợi như:

- **Sử dụng nhiều tài nguyên:** khi kiểm thử với số lượng lớn người dùng ảo, JMeter có thể tiêu tốn nhiều tài nguyên CPU và bộ nhớ. Điều này có thể giới hạn khả năng mô phỏng lượng lớn người dùng trên một máy duy nhất.
- **Phân phối tải:** để kiểm thử với hàng ngàn người dùng ảo, chúng ta có thể cần phải sử dụng tính năng kiểm thử phân tán của JMeter, yêu cầu cấu hình phức tạp và quản lý nhiều máy chủ kiểm thử.

CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

3.1. Đặc tả hệ thống

3.1.1. Nhiệm vụ

Xây dựng một hệ thống thương mại điện tử với các chức năng cơ bản của một hệ thống bán hàng. Dưới đây là một số điểm nổi bật mà hệ thống này mang lại:

- Tối ưu hóa trải nghiệm người dùng.
- Quản lý dễ dàng các hoạt động mua bán.
- Đề xuất sản phẩm.
- Bảo mật và an toàn thông tin.
- Tính năng cá nhân hóa.
- Đa thiết bị.
- Khả năng mở rộng, bảo trì, sửa chữa và nâng cấp.

3.1.2. Quy trình nghiệp vụ

• Khách hàng truy cập trang web

- *Truy cập và duyệt web:* khách hàng truy cập trang web, duyệt qua các danh mục sản phẩm, và tìm kiếm sản phẩm mà họ quan tâm.
- *Đăng ký/Đăng nhập:* khách hàng có thể tạo tài khoản hoặc đăng nhập nếu đã có tài khoản. Nếu khách hàng muốn mua sản phẩm thì bắt buộc yêu cầu đăng nhập.

• Tìm kiếm và chọn sản phẩm

- *Tìm kiếm sản phẩm:* khách hàng sử dụng thanh tìm kiếm hoặc duyệt qua các danh mục để tìm sản phẩm.
- *Lọc và sắp xếp:* khách hàng sử dụng các bộ lọc (giá, loại, màu, v.v.) để thu hẹp kết quả tìm kiếm.
- *Chi tiết sản phẩm:* khách hàng nhấp vào sản phẩm để xem chi tiết, bao gồm mô tả, hình ảnh, đánh giá và giá cả.

• Thêm sản phẩm vào giỏ hàng

- *Chọn sản phẩm:* khách hàng chọn số lượng và các thuộc tính sản phẩm (màu sắc, kích thước, v.v.) rồi thêm vào giỏ hàng.
- *Xem giỏ hàng:* khách hàng có thể xem và chỉnh sửa giỏ hàng của mình trước khi tiến hành thanh toán.

• Tiến hành thanh toán

- *Xác nhận giỏ hàng:* khách hàng kiểm tra lại các sản phẩm trong giỏ hàng và xác nhận mua hàng.
- *Thông tin giao hàng:* khách hàng nhập thông tin giao hàng, bao gồm địa chỉ và phương thức giao hàng.

- o *Lựa chọn phương thức thanh toán:* khách hàng chọn phương thức thanh toán (thẻ tín dụng, ví điện tử, chuyển khoản ngân hàng, COD, v.v.).

- **Xử lý đơn hàng**

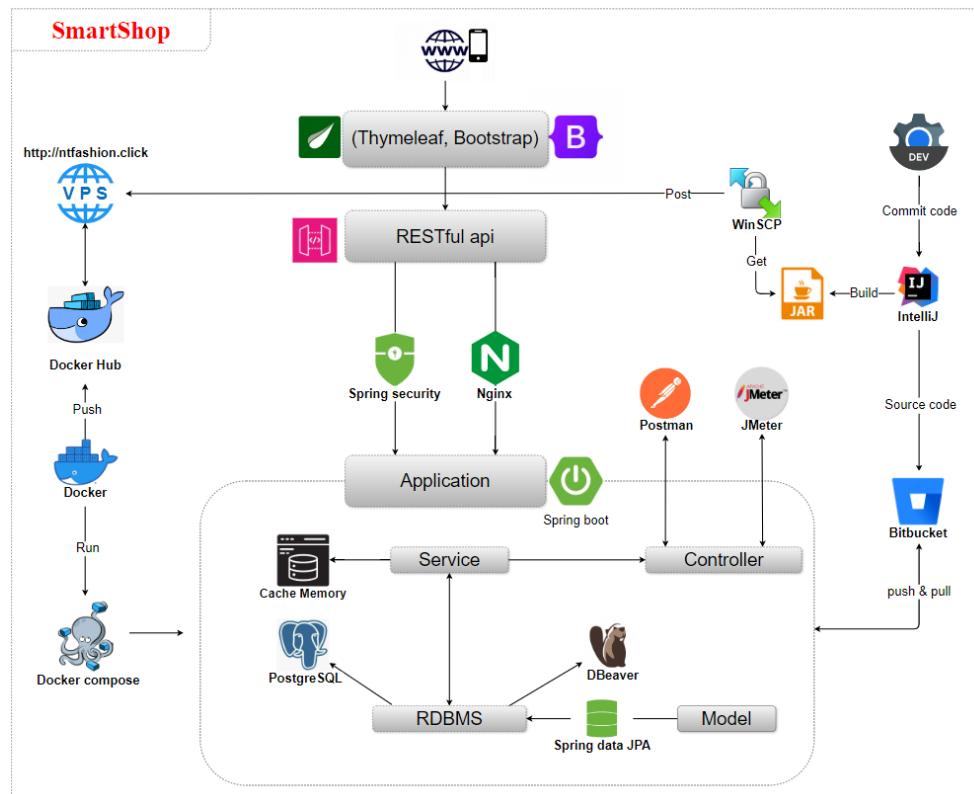
- o *Xác nhận đơn hàng:* hệ thống gửi thông báo qua email hoặc SMS cho khách hàng.
- o *Kiểm tra kho hàng:* hệ thống kiểm tra tồn kho, xác nhận tính khả dụng của sản phẩm.
- o *Chuẩn bị hàng:* bộ phận kho hàng chuẩn bị sản phẩm để giao.

- **Giao hàng**

- o *Giao cho đơn vị vận chuyển:* sản phẩm được bàn giao cho đơn vị vận chuyển.
- o *Theo dõi giao hàng:* khách hàng có thể theo dõi tình trạng giao hàng qua mã vận đơn hoặc trang web của đơn vị vận chuyển.
- o *Nhận hàng:* khách hàng nhận sản phẩm và ký nhận.

3.1.3. Tổng quan hệ thống (Big Picture)

Để mang đến một cái nhìn tổng quan sinh động, chúng em đã thiết kế một bức tranh tổng thể về hệ thống, minh họa rõ ràng các thành phần chính và mối quan hệ giữa chúng. Bức tranh này giúp mọi người dễ dàng hiểu rõ cấu trúc và cách thức hoạt động của hệ thống một cách nhanh chóng. Hệ thống của chúng em được xây dựng dựa trên những công nghệ hiệu quả và tiên tiến, giúp đảm bảo hoạt động mượt mà, hiệu suất vượt trội, bảo mật cao và khả năng mở rộng dễ dàng. Những công nghệ này không chỉ nâng cao hiệu quả mà còn mang đến sự tin cậy và linh hoạt cho toàn bộ hệ thống.



Hình 3.1.1. Tổng quan hệ thống

Đặc tả tổng quát hệ thống (hình 3.1.3):

Nhóm	Đặc tả
Hạ tầng và triển khai	<p>VPS (Virtual Private Server): Toàn bộ hệ thống được triển khai trên một VPS, đảm bảo quyền kiểm soát cao, bảo mật tốt và hiệu suất ổn định.</p> <p>Docker, Docker compose & Docker Hub: Docker được sử dụng để đóng gói ứng dụng thành các container, đảm bảo môi trường triển khai nhất quán và dễ dàng mở rộng. Docker Compose quản lý và khởi động nhiều container cùng lúc. Hình ảnh Docker của ứng dụng được lưu trữ trên Docker Hub để thuận tiện cho việc quản lý và triển khai.</p>
Phát triển và tích hợp liên tục	<p>Bitbucket: Là nơi lưu trữ mã nguồn, giúp quản lý phiên bản và hợp tác nhóm hiệu quả.</p> <p>IntelliJ IDEA: Môi trường phát triển mạnh mẽ, hỗ trợ đầy đủ các tính năng giúp tăng năng suất lập trình.</p> <p>WinSCP: Công cụ chuyển tệp tin, giúp tải các tệp JAR lên VPS một cách dễ dàng và nhanh chóng.</p>
Kiến trúc ứng dụng	<p>NTFashion (Thymeleaf, Bootstrap): Giao diện người dùng được xây dựng với Thymeleaf và Bootstrap, mang lại trải nghiệm trực quan và thân thiện.</p> <p>RESTful API: Cung cấp các dịch vụ backend, cho phép giao tiếp giữa frontend và các thành phần khác của hệ thống một cách mượt mà.</p> <p>Spring Boot: Nền tảng chính của backend, cung cấp các tính năng mạnh mẽ và dễ sử dụng để phát triển ứng dụng.</p> <p>Spring Security: Bảo vệ hệ thống với các cơ chế xác thực và phân quyền nghiêm ngặt.</p>
Quản lý hiệu suất và tải	<p>Nginx: Được sử dụng như một reverse proxy để cân bằng tải và cải thiện khả năng chịu tải của hệ thống. Nginx đảm bảo ứng dụng luôn sẵn sàng và hoạt động mượt mà, ngay cả khi lượng truy cập tăng cao.</p> <p>Cache Memory: Bộ nhớ đệm giúp tăng tốc độ phản hồi và giảm tải cho cơ sở dữ liệu, tối ưu hóa hiệu suất hệ thống.</p>
Phân tầng ứng dụng	<p>Controller: Xử lý các yêu cầu từ người dùng, đảm bảo phản hồi nhanh chóng và chính xác.</p> <p>Service Layer: Chứa logic nghiệp vụ, tương tác với cơ sở dữ liệu và các thành phần khác.</p> <p>Model: Đại diện cho cấu trúc dữ liệu của ứng dụng, đảm bảo dữ liệu được quản lý và truy xuất hiệu quả.</p> <p>RDBMS (Relational Database Management System): cơ sở dữ liệu quan hệ sử dụng PostgreSQL.</p> <p>DBeaver: công cụ quản lý cơ sở dữ liệu, hỗ trợ tối ưu hóa và kiểm soát dữ liệu.</p>
Kiểm thử và đảm bảo chất lượng	<p>Postman: Công cụ kiểm thử API, đảm bảo các API hoạt động đúng chức năng và hiệu quả.</p> <p>Apache JMeter: Công cụ kiểm thử tải, đánh giá khả năng chịu tải và hiệu suất của hệ thống, đảm bảo hệ thống có thể xử lý lượng lớn yêu cầu mà không gặp vấn đề.</p>

3.2. Phân tích hệ thống

3.2.1. Yêu cầu ứng dụng

- **Môi trường hoạt động:** tất cả trình duyệt thông dụng và có thể truy cập trên các thiết bị di động, laptop.
- **Công nghệ phát triển:** sử dụng Spring boot, Spring Data JPA, Spring Security, Spring MVC, Thymeleaf, Bootstrap, RESTful API, Docker, DBeaver.
- **Lưu trữ dữ liệu:** sử dụng PostgreSQL.
- **Yêu cầu hiệu suất:**
 - Thời gian tải trang web: chậm nhất là 10 giây.
 - Thời gian xử lý các thao tác truy vấn dữ liệu: chậm nhất là 4 giây.
- **Yêu cầu về bảo mật:**
 - Mật khẩu người dùng được mã hóa trước khi được lưu vào cơ sở dữ liệu.
 - Không cho mã xác nhận, tên đăng nhập và mật khẩu được truyền trên môi trường mạng mà không được mã hóa.
 - Sử dụng kênh kết nối an toàn trong việc truy cập máy chủ ứng dụng.
 - Sao lưu và an toàn bảo mật dữ liệu.

3.2.2. Yêu cầu người dùng

3.2.2.1. Đối với khách hàng

- Mua sắm trực tuyến dễ dàng, thuận tiện và mượt mà. Bên cạnh đó giao diện người dùng thân thiện, dễ sử dụng, thân thiện phù hợp với đa dạng thiết bị.
- Mong muốn có nhiều sự lựa chọn sản phẩm từ nhiều danh mục khác nhau, đảm bảo rằng họ có thể tìm thấy sản phẩm phù hợp với nhu cầu và sở thích của mình.
- Thông tin chi tiết và chính xác về sản phẩm, bao gồm mô tả sản phẩm, hình ảnh, đánh giá và nhận xét từ các khách hàng trước đó.
- Quy trình thanh toán an toàn và tiện lợi.
- Thời gian giao hàng nhanh chóng và dịch vụ giao nhận đáng tin cậy.
- Thông tin cá nhân và thanh toán của họ được bảo mật và không bị lộ ra bên ngoài.
- Luôn cập nhật, giới thiệu những sản phẩm tương tự hoặc được mua nhiều.

3.2.2.2. Đối với quản trị viên

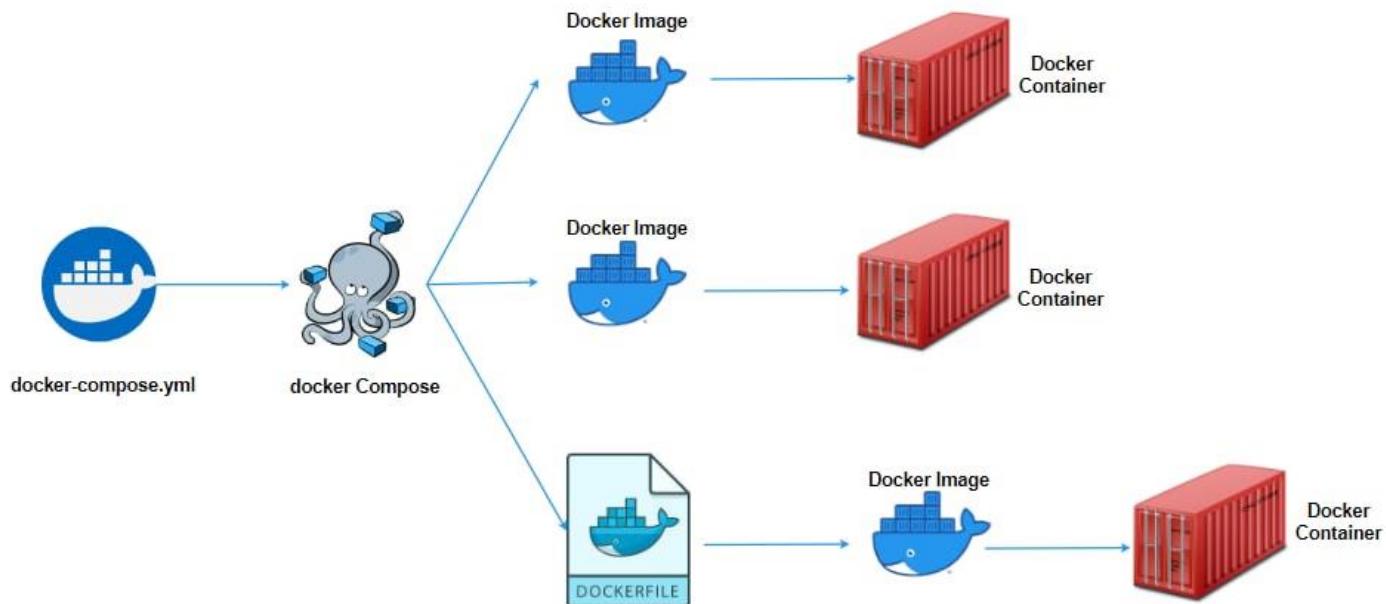
- Có khả năng quản lý sản phẩm và dịch vụ một cách hiệu như việc thêm, sửa đổi và xóa sản phẩm, cũng như quản lý danh mục sản phẩm.
- Xử lý và theo dõi đơn hàng, quản lý thông tin giao hàng và thanh toán, và tương tác với khách hàng thông qua email hoặc tin nhắn.
- Đảm bảo rằng dữ liệu và thông tin cá nhân của khách hàng được bảo vệ an toàn.
- Quản lý số lượng sản phẩm, khách hàng v.v.

3.3. Hạ tầng triển khai ứng dụng

3.3.1. Docker compose

3.3.1.1. Giới thiệu

Docker Compose là một công cụ mạnh mẽ và linh hoạt, giúp biến việc quản lý và triển khai các ứng dụng phức tạp trở nên đơn giản và hiệu quả hơn bao giờ hết. Với Docker Compose, chúng ta có thể định nghĩa, chạy và quản lý nhiều container Docker chỉ trong một tệp cấu hình **docker-compose.yml** duy nhất.



Hình 3.1.2. Cách thức hoạt động của Docker Compose

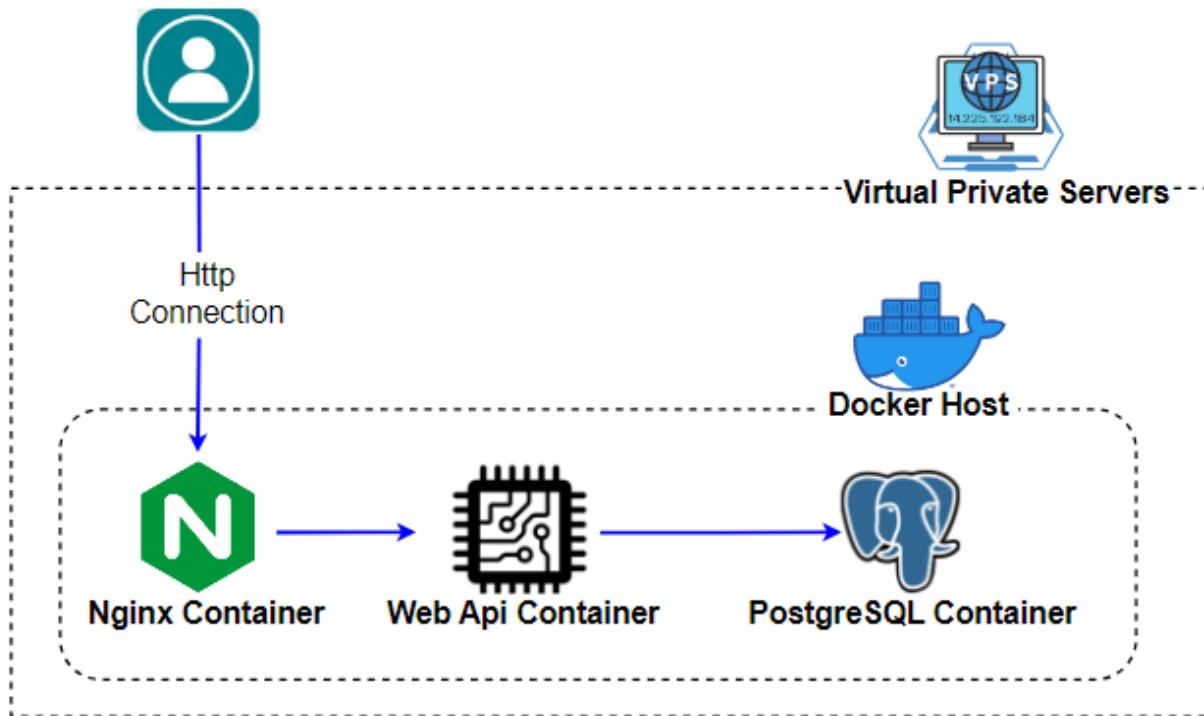
3.3.1.2. Lợi ích

- Docker Compose giúp quản lý vòng đời của các container một cách hiệu quả. Việc khởi động, dừng và cấu hình lại các container trở nên cực kỳ đơn giản với các lệnh Docker Compose.
- Đảm bảo rằng môi trường làm việc giữa các giai đoạn phát triển, thử nghiệm và triển khai là nhất quán. Điều này giảm thiểu các vấn đề không tương thích giữa các môi trường.
- Hỗ trợ việc mở rộng quy mô ứng dụng một cách dễ dàng. Khi nhu cầu tăng lên, các dịch vụ có thể được mở rộng bằng cách thay đổi cấu hình trong tệp YAML và khởi động lại các container.
- Với Docker Compose, các ứng dụng và dịch vụ được đóng gói trong các container, đảm bảo tính di động và có thể chạy nhất quán trên mọi hệ thống hỗ trợ Docker.

3.3.2. VPS (Virtual Private Server)

3.3.2.1. Giới thiệu

Virtual Private Server (VPS) là một giải pháp máy chủ ảo riêng biệt, mang lại nhiều lợi ích vượt trội cho việc triển khai và quản lý các ứng dụng. VPS không chỉ cung cấp tài nguyên riêng biệt mà còn mang lại sự linh hoạt và kiểm soát cao, phù hợp với nhiều loại dự án từ nhỏ đến lớn.



Hình 3.1.3. Triển khai VPS với Docker

3.3.2.2. Lợi ích

- Với VPS, chúng ta có quyền kiểm soát hoàn toàn cấu hình máy chủ, hệ điều hành và các ứng dụng cài đặt. Điều này cho phép chúng ta tối ưu hóa môi trường làm việc theo nhu cầu cụ thể của dự án.
- VPS cung cấp tài nguyên CPU, RAM và dung lượng lưu trữ riêng biệt, giúp các ứng dụng chạy ổn định và mượt mà hơn so với các dịch vụ hosting chia sẻ.
- VPS cung cấp hiệu năng và tính năng tương tự như máy chủ vật lý với chi phí thấp hơn nhiều. Điều này làm cho VPS trở thành lựa chọn lý tưởng cho các doanh nghiệp nhỏ và các dự án khởi nghiệp.
- VPS cho phép dễ dàng nâng cấp tài nguyên khi nhu cầu tăng lên mà không cần di chuyển sang một máy chủ mới. Điều này giúp doanh nghiệp linh hoạt trong việc mở rộng hệ thống.
- VPS cung cấp các giải pháp bảo mật mạnh mẽ như tường lửa, VPN và mã hóa dữ liệu, giúp bảo vệ dữ liệu và ứng dụng khỏi các mối đe dọa.
- Hơn hết dịch vụ hỗ trợ kỹ thuật chuyên nghiệp, giúp giải quyết nhanh chóng các vấn đề kỹ thuật và đảm bảo hệ thống hoạt động ổn định.

3.4. Thiết kế hệ thống

3.4.1. Mô hình cơ sở dữ liệu

3.4.1.1 Đặc tả dữ liệu

TÊN THỰC THỂ	CÁC THUỘC TÍNH
CartItem	cartItemId, productId, imageName, quantity, colorName, sizeName, totalPrice, totalSalePrice, shoppingCartId, orderId
Category	categoryId, categoryName
Color	colorId, colorName
User	userId, username, password, phoneNumber, address, email, role, avatar, gender, createdAt
Image	imageId, imageName, colorId, productId
Product	productId, productName, description, costPrice, salePrice, categoryId, createAt, type, discount
Order	orderId, orderDate, orderStatus, totalPrice, address, userId, paymentMethod, transportPrice, deliveryDate
Size	sizeId, sizeName
ShoppingCart	shoppingCartId, totalPrice, grandTotal, totalSalePrice, totalItems, userId
Preference	preferenceId, avatar, category, color, gender, userId

3.4.1.2. Mô hình hóa thông tin ở mức tổ chức, danh mục ngữ vựng cho các thuộc tính

Bảng 3.4.1.2.1. CartItem

SNO	NAME	DATA TYPE	CONSTRAINTS	MEANS
1	cartItemId	bigint	Primary key, Not Null	Mã mục trong giỏ hàng
2	productId	bigint	Foreign Key, Not Null	Mã sản phẩm
3	imageName	varchar(255)	Not Null	Ảnh sản phẩm
4	quantity	int(10)	Not Null	Số lượng sản phẩm
5	colorName	varchar(15)	Not Null	Màu sản phẩm
6	sizeName	varchar(15)	Not Null	Size sản phẩm
7	totalPrice	decimal(10,2)	Not Null	Tổng giá
8	totalSalePrice	decimal(10,2)	Not Null	Tổng giá đã giảm
9	shoppingCartId	bigint	Foreign Key, Not Null	Mã giỏ hàng
10	orderId	bigint	Foreign Key, Not Null	Mã đơn đặt hàng

Bảng 3.4.1.2.2. Category

SNO	NAME	DATA TYPE	CONSTRAINTS	MEANS
1	categoryId	bigint	Primary key, Not Null	Mã danh mục
2	categoryName	varchar(15)	Not Null	Tên danh mục

Bảng 3.4.1.2.3. Color

SNO	NAME	DATA TYPE	CONSTRAINTS	MEANS
1	colorId	bigint	Primary key, Not Null	Mã màu
2	colorName	varchar(15)	Not Null	Tên màu

Bảng 3.4.1.2.4. User

SNO	NAME	DATA TYPE	CONSTRAINTS	MEANS
1	userId	bigint	Primary key, Not Null	Mã thành viên
2	username	varchar(15)	Not Null	Tên thành viên
3	password	varchar(15)	Not Null	Mật khẩu
4	phoneNumber	varchar(15)	Not Null	Số điện thoại
5	address	varchar(255)	Not Null	Địa chỉ
6	email	varchar(255)	Not Null	Thư điện tử
7	role	varchar(15)	Not Null	Vai trò
8	avatar	varchar(15)	Not Null	Ảnh đại diện
9	gender	varchar(15)	Not Null	Giới tính

Bảng 3.4.1.2.5. Image

SNO	NAME	DATA TYPE	CONSTRAINTS	MEANS
1	imageId	bigint	Primary key, Not Null	Mã ảnh
2	imageName	varchar(15)	Not Null	Tên ảnh
3	colorId	bigint	Foreign Key, Not Null	Mã màu
4	productId	bigint	Foreign Key, Not Null	Mã sản phẩm

Bảng 3.4.1.2.6. Product

SNO	NAME	DATA TYPE	CONSTRAINTS	MEANS
1	productId	bigint	Primary key, Not Null	Mã sản phẩm
2	productName	varchar(255)	Not Null	Tên sản phẩm
3	description	varchar(255)	Not Null	Mô tả
4	createdAt	date	Not Null	Ngày phát hành
5	costPrice	decimal(10,2)	Not Null	Giá gốc
6	salePrice	decimal(10,2)	Not Null	Giá đã giảm

7	type	varchar(15)	Not Null	Loại sản phẩm
8	discount	int	Not Null	Phần trăm giảm giá
9	categoryId	bigint	Foreign Key, Not Null	Mã danh mục

Bảng 3.4.1.2.7. Order

SNO	NAME	DATA TYPE	CONSTRAINTS	MEANS
1	orderId	bigint	Primary key, Not Null	Mã đơn hàng
2	orderDate	date	Not Null	Ngày đặt hàng
3	orderStatus	varchar(255)	Not Null	Trạng thái đơn hàng
4	totalPrice	decimal(10,2)	Not Null	Tổng giá đơn hàng
5	transportPrice	decimal(10,2)	Not Null	Giá vận chuyển
6	paymentMethod	varchar(255)	Not Null	Phương thức thanh toán
7	address	varchar(255)	Not Null	Địa chỉ
8	userId	bigint	Foreign Key, Not Null	Mã khách hàng

Bảng 3.4.1.2.8. Size

SNO	NAME	DATA TYPE	CONSTRAINTS	MEANS
1	sizeId	bigint	Primary key, Not Null	Mã kích cỡ
2	sizeName	varchar(5)	Not Null	Tên kích cỡ

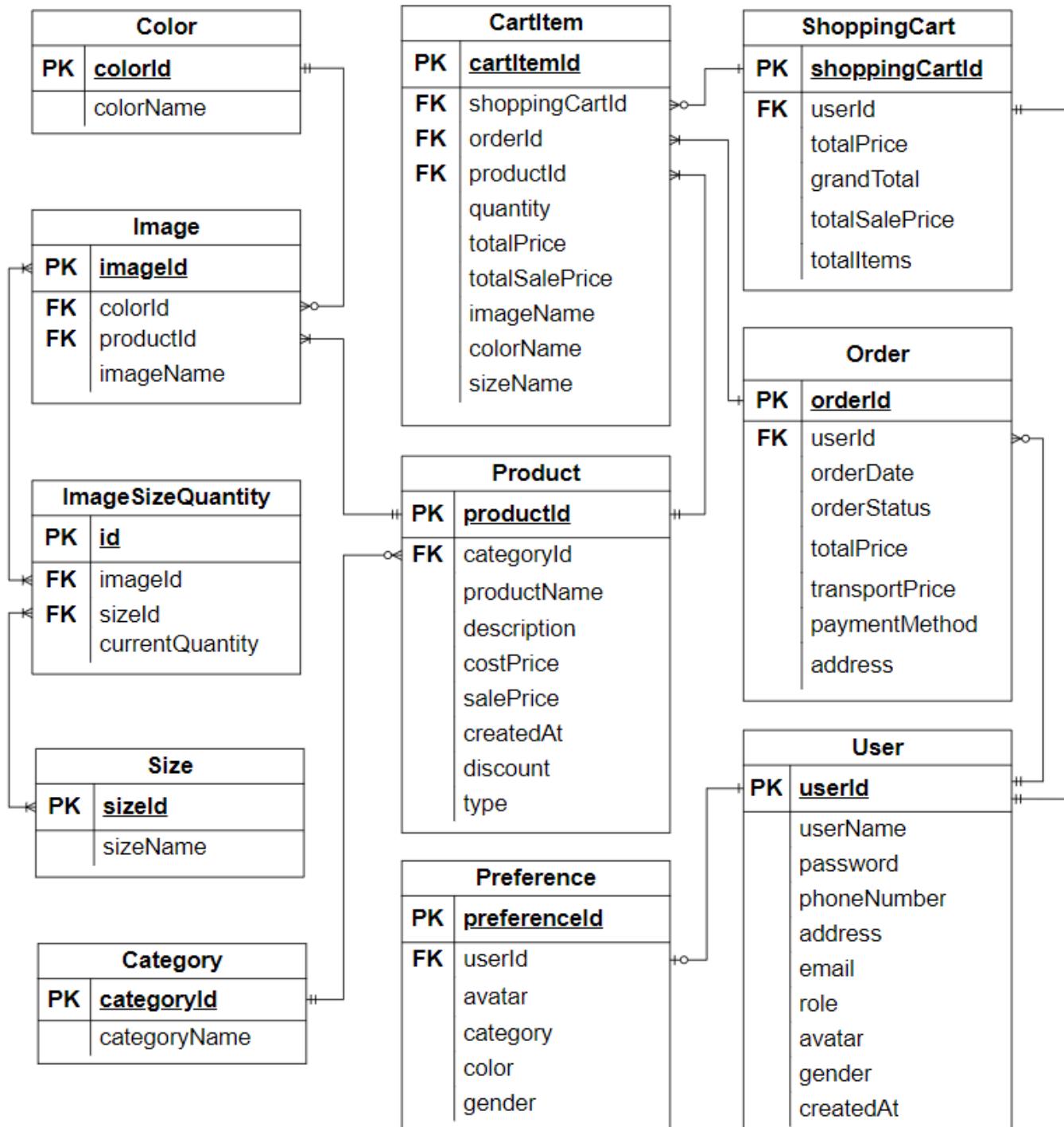
Bảng 3.4.1.2.9. ShoppingCart

SNO	NAME	DATA TYPE	CONSTRAINTS	MEANS
1	shoppingCartId	bigint	Primary key, Not Null	Mã giỏ hàng
2	totalPrice	decimal(10,2)	Not Null	Giá
3	grandTotal	decimal(10,2)	Not Null	Tổng giá
4	totalSalePrice	decimal(10,2)	Not Null	Tổng giá giảm
5	totalItems	int	Not Null	Tổng số sản phẩm
6	userId	bigint	Foreign Key, Not Null	Mã khách hàng

Bảng 3.4.1.2.10. Preference

SNO	NAME	DATA TYPE	CONSTRAINTS	MEANS
1	preferenceId	bigint	Primary key, Not Null	Mã khảo sát sở thích
2	avatar	varchar(5)	Not Null	Ảnh đại diện
3	category	varchar(5)	Not Null	Danh mục
4	color	varchar(5)	Not Null	Màu sắc
5	gender	varchar(5)	Not Null	Giới tính
6	userId	bigint	Foreign Key, Not Null	Mã khách hàng

3.4.1.3. Mô hình quan hệ thực thể (ERD)



Hình 3.4.1.3.1. Sơ đồ ERD thể hiện các mối quan hệ trong hệ thống

Đặc tả ràng buộc toàn vẹn dữ liệu:

- Giữa **User** và **Order**: một khách hàng có thể có nhiều đơn đặt hàng hoặc không đặt bất kì đơn hàng nào. Một đơn đặt hàng chỉ thuộc về duy nhất một khách hàng.

- Giữa **ShoppingCart** và **User**: một khách hàng buộc phải có một giỏ hàng và chỉ được một giỏ hàng duy nhất. Một giỏ hàng chỉ thuộc về duy nhất một khách hàng.
- Giữa **CartItem** và **Order**: một đơn hàng buộc phải có một hoặc nhiều sản phẩm cụ thể. Một hoặc nhiều sản phẩm cụ thể sẽ thuộc trong một đơn đặt hàng.
- Giữa **ShoppingCart** và **CartItem**: một giỏ hàng có thể không có sản phẩm cụ thể nào hoặc có nhiều sản phẩm. Một hoặc nhiều sản phẩm cụ thể sẽ thuộc một giỏ hàng.
- Giữa **CartItem** và **Product**: một sản phẩm có thể xuất hiện trong nhiều mục giỏ hàng. Mỗi mục giỏ hàng chỉ chứa một sản phẩm.
- Giữa **Product** và **Image**: một sản phẩm buộc phải có một ảnh hoặc nhiều ảnh. Một hoặc nhiều ảnh đó chỉ thuộc duy nhất một sản phẩm.
- Giữa **Category** và **Product**: một danh mục thuộc nhiều sản phẩm. Một sản phẩm có duy nhất một danh mục cụ thể.
- Giữa **Color** và **Image**: một ảnh buộc phải có một màu và chỉ được một màu duy nhất. Một màu có thể thuộc nhiều ảnh.
- **ImageSizeQuantity**: thực thể thể hiện mối quan hệ nhiều nhiều giữa **Image** và **Size**.
- Giữa **User** và **Preference**: Một khách hàng có thể có sở thích hoặc không. Một sở thích chỉ thuộc một khách hàng duy nhất.

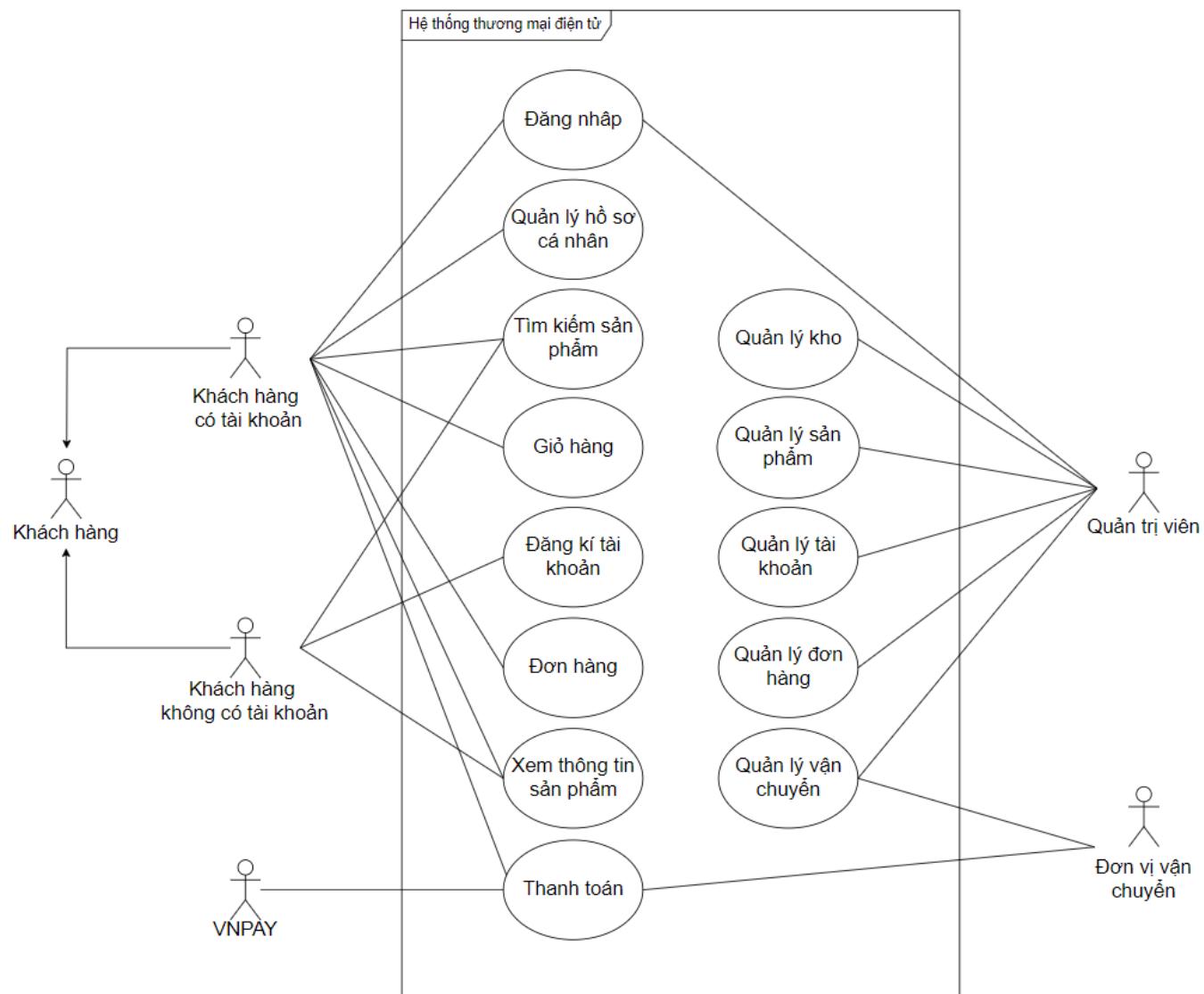
3.4.2. Sơ đồ Use Case (Use Case diagram)

Tổng quan:

Sơ đồ Use Case là công cụ quan trọng trong phân tích và thiết kế hệ thống, mô tả sự tương tác giữa các actor và hệ thống. Sơ đồ này cung cấp một cái nhìn tổng thể về những gì mà hệ thống phải thực hiện và ai sẽ sử dụng nó, giúp hiểu rõ hơn về các yêu cầu và chức năng của hệ thống.

Trong đó:

- **Actor**: đại diện cho người sử dụng hoặc một đối tượng nào đó bên ngoài, tương tác với hệ thống mà chúng ta đang xem xét. Actor có thể là con người, hệ thống khác hoặc bất kỳ thực thể nào tương tác với hệ thống của chúng ta.
- **Use Case «include»**: một Use Case có thể chứa («include») chức năng của một Use Case khác như một phần xử lý của nó.
- **Use Case «extend»**: một Use Case mở rộng có thể được mở rộng («extend») hành vi từ một Use Case khác.
- **Use Case Generalizations**: một quan hệ generalization có giữa một Use Case cụ thể hơn (specifilized) đến với một Use Case tổng quát hơn (generalized).



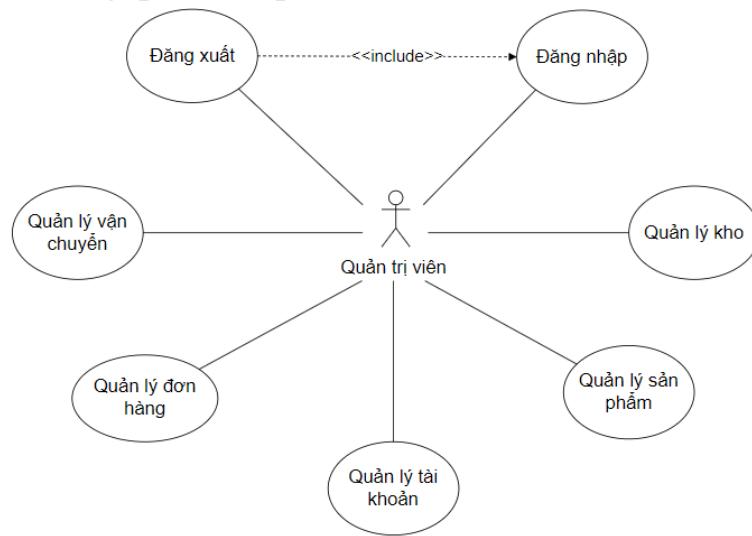
Hình 3.4.2.1. Sơ đồ Use Case tổng quát hệ thống

Bảng 3.4.2.1. Đặc tả Use Case tổng quát hệ thống

Tác nhân	Use Case	Mô tả
VNPay	Thanh toán	
Đơn vị vận chuyển	Quản lý vận chuyển	
	Thanh toán	
Quản trị viên	Đăng nhập	Mô tả sẽ được mô tả chi tiết ở các Use Case bên dưới
	Quản lý kho	
	Quản lý sản phẩm	
	Quản lý tài khoản	
	Quản lý đơn hàng	
	Quản lý vận chuyển	

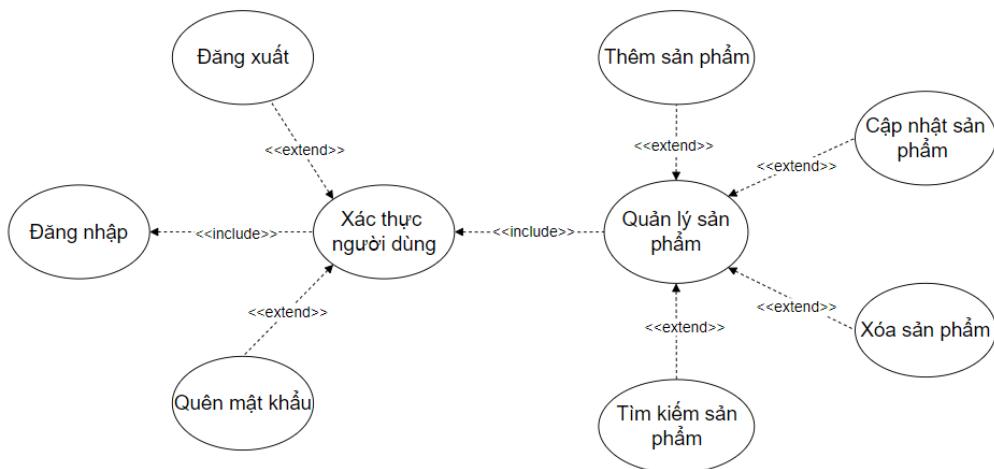
Khách hàng có tài khoản	Đăng nhập
	Quản lý hồ sơ cá nhân
	Tìm kiếm sản phẩm
	Giỏ hàng
	Thanh toán
	Đơn hàng
	Xem thông tin sản phẩm
Khách hàng không có tài khoản	Đăng ký tài khoản
	Tìm kiếm sản phẩm
	Xem thông tin sản phẩm

✚ Sơ đồ Use Case tổng quát của quản trị viên



Hình 3.4.2.2. Sơ đồ Use Case tổng quát của quản trị viên

✚ Quản lý sản phẩm

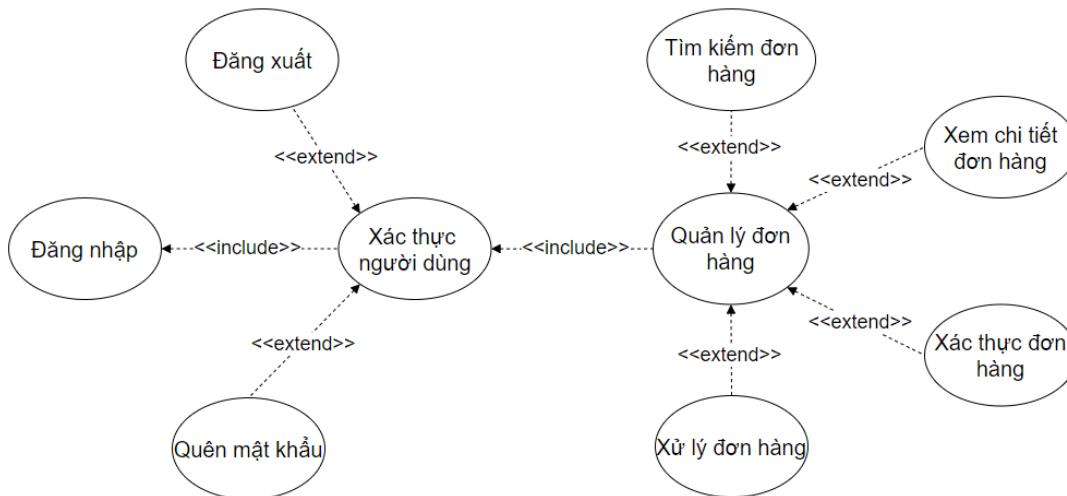


Hình 3.4.2.3. Sơ đồ Use Case Quản lý sản phẩm của quản trị viên

Bảng 3.4.2.2. Đặc tả Use Case Quản lý sản phẩm của quản trị viên

Use Case	Đặc tả
Quản lý sản phẩm	Quản trị viên có quyền thêm mới sản phẩm vào hệ thống. Nhập thông tin về sản phẩm như tên, mô tả, giá, số lượng, v.v. Hệ thống xác nhận và lưu trữ thông tin sản phẩm trong cơ sở dữ liệu. Ngoài ra, quản trị viên có khả năng tìm kiếm sản phẩm dựa trên các tiêu chí như tên, danh mục, giá, v.v. Kết quả tìm kiếm hiển thị danh sách sản phẩm phù hợp với tiêu chí tìm kiếm. Và quản trị viên có thể xóa sản phẩm khỏi hệ thống nếu sản phẩm không còn cần thiết hoặc không khả dụng nữa. Hệ thống xác nhận và xóa thông tin sản phẩm khỏi cơ sở dữ liệu. Quản trị viên có quyền chỉnh sửa thông tin của sản phẩm như tên, mô tả, giá, số lượng, v.v. Nhập thông tin mới, hệ thống cập nhật thông tin sản phẩm trong cơ sở dữ liệu.

♦ Quản lý đơn hàng



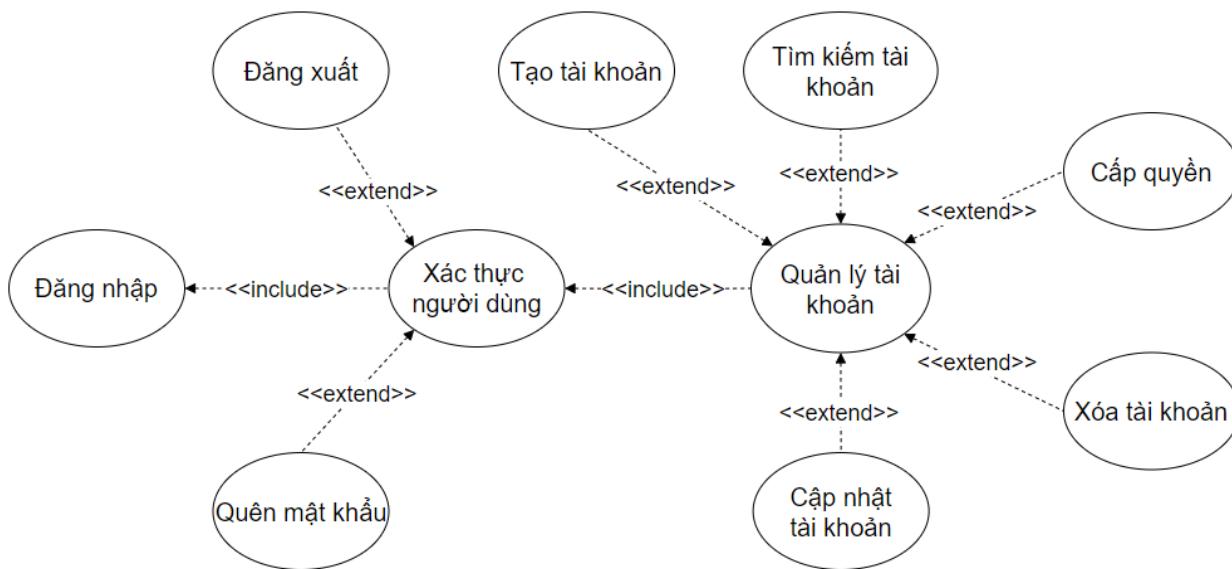
Hình 3.4.2.4. Sơ đồ Use Case Quản lý đơn hàng của quản trị viên

Bảng 3.4.2.3. Đặc tả Use Case Quản lý đơn hàng của quản trị viên

Use Case	Đặc tả
Quản lý đơn hàng	Quản trị viên có chức năng tìm kiếm đơn hàng cho phép tìm kiếm các đơn hàng trong hệ thống theo các tiêu chí như mã đơn hàng, tên khách hàng, địa chỉ giao hàng, hoặc trạng thái đơn hàng. Người quản lý có thể sử dụng các bộ lọc và tùy chọn tìm kiếm để thu hẹp kết quả tìm kiếm và dễ dàng tìm ra đơn hàng cần thiết. Chức năng xử lý đơn hàng cho phép người quản lý xem và chỉnh sửa thông tin của đơn hàng. Người quản lý có thể thay đổi trạng thái của đơn hàng, như chuyển từ "Chờ xử lý" sang "Đang vận chuyển" hoặc "Đã giao hàng". Họ cũng có thể cập nhật thông tin liên quan đến

	sản phẩm, địa chỉ giao hàng, hoặc bất kỳ yêu cầu đặc biệt nào từ khách hàng. Đồng thời xác thực và xem chi tiết đảm bảo rằng chỉ những người được ủy quyền mới có thể xem và chỉnh sửa thông tin của đơn hàng. Người quản lý cần xác thực danh tính của mình bằng cách đăng nhập vào hệ thống với tài khoản và mật khẩu của mình trước khi được phép truy cập vào chi tiết của đơn hàng. Sau khi xác thực, họ có thể xem chi tiết đầy đủ của đơn hàng, bao gồm thông tin về sản phẩm, địa chỉ giao hàng, thông tin liên hệ của khách hàng và bất kỳ ghi chú nào.
--	--

❖ Quản lý tài khoản



Hình 3.4.2.5. Sơ đồ Use Case Quản lý tài khoản của quản trị viên

Bảng 3.4.2.Error! Bookmark not defined. Đặc tả Use Case Quản lý tài khoản của quản trị viên

Use Case	Đặc tả
Quản lý tài khoản	Quản trị viên có khả năng tìm kiếm tài khoản người dùng hoặc quản trị viên dựa trên các tiêu chí như tên, email, v.v. Kết quả tìm kiếm hiển thị danh sách tài khoản phù hợp với tiêu chí tìm kiếm. Bên cạnh đó, quản trị viên có quyền cấp quyền quản trị cho người dùng, cho phép họ có quyền truy cập và thực hiện các chức năng quản trị trong hệ thống. Mặt khác, quản trị viên có thể xóa tài khoản người dùng hoặc quản trị viên khỏi hệ thống nếu không cần thiết hoặc theo yêu cầu.

❖ Quản lý vận chuyển

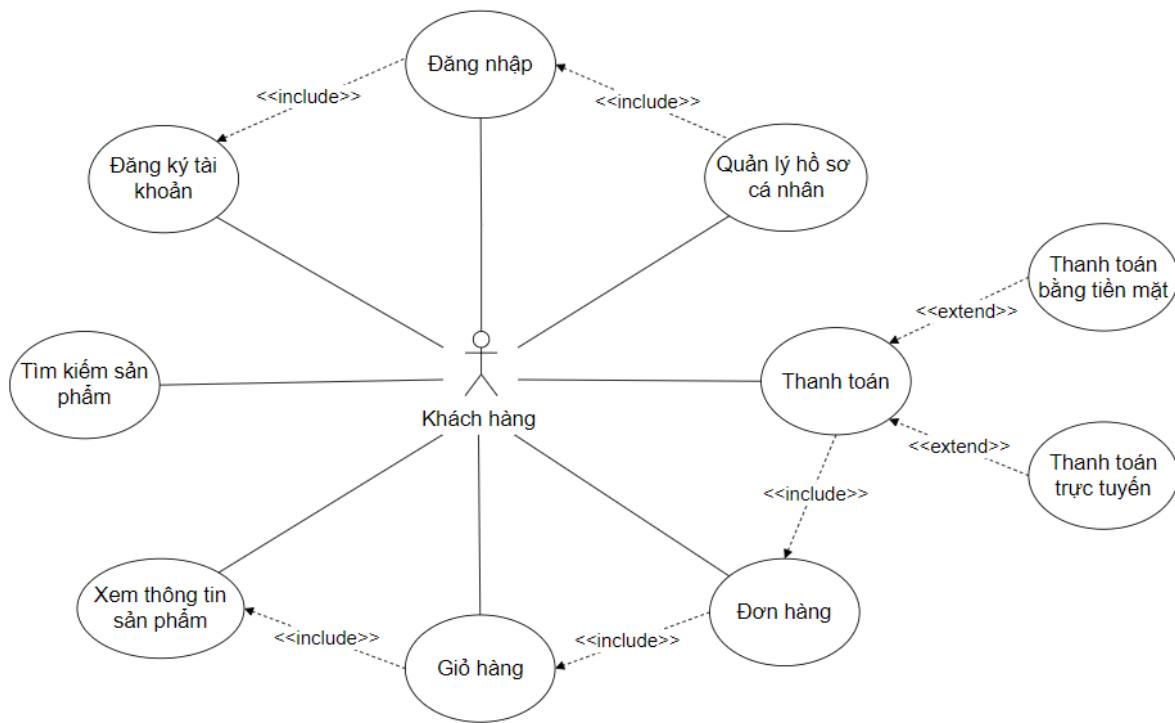


Hình 3.4.2.6. Sơ đồ Use Case Quản lý vận chuyển của quản trị viên

Bảng 3.4.2. Error! Switch argument not specified. Đặc tả Use Case Quản lý vận chuyển của quản trị viên

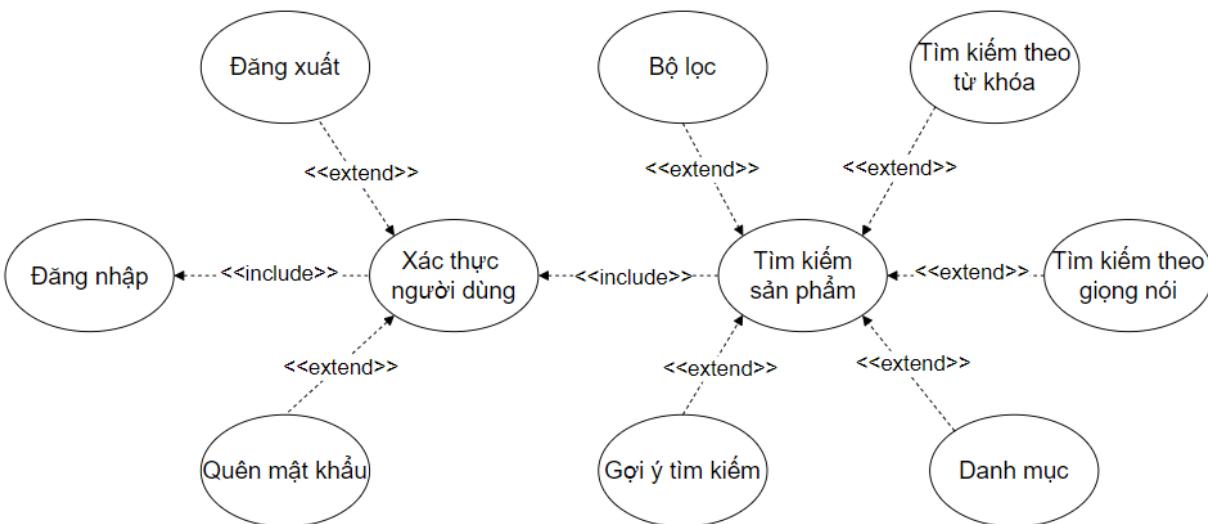
Use Case	Đặc tả
Quản lý vận chuyển	Quản trị viên có khả năng quản lý quá trình đóng gói sản phẩm trước khi vận chuyển. Quản trị viên đảm bảo sản phẩm được đóng gói đúng cách để đảm bảo an toàn và chất lượng trong quá trình vận chuyển. Bên cạnh đó, quản trị viên có khả năng quản lý quá trình vận chuyển hàng hóa từ kho đến địa chỉ giao hàng. Sau đó, xác định phương thức vận chuyển, tạo đơn vận chuyển và cung cấp thông tin cho đơn vị vận chuyển. Quản trị viên có thể chọn và quản lý các phương thức vận chuyển khác nhau, ví dụ như giao hàng nhanh, giao hàng tiết kiệm, v.v. Tiếp theo, cung cấp thông tin về phương thức vận chuyển cho đơn vị vận chuyển và theo dõi quá trình vận chuyển. Quản trị viên có khả năng theo dõi và báo cáo về quá trình quản lý đơn hàng và vận chuyển. Hơn hết, quản trị viên có thể xem thông tin về số lượng đơn hàng, trạng thái, thời gian vận chuyển, v.v. và tạo báo cáo tổng quan về hoạt động của hệ thống. Quản trị viên có trách nhiệm bảo vệ thông tin đơn hàng và dữ liệu khách hàng. Thực hiện các biện pháp bảo mật như mã hóa dữ liệu, kiểm tra danh tính, v.v. Mặt khác, quản trị viên có khả năng xác thực danh tính khách hàng và đơn hàng trước khi xử lý giao dịch hoặc cung cấp thông tin.

+Sơ đồ Use case tổng quát của khách hàng



Hình 3.4.2.7. Sơ đồ Use Case Tổng quát của khách hàng

✚ Tìm kiếm sản phẩm



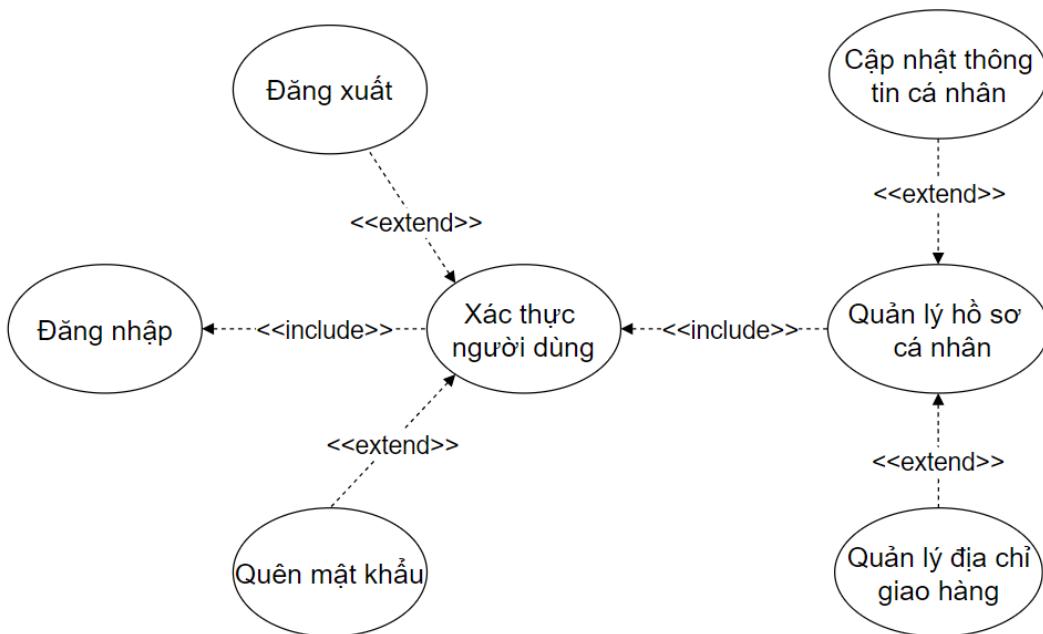
Hình 3.4.2.8. Sơ đồ Use Case Tìm kiếm sản phẩm của khách hàng

Bảng 3.4.2. Error! Switch argument not specified. Đặc tả Use Case Tìm kiếm sản phẩm của khách hàng

Use Case	Đặc tả
Tìm kiếm sản phẩm	Người dùng nhập từ khóa vào ô tìm kiếm. Hệ thống xử lý và tìm kiếm trong cơ sở dữ liệu các sản phẩm có liên quan đến từ khóa. Áp dụng bộ lọc để hạn chế kết quả

tìm kiếm. Hệ thống cung cấp các tiêu chí bộ lọc như giá, thương hiệu, kích thước, màu sắc, đánh giá khách hàng, v.v. Khi người dùng nhập từ khóa vào ô tìm kiếm, hệ thống gợi ý danh sách các từ khóa liên quan. Các gợi ý tìm kiếm dựa trên lịch sử tìm kiếm trước đó hoặc các từ khóa phổ biến. Người dùng có thể chọn một trong các gợi ý hoặc tiếp tục nhập từ khóa khác để tìm kiếm chính xác hơn. Hệ thống cung cấp danh sách các danh mục sản phẩm. Người dùng có thể chọn từ danh sách danh mục để xem các sản phẩm thuộc danh mục đó. Từ đó, danh sách các sản phẩm liên quan được hiển thị cho người dùng.

Quản lý hồ sơ cá nhân



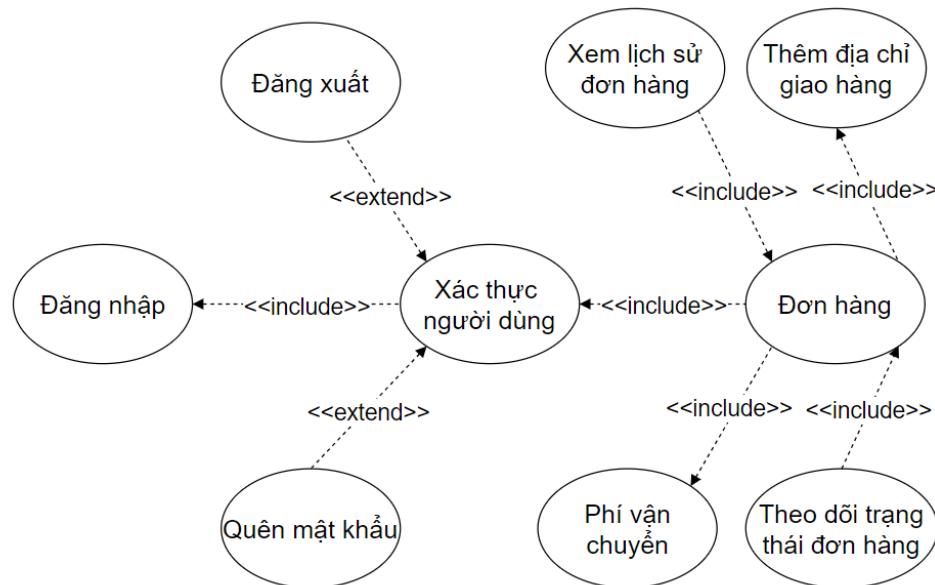
Hình 3.4.2.9. Sơ đồ Use Case Quản lý hồ sơ cá nhân của khách hàng

Bảng 3.4.2. Error! Switch argument not specified. Đặc tả Use Case Quản lý hồ sơ cá nhân của khách hàng

Use Case	Đặc tả
Quản lý hồ sơ cá nhân	Người dùng truy cập vào trang quản lý hồ sơ cá nhân. Hệ thống hiển thị thông tin cá nhân hiện tại của người dùng, bao gồm tên, địa chỉ, số điện thoại, email, mật khẩu v.v. Người dùng có thể chỉnh sửa thông tin cá nhân và lưu lại các thay đổi. Bên cạnh đó, người dùng truy cập vào mục quản lý địa chỉ giao hàng. Hệ thống hiển thị danh sách các địa chỉ giao hàng hiện có của

	người dùng. Người dùng có thể thêm địa chỉ mới, chỉnh sửa hoặc xóa các địa chỉ hiện có.
--	---

❖ Đơn hàng

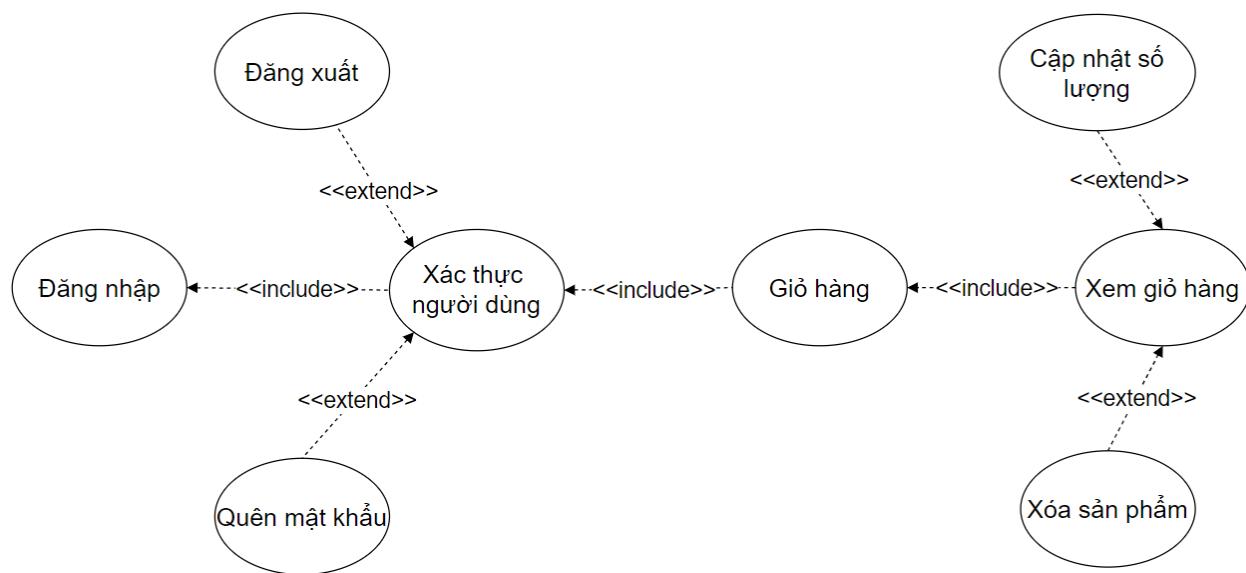


Hình 3.4.2.10. Sơ đồ Use Case Đơn hàng của khách hàng

Bảng 3.4.2. Error! Switch argument not specified. Đặc tả Use Case Đơn hàng của khách hàng

Use Case	Đặc tả
Đơn hàng	Người dùng truy cập vào trang lịch sử đơn hàng. Hệ thống hiển thị danh sách các đơn hàng đã được người dùng đặt trong quá khứ và hiện tại. Mỗi đơn hàng được hiển thị với thông tin như số đơn hàng, ngày đặt hàng, tổng giá trị đơn hàng, trạng thái, vận chuyển, v.v. Người dùng có thể chọn một đơn hàng từ danh sách lịch sử đơn hàng để xem chi tiết và theo dõi trạng thái vận chuyển. Hệ thống hiển thị trạng thái hiện tại của đơn hàng, bao gồm thông tin vận chuyển, dự kiến giao hàng, v.v. Sau khi nhận được đơn hàng, người dùng có thể truy cập vào trang đánh giá và nhận xét về sản phẩm. Hệ thống hiển thị danh sách các sản phẩm trong đơn hàng và yêu cầu người dùng đánh giá và viết nhận xét về từng sản phẩm. Người dùng có thể đánh giá bằng sao, viết nhận xét về trải nghiệm sử dụng sản phẩm và gửi đánh giá đi.

❖ Giỏ hàng



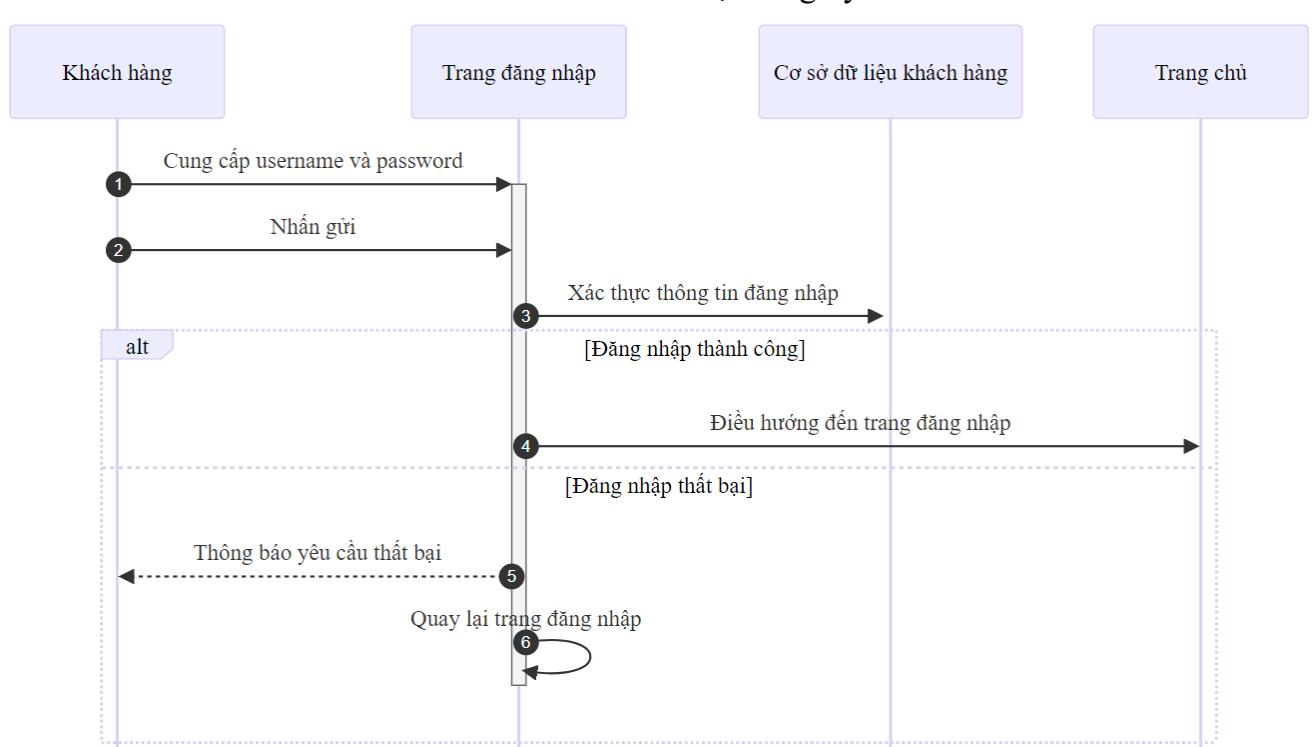
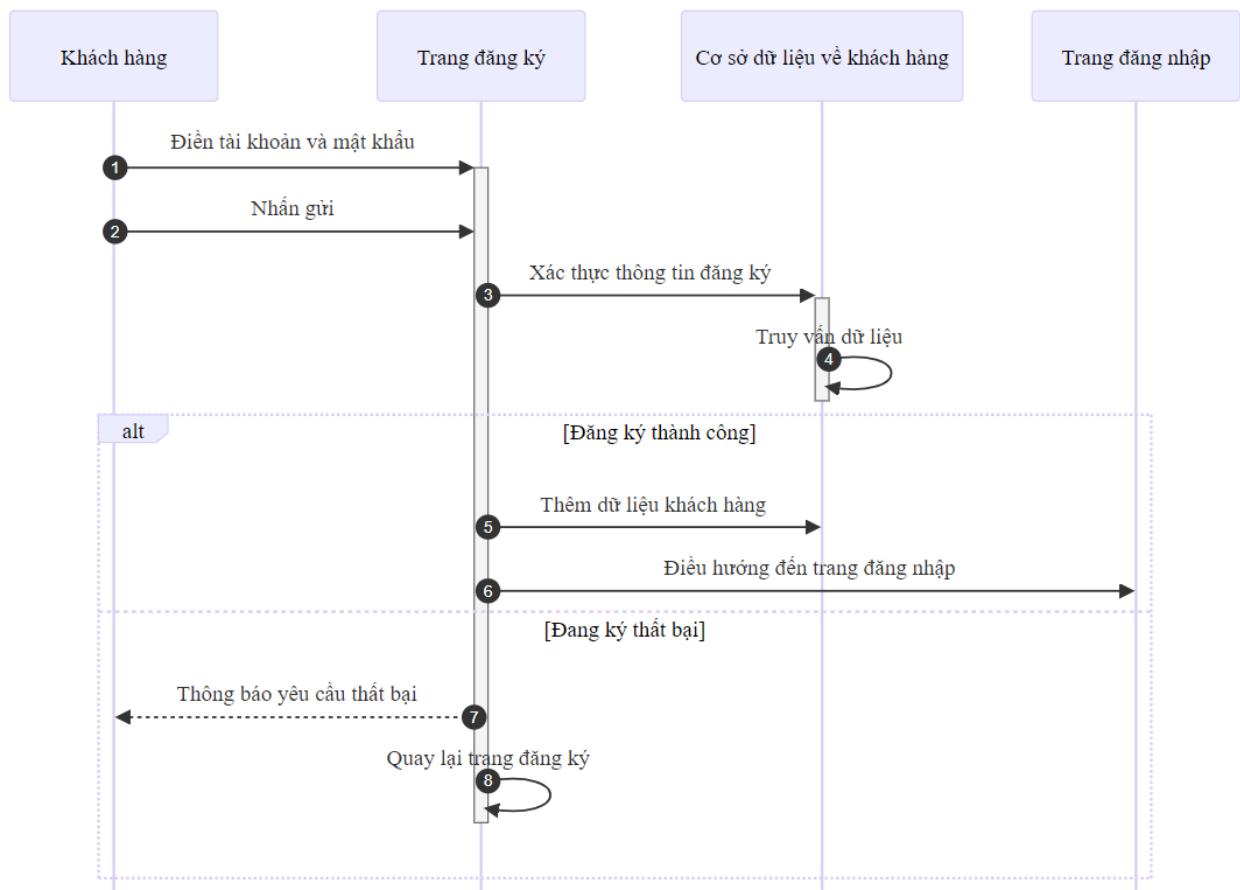
Hình 3.4.2.11. Sơ đồ Use Case Giỏ hàng của khách hàng

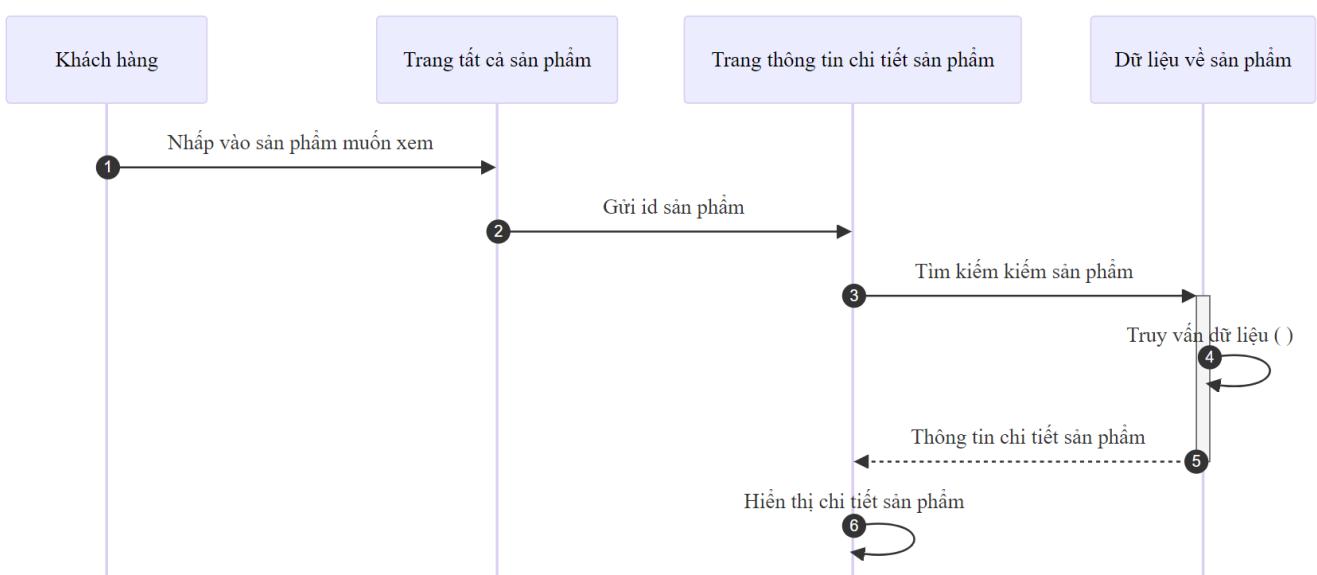
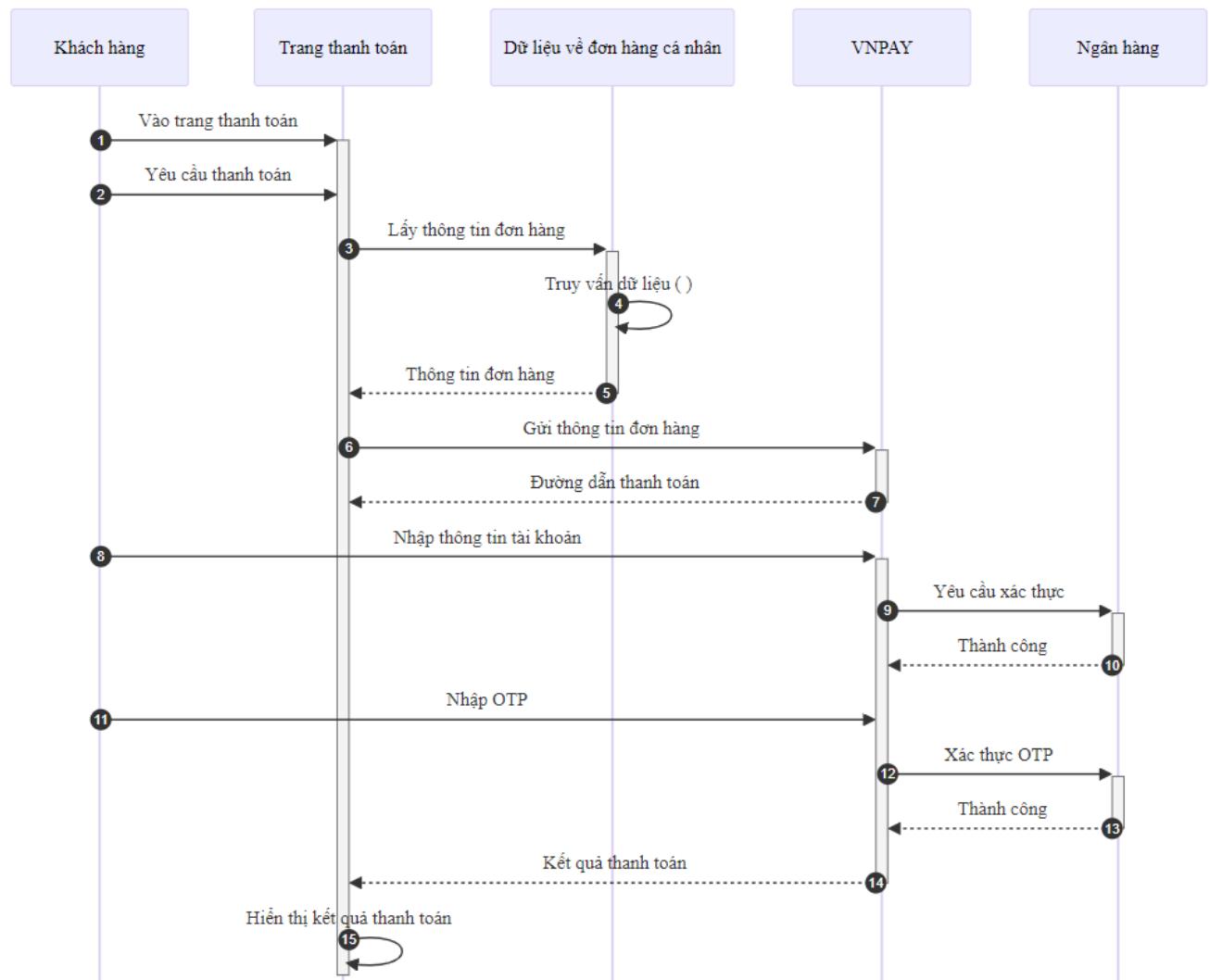
Bảng 3.4.2 Error! No text of specified style in document. Đặc tả Use Case Giỏ hàng của khách hàng

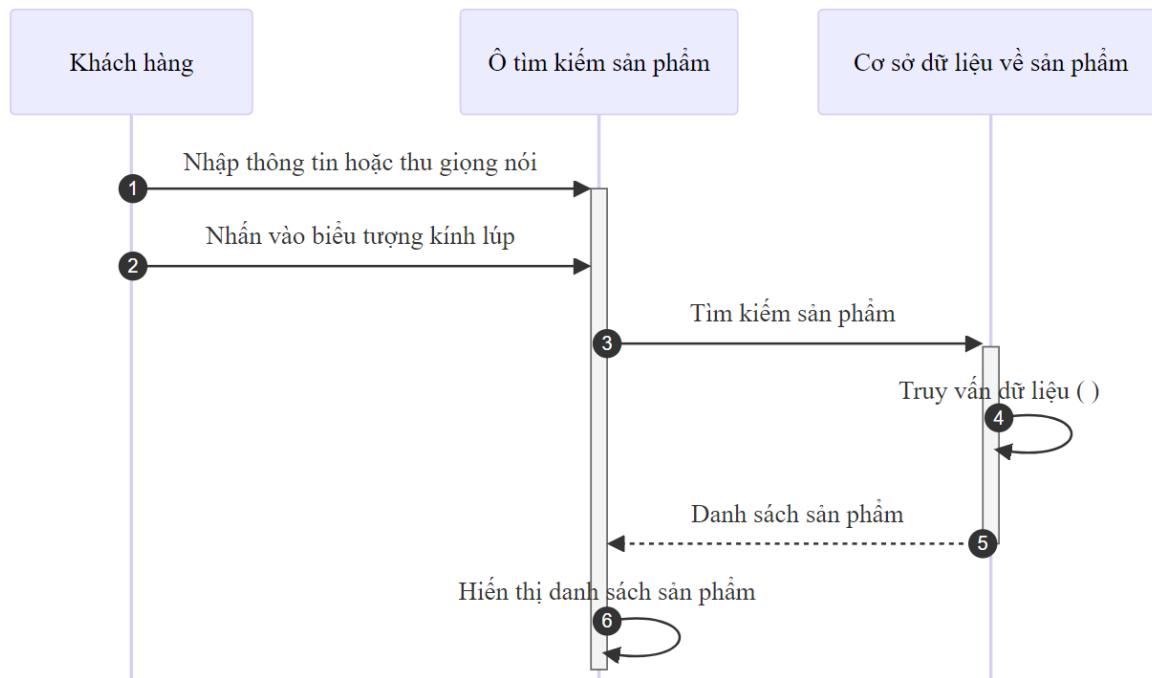
Use Case	Đặc tả
Giỏ hàng	<p>Người dùng chọn một sản phẩm từ trang thông tin sản phẩm hoặc danh sách sản phẩm. Người dùng chọn tùy chọn "Thêm vào giỏ hàng" hoặc biểu tượng giỏ hàng. Hệ thống thực hiện thêm sản phẩm vào giỏ hàng và chuyển hướng đến trang giỏ hàng. Trong giỏ hàng thể hiện các mặt hàng mà khách hàng đã đặt cũng như là tổng số lượng, giá tạm tính, giá giảm,... Người dùng có thể cập nhật số lượng sản phẩm bằng cách thay đổi giá trị trong ô số lượng hoặc sử dụng các nút tăng/giảm số lượng và nhấn gửi. Hệ thống tự động cập nhật giá tiền và tổng giá trị đơn hàng khi số lượng sản phẩm được thay đổi. Người dùng có thể chọn tùy chọn "Xóa" hoặc biểu tượng thùng rác để xóa sản phẩm khỏi giỏ hàng. Hệ thống thực hiện xóa sản phẩm và cập nhật tổng giá trị đơn hàng. Trong trang giỏ hàng, người dùng có thể xem chi tiết sản phẩm bằng cách nhấp vào tên hoặc hình ảnh sản phẩm. Nếu khách hàng muốn tiếp tục lựa chọn thêm các mặt hàng khác có thể nhấn nút back hoặc thêm sản phẩm.</p>

3.4.3. Sơ đồ tuần tự (Sequence diagram)

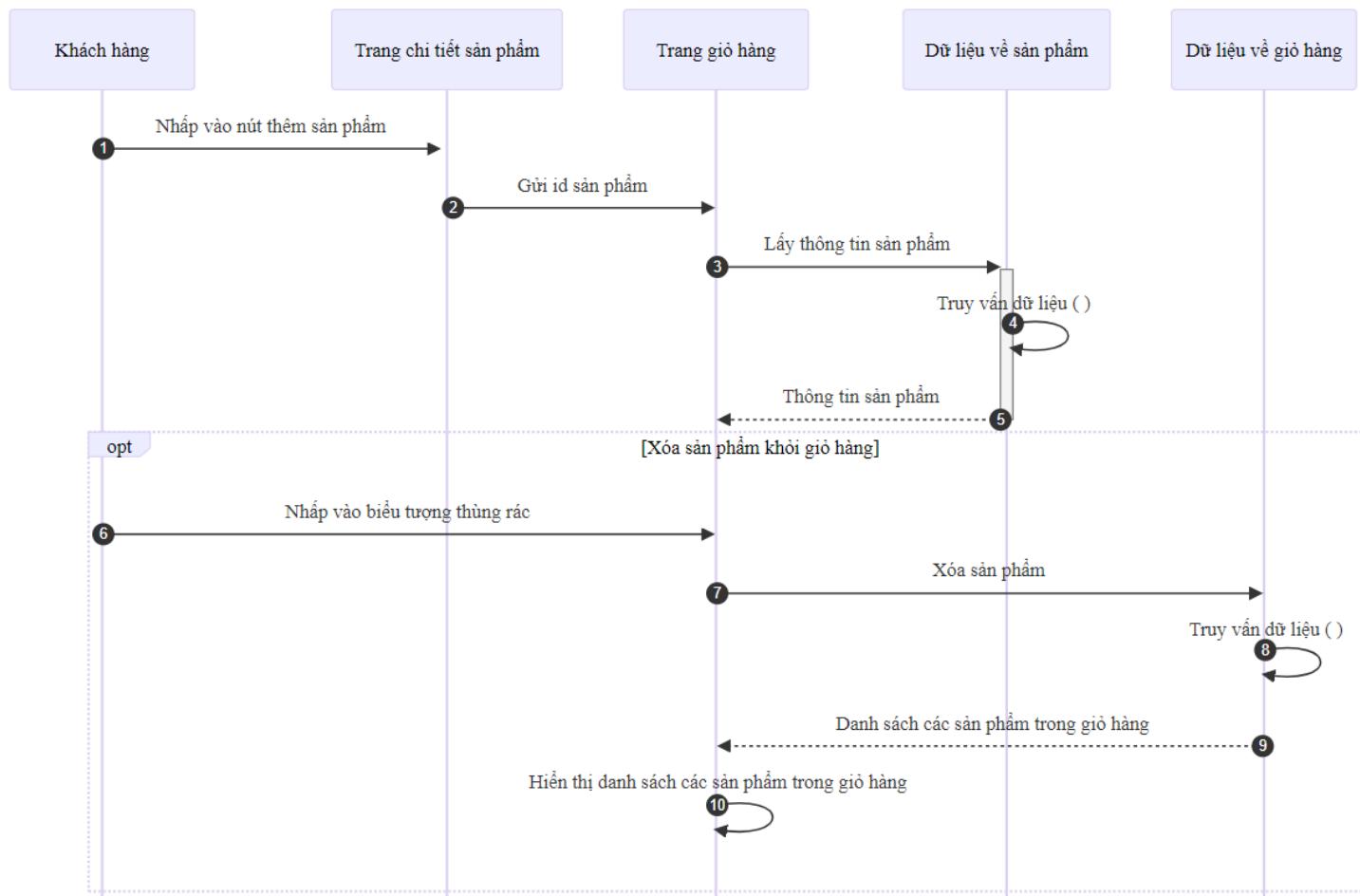
Khách hàng:



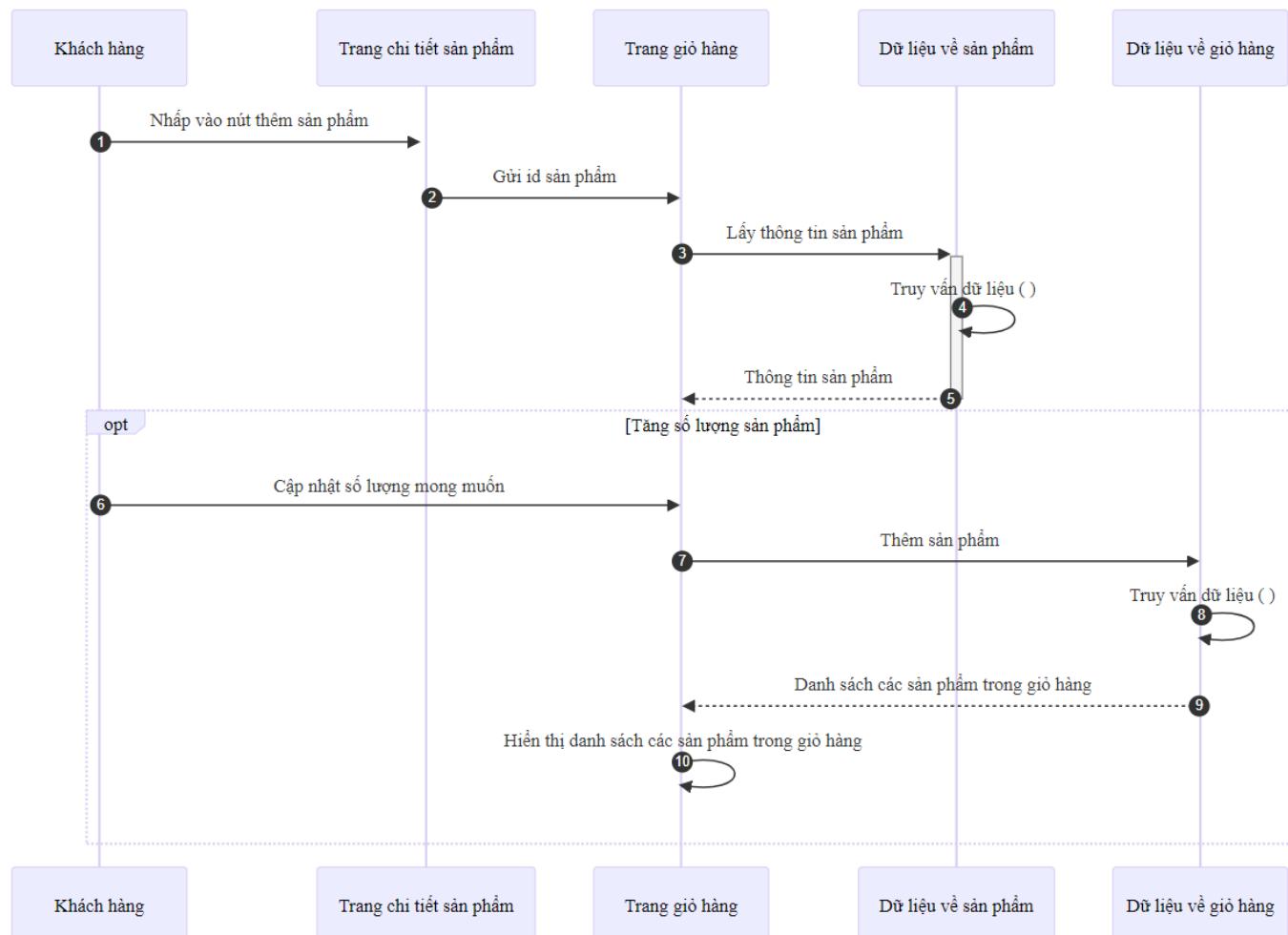




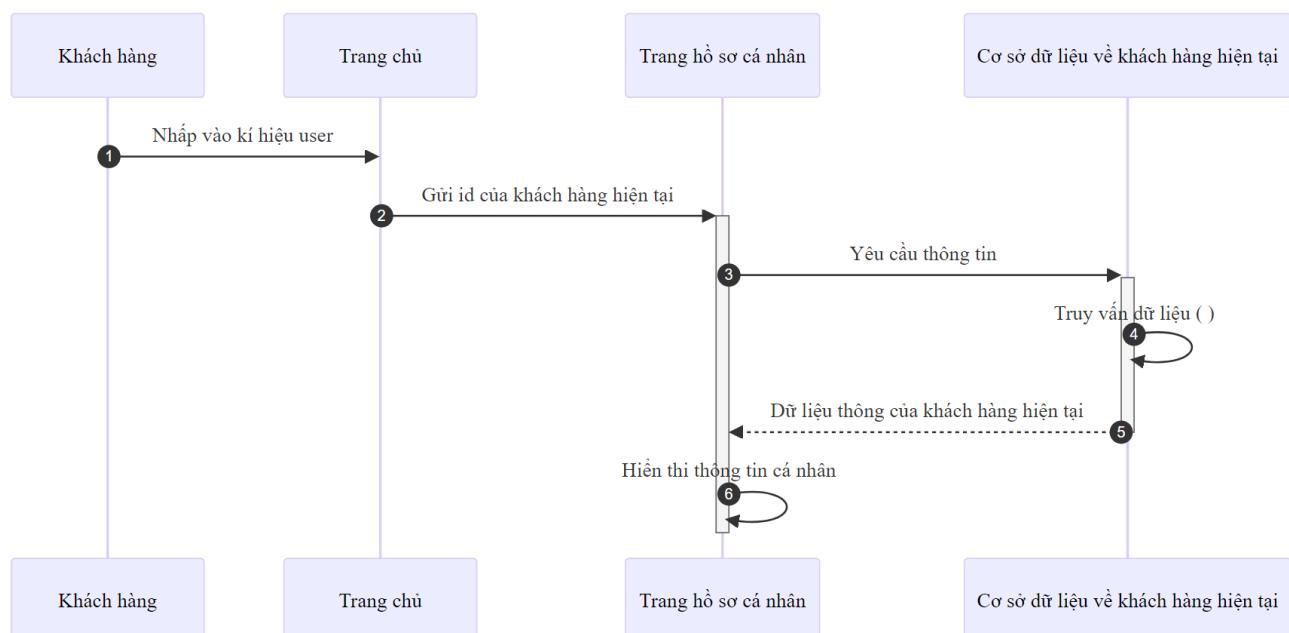
Hình 3.4.3.5. Biểu đồ tuần tự Tìm kiếm sản phẩm



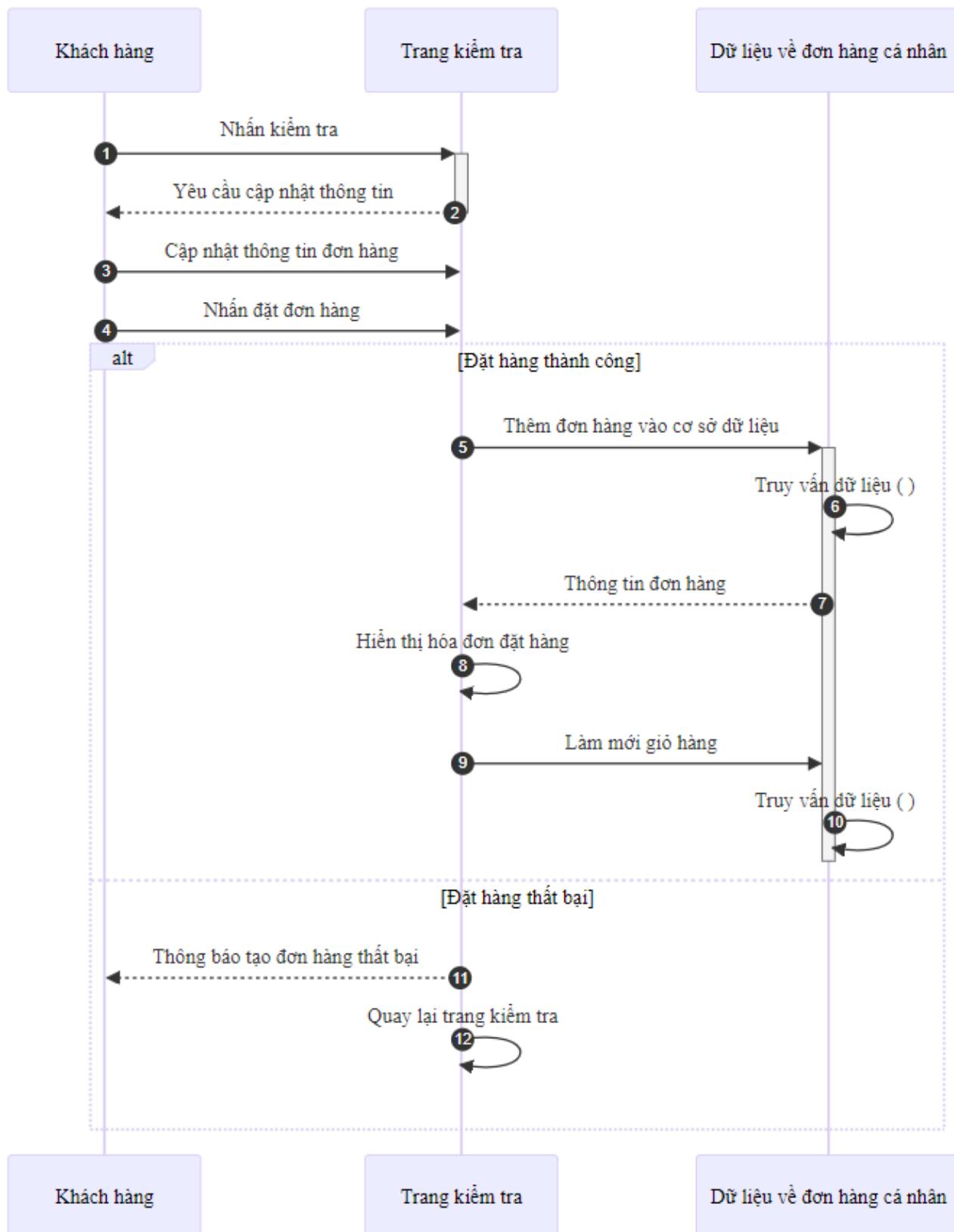
Hình 3.4.3.6. Biểu đồ tuần tự Giỏ hàng (Xóa sản phẩm)



Hình 3.4.3.7. Biểu đồ tuần tự Giỏ hàng (Cập nhật số lượng)

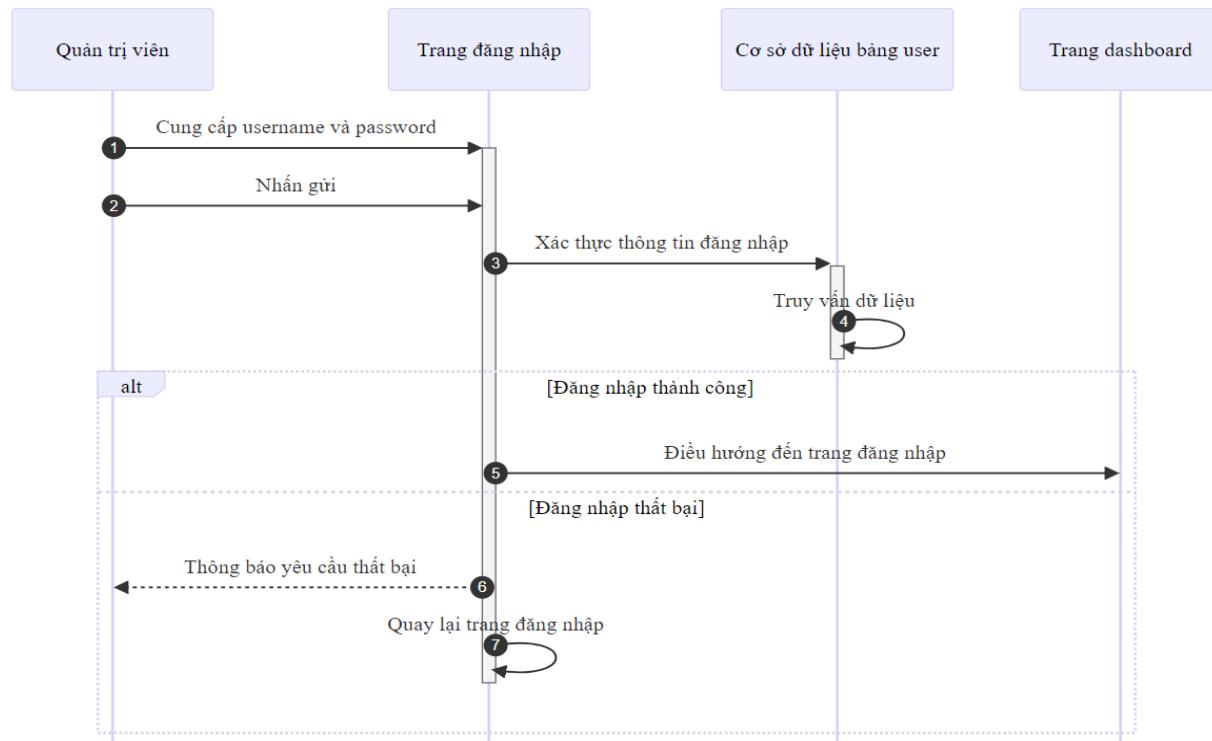


Hình 3.4.3.8. Biểu đồ tuần tự Quản lý hồ sơ cá nhân

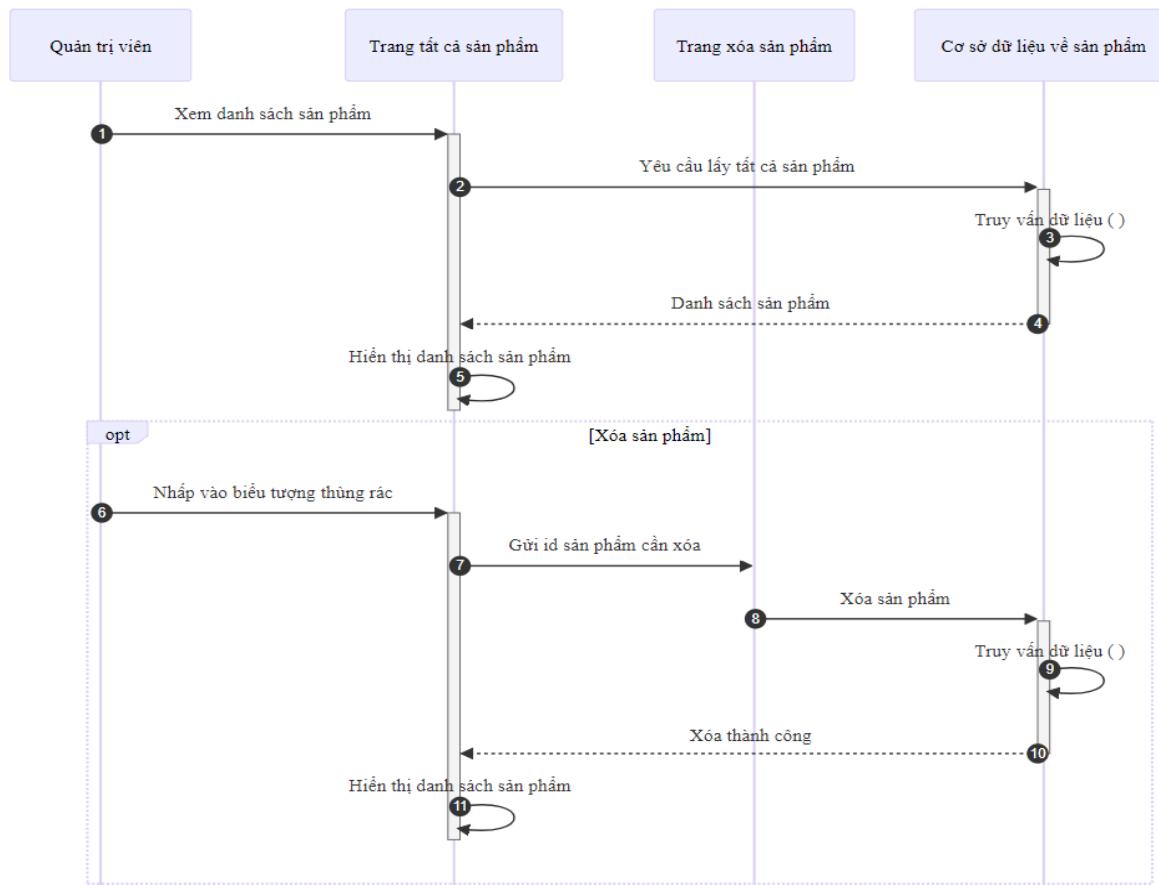


Hình 3.4.3.9. Biểu đồ tuần tự Đơn hàng

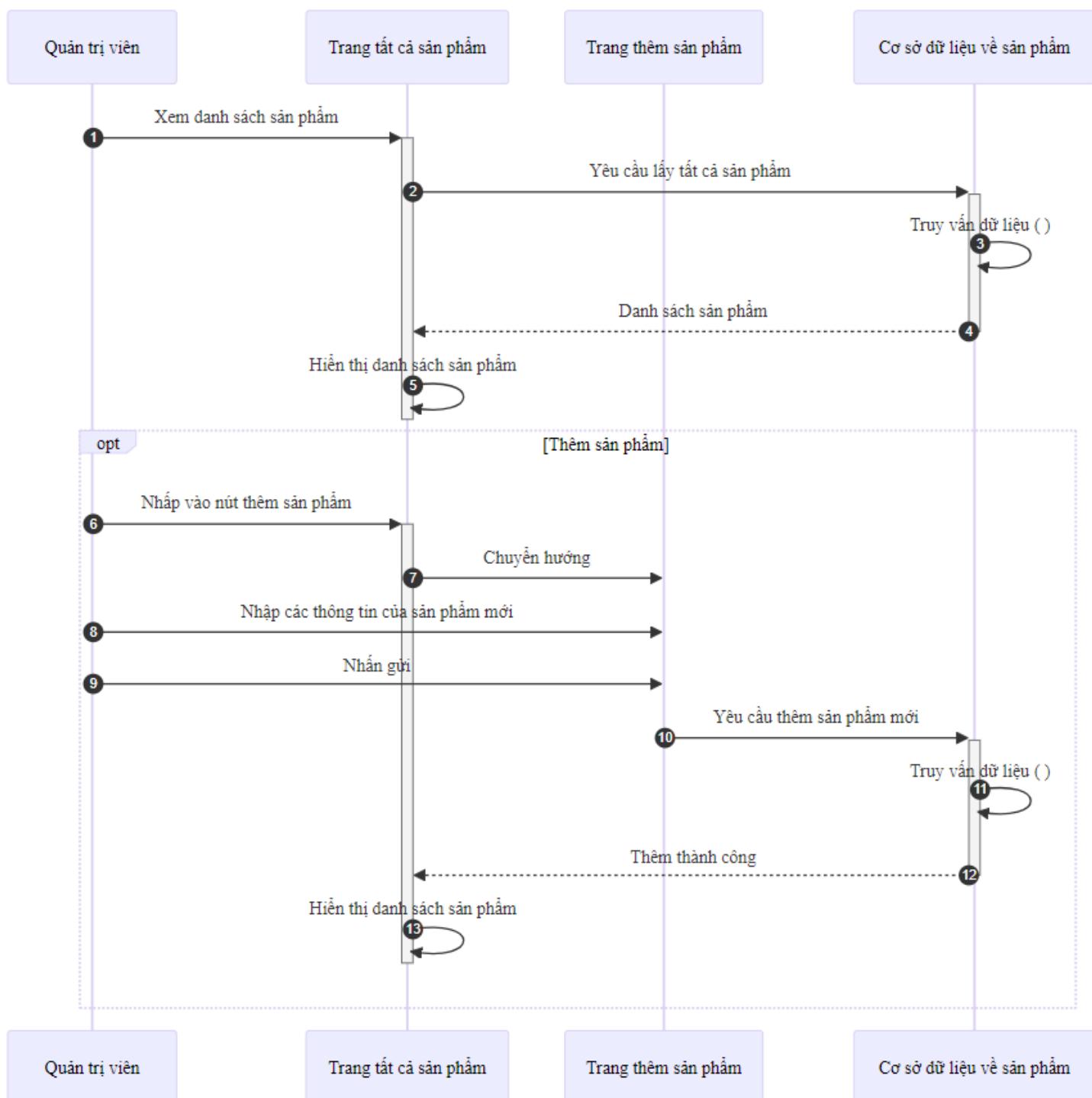
Quản trị viên:



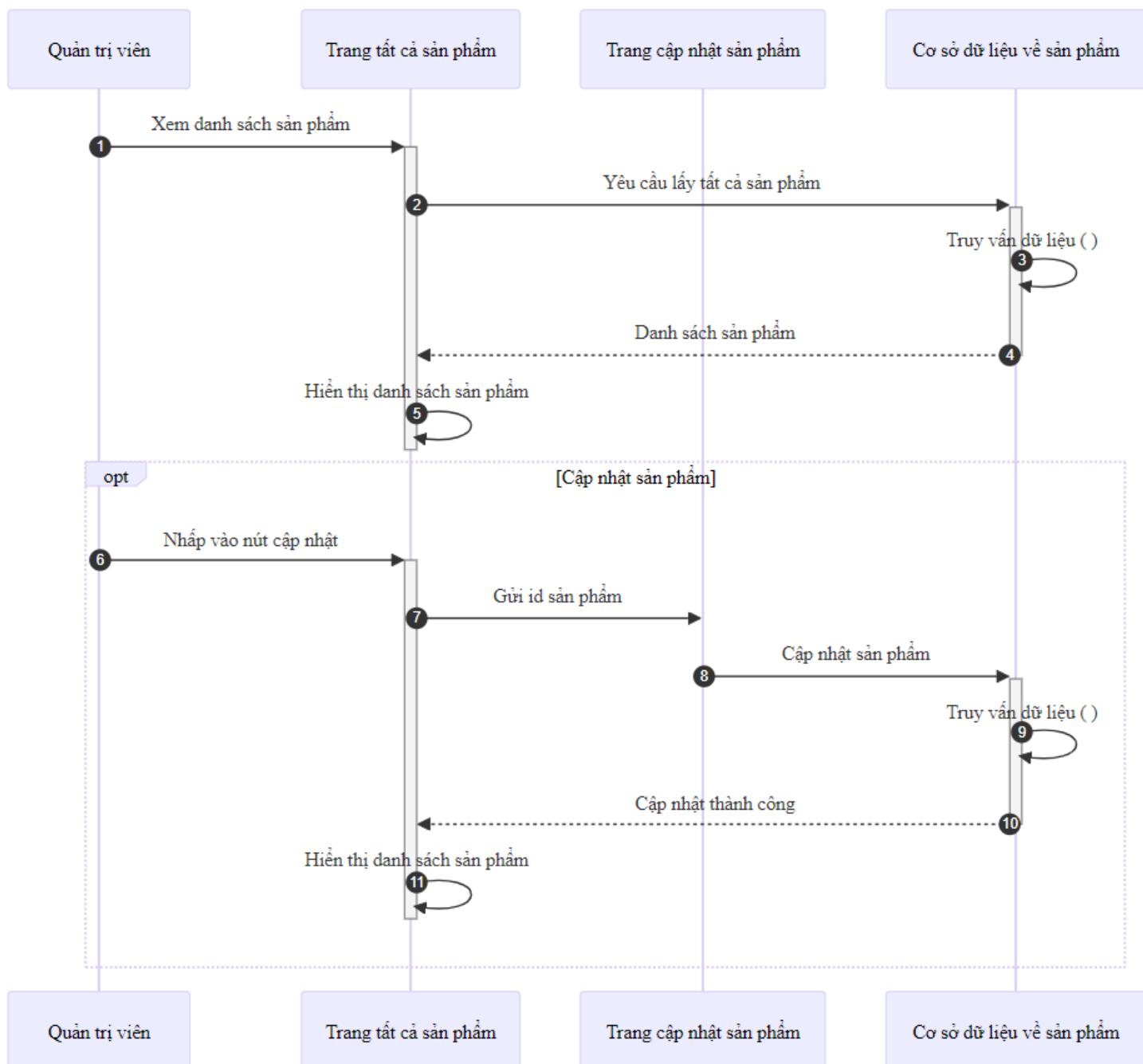
Hình 3.4.3.10. Biểu đồ tuần tự Đăng nhập trang quản trị viên



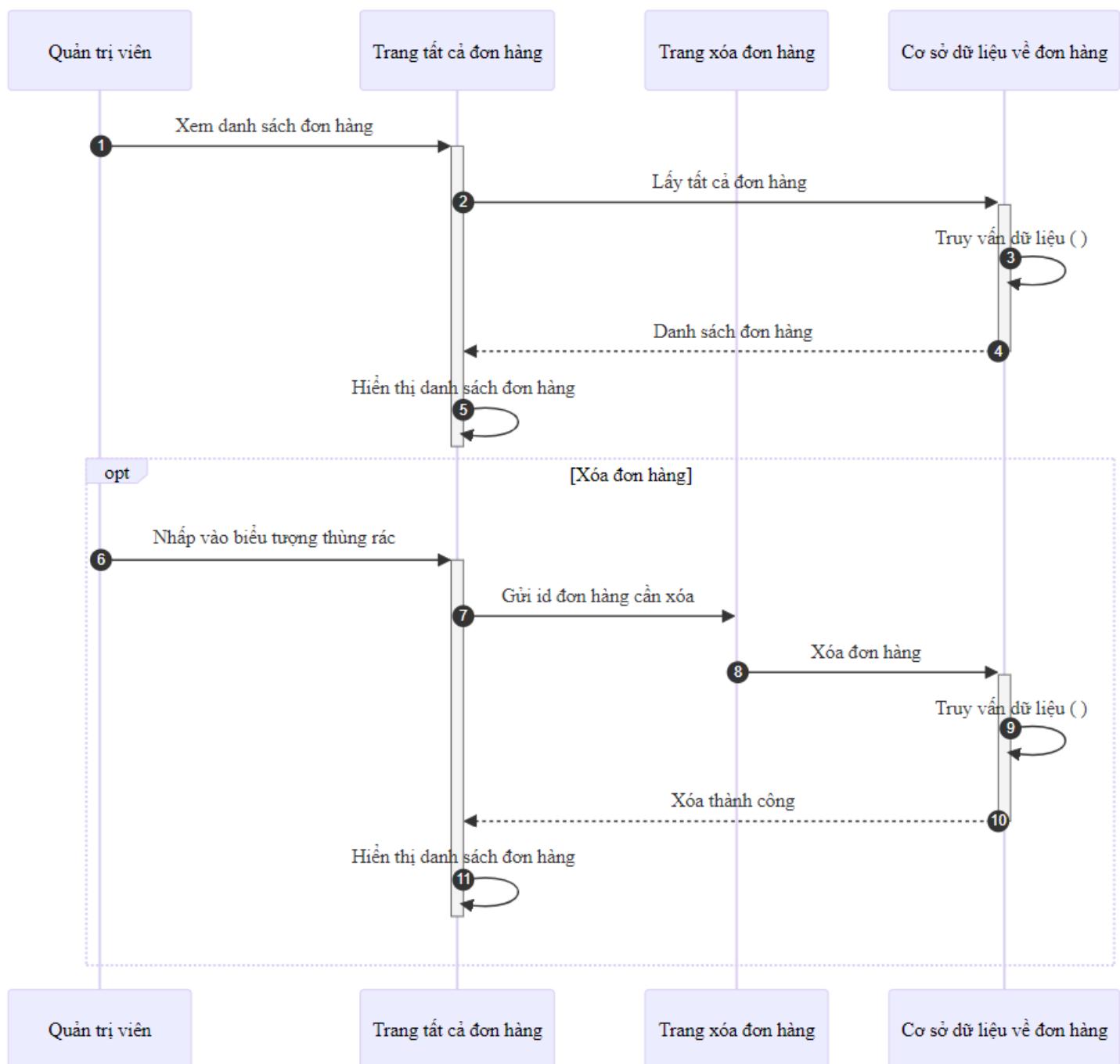
Hình 3.4.3.11. Biểu đồ tuần tự Quản lý sản phẩm (Xóa sản phẩm)



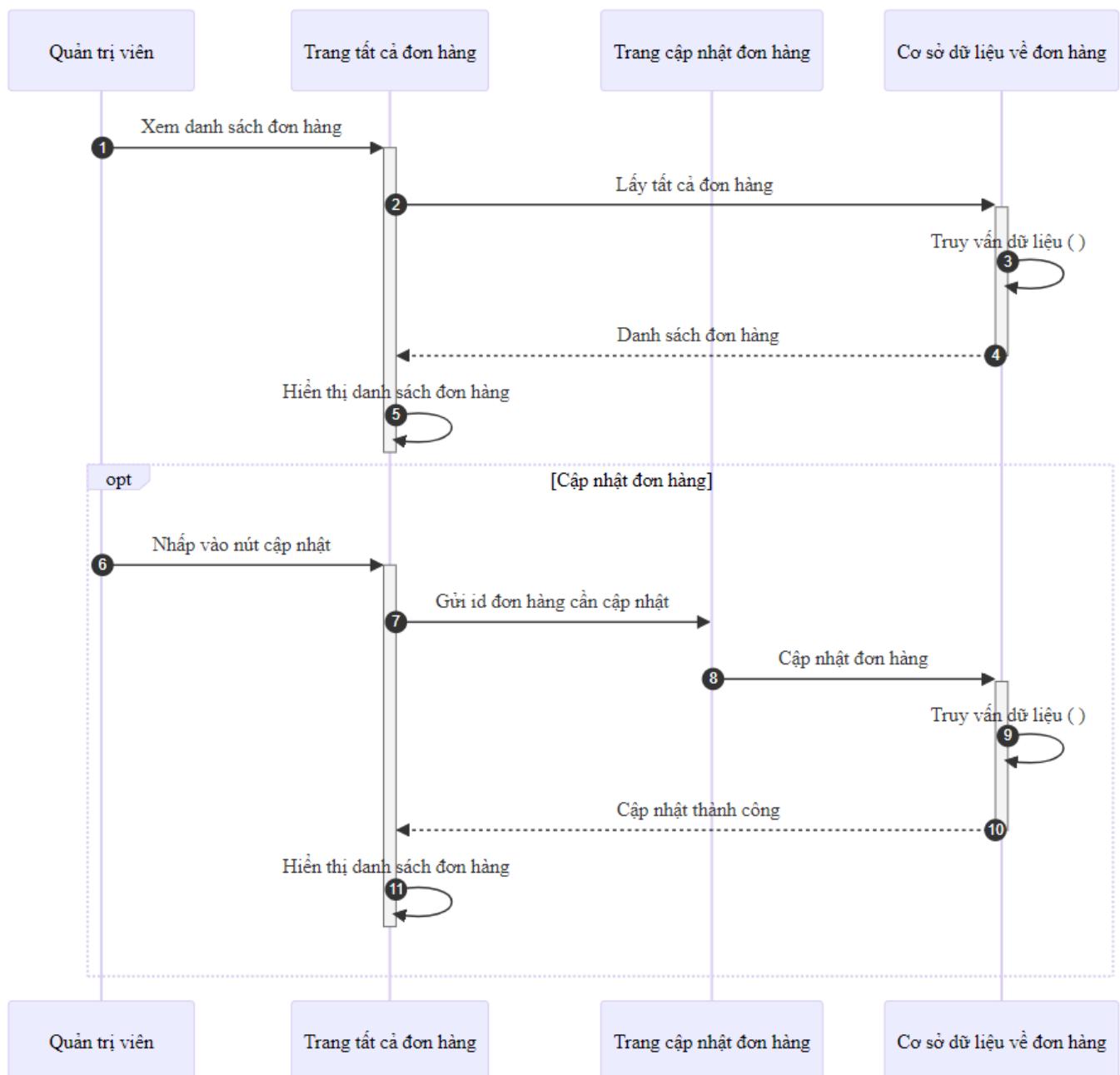
Hình 3.4.3.12. Biểu đồ tuần tự Quản lý sản phẩm (Thêm sản phẩm)



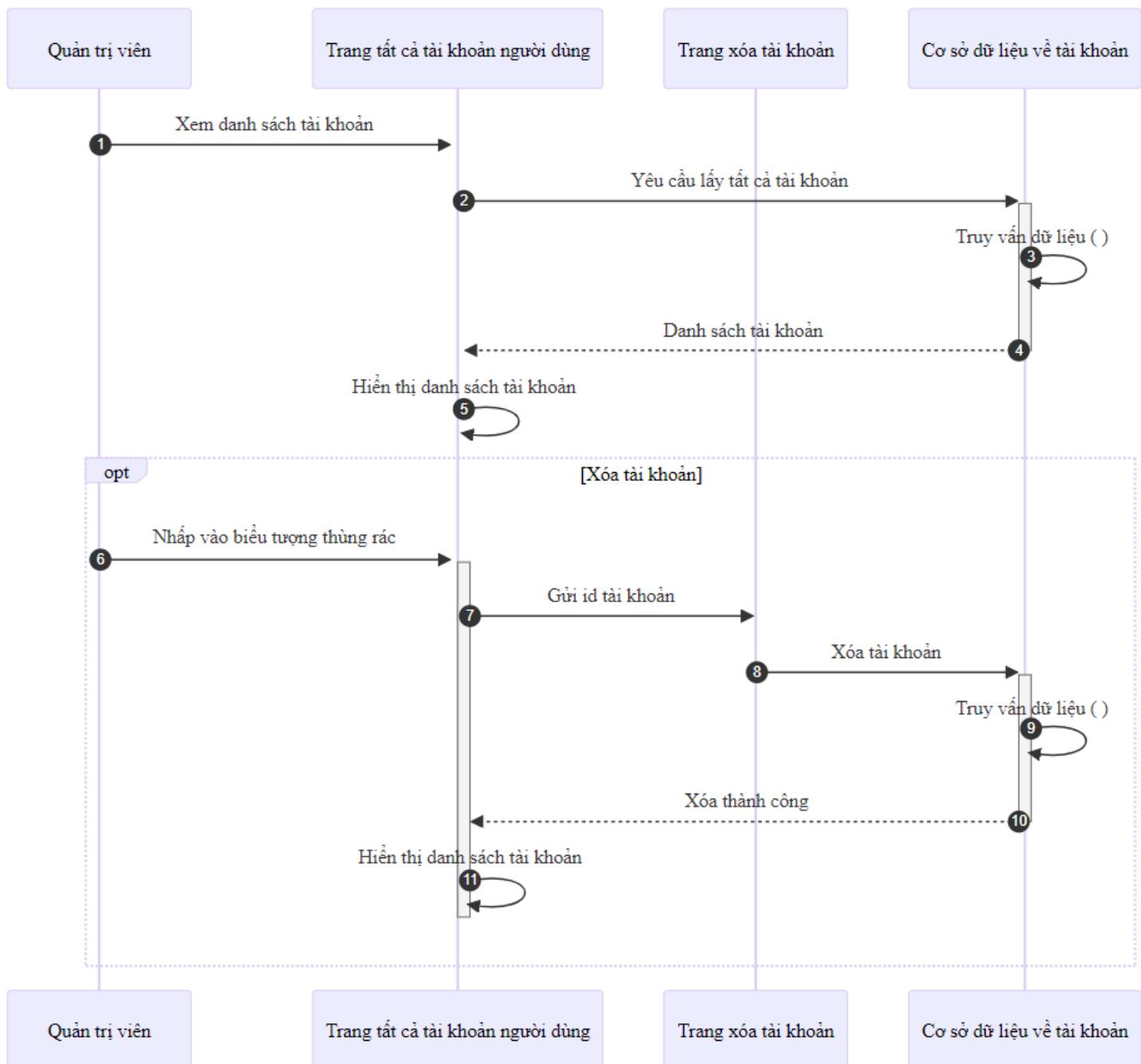
Hình 3.4.3.13. Biểu đồ tuần tự Quản lý đơn hàng (Cập nhật sản phẩm)



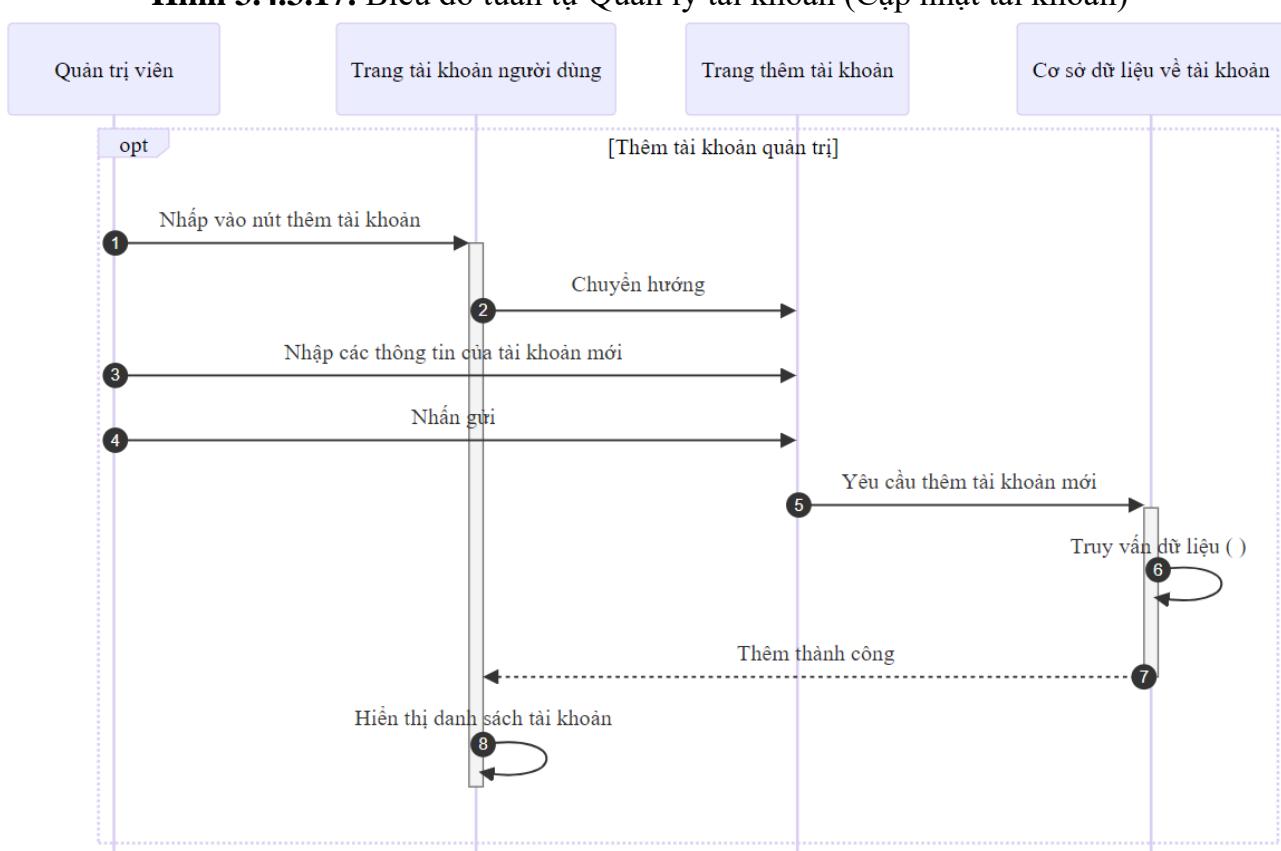
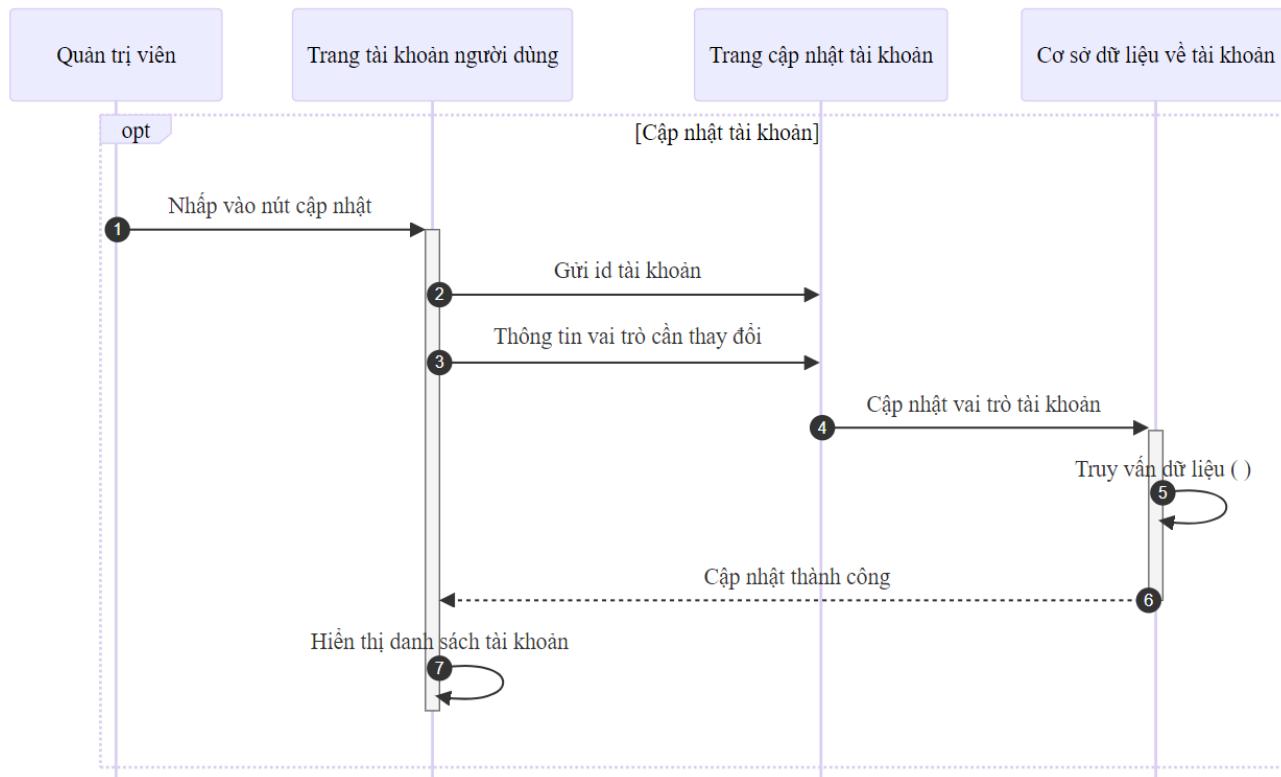
Hình 3.4.3.14. Biểu đồ tuần tự Quản lý đơn hàng (Xóa đơn hàng)



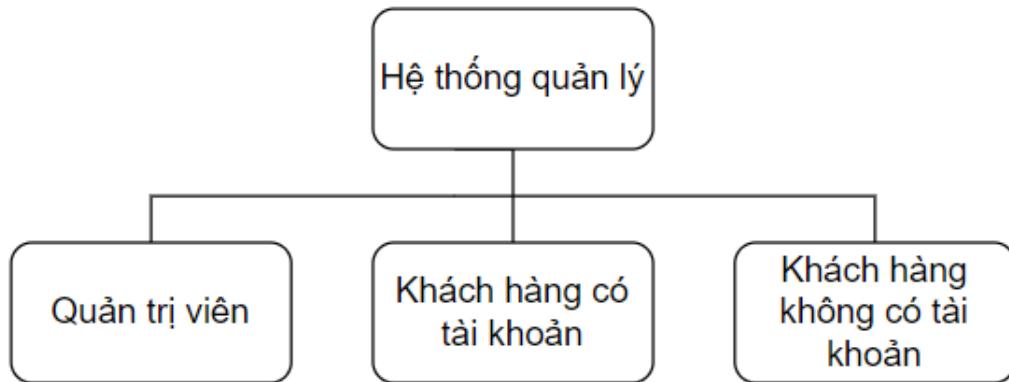
Hình 3.4.3.15. Biểu đồ tuần tự Quản lý đơn hàng (Cập nhật đơn hàng)



Hình 3.4.3.16. Biểu đồ tuần tự Quản lý tài khoản (Xóa tài khoản)

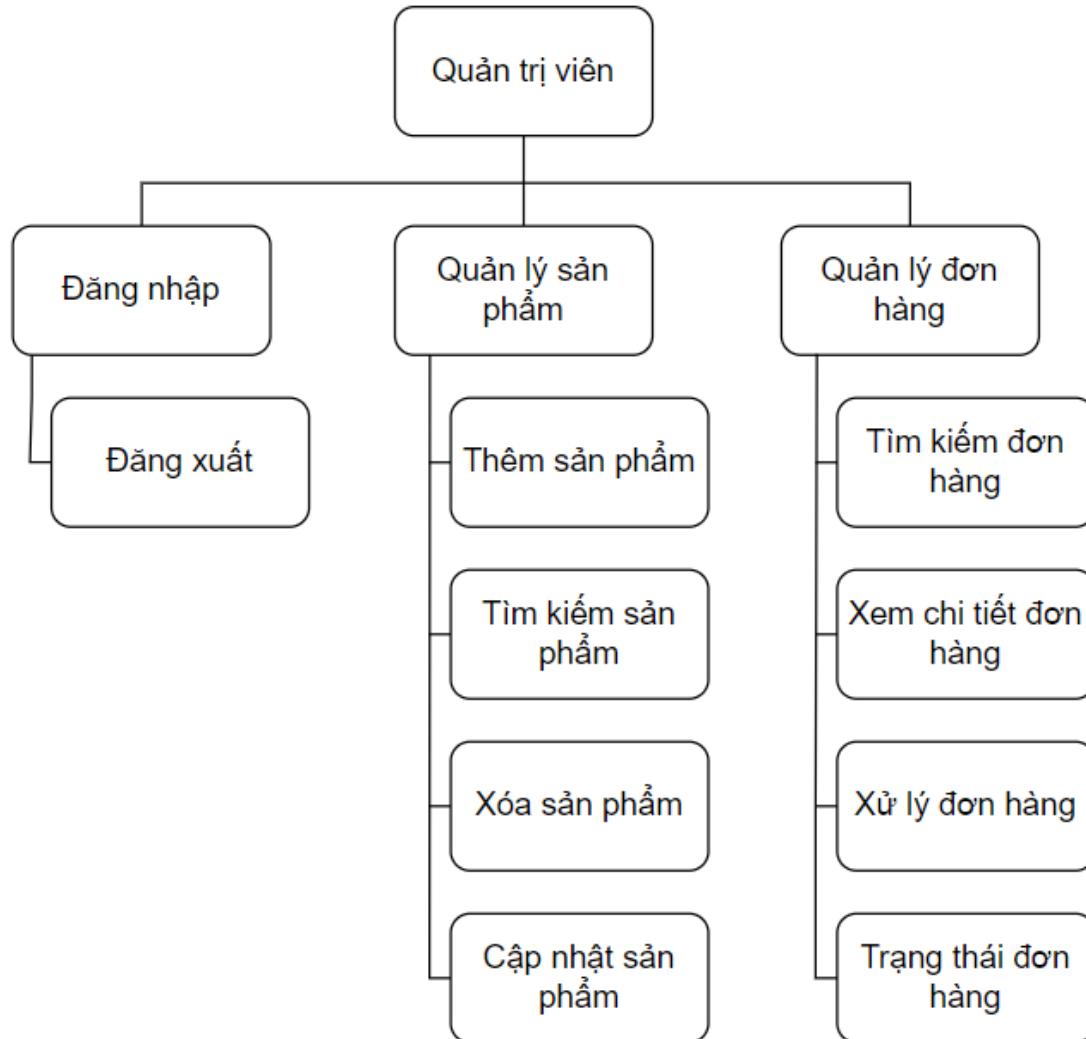


3.4.4. Mô hình phân rã chức năng (BFD)

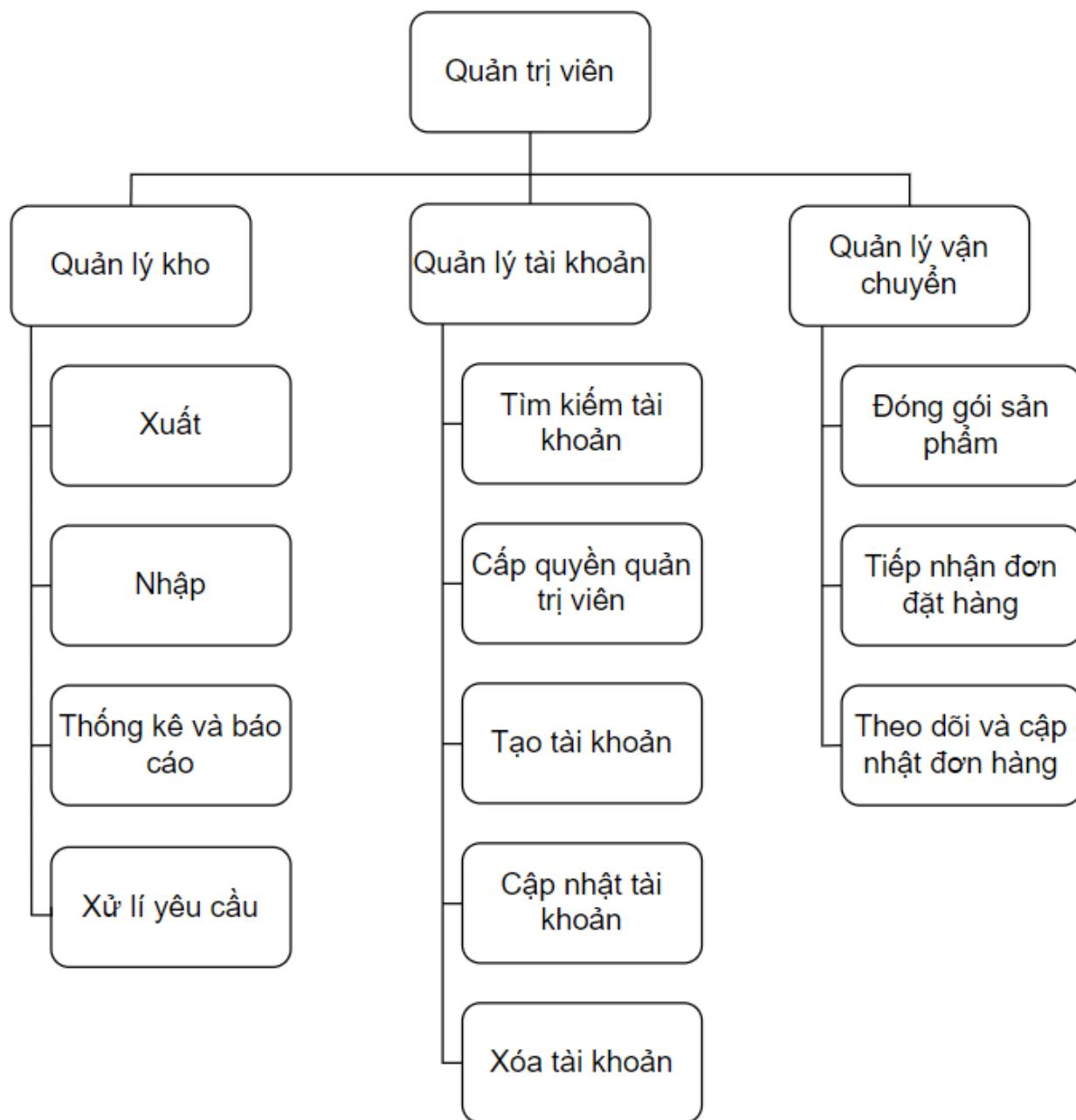


Hình 3.4.4.1. Sơ đồ phân rã chức năng Hệ thống quản lý bán sản phẩm

Hệ thống quản lý bán sản phẩm bao gồm 3 đối tượng chính: Quản trị viên, Khách hàng có tài khoản, Khách hàng không có tài khoản (khách vãng lai).

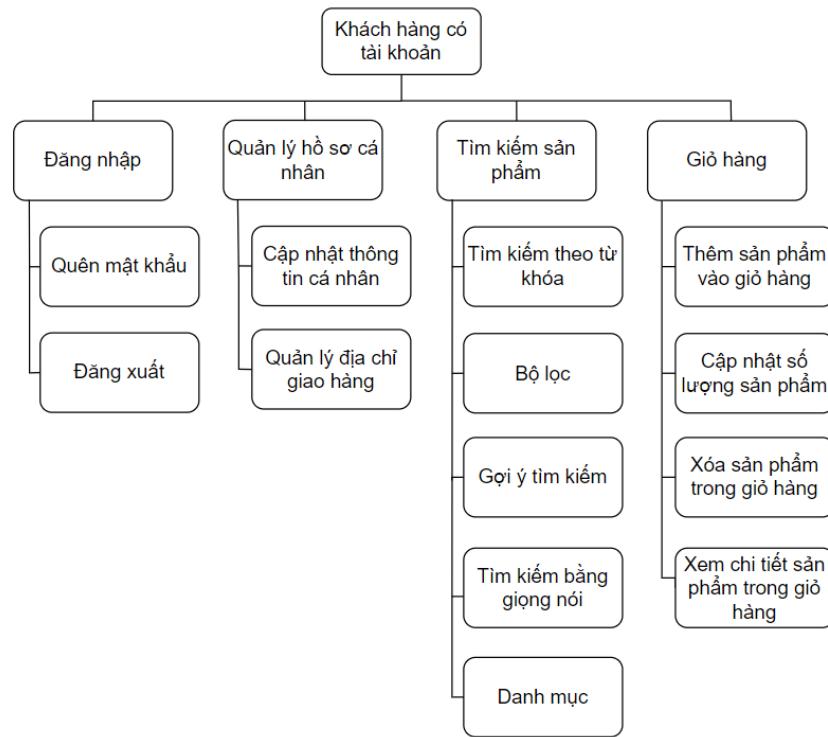


Hình 3.4.4.2. Sơ đồ phân rã chức năng chi tiết của Quản trị viên (phần 1)



Hình 3.4.4.3. Sơ đồ phân rã chức năng chi tiết của Quản trị viên (phần 2)

Quản trị viên có các chức năng như: Đăng nhập, Đăng xuất, Quản lý kho (xuất, nhập, thống kê và báo cáo, số lượng sản phẩm tồn kho, xử lý yêu cầu), Quản lý sản phẩm (thêm, xóa, chỉnh sửa thông tin, tìm kiếm sản phẩm), Quản lý tài khoản (cấp quyền quản trị viên, tìm kiếm/xóa tài khoản), Quản lý đơn hàng (xử lý giao dịch, theo dõi và báo cáo, bảo mật, xác thực), Quản lý vận chuyển (đóng gói sản phẩm, tiếp nhận đơn đặt hàng, theo dõi và cập nhật đơn hàng, vận chuyển, phương thức vận chuyển).



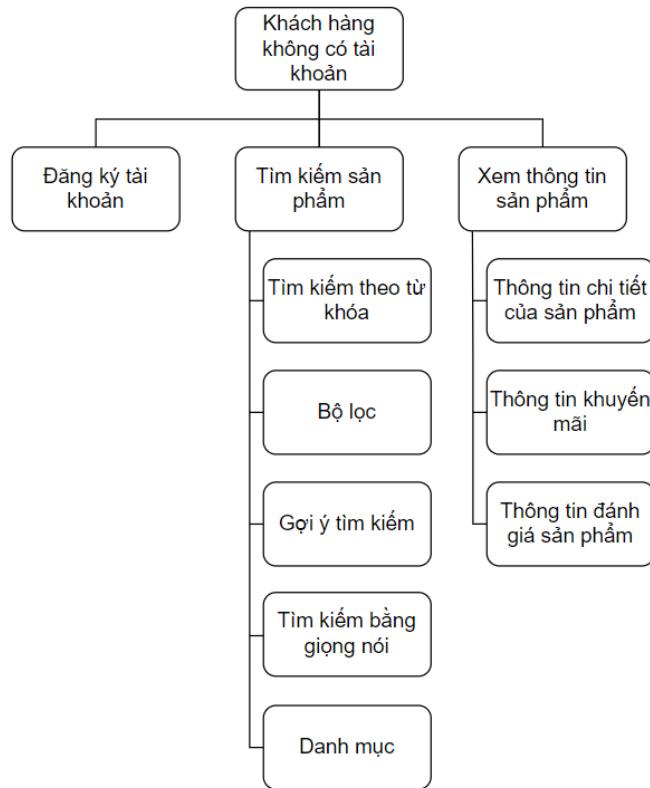
Hình 3.4.4.4. Sơ đồ phân rã chức năng chi tiết của Khách hàng có tài khoản (phần 1)



Hình 3.4.4.5. Sơ đồ phân rã chức năng chi tiết của Khách hàng có tài khoản (phần 2)

Khách hàng có tài khoản có các chức năng: Đăng nhập (quên mật khẩu), Đăng xuất, Quản lý hồ sơ cá nhân (cập nhật thông tin cá nhân, quản lý địa chỉ giao hàng), Tìm kiếm sản phẩm (tìm kiếm theo từ khóa, bộ lọc, gợi ý tìm kiếm, danh mục), Giỏ hàng (thêm, xóa, cập

nhật, xem chi tiết sản phẩm trong giỏ hàng), Thanh toán (áp mã giảm giá, chọn phương thức thanh toán, nhập thông tin thanh toán và giao hàng, xác nhận thanh toán), Đơn hàng (xem lịch sử/theo dõi trạng thái đơn hàng, đánh giá và viết nhận xét về sản phẩm), Xem thông tin sản phẩm (thông tin chi tiết/khuyến mãi/đánh giá sản phẩm).

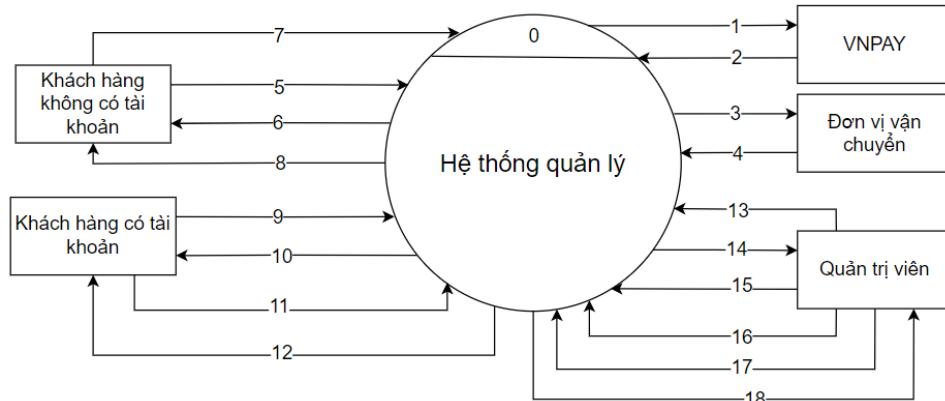


Hình 3.4.4.6. Sơ đồ phân rã chức năng chi tiết của Khách hàng không có tài khoản

Khách hàng không có tài khoản có các chức năng: Đăng ký tài khoản, Tìm kiếm sản phẩm (tìm kiếm theo từ khóa, bộ lọc, gợi ý tìm kiếm, danh mục), Xem thông tin sản phẩm (thông tin chi tiết/khuyến mãi/đánh giá sản phẩm).

3.4.5. Mô hình dòng dữ liệu (DFD)

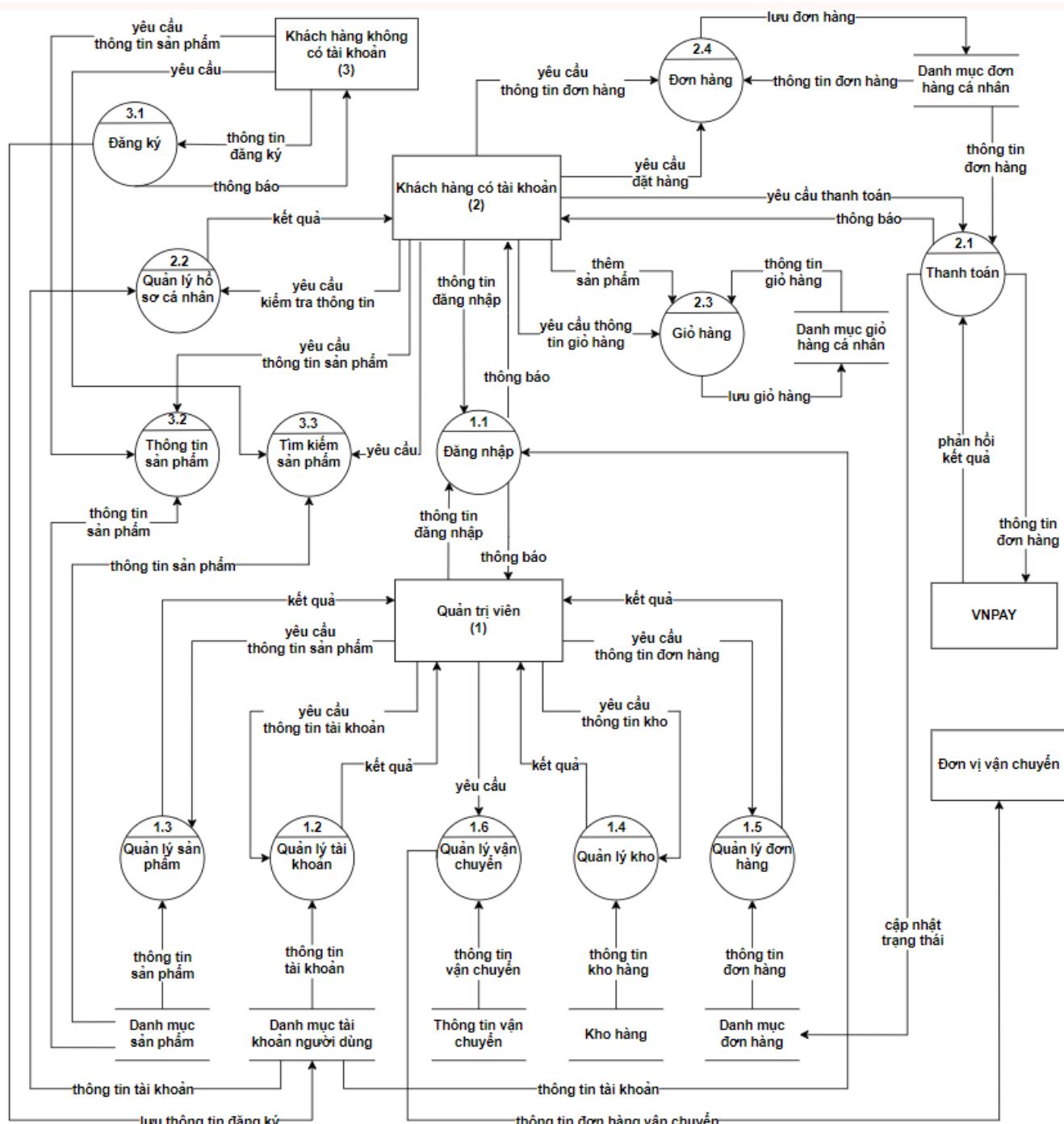
Mức ngũ cảnh



Hình 3.3.5.1. Mô hình hóa xử lý ở mức ngũ cảnh

Trong đó: (1) Thông tin yêu cầu thanh toán, (2) Thông tin thanh toán, (3) Thông tin đơn hàng, (4) Cập nhật thông tin vận chuyển, (5) Yêu cầu kích hoạt tài khoản, (6) Thông báo, (7) Yêu cầu xem sản phẩm, (8) Kết quả, (9) Yêu cầu đăng nhập, (10) Thông báo, (11) Yêu cầu đặt hàng, (12) Kết quả, (13) Yêu cầu đăng nhập, (14) Thông báo, (15) Yêu cầu xem danh sách tài khoản, (16) Yêu cầu xem danh sách sản phẩm, (17) Yêu cầu xem danh sách đơn hàng, (18) Kết quả.

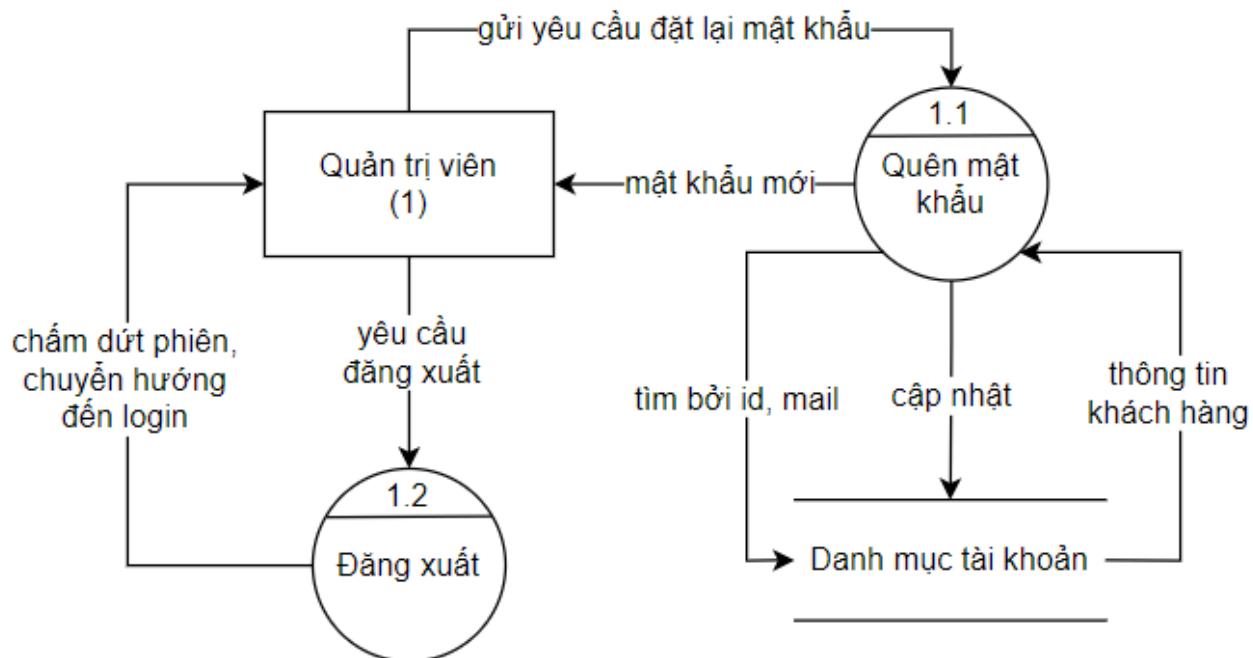
Mức đỉnh



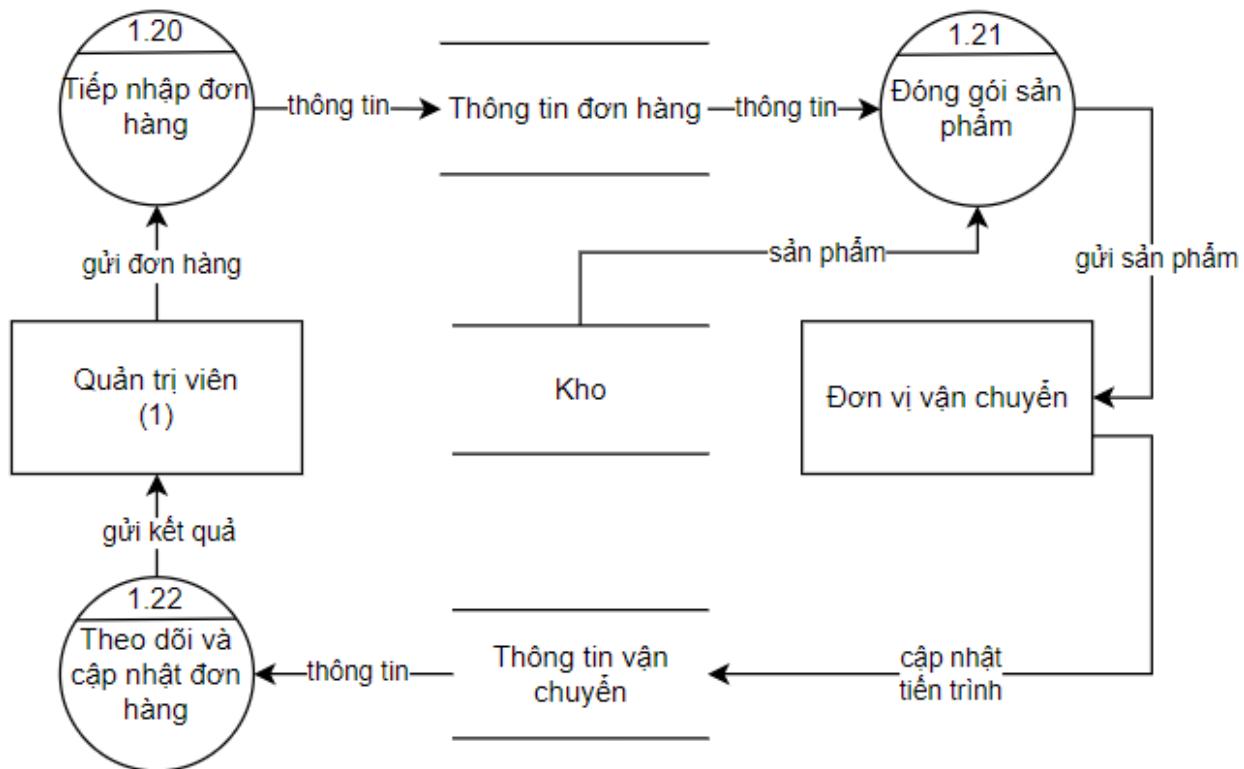
Hình 3.4.5.2. Mô hình hóa xử lý ở mức đỉnh

Mức dưới định

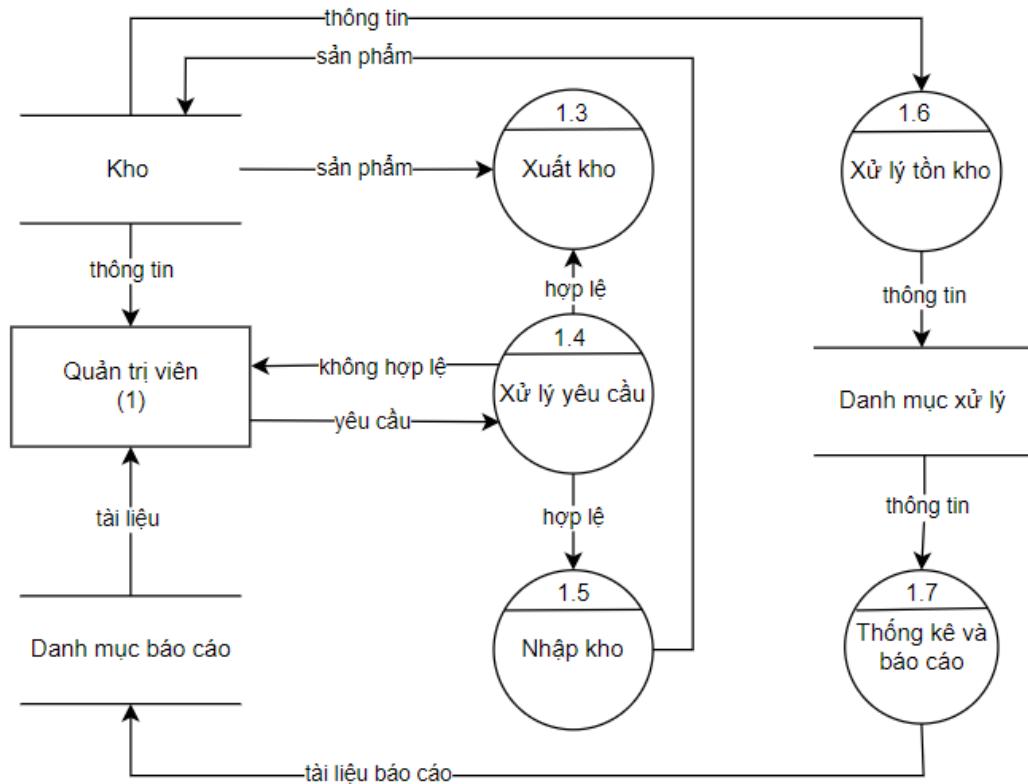
Quản trị viên:



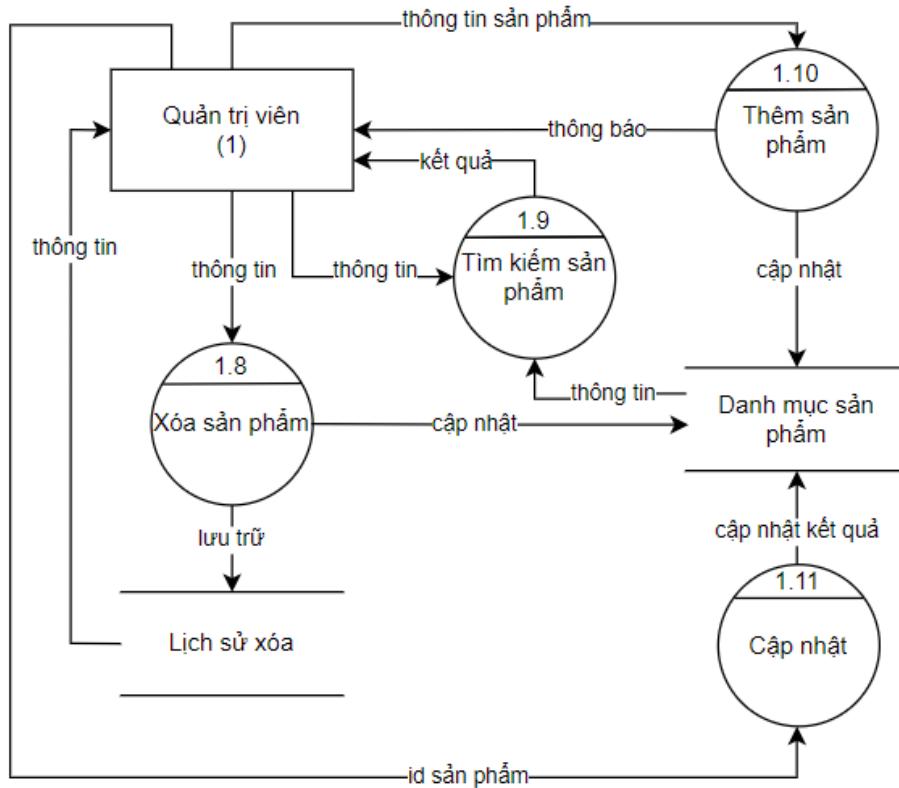
Hình 3.4.5.3. Sơ đồ luồng dữ liệu chi tiết của Đăng nhập



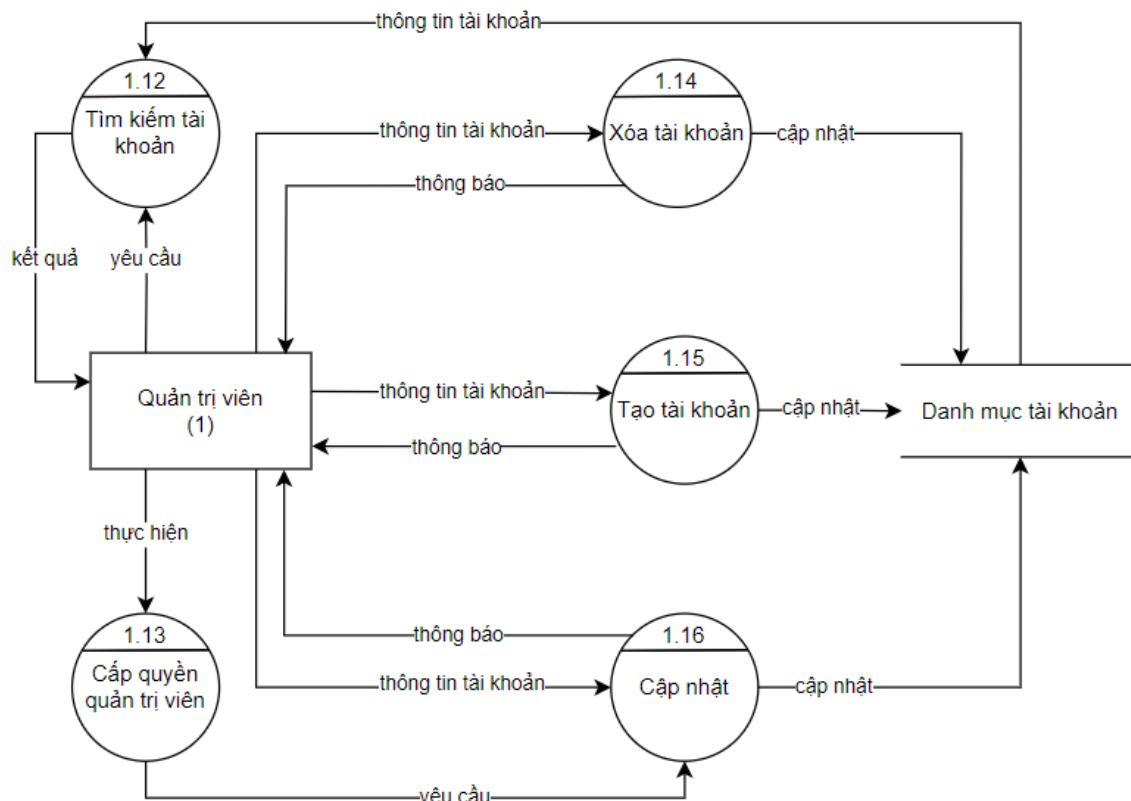
Hình 3.4.5.4. Sơ đồ luồng dữ liệu chi tiết của Quản lý vận chuyển



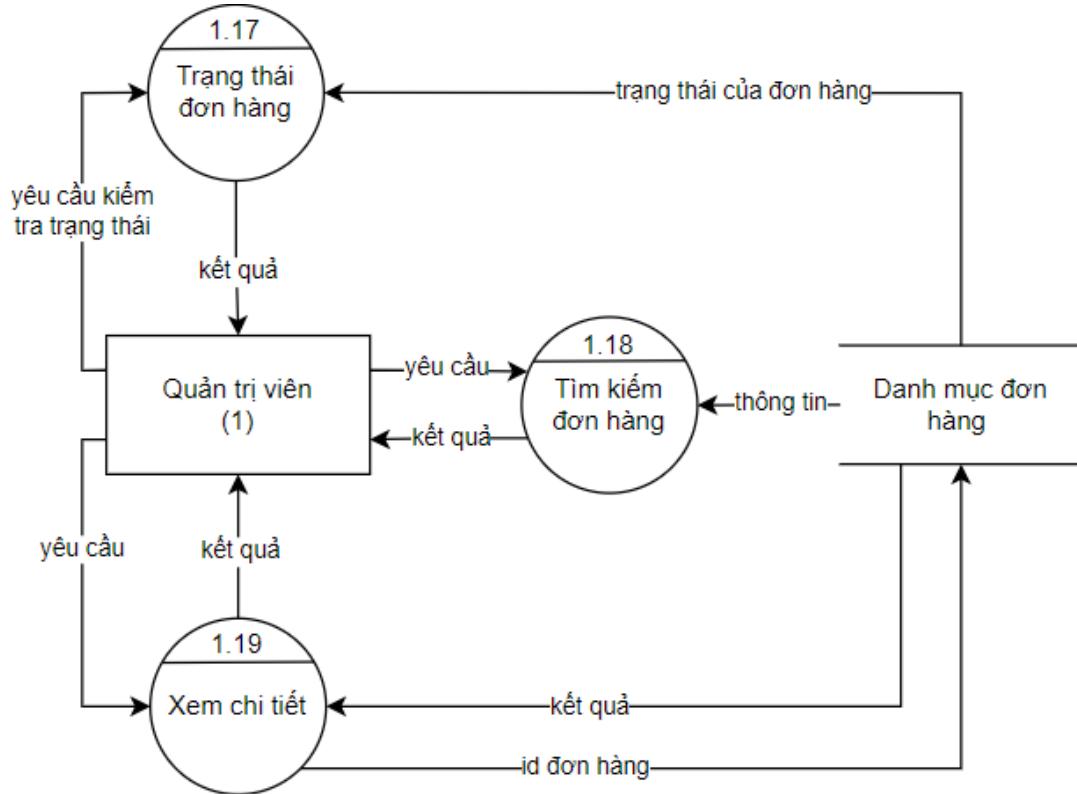
Hình 3.4.5.5. Sơ đồ luồng dữ liệu chi tiết của Quản lý kho



Hình 3.4.5.6. Sơ đồ luồng dữ liệu chi tiết của Quản lý sản phẩm

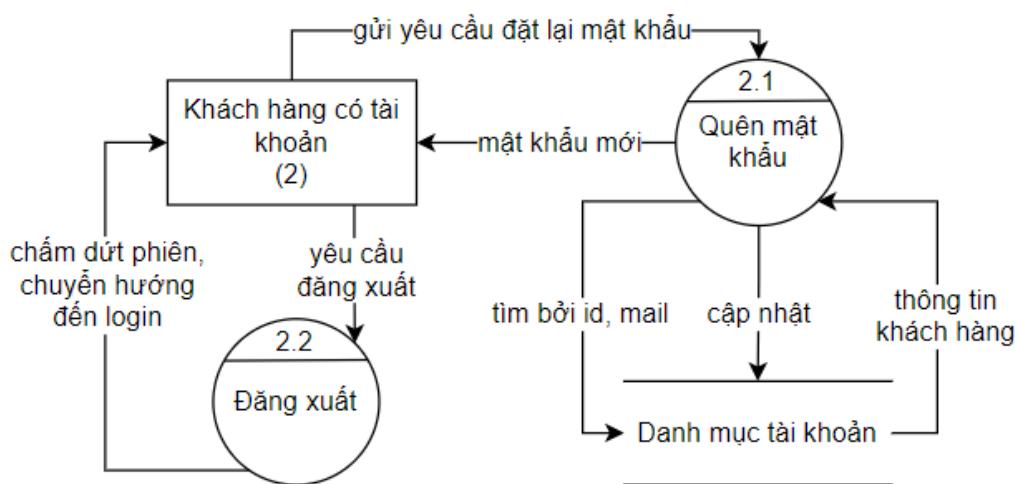


Hình 3.4.5.7. Sơ đồ luồng dữ liệu chi tiết của Quản lý tài khoản

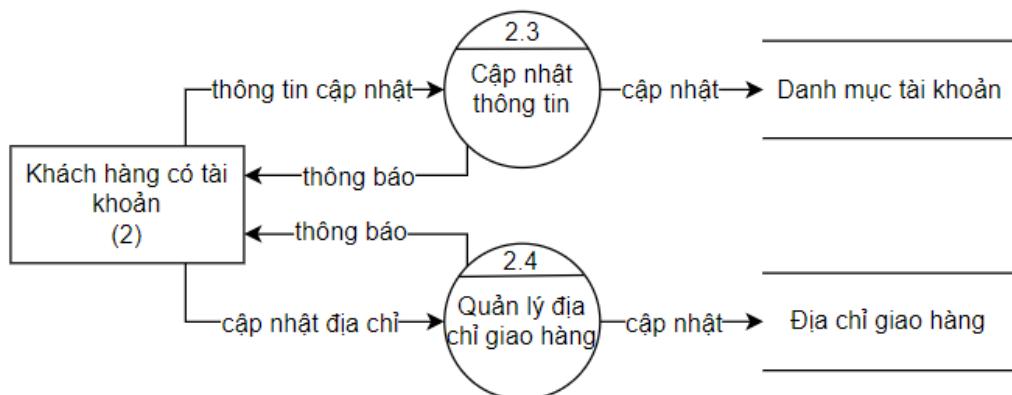


Hình 3.4.5.8. Sơ đồ luồng dữ liệu chi tiết của Quản lý đơn hàng

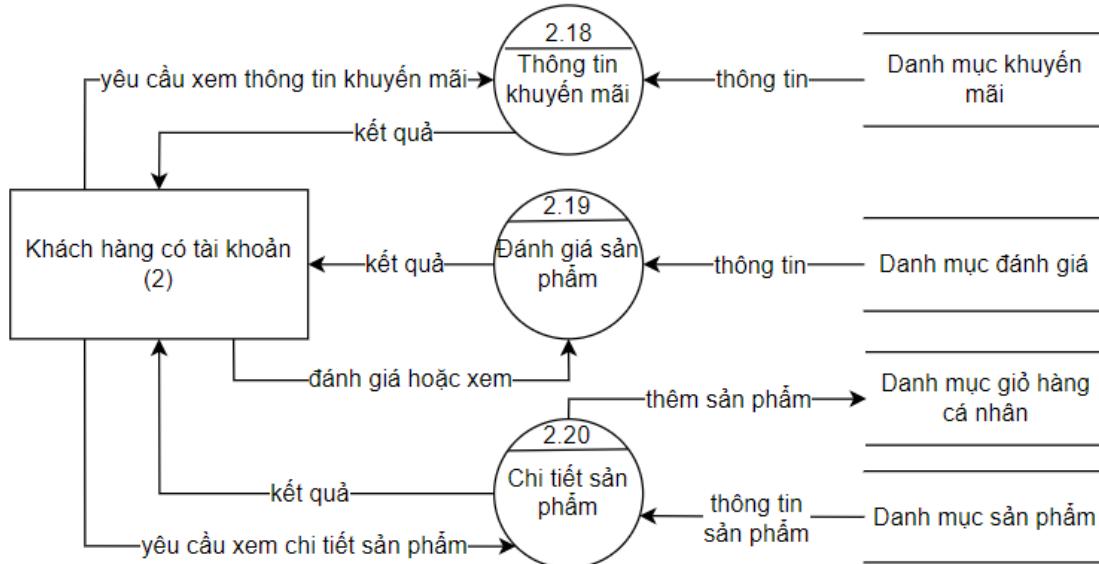
Khách hàng có tài khoản:



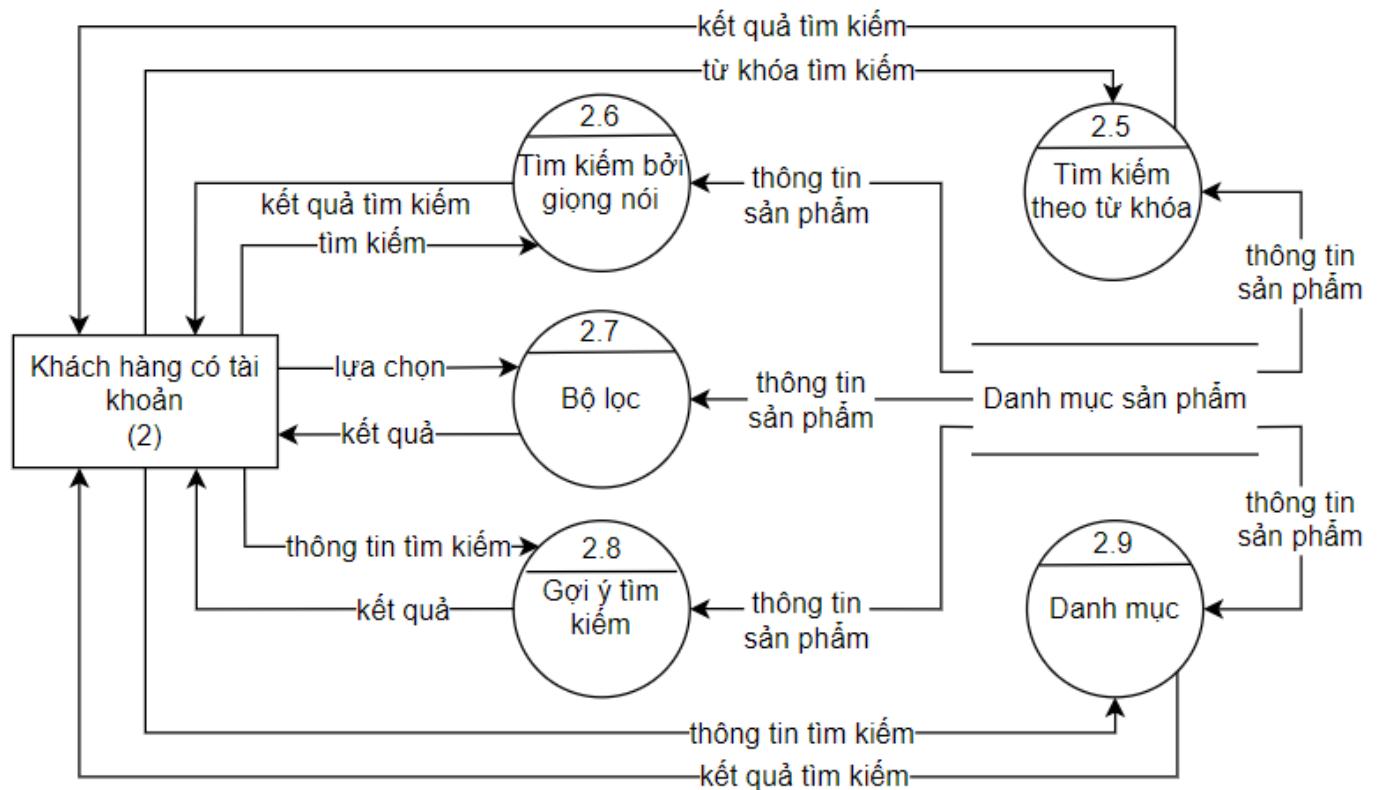
Hình 3.4.5.9. Sơ đồ luồng dữ liệu chi tiết của Đăng nhập



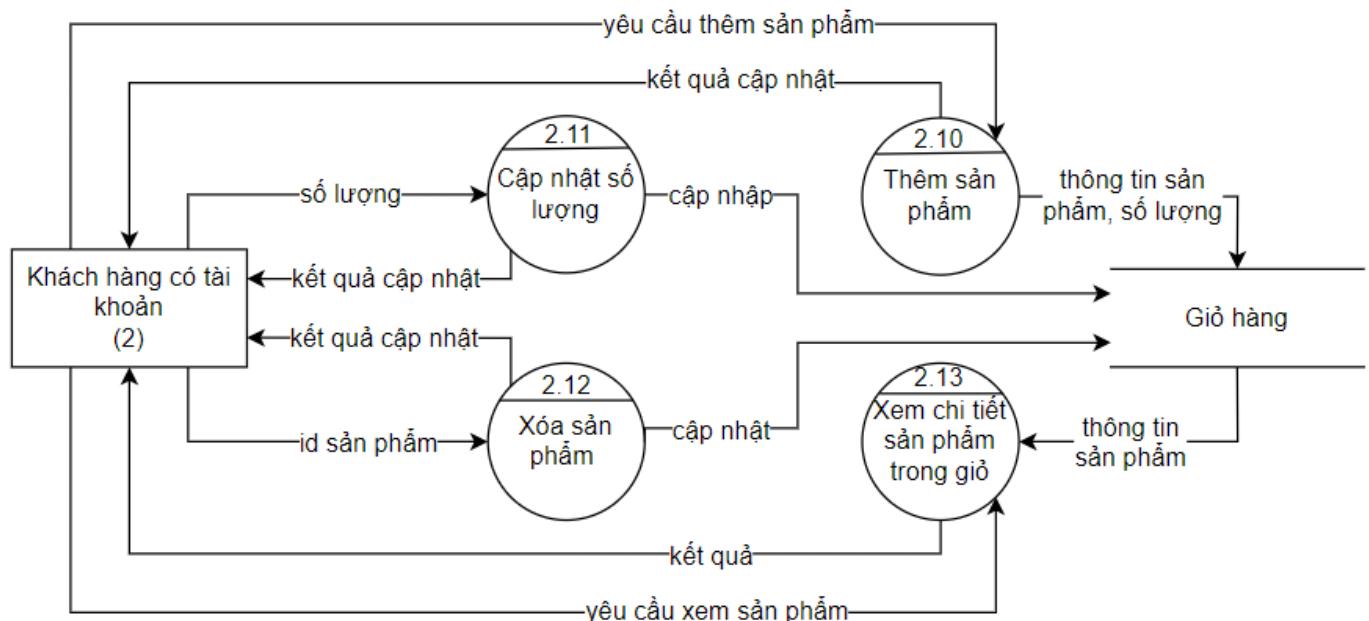
Hình 3.4.5.10. Sơ đồ luồng dữ liệu chi tiết của Quản lý hồ sơ cá nhân



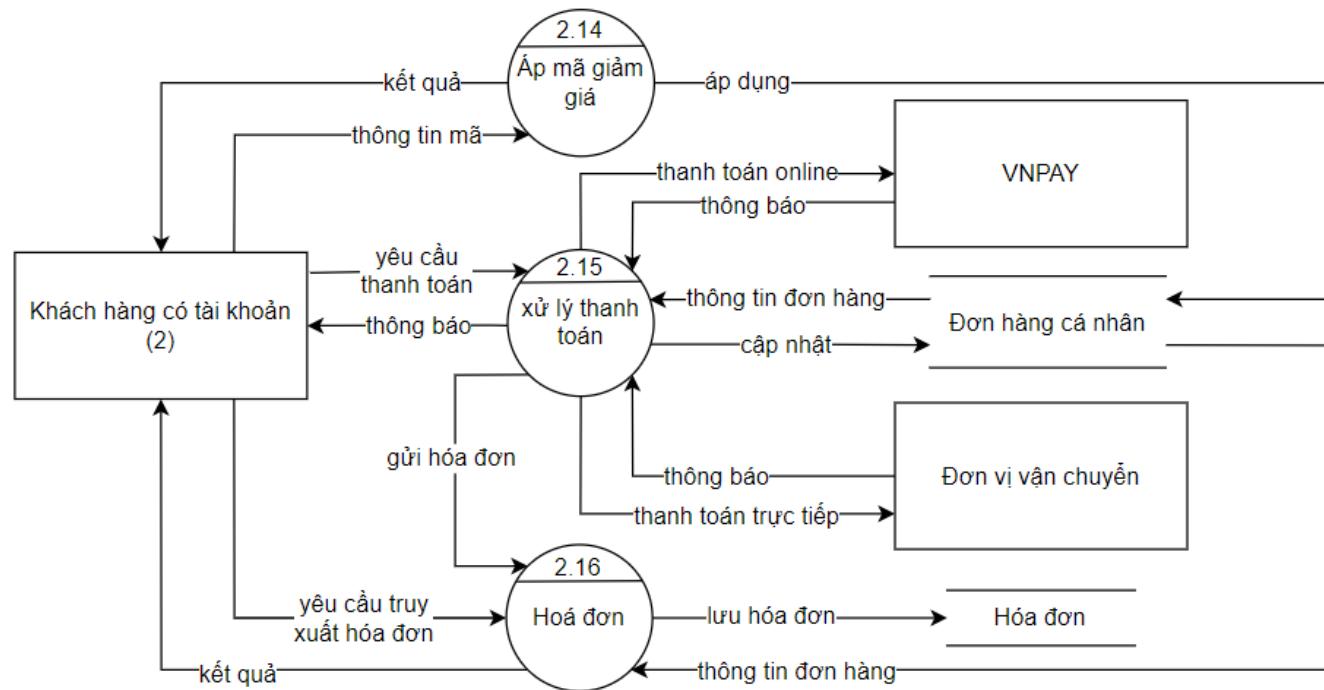
Hình 3.4.5.11. Sơ đồ luồng dữ liệu chi tiết của Xem thông tin sản phẩm



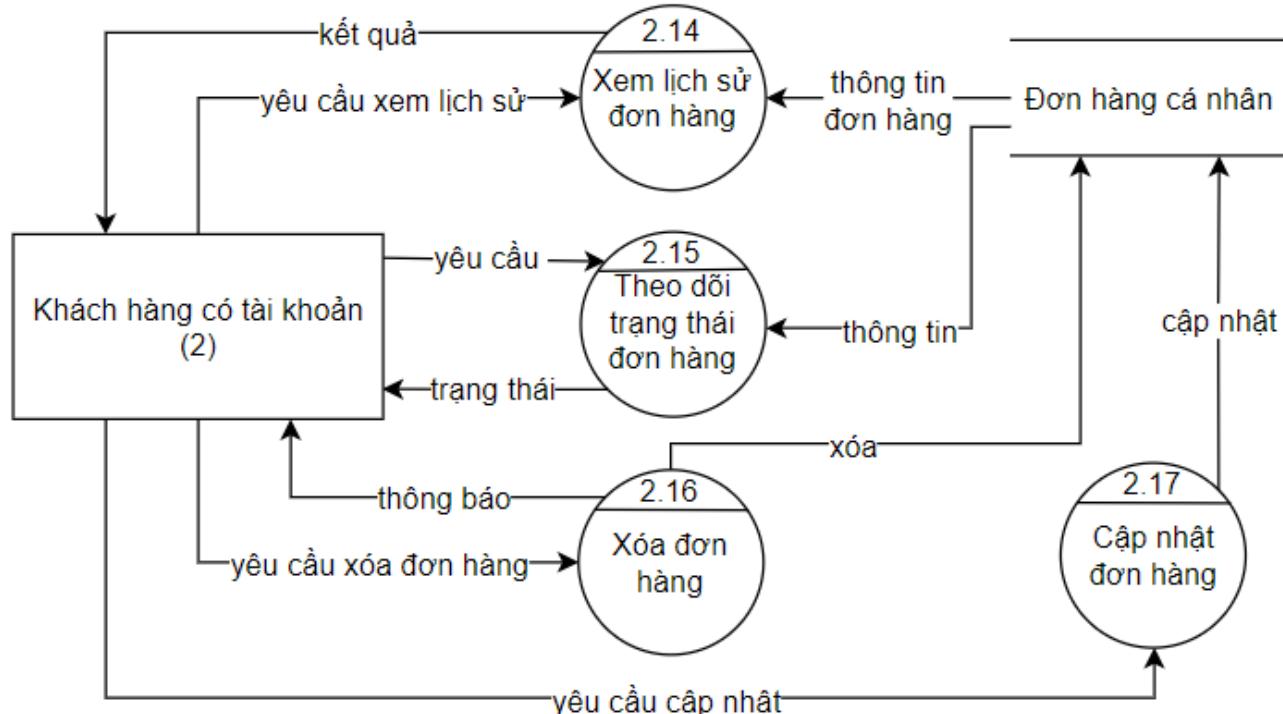
Hình 3.4.5.12. Sơ đồ luồng dữ liệu chi tiết của Tim kiem sản phẩm



Hình 3.4.5.13. Sơ đồ luồng dữ liệu chi tiết của Gioi hang

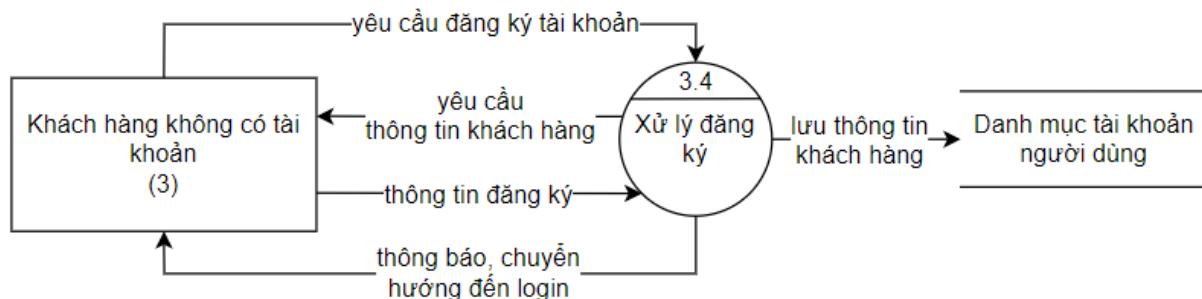


Hình 3.4.5.14. Sơ đồ luồng dữ liệu chi tiết của Thanh toán

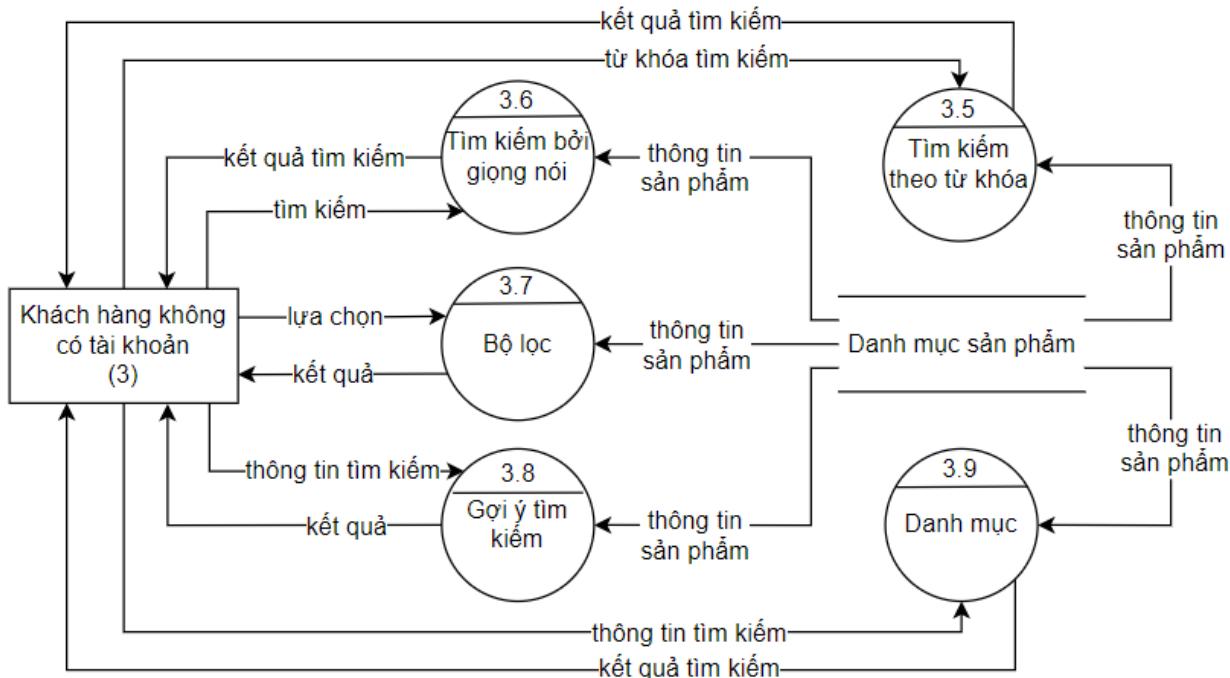


Hình 3.4.5.15. Sơ đồ luồng dữ liệu chi tiết của Đơn hàng

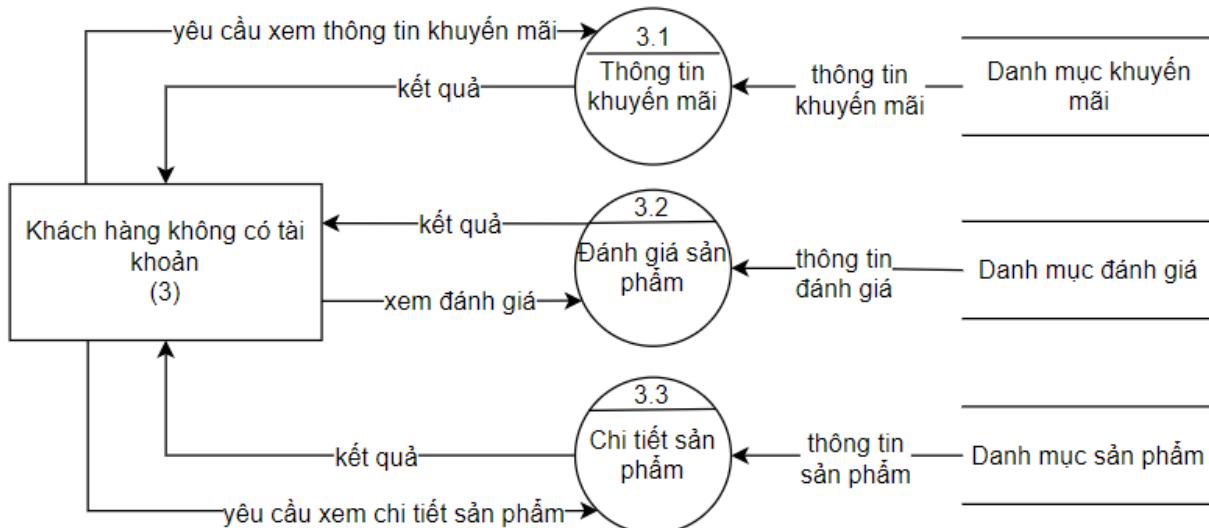
Khách hàng không có tài khoản:



Hình 3.4.5.16. Sơ đồ luồng dữ liệu chi tiết của Đăng ký



Hình 3.4.5.17. Sơ đồ luồng dữ liệu chi tiết của Tìm kiếm sản phẩm



Hình 3.4.5.18. Sơ đồ luồng dữ liệu chi tiết của Xem thông tin sản phẩm

CHƯƠNG 4: KẾT QUẢ CHƯƠNG TRÌNH

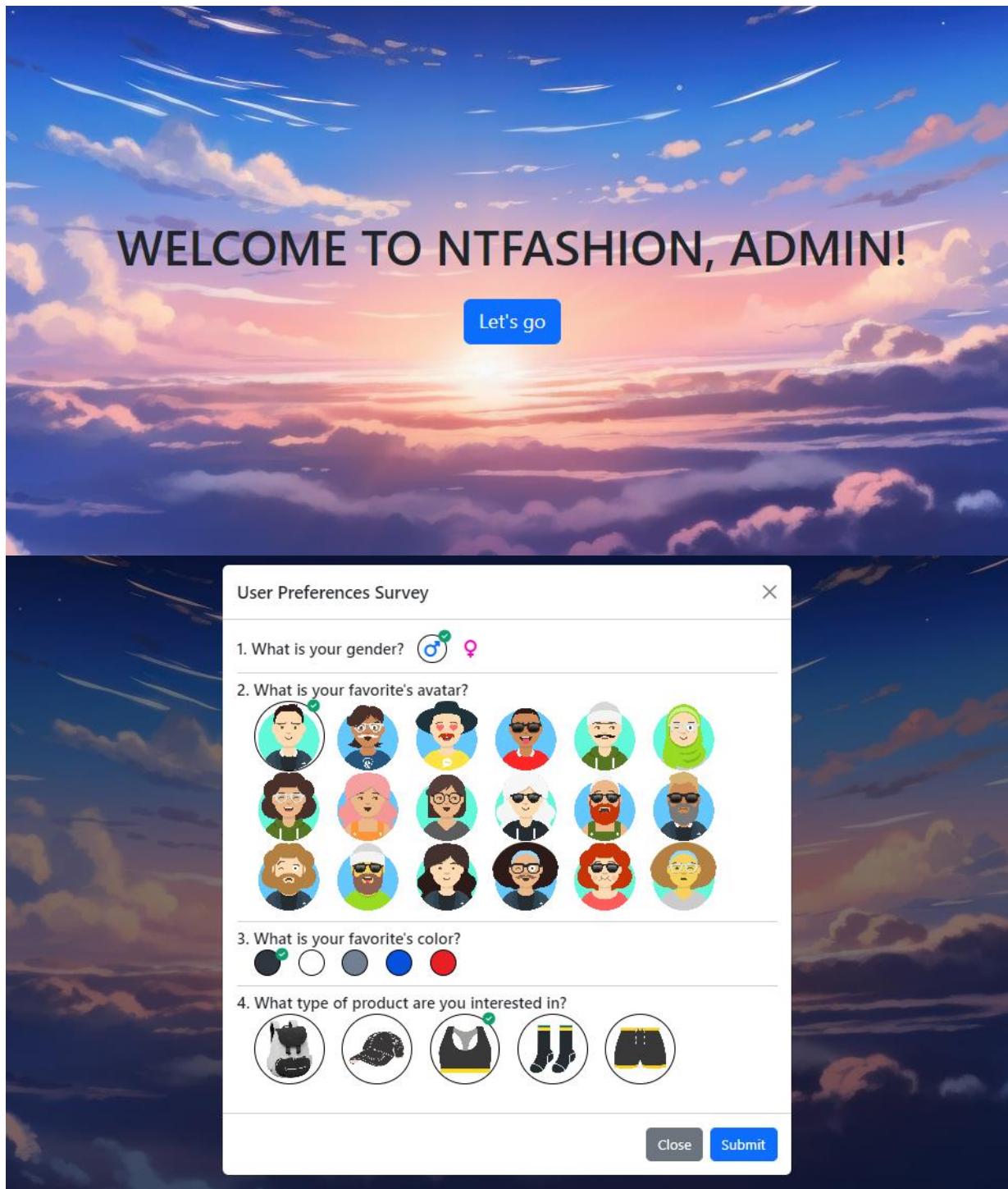
4.1. Quy trình đăng nhập và đăng ký

The login page for SmartShop features a large background image of a woman in workout gear with the text "YOUR TIME HERE" and "25% DISCOUNT". The login form includes fields for Username and Password, a purple "Log in" button, and links for forgot password and create account.

Hình 4.1.1. Trang đăng nhập của khách hàng

The registration page for SmartShop features a large background image of a woman in workout gear with the text "NO FEAR". The registration form includes fields for Email, Username, Phone, Gender, Password, Confirm password, Address, and a checkbox for privacy policy. A purple "Create account" button is at the bottom.

Hình 4.1.2. Trang đăng ký của khách hàng



Hình 4.1.3. Trang khảo sát sở thích của khách hàng

4.2. Quy trình quản lý và hiển thị sản phẩm

✚ Trang chủ



HEALTHY LIFE FOR YOU

Every outfit is a story, and we are the ones writing that story.

 **Free shipping**
Receive goods within 3 days

 **Attractive gifts**
Many hot promotions

 **Quality**
The product has been tested

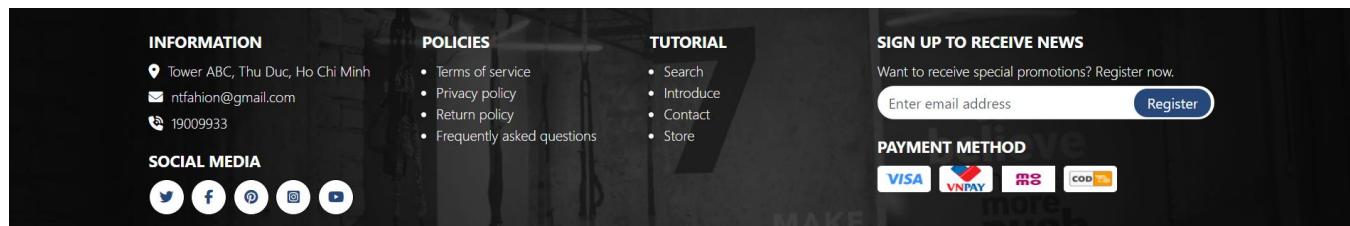
 **Hotline: 19009933**
24/7 support service

New products  [View all](#)

 SHIRTS Crop top for women 55.0\$	 SHIRTS Shirt Long Sleeves for men 35.0\$	 SHIRTS Tanktop for men 60.0\$	 SHIRTS T-Shirt for men 50.0\$
 SHIRTS Jacket for women 15.0\$ 30.0\$ 50%	 SHIRTS T-Shirt for men 25.0\$ 50.0\$ 50%	 SHIRTS T-Shirt for men 25.0\$ 50.0\$ 50%	 SHIRTS Tanktop for men 30.0\$ 60.0\$ 50%

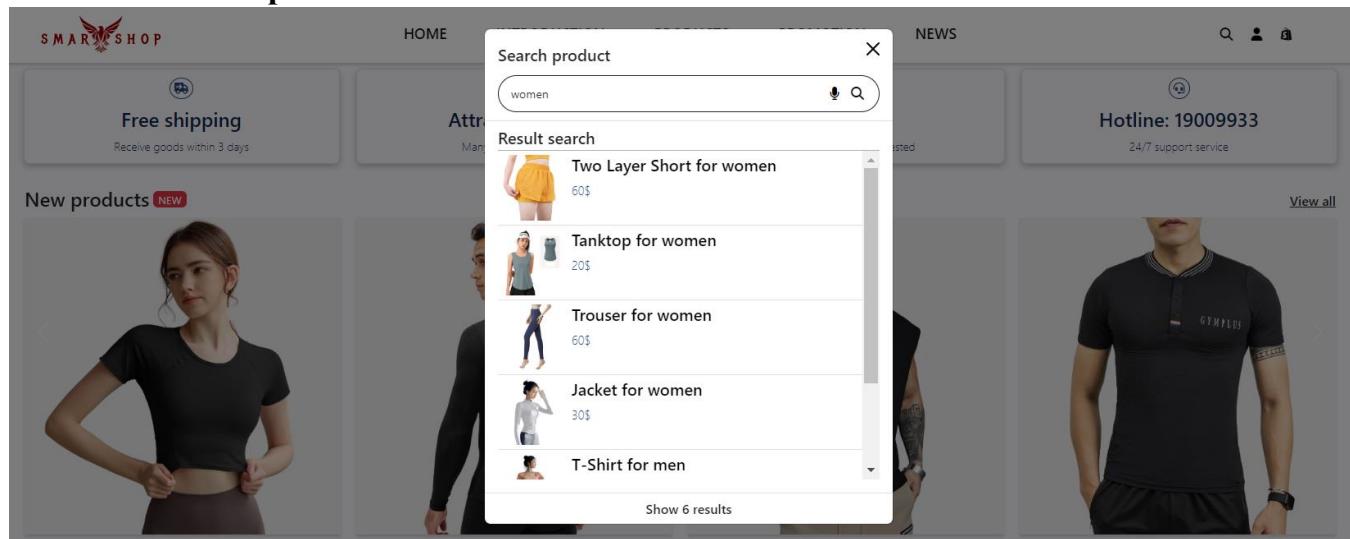
Bestseller  [View all](#)

 SHIRTS T-Shirt Zipp for men 30.0\$	 SHIRTS Tanktop for women 20.0\$	 SHIRTS Jacket for women 30.0\$	 SHIRTS T-Shirt for men 50.0\$
--	---	---	---



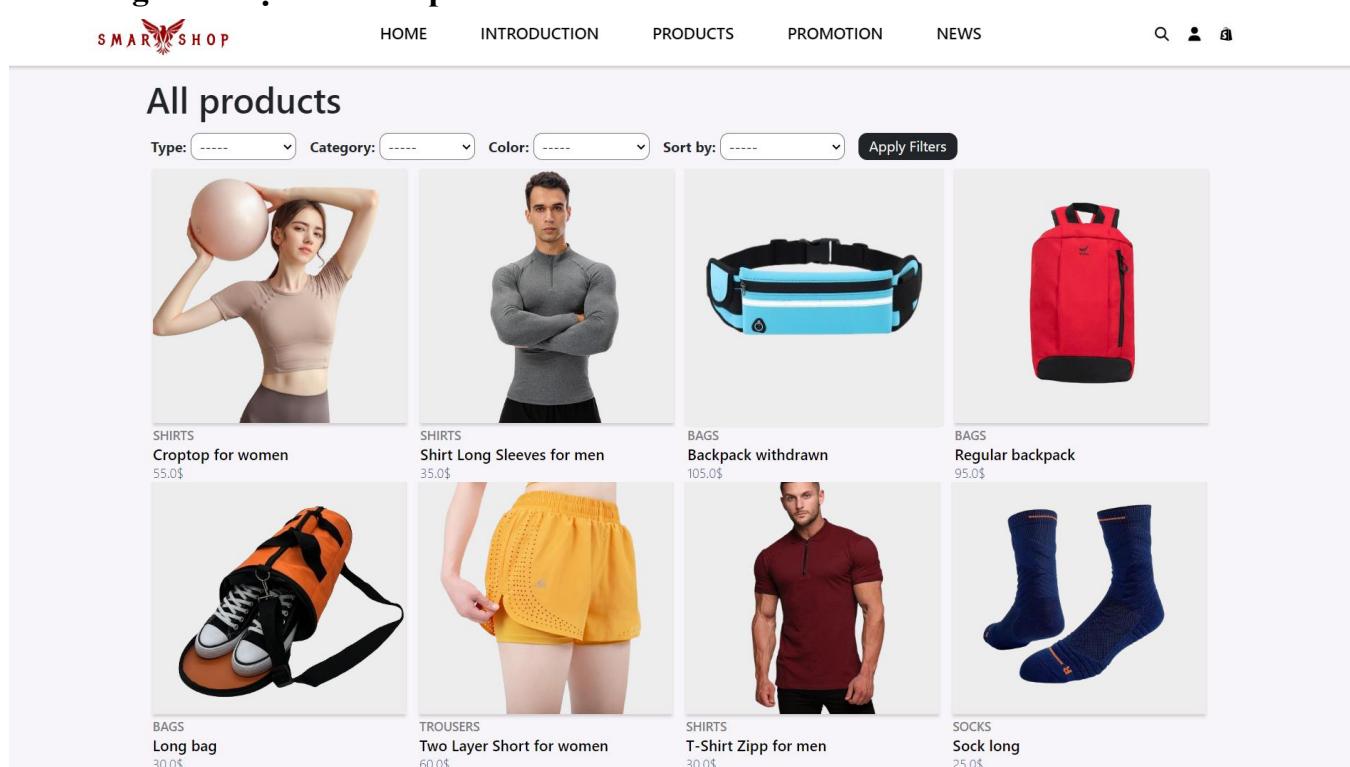
Hình 4.2.1. Trang chủ

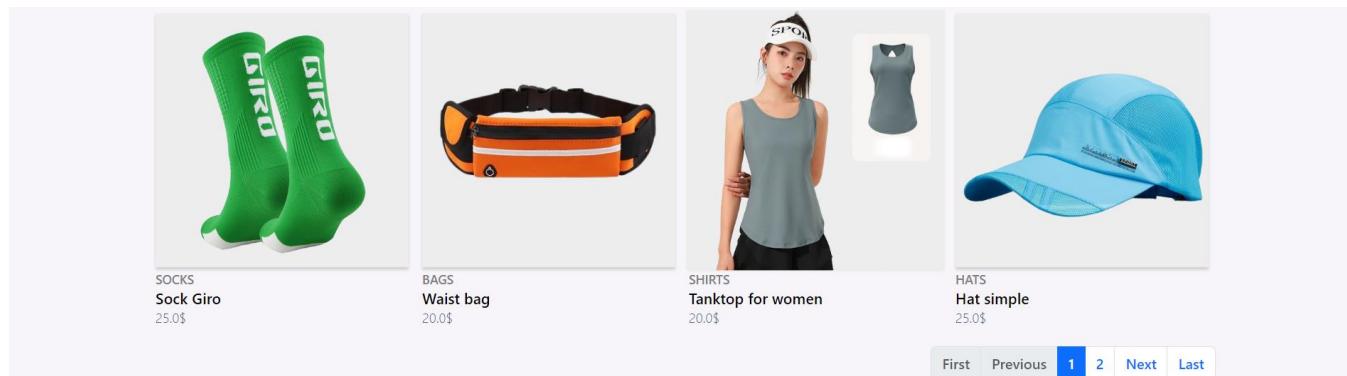
Tìm kiếm sản phẩm



Hình 4.2.2. Tìm kiếm sản phẩm

Trang hiển thị tất cả sản phẩm





Hình 4.2.3. Trang hiển thị sản phẩm

All products

Type: ----- Category: Shirts Color: Dark Sort by: Name A → Z Apply Filters

The screenshot shows a grid of four products after applying filters. From left to right: a man in a black zip-up shirt labeled 'SHIRTS T-Shirt Zipp for men 30.0\$'; a woman in a black tank top labeled 'SHIRTS T-Shirt for men 50.0\$'; a man in a black t-shirt labeled 'SHIRTS T-Shirt for men 50.0\$'; and a woman in a black tank top labeled 'SHIRTS Tanktop for women 20.0\$'.

Hình 4.2.4. Trang hiển thị sản phẩm sau khi áp dụng bộ lọc

Chi tiết sản phẩm

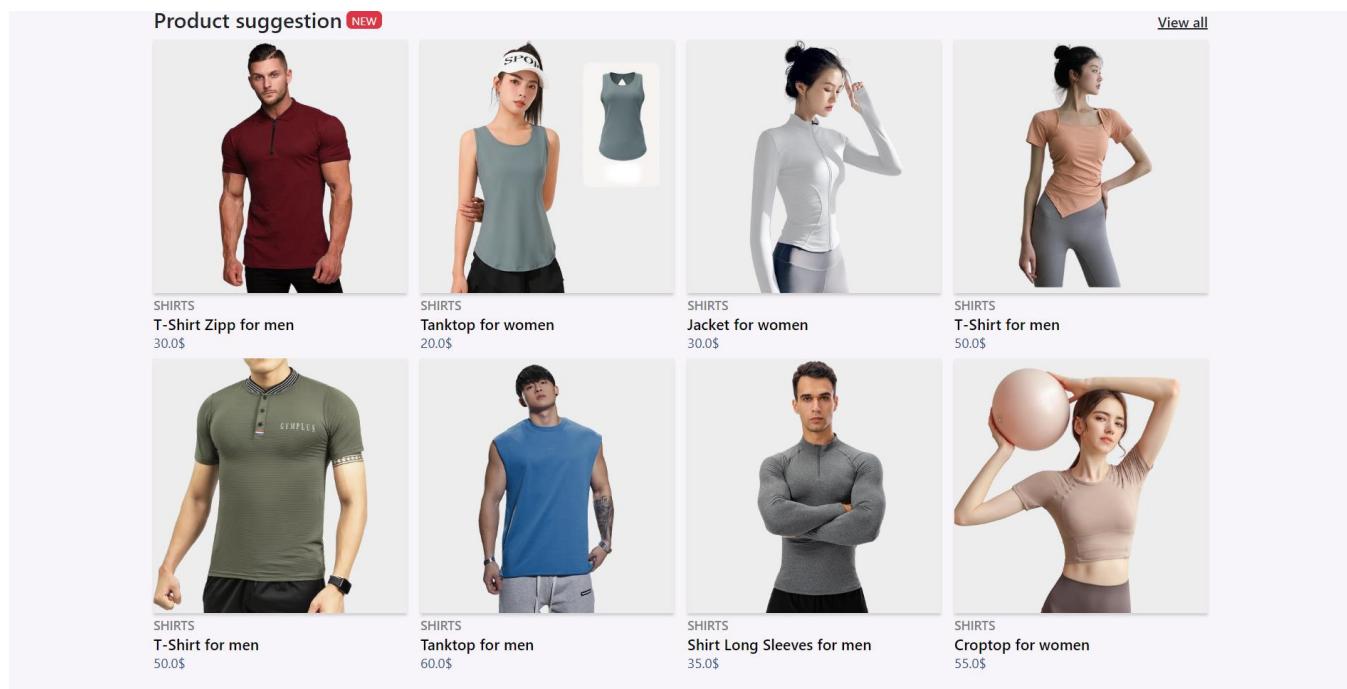
Tower ABC, Thu Duc, Ho Chi Minh | ntfahion@gmail.com | 19009933 | [Twitter](#) [Facebook](#) [Instagram](#) [YouTube](#)

HOME INTRODUCTION PRODUCTS PROMOTION NEWS

Home / All products / Croptop for women

The screenshot shows a product detail page for a 'Croptop for women'. It features a large image of a woman wearing the black crop top, a title 'Croptop for women', a price of '50.0\$', a discount of '9%', and a '9%' badge. It also shows discount codes for 'Free shipping', 'Sale 50%', 'Sale 10%', and 'Sale 25%'. Below the image are dropdown menus for 'Color' (Pink, Dark) and 'Size' (S). A quantity input field shows '3' and an 'Add to cart' button. At the bottom, there is a 'Description' section with text about the iMac M1.

Hình 4.2.5. Trang chi tiết sản phẩm (phần chi tiết)



Hình 4.2.6. Trang chi tiết sản phẩm (phần sản phẩm đề xuất)

4.3. Quy trình đặt hàng và thanh toán

Giỏ hàng

SmartShop

HOME INTRODUCTION PRODUCTS PROMOTION NEWS

Home / Shopping Cart

Shopping cart

IMAGE	PRODUCTS	COLOR/SIZE	PRICE	QUANTITY	UPDATE	TOTAL	SALE PRICE
	Hat sport	Pink/Freesize	30.0	1	Update	30.0	15.0
	Two Layer Short for women	Yellow/S	60.0	1	Update	60.0	50.0
	Crop top for women	Dark/S	55.0	1	Update	55.0	50.0

Total item: 3
 Subtotal : 145.0\$
 Discount : -30.0\$
Total 115.0\$

Notes

Hình 4.3.1. Trang giỏ hàng

Đặt hàng

Tower ABC, Thu Duc, Ho Chi Minh
ntfahion@gmail.com
19009933

Home / Shopping cart / Checkout

Billing address		Your order			
Name*	tinh	Type	Color/Size	Quantity	Cost Price
Phone*	0896414655	Two Layer Short for women	Yellow/S	1	60.0\$ 50.0\$
Email *	phanhuutinh@gmail.com	Hat sport	Pink/Freesize	1	30.0\$ 15.0\$
Address*	An Lac A	Croptop for women	Dark/S	1	55.0\$ 50.0\$
Payment method*	<input type="radio"/> Cash <input checked="" type="radio"/> Credit card	Provisional			115.0
		Transport			5\$
		Grand total			120.0\$

Back
Order

Hình 4.3.2. Trang đơn hàng

Trang thanh toán VNPAY

Giao dịch hết hạn sau **14 : 44**

Thông tin đơn hàng (Test)

Số tiền thanh toán
10.000VND

Giá trị đơn hàng
10.000VND

Phí giao dịch
0VND

Mã đơn hàng
49365242

Nhà cung cấp
Công ty CTT HTT1 (Test)

Thanh toán qua Ngân hàng NCB

Thẻ nội địa

Số thẻ

Tên chủ thẻ

Ngày phát hành (MM/YY)

Mã khuyến mại

[Điều kiện sử dụng dịch vụ](#)

Hủy thanh toán
Tiếp tục

Hình 4.3.3. Trang thanh toán VNPAY

Trang đặt hàng thành công

The screenshot shows a confirmation message: "Thank you for your order" with a checkmark icon. It includes a note: "A confirmation email has been sent to john.doe@example.com. Please check your email." Below this is a "Purchase information" section containing the following details:

- Username:** john_doe
- Email:** john.doe@example.com
- Phone:** 12345678901
- Address:** 123 Maple Street, Springfield, IL
- Payment method:** Payment VNPay

To the right is a table titled "Order" showing the items purchased:

Image	Name	Color/Size	Quantity	Price
	Sock long	White/S	2	40.0\$
	Croptop for women	Pink/S	1	50.0\$
	Shirt Long Sleeves for men	Dark/S	1	30.0\$
Provisional				225.0\$
Transport				5.0\$
Total				230.0\$

A "Continue shopping" button is located at the bottom.

Hình 4.3.4. Trang thanh toán thành công

4.4. Quy trình quản lý cửa hàng

Trang chủ cửa hàng (quản trị viên)

The dashboard features a top navigation bar with icons for Home, Search, Notifications, and User (JOHN_DOE). On the left is a sidebar with links for Dashboard, E-commerce, Pages, and a user profile for JOHN_DOE.

The main area is divided into several sections:

- E-commerce Dashboard:** Includes four cards: New products (21), New members (21), New sales (7), and Highest access (1000).
- Lasted reviews:** A table showing recent reviews for the Fitbit Sense Advanced Smartwatch. All reviews are 5 stars from Richard Dawkins, posted "Just now".
- Revenue:** A donut chart titled "World Wide Wine Production 2018" showing proportions for Shoes (teal), Shirts (blue), and Bags (purple).
- Traffic:** A line graph showing Organic and Paid traffic over time from January to July. Organic traffic peaks in April, while Paid traffic shows more fluctuation.
- Traffic:** A line graph showing Organic and Paid traffic over time from January to July. Organic traffic peaks in April, while Paid traffic shows more fluctuation.

Hình 4.4.1. Trang chủ cửa hàng

Thêm sản phẩm (quản trị viên)

E commerce / Admin / Add product

Add a product

Date: 07/16/2024 11:39 AM

Product Title: Bra fore women

Product Description: Good Bra for women

Regular price: 90

Sale price: 45

Category: Shirts

Select Files to upload image

Choose Files [2 files]

Color: Dark... Sizes: S 4 M SL L 12 XL SL XXL 5 FreeSize SL

Color: Dark... Sizes: S SL M 3 L SL XL 3 XXL SL FreeSize SL

Hình 4.4.2. Trang thêm sản phẩm sản phẩm

Quản lý sản phẩm (quản trị viên)

E commerce / Admin / Products

Products

ID	NAME	CATEGORY	PRICE	SALE PRICE	DISCOUNT%	DESCRIPTION	DATE	Actions
18	Backpack withdrawn	Bags	105.0\$	90.0\$	14%	Description of Backpack withdrawn	2024-07-10T16:07:20.992898	
19	Regular backpack	Bags	95.0\$	80.0\$	16%	Description of Regular backpack	2024-07-10T16:07:20.987979	
20	Two Layer Short for women	Trousers	60.0\$	50.0\$	17%	Description of Two Layer Short for women	2024-07-10T16:07:20.993437	
1	T-Shirt Zipp for men	Shirts	30.0\$	25.0\$	17%	Description of T-Shirt for men	2024-07-10T16:07:20.919172	
2	Tanktop for women	Shirts	20.0\$	15.0\$	25%	Description of Tanktop for women	2024-07-10T16:07:20.928985	

First Previous 1 2 3 4 5 Next Last

Hình 4.4.3. Trang quản lý sản phẩm

Quản lý tài khoản (quản trị viên)

E commerce / Admin / Members

Members

ID	IMAGE	NAME	EMAIL	PHONE	ADDRESS	Customer
21		tinh	phanhuutinh@gmail.com	0896414655	An Lac A	
20		rachel_garcia	rachel.garcia@example.com	12345678920	1717 Elm Boulevard, Minneapolis, MN	
19		quincy_ford	quincy.ford@example.com	12345678919	1616 Birch Court, Nashville, TN	
18		peter_evans	peter.evans@example.com	12345678918	1515 Oak Street, New Orleans, LA	
17		olivia_davis	olivia.davis@example.com	12345678917	1414 Maple Avenue, Tampa, FL	
16		nate_clark	nate.clark@example.com	12345678916	1313 Spruce Road, Atlanta, GA	
15		molly_baker	molly.baker@example.com	12345678915	1212 Oak Lane, Charlotte, NC	
14		leo_scott	leo.scott@example.com	12345678914	1111 Pine Avenue, Columbus, OH	
13		karen_moore	karen.moore@example.com	12345678913	1010 Cedar Street, Detroit, MI	
12		judy_hall	judy.hall@example.com	12345678912	909 Birch Lane, Baltimore, MD	

Hình 4.4.4. Trang quản lý tài khoản (danh sách khách hàng)

**FACULTY OF MATHEMATICS AND COMPUTER SCIENCE
UNIVERSITY OF SCIENCE, HO CHI MINH CITY, VIET NAM**

ID	IMAGE	NAME	EMAIL	PHONE	ADDRESS
3		alice_jones	alice.jones@example.com	12345678903	789 Pine Road, Albany, NY
2		jane_smith	jane.smith@example.com	12345678902	456 Oak Avenue, Riverside, CA
1		john_doe	john.doe@example.com	12345678901	123 Maple Street, Springfield, IL

Hình 4.4.5. Trang quản lý tài khoản (danh sách quản trị viên)

ORDER	CUSTOMER	TOTAL	PAYMENT STATUS	DELIVERY METHOD	DATE
#1	john_doe	\$55.0	Pending	Cash	2024-07-15 15:54:08.146
#2	john_doe	\$95.0	Pending	Credit card	2024-07-15 15:58:13.946
#3	tinh	\$120.0	Pending	Credit card	2024-07-16 11:24:27.975

Hình 4.4.6. Trang quản lý đơn hàng

IMAGE	PRODUCTS	COLOR/SIZE	PRICE	QUANTITY	TOTAL	SALE PRICE
	Two Layer Short for women	Yellow/M	60.0	3	180.0	5.0
	T-Shirt Zipp for men	Dark/M	30.0	2	60.0	20.0
	T-Shirt Zipp for men	Gray/L	30.0	4	120.0	20.0
	Trouser for women	Blue/M	60.0	2	120.0	45.0

Total item:	4
Subtotal :	480.0\$
Discount :	90.0\$
Transport price:	5.0\$
Total	95.0\$

Hình 4.4.7. Trang quản lý đơn hàng chi tiết

4.5. Quá trình kiểm thử

Postman

**FACULTY OF MATHEMATICS AND COMPUTER SCIENCE
UNIVERSITY OF SCIENCE, HO CHI MINH CITY, VIET NAM**

```

1 [
2   {
3     "productId": 1,
4     "productName": "T-Shirt Zipp for men",
5     "description": "Description of T-Shirt for men",
6     "costPrice": 30.0,
7     "salePrice": 20.0,
8     "images": [
9       {
10        "imageId": 1,
11        "imageName": "T-Shirt-with-zipper-men-red.jfif",
12        "color": {
13          "colorId": 1,
14          "colorName": "Red"
15        },
16        "imageSizeQuantities": [
17          {
18            "size": {
19              "sizeName": "M",
20              "currentQuantity": 15
21            },
22            {
23              "sizeName": "L",
24              "currentQuantity": 20
25            },
26            {
27              "sizeName": "XL",
28              "currentQuantity": 36
29            },
30            {
31              "sizeName": "XXL"
32            }
33          ]
34        }
35      }
36    ]
37  }
38]
  
```

Hình 4.5.1. Api thêm danh sách sản phẩm

```

1 Delete product successful
  
```

Hình 4.5.2. Api xóa sản phẩm

```

1 [
2   {
3     "productId": 10,
4     "productName": "T-Shirt for men",
5     "description": "Description of T-Shirt for men",
6     "costPrice": 50.0,
7     "salePrice": 10.0,
8     "createdAt": "2024-07-10T08:40:55.766903",
9     "images": [
10    ],
11     "category": {
12       "categoryId": 1,
13       "categoryName": "Shirts"
14     },
15     "cartItems": []
16   },
17   {
18     "productId": 11,
19     "productName": "Trouser for men",
20   }
21 ]
  
```

Hình 4.5.3. Api lấy tất cả sản phẩm

**FACULTY OF MATHEMATICS AND COMPUTER SCIENCE
UNIVERSITY OF SCIENCE, HO CHI MINH CITY, VIET NAM**

HTTP MyProject / RestController / GET /getProduct

GET http://localhost:8095/getProduct?productId=10

Params

Key	Value	Description
productId	10	
Key	Value	Description

Body

Pretty

```

1
2     "productId": 10,
3     "productName": "T-Shirt for men",
4     "description": "Description of T-Shirt for men",
5     "costPrice": 50.0,
6     "salePrice": 10.0,
7     "createdAt": "2024-07-10T08:40:55.766903",
8     "images": [
...

```

Raw

Preview

Visualize

JSON

200 OK 58 ms 2.4 KB

Save as example

Send

Hình 4.5.4. Api lấy sản phẩm cụ thể

HTTP MyProject / RestController / PUT /updateProductR/10

PUT http://localhost:8095/updateProductR/10

Body

raw

```

1 {
2   "productId": 10,
3   "productName": "T-Shirt for men",
4   "description": "Description of T-Shirt for men",
5   "costPrice": 900.0,
6   "salePrice": 300.0
}

```

JSON

Beautify

Pretty

Raw

Preview

Visualize

Text

200 OK 114 ms 448 B

Save as example

Send

Hình 4.5.5. Api cập nhật sản phẩm

HTTP MyProject / RestController / GET /getProduct

GET http://localhost:8095/getProduct?productId=10

Params

Key	Value	Description
productId	10	
Key	Value	Description

Body

Pretty

```

1
2     "productId": 10,
3     "productName": "T-Shirt for men",
4     "description": "Description of T-Shirt for men",
5     "costPrice": 900.0,
6     "salePrice": 300.0,

```

Raw

Preview

Visualize

JSON

200 OK 56 ms 599 B

Save as example

Send

Hình 4.5.6. Kết quả cập nhật sản phẩm với id = 10

JMeter

Thread Group

Name: homepage

Comments:

Action to be taken after a Sampler error

Continue Start Next Thread Loop Stop Thread Stop Test Stop Test Now

Thread Properties

Number of Threads (users): 600

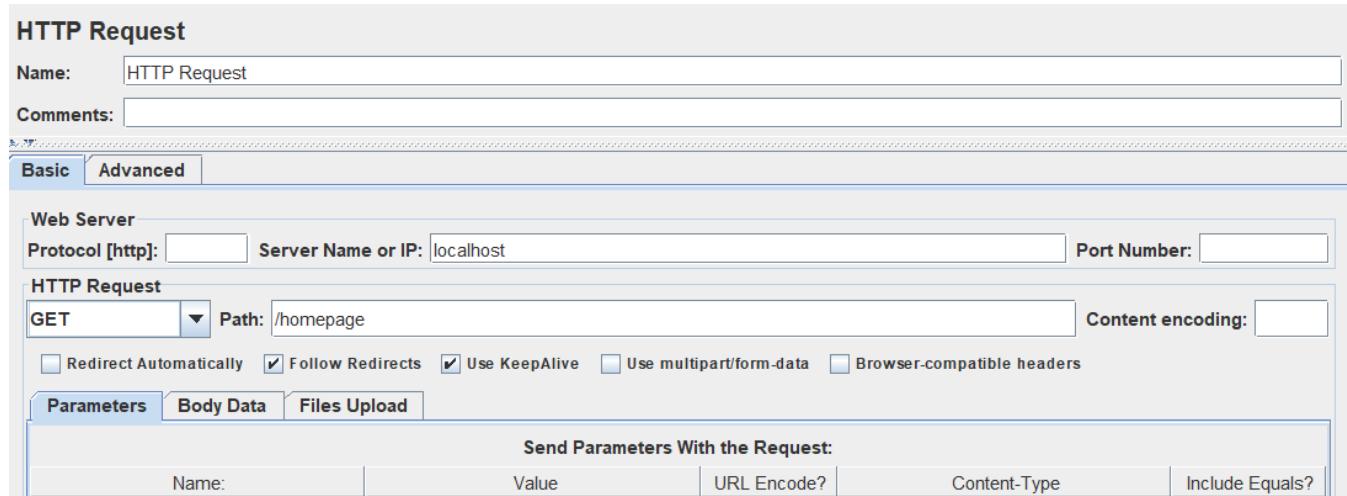
Ramp-up period (seconds): 0

Loop Count: Infinite 1

Hình 4.5.7. Tạo Thread Group

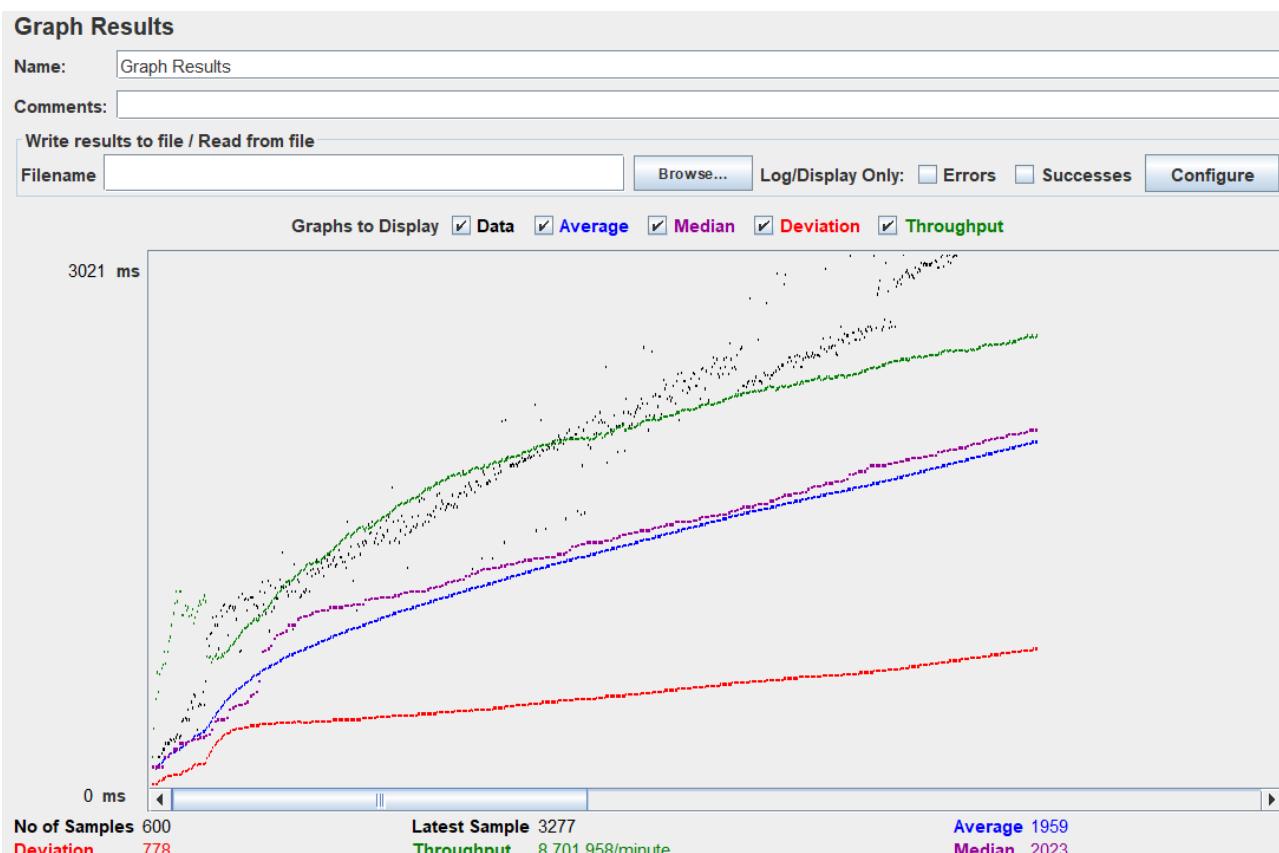
Trong đó (Hình 4.5.7):

- **Number of Threads** là số lượng người dùng ảo (threads) sẽ được JMeter mô phỏng để gửi yêu cầu đến server.
- **Ramp-up** là thời gian (tính bằng giây) để khởi động toàn bộ số lượng người dùng ảo. JMeter sẽ khởi động các threads theo cách phân bổ đều trong khoảng thời gian này.
- **Loop count** là số lần lặp lại của mỗi thread để thực hiện kịch bản kiểm thử.



Hình 4.5.8. Truy cập đến trang chủ

Hình 4.5.9. Kết quả đạt được với View Results Tree



Hình 4.5.10. Kết quả đạt được với Graph Results

Trong đó (Hình 4.5.10):

- **Deviation (độ lệch chuẩn):** là một thước đo sự phân tán của thời gian phản hồi so với giá trị trung bình.
- **Throughput (requests/minute):** là số lượng yêu cầu mà hệ thống có thể xử lý trong một khoảng thời gian cụ thể, thường được tính bằng số lượng yêu cầu mỗi giây hoặc mỗi phút.

Tổng kết:

Với kịch bản kiểm thử này, chúng tôi giả định sẽ tạo ra 600 người dùng ảo và thực hiện đồng thời gửi 600 yêu cầu cùng lúc đến server. Kết quả cho thấy:

- **Deviation là 778 ms.** Mức độ dao động này khá nhỏ, chỉ ra rằng hệ thống của chúng tôi có thể cung cấp thời gian phản hồi khá ổn định cho hầu hết các yêu cầu.
- **Throughput là 8,701.958 requests/minute,** điều này cho thấy hệ thống của chúng tôi có khả năng xử lý gần 8700 yêu cầu mỗi phút. Đây là một thông số rất tốt, chứng tỏ hệ thống có khả năng chịu tải cao và xử lý một lượng lớn yêu cầu mà không gặp vấn đề lớn.

Kết luận: Thấy được khả năng xử lý đồng thời hoạt động ổn định và hiệu quả, giúp đảm bảo trải nghiệm người dùng mượt mà và không bị gián đoạn ngay cả khi có nhiều người truy cập cùng một lúc.

CHƯƠNG 5: TỔNG KẾT VÀ PHÁT TRIỂN ĐỀ TÀI

5.1. Những đóng góp của đề tài

Sau bao nhiêu lâu tìm tòi và không ngừng học hỏi nhiều kiến thức mới. Đến nay chúng em cũng đã hoàn thành một website khá đầy đủ các tính năng của một trang web TMĐT. Trong quá trình này giúp tôi đã nắm vững các công nghệ và áp dụng chúng vào thực tế một cách hiệu quả.

Kết quả thực tế đã đạt được:

- Xây dựng thành công một ứng dụng full-stack cho website TMĐT sử dụng Spring Boot, PostgreSQL, Bootstrap, và Thymeleaf. Dự án này không chỉ đáp ứng nhu cầu cơ bản của một trang web bán hàng trực tuyến mà còn mang đến giao diện thân thiện, trải nghiệm người dùng mượt mà và trực quan.
- Nắm vững các khía cạnh quan trọng của lĩnh vực TMĐT, đặc biệt là mô hình kinh doanh B2C (Business-to-Consumer).
- Nghiên cứu và triển khai các cơ chế bảo mật mạnh mẽ, bảo vệ dữ liệu và quyền truy cập của người dùng thông qua Spring Security.
- Khai thác sức mạnh của Docker để chuẩn hóa môi trường phát triển, đóng gói ứng dụng và triển khai nhanh chóng.
- Sử dụng Nginx như một Reverse proxy để cân bằng tải và cải thiện khả năng chịu tải của hệ thống, đảm bảo ứng dụng hoạt động mượt mà ngay cả khi lưu lượng truy cập cao.
- Đặt chỉ mục và sử dụng các thuộc tính unique của RDBMS để tối ưu hóa hiệu suất truy vấn và quản lý dữ liệu hiệu quả.
- Sử dụng Cache để lưu trữ dữ liệu tạm thời, giúp tăng tốc độ phản hồi, tiết kiệm băng thông và tài nguyên hệ thống.
- Nghiên cứu và tích hợp thành công phương thức thanh toán trực tuyến VNPay, mang đến sự tiện lợi cho khách hàng trong quá trình mua sắm.
- Ứng dụng các công cụ kiểm thử API như Postman và JMeter để đảm bảo chất lượng và hiệu suất của hệ thống.
- Hoàn thiện quá trình triển khai ứng dụng trên VPS, đảm bảo website hoạt động ổn định và hiệu quả trên môi trường internet.
- Url website: <http://ntfashion.click/>

5.2. Hướng phát triển

- Chuyển đổi từ kiến trúc Monolithic sang Microservices
- Tự động điều chỉnh giao diện dựa trên sở thích và thói quen người dùng
- Hỗ trợ đa ngôn ngữ và đa tiền tệ
- Tối ưu chuẩn SEO, marketing và quảng cáo
- Quản lý kho hàng và logistics
- Cung cấp chức năng gửi email đánh giá và nhận xét sản phẩm

- Tích hợp JWT và Redis, Kafka
- Tăng khả năng bảo mật với chứng chỉ SSL
- Đăng nhập bằng tài khoản mạng xã hội và nhúng chat box

TÀI LIỆU THAM KHẢO

- [1] Vishal Shah. “How to do Distributed Load Testing using Apache jMeter - TatvaSoft Blog”. *TatvaSoft Blog*. 2016. <https://www.tatvasoft.com/blog/how-to-do-distributed-load-testing-using-apache-jmeter/>
- [2] Devteria. “Spring boot 3Microservices”. www.youtube.com/@devteriadotcom
<https://www.youtube.com/watch?v=X5UxbyPT9zg>
- [3] Priya Pedamkar. “Docker Architecture | Learn the Objects and Benefits of Docker”. *EDUCBA*. 2019. <https://www.educba.com/docker-architecture/>
- [4] VNPay. “Kết nối Cổng thanh toán VNPay · Cổng thanh toán VNPay”. 2022. <https://sandbox.vnpayment.vn/apis/docs/thanh-toan-token/token.html>
- [5] Spring. “Form Login: Spring Security”. 2005 - 2024.
<https://docs.spring.io/springsecurity/reference/servlet/authentication/passwords/form.html>
- [6] Thinhnotes. “Use Case Diagram và 5 sai lầm thường gặp - Thinhnotes”. Thinhnotes.com. 2019. <https://thinhnotes.com/chuyen-nghe-ba/use-case-diagram-va-5-sai-lam-thuong-gap/>
- [7] Tpoint Tech. “IntelliJ IDEA Features - javatpoint”. [www.javatpoint.com](https://www.javatpoint.com/intellij-idea-feature). 2021.
<https://www.javatpoint.com/intellij-idea-feature>
- [8] Amigoscode. “How to get started with Spring Boot and Java”. www.youtube.com/@amigoscode. 2023.
<https://www.youtube.com/watch?v=9SGDpanrc8U&t=2422s>
- [9] Hỏi Dân IT. “Fullstack SERN (SQL, Express, React và Node.js)”. www.youtube.com/@hoidanit. 2022.
https://www.youtube.com/watch?v=VvvXhNbFWKY&list=PLncHg6Kn2JT6E38Z3kit9Hnif1xC_9VqI