

NGUYỄN HẢI DĂNG

KHOA HỌC MÁY TÍNH

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

-----***-----

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC
NGÀNH KHOA HỌC MÁY TÍNH

ĐỀ TÀI: XÂY DỰNG WEBSITE THƯƠNG MẠI GIỚI
THIỆU VÀ BÁN SÁCH SỬ DỤNG NODEJS

Giảng viên hướng dẫn: ThS. Trần Thanh Huân

Sinh viên: Nguyễn Hải Đăng

Mã số sinh viên: 2021602703

Hà Nội - 2025

MỤC LỤC

MỤC LỤC.....	ii
LỜI CẢM ƠN	iv
DANH MỤC HÌNH ẢNH	v
DANH MỤC BẢNG BIỂU	viii
DANH MỤC CHỮ VIẾT TẮT.....	ix
MỞ ĐẦU.....	1
CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI.....	2
1.1. Lý do chọn đề tài	2
1.2. Mục tiêu của đề tài.....	2
1.3. Phương pháp nghiên cứu	3
1.4. Giới hạn và phạm vi của đề tài	3
1.5. Dữ liệu đầu vào và đầu ra của hệ thống	4
1.6. Ý nghĩa của đề tài	5
1.7. Kết quả dự kiến đạt được.....	5
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT	6
2.1. Phân tích, thiết kế và lập trình hướng đối tượng	6
2.1.1. Phân tích và thiết kế hướng đối tượng	6
2.1.2. Lập trình hướng đối tượng	7
2.2. Công nghệ phát triển ứng dụng website	8
2.2.1. Ngôn ngữ mô hình hóa UML.....	8
2.2.2. Ngôn ngữ thiết kế website.....	10
2.2.3. Mô hình Client – Server.....	13
2.2.4. NodeJS – Express.....	15
2.2.5. Hệ quản trị cơ sở dữ liệu	15
CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG	18
3.1. Khảo sát hệ thống	18
3.2. Mô tả bài toán	19

3.3. Xác định các yêu cầu của hệ thống.....	19
3.4. Thiết kế hệ thống	20
3.4.1. Các tác nhân hệ thống	20
3.4.2. Các use case trong hệ thống	21
3.4.3. Biểu đồ lớp	26
3.4.4. Biểu đồ hoạt động	27
3.4.5. Biểu đồ trình tự.....	41
3.4.6. Cơ sở dữ liệu	71
3.4.7. Mô hình thực thể - quan hệ	77
CHƯƠNG 4. CÀI ĐẶT VỀ KIỂM THỬ HỆ THỐNG	78
4.1. Cài đặt môi trường.....	78
4.1.2. Môi trường hoạt động.....	78
4.1.3. Cài đặt môi trường phát triển	78
4.2. Một số giao diện hệ thống	79
4.2.1. Giao diện khách hàng.....	79
4.2.2. Giao diện nhân viên quản lý.....	81
4.2.3. Giao diện nhân viên kiểm kho	84
4.3. Kiểm thử hệ thống	85
4.3.1. Khái niệm và đặc điểm của kiểm thử hệ thống.....	85
4.3.2. Các lĩnh vực của Kiểm thử hệ thống.....	87
4.3.3. Quá trình kiểm thử	87
4.3.4. Lý do thực hiện kiểm thử hệ thống	88
4.4. Thiết kế các test case cho hệ thống.....	89
KẾT LUẬN	93
TÀI LIỆU THAM KHẢO.....	94

LỜI CẢM ƠN

Để hoàn thành được đề tài đồ án tốt nghiệp này, trước hết em gửi lời cảm ơn chân thành nhất đến các thầy cô giáo tại Đại học Công nghiệp Hà Nội nói chung, các thầy cô trong trường Công Nghệ Thông Tin và Truyền Thông nói riêng đã giảng dạy cho em kiến thức về các môn đại cương cũng như các môn chuyên ngành, giúp em có được cơ sở lý thuyết vững vàng và tạo điều kiện giúp đỡ em trong suốt quá trình học tập để em có được ngày hôm nay.

Đặc biệt, em xin gửi lời cảm ơn chân thành về sự chỉ dạy, hướng dẫn tận tình của thầy Trần Thanh Huân trong suốt thời gian thực hiện đồ án. Mặc dù trong quá trình làm đồ án em đã rất cố gắng tuy nhiên cũng không tránh khỏi những thiếu sót. Em hy vọng rằng là sẽ nhận được những ý kiến nhận xét, góp ý của các thầy cô giáo về những vấn đề được triển khai trong bài báo cáo đồ án tốt nghiệp của em được hoàn thiện và giúp em có thêm những kinh nghiệm quý báu.

Cuối cùng, em xin kính chúc các thầy cô giảng viên tại Đại học Công Nghiệp Hà Nội nói chung, các thầy cô trường Công Nghệ Thông Tin và Truyền Thông nói riêng luôn luôn dồi dào sức khỏe và thành công trong sự nghiệp giảng dạy cao quý.

Em xin chân thành cảm ơn!

Sinh viên thực hiện

Đặng

Nguyễn Hải Đăng

DANH MỤC HÌNH ẢNH

Hình 2.1	Trực quan mô hình Client - Server.....	14
Hình 3.1	Sơ đồ use case tổng quan hệ thống	21
Hình 3.2	Sơ đồ use case Khách hàng – Khách hàng Tiềm năng	22
Hình 3.3	Sơ đồ use case Nhân viên quản lý và Kiểm kho.....	24
Hình 3.4	Biểu đồ lớp hệ thống	26
Hình 3.5	Biểu đồ hoạt động use case Đăng nhập	27
Hình 3.6	Biểu đồ hoạt động use case Đăng ký	28
Hình 3.7	Biểu đồ hoạt động use case Thêm sản phẩm vào giỏ hàng.....	29
Hình 3.8	Biểu đồ hoạt động use case Quản lý giỏ hàng	30
Hình 3.9	Biểu đồ hoạt động use case Đặt hàng	31
Hình 3.10	Biểu đồ hoạt động use case Quản lý đơn hàng cá nhân.....	32
Hình 3.11	Biểu đồ hoạt động use case Quản lý tài khoản	33
Hình 3.12	Biểu đồ hoạt động use case Quản lý sản phẩm.....	34
Hình 3.13	Biểu đồ hoạt động use case Quản lý đơn hàng	35
Hình 3.14	Biểu đồ hoạt động use case Quản lý kho hàng	36
Hình 3.15	Biểu đồ hoạt động use case Quản lý Blog	37
Hình 3.16	Biểu đồ hoạt động use case Quản lý thể loại	38
Hình 3.17	Biểu đồ hoạt động use case Quản lý video	39
Hình 3.18	Biểu đồ hoạt động use case Thống kê.....	40
Hình 3.19	Biểu đồ trình tự use case Đăng nhập.....	41
Hình 3.20	Biểu đồ trình tự use case Đăng ký	42
Hình 3.21	Biểu đồ trình tự use case Đăng xuất	43
Hình 3.22	Biểu đồ trình tự use case Tìm kiếm sản phẩm.....	44
Hình 3.23	Biểu đồ trình tự use case Xem chi tiết sản phẩm.....	45
Hình 3.24	Biểu đồ trình tự use case Xem sản phẩm theo tiêu chí.....	46
Hình 3.25	Biểu đồ trình tự use case Quản lý giỏ hàng (1).....	47

Hình 3.26 Biểu đồ trình tự use case Quản lý giỏ hàng (2).....	48
Hình 3.27 Biểu đồ trình tự use case Quản lý thông tin cá nhân	49
Hình 3.28 Biểu đồ trình tự use case Quản lý đơn hàng cá nhân (1)	50
Hình 3.29 Biểu đồ trình tự use case Quản lý đơn hàng cá nhân (2)	51
Hình 3.30 Biểu đồ trình tự use case Đánh giá sản phẩm	52
Hình 3.31 Biểu đồ trình tự use case Đặt hàng (1).....	53
Hình 3.32 Biểu đồ trình tự use case Đặt hàng (2).....	54
Hình 3.33 Biểu đồ trình tự use case Quản lý tài khoản (1).....	55
Hình 3.34 Biểu đồ trình tự use case Quản lý tài khoản (2).....	56
Hình 3.35 Biểu đồ trình tự use case Quản lý sản phẩm (1)	57
Hình 3.36 Biểu đồ trình tự use case Quản lý sản phẩm (2)	58
Hình 3.37 Biểu đồ trình tự use case Quản lý đơn hàng (1).....	59
Hình 3.38 Biểu đồ trình tự use case Quản lý đơn hàng (2).....	60
Hình 3.39 Biểu đồ trình tự use case Quản lý thể loại (1).....	61
Hình 3.40 Biểu đồ trình tự use case Quản lý thể loại (2).....	62
Hình 3.41 Biểu đồ trình tự use case Quản lý kho hàng (1).....	63
Hình 3.42 Biểu đồ trình tự use case Quản lý kho hàng (2).....	64
Hình 3.43 Biểu đồ trình tự use case Quản lý video (1).....	65
Hình 3. 44 Biểu đồ trình tự use case Quản lý video (2).....	66
Hình 3.45 Biểu đồ trình tự use case Quản lý blog (1)	67
Hình 3.46 Biểu đồ trình tự use case Quản lý blog (2)	68
Hình 3.47 Biểu đồ trình tự use case Thống kê (1)	69
Hình 3.48 Biểu đồ trình tự use case Thống kê (2)	70
Hình 3.49 Mô hình thực thể liên kết hệ thống	77
Hình 4.1 Giao diện trang chủ Khách hàng.....	79
Hình 4.2 Giao diện trang chi tiết sản phẩm Khách hàng	79
Hình 4.3 Giao diện danh sách đơn hàng Khách hàng.....	80

Hình 4.4 Giao diện đăng nhập Khách hàng	80
Hình 4.5 Giao diện chi tiết bài Blog	81
Hình 4.6 Giao diện trang chủ quản lý	81
Hình 4.7 Giao diện trang quản lý sản phẩm.....	82
Hình 4.8 Giao diện trang quản lý đơn hàng	82
Hình 4.9 Giao diện trang quản lý tài khoản	83
Hình 4.10 Giao diện trang quản lý thống kê.....	83
Hình 4.11 Giao diện trang quản lý kho hàng	84
Hình 4.12 Giao diện trang quản lý nhà cung cấp.....	84
Hình 4.13 Giao diện trang chi tiết nhà cung cấp.....	85

DANH MỤC BẢNG BIỂU

Bảng 3.1 Collection User	71
Bảng 3.2 Collection Category	71
Bảng 3.3 Collection Suppliers	72
Bảng 3.4 Collection Product	72
Bảng 3.5 Collection Reviews.....	73
Bảng 3.6 Collection ImportRecord	73
Bảng 3.7 Collection Order	73
Bảng 3.8 Collection OrderItem	75
Bảng 3.9 Collection OrderStatusHistoryItem	75
Bảng 3.10 Collection Video	75
Bảng 3.11 Collection Blog	76
Bảng 3.12 Collection ImportRecord	76
Bảng 4.1 Test case use case Đăng nhập	89
Bảng 4.2 Test case use case Đăng ký	90
Bảng 4.3 Test case use case Đăng xuất	90
Bảng 4.4 Test case use case Quản lý tài khoản	91
Bảng 4.5 Test case use case Quản lý sản phẩm	91
Bảng 4.6 Test case use case Quản lý đơn hàng	92

DANH MỤC CHỮ VIẾT TẮT

Chữ viết tắt	Ý nghĩa
DBMS	Database Management System
UML	Unified Modeling Language
OOAD	Object-Oriented Analysis and Design
OOP	Object-Oriented Programming
AJAX	Asynchronous JavaScript and XML
I/O	In / Out
NCC	Nhà cung cấp
SQL	Structured Query Language
noSQL	Not only Structured Query Language

MỞ ĐẦU

Trong thời đại công nghệ thông tin phát triển mạnh mẽ, việc ứng dụng công nghệ vào hoạt động kinh doanh và quản lý bán hàng đã trở thành xu hướng tất yếu. Một website bán sách trực tuyến không chỉ giúp các nhà sách mở rộng thị trường tiêu thụ mà còn mang lại sự tiện lợi cho người mua trong việc tìm kiếm và lựa chọn sản phẩm phù hợp. Đặc biệt, hệ thống này còn hỗ trợ quản lý kho sách, đơn hàng, khách hàng và các hoạt động bán hàng một cách hiệu quả, tiết kiệm thời gian và chi phí vận hành. Hiểu được nhu cầu thực tế và tiềm năng phát triển của mô hình kinh doanh này, em đã chọn đề tài **“Xây dựng website thương mại giới thiệu và bán sách sử dụng nodejs”** cho đề án tốt nghiệp của mình. Nội dung đề án gồm 4 chương:

Chương 1: Tổng quan về đề tài. Giới thiệu chung về đề tài: lý do chọn đề tài, mục tiêu nghiên cứu, giới hạn và phạm vi thực hiện, phương pháp nghiên cứu và kết quả dự kiến đạt được thông qua việc xây dựng hệ thống website bán sách.

Chương 2: Cơ sở lý thuyết. Trình bày các kiến thức nền tảng và công nghệ sẽ được áp dụng trong hệ thống như ngôn ngữ lập trình, framework, hệ quản trị cơ sở dữ liệu, cũng như các khái niệm liên quan đến thương mại điện tử và quản lý bán hàng.

Chương 3: Phân tích và thiết kế hệ thống. Tiến hành khảo sát, thu thập yêu cầu và phân tích hệ thống. Trình bày các sơ đồ chức năng như sơ đồ use case, biểu đồ tuần tự, sơ đồ hoạt động,... Đồng thời thiết kế cơ sở dữ liệu và kiến trúc tổng thể của hệ thống bán sách.

Chương 4: Cài đặt và kiểm thử hệ thống. Dựa trên các phân tích và thiết kế đã trình bày, tiến hành xây dựng và triển khai hệ thống website bán sách. Thực hiện kiểm thử các chức năng chính như tìm kiếm, đặt sách, quản lý giỏ hàng, quản trị sản phẩm,... để đánh giá hiệu quả và tính ổn định của hệ thống.

CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI

1.1. Lý do chọn đề tài

Thị trường thương mại điện tử đang bùng nổ mạnh mẽ, đặc biệt từ sau đại dịch COVID-19, thời điểm mà người tiêu dùng đã hình thành và củng cố thói quen mua sắm trên các nền tảng số. Trong bối cảnh đó, việc xây dựng website thương mại trực tuyến không chỉ trở thành xu hướng phổ biến mà còn là yếu tố sống còn đối với nhiều doanh nghiệp. Đối với thị trường sách, mặc dù nhu cầu đọc vẫn duy trì ổn định qua thời gian, phương thức tiếp cận đã có sự chuyển dịch đáng kể sang kênh số hóa, khi người đọc ngày càng ưa chuộng việc tìm kiếm thông tin, đọc đánh giá và thực hiện giao dịch mua sách thông qua các nền tảng trực tuyến trước khi đưa ra quyết định cuối cùng. Các hiệu sách truyền thống gặp khó khăn do không gian vật lý hạn chế và phạm vi địa lý nhỏ.

Ngược lại, website bán sách vượt qua rào cản địa lý, tiếp cận khách hàng toàn quốc, mở rộng thị trường tiềm năng và đưa sách đến vùng sâu vùng xa. Mô hình kinh doanh trực tuyến tiết kiệm chi phí vận hành, không cần mặt bằng lớn, giảm nhân viên bán hàng, và tối ưu quy trình từ nhập đến giao hàng. Hệ thống quản lý kho thông minh giúp theo dõi tồn kho, dự báo nhu cầu và lập kế hoạch nhập hàng hiệu quả.

Công nghệ web hiện đại cung cấp trải nghiệm mua sắm cá nhân hóa thông qua phân tích dữ liệu người dùng, đề xuất sách phù hợp, tăng khả năng chuyển đổi và giá trị đơn hàng. Xu hướng sử dụng thiết bị di động để mua sắm cũng đòi hỏi website responsive để tiếp cận đối tượng khách hàng trẻ.

1.2. Mục tiêu của đề tài

Mục tiêu chính của đề tài là xây dựng một nền tảng thương mại điện tử chuyên về sách với các mục tiêu cụ thể:

- + Thiết kế và phát triển website bán sách trực tuyến có giao diện thân thiện, dễ sử dụng trên cả máy tính và thiết bị di động.

- + Xây dựng hệ thống quản lý sản phẩm, quản lý đơn hàng và quản lý khách hàng hiệu quả.
- + Tích hợp các phương thức thanh toán an toàn.
- + Phát triển phương thức tìm kiếm và lọc sách thông minh giúp người dùng dễ dàng tìm kiếm sách theo nhiều tiêu chí.
- + Tích hợp hệ thống đánh giá và bình luận sản phẩm.
- + Đảm bảo tính bảo mật cho thông tin cá nhân và giao dịch của khách hàng.

1.3. Phương pháp nghiên cứu

Quá trình xây dựng website bán sách trực tuyến đòi hỏi một phương pháp nghiên cứu có hệ thống và khoa học để đảm bảo sản phẩm cuối cùng đáp ứng đúng nhu cầu của người dùng. Để thực hiện đề tài, các phương pháp nghiên cứu sẽ được áp dụng linh hoạt để thích ứng nhanh chóng với các thay đổi:

Nghiên cứu tổng quan: Thu thập thông tin về các website bán sách hiện có, phân tích ưu nhược điểm, tìm hiểu xu hướng thiết kế và công nghệ mới nhất.

Phân tích yêu cầu: Xác định rõ nhu cầu của người dùng và nhà quản lý hệ thống thông qua khảo sát, phỏng vấn.

Thiết kế mô hình: Sử dụng UML để thiết kế các sơ đồ use case, activity diagram, class diagram, sequence diagram.

Kiểm thử: Áp dụng kiểm thử đơn vị, kiểm thử tích hợp và kiểm thử hệ thống đảm bảo chất lượng phần mềm.

Đánh giá và cải tiến: Thu thập phản hồi từ người dùng để liên tục cải thiện sản phẩm.

1.4. Giới hạn và phạm vi của đề tài

Về thời gian, đề tài được thực hiện trong khoảng thời gian giới hạn của một đồ án tốt nghiệp (cụ thể là 9 tuần), do đó:

- Thời gian thực hiện: Từ tháng 17/03/2025 đến tháng 17/05/2025

- Tập trung vào các chức năng cốt lõi: Đăng ký/đăng nhập, quản lý sản phẩm (sách in), giỏ hàng, đặt hàng, thanh toán, và quản lý đơn hàng.
- Không triển khai các chức năng nâng cao: Như AI, hệ thống gợi ý thông minh, phân tích hành vi người dùng, hay hỗ trợ quốc tế.

Về không gian, đề tài chỉ tập trung xây dựng hệ thống trên nền tảng web. Phạm vi thị trường chỉ hỗ trợ thị trường nội địa Việt Nam, bao gồm:

- Hiện thị giao diện tiếng Việt.
- Hệ thống vận chuyển/giao hàng giới hạn trong phạm vi lãnh thổ Việt Nam.
- Không bao gồm: Bán sách điện tử (ebook) hay sách nói (audiobook); Giao dịch, vận chuyển hoặc thanh toán quốc tế; Tích hợp AI hoặc các công nghệ tiên tiến vượt ngoài khả năng thực hiện trong thời gian cho phép.

1.5. Dữ liệu đầu vào và đầu ra của hệ thống

Dữ liệu đầu vào:

Thông tin sách: tên, tác giả, nhà xuất bản, năm xuất bản, giá bán, số lượng tồn kho, mô tả, hình ảnh, thể loại,...

Thông tin người dùng: tên, email, mật khẩu, địa chỉ, số điện thoại.

Thông tin đơn hàng: sản phẩm, số lượng, giá, thông tin giao hàng, phương thức thanh toán.

Dữ liệu từ người dùng: đánh giá, bình luận, phản hồi.

Dữ liệu đầu ra:

Trang web hiển thị danh sách sách, thông tin chi tiết sách.

Giỏ hàng và thông tin đặt hàng.

Hóa đơn điện tử.

Thông báo trạng thái đơn hàng.

Báo cáo thống kê doanh số, hàng tồn kho, hành vi người dùng (cho admin).

Đề xuất sách phù hợp với người dùng.

1.6. Ý nghĩa của đề tài

Đề tài xây dựng website bán sách trực tuyến mang nhiều giá trị cả về mặt ứng dụng thực tiễn lẫn học thuật.

Về ý nghĩa thực tiễn, đề tài hướng đến việc tạo ra một nền tảng số hiện đại, không chỉ đơn thuần là kênh bán hàng mà còn đóng vai trò cầu nối quan trọng giữa người đọc và thế giới sách, từ đó góp phần thúc đẩy và phát triển văn hóa đọc trong xã hội đương đại.

Song song với đó, về mặt học thuật, đề tài tạo cơ hội quý báu để vận dụng toàn diện và sáng tạo các kiến thức chuyên môn đã được trang bị trong lĩnh vực phát triển web, thiết kế và quản trị cơ sở dữ liệu, các giải pháp đảm bảo an ninh bảo mật, cùng các nguyên lý và mô hình thương mại điện tử hiện đại vào một dự án có tính thực tiễn cao.

1.7. Kết quả dự kiến đạt được

Dự kiến, sau khi hoàn thành, đề tài sẽ cung cấp một hệ thống quản lý đăng ký thực hành hiệu quả, đáp ứng được nhu cầu thực tế của cả sinh viên và giảng viên. Kết quả dự kiến của đề tài bao gồm:

- + Website bán sách trực tuyến hoàn chỉnh với đầy đủ tính năng cơ bản
- + Hệ thống quản trị backend cho phép quản lý sách, người dùng và đơn hàng hiệu quả
- + Tài liệu kỹ thuật chi tiết về thiết kế và phát triển hệ thống
- + Báo cáo kết quả thử nghiệm và đánh giá hiệu năng
- + Kế hoạch phát triển trong tương lai để mở rộng quy mô và tính năng của hệ thống
- + Website sẽ được xây dựng đáp ứng các tiêu chí về tính khả dụng, hiệu năng, bảo mật và khả năng mở rộng, tạo nền tảng vững chắc cho việc phát triển tiếp theo.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1. Phân tích, thiết kế và lập trình hướng đối tượng

2.1.1. Phân tích và thiết kế hướng đối tượng

Phân tích và thiết kế hướng đối tượng (Object-Oriented Analysis and Design – OOAD) là một phương pháp phát triển phần mềm dựa trên việc trừu tượng hóa các thành phần trong thế giới thực thành các đối tượng phần mềm. Phương pháp này hướng tới việc mô hình hóa các thực thể một cách trực quan, dễ hiểu, và phản ánh đúng mối quan hệ giữa các thành phần trong hệ thống.

Khái niệm và vai trò của OOAD

OOAD gồm hai giai đoạn chính:

Phân tích hướng đối tượng (Object-Oriented Analysis – OOA): là quá trình xác định các yêu cầu của hệ thống thông qua việc nhận diện các đối tượng có trong thế giới thực, xác định đặc điểm và hành vi của chúng, cũng như mối quan hệ giữa các đối tượng đó. OOA tập trung vào vấn đề "cái gì" cần được xây dựng.

Thiết kế hướng đối tượng (Object-Oriented Design – OOD): là bước tiếp theo sau phân tích, nhằm định nghĩa rõ ràng "cách thức" triển khai các đối tượng đã xác định trong hệ thống phần mềm. Thiết kế bao gồm mô tả cấu trúc lớp (class), giao diện (interface), các phương thức và cách các đối tượng tương tác với nhau.

Phương pháp OOAD mang lại nhiều lợi ích, trong đó nổi bật là:

- + Tính tái sử dụng cao: Nhờ vào việc đóng gói dữ liệu và hành vi trong cùng một đối tượng.
- + Tính bảo trì và mở rộng tốt: Hệ thống được chia nhỏ thành các thành phần độc lập dễ thay đổi, nâng cấp.
- + Tính trừu tượng và mô hình hóa mạnh mẽ: Giúp giảm độ phức tạp của hệ thống.

Các nguyên lý cơ bản trong thiết kế hướng đối tượng

+ Tính đóng gói (Encapsulation): Giấu thông tin nội bộ của đối tượng, chỉ cho phép truy cập thông qua các phương thức công khai.

+ Tính kế thừa (Inheritance): Cho phép một lớp con kế thừa thuộc tính và phương thức từ lớp cha, giúp tái sử dụng mã nguồn hiệu quả.

+ Tính đa hình (Polymorphism): Các đối tượng có thể thực hiện hành vi giống nhau nhưng theo các cách khác nhau.

+ Tính trừu tượng (Abstraction): Chỉ tập trung vào các đặc điểm quan trọng của đối tượng, loại bỏ các chi tiết không cần thiết.

Việc áp dụng OOAD không chỉ giúp cho hệ thống dễ phát triển và bảo trì mà còn tạo điều kiện thuận lợi cho việc mở rộng chức năng trong tương lai như tích hợp các phương thức thanh toán mới, bổ sung tính năng đánh giá sách,...

2.1.2. Lập trình hướng đối tượng

Lập trình hướng đối tượng (Object-Oriented Programming – OOP) là một mô hình lập trình trong đó phần mềm được tổ chức dựa trên khái niệm "đối tượng" – các thực thể kết hợp dữ liệu (thuộc tính) và hành vi (phương thức) để mô tả và điều khiển trạng thái trong hệ thống.

Khái niệm và nguyên lý cốt lõi của OOP

OOP là sự mở rộng tự nhiên của phân tích và thiết kế hướng đối tượng (OOAD), trong đó các lớp (class) và đối tượng (object) được cài đặt bằng ngôn ngữ lập trình. Mục tiêu là giảm sự phức tạp của hệ thống phần mềm thông qua các nguyên lý cấu trúc:

+ Đóng gói (Encapsulation): Đóng gói dữ liệu và phương thức xử lý trong một lớp duy nhất, giúp ẩn chi tiết nội bộ và chỉ để lộ ra các giao diện cần thiết.

+ Kế thừa (Inheritance): Cho phép lớp con kế thừa thuộc tính và hành vi từ lớp cha, hỗ trợ tái sử dụng mã và phát triển mở rộng.

+ Đa hình (Polymorphism): Các đối tượng thuộc các lớp khác nhau có thể thực thi cùng một phương thức theo cách riêng của mình, giúp tăng tính linh hoạt trong xử lý.

+ Trừu tượng hóa (Abstraction): Chỉ hiển thị các thông tin quan trọng, ẩn đi chi tiết phức tạp không cần thiết.

Ưu điểm của lập trình hướng đối tượng

OOP mang lại nhiều lợi ích vượt trội so với các phương pháp lập trình truyền thống (như lập trình thủ tục), bao gồm:

+ Khả năng mở rộng cao: Dễ dàng thêm chức năng mới bằng cách mở rộng các lớp hiện có.

+ Tái sử dụng mã: Nhờ cơ chế kế thừa và xây dựng các thư viện lớp dùng lại được.

+ Tăng tính bảo trì: Mã nguồn có cấu trúc rõ ràng, dễ đọc, dễ kiểm soát lỗi.

+ Hỗ trợ kiểm thử hiệu quả: Mỗi đối tượng hoặc lớp có thể được kiểm thử riêng biệt.

2.2. Công nghệ phát triển ứng dụng website

2.2.1. Ngôn ngữ mô hình hóa UML

Unified Modeling Language (UML) là một ngôn ngữ mô hình hóa tiêu chuẩn, được phát triển nhằm hỗ trợ quá trình phân tích và thiết kế hệ thống phần mềm theo hướng đối tượng. UML không phải là một ngôn ngữ lập trình, mà là ngôn ngữ trực quan hóa các thành phần cấu trúc và hành vi của phần mềm thông qua các loại sơ đồ khác nhau.

Mục đích và vai trò của UML

UML cung cấp một tập hợp quy ước thống nhất giúp cho:

+ Diễn đạt yêu cầu và kiến trúc hệ thống một cách trực quan.

+ Tăng khả năng giao tiếp giữa các thành viên trong nhóm phát triển.

+ Giảm rủi ro sai lệch trong quá trình thiết kế và cài đặt.

+ Tái sử dụng các mô hình, cấu trúc hệ thống trong các dự án khác.

UML đặc biệt hữu ích khi áp dụng vào phương pháp phân tích và thiết kế hướng đối tượng (OOAD), vì nó giúp biểu diễn rõ ràng các khái niệm như lớp, đối tượng, mối quan hệ, luồng xử lý,...

Các loại sơ đồ UML thường dùng

Trong dự án xây dựng website bán sách trực tuyến, một số sơ đồ UML cơ bản sẽ được sử dụng để mô hình hóa hệ thống bao gồm:

Use Case Diagram (Sơ đồ ca sử dụng): Mô tả các chức năng chính của hệ thống thông qua các tương tác giữa người dùng (actor) và hệ thống. Ví dụ: đặt hàng, đăng nhập, thêm sách mới, duyệt đơn hàng,...

Class Diagram (Sơ đồ lớp): Mô hình hóa các lớp (classes) như Book, User, Order, Cart, Admin cùng các thuộc tính, phương thức và mối quan hệ (kế thừa, kết hợp, liên kết) giữa các lớp.

Sequence Diagram (Sơ đồ tuần tự): Diễn tả sự tương tác giữa các đối tượng theo trình tự thời gian. Ví dụ, khi người dùng đặt hàng, các bước gọi hàm từ frontend đến backend và cơ sở dữ liệu sẽ được mô tả rõ ràng

Ngoài ra, UML còn có các sơ đồ khác như Activity Diagram, Component Diagram, Deployment Diagram,... nhưng trong phạm vi đề tài, các sơ đồ nêu trên là cốt lõi và cần thiết nhất.

Lợi ích khi sử dụng UML trong dự án, việc áp dụng UML trong quá trình thiết kế hệ thống mang lại nhiều lợi ích:

- + Giảm thiểu lỗi thiết kế nhờ mô hình hóa trực quan trước khi lập trình
- + Dễ dàng trao đổi và cập nhật ý tưởng trong nhóm phát triển
- + Hỗ trợ tạo tài liệu kỹ thuật rõ ràng, phục vụ cho bảo trì và nâng cấp sau này
- + Làm nền tảng cho việc chuyển đổi mô hình thành mã nguồn (Model-to-Code)

2.2.2. Ngôn ngữ thiết kế website

Trong phát triển giao diện người dùng cho ứng dụng web, ba công nghệ cốt lõi là HTML5, CSS3 và JavaScript. Đây là nền tảng kỹ thuật của bất kỳ website hiện đại nào, giúp hiện thực hóa cấu trúc, phong cách trình bày và hành vi tương tác của trang web.

2.2.2.1. HTML 5

HTML5 (Hypertext Markup Language phiên bản 5) là phiên bản mới nhất của ngôn ngữ đánh dấu siêu văn bản, được phát triển bởi World Wide Web Consortium (W3C) và Web Hypertext Application Technology Working Group (WHATWG). HTML5 được thiết kế để cải thiện khả năng tương tác, tính năng đa phương tiện và khả năng phát triển ứng dụng web.

Đặc điểm nổi bật:

Bổ sung các thẻ ngữ nghĩa như `<header>`, `<footer>`, `<section>`, `<article>` giúp mã nguồn rõ ràng, dễ hiểu và thân thiện với công cụ tìm kiếm.

Tích hợp các thẻ đa phương tiện như `<audio>`, `<video>`, cho phép nhúng media mà không cần plugin bên ngoài.

Hỗ trợ form nâng cao với các kiểu input như email, date, number, giúp tăng trải nghiệm người dùng.

Tích hợp khả năng lưu trữ cục bộ như `localStorage`, `sessionStorage`, phục vụ các ứng dụng offline.

2.2.2.2. CSS 3

CSS3 (Cascading Style Sheets Level 3) là phiên bản mới nhất của ngôn ngữ định kiểu CSS, được sử dụng để kiểm soát cách mà các phần tử HTML được hiển thị trên trang web. CSS3 mang đến nhiều tính năng và cải tiến so với các phiên bản trước, giúp các nhà phát triển và thiết kế web tạo ra giao diện người dùng phong phú và hấp dẫn hơn.

Tính năng nổi bật:

Responsive design: Thiết kế giao diện linh hoạt, tương thích với các thiết bị khác nhau thông qua media queries.

Flexbox và Grid Layout: Cung cấp hệ thống bố cục mạnh mẽ và hiện đại, giúp dễ dàng căn chỉnh và sắp xếp phần tử.

Animation và transition: Cho phép tạo hiệu ứng chuyển động mượt mà mà không cần JavaScript.

Hỗ trợ biến CSS, pseudo-classes và pseudo-elements giúp mở rộng khả năng tái sử dụng và kiểm soát chi tiết giao diện.

2.2.2.3. Javascript

JavaScript là một ngôn ngữ lập trình mạnh mẽ và linh hoạt, chủ yếu được sử dụng để tạo ra các hành vi tương tác động trên trang web. Được phát triển lần đầu vào giữa những năm 1990, JavaScript đã trở thành một phần không thể thiếu trong phát triển web hiện đại.

Tạo tương tác người dùng

JavaScript cho phép các nhà phát triển tạo ra các tương tác phong phú và động trên trang web.

Xác thực form: JavaScript có thể kiểm tra dữ liệu người dùng nhập vào trước khi gửi đến máy chủ, giúp giảm thiểu lỗi và cải thiện trải nghiệm người dùng. Ví dụ, kiểm tra xem một trường email có đúng định dạng hay không.

Hiển thị thông báo: Các thông báo có thể được hiển thị hoặc ẩn đi dựa trên hành động của người dùng, như thông báo thành công hoặc lỗi khi gửi form.

Hiệu ứng động: JavaScript có thể tạo ra các hiệu ứng như cuộn mượt mà, ẩn/hiện các phần tử, hoặc thay đổi màu sắc và kích thước của các phần tử khi người dùng tương tác.

DOM Manipulation

DOM (Document Object Model) là một cấu trúc dữ liệu đại diện cho tài liệu HTML. JavaScript cho phép thay đổi nội dung và thuộc tính của các phần tử HTML mà không cần tải lại trang. Một số thao tác phổ biến bao gồm:

Thêm, xóa hoặc thay đổi các phần tử: JavaScript có thể thêm các phần tử mới vào DOM, xóa các phần tử hiện có hoặc thay đổi nội dung và thuộc tính của các phần tử.

Thay đổi kiểu dáng: Các thuộc tính CSS có thể được thay đổi thông qua JavaScript, cho phép tạo ra các hiệu ứng động và tương tác.

Sự kiện: JavaScript có thể lắng nghe và xử lý các sự kiện như nhấp chuột, di chuột, và nhập liệu, từ đó thực hiện các hành động tương ứng.

AJAX và Fetch API

AJAX (Asynchronous JavaScript and XML) và Fetch API cho phép gửi và nhận dữ liệu từ máy chủ một cách bất đồng bộ, giúp cải thiện hiệu suất và trải nghiệm người dùng. Một số điểm nổi bật:

Gửi yêu cầu mà không tải lại trang: AJAX cho phép gửi yêu cầu đến máy chủ và nhận dữ liệu mà không cần làm mới toàn bộ trang, giúp cải thiện tốc độ và trải nghiệm người dùng.

Fetch API: Đây là một API hiện đại hơn để thực hiện các yêu cầu HTTP. Fetch API hỗ trợ Promise, giúp xử lý các yêu cầu bất đồng bộ một cách dễ dàng và rõ ràng hơn.

Xử lý dữ liệu: Dữ liệu nhận được từ máy chủ có thể được xử lý và hiển thị ngay lập tức trên trang mà không cần tải lại, tạo ra trải nghiệm người dùng mượt mà hơn.

Hỗ trợ lập trình hướng đối tượng

JavaScript hỗ trợ lập trình hướng đối tượng, cho phép tổ chức mã một cách có cấu trúc và dễ bảo trì. Một số tính năng nổi bật bao gồm:

Class và Object: ES6 giới thiệu cú pháp class, giúp việc định nghĩa và sử dụng các đối tượng trở nên dễ dàng hơn. Các đối tượng có thể chứa các thuộc tính và phương thức, cho phép tái sử dụng mã.

Kế thừa: JavaScript hỗ trợ kế thừa, cho phép một class có thể kế thừa các thuộc tính và phương thức từ một class khác, giúp tổ chức mã hiệu quả hơn.

Hỗ trợ lập trình bất đồng bộ

JavaScript cung cấp nhiều cách để xử lý lập trình bất đồng bộ, giúp cải thiện khả năng xử lý các tác vụ không đồng bộ mà không làm chậm giao diện người dùng. Một số tính năng quan trọng bao gồm:

Promise: Promise là một đối tượng đại diện cho một giá trị có thể có trong tương lai, giúp xử lý các tác vụ bất đồng bộ một cách dễ dàng hơn.

Async/Await: Đây là cú pháp mới trong ES6+ cho phép viết mã bất đồng bộ trông giống như mã đồng bộ, giúp cải thiện khả năng đọc và bảo trì mã. (Promise, async/await).

2.2.3. Mô hình Client – Server

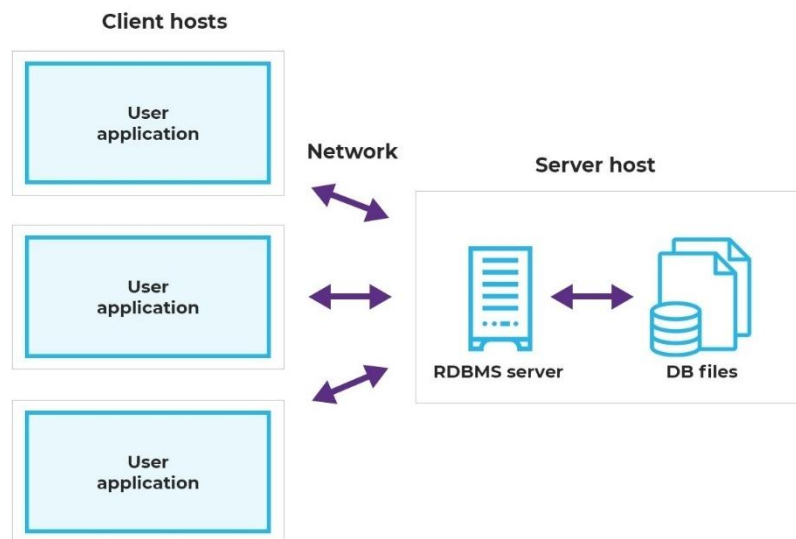
Mô hình Client - Server là một kiến trúc phần mềm trong đó các thiết bị (Client) gửi yêu cầu đến một máy chủ trung tâm (Server), nơi xử lý yêu cầu và phản hồi lại dữ liệu. Mô hình này được sử dụng rộng rãi trong các hệ thống mạng, đặc biệt là trong phát triển ứng dụng web và hệ thống phân tán. Việc tách biệt vai trò của client và server giúp tổ chức hệ thống rõ ràng, dễ quản lý và bảo trì.

Các thành phần chính của mô hình:

Client (Khách): Đây là thành phần giao tiếp trực tiếp với người dùng. Client gửi các yêu cầu đến server, chẳng hạn như yêu cầu truy xuất dữ liệu, điền biểu mẫu, đăng nhập,... Client thường là trình duyệt web, ứng dụng desktop, hoặc ứng dụng di động. Nó không xử lý logic nghiệp vụ, chỉ chịu trách nhiệm giao tiếp và hiển thị thông tin.

Server (Chủ): Đây là thành phần tiếp nhận và xử lý các yêu cầu từ client. Server thực hiện các công việc như xử lý logic nghiệp vụ, truy vấn cơ sở dữ liệu, và phản hồi dữ liệu về client. Server có thể được xây dựng bằng các công nghệ như Node.js, Java, PHP, Python,...

Database (Cơ sở dữ liệu): Là nơi lưu trữ dữ liệu của hệ thống. Server sẽ tương tác với cơ sở dữ liệu để thực hiện các thao tác như tìm kiếm, thêm, sửa, xóa dữ liệu khi có yêu cầu từ phía client.



Hình 2.1 Trực quan mô hình Client - Server

Lợi ích của mô hình Client – Server:

- + Tổ chức rõ ràng, dễ quản lý: Phân chia rõ vai trò giữa client và server giúp hệ thống có cấu trúc rõ ràng, dễ theo dõi và bảo trì.
- + Dễ mở rộng: Có thể mở rộng hệ thống bằng cách nâng cấp server, thêm client hoặc triển khai nhiều server xử lý song song.
- + Tăng tính bảo mật: Dữ liệu được lưu trữ và xử lý tập trung trên server, client không truy cập trực tiếp đến cơ sở dữ liệu.
- + Thích hợp cho hệ thống đa người dùng: Nhiều client có thể tương tác đồng thời với một server duy nhất mà không ảnh hưởng đến nhau.

2.2.4. NodeJS – Express

Node.js là một môi trường chạy JavaScript phía máy chủ, được phát triển dựa trên V8 JavaScript Engine của trình duyệt Google Chrome. Điểm nổi bật của Node.js là cơ chế xử lý bất đồng bộ (asynchronous) và I/O không đồng bộ chặn luồng (non-blocking I/O), cho phép hệ thống có khả năng xử lý hàng nghìn yêu cầu đồng thời mà không cần tạo luồng mới cho mỗi kết nối. Điều này giúp tối ưu hiệu suất, giảm thiểu tài nguyên máy chủ và tăng tốc độ phản hồi – đặc biệt thích hợp cho các ứng dụng web có tính tương tác cao như hệ thống thương mại điện tử.

Bên cạnh đó, Express.js là một framework nhẹ chạy trên nền Node.js, cung cấp các công cụ và tính năng giúp xây dựng ứng dụng web và API một cách hiệu quả. Express hỗ trợ:

- + Routing: Định tuyến các yêu cầu HTTP đến các hàm xử lý (controller) tương ứng.
- + Middleware: Cho phép xử lý các chức năng trung gian như kiểm tra xác thực, log dữ liệu, xử lý lỗi,...
- + RESTful API: Hỗ trợ dễ dàng xây dựng các giao diện lập trình ứng dụng tuân thủ chuẩn REST, phù hợp cho các hệ thống frontend-backend tách biệt.

2.2.5. Hệ quản trị cơ sở dữ liệu

2.2.5.1. Các khái niệm cơ bản

Cơ sở dữ liệu là nơi lưu trữ có tổ chức các thông tin quan trọng liên quan đến hoạt động của hệ thống, chẳng hạn như thông tin người dùng, sản phẩm, đơn hàng,... Để quản lý dữ liệu một cách hiệu quả, người ta sử dụng Hệ quản trị cơ sở dữ liệu (Database Management System – DBMS) là phần mềm trung gian cho phép người dùng tương tác với dữ liệu thông qua các thao tác tiêu chuẩn.

Một DBMS hỗ trợ các chức năng như tạo lập cấu trúc dữ liệu (bảng hoặc tài liệu), thực hiện các thao tác CRUD (Create – thêm mới, Read – đọc, Update – cập

nhập, Delete – xóa), quản lý truy cập và phân quyền người dùng, bảo đảm tính bảo mật dữ liệu, đồng thời tối ưu hóa hiệu suất truy vấn và lưu trữ dữ liệu.

Hiện nay, hệ quản trị cơ sở dữ liệu được chia làm hai nhóm chính:

Hệ quản trị cơ sở dữ liệu quan hệ (Relational DBMS – RDBMS) như MySQL, PostgreSQL, sử dụng mô hình bảng và quan hệ khóa chính – khóa ngoại

Hệ quản trị cơ sở dữ liệu phi quan hệ (NoSQL DBMS) như MongoDB, CouchDB, thường sử dụng mô hình tài liệu (document-based) hoặc cặp khóa-giá trị để lưu trữ dữ liệu linh hoạt và dễ mở rộng.

Việc lựa chọn loại DBMS phù hợp phụ thuộc vào đặc điểm của dữ liệu và yêu cầu hệ thống.

2.2.5.2. *MongoDB*

MongoDB là một hệ quản trị cơ sở dữ liệu NoSQL rất phổ biến và nổi bật trong thế giới các hệ cơ sở dữ liệu phi quan hệ (Non-relational). Điều này có nghĩa là nó không sử dụng các bảng (tables), hàng (rows), và cột (columns) như trong các hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) truyền thống như MySQL, PostgreSQL. Thay vào đó, MongoDB lưu trữ dữ liệu dưới dạng tài liệu (document), thường sử dụng cấu trúc JSON hoặc BSON (Binary JSON), cho phép linh hoạt và dễ dàng mở rộng.

Điểm mạnh của MongoDB với tư cách là một cơ sở dữ liệu NoSQL:

Lưu trữ dữ liệu linh hoạt

MongoDB không yêu cầu định nghĩa schema cố định như trong các hệ quản trị cơ sở dữ liệu quan hệ (RDBMS). Các tài liệu trong MongoDB có thể có cấu trúc khác nhau, và việc thay đổi cấu trúc dữ liệu trong quá trình phát triển hoặc vận hành là rất dễ dàng.

Điều này giúp MongoDB trở thành một sự lựa chọn tuyệt vời cho các ứng dụng có yêu cầu thay đổi hoặc phát triển nhanh chóng, vì việc điều chỉnh cơ sở dữ liệu không bị ràng buộc bởi schema cứng nhắc.

Mở rộng theo chiều ngang (Horizontal scaling)

MongoDB được thiết kế để hỗ trợ mở rộng theo chiều ngang (horizontal scaling). Điều này có nghĩa là khi cần xử lý nhiều dữ liệu hơn hoặc khi ứng dụng cần phục vụ nhiều người dùng hơn, MongoDB có thể dễ dàng phân phối dữ liệu trên nhiều máy chủ (server) khác nhau.

Các cơ sở dữ liệu NoSQL như MongoDB cho phép người dùng chia nhỏ dữ liệu và lưu trữ chúng trên nhiều máy chủ khác nhau, giúp dễ dàng mở rộng mà không gặp phải các vấn đề về hiệu suất như trong các hệ quản trị cơ sở dữ liệu quan hệ truyền thống.

Tương thích với Node.js nhờ thư viện Mongoose

MongoDB có sự tương thích tuyệt vời với Node.js nhờ thư viện Mongoose, một thư viện rất phổ biến giúp việc tương tác với MongoDB dễ dàng hơn trong các ứng dụng Node.js.

Mongoose cung cấp các công cụ mạnh mẽ như mô hình hóa dữ liệu, xác thực, middleware, và nhiều tính năng khác để giúp việc tương tác với cơ sở dữ liệu trở nên mượt mà hơn.

CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

3.1. Khảo sát hệ thống

Hiện nay, nhu cầu mua sắm sách trực tuyến tại Việt Nam đang phát triển mạnh mẽ nhờ vào sự phổ biến của thương mại điện tử và thói quen tiêu dùng hiện đại. Nhiều người dùng, đặc biệt là học sinh, sinh viên và người đi làm, có xu hướng đặt mua sách online thay vì đến cửa hàng truyền thống để tiết kiệm thời gian và tiếp cận kho sách đa dạng hơn.

Nắm bắt xu thế đó, nhiều nhà sách, nhà xuất bản và các cá nhân kinh doanh đã triển khai các website bán sách. Các website này cho phép người dùng tra cứu thông tin sách, đặt mua, thanh toán và nhận hàng tại nhà. Tuy nhiên, nhiều hệ thống vẫn còn tồn tại nhiều bất cập về tính năng, hiệu năng, và trải nghiệm người dùng.

Một số website bán sách hiện nay được xây dựng với giao diện lỗi thời, thiếu tính thân thiện, đặc biệt là khi truy cập trên thiết bị di động. Ngoài ra, khả năng tìm kiếm, phân loại và đề xuất sách còn hạn chế, khiến người dùng mất nhiều thời gian để chọn được sản phẩm phù hợp.

Đặc biệt, nhiều hệ thống thiếu chức năng đánh giá và bình luận sách một cách hiệu quả, dẫn tới việc người dùng không có cơ sở để tham khảo trước khi đặt mua. Việc quản lý hàng tồn kho, giá khuyến mãi, hay cập nhật thông tin sản phẩm ở nhiều nơi vẫn thực hiện thủ công, dễ gây ra sai sót và khó khăn trong quản lý.

Ngoài ra, chức năng quản trị viên trong nhiều hệ thống còn đơn giản, chưa hỗ trợ tốt việc theo dõi đơn hàng, xử lý yêu cầu khách hàng, hay thống kê doanh thu, dẫn đến hiệu suất vận hành chưa tối ưu. Tình trạng chưa tích hợp thanh toán trực tuyến linh hoạt, hay thiếu bảo mật trong quản lý thông tin người dùng cũng là những vấn đề đáng lưu ý.

3.2. Mô tả bài toán

Từ thực tế nhu cầu và những hạn chế trong các hệ thống bán sách trực tuyến hiện nay, bài toán đặt ra là: Xây dựng một website bán sách trực tuyến nhằm cung cấp nền tảng mua sắm hiện đại, thân thiện với người dùng, hỗ trợ đầy đủ chức năng từ hiển thị sản phẩm, quản lý đơn hàng, đánh giá sách đến quản trị hệ thống, góp phần nâng cao trải nghiệm khách hàng và hiệu quả vận hành.

Cụ thể, hệ thống cần giải quyết các vấn đề sau:

Cung cấp giao diện người dùng thân thiện, dễ sử dụng trên cả máy tính và thiết bị di động, cho phép người dùng:

Duyệt danh mục sách theo thể loại, tên tác giả, mức giá,...

Tìm kiếm sách nhanh chóng, có bộ lọc nâng cao.

Xem chi tiết thông tin sản phẩm (hình ảnh, mô tả, đánh giá,...).

Thêm sản phẩm vào giỏ hàng, đặt mua và theo dõi đơn hàng.

Xây dựng hệ thống quản trị dành cho quản trị viên, bao gồm:

Quản lý sách (thêm/sửa/xóa); Quản lý danh mục, khuyến mãi, tồn kho; Quản lý người dùng, đơn hàng, lịch sử giao dịch.; Xem thống kê bán hàng theo thời gian.

Tối ưu quá trình xử lý đơn hàng: Cho phép thanh toán (giả lập hoặc tích hợp thanh toán online); Theo dõi trạng thái đơn hàng (chờ xác nhận, vận chuyển, hoàn tất,...).

Tích hợp chức năng đánh giá và bình luận sản phẩm, giúp người dùng đưa ra quyết định mua hàng tốt hơn.

Bảo mật thông tin người dùng, phân quyền rõ ràng giữa quản trị viên và người mua.

Dễ dàng mở rộng, bảo trì và nâng cấp trong tương lai (ví dụ: tích hợp thêm sách điện tử, hệ thống đề xuất cá nhân hóa,...).

3.3. Xác định các yêu cầu của hệ thống

Các yêu cầu chức năng đối với trang web bao gồm:

- + Lưu trữ các thông tin về các sản phẩm sách.
- + Giao diện bắt mắt, thân thiện, dễ dàng sử dụng cho mọi đối tượng người dùng.
- + Trang bị chức năng tìm kiếm linh hoạt giúp người dùng có thể nhanh chóng tìm kiếm ra thông tin các sản phẩm.
- + Trang bị chức năng đặt hàng và thanh toán để người dùng có thể dễ dàng thanh toán khi nhận hàng.

3.4. Thiết kế hệ thống

3.4.1. Các tác nhân hệ thống

Trong hệ thống website bán sách, có các tác nhân chính: Nhân viên quản lý, Kiểm kho, Khách hàng và Khách hàng tiềm năng.

Nhân viên quản lý (Administrator) là người có quyền quản lý toàn bộ hệ thống, bao gồm việc thêm, sửa, xóa danh mục và sản phẩm, quản lý tài khoản khách hàng, xử lý đơn hàng, cập nhật chương trình khuyến mãi, theo dõi doanh thu,... Họ truy cập vào giao diện quản trị và thực hiện các thao tác nhằm đảm bảo hoạt động của website diễn ra suôn sẻ.

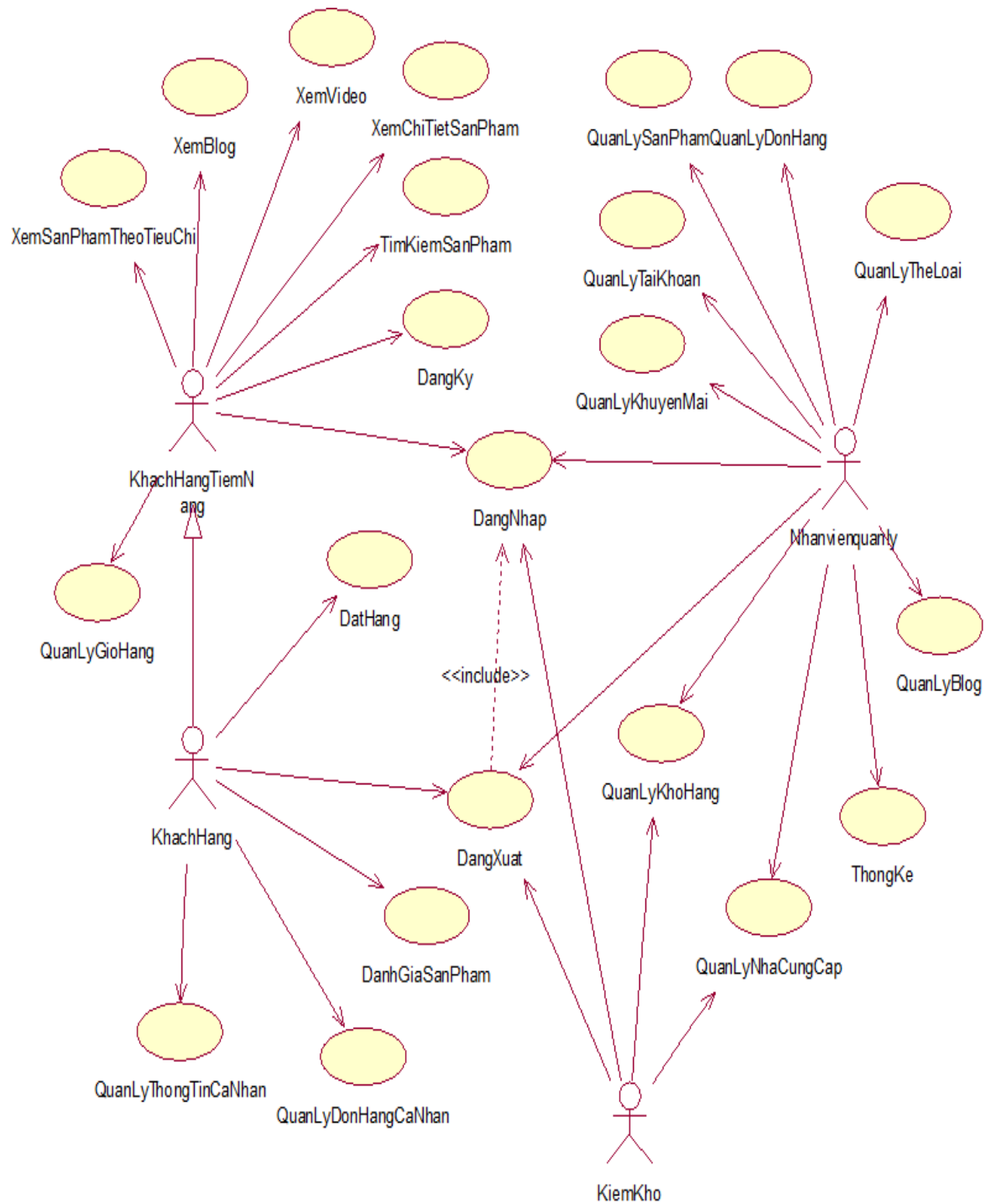
Kiểm kho (Inventory Manager) là người có quyền quản lý toàn bộ kho hàng và các nhà cung cấp. Họ truy cập vào giao diện quản trị và thực hiện các thao tác nhằm đảm bảo hoạt động nhập xuất kho hàng của website diễn ra suôn sẻ.

Trong khi đó, Khách hàng (Customer) là những người sử dụng website để tìm kiếm và mua sách. Khách hàng tiềm năng (Khách vãng lai) chỉ có thể xem sản phẩm, khuyến mãi mà không thể đặt hàng, trong khi Khách hàng đã đăng ký có thể đăng nhập, thêm sản phẩm vào giỏ hàng, đặt hàng, hủy đơn, theo dõi trạng thái giao hàng và chỉnh sửa thông tin cá nhân.

Hệ thống đảm bảo rằng mỗi tác nhân có quyền hạn phù hợp để thực hiện các chức năng của mình, giúp quá trình mua bán diễn ra thuận tiện và hiệu quả.

3.4.2. Các use case trong hệ thống

3.4.2.1. Sơ đồ use case tổng quan hệ thống



Hình 3.1 Sơ đồ use case tổng quan hệ thống

Nhóm khách hàng (Customer): Khách hàng có thể là nhóm khách hàng tiềm năng (khách vãng lai) hoặc là khách hàng có tài khoản cá nhân.

Đối với nhóm khách hàng tiềm năng ghé thăm không có tài khoản:

+ Đăng ký: Khách hàng tiềm năng (ghé thăm) có thể đăng ký tài khoản. Sau khi đăng ký và xác thực thành công, người dùng có thể đăng nhập vào hệ thống.

+ Đăng nhập: Khách hàng có thể đăng nhập bằng tài khoản đã đăng ký.

+ Tìm kiếm sản phẩm: Khách hàng có thể tìm kiếm thông tin sản phẩm theo tên sản phẩm và theo tác giả.

+ Xem chi tiết sản phẩm: Khách hàng cũng có thể xem được chi tiết sản phẩm.

+ Xem sản phẩm theo tiêu chí: Khách hàng cũng có thể xem các sản phẩm theo lựa chọn thể loại và lựa chọn sắp xếp của mình.

+ Xem blog: Cho phép khách hàng xem được các bài viết Blog mới nhất về cửa hàng

+ Thêm sản phẩm vào giỏ hàng: Khách hàng chọn sản phẩm cần mua để thêm vào giỏ hàng.

+ Quản lý giỏ hàng: Khách hàng có thể Thêm/Xóa sản phẩm vào/khỏi giỏ hàng, thay đổi số lượng của sản phẩm trong giỏ hàng.

Đối với nhóm khách hàng có tài khoản (Nhóm này kế thừa các usecase của nhóm khách hàng tiềm năng):

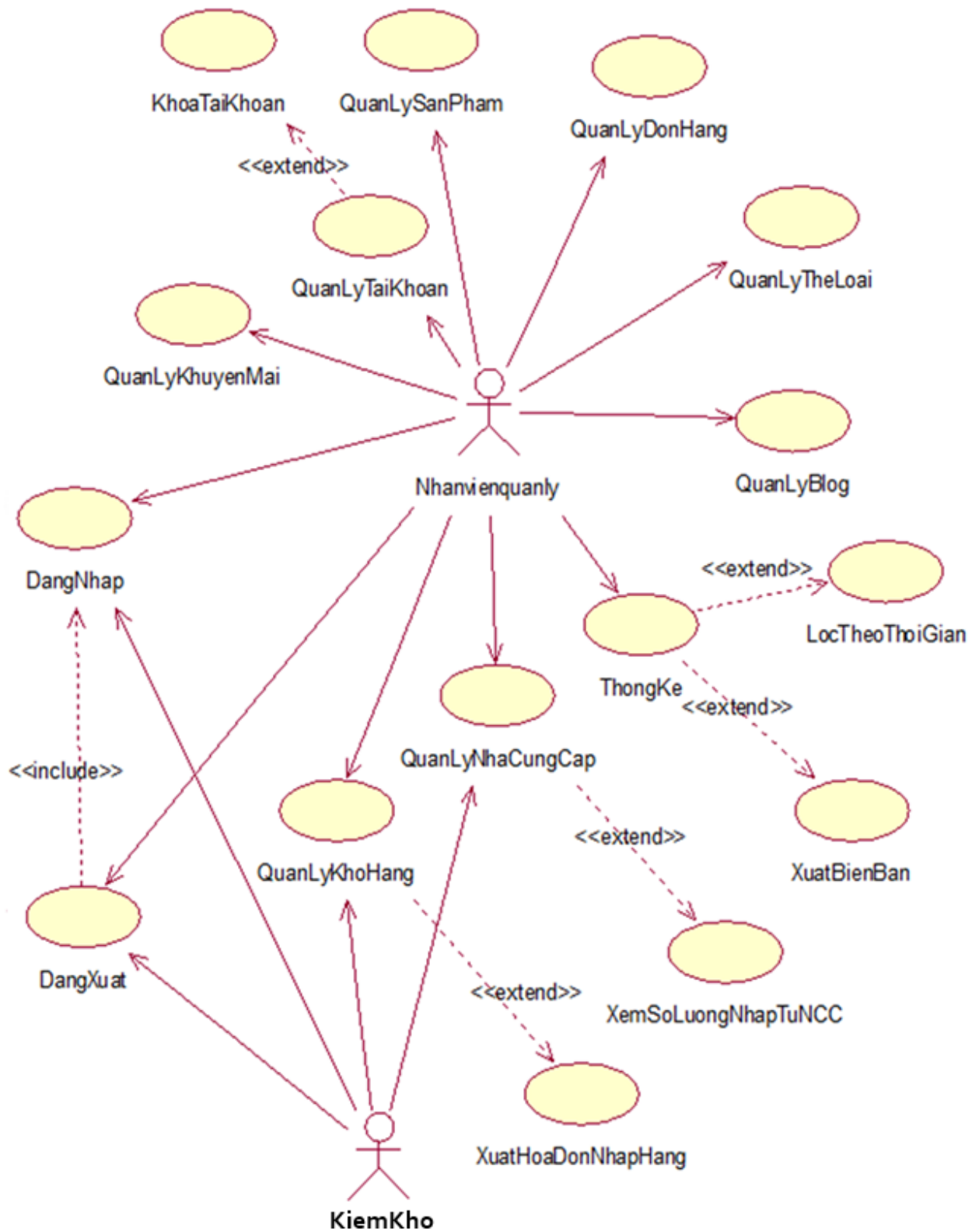
+ Quản lý đơn hàng cá nhân: Khách hàng có thể xem danh sách các đơn hàng, xem chi tiết các đơn hàng của chính khách hàng đó.

+ Quản lý thông tin cá nhân: Khách hàng có thể tự mình quản lý thông tin cá nhân của mình bao gồm thay đổi mật khẩu, tài khoản, tên đăng nhập.

+ Đặt hàng: Khách hàng có thể lựa chọn phương thức thanh toán cho đơn hàng của chính mình.

+ Đánh giá sản phẩm: Khách hàng có thể đánh giá và nhận xét sản phẩm dựa trên thứ hạng sao (gồm 5 hạng).

3.4.2.3. Sơ đồ use case Nhân viên quản lý



Hình 3.3 Sơ đồ use case Nhân viên quản lý và Kiểm kho

Nhóm nhân viên quản lý (Admin):

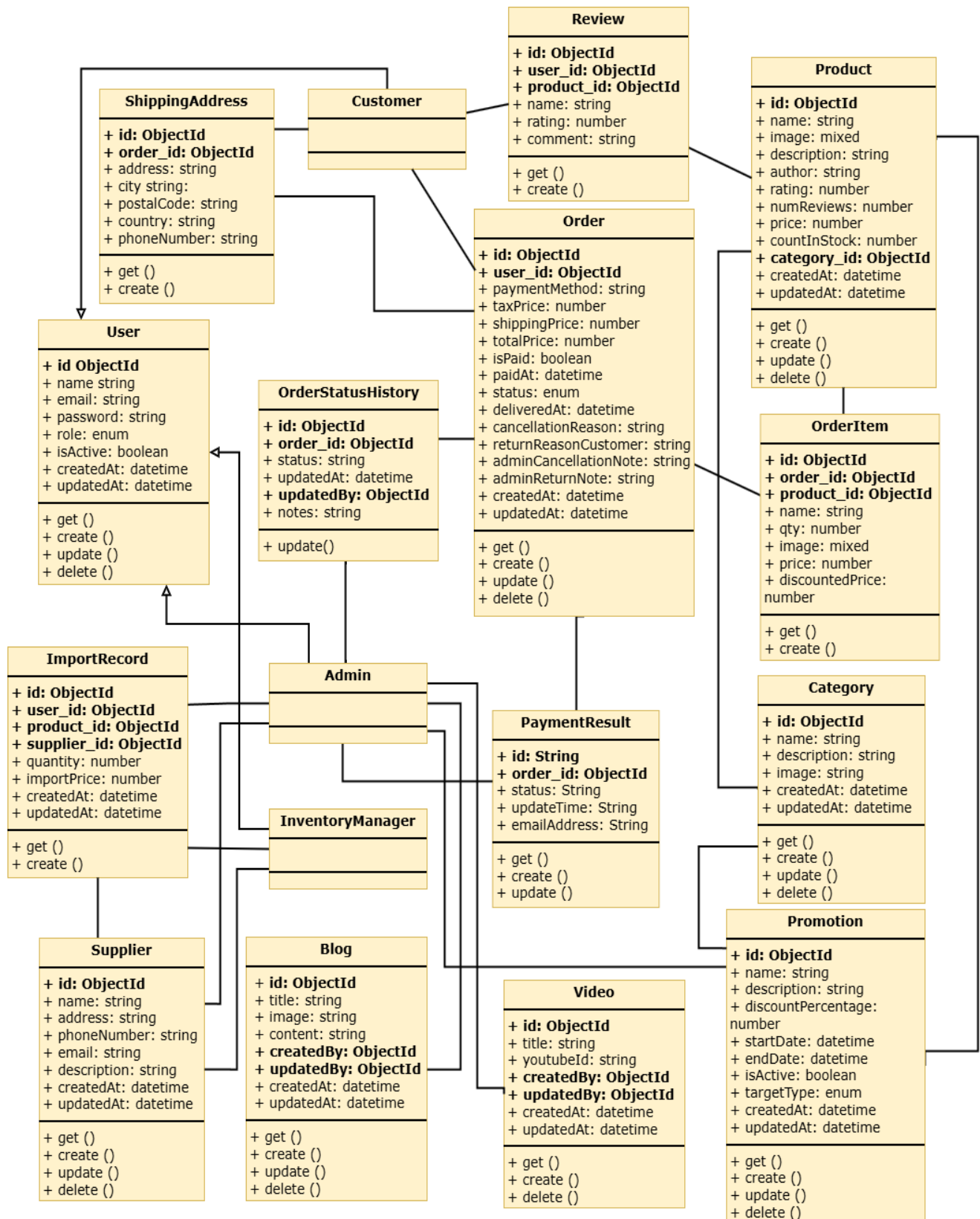
- + Đăng nhập: Nhân viên quản lý có thể đăng nhập và trang quản lý bằng tài khoản đã được cấp.
- + Quản lý tài khoản: Quản lý thông tin của các khách hàng đã đăng ký, có thể xem hoặc khóa danh sách các khách hàng trên hệ thống.
- + Quản lý sản phẩm: Quản lý thông tin sản phẩm cho phép nhân viên quản lý có quyền Thêm/Sửa/Xóa sản phẩm được lựa chọn. Ngoài ra còn có thể tìm kiếm sản phẩm cần quản lý.
- + Quản lý đơn hàng: Quản lý thông tin đơn hàng, cho phép xem danh sách đơn hàng, xem chi tiết đơn hàng của các khách hàng và sửa trạng thái đơn hàng.
- + Quản lý thể loại: Quản lý thể loại cho phép nhân viên quản lý có quyền Thêm/Sửa/Xóa thể loại được lựa chọn.
- + Quản lý blog: Cho phép nhân viên quản lý sửa/xóa và đăng bài viết mới.
- + Quản lý video: Cho phép nhân viên quản lý thêm/xóa video.
- + Thống kê: Quan sát Thống kê doanh thu và số lượng sản phẩm, số lượng đơn hàng và đơn hàng mới.

Ngoài ra, nhân viên quản lý cũng có thể sử dụng các chức năng của Kiểm kho

Nhóm Kiểm kho (Inventory Manager):

- + Quản lý kho hàng: Quản lý kho hàng cho phép nhân viên Kiểm kho có quyền nhập hàng và truy xuất ra hóa đơn nhập hàng.
- + Quản lý nhà cung cấp: Quản lý nhà cung cấp cho phép nhân viên Kiểm kho có quyền Thêm/Sửa/Xóa nhà cung cấp hàng cho kho hàng. Ngoài ra còn có thể xem số lượng hàng nhập từ nhà cung cấp.

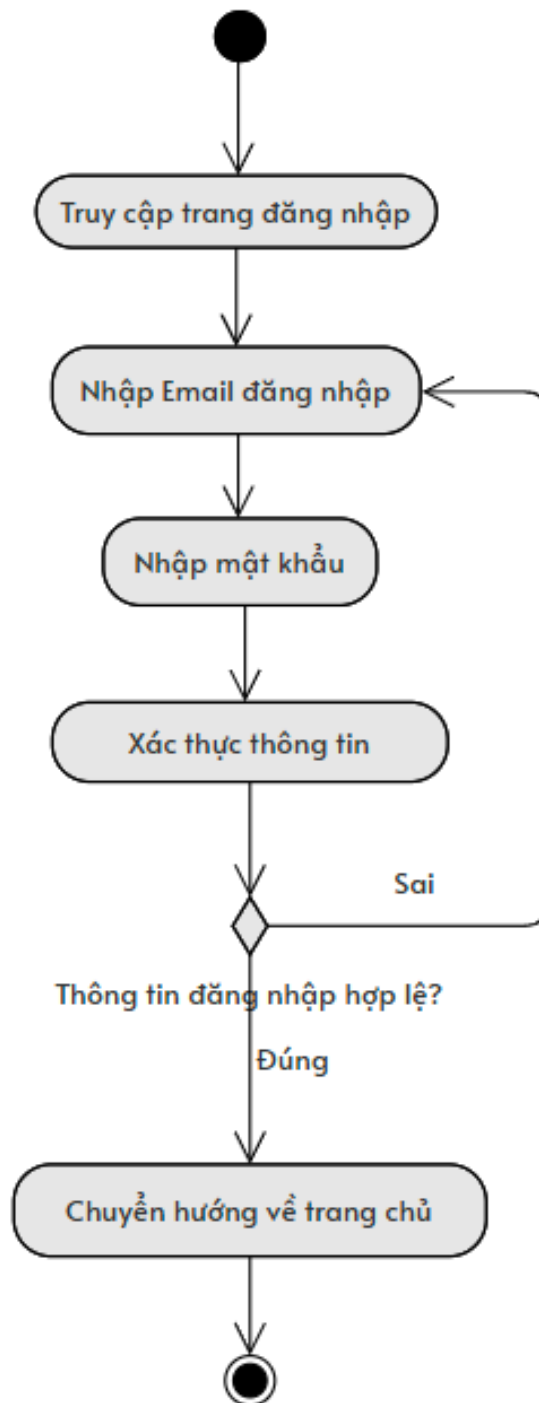
3.4.3. Biểu đồ lớp



Hình 3.4 Biểu đồ lớp hệ thống

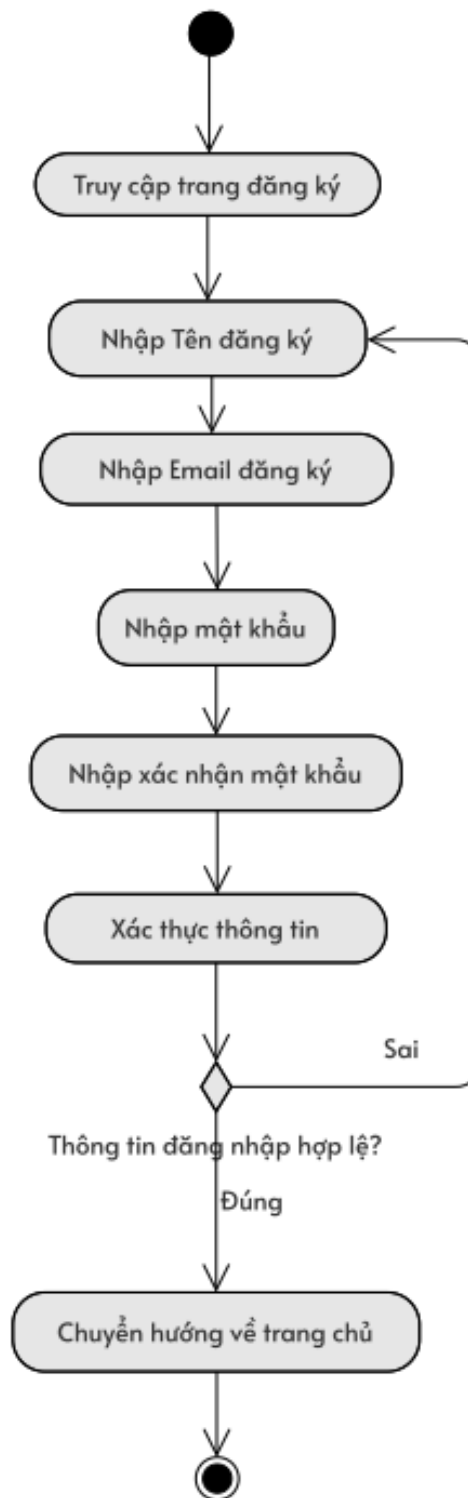
3.4.4. Biểu đồ hoạt động

3.4.4.1. Use case Đăng nhập



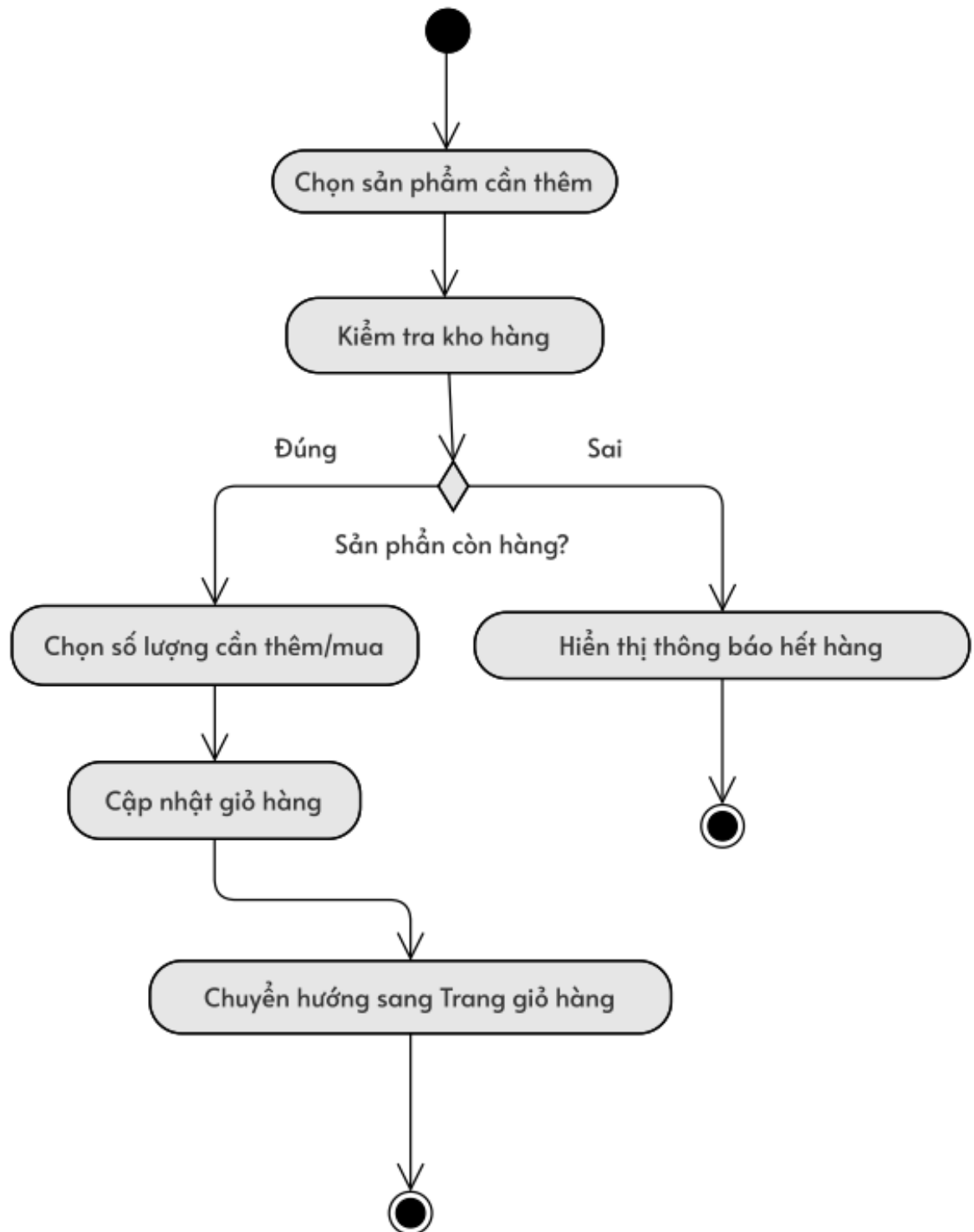
Hình 3.5 Biểu đồ hoạt động use case Đăng nhập

3.4.4.2. Use case Đăng ký



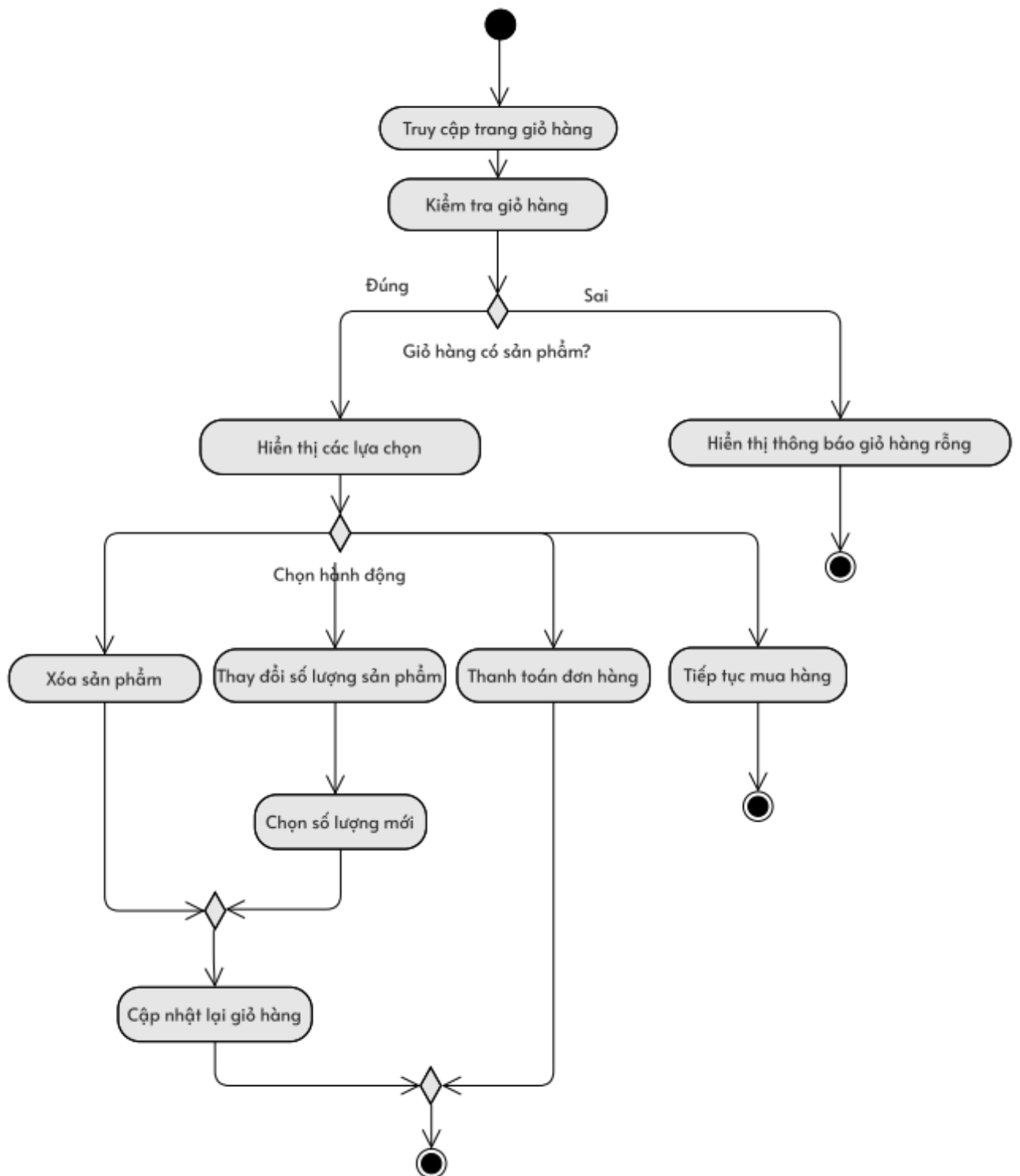
Hình 3.6 Biểu đồ hoạt động use case Đăng ký

3.4.4.3. Use case Thêm sản phẩm vào giỏ hàng



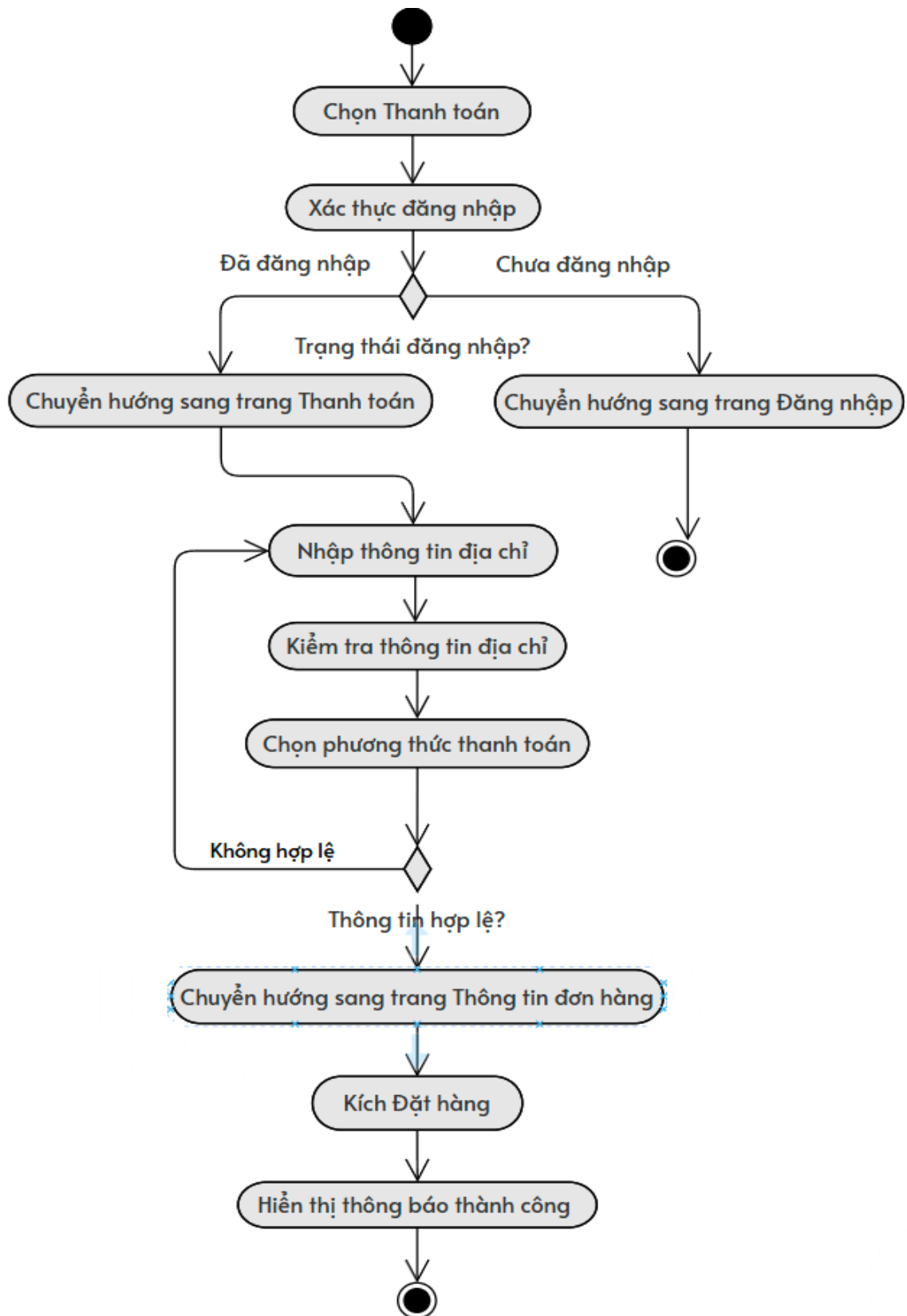
Hình 3.7 Biểu đồ hoạt động use case Thêm sản phẩm vào giỏ hàng

3.4.4.4. Use case Quản lý giỏ hàng



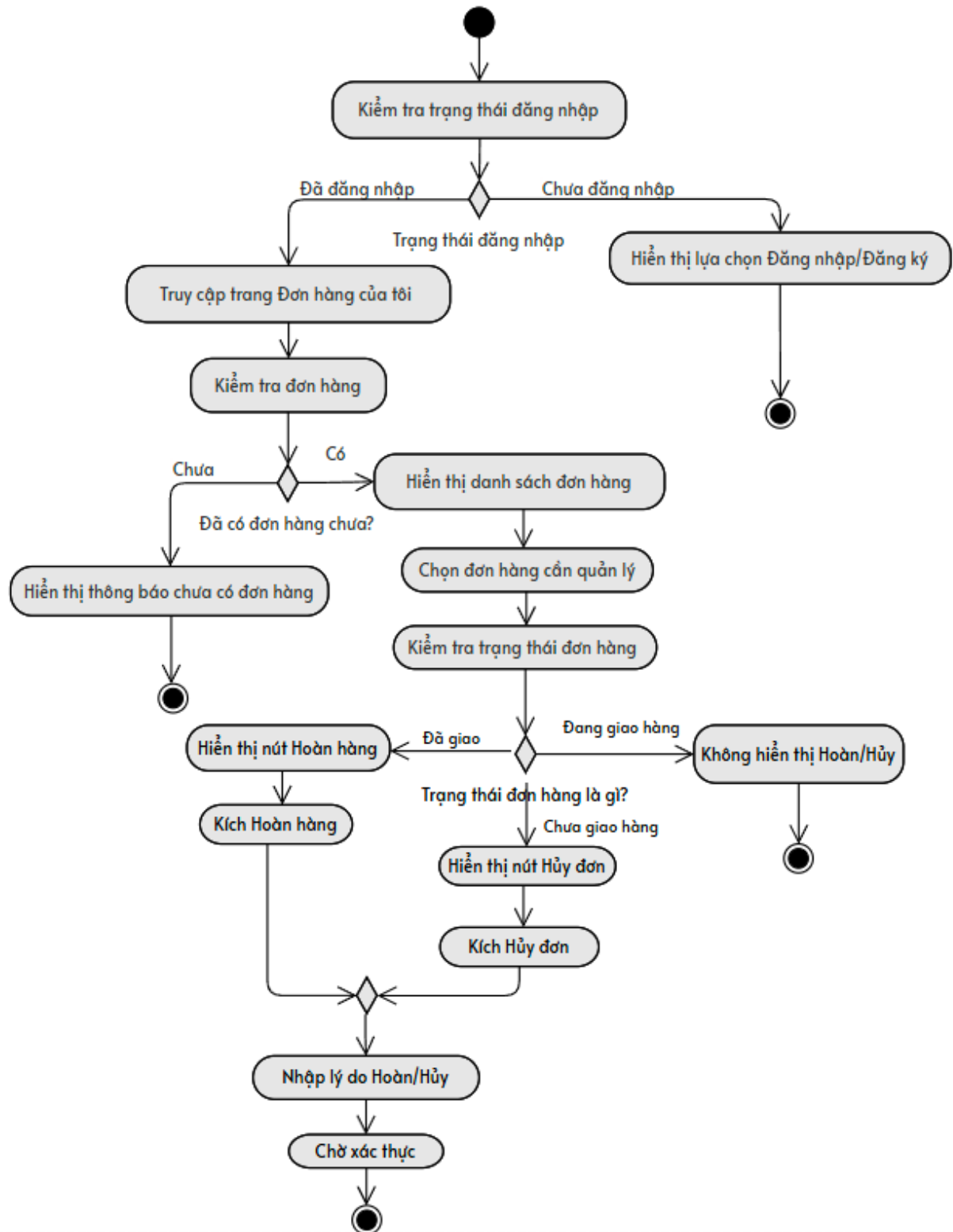
Hình 3.8 Biểu đồ hoạt động use case Quản lý giỏ hàng

3.4.4.5. Use case Đặt hàng



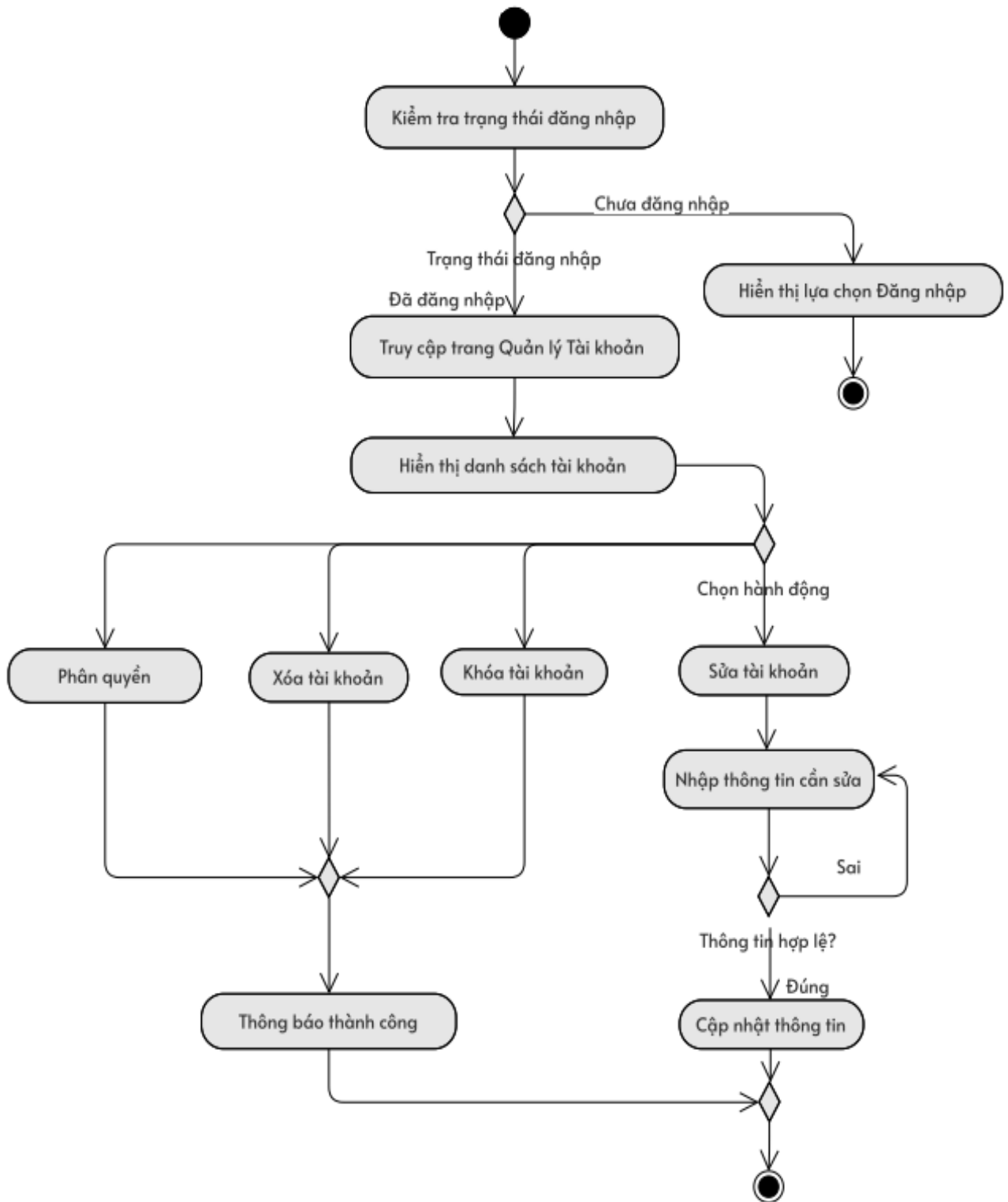
Hình 3.9 Biểu đồ hoạt động use case Đặt hàng

3.4.4.6. Use case Quản lý đơn hàng cá nhân



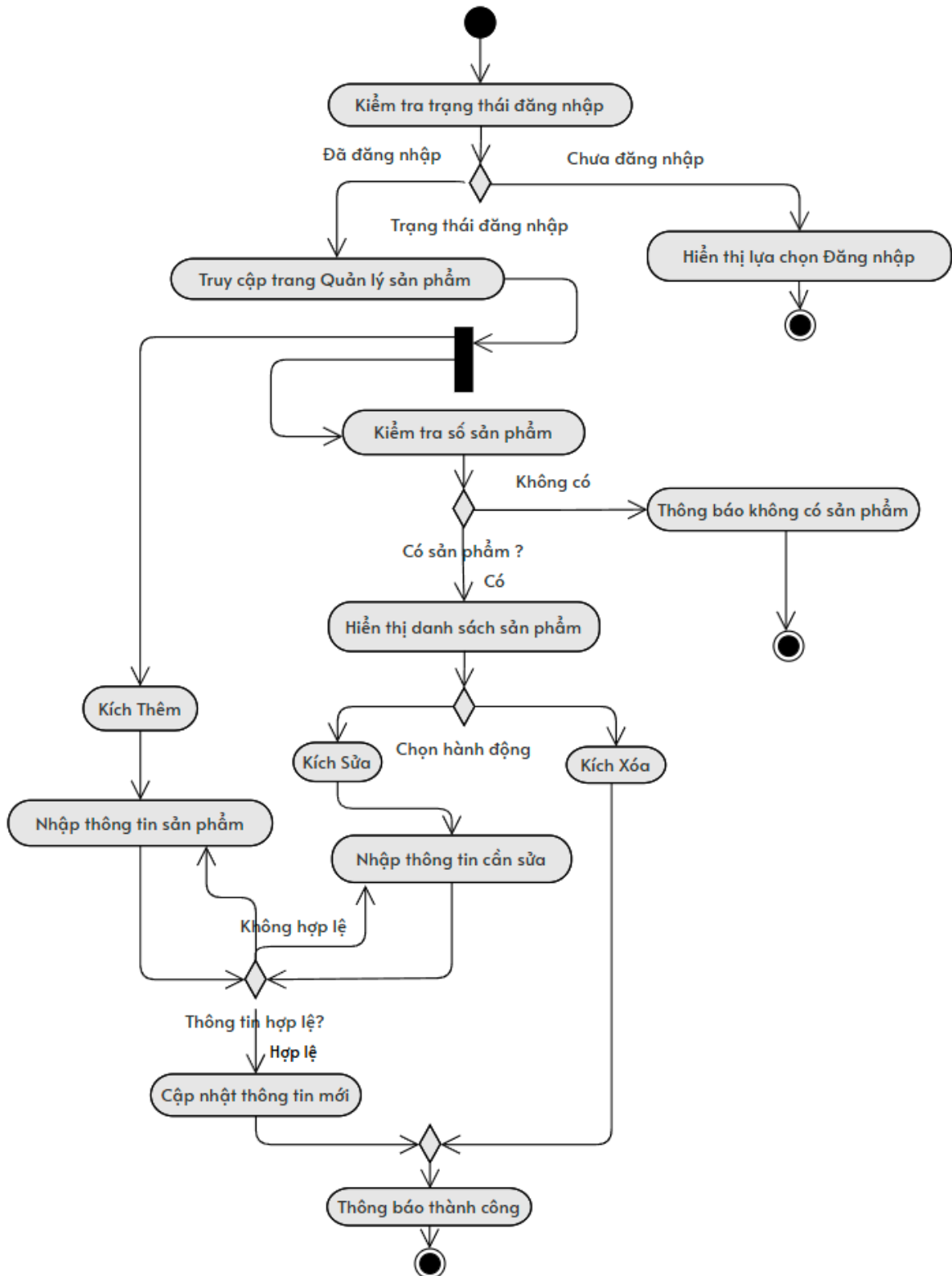
Hình 3.10 Biểu đồ hoạt động use case Quản lý đơn hàng cá nhân

3.4.4.7. Use case Quản lý tài khoản



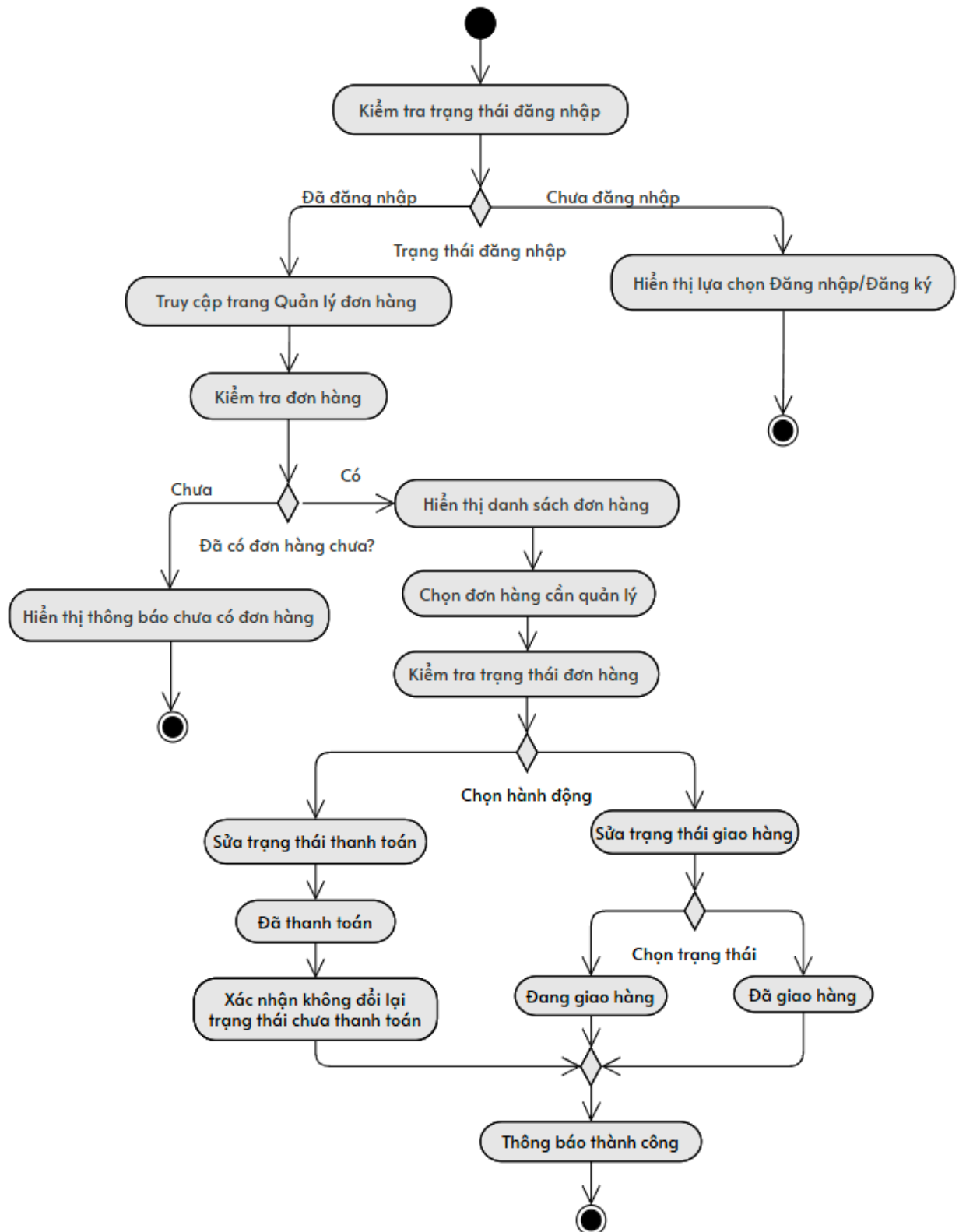
Hình 3.11 Biểu đồ hoạt động use case Quản lý tài khoản

3.4.4.8. Use case Quản lý sản phẩm



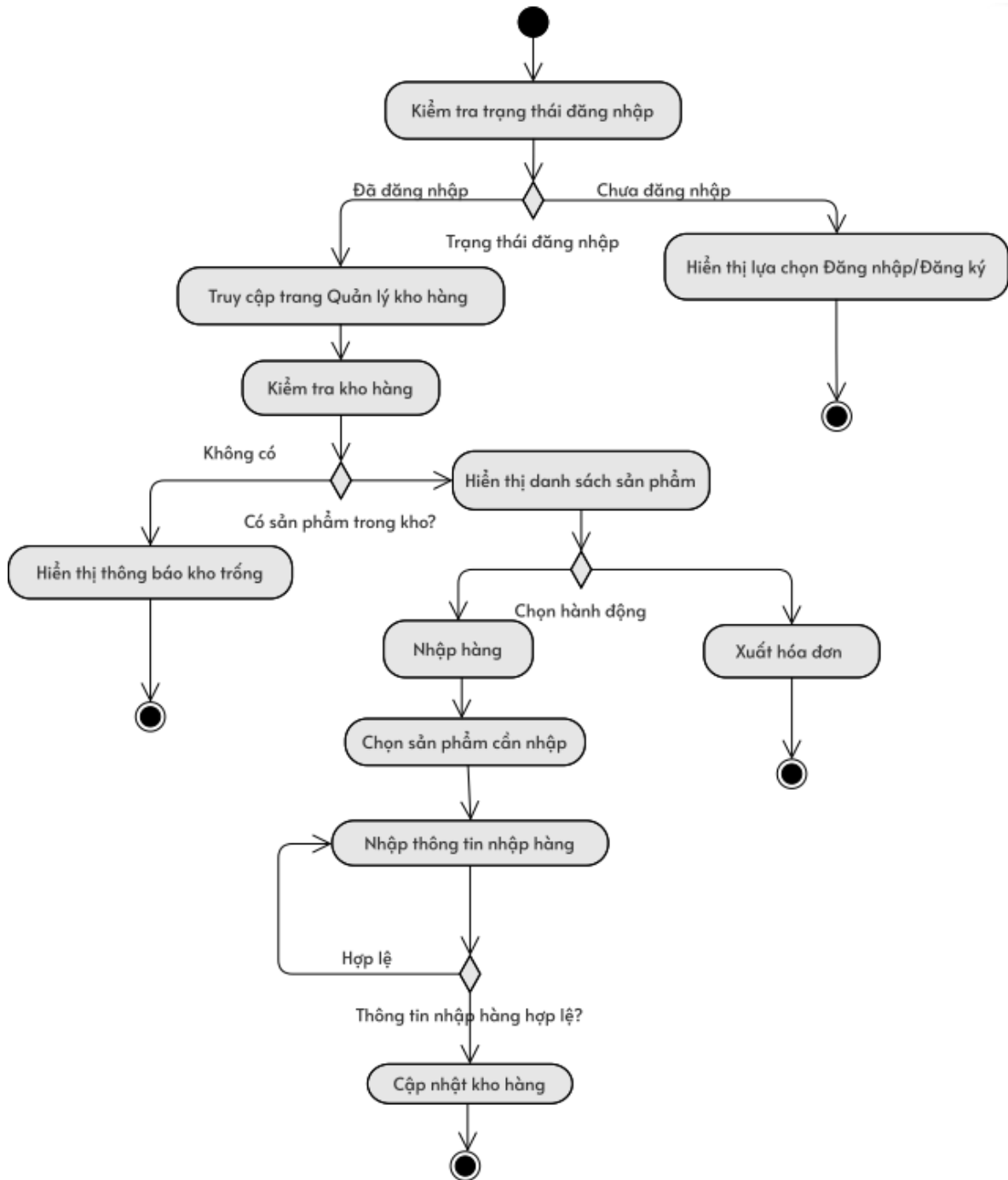
Hình 3.12 Biểu đồ hoạt động use case Quản lý sản phẩm

3.4.4.9. Use case Quản lý đơn hàng



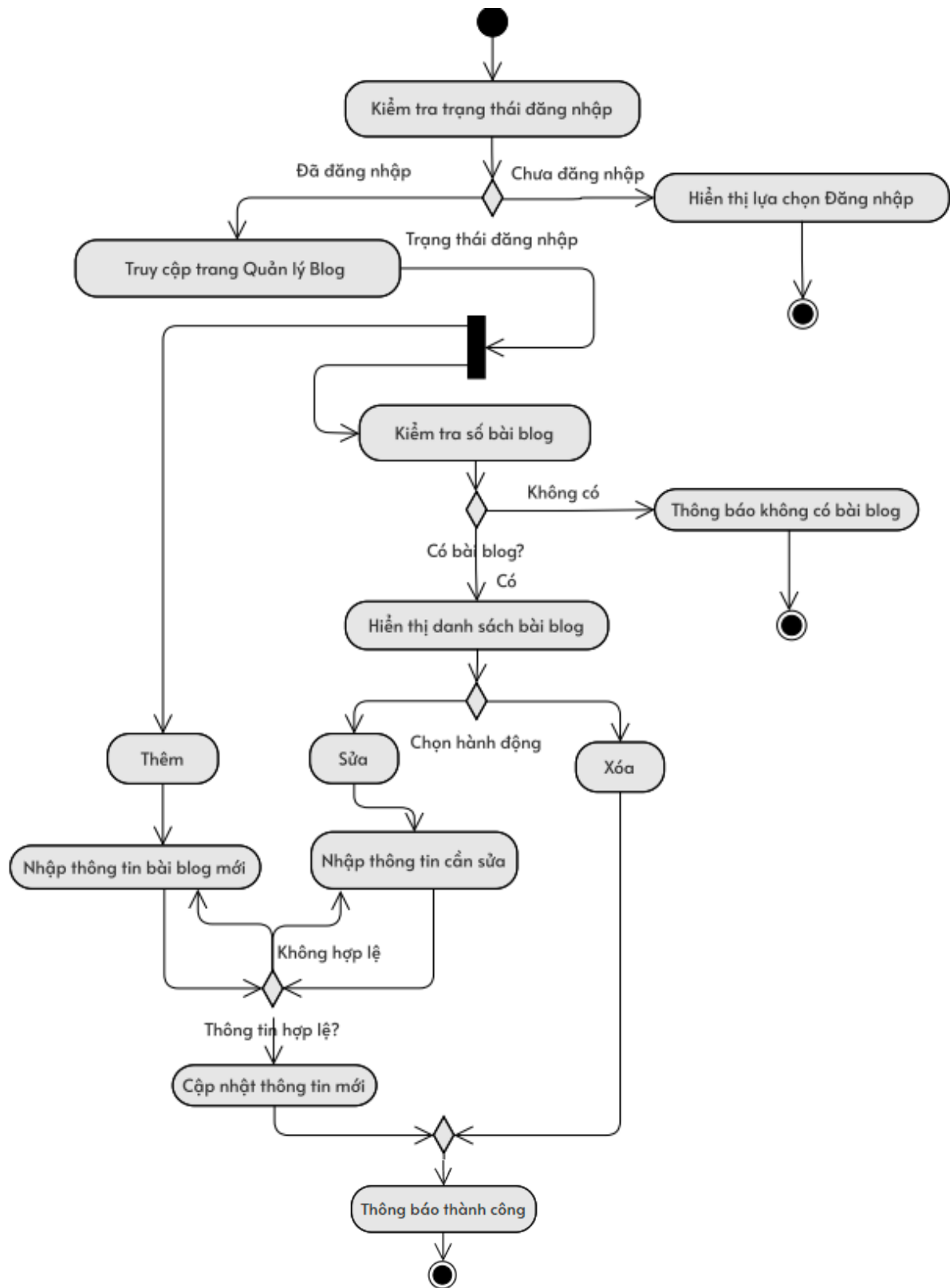
Hình 3.13 Biểu đồ hoạt động use case Quản lý đơn hàng

3.4.4.10. Use case Quản lý kho hàng



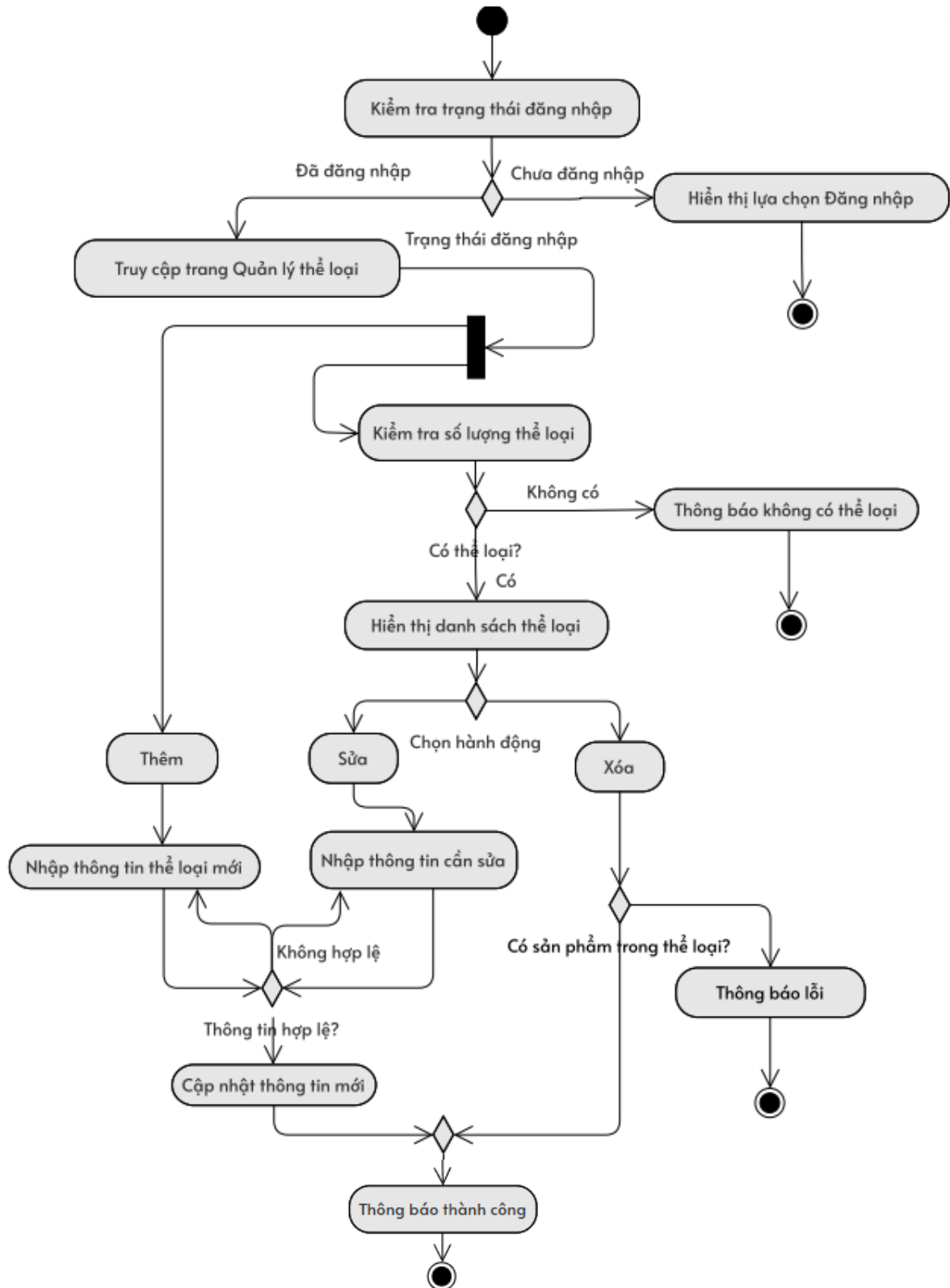
Hình 3.14 Biểu đồ hoạt động use case Quản lý kho hàng

3.4.4.11. Use case Quản lý blog



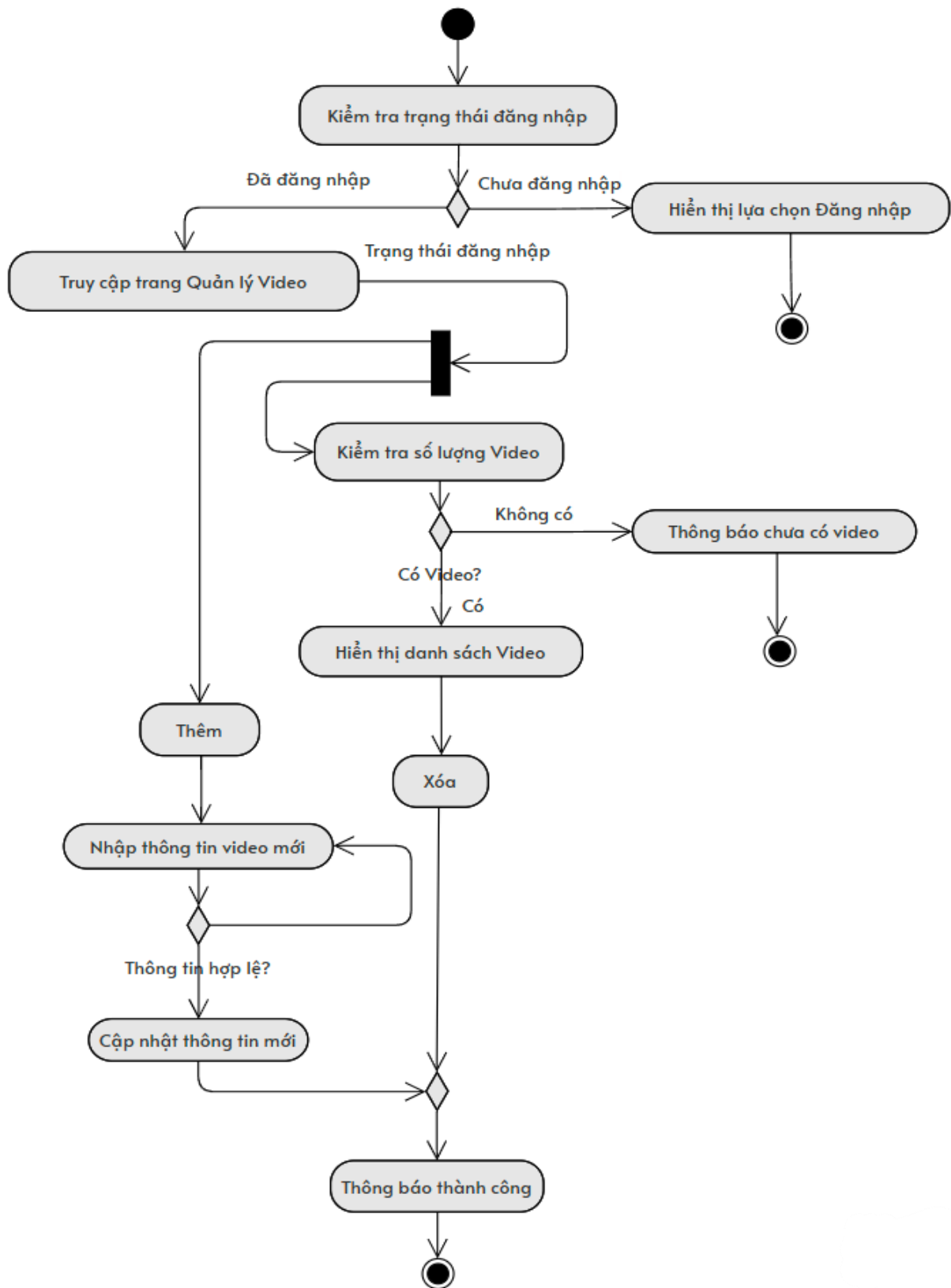
Hình 3.15 Biểu đồ hoạt động use case Quản lý Blog

3.4.4.12. Use case Quản lý thẻ loại



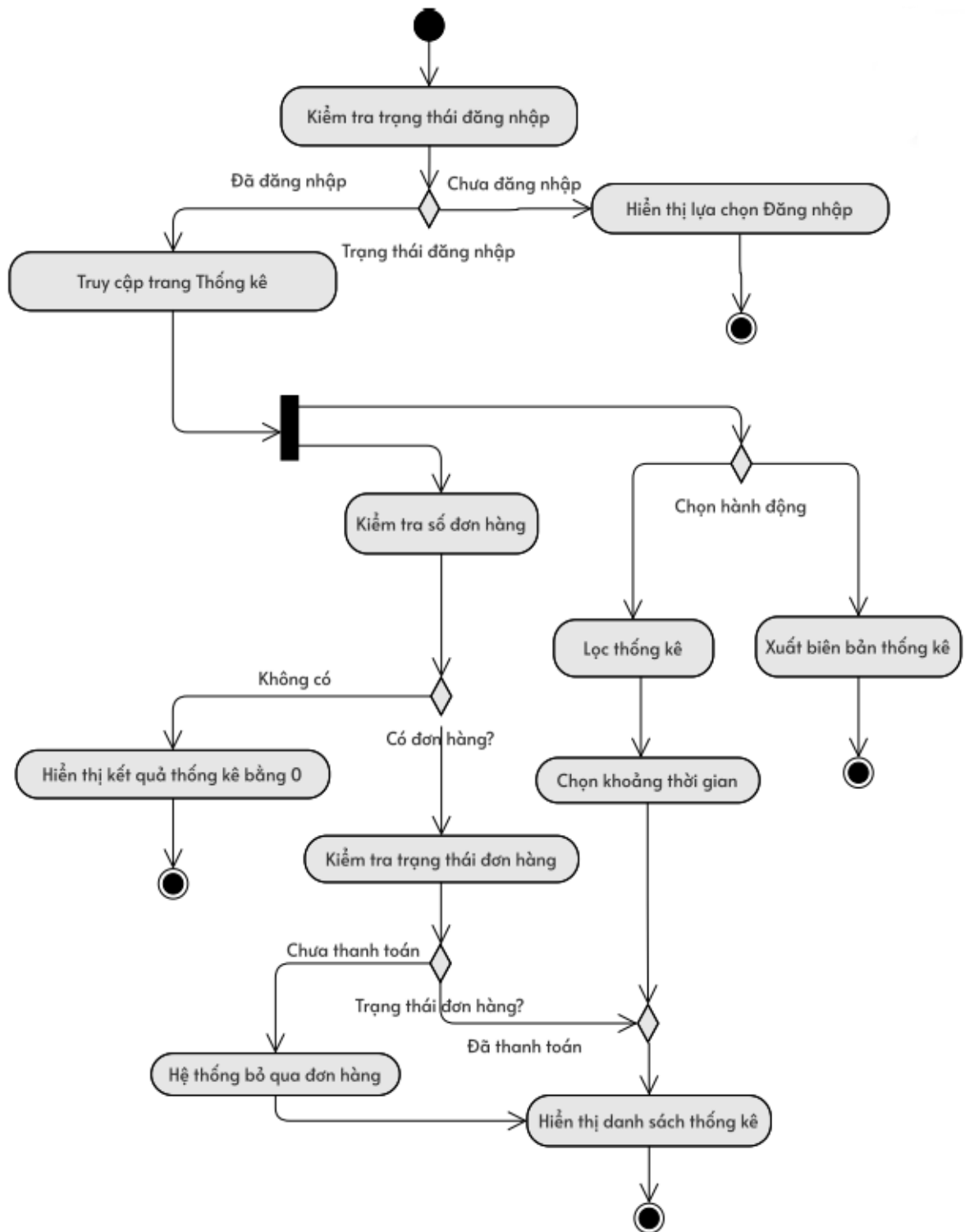
Hình 3.16 Biểu đồ hoạt động use case Quản lý thẻ loại

3.4.4.13. Use case Quản lý video



Hình 3.17 Biểu đồ hoạt động use case Quản lý video

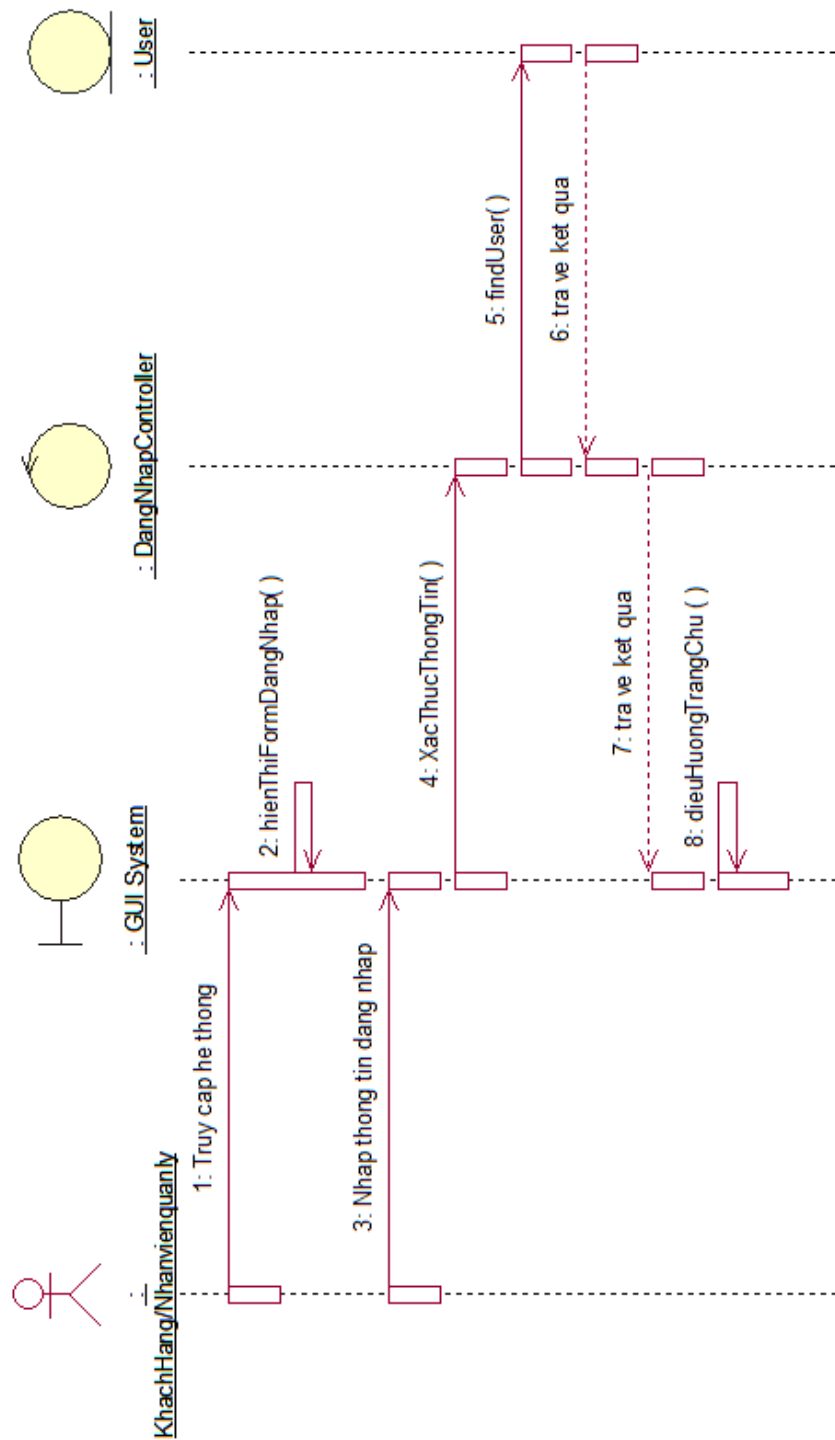
3.4.4.14. Use case Thống kê



Hình 3.18 Biểu đồ hoạt động use case Thống kê

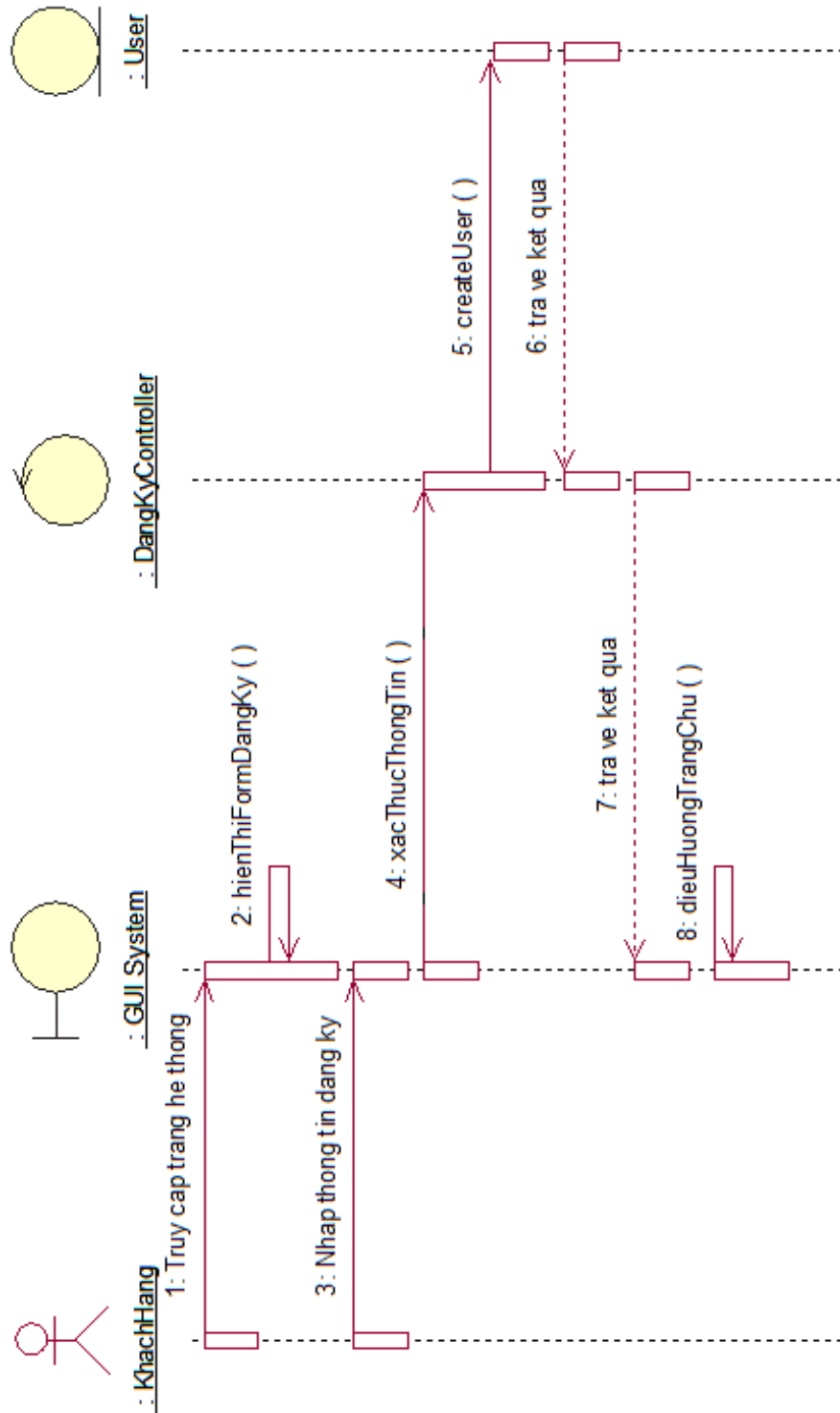
3.4.5. Biểu đồ trình tự

3.4.5.1. Use case Đăng nhập



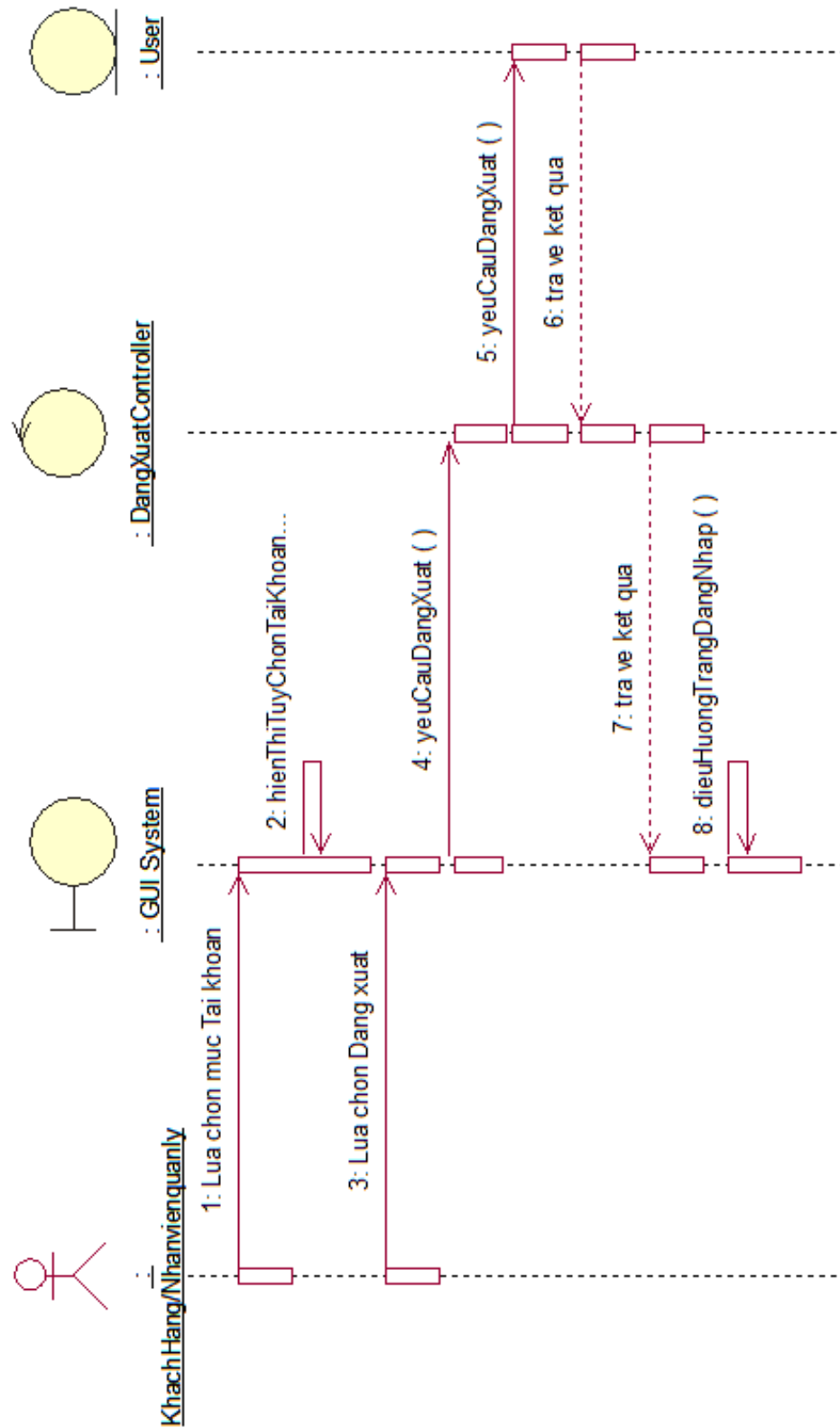
Hình 3.19 Biểu đồ trình tự use case Đăng nhập

3.4.5.2. Use case Đăng ký



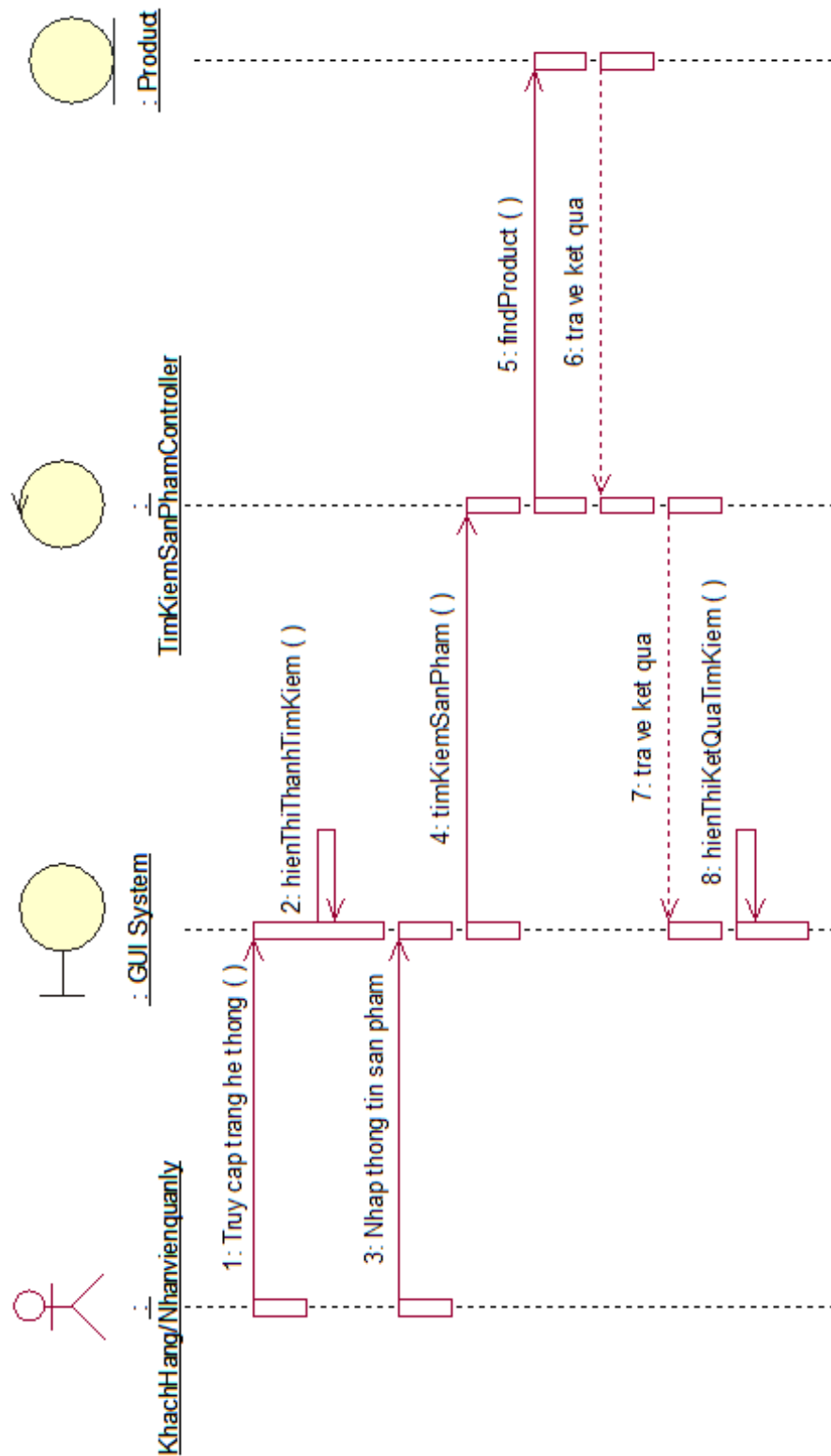
Hình 3.20 Biểu đồ trình tự use case Đăng ký

3.4.5.3. Use case Đăng xuất



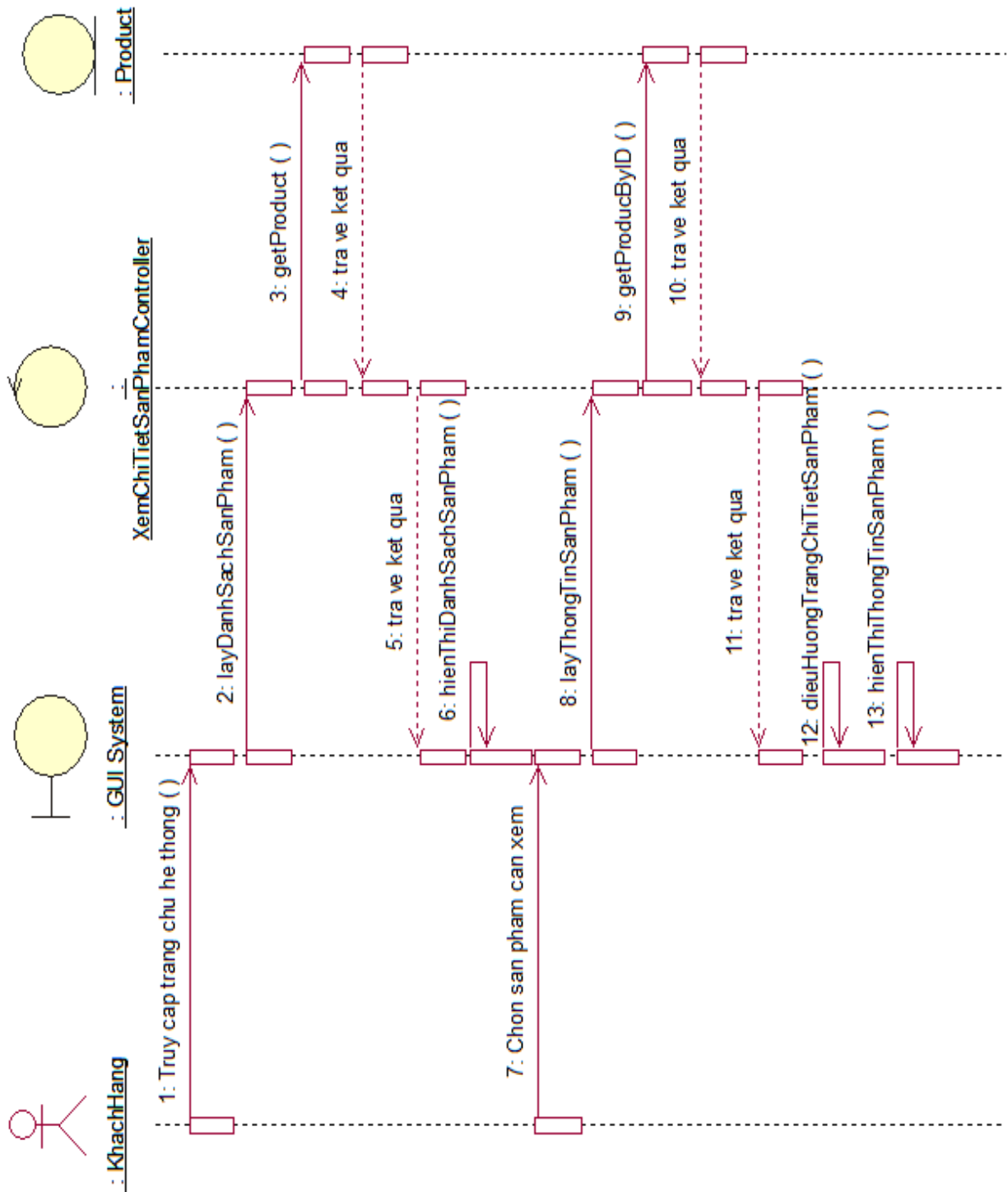
Hình 3.21 Biểu đồ trình tự use case Đăng xuất

3.4.5.4. Use case Tìm kiếm sản phẩm



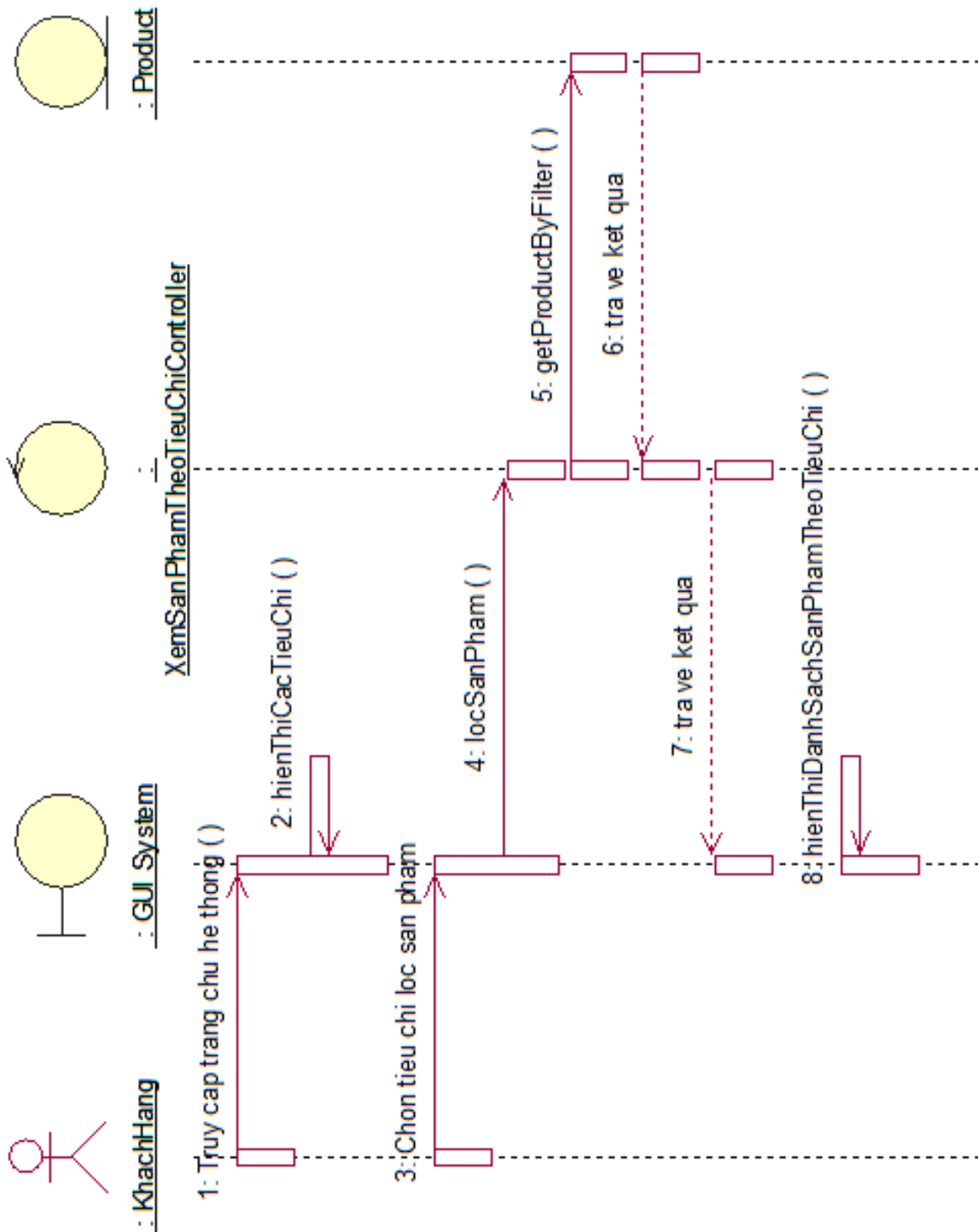
Hình 3.22 Biểu đồ trình tự use case Tìm kiếm sản phẩm

3.4.5.5. Use case Xem chi tiết sản phẩm



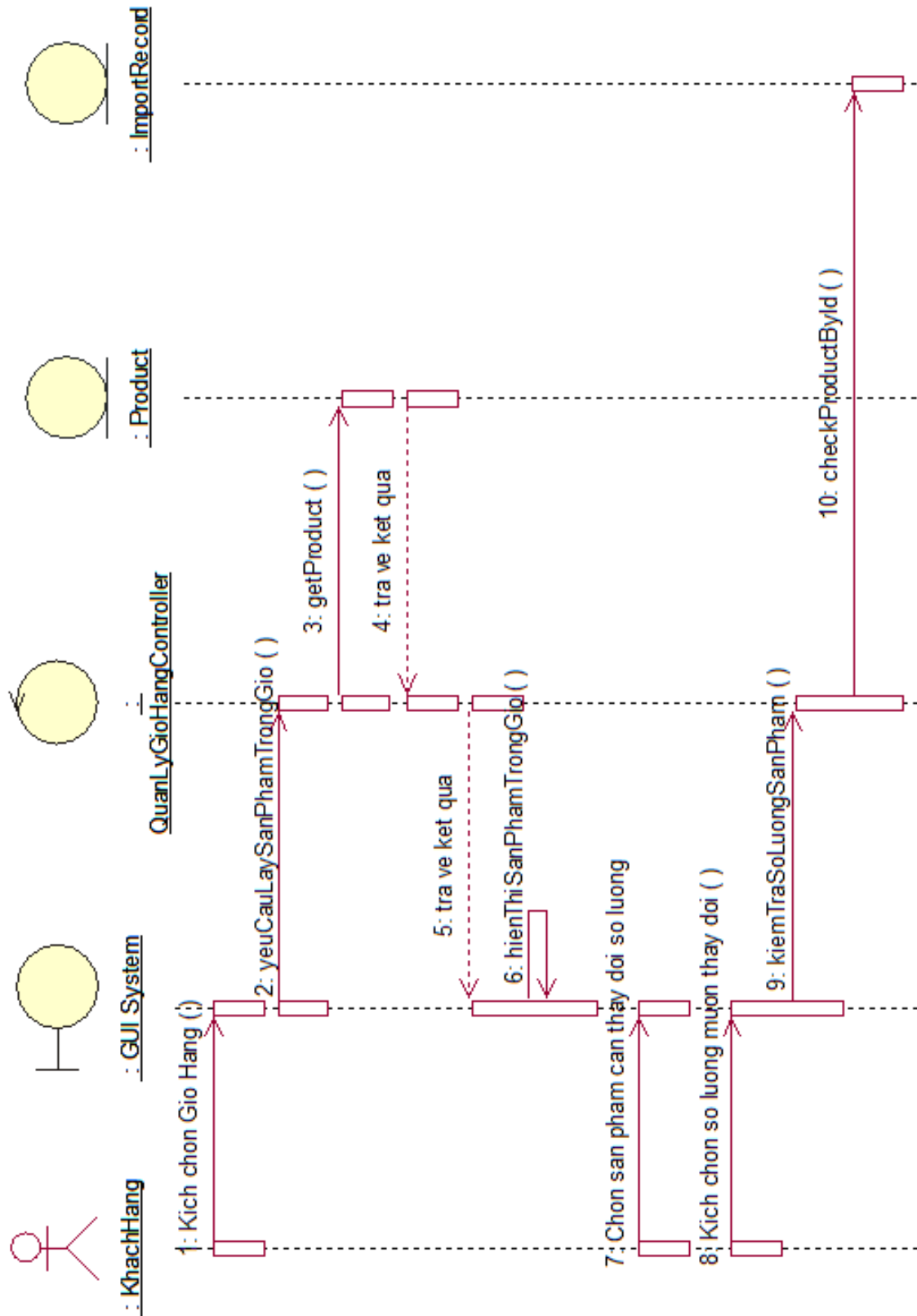
Hình 3.23 Biểu đồ trình tự use case Xem chi tiết sản phẩm

3.4.5.6. Use case Xem sản phẩm theo tiêu chí

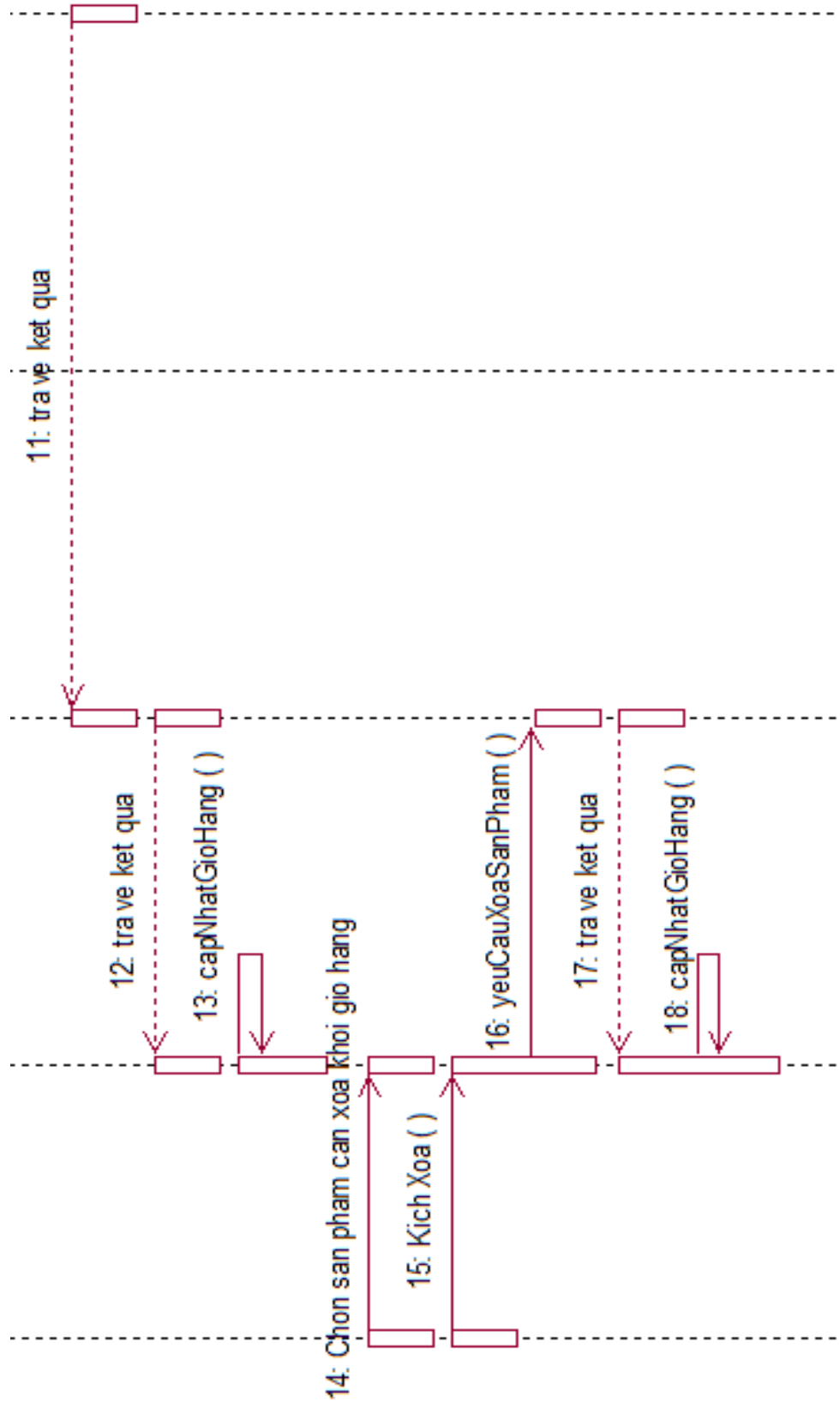


Hình 3.24 Biểu đồ trình tự use case Xem sản phẩm theo tiêu chí

3.4.5.7. Use case Quản lý giỏ hàng

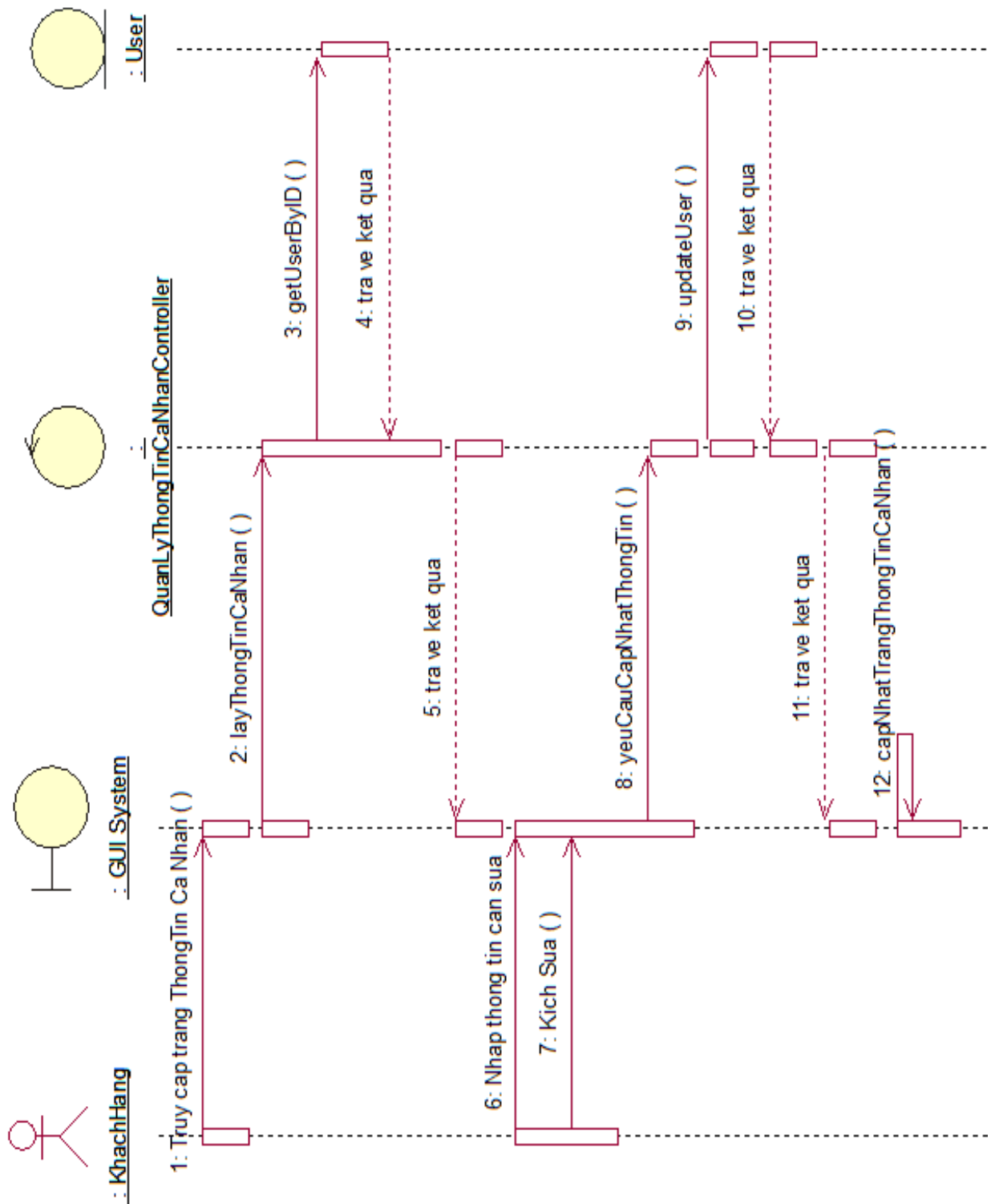


Hình 3.25 Biểu đồ trình tự use case Quản lý giỏ hàng (1)



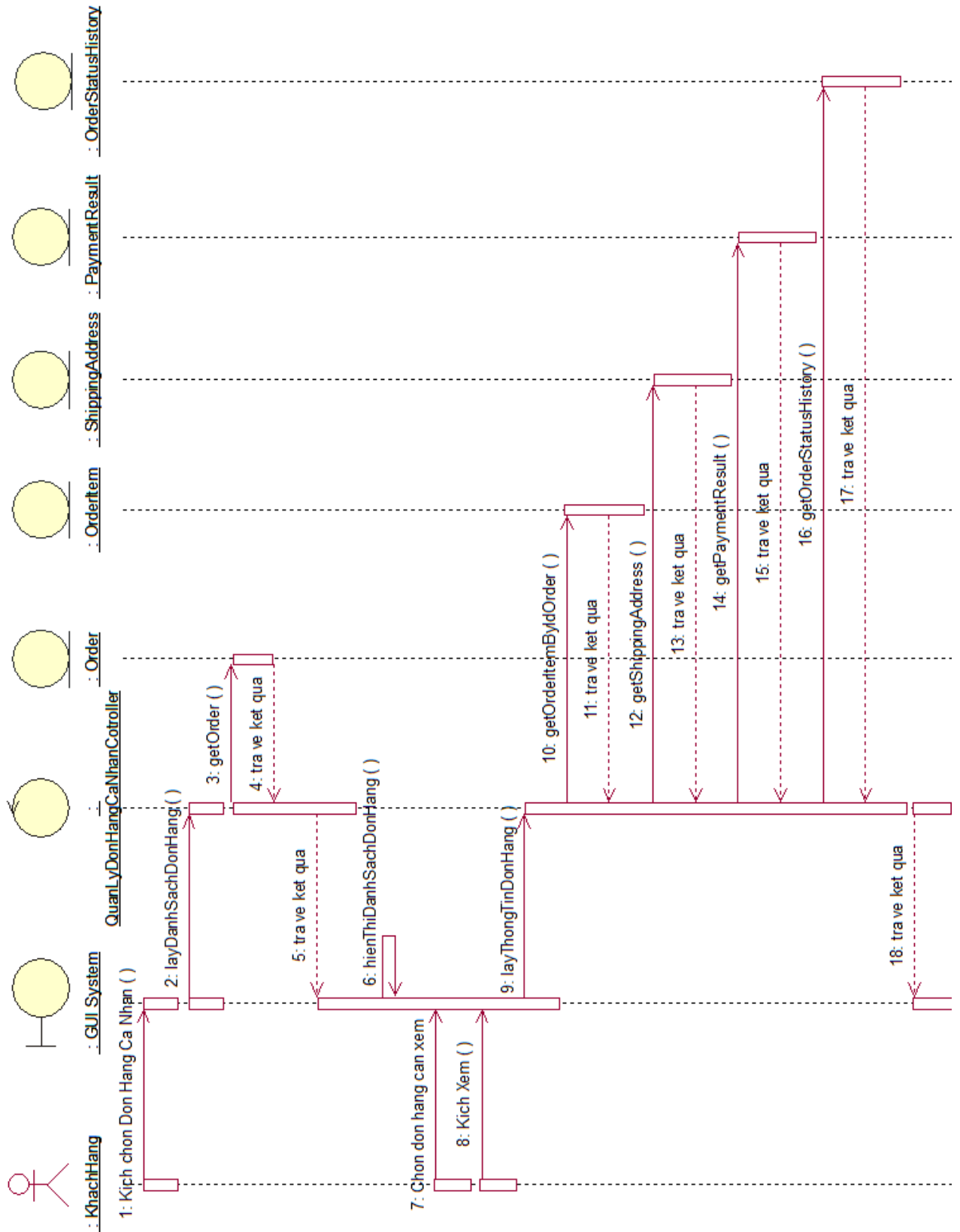
Hình 3.26 Biểu đồ trình tự use case Quản lý giỏ hàng (2)

3.4.5.8. Use case Quản lý thông tin cá nhân

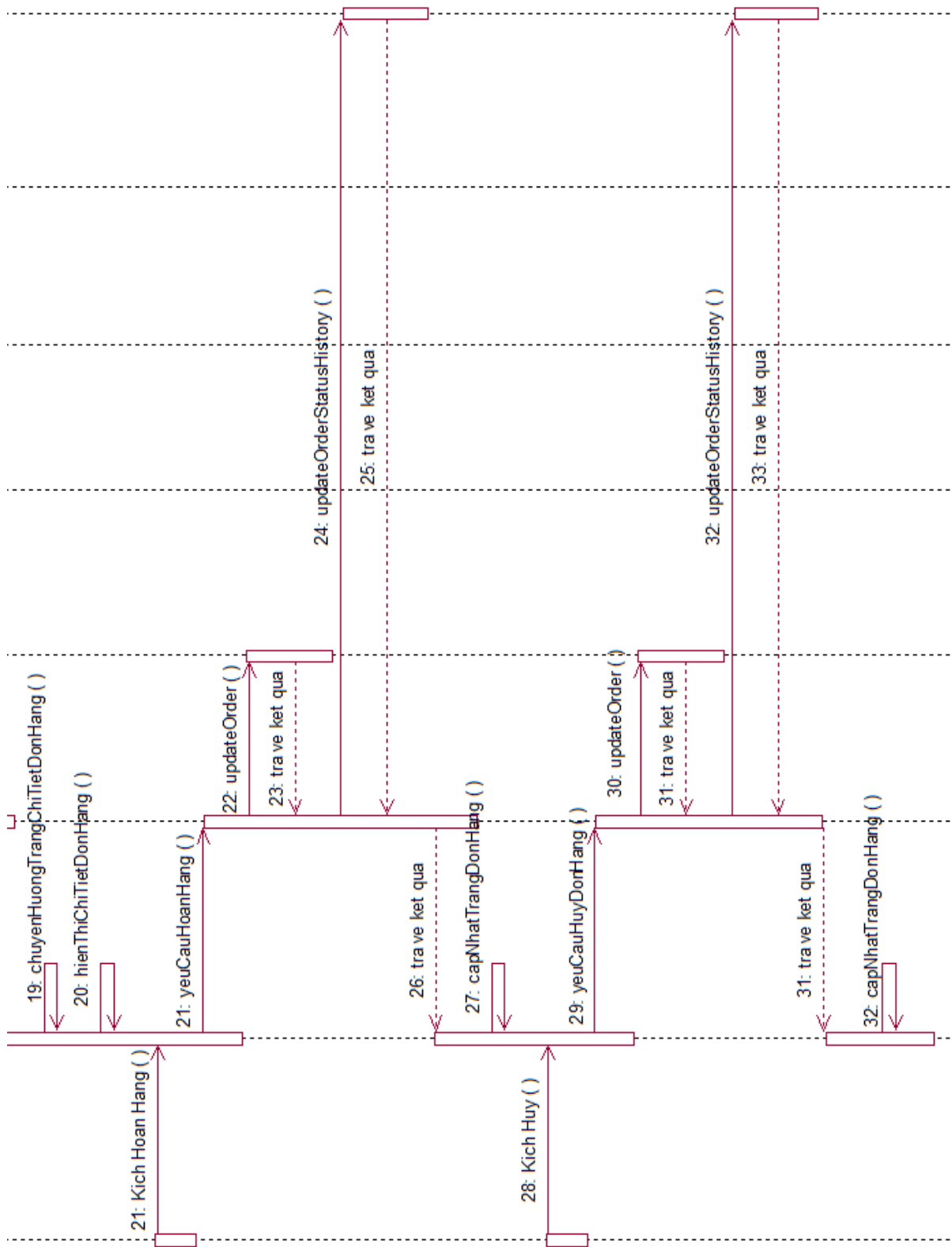


Hình 3.27 Biểu đồ trình tự use case Quản lý thông tin cá nhân

3.4.5.9. Use case Quản lý đơn hàng cá nhân

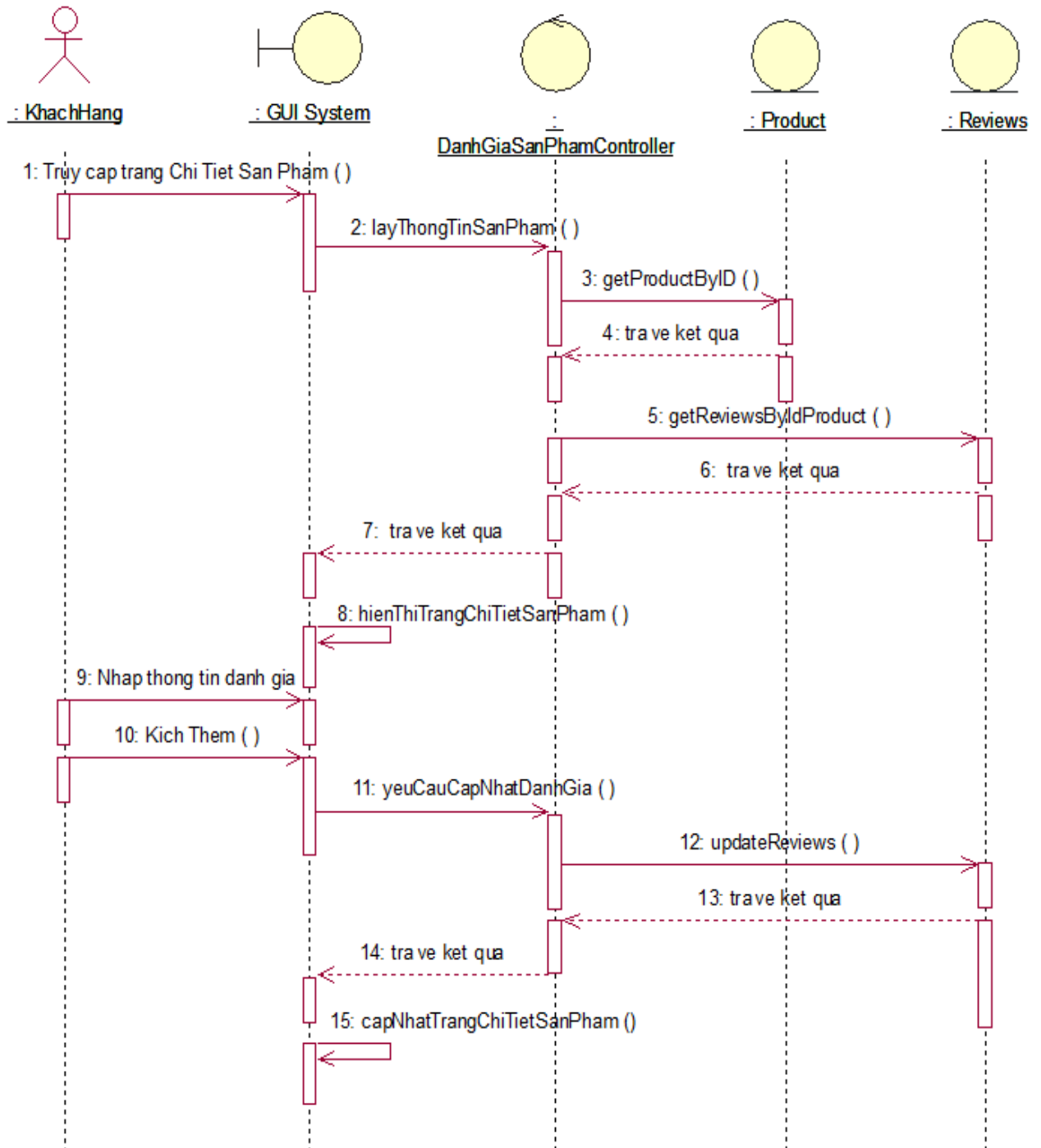


Hình 3.28 Biểu đồ trình tự use case Quản lý đơn hàng cá nhân (1)



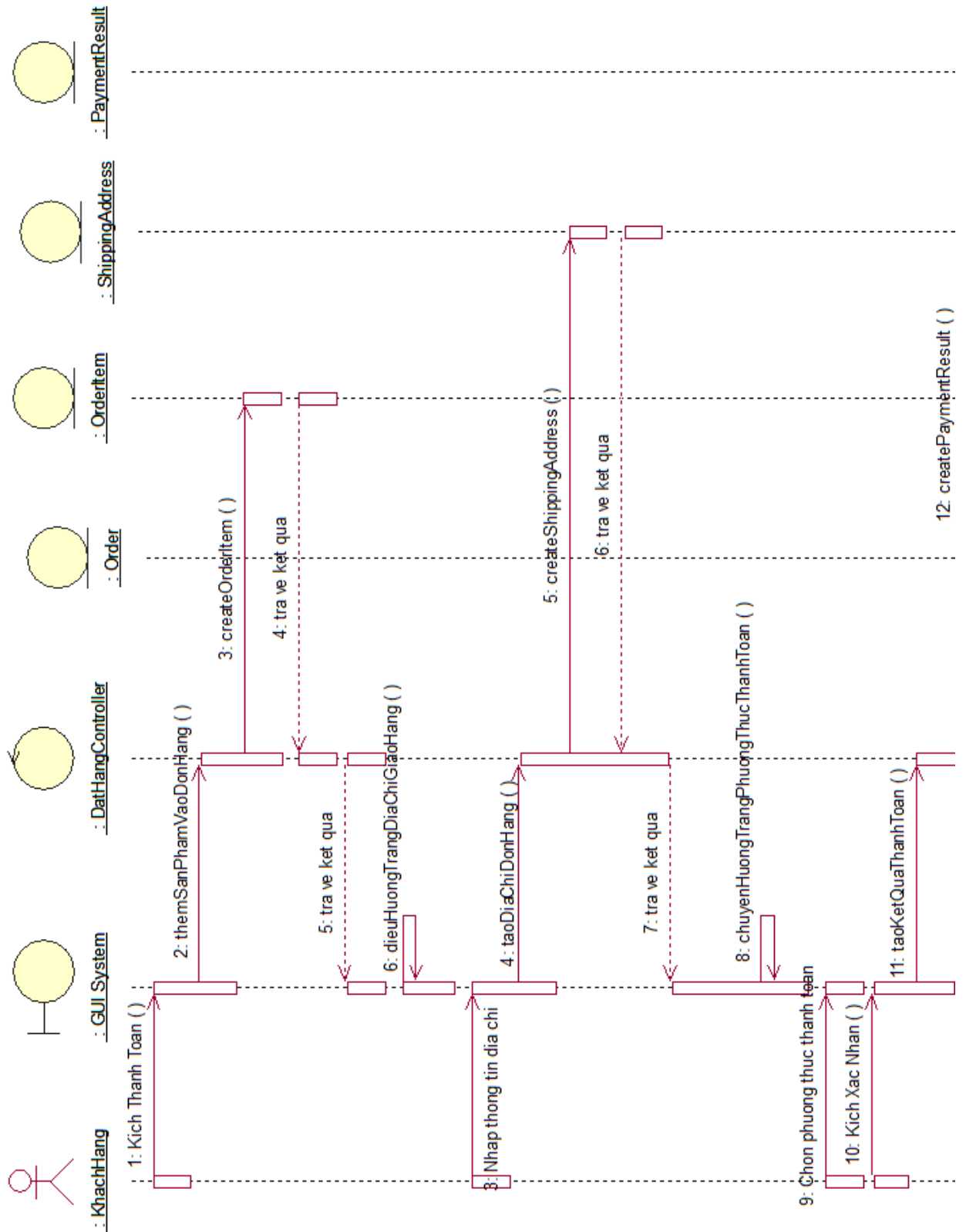
Hình 3.29 Biểu đồ trình tự use case Quản lý đơn hàng cá nhân (2)

3.4.5.10. Use case Đánh giá sản phẩm

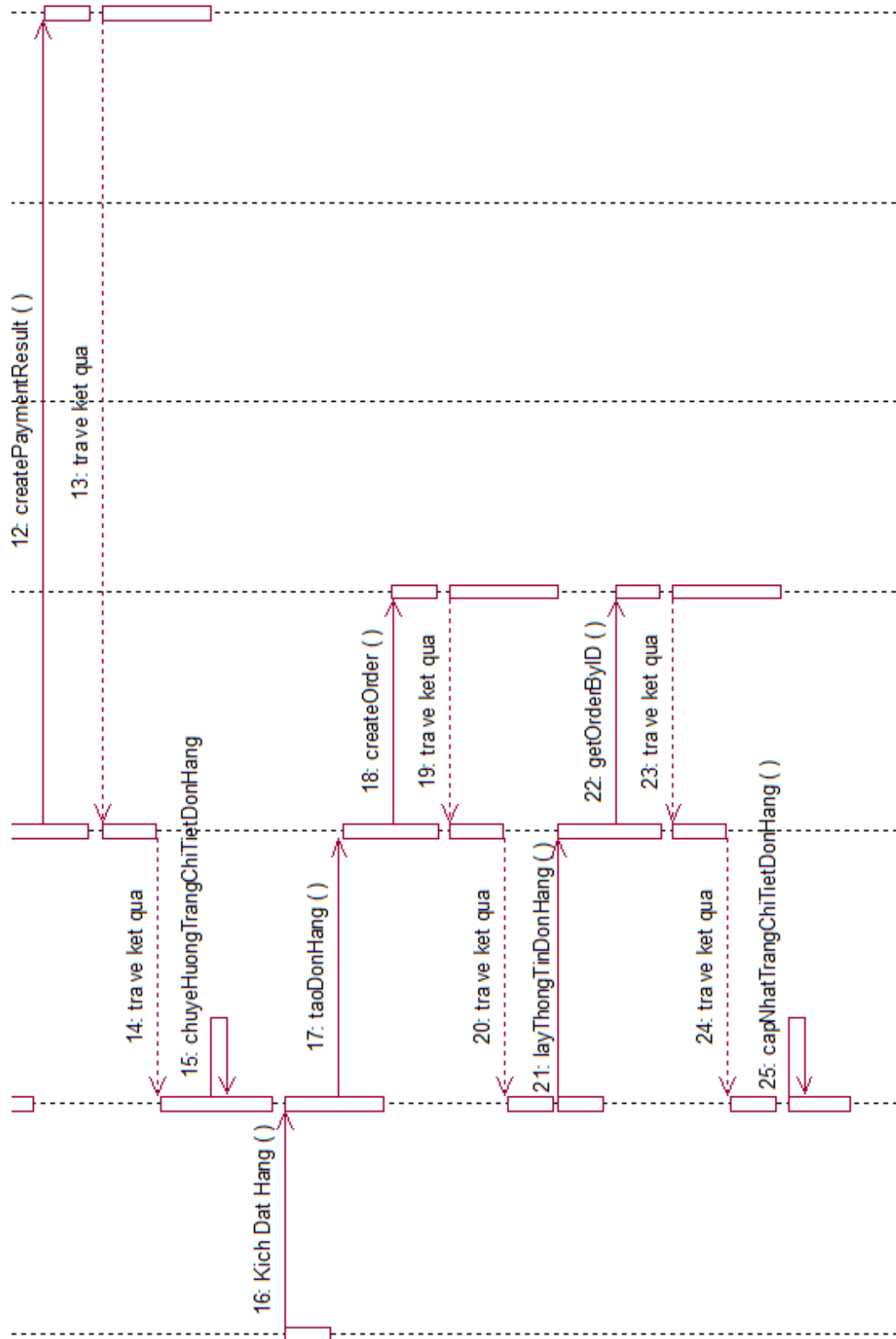


Hình 3.30 Biểu đồ trình tự use case Đánh giá sản phẩm

3.4.5.11. Use case Đặt hàng

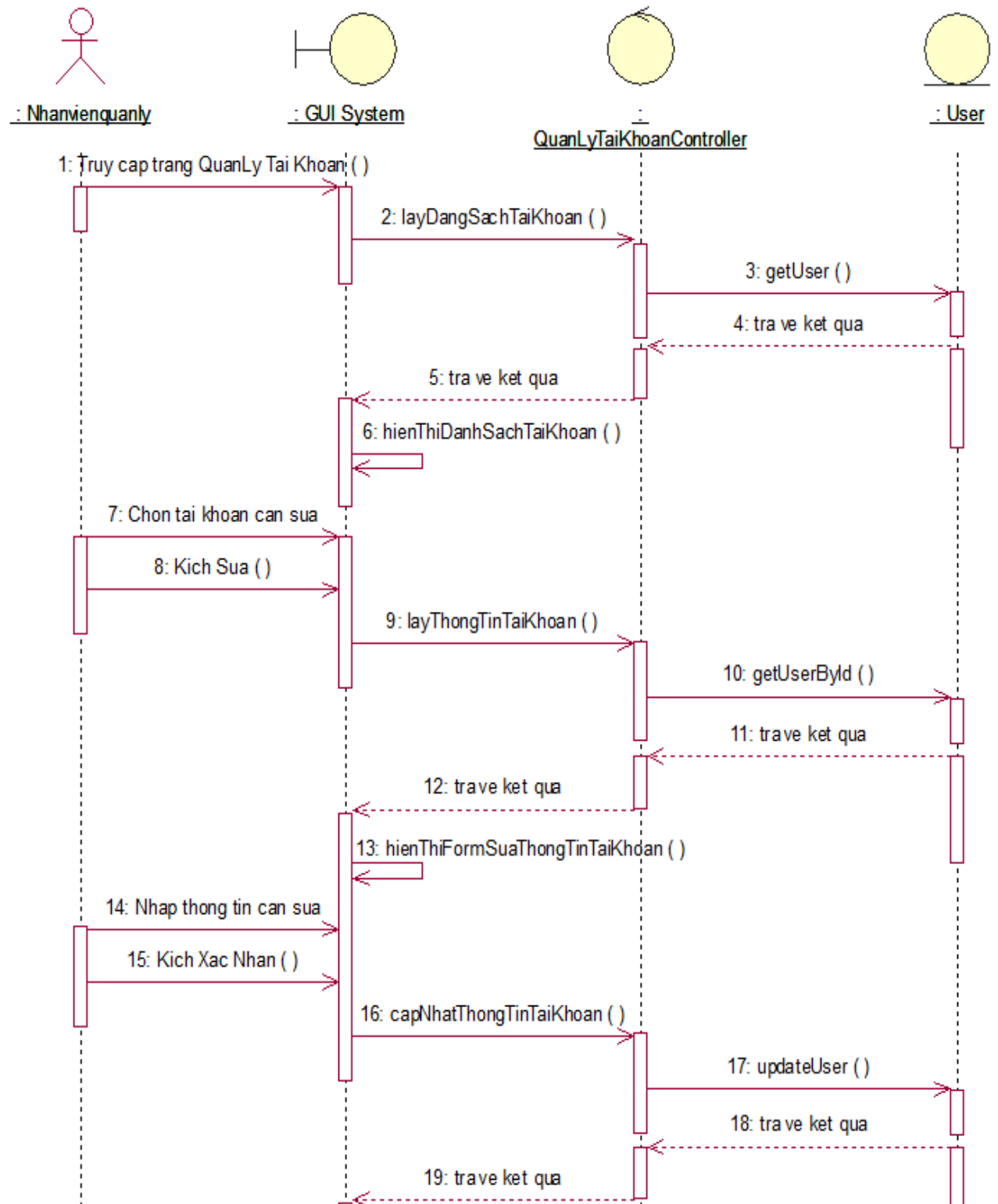


Hình 3.31 Biểu đồ trình tự use case Đặt hàng (1)

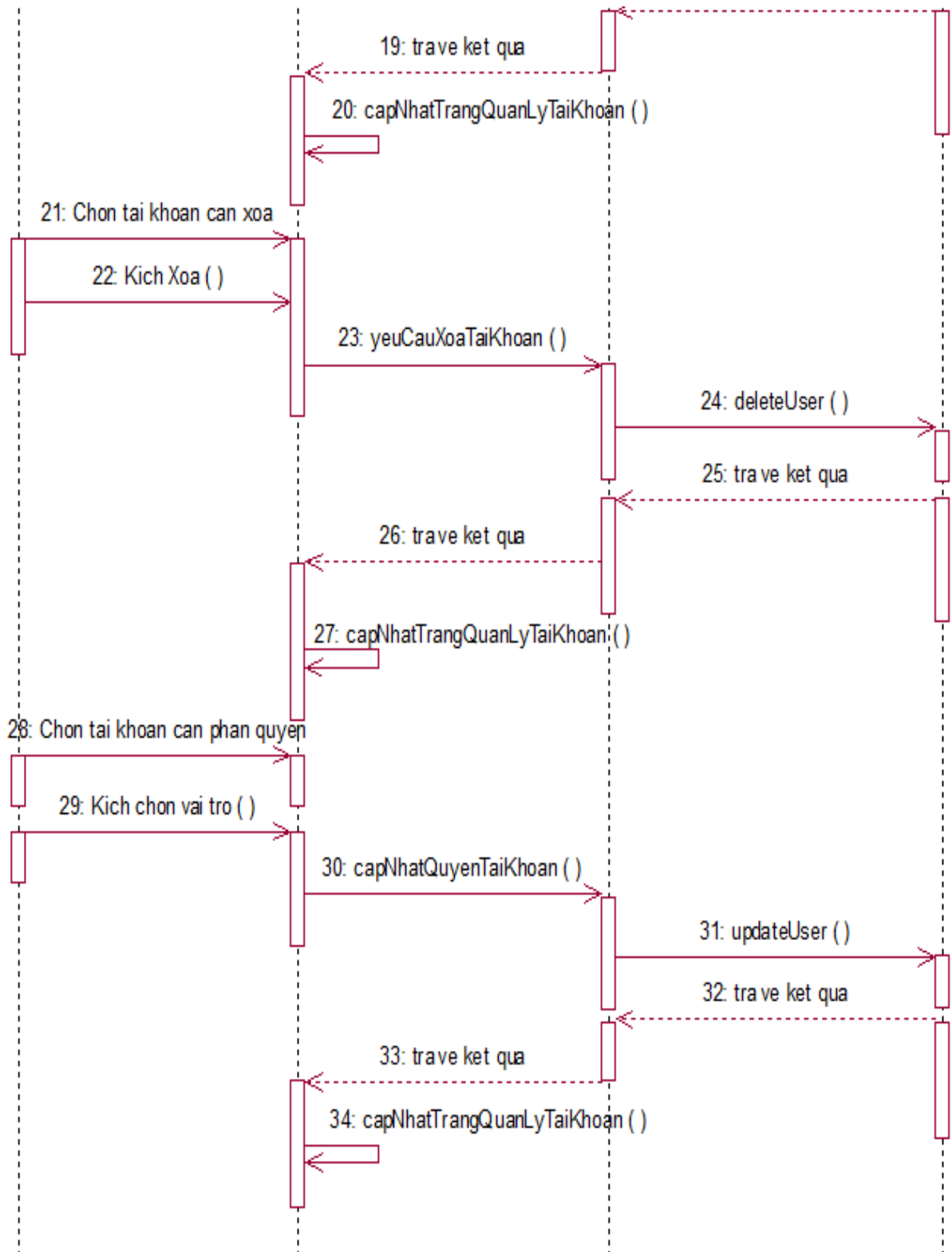


Hình 3.32 Biểu đồ trình tự use case Đặt hàng (2)

3.4.5.12. Use case Quản lý tài khoản

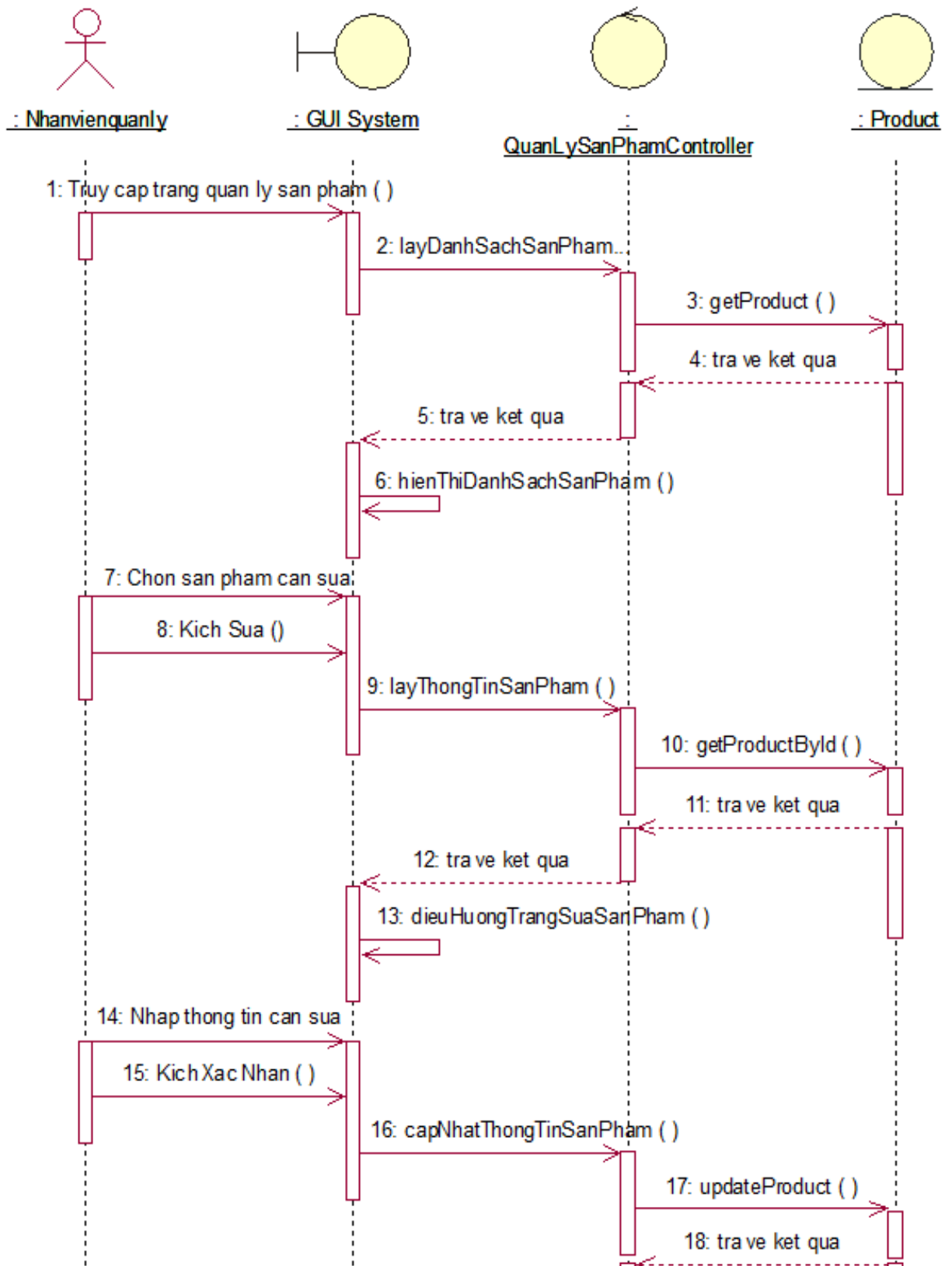


Hình 3.33 Biểu đồ trình tự use case Quản lý tài khoản (1)

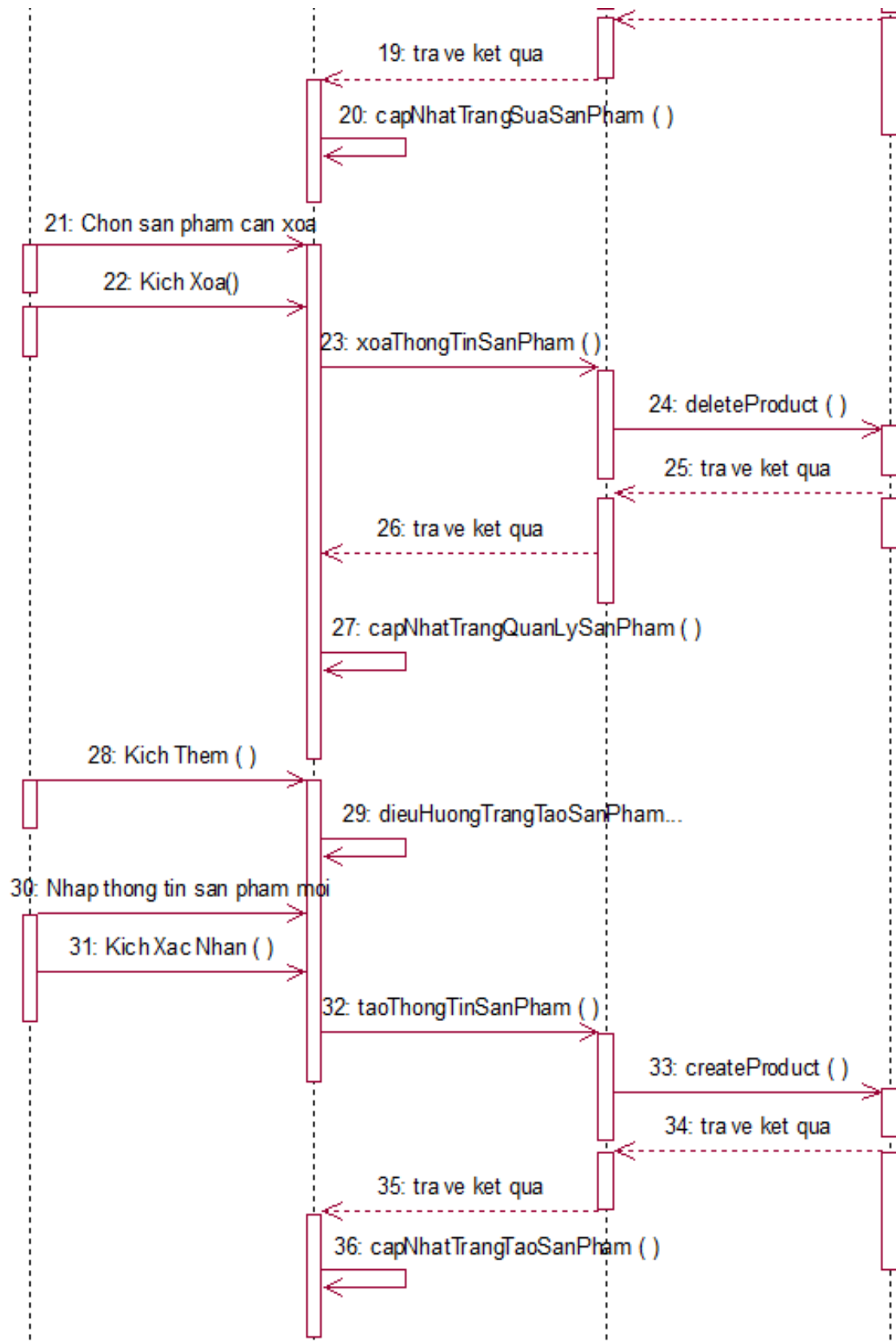


Hình 3.34 Biểu đồ trình tự use case Quản lý tài khoản (2)

3.4.5.13. Use case Quản lý sản phẩm

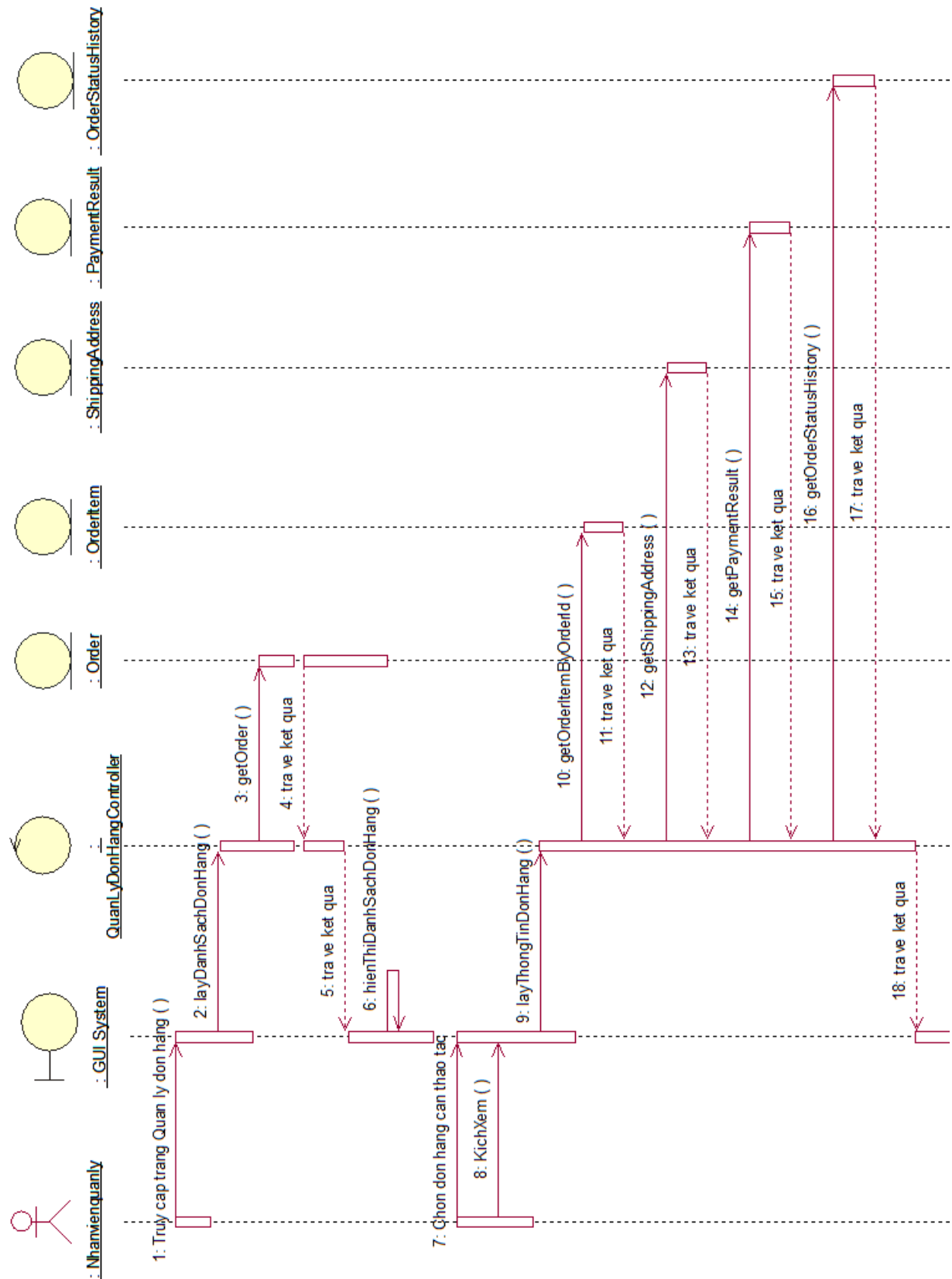


Hình 3.35 Biểu đồ trình tự use case Quản lý sản phẩm (1)

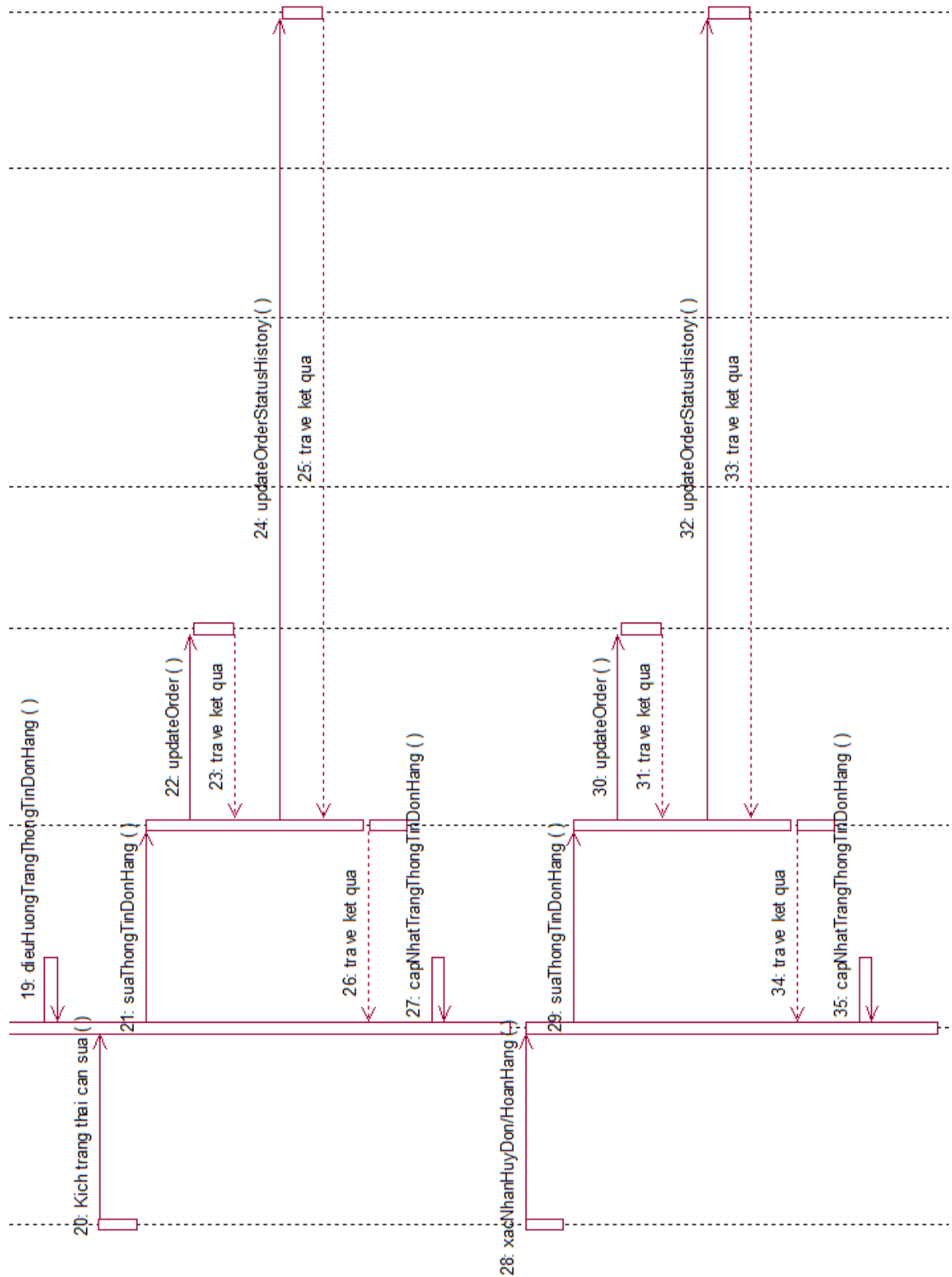


Hình 3.36 Biểu đồ trình tự use case Quản lý sản phẩm (2)

3.4.5.14. Use case Quản lý đơn hàng

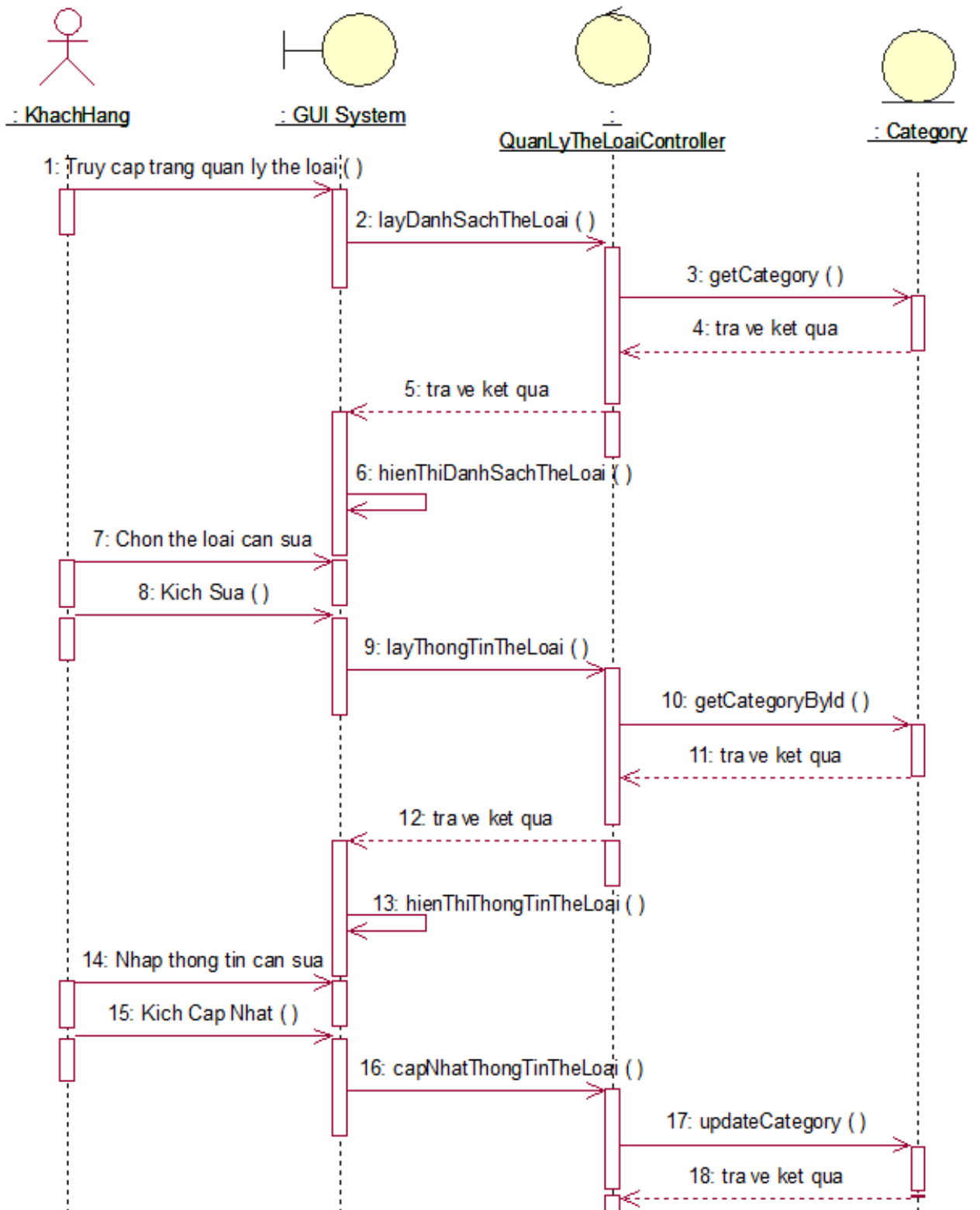


Hình 3.37 Biểu đồ trình tự use case Quản lý đơn hàng (1)

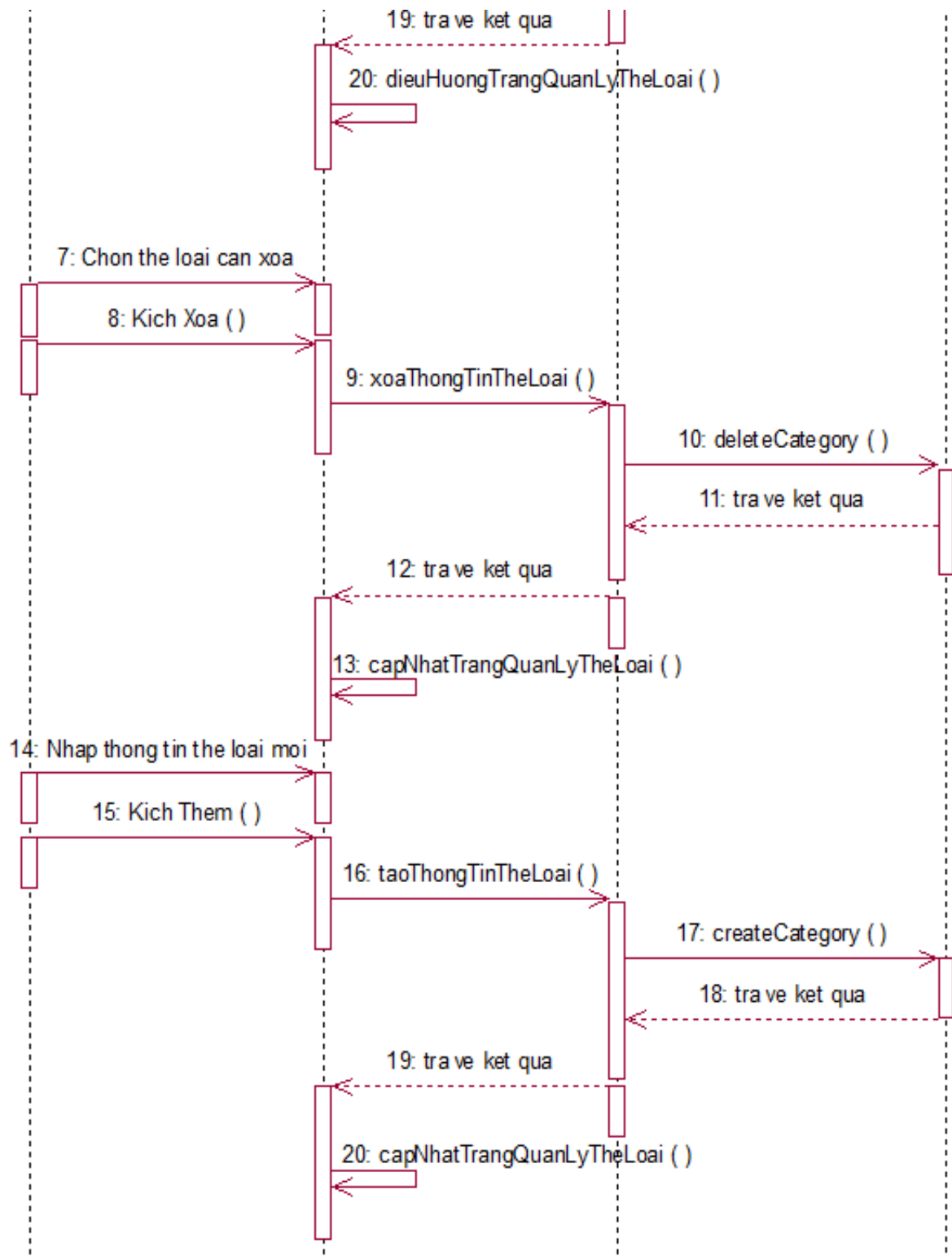


Hình 3.38 Biểu đồ trình tự use case Quản lý đơn hàng (2)

3.4.5.15. Use case Quản lý thể loại

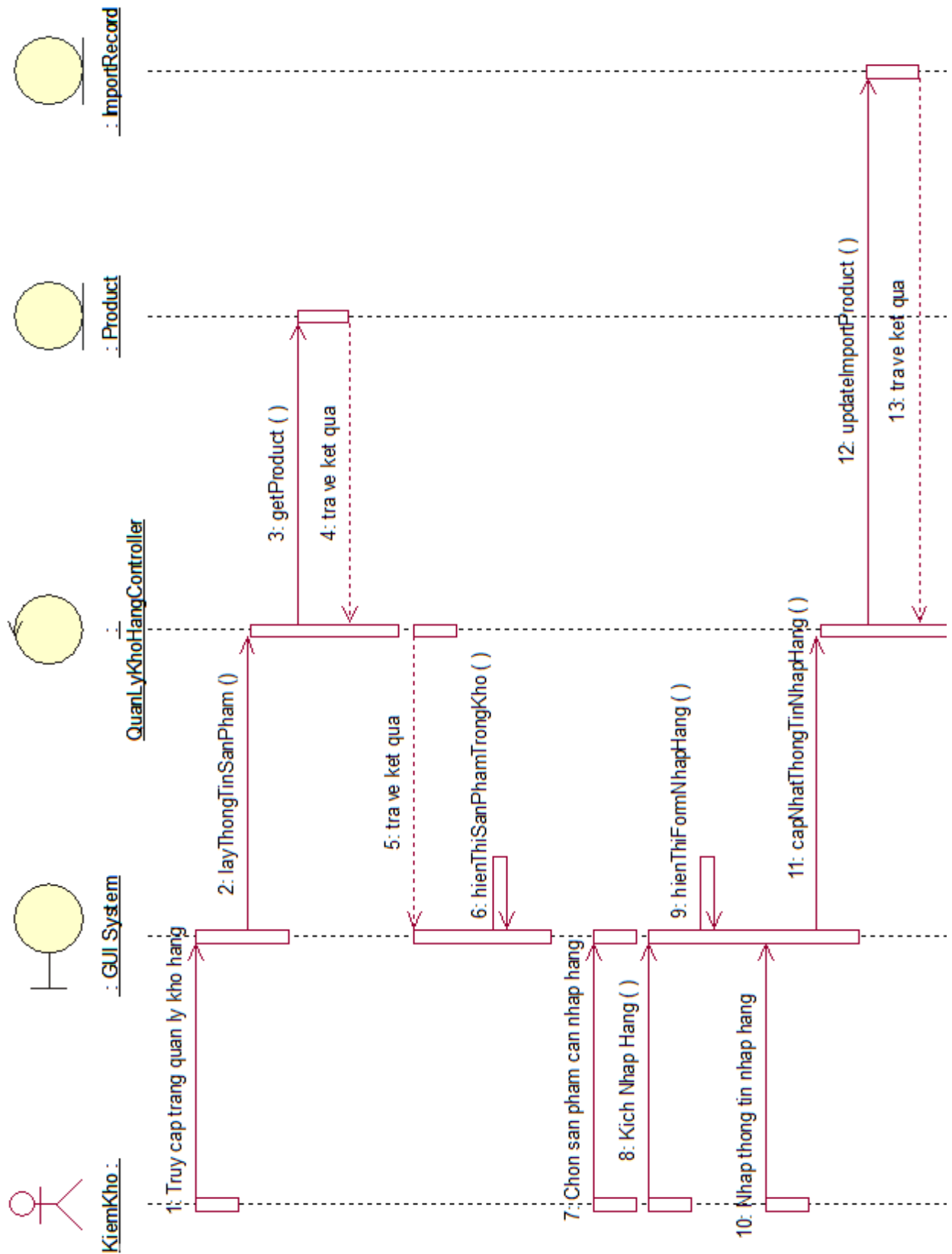


Hình 3.39 Biểu đồ trình tự use case Quản lý thể loại (1)



Hình 3.40 Biểu đồ trình tự use case Quản lý thể loại (2)

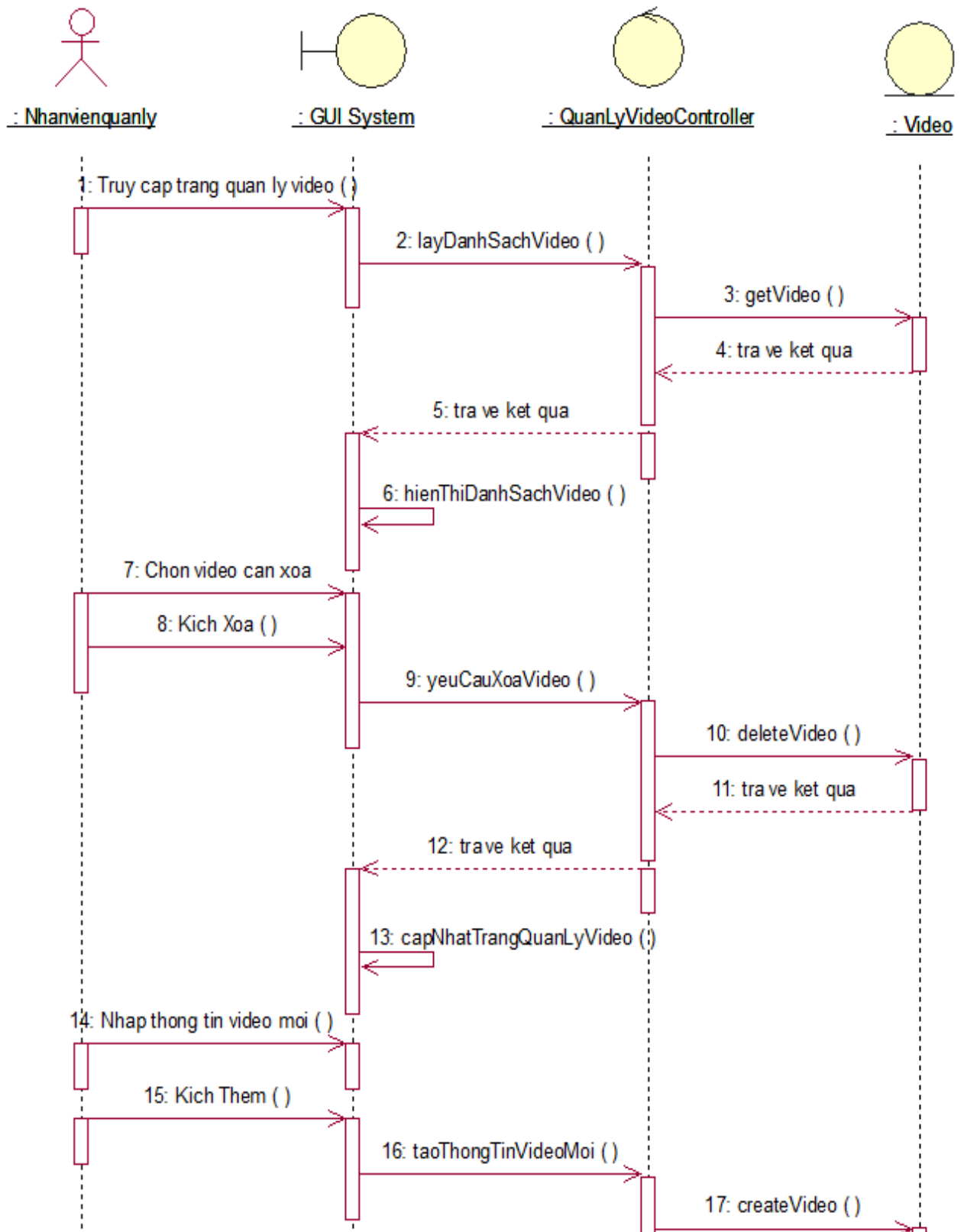
3.4.5.16. Use case Quản lý kho hàng



Hình 3.41 Biểu đồ trình tự use case Quản lý kho hàng (1)

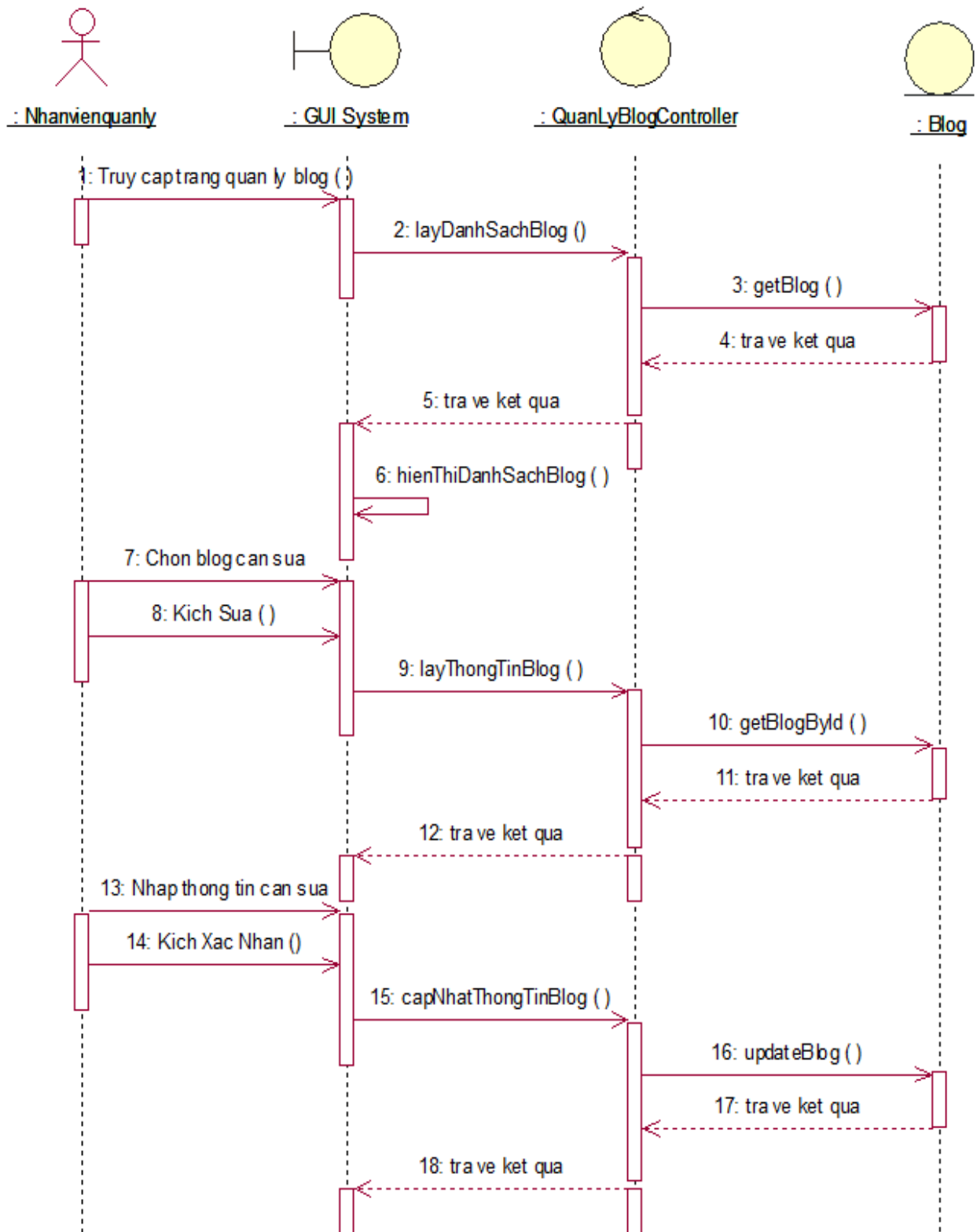
Hình 3.42 Biểu đồ trình tự use case Quản lý kho hàng (2)

3.4.5.17. Use case Quản lý video

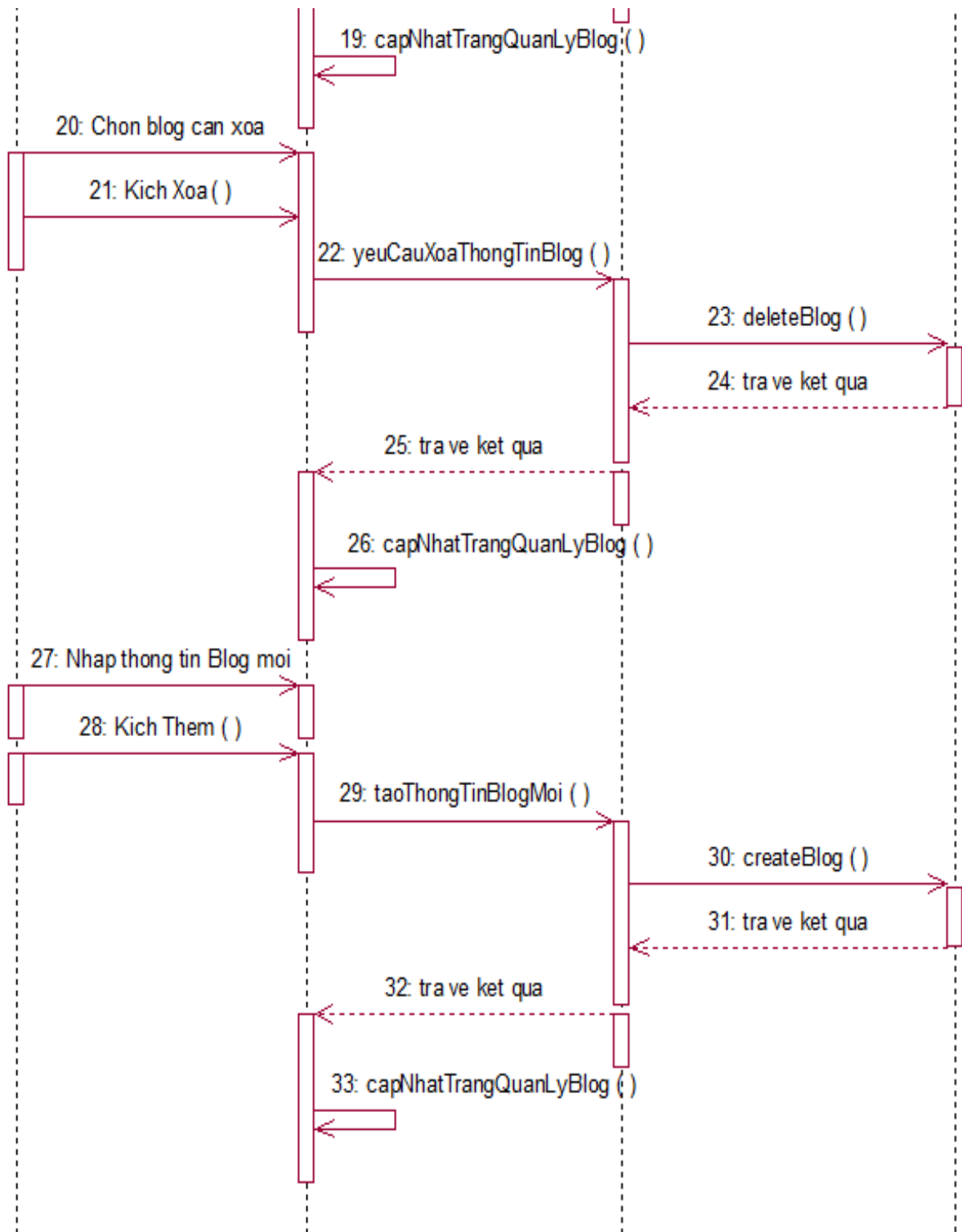


Hình 3.43 Biểu đồ trình tự use case Quản lý video (1)

3.4.5.18. Use case Quản lý blog

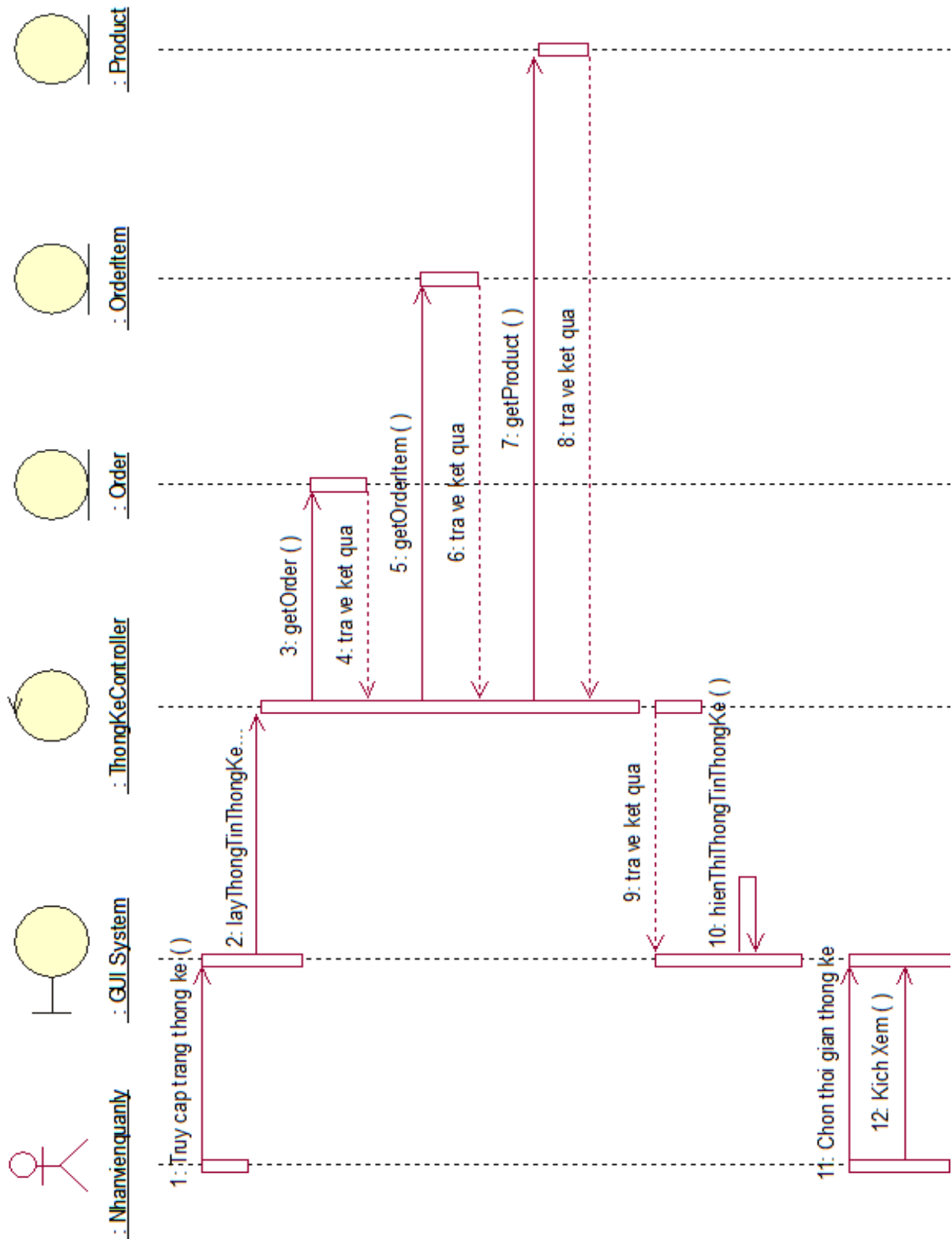


Hình 3.45 Biểu đồ trình tự use case Quản lý blog (1)

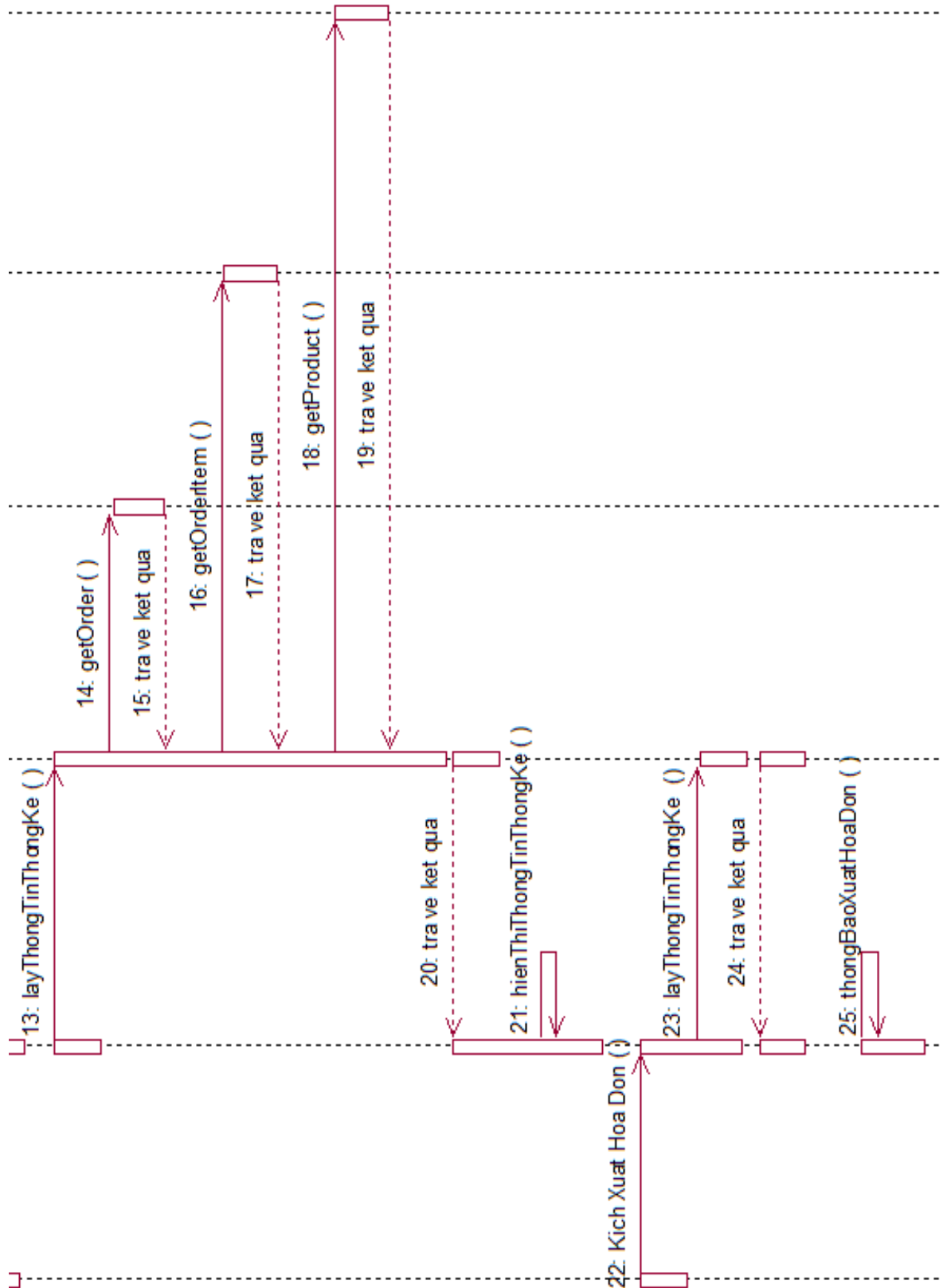


Hình 3.46 Biểu đồ trình tự use case Quản lý blog (2)

3.4.5.19. Use case Thống kê



Hình 3.47 Biểu đồ trình tự use case Thống kê (1)



Hình 3.48 Biểu đồ trình tự use case Thống kê (2)

3.4.6. Cơ sở dữ liệu

Cơ sở dữ liệu của đề tài được lưu trong MongoDB và được tổ chức theo dạng:

Database → chứa nhiều collection

Collection → chứa nhiều document

Document → là một đối tượng JSON

❖ *Collection User (Khách hàng/Nhân viên quản lý/Kiểm kho)*

Bảng 3.1 Collection User

STT	Tên trường	Kiểu dữ liệu	Ý nghĩa	Điều kiện
1	_id	ObjectId	ID nhà cung cấp	PK
2	name	String	Tên NCC	Not null, Unique
3	address	String	Địa chỉ NCC	—
4	phoneNumber	String	Số điện thoại	—
5	email	String	Email của NCC	Unique, Đúng định dạng
6	description	String	Mô tả về NCC	—
7	createdAt	Date	Ngày tạo tài khoản	—
8	updatedAt	Date	Ngày cập nhật thông tin tài khoản gần nhất	—

❖ *Collection Category (Danh mục)*

Bảng 3.2 Collection Category

STT	Tên trường	Kiểu dữ liệu	Ý nghĩa	Điều kiện
1	_id	ObjectId	ID danh mục	PK
2	name	String	Tên danh mục	Not null, Unique
3	description	String	Mô tả danh mục	—
4	image	String	Hình ảnh (URL/path)	—
5	promotion_id	ObjectId	Mã khuyến mãi áp dụng	Ref → Promotion
6	createdAt	Date	Thời gian tạo	—
7	updatedAt	Date	Thời gian cập nhật	—

❖ *Collection Suppliers (Nhà cung cấp)*

Bảng 3.3 Collection Suppliers

STT	Tên trường	Kiểu dữ liệu	Ý nghĩa	Điều kiện
1	_id	ObjectId	ID nhà cung cấp	PK, Increment
2	name	String	Tên NCC	Not null, Unique
3	address	String	Địa chỉ NCC	—
4	phoneNumber	String	Số điện thoại	—
5	email	String	Email của NCC	Unique, Đúng định dạng
6	description	String	Mô tả về NCC	—
7	createdAt	Date	Thời gian tạo	—
8	updatedAt	Date	Thời gian cập nhật	—

❖ *Collection Product (Sản phẩm)*

Bảng 3.4 Collection Product

STT	Tên trường	Kiểu dữ liệu	Ý nghĩa	Điều kiện
1	_id	ObjectId	ID sách	PK
2	name	String	Tên sách	Not null
3	image	String	Ảnh bìa (URL hoặc JSON)	—
4	description	String	Mô tả sách	Not null
5	author	String	Tác giả	Not null
6	rating	Float	Điểm đánh giá trung bình	Not null, Default: 0
7	numReviews	Int	Số lượt đánh giá	Not null, Default: 0
8	category_id	ObjectId	Danh mục chứa sách	Not null, Ref → Category
9	price	Float	Giá gốc	Not null, Default: 0
10	countInStock	Int	Số lượng tồn kho	Not null, Default: 0
11	promotion_id	ObjectId	Mã khuyến mãi	Ref → Promotion
13	createdAt	Date	Thời gian tạo	—
14	updatedAt	Date	Thời gian cập nhật	—

❖ *Collection Reviews (Đánh giá)*

Bảng 3.5 Collection Reviews

STT	Tên trường	Kiểu dữ liệu	Ý nghĩa	Điều kiện
1	_id	ObjectId	ID đánh giá	PK
2	user_id	ObjectId	Người đánh giá	Not null, Ref → User
3	product_id	ObjectId	Sản phẩm được đánh giá	Ref → Product
4	name	String	Tên người đánh giá	Not null
5	rating	Float	Số sao	Not null, Default: 0
6	comment	String	Nội dung đánh giá	Not null

❖ *Collection Promotion (Khuyến mại)*

Bảng 3.6 Collection ImportRecord

STT	Tên trường	Kiểu dữ liệu	Ý nghĩa	Điều kiện
1	_id	ObjectId	ID khuyến mãi	PK
2	name	String	Tên chương trình	Not null
3	description	String	Mô tả	—
4	discountPercentage	Float	% giảm giá	Not null
5	startDate	Date	Thời gian bắt đầu	Not null
6	endDate	Date	Thời gian kết thúc	Not null
7	isActive	Boolean	Đang hoạt động?	Default: true
8	targetType	String	Mục tiêu khuyến mãi	Enum: 'All', 'Category', 'Product'
9	createdAt	Date	Thời gian tạo	—
10	updatedAt	Date	Thời gian cập nhật	—

❖ *Collection Order*

Bảng 3.7 Collection Order

STT	Tên trường	Kiểu dữ liệu	Ý nghĩa	Điều kiện
1	_id	ObjectId	ID đơn hàng	PK

2	user_id	ObjectId	Khách hàng đặt	Not null, Ref → User
3	shippingAddress_*	String	Thông tin giao hàng	—
	shippingAddress_city	String		
	shippingAddress_postalCode	String		
	shippingAddress_country	String		
	shippingAddress_phoneNumber	String		
4	paymentMethod	String	Phương thức thanh toán	Not null, Default: Paypal
5	paymentResult_*	String	Kết quả từ cổng thanh toán	—
	paymentResult_id	String		
	paymentResult_status	String		
	paymentResult_update_time	String		
	paymentResult_email_address	String		
6	taxPrice	Float	Các khoản phí	Not null, Default: 0
7	shippingPrice	Float		
8	totalPrice	Float		
9	isPaid	Boolean	Đã thanh toán chưa?	Not null, Default: false
10	paidAt	Date	Ngày thanh toán	—
11	deliveredAt	Date	Ngày giao	
12	status	String	Trạng thái đơn	Default: 'pending'
13	cancellationReason	String	Lý do hủy	—
14	cancellationReason	String	Lý do trả hàng	—
15	note	String	Ghi chú của quản trị viên	—
16	createdAt	Date	Thời gian tạo	—

17	updatedAt	Date	Thời gian cập nhật	—
----	-----------	------	--------------------	---

❖ **Collection OrderItem (Sản phẩm trong đơn)**

Bảng 3.8 Collection OrderItem

STT	Tên trường	Kiểu dữ liệu	Ý nghĩa	Điều kiện
1	_id	ObjectId	ID mục hàng	PK
2	order_id	ObjectId	Đơn hàng chứa mục	Ref → Order
3	product_id	ObjectId	Sản phẩm được đặt	Not null
4	name	String	Tên sản phẩm	Not null
5	qty	Int	Số lượng	Not null
6	image	String	Ảnh sản phẩm	—
7	price	Float	Giá gốc	Not null
8	discountedPrice	Float	Giá sau khuyến mãi	Not null

❖ **Collection OrderStatusHistory (Lịch sử đơn hàng)**

Bảng 3.9 Collection OrderStatusHistory

STT	Tên trường	Kiểu dữ liệu	Ý nghĩa	Điều kiện
1	_id	ObjectId	ID lịch sử trạng thái	PK
2	order_id	ObjectId	Thuộc đơn hàng nào	Ref → Order
3	updatedAtBy_user_id	ObjectId	Ai cập nhật	Ref → User
4	status	String	Trạng thái cập nhật	Not null
5	updatedAt	Date	Thời gian cập nhật	Default: now()
6	notes	String	Ghi chú nội bộ	—

❖ **Collection Video**

Bảng 3.10 Collection Video

STT	Tên trường	Kiểu dữ liệu	Ý nghĩa	Điều kiện
1	_id	ObjectId	ID video	PK
2	title	String	Tiêu đề video	Not null
3	youtubeId	String	ID YouTube	Not null

4	createdBy_user_id	Date	Ai tạo?	Ref → User
5	updatedBy_user_id	Date	Ai cập nhật?	Ref → User
6	createdAt	Date	Ngày tạo	—
7	updatedAt	Date	Ngày cập nhật gần nhất	—

❖ *Collection Blog*

Bảng 3.11 Collection Blog

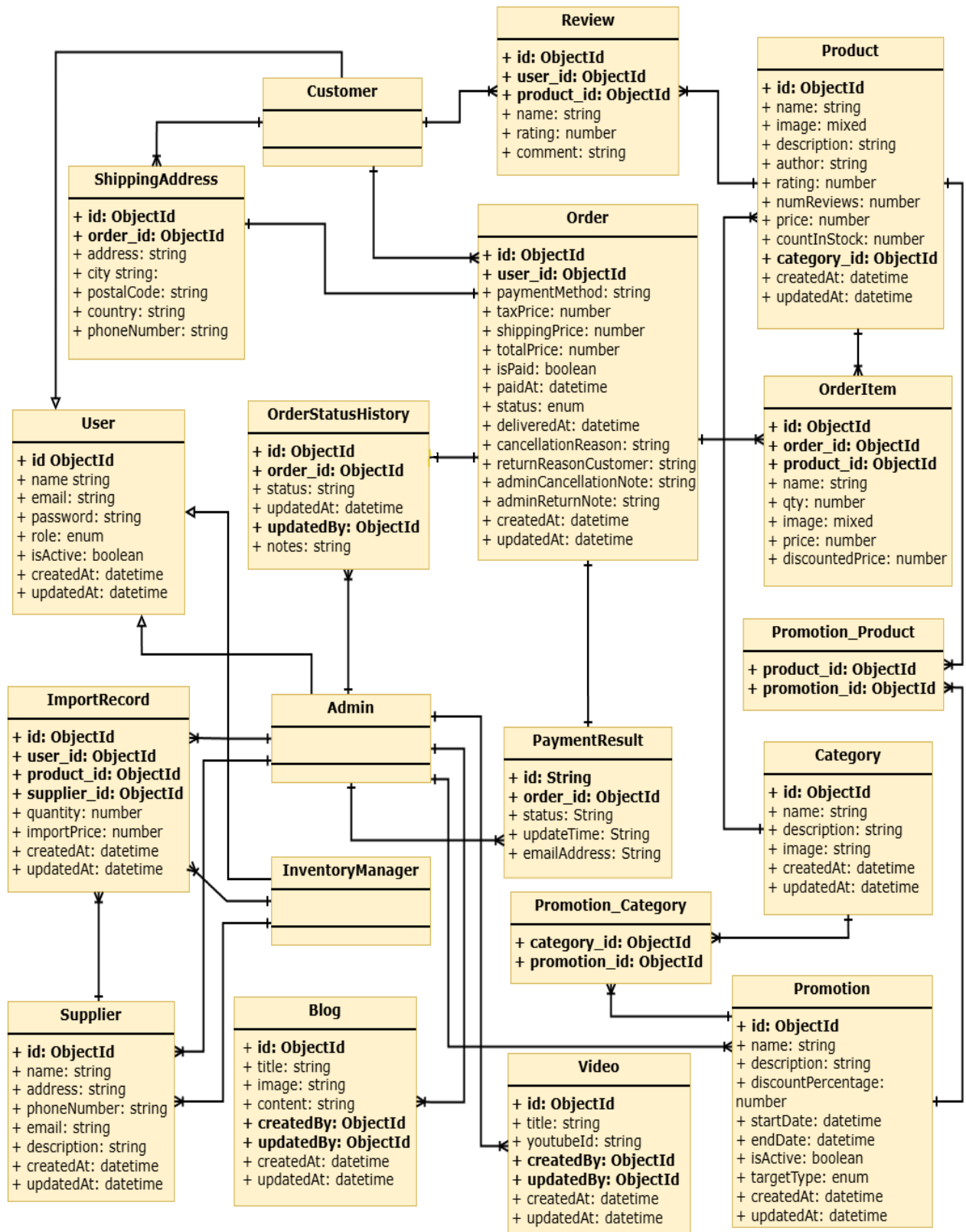
STT	Tên trường	Kiểu dữ liệu	Ý nghĩa	Điều kiện
1	_id	ObjectId	ID bài viết	PK
2	title	String	Tiêu đề bài viết	Not null
3	image	String	Ảnh đại diện	Not null
4	content	String	Nội dung	Not null
5	createdBy_user_id	Date	Ai tạo?	Ref → User
6	updatedBy_user_id	Date	Ai cập nhật?	Ref → User
7	createdAt	Date	Ngày tạo	—
8	updatedAt	Date	Ngày cập nhật gần nhất	—

❖ *Collection ImportRecord (Kho hàng)*

Bảng 3.12 Collection ImportRecord

STT	Tên trường	Kiểu dữ liệu	Ý nghĩa	Điều kiện
1	_id	ObjectId	ID nhập hàng	PK
2	user_id	ObjectId	Người thực hiện nhập	Not null
3	product_id	ObjectId	Sản phẩm được nhập	Not null
4	quantity	Int	Số lượng nhập	Not null, Default: 0
5	importPrice	Float	Giá nhập mỗi đơn vị	Not null, Default: 0
6	supplier	ObjectId	Nhà cung cấp	Ref → Suppliers
7	createdAt	Date	Ngày tạo	—
8	updatedAt	Date	Ngày cập nhật gần nhất	—

3.4.7. Mô hình thực thể - quan hệ



Hình 3.49 Mô hình thực thể liên kết hệ thống

CHƯƠNG 4. CÀI ĐẶT VỀ KIỂM THỬ HỆ THỐNG

4.1. Cài đặt môi trường

4.1.2. Môi trường hoạt động

- Hệ điều hành: Windows 11 Pro 64-bit (10.0, Build 26100)
- Ngôn ngữ lập trình: JavaScript
- Cơ sở dữ liệu: Cloud MongoDB, Mongoose 6.1.4
- Môi trường chạy JavaScript: Node.js 16.16.0, Express 4.17.2
- Giao diện: React 17.0.2
- Công cụ: Visual Studio Code

4.1.3. Cài đặt môi trường phát triển

Cấu trúc dự án:

BookShop

/Server

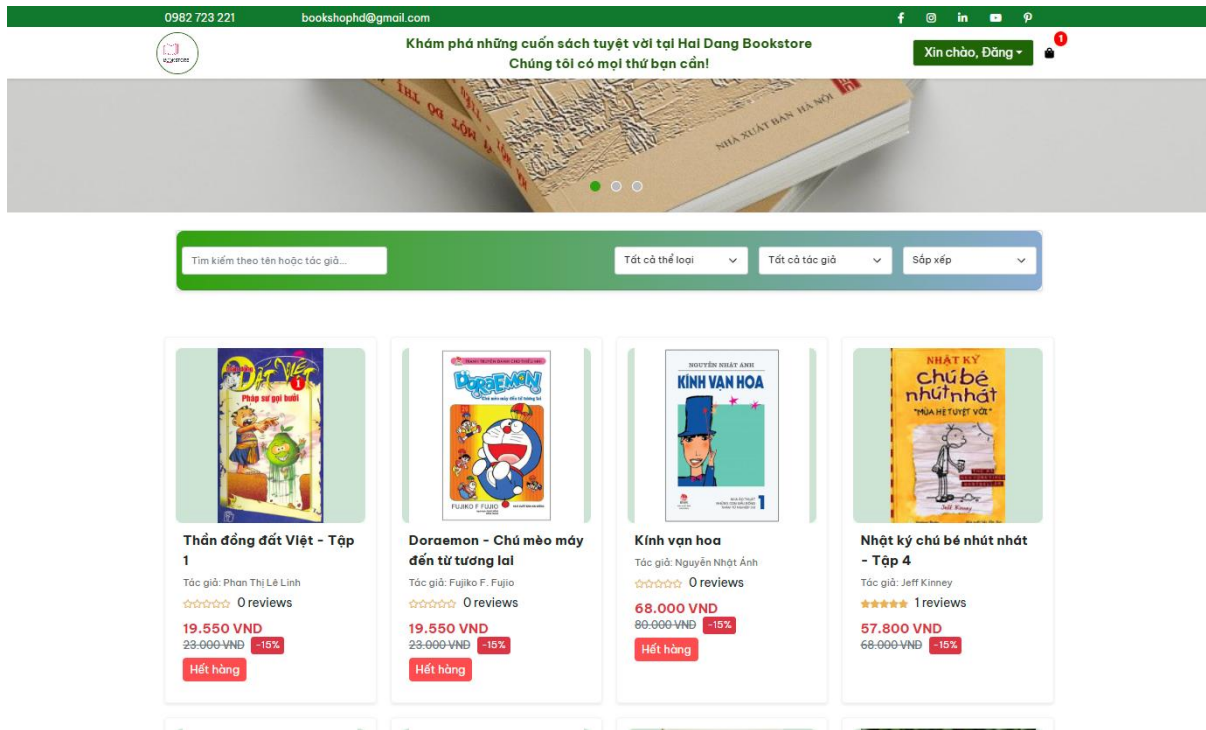
/Customer

/Admin

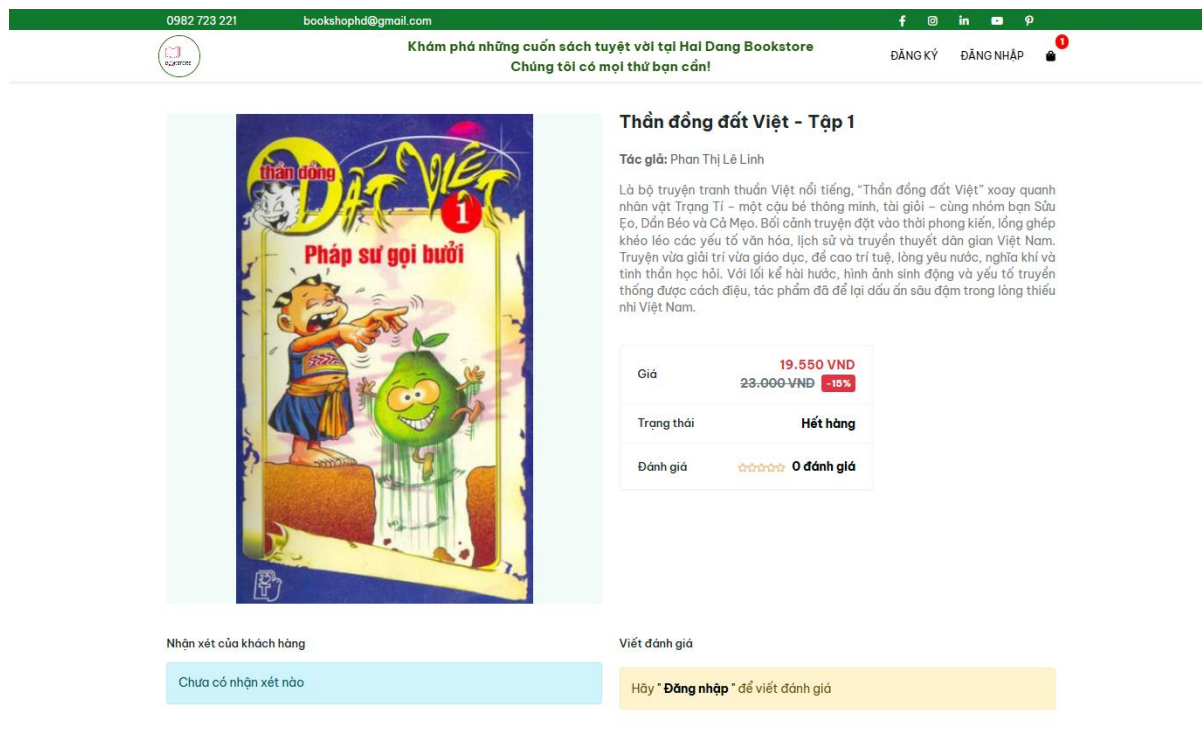
Bước	Server	Customer	Admin
1. Điều hướng	cd Server	cd Customer	cd Admin
2. Cài đặt thư viện (Cho lần đầu cài đặt, từ lần sau bỏ qua bước này)	npm i	npm i	npm i
3. Khởi chạy	npm run server	npm start	npm start

4.2. Một số giao diện hệ thống

4.2.1. Giao diện khách hàng



Hình 4.1 Giao diện trang chủ Khách hàng



Hình 4.2 Giao diện trang chi tiết sản phẩm Khách hàng

0982 723 221
bookshophd@gmail.com
f @ in v p
Khám phá những cuốn sách tuyệt vời tại Hai Dang Bookstore
Chúng tôi có mọi thứ bạn cần!
Xin chào, Đăng

Đăng
Joined March 23, 2025
CÀI ĐẶT TRANG CÁ NHÂN
ĐƠN HÀNG CỦA BẠN

Mã ĐH	Ngày Đặt	Tổng Tiền	Thanh Toán	Trạng Thái Đơn Hàng	
682b8cb31309b5561094a2f1	20/05/2025 02:55	123.000 VND		Đã thanh toán (Chờ xử lý)	Xem Chi Tiết
682b623bcc9983e2e3256db3	19/05/2025 23:54	75.000 VND		Đã thanh toán (Chờ xử lý)	Xem Chi Tiết
682b61e1cc9983e2e3256c46	19/05/2025 23:52	120.000 VND		Đang giao hàng	Xem Chi Tiết
682b55f4cc9983e2e32569f3	19/05/2025 23:01	135.000 VND		Đã giao hàng	Xem Chi Tiết
682862867f312c39530d4db8	17/05/2025 17:18	85.000 VND		Yêu cầu hoàn bị từ chối	Xem Chi Tiết
682860575cb15e260d820e67	17/05/2025 17:09	82.800 VND		Đã hủy	Xem Chi Tiết
68285fa55cb15e260d820c09	17/05/2025 17:06	82.800 VND		Yêu cầu hoàn bị từ chối	Xem Chi Tiết
6828292ad10f4fc6564f1651	17/05/2025 13:14	327.643 VND		Yêu cầu hoàn bị từ chối	Xem Chi Tiết
682610b56e9419b210e2d506	15/05/2025 23:05	218.800 VND		Đã giao hàng	Xem Chi Tiết
6825db8e88eca4d179cb7c85	15/05/2025 19:18	89.600 VND		Đã giao hàng	Xem Chi Tiết

Hình 4.3 Giao diện danh sách đơn hàng Khách hàng

0982 723 221
bookshophd@gmail.com
f @ in v p
Khám phá những cuốn sách tuyệt vời tại Hai Dang Bookstore
Chúng tôi có mọi thứ bạn cần!
ĐĂNG KÝ ĐĂNG NHẬP

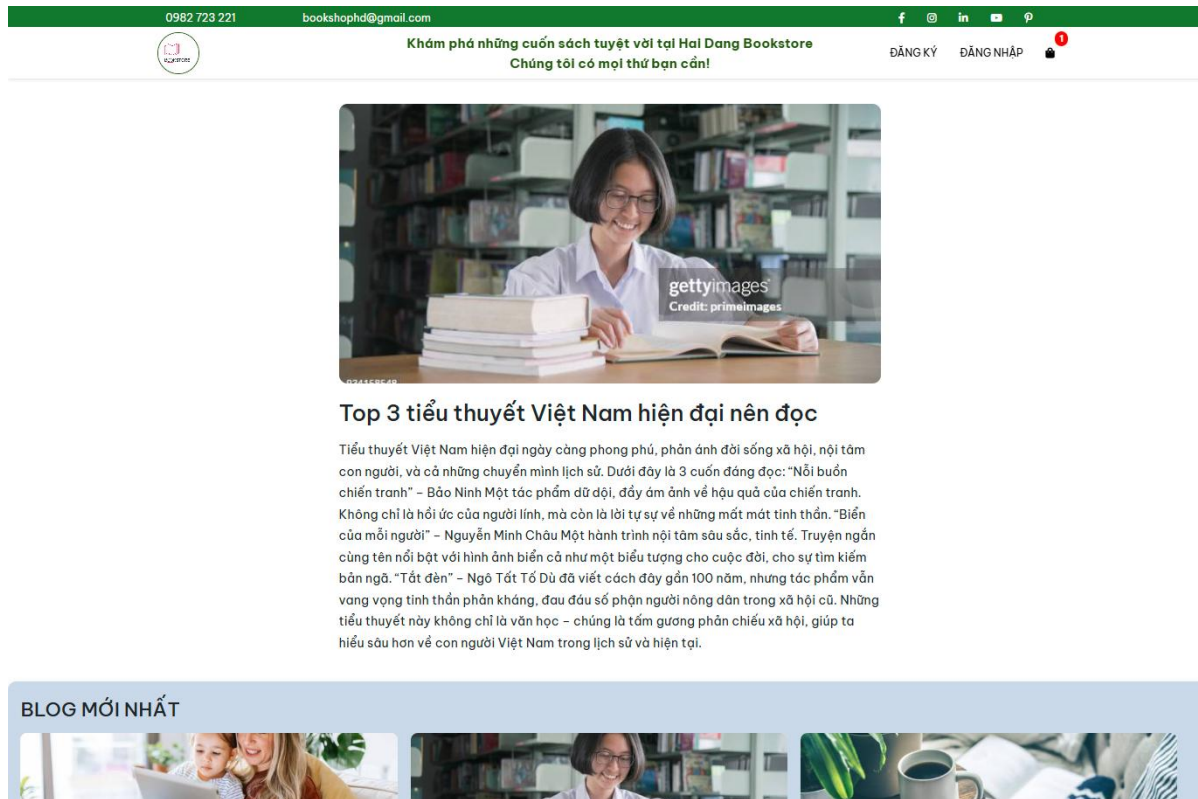
Email

Mật khẩu

ĐĂNG NHẬP

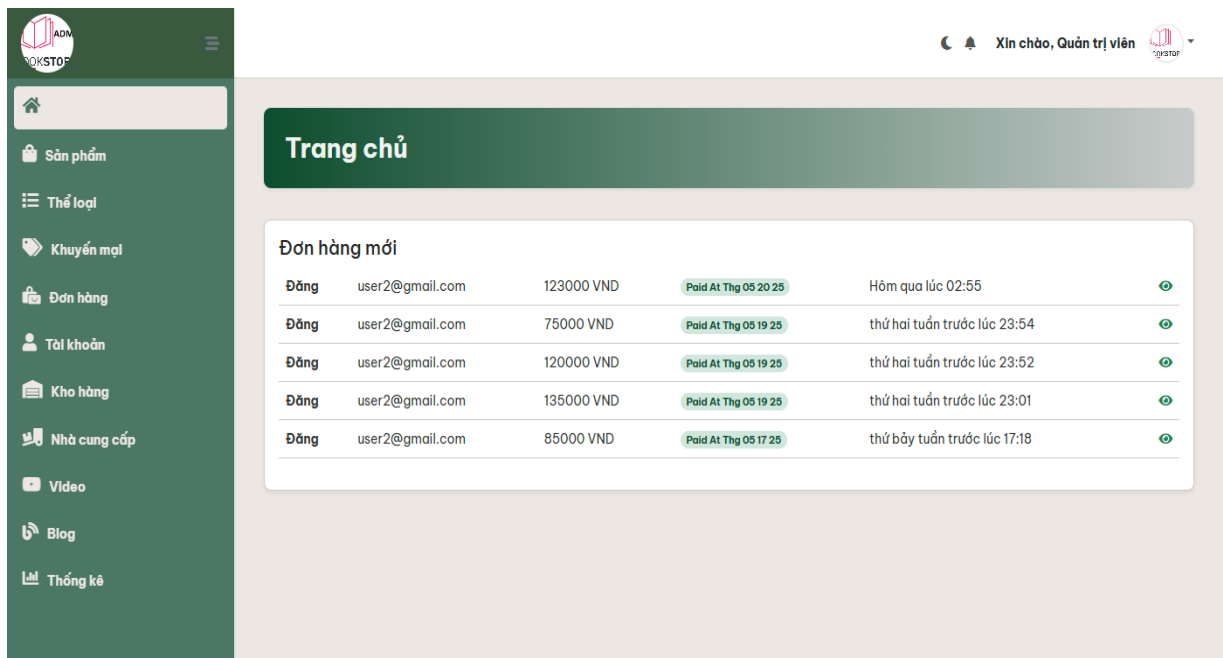
Bạn chưa có tài khoản? Tạo tài khoản

Hình 4.4 Giao diện đăng nhập Khách hàng

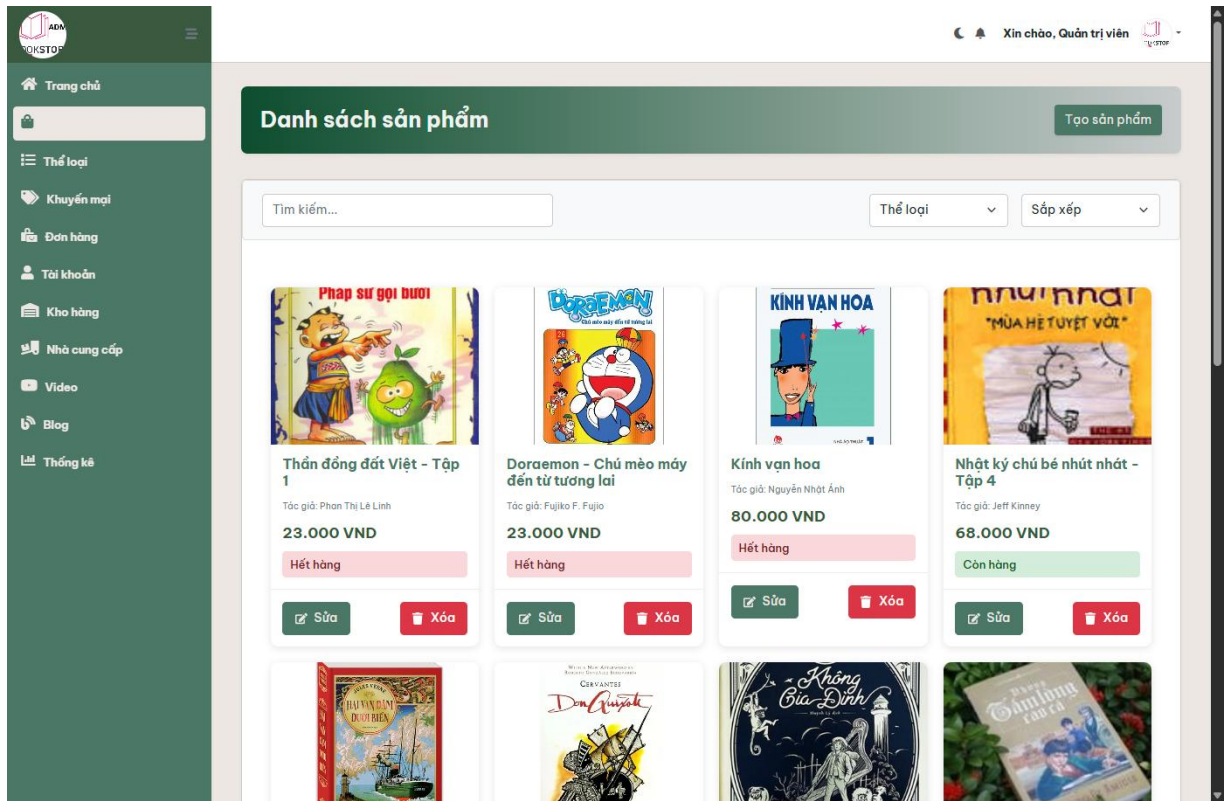


Hình 4.5 Giao diện chi tiết bài Blog

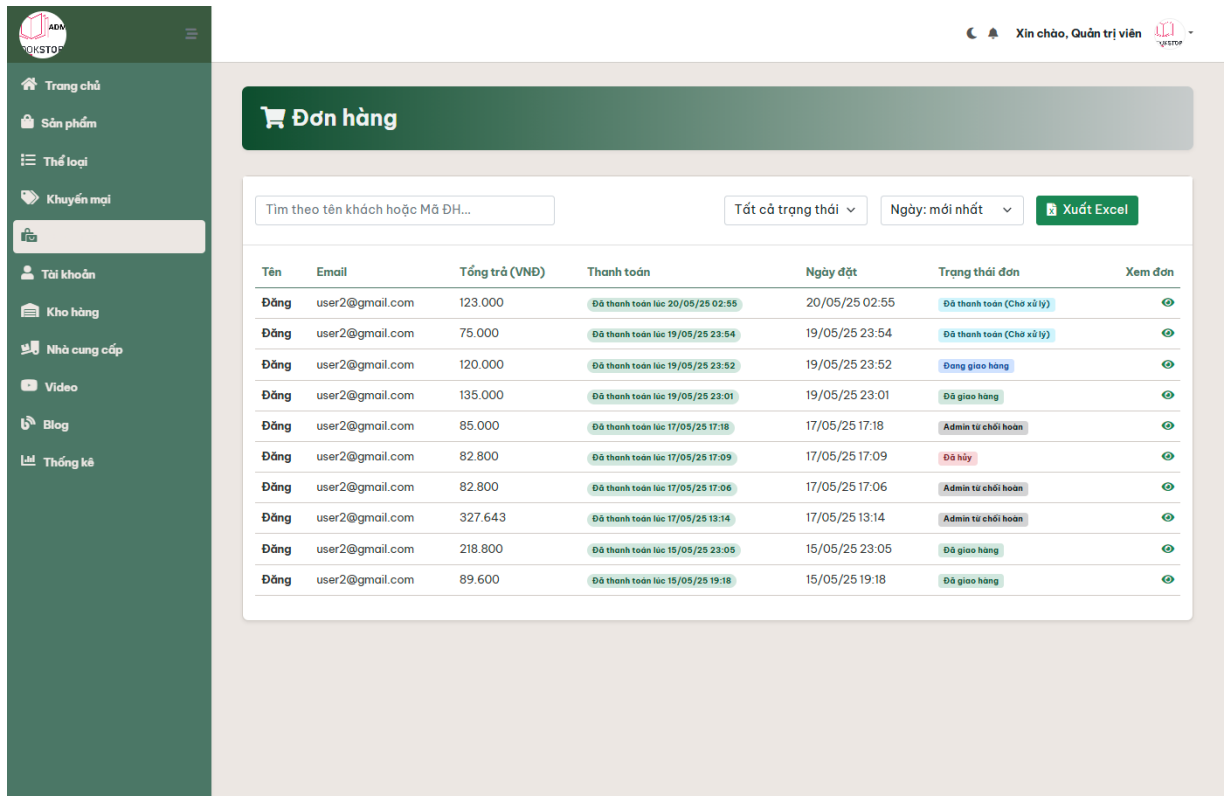
4.2.2. Giao diện nhân viên quản lý



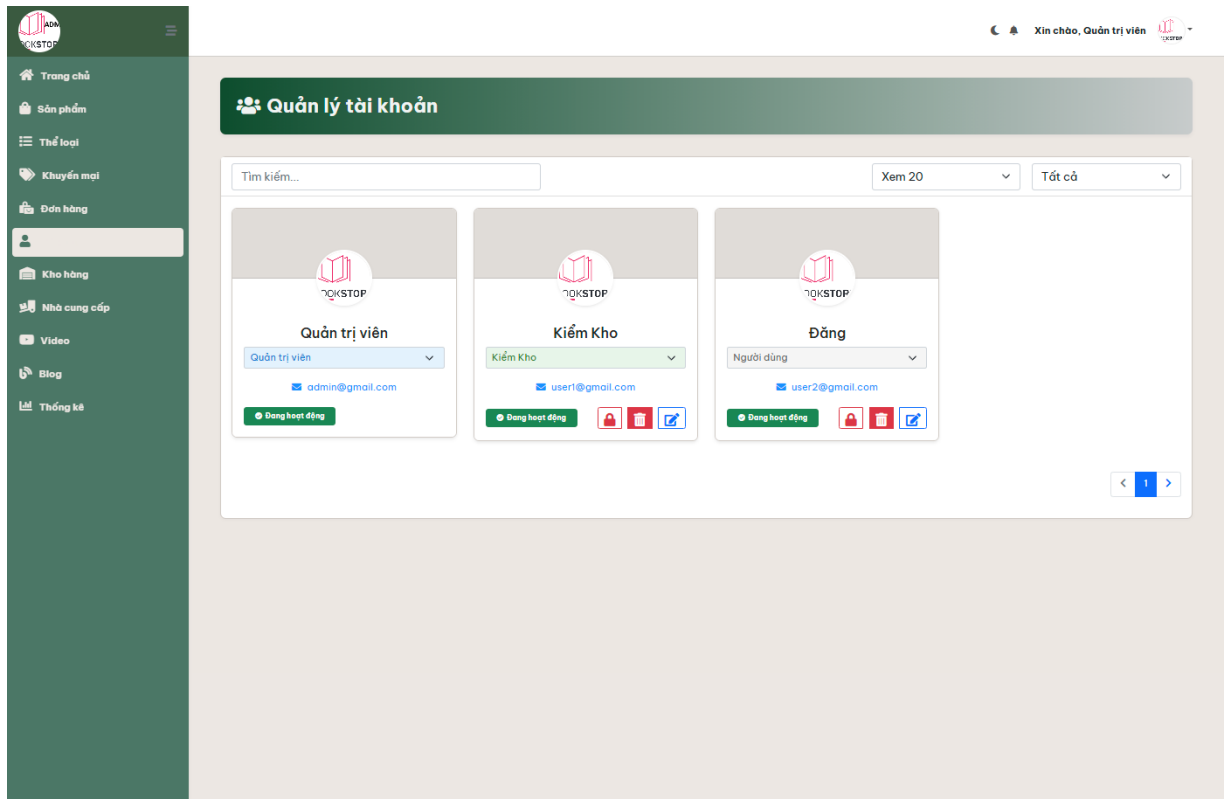
Hình 4.6 Giao diện trang chủ quản lý



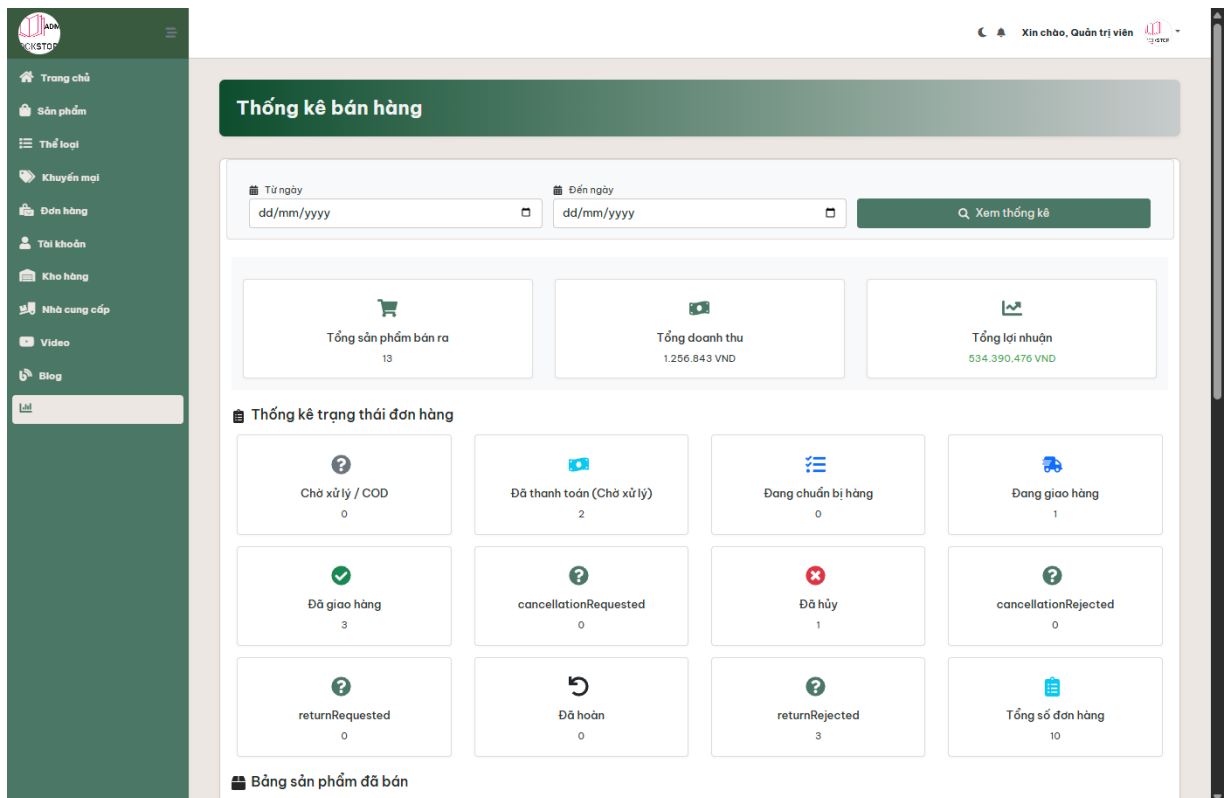
Hình 4.7 Giao diện trang quản lý sản phẩm



Hình 4.8 Giao diện trang quản lý đơn hàng

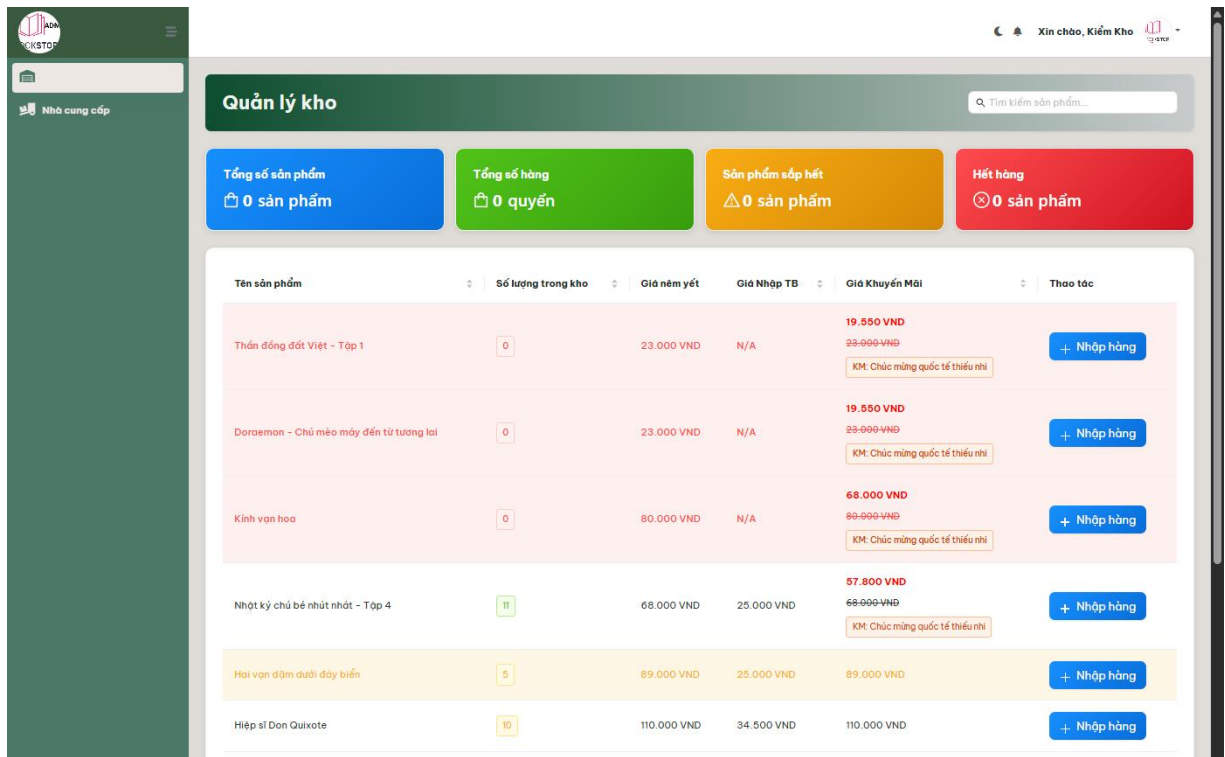


Hình 4.9 Giao diện trang quản lý tài khoản

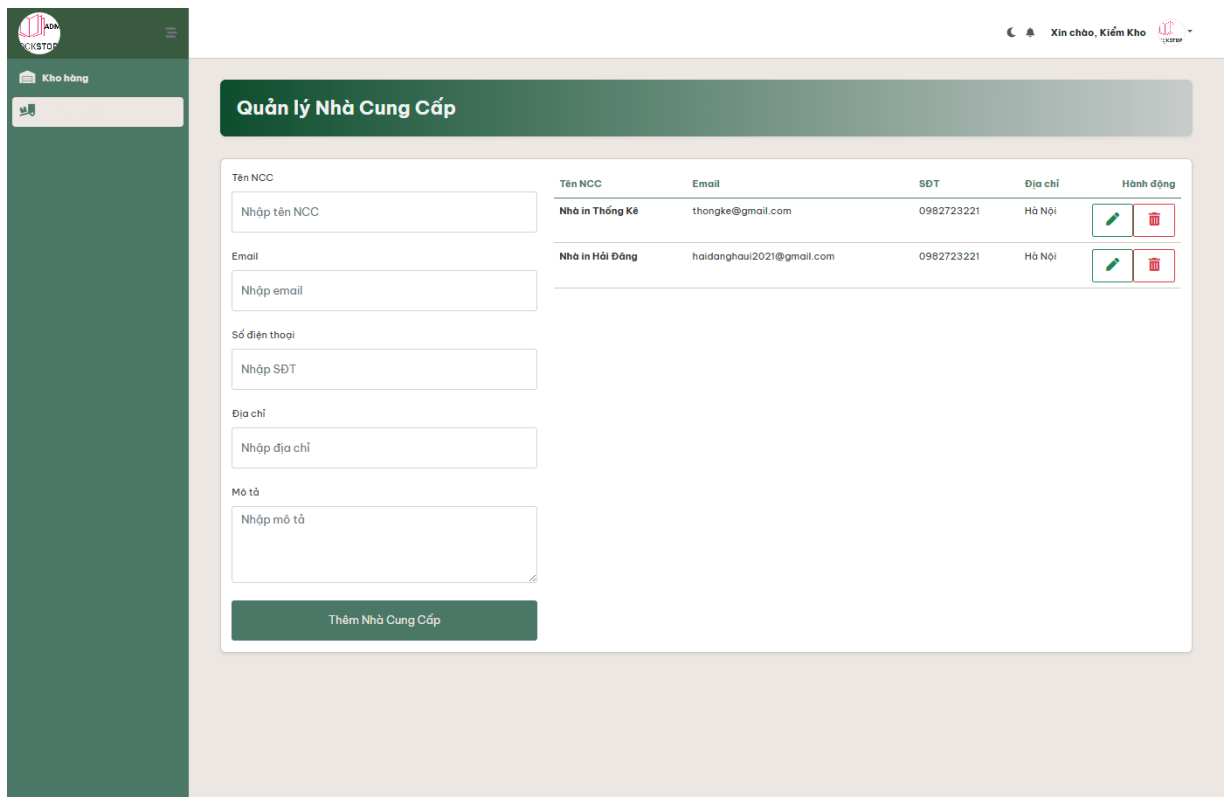


Hình 4.10 Giao diện trang quản lý thống kê

4.2.3. Giao diện nhân viên kiểm kho



Hình 4.11 Giao diện trang quản lý kho hàng



Hình 4.12 Giao diện trang quản lý nhà cung cấp

Chỉnh Sửa Nhà Cung Cấp

Quay lại Danh Sách NCC Cập Nhật

Tên NCC
Nhà in Thống Kê

Email
thongke@gmail.com

Số điện thoại
0982723221

Địa chỉ
Hà Nội

Mô tả
NCC1

Lịch sử nhập hàng từ nhà cung cấp này

Tên sản phẩm	Số lượng	Giá nhập / đơn vị	Tổng tiền	Ngày nhập	Người nhập
Tôi đi học	44	32.000 VND	1.408.000 VND	18/5/2025	Quản trị viên
Bố của Xi-mông	4	13.000 VND	52.000 VND	18/5/2025	Quản trị viên
Những tấm lòng cao cả	13	24.000 VND	312.000 VND	18/5/2025	Quản trị viên
Hai ván đầm dưới đầu biển	5	25.000 VND	125.000 VND	18/5/2025	Quản trị viên

Hình 4.13 Giao diện trang chi tiết nhà cung cấp

4.3. Kiểm thử hệ thống

4.3.1. Khái niệm và đặc điểm của kiểm thử hệ thống

Kiểm thử phần mềm (Software Testing) là quá trình nhằm phát hiện và tìm ra các lỗi trong phần mềm, đồng thời đảm bảo rằng phần mềm hoạt động chính xác, đầy đủ và đáp ứng đúng các yêu cầu đã được đặt ra bởi khách hàng hoặc theo đặc tả kỹ thuật ban đầu. Quá trình này giúp cung cấp một cái nhìn độc lập và khách quan về phần mềm, từ đó đánh giá mức độ rủi ro khi vận hành phần mềm.

Không chỉ dừng lại ở việc tìm lỗi, kiểm thử phần mềm còn nhằm mục đích xác định xem phần mềm có đáp ứng được kỳ vọng của các bên liên quan (cổ đông, khách hàng,...) hay không, cũng như đánh giá tổng thể chất lượng sản phẩm thông qua các yếu tố như hiệu suất, tính năng, tính đầy đủ, độ tiện dụng và độ ổn định. Nói cách khác, kiểm thử phần mềm đóng vai trò quan trọng trong việc kiểm soát chất lượng sản phẩm.

Trong quy trình kiểm thử, người kiểm thử tiến hành ở nhiều cấp độ khác nhau. Một trong số đó là kiểm thử hệ thống (System Testing) – đây là giai đoạn đánh giá toàn bộ hệ thống phần mềm sau khi đã được tích hợp hoàn chỉnh, dựa theo các yêu cầu chức năng và đặc tả kỹ thuật đã xác định từ trước. Mục tiêu chính là trả lời cho câu hỏi: "Liệu toàn bộ hệ thống có hoạt động đúng như mong đợi hay không?"

Kiểm thử hệ thống được tiến hành theo phương pháp hộp đen (Black-box Testing) – nghĩa là chỉ tập trung vào các đầu ra và hành vi bên ngoài của phần mềm, không xét đến cấu trúc bên trong như mã nguồn hay thiết kế. Do đó, người thực hiện không cần có kiến thức sâu về lập trình mà chỉ cần đánh giá từ góc nhìn của người sử dụng cuối.

Đặc điểm của Kiểm thử hệ thống:

- + Kiểm thử hộp đen (Black-box testing): Chỉ tập trung vào đầu vào và đầu ra của hệ thống. Không cần biết đến cấu trúc bên trong, mã nguồn hay thuật toán của phần mềm.

- + Được thực hiện sau kiểm thử tích hợp (Integration Testing): Khi tất cả các mô-đun đã được tích hợp thành một hệ thống hoàn chỉnh. Là bước trung gian trước khi chuyển sang kiểm thử chấp nhận (Acceptance Testing).

- + Đánh giá toàn bộ hệ thống: Kiểm tra các chức năng chính và phụ của phần mềm như mô tả trong tài liệu đặc tả yêu cầu. Bao gồm cả các khía cạnh phi chức năng như hiệu năng, bảo mật, khả năng sử dụng, tương thích...

- + Đóng vai trò như người dùng cuối: Kiểm thử được thực hiện từ góc nhìn của người sử dụng, đảm bảo hệ thống thân thiện và đúng kỳ vọng.

- + Đảm bảo tính đúng đắn và đầy đủ: Xác minh rằng hệ thống đáp ứng đầy đủ các yêu cầu đã đặt ra ban đầu. Là bước đánh giá chất lượng tổng thể của sản phẩm phần mềm.

- + Môi trường kiểm thử gần giống môi trường thật: Thường được tiến hành trong môi trường tương đương hoặc giống với môi trường triển khai thực tế.

4.3.2. Các lĩnh vực của Kiểm thử hệ thống

Một số khía cạnh chính mà kiểm thử hệ thống tập trung vào bao gồm:

- + Hiệu suất: Đảm bảo hệ thống phần mềm hoạt động đúng theo các yêu cầu của người dùng, không phát sinh lỗi hoặc sự cố trong quá trình sử dụng.
- + Bảo mật: Bảo vệ hệ thống khỏi các mối đe dọa bảo mật, ngăn chặn việc truy cập trái phép, đánh cắp hoặc mất dữ liệu quan trọng của người dùng và tổ chức.
- + Giao diện: Đánh giá tính đúng đắn và đầy đủ của giao diện người dùng, bảo đảm rằng mọi yêu cầu giao diện được đáp ứng và các thành phần tích hợp với nhau một cách ổn định, không gây lỗi.
- + Khả năng cài đặt: Xác minh rằng hệ thống có thể được cài đặt và triển khai vào môi trường thực tế (production) một cách suôn sẻ, không gặp khó khăn hoặc sự cố.
- + Tải và chịu tải (Load/Stress): Kiểm tra khả năng hoạt động ổn định của hệ thống dưới các điều kiện tải khác nhau, từ bình thường đến cao, nhằm đảm bảo hệ thống không bị quá tải hoặc sụp đổ.

4.3.3. Quá trình kiểm thử

Do tầm quan trọng lớn của kiểm thử hệ thống trong Vòng đời phát triển phần mềm (SDLC), việc xác định và tuân thủ một quy trình kiểm thử cụ thể là điều hết sức cần thiết. Điều này giúp đảm bảo rằng mọi bước trong quy trình đều được thực hiện đầy đủ, chính xác, không bỏ sót bất kỳ chi tiết quan trọng nào. Mặc dù quy trình kiểm thử có thể thay đổi tùy theo từng dự án, nhưng nhìn chung, quá trình kiểm thử hệ thống thường bao gồm sáu bước cơ bản sau:

Xây dựng kế hoạch kiểm thử (Test Plan): Đây là bước đầu tiên, nơi trưởng nhóm kiểm thử hoặc quản lý kiểm thử xác định phạm vi và mục tiêu kiểm thử, lựa chọn chiến lược kiểm thử phù hợp, quyết định giữa kiểm thử thủ công hay tự động, thiết lập tiêu chí đầu vào/đầu ra, và phân công vai trò, trách nhiệm cụ thể cho từng thành viên.

Thiết kế các ca kiểm thử (Test Case): Các test case được xây dựng dựa trên các use case và yêu cầu hệ thống, bao gồm các khía cạnh như kỹ thuật, giao diện người dùng, chức năng, phi chức năng, hiệu suất,...

Chuẩn bị dữ liệu kiểm thử (Test Data): Sau khi tạo test case, nhóm kiểm thử tiến hành lựa chọn hoặc tạo dữ liệu kiểm thử tương ứng. Đây là các điều kiện đầu vào giúp xác minh tính đúng đắn và độ tin cậy của hệ thống thông qua các kết quả đầu ra mong đợi.

Thực thi kiểm thử (Test Execution): Các test case được triển khai trên hệ thống, quá trình thực hiện được giám sát chặt chẽ, đồng thời ghi lại bất kỳ sự khác biệt, lỗi hoặc sự cố nào phát sinh. Kết quả đầu ra cũng được lưu trữ để phục vụ đánh giá sau này.

Báo cáo và xử lý lỗi (Defect Reporting & Fixing): Mọi lỗi phát hiện được sẽ được ghi nhận và báo cáo chi tiết cho các thành viên liên quan. Sau đó, các lập trình viên sẽ phối hợp cùng nhóm kiểm thử để phân tích nguyên nhân và khắc phục lỗi.

Lặp lại kiểm thử (Retesting – nếu cần): Sau khi các lỗi đã được sửa chữa, quá trình kiểm thử sẽ được lặp lại để đảm bảo rằng các vấn đề đã được giải quyết triệt để và hệ thống hoạt động đúng như mong đợi.

4.3.4. Lý do thực hiện kiểm thử hệ thống

Kiểm thử hệ thống là một bước không thể thiếu trong quy trình phát triển phần mềm nhằm đảm bảo rằng toàn bộ hệ thống hoạt động đúng với các yêu cầu đã đề ra. Việc thực hiện kiểm thử hệ thống mang lại những lợi ích quan trọng sau:

+ **Đảm bảo chất lượng sản phẩm:** Kiểm thử giúp phát hiện và loại bỏ các lỗi hoặc sự cố trước khi hệ thống được triển khai chính thức, từ đó nâng cao độ tin cậy và chất lượng phần mềm.

+ Đáp ứng yêu cầu người dùng: Đảm bảo rằng phần mềm hoạt động đúng theo các chức năng đã được xác định trong tài liệu yêu cầu, mang lại trải nghiệm đúng như mong đợi cho người dùng cuối.

+ Phát hiện lỗi sớm: Giúp phát hiện các vấn đề về logic, hiệu suất, bảo mật hoặc tích hợp giữa các thành phần hệ thống một cách sớm nhất, giảm chi phí sửa lỗi về sau.

+ Đảm bảo khả năng tích hợp: Đảm bảo các module, chức năng trong hệ thống có thể hoạt động cùng nhau một cách liền mạch, không gây ra xung đột hoặc lỗi hệ thống.

+ Tăng tính bảo mật: Kiểm thử giúp phát hiện các lỗ hổng có thể bị khai thác, từ đó tăng cường khả năng bảo vệ dữ liệu và an toàn hệ thống.

+ Đảm bảo hệ thống sẵn sàng triển khai: Thông qua kiểm thử, đội ngũ phát triển xác minh rằng hệ thống có thể hoạt động ổn định trong môi trường thực tế và sẵn sàng đưa vào vận hành.

4.4. Thiết kế các test case cho hệ thống

Use case Đăng nhập

Bảng 4.1 Test case use case Đăng nhập

STT	Mô tả	Dữ liệu đầu vào	Bước thực hiện	Kết quả mong đợi	Trạng thái
TC01					
1	Đăng nhập thành công	Email hợp lệ, mật khẩu đúng	Nhập email và mật khẩu > Submit	Chuyển đến trang chủ người dùng	Đạt
2	Sai mật khẩu	Email hợp lệ, mật khẩu sai	Nhập email và mật khẩu > Submit	Hiển thị thông báo “Mật khẩu không đúng”	Đạt
3	Email không tồn tại	Email không tồn tại	Nhập email và mật khẩu > Submit	Hiển thị “Tài khoản không tồn tại”	Đạt

4	Để trống trường dữ liệu	Trống email hoặc mật khẩu	Submit không điền dữ liệu	Hiển thị cảnh báo “Vui lòng điền đầy đủ thông tin”	Đạt
---	-------------------------	---------------------------	---------------------------	--	-----

Use case Đăng ký

Bảng 4.2 Test case use case Đăng ký

STT	Mô tả	Dữ liệu đầu vào	Bước thực hiện	Kết quả mong đợi	Trạng thái
TC02					
1	Đăng ký thành công	Tên, email, mật khẩu hợp lệ	Điền form đăng ký > Submit	Chuyển đến trang đăng nhập và hiển thị thông báo đăng ký thành công	Đạt
2	Email đã tồn tại	Email đã có trong hệ thống	Điền form > Submit	Hiển thị thông báo “Email đã được đăng ký”	Đạt
3	Mật khẩu không khớp xác nhận	Mật khẩu và xác nhận không trùng	Điền form > Submit	Hiển thị lỗi “Mật khẩu không khớp”	Đạt
4	Trường thông tin thiếu	Bỏ trống một hoặc nhiều trường	Submit	Hiển thị lỗi bắt buộc điền	Đạt

Use case Đăng xuất

Bảng 4.3 Test case use case Đăng xuất

STT	Mô tả	Dữ liệu đầu vào	Bước thực hiện	Kết quả mong đợi	Trạng thái
TC03					
1	Đăng xuất thành công	Tài khoản đã đăng nhập	Nhấn nút “Đăng xuất”	Quay về trang đăng nhập và kết thúc phiên làm việc	Đạt

2	Đăng xuất khi chưa đăng nhập	Người dùng chưa đăng nhập	Truy cập đường dẫn logout	Hiển thị thông báo lỗi hoặc chuyển đến trang đăng nhập	Đạt
---	------------------------------	---------------------------	---------------------------	--	-----

Use case: Quản lý tài khoản

Bảng 4.4 Test case use case Quản lý tài khoản

STT	Mô tả	Dữ liệu đầu vào	Bước thực hiện	Kết quả mong đợi	Trạng thái
TC04					
1	Xem danh sách tài khoản	Quyền admin	Truy cập mục “Quản lý tài khoản”	Hiển thị danh sách người dùng	Đạt
2	Chỉnh sửa thông tin người dùng	ID người dùng, dữ liệu mới	Click “Sửa” > Nhập dữ liệu > Lưu	Cập nhật thông tin thành công	Đạt
3	Xóa tài khoản người dùng	ID người dùng	Click “Xóa”	Hiển thị cảnh báo xác nhận > Tài khoản bị xóa	Đạt
4	Không có quyền truy cập	Người dùng thường	Truy cập link quản lý tài khoản	Hiển thị lỗi “Không có quyền truy cập”	Đạt

Use case Quản lý sản phẩm

Bảng 4.5 Test case use case Quản lý sản phẩm

STT	Mô tả	Dữ liệu đầu vào	Bước thực hiện	Kết quả mong đợi	Trạng thái
TC05					
1	Thêm sản phẩm mới	Thông tin sản phẩm đầy đủ	Click “Thêm mới” > Nhập dữ liệu > Lưu	Sản phẩm mới hiển thị trong danh sách	Đạt

2	Sửa thông tin sản phẩm	ID sản phẩm, dữ liệu chỉnh sửa	Click “Sửa” > Cập nhật dữ liệu > Lưu	Thông tin được cập nhật	Đạt
3	Xóa sản phẩm	ID sản phẩm	Click “Xóa” > Xác nhận	Sản phẩm bị loại khỏi danh sách	Đạt
4	Tìm kiếm sản phẩm	Tên sản phẩm, từ khóa	Nhập từ khóa vào ô tìm kiếm	Hiển thị sản phẩm phù hợp	Đạt

Use case: Quản lý đơn hàng

Bảng 4.6 Test case use case Quản lý đơn hàng

STT	Mô tả	Dữ liệu đầu vào	Bước thực hiện	Kết quả mong đợi	Trạng thái
TC06					
1	Xem danh sách đơn hàng	Quyền admin hoặc nhân viên	Truy cập mục “Đơn hàng”	Hiển thị danh sách đơn hàng	Đạt
2	Cập nhật trạng thái đơn hàng	ID đơn hàng, trạng thái mới	Chọn đơn hàng > Chỉnh sửa trạng thái	Trạng thái được cập nhật	Đạt
3	Xem chi tiết đơn hàng	ID đơn hàng	Click vào mã đơn	Hiển thị chi tiết sản phẩm và thông tin khách hàng	Đạt

KẾT LUẬN

Với kiến thức nền tảng đã được học tại trường cùng với sự nỗ lực, tìm tòi của bản thân, em đã hoàn thành đồ án với đề tài **“Xây dựng website thương mại giới thiệu và bán sách sử dụng nodejs”**. Mặc dù đã cố gắng hết sức trong quá trình thực hiện, nhưng do thời gian và kiến thức còn hạn chế, hệ thống vẫn còn một số điểm chưa thực sự hoàn thiện. Em rất mong nhận được sự góp ý, nhận xét từ quý thầy cô để có thể tiếp tục hoàn thiện và nâng cao chất lượng của hệ thống trong thời gian tới.

Kết quả đạt được:

Về nội dung:

- Trưng bày sản phẩm và hỗ trợ mua sách trực tuyến.
- Tìm kiếm sách theo tên, danh mục.
- Quản lý tài khoản người dùng (thành viên).
- Quản lý danh mục sách.
- Quản lý thông tin sách (thêm, sửa, xóa, cập nhật).
- Chức năng giỏ hàng trực tuyến.
- Quản lý và xử lý đơn đặt hàng.

Về hình thức:

- Giao diện thân thiện, dễ sử dụng và phù hợp với nhiều đối tượng.
- Quy trình đặt hàng đơn giản, tiết kiệm thời gian.
- Trình bày sách rõ ràng, có hình ảnh minh họa và thông tin đầy đủ.

Hướng phát triển trong tương lai:

- Cải tiến giao diện theo hướng hiện đại hơn, phù hợp với thị hiếu người dùng. Phát triển thêm tính năng sao lưu, phục hồi dữ liệu trong trường hợp hệ thống gặp sự cố.
- Ứng dụng các nguyên tắc UI/UX nâng cao trải nghiệm người dùng. Tích hợp chức năng hỗ trợ khách hàng trực tuyến như chat live, chatbot tự động.

TÀI LIỆU THAM KHẢO

Tiếng Việt

- [1] Nguyễn Hữu Thái (2020). *Tài liệu UML cơ bản*. Đại học Công nghệ Thông tin – ĐHQG TP.HCM.
- [2] Nguyễn Tấn Trần Minh Khang (2015). *Phân tích và thiết kế hệ thống thông tin hướng đối tượng*. NXB Đại học Quốc gia TP.HCM.
- [3] Nguyễn Văn Hiệp (2018). *Lập trình hướng đối tượng với JavaScript*. Đại học FPT.

Tiếng Anh

- [4] Flanagan, D. (2020). *JavaScript: The Definitive Guide* (7th ed.). O'Reilly Media.
- [5] Larman, C. (2004). *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development*. Prentice Hall.
- [6] W3C. (2021). *HTML5 Specification*. <https://www.w3.org/TR/html52/> (Lần truy cập cuối cùng 30/04/2025)
- [7] MDN Web Docs. *CSS: Cascading Style Sheets*. <https://developer.mozilla.org/en-US/docs/Web/CSS> (Lần truy cập cuối cùng 29/04/2025)
- [8] Tilkov, S., & Vinoski, S. (2010). *Node.js: Using JavaScript to Build High-Performance Network Programs*. *IEEE Internet Computing*.
- [9] Express.js Guide (2024). <https://expressjs.com> (Lần truy cập cuối cùng 01/05/2025)