

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

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



ĐỒ ÁN TỐT NGHIỆP
NGÀNH KHOA HỌC MÁY TÍNH

ĐỀ TÀI: XÂY DỰNG WEBSITE GIỚI THIỆU VÀ BÁN
HÀNG CHO NHÀ SÁCH QUANG VŨ

Giảng viên hướng dẫn : ThS. Trần Thanh Huân

Sinh viên thực hiện : Nguyễn Quang Vũ

Mã sinh viên : 2021608053

Hà Nội – 2025

MỤC LỤC

| | |
|---|-----|
| MỤC LỤC..... | i |
| LỜI CẢM ƠN | iii |
| DANH MỤC CHỮ VIẾT TẮT | iv |
| DANH MỤC HÌNH ẢNH | vi |
| DANH MỤC BẢNG BIỂU | ix |
| LỜI MỞ ĐẦU | 1 |
| CHƯƠNG 1: MỘT SỐ CÔNG NGHỆ XÂY DỰNG WEBSITE..... | 3 |
| 1.1. Phân tích, thiết kế và lập trình hướng đối tượng | 3 |
| 1.1.1. Phân tích hướng đối tượng..... | 3 |
| 1.1.2. Thiết kế hướng đối tượng | 3 |
| 1.1.3. Lập trình hướng đối tượng..... | 4 |
| 1.2. Ngôn ngữ mô hình hóa UML..... | 5 |
| 1.3. Ngôn ngữ phát triển ứng dụng website..... | 7 |
| 1.3.1. NodeJS | 7 |
| 1.3.2. Javascript..... | 9 |
| 1.3.3. Bootstrap | 10 |
| 1.3.4. HTML và CSS | 11 |
| 1.3.5. ReactJS..... | 12 |
| 1.4. Mô hình Client – Server..... | 14 |
| 1.5. Hệ quản trị cơ sở dữ liệu..... | 16 |
| 1.5.1. Các khái niệm | 16 |
| 1.5.2. MongoDB | 17 |
| CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG..... | 21 |
| 2.1. Khảo sát hệ thống..... | 21 |
| 2.2. Mô tả bài toán | 22 |
| 2.3. Xác định các yêu cầu của hệ thống..... | 22 |
| 2.4. Thiết kế hệ thống..... | 23 |
| 2.4.1. Các tác nhân của hệ thống | 23 |
| 2.4.2. Các use case trong hệ thống..... | 24 |
| 2.4.3. Biểu đồ lớp..... | 28 |

| | |
|---|-----------|
| 2.4.4. Biểu đồ hoạt động | 29 |
| 2.4.5. Biểu đồ trình tự | 43 |
| 2.4.6. Các yêu cầu về dữ liệu | 63 |
| 2.4.7. Mô hình thực thể - quan hệ | 67 |
| CHƯƠNG 3: CÀI ĐẶT VÀ KIỂM THỬ HỆ THỐNG | 68 |
| 3.1. Cài đặt môi trường | 68 |
| 3.1.1. Môi trường hoạt động | 68 |
| 3.1.2. Cài đặt môi trường phát triển | 68 |
| 3.1.3. Giao diện hệ thống | 69 |
| 3.2. Kiểm thử hệ thống | 75 |
| 3.2.1. Khái niệm | 75 |
| 3.2.2. Các lĩnh vực kiểm thử hệ thống | 76 |
| 3.2.3. Quy trình kiểm thử hệ thống | 76 |
| 3.3. Thiết kế các test case cho hệ thống | 77 |
| KẾT LUẬN | 81 |
| TÀI LIỆU THAM KHẢO | 82 |

LỜI CẢM ƠN

Lời đầu tiên cho phép em gửi lời cảm ơn sâu sắc tới các thầy cô trong khoa Công nghệ thông tin - Trường Đại học Công Nghiệp Hà Nội, những người đã hết mình truyền đạt và chỉ dẫn cho em những kiến thức, những bài học quý báu và bổ ích. Đặc biệt em xin được bày tỏ sự tri ân và xin chân thành cảm ơn giảng viên Thạc sĩ Trần Thanh Huân, người trực tiếp hướng dẫn, chỉ bảo em trong suốt quá trình học tập, nghiên cứu và hoàn thành được đồ án tốt nghiệp. Sau nữa, em xin gửi tình cảm sâu sắc tới gia đình và bạn bè vì đã luôn bên cạnh khuyến khích, động viên, giúp đỡ cả về vật chất lẫn tinh thần cho em trong suốt quy trình học tập để em hoàn thành tốt việc học tập của bản thân.

Trong quá trình nghiên cứu và làm đề tài, do năng lực, kiến thức, trình độ bản thân em còn hạn hẹp nên không tránh khỏi những thiếu sót và em mong mọi nhận được sự thông cảm và những góp ý từ quý thầy cô cũng như các bạn trong lớp.

Em xin trân trọng cảm ơn!

Sinh Viên thực hiện

Nguyễn Quang Vũ

DANH MỤC CHỮ VIẾT TẮT

| Từ viết tắt | Giải nghĩa |
|-----------------|--|
| UML | Unified Modeling Language – Ngôn ngữ mô hình hóa hợp nhất |
| OOA | Object-Oriented Analysis – Phân tích hướng đối tượng |
| OOD | Object-Oriented Design – Thiết kế hướng đối tượng |
| OOP | Object-Oriented Programming – Lập trình hướng đối tượng |
| SRP | Single Responsibility Principle – Nguyên tắc đơn nhiệm |
| LSP | Liskov Substitution Principle – Nguyên tắc thay thế Liskov |
| ISP | Interface Segregation Principle – Nguyên tắc phân tách giao diện |
| DIP | Dependency Inversion Principle – Nguyên tắc đảo ngược phụ thuộc |
| MVC | Model - View - Controller – Mô hình kiến trúc phần mềm |
| REST API | Representational State Transfer Application Programming Interface – Giao diện lập trình ứng dụng kiểu REST |
| DBMS | Database Management System – Hệ quản trị cơ sở dữ liệu |
| RDBMS | Relational Database Management System – Hệ quản trị cơ sở dữ liệu quan hệ |
| NoSQL | Not Only SQL – Cơ sở dữ liệu phi quan hệ |
| JSON | JavaScript Object Notation – Định dạng dữ liệu dạng đối tượng |

| | |
|---------------|--|
| SPA | Single Page Application – Ứng dụng một trang |
| JWT | JSON Web Token – Chuỗi mã hóa xác thực người dùng |
| API | Application Programming Interface – Giao diện lập trình ứng dụng |
| HTML | HyperText Markup Language – Ngôn ngữ đánh dấu siêu văn bản |
| CSS | Cascading Style Sheets – Ngôn ngữ định kiểu trang |
| JSX | JavaScript XML – Cú pháp mở rộng của JavaScript |
| TCP/IP | Transmission Control Protocol / Internet Protocol – Giao thức truyền tải và giao thức Internet |
| HTTPS | HyperText Transfer Protocol Secure – Giao thức truyền tải siêu văn bản bảo mật |
| SASS | Syntactically Awesome Stylesheets – Tiền xử lý CSS |
| NPM | Node Package Manager – Trình quản lý gói của Node.js |
| UI | User Interface – Giao diện người dùng |
| UX | User Experience – Trải nghiệm người dùng |
| DOM | Document Object Model |
| JS | JavaScript |
| SQL | Structured Query Language |
| OS | Operating System – Hệ điều hành |
| PHP | Hypertext Preprocessor |
| FTP | File Transfer Protocol |
| RAM | Random Access Memory |
| TCP | Transmission Control Protocol |
| IP | Internet Protocol |
| REST | Representational State Transfer |
| XHTML | Extensible HyperText Markup Language |
| GIS | Geographic Information System |

DANH MỤC HÌNH ẢNH

| | |
|---|----|
| Hình 1.1: NodeJS | 7 |
| Hình 1.2: HTML và CSS..... | 11 |
| Hình 1.3: ReactJS..... | 12 |
| Hình 1.4: Virtual DOM | 12 |
| Hình 1.5: Mô Hình Client - Server | 14 |
| Hình 1.6: Nguyên tắc hoạt động của mô hình Client Server | 15 |
| Hình 1.7: MongoDB NoSQL | 18 |
| Hình 1.8: Nexus Architecture..... | 20 |
| Hình 2.1: Sơ đồ use case tổng quan hệ thống | 24 |
| Hình 2.2: Sơ đồ use case Khách hàng – Khách hàng Tiềm năng | 25 |
| Hình 2.3: Sơ đồ use case Nhân viên quản lý..... | 26 |
| Hình 2.4: Biểu đồ lớp hệ thống..... | 28 |
| Hình 2.5: Biểu đồ hoạt động use case Đăng nhập | 29 |
| Hình 2.6: Biểu đồ hoạt động use case Đăng ký | 30 |
| Hình 2.7: Biểu đồ hoạt động use case Thêm sản phẩm vào giỏ hàng | 31 |
| Hình 2.8: Biểu đồ hoạt động use case Quản lý giỏ hàng | 32 |
| Hình 2.9: Biểu đồ hoạt động use case Đặt hàng | 33 |
| Hình 2.10: Biểu đồ hoạt động use case Quản lý đơn hàng cá nhân..... | 34 |
| Hình 2.11: Biểu đồ hoạt động use case Quản lý tài khoản | 35 |
| Hình 2.12: Biểu đồ hoạt động use case Quản lý sản phẩm..... | 36 |
| Hình 2.13: Biểu đồ hoạt động use case Quản lý đơn hàng | 37 |
| Hình 2.14: Biểu đồ hoạt động use case Quản lý kho hàng | 38 |
| Hình 2.15: Biểu đồ hoạt động use case Quản lý nhà xuất bản | 39 |
| Hình 2.16: Biểu đồ hoạt động use case Quản lý thể loại | 40 |
| Hình 2.17: Biểu đồ hoạt động use case Quản lý tác giả | 41 |
| Hình 2.18: Biểu đồ hoạt động use case Thống kê..... | 42 |
| Hình 2.19: Biểu đồ trình tự use case đăng nhập | 43 |
| Hình 2.20: Biểu đồ trình tự use case đăng ký | 43 |

| | |
|---|----|
| Hình 2.21: Biểu đồ trình tự use case đăng xuất | 44 |
| Hình 2.22: Biểu đồ trình tự use case xem chi tiết sản phẩm..... | 44 |
| Hình 2.23: Biểu đồ trình tự use case Xem sản phẩm theo tiêu chí | 45 |
| Hình 2.24: Biểu đồ trình tự use case Tìm kiếm sản phẩm..... | 45 |
| Hình 2.25: Biểu đồ trình tự use case Quản lý giỏ hàng | 46 |
| Hình 2.26: Biểu đồ trình tự use case Đặt hàng | 47 |
| Hình 2.27: Biểu đồ trình tự use case Quản lý đơn hàng cá nhân..... | 48 |
| Hình 2.28: Biểu đồ trình tự use case Đánh giá sản phẩm..... | 49 |
| Hình 2.29: Biểu đồ trình tự use case Quản lý thông tin cá nhân | 49 |
| Hình 2.30: Biểu đồ trình tự use case Quản lý tài khoản (1)..... | 50 |
| Hình 2.31: Biểu đồ trình tự use case Quản lý tài khoản (2)..... | 51 |
| Hình 2.32: Biểu đồ trình tự use case Quản lý thẻ loại (1) | 52 |
| Hình 2.33: Biểu đồ trình tự use case Quản lý thẻ loại (2) | 53 |
| Hình 2.34: Biểu đồ trình tự use case Quản lý tác giả (1)..... | 54 |
| Hình 2.35: Biểu đồ trình tự use case Quản lý tác giả (2)..... | 55 |
| Hình 2.36: Biểu đồ trình tự use case Quản lý nhà xuất bản (1)..... | 56 |
| Hình 2.37: Biểu đồ trình tự use case Quản lý nhà xuất bản (2)..... | 57 |
| Hình 2.38: Biểu đồ trình tự use case Quản lý sản phẩm (1) | 58 |
| Hình 2.39: Biểu đồ trình tự use case Quản lý sản phẩm (2) | 59 |
| Hình 2.40: Biểu đồ trình tự use case Quản lý đơn hàng | 60 |
| Hình 2.41: Biểu đồ trình tự use case Quản lý kho hàng | 61 |
| Hình 2.42: Biểu đồ trình tự use case Thống kê..... | 62 |
| Hình 2.43: Mô hình thực thể liên kết hệ thống | 67 |
| Hình 3.1: Giao diện đăng ký | 69 |
| Hình 3.2: Giao diện đăng nhập | 69 |
| Hình 3.3: Giao diện chi tiết sản phẩm..... | 69 |
| Hình 3.4: Giao diện giỏ hàng | 70 |
| Hình 3.5: Giao diện bộ lọc sản phẩm..... | 70 |
| Hình 3.6: Giao diện voucher khuyến mãi | 70 |

| | |
|--|----|
| Hình 3.7: Giao diện thanh toán | 71 |
| Hình 3.8: Giao diện thông tin cá nhân | 71 |
| Hình 3.9: Giao diện đơn hàng cá nhân..... | 72 |
| Hình 3.10: Giao diện địa chỉ nhận hàng..... | 72 |
| Hình 3.11: Giao diện quản lý sản phẩm..... | 72 |
| Hình 3.12: Giao diện quản lý kho hàng | 73 |
| Hình 3.13: Giao diện quản lý người dùng..... | 73 |
| Hình 3.14: Giao diện quản lý chương trình khuyến mãi | 73 |
| Hình 3.15: Giao diện quản lý thể loại | 74 |
| Hình 3.16: Giao diện quản lý đơn hàng | 74 |
| Hình 3.17: Giao diện thống kê | 74 |

DANH MỤC BẢNG BIỂU

| | |
|---|----|
| Bảng 2.1: Collection User | 63 |
| Bảng 2.2: Collection Product | 63 |
| Bảng 2.3: Collection Order | 64 |
| Bảng 2.4: Collection Category | 65 |
| Bảng 2.5: Collection Author | 65 |
| Bảng 2.6: Collection Publisher | 65 |
| Bảng 2.7: Collection StockReceipt | 65 |
| Bảng 2.8: Collection StatusHistory..... | 66 |
| Bảng 2.9: Collection Supplier | 66 |
| Bảng 3.1: Test case use case Đăng nhập | 77 |
| Bảng 3.2: Test case use case Đăng ký..... | 78 |
| Bảng 3.3: Test case use case Quản lý tài khoản..... | 78 |
| Bảng 3.4: Test case use case Quản lý sản phẩm | 79 |

LỜI MỞ ĐẦU

Trong kỷ nguyên số hóa hiện nay, thương mại điện tử đã trở thành động lực không thể thiếu của nền kinh tế toàn cầu. Đặc biệt, ngành kinh doanh sách trực tuyến đang phát triển mạnh mẽ, không chỉ góp phần lan tỏa tri thức mà còn đáp ứng nhu cầu tiếp cận thông tin nhanh chóng của độc giả hiện đại. Tuy nhiên, nhiều nền tảng bán sách trực tuyến hiện tại còn hạn chế về trải nghiệm người dùng và khả năng mở rộng hệ thống.

Nhận thấy tiềm năng to lớn và nhu cầu cấp thiết này, đề tài "*Xây dựng website giới thiệu và bán hàng cho nhà sách*" ra đời. Node.js với ưu điểm xử lý không đồng bộ hiệu quả và hệ sinh thái phong phú, là lựa chọn lý tưởng để phát triển một nền tảng linh hoạt, an toàn và dễ mở rộng cho hệ thống thương mại sách trực tuyến.

Phân tích và thiết kế hệ thống website bán sách trực tuyến đáp ứng nhu cầu người dùng hiện đại

- Xây dựng nền tảng backend sử dụng Node.js với hiệu năng cao và khả năng mở rộng
- Phát triển giao diện người dùng thân thiện, tương thích đa nền tảng và thiết bị
- Tích hợp các tính năng quản lý sách, quản lý đơn hàng, thanh toán trực tuyến và báo cáo thống kê
- Đảm bảo tính bảo mật và an toàn cho dữ liệu người dùng và giao dịch trực tuyến

Đối tượng nghiên cứu của đề tài là các công nghệ, kiến trúc và giải pháp liên quan đến việc phát triển hệ thống thương mại điện tử, cụ thể là: Nền tảng Node.js và hệ sinh thái các thư viện hỗ trợ (Express.js, Mongoose...); Mô hình kiến trúc ứng dụng web hiện đại (MVC, REST API); Cơ sở dữ liệu NoSQL (MongoDB) và quản lý dữ liệu cho hệ thống bán sách; Các giải pháp bảo mật và xác thực người dùng trên nền tảng web; Quy trình quản lý và vận hành website bán sách trực tuyến.

Đề tài được nghiên cứu và triển khai trong khoảng thời gian 9 tuần. Tập trung xây dựng hệ thống phục vụ thị trường bán sách trực tuyến tại Việt Nam với lĩnh vực: thương mại điện tử, công nghệ phát triển web fullstack và quản lý sách điện tử. Công nghệ: Tập trung vào nền tảng Node.js và các công nghệ liên quan (Express, MongoDB, React/Angular/Vue cho frontend)

Kết thúc quá trình nghiên cứu với kết quả mong muốn đạt được là:

- Một hệ thống website bán sách trực tuyến hoàn chỉnh với đầy đủ chức năng cơ bản
- Báo cáo chi tiết về quá trình phân tích, thiết kế và triển khai hệ thống
- Tài liệu hướng dẫn sử dụng và quản trị hệ thống
- Bộ mã nguồn hoàn thiện với các tính năng bảo mật được tích hợp
- Giải pháp kỹ thuật có thể mở rộng và phát triển thêm trong tương lai

Cấu trúc của báo cáo

Báo cáo này là kết quả của một quá trình nghiên cứu và thực hành nghiêm túc, bao gồm các bước từ phân tích yêu cầu, thiết kế cơ sở dữ liệu, xây dựng giao diện, phát triển tính năng chính, đến kiểm thử và đánh giá hiệu quả của hệ thống. Sản phẩm cuối cùng không chỉ hướng đến việc số hóa trải nghiệm người dùng mà còn hỗ trợ người quản trị trong việc vận hành và quản lý hệ thống một cách hiệu quả. Nội dung đồ án gồm 3 chương:

Chương 1: Một số công nghệ sử dụng website

Nêu các cơ sở lý thuyết và các công nghệ sẽ áp dụng vào hệ thống phần mềm sẽ triển khai, các ngôn ngữ lập trình và hệ quản trị cơ sở dữ liệu.

Chương 2: Phân tích và thiết kế hệ thống

Thực hiện tìm hiểu và khảo sát hệ thống, từ đó trình bày các sơ đồ, mô hình use case, biểu đồ tuần tự... của các chức năng. Đồng thời, tiến hành thiết kế cơ sở dữ liệu cho hệ thống.

Chương 3: Cài đặt và kiểm thử hệ thống

Từ những phân tích và thiết kế đã đưa ra, sau khi hoàn thiện xây dựng, áp dụng triển khai và kiểm thử các chức năng của hệ thống.

CHƯƠNG 1: MỘT SỐ CÔNG NGHỆ XÂY DỰNG WEBSITE

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

1.1.1. Phân tích hướng đối tượng

Phân tích hướng đối tượng (Object-Oriented Analysis - OOA) là giai đoạn đầu tiên trong quy trình phát triển phần mềm, tập trung vào việc hiểu rõ bài toán và mô hình hóa các thực thể thực tế thành các đối tượng phần mềm.

Khác với phương pháp truyền thống, OOA không chỉ xác định chức năng mà còn phân tích các đối tượng liên quan, các thuộc tính và mối quan hệ giữa chúng. Việc này giúp xây dựng mô hình gần với thực tế, dễ hiểu với cả người dùng cuối và lập trình viên.

Một số nguyên tắc quan trọng trong thiết kế hướng đối tượng:

- *Nguyên tắc đơn nhiệm (Single Responsibility Principle - SRP)*: Mỗi lớp nên chỉ có một trách nhiệm duy nhất. Điều này giúp lớp có độ tập trung cao và dễ bảo trì.
- *Nguyên tắc mở/đóng (Open/Closed Principle)*: Các lớp nên được mở rộng để thêm chức năng mới mà không cần thay đổi mã nguồn hiện tại.
- *Nguyên tắc thay thế Liskov (Liskov Substitution Principle - LSP)*: Lớp con phải có thể thay thế hoàn toàn lớp cha mà không làm ảnh hưởng đến logic của chương trình.
- *Nguyên tắc phân tách giao diện (Interface Segregation Principle - ISP)*: Các lớp không nên bị ép phải triển khai những phương thức không sử dụng. Nên chia nhỏ các interface để phù hợp với từng lớp cụ thể.
- *Nguyên tắc đảo ngược phụ thuộc (Dependency Inversion Principle - DIP)*: Các thành phần cấp cao không nên phụ thuộc trực tiếp vào các thành phần cấp thấp, mà cả hai nên phụ thuộc vào abstraction.

1.1.2. Thiết kế hướng đối tượng

Thiết kế hướng đối tượng (Object-Oriented Design – OOD) là giai đoạn tiếp theo của phân tích hướng đối tượng (OOA), đóng vai trò chuyển hóa các mô hình khái niệm trừu tượng thành bản thiết kế chi tiết có thể triển khai thành

mã nguồn thực tế. Đây là bước quan trọng nhằm đảm bảo hệ thống được phát triển theo đúng định hướng đã xác lập, đồng thời có khả năng mở rộng, bảo trì và tái sử dụng cao.

Trong quá trình thiết kế, hệ thống phần mềm được tổ chức thành các lớp (classes), mỗi lớp đại diện cho một thực thể trong hệ thống. Mỗi lớp sẽ bao gồm các thuộc tính (attributes) để mô tả dữ liệu và các phương thức (methods) để thực hiện các hành vi. Các lớp có thể được phân loại theo vai trò hoặc chức năng, từ đó hình thành nên một hệ thống các thành phần logic rõ ràng và chặt chẽ.

Một trong những mục tiêu chính của thiết kế hướng đối tượng là xây dựng các lớp có độ liên kết nội bộ cao (high cohesion) và liên kết giữa các lớp thấp (low coupling). Điều này giúp hệ thống dễ bảo trì, thay đổi một phần không ảnh hưởng đến toàn bộ ứng dụng. Ngoài ra, các mối quan hệ giữa các lớp như kế thừa (inheritance), kết hợp (association), phụ thuộc (dependency), và kết tập (aggregation) cũng cần được xác định rõ ràng để đảm bảo tính nhất quán và chính xác trong cấu trúc hệ thống.

1.1.3. Lập trình hướng đối tượng

Lập trình hướng đối tượng (Object-Oriented Programming – OOP) là một phương pháp lập trình hiện đại, dựa trên mô hình các "đối tượng" mô phỏng các thực thể trong thế giới thực. Mỗi đối tượng bao gồm dữ liệu (thuộc tính) và các thao tác xử lý dữ liệu đó (phương thức). OOP giúp tổ chức chương trình thành các thành phần có tính độc lập cao, hỗ trợ tốt cho việc phát triển, mở rộng và bảo trì phần mềm.

OOP hiện diện trong hầu hết các ngôn ngữ lập trình phổ biến như Java, C++, Python, PHP và JavaScript. Đặc biệt, JavaScript trong môi trường Node.js hay trình duyệt hiện đại cũng hỗ trợ lập trình hướng đối tượng với cú pháp rõ ràng và linh hoạt (class, constructor, inheritance,...).

Bốn đặc trưng cốt lõi của OOP gồm:

- *Tính đóng gói (Encapsulation)*: Giúp ẩn giấu thông tin bên trong đối tượng và chỉ cho phép truy cập thông qua các phương thức được định nghĩa trước. Điều này làm tăng tính bảo mật và khả năng kiểm soát truy cập dữ liệu.
- *Tính kế thừa (Inheritance)*: Cho phép lớp con kế thừa các thuộc tính và phương thức từ lớp cha. Cơ chế này giúp giảm thiểu việc viết lại mã và tăng tính tái sử dụng.
- *Tính đa hình (Polymorphism)*: Cho phép các đối tượng có thể xử lý theo nhiều cách khác nhau tùy thuộc vào ngữ cảnh. Tính đa hình tạo điều kiện thuận lợi cho việc mở rộng hệ thống và thiết kế linh hoạt.
- *Tính trừu tượng (Abstraction)*: Giúp ẩn đi những chi tiết phức tạp và chỉ hiển thị những gì cần thiết đối với người dùng, giúp hệ thống dễ sử dụng và dễ phát triển.

Việc vận dụng OOP vào phát triển phần mềm giúp chia nhỏ hệ thống thành các mô-đun có khả năng độc lập cao, dễ kiểm thử và dễ bảo trì. Đồng thời, OOP cũng là tiền đề để áp dụng các mẫu thiết kế (Design Patterns) và mô hình kiến trúc phần mềm hiện đại như MVC (Model - View - Controller).

1.2. Ngôn ngữ mô hình hóa UML

UML (Unified Modeling Language – Ngôn ngữ mô hình hóa hợp nhất) là một ngôn ngữ chuẩn được sử dụng để mô hình hóa, trực quan hóa, xây dựng và lập tài liệu cho các hệ thống phần mềm. UML được phát triển nhằm mục tiêu thống nhất các phương pháp biểu diễn phần mềm hướng đối tượng và hiện đã trở thành tiêu chuẩn công nghiệp trong thiết kế hệ thống phần mềm hiện đại.

Khác với các ngôn ngữ lập trình truyền thống, UML không nhằm mục đích mô tả chi tiết mã nguồn, mà tập trung mô hình hóa cấu trúc và hành vi của hệ thống ở mức trừu tượng. Việc sử dụng UML giúp các bên liên quan – bao gồm nhà phát triển, nhà phân tích hệ thống, kiểm thử viên và khách hàng – có thể hiểu được thiết kế hệ thống một cách trực quan, rõ ràng và thống nhất.

+ Mục tiêu của UML

- Hỗ trợ mô hình hóa toàn diện các cấu trúc và hành vi của hệ thống.
- Tạo ra ngôn ngữ chung trong nhóm phát triển phần mềm.
- Cung cấp công cụ hỗ trợ thiết kế, phân tích và kiểm thử hệ thống.
- Tạo tài liệu chính thức cho phần mềm trong suốt vòng đời phát triển.

+ Một số sơ đồ UML phổ biến bao gồm:

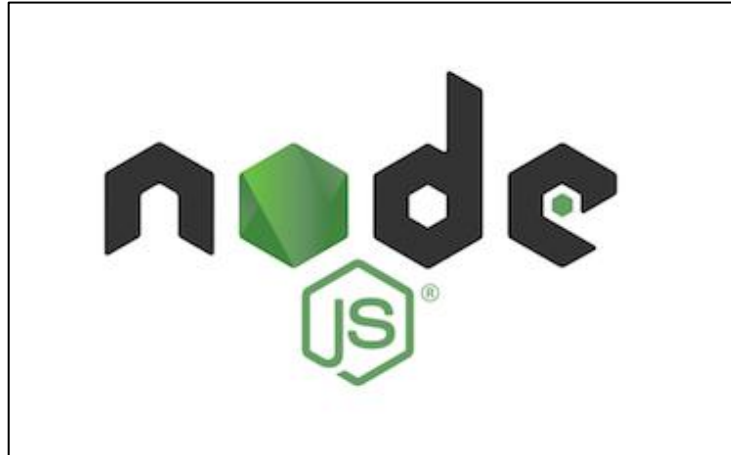
- *Biểu đồ lớp (Class Diagram)*: Mô tả cấu trúc tĩnh của hệ thống, thể hiện các lớp, thuộc tính, phương thức và mối quan hệ giữa các lớp như kế thừa, liên kết, phụ thuộc,...
- *Biểu đồ gói (Package Diagram)*: Giúp nhóm các lớp có liên quan vào một gói (package), thuận tiện cho việc tổ chức và quản lý mã nguồn.
- *Biểu đồ chức năng (Use Case Diagram)*: Mô tả chức năng của hệ thống. Biểu đồ này xác định các tác nhân (actor) và các trường hợp sử dụng (use case) mà hệ thống cần đáp ứng.
- *Biểu đồ hoạt động (Activity Diagram)*: Diễn tả luồng công việc hoặc các bước xử lý trong một quy trình nghiệp vụ cụ thể. Biểu đồ này thường dùng để mô hình hóa thuật toán, logic nghiệp vụ hoặc quy trình xử lý đơn hàng, đăng ký, thanh toán,...
- *Biểu đồ tuần tự (Sequence Diagram)*: Minh họa thứ tự tương tác giữa các đối tượng theo thời gian để thực hiện một chức năng cụ thể. Biểu đồ này giúp mô tả cách các đối tượng gọi lẫn nhau để hoàn thành nghiệp vụ.

UML không chỉ giúp định hình kiến trúc hệ thống rõ ràng mà còn là tài liệu hữu ích trong suốt vòng đời phát triển phần mềm, từ phân tích, thiết kế, lập trình đến kiểm thử và bảo trì.

1.3. Ngôn ngữ phát triển ứng dụng website

1.3.1. NodeJS

NodeJS là một mã nguồn mở, đa nền tảng, chạy trên môi trường JavaScript, được xây dựng trên V8 JavaScript engine của Chrome - V8 thực thi mã JavaScript bên ngoài trình duyệt. Nó được tạo ra vào năm 2009 đi kèm với một lợi thế chính - NodeJS cho phép thực hiện lập trình bất đồng bộ.



Hình 1.1: NodeJS

Ở chế độ đồng bộ thực thi từng dòng và tiến hành thực thi dòng tiếp theo khi dòng hiện tại đã thực thi xong. Khi bất đồng bộ thực thi tất cả dòng code cùng một lúc.

NodeJS là một nền tảng được xây dựng trên JavaScript runtime của Chrome với mục đích xây dựng các ứng dụng mạng nhanh chóng và có thể mở rộng được một cách dễ dàng hơn. NodeJS sử dụng mô hình I/O lập trình theo sự kiện, non-blocking, do đó nodeJS khá gọn nhẹ và hiệu quả - công cụ hoàn hảo cho các ứng dụng chuyên sâu về dữ liệu theo thời gian thực chạy trên các thiết bị phân tán.

NodeJS là môi trường runtime mã nguồn mở đa nền tảng, được sử dụng để phát triển các ứng dụng mạng và ứng dụng server-side. Các ứng dụng NodeJS được viết bằng JavaScript và có thể chạy trong NodeJS runtime trên OS X, Microsoft Windows và Linux.

NodeJS cũng cung cấp một thư viện bao gồm rất nhiều các module JavaScript khác nhau nhằm đơn giản hóa việc phát triển các ứng dụng web, qua đó giảm thiểu tình trạng sử dụng quá nhiều Node.js.

*** Ưu điểm**

- + Hiệu suất cao với single-thread: Xử lý nhiều kết nối cùng lúc, tiết kiệm RAM và tăng tốc độ phản hồi.
- + Non-blocking I/O: Thực hiện nhiều tác vụ đồng thời không gây chậm trễ, tối ưu cho website có nhiều người dùng.
- + Event-driven: Xử lý hiệu quả các sự kiện như cập nhật giỏ hàng, thông báo sản phẩm mới theo thời gian thực.
- + Phù hợp cho SPA: Tối ưu cho website bán sách hiện đại không cần tải lại trang khi chuyển đổi chức năng.
- + Ứng dụng thời gian thực: Dễ dàng tích hợp chat hỗ trợ, thông báo hàng mới, cập nhật trạng thái đơn hàng.
- + JavaScript đồng nhất: Dùng cùng ngôn ngữ ở cả frontend và backend, tăng hiệu quả phát triển.

*** Nhược điểm**

- + Hạn chế với tác vụ nặng: Không tối ưu cho xử lý hình ảnh sách số lượng lớn hoặc báo cáo phức tạp.
- + Vấn đề với package: Nhiều NPM package thiếu kiểm duyệt chất lượng, tiềm ẩn rủi ro bảo mật.
- + Triển khai phức tạp: Một số hosting chưa hỗ trợ tốt, đòi hỏi kiến thức server chuyên sâu.
- + Callback hell: Có thể gây khó khăn khi xử lý quy trình phức tạp như thanh toán nhiều bước.
- + API không ổn định: Thay đổi giữa các phiên bản, gây khó khăn khi nâng cấp website trong tương lai.

1.3.2. Javascript

JavaScript được phát minh bởi Brendan Eich của Netscape (với trình duyệt Navigator 2.0) vào năm 1995, và xuất hiện ở hầu hết các công cụ trình duyệt của Netscape và Microsoft kể từ năm 1996.

Ngày nay, gần như các trình duyệt đình đám đều có hỗ trợ Javascript, và chính Javascript là nhân tố gần đây đã châm ngòi cho cuộc chiến các trình duyệt. Javascript đã ra đời từ khá lâu, nhưng cho đến ngày nay, nó vẫn mang 1 tầm quan trọng rất lớn, vì vậy học Javascript là điều cần thiết.

Về phương diện thực thi ngôn ngữ, hiện nay khi phát triển các trang web động, có hai phương thức rất phổ biến đó là Server-Side và Client-Side.

+ **Client-Side**: là phương thức sẽ xử lý các mã lệnh bởi mỗi browser riêng biệt tại máy người dùng, sau đó gửi kết quả lên cho Server.

+ **Server-Side**: thì lại gửi các mã lệnh cho server xử lý trước, sau đó server gửi kết quả lại cho browser.

JavaScript là một ngôn ngữ lập trình kịch bản Client-Side (Client-Side Script Language), và dĩ nhiên nó được thực thi tại máy người dùng. Và vì javascript là một Client-Side Script Language nên nó không có khả năng kết nối và thao tác với CSDL trên Server.

Về phương diện phong cách ngôn ngữ lập trình: Các tài liệu cũ trước đây nói Javascript là ngôn ngữ lập trình dựa vào đối tượng (Object-based language). Nhưng từ lúc Ajax ra đời, Javascript trở nên hướng đối tượng hơn rất nhiều. Vì vậy, chúng ta có thể coi Javascript là một ngôn ngữ lập trình hướng đối tượng (Object-Oriented programming). Về cú pháp, Javascript cũng tương tự như C, Perl và Java, ...ví dụ mệnh đề lặp if, while, for, tương tự nhau.

* Ưu điểm của Javascript:

+ Hoàn toàn miễn phí và dễ học. JavaScript được thiết kế độc lập với hệ điều hành. Nó có thể chạy trên bất kỳ hệ điều hành nào có trình duyệt hỗ trợ JavaScript.

+ Dễ dàng tương tác, điều khiển và tránh bớt việc xử lý từ phía server.
 Nắm vững kiến thức JavaScript bây giờ rất hữu dụng cho các bạn sau này để có thể tiếp thu những công nghệ mới mà nó được gói gọn vào những ngôn ngữ như : Ajax, Atlas

*** Nhược điểm của Javascript:**

+ Javascript không có trình biên dịch riêng mà được diễn dịch và chạy bởi trình duyệt hỗ trợ nó. Chính vì thế, nếu trình duyệt không hỗ trợ, hoặc không bật JS, nó sẽ không chạy được.

+ JavaScript có thể làm ứng dụng web của bạn trở nên nặng nề hơn. Bảo mật kém. Không có khả năng giấu mã.

1.3.3. Bootstrap

Bootstrap là một Font-end Framework được viết bằng SASS và biên dịch thành CSS, nó là một bộ thư viện mạnh mẽ được tích hợp nhiều CSS có sẵn giúp cho việc lập trình HTML & CSS trở nên đơn giản và dễ dàng hơn. Thông thường khi chúng ta lập trình bằng CSS thuần thì công việc kiểm tra tính tương thích trên các trình duyệt khác nhau và trên các thiết bị di động khác nhau rất là khó khăn, nhưng với Bootstrap thì lập trình viên không cần phải tốn nhiều công sức nữa vì mọi thứ đã có người tạo nên Bootstrap lo.

Bootstrap chia layout của một trang web ra thành 960 Grid và gồm 12 cột, mỗi cột 80Grid và đây chính là chuẩn thiết kế HTML & CSS của Bootstrap. Nó có hỗ trợ hầu hết các module của một trang web như menu, tabs, tooltip, popup, ... Ngoài ra nó còn sử dụng thêm Javascript để xử lý các hiệu ứng cấp cao, javascript của Bootstrap sử dụng jquery nên để sử dụng được bắt buộc bạn phải bổ sung thêm thư viện jQuery nữa.

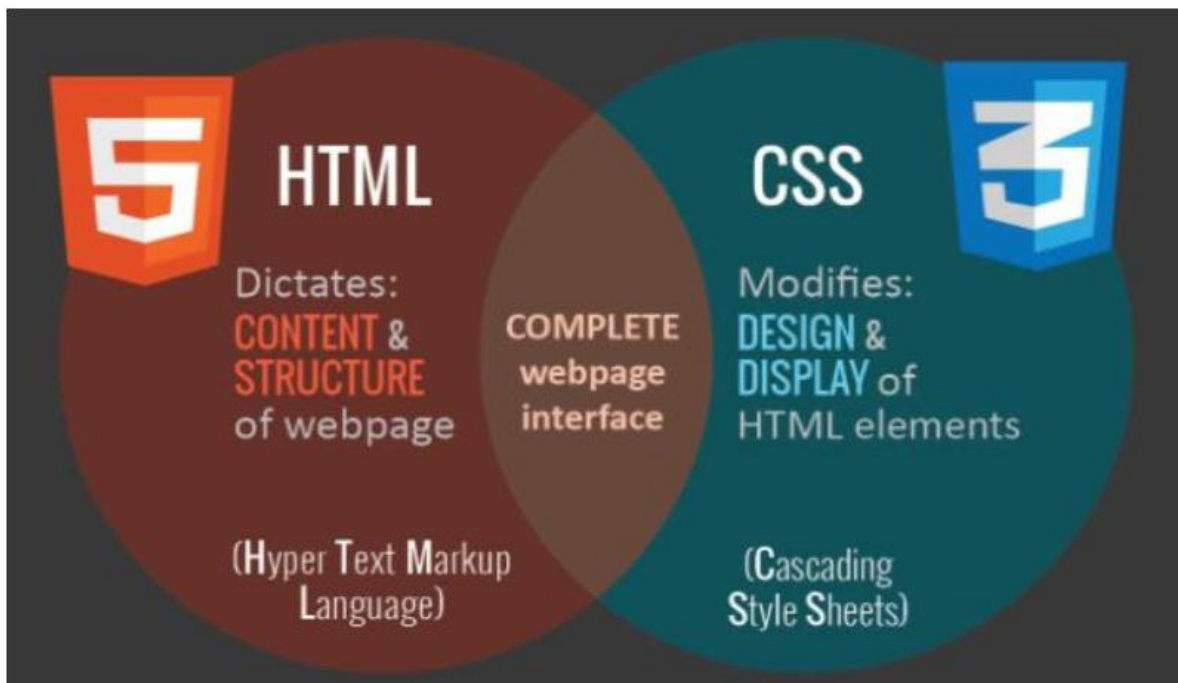
Các lợi ích khi sử dụng bootstrap:

- Phát triển giao diện nhanh chóng.
- Dễ học, dễ sử dụng
- Hỗ trợ javascript ,jquery

1.3.4. HTML và CSS

HTML (Hypertext Markup Language) là mã văn bản được dùng để xây dựng nên cấu trúc và nội dung của một trang web. Ví dụ, nội dung có thể được cấu thành bởi một loạt các đoạn văn, một danh sách liệt kê, hoặc sử dụng những hình ảnh và bảng biểu,...

CSS là viết tắt của Cascading Style Sheets – được dùng để miêu tả cách trình bày các tài liệu viết bằng ngôn ngữ HTML hay XHTML. CSS được hiểu một cách đơn giản là cách mà chúng ta thêm các kiểu hiển thị (font chữ, kích thước, màu sắc,...) cho một tài liệu Web.



Hình 1.2: HTML và CSS

HTML được tạo ra để giúp các lập trình viên mô tả nội dung trên một trang web. HTML sử dụng các thẻ để giúp bạn thêm đoạn văn, tiêu đề, hình ảnh, dấu đầu dòng và các phần cấu trúc khác. Giống như việc viết một cái gì đó trên một tài liệu word, HTML giúp bạn viết một cái gì đó trên một trang web.

CSS là công cụ giúp cho một trang web trông tuyệt vời hơn. Nó liên quan nhiều đến việc thay đổi phong cách của trang web, chịu trách nhiệm về giao diện và những nội dung khác trên đó. Giống như thay đổi kích thước phông chữ, màu chữ và định dạng trên một tài liệu văn bản Word.

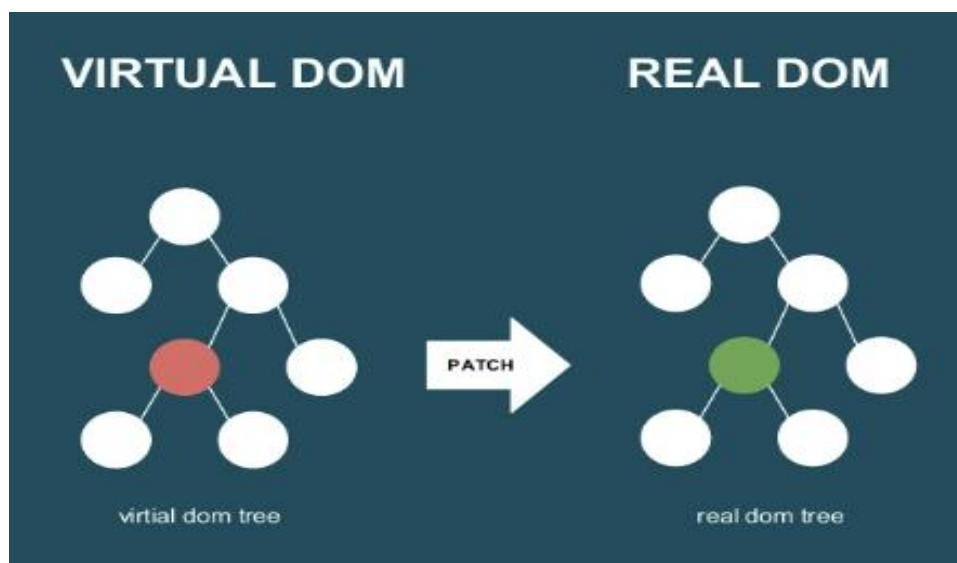
1.3.5. ReactJS

ReactJS là một thư viện JavaScript mã nguồn mở được phát triển bởi Facebook. Nó là một phần quan trọng của cơ sở công nghệ để xây dựng giao diện người dùng (UI) tương tác. ReactJS tập trung vào việc xây dựng các thành phần (components) UI có khả năng tái sử dụng, giúp tạo ra các giao diện phức tạp bằng cách tổ hợp các thành phần nhỏ



Hình 1.3: ReactJS

ReactJS đi kèm với nhiều tính năng mạnh mẽ như: Virtual DOM: React sử dụng Virtual DOM để tối ưu hóa hiệu suất. Virtual DOM là một bản sao của DOM thực tế trong bộ nhớ, và React sẽ cập nhật trạng thái của ứng dụng trên Virtual DOM trước khi áp dụng thay đổi lên DOM thực tế, giúp tăng tốc quá trình cập nhật UI.



Hình 1.4: Virtual DOM

Thành phần (Components): React chia giao diện thành các thành phần độc lập, có thể tái sử dụng và kết hợp lại với nhau để tạo thành các giao diện phức tạp hơn.

JSX: JSX là một cú pháp mở rộng của JavaScript cho phép bạn viết giao diện người dùng bằng cách kết hợp HTML và JavaScript trong cùng một tệp.

Unidirectional Data Flow: React thúc đẩy luồng dữ liệu một chiều, giúp quản lý trạng thái của ứng dụng trở nên dễ dàng và dự đoán hơn.

Hỗ trợ cộng đồng mạnh mẽ: React có một cộng đồng lớn và nhiều tài liệu, thư viện bổ sung, và công cụ hỗ trợ.

*** Ưu điểm:**

+ **Hiệu suất cao:** Nhờ Virtual DOM và cách quản lý trạng thái, React có hiệu suất tốt.

+ **Tái sử dụng thành phần:** Việc có thể tái sử dụng các thành phần giúp tiết kiệm thời gian và công sức trong việc phát triển.

+ **Cộng đồng lớn:** Cộng đồng đông đảo đồng nghĩa với nhiều tài liệu học và hỗ trợ.

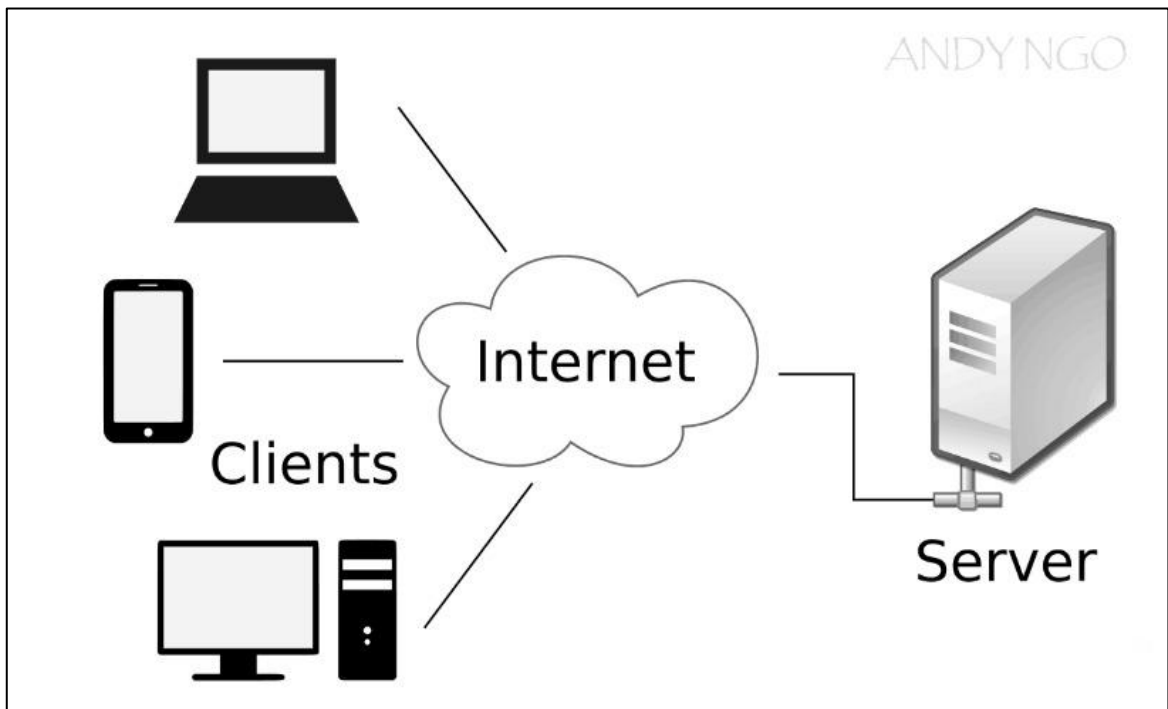
+ **Phát triển ứng dụng di động:** React Native, dựa trên React, cho phép phát triển ứng dụng di động đa nền tảng.

*** Nhược điểm:**

+ **Học khái niệm mới:** Việc làm quen với cách quản lý trạng thái và JSX có thể tốn thời gian cho người mới.

+ **Khả năng quản lý ứng dụng lớn:** Đối với các ứng dụng lớn và phức tạp, việc quản lý trạng thái có thể trở nên khó khăn.

1.4. Mô hình Client – Server



Hình 1.5: Mô Hình Client - Server

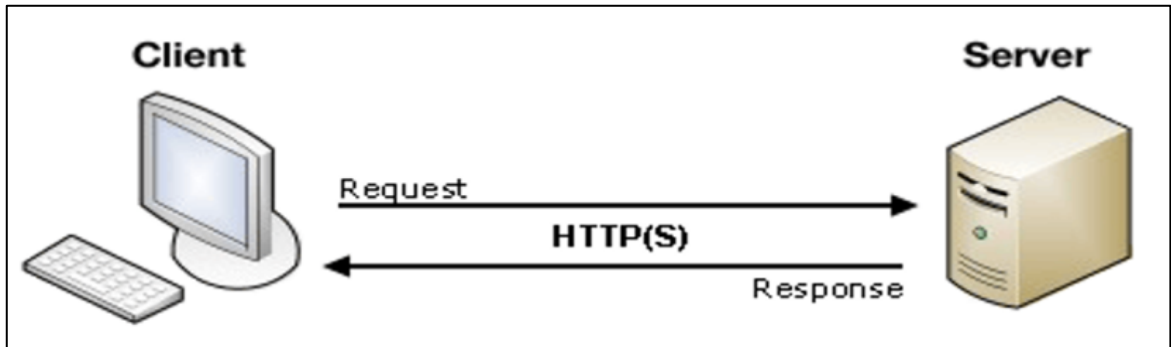
Mô hình client - server là mô hình giúp các máy tính giao tiếp truyền tải dữ liệu cho nhau. Client và server về bản chất thì nó là 2 máy tính giao tiếp và truyền tải dữ liệu cho nhau.

Máy tính đóng vai trò là máy khách – **Client**: Với vai trò là máy khách, chúng sẽ không cung cấp tài nguyên đến các máy tính khác mà chỉ sử dụng tài nguyên được cung cấp từ máy chủ. Một client trong mô hình này có thể là một server cho mô hình khác, tùy thuộc vào nhu cầu sử dụng của người dùng.

Máy tính đóng vai trò là máy chủ – **Server**: Là máy tính có khả năng cung cấp tài nguyên và các dịch vụ đến các máy khách khác trong hệ thống mạng. Server đóng vai trò hỗ trợ cho các hoạt động trên máy khách client diễn ra hiệu quả hơn.

Mô hình Client Server là mô hình mạng máy tính trong đó các máy tính con được đóng vai trò như một máy khách, chúng làm nhiệm vụ gửi yêu cầu đến máy chủ. Để máy chủ xử lý yêu cầu và trả kết quả về cho máy khách đó.

Trong mô hình Client Server, server chấp nhận tất cả các yêu cầu hợp lệ từ mọi nơi khác nhau trên Internet, sau đó trả kết quả về máy tính đã gửi yêu cầu đó. Máy tính được coi là máy khách khi chúng làm nhiệm vụ gửi yêu cầu đến các máy chủ và đợi câu trả lời được gửi về.



Hình 1.6: Nguyên tắc hoạt động của mô hình Client Server

Để máy khách và máy chủ có thể giao tiếp được với nhau thì giữa chúng phải có một chuẩn nhất định, và chuẩn đó được gọi là giao thức. Một số giao thức được sử dụng phổ biến hiện nay như: HTTPS, TCP/IP, FTP,...

Nếu máy khách muốn lấy được thông tin từ máy chủ, chúng phải tuân theo một giao thức mà máy chủ đó đưa ra. Nếu yêu cầu đó được chấp nhận thì máy chủ sẽ thu thập thông tin và trả về kết quả cho máy khách yêu cầu. Bởi vì Server - máy chủ luôn luôn trong trạng thái sẵn sàng để nhận request từ client nên chỉ cần client gửi yêu cầu tín hiệu và chấp nhận yêu cầu đó thì server sẽ trả kết quả về phía client trong thời gian ngắn nhất.

*** Ưu điểm của mô hình**

+ Giúp chúng ta có thể làm việc trên bất kì một máy tính nào có hỗ trợ giao thức truyền thông. Giao thức chuẩn này cũng giúp các nhà sản xuất tích hợp lên nhiều sản phẩm khác nhau mà không gặp phải khó khăn gì.

+ Có thể có nhiều server cùng làm một dịch vụ, chúng có thể nằm trên nhiều máy tính hoặc một máy tính.

+ Chỉ mang đặc điểm của phần mềm mà không hề liên quan đến phần cứng, ngoài yêu cầu duy nhất là server phải có cấu hình cao hơn các client.

+ Hỗ trợ người dùng nhiều dịch vụ đa dạng và sự tiện dụng bởi khả năng truy cập từ xa.

+ Cung cấp một nền tảng lý tưởng, cho phép cung cấp tích hợp các kỹ thuật hiện đại như mô hình thiết kế hướng đối tượng, hệ chuyên gia, hệ thông tin địa lý (GIS).

*** Nhược điểm của mô hình**

+ Vấn đề bảo mật dữ liệu thông tin đôi khi còn chưa được an toàn lắm. Vì do phải trao đổi dữ liệu giữa 2 máy tính khác nhau ở 2 khu vực địa lý cách xa nhau. Và đây cũng nhược điểm duy nhất của mô hình này.

+ Tuy nhiên vấn đề này thì có một số giao thức đã hỗ trợ bảo mật dữ liệu khi truyền tải. Giao thức được sử dụng phổ biến như HTTPS.

1.5. Hệ quản trị cơ sở dữ liệu

1.5.1. Các khái niệm

Cơ sở dữ liệu (CSDL – Database) là một tập hợp dữ liệu có tổ chức, được lưu trữ và quản lý theo một cấu trúc nhất định, nhằm phục vụ cho việc truy xuất, cập nhật và quản lý thông tin một cách hiệu quả. Cơ sở dữ liệu thường được sử dụng trong các hệ thống thông tin, từ quy mô nhỏ (như trang web cá nhân) đến các hệ thống lớn (như ngân hàng, thương mại điện tử, chính phủ,...).

Dữ liệu trong cơ sở dữ liệu thường phản ánh các thực thể trong thế giới thực và mối quan hệ giữa chúng. Ví dụ, trong một hệ thống bán hàng, cơ sở dữ liệu có thể lưu trữ thông tin về khách hàng, sản phẩm, đơn hàng, và các giao dịch tương ứng.

Hệ quản trị cơ sở dữ liệu (Database Management System – DBMS) là phần mềm trung gian cho phép người dùng tương tác với cơ sở dữ liệu. DBMS thực hiện các nhiệm vụ như:

- Lưu trữ, truy xuất, cập nhật và xóa dữ liệu.
- Quản lý truy cập đồng thời từ nhiều người dùng.
- Đảm bảo tính toàn vẹn và nhất quán của dữ liệu.
- Cung cấp các công cụ bảo mật, sao lưu và phục hồi dữ liệu.

DBMS giúp người dùng không cần phải quan tâm đến cách dữ liệu được tổ chức và lưu trữ ở mức thấp, mà chỉ cần tập trung vào cách khai thác và sử dụng dữ liệu ở mức logic.

Có nhiều loại DBMS khác nhau, được phân loại dựa trên mô hình dữ liệu mà chúng sử dụng. Hai loại phổ biến nhất là:

- Hệ quản trị cơ sở dữ liệu quan hệ (Relational DBMS – RDBMS): Sử dụng mô hình bảng (table) để lưu trữ dữ liệu. Dữ liệu được tổ chức dưới dạng hàng (record) và cột (field), có khóa chính và khóa ngoại để liên kết giữa các bảng. Ví dụ: MySQL, PostgreSQL, Oracle, SQL Server.
- Hệ quản trị cơ sở dữ liệu phi quan hệ (NoSQL DBMS): Không sử dụng mô hình bảng, thay vào đó dữ liệu có thể được lưu dưới dạng document (tài liệu), key-value, column-family, hoặc graph. NoSQL thích hợp với các hệ thống yêu cầu linh hoạt về cấu trúc dữ liệu, quy mô lớn, hiệu năng cao. Ví dụ: MongoDB, Redis, Cassandra, Neo4j.

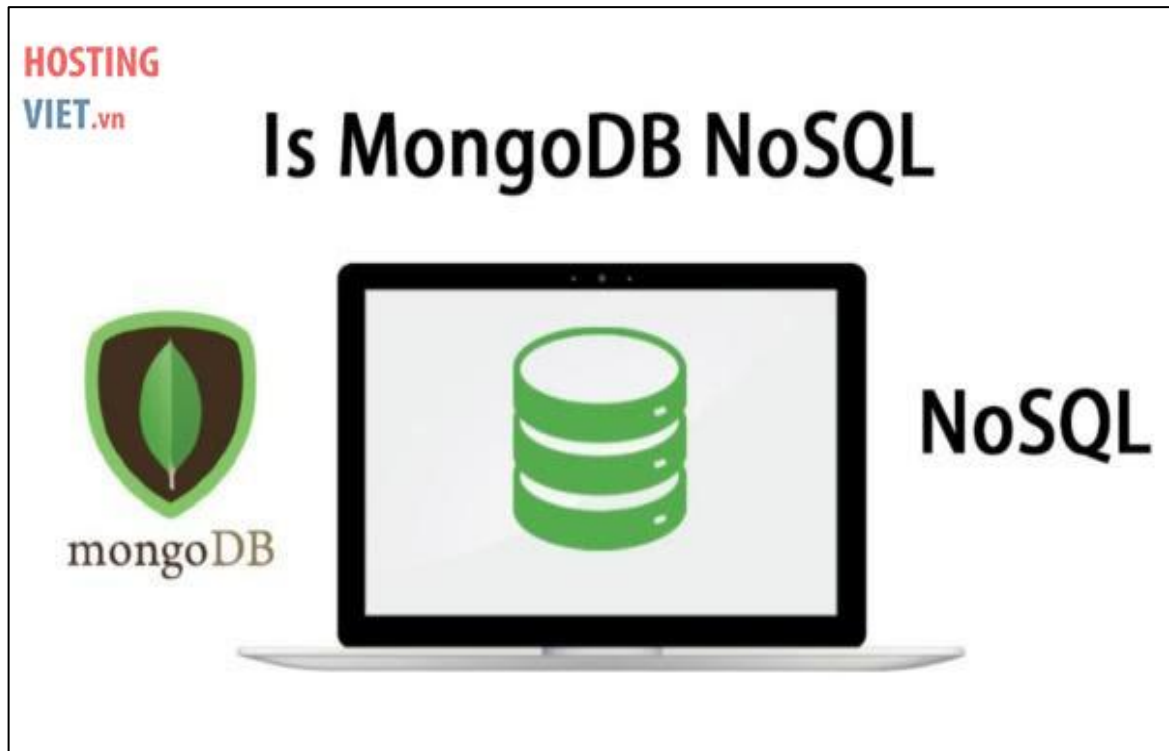
Trong phát triển phần mềm, DBMS đóng vai trò như trái tim của hệ thống, nơi lưu trữ và quản lý toàn bộ dữ liệu nghiệp vụ. Một hệ thống phần mềm hiệu quả không chỉ cần logic xử lý tốt, mà còn cần có cấu trúc dữ liệu khoa học và khả năng truy xuất tối ưu.

1.5.2. MongoDB

MongoDB là một dạng phần mềm cơ sở dữ liệu sử dụng mã nguồn mở NoSQL. Nó có thể hỗ trợ trên nhiều nền tảng khác nhau và được thiết kế với mục đích hướng đến đối tượng. MongoDB hoạt động dựa vào các khái niệm Collection và Document. Đồng thời, nó có hiệu suất cao cùng với tính khả dụng tốt và dễ dàng mở rộng.

NoSQL còn được xem là mảnh vá cho những hạn chế, khiếm khuyết của mô hình Hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System - RDBMS). Đó là cải tiến về tốc độ, tính năng cũng như khả năng mở rộng.

NoSQL cho phép mở rộng dữ liệu mà không cần tạo khóa ngoại, khóa chính hay kiểm tra tính ràng buộc... Có thể thấy, dạng cơ sở dữ liệu này có thể linh động, sẵn sàng bỏ qua tính toàn vẹn cứng nhắc của transaction hay dữ liệu nhằm đổi lấy hiệu suất, cùng với khả năng mở rộng tốt. Chính vì ưu điểm này mà nhiều Tập đoàn lớn rất ưa chuộng NoSQL như Facebook, Google,...



Hình 1.7: MongoDB NoSQL

Nguyên tắc hoạt động của MongoDB là dưới một tiến trình dịch vụ ngầm và mở một cổng (mặc định là cổng 27017), để có thể tiếp nhận các yêu cầu truy vấn, thao tác; sau đó tiến hành xử lý.

Mỗi bản ghi của MongoDB (document) được gán một trường có tên “_id” nhằm xác định tính duy nhất của bản ghi. Có thể hiểu id này như tên gọi của một bản ghi và dùng phân biệt chúng với các bản ghi khác. Đồng thời, nó còn được sử dụng cho mục đích truy vấn hoặc tìm kiếm thông tin. Trường dữ liệu “_id” được tự động đánh chỉ mục (index) để đảm bảo tốc độ truy vấn đạt hiệu suất tối ưu.

Mỗi truy vấn dữ liệu đều được ghi đệm lên bộ nhớ RAM nên các truy vấn sau đó sẽ diễn ra nhanh hơn. Bởi nó không cần đọc dữ liệu từ ổ cứng.

Khi thực hiện thêm, xóa hay sửa bản ghi thì MongoDB đều mất 60s để ghi các dữ liệu được thay đổi từ RAM xuống ổ cứng. Điều này nhằm mục đích đảm bảo hiệu suất mặc định của chương trình.

*** Ưu điểm của MongoDB**

+ Linh hoạt trong lưu trữ các kích cỡ dữ liệu khác nhau. Nhờ chúng được lưu dưới dạng JSON nên bạn thoải mái chèn bất kỳ thông tin nào tùy theo nhu cầu sử dụng.

+ Tiết kiệm thời gian trong việc kiểm tra sự tương thích về cấu trúc khi thêm, xóa hoặc cập nhật dữ liệu. Nhờ MongoDB không có sự ràng buộc trong một khuôn khổ, quy tắc nhất định nào.

+ Bạn dễ dàng mở rộng hệ thống thông qua việc thêm node vào cluster. Cụm các node này đóng vai trò như thư viện chứa các dữ liệu giao tiếp với nhau.

+ Tốc độ truy vấn của MongoDB nhanh hơn so với RDBMS do toàn bộ dữ liệu truy vấn đã được ghi đệm lên bộ nhớ RAM. Nhờ thế, những lượt truy vấn sau sẽ được rút ngắn thời gian vì chúng không cần đọc từ ổ cứng.

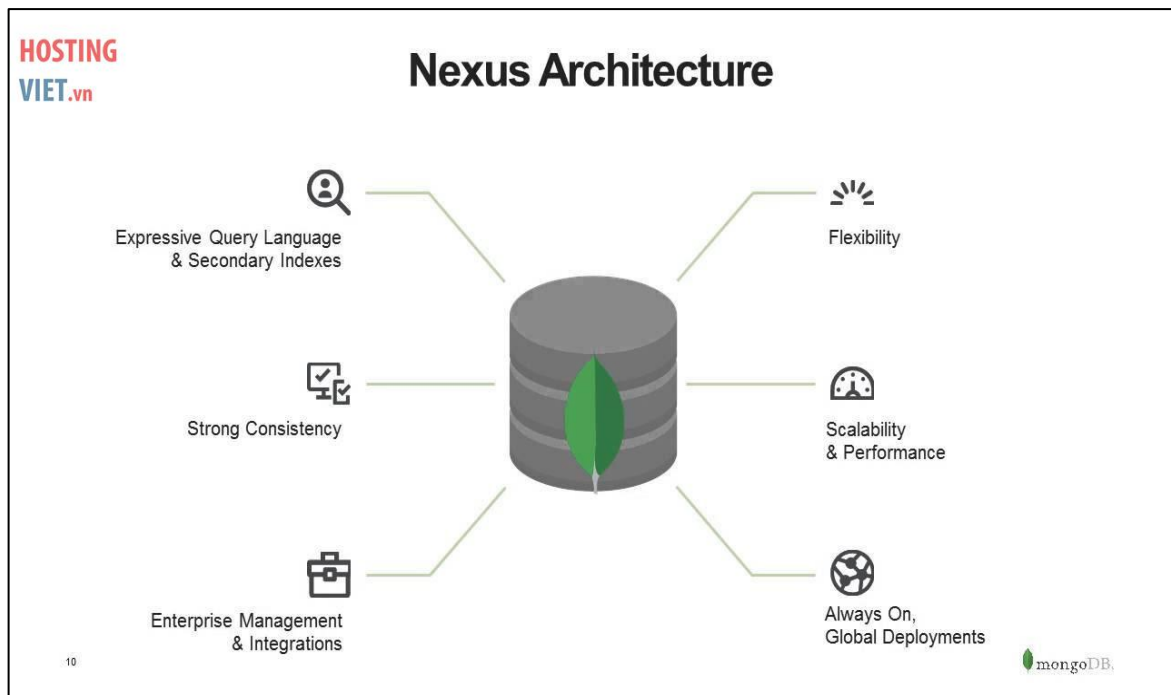
+ Trường dữ liệu “_id” (đại diện cho giá trị duy nhất trong mỗi document) được tự động đánh chỉ mục nên hiệu suất luôn đạt mức cao nhất.

*** Nhược điểm của MongoDB**

+ Vì dữ liệu không bị ràng buộc nên trong quá trình sử dụng, bạn cần cẩn thận trong mọi thao tác nhằm tránh xảy ra những điều không mong muốn, làm ảnh hưởng đến dữ liệu.

+ Chương trình MongoDB tiêu tốn khá nhiều dung lượng bộ nhớ do dữ liệu được lưu dưới dạng key và value. Bên cạnh đó, một số collection chỉ có sự khác biệt về value nên việc lặp lại key là điều khó tránh khỏi. Điều này dẫn đến thừa dữ liệu.

+ Thông thường, thời gian để dữ liệu chuyển đổi từ RAM xuống ổ cứng khoảng 60s nên nguy cơ bị mất dữ liệu nếu xảy ra mất điện là điều có thể xảy ra.



Hình 1.8: Nexus Architecture

Nên dùng MongoDB trong trường hợp nào?

Tuy MongoDB khá hữu dụng nhưng không phải bạn có thể dùng nó trong mọi tình huống. Theo kinh nghiệm của Hosting Việt, bạn chỉ dùng MongoDB khi gặp các trường hợp sau:

- Khi website cần có tính năng thêm dữ liệu thường xuyên, bởi MongoDB có thể ghi với tốc độ cao và cực kỳ bảo mật. Điều này giúp trang web không bị downtime khi cập nhật dữ liệu.
- Khi có hệ thống dữ liệu với dung lượng lớn và cần truy vấn nhanh.
- Khi muốn tìm kiếm thông tin trong bộ nhớ nhanh chóng vì MongoDB có tốc độ tìm kiếm thông tin liên quan cực kỳ nhanh.
- Mongo rất thích hợp để sử dụng trong trường hợp máy chủ chưa có hệ quản trị cơ sở dữ liệu.

Không nên dùng MongoDB trong trường hợp nào?

- Nếu website cần nhiều ứng dụng giao dịch (transaction), điển hình như các trang web của ngân hàng thì không nên dùng MongoDB. Do chương trình này không hỗ trợ cơ chế transaction.
- Những ứng dụng cần đến SQL.

CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

2.1. Khảo sát hệ thống

Hiện trạng kinh doanh sách trực tuyến tại Việt Nam đang phát triển mạnh mẽ, thu hút đông đảo người dùng đăng ký mua sắm bởi nhu cầu tiếp cận sách và tài liệu đọc ngày càng cao. Nhiều doanh nghiệp đã đầu tư xây dựng các nền tảng thương mại điện tử chuyên về sách phù hợp với xu hướng công nghệ mới nhất như trí tuệ nhân tạo, học máy, và phát triển ứng dụng di động.

Việc mua sắm sách trực tuyến tại các website bán sách ở Việt Nam là một phần quan trọng của thị trường thương mại điện tử, nhằm giúp người dùng tiếp cận đa dạng đầu sách một cách nhanh chóng, cũng như tìm hiểu thông tin chi tiết về nội dung. Tuy nhiên việc lọc, tìm kiếm và đánh giá sách phù hợp của người dùng còn tồn tại một số bất cập. Ở nhiều website bán sách, hệ thống tìm kiếm, phân loại và gợi ý sách còn lỗi thời, không đầy đủ tính năng hoặc thiếu tính thẩm mỹ, thân thiện với trải nghiệm người dùng.

Qua khảo sát một số hệ thống bán sách trực tuyến hiện có, nhiều trang web còn tồn tại các vấn đề như giao diện cũ kỹ, không được tối ưu cho thiết bị di động. Quá trình tìm kiếm sách thường gặp lỗi và không hiển thị kết quả chính xác theo yêu cầu của người dùng. Hệ thống phân loại sách thiếu logic, gây khó khăn cho người dùng khi muốn tìm kiếm theo thể loại cụ thể. Trong khi đăng ký tài khoản và đặt hàng, nhiều hệ thống thường xuyên bị treo hoặc báo lỗi không rõ nguyên nhân. Chỉ khi người dùng thực hiện nhiều lần thao tác, đơn hàng mới được xử lý. Điều này gây ra lãng phí thời gian, công sức và dễ dẫn tới phiền hà khi mua hàng.

Đồng thời, nhiều hệ thống không có tính năng cho phép người dùng tự chủ động đánh giá và bình luận về sách. Thông tin mô tả sách thường sơ sài, thiếu chi tiết về nội dung, tác giả và nhà xuất bản. Quá trình thanh toán phức tạp với nhiều bước không cần thiết và thiếu đa dạng phương thức thanh toán

2.2. Mô tả bài toán

Xây dựng một website bán sách trực tuyến, đem lại sự thuận tiện nhất cho người mua, nhà xuất bản và các đối tác vận chuyển. Thay vì để người dùng lúng túng tìm kiếm sách, hệ thống sẽ có nhiệm vụ tạo ra các danh mục sách được phân loại rõ ràng và chức năng tìm kiếm.

Sau khi tài khoản được tạo, người dùng khi đăng nhập hệ thống sẽ có danh sách các sách mới nhất. Lúc này, người dùng có thể chọn một cuốn sách mình quan tâm, hệ thống sẽ xác nhận và đưa vào giỏ hàng. Người dùng khi đăng nhập hệ thống sẽ có thể thấy danh sách các sách mới nhất, sách trong giỏ hàng. Lúc này, người dùng cũng có thể chọn thêm sách khác phù hợp và tiến hành thanh toán, hệ thống sẽ xác nhận và gắn đơn hàng với người dùng đó.

Hệ thống này đem lại sự thuận tiện, dễ dàng cho công tác mua sắm, đáp ứng các yêu cầu nghiệp vụ một cách tin cậy, chính xác, an toàn. Đối với người dùng và nhà xuất bản, có thể theo dõi, quản lý đơn hàng một cách khoa học, dễ dàng hơn.

2.3. Xác định các yêu cầu của hệ thống

1. Quản lý danh mục sách và thông tin liên quan:
2. Tài khoản người dùng và quản lý:
 - Đăng ký tài khoản với xác thực email/số điện thoại.
 - Đăng nhập/đăng xuất.
 - Quản lý thông tin cá nhân: họ tên, địa chỉ, số điện thoại, email.
 - Quản lý địa chỉ giao hàng với nhiều địa chỉ khác nhau.
 - Quản lý phương thức thanh toán ưa thích.
 - Lịch sử mua hàng và theo dõi đơn hàng.
 - Danh sách yêu thích và sách đã xem.
3. Tìm kiếm và gợi ý:
 - Tìm kiếm cơ bản theo từ khóa.
 - Tìm kiếm nâng cao với nhiều tiêu chí (tác giả, nhà xuất bản, giá, thể loại).

- Bộ lọc kết quả tìm kiếm linh hoạt.
 - Hiện thị sách phổ biến, mới phát hành, giảm giá.
4. Giỏ hàng và thanh toán:
 5. Đánh giá và bình luận:
 6. Quản lý đơn hàng:
 - Theo dõi trạng thái đơn hàng (đã đặt, đang xử lý, đang giao, đã giao).
 - Cập nhật thông tin vận chuyển với mã theo dõi.
 - Hủy đơn hàng hoặc yêu cầu đổi/trả.
 - Lịch sử đơn hàng đầy đủ.

Các yêu cầu trên đảm bảo hệ thống website bán sách không chỉ đáp ứng nhu cầu kinh doanh hiện tại mà còn có khả năng phát triển trong tương lai, đem lại trải nghiệm tốt nhất cho người dùng và hiệu quả cao cho doanh nghiệp.

2.4. Thiết kế hệ thống

2.4.1. Các tác nhân của hệ thống

Quản trị viên (Administrator): Quản lý toàn bộ danh mục sách, tác giả, nhà xuất bản và các tài nguyên liên quan trên website. Quản trị viên có quyền truy cập đến toàn bộ các chức năng của hệ thống, có toàn quyền xem, thêm, sửa, xóa thông tin sản phẩm và người dùng trong hệ thống. Quản lý các nhà cung cấp, chương trình khuyến mãi, mã giảm giá, và theo dõi hoạt động của toàn bộ website.

Kiểm kho (Inventory Manager): Quản lý toàn bộ kho hàng, theo dõi số lượng sách còn tồn trong kho, cập nhật nhập xuất kho, tồn kho. Có thể truy xuất và xem các báo cáo thống kê về doanh thu, sản phẩm bán chạy và tỷ lệ chuyển đổi.

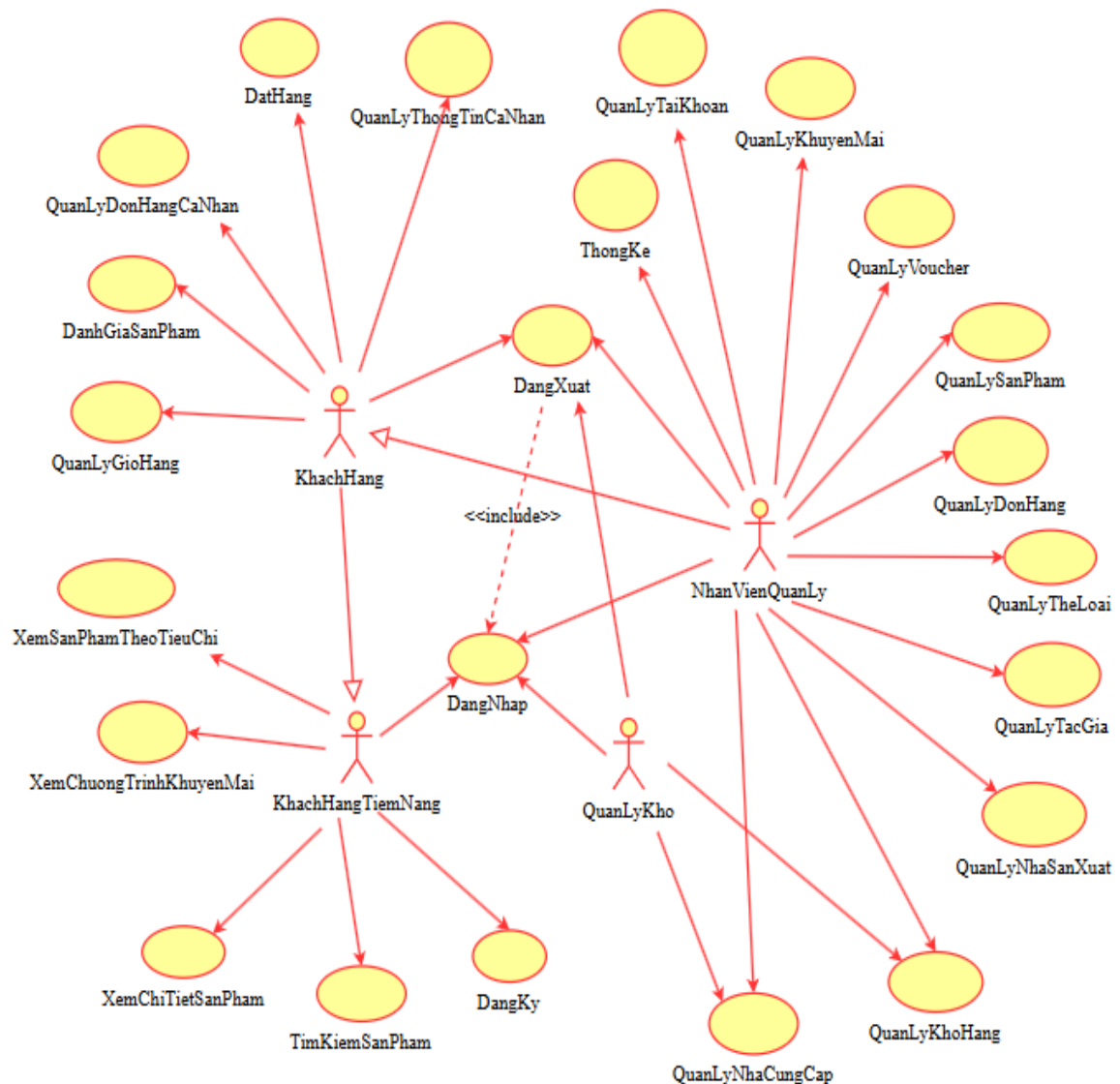
Khách hàng (User): Người dùng đã đăng ký tài khoản trên hệ thống. Có thể tìm kiếm, xem thông tin chi tiết sách, thêm sách vào giỏ hàng, thực hiện thanh toán và theo dõi đơn hàng của mình. Khách hàng đăng ký có thể đánh giá, bình luận về sách đã mua, quản lý thông tin cá nhân của mình trong hệ

thống. Họ cũng nhận được các ưu đãi, khuyến mãi dành riêng cho thành viên và tích lũy điểm thưởng khi mua hàng.

Khách hàng tiềm năng (Guest Customer): Người dùng chưa đăng ký tài khoản nhưng vẫn có thể truy cập website để tìm kiếm, xem thông tin sách.

2.4.2. Các use case trong hệ thống

* Sơ đồ use case tổng quan hệ thống



Hình 2.1: Sơ đồ use case tổng quan hệ thống

* Sơ đồ use case Khách hàng – Khách hàng Tiềm năng



Hình 2.2: Sơ đồ use case Khách hàng – Khách hàng Tiềm năng

Phân Loại Khách Hàng và Quyền Hạn:

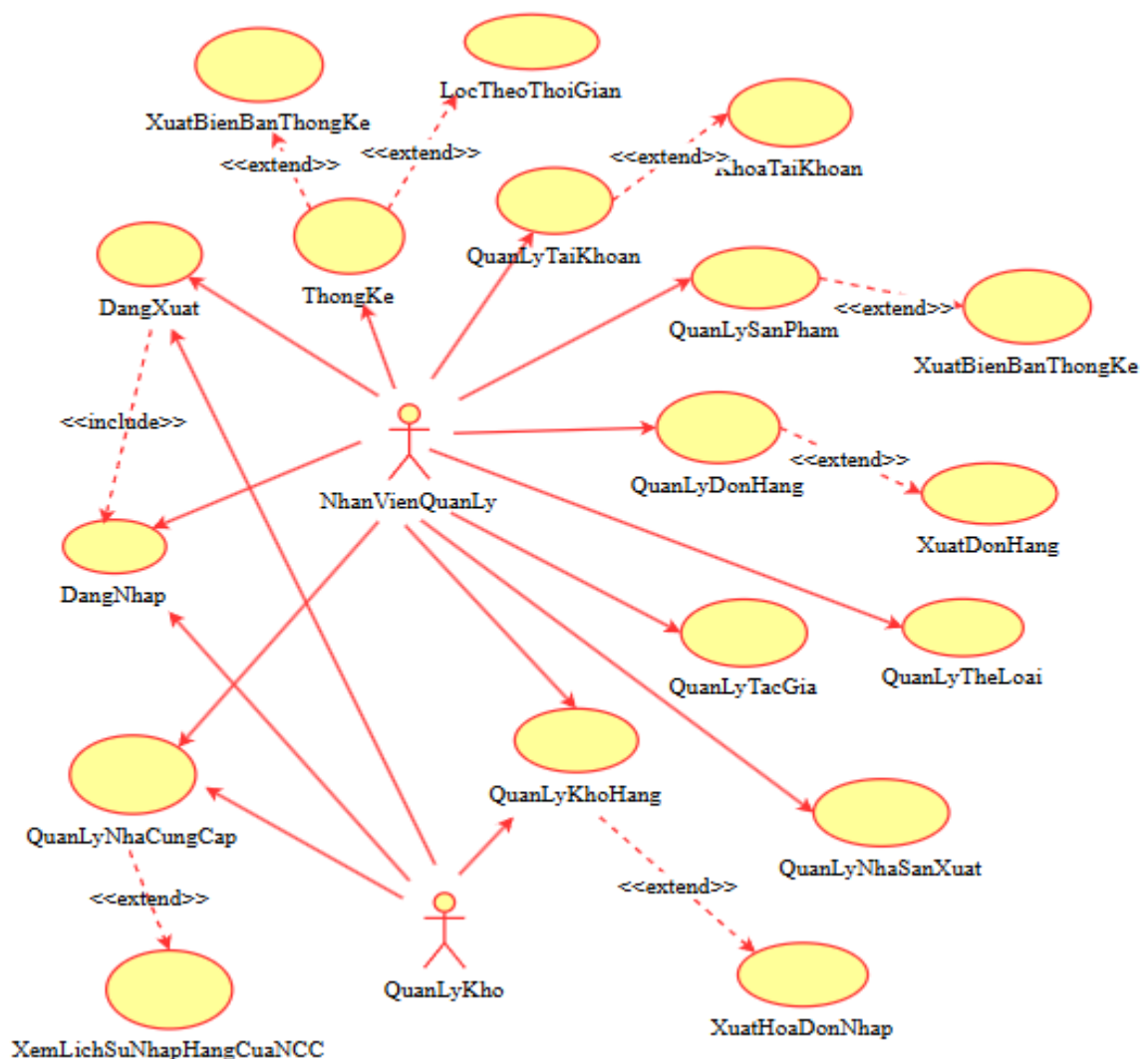
+ Khách Hàng Tiềm Năng (Khách Vãng Lai) là khách hàng không có tài khoản có thể:

- Đăng ký & Xác thực: Tạo tài khoản cá nhân và xác thực để trở thành khách hàng chính thức.
- Đăng nhập: Truy cập vào tài khoản đã đăng ký hoặc khôi phục mật khẩu khi cần.
- Tìm kiếm sản phẩm: Dễ dàng tìm kiếm theo tên sản phẩm hoặc tác giả.
- Duyệt sản phẩm: Xem chi tiết sản phẩm và lọc theo thể loại, sắp xếp theo tiêu chí tùy chọn.

+ Khách Hàng Đã Đăng Ký: Ngoài tất cả quyền hạn của khách vãng lai, khách hàng đã đăng ký còn được:

- Quản lý đơn hàng: Xem lịch sử và chi tiết các đơn hàng đã đặt.
- Cá nhân hóa thông tin: Cập nhật thông tin cá nhân, thay đổi mật khẩu và tùy chỉnh tài khoản.
- Thanh toán linh hoạt: Chọn phương thức thanh toán phù hợp khi đặt hàng
- Đánh giá sản phẩm: Chia sẻ trải nghiệm và đánh giá sản phẩm theo thang điểm 5 sao.
- Quản lý giỏ hàng: Thêm, xóa sản phẩm và điều chỉnh số lượng trong giỏ hàng.

* Sơ đồ use case Nhân viên quản lý:

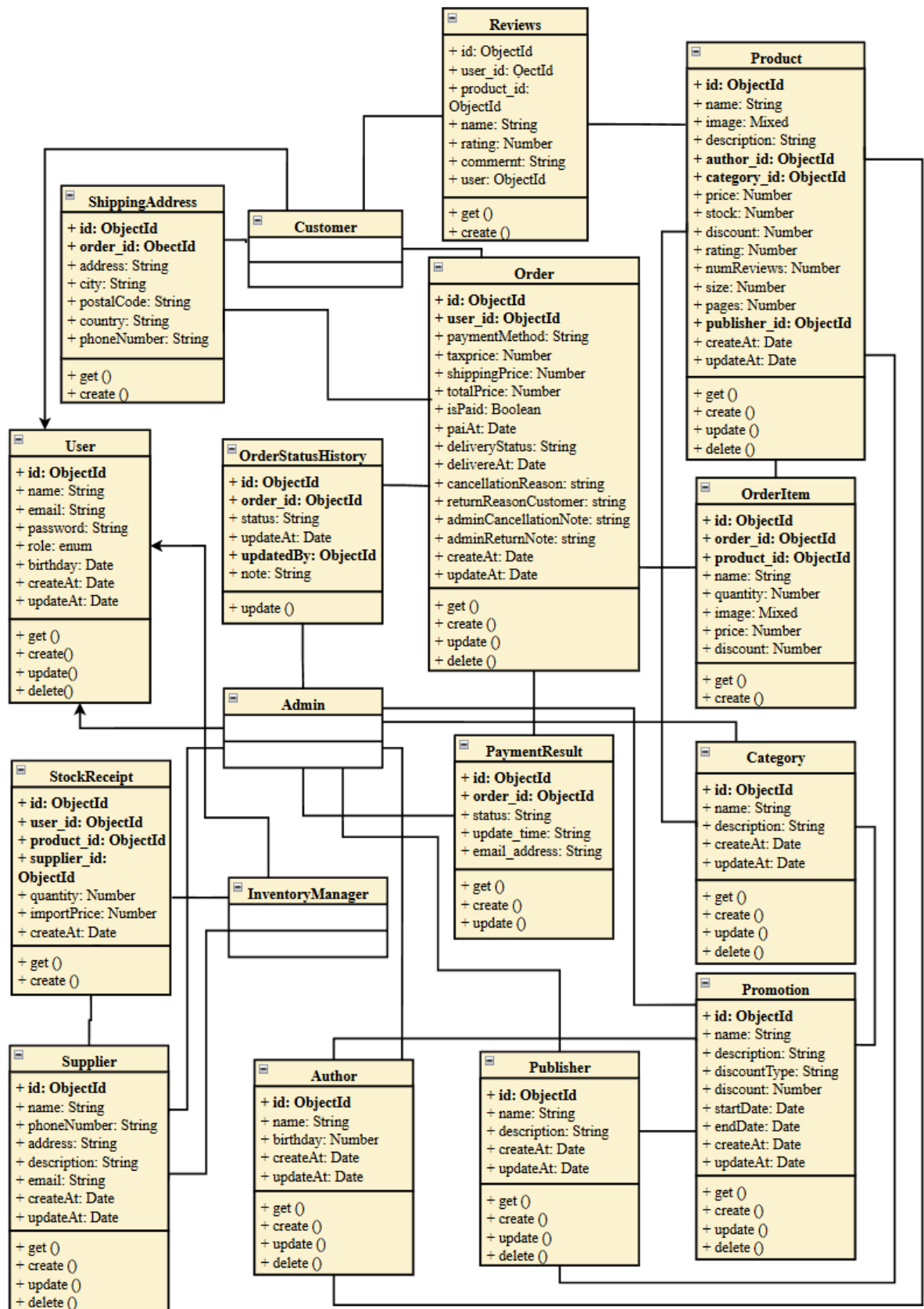


Hình 2.3: Sơ đồ use case Nhân viên quản lý

Quyền Hạn Nhân Viên Quản Lý:

- + Nhân viên quản lý được cấp quyền truy cập vào hệ thống quản trị với các chức năng toàn diện:
- + Đăng nhập quản trị: Truy cập vào trang quản lý thông qua tài khoản được cấp quyền quản lý.
- + Quản lý tài khoản khách hàng: Xem thông tin chi tiết khách hàng đã đăng ký, khóa/mở khóa tài khoản người dùng khi cần thiết.
- + Quản lý sản phẩm: Thêm sản phẩm mới vào hệ thống, cập nhật thông tin sản phẩm hiện có, xóa các sản phẩm không còn kinh doanh, tìm kiếm nhanh sản phẩm để thực hiện quản lý.
- + Quản lý đơn hàng: Xem danh sách tổng hợp đơn hàng, truy cập chi tiết từng đơn hàng, cập nhật trạng thái đơn hàng (đang xử lý, đã giao, đã hủy...).
- + Quản lý thể loại: Thêm thể loại sản phẩm mới, điều chỉnh thông tin thể loại, xóa thể loại không còn sử dụng.
- + Quản lý tác giả: Thêm tác giả mới, điều chỉnh thông tin tác giả, xóa tác giả không còn sử dụng.
- + Quản lý nhà sản xuất: Thêm nhà sản xuất sản phẩm mới, điều chỉnh thông tin nhà sản xuất, xóa nhà sản xuất không còn sử dụng.
- + Quản lý nhà cung cấp: Thêm nhà cung cấp sản phẩm mới, điều chỉnh thông tin nhà cung cấp, xóa nhà cung cấp.
- + Quản lý kho hàng: Thực hiện nhập hàng vào kho, truy xuất và in hóa đơn nhập hàng.
- + Thống kê và báo cáo: Theo dõi doanh thu theo thời gian, nắm bắt số lượng sản phẩm tồn kho, thống kê số lượng đơn hàng đã xử lý, giám sát đơn hàng mới phát sinh.

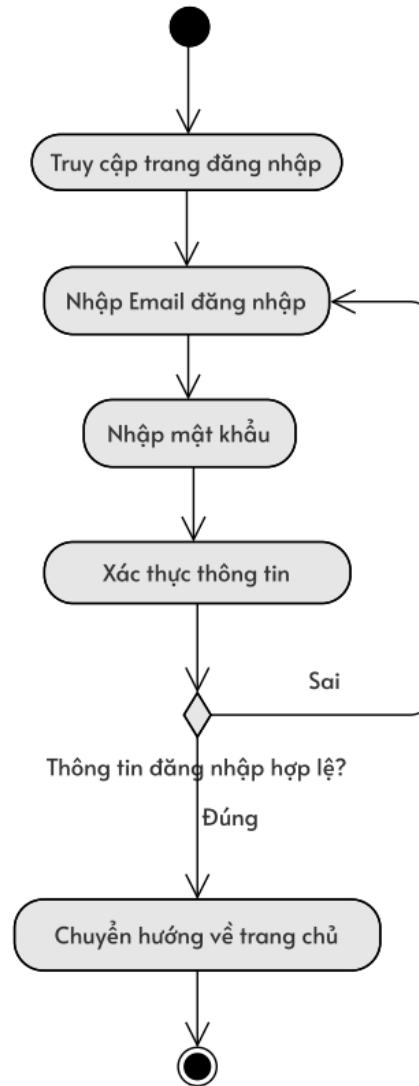
2.4.3. Biểu đồ lớp



Hình 2.4: Biểu đồ lớp hệ thống

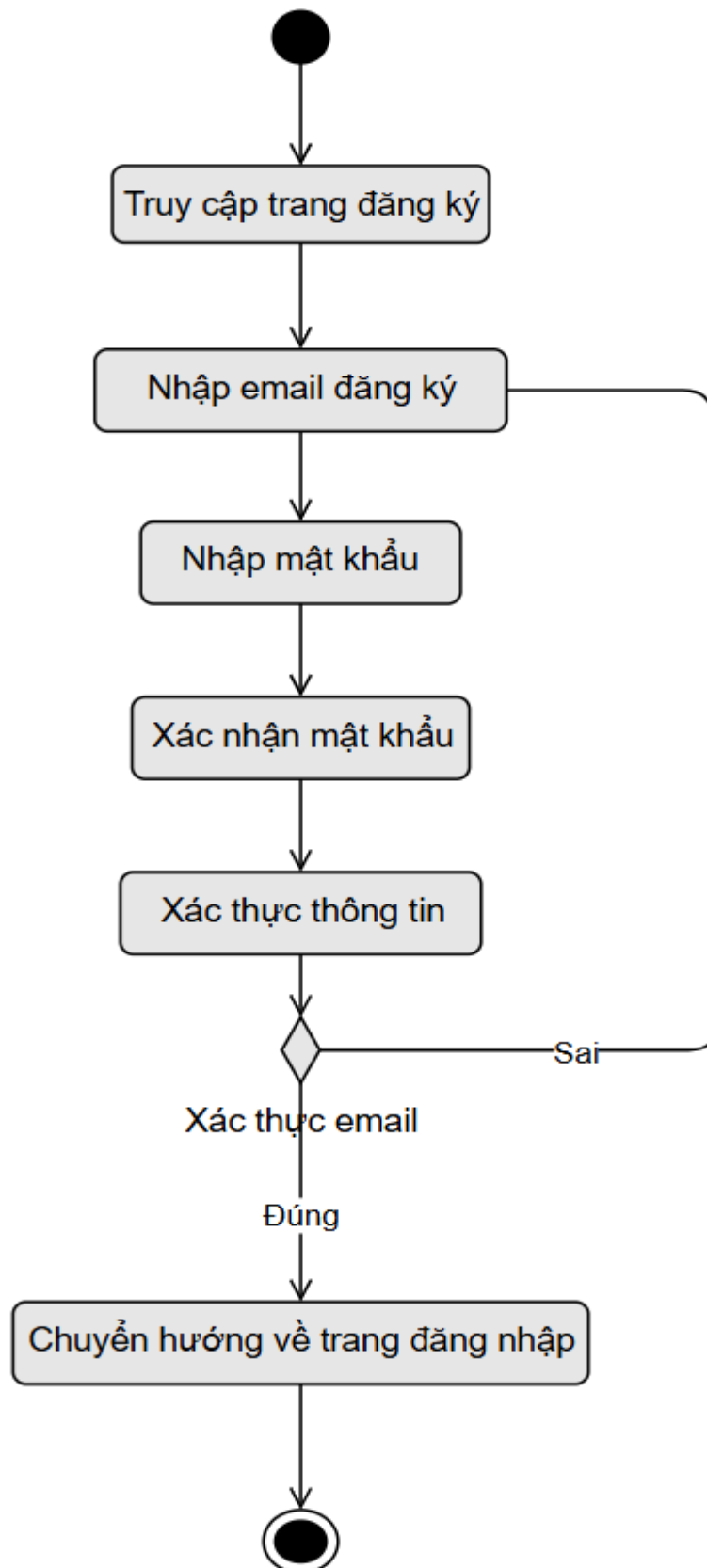
2.4.4. Biểu đồ hoạt động

. * Use case Đăng nhập



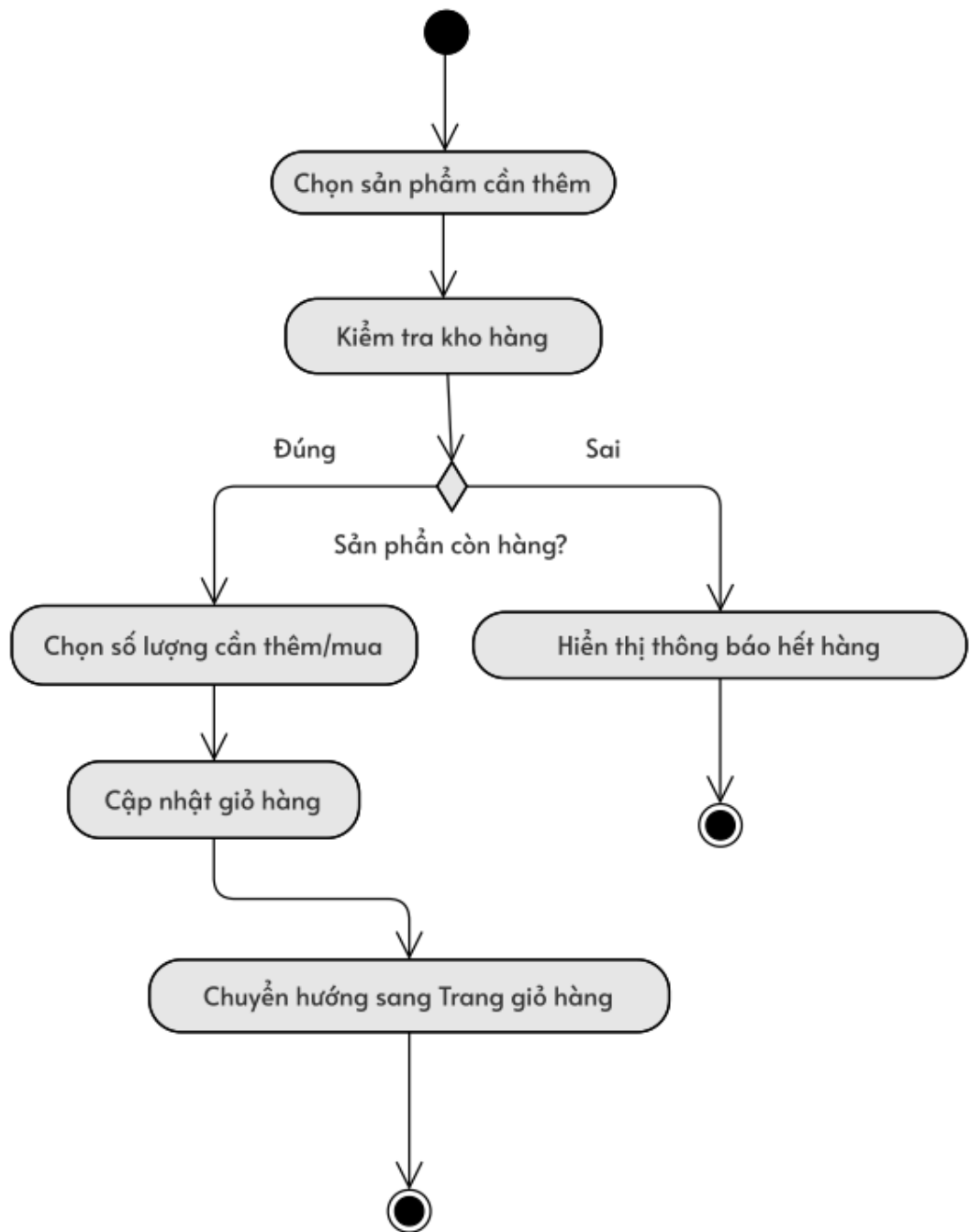
Hình 2.5: Biểu đồ hoạt động use case Đăng nhập

* Use case Đăng ký



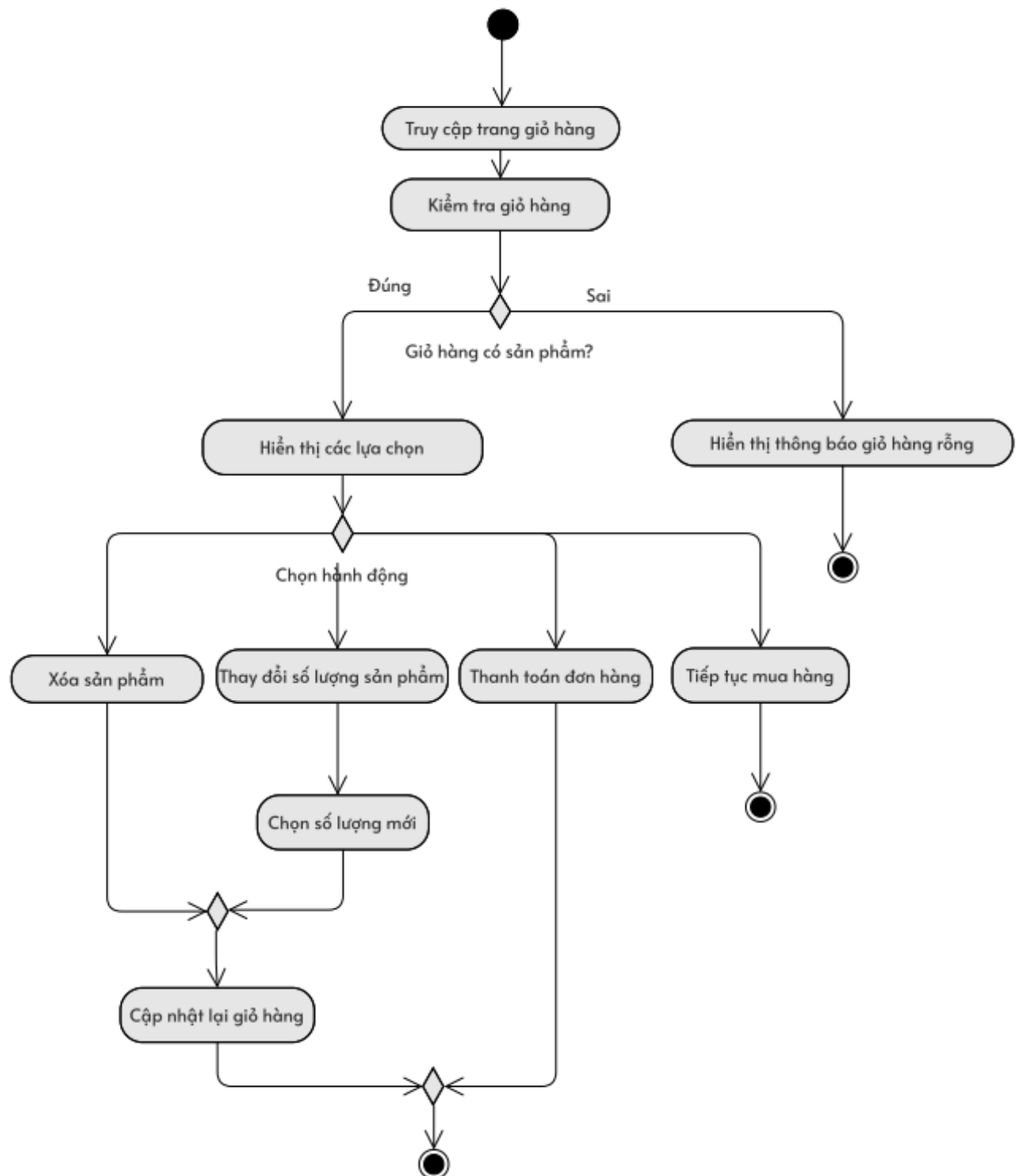
Hình 2.6: Biểu đồ hoạt động use case Đăng ký

* Use case Thêm sản phẩm vào giỏ hàng



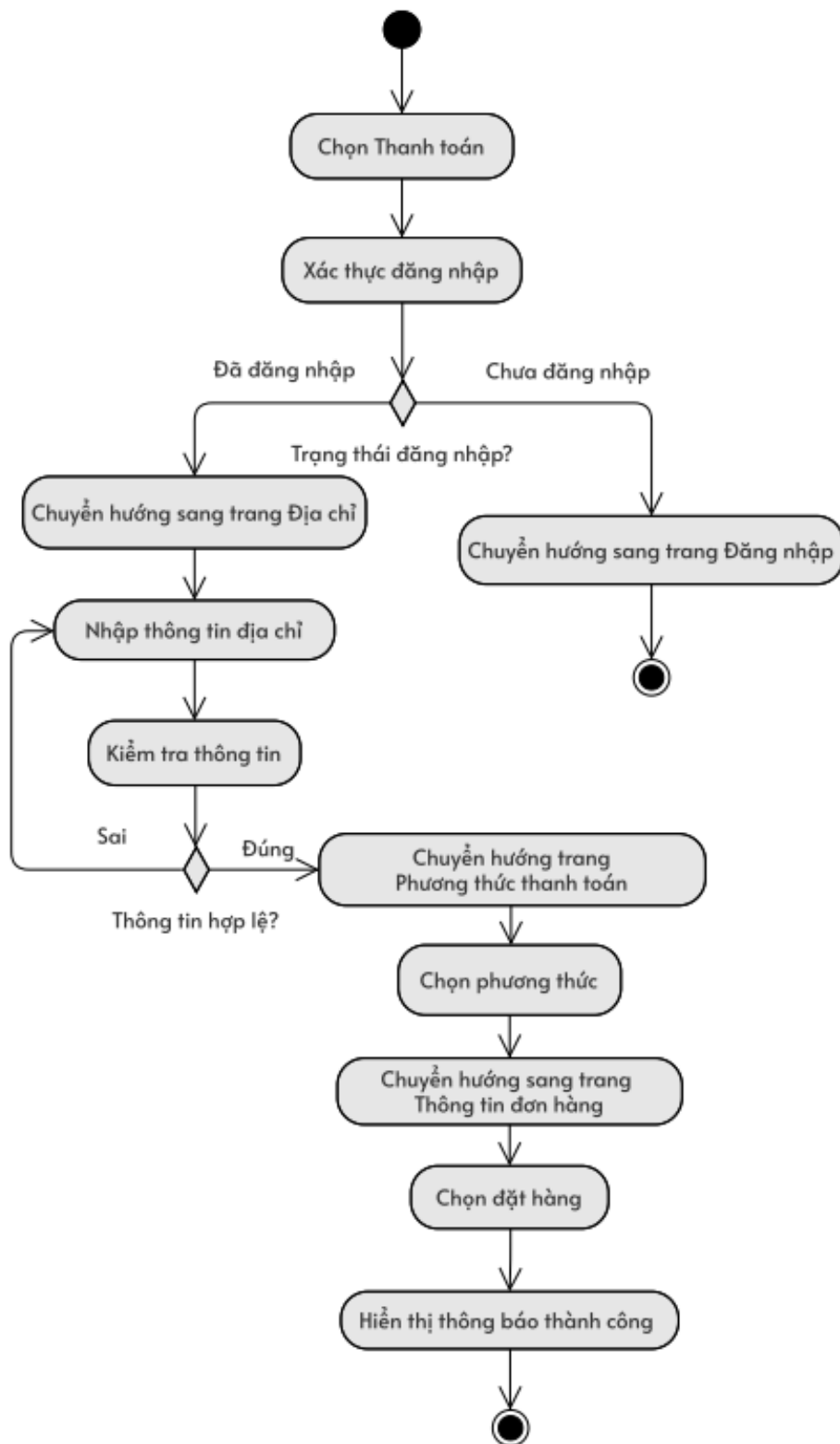
Hình 2.7: Biểu đồ hoạt động use case Thêm sản phẩm vào giỏ hàng

* Use case Quản lý giỏ hàng



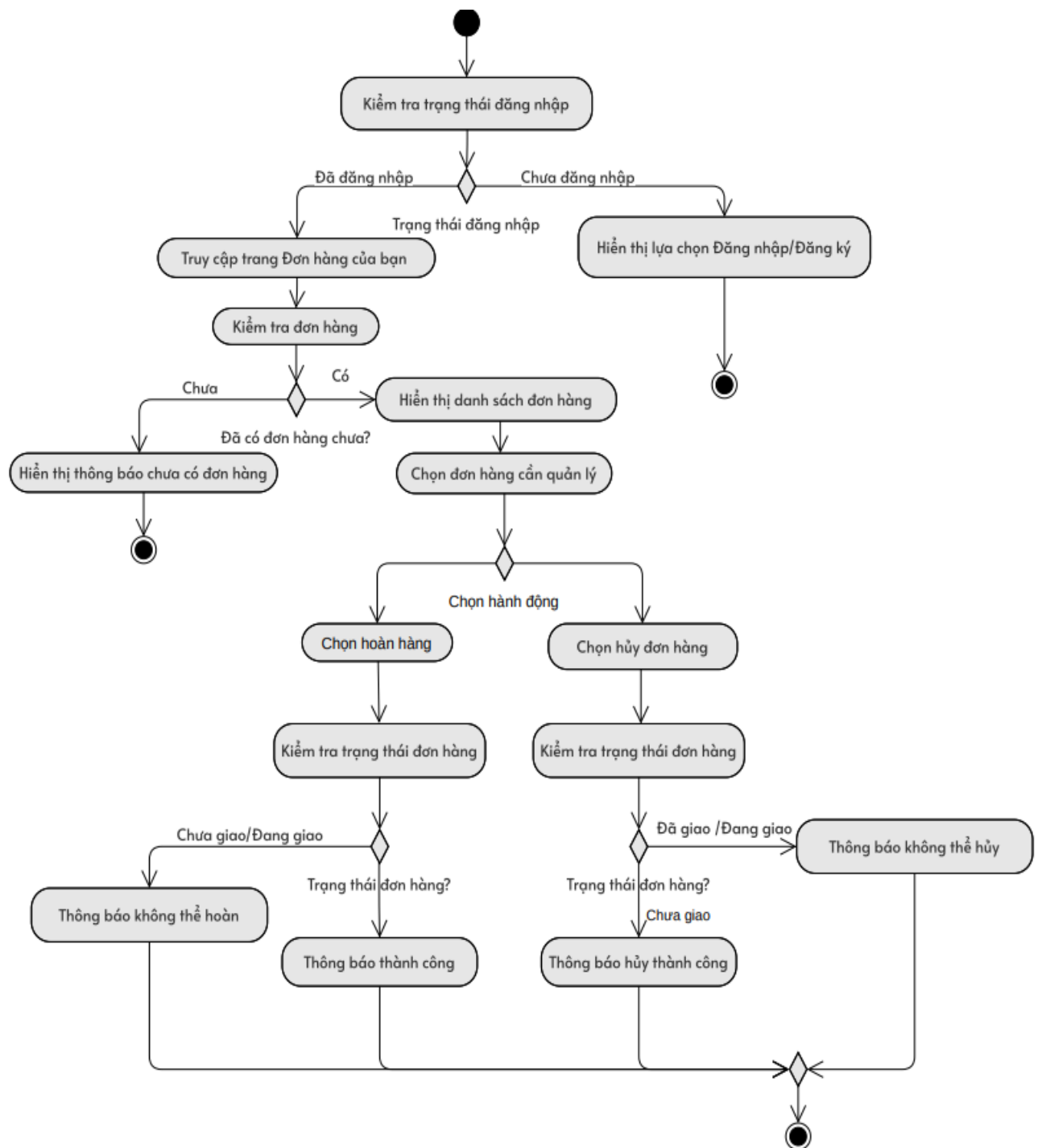
Hình 2.8: Biểu đồ hoạt động use case Quản lý giỏ hàng

* Use case Đặt hàng



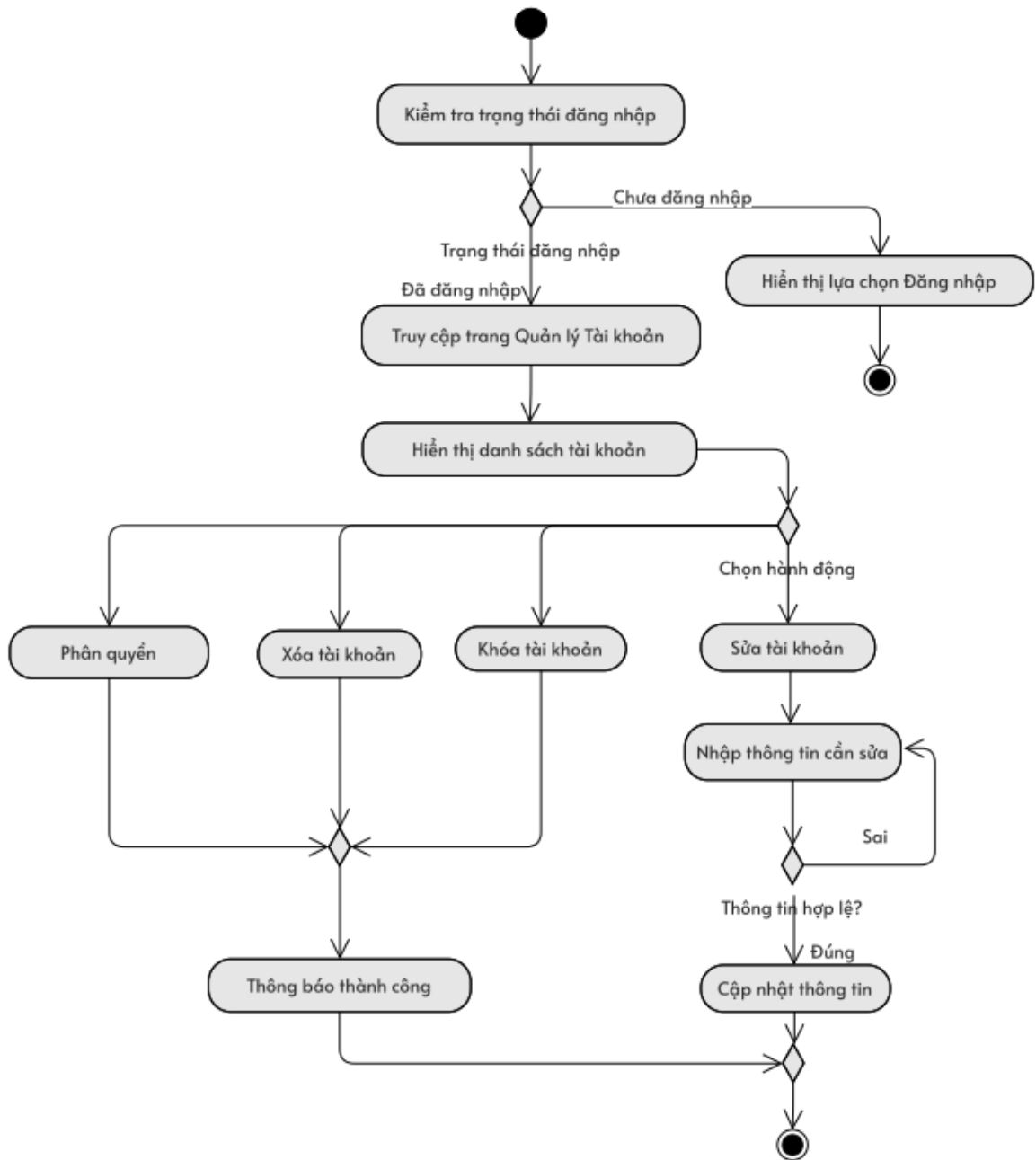
Hình 2.9: Biểu đồ hoạt động use case Đặt hàng

* Use case Quản lý đơn hàng cá nhân



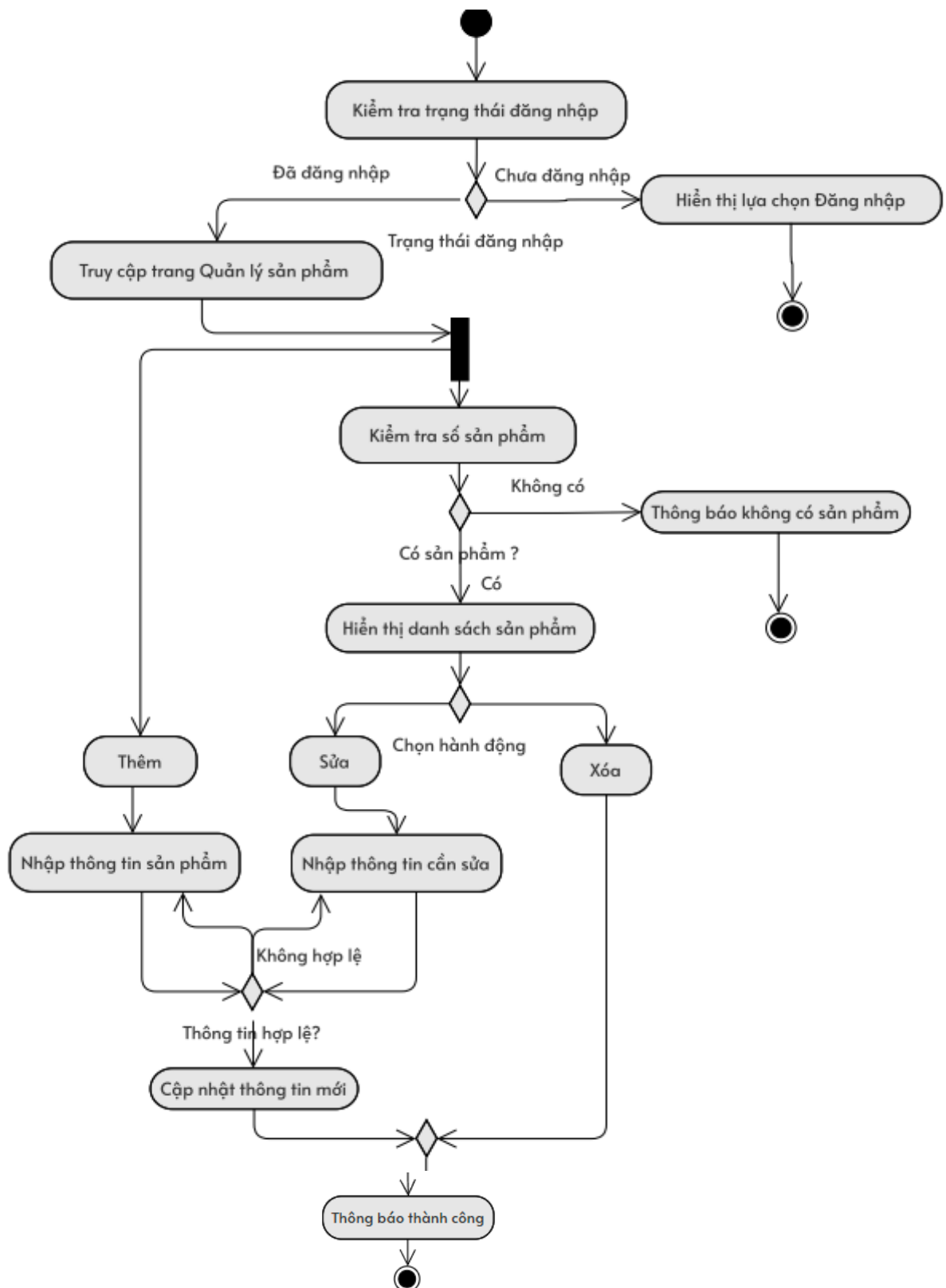
Hình 2.10: Biểu đồ hoạt động use case Quản lý đơn hàng cá nhân

* Use case Quản lý tài khoản



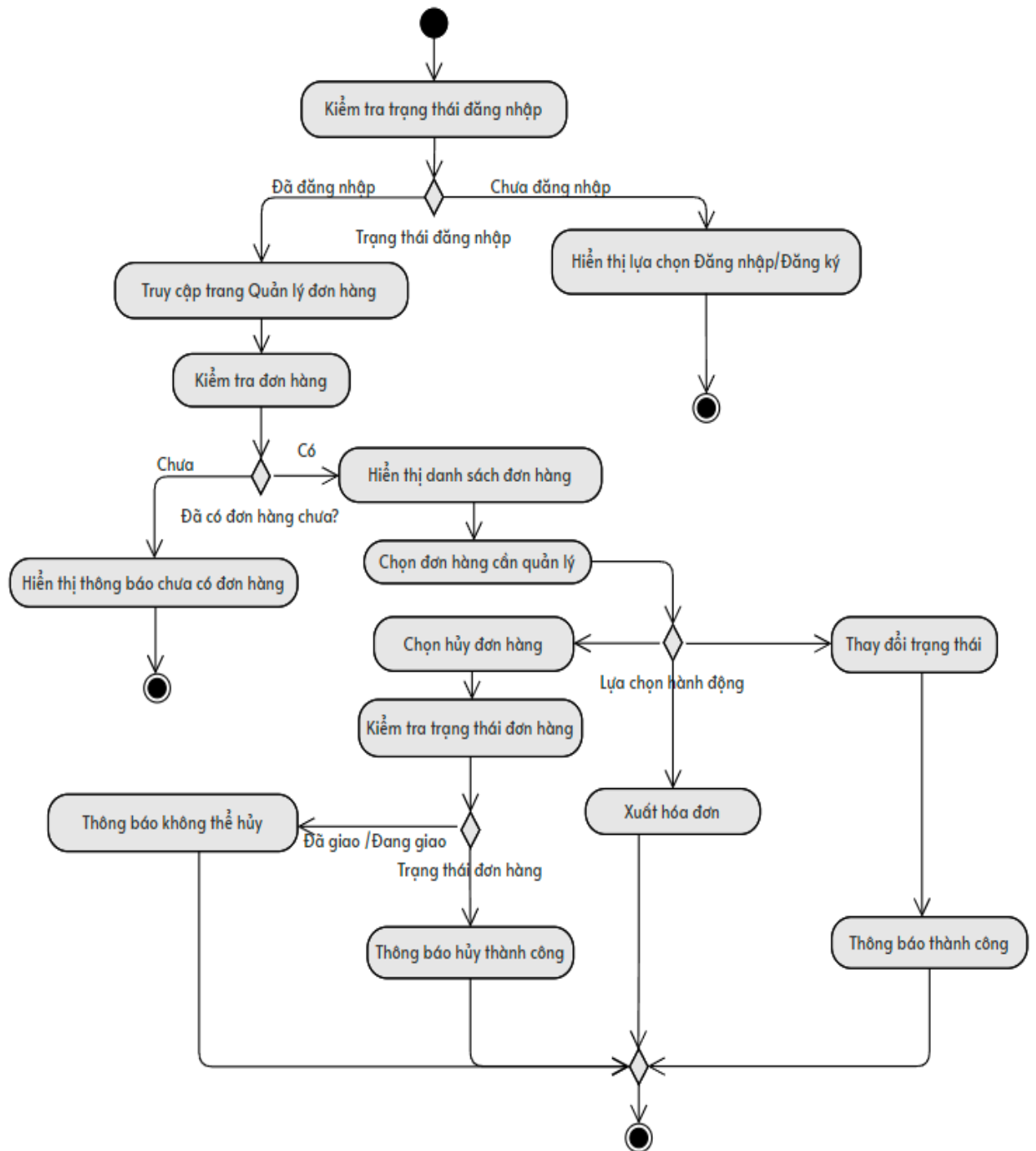
Hình 2.11: Biểu đồ hoạt động use case Quản lý tài khoản

* Use case Quản lý sản phẩm



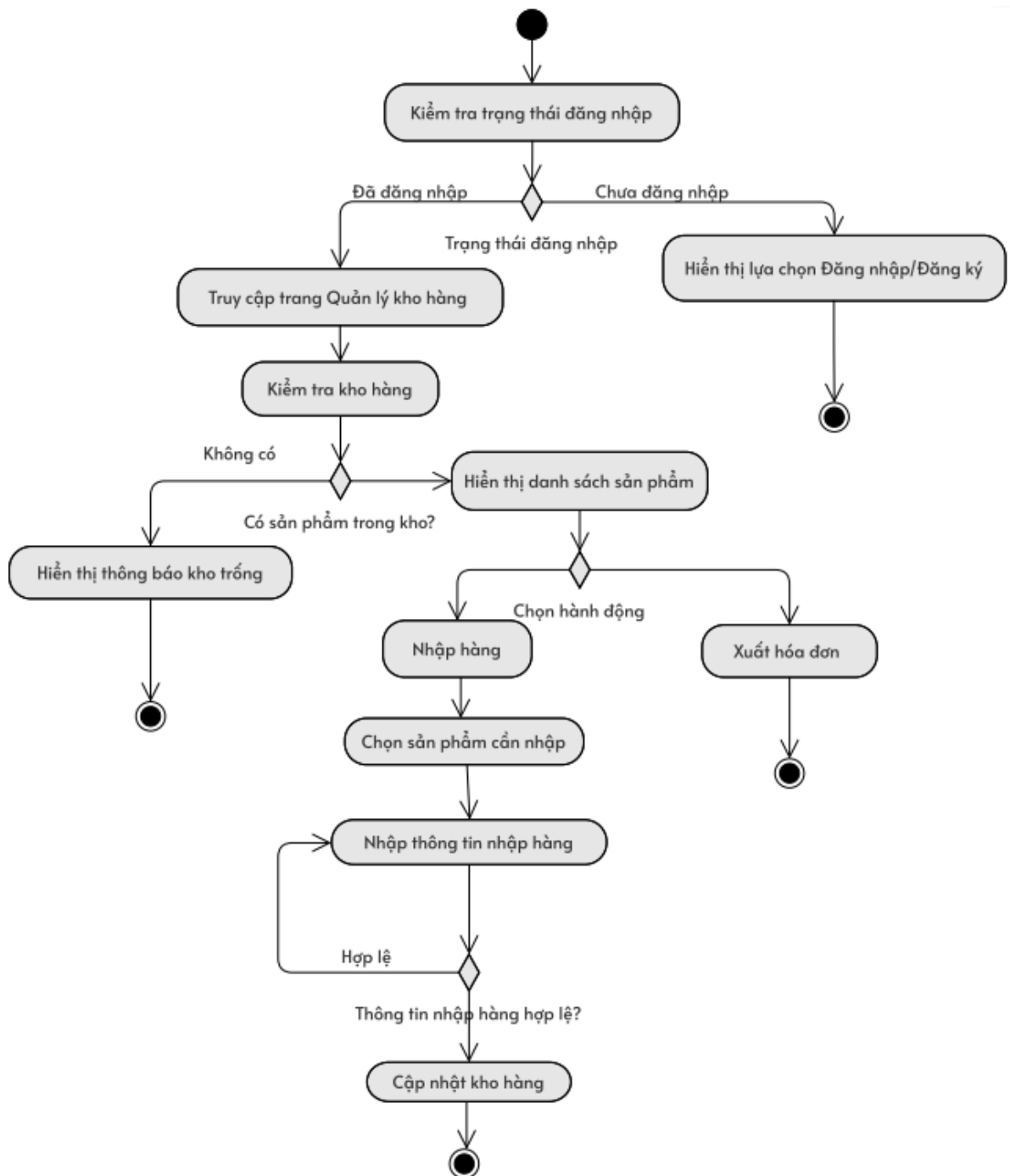
Hình 2.12: Biểu đồ hoạt động use case Quản lý sản phẩm

* Use case Quản lý đơn hàng



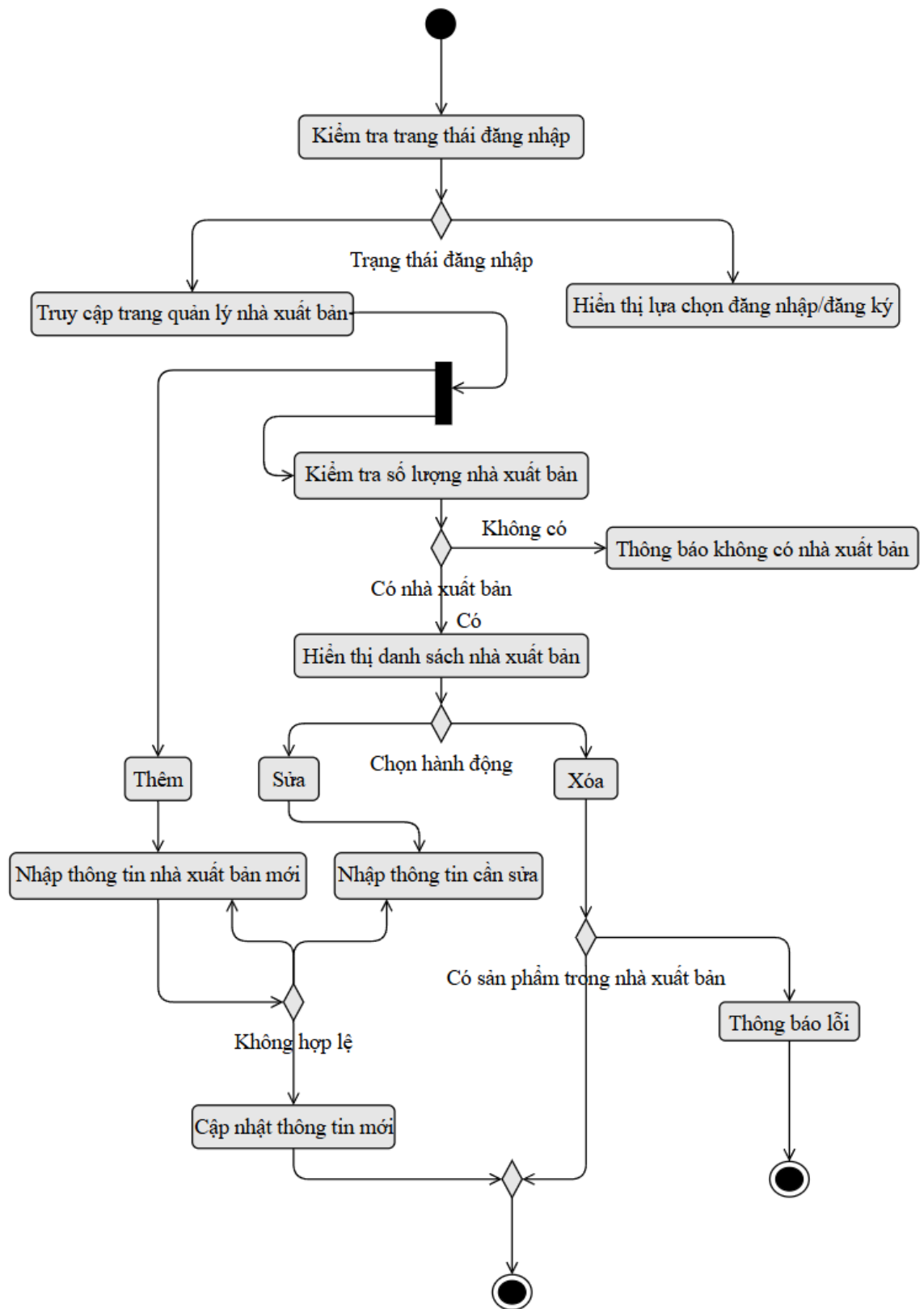
Hình 2.13: Biểu đồ hoạt động use case Quản lý đơn hàng

* Use case Quản lý kho hàng



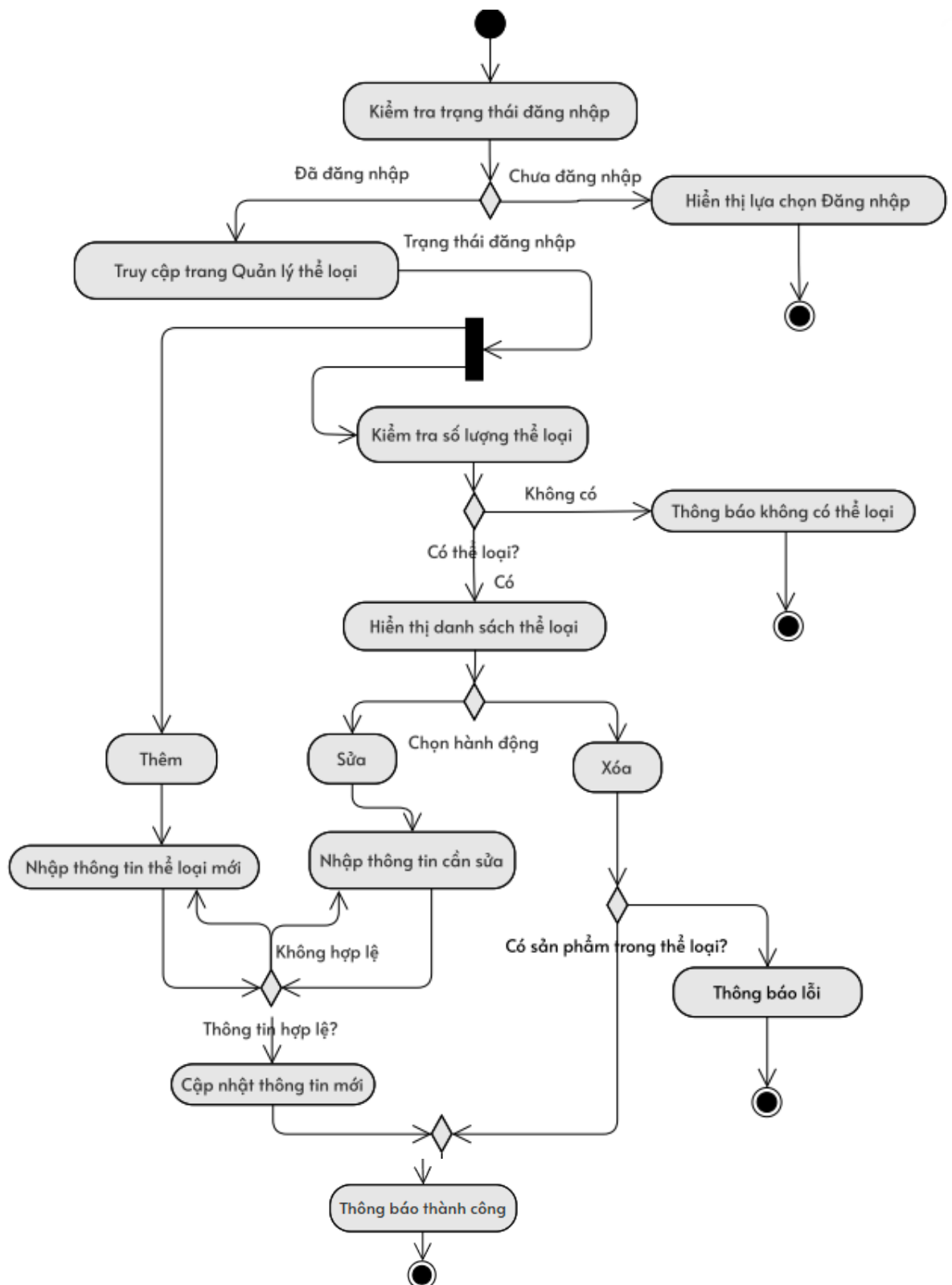
Hình 2.14: Biểu đồ hoạt động use case Quản lý kho hàng

* Use case Quản lý nhà xuất bản



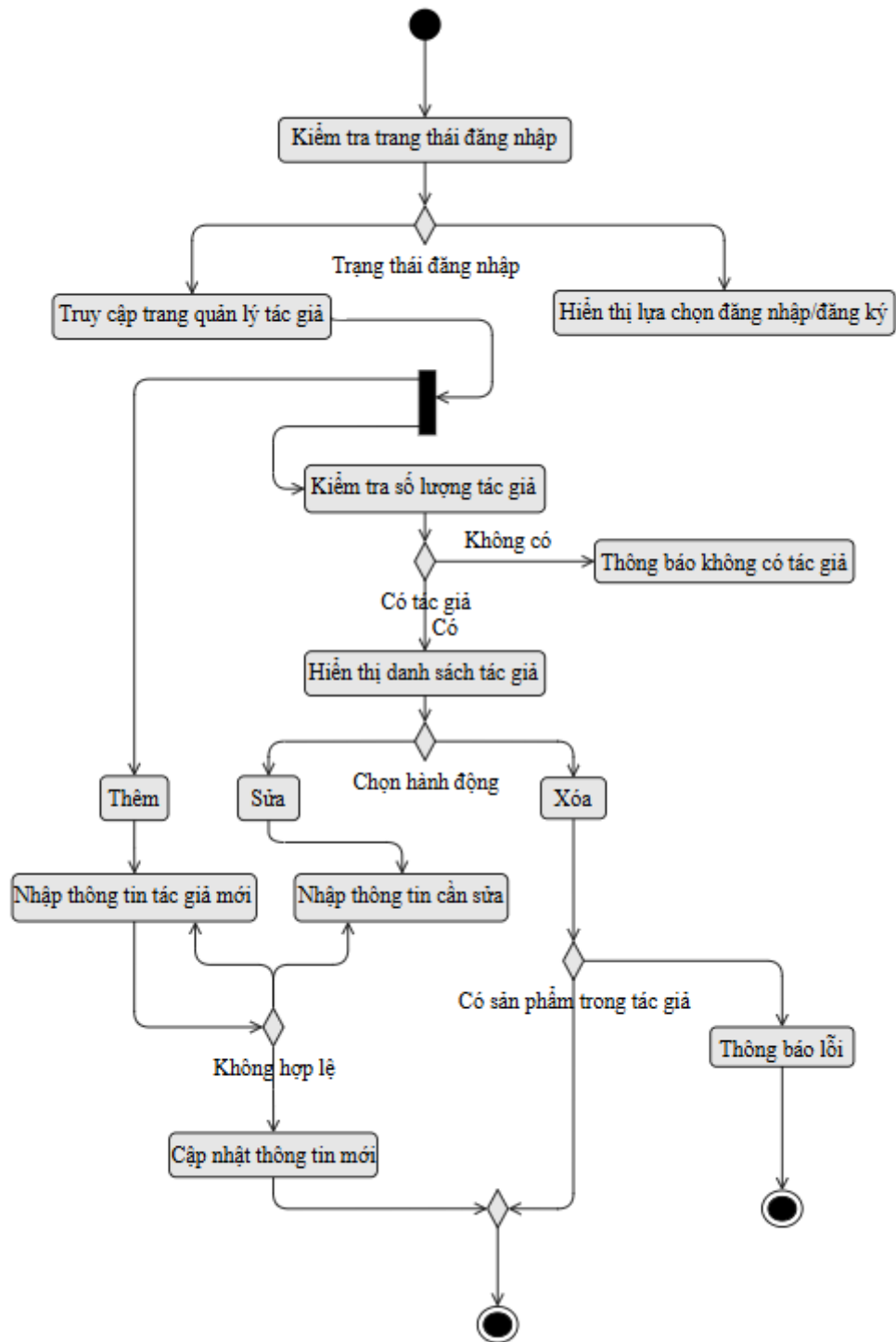
Hình 2.15: Biểu đồ hoạt động use case Quản lý nhà xuất bản

* Use case Quản lý thẻ loại



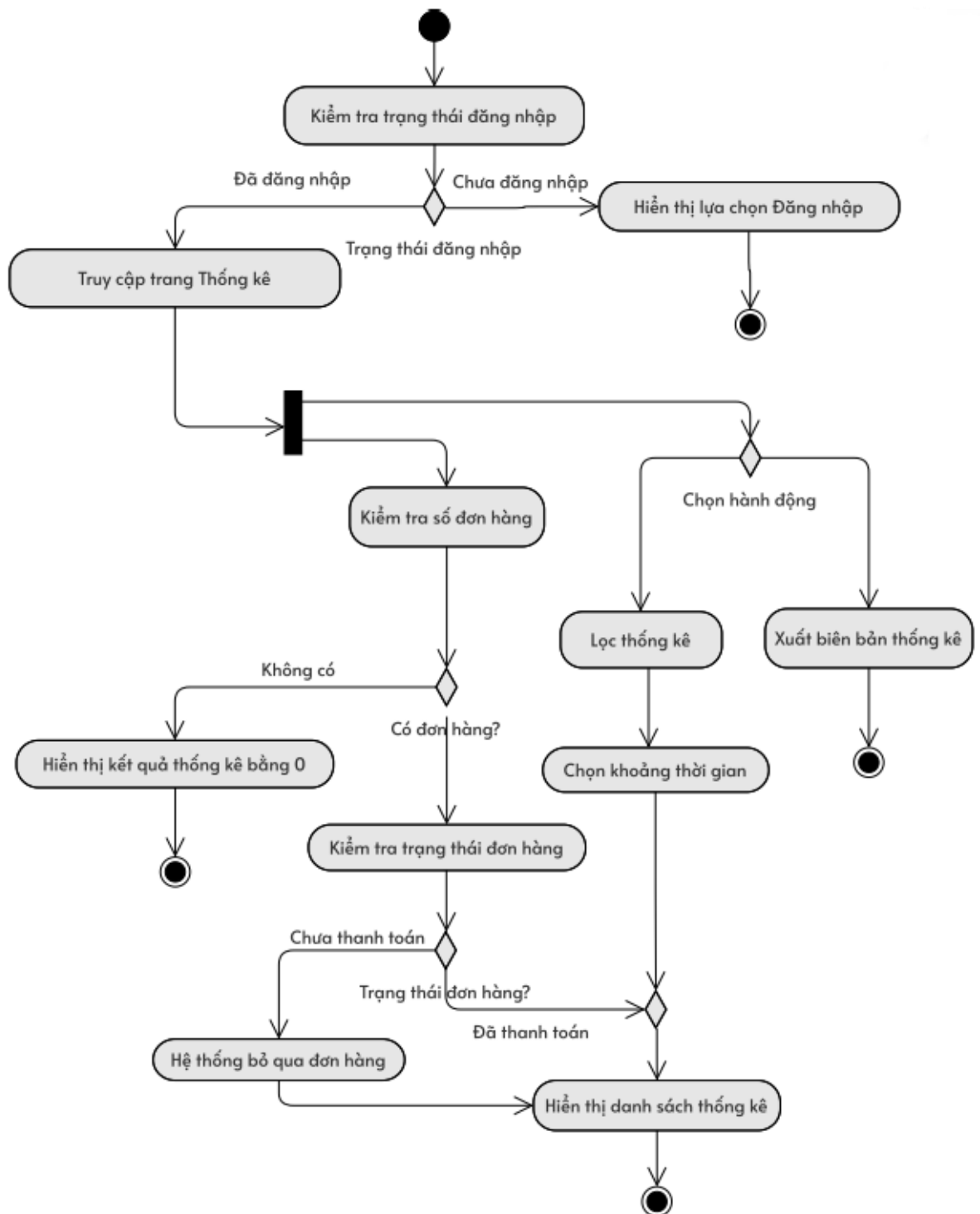
Hình 2.16: Biểu đồ hoạt động use case Quản lý thẻ loại

* Use case Quản lý tác giả



Hình 2.17: Biểu đồ hoạt động use case Quản lý tác giả

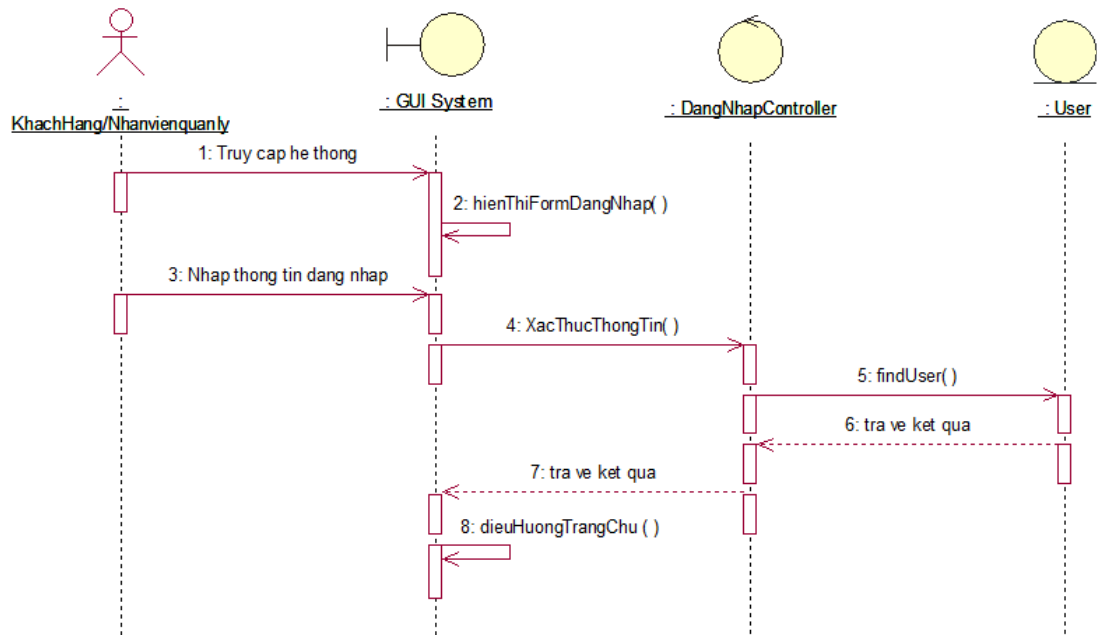
* Use case Thống kê



Hình 2.18: Biểu đồ hoạt động use case Thống kê

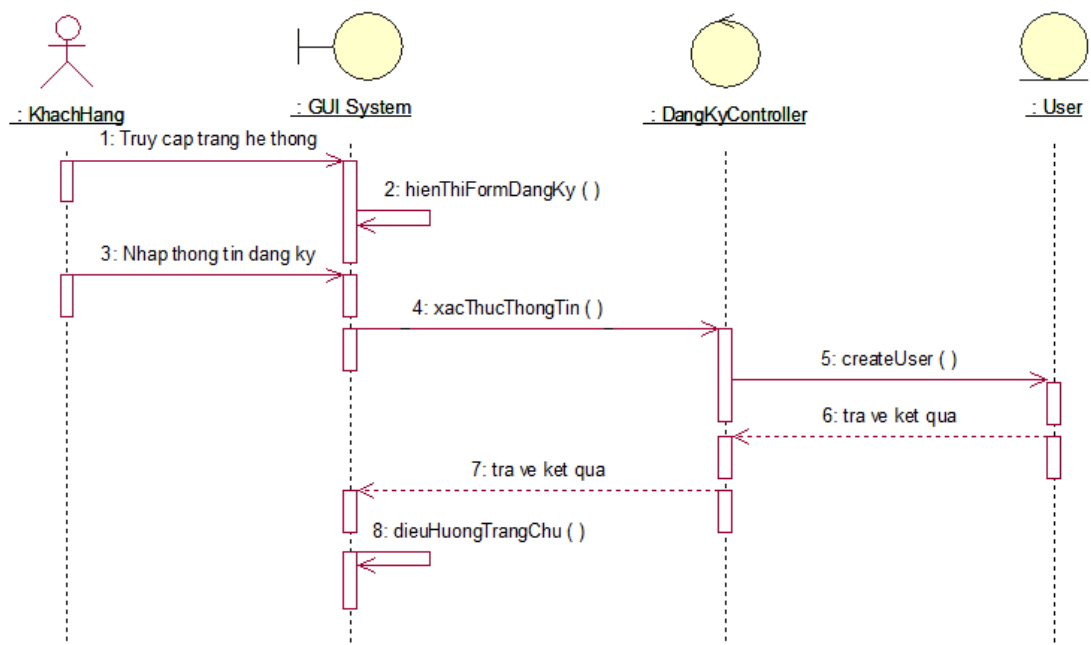
2.4.5. Biểu đồ trình tự

* Use case Đăng nhập



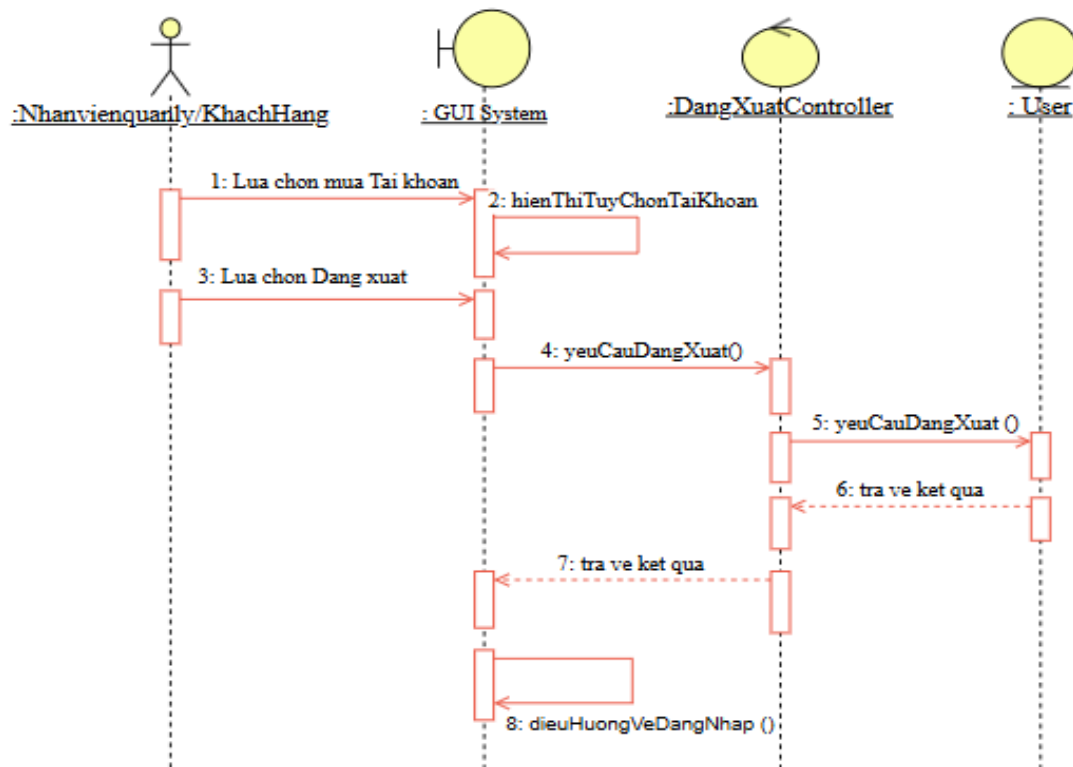
Hình 2.19: Biểu đồ trình tự use case đăng nhập

* Use case Đăng ký



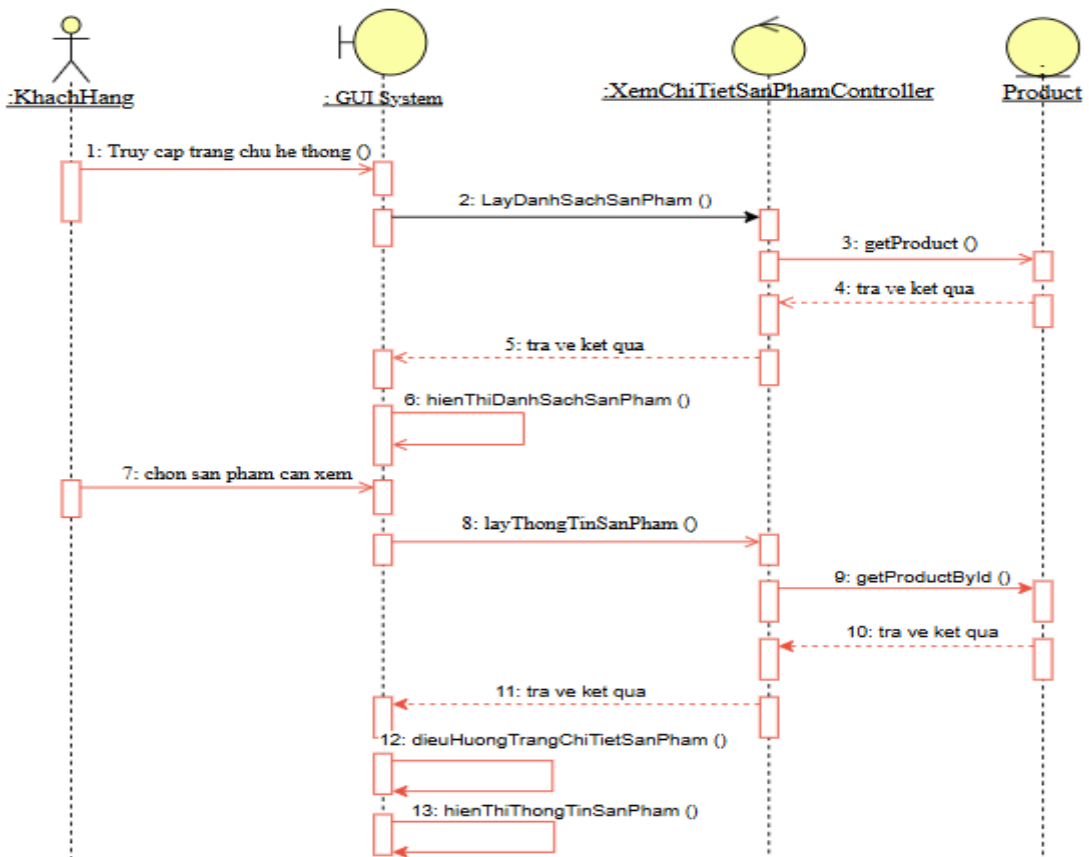
Hình 2.20: Biểu đồ trình tự use case đăng ký

* Use case đăng xuất



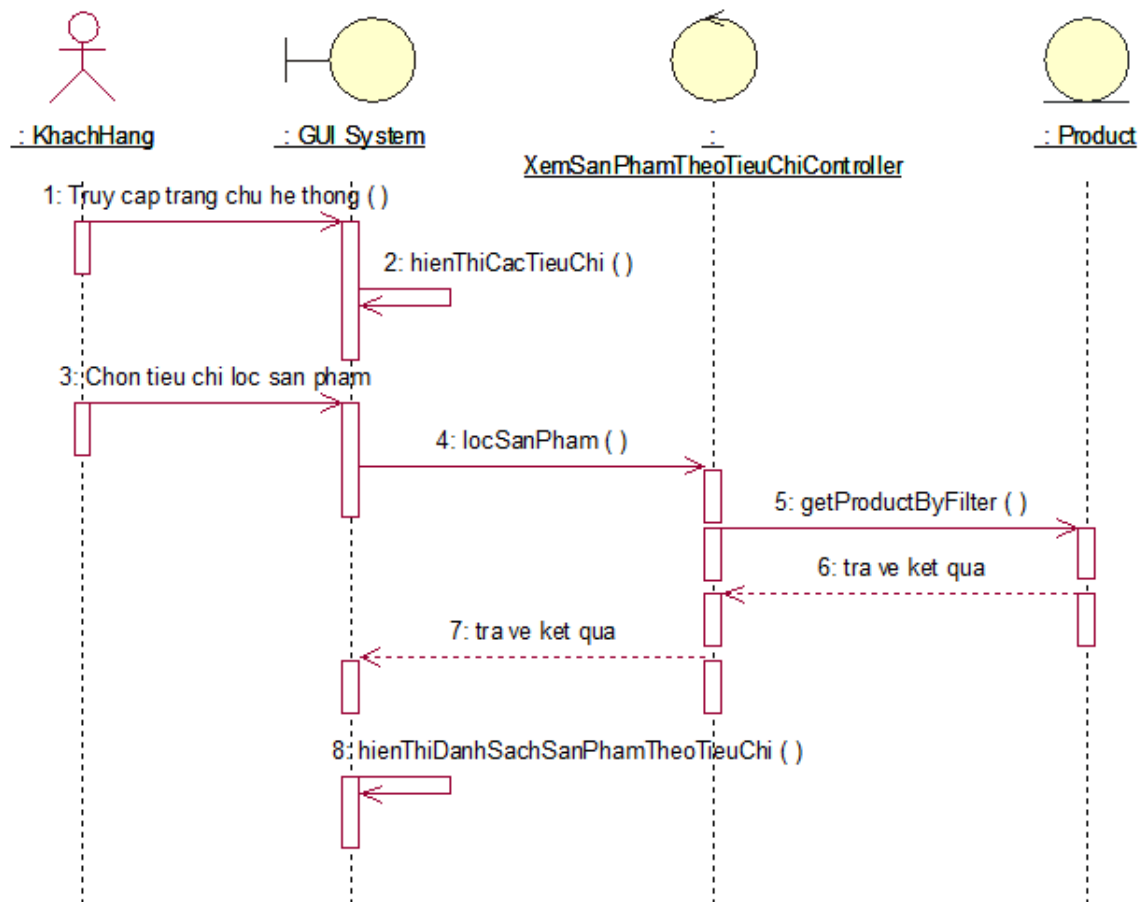
Hình 2.21: Biểu đồ trình tự use case đăng xuất

* Use case Xem chi tiết sản phẩm



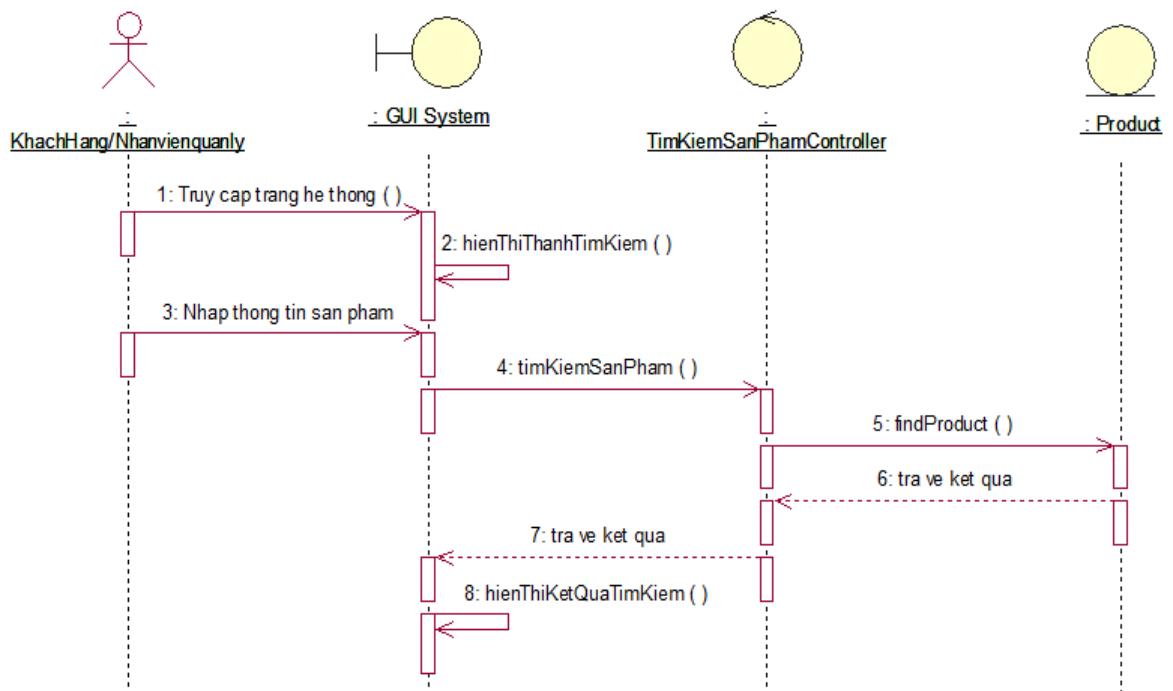
Hình 2.22: Biểu đồ trình tự use case xem chi tiết sản phẩm

* Use case Xem sản phẩm theo tiêu chí



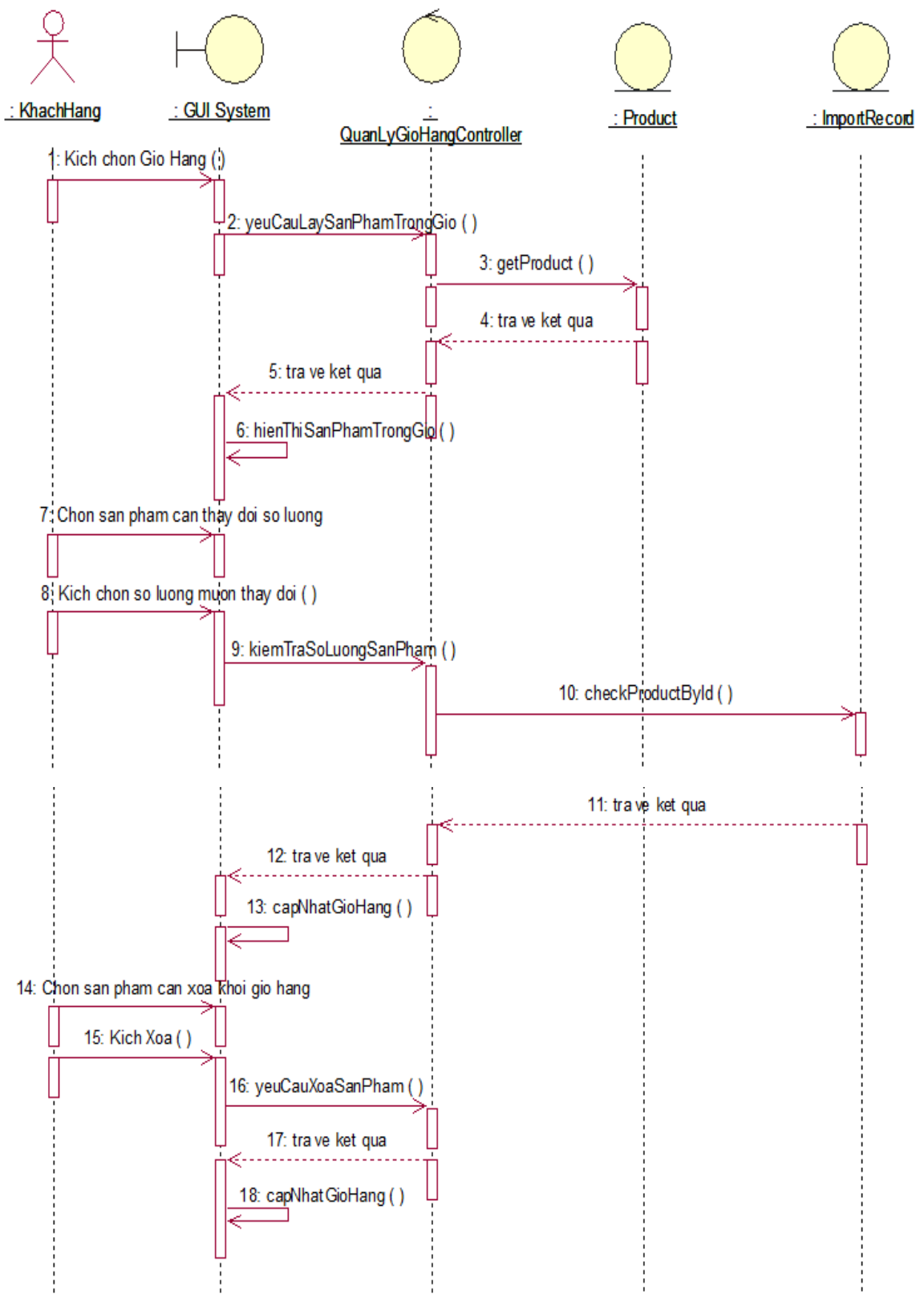
Hình 2.23: Biểu đồ trình tự use case Xem sản phẩm theo tiêu chí

* Use case Tìm kiếm sản phẩm



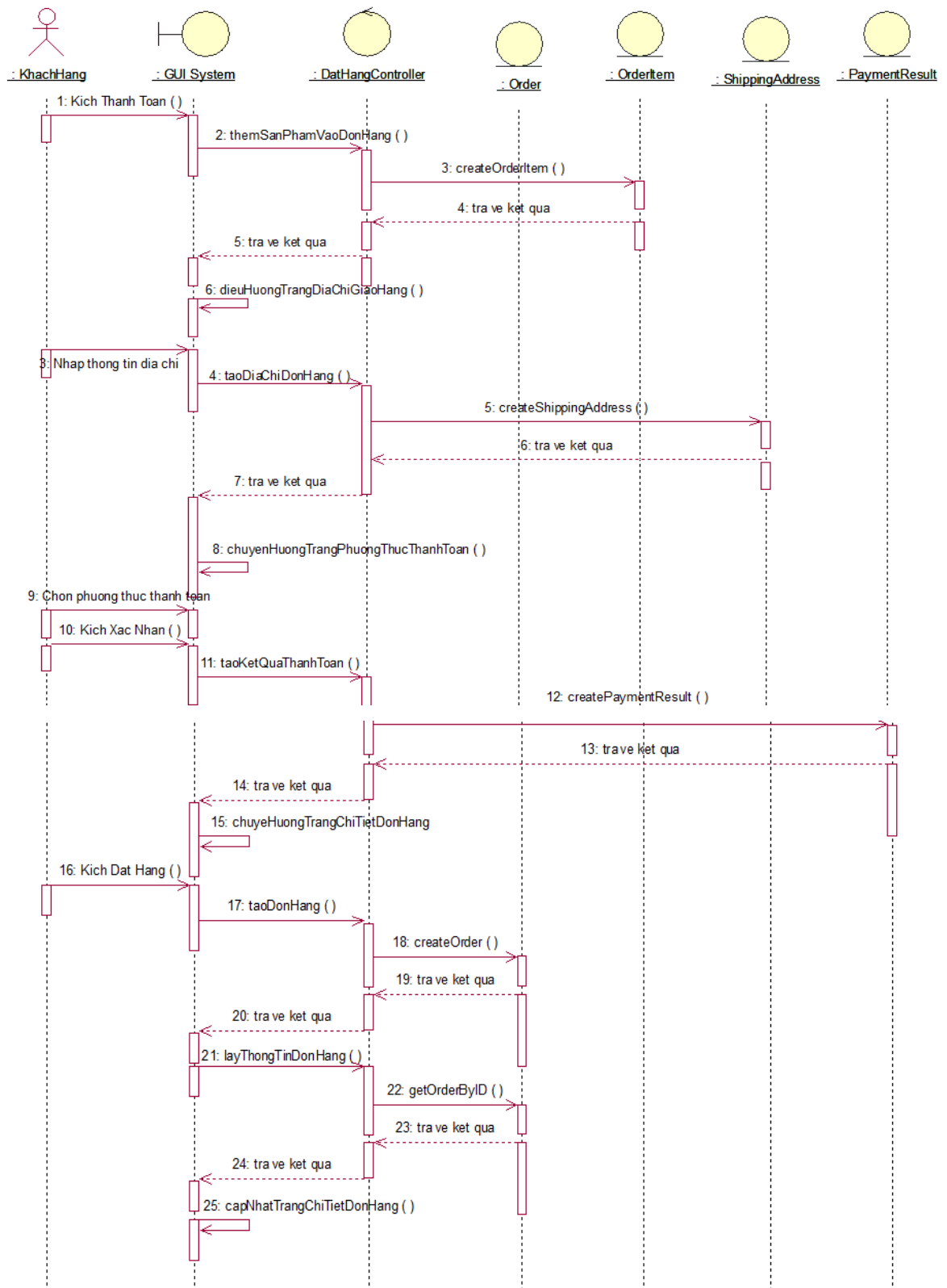
Hình 2.24: Biểu đồ trình tự use case Tìm kiếm sản phẩm

* Use case Quản lý giỏ hàng



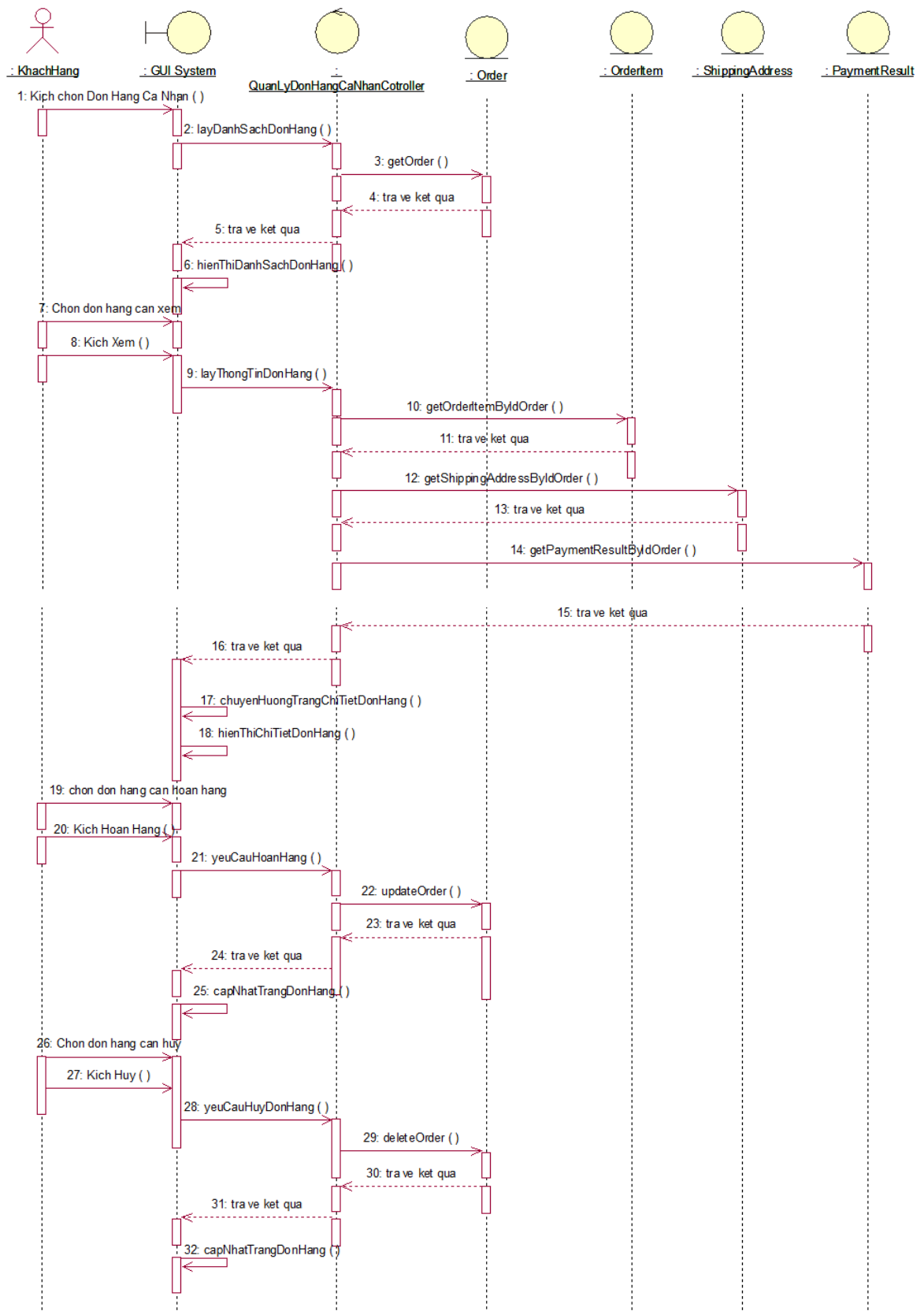
Hình 2.25: Biểu đồ trình tự use case Quản lý giỏ hàng

* Use case Đặt hàng



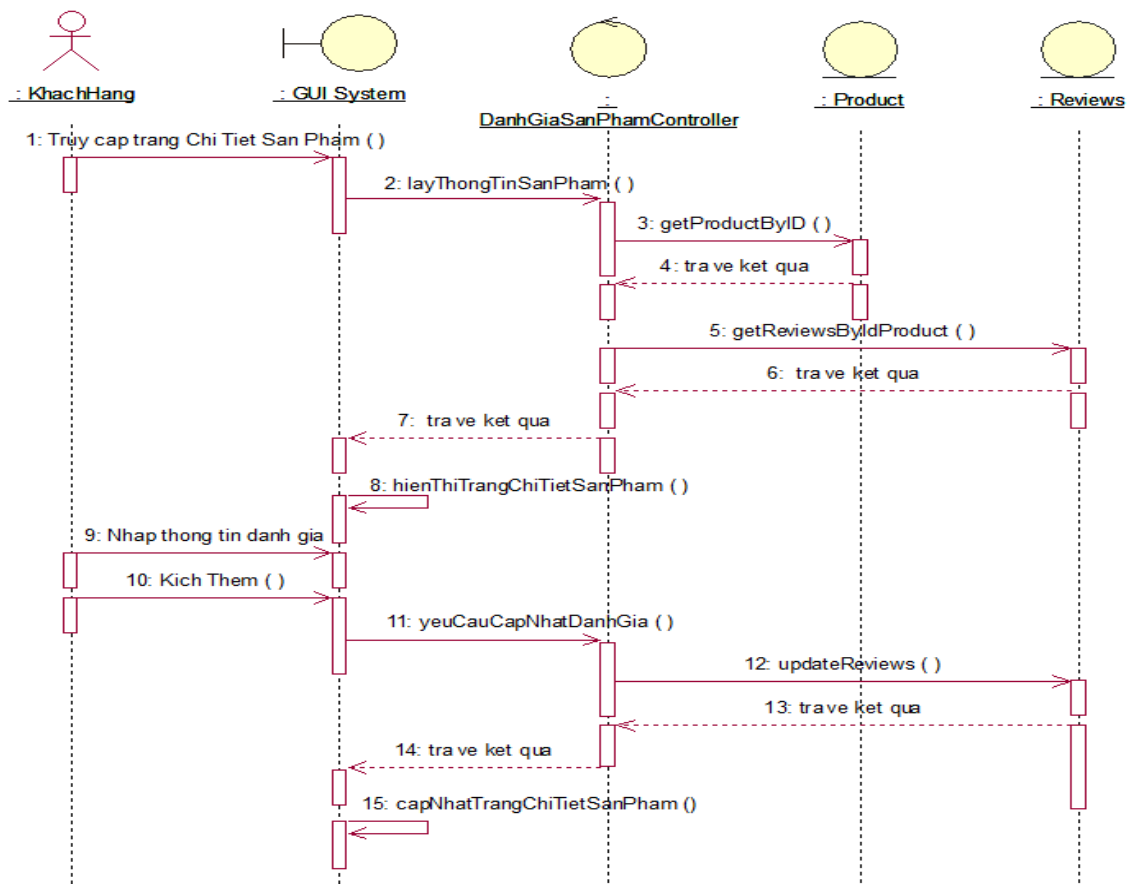
Hình 2.26: Biểu đồ trình tự use case Đặt hàng

* Use case Quản lý đơn hàng cá nhân



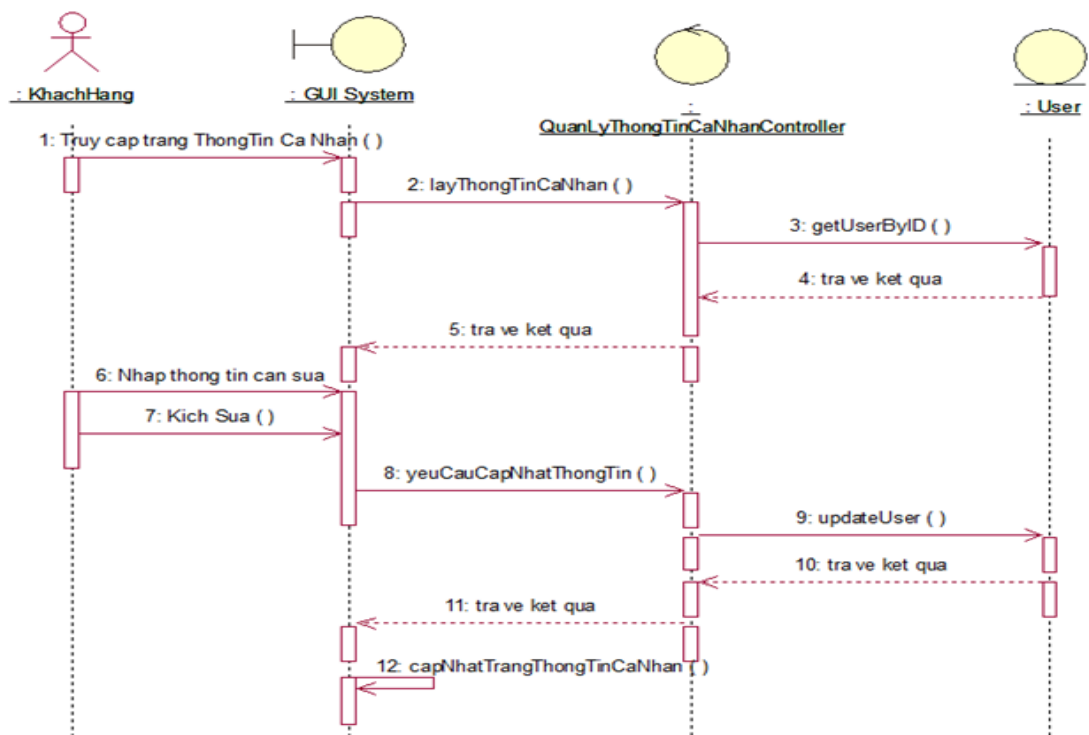
Hình 2.27: Biểu đồ trình tự use case Quản lý đơn hàng cá nhân

* Use case Đánh giá sản phẩm



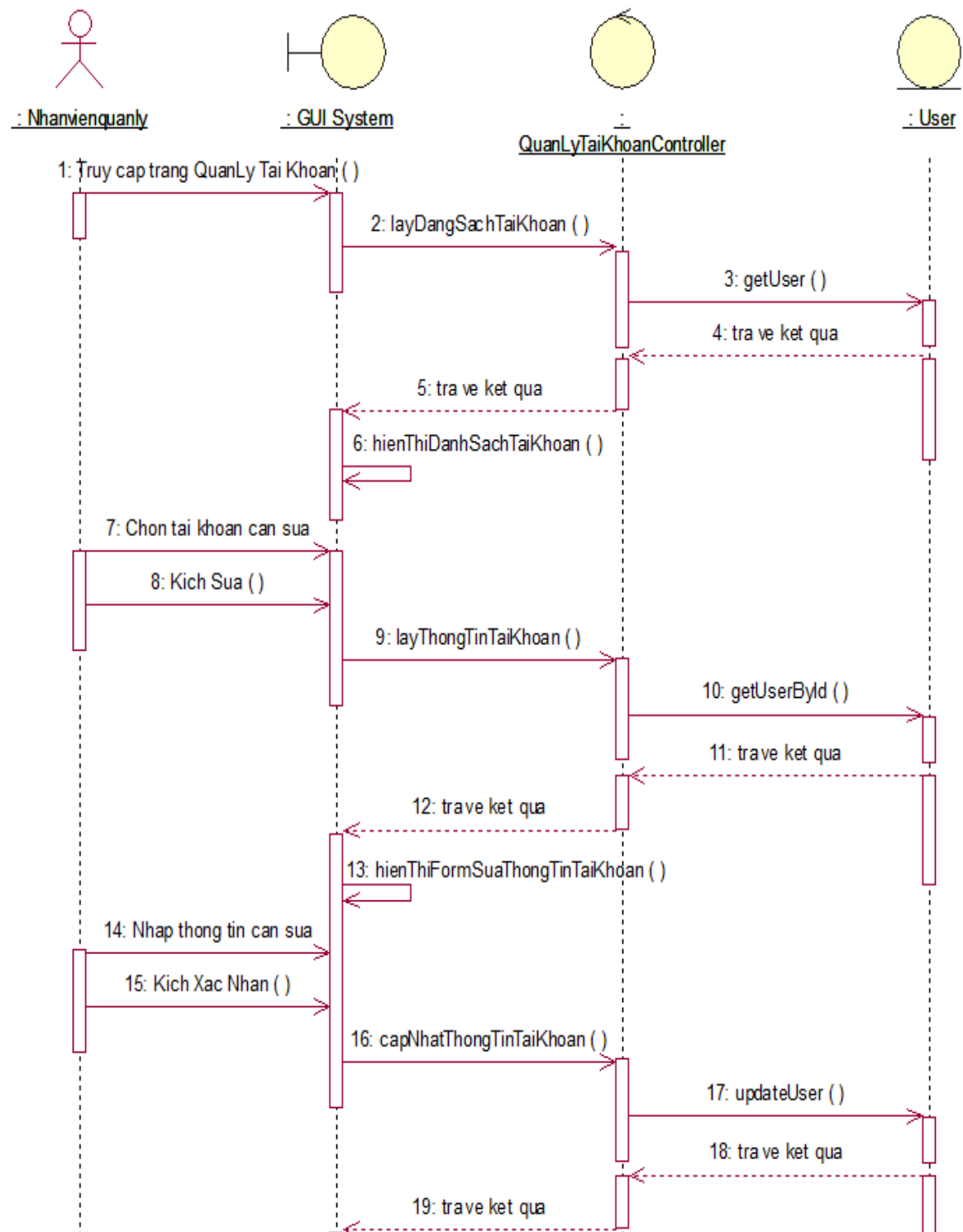
Hình 2.28: Biểu đồ trình tự use case Đánh giá sản phẩm

* Use case Quản lý thông tin cá nhân

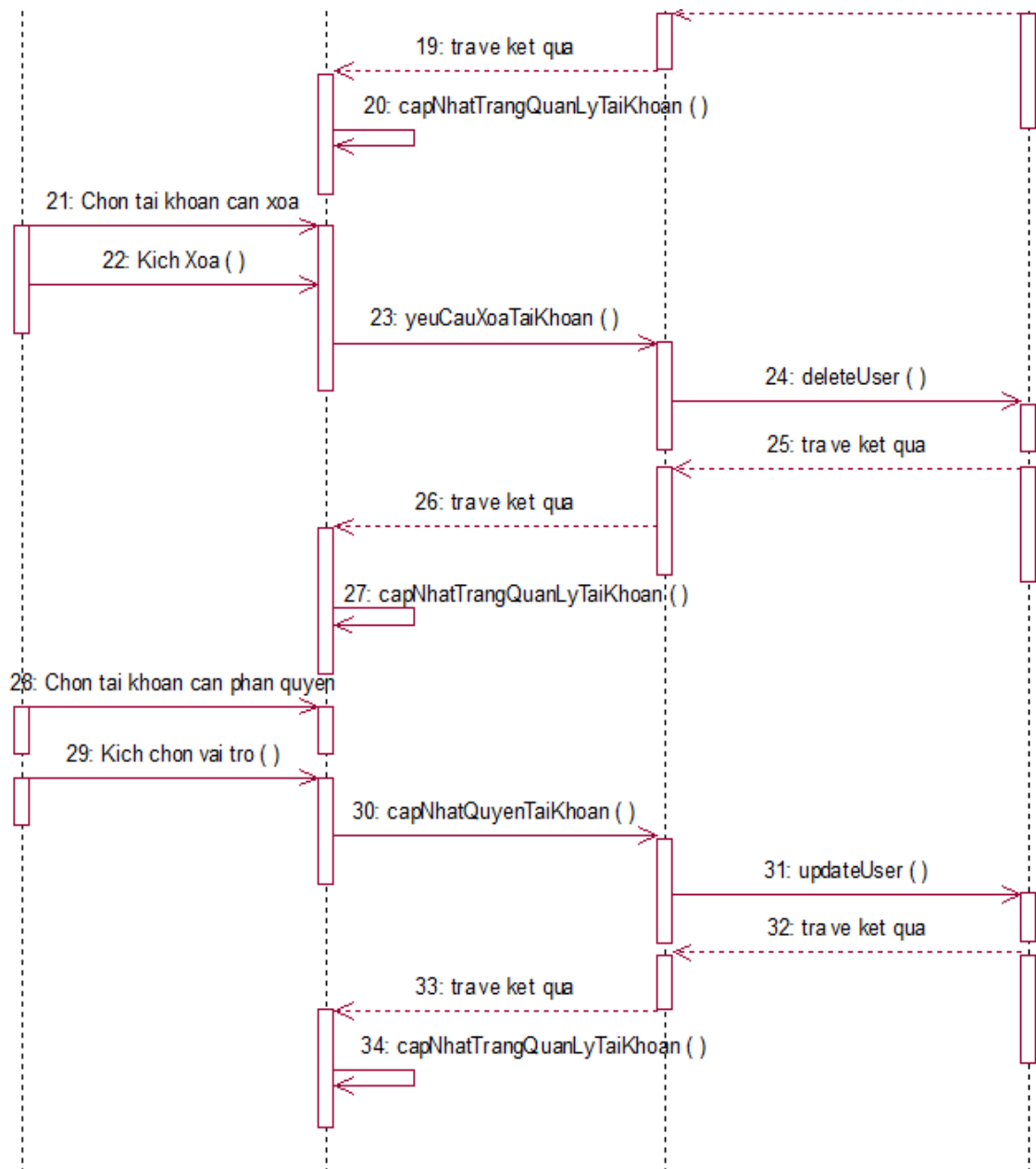


Hình 2.29: Biểu đồ trình tự use case Quản lý thông tin cá nhân

* Use case Quản lý tài khoản

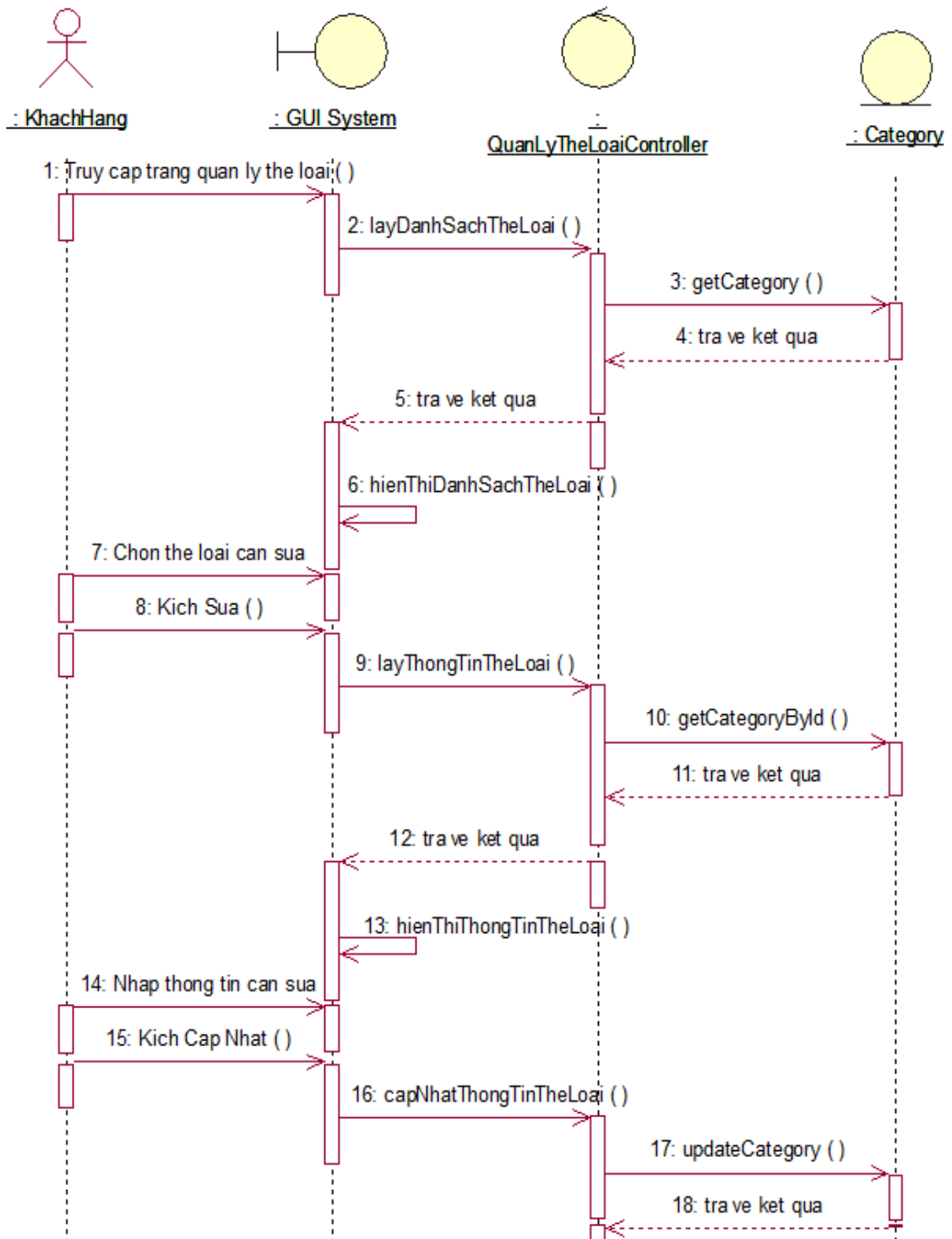


Hình 2.30: Biểu đồ trình tự use case Quản lý tài khoản (1)

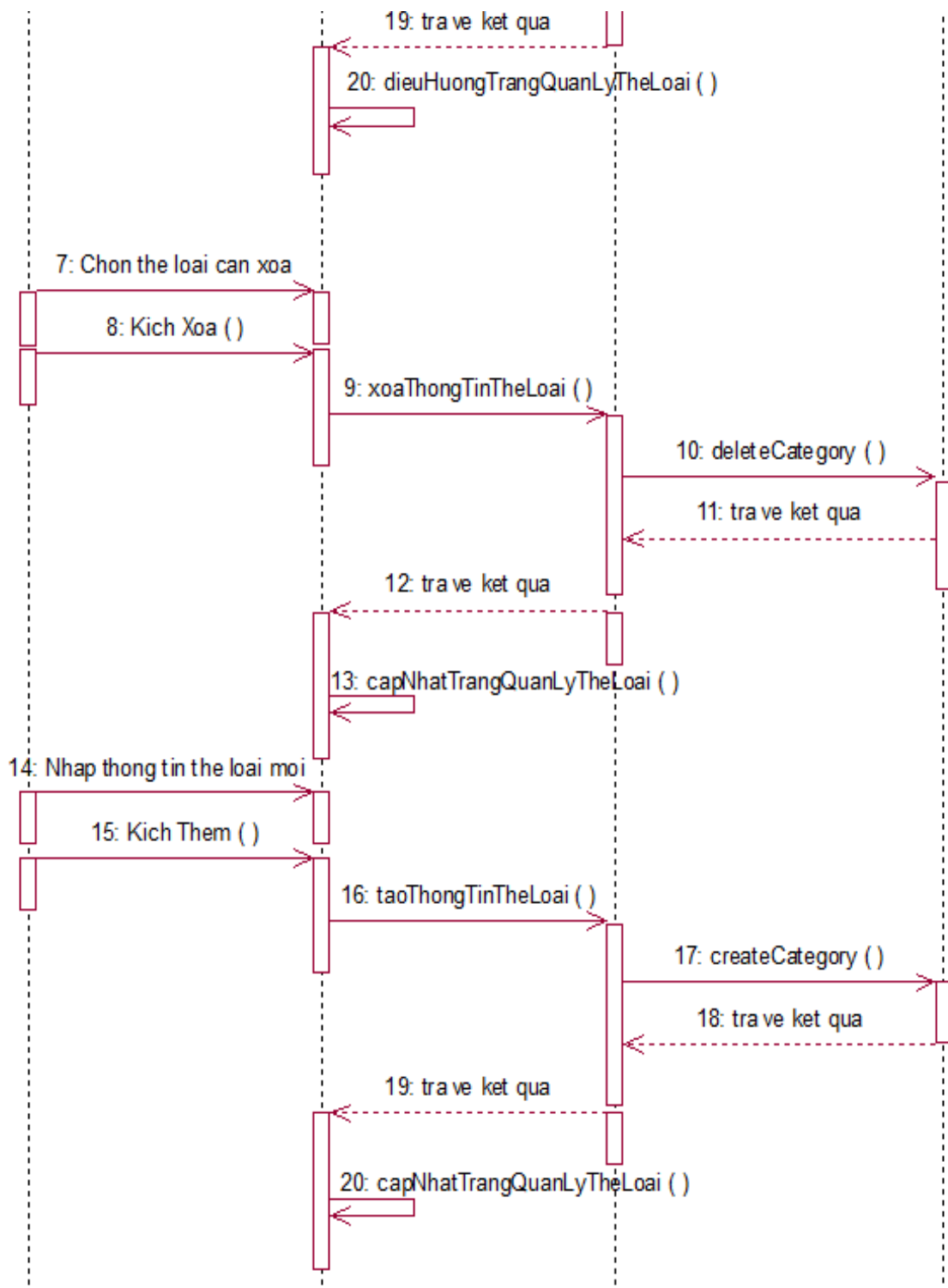


Hình 2.31: Biểu đồ trình tự use case Quản lý tài khoản (2)

* Use case Quản lý thể loại

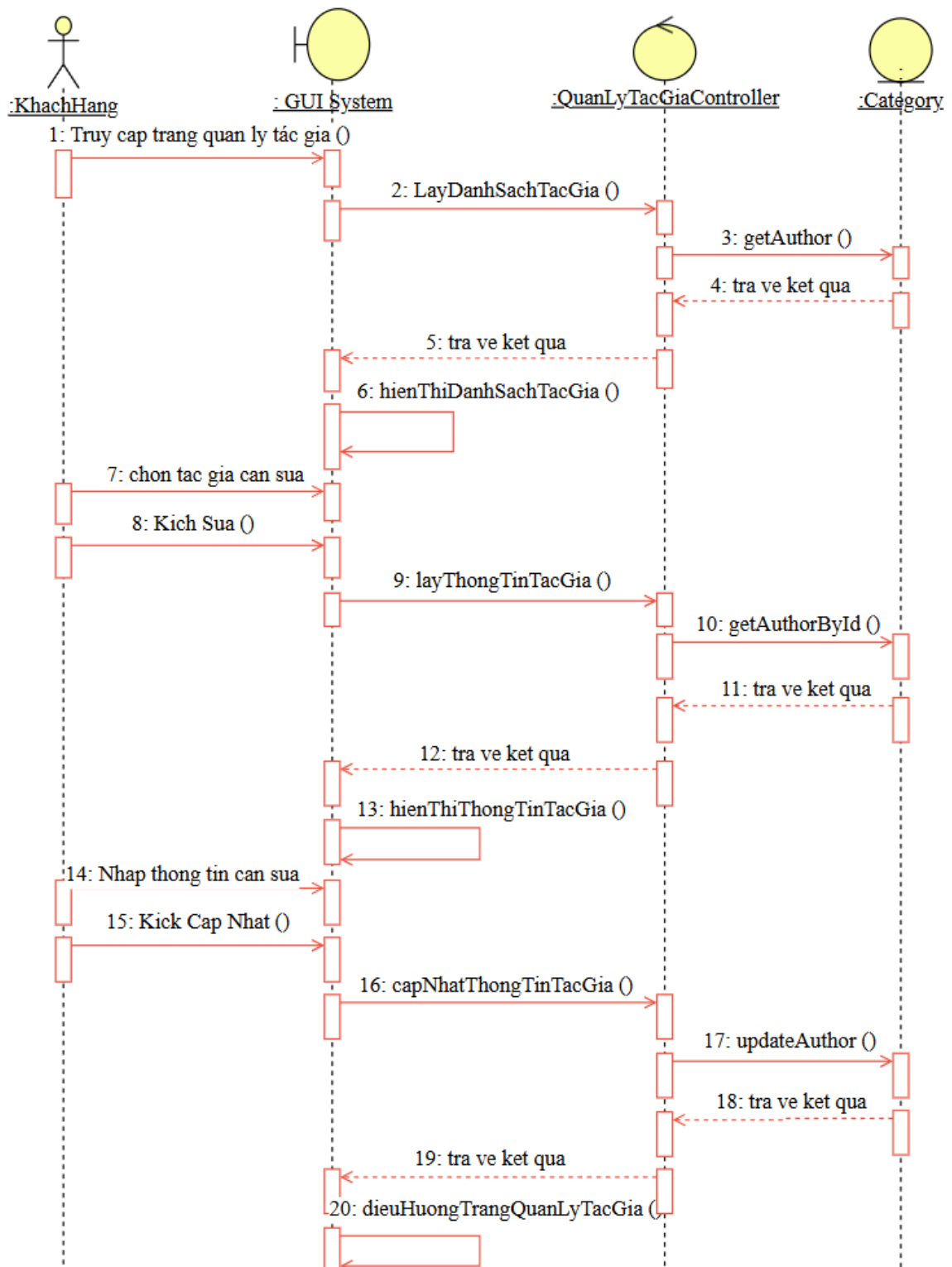


Hình 2.32: Biểu đồ trình tự use case Quản lý thể loại (1)

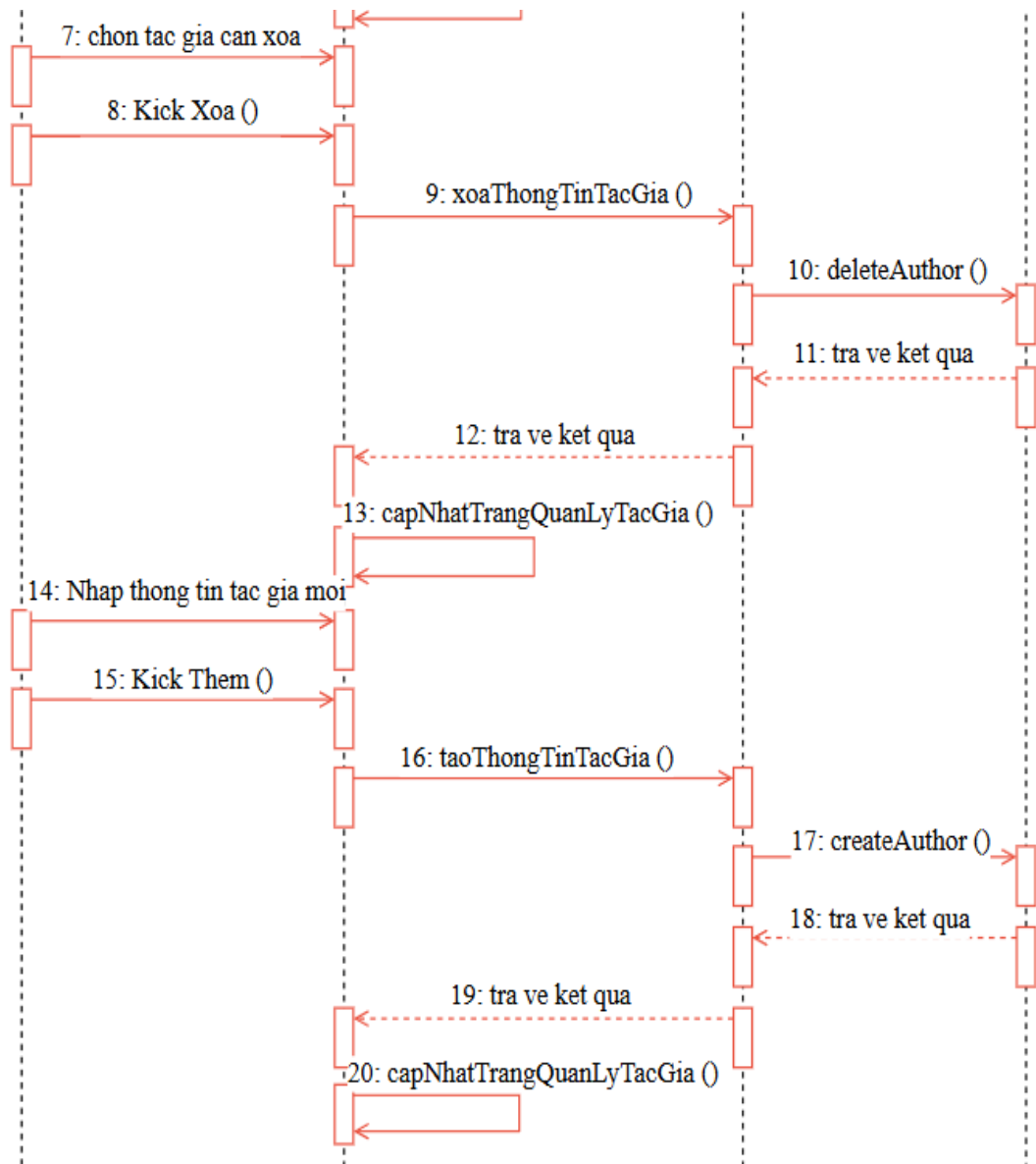


Hình 2.33: Biểu đồ trình tự use case Quản lý thể loại (2)

* Use case Quản lý tác giả

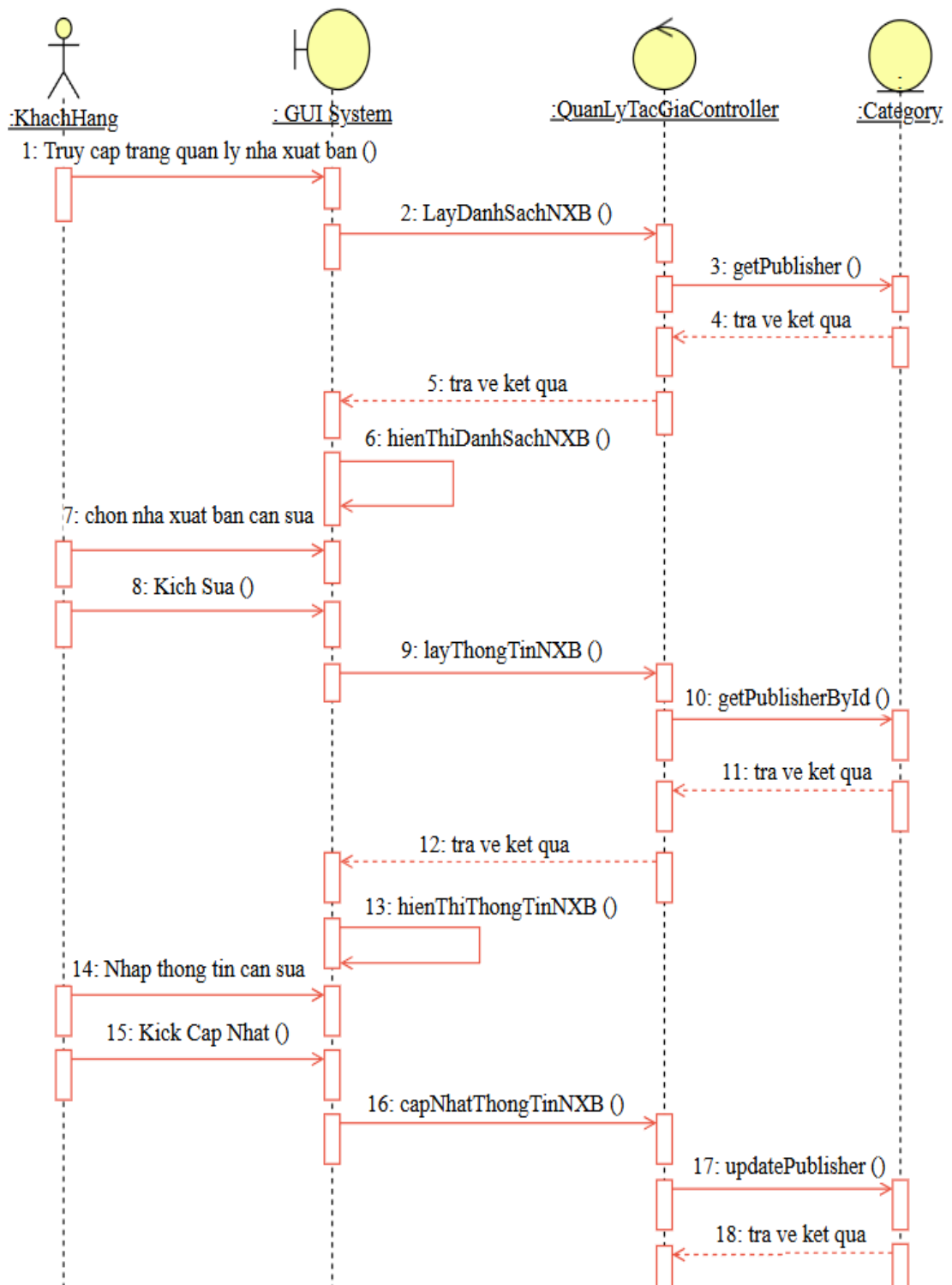


Hình 2.34: Biểu đồ trình tự use case Quản lý tác giả (1)

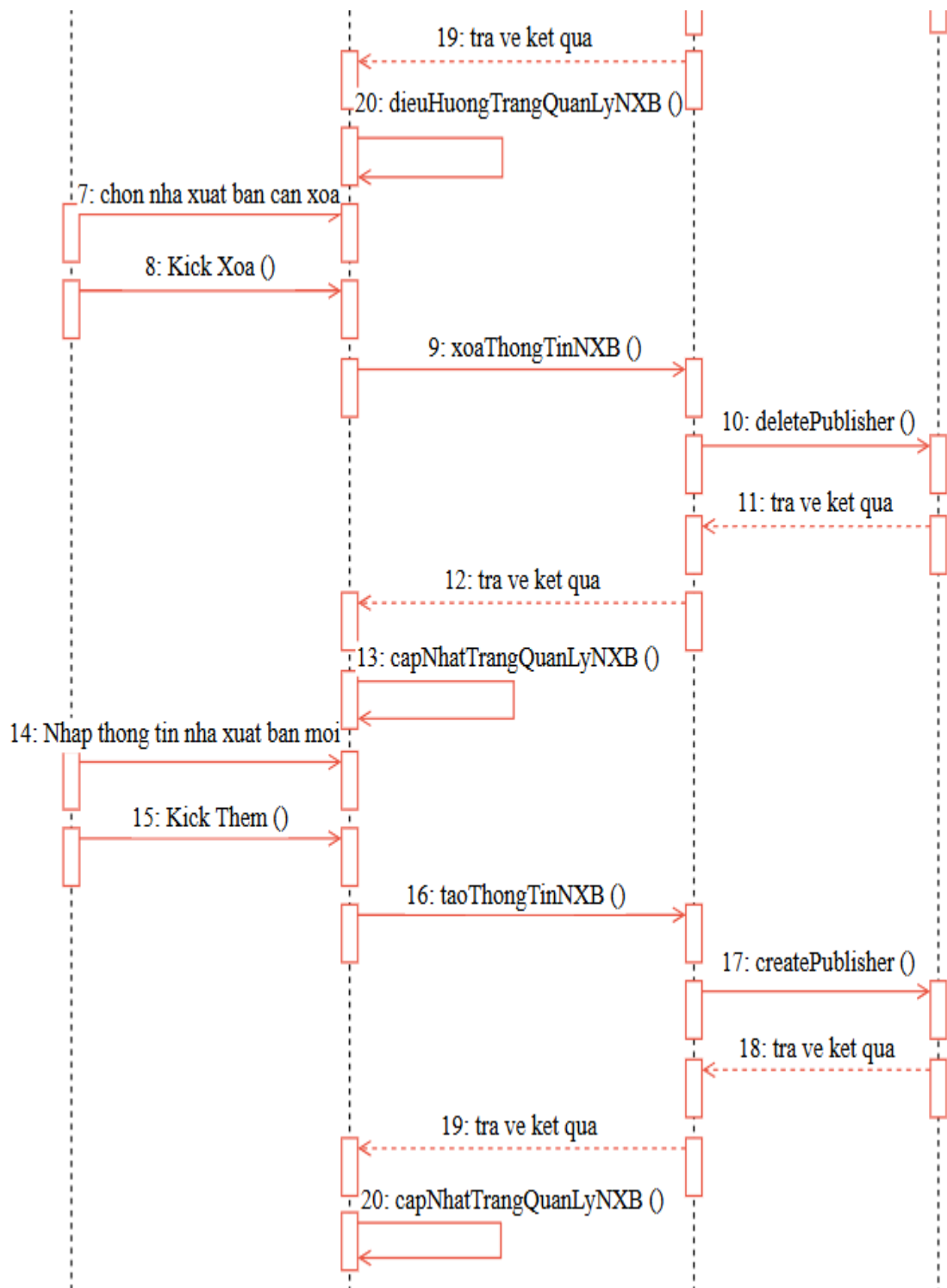


Hình 2.35: Biểu đồ trình tự use case Quản lý tác giả (2)

* Use case Quản lý nhà xuất bản

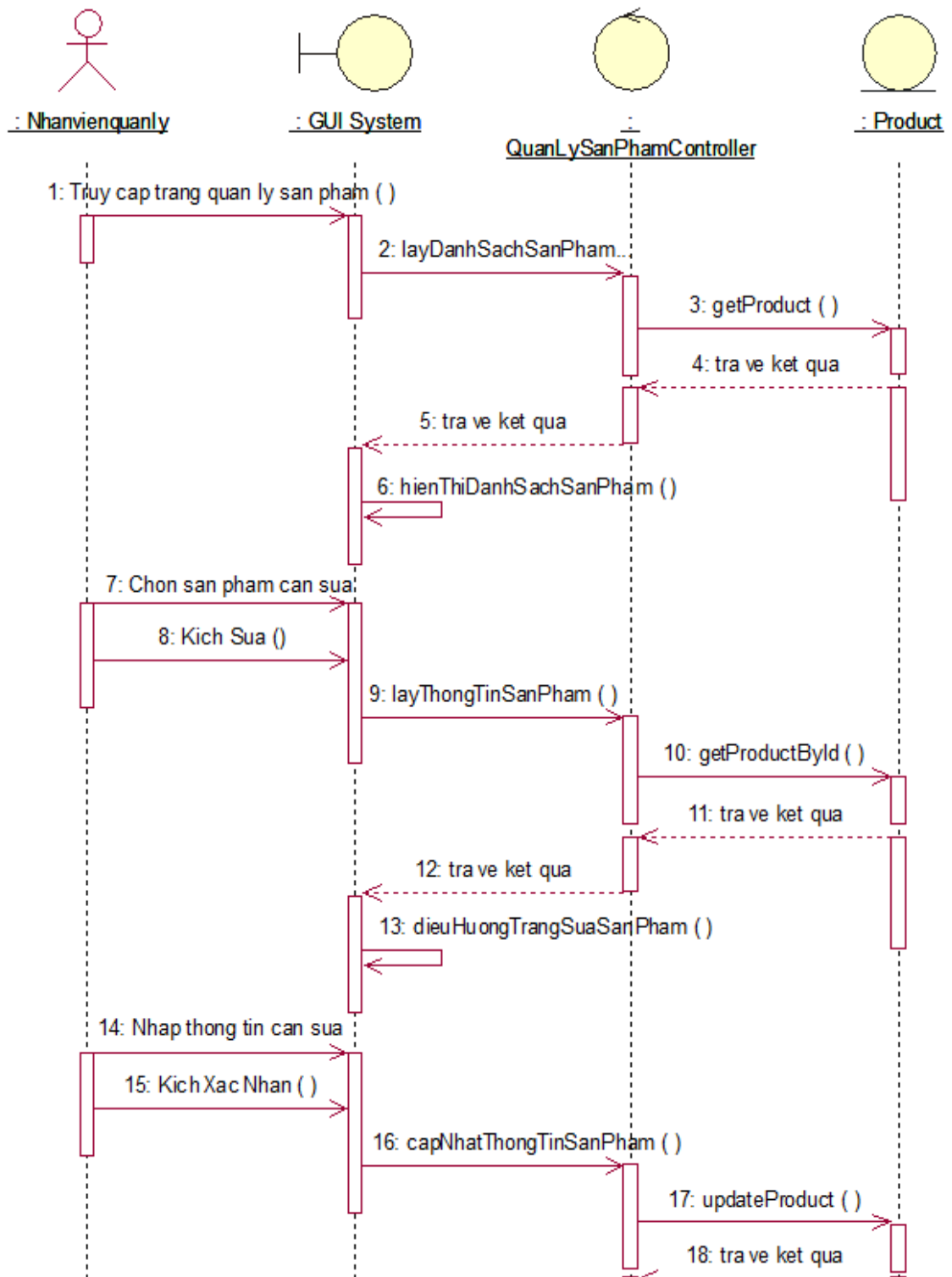


Hình 2.36: Biểu đồ trình tự use case Quản lý nhà xuất bản (1)

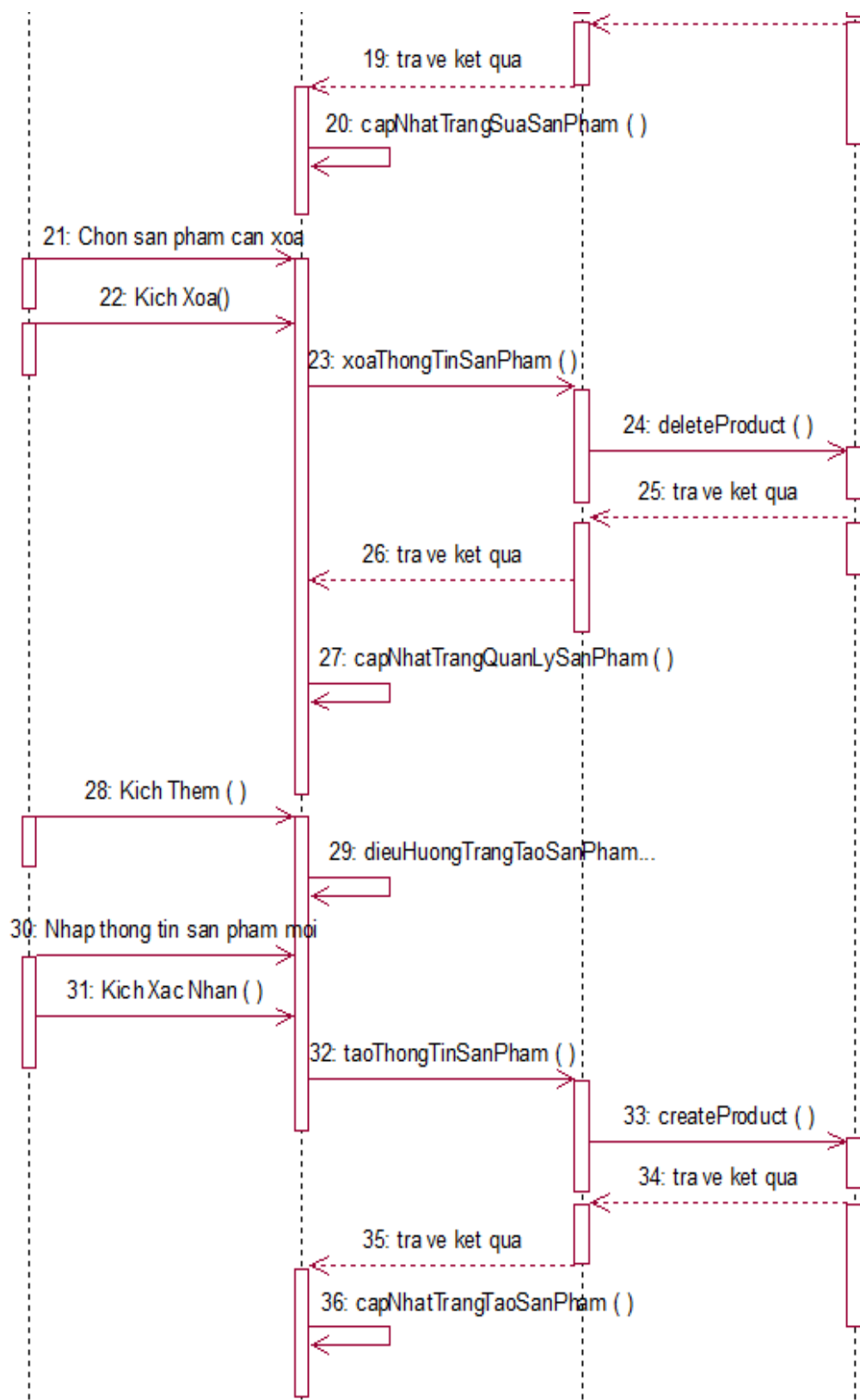


Hình 2.37: Biểu đồ trình tự use case Quản lý nhà xuất bản (2)

* Use case Quản lý sản phẩm

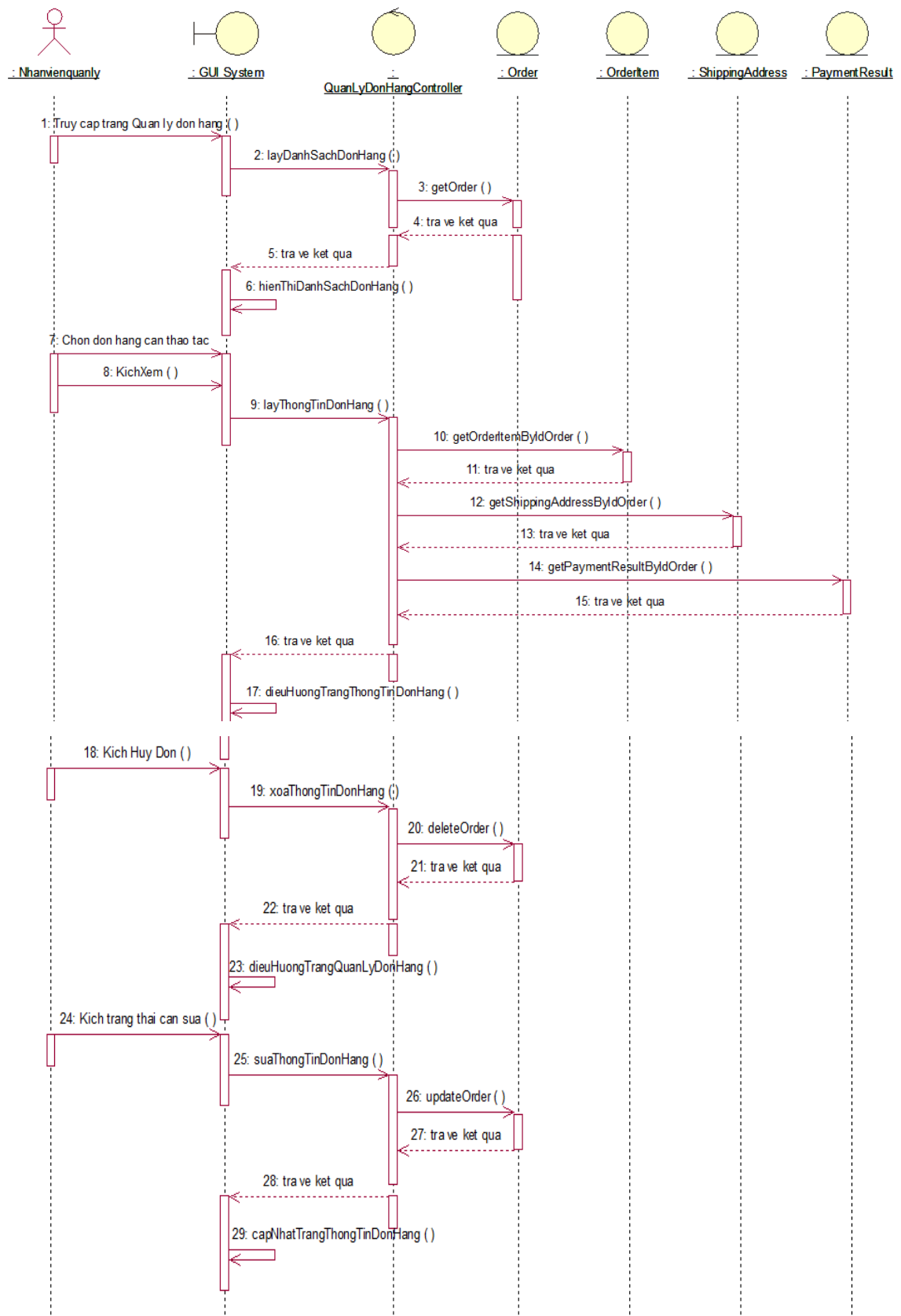


Hình 2.38: Biểu đồ trình tự use case Quản lý sản phẩm (1)



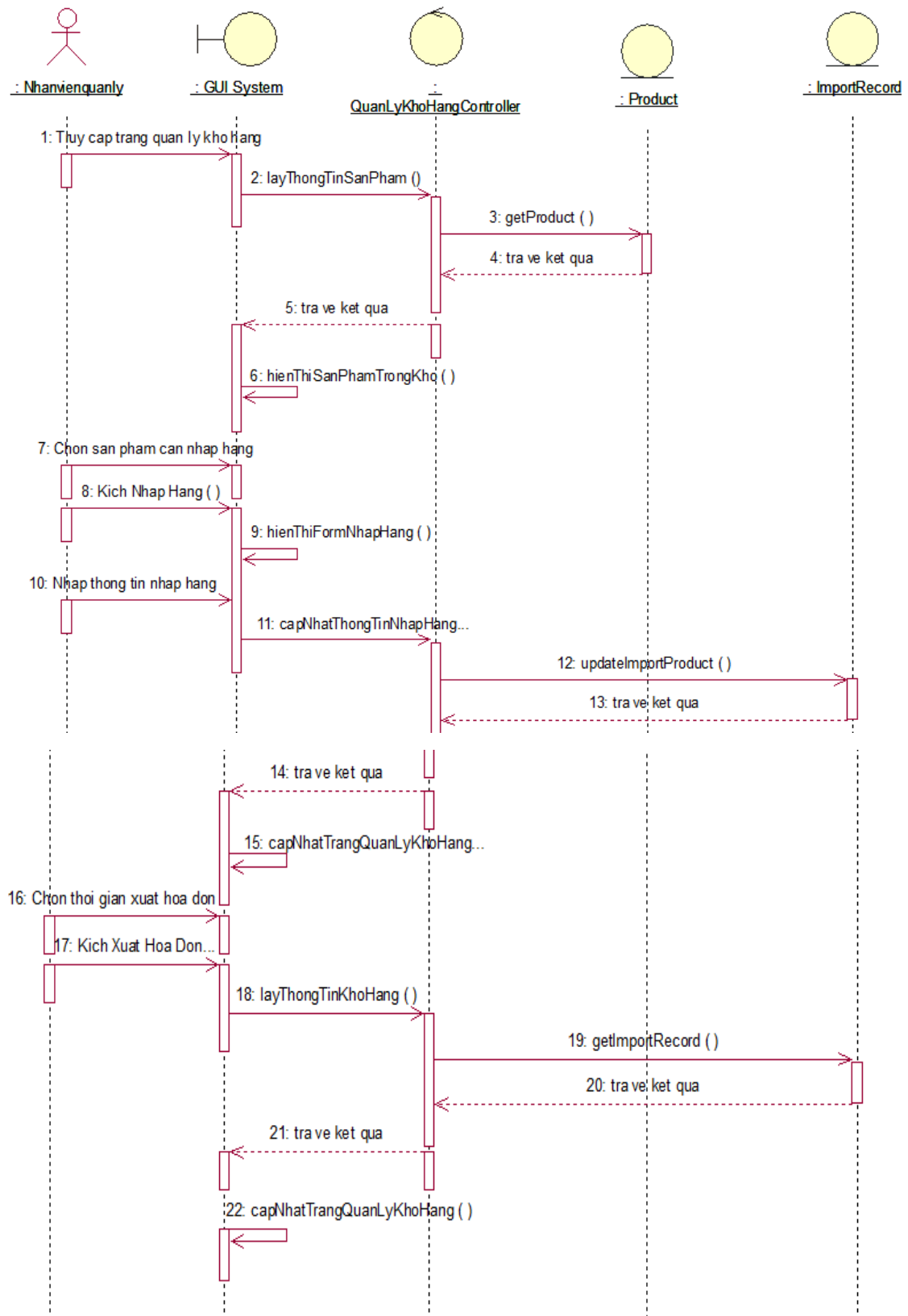
Hình 2.39: Biểu đồ trình tự use case Quản lý sản phẩm (2)

* Use case Quản lý đơn hàng



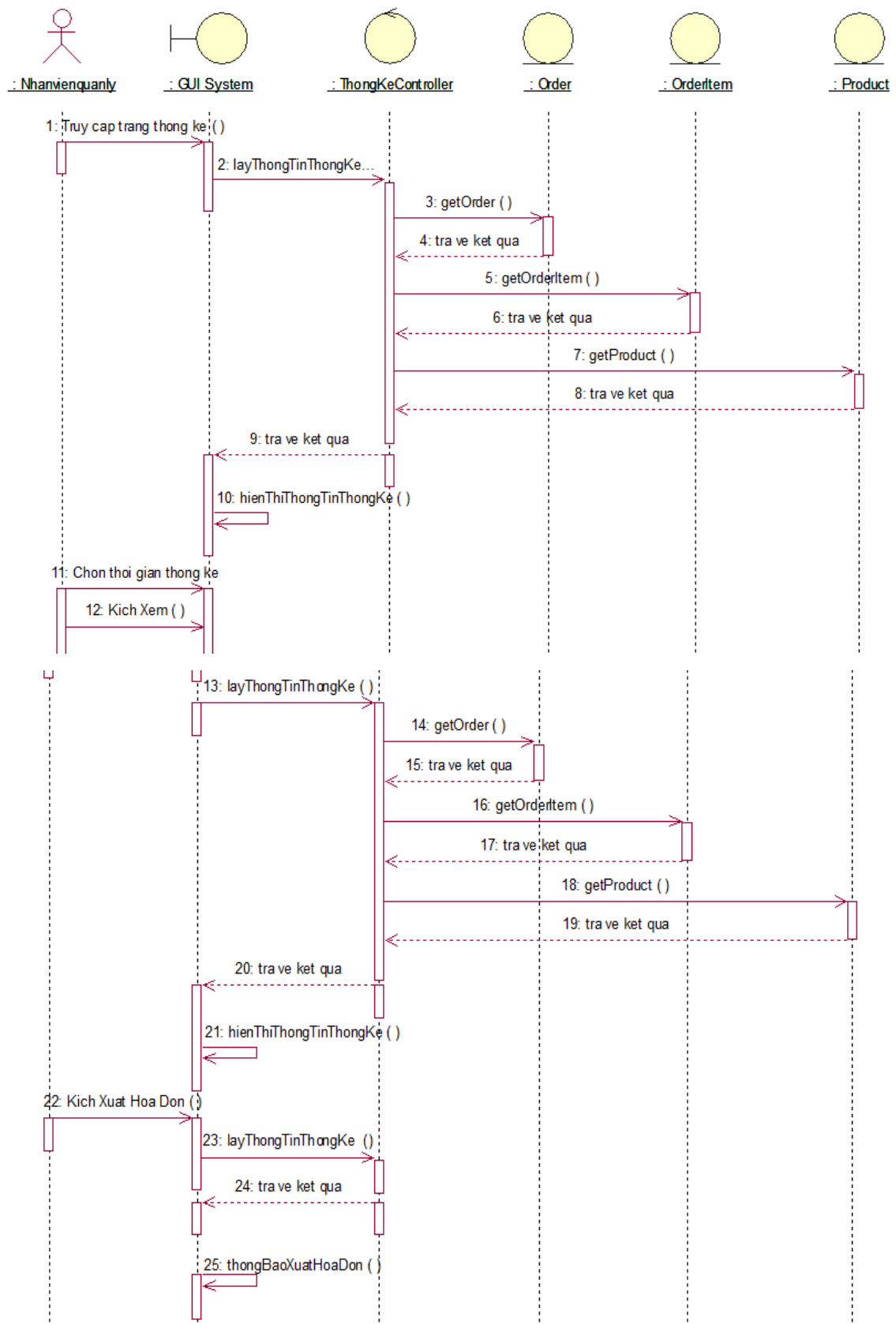
Hình 2.40: Biểu đồ trình tự use case Quản lý đơn hàng

* Use case Quản lý kho hàng



Hình 2.41: Biểu đồ trình tự use case Quản lý kho hàng

* Use case Thống kê



Hình 2.42: Biểu đồ trình tự use case Thống kê

2.4.6. Các yêu cầu về dữ liệu

Là một nền tảng thương mại điện tử chuyên về mua bán sách trực tuyến, hệ thống cần đảm bảo khả năng lưu trữ và quản lý dữ liệu một cách hiệu quả, chính xác và bảo mật. Dữ liệu của hệ thống không chỉ phục vụ cho việc hiển thị sản phẩm mà còn hỗ trợ quá trình đặt hàng, xử lý thanh toán, quản lý người dùng và phản hồi từ khách hàng.

* Thiết kế Collections

a) Quản lý thông tin người dùng (tài khoản đăng nhập)

Bảng 2.1: Collection User

| STT | Tên trường | Kiểu dữ liệu | Mặc định |
|-----|------------|--------------|-----------|
| 1 | name | String | NOT NULL |
| 2 | email | String | NOT NULL |
| 3 | password | String | NOT NULL |
| 4 | isAdmin | Boolean | NULL |
| 5 | birthday | Date | NULL |
| 6 | createdAt | Date | timestamp |
| 7 | updatedAt | Date | timestamp |

b) Quản lý thông tin sản phẩm

Bảng 2.2: Collection Product

| STT | Tên trường | Kiểu dữ liệu | Mặc định |
|-----|----------------|--------------|----------|
| 1 | name | String | NOT NULL |
| 2 | image | Mixed | NOT NULL |
| 3 | description | String | NOT NULL |
| 4 | author | String | NOT NULL |
| 5 | category | String | NOT NULL |
| 6 | price | Number | NOT NULL |
| 7 | countInStock | Number | NOT NULL |
| 8 | discount | Number | NULL |
| 9 | rating | Number | NULL |
| 10 | numReviews | Number | NULL |
| 11 | reviews | | NULL |
| | name | String | NOT NULL |

| | | | |
|----|-----------|----------|-----------|
| | Rating | Number | NOT NULL |
| | comment | String | NULL |
| | User | ObjectId | NOT NULL |
| 12 | createdAt | Date | timestamp |
| 13 | updatedAt | Date | timestamp |

c) Quản lý thông tin đơn hàng

Bảng 2.3: Collection Order

| STT | Tên trường | Kiểu dữ liệu | Mặc định |
|-----|------------------------|--------------|-----------|
| 1 | user | ObjectId | NOT NULL |
| 2 | orderItems | | NOT NULL |
| | name | String | NOT NULL |
| | qty | Number | NOT NULL |
| | image | Mixed | NOT NULL |
| | price | Number | NOT NULL |
| | discount | Number | NOT NULL |
| | discountedPrice | Number | NOT NULL |
| | product | ObjectId | NOT NULL |
| 3 | shippingAddress | ? | NOT NULL |
| | address | String | NOT NULL |
| | city | String | NOT NULL |
| | postalCode | String | NOT NULL |
| | country | String | NOT NULL |
| | phoneNumber | String | NOT NULL |
| 4 | paymentMethod | String | NOT NULL |
| 5 | paymentResult | ? | |
| | id | String | NOT NULL |
| | status | String | NOT NULL |
| | update_time | String | timestamp |
| | email_address: | String | NULL |
| 6 | taxPrice | Number | NOT NULL |
| 7 | shippingPrice | Number | NOT NULL |
| 8 | totalPrice | Number | NOT NULL |
| 9 | isPaid | Boolean | NOT NULL |
| 10 | paidAt | Date | timestamp |
| 11 | deliveryStatus | String | NOT NULL |
| 12 | deliveredAt | Date | timestamp |

| | | | |
|----|--------------|---------|-----------|
| 13 | isReturned | Boolean | |
| 14 | returnedAt | Date | timestamp |
| 15 | returnReason | String | |
| 16 | createdAt | Date | timestamp |
| 17 | updatedAt | Date | timestamp |

d) Quản lý thể loại

Bảng 2.4: Collection Category

| STT | Tên trường | Kiểu dữ liệu | Mặc định |
|-----|-------------|--------------|-----------|
| 1 | name | String | NOT NULL |
| 2 | description | String | NOT NULL |
| 3 | discount | Number | NULL |
| 4 | createdAt | Date | timestamp |
| 5 | updatedAt | Date | timestamp |

e) Quản lý tác giả

Bảng 2.5: Collection Author

| STT | Tên trường | Kiểu dữ liệu | Mặc định |
|-----|------------|--------------|-----------|
| 1 | name | String | NOT NULL |
| 2 | birthday | Number | NOT NULL |
| 3 | discount | Number | NULL |
| 4 | createdAt | Date | timestamp |
| 5 | updatedAt | Date | timestamp |

f) Quản lý nhà xuất bản

Bảng 2.6: Collection Publisher

| STT | Tên trường | Kiểu dữ liệu | Mặc định |
|-----|-------------|--------------|-----------|
| 1 | name | String | NOT NULL |
| 2 | description | String | NOT NULL |
| 3 | discount | Number | NULL |
| 4 | createdAt | Date | timestamp |
| 5 | updatedAt | Date | timestamp |

g) Quản lý kho hàng

Bảng 2.7: Collection StockReceipt

| STT | Tên trường | Kiểu dữ liệu | Mặc định |
|-----|------------|--------------|----------|
| 1 | product | ObjectId | NOT NULL |

| | | | |
|---|-------------|----------|-----------|
| 2 | quantity | Number | NOT NULL |
| 3 | importPrice | Number | NUT NULL |
| 4 | user | ObjectId | NOT NULL |
| 5 | createdAt | Date | timestamp |
| 6 | updatedAt | Date | timestamp |

h) Lịch sử đơn hàng

Bảng 2.8: Collection StatusHistory

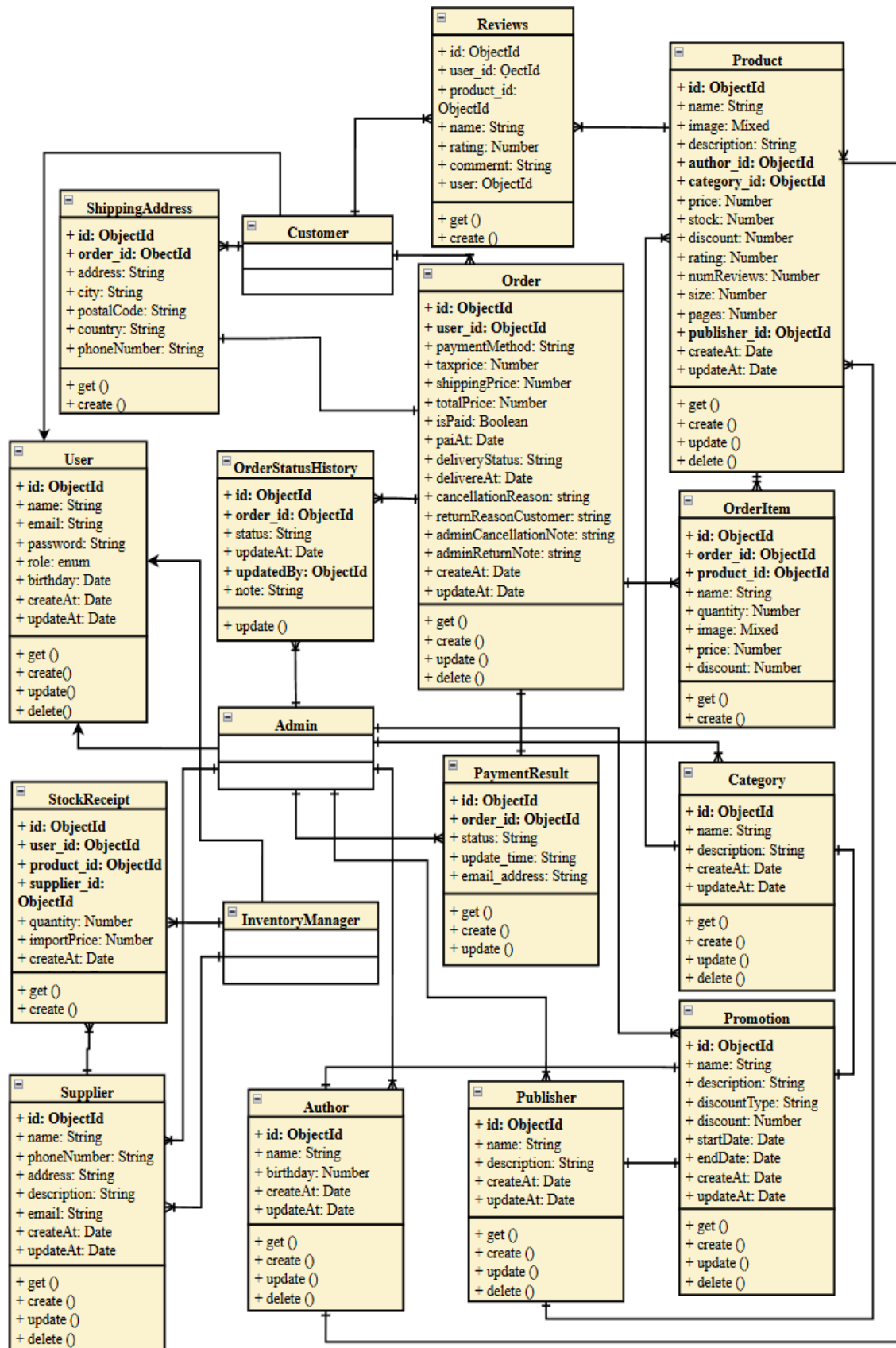
| STT | Tên trường | Kiểu dữ liệu | Mặc định |
|-----|-------------------|--------------|----------|
| 1 | _id | ObjectId | NOT NULL |
| 2 | order_id | ObjectId | NOT NULL |
| 3 | updatedBy_user_id | ObjectId | NOT NULL |
| 4 | status | String | NOT NULL |
| 5 | updatedAt | Date | NOT NULL |

i) Quản lý nhà cung cấp

Bảng 2.9: Collection Supplier

| STT | Tên trường | Kiểu dữ liệu | Mặc định |
|-----|-------------|--------------|-----------|
| 1 | _id | ObjectId | NOT NULL |
| 2 | name | String | NOT NULL |
| 3 | address | String | NUT NULL |
| 4 | phoneNumber | String | NOT NULL |
| 5 | email | String | NOT NULL |
| 6 | description | String | NOT NULL |
| 7 | createdAt | Date | timestamp |
| 8 | updatedAt | Date | timestamp |

2.4.7. Mô hình thực thể - quan hệ



Hình 2.43: Mô hình thực thể liên kết hệ thống

CHƯƠNG 3: CÀI ĐẶT VÀ KIỂM THỬ HỆ THỐNG

3.1. Cài đặt môi trường

3.1.1. Môi trường hoạt động

- Hệ điều hành: Windows 11 Pro 64-bit
- Ngôn ngữ lập trình: JavaScript, CSS
- Cơ sở dữ liệu: MongoDB Compass
- Framework: NodeJS 20.18.0, Express 4.18.1, React 18.2.0
- Công cụ phát triển:
 - Visual Studio Code với các extension hỗ trợ JavaScript, React và Node.js
 - Git để quản lý mã nguồn
 - Postman để kiểm thử API

3.1.2. Cài đặt môi trường phát triển

* Cài đặt môi trường trên cả backend và frontend:

`npm install`

* Chạy server:

| Bước | Chạy backend | Chạy frontend |
|------|------------------------|------------------------|
| 1 | <code>cd server</code> | <code>cd client</code> |
| 2 | <code>npm start</code> | <code>npm start</code> |

3.1.3. Giao diện hệ thống

* Giao diện đăng ký

BookStore [Trang chủ](#) [Sản phẩm](#) [Khuyến mãi](#) [Tài khoản](#) [Giỏ hàng](#)

ĐĂNG KÝ

Email

Họ và tên

Mật khẩu

Xác nhận mật khẩu

HOẶC

Hình 3.1: Giao diện đăng ký

*Giao diện đăng nhập

BookStore [Trang chủ](#) [Sản phẩm](#) [Khuyến mãi](#) [Tài khoản](#) [Giỏ hàng](#)

ĐĂNG NHẬP

Email...

Mật khẩu...

[Quên mật khẩu?](#)

Bạn chưa có tài khoản? [Đăng ký tại đây](#)

HOẶC

Hình 3.2: Giao diện đăng nhập

* Giao diện chi tiết sản phẩm

BookStore [Trang chủ](#) [Sản phẩm](#) [Khuyến mãi](#) [Nguyễn Văn A](#) [Giỏ hàng](#)

Khi Mọi Điều Không Như Ý

120.000 đ

Tác giả: Hae Min

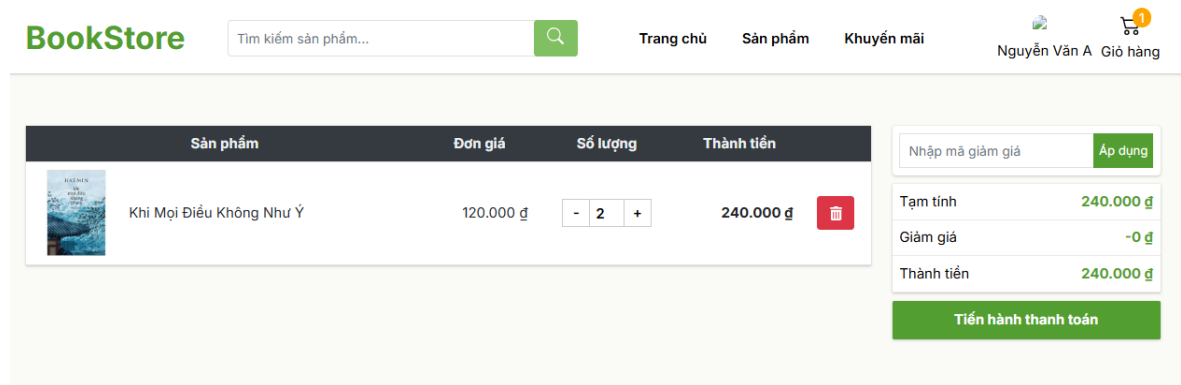
Nhà xuất bản: NXB Kim Đồng - 2025

Số lượng:

| Thông tin chi tiết | |
|--------------------|-----------------|
| Tác giả | Hae Min |
| NXB | NXB Kim Đồng |
| Năm xuất bản | 2025 |
| Kích thước | 20.5 x 14 x 1.3 |

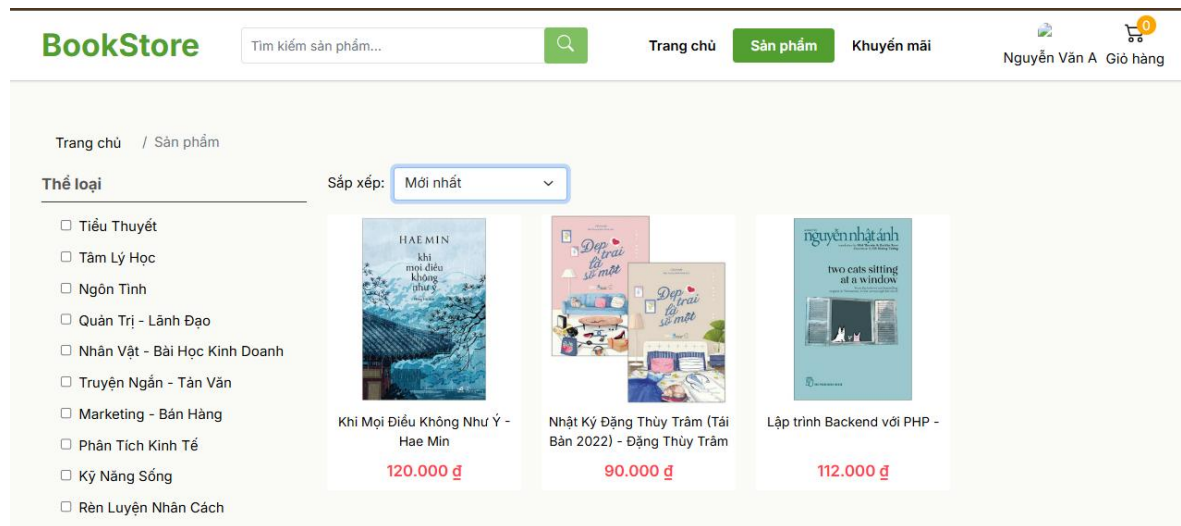
Hình 3.3: Giao diện chi tiết sản phẩm

* Giao diện giỏ hàng



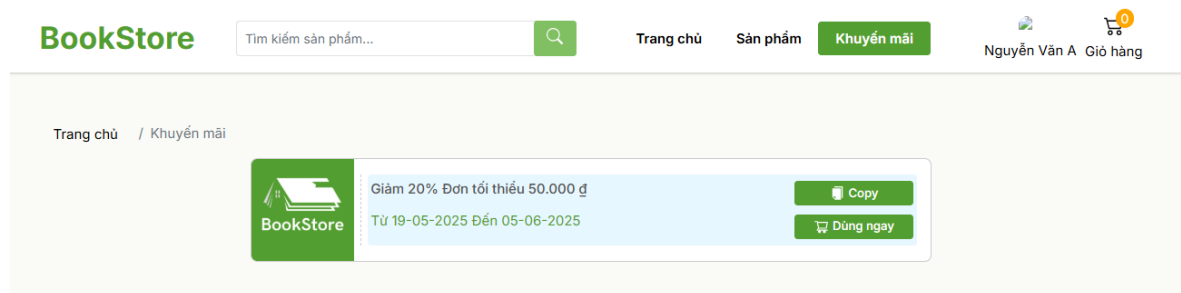
Hình 3.4: Giao diện giỏ hàng

* Giao diện bộ lọc sản phẩm



Hình 3.5: Giao diện bộ lọc sản phẩm

* Giao diện voucher khuyến mãi



Hình 3.6: Giao diện voucher khuyến mãi

* Giao diện thanh toán

BookStore

Tìm kiếm sản phẩm...

Trang chủ

Sản phẩm

Khuyến mãi

Nguyễn Văn A

Giỏ hàng

Trang chủ / Thanh toán

THÔNG TIN NHẬN HÀNG

Họ và tên

Nguyễn Văn A

Email

quangvnguyen1605@gmail.com

Số điện thoại

0349641705

Địa chỉ giao hàng: 35 Ngõ 2 Mễ Trì Hạ, Phường Mễ Trì, Quận Nam Từ Liêm, Thành phố Hà Nội

Thành phố Hà Nội

Quận Nam Từ Liêm

Phường Mễ Trì

35 Ngõ 2 Mễ Trì Hạ

Xác nhận địa chỉ

ĐƠN HÀNG CỦA BẠN

| SẢN PHẨM | TỔNG |
|-----------------------------|-----------|
| Khi Mọi Điều Không Như Ý x2 | 240.000 đ |
| Tạm tính | 240.000 đ |
| Giảm giá | -0 đ |
| Phí vận chuyển | +30.000 đ |
| Tổng | 270.000 đ |

DỊCH VỤ VẬN CHUYỂN

☒ Giao hàng tiêu chuẩn

☐ Giao hàng nhanh

☐ Giao hàng hỏa tốc

Thời gian giao hàng dự kiến: 24-05-2025

PHƯƠNG THỨC THANH TOÁN

☒ Trả tiền mặt khi nhận hàng

☐ Ví MoMo

☐ Paypal

ĐẶT HÀNG

Hình 3.7: Giao diện thanh toán

* Giao diện thông tin cá nhân

BookStore

Tìm kiếm sản phẩm...

Trang chủ

Sản phẩm

Khuyến mãi

Nguyễn Văn A

Giỏ hàng

Ảnh

lỗi

Nguyễn Văn A

Thông tin tài khoản

Đơn hàng

Địa chỉ

Email

nguyenquangvvvv@gmail.com

Họ và tên

Nguyễn Văn A

Số điện thoại

0341256987

Giới tính

☒ Nam

☐ Nữ

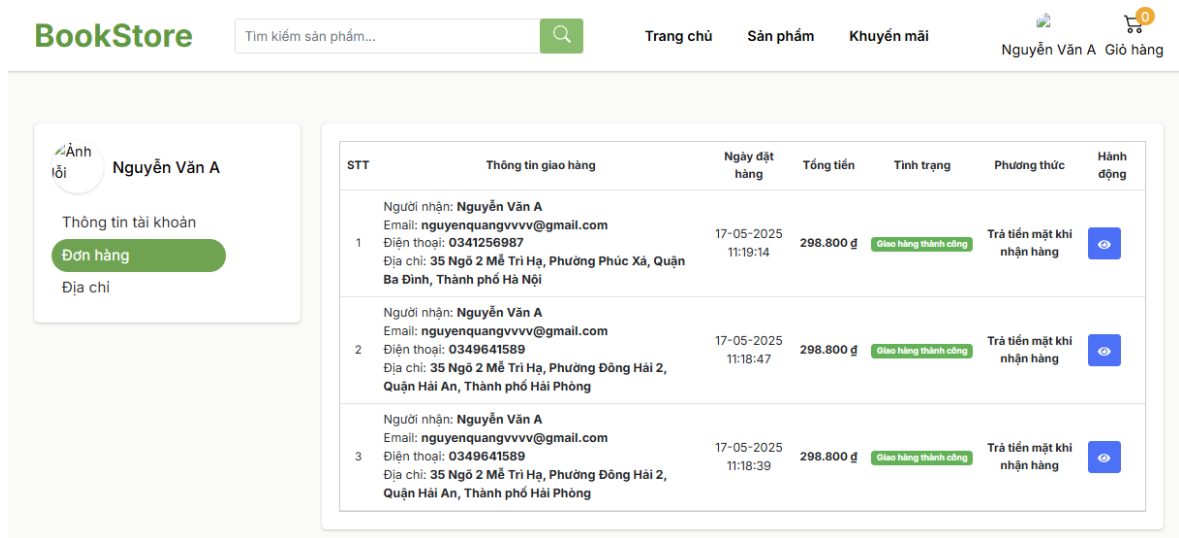
Ngày sinh

08/25/2006

Cập nhật

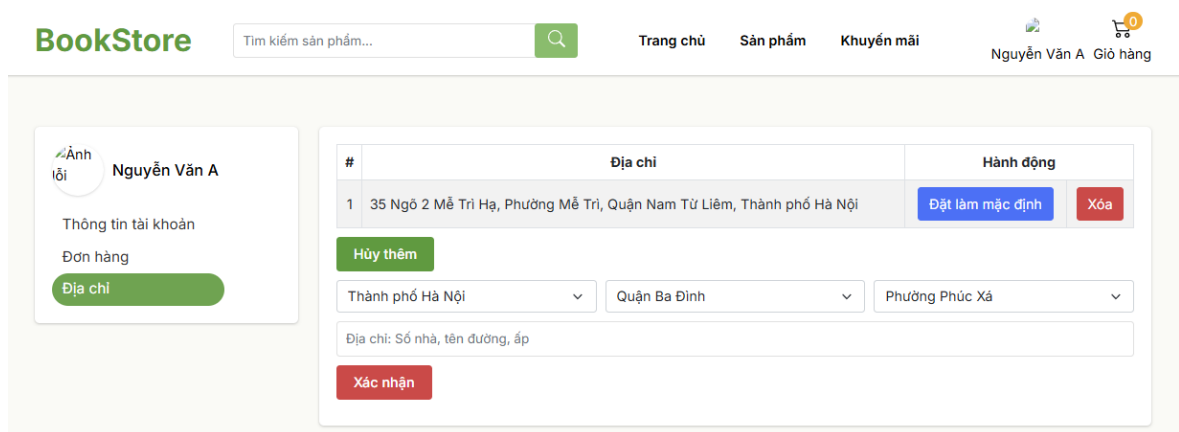
Hình 3.8: Giao diện thông tin cá nhân

* Giao diện đơn hàng cá nhân



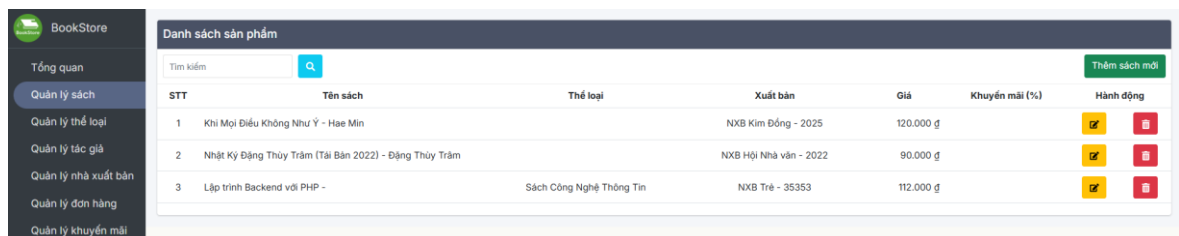
Hình 3.9: Giao diện đơn hàng cá nhân

* Giao diện địa chỉ nhận hàng



Hình 3.10: Giao diện địa chỉ nhận hàng

* Giao diện quản lý sản phẩm



Hình 3.11: Giao diện quản lý sản phẩm

*Giao diện quản lý kho hàng

| | | | | | |
|---|--|--------------------|------------------|--------------|----------------|
| <div>BookStore</div> <ul style="list-style-type: none"> Tổng quan Quản lý sách Quản lý thể loại Quản lý tác giả Quản lý nhà xuất bản Quản lý đơn hàng Quản lý khuyến mãi Mã giảm giá Khách hàng Nhà cung cấp Quản lý kho | Quản lý kho | | | | |
| | Tổng số sản phẩm | Tổng số hàng | Sản phẩm sắp hết | Hết hàng | |
| | 3 sản phẩm | 55 quyển | 1 sản phẩm | 1 sản phẩm | |
| | Tên sản phẩm | Số lượng trong kho | Giá niêm yết | Giá Nhập TB | Giá Khuyến Mãi |
| | Khí Mọi Điều Không Như Ý Hết hàng | 5 | 120,000 VND | — | 120,000 VND |
| | Nhật Ký Đặng Thùy Trâm (Tái Bản 2022) Sắp hết | 5 | 90,000 VND | 34,297.3 VND | 90,000 VND |
| | Lập trình Backend với PHP | 50 | 112,000 VND | 80,000 VND | 112,000 VND |
| | Lịch sử nhập kho | | | | |
| | Sách | Nhà cung cấp | Số lượng nhập | Giá nhập | Ngày nhập |
| | Nhật Ký Đặng Thùy Trâm (Tái Bản 2022) | vu | 5 | 50,000 VND | 5/17/2025 |
| | Nhật Ký Đặng Thùy Trâm (Tái Bản 2022) | vu | 3 | 50,000 VND | 5/17/2025 |
| | Nhật Ký Đặng Thùy Trâm (Tái Bản 2022) | vu | 2 | 45,000 VND | 5/17/2025 |

Hình 3.12: Giao diện quản lý kho hàng

* Giao diện quản lý người dùng

| | | | | | |
|---|------------------------------|--|---------------------|--------------------------|-------------------------------|
| <div>BookStore</div> <ul style="list-style-type: none"> Tổng quan Quản lý sách Quản lý thể loại Quản lý tác giả Quản lý nhà xuất bản Quản lý đơn hàng Quản lý khuyến mãi Mã giảm giá Khách hàng Nhà cung cấp Quản lý kho | Danh sách khách hàng | | | | |
| | Tìm kiếm bằng tên, email, SD | | | | |
| | STT | Khách hàng | Tài khoản | Trạng thái | Hành động |
| | 1 | Họ tên: Nguyễn Văn A Email: nguyenvanvv@gmail.com Điện thoại: 0341256987 | Tài khoản BookStore | Đã xác minh | Lịch sử mua hàng |
| | | | | | |

Hình 3.13: Giao diện quản lý người dùng

* Giao diện quản lý chương trình khuyến mãi

| | | | | | | |
|---|-----------------------------------|-------------------|----------------|---------|-------------------------|---|
| <div>BookStore</div> <ul style="list-style-type: none"> Tổng quan Quản lý sách Quản lý thể loại Quản lý tác giả Quản lý nhà xuất bản Quản lý đơn hàng Quản lý khuyến mãi Mã giảm giá Khách hàng Nhà cung cấp Quản lý kho | Danh sách chương trình khuyến mãi | | | | | |
| | Thêm chương trình | | | | | |
| | STT | Tên chương trình | Loại | Giá trị | Thời gian | Hành động |
| | 1 | quốc tế thiếu nhi | Giảm phần trăm | 50% | 2025-05-28 - 2025-06-03 | Tiểu Thuyết, Tâm Lý Học, Ngôn Tình, Quân Trĩ - Lãnh Đạo, Nhân Vật - Bài Học Kinh Doanh, Truyền Ngắn - Tân Văn, Marketing - Bán Hàng |
| | | | | | | |

Hình 3.14: Giao diện quản lý chương trình khuyến mãi

* Giao diện quản lý thể loại

BookStore

Tổng quan

Quản lý sách

Quản lý thể loại

Quản lý tác giả

Quản lý nhà xuất bản

Quản lý đơn hàng

Quản lý khuyến mãi

Mã giảm giá

Khách hàng

Nhà cung cấp

Quản lý kho

Danh sách thể loại

Thêm thể loại

| STT | Thể loại | Slug | Hành động |
|-----|-------------------------------|-----------------------------|-----------------------------------|
| 1 | Tiểu Thuyết | tiểu_thuyet | <div><div></div><div></div></div> |
| 2 | Tâm Lý Học | tam_ly_hoc | <div><div></div><div></div></div> |
| 3 | Ngôn Tình | ngon_tinh | <div><div></div><div></div></div> |
| 4 | Quản Trị - Lãnh Đạo | quan_tri_lanh_dao | <div><div></div><div></div></div> |
| 5 | Nhân Vật - Bài Học Kinh Doanh | nhan_vat_bai_hoc_kinh_doanh | <div><div></div><div></div></div> |
| 6 | Truyện Ngắn - Tân Văn | truyen_ngan_tan_van | <div><div></div><div></div></div> |
| 7 | Marketing - Bán Hàng | marketing_ban_hang | <div><div></div><div></div></div> |
| 8 | Phân Tích Kinh Tế | phan_tich_kinh_te | <div><div></div><div></div></div> |

Hình 3.15: Giao diện quản lý thể loại

* Giao diện quản lý đơn hàng

BookStore

Tổng quan

Quản lý sách

Quản lý thể loại

Quản lý tác giả

Quản lý nhà xuất bản

Quản lý đơn hàng

Quản lý khuyến mãi

Mã giảm giá

Khách hàng

Nhà cung cấp

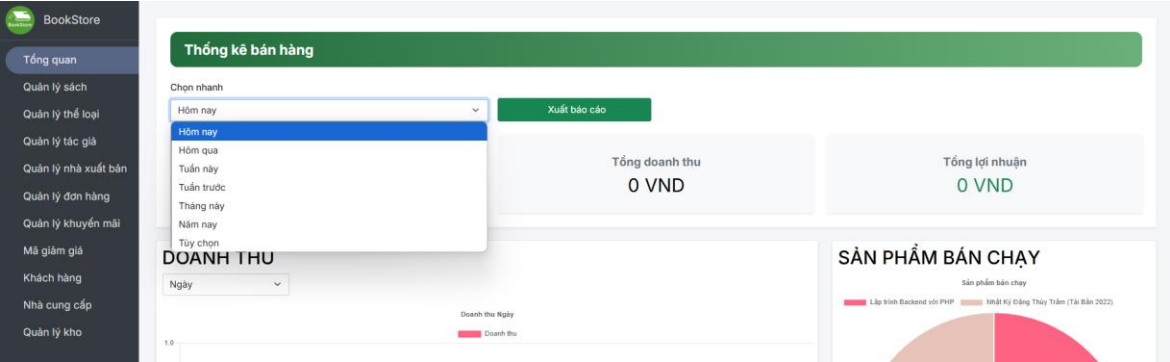
Quản lý kho

Danh sách đơn hàng

| STT | Thông tin giao hàng | Ngày đặt hàng | Tổng tiền | Tình trạng | Hành động |
|-----|--|---------------------|-----------|----------------------|-----------------------------------|
| 1 | Người nhận: Nguyễn Văn A Email: quangvuonguyen1605@gmail.com Điện thoại: 0349641705 Địa chỉ: aaaa, Xã Cẩm Sơn, Huyện Cẩm Xuyên, Tỉnh Hà Tĩnh | 18-05-2025 08:13:43 | 343.600 đ | Giao hàng thành công | <div><div></div><div></div></div> |
| 2 | Người nhận: Nguyễn Văn A Email: quangvuonguyen1605@gmail.com Điện thoại: 0349641705 Địa chỉ: aaaa, Xã Cẩm Sơn, Huyện Cẩm Xuyên, Tỉnh Hà Tĩnh | 17-05-2025 11:27:50 | 172.000 đ | Giao hàng thành công | <div><div></div><div></div></div> |
| 3 | Người nhận: Nguyễn Văn A Email: nguyenguangvvvv@gmail.com Điện thoại: 0341256987 Địa chỉ: 35 Ngõ 2 Mễ Trì Hạ, Phường Phúc Xá, Quận Ba Đình, Thành phố Hà Nội | 17-05-2025 11:19:14 | 298.800 đ | Giao hàng thành công | <div><div></div><div></div></div> |
| 4 | Người nhận: Nguyễn Văn A Email: nguyenguangvvvv@gmail.com Điện thoại: 0349641599 Địa chỉ: 35 Ngõ 2 Mễ Trì Hạ, Phường Đồng Hải 2, Quận Hải An, Thành phố Hải Phòng | 17-05-2025 11:18:47 | 298.800 đ | Giao hàng thành công | <div><div></div><div></div></div> |
| 5 | Người nhận: Nguyễn Văn A Email: nguyenguangvvvv@gmail.com Điện thoại: 0349641599 Địa chỉ: 35 Ngõ 2 Mễ Trì Hạ, Phường Đồng Hải 2, Quận Hải An, Thành phố Hải Phòng | 17-05-2025 11:18:39 | 298.800 đ | Giao hàng thành công | <div><div></div><div></div></div> |

Hình 3.16: Giao diện quản lý đơn hàng

* Giao diện thống kê



Hình 3.17: Giao diện thống kê

3.2. Kiểm thử hệ thống

3.2.1. Khái niệm

Kiểm thử phần mềm (Software Testing) là một quá trình có hệ thống nhằm phát hiện lỗi, đảm bảo phần mềm hoạt động chính xác, đầy đủ và tuân thủ theo các yêu cầu đã được đặt ra trong tài liệu đặc tả kỹ thuật hoặc từ phía khách hàng. Đây không chỉ là công cụ tìm lỗi, mà còn là phương pháp đánh giá chất lượng phần mềm một cách khách quan, giúp giảm thiểu rủi ro khi triển khai hệ thống vào môi trường thực tế.

Trong quy trình kiểm thử, một giai đoạn đặc biệt quan trọng là kiểm thử hệ thống (System Testing). Đây là bước kiểm tra tổng thể toàn bộ hệ thống sau khi đã hoàn tất việc tích hợp các thành phần, nhằm đánh giá phần mềm có đáp ứng đúng các yêu cầu chức năng và phi chức năng đã đề ra hay không. Phương pháp chủ yếu được sử dụng trong giai đoạn này là kiểm thử hộp đen (Black-box Testing) – nghĩa là kiểm thử dựa trên đầu vào và đầu ra mà không cần quan tâm đến mã nguồn hay cấu trúc nội bộ của phần mềm.

Đặc điểm của kiểm thử hệ thống:

- Áp dụng kiểm thử hộp đen: Tập trung kiểm tra đầu vào – đầu ra mà không xét đến cấu trúc nội bộ.
- Thực hiện sau kiểm thử tích hợp: Là giai đoạn tiếp theo sau khi các module đã được tích hợp thành một hệ thống hoàn chỉnh.
- Đánh giá tổng thể: Kiểm tra đầy đủ chức năng chính, phụ và các yếu tố phi chức năng như hiệu suất, bảo mật, khả năng sử dụng,...
- Góc nhìn người dùng cuối: Mô phỏng trải nghiệm thực tế từ phía người sử dụng để đảm bảo tính thân thiện và phù hợp.
- Kiểm tra sự đầy đủ và đúng đắn: Đảm bảo hệ thống thực sự thỏa mãn các yêu cầu đã được đề xuất ban đầu.
- Môi trường tương đương thực tế: Thường thực hiện trong môi trường gần giống hoặc tương đồng với môi trường triển khai thật.

3.2.2. Các lĩnh vực kiểm thử hệ thống

Kiểm thử hệ thống không chỉ giới hạn ở chức năng phần mềm mà còn bao gồm nhiều khía cạnh khác, cụ thể:

- **Hiệu suất:** Đảm bảo hệ thống hoạt động ổn định, đúng yêu cầu và không phát sinh lỗi trong điều kiện sử dụng thực tế.
- **Bảo mật:** Ngăn chặn truy cập trái phép, bảo vệ dữ liệu người dùng và hệ thống khỏi các rủi ro bảo mật.
- **Giao diện:** Kiểm tra sự đầy đủ, chính xác và ổn định của giao diện người dùng theo yêu cầu thiết kế.
- **Khả năng cài đặt:** Đảm bảo phần mềm có thể được triển khai vào môi trường thật mà không phát sinh lỗi hoặc khó khăn.
- **Kiểm thử tải và chịu tải (Load/Stress Testing):** Đánh giá khả năng hệ thống xử lý các tình huống tải nặng, đảm bảo không bị quá tải hoặc sập hệ thống.

3.2.3. Quy trình kiểm thử hệ thống

Để kiểm thử hệ thống đạt hiệu quả cao, quy trình cần được xây dựng và tuân thủ chặt chẽ. Một quy trình kiểm thử hệ thống hoàn chỉnh thường bao gồm các bước sau:

- **Lập kế hoạch kiểm thử (Test Plan):** Xác định phạm vi, mục tiêu, chiến lược kiểm thử, tiêu chí thành công/thất bại, lựa chọn công cụ và phân công nhân sự.
- **Thiết kế test case (Test Case):** Dựa trên tài liệu yêu cầu và các kịch bản sử dụng (use case), xây dựng các tình huống kiểm thử cụ thể bao gồm chức năng, giao diện và phi chức năng.
- **Chuẩn bị dữ liệu kiểm thử (Test Data):** Tạo hoặc lựa chọn dữ liệu đầu vào phù hợp cho các test case để đảm bảo việc kiểm thử phản ánh đúng các kịch bản sử dụng thực tế.
- **Thực thi kiểm thử (Test Execution):** Triển khai các test case trên hệ thống, ghi nhận kết quả và phát hiện lỗi (nếu có).

- **Báo cáo và xử lý lỗi (Defect Reporting & Fixing):** Ghi nhận chi tiết các lỗi phát hiện, chuyển đến nhóm phát triển để khắc phục và phối hợp xác minh sau sửa lỗi.
- **Kiểm thử lại (Retesting):** Sau khi lỗi được sửa, lặp lại các test case liên quan để đảm bảo hệ thống không còn tồn tại vấn đề.

3.2.4. Vai trò và ý nghĩa của kiểm thử hệ thống

Kiểm thử hệ thống đóng vai trò thiết yếu trong quá trình phát triển phần mềm, với các lợi ích nổi bật:

- **Nâng cao chất lượng sản phẩm:** Giúp phát hiện và loại bỏ lỗi, sự cố trước khi hệ thống được triển khai chính thức.
- **Đảm bảo phần mềm đúng yêu cầu:** Xác minh phần mềm thực hiện đúng các chức năng đã cam kết với người dùng.
- **Phát hiện lỗi sớm:** Giảm chi phí sửa lỗi bằng cách phát hiện vấn đề ở giai đoạn sớm.
- **Tăng khả năng tích hợp:** Kiểm tra các thành phần trong hệ thống có thể phối hợp và hoạt động trơn tru với nhau.
- **Tăng cường bảo mật:** Giúp phát hiện lỗ hổng, từ đó giảm nguy cơ bị khai thác.
- **Đảm bảo khả năng triển khai:** Xác minh rằng hệ thống có thể được vận hành ổn định trong môi trường thực tế.

3.3. Thiết kế các test case cho hệ thống

Use case: Đăng nhập

Bảng 3.1: Test case use case Đăng nhập

| STT | Mô tả | Dữ liệu đầu vào | Bước thực hiện | Kết quả mong đợi | Trạng thái |
|------|----------------------|-----------------------------|---------------------------------|---------------------------------|------------|
| TC01 | Đăng nhập thành công | Email hợp lệ, mật khẩu đúng | Nhập email và mật khẩu > Submit | Chuyển đến trang chủ người dùng | Đạt |

| | | | | | |
|------|-------------------------|----------------------------|---------------------------------|--|-----|
| TC02 | Sai mật khẩu | Email hợp lệ, mật khẩu sai | Nhập email và mật khẩu > Submit | Hiện thị thông báo “Mật khẩu không đúng” | Đạt |
| TC03 | Email không tồn tại | Email không tồn tại | Nhập email và mật khẩu > Submit | Hiện thị “Tài khoản không tồn tại” | Đạt |
| TC04 | Để trống trường dữ liệu | Trống email hoặc mật khẩu | Submit không điền dữ liệu | Hiện thị cảnh báo “Vui lòng điền đầy đủ thông tin” | Đạt |

Use case: Đăng ký

Bảng 3.2: Test case use case Đăng ký

| STT | Mô tả | Dữ liệu đầu vào | Bước thực hiện | Kết quả mong đợi | Trạng thái |
|------|------------------------------|----------------------------------|----------------------------|---|------------|
| TC01 | Đăng ký thành công | Tên, email, mật khẩu hợp lệ | Điền form đăng ký > Submit | Chuyển đến trang đăng nhập và hiện thị thông báo đăng ký thành công | Đạt |
| TC02 | Email đã tồn tại | Email đã có trong hệ thống | Điền form > Submit | Hiện thị thông báo “Email đã được đăng ký” | Đạt |
| TC03 | Mật khẩu không khớp xác nhận | Mật khẩu và xác nhận không trùng | Điền form > Submit | Hiện thị lỗi “Mật khẩu không khớp” | Đạt |
| TC04 | Trường thông tin thiếu | Bỏ trống một hoặc nhiều trường | Submit | Hiện thị lỗi bắt buộc điền | Đạt |

Use case: Quản lý tài khoản

Bảng 3.3: Test case use case Quản lý tài khoản

| STT | Mô tả | Dữ liệu đầu vào | Bước thực hiện | Kết quả mong đợi | Trạng thái |
|------|-------------------------|-----------------|----------------------------------|-------------------------------|------------|
| TC01 | Xem danh sách tài khoản | Quyền admin | Truy cập mục “Quản lý tài khoản” | Hiện thị danh sách người dùng | Đạt |

| | | | | | |
|------|--------------------------------|----------------------------|----------------------------------|---|-----|
| TC02 | 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 |
| TC03 | Xóa tài khoản người dùng | ID người dùng | Click “Xóa” | Hiện thị cảnh báo xác nhận > Tài khoản bị xóa | Đạt |
| TC04 | Không có quyền truy cập | Người dùng thường | Truy cập link quản lý tài khoản | Hiện thị lỗi “Không có quyền truy cập” | Đạt |

Use case: Quản lý sản phẩm

Bảng 3.4: Test case use case Quản lý sản phẩm

| STT | Mô tả | Dữ liệu đầu vào | Bước thực hiện | Kết quả mong đợi | Trạng thái |
|------|------------------------|--------------------------------|---------------------------------------|---------------------------------------|------------|
| TC01 | 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 |
| TC02 | 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 |
| TC03 | Xóa sản phẩm | ID sản phẩm | Click “Xóa” > Xác nhận | Sản phẩm bị loại khỏi danh sách | Đạt |
| TC04 | Tìm kiếm sản phẩm | Tên sản phẩm, từ khóa | Nhập từ khóa vào ô tìm kiếm | Hiện thị sản phẩm phù hợp | Đạt |

Use case: Quản lý đơn hàng

Bảng 3.5: Test case use case Quản lý đơn hàng

| STT | Mô tả | Dữ liệu đầu vào | Bước thực hiện | Kết quả mong đợi | Trạng thái |
|------|------------------------|----------------------------|-------------------------|-----------------------------|------------|
| TC01 | Xem danh sách đơn hàng | Quyền admin hoặc nhân viên | Truy cập mục “Đơn hàng” | Hiện thị danh sách đơn hàng | |

| | | | | | |
|------|------------------------------|-----------------------------|--------------------------------------|--|--|
| TC02 | Cập nhật trạng thái đơn hàng | ID đơn hàng, trạng thái mới | Chọn đơn hàng > Chính sửa trạng thái | Trạng thái được cập nhật | |
| TC03 | Xem chi tiết đơn hàng | ID đơn hàng | Click vào mã đơn | Hiển thị chi tiết sản phẩm và thông tin khách hàng | |
| TC04 | Tìm kiếm đơn hàng theo mã | Mã đơn hàng cụ thể | Nhập mã đơn vào ô tìm kiếm | Hiển thị đơn hàng tương ứng | |

KẾT LUẬN

Qua quá trình nghiên cứu, thiết kế và triển khai, đề tài "Xây dựng website bán sách trực tuyến sử dụng Node.js" đã đạt được những kết quả tích cực và hoàn thành các mục tiêu đề ra ban đầu. Website bán sách trực tuyến đã được phát triển với đầy đủ các chức năng cốt lõi của hệ thống thương mại điện tử. Người dùng có thể đăng ký, đăng nhập, tìm kiếm sách, xem thông tin chi tiết sản phẩm, quản lý giỏ hàng và thực hiện thanh toán. Hệ thống cũng cung cấp giao diện quản trị cho phép quản lý sản phẩm, đơn hàng và khách hàng hiệu quả. Việc sử dụng Node.js kết hợp Express.js đã tạo nên ứng dụng web có hiệu suất cao. MongoDB mang lại sự linh hoạt trong lưu trữ dữ liệu, JWT cung cấp cơ chế xác thực an toàn, trong khi React.js tạo giao diện người dùng hiện đại và thân thiện.

Hệ thống vẫn còn một số điểm cần cải thiện như tối ưu giao diện người dùng, bổ sung các biện pháp mã hóa dữ liệu và xác thực hai yếu tố (2FA). Hiệu suất cần được nâng cao thông qua tối ưu hóa truy vấn cơ sở dữ liệu và implement caching. Ngoài ra, cần tích hợp thêm các phương thức thanh toán đa dạng. Đề tài không chỉ mang ý nghĩa học thuật mà còn có giá trị thực tiễn cao. Việc ứng dụng thành công các công nghệ hiện đại như Node.js, React.js, MongoDB vào xây dựng hệ thống thương mại điện tử đã góp phần nâng cao hiểu biết và kỹ năng phát triển phần mềm. Đồng thời, sản phẩm tạo ra có thể được ứng dụng trong thực tế, đáp ứng nhu cầu mua bán sách trực tuyến của người dùng.

Website đã xây dựng được nền tảng vững chắc, có khả năng mở rộng và phát triển thành một hệ thống thương mại điện tử chuyên nghiệp trong tương lai. Điều này không chỉ góp phần vào việc số hóa hoạt động kinh doanh mà còn mở ra cơ hội ứng dụng công nghệ thông tin vào giải quyết các bài toán thực tế trong kỷ nguyên chuyển đổi số.

TÀI LIỆU THAM KHẢO

Tài liệu Tiếng Việt

- [1] Nguyễn Văn A, Trần Thị B (2024), “Phát triển website bán sách sử dụng Node.js và Express.js”, *Tạp chí Công nghệ Thông tin*, 15(3), Tr. 45-56.
- [2] Lê Minh Cường, Phạm Thanh Hải (2023), “Ứng dụng MongoDB trong phát triển hệ thống thương mại điện tử”, *Tạp chí Khoa học và Công nghệ*, no. 21, Tr. 67-74.
- [3] Hoàng Quang Huy (2022), *Giáo trình phát triển web với Node.js và Express.js*, Nhà xuất bản Khoa học Kỹ thuật.
- [4] Trần Văn Bình (2021), “Phân tích hiệu suất của Node.js so với các công nghệ backend truyền thống”, *Hội thảo Quốc gia về Công nghệ Phần mềm*, Tr. 112-124.
- [5] Nguyễn Quốc Hưng (2019), “Mô hình RESTful API trong phát triển ứng dụng web hiện đại”, *Tạp chí Khoa học Máy tính*, 7(1), Tr. 20-29.

Tài liệu Tiếng Anh

- [6] Brown, J., & Smith, A. (2022). "Building Scalable Web Applications with Node.js and MongoDB.", *Journal of Web Development*, 28(4), pp. 150-165.
- [7] Wilson, R., & Carter, M. (2021). "Security Best Practices in Express.js: Protecting Web Applications.", *International Journal of Software Engineering*, 17(2), pp. 89-105.
- [8] Martin, K. (2021). "A Comparative Study of SQL and NoSQL Databases in E-Commerce Applications.", *Computer Science Review*, 14(3), pp. 205-220.
- [9] Johnson, D., & Lee, P. (2020). "Optimizing Performance in Node.js Applications: A Deep Dive.", *Software Engineering Journal*, 12(5), pp. 310-325.
- [10] Anderson, T. (2019). "RESTful API Design: Best Practices and Implementation Strategies.", *Journal of Web Technologies*, 10(1), pp. 45-60.