**ĐẠI HỌC PHENIKAA**

**TRƯỜNG CÔNG NGHỆ THÔNG TIN PHENIKAA**

\*\*\*\*\*\*

A logo for a university

AI-generated content may be incorrect.

**PHÂN TÍCH VÀ THIẾT KẾ PHẦN MỀM**

**ĐỀ TÀI: “XÂY DỰNG ỨNG DỤNG BÁN HÀNG QUẦN ÁO THỜI TRANG TRÊN WEBSITE”**

(Nhóm 2)

|  |  |
| --- | --- |
| Sinh viên thực hiện: | Trần Văn Bun - 23010370 |
|  | Nguyễn Võ Quốc Đạt - 23010306 |
|  | Phan Thị Gia Hân - 23010840 |
|  | Phan Minh Trúc - 23010818 |
|  | Đỗ Thanh Tùng - 23010811 |
|  |  |
| Giảng viên hướng dẫn: | ThS.Vũ Quang Dũng  ThS.Nguyễn Xuân Quế |

Lớp: Phân tích và thiết kế phần mềm-1-1-25 (N05)

Hà Nội, ngày 24 tháng 10 năm 2025

**Lời cảm ơn**

Để hoàn thành được đề tài này, chúng em xin tỏ lòng biết ơn sâu sắc đến thầy Vũ Quang Dũng và thầy Nguyễn Xuân Quế đã tận tình hướng dẫn và tạo điều kiện thuận lợi cho chúng em trong suốt quá trình thực hiện đề tài. Chúng em xin chân thành cảm ơn quý thầy cô trong khoa đã tận tình chỉ dạy cho em tạo nền tảng kiến thức vững chắc để hoàn thành đề tài này.

Trong quá trình thực hiện đề tài không thể tránh khỏi những sai sót kính mong các quý thầy cô góp ý và chỉ dẫn để chúng em có thể hoàn thiện hơn.

Chúng em xin chân thành cảm ơn!

**Bảng Phân chia công việc từng thành viên trong nhóm**

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Họ và Tên | MSSV | Nhiệm vụ |
| 1 | Trần Văn Bun | 23010370 | Thiết kế và thực hiện phần xác thực khách hàng và phần doanh mục sản phẩm |
| 2 | Nguyễn Võ Quốc Đạt | 23010306 | Thực hiện và thiết kế chức năng đặt hàng |
| 3 | Phan Thị Gia Hân | 23010840 | Thiết kế các chức năng quản trị người dùng |
| 4 | Phan Minh Trúc | 23010818 | Thiết kế các chức năng thanh toán và sản phẩm |
| 5 | Đỗ Thanh Tùng | 23010811 | Thiết kế và thực hiện các chức năng của giỏ hàng |

*Mục lục*

[1 Đặt vấn đề 7](#_Toc212206951)

[1.1 Yêu cầu 7](#_Toc212206952)

[1.1.1 Mô tả vấn đề 7](#_Toc212206953)

[1.1.2 Giải pháp 7](#_Toc212206954)

[1.2 Thực thể 8](#_Toc212206955)

[1.2.1 Thực thể 8](#_Toc212206956)

[1.3 Thông số kỹ thuật bổ sung/Yêu cầu phi chức năng 8](#_Toc212206957)

[1.3.1 Tốc độ 8](#_Toc212206958)

[1.3.2 Độ tin cậy 8](#_Toc212206959)

[1.3.3 Hiệu năng 8](#_Toc212206960)

[1.3.4 Ràng buộc thiết kế 9](#_Toc212206961)

[1.3.5 Khả năng mở rộng 9](#_Toc212206962)

[1.3.6 Mô hình 9](#_Toc212206963)

[1.4 Mô hình hóa chức năng 9](#_Toc212206964)

[1.4.1 Các yêu cầu chức năng 9](#_Toc212206965)

[1.4.1.1 Quản lý sản phẩm 9](#_Toc212206966)

[1.4.1.2 Quản lý đơn hàng 9](#_Toc212206967)

[1.4.1.3 Quản lý khách hàng 10](#_Toc212206968)

[1.4.1.4 Quản lý khách hàng và phân quyền 10](#_Toc212206969)

[1.4.1.5 Quản lý thanh toán 10](#_Toc212206970)

[1.4.1.6 Tìm kiếm và xem sản phẩm 10](#_Toc212206971)

[1.4.1.7 Tích hợp giỏ hàng 10](#_Toc212206972)

[1.4.1.8 Bảo mật thông tin khách hàng 11](#_Toc212206973)

[● Nhân viên quản trị (admin) 11](#_Toc212206974)

[1.4.1.9 Quản lý sản phẩm 11](#_Toc212206975)

[1.4.1.10 Quản lý đơn hàng 11](#_Toc212206976)

[1.4.2 Sơ đồ Use-case 11](#_Toc212206977)

[1.4.2.1 Sơ đồ Use-case tổng quan 12](#_Toc212206978)

[1.4.2.2 Use-case đăng nhập 12](#_Toc212206979)

[1.4.2.3 Use-case đăng ký 13](#_Toc212206980)

[1.4.2.4 Use-case quản lý khách hàng 14](#_Toc212206981)

[1.4.2.5 Use-case quản lý sản phẩm 14](#_Toc212206982)

[1.4.2.6 Use-case xem đơn hàng: 15](#_Toc212206983)

[1.4.2.7 Use-case quản lý đơn hàng 15](#_Toc212206984)

[1.4.2.8 Use-case quản lý thanh toán 16](#_Toc212206985)

[1.5 Đặc tả các Use-case 16](#_Toc212206986)

[1.5.1 Đăng nhập tài khoản admin: 16](#_Toc212206987)

[1.5.2 Quản lý tài khoản: 16](#_Toc212206988)

[1.5.3 Đăng nhập tài khoản khách hàng: 17](#_Toc212206989)

[1.5.4 Đăng ký tài khoản khách hàng: 17](#_Toc212206990)

[1.5.5 Quản lý giỏ hàng: 17](#_Toc212206991)

[1.5.6 Quản lý thanh toán: 18](#_Toc212206992)

[1.5.7 Quản lý bán hàng: 18](#_Toc212206993)

[1.5.8 Quản lý đơn hàng: 19](#_Toc212206994)

[1.6 Sơ đồ hoạt động 19](#_Toc212206995)

[1.6.1 Hệ thống con quản lý việc đăng ký 19](#_Toc212206996)

[1.6.2 Hệ thống con Quản lý Sản phẩm: 20](#_Toc212206997)

[1.6.3 Hệ thống con Quản lý Giỏ hàng: 22](#_Toc212206998)

[1.6.4 Hệ thống con Thanh toán: 24](#_Toc212206999)

[1.6.5 Hệ thống con Quản lý Khách hàng: 26](#_Toc212207000)

[2 Thiết kế sơ đồ lớp 28](#_Toc212207001)

[2.1 Phân tích kiến trúc hệ thống 28](#_Toc212207002)

[2.1.2 Sơ đồ lớp: 28](#_Toc212207003)

[2.1.3Giải thích sơ đồ 29](#_Toc212207004)

[2.1.3.1. QuanTri (Quản trị viên) 29](#_Toc212207005)

[2.1.3.2. KhachHang 29](#_Toc212207006)

[2.1.3.3. DonHang 30](#_Toc212207007)

[2.1.3.4. TaiKhoan 30](#_Toc212207008)

[2.1.3.5. GioHang 30](#_Toc212207009)

[2.1.3.6. SanPham (Sản phẩm) 31](#_Toc212207010)

[2.1.3.7. ThanhToan (Thanh toán) 31](#_Toc212207011)

[2.1.3.8. DanhMuc (Danh Mục) 31](#_Toc212207012)

[3 Sơ đồ trình tự 32](#_Toc212207013)

[3.1 Sơ đồ trình tự quản lý sản phẩm 32](#_Toc212207014)

[3.1.1 Sơ đồ trình tự quản lý sản phẩm 32](#_Toc212207015)

[3.1.2 Giải thích biều đồ 34](#_Toc212207016)

[3.2 sơ đồ trình tự quản lý đăng nhập và khách hàng 34](#_Toc212207017)

[3.2.1 Sơ đồ tuần tự chức năng đăng nhập và khách hàng 34](#_Toc212207018)

[3.2.2 Giải thích sơ đồ 35](#_Toc212207019)

[3.3 Sơ đồ tuần tự giỏ hàng 36](#_Toc212207020)

[3.3.1 Sơ đồ tuần tự giỏ hàng 36](#_Toc212207021)

[3.3.2 Giải thích sơ đồ 38](#_Toc212207022)

[3.4 Sơ đồ quản lý đơn hàng 38](#_Toc212207023)

[3.4.1 Sơ đồ quản lý đơn hàng 38](#_Toc212207024)

[3.4.2 Giải thích sơ đồ 39](#_Toc212207025)

[3.5 Sơ đồ quản lý thanh toán 40](#_Toc212207026)

[3.5.1 Sơ đồ quản lý thanh toán 40](#_Toc212207027)

[3.6 Sơ đồ quản lý danh mục 42](#_Toc212207028)

[3.6.1 Sơ đồ quản lý danh mục 42](#_Toc212207029)

[3.6.2 Giải thích sơ đồ 43](#_Toc212207030)

[3.7 Thiết kế 44](#_Toc212207031)

[3.7.1 Thiết kế giao diện 44](#_Toc212207032)

[4 Thiết kế Hành vi và Trạng thái 44](#_Toc212207033)

[4.1 Sơ đồ trạng thái 44](#_Toc212207034)

[4.1.1 Sơ đồ trạng thái quản lý sản phẩm 44](#_Toc212207035)

[Sơ đồ trạng thái sản phẩm 45](#_Toc212207036)

[1. Trạng thái ban đầu 45](#_Toc212207037)

[2. Nhap (Nhập) 45](#_Toc212207038)

[3. DangHoatDong (Đang hoạt động) 46](#_Toc212207039)

[4. HetHang (Hết hàng) 46](#_Toc212207040)

[5. DaCapNhat (Đã cập nhật) 46](#_Toc212207041)

[6. DaXoa (Đã xóa) 46](#_Toc212207042)

[4.1.2 Sơ đồ trạng thái quản lý thanh toán 46](#_Toc212207043)

[Sơ đồ trạng thái đơn hàng 47](#_Toc212207044)

[1. Trạng thái ban đầu 47](#_Toc212207045)

[2. ChuaThanhToan (Chưa thanh toán) 47](#_Toc212207046)

[3. DangXuLy (Đang xử lý) 48](#_Toc212207047)

[4. ThanhToanThanhCong (Thanh toán thành công) 48](#_Toc212207048)

[5. DangGiao (Đang giao) 48](#_Toc212207049)

[6. ThanhToanThatBai (Thanh toán thất bại) 48](#_Toc212207050)

[7. DaHuy (Đã hủy) 48](#_Toc212207051)

[8. HoanThanh (Hoàn thành) 48](#_Toc212207052)

[4.1.3 Sơ đồ trạng thái quản lý đơn hàng 48](#_Toc212207053)

[Sơ đồ trạng thái đơn hàng 49](#_Toc212207054)

[1. Trạng thái ban đầu 49](#_Toc212207055)

[2. ChuaThanhToan (Chưa thanh toán) 49](#_Toc212207056)

[3. DangXuLy (Đang xử lý) 49](#_Toc212207057)

[4. DaThanhToan (Đã thanh toán) 50](#_Toc212207058)

[5. ThatBai (Thất bại) 50](#_Toc212207059)

[6. DaHuy (Đã hủy) 50](#_Toc212207060)

[7. HoanThanh (Hoàn thành) 50](#_Toc212207061)

[4.1.4 Sơ đồ trạng thái mua hàng 50](#_Toc212207062)

[Sơ đồ trạng thái giỏ hàng 50](#_Toc212207063)

[1. Rong (Rỗng) 50](#_Toc212207064)

[2. CoSanPham (Có sản phẩm) 51](#_Toc212207065)

[3. DaXoa (Đã xóa) 51](#_Toc212207066)

[4. DaCapNhat (Đã cập nhật) 51](#_Toc212207067)

[5. DaXoaHet (Đã xóa hết) 51](#_Toc212207068)

[5 Thiết kế hệ thống. 52](#_Toc212207069)

[5.1 Sơ đồ gói 52](#_Toc212207070)

[Mối quan hệ giữa các tầng 52](#_Toc212207071)

[1. Tầng UI → Tầng Controller 52](#_Toc212207072)

[2. Tầng Controller → Tầng Service 52](#_Toc212207073)

[3. Tầng Service → Tầng Database (MongoDB) 52](#_Toc212207074)

[6 Design Pattern 53](#_Toc212207075)

[6.1 Design Pattern 53](#_Toc212207076)

[7 Tổng kết và hướng phát triển 54](#_Toc212207077)

[7.1 Tổng kết 54](#_Toc212207078)

[7.2 Hướng phát triển 55](#_Toc212207079)

[7.3 Kết luận chung 55](#_Toc212207080)

# 1 Đặt vấn đề

## 1.1 Yêu cầu

### 1.1.1 Mô tả vấn đề

Việc mua sắm trực tuyến đã trở thành một lựa chọn an toàn và phổ biến trong những năm qua, vì đa số người tiêu dùng đã tìm thấy sự thuận tiện và dễ dàng để chọn lựa cửa hàng, sự thoải mái trong việc mua bán và thanh toán. Một trong những yếu tố hấp dẫn nhất về mua sắm trực tuyến, là trong các kỳ nghỉ lễ, nó làm giảm bớt việc phải xếp hàng chờ đợi để tìm kiếm những mặt yêu thích trong các cửa hàng.

Những ưu thế của việc mua sắm trực tuyến đã làm thay thế được những bất tiện trong cách mua bán truyền thống. Người tiêu dùng, khi mua sắm trực tuyến, họ có thể mua sản phẩm được lựa chọn nhanh chóng bằng cách thực hiện một số cú nhấp chuột, họ có thể chọn và mua những mặt hàng yêu thích từ thị trường vô tận và không giới hạn do internet cung cấp, họ không cần phải đi đến các cửa hàng, và không bị giới hạn bởi giờ mở cửa, họ có thể đặt hàng mọi lúc, mọi nơi. và hơn nữa việc giao hàng có thể được yêu cầu không chỉ đến nơi cư trú mà còn ở nơi làm việc, do đó người tiêu dùng có thể mua sản phẩm với điều kiện thuận lợi nhất (giá cả, chất lượng, giảm giá khác) phù hợp với nhu cầu cá nhân

Theo một cuộc khảo sát toàn cầu gần đây do Công ty Nielsen thực hiện, hơn 85% dân số trên thế giới đã sử dụng Internet để mua hàng, tăng 40% so với hai năm trước và hơn một nửa số khách hàng Internet là những người mua sắm trực tuyến thường xuyên.

### 1.1.2 Giải pháp

Website bán hàng online là một cửa hàng thương mại điện tử, nơi mà các sản phẩm có thể được chọn lựa và đặt mua dễ dàng thông qua Internet. Khách hàng có thể duyệt qua danh mục và chọn các sản phẩm quan tâm. Khách hàng có thể xem thông tin chi tiết của các mặt hàng, khách hàng cũng có thể so sánh các mặt hàng cùng loại của các nhà sản xuất khác nhau để chọn sản phẩm phù hợp nhất trước khi quyết định đặt hàng. Các sản phẩm đã chọn được đưa vào giỏ hàng. Tại thời điểm thanh toán, các mặt hàng trong giỏ hàng sẽ được trình bày dưới dạng đơn đặt hàng. Khi đó khách hàng phải cung cấp các thông tin để hoàn thành giao dịch.

Thông thường, khách hàng sẽ được yêu cầu điền hoặc chọn địa chỉ thanh toán, địa chỉ giao hàng, tùy chọn giao hàng và thông tin thanh toán như số thẻ tín dụng. Một thông báo e-mail được gửi đến khách hàng ngay sau khi thực hiện đặt hàng. Khách hàng cần phải có tài khoản trên website và phải đăng nhập khi thanh toán.

Khách hàng có thể thanh toán ngay khi đặt hàng hoặc sau 24h, nếu quá hạn thì hệ thống tự động hủy đơn hàng. Website cũng có chức năng cho phép khách hàng có thể gửi phàn hồi hoặc đánh giá sản phẩm.

Ngoài ra, hệ thống website có chức năng giúp các quản trị thường xuyên cập nhật những sản phẩm mới theo nhiều chủng loại, của các nhà sản xuất khác nhau, chức năng quản lý các đơn đặt hàng của khách hàng, thống kê doanh thu bán hàng.

Giao diện khách hàng của website bán hàng online được thiết kế đơn giản, thân thiện với khách hàng và dễ sử dụng. Các chức năng được đặt ở vị trí dễ tìm kiếm và dễ truy cập, thông tin được bảo mật cao.

## 1.2 Thực thể

### 1.2.1 Thực thể

|  |  |
| --- | --- |
| Thực thể | Mô tả |
| 1. Khách hàng | Người mua hàng, có thể xem, chọn sản phẩm, thêm vào giỏ hàng, thanh toán, và theo dõi đơn hàng. |
| 2. Quản trị viên (Admin) | Người quản lý hệ thống, có thể thêm, xóa, sửa sản phẩm, quản lý tài khoản quản trị và đảm bảo vận hành hệ thống hiệu quả. |
| 4. Thanh toán | Quy trình xử lý thanh toán đơn hàng của khách hàng. Khách hàng có thể thanh toán trực tuyến hoặc lựa chọn phương thức thanh toán khi nhận hàng. |

## 1.3 Thông số kỹ thuật bổ sung/Yêu cầu phi chức năng

### 1.3.1 Tốc độ

* Thời gian phản hồi: Hệ thống phải phản hồi các yêu cầu từ khách hàng trong vòng 2 giây cho các thao tác thông thường như xem báo cáo, kiểm tra tồn kho.
* Thời gian xử lý dữ liệu: Các quá trình tính toán phức tạp như tạo báo cáo tài chính hoặc dự đoán nhu cầu nguyên liệu không được kéo dài quá 5 giây.

### 1.3.2 Độ tin cậy

* Tỷ lệ sẵn sàng của hệ thống: Hệ thống phải đảm bảo hoạt động với tỷ lệ uptime tối thiểu là 99.5%, để đảm bảo nhà hàng có thể truy cập và sử dụng phần mềm liên tục.
* Khả năng sao lưu và phục hồi dữ liệu: Hệ thống phải cung cấp cơ chế sao lưu tự động hàng ngày và có khả năng phục hồi dữ liệu trong trường hợp xảy ra lỗi.

### 1.3.3 Hiệu năng

* Quản lý tài nguyên hệ thống: Phần mềm phải tối ưu hóa sử dụng CPU và bộ nhớ, hoạt động tốt trên các máy tính có cấu hình trung bình để không gây tốn tài nguyên.
* Khả năng mở rộng: Hệ thống phải có khả năng xử lý tốt ngay cả khi số lượng giao dịch hàng ngày hoặc dữ liệu hàng hóa tăng lên đáng kể.

### 1.3.4 Ràng buộc thiết kế

* Khả năng tương thích: Phần mềm cần tương thích với các hệ điều hành phổ biến như Windows, macOS, và Linux.
* Giao diện khách hàng: Phần mềm phải có giao diện đơn giản, dễ sử dụng, thân thiện với khách hàng không chuyên về công nghệ.
* Bảo mật: Phần mềm cần áp dụng các biện pháp bảo mật tiên tiến, bao gồm mã hóa dữ liệu và quản lý quyền truy cập khách hàng, nhằm bảo vệ dữ liệu tài chính và hàng hóa khỏi các truy cập trái phép.

### 1.3.5 Khả năng mở rộng

* Hệ thống có thể mở rộng khi nhu cầu sử dụng và lượt sử dụng tăng cao( có thể phát triển theo hướng microservice hoặc kiến trúc 3-tier,…)

### **1.3.6 Mô hình**

* Mô hình waterfall
  + Nhóm lựa chọn mô hình **Waterfall** vì đây là mô hình **tuần tự, dễ triển khai và phù hợp với quy mô dự án vừa và nhỏ,** đặc biệt trong môi trường học tập khi yêu cầu và phạm vi hệ thống đã được xác định rõ ràng từ đầu.
  + Mô hình Waterfall chia quá trình phát triển phần mềm thành nhiều giai đoạn nối tiếp nhau, mỗi giai đoạn phải **hoàn thành và được kiểm duyệt** trước khi chuyển sang giai đoạn tiếp theo. Việc này giúp nhóm đảm bảo tiến độ, kiểm soát chất lượng và tránh bỏ sót yêu cầu trong quá trình phát triển

## 1.4 Mô hình hóa chức năng

### 1.4.1 Các yêu cầu chức năng

#### 1.4.1.1 Quản lý sản phẩm

* Mô tả: Cho phép quản trị viên và quản trị thêm mới, xóa, sửa thông tin sản phẩm.
* Ví du:
  + Cho phép thay đổi hình ảnh, thêm mô tả hoặc cập nhật thông tin tồn kho khi có thay đổi.

#### 1.4.1.2 Quản lý đơn hàng

* Mô tả: Theo dõi và cập nhật trạng thái đơn hàng.
* Ví dụ:
  + Ghi nhận trạng thái của đơn hàng từ lúc đặt hàng cho đến khi hoàn tất giao hàng.
  + Cập nhật trạng thái đơn hàng (đang xử lý, đã giao hàng, đã hủy) và thông báo đến khách hàng qua hệ thống.

#### 1.4.1.3 Quản lý khách hàng

* Mô tả: Quản lý thông tin khách hàng và lịch sử mua sắm.
* Ví dụ:
  + Lưu trữ thông tin cá nhân của khách hàng như tên, địa chỉ, email, số điện thoại.
  + Cập nhật trạng thái đăng ký và theo dõi lịch sử mua hàng để cung cấp trải nghiệm cá nhân hóa.

#### 1.4.1.4 Quản lý khách hàng và phân quyền

* Mô tả: Tạo và quản lý tài khoản cho khách hàng với các vai trò khác nhau.
* Ví dụ:
  + Tạo tài khoản cho các vai trò như khách hàng, quản trị, và quản trị viên.
  + Phân quyền cho các chức năng, giới hạn quyền truy cập dựa trên vai trò để bảo vệ dữ liệu hệ thống.

#### 1.4.1.5 Quản lý thanh toán

* Mô tả: Xử lý thanh toán và tạo hóa đơn cho khách hàng.
* Ví dụ:
* Cho phép thanh toán bằng nhiều phương thức như chuyển khoản, ví điện tử hoặc thẻ tín dụng.
* Tạo hóa đơn chi tiết cho từng đơn hàng và lưu trữ hóa đơn trong hệ thống.

#### 1.4.1.6 Tìm kiếm và xem sản phẩm

* Mô tả: Cho phép khách hàng tìm kiếm sản phẩm và xem thông tin chi tiết.
* Ví dụ:
* Khách hàng có thể tìm kiếm sản phẩm theo tên.
* Cung cấp thông tin chi tiết về sản phẩm bao gồm mô tả, hình ảnh và đánh giá của khách hàng.

#### 1.4.1.7 Tích hợp giỏ hàng

* Mô tả: Cho phép khách hàng thêm sản phẩm vào giỏ hàng và điều chỉnh số lượng trước khi thanh toán.
* Ví dụ:
* Khách hàng có thể thêm nhiều sản phẩm vào giỏ hàng, xem giỏ hàng và điều chỉnh số lượng từng sản phẩm.
* Tự động cập nhật tổng chi phí khi thay đổi số lượng hoặc xóa sản phẩm.

#### 1.4.1.8 Bảo mật thông tin khách hàng

* Mô tả: Bảo vệ thông tin khách hàng và dữ liệu hệ thống khỏi các truy cập trái phép.
* Ví dụ:
* Sử dụng mã hóa để bảo vệ dữ liệu khách hàng.
* Hạn chế truy cập trái phép và yêu cầu đăng nhập trước khi truy cập vào các thông tin quan trọng.

#### Nhân viên quản trị (admin)

#### 1.4.1.9 Quản lý sản phẩm

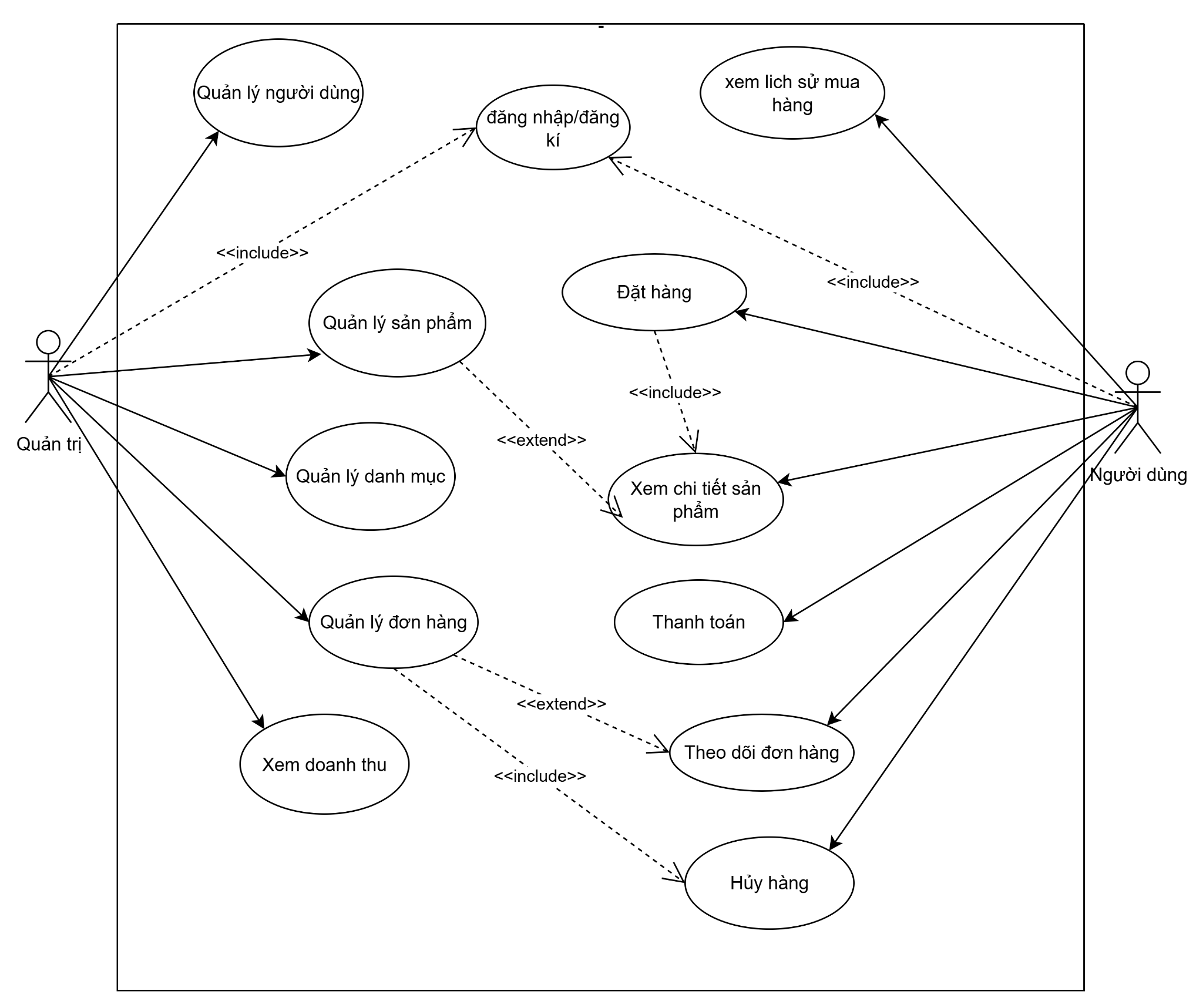
* Mô tả: Thêm, sửa, xóa sản phẩm. Cập nhật thông tin chi tiết và các giá cả sản phẩm.
* Ví dụ:
* Thêm sản phẩm mới hoặc cập nhật giá trị các sàn phẩm khi có thay đổi.
* Xóa các sản phẩm đã ngừng kinh doanh.

#### 1.4.1.10 Quản lý đơn hàng

* Mô tả: Xem và xác nhận các trạng thái đơn hàng đang trong quá trình xử lý(đang giao,đã đóng gói, đã giao hàng thành công).
* Ví dụ:
  + Cập nhật trạng thái đơn hảng.
  + Xác nhận đơn hàng và gửi đi cho khách hàng.

### 1.4.2 Sơ đồ Use-case

#### 1.4.2.1 Sơ đồ Use-case tổng quan



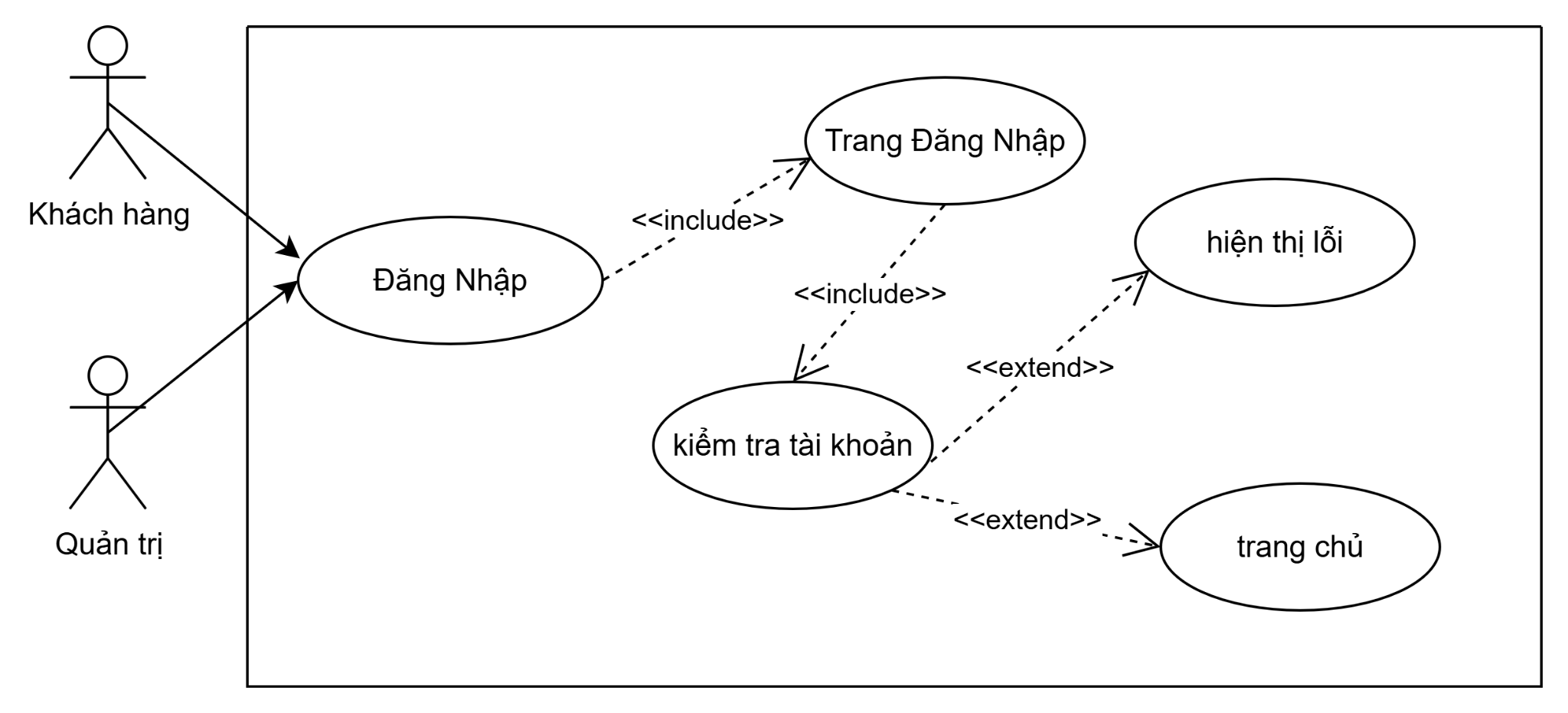
Hình 1. Sơ đồ Use-case tổng quan

Mô tả ngắn gọn về use case:

Các Nhân tố (Actors):

* Khách hàng: Người sử dụng chính của hệ thống, có khả năng duyệt và mua sản phẩm.
* Quản trị viên (Admin): Người quản lý toàn bộ hệ thống, có quyền cao nhất trong việc cài đặt và bảo trì.

#### 1.4.2.2 Use-case đăng nhập



Hình 2. Sơ đồ Use-case đăng nhập

1. Khách hàng chọn đăng nhập vào hệ thống

2. Hệ thống hiển thị màn hình đăng nhập

3. Hệ thống yêu cầu khách hàng nhập tên đăng nhập mât khẩu

4. Khách hàng nhập tên đăng nhập và mật khẩu

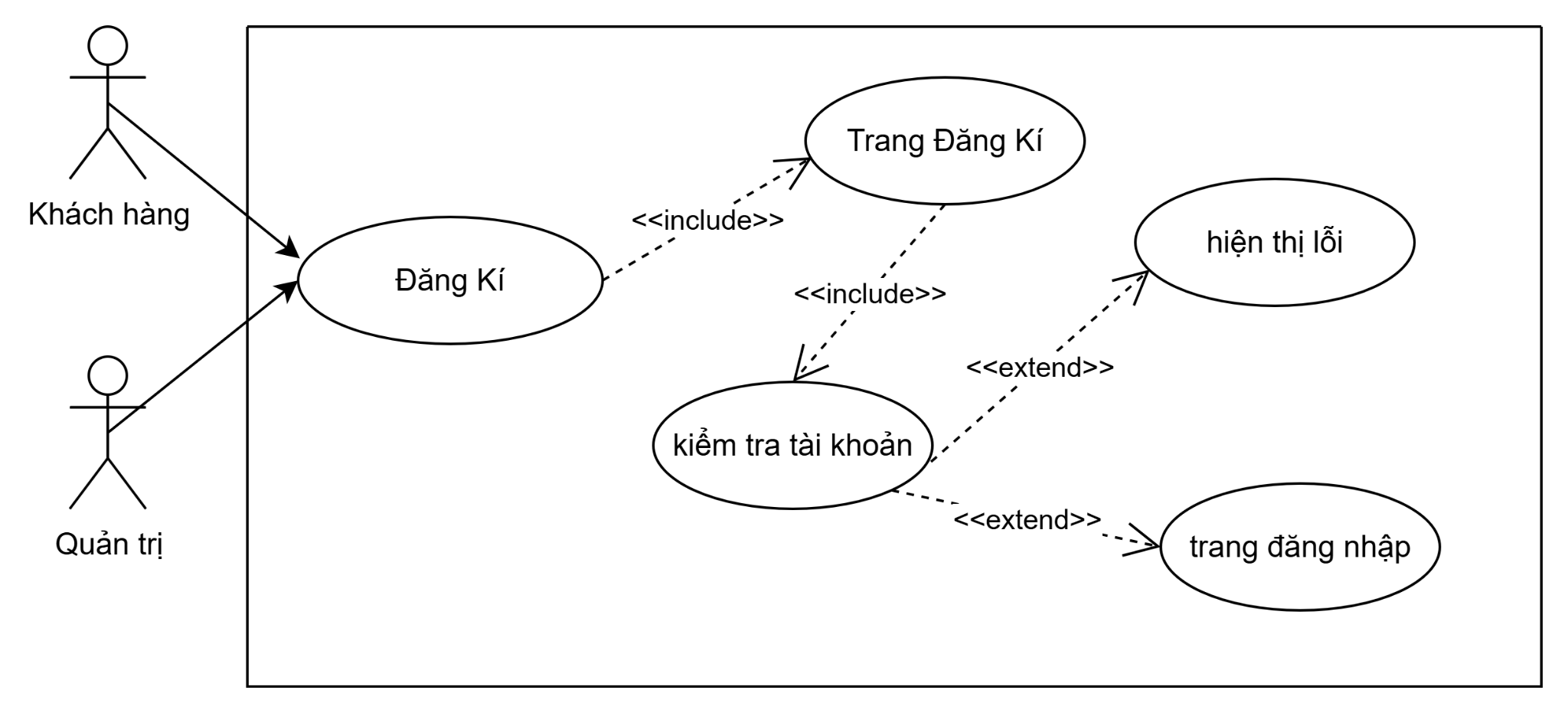
5. Hệ thống kiểm tra thông tin đăng nhập

6. Hệ thống lấy thông tin đăng nhập

7. Trả về kết quả đăng nhập

8. Hệ thống thông báo đăng nhập thành công

#### 1.4.2.3 Use-case đăng ký



Hình 3. Sơ đồ Use-case đăng ký

1. Load form giao diện chính

2. Khách hàng chọn chức năng “Đăng ký tài khoản”

3. Hệ thống hiển thị các thông tin yêu cầu để đăng ký : tên đăng nhập, mặt khẩu, email, ..các thông tin cá nhân liên quan khác

4. Khách hàng cung cấp các thông tin và tạo tài khoản

5. Hệ thống xác nhận thông tin và tạo tài khoản cho khách hàng

6. Lấy thông tin khách hàng

7. Hệ thống trả về kết quả

8. Hiển thị trang đăng nhập

#### 1.4.2.4 Use-case quản lý khách hàng

A diagram of a diagram

AI-generated content may be incorrect.

Hình 4. Sơ đồ Use-case quản lý khách hàng

1 Quản trị đăng nhập vào hệ thống.

2 Hệ thống xác thực thông tin tài khoản đăng nhập.

3 Hệ thống hiển thị màn hình chính theo vai trò của quản trị.

4 Quản trị chọn mục Quản lý khách hàng.

5 Hệ thống hiển thị giao diện quản lý khách hàng.

6 Quản trị có thể thực hiện các chức năng:

**7** Hệ thống cập nhật và hiển thị danh sách khách hàng sau mỗi thao tác.

#### 1.4.2.5 Use-case quản lý sản phẩm

A diagram of a diagram

AI-generated content may be incorrect.

Hình 5. Sơ đồ Use-case quản lý sản phẩm

#### 1.4.2.6 Use-case xem đơn hàng:

A diagram of a diagram

AI-generated content may be incorrect.

Hình 6. Sơ đồ Use-case xem đơn hàng

Khách hàng đăng nhập vào hệ thống.

Hệ thống xác thực tài khoản.

Hệ thống hiển thị màn hình chính theo vai trò khách hàng.

Khách hàng chọn mục Đơn hàng của tôi.

Hệ thống hiển thị danh sách các đơn hàng đã đặt.

#### 1.4.2.7 Use-case quản lý đơn hàng

A diagram of a diagram

AI-generated content may be incorrect.

Hình 7. Sơ đồ Use-case quản lý đơn hàng

1 Quản trị đăng nhập vào hệ thống.

2 Hệ thống xác thực thông tin tài khoản.

3 Hệ thống hiển thị màn hình chính theo vai trò của quản trị.

4 Quản trị chọn mục Quản lý đơn hàng.

5 Hệ thống hiển thị giao diện quản lý đơn hàng.

6 Quản trị có thể thực hiện các chức năng

7 Hệ thống lưu thông tin và hiển thị danh sách đơn hàng sau mỗi thao tác.

#### 1.4.2.8 Use-case quản lý thanh toán

A diagram of a diagram

AI-generated content may be incorrect.

Hình 8. Sơ đồ Use-case quản lý thanh toán

1. Khách hàng đăng nhập vào hệ thống

2. Hệ thống kiểm tra tên đăng nhập và mặt khẩu

3. Hệ thống thông báo đăng nhập thành công

4. Khách hàng chọn mục thanh toán

5. Hệ thống hiển thị màn hình chức năng thanh toán

6. Khách hàng chọn hóa đơn cần thanh toán(khi nhận hàng, VNpay)

7. Hệ thống xử lý yêu cầu, xác nhận việc thanh toán.

## 1.5 Đặc tả các Use-case

### 1.5.1 Đăng nhập tài khoản admin:

* Mô tả ngắn gọn: Quản trị đăng nhập vào hệ thống để quản lý thông tin
* Tác nhân thực hiện: Quản trị
* Luồng sự kiện:
* Luồng sự kiện chính:
* Quản trị truy cập trang đăng nhập.
* Nhập thông tin tài khoản (tên đăng nhập, mật khẩu).
* Hệ thống kiểm tra thông tin và đăng nhập thành công.
* Luồng phát sinh:
  + Nếu thông tin không đúng, hệ thống thông báo lỗi và yêu cầu nhập lại.
* Tiền điều kiện: Quản trị có tài khoản hợp lệ.
* Hậu điều kiện: Quản trị được chuyển đến trang quản trị hệ thống.

### 1.5.2 Quản lý tài khoản:

* Mô tả ngắn gọn: Quản trị thực hiện quản lý các tài khoản khách hàng.
* Tác nhân thực hiện: Quản trị
* Luồng sự kiện:
  + Luồng sự kiện chính:
* Quản trị truy cập vào trang quản lý tài khoản.
* Chọn thêm, sửa hoặc xóa tài khoản khách hàng.
* Hệ thống thực hiện yêu cầu và cập nhật thông tin.
* Luồng phát sinh:
  + Nếu có lỗi, hệ thống thông báo cho Quản trị .
* Tiền điều kiện: Quản trị đã đăng nhập.
* Hậu điều kiện: Tài khoản khách hàng được cập nhật

### 1.5.3 Đăng nhập tài khoản khách hàng:

* Mô tả ngắn gọn: Khách hàng đăng nhập để mua hàng và quản lý đơn hàng.
* Tác nhân thực hiện: Khách hàng
* Luồng sự kiện:
  + Luồng sự kiện chính:
* Khách hàng truy cập trang đăng nhập.
* Nhập tên đăng nhập và mật khẩu.
* Hệ thống kiểm tra và đăng nhập thành công.
* Luồng phát sinh:
  + Nếu thông tin không chính xác, hệ thống yêu cầu nhập lại.
* Tiền điều kiện: Khách hàng có tài khoản hợp lệ.
* Hậu điều kiện: Khách hàng được chuyển đến trang cá nhân.

### 1.5.4 Đăng ký tài khoản khách hàng:

* Mô tả ngắn gọn: Khách hàng đăng ký tài khoản để sử dụng các dịch vụ
* Tác nhân thực hiện: Khách hàng
* Luồng sự kiện:
  + Luồng sự kiện chính:
* Khách hàng truy cập trang đăng ký.
* Nhập thông tin yêu cầu (tên, email, mật khẩu, v.v.).
* Hệ thống kiểm tra và tạo tài khoản thành công.
* Luồng phát sinh:
  + Nếu thông tin không hợp lệ, hệ thống thông báo lỗi.
* Tiền điều kiện: Không có.
* Hậu điều kiện: Khách hàng có tài khoản mới để đăng nhập.

### 1.5.5 Quản lý giỏ hàng:

* Mô tả ngắn gọn: Khách hàng thêm hoặc xóa sản phẩm trong giỏ hàng
* Tác nhân thực hiện: Khách hàng
* Luồng sự kiện:
  + Luồng sự kiện chính:
* Khách hàng thêm sản phẩm vào giỏ.
* Khách hàng xóa sản phẩm khỏi giỏ.
* Hệ thống cập nhật giỏ hàng.
* Luồng phát sinh:
  + Nếu sản phẩm không có sẵn, hệ thống thông báo lỗi.
* Tiền điều kiện: Khách hàng đã đăng nhập.
* Hậu điều kiện: Giỏ hàng của khách hàng được cập nhật.

### 1.5.6 Quản lý thanh toán:

* Mô tả ngắn gọn: Khách hàng thực hiện thanh toán cho giỏ hàng
* Tác nhân thực hiện: Khách hàng
* Luồng sự kiện:
  + Luồng sự kiện chính:
* Khách hàng truy cập trang thanh toán
* Chọn phương thức thanh toán
* Hệ thống xử lý thanh toán và xác nhận đơn hàng
* Luồng phát sinh:
  + Nếu thanh toán thất bại, hệ thống yêu cầu khách hàng thử lại
* Tiền điều kiện: Giỏ hàng không trống
* Hậu điều kiện: Đơn hàng được xác nhận, giỏ hàng trống

### 1.5.7 Quản lý bán hàng:

* Mô tả ngắn gọn: quản trị quản lý thông tin bán hàng
* Tác nhân thực hiện: Quản trị
* Luồng sự kiện:
  + Luồng sự kiện chính:
* Quản trị truy cập vào trang quản lý bán hàng
* Chọn chức năng cần thực hiện (thêm/sửa/xóa sản phẩm)
  + Hệ thống thực hiện cập nhật thông tin sản phẩm
* Luồng phát sinh:
  + Nếu lỗi xảy ra, hệ thống thông báo lỗi
* Tiền điều kiện: Quản trị đã đăng nhập
* Hậu điều kiện: Thông tin sản phẩm được cập nhật

### 1.5.8 Quản lý đơn hàng:

* Mô tả ngắn gọn: Quản trị quản lý các đơn hàng của khách hàng
* Tác nhân thực hiện: Quản trị
* Luồng sự kiện:
  + Luồng sự kiện chính:
* Quản trị truy cập vào trang quản lý đơn hàng
* Chọn đơn hàng cần xử lý (xác nhận, giao hàng)
  + Hệ thống cập nhật trạng thái đơn hàng
* Luồng phát sinh:
  + Nếu lỗi xảy ra, hệ thống thông báo lỗi
* Tiền điều kiện: Quản trị đã đăng nhập
* Hậu điều kiện: Trạng thái đơn hàng được cập nhật

## 1.6 Sơ đồ hoạt động

### 1.6.1 Hệ thống con quản lý việc đăng ký

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Nội dung do AI tạo ra có thể không chính xác.

Hình 1. Sơ đồ hệ thống đăng ký

* Chức năng
  + Rà soát: Kiểm tra các thông tin được người dùng nhập vào và kiểm tra email người dùng sử dụng đã tồn tại trong hệ thống chưa.
  + Tạo người dùng mới: Nếu các điều kiện đã thỏa mãn thì sẽ ghi thông tin người dùng vào trong hệ thống, cho phép người dùng sử dụng tài khoản đăng nhập vào trang web.
* Giao diện người dùng:
  + Trang đăng kí: Hiện thị đầy đủ các ô thông tin để người dùng nhập thông tin vào và tiến hành đăng kí tài khoản.

### 1.6.2 Hệ thống con Quản lý Sản phẩm:

A screenshot of a cell phone

AI-generated content may be incorrect.Hình 9. Sơ đồ hệ thống con quản lý sản phẩm.

* Chức năng:
  + Thêm sản phẩm: Cho phép quản trị viên hoặc quản trị thêm sản phẩm mới vào hệ thống, bao gồm các thông tin như tên sản phẩm, mô tả, giá cả, số lượng và hình ảnh.
  + Chỉnh sửa sản phẩm: Cung cấp tính năng chỉnh sửa thông tin của sản phẩm hiện có, bao gồm cập nhật giá cả, số lượng, và mô tả.
  + Xóa sản phẩm: Cho phép xóa một sản phẩm khỏi hệ thống nếu không còn kinh doanh.
  + Xem danh sách sản phẩm: Khách hàng có thể xem danh sách sản phẩm với các thông tin cơ bản như tên, giá, hình ảnh và mô tả ngắn gọn.
  + Tìm kiếm: Cho phép khách hàng tìm kiếm sản phẩm theo từ khóa
* Giao diện khách hàng:
  + Trang danh sách sản phẩm: Hiển thị tất cả các sản phẩm dưới dạng danh sách hoặc lưới với các tùy chọn lọc và sắp xếp.
  + Trang chi tiết sản phẩm: Hiển thị chi tiết về từng sản phẩm, bao gồm tên, mô tả, hình ảnh, giá cả, và tùy chọn thêm vào giỏ hàng.
  + Trang quản trị sản phẩm: Chỉ dành cho quản trị viên, bao gồm các chức năng như thêm, chỉnh sửa, và xóa sản phẩm.

### 1.6.3 Hệ thống con Quản lý Giỏ hàng:

A screenshot of a phone

AI-generated content may be incorrect. Hình 10. Sơ đồ hệ thống con quản lý giỏ hàng.

* Chức năng:
  + Thêm vào giỏ hàng: Cho phép khách hàng thêm sản phẩm vào giỏ hàng với số lượng mong muốn.
  + Xem giỏ hàng: Khách hàng có thể xem danh sách các sản phẩm trong giỏ hàng, bao gồm tên sản phẩm, số lượng, giá từng sản phẩm và tổng tiền.
  + Cập nhật giỏ hàng: Cho phép khách hàng thay đổi số lượng hoặc xóa sản phẩm khỏi giỏ hàng.
  + Tính tổng tiền: Tự động tính tổng tiền giỏ hàng dựa trên số lượng và giá của từng sản phẩm.
  + Lưu trạng thái giỏ hàng: Đảm bảo giỏ hàng được lưu trữ cho khách hàng ngay cả khi họ rời khỏi trang và quay lại sau đó.
* Giao diện khách hàng:
  + Trang giỏ hàng: Hiển thị danh sách các sản phẩm trong giỏ hàng của khách hàng với các tùy chọn cập nhật số lượng hoặc xóa sản phẩm. Bao gồm nút tiến hành thanh toán để chuyển tới trang thanh toán.
  + Biểu tượng giỏ hàng: Luôn hiển thị biểu tượng giỏ hàng trên thanh điều hướng để khách hàng biết số lượng sản phẩm hiện có trong giỏ hàng.

### 1.6.4 Hệ thống con Thanh toán:

A screenshot of a black screen

AI-generated content may be incorrect.

Hình 10. Sơ đồ hệ thống con thanh toán.

* Chức năng:
  + Chọn phương thức thanh toán: Cung cấp cho khách hàng nhiều lựa chọn thanh toán như tiền mặt, ví điện tử.
  + Nhập thông tin thanh toán: Khách hàng nhập thông tin cần thiết cho phương thức thanh toán đã chọn (thông tin giao hàng).
  + Xác nhận và hoàn tất đơn hàng: Khách hàng xem lại thông tin giỏ hàng và xác nhận trước khi thanh toán.
  + Xử lý giao dịch thanh toán: Hệ thống xử lý thanh toán, bao gồm kiểm tra số dư và thực hiện giao dịch qua cổng thanh toán.
  + Gửi thông báo: Thông báo cho khách hàng khi giao dịch hoàn tất thành công hoặc thất bại.
* Giao diện khách hàng:
  + Trang thanh toán: Hiển thị thông tin chi tiết về giỏ hàng, các phương thức thanh toán có sẵn, và các trường thông tin thanh toán.
  + Trang xác nhận thanh toán: Sau khi thanh toán thành công, khách hàng sẽ được chuyển đến trang xác nhận, bao gồm mã đơn hàng và các thông tin về thời gian giao hàng.

### 1.6.5 Hệ thống con Quản lý Khách hàng:

A screenshot of a black screen

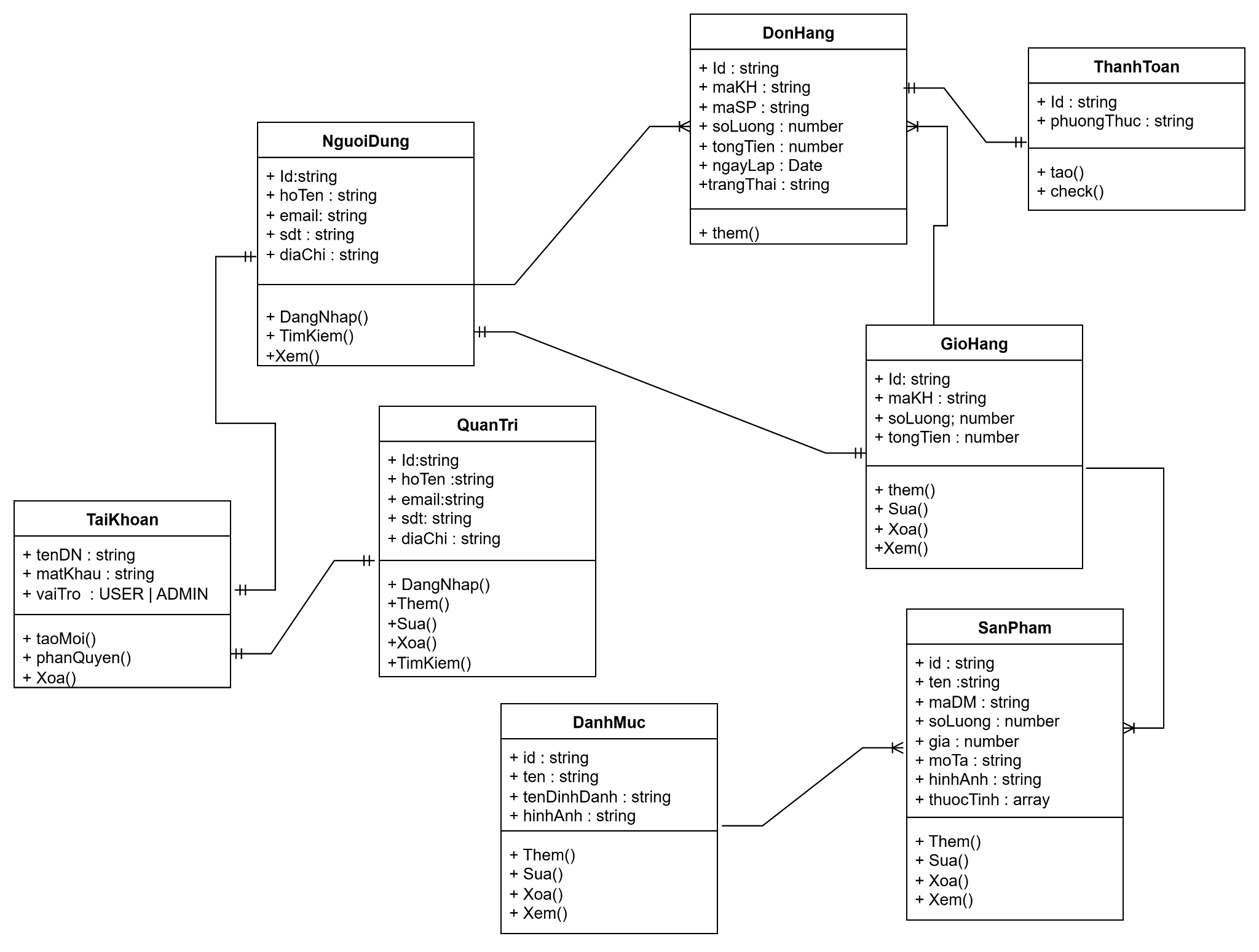
AI-generated content may be incorrect.Hình 10. Sơ đồ hệ thống con quản lý khách hàng.

* Chức năng:
  + Đăng ký tài khoản: Cho phép khách hàng mới đăng ký tài khoản bằng cách cung cấp thông tin cá nhân như tên, email, mật khẩu, và số điện thoại.
  + Đăng nhập/Đăng xuất: Cho phép khách hàng đăng nhập vào hệ thống để quản lý tài khoản cá nhân và lịch sử mua hàng. Cung cấp chức năng đăng xuất để thoát khỏi hệ thống.
  + Quản lý thông tin cá nhân: Khách hàng có thể chỉnh sửa thông tin cá nhân như địa chỉ giao hàng, số điện thoại, mật khẩu.
  + Quản lý lịch sử đơn hàng: Khách hàng có thể xem lại lịch sử mua hàng của mình, bao gồm thông tin chi tiết về từng đơn hàng.
* Giao diện khách hàng:
  + Trang đăng ký: Giao diện cho phép khách hàng mới nhập thông tin cá nhân và tạo tài khoản.
  + Trang đăng nhập: Cho phép khách hàng đăng nhập bằng email và mật khẩu.
  + Trang quản lý tài khoản: Hiển thị thông tin cá nhân của khách hàng, các tùy chọn chỉnh sửa thông tin cá nhân, và danh sách lịch sử đơn hàng.
  + Trang lịch sử đơn hàng: Liệt kê các đơn hàng trước đây của khách hàng, với tùy chọn xem chi tiết từng đơn hàng.

# 2 Thiết kế sơ đồ lớp

## 2.1 Phân tích kiến trúc hệ thống

### 2.1.2 Sơ đồ lớp:



Hình 12. Sơ đồ lớp

### 2.1.3Giải thích sơ đồ

#### 2.1.3.1. QuanTri (Quản trị viên)

* **Thuộc tính:**
  + Id: string → Mã quản trị viên.
  + hoTen: string → Họ tên.
  + email: string → Email.
* **Phương thức:**
  + DangNhap() → Đăng nhập vào hệ thống.
  + Them() → Thêm quản trị viên mới.
  + Sua() → Sửa thông tin quản trị viên.
  + Xoa() → Xóa quản trị viên.
  + TimKiem() → Tìm kiếm quản trị viên.

#### 2.1.3.2. KhachHang

* **Thuộc tính:**
  + maKH: string → Mã khách hàng.
  + hoTen: string → Họ tên.
  + email: string → Email.
  + diaChi: string → Địa chỉ.
  + sdt: string → Số điện thoại.
* **Phương thức:**
  + DangNhap() → Khách hàng đăng nhập.
  + TimKiem() → Tìm kiếm sản phẩm/thông tin.
  + Xem() → Xem sản phẩm, đơn hàng, giỏ hàng.

#### 2.1.3.3. DonHang

* **Thuộc tính:**
  + ID: string → Mã đơn hàng.
  + maKH: string → Mã khách hàng (ai đặt hàng).
  + maSP: string → Mã sản phẩm.
  + soLuong: number → Số lượng sản phẩm đặt.
  + tongTien: number → Tổng tiền đơn hàng.
  + ngayLap: Date → Ngày lập đơn.
  + trangThai: string → Trạng thái đơn hàng
* **Phương thức:**
  + Them() → Tạo đơn hàng.

#### 2.1.3.4. TaiKhoan

* **Thuộc tính:**
  + tenDN: string → Tên đăng nhập.
  + matKhau: string → Mật khẩu.
  + vaiTro: string → Vai trò (quản trị, khách hàng).
* **Phương thức:**
  + DangNhap() → Đăng nhập
  + taoMoi() → Tạo tài khoản mới.
  + Xoa() → Xóa tài khoản
  + phanQuyen() → Phân quyền khách hàng.

#### 2.1.3.5. GioHang

* **Thuộc tính:**
  + ID: string → Mã sản phẩm.
  + maKH: string → Mã khách hàng.
  + soLuong: number → Số lượng sản phẩm trong giỏ.
  + tongTien: number → Tổng tiền giỏ hàng.
* **Phương thức:**
  + Them() → Thêm sản phẩm vào giỏ.
  + Sua() → Sửa sản phẩm ở giỏ.
  + Xoa() → Xóa sản phẩm ở giỏ.
  + Xem() → Xem giỏ hàng ở giỏ.

#### 2.1.3.6. SanPham (Sản phẩm)

* **Thuộc tính:**
  + ID: string → Mã sản phẩm.
  + maDM : string → Mã danh mục
  + tenSP: string → Tên sản phẩm.
  + soLuong: number → Số lượng tồn kho.
  + gia: number → Giá sản phẩm.
  + moTa: string → Mô tả chi tiết.
  + hinhAnh: string → Hình ảnh sản phẩm.
* **Phương thức:**
  + Them() → Thêm sản phẩm
  + Sua() → Sửa sản phẩm
  + Xoa() → Xóa sản phẩm
  + Xem() → Xem sản phẩm

#### 2.1.3.7. ThanhToan (Thanh toán)

* **Thuộc tính:**
  + ID: string → Mã thanh toán.
  + phuongThuc: string → Phương thức (COD, chuyển khoản…).
* **Phương thức:**
  + Tao() → Tạo thanh toán
  + check() →Kiểm tra

#### 2.1.3.8. DanhMuc (Danh Mục)

* **Thuộc tính:**
  + ID: string → Mã danh mục
  + Ten : string → Tên danh mục
  + tenDinhDanh → Tên định danh
  + hinhAnh : string → Ảnh danh mục
* **Phương thức:**
  + Them() → Thêm danh mục
  + Sua() → Sửa danh mục
  + Xoa() → Xóa danh mục

# 3 Sơ đồ trình tự

## 3.1 Sơ đồ trình tự quản lý sản phẩm

### 3.1.1 Sơ đồ trình tự quản lý sản phẩm

A black background with white rectangles

AI-generated content may be incorrect.

Hình 13. Sơ đồ trình tự quản lý sản phẩm

### 3.1.2 Giải thích biều đồ

* **Hiển thị chi tiết sản phẩm**
  + Khách hàng chọn chức năng Hiển thị chi tiết sản phẩm.
  + productController kích hoạt và gửi yêu cầu đến productService.
  + productService truy vấn product từ database và trả về chi tiết sản phẩm cho UI.
* **Thêm sản phẩm**
  + Sau khi đăng nhập, khách hàng chọn chức năng Thêm sản phẩm.
  + Module productController được kích hoạt.
  + Hệ thống gửi yêu cầu thêm sản phẩm đến productService.
  + productService tạo mới product và lưu vào database, trả về "new product".
* **Sửa sản phẩm**
  + Khách hàng chọn chức năng Sửa sản phẩm.
  + productController kích hoạt và gửi yêu cầu sửa đến productService.
  + productService kiểm tra điều kiện và cập nhật product trong database.
  + Trả kết quả cho khách hàng qua "update product" và "thành công" hoặc "thất bại".
* **Xóa sản phẩm**
  + Khách hàng chọn chức năng Xóa sản phẩm.
  + productController kích hoạt và gửi yêu cầu xóa đến productService.
  + productService xóa product khỏi database và trả về "delete product" cùng "thành công" hoặc "thất bại".
* **Hiển thị thông báo**
  + Hệ thống phản hồi kết quả: thành công hoặc thất bại.
  + Nếu thành công → dữ liệu được cập nhật/xóa và lưu trong database.
  + Nếu thất bại → báo lỗi "hiển thị thông báo".

## 3.2 sơ đồ trình tự quản lý đăng nhập và khách hàng

### 3.2.1 Sơ đồ tuần tự chức năng đăng nhập và khách hàng

A black background with white rectangles

AI-generated content may be incorrect.

Hình 14. Sơ đồ trình tự quản lý đăng nhập và khách hàng

### 3.2.2 Giải thích sơ đồ

* **Đăng nhập**
  + Khách hàng chọn chức năng đăng nhập.
  + Hệ thống gửi yêu cầu xác thực đến authController: kiểm tra thông tin đăng nhập.
  + Nếu đúng thông tin → đăng nhập thành công và chuyển hướng đến UI.
  + Nếu sai thông tin → hiển thị thông báo "đăng nhập thất bại".
* **Chọn chỉnh sửa thông tin**
  + Sau khi đăng nhập, khách hàng chọn chức năng Chỉnh sửa tài khoản.
  + Module userController được kích hoạt.
  + Hệ thống kiểm tra điều kiện chỉnh sửa thông tin qua userService.
  + userService gọi để xác minh thông tin cần sửa có hợp lệ.
* **Cập nhật dữ liệu**
  + Nếu hợp lệ, yêu cầu chỉnh sửa được gửi tới danh sách user.
  + danh sách user cập nhật dữ liệu trong cơ sở dữ liệu.
  + Trả kết quả cho khách hàng qua "cập nhật" và "thành công" hoặc "thất bại".
* **Trả kết quả cho khách hàng**
  + Hệ thống phản hồi kết quả: thành công hoặc thất bại.
  + Nếu thành công → tài khoản được chỉnh sửa và lưu lại trong database.
  + Nếu thất bại → báo lỗi "thông báo lỗi".
* **Xóa tài khoản**
  + Khách hàng chọn chức năng xóa tài khoản.
  + userController kích hoạt và gửi yêu cầu xóa tới userService.
  + Nếu hợp lệ, dữ liệu được xóa khỏi database và trả về "thành công".
  + Nếu thất bại, hiển thị "thông báo lỗi".
* **Hiển thị danh sách khách hàng**
  + Khách hàng chọn chức năng hiển thị danh sách khách hàng.
  + userController kích hoạt và gửi yêu cầu tới userService.
  + userService truy vấn danh sách user từ database và trả về "list user" cho UI.

## 3.3 Sơ đồ tuần tự giỏ hàng

### 3.3.1 Sơ đồ tuần tự giỏ hàng

A black background with white text

AI-generated content may be incorrect.

Hình 15. Sơ đồ tuần tự giỏ hàng.

### 3.3.2 Giải thích sơ đồ

* **Thêm sản phẩm vào giỏ hàng**
  + Khách hàng chọn chức năng Thêm sản phẩm vào giỏ hàng.
  + Hệ thống gửi yêu cầu đến cartController.
  + cartController chuyển yêu cầu đến cartService.
  + cartService thêm sản phẩm (cart) vào database và trả về "add cart".
* **Hiển thị giỏ hàng**
  + Khách hàng chọn chức năng Hiển thị giỏ hàng.
  + cartController kích hoạt và gửi yêu cầu đến cartService.
  + cartService truy vấn giỏ hàng (cart) từ database và trả về chi tiết cho UI.
* **Cập nhật giỏ hàng**
  + Khách hàng chọn chức năng Cập nhật giỏ hàng.
  + cartController kích hoạt và gửi yêu cầu đến cartService.
  + cartService kiểm tra và cập nhật giỏ hàng (cart) trong database.
  + Trả kết quả cho khách hàng qua "update cart" và "cập nhật sản phẩm" hoặc "thông báo lỗi".
* **Xóa sản phẩm khỏi giỏ hàng**
  + Khách hàng chọn chức năng Xóa sản phẩm khỏi giỏ hàng.
  + cartController kích hoạt và gửi yêu cầu đến cartService.
  + cartService xóa sản phẩm khỏi giỏ hàng trong database và trả về "xóa sản phẩm khỏi hàng".
* **Xóa toàn bộ giỏ hàng**
  + Khách hàng chọn chức năng Xóa toàn bộ giỏ hàng.
  + cartController kích hoạt và gửi yêu cầu đến cartService.
  + cartService xóa toàn bộ giỏ hàng (empty cart) khỏi database và trả về "xóa toàn bộ giỏ hàng".

## 3.4 Sơ đồ quản lý đơn hàng

### 3.4.1 Sơ đồ quản lý đơn hàng

A black background with white text

AI-generated content may be incorrect.

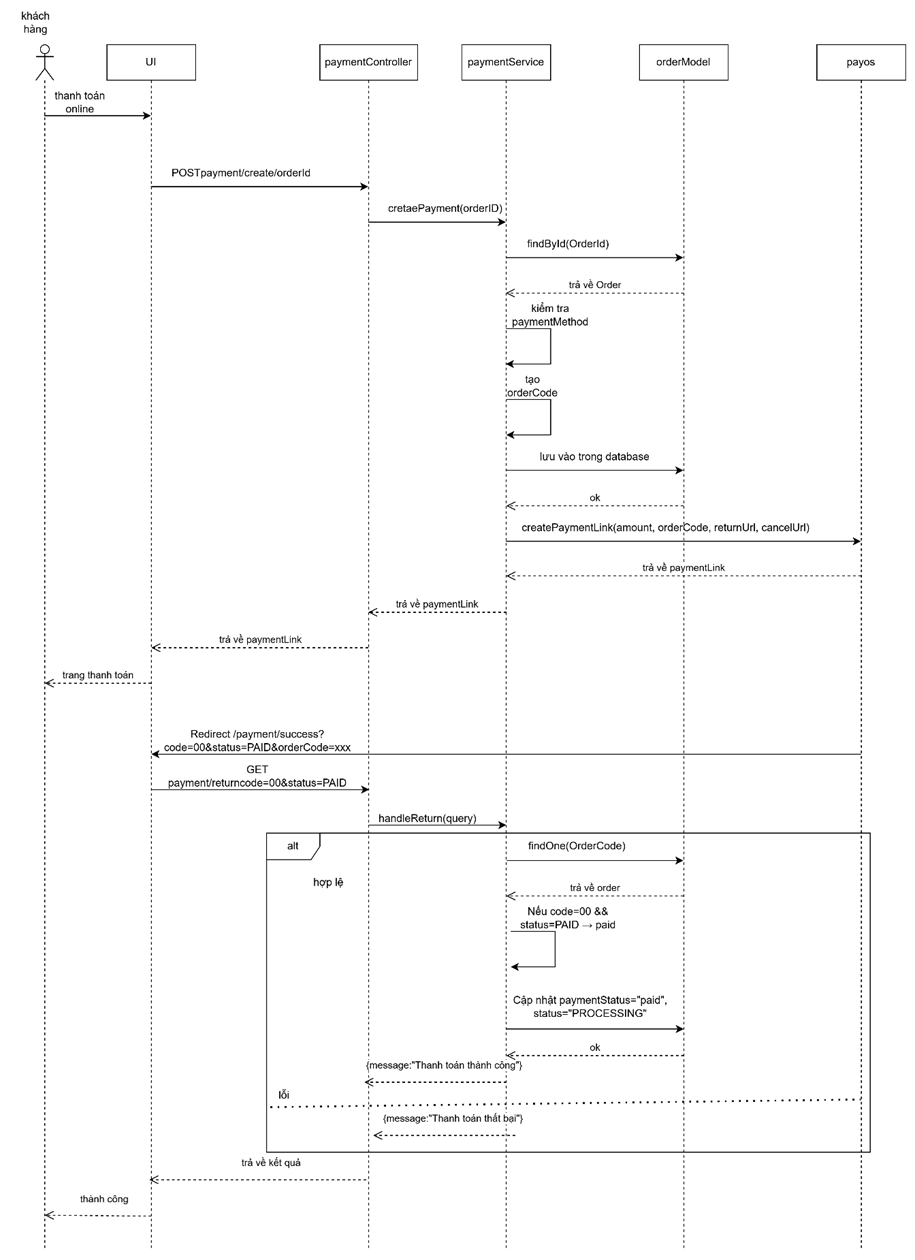
Hình 16. Sơ đồ quản lý đơn hàng.

### 3.4.2 Giải thích sơ đồ

* **Tạo đơn hàng**
  + Khách hàng chọn chức năng Tạo đơn hàng.
  + Hệ thống gửi yêu cầu đến orderController: kiểm tra thông tin đơn hàng.
  + orderController chuyển yêu cầu đến orderService.
  + orderService tạo đơn hàng (order) và lưu vào database, trả về "create order" cùng "order + total".
* **Hiển thị danh sách đơn hàng**
  + Khách hàng chọn chức năng Hiển thị danh sách đơn hàng.
  + orderController kích hoạt và gửi yêu cầu đến orderService.
  + orderService truy vấn danh sách đơn hàng từ database và trả về "danh sách đơn hàng" cho UI.
* **Cập nhật đơn hàng**
  + Khách hàng chọn chức năng Cập nhật đơn hàng.
  + orderController kích hoạt và gửi yêu cầu đến orderService.
  + orderService kiểm tra và cập nhật đơn hàng (order) trong database.
  + Trả kết quả cho khách hàng qua "update order" và "thành công" hoặc "thất bại".
* **Trả kết quả cho khách hàng**
  + Hệ thống phản hồi kết quả: thành công hoặc thất bại.
  + Nếu thành công → đơn hàng được cập nhật và lưu trong database.
  + Nếu thất bại → hiển thị "thông báo lỗi".

## 3.5 Sơ đồ quản lý thanh toán

### 3.5.1 Sơ đồ quản lý thanh toán



**3.5.2 Giải thích Sơ đồ**

**1. Tạo yêu cầu thanh toán**

* Khách hàng trên UI chọn chức năng Thanh toán đơn hàng.
* paymentController nhận yêu cầu thanh toán từ UI và chuyển tiếp đến paymentService.
* paymentService gọi orderModel để tạo một đơn hàng mới trong cơ sở dữ liệu
* orderModel lưu thông tin đơn hàng và trả về đối tượng Order cho paymentService.
* paymentService gửi yêu cầu đến PayOS để tạo liên kết thanh toán (payment link).
* PayOS xử lý và trả về paymentLink (liên kết thanh toán) cho paymentService.
* paymentService trả kết quả (paymentLink) về cho paymentController, sau đó gửi lại cho UI để hiển thị trang thanh toán cho khách hàng.

**2. Xử lý kết quả thanh toán**

* Khách hàng thực hiện thanh toán trên trang PayOS.
* Sau khi thanh toán, PayOS gọi webhook (callback) hoặc trả kết quả về paymentService thông qua paymentController.
* paymentService kiểm tra trạng thái thanh toán từ PayOS:
  + Nếu thành công → cập nhật trạng thái đơn hàng trong orderModel thành “paid”.
  + Nếu thất bại → cập nhật trạng thái đơn hàng thành “failed”.
* orderModel lưu lại thay đổi trong cơ sở dữ liệu.

**3. Phản hồi kết quả**

* paymentService trả về thông báo kết quả (message) cho paymentController:
  + Nếu thành công → trả về {message: "Thanh toán thành công"}.
  + Nếu thất bại → trả về {message: "Thanh toán thất bại"}.
* paymentController gửi kết quả cuối cùng về cho UI, hiển thị thông báo cho khách hàng:
  + “Thanh toán thành công” hoặc
  + “Thanh toán thất bại”.

## 3.6 Sơ đồ quản lý danh mục

### 3.6.1 Sơ đồ quản lý danh mục

Ảnh có chứa văn bản, ảnh chụp màn hình, số, Song song

Nội dung do AI tạo ra có thể không chính xác.

Hình 17 Quản lý danh mục

### 3.6.2 Giải thích sơ đồ

* Tạo danh mục:
  + Quản trị viên gửi yêu cầu POST /categories kèm CreateCategoryDto.
  + CategoryController gọi CategoryService.create(dto).
  + CategoryService gọi CategoryFactory.create(dto) để khởi tạo dữ liệu chuẩn.
  + Sau khi nhận CategoryData, Service tạo instance mới của model (new Category(...)) và gọi .save() để lưu vào MongoDB.
  + MongoDB trả về bản ghi đã lưu.
* Xem danh mục:
  + Quản trị viên gửi GET /categories?page=&limit=.
  + Controller gọi Service.findAllPaginated(page, limit).
  + Service dùng Model.find(skip, limit) để truy vấn dữ liệu theo phân trang và countDocuments() để đếm tổng số bản ghi.
  + Kết quả { data, total, page, limit } trả về qua Service → Controller → Admin.
* Cập nhật danh mục :
  + Quản trị viên gửi PUT /categories/:id kèm dữ liệu cập nhật.
  + Controller gọi Service.update(id, updateData).
  + Service gọi Model.findByIdAndUpdate(id, updateData).
  + MongoDB trả về bản ghi sau khi cập nhật (hoặc null nếu không tồn tại.
  + Service và Controller trả phản hồi 200 OK hoặc 404 Not Found cho người dùng.
* Xóa danh mục :
  + Quản trị viên gửi DELETE /categories/:id.
  + Controller gọi Service.remove(id).
  + Service gọi Model.findByIdAndDelete(id) để xóa danh mục.
  + MongoDB trả về bản ghi đã xóa hoặc null.
  + Service và Controller phản hồi 200 OK hoặc 404 Not Found cho Admin.

## 3.7 Thiết kế

### 3.7.1 Thiết kế giao diện

<https://www.figma.com/design/iYz587Jz8j7YsshYBJW1nn/Thi%E1%BA%BFt-k%E1%BA%BF?node-id=0-1&m=dev&t=osQRIxiuLHIVp5Sk-1>

# 4 Thiết kế Hành vi và Trạng thái

## 4.1 Sơ đồ trạng thái

### 4.1.1 Sơ đồ trạng thái quản lý sản phẩm

**A diagram of a flowchart

AI-generated content may be incorrect.**

Hình 18. Sơ đồ trạng thái quản lý sản phẩm

### Sơ đồ trạng thái sản phẩm

Sơ đồ trạng thái mô tả vòng đời của một sản phẩm trong hệ thống thương mại điện tử, thể hiện các trạng thái chính và các chuyển đổi giữa chúng khi người quản trị hoặc hệ thống thực hiện các thao tác như thêm hàng, cập nhật thông tin, hoặc xóa sản phẩm.

#### 1. Trạng thái ban đầu

**Tạo sản phẩm:** Khi người quản trị thêm sản phẩm mới vào hệ thống, sản phẩm được khởi tạo.  
 → Sản phẩm chuyển sang trạng thái Nhap.

#### 2. Nhap (Nhập)

Đây là trạng thái sản phẩm vừa được tạo và đang trong giai đoạn nhập kho (chưa đăng bán).  
 **Hành động:**

* Khi sản phẩm được đăng bán, hệ thống chuyển sang DangHoatDon.

#### 3. DangHoatDong (Đang hoạt động)

Sản phẩm đang được bán trên hệ thống, có sẵn hàng để khách hàng đặt mua.  
 Hành động:

* Khi số lượng hàng (amount) = 0, hệ thống chuyển sang HetHang.
* Khi nhập thêm hàng, hệ thống vẫn ở DangHoatDong.
* Khi sửa thông tin sản phẩm, hệ thống chuyển sang DaCapNhat.
* Khi xóa sản phẩm, hệ thống chuyển sang DaXoa.

#### 4. HetHang (Hết hàng)

Sản phẩm đã bán hết, không còn hàng tồn kho.  
 **Hành động:**

* Khi nhập thêm hàng, sản phẩm quay lại DangHoatDong.
* Khi sửa thông tin sản phẩm, hệ thống chuyển sang DaCapNhat.
* Khi xóa sản phẩm, hệ thống chuyển sang DaXoa.

#### 5. DaCapNhat (Đã cập nhật)

Sản phẩm đã được sửa thông tin (ví dụ: thay đổi giá, mô tả, số lượng...).  
**Hành động**:

* Nếu sản phẩm còn hàng, hệ thống chuyển lại DangHoatDong.
* Nếu amount = 0, chuyển sang HetHang.
* Nếu xóa sản phẩm, chuyển sang DaXoa.

#### 6. DaXoa (Đã xóa)

Sản phẩm đã bị xóa khỏi hệ thống — không còn được hiển thị hay kinh doanh nữa.  
Đây là trạng thái kết thúc của vòng đời sản phẩm.

### 4.1.2 Sơ đồ trạng thái quản lý thanh toán

**A diagram of a flowchart

AI-generated content may be incorrect.** Hình 19. Sơ đồ trạng thái quản lý thanh toán

### Sơ đồ trạng thái đơn hàng

Sơ đồ trạng thái mô tả vòng đời của một đơn hàng trong hệ thống thương mại điện tử, thể hiện các trạng thái chính và các chuyển đổi giữa chúng khi khách hàng hoặc hệ thống thực hiện các thao tác liên quan đến thanh toán, giao hàng hoặc hủy đơn.

#### 1. Trạng thái ban đầu

**Khách hàng tạo đơn hàng:** Khi khách hàng đặt hàng, hệ thống khởi tạo một đơn hàng mới.  
 → Đơn hàng chuyển sang trạng thái ChuaThanhToan.

#### 2. ChuaThanhToan (Chưa thanh toán)

Đây là trạng thái đơn hàng vừa được tạo, nhưng chưa xác nhận hoặc chưa thanh toán.  
 Hành động/Chuyển đổi:

* Khi khách hàng xác nhận đơn hàng, hệ thống chuyển sang DangXuLy.

#### 3. DangXuLy (Đang xử lý)

Đơn hàng đang được xử lý — bao gồm việc xác minh thanh toán với VNPay và chuẩn bị đơn hàng.  
 **Hành động:**

* Nếu VNPay ResponseCode = 00 (thanh toán thành công) → chuyển sang ThanhToanThanhCong.
* Nếu VNPay ResponseCode ≠ 00 (thanh toán thất bại) → chuyển sang ThanhToanThatBai.
* Nếu User/Admin hủy đơn → chuyển sang DaHuy.

#### 4. ThanhToanThanhCong (Thanh toán thành công)

Đơn hàng đã được thanh toán thành công qua VNPay.  
 Hành động/Chuyển đổi:

* Hệ thống bắt đầu xử lý đơn hàng → chuyển sang DangGiao.

#### 5. DangGiao (Đang giao)

Đơn hàng đang trong quá trình giao đến khách hàng.  
**Hành động:**

* Giao hàng thành công → chuyển sang HoanThanh.
* Hủy đơn (Admin hoặc User) → chuyển sang DaHuy.

#### 6. ThanhToanThatBai (Thanh toán thất bại)

Đơn hàng ở trạng thái này khi quá trình thanh toán không thành công.  
**Hành động:**

* Có thể bị User/Admin hủy đơn, hệ thống chuyển sang DaHuy.

#### 7. DaHuy (Đã hủy)

Đơn hàng đã bị hủy — do khách hàng, admin, hoặc do thanh toán thất bại.  
 Đây là trạng thái kết thúc, không thể quay lại xử lý hay thanh toán tiếp.

#### 8. HoanThanh (Hoàn thành)

Đơn hàng đã được giao hàng thành công và hoàn tất toàn bộ quy trình.  
Đây là trạng thái kết thúc của vòng đời đơn hàng.

### 4.1.3 Sơ đồ trạng thái quản lý đơn hàng

**A diagram of a flowchart

AI-generated content may be incorrect.**

Hình 20 . Sơ đồ trạng thái quản lý đơn hàng

### Sơ đồ trạng thái đơn hàng

Sơ đồ trạng thái mô tả vòng đời của một đơn hàng trong hệ thống thương mại điện tử, thể hiện các trạng thái chính và các chuyển đổi giữa chúng khi khách hàng hoặc hệ thống thực hiện các thao tác liên quan đến thanh toán, giao hàng hoặc hủy đơn.

#### 1. Trạng thái ban đầu

* **Tạo đơn hàng**:  
   Khi khách hàng đặt hàng, hệ thống khởi tạo một đơn hàng mới.  
   → Đơn hàng chuyển sang trạng thái ChuaThanhToan.

#### 2. ChuaThanhToan (Chưa thanh toán)

* Đây là trạng thái đơn hàng vừa được tạo nhưng chưa xác nhận hoặc chưa thanh toán.
* Hành động/Chuyển đổi:
  + Khi khách hàng xác nhận đơn hàng, hệ thống chuyển sang DangXuLy.

#### 3. DangXuLy (Đang xử lý)

* Đơn hàng đang được xử lý, có thể bao gồm việc xác minh thanh toán, chuẩn bị hàng, v.v.
* Hành động:  
  Thanh toán thành công → chuyển sang DaThanhToan.
  + Thanh toán thất bại → chuyển sang ThatBai.
  + Hủy đơn (do khách hàng hoặc hệ thống) → chuyển sang DaHuy.

#### 4. DaThanhToan (Đã thanh toán)

* Đơn hàng đã được thanh toán thành công.
* Hành động:
  + Giao hàng thành công → chuyển sang HoanThanh (đơn hàng hoàn tất).
  + Hủy đơn (do admin hoặc khách hàng) → chuyển sang DaHuy.

#### 5. ThatBai (Thất bại)

* Đơn hàng ở trạng thái này khi quá trình thanh toán không thành công.
* Hành động:
  + Có thể bị hủy hoàn toàn, dẫn đến trạng thái DaHuy.

#### 6. DaHuy (Đã hủy)

* Đơn hàng đã bị hủy — do khách hàng, admin, hoặc do thanh toán thất bại.
* Đây là một trạng thái kết thúc, không thể quay lại xử lý hay thanh toán tiếp.

#### 7. HoanThanh (Hoàn thành)

* Đơn hàng đã được giao hàng thành công và hoàn tất toàn bộ quy trình.
* Đây là trạng thái kết thúc của vòng đời đơn hàng.

### 4.1.4 Sơ đồ trạng thái mua hàng

**A diagram of a company

AI-generated content may be incorrect.** Hình 21. Sơ đồ trạng thái giỏ hàng

### Sơ đồ trạng thái giỏ hàng

Sơ đồ trạng thái mô tả **vòng đời của giỏ hàng** trong hệ thống thương mại điện tử, thể hiện **các trạng thái chính** và **các chuyển đổi** giữa chúng khi khách hàng thực hiện các thao tác thêm, cập nhật hoặc xóa sản phẩm trong giỏ.

#### 1. Rong (Rỗng)

Đây là trạng thái ban đầu, khi khách hàng chưa có sản phẩm nào trong giỏ hàng.  
**Hành động:**

* Khi khách hàng thêm sản phẩm vào giỏ, hệ thống chuyển sang CoSanPham.

#### 2. CoSanPham (Có sản phẩm)

Giỏ hàng hiện đang chứa ít nhất một sản phẩm.  
**Hành động:**

* Cập nhật số lượng sản phẩm (tăng hoặc giảm) → chuyển sang DaCapNhat.
* Xóa một sản phẩm → chuyển sang DaXoa.
* Xóa toàn bộ giỏ hàng → chuyển sang DaXoaHet.
* Nếu toàn bộ sản phẩm bị xóa, hệ thống tự động quay lại Rong.

#### 3. DaXoa (Đã xóa)

Giỏ hàng ở trạng thái này khi khách hàng xóa một sản phẩm khỏi giỏ.  
**Hành động:**

* Nếu vẫn còn sản phẩm trong giỏ, hệ thống quay lại CoSanPham.
* Nếu không còn sản phẩm nào, hệ thống chuyển sang Rong.

#### 4. DaCapNhat (Đã cập nhật)

Giỏ hàng đã được khách hàng thay đổi số lượng sản phẩm (tăng hoặc giảm).  
**Hành động:**

* Khi khách hàng lưu thay đổi, giỏ hàng quay lại CoSanPham để hiển thị danh sách mới nhất.

#### 5. DaXoaHet (Đã xóa hết)

Trạng thái này xảy ra khi khách hàng xóa toàn bộ sản phẩm trong giỏ hàng.  
**Hành động:**

* Sau thao tác này, giỏ hàng trở lại Rong, nghĩa là không còn sản phẩm nào.

# 5 Thiết kế hệ thống.

## 5.1 Sơ đồ gói

A screenshot of a computer

AI-generated content may be incorrect. Hình 22. Sơ đồ gói

### Mối quan hệ giữa các tầng

#### 1. Tầng UI → Tầng Controller

Tầng UI (Web Client) là nơi khách hàng trực tiếp thao tác trên giao diện — chẳng hạn như đăng nhập, thêm sản phẩm vào giỏ hàng, hoặc đặt mua hàng.  
 Khi khách hàng thực hiện một hành động, UI sẽ gửi yêu cầu (request) đến API tương ứng trong tầng Controller thông qua giao thức RESTful API.  
 → Tầng Controller tiếp nhận yêu cầu này và chuyển tiếp đến tầng xử lý nghiệp vụ (Service).

#### 2. Tầng Controller → Tầng Service

Các Controller (như UserController, CartController, OrderController, ProductController, CategoryController) không trực tiếp xử lý nghiệp vụ.  
 Thay vào đó, mỗi Controller đóng vai trò trung gian, nhận request từ UI, sau đó gọi đến Service tương ứng (UserService, CartService, OrderService, ProductService, CategoryService) để thực hiện các logic nghiệp vụ.  
 → Nhờ đó, Controller tách biệt hoàn toàn với logic xử lý, giúp hệ thống dễ bảo trì và mở rộng.

#### 3. Tầng Service → Tầng Database (MongoDB)

Tầng Service chịu trách nhiệm chính trong việc xử lý nghiệp vụ và tương tác với cơ sở dữ liệu.  
Tại đây, các nghiệp vụ như thêm, sửa, xóa, tìm kiếm dữ liệu sẽ được thực hiện thông qua các truy vấn đến MongoDB.  
Sau khi hoàn tất xử lý, Service trả kết quả về cho Controller, và Controller sẽ gửi phản hồi tương ứng về UI để hiển thị cho khách hàng.

# 6 Design Pattern

## 6.1 Design Pattern

**A diagram of a computer

AI-generated content may be incorrect.**

Hình 23. Design pattern

Trong hệ thống, mẫu thiết kế Singleton được áp dụng cho lớp DatabaseConnection. Mục đích chính của mẫu thiết kế này là đảm bảo trong toàn bộ chương trình chỉ tồn tại duy nhất một đối tượng kết nối đến cơ sở dữ liệu. Nhờ đó, các lớp dịch vụ khác như ProductService, UserService, CategoryService, CartService và OrderService đều có thể sử dụng chung một kết nối, tránh việc khởi tạo lại nhiều lần gây tốn tài nguyên và giảm hiệu suất của hệ thống.

Lớp DatabaseConnection có một thuộc tính tĩnh (static) là instance, dùng để lưu trữ duy nhất một thể hiện của lớp. Khi một lớp khác cần truy cập cơ sở dữ liệu, nó sẽ gọi phương thức getInstance(). Nếu thể hiện này chưa được tạo, hệ thống sẽ khởi tạo một đối tượng mới và lưu vào thuộc tính instance. Nếu đã tồn tại, phương thức sẽ trả về đối tượng đã có mà không cần tạo thêm. Cách làm này giúp duy trì một kết nối thống nhất trong suốt vòng đời của ứng dụng.

Các lớp như ProductService, UserService, CategoryService, CartService và OrderService không trực tiếp khởi tạo kết nối cơ sở dữ liệu, mà thông qua việc gọi DatabaseConnection.getInstance(). Điều này giúp các lớp dịch vụ chỉ tập trung xử lý nghiệp vụ, không cần quan tâm đến việc quản lý kết nối. Nhờ đó, hệ thống trở nên gọn gàng, dễ mở rộng và dễ bảo trì hơn.

Việc sử dụng Singleton trong hệ thống mang lại nhiều lợi ích quan trọng. Thứ nhất, nó giúp tiết kiệm tài nguyên vì chỉ có một kết nối được sử dụng xuyên suốt. Thứ hai, nó đảm bảo tính nhất quán của dữ liệu khi nhiều lớp cùng thao tác trên một cơ sở dữ liệu duy nhất. Thứ ba, việc thay đổi hoặc mở rộng cách thức kết nối chỉ cần thực hiện tại một nơi duy nhất trong mã nguồn, giúp giảm thiểu lỗi và tăng tính linh hoạt.

# 7 Tổng kết và hướng phát triển

## 7.1 Tổng kết

Sau quá trình thực hiện đề tài *“Xây dựng ứng dụng bán hàng quần áo thời trang trên website”*, nhóm đã hoàn thành đầy đủ các bước của quy trình phát triển phần mềm theo mô hình Waterfall, bao gồm: phân tích yêu cầu, thiết kế hệ thống, xây dựng chức năng, kiểm thử, và hoàn thiện báo cáo.

Hệ thống được xây dựng đã đáp ứng hầu hết các yêu cầu đề ra ban đầu:

* Cung cấp giao diện web thân thiện, dễ sử dụng, giúp khách hàng có thể xem sản phẩm, thêm vào giỏ hàng, thanh toán, và theo dõi đơn hàng.
* Cho phép quản trị viên (Admin) quản lý toàn bộ hệ thống bao gồm sản phẩm, danh mục, khách hàng và đơn hàng.
* Các chức năng chính đều được mô hình hóa thông qua Use Case Diagram, Activity Diagram, Sequence Diagram, Class Diagram, State Diagram, và được triển khai theo kiến trúc 3 tầng (UI – Controller – Service – Database) giúp hệ thống dễ mở rộng và bảo trì.
* Việc áp dụng mẫu thiết kế Singleton trong quản lý kết nối cơ sở dữ liệu giúp tăng hiệu năng và tính nhất quán dữ liệu trong toàn hệ thống.

Trong quá trình thực hiện, nhóm đã rèn luyện và củng cố nhiều kỹ năng quan trọng:

* Kỹ năng làm việc nhóm, phân chia công việc và phối hợp giữa các thành viên.
* Kỹ năng phân tích yêu cầu, thiết kế UML và viết tài liệu kỹ thuật.
* Kỹ năng lập trình web full-stack, sử dụng Node.js, Express, MongoDB, ReactJS và triển khai hệ thống qua Docker.

Tuy nhiên, do giới hạn thời gian và phạm vi dự án, hệ thống vẫn còn một số hạn chế:

* Giao diện chưa được tối ưu hoàn toàn cho thiết bị di động.
* Chưa tích hợp chức năng đánh giá sản phẩm và thống kê doanh thu chi tiết.
* Bảo mật tài khoản người dùng mới chỉ ở mức cơ bản (chưa có OTP hoặc xác thực hai lớp).

Dù vậy, sản phẩm đã đáp ứng đúng mục tiêu đề ra, thể hiện được toàn bộ quy trình phân tích và thiết kế phần mềm chuyên nghiệp, từ ý tưởng đến triển khai mô hình hoạt động thực tế.

## 7.2 Hướng phát triển

Trong tương lai, nhóm dự định mở rộng và hoàn thiện hệ thống theo các hướng sau:

1. Phát triển kiến trúc Microservices
   * Tách riêng các module như *User Service*, *Product Service*, *Order Service*, *Payment Service* để tăng khả năng mở rộng, chịu tải cao và dễ bảo trì.
2. Tích hợp thanh toán trực tuyến đa dạng
   * Bổ sung các cổng thanh toán khác ngoài VNPay (MoMo, ZaloPay, PayPal) để hỗ trợ nhiều lựa chọn hơn cho khách hàng.
3. Xây dựng ứng dụng di động (Mobile App)
   * Phát triển thêm ứng dụng di động bằng Flutter hoặc React Native, giúp người dùng trải nghiệm mua sắm thuận tiện trên điện thoại.
4. Cải thiện bảo mật và trải nghiệm người dùng (UX/UI)
   * Tích hợp xác thực hai lớp (2FA), JWT Refresh Token, và mã hóa dữ liệu mạnh hơn.
   * Thiết kế lại giao diện hiện đại, tối ưu thao tác và tốc độ phản hồi.
5. Ứng dụng trí tuệ nhân tạo (AI)
   * Tích hợp hệ thống gợi ý sản phẩm (Recommendation System) dựa trên hành vi người dùng và lịch sử mua hàng.
   * Phát triển chatbot hỗ trợ khách hàng tự động giải đáp thắc mắc.
6. Thêm tính năng báo cáo & thống kê cho quản trị viên
   * Biểu đồ doanh thu theo tháng, sản phẩm bán chạy, tỉ lệ chuyển đổi đơn hàng.
   * Dashboard trực quan phục vụ phân tích kinh doanh.

## 7.3 Kết luận chung

Đề tài là một bước thực hành quan trọng giúp nhóm tiếp cận quy trình phân tích – thiết kế – triển khai – kiểm thử phần mềm một cách có hệ thống và thực tế.  
Việc áp dụng mô hình Waterfall cùng các công cụ UML đã giúp nhóm hiểu sâu hơn về cách tổ chức và quản lý dự án phần mềm từ đầu đến cuối.

Qua đó, nhóm nhận thức rõ vai trò của việc thiết kế bài bản trước khi lập trình, cũng như tầm quan trọng của việc kiểm thử và tối ưu hóa trong quá trình phát triển.  
Đề tài không chỉ giúp nhóm củng cố kiến thức học phần mà còn là nền tảng để phát triển những dự án phần mềm quy mô lớn hơn trong tương lai.