**ĐẠI HỌC PHENIKAA**

**TRƯỜNG CÔNG NGHỆ THÔNG TIN PHENIKAA**

**A logo for a university

AI-generated content may be incorrect.**

**MÔN HỌC: PHÂN TÍCH VÀ THIẾT KẾ PHẦN MỀM**

**ĐỀ TÀI: “HỆ THỐNG QUẢN LÝ CỬA HÀNG THIẾT BỊ ĐIỆN TỬ TRỰC TUYẾN”**

**Thành viên nhóm**

|  |  |  |
| --- | --- | --- |
| **Nguyễn Văn Thăng** | **23010572** | **K17-KHMT(AI&KHDL)\_1** |
| **Phạm Văn Sự** | **23010523** | **K17-KHMT(AI&KHDL)\_1** |
| **Đặng Anh Tuyền** | **23010912** | **K17-KHMT(AI&KHDL)\_1** |
| **Trần Đình Dũng** | **23010596** | **K17-KHMT(AI&KHDL)\_1** |
| **Nguyễn Huy Toàn** | **23017052** | **K17-KHMT(AI&KHDL)\_1** |

**NHÓM 16**

**GIÁO VIÊN GIẢNG DẠY:**

**ThS VŨ QUANG DŨNG & ThS NGUYỄN XUÂN QUẾ**

**Thành viên nhóm**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Họ và tên** | **Mã sinh viên** | **Điểm bằng số** | **Điểm bằng chữ** | **Đóng góp** |
| **Nguyễn Văn Thăng** | **23010572** |  |  | **100%** |
| **Phạm Văn Sự** | **23010523** |  |  | **100%** |
| **Đặng Anh Tuyền** | **23010912** |  |  | **100%** |
| **Trần Đình Dũng** | **23010596** |  |  | **40%** |
| **Nguyễn Huy Toàn** | **23017052** |  |  | **10%** |

|  |  |
| --- | --- |
| **Chữ ký giám thị 1** | **Chữ ký giám thị 2** |
|  |  |

Bảng phân chia công việc.

|  |  |  |
| --- | --- | --- |
| STT | Họ và Tên | Nhiệm vụ |
| 1. | Nguyễn Văn Thăng  (Trưởng nhóm) | Phân tích yêu cầu, thiết kế lớp đối tượng, phân tích thiết kế database,thiết kế biểu đồ gói, code Giao diện và logic các giao diện trong admin và customer |
| 2. | Phạm Văn Sự | Vẽ biểu đồ lớp, Biểu đồ gói,Biểu đồ trạng thái, Phân tích thiết kế database, thiết kế biểu đồ gói, code Giao diện và logic các giao diện trong admin và customer |
| 3. | Đặng Anh Tuyền | Phân tích thiết kế database, Tạo database, Phân tích và code Giao diện và logic phần Auth, Hỗ trợ khách hàng |
| 4. | Trần Đình Dũng | Vẽ sơ đồ trình tự của các giao diện, chỉnh sửa chức vài chức năng trong giao diện |
| 5. | Nguyễn Huy Toàn | Viết kịch bản cho các giao diện, Vẽ biểu đồ hoạt động |

Phân bổ thời gian dự kiến.

|  |  |
| --- | --- |
| Giai đoạn/Nghiệp vụ | Thời gian ước tính |
| Giai đoạn 1: Khởi động & Lên kế hoạch | 2-4 tuần |
|  | Lấy yêu cầu, phân tích nghiệp vụ, lên kế hoạch sơ bộ. |
| Giai đoạn 2: Thực thi dự án (Phát triển) | 8-10 tuần |
|  | Chia thành 8-10 sprint (mỗi sprint 2 tuần) để phát triển: |
|  | Các module cốt lõi (sản phẩm, tồn kho, đơn hàng) |
|  | Các module phức tạp (đổi trả, khuyến mãi) |
|  | Các module hỗ trợ (báo cáo, quản trị) |
| Giai đoạn 3: Kiểm thử và ra mắt | 2-4 tuần |
|  | Kiểm thử toàn diện, sửa lỗi và triển khai hệ thống. |
| Tổng thời gian dự kiến | 12-18 tuần |

[**1. GIỚI THIỆU VÀ LẬP KẾ HOẠCH 2**](#_Toc212396121)

[**1.1. Giới thiệu 2**](#_Toc212396122)

[**1.1.1. Giới thiệu đề tài 2**](#_Toc212396123)

[**1.1.2. Mục Tiêu dự án 2**](#_Toc212396124)

[**1.1.3. Phạm vi triển khai 2**](#_Toc212396125)

[**1.2. Mô hình quy trình 4**](#_Toc212396126)

[**2. PHÂN TÍCH YÊU CẦU 4**](#_Toc212396127)

[**2.1. Yêu cầu chức năng 4**](#_Toc212396128)

[**2.2.1. Khách hàng 4**](#_Toc212396129)

[**2.2.2. Quản trị viên 5**](#_Toc212396130)

[**2.2. Yêu cầu phi chức năng 6**](#_Toc212396131)

[**2.3. Mô hình hóa 11**](#_Toc212396132)

[**2.3.1. Phân tích tác nhân 11**](#_Toc212396133)

[**2.3.2. Nhóm chức năng Quản lý Đơn hàng & Hỗ trợ 12**](#_Toc212396134)

[**2.3.3. Nhóm chức năng Quản trị Hệ thống 13**](#_Toc212396135)

[**3. THIẾT KẾ HỆ THỐNG 13**](#_Toc212396136)

[**3.1. Thiết kế kiến trúc tổng thể 14**](#_Toc212396137)

[**3.1.1. Lựa chọn mô hình kiến trúc 14**](#_Toc212396138)

[**3.2. Biểu đồ Gói 15**](#_Toc212396139)

[**3.2.1. Customer 15**](#_Toc212396140)

[**3.2.2. Admin 16**](#_Toc212396141)

[**3.3. Biểu đồ dữ liệu 19**](#_Toc212396142)

[**3.3.1. Quy trình đặt hàng 19**](#_Toc212396143)

[**3.3.2. Quy trình hỗ trợ 20**](#_Toc212396144)

[**3.4. Mô hình luồng dữ liệu 21**](#_Toc212396145)

[**3.4.1. Cấp 0 21**](#_Toc212396146)

[**3.4.1.1. Tác nhân 21**](#_Toc212396147)

[**3.4.1.2. Hộp đen hệ thống 22**](#_Toc212396148)

[**3.4.1.3. Luồng dữ liệu vào và ra 22**](#_Toc212396149)

[**3.4.1.4. Bảng đặc tả trao đổi dữ liệu 22**](#_Toc212396150)

[**3.4.2. Cấp 1 23**](#_Toc212396151)

[**3.4.2.1. Tổng quan 23**](#_Toc212396152)

[**3.4.2.2. Bảng chức năng 23**](#_Toc212396153)

[**3.5. Thiết kế chi tiết 25**](#_Toc212396154)

[**3.5.1. Sơ đồ lớp 25**](#_Toc212396155)

[**3.5.1.1. Nhóm Người dùng 26**](#_Toc212396156)

[**3.5.1.2. Nhóm khách hàng 28**](#_Toc212396157)

[**3.2.1.3. Nhóm quản trị 31**](#_Toc212396158)

[**3.5.1.4. Nhóm Catalog & Inventory 36**](#_Toc212396159)

[**3.5.1.5. Nhóm Cart 39**](#_Toc212396160)

[**3.5.1.6. Nhóm Order & Discount 43**](#_Toc212396161)

[**3.5.1.7. Nhóm Payment 48**](#_Toc212396162)

[**3.5.1.8. Nhóm Support, Review & Notification 49**](#_Toc212396163)

[**3.5.2. Sơ đồ trình tự 54**](#_Toc212396164)

[**3.5.2.1. Sơ đồ trình tự trạng thái đặt hàng 54**](#_Toc212396165)

[**3.5.2.2. Sơ đồ trình tự trạng thái hỗ trợ 55**](#_Toc212396166)

[**3.5.2.3. Sơ đồ trình tự trạng thái thêm sản phẩm mới. 56**](#_Toc212396167)

[**3.5.2.4. Sơ đồ trình tự trạng thái thêm khuyến mãi 57**](#_Toc212396168)

[**3.5.2.5. Sơ đồ trình tự trạng thái theo dõi đơn hàng 58**](#_Toc212396169)

[**3.5.3. Biểu đồ hoạt động 59**](#_Toc212396170)

[**3.5.4. Biểu đồ trạng thái 61**](#_Toc212396171)

[**3.5.4.1. Biểu đồ trạng thái – Tài khoản người dùng 61**](#_Toc212396172)

[**3.5.4.2. Biểu đồ trạng thái – Tồn kho Sản phẩm 62**](#_Toc212396173)

[**3.5.4.3. Biểu đồ trạng thái – Chọn sản phẩm 64**](#_Toc212396174)

[**3.5.4.4. Biểu đồ trạng thái – Thêm giỏ hàng 66**](#_Toc212396175)

[**3.5.4.5. Biểu đồ trạng thái – Đơn hàng 68**](#_Toc212396176)

[**4. TỔNG KẾT VÀ HƯỚNG PHÁT TRIỂN 71**](#_Toc212396177)

[**4.1. Đánh giá 71**](#_Toc212396178)

[**4.2. Hướng phát triển 71**](#_Toc212396179)

**Lời mở đầu**

Sự phát triển mạnh mẽ của thương mại điện tử đang định hình lại cách doanh nghiệp tiếp cận khách hàng và vận hành bán lẻ, đặc biệt trong ngành thiết bị điện tử với danh mục rộng, nhiều biến thể sản phẩm (màu sắc, dung lượng, cấu hình), chu kỳ đổi mẫu ngắn và nhu cầu hậu mãi cao. Từ thực tế đó, nhóm thực hiện đề tài “Hệ thống Quản lý Cửa hàng Thiết Bị Điện Tử Trực Tuyến” nhằm xây dựng một nền tảng hỗ trợ trọn vẹn quy trình bán hàng: quản lý sản phẩm và tồn kho, giỏ hàng và đơn hàng, thanh toán trực tuyến, khuyến mãi và voucher, theo dõi vận chuyển và chăm sóc khách hàng, đồng thời đáp ứng các yêu cầu về hiệu năng, bảo mật và trải nghiệm người dùng.

Trong khuôn khổ học phần Phân tích & Thiết kế Phần mềm, dự án là cơ hội để nhóm em áp dụng một quy trình phát triển hoàn chỉnh: bắt đầu từ khai thác và làm rõ yêu cầu, tiếp tục mô hình hóa nghiệp vụ, tiến hành thiết kế kiến trúc và dữ liệu, triển khai hiện thực các chức năng và kiểm thử để bảo đảm chất lượng trước khi bàn giao. Báo cáo trình bày mục tiêu, phạm vi và kiến trúc hệ thống, các mô-đun chức năng trọng tâm, những tiêu chí phi chức năng (hiệu năng, an toàn thông tin, khả dụng), cùng thách thức kỹ thuật và định hướng mở rộng như tối ưu tìm kiếm, gợi ý sản phẩm và báo cáo phân tích. Kỳ vọng của nhóm em là sản phẩm vừa giải quyết bài toán quản trị vận hành cho cửa hàng điện tử trực tuyến, vừa mang lại trải nghiệm mua sắm tin cậy, thuận tiện và nhất quán cho người dùng.

# **1. GIỚI THIỆU VÀ LẬP KẾ HOẠCH**

## **1.1. Giới thiệu**

### **1.1.1. Giới thiệu đề tài**

Đề tài xây dựng Hệ thống Quản lý Cửa hàng Thiết bị Điện tử trực tuyến, giúp khách hàng tìm kiếm, đặt mua, thanh toán an toàn và theo dõi đơn hàng mạch lạc. Ở phía vận hành, hệ thống cung cấp nền tảng quản trị tập trung để quản lý sản phẩm, tồn kho, đơn hàng, khuyến mãi và chăm sóc khách hàng. Ba tác nhân chính gồm Khách hàng, Quản trị viên và Cổng thanh toán tích hợp, phối hợp để bảo đảm trải nghiệm liền mạch, vận hành minh bạch và khả năng mở rộng ổn định.

**1.1.2. Mục Tiêu dự án**

Trải nghiệm khách hàng diễn ra mượt mà và rõ ràng: người dùng đăng ký hoặc đăng nhập, sau đó tìm kiếm và lọc sản phẩm, xem chi tiết với hình ảnh, mô tả, tồn kho và đánh giá. Tiếp theo, họ quản lý giỏ hàng, đặt mua, thanh toán, theo dõi trạng thái đơn và xem lịch sử giao dịch. Sau khi nhận hàng, khách có thể để lại đánh giá và nhận hỗ trợ trực tuyến ngay trong tài khoản.

Phía vận hành dành cho quản trị viên. Quản lý người dùng, danh mục, sản phẩm, tồn kho. Điều phối quy trình đơn từ xác nhận đến giao nhận. Thực hiện thanh toán và hoàn tiền. Thiết lập khuyến mãi và triển khai chiến dịch marketing. Theo dõi báo cáo và thống kê theo thời gian thực để ra quyết định nhanh và chính xác.

Thanh toán an toàn với tích hợp cổng thanh toán, xác thực giao dịch bằng OTP và 3DS, mã hóa dữ liệu khi truyền và khi lưu trữ, hỗ trợ hoàn tiền và đối soát minh bạch, giảm rủi ro gian lận, bảo vệ thông tin nhạy cảm của khách hàng.

Mục tiêu phi chức năng được lượng hóa rõ ràng. Thời gian tải trang dưới 5 giây khi số người dùng đồng thời dưới 1000. Thông lượng xử lý ít nhất 500 đơn/phút. Bảo mật đáp ứng TLS từ trở lên 1.2 và mã hóa dữ liệu khi lưu trữ. Hạ tầng mở rộng để tăng gấp đôi quy mô trong 6 tháng. Người dùng mới hoàn tất đặt hàng trong dưới 2 phút. Những chuẩn này giúp hệ thống nhanh, an toàn, dễ mở rộng.

### **1.1.3. Phạm vi triển khai**

- **Đối tượng sử dụng**: Khách hàng, Quản trị viên/nhân sự vận hành

**- Chức năng trong phạm vi**:

+ Khách hàng: đăng ký/đăng nhập, tìm–lọc, xem chi tiết, giỏ hàng, đặt hàng, thanh toán, theo dõi & lịch sử đơn, đánh giá, yêu cầu hỗ trợ, nhận thông báo.

+ Quản trị viên: quản lý tài khoản, sản phẩm & tồn kho (thêm/sửa/xóa, cảnh báo hết hàng), quy trình đơn (xác nhận–đóng gói–giao–hủy/hoàn), CSKH (xử lý ticket/đánh giá), thanh toán/hoàn tiền, khuyến mãi/voucher, báo cáo–thống kê doanh thu/tồn kho/tỷ lệ đơn.

- Cổng thanh toán: nhận yêu cầu, xác thực giao dịch, trả kết quả, hỗ trợ hoàn tiền, ghi log & báo cáo. Đây là phần cải tiến thêm đối với phần mềm để có thể phù hơp với xu hướng hiện nay. Nhưng đối với tâm lý khách hàng, việc thanh toán một số tiền lớn sẽ khiến cho tâm lý khách hàng không được thoải mái và thiếu sự tin tưởng vì tình trạng lừa đảo hiện nay đnag có xu hướng tăng mạnh, phần mềm chưa được thiết kế chi tiết để tránh được các lỗ hổng bảo mật.

**- Giới hạn/ngoài phạm vi**: Tập trung mô hình một cửa hàng trực tuyến; chưa bao gồm quản lý đa chi nhánh/ERP phức tạp (có thể mở rộng về sau)

**Đối tượng sử dụng**: Khách hàng. Quản trị viên và nhân sự vận hành. Cổng thanh toán bên thứ ba.

**Chức năng trong phạm vi**:

* **Khách hàng**: Đăng ký, đăng nhập. Cập nhật hồ sơ, địa chỉ, phương thức liên hệ. Tìm và lọc theo từ khóa, danh mục, thương hiệu, khoảng giá, còn hàng. Xem chi tiết với ảnh, mô tả, thông số, tồn kho theo kho, thời gian giao dự kiến. Quản lý giỏ hàng với thêm, bớt, đổi số lượng, áp mã giảm giá hợp lệ. Đặt hàng với địa chỉ, vận chuyển, phí hiển thị trước thanh toán. Thanh toán qua cổng, xác thực OTP hoặc 3DS, nhận kết quả. Theo dõi trạng thái đơn theo thời gian thực, xem lịch sử đơn. Đánh giá sản phẩm sau Delivered, một đơn một đánh giá cho mỗi sản phẩm. Yêu cầu hỗ trợ, tạo ticket, theo dõi phản hồi. Nhận thông báo qua email, sms, đẩy ứng dụng.  
  => Luồng điển hình: Người dùng đăng ký hoặc đăng nhập, tìm và lọc sản phẩm, xem chi tiết, thêm vào giỏ, đặt hàng, thanh toán, theo dõi trạng thái đơn và đánh giá sau khi nhận hàng.
* **Quản trị viên**: Quản lý tài khoản người dùng với khóa, mở, đặt lại mật khẩu. Quản lý sản phẩm và tồn kho với thêm, sửa, xóa, ảnh, thuộc tính, cảnh báo sắp hết hàng, giữ hàng khi có đơn. Quy trình đơn với tiếp nhận, xác nhận, đóng gói, bàn giao vận chuyển, theo dõi giao thành công, hủy hoặc hoàn khi cần. CSKH với xử lý ticket, phân loại, gán người phụ trách, quản lý phản hồi và đánh giá. Thanh toán và hoàn tiền với tra cứu giao dịch, tạo yêu cầu hoàn, theo dõi trạng thái hoàn. Khuyến mãi và voucher với tạo, lên lịch, điều kiện áp dụng, giới hạn theo nhóm khách. Báo cáo và thống kê doanh thu, tồn kho, tỷ lệ đơn, tỷ lệ hủy, hiệu năng thanh toán. Phân quyền theo vai trò kho, chăm sóc, kế toán, quản trị hệ thống.

**Giới hạn hoặc ngoài phạm vi**: Tập trung một cửa hàng trực tuyến. Chưa bao gồm đa chi nhánh, đồng bộ kho đa điểm, tích hợp ERP phức tạp. Chưa bao gồm chương trình thành viên nâng cao như tier, tích điểm liên kết đối tác. Có thể mở rộng về sau theo nhu cầu.

**Tiêu chí và ràng buộc chính**: Thời gian tải trang chủ và danh mục không quá 5 giây với người dùng đồng thời dưới 1000. Thông lượng xử lý tối thiểu 500 đơn/phút trong giờ cao điểm. Bảo mật với HTTPS TLS từ 1.2 trở lên, khuyến nghị 1.3. Mã hóa dữ liệu nhạy cảm khi lưu trữ. Quy trình đặt hàng cho người dùng mới dưới 2 phút. Độ trễ cập nhật trạng thái đơn từ webhook vận chuyển tối thiểu 5 giây. Tỷ lệ thất bại thanh toán trên tổng giao dịch không quá 2 phần trăm.

## **1.2. Mô hình quy trình**

Nhóm lựa chọn Agile – Scrum làm mô hình phát triển vì bảo đảm lặp–tăng trưởng, bàn giao sớm, và thích ứng nhanh với thay đổi.

Tóm lược cách vận hành: Dự án chia thành các đợt làm việc ngắn 1–2 tuần; cuối mỗi đợt làm việc tạo chạy để trình diễn và nhận phản hồi từ khách hàng và người dùng. Công việc quản lý qua hai bước quan trọng là phân tích yêu cầu, các tính năng, ý tưởng cần thực hiện cho sản phẩm và làm việc nhóm chọn ra những yêu cầu chức năng để hoàn thành. Mỗi ngày hoặc mỗi tuần nhóm sẽ họp ngắn trong một buổi để cập nhật tiến độ, nêu vấn đề và điều chỉnh kế hoạch nếu cần. Tiến độ của nhóm được theo dõi bằng biểu đồ burn-down để cho thấy lượng công việc còn lại theo thới gian. Chất lượng công việc được đảm bảo nhờ 2 tiêu chuẩn là Definition of Ready( quy định điều kiện mà mọi công việc phải đạt được trước khi nhóm bắt đầu và Definition of Done(quy định khi nào một công việc được coi là hoàn thành.

Lý do lựa chọn (phù hợp đặc thù dự án & nhóm 5 thành viên):

* Tối ưu cho nhóm nhỏ: Mô hình khuyến khích giao tiếp trực tiếp; hàng ngày ít nhất 15 phút giúp mọi người nắm rõ tiến độ, vướng mắc và phối hợp nhanh.
* Thích ứng thay đổi yêu cầu: Bài tập lớn thường biến động theo phản hồi giảng viên; Mô hình cho phép ưu tiên lại sau mỗi cập nhật tiến độ, giảm rủi ro “đi sai hướng”.
* Minh bạch & dễ quản trị: Tất cả chức năng tập trung trong bước phân tích yêu cầu và chức năng; mỗi chức năng cam kết phạm vi cụ thể , phân rã thành công việc có ước lượng, dễ theo dõi trạng thái.
* Bàn giao liên tục, cải tiến liên tục: Mỗi chức năng có thể sẵn sàng chạy thử, nhận phản hồi sớm để tinh chỉnh; nhóm họp và chốt 1–2 hành động cải tiến quy trình/kỹ thuật cho các chức năng kế tiếp.
* Bám mục tiêu chất lượng: DoD yêu cầu hoàn tất khi đã vượt qua, đáp ứng đúng yêu cầu (hiệu năng, bảo mật, khả dụng), tài liệu & nhật ký hệ thống được cập nhật, qua đó nâng chất lượng bản phát hành.

Kết luận: Mô hình cung cấp khung làm việc gọn, rõ và hiệu quả cho nhóm nhỏ, giúp dự án bám mục tiêu, giảm rủi ro, nâng chất lượng và tăng tốc độ bàn giao trong suốt học kỳ.

# **2. PHÂN TÍCH YÊU CẦU**

## **2.1. Yêu cầu chức năng**

**2.2.1. Khách hàng**

Đăng ký, đăng nhập, tìm kiếm, lọc, xem chi tiết:

Đăng ký hoặc đăng nhập nhanh bằng email hoặc số điện thoại, có thể xác thực OTP. Tìm theo từ khóa, nhận gợi ý tức thì, lọc theo danh mục, giá, thương hiệu, còn hàng, sắp xếp theo mới nhất hoặc bán chạy. Mở trang chi tiết để xem ảnh rõ nét, mô tả, thông số, điểm đánh giá và số lượt, tình trạng tồn kho, thời gian giao dự kiến theo địa chỉ đã lưu.  
=> Luồng: Người dùng đăng ký hoặc đăng nhập, sau đó tìm và lọc sản phẩm, xem chi tiết, chọn biến thể nếu có rồi thêm vào giỏ hàng.

Giỏ hàng, đặt hàng, theo dõi và lịch sử:

Giỏ hàng cho phép đổi số lượng, xóa mục, lưu để mua sau, áp mã giảm giá và thấy lý do nếu không hợp lệ. Tại bước đặt hàng, chọn địa chỉ, ghi chú, chọn hãng vận chuyển, phí và thời gian giao hiển thị tức thì. Sau khi xác nhận, nhận mã đơn, theo dõi dòng thời gian giao hàng, xem lại lịch sử và tải hóa đơn bất cứ lúc nào.  
=> Luồng: Người dùng chọn sản phẩm vào giỏ hàng, nhập địa chỉ nhận hàng, chọn hình thức vận chuyển và thanh toán, sau đó xác nhận đơn hàng, theo dõi quá trình giao hàng và xem lại lịch sử mua hàng.

Hỗ trợ, chat, đánh giá, thông báo:

Khi cần trợ giúp, mở chat trong tài khoản, chọn chủ đề thanh toán hoặc vận chuyển hoặc đổi trả, đính kèm ảnh nếu cần, nhận phản hồi kịp thời. Sau khi nhận hàng, để lại đánh giá có sao và nhận xét, một đơn một đánh giá cho mỗi sản phẩm, hiển thị sau khi duyệt nhanh. Nhận thông báo về cập nhật đơn, hoàn tiền, mã giảm giá và ưu đãi mới, có thể bật tắt trong cài đặt.  
Luồng: Người dùng cần hỗ trợ sẽ mở khung chat để trao đổi và nhận phản hồi từ nhân viên. Sau khi nhận hàng, họ có thể viết đánh giá sản phẩm và nhận thông báo từ hệ thống.

**2.2.2. Quản trị viên**

Đăng nhập hệ thống quản trị: Quản trị viên có quyền cao nhất, đăng nhập bằng tài khoản bảo mật cao, được mã hóa để tránh rò rỉ dữ liệu, mọi thao tác đều được kiểm soát theo vai trò.

Quản lý người dùng và tài khoản: Xem danh sách người dùng theo bộ lọc, khóa hoặc mở tài khoản khi phát hiện rủi ro, đặt lại mật khẩu khi có yêu cầu hợp lệ, xem lịch sử đăng nhập và lịch sử thao tác để truy vết, gán vai trò và phạm vi quyền phù hợp.

Quản lý sản phẩm và kho: Tạo mới, chỉnh sửa, xóa sản phẩm, quản lý thuộc tính, ảnh, biến thể, cập nhật tồn kho theo từng kho, thực hiện nhập và xuất kho có chứng từ, đặt ngưỡng cảnh báo sắp hết hàng, theo dõi lịch sử biến động tồn.

Luồng điển hình: Hệ thống phát hiện sản phẩm có tồn kho thấp, quản trị viên tạo phiếu nhập bổ sung. Sau khi cập nhật tồn kho thành công, cảnh báo tự động được gỡ bỏ.

Quản lý đơn hàng: Xem danh sách theo trạng thái, tiếp nhận và xác nhận đơn hợp lệ, in phiếu lấy hàng và phiếu đóng gói, bàn giao vận chuyển, cập nhật giao thành công hoặc thất bại, xử lý hủy theo quy định, tiếp nhận yêu cầu trả hàng và đổi hàng, in hóa đơn và chứng từ liên quan.  
Luồng điển hình: Đơn hàng mới được tạo, quản trị viên xác nhận đơn, tiến hành đóng gói và giao hàng. Kết quả có thể là giao thành công, hủy đơn hoặc xử lý trả hàng.

Chăm sóc khách hàng: Xem hồ sơ và lịch sử mua hàng, đọc và phản hồi yêu cầu hỗ trợ, ghi chú nội bộ để đồng bộ giữa các bộ phận, duyệt hoặc xóa đánh giá vi phạm tiêu chuẩn, đảm bảo phản hồi đúng thời gian cam kết.

Quản lý thanh toán: Theo dõi giao dịch theo thời gian thực, xác nhận giao dịch khi nhận được đối soát hợp lệ, xử lý hoàn tiền toàn phần hoặc một phần, kiểm tra kết nối với cổng thanh toán, lưu log yêu cầu và phản hồi để tra soát, bảo đảm không phát sinh giao dịch trùng.

Khuyến mãi và marketing: Tạo mã giảm theo sản phẩm, theo thương hiệu, theo giá trị giỏ, đặt điều kiện áp dụng và thời gian hiệu lực, giới hạn cho nhóm khách hàng cụ thể, theo dõi hiệu quả chiến dịch, dừng hoặc gia hạn khi cần.

Báo cáo và thống kê: Theo dõi doanh thu theo ngày và theo kênh, xem tồn kho hiện tại và vòng quay tồn, theo dõi tỷ lệ đơn theo trạng thái từ xác nhận đến giao thành công, quan sát tỷ lệ hủy và tỷ lệ trả hàng, xuất báo cáo định kỳ để phục vụ lập kế hoạch vận hành và nhập hàng

## **2.2. Yêu cầu phi chức năng**

Bên cạnh các yêu cầu về chức năng, hệ thống cũng cần đáp ứng các tiêu chí về chất lượng, hiệu suất, bảo mật và trải nghiệm người dùng. Các yêu cầu phi chức năng chính bao gồm:

**Hiệu năng**

* **Tốc độ phản hồi:** Thời gian phản hồi trung bình (TTFB) của trang không vượt quá 800 ms (p95). Thời gian tải đầy đủ dưới 5 giây với tối đa 1000 người dùng đồng thời.  
  Các API chính như danh mục, sản phẩm và tìm kiếm có p95 dưới 300 ms; các bước thanh toán có p95 dưới 500 ms.
* **Thông lượng xử lý:** Hệ thống có thể xử lý ít nhất 500 đơn hàng mỗi phút, 5000 truy vấn tìm kiếm mỗi phút và 1000 lượt callback thanh toán mỗi phút ổn định.
* **Tình huống tải cao:** Có khả năng chịu tải đột biến gấp 5 lần trong vòng 10 phút với tỷ lệ lỗi không vượt quá 0,5 phần trăm. Cơ chế retry và idempotency đảm bảo không xảy ra trùng giao dịch.
* **Giám sát:** Theo dõi thời gian thực, cảnh báo khi p95 vượt ngưỡng trong hơn 10 phút liên tục. Ghi log chi tiết để phục vụ truy vết và phân tích sự cố.

**Khả năng mở rộng**

* **Mục tiêu:** Hệ thống có thể mở rộng gấp đôi số lượng sản phẩm (SKU) và người dùng hoạt động hàng tháng trong thời gian không quá 6 tháng, đồng thời duy trì chỉ số SLA ổn định.
* **Chiến lược:** Hỗ trợ mở rộng ngang bằng cách thêm node cho web, API, search và cache; mở rộng dọc bằng cách tăng CPU hoặc RAM cho các node quan trọng.  
  Thiết kế kiến trúc phân tầng rõ ràng gồm Presentation, Service và Data; sử dụng cache phân tán, hàng đợi xử lý nền và cơ chế idempotency cho các tác vụ quan trọng.
* **Tiêu chí chấp nhận:** Khi tải tăng gấp đôi, p95 của các API vẫn đạt chuẩn, tỷ lệ lỗi 5xx nằm trong ngưỡng cho phép và chi phí hạ tầng tăng tuyến tính hoặc thấp hơn.

**Tương thích và Di động**

* **Thiết bị và trình duyệt:** Hỗ trợ Chrome, Edge, Safari và Firefox trong hai phiên bản gần nhất. Thiết kế responsive cho viewport từ 360 đến 414 px, đảm bảo giao diện thống nhất trên desktop, tablet và mobile.
* **Trải nghiệm người dùng:** Điều hướng cảm ứng mượt mà, phông chữ dễ đọc, kích thước nút chạm đạt chuẩn tiếp cận, hiển thị trạng thái tải và lỗi nhất quán.
* **Tích hợp:** Tương thích với các cổng thanh toán và đơn vị vận chuyển phổ biến.  
  Hỗ trợ định dạng CSV và XLSX, xử lý tốt tiếng Việt và ký tự đặc biệt.
* **Độ tin cậy và Khả dụng**
* **Mức độ khả dụng (SLA):** Giao diện người dùng đạt tối thiểu 99,9 phần trăm mỗi tháng (downtime không quá 0,72 giờ). Quy trình thanh toán và đơn hàng đạt tối thiểu 99,95 phần trăm mỗi tháng (downtime không quá 0,36 giờ).
* **Khả năng chịu lỗi:** Cơ chế retry có backoff và jitter, circuit breaker cho các dịch vụ bên ngoài, hàng đợi cho tác vụ nền, tự động failover trong vòng dưới 1 phút.
* **Theo dõi:** Báo cáo thời gian phản hồi p95 và p99, tách rõ thời gian phía client và phía server, cảnh báo khi tiêu thụ hơn 50 phần trăm ngân sách lỗi trong 7 ngày.

**Độ tin cậy và Khả dụng**

* **Mức độ khả dụng (SLA):** Giao diện người dùng đạt tối thiểu 99,9 phần trăm mỗi tháng (downtime không quá 0,72 giờ). Quy trình thanh toán và đơn hàng đạt tối thiểu 99,95 phần trăm mỗi tháng (downtime không quá 0,36 giờ).
* **Khả năng chịu lỗi:** Cơ chế retry có backoff và jitter, circuit breaker cho các dịch vụ bên ngoài, hàng đợi cho tác vụ nền, tự động failover trong vòng dưới 1 phút.
* **Theo dõi:** Báo cáo thời gian phản hồi p95 và p99, tách rõ thời gian phía client và phía server, cảnh báo khi tiêu thụ hơn 50 phần trăm ngân sách lỗi trong 7 ngày.

**Bảo trì**

* **Phục hồi sự cố:** Thời gian khôi phục trung bình (MTTR) không quá 60 phút đối với lỗi nghiêm trọng, thời gian phát hiện (MTTD) không quá 5 phút.
* **Triển khai an toàn:** Thực hiện canary deployment theo tỷ lệ 5 phần trăm – 25 phần trăm – 100 phần trăm, rollback trong vòng 5 phút, triển khai tự động qua CI/CD.
* **Sao lưu và khôi phục:** RPO không vượt quá 5 phút nhờ binlog hoặc CDC, backup hàng ngày, diễn tập khôi phục mỗi quý.
* **Quản lý thay đổi:** Yêu cầu review bởi ít nhất hai người, checklist phát hành đầy đủ, migration tương thích ngược, bật tắt tính năng bằng feature flag.

**Bảo mật và Tuân thủ**

* **Mã hóa:** Dữ liệu truyền tải sử dụng TLS phiên bản 1.2 trở lên (ưu tiên 1.3). Dữ liệu nhạy cảm được mã hóa bằng AES-256. Mật khẩu lưu trữ bằng bcrypt hoặc argon2 kèm salt và tham số chi phí phù hợp. Không lưu thông tin thẻ thanh toán, chỉ lưu token từ cổng thanh toán.
* **Thanh toán an toàn:** Tuân thủ tiêu chuẩn PCI DSS, xác minh chữ ký và nội dung callback, đối chiếu số tiền và mã đơn hàng. Ghi log đầy đủ cho mọi giao dịch, sử dụng idempotency để tránh trùng lặp.
* **Phân quyền:** Áp dụng RBAC với nguyên tắc ít quyền nhất, bắt buộc xác thực hai lớp (2FA) cho tài khoản quản trị. Audit log bất biến cho các thay đổi quan trọng.
* **Kiểm thử bảo mật:** Không còn lỗ hổng mức High hoặc Critical trước khi phát hành. Thực hiện kiểm thử xâm nhập định kỳ và sau các thay đổi lớn.

**Khả năng sử dụng**

* **Luồng thao tác:** Người dùng mới có thể tìm sản phẩm, thêm vào giỏ và thanh toán trong thời gian dưới 2 phút. Quy trình checkout giới hạn tối đa 3 bước, có tự động gợi ý địa chỉ và phương thức vận chuyển. Giỏ hàng và thông tin người dùng được lưu khi tải lại trang.
* **Đo lường theo NN Group:**
  + Tính học được: Người mới hoàn thành lần đầu trong dưới 2 phút.
  + Hiệu quả: Người quen hoàn tất đơn hàng lặp lại trong dưới 60 giây.
  + Ghi nhớ: Sau 2 tuần không dùng, tỷ lệ thao tác thành công đạt ít nhất 95 phần trăm.
  + Lỗi: Tỷ lệ lỗi nhập thẻ thấp hơn 10 phần trăm.
  + Hài lòng: Điểm đánh giá trung bình sau giao dịch đạt từ 4 trên 5 trở lên.
* **Giám sát:** Theo dõi thời gian hoàn tất tác vụ end-to-end và cảnh báo khi thời gian trung bình p95 vượt 2 phút hoặc tỷ lệ lỗi cao hơn 10 phần trăm.

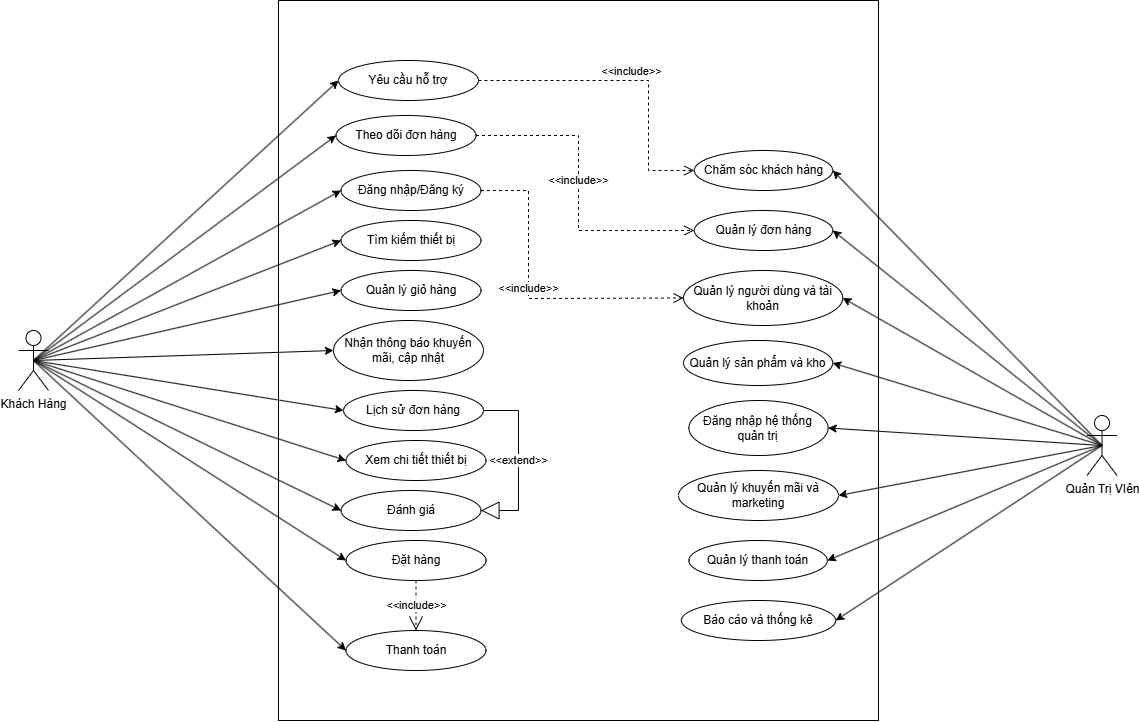
**Quan sát và Vận hành**

* **Giám sát:** Áp dụng mô hình RED (request rate, error rate, duration) cho dịch vụ web và USE (utilization, saturation, errors) cho hạ tầng. Log tập trung dạng JSON có requestId, lưu ít nhất 30 ngày cho vận hành và 180 ngày cho bảo mật.
* **Dashboard thời gian thực:** Theo dõi p95, p99 của API chính, tỷ lệ lỗi 4xx và 5xx, độ trễ cơ sở dữ liệu, cache hit ratio, và tỷ lệ callback thành công.
* **Khôi phục thảm họa (Disaster Recovery):** Backup cơ sở dữ liệu hàng ngày, RPO không quá 15 phút và RTO không quá 60 phút. Có bản sao dữ liệu tại vùng dự phòng, kiểm thử khôi phục định kỳ hàng quý để đảm bảo đối soát giao dịch đạt 100 phần trăm sau phục hồi.

**Bảng giải thích thuật ngữ 1: Dành cho phần 2.**

| **Thuật ngữ** | **Tên đầy đủ** | **Giải thích** |
| --- | --- | --- |
| TTFB | Time To First Byte | Thời gian từ lúc người dùng gửi yêu cầu đến khi nhận byte đầu tiên phản hồi từ máy chủ; chỉ số đo hiệu năng phản hồi. |
| p95 / p99 | Percentile 95 / 99 | Chỉ số phần trăm cho biết 95% (hoặc 99%) yêu cầu hoàn tất trong thời gian dưới mức đo được; dùng để đánh giá hiệu suất thực tế. |
| SKU | Stock Keeping Unit | Mã hàng hóa duy nhất để quản lý tồn kho từng loại sản phẩm hoặc biến thể. |
| SLA | Service Level Agreement | Thỏa thuận mức dịch vụ – cam kết về hiệu suất, thời gian hoạt động và chất lượng mà hệ thống phải đáp ứng. |
| MTTR | Mean Time To Recovery | Thời gian trung bình để khôi phục hệ thống sau khi xảy ra sự cố. |
| MTTD | Mean Time To Detect | Thời gian trung bình để phát hiện lỗi hoặc sự cố trong hệ thống. |
| RPO | Recovery Point Objective | Mức dữ liệu tối đa có thể mất (tính theo thời gian) khi xảy ra sự cố – thường đo bằng phút. |
| RTO | Recovery Time Objective | Thời gian tối đa cho phép để khôi phục hệ thống sau sự cố. |
| PCI DSS | Payment Card Industry Data Security Standard | Bộ tiêu chuẩn bảo mật dữ liệu dành cho các hệ thống xử lý thẻ thanh toán, bảo vệ thông tin chủ thẻ. |
| AES-256 | Advanced Encryption Standard 256-bit | Thuật toán mã hóa đối xứng mạnh mẽ dùng để bảo vệ dữ liệu nhạy cảm khi lưu trữ hoặc truyền. |
| bcrypt / argon2 | – | Hai thuật toán băm mật khẩu hiện đại, có cơ chế sinh salt và chi phí tính toán để tăng độ bảo mật. |
| RBAC | Role-Based Access Control | Cơ chế phân quyền dựa trên vai trò – người dùng được cấp quyền theo vai trò của họ trong hệ thống. |
| 2FA | Two-Factor Authentication | Xác thực hai lớp – yêu cầu hai yếu tố (mật khẩu + mã OTP/thiết bị) để đăng nhập an toàn hơn. |
| Feature Flag | Feature Toggle | Cơ chế bật/tắt chức năng trong hệ thống mà không cần triển khai lại mã nguồn. |
| Canary Deployment | – | Triển khai dần tính năng mới cho một phần nhỏ người dùng để kiểm tra ổn định trước khi áp dụng toàn hệ thống. |
| Backoff & Jitter | – | Kỹ thuật trì hoãn và ngẫu nhiên hóa khoảng thời gian khi retry (thử lại) để tránh quá tải hệ thống. |
| Circuit Breaker | – | Cơ chế bảo vệ hệ thống bằng cách tạm ngắt kết nối tới dịch vụ lỗi liên tục, tránh lan truyền sự cố. |
| RED Model | Request, Error, Duration | Mô hình giám sát hiệu năng dịch vụ web dựa trên tần suất yêu cầu, tỷ lệ lỗi và thời lượng phản hồi. |
| USE Model | Utilization, Saturation, Errors | Mô hình giám sát hạ tầng dựa trên mức sử dụng tài nguyên, độ bão hòa và lỗi hệ thống. |
| Cache Hit Ratio | – | Tỷ lệ truy vấn được phục vụ từ bộ nhớ đệm so với tổng truy vấn – càng cao thì hệ thống càng hiệu quả. |
| Idempotency | – | Tính chất đảm bảo rằng lặp lại cùng một yêu cầu nhiều lần vẫn cho cùng một kết quả, tránh trùng giao dịch. |
| CSV | Comma-Separated Values | Định dạng tệp văn bản dùng dấu phẩy phân tách dữ liệu, phổ biến khi nhập/xuất bảng. |
| XLSX | Excel Spreadsheet Format | Định dạng bảng tính Microsoft Excel, dùng để xuất/nhập dữ liệu có cấu trúc. |
| NN Group | Nielsen Norman Group | Tổ chức nghiên cứu UX nổi tiếng, đề xuất các tiêu chuẩn đo lường khả năng sử dụng (usability metrics). |
| End-to-End | – | Toàn bộ luồng thao tác từ đầu đến cuối – ví dụ: từ đặt hàng đến nhận hàng thành công. |
| CDC | Change Data Capture | Cơ chế ghi nhận và đồng bộ thay đổi dữ liệu trong cơ sở dữ liệu theo thời gian thực. |
| Log JSON | JSON Log Format | Dạng log có cấu trúc JSON giúp dễ dàng lưu trữ, truy vấn và phân tích tự động. |

## **2.3. Mô hình hóa**



### **2.3.1. Phân tích tác nhân**

Tác nhân là một thực thể bên ngoài (người dùng hoặc hệ thống khác) tương tác trực tiếp với hệ thống.Nhóm này mô tả luồng nghiệp vụ chính của khách hàng khi tương tác với website, từ tìm kiếm đến đặt hàng

Khách hàng là người dùng cuối của hệ thống — những người truy cập vào website hoặc ứng dụng để mua sắm. Họ có thể tìm kiếm và xem thông tin sản phẩm, thêm sản phẩm vào giỏ hàng, đặt hàng, thanh toán qua các phương thức khác nhau, và theo dõi trạng thái đơn hàng của mình.  
 Ngoài ra, khách hàng cũng có thể đăng ký tài khoản, cập nhật thông tin cá nhân, xem lịch sử mua hàng, và đánh giá sản phẩm sau khi nhận hàng.

Quản trị viên là người chịu trách nhiệm vận hành và giám sát toàn bộ hệ thống.  
 Họ có quyền thêm, sửa, xóa sản phẩm, quản lý danh mục hàng hóa, kiểm soát đơn hàng và thông tin khách hàng. Ngoài ra, quản trị viên còn có thể tạo và quản lý các chương trình khuyến mãi, xử lý yêu cầu đổi trả, và xem báo cáo thống kê để theo dõi hiệu quả kinh doanh. Mục tiêu của họ là đảm bảo hệ thống hoạt động ổn định, dữ liệu chính xác và trải nghiệm của khách hàng luôn tốt nhất.

### **2.3.2. Nhóm chức năng Quản lý Đơn hàng & Hỗ trợ**

Nhóm này bao gồm các chức năng hỗ trợ khách hàng và các chức năng quản trị tương ứng để xử lý các yêu cầu đó.

Lịch sử đơn hàng: Khách hàng có thể xem lại toàn bộ các đơn hàng mà mình đã đặt trước đây. Trong mỗi đơn hàng, hệ thống hiển thị thông tin chi tiết như ngày đặt hàng, tổng tiền, sản phẩm đã mua và trạng thái hiện tại (đang xử lý, đang giao, đã giao hoặc đã hủy). Chức năng này giúp khách hàng dễ dàng theo dõi quá trình mua sắm của mình, kiểm tra lại đơn cũ khi cần đổi trả hoặc khiếu nại, và tạo cảm giác tin cậy khi mua hàng trên hệ thống.

Theo dõi đơn hàng: Khách hàng có thể theo dõi quá trình vận chuyển của từng đơn hàng cụ thể. Hệ thống cung cấp thông tin cập nhật theo từng giai đoạn, từ khi đơn được xác nhận, đóng gói, bàn giao cho đơn vị giao hàng cho đến khi đến tay người nhận. Ngoài ra, khách hàng cũng có thể xem mã vận đơn, tên đơn vị giao hàng và thời gian dự kiến nhận hàng, giúp họ chủ động hơn trong việc nhận hàng và đảm bảo trải nghiệm mua sắm minh bạch, thuận tiện.

Gửi yêu cầu hỗ trợ: Khi gặp vấn đề trong quá trình mua sắm, khách hàng có thể gửi yêu cầu hỗ trợ hoặc khiếu nại thông qua biểu mẫu trên hệ thống. Các yêu cầu này có thể liên quan đến sản phẩm, đơn hàng, thanh toán hoặc vận chuyển. Sau khi gửi, thông tin sẽ được chuyển đến bộ phận quản trị hoặc chăm sóc khách hàng, để họ xem xét và phản hồi trong thời gian sớm nhất. Tính năng này giúp tăng sự tương tác và hài lòng của khách hàng, đồng thời tạo kênh liên hệ trực tiếp giữa người mua và người quản lý hệ thống.

Chăm sóc khách hàng: Quản trị viên là người tiếp nhận và xử lý các yêu cầu hỗ trợ từ phía khách hàng. Họ có thể trả lời thắc mắc, giải quyết khiếu nại, hoặc hướng dẫn khách hàng trong quá trình sử dụng hệ thống. Mục tiêu của chức năng này là đảm bảo mọi phản hồi của khách hàng đều được lắng nghe và xử lý kịp thời, từ đó nâng cao chất lượng dịch vụ và uy tín của hệ thống. Chức năng này có mối liên hệ trực tiếp với phần “Gửi yêu cầu hỗ trợ” của khách hàng.

Quản lý đơn hàng: Quản trị viên có quyền xem và quản lý toàn bộ đơn hàng của khách. Họ có thể xác nhận, cập nhật trạng thái, hoặc hủy đơn hàng khi có vấn đề phát sinh. Ngoài ra, quản trị viên còn có thể xử lý các yêu cầu đổi trả, hoàn tiền, và kiểm tra lịch sử giao dịch để đảm bảo mọi thông tin đều chính xác. Những thay đổi do quản trị viên thực hiện sẽ được hiển thị cho khách hàng trong phần “Theo dõi đơn hàng” và “Lịch sử đơn hàng”, giúp hai bên nắm bắt thông tin đồng nhất và minh bạch.

### **2.3.3. Nhóm chức năng Quản trị Hệ thống**

Đăng nhập hệ thống quản trị: Quản trị viên sử dụng tài khoản và mật khẩu riêng để đăng nhập vào trang quản trị của hệ thống. Sau khi đăng nhập thành công, họ có thể truy cập các chức năng quản lý như sản phẩm, đơn hàng, người dùng, khuyến mãi và báo cáo. Chức năng này giúp đảm bảo chỉ những người có quyền hạn mới được phép thao tác, từ đó bảo vệ an toàn dữ liệu và duy trì hoạt động ổn định của hệ thống.

Quản lý người dùng: Quản trị viên có thể tạo mới, chỉnh sửa, xóa hoặc khóa tài khoản người dùng, bao gồm cả tài khoản khách hàng và các quản trị viên khác. Chức năng này giúp kiểm soát quyền truy cập, ngăn chặn hành vi vi phạm, đồng thời duy trì sự an toàn và minh bạch cho toàn bộ hệ thống. Việc quản lý người dùng chặt chẽ giúp hệ thống vận hành hiệu quả và bảo mật hơn.

Quản lý sản phẩm vào kho: Quản trị viên có thể thêm mới, chỉnh sửa hoặc xóa thông tin sản phẩm. Ngoài ra, họ còn có thể cập nhật số lượng hàng tồn kho, sắp xếp sản phẩm theo danh mục và theo dõi tình trạng nhập – xuất hàng hóa. Nhờ chức năng này, thông tin sản phẩm luôn được hiển thị chính xác cho khách hàng, đồng thời giúp quản trị viên chủ động trong việc điều phối nguồn hàng và kế hoạch kinh doanh.

Quản lý khuyến mãi: Quản trị viên có thể tạo mới và quản lý các chương trình khuyến mãi, như giảm giá trực tiếp, giảm theo phần trăm hoặc cung cấp mã giảm giá (coupon) cho khách hàng. Họ có thể đặt thời gian áp dụng, giới hạn số lần sử dụng và xác định nhóm sản phẩm được hưởng ưu đãi. Chức năng này giúp thu hút khách hàng, thúc đẩy doanh số và hỗ trợ các chiến dịch tiếp thị hiệu quả.

Báo cáo thống kê: Quản trị viên có thể xem và phân tích các báo cáo tổng hợp về hoạt động của hệ thống, bao gồm doanh thu, số lượng đơn hàng, sản phẩm bán chạy và lượng truy cập của khách hàng. Các số liệu này được thể hiện trực quan bằng bảng biểu hoặc đồ thị, giúp quản trị viên dễ dàng theo dõi, đánh giá hiệu quả kinh doanh và đưa ra quyết định phù hợp để cải thiện hiệu suất hoạt động của hệ thống.

# **3. THIẾT KẾ HỆ THỐNG**

Chương này mô tả chi tiết quy trình thiết kế hệ thống dành cho nền tảng quản lý cửa hàng thiết bị điện tử trực tuyến. Quy trình được phân chia thành hai phần chính: thiết kế kiến trúc tổng quát nhằm xác định cấu trúc tổng thể, và thiết kế cụ thể với các biểu đồ UML để minh họa các yếu tố cùng sự tương tác nội bộ.

## **3.1. Thiết kế kiến trú****c tổng thể**

Thiết kế kiến trúc đóng vai trò cốt lõi, định hình cấu trúc cấp cao của hệ thống, bao gồm cách tổ chức và kết nối các thành phần chính. Việc chọn mô hình kiến trúc đúng đắn là yếu tố quyết định, ảnh hưởng lớn đến các tiêu chí phi chức năng như khả năng bảo trì, mở rộng quy mô, hiệu suất vận hành và an ninh.

### **3.1.1. Lựa chọn mô hình kiến trúc**

Dựa vào quy mô dự án và các yêu cầu đã nêu, nhóm em đã đánh giá các mô hình kiến trúc phổ biến như Monolithic (kiến trúc nguyên khối), Microservices, và Three-Tier (kiến trúc ba tầng) để tìm ra lựa chọn phù hợp nhất:

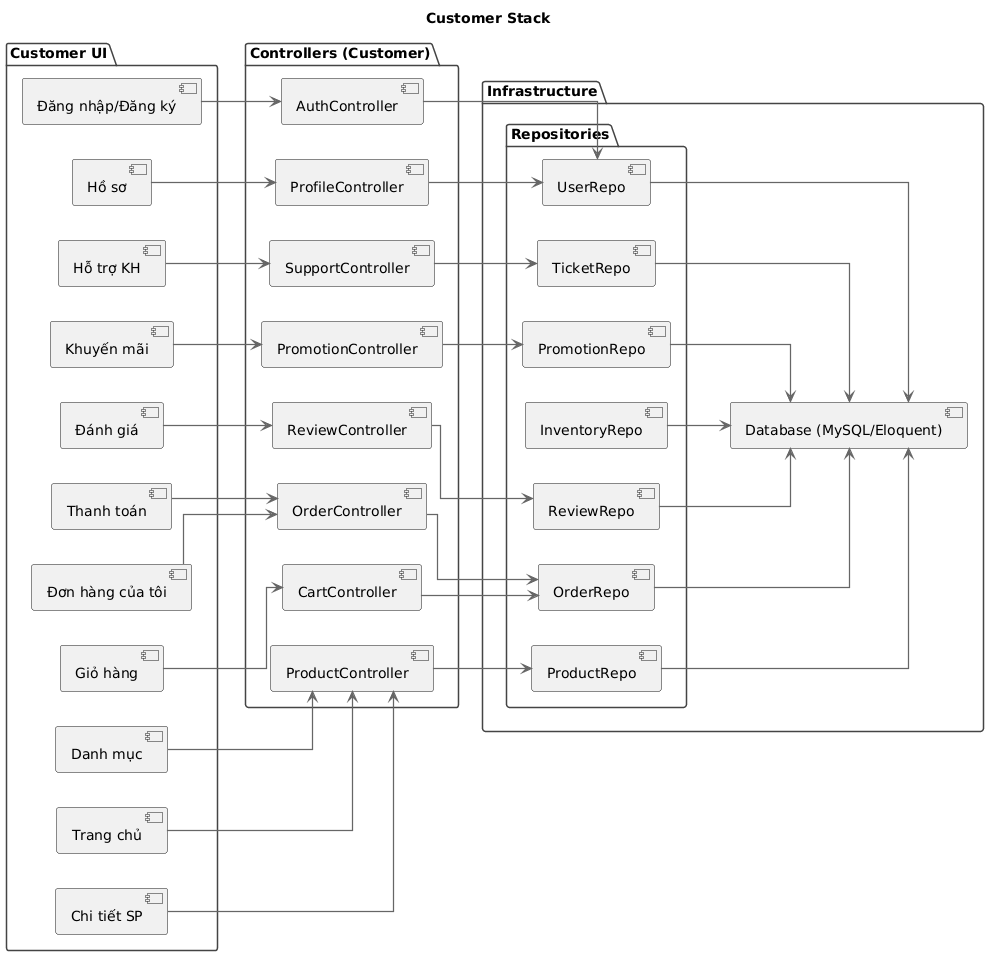
* **Kiến trúc Monolithic**: Toàn bộ ứng dụng được phát triển dưới dạng một khối thống nhất. Mô hình này nổi bật với sự đơn giản trong phát triển và triển khai ban đầu. Tuy nhiên, khi hệ thống trở nên phức tạp, việc bảo trì, khắc phục lỗi hoặc cập nhật một phần nhỏ có thể làm ảnh hưởng toàn bộ, trái với yêu cầu về "Khả năng bảo trì" đã đặt ra.
* **Kiến trúc Three-Tier**: Mô hình này chia ứng dụng thành ba tầng logic riêng biệt: Trình bày, Nghiệp vụ và Dữ liệu. Nó mang lại sự cân bằng tốt giữa tính dễ dàng và khả năng sắp xếp, phù hợp với các ứng dụng web tiêu chuẩn và đáp ứng hiệu quả các yêu cầu dự án.
* **Kiến trúc Microservices**: Hệ thống được tách thành nhiều dịch vụ nhỏ, hoạt động độc lập. Ưu điểm là tính linh hoạt cao và khả năng mở rộng mạnh mẽ. Thế nhưng, việc xây dựng và quản lý Microservices đòi hỏi hạ tầng phức tạp cùng chi phí vận hành lớn, không lý tưởng cho dự án quy mô trung bình với đội ngũ 5 thành viên.

**Lý do lựa chọn kiến trúc Three-Tier:**

* Phân tách rõ ràng Kiến trúc 3-Lớp chia hệ thống thành ba tầng logic độc lập: Tầng Trình bày, Tầng Nghiệp vụ và Tầng Dữ liệu. Sự phân chia này khớp hoàn hảo với yêu cầu "Mã nguồn có cấu trúc" và "Khả năng bảo trì". Nó hỗ trợ các thành viên nhóm làm việc song song trên các tầng khác nhau (ví dụ: một người chuyên về giao diện ở Tầng Trình bày, trong khi những người khác xử lý logic ở Tầng Nghiệp vụ), giảm thiểu tác động lẫn nhau và nâng cao hiệu quả nhóm.
* Khả năng mở rộng linh hoạt: Dù không uyển chuyển như Microservices, kiến trúc 3-Lớp vẫn cho phép mở rộng từng tầng một cách riêng lẻ khi cần. Chẳng hạn, nếu lượng truy cập tăng vọt, hệ thống có thể thêm máy chủ web ở Tầng Trình bày hoặc nhân rộng máy chủ ứng dụng ở Tầng Nghiệp vụ mà không làm gián đoạn các tầng còn lại.
* Tăng cường bảo mật: Việc cách ly Tầng Dữ liệu khỏi truy cập trực tiếp từ phía client (Tầng Trình bày) là nguyên tắc an ninh cơ bản. Mọi truy vấn từ người dùng đều phải qua Tầng Nghiệp vụ để xác thực và kiểm soát quyền hạn trước khi tiếp cận cơ sở dữ liệu. Điều này trực tiếp thỏa mãn yêu cầu "Phân quyền truy cập", giúp ngăn chặn các truy cập không được phép vào dữ liệu quan trọng.
* Tương thích với công nghệ đương đại: Kiến trúc này hòa hợp tốt với các framework web hiện đại. Tầng Trình bày có thể sử dụng ReactJS, còn Tầng Nghiệp vụ có thể áp dụng các công cụ mạnh như Node.js hoặc Laravel – những lựa chọn đã được đề cập trong kế hoạch phát triển.

**3.2. Biểu đồ Gói**

**3.2.1. Customer**



Biểu đồ Customer Stack mô tả toàn bộ cấu trúc và mối liên hệ giữa các thành phần trong hệ thống dành cho người dùng cuối (khách hàng). Tầng Customer UI là lớp giao diện trực quan, cho phép khách hàng thao tác trực tiếp với hệ thống qua các chức năng như: Trang chủ, Danh mục sản phẩm, Chi tiết sản phẩm, Giỏ hàng, Thanh toán, Đơn hàng của tôi, Đánh giá, Khuyến mãi, Hỗ trợ khách hàng, Hồ sơ cá nhân, và Đăng nhập/Đăng ký. Đây là các điểm chạm chính trong hành trình trải nghiệm mua sắm của khách hàng – từ khi tìm hiểu sản phẩm đến khi hoàn tất đơn hàng hoặc gửi phản hồi sau mua.

Các giao diện người dùng này kết nối trực tiếp với tầng Controllers (Customer) – lớp điều khiển chịu trách nhiệm xử lý yêu cầu từ phía người dùng và trung gian truyền dữ liệu giữa giao diện và tầng hạ tầng dữ liệu. Mỗi controller phụ trách một mảng nghiệp vụ cụ thể:

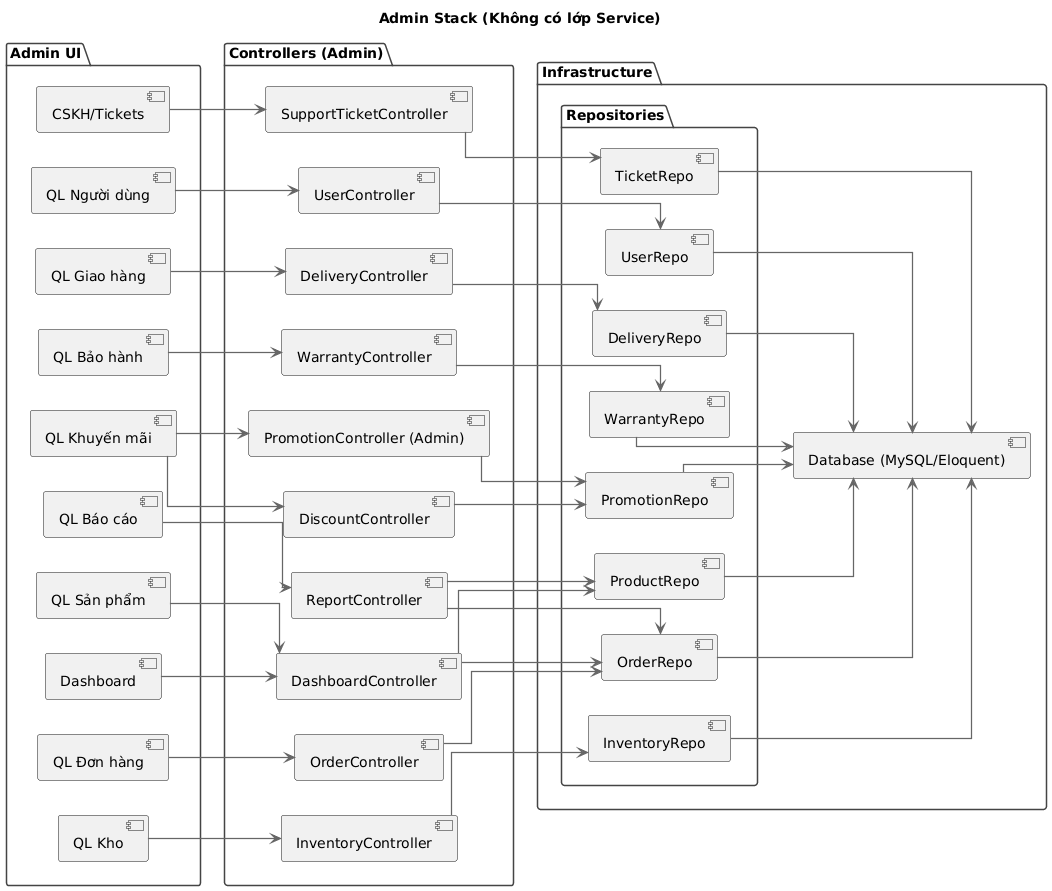
* AuthController đảm nhận chức năng đăng nhập, đăng ký và xác thực người dùng.
* ProfileController quản lý thông tin tài khoản cá nhân.
* SupportController tiếp nhận yêu cầu hỗ trợ và ticket từ khách hàng.
* PromotionController hiển thị và áp dụng chương trình khuyến mãi.
* ReviewController xử lý đánh giá và phản hồi sản phẩm.
* OrderController điều phối quá trình thanh toán và đặt hàng.
* CartController quản lý giỏ hàng và thao tác thêm/xóa sản phẩm.
* ProductController là trung tâm hiển thị thông tin sản phẩm, danh mục và tìm kiếm.

Các controller này không thao tác trực tiếp với cơ sở dữ liệu mà thông qua tầng Infrastructure, cụ thể là nhóm Repositories. Đây là nơi thực hiện các truy vấn, lưu trữ, và cập nhật dữ liệu thông qua mô hình ORM Eloquent trong MySQL Database. Hệ thống gồm nhiều repository tương ứng với từng loại dữ liệu:

* UserRepo quản lý thông tin người dùng, đăng nhập, phân quyền.
* TicketRepo lưu trữ yêu cầu hỗ trợ khách hàng.
* PromotionRepo lưu dữ liệu các chương trình giảm giá và khuyến mãi.
* InventoryRepo quản lý tồn kho và số lượng sản phẩm.
* ReviewRepo lưu nhận xét, đánh giá và phản hồi của khách hàng.
* OrderRepo quản lý đơn hàng, trạng thái vận chuyển và thanh toán.
* ProductRepo truy xuất danh sách sản phẩm, chi tiết, giá bán, và phân loại.

Tất cả các repository đều liên kết đến Database (MySQL/Eloquent) – nơi lưu trữ dữ liệu cốt lõi của hệ thống. Nhờ kiến trúc phân tầng này, Customer Stack đảm bảo sự tách biệt rõ ràng giữa giao diện, điều khiển và dữ liệu, giúp hệ thống dễ mở rộng, bảo trì, và tăng khả năng tái sử dụng mã nguồn. Đồng thời, mô hình này cũng thể hiện quy trình khép kín trong trải nghiệm khách hàng: từ khám phá sản phẩm → chọn mua → thanh toán → đánh giá → nhận hỗ trợ, đảm bảo luồng thông tin được xử lý nhất quán và chính xác trong toàn bộ hành trình mua sắm.

**3.2.2. Admin**



Biểu đồ Admin Stack mô tả cấu trúc hoạt động của hệ thống dành cho người quản trị và nhân viên vận hành. Không giống Customer Stack, Admin Stack được thiết kế đơn giản hơn khi bỏ qua lớp Service trung gian, cho phép các Controller tương tác trực tiếp với tầng Repository để thao tác dữ liệu. Tầng Admin UI bao gồm các module chức năng chính phục vụ công tác quản lý và vận hành hệ thống:

Dashboard, Quản lý sản phẩm, Quản lý kho, Quản lý đơn hàng, Quản lý khuyến mãi, Quản lý giao hàng, Quản lý người dùng, Quản lý bảo hành, Chăm sóc khách hàng (Tickets) và Báo cáo thống kê.

Các chức năng trên được xử lý bởi các Controller tương ứng trong tầng Controllers (Admin):

* Tiếp nhận và phân loại yêu cầu hỗ trợ từ khách hàng để nhân viên CSKH xử lý.
* UserController quản lý tài khoản, vai trò và quyền truy cập của người dùng trong hệ thống.
* DeliveryController đảm nhiệm công tác vận chuyển và đồng bộ với các đơn vị giao hàng.
* WarrantyController quản lý yêu cầu bảo hành, trạng thái và lịch sử sửa chữa.
* PromotionController (Admin) cùng DiscountController cho phép tạo, cập nhật và quản lý các chương trình khuyến mãi hoặc mã giảm giá.
* ReportController tổng hợp dữ liệu, lập báo cáo doanh thu, số lượng đơn hàng, sản phẩm bán chạy.
* DashboardController hiển thị thông tin tổng quan và chỉ số hiệu suất của toàn hệ thống.
* OrderController theo dõi, phê duyệt hoặc hủy đơn hàng của khách hàng.
* InventoryController quản lý số lượng hàng tồn, nhập kho, xuất kho và điều chỉnh tồn.

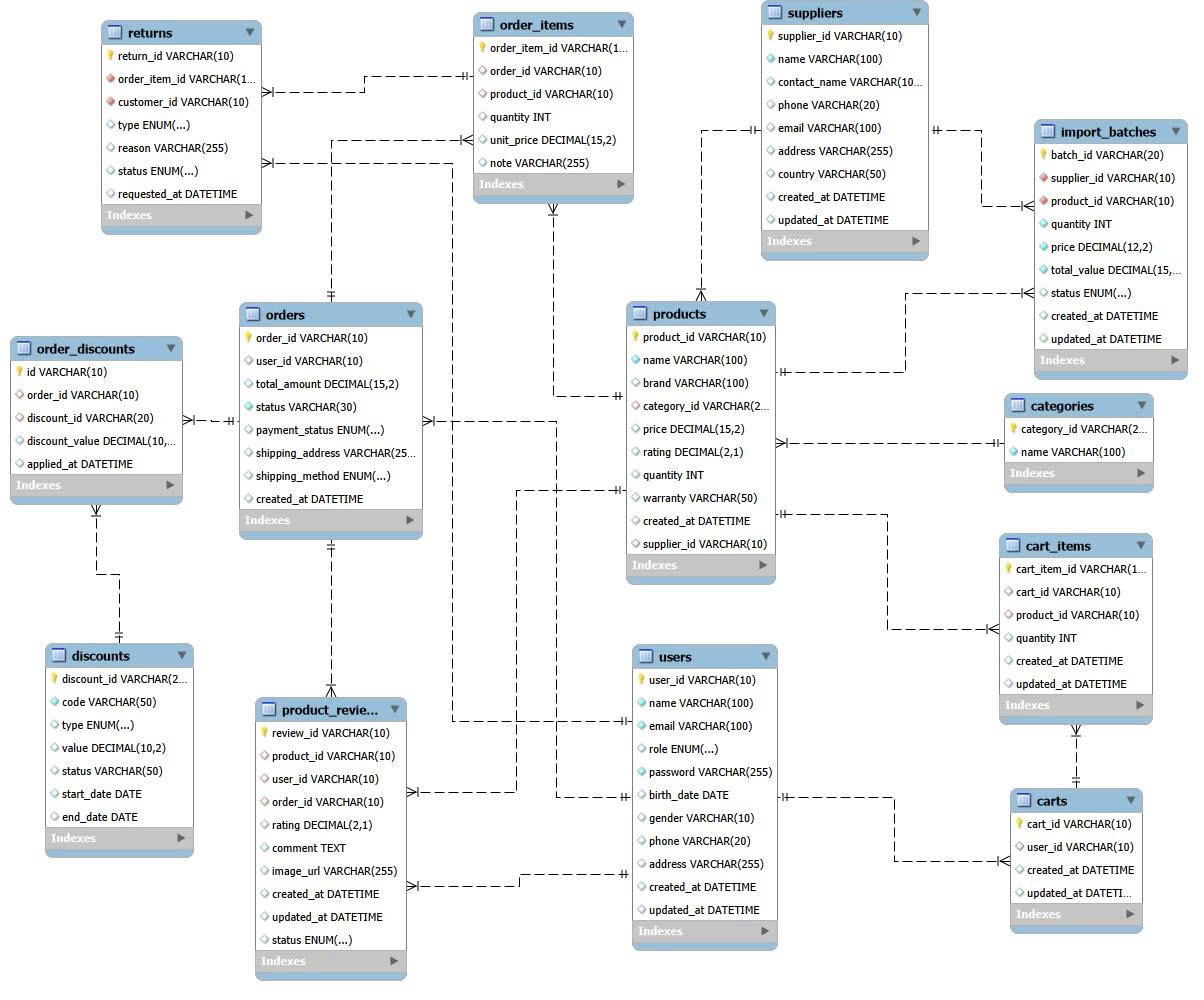
Infrastructure của Admin Stack bao gồm các Repositories tương tác trực tiếp với cơ sở dữ liệu MySQL để lấy và cập nhật thông tin. Cụ thể:

* TicketRepo lưu trữ dữ liệu yêu cầu hỗ trợ khách hàng.
* UserRepo quản lý dữ liệu tài khoản quản trị, nhân viên, và quyền hạn.
* DeliveryRepo lưu thông tin đơn vị giao hàng và trạng thái vận đơn.
* WarrantyRepo quản lý dữ liệu yêu cầu bảo hành và lịch sử xử lý.
* PromotionRepo và ProductRepo lưu thông tin khuyến mãi và sản phẩm.
* OrderRepo quản lý toàn bộ dữ liệu đơn hàng, hóa đơn, và trạng thái xử lý.
* InventoryRepo chịu trách nhiệm quản lý lượng hàng tồn kho.
* Tất cả repository này đều liên kết trực tiếp với Database (MySQL/Eloquent), giúp đồng bộ và cập nhật dữ liệu nhanh chóng giữa hệ thống quản trị và cơ sở dữ liệu trung tâm.

Nhìn tổng thể, Admin Stack thể hiện rõ vai trò của hệ thống quản lý trong việc giám sát, điều phối, và kiểm soát toàn bộ hoạt động vận hành của cửa hàng. Việc loại bỏ lớp Service giúp rút ngắn luồng xử lý, tăng tốc độ truy cập dữ liệu và phù hợp với quy mô quản trị nội bộ. Tuy nhiên, các Controller cần được thiết kế cẩn thận để tránh dồn quá nhiều nghiệp vụ, đảm bảo tính ổn định, dễ bảo trì và tránh vi phạm nguyên tắc “Separation of Concerns”. Cấu trúc này cho phép bộ phận quản trị có thể theo dõi hàng tồn, doanh thu, đơn hàng, khuyến mãi, giao hàng, bảo hành và phản hồi khách hàng một cách tập trung, chính xác và hiệu quả, góp phần đảm bảo hoạt động trơn tru giữa front-end (Customer) và back-office (Admin) trong toàn bộ hệ thống thương mại điện tử.

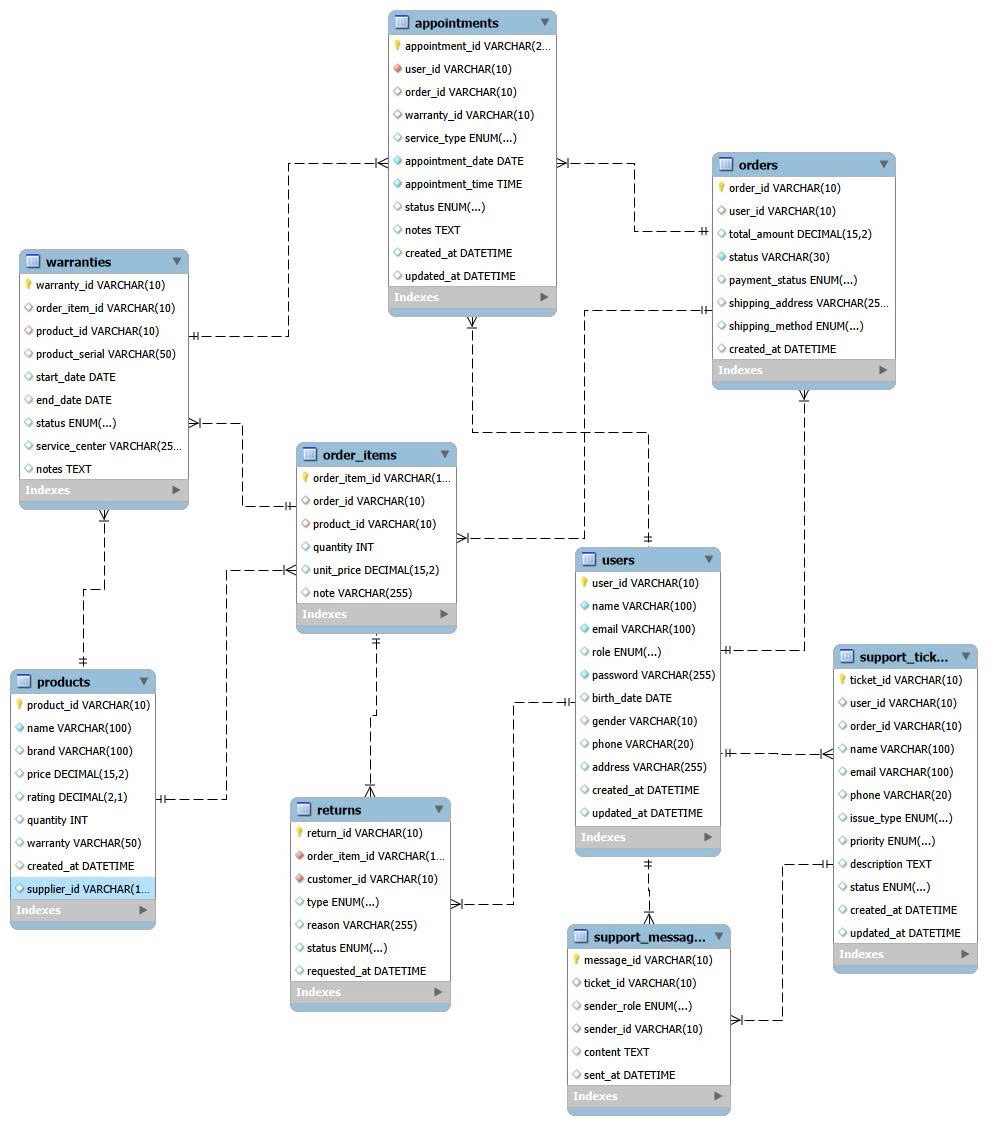
## **3.3. Biểu đồ dữ liệu**

### **3.3.1. Quy trình đặt hàng**



|  |  |  |  |
| --- | --- | --- | --- |
| **Bảng nguồn** | **Liên kết đến** | **Kiểu quan hệ** | **Ý nghĩa** |
| users | carts | 1 - n | Một người dùng có thể có nhiều giỏ hàng |
| carts | cart\_items | 1 - n | Một giỏ hàng chứa nhiều sản phẩm |
| cart\_items | products | n - 1 | Mỗi mục giỏ hàng thuộc về một sản phẩm |
| users | orders | 1 - n | Một người dùng có thể tạo nhiều đơn hàng |
| orders | order\_items | 1 - n | Một đơn hàng có nhiều sản phẩm |
| order\_items | products | n - 1 | Mỗi sản phẩm trong đơn hàng thuộc về một sản phẩm cụ thể |
| orders | order\_discounts | 1 - n | Một đơn hàng có thể áp nhiều mã giảm giá |
| order\_discounts | discounts | n - 1 | Mỗi mã giảm giá có thể dùng trong nhiều đơn |
| orders | returns | 1 - n | Một đơn hàng có thể phát sinh nhiều yêu cầu hoàn trả |
| returns | users | n - 1 | Mỗi yêu cầu hoàn trả thuộc về một người dùng |
| users | product\_reviews | 1 - n | Người dùng có thể đánh giá nhiều sản phẩm |
| product\_reviews | products | n - 1 | Mỗi đánh giá thuộc về một sản phẩm |
| products | categories | n - 1 | Mỗi sản phẩm thuộc một danh mục |
| products | suppliers | n - 1 | Mỗi sản phẩm do một nhà cung cấp cung cấp |
| suppliers | import\_batches | 1 - n | Một nhà cung cấp có nhiều lô hàng nhập |
| import\_batches | products | n - 1 | Mỗi lô hàng nhập liên quan đến một sản phẩm |

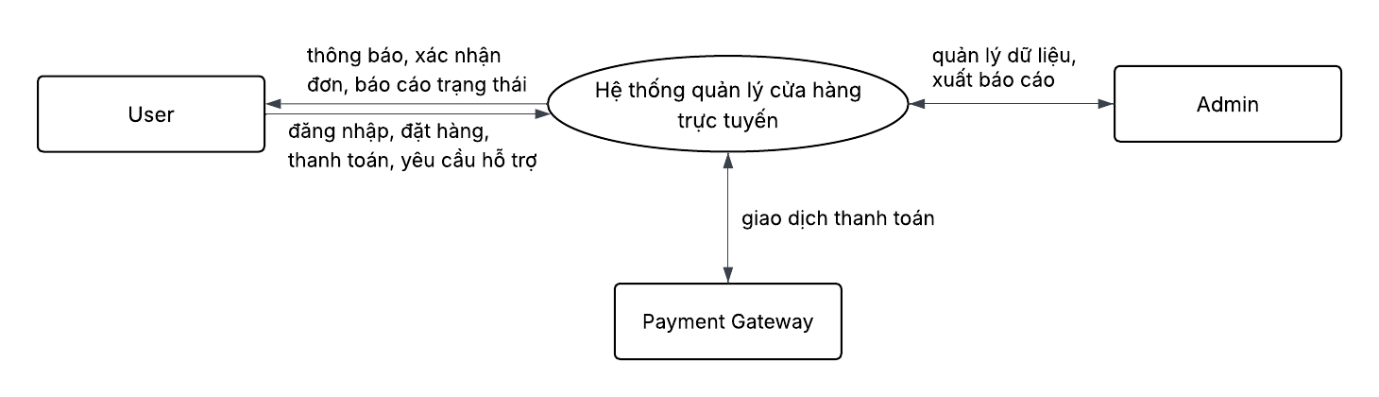
### **3.3.2. Quy trình hỗ trợ**



|  |  |  |  |
| --- | --- | --- | --- |
| **Bảng nguồn** | **Liên kết đến** | **Kiểu quan hệ** | **Ý nghĩa** |
| users | orders | 1 - n | Người dùng có thể có nhiều đơn hàng |
| orders | order\_items | 1 - n | Một đơn hàng gồm nhiều sản phẩm |
| order\_items | products | n - 1 | Mỗi mục đơn hàng thuộc về một sản phẩm |
| order\_items | warranties | 1 - n | Một mục đơn hàng có thể có nhiều bảo hành |
| order\_items | returns | 1 - n | Một sản phẩm trong đơn có thể bị trả hàng |
| warranties | appointments | 1 - n | Một bảo hành có thể tạo nhiều lịch hẹn dịch vụ |
| users | appointments | 1 - n | Người dùng có thể đặt nhiều lịch hẹn sửa chữa/bảo hành |
| orders | appointments | 1 - n | Lịch hẹn gắn liền với đơn hàng cụ thể |
| users | support\_tickets | 1 - n | Người dùng có thể gửi nhiều phiếu hỗ trợ (ticket) |
| orders | support\_tickets | 1 - n | Mỗi ticket có thể liên quan tới một đơn hàng cụ thể |
| support\_tickets | support\_messages | 1 - n | Một ticket có thể chứa nhiều tin nhắn trao đổi giữa người dùng và nhân viên |
| users | support\_messages | 1 - n | Người dùng là người gửi hoặc nhận tin nhắn hỗ trợ |
| products | returns | 1 - n | Sản phẩm có thể bị trả lại trong các yêu cầu hoàn hàng |

**3.4. Mô hình luồng dữ liệu**

**3.4.1. Cấp 0**



**3.4.1.1. Tác nhân**

|  |  |
| --- | --- |
| **Tác nhân** | **Vai trò** |
| User | Thực hiện mua sắm: duyệt SP, thao tác giỏ, tạo đơn, chọn thanh toán và gửi yêu cầu hỗ trợ. |
| Admin | Quản trị vận hành: danh mục, tồn kho, voucher, người dùng, thao tác đơn và yêu cầu báo cáo. |
| Payment Gateway | Nhận yêu cầu khởi tạo giao dịch và trả kết quả giao dịch (return/webhook) |

**3.4.1.2. Hộp đen hệ thống**

|  |  |
| --- | --- |
| **Thành phần** | **Mô tả** |
| Hệ thống quản lý cửa hàng trực tuyến | Nhận dữ liệu từ tác nhân, xử lý nội bộ (ẩn), và trả dữ liệu/ thông tin cần thiết cho tác nhân. |

**3.4.1.3. Luồng dữ liệu vào và ra**

|  |  |  |
| --- | --- | --- |
| **Từ** | **Đến** | **Tên luồng** |
| User | |  | | --- | | Hệ thống |  |  | | --- | |  | | |  | | --- | | Auth&Profile |  |  | | --- | |  | |
| User | Hệ thống | |  | | --- | | CartOps |  |  | | --- | |  | |
| User | Hệ thống | PlaceOrder |
| User | Hệ thống | |  | | --- | | SelectPayment |  |  | | --- | |  | |
| User | Hệ thống | |  | | --- | | Support/Review req |  |  | | --- | |  | |
| Hệ thống | Admin | |  | | --- | | OrderConfirmation / OrderStatus |  |  | | --- | |  | |
| Hệ thống | Payment Gateway | |  | | --- | | Notify/Support resp |  |  | | --- | |  | |
| Admin | Hệ thống | |  | | --- | | Catalog&InventoryOps |  |  | | --- | |  | |
| Admin | Hệ thống | |  | | --- | | DiscountOps |  |  | | --- | |  | |
| Admin | Hệ thống | OrderOps |
| Admin | Hệ thống | ReportReq |
| Hệ thống | Admin | |  | | --- | | ReportOut / OpsResult |  |  | | --- | |  | |
| Hệ thống | Payment Gateway | InitTransaction |
| Payment Gateway | Hệ thống | TxResult(Return/Webhook) |

**3.4.1.4. Bảng đặc tả trao đổi dữ liệu**

|  |  |
| --- | --- |
| **Luồng** | **Ý nghĩa** |
| Auth&Profile | Thông tin xác thực & hồ sơ cơ bản của người dùng gửi vào/nhận về xác nhận |
| |  | | --- | | CartOps |  |  | | --- | |  | | |  | | --- | | Các thao tác thêm/sửa/xóa mục trong giỏ hiện hành |  |  | | --- | |  | |
| PlaceOrder | |  | | --- | | Yêu cầu tạo đơn từ ảnh chụp giỏ (snapshot) tại thời điểm đặt |  |  | | --- | |  | |
| |  | | --- | | SelectPayment |  |  | | --- | |  | | |  | | --- | | Yêu cầu lựa chọn/khởi tạo phương thức thanh toán cho một đơn |  |  | | --- | |  | |
| InitTransaction | |  | | --- | | Hệ thống gửi thông tin giao dịch sang cổng thanh toán |  |  | | --- | |  | |
| TxResult | |  | | --- | | Cổng thanh toán trả trạng thái/kết quả giao dịch về hệ thống |  |  | | --- | |  | |
| |  | | --- | | OrderConfirmation/Status |  |  | | --- | |  | | |  | | --- | | Hệ thống xác nhận đơn mới và thông báo các thay đổi trạng thái |  |  | | --- | |  | |
| |  | | --- | | Catalog&InventoryOps / DiscountOps / OrderOps |  |  | | --- | |  | | |  | | --- | | Yêu cầu vận hành từ Admin gửi vào hệ thống |  |  | | --- | |  | |
| |  | | --- | | ReportReq / ReportOut |  |  | | --- | |  | | |  | | --- | | Admin yêu cầu báo cáo và hệ thống trả về kết quả |  |  | | --- | |  | |
| |  | | --- | | Support/Review req / Notify/Support resp |  |  | | --- | |  | | Yêu cầu hỗ trợ/đánh giá từ User và phản hồi/nhắc nhở từ hệ thống |

**3.4.2. Cấp 1**

**A diagram of a company

AI-generated content may be incorrect.**

**3.4.2.1. Tổng quan**

- Tác nhân: User và Admin.

Khối chức năng chính gắn với kho dữ liệu tương ứng. Sơ đồ cho thấy pipeline chính: Sản phẩm → Giỏ hàng → Đặt hàng → Thanh toán → Quản lý đơn → Báo cáo/Khuyến mãi, kèm các dịch vụ hỗ trợ (*CSKH/Review, Support*).

**3.4.2.2. Bảng chức năng**

**A. Tác nhân: User**

| **Chức năng** | **Mục tiêu chính** | **Dữ liệu chính** | **Luồng vào** | **Luồng ra** |
| --- | --- | --- | --- | --- |
| Quản lý người dùng | Đăng ký, đăng nhập, cập nhật hồ sơ cá nhân | User DB | Thông tin cá nhân (email, mật khẩu, token SSO, họ tên, địa chỉ, điện thoại) | Trạng thái xác thực, hồ sơ cập nhật thành công |
| CSKH & Đánh giá | Gửi yêu cầu hỗ trợ và viết đánh giá sản phẩm | Support DB, Review DB | Ticket hỗ trợ, nội dung đánh giá, số sao, hình ảnh minh họa | Phản hồi CSKH, đánh giá hiển thị sau duyệt |
| Giỏ hàng & Đặt hàng | Lưu sản phẩm chọn mua, tạo và xác nhận đơn hàng | Cart DB, Order DB | Thao tác giỏ hàng (add/update/remove), địa chỉ giao hàng, mã giảm giá | Mã đơn hàng, tổng tiền, trạng thái xác nhận |
| Thanh toán | Xử lý giao dịch thanh toán đơn hàng | Payment DB | orderId, amount, phương thức thanh toán (COD/VNPAY/MOMO) | Biên lai thanh toán, trạng thái giao dịch, mã xác nhận |

B. **Tác nhân**: **Admin**

| **Chức năng** | **Mục tiêu chính** | **Dữ liệu chính** | **Luồng vào** | **Luồng ra** |
| --- | --- | --- | --- | --- |
| Quản lý đơn hàng | Duyệt, thay đổi trạng thái, xử lý hoàn tiền | Order DB | Yêu cầu xác nhận, cập nhật hoặc hoàn tiền | Trạng thái đơn hàng (Packed/Shipped/Cancelled/Refunded) |
| Quản lý sản phẩm & kho | Thêm, sửa, xóa sản phẩm; theo dõi và cập nhật tồn kho | Product DB, Inventory DB | Thông tin sản phẩm, dữ liệu nhập–xuất kho | Danh mục sản phẩm, cảnh báo tồn kho, lịch sử nhập–xuất |
| Khuyến mãi & Báo cáo | Tạo mã giảm giá, theo dõi hiệu quả và xuất báo cáo | Discount DB, Report DB | Cấu hình mã, điều kiện áp dụng, tham số báo cáo | Báo cáo doanh thu, thống kê tồn kho, danh sách khuyến mãi |

## **3.5. Thiết kế chi tiết**

### **3.5.1. Sơ đồ lớp**

**A screenshot of a computer

AI-generated content may be incorrect.**

#### **3.5.1.1. Nhóm Người dùng**

A screenshot of a computer

AI-generated content may be incorrect.

**Mục đích:** Lớp User được thiết kế nhằm mô hình hóa thông tin và hành vi của người dùng trong hệ thống quản lý cửa hàng trực tuyến.

Lớp này đóng vai trò trung tâm trong việc xác thực, quản lý hồ sơ và duy trì trạng thái đăng nhập của người dùng.

Việc phân tích lớp User giúp xác định rõ các thuộc tính dữ liệu cần lưu trữ và phương thức nghiệp vụ cần hỗ trợ, làm cơ sở cho thiết kế cơ sở dữ liệu, API xác thực và các chức năng người dùng trong ứng dụng.

Các thuộc tính:

| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| --- | --- | --- |
| name | String | Họ và tên người dùng hiển thị trong hồ sơ và các hoạt động hệ thống. |
| email | String | Địa chỉ email dùng để đăng nhập và xác thực tài khoản. |
| password | String | Mật khẩu đăng nhập (được mã hóa khi lưu trữ). |
| phone | String | Số điện thoại liên hệ của người dùng. |
| address | String | Địa chỉ mặc định của người dùng (dùng khi giao hàng). |
| status | boolean | Trạng thái tài khoản (đang hoạt động hoặc bị khóa). |
| createdAt | Date | Ngày tạo tài khoản người dùng trong hệ thống. |
| role | String | Vai trò người dùng (ví dụ: “customer”, “admin”). |

Các phương thức:

| **Phương thức** | **Tham số / Đối số** | **Mô tả chức năng** |
| --- | --- | --- |
| createUser(userInfo: Object) | Thông tin người dùng (tên, email, mật khẩu, v.v.) | Tạo tài khoản mới trong hệ thống. |
| login(email: String, password: String) | Thông tin đăng nhập | Xác thực người dùng và khởi tạo phiên đăng nhập. |
| logout() | – | Kết thúc phiên đăng nhập, xóa token hoặc session. |
| changePassword(password: String, newPassword: String) | Mật khẩu cũ và mới | Cho phép người dùng đổi mật khẩu để tăng cường bảo mật. |
| updateProfile(name: String, phone: String, address: String) | Thông tin hồ sơ mới | Cập nhật thông tin cá nhân của người dùng. |

Quan hệ với các lớp khác:

* User – Order: Quan hệ 1–n, một người dùng có thể tạo nhiều đơn hàng.
* User – Cart: Quan hệ 1–1, mỗi người dùng có một giỏ hàng đang hoạt động.
* User – ProductReview: Quan hệ 1–n, người dùng có thể viết nhiều đánh giá sản phẩm.
* User – SupportRequest: Quan hệ 1–n, người dùng có thể gửi nhiều yêu cầu hỗ trợ.

**Kết luận:**

Lớp User là nền tảng của hệ thống, chịu trách nhiệm quản lý danh tính, vai trò và hành vi của người dùng.

Thông qua lớp này, hệ thống có thể thực hiện đăng ký, đăng nhập, bảo mật tài khoản và quản lý hồ sơ cá nhân một cách hiệu quả.

Việc phân tích lớp User giúp đảm bảo dữ liệu người dùng được lưu trữ nhất quán, an toàn và dễ dàng mở rộng cho các tính năng quản trị, thống kê hoặc hỗ trợ khách hàng trong tương lai.

#### **3.5.1.2. Nhóm khách hàng**

A screenshot of a computer program

AI-generated content may be incorrect.

**Mục đích:**

Lớp Customer được xây dựng nhằm mô hình hóa người dùng phía khách hàng trong hệ thống thương mại điện tử, thể hiện các thông tin, hành vi và mối quan hệ giữa khách hàng với các thực thể khác như *Product*, *Cart*, *Order* và *Review*.

Lớp này cho phép khách hàng thực hiện toàn bộ chu trình mua sắm trực tuyến – từ tìm kiếm sản phẩm, thêm vào giỏ hàng, đặt hàng, thanh toán, theo dõi đơn đến đánh giá và nhận hỗ trợ.

Việc phân tích lớp Customer giúp xác định rõ thuộc tính dữ liệu cần lưu, phương thức nghiệp vụ cần hỗ trợ, và mối liên kết với các bảng dữ liệu trong ERD, bảo đảm tính đồng nhất giữa mô hình hướng đối tượng và mô hình quan hệ .

**Các thuộc tính:**

| **Thuộc tính** | **Kiểu dữ liệu** | **Liên kết trong CSDL (ERD)** | **Mô tả** |
| --- | --- | --- | --- |
| customerID | Int | Khóa chính tương ứng users.user\_id | Định danh duy nhất của khách hàng trong hệ thống. |
| addresses | List<String> | Trích từ users.address hoặc bảng phụ shipping\_address (nếu tách riêng) | Danh sách các địa chỉ giao hàng đã lưu. |
| paymentMethods | List<String> | Liên quan đến orders.payment\_method hoặc bảng trung gian payment\_methods | Các phương thức thanh toán mà khách hàng sử dụng (COD, VNPAY, MoMo...). |

- Các phương thức:

| **Phương thức** | **Tham số / Đối số** | **Mô tả chức năng** |
| --- | --- | --- |
| searchProduct(filters: Object) | Thông tin lọc (danh mục, giá, thương hiệu, từ khóa) | Tìm kiếm sản phẩm theo các tiêu chí mong muốn. |
| viewProductDetail(productID: String) | Mã sản phẩm | Hiển thị chi tiết sản phẩm gồm mô tả, giá, tồn kho và đánh giá. |
| manageCart(action: String, productID: String, quantity: Int) | Hành động (add/update/remove), mã sản phẩm, số lượng | Quản lý giỏ hàng cá nhân: thêm, sửa hoặc xóa sản phẩm. |
| placeOrder(orderInfo: Object) | Thông tin đơn hàng (địa chỉ, phương thức thanh toán, tổng tiền) | Tạo đơn hàng mới dựa trên dữ liệu giỏ hàng hiện tại. |
| trackOrder(orderID: String) | Mã đơn hàng | Theo dõi tiến trình vận chuyển của đơn hàng. |
| viewOrderHistory() | – | Xem lại lịch sử các đơn hàng đã mua. |
| requestSupport(issueInfo: Object) | Thông tin yêu cầu hỗ trợ | Gửi yêu cầu hỗ trợ liên quan đến đơn hàng, thanh toán hoặc sản phẩm. |
| rateProduct(productID: String, rating: Int, comment: String) | Mã sản phẩm, số sao, nhận xét | Đánh giá và bình luận sản phẩm đã mua. |
| receiveNotification(notification: Object) | Thông báo hệ thống | Nhận thông tin cập nhật về đơn hàng, khuyến mãi hoặc phản hồi hỗ trợ. |
| applyDiscountCode(code: String) | Mã giảm giá | Áp dụng mã khuyến mãi hợp lệ vào đơn hàng hiện tại. |

- Quan hệ với các lớp khác:

| **Lớp liên quan** | **Kiểu quan hệ** | **Mô tả** |
| --- | --- | --- |
| Cart | 1 – 1 | Mỗi khách hàng có một giỏ hàng đang hoạt động. |
| Order | 1 – n | Một khách hàng có thể có nhiều đơn hàng khác nhau. |
| ProductReview | 1 – n | Một khách hàng có thể đánh giá nhiều sản phẩm đã mua. |
| SupportRequest | 1 – n | Một khách hàng có thể gửi nhiều yêu cầu hỗ trợ. |
| Discount | n – n | Một khách hàng có thể sử dụng nhiều mã giảm giá khác nhau. |

Kết luận:

Lớp Customer đóng vai trò trung tâm trong hệ thống ở phía người dùng, chịu trách nhiệm quản lý toàn bộ hoạt động của khách hàng — từ lúc tìm kiếm sản phẩm đến sau khi nhận hàng.  
 Việc phân tích lớp Customer giúp mô hình hóa rõ ràng dữ liệu và hành vi của khách hàng, đồng thời tạo nền tảng cho việc thiết kế cơ sở dữ liệu, API và giao diện người dùng.  
Nhờ đó, hệ thống bảo đảm được sự liền mạch trong trải nghiệm mua sắm, đồng thời duy trì tính toàn vẹn dữ liệu và khả năng mở rộng trong tương lai.

#### **3.2.1.3. Nhóm quản trị**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Mục đích:

Lớp Admin được thiết kế để mô hình hóa tác nhân quản trị hệ thống, chịu trách nhiệm quản lý dữ liệu, vận hành và giám sát hoạt động toàn bộ cửa hàng trực tuyến.  
Admin có thể thực hiện các thao tác như quản lý người dùng, sản phẩm, tồn kho, đơn hàng, khuyến mãi, thanh toán và báo cáo.

Việc phân tích lớp Admin giúp xác định dữ liệu cần quản lý và hành vi nghiệp vụ cần hỗ trợ, nhằm bảo đảm hệ thống vận hành ổn định, minh bạch và có thể mở rộng.

Các thuộc tính:

| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| --- | --- | --- |
| adminID | Int | Mã định danh duy nhất của quản trị viên trong hệ thống. |

Các phương thức chính:

Lớp Admin đóng vai trò trung tâm, liên kết với nhiều lớp quản lý chức năng khác nhau. Các nhóm phương thức dưới đây thể hiện nhiệm vụ chính của từng module quản trị.

**1. UserManagement**

| **Phương thức** | **Tham số** | **Mô tả chức năng** |
| --- | --- | --- |
| getCustomerProfile(customerID: Int) | Mã khách hàng | Xem thông tin chi tiết hồ sơ người dùng. |
| updateCustomerProfile(customerID: Int, newProfile: Object) | Mã khách hàng, dữ liệu mới | Cập nhật hồ sơ khách hàng. |
| lockCustomerAccount(userID: Int) | Mã tài khoản | Khóa tạm thời tài khoản khi phát hiện rủi ro. |
| unlockCustomerAccount(userID: Int) | Mã tài khoản | Mở khóa tài khoản khi xác minh hợp lệ. |
| getAdminProfile(adminID: Int) | Mã quản trị | Xem thông tin tài khoản quản trị viên. |
| updateAdminProfile(adminID: Int, newProfile: Object) | Mã quản trị, dữ liệu mới | Cập nhật hồ sơ quản trị viên. |

**2. ProductManagement**

| **Phương thức** | **Tham số** | **Mô tả chức năng** |
| --- | --- | --- |
| addProduct(productInfo: Object) | Dữ liệu sản phẩm | Thêm sản phẩm mới vào hệ thống. |
| updateProduct(productID: String, newInfo: Object) | Mã sản phẩm, dữ liệu mới | Cập nhật thông tin sản phẩm hiện có. |
| deleteProduct(productID: String) | Mã sản phẩm | Xóa sản phẩm khỏi danh mục. |
| getProductDetail(productID: String) | Mã sản phẩm | Xem thông tin chi tiết sản phẩm. |
| searchProduct(criteria: Object) | Điều kiện tìm kiếm | Tìm kiếm sản phẩm theo danh mục, thương hiệu, hoặc tồn kho. |

**3. InventoryManagement**

| **Phương thức** | **Tham số** | **Mô tả chức năng** |
| --- | --- | --- |
| getLowStockProducts() | – | Liệt kê các sản phẩm sắp hết hàng. |
| checkProductAvailability(productID: String) | Mã sản phẩm | Kiểm tra tình trạng tồn kho của sản phẩm. |
| alertLowStock() | – | Gửi cảnh báo tự động khi lượng hàng thấp hơn ngưỡng. |

**4. OrderManagement**

| **Phương thức** | **Tham số** | **Mô tả chức năng** |
| --- | --- | --- |
| viewOrders(status: String) | Trạng thái đơn | Xem danh sách đơn hàng theo trạng thái (mới, đang giao, hoàn tất, hủy). |
| updateOrderStatus(orderID: String, newStatus: String) | Mã đơn, trạng thái mới | Cập nhật trạng thái đơn hàng. |
| processReturnOrder(orderID: String) | Mã đơn | Xử lý yêu cầu hoàn hoặc đổi hàng. |
| assignShipment(orderID: String, shipmentInfo: Object) | Mã đơn, thông tin vận chuyển | Gán đơn hàng cho đơn vị giao hàng. |
| printInvoice(orderID: String) | Mã đơn | In hóa đơn hoặc phiếu giao hàng. |

**5. DiscountManagement**

| **Phương thức** | **Tham số** | **Mô tả chức năng** |
| --- | --- | --- |
| createDiscountCode(discountCode: Discount) | Dữ liệu mã giảm giá | Tạo mã khuyến mãi mới. |
| updateDiscountCode(codeID: String, newInfo: Object) | Mã khuyến mãi, dữ liệu mới | Cập nhật điều kiện hoặc thời gian hiệu lực của mã. |
| deleteDiscountCode(codeID: String) | Mã khuyến mãi | Xóa mã khuyến mãi khỏi hệ thống. |
| listActiveDiscounts() | – | Hiển thị danh sách mã khuyến mãi đang hoạt động. |
| expireDiscount(codeID: String) | Mã khuyến mãi | Kết thúc hiệu lực của mã khuyến mãi. |

**6. CustomerSupportManagement**

| **Phương thức** | **Tham số** | **Mô tả chức năng** |
| --- | --- | --- |
| getRequestList() | – | Xem danh sách yêu cầu hỗ trợ của khách hàng. |
| resolveRequest(requestID: String) | Mã yêu cầu | Xử lý hoặc phản hồi yêu cầu hỗ trợ. |
| moderateReview(reviewID: Int) | Mã đánh giá | Duyệt hoặc ẩn các đánh giá vi phạm quy định. |

**7. ReportManagement**

| **Phương thức** | **Tham số** | **Mô tả chức năng** |
| --- | --- | --- |
| generateSalesReport(timeRange: String) | Khoảng thời gian | Tạo báo cáo doanh thu theo ngày, tháng hoặc quý. |
| generateStockReport() | – | Lập báo cáo tồn kho. |
| generateOrderReport() | – | Thống kê đơn hàng theo trạng thái và kênh. |
| generateCustomerReport(timeRange: String) | Khoảng thời gian | Thống kê hoạt động của khách hàng. |
| exportReport(reportID: String, format: String) | Mã báo cáo, định dạng | Xuất báo cáo ra file PDF, Excel, hoặc CSV. |

**8. Quan hệ với các lớp khác**

| **Lớp liên quan** | **Kiểu quan hệ** | **Mô tả** |
| --- | --- | --- |
| ProductManagement | 1 – 1 | Quản trị viên điều khiển danh mục sản phẩm và thuộc tính liên quan. |
| InventoryManagement | 1 – 1 | Quản trị viên theo dõi và cập nhật tồn kho. |
| OrderManagement | 1 – 1 | Quản trị viên xử lý đơn hàng, giao hàng và hoàn tiền. |
| UserManagement | 1 – 1 | Quản trị viên quản lý tài khoản người dùng và phân quyền. |
| DiscountManagement | 1 – 1 | Quản trị viên tạo, cập nhật và theo dõi các chương trình khuyến mãi. |
| ReportManagement | 1 – 1 | Quản trị viên lập báo cáo thống kê và xuất dữ liệu vận hành. |

Kết luận

Lớp Admin là thành phần lõi trong khối vận hành của hệ thống, đóng vai trò quản trị toàn diện các quy trình nghiệp vụ: từ quản lý sản phẩm, tồn kho, đơn hàng, người dùng cho đến khuyến mãi, thanh toán và báo cáo.

Thông qua lớp này, hệ thống đảm bảo hoạt động ổn định – minh bạch – kiểm soát được, đồng thời hỗ trợ mở rộng linh hoạt khi quy mô dữ liệu hoặc số lượng người dùng tăng.  
Phân tích lớp Admin giúp định nghĩa rõ các trách nhiệm chức năng, làm cơ sở cho việc thiết kế module Admin Portal trong kiến trúc hệ thống thương mại điện tử.

#### **3.5.1.4. Nhóm Catalog & Inventory**

A screenshot of a computer

AI-generated content may be incorrect.

Mục đích

Nhóm lớp Catalog & Inventory được thiết kế để quản lý danh mục sản phẩm và tồn kho trong hệ thống thương mại điện tử.

Các lớp trong nhóm này giúp đảm bảo quy trình từ việc tạo, phân loại, hiển thị, đến theo dõi lượng hàng tồn luôn chính xác và cập nhật theo thời gian thực.

Nhóm lớp bao gồm:

* Product: quản lý thông tin chi tiết từng sản phẩm.
* Category: tổ chức và phân loại sản phẩm theo danh mục cha – con.
* Inventory: kiểm soát số lượng hàng tồn, bổ sung hoặc cảnh báo khi sắp hết hàng.

Việc phân tích nhóm lớp này giúp mô hình hóa chuỗi nghiệp vụ lõi của hệ thống: hiển thị sản phẩm cho khách hàng và đảm bảo dữ liệu tồn kho chính xác cho quản trị viên.

**1. Product**

Các thuộc tính

| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| --- | --- | --- |
| productID | String | Mã định danh duy nhất của sản phẩm. |
| name | String | Tên sản phẩm hiển thị trong hệ thống. |
| description | String | Mô tả chi tiết sản phẩm. |
| price | Double | Giá bán hiện tại của sản phẩm. |
| category | String | Mã danh mục mà sản phẩm thuộc về. |
| stockQuantity | Int | Số lượng sản phẩm hiện có trong kho. |
| discount | Int | Phần trăm hoặc giá trị giảm giá (nếu có). |
| images | List<String> | Danh sách các đường dẫn ảnh của sản phẩm. |
| status | String | Trạng thái sản phẩm (còn hàng, ngừng bán, ẩn,...). |
| warranty | String | Thông tin bảo hành đi kèm (ví dụ: 12 tháng). |

Các phương thức:

| **Phương thức** | **Tham số** | **Mô tả chức năng** |
| --- | --- | --- |
| getProductInfo() | – | Lấy thông tin chi tiết của sản phẩm. |
| updateStock(quantity: Int) | Số lượng mới | Cập nhật tồn kho của sản phẩm. |
| applyDiscount(discountPercent: Int) | Phần trăm giảm giá | Thiết lập hoặc cập nhật chương trình giảm giá cho sản phẩm. |
| checkAvailability() | – | Kiểm tra xem sản phẩm còn hàng hay không. |
| calculateFinalPrice() | – | Tính giá cuối cùng sau khi áp dụng giảm giá. |
| addImage(imageURL: String) | Đường dẫn ảnh | Thêm ảnh minh họa cho sản phẩm. |
| removeImage(imageURL: String) | Đường dẫn ảnh | Xóa ảnh khỏi danh sách ảnh sản phẩm. |

**2. Category**

- Các thuộc tính:

| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| --- | --- | --- |
| categoryID | Int | Mã danh mục duy nhất. |
| name | String | Tên danh mục sản phẩm. |
| description | String | Mô tả ngắn gọn về danh mục. |
| parentCategoryID | Int | Mã danh mục cha (nếu là danh mục con). |
| status | String | Trạng thái hoạt động của danh mục (kích hoạt, ẩn,...). |
| imageURL | String | Ảnh minh họa đại diện cho danh mục. |
| createdDate | Date | Ngày tạo danh mục. |
| updatedDate | Date | Ngày cập nhật danh mục gần nhất. |

- Các phương thức:

| **Phương thức** | **Tham số** | **Mô tả chức năng** |
| --- | --- | --- |
| createCategory(name: String, description: String, parentCategoryID: Int, imageURL: String) | Thông tin danh mục | Tạo mới danh mục sản phẩm. |
| updateCategory(categoryID: Int, name: String, description: String, status: String, imageURL: String) | Thông tin cập nhật | Chỉnh sửa thông tin danh mục. |
| deleteCategory(categoryID: Int) | Mã danh mục | Xóa danh mục khỏi hệ thống. |
| listSubCategories(parentCategoryID: Int) | Mã danh mục cha | Liệt kê các danh mục con trực thuộc. |
| listProducts(categoryID: Int) | Mã danh mục | Lấy danh sách sản phẩm thuộc danh mục đó. |
| searchCategory(keyword: String) | Từ khóa tìm kiếm | Tìm danh mục theo tên hoặc mô tả. |

**3. Lớp Inventory**

- Các thuộc tính:

| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| --- | --- | --- |
| inventoryID | String | Mã kho hoặc dòng ghi tồn kho. |
| product | Product | Đối tượng sản phẩm liên quan đến tồn kho. |
| quantity | Int | Số lượng hàng tồn thực tế. |
| lastUpdated | Date | Thời điểm cập nhật gần nhất. |

- Các phương thức:

| **Phương thức** | **Tham số** | **Mô tả chức năng** |
| --- | --- | --- |
| increaseStock(product: Product, quantity: Int) | Sản phẩm, số lượng | Tăng lượng hàng trong kho (nhập hàng mới). |
| decreaseStock(product: Product, quantity: Int) | Sản phẩm, số lượng | Giảm lượng hàng khi bán hoặc hủy. |
| alertLowStock(productID: String) | Mã sản phẩm | Gửi cảnh báo khi sản phẩm sắp hết hàng. |
| checkStock(productID: String) | Mã sản phẩm | Kiểm tra số lượng tồn kho hiện tại. |

- Quan hệ giữa các lớp:

| **Lớp nguồn** | **Lớp đích** | **Kiểu quan hệ** | **Mô tả** |
| --- | --- | --- | --- |
| Product | Category | N–1 (*belongsTo*) | Mỗi sản phẩm thuộc về một danh mục. |
| Category | Product | 1–N | Một danh mục có thể chứa nhiều sản phẩm. |
| Inventory | Product | 1–1 | Mỗi bản ghi tồn kho gắn với đúng một sản phẩm. |

Kết luận:

Nhóm lớp Catalog & Inventory đảm nhiệm chức năng cốt lõi trong quản lý hàng hóa của hệ thống.

* Lớp Product lưu trữ và xử lý thông tin chi tiết sản phẩm.
* Lớp Category tổ chức phân cấp danh mục, giúp việc tìm kiếm và hiển thị dễ dàng hơn.
* Lớp Inventory đảm bảo dữ liệu tồn kho được cập nhật chính xác và kịp thời.

Phân tích nhóm lớp này cho phép hệ thống duy trì thông tin sản phẩm nhất quán giữa giao diện khách hàng và module quản trị, đồng thời hỗ trợ việc theo dõi, nhập hàng, và tối ưu tồn kho một cách hiệu quả.

#### **3.5.1.5. Nhóm Cart**

A screenshot of a computer

AI-generated content may be incorrect.

Mục đích:

Nhóm lớp Cart được thiết kế nhằm mô hình hóa toàn bộ quy trình giỏ hàng của khách hàng trong hệ thống thương mại điện tử.

Cụ thể, các lớp trong nhóm này giúp lưu trữ và xử lý các thông tin về sản phẩm mà khách hàng đã chọn, tính toán tổng tiền, áp dụng mã giảm giá, và chuẩn bị dữ liệu để tiến hành đặt hàng.  
Nhóm bao gồm:

* Customer: đại diện cho người dùng mua sắm.
* Cart: chứa các sản phẩm tạm thời mà khách hàng muốn mua.
* CartItem: mô tả chi tiết từng mặt hàng trong giỏ.
* Product: tham chiếu thông tin sản phẩm từ danh mục chung.

Việc phân tích nhóm lớp Cart giúp đảm bảo quy trình mua hàng diễn ra chính xác, tiện lợi và dữ liệu được đồng bộ với các phần như Order, Payment và Inventory

**1. Customer**

- Các thuộc tính:

| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| --- | --- | --- |
| customerID | Int | Mã định danh duy nhất của khách hàng. |
| addresses | List<String> | Danh sách địa chỉ giao hàng của khách hàng. |
| paymentMethods | List<String> | Danh sách phương thức thanh toán mà khách hàng có thể sử dụng. |

- Các phương thức:

| **Phương thức** | **Tham số / Đối số** | **Mô tả chức năng** |
| --- | --- | --- |
| searchProduct(filters: Object) | Điều kiện lọc sản phẩm | Tìm kiếm sản phẩm theo danh mục, giá hoặc thương hiệu. |
| viewProductDetail(productID: String) | Mã sản phẩm | Hiển thị chi tiết sản phẩm. |
| manageCart(action: String, productID: String, quantity: Int) | Hành động, mã sản phẩm, số lượng | Thêm, cập nhật hoặc xóa sản phẩm trong giỏ hàng. |
| placeOrder(orderInfo: Object) | Thông tin đơn hàng | Tạo đơn hàng từ dữ liệu giỏ hàng hiện tại. |
| applyDiscountCode(code: String) | Mã khuyến mãi | Áp dụng mã giảm giá vào giỏ hàng. |

**2. Cart**

- Các thuộc tính:

| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| --- | --- | --- |
| cartID | Int | Mã định danh giỏ hàng. |
| customerID | Int | Mã khách hàng sở hữu giỏ hàng. |

- Các phương thức:

| **Phương thức** | **Tham số / Đối số** | **Mô tả chức năng** |
| --- | --- | --- |
| addItem(productID: Int, quantity: Int) | Mã sản phẩm, số lượng | Thêm sản phẩm vào giỏ hàng. |
| updateItem(productID: Int, quantity: Int) | Mã sản phẩm, số lượng | Cập nhật số lượng sản phẩm trong giỏ. |
| getItems() | – | Lấy danh sách các sản phẩm hiện có trong giỏ. |
| getItemCount() | – | Đếm số lượng mặt hàng trong giỏ. |
| calculateTotal() | – | Tính tổng giá trị giỏ hàng. |
| applyPromotion(promoID: String) | Mã khuyến mãi | Áp dụng chương trình khuyến mãi. |
| clearCart() | – | Xóa toàn bộ sản phẩm trong giỏ hàng. |
| saveCart() | – | Lưu lại trạng thái giỏ hàng hiện tại. |
| restoreCart(cartID: Int) | Mã giỏ hàng | Khôi phục giỏ hàng đã lưu. |

**3. CartItem**

- Các thuộc tính:

| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| --- | --- | --- |
| cartItemID | Int | Mã định danh của sản phẩm trong giỏ. |
| cartID | Int | Mã giỏ hàng mà sản phẩm thuộc về. |
| product | Product | Đối tượng sản phẩm được thêm vào giỏ. |
| addedDate | Date | Ngày sản phẩm được thêm vào giỏ hàng. |

- Các phương thức:

| **Phương thức** | **Tham số / Đối số** | **Mô tả chức năng** |
| --- | --- | --- |
| updateQuantity(newQuantity: Int) | Số lượng mới | Thay đổi số lượng sản phẩm trong giỏ hàng. |
| calculateSubtotal() | – | Tính giá tạm thời cho sản phẩm (đơn giá × số lượng). |
| getCartItemInfo() | – | Lấy thông tin chi tiết sản phẩm trong giỏ. |

**4. Product**

- Các thuộc tính

| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| --- | --- | --- |
| productID | String | Mã định danh của sản phẩm. |
| name | String | Tên sản phẩm hiển thị. |
| price | Double | Giá sản phẩm. |
| stockQuantity | Int | Số lượng tồn kho hiện tại. |
| status | String | Trạng thái kinh doanh (Đang bán/Ngừng bán). |

- Các phương thức chính:

| **Phương thức** | **Mô tả chức năng** |
| --- | --- |
| getProductInfo() | Trả về thông tin chi tiết của sản phẩm. |
| updateStock(quantity: Int) | Cập nhật tồn kho khi có thay đổi. |
| checkAvailability() | Kiểm tra xem sản phẩm còn hàng không. |
| calculateFinalPrice() | Tính giá sau giảm giá (nếu có). |

- Quan hệ giữa các lớp

| **Lớp nguồn** | **Lớp đích** | **Kiểu quan hệ** | **Mô tả** |
| --- | --- | --- | --- |
| Customer | Cart | 1 – 1 (*owns*) | Mỗi khách hàng chỉ có một giỏ hàng đang hoạt động. |
| Cart | CartItem | 1 – N (*contains*) | Một giỏ hàng có thể chứa nhiều sản phẩm. |
| CartItem | Product | N – 1 (*refers*) | Mỗi mục trong giỏ hàng tham chiếu đến một sản phẩm cụ thể. |

Kết luận

Nhóm lớp Cart mô tả đầy đủ quá trình tương tác giữa khách hàng và giỏ hàng trong hệ thống thương mại điện tử.

Các lớp được phân chia rõ ràng theo trách nhiệm:

* Customer chịu trách nhiệm thao tác và khởi tạo hành động.
* Cart quản lý tập hợp các sản phẩm.
* CartItem lưu chi tiết từng mặt hàng.
* Product cung cấp dữ liệu gốc cho hiển thị và tính toán.

Phân tích nhóm lớp này giúp hệ thống đảm bảo tính chính xác của dữ liệu trong quá trình mua sắm, đồng thời hỗ trợ việc tích hợp thuận lợi với các module Order, Payment và Inventory, hình thành luồng nghiệp vụ thống nhất từ “thêm vào giỏ” đến “thanh toán và xác nhận đơn hàng”.

#### **3.5.1.6. Nhóm Order & Discount**

A screenshot of a computer

AI-generated content may be incorrect.

**1. Customer**

- Các thuộc tính:

| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| --- | --- | --- |
| customerID | Int | Mã định danh duy nhất của khách hàng. |
| addresses | List<String> | Danh sách địa chỉ giao hàng. |
| paymentMethods | List<String> | Các phương thức thanh toán mà khách hàng có thể sử dụng. |

- Các phương thức:

| **Phương thức** | **Tham số / Đối số** | **Mô tả chức năng** |
| --- | --- | --- |
| placeOrder(orderInfo: Object) | Thông tin đơn hàng | Gửi yêu cầu tạo đơn hàng mới. |
| trackOrder(orderID: String) | Mã đơn hàng | Theo dõi trạng thái và quá trình giao hàng. |
| applyDiscountCode(code: String) | Mã giảm giá | Áp dụng mã khuyến mãi vào đơn hàng. |
| viewOrderHistory() | – | Xem lại danh sách đơn hàng đã đặt. |

**2. Order**

**- Các thuộc tính:**

| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| --- | --- | --- |
| orderID | String | Mã đơn hàng duy nhất trong hệ thống. |
| customerID | Int | Mã khách hàng đã đặt đơn. |
| orderItems | List<OrderItem> | Danh sách các sản phẩm trong đơn hàng. |
| totalAmount | Double | Tổng giá trị đơn hàng sau khi áp dụng giảm giá. |
| status | String | Trạng thái hiện tại của đơn (đã đặt, đang xử lý, đã giao, hủy,...). |
| paymentMethod | String | Phương thức thanh toán được chọn. |
| shippingMethod | String | Hình thức vận chuyển. |
| createdAt | Date | Thời điểm tạo đơn hàng. |
| updatedAt | Date | Thời điểm cập nhật gần nhất. |

**- Các phương thức:**

| **Phương thức** | **Tham số / Đối số** | **Mô tả chức năng** |
| --- | --- | --- |
| updateStatus(newStatus: String) | Trạng thái mới | Cập nhật trạng thái của đơn hàng. |
| calculateTotal() | – | Tính tổng giá trị đơn hàng dựa trên các sản phẩm và giảm giá. |
| cancelOrder() | – | Hủy đơn hàng khi chưa xử lý. |
| processPayment() | – | Xác nhận thanh toán cho đơn hàng. |
| refundOrder() | – | Thực hiện hoàn tiền cho đơn hàng bị hủy hoặc trả hàng. |
| trackShipment() | – | Theo dõi quá trình vận chuyển của đơn hàng. |

**3. OrderItem**

- Các thuộc tính:

| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| --- | --- | --- |
| orderItemID | Int | Mã định danh của mục chi tiết đơn hàng. |
| orderID | String | Mã đơn hàng chứa sản phẩm này. |
| product | Product | Đối tượng sản phẩm liên quan. |
| quantity | Int | Số lượng sản phẩm được đặt mua. |
| unitPrice | Double | Đơn giá tại thời điểm đặt hàng. |

- Các phương thức:

| **Phương thức** | **Tham số / Đối số** | **Mô tả chức năng** |
| --- | --- | --- |
| calculateSubtotal() | – | Tính tổng tiền tạm tính của sản phẩm (số lượng × đơn giá). |
| getOrderItemInfo() | – | Lấy thông tin chi tiết về sản phẩm trong đơn hàng. |

**4. Product**

- Các thuộc tính:

| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| --- | --- | --- |
| productID | String | Mã sản phẩm duy nhất. |
| name | String | Tên sản phẩm. |
| price | Double | Giá sản phẩm. |
| stockQuantity | Int | Số lượng tồn kho hiện có. |
| status | String | Trạng thái kinh doanh (Đang bán/Ngừng bán). |

- Các phương thức:

| **Phương thức** | **Mô tả chức năng** |
| --- | --- |
| getProductInfo() | Lấy thông tin sản phẩm. |
| updateStock(quantity: Int) | Cập nhật số lượng tồn kho sau mỗi đơn hàng. |
| checkAvailability() | Kiểm tra xem sản phẩm còn hàng hay không. |
| calculateFinalPrice() | Tính giá sau khi áp dụng giảm giá. |

**5. Discount**

- Các thuộc tính:

| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| --- | --- | --- |
| discountID | String | Mã khuyến mãi duy nhất. |
| code | String | Mã giảm giá do khách hàng nhập. |
| discountType | String | Loại khuyến mãi (theo % hoặc theo giá trị). |
| maxDiscount | Double | Mức giảm tối đa được áp dụng. |
| usageLimit | Int | Số lần mã có thể được sử dụng. |
| startDate | Date | Ngày bắt đầu hiệu lực. |
| endDate | Date | Ngày hết hạn sử dụng. |
| status | String | Trạng thái của mã (đang hoạt động, hết hạn, ẩn,...). |

- Các phương thức:

| **Phương thức** | **Tham số / Đối số** | **Mô tả chức năng** |
| --- | --- | --- |
| validatePromotionCode(code: String) | Mã khuyến mãi | Kiểm tra tính hợp lệ của mã giảm giá. |
| applyDiscount(orderAmount: Double) | Tổng tiền đơn hàng | Tính số tiền giảm dựa trên giá trị đơn. |
| isExpired() | – | Kiểm tra xem mã khuyến mãi đã hết hạn hay chưa. |
| updateDiscountStatus(codeID: String, orderInfo: Object) | Mã khuyến mãi, thông tin đơn | Cập nhật trạng thái sử dụng sau khi áp dụng. |

- Quan hệ giữa các lớp:

| **Lớp nguồn** | **Lớp đích** | **Kiểu quan hệ** | **Mô tả** |
| --- | --- | --- | --- |
| Customer | Order | 1 – N (*places*) | Một khách hàng có thể đặt nhiều đơn hàng. |
| Order | OrderItem | 1 – N (*contains*) | Mỗi đơn hàng có thể chứa nhiều sản phẩm. |
| OrderItem | Product | N – 1 (*refers*) | Mỗi mục đơn hàng liên kết đến một sản phẩm cụ thể. |
| Order | Discount | N – 1 (*applies*) | Mỗi đơn hàng có thể áp dụng một mã giảm giá. |

**Kết luận:**

Nhóm lớp Order & Discount là phần trung tâm của hệ thống xử lý nghiệp vụ, đảm bảo toàn bộ quy trình từ đặt hàng đến áp dụng khuyến mãi và thanh toán được tự động hóa, chính xác và an toàn.

* Lớp Order điều phối quy trình đơn hàng và lưu toàn bộ thông tin giao dịch.
* Lớp OrderItem lưu chi tiết từng mặt hàng cụ thể.
* Lớp Discount cung cấp công cụ khuyến mãi linh hoạt và dễ mở rộng.
* Lớp Customer và Product đóng vai trò cầu nối giúp dữ liệu khách hàng và sản phẩm được đồng bộ hóa.

Việc phân tích nhóm lớp này giúp đảm bảo luồng dữ liệu xuyên suốt từ giỏ hàng đến thanh toán, đồng thời tăng cường khả năng quản lý và kiểm soát chính sách giá – giảm giá trong toàn hệ thống.

#### **3.5.1.7. Nhóm Payment**

A screenshot of a computer program

AI-generated content may be incorrect.

Mục đích

Nhóm lớp Payments mô hình hóa toàn bộ quy trình xử lý thanh toán trong hệ thống thương mại điện tử — từ khi khách hàng đặt đơn hàng, lựa chọn phương thức thanh toán, cho đến khi giao dịch được xử lý qua cổng thanh toán và phản hồi lại kết quả.  
Nhóm này đảm bảo rằng mọi giao dịch tài chính diễn ra an toàn, có thể truy vết và tương thích với nhiều cổng thanh toán khác nhau (VNPAY, SePay, PayPal, v.v.).

Các lớp chính trong nhóm gồm:

* Customer: người khởi tạo đơn hàng cần thanh toán.
* Order: chứa thông tin đơn hàng được gắn với giao dịch thanh toán.
* Payment: đại diện cho giao dịch thanh toán cụ thể.
* PaymentGateway: xử lý giao tiếp với cổng thanh toán bên thứ ba.
* PaymentMethod: lưu và quản lý danh sách các phương thức thanh toán có sẵn.

**Quan hệ giữa các lớp:**

| **Lớp nguồn** | **Lớp đích** | **Kiểu quan hệ** | **Mô tả** |
| --- | --- | --- | --- |
| Customer | Order | 1 – N (*places*) | Mỗi khách hàng có thể tạo nhiều đơn hàng. |
| Order | Payment | 1 – 1 hoặc 1 – N (*has*) | Mỗi đơn hàng gắn với một hoặc nhiều giao dịch thanh toán. |
| Payment | PaymentGateway | N – 1 (*processedBy*) | Mỗi giao dịch được xử lý qua một cổng thanh toán cụ thể. |
| Payment | PaymentMethod | N – 1 (*uses*) | Mỗi giao dịch sử dụng một phương thức thanh toán. |

Kết luận:

Nhóm lớp Payments là thành phần quan trọng bảo đảm tính toàn vẹn và an toàn của giao dịch tài chính trong hệ thống.

* Lớp Payment giữ vai trò trung tâm, kết nối giữa Order, PaymentGateway và PaymentMethod.
* Lớp PaymentGateway chịu trách nhiệm xử lý kỹ thuật với cổng thanh toán bên ngoài, đảm bảo tính xác thực và phản hồi thời gian thực.
* Lớp PaymentMethod giúp hệ thống mở rộng linh hoạt nhiều hình thức thanh toán khác nhau mà không ảnh hưởng đến logic cốt lõi.

Phân tích nhóm lớp này giúp hệ thống đạt được tính bảo mật, mở rộng và tích hợp cao, đồng thời đáp ứng các yêu cầu nghiệp vụ thanh toán, hoàn tiền và kiểm soát giao dịch trong môi trường thương mại điện tử hiện đại. Nhóm em đang trong quá trình hoàn thiện phần này.

#### **3.5.1.8. Nhóm Support, Review & Notification**

**A screenshot of a computer

AI-generated content may be incorrect.**

Mục đích:

Nhóm lớp Support – Reviews – Notifications được xây dựng nhằm mô hình hóa các chức năng hỗ trợ sau bán hàng, tương tác người dùng và truyền thông nội bộ trong hệ thống thương mại điện tử.  
Các lớp này đảm nhiệm ba vai trò then chốt:

* SupportRequest và CustomerSupportManagement: tiếp nhận, quản lý và xử lý yêu cầu hỗ trợ từ khách hàng.
* Review: lưu trữ và hiển thị đánh giá sản phẩm, giúp tăng tính minh bạch và cải thiện trải nghiệm mua sắm.
* Notification: cung cấp cơ chế thông báo tự động đến người dùng (xác nhận đơn, phản hồi hỗ trợ, thông báo khuyến mãi,…).
* Ngoài ra, ReportManagement hỗ trợ tổng hợp và xuất báo cáo liên quan đến hoạt động đánh giá, hỗ trợ hoặc vận hành.

Nhờ đó, nhóm lớp này giúp đảm bảo chu trình dịch vụ khách hàng khép kín, từ tiếp nhận yêu cầu đến xử lý đến phản hồi đến báo cáo hiệu quả.

**1. Notification**

- Các thuộc tính:

| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| --- | --- | --- |
| notificationID | Int | Mã định danh của thông báo. |
| userID | Int | Người nhận thông báo. |
| title | String | Tiêu đề nội dung thông báo. |
| message | String | Nội dung chi tiết thông báo. |
| type | String | Loại thông báo (đơn hàng, hỗ trợ, khuyến mãi, hệ thống). |
| priority | String | Mức độ ưu tiên (cao, trung bình, thấp). |
| isRead | Boolean | Trạng thái đã đọc hay chưa. |
| createdAt | Date | Thời gian tạo thông báo. |

- Các phương thức

| **Phương thức** | **Mô tả chức năng** |
| --- | --- |
| createRequest(customerID, issueType, details) | Khởi tạo yêu cầu hỗ trợ mới. |
| updateRequestStatus(requestID, newStatus) | Cập nhật tiến độ hoặc kết quả xử lý. |
| getRequestByID(requestID) | Truy xuất thông tin chi tiết của yêu cầu. |

**2. Review**

- Các thuộc tính:

| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| --- | --- | --- |
| reviewID | String | Mã định danh đánh giá. |
| productID | String | Mã sản phẩm được đánh giá. |
| customerID | Int | Mã khách hàng viết đánh giá. |
| rating | Int | Số sao đánh giá (1–5). |
| title | String | Tiêu đề đánh giá. |
| comment | String | Nội dung chi tiết của đánh giá. |
| images | List<String> | Danh sách hình ảnh kèm theo. |

- Các phương thức:

| **Phương thức** | **Mô tả chức năng** |
| --- | --- |
| createReview(productID, title, rating, comment) | Gửi đánh giá mới cho sản phẩm. |
| editReview(reviewID, newTitle, newComment, newRating) | Chỉnh sửa nội dung đánh giá. |
| deleteReview(reviewID) | Xóa một đánh giá không còn phù hợp. |

**3. SupportRequest**

- Các thuộc tính:

| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| --- | --- | --- |
| requestID | Int | Mã định danh yêu cầu hỗ trợ. |
| customerID | Int | Mã khách hàng gửi yêu cầu. |
| issueType | String | Loại vấn đề (đơn hàng, thanh toán, bảo hành, tài khoản,…). |
| status | String | Trạng thái xử lý (mới, đang xử lý, đã phản hồi, đóng). |
| createdAt | DateTime | Ngày giờ tiếp nhận yêu cầu. |
| resolvedAt | DateTime | Ngày giờ kết thúc xử lý (nếu có). |

- Các phương thức:

| **Phương thức** | **Mô tả chức năng** |
| --- | --- |
| createRequest(customerID, issueType, details) | Khởi tạo yêu cầu hỗ trợ mới. |
| updateRequestStatus(requestID, newStatus) | Cập nhật tiến độ hoặc kết quả xử lý. |
| getRequestByID(requestID) | Truy xuất thông tin chi tiết của yêu cầu. |

**4. CustomerSupportManagement**

- Các thuộc tính:

| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| --- | --- | --- |
| supportRequests | List<SupportRequest> | Danh sách các yêu cầu hỗ trợ hiện tại. |

- Các phương thức:

| **Phương thức** | **Mô tả chức năng** |
| --- | --- |
| viewRequestList() | Hiển thị danh sách tất cả yêu cầu hỗ trợ. |
| resolveRequest(requestID) | Đánh dấu yêu cầu hỗ trợ đã hoàn tất. |
| moderateReview(reviewID) | Duyệt và kiểm soát nội dung đánh giá người dùng. |
| generateSupportReport() | Tạo báo cáo tổng hợp tình hình hỗ trợ khách hàng. |

**5. ReportManagement**

- Các thuộc tính:

| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| --- | --- | --- |
| reportHistory | List<Report> | Lưu trữ danh sách các báo cáo đã được tạo. |

- Các phương thức:

| **Phương thức** | **Mô tả chức năng** |
| --- | --- |
| generateSalesReport(timeRange) | Lập báo cáo doanh thu theo thời gian. |
| generateSupportReport() | Tạo báo cáo thống kê số lượng và trạng thái hỗ trợ. |
| generateReviewReport() | Báo cáo thống kê đánh giá người dùng. |
| exportReport(reportID, format) | Xuất báo cáo ra định dạng PDF, Excel,… |
| scheduleReport(type, frequency) | Lập lịch tự động tạo báo cáo định kỳ. |

**6. Các quan hệ giữa các lớp**

| **Lớp nguồn** | **Lớp đích** | **Kiểu quan hệ** | | **Mô tả** |
| --- | --- | --- | --- | --- |
| User | Notification | | 1 – N (*sentTo*) | Mỗi người dùng có thể nhận nhiều thông báo. |
| Customer | SupportRequest | | 1 – N (*creates*) | Mỗi khách hàng có thể gửi nhiều yêu cầu hỗ trợ. |
| SupportRequest | CustomerSupportManagement | | N – 1 (*manages*) | Tất cả yêu cầu được quản lý bởi bộ phận hỗ trợ. |
| Review | Product | | N – 1 (*belongsTo*) | Mỗi đánh giá liên kết đến một sản phẩm. |
| ReportManagement | Report | | 1 – N (*manages*) | Bộ phận báo cáo có thể tạo nhiều bản báo cáo định kỳ. |

**Kết luận:**

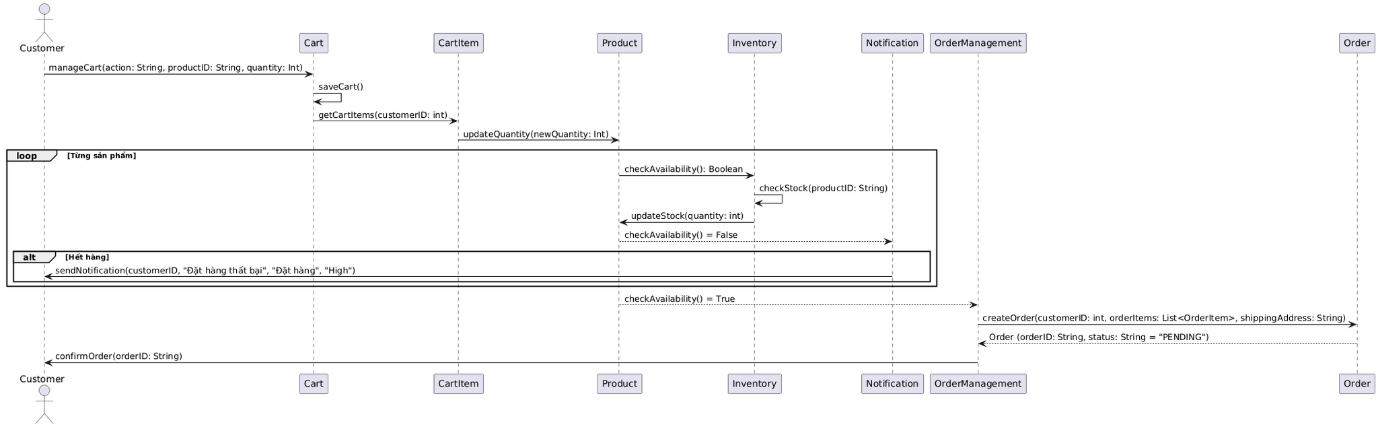
Nhóm lớp Support – Reviews – Notifications là phần hậu cần quan trọng sau bán hàng, đóng vai trò duy trì sự hài lòng và niềm tin của khách hàng.

* SupportRequest và CustomerSupportManagement giúp xử lý vấn đề phát sinh một cách có hệ thống.
* Review tăng tính minh bạch và cải thiện chất lượng dịch vụ.
* Notification đảm bảo khách hàng luôn được cập nhật thông tin kịp thời.
* ReportManagement cung cấp dữ liệu tổng hợp cho nhà quản trị để phân tích hiệu quả hỗ trợ và phản hồi người dùng.

Nhờ đó, hệ thống đạt được chu trình chăm sóc khách hàng khép kín, giúp nâng cao chất lượng dịch vụ, giảm thời gian xử lý sự cố và tăng cường trải nghiệm người dùng trong toàn bộ vòng đời mua sắm.

### **3.5.2. Sơ đồ trình tự**

#### **3.5.2.1. Sơ đồ trình tự trạng thái đặt hàng**



Biểu đồ tuần tự trong hình mô tả quá trình khách hàng thực hiện thao tác đặt hàng trên hệ thống, bắt đầu từ việc quản lý giỏ hàng cho đến khi đơn hàng được tạo thành công.

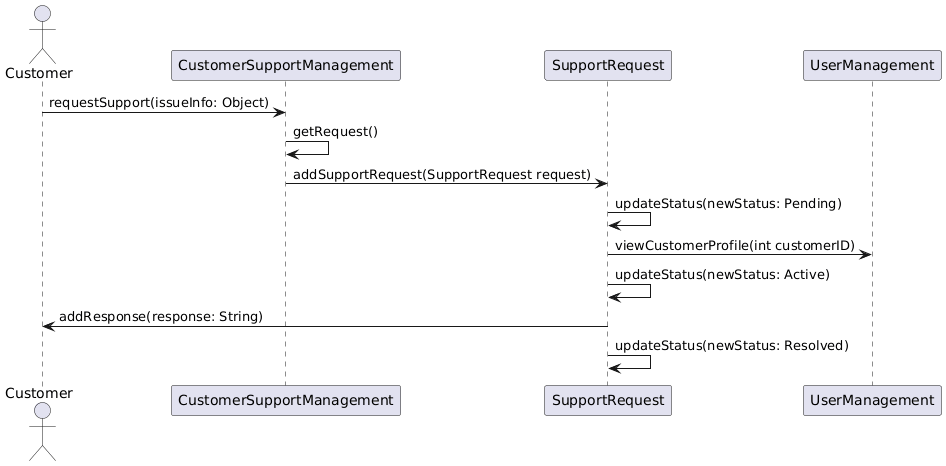
Đầu tiên, khách hàng gửi yêu cầu thêm sản phẩm vào giỏ hàng bằng cách gọi hàm manageCartAction. Hệ thống tiến hành lưu lại giỏ hàng thông qua lớp Cart, đồng thời cập nhật chi tiết mặt hàng trong CartItem, bao gồm mã sản phẩm và số lượng. Sau đó, thông tin về số lượng được gửi đến lớp Product để kiểm tra tồn kho.

Tiếp theo, lớp Product gọi đến lớp Inventory để thực hiện hàm checkAvailability, kiểm tra xem sản phẩm có đủ hàng hay không. Nếu tồn kho không đủ, hệ thống trả về kết quả sai (False) và gửi thông báo lỗi đến khách hàng thông qua lớp Notification, với nội dung như “Hết hàng” hoặc “Số lượng không đủ”.

Ngược lại, nếu sản phẩm còn hàng (True), hệ thống sẽ cập nhật lại số lượng tồn kho bằng phương thức updateStockQuantity, sau đó chuyển sang bước xác nhận đơn hàng. Lúc này, lớp OrderManagement tạo đơn hàng mới với các thông tin cần thiết như mã khách hàng, danh sách sản phẩm, địa chỉ giao hàng và trạng thái ban đầu là “Đang chờ”.

Cuối cùng, sau khi đơn hàng được tạo thành công, khách hàng nhận được thông báo xác nhận từ hệ thống, kết thúc quy trình đặt hàng trong biểu đồ tuần tự này.

#### **3.5.2.2. Sơ đồ trình tự trạng thái hỗ trợ**



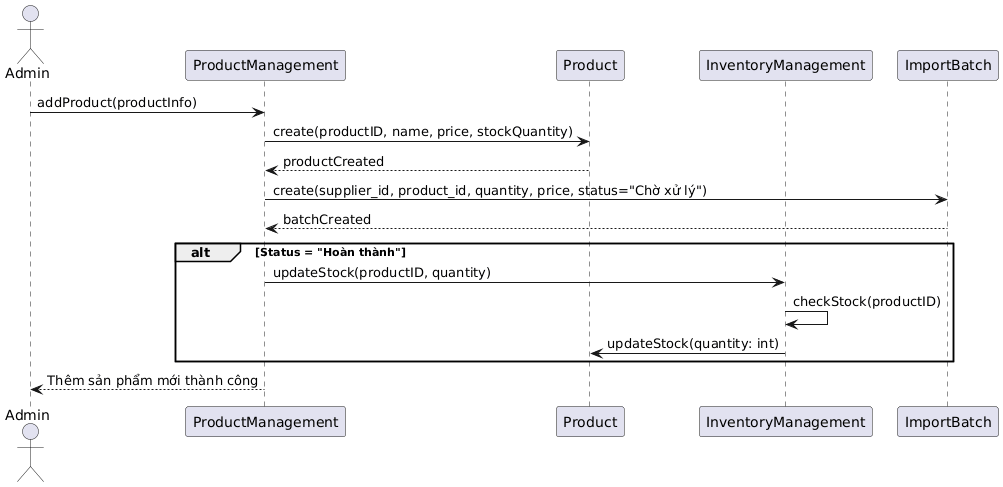
Biểu đồ tuần tự này mô tả quy trình khách hàng gửi yêu cầu hỗ trợ và cách hệ thống tiếp nhận, xử lý cũng như phản hồi lại cho khách hàng.

Đầu tiên, khách hàng gửi yêu cầu hỗ trợ thông qua hàm requestSupport(issueInfo) đến mô-đun CustomerSupportManagement. Mô-đun này tiếp nhận thông tin yêu cầu và tạo một đối tượng SupportRequest mới. Sau đó, nó gọi phương thức addSupportRequest để lưu lại yêu cầu này vào hệ thống. Tiếp theo, lớp SupportRequest được kích hoạt và bắt đầu quá trình xử lý. Nó cập nhật trạng thái yêu cầu sang “Pending” (đang chờ xử lý) bằng phương thức updateStatus. Sau đó, hệ thống gọi đến UserManagement để lấy thông tin chi tiết của khách hàng thông qua phương thức viewCustomerProfile, nhằm hiểu rõ người gửi yêu cầu và ngữ cảnh liên quan.

Sau khi thông tin khách hàng được xem xét, trạng thái của yêu cầu hỗ trợ được chuyển sang “Active” (đang được xử lý). Khi bộ phận hỗ trợ đã giải quyết vấn đề, trạng thái tiếp tục được cập nhật thành “Resolved” (đã xử lý xong).

Cuối cùng, hệ thống gửi phản hồi cho khách hàng thông qua phương thức addResponse(response). Khách hàng nhận được phản hồi này và quy trình hỗ trợ kết thúc, đảm bảo rằng mọi yêu cầu đều được ghi nhận, xử lý và phản hồi rõ ràng trong hệ thống.

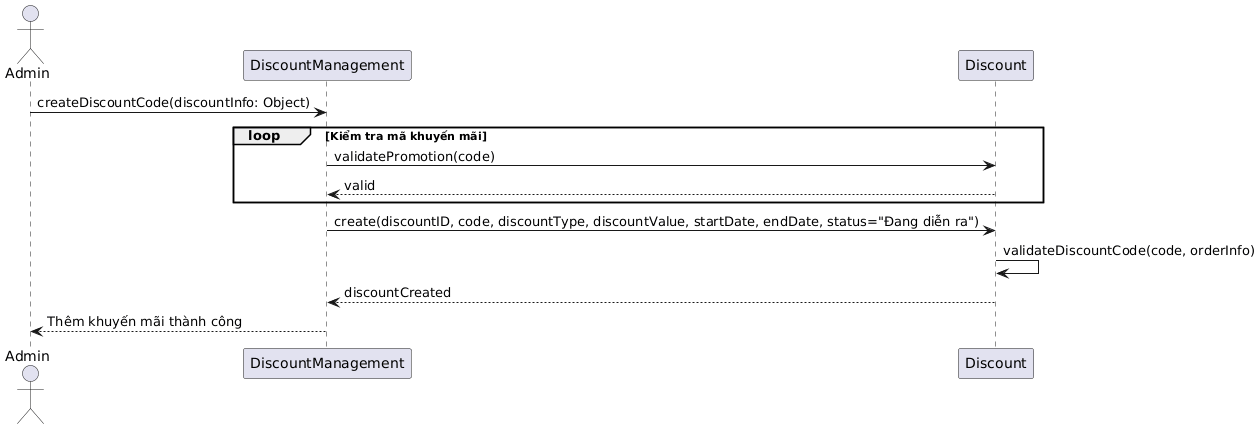
#### **3.5.2.3. Sơ đồ trình tự trạng thái thêm sản phẩm mới.**



Biểu đồ tuần tự này mô tả quy trình quản trị viên thêm sản phẩm mới và ghi nhận lô nhập liên quan. Đầu tiên, Admin gửi addProduct(productInfo) tới ProductManagement. Module này lần lượt: (1) gọi Product.create(productID, name, price, stockQuantity) để sinh bản ghi sản phẩm và nhận productCreated; (2) tạo lô nhập qua ImportBatch.create(supplier\_id, product\_id, quantity, price, status="Chờ xử lý") và nhận batchCreated. Phần khung alt thể hiện nhánh khi trạng thái lô đạt “Hoàn thành”: ProductManagement gọiInventoryManagement.updateStock(productID, quantity); Inventory kiểm tra hiện tồn bằng checkStock(productID) rồi thực thi updateStock(quantity:int). Cuối cùng hệ thống phản hồi cho Admin: “Thêm sản phẩm mới thành công”.

Đầu vào chính gồm thông tin sản phẩm (ID, tên, giá, tồn ban đầu) và thông tin lô (nhà cung cấp, số lượng, giá mua, trạng thái). Đầu ra là các sự kiện xác nhận productCreated, batchCreated và thông điệp hoàn tất cho người vận hành; đồng thời tồn kho được cập nhật khi lô hoàn tất. Để hoàn thiện, nên bổ sung các nhánh lỗi: trùng productID, thất bại ghi ImportBatch, hoặc cập nhật tồn kho không thành công; kèm ràng buộc giao dịch (transaction/idempotency) để tránh lệch tồn khi một bước thất bại. Nếu có nhiều kho, biểu đồ có thể mở rộng thêm WarehouseService và quy tắc phân bổ số lượng theo kho.

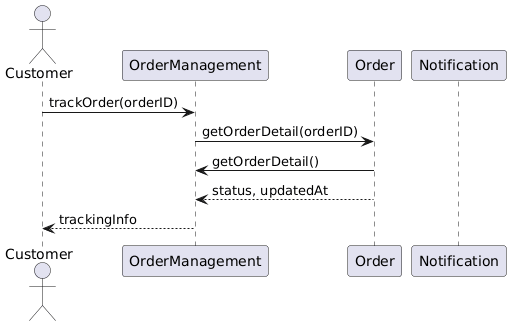
#### **3.5.2.4. Sơ đồ trình tự trạng thái thêm khuyến mãi**



Biểu đồ tuần tự này mô tả quy trình quản trị viên tạo mã khuyến mãi mới. Trước hết, Admin gửi yêu cầu createDiscountCode(discountInfo) tới mô-đun DiscountManagement. Mô-đun này bước vào pha kiểm tra hợp lệ: gọi validatePromotion(code) sang lớp Discount để xác minh mã chưa trùng, thời gian hiệu lực hợp lệ, giá trị giảm (value/percent) trong ngưỡng cho phép và các ràng buộc như giới hạn lượt dùng hay điều kiện đơn tối thiểu. Discount phản hồi kết quả kiểm tra valid.

Khi hợp lệ, DiscountManagement yêu cầu Discount tạo bản ghi: create(discountID, code, discountType, discountValue, startDate, endDate, status="Đang diễn ra"). Sau khi lưu thành công, Discount trả về sự kiện discountCreated, từ đó DiscountManagement thông báo cho Admin “Thêm khuyến mãi thành công”. Ở các luồng mở rộng, Discount còn cung cấp hàm validateDiscountCode(code, orderInfo) để được gọi trong quá trình checkout nhằm kiểm tra mã theo ngữ cảnh đơn (giỏ hàng, khách hàng, thời gian, số lần dùng), đảm bảo áp dụng đúng chính sách trước khi tính giảm giá.

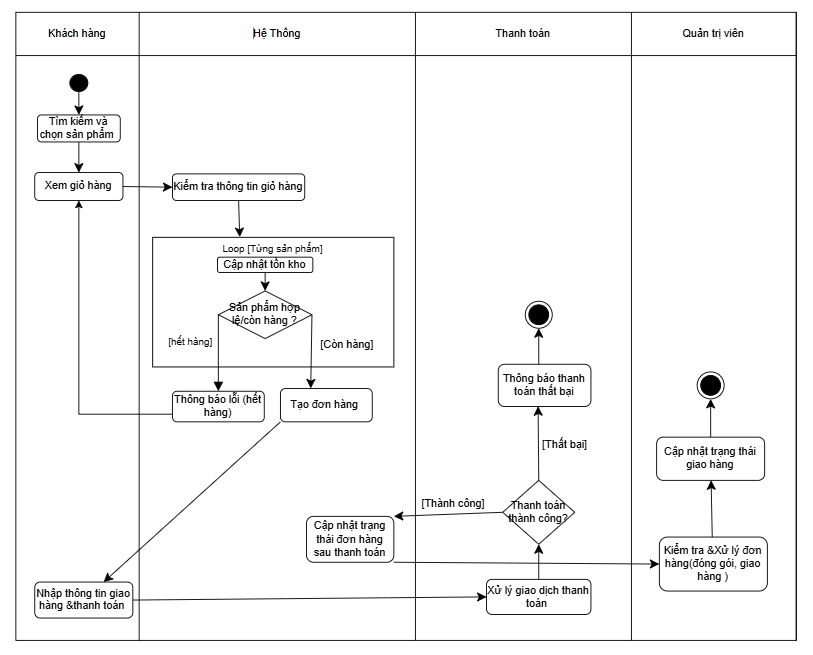
#### **3.5.2.5. Sơ đồ trình tự trạng thái theo dõi đơn hàng**



Biểu đồ tuần tự này mô tả quy trình khách hàng theo dõi trạng thái đơn hàng. Trước hết, khách hàng gửi yêu cầu trackOrder(orderID) tới mô-đun OrderManagement. Mô-đun này tiếp nhận và gọi getOrderDetail(orderID) sang lớp Order để truy xuất chi tiết đơn. Order phản hồi bằng dữ liệu getOrderDetail() chứa các trường phục vụ theo dõi như status (trạng thái hiện tại) và updatedAt (thời điểm cập nhật gần nhất).

Nhận xong dữ liệu, OrderManagement tổng hợp thành trackingInfo (có thể gồm status, updatedAt, mã vận đơn, lịch sử sự kiện) và trả về cho Customer để hiển thị. Trong phiên bản mở rộng, khi trạng thái đơn thay đổi, hệ thống có thể kích hoạt Notification để đẩy thông báo thời gian thực cho khách hàng, giúp việc theo dõi minh bạch và kịp thời.

### **3.5.3. Biểu đồ hoạt động**



Sơ đồ swimlane mô tả trọn quy trình đặt hàng từ phía khách hàng đến khâu thanh toán và vận hành, đồng thời phân tách rõ trách nhiệm giữa 4 vai trò: Khách hàng, Hệ thống, Cổng thanh toán và Quản trị viên. Luồng chuẩn bắt đầu khi khách tìm và chọn sản phẩm, xem giỏ, nhập thông tin giao hàng và lựa chọn phương thức thanh toán. Hệ thống kiểm tra giỏ theo từng mặt hàng để xác nhận tồn kho; nếu còn hàng thì tạo đơn. Cổng thanh toán xử lý giao dịch; khi giao dịch thành công, Hệ thống cập nhật trạng thái đơn. Sau đó, bộ phận quản trị tiếp nhận, đóng gói, bàn giao cho đơn vị vận chuyển và cập nhật trạng thái giao hàng. Các nhánh ngoại lệ đã thể hiện gồm hết hàng và thanh toán thất bại, hệ thống đều thông báo rõ cho khách.

Để vận hành bền vững và tránh tranh chấp dữ liệu, quy trình nên chuyển sang mô hình “giữ chỗ rồi mới trừ kho”. Cụ thể, khi khách xác nhận giỏ, Hệ thống chỉ reserve tồn kho trong thời hạn 30 đến 45 phút và tạo đơn ở trạng thái chờ thanh toán. Chỉ khi nhận webhook từ cổng thanh toán xác nhận giao dịch thành công, đối soát đúng số tiền và có cơ chế idempotency chống lặp, Hệ thống mới chuyển giữ chỗ thành trừ kho vĩnh viễn. Nếu thanh toán thất bại hoặc quá hạn, hệ thống giải phóng phần giữ chỗ và cho phép khách thanh toán lại. Khối kiểm tra, giữ chỗ và tạo đơn nên được bao gói trong transaction với cơ chế khóa thích hợp để tránh race condition. Đồng thời, nên hỗ trợ tình huống hết hàng một phần bằng cách cho phép giảm số lượng, tách đơn hoặc gợi ý sản phẩm thay thế, thay vì báo lỗi chung cho toàn bộ đơn.

Vòng đời đơn hàng nên chuẩn hóa theo các trạng thái: Chờ thanh toán, Đã thanh toán, Đã đóng gói, Đang giao, và kết thúc ở một trong 3 trạng thái Đã giao, Đã hủy hoặc Đã hoàn tiền. Hệ thống thông báo nên gửi email hoặc thông báo đẩy ở các mốc quan trọng: tạo đơn, thanh toán thành công hoặc thất bại, bàn giao cho đơn vị vận chuyển và giao thành công. Trạng thái vận chuyển nên đồng bộ tự động qua webhook của GHN, Viettel Post hoặc GHTK để tránh cập nhật thủ công dễ sai.

Về dữ liệu, ảnh chụp giỏ hàng là cơ sở để tạo bản ghi orders và order\_items; phần giữ chỗ tồn kho nên lưu riêng trong bảng inventory\_reservations với các trường order\_id, product\_id, quantity và expire\_at; giao dịch thanh toán và payload phản hồi của cổng thanh toán cần lưu tại payments và payment\_logs để phục vụ tra soát; thông tin vận chuyển theo dõi trong shipments (nhà vận chuyển, mã vận đơn, trạng thái); hệ thống thông báo lưu lịch sử gửi và trạng thái đọc trong notifications.

Cuối cùng, nên đặt mục tiêu chất lượng và kiểm thử tải thực tế: p95 thời gian tạo đơn phía Hệ thống (không tính bước chuyển hướng đến cổng thanh toán) ở mức dưới 500 ms; p95 thời gian tiếp nhận và xử lý webhook ở mức dưới 800 ms. Cần kiểm thử đồng thời từ 100 đến 1.000 người dùng mỗi giây để phát hiện khóa chết và lỗi tranh chấp, đồng thời mô phỏng webhook đến trễ hoặc gửi lặp để kiểm tra cơ chế chống lặp và khả năng tự phục hồi của quy trình.

### **3.5.4. Biểu đồ trạng thái**

#### **3.5.4.1. Biểu đồ trạng thái – Tài khoản người dùng**

A diagram of a flowchart

AI-generated content may be incorrect.

Vòng đời của một tài khoản người dùng bắt đầu khi người dùng tạo tài khoản mới, sau khi đăng ký thành công, tài khoản sẽ ở trạng thái đang hoạt động và có thể sử dụng đầy đủ các chức năng của hệ thống như mua hàng, thanh toán hoặc cập nhật thông tin cá nhân.

Trong quá trình sử dụng, tài khoản có thể chuyển giữa hai trạng thái “đang hoạt động” và “khóa tạm thời”. Khi phát hiện hoạt động bất thường, vi phạm quy định hoặc do yêu cầu bảo mật, tài khoản có thể bị khóa tạm thời. Ở trạng thái này, người dùng không thể đăng nhập hoặc thực hiện giao dịch, nhưng quản trị viên có thể mở khóa để khôi phục quyền sử dụng bình thường.

Nếu tài khoản không còn được sử dụng hoặc vi phạm nghiêm trọng, quản trị viên có thể vô hiệu hóa hoặc xóa tài khoản. Khi tài khoản ở trạng thái vô hiệu hóa, nó coi như kết thúc vòng đời hoạt động và không thể phục hồi hoặc thực hiện bất kỳ thao tác nào khác.

Nguyên tắc then chốt trong quy trình này là: mọi tài khoản không ở trạng thái “đang hoạt động” đều không được phép thực hiện thanh toán hoặc đặt hàng. Bên cạnh đó, tài khoản bị khóa tạm thời vẫn có thể được mở lại khi cần thiết, giúp linh hoạt trong việc xử lý các tình huống vi phạm nhẹ hoặc lỗi tạm thời.

#### **3.5.4.2. Biểu đồ trạng thái – Tồn kho Sản phẩm**

A diagram of a flowchart

AI-generated content may be incorrect.

Vòng đời của một sản phẩm trong hệ thống bắt đầu khi quản trị viên thêm mới sản phẩm, lúc này sản phẩm được đặt ở trạng thái “Còn hàng” và sẵn sàng để hiển thị trong danh mục cho khách hàng mua.

Trong quá trình kinh doanh, sản phẩm có thể chuyển đổi giữa hai trạng thái “Còn hàng” và “Hết hàng” tùy theo số lượng tồn kho. Khi hết hàng, nếu quản trị viên nhập thêm hàng, trạng thái sẽ quay lại “Còn hàng”. Ngược lại, nếu sản phẩm không còn được bán nữa, quản trị viên có thể đưa sản phẩm về trạng thái “Ngừng kinh doanh”, lúc này sản phẩm không còn xuất hiện trong danh mục bán hàng và khách hàng không thể đặt mua.

Các sự kiện chính trong vòng đời của sản phẩm gồm: hết hàng, nhập thêm hàng và ngừng bán. Khi sản phẩm đạt trạng thái “Ngừng kinh doanh”, quy trình quản lý sản phẩm coi như kết thúc.

Nguyên tắc quan trọng trong quá trình này là việc cập nhật số lượng tồn kho phải được thực hiện an toàn bằng cơ chế cập nhật nguyên tử (atomic) hoặc khóa lạc quan (optimistic locking) để tránh tình trạng bán vượt số lượng thực tế (oversell). Đồng thời, khi sản phẩm ở trạng thái “Ngừng kinh doanh”, hệ thống phải ẩn sản phẩm khỏi danh mục hiển thị và khóa chức năng mua hàng để đảm bảo tính nhất quán của dữ liệu.

#### **3.5.4.3. Biểu đồ trạng thái – Chọn sản phẩm**

A diagram of a diagram

AI-generated content may be incorrect.

Quá trình chọn sản phẩm trong hệ thống bắt đầu khi người dùng truy cập vào trang sản phẩm. Lúc này, hệ thống tự động chuyển đến trạng thái DuyệtSảnPhẩm, cho phép người dùng xem danh sách các mặt hàng đang có sẵn. Trong danh sách này, người dùng có thể lựa chọn một sản phẩm cụ thể để tìm hiểu thêm thông tin chi tiết.

Khi người dùng nhấn xem chi tiết, hệ thống chuyển sang trạng thái XemChiTiet, nơi hiển thị đầy đủ các thông số kỹ thuật, hình ảnh và giá bán của sản phẩm. Tại đây, người dùng có thể thiết lập số lượng và đặc tính mong muốn như màu sắc hoặc dung lượng — quá trình này được thể hiện bằng trạng thái ChonSoLuongDacTinh.

Sau khi hoàn tất lựa chọn, người dùng thực hiện thao tác “Thêm vào giỏ hàng”, khiến hệ thống chuyển sang trạng thái ThemVaoGio để xử lý dữ liệu và cập nhật thông tin trong giỏ. Khi việc thêm sản phẩm được xác nhận thành công, hệ thống sẽ sang trạng thái XacNhanGio, đồng thời cung cấp hai hướng hành động: tiếp tục duyệt sản phẩm hoặc mua ngay để đi đến trạng thái GioHang và tiến hành thanh toán.

Toàn bộ tiến trình này đảm bảo người dùng có thể quay lại danh sách sản phẩm bất kỳ lúc nào, duy trì sự linh hoạt trong trải nghiệm mua sắm.

Nguyên tắc quan trọng của sơ đồ là: mỗi thao tác chỉ được thực hiện sau khi trạng thái trước hoàn tất hợp lệ — đặc biệt, việc thêm sản phẩm vào giỏ chỉ diễn ra sau khi người dùng đã xác định rõ đặc tính và số lượng. Nhờ đó, hệ thống đảm bảo quy trình mua hàng diễn ra chính xác, logic và không phát sinh lỗi dữ liệu.

#### **3.5.4.4. Biểu đồ trạng thái – Thêm giỏ hàng**

A diagram of a diagram

AI-generated content may be incorrect.

Vòng đời của quá trình thêm sản phẩm vào giỏ hàng bắt đầu khi người dùng nhấn nút “Thêm vào giỏ hàng” trên trang chi tiết sản phẩm. Lúc này, hệ thống chuyển sang trạng thái Kiểm tra đăng nhập để xác định người dùng đã đăng nhập hay chưa. Nếu chưa đăng nhập, hệ thống yêu cầu người dùng đăng nhập và chuyển sang trạng thái Đăng nhập thành công trước khi tiếp tục xử lý.

Sau khi xác thực tài khoản, hệ thống tiến hành kiểm tra tình trạng sản phẩm. Nếu sản phẩm đã hết hàng, hệ thống hiển thị thông báo “Sản phẩm tạm hết hàng” và quay lại danh sách sản phẩm. Ngược lại, nếu sản phẩm còn hàng, quy trình tiếp tục sang kiểm tra giỏ hàng hiện tại.

Trong giai đoạn này, hệ thống phân biệt hai tình huống: nếu đã tồn tại giỏ hàng, hệ thống chỉ cập nhật số lượng sản phẩm hiện có; còn nếu chưa có giỏ hàng, hệ thống sẽ tạo giỏ hàng mới và thêm sản phẩm vào đó. Hai nhánh xử lý này đều dẫn đến trạng thái Thêm sản phẩm, nơi thông tin được ghi nhận và cập nhật vào cơ sở dữ liệu thông qua trạng thái Cập nhật giỏ hàng.

Khi dữ liệu được lưu thành công, hệ thống chuyển sang trạng thái Hiển thị thông báo, xác nhận rằng sản phẩm đã được thêm vào giỏ hàng thành công. Sau đó, quy trình kết thúc bằng việc quay lại danh sách sản phẩm, cho phép người dùng tiếp tục mua sắm hoặc xem lại giỏ hàng.

Nguyên tắc then chốt của quy trình là: việc thêm sản phẩm chỉ được thực hiện khi người dùng đã đăng nhập và sản phẩm còn tồn kho. Cơ chế này giúp đảm bảo tính chính xác trong dữ liệu đơn hàng, đồng thời ngăn ngừa tình trạng thêm các sản phẩm không hợp lệ vào hệ thống. Ngoài ra, việc kiểm tra giỏ hàng hiện có giúp tối ưu hóa trải nghiệm người dùng, tránh việc tạo nhiều giỏ hàng trùng lặp và đảm bảo mỗi hành động “Thêm vào giỏ hàng” diễn ra an toàn, thống nhất.

#### **3.5.4.5. Biểu đồ trạng thái – Đơn hàng**

A diagram of a flowchart

AI-generated content may be incorrect.

Quy trình xử lý một đơn hàng trong hệ thống bắt đầu khi khách hàng tạo đơn hàng mới, lúc đó trạng thái ban đầu của đơn là “Chờ xử lý”. Sau khi quản trị viên kiểm tra thông tin, đơn sẽ được xác nhận, chuyển sang trạng thái “Đã xác nhận”, và tiếp theo là “Đang giao” khi hàng được bàn giao cho đơn vị vận chuyển.

Trong quá trình này, có thể xảy ra nhiều sự kiện khác nhau như: xác nhận đơn hàng, tiến hành giao hàng, giao thành công, giao thất bại hoặc bị hủy giữa chừng, hay khách hàng chủ động hủy đơn. Khi đơn hàng đã được giao thành công, hệ thống sẽ chuyển sang trạng thái “Đã hoàn thành”. Ngược lại, nếu đơn bị hủy ở bất kỳ giai đoạn nào, trạng thái sẽ là “Đã hủy” — và đó cũng là điểm kết thúc vòng đời của đơn hàng.

Nguyên tắc quan trọng trong quy trình này là: chỉ được phép thực hiện việc giao hàng sau khi đơn đã được xác nhận, và khi đơn đạt trạng thái “Đã hoàn thành” hoặc “Đã hủy”, quá trình xử lý đơn hàng coi như kết thúc hoàn toàn.

A diagram of a flowchart

AI-generated content may be incorrect.

Vòng đời của một sản phẩm trong hệ thống bắt đầu khi quản trị viên thêm mới sản phẩm, lúc này sản phẩm được đặt ở trạng thái “Còn hàng” và sẵn sàng để hiển thị trong danh mục cho khách hàng mua.

Trong quá trình kinh doanh, sản phẩm có thể chuyển đổi giữa hai trạng thái “Còn hàng” và “Hết hàng” tùy theo số lượng tồn kho. Khi hết hàng, nếu quản trị viên nhập thêm hàng, trạng thái sẽ quay lại “Còn hàng”. Ngược lại, nếu sản phẩm không còn được bán nữa, quản trị viên có thể đưa sản phẩm về trạng thái “Ngừng kinh doanh”, lúc này sản phẩm không còn xuất hiện trong danh mục bán hàng và khách hàng không thể đặt mua.

Các sự kiện chính trong vòng đời của sản phẩm gồm: hết hàng, nhập thêm hàng và ngừng bán. Khi sản phẩm đạt trạng thái “Ngừng kinh doanh”, quy trình quản lý sản phẩm coi như kết thúc.

Nguyên tắc quan trọng trong quá trình này là việc cập nhật số lượng tồn kho phải được thực hiện an toàn bằng cơ chế cập nhật nguyên tử (atomic) hoặc khóa lạc quan (optimistic locking) để tránh tình trạng bán vượt số lượng thực tế (oversell). Đồng thời, khi sản phẩm ở trạng thái “Ngừng kinh doanh”, hệ thống phải ẩn sản phẩm khỏi danh mục hiển thị và khóa chức năng mua hàng để đảm bảo tính nhất quán của dữ liệu.

Vòng đời của một mã khuyến mãi bắt đầu khi quản trị viên tạo mới mã giảm giá thông qua chức năng *DiscountManagement*. Ở bước này, hệ thống sẽ kiểm tra xem mã khuyến mãi có hợp lệ hay chưa bằng quy trình validatePromotion(code). Nếu mã hợp lệ và không trùng lặp, hệ thống sẽ tiến hành tạo bản ghi khuyến mãi mới với các thông tin như discountID, code, discountType, discountValue, startDate, endDate và đặt trạng thái ban đầu là “Đang diễn ra”. Sau khi tạo thành công, thông báo “Thêm khuyến mãi thành công” sẽ được gửi lại cho quản trị viên.

Trong quá trình hoạt động, mã khuyến mãi có thể được xác thực lại khi khách hàng sử dụng trong đơn hàng thông qua hàm validateDiscountCode(code, orderInfo) để đảm bảo rằng mã vẫn còn hiệu lực, còn trong thời gian áp dụng và đáp ứng các điều kiện giảm giá. Khi mã đã hết hạn hoặc bị vô hiệu hóa bởi quản trị viên, hệ thống sẽ tự động chuyển trạng thái của mã sang “Hết hạn” hoặc “Ngừng áp dụng”.

Các sự kiện chính trong vòng đời của mã khuyến mãi bao gồm: tạo mới, kiểm tra hợp lệ, kích hoạt, sử dụng trong đơn hàng, hết hạn và ngừng áp dụng. Khi mã đạt trạng thái “Ngừng áp dụng”, quy trình quản lý mã khuyến mãi coi như kết thúc.

Nguyên tắc quan trọng trong quá trình này là việc xác thực và cập nhật trạng thái mã giảm giá phải được thực hiện một cách nhất quán và đồng bộ để tránh tình trạng dùng mã hết hạn hoặc trùng lặp. Đồng thời, hệ thống cần đảm bảo rằng khi mã ở trạng thái “Ngừng áp dụng” thì nó không còn được hiển thị hoặc sử dụng trong bất kỳ đơn hàng nào nhằm đảm bảo tính toàn vẹn dữ liệu và trải nghiệm người dùng.

# **4. TỔNG KẾT VÀ HƯỚNG PHÁT TRIỂN**

## **4.1. Đánh giá**

Dự án “Hệ thống Quản lý Cửa Hàng Thiết Bị Điện Tử Trực Tuyến” được triển khai theo Agile–Scrum, giúp nhóm lặp–cải tiến nhanh và kiểm soát phạm vi hiệu quả. Trên nền Laravel + MySQL (Laragon) và giao diện Tailwind, hệ thống đã hiện thực đầy đủ các luồng cốt lõi: đăng ký/đăng nhập, tìm kiếm–lọc, giỏ hàng, đặt hàng, thanh toán, theo dõi đơn, đánh giá và hỗ trợ khách hàng; dữ liệu tuân thủ ràng buộc quan hệ, đảm bảo tính toàn vẹn. Về kỹ thuật, các thao tác đọc/ghi chính đáp ứng mục tiêu tải trang dưới 5 giây trong điều kiện thử nghiệm, giao diện thân thiện, phân quyền RBAC rõ ràng, mật khẩu băm bcrypt và truyền thông qua HTTPS.

Về mô hình hóa, các sơ đồ Use Case, Class, Sequence, Activity, State thể hiện đúng nghiệp vụ và kiến trúc ba tầng, tạo nền tảng bảo trì–mở rộng thuận lợi. Hạn chế còn lại chủ yếu do thời gian: các module nâng cao như phân tích dữ liệu bán hàng, gợi ý sản phẩm, chat thời gian thực mới ở mức nguyên mẫu; thanh toán chưa kết nối API thực (VNPAY/Momo/ZaloPay); giao diện mobile cần tối ưu thêm, và kiểm thử tải/stress test ở quy mô lớn cần được bổ sung để xác nhận biên hiệu năng.

## **4.2. Hướng phát triển**

Trong giai đoạn tới, hệ thống sẽ được nâng cấp theo năm hướng chính nhằm đưa sản phẩm từ mức bài tập lớn lên mức vận hành thực tế. Trước hết là hoàn thiện nghiệp vụ và tích hợp thực tế: kết nối API thanh toán của VNPAY, Momo, ZaloPay với cơ chế webhook idempotent và đối soát tự động; đồng thời tích hợp GHN, Viettel Post, GHTK để đồng bộ trạng thái giao hàng theo thời gian thực, và chuẩn hóa quy trình hoàn/đổi/bảo hành có lịch hẹn, nhắc việc và thông báo. Song song, nhóm tập trung nâng cao UX/UI theo hướng mobile-first, bổ sung dark mode và thông báo đẩy, cá nhân hóa đề xuất dựa trên hành vi mua sắm, và xây dựng live chat real-time bằng WebSocket/Pusher/Firebase để rút ngắn thời gian phản hồi.

Tiếp theo là tăng cường năng lực phân tích và báo cáo: phát triển dashboard doanh thu, vòng quay tồn, tỷ lệ chuyển đổi; áp dụng mô hình học máy để dự đoán nhu cầu và gợi ý sản phẩm; xem xét kiến trúc ETL và Data Warehouse phục vụ phân tích dài hạn. Ở tầng nền tảng, hệ thống sẽ nâng cấp bảo mật và khả năng mở rộng bằng JWT/OAuth2/SSO, hoàn thiện logging–monitoring–backup định kỳ, đóng gói triển khai với Docker và thiết lập CI/CD, đồng thời cân nhắc tiến hóa sang modular monolith hoặc microservices để tách dần các domain (product, order, payment, support). Cuối cùng, về định hướng sản phẩm, giải pháp sẽ hỗ trợ đa chi nhánh/multi-store, phát triển ứng dụng di động (Flutter/React Native), tích hợp chatbot hỗ trợ tự động và hoàn thiện hệ gợi ý theo collaborative/content-based filtering—tất cả nhằm bảo đảm hiệu năng, an toàn và trải nghiệm người dùng ở quy mô lớn hơn.