Software Requirements Specification

for

UTE-PhoneHub

Version 2.0 approved

Prepared by Do Kien Hung

Group 01 (Web Development)

Group 08 (Software Engineering)

01.01.2026

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
For Web Development	01/01/2026	- Cập nhật SRS cho môn học Công nghệ phần mềm.	2.0
For Web Development	10/10/2025	- Cập nhật SRS cho môn học Lập trình Web..	1.0
v0.3	21/09/2025	- Làm rõ ràng buộc toàn vẹn dữ liệu: Bổ sung yêu cầu không cho phép xóa các thực thể cha (danh mục, thương hiệu) khi vẫn còn các thực thể con liên kết. - Tăng cường yêu cầu bảo mật: Bổ sung yêu cầu chống tấn công brute-force cho các chức năng nhạy cảm.	
v0.2	15/09/2025	- Bổ sung chức năng Đặt hàng cho Khách (Guest Checkout) và Tra cứu đơn hàng công khai. - Tăng cường yêu cầu về mật khẩu mạnh. - Làm rõ logic quản lý tồn kho. - Định nghĩa cơ chế xóa mềm (soft-delete) cho các dữ liệu quan trọng. - Thêm các yêu cầu phi chức năng về bảo mật và hiệu năng.	
v0.1	26/08/2025	Phiên bản ban đầu của tài liệu.	

1. Introduction

Phần này cung cấp cái nhìn tổng quan về tài liệu SRS, xác định mục đích, phạm vi, các định nghĩa và cấu trúc của tài liệu cho dự án UTE Phone Hub

1.1 Purpose

Tài liệu này đặc tả chi tiết các yêu cầu phần mềm cho hệ thống thương mại điện tử **UTE Phone Hub**. Mục đích chính của tài liệu là:

1. **Thống nhất yêu cầu:** Là "nguồn sự thật duy nhất" (Single Source of Truth) giữa các bên liên quan (Giảng viên, PM, Tech Lead, Developers) về những gì hệ thống phải làm.
2. **Cơ sở thiết kế & phát triển:** Cung cấp thông tin chi tiết để đội ngũ Backend xây dựng API và đội ngũ Frontend phát triển giao diện người dùng (UI/UX).
3. **Căn cứ kiểm thử:** Giúp đội ngũ QA/Tester (hoặc chính Developer) viết các kịch bản kiểm thử (Test Cases) để đảm bảo sản phẩm đạt chất lượng.
4. **Định hướng công nghệ:** Xác nhận việc áp dụng kiến trúc hiện đại (Headless Commerce) với Spring Boot và Next.js.

Tài liệu này dành cho:

- **Developers:** Để hiểu rõ nghiệp vụ và logic cần code.
- **Project Manager:** Để theo dõi tiến độ và phạm vi dự án (Scope Creep).
- **Giảng viên/Hội đồng bảo vệ:** Để đánh giá độ phức tạp và tính hoàn thiện của đồ án.

1.2 Document Conventions

Các yêu cầu trong tài liệu này được định danh bằng một mã duy nhất để dễ dàng theo dõi và tham chiếu, theo quy ước sau:

- **[FR-XXX-NN]:** Dành cho Yêu cầu Chức năng (Functional Requirement).
    - FR: Viết tắt của Functional Requirement.

  - XXX: Tên nhóm chức năng (ví dụ: AUTH, PRODUCT).
  - NN: Số thứ tự của yêu cầu.
  - Ví dụ: [FR-PRODUCT-01] là yêu cầu chức năng số 01 thuộc nhóm Sản phẩm.
- **[NFR-NN]:** Dành cho Yêu cầu Phi chức năng (Non-functional Requirement).
  - NFR: Viết tắt của Non-functional Requirement.
  - NN: Số thứ tự của yêu cầu.
  - Ví dụ: [NFR-01] là yêu cầu phi chức năng về hiệu năng.

1.3 Project Scope

UTE Phone Hub là một nền tảng thương mại điện tử B2C (Business-to-Consumer) chuyên kinh doanh điện thoại di động và phụ kiện công nghệ. Hệ thống được xây dựng theo kiến trúc tách biệt hoàn toàn giữa Frontend và Backend (Decoupled Architecture).

1.3.1 Các chức năng trong phạm vi

A. Phân hệ Khách hàng (Client Portal - Next.js):

1. **Khám phá sản phẩm:** Tìm kiếm, Lọc nâng cao (theo cấu hình, giá, hãng), Xem chi tiết, So sánh sản phẩm.
2. **Tài khoản & Bảo mật:** Đăng ký/Đăng nhập (Email & **Google OAuth2**), Quên mật khẩu (OTP), Quản lý hồ sơ, Sổ địa chỉ.
3. **Mua sắm:** Giỏ hàng thông minh (đồng bộ Redis), Thanh toán đa phương thức (**VNPay**, COD), Áp dụng Mã giảm giá (Voucher).
4. **Tương tác:** Đánh giá sản phẩm, Yêu thích (Wishlist), **Chatbot AI** hỗ trợ tư vấn.
5. **Theo dõi:** Tra cứu trạng thái đơn hàng (Real-time tracking), Lịch sử mua hàng.

B. Phân hệ Quản trị (Admin Portal - Next.js/Admin Layout):

1. **Dashboard:** Biểu đồ thống kê doanh thu, đơn hàng, người dùng mới (Chart.js/Recharts).
2. **Quản lý Catalog:** CRUD Sản phẩm, Danh mục, Thương hiệu, Upload ảnh (Cloudinary/Local).

3. **Vận hành:** Quản lý Đơn hàng (Duyệt/Hủy/Cập nhật trạng thái), Quản lý Kho hàng.
4. **Marketing:** Tạo và quản lý chiến dịch Khuyến mãi (Voucher).
5. **Hệ thống:** Quản lý Người dùng, Phân quyền (Role-based).

1.3.2 Các chức năng ngoài phạm vi

- Xây dựng ứng dụng di động Native (iOS/Android) - Chỉ tập trung vào Responsive Web Design (RWD).
- Tích hợp hệ thống vận chuyển thực tế (GiaoHangNhanh/GHTK API) - Sử dụng phí ship giả lập hoặc cố định.
- Hệ thống Affiliate Marketing (Tiếp thị liên kết).
- Quản lý đa chi nhánh/kho bãi vật lý phức tạp.

1.4 Definitions, Acronyms, and Abbreviations

Thuật ngữ	Định nghĩa
BFF	Backend For Frontend (Mô hình Backend phục vụ riêng cho Frontend).
JWT	JSON Web Token (Chuẩn bảo mật dùng để xác thực không trạng thái).
OAuth2	Giao thức ủy quyền mở (Dùng cho chức năng Login with Google).
ORM	Object-Relational Mapping (Ánh xạ đối tượng - CSDL, cụ thể là Hibernate).
DTO	Data Transfer Object (Đối tượng chuyển dữ liệu giữa các lớp).

Redis	Remote Dictionary Server (Dùng làm Database in-memory để Caching).

2. OVERALL DESCRIPTION

Phần này mô tả bối cảnh chung của sản phẩm, đối tượng người dùng, môi trường vận hành và các giả định, ràng buộc kỹ thuật của dự án **UTE Phone Hub**.

2.1 Product Perspective

UTE Phone Hub là một hệ thống thương mại điện tử thế hệ mới, được thiết kế để thay thế các mô hình web bán hàng truyền thống (Monolithic JSP/Servlet). Hệ thống hoạt động như một giải pháp độc lập nhưng có khả năng tích hợp linh hoạt với các dịch vụ bên thứ ba.

Hệ thống được xây dựng dựa trên kiến trúc **Restful API** (API-First Design), trong đó:

- **Backend (Spring Boot):** Đóng vai trò là trung tâm xử lý nghiệp vụ, quản lý dữ liệu và cung cấp API. Nó hoạt động như một "Black box" đối với Frontend.
- **Frontend (Next.js):** Là giao diện người dùng hiện đại, tương tác với Backend thông qua HTTP Requests và đảm nhiệm việc hiển thị dữ liệu (Server-Side Rendering).
- **Hệ sinh thái tích hợp:** Hệ thống không đứng một mình mà kết nối với:
    - **Google Identity Platform:** Để xác thực người dùng (SSO).
    - **VNPay Sandbox:** Để xử lý thanh toán trực tuyến.
    - **Redis Server:** Để quản lý phiên làm việc hiệu suất cao và lưu trữ giỏ hàng tạm.
    - **PostgreSQL:** Để lưu trữ dữ liệu bền vững.

2.2 User Classes and Characteristics

Hệ thống phục vụ 3 nhóm đối tượng chính với các quyền hạn và đặc điểm kỹ thuật khác nhau:

Nhóm người dùng	Mô tả & Đặc điểm	Mục tiêu sử dụng

1. Khách vãng lai (Guest)	Người dùng chưa đăng nhập. Thường truy cập qua tìm kiếm hoặc quảng cáo.	Tìm kiếm sản phẩm, xem đánh giá, so sánh cấu hình, thêm vào giỏ hàng, đăng ký tài khoản.
2. Thành viên (Member)	Người dùng đã có tài khoản và đăng nhập. Có hiểu biết cơ bản về công nghệ.	Mua hàng, thanh toán online, theo dõi đơn hàng, quản lý sổ địa chỉ, tích lũy điểm (nếu có), đánh giá sản phẩm.
3. Quản trị viên (Admin)	Đội ngũ vận hành hệ thống. Có kỹ năng quản trị và nghiệp vụ bán hàng.	Quản lý kho, xử lý đơn hàng, quản lý người dùng, thiết lập khuyến mãi, xem báo cáo doanh thu.

2.3 Operating Environment

Hệ thống được thiết kế để triển khai trên hạ tầng **Containerization (Docker)**, đảm bảo tính nhất quán giữa môi trường Dev và Production.

2.3.1 Phía Server

- **OS:** Linux (Ubuntu/Alpine) hoặc Windows Server (thông qua Docker Desktop).
- **Application Server:** Spring Boot Embedded Tomcat (chạy trên port 8081).
- **Runtime:** Java Development Kit (JDK) 17 LTS.
- **Database:** PostgreSQL 15.x.
- **Cache/Session Store:** Redis 7.x.
- **Containerization:** Docker & Docker Compose.

2.3.2 Phía Client

- **Web Server:** Node.js 18+ / Next.js Runtime (chạy trên port 3000).
- **Browsers:** Tương thích với tất cả trình duyệt hiện đại (Chrome 90+, Firefox 90+, Safari 14+, Edge).
- **Devices:** Responsive hoàn toàn trên Desktop, Laptop, Tablet và Mobile Phone.

2.4 Design and Implementation Constraints (Ràng buộc thiết kế & Triển khai)

Đội ngũ phát triển phải tuân thủ nghiêm ngặt các ràng buộc sau để đảm bảo tính thống nhất và khả năng bảo trì:

1. **Tech Stack bắt buộc:**
    - **Backend:** Spring Boot 3.5.8, Spring Security 6, JPA/Hibernate.
    - **Frontend:** Next.js 15.5.6 (App Router), React 19, Tailwind CSS 4, Shadcn/UI.
    - **State Management:** Zustand (Client-side), React Query (Server-state - Optional).
2. **Kiến trúc:**
    - Backend phải tuân thủ mô hình **Layered Architecture** (Controller -> Service -> Repository).
    - Không được viết logic nghiệp vụ trong Controller.
    - Tuyệt đối không trả về Entity trực tiếp qua API (Phải dùng DTO).
3. **Bảo mật:**
    - Không lưu mật khẩu dạng Plain-text (Bắt buộc dùng BCrypt).
    - Giao tiếp API phải được bảo vệ bằng JWT (Access Token & Refresh Token).
    - Các API nhạy cảm (Admin, Payment) phải có cơ chế phân quyền (RBAC).
4. **Quy chuẩn Code:**
    - Tuân thủ tài liệu CONVENTIONS.md.
    - Commit message theo chuẩn Conventional Commits.

2.5 Assumptions and Dependencies

2.5.1 Giả định

- Người dùng có kết nối Internet ổn định khi sử dụng hệ thống.
- Số lượng sản phẩm ban đầu khoảng 100-500 SKU (Không yêu cầu xử lý Big Data ngay lập tức).
- Tỷ giá tiền tệ là VNĐ, không xử lý đa tiền tệ trong giai đoạn này.

2.5.2 Phụ thuộc

- **VNPay Sandbox:** Hệ thống phụ thuộc vào môi trường thử nghiệm của VNPay. Nếu VNPay bảo trì, chức năng thanh toán online sẽ tạm ngưng.
- **Google OAuth2 Service:** Phụ thuộc vào Google Cloud Platform để thực hiện đăng nhập nhanh.
- **SMTP Server:** Phụ thuộc vào Google Mail Server để gửi email xác thực/OTP.

3. SYSTEM FEATURES

Phần này đặc tả chi tiết các yêu cầu chức năng (Functional Requirements) được tổ chức theo 10 Module của dự án.

3.1 Module 1: Authentication & Security

Phụ trách: Đỗ Kiến Hưng

3.1.1 Description

Module quản lý định danh người dùng, đảm bảo tính bảo mật cho toàn bộ hệ thống thông qua cơ chế JWT và OAuth2.

3.1.1.1. Đăng ký Tài khoản

Mục	Nội dung
Use Case ID	**[UC-M01-01]**
Tên Use Case	**Đăng ký Tài khoản**
Actor	Guest
Trigger	Khách hàng truy cập website và chọn chức năng "Đăng ký" trên thanh navbar giao diện chính.
Description	Use case này cho phép khách hàng mới tạo một tài khoản trên hệ thống UTE Phone Hub bằng cách cung cấp thông tin cá nhân cần thiết.
Pre-Conditions	• Khách hàng chưa có tài khoản trên hệ thống. • Email cần đăng ký đúng định dạng và chưa từng được sử dụng để đăng ký.
Post-Conditions	• Tài khoản khách hàng được tạo thành công trong cơ sở dữ liệu với trạng thái "Hoạt động".

	• Khách hàng nhận được email chào mừng. • Khách hàng được điều hướng đến trang đăng nhập.
Main Flow	1. Khách hàng chọn nút "Đăng nhập" trên thanh navbar giao diện chính. 2. Tại giao diện đăng nhập, chọn đăng ký. Hệ thống hiển thị trang Đăng ký với các trường thông tin. 3. Khách hàng nhập các thông tin bắt buộc: Họ tên, Email, Mật khẩu, Nhập lại mật khẩu. 4. Khách hàng nhấn nút "Đăng ký". 5. Hệ thống (Frontend) gửi yêu cầu đăng ký đến API Backend. 6. Hệ thống (Backend) thực hiện xác thực đầu vào: 6.A. Kiểm tra định dạng Email (ít nhất 6 ký tự, bao gồm ký tự @ và ký tự). 6.B. Kiểm tra độ mạnh của Mật khẩu (ít nhất 8 ký tự, bao gồm chữ hoa, chữ thường, số và ký tự đặc biệt). 6.C. Kiểm tra Mật khẩu và Nhập lại mật khẩu có khớp nhau không. 6.D. Kiểm tra xem Email đã tồn tại trong hệ thống chưa. 7. Nếu tất cả các bước xác thực đều thành công: 7.A. Hệ thống tạo một tài khoản người dùng mới với trạng thái "Hoạt động". 7.B. Mã hóa mật khẩu bằng BCrypt và lưu vào cơ sở dữ liệu. 7.C. Gửi một email chào mừng đến địa chỉ email của khách hàng. 7.D. Trả về thông báo thành công cho Frontend. 8. Hệ thống (Frontend) hiển thị thông báo thành công và chuyển hướng khách hàng về trang đăng nhập.
Alternate Flow	Không có.

	• **6.A, 6.B, 6.C, 6.D. Dữ liệu không hợp lệ:** Hệ thống (Frontend) trả về lỗi 400 (Bad Request) với thông báo chi tiết. Frontend hiển thị thông báo lỗi tương ứng tại trường nhập liệu.
Exception Flow	• **6.D. Email đã tồn tại:** Hệ thống (Frontend) trả về lỗi 409 (Conflict). Frontend hiển thị thông báo "Email này đã được sử dụng" và gợi ý đăng nhập hoặc quên mật khẩu. • **7.C. Lỗi gửi email:** Hệ thống ghi nhận lỗi, nhưng vẫn tạo tài khoản và cho phép đăng ký thành công.

3.1.1.2. Đăng nhập hệ thống

Mục	**Nội dung**
Use Case ID	**[UC-M01-02]**
Tên Use Case	**Đăng nhập bằng Email và Mật khẩu**
Actor	CUSTOMER
Trigger	Khách hàng truy cập trang Đăng nhập và điền thông tin.
Description	Cho phép khách hàng đã đăng ký truy cập vào tài khoản của mình bằng email hoặc tên đăng nhập và mật khẩu.
Pre-Conditions	Khách hàng đã có tài khoản.
Post-Conditions	• Khách hàng đăng nhập thành công. • Hệ thống tạo và trả về cặp JWT (Access Token & Refresh Token). • Frontend lưu trữ token và điều hướng khách hàng đến trang chủ hoặc trang trước đó.

Main Flow	1. Khách hàng truy cập trang Đăng nhập. 2. Khách hàng nhập Email và Mật khẩu. 3. Khách hàng nhấn nút "Đăng nhập". 4. Hệ thống (Frontend) gửi thông tin đăng nhập đến API Backend. 5. Hệ thống (Backend) tìm người dùng bằng email hoặc tên đăng nhập. 6. So sánh mật khẩu đã nhập với mật khẩu đã băm trong DB bằng BCrypt. 7. Nếu mật khẩu khớp, hệ thống kiểm tra trạng thái tài khoản (Active/Locked). 8. Hệ thống tạo một Access Token (thời hạn 24 giờ) và một Refresh Token (thời hạn 7 ngày). 9. Lưu Refresh Token vào Redis để quản lý. 10. Trả về cặp token và thông tin cơ bản của người dùng cho Frontend. 11. Hệ thống (Frontend) lưu trữ token (ví dụ: Access Token trong memory, Refresh Token trong HttpOnly cookie) và cập nhật trạng thái đăng nhập trên UI. 12. Điều hướng khách hàng đến trang chủ.
Alternate Flow	Không có.
Exception Flow	• **5. Không tìm thấy email:** Hệ thống trả về lỗi 401 (Unauthorized) với thông báo "Email hoặc mật khẩu không chính xác". • **6. Mật khẩu không khớp:** Hệ thống trả về lỗi 401 (Unauthorized) với thông báo "Email hoặc mật khẩu không chính xác". • **7. Tài khoản bị khóa:** Hệ thống trả về lỗi 403 (Forbidden) với thông báo "Tài khoản của bạn đã bị khóa".

3.1.1.3. Đăng nhập Google (OAuth2)

Mục	Nội dung
Use Case ID	**[UC-M01-03]**
Tên Use Case	**Đăng nhập bằng Google**
Actor	CUSTOMER
Trigger	Khách hàng chọn nút "Đăng nhập bằng Google" trên trang Đăng nhập/Đăng ký.
Description	Cho phép khách hàng đăng nhập hoặc đăng ký nhanh chóng bằng tài khoản Google của họ.
Pre-Conditions	Khách hàng có một tài khoản Google đang hoạt động.
Post-Conditions	• Khách hàng đăng nhập thành công. • Nếu là người dùng mới, một tài khoản mới sẽ được tự động tạo và xác thực. • Hệ thống tạo và trả về cặp JWT.
Main Flow	1. Khách hàng nhấn nút "Đăng nhập bằng Google". 2. Hệ thống (Frontend) mở cửa sổ popup/redirect đến trang xác thực của Google. 3. Khách hàng đăng nhập vào tài khoản Google và đồng ý cấp quyền cho ứng dụng. 4. Google trả về một mã ủy quyền (authorization code) cho Frontend. 5. Hệ thống (Frontend) gửi mã ủy quyền này đến API Backend.

	6. Hệ thống (Backend) dùng mã ủy quyền để trao đổi với Google và nhận về thông tin người dùng (email, họ tên). 7. Hệ thống (Backend) kiểm tra xem email nhận được từ Google đã tồn tại trong DB hay chưa. 7.A. Nếu đã tồn tại: Tiến hành đăng nhập cho người dùng đó (chuyển đến bước 8 của UC-M01-02). 7.B. Nếu chưa tồn tại: Tự động tạo một tài khoản mới với thông tin từ Google, đặt trạng thái là "Hoạt động", và tiến hành đăng nhập (chuyển đến bước 8 của UC-M01-02). 8. Hệ thống (Backend) trả về cặp JWT và thông tin người dùng cho Frontend. 9. Hệ thống (Frontend) lưu trữ token và điều hướng khách hàng đến trang chủ.
Alternate Flow	Không có.
Exception Flow	• **3. Khách hàng từ chối cấp quyền:** Cửa sổ của Google đóng lại, không có hành động nào xảy ra. • **6. Lỗi giao tiếp với Google:** Hệ thống (Backend) trả về lỗi 500 (Internal Server Error). Frontend hiển thị thông báo "Đã có lỗi xảy ra với việc đăng nhập Google, vui lòng thử lại".

3.1.1.4. Đăng xuất

Mục	**Nội dung**
Use Case ID	**[UC-M01-04]**
Tên Use Case	**Đăng xuất**

Actor	CUSTOMER
Trigger	Khách hàng chọn chức năng "Đăng xuất" trên giao diện.
Description	Cho phép khách hàng kết thúc phiên làm việc và đăng xuất an toàn khỏi hệ thống.
Pre-Conditions	Khách hàng đang trong một phiên đăng nhập hợp lệ.
Post-Conditions	• Phiên làm việc của khách hàng kết thúc. • Refresh Token bị vô hiệu hóa. • Frontend xóa bỏ các token đã lưu và cập nhật UI về trạng thái chưa đăng nhập.
Main Flow	1. Khách hàng nhấn vào nút "Đăng xuất". 2. Hệ thống (Frontend) gửi yêu cầu đăng xuất đến API Backend, kèm theo Refresh Token. 3. Hệ thống (Backend) nhận Refresh Token và xóa nó khỏi Redis. 4. Hệ thống (Backend) trả về thông báo đăng xuất thành công. 5. Hệ thống (Frontend) xóa Access Token và Refresh Token khỏi bộ nhớ/cookie. 6. Cập nhật giao diện về trạng thái chưa đăng nhập và điều hướng về trang chủ.
Alternate Flow	Không có.
Exception Flow	Không có.

3.1.1.5. Quên Mật khẩu

Mục	Nội dung
Use Case ID	**[UC-M01-05]**
Tên Use Case	**Quên Mật khẩu**
Actor	CUSTOMER
Trigger	Khách hàng chọn liên kết "Quên mật khẩu" trên trang Đăng nhập.
Description	Cung cấp một quy trình an toàn để khách hàng có thể đặt lại mật khẩu của mình thông qua email.
Pre-Conditions	Khách hàng phải nhớ và truy cập được địa chỉ email đã dùng để đăng ký tài khoản.
Post-Conditions	Mật khẩu của khách hàng được cập nhật thành công.
Main Flow	**Giai đoạn 1: Yêu cầu đặt lại mật khẩu** 1. Khách hàng nhấn "Quên mật khẩu". 2. Hệ thống hiển thị trang yêu cầu nhập email. 3. Khách hàng nhập email và nhấn "Gửi". 4. Hệ thống (Backend) kiểm tra email có tồn tại trong DB không. 5. Nếu có, hệ thống tạo một mã OTP (ví dụ: 6 chữ số), lưu vào Redis kèm email và thời gian hết hạn (ví dụ: 5 phút). 6. Hệ thống gửi email chứa mã OTP đến cho khách hàng. 7. Hệ thống (Frontend) điều hướng đến trang nhập OTP. **Giai đoạn 2: Xác thực OTP và Đặt lại mật khẩu** 8. Khách hàng nhập mã OTP nhận được từ email.

	9. Khách hàng nhập Mật khẩu mới và Nhập lại mật khẩu mới. 10. Khách hàng nhấn "Xác nhận". 11. Hệ thống (Backend) xác thực OTP (so sánh với giá trị trong Redis, kiểm tra chưa hết hạn). 12. Nếu OTP hợp lệ, hệ thống xác thực mật khẩu mới (độ mạnh, khớp nhau). 13. Hệ thống băm mật khẩu mới bằng BCrypt và cập nhật vào DB cho người dùng tương ứng. 14. Xóa OTP khỏi Redis. 15. Hệ thống (Frontend) hiển thị thông báo thành công và điều hướng đến trang Đăng nhập.
Alternate Flow	Không có.
Exception Flow	• **4. Email không tồn tại:** Hệ thống vẫn hiển thị thông báo "Nếu email tồn tại, một mã OTP sẽ được gửi đi" để tránh việc dò email. • **11. OTP không chính xác hoặc hết hạn:** Hệ thống trả về lỗi. Frontend hiển thị thông báo "Mã OTP không hợp lệ hoặc đã hết hạn" và cung cấp tùy chọn gửi lại OTP. • **12. Mật khẩu mới không hợp lệ:** Frontend hiển thị thông báo lỗi tương ứng.

3.1.1.6. Cập nhật Thông tin cá nhân

Mục	**Nội dung**
Use Case ID	**[UC-M01-06]**
Tên Use Case	**Cập nhật Thông tin cá nhân**

Actor	CUSTOMER
Trigger	Khách hàng truy cập trang "Tài khoản của tôi" và chọn chỉnh sửa thông tin.
Description	Cho phép khách hàng xem và cập nhật các thông tin cá nhân của mình như Họ tên, Số điện thoại.
Pre-Conditions	Khách hàng đã đăng nhập.
Post-Conditions	Thông tin cá nhân của khách hàng được cập nhật thành công trong cơ sở dữ liệu.
Main Flow	1. Khách hàng điều hướng đến trang "Hồ sơ cá nhân". 2. Hệ thống hiển thị thông tin hiện tại của khách hàng. 3. Khách hàng nhấn nút "Chỉnh sửa". 4. Khách hàng thay đổi các thông tin (ví dụ: Họ tên, Số điện thoại) và nhấn "Lưu thay đổi". 5. Hệ thống (Frontend) gửi yêu cầu cập nhật đến API Backend, kèm theo JWT trong header Authorization. 6. Hệ thống (Backend) xác thực JWT, lấy thông tin người dùng từ token. 7. Cập nhật thông tin mới vào bảng users trong DB. 8. Trả về thông tin người dùng đã được cập nhật. 9. Hệ thống (Frontend) hiển thị thông báo thành công và cập nhật lại thông tin trên giao diện.
Alternate Flow	Không có.

Exception Flow	Không có.

3.1.1.7. Quản lý Địa chỉ Giao hàng

Mục	**Nội dung**
Use Case ID	**[UC-M01-07]**
Tên Use Case	**Quản lý Địa chỉ Giao hàng**
Actor	CUSTOMER
Trigger	Khách hàng truy cập trang "Quản lý địa chỉ" trong phần tài khoản.
Description	Cho phép khách hàng thêm, xem, sửa, xóa các địa chỉ giao hàng và đặt một địa chỉ làm mặc định.
Pre-Conditions	Khách hàng đã đăng nhập.
Post-Conditions	Danh sách địa chỉ của khách hàng được cập nhật trong cơ sở dữ liệu.
Main Flow	1. Khách hàng truy cập trang "Quản lý địa chỉ". 2. Hệ thống hiển thị danh sách các địa chỉ đã lưu của khách hàng. **Luồng A: Thêm địa chỉ mới** A1. Khách hàng nhấn "Thêm địa chỉ mới". A2. Hệ thống hiển thị form nhập thông tin địa chỉ (Họ tên người nhận, SĐT, Tỉnh/Thành, Phường/Xã, địa chỉ cụ thể). A3. Khách hàng điền thông tin và nhấn "Lưu".

	A4. Hệ thống (Backend) lưu địa chỉ mới vào bảng addresses và liên kết với người dùng. A5. Giao diện được cập nhật với địa chỉ mới. **Luồng B: Sửa địa chỉ** B1. Khách hàng chọn "Sửa" trên một địa chỉ đã có. B2. Hệ thống hiển thị form với thông tin cũ, cho phép khách hàng chỉnh sửa. B3. Khách hàng nhấn "Lưu". B4. Hệ thống (Backend) cập nhật thông tin địa chỉ. **Luồng C: Xóa địa chỉ** C1. Khách hàng chọn "Xóa" trên một địa chỉ. C2. Hệ thống hiển thị hộp thoại xác nhận. C3. Khách hàng xác nhận xóa. C4. Hệ thống (Backend) xóa địa chỉ khỏi DB. **Luồng D: Đặt làm mặc định** D1. Khách hàng chọn "Đặt làm mặc định" trên một địa chỉ không phải là mặc định. D2. Hệ thống (Backend) cập nhật trạng thái mặc định cho địa chỉ này và bỏ trạng thái mặc định của địa chỉ cũ (nếu có).
Alternate Flow	Không có.
Exception Flow	Không có.

3.1.1.8. Gửi Email OTP

Mục	Nội dung
Use Case ID	**[UC-M01-08]**
Tên Use Case	**Gửi Email OTP**
Actor	System (Email Service)
Trigger	Được kích hoạt bởi Use Case "Quên Mật khẩu" hoặc "Đăng ký" (nếu có xác thực email).
Description	Hệ thống gửi email chứa mã OTP đến người dùng để xác thực quyền sở hữu email.
Pre-Conditions	• Email người nhận hợp lệ. • Mã OTP đã được tạo.
Post-Conditions	Email chứa mã OTP được gửi đến hộp thư người dùng.
Main Flow	1. Component yêu cầu (AuthService) gọi EmailService.sendOtpEmail(). 2. EmailService chuẩn bị template email HTML (cấu trúc Thymeleaf), điền OTP và tên người dùng. 3. Hệ thống tạo MimeMessage. 4. Gửi email qua SMTP server (JavaMailSender). 5. Ghi log trạng thái gửi thành công.
Alternate Flow	Không có.

Exception Flow	• **4. Lỗi kết nối SMTP:** Hệ thống ném ngoại lệ EmailServiceException. Log lỗi chi tiết.

3.1.1.9. Gửi Email Thanh toán Thành công

Mục	**Nội dung**
Use Case ID	**[UC-M01-09]**
Tên Use Case	**Gửi Email Thanh toán Thành công**
Actor	System (Email Service)
Trigger	Được kích hoạt bởi Use Case "Thanh toán Đơn hàng" khi thanh toán thành công.
Description	Hệ thống gửi email xác nhận chi tiết đơn hàng và thanh toán đến khách hàng sau khi hoàn tất giao dịch.
Pre-Conditions	• Đơn hàng đã được tạo và thanh toán thành công. • Có thông tin email người nhận.
Post-Conditions	Email xác nhận đơn hàng được gửi đến hộp thư khách hàng.
Main Flow	1. Component PaymentService/OrderService gọi EmailService.sendOrderPaymentSuccessEmail(). 2. EmailService chuẩn bị data context (Mã đơn hàng, Tổng tiền, Tên người nhận, Phương thức thanh toán, Thời gian). 3. Render template HTML order-payment-success. 4. Gửi email qua SMTP server. 5. Ghi log trạng thái gửi thành công.

Alternate Flow	Không có.
Exception Flow	• **4. Lỗi kết nối SMTP:** Hệ thống ghi log lỗi nhưng không rollback giao dịch thanh toán (Email là bước phụ).

3.1.2 Functional Requirements

- **[FR-AUTH-01] Đăng ký (Register):** Người dùng Guest có thể tạo tài khoản mới bằng Email. Hệ thống phải gửi email xác thực (chứa Link hoặc OTP) để kích hoạt tài khoản.
- **[FR-AUTH-02] Đăng nhập (Login):** Người dùng đăng nhập bằng Email/Password. Hệ thống trả về cặp Access Token (ngắn hạn) và Refresh Token (dài hạn).
- **[FR-AUTH-03] Đăng nhập Google (OAuth2):** Tích hợp "Login with Google". Nếu email chưa tồn tại, hệ thống tự động tạo tài khoản mới (Auto-provisioning).
- **[FR-AUTH-04] Quên mật khẩu (Forgot Password):** Quy trình 3 bước: Yêu cầu OTP qua email -> Xác thực OTP -> Đặt mật khẩu mới.
- **[FR-AUTH-05] Làm mới Token (Refresh Token):** API cấp lại Access Token mới khi cái cũ hết hạn mà không cần người dùng đăng nhập lại.
- **[FR-AUTH-06] Đăng xuất (Logout):** Vô hiệu hóa Refresh Token (xóa khỏi Redis hoặc Database) để ngăn chặn truy cập trái phép.

3.2 Module 2: Product Management (Quản lý Sản phẩm - Admin)

Phụ trách: Võ Đức Hoàng

3.2.1 Description

Cung cấp công cụ cho Admin quản lý vòng đời sản phẩm, từ nhập liệu đến hiển thị.

3.2.1.1. Xem danh sách sản phẩm

Mục	Nội dung

Actor	Admin
Trigger	Admin truy cập trang "Quản lý sản phẩm".
Description	Hiển thị danh sách sản phẩm với chức năng phân trang, lọc (filter), và tìm kiếm (search).
Pre-Conditions	Admin đã đăng nhập thành công.
Post-Conditions	Danh sách sản phẩm được hiển thị theo cấu hình lọc/tìm kiếm.
Main Flow	1. Admin vào menu "Quản lý sản phẩm". 2. Hệ thống hiển thị dashboard stats (Tổng số, Đang hoạt động, Ngừng kinh doanh, Đã xóa, Sắp hết hàng). 3. Hiển thị bảng dữ liệu: Ảnh | Tên | SKU | Giá | Tồn kho | Trạng thái | Thao tác. 4. Hệ thống phân trang mặc định (20 items/page). 5. Admin thực hiện lọc, tìm kiếm hoặc sắp xếp. **Chi tiết chức năng Lọc (Filter):** • **Category/Brand:** Dropdown/Multi-select. • **Price Range:** Slider hoặc input min-max (0 - 50tr VND). • **Stock Level:** Tất cả | Còn hàng (>0) | Sắp hết (<10) | Hết hàng (=0). • **Status:** Active | Inactive | Deleted. • **Created Date:** Từ ngày - Đến ngày. • **Logic:** AND (thỏa mãn tất cả điều kiện). **Chi tiết chức năng Tìm kiếm (Search):** • **Fields:** Tên (fuzzy matching), SKU (exact match), Mô tả (full-text).

	• **Cơ chế:** Debounce 500ms, highlight từ khóa. **Chi tiết Sắp xếp (Sorting):** • **Default:** Ngày tạo mới nhất (Created Date DESC). • **Options:** Tên (A-Z/Z-A), Giá (Cao-Thấp), Tồn kho.
Alternate Flow	• **AF-1:** Không có sản phẩm → Hiển thị thông báo "Chưa có sản phẩm". • **AF-2:** Lọc/tìm kiếm không có kết quả → Hiển thị "Không tìm thấy kết quả phù hợp".

3.2.1.2. Thêm sản phẩm mới

Mục	Nội dung
Actor	Admin
Trigger	Admin click nút "Thêm sản phẩm mới".
Description	Tạo sản phẩm mới bao gồm thông tin cơ bản, Template (biến thể), Metadata (thông số kỹ thuật) và hình ảnh.
Pre-Conditions	Admin có quyền (Role) ADMIN.
Post-Conditions	Sản phẩm cùng các bảng liên quan (Templates, Metadata, Images) được lưu vào database.
Main Flow	1. Admin click "Thêm sản phẩm". 2. Hiển thị form nhập liệu: • Thông tin: Tên (), *Mô tả ()*, Category (), *Brand ()*. • Templates (*): SKU, Màu sắc, Bộ nhớ, RAM, Giá, Tồn kho.

	• Thông số kỹ thuật (ProductMetadata). • Ảnh (Tối đa 10 ảnh). 3. Admin nhập thông tin và thêm ít nhất 1 template (biến thể). 4. Admin nhấn "Lưu". 5. Hệ thống Validate dữ liệu (Tên 5-200 ký tự, SKU unique, Giá > 0, Stock >= 0). 6. Hệ thống tạo dữ liệu trong DB (Transaction). 7. Hiển thị thông báo "Thêm thành công!".
Exception Flow	• **EF-1: Lỗi Validation:** - Tên < 5 hoặc > 200 ký tự. - SKU sai định dạng hoặc trùng lặp. - Không có ảnh hoặc ảnh sai định dạng (chỉ chấp nhận JPG/PNG/WebP, max 5MB). • **EF-2: Lỗi Upload ảnh:** Network timeout hoặc Storage full → Rollback và báo lỗi. • **EF-3: Lỗi Database:** Connection timeout hoặc Constraint violation.
Data Specs (Metadata)	**Cấu trúc JSON lưu trữ thông số kỹ thuật:** • **Display:** screenSize (1.0-50.0 inch), screenResolution, screenTechnology, refreshRate (Hz). • **Processor:** cpuChipset, gpu. • **Camera:** cameraMegapixels, frontCameraMegapixels, cameraDetails. • **Battery:** batteryCapacity (mAh), chargingPower (W), chargingType. • **Physical:** weight (grams), dimensions, material. • **Connectivity:** 5G, Wi-Fi, Bluetooth, SimType.

• **Other:** waterResistance, audioFeatures, securityFeatures (FaceID).

3.2.1.3. Cập nhật sản phẩm

Mục	Nội dung
Actor	Admin
Trigger	Admin click nút "Edit" trên dòng sản phẩm.
Description	Chỉnh sửa thông tin sản phẩm (Tên, mô tả, templates, metadata, hình ảnh).
Pre-Conditions	Admin có quyền ADMIN.
Post-Conditions	Thông tin sản phẩm và các bảng liên quan được cập nhật.
Main Flow	1. Admin click "Edit". 2. Hệ thống gọi API GET /api/v1/admin/products/{id} để tải dữ liệu. 3. Form hiển thị với dữ liệu cũ đã được điền sẵn. 4. Admin sửa thông tin cần thiết. 5. Admin nhấn "Cập nhật". 6. Hệ thống Validate và kiểm tra thay đổi. 7. Cập nhật Database, ghi log Audit, xóa Cache. 8. Hiển thị "Cập nhật thành công!".
Alternate Flow	• **AF-1:** Admin không thay đổi gì → Thông báo "Không có thay đổi".

Exception Flow	• **EF-1: Lỗi Validation:** Tương tự như Use Case Thêm mới. • **EF-2: Xung đột dữ liệu (Concurrent modification):** Nếu sản phẩm bị sửa bởi admin khác cùng lúc (check updated_at) → Báo lỗi và yêu cầu tải lại. • **EF-3: Sản phẩm đã bị xóa:** Redirect về danh sách. • **EF-4: Lỗi xử lý ảnh:** Upload ảnh mới thất bại → Giữ nguyên ảnh cũ.

3.2.1.4. Xóa mềm sản phẩm (Soft Delete)

Mục	Nội dung
Actor	Admin
Trigger	Admin click nút "Delete".
Description	Đánh dấu sản phẩm là đã xóa (isDeleted = true) nhưng vẫn giữ dữ liệu trong DB.
Pre-Conditions	Admin có quyền ADMIN.
Post-Conditions	Trạng thái Product.isDeleted = true, sản phẩm ẩn khỏi danh sách hiển thị.
Main Flow	1. Admin xác nhận xóa. 2. Frontend gửi request DELETE /api/v1/admin/products/{id}. 3. Hệ thống kiểm tra sản phẩm tồn tại và chưa bị xóa. 4. Hệ thống kiểm tra ràng buộc (ví dụ: đơn hàng đang xử lý). 5. Cập nhật isDeleted = true. 6. Hiển thị "Xóa sản phẩm thành công".

Alternate Flow	• **AF-1:** Sản phẩm không tồn tại → Lỗi 404. • **AF-2:** Sản phẩm đã bị xóa trước đó → Lỗi 400.
Exception Flow	• **EF-2: Ràng buộc nghiệp vụ:** Có đơn hàng PENDING/SHIPPING liên quan đến sản phẩm → Chặn xóa và báo lỗi chi tiết. • **EF-3: Lỗi Database:** Update thất bại → Rollback.

3.2.1.5. Khôi phục sản phẩm

Mục	**Nội dung**
Actor	Admin
Trigger	Admin click nút "Restore" (trong tab Thùng rác/Đã xóa).
Description	Khôi phục sản phẩm đã bị xóa mềm về trạng thái hoạt động.
Pre-Conditions	Sản phẩm đang ở trạng thái đã xóa (isDeleted = true).
Post-Conditions	Trạng thái Product.isDeleted = false, sản phẩm hiển thị lại trong danh sách.
Main Flow	1. Admin gửi request POST /api/v1/admin/products/{id}/restore. 2. Hệ thống kiểm tra sản phẩm tồn tại và đang bị xóa. 3. Cập nhật isDeleted = false. 4. Lưu Database. 5. Hiển thị "Khôi phục sản phẩm thành công".
Exception Flow	• **EF-1:** Product ID không tồn tại → Lỗi 404. • **EF-2:** Sản phẩm chưa bị xóa (đang active) → Lỗi 400. • **EF-3:** Danh mục hoặc Thương hiệu gốc của sản phẩm đã bị xóa vĩnh viễn → Yêu cầu chọn lại danh mục/thương hiệu trước khi khôi phục.

3.2.1.6. Quản lý hình ảnh

3.2.1.6.1. Thay thế toàn bộ ảnh (Replace All)

Mục	Nội dung
API Endpoint	POST /api/v1/admin/products/{id}/images
Description	Xóa tất cả ảnh cũ và thêm danh sách ảnh mới (Thao tác REPLACE).
Main Flow	1. Admin gửi danh sách ảnh mới (Array gồm imageUrl, isPrimary, imageOrder). 2. Validate: Tối đa 10 ảnh, phải có duy nhất 1 ảnh isPrimary=true. 3. Hệ thống xóa toàn bộ ProductImage cũ của sản phẩm (Cascade delete). 4. Tạo ProductImage mới từ request (Insert batch). 5. Hiển thị "Cập nhật hình ảnh thành công".
Alternate Flow	• **AF-1:** Product không tồn tại → 404. • **AF-2:** Gửi quá 10 ảnh → 400 "Tối đa 10 ảnh sản phẩm".

3.2.1.6.2. Xóa 1 ảnh cụ thể

Mục	Nội dung
API Endpoint	DELETE /api/v1/admin/products/{id}/images/{imageId}
Description	Xóa một hình ảnh cụ thể của sản phẩm.
Main Flow	1. Admin gửi request xóa với productId và imageId. 2. Hệ thống kiểm tra sản phẩm phải còn nhiều hơn 1 ảnh (không được xóa ảnh cuối cùng).

	3. Thực hiện xóa bản ghi ProductImage. 4. **Logic đặc biệt:** Nếu xóa ảnh đang là isPrimary=true → Hệ thống tự động chọn ảnh khác (có imageOrder nhỏ nhất) làm ảnh chính mới. 5. Hiển thị "Xóa hình ảnh thành công".
Alternate Flow	• **AF-1:** Không tìm thấy ảnh hoặc sản phẩm → 404. • **AF-2:** Sản phẩm chỉ còn duy nhất 1 ảnh → 400 "Không thể xóa ảnh cuối cùng".

3.2.2 Functional Requirements

- **[FR-PROD-ADM-01] Danh sách sản phẩm:** Hiển thị dạng bảng (Table), hỗ trợ Phân trang (Pagination), Tìm kiếm, Lọc theo Danh mục/Trạng thái.
- **[FR-PROD-ADM-02] Thêm mới sản phẩm:** Form nhập liệu chi tiết bao gồm: Thông tin chung, Giá, Tồn kho, Upload nhiều ảnh (Thumbnail & Gallery), Thông số kỹ thuật (Dynamic JSON fields).
- **[FR-PROD-ADM-03] Chỉnh sửa sản phẩm:** Cho phép cập nhật toàn bộ thông tin. Hỗ trợ khóa/mở khóa sản phẩm (ẩn khỏi trang chủ).
- **[FR-PROD-ADM-04] Xóa mềm (Soft Delete):** Sản phẩm bị xóa sẽ đổi trạng thái sang DELETED, không mất vĩnh viễn khỏi Database để giữ lịch sử đơn hàng.

3.3 Module 3: Category & Brand Management

Phụ trách: Huỳnh Ngọc Thạch

3.3.1 Description

Quản lý các thuộc tính phân loại sản phẩm.

3.3.1.1. [UC-C1] Quản lý Danh mục

Mục	Nội dung
Use Case ID	**[UC-C1]**

Tên Use Case	**Manage Category – Quản lý danh mục**
Actor	Admin
Trigger	Actor chọn chức năng **"Quản lý Danh mục"** trên giao diện quản trị.
Description	Use case này cho phép actor quyền thêm, sửa, xóa và xem danh sách các danh mục sản phẩm.
Pre-Conditions	Actor đăng nhập tài khoản thành công vào hệ thống với quyền Admin.
Post-Conditions	• Thông tin danh mục được cập nhật trong cơ sở dữ liệu và hệ thống hiển thị các thông tin được cập nhật. • Actor xem chi tiết các danh mục.
Main Flow	**1.** Actor chọn thẻ **"Danh mục"** trên thanh bên ở trang quản lý. **2.** Hệ thống hiển thị danh sách các danh mục hiện có. **3.** Actor có thể thực hiện các thao tác sau: **3.A. Tìm kiếm danh mục:** 3.A.1. Actor nhấn vào thanh **"Tìm kiếm"**. 3.A.2. Actor nhập tên danh mục cần tìm. 3.A.3. Hệ thống hiển thị danh sách danh mục phù hợp. 3.A.4. Actor chọn danh mục từ kết quả để xem chi tiết hoặc chỉnh sửa. **3.B. Thêm danh mục:** 3.B.1. Actor nhấn nút **"Thêm danh mục"**.

3.B.2. Actor nhập thông tin (Tên, Mô tả, Danh mục cha) và nhấn **"Thêm mới"**.

3.B.3. Hệ thống kiểm tra tính hợp lệ.

3.B.4. Hệ thống lưu vào DB, thông báo thành công và về trang danh sách.

3.C. Cập nhật thông tin danh mục:

3.C.1. Actor chọn danh mục và nhấn nút **"Cập nhật"**.

3.C.2. Actor điều chỉnh thông tin và nhấn **"Cập nhật"**.

3.C.3. Hệ thống kiểm tra tính hợp lệ.

3.C.4. Hệ thống cập nhật vào DB, thông báo thành công và về trang danh sách.

3.D. Xóa danh mục:

3.D.1. Actor chọn danh mục và nhấn nút **"Xóa"**.

3.D.2. Hệ thống hỏi: **"Bạn có chắc chắn muốn xóa danh mục này không?"**.

3.D.3. Actor ấn **"Xác nhận"**.

3.D.4. Hệ thống kiểm tra ràng buộc (sản phẩm, danh mục con).

3.D.5. Hệ thống xóa khỏi DB và thông báo **"Xóa danh mục thành công"**.

3.E. Thêm danh mục con (subcategory):

3.E.1. Actor chọn danh mục cha và nhấn **"Thêm danh mục con"**.

3.E.2. Actor nhập thông tin danh mục con (Tên, Mô tả) và nhấn **"Thêm mới"**.

3.E.3. Hệ thống kiểm tra tính hợp lệ và trùng tên trong cùng danh mục cha.

	3.E.5. Hệ thống lưu và liên kết với danh mục cha. 3.E.6. Hệ thống cập nhật và hiển thị danh sách.
Alternate Flow	• **3.B.1.1.** Hệ thống yêu cầu người dùng nhập thông tin danh mục mới. • **3.C.1.1.** Hệ thống yêu cầu người dùng chỉnh sửa thông tin danh mục. • **3.E.1.1.** Hệ thống yêu cầu người dùng nhập thông tin danh mục con với danh mục cha đã được chọn.
Exception Flow	• **3.B.3.1, 3.C.3.1.** Nếu tên danh mục rỗng hoặc bị trùng: Hệ thống hiển thị nhắc nhở **"Tên danh mục không hợp lệ hoặc đã tồn tại!"**. • **3.D.4.1.** Nếu danh mục còn sản phẩm hoặc danh mục con: Hiển thị thông báo **"Không thể xóa danh mục vì đang chứa sản phẩm hoặc danh mục con!"** và chặn thao tác xóa.

3.3.1.2. UC-B1] Quản lý Thương hiệu

Mục	**Nội dung**
Use Case ID	**[UC-B1]**
Tên Use Case	**Manage Brand – Quản lý thương hiệu**
Actor	Admin
Trigger	Actor chọn chức năng **"Quản lý Thương hiệu"** trên giao diện quản trị.
Description	Use case này cho phép actor quyền thêm, sửa, xóa và xem danh sách các thương hiệu sản phẩm.
Pre-Conditions	Actor đăng nhập tài khoản thành công vào hệ thống với quyền Admin.

Post-Conditions

• Thông tin thương hiệu được cập nhật trong cơ sở dữ liệu và hệ thống hiển thị các thông tin được cập nhật.
• Actor xem chi tiết các thương hiệu.

Main Flow

1. Actor chọn thẻ **"Thương hiệu"** trên thanh bên ở trang quản lý.
2. Hệ thống hiển thị danh sách các thương hiệu hiện có.
3. Actor có thể thực hiện các thao tác sau:

3.A. Tìm kiếm thương hiệu:

3.A.1. Actor nhấn vào thanh **"Tìm kiếm"**.
3.A.2. Actor nhập tên thương hiệu cần tìm.
3.A.3. Hệ thống hiển thị danh sách thương hiệu phù hợp.
3.A.4. Actor chọn thương hiệu từ kết quả để xem chi tiết hoặc chỉnh sửa.

3.B. Thêm thương hiệu:

3.B.1. Actor nhấn nút **"Thêm thương hiệu"**.
3.B.2. Actor nhập thông tin (Tên, Logo, Mô tả) và nhấn **"Thêm mới"**.
3.B.3. Hệ thống kiểm tra tính hợp lệ.
3.B.4. Hệ thống lưu vào DB, thông báo thành công và về trang danh sách.

3.C. Cập nhật thông tin thương hiệu:

3.C.1. Actor chọn thương hiệu và nhấn nút **"Cập nhật"**.
3.C.2. Actor điều chỉnh thông tin và nhấn **"Cập nhật"**.
3.C.3. Hệ thống kiểm tra tính hợp lệ.

	3.C.4. Hệ thống cập nhật vào DB, thông báo thành công và về trang danh sách. **3.D. Xóa thương hiệu:** 3.D.1. Actor chọn thương hiệu và nhấn nút **"Xóa"**. 3.D.2. Hệ thống hỏi: **"Bạn có chắc chắn muốn xóa thương hiệu này không?"**. 3.D.3. Actor ấn **"Xác nhận"**. 3.D.4. Hệ thống kiểm tra tính hợp lệ (ràng buộc sản phẩm). 3.D.5. Hệ thống xóa khỏi DB và thông báo **"Xóa thương hiệu thành công"**.
Alternate Flow	• **3.B.1.1.** Hệ thống yêu cầu người dùng nhập thông tin thương hiệu mới. • **3.C.1.1.** Hệ thống yêu cầu người dùng chỉnh sửa thông tin thương hiệu.
Exception Flow	• **3.B.3.1, 3.C.3.1.** Nếu tên thương hiệu rỗng hoặc bị trùng: Hệ thống hiển thị nhắc nhở **"Tên thương hiệu không hợp lệ hoặc đã tồn tại!"**. • **3.D.4.1.** Nếu thương hiệu còn sản phẩm liên kết: Hiển thị thông báo **"Có sản phẩm cùng thương hiệu!"** và chặn thao tác xóa.

3.3.2 Functional Requirements

- **[FR-CAT-01] Quản lý Danh mục:** CRUD Danh mục (Ví dụ: Điện thoại, Laptop, Phụ kiện). Hỗ trợ cấu trúc phân cấp (Cha - Con) nếu cần.
- **[FR-BRAND-01] Quản lý Thương hiệu:** CRUD Thương hiệu (Apple, Samsung...). Hỗ trợ upload Logo thương hiệu.
- **[FR-CAT-02] Ràng buộc toàn vẹn:** Không cho phép xóa Danh mục/Thương hiệu nếu đang có Sản phẩm liên kết (Phải cảnh báo hoặc yêu cầu gỡ liên kết trước).

3.4 Module 4: Product Discovery (Khám phá Sản phẩm - Client)

Phụ trách: Trần Quốc Giăng

3.4.1 Description

Giao diện chính dành cho khách hàng để tìm kiếm và xem sản phẩm.

3.4.1.1. [UC-PV-01] Xem trang chủ sản phẩm

Mục	Nội dung
Mô tả	Hiển thị 4 section sản phẩm nổi bật trên trang chủ: Flash Sale, Sản phẩm Nổi bật, Bán chạy nhất, Mới nhất.
Actor	Khách hàng
Trigger	Truy cập trang chủ /
API gọi song song	• GET /products/on-sale?limit=8: Flash Sale – sản phẩm đang giảm giá • GET /products/featured?limit=8: Sản phẩm nổi bật (rating ≥ 4.5) • GET /products/best-selling?limit=8: Sản phẩm bán chạy nhất • GET /products/new-arrivals?limit=8: Sản phẩm mới nhất
Luồng chính	1. Người dùng truy cập trang chủ. 2. Hệ thống gọi 4 API song song (SWR). 3. Hiển thị 4 section với lưới ProductCard.

3.4.1.2. [UC-PV-02] Tìm kiếm sản phẩm

Mục	Nội dung
Mô tả	Cho phép khách hàng tìm kiếm sản phẩm theo từ khóa (tên, thương hiệu, model…).
Trigger	Nhập từ khóa vào ô tìm kiếm và nhấn Enter hoặc nút tìm kiếm.

API chính	GET /products/search Params ví dụ: ?keyword=iPhone 15&sortBy=created_date&sortDirection=desc&page=0&size=20
Backend xử lý	Chuẩn hóa từ khóa → xây dựng pattern LIKE linh hoạt (%iphone%15%).
Luồng chính	1. Người dùng nhập từ khóa → submit. 2. Gọi API search. 3. Hiển thị danh sách sản phẩm dạng lưới + phân trang.

3.4.1.3. [UC-PV-03] Lọc sản phẩm đa tiêu chí

Mục	**Nội dung**
Mô tả	Lọc sản phẩm theo nhiều tiêu chí kết hợp (thương hiệu, giá, RAM, bộ nhớ, pin, hệ điều hành…).
Trigger	Thay đổi bất kỳ bộ lọc nào trong sidebar lọc.
API chính	POST /products/filter (Body: JSON Filter)
Tiêu chí lọc	• categoryIds, brandIds • minPrice, maxPrice • ramOptions, storageOptions • minBattery, osOptions • inStockOnly, hasDiscountOnly • sortBy, sortDirection, page, size
Luồng chính	1. Người dùng thay đổi filter (checkbox, range slider…). 2. State filter được cập nhật. 3. Gọi API Filter với body hiện tại. 4. Backend thực hiện stream filter → sort → phân trang.

5. Cập nhật danh sách sản phẩm.

3.4.1.4. [UC-PV-04] Xem chi tiết sản phẩm

Mục	Nội dung
Mô tả	Hiển thị toàn bộ thông tin chi tiết của một sản phẩm: hình ảnh, biến thể, thông số kỹ thuật, đánh giá, sản phẩm liên quan.
Trigger	Nhấn vào ProductCard từ bất kỳ danh sách nào.
API gọi song song	• GET /products/{id}: Chi tiết sản phẩm. • GET /products/{id}/related?limit=8: Sản phẩm liên quan.
Luồng chính	1. Điều hướng đến /products/[id]. 2. Gọi song song 2 API. 3. Tự động chọn biến thể đầu tiên. 4. Hiển thị gallery, specs, reviews, related products.

3.4.1.5. [UC-PV-05] So sánh sản phẩm

Mục	Nội dung
Mô tả	Cho phép so sánh đồng thời 2–4 sản phẩm theo các thông số kỹ thuật.
Trigger	Chọn chế độ so sánh → tick sản phẩm → nhấn "So sánh ngay".
Business Rules	• Tối thiểu 2 sản phẩm. • Tối đa 4 sản phẩm. • Dưới 2 sản phẩm → redirect về trang danh sách.
API chính	POST /products/compare (Body: [1, 5, 8])
Luồng chính	1. Bật chế độ so sánh → tick checkbox. 2. Nhấn nút "So sánh ngay". 3. Điều hướng /products/compare?ids=1,5,8.

4. Gọi API compare → render bảng so sánh.

3.4.1.5. [UC-PV-06] Xem sản phẩm đặc biệt

Trang	URL	Sort mặc định	Filter đặc biệt
Bán chạy	/products/best-selling	soldCount:desc	—
Mới nhất	/products/new-arrivals	createdAt:desc	—
Nổi bật	/products/featured	rating:desc	—
Flash Sale	/products/flash-sale	discountPercentage:desc	hasDiscountOnly = true

3.4.1.6. [UC-PV-07] Thêm vào giỏ hàng

Mục	Nội dung
Mô tả	Thêm sản phẩm (có thể kèm biến thể) vào giỏ hàng từ trang danh sách hoặc chi tiết.
Trigger	Nhấn nút “Thêm vào giỏ” hoặc “Mua ngay”.
Store	CartStore (Zustand)
Hàm chính	addToCart(product), addToCartWithDetails(details), buyNowWithDetails(details).
Luồng chính	1. Nhấn nút thêm. 2. Gọi hàm store → cập nhật state + localStorage. 3. Hiển thị toast thành công.

3.4.1.7. [UC-PV-08] Thêm vào yêu thích

Mục	Nội dung
Mô tả	Thêm / xóa sản phẩm khỏi danh sách yêu thích (wishlist).
Trigger	Nhấn icon trái tim trên ProductCard hoặc trang chi tiết.
Store	WishlistStore (Zustand)
Hàm chính	toggleItem(product), isInWishlist(id).

3.4.1.8. [UC-CB-01] Gửi câu hỏi tư vấn sản phẩm

Mục	Nội dung
Mô tả	Khách hàng gửi câu hỏi ngôn ngữ tự nhiên để tư vấn sản phẩm.
Trigger	Nhập câu hỏi và gửi.
Pre-Conditions	1. Chatbot đang bật (enabled = true). 2. Truy cập trang/widget chatbot. 3. Kết nối mạng ổn định.
Post-Conditions	• Thành công: Xử lý intent, hiển thị phản hồi AI và sản phẩm. • Thất bại: Hiển thị lỗi.
Main Flow	1. Khách hàng mở chatbot → Hiển thị welcome message. 2. Nhập câu hỏi (ví dụ: "Tư vấn điện thoại chụp ảnh đẹp dưới 15 triệu"). 3. Gửi → Hiển thị loading. 4. Hệ thống gửi API /api/v1/chatbot-assistant/chat. 5. Kiểm tra trạng thái chatbot. 6. Phát hiện intent (ví dụ: FILTER_CAMERA). 7. Trích xuất thông số (ví dụ: maxPrice = 15,000,000). 8. Tìm sản phẩm từ database.

	9. Gọi Gemini AI sinh phản hồi. 10. Gemini AI trả văn bản. 11. Xây dựng response (AI + products). 12. Trả JSON cho frontend. 13. Hiển thị phản hồi và product cards.
Alternate Flow	• **AF-01 (Chatbot tắt):** Trả fallback với sản phẩm nổi bật, thông báo bảo trì. • **AF-02 (Không tìm thấy sản phẩm):** Áp dụng fallback: Top Brands → New Arrivals → Featured Products. • **AF-03 (Câu hỏi không liên quan):** Gemini AI phản hồi hướng dẫn, kèm sản phẩm nổi bật. • **AF-04 (Thương hiệu không hỗ trợ):** Thông báo và gợi ý thay thế.
Exception Flow	• **EF-01 (Quota Gemini hết):** Chuyển key dự phòng (5 keys, round-robin); nếu hết, trả error với featured products. • **EF-02 (Network error):** Thông báo thử lại. • **EF-03 (Server error):** Trả HTTP 500, hiển thị lỗi.
Business Rules	• **BR-CB-01:** Câu hỏi không rỗng. • **BR-CB-02:** Tối đa 5 sản phẩm/response. • **BR-CB-03:** Xử lý ≤ 30 giây. • **BR-CB-04:** Hỗ trợ 17 intents. • **BR-CB-05:** 5 keys Gemini, cơ chế round-robin.

3.4.1.8. [UC-CB-02] Xem gợi ý sản phẩm từ chatbot

Mục	**Nội dung**
Mô tả	Xem và tương tác với các sản phẩm được Chatbot gợi ý.
Trigger	Chatbot trả về danh sách sản phẩm.

Main Flow	1. Hiển thị product cards (tên, giá, ảnh, rating). 2. Khách hàng xem thông tin. 3. Click sản phẩm → Navigate /products/{id}. 4. Xem chi tiết sản phẩm.
Alternate Flow	• **AF-01:** Click "Thêm giỏ hàng" → Thêm vào CartStore, toast xác nhận. • **AF-02:** Hỏi thêm về sản phẩm → Gửi message với productId, chatbot trả thông tin chi tiết.

3.4.1.9. [UC-CB-03] Sử dụng Quick Actions

Mục	Nội dung
Mô tả	Sử dụng nút nhanh để hỏi mà không cần gõ phím.
Danh sách Action	• **Bán chạy:** "Xem sản phẩm bán chạy" (BEST_SELLING) • **Mới nhất:** "Xem sản phẩm mới nhất" (NEW_ARRIVALS) • **Nổi bật:** "Xem sản phẩm nổi bật" (FEATURED) • **iPhone/Samsung:** Tư vấn thương hiệu (FILTER_BRAND) • **Gaming/Camera/Giá rẻ:** Tư vấn theo nhu cầu (FILTER_...)
Main Flow	1. Hiển thị các buttons Quick Action. 2. Click button → Điền prompt tương ứng vào khung chat và gửi. 3. Thực hiện quy trình như UC-CB-01.

3.4.1.10. [UC-CB-04] Xóa lịch sử chat

Mục	Nội dung
Mô tả	Xóa toàn bộ lịch sử trò chuyện hiện tại.
Trigger	Click biểu tượng Trash (Thùng rác).
Main Flow	1. Click Trash → Hiển thị xác nhận.

	2. Xác nhận → Gọi clearChat(). 3. Reset messages, error state. 4. Hiển thị lại welcome message.
Business Rules	• **BR-CB-06:** Lưu session ở client (localStorage/state). • **BR-CB-07:** Không lưu lịch sử chat vào database.

3.4.1.11. [UC-CB-05] Bật/Tắt Chatbot

Mục	**Nội dung**
Actor	Admin
Mô tả	Quản trị viên thay đổi trạng thái hoạt động của chatbot.
Main Flow	1. Truy cập trang quản lý → Xem trạng thái. 2. Click Tắt/Bật → Xác nhận. 3. Gửi POST /admin/chatbot/disable (hoặc enable). 4. Server cập nhật file config YAML. 5. Thông báo thành công.
Business Rules	• **BR-CB-08:** Chỉ Admin mới có quyền. • **BR-CB-09:** Khi tắt → Chatbot trả về Fallback response (bảo trì). • **BR-CB-10:** Ghi log người thay đổi và thời gian.

3.4.1.12. [UC-CB-06] Xem trạng thái Chatbot

Mục	**Nội dung**
Actor	Admin
API	GET /admin/chatbot/status
Main Flow	1. Admin truy cập Dashboard. 2. Hệ thống đọc file YAML config.

3. Trả về JSON trạng thái.

4. Hiển thị: Trạng thái (On/Off), Người cập nhật, Thời gian, Thống kê requests.

3.4.1.13. [UC-CB-07] Xóa cache Chatbot

Mục	Nội dung
Actor	Admin
Mô tả	Xóa cache dữ liệu sản phẩm để chatbot cập nhật thông tin mới nhất.
Main Flow	1. Click nút Xóa cache → Xác nhận. 2. Gọi POST /chatbot-assistant/clear-cache. 3. Server xóa key trong Redis. 4. Thông báo thành công.

3.4.2 Functional Requirements

- **[FR-PROD-CLIENT-01] Trang chủ (Home):** Hiển thị Banner, Sản phẩm mới nhất, Sản phẩm bán chạy, Sản phẩm theo danh mục nổi bật.
- **[FR-PROD-CLIENT-02] Tìm kiếm & Lọc (Search & Filter):**
  - Thanh tìm kiếm (Search Bar) với gợi ý (Suggestion).
  - Bộ lọc nâng cao (Sidebar): Theo Khoảng giá, Thương hiệu, RAM, ROM, Màu sắc.
- **[FR-PROD-CLIENT-03] Chi tiết sản phẩm:** Hiển thị đầy đủ thông tin, Slide ảnh, chọn biến thể (Màu/Dung lượng), Thông số kỹ thuật chi tiết.
- **[FR-PROD-CLIENT-04] So sánh (Compare):** Cho phép chọn 2-3 sản phẩm để so sánh thông số kỹ thuật trên cùng một bảng.
- **[FR-CHATBOT-01] Chatbot AI:** Tích hợp Widget Chatbot để hỗ trợ trả lời câu hỏi cơ bản của khách hàng.

3.5 Module 5: Shopping Cart (Giỏ hàng)

Phụ trách: Lưu Trần Kim Phú

3.5.1 Description

Quản lý trạng thái mua sắm của người dùng trước khi thanh toán.

3.5.1.1. Thêm sản phẩm vào giỏ hàng

Mục	Nội dung
Use Case ID	**[UC-CART-01]**
Tên Use Case	**Thêm sản phẩm vào giỏ hàng**
Actor	Registered Customer (Khách hàng đã đăng ký)
Trigger	Actor chọn nút **"Thêm vào giỏ hàng"** trên trang sản phẩm.
Description	Use case này cho phép Actor thêm sản phẩm điện thoại vào giỏ hàng với số lượng mong muốn. Nếu sản phẩm đã có trong giỏ, hệ thống tự động cộng dồn số lượng.
Pre-Conditions	Actor đã đăng nhập thành công vào hệ thống.
Post-Conditions	• Sản phẩm được thêm vào giỏ hàng hoặc số lượng được cập nhật. • Tổng tiền giỏ hàng được tính toán lại. • Badge số lượng giỏ hàng trên header tăng lên.
Main Flow	1. Từ trang chi tiết sản phẩm, Actor chọn số lượng muốn mua (mặc định = 1). 2. Actor click nút **"Thêm vào giỏ hàng"**. 3. Hệ thống validate JWT token và xác thực người dùng. 4. Hệ thống kiểm tra sản phẩm tồn tại trong database và có status = ACTIVE.

	5. Hệ thống kiểm tra tồn kho: product.stock_quantity >= requested_quantity. 6. Hệ thống lấy giỏ hàng của Actor: 6.A. Nếu chưa có Cart → tạo mới với user_id. 6.B. Nếu đã có Cart → sử dụng Cart hiện tại. 7. Hệ thống kiểm tra sản phẩm trong giỏ: 7.A. Nếu đã có → cập nhật quantity = old_quantity + new_quantity. 7.B. Nếu chưa có → tạo CartItem mới. 8. Hệ thống lưu thay đổi vào database. 9. Hệ thống tính toán: 9.A. subtotal = price × quantity cho từng item. 9.B. totalAmount = SUM(all subtotals). 9.C. itemCount = COUNT(cart_items). 10. Hệ thống trả về CartResponse với thông tin cập nhật. 11. UI hiển thị toast "Đã thêm vào giỏ hàng" và cập nhật badge.
Alternate Flow	• **3.1.** Nếu Actor chưa đăng nhập, hệ thống chuyển hướng đến trang **Login**. Sau khi login thành công, Actor quay lại trang sản phẩm. • **5.1.** Nếu stock_quantity < requested_quantity, hệ thống hiển thị thông báo **"Chỉ còn X sản phẩm trong kho"** và disable nút "Thêm vào giỏ". • **7.A.1.** Nếu old_quantity + new_quantity > 10, hệ thống hiển thị **"Chỉ được mua tối đa 10 sản phẩm"** và không thêm. • **6.A.1.** Nếu Actor vừa đăng nhập và có guest cart trong localStorage, hệ thống merge items từ guest cart vào user cart, validate tồn kho, xóa guest cart và hiển thị **"Đã đồng bộ X sản phẩm từ giỏ hàng tạm"**.
Exception Flow	• **4.1.** Nếu sản phẩm không tồn tại hoặc bị xóa, hệ thống trả về 404 Not Found và hiển thị **"Sản phẩm không tồn tại"**.

• **8.1.** Nếu database timeout (> 5s), hệ thống trả về 503 Service Unavailable, hiển thị **"Hệ thống đang bận, vui lòng thử lại"** và lưu draft vào localStorage.

• **8.2.** Nếu database transaction fail (UNIQUE constraint, FK constraint), hệ thống ROLLBACK và trả về 500 với message **"Có lỗi xảy ra, vui lòng thử lại"**.

• **7.A.2.** Nếu có concurrent update (optimistic locking conflict), hệ thống phát hiện version mismatch, trả về 409 Conflict với state mới nhất, UI reload giỏ hàng và highlight items thay đổi.

• **3.2.** Nếu backend service unavailable, UI retry 3 lần (5s interval). Nếu vẫn fail, lưu request vào IndexedDB và background worker retry mỗi 60s.

3.5.1.2. Xem giỏ hàng

Mục	Nội dung
Use Case ID	**[UC-CART-02]**
Tên Use Case	**Xem giỏ hàng**
Actor	Registered Customer.
Trigger	Actor chọn icon **"Giỏ hàng"** trên header hoặc truy cập /cart.
Description	Use case này cho phép Actor xem danh sách đầy đủ các sản phẩm trong giỏ hàng, bao gồm hình ảnh, tên, giá, số lượng, thành tiền và tổng tiền.
Pre-Conditions	Actor đã đăng nhập thành công vào hệ thống.

Post-Conditions	• Danh sách sản phẩm trong giỏ được hiển thị đầy đủ. • Trạng thái tồn kho được cập nhật realtime. • Tổng tiền được tính toán chính xác.
Main Flow	1. Actor click icon **"Giỏ hàng"** trên header hoặc truy cập /cart. 2. Hệ thống validate JWT token. 3. Hệ thống thực hiện JOIN query lấy cart + cart_items + products: SELECT c.*, ci.*, p.* FROM carts c LEFT JOIN cart_items ci ON c.id = ci.cart_id LEFT JOIN products p ON ci.product_id = p.id WHERE c.user_id = ? 4. Hệ thống kiểm tra tồn kho cho từng sản phẩm: 4.A. Nếu product.stock = 0 → đánh dấu "Hết hàng". 4.B. Nếu cart_item.quantity > product.stock → highlight warning. 5. Hệ thống tính toán subtotal và totalAmount. 6. Hệ thống trả về CartResponse. 7. UI hiển thị: 7.A. Danh sách sản phẩm với image, name, price, quantity, subtotal. 7.B. Nút +/- để điều chỉnh số lượng. 7.C. Nút "Xóa" từng sản phẩm. 7.D. Tổng tiền tạm tính và nút **"Thanh toán"**.
Alternate Flow	• **3.1.** Nếu giỏ hàng trống (không có cart_items), hệ thống hiển thị empty state với icon giỏ hàng trống, message **"Giỏ hàng của bạn đang trống"** và button **"Tiếp tục mua sắm"**. • **4.A.1.** Nếu product không còn tồn tại (bị admin xóa), hệ thống tự động DELETE cart_item đó và hiển thị toast **"1 sản phẩm đã bị xóa khỏi giỏ hàng"**.

	• **5.1.** Nếu giá sản phẩm thay đổi, UI hiển thị badge **"Giá đã thay đổi"** cạnh sản phẩm và tính toán với giá mới nhất.
Exception Flow	• **3.1.** Nếu database query timeout (> 5s), hệ thống cancel query và trả về 504 Gateway Timeout, UI hiển thị **"Không thể tải giỏ hàng, vui lòng thử lại"** và nút "Retry". • **3.2.** Nếu partial data load (cart_items OK nhưng products NULL), hệ thống filter items có product = NULL, DELETE orphan cart_items và hiển thị **"Một số sản phẩm không còn tồn tại đã bị xóa"**. • **6.1.** Nếu có concurrent view từ nhiều thiết bị với cart version khác nhau, hệ thống trả về cart với lastModified timestamp, UI compare với localStorage, hiển thị version mới nhất và subscribe WebSocket để nhận realtime updates.

3.5.1.3. Cập nhật số lượng sản phẩm

Mục	**Nội dung**
Use Case ID	**[UC-CART-03]**
Tên Use Case	**Cập nhật số lượng sản phẩm**
Actor	Registered Customer.
Trigger	Actor click nút **tăng (+) / giảm (-)** số lượng hoặc nhập số lượng trực tiếp.
Description	Use case này cho phép Actor thay đổi số lượng sản phẩm trong giỏ hàng. Nếu số lượng giảm xuống 0, sản phẩm tự động bị xóa.

Pre-Conditions	Actor đang ở trang giỏ hàng và sản phẩm đã có trong giỏ.
Post-Conditions	• Số lượng sản phẩm được cập nhật. • Tổng tiền được tính lại. • Nếu quantity = 0 → sản phẩm bị xóa khỏi giỏ.
Main Flow	1. Actor click nút **(+)** hoặc **(-)** để thay đổi số lượng. 2. UI cập nhật số lượng tạm thời (optimistic update). 3. UI gửi request: PUT /api/cart/items/{itemId} với {quantity: new_value}. 4. Hệ thống validate JWT token. 5. Hệ thống kiểm tra quyền sở hữu.
Alternate Flow	• **6.1.** Nếu new_quantity > stock_quantity, hệ thống trả về 400 Bad Request với availableStock, UI rollback về số lượng cũ và hiển thị **"Chỉ còn X sản phẩm"**. • **7.B.1.** Khi Actor giảm quantity xuống 0, hệ thống tự động DELETE cart_item, UI remove item khỏi UI với animation slide-out và hiển thị **"Đã xóa sản phẩm"**. • **5.1.** Nếu cart_item không thuộc về Actor (query không trả về kết quả), hệ thống trả về 403 Forbidden và UI hiển thị **"Không có quyền thực hiện thao tác này"**. • **1.1.** Nếu Actor nhập số lượng trực tiếp, UI debounce 500ms sau lần nhập cuối mới gửi request để tránh spam.
Exception Flow	• **2.1.** Nếu server reject do hết hàng, UI rollback optimistic update về quantity cũ, flash red border trên input và hiển thị error message.

• **7.1.** Nếu có version conflict (concurrent update từ tab/device khác), hệ thống trả về 409 Conflict với {currentQuantity: 5, requestedQuantity: 3, message: "..."}, UI reload cart và highlight item với animation blink.
• **7.2.** Nếu database deadlock, hệ thống retry transaction tối đa 3 lần với exponential backoff. Sau 3 lần fail → trả về 500 và hiển thị **"Không thể cập nhật, vui lòng thử lại"**.
• **3.1.** Nếu network interruption mid-request, request timeout sau 30s, UI hiển thị **"Mất kết nối mạng"**, rollback optimistic update, lưu pending request vào queue và tự động retry.

3.5.1.4. Xóa sản phẩm khỏi giỏ hàng

Mục	Nội dung
Use Case ID	**[UC-CART-04]**
Tên Use Case	**Xóa sản phẩm khỏi giỏ hàng**
Actor	Registered Customer.
Trigger	Actor chọn nút **"Xóa"** trên sản phẩm trong giỏ hàng.
Description	Use case này cho phép Actor xóa một sản phẩm cụ thể khỏi giỏ hàng. Hệ thống hiển thị confirm dialog trước khi xóa và hỗ trợ undo trong 5 giây.
Pre-Conditions	Actor đang ở trang giỏ hàng và sản phẩm có trong giỏ.

Post-Conditions	• Sản phẩm bị xóa khỏi giỏ. • Tổng tiền được cập nhật. • itemCount giảm đi 1.
Main Flow	1. Actor click nút **"Xóa"** (icon thùng rác) trên sản phẩm. 2. UI hiển thị confirm dialog: 2.A. Message: "Bạn có chắc muốn xóa sản phẩm này?" 2.B. Button: "Hủy" và "Xóa". 3. Actor click **"Xóa"** để xác nhận. 4. UI gửi request: DELETE /api/cart/items/{itemId}. 5. Hệ thống validate JWT token. 6. Hệ thống kiểm tra quyền sở hữu. 7. Hệ thống xóa: DELETE FROM cart_items WHERE id = ?. 8. Hệ thống tính lại totalAmount và itemCount. 9. Hệ thống trả về CartResponse. 10. UI remove item khỏi danh sách với animation fade-out. 11. UI hiển thị toast **"Đã xóa sản phẩm"** với nút **"Hoàn tác"** (5s countdown).
Alternate Flow	• **3.1.** Nếu Actor click **"Hủy"** trong confirm dialog, UI đóng dialog và không có thay đổi nào. • **11.1.** Nếu Actor click **"Hoàn tác"** trong 5s, hệ thống thêm lại sản phẩm vào giỏ với số lượng cũ. Nếu trong lúc đó Actor đã thêm lại sản phẩm từ trang product, hệ thống phát hiện duplicate và trả về 409 Conflict với message **"Sản phẩm đã có trong giỏ hàng"**.
Exception Flow	• **6.1.** Nếu item đã bị xóa bởi tab/device khác (race condition), hệ thống trả về 404 Not Found, UI silent remove item khỏi UI (không hiển thị error vì kết quả mong muốn đã đạt).

• **7.1.** Nếu database CASCADE delete fail do related table bị lock, hệ thống timeout sau 5s, ROLLBACK transaction và trả về 500 với message **"Không thể xóa, vui lòng thử lại sau"**.

3.5.1.5. Xóa toàn bộ giỏ hàng

Mục	Nội dung
Use Case ID	**[UC-CART-05]**
Tên Use Case	**Xóa toàn bộ giỏ hàng**
Actor	Registered Customer.
Trigger	Actor chọn nút **"Xóa tất cả"** ở header/footer giỏ hàng.
Description	Use case này cho phép Actor xóa tất cả sản phẩm trong giỏ hàng bằng một thao tác. Do tính nghiêm trọng, yêu cầu nhập "XOA" để xác nhận.
Pre-Conditions	Giỏ hàng có ít nhất 1 sản phẩm.
Post-Conditions	• Tất cả cart_items bị xóa. • Giỏ hàng trống (totalAmount = 0, itemCount = 0). • Cart entity vẫn tồn tại (chỉ xóa items).
Main Flow	1. Actor click nút **"Xóa tất cả"**. 2. UI hiển thị confirm dialog nghiêm trọng: 2.A. Warning icon + Message: "Bạn có chắc muốn xóa TOÀN BỘ giỏ hàng?" 2.B. Input field yêu cầu nhập "XOA" (case-insensitive).

	2.C. Button "Hủy" và "Xóa tất cả" (màu đỏ, disable cho đến khi nhập đúng). 3. Actor nhập **"XOA"** để xác nhận. 4. Actor click **"Xóa tất cả"**. 5. UI gửi request: DELETE /api/cart/clear. 6. Hệ thống validate JWT token. 7. Hệ thống lấy cart của Actor. 8. Hệ thống xóa tất cả items: DELETE FROM cart_items WHERE cart_id = ?. 9. Hệ thống trả về EmptyCartResponse. 10. UI hiển thị empty state. 11. UI hiển thị toast **"Đã xóa toàn bộ giỏ hàng"**.
Alternate Flow	• **3.1.** Nếu Actor không nhập hoặc nhập sai "XOA", nút "Xóa tất cả" bị disable và Actor không thể proceed. • **8.1.** Sau khi đặt hàng thành công (trigger khác từ Order Module), hệ thống tự động clear cart mà không cần confirm (silent delete). • **8.1.** Nếu Actor có 100+ items (edge case), DELETE query timeout. Hệ thống chuyển sang batch delete: DELETE FROM cart_items WHERE cart_id = ? LIMIT 50, lặp lại cho đến khi xóa hết. UI hiển thị progress bar "Đang xóa... 50/100".
Exception Flow	• **8.2.** Nếu có concurrent clear và add (Actor click "Xóa tất cả" trên Desktop, đồng thời admin thêm voucher item vào giỏ từ hệ thống), hệ thống dùng distributed lock (Redis) với TTL 5s để lock cart trước khi clear, đảm bảo không có race condition. • **8.3.** Nếu có background job đang tạo order_items reference đến cart_items, DELETE vi phạm FK constraint, database ROLLBACK

transaction và trả về 409 Conflict với message "Không thể xóa vì có đơn hàng đang xử lý". UI disable nút "Xóa tất cả" trong 30s.

3.5.2 Functional Requirements

- **[FR-CART-01] Thêm vào giỏ:** Từ trang chi tiết hoặc danh sách. Kiểm tra tồn kho trước khi thêm.
- **[FR-CART-02] Xem giỏ hàng:** Hiển thị danh sách item, cho phép cập nhật số lượng (+/-), xóa item. Tự động tính tạm tính (Subtotal).
- **[FR-CART-03] Lưu trữ thông minh:**
    - **Guest:** Lưu giỏ hàng trong LocalStorage hoặc Redis (theo Session ID).
    - **Member:** Đồng bộ giỏ hàng vào Database/Redis khi đăng nhập (Merge giỏ hàng Guest vào Member).

3.6 Module 6: Checkout & Payment (Đặt hàng & Thanh toán)

Phụ trách: Huỳnh Hữu Huy

3.6.1 Description

Quy trình chuyển đổi từ Giỏ hàng thành Đơn hàng.

3.6.1.1. [UC-14] Pay order - Thanh toán đơn hàng

Mục	**Nội dung**
Use Case ID	**[14]**
Tên Use Case	**Pay order**
Actor	Registered Customer

Trigger	Actor chọn các sản phẩm muốn mua trong giỏ hàng và thực hiện thanh toán.
Description	Use case này cho phép actor thực hiện thanh toán các sản phẩm cần mua với các phương thức thanh toán được hỗ trợ từ cửa hàng. Đồng thời, actor cũng có thể sử dụng các mã khuyến mãi trong quá trình thanh toán.
Pre-Conditions	• Actor đã đăng nhập vào hệ thống. • Giỏ hàng có ít nhất 1 sản phẩm. • Sản phẩm còn đủ tồn kho.
Post-Conditions	• Đơn hàng được tạo thành công với Order Code duy nhất (format: ORD_YYMMDDHHMMSS). • OrderItems được tạo cho tất cả sản phẩm trong đơn hàng. • CartItems tương ứng được xóa khỏi giỏ hàng. **Nếu COD/BANK_TRANSFER:** • Order status = CONFIRMED (Đã xác nhận). • Payment được tạo với: provider = null, status = SUCCESS, amount = totalAmount. • Tồn kho sản phẩm giảm ngay (trừ từ ProductTemplate sequentially). • Email xác nhận đơn hàng được gửi. **Nếu VNPAY:** • Order status = PENDING (Chờ thanh toán). • Payment được tạo với: provider = VNPAY, status = PENDING. • Tồn kho CHƯA giảm (sẽ giảm sau khi callback thành công). • PaymentUrl được trả về để redirect.
Main Flow	1. Actor vào chức năng "Thanh toán" bằng một trong các cách:

1.A. Chọn nút "Thanh toán" từ trang Giỏ hàng.

1.B. Chọn nút "Mua ngay" trên một sản phẩm từ trang Danh sách sản phẩm.

1.C. Chọn nút "Mua ngay" khi xem Chi tiết sản phẩm.

2. Hệ thống hiển thị thông tin về sách, số lượng, đơn giá, thành tiền, tên khách hàng, số điện thoại, email và địa chỉ.

3. Actor có thể thay đổi địa chỉ giao hàng.

4. Actor có thể áp dụng mã khuyến mãi:

4.1. Nhập mã giảm giá sản phẩm (promotionId).

4.2. Nhập mã miễn phí vận chuyển (freeshippingPromotionId).

4.3. Hệ thống kiểm tra promotion có tồn tại không.

4.4. Hiện tại chỉ validate tồn tại, chưa tính discount (TODO).

5. Actor chọn phương thức thanh toán (COD/BANK_TRANSFER/VNPAY).

6. Actor có thể để lại lời nhắn.

7. Hệ thống hiển thị tóm tắt đơn hàng (tổng tiền sách, phí vận chuyển, giảm giá, voucher, tổng thanh toán).

8. Actor nhấn chọn "Đặt hàng".

9. Hệ thống tạo đơn hàng (Order) với trạng thái:

- COD/BANK_TRANSFER: Order status = CONFIRMED.
- VNPAY: Order status = PENDING.

10. Hệ thống tạo các OrderItem cho đơn hàng.

11. **Nếu paymentMethod = COD hoặc BANK_TRANSFER:**

11.1. Giảm tồn kho ngay lập tức.

11.2. Tạo Payment record (provider=null, status=SUCCESS).

11.3. Gửi email xác nhận.

11.4. Xóa CartItem khỏi giỏ hàng.

11.5. Hiển thị thông báo thành công và chuyển đến chi tiết đơn hàng.

	12. **Nếu paymentMethod = VNPAY:** 12.1. KHÔNG giảm tồn kho ngay. 12.2. Tạo Payment record (status=PENDING). 12.3. Tạo URL thanh toán VNPay (gọi UC [15]). 12.4. Trả về paymentUrl. 12.5. Redirect Actor đến trang VNPay. 12.6. Actor thanh toán trên VNPay. 12.7. Chuyển sang UC [16] "Handle VNPay Payment Callback". 13. Đơn hàng được lưu vào database.
Alternate Flow	• Nếu Actor chưa đăng nhập, hệ thống chuyển đến trang đăng nhập.
Exception Flow	• Nếu mã khuyến mãi không hợp lệ: Hệ thống bỏ qua mã và ghi log warning. • Nếu sản phẩm không đủ tồn kho: Thông báo "Sản phẩm {tên_sản_phẩm} chỉ còn {số_lượng} sản phẩm" và dừng thanh toán. • Nếu COD/BANK_TRANSFER không đủ tồn kho khi trừ: Throw BadRequestException và rollback transaction. • Nếu gửi email/xóa CartItem thất bại: Ghi log error nhưng KHÔNG làm thất bại việc tạo order.

3.6.1.2. [UC-15] Create VNPay Payment URL

Mục	Nội dung
Use Case ID	**[15]**

Tên Use Case	**Create VNPay Payment URL**
Actor	Registered Customer
Trigger	Hệ thống tự động gọi use case này trong quá trình tạo đơn hàng khi paymentMethod = VNPAY. (Sub-flow tự động).
Description	Use case này cho phép hệ thống tạo URL thanh toán từ cổng VNPay để thực hiện giao dịch online. Hệ thống sẽ tạo một liên kết an toàn có chữ ký điện tử để chuyển hướng actor sang trang thanh toán của VNPay.
Pre-Conditions	• Actor đã đăng nhập vào hệ thống. • Đơn hàng đã được tạo với paymentMethod = VNPAY. • Order có status = PENDING. • Cấu hình VNPay đã được thiết lập (TMN_CODE, HASH_SECRET, URL).
Post-Conditions	• Payment record được tạo với status = PENDING. • Hệ thống trả về URL thanh toán VNPay hợp lệ. • Actor được chuyển hướng đến trang thanh toán VNPay.
Main Flow	1. Hệ thống nhận request tạo payment URL (orderId, amount, orderInfo, locale, ipAddress). 2. Kiểm tra trạng thái đơn hàng (phải là PENDING). 3. Lấy thông tin đơn hàng. 4. Đọc cấu hình VNPay (TmnCode, HashSecret, Url, ReturnUrl, IpnUrl). 5. Xây dựng tham số thanh toán VNPay: - Version, Command, TmnCode, Amount (x100), CurrCode, CreateDate, ExpireDate, TxnRef (OrderCode), ReturnUrl, IpAddr... 6. Sắp xếp tham số theo alphabet.

	7. Tạo chuỗi query string. 8. Tạo chữ ký bảo mật (HMAC SHA512) từ query string + HashSecret. 9. Kiểm tra Payment tồn tại: 9.1. Nếu có Payment PENDING: Reuse payment cũ. 9.2. Nếu chưa có hoặc khác PENDING: Tạo Payment mới (provider=VNPAY, status=PENDING). 10. Ghép URL hoàn chỉnh: {vnp_Url}?{query_string}&vnp_SecureHash={hash}. 11. Trả về paymentUrl, orderId, amount. 12. Actor được redirect đến VNPay. 13. Actor nhập thông tin thẻ/ngân hàng. 14. VNPay xử lý và gọi UC "Handle Callback".
Alternate Flow	• **Nếu Actor hủy thanh toán trên VNPay:** VNPay gọi callback với code != "00". Hệ thống cập nhật Payment = FAILED, Order = CANCELLED. Redirect Actor về trang kết quả với thông báo hủy.
Exception Flow	• **Order không phải PENDING:** Thông báo lỗi. • **Đã có Payment PENDING:** Reuse payment cũ, vẫn tạo URL mới (do URL cũ có thể hết hạn). • **Order đã SUCCESS:** Thông báo "Đơn hàng đã được thanh toán". • **Thiếu cấu hình/Lỗi tạo chữ ký/Lỗi DB/Lỗi mạng:** Thông báo lỗi tương ứng. • **Số tiền không khớp:** Thông báo lỗi dữ liệu.

3.6.1.3. [UC-16] Handle VNPay Payment Callback (IPN)

Mục	Nội dung

Use Case ID	**[16]**
Tên Use Case	**Handle VNPay Payment Callback (IPN)**
Actor	Payment Gateway (VNPay Server) - Server-to-server communication
Trigger	Sau khi khách hàng thanh toán trên VNPay, VNPay server tự động gửi HTTP request callback đến URL webhook của hệ thống.
Description	Xử lý kết quả thanh toán từ VNPay thông qua IPN. Hệ thống nhận dữ liệu, kiểm tra chữ ký bảo mật, xác thực giao dịch, cập nhật trạng thái Payment/Order, giảm tồn kho nếu thành công và lưu log audit.
Pre-Conditions	• Payment record status = PENDING. • Order status = PENDING. • URL callback đã được cấu hình và đăng ký với VNPay.
Post-Conditions	**Thanh toán thành công & Đủ tồn kho:** • Payment status = SUCCESS, transaction_id lưu. • Order status = CONFIRMED. • Giảm tồn kho. • Gửi email xác nhận. **Thanh toán thành công NHƯNG Hết tồn kho:** • Payment status = SUCCESS (để hoàn tiền). • Order status = CANCELLED. • Note: "Payment successful but out of stock". • KHÔNG giảm tồn kho. **Thanh toán thất bại/hủy:**

• Payment status = FAILED.

• Order status = CANCELLED.

Main Flow

1. VNPay gửi GET request đến endpoint callback.

2. Hệ thống extract query parameters vào Map.

3. Lấy vnp_SecureHash từ request.

4. Loại bỏ các hash fields khỏi Map.

5. Tạo secure hash (checksum) để verify chữ ký.

6. Lấy thông tin giao dịch (TxnRef, TransactionNo, ResponseCode, Status, Amount).

7. Tìm Order theo vnp_TxnRef. Nếu không thấy → Throw Exception.

8. Tìm hoặc tạo Payment record (nếu chưa có).

9. **KIỂM TRA SIGNATURE:**

- Nếu Invalid: Throw BadRequestException, Dừng xử lý.
- Nếu Valid: Tiếp tục.

10. **IDEMPOTENCY CHECK:** Nếu Order đã CONFIRMED → Log info, cập nhật TransactionId nếu thiếu, trả về success (Dừng xử lý).

11. Kiểm tra ResponseCode & TransactionStatus:

11.A. Thành công ("00"):

11.A.1. Payment.status = SUCCESS.

11.A.2. Kiểm tra tồn kho (Stock Validation).

11.A.3. Xử lý tồn kho:

- Nếu thiếu stock: Order = CANCELLED, KHÔNG trừ kho.
- Nếu đủ stock: Order = CONFIRMED, Trừ kho (Sequential Deduction).

11.A.4. Gửi email xác nhận.

11.B. Thất bại/Hủy (!="00"):

11.B.1. Payment.status = FAILED.

	11.B.2. Order.status = CANCELLED. 11.B.3. KHÔNG giảm tồn kho. 12. Lưu Payment và Order vào database. 13. Tạo và lưu PaymentCallbackLog để audit. 14. Trả về response 200 OK cho VNPay.
Alternate Flow	• **Callback trùng lặp:** Hệ thống phát hiện Order đã Confirmed, chỉ cập nhật TransactionId (nếu cần) và trả về Success. Không trừ kho/gửi mail lần 2.
Exception Flow	• **Chữ ký không hợp lệ:** Trả về HTTP 400, Transaction rollback. • **Không tìm thấy Order:** Trả về HTTP 404, Transaction rollback. • **Lỗi lưu Log:** Log error nhưng vẫn tiếp tục flow chính (cập nhật Order/Payment). • **Lỗi Database/System:** Transaction rollback toàn bộ, trả về HTTP 500. VNPay sẽ retry sau.

3.6.1.4. [UC-17] View Payment History

Mục	**Nội dung**
Use Case ID	**[17]**
Tên Use Case	**View Payment History**
Actor	Registered Customer
Trigger	Actor chọn chức năng 'Lịch sử thanh toán' hoặc 'Giao dịch của tôi' trong trang cá nhân.

Description	Cho phép actor xem danh sách tất cả các giao dịch thanh toán đã thực hiện (COD, BANK_TRANSFER, VNPAY). Xem chi tiết mã đơn hàng, số tiền, trạng thái, thời gian.
Pre-Conditions	• Actor đã đăng nhập. • Actor có ít nhất một đơn hàng đã được tạo.
Post-Conditions	• Hiển thị danh sách giao dịch. • Thông tin chi tiết (Mã GD, Mã đơn, Phương thức, Số tiền, Trạng thái, Ngày). • Hỗ trợ phân trang.
Main Flow	1. Actor vào trang cá nhân -> "Lịch sử thanh toán". 2. Hệ thống lấy userId từ JWT. 3. Kiểm tra quyền truy cập. 4. Query database lấy danh sách Payment theo userId (có phân trang). 5. Hiển thị bảng danh sách giao dịch: - Cột: Mã giao dịch, Mã đơn hàng, Phương thức, Mã GD VNPay, Số tiền, Trạng thái, Ngày tạo. 6. Hiển thị thông tin phân trang (Page, Total pages, Next/Prev). 7. Actor có thể: 7.A. Chuyển trang. 7.B. Click vào Mã đơn hàng để xem chi tiết. 8. Xử lý chuyển trang: Gọi API update danh sách.
Alternate Flow	• **Chưa có giao dịch:** Hiển thị "Bạn chưa có giao dịch thanh toán nào" và nút "Mua sắm ngay".
Exception Flow	• **Token hết hạn:** Trả về 401, Redirect login. • **Truy cập dữ liệu người khác:** Trả về 403 Forbidden. • **Lỗi Database:** Trả về 500, hiển thị thông báo lỗi hệ thống.

• **Giao dịch bị xóa:** Bỏ qua và thông báo "Giao dịch không tồn tại".

3.6.2 Functional Requirements

- **[FR-CHECKOUT-01] Checkout Flow:** Quy trình 3 bước: Thông tin giao hàng -> Chọn vận chuyển/Thanh toán -> Xác nhận.
- **[FR-CHECKOUT-02] Phương thức thanh toán:**
  - **COD:** Thanh toán khi nhận hàng.
  - **VNPay:** Tích hợp cổng thanh toán (Sandbox). Chuyển hướng sang VNPay và xử lý IPN (Callback) để cập nhật trạng thái đơn hàng tự động.
- **[FR-CHECKOUT-03] Xử lý đồng thời:** Sử dụng Transaction Database để đảm bảo trừ tồn kho và tạo đơn hàng xảy ra đồng thời (Atomic).

3.7 Module 7: Order Management (Quản lý Đơn hàng)

Phụ trách: Nguyễn Tuấn Huy

3.7.1 Description

Theo dõi vòng đời đơn hàng cho cả Khách hàng và Admin.

3.7.1.1. Xem lịch sử & chi tiết đơn hàng của khách hàng

Mục	**Nội dung**
Use case	**Xem lịch sử & chi tiết đơn hàng (Registered Customer)**
Actor	Registered Customer
Trigger	Actor đăng nhập và chọn mục “Đơn hàng của tôi” trên giao diện người dùng.

Description	Use case này cho phép khách hàng đã đăng ký xem danh sách các đơn hàng của chính họ và xem chi tiết từng đơn hàng. Hệ thống lấy dữ liệu từ các bảng đơn hàng (orders), chi tiết đơn hàng (order_items) và sản phẩm (products).
Pre-Conditions	• Actor đã đăng nhập hệ thống với vai trò khách hàng. • Trong hệ thống có thể đã tồn tại 0 hoặc nhiều đơn hàng gắn với tài khoản của Actor.
Post-Conditions	• Danh sách đơn hàng của đúng khách hàng được hiển thị trên giao diện. • Khi xem chi tiết, hệ thống chỉ đọc dữ liệu, không làm thay đổi bất kỳ thông tin nào của đơn hàng.
Main Flow	1. Actor đăng nhập vào hệ thống. 2. Actor chọn mục "Đơn hàng của tôi". 3. Hệ thống xác định tài khoản hiện tại và truy xuất tất cả đơn hàng thuộc khách hàng này từ cơ sở dữ liệu. 4. Hệ thống hiển thị danh sách đơn hàng: mã đơn (orderCode), ngày tạo, tổng tiền, trạng thái, phương thức thanh toán… 5. Actor chọn một đơn hàng cụ thể trong danh sách. 6. Hệ thống kiểm tra quyền truy cập, đảm bảo đơn hàng thuộc khách hàng hiện tại. 7. Hệ thống lấy chi tiết đơn hàng tương ứng (thông tin người nhận, địa chỉ, danh sách sản phẩm, tổng tiền, trạng thái, phương thức thanh toán…). 8. Hệ thống hiển thị đầy đủ chi tiết đơn hàng cho Actor, bao gồm cả mã QR của đơn (nếu có).
Alternate Flow	**AF-1: Khách chưa có đơn hàng nào**

	• Ở bước 3, nếu không tìm thấy đơn hàng nào, hệ thống hiển thị danh sách rỗng. • Giao diện hiển thị thông báo: "Bạn chưa có đơn hàng nào." và gợi ý khách quay về trang sản phẩm.
Exception Flow	**EX-1: Phiên đăng nhập không hợp lệ hoặc hết hạn** • Khi Actor truy cập "Đơn hàng của tôi", nếu phiên làm việc không còn hợp lệ, hệ thống chuyển hướng về màn hình đăng nhập. **EX-2: Đơn hàng không thuộc khách hiện tại** • Ở bước 6, nếu đơn hàng không thuộc tài khoản đang đăng nhập, hệ thống từ chối và hiển thị thông báo "Không tìm thấy đơn hàng.", giữ nguyên tại danh sách. **EX-3: Lỗi hệ thống** • Nếu có lỗi khi truy xuất dữ liệu, hệ thống hiển thị thông báo chung "Có lỗi xảy ra, vui lòng thử lại sau." và không hiển thị danh sách/chi tiết.

3.7.1.2. Quản lý & cập nhật trạng thái đơn hàng (Admin)

Mục	**Nội dung**
Use case	**Quản lý & cập nhật trạng thái đơn hàng (Admin)**
Actor	Admin
Trigger	Actor chọn menu "Quản lý Đơn hàng" trên giao diện quản trị.

Description	Use case này cho phép Admin xem danh sách đơn hàng, xem chi tiết từng đơn và cập nhật trạng thái xử lý của đơn (pending → processing → shipped → delivered hoặc cancelled).
Pre-Conditions	• Actor đã đăng nhập hệ thống với vai trò Admin. • Trong cơ sở dữ liệu đã có các đơn hàng (bảng orders). • Với luồng quét QR tại cửa hàng: khách hàng có mã QR hợp lệ của đơn (từ hệ thống).
Post-Conditions	• Thông tin đơn hàng và trạng thái hiện tại được hiển thị đúng trên giao diện quản trị. • Khi Actor cập nhật trạng thái, giá trị trạng thái của đơn hàng trong cơ sở dữ liệu được thay đổi theo quy tắc nghiệp vụ.
Main Flow	1. Actor đăng nhập vào khu vực quản trị. 2. Actor chọn chức năng "Quản lý Đơn hàng". 3. Hệ thống truy xuất danh sách đơn hàng và hiển thị theo các tiêu chí mặc định (ví dụ: sắp xếp theo ngày tạo, lọc theo trạng thái). 4. Actor có thể sử dụng các bộ lọc (mã đơn, email/điện thoại khách, trạng thái, khoảng thời gian…) để thu hẹp danh sách. 5. Hệ thống áp dụng bộ lọc và cập nhật lại danh sách đơn hàng. 6. Actor chọn một đơn bất kỳ để xem chi tiết. 7. Hệ thống lấy thông tin chi tiết đơn hàng (thông tin khách, địa chỉ, danh sách sản phẩm, tổng tiền, trạng thái hiện tại, hình thức thanh toán, ngày tạo/cập nhật…) và hiển thị cho Actor. 8. Actor bấm nút "Cập nhật trạng thái", chọn trạng thái mới phù hợp (ví dụ: từ pending sang processing, từ shipped sang delivered, hoặc cancelled) và xác nhận.

	9. Hệ thống kiểm tra tính hợp lệ của bước chuyển trạng thái (không cho phép chuyển từ trạng thái cuối cùng như delivered/cancelled sang trạng thái xử lý khác). 10. Nếu hợp lệ, hệ thống cập nhật trạng thái đơn hàng trong cơ sở dữ liệu, ghi nhận thời gian cập nhật và (tuỳ thiết kế) lưu log lịch sử. 11. Hệ thống hiển thị thông báo "Cập nhật trạng thái thành công" và làm mới thông tin đơn trên giao diện.
Alternate Flow	Không có
Exception Flow	**EX-1: Actor không đủ quyền Admin** • Khi truy cập khu vực quản trị hoặc chức năng quản lý đơn hàng, nếu Actor không có quyền phù hợp, hệ thống chặn truy cập và hiển thị thông báo lỗi. **EX-2: Đơn hàng không tồn tại** • Khi Actor chọn xem chi tiết một đơn hoặc tra cứu đơn bằng QR nhưng cơ sở dữ liệu không có bản ghi tương ứng, hệ thống hiển thị "Không tìm thấy đơn hàng." **EX-3: Chuyển trạng thái không hợp lệ** • Khi Actor chọn trạng thái mới không nằm trong chuỗi xử lý hợp lệ (ví dụ chuyển từ delivered về shipped), hệ thống hiển thị thông báo "Trạng thái đơn hàng không hợp lệ." và không cập nhật. **EX-4: Mã QR/Thông tin tra cứu không khớp**

• Trong luồng quét QR, nếu hệ thống không thể tìm thấy đơn phù hợp với dữ liệu trên mã QR, hệ thống hiển thị "Mã QR không hợp lệ hoặc đơn hàng không tồn tại."

EX-5: Lỗi hệ thống

• Nếu xảy ra lỗi khi truy xuất/cập nhật dữ liệu, hệ thống hiển thị thông báo lỗi chung và giữ nguyên trạng thái đơn hàng.

3.7.1.3. Hiển thị mã QR Code cho đơn hàng

Mục	Nội dung
Use case	**Hiển thị mã QR Code cho đơn hàng**
Actor	Registered Customer, Admin
Trigger	• Sau khi khách hàng đặt hàng thành công. • Hoặc Actor mở màn hình chi tiết của một đơn hàng cụ thể.
Description	Use case này cho phép hệ thống sinh và hiển thị một mã QR dành cho từng đơn hàng. Mã QR chứa thông tin/đường dẫn giúp tra cứu nhanh đơn hàng hoặc hỗ trợ xác nhận đơn tại cửa hàng.
Pre-Conditions	• Đơn hàng đã được tạo và lưu trong bảng orders với mã đơn (order_code) và thông tin liên hệ của khách. • Hệ thống có chức năng sinh mã QR từ chuỗi dữ liệu đầu vào.
Post-Conditions	• Mã QR của đơn hàng được hiển thị trên giao diện chi tiết đơn cho khách hàng và/hoặc Admin. • Khách hàng có thể lưu hoặc trình mã QR này cho nhân viên khi đến cửa hàng.

Main Flow	1. Sau khi hoàn tất đặt hàng, hệ thống chuyển khách đến màn hình hiển thị thông tin đơn hàng hoặc cho phép khách mở chi tiết đơn trong phần "Đơn hàng của tôi". 2. Khi hiển thị chi tiết đơn, hệ thống chuẩn bị dữ liệu định danh đơn hàng (ví dụ: order_code và email/định danh khách hàng). 3. Hệ thống tạo một chuỗi dữ liệu hoặc đường dẫn dùng để tra cứu đơn hàng dựa trên thông tin định danh đó. 4. Chức năng sinh QR nhận chuỗi dữ liệu/đường dẫn và tạo ra ảnh mã QR tương ứng. 5. Hệ thống hiển thị ảnh mã QR trong khu vực "Mã QR đơn hàng" trên giao diện chi tiết đơn. 6. Khi Admin xem chi tiết đơn trên giao diện quản trị, hệ thống cũng hiển thị mã QR tương tự để nhân viên có thể quét nếu cần.
Alternate Flow	**AF-1: Mã QR được tạo sẵn và lưu trữ** • Thay vì sinh QR mỗi lần mở chi tiết, hệ thống có thể sinh mã QR ngay khi đơn được tạo và lưu đường dẫn/ảnh QR trong cơ sở dữ liệu. • Khi người dùng mở chi tiết đơn, hệ thống chỉ cần tải ảnh QR đã lưu sẵn lên giao diện.
Exception Flow	**EX-1: Lỗi khi sinh hoặc tải mã QR** • Nếu việc sinh ảnh QR thất bại hoặc ảnh QR không tồn tại, hệ thống hiển thị thông báo "Hiện tại không thể hiển thị mã QR, vui lòng thử lại sau." nhưng không ảnh hưởng đến thông tin đơn hàng. **EX-2: Đơn hàng không tồn tại hoặc không đủ dữ liệu định danh** • Nếu hệ thống không xác định được mã đơn hoặc thông tin cần thiết, chức năng hiển thị QR sẽ bị vô hiệu và hệ thống có thể ẩn phần hiển thị mã QR trên giao diện.

3.7.1.4. Tra cứu đơn hàng công khai (không cần đăng nhập)

Mục	Nội dung
Use case	**Tra cứu đơn hàng công khai (không cần đăng nhập)**
Actor	Unregistered Customer / Registered Customer
Trigger	• Actor truy cập trang "Tra cứu đơn hàng" trên website. • Hoặc Actor quét mã QR của đơn hàng, trình duyệt mở trang tra cứu kèm sẵn một số thông tin.
Description	Use case này cho phép người dùng không cần đăng nhập vẫn có thể tra cứu trạng thái và thông tin tóm tắt của một đơn hàng, thông qua việc cung cấp mã đơn và thông tin liên hệ (ví dụ: email).
Pre-Conditions	• Đơn hàng đã tồn tại trong hệ thống với mã đơn (order_code) và thông tin liên hệ (email/điện thoại) tương ứng. • Trang tra cứu đơn hàng công khai đã được cấu hình và có thể kết nối tới cơ sở dữ liệu.
Post-Conditions	• Nếu thông tin tra cứu hợp lệ, hệ thống hiển thị trạng thái và thông tin tóm tắt của đơn hàng cho người dùng. • Không có thao tác nào từ use case này thay đổi dữ liệu đơn hàng.
Main Flow	1. Actor truy cập trang "Tra cứu đơn hàng". 2. Hệ thống hiển thị form với các trường: Mã đơn hàng (orderCode) và Email (hoặc trường thông tin liên hệ khác được yêu cầu). 3. Actor nhập đầy đủ thông tin và bấm nút "Tra cứu". 4. Hệ thống kiểm tra dữ liệu nhập vào (không để trống, đúng định dạng cơ bản của email/mã đơn).

	5. Hệ thống tìm kiếm trong cơ sở dữ liệu đơn hàng có mã đơn và thông tin liên hệ trùng khớp. 6. Nếu tìm thấy, hệ thống lấy thông tin tóm tắt của đơn hàng: mã đơn, ngày đặt, tổng tiền, trạng thái hiện tại, phương thức nhận hàng/thanh toán, người nhận, khu vực giao hàng… 7. Hệ thống hiển thị kết quả tra cứu cho Actor trên giao diện, kèm theo (tuỳ chọn) một timeline trạng thái rút gọn.
Alternate Flow	**AF-1: Tra cứu thông qua mã QR** 1. Actor sử dụng camera hoặc ứng dụng quét mã QR của đơn hàng. 2. Nội dung QR chứa sẵn mã đơn và (có thể) email hoặc một đường dẫn có tham số tương đương. 3. Trình duyệt mở trang tra cứu và hệ thống tự động điền trước thông tin mã đơn/email hoặc đọc dữ liệu từ đường dẫn. 4. Hệ thống thực hiện các bước kiểm tra và tìm kiếm trong cơ sở dữ liệu tương tự bước 4–6 của luồng chính. 5. Kết quả tra cứu được hiển thị như luồng chính.
	EX-1: Dữ liệu nhập không hợp lệ • Nếu Actor bỏ trống hoặc nhập sai định dạng các trường bắt buộc, hệ thống hiển thị thông báo lỗi và yêu cầu nhập lại, không thực hiện tra cứu.
Exception Flow	**EX-2: Không tìm thấy đơn hàng** • Nếu không có đơn hàng nào trùng khớp với mã đơn và thông tin liên hệ, hệ thống hiển thị thông báo “Không tìm thấy đơn hàng, vui lòng kiểm tra lại thông tin.”
	EX-3: Dữ liệu từ mã QR không hợp lệ

	• Trong luồng tra cứu bằng QR, nếu dữ liệu đọc được không đầy đủ hoặc không khớp với bất kỳ đơn hàng nào, hệ thống hiển thị thông báo "Mã QR hoặc liên kết tra cứu không hợp lệ." và gợi ý người dùng nhập lại bằng form. **EX-4: Lỗi hệ thống** • Nếu xảy ra lỗi khi xử lý tra cứu, hệ thống hiển thị thông báo chung "Có lỗi xảy ra khi tra cứu đơn hàng, vui lòng thử lại sau."

3.7.1.5. Hủy đơn hàng (Customer Cancel Order)

Mục	Nội dung
Use case	**Hủy đơn hàng (Customer Cancel Order)**
Actor	Registered Customer
Trigger	Actor mở "Đơn hàng của tôi", chọn một đơn hàng đang ở trạng thái cho phép hủy và bấm nút "Hủy đơn hàng".
Description	Use case này cho phép khách hàng đã đăng ký chủ động hủy một đơn hàng của chính họ khi đơn hàng vẫn đang ở trạng thái chờ xác nhận (pending), chưa được hệ thống hoặc Admin xử lý/giao hàng. Sau khi hủy thành công, trạng thái đơn được chuyển sang cancelled.
Pre-Conditions	• Actor đã đăng nhập hệ thống với vai trò khách hàng. • Đơn hàng thuộc về tài khoản của Actor. • Đơn hàng đang ở trạng thái pending (Chờ xác nhận). Đây là trạng thái duy nhất cho phép khách hàng tự hủy.

Post-Conditions	• Nếu hủy thành công: trạng thái đơn hàng được chuyển sang cancelled, thời gian cập nhật được ghi nhận. • Nếu việc hủy không hợp lệ (do trạng thái đã thay đổi…), đơn hàng giữ nguyên trạng thái ban đầu.
Main Flow	1. Actor đăng nhập và đi tới mục "Đơn hàng của tôi". 2. Hệ thống hiển thị danh sách các đơn hàng của Actor, bao gồm trạng thái hiện tại của từng đơn. 3. Actor chọn một đơn hàng đang ở trạng thái pending và bấm nút "Xem chi tiết". 4. Hệ thống hiển thị chi tiết đơn hàng, kèm nút "Hủy đơn hàng" (vì đơn đang ở trạng thái cho phép hủy). 5. Actor bấm "Hủy đơn hàng". 6. Hệ thống hiển thị hộp thoại xác nhận, thông báo rõ: "Bạn có chắc muốn hủy đơn này? Sau khi hủy, đơn không thể khôi phục." với các lựa chọn "Xác nhận" và "Không". 7. Actor bấm "Xác nhận". 8. Hệ thống kiểm tra lại trạng thái hiện tại của đơn hàng để đảm bảo đơn vẫn đang ở trạng thái pending. 9. Nếu trạng thái vẫn là pending, hệ thống cập nhật trạng thái đơn sang cancelled và ghi nhận thời gian cập nhật (và lý do hủy nếu có). 10. Hệ thống hiển thị thông báo "Hủy đơn hàng thành công" và cập nhật giao diện chi tiết đơn hiển thị trạng thái "Đã hủy".
Alternate Flow	**AF-1: Khách hàng đổi ý, không hủy nữa** 1. Ở bước 6, Actor bấm "Không" hoặc đóng hộp thoại xác nhận. 2. Hệ thống đóng hộp thoại, giữ nguyên trạng thái đơn hàng (pending) và không có thay đổi nào khác.

	EX-1: Trạng thái đơn không còn cho phép hủy • Ở bước 8, nếu hệ thống phát hiện đơn hàng đã chuyển sang processing, shipped, delivered hoặc cancelled, hệ thống không cập nhật sang cancelled. • Hệ thống hiển thị thông báo: "Đơn hàng đã được xử lý, không thể hủy." và giữ nguyên trạng thái mới nhất của đơn.
Exception Flow	**EX-2: Đơn hàng không thuộc khách hiện tại** • Nếu đơn không còn thuộc về tài khoản đang đăng nhập, hệ thống từ chối thao tác và hiển thị thông báo: "Không tìm thấy đơn hàng."
	EX-3: Lỗi hệ thống khi cập nhật • Nếu xảy ra lỗi trong quá trình ghi dữ liệu, hệ thống thông báo: "Hủy đơn hàng không thành công, vui lòng thử lại sau." và giữ nguyên trạng thái đơn ban đầu (pending).

3.7.2 Functional Requirements

- **[FR-ORDER-CLIENT-01] Lịch sử đơn hàng:** Khách hàng xem danh sách đơn hàng đã đặt, lọc theo trạng thái.
- **[FR-ORDER-CLIENT-02] Tra cứu đơn hàng (Tracking):** Trang tra cứu công khai (dành cho Guest) bằng Mã đơn hàng + Số điện thoại/Email.
- **[FR-ORDER-ADM-01] Xử lý đơn hàng:** Admin xem danh sách, chi tiết đơn. Có quyền chuyển trạng thái: PENDING -> CONFIRMED -> SHIPPING -> COMPLETED / CANCELLED.
- **[FR-ORDER-03] QR Code:** Tạo mã QR cho mỗi đơn hàng để hỗ trợ tra cứu nhanh hoặc check-in tại cửa hàng (nếu có).

3.8 Module 8: Reviews & Interaction (Đánh giá & Tương tác)

Phụ trách: Hoàng Danh

3.8.1 Description

Hệ thống Social Proof giúp tăng độ tin cậy cho sản phẩm.

3.8.1.1. Đặc tả Use Case: Đăng/Sửa/Xóa Đánh giá Sản phẩm

Mục	Nội dung
Use case	**Đăng/Sửa/Xóa Đánh giá Sản phẩm** (Create/Edit/Delete Review)
Actor	**Registered Customer** (Khách hàng đã đăng ký)
Trigger	Actor xem chi tiết đơn hàng hoặc chi tiết sản phẩm và chọn hành động: "Viết đánh giá", "Sửa đánh giá" hoặc "Xóa đánh giá".
Description	Cho phép Actor viết nhận xét, chấm số sao (1-5), đính kèm hình ảnh/video cho sản phẩm **đã mua và hoàn thành đơn hàng**. Actor cũng có thể sửa hoặc xóa đánh giá của chính mình.
Pre-Conditions	- Actor đã đăng nhập vào hệ thống. - Sản phẩm tồn tại trong hệ thống. - **Quan trọng:** Actor phải có đơn hàng chứa sản phẩm này ở trạng thái **"Đã hoàn thành"** (Completed/Delivered).
Post-Conditions	**Trường hợp Đăng/Sửa:** - Bản ghi đánh giá được tạo mới hoặc cập nhật trong cơ sở dữ liệu. - Rating trung bình của sản phẩm được tính toán lại. **Trường hợp Xóa:** - Bản ghi đánh giá bị xóa (hoặc đánh dấu xóa mềm). - Rating trung bình của sản phẩm được tính toán lại.

Main Flow

(Luồng Đăng đánh giá)

1. Actor nhấn nút "Viết đánh giá".
2. Hệ thống kiểm tra điều kiện (Pre-Conditions).
3. Hệ thống hiển thị form đánh giá.
4. Actor chọn số sao (1-5), nhập nội dung đánh giá và tải lên hình ảnh/video (nếu có).
5. Actor nhấn nút "Gửi đánh giá".
6. Hệ thống validate dữ liệu (số sao bắt buộc, định dạng file, dung lượng file).
7. Hệ thống lưu đánh giá vào cơ sở dữ liệu.
8. Hệ thống tính lại điểm Rating trung bình của sản phẩm.
9. Hệ thống hiển thị thông báo thành công và cập nhật danh sách đánh giá trên giao diện.

Alternate Flow

AF-1: Sửa đánh giá

1. Actor nhấn nút "Sửa đánh giá" trên review cũ của mình.
2. Hệ thống hiển thị form với dữ liệu cũ (sao, nội dung, ảnh).
3. Actor thay đổi thông tin và nhấn "Cập nhật".
4. Tiếp tục từ bước 6 của Main Flow.

AF-2: Xóa đánh giá

1. Actor nhấn nút "Xóa đánh giá" của chính mình.
2. Hệ thống hiển thị hộp thoại xác nhận: "Bạn có chắc chắn muốn xóa đánh giá này?".
3. Actor chọn "Đồng ý".
4. Hệ thống xóa đánh giá và tính lại điểm Rating trung bình.
5. Hệ thống thông báo thành công và ẩn đánh giá khỏi giao diện.

Exception Flow	**EX-1: Chưa đăng nhập** - Hệ thống hiển thị thông báo yêu cầu đăng nhập hoặc chuyển hướng sang trang Login. **EX-2: Không đủ điều kiện (Chưa mua/Đơn chưa hoàn thành)** - Hệ thống ẩn nút "Viết đánh giá" hoặc hiển thị thông báo: "Bạn chỉ có thể đánh giá sau khi đã mua và nhận hàng thành công." **EX-3: Dữ liệu không hợp lệ** - Actor không chọn số sao hoặc nội dung quá ngắn/quá dài. - Hệ thống hiển thị lỗi ngay trên form: "Vui lòng chọn số sao và nhập nội dung đánh giá." **EX-4: Lỗi Upload Media** - File quá lớn (>5MB) hoặc sai định dạng (không phải ảnh/video). - Hệ thống báo lỗi: "File không hợp lệ. Vui lòng tải ảnh/video dưới 5MB." **EX-5: Lỗi Server/API** - Quá trình lưu thất bại do lỗi hệ thống. - Hệ thống hiển thị: "Có lỗi xảy ra. Vui lòng thử lại sau."

3.8.1.2. Đặc tả Use Case: Tương tác Đánh giá (Like/Unlike)

Mục	Nội dung
Use case	**Tương tác Like/Unlike Đánh giá** (Review Interaction)
Actor	**Registered Customer** (Khách hàng đã đăng ký)

Trigger	Actor xem danh sách đánh giá và nhấn nút "Hữu ích" (Like) hoặc "Không hữu ích" (Unlike).
Description	Cho phép Actor thể hiện sự đồng tình hoặc không đồng tình với đánh giá của người khác. Mỗi người dùng chỉ được chọn 1 trạng thái tại 1 thời điểm.
Pre-Conditions	- Actor đã đăng nhập. - Đánh giá cần tương tác đang hiển thị và tồn tại.
Post-Conditions	- Trạng thái tương tác của Actor được cập nhật. - Số lượng Like/Unlike của đánh giá thay đổi tương ứng.
Main Flow	**(Trường hợp chưa tương tác -> Click Like)** 1. Actor nhấn nút **Like**. 2. Hệ thống kiểm tra trạng thái đăng nhập. 3. Hệ thống ghi nhận trạng thái **Like** cho Actor đối với đánh giá này. 4. Hệ thống tăng số lượng Like (+1) trên giao diện tức thời (Real-time UI update). 5. Hệ thống gọi API cập nhật xuống cơ sở dữ liệu ngầm (Background sync).
Alternate Flow	**AF-1: Hủy tương tác (Toggle Off)** - Actor đang Like, nhấn lại nút **Like** lần nữa. - Hệ thống hủy trạng thái Like. - Số lượng Like giảm (-1). **AF-2: Đổi trạng thái (Switch State)** - Actor đang Like, nhấn nút **Unlike**. - Hệ thống hủy Like (Like -1) và ghi nhận Unlike (Unlike +1). - Trạng thái chuyển từ Like sang Unlike.

	EX-1: Chưa đăng nhập - Khi nhấn nút, hệ thống hiển thị thông báo: "Vui lòng đăng nhập để thực hiện chức năng này."
Exception Flow	**EX-2: Lỗi kết nối (API Fail)** - Nếu gọi API cập nhật thất bại, hệ thống hoàn tác (rollback) số lượng Like/Unlike trên giao diện về trạng thái cũ và báo lỗi nhỏ (Toast message).

3.8.1.3. Đặc tả Use Case: Xem và Lọc Đánh giá

Mục	**Nội dung**
Use case	**Xem và Lọc Đánh giá** (View & Filter Reviews)
Actor	**Tất cả người dùng** (Guest & Registered User)
Trigger	Actor truy cập trang chi tiết sản phẩm và cuộn xuống phần Đánh giá.
Description	Hiển thị danh sách đánh giá, tóm tắt điểm số, biểu đồ phân bố sao. Cho phép lọc theo số sao, hình ảnh và sắp xếp.
Pre-Conditions	- Sản phẩm tồn tại trong hệ thống.
Post-Conditions	- Danh sách đánh giá được hiển thị theo đúng bộ lọc/sắp xếp Actor chọn.
Main Flow	1. Actor cuộn đến phần Đánh giá. 2. Hệ thống lấy dữ liệu tóm tắt (Điểm trung bình, tổng số đánh giá, biểu đồ sao). 3. Hệ thống lấy danh sách đánh giá (mặc định: Mới nhất).

	4. Actor chọn bộ lọc (ví dụ: "5 sao", "Có hình ảnh") hoặc sắp xếp (ví dụ: "Hữu ích nhất"). 5. Hệ thống gọi API lấy lại danh sách đánh giá dựa trên tiêu chí lọc. 6. Hệ thống cập nhật danh sách hiển thị (dùng AJAX/React State không tải lại cả trang).
Alternate Flow	**AF-1: Sản phẩm chưa có đánh giá** - Hệ thống kiểm tra tổng số đánh giá = 0. - Hệ thống hiển thị thông báo thân thiện: "Sản phẩm này chưa có đánh giá nào. Hãy là người đầu tiên đánh giá!" (thay vì để trống). **AF-2: Tải thêm (Load more/Pagination)** - Actor cuộn xuống cuối danh sách hoặc nhấn "Xem thêm". - Hệ thống tải các đánh giá tiếp theo và nối vào danh sách hiện tại.
Exception Flow	**EX-1: Lỗi tải dữ liệu** - Hệ thống không lấy được danh sách đánh giá do lỗi Server. - Hiển thị thông báo: "Không thể tải đánh giá lúc này."

3.8.1.4. Đặc tả Use Case: Quản lý Đánh giá (Admin)

Mục	Nội dung
Use case	**Quản lý Đánh giá** (Review Moderation)
Actor	**Administrator** (Quản trị viên)
Trigger	Admin truy cập trang CMS quản lý đánh giá hoặc nhận thông báo về đánh giá bị báo cáo.

Description	Cho phép Admin xem danh sách đánh giá, lọc các đánh giá vi phạm và thực hiện ẩn hoặc xóa đánh giá.
Pre-Conditions	- Admin đã đăng nhập và có quyền quản lý nội dung.
Post-Conditions	- Đánh giá vi phạm bị ẩn hoặc xóa khỏi hệ thống hiển thị (Frontend).
Main Flow	1. Admin vào trang Quản lý Đánh giá. 2. Hệ thống hiển thị danh sách, hỗ trợ lọc theo: Số sao, Sản phẩm, **Trạng thái báo cáo**. 3. Admin chọn một đánh giá vi phạm chính sách (spam, ngôn từ đả kích...). 4. Admin chọn hành động: **"Ẩn đánh giá"** (Unpublish) hoặc **"Xóa đánh giá"**. 5. Hệ thống yêu cầu xác nhận. 6. Hệ thống cập nhật trạng thái đánh giá. 7. Hệ thống (Backend) tính toán lại Rating trung bình của sản phẩm tương ứng.
Exception Flow	**EX-1: Lỗi xử lý** - Hệ thống báo lỗi khi Admin cố gắng xóa đánh giá không tồn tại hoặc lỗi DB. - Hiển thị thông báo: "Thao tác thất bại."

3.8.2 Functional Requirements

- **[FR-REVIEW-01] Viết đánh giá:** Chỉ cho phép đánh giá khi người dùng **đã mua sản phẩm** và đơn hàng ở trạng thái "Hoàn thành".

- **[FR-REVIEW-02] Hiển thị đánh giá:** Hiển thị sao trung bình, danh sách bình luận phân trang.
- **[FR-REVIEW-03] Admin kiểm duyệt:** Admin có quyền ẩn/xóa các bình luận spam hoặc vi phạm tiêu chuẩn cộng đồng.

3.9 Module 9: User Management (Quản lý Người dùng - Admin)

Phụ trách: Trần Thị Thanh Trang

3.9.1 Description

Công cụ quản trị người dùng hệ thống.

3.9.1.1. Manage User - Quản lý người dùng

Mục	Nội dung
Use Case ID	**[10.1]**
Tên Use Case	**Manage User - Quản lý người dùng**
Actor	Admin
Trigger	Actor chọn chức năng "Quản lý người dùng" trong trang quản lý.
Description	Use case này cho phép actor quản lý tài khoản người dùng: xem danh sách tài khoản (khách hàng và quản trị viên), khóa/gỡ khóa tài khoản người dùng (không thể khoá tài khoản quản trị viên), tìm kiếm tài khoản khách hàng, tạo tài khoản mới.
Pre-Conditions	• Actor đăng nhập tài khoản thành công vào hệ thống. • Actor có quyền Admin.
Post-Conditions	• Danh sách người dùng được cập nhật. • Trạng thái tài khoản được thay đổi (Active/Locked).

• Tài khoản mới được tạo trong hệ thống (nếu có).

Main Flow

1. Actor chọn biểu tượng "Quản lý người dùng" trên thanh menu trang quản lý.
2. Hệ thống hiển thị danh sách các tài khoản đã tạo.
3. Actor thực hiện các thao tác:

3.A. Xem chi tiết/Tìm kiếm tài khoản:

3.A.1. Actor nhập ID hoặc tên vào ô tìm kiếm.

3.A.2. Hệ thống hiển thị thông tin chi tiết tài khoản tương ứng.

3.B. Khóa tài khoản khách hàng:

3.B.1. Actor bấm vào nút "Khóa" kế bên tài khoản.

3.B.2. Hệ thống hiển thị hộp thoại xác nhận.

3.B.3. Actor nhấn "Xác nhận".

3.C. Gỡ khóa tài khoản khách hàng:

3.C.1. Actor bấm vào nút "Mở khóa" kế bên tài khoản bị khóa.

3.C.2. Hệ thống hiển thị hộp thoại xác nhận.

3.C.3. Actor nhấn "Xác nhận".

3.D. Tạo tài khoản mới:

3.D.1. Actor nhấn nút "Tạo tài khoản mới".

3.D.2. Hệ thống hiển thị form đăng ký (Email, Mật khẩu, Họ tên, SĐT, Giới tính, Ngày sinh, Loại tài khoản).

3.D.3. Actor điền thông tin và nhấn "Tạo tài khoản".

3.D.4. Hệ thống kiểm tra hợp lệ và tạo tài khoản mới.

Alternate Flow	**A1: Lọc danh sách theo loại tài khoản** • A1.1. Tại bước 2, actor chọn bộ lọc (Khách hàng/Quản trị viên hoặc Đang hoạt động/Đã khóa). • A1.2. Hệ thống hiển thị danh sách đã lọc.
Exception Flow	**E1: Không tìm thấy kết quả** • Nếu tìm kiếm không có kết quả phù hợp, hệ thống thông báo "Không có người dùng nào". **E2: Lỗi tạo tài khoản** • Nếu email đã tồn tại: Hiển thị "Email đã tồn tại trong hệ thống". • Nếu mật khẩu không hợp lệ: Hiển thị lỗi xác thực.

3.9.1.2. Xem bảng Thống kê (Dashboard)

Mục	**Nội dung**
Use Case ID	**[10.2]**
Tên Use Case	**Xem bảng Thống kê (Dashboard)**
Actor	Admin
Trigger	Actor đăng nhập thành công và truy cập trang chủ quản trị.
Description	Use case này cho phép actor xem tổng quan hệ thống với các chỉ số thống kê, biểu đồ trực quan về doanh thu, đơn hàng, người dùng và sản phẩm.
Pre-Conditions	• Actor đăng nhập tài khoản thành công với quyền ADMIN. • Hệ thống đã có dữ liệu để thống kê.

Post-Conditions	• Actor nắm bắt được tình hình hoạt động hệ thống. • Dữ liệu thống kê được hiển thị chính xác theo thời gian thực hoặc cache.
Main Flow	1. **Truy cập Dashboard:** Sau khi đăng nhập, actor được điều hướng đến trang Dashboard (hoặc click menu "Dashboard"). 2. **Hiển thị thẻ tổng quan (Summary Cards):** • Tổng doanh thu. • Tổng đơn hàng. • Tổng sản phẩm. • Tổng người dùng. 3. **Hiển thị danh sách nhanh:** • Danh sách đơn hàng gần đây. • Danh sách sản phẩm sắp hết hàng. 4. **Hiển thị biểu đồ phân tích:** 4.1. **Biểu đồ doanh thu:** Dạng đường (Line chart), hiển thị xu hướng 30 ngày. 4.2. **Biểu đồ đơn hàng:** Dạng tròn (Pie chart), hiển thị tỷ lệ theo trạng thái. 4.3. **Biểu đồ người dùng mới:** Dạng đường, hiển thị tăng trưởng theo tuần/tháng. 4.4. **Top 5 sản phẩm bán chạy:** Biểu đồ ngang (Bar chart). 5. **Tương tác:** 5.1. Actor di chuột vào biểu đồ để xem chi tiết số liệu. 5.2. Actor chọn khoảng thời gian lọc (7 ngày, 30 ngày, v.v.).
Alternate Flow	**A1: Làm mới dữ liệu** • Actor click nút "Refresh", hệ thống tải lại dữ liệu mới nhất.

Exception Flow	**A2: Thay đổi khoảng thời gian lọc** • Actor chọn bộ lọc thời gian (Tuần/Tháng/Quý). • Hệ thống render lại biểu đồ theo dữ liệu lọc. **E1: Không có dữ liệu** • Nếu hệ thống mới (chưa có data), hiển thị thông báo "Chưa có dữ liệu thống kê" hoặc hiển thị biểu đồ rỗng. **E2: Lỗi tải dữ liệu (API Error)** • Nếu API timeout hoặc lỗi server, hiển thị thông báo "Không thể tải dữ liệu, vui lòng thử lại" và nút "Reload".

3.9.2 Functional Requirements

- **[FR-USER-ADM-01] Danh sách người dùng:** Xem danh sách, tìm kiếm theo Email/SĐT.
- **[FR-USER-ADM-02] Khóa tài khoản (Ban):** Chuyển trạng thái ACTIVE -> LOCKED (ngăn đăng nhập).
- **[FR-DASHBOARD-01] Dashboard tổng quan:** Hiển thị các chỉ số Key Metrics: Tổng doanh thu, Đơn hàng mới, Top sản phẩm bán chạy (Sử dụng thư viện Chart).

3.10 Module 10: Promotion & Vouchers (Khuyến mãi)

Phụ trách: Nguyễn Văn Quang Duy

3.10.1 Description

Quản lý chiến dịch marketing qua mã giảm giá.

3.10.1.1. Apply Promotion - Áp dụng khuyến mãi

Mục	Nội dung
Use Case ID	**[29]**
Tên Use Case	**Apply Promotion (Áp dụng khuyến mãi)**
Actor	Registered Customer
Trigger	Actor chọn Promotion muốn áp dụng, sau đó chọn chức năng "Áp dụng".
Description	Use Case này cho phép Actor chọn một khuyến mãi có sẵn để nhận ưu đãi giảm giá đơn hàng. Lưu ý: Có 2 loại Voucher: • **Voucher miễn phí ship:** Chi phí vận chuyển sẽ được miễn/giảm. • **Voucher giảm giá đơn hàng:** Giảm theo phần trăm nhất định ghi trong mô tả. Với mỗi loại, Actor chỉ được phép chọn tối đa một Voucher để áp dụng cho đơn hàng.
Pre-Conditions	Actor đã đăng nhập vào hệ thống.
Post-Conditions	Actor áp dụng được ưu đãi mà Promotion mang lại cho tổng chi phí đơn hàng.
Main Flow	1. Actor đã xác định đơn hàng và chuẩn bị thanh toán (chưa chọn Voucher). Hệ thống kiểm tra tính hợp lệ của danh sách Voucher: 1.A. Kiểm tra hạn sử dụng. 1.B. Kiểm tra trạng thái (phải "Đang hoạt động"). 1.C. Kiểm tra giá trị đơn hàng tối thiểu. 2. Actor nhìn thấy danh sách Voucher hợp lệ.

	3. Actor chọn Voucher (có thể chọn 1 Freeship + 1 Giảm giá đơn hàng, hoặc 1 trong 2, hoặc không chọn). 4. Actor ấn nút "Áp dụng". 5. Hệ thống tính toán và hiển thị mức ưu đãi dựa trên Voucher đã chọn.
Alternate Flow	• **3.1.** Danh sách Voucher trống vì đơn hàng không thỏa mãn điều kiện nào. • **3.2.** Đơn hàng thỏa mãn điều kiện nhưng Actor quyết định không áp dụng Voucher nào.
Exception Flow	• **1.A, 1.B, 1.C:** Nếu Voucher vi phạm các điều kiện này, hệ thống sẽ ẩn Voucher và không cho phép Actor chọn.

3.10.1.2. Quản lý Promotion

Mục	**Nội dung**
Use Case ID	**[30]**
Tên Use Case	**Manage Promotion (Quản lý khuyến mãi)**
Actor	Administrator
Trigger	Actor chọn chức năng "Quản lý khuyến mãi" trên giao diện chính.
Description	Cho phép Actor tìm kiếm, thêm mới, sửa, xóa, kích hoạt/vô hiệu hóa Promotion. Có 2 loại khuyến mãi: • **Discount:** Giảm giá trực tiếp trên sản phẩm.

	• **Voucher:** Áp dụng trên hóa đơn (Freeship hoặc giảm tiền/phần trăm tổng hóa đơn).
Pre-Conditions	Actor đã đăng nhập và có quyền truy cập chức năng quản lý Promotion.
Post-Conditions	Thông tin Promotion được cập nhật trong cơ sở dữ liệu và hiển thị lên hệ thống.
Main Flow	1. Actor truy cập chức năng "Quản lý khuyến mãi". 2. Actor chọn loại khuyến mãi cần thao tác: **TRƯỜNG HỢP 1: DISCOUNT (Khuyến mãi sản phẩm)** **2.Dis.1.** Actor nhấn nút "Discount". **2.Dis.2.** Hiển thị danh sách Discount. Actor có thể xem chi tiết. **2.Dis.3. Thêm mới:** - Nhập thông tin: Mã giảm giá, tiêu đề, % giảm, mô tả, thời gian (ngày/giờ bắt đầu & kết thúc). - Chọn đối tượng áp dụng: + Sản phẩm cụ thể (Tìm và chọn). + Brand (Thương hiệu). + Categories (Danh mục). + SubCategories (Danh mục con). **2.Dis.4. Xóa/Vô hiệu hóa:** - Chọn Discount → Chọn "Vô hiệu hóa". - Xác nhận: *"Bạn có chắc muốn vô hiệu hóa Discount này không?"* → Hệ thống xóa. **2.Dis.5. Chỉnh sửa:** - Chọn Discount → Chọn "Chỉnh sửa". - Cập nhật các trường thông tin → Nhấn "Lưu".

	TRƯỜNG HỢP 2: VOUCHER (Khuyến mãi hóa đơn) **2.Vou.1.** Actor nhấn nút "Voucher". **2.Vou.2.** Hiển thị danh sách Voucher. **2.Vou.3. Thêm mới:** - Nhập thông tin: Mã, tiêu đề, % giảm, giá trị đơn tối thiểu, số lượng, mô tả, loại (Freeship/Giảm giá), thời gian. **2.Vou.4. Xóa/Vô hiệu hóa:** - Chọn Voucher → Chọn "Vô hiệu hóa". - Xác nhận → Hệ thống xóa. **2.Vou.5. Chỉnh sửa:** - Chọn Voucher → Chọn "Chỉnh sửa". - Cập nhật thông tin → Nhấn "Lưu". 3. Hệ thống kiểm tra tính hợp lệ (Ngày bắt đầu >= Hiện tại; Ngày kết thúc > Ngày bắt đầu). 4. Nếu hợp lệ, lưu vào CSDL và thông báo thành công.
Alternate Flow	Không có.
Exception Flow	• **Sai định dạng:** Nếu thông tin nhập vào không đúng định dạng, hệ thống hiển thị đỏ ở ô lỗi và yêu cầu nhập lại. • **Lỗi thời gian:** Nếu ngày bắt đầu trước ngày tạo hoặc ngày kết thúc trước ngày bắt đầu, hệ thống thông báo: *"Thời gian hiệu lực không hợp lệ. Vui lòng nhập lại"* và điều hướng về ô nhập ngày tháng.

3.10.2 Functional Requirements

- **[FR-PROMO-01] Quản lý Voucher:** Tạo mã code (ví dụ SALE50), cài đặt: Loại giảm (Tiền mặt/%), Giá trị giảm, Đơn tối thiểu, Ngày hết hạn, Số lượng giới hạn.
- **[FR-PROMO-02] Áp dụng Voucher:** Tại bước Checkout, hệ thống kiểm tra tính hợp lệ của mã và trừ tiền trực tiếp vào tổng đơn hàng.

4. DATA REQUIREMENTS

Phần này mô tả chi tiết mô hình dữ liệu logic, từ điển dữ liệu (Data Dictionary) và các ràng buộc toàn vẹn mà hệ thống **UTE Phone Hub** phải tuân thủ.

4.1 Logical Data Model

Hệ thống sử dụng cơ sở dữ liệu quan hệ **PostgreSQL 15** làm kho lưu trữ chính. Mô hình dữ liệu được chia thành 5 cụm (domain) chính, liên kết chặt chẽ với nhau:

1. **Users Domain:** Quản lý thông tin định danh (users) và địa chỉ giao hàng (addresses).
2. **Products Domain:** Quản lý danh mục (categories), thương hiệu (brands), và thông tin sản phẩm (products, product_images). Sử dụng kỹ thuật lưu trữ **JSONB** cho các thông số kỹ thuật động.
3. **Orders Domain:** Quản lý vòng đời đơn hàng (orders), chi tiết đơn hàng (order_items), lịch sử trạng thái (order_status_history) và giao dịch thanh toán (payments).
4. **Promotions Domain:** Quản lý các chương trình khuyến mãi và mã giảm giá (promotions).
5. **Interactions Domain:** Quản lý đánh giá (reviews) và dữ liệu chat (conversations, messages).

4.2 Data Dictionary

Dưới đây là đặc tả chi tiết các bảng dữ liệu cốt lõi của hệ thống.

4.2.1 Table: users

Lưu trữ thông tin tài khoản cho cả Khách hàng và Quản trị viên.

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
id	BIGINT	PK, Auto Inc	ID định danh duy nhất.
email	VARCHAR(100)	UNIQUE, NOT NULL	Email đăng nhập/liên hệ.
password_hash	VARCHAR(255)	NOT NULL	Mật khẩu đã mã hóa (BCrypt).
full_name	VARCHAR(100)	NOT NULL	Họ và tên hiển thị.
role	VARCHAR(20)	NOT NULL	Vai trò: ADMIN, MEMBER.
status	VARCHAR(20)	DEFAULT 'ACTIVE'	Trạng thái: ACTIVE, LOCKED, UNVERIFIED.
auth_provider	VARCHAR(20)	DEFAULT 'LOCAL'	LOCAL (Pass) hoặc GOOGLE.

4.2.2 Table: products

Lưu trữ thông tin sản phẩm. Áp dụng chiến lược xóa mềm (Soft Delete).

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
id	BIGINT	PK, Auto Inc	ID sản phẩm.

name	VARCHAR(255)	NOT NULL	Tên sản phẩm.
price	DECIMAL(15,2)	NOT NULL, Check > 0	Giá bán hiện tại.
stock_quantity	INTEGER	DEFAULT 0, Check >= 0	Số lượng tồn kho.
specifications	**JSONB**		Lưu cấu hình chi tiết (RAM, Chip, Màn hình...).
status	BOOLEAN	DEFAULT TRUE	TRUE (Đang bán), FALSE (Ngừng kinh doanh).
is_deleted	BOOLEAN	DEFAULT FALSE	Cờ xóa mềm (Soft Delete).
category_id	BIGINT	FK -> categories	Danh mục sản phẩm.
brand_id	BIGINT	FK -> brands	Thương hiệu.

4.2.3 Table: orders

Lưu trữ thông tin đơn đặt hàng.

Tên trường	**Kiểu dữ liệu**	**Ràng buộc**	**Mô tả**
id	BIGINT	PK, Auto Inc	ID nội bộ.

order_code	VARCHAR(20)	UNIQUE, NOT NULL	Mã đơn hàng (VD: ORD-8888).
user_id	BIGINT	FK -> users	Khách hàng đặt đơn.
total_amount	DECIMAL(15,2)	NOT NULL	Tổng giá trị thanh toán.
status	VARCHAR(20)	NOT NULL	PENDING, CONFIRMED, SHIPPING, COMPLETED, CANCELLED.
payment_method	VARCHAR(20)	NOT NULL	COD, VNPAY.
recipient_info	**JSONB**	NOT NULL	Snapshot thông tin giao hàng (Tên, SĐT, Địa chỉ) tại thời điểm đặt.

4.2.4 Table: payments

Lưu trữ lịch sử giao dịch thanh toán điện tử.

Tên trường	**Kiểu dữ liệu**	**Ràng buộc**	**Mô tả**
id	BIGINT	PK, Auto Inc	ID giao dịch.
order_id	BIGINT	FK -> orders	Liên kết đơn hàng.
provider	VARCHAR(20)		VNPAY, MOMO.

transaction_code	VARCHAR(100)		Mã giao dịch từ cổng thanh toán.
amount	DECIMAL(15,2)	NOT NULL	Số tiền đã thanh toán.
status	VARCHAR(20)	NOT NULL	SUCCESS, FAILED.

4.2.5 Table: promotions

Lưu trữ mã giảm giá (Voucher).

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
id	BIGINT	PK, Auto Inc	ID khuyến mãi.
code	VARCHAR(20)	UNIQUE, NOT NULL	Mã nhập (VD: SALE2024).
discount_type	VARCHAR(20)	NOT NULL	PERCENT, AMOUNT.
discount_value	DECIMAL(15,2)	NOT NULL	Giá trị giảm.
start_date	TIMESTAMP	NOT NULL	Thời gian bắt đầu.
end_date	TIMESTAMP	NOT NULL	Thời gian kết thúc.

4.3 Data Integrity & Constraints (Ràng buộc toàn vẹn)

4.3.1 Ràng buộc tham chiếu (Referential Integrity)

- **Ngăn chặn xóa cha (Restrict Delete):** Hệ thống không cho phép xóa Category hoặc Brand nếu vẫn còn Product tham chiếu đến nó.
- **Xóa mềm (Soft Delete):** Đối với dữ liệu quan trọng như Product, User, Order, hệ thống không thực hiện lệnh DELETE vật lý mà chỉ cập nhật cờ trạng thái (is_deleted

= true hoặc status = DELETED) để đảm bảo tính toàn vẹn của dữ liệu lịch sử và báo cáo thống kê.

4.3.2 Ràng buộc nghiệp vụ (Business Rules)

- **Tồn kho:** stock_quantity không được phép âm. Khi đặt hàng, hệ thống phải sử dụng cơ chế khóa (Locking) hoặc Transaction để đảm bảo không bị bán quá số lượng tồn (Overselling).
- **Giá sản phẩm:** price phải luôn lớn hơn 0.
- **Snapshot dữ liệu:** Thông tin địa chỉ giao hàng và giá sản phẩm trong bảng order_items phải được lưu cứng (Snapshot) tại thời điểm đặt hàng. Việc cập nhật giá sản phẩm gốc hoặc địa chỉ User sau đó không được làm thay đổi lịch sử đơn hàng.

4.4 Caching & Temporary Data (Dữ liệu tạm thời - Redis)

Hệ thống sử dụng **Redis** để lưu trữ các dữ liệu có vòng đời ngắn hạn hoặc truy xuất tần suất cao:

Key Pattern	Loại dữ liệu	TTL (Time-To-Live)	Mô tả
auth:refresh_token:{email}	String	7 ngày	Lưu Refresh Token để quản lý phiên đăng nhập.
auth:otp:{email}	String	5 phút	Lưu mã OTP xác thực/quên mật khẩu.
cart:{session_id}	Hash	30 ngày	Lưu giỏ hàng cho khách vãng lai (Guest).
product:detail:{id}	String (JSON)	1 giờ	Cache thông tin chi tiết sản phẩm để giảm tải DB.

5. EXTERNAL INTERFACE REQUIREMENTS

Phần này mô tả chi tiết các giao diện mà hệ thống cung cấp để tương tác với người dùng, các thành phần phần cứng và các hệ thống phần mềm khác.

5.1 User Interfaces (Giao diện người dùng)

Giao diện người dùng (UI) được thiết kế theo phong cách **Modern Clean E-commerce** (Tối giản, Hiện đại, Tập trung vào sản phẩm), sử dụng **Next.js 15**, **Tailwind CSS 4** và thư viện **Shadcn/UI**.

5.1.1 Nguyên tắc thiết kế chung

- **Responsive Web Design (RWD):** Giao diện phải tự động thích ứng với mọi kích thước màn hình theo nguyên tắc "Mobile-First".
    - **Mobile:** < 640px.
    - **Tablet:** 768px - 1024px.
    - **Desktop:** > 1024px.
- **Tính nhất quán:** Sử dụng bộ màu sắc, font chữ (Inter/Roboto) và các component (Nút bấm, Input, Modal) thống nhất trên toàn bộ website.
- **Feedback:** Hệ thống phải cung cấp phản hồi tức thì cho mọi hành động (Loading spinner khi gọi API, Toast notification khi thành công/thất bại).

5.1.2 Các màn hình chính (Key Screens)

A. Phân hệ Khách hàng (Storefront):

1. **Header & Navigation:** Thanh điều hướng cố định (Sticky), bao gồm Logo, Thanh tìm kiếm thông minh, Giỏ hàng (có badge số lượng), và Menu tài khoản.
2. **Trang chủ (Homepage):** Bố cục Grid giới thiệu Banner khuyến mãi, Sản phẩm mới, Sản phẩm bán chạy.
3. **Trang danh sách (Product Listing):** Sidebar bộ lọc (Filter) bên trái, Lưới sản phẩm bên phải. Hỗ trợ Lazy Loading hoặc Pagination.

4. **Trang chi tiết (Product Detail):** Slide ảnh sản phẩm, chọn biến thể (Màu/ROM), Tab thông số kỹ thuật, Phần đánh giá & Bình luận.
5. **Checkout Flow:** Giao diện tối giản, loại bỏ Header/Footer rườm rà để người dùng tập trung hoàn tất thanh toán.

B. Phân hệ Quản trị (Admin Dashboard):

1. **Layout:** Sidebar menu bên trái (Collapsible), Nội dung chính bên phải.
2. **Dashboard:** Các Widget biểu đồ (Chart.js) hiển thị Doanh thu, Đơn hàng.
3. **Data Tables:** Bảng dữ liệu nâng cao hỗ trợ Sort, Filter, Pagination, và Bulk Actions (Xóa nhiều, Duyệt nhiều).

5.2 Software Interfaces (Giao diện phần mềm)

Hệ thống bao gồm các thành phần phần mềm tương tác với nhau thông qua API và Driver.

5.2.1 5.2.1. Nội bộ hệ thống (Internal)

- **Frontend - Backend:** Giao tiếp hoàn toàn qua **RESTful API** (JSON format) trên giao thức HTTP/HTTPS.
    - Frontend (Next.js Server) gọi Backend API: Server-to-Server communication (cho SSR).
    - Frontend (Browser) gọi Backend API: Client-to-Server communication (cho CSR).
- **Backend - Database:** Sử dụng **Spring Data JPA (Hibernate)** để giao tiếp với PostgreSQL thông qua JDBC Driver.
- **Backend - Redis:** Sử dụng **Spring Data Redis (Lettuce Client)** để giao tiếp với Redis Server.

5.2.2 Tích hợp bên thứ 3 (External Integrations)

1. **Google Identity Platform (OAuth2):**
    - **Giao thức:** OAuth 2.0 / OpenID Connect.
    - **Mục đích:** Xác thực người dùng, lấy thông tin Profile (Email, Name, Avatar).

2. **VNPay Payment Gateway:**
    - **Cơ chế:** Redirect (Chuyển hướng người dùng sang trang thanh toán VNPay) và IPN (Instant Payment Notification - Callback về Server để xác nhận giao dịch).
    - **Môi trường:** Sandbox (Thử nghiệm).
3. **Email Service (SMTP):**
    - **Giao thức:** SMTP (qua Google Mail Server).
    - **Mục đích:** Gửi email xác thực, reset mật khẩu.
4. **Chatbot Widget:**
    - **Cơ chế:** Nhúng mã JavaScript (Script Tag) vào Frontend.

5.3 Hardware Interfaces

Vì là ứng dụng Web, hệ thống không tương tác trực tiếp với phần cứng đặc thù, nhưng yêu cầu cấu hình tối thiểu để vận hành:

5.3.1 Phía Client (Thiết bị người dùng)

- **Thiết bị:** Smartphone, Tablet, Laptop, Desktop PC.
- **Kết nối mạng:** Yêu cầu kết nối Internet (3G/4G/5G/Wifi).

5.3.2 Phía Server (Môi trường triển khai Docker)

- **CPU:** Tối thiểu 2 Core (Khuyến nghị 4 Core).
- **RAM:** Tối thiểu 4GB (Khuyến nghị 8GB để chạy mượt Java + PostgreSQL + Redis).
- **Storage:** SSD tối thiểu 20GB.

5.4 Communications Interfaces (Giao diện truyền thông)

Định nghĩa các chuẩn giao tiếp mạng được sử dụng:

1. **Giao thức mạng:**
    - **HTTP/1.1 & HTTP/2:** Sử dụng cho toàn bộ giao tiếp Web và API.
    - **HTTPS (TLS 1.2/1.3):** Bắt buộc sử dụng mã hóa SSL cho môi trường Production để bảo vệ dữ liệu (đặc biệt là Token và Password).
2. **Định dạng dữ liệu:**

- **JSON (JavaScript Object Notation):** Chuẩn trao đổi dữ liệu chính cho Request và Response Body.

3. **Chuẩn API:**
   - **RESTful:** Tuân thủ các phương thức chuẩn (GET, POST, PUT, DELETE, PATCH).
   - **Status Codes:** Sử dụng đúng mã lỗi HTTP (200, 201, 400, 401, 403, 404, 500).
4. **Bảo mật truyền thông:**
   - **Authorization Header:** Bearer <JWT_ACCESS_TOKEN> được đính kèm trong mọi request cần xác thực.
   - **CORS (Cross-Origin Resource Sharing):** Backend phải cấu hình cho phép Frontend (từ domain khác hoặc port khác - ví dụ localhost:3000) được phép gọi API.

6. QUALITY ATTRIBUTES

Phần này đặc tả các yêu cầu phi chức năng (Non-functional Requirements) quan trọng để đảm bảo chất lượng tổng thể của hệ thống **UTE Phone Hub**.

6.1 Usability

- **[NFR-USABILITY-01] Responsive:** Giao diện website phải hiển thị tốt và không bị vỡ bố cục trên các thiết bị có độ phân giải chiều ngang từ 320px (Mobile) đến 1920px (Desktop).
- **[NFR-USABILITY-02] Ease of Use:** Quy trình đặt hàng (Checkout) cho khách vãng lai (Guest) không được vượt quá **3 bước chính** (Thông tin -> Thanh toán -> Xác nhận).
- **[NFR-USABILITY-03] Feedback:** Hệ thống phải phản hồi mọi tác vụ của người dùng.
    - Trạng thái chờ (Loading) phải xuất hiện trong vòng **0.5 giây** sau khi click.
    - Thông báo lỗi (Error Message) phải rõ ràng, dễ hiểu (Ví dụ: "Mật khẩu sai" thay vì "Error 401").
- **[NFR-USABILITY-04] Consistency:** Màu sắc, font chữ và các icon phải tuân thủ nhất quán theo Design System (Shadcn/UI) trên toàn bộ các trang.

6.2 Performance

- **[NFR-PERF-01] Load Time:** Thời gian tải trang chủ (First Contentful Paint - FCP) phải dưới **1.5 giây** trong điều kiện mạng 4G tiêu chuẩn, nhờ sử dụng công nghệ Server-Side Rendering (Next.js).
- **[NFR-PERF-02] API Latency:** 95% các API đọc dữ liệu (GET) phải phản hồi dưới **200ms**. Các API xử lý nghiệp vụ phức tạp (Đặt hàng, Thanh toán) phải phản hồi dưới **2 giây**.
- **[NFR-PERF-03] Concurrency:** Hệ thống Backend phải hỗ trợ tối thiểu **100 người dùng đồng thời (Concurrent Users)** thao tác mà không gây lỗi Timeout hay Crash hệ thống.

- **[NFR-PERF-04] Caching:** Các dữ liệu ít thay đổi (Danh mục, Cấu hình sản phẩm) phải được cache tại Redis để giảm tải cho Database.

6.3 Security

- **[NFR-SEC-01] Authentication:** Mọi API truy cập tài nguyên riêng tư phải được bảo vệ bằng **JWT (JSON Web Token)**. Token phải có thời hạn (Access Token: 30-60 phút, Refresh Token: 7-30 ngày).
- **[NFR-SEC-02] Password Storage:** Tuyệt đối không lưu mật khẩu dạng rõ (Plain text). Bắt buộc sử dụng thuật toán băm **BCrypt** với độ phức tạp (Work factor) tối thiểu là 10.
- **[NFR-SEC-03] Authorization:** Phân quyền chặt chẽ (RBAC). User thường không thể gọi API của Admin (/api/v1/admin/**). Backend phải check quyền tại tầng Controller/Security Filter, không chỉ ẩn nút ở Frontend.
- **[NFR-SEC-04] Data Protection:** Các dữ liệu nhạy cảm (Token, Session ID) lưu ở Client phải được bảo vệ:
    - Refresh Token lưu trong **HttpOnly Cookie** (ngăn chặn XSS).
    - Sử dụng HTTPS để mã hóa đường truyền.
- **[NFR-SEC-05] Injection Prevention:** Sử dụng Prepared Statement (thông qua JPA/Hibernate) để chống SQL Injection. Validate kỹ đầu vào để chống XSS.

6.4 Reliability & Availability

- **[NFR-REL-01] Data Integrity:** Dữ liệu đơn hàng và thanh toán phải đảm bảo tính toàn vẹn (ACID). Sử dụng Transaction Management của Spring để đảm bảo "Trừ tiền" và "Tạo đơn" thành công cùng lúc hoặc thất bại cùng lúc (Rollback).
- **[NFR-REL-02] Error Handling:** Khi có lỗi hệ thống (500 Internal Server Error), Backend không được trả về Stack Trace chi tiết ra Client mà phải trả về thông báo lỗi chung chung (Generic Error) để tránh lộ thông tin kỹ thuật.
- **[NFR-REL-03] Availability:** Hệ thống được thiết kế để hoạt động 24/7. Thời gian downtime cho phép (để bảo trì/update) tối đa là 4 giờ/tháng.

6.5 Maintainability

- **[NFR-MAINT-01] Code Quality:** Mã nguồn phải tuân thủ Coding Conventions đã thống nhất. Code phải sạch (Clean Code), tách biệt rõ ràng giữa các tầng (Controller, Service, Repository).
- **[NFR-MAINT-02] Documentation:** API phải được mô tả đầy đủ bằng **OpenAPI/Swagger**. Mọi thay đổi về API phải được cập nhật vào tài liệu.
- **[NFR-MAINT-03] Containerization:** Toàn bộ môi trường (Backend, Frontend, DB, Redis) phải được đóng gói trong **Docker Compose** để bất kỳ thành viên mới nào cũng có thể setup dự án trong vòng 15 phút.
- **[NFR-MAINT-04] Modularity:** Hệ thống được thiết kế theo hướng Module hóa. Việc thêm một tính năng mới (ví dụ: Module Blog) không được làm ảnh hưởng đến code của các Module cũ (Order, Auth).

7. APPENDIX A: GLOSSARY

Bảng dưới đây định nghĩa các thuật ngữ kỹ thuật, từ viết tắt và khái niệm nghiệp vụ được sử dụng xuyên suốt trong tài liệu này để đảm bảo sự hiểu biết nhất quán.

Thuật ngữ / Viết tắt	Định nghĩa / Giải thích
API	**Application Programming Interface** (Giao diện Lập trình Ứng dụng). Cơ chế cho phép Frontend và Backend giao tiếp với nhau.
BFF	**Backend For Frontend**. Kiến trúc trong đó Backend được tối ưu hóa để phục vụ dữ liệu cụ thể cho Frontend (Next.js).
COD	**Cash On Delivery** (Thanh toán khi nhận hàng). Một phương thức thanh toán phổ biến tại Việt Nam.
CRUD	**Create, Read, Update, Delete**. Bốn thao tác cơ bản đối với dữ liệu.
DTO	**Data Transfer Object**. Đối tượng dùng để đóng gói dữ liệu truyền giữa Client và Server, giúp ẩn giấu cấu trúc Entity thực tế.
ERD	**Entity Relationship Diagram** (Sơ đồ quan hệ thực thể). Bản vẽ thiết kế cấu trúc Cơ sở dữ liệu.
JWT	**JSON Web Token**. Chuẩn mở (RFC 7519) dùng để truyền tải thông tin xác thực an toàn giữa các bên dưới dạng đối tượng JSON.
MVP	**Minimum Viable Product**. Phiên bản sản phẩm với các tính năng cốt lõi nhất, đủ để triển khai và nhận phản hồi.
OAuth2	Giao thức ủy quyền mở, cho phép ứng dụng truy cập tài nguyên của người dùng trên một dịch vụ khác (như Google) mà không cần biết mật khẩu.

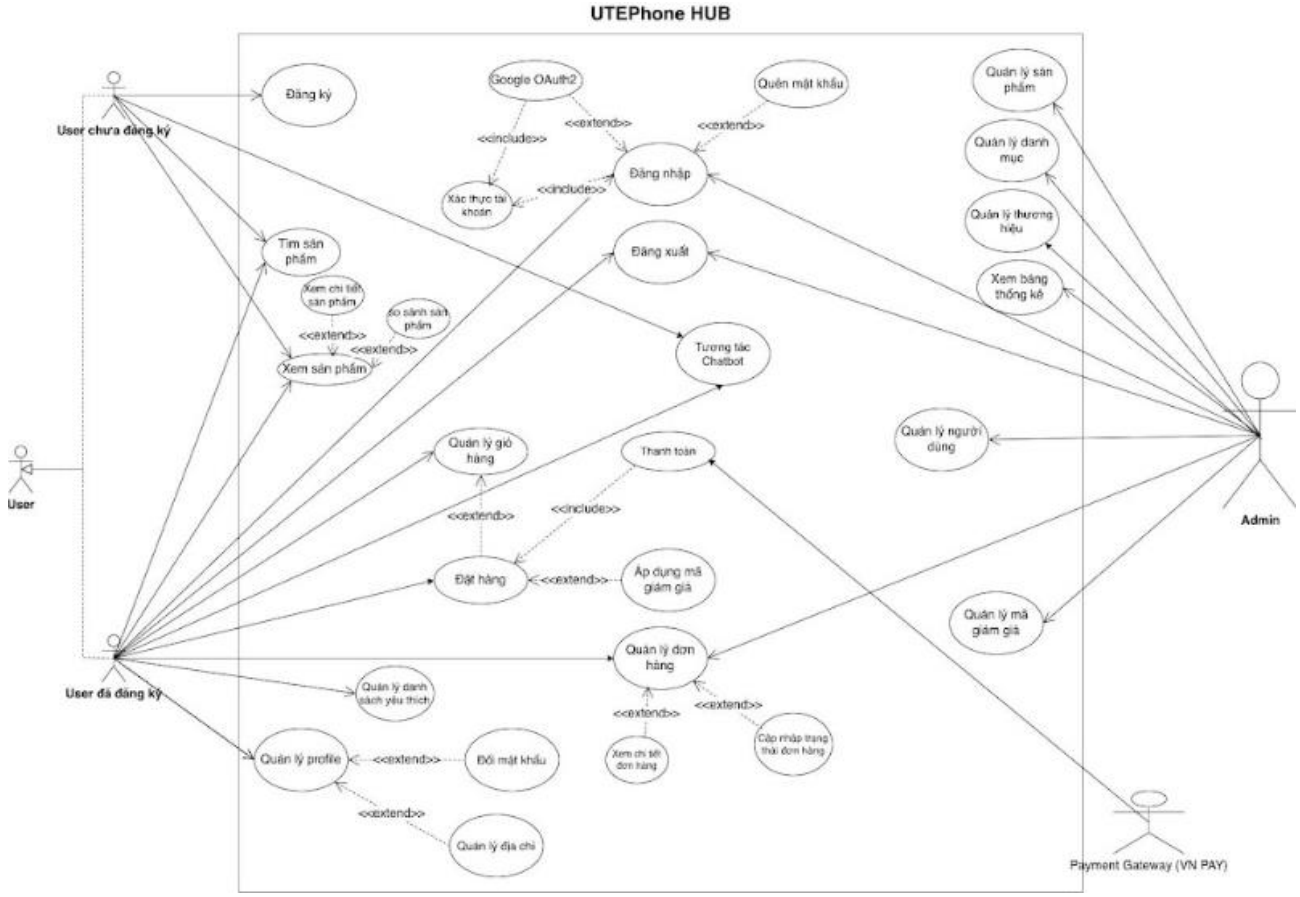
ORM	**Object-Relational Mapping**. Kỹ thuật ánh xạ dữ liệu từ Database quan hệ sang đối tượng trong lập trình hướng đối tượng (Java). Hibernate là một ORM framework.
RBAC	**Role-Based Access Control**. Cơ chế kiểm soát truy cập dựa trên vai trò của người dùng (Admin, Member, Guest).
Redis	Hệ quản trị cơ sở dữ liệu in-memory (lưu trong RAM), thường được dùng làm bộ nhớ đệm (Cache) hoặc lưu Session.
Soft Delete	**Xóa mềm**. Kỹ thuật đánh dấu dữ liệu là "đã xóa" (ví dụ: is_deleted = true) thay vì xóa vĩnh viễn khỏi Database, giúp khôi phục dữ liệu khi cần.
SSR	**Server-Side Rendering**. Kỹ thuật render trang web từ phía Server (Next.js) rồi gửi HTML về trình duyệt, giúp tăng tốc độ tải trang ban đầu và tốt cho SEO.
SKU	**Stock Keeping Unit**. Mã định danh duy nhất cho mỗi sản phẩm trong kho.

8. APPENDIX B: ANALYSIS MODELS

Phần này liệt kê danh sách các biểu đồ phân tích hệ thống (Analysis Models) đã được thực hiện và lưu trữ trong tài liệu thiết kế chi tiết.

8.1 Use Case Model (Mô hình Ca sử dụng)

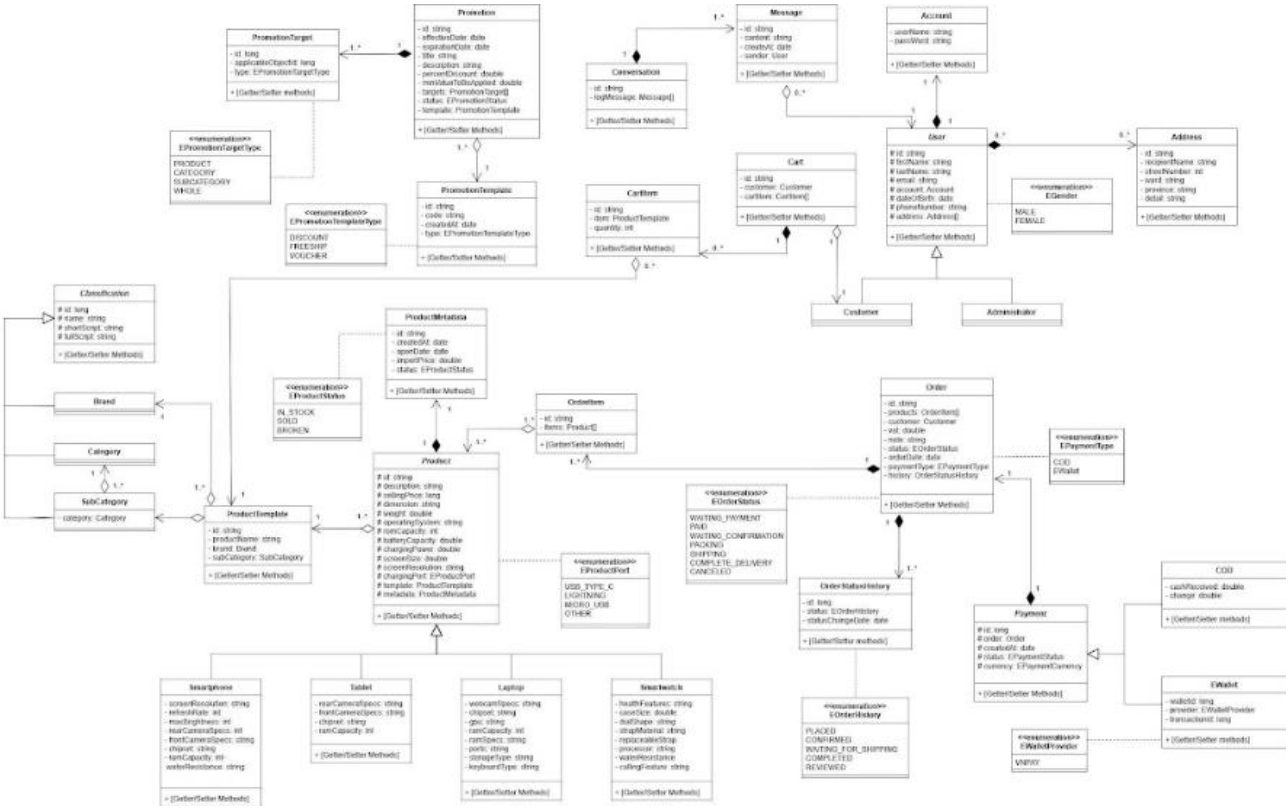
- **Mô tả:** Biểu diễn các tác nhân (Actors) và các chức năng (Use Cases) của hệ thống.
- **Danh sách Use Case chính:**
    - UC-AUTH: Đăng ký, Đăng nhập, Quên mật khẩu.
    - UC-PROD: Tìm kiếm, Xem chi tiết, So sánh.
    - UC-CART: Quản lý giỏ hàng.
    - UC-ORDER: Đặt hàng, Thanh toán, Theo dõi đơn.
    - UC-ADMIN: Quản lý Sản phẩm, Đơn hàng, Người dùng.



Hình: Usecase Diagram

8.2 Domain Model (Mô hình Nghiệp vụ - Class Diagram)

- **Mô tả:** Biểu diễn các lớp thực thể (Entities), thuộc tính và mối quan hệ giữa chúng trong hệ thống.
- **Thực thể chính:** User, Product, Order, Category, Cart, Review, Promotion.



Hình: Class Diagram