|  |  |  |
| --- | --- | --- |
| **Họ và Tên** | **MSSV** | **Chức vụ** |
| Nguyễn Minh Kiên | 3122411103 | Nhóm trưởng |
| Lê Nhựt Huy | 3122411063 | Thành viên |
| Phạm Nhật Phương | 3122411162 | Thành viên |
| Phạm Nguyễn Thế Hào | 3122411048 | Thành viên |

**BÁO CÁO**

**Lớp: DCT122C3**

**Năm học: 2025 – 2026**

**TP. HỒ CHÍ MINH, THÁNG 10 NĂM 2025**

ỦY BAN NHÂN DÂN TP. HỒ CHÍ MINH

**TRƯỜNG ĐẠI HỌC SÀI GÒN**

**MỤC LỤC**

[Chương 1: Phân tích, thiết kế hệ thống 4](#_Toc212069403)

[1.1. Ngữ cảnh nghiệp vụ 5](#_Toc212069404)

[1.2. Yêu cầu về chức năng và phi chức năng 7](#_Toc212069405)

[1.2.1. Chức năng chính 7](#_Toc212069406)

[1.2.2. Phi chức năng 7](#_Toc212069407)

[1.3. Số lượng người dùng ở mức nghiệp vụ và mức hệ thống 7](#_Toc212069408)

[1.3.1. Mức nghiệp vụ (Business Level Actors) 7](#_Toc212069409)

[1.3.2. Mức hệ thống (System Level Actors) 7](#_Toc212069410)

[1.4. Kế hoạch triển khai dự án 8](#_Toc212069411)

[1.5. Đối tượng và phạm vi kiểm thử 8](#_Toc212069412)

[1.6. Use Case 8](#_Toc212069413)

[1.6.1 Use Case: Danh mục sản phẩm 8](#_Toc212069414)

[1.6.2. Use Case: Giỏ hàng 10](#_Toc212069415)

[1.6.3. Use Case Quy trình thanh toán 11](#_Toc212069416)

[1.6.4. Use Case: Kho hàng 12](#_Toc212069417)

[1.6.5. Use Case: Đánh giá 13](#_Toc212069418)

[1.6.6. Use Case: Đơn hàng 14](#_Toc212069419)

[1.6.7. Use Case: Kiểm soát truy cập 15](#_Toc212069420)

[1.7. User Story 16](#_Toc212069421)

[1.7.1. Danh mục sản phẩm (Product Catalog) 16](#_Toc212069422)

[1.7.2. Giỏ hàng (Shopping Cart) 16](#_Toc212069423)

[1.7.3. Quy trình thanh toán (Payment Process) 17](#_Toc212069424)

[1.7.4. Đơn hàng (Order) 17](#_Toc212069425)

[1.7.5. Quản lý kho (Inventory) 18](#_Toc212069426)

[1.7.6. Đánh giá sản phẩm (Rating) 18](#_Toc212069427)

[1.7.7. Kiểm soát truy cập (Access Control) 18](#_Toc212069428)

[1.8. State Diagram cho đơn hàng 19](#_Toc212069429)

[1.9. Database 20](#_Toc212069430)

[1.10. ERD 21](#_Toc212069431)

[1.11. Sơ đồ khối 22](#_Toc212069432)

[1.11.1. Giới thiệu về sơ đồ khối 22](#_Toc212069433)

[1.11.2 Các thành phần chính 22](#_Toc212069434)

[1.11.3. Luồng hoạt động 24](#_Toc212069435)

[1.11.4. Ý nghĩa và mối liên hệ với sơ đồ khái niệm 25](#_Toc212069436)

[1.12. Decomposition View: C4 26](#_Toc212069437)

[1.12.1. sơ đồ C1 - System Context 26](#_Toc212069438)

[1.12.2. sơ đồ C2 – Container 28](#_Toc212069439)

[1.12.3. Sơ đồ C3 - Component (Module Level) 30](#_Toc212069440)

[1.12.4. sơ đồ C3 – Component (High-Level) 30](#_Toc212069441)

[1.13.Communication View 54](#_Toc212069442)

[1.14. Deployment View 57](#_Toc212069443)

[1.15. Sơ đồ kiến trúc SPA (Single Page Application): 61](#_Toc212069444)

[1.16. Sơ đồ triển khai CI/CD: 61](#_Toc212069445)

[1.17. Sơ đồ API của hệ thống: 62](#_Toc212069446)

[1.18. sơ đồ xử lý 1 request: 63](#_Toc212069447)

[1.19. Mô hình khái niệm 64](#_Toc212069448)

[1.20. Business Process Diagram 65](#_Toc212069449)

[1.21. Class Diagram 66](#_Toc212069450)

[1.22. Squence 67](#_Toc212069451)

[1.22.1. Danh mục sản phẩm 67](#_Toc212069452)

[1.22.2. Giỏ hàng 67](#_Toc212069453)

[1.22.3. Quy trình thanh toán 67](#_Toc212069454)

[1.22.4. Kho hàng 68](#_Toc212069455)

[1.22.5. Đánh giá 69](#_Toc212069456)

[1.22.6. Đơn hàng 69](#_Toc212069457)

[1.22.7. Kiểm soát truy cập 69](#_Toc212069458)

[Chương 2. Kế hoạch kiểm thử 71](#_Toc212069459)

[2.1. Giới thiệu 71](#_Toc212069460)

[2.1.1 Mục đích 71](#_Toc212069461)

[2.1.2 Bối cảnh 71](#_Toc212069462)

[2.1.3 Phạm vi 72](#_Toc212069463)

[2.1.4 Thông tin nhận diện dự án 72](#_Toc212069464)

[2.1.5 References 75](#_Toc212069465)

[2.2 Các tính năng cần kiểm thử 76](#_Toc212069466)

[2.2.1 Chức năng 76](#_Toc212069467)

[2.2.2 Khả năng sử dụng 96](#_Toc212069468)

[2.2.3 Ràng buộc thiết kế 98](#_Toc212069469)

[2.2.4 Giao diện 99](#_Toc212069470)

[2.2.5. Các tính năng không kiểm thử 102](#_Toc212069471)

[2.2.6. Chiến lược kiểm thử 104](#_Toc212069472)

[2.2.7 Ưu tiên kiểm thử 108](#_Toc212069473)

[2.2.8 Tiêu chí chấp nhận 111](#_Toc212069474)

[2.3 Phương pháp và loại kiểm thử 112](#_Toc212069475)

[2.4 Môi trường & công cụ kiểm thử 114](#_Toc212069476)

[2.8 Quản lý rủi ro 116](#_Toc212069477)

[2.9 Báo cáo kết quả 117](#_Toc212069478)

# Chương 1: Phân tích, thiết kế hệ thống

Website bán laptop là một nền tảng thương mại điện tử chuyên cung cấp laptop, được thiết kế để đáp ứng yêu cầu học thuật. Hệ thống sử dụng kiến trúc microservices từ CoolStore, xây dựng trên .NET Core, NodeJS, và chạy trên Service Mesh (Istio) với Kubernetes. Website tập trung vào các tính năng cơ bản: danh mục sản phẩm (chỉ laptop), giỏ hàng, thanh toán, quản lý kho, kiểm soát truy cập, và đánh giá sản phẩm. Danh mục sản phẩm cố định (chỉ laptop với thông tin chuẩn hóa như hãng, CPU, RAM, storage) giúp giảm thiểu công việc định dạng, trong khi tính năng đánh giá cho phép người mua đánh giá từng laptop (thang điểm 1-5 sao). Mục tiêu là minh họa các nguyên tắc microservices trong một ứng dụng đơn giản nhưng đầy đủ chức năng.

## 1.1. Ngữ cảnh nghiệp vụ

Website Bán Máy Tính có các kịch bản nghiệp vụ cơ bản cho Danh mục sản phẩm, Giỏ hàng, Thanh toán, Kho hàng, Đánh giá và Kiểm soát truy cập.

Với **Danh mục sản phẩm:** Khách hàng có thể duyệt danh sách các loại máy tính và linh kiện với tính năng lọc và sắp xếp theo tên, giá hoặc cấu hình. Họ có thể xem chi tiết sản phẩm bằng cách nhấn vào sản phẩm đó trong danh sách. Trong trang chi tiết, họ sẽ thấy tên sản phẩm, mô tả, thông số kỹ thuật, tình trạng tồn kho, thông tin kho hàng (địa chỉ, vị trí), nhãn “hot” (nếu là sản phẩm được nhiều người mua), và đánh giá của Khách hàng khác. Quản trị viên (Admin) có thể quản lý sản phẩm (CRUD) và gán sản phẩm vào kho có sẵn.

Với **Giỏ hàng:** Khách hàng có thể thêm sản phẩm vào giỏ hàng từ trang danh sách hoặc trang chi tiết sản phẩm. Sau khi thêm, sản phẩm sẽ được hiển thị trong giỏ hàng cùng bảng tóm tắt thông tin như tổng chi phí giỏ hàng, số tiền tiết kiệm từ khuyến mãi, tổng phụ (subtotal), phí vận chuyển, số tiền tiết kiệm phí vận chuyển, và tổng đơn hàng. Khi Khách hàng thêm, sửa số lượng, hoặc xóa sản phẩm, giỏ hàng và bảng tóm tắt sẽ tự động cập nhật. Sau cùng, Khách hàng có thể nhấn nút “Thanh toán”. Quản trị viên có thể xem giỏ hàng của tất cả người dùng và bật/tắt các giỏ hàng không hợp lệ.

Với **Thanh toán:** Sau khi Khách hàng nhấn “Thanh toán”, hệ thống sẽ kiểm tra lại thông tin sản phẩm, xử lý thanh toán, và gửi email thông báo xác nhận đơn hàng cho Khách hàng.

Với **Kho hàng:** Quản trị viên có thể quản lý kho máy tính và linh kiện (CRUD).

Với **Đánh giá:** Khách hàng có thể đánh giá các sản phẩm họ đã mua (từ 1 → 5 sao).

Với **Kiểm soát truy cập:** Khách hàng và Quản trị viên có thể đăng nhập/đăng xuất hệ thống. Nếu là Khách hàng, họ sẽ được đưa đến trang danh mục sản phẩm; nếu là quản trị viên, họ sẽ được đưa đến trang quản trị.

Với **Cấu hình ban đầu:** Khi hệ thống Website Bán Máy Tính được khởi động lần đầu, cần tạo sẵn một tài khoản quản trị viên, vài tài khoản Khách hàng, và dữ liệu mẫu cho sản phẩm, kho hàng, và đánh giá.

## 1.2. Yêu cầu về chức năng và phi chức năng

### 1.2.1. Chức năng chính

1. Đăng nhập/Đăng ký/Đăng xuất
2. Quản lý sản phẩm (CRUD)
3. Giỏ hàng (Thêm/Xóa/Cập nhật)
4. Thanh toán (COD/PayPal)
5. Đánh giá sản phẩm
6. Quản lý người dùng (Admin)
7. Xem đơn hàng cá nhân
8. Tìm kiếm, lọc sản phẩm

### 1.2.2. Phi chức năng

1. Dữ liệu được bảo mật (JWT, mã hóa mật khẩu).
2. Phản hồi nhanh (<2s).
3. Giao diện thân thiện, responsive.
4. Kiến trúc microservice tách biệt dễ mở rộng.

## 1.3. Số lượng người dùng ở mức nghiệp vụ và mức hệ thống

### 1.3.1. Mức nghiệp vụ (Business Level Actors)

|  |  |  |
| --- | --- | --- |
| **STT** | **Tác nhân** | **Vai trò** |
| 1 | Khách hàng | Tìm kiếm sản phẩm, thêm vào giỏ hàng, đặt hàng, đánh giá sản phẩm |
| 2 | Quản trị viên | Quản lý sản phẩm, đơn hàng, tài khoản, kiểm duyệt đánh giá, thống kê doanh thu |

### 1.3.2. Mức hệ thống (System Level Actors)

|  |  |  |
| --- | --- | --- |
| **STT** | **Tác nhân** | **Vai trò** |
| 1 | Hệ thống thanh toán online | Xử lý giao dịch thanh toán online |
| 2 | |  | | --- | | Cơ sở dữ liệu MongoDB |  |  | | --- | |  | | |  | | --- | | Lưu trữ dữ liệu sản phẩm, người dùng, đơn hàng, đánh giá |  |  | | --- | |  | |
| 3 | |  | | --- | | FrontEnd (ReactJS) |  |  | | --- | |  | | |  | | --- | | Giao diện người dùng, tương tác qua API |  |  | | --- | |  | |
| 4 | |  | | --- | | BackEnd (NodeJS/Express) |  |  | | --- | |  | | Xử lý logic nghiệp vụ, trung gian giữa người dùng và dữ liệu |

## 1.4. Kế hoạch triển khai dự án

|  |  |
| --- | --- |
| **Giai đoạn** | **Công việc chính** |
| 1 | Phân tích yêu cầu, thiết kế ERD |
| 2 | Xây dựng Backend (API + Models) |
| 3 | Xây dựng Frontend (ReactJS) |
| 4 | Tích hợp FE + BE |
| 5 | Kiểm thử và viết báo cáo |

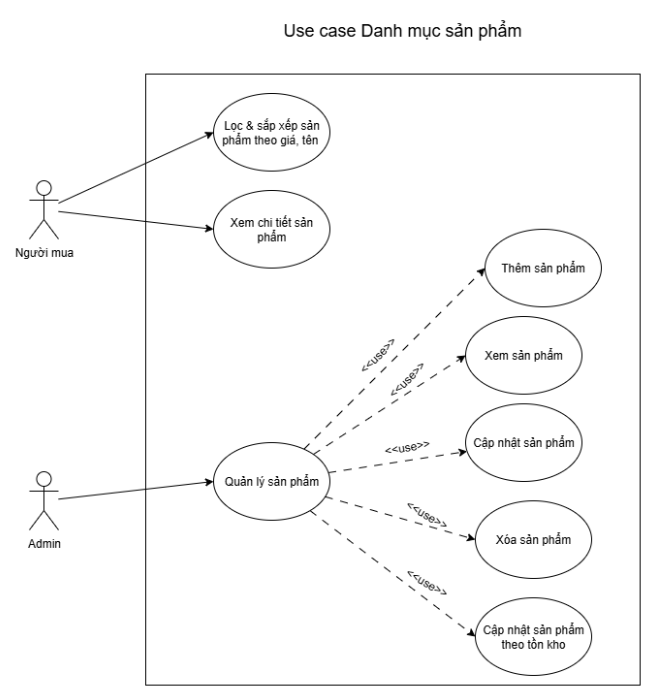
## 1.5. Đối tượng và phạm vi kiểm thử

1. Đối tượng: Các chức năng chính (Auth, Product, Cart, Order, Payment, Review)
2. Phạm vi:

* Không kiểm thử hiệu năng (non-functional test).
* Không kiểm thử bảo mật ở cấp độ mạng.
* Tập trung Black-box và White-box test.

## 1.6. Use Case

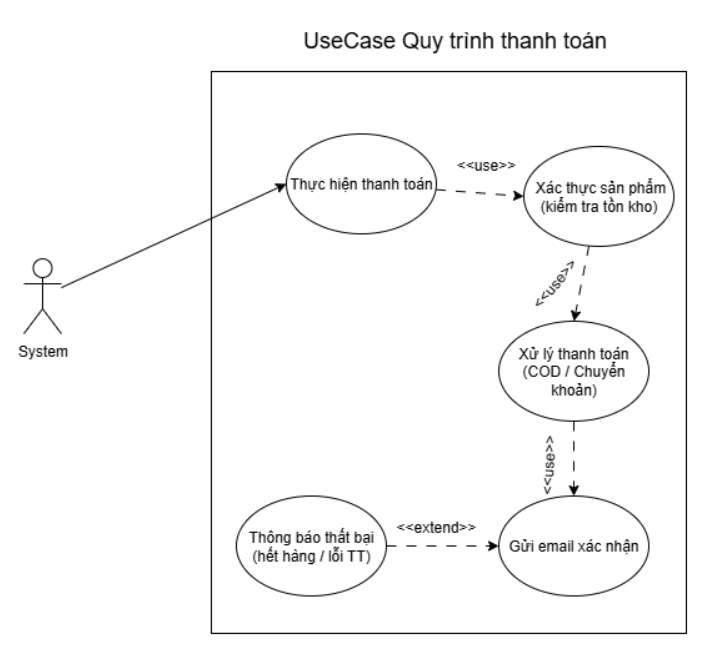
### 1.6.1 Use Case: Danh mục sản phẩm



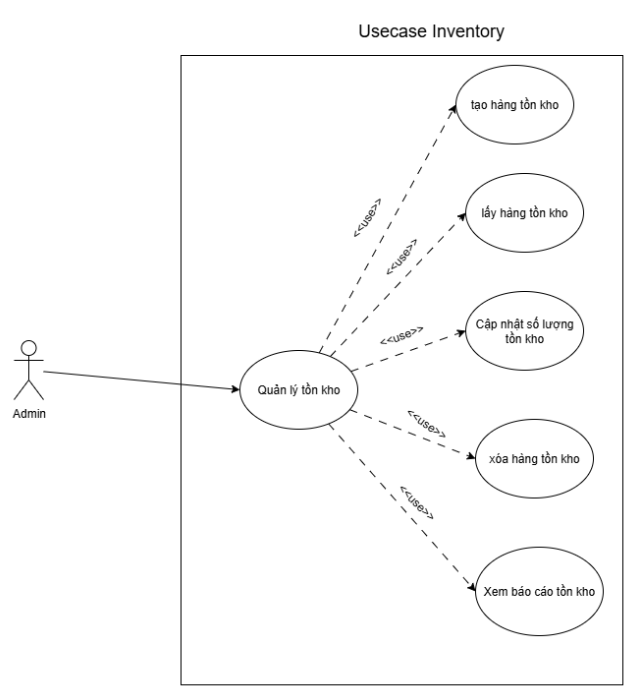
### 1.6.2. Use Case: Giỏ hàng



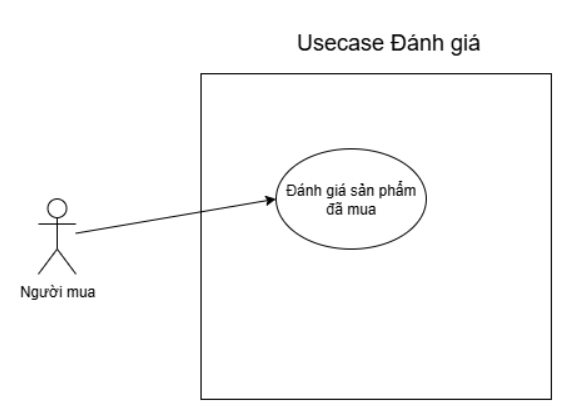
### 1.6.3. Use Case Quy trình thanh toán



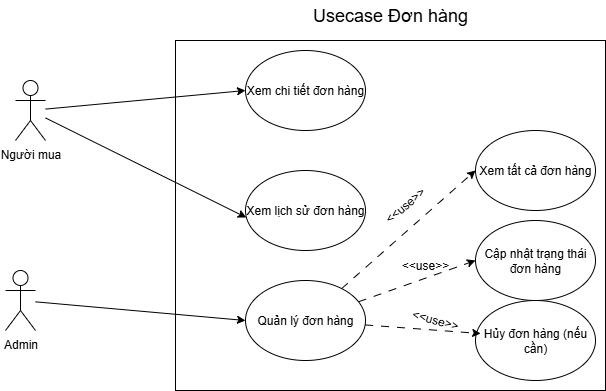
### 1.6.4. Use Case: Kho hàng



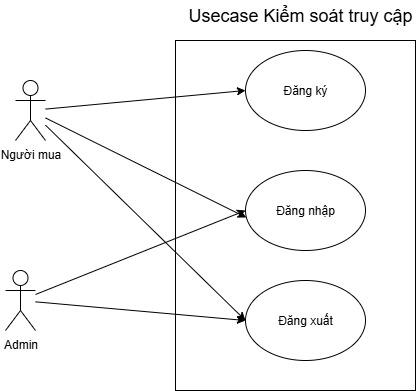
### 1.6.5. Use Case: Đánh giá



### 1.6.6. Use Case: Đơn hàng



### 1.6.7. Use Case: Kiểm soát truy cập



## 1.7. User Story

### 1.7.1. Danh mục sản phẩm (Product Catalog)

* Là một Người mua, tôi muốn xem danh sách sản phẩm với chức năng lọc và sắp xếp trên trang chủ (theo tên, ảnh, mô tả ngắn, đánh giá, và cờ “sản phẩm nổi bật”) để dễ dàng tìm được laptop tôi cần.
* Khi tôi lọc theo giá hoặc tên, danh sách sản phẩm phải thu hẹp lại đúng theo điều kiện.
* Khi tôi sắp xếp tăng dần/giảm dần theo giá hoặc tên, danh sách sản phẩm phải thay đổi đúng thứ tự.
* Khi tôi vừa lọc vừa sắp xếp, danh sách sản phẩm phải áp dụng đồng thời cả hai điều kiện.
* Là một Người mua, tôi muốn xem chi tiết sản phẩm gồm: tên, mô tả, số lượng tồn kho, thông tin kho (địa chỉ/lưu trữ), cờ sản phẩm nổi bật, và đánh giá để có đủ thông tin trước khi mua.
* Là một Quản trị viên, tôi muốn quản lý sản phẩm (thêm, xem, sửa, xóa) và gán sản phẩm đó vào một kho tồn tại để đảm bảo dữ liệu chính xác.

### 1.7.2. Giỏ hàng (Shopping Cart)

* Là một Người mua, tôi muốn thêm sản phẩm từ trang danh mục vào giỏ hàng (mặc định số lượng = 1) để mua.
* Là một Người mua, tôi muốn thêm sản phẩm từ trang chi tiết vào giỏ hàng để mua ngay nếu tôi thích.
* Là một Người mua, tôi muốn xem danh sách sản phẩm đã thêm trong giỏ và có một bảng tóm tắt hiển thị: tổng chi phí giỏ hàng, chi phí vận chuyển, khuyến mãi (nếu có), tiền tiết kiệm từ khuyến mãi, và tổng giá trị đơn hàng.
* Là một Người mua, tôi muốn cập nhật số lượng sản phẩm trong giỏ.
* Khi số lượng thay đổi, bảng tóm tắt phải tự động cập nhật.
* Là một Người mua, tôi muốn xóa sản phẩm khỏi giỏ hàng.
* Khi sản phẩm bị xóa, bảng tóm tắt cũng phải tự động cập nhật.
* Là một Người mua, tôi muốn tiến hành thanh toán giỏ hàng.
* Nếu giỏ hàng trống (không có sản phẩm), quá trình thanh toán không được phép xảy ra.
* Khi giỏ hàng được thanh toán, hệ thống sẽ chuyển sang quy trình thanh toán.
* Là một Quản trị viên, tôi muốn xem giỏ hàng của tất cả người mua với thông tin chi tiết (tổng chi phí, khuyến mãi, chi phí vận chuyển, tổng giá trị đơn hàng) để quản lý.
* Là một Quản trị viên, tôi muốn bật/tắt (enable/disable) giỏ hàng của bất kỳ người mua nào để xử lý trường hợp không hợp lệ.

### 1.7.3. Quy trình thanh toán (Payment Process)

* Là một Người mua, tôi có thể thanh toán cho giỏ hàng của mình.
* Khi bắt đầu thanh toán, hệ thống sẽ:
  + Kiểm tra thông tin sản phẩm (tồn kho).
  + Xử lý thanh toán (COD hoặc chuyển khoản, mô phỏng).
  + Gửi email xác nhận cho người mua.
* Nếu sản phẩm không hợp lệ (hết hàng hoặc lỗi), quy trình thanh toán bị hủy và email thông báo sẽ được gửi đến người mua.
* Khi quy trình thanh toán kết thúc, đơn hàng sẽ được đánh dấu trạng thái đã xử lý, và email xác nhận được gửi cho người mua.

### 1.7.4. Đơn hàng (Order)

* Là một **Người mua**, tôi muốn xem danh sách **lịch sử đơn hàng** của mình để theo dõi các giao dịch đã thực hiện.
* Là một **Người mua**, tôi muốn xem **chi tiết đơn hàng** (mã đơn, sản phẩm, số lượng, tổng tiền, trạng thái, ngày mua) để biết chính xác thông tin đặt hàng.
* Là một **Người mua**, tôi muốn biết **trạng thái đơn hàng** (đang xử lý, đã thanh toán, đang giao, hoàn tất, hủy) để theo dõi quá trình.
* Là một **Quản trị viên**, tôi muốn **xem danh sách tất cả đơn hàng** để quản lý tình hình kinh doanh.
* Là một **Quản trị viên**, tôi muốn **cập nhật trạng thái đơn hàng** (ví dụ: từ “đang xử lý” sang “đang giao hàng” hoặc “hoàn tất”) để đồng bộ với thực tế.
* Là một **Quản trị viên**, tôi muốn có thể **hủy đơn hàng** nếu phát hiện gian lận hoặc đơn hàng không hợp lệ.

### 1.7.5. Quản lý kho (Inventory)

* Là một Quản trị viên, tôi muốn quản lý kho hàng (thêm, xem, sửa, xóa) để đảm bảo số lượng tồn kho luôn chính xác.

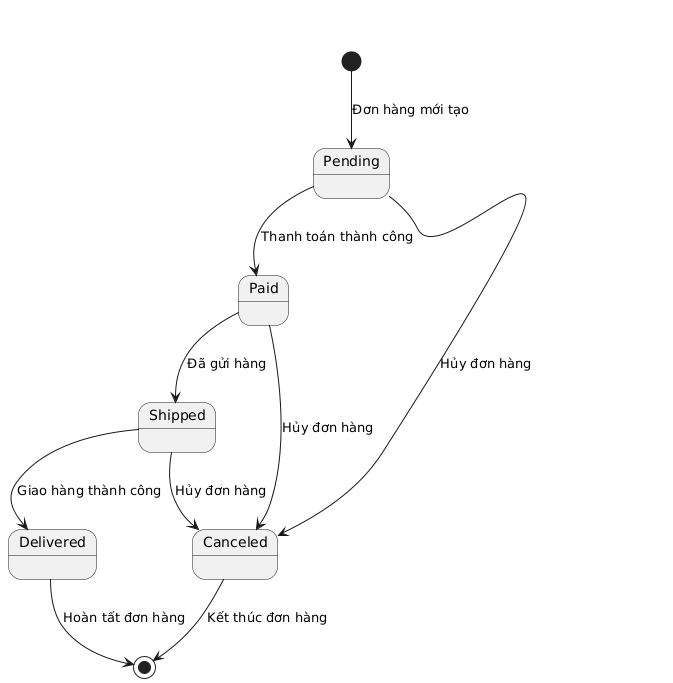
### 1.7.6. Đánh giá sản phẩm (Rating)

* Là một Người mua, tôi muốn đánh giá sản phẩm đã mua bằng 1 → 5 sao, kèm bình luận tùy chọn để chia sẻ trải nghiệm.
* Là một Người mua, tôi muốn xem điểm trung bình và các bình luận của người mua khác để tham khảo trước khi quyết định mua.

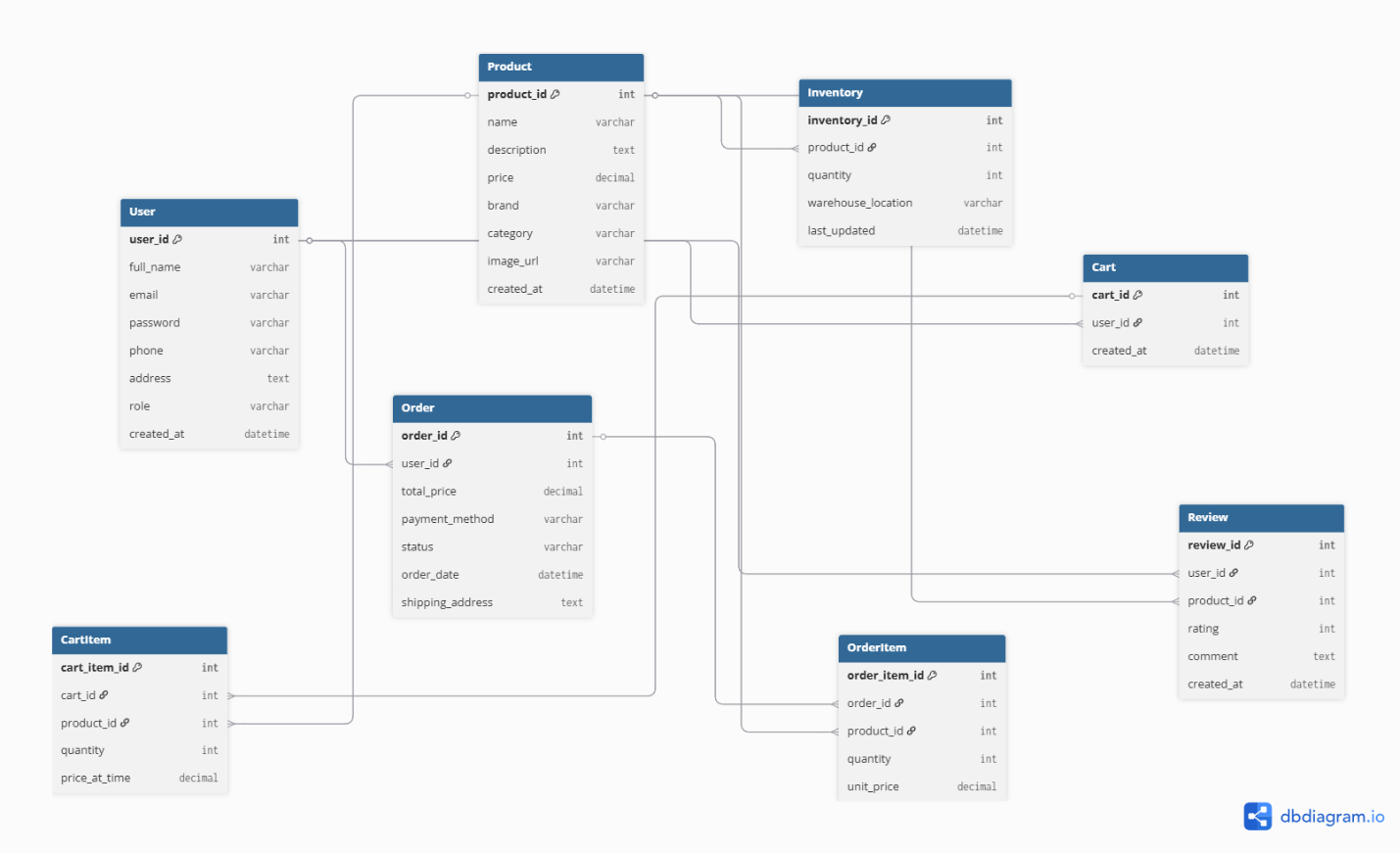
### 1.7.7. Kiểm soát truy cập (Access Control)

* Mỗi Người mua hoặc Quản trị viên đều là Người dùng.
* Là một Người mua/Quản trị viên, tôi muốn đăng nhập vào hệ thống.
* Khi Người mua đăng nhập, hệ thống chuyển hướng đến trang Danh mục sản phẩm.
* Khi Quản trị viên đăng nhập, hệ thống chuyển hướng đến trang Quản trị.
* Là một Người mua/Quản trị viên, tôi muốn đăng xuất để bảo mật tài khoản.

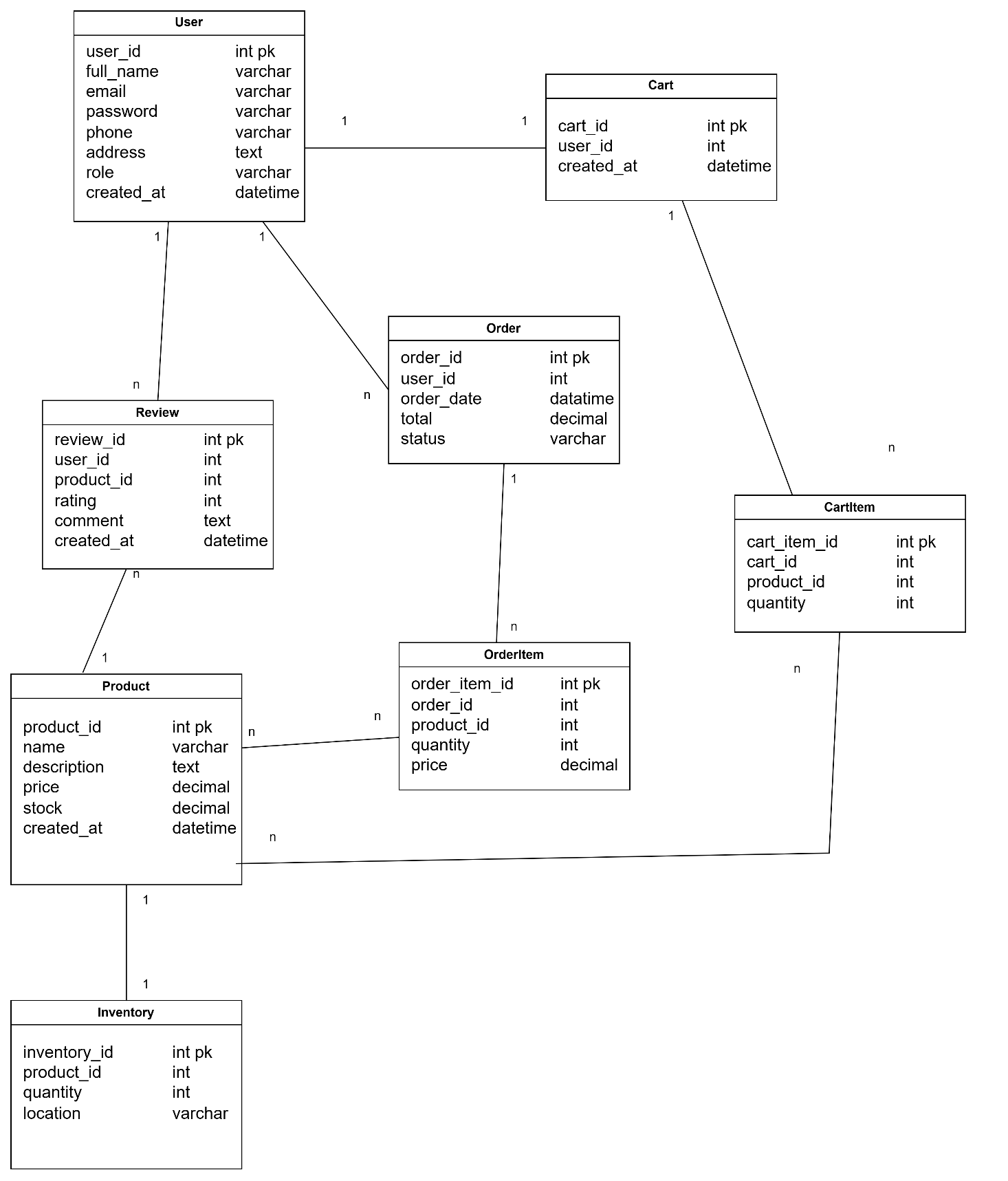
## 1.8. State Diagram cho đơn hàng



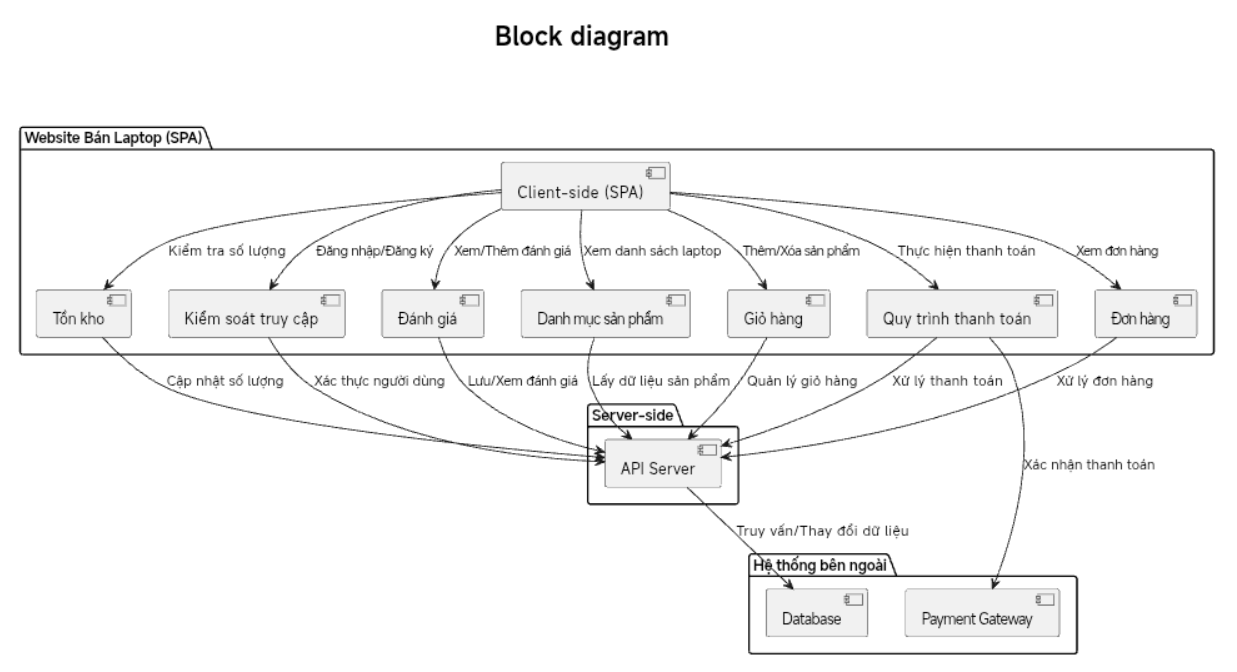
## 1.9. Database



## 1.10. ERD



## 1.11. Sơ đồ khối



### 1.11.1. Giới thiệu về sơ đồ khối

Sơ đồ khối mô tả kiến trúc tổng thể của hệ thống Website bán laptop (SPA – Single Page Application). Sơ đồ này ở mức khái quát hóa kiến trúc, cho thấy các thành phần chính của hệ thống, cách chúng được tổ chức và cách chúng liên kết với nhau.

Khác với sơ đồ khái niệm (conceptual diagram) chỉ mô tả các đối tượng nghiệp vụ và mối quan hệ giữa chúng, sơ đồ khối đi thêm một bước, thể hiện cách các khối chức năng vận hành trên từng lớp: Client-side, Server-side, và hệ thống bên ngoài. Giúp người đọc hiểu rõ không chỉ có cái gì tồn tại trong hệ thống, mà còn chỉ ra các khối đó giao tiếp với nhau như thế nào.

### 1.11.2 Các thành phần chính

a. Client-side (SPA)

Là lớp giao diện người dùng, chạy trên trình duyệt. SPA (Single Page Application), toàn bộ trải nghiệm của người dùng được xử lý trong một trang web duy nhất, không phải tải lại toàn bộ trang khi chuyển đổi chức năng. Các module chính gồm:

Danh mục sản phẩm: cung cấp cho người dùng khả năng duyệt qua toàn bộ danh sách laptop có trong hệ thống. Dữ liệu này không tự chứa trên client mà sẽ được lấy từ server qua API.

* Giỏ hàng: nơi người dùng thêm hoặc xóa sản phẩm. Thông tin giỏ hàng có thể tạm thời lưu trên client, nhưng để đồng bộ và duy trì trạng thái giữa nhiều thiết bị, nó cần gửi về server.
* Tồn kho: hiển thị số lượng sản phẩm hiện có. Đây là tính năng quan trọng để đảm bảo khách không đặt mua vượt quá số lượng sẵn có
* Kiểm soát truy cập: hỗ trợ người dùng đăng nhập, đăng ký, hoặc quản lý tài khoản. Cơ chế này thường đi kèm với session hoặc token (JWT).
* Đánh giá: cho phép người dùng xem nhận xét từ khách hàng khác và thêm nhận xét mới cho sản phẩm đã mua.
* Quy trình thanh toán: tập trung các bước thanh toán, từ nhập thông tin giao hàng, chọn phương thức thanh toán, đến xác nhận đơn hàng.
* Đơn hàng: nơi người dùng có thể theo dõi trạng thái đơn hàng đã đặt (ví dụ: đang xử lý, đã giao hàng, đã hủy).

b. Server-side (API Server)

Là tầng trung gian, giữ vai trò xử lý logic nghiệp vụ. API Server tiếp nhận các yêu cầu từ Client-side, phân tích và xử lý, sau đó giao tiếp với hệ thống bên ngoài. Các nhiệm vụ chính:

1. Xử lý yêu cầu dữ liệu: ví dụ, khi Client cần danh sách sản phẩm, API Server sẽ truy vấn Database và trả về kết quả.
2. Đảm bảo tính toàn vẹn dữ liệu: API Server đảm nhận việc kiểm soát dữ liệu (giỏ hàng, tồn kho, đơn hàng, đánh giá) để không xảy ra sai lệch.
3. Xác thực và phân quyền: khi Client gửi yêu cầu có liên quan đến tài khoản người dùng, API Server sẽ xác minh token hoặc session để đảm bảo tính hợp lệ.
4. Tích hợp các dịch vụ bên ngoài: API Server chính là cầu nối để Client có thể gián tiếp tương tác với Database và Payment Gateway.
5. Nói cách khác, Client-side không bao giờ truy cập trực tiếp vào Database hay Payment Gateway, mà tất cả đều phải đi qua API Server.

c. Hệ thống bên ngoài

* Database: lưu trữ toàn bộ thông tin cốt lõi của hệ thống, bao gồm người dùng, sản phẩm, tồn kho, giỏ hàng, đơn hàng, đánh giá. Đây là nền tảng dữ liệu để mọi module có thể hoạt động ổn định.
* Payment Gateway: cổng thanh toán trung gian, chịu trách nhiệm xác thực và xử lý các giao dịch tài chính trực tuyến. Hệ thống website không tự thực hiện giao dịch, mà chỉ gửi yêu cầu đến Payment Gateway để đảm bảo an toàn và tuân thủ pháp lý.

### 1.11.3. Luồng hoạt động

1. Người dùng thao tác trên Client-side (SPA), ví dụ: duyệt sản phẩm, thêm laptop vào giỏ, hoặc nhấn thanh toán.
2. Ứng dụng SPA gửi yêu cầu (HTTP request, thường ở dạng JSON qua REST API hoặc GraphQL) đến API Server.
3. API Server xử lý logic nghiệp vụ:

* Nếu yêu cầu là đọc dữ liệu (xem sản phẩm, tồn kho, đơn hàng), nó truy vấn Database và trả kết quả về.
* Nếu yêu cầu là ghi dữ liệu (thêm giỏ hàng, tạo đơn hàng, gửi đánh giá), nó cập nhật Database và phản hồi trạng thái cho Client.

1. Với thao tác thanh toán, Quy trình thanh toán từ Client sẽ gọi đến Payment Gateway qua API Server. Payment Gateway xác nhận giao dịch, sau đó kết quả được trả lại API Server.
2. API Server ghi nhận kết quả thanh toán và cập nhật lại Database để đảm bảo dữ liệu đơn hàng chính xác.
3. Cuối cùng, Client-side (SPA) nhận phản hồi và hiển thị kết quả cho người dùng (ví dụ: “Thanh toán thành công”, “Đơn hàng #12345 đang được xử lý”).

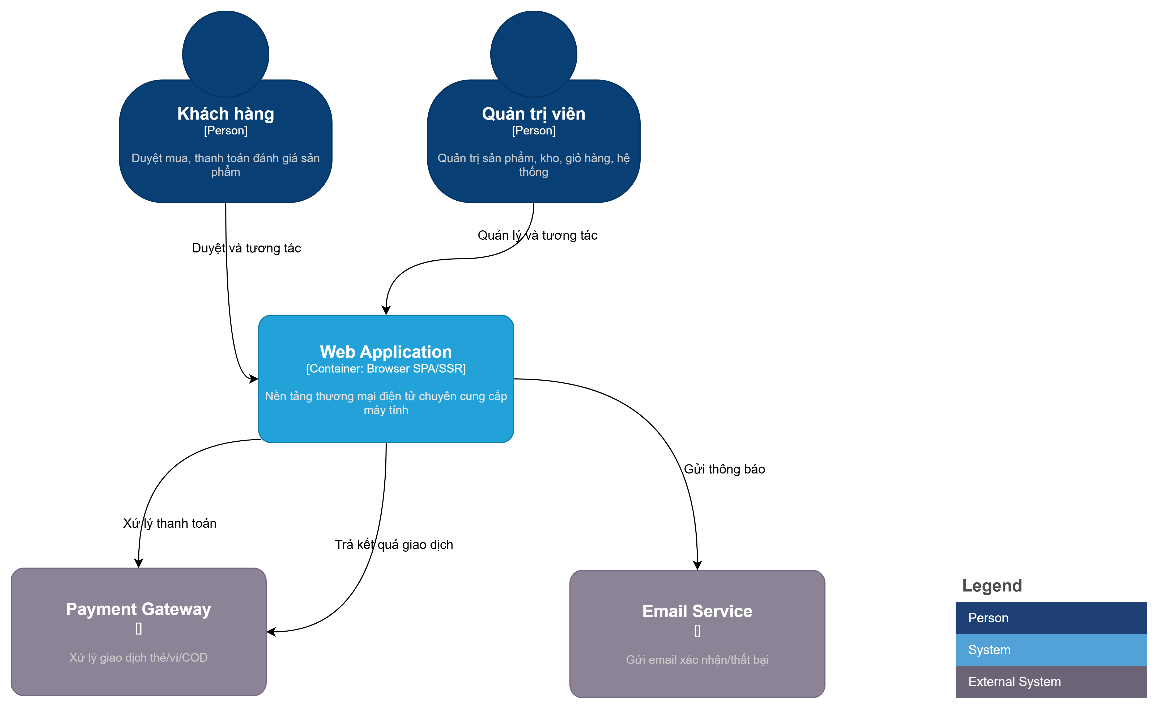
### 1.11.4. Ý nghĩa và mối liên hệ với sơ đồ khái niệm

* Sơ đồ khái niệm: tập trung vào nghiệp vụ (ai, cái gì, quan hệ gì). Ví dụ: Người dùng → tạo → Đơn hàng → chứa → Sản phẩm.
* Sơ đồ khối: tập trung vào kiến trúc triển khai, tức là những nghiệp vụ đó được hiện thực bằng module nào, chạy ở lớp nào, giao tiếp qua đâu.

Trong thực tế, sơ đồ khối này là bước nối giữa sơ đồ khái niệm và sơ đồ chi tiết (như sequence diagram hoặc activity diagram). Nó giúp người thiết kế hệ thống và người phát triển có cái nhìn tổng thể, từ đó dễ dàng phân công công việc, thiết kế API, và đảm bảo rằng các chức năng nghiệp vụ đã được ánh xạ đầy đủ vào các khối kỹ thuật.

## 1.12. Decomposition View: C4

### 1.12.1. sơ đồ C1 - System Context



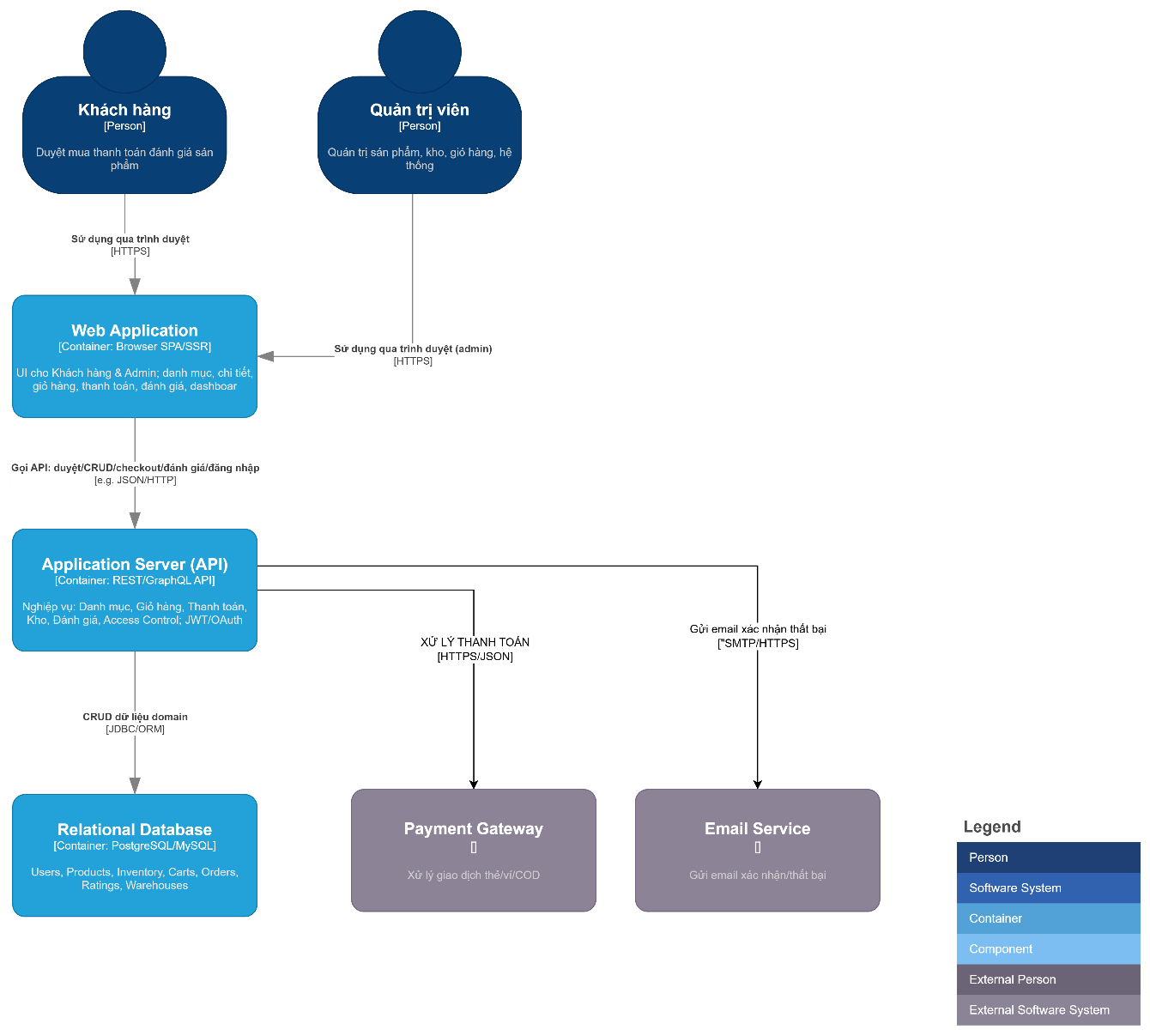
**1. Giải thích**

* My Meetings (Hệ thống): Là hệ thống chính đang được phân tích. Nó có vai trò quản lý các cuộc họp, nhóm họp và các thành viên.
* Member: Là người dùng thông thường của hệ thống. Có thể tổ chức, tham gia và quản lý các cuộc họp và là người thực hiện các giao dịch thanh toán.
* Administrator: Là người quản trị hệ thống, có quyền cao nhất để quản lý các thành viên, nhóm họp và cuộc họp.
* Payment Gateway: Là một dịch vụ bên ngoài, chuyên xử lý các giao dịch thanh toán.
* Email Service: Là một dịch vụ bên ngoài, dùng để gửi các email thông báo.

**2. Luồng hoạt động**

* Luồng từ Thành viên (Member):
  + Tổ chức & Tham gia cuộc họp**:** Thành viên tương tác trực tiếp với hệ thống "My Meetings" để tổ chức các cuộc họp, nhóm họp hoặc tham gia vào các cuộc họp đã có.
  + Thanh toán: Khi cần thanh toán, thành viên gửi yêu cầu "Pay in" đến "My Meetings". Hệ thống này sau đó sẽ "ủy thác việc thanh toán" cho Payment Gateway. Cổng thanh toán sẽ xử lý giao dịch và "trả về thông tin về việc thanh toán"
  + Gửi email: Thành viên có thể yêu cầu "My Meetings" "gửi email" cho họ hoặc người khác. Hệ thống sẽ chuyển yêu cầu này đến Email Service, và dịch vụ email sẽ thực hiện "gửi email".
* Luồng từ Quản trị viên (Administrator):
  + Quản lý hệ thống: Quản trị viên có quyền "quản lý các thành viên, nhóm họp, cuộc họp" thông qua hệ thống "My Meetings".
  + Gửi email: Tương tự như thành viên, quản trị viên cũng có thể sử dụng dịch vụ Email Service để gửi email.
* Luồng giữa các hệ thống:
  + My Meetings và Payment Gateway: My Meetings gửi yêu cầu thanh toán và nhận lại kết quả.
  + My Meetings và Email Service: My Meetings gửi yêu cầu gửi email và nhận thông báo trạng thái từ Email Service.

### 1.12.2. sơ đồ C2 – Container



**Giải thích**

* Web Application (Ứng dụng Web): Là giao diện người dùng, nơi khách hàng và quản trị viên tương tác với hệ thống. Được xây dựng dưới dạng một ứng dụng trang đơn (SPA) hoặc kết xuất phía máy chủ (SSR).
* Application Server (API): Xử lý logic nghiệp vụ, giỏ hàng, thanh toán, quản lý kho, và kiểm soát truy cập.
* Relational Database (Cơ sở dữ liệu quan hệ): Lưu trữ tất cả dữ liệu của hệ thống, bao gồm thông tin người dùng, sản phẩm, kho hàng, giỏ hàng, đơn hàng và đánh giá.
* Payment Gateway (Cổng thanh toán): Một hệ thống bên ngoài xử lý các giao dịch thanh toán.
* Email Service (Dịch vụ email): Một hệ thống bên ngoài dùng để gửi email xác nhận hoặc thông báo.

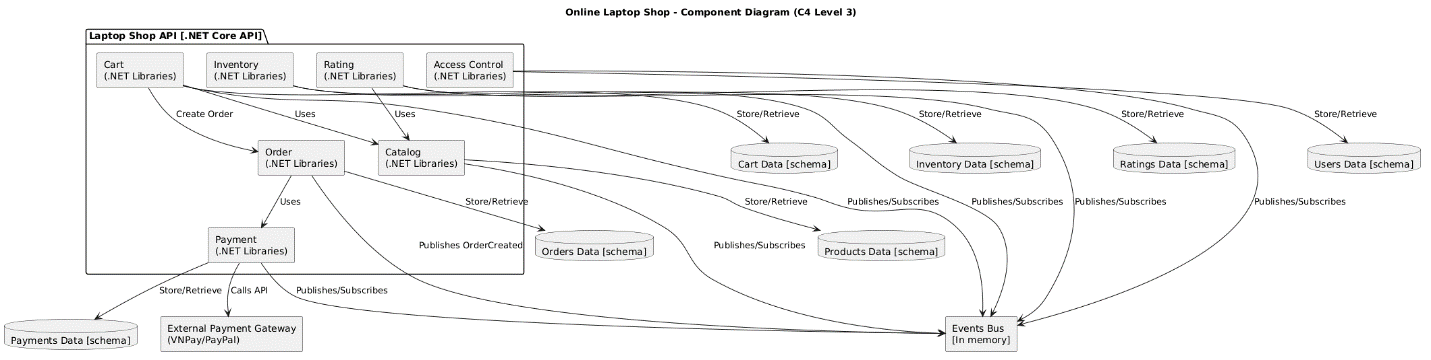
**Luồng hoạt động**

* Khách hàng và Quản trị viên sử dụng trình duyệt web (HTTP) để tương tác với Ứng dụng Web.
* Ứng dụng web sau đó gọi các API của Application Server (sử dụng JSON/HTTP) để thực hiện các chức năng như duyệt sản phẩm, thêm vào giỏ hàng, thanh toán, v.v.
* Luồng dữ liệu nội bộ:
* Application Server thực hiện các thao tác CRUD (tạo, đọc, cập nhật, xóa) trên dữ liệu trong Relational Database để quản lý thông tin.
* Tương tác với hệ thống bên ngoài:
* Khi có giao dịch thanh toán, Application Server sẽ gửi yêu cầu xử lý thanh toán (thường là qua JSON/HTTP) đến Payment Gateway.
* Khi cần gửi thông báo, Application Server sẽ gửi yêu cầu gửi email (thường qua SMTP/HTTPS) đến Email Service.

### 1.12.3. Sơ đồ C3 - Component (Module Level)



### 1.12.4. sơ đồ C3 – Component (High-Level)



**Các thành phần chính**

**1. Catalog Component**

* Chức năng: Quản lý danh mục sản phẩm laptop (CRUD sản phẩm, xem chi tiết).
* Giao tiếp:
  + Lưu/truy vấn dữ liệu từ Products Data schema.
  + Được Cart gọi khi người dùng thêm sản phẩm vào giỏ.
  + Được Rating gọi khi cần hiển thị sản phẩm kèm đánh giá.

**2. Cart Component**

* Chức năng: Quản lý giỏ hàng của người dùng (thêm/xóa/sửa, hiển thị giỏ).
* Giao tiếp:
  + Lưu/truy xuất dữ liệu từ Cart Data schema.
  + Gọi Catalog để kiểm tra thông tin sản phẩm.
  + Gọi Order khi người dùng bấm checkout.

**3. Order Component**

* Chức năng: Quản lý đơn hàng (tạo đơn, trạng thái, lịch sử đơn hàng).
* Giao tiếp:
  + Lưu/truy xuất dữ liệu từ Orders Data schema.
  + Nhận dữ liệu từ Cart khi checkout.
  + Gọi Payment để xử lý thanh toán.

**4. Payment Component**

* Chức năng: Quản lý thanh toán (phí thành viên, phí trễ hạn, thanh toán đơn hàng).
* Giao tiếp:
  + Lưu dữ liệu vào Payments Data schema.
  + Nhận yêu cầu từ Order để xử lý thanh toán.
  + Gọi External Payment Gateway (VNPay, PayPal) qua API.

**5. Inventory Component**

* Chức năng: Quản lý kho (số lượng tồn kho của từng laptop).
* Giao tiếp:
  + Lưu/truy vấn dữ liệu từ Inventory Data schema.
  + Giảm số lượng tồn kho khi Order thành công.

**6. Rating Component**

* Chức năng: Người dùng đánh giá sản phẩm (rating, comment).
* Giao tiếp:
  + Lưu/truy xuất dữ liệu từ Ratings Data schema.
  + Gọi Catalog để hiển thị đánh giá theo sản phẩm.

**7. Access Control Component**

* Chức năng: Kiểm soát truy cập (login, đăng ký, phân quyền User/Admin).
* Giao tiếp:
  + Lưu/truy vấn từ Users Data schema.
  + Các service khác sẽ dùng nó để xác thực người dùng (Cart, Order, Payment…).

**Các thành phần hỗ trợ**

**a. Database schemas**

* Mỗi component có schema riêng (Products, Cart, Orders, Payments, Inventory, Ratings, Users).
* Điều này tuân theo nguyên tắc Database-per-service trong microservices.

**b. Events Bus (In memory)**

* Đảm bảo giao tiếp bất đồng bộ giữa các component.

**c. External Payment Gateway**

* Dịch vụ bên ngoài (VNPay, PayPal).
* Payment gọi API để xác thực và xử lý giao dịch.

**2.4. Sơ đồ C4: Implementation**



**1. Nhóm thực thể chính**

1. User (Người dùng)

* Thuộc module kiểm soát truy cập.
* Thuộc tính:
  + ID: định danh người dùng
  + Email: dùng để đăng nhập.
  + Mật\_khẩu: thông tin bảo mật.
* Đây là lớp cha, đóng vai trò trừu tượng cho các loại người dùng khác.

1. Buyer (Người mua)

* Kế thừa từ User.
* Thuộc tính:  
  + Tên, Số\_điện\_thoại.
* Phương thức:  
  + Lọc\_sản\_phẩm(): lọc sản phẩm theo tiêu chí.
  + Xem\_chi\_tiết\_sản\_phẩm(): xem thông tin chi tiết sản phẩm.
* Vai trò: đại diện cho khách hàng mua laptop.

1. Admin (Quản trị viên)

* **Kế thừa từ User**.
* Phương thức:  
  + CRUD\_sản\_phẩm(): thực hiện thêm, sửa, xóa, xem sản phẩm.
  + Gán\_số\_lượng\_tồn\_kho(): gán số lượng tồn ban đầu cho sản phẩm.
* Vai trò: quản lý sản phẩm và số lượng tồn trong kho.

**2. Nhóm thực thể về kho và sản phẩm**

1. Kho (Warehouse)

* Thuộc module Tồn kho.
* Thuộc tính:
  + ID, Tên\_kho, Tổng\_số\_lượng, Địa\_điểm.
* Là lớp cha cho các thông tin chi tiết tồn kho.

1. InventoryInfo (Thông tin tồn kho)

* Kế thừa từ Kho.
* Thuộc tính:
  + Số\_lượng\_tồn: số lượng hiện
  + Ngày\_cập\_nhật: ngày cập nhật tồn kho gần nhất.
* Vai trò: cung cấp thông tin tồn kho cụ thể cho từng sản phẩm.

1. Product (Sản phẩm)

* Thuộc tính:
* ID, Tên, Giá, Hình\_ảnh, Info\_tồn\_kho\_ID.
* Phương thức:
* Xem\_thông\_tin(): hiển thị chi tiết sản phẩm.
* Vai trò: đối tượng trung tâm mà cả Buyer và Admin đều thao tác.

1. HotProduct (Sản phẩm hot)

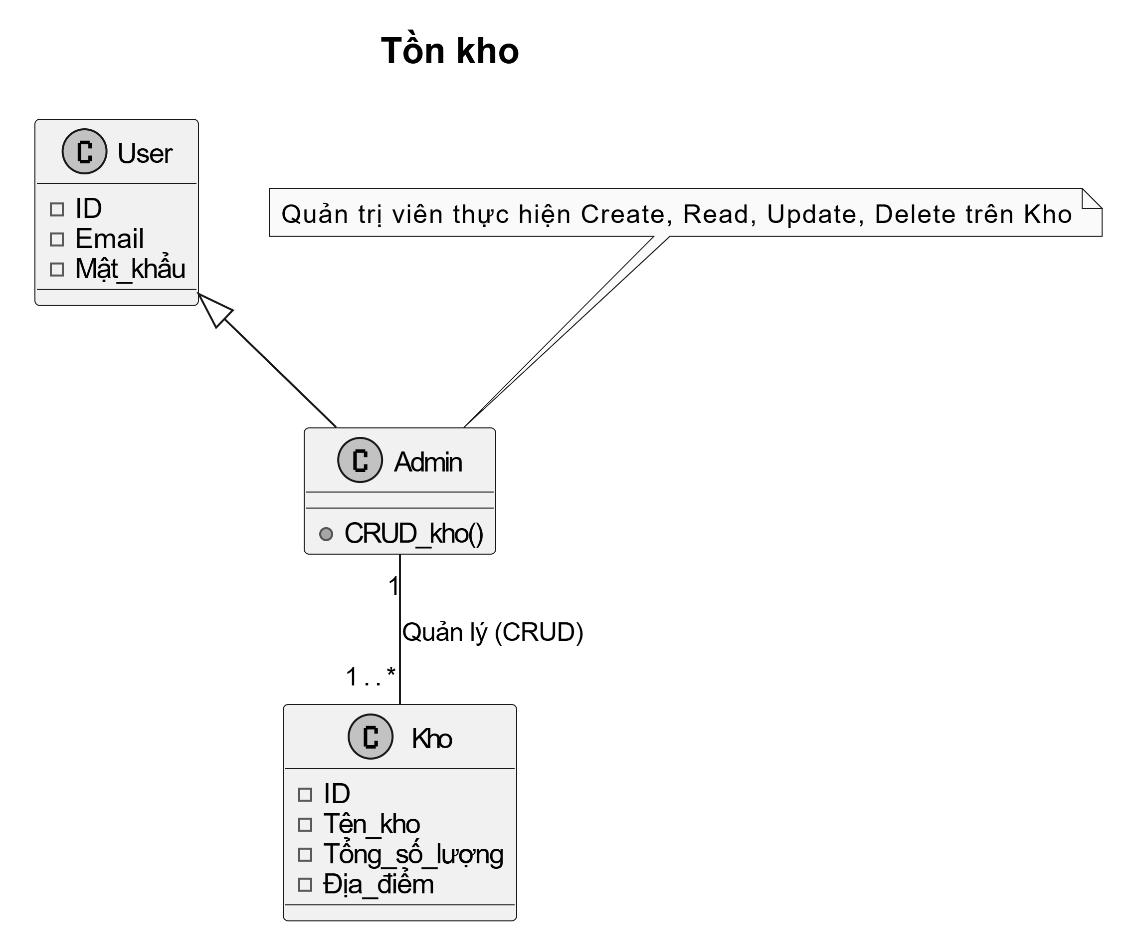
* Kế thừa từ Product.
* Thuộc tính:
  + Ưu\_đãi.
* Phương thức:
  + Xem\_ưu\_đãi().
* Vai trò: thể hiện sản phẩm đặc biệt, có khuyến mãi/ưu đãi.

3. Quan hệ giữa các lớp

* User <|-- Buyer và User <|-- Admin: thể hiện kế thừa.
* Kho <|-- InventoryInfo: kế thừa để chuyên biệt hóa thông tin kho.
* Product <|-- HotProduct: kế thừa để mở rộng sản phẩm thường thành sản phẩm hot.
* InventoryInfo "1" -- "1..\*" Product: một thông tin tồn kho gắn với nhiều sản phẩm.
* Buyer "1" -- "1..\*" Product: Buyer có thể lọc nhiều sản phẩm.
* Buyer -- Product (chi tiết): Buyer có thể xem chi tiết sản phẩm.
* Admin -- Product: Admin quản lý sản phẩm (CRUD và gán số lượng tồn).

4. Ý nghĩa sơ đồ

* User là thực thể gốc, chia thành Buyer và Admin với vai trò khác nhau.
* Product là trung tâm, kết nối với cả Buyer (xem, lọc) và Admin (quản lý).
* Kho và InventoryInfo gắn liền với sản phẩm để quản lý tồn kho.
* Các mối quan hệ kế thừa và kết nối giúp hệ thống vừa có tính tái sử dụng (User chung), vừa có tính mở rộng (HotProduct từ Product).



**1. Thực thể chính**

1. User (Người dùng)

* Nằm trong module Kiểm soát truy cập.
* Thuộc tính:
* ID: định danh duy nhất cho mỗi người dùng.
* Email: thông tin đăng nhập.
* Mật\_khẩu: dùng để xác thực tài khoản.
* Đây là lớp cha, từ đó sinh ra Admin.

1. Admin (Quản trị viên)

* Kế thừa từ User.
* Phương thức:
* CRUD\_kho(): thực hiện các thao tác cơ bản với kho:

1. Create: Tạo kho mới.
2. Read: Xem thông tin kho.
3. Update: Cập nhật thông tin kho.
4. Delete: Xóa kho.
5. Kho (Inventory/Warehouse)

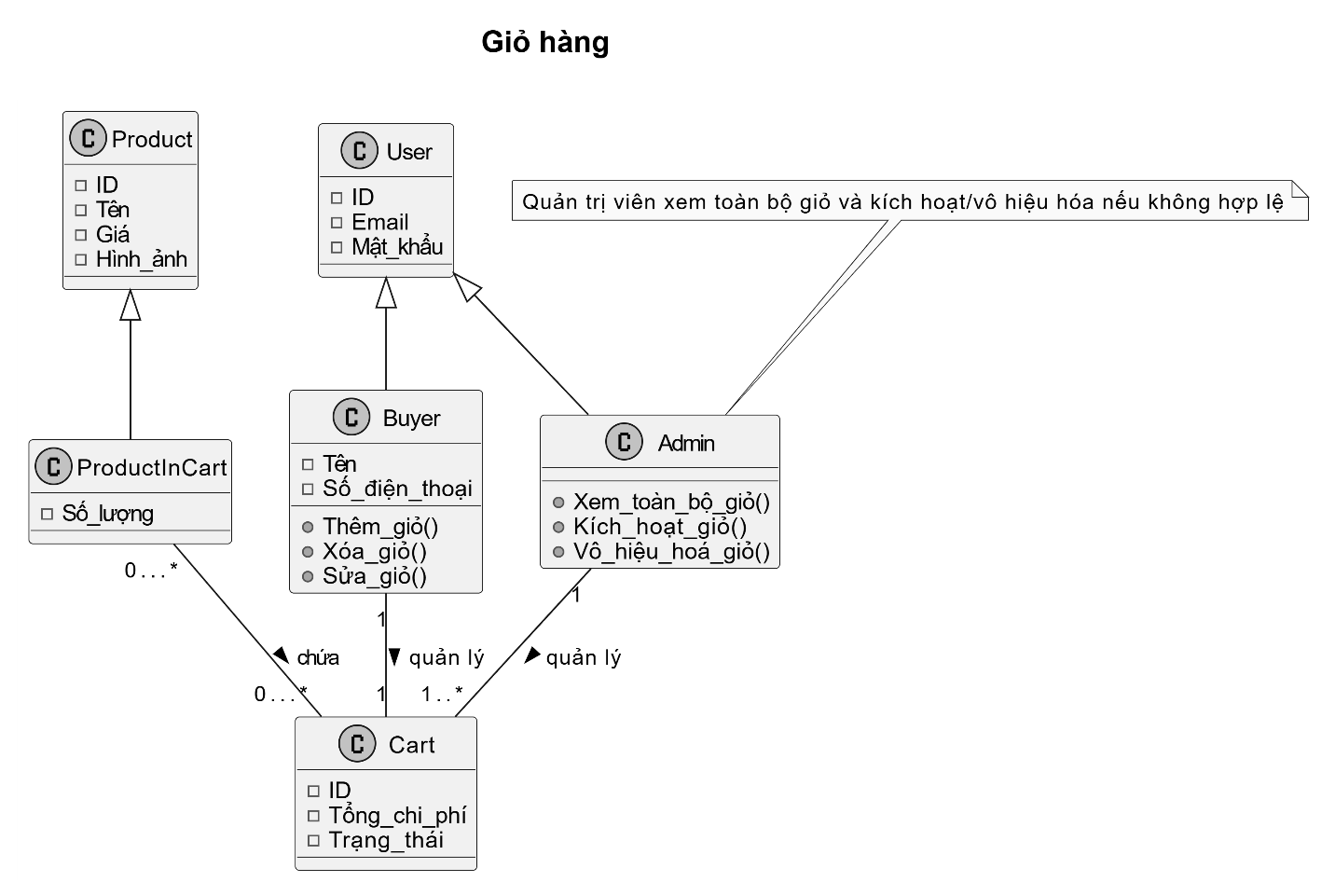
* Thuộc tính:
* ID: mã định danh của kho.
* Tên\_kho: tên gọi của kho.
* Tổng\_số\_lượng: tổng số lượng sản phẩm/đơn vị đang có trong kho.
* Địa\_điểm: vị trí kho hàng.
* Đây là thực thể được quản trị viên thao tác trực tiếp.

**2. Quan hệ giữa các lớp**

* User <|-- Admin: Admin là một loại người dùng, được kế thừa từ lớp User.
* Admin "1" -- "1..\*" Kho: Một quản trị viên có thể quản lý nhiều kho (ít nhất một).
* Mối quan hệ được gắn nhãn: Quản lý (CRUD) để nhấn mạnh rằng Admin có quyền đầy đủ thao tác trên kho.

**3. Ý nghĩa sơ đồ**

* Sơ đồ mô tả **chức năng quản lý kho** trong hệ thống:
* **User** là cơ sở cho các loại người dùng khác nhau.
* **Admin** có quyền cao nhất với kho, đảm nhận các thao tác CRUD.
* **Kho** là tài nguyên trọng tâm cần quản lý.
* Ghi chú ở trên Admin làm rõ chức năng CRUD mà quản trị viên có thể thực hiện.



**1. Nhóm thực thể chính**

1. User (Người dùng)

* Nằm trong module Kiểm soát truy cập.
* Thuộc tính:
* ID: định danh người dùng.
* Email: dùng để đăng nhập.
* Mật\_khẩu: thông tin bảo mật.
* Là lớp cha, từ đó sinh ra hai vai trò chính: Buyer và Admin.

1. Buyer (Người mua)

* Kế thừa từ User.
* Thuộc tính:
* Tên: tên người mua.
* Số\_điện\_thoại: thông tin liên hệ.
* Phương thức:
* Thêm\_giỏ(): thêm sản phẩm vào giỏ.
* Xóa\_giỏ(): xóa sản phẩm trong giỏ.
* Sửa\_giỏ(): thay đổi số lượng sản phẩm trong giỏ.
* Vai trò: khách hàng thao tác trực tiếp với giỏ hàng.

1. Admin (Quản trị viên)

* Kế thừa từ User.
* Phương thức:
* Xem\_toàn\_bộ\_giỏ(): xem tất cả giỏ hàng trong hệ thống.
* Kích\_hoạt\_giỏ(): cho phép giỏ hoạt động.
* Vô\_hiệu\_hoá\_giỏ(): tạm ngưng hoặc xóa giỏ không hợp lệ.
* Vai trò: giám sát và quản lý các giỏ hàng.

**2. Nhóm thực thể sản phẩm và giỏ**

1. Product (Sản phẩm)

* Nằm trong module Danh mục sản phẩm.
* Thuộc tính:
* ID, Tên, Giá, Hình\_ảnh.
* Là lớp cơ bản cho mọi loại sản phẩm.

1. ProductInCart (Sản phẩm trong giỏ)

* Kế thừa từ Product.
* Thuộc tính:
* Số\_lượng: số lượng sản phẩm người mua chọn.
* Vai trò: cụ thể hóa sản phẩm khi nó đã được đưa vào giỏ.

1. Cart (Giỏ hàng)

* Thuộc tính:
* ID: mã giỏ hàng.
* Tổng\_chi\_phí: tổng giá trị sản phẩm trong giỏ.
* Trạng\_thái: cho biết giỏ đang hoạt động, bị vô hiệu, hay đã thanh toán.
* Vai trò: trung tâm chứa các sản phẩm được chọn bởi người mua.

**3. Quan hệ giữa các lớp**

* Product <|-- ProductInCart: sản phẩm trong giỏ là một loại mở rộng của sản phẩm.
* User <|-- Buyer, User <|-- Admin: Buyer và Admin đều kế thừa từ User.
* ProductInCart "0..*" -- "0..*" Cart: một giỏ chứa nhiều sản phẩm, và một sản phẩm có thể nằm trong nhiều giỏ.
* Buyer "1" -- "1" Cart: một người mua gắn với đúng một giỏ hàng.
* Admin "1" -- "1..\*" Cart: một quản trị viên có thể quản lý nhiều giỏ hàng.

**4. Ý nghĩa sơ đồ**

* Buyer đại diện cho khách hàng: thêm, sửa, xóa giỏ hàng.
* Admin có quyền cao hơn: giám sát toàn bộ giỏ, kích hoạt/vô hiệu hóa khi cần.
* Cart là trung tâm kết nối Buyer và các sản phẩm.
* ProductInCart cho thấy sự khác biệt giữa “sản phẩm” (Product) và “sản phẩm được thêm vào giỏ” (có số lượng cụ thể).





**1. Thực thể chính**

1. User (Người dùng)

* Thuộc module Kiểm soát truy cập.
* Thuộc tính:
* ID: định danh duy nhất.
* Email: email đăng nhập.
* Mật\_khẩu: thông tin bảo mật.
* Đây là lớp cơ sở cho các loại người dùng.

1. Buyer (Người mua)

* Kế thừa từ User.
* Thuộc tính:
* Tên: tên của người mua.
* Số\_điện\_thoại: liên hệ.
* Phương thức:
* Nhapthongtin(): nhập thông tin cần thiết để đặt hàng/thanh toán.

1. Cart (Giỏ hàng)

* Thuộc module Giỏ hàng.
* Thuộc tính:
* ID: mã giỏ.
* Tổng\_chi\_phí: tổng giá trị sản phẩm trong giỏ.
* Trạng\_thái: trạng thái giỏ (mở, chờ thanh toán, đã thanh toán).

1. CartInfo (Thông tin giỏ)

* Kế thừa từ Cart.
* Thuộc tính:
* Sản\_phẩm\_danh\_sách: chứa danh sách sản phẩm trong giỏ.

1. Payment (Thanh toán)

* Là trung tâm của module thanh toán.
* Thuộc tính:
* ID: mã thanh toán.
* Tổng\_tiền: số tiền cần trả.
* Thời\_gian: thời điểm giao dịch.
* PaymentMethod\_ID: phương thức thanh toán (liên kết với PaymentMethod).
* PaymentStatus\_ID: trạng thái thanh toán (liên kết với PaymentStatus)
* Địa\_chỉ\_giao: nơi giao hàng
* Danh\_sách\_sản\_phẩm: sản phẩm được thanh toán.
* Tên\_người\_mua: tên người thực hiện thanh toán.
* Số\_điện\_thoại: số liên hệ khi giao hàng.
* Phương thức:
* Thực\_hiện\_thanh\_toán(): tiến hành xử lý thanh toán.

1. PaymentMethod (Phương thức thanh toán)

* Thuộc tính:
* ID: mã phương thức.
* Tên\_phương\_thức: ví dụ: thẻ tín dụng, ví điện tử, COD.
* Mô\_tả: chi tiết thêm.

1. PaymentStatus (Trạng thái thanh toán)

* Thuộc tính:
* ID: mã trạng thái.
* Tên\_trạng\_thái: ví dụ: Thành công, Thất bại, Đang chờ.
* Mô\_tả: diễn giải chi tiết.

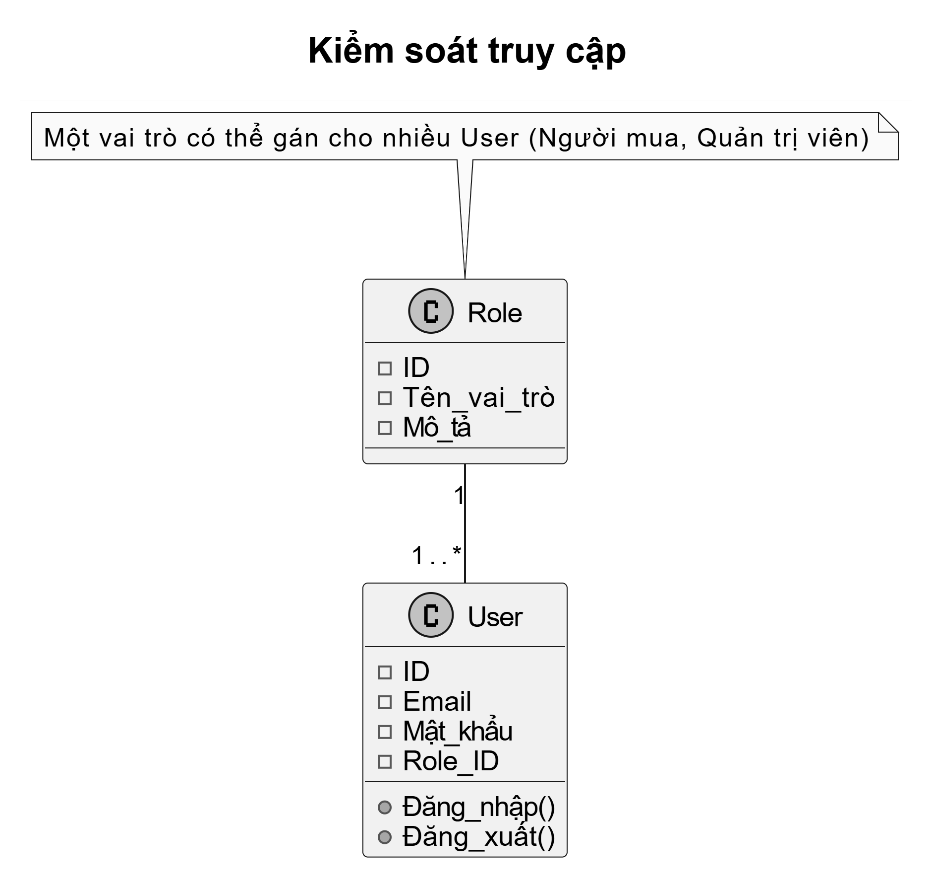
**2. Mối quan hệ giữa các lớp**

* Buyer "1" -- "0..\*" Payment: Một người mua có thể thực hiện nhiều giao dịch thanh toán.
* CartInfo "1" -- "1" Payment: Mỗi thanh toán được tạo từ một giỏ hàng.
* Payment "1" -- "1" PaymentMethod: Một giao dịch thanh toán chỉ sử dụng duy nhất một phương thức.
* PaymentStatus "1" -- "1..\*" Payment: Một trạng thái có thể áp dụng cho nhiều giao dịch khác nhau (ví dụ: nhiều giao dịch cùng ở trạng thái “Thành công”).

**3. Ý nghĩa sơ đồ**

* Sơ đồ thể hiện rõ luồng từ người mua → giỏ hàng → thanh toán → phương thức + trạng thái.
* Nhấn mạnh tính kế thừa (User → Buyer, Cart → CartInfo) và ràng buộc quan hệ 1-1, 1-nhiều giữa các thực thể.
* Hệ thống đảm bảo:
* Mỗi giao dịch luôn gắn với một giỏ hàng.
* Thanh toán chỉ dùng một phương thức tại một thời điểm.
* Trạng thái thanh toán được tái sử dụng, không cần tạo mới cho từng giao dịch.

Nhờ vậy, mô hình vừa rõ ràng về **vai trò thực thể**, vừa đảm bảo **toàn vẹn dữ liệu** trong quy trình thanh toán.



**1. Thực thể chính**

1. User (Người dùng)

* Đại diện cho mọi đối tượng đăng nhập vào hệ thống.
* Thuộc tính:
* ID: định danh duy nhất cho người dùng.
* Email: dùng để đăng nhập hoặc liên lạc.
* Mật\_khẩu: thông tin bảo mật cho việc xác thực.
* Role\_ID: liên kết đến vai trò mà người dùng đảm nhận.
* Phương thức:
* Đăng\_nhập(): xác thực thông tin và cho phép truy cập.
* Đăng\_xuất(): kết thúc phiên làm việc của người dùng.

1. Role (Vai trò)

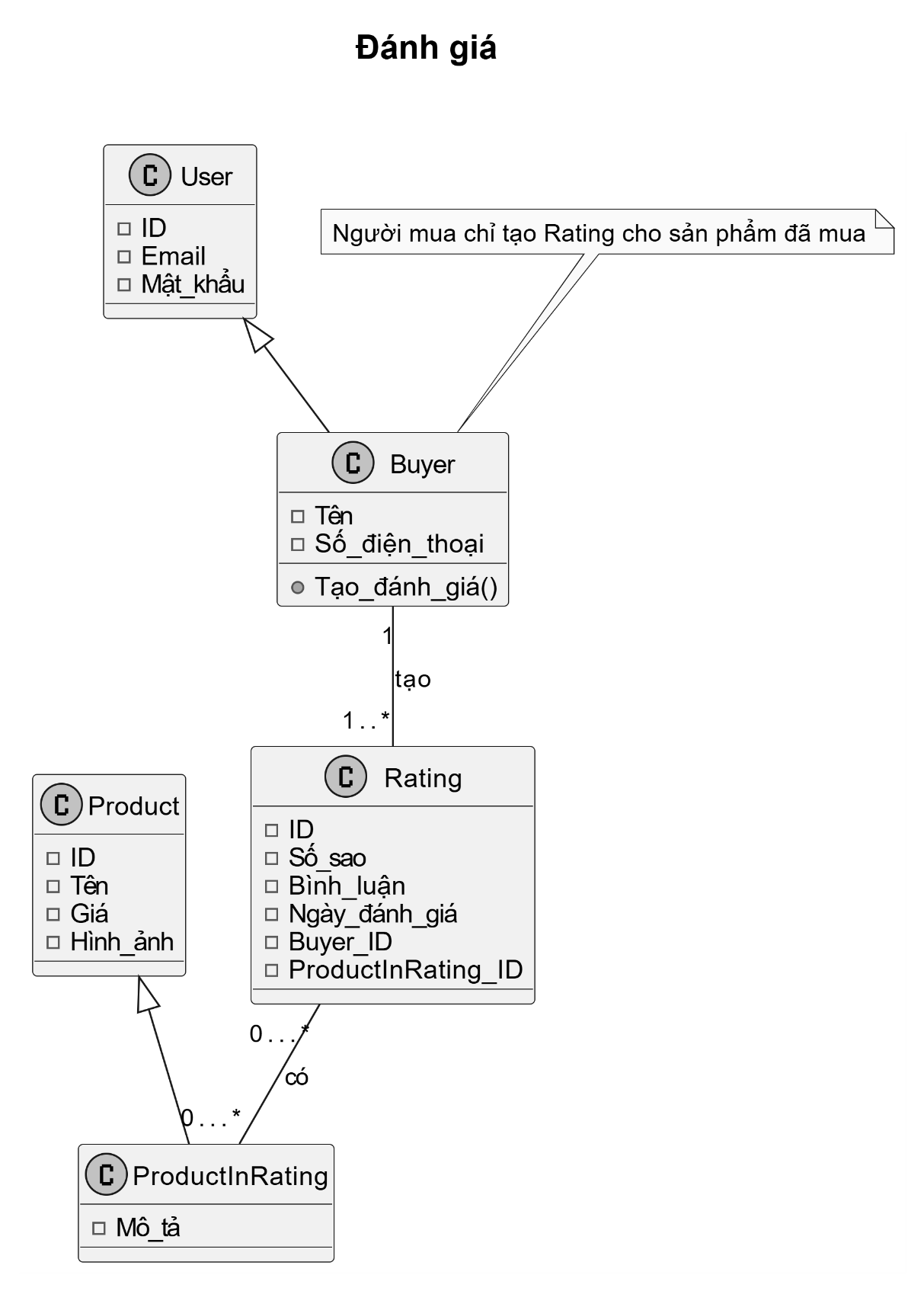
* Đại diện cho **nhóm quyền hạn** trong hệ thống.
* Thuộc tính:
* ID: định danh vai trò.
* Tên\_vai\_trò: tên gọi, ví dụ: *Buyer*, *Admin*.
* Mô\_tả: diễn giải chi tiết về vai trò.

**2. Mối quan hệ**

* Role "1" -- "1..\*" User
* Một vai trò có thể gán cho nhiều người dùng khác nhau.

**3. Ý nghĩa sơ đồ**

* Sơ đồ này nằm trong module kiểm soát truy cập (Access Control).
* Nó giúp quản lý hệ thống theo hướng Role-based Access Control (RBAC):
* Người dùng không được gắn trực tiếp quyền, mà gián tiếp thông qua vai trò.
* Quản lý dễ dàng hơn: chỉ cần thay đổi quyền trong Role, tất cả User thuộc Role đó sẽ thay đổi theo.
* Đây là thiết kế chuẩn trong nhiều hệ thống web/app hiện đại.



**1. Thực thể chính**

1. Product (Sản phẩm)

* Thuộc module Danh mục sản phẩm.
* Thuộc tính:
* ID: mã sản phẩm duy nhất.
* Tên: tên sản phẩm (ví dụ: Laptop Dell XPS).
* Giá: giá bán.
* Hình\_ảnh: hình minh họa cho sản phẩm.

1. User (Người dùng)

* Thuộc module **Kiểm soát truy cập**.
* Thuộc tính:
* ID: định danh duy nhất cho người dùng.
* Email: thông tin đăng nhập/liên lạc.
* Mật\_khẩu: bảo mật.

1. Buyer (Người mua)

* Kế thừa từ User.
* Thuộc tính:
* Tên: tên đầy đủ của người mua.
* Số\_điện\_thoại: liên lạc.
* Phương thức:
* Tạo\_đánh\_giá(): thực hiện việc viết đánh giá cho sản phẩm.

1. ProductInRating (Sản phẩm trong đánh giá)

* Kế thừa từ Product.
* Thuộc tính:
* Mô\_tả: phần mô tả chi tiết sản phẩm trong ngữ cảnh đánh giá (có thể khác với mô tả chính thức).

1. Rating (Đánh giá)

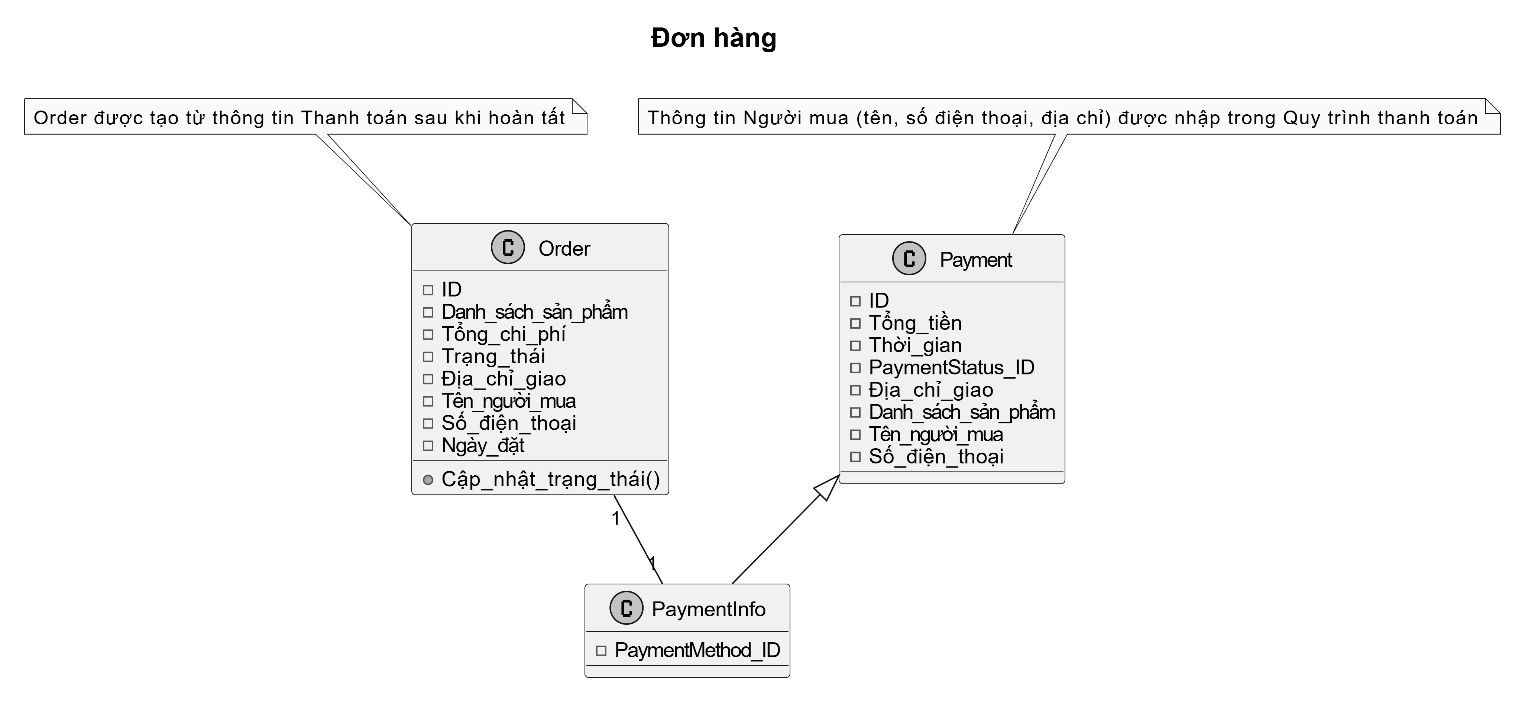
* Đại diện cho một đánh giá mà người mua để lại.
* Thuộc tính:
* ID: mã định danh đánh giá.
* Số\_sao: số sao (ví dụ: 1–5).
* Bình\_luận: nội dung bình luận.
* Ngày\_đánh\_giá: thời điểm viết đánh giá.
* Buyer\_ID: liên kết đến người mua đã viết đánh giá.
* ProductInRating\_ID: liên kết đến sản phẩm được đánh giá.

**2. Mối quan hệ**

* Buyer "1" -- "1..\*" Rating:
* Mỗi người mua có thể tạo nhiều đánh giá cho các sản phẩm khác nhau.
* Nhưng một đánh giá chỉ thuộc về duy nhất một người mua.
* Rating "0..*" -- "0..*" ProductInRating:
* Một sản phẩm có thể nhận nhiều đánh giá từ các người mua khác nhau.
* Ngược lại, một đánh giá gắn với một sản phẩm cụ thể.

**3. Ý nghĩa sơ đồ**

* Buyer đóng vai trò chủ động tạo đánh giá.
* Rating là đối tượng trung gian chứa thông tin sao và bình luận.
* ProductInRating cho thấy rằng việc đánh giá gắn liền với sản phẩm trong ngữ cảnh đánh giá (kế thừa từ Product).
* Mô hình giúp quản lý được
* Ai đánh giá.
* Sản phẩm nào được đánh giá.
* Nội dung và chất lượng đánh giá.
* Nhờ mối quan hệ rõ ràng, hệ thống có thể dễ dàng thống kê mức độ hài lòng của khách hàng, lọc đánh giá theo sản phẩm, và ngăn chặn đánh giá giả mạo (vì Buyer phải mua hàng trước).



**1. Thực thể chính**

1. Payment (Thanh toán)

* Thuộc module Quy trình thanh toán.
* Thuộc tính:
* ID: mã định danh giao dịch thanh toán.
* Tổng\_tiền: số tiền phải thanh toán.
* Thời\_gian: thời điểm giao dịch diễn ra.
* PaymentStatus\_ID: trạng thái thanh toán (thành công, thất bại, đang xử lý).
* Địa\_chỉ\_giao: nơi nhận hàng.
* Danh\_sách\_sản\_phẩm: danh sách sản phẩm được thanh toán
* Tên\_người\_mua: thông tin người thực hiện giao dịch.
* Số\_điện\_thoại: số điện thoại liên hệ khi giao hàng.

1. PaymentInfo (Thông tin thanh toán)

* Kế thừa từ Payment.
* Thuộc tính bổ sung:
* PaymentMethod\_ID: phương thức thanh toán được chọn (VD: thẻ tín dụng, COD, ví điện tử).
* Đây là lớp mở rộng để lưu **chi tiết phương thức thanh toán**, ngoài những thông tin cơ bản của Payment.

1. Order (Đơn hàng)

* Đại diện cho đơn hàng được tạo sau khi thanh toán thành công.
* Thuộc tính:
* ID: mã định danh đơn hàng.
* Danh\_sách\_sản\_phẩm: các sản phẩm thuộc đơn hàng.
* Tổng\_chi\_phí: tổng số tiền của đơn hàng (có thể trùng với Payment).
* Trạng\_thái: trạng thái đơn hàng (đang xử lý, đã giao, bị hủy).
* Địa\_chỉ\_giao: địa chỉ giao hàng (lấy từ Payment).
* Tên\_người\_mua: tên khách hàng.
* Số\_điện\_thoại: số điện thoại liên hệ.
* Ngày\_đặt: ngày mà đơn hàng được tạo.
* Phương thức:
* Cập\_nhật\_trạng\_thái(): thay đổi trạng thái đơn hàng trong quá trình xử lý.

**2. Mối quan hệ**

* Order "1" -- "1" PaymentInfo:
* Mỗi Order được sinh ra từ đúng một PaymentInfo (nghĩa là một giao dịch thanh toán).
* Ngược lại, một PaymentInfo chỉ tạo ra duy nhất một Order tương ứng.
* Đây là mối quan hệ 1-1, bảo đảm rằng một giao dịch thanh toán chỉ tạo ra đúng một đơn hàng, và mỗi đơn hàng phải có một giao dịch hợp lệ.

**3. Ý nghĩa sơ đồ**

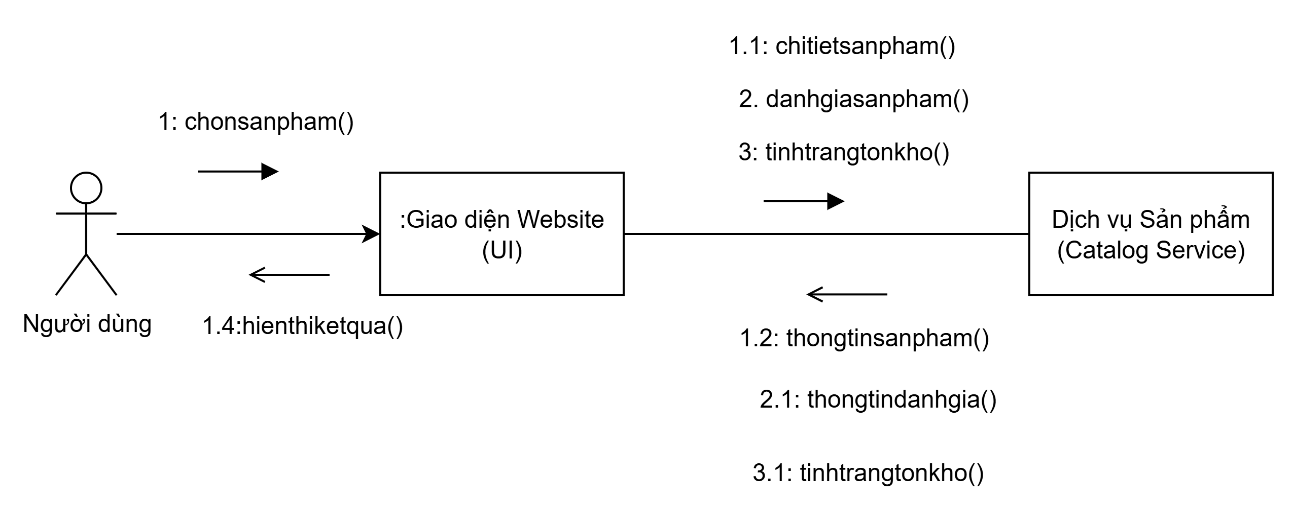
* Sơ đồ này thể hiện mối quan hệ trực tiếp giữa Thanh toán và Đơn hàng.
* Các điểm chính:
* Quy trình thanh toán diễn ra trước, khi thành công thì sinh ra Order.
* Thông tin người mua và địa chỉ giao hàng được nhập ở bước thanh toán, sau đó được chuyển nguyên sang Order.
* Order đóng vai trò theo dõi quá trình sau thanh toán: vận chuyển, cập nhật trạng thái, quản lý sản phẩm.
* Mô hình này giúp đảm bảo:
* Dữ liệu đồng nhất (Order và Payment chia sẻ cùng thông tin người mua, sản phẩm, địa chỉ).
* Quy trình nghiệp vụ rõ ràng: Payment thành công → Order được tạo → Quản lý đơn hàng tiếp tục xử lý.

## 1.13.Communication View

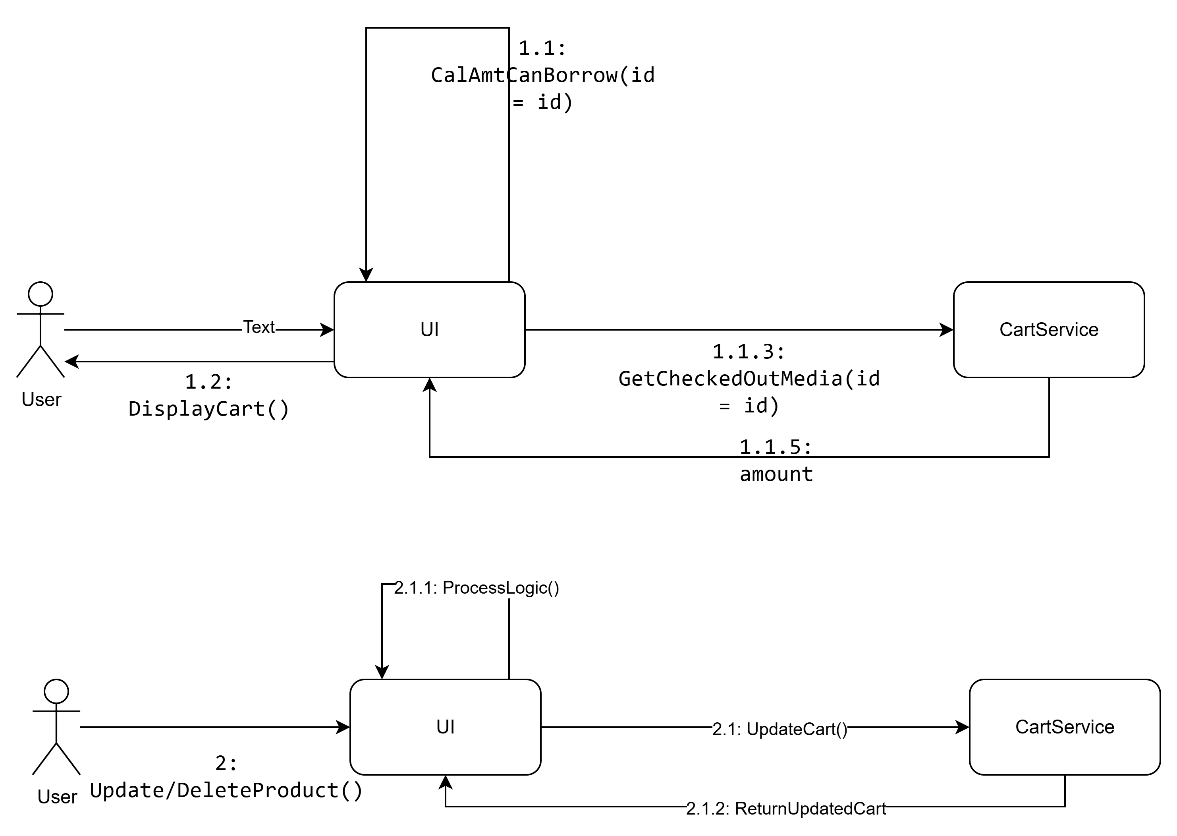
**1.13.1. Xem và Lọc/Sắp xếp Sản phẩm**

****

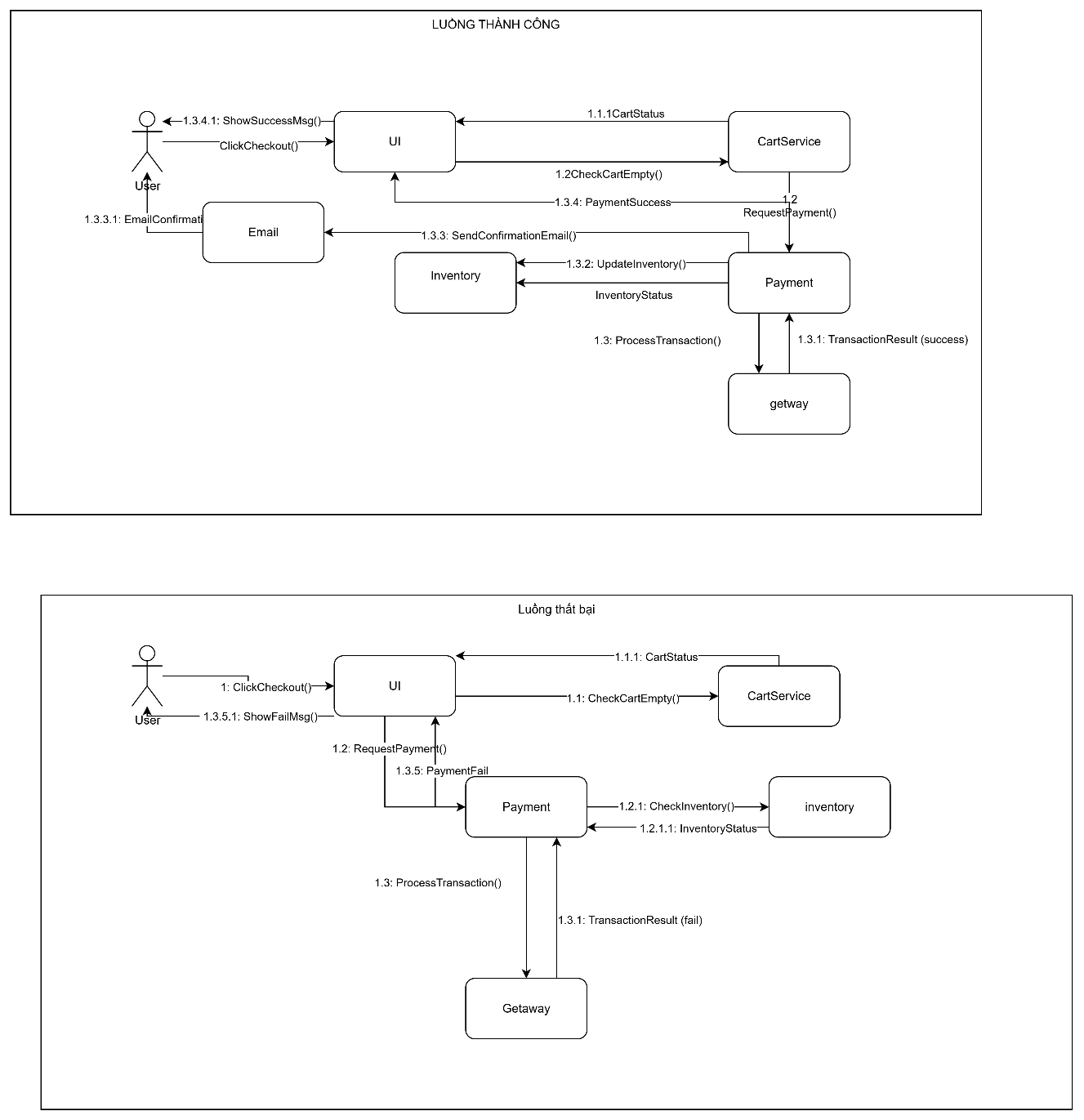
**3.2. Thêm sản phẩm vào giỏ hang**



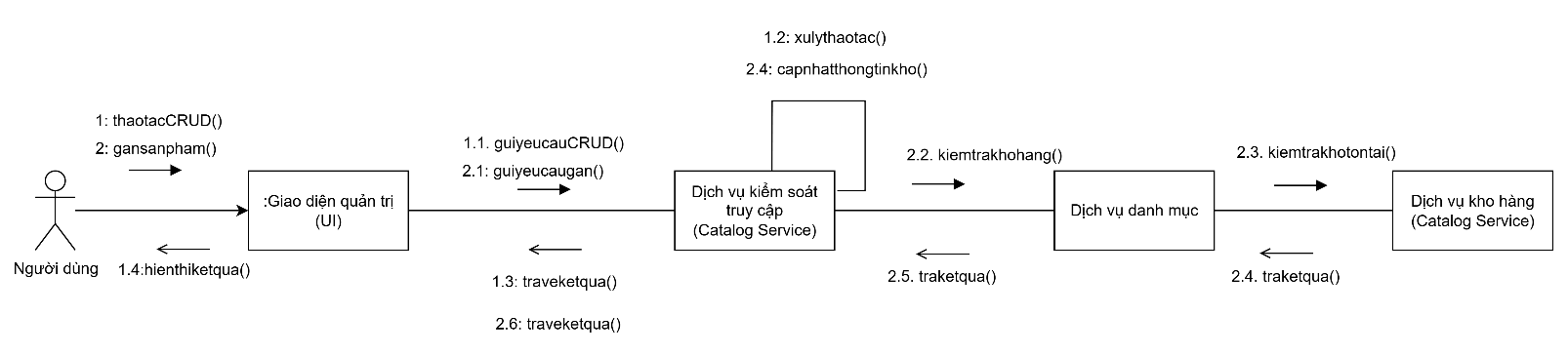
**3.3. Quản Lý Giỏ Hàng**



**3.4. Thanh toán đơn hàng**



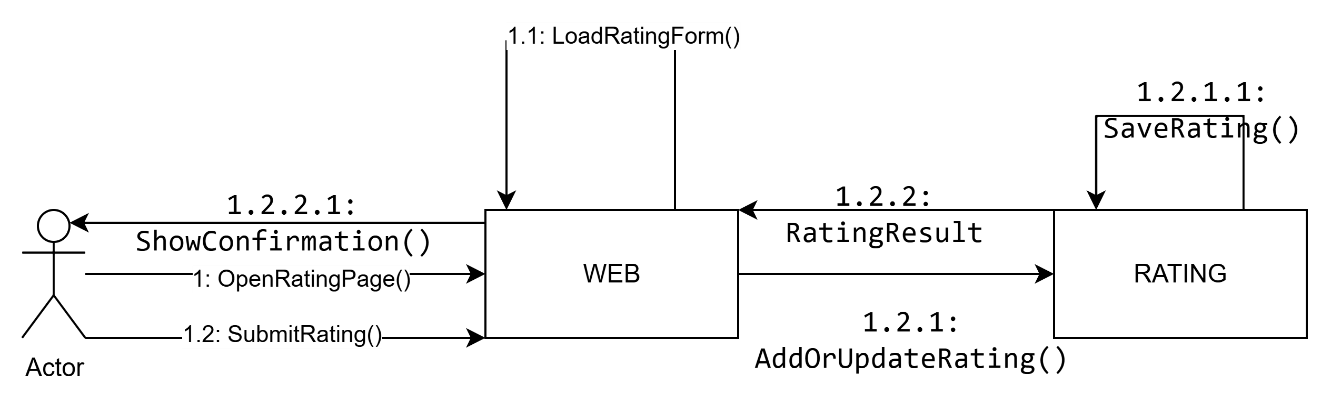
**3.5. Quản lý sản phẩm ( Admin)**



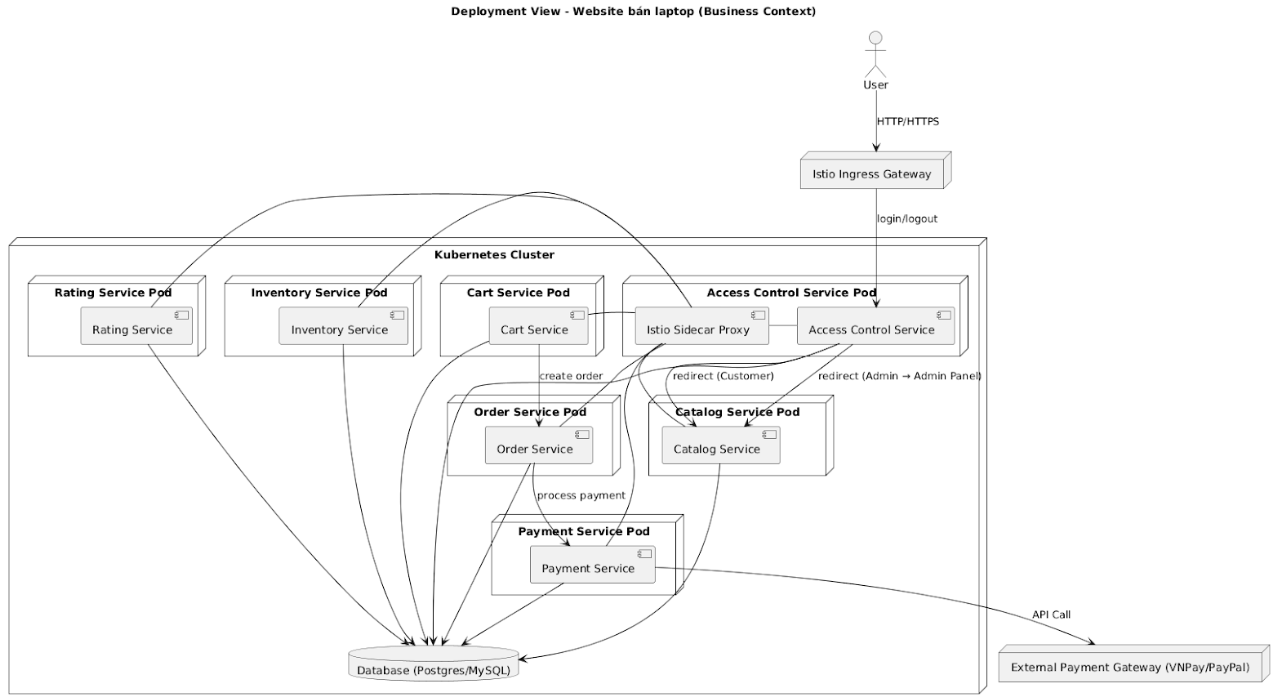
**3.6. Xem Lịch sử Đơn hàng**



**3.7. Đánh giá sản phẩm**



## 1.14. Deployment View



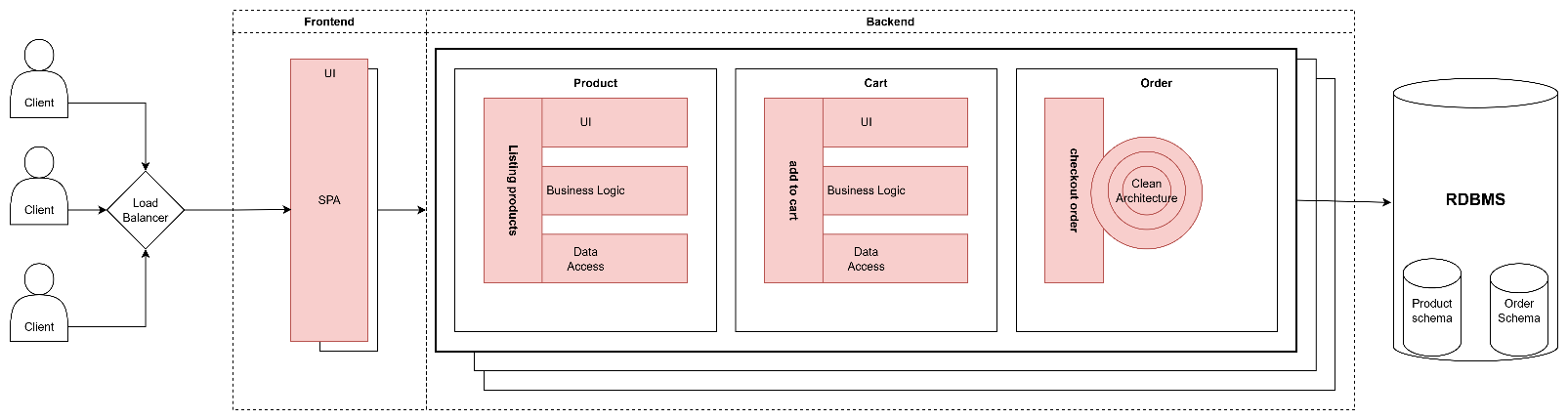
**Thành phần chính trong sơ đồ**

1. **User (Người dùng)**
   * Gồm cả **Khách hàng** và **Quản trị viên**.
   * Gửi yêu cầu qua **HTTP/HTTPS** để truy cập hệ thống.
2. **Istio Ingress Gateway**
   * Cổng vào duy nhất của hệ thống trên Kubernetes.
   * Nhận tất cả request từ User và điều hướng đến đúng service bên trong cluster.
3. **Access Control Service Pod**
   * Nhiệm vụ: **đăng nhập / đăng xuất (login/logout)**.
   * Sau khi User login thành công:
     + Nếu là **Khách hàng** → redirect đến **Catalog Service** (trang danh mục sản phẩm).
     + Nếu là **Quản trị viên** → redirect đến **Catalog Service** nhưng với giao diện **Admin Panel**.
4. **Catalog Service Pod**
   * Quản lý danh mục sản phẩm.
   * Là điểm truy cập đầu tiên sau khi login thành công.
   * Quản trị viên dùng để thêm/sửa/xóa sản phẩm.
   * Người dùng bình thường dùng để xem/lọc sản phẩm.
5. **Cart Service Pod**
   * Quản lý giỏ hàng.
   * Cho phép người dùng thêm, xóa, cập nhật sản phẩm trong giỏ.
   * Tích hợp với **Order Service** để tạo đơn hàng.
6. **Order Service Pod**
   * Quản lý quy trình đặt hàng.
   * Nhận dữ liệu từ giỏ hàng (**Cart Service**) để tạo đơn hàng mới.
   * Liên kết với **Payment Service** để xử lý thanh toán.
7. **Payment Service Pod**
   * Xử lý thanh toán cho đơn hàng.
   * Kết nối với **External Payment Gateway** (như VNPay, PayPal) để thực hiện giao dịch thực.
   * Cập nhật trạng thái thanh toán trong Database.
8. **Inventory Service Pod**
   * Quản lý tồn kho sản phẩm.
   * Mỗi lần có đơn hàng → giảm số lượng tồn.
   * Quản trị viên có thể nhập thêm hàng.
9. **Rating Service Pod**
   * Cho phép khách hàng đánh giá sản phẩm.
   * Lưu trữ rating vào Database.
   * Hỗ trợ hiển thị đánh giá khi khách hàng xem danh mục.
10. **Database (Postgres/MySQL)**
    * Lưu trữ dữ liệu của toàn hệ thống:
      + Người dùng, quyền truy cập.
      + Danh mục sản phẩm.
      + Giỏ hàng, đơn hàng.
      + Thanh toán, tồn kho, đánh giá.
11. **External Payment Gateway (VNPay/PayPal)**
    * Hệ thống thanh toán ngoài.
    * Payment Service gọi API đến đây để thực hiện giao dịch.

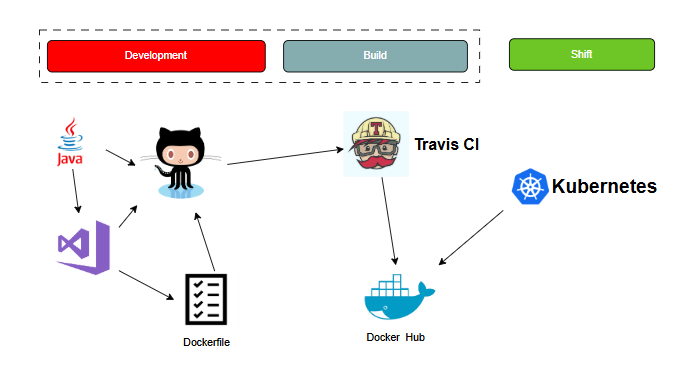
**Luồng hoạt động chính**

1. **User đăng nhập**
   * User → Ingress Gateway → Access Control Service.
   * Xác thực thành công → redirect:
     + Khách hàng → Catalog Service (trang sản phẩm).
     + Admin → Catalog Service (Admin Panel).
2. **Duyệt sản phẩm**
   * Catalog Service lấy dữ liệu từ Database.
   * User có thể xem thông tin tồn kho (Inventory Service cung cấp).
3. **Thêm sản phẩm vào giỏ**
   * User thao tác → Cart Service → lưu giỏ hàng vào Database.
4. **Đặt hàng**
   * Cart Service → Order Service (tạo đơn hàng).
   * Order Service → Payment Service (yêu cầu thanh toán).
   * Payment Service → External Payment Gateway (VNPay/PayPal).
   * Nếu thanh toán thành công → lưu trạng thái vào Database, cập nhật tồn kho.
5. **Đánh giá sản phẩm**
   * Sau khi mua hàng → User gửi đánh giá → Rating Service → lưu vào Database.

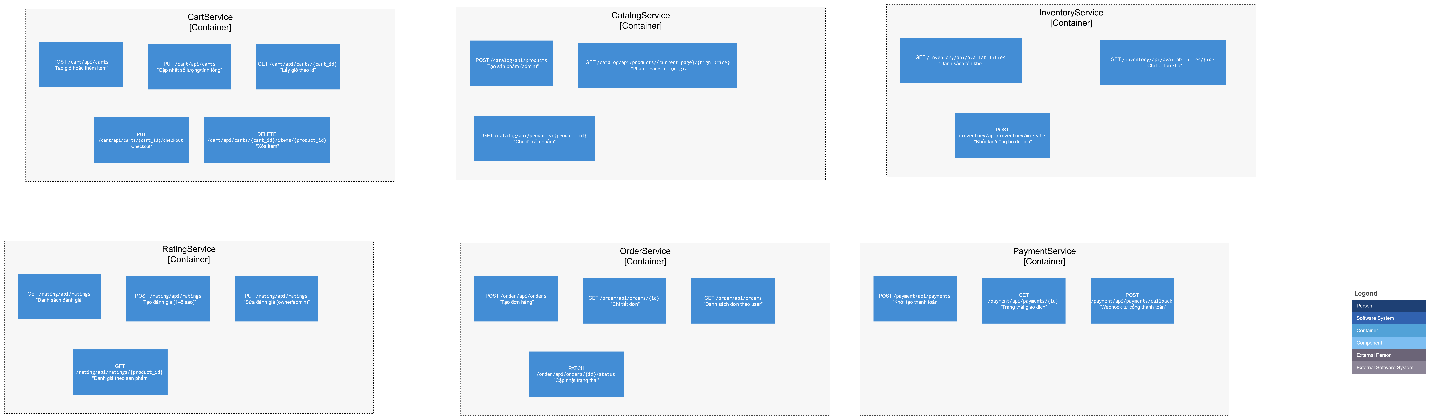
## 1.15. Sơ đồ kiến trúc SPA (Single Page Application):



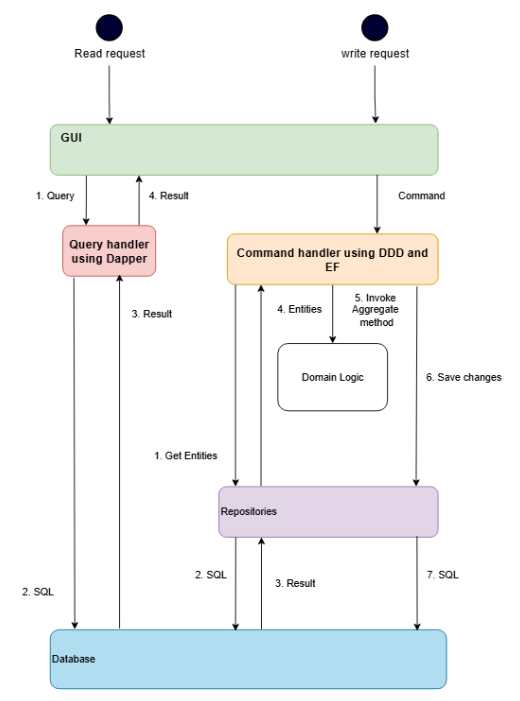
## 1.16. Sơ đồ triển khai CI/CD:



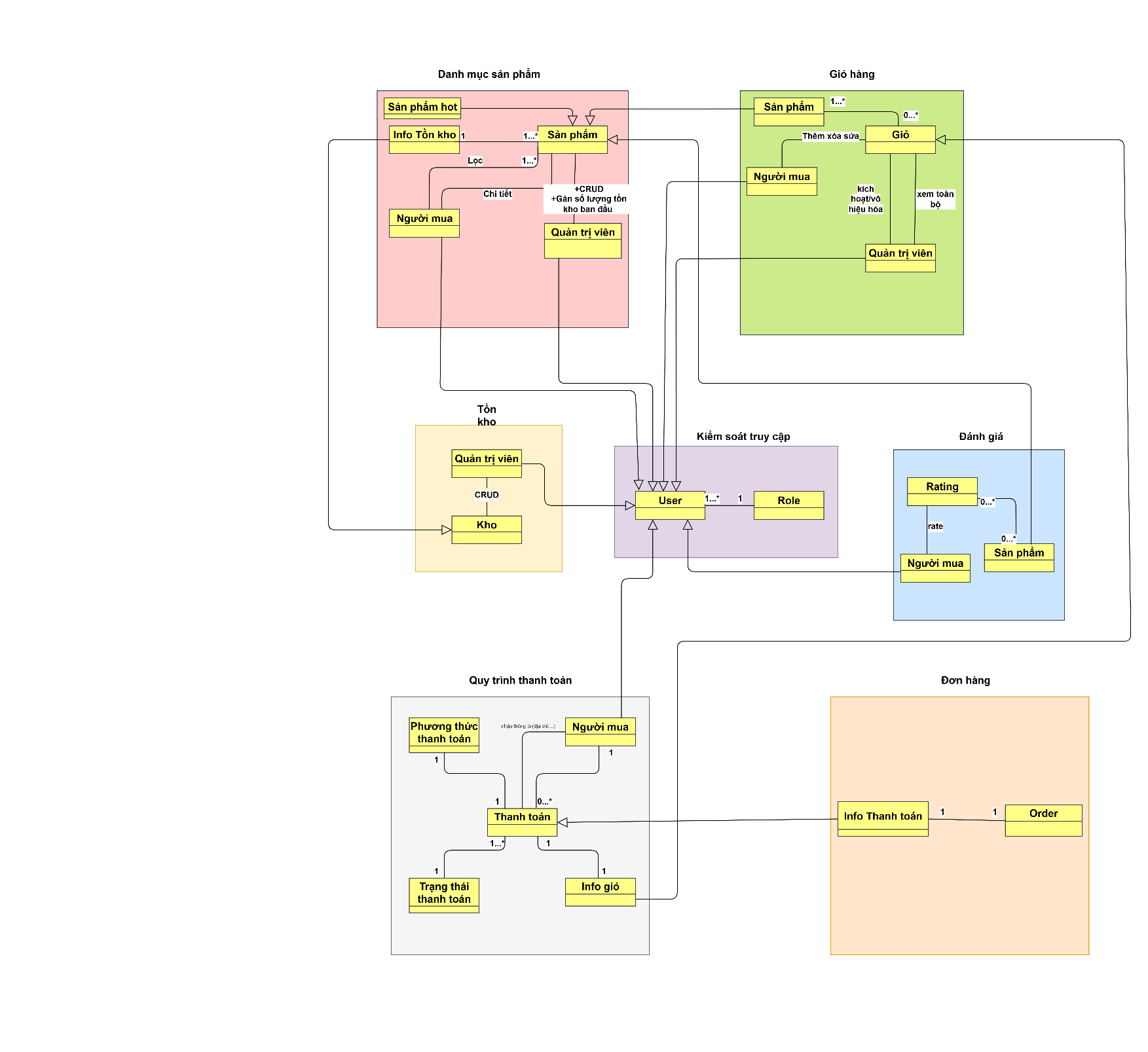
## 1.17. Sơ đồ API của hệ thống:



## 1.18. sơ đồ xử lý 1 request:



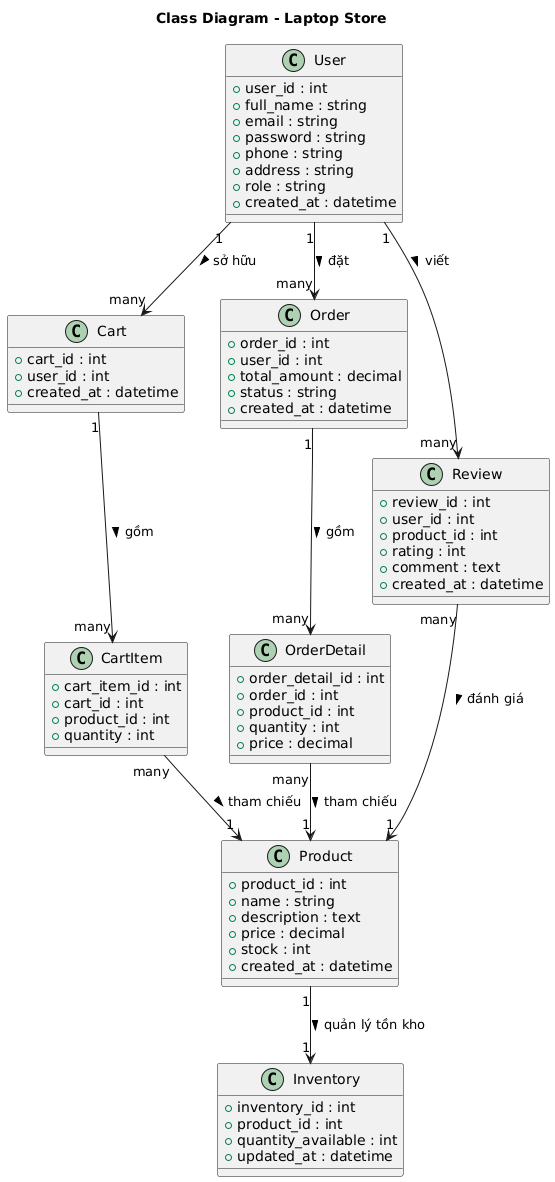
## 1.19. Mô hình khái niệm



## 1.20. Business Process Diagram

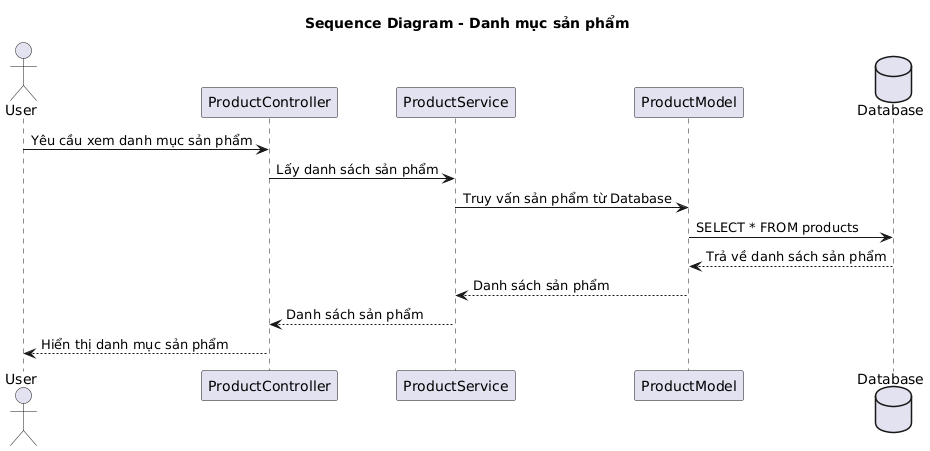


## 1.21. Class Diagram

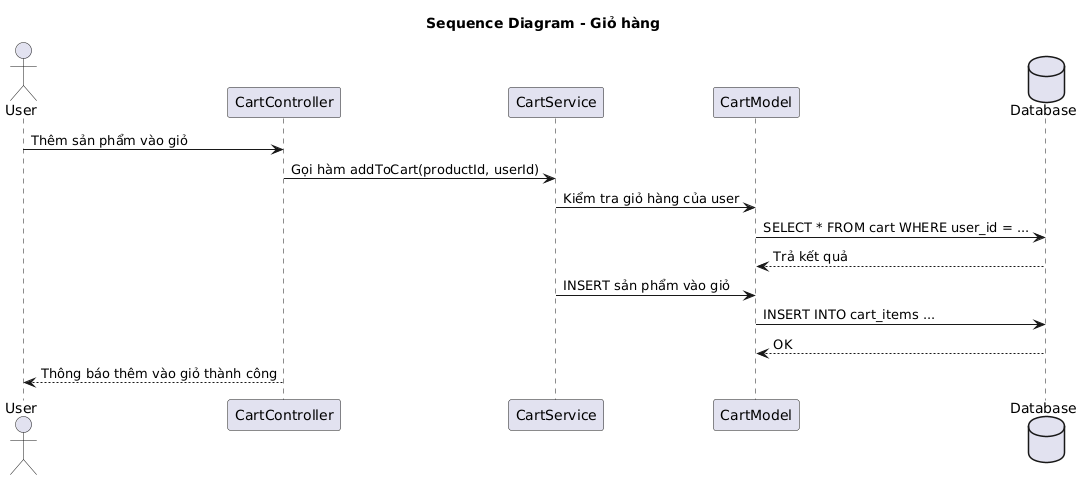


## 1.22. Squence

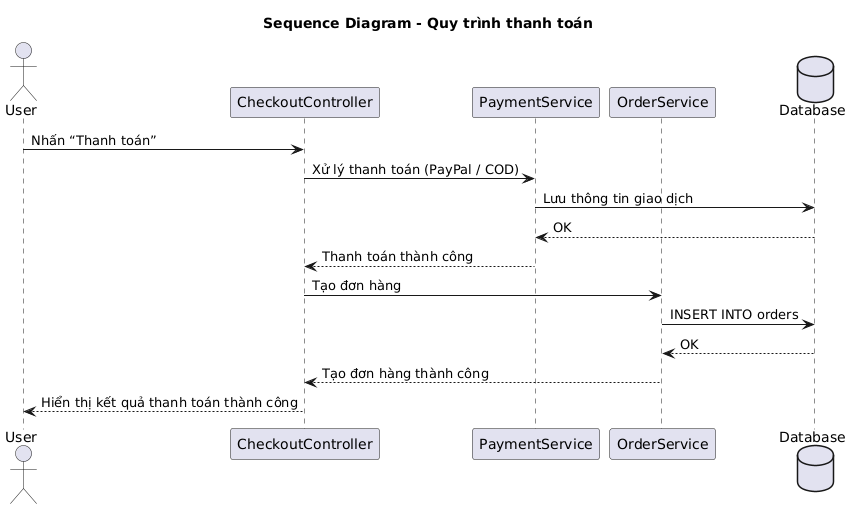
### 1.22.1. Danh mục sản phẩm



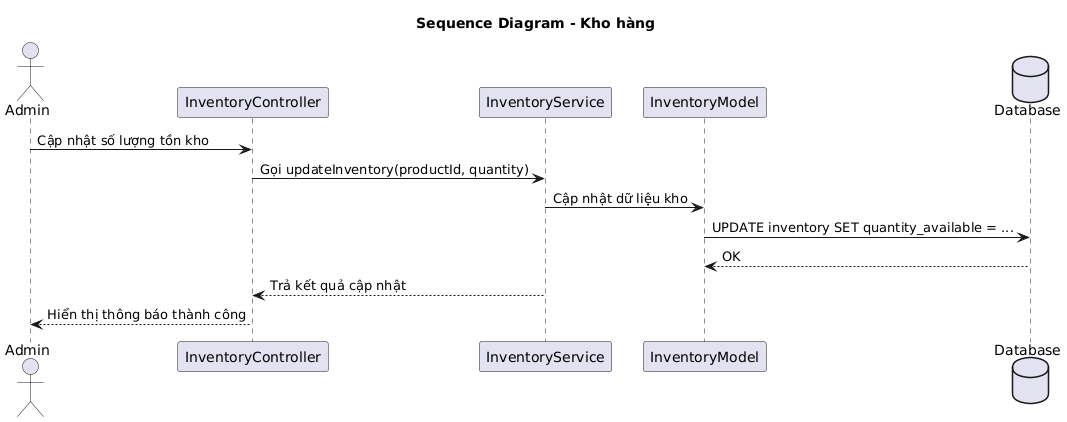
### 1.22.2. Giỏ hàng



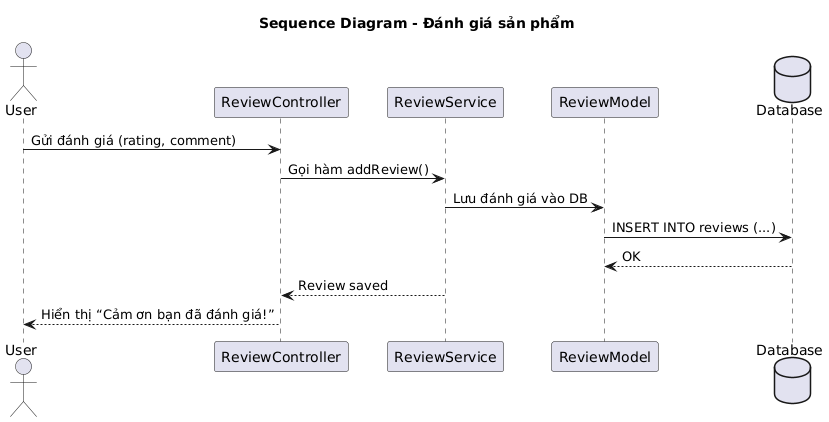
### 1.22.3. Quy trình thanh toán



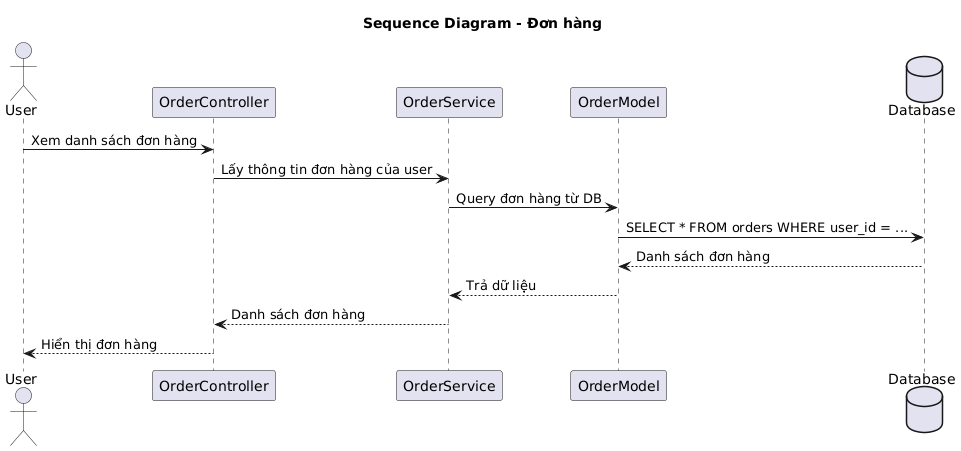
### 1.22.4. Kho hàng



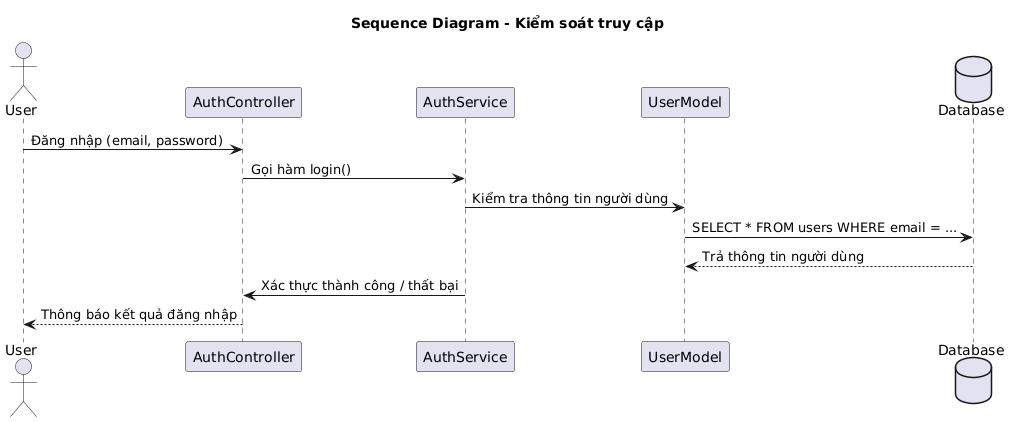
### 1.22.5. Đánh giá



### 1.22.6. Đơn hàng



### 1.22.7. Kiểm soát truy cập



# Chương 2. Kế hoạch kiểm thử

## 2.1. Giới thiệu

### 2.1.1 Mục đích

Tài liệu Test Plan này được xây dựng nhằm mô tả chi tiết kế hoạch kiểm thử cho hệ thống E-commerce Website Laptop Store.

Mục tiêu của kế hoạch là xác định phạm vi kiểm thử, chiến lược kiểm thử, các nguồn lực, lịch trình và tiêu chí để đánh giá việc kiểm thử đạt yêu cầu.

Việc kiểm thử nhằm đảm bảo rằng các chức năng chính của hệ thống hoạt động đúng như yêu cầu, ổn định, thân thiện với người dùng và đáp ứng mục tiêu nghiệp vụ của dự án.

### 2.1.2 Bối cảnh

Dự án E-commerce Website Laptop Store là một nền tảng thương mại điện tử cho phép người dùng (Khách hàng) mua máy tính và linh kiện trực tuyến, đồng thời cho phép Quản trị viên (Admin) quản lý hệ thống sản phẩm, kho hàng, đơn hàng và người dùng.  
Hệ thống gồm nhiều module nghiệp vụ chính như:

* **Danh mục sản phẩm:** hiển thị, lọc và xem chi tiết sản phẩm.
* **Giỏ hàng:** thêm, sửa, xóa sản phẩm và hiển thị tổng chi phí.
* **Thanh toán:** xử lý đơn hàng và gửi xác nhận cho khách hàng.
* **Kho hàng:** quản lý kho, số lượng tồn và vị trí lưu trữ.
* **Đánh giá:** khách hàng đánh giá sản phẩm đã mua.
* **Kiểm soát truy cập:** đăng nhập, đăng ký, phân quyền người dùng.
* **Đơn hàng:** lưu trữ và theo dõi đơn hàng được tạo sau khi thanh toán.

### 2.1.3 Phạm vi

Phạm vi kiểm thử bao gồm toàn bộ các module chức năng chính của hệ thống:

* **Danh mục sản phẩm:** kiểm thử chức năng lọc, tìm kiếm, sắp xếp và xem chi tiết sản phẩm.
* **Giỏ hàng:** kiểm thử thêm, sửa, xóa sản phẩm trong giỏ và cập nhật tổng tiền.
* **Thanh toán:** kiểm thử quá trình đặt hàng, xử lý thanh toán, xác nhận đơn hàng.
* **Kho hàng:** kiểm thử CRUD kho, gán số lượng tồn kho cho sản phẩm.
* **Đánh giá:** kiểm thử tạo và hiển thị đánh giá sản phẩm.
* **Kiểm soát truy cập:** kiểm thử đăng ký, đăng nhập, đổi mật khẩu và phân quyền.
* **Đơn hàng:** kiểm thử quản lý đơn hàng, cập nhật trạng thái và xem lịch sử.

Ngoài ra, phạm vi còn bao gồm kiểm thử **hiệu năng cơ bản, giao diện người dùng (UI) và tính khả dụng (usability).**Các chức năng không nằm trong phạm vi (ví dụ: tích hợp với hệ thống bên thứ ba hoặc module báo cáo nâng cao) sẽ được liệt kê ở phần “*Feature not to be tested*”.

### 2.1.4 Thông tin nhận diện dự án

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Document**  **(and version /date)** | **Created or Available** | **Received or Reviewed** | **Author or**  **Resource** | **Notes** |
| Requirements Specification | **** Yes  No | **** Yes  No | Business Analyst | Định nghĩa yêu cầu hệ thống |
| Functional Specification | **** Yes  No |  Yes **** No | System Analyst | Đang cập nhật |
| Use-Case Reports | **** Yes  No | **** Yes  No | BA & QA | Dùng làm cơ sở viết test case |
| Project Plan | **** Yes  No | **** Yes  No | Project Manager | Lịch trình phát triển |
| Design Specifications | **** Yes  No |  Yes **** No | Architect | Chưa hoàn tất |
| Prototype |  Yes No |  Yes No | Dev Team | Đã dùng để đánh giá UI |
| User’s Manuals |  Yes  No |  Yes  No |  | Sẽ bổ sung ở bản RC |
| Business Model or Flow |  Yes No |  Yes No | Business Analyst | Xác định luồng nghiệp vụ |
| Data Model or Flow |  Yes No |  Yes No | DB Engineer | Dùng cho test database |
| Business Functions and Rules |  Yes No |  Yes No | Business Analyst | Các quy tắc và chức năng nghiệp vụ liên quan |
| Project or Business Risk  Assessment |  Yes No |  Yes No | PM & QA Lead | Liệt kê rủi ro chính |

### 2.1.5 References

|  |  |  |
| --- | --- | --- |
| **No.** | **Document** | **Description** |
| 1 | ABC project Software Requirements Specification, Version 1.0.3.3, Monday, 25/Aug/10 | The **SRS** fully describes the external behavior of the application or subsystem identified. It also describes non-functional requirements, design constraints, and other factors necessary to provide a complete and comprehensive description of the software requirements. |
| 2 | ABC project Software Development Plan, Version 1.0.5, 05/Sep/10 | The purpose of Software Development Plan is to define tasks of ABC project (hereinafter referred to as IM or the project) to be developed. It focuses on the capabilities sought after by stakeholders, the target users in particular, and the reasons for this search. |
| 3 | IEEE 829 Standard for Software Test Documentation | International standard for writing software testing documentation. |

## 2.2 Các tính năng cần kiểm thử

### 2.2.1 Chức năng

**Module Kiểm soát truy cập**

**Đăng ký tài khoản**

* **Mục tiêu kiểm thử:** Đảm bảo người dùng có thể tạo tài khoản hợp lệ.
* **Điều kiện đầu vào:**
  + Người dùng truy cập trang “Đăng ký”.
  + Nhập thông tin: Họ tên, email, mật khẩu, xác nhận mật khẩu.
* **Các bước kiểm thử:**
  1. Nhập thông tin hợp lệ → Nhấn “Đăng ký”.
  2. Nhập email đã tồn tại → Nhấn “Đăng ký”.
  3. Nhập mật khẩu không đạt yêu cầu (ngắn, thiếu ký tự đặc biệt).
  4. Để trống một trường bắt buộc.
* **Kết quả mong đợi:**
  + TH1: Hiển thị thông báo “Đăng ký thành công”, dữ liệu lưu vào CSDL.
  + TH2: Hiển thị lỗi “Email đã tồn tại”.
  + TH3: Hiển thị lỗi “Mật khẩu không hợp lệ”.
  + TH4: Hiển thị lỗi “Vui lòng nhập đầy đủ thông tin”.

**Đăng nhập**

* **Mục tiêu kiểm thử:** Đảm bảo người dùng có thể đăng nhập với thông tin chính xác và được phân hướng đúng trang (User hoặc Admin).
* **Điều kiện đầu vào:**
  + Người dùng truy cập trang “Đăng nhập”.
  + Nhập email và mật khẩu.
* **Các bước kiểm thử:**
  1. Nhập thông tin hợp lệ → Nhấn “Đăng nhập”.
  2. Nhập sai mật khẩu → Nhấn “Đăng nhập”.
  3. Nhập email không tồn tại → Nhấn “Đăng nhập”.
  4. Để trống email hoặc mật khẩu.
* **Kết quả mong đợi:**
  + TH1: Đăng nhập thành công, điều hướng đến trang chủ tương ứng.
  + TH2: Hiển thị lỗi “Email hoặc mật khẩu không đúng”.
  + TH3: Hiển thị lỗi “Email chưa được đăng ký”.
  + TH4: Hiển thị cảnh báo “Vui lòng nhập thông tin đăng nhập”.

**Đổi mật khẩu**

* **Mục tiêu kiểm thử:** Đảm bảo chức năng đổi mật khẩu hoạt động đúng và bảo mật.
* **Điều kiện đầu vào:** Người dùng đã đăng nhập.
* **Các bước kiểm thử:**

1. Nhập mật khẩu cũ chính xác và mật khẩu mới hợp lệ → Nhấn “Xác nhận”.
2. Nhập sai mật khẩu cũ.
3. Nhập mật khẩu mới không đủ điều kiện (ngắn hơn 8 ký tự, không có ký tự đặc biệt).
4. Mật khẩu mới trùng mật khẩu cũ.

* **Kết quả mong đợi:**
* TH1: Hiển thị “Đổi mật khẩu thành công”.
* TH2: Hiển thị lỗi “Mật khẩu cũ không chính xác”.
* TH3: Hiển thị lỗi “Mật khẩu mới không hợp lệ”.
* TH4: Hiển thị “Mật khẩu mới không được trùng với mật khẩu cũ”.

**Đăng xuất**

**Mục tiêu kiểm thử:** Đảm bảo người dùng có thể đăng xuất và kết thúc phiên làm việc.

**Điều kiện đầu vào:** Người dùng đã đăng nhập.

* **Các bước kiểm thử:**

1. Nhấn nút “Đăng xuất”.
2. Thử truy cập lại trang yêu cầu đăng nhập (ví dụ: giỏ hàng, quản lý sản phẩm).

* **Kết quả mong đợi:**
* Hệ thống xóa session hiện tại.
* Chuyển hướng về trang “Đăng nhập”.
* Khi truy cập lại trang cần đăng nhập → hiển thị yêu cầu đăng nhập lại.

**Module Danh mục sản phẩm**

**Xem danh sách sản phẩm**

**Mục tiêu kiểm thử:** Đảm bảo người dùng có thể xem được danh sách sản phẩm đầy đủ cùng các thông tin cơ bản như tên, giá, hình ảnh và tình trạng tồn kho.

**Điều kiện đầu vào:**

* Người dùng truy cập trang “Danh mục sản phẩm”.
* CSDL có sẵn dữ liệu sản phẩm.

**Các bước kiểm thử:**

1. Truy cập vào trang danh mục sản phẩm.
2. Xác minh rằng hệ thống hiển thị danh sách sản phẩm.
3. Quan sát thông tin mỗi sản phẩm: tên, giá, hình ảnh, tình trạng tồn kho.
4. Cuộn trang để kiểm tra phân trang (pagination) hoặc tải thêm sản phẩm.

**Kết quả mong đợi:**

* Hệ thống hiển thị đầy đủ danh sách sản phẩm.
* Mỗi sản phẩm hiển thị đúng thông tin trong CSDL.
* Khi có nhiều sản phẩm, hệ thống phân trang hoặc tải thêm hợp lý, không lỗi hiển thị.

**Tìm kiếm sản phẩm**

**Mục tiêu kiểm thử:** Đảm bảo người dùng có thể tìm kiếm sản phẩm theo từ khóa và kết quả trả về chính xác.

**Điều kiện đầu vào:**

* Trang danh mục sản phẩm đang hiển thị.
* Thanh tìm kiếm khả dụng.

**Các bước kiểm thử:**

1. Nhập từ khóa hợp lệ (ví dụ: “Dell XPS”).
2. Nhấn nút “Tìm kiếm”.
3. Nhập từ khóa không tồn tại trong hệ thống.
4. Để trống ô tìm kiếm rồi nhấn “Tìm”.

**Kết quả mong đợi:**

* TH1: Hiển thị sản phẩm có chứa từ khóa trong tên hoặc mô tả.
* TH2: Hiển thị thông báo “Không tìm thấy sản phẩm phù hợp”.
* TH3: Khi không nhập từ khóa, hiển thị toàn bộ danh sách sản phẩm hoặc cảnh báo “Vui lòng nhập từ khóa tìm kiếm”.

**Lọc và sắp xếp sản phẩm**

**Mục tiêu kiểm thử:** Xác minh hệ thống có thể lọc và sắp xếp sản phẩm theo giá, tên hoặc cấu hình.

**Điều kiện đầu vào:** Có danh sách sản phẩm với các mức giá, tên và cấu hình khác nhau.

**Các bước kiểm thử:**

1. Chọn lọc theo giá (từ thấp đến cao / cao đến thấp).
2. Chọn lọc theo cấu hình (RAM, CPU, thương hiệu).
3. Chọn sắp xếp theo tên (A → Z / Z → A).
4. Kết hợp nhiều điều kiện lọc và sắp xếp cùng lúc.

**Kết quả mong đợi:**

* Kết quả hiển thị đúng theo tiêu chí lọc/sắp xếp.
* Dữ liệu hiển thị không bị trùng hoặc mất sản phẩm.
* Tốc độ xử lý và cập nhật danh sách ổn định, không làm treo trang.

**Xem chi tiết sản phẩm**

**Mục tiêu kiểm thử:** Đảm bảo người dùng có thể xem thông tin chi tiết chính xác của sản phẩm được chọn.

**Điều kiện đầu vào:** Người dùng đang ở trang danh sách sản phẩm.

**Các bước kiểm thử:**

1. Nhấn vào một sản phẩm trong danh sách.
2. Quan sát các thông tin: tên, giá, mô tả, hình ảnh, tồn kho, đánh giá.
3. Kiểm tra sản phẩm “Hot” có hiển thị nhãn ưu đãi.

**Kết quả mong đợi:**

* Hiển thị đúng thông tin chi tiết của sản phẩm.
* Hình ảnh và mô tả rõ ràng, không lỗi hiển thị.
* Sản phẩm “Hot” có gắn nhãn “Hot” hoặc “Ưu đãi đặc biệt”.

**Hiển thị sản phẩm Hot**

**Mục tiêu kiểm thử:** Đảm bảo hệ thống hiển thị danh sách sản phẩm “Hot” đúng theo dữ liệu được đánh dấu trong hệ thống.

**Điều kiện đầu vào:** Một số sản phẩm trong CSDL được gắn cờ “HotProduct = true”.

**Các bước kiểm thử:**

1. Truy cập trang chủ hoặc danh mục sản phẩm.
2. Quan sát khu vực hiển thị “Sản phẩm nổi bật” hoặc “Sản phẩm hot”.

**Kết quả mong đợi:**

* Chỉ hiển thị sản phẩm được đánh dấu “Hot”.
* Thông tin sản phẩm hiển thị chính xác.
* Sản phẩm thường không bị lẫn vào danh sách “Hot

**Module Giỏ hàng**

**Thêm sản phẩm vào giỏ hàng**

**Mục tiêu kiểm thử:** Đảm bảo người dùng có thể thêm sản phẩm hợp lệ vào giỏ hàng từ trang danh sách hoặc trang chi tiết sản phẩm.

**Điều kiện đầu vào:**

* Người dùng đã đăng nhập.
* Sản phẩm còn hàng trong kho.

**Các bước kiểm thử:**

1. Từ trang danh sách, nhấn nút “Thêm vào giỏ” ở một sản phẩm bất kỳ.
2. Từ trang chi tiết, chọn số lượng → nhấn “Thêm vào giỏ”.
3. Thử thêm cùng sản phẩm nhiều lần.
4. Thử thêm sản phẩm đã hết hàng.

**Kết quả mong đợi:**

* TH1: Hiển thị thông báo “Đã thêm vào giỏ hàng”, giỏ hàng được cập nhật.
* TH2: Nếu sản phẩm đã có trong giỏ, hệ thống cập nhật số lượng thay vì tạo dòng mới.
* TH3: Nếu sản phẩm hết hàng, hiển thị “Sản phẩm tạm hết hàng” và không thêm vào giỏ.
* TH4: Tổng tiền trong giỏ thay đổi tương ứng với sản phẩm vừa thêm.

**Xem giỏ hàng**

**Mục tiêu kiểm thử:** Xác minh rằng người dùng có thể xem giỏ hàng với thông tin sản phẩm chính xác và tổng hợp đầy đủ chi phí.

**Điều kiện đầu vào:** Người dùng đã thêm ít nhất một sản phẩm vào giỏ.

**Các bước kiểm thử:**

1. Truy cập trang “Giỏ hàng”.
2. Quan sát danh sách sản phẩm hiển thị.
3. Kiểm tra các thông tin: tên sản phẩm, số lượng, giá, tổng phụ, phí vận chuyển, tổng cộng.
4. Thay đổi số lượng để kiểm tra cập nhật tổng giá trị.

**Kết quả mong đợi:**

* Danh sách hiển thị đầy đủ thông tin chính xác.
* Khi thay đổi số lượng, tổng chi phí tự động cập nhật.
* Nếu giỏ hàng trống, hiển thị thông báo “Giỏ hàng của bạn đang trống”.

**Sửa số lượng hoặc xóa sản phẩm trong giỏ**

**Mục tiêu kiểm thử:** Đảm bảo người dùng có thể chỉnh sửa số lượng hoặc xóa sản phẩm khỏi giỏ hàng.

**Điều kiện đầu vào:** Có sản phẩm trong giỏ hàng.

**Các bước kiểm thử:**

1. Chọn sản phẩm trong giỏ → chỉnh sửa số lượng.
2. Giảm số lượng về 0 → kiểm tra hành vi của hệ thống.
3. Nhấn nút “Xóa” ở một sản phẩm.

**Kết quả mong đợi:**

* Khi chỉnh sửa số lượng, tổng giá trị giỏ cập nhật ngay.
* Nếu số lượng = 0 → sản phẩm bị xóa khỏi giỏ.
* Khi xóa sản phẩm → hiển thị thông báo “Đã xóa sản phẩm khỏi giỏ hàng”.
* Giỏ trống thì không hiển thị bảng tóm tắt chi phí.

**Tự động cập nhật giỏ hàng**

**Mục tiêu kiểm thử:** Xác minh hệ thống tự động cập nhật thông tin giỏ khi người dùng thay đổi dữ liệu (thêm, sửa, xóa).

**Điều kiện đầu vào:** Người dùng đang ở trang “Giỏ hàng”.

**Các bước kiểm thử:**

1. Thêm sản phẩm mới trong tab khác → quay lại trang giỏ hàng hiện tại.
2. Thay đổi số lượng → kiểm tra cập nhật tự động.
3. Làm mới (reload) trang giỏ hàng.

**Kết quả mong đợi:**

* Thông tin giỏ luôn đồng bộ với dữ liệu hệ thống.
* Không cần tải lại trang để thấy thay đổi.
* Reload không làm mất dữ liệu giỏ.

**Quản lý giỏ hàng của người dùng**

**Mục tiêu kiểm thử:** Đảm bảo quản trị viên có thể xem và quản lý các giỏ hàng của người dùng.

**Điều kiện đầu vào:**

* Admin đã đăng nhập.
* Có nhiều giỏ hàng từ các người dùng khác nhau trong hệ thống.

**Các bước kiểm thử:**

1. Truy cập trang “Quản lý giỏ hàng” của admin.
2. Chọn xem chi tiết giỏ hàng của người dùng bất kỳ.
3. Thử bật/tắt giỏ hàng (kích hoạt / vô hiệu hóa).

**Kết quả mong đợi:**

* Admin xem được danh sách giỏ hàng và người dùng tương ứng.
* Thao tác bật/tắt giỏ hàng hiển thị thông báo thành công.
* Giỏ hàng bị vô hiệu hóa không thể tiếp tục sử dụng bởi người dùng.

**Module Quy trình Thanh toán**

**Thực hiện thanh toán**

**Mục tiêu kiểm thử:** Đảm bảo hệ thống xử lý thanh toán chính xác, xác nhận thông tin sản phẩm, người mua và tổng số tiền cần trả.

**Điều kiện đầu vào:**

* Người dùng đã đăng nhập và có ít nhất một sản phẩm trong giỏ hàng.
* Giỏ hàng ở trạng thái “Chờ thanh toán”.

**Các bước kiểm thử:**

1. Mở trang giỏ hàng → nhấn nút “Thanh toán”.
2. Kiểm tra hệ thống hiển thị đúng tổng giá trị đơn hàng.
3. Nhập thông tin giao hàng (họ tên, địa chỉ, số điện thoại).
4. Nhấn nút “Xác nhận thanh toán”.
5. Quan sát phản hồi của hệ thống.

**Kết quả mong đợi:**

* Hiển thị trang tóm tắt đơn hàng đúng thông tin sản phẩm và tổng chi phí.
* Khi nhấn “Xác nhận thanh toán”, hệ thống xử lý và hiển thị “Đặt hàng thành công”.
* Dữ liệu đơn hàng được lưu vào cơ sở dữ liệu.
* Nếu thiếu thông tin giao hàng, hiển thị lỗi “Vui lòng nhập đầy đủ thông tin”.

**Chọn phương thức thanh toán**

**Mục tiêu kiểm thử:** Đảm bảo người dùng có thể chọn phương thức thanh toán hợp lệ và hệ thống xử lý đúng theo lựa chọn đó.

**Điều kiện đầu vào:**

* Trang thanh toán đang hiển thị.
* Có các phương thức thanh toán khả dụng trong hệ thống (COD, Thẻ tín dụng, Ví điện tử).

**Các bước kiểm thử:**

1. Chọn phương thức “Thanh toán khi nhận hàng (COD)” → nhấn xác nhận.
2. Chọn “Thẻ tín dụng” → nhập thông tin thẻ hợp lệ.
3. Nhập sai thông tin thẻ hoặc để trống trường bắt buộc.
4. Chọn “Ví điện tử” (VD: MoMo, ZaloPay) và xác minh kết nối.

**Kết quả mong đợi:**

* TH1: Với COD, hiển thị xác nhận đơn hàng ngay lập tức.
* TH2: Với Thẻ tín dụng, giao dịch được xác thực và hiển thị “Thanh toán thành công”.
* TH3: Nếu thông tin thẻ sai, hiển thị lỗi “Thông tin thanh toán không hợp lệ”.
* TH4: Với Ví điện tử, hiển thị giao diện xác thực và trả về kết quả hợp lệ.

**Xác nhận đơn hàng**

**Mục tiêu kiểm thử:** Đảm bảo hệ thống tạo đơn hàng hợp lệ sau khi thanh toán thành công và gửi thông báo xác nhận cho người dùng.

**Điều kiện đầu vào:**

* Thanh toán đã hoàn tất.
* Có thông tin người mua, địa chỉ giao hàng và danh sách sản phẩm.

**Các bước kiểm thử:**

1. Sau khi thanh toán, hệ thống tự động tạo đơn hàng.
2. Kiểm tra email người mua để xác minh có thông báo “Xác nhận đơn hàng”.
3. Truy cập mục “Lịch sử đơn hàng” để kiểm tra đơn mới tạo.

**Kết quả mong đợi:**

* Hệ thống tạo bản ghi đơn hàng với trạng thái “Đang xử lý”.
* Email xác nhận gửi đến người mua với đầy đủ thông tin: mã đơn, danh sách sản phẩm, tổng tiền, địa chỉ giao hàng.
* Đơn hàng xuất hiện trong danh sách “Lịch sử đơn hàng”.

**Xử lý lỗi thanh toán**

**Mục tiêu kiểm thử:** Đảm bảo hệ thống phát hiện và xử lý đúng các tình huống thanh toán không thành công.

**Điều kiện đầu vào:** Người dùng đang ở bước xác nhận thanh toán.

**Các bước kiểm thử:**

1. Nhập thông tin thẻ sai (số thẻ hoặc mã bảo mật).
2. Mất kết nối mạng trong quá trình xử lý.
3. Hệ thống thanh toán trả về lỗi “Thanh toán thất bại”.
4. Hủy giao dịch giữa chừng.

**Kết quả mong đợi:**

* Hiển thị thông báo “Thanh toán thất bại” kèm lý do cụ thể.
* Giỏ hàng vẫn giữ nguyên, không bị xóa sản phẩm.
* Không tạo đơn hàng mới trong hệ thống.
* Người dùng có thể thử lại thanh toán sau.

**Module Kho hàng**

**Tạo kho mới**

**Mục tiêu kiểm thử:** Đảm bảo quản trị viên có thể tạo kho mới với thông tin hợp lệ (tên, địa điểm, tổng số lượng ban đầu).

**Điều kiện đầu vào:**

* Quản trị viên (Admin) đã đăng nhập.
* Có quyền truy cập trang “Quản lý kho hàng”.

**Các bước kiểm thử:**

1. Truy cập trang “Kho hàng”.
2. Nhấn nút “Thêm kho mới”.
3. Nhập thông tin: Tên kho, địa điểm, tổng số lượng ban đầu.
4. Nhấn “Lưu”.
5. Thử thêm kho trùng tên hoặc thiếu thông tin bắt buộc.

**Kết quả mong đợi:**

* TH1: Hiển thị thông báo “Thêm kho thành công”, kho mới được hiển thị trong danh sách.
* TH2: Nếu để trống trường bắt buộc → hiển thị lỗi “Vui lòng nhập đầy đủ thông tin”.
* TH3: Nếu tên kho bị trùng → hiển thị lỗi “Tên kho đã tồn tại”.

**Xem thông tin kho**

**Mục tiêu kiểm thử:** Đảm bảo quản trị viên có thể xem đầy đủ thông tin của từng kho hàng.

**Điều kiện đầu vào:** Có ít nhất một kho hàng được tạo trong hệ thống.

**Các bước kiểm thử:**

1. Truy cập danh sách kho hàng.
2. Chọn kho bất kỳ → xem thông tin chi tiết.
3. Quan sát thông tin hiển thị (tên kho, địa điểm, tổng số lượng, ngày cập nhật gần nhất).

**Kết quả mong đợi:**

* Thông tin hiển thị đúng và đầy đủ.
* Không xảy ra lỗi hiển thị khi dữ liệu có ký tự đặc biệt (ví dụ tên kho tiếng Việt có dấu).

**Cập nhật kho**

**Mục tiêu kiểm thử:** Đảm bảo chức năng cập nhật thông tin kho hoạt động chính xác và dữ liệu được lưu lại.

**Điều kiện đầu vào:**

* Quản trị viên đã đăng nhập.
* Có ít nhất một kho trong hệ thống.

**Các bước kiểm thử:**

1. Chọn một kho trong danh sách → nhấn “Sửa”.
2. Thay đổi thông tin (tên kho, địa điểm hoặc tổng số lượng).
3. Nhấn “Lưu thay đổi”.
4. Thử cập nhật với dữ liệu trống hoặc tổng số lượng âm.

**Kết quả mong đợi:**

* TH1: Thay đổi được lưu thành công → hiển thị “Cập nhật thành công”.
* TH2: Nếu dữ liệu không hợp lệ → hiển thị lỗi tương ứng.
* TH3: Dữ liệu hiển thị mới phản ánh chính xác sau khi tải lại trang.

**Xóa kho**

**Mục tiêu kiểm thử:** Đảm bảo quản trị viên chỉ có thể xóa kho khi kho đó không còn sản phẩm được liên kết.

**Điều kiện đầu vào:** Có ít nhất hai kho hàng (một kho trống, một kho có sản phẩm).

**Các bước kiểm thử:**

1. Chọn kho **trống** → nhấn “Xóa”.
2. Chọn kho **có sản phẩm liên kết** → nhấn “Xóa”.

**Kết quả mong đợi:**

* TH1: Với kho trống → hiển thị “Đã xóa thành công”.
* TH2: Với kho có sản phẩm liên kết → hiển thị lỗi “Không thể xóa kho đang chứa sản phẩm”.
* Sau khi xóa thành công, kho biến mất khỏi danh sách.

**Gán số lượng tồn kho cho sản phẩm**

**Mục tiêu kiểm thử:** Đảm bảo quản trị viên có thể gán đúng số lượng tồn kho cho sản phẩm và hệ thống cập nhật tự động.

**Điều kiện đầu vào:**

* Có ít nhất một kho và danh sách sản phẩm.
* Admin đã đăng nhập.

**Các bước kiểm thử:**

1. Truy cập trang “Quản lý kho”.
2. Chọn một kho → chọn “Gán số lượng tồn kho”.
3. Chọn sản phẩm → nhập số lượng tồn (VD: 20).
4. Nhấn “Lưu”.
5. Thử nhập số lượng âm hoặc ký tự không hợp lệ.

**Kết quả mong đợi:**

* TH1: Gán thành công → hiển thị “Cập nhật tồn kho thành công”.
* TH2: Nếu nhập số lượng âm hoặc ký tự → hiển thị “Số lượng không hợp lệ”.
* TH3: Khi cập nhật thành công, thông tin tồn kho của sản phẩm thay đổi tương ứng.

**Module Đánh giá sản phẩm**

**Tạo đánh giá sản phẩm**

**Mục tiêu kiểm thử:** Đảm bảo người dùng đã mua sản phẩm có thể tạo đánh giá mới, gồm số sao và bình luận.

**Điều kiện đầu vào:**

* Người dùng đã đăng nhập.
* Người dùng đã có đơn hàng hoàn tất chứa sản phẩm cần đánh giá.

**Các bước kiểm thử:**

1. Truy cập trang chi tiết sản phẩm.
2. Nhấn nút “Đánh giá sản phẩm”.
3. Nhập số sao (1–5) và bình luận.
4. Nhấn “Gửi đánh giá”.
5. Thử gửi khi để trống nội dung hoặc chưa chọn số sao.

**Kết quả mong đợi:**

* TH1: Hiển thị thông báo “Cảm ơn bạn đã đánh giá sản phẩm”.
* TH2: Đánh giá mới được lưu và hiển thị trong danh sách đánh giá.
* TH3: Nếu để trống thông tin → hiển thị lỗi “Vui lòng nhập đầy đủ thông tin đánh giá”.

**Nhập số sao và bình luận**

**Mục tiêu kiểm thử:** Đảm bảo hệ thống ghi nhận chính xác số sao và nội dung bình luận người dùng nhập.

**Điều kiện đầu vào:** Người dùng đã truy cập giao diện “Đánh giá sản phẩm”.

**Các bước kiểm thử:**

1. Chọn số sao 1, 3, 5 → gửi đánh giá.
2. Nhập bình luận ngắn (<10 ký tự).
3. Nhập bình luận dài (>200 ký tự).
4. Dán nội dung chứa ký tự đặc biệt (VD: emoji, ký tự tiếng Việt có dấu).

**Kết quả mong đợi:**

* TH1: Hệ thống lưu đúng số sao (1–5).
* TH2: Chấp nhận bình luận hợp lệ, cắt gọn nếu quá dài.
* TH3: Hiển thị đúng tiếng Việt, không lỗi mã hóa.
* TH4: Nếu vượt giới hạn ký tự → hiển thị cảnh báo “Nội dung quá dài”.

**Hiển thị đánh giá sản phẩm**

**Mục tiêu kiểm thử:** Xác minh rằng các đánh giá được hiển thị chính xác trên trang chi tiết sản phẩm, theo đúng thứ tự thời gian.

**Điều kiện đầu vào:** Sản phẩm đã có ít nhất một đánh giá được tạo.

**Các bước kiểm thử:**

1. Truy cập trang chi tiết sản phẩm có đánh giá.
2. Kiểm tra danh sách đánh giá hiển thị (số sao, bình luận, tên người đánh giá, ngày đăng).
3. Tạo thêm một đánh giá mới → làm mới trang.

**Kết quả mong đợi:**

* Các đánh giá hiển thị đầy đủ thông tin.
* Thứ tự hiển thị theo thời gian (mới nhất ở trên).
* Số sao trung bình của sản phẩm cập nhật đúng.

**Kiểm soát quyền đánh giá**

**Mục tiêu kiểm thử:** Đảm bảo chỉ người dùng đã mua sản phẩm mới có quyền đánh giá.

**Điều kiện đầu vào:**

* Một người dùng **đã mua** sản phẩm.
* Một người dùng **chưa từng mua** sản phẩm.

**Các bước kiểm thử:**

1. Dùng tài khoản đã mua → mở sản phẩm → chọn “Đánh giá”.
2. Dùng tài khoản chưa mua → thử thực hiện tương tự.

**Kết quả mong đợi:**

* Người dùng đã mua: có thể nhập và gửi đánh giá.
* Người dùng chưa mua: hiển thị thông báo “Bạn chỉ có thể đánh giá sau khi mua sản phẩm này”.
* Hệ thống ngăn chặn gửi đánh giá trái phép (không bypass được qua URL hoặc API).

**Module Đơn hàng**

**Tạo đơn hàng**

**Mục tiêu kiểm thử:** Đảm bảo đơn hàng được tạo tự động sau khi thanh toán thành công.

**Điều kiện đầu vào:**

* Người dùng đã chọn sản phẩm và thực hiện thanh toán.
* Tài khoản người dùng đã đăng nhập.

**Các bước kiểm thử:**

1. Thêm sản phẩm vào giỏ hàng.
2. Thực hiện thanh toán với thông tin hợp lệ.
3. Kiểm tra danh sách đơn hàng của người dùng.

**Kết quả mong đợi:**

* Đơn hàng được tạo tự động sau khi thanh toán thành công.
* Thông tin đơn hàng (sản phẩm, số lượng, giá, người mua) hiển thị chính xác.

**Cập nhật trạng thái đơn hàng**

**Mục tiêu kiểm thử:** Đảm bảo chức năng thay đổi trạng thái đơn hàng hoạt động chính xác.

**Điều kiện đầu vào:**

* Đơn hàng đã được tạo.
* Người dùng hoặc admin có quyền thay đổi trạng thái.

**Các bước kiểm thử:**

1. Chọn đơn hàng cần cập nhật.
2. Thay đổi trạng thái sang “Đang xử lý”.
3. Thay đổi trạng thái sang “Đã giao”.
4. Thay đổi trạng thái sang “Hủy”.

**Kết quả mong đợi:**

* TH1: Trạng thái hiển thị “Đang xử lý”.
* TH2: Trạng thái hiển thị “Đã giao”.
* TH3: Trạng thái hiển thị “Hủy”.
* Mỗi trạng thái phản ánh đúng tiến trình thực tế của đơn hàng.

**Xem danh sách đơn hàng**

**Mục tiêu kiểm thử:** Đảm bảo danh sách đơn hàng hiển thị đầy đủ và chính xác thông tin cơ bản.

**Điều kiện đầu vào:** Người dùng đã tạo ít nhất 1 đơn hàng.

**Các bước kiểm thử:**

1. Truy cập trang danh sách đơn hàng.
2. Kiểm tra các thông tin: mã đơn hàng, sản phẩm, giá, trạng thái, ngày tạo, người mua.

**Kết quả mong đợi:**

* Danh sách đơn hàng hiển thị đúng thông tin từng đơn hàng.
* Thông tin người mua, sản phẩm và giá trùng khớp với dữ liệu thực tế.

**Xem chi tiết đơn hàng**

**Mục tiêu kiểm thử:** Đảm bảo thông tin chi tiết của đơn hàng trùng khớp với giao dịch thanh toán.

**Điều kiện đầu vào:** Người dùng có đơn hàng đã tạo.

**Các bước kiểm thử:**

1. Nhấp vào một đơn hàng trong danh sách.
2. Kiểm tra chi tiết đơn hàng: sản phẩm, số lượng, giá, địa chỉ giao hàng, thông tin thanh toán.

**Kết quả mong đợi:**

* Chi tiết đơn hàng hiển thị đầy đủ và chính xác.
* Thông tin trùng khớp với giao dịch thanh toán đã thực hiện.

### 2.2.2 Khả năng sử dụng

**Hỗ trợ phiên đăng nhập**

**Mục tiêu kiểm thử:** Đảm bảo trạng thái đăng nhập được duy trì trong suốt quá trình mua hàng.

**Điều kiện đầu vào:**

* Người dùng đã đăng nhập thành công.
* Người dùng đã thêm sản phẩm vào giỏ hàng.

**Các bước kiểm thử:**

1. Đăng nhập vào tài khoản.
2. Thêm sản phẩm vào giỏ hàng.
3. Điều hướng giữa các trang (sản phẩm, giỏ hàng, thanh toán).
4. Kiểm tra xem trạng thái đăng nhập có được duy trì hay không.

**Kết quả mong đợi:**

* Người dùng không bị đăng xuất khi chuyển trang.
* Giỏ hàng và thông tin cá nhân vẫn hiển thị chính xác.

**Khả năng hiển thị**

**Mục tiêu kiểm thử:** Đảm bảo bố cục, kích thước chữ, phông chữ và hình ảnh hiển thị đúng, dễ nhìn.

**Điều kiện đầu vào:** Truy cập website trên máy tính hoặc thiết bị di động.

**Các bước kiểm thử:**

1. Kiểm tra bố cục tổng thể của trang: header, footer, menu, nội dung.
2. Kiểm tra kích thước chữ và phông chữ cho từng phần: tiêu đề, mô tả, nút.
3. Kiểm tra hình ảnh: chất lượng, hiển thị đúng vị trí, không bị vỡ hình.

**Kết quả mong đợi:**

* Bố cục gọn gàng, dễ nhìn, các phần tử không bị chồng chéo.
* Kích thước chữ và phông chữ dễ đọc.
* Hình ảnh hiển thị đầy đủ, rõ nét, đúng vị trí.

**Tương tác người dùng**

**Mục tiêu kiểm thử:** Đảm bảo các nút, biểu mẫu và thông báo dễ hiểu, phản hồi nhanh.

**Điều kiện đầu vào:** Người dùng đã truy cập trang web và có ít nhất 1 sản phẩm trong giỏ hàng.

**Các bước kiểm thử:**

1. Nhấn các nút chính: Thêm vào giỏ hàng, Thanh toán, Đăng nhập, Đăng ký.
2. Điền biểu mẫu: Thông tin cá nhân, địa chỉ, thông tin thanh toán.
3. Kiểm tra thông báo: thành công, lỗi, cảnh báo.

**Kết quả mong đợi:**

* Nút và biểu mẫu hoạt động chính xác, phản hồi nhanh (<2 giây).
* Thông báo rõ ràng, dễ hiểu, hiển thị đúng ngữ cảnh.

**Khả năng hiển thị đa nền tảng**

**Mục tiêu kiểm thử:** Đảm bảo website hiển thị đúng và hoạt động ổn định trên các trình duyệt khác nhau.

**Điều kiện đầu vào:** Cài đặt các trình duyệt: Chrome, Edge, Firefox, Safari (nếu có).

**Các bước kiểm thử:**

1. Mở website trên từng trình duyệt.
2. Kiểm tra bố cục, hình ảnh, phông chữ, nút và biểu mẫu.
3. Thực hiện thao tác mua hàng từ chọn sản phẩm → thanh toán → xem đơn hàng.

**Kết quả mong đợi:**

* Website hiển thị đồng nhất trên tất cả trình duyệt.
* Các chức năng (thêm giỏ hàng, thanh toán, đăng nhập) hoạt động chính xác.

### 2.2.3 Ràng buộc thiết kế

**Giới hạn độ dài đầu vào**

**Mục tiêu kiểm thử:** Đảm bảo các trường văn bản không vượt quá giới hạn ký tự cho phép.

**Điều kiện đầu vào:** Truy cập các form nhập liệu: Email, Mật khẩu, Tên sản phẩm, Mô tả sản phẩm.

**Các bước kiểm thử:**

1. Nhập dữ liệu trong giới hạn ký tự cho phép → kiểm tra lưu thành công.
2. Nhập dữ liệu vượt giới hạn ký tự → kiểm tra phản hồi hệ thống.

**Kết quả mong đợi:**

* TH1: Dữ liệu hợp lệ được lưu thành công.
* TH2: Hệ thống hiển thị thông báo lỗi khi vượt giới hạn (ví dụ: “Vui lòng nhập tối đa 50 ký tự”).

**Môi trường kiểm thử**

**Mục tiêu kiểm thử:** Đảm bảo hệ thống hoạt động ổn định trên môi trường desktop với độ phân giải tối thiểu.

**Điều kiện đầu vào:**

* Trình duyệt desktop: Chrome, Edge, Firefox trên Windows/macOS.
* Độ phân giải màn hình: ≥1366x768.

**Các bước kiểm thử:**

1. Mở website trên từng trình duyệt với độ phân giải ≥1366x768.
2. Kiểm tra bố cục, hình ảnh, font chữ và các thành phần giao diện.

**Kết quả mong đợi:**

* Website hiển thị đúng bố cục và các thành phần giao diện.
* Không xảy ra hiện tượng tràn hoặc lệch bố cục.

**Ký tự hợp lệ**

**Mục tiêu kiểm thử:** Đảm bảo hệ thống xử lý chính xác dữ liệu tiếng Việt có dấu và các ký tự đặc biệt.

**Điều kiện đầu vào:** Form nhập liệu: Họ tên, Địa chỉ, Mô tả sản phẩm, Ghi chú.

**Các bước kiểm thử:**

1. Nhập dữ liệu tiếng Việt có dấu → kiểm tra lưu thành công.
2. Nhập ký tự đặc biệt hợp lệ (ví dụ: @, #, %, &, \*) → kiểm tra lưu thành công.
3. Nhập ký tự không hợp lệ (nếu có quy định) → kiểm tra phản hồi lỗi.

**Kết quả mong đợi:**

* Dữ liệu tiếng Việt có dấu và ký tự đặc biệt hợp lệ được lưu chính xác.
* Hệ thống hiển thị thông báo lỗi khi nhập ký tự không hợp lệ.

### 2.2.4 Giao diện

**Giao diện người dùng – Trang đăng nhập, đăng ký**

**Mục tiêu kiểm thử:** Đảm bảo các trang đăng nhập và đăng ký hiển thị chính xác, các chức năng hoạt động bình thường.

**Điều kiện đầu vào:** Truy cập website trên trình duyệt desktop hoặc mobile.

**Các bước kiểm thử:**

1. Mở trang đăng nhập → kiểm tra bố cục, phông chữ, màu sắc, logo.
2. Mở trang đăng ký → kiểm tra các trường nhập liệu, nút, thông báo lỗi.
3. Thực hiện đăng nhập với thông tin hợp lệ → kiểm tra chuyển hướng đúng trang chính.
4. Thực hiện đăng ký với thông tin hợp lệ → kiểm tra dữ liệu lưu vào cơ sở dữ liệu.

**Kết quả mong đợi:**

* Bố cục, màu sắc, font chữ hiển thị đúng thiết kế.
* Nút, form và thông báo lỗi hiển thị rõ ràng, dễ hiểu.
* Đăng nhập/đăng ký thành công phản hồi đúng.

**Giao diện người dùng – Trang danh mục sản phẩm và chi tiết sản phẩm**

**Mục tiêu kiểm thử:** Đảm bảo trang danh mục và chi tiết sản phẩm hiển thị đúng thông tin và hình ảnh.

**Điều kiện đầu vào:** Sản phẩm đã được thêm vào hệ thống.

**Các bước kiểm thử:**

1. Truy cập trang danh mục sản phẩm → kiểm tra danh sách sản phẩm, hình ảnh, giá, nút “Xem chi tiết”.
2. Nhấp vào sản phẩm → mở trang chi tiết sản phẩm.
3. Kiểm tra thông tin: tên, giá, mô tả, hình ảnh, các tùy chọn (size, màu).

**Kết quả mong đợi:**

* Tất cả sản phẩm hiển thị đầy đủ, bố cục gọn gàng.
* Thông tin chi tiết trùng khớp dữ liệu thực tế.
* Hình ảnh rõ ràng, nút “Thêm vào giỏ hàng” hoạt động.

**Giao diện người dùng – Trang giỏ hàng**

**Mục tiêu kiểm thử:** Đảm bảo giỏ hàng hiển thị sản phẩm, số lượng, giá và các nút chức năng đúng.

**Điều kiện đầu vào:** Người dùng đã thêm sản phẩm vào giỏ hàng.

**Các bước kiểm thử:**

1. Mở trang giỏ hàng → kiểm tra danh sách sản phẩm, số lượng, giá từng sản phẩm và tổng giá.
2. Thay đổi số lượng sản phẩm → kiểm tra cập nhật tổng giá.
3. Xóa sản phẩm → kiểm tra sản phẩm biến mất khỏi giỏ hàng.
4. Nhấn nút “Thanh toán” → kiểm tra chuyển hướng đúng trang thanh toán.

**Kết quả mong đợi:**

* Giỏ hàng hiển thị chính xác thông tin sản phẩm.
* Nút và chức năng hoạt động bình thường, phản hồi nhanh.

**Giao diện người dùng – Trang thanh toán**

**Mục tiêu kiểm thử:** Đảm bảo người dùng có thể nhập thông tin thanh toán và hoàn tất đơn hàng.

**Điều kiện đầu vào:** Người dùng đã thêm sản phẩm vào giỏ hàng.

**Các bước kiểm thử:**

1. Mở trang thanh toán → kiểm tra các trường nhập liệu (thông tin cá nhân, địa chỉ, phương thức thanh toán).
2. Nhập thông tin hợp lệ → nhấn “Thanh toán”.
3. Kiểm tra xác nhận thanh toán thành công.

**Kết quả mong đợi:**

* Form thanh toán hiển thị đầy đủ và dễ thao tác.
* Thanh toán thành công → đơn hàng được tạo → thông tin phản hồi rõ ràng.

**Giao diện quản trị – Trang Admin**

**Mục tiêu kiểm thử:** Đảm bảo giao diện quản trị hiển thị đầy đủ các chức năng và dễ thao tác.

**Điều kiện đầu vào:** Đăng nhập với tài khoản Admin.

**Các bước kiểm thử:**

1. Truy cập trang quản trị → kiểm tra menu, bố cục, các tab chức năng.
2. Quản lý sản phẩm: thêm, sửa, xóa sản phẩm → kiểm tra cập nhật chính xác.
3. Quản lý kho: cập nhật số lượng, kiểm tra tồn kho → dữ liệu hiển thị đúng.
4. Quản lý đơn hàng: xem danh sách, cập nhật trạng thái → kiểm tra chính xác.
5. Quản lý người dùng: xem danh sách, thay đổi quyền, khóa/mở tài khoản → kiểm tra phản hồi.

**Kết quả mong đợi:**

* Bố cục quản trị rõ ràng, dễ thao tác.
* Tất cả chức năng hoạt động bình thường và phản hồi nhanh.
* Dữ liệu hiển thị chính xác, không bị lỗi.

## 2.2.5. Các tính năng không kiểm thử

**Các tính năng nằm ngoài phạm vi kiểm thử**

Một số tính năng trong hệ thống chưa được đưa vào phạm vi kiểm thử của đề tài hiện tạido chưa hoàn thiện hoặc không ảnh hưởng trực tiếp đến quy trình chính.

**Chi tiết như sau:**

|  |  |  |
| --- | --- | --- |
| STT | Tính năng | Lý do không kiểm thử |
| 1 | Tích hợp thanh toán bên thứ ba (PayPal, MoMo, ZaloPay, v.v.) | Phụ thuộc vào API bên thứ ba, môi trường sandbox chưa ổn định. |
| 2 | Gửi email tự động sau khi đặt hàng / đăng ký | Chức năng gửi email đang được mô phỏng (mock), chưa kết nối mail server thật. |
| 3 | Ứng dụng di động (Mobile App) | hiên bản mobile chưa phát triển hoàn chỉnh, chỉ kiểm thử giao diện web. |
| 4 | Báo cáo thống kê nâng cao (Advanced Analytics, Dashboard) | Module chưa hoàn thiện, dữ liệu thống kê chưa đầy đủ. |
| 5 | Tối ưu hiệu năng (Performance, Stress, Load Testing) | Giai đoạn hiện tại chỉ tập trung vào kiểm thử chức năng (Functional Testing). |
| 6 | Giao diện đa ngôn ngữ (Multi-language UI) | Hệ thống hiện tại chỉ triển khai tiếng Việt, bản dịch tiếng Anh chưa hoàn thiện. |

**Lý do loại trừ các tính năng trên khỏi phạm vi kiểm thử**

* **Tài nguyên hạn chế:** Nhóm kiểm thử hiện tập trung vào các luồng nghiệp vụ chính (mua hàng, thanh toán, quản trị).
* **Chưa sẵn sàng về môi trường hoặc dữ liệu:** Một số module phụ thuộc hệ thống hoặc API bên ngoài.

### 2.2.6. Chiến lược kiểm thử

**Các loại kiểm thử**

**Kiểm thử chức năng và khả năng sử dụng**

| **Module** | **Mục tiêu kiểm thử** | **Kỹ thuật** | **Phạm vi áp dụng kỹ thuật** | **Tiêu chí hoàn thành** | **Các cân nhắc đặc biệt** |
| --- | --- | --- | --- | --- | --- |
| *Danh mục sản phẩm* | Đảm bảo sản phẩm được hiển thị chính xác, thêm, sửa, xóa sản phẩm hoạt động | Manual testing, Exploratory testing | Tất cả chức năng CRUD của sản phẩm, danh mục sản phẩm | Tất cả thao tác CRUD thực hiện chính xác, dữ liệu hiển thị đúng | Kiểm thử trên nhiều loại sản phẩm khác nhau, bao gồm tên dài và ký tự đặc biệt |
| *Tồn kho* | Kiểm tra số lượng tồn kho được cập nhật đúng khi nhập/xuất hàng | Manual testing, Boundary value analysis | Chức năng nhập, xuất, điều chỉnh tồn kho | Tồn kho cập nhật đúng, cảnh báo tồn kho thấp hiển thị chính xác | Kiểm tra với số lượng cực đại và cực tiểu |
| *Giỏ hàng* | Đảm bảo thêm/xóa/sửa số lượng sản phẩm trong giỏ hàng hoạt động bình thường | Manual testing, Exploratory testing | Tất cả thao tác giỏ hàng | Giỏ hàng hiển thị chính xác tổng số lượng và giá | Kiểm tra phản hồi nhanh, không mất dữ liệu khi chuyển trang |
| *Quy trình thanh toán* | Kiểm tra người dùng có thể thanh toán thành công, đơn hàng được tạo | Manual testing, Transaction testing | Tất cả luồng thanh toán (thanh toán thành công, thất bại) | Thanh toán thành công → đơn hàng tạo đúng, thông báo chính xác | Kiểm thử dữ liệu hợp lệ và dữ liệu sai, xử lý ngoại lệ |
| *Kiểm soát truy cập* | Đảm bảo người dùng đăng ký/đăng nhập/đổi mật khẩu đúng | Manual testing, Positive/Negative testing | Form đăng ký, đăng nhập, phân quyền người dùng | Người dùng hợp lệ đăng nhập thành công, thông báo lỗi hiển thị đúng | Kiểm tra email tồn tại, mật khẩu không hợp lệ, quyền admin/user |
| *Đánh giá* | Người dùng có thể thêm/xem/sửa/xóa đánh giá sản phẩm | Manual testing, Exploratory testing | Tất cả thao tác đánh giá | Đánh giá hiển thị đúng sản phẩm và người dùng | Kiểm thử với đánh giá dài, ký tự đặc biệt, dấu tiếng Việt |
| *Đơn hàng* | Đảm bảo đơn hàng được tạo, trạng thái cập nhật và xem chi tiết đúng | Manual testing, Transaction testing | Tạo đơn hàng, cập nhật trạng thái, xem danh sách & chi tiết | Đơn hàng tạo và cập nhật đúng trạng thái, thông tin trùng khớp thanh toán | Kiểm tra đa trạng thái, nhiều sản phẩm trong cùng đơn hàng |

**Kiểm thử giao diện người dùng**

| **Module** | **Mục tiêu kiểm thử** | **Kỹ thuật** | **Phạm vi áp dụng kỹ thuật** | **Tiêu chí hoàn thành** | **Các cân nhắc đặc biệt** |
| --- | --- | --- | --- | --- | --- |
| *Danh mục sản phẩm* | Giao diện danh mục và chi tiết sản phẩm hiển thị đúng bố cục, phông chữ, màu sắc | Manual testing, Visual inspection, Cross-browser testing | Trang danh mục và chi tiết sản phẩm | Bố cục, font, màu sắc hiển thị đúng, nút hoạt động | Kiểm thử trên Chrome, Edge, Firefox; responsive trên màn hình 1366x768 |
| *Tồn kho* | Giao diện cập nhật tồn kho rõ ràng, dễ thao tác | Manual testing, Visual inspection | Form nhập/xuất tồn kho | Các trường, nút hiển thị rõ ràng, thao tác dễ hiểu | Kiểm tra hiển thị số lượng lớn và nhỏ |
| *Giỏ hàng* | Giao diện giỏ hàng dễ thao tác, tổng giá hiển thị chính xác | Manual testing, Visual inspection, Usability testing | Trang giỏ hàng | Thông tin sản phẩm, nút, tổng giá hiển thị rõ ràng | Kiểm tra phản hồi nhanh khi thay đổi số lượng, xóa sản phẩm |
| *Quy trình thanh toán* | Giao diện thanh toán rõ ràng, các bước dễ theo dõi | Manual testing, Visual inspection, Usability testing | Form thanh toán, nút xác nhận, thông báo lỗi | Form thanh toán dễ thao tác, thông báo lỗi hiển thị đúng | Kiểm tra nhập dữ liệu dài, ký tự đặc biệt |
| *Kiểm soát truy cập* | Giao diện đăng ký/đăng nhập hiển thị chuẩn, thông báo dễ hiểu | Manual testing, Visual inspection, Usability testing | Trang đăng ký, đăng nhập | Form hiển thị rõ ràng, thông báo lỗi dễ hiểu | Kiểm tra thông báo lỗi email tồn tại, mật khẩu không hợp lệ |
| *Đánh giá* | Giao diện đánh giá rõ ràng, dễ thao tác, hiển thị đúng | Manual testing, Visual inspection | Form thêm/sửa/xóa đánh giá | Đánh giá hiển thị chính xác, dễ thao tác | Kiểm tra đánh giá dài, ký tự đặc biệt |
| *Đơn hàng* | Giao diện danh sách và chi tiết đơn hàng hiển thị đúng | Manual testing, Visual inspection, Usability testing | Trang danh sách & chi tiết đơn hàng | Thông tin đơn hàng hiển thị chính xác, dễ đọc | Kiểm tra trạng thái đa dạng, nhiều sản phẩm trong đơn hàng |

**Công cụ**

| **Công cụ** | **Mục đích sử dụng** |
| --- | --- |
| Browser DevTools (Chrome, Edge, Firefox) | Kiểm tra bố cục, style, console lỗi, responsive layout |
| Postman / Insomnia | Kiểm thử API (CRUD sản phẩm, đơn hàng, kiểm soát truy cập) |
| Excel / Google Sheets | Lập bảng test case, theo dõi kết quả kiểm thử |
| Selenium / Cypress (tùy chọn giai đoạn sau) | Tự động hóa kiểm thử giao diện và luồng nghiệp vụ chính |
| JIRA / Trello | Quản lý bug, ghi nhận kết quả test |

### 2.2.7 Ưu tiên kiểm thử

**Mục đích**

Phần này xác định thứ tự ưu tiên các chức năng, tính năng cần kiểm thử nhằm tập trung nguồn lực vào các phần quan trọng, rủi ro cao, đảm bảo hiệu quả kiểm thử và chất lượng sản phẩm trước khi phát hành.

**Tiêu chí xác định ưu tiên**

**Ưu tiên kiểm thử được dựa trên các tiêu chí chính sau:**

* Tầm quan trọng của chức năng đối với người dùng cuối và mục tiêu kinh doanh.
* Mức độ ảnh hưởng của lỗi trong chức năng đến toàn hệ thống.
* Khối lượng và phức tạp của chức năng, mức độ thay đổi so với phiên bản trước.
* Mức độ tương tác với các module khác và tính liên kết trong hệ thống.

**Các chức năng ưu tiên kiểm thử**

|  |  |  |
| --- | --- | --- |
| **Mức Ưu tiên** | **Chức năng** | **Lý do** |
| Cao | Kiểm soát truy cập (đăng nhập, đăng ký, đăng xuất, phân quyền) | Cổng vào hệ thống; lỗi ngăn truy cập. |
| Cao | Quy trình thanh toán và đơn hàng | Liên quan tài chính, trải nghiệm khách hàng. |
| Cao | Giỏ hàng (thêm, sửa, xóa, cập nhật chi phí) | Bước trung gian quan trọng dẫn đến thanh toán. |
| Trung bình | Danh mục sản phẩm (xem, lọc, sắp xếp, chi tiết, sản phẩm hot) | Hỗ trợ khám phá sản phẩm, ảnh hưởng gián tiếp. |
| Trung bình | Đánh giá sản phẩm | Tăng tương tác, ít rủi ro hơn thanh toán. |
| Trung bình | Quản lý kho hàng (CRUD, gán tồn kho) | Hỗ trợ vận hành, không phải luồng chính. |
| Trung bình | Giao diện và usability (responsive, dễ dùng) | Ảnh hưởng trải nghiệm, không gây lỗi nghiêm trọng. |
| Thấp | Ràng buộc thiết kế (giới hạn đầu vào, ký tự hợp lệ) | Tác động thấp, ít sử dụng thường xuyên. |
| Thấp | Kiểm tra đa nền tảng (trình duyệt phụ, độ phân giải) | Có thể xử lý sau, rủi ro thấp. |

**Quy trình ưu tiên**:

* Sử dụng ma trận ưu tiên (Impact x Likelihood) để xếp hạng test case. Ví dụ: Thanh toán (Impact: 5, Likelihood: 4) có điểm cao hơn hiển thị sản phẩm hot (Impact: 3, Likelihood: 2).
* Đánh giá lại ưu tiên hàng tuần dựa trên feedback từ developer và stakeholder. Nếu có thay đổi yêu cầu, điều chỉnh test case ngay lập tức.

### 2.2.8 Tiêu chí chấp nhận

Tiêu chí chấp nhận được thiết lập để xác định khi nào hệ thống đạt yêu cầu triển khai, dựa trên độ bao phủ, chất lượng, và mức độ lỗi.

|  |  |  |
| --- | --- | --- |
| **Danh mục** | **Tiêu chí** | **Mô tả/Chi tiết** |
| Độ bao phủ | 100% test case ưu tiên cao pass | Đăng nhập, thanh toán, giỏ hàng đạt 100% pass. |
| Độ bao phủ | 85% test case ưu tiên trung bình pass | Danh mục, đánh giá, kho đạt ≥85%. |
| Độ bao phủ | Code coverage ≥75% | White-box testing (nếu áp dụng). |
| Độ bao phủ | Mỗi User Story kiểm thử 1 lần | Tất cả yêu cầu nghiệp vụ được kiểm tra. |
| Mức độ lỗi | Không lỗi critical/high | Crash, mất dữ liệu, thanh toán thất bại, chức năng chính lỗi. |
| Mức độ lỗi | Lỗi medium/low ghi nhận | Medium: Fix trong 7 ngày; Low: Không bắt buộc fix ngay. |
| Mức độ lỗi | Bug tái phát <3% | Đảm bảo chất lượng sửa lỗi. |
| Hiệu suất & Usability | Phản hồi trung bình <2 giây | Theo yêu cầu phi chức năng phần 1.2.2. |
| Hiệu suất & Usability | Responsive trên desktop/mobile | Chrome, Edge, Firefox; độ phân giải ≥1366x768. |
| Hiệu suất & Usability | Không lỗi usability nghiêm trọng | Ví dụ: Nút thanh toán không click được. |
| Tiêu chí chung | Log đầy đủ test case | TestRail/JIRA ghi nhận toàn bộ kết quả. |
| Tiêu chí chung | Báo cáo được phê duyệt | QA Lead và Project Manager phê duyệt. |
| Tiêu chí chung | Không thay đổi yêu cầu trong 48 giờ | Trước khi kết thúc kiểm thử. |

Nếu không đạt, vòng kiểm thử sẽ lặp lại với focus vào bug critical/high, tối đa 2 vòng trước khi escalate lên stakeholder.

## 2.3 Phương pháp và loại kiểm thử

Phương pháp kiểm thử kết hợp manual và automation để tối ưu hóa hiệu quả và độ chính xác, tuân theo phạm vi kiểm thử (phần 1.5) với trọng tâm black-box và white-box, không bao gồm kiểm thử hiệu năng hoặc bảo mật mạng.

|  |  |  |
| --- | --- | --- |
| **Phương pháp/Loại** | **Mô tả** | **Ứng dụng trong hệ thống** |
| Black-box Testing | Kiểm tra chức năng dựa trên yêu cầu, không quan tâm code. | Đăng nhập (dữ liệu hợp lệ/không hợp lệ), lọc sản phẩm, thanh toán. |
| White-box Testing | Kiểm tra logic code, đo độ bao phủ code. | Thanh toán (kiểm tra tồn kho), giỏ hàng (tính tổng chi phí). |
| Exploratory Testing | Kiểm thử tự do để tìm lỗi bất ngờ. | Giao diện (UI), luồng thêm sản phẩm từ nhiều tab. |
| Regression Testing | Chạy lại test case sau fix bug để tránh hồi quy. | Test lại giỏ hàng sau fix lỗi tính tổng chi phí. |
| Automation Testing | Tự động hóa test case lặp lại. | Đăng nhập, thêm giỏ hàng, kiểm tra API Catalog/Payment. |
| **Loại kiểm thử** |  |  |
| Unit Testing | Kiểm tra từng hàm/module riêng lẻ. | Hàm tính tổng chi phí giỏ hàng, CRUD sản phẩm. |
| Integration Testing | Kiểm tra tương tác giữa các module. | Catalog gọi Inventory để kiểm tra tồn kho. |
| System Testing | Kiểm tra luồng end-to-end. | Xem sản phẩm → thêm giỏ → thanh toán → đơn hàng. |
| Acceptance Testing | Xác nhận đáp ứng User Story. | Stakeholder kiểm tra luồng mua hàng. |
| Usability Testing | Đánh giá giao diện và tương tác. | Responsive trên desktop/mobile, bố cục nút. |

**Quy trình**: Áp dụng V-Model, bắt đầu từ unit test trong giai đoạn phát triển, sau đó tích hợp và system test. Automation được triển khai dần cho các test case ưu tiên cao để giảm công sức manual.

## 2.4 Môi trường & công cụ kiểm thử

* **Môi trường kiểm thử**:
  + **Development Environment**: Local machine (Windows/Linux) với .NET Core, NodeJS, và Minikube để mô phỏng Kubernetes/Service Mesh (Istio). Dùng cho unit và integration testing.
  + **Staging Environment**: Cloud-based (AWS/Azure), chạy Kubernetes với Istio, database Postgres/MySQL, và dữ liệu mẫu (tài khoản admin, khách hàng, sản phẩm từ phần 1.1). Truy cập qua URL staging, đảm bảo giống production.
  + **Thiết bị**:
    - Desktop: Windows 10/11, macOS, độ phân giải ≥1366x768.
    - Mobile: Android (Samsung Galaxy S21, Pixel 6), iOS (iPhone 12/13).
    - Trình duyệt: Chrome (latest), Edge, Firefox. Safari chỉ kiểm tra nếu còn thời gian (ưu tiên thấp).
  + **Dữ liệu kiểm thử**: Dữ liệu mẫu từ phần 1.1 (10 sản phẩm, 2 kho, 5 tài khoản khách hàng, 1 admin). Công cụ Faker để sinh email, tên, địa chỉ ngẫu nhiên. Đảm bảo không dùng dữ liệu thật.
* **Công cụ kiểm thử**:
  + **Quản lý test case**: TestRail để viết, lưu trữ, và theo dõi test case. Google Sheets làm phương án dự phòng.
  + **Automation**:
    - Selenium/WebDriverIO: UI testing (đăng nhập, thêm giỏ hàng, thanh toán).
    - Postman/Newman: API testing cho REST API (Catalog, Cart, Payment).
  + **Bug Tracking**: JIRA để log bug, gắn severity (critical/high/medium/low), và assign developer.
  + **Monitoring**: Chrome DevTools để đo thời gian phản hồi (<2s), Kubernetes Dashboard để kiểm tra trạng thái pod.
  + **Reporting**: Allure Report cho automation test, Excel để tạo biểu đồ (pass/fail ratio, bug distribution).
  + **Hỗ trợ**: Git (version control script kiểm thử), Docker (containerize môi trường staging).

**Setup**: Môi trường staging được deploy trước kiểm thử, backup database hàng ngày. Tester xác nhận môi trường ổn định trước khi chạy test case.

## 2.8 Quản lý rủi ro

Quản lý rủi ro nhằm giảm thiểu các vấn đề ảnh hưởng đến chất lượng và tiến độ kiểm thử. Rủi ro được đánh giá dựa trên mức độ tác động (Impact) và xác suất xảy ra (Likelihood).

* **Rủi ro chính**:
  + **Chậm trễ từ developer (Impact: High, Likelihood: Medium)**: Bug fix chậm hoặc code chưa hoàn thiện. **Giảm thiểu**: Daily stand-up để đồng bộ tiến độ, ưu tiên fix lỗi critical/high.
  + **Lỗi môi trường staging (Impact: High, Likelihood: Medium)**: Server down, cấu hình Istio/Kubernetes sai. **Giảm thiểu**: Kiểm tra môi trường trước mỗi vòng test, dùng Prometheus/Grafana để monitor, có backup server.
  + **Thiếu nhân lực kiểm thử (Impact: Medium, Likelihood: High)**: Tester bị phân tán hoặc thiếu kinh nghiệm automation. **Giảm thiểu**: Đào tạo Selenium/Postman trong 2 ngày, thuê freelancer nếu cần.
  + **Bug nghiêm trọng phát hiện muộn (Impact: High, Likelihood: Low)**: Lỗi thanh toán hoặc bảo mật (JWT) xuất hiện ở giai đoạn cuối. **Giảm thiểu**: Smoke test trước mỗi vòng, ưu tiên test thanh toán sớm.
  + **Thay đổi yêu cầu bất ngờ (Impact: Medium, Likelihood: Medium)**: Stakeholder thêm/chỉnh sửa User Story. **Giảm thiểu**: Freeze yêu cầu trước kiểm thử, chỉ chấp nhận thay đổi với phê duyệt từ Project Manager.
* **Kế hoạch ứng phó**:
  + Ma trận rủi ro: Ghi nhận Impact x Likelihood, cập nhật hàng tuần.
  + Contingency plan: Nếu rủi ro xảy ra, mở rộng deadline 1-3 ngày hoặc giảm phạm vi test cho ưu tiên thấp.
  + Escalation: Báo cáo rủi ro critical cho stakeholder qua email/JIRA trong 24 giờ.

## 2.9 Báo cáo kết quả

Báo cáo kiểm thử cung cấp cái nhìn tổng quan và chi tiết về tiến độ, lỗi, và khuyến nghị, được gửi định kỳ (hàng tuần) và cuối giai đoạn kiểm thử.

* **Nội dung báo cáo**:
  + **Tổng quan**:
    - Tổng số test case: Planned, Executed, Passed, Failed.
    - Độ bao phủ: % test case pass, % code coverage (nếu có).
    - Trạng thái: Đạt/không đạt tiêu chí chấp nhận (2.2.8).
  + **Chi tiết bug**:
    - Danh sách bug: ID, module (Catalog, Cart, Payment…), severity, mô tả, trạng thái (Open/Fixed/Closed), assigned developer.
    - Phân bố bug theo module (ví dụ: 5 bug ở Payment, 3 bug ở Cart).
  + **Metrics**:
    - Defect density: Số bug trên mỗi module (bug/module).
    - Pass rate: % test case pass so với tổng số.
    - Thời gian kiểm thử: Tổng giờ thực hiện, thời gian trung bình/test case.
  + **Khuyến nghị**:
    - Nguyên nhân gốc rễ của bug critical/high (ví dụ: thiếu validation trong API thanh toán).
    - Gợi ý cải thiện: Thêm unit test cho logic thanh toán, tối ưu query database.
  + **Kế hoạch tiếp theo**: Fix bug medium/low, chạy regression test, hoặc chuyển sang UAT.
* **Định dạng và công cụ**:
  + **Định dạng**: PDF (báo cáo chính thức), Excel (biểu đồ chi tiết: pie chart cho pass/fail, bar chart cho bug/module).
  + **Công cụ**: TestRail (dashboard tổng quan), Allure Report (automation result), Excel (biểu đồ phân tích).
  + **Lưu trữ**: Repository Git hoặc Google Drive, có version control (ví dụ: Test\_Report\_v1.0.pdf).
* **Quy trình**:
  + Tester log kết quả hàng ngày trên JIRA/TestRail.
  + QA Lead tổng hợp, kiểm tra báo cáo, gửi cho Project Manager/stakeholder trong 24 giờ sau mỗi vòng test.
  + Stakeholder phản hồi trong 48 giờ để phê duyệt hoặc yêu cầu chạy lại test.