

ỦY BAN NHÂN DÂN TP . HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC SÀI GÒN
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỒ ÁN CUỐI KỲ
Học phần: Kiểm thử phần mềm

Đề tài: Xây dựng và kiểm thử ứng dụng web theo quy trình CI/CD và
phương pháp Agile

Giảng viên hướng dẫn: TS. Đỗ Như Tài

Nhóm sinh viên thực hiện:

STT	Họ và tên	MSSV
1	Trang Gia Huy	3122411068
2	Nguyễn Lê Quỳnh Hương	3122411078

TP. HỒ CHÍ MINH, THÁNG 4 NĂM 2025

MỤC LỤC

BẢNG PHÂN CÔNG CÔNG VIỆC	5
CHƯƠNG 1: TỔNG QUAN DỰ ÁN:.....	6
1.1. GIỚI THIỆU ĐỀ TÀI:	6
1.2. NGỮ CẢNH NGHIỆP VỤ:	6
1.2.1. Cơ sở lý thuyết.....	7
1.2.2. Actors (mức nghiệp vụ):	7
1.2.3. Actors (mức hệ thống).....	8
1.2.4. Sơ đồ các tác nhân tham gia nghiệp vụ.....	8
1.2.5. Sơ đồ các tác nhân tham gia hệ thống.....	9
1.2.6. Sơ đồ ngữ cảnh hệ thống.....	9
1.3. QUY TRÌNH NGHIỆP VỤ:	10
1.4. PHÂN TÍCH YÊU CẦU:	15
1.4.1. Các yêu cầu chức năng:.....	15
1.4.2. Các yêu cầu phi chức năng:	18
1.6. USER STORY:.....	20
1.6.1. User story.....	20
1.6.2. Use case:.....	22
1.7. KẾ HOẠCH TRIỂN KHAI DỰ ÁN:.....	36
1.7.1. Cơ sở lý thuyết:.....	36
1.7.2. Kế hoạch thực hiện:	37
CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG:	44
2.1. TỔNG QUAN KIẾN TRÚC HỆ THỐNG:	44
2.1.1. Cơ sở phương pháp luận:	44
2.1.2. Thiết kế sơ đồ khối:	46
2.2. COMMUNICATION VIEW:	48
2.2.1. Cơ sở phương pháp luận:	48
2.2.2. Thiết kế sơ đồ:	49
2.3. DEPLOYMENT VIEW:.....	51
2.3.1. Cơ sở phương pháp luận:	51
2.3.2. Thiết kế sơ đồ:	51
2.4. DECOMPOSITION VIEW C4:	54
2.4.1. Sơ đồ C1 - System Context:.....	54
2.4.2. Sơ đồ C2 - Container:	57

2.4.3. Sơ đồ C3 – Component:	60
2.4.4. Sơ đồ C4 - Code/Implementation Level:.....	63
2.5. ERD	70
2.5.1. Cơ sở lý thuyết:.....	70
2.5.2. Thiết kế sơ đồ:	71
Mức quan niệm (Conceptual):	71
CHƯƠNG 3: KẾ HOẠCH KIỂM THỬ (TEST PLAN):	84
3.1. Cơ sở LÝ THUYẾT VỀ KIỂM THỬ PHẦN MỀM:	84
3.1.1. Khái niệm và Mục tiêu của Kiểm thử Phần mềm	84
3.1.2. Khái niệm Kế hoạch Kiểm thử (Test Plan):	84
3.2. TỔNG QUAN KẾ HOẠCH KIỂM THỬ DỰ ÁN SHOPEASE.....	85
3.2.1. Mục đích và Phạm vi (Scope):	85
3.2.2. Chiến lược Kiểm thử (Test Strategy):.....	86
3.2.3. Môi trường và Nguồn lực	88
CHƯƠNG 4. THIẾT KẾ KIỂM THỬ	91
4.1. GIỚI THIỆU	91
4.2. QUY TRÌNH THIẾT KẾ KIỂM THỬ THEO V-MODEL	91
4.2.2. THIẾT KẾ HỆ THỐNG - 2A	92
4.2.3. THIẾT KẾ KIẾN TRÚC - 3A	92
4.2.4. THIẾT KẾ MODULE - 4A	93
4.2.5. KIỂM THỬ ĐƠN VỊ - 1B	93
4.2.6. KIỂM THỬ TÍCH HỢP - 2B.....	93
4.2.7. KIỂM THỬ HỆ THỐNG - 3B	94
4.2.8. KIỂM THỬ CHẤP NHẬN - 4B.....	94
4.3. KỸ THUẬT THIẾT KẾ KIỂM THỬ.....	95
4.3.1. Kiểm thử hộp đen	95
4.3.2. Kiểm thử hộp trắng.....	98
4.4. PHƯƠNG PHÁP THIẾT KẾ KIỂM THỬ	99
4.4.1. Kiểm thử thủ công.....	99
4.4.2. Kiểm thử tự động	101
CHƯƠNG 5. BÁO CÁO KIỂM THỬ.....	103
5.1. TỔNG QUAN VỀ QUÁ TRÌNH KIỂM THỬ	103
5.2. BÁO CÁO TRƯỜNG HỢP KIỂM THỬ	103

5.2.1. Giới thiệu	103
5.2.2. Phạm vi bao phủ.....	103
5.2.3. Kết quả thực thi.....	104

BẢNG PHÂN CÔNG CÔNG VIỆC

STT	Họ và tên	MSSV	Nội dung công việc
1	Trang Gia Huy	3122411068	<ul style="list-style-type: none">- Soạn Báo cáo đề tài- Soạn Kế hoạch dự án- Phân tích thiết kế hệ thống- Soạn Kế hoạch kiểm thử (hỗ trợ)- Phát triển Backend và Deploy dự án
2	Nguyễn Lê Quỳnh Hương	3122411078	<ul style="list-style-type: none">- Soạn Tổng quan dự án- Vẽ các sơ đồ Use Cases, Screen- Soạn Kế hoạch kiểm thử (chính)- Phát triển Frontend và Deploy dự án

Chương 1: Tổng quan dự án:

1.1. Giới thiệu đề tài:

Dự án kiểm thử “Website thương mại điện tử Electro” được thực hiện nhằm đánh giá mức độ đáp ứng các yêu cầu chức năng của hệ thống và phát hiện các lỗi phát sinh trong quá trình phát triển. Hoạt động kiểm thử tập trung vào việc xác minh tính đúng đắn của dữ liệu, sự ổn định trong quá trình vận hành và tính chính xác của các chức năng nghiệp vụ chính của website.

Quá trình kiểm thử bao gồm việc phân tích yêu cầu, xây dựng và thực thi các trường hợp kiểm thử, thực hiện kiểm thử đơn vị, kiểm thử tích hợp, kiểm thử hệ thống và kiểm thử chấp nhận nhằm đảm bảo hệ thống hoạt động đúng theo đặc tả trước khi triển khai sử dụng.

1.2. Ngữ cảnh nghiệp vụ:

Website Electro E-Commerce bao gồm các nghiệp vụ chính: Xác thực & Phân quyền người dùng, Danh mục sản phẩm, Giỏ hàng & Đặt hàng & Thanh toán và Quản lý tồn kho. Các nghiệp vụ này mô phỏng quy trình kinh doanh cơ bản của một hệ thống thương mại điện tử hiện đại, đáp ứng nhu cầu của cả khách hàng và quản trị viên hệ thống.

Với chức năng **Xác thực & Phân quyền**, người mua có thể đăng ký tài khoản mới bằng email, mật khẩu và các thông tin cá nhân khác sau đó đăng nhập để truy cập vào các chức năng mua sắm trong hệ thống. Sau khi được xác thực, người dùng có thể xem và cập nhật thông tin cá nhân cũng như đăng xuất khi cần. Quản trị viên trong hệ thống cũng có thể đăng nhập để truy cập vào trang quản trị, nơi họ có thể xem danh sách người dùng, xem chi tiết thông tin từng tài khoản, và cập nhật trạng thái hoạt động của người dùng (kích hoạt hoặc vô hiệu hóa tài khoản). Cơ chế xác thực của hệ thống dựa trên Spring Security kết hợp với JWT (JSON Web Token) nhằm đảm bảo an toàn truy cập và phân quyền hợp lý giữa người dùng thông thường và quản trị viên.

Với chức năng **Danh mục sản phẩm**, người mua có thể xem danh sách sản phẩm mới được hiển thị trong cửa hàng, đồng thời sử dụng bộ lọc theo danh mục, khoảng giá, thương hiệu hoặc từ khóa tìm kiếm để dễ dàng tìm được sản phẩm mong muốn. Khi truy cập trang chi tiết sản phẩm, người dùng có thể xem tên sản phẩm, mô tả, giá, hình ảnh, tình trạng còn hàng và số lượng tồn kho. Quản trị viên có toàn quyền quản lý sản phẩm, danh mục sản phẩm và các thông tin có liên quan đến sản phẩm bao gồm thêm mới, cập nhật, xóa sản phẩm cũ hoặc không còn kinh doanh, đồng thời có thể phân loại sản phẩm theo danh mục để tối ưu việc hiển thị và tìm kiếm. Hệ thống hỗ trợ upload hình ảnh sản phẩm để đảm bảo sản phẩm luôn có hình ảnh trực quan, hấp dẫn với người mua.

Với chức năng **Giỏ hàng & Đặt hàng & Thanh toán**, người mua có thể thêm sản phẩm vào giỏ hàng từ trang danh sách hoặc trang chi tiết sản phẩm. Giỏ hàng sẽ hiển thị đầy đủ danh sách sản phẩm đã chọn, số lượng, đơn giá, tổng tiền tạm tính. Mỗi khi người dùng thay đổi số lượng hoặc xóa sản phẩm, giỏ hàng và bảng tổng kết sẽ được cập nhật tự động. Khi người mua quyết định thanh toán, hệ thống sẽ cho phép đặt hàng theo phương thức thanh toán khi nhận hàng (COD). Sau khi đặt hàng, một đơn hàng mới sẽ được tạo và lưu trong hệ thống với trạng thái ban đầu là “Chờ xác nhận”. Người dùng có thể theo dõi trạng thái đơn hàng của mình (Chờ xác nhận, Đang giao, Hoàn tất hoặc Đã hủy). Phía quản trị viên có thể xem danh sách tất cả các đơn hàng, xem chi tiết từng đơn, cập nhật trạng thái đơn hàng khi được xử lý hoặc hủy đơn hàng với lý do hủy được ghi nhận rõ ràng để phục vụ quản lý nội bộ.

Với chức năng **Quản lý tồn kho**, quản trị viên có thể theo dõi số lượng tồn của từng sản phẩm, thực hiện nhập kho khi có đơn mua hàng từ nhà cung cấp, hoặc xuất kho khi sản phẩm được giao cho khách. Hệ thống cho phép quản trị viên xem danh sách tồn kho chi tiết, bao gồm mã sản phẩm, tên sản phẩm, số lượng hiện tại, lịch sử nhập – xuất. Ngoài ra, quản trị viên có thể tạo đơn mua hàng để nhập bổ sung hàng hóa khi lượng tồn xuống thấp, đảm bảo hoạt động kinh doanh không bị gián đoạn. Bên cạnh đó, quản trị viên còn có thể xem chi tiết, cập nhật các phiếu xuất, nhập kho, đơn mua hàng để có thể nắm rõ chi tiết thông tin các phiếu.

Dữ liệu khởi tạo (One-off Tasks): Khi hệ thống được khởi động lần đầu, một số dữ liệu mẫu sẽ được khởi tạo để phục vụ quá trình thử nghiệm và trình diễn chức năng, bao gồm: Một tài khoản Admin mặc định để truy cập trang quản trị. Hai tài khoản Client để thử nghiệm luồng mua hàng. Bộ dữ liệu sản phẩm mẫu gồm nhiều danh mục và hình ảnh. Một số đơn hàng mẫu ở các trạng thái khác nhau (chờ xác nhận, đang giao, đã giao, đã hủy). Những dữ liệu khởi tạo này giúp hệ thống có thể được kiểm thử và trình bày đầy đủ chức năng ngay sau khi khởi động.

1.2.1. Cơ sở lý thuyết

1.2.1.1. Actor là gì ?

Actor là tác nhân tương tác với hệ thống từ bên ngoài. Actor có thể là con người, hệ thống khác, thiết bị.

Actor (trong môi trường nghiệp vụ): ám chỉ vai trò mà một người hay một thứ gì đó nắm giữ trong khi tương tác với nghiệp vụ. Những loại người dùng nghiệp vụ sau đây có khả năng được xem là những tác nhân nghiệp vụ: khách hàng, nhà cung cấp, đối tác, đồng nghiệp ở những nghiệp vụ không được mô hình hóa... Như vậy, một tác nhân thường tương ứng với một người sử dụng. Tuy nhiên, có những tình huống, chẳng hạn như một hệ thống thông tin đóng vai trò của một tác nhân. Ví dụ, ngân hàng có thể quản lý hầu hết các giao dịch trực tuyến từ một máy tính thì các use case của hệ thống sẽ tương tác với ngân hàng, khi đó ngân hàng được xem là một tác nhân, điều đó có nghĩa tác nhân lúc này là một hệ thống thông tin.

1.2.1.2. Cách xác định actor mức nghiệp vụ

Để xác định Actor ở mức nghiệp vụ (Business Level), ta tập trung vào vai trò con người hoặc tổ chức trong hoạt động kinh doanh, chứ chưa đi sâu vào phần mềm cụ thể. Các bước xác định gồm:

Xác định các bên liên quan (Stakeholders): Ai là người tham gia vào quy trình nghiệp vụ chính (ví dụ: khách hàng, nhân viên, quản lý)?

Xác định mục tiêu tương tác: Họ muốn đạt được điều gì khi tương tác với hệ thống? (mua hàng, quản lý kho, xác nhận đơn, v.v.)

Phân biệt vai trò: Một người có thể đảm nhiệm nhiều vai trò khác nhau → mỗi vai trò tương ứng với một Actor riêng.

Chỉ chọn Actor bên ngoài hệ thống: Actor không phải là phần của hệ thống phần mềm, mà là bên tương tác với nó.

1.2.2. Actors (mức nghiệp vụ):

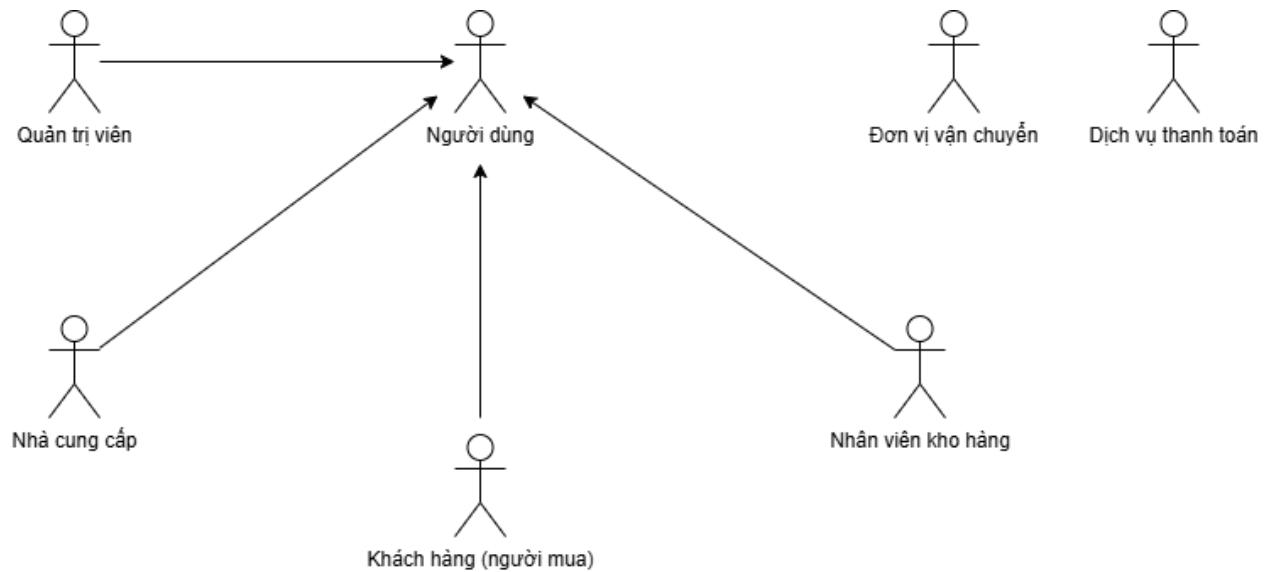
Tên actor	Mô tả vai trò	Mục tiêu trong quy trình nghiệp vụ

Người mua (Buyer)	Người mua hàng hóa qua website hoặc cửa hàng. Là người tạo ra nhu cầu mua sắm.	Chọn mua sản phẩm, thanh toán, nhận hàng.
Quản trị viên (Admin)	Người chịu trách nhiệm quản lý và điều hành hoạt động kinh doanh của hệ thống.	Quản lý sản phẩm, người dùng, đơn hàng, tồn kho và xử lý các sự cố.
Nhà cung cấp (Supplier)	Đơn vị cung cấp sản phẩm hoặc hàng hóa cho hệ thống.	Cung cấp sản phẩm đúng số lượng, chất lượng và thời gian để nhập kho.
Nhân viên kho hàng	Người chịu trách nhiệm kiểm kê, nhập và xuất hàng trong kho.	Cập nhật tình trạng tồn kho, kiểm tra số lượng, chất lượng sản phẩm khi nhập/xuất kho.
Hệ thống thanh toán (Payment Gateway / COD)	Module hoặc API xử lý thanh toán COD / VNPay	Ghi nhận và xác thực các giao dịch thanh toán của khách hàng.
Đơn vị vận chuyển	Bên thứ ba hỗ trợ vận chuyển hàng hóa.	Tiếp nhận và thực hiện việc giao hàng thay cho hệ thống.

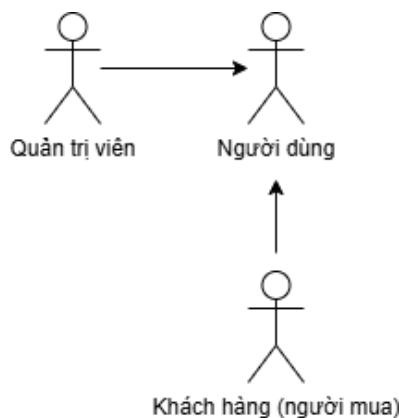
1.2.3. Actors (mức hệ thống)

Tên Actor	Mô tả vai trò trong hệ thống phần mềm	Mục tiêu hệ thống
Người mua (Buyer)	Người dùng đã đăng ký trong hệ thống, thực hiện mua hàng qua giao diện web	Tương tác với hệ thống qua API: đăng nhập, xem sản phẩm, thêm vào giỏ hàng, đặt hàng
Quản trị viên (Admin)	Tài khoản quản trị có toàn quyền quản lý dữ liệu	Quản lý sản phẩm, danh mục, đơn hàng, người dùng và kho hàng

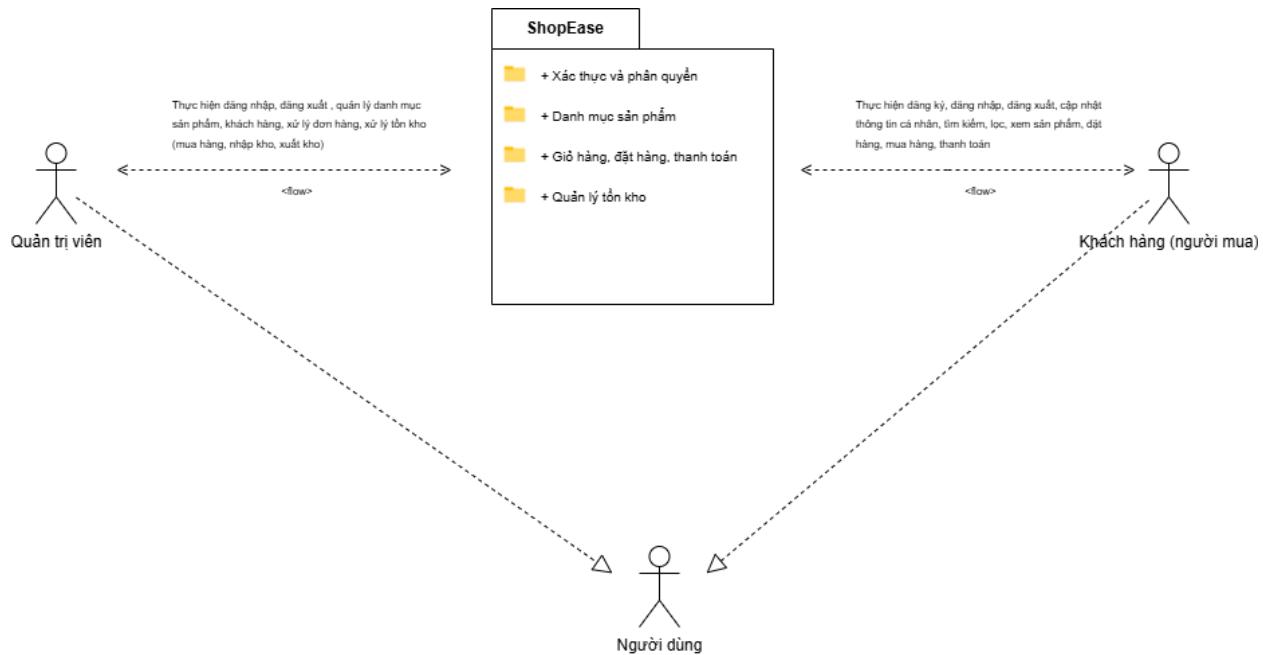
1.2.4. Sơ đồ các tác nhân tham gia nghiệp vụ



1.2.5. Sơ đồ các tác nhân tham gia hệ thống



1.2.6. Sơ đồ ngũ cảnh hệ thống



1.3. Quy trình nghiệp vụ:

Trong doanh nghiệp, bất cứ hoạt động nào tiếp nhận các đầu vào và chuyển thành các đầu ra có thể coi như một quy trình.

Quy trình là một tập hợp các bước/công việc được lập thành văn bản có thể được lặp lại để hoàn thành một mục tiêu cụ thể mà không tạo ra lỗi.

Quy trình doanh nghiệp (Business Process) là tập hợp các công việc được thực hiện theo một thứ tự cố định, nhằm biến đổi các yếu tố đầu vào thành các kết quả đầu ra.

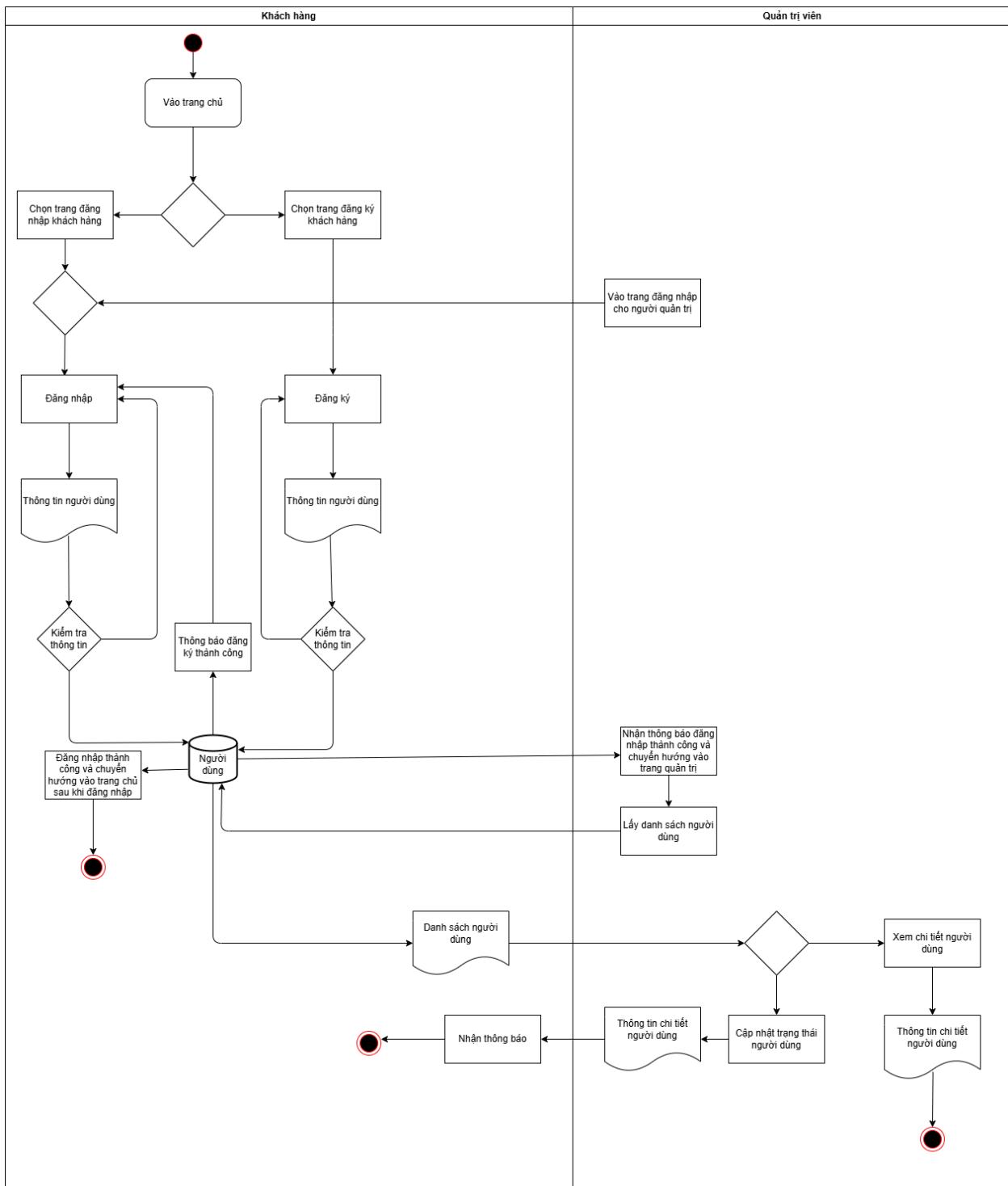
BPM (Business Process Management - “Quản lý quy trình doanh nghiệp” hay “Quản lý quy trình nghiệp vụ”) là việc quản lý và cải tiến các quy trình của công ty sao cho kết quả có thể lặp lại và tránh được các lỗi.

a. Quy trình xác thực và phân quyền:

Quy trình xác thực và phân quyền được thực hiện như sau:

Khách hàng truy cập vào website để mua sắm thì phải thực hiện đăng ký tài khoản nếu chưa có tài khoản, sau đó đăng nhập vào hệ thống, sau khi đăng nhập vào hệ thống thì khách hàng có thể xem được thông tin cá nhân của mình và tiến hành mua sắm. Quản trị viên có thể quản lý danh sách người dùng qua việc xem danh sách, chi tiết thông tin người dùng, cập nhật trạng thái tài khoản người dùng.

Đối tượng, bộ phận tham gia: khách hàng, quản trị viên.

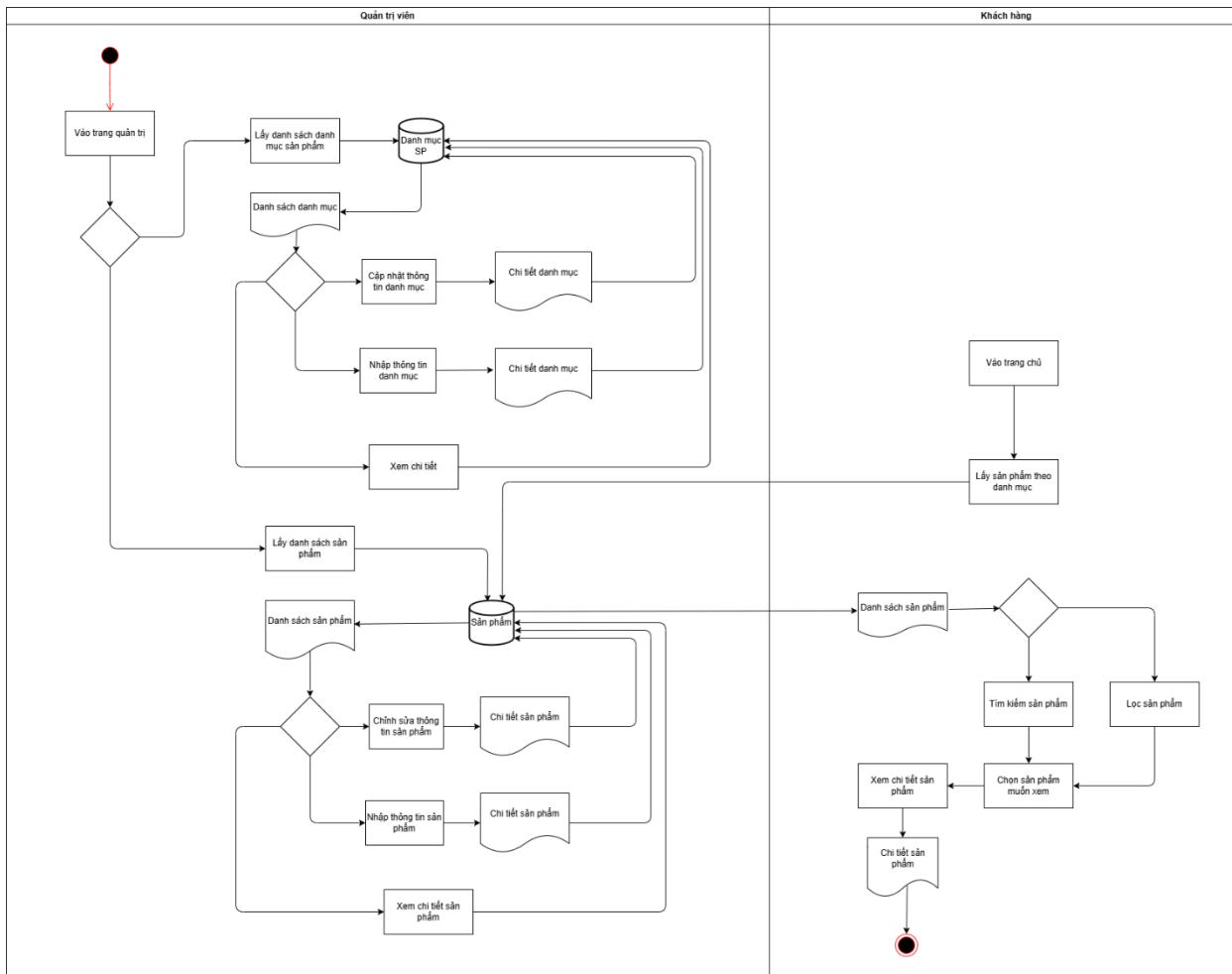


b. Quy trình quản lý danh mục sản phẩm:

Quy trình quản lý danh mục sản phẩm được thực hiện như sau:

Quản trị viên có thể quản lý danh mục sản phẩm qua việc thêm, cập nhật, xóa sản phẩm và danh mục sản phẩm. Khách hàng có thể xem danh sách sản phẩm, chi tiết sản phẩm, ngoài ra còn có thể tìm kiếm, lọc theo các tiêu chí để có thể tìm kiếm sản phẩm nhanh hơn.

Đối tượng, bộ phận tham gia: quản trị viên, khách hàng



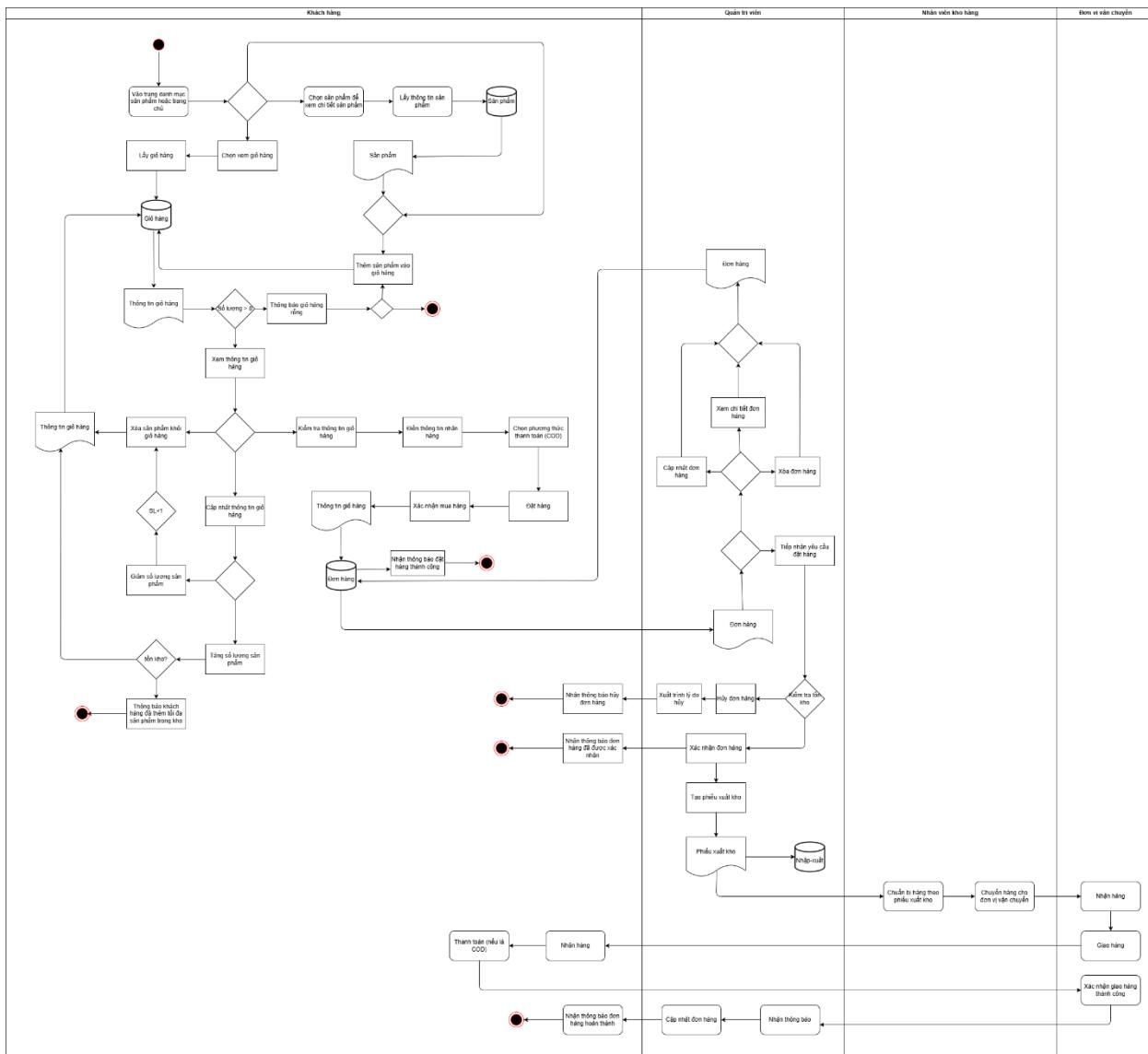
c. Quy trình quản lý giỏ hàng & đặt hàng & thanh toán

Quy trình quản lý giỏ hàng & đặt hàng & thanh toán được thực hiện như sau:

Khách hàng sau khi đăng nhập vào hệ thống thì có thể bắt đầu quá trình mua sắm, khách hàng có thể thêm sản phẩm vào giỏ hàng ở trang chủ/danh mục sản phẩm/chi tiết sản phẩm. Khi đã có sản phẩm trong giỏ hàng, khách hàng có thể điều chỉnh số lượng của sản phẩm hoặc xóa sản phẩm ra khỏi giỏ hàng nếu không còn nhu cầu mua sắm nữa. Sau khi đã chọn được sản phẩm cần mua ở giỏ hàng, khách hàng có thể tiến hành kiểm tra các thông tin về phí mua hàng, thông tin nhận hàng và hình thức thanh toán, sau khi kiểm tra xong thì chọn đặt hàng. Sau khi khách hàng đặt hàng thành công, quản trị viên nhận được thông báo có đơn hàng, sau đó quản trị viên kiểm tra hàng

trong kho, xác nhận đơn hàng và tạo phiếu xuất kho cho nhân viên kho hàng đóng gói sản phẩm và đưa đơn hàng cho đơn vị vận chuyển. Sau khi giao hàng thành công cho khách hàng thì quản trị viên sẽ nhận được thông báo giao hàng thành công và sau đó cập nhật đơn hàng cho khách hàng.

Đối tượng, bộ phận tham gia: quản trị viên, nhân viên kho hàng, đơn vị vận chuyển.

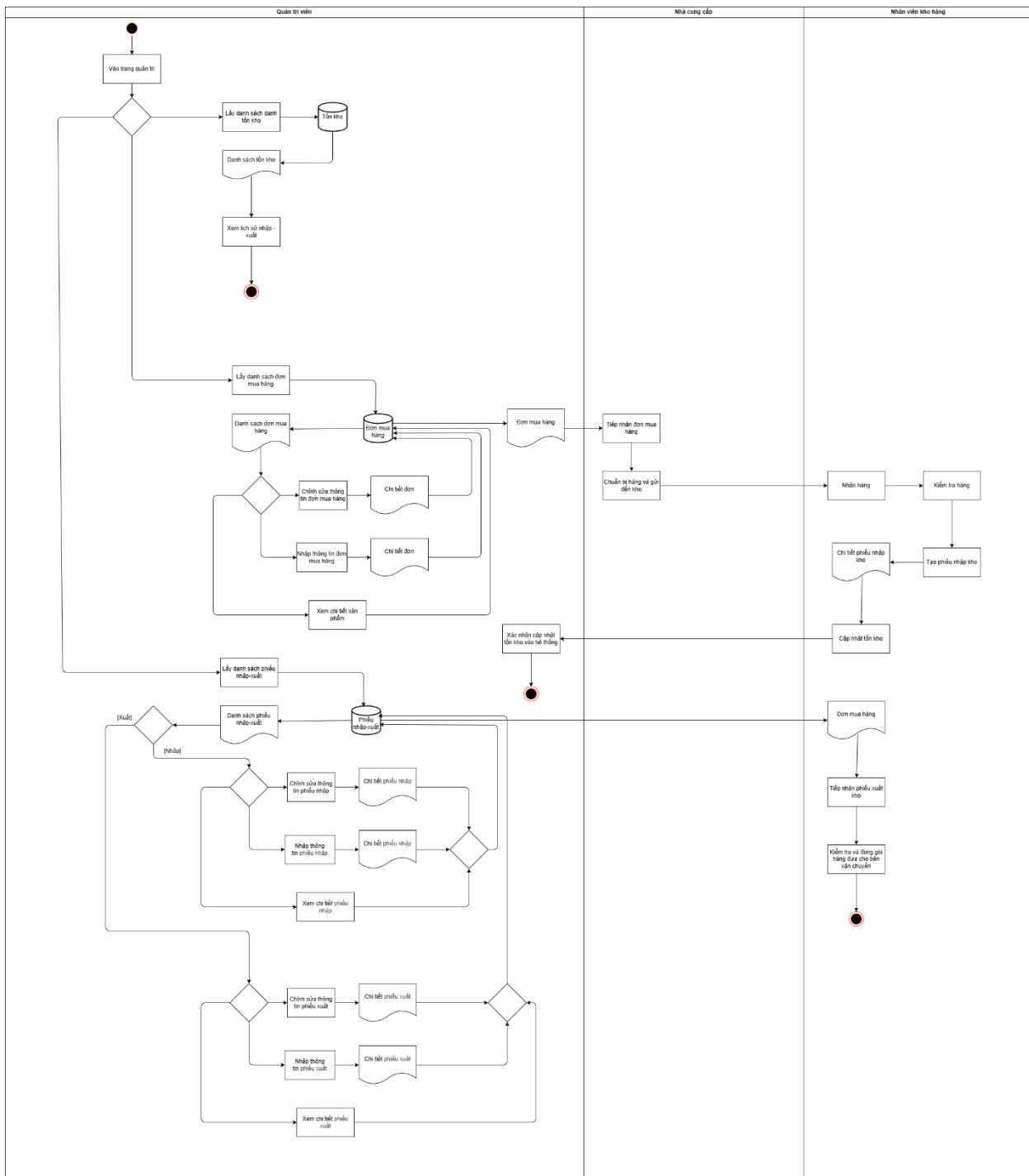


d. Quy trình quản lý tồn kho

Quy trình quản lý tồn kho được thực hiện như sau:

Quản trị viên theo dõi tồn kho, lập đơn mua hàng khi kiểm tra tồn kho trong cửa hàng không còn đủ số lượng theo quy định của cửa hàng, lập phiếu nhập kho khi nhập hàng từ nhà cung cấp, lập phiếu xuất kho khi bán hàng và đưa phiếu xuất kho cho nhân viên kho hàng tiến hành đóng gói và đưa đến đơn vị vận chuyển.

Đối tượng, bộ phận tham gia: quản trị viên, nhà cung cấp, nhân viên kho hàng.



1.4. Phân tích yêu cầu:

1.4.1. Các yêu cầu chức năng:

- **Yêu cầu chức năng** là các mô tả cụ thể về những chức năng hoặc hành vi mà hệ thống phải thực hiện nhằm đáp ứng mục tiêu nghiệp vụ.

- Cách xác định:

Phân tích các use case hoặc user story để tìm hành động chính mà người dùng thực hiện.

Phỏng vấn, khảo sát người dùng để hiểu mục tiêu nghiệp vụ.

Dựa vào mô tả quy trình nghiệp vụ (business process) để tách ra các chức năng hệ thống cần có.

Xác định ai (actor) tương tác với hệ thống và tương tác để làm gì.

#	Tên chức năng	Nội dung yêu cầu
BR1	Xác thực & Phân quyền	
BR1.1	Đăng ký tài khoản	Người mua có thể đăng ký tài khoản mới bằng email, mật khẩu và thông tin cá nhân (họ tên, địa chỉ, số điện thoại) để có thể mua sắm trực tuyến.
BR1.2	Đăng nhập	Người mua và quản trị viên có thể đăng nhập bằng thông tin tài khoản hợp lệ để truy cập các chức năng được phân quyền.
BR1.3	Cập nhật thông tin cá nhân	Người mua và quản trị viên có thể đăng nhập bằng thông tin tài khoản hợp lệ để truy cập các chức năng được phân quyền.
BR1.4	Xem danh sách người dùng	Quản trị viên có thể danh sách người dùng để nắm được các tài khoản đang hoạt động trên hệ thống.
BR1.5	Xem chi tiết người dùng	Quản trị viên có thể xem chi tiết thông tin từng người dùng để quản lý và xử lý khi cần thiết.
BR1.6	Cập nhật trạng thái tài khoản	Quản trị viên có thể cập nhật trạng thái tài khoản (kích hoạt / vô hiệu hóa) để đảm bảo tính an toàn và kiểm soát truy cập hợp lý.
BR1.7	Đăng xuất	Người mua và quản trị viên có thể đăng xuất để bảo mật tài khoản.

BR2	Danh mục sản phẩm	
BR2.1	Xem danh sách sản phẩm mới nhất	Người mua có thể xem danh sách tất cả sản phẩm mới nhất
BR2.2	Lọc sản phẩm theo danh mục, khoảng giá, thương hiệu	Người mua có thể lọc sản phẩm theo danh mục, khoảng giá, thương hiệu để tìm sản phẩm nhanh hơn.
BR2.3	Tìm kiếm sản phẩm theo từ khóa	Người mua có thể tìm kiếm sản phẩm theo từ khóa để tìm sản phẩm nhanh hơn.
BR2.4	Xem thông tin chi tiết sản phẩm	Người mua có thể xem thông tin chi tiết của sản phẩm (tên, mô tả, giá, hình ảnh, tình trạng còn hàng)
BR2.5	Kiểm tra số lượng tồn kho	Người mua có thể xem số lượng tồn kho của từng sản phẩm
BR2.6	Thêm mới sản phẩm	Quản trị viên có thể thêm mới sản phẩm vào hệ thống
BR2.7.	Xem chi tiết sản phẩm	Quản trị viên có thể xem chi tiết sản phẩm có trong hệ thống
BR2.7	Cập nhật thông tin sản phẩm	Quản trị viên có thể cập nhật thông tin sản phẩm.
BR2.8	Xóa sản phẩm	Quản trị viên có thể xóa sản phẩm cũ hoặc ngừng kinh doanh
BR2.9	Quản lý danh mục sản phẩm	Quản trị viên có thể quản lý danh mục sản phẩm (xem, cập nhật, xóa)
BR3	Giỏ hàng – Đặt hàng – Thanh toán	
BR3.1	Thêm sản phẩm vào giỏ hàng	Người mua có thể thêm sản phẩm vào giỏ hàng
BR3.2	Xem danh sách sản phẩm trong giỏ hàng	Người mua có thể xem danh sách sản phẩm trong giỏ hàng
BR3.3	Điều chỉnh giỏ hàng	Người mua có thể thay đổi số lượng hoặc xóa sản phẩm khỏi giỏ hàng

BR3.4	Xem tổng tiền và phí	Người mua có thể xem tổng tiền tạm tính, phí vận chuyển và tổng đơn hàng
BR3.5	Thanh toán bằng COD	Người mua có thể thực hiện thanh toán bằng phương thức COD (thanh toán khi nhận hàng) để hoàn tất đơn mua.
BR3.6	Theo dõi trạng thái đơn hàng	Người mua có thể theo dõi trạng thái đơn hàng (chờ xác nhận, đang giao, hoàn tất, đã hủy)
BR3.7	Xem danh sách đơn hàng	Quản trị viên có thể xem danh sách tất cả đơn hàng của khách hàng
BR3.8	Xem chi tiết đơn hàng	Quản trị viên có thể xem chi tiết từng đơn hàng
BR3.9	Cập nhật trạng thái đơn hàng	Quản trị viên có thể cập nhật trạng thái đơn hàng (chuẩn bị, giao hàng, hoàn tất).
BR3.10	Hủy đơn hàng	Quản trị viên có thể hủy đơn hàng và ghi lại lý do hủy
BR4	Quản lý tồn kho	
BR4.1	Xem danh sách tồn kho	Quản trị viên có thể xem danh sách tồn kho chi tiết
BR4.2	Tạo phiếu nhập kho	Quản trị viên có thể tạo phiếu nhập kho khi có đơn mua hàng từ nhà cung cấp
BR4.3	Xem chi tiết phiếu nhập kho	Quản trị viên có thể xem chi tiết phiếu nhập kho
BR4.4	Cập nhật phiếu nhập kho	Quản trị viên có thể cập nhật phiếu nhập kho
BR4.5	Tạo phiếu xuất kho	Quản trị viên có thể tạo phiếu xuất kho khi có đơn hàng giao cho khách để ghi nhận sự biến động tồn kho chính xác.
BR4.6	Xem chi tiết phiếu xuất kho	Quản trị viên có thể xem chi tiết phiếu xuất kho

BR4.7	Cập nhật phiếu xuất kho	Quản trị viên có thể cập nhật phiếu xuất kho
BR4.8	Xem lịch sử nhập – xuất kho	Quản trị viên có thể xem lịch sử nhập – xuất kho
BR4.9	Tạo đơn mua hàng mới	Quản trị viên có thể tạo đơn mua hàng mới khi hàng sắp hết
BR4.10	Xem chi tiết đơn mua hàng	Quản trị viên có thể xem chi tiết đơn mua hàng
BR4.11	Cập nhật đơn mua hàng	Quản trị viên có thể cập nhật đơn mua hàng

1.4.2. Các yêu cầu phi chức năng:

- Yêu cầu phi chức năng bao gồm tất cả những yêu cầu mà yêu cầu chức năng không có. Chúng chỉ ra những tiêu chí để đánh giá hoạt động của hệ thống thay vì hành vi.

- Cách xác định:

Xác định các ràng buộc kỹ thuật, thiết kế, bảo mật mà hệ thống cần tuân theo.

Xem xét trải nghiệm người dùng (Usability), hiệu năng (Performance), tính tương thích (Compatibility), bảo mật (Security), tính mở rộng (Scalability),...

BR5	Usability (Tính dễ sử dụng)	
BR5.1		<p>Hệ thống phải hỗ trợ quản lý phiên người dùng (session) để duy trì trạng thái đăng nhập trong suốt quá trình sử dụng</p> <p>Hệ thống phải cung cấp điều hướng trực quan, cho phép người dùng cập nhật giờ hàng và các thao tác khác theo thời gian thực.</p>
BR6	Compatibility (Khả năng tương thích)	

BR6.1		<p>Hệ thống phải hoạt động ổn định và chính xác trên các trình duyệt phổ biến, bao gồm:</p> <ul style="list-style-type: none"> - Google Chrome (phiên bản mới nhất, Windows 11) - Mozilla Firefox (phiên bản mới nhất, Windows 11) - Microsoft Edge (phiên bản mới nhất, Windows 11)
BR7	Interfaces (Giao diện)	
BR7.1		<p>Giao diện người dùng phải hiển thị đúng thông tin và chức năng tương ứng với từng vai trò người dùng (Người mua, Quản trị viên).</p> <p>Giao diện đơn giản, dễ sử dụng</p>

1.6. User story:

1.6.1. User story

- User Story là mô tả ngắn gọn về một yêu cầu hoặc chức năng của hệ thống được viết dưới góc nhìn của người dùng cuối (end user).

- Cách xác định user story:

- Xác định đối tượng người dùng: Ai sẽ sử dụng hệ thống? (VD: người mua, quản trị viên)
- Xác định mục tiêu của họ: Họ muốn đạt được điều gì khi sử dụng hệ thống?
- Chuyển mục tiêu thành user story: Viết theo mẫu “Là ... tôi muốn ... để ...”
- Nhóm user story theo module chức năng: Gom các story có liên quan (Auth, Product, Cart, Order, Inventory, ...)

- User stories cho Website ShopEase E-Commerce:

Xác thực & Phân quyền (Authentication & Access Control)

- Là người mua, tôi muốn đăng ký tài khoản để có thể mua sắm trực tuyến.
- Là người mua, tôi muốn đăng nhập bằng thông tin tài khoản để truy cập vào tài khoản cá nhân của mình.
- Là người mua, tôi muốn cập nhật thông tin cá nhân để đảm bảo dữ liệu của tôi luôn chính xác và mới nhất.
- Là người mua, tôi muốn đăng xuất khỏi hệ thống để bảo mật tài khoản sau khi mua sắm.
- Là quản trị viên, tôi muốn đăng nhập bằng tài khoản quản trị để truy cập trang quản lý hệ thống.
- Là quản trị viên, tôi muốn xem danh sách người dùng để nắm được các tài khoản đang hoạt động trên hệ thống.
- Là quản trị viên, tôi muốn xem chi tiết thông tin từng người dùng để quản lý và xử lý khi cần thiết.
- Là quản trị viên, tôi muốn cập nhật trạng thái tài khoản (kích hoạt / vô hiệu hóa) để đảm bảo tính an toàn và kiểm soát truy cập hợp lý.
- Là quản trị viên, tôi muốn đăng xuất khỏi hệ thống để bảo mật tài khoản quản trị.

Danh mục sản phẩm (Product Catalog)

- Là người mua, tôi muốn xem danh sách tất cả sản phẩm mới nhất để dễ dàng tìm kiếm mặt hàng cần mua.
- Là người mua, tôi muốn lọc sản phẩm theo danh mục, khoảng giá, thương hiệu để tìm sản phẩm nhanh hơn.
- Là người mua, tôi muốn tìm kiếm sản phẩm theo từ khóa để tìm kiếm sản phẩm nhanh hơn
- Là người mua, tôi muốn xem thông tin chi tiết của sản phẩm (tên, mô tả, giá, hình ảnh, tình trạng còn hàng) để có thể quyết định mua hay không.

- Là người mua, tôi muốn xem số lượng tồn kho của từng sản phẩm để chắc chắn rằng sản phẩm còn hàng.
- Là quản trị viên, tôi muốn xem chi tiết sản phẩm để nắm rõ được các thông tin của sản phẩm
- Là quản trị viên, tôi muốn thêm mới sản phẩm vào hệ thống để mở rộng danh mục hàng hóa.
- Là quản trị viên, tôi muốn cập nhật thông tin sản phẩm để đảm bảo dữ liệu hiển thị luôn chính xác.
- Là quản trị viên, tôi muốn xóa sản phẩm cũ hoặc ngừng kinh doanh để giữ cho danh mục luôn gọn gàng và hợp lệ.
- Là quản trị viên, tôi muốn quản lý danh mục sản phẩm (xem, cập nhật, xóa) để quản lý sản phẩm logic hơn và để người mua dễ dàng tìm kiếm hơn

Giỏ hàng – Đặt hàng – Thanh toán (Shopping Cart, Checkout & Payment)

- Là người mua, tôi muốn thêm sản phẩm vào giỏ hàng để chuẩn bị cho việc thanh toán sau này.
- Là người mua, tôi muốn xem danh sách sản phẩm trong giỏ hàng để kiểm tra lại các mặt hàng đã chọn.
- Là người mua, tôi muốn thay đổi số lượng hoặc xóa sản phẩm khỏi giỏ hàng để điều chỉnh đơn hàng trước khi thanh toán.
- Là người mua, tôi muốn xem địa chỉ, phương thức vận chuyển, tổng tiền tạm tính, phí vận chuyển và tổng đơn hàng để nắm được chi phí trước khi mua.
- Là người mua, tôi muốn thực hiện thanh toán bằng phương thức COD (thanh toán khi nhận hàng) để hoàn tất đơn mua.
- Là người mua, tôi muốn theo dõi trạng thái đơn hàng để biết tiến trình đơn hàng của mình.
- Là quản trị viên, tôi muốn xem danh sách tất cả đơn hàng của khách hàng để quản lý quá trình bán hàng hiệu quả.
- Là quản trị viên, tôi muốn xem chi tiết từng đơn hàng để kiểm tra thông tin đặt hàng và xử lý nếu cần.
- Là quản trị viên, tôi muốn cập nhật trạng thái đơn hàng để phản ánh đúng tiến độ xử lý.
- Là quản trị viên, tôi muốn hủy đơn hàng và ghi lại lý do hủy để phục vụ công tác theo dõi và quản lý sau này.

Quản lý tồn kho (Inventory Management)

- Là quản trị viên, tôi muốn xem danh sách tồn kho chi tiết để nắm được số lượng hàng hóa hiện có.
- Là quản trị viên, tôi muốn tạo phiếu nhập kho khi có đơn mua hàng từ nhà cung cấp để bổ sung lượng hàng thiếu hụt.

- Là quản trị viên, tôi muốn xem chi tiết thông tin phiếu nhập kho để kiểm soát thông tin phiếu nhập kho
- Là quản trị viên, tôi muốn cập nhật thông tin phiếu nhập kho để kiểm soát thông tin phiếu nhập kho
- Là quản trị viên, tôi muốn tạo phiếu xuất kho khi có đơn hàng giao cho khách để ghi nhận sự biến động tồn kho chính xác.
- Là quản trị viên, tôi muốn xem chi tiết thông tin phiếu xuất kho để kiểm soát thông tin phiếu xuất kho.
- Là quản trị viên, tôi muốn cập nhật thông tin phiếu xuất kho để kiểm soát thông tin phiếu xuất kho.
- Là quản trị viên, tôi muốn xem lịch sử nhập – xuất kho để theo dõi các giao dịch kho trước đó.
- Là quản trị viên, tôi muốn tạo đơn mua hàng mới khi hàng sắp hết để đảm bảo quá trình kinh doanh không bị gián đoạn.
- Là quản trị viên, tôi muốn xem chi tiết thông tin đơn mua hàng để kiểm soát thông tin đơn mua hàng
- Là quản trị viên, tôi muốn cập nhật thông tin đơn mua hàng để kiểm soát thông tin đơn mua hàng

1.6.2. Use case:

- **Use Case** là mô tả chi tiết cách người dùng tương tác với hệ thống để đạt được một mục tiêu cụ thể.

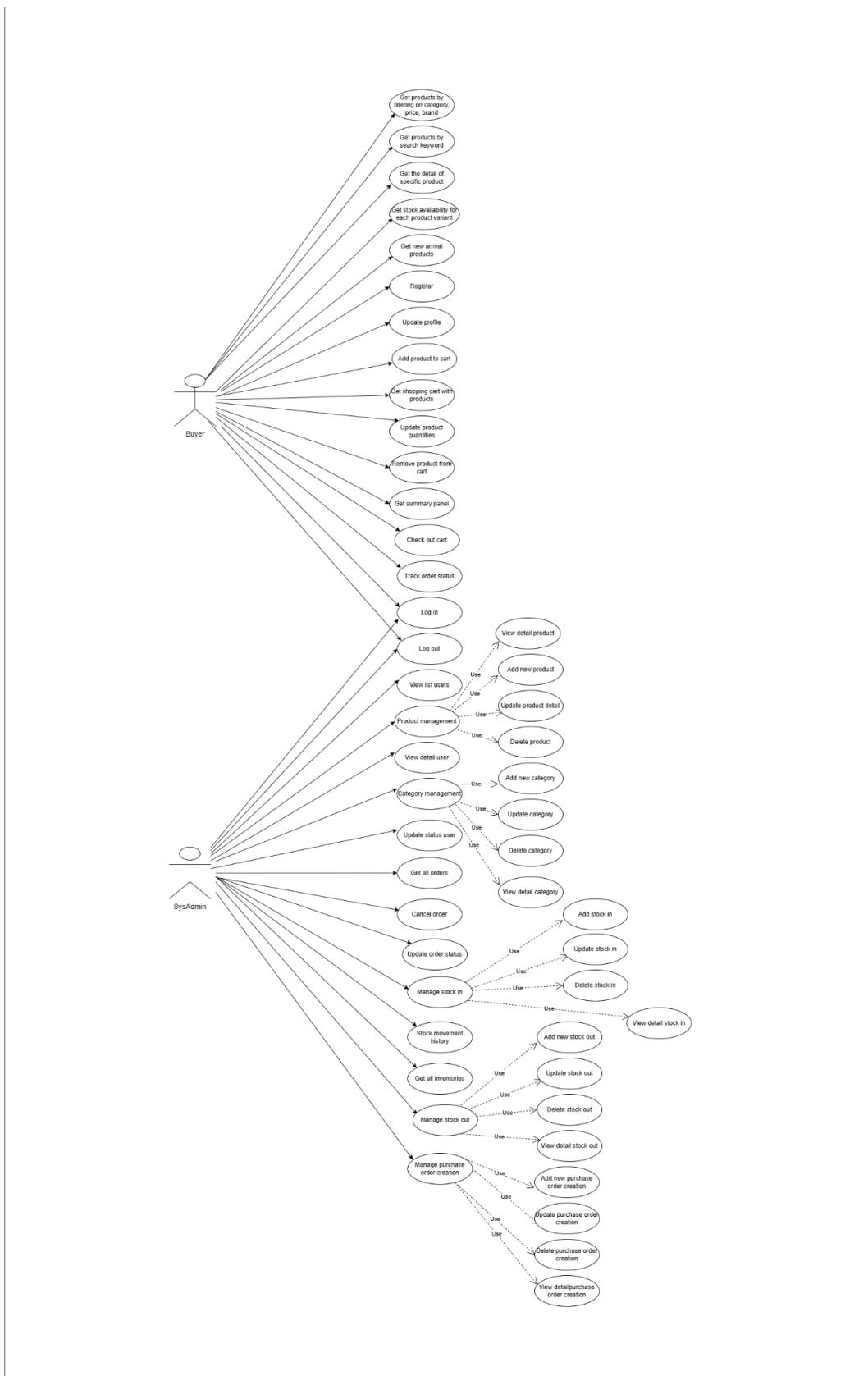
- Cách xác định Use Case:

Xác định Actor Người hoặc hệ thống bên ngoài tương tác với hệ thống

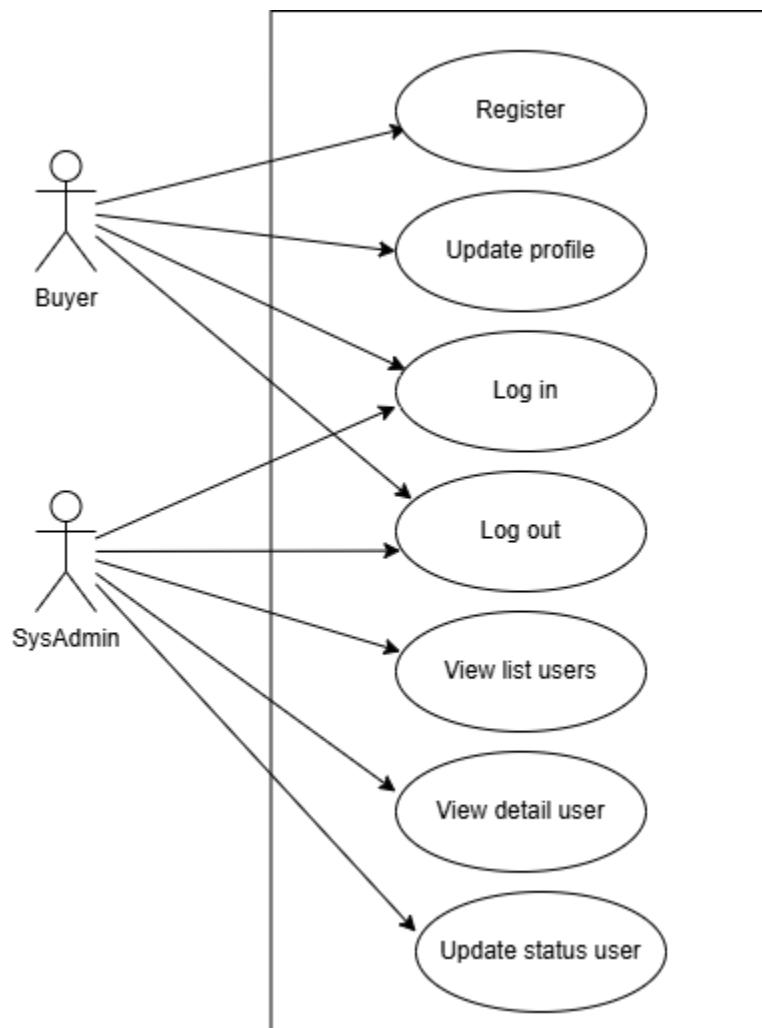
Xác định mục tiêu Actor muốn đạt được

Xác định phạm vi hệ thống

Use Case cho Website ShopEase E-Commerce:



Xác thực & Phân quyền (Authentication & Access Control)



UC1: Xác thực & Phân quyền

Description	Cho phép người mua đăng nhập, đăng ký, đăng xuất, cập nhật tài khoản cá nhân và cho phép quản trị viên đăng nhập, đăng xuất, quản lý người dùng
Pre-condition	<p>Actor: Người mua, quản trị viên</p> <p>Điều kiện: Người mua chưa có tài khoản khi đăng ký. Quản trị viên đã được tạo sẵn tài khoản quản trị. Đăng nhập đúng username, password.</p>
Trigger	<p>Người mua hoặc quản trị viên chọn “Đăng nhập”, Người mua chọn “Đăng ký”, hoặc “Cập nhật thông tin cá nhân” trên giao diện người dùng.</p> <p>Người mua hoặc quản trị viên chọn “Đăng xuất” để kết thúc phiên làm việc.</p>

	Quản trị viên truy cập trang quản trị người dùng để thực hiện thao tác quản lý người dùng.
Basic flow	
	Chức năng bắt đầu khi tác nhân truy cập website. Hệ thống hiển thị form tương ứng và các chức năng cho phép tác nhân thực hiện: Đăng ký: chức năng này cho phép người mua đăng ký tài khoản để thực hiện mua sắm Đăng nhập: chức năng này cho phép tác nhân đăng nhập tài khoản vào hệ thống Cập nhật thông tin cá nhân: chức năng này cho phép người mua chỉnh sửa thông tin cá nhân của chính mình Quản lý người dùng: chức năng này cho phép quản trị viên xem danh sách người dùng và cập nhật trạng thái tài khoản của người dùng.
	a. Đăng ký Luồng đăng ký bắt đầu khi người mua chọn “Đăng ký”. Hệ thống hiển thị form yêu cầu nhập thông tin: <ul style="list-style-type: none">- Tên tài khoản- Mật khẩu- Email- Số điện thoại- Giới tính- Tỉnh thành- Quận huyện- Phường xã- Địa chỉ Người mua nhập thông tin Chọn đăng ký Hệ thống tiến hành kiểm tra thông tin hợp lệ: (Alternative: Khi kiểm tra không hợp lệ) Hệ thống gửi otp qua email cho người mua. Hệ thống hiển thị form cho người mua nhập otp Người mua nhập mã otp. Hệ thống tiến hành kiểm tra mã otp hợp lệ: (Alternative: Khi kiểm tra không hợp lệ) Hệ thống cập nhật thông tin người dùng vào cơ sở dữ liệu và thông báo đăng ký thành công.

b. Đăng nhập

Luồng đăng nhập bắt đầu khi người mua hoặc quản trị viên chọn “Đăng nhập”.

Hệ thống yêu cầu nhập

- Tên tài khoản

- Mật khẩu

Hệ thống tiến hành kiểm tra thông tin hợp lệ

(Alternative: Khi kiểm tra không hợp lệ)

Hệ thống cấp token và điều hướng người dùng:

- Buyer → Trang mua sắm.

- Admin → Trang quản trị.

c. Cập nhật thông tin cá nhân

Luồng cập nhật thông tin cá nhân bắt đầu khi người mua đã đăng nhập chọn “Trang cá nhân”.

Hệ thống hiển thị thông tin hiện tại.

Người dùng chỉnh sửa thông tin:

- Số điện thoại

- Giới tính

- Tỉnh thành

- Quận huyện

- Phường xã

- Địa chỉ

Hệ thống tiến hành kiểm tra thông tin hợp lệ

(Alternative: Khi kiểm tra không hợp lệ)

Hệ thống cập nhật thông tin vào cơ sở dữ liệu

Hệ thống hiển thị thông báo thành công.

d. Quản lý người dùng

Luồng quản lý người dùng bắt đầu khi người quản trị đăng nhập thành công vào trang quản trị.

Hệ thống hiển thị danh sách người dùng.

Quản trị viên chọn:

- Xem chi tiết người dùng:**

Luồng xem chi tiết người dùng bắt đầu khi quản trị viên chọn "xem"

Hệ thống hiển thị chi tiết thông tin người dùng:

- Chỉnh sửa trạng thái tài khoản:**

Luồng xem chi tiết người dùng bắt đầu khi quản trị viên chọn "cập nhật"
Quản trị viên chọn "Đã kích hoạt" hoặc "Chưa kích hoạt" để cập nhật trạng thái tài khoản người dùng

Quản trị viên chọn cập nhật

Hệ thống cập nhật dữ liệu và thông báo cập nhật thành công

e. Đăng xuất

Luồng đăng xuất bắt đầu khi người dùng chọn “Đăng xuất”.

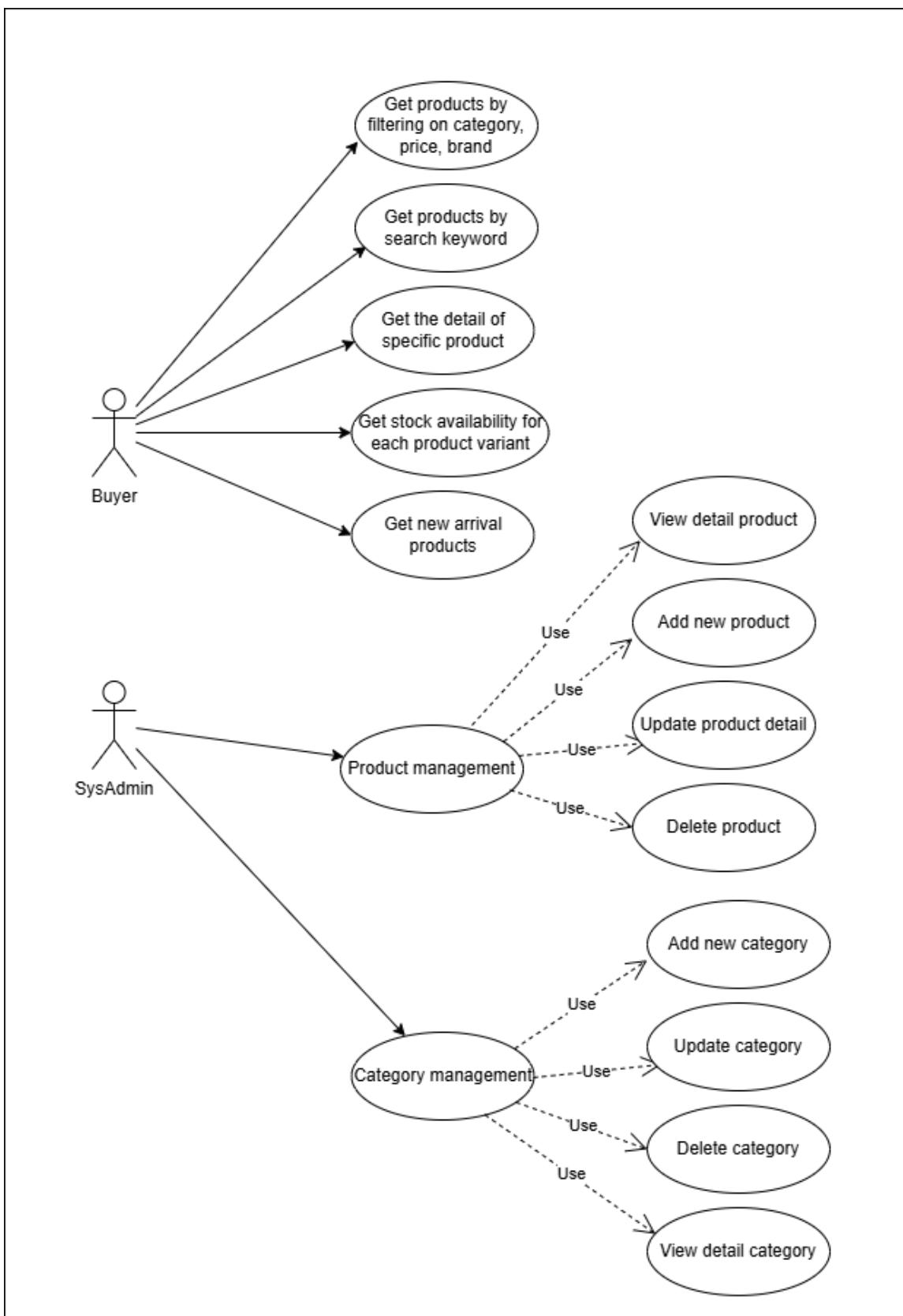
Hệ thống xóa refresh token, kết thúc phiên làm việc.

Hệ thống thông báo đăng xuất tài khoản thành công.

Luồng thay thế

Kiểm tra không hợp lệ: Hệ thống hiển thị thông báo lỗi và cho phép người dùng điền lại thông tin

Danh mục sản phẩm (Product Catalog)



UC1: Danh mục sản phẩm	
Description	Cho phép người mua xem, tìm kiếm, lọc sản phẩm và cho phép quản trị viên quản lý sản phẩm, danh mục sản phẩm
Pre-condition	Actor: Người mua, quản trị viên Điều kiện: Tác nhân đã đăng nhập vào hệ thống.
Trigger	Người mua chọn “Danh mục sản phẩm”, “Tìm kiếm”, “Lọc” hoặc “Xem chi tiết sản phẩm”. Quản trị viên chọn “Quản lý sản phẩm”, “Quản lý danh mục sản phẩm” từ trang quản trị.
Basic flow	
Chức năng bắt đầu khi tác nhân truy cập trang danh mục sản phẩm. Chức năng bắt đầu khi tác nhân truy cập trang danh mục sản phẩm. Hệ thống hiển thị danh sách sản phẩm đang có trong hệ thống và các chức năng cho phép tác nhân thực hiện. Tìm kiếm & lọc sản phẩm: cho phép người mua tìm kiếm sản phẩm theo từ khóa và lọc sản phẩm theo danh mục Xem chi tiết sản phẩm: cho phép người mua xem chi tiết 1 sản phẩm Quản lý sản phẩm/danh mục sản phẩm: cho phép quản trị viên thực hiện xem, thêm, sửa, xóa sản phẩm/danh mục sản phẩm	
<p>a. Tìm kiếm & lọc sản phẩm</p> <p>Luồng tìm kiếm hoặc lọc bắt đầu khi người mua nhập từ khóa tìm kiếm hoặc chọn lọc theo danh mục/giá/thương hiệu</p> <p>Hệ thống truy vấn cơ sở dữ liệu và hiển thị danh sách phù hợp.</p> <p>b. Xem chi tiết sản phẩm</p> <p>Luồng xem chi tiết sản phẩm bắt đầu khi người mua chọn một sản phẩm bất kỳ.</p> <p>Hệ thống hiển thị thông tin chi tiết:</p> <ul style="list-style-type: none"> - Tên sản phẩm - Giá - Mô tả - Hình ảnh - Số lượng - Danh mục - Nhãn hiệu - Phiên bản 	

c. Xem danh sách sản phẩm mới nhất

Luồng xem danh sách sản phẩm mới nhất bắt đầu khi người mua truy cập vào trang chủ

Hệ thống hiển thị danh sách tất cả các sản phẩm mới nhất của cửa hàng

d. Quản lý sản phẩm

Luồng quản lý sản phẩm bắt đầu khi người quản trị chọn "Quản lý sản phẩm" trên trang quản trị.

Hệ thống hiển thị danh sách sản phẩm hiện có cùng các thao tác:

- Thêm mới sản phẩm:

Luồng thêm mới sản phẩm bắt đầu khi người quản trị chọn "Thêm mới"

Hệ thống hiển thị form để nhập thông tin sản phẩm

Nhập thông tin sản phẩm:

- Tên sản phẩm
- Mã sản phẩm
- Slug sản phẩm
- Mô tả ngắn sản phẩm
- Mô tả sản phẩm
- Hình sản phẩm
- Thông số sản phẩm
- Thuộc tính sản phẩm
- Phiên bản sản phẩm
- Thông tin bổ sung

Nhấn nút "Thêm"

Hệ thống tiến hành kiểm tra thông tin hợp lệ:

(Alternative: Khi kiểm tra không hợp lệ)

Hệ thống cập nhật thông tin sản phẩm vào cơ sở dữ liệu và thông báo thêm sản phẩm thành công.

- Xem chi tiết sản phẩm:

Luồng xem chi tiết sản phẩm bắt đầu khi người quản trị chọn "Xem"

Hệ thống hiển thị modal thông tin sản phẩm

- ID
- Ngày tạo
- Ngày cập nhật
- Tên sản phẩm

- Mã sản phẩm
- Slug sản phẩm
- Mô tả ngắn sản phẩm
- Mô tả sản phẩm
- Hình sản phẩm
- Thông số sản phẩm
- Thuộc tính sản phẩm
- Phiên bản sản phẩm
- Khối lượng sản phẩm
- Tên bảo hành sản phẩm

- Chính sửa sản phẩm:

Luồng chỉnh sửa sản phẩm bắt đầu khi người quản trị chọn "Cập nhật"

Hệ thống hiển thị form cập nhật để người quản trị cập nhật thông tin cần cập nhật

Cập nhật thông tin cần cập nhật

Nhấn nút "Cập nhật"

Hệ thống tiến hành kiểm tra thông tin hợp lệ:

(Alternative: Khi kiểm tra không hợp lệ)

Hệ thống cập nhật thông tin sản phẩm vào cơ sở dữ liệu và thông báo cập nhật sản phẩm thành công.

- Xóa sản phẩm:

Luồng xóa sản phẩm bắt đầu khi người quản trị chọn "Xóa"

Hệ thống hiển thị catalog xác nhận xóa sản phẩm

(Alternative: Chọn "không xóa")

Chọn "xóa"

Hệ thống xóa sản phẩm khỏi cơ sở dữ liệu và thông báo xóa sản phẩm thành công

Cập nhật thông tin cần cập nhật

Nhấn nút "Cập nhật"

Hệ thống tiến hành kiểm tra thông tin hợp lệ:

(Alternative: Khi kiểm tra không hợp lệ)

Hệ thống cập nhật thông tin sản phẩm vào cơ sở dữ liệu và thông báo cập nhật sản phẩm thành công.

d. Quản lý danh mục (loại) sản phẩm

Luồng quản lý danh mục sản phẩm bắt đầu khi người quản trị chọn “Quản lý sản phẩm” trên trang quản trị.

Quản trị viên chọn "Danh mục sản phẩm"

Hệ thống hiển thị danh sách danh mục sản phẩm hiện có cùng các thao tác:

- Thêm mới danh mục sản phẩm:

Luồng thêm mới sản phẩm bắt đầu khi người quản trị chọn "Thêm mới"

Hệ thống hiển thị form để nhập thông tin sản phẩm

Nhập thông tin sản phẩm:

- Tên danh mục sản phẩm
- Slug danh mục sản phẩm
- Mô tả danh mục sản phẩm
- Hình đại diện
- Danh mục cha
- Trạng thái danh mục sản phẩm

Nhấn nút "Thêm"

Hệ thống tiến hành kiểm tra thông tin hợp lệ:

(Alternative: Khi kiểm tra không hợp lệ)

Hệ thống thêm thông tin danh mục sản phẩm vào cơ sở dữ liệu và thông báo thêm danh mục sản phẩm thành công.

- Xem chi tiết danh mục sản phẩm:

Luồng xem chi tiết danh mục sản phẩm bắt đầu khi người quản trị chọn "Xem"

Hệ thống hiển thị modal thông tin danh mục sản phẩm

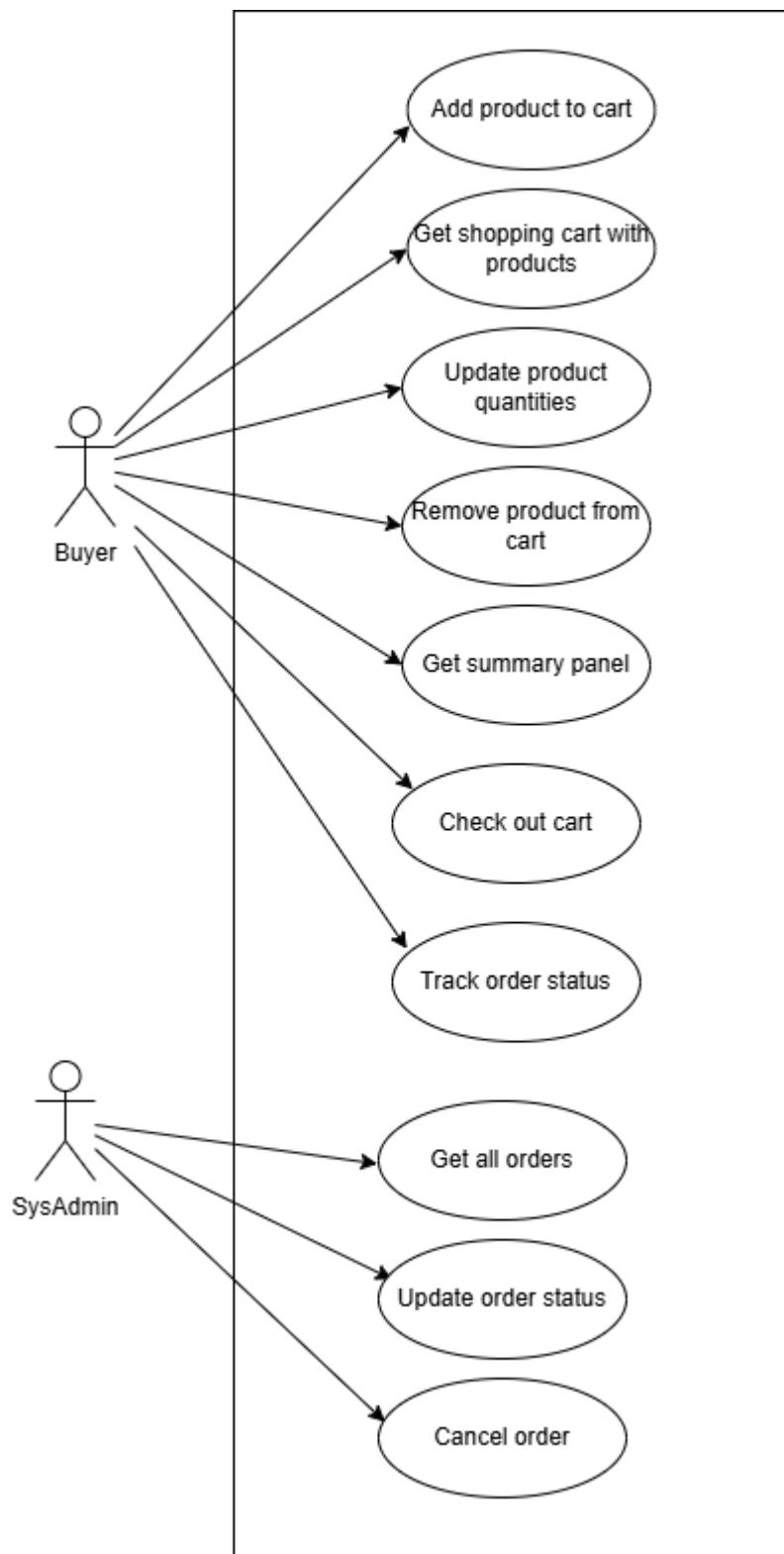
- ID
- Ngày tạo
- Ngày cập nhật
- Tên danh mục sản phẩm
- Slug danh mục sản phẩm
- Mô tả danh mục sản phẩm
- Hình đại diện
- Tên danh mục cha
- Trạng thái danh mục sản phẩm

- Chỉnh sửa danh mục sản phẩm:

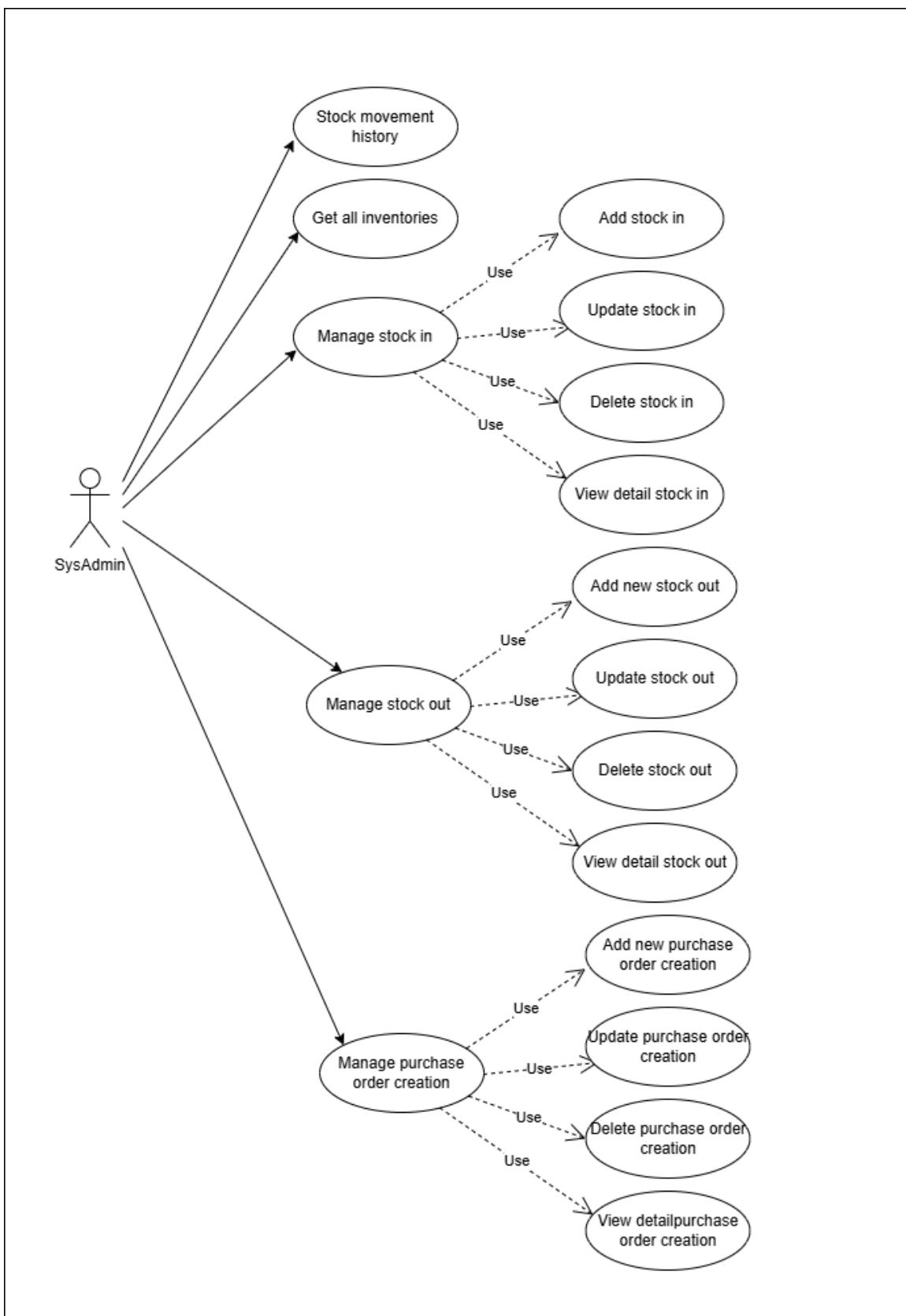
Luồng chỉnh sửa danh mục sản phẩm bắt đầu khi người quản trị chọn "Cập nhật"

<p>Hệ thống hiển thị form cập nhật để người quản trị cập nhật thông tin cần cập nhật</p> <p>Cập nhật thông tin cần cập nhật</p> <p>Nhấn nút "Cập nhật"</p> <p>Hệ thống tiến hành kiểm tra thông tin hợp lệ:</p> <p>(Alternative: Khi kiểm tra không hợp lệ)</p> <p>Hệ thống cập nhật thông tin sản phẩm vào cơ sở dữ liệu và thông báo cập nhật sản phẩm thành công.</p> <p>- Xóa danh mục sản phẩm:</p> <p>Luồng xóa sản phẩm bắt đầu khi người quản trị chọn "Xóa"</p> <p>Hệ thống hiển thị catalog xác nhận xóa danh mục sản phẩm</p> <p>(Alternative: Chọn "không xóa")</p> <p>Chọn "xóa"</p> <p>Hệ thống xóa danh mục sản phẩm khỏi cơ sở dữ liệu và thông báo xóa thành công</p>
<p>Luồng thay thế</p> <p>Kiểm tra không hợp lệ: Hệ thống hiển thị thông báo lỗi và cho phép người dùng điền lại thông tin</p> <p>Chọn không xóa: Hệ thống xác nhận không xóa sản phẩm/loại sản phẩm và ẩn catalog</p>

Giỏ hàng – Đặt hàng – Thanh toán (Shopping Cart, Checkout & Payment)



Quản lý tồn kho (Inventory Management)



1.7. Kế hoạch triển khai dự án:

1.7.1. Cơ sở lý thuyết:

1.7.1.1. Mô hình phát triển phần mềm Agile:

Agile là mô hình phát triển phần mềm linh hoạt, nhấn mạnh việc phản hồi nhanh với thay đổi, phát hành sản phẩm theo từng giai đoạn ngắn, và hợp tác chặt chẽ giữa các thành viên trong nhóm.

Thay vì lập kế hoạch chi tiết toàn bộ dự án ngay từ đầu, Agile chia dự án thành nhiều chu kỳ nhỏ (iteration), trong đó sản phẩm được phát triển, kiểm thử và đánh giá liên tục.

Nguyên tắc cốt lõi của Agile (Agile Manifesto):

- Cá nhân và sự tương tác quan trọng hơn quy trình và công cụ.
- Phần mềm chạy được quan trọng hơn tài liệu đầy đủ.
- Hợp tác với khách hàng quan trọng hơn đàm phán hợp đồng.
- Phản hồi với thay đổi quan trọng hơn bám sát kế hoạch.

1.7.1.2. Mô hình Scrum:

Scrum là một khung làm việc (framework) thuộc họ Agile, tập trung vào việc phát triển sản phẩm theo từng Sprint – mỗi Sprint thường kéo dài từ 1 đến 4 tuần.

Scrum giúp nhóm dễ dàng kiểm soát tiến độ, phản ứng nhanh với thay đổi, và liên tục cải thiện chất lượng sản phẩm.

Các vai trò chính trong Scrum:

- **Product Owner:** Người chịu trách nhiệm xác định nhu cầu và ưu tiên các tính năng (Product Backlog).
- **Scrum Master:** Đảm bảo nhóm tuân thủ quy trình Scrum, hỗ trợ loại bỏ các trở ngại.
- **Development Team:** Nhóm phát triển (Backend, Frontend, QA, UI/UX,...) chịu trách nhiệm hoàn thiện sản phẩm.

Các sự kiện trong Scrum:

Sự kiện	Mục đích	Kết quả
Sprint	Chu kỳ phát triển sản phẩm (1–4 tuần)	Một phần sản phẩm có thể bàn giao
Sprint Planning	Lên kế hoạch cho Sprint (chọn user story để làm)	Sprint Backlog
Daily Scrum	Họp ngắn hằng ngày (15 phút)	Cập nhật tiến độ
Sprint Review	Trình bày kết quả với Product Owner	Phản hồi, đánh giá
Sprint Retrospective	Rút kinh nghiệm sau Sprint	Cải tiến quy trình làm việc

Các tài liệu chính trong Scrum:

Tài liệu	Ý nghĩa
Product Backlog	Danh sách toàn bộ yêu cầu (user stories) của dự án
Sprint Backlog	Danh sách các công việc của Sprint hiện tại
Burndown Chart	Biểu đồ theo dõi tiến độ thực hiện Sprint

1.7.1.3. Lý do chọn mô hình Scrum:

Đề tài có nhiều tính năng nhỏ, liên quan chặt chẽ, cần được phát triển và kiểm thử liên tục như: xem danh sách sản phẩm; quản lý thông tin, danh sách và trạng thái sản phẩm, đặt hàng và thanh toán, theo dõi đơn hàng, nhập/xuất kho, theo dõi tình trạng kho, v.v

Do đó, nhóm chọn Scrum vì:

- Phù hợp với dự án có phạm vi mở, yêu cầu thay đổi linh hoạt.
- Giúp phát hành nhanh từng phần của hệ thống để demo, nhận phản hồi.
- Tăng tính cộng tác giữa các thành viên (backend, frontend, tester).
- Dễ dàng đo lường tiến độ và hiệu quả công việc qua các sprint.

1.7.1.4. Tích hợp và triển khai liên tục (CI/CD):

Continuous Integration (CI): mỗi khi lập trình viên commit code, hệ thống sẽ tự động build và test để phát hiện lỗi sớm.

Continuous Delivery (CD): sau khi build và test thành công, hệ thống tự động triển khai bản mới lên môi trường staging hoặc production.

Lợi ích:

- Giảm lỗi do thao tác thủ công.
- Dễ dàng phát hành bản cập nhật nhanh chóng.
- Bảo đảm sản phẩm luôn ở trạng thái có thể triển khai.

1.7.2. Kế hoạch thực hiện:

1.7.2.1. Phạm vi và mục tiêu:

Hệ thống Shopease được chia thành 4 phân hệ chính:

Phân hệ	Mô tả tóm tắt
1. Product Catalog	Hiển thị danh sách và chi tiết sản phẩm, tìm kiếm, lọc sản phẩm cho người dùng Quản trị viên thêm sản phẩm, cập nhật thông tin, trạng thái sản phẩm

2. Authentication & Authorization	Người dùng đăng ký, đăng nhập, xem thông tin cá nhân, xem thông tin đơn hàng, xem lịch sử đơn hàng Quản trị viên xem danh sách người dùng, xem chi tiết người dùng, cập nhật thông tin chi tiết và trạng thái người dùng
3. Cart / Checkout / Payment	Người dùng thêm sản phẩm vào giỏ hàng, tiến hành thanh toán và theo dõi đơn hàng Quản trị viên xem danh sách và chi tiết đơn hàng, cập nhật trạng thái, xử lý hủy đơn
4. Inventory Management	Quản trị viên xem danh sách tồn kho, tạo đơn mua hàng, nhập kho, xuất kho

1.7.2.2. Cơ sở hạ tầng và môi trường phát triển:

Công việc/ Sản phẩm	Mục đích
PostgreSQL	Hệ quản trị cơ sở dữ liệu
Javascript & Java	Ngôn ngữ sử dụng để phát triển chương trình
VSCode & IntelliJ IDEA	Phần mềm viết code
Docker & Docker Hub	
Github & Github Actions	
Microsoft Word	Phần mềm viết tài liệu
Draw.io & PlanUML	Phần mềm vẽ biểu đồ

1.7.2.3. Lịch trình thực hiện:

Giai đoạn	Nội dung chính	Bắt đầu	Kết thúc	Độ dài (ngày)	Người làm
Khởi tạo dự án	<p>Yêu cầu:</p> <ul style="list-style-type: none"> Xác định tổng quan yêu cầu dự án, mục tiêu dự án, phạm vi dự án Xác định các quy trình nghiệp vụ, ràng buộc nghiệp vụ của khách hàng Xác định các user story và use case cần thiết cho hệ thống <p>Kết quả:</p> <ul style="list-style-type: none"> Tài liệu Mô tả yêu cầu kinh doanh (BRD) 	01/09/2025	15/09/2025	15	Nguyễn Lê Quỳnh Hương

Phân tích	<p>Yêu cầu:</p> <ul style="list-style-type: none"> Tài liệu Mô tả yêu cầu kinh doanh (BRD) được xác nhận và ổn định Phân tích yêu cầu khách hàng thành sơ đồ use case Đặc tả chi tiết từng use case Mô tả giao diện cho từng use case Xác định các nhiệm vụ tổng quát và mốc thời gian chính cho dự án <p>Kết quả:</p> <ul style="list-style-type: none"> Tài liệu Đặc tả yêu cầu phần mềm (SRS): gồm các yêu cầu chức năng/phi chức năng, use case, focused use case, giao diện người dùng, các ràng buộc và giả định Excel tổng hợp các use case Excel tổng hợp các GUI và hướng dẫn sử dụng giao diện Excel Kế hoạch dự án: gồm lịch trình thực hiện, sprint backlog, product backlog, burndown chart 	16/09/2025	05/10/2025	20	Trang Gia Huy Nguyễn Lê Quỳnh Hương
Thiết kế	<p>Yêu cầu:</p> <ul style="list-style-type: none"> Tài liệu Đặc tả yêu cầu phần mềm (SRS) được xác nhận và ổn định Xác định Kiến trúc hệ thống (System Architecture) Xác định Sơ đồ liên lạc (Communication Diagram) Xác định Bản vẽ triển khai (Deployment Diagram) Xác định Mô hình kiến trúc C4 (C4 Model) Thiết kế cơ sở dữ liệu Xác định Kịch bản kiểm thử (Test plan) <p>Kết quả:</p> <ul style="list-style-type: none"> Tài liệu Phân tích thiết kế hệ thống 	06/10/2025	19/10/2025	14	Trang Gia Huy Nguyễn Lê Quỳnh Hương

	<ul style="list-style-type: none"> Tài liệu Kịch bản kiểm thử (Test plan) 				
Sprint 1	<p>Yêu cầu:</p> <ul style="list-style-type: none"> Người dùng xem danh sách sản phẩm, xem thông tin chi tiết sản phẩm, tìm kiếm và lọc sản phẩm Quản trị viên thêm sản phẩm, sửa thông tin chi tiết sản phẩm, cập nhật trạng thái sản phẩm Pipeline CI/CD cơ bản được thiết lập bằng Github Action: <ul style="list-style-type: none"> Tự động build và test backend khi có commit lên branch develop. Push image mới lên Docker Hub sau khi test pass. Deploy bản mới từ Docker Hub lên môi trường staging Thiết lập môi trường staging để demo chức năng thật <p>Kết quả:</p> <ul style="list-style-type: none"> CI/CD pipeline hoạt động tự động Người dùng có thể truy cập và dùng thử module Product Catalog trên môi trường staging 	20/10/2025	26/10/2025	7	Trang Gia Huy Nguyễn Lê Quỳnh Hương

Sprint 2	<p>Yêu cầu:</p> <ul style="list-style-type: none"> Người dùng đăng ký, đăng nhập, xem thông tin cá nhân, xem thông tin đơn hàng, xem lịch sử đơn hàng Quản trị viên xem danh sách người dùng, xem chi tiết người dùng, cập nhật thông tin chi tiết và trạng thái người dùng Tiếp tục sử dụng pipeline CI/CD đã thiết lập: <ul style="list-style-type: none"> Mở rộng pipeline để chạy test tự động cho module Authen Khi module Authen được merge vào branch main, pipeline tự động build và deploy bản mới lên môi trường staging <p>Kết quả:</p> <ul style="list-style-type: none"> CI/CD pipeline vận hành ổn định Người dùng có thể truy cập và dùng các module: Product Catalog, Authen trên môi trường staging 	27/10/2025	02/11/2025	7	Trang Gia Huy Nguyễn Lê Quỳnh Hương
Sprint 3	<p>Yêu cầu:</p> <ul style="list-style-type: none"> Người dùng thêm sản phẩm vào giỏ hàng, tiến hành thanh toán và theo dõi đơn hàng Quản trị viên xem danh sách và chi tiết đơn hàng, cập nhật trạng thái, xử lý hủy đơn Tiếp tục sử dụng pipeline CI/CD đã thiết lập: <ul style="list-style-type: none"> Mở rộng pipeline để chạy test tự động cho module Cart, Checkout & Payment Khi module Cart, Checkout & Payment được merge vào branch main, pipeline tự động build và deploy bản mới lên môi trường staging <p>Kết quả:</p>	03/11/2025	09/11/2025	7	Trang Gia Huy Nguyễn Lê Quỳnh Hương

	<ul style="list-style-type: none"> • CI/CD pipeline vận hành ổn định • Người dùng có thể truy cập và dùng các module: Product Catalog, Authen, Cart, Checkout & Payment trên môi trường staging 				
Sprint 4	<p>Yêu cầu:</p> <ul style="list-style-type: none"> • Quản trị viên xem danh sách tồn kho, tạo đơn mua hàng, nhập kho, xuất kho • Tiếp tục sử dụng pipeline CI/CD đã thiết lập: <ul style="list-style-type: none"> ◦ Mở rộng pipeline để chạy test tự động cho module Inventory ◦ Khi module Inventory được merge vào branch main, pipeline tự động build và deploy bản mới lên môi trường staging ◦ Triển khai trên môi trường production. CI/CD pipeline hỗ trợ cả staging và production. <p>Kết quả:</p> <ul style="list-style-type: none"> • CI/CD pipeline vận hành ổn định, triển khai trên môi trường production • Hệ thống sẵn sàng nghiệm thu và vận hành 	10/11/2025	16/11/2025	7	Trang Gia Huy Nguyễn Lê Quỳnh Hương

Chương 2: Phân tích thiết kế hệ thống:

2.1. Tổng quan kiến trúc hệ thống:

2.1.1. Cơ sở phương pháp luận:

2.1.1.1. Kiến trúc nguyên khối (Monolith Architecture):

Kiến trúc nguyên khối (Monolithic Architecture) là một mô hình thiết kế trong đó toàn bộ các thành phần của hệ thống được **tích hợp và triển khai** như **một ứng dụng duy nhất**.

Tất cả các phần từ giao diện người dùng (UI), xử lý nghiệp vụ (Business Logic), đến truy cập dữ liệu (Data Access) đều nằm trong cùng một khối mã nguồn (codebase) và chạy trong cùng một tiến trình (process).

Trong mô hình này, mọi thành phần đều được đóng gói, biên dịch và triển khai cùng nhau.

Các lớp/tầng gọi trực tiếp lẫn nhau mà không thông qua API hoặc message queue.

2.1.1.2. Kiến trúc ba tầng (N-Tier Architecture):

Kiến trúc ba tầng (Three-Tier Architecture hay N-Tier Architecture) là mô hình thiết kế phân mềm thuộc loại **kiến trúc monolithic** trong đó hệ thống được chia thành ba tầng chức năng chính:

- Tầng trình bày (Presentation Layer)
- Tầng ứng dụng (Application/Business Layer)
- Tầng dữ liệu (Data Layer)

Mỗi tầng đảm nhận một vai trò riêng biệt, giúp tách biệt giao diện người dùng, xử lý nghiệp vụ và lưu trữ dữ liệu

Tầng	Vai trò	Công nghệ lựa chọn
Presentation Layer	Hiển thị giao diện người dùng, nhận và gửi yêu cầu đến tầng ứng dụng.	React
Application Layer	Xử lý logic nghiệp vụ, xác thực, điều phối luồng dữ liệu.	Spring Boot,
Data Layer	Lưu trữ và truy xuất dữ liệu bền vững.	MySQL

Kiến trúc ba tầng (N-Tier Architecture) được lựa chọn sử dụng trong dự án này vì các ưu điểm sau:

- Dễ mở rộng và bảo trì.
- Tách biệt logic nghiệp vụ và giao diện.
- Dễ tích hợp các công nghệ khác nhau ở từng tầng.

2.1.1.3. Mô hình SPA (Single Page Application):

SPA là mô hình ứng dụng web chỉ tải một trang HTML duy nhất và động cập nhật nội dung khi người dùng tương tác, không tải lại toàn bộ trang.

Cơ chế hoạt động:

- Khi người dùng truy cập trang, trình duyệt tải toàn bộ mã JavaScript.
- JavaScript xử lý routing, hiển thị nội dung tương ứng, và gọi API đến backend để lấy dữ liệu dạng JSON.
- Giao diện được cập nhật động mà không tải lại toàn trang.

Mô hình SPA được lựa chọn sử dụng trong dự án này vì những ưu điểm sau:

- Tốc độ phản hồi nhanh, trải nghiệm mượt.
- Giảm tải cho máy chủ vì phần giao diện xử lý phía client.
- Dễ tách biệt frontend và backend (client-server model).

2.1.1.4. Mô hình MVC (Model–View–Controller):

MVC là một mô hình tổ chức mã nguồn phổ biến trong các ứng dụng web, chia hệ thống Backend thành ba phần riêng biệt:

- **Model:** Quản lý dữ liệu và logic nghiệp vụ.
- **View:** Hiển thị thông tin cho người dùng.
- **Controller:** Nhận yêu cầu, xử lý và điều phối dữ liệu giữa Model và View.

Các thành phần được sắp xếp theo tầng kỹ thuật, bao gồm:

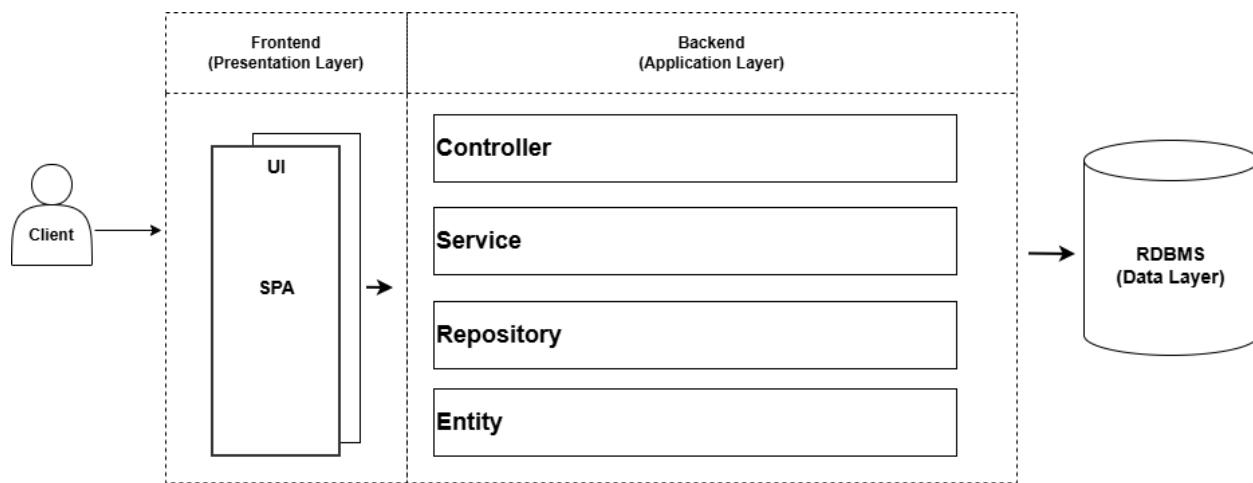
- **Controller:** tiếp nhận và định tuyến các yêu cầu từ client;
- **Service:** xử lý logic nghiệp vụ;
- **Repository:** truy cập dữ liệu;
- **Entity:** đại diện cho các bảng dữ liệu.

Các lớp thuộc các miền nghiệp vụ (product, order, user, ...) được quản lý chung trong cùng tầng, chưa tách thành module độc lập, vì vậy hệ thống thuộc loại kiến trúc nguyên khôi truyền thống, không phải nguyên khôi module (Modular Monolith).

Triển khai trong Spring Boot:

Thành phần	Vai trò trong dự án	Thực thể trong hệ thống
Model	Xử lý nghiệp vụ, lưu trữ dữ liệu	Entity, Repository, Service
View	Hiển thị kết quả cho người dùng	JSON Response trả về SPA
Controller	Định tuyến request, gọi service	Các class Controller

2.1.2. Thiết kế sơ đồ khái:



Hệ thống được thiết kế dựa trên **mô hình nguyên khối (Monolithic Architecture)** kết hợp với **kiến trúc ba tầng (N-Tier Architecture)**.

Trong đó tầng trình bày sử dụng mô hình **ứng dụng đơn trang (SPA – Single Page Application)**, và tầng ứng dụng sử dụng **mô hình MVC (Model–View–Controller)** nhằm đảm bảo tính phân lớp rõ ràng, dễ mở rộng và dễ bảo trì.

1. Tầng trình bày (Presentation Layer)

- Được phát triển bằng ReactJS, hoạt động theo mô hình SPA, nghĩa là toàn bộ giao diện người dùng được tải một lần duy nhất, sau đó nội dung được cập nhật động thông qua các lời gọi API (RESTful API) đến backend mà không cần tải lại toàn bộ trang.
- Tầng này chịu trách nhiệm hiển thị dữ liệu, tương tác với người dùng, và gửi yêu cầu HTTP đến tầng ứng dụng.

2. Tầng ứng dụng (Application Layer)

- Được xây dựng bằng Spring Boot (Java), tuân theo mô hình MVC với bốn lớp chính:
- Controller: Tiếp nhận và xử lý request từ frontend, định tuyến đến các service tương ứng.
- Service: Thực hiện các nghiệp vụ cốt lõi, xử lý logic chính của hệ thống.
- Repository: Giao tiếp với cơ sở dữ liệu thông qua JPA/Hibernate, thực hiện các thao tác CRUD.
- Entity (Model): Đại diện cho các bảng dữ liệu trong cơ sở dữ liệu.

Các chức năng chính được triển khai trong tầng ứng dụng bao gồm:

Nhóm chức năng	Mô tả tóm tắt
1. Product Catalog	Hiển thị danh sách và chi tiết sản phẩm, tìm kiếm, lọc sản phẩm cho người dùng Quản trị viên thêm sản phẩm, cập nhật thông tin, trạng thái sản phẩm

2. Authentication & Authorization	Người dùng đăng ký, đăng nhập, xem thông tin cá nhân, xem thông tin đơn hàng, xem lịch sử đơn hàng Quản trị viên xem danh sách người dùng, xem chi tiết người dùng, cập nhật thông tin chi tiết và trạng thái người dùng
3. Cart / Checkout / Payment	Người dùng thêm sản phẩm vào giỏ hàng, tiến hành thanh toán và theo dõi đơn hàng Quản trị viên xem danh sách và chi tiết đơn hàng, cập nhật trạng thái, xử lý hủy đơn
4. Inventory Management	Quản trị viên xem danh sách tồn kho, tạo đơn mua hàng, nhập kho, xuất kho

Ngoài ra, tầng này có các thành phần hỗ trợ như:

- DTO, Mapper, Constant, Exception Handler, Config, giúp chuẩn hóa dữ liệu, xử lý lỗi, và đảm bảo khả năng mở rộng của hệ thống.

3. Tầng dữ liệu (Data Layer)

- Sử dụng hệ quản trị cơ sở dữ liệu MySQL để lưu trữ toàn bộ dữ liệu của hệ thống.
- Tầng này giao tiếp với tầng ứng dụng thông qua lớp Repository, đảm bảo tính toàn vẹn và an toàn của dữ liệu.

Mỗi quan hệ giữa các tầng được tổ chức theo nguyên tắc phân tách trách nhiệm (Separation of Concerns):

- Frontend (React SPA) chỉ chịu trách nhiệm hiển thị và tương tác.
- Backend (Spring Boot MVC) chịu trách nhiệm xử lý nghiệp vụ và cung cấp API.
- Database (MySQL) chịu trách nhiệm lưu trữ và truy xuất dữ liệu.

Nhờ sự kết hợp giữa N-Tier Architecture, SPA, và MVC, hệ thống đạt được:

- Tính module hóa và dễ bảo trì, do mỗi tầng đảm nhiệm một vai trò riêng biệt.
- Hiệu năng cao, vì giao diện người dùng được cập nhật động thông qua API mà không cần tải lại trang.
- Khả năng mở rộng tốt, khi có thể thay thế hoặc mở rộng từng tầng mà không ảnh hưởng đến toàn bộ hệ thống.

2.2. Communication View:

2.2.1. Cơ sở phương pháp luận:

Communication View (hay còn gọi là Communication Diagram/Component Interaction View) là góc nhìn mô tả cách các thành phần trong hệ thống giao tiếp với nhau, bao gồm:

- Các thành phần (components / modules): ví dụ như frontend, backend, database, service bên thứ ba.
- Luồng giao tiếp (interactions) giữa chúng: qua REST API, HTTP, message queue, database query, v.v.

Communication View giải thích cách các phần của hệ thống nói chuyện với nhau như thế nào, qua giao thức gì, ở mức logic hay vật lý ra sao

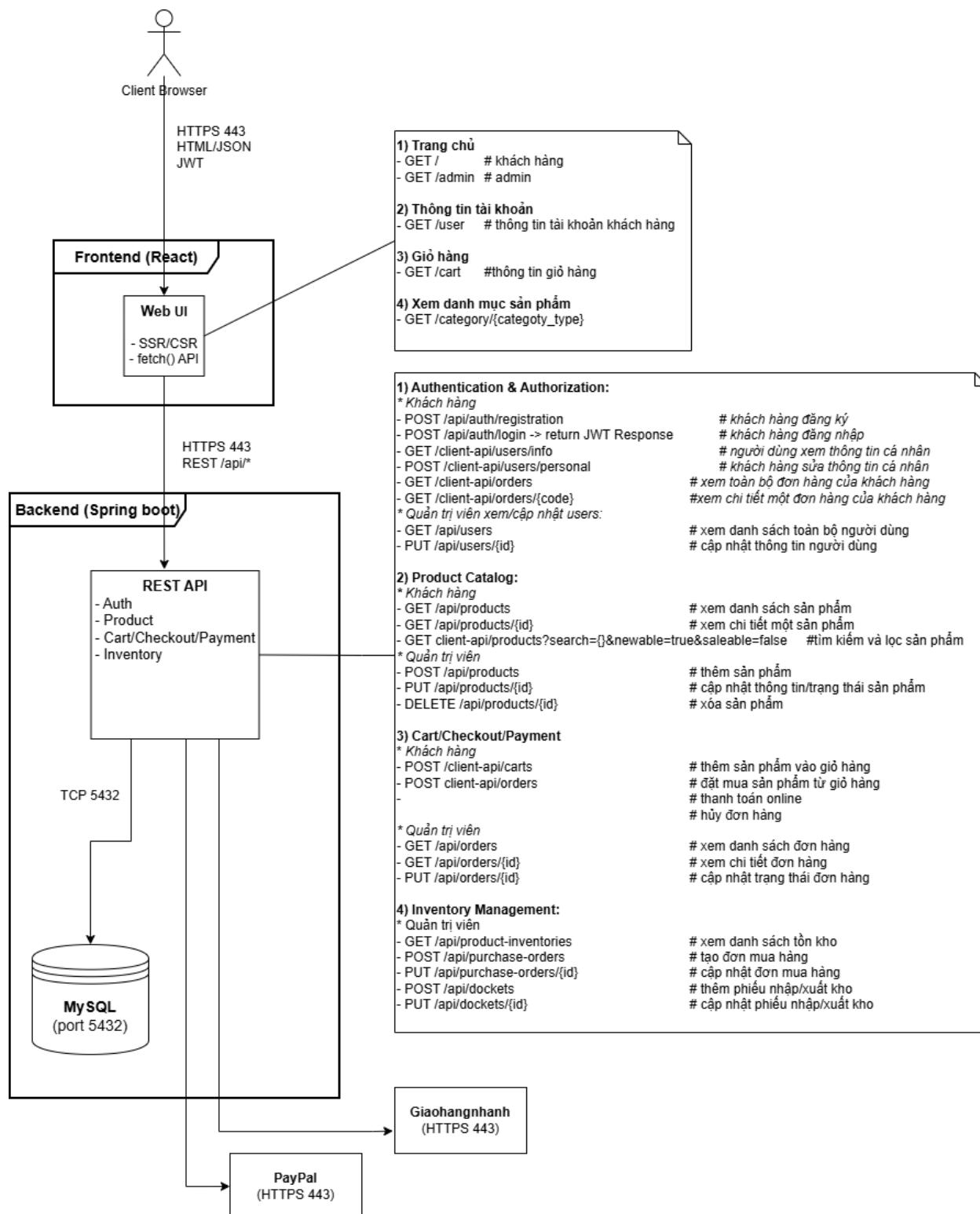
Communication View dùng để:

- Mô tả cách dữ liệu hoặc yêu cầu di chuyển giữa các phần của hệ thống.
- Làm rõ quan hệ phụ thuộc (dependencies) giữa các lớp/tầng hoặc giữa hệ thống với bên thứ ba.
- Giúp lập trình viên, tester và kiến trúc sư hệ thống hiểu rõ các luồng tương tác chính, phục vụ cho triển khai và kiểm thử tích hợp.

Cách xác định Communication View:

1. Xác định các thành phần (components) trong hệ thống (frontend, backend, DB, external service,...).
2. Xác định các kênh giao tiếp giữa chúng (HTTP, REST API, JDBC, gRPC,...).
3. Xác định các luồng dữ liệu hoặc hành động tiêu biểu, ví dụ:
4. Người dùng gửi request từ web UI → Backend xử lý → Ghi/đọc DB → Trả kết quả về frontend.
5. Backend gọi API bên thứ ba (PayPal, GiaoHangNhanh,...).
6. Vẽ sơ đồ thể hiện các thành phần và mũi tên tương tác

2.2.2. Thiết kế sơ đồ:



Communication View của hệ thống mô tả cách các thành phần chính trao đổi dữ liệu trong một kiến trúc nguyên khôi ba tầng:

- Frontend (React): tầng giao diện người dùng (Presentation Layer).
- Backend (Spring Boot): tầng nghiệp vụ và xử lý dữ liệu (Application Layer + Business Logic).
- Database (MySQL): tầng lưu trữ dữ liệu (Data Layer).
- Dịch vụ bên thứ ba: PayPal (xử lý thanh toán) và GiaoHangNhanh (vận chuyển).

Các luồng giao tiếp chính:

1. Frontend ↔ Backend:

Frontend (React) gửi các yêu cầu HTTP (fetch API) đến Backend (Spring Boot REST API).

Các endpoint chính được chia theo nhóm:

- /api/auth/... — xác thực, đăng nhập, quản lý người dùng.
- /api/products/... — hiển thị, tìm kiếm, thêm/xóa sản phẩm.
- /api/orders/... — đặt hàng, thanh toán, xem lịch sử đơn hàng.
- /api/inventories/... — quản lý tồn kho, nhập/xuất hàng.

Giao tiếp này diễn ra qua HTTP (port 8080), theo mô hình client–server.

2. Backend ↔ Database (MySQL):

Backend giao tiếp với MySQL qua JDBC (port 5432).

Các Repository (JPA/Hibernate) đảm nhận truy vấn dữ liệu cho các entity: User, Product, Order, Inventory.

3. Backend ↔ Dịch vụ bên thứ ba

Giao tiếp với PayPal API (HTTPS 443) để xử lý thanh toán trực tuyến.

Giao tiếp với GiaoHangNhanh API (HTTPS 443) để tạo vận đơn và tra cứu trạng thái giao hàng.

2.3. Deployment View:

2.3.1. Cơ sở phương pháp luận:

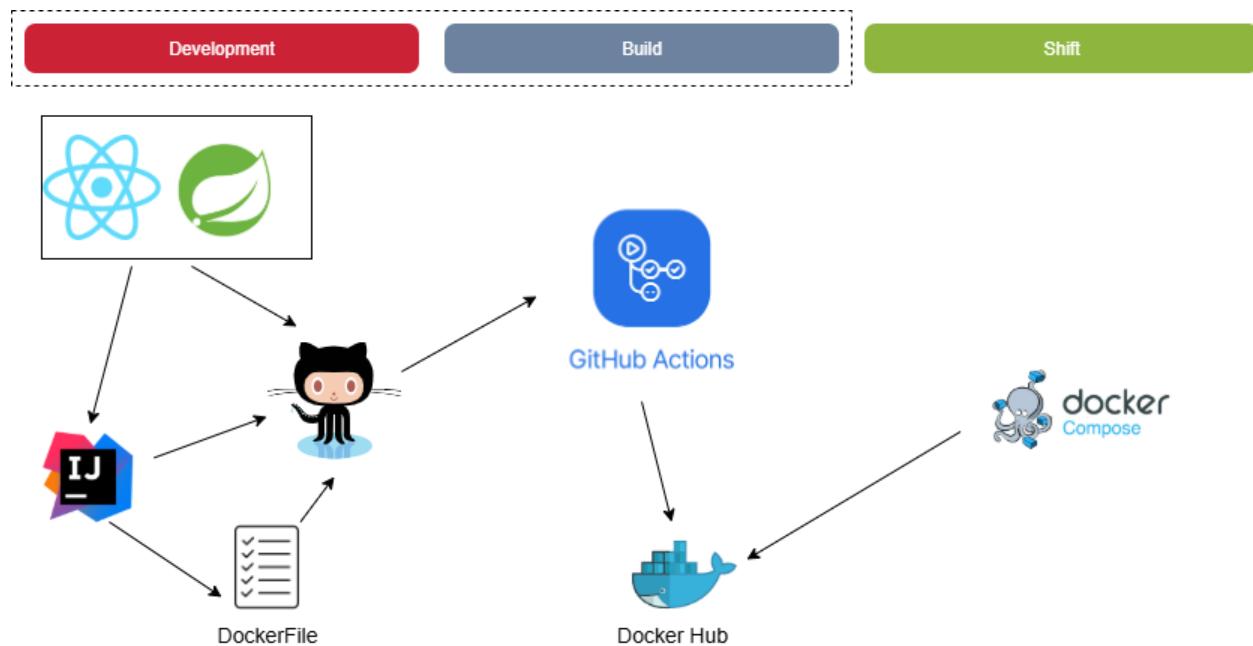
Deployment View (hay còn gọi là Physical View trong mô hình kiến trúc 4+1) là góc nhìn mô tả cách phần mềm được triển khai trên hạ tầng phần cứng hoặc môi trường vận hành thực tế.

Deployment View giải thích hệ thống được triển khai ở đâu, như thế nào, và các thành phần chạy ở môi trường nào.

Mục tiêu của Deployment View gồm:

- Mô tả các nút vật lý (physical nodes) hoặc môi trường triển khai
- Thể hiện mối liên kết giữa các thành phần phần mềm và hạ tầng mà chúng được triển khai lên.
- Cho thấy quy trình build, đóng gói và triển khai phần mềm từ giai đoạn phát triển đến khi vận hành thực tế.
- Là cơ sở cho việc tự động hóa triển khai (CI/CD) và quản lý vòng đời phần mềm.

2.3.2. Thiết kế sơ đồ:



Giai đoạn	Trong CI/CD	Giải thích
Development	Trước Build	Giai đoạn code và commit lên GitHub. Sử dụng IDE (IntelliJ) để viết code React + Spring Boot.
Build	Build + Test	GitHub Actions build ứng dụng, chạy test, và tạo Docker image.
Shift	Deploy	Image được đẩy lên Docker Hub, rồi triển khai bằng Docker Compose.

Deployment View mô tả quy trình triển khai hệ thống từ giai đoạn phát triển đến khi vận hành thực tế, trong bối cảnh dự án được phát triển theo mô hình Scrum kết hợp với CI/CD pipeline.

Toàn bộ quy trình được tự động hóa thông qua GitHub Actions và Docker, giúp đảm bảo tính liên tục, đồng nhất và đáng tin cậy của quá trình triển khai.

Giai đoạn 1: Development

Ở giai đoạn phát triển (Development), nhóm lập trình viên thực hiện phát triển song song hai thành phần:

Frontend: Ứng dụng React (SPA – Single Page Application).

Backend: Ứng dụng Spring Boot (RESTful API).

Cả hai được xây dựng và quản lý mã nguồn trên GitHub, với môi trường phát triển chính là IntelliJ IDEA.

Trong giai đoạn này, các tệp Dockerfile được định nghĩa sẵn để mô tả cách build và cấu hình môi trường chạy cho từng thành phần.

Khi hoàn thành các User Story trong sprint, lập trình viên commit và push mã nguồn lên GitHub, khởi động quy trình CI/CD tự động.

Giai đoạn 2: Build & Test (Continuous Integration)

Sau khi có thay đổi mới trên repository, GitHub Actions tự động kích hoạt quy trình CI (Continuous Integration).

Các bước chính bao gồm:

1. Build code: GitHub Actions tự động compile và build ứng dụng backend (Spring Boot) và frontend (React) theo Dockerfile.
2. Run test: Hệ thống chạy các bài kiểm thử tự động (unit test, integration test) nhằm đảm bảo chất lượng mã nguồn.
3. Build Docker image: Nếu build và test đều thành công, Docker image sẽ được tạo ra từ source code.
4. Push image lên Docker Hub: Image được lưu trữ trên Docker Hub, nơi đóng vai trò kho lưu trữ trung tâm cho các container image đã sẵn sàng triển khai.

Giai đoạn này đảm bảo rằng mọi thay đổi trên Git đều được tích hợp, kiểm thử và đóng gói tự động, loại bỏ rủi ro do thao tác thủ công và đảm bảo tính ổn định giữa các môi trường.

Giai đoạn 3: Deploy (Continuous Deployment)

Tại giai đoạn này, các Docker image được triển khai (Deploy) sang môi trường chạy thực tế bằng Docker Compose.

Docker Compose đảm nhận việc:

- Kéo (pull) các image mới nhất từ Docker Hub.
- Khởi tạo và chạy các container cho từng service (Frontend React, Backend Spring Boot, Database MySQL).
- Thiết lập cấu hình mạng, biến môi trường, và mapping port giữa các container.

Quá trình này giúp việc triển khai trở nên tự động, đồng nhất và có thể lặp lại dễ dàng.

Nhờ Docker, nhóm phát triển và nhóm vận hành (DevOps) có thể đảm bảo rằng ứng dụng sẽ chạy nhất quán trên mọi môi trường (từ máy cá nhân đến máy chủ thật).

2.4. Decomposition View C4:

C4 Model là mô hình trực quan mô tả kiến trúc phần mềm ở nhiều mức độ chi tiết khác nhau — từ tổng quan toàn hệ thống đến mã nguồn cụ thể.

Được đề xuất bởi Simon Brown, mô hình này chia kiến trúc thành 4 mức (C1–C4), mỗi mức trả lời một câu hỏi khác nhau:

Mức	Tên gọi	Câu hỏi trả lời	Mục tiêu
C1	Context Diagram	Hệ thống này nằm trong bối cảnh nào?	Cho người xem “bức tranh toàn cảnh” – ai tương tác với hệ thống và vì mục đích gì.
C2	Container Diagram	Hệ thống được chia thành các phần (container) nào?	Thể hiện các thành phần triển khai chính: web app, database, API, v.v.
C3	Component Diagram	Bên trong mỗi container có những thành phần chức năng nào?	Giúp nhà phát triển hiểu cấu trúc logic nội bộ.
C4	Code Diagram	Cụ thể thành phần đó được triển khai bằng gì?	Dành cho lập trình viên – thể hiện cấu trúc lớp, module, package.

2.4.1. Sơ đồ C1 - System Context:

2.4.1.1. Cơ sở lý thuyết:

Thể hiện bức tranh tổng thể về hệ thống trong môi trường hoạt động của nó: mô tả mối quan hệ giữa hệ thống và các tác nhân bên ngoài (người dùng, hệ thống khác, dịch vụ tích hợp).

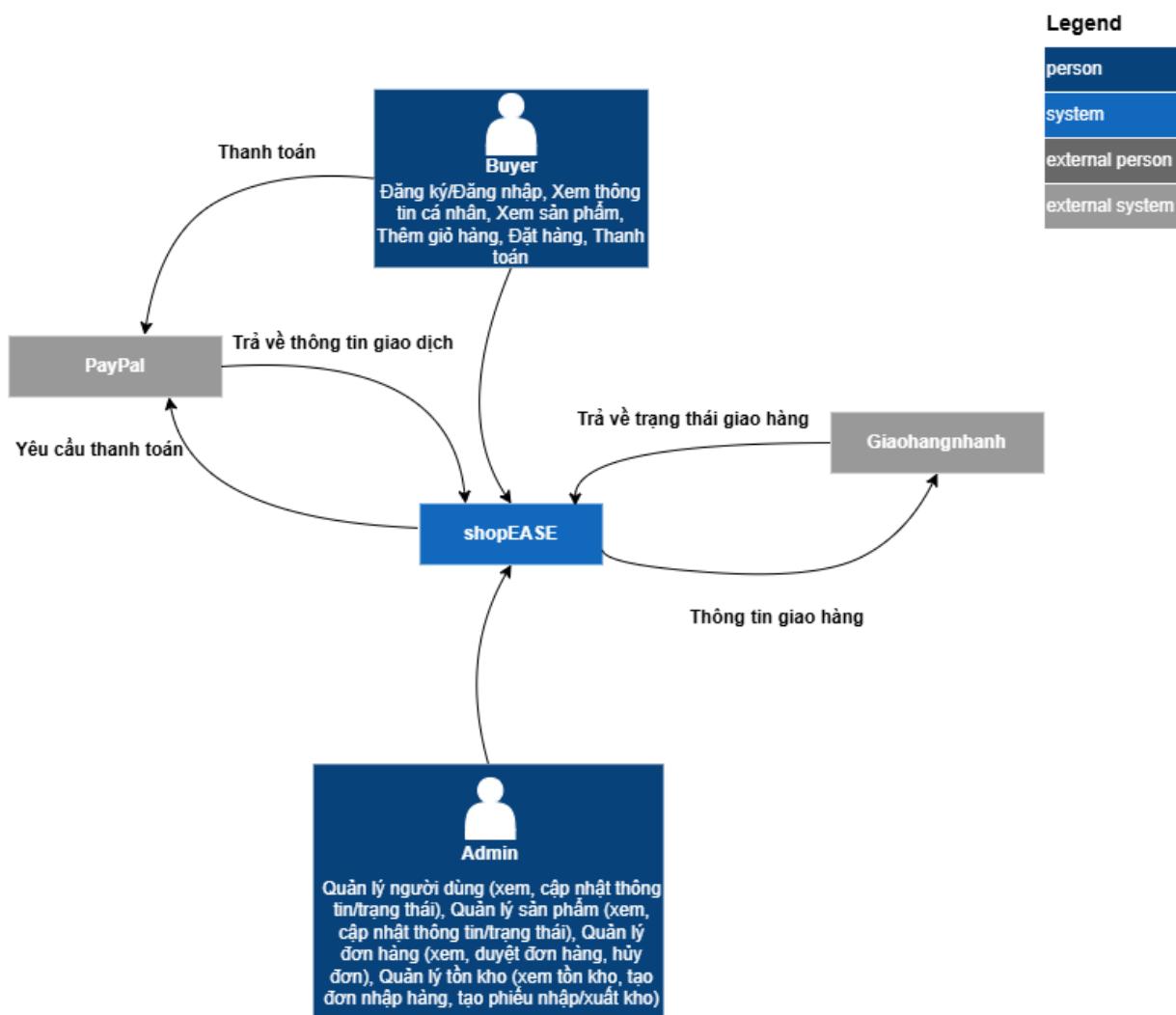
Giúp người đọc (nhất là stakeholder phi kỹ thuật) hiểu:

- Hệ thống đang giải quyết vấn đề gì;
- Ai sử dụng nó;
- Nó tương tác với những hệ thống nào khác.

Cách xác định:

- Xác định các actor chính (user, admin, partner system).
- Xác định hệ thống trung tâm (System Under Design).
- Vẽ các mối quan hệ (flow of data hoặc interaction) giữa actor ↔ system ↔ external system.

2.4.1.2. Thiết kế sơ đồ:



Tóm lược:

Sơ đồ ngữ cảnh C1 cho thấy hệ thống shopEASE hoạt động như một trung tâm điều phối thương mại điện tử, kết nối giữa người mua, nhà quản trị, cổng thanh toán PayPal và đơn vị vận chuyển GiaoHangNhanh.

Mô hình này giúp làm rõ phạm vi hệ thống, ranh giới chức năng, và luồng tương tác chính, là nền tảng cho việc phân rã chi tiết trong các mức C2 (Container) và C3 (Component) tiếp theo.

Trong sơ đồ, hệ thống nằm ở trung tâm và tương tác với các tác nhân bên ngoài như sau:

1. Buyer (Người mua hàng):

Là người dùng chính của hệ thống.

Thực hiện các chức năng: đăng ký, đăng nhập, xem và cập nhật thông tin cá nhân, duyệt danh mục sản phẩm, thêm sản phẩm vào giỏ hàng, đặt hàng và thanh toán.

Gửi yêu cầu thanh toán qua hệ thống shopEASE.

Nhận phản hồi về trạng thái giao hàng và thông tin thanh toán từ hệ thống.

2. Admin (Quản trị viên):

Quản trị viên chịu trách nhiệm vận hành và kiểm soát dữ liệu của hệ thống.

Các chức năng chính gồm:

- Quản lý người dùng (xem, cập nhật thông tin).
- Quản lý sản phẩm (thêm, sửa, cập nhật trạng thái).
- Quản lý đơn hàng (duyệt, hủy, cập nhật trạng thái).
- Quản lý tồn kho và lập phiếu nhập/xuất.

Tương tác trực tiếp với hệ thống shopEASE để thực hiện các nghiệp vụ quản trị.

3. PayPal (Hệ thống thanh toán bên ngoài):

Là external system được tích hợp để xử lý thanh toán trực tuyến.

Hệ thống shopEASE gửi yêu cầu thanh toán đến PayPal, sau đó PayPal trả về thông tin giao dịch (thành công hoặc thất bại).

Giao tiếp được thực hiện qua API bảo mật sử dụng giao thức HTTPS.

4. GiaoHangNhanh (Hệ thống vận chuyển):

Là external system đảm nhiệm chức năng vận chuyển hàng hóa.

Hệ thống shopEASE gửi thông tin giao hàng (địa chỉ, mã đơn, sản phẩm, trọng lượng, v.v.) đến GiaoHangNhanh.

Nhận lại trạng thái giao hàng (đã tiếp nhận, đang giao, hoàn thành hoặc hủy).

2.4.2. Sơ đồ C2 – Container:

2.4.2.1. Cơ sở lý thuyết:

Cho biết hệ thống được chia thành những container nào, mỗi container là một phần có thể triển khai độc lập (chẳng hạn web app, mobile app, database, API service...).

Giúp hiểu kiến trúc triển khai logic và công nghệ chính được dùng, như:

- Các thành phần chạy độc lập (backend, frontend, DB, external service).
- Giao tiếp giữa chúng (REST API, message queue, DB connection).

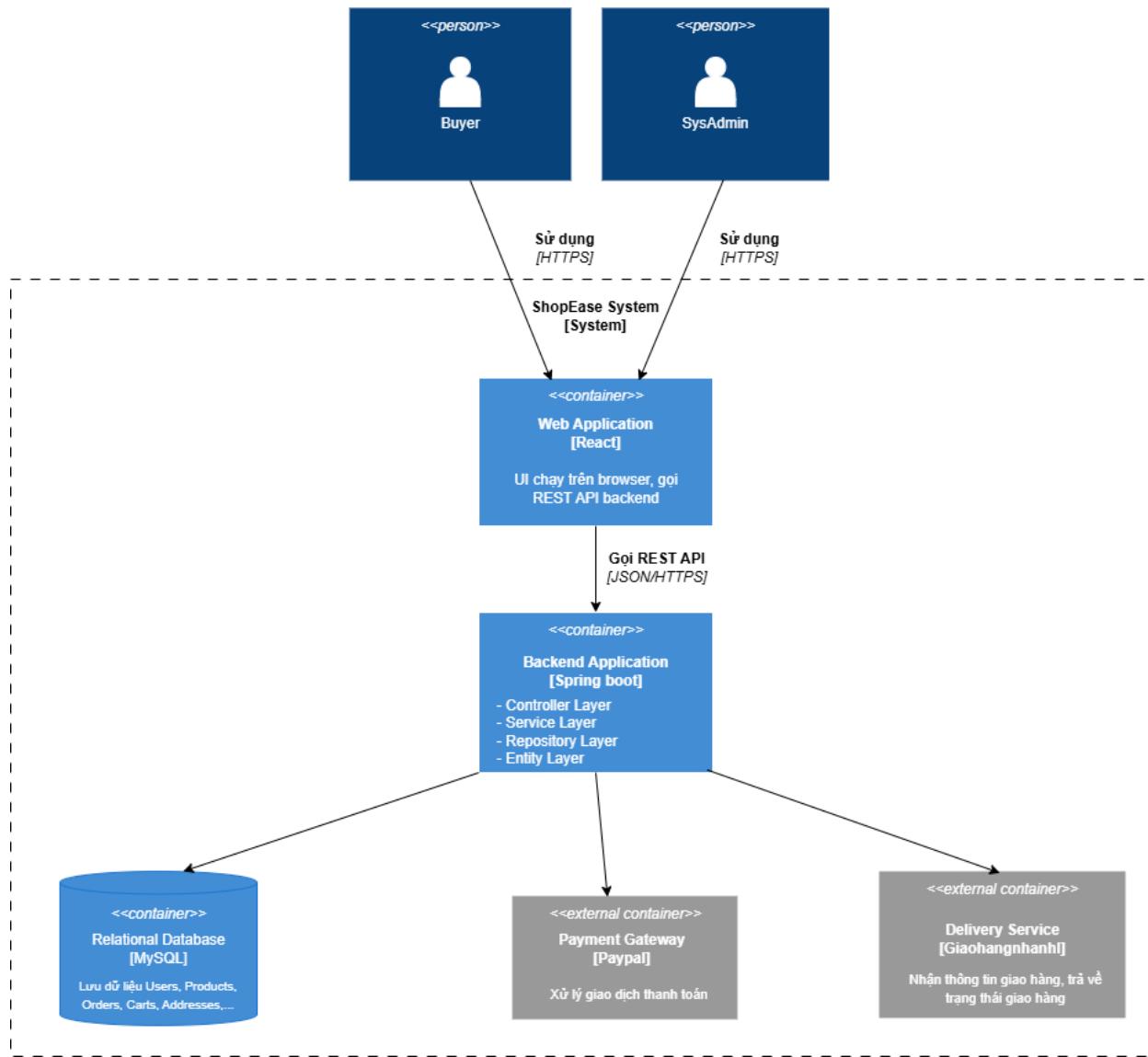
Cách xác định:

- Phân tách hệ thống theo ràng buộc triển khai hoặc trách nhiệm chức năng.

- Mỗi container có thể tương ứng với:

- Một ứng dụng triển khai riêng biệt (Spring Boot app, React frontend),
- Một dịch vụ nền (background worker),
- Một database

2.4.2.2. Thiết kế sơ đồ:



Sơ đồ container C2 của shopEASE cho thấy hệ thống được xây dựng theo **kiến trúc nguyên khối (Monolithic Architecture)** kết hợp với **kiến trúc ba tầng (N-Tier Architecture)** và **mô hình MVC**. Các thành phần chính gồm:

1. Web Application [React]:

Vai trò: Là giao diện người dùng (UI) của hệ thống, chạy trên trình duyệt.

Chức năng:

- Cho phép Buyer và SysAdmin tương tác trực tiếp với hệ thống qua giao diện web.
- Thực hiện các thao tác như xem danh sách sản phẩm, giỏ hàng, đặt hàng, quản lý đơn hàng, người dùng, tồn kho, v.v.

Giao tiếp:

- Gửi và nhận dữ liệu từ Backend Application thông qua REST API sử dụng giao thức HTTPS/JSON.
- Toàn bộ phần giao diện được xây dựng theo mô hình Single Page Application (SPA), giúp trải nghiệm người dùng mượt mà.

2. Backend Application [Spring Boot]

Vai trò: Xử lý toàn bộ nghiệp vụ, logic hệ thống và truy cập dữ liệu.

- Được tổ chức theo mô hình MVC và phân lớp rõ ràng:
- Controller Layer: Tiếp nhận yêu cầu từ frontend, ánh xạ API.
- Service Layer: Xử lý nghiệp vụ chính (business logic).
- Repository Layer: Tương tác với cơ sở dữ liệu qua JPA/Hibernate.
- Entity Layer: Biểu diễn các đối tượng dữ liệu

Giao tiếp:

- Kết nối đến CSDL MySQL để lưu trữ dữ liệu.
- Gọi API đến PayPal để xử lý thanh toán.
- Gọi API đến GiaoHangNhanh để quản lý thông tin giao hàng và trạng thái vận chuyển.

3. Relational Database [MySQL]:

Vai trò: Lưu trữ dữ liệu cốt lõi của hệ thống, bao gồm:

- Thông tin người dùng, sản phẩm, đơn hàng, giỏ hàng, tồn kho, địa chỉ, trạng thái giao hàng, v.v.
- Backend kết nối với CSDL thông qua JPA Repository sử dụng MySQL Connector (JDBC).

4. External Container – Payment Gateway [PayPal]:

Là hệ thống bên ngoài (external service) dùng để xử lý thanh toán trực tuyến.

Khi người dùng đặt hàng và chọn phương thức thanh toán, backend sẽ:

- Gửi yêu cầu thanh toán đến PayPal (HTTPS API).
- Nhận kết quả giao dịch (thành công/thất bại) và lưu vào hệ thống.

5. External Container – Delivery Service [GiaoHangNhanh]:

Là dịch vụ bên ngoài dùng để quản lý giao nhận đơn hàng.

Backend gửi thông tin giao hàng đến GiaoHangNhanh.

Nhận lại trạng thái giao hàng (đã nhận, đang giao, hoàn thành hoặc hủy).

2.4.3. Sơ đồ C3 – Component:

2.4.3.1. Cơ sở lý thuyết:

Ở mức C3, ta mô tả các thành phần phần mềm (components) trong từng container.

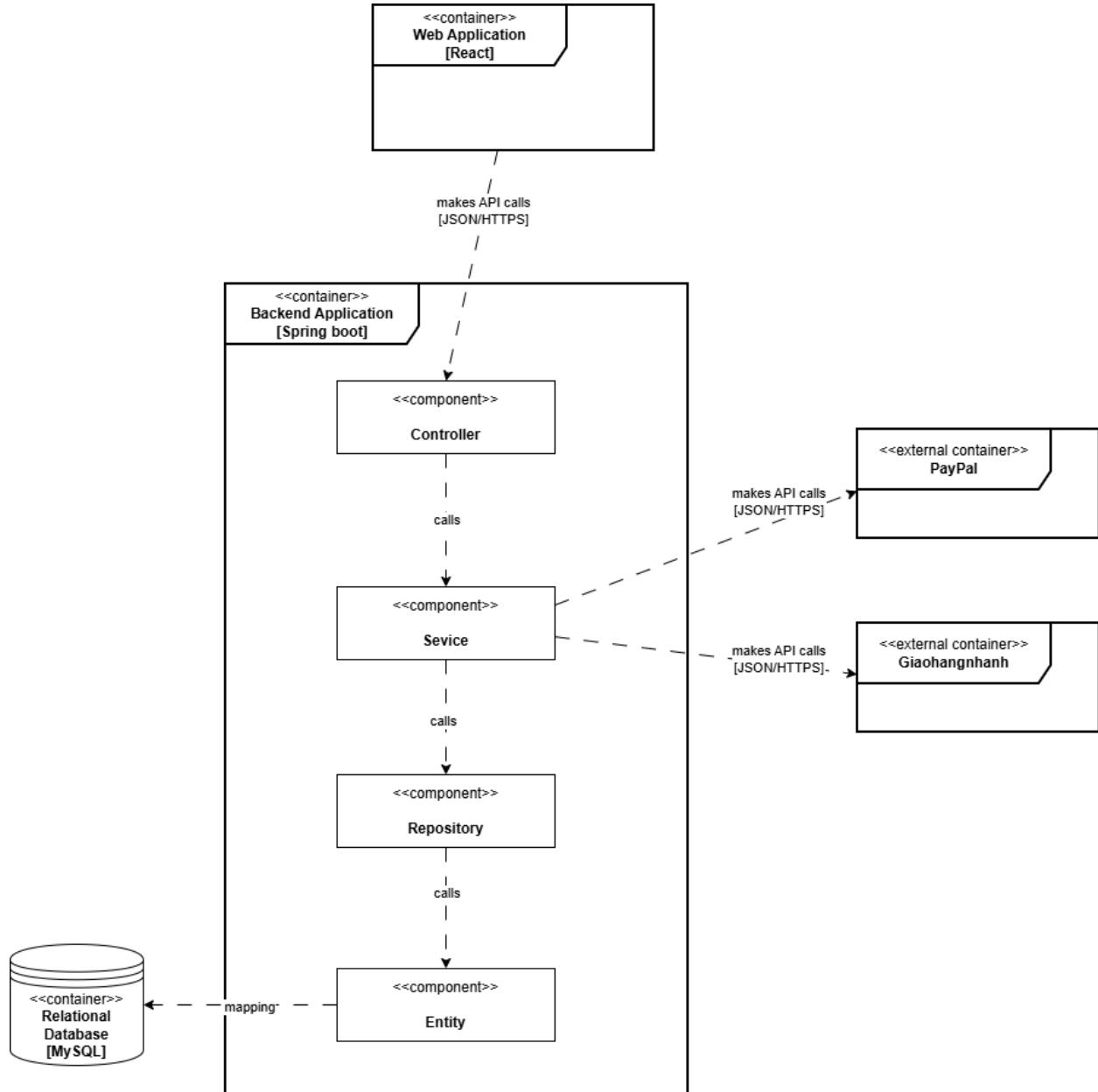
Với backend của hệ thống này, các component chính là các layer kỹ thuật (controller, service, repository, entity).

Xem mỗi layer như một thành phần độc lập có trách nhiệm riêng.

Component	Mô tả
Controller Layer	Tiếp nhận và xử lý yêu cầu HTTP từ client (React). Ánh xạ các endpoint REST tới các service tương ứng.
Service Layer	Chứa logic nghiệp vụ, điều phối giữa controller và repository.
Repository Layer	Giao tiếp với cơ sở dữ liệu (MySQL) thông qua Spring Data JPA.
Entity Layer	Đại diện cho các bảng dữ liệu
External Integration	Gọi API bên ngoài như PayPal (xử lý thanh toán) và GiaoHangNhanh (theo dõi vận chuyển).

2.4.3.2. Thiết kế sơ đồ:

High-Level:



Sơ đồ C3 – Component Diagram mô tả chi tiết các thành phần bên trong container Backend Application (Spring Boot).

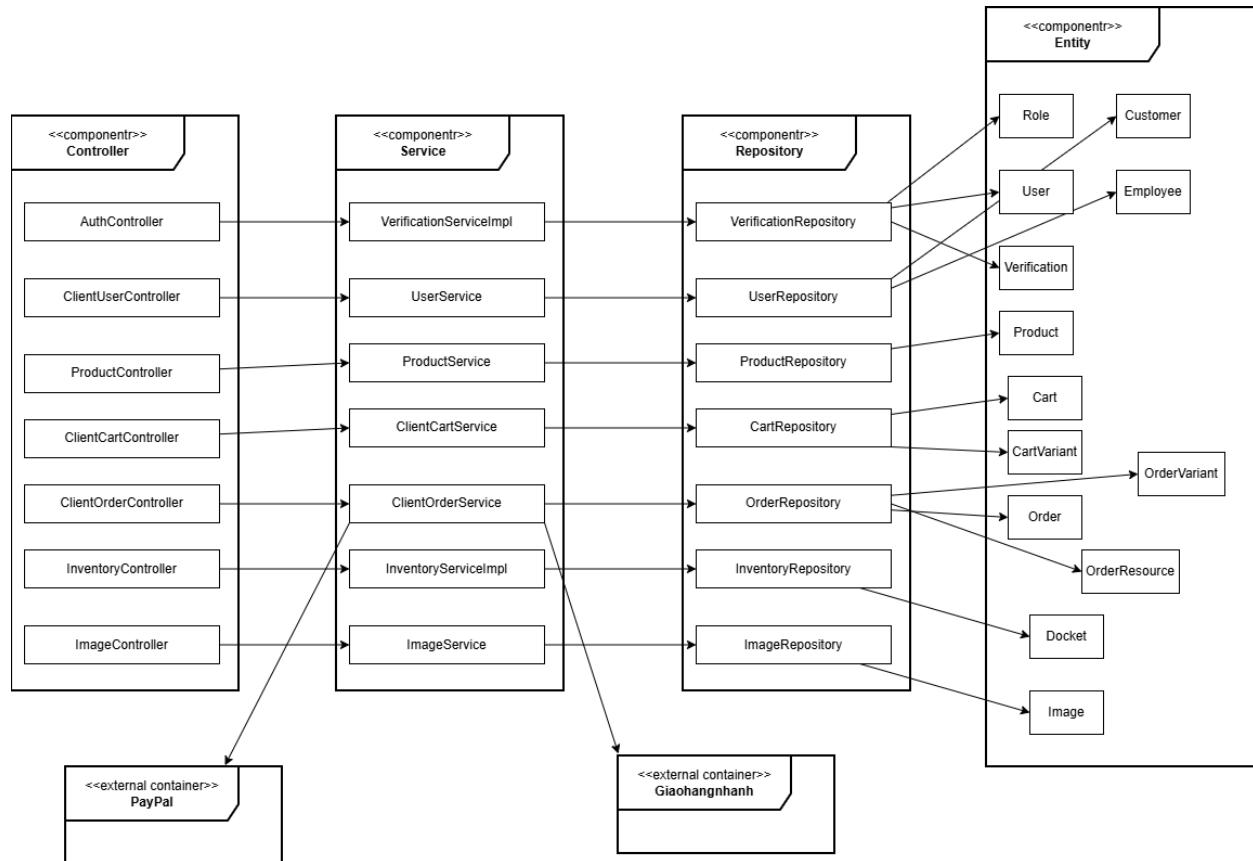
Ứng dụng được chia thành 4 tầng chính gồm:

- Controller layer: xử lý yêu cầu từ client và định tuyến các lời gọi API.
- Service layer: triển khai logic nghiệp vụ, đồng thời tích hợp với các hệ thống bên ngoài như PayPal (xử lý thanh toán) và GiaoHangNhanh (quản lý vận chuyển).
- Repository layer: cung cấp các thao tác truy cập dữ liệu

- Entity layer: ánh xạ các đối tượng trong mã nguồn với các bảng trong cơ sở dữ liệu.

Sơ đồ cũng thể hiện mối liên kết giữa backend và các thành phần khác: frontend (React), cơ sở dữ liệu MySQL, cùng các dịch vụ bên ngoài. Luồng dữ liệu được truyền qua giao thức HTTPS và định dạng JSON, đảm bảo tính bảo mật và dễ tích hợp.

Module-Level:



2.4.4. Sơ đồ C4 - Code/Implementation Level:

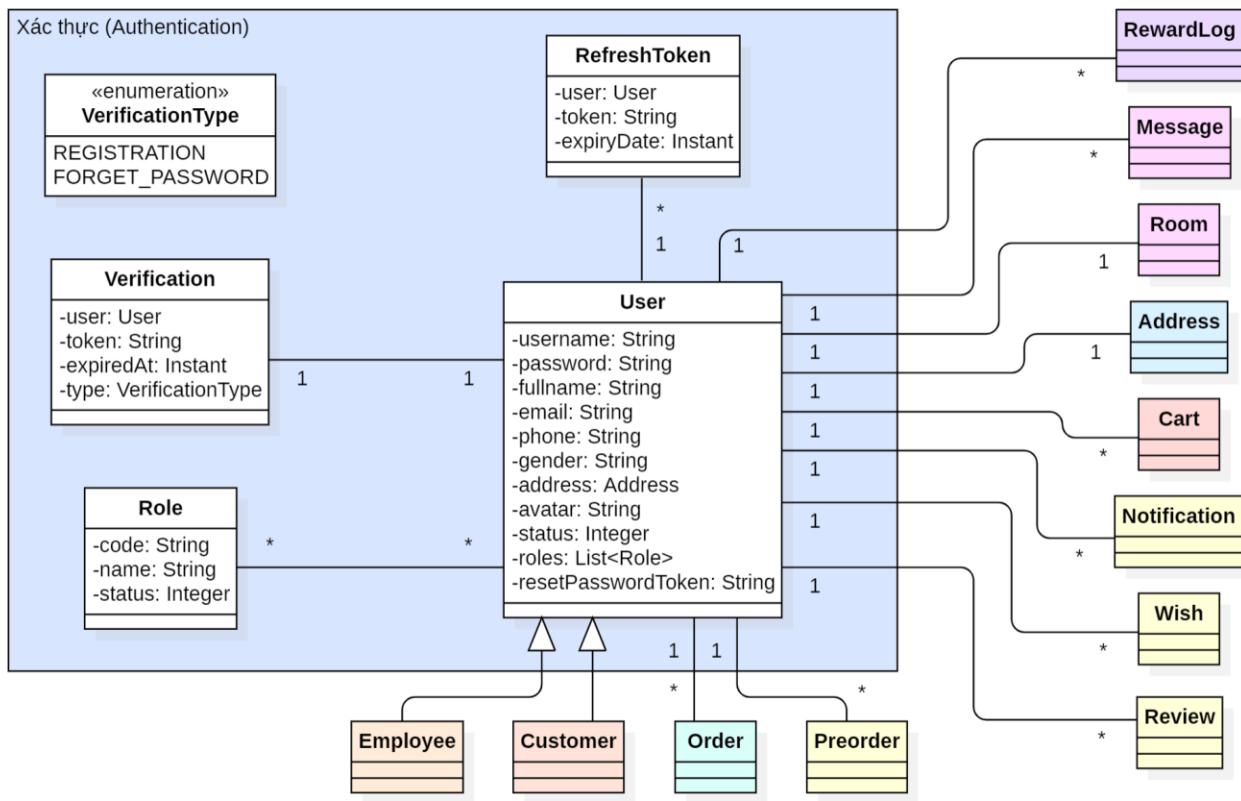
2.4.4.1. Cơ sở lý thuyết:

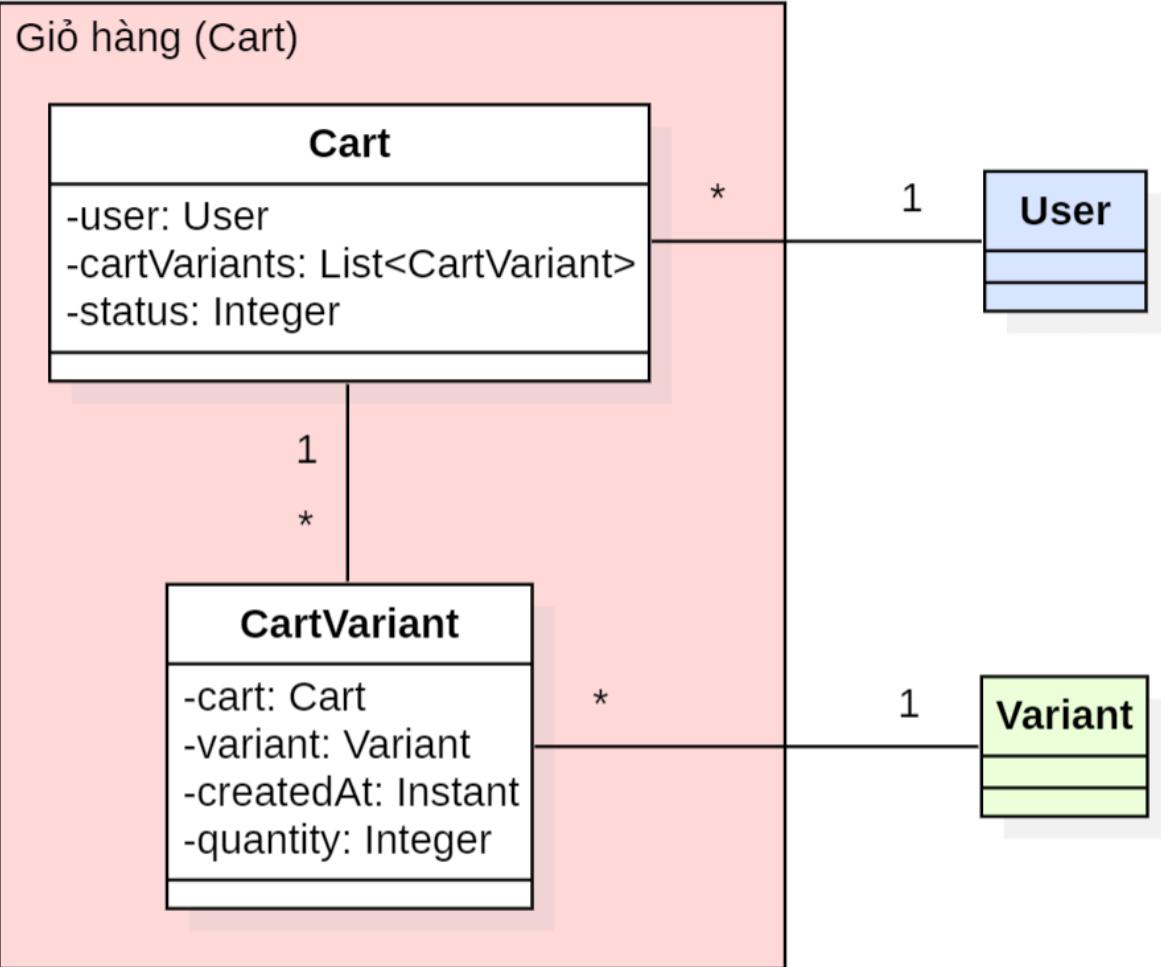
Cấp độ Code (Code/Implementation Level) là cấp độ chi tiết nhất trong mô hình C4, mô tả cấu trúc nội bộ của các thành phần (Component) ở cấp C3. Nếu ở cấp Component Diagram ta thấy các khái niệm chính như Controller, Service, Repository, thì cấp Code Level đi sâu hơn, thể hiện mối quan hệ giữa các lớp (Class), interface, phương thức, và các thành phần mã nguồn cụ thể của thành phần nào đó.

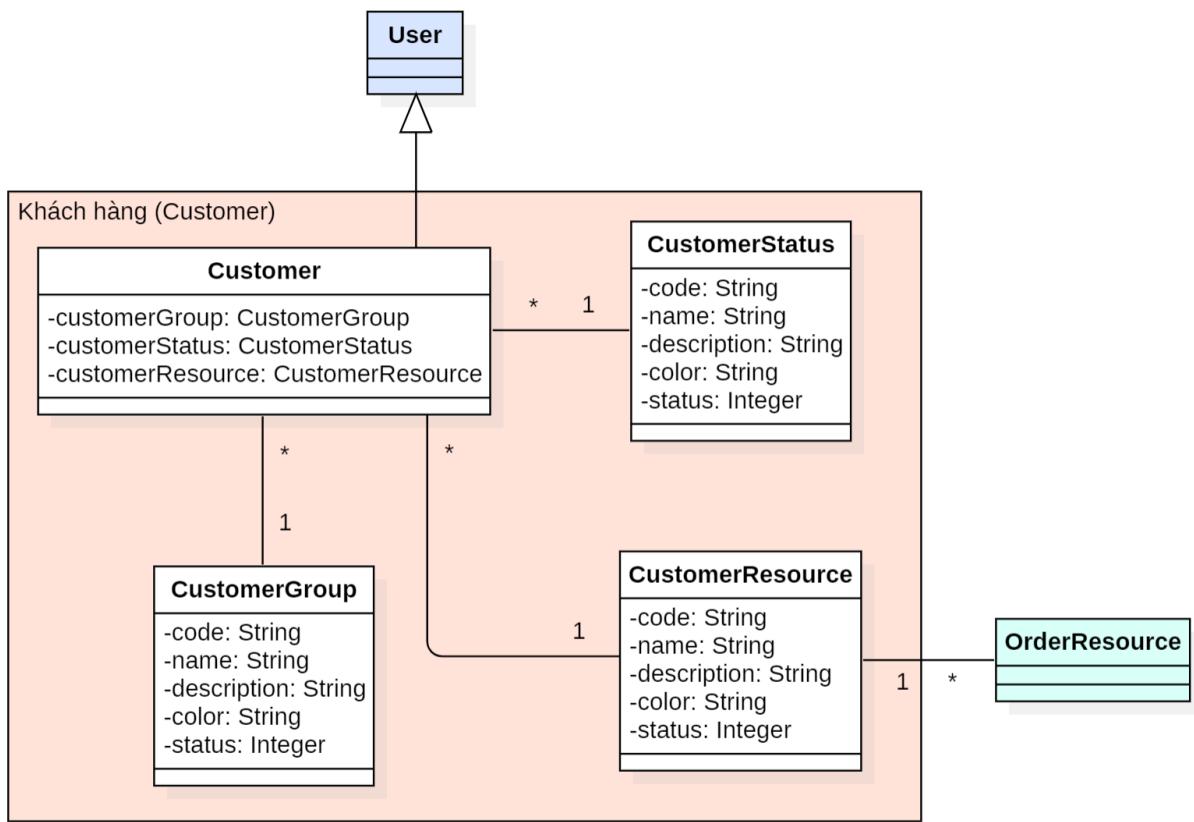
Mục tiêu của cấp độ này là giúp nhà phát triển, kỹ sư phần mềm, hay người bảo trì hiểu rõ:

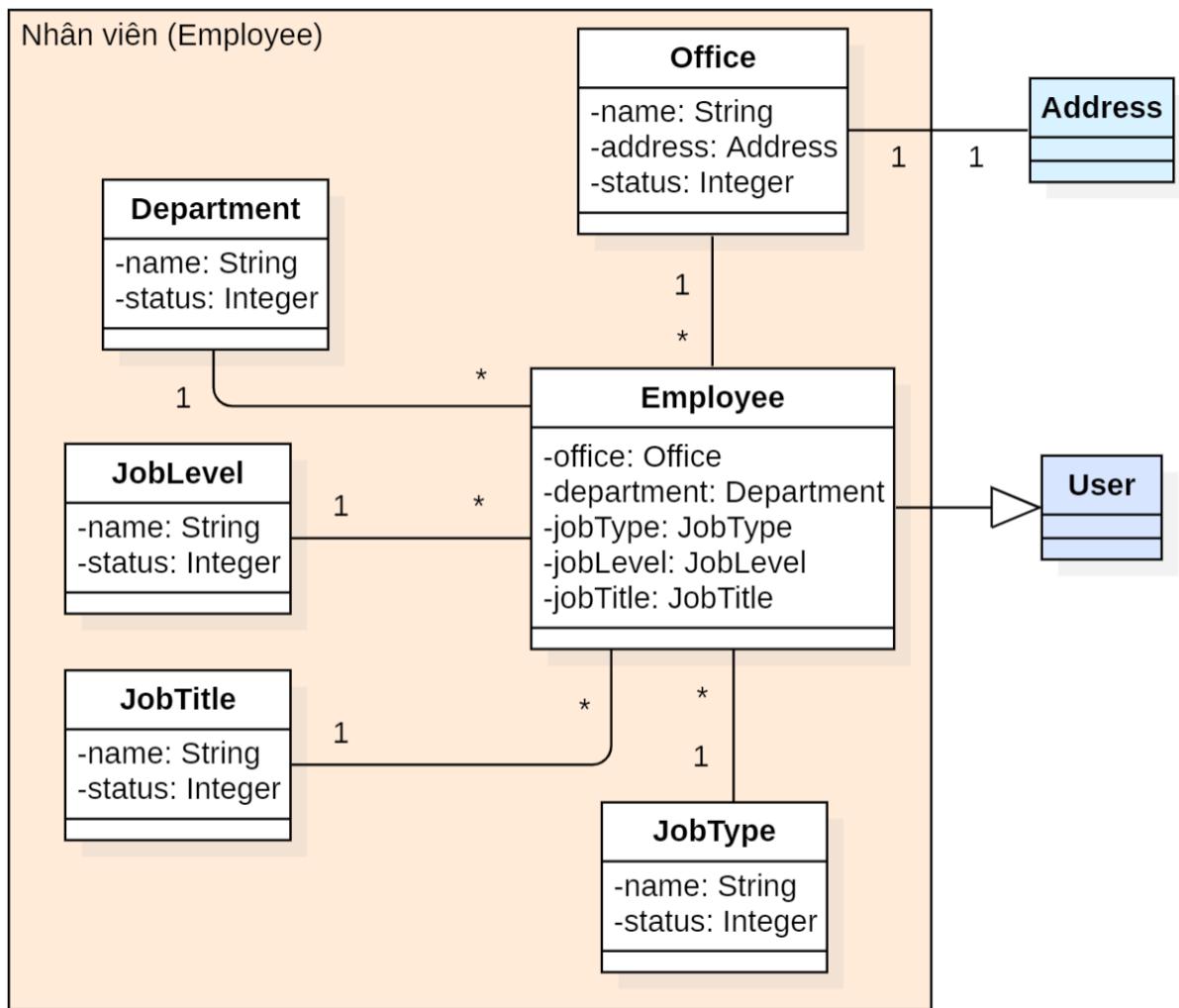
- Cách các lớp được tổ chức và tương tác bên trong một component.
- Các phụ thuộc giữa các lớp, interface, entity, hay thư viện bên ngoài.
- Luồng xử lý chi tiết từ yêu cầu nghiệp vụ đến mức mã nguồn.

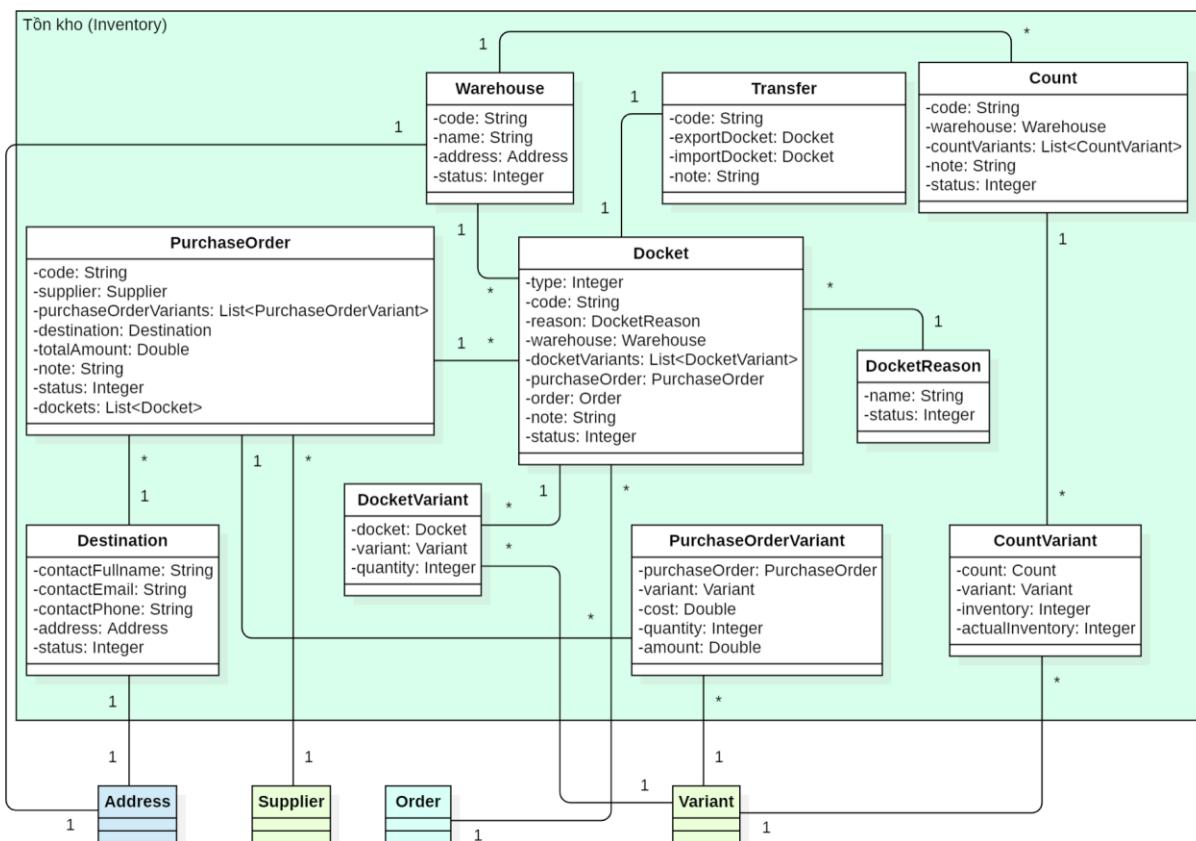
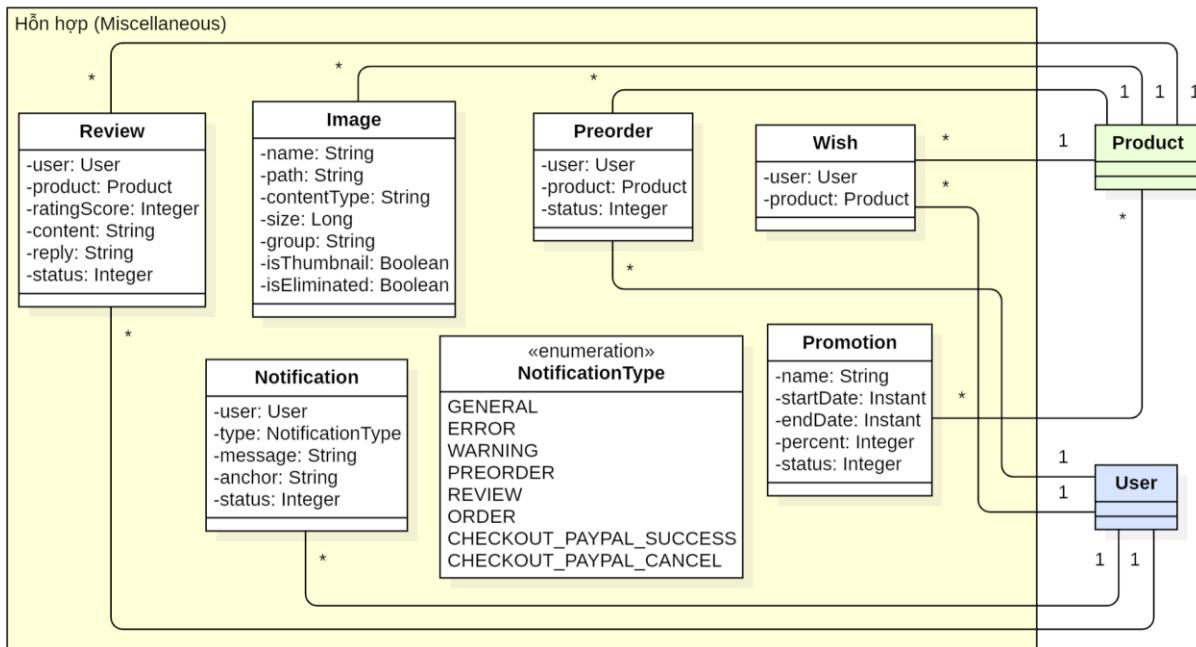
2.4.4.2. Thiết kế sơ đồ:

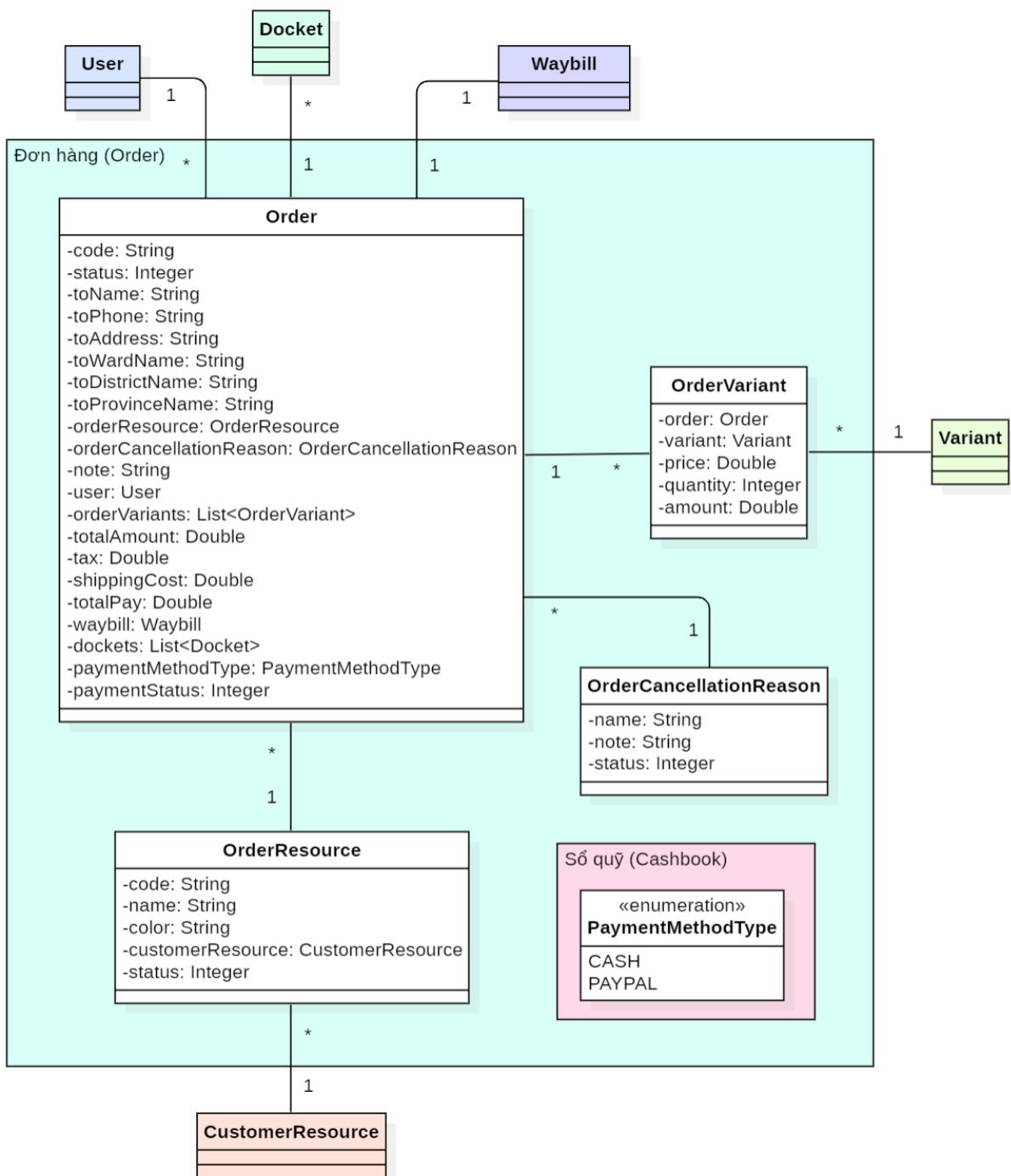


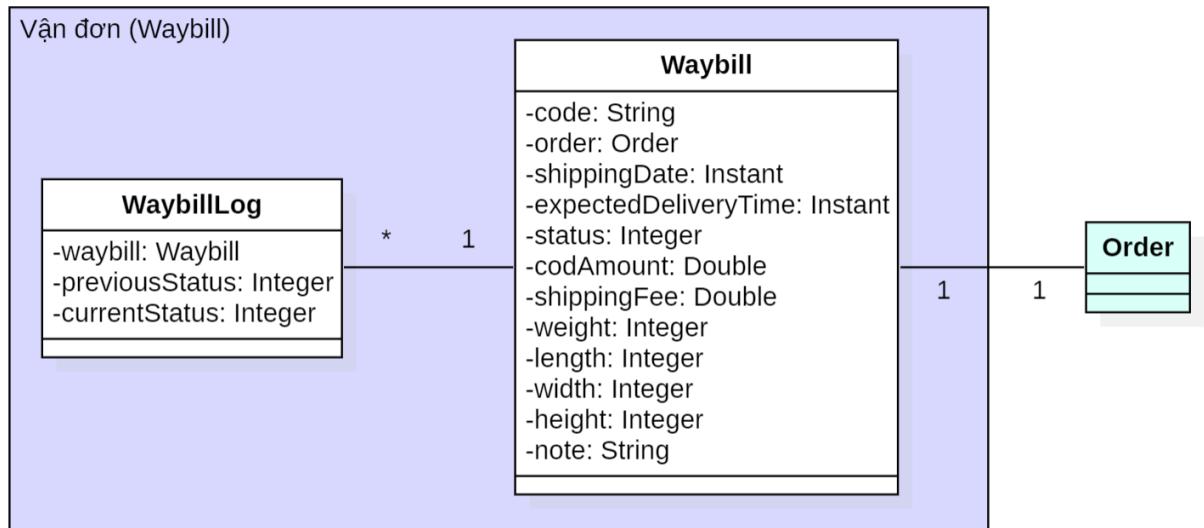
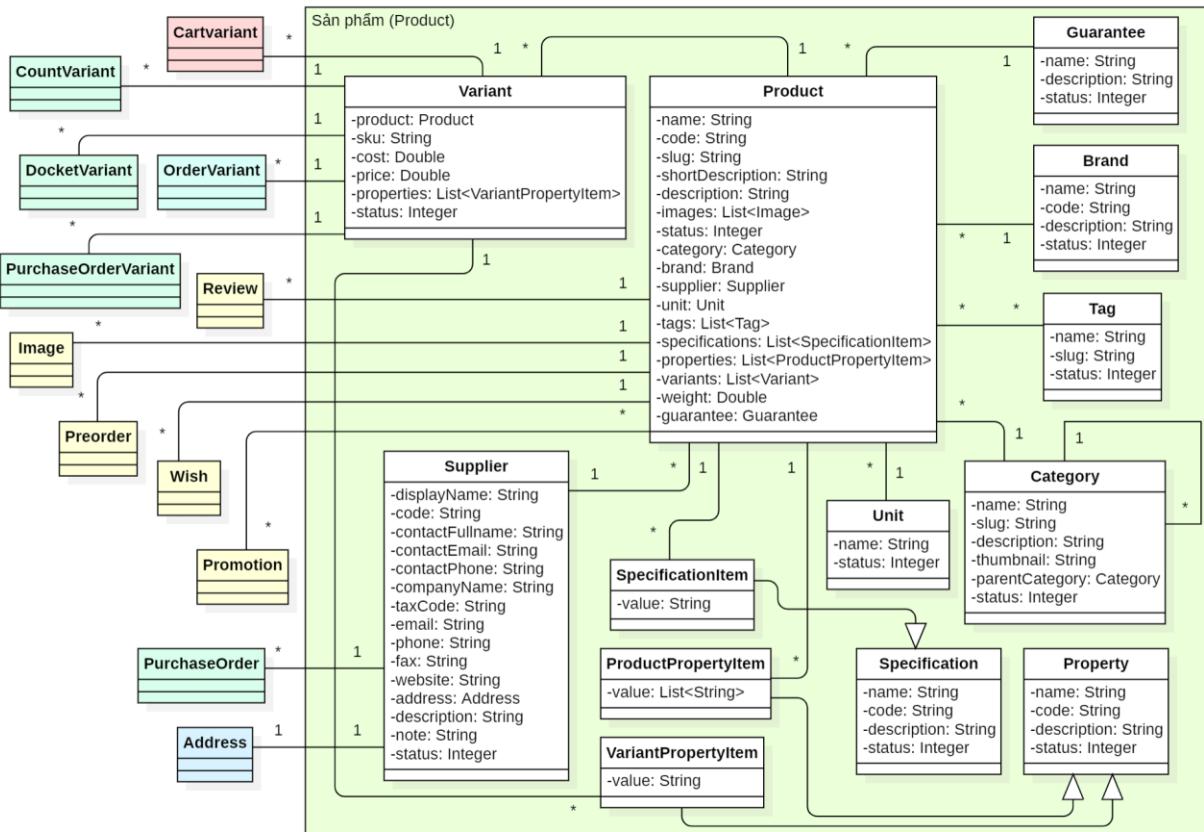












2.5. ERD

2.5.1. Cơ sở lý thuyết:

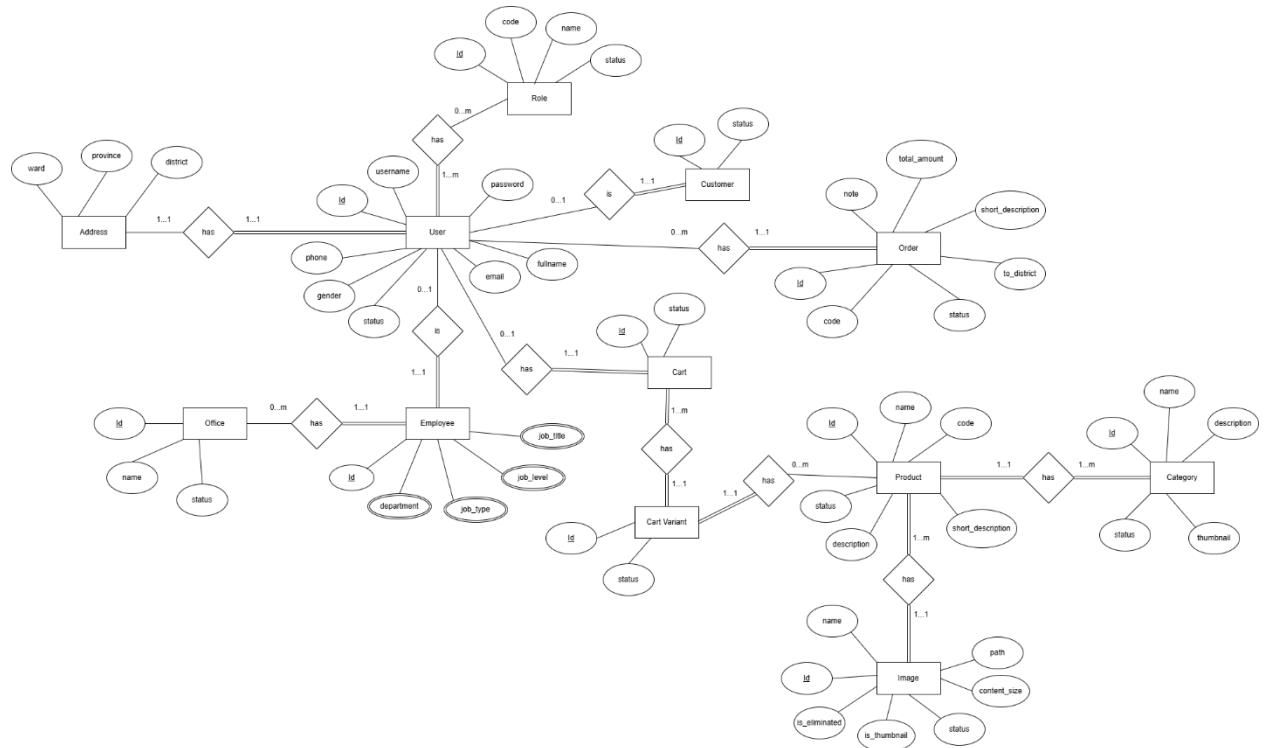
ERD là viết tắt của Entity Relationship Diagram, tạm dịch là sơ đồ quan hệ thực thể. Đây là một loại biểu đồ trực quan mô tả cách các thực thể (như khách hàng, sản phẩm, đơn hàng) và mối quan hệ giữa chúng được tổ chức trong một hệ thống cơ sở dữ liệu. ERD được sử dụng phổ biến trong thiết kế cơ sở dữ liệu để hình dung cấu trúc logic và các quy tắc liên kết dữ liệu một cách rõ ràng.

Các thành phần chính của ERD

- Thực thể (Entity): Là một đối tượng hoặc khái niệm trong hệ thống, thường được biểu diễn bằng hình chữ nhật (ví dụ: "Khách hàng", "Sản phẩm").
- Thuộc tính (Attribute): Là các đặc điểm hoặc thông tin mô tả một thực thể (ví dụ: "Tên khách hàng", "Giá sản phẩm").
- Mối quan hệ (Relationship): Là sự liên kết giữa hai hoặc nhiều thực thể (ví dụ: một "Khách hàng" có thể có nhiều "Đơn hàng").

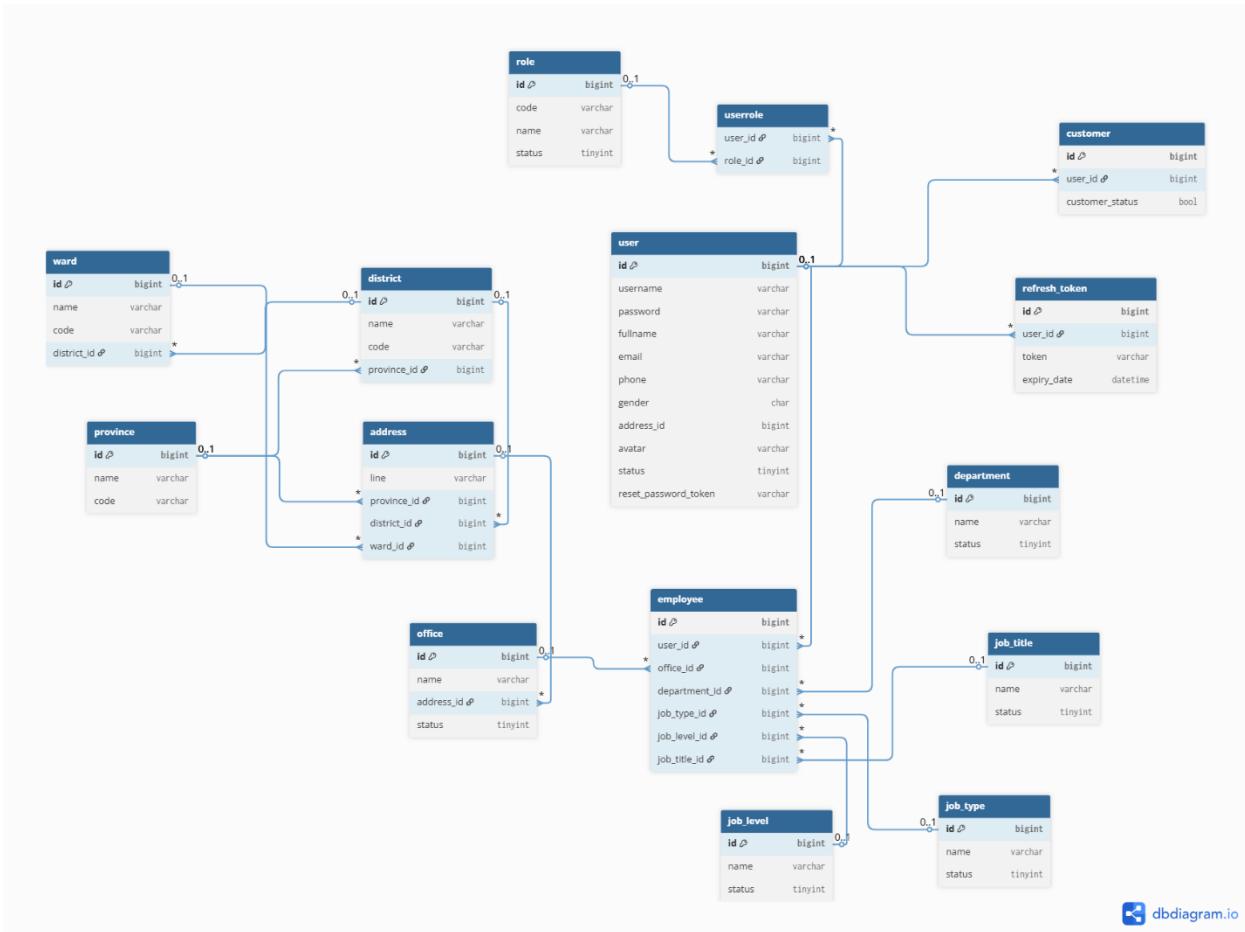
2.5.2. Thiết kế sơ đồ:

Mức quan niệm (Conceptual):



Mức Logic

2.5.2.1. Nhóm quan hệ User – Customer – Employee:



1. Bảng: user

Tên thuộc tính	Kiểu dữ liệu	Mô tả
id	bigint	Khóa chính định danh người dùng.
username	varchar	Tên đăng nhập của người dùng.
password	varchar	Mật khẩu đã được mã hóa.
fullname	varchar	Họ và tên người dùng.
email	varchar	Địa chỉ email đăng ký.
phone	varchar	Số điện thoại liên hệ.
gender	char	Giới tính của người dùng (M/F).
address_id	bigint	Khóa ngoại tham chiếu đến bảng address.
avatar	varchar	Đường dẫn ảnh đại diện.
status	tinyint	Trạng thái tài khoản (0: khóa, 1: hoạt động).
reset_password_token	varchar	Mã token dùng để đặt lại mật khẩu.

2. Bảng: userrole

Tên thuộc tính	Kiểu dữ liệu	Mô tả
user_id	bigint	Khóa ngoại tham chiếu đến bảng user.
role_id	bigint	Khóa ngoại tham chiếu đến bảng role.

3. Bảng: refresh_token

Tên thuộc tính	Kiểu dữ liệu	Mô tả
id	bigint	Khóa chính của token.
user_id	bigint	Khóa ngoại liên kết đến người dùng (user.id).
token	varchar	Mã token dùng để xác thực truy cập.
expiry_date	datetime	Thời điểm token hết hạn.

4. Bảng: address

Tên thuộc tính	Kiểu dữ liệu	Mô tả
id	bigint	Khóa chính địa chỉ.
line	varchar	Địa chỉ chi tiết (số nhà, đường,...).
province_id	bigint	Khóa ngoại đến bảng province.
district_id	bigint	Khóa ngoại đến bảng district.
ward_id	bigint	Khóa ngoại đến bảng ward.

5. Bảng: province

Tên thuộc tính	Kiểu dữ liệu	Mô tả
id	bigint	Khóa chính của tỉnh/thành phố.
name	varchar	Tên tỉnh/thành phố.
code	varchar	Mã định danh hành chính.

6. Bảng: district

Tên thuộc tính	Kiểu dữ liệu	Mô tả
id	bigint	Khóa chính của quận/huyện.
name	varchar	Tên quận/huyện.
code	varchar	Mã hành chính.
province_id	bigint	Khóa ngoại đến bảng province.

7. Bảng: ward

Tên thuộc tính	Kiểu dữ liệu	Mô tả
id	bigint	Khóa chính của phường/xã.
name	varchar	Tên phường/xã.
code	varchar	Mã hành chính.
district_id	bigint	Khóa ngoại đến bảng district.

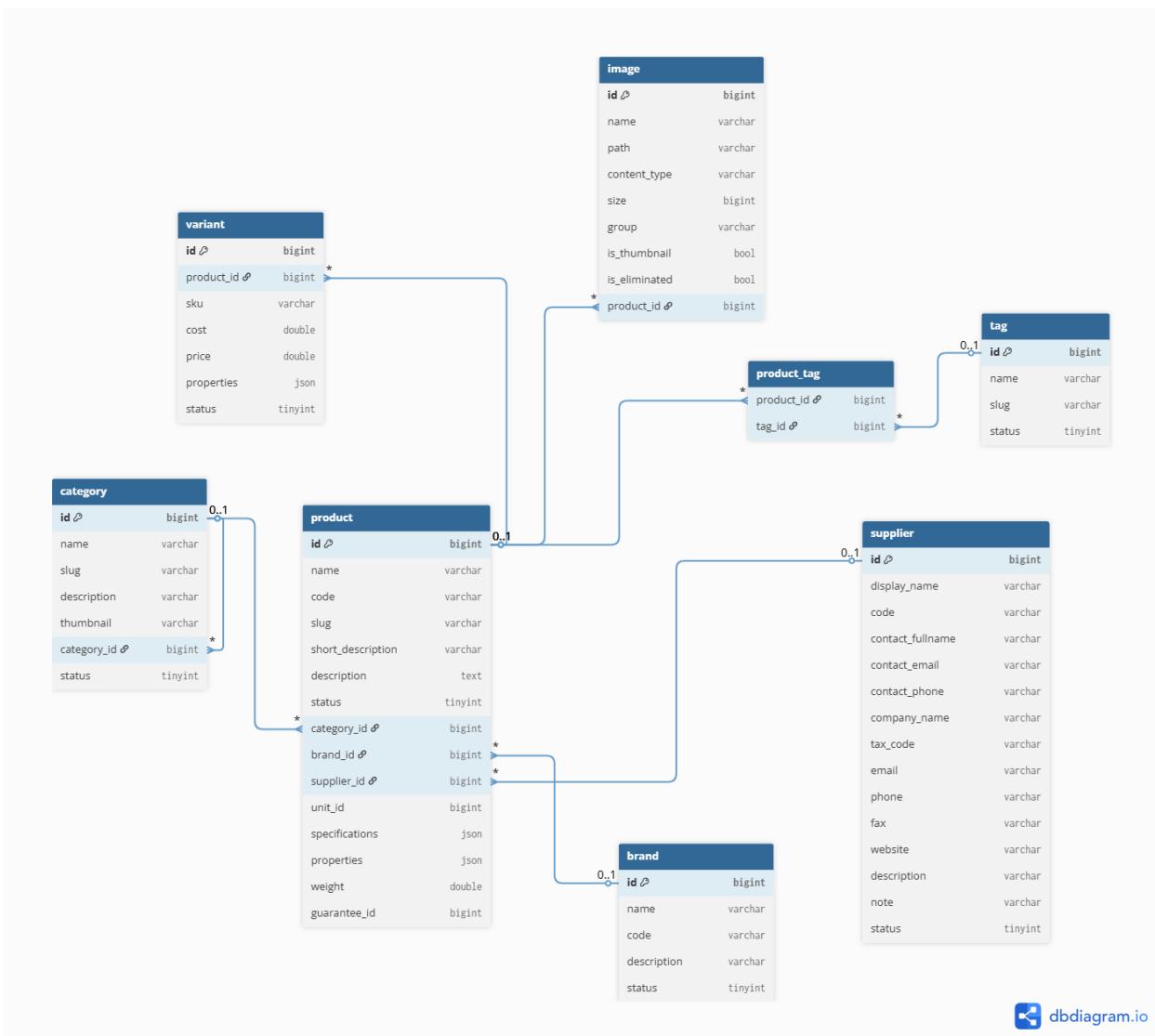
8. Bảng: employee

Tên thuộc tính	Kiểu dữ liệu	Mô tả
id	bigint	Khóa chính nhân viên.
user_id	bigint	Tham chiếu đến bảng user.
office_id	bigint	Tham chiếu đến bảng office.
department_id	bigint	Tham chiếu đến bảng department.
job_type_id	bigint	Tham chiếu đến bảng job_type.
job_level_id	bigint	Tham chiếu đến bảng job_level.
job_title_id	bigint	Tham chiếu đến bảng job_title.

9. Bảng: customer

Tên thuộc tính	Kiểu dữ liệu	Mô tả
id	bigint	Khóa chính khách hàng.
user_id	bigint	Khóa ngoại đến bảng user.
customer_status	bool	Trạng thái khách hàng (đang hoạt động hay không).

2.5.2.2. Nhóm quan hệ Product:



1. Bảng: product

Tên thuộc tính	Kiểu dữ liệu	Mô tả
id	bigint	Khóa chính sản phẩm.
name	varchar	Tên sản phẩm.
code	varchar	Mã định danh sản phẩm (SKU hoặc mã hệ thống).
slug	varchar	Đường dẫn thân thiện (dạng URL).
short_description	varchar	Mô tả ngắn của sản phẩm.
description	text	Mô tả chi tiết sản phẩm.
status	tinyint	Trạng thái hiển thị (1: hoạt động, 0: ẩn).

category_id	bigint	Khóa ngoại tham chiếu đến bảng category.
brand_id	bigint	Khóa ngoại tham chiếu đến bảng brand.
supplier_id	bigint	Khóa ngoại tham chiếu đến bảng supplier.
unit_id	bigint	Mã đơn vị tính (nếu có bảng đơn vị riêng).
specifications	json	Thông tin thông số kỹ thuật chi tiết.
properties	json	Thuộc tính sản phẩm (màu, kích thước,...).
weight	double	Trọng lượng sản phẩm.
guarantee_id	bigint	Khóa ngoại đến chính sách bảo hành (nếu có).

2. Bảng: category

Tên thuộc tính	Kiểu dữ liệu	Mô tả
id	bigint	Khóa chính danh mục.
name	varchar	Tên danh mục sản phẩm.
slug	varchar	Đường dẫn thân thiện cho danh mục.
description	varchar	Mô tả ngắn về danh mục.
thumbnail	varchar	Đường dẫn ảnh đại diện của danh mục.
category_id	bigint	Khóa ngoại tự tham chiếu (danh mục cha).
status	tinyint	Trạng thái hiển thị danh mục.

3. Bảng: brand

Tên thuộc tính	Kiểu dữ liệu	Mô tả
id	bigint	Khóa chính thương hiệu.
name	varchar	Tên thương hiệu.
code	varchar	Mã định danh thương hiệu.
description	varchar	Mô tả chi tiết thương hiệu.
status	tinyint	Trạng thái hiển thị thương hiệu.

4. Bảng: supplier

Tên thuộc tính	Kiểu dữ liệu	Mô tả
id	bigint	Khóa chính nhà cung cấp.
display_name	varchar	Tên hiển thị của nhà cung cấp.

code	varchar	Mã nhà cung cấp.
contact_fullname	varchar	Tên người liên hệ.
contact_email	varchar	Email người liên hệ.
contact_phone	varchar	Số điện thoại người liên hệ.
company_name	varchar	Tên công ty của nhà cung cấp.
tax_code	varchar	Mã số thuế của công ty.
email	varchar	Email chính thức của công ty.
phone	varchar	Số điện thoại công ty.
fax	varchar	Số fax (nếu có).
website	varchar	Trang web công ty.
address_id	bigint	Khóa ngoại tham chiếu đến bảng address.
description	varchar	Mô tả thêm về nhà cung cấp.
note	varchar	Ghi chú nội bộ.
status	tinyint	Trạng thái hoạt động của nhà cung cấp.

5. Bảng: variant

Tên thuộc tính	Kiểu dữ liệu	Mô tả
id	bigint	Khóa chính biến thể sản phẩm.
product_id	bigint	Khóa ngoại tham chiếu đến bảng product.
sku	varchar	Mã SKU riêng cho biến thể.
cost	double	Giá vốn của biến thể.
price	double	Giá bán của biến thể.
properties	json	Thuộc tính biến thể (VD: màu sắc, kích thước).
status	tinyint	Trạng thái hiển thị.

6. Bảng: image

Tên thuộc tính	Kiểu dữ liệu	Mô tả
id	bigint	Khóa chính hình ảnh.
name	varchar	Tên tệp hình ảnh.
path	varchar	Đường dẫn lưu trữ hình ảnh.
content_type	varchar	Loại nội dung (image/png, image/jpeg,...).

size	bigint	Kích thước tệp (bytes).
group	varchar	Nhóm hình ảnh (VD: gallery, thumbnail).
is_thumbnail	bool	Có phải ảnh đại diện không.
is_eliminated	bool	Ảnh đã bị xóa logic chưa.
product_id	bigint	Khóa ngoại tham chiếu đến bảng product.

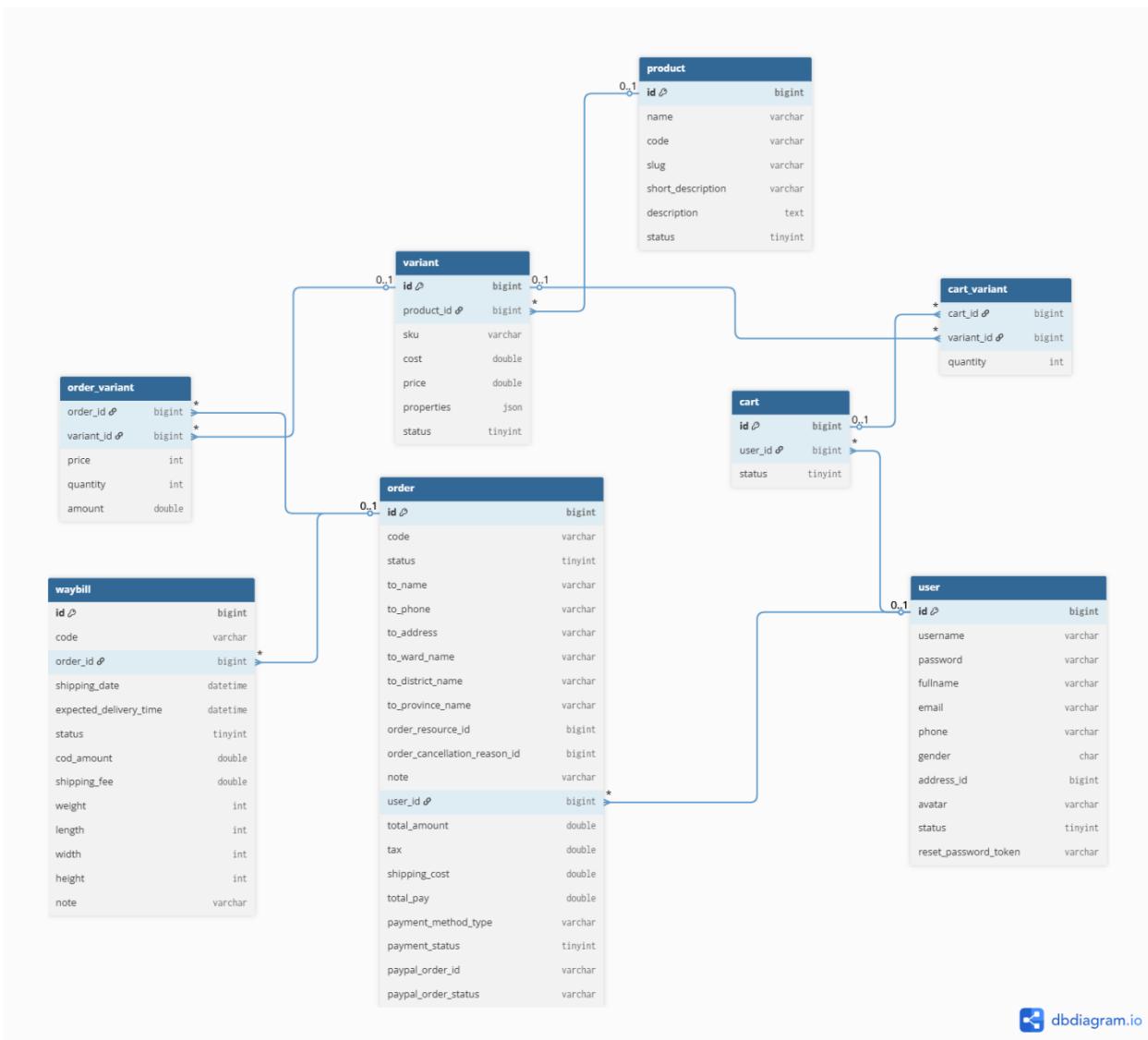
7. Bảng: tag

Tên thuộc tính	Kiểu dữ liệu	Mô tả
id	bigint	Khóa chính thẻ (tag).
name	varchar	Tên thẻ.
slug	varchar	Đường dẫn thân thiện cho thẻ.
status	tinyint	Trạng thái hoạt động của thẻ.

8. Bảng: product_tag

Tên thuộc tính	Kiểu dữ liệu	Mô tả
product_id	bigint	Khóa ngoại đến bảng product.
tag_id	bigint	Khóa ngoại đến bảng tag.

2.5.2.3. Nhóm quan hệ Order/Payment/Waybill:



1. Bảng: order

Tên thuộc tính	Kiểu dữ liệu	Mô tả
id	bigint	Khóa chính đơn hàng.
code	varchar	Mã định danh đơn hàng.
status	tinyint	Trạng thái đơn hàng (ví dụ: 0 – chờ xử lý, 1 – đang giao, 2 – hoàn tất).
to_name	varchar	Tên người nhận hàng.
to_phone	varchar	Số điện thoại người nhận hàng.
to_address	varchar	Địa chỉ giao hàng.
to_ward_name	varchar	Tên phường/xã nơi giao hàng.

to_district_name	varchar	Tên quận/huyện nơi giao hàng.
to_province_name	varchar	Tên tỉnh/thành phố nơi giao hàng.
order_resource_id	bigint	Nguồn đơn hàng (ví dụ: website, mobile app).
order_cancellation_reason_id	bigint	Lý do hủy đơn hàng (nếu có).
note	varchar	Ghi chú của người dùng hoặc nhân viên xử lý đơn.
user_id	bigint	Khóa ngoại tham chiếu đến bảng user.
total_amount	double	Tổng giá trị sản phẩm trong đơn.
tax	double	Thuế áp dụng cho đơn hàng.
shipping_cost	double	Phí vận chuyển.
total_pay	double	Tổng số tiền phải thanh toán.
payment_method_type	varchar	Hình thức thanh toán (PayPal, COD, v.v.).
payment_status	tinyint	Trạng thái thanh toán (0 – chưa thanh toán, 1 – đã thanh toán).
paypal_order_id	varchar	Mã đơn hàng trên PayPal (nếu thanh toán qua PayPal).
paypal_order_status	varchar	Trạng thái đơn hàng PayPal.

2. Bảng: order_variant

Tên thuộc tính	Kiểu dữ liệu	Mô tả
order_id	bigint	Khóa ngoại tham chiếu đến bảng order.
variant_id	bigint	Khóa ngoại tham chiếu đến bảng variant.
price	int	Giá của biến thể tại thời điểm đặt hàng.
quantity	int	Số lượng sản phẩm được đặt.
amount	double	Thành tiền (price × quantity).

3. Bảng: waybill

Tên thuộc tính	Kiểu dữ liệu	Mô tả
id	bigint	Khóa chính vận đơn.
code	varchar	Mã vận đơn (do hệ thống hoặc đơn vị vận chuyển cấp).

order_id	bigint	Khóa ngoại tham chiếu đến bảng order.
shipping_date	datetime	Ngày giao hàng dự kiến.
expected_delivery_time	datetime	Thời gian dự kiến giao đến.
status	tinyint	Trạng thái vận chuyển (ví dụ: đang giao, đã giao, bị hoàn trả).
cod_amount	double	Số tiền thu hộ (nếu COD).
shipping_fee	double	Phí giao hàng.
weight	int	Trọng lượng gói hàng.
length	int	Chiều dài kiện hàng.
width	int	Chiều rộng kiện hàng.
height	int	Chiều cao kiện hàng.
note	varchar	Ghi chú cho đơn vị giao hàng.

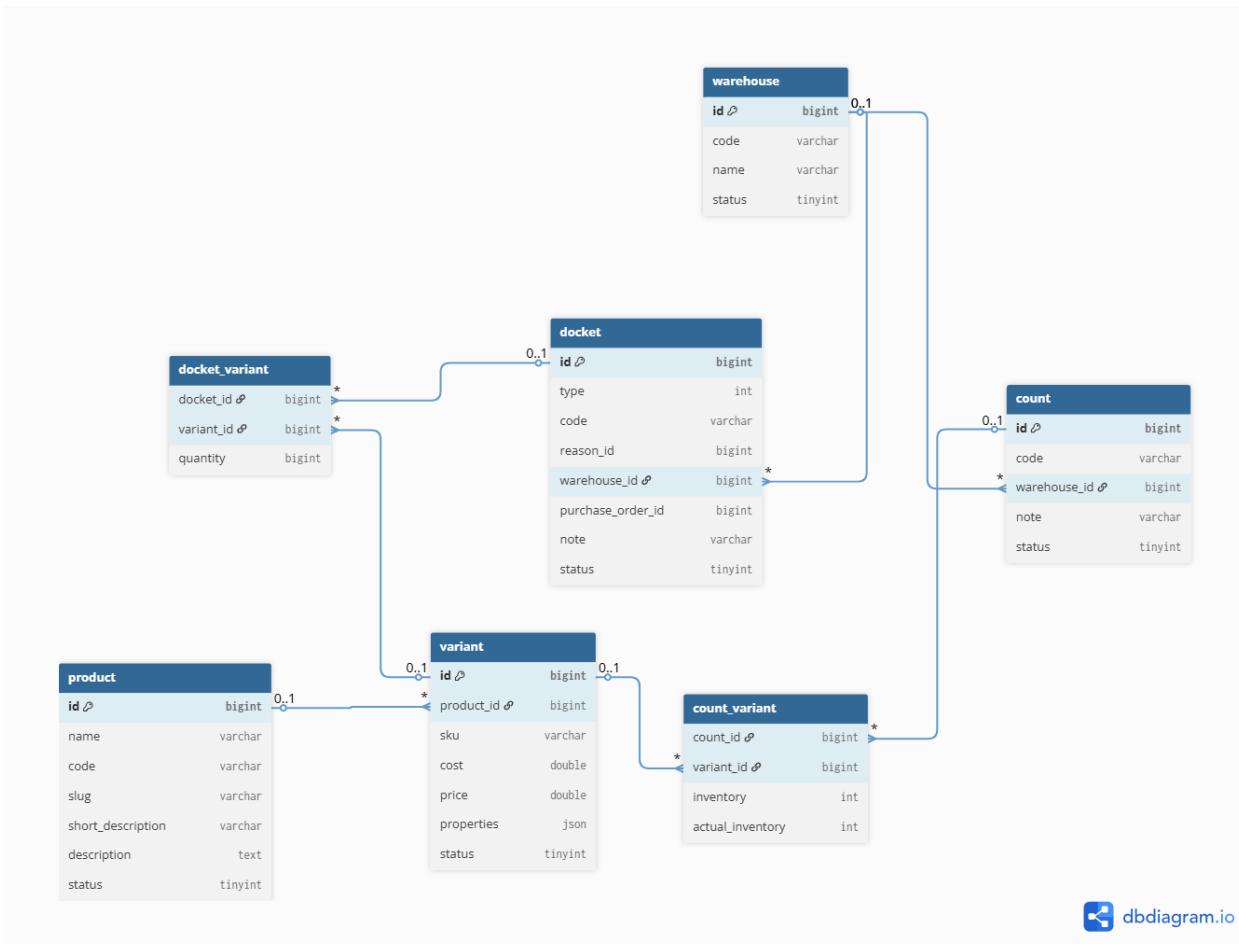
4. Bảng: cart

Tên thuộc tính	Kiểu dữ liệu	Mô tả
id	bigint	Khóa chính giỏ hàng.
user_id	bigint	Khóa ngoại tham chiếu đến bảng user.
status	tinyint	Trạng thái giỏ hàng (ví dụ: 0 – đang hoạt động, 1 – đã thanh toán).

5. Bảng: cart_variant

Tên thuộc tính	Kiểu dữ liệu	Mô tả
cart_id	bigint	Khóa ngoại tham chiếu đến bảng cart.
variant_id	bigint	Khóa ngoại tham chiếu đến bảng variant.
quantity	int	Số lượng sản phẩm trong giỏ hàng.

2.5.2.4. Nhóm quan hệ Inventory (tồn kho):



1. Bảng: warehouse (kho hàng)

Tên thuộc tính	Kiểu dữ liệu	Mô tả
id	bigint	Khóa chính. Mã định danh duy nhất cho kho hàng.
code	varchar	Mã code nội bộ của kho hàng.
name	varchar	Tên của kho hàng.
status	tinyint	Trạng thái của kho hàng.

2. Bảng: docket (phiếu nhập/xuất kho)

Tên thuộc tính	Kiểu dữ liệu	Mô tả
id	bigint	Khóa chính. Mã định danh duy nhất cho phiếu nhập/xuất/chuyển kho.
type	int	Loại phiếu (1-Nhập, 2-Xuất).

code	varchar	Mã code/số hiệu của phiếu.
warehouse_id	bigint	Khóa ngoại tham chiếu đến warehouse.id. Kho hàng bị ảnh hưởng bởi phiếu này.
purchase_order_id	bigint	Liên kết với mã đơn mua hàng (nếu là phiếu nhập), có thể là Khóa ngoại.
note	varchar	Ghi chú cho phiếu.
status	tinyint	Trạng thái của phiếu

3. Bảng: docket_variant (chi tiết phiếu nhập/xuất)

Tên thuộc tính	Kiểu dữ liệu	Mô tả
docket_id	bigint	Khóa chính và Khóa ngoại tham chiếu đến docket.id. Liên kết với phiếu nhập/xuất cụ thể.
variant_id	bigint	Khóa chính và Khóa ngoại tham chiếu đến variant.id. Biến thể sản phẩm trong phiếu.
quantity	bigint	Số lượng của biến thể sản phẩm trong phiếu này.

Chương 3: Kế hoạch kiểm thử (Test plan):

3.1. Cơ sở Lý thuyết về Kiểm thử Phần mềm:

3.1.1. Khái niệm và Mục tiêu của Kiểm thử Phần mềm

Khái niệm

Kiểm thử phần mềm (Software Testing) là quá trình thực hiện một chương trình hoặc hệ thống với mục đích tìm ra lỗi (bugs/defects). Nó là hoạt động bắt buộc trong vòng đời phát triển phần mềm (SDLC) nhằm đánh giá chất lượng sản phẩm bằng cách kiểm tra và xác minh rằng phần mềm đáp ứng các yêu cầu đã được xác định.

Kiểm thử là quá trình bao gồm không chỉ việc thực hiện chương trình mà còn các hoạt động lập kế hoạch, phân tích, thiết kế, triển khai và báo cáo kết quả.

Mục tiêu

- Mục tiêu cốt lõi của kiểm thử không chỉ là tìm lỗi mà còn là:
- Xác minh (Verification): Đảm bảo rằng phần mềm đã được xây dựng đúng theo yêu cầu kỹ thuật và thiết kế ban đầu (làm đúng sản phẩm).
- Thảm định (Validation): Đảm bảo rằng phần mềm đáp ứng nhu cầu thực tế của người dùng và giải quyết được vấn đề nghiệp vụ (làm sản phẩm đúng).
- Đánh giá Chất lượng: Cung cấp thông tin khách quan về mức độ rủi ro và chất lượng của sản phẩm cho các bên liên quan.
- Ngăn ngừa lỗi (Prevention): Thông qua việc kiểm thử sớm (Shift-Left Testing) và phân tích lỗi, giúp cải thiện quy trình phát triển và ngăn ngừa lỗi tương tự tái diễn.

3.1.2. Khái niệm Kế hoạch Kiểm thử (Test Plan):

Kế hoạch kiểm thử (Test Plan) là tài liệu mô tả mục tiêu, phạm vi, chiến lược, phương pháp tiếp cận, nguồn lực cần thiết và lịch trình cho các hoạt động kiểm thử sắp tới. Đây là tài liệu sống, được phê duyệt bởi các bên liên quan, nhằm định hướng và quản lý toàn bộ nỗ lực kiểm thử của dự án.

Các thành phần chính của Test Plan

Một Test Plan tiêu chuẩn (dựa trên IEEE 829) bao gồm các mục chính:

- Giới thiệu (Introduction): Mục đích, Tổng quan hệ thống (Overview), Định nghĩa & Thuật ngữ.
- Phạm vi Kiểm thử (Scope): Các tính năng cần và không cần kiểm thử.
- Chiến lược Kiểm thử (Test Strategy): Phương pháp tiếp cận, các cấp độ và loại hình kiểm thử sẽ được áp dụng.
- Môi trường Kiểm thử (Test Environment): Phần cứng, phần mềm, công cụ, và dữ liệu kiểm thử.
- Nguồn lực & Lịch trình (Resources & Schedule): Phân bổ nhân sự, ước tính nỗ lực và các mốc quan trọng (Milestones).

- Sản phẩm bàn giao (Deliverables): Các tài liệu kiểm thử sẽ được tạo ra (Test Cases, Defect Logs, Test Reports).

3.2. Tổng quan Kế hoạch Kiểm thử Dự án ShopEase

3.2.1. Mục đích và Phạm vi (Scope):

Mục đích (Purpose)

Kế hoạch kiểm thử này nhằm mục đích:

- Xác định các thành phần phần mềm cần kiểm thử của ứng dụng web thương mại điện tử ShopEase (Backend: Spring Boot, Frontend: React).
- Đề xuất và mô tả các chiến lược kiểm thử cụ thể (chức năng, phi chức năng) để đảm bảo chất lượng.
- Cung cấp khung ước lượng nỗ lực, quản lý rủi ro và sản phẩm bàn giao trong suốt quá trình kiểm thử.

Phạm vi Kiểm thử Chính (Features to be tested):

Phần mềm sẽ được kiểm thử toàn diện các tính năng sau:

Loại	Chức năng (Functional)
Xác thực & Quyền	Đăng ký, Đăng nhập (Buyer/SysAdmin), Đăng xuất, Kiểm soát truy cập (JWT token), Quản lý trạng thái người dùng (Admin).
Danh mục Sản phẩm	Xem, Lọc, Tim kiếm sản phẩm (Buyer); Thêm, Sửa, Xóa, Quản lý danh mục sản phẩm (SysAdmin).
Giỏ hàng & Đặt hàng	Thêm/Xóa/Cập nhật sản phẩm trong Giỏ hàng, Tính tổng chi phí, Đặt hàng (COD), Theo dõi trạng thái đơn hàng (Buyer), Xử lý/Cập nhật/Hủy đơn hàng (SysAdmin).
Quản lý Tồn kho	Xem chi tiết tồn kho, Tạo/Xem/Sửa/Xóa Phiếu Nhập/Xuất kho, Xem lịch sử nhập – xuất, Quản lý đơn mua hàng (SysAdmin).
Khởi tạo Dữ liệu	Kiểm tra việc tạo tài khoản SysAdmin, Người mua mẫu, Sản phẩm mẫu và Đơn hàng mẫu ban đầu.

Phạm vi Kiểm thử Phi Chức năng (Non-Functional)

Loại	Mô tả
Khả năng sử dụng (Usability)	Quản lý phiên người dùng, Tính đáp ứng của giao diện React (Responsive Design), Điều hướng trực quan.
Tương thích (Compatibility)	Kiểm thử trên các trình duyệt chính (Chrome, Firefox, Edge, Safari) và hệ điều hành Windows 11/macOS.
Giới hạn Thiết kế (Design Constraints)	Giới hạn độ dài dữ liệu, hỗ trợ mã hóa ký tự UTF-8.

Giao diện (Interfaces)	Hiển thị giao diện người dùng chính xác, API endpoints hoạt động đúng giữa Frontend và Backend.
------------------------	---

Phạm vi Không Kiểm thử (Not in scope):

- Tích hợp công thanh toán bên thứ ba thực tế (chỉ mô phỏng thanh toán).
- Kiểm thử ứng dụng di động riêng biệt (chỉ tập trung vào tính đáp ứng web).
- Kiểm thử chịu đựng cực hạn (Stress testing) và khôi phục sau sự cố (Crash/Recovery testing).

3.2.2. Chiến lược Kiểm thử (Test Strategy):

Chiến lược kiểm thử được thiết kế theo **mô hình Agile (Scrum)** và phương pháp **Kiểm thử Liên tục (Continuous Testing)**, đảm bảo chất lượng được tích hợp vào mọi giai đoạn phát triển (Shift-Left Testing) và được duy trì qua CI/CD Pipeline.

A. Các Giai đoạn Kiểm thử (Test Stages):

Chiến lược này được thiết kế dựa trên mô hình kiểm thử theo chức năng (Feature-driven) và theo từng giai đoạn (Sprint) trong Kế hoạch phát triển dự án dự án (Project Plan).

Giai đoạn	Mục tiêu Ánh xạ từ Sprint Backlog	Nhóm Chịu trách nhiệm	Tần suất
Unit Test (UT)	Xác minh logic nghiệp vụ cấp độ hàm/phương thức (Services, Controllers, Components). Đảm bảo các Tasks trong Sprint Backlog được mã hóa đúng logic.	Development Team	Chạy tự động khi developer commit code và khi merge vào nhánh chính.
Integration Test (IT)	Đảm bảo sự tương tác giữa các module (React ↔ API Spring Boot) và với Cơ sở Dữ liệu (Spring Boot ↔ MySQL) là chính xác.	Development/Testing Team	Chạy tự động sau Unit Test và trước khi Build.
System Test (ST)	Kiểm thử toàn bộ hệ thống (end-to-end) cho các User Story được chọn trong Sprint. Đảm bảo toàn bộ luồng nghiệp vụ của Sprint đó hoạt động.	Testing Team	Chạy tự động sau khi bản Build mới được triển khai lên môi trường Staging. Mô phỏng luồng người dùng cuối
Regression Test (RT)	Tái kiểm tra toàn bộ các chức năng đã hoàn thành ở các Sprint trước đó (ví dụ: Auth, Catalog) sau khi mã Sprint mới được hợp nhất.	Testing Team	Toàn bộ bộ Test Tự động (Unit, IT, E2E) từ các Sprint trước sẽ được chạy lại để bảo vệ các

			tính năng đã hoàn thành.
Acceptance Test (UAT)	Xác nhận rằng Product Increment (tổng sản phẩm sau Sprint) đáp ứng Definition of Done (DoD) và nhu cầu của đại diện SysAdmin/Buyer.	Đại diện Người dùng	Cuối mỗi Sprint, trên môi trường Staging.

B. Chiến lược Tự động hóa trong CI/CD (Test Automation in CI/CD):

Chiến lược này tập trung vào việc tự động hóa kiểm thử để hỗ trợ quy trình Tích hợp Liên tục (CI) và Triển khai Liên tục (CD) của dự án:

1. Unit & Integration Test Automation (CI Focus):

- Mục đích: Đảm bảo tính ổn định của mã nguồn mới.
- Kích hoạt: Tự động chạy khi Developer push code và khi merge vào nhánh chính.
- Hành động: Nếu các bài kiểm tra này thất bại, CI Pipeline sẽ bị ngắt (break), ngăn chặn việc tạo ra bản build lỗi.

2. System & Regression Test Automation (CD Focus):

- Mục đích: Bảo vệ các chức năng đã có (Hồi quy) và xác minh tính đúng đắn của bản build mới.
- Kích hoạt: Tự động chạy sau khi bản build mới được triển khai thành công lên môi trường Staging.

3. Performance Testing (Load) - Periodic:

- Mục đích: Đánh giá hiệu năng phản hồi và khả năng chịu tải của API dưới mức tải trung bình (50–100 người dùng đồng thời).
- Tần suất: Thực hiện vào cuối các Sprint quan trọng (Sprint 3, 4) sau khi các module trọng yếu (Order, Inventory) hoàn thành.

C. Các Loại Hình Kiểm thử Áp dụng (Testing Types):

Loại kiểm thử	Mục đích chính theo Product Backlog	Kỹ thuật áp dụng
Functional Testing	Xác minh từng chức năng theo User Story (AUTH-01, CAT-02, INV-03).	Phân vùng tương đương (Equivalence Partitioning) và Phân tích giá trị biên (Boundary Value Analysis) cho các trường nhập liệu (số lượng sản phẩm, giá trị tồn kho,...).
Business Cycle Testing	Đảm bảo chu kỳ nghiệp vụ hoàn chỉnh (ví dụ: Đặt hàng (ORDER-05) → Xử lý Đơn hàng (ORDER-09) → Xuất kho (INV-03)).	Sử dụng các luồng nghiệp vụ cốt lõi (User Stories có Priority: High) làm kịch bản kiểm thử End-to-End.

Data Integrity Testing	Đảm bảo dữ liệu tồn kho được cập nhật chính xác giữa các bảng docket_variant, variant và count_variant.	Kiểm tra các ràng buộc dữ liệu (Data Constraints) và tính toàn vẹn sau các giao dịch (ví dụ: tồn kho không âm sau khi xuất kho).
Performance Testing	Đo lường hiệu năng của các API quan trọng (ví dụ: Tạo API lấy danh sách tồn kho INV-01, API Đặt hàng ORDER-05).	Sử dụng công cụ tạo tải (JMeter/k6) để đo Thời gian Phản hồi (Response Time) và Tỷ lệ Lỗi.
Security/Access Control	Đảm bảo phân quyền truy cập (AUTH-05, AUTH-08) được tuân thủ nghiêm ngặt theo vai trò (Buyer vs. SysAdmin).	Thử truy cập các tài nguyên Admin bằng tài khoản Buyer và ngược lại.

3.2.3. Môi trường và Nguồn lực

A. Môi trường Kiểm thử (Test Environment):

Môi trường kiểm thử được cấu hình để mô phỏng gần nhất với môi trường Production, bao gồm:

Yếu tố	Chi tiết kỹ thuật
Web Server	Embedded Tomcat (phiên bản 11.0.13)
Backend	Spring Boot (3.5.6) trên Windows 11
Frontend	ReactJS (19) với Mantine UI
Cơ sở Dữ liệu	MySQL (8.0.44)
Trình Duyệt	Chrome, Firefox (Execution), Edge, Safari (Certification)
Phần cứng (Tối thiểu)	CPU Intel Core i5, RAM 4GB, HDD 40GB (cho Web server)

B. Nguồn lực (Human Resource)

Đội ngũ kiểm thử bao gồm các thành viên với vai trò và trách nhiệm cụ thể:

- Trang Gia Huy (Tester):** Quản lý đội kiểm thử, lập kế hoạch, thiết kế và thực hiện Test Cases cho các chức năng cốt lõi (Xác thực, Danh mục).
- Nguyễn Lê Quỳnh Hương (Tester):** Thiết kế và thực hiện Test Cases cho hiệu năng (Performance), tương thích (Compatibility), và hỗ trợ Unit Testing cho các mô-đun Backend.

C. Các Mốc Quan trọng (Test Milestones)

Các mốc thời gian được gắn kết trực tiếp với lịch trình Sprint của Kế hoạch Dự án, nhấn mạnh việc kiểm thử được thực hiện theo từng đợt lặp.

Công việc	Giai đoạn trong Kế hoạch dự án	Ngày Bắt đầu (Dự kiến)	Ngày Kết thúc (Dự kiến)

I. PHA THIẾT KẾ & LẬP KẾ HOẠCH (PRE-SPRINT & SPRINT 1)	Khởi tạo/Phân tích		
Lập Kế hoạch Kiểm thử Tổng thể (Test Plan)	Khởi tạo Dự án	2025-10-01	2025-10-06
Đánh giá & Cập nhật Kế hoạch Kiểm thử (TP)	Khởi tạo Dự án	2025-10-07	2025-10-08
Lập Kế hoạch Kiểm thử Đơn vị (Unit Test Plan)	Trước Sprint 1	2025-10-09	2025-10-10
Đánh giá & Cập nhật Kế hoạch UT	Trước Sprint 1	2025-10-11	2025-10-11
Thiết kế Trường hợp Kiểm thử Đơn vị (UTC)	Sprint 1 (Catalog)	2025-10-12	2025-10-17
Đánh giá & Cập nhật Trường hợp UT (UTC)	Sprint 1 (Catalog)	2025-10-18	2025-10-19
II. THỰC THI KIỂM THỬ THEO SPRINT			
Thực hiện UT, IT, ST & Báo cáo lỗi (Chức năng Catalog)	Cuối Sprint 1	2025-10-20	2025-10-24
Kiểm thử Chấp nhận Người dùng (UAT) cho S1	Sau Sprint 1	2025-10-25	2025-10-27
Thiết kế Trường hợp Kiểm thử Tích hợp (ITC) (Auth, Order)	Bắt đầu Sprint 2	2025-11-04	2025-11-08
Đánh giá & Cập nhật Trường hợp Tích hợp (ITC)	Bắt đầu Sprint 2	2025-11-09	2025-11-10
Thiết kế Trường hợp Kiểm thử Hệ thống (STC) (Order, Inventory)	Bắt đầu Sprint 2	2025-11-11	2025-11-15
Đánh giá & Cập nhật Trường hợp Hệ thống (STC)	Bắt đầu Sprint 2	2025-11-16	2025-11-17
Thực hiện UT, IT, ST, RT & Báo cáo lỗi (S2: Auth, S3: Cart/Order)	Cuối Sprint 2	2025-11-28	2025-12-02
Kiểm thử Chấp nhận Người dùng (UAT) cho S2 & S3	Sau Sprint 2	2025-12-04	2025-12-08
Thực hiện UT, IT, ST, RT & Báo cáo lỗi (S4: Inventory)	Tiếp nối S2 UAT	2025-12-09	2025-12-13
Kiểm thử Chấp nhận Người dùng Cuối cùng (Final UAT)	Cuối Dự án	2025-12-14	2025-12-16
Triển khai Production & Kiểm thử sau Triển khai (PDT)	Sau UAT	2025-12-17	2025-12-18

D. Sản phẩm Bàn giao (Deliverables)

Các tài liệu chính sẽ được bàn giao sau khi hoàn thành kiểm thử bao gồm:

- Test Plan (Kế hoạch Kiểm thử).
- Bộ Test Case Thủ công: Unit Test Cases, Integration Test Cases, System Test Cases.
- Bộ Test Tự động: Các mã nguồn (Scripts) của API và E2E Test Tự động được lưu trữ cùng mã nguồn dự án.
- Defect Log (Nhật ký lỗi) chi tiết.
- Test Reports:
 - Test Summary Report (cho mỗi Sprint): Báo cáo kết quả kiểm thử thủ công và tự động cho các User Story trong Sprint đó.
 - Final Test Report (cuối dự án): Báo cáo chất lượng tổng thể và tỷ lệ lỗi được tìm thấy.

CHƯƠNG 4. THIẾT KẾ KIỂM THỬ

4.1. Giới thiệu

Chương này trình bày quy trình và kết quả của việc thiết kế các trường hợp kiểm thử (Test Case) chi tiết dựa trên chiến lược V-Model dựa vào các tài liệu ở Chương 1, Chương 2, Chương 3.

Mục tiêu của chương là đảm bảo:

- Mọi yêu cầu nghiệp vụ đều được ánh xạ thành các đối tượng kiểm thử cụ thể
- Các luồng nghiệp vụ chính và ngoại lệ đều được bao phủ
- Hoạt động kiểm thử có thể truy vết ngược từ Test Case → Workflow → Requirement

4.2. Quy trình thiết kế kiểm thử theo V-Model

Quy trình thiết kế kiểm thử của nhóm được xây dựng hoàn toàn dựa trên mô hình V-Model – một mô hình nhấn mạnh tính đối xứng và tương hỗ giữa hoạt động phát triển và hoạt động kiểm thử.

Trong cấu trúc dạng chữ V, mỗi giai đoạn phân tích và thiết kế ở nhánh trái đều có một hoạt động kiểm thử tương ứng ở nhánh phải, nhằm đảm bảo rằng mọi yêu cầu đều có thể được xác minh thông qua một trường hợp kiểm thử cụ thể.

4.2.1. Phân tích yêu cầu - 1a

Ở giai đoạn đầu tiên, nhóm tiến hành thu thập, phân tích và làm rõ các yêu cầu nghiệp vụ của hệ thống bán hàng trực tuyến.

Mục tiêu của bước này là đảm bảo rằng tất cả yêu cầu:

- Được mô tả đầy đủ
- Đúng phạm vi
- Có thể kiểm thử được

Kết quả của giai đoạn này là tài liệu SRS, đóng vai trò là cơ sở cho toàn bộ hoạt động thiết kế kiểm thử về sau.

Tùy quá trình phân tích yêu cầu, nhóm xác định hệ thống có các tác nhân chính:

- Khách hàng
- Quản trị viên
- Nhân viên kho
- Nhà cung cấp
- Đơn vị vận chuyển

Hệ thống được phân rã thành các module chức năng chính:

- Xác thực & phân quyền
- Danh mục sản phẩm

- Giỏ hàng – Đặt hàng – Thanh toán
- Quản lý đơn hàng
- Quản lý tồn kho

Các yêu cầu nghiệp vụ ở giai đoạn này được chuyển hóa thành Test Scenario và tiêu chí nghiệm thu, đảm bảo mỗi yêu cầu đều có ít nhất một kịch bản kiểm thử tương ứng.

4.2.2. Thiết kế hệ thống - 2a

Sau khi yêu cầu đã được xác định rõ, nhóm tiến hành thiết kế hệ thống ở mức tổng thể. Nội dung thiết kế bao gồm:

- Các sơ đồ Use Case
- Các luồng nghiệp vụ chính và luồng ngoại lệ
- Quan hệ giữa các tác nhân và hệ thống

Từ các sơ đồ Use Case và quy trình nghiệp vụ (Business Process), nhóm tiến hành:

- Xây dựng **Business Process Steps**
- Ánh xạ từng bước nghiệp vụ sang **Scenario ID**
- Thiết kế **Test Workflow** cho từng luồng nghiệp vụ quan trọng

Các Test Workflow được thiết kế cho:

- Quy trình mua hàng của khách hàng
- Quy trình xử lý đơn hàng
- Quy trình quản lý tồn kho
- Các luồng ngoại lệ như hết hàng, hủy đơn, giao hàng thất bại

4.2.3. Thiết kế kiến trúc - 3a

Trong giai đoạn thiết kế kiến trúc, nhóm áp dụng C4 Model (Context – Container – Component – Code) để phân rã hệ thống từ mức tổng thể đến mức thành phần.

Thiết kế kiến trúc giúp:

- Xác định rõ ranh giới giữa frontend, backend và database
- Xác định các điểm giao tiếp giữa các module
- Làm cơ sở cho việc thiết kế kiểm thử tích hợp

Tùy thiết kế này, các điểm cần kiểm thử tích hợp được xác định rõ ràng, ví dụ:

- Controller ↔ Service
- Service ↔ Repository
- Backend ↔ Database

4.2.4. Thiết kế module - 4a

Ở bước thiết kế module, nhóm tiến hành phân tích chi tiết từng module chức năng thông qua:

- Class Diagram
- Quan hệ giữa các lớp
- Thuộc tính và phương thức chính

Kết quả của bước này được sử dụng làm cơ sở để xây dựng kiểm thử đơn vị (Unit Test) cho các lớp và phương thức trọng yếu trong từng module.

4.2.5. Kiểm thử đơn vị - 1b

Test Case ID	Unit Under Test	Test Description	Test Data	Expected result	Result
UT-CTL-AUTH-001	POST /api/auth/login	Đăng nhập thành công	Body: { username: "jasmine", password: "123123123" } Mock: authenticationManager authenticate("jasmine", "123123123") → Authentication	HTTP 200 OK JSON trả về jwt + refreshToken	Passed
UT-CTL-AUTH-002	POST /api/auth/login	Đăng nhập thất bại do sai mật khẩu	Body: {username: "jasmine", password: "12345"} Mock: authenticationManager authenticate("jasmine", "123123123") → authenticate throw exception	HTTP 401 Unauthorized	Passed
UT-CTL-AUTH-003	POST /api/auth/refresh-token	Refresh token thành công	Body: refreshToken hợp lệ Body: { refreshToken: "refresh-xyz" } Mock: refreshTokenService.findByToken("refresh-xyz") → RefreshToken	HTTP 200 OK JWT mới được trả về	Passed
UT-CTL-AUTH-004	POST /api/auth/refresh-token	Refresh token hết hạn	Body: refreshToken expired Body: { refreshToken: "refresh-expired" } Mock: refreshTokenService.findByToken("refresh-expired") → Optional.empty()	HTTP 403 Forbidden	Passed

4.2.6. Kiểm thử tích hợp - 2b

Sau khi các module đã ổn định qua kiểm thử đơn vị, nhóm tiến hành kiểm thử tích hợp nhằm đánh giá sự tương tác giữa các thành phần trong hệ thống.

Nhóm sử dụng:

- Spring Boot Test
- MockMvc
- H2 Database (in-memory)

Các ca kiểm thử tích hợp tập trung vào:

- Luồng xử lý API
- Truyền dữ liệu giữa các tầng
- Tính nhất quán dữ liệu giữa backend và database

Ví dụ kiểm thử tích hợp của dự án:

Integration Test Case	Test Method
IT_PROD_01	testGetAllProducts_Pagination()
IT_PROD_02	testSearchProducts_ByKeyword()
IT_PROD_03	testFilterProducts_ByCategory()
IT_PROD_04	testGetNewestProducts()
IT_PROD_05	testGetProductDetail_Success()
IT_PROD_06	testGetProductDetail_NotFound()
IT_PROD_07	testGetRelatedProducts()
IT_CAT_01	test GetAllCategories_Client()
IT_CAT_02	testGetCategoryBySlug_Success()

4.2.7. Kiểm thử hệ thống - 3b

Ở cấp độ cao hơn, nhóm tiến hành kiểm thử hệ thống nhằm đánh giá toàn bộ hoạt động của phần mềm như một thể thống nhất. Kiểm thử hệ thống được thực hiện theo cách tiếp cận thủ công dựa trên tập Test Case đã được xây dựng ở giai đoạn thiết kế kiểm thử, bao gồm đầy đủ các tình huống theo use case và luồng nghiệp vụ chính của người dùng.

Các hoạt động kiểm thử bao gồm:

- Kiểm thử chức năng: kiểm tra các luồng nghiệp vụ từ đầu đến cuối (end-to-end) như đăng nhập, mua hàng, thanh toán...
- Kiểm thử giao diện: đảm bảo giao diện trực quan, nhất quán và đáp ứng tiêu chuẩn hiển thị.
- Kiểm thử bảo mật cơ bản: xác minh luồng đăng nhập, phân quyền và kiểm soát truy cập.

Kết quả kiểm thử hệ thống phản ánh chất lượng tổng thể của phần mềm trước khi chuyển sang giai đoạn nghiệm thu.

4.2.8. Kiểm thử chấp nhận - 4b

Kiểm thử chấp nhận là giai đoạn cuối cùng của quy trình kiểm thử, nhằm đánh giá hệ thống dưới góc nhìn người dùng thực tế. Ở bước này, nhóm đổi chiều các tiêu chí nghiệm thu với tài liệu SRS để đảm bảo hệ thống đáp ứng đầy đủ yêu cầu của khách hàng.

Sau khi hoàn thành kiểm thử hệ thống nội bộ, phần mềm được bàn giao cho Product Owner tiến hành kiểm thử chấp nhận. Product Owner sẽ đóng vai trò người dùng cuối, mô phỏng các tình huống thực tế và đưa ra đánh giá về mức độ đáp ứng nhu cầu, tính tiện dụng, độ ổn định và mức độ phù hợp với nghiệp vụ.

Kết quả kiểm thử chấp nhận là căn cứ quan trọng để đưa ra quyết định cuối cùng về việc triển khai hệ thống lên môi trường vận hành.

4.3. Kỹ thuật thiết kế kiểm thử

4.3.1. Kiểm thử hộp đen

Kiểm thử hộp đen được sử dụng để kiểm tra chức năng của hệ thống dựa trên yêu cầu nghiệp vụ, mà không cần quan tâm đến cấu trúc bên trong của phần mềm.

Trong dự án, phương pháp này áp dụng cho kiểm thử hệ thống và kiểm thử chấp nhận, vì mục tiêu là xác minh phần mềm hoạt động đúng như kỳ vọng của người dùng cuối và đáp ứng các tiêu chí nghiệm thu.

Việc sử dụng hộp đen giúp nhóm mô phỏng các tình huống thực tế và kiểm tra toàn bộ luồng nghiệp vụ từ đầu đến cuối, phù hợp với thực hiện thủ công hoặc tự động hóa ở mức giao diện.

Phân vùng tương đương:

Trong test case Access Control, dữ liệu đầu vào được phân chia thành các lớp hợp lệ và không hợp lệ dựa trên trạng thái tài khoản và tính hợp lệ của thông tin người dùng. Cụ thể:

Chức năng đăng nhập

Lớp hợp lệ:

- Tên tài khoản tồn tại
- Mật khẩu đúng
- Tài khoản đang hoạt động

Lớp không hợp lệ:

- Sai mật khẩu
- Tài khoản bị khóa
- Tên tài khoản không tồn tại

Chức năng đăng ký

Lớp hợp lệ:

- Tên tài khoản chưa tồn tại
- Email đúng định dạng
- Nhập đầy đủ các trường bắt buộc

Lớp không hợp lệ:

- Email đã tồn tại
- Bỏ trống trường bắt buộc
- Email / số điện thoại sai định dạng

Ánh xạ vào test case thực tế

Kỹ thuật này được áp dụng trong các test case:

- TC_AC_LOGIN_VALID_*
- TC_AC_LOGIN_INVALID_*
- TC_AC_REGISTER_VALID_*
- TC_AC_REGISTER_INVALID_*

Kỹ thuật chuyển trạng thái:

Trong module Access Control, các trạng thái quan trọng bao gồm:

Trạng thái tài khoản người dùng gồm bình thường và bị khóa.

Trạng thái phiên làm việc gồm đã đăng nhập và đã đăng xuất.

Các test case được xây dựng để kiểm tra: Việc chuyển trạng thái từ đã đăng nhập → đăng xuất, Việc chuyển trạng thái tài khoản từ bình thường → bị khóa → mở khóa, Xác minh hành vi hệ thống sau khi trạng thái thay đổi.

Mình chứng test case: TC_AC_007: đăng xuất thành công, TC_AC_017: khóa tài khoản người dùng, TC_AC_018: kiểm tra đăng nhập bằng tài khoản đã bị khóa, TC_AC_019: mở khóa tài khoản người dùng.

Bảng quyết định:

Bước 1: Xác định điều kiện.

Ký hiệu	Điều kiện	Giá trị
C1	Thông tin đăng nhập hợp lệ	Y / N
C2	Tài khoản bị khóa	Y / N
C3	Vai trò người dùng	Khách hàng / Quản trị viên
C4	Trang truy cập	User / Admin

Bước 2: Xác định hành động.

Ký hiệu	Hành động
A1	Cho phép đăng nhập
A2	Cho phép truy cập trang

A3	Hiển thị thông báo lỗi
----	------------------------

Bước 3: Tính số luật.

- $2 \times 2 \times 3 \times 2 = 24$ rules
- Mỗi rules đại diện cho một tổ hợp điều kiện có thể xảy ra.

Bước 4: Bảng quyết định đầy đủ.

(N: No, Y: Yes, DC: Don't Care, KH: Khách hàng, QTV: Quản trị viên)

Rule	R1	R2	R3	R4	R5	R6	R7	R8
C1: Đăng nhập hợp lệ	N	N	N	N	Y	Y	Y	Y
C2: Tài khoản bị khóa	DC	DC	DC	DC	Y	Y	N	N
C3: Chức vụ	DC	DC	DC	DC	KH	QTV	KH	QTV
C4: Trang	Public	QTV	Public	QTV	Public	QTV	QTV	QTV
A1: Đăng nhập thành công	N	N	N	N	N	N	Y	Y
A2: Truy cập được cho phép	N	N	N	N	N	N	N	Y
A3: Lỗi đã xuất hiện	Y	Y	Y	Y	Y	Y	Y	N

DC (Don't Care): điều kiện không ảnh hưởng đến kết quả

- Nếu thông tin đăng nhập không hợp lệ → luôn bị từ chối (R1–R4)
- Nếu tài khoản bị khóa → đăng nhập thất bại (R5–R6)
- Nếu Customer truy cập trang Admin → bị từ chối (R7)
- Nếu Admin truy cập trang Admin → cho phép (R8)

Bước 5: Rút gọn bảng quyết định.

Rule	C1	C2	C3	C4
R1	N	DC	DC	DC
R2	Y	Y	DC	DC
R3	Y	N	Khách hàng	Admin
R4	Y	N	Quản trị viên	Admin

Mapping Rule → Test Case

Rule	Test Case
R1	TC_AC_LOGIN_INVALID
R2	TC_AC_LOGIN_DISABLED
R3	TC_AC_ACCESS_DENIED
R4	TC_AC_ACCESS_ALLOWED

4.3.2. Kiểm thử hộp trắng

Kiểm thử hộp trắng được sử dụng để kiểm tra logic bên trong code, luồng dữ liệu, nhánh điều kiện và vòng lặp, giúp phát hiện lỗi ở mức module hoặc phương thức riêng lẻ.

Trong dự án, phương pháp này áp dụng cho kiểm thử đơn vị và kiểm thử tích hợp, vì mục tiêu là đảm bảo từng thành phần hoạt động chính xác theo thiết kế và các module tương tác đúng.

Việc sử dụng hộp trắng giúp nhóm phát hiện lỗi sớm, giảm rủi ro khi tích hợp, và thuận tiện cho kiểm thử tự động.

a. Phân tích hàm deleteAddress

Node	Mô tả
1	Kiểm tra tồn tại địa chỉ
2	Ném exception
3	Thực hiện xóa
4	Kết thúc
Edge	Điễn giải
1 → 2	Không tồn tại
1 → 3	Tồn tại
2 → End	Kết thúc do lỗi
3 → 4	Kết thúc thành công

4.4. Phương pháp thiết kế kiểm thử

4.4.1. Kiểm thử thủ công

Dự án áp dụng kiểm thử thủ công cho giai đoạn kiểm thử chấp nhận. Phương pháp này được sử dụng nhằm đánh giá hệ thống dưới góc nhìn người dùng cuối, kiểm tra các luồng nghiệp vụ chính và xác nhận mức độ đáp ứng của hệ thống so với các yêu cầu đã đặc tả.

Các ca kiểm thử thủ công được thực hiện dựa trên Test Case và Review Checklist đã thiết kế, kết quả kiểm thử được ghi nhận và đối chiếu với kết quả mong đợi trước khi tiến hành nghiệm thu hệ thống.

Dùng công cụ Postman để kiểm thử các api

API_PROD_02

The screenshot shows the Postman interface with the following details:

- Method:** GET
- URL:** <http://localhost:8085/client-api/products?search=laptop>
- Params:** search: laptop, page: 1
- Body:** (Empty)
- Headers:** (14)
- Test Results:** (Empty)
- Response Status:** 200 OK
- Response Time:** 231 ms
- Response Size:** 3.44 KB
- Save Response:** (Enabled)
- Content:** (JSON) [5] (page: 1, size: 5, totalElements: 94, totalPages: 19, last: false)

API_PROD_03

HTTP electro / Filter

GET http://localhost:8085/client-api/products?category=1

Send

Docs Params Authorization Headers (7) Body Scripts Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
category	1		
Key	Value	Description	

Body Cookies Headers (14) Test Results

200 OK 1.95 s 3.44 KB Save Response

{ } JSON Preview Visualize

content [5]

page	1
size	5
totalElements	101
totalPages	21
last	false

API_PROD_07

HTTP electro / ProductDetail

GET http://localhost:8085/client-api/products/ealdus0

Send

Docs Params Authorization Headers (7) Body Scripts Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
productId	1		

Body Cookies Headers (14) Test Results

200 OK 548 ms 4.12 KB Save Response

{ } JSON Preview Visualize

productDetail [1]

productName	Dell XPS 13 9315
productSlug	ealdus0
productShortDescription	Pellentesque ultrices mattis odio. Donec vitae nisi.
productDescription	Praesent blandit. Nam nulla. Integer pede justo, lacinia eget, tincidunt eget, tempus vel, pede.
productImages [3]	
productCategory {3}	
productBrand {2}	
productSpecifications {2}	
productVariants [3]	
productSaleable	true
productAverageRatingScore	4

API_PROD_05

The screenshot shows the Postman interface with a GET request to `http://localhost:8085/client-api/products?newable=true`. The 'Params' tab is selected, showing two entries: `newable` with value `true` and `Key` with value `Description`. The response section shows a `200 OK` status with a page size of 5.

Key	Value	Description	Bulk Edit
<input checked="" type="checkbox"/> newable	true		
<input checked="" type="checkbox"/> Key	Description		

Body Cookies Headers (14) Test Results ⚙️ 200 OK • 295 ms • 3.44 KB ⓘ Save Response ⚙️

{ } JSON ▶ Preview 📐 Visualize ↴

> content [5]	
page	1
size	5
totalElements	101
totalPages	21
last	false

4.4.2. Kiểm thử tự động

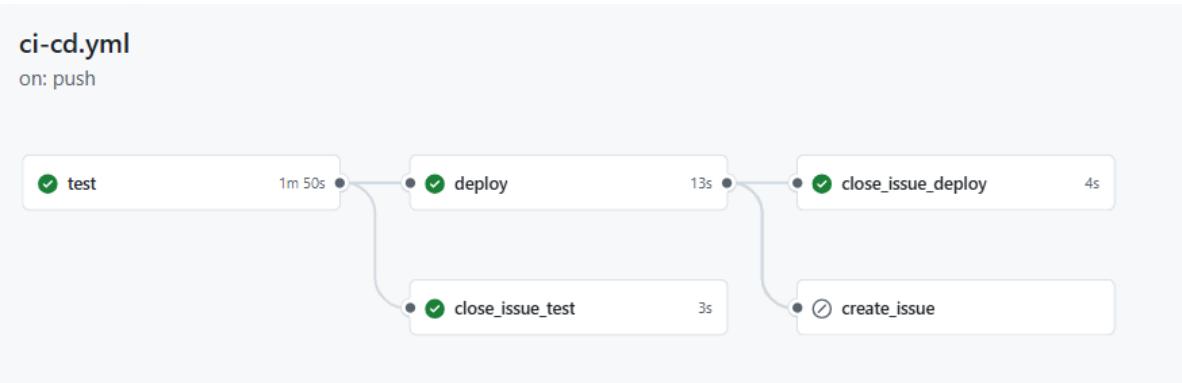
Dự án áp dụng kiểm thử tự động thông qua việc tích hợp quy trình CI/CD bằng GitHub Actions nhằm tự động hóa quá trình build, kiểm thử và triển khai hệ thống. Kiểm thử tự động được sử dụng cho các kiểm thử hộp trắng bao gồm kiểm thử đơn vị và kiểm thử tích hợp và kiểm thử hệ thống.

Mỗi khi có thay đổi mã nguồn được đẩy lên kho lưu trữ GitHub, pipeline CI/CD sẽ tự động kích hoạt job test Job này thực hiện các bước build dự án và chạy toàn bộ các bộ kiểm thử tự động đã được xây dựng.

Nếu job test thất bại, pipeline sẽ tự động tạo Issue trên GitHub để ghi nhận lỗi phát sinh, hỗ trợ nhóm phát triển theo dõi và xử lý kịp thời.

Nếu job test thành công, pipeline sẽ tự động đóng các Issue liên quan đến lỗi kiểm thử trước đó, đảm bảo trạng thái hệ thống luôn được cập nhật chính xác.

Khi mã nguồn được merge vào nhánh chính (Main), pipeline CI/CD sẽ kích hoạt job deploy, thực hiện gọi đến nền tảng Render để tiến hành triển khai lại hệ thống.



Hình 4.3. Pipeline CI/CD của GitHub Actions.

Workflow Run	Event	Status	Branch	Actor	Created At	Duration	Actions
fix github action	Backend CI #3: Commit 5e1d522 pushed by Chin2109	Success	main		Today at 1:39 AM	1m 47s	...
fix github action	Frontend CI #3: Commit 5e1d522 pushed by Chin2109	Success	main		Today at 1:39 AM	43s	...
test github action	Backend CI #2: Commit 6eac73f pushed by Chin2109	Success	main		Today at 1:36 AM	14s	...
test github action	Frontend CI #2: Commit 6eac73f pushed by Chin2109	Success	main		Today at 1:36 AM	15s	...

The screenshot shows the GitHub Actions workflow runs page with six workflow runs listed. The runs are categorized by event (fix github action or test github action) and branch (main). Each run includes details like commit hash, pusher, creation time, duration, and an ellipsis for more options. The runs are all successful (Status: Success).

CHƯƠNG 5. BÁO CÁO KIỂM THỬ

5.1. Tổng quan về quá trình kiểm thử

Sau khi hoàn tất giai đoạn thiết kế kiểm thử ở Chương 4, nhóm đã tiến hành thực hiện kiểm thử cho hệ thống Electro E-Commerce theo đúng kế hoạch đã đề ra. Quá trình kiểm thử được triển khai trong 6 tuần bao gồm các giai đoạn từ kiểm thử đơn vị, kiểm thử tích hợp, kiểm thử hệ thống và kiểm thử chấp nhận.

Nhóm tập trung vào kiểm thử toàn bộ các chức năng nghiệp vụ chính của hệ thống. Các Test Case được thực hiện một cách cẩn thận, chi tiết và được ghi lại đầy đủ kết quả trong các tài liệu bàn giao.

5.2. Báo cáo trường hợp kiểm thử

5.2.1. Giới thiệu

Các trường hợp kiểm thử này được phân loại thành hai nhóm Positive (kiểm tra hành vi đúng với mong đợi của hệ thống) và Negative (kiểm tra khả năng xử lý lỗi, ngoại lệ và điều kiện không hợp lệ).

Bộ kiểm thử được xây dựng cho 6 module chức năng chính của hệ thống gồm kiểm soát truy cập, sản phẩm, giỏ hàng, đơn hàng, thanh toán và số địa chỉ, với tổng cộng 72 trường hợp kiểm thử.

Module	Mã	Số lượng Test Case	Mô tả	Kết quả
Xác thực & phân quyền	AC	23	Đăng ký, đăng nhập, đăng xuất, cập nhật hồ sơ, phân quyền Admin/Buyer	20 Pass 3 Fail
Quản lý sản phẩm & danh mục	PR	19	Xem, lọc, tìm kiếm, thêm/sửa/xóa sản phẩm và danh mục	19 Pass
Giỏ hàng & đơn hàng	OR	18	Giỏ hàng, thanh toán, theo dõi và quản lý đơn hàng	18 Pass
Quản lý tồn kho	INV	20	Nhập kho, xuất kho, lịch sử kho, đơn mua hàng	20 Pass
Tổng cộng		80	Bao phủ các chức năng chính của hệ thống	77 Pass 3 Fail

5.2.2. Phạm vi bao phủ

Nhóm đã xây dựng **80 Test Case**, bao phủ **100% các chức năng chính (Functional Requirements)** của hệ thống.

Các nhóm chức năng được bao phủ bao gồm:

- Xác thực và phân quyền người dùng

- Quản lý người dùng
- Quản lý sản phẩm và danh mục
- Quản lý giỏ hàng và xử lý đơn hàng
- Quản lý tồn kho, nhập – xuất kho và đơn mua hàng

Các yêu cầu phi chức năng như:

- Hiệu năng hệ thống
- Khả năng chịu tải
- Bảo mật nâng cao
- chưa được kiểm thử trong phạm vi đồ án.

Tỷ lệ bao phủ:

- Chức năng (Functional): **100%**
- Phi chức năng (Non-functional): **0%**

5.2.3. Kết quả thực thi

Thống kê chi tiết

- **Tổng số Test Case:** 80
- **Passed:** 77
- **Failed:** 3
- **Blocked:** 0
- **Not run:** 0

Tỷ lệ phần trăm

- **96.25% Test Case thành công (Passed)**
- **3.75% Test Case thất bại (Failed)**
- **0% Test Case bị chặn (Blocked)**
- **0% Test Case chưa chạy (Not run)**