

----- TIÊU CHÍ ĐÁNH GIÁ-----

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

Mục tiêu:

- Phát triển một ứng dụng web hoàn chỉnh (gồm front-end và back-end, ngôn ngữ công nghệ tự chọn).
- Xây dựng kế hoạch kiểm thử toàn diện, tích hợp vào quy trình phát triển Agile và triển khai CI/CD.
- Thực hành kết hợp kiểm thử tự động và kiểm thử thủ công.
- Ứng dụng công cụ GitHub/GitHub Actions trong quản lý mã nguồn, kiểm thử, và triển khai.

Yêu cầu cụ thể:

1. Phát triển ứng dụng web

- Gồm front-end và back-end (framework/ngôn ngữ tùy chọn: React/Angular/Vue, Node.js/Django/Spring Boot, ...).
- Ứng dụng có tối thiểu 3–5 chức năng chính (CRUD hoặc module nghiệp vụ cụ thể).

2. Kế hoạch kiểm thử (Test Plan)

- Viết kế hoạch kiểm thử theo chuẩn Agile (User Story, Test Case, Test Scenario).
- Bao gồm các loại kiểm thử: Unit Test, Integration Test, System Test, Acceptance Test.
- Tích hợp kiểm thử vào pipeline CI/CD.

3. Kiểm thử tự động và thủ công

- Kiểm thử tự động: Sử dụng framework (JUnit, PyTest, Selenium, Cypress, v.v.).
- Kiểm thử thủ công: Viết báo cáo kết quả chạy test case trên môi trường triển khai.

4. Quy trình CI/CD và Agile

- Quản lý phát triển theo Sprint (Agile/Scrum hoặc Kanban).
- Lưu trữ code trên GitHub, sử dụng GitHub Actions để:
 - § Tự động build
 - § Chạy kiểm thử
 - § Triển khai (deploy) ứng dụng sang môi trường staging/production
- Sau mỗi giai đoạn (iteration/sprint), tiến hành đánh giá thủ công trên bản triển khai thực tế.

Kết quả bàn giao:

- Source code trên GitHub (private có chia quyền cho giảng viên qua tài khoản: dntai).
- Tài liệu kế hoạch kiểm thử (Test Plan, Test Cases, Test Report).
- Pipeline CI/CD (workflow GitHub Actions).
- Báo cáo + Slide thuyết trình cuối kỳ (quy trình phát triển, kết quả kiểm thử, đánh giá). (tiếng anh)

MỤC LỤC BÁO CÁO KIỂM THỬ DỰ ÁN SHOESHOP E-COMMERCE

CHƯƠNG 1 – INTRODUCTION

Tổng quan chương

Chương 1 cung cấp bức tranh toàn cảnh về hệ thống Shoeshop dưới góc nhìn KIỂM THỬ, nhằm tạo nền tảng để xây dựng kế hoạch kiểm thử (Test Plan) và phương pháp luận kiểm thử (Test Methodology) ở các chương tiếp theo.

Mục tiêu của chương này không phải mô tả chi tiết hệ thống, mà là xác định đúng và đủ các thông tin cần thiết phục vụ cho hoạt động kiểm thử, bao gồm:

- Mục tiêu và phạm vi dự án → để xác định *phạm vi kiểm thử (test scope)*.
- Quy trình nghiệp vụ → để hiểu *luồng xử lý* và xác định *điểm kiểm thử quan trọng*.
- Yêu cầu nghiệp vụ (User Story, Acceptance Criteria) → để làm *cơ sở xây dựng Test Scenario và Test Case*.
- Yêu cầu chức năng và phi chức năng → để xác định *loại kiểm thử* cần thực hiện (functional, UI, API, performance, security...).
- Quy trình phát triển (Agile, CI/CD) → để xác định *cách kiểm thử được lồng ghép theo sprint và trong pipeline CI/CD*.

Chỉ trình bày 20% thông tin theo hướng phương pháp luận, còn toàn bộ tài liệu chi tiết (user story, SRS, use case, mockup...) sẽ nằm trong Phụ lục.

1.1. Tổng quan đề tài

1.1.1. Bối cảnh và lý do chọn đề tài

Trong bối cảnh chuyển đổi số đang diễn ra mạnh mẽ, thương mại điện tử đã trở thành một phần không thể thiếu trong hoạt động kinh doanh hiện đại, đặc biệt là ở lĩnh vực thời trang và giày dép. Người tiêu dùng ngày nay có xu hướng mua sắm trực tuyến nhiều hơn, nhờ sự tiện lợi trong việc tìm kiếm, so sánh giá, đặt hàng và thanh toán ngay tại nhà. Tuy nhiên, khảo sát thực tế cho thấy nhiều hệ thống bán hàng trực tuyến hiện nay vẫn tồn tại các hạn chế về chất lượng phần mềm, dẫn đến trải nghiệm người dùng chưa tối ưu và rủi ro trong vận hành. Vậy nên để đảm bảo chất lượng phần mềm trước khi được đưa vào vận hành và tạo sự yên tâm cho doanh nghiệp trước khi sử dụng, nhóm đề tài lựa chọn dự án **kiểm thử hệ thống Website bán giày trực tuyến – ShoeShop Store** nhằm đánh giá, kiểm tra và đảm bảo chất lượng phần mềm của một nền tảng thương mại điện tử mô phỏng thực tế. Đề tài tập trung vào việc xây dựng kế hoạch kiểm thử,

thiết kế ca kiểm thử, thực thi và phân tích lỗi, nhằm xác định mức độ ổn định, tính chính xác và khả năng vận hành của hệ thống trước khi đưa vào sử dụng.

1.1.2. Mục tiêu của hệ thống Shoeshop

Hệ thống ShoeShop Store được xây dựng với mục tiêu trở thành một nền tảng thương mại điện tử chuyên biệt cho ngành hàng giày dép, giải quyết các vấn đề như:

- **Đối với khách hàng:** Cung cấp trải nghiệm mua sắm trực tuyến mượt mà, tiện lợi và an toàn, từ khâu tìm kiếm, lựa chọn sản phẩm, quản lý giỏ hàng đến thanh toán (COD, VNPay) và theo dõi đơn hàng
- **Đối với Vận hành (Staff, Manager):** Tối ưu hóa quy trình nghiệp vụ nội bộ, bao gồm quản lý đơn hàng theo thời gian thực, kiểm soát tồn kho chính xác, và quản lý linh hoạt các chương trình khuyến mãi.
- **Đối với Quản trị (Admin):** Đảm bảo an ninh, an toàn dữ liệu thông qua cơ chế phân quyền chi tiết cho bốn vai trò (Customer, Staff, Manager, Admin), và cung cấp công cụ giám sát hoạt động tổng thể.

Mục tiêu của **dự án kiểm thử** là xác minh rằng hệ thống Shoeshop đã đạt được các mục tiêu trên.

1.1.3. Phạm vi dự án và phạm vi kiểm thử

Phạm vi dự án (Project Scope): Phạm vi dự án tập trung vào hệ thống website bán giày ShoeShop Store, bao gồm các chức năng được phát triển cho bốn vai trò: Customer, Staff, Manager và Admin mà tập trung chủ yếu vào 4 chức năng cốt lõi: tìm kiếm sản phẩm, giỏ hàng và thanh toán, đơn hàng, bảo mật và phân quyền. Dự án không bao gồm các hệ thống của bên thứ ba, các hoạt động marketing, hay quy trình logistics vật lý bên ngoài.

Phạm vi kiểm thử (Testing Scope): Phạm vi kiểm thử được xác định bám sát phạm vi dự án, tập trung vào việc xác minh và đánh giá chất lượng của các chức năng cốt lõi.

- **Đối với Customer:** Kiểm thử quy trình người dùng cuối (End-to-End) bao gồm: đăng ký, đăng nhập, tìm kiếm sản phẩm, quản lý giỏ hàng, áp dụng mã giảm giá, thanh toán (COD & VNPay), theo dõi và hủy đơn hàng.
- **Đối với Staff:** Kiểm thử quy trình nghiệp vụ xử lý đơn hàng: xác nhận đơn, cập nhật trạng thái (từ CREATED đến DELIVERED), và kiểm tra tính chính xác của tồn kho khi xử lý
- **Đối với Admin:** Kiểm thử các chức năng quản trị hệ thống: phân quyền, quản lý tài khoản người dùng, và bảo mật truy cập.
- **Đối với Manager:** Kiểm thử các chức năng quản lý nghiệp vụ: tạo và quản lý mã giảm giá, giám sát và can thiệp đơn hàng, xem báo cáo doanh thu

Hoạt động kiểm thử bao gồm cả kiểm thử chức năng (Functional Testing) và kiểm thử phi chức năng (Non-functional Testing)

1.1.4. Quy trình phát triển phần mềm (Agile Scrum)

Dự án được phát triển theo mô hình Agile, cụ thể là phương pháp luận Scrum. Dự án được chia thành các chu kỳ phát triển ngắn (Sprint) với mục tiêu cung cấp các phần chức năng hoàn chỉnh sau mỗi Sprint. Kiểm thử được thực hiện liên tục trong suốt Sprint (Testing in-sprint) để đảm bảo chất lượng, phát hiện lỗi sớm và cung cấp phản hồi nhanh cho đội ngũ phát triển (Dev) trước khi Sprint kết thúc. Cách tiếp cận này giúp giảm thiểu rủi ro và đảm bảo sản phẩm bám sát yêu cầu người dùng

1.1.5. Tổng quan CI/CD và vai trò trong kiểm thử

Dự án áp dụng phương pháp tự động hóa CI/CD các bước trong quy trình phát triển phần mềm, từ lúc mã nguồn được thay đổi cho đến khi được triển khai. **Continuous Integration (CI)** tự động hợp nhất mã nguồn từ nhiều lập trình viên vào một kho lưu trữ chung. Quan trọng hơn, mỗi khi hợp nhất, hệ thống sẽ tự động chạy các bài kiểm thử cơ bản (như Unit Test, Integration Test) để phát hiện xung đột hoặc lỗi sớm. **Continuous Delivery (CD):** Tự động triển khai các thay đổi đã vượt qua CI lên môi trường kiểm thử (Testing/Staging).

CI/CD được sử dụng trong dự án để đảm bảo các thay đổi mới không phá vỡ chức năng cũ (thông qua kiểm thử hồi quy tự động), cung cấp một môi trường Staging ổn định và cập nhật liên tục cho QA thực hiện kiểm thử thủ công (manual testing) các tính năng mới, và rút ngắn đáng kể chu kỳ phản hồi lỗi

1.1.6. Tổng quan kiểm thử phần mềm

Dự án đánh giá hệ thống phần mềm nhằm phát hiện sự khác biệt giữa yêu cầu đã định nghĩa và kết quả thực tế. Mục tiêu không chỉ là tìm lỗi (bug) mà còn là đảm bảo phần mềm đáp ứng các yêu cầu về chức năng, hiệu năng, bảo mật và trải nghiệm người dùng. Các quy trình phân tích yêu cầu, lập kế hoạch kiểm thử (Test Planning), thiết kế kiểm thử (Test Design), thực thi kiểm thử (Test Execution), quản lý lỗi (Bug Management), báo cáo (Reporting) được thực hiện trong suốt vòng đời phát triển sẽ được trình bày chi tiết và cụ thể trong bài báo cáo

1.2. Phân tích yêu cầu hệ thống

1.2.1. Khảo sát hiện trạng và pain point nghiệp vụ

Bối cảnh chung: Trong xu hướng chuyển đổi số mạnh mẽ, thương mại điện tử đã trở thành một phần không thể thiếu trong hoạt động kinh doanh, đặc biệt là trong ngành thời trang và giày dép.

Nhu cầu người dùng: Người tiêu dùng ngày càng có xu hướng mua sắm trực tuyến nhiều hơn nhờ sự tiện lợi trong việc tìm kiếm, so sánh giá, đặt hàng và thanh toán ngay tại nhà.

Mục tiêu dự án: Dự án này được lựa chọn nhằm kiểm thử hệ thống Website bán giày trực tuyến – ShoeShop Store để đánh giá, kiểm tra và đảm bảo chất lượng phần mềm của một nền tảng thương mại điện tử mô phỏng thực tế.

Phương pháp khảo sát: Quá trình khảo sát được thực hiện thông qua việc quan sát quy trình vận hành tại các cửa hàng truyền thống và tham khảo các website thương mại điện tử phổ biến để xác định luồng hoạt động thực tế

1.2.2. Phạm vi chức năng hệ thống (Module khách hàng – Module admin)

Danh mục sản phẩm (Product Catalog Service): Xem danh sách sản phẩm, lọc theo tiêu chí (thương hiệu, kích cỡ, màu sắc, tìm kiếm sản phẩm theo từ khóa. Xem chi tiết sản phẩm, bao gồm mô tả, hình ảnh và tồn kho. Thêm mới, chỉnh sửa, xóa sản phẩm và thông tin liên quan. Quản lý các danh mục, thương hiệu, kích cỡ, màu sắc.

Giỏ hàng (Shopping Cart Service): thêm/sửa/xóa số lượng sản phẩm trong giỏ hàng. Áp dụng mã giảm giá. Tính toán lại tổng tiền giỏ hàng sau khi áp dụng khuyến mãi. Điền thông tin nhận hàng, chọn phương thức vận chuyển. Tích hợp các cổng thanh toán trực tuyến hoặc chọn Thanh toán khi nhận hàng (COD).

Quản lý đơn hàng (Order Service): Xem danh sách các đơn hàng đã đặt. Theo dõi trạng thái xử lý chi tiết của từng đơn hàng. Tiếp nhận, xác nhận, và thay đổi trạng thái đơn hàng từ Khách hàng. Theo dõi, lọc, và tìm kiếm đơn hàng theo nhiều tiêu chí.

Quản lý người dùng (User Management Service): Đăng ký/Đăng nhập/Đăng xuất cho khách hàng và nhân viên. Quản lý/cập nhật hồ sơ cá nhân và thay đổi mật khẩu. Quản trị viên có thể xem và quản lý danh sách người dùng.

Kiểm soát truy cập (Access Control Service): Thiết lập và quản lý các nhóm như Admin, Manager, Staff, Customer. Đảm bảo mỗi vai trò chỉ truy cập được các chức năng theo đúng quyền hạn.

1.2.3. Công nghệ sử dụng (ReactJS – Spring Boot – MySQL – Vercel/Railway)

Hệ thống ShoeShop E-commerce được xây dựng theo kiến trúc Full-stack hiện đại, phân tách rõ ràng giữa Frontend và Backend để đảm bảo tính linh hoạt và khả năng mở rộng.

1. Frontend (Giao diện người dùng):

Thành phần giao diện người dùng được phát triển bằng ReactJS. Đây là một thư viện JavaScript mạnh mẽ, cho phép xây dựng giao diện tương tác cao dưới dạng ứng dụng trang đơn (Single Page Application - SPA). Việc sử dụng ReactJS giúp tối ưu hóa tốc độ tải trang và mang lại trải nghiệm mua sắm mượt mà, thân thiện với người dùng cuối.

2. Backend (Xử lý nghiệp vụ):

Phần xử lý nghiệp vụ cốt lõi được xây dựng trên framework Spring Boot sử dụng ngôn ngữ lập trình Java. Spring Boot chịu trách nhiệm cung cấp các RESTful APIs để quản lý toàn bộ logic hệ thống, bao gồm quản lý sản phẩm, giỏ hàng, tiến trình thanh toán, và các chức năng dành cho quản trị viên. Việc lựa chọn Spring Boot đảm bảo hệ thống có hiệu suất cao, tính bảo mật tốt và dễ dàng mở rộng khi quy mô kinh doanh tăng lên.

3. Cơ sở dữ liệu (Database):

Hệ quản trị cơ sở dữ liệu quan hệ MySQL được sử dụng để lưu trữ dữ liệu bền vững của hệ thống. MySQL được chọn vì tính ổn định, độ tin cậy cao và khả năng xử lý tốt các giao dịch phức tạp liên quan đến đơn hàng, tồn kho và thông tin khách hàng.

4. Triển khai (Deployment):

Ứng dụng được triển khai trên nền tảng đám mây để đảm bảo tính sẵn sàng và dễ dàng quản lý. Cụ thể, giao diện Frontend (ReactJS) được hosting trên Vercel nhờ khả năng tích hợp CI/CD

(Continuous Integration/Continuous Deployment) và hiệu suất phân phối nội dung cao. Phần Backend (Spring Boot) và MySQL được triển khai trên Railway, một nền tảng hosting linh hoạt, hỗ trợ tốt cho kiến trúc microservices/API. Ngoài ra, Docker Compose được sử dụng trong môi trường phát triển để đóng gói và quản lý các thành phần Backend và Database, đảm bảo tính đồng nhất giữa các môi trường làm việc.

1.3. Quy trình nghiệp vụ (Business Process)

(nghiệp vụ chia 4 role : staff, manager, admin, customer)

- tập trung vào 4 quy trình chính:

+ Đăng nhập

+ tìm kiếm sản phẩm

+ quy trình đặt hàng

+ phân quyền.

1.3.3. Sơ đồ nghiệp vụ (Business Process Diagram)

1.4. Yêu cầu nghiệp vụ

1.4.1. User Story (As a... I want... so that...)

1.4.2. Điều kiện chấp nhận (UAC Test)

1.4.3. Yêu cầu chức năng (Functional Requirements)

1.4.4. Yêu cầu phi chức năng (NFR: hiệu năng, bảo mật, khả dụng)

User Story

→ mô tả *nhu cầu của người dùng* ở mức cao.

UAC

→ mô tả *điều kiện để xem User Story là DONE*,

→ *điều kiện thành công nghiệp vụ*,

→ là **đầu vào trực tiếp cho QA viết Test Scenario/Test Case**.

Functional Requirements (FR)

→ được trích từ User Story + UAC

→ mô tả *hệ thống phải làm gì* (system behavior).

Non-Functional Requirements (NFR)

→ mô tả các yêu cầu chất lượng (*performance, security, usability*).

1.5. Kế hoạch triển khai

1.5.1. Kế hoạch phát triển (Development Plan)

Hệ thống ShoeShop Store được phát triển theo mô hình Agile Scrum, với các chu kỳ Sprint. Mỗi Sprint cung cấp một phần chức năng hoàn chỉnh để kiểm thử liên tục. Áp dụng theo mô hình chữ V phân chia việc phát triển phần mềm thành thiết kế, triển khai, tích hợp và kiểm tra chất lượng. Chiến lược có hệ thống này đảm bảo mỗi giai đoạn phát triển đều có một bài kiểm thử rõ ràng, từ đó giúp tạo ra phần mềm chất lượng cao. Việc kiểm thử trong từng giai đoạn giúp phát hiện sớm các lỗi, tăng tốc độ hoàn thành dự án, giảm chi phí và cải thiện chất lượng.

1.5.2. Kế hoạch kiểm thử tổng quan (Overview Test Plan)

Mục đích của hoạt động kiểm thử hệ thống SuperShoe là xác định rõ phạm vi, chiến lược, môi trường, nguồn lực và các deliverables, nhằm đảm bảo toàn bộ nghiệp vụ thương mại điện tử bao gồm Danh mục Sản phẩm, Giỏ hàng, Đơn hàng, Giảm giá, Quản lý Người dùng và Kiểm soát Truy cập hoạt động đúng theo yêu cầu chức năng và phi chức năng. Kế hoạch kiểm thử sử dụng phương pháp kiểm thử dựa trên rủi ro, tập trung vào những khu vực có mức ảnh hưởng cao. Phạm vi kiểm thử bao gồm kiểm thử chức năng của các module Product Catalog, Shopping Cart, Order, Discount, User Management và Access Control; đồng thời bao gồm kiểm thử phi chức năng về hiệu năng, bảo mật, khả năng sử dụng, độ tin cậy, khả năng tương thích, khả năng bảo trì, khả năng mở rộng và khả năng phục hồi của hệ thống. Riêng module báo cáo doanh thu sẽ không được kiểm thử trong phạm vi này. Các chuẩn và công cụ được sử dụng gồm Selenium cho kiểm thử giao diện tự động, JMeter cho kiểm thử tải và hiệu năng, Postman để kiểm thử API và xác minh dữ liệu backend, cùng với JUnit cho unit testing. Toàn bộ chiến lược được triển khai qua bốn giai đoạn chính gồm Unit Test do nhà phát triển thực hiện, Integration Test để kiểm tra sự tương tác giữa các module như Giỏ hàng và Thanh toán, System Test để đánh giá luồng end-to-end từ đăng nhập đến thanh toán, và cuối cùng là UAT để người dùng nghiệm thu hệ thống.

CHƯƠNG 2 – PHÂN TÍCH & THIẾT KẾ HỆ THỐNG

TỔNG QUAN CHƯƠNG 2

Chương 2 trình bày thiết kế hệ thống và kiến trúc kỹ thuật của Shoeshop **với mục tiêu phục vụ trực tiếp cho hoạt động kiểm thử phần mềm**. Nội dung tập trung mô tả cách hệ thống vận hành ở **hai góc nhìn: nghiệp vụ (System Design) và kỹ thuật (Architecture Design)**, qua đó giúp xác định đối tượng kiểm thử, luồng dữ liệu quan trọng và các điểm rủi ro cần được ưu tiên khi thiết kế kịch bản kiểm thử.

Ở mức nghiệp vụ, chương mô tả phân rã chức năng của hệ thống, luồng xử lý chính của khách hàng và quản trị viên, cùng mô hình dữ liệu khái niệm. => giúp kiểm thử viên hiểu rõ hệ thống “làm gì” và “xử lý theo trình tự nào”, làm cơ sở xây dựng Test Scenario và xác định vùng kiểm thử quan trọng.

Ở mức kỹ thuật, chương mô tả kiến trúc hệ thống gồm các lớp Frontend – Backend – Database, cơ chế giao tiếp thông qua API và luồng dữ liệu nội bộ. Thông tin này hỗ trợ việc lựa chọn loại kiểm thử phù hợp như kiểm thử API, kiểm thử tích hợp, kiểm thử cơ sở dữ liệu hay kiểm thử hiệu năng.

Chương 2 đóng vai trò nền tảng cho Chương 3 và Chương 4: các thiết kế, luồng xử lý và kiến trúc trình bày tại đây được sử dụng để xây dựng kế hoạch kiểm thử, chiến lược kiểm thử và phương pháp luận kiểm thử của toàn đề tài. Các mô hình chi tiết như ERD đầy đủ, sequence diagram, API list và mockup giao diện sẽ được trình bày tại Phụ lục.

2.1. Thiết kế hệ thống (System Design – Business View)

2.1.1. Thiết kế chức năng

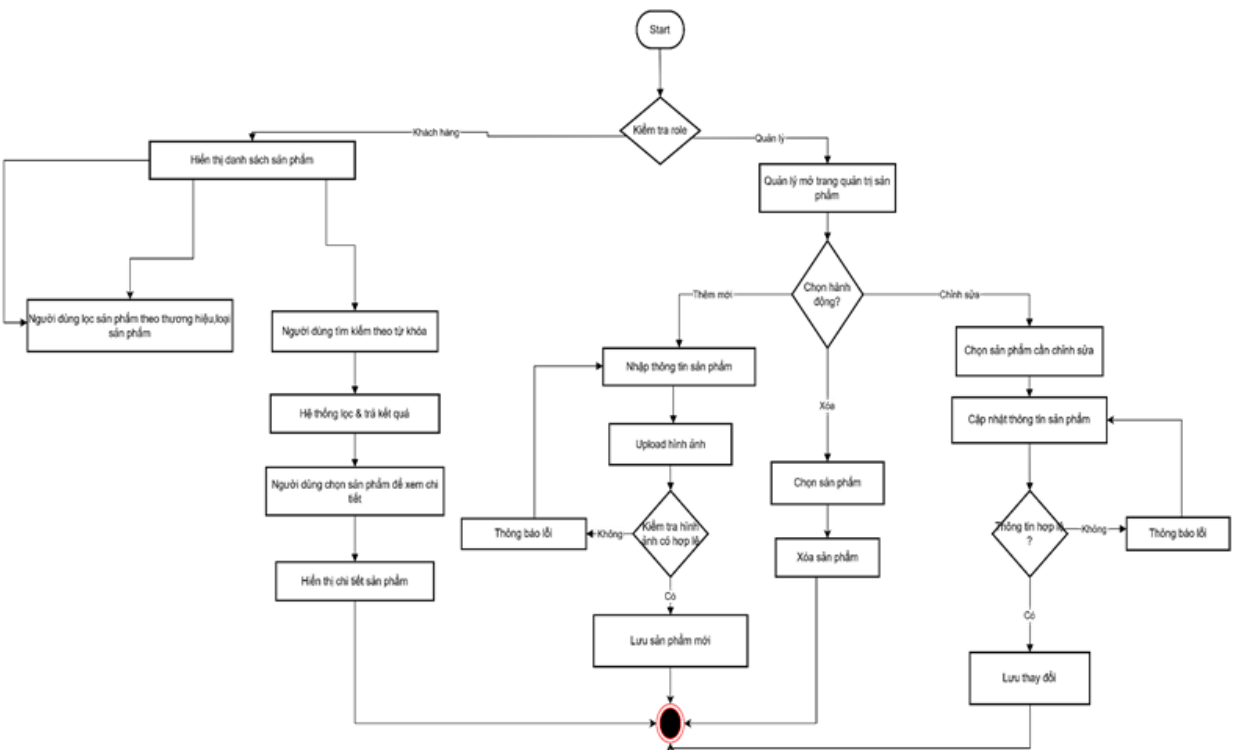
Sơ đồ bối cảnh nghiệp vụ (Business Context)

Mục này trình bày chi tiết về các chức năng nghiệp vụ cốt lõi của hệ thống **ShoeShop E-commerce**.

1. Danh mục sản phẩm (Product Catalog Service)

Module này quản lý và hiển thị tất cả dữ liệu liên quan đến sản phẩm của cửa hàng.

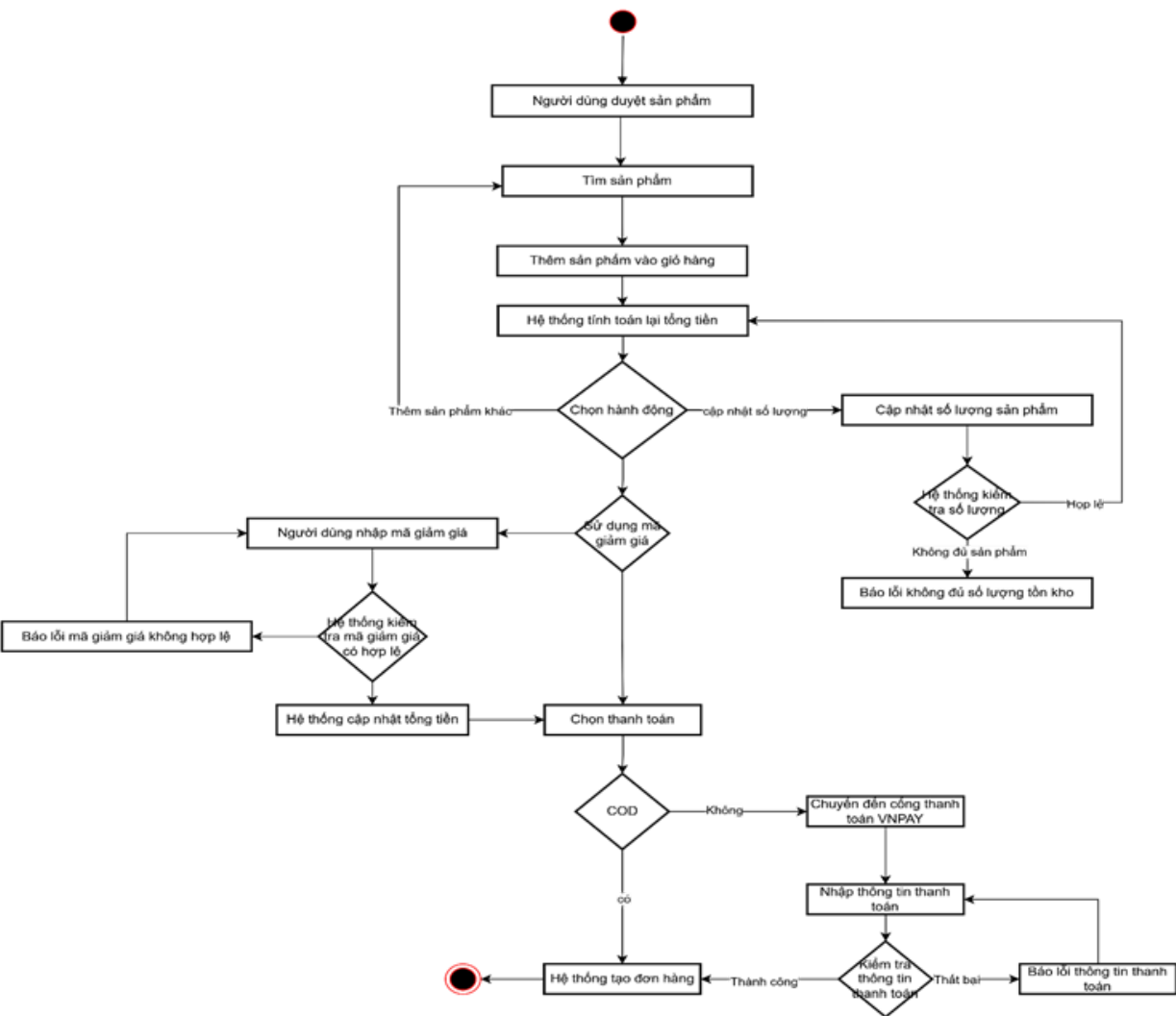
- **Người dùng (Khách hàng) có thể:**
 - Xem danh sách sản phẩm, lọc theo các tiêu chí như **thương hiệu, loại sản phẩm**
 - Thực hiện **tìm kiếm sản phẩm** theo từ khóa.
 - Xem **chi tiết sản phẩm**, bao gồm mô tả, hình ảnh và thông tin về **số lượng tồn kho** (Availability).
- **Quản lý có thể:**
 - Thực hiện các thao tác **Thêm mới, Chỉnh sửa, Xóa** sản phẩm và thông tin liên quan (giá, mô tả, hình ảnh).



2. Giỏ hàng & Thanh toán (Shopping Cart & Checkout Service)

Module này quản lý quy trình chọn sản phẩm, áp dụng khuyến mãi và khởi tạo đơn hàng.

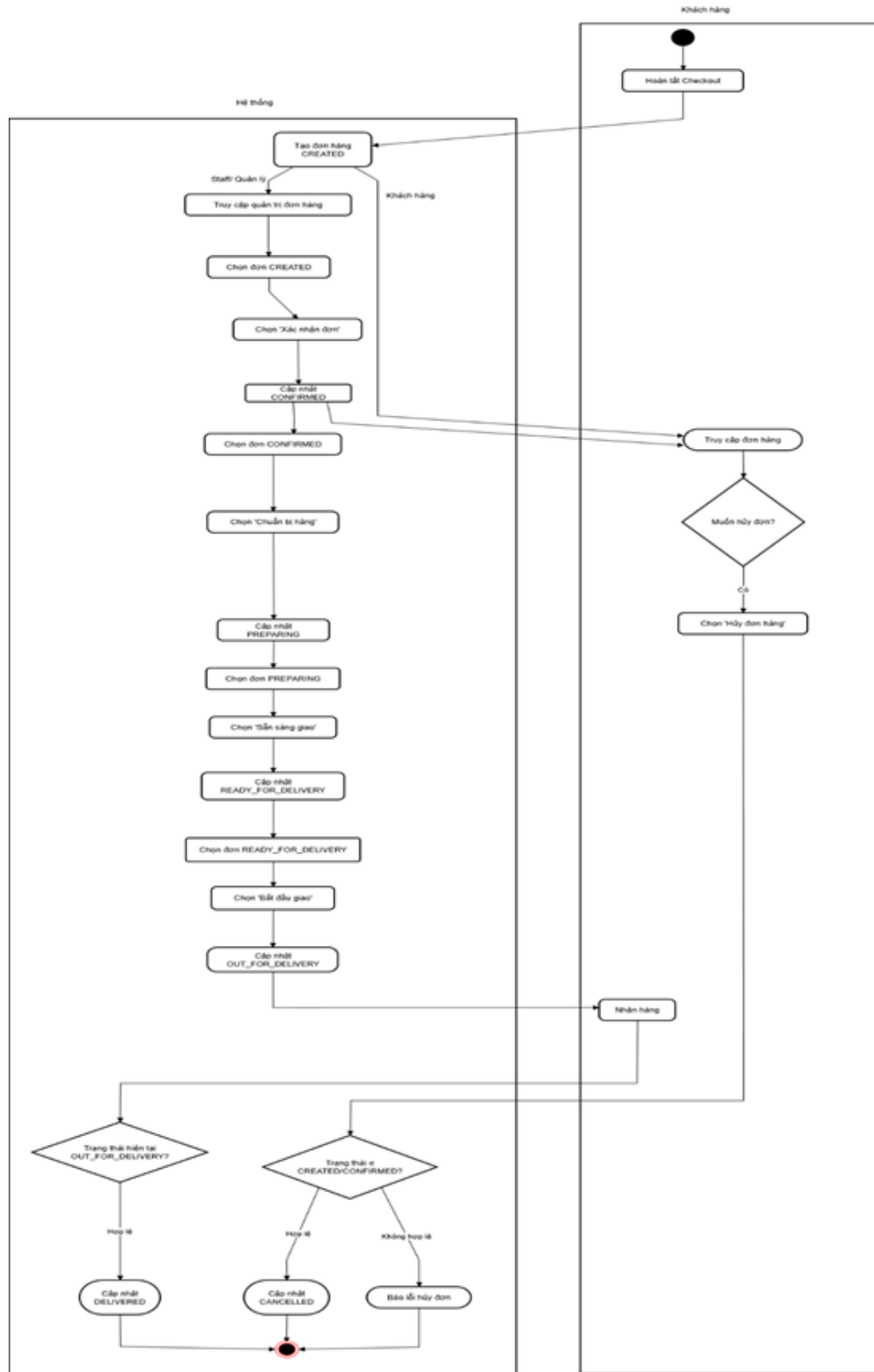
- **Người dùng (Khách hàng) có thể:**
 - Thực hiện các thao tác **Thêm/Sửa/Xóa** số lượng sản phẩm trong giỏ hàng.
 - **Áp dụng mã giảm giá** (Coupon Code).
 - Hệ thống **tính toán lại tổng tiền** giỏ hàng sau khi áp dụng khuyến mãi.
 - **Điền thông tin nhận hàng**, chọn phương thức vận chuyển.
 - Chọn phương thức thanh toán: **Tích hợp các cổng thanh toán trực tuyến** hoặc chọn **Thanh toán khi nhận hàng (COD)**.



3. Quản lý đơn hàng (Order Service)

Module này chịu trách nhiệm cho việc theo dõi, xử lý và cập nhật trạng thái các đơn hàng.

- **Người dùng (Khách hàng) có thể:**
 - Xem danh sách các đơn hàng đã đặt trong lịch sử.
 - Theo dõi trạng thái xử lý chi tiết của từng đơn hàng (ví dụ: Chờ xử lý, Đang giao, Hoàn thành).
 - Hủy đơn hàng khi chưa xác nhận
- **Quản lý/Nhân viên có thể:**
 - **Tiếp nhận, xác nhận, và thay đổi trạng thái đơn hàng** từ Khách hàng.
 - **Theo dõi, lọc, và tìm kiếm** đơn hàng theo nhiều tiêu chí (ngày, trạng thái, khách hàng).



4. Quản lý người dùng (User Management Service)

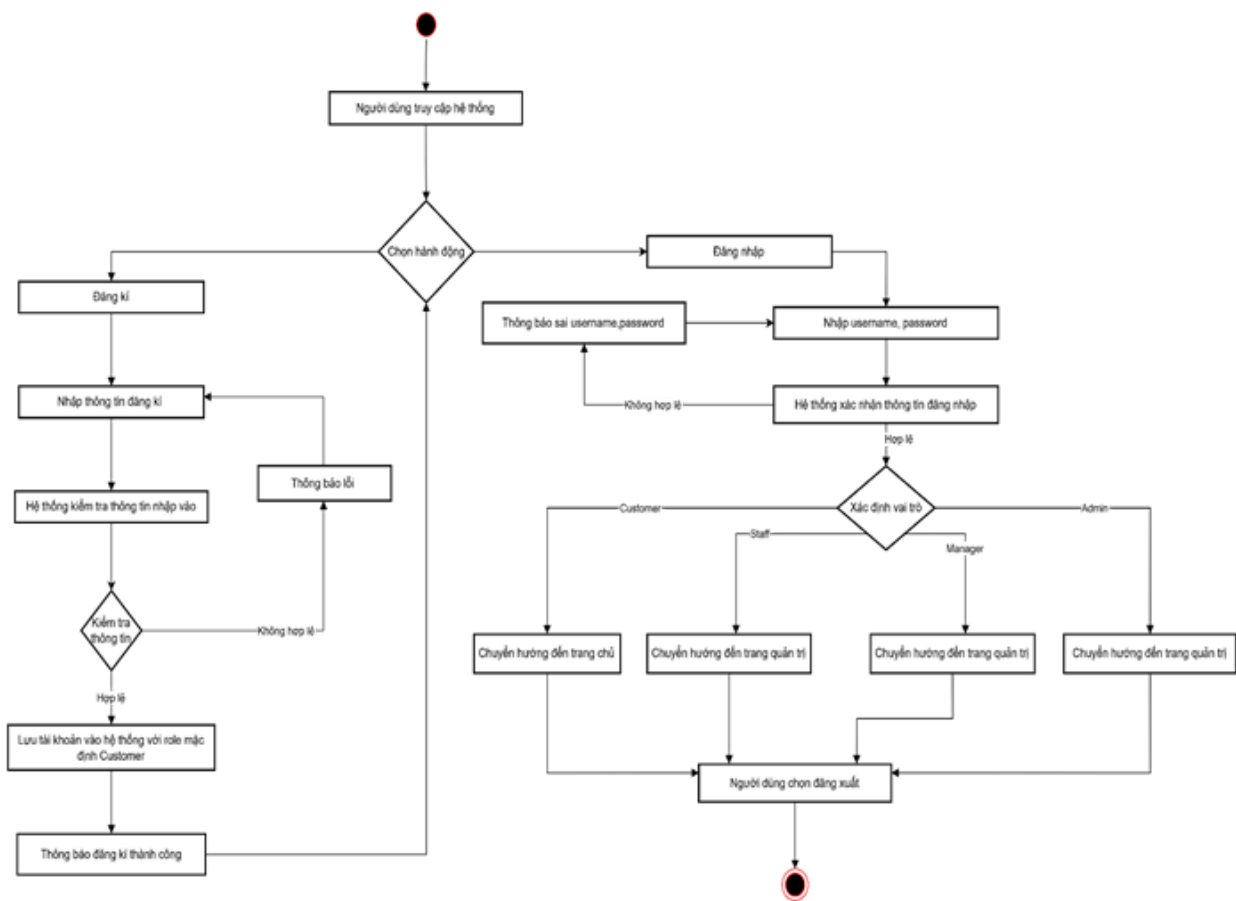
Module này quản lý thông tin tài khoản và hồ sơ người dùng trong hệ thống.

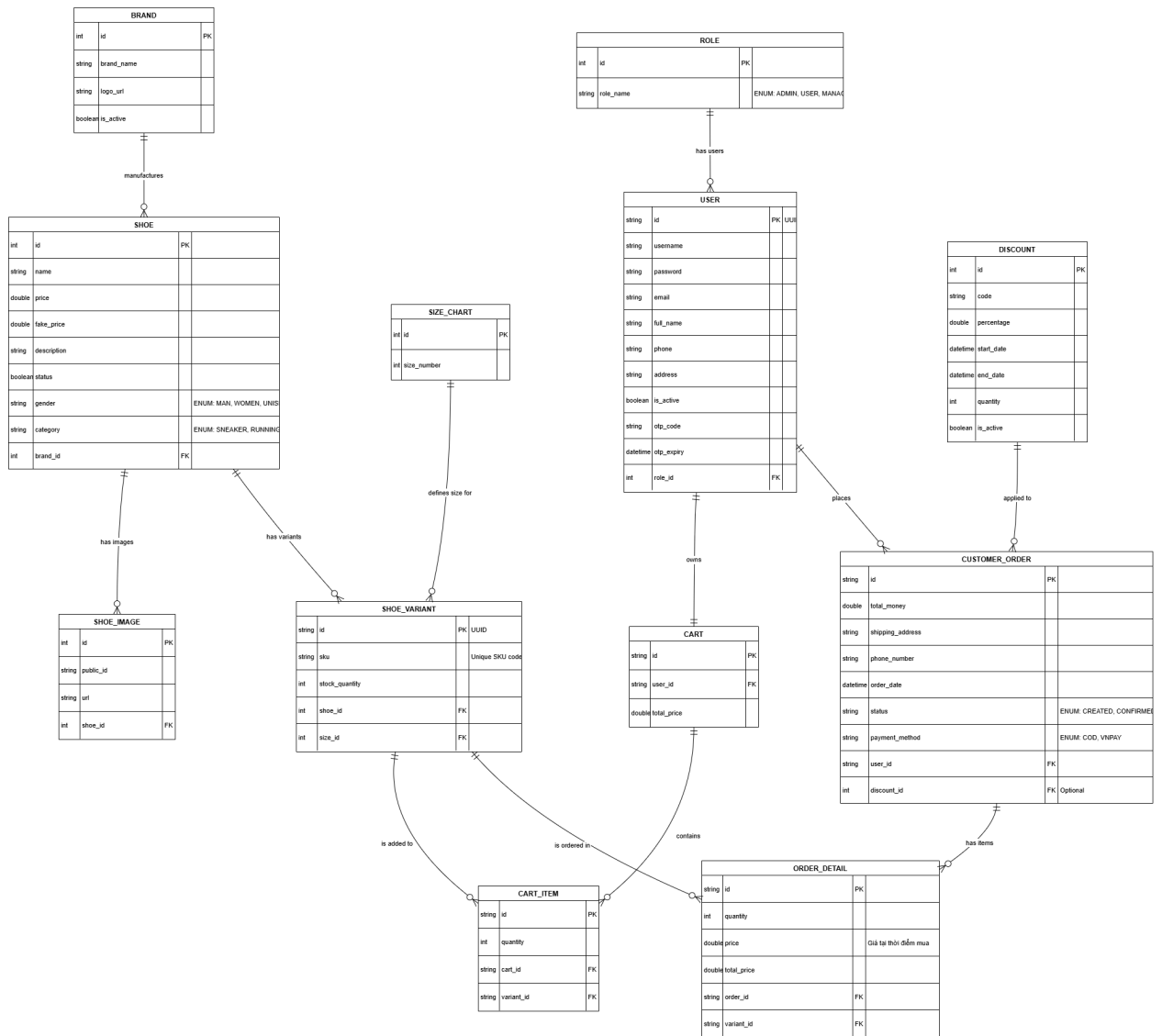
- **Chức năng chung:**
 - Hỗ trợ quy trình **Đăng ký/Đăng nhập/Đăng xuất** cho cả khách hàng và nhân viên.
 - Cho phép người dùng **Quản lý/cập nhật hồ sơ cá nhân** và **thay đổi mật khẩu**.
- **Admin có thể:**
 - **Xem và quản lý** danh sách tài khoản người dùng, bao gồm cả khách hàng và nhân viên.

5. Kiểm soát truy cập (Access Control Service)

Module này đảm bảo tính bảo mật và phân quyền chức năng cho các vai trò khác nhau.

- **Chức năng chính:**
 - **Thiết lập và quản lý các nhóm quyền (Roles)** như **Admin, Manager, Staff, Customer**.
 - Đảm bảo rằng **mỗi vai trò chỉ truy cập được các chức năng** và dữ liệu theo đúng quyền hạn được định nghĩa.





Ghi chú: Data dictionary: file sheet data dictionary

2.1.3. Thiết kế giao diện

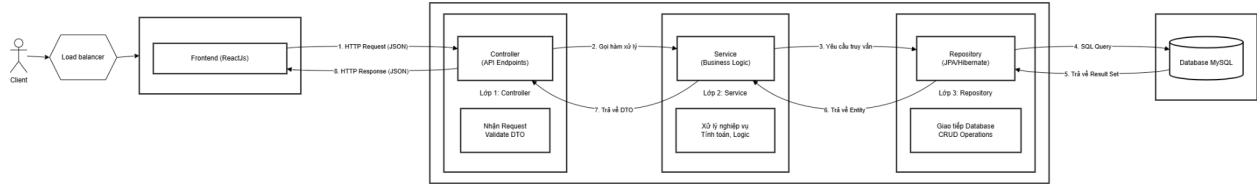
UI Flow (Customer → Cart → Payment → Orders)

Wireframe các trang chính

Mapping UI → Test nhóm giao diện

2.2. Thiết kế kiến trúc (Architecture Design – Technical View)

2.2.1. Tổng quan kiến trúc

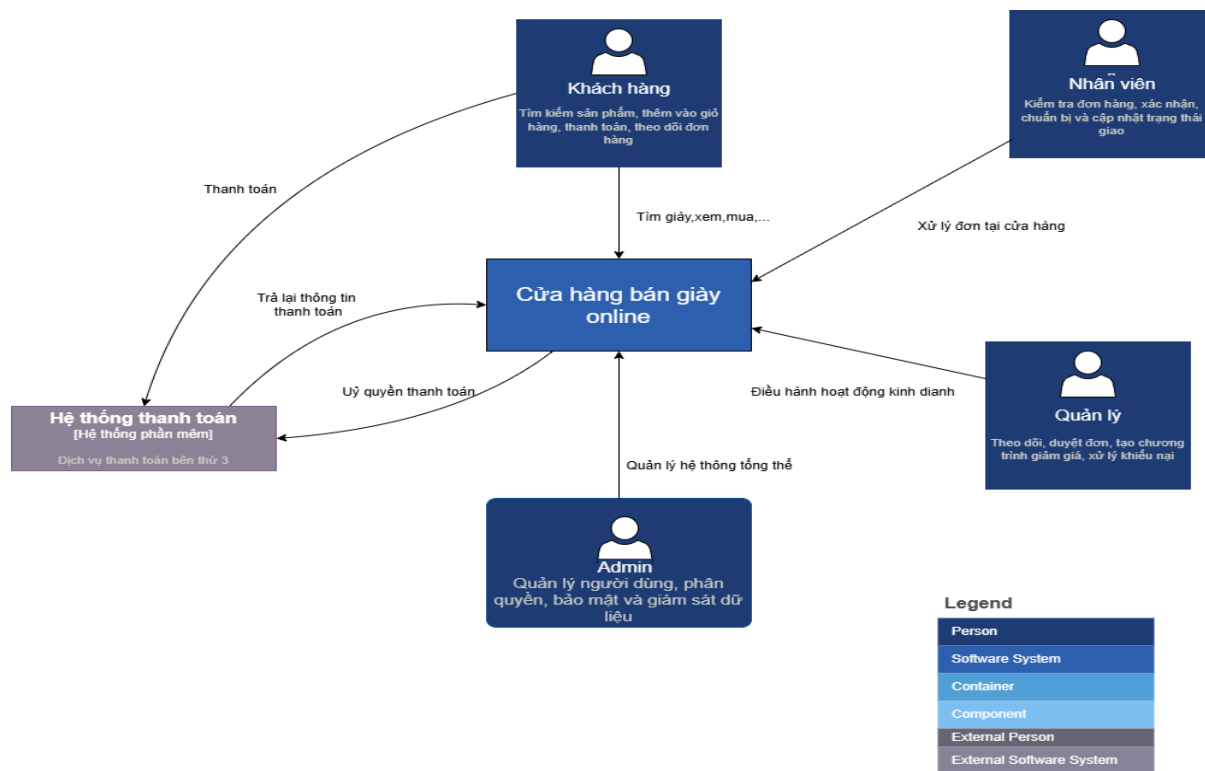


- Luồng đi của dữ liệu sẽ là: Client (Frontend) -> Controller -> Service -> Repository -> Database
- Khối Client (Phía người dùng): Đại diện là trình duyệt web (Browser). Chức năng là gửi yêu cầu HTTP (GET, POST, PUT, DELETE) đến Server, gửi dữ liệu dưới dạng JSON (ví dụ: thông tin đặt hàng, thông tin đăng nhập), nhận phản hồi (Response) từ Server để hiển thị cho người dùng.
- Khối Controller Layer (Tầng Giao diện/Điều hướng): Đại diện trong code: Các class ShoeController, CartController, AuthController. Chức năng là hứng các request từ Client gửi lên, validate dữ liệu đầu vào, không xử lý logic phức tạp, nó chỉ điều phối việc "ai sẽ làm gì". Gọi xuống Service Layer để yêu cầu xử lý nghiệp vụ, trả về kết quả (APIResponse, HTTP Status 200, 400, 500) cho Client.
- Khối Service Layer (Tầng Nghiệp vụ): Đại diện trong code là các class ShoeService, CartService, AuthService, chức năng: Là "Bộ não" của hệ thống. Chứa toàn bộ logic kinh doanh (Business Logic), Ví dụ: Kiểm tra tồn kho, tính tổng tiền giỏ hàng, áp mã giảm giá, mã hóa mật khẩu, chuyển đổi dữ liệu giữa DTO (Data Transfer Object) và Entity, gọi xuống Repository Layer để lấy hoặc lưu dữ liệu
- Khối Repository Layer (Tầng Truy xuất dữ liệu): Đại diện trong code là các interface ShoeRepository, CartRepository, UserRepository. Chức năng: Trục tiếp làm việc với Cơ sở dữ liệu, thực hiện các câu lệnh truy vấn (Query) như: findById, save, delete, findByStatusTrue..., map dữ liệu từ bảng trong Database thành các Java Object (Entity) thông qua Hibernate/JPA.
- Khối Database (Cơ sở dữ liệu): Đại diện: MySQL. Chức năng là lưu trữ bền vững toàn bộ dữ liệu (Bảng User, Shoe, Order, Cart...).
- Luồng dữ liệu (Data Flow) ví dụ cho chức năng "Tạo sản phẩm":
 1. Client: Gửi POST /shoes kèm JSON thông tin giày.
 2. Controller (ShoeController):
 - Nhận request.
 - Kiểm tra @Valid (tên không rỗng, giá > 0).
 - Gọi shoeService.createShoe(request).
 3. Service (ShoeService):
 - Nhận ShoeCreateRequest.
 - Kiểm tra Brand có tồn tại không?

- Chuyển ShoeCreateRequest -> Shoe Entity.
 - Gọi shoeRepository.save(shoe).
4. Repository (ShoeRepository):
 - Sinh câu lệnh INSERT INTO shoe
 - Gửi xuống Database.
 5. Database: Lưu dữ liệu và trả về ID mới tạo.
 6. Service: Nhận lại Entity đã lưu, chuyển thành ShoeResponse.
 7. Controller: Đóng gói ShoeResponse vào APIResponse và trả về 200

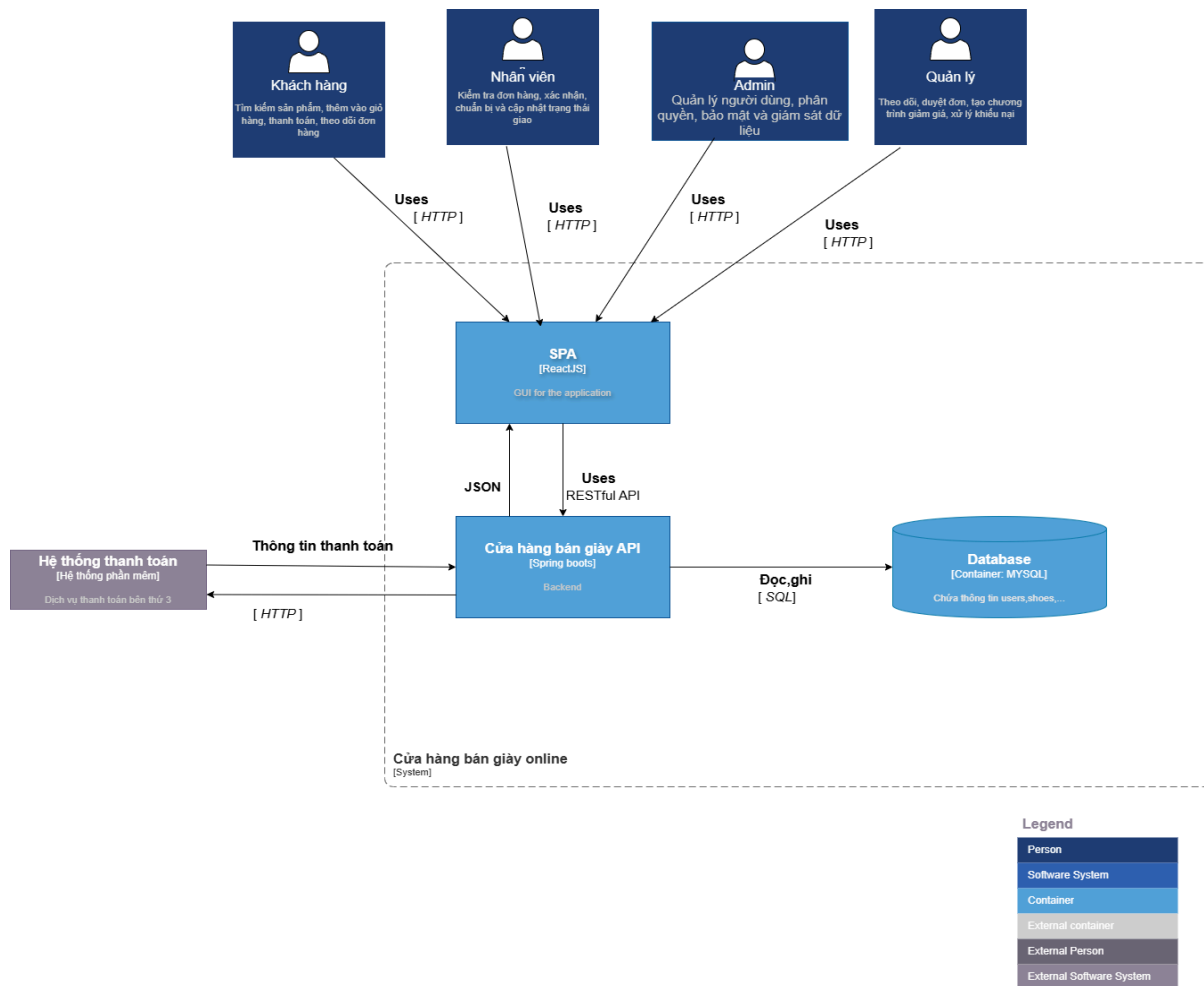
2.2.2. Phân tích thành phần

a) Lược đồ C1



- Hệ thống gồm 4 người dùng là admin, khách hàng, quản lý và nhân viên.
- Customer (Khách hàng) là Người dùng cuối, mua sắm sản phẩm, có thể tìm kiếm sản phẩm, thêm vào giỏ hàng, thanh toán, theo dõi đơn hàng
- Nhân viên cửa hàng là Người xử lý đơn hàng tại cửa hàng, có thể Kiểm tra đơn hàng, xác nhận, chuẩn bị và cập nhật trạng thái giao
- Quản lý cửa hàng là người chịu trách nhiệm điều hành hoạt động kinh doanh có thể theo dõi, duyệt đơn, tạo chương trình giảm giá, xử lý khiếu nại
- Admin là người quản lý hệ thống tổng thể, có thể quản lý người dùng, phân quyền, bảo mật và giám sát dữ liệu

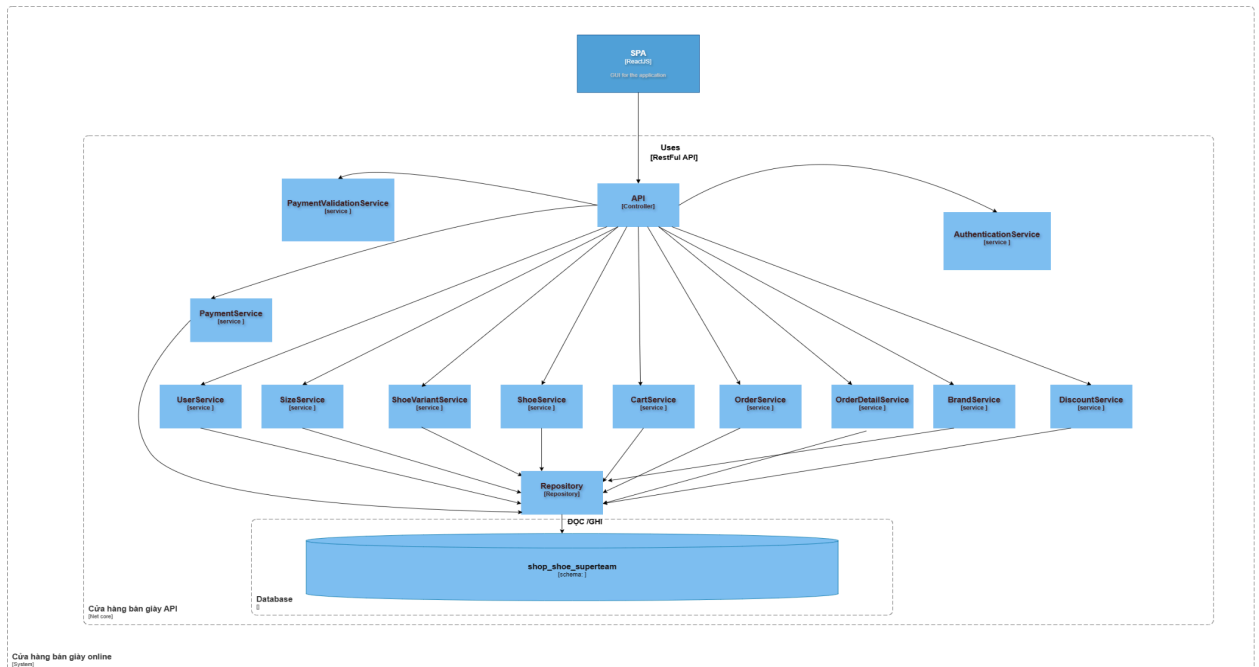
b) Lược đồ C2



- SPA (Single Page Application - Ứng dụng một trang):
 - + Loại: Container (biểu diễn bằng hình chữ nhật màu xanh lam nhạt).
 - + Công nghệ: ReactJS.
 - + Vai trò: Cung cấp Giao diện người dùng đồ họa (GUI) cho toàn bộ ứng dụng. Đây là nơi người dùng (Khách hàng, Nhân viên, Quản lý, Admin) tương tác trực tiếp.
 - + Tương tác: Nhận yêu cầu qua HTTP từ tất cả 4 loại người dùng. Giao tiếp với "Cửa hàng bán giày API" thông qua JSON để gửi/nhận dữ liệu.
- Cửa hàng bán giày API (Backend)
 - + Loại: Container (biểu diễn bằng hình chữ nhật màu xanh lam nhạt).
 - + Công nghệ: Spring Boot (ngôn ngữ Java).

- + Vai trò: Đây là trái tim của hệ thống, xử lý toàn bộ logic nghiệp vụ (business logic) của ứng dụng. Nó cung cấp các dịch vụ RESTful API để SPA có thể gọi và tương tác
- + Tương tác: Nhận yêu cầu (Uses) RESTful API từ SPA. Đọc/Ghi (Đọc/ghi) dữ liệu vào "Database". Tương tác (Thông tin thanh toán) với "Hệ thống thanh toán" bên ngoài
- Các Hệ thống Bên ngoài (External Software System): Hệ thống tương tác với một hệ thống phần mềm bên ngoài là hệ thống thanh toán (Hệ thống ngân hàng):
 - + Loại: External Software System (biểu diễn bằng hình chữ nhật màu xám).
 - + Công nghệ: Dịch vụ thanh toán bên thứ 3.
 - + Vai trò: Xử lý các giao dịch thanh toán.
 - + Tương tác: "Cửa hàng bán giày API" gửi yêu cầu thanh toán (HTTP) tới hệ thống này và nhận lại kết quả.
- Cơ sở dữ liệu (Database)
 - + Loại: Container (biểu diễn bằng hình trụ màu xanh lam đậm).
 - + Công nghệ: MySQL.
 - + Vai trò: Lưu trữ tất cả thông tin về giày, người dùng, đơn hàng, v.v. (Chứa thông tin users, shoes...).
 - + Tương tác: "Cửa hàng bán giày API" là thành phần duy nhất đọc và ghi trực tiếp vào cơ sở dữ liệu.
- Mối quan hệ và Giao thức
 - + Mũi tên: Biểu thị luồng thông tin hoặc sự phụ thuộc.
 - + Nhãn trên mũi tên: Mô tả loại thông tin được truyền tải hoặc giao thức sử dụng (HTTP, JSON, RESTful API, Đọc/ghi, Thông tin thanh toán).

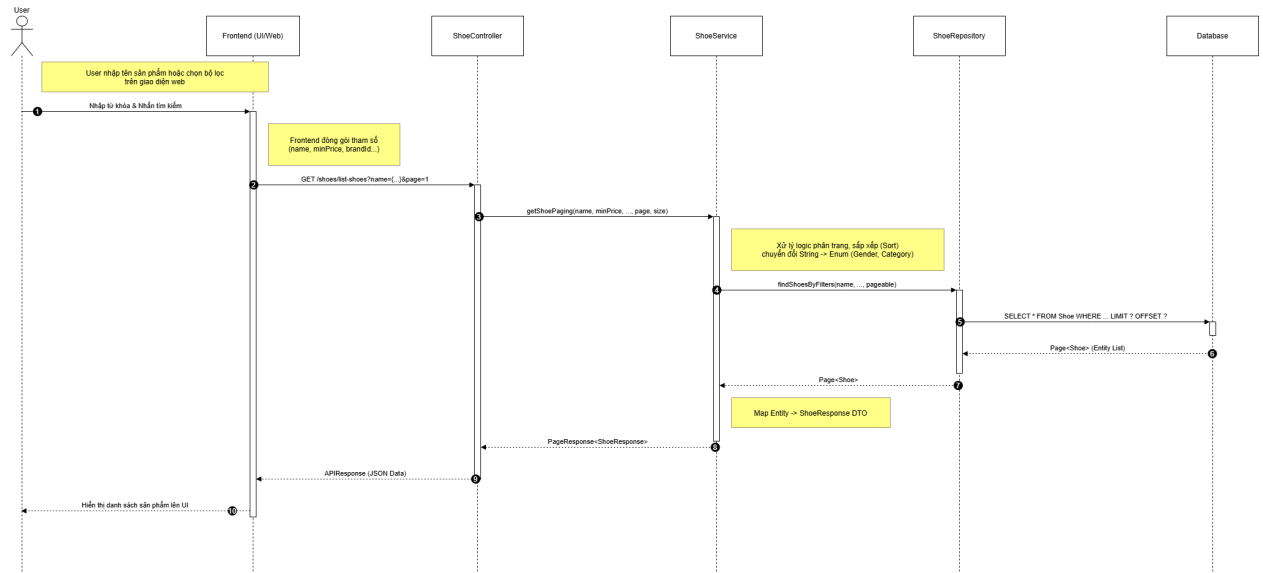
c) Lược đồ C3



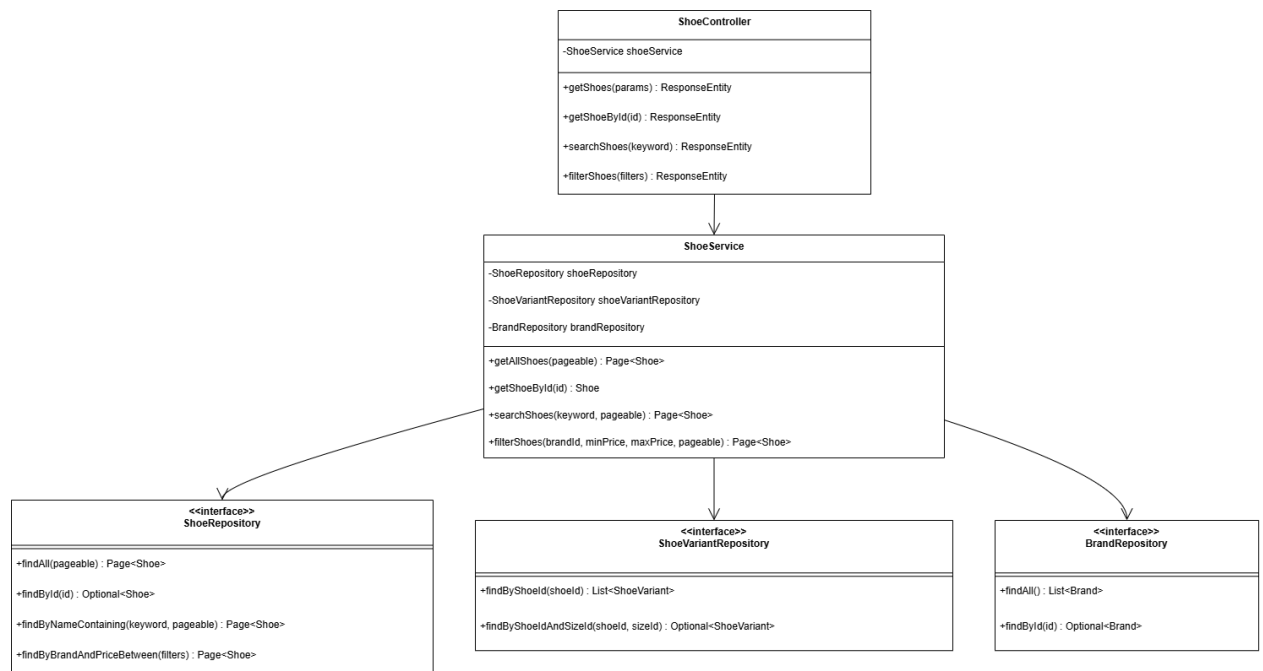
- Luồng dữ liệu từ Trình duyệt Web đi vào Ứng dụng API thông qua giao thức HTTP với định dạng dữ liệu JSON
- Bên trong phần API được chia thành các thành phần Controller riêng biệt xử lý từng nghiệp vụ cụ thể, ví dụ như AuthenticationController chịu trách nhiệm về xác thực, phân quyền, đăng nhập và đăng ký tài khoản. ShoeController quản lý các chức năng liên quan đến sản phẩm giày như xem danh sách, chi tiết và tìm kiếm. CartController xử lý các thao tác với giỏ hàng như thêm sản phẩm, cập nhật số lượng và xóa. OrderController quản lý vòng đời của đơn hàng từ lúc khởi tạo, xác nhận đến khi giao hàng. PaymentController xử lý các giao dịch thanh toán, DiscountController quản lý việc tạo và áp dụng các mã giảm giá cho đơn hàng. UserController cung cấp các chức năng quản lý thông tin cá nhân và tài khoản người dùng. BrandController quản lý thông tin về thương hiệu và phân loại sản phẩm
- Các controller này để thực hiện chức năng của mình sẽ gọi với các service tương ứng, các service muốn thao tác với cơ sở dữ liệu sẽ gọi tới các repository tương ứng.

b) Phân tích luồng dữ liệu

Lược đồ Sequence của tìm kiếm sản phẩm

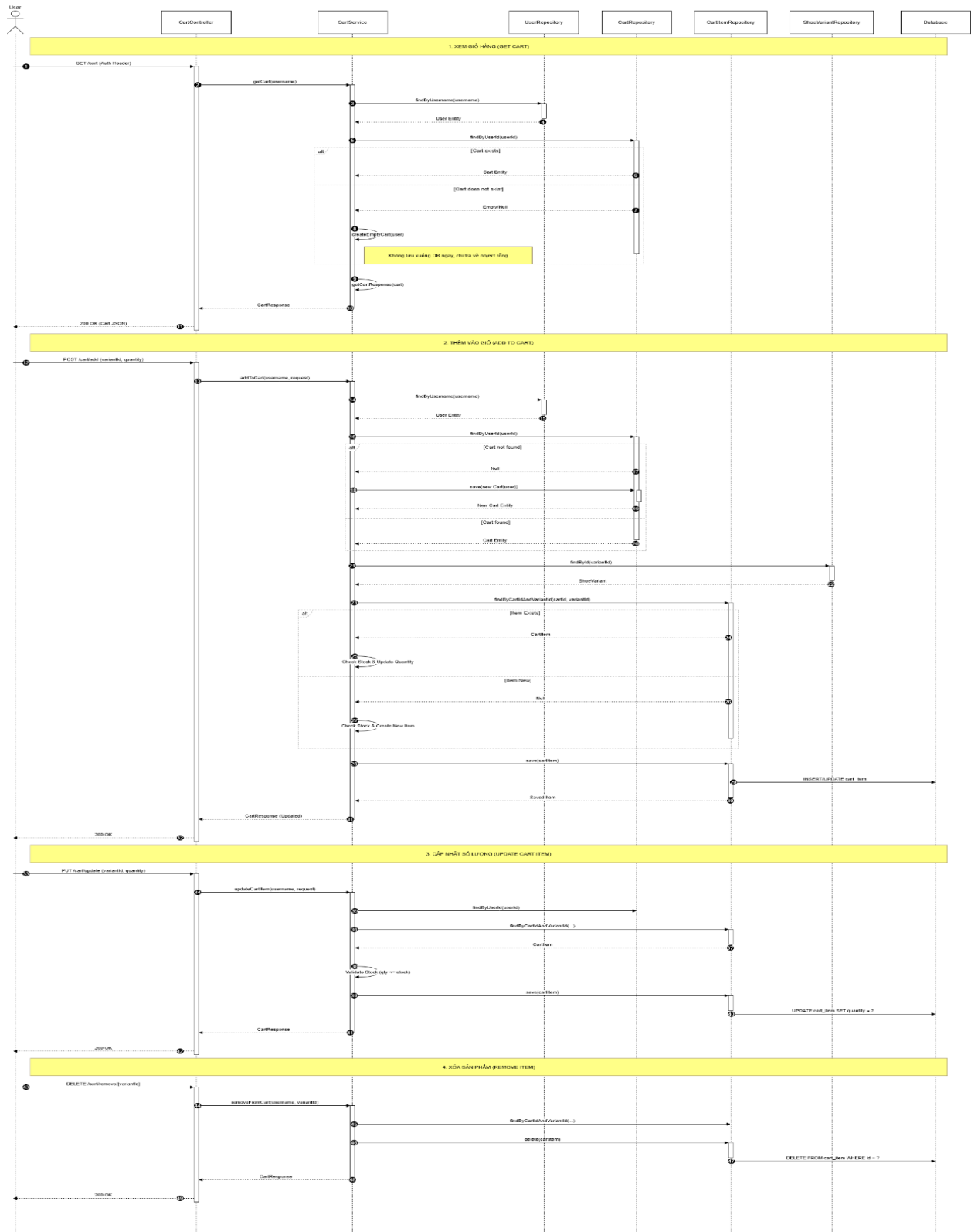


Class diagram của tìm kiếm sản phẩm

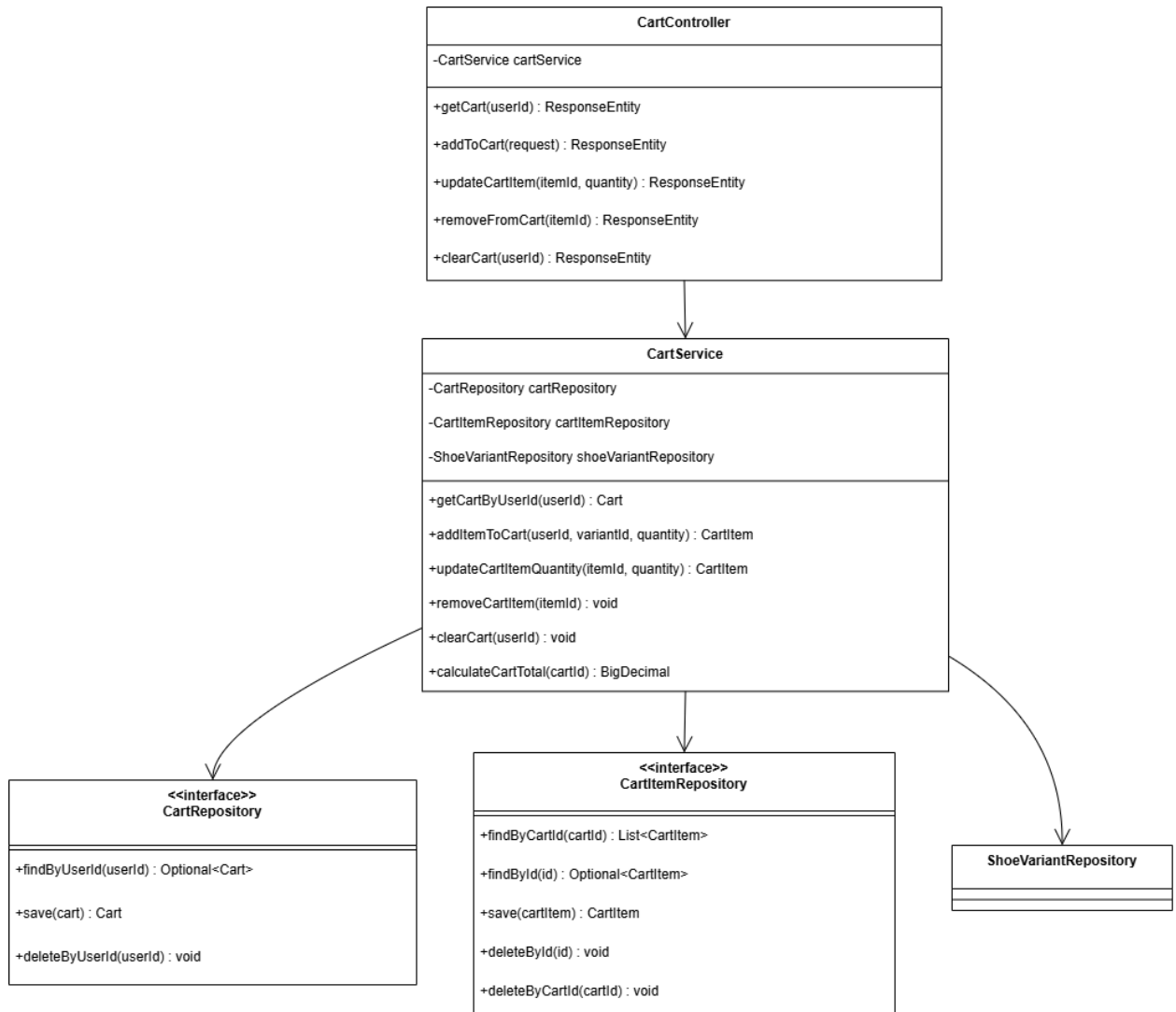


Lược đồ mô tả quy trình Khách hàng tìm kiếm sản phẩm bằng từ khóa. Khách hàng nhập từ khóa và nhấn nút tìm kiếm, Giao diện kiểm tra tính hợp lệ của từ khóa. Nếu không hợp lệ, lỗi được hiển thị ngay. Nếu hợp lệ, Giao diện gửi yêu cầu đến ProductController, controller chuyển tiếp cho ProductService xử lý, và service truy vấn ProductRepository. Repository thực hiện câu lệnh SQL xuống Database để lấy danh sách sản phẩm chứa từ khóa. Sau khi nhận kết quả, nếu danh sách rỗng, hệ thống thông báo không tìm thấy sản phẩm; nếu có dữ liệu, ProductController trả danh sách về Giao diện để hiển thị cho Khách hàng.

Lược đồ sequence của chức năng giỏ hàng

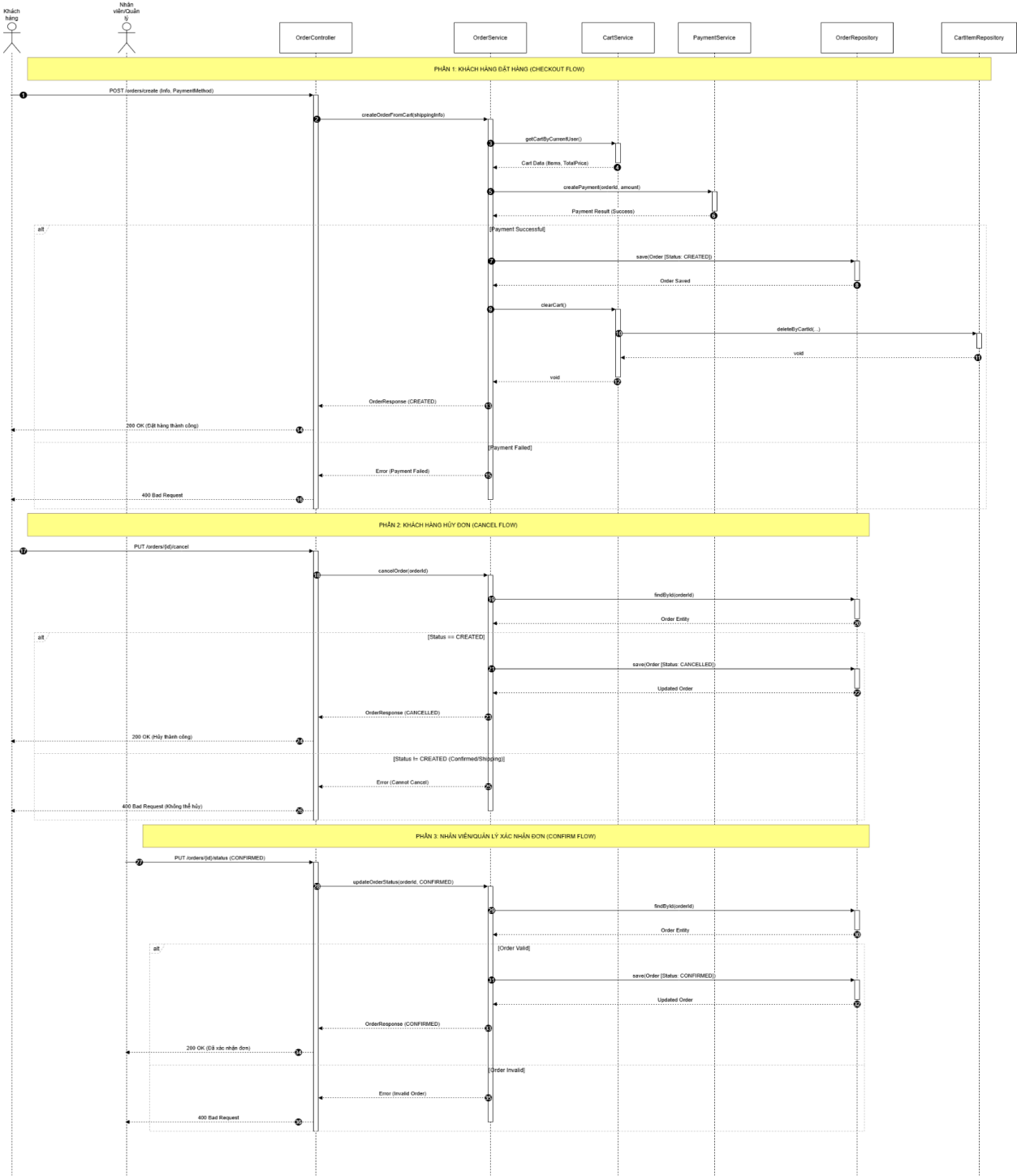


Class diagram của chức năng giỏ hàng

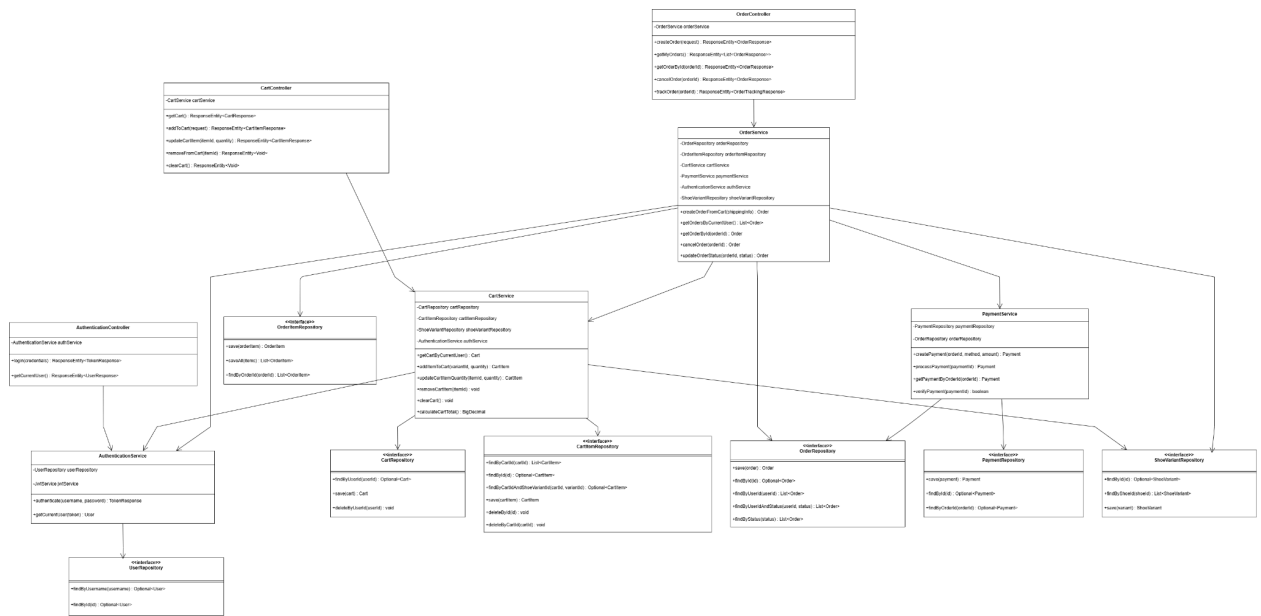


- Lược đồ mô tả quy trình Khách hàng thêm sản phẩm vào giỏ hàng. Khi nhấn “Thêm vào giỏ”, Giao diện gửi yêu cầu `addToCart` (gồm mã sản phẩm và số lượng) đến `CartController`, sau đó controller chuyển tiếp cho `CartService`. Service kiểm tra sản phẩm qua `ProductRepository`; nếu Database không trả về sản phẩm, hệ thống báo lỗi “Sản phẩm không tồn tại”. Nếu có, `CartService` lấy giỏ hàng hiện tại của người dùng từ `CartRepository`; nếu chưa có giỏ, hệ thống tạo giỏ mới. Tiếp theo, service duyệt danh sách sản phẩm trong giỏ: nếu sản phẩm đã tồn tại thì tăng số lượng, nếu chưa thì thêm mới. Sau khi cập nhật, `CartRepository` lưu dữ liệu xuống Database. Khi lưu thành công, `CartController` trả phản hồi để Giao diện hiển thị thông báo “Thêm thành công” cho Khách hàng.

Lược đồ sequence cho chức năng đặt hàng

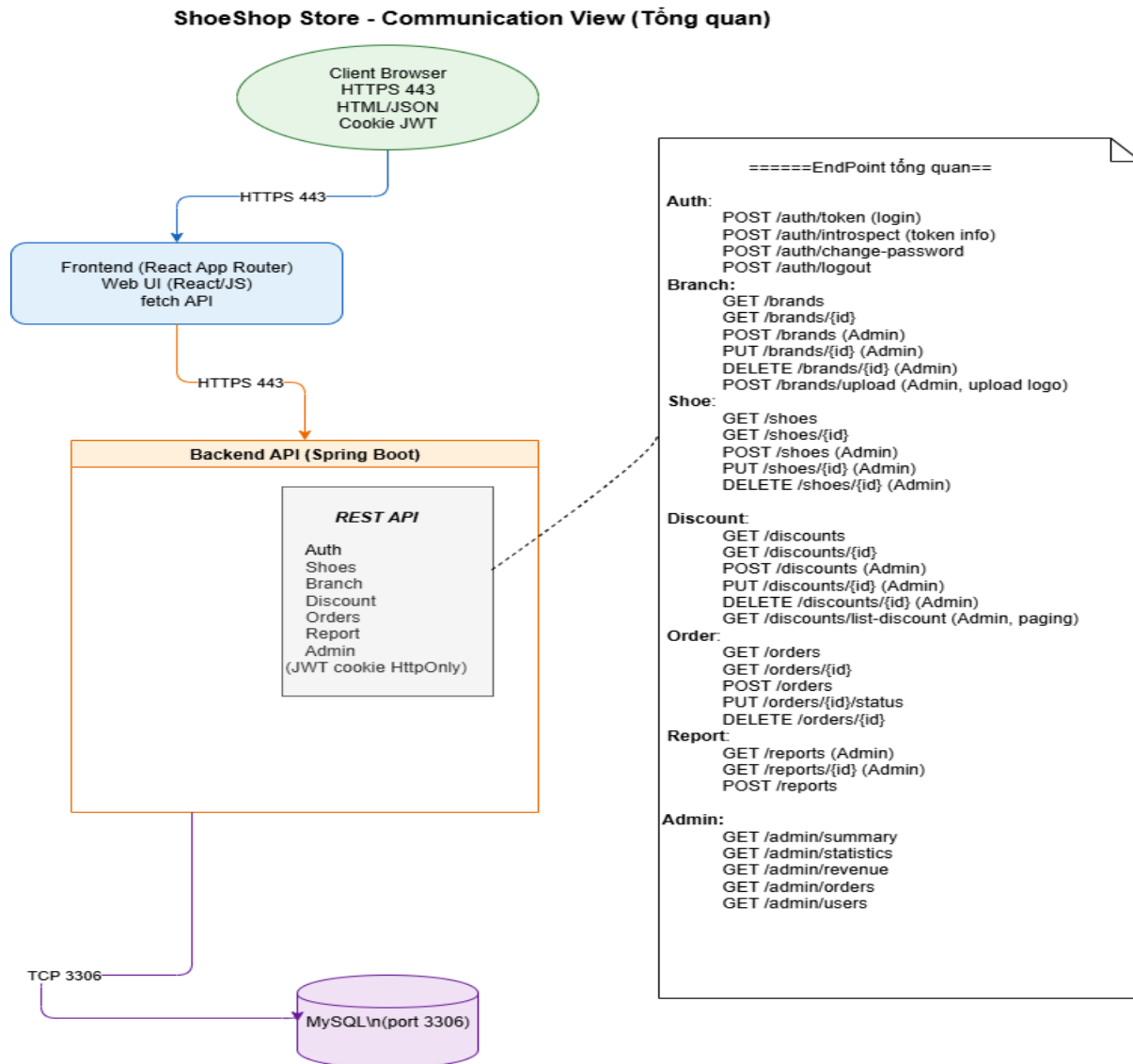


Class diagram cho chức năng đặt hàng



- Lược đồ mô tả ba quy trình chính: Khách hàng đặt hàng, Khách hàng hủy đơn và Nhân viên xác nhận đơn. Khi Khách hàng tạo đơn (POST), OrderController gọi OrderService, service lấy dữ liệu giỏ hàng từ CartService và gửi thanh toán qua PaymentService. Nếu thanh toán thành công, OrderService lưu đơn trạng thái CREATED, xóa giỏ hàng và trả về 200 OK; nếu thất bại, trả về 400 Bad Request. Ở quy trình hủy đơn, Khách hàng gửi yêu cầu hủy (PUT cancel). OrderService tìm đơn trong OrderRepository và kiểm tra trạng thái. Nếu đơn đang ở trạng thái CREATED, hệ thống cập nhật sang CANCELLED và trả thành công; nếu trạng thái khác, báo lỗi không thể hủy. Trong quy trình xác nhận đơn, Nhân viên gửi yêu cầu cập nhật trạng thái sang CONFIRMED. OrderService truy vấn đơn hàng và kiểm tra điều kiện hợp lệ (như tồn kho, thanh toán). Nếu hợp lệ, đơn được cập nhật sang CONFIRMED và lưu xuống database; nếu không hợp lệ, hệ thống trả về 400 Bad Request.

2.2.3. Phân tích giao tiếp



CHƯƠNG 3 – TEST PLAN

TỔNG QUAN CHƯƠNG 3 – KẾ HOẠCH KIỂM THỬ (TEST PLAN)

Chương 3 trình bày kế hoạch kiểm thử tổng thể cho hệ thống Shoeshop, nhằm xác định rõ phạm vi kiểm thử, đối tượng kiểm thử, cách tiếp cận và chiến lược kiểm thử sẽ áp dụng trong dự án.

Nội dung chương tập trung mô tả phương pháp luận ở mức tổng quan, bao gồm: phạm vi kiểm thử (in-scope/out-of-scope), phương pháp kiểm thử (tĩnh – động, hộp trắng – hộp đen – hộp xám), các mức kiểm thử (Unit Test, Integration Test, System Test, Acceptance Test), tiêu chí chấp nhận, môi trường kiểm thử, nguồn lực và lộ trình triển khai theo Agile.

3.1. Tổng quan kế hoạch kiểm thử

3.1.1. Test Scope (In-scope/Out-of-scope)

3.1.2. Brief Test Method (Agile, V-model, CI/CD)

3.2. Đối tượng kiểm thử (Test Object)

- Module khách hàng
- Module admin
- Module API
- Module Database

3.3. Chiến lược kiểm thử (Test Strategy)

3.3.1. Phương pháp kiểm thử (tĩnh/động, trắng/đen/xám)

3.3.2. Loại kiểm thử (Functional, UI, API, Performance, Security...)

3.3.3. Điều kiện chấp nhận (Acceptance Criteria)

3.3.4. Risk-Based Testing (Quản lý rủi ro kiểm thử)

3.4. Triển khai kiểm thử

3.4.1. Test Environment

3.4.2. Resource & Tools

3.4.3. Milestones theo Sprint

3.4.4. Deliverables (Test Plan, Test Case, Bug Report...)

3.5. Kế hoạch mở rộng

3.5.1. Kiểm thử đơn vị (Unit Testing – TDD)

3.5.2. Kiểm thử tích hợp (Integration Test)

3.5.3. Kiểm thử xác minh (Verification Testing)

3.5.4. Kiểm thử tự động (Automation Test Plan)

CHƯƠNG 4 – TEST DESIGN

Nội dung chương được xây dựng trên bốn góc nhìn (views) chính: phương pháp kiểm thử theo mô hình chữ V (V-Model), kiểm thử trong quy trình Agile, phân loại phương pháp kiểm thử (tĩnh

– động, hộp trắng – hộp đen – hộp xám), và lựa chọn giữa kiểm thử thủ công và kiểm thử tự động. Mỗi góc nhìn thể hiện một khía cạnh quan trọng của hoạt động kiểm thử và giúp đảm bảo độ bao phủ kiểm thử toàn diện trong suốt vòng đời phát triển sản phẩm.

Những phương pháp được trình bày trong chương là cơ sở để đội kiểm thử lựa chọn kỹ thuật phù hợp cho từng loại chức năng, từng tầng kiến trúc (UI, API, Database), và từng mức kiểm thử (Unit, Integration, System, Acceptance). Đồng thời, chương cũng xác định rõ cách áp dụng kiểm thử trong quy trình Agile và trong pipeline CI/CD, đảm bảo kiểm thử được tích hợp xuyên suốt từ lúc phát triển cho đến triển khai.

4.1. Tổng quan pipeline kiểm thử

- Luồng kiểm thử tổng thể từ yêu cầu → triển khai → báo cáo
- Vai trò QA trong CI/CD pipeline

4.3. View 2 – Kiểm thử theo Agile

4.3.1. Kiểm thử theo Sprint

4.3.2. Chu trình SDi – ADi – Test i

4.3.3. GitHub Flow + GitHub Action + Staging Test

4.4. View 3 – Phương pháp kiểm thử

4.4.1. Kiểm thử tĩnh

4.4.2. Kiểm thử động

4.4.3. Hộp trắng – Hộp đen – Hộp xám

4.4.4. Áp dụng từng phương pháp cho Shoeshop (UI, API, DB)

4.5. View 4 – Kiểm thử thủ công & tự động

4.5.1. Khi nào cần manual

4.5.2. Khi nào cần automation

4.5.3. Regression Test Suite

4.5.4. Công cụ automation đề xuất (Selenium / Playwright / Postman)

CHƯƠNG 5 – TEST OUTPUT & EVALUATION

5.1. Test Execution

5.1.1. Manual Execution

5.1.2. Automation Execution

5.1.3. Retest & Regression

5.2. Test Report

5.2.1. Kết quả Pass/Fail

5.2.2. Biểu đồ Test Coverage

5.3. Bug / Defect Report

5.3.1. Phân loại mức độ lỗi (Critical, Major, Minor)

5.3.2. Quy trình xử lý lỗi (Life Cycle)

5.4. Test Summary & QA Review

5.4.1. Đánh giá kết quả kiểm thử

5.4.2. Nhận xét tổng kết QA Manager

CHƯƠNG 1 – INTRODUCTION

1.1 Giới thiệu dự án ShoeShop E-commerce

Trình bày tổng quan về hệ thống thương mại điện tử ShoeShop: mục tiêu phát triển, phạm vi và đối tượng khách hàng. Giới thiệu ngắn gọn các công nghệ sử dụng (Spring Boot, React, MySQL) và kiến trúc đa tầng web.

Mình chứng chi tiết: *SRS_FunctionalReq.xlsx, SRS_NonFunctional.xlsx, Mô tả dự án.docx.*

1.2 Business Requirements

Tóm tắt yêu cầu chức năng (login, giỏ hàng, thanh toán, đơn hàng, giảm giá, tồn kho) và phi chức năng (hiệu năng, bảo mật, khả năng mở rộng).

Mình chứng chi tiết: *SRS_FunctionalReq.xlsx, SRS_NonFunctional.xlsx.*

1.3 Project Implementation Plan

Giới thiệu timeline triển khai, các giai đoạn theo mô hình V-Model và milestone từ Tuần 1 đến Tuần 6.

Mình chứng chi tiết: *TestPlan_Master.xlsx.*

CHƯƠNG 2 – ANALYSIS & DESIGN

2.1 Business Context

Trình bày bối cảnh nghiệp vụ, đối tượng tham gia (Admin, Customer, Staff). Giải thích quy trình từ đặt hàng đến giao hàng.

Mình chứng: *Business_Context.docx.*

2.2 Business Process

Trình bày sơ đồ BPMN/Activity Diagram về luồng đặt hàng – thanh toán – cập nhật kho.

Mình chứng: *Business_Process.png.*

2.3 Context Diagram

Mô tả sơ đồ tổng quan giữa hệ thống và các actor (Admin, Customer, Staff).

Mình chứng: *Context_Diagram.png.*

2.4 User Stories & Use Cases

Trình bày danh sách Use Case chính, minh họa 1–2 Use Case tiêu biểu (Login, Checkout).

Mình chứng: *UseCaseList.xlsx, UserStory.xlsx.*

2.5 Conceptual Model

Giới thiệu ERD để thể hiện mối quan hệ giữa sản phẩm, đơn hàng, tài khoản, giỏ hàng.

Mình chứng: *ERD_Summary.xlsx.*

2.6 Architecture Model (C4)

Trình bày tổng quan 4 cấp độ: System Context, Container, Component, Code. Minh họa bằng

hình mẫu.

Minh chứng: *Architecture_C4.pptx*.

2.7 Communication View

Mô tả Sequence Diagram FE – BE – DB khi login và khi checkout.

Minh chứng: *Communication_View.png*.

CHƯƠNG 3 – TEST PLAN

(tham khảo ABC test plan, fsoft)

3.1 Purpose & Background

Giới thiệu mục tiêu, phương pháp V-Model, đối tượng test và vai trò QA.

Minh chứng: *TestPlan_Master.xlsx*.

3.2 Scope of Testing

Xác định module trong/ngoài phạm vi test, tiêu chí pass/fail.

Minh chứng: *TestPlan_Master.xlsx*.

3.3 Test Requirements

Trình bày các tính năng sẽ được test (login, CRUD, checkout).

Minh chứng: *TestPlan_Master.xlsx*.

3.4 Test Strategy

Tóm tắt các loại test: Functional, UI, Data, Performance, Security, Regression.

Minh chứng: *TestPlan_Master.xlsx*.

3.5 Resource & Environment

Giới thiệu môi trường test (OS, DB, Tools, Browser) và vai trò QA.

Minh chứng: *TestPlan_Master.xlsx*.

CHƯƠNG 4 – TEST DESIGN

4.1 V-Model Overview

Giới thiệu mô hình chữ V và liên kết giữa phát triển – kiểm thử.

Minh chứng: *VModel_Diagram.png*.

4.2 Generate Manual Test Cases (đọc template file gen testcase from UC)

Trình bày quy trình sinh test case từ Use Case, minh họa một ví dụ.

Minh chứng: *Test_Case.xlsx*, *Test_Data.xlsx*.

4.3 Automation Testing

Giới thiệu Selenium, cách tạo kịch bản login, add product, checkout.

Mình chứng: *Automation_Scripts/*.

4.4 Test Techniques (**DYNAMIC và STATIC test , đọc slide**)

Giới thiệu các kỹ thuật: EP, BVA, Decision Table, State Transition.

Mình chứng: *Test_Design.xlsx*.

4.5 Integrate GenAI (**đọc tham khảo sánh genAI . trình bày cách mà nhóm đã áp dụng AI để sinh testcase**)

Trình bày ứng dụng ChatGPT/Copilot trong sinh testcase, dữ liệu và gợi ý bug.

Mình chứng: *GenAI_TestAssist.md*.

CHƯƠNG 5 – TEST OUTPUT

5.1 Test Execution

Mô tả quy trình chạy test, gắn trạng thái Pass/Fail, và ảnh minh họa.

Mình chứng: *Test_Report.xlsx*.

5.2 Test Report & Summary

Tổng hợp tỷ lệ Pass/Fail, biểu đồ bug, coverage.

Mình chứng: *Test_Report.xlsx*, *Test_Report_Chart.png*.

5.3 Bug Report

Tóm tắt danh sách bug (ID, module, severity, steps).

Mình chứng: *Bug_Report.xlsx*.

5.4 QA Review & Conclusion

📋 Tóm tắt đánh giá chất lượng hệ thống.

📋 Nhận định đạt tiêu chuẩn nghiệm thu.

Mình chứng: *QA_Checklist.xlsx*.

PHỤ LỤC: DANH SÁCH FILE MINH CHỨNG

- UseCaseList.xlsx
- UserStory.xlsx
- ERD_Summary.xlsx
- TestPlan_Master.xlsx
- Test_Case.xlsx
- Test_Data.xlsx
- Bug_Report.xlsx
- Test_Report.xlsx

- QA_Checklist.xlsx
- Automation_Scripts/

TỔNG CÁC SHEET cần phải có:

1. Usecase

2. Screendesign

3. Design Workflow Test (luồng kiểm thử tổng thể)

4. Testdesign

SRS / Use Case



Design Workflow Test (luồng kiểm thử tổng thể)



Test Design .xlsx (điều kiện kiểm thử chi tiết)



Test Case .xlsx + Test Data .xlsx

5. Testcase kèm theo testdata sheet

6. testdata sheet

7. Checklist review testcase

8 QA

- 4. Test Design Slide

- Test Design Workflow - Thêm 1 sheet điền data như thế nào ở từng workflow

- Test Design Template (File riêng - kiểm soát phủ đủ chức năng)

- Test Design Techniques

+ Static test (Slide Software Testing Basic): Nhìn code, không chạy, phán đoán code đúng hay sai

+ Dynamic Test (Slide Software Testing Basic): Hộp trắng: biết rõ cấu trúc bên trong Hộp đen: Dựa trên đặc tả

- Test Plan --> Test Design (Ucs + GUI + Data) --> Test Case: Đánh số - Phải có test data đi kèm - Quản lý bang Excel --- 3 sheet này phải liên quan

-- đọc slide testcase:

Test case description

Test Pre-Condition

Test case procedure

Ví dụ minh họa: Luồng từ Test Design → Test Case

Requirement (FR-01):

Người dùng có thể đăng nhập hệ thống bằng tài khoản hợp lệ.

Test Design:

| Test Design ID | Test Condition | Technique | Expected Behavior |
|----------------|------------------------------------|--------------------------|--------------------------------|
| TD-LOGIN-01 | Nhập đúng tài khoản, đúng mật khẩu | Equivalence Partitioning | Cho phép đăng nhập |
| TD-LOGIN-02 | Nhập đúng tài khoản, sai mật khẩu | Equivalence Partitioning | Thông báo lỗi |
| TD-LOGIN-03 | Bỏ trống tài khoản hoặc mật khẩu | Boundary Value | Cảnh báo “Không được để trống” |

Test Case tương ứng:

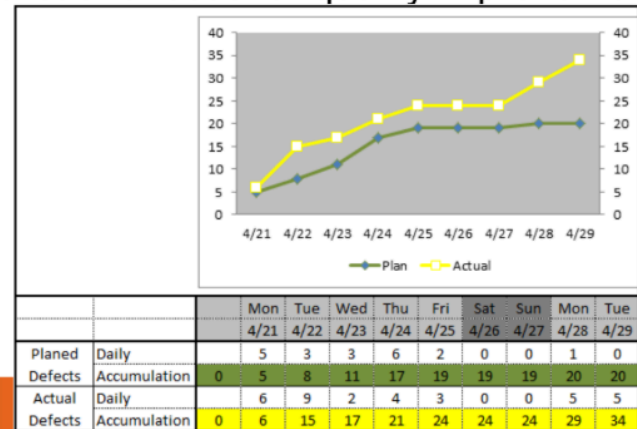
| Test Case ID | Input Data | Expected Result | Status |
|--------------|-------------------------|--------------------------|--------|
| TC-LOGIN-01 | user=admin, pass=123456 | Thành công | Pass |
| TC-LOGIN-02 | user=admin, pass=abc | Thông báo “Sai mật khẩu” | Pass |
| TC-LOGIN-03 | user= , pass= | Cảnh báo lỗi | Pass |

Test report process

Defect Trend

Defect Trend:

- Give an overall picture of product quality
- Predict defects in the near future (next days)
- Identify high-risk areas to focus for quality improvement



11

Test Result Report Template

TEST REPORT

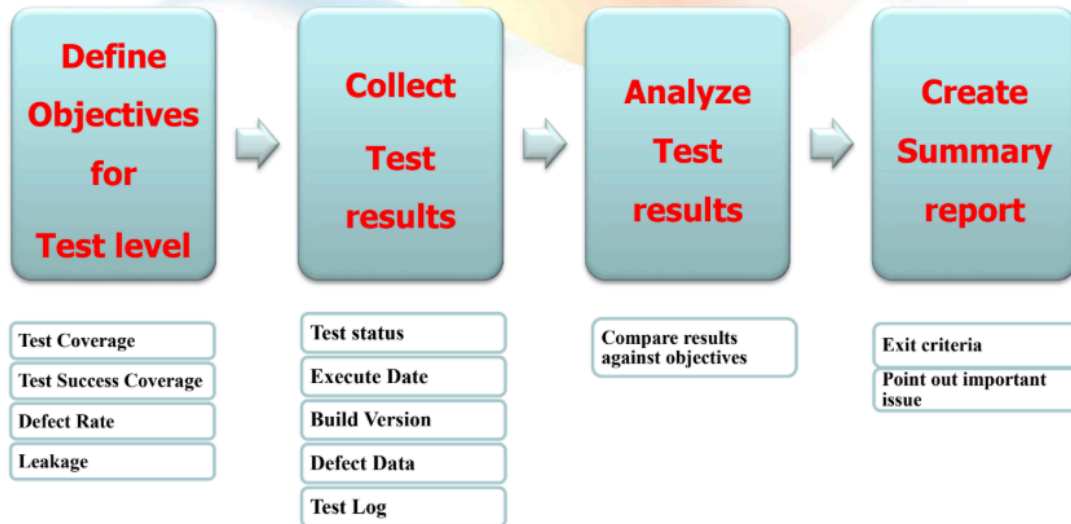
| | | | |
|---------------------|----------------|-----------------|------------|
| Project name | <Project name> | Reviewer | <Reviewer> |
| Creator | <Creator> | Approver | <Approver> |
| Note | | | |

Test Coverage: **46%**
Successful Test Coverage: **33%**

Date: <yyyy/mm/dd>

| No | Items | Tested | Passed | Failed | Blocked | Skipped | Not Yet Tested | Total | Tested Coverage |
|--------------|------------|-----------|-----------|----------|----------|-----------|----------------|------------|-----------------|
| 1 | Function 1 | 23 | 15 | 5 | 3 | 7 | 18 | 48 | 48% |
| 2 | Function 2 | 26 | 20 | 4 | 2 | 10 | 22 | 58 | 45% |
| 3 | | | | | | | | | |
| 4 | | | | | | | | | |
| 5 | | | | | | | | | |
| Total | | 49 | 35 | 9 | 5 | 17 | 40 | 106 | |

Test Report Process



Defect by Test Case

Purpose: List defect found by Test Case

| Test case name/ID | Fatal | Serious | Medium | Cosmetic | Total Defect | Total (W.def) |
|-------------------|-------|----------|----------|-----------|--------------|---------------|
| 5910-0001 | | | | 1 | 1 | 1 |
| 5945-0001 | | | 1 | | 1 | 3 |
| 5897-0001 | | | | 1 | 1 | 1 |
| 5951 | | | | 1 | 1 | 1 |
| 6011 | | | | 1 | 1 | 1 |
| 5964 | | | | 1 | 1 | 1 |
| 5943-0001 | | | | 1 | 1 | 1 |
| 5902-0001 | | | | 1 | 1 | 1 |
| 5920-0001 | | | | 1 | 1 | 1 |
| 5952 | | | | 1 | 1 | 1 |
| 5963 | | | | 1 | 1 | 1 |
| 1474240 | | | | 1 | 1 | 1 |
| 5946 | | | | 1 | 1 | 1 |
| 5965 | | | | 1 | 1 | 5 |
| 5898-0001 | | 1 | | 1 | 1 | 1 |
| 5944-0001 | | | | 1 | 1 | 1 |
| 5905-0001 | | | | 1 | 1 | 1 |
| 5917-0001 | | | | 2 | 2 | 2 |
| 5962 | | 1 | | 2 | 3 | 7 |
| 5895-0001 | | | | 1 | 1 | 1 |
| 5918-0001 | | | | 1 | 1 | 1 |
| 5878-0001 | | | | 1 | 1 | 1 |
| 5888-0001 | | | | 1 | 1 | 1 |
| 5948 | | | | 1 | 1 | 1 |
| 6019 | | | 1 | | 1 | 3 |
| 5907-0001 | | | | 1 | 1 | 1 |
| Total | | 2 | 2 | 25 | 29 | 41 |

Test Summary Report

Test Summary Report

Test Summary Report: a document summarizing testing activities and results. It also contains an evaluation of the corresponding test items against exit criteria (E.g. final metrics, status of passed or failed, resource use, efficiency...)

Test Summary Structure:

- Summary
- Test Case result report
- Defect Report
- Open point

Kiểm thử tự động

Selenium

- **Manual Test**
- **Auto Test**
- **Test script (từ test case)**

■ Đồ án

- **Chạy test case – viết test report - auto test (selenium)**
- **Chương 4: Thiết kế kiểm thử (phương pháp luận)**
- **+ cách sinh ra testcase**
- **+ bổ sung automation test để kiểm tra các tác vụ tự động**
- **SL test case**
- **Chương 5: kết quả kiểm thử**
Chạy test -> Testreport ->bug defect report

Note tuần 12:

test pre condition: trong template TESTCASE phải ghi rõ các bước chi tiết: chứ không chung chung

- tham khảo slide

- test case procedure (slide) mô tả chi tiết

- test data nên tạo một sheet khác để tham chiếu

- testcase review checklist: giải thích tại sao

slide(testing web application)

funcionality testing:

- kiểm tra page, chuyển page, các link có bị broken ko
- test suit là gì? (theo pp kiểm thử, nghiệp vụ,...)
- data testing: kiểm tra tính toàn vẹn dữ liệu (insert) ràng buộc, kiểm tra mức data sql (trigger)

crowdtesting: cho người khác test

Apache jmeter: kiểm thử lượng truy cập performance