Trường Đại học Phenikaa

A logo for a university

AI-generated content may be incorrect.

Bài tập lớn môn học Đánh giá và kiểm định chất lượng phần mềm

Đề tài: “Xây dựng Hệ thống E-commerce Bán Quần Áo Phân Tán”

Tài liệu Test Plan

Nhóm:

Họ tên thành viên

Đinh Thị Ngọc Anh - 22010069

Nguyễn Thanh Phong - 22010251

Hà Nội, 12/06/2025

Lịch sử các lần cập nhật

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Prepared by** | **Reviewed by** | **Approved By** |
| 15/03/2023 | 1.0 | Kế hoạch cơ bản |  |  |  |
| Dd/mm/yyyy | 2.0 | Final Test Plan |  |  |  |

Mục Lục

[1 Giới thiệu 5](#_Toc200210290)

[1.1 Mục đích 5](#_Toc200210291)

[1.2 Tổng quan 5](#_Toc200210292)

[1.3 Phạm vi 5](#_Toc200210293)

[1.4 Các định nghĩa và các từ viết tắt được dùng 6](#_Toc200210294)

[1.4.1 Các định nghĩa 6](#_Toc200210295)

[1.4.2 Các từ viết tắt 7](#_Toc200210296)

[1.5 Những người sử dụng tài liệu này 7](#_Toc200210297)

[2 Tài liệu tham khảo 7](#_Toc200210298)

[3 Lịch trình công việc 9](#_Toc200210299)

[4 Những yêu cầu về tài nguyên 9](#_Toc200210300)

[4.1 Phần cứng 9](#_Toc200210301)

[4.2 Phần mềm 9](#_Toc200210302)

[4.3 Công cụ kiểm thử 10](#_Toc200210303)

[4.4 Môi trường kiểm thử 10](#_Toc200210304)

[4.5 Nhân sự 11](#_Toc200210305)

[4.5.1 Vai trò và trách nhiệm 11](#_Toc200210306)

[4.5.2 Đào tạo 11](#_Toc200210307)

[5 Phạm vi kiểm thử 12](#_Toc200210308)

[5.1 Những chức năng được kiểm thử 12](#_Toc200210309)

[5.1.1 Xác thực người dùng 12](#_Toc200210310)

[5.1.2 Quản lý Sản phẩm (Phía User) 12](#_Toc200210311)

[5.1.3 Quản lý giỏ hàng 12](#_Toc200210312)

[5.1.4 Quy trình Đặt hàng và Thanh toán 12](#_Toc200210313)

[5.1.5 Quản lý Đơn hàng Cá nhân 12](#_Toc200210314)

[5.1.6 Trang Thông tin Cá nhân 13](#_Toc200210315)

[5.1.7 Xác thực Admin 13](#_Toc200210316)

[5.1.8 Dashboard Admin 13](#_Toc200210317)

[5.1.9 Quản lý Người dùng (Admin) 13](#_Toc200210318)

[5.1.10 Quản lý Danh mục (Admin) 13](#_Toc200210319)

[5.1.11 Quản lý Sản phẩm (Admin): 13](#_Toc200210320)

[5.1.12 Quản lý đơn hàng (Admin): 13](#_Toc200210321)

[5.2 Những chức năng không được kiểm thử 14](#_Toc200210322)

[6 Chiến lược kiểm thử 14](#_Toc200210323)

[6.1 Test Stages 14](#_Toc200210324)

[6.2 Các loại kiểm thử 15](#_Toc200210325)

[7 Điều kiện chấp nhận 15](#_Toc200210326)

[8 Defect Tracking (lỗi) 17](#_Toc200210327)

[8.1 Phân loại lỗi 17](#_Toc200210328)

[8.2 Quy trình xử lý lỗi 17](#_Toc200210329)

[9 Test Deliverables 17](#_Toc200210330)

[9.1 Test Cases 17](#_Toc200210331)

[9.2 Test Reports 17](#_Toc200210332)

# Giới thiệu

## Mục đích

Tài liệu kế hoạch kiểm thử này được lập ra với các mục đích sau:

* Xác định thông tin cơ bản về dự án "Hệ thống E-commerce Bán Quần Áo Phân Tán", các thành phần chức năng chính sẽ được kiểm thử và những phần nằm ngoài phạm vi kiểm thử trong khuôn khổ đồ án này.
* Liệt kê các yêu cầu cần thiết cho việc thực hiện kiểm thử, bao gồm tài nguyên phần cứng, phần mềm và công cụ.
* Đề xuất chiến lược kiểm thử phù hợp, tập trung vào kiểm thử chức năng End-to-End (E2E) và một số khía cạnh phi chức năng cơ bản.
* Phân công vai trò và trách nhiệm cho các thành viên tham gia kiểm thử (nếu có).
* Xác định các sản phẩm đầu ra (deliverables) của quá trình kiểm thử.

## Tổng quan

Dự án "Hệ thống E-commerce Bán Quần Áo Phân Tán" nhằm mục đích xây dựng một ứng dụng web cho phép người dùng thực hiện các thao tác mua sắm trực tuyến như duyệt sản phẩm, quản lý giỏ hàng, đặt hàng và thanh toán (mô phỏng VNPay). Hệ thống cũng bao gồm một giao diện quản trị để quản lý các khía cạnh của cửa hàng. Kiến trúc của hệ thống dựa trên microservices, được container hóa bằng Docker và quản lý bởi Docker Compose.

Việc kiểm thử sẽ tập trung vào việc xác minh các luồng chức năng chính của cả người dùng và quản trị viên, cũng như đánh giá một số yếu tố phi chức năng cơ bản như hiệu năng API, bảo mật xác thực/phân quyền và khả năng chịu lỗi ở mức độ cơ bản.

## Phạm vi

Tài liệu kế hoạch kiểm thử này áp dụng cho việc kiểm thử các chức năng và một số khía cạnh phi chức năng của dự án "Hệ thống E-commerce Bán Quần Áo Phân Tán" được phát triển trong khuôn khổ đồ án.

Các dịch vụ và chức năng chính cần kiểm thử bao gồm:

* **Giao diện người dùng (Frontend User):**
  + Luồng đăng ký, đăng nhập, đăng xuất.
  + Xem danh sách sản phẩm, chi tiết sản phẩm.
  + Quản lý giỏ hàng (thêm, sửa, xóa).
  + Quy trình Checkout và tạo đơn hàng.
  + Luồng thanh toán VNPay (tạo URL, xử lý trang return).
  + Xem lịch sử đơn hàng, chi tiết đơn hàng.
* **Giao diện quản trị (Frontend Admin Panel):**
  + Đăng nhập Admin.
  + Quản lý người dùng (xem danh sách).
  + Quản lý sản phẩm (xem, thêm, sửa, xóa, cập nhật tồn kho).
  + Quản lý danh mục (xem, thêm, sửa, xóa).
  + Quản lý đơn hàng (xem danh sách, xem chi tiết, cập nhật trạng thái).
  + Dashboard Admin (hiển thị dữ liệu thống kê).
* **Kiểm thử API Backend (Thông qua API Gateway):** Xác minh các endpoint API chính liên quan đến các chức năng trên.
* **Kiểm thử Phi chức năng (cơ bản):**
  + Thời gian phản hồi API cơ bản.
  + Kiểm tra xác thực (JWT) và phân quyền.
  + Demo Health Checks và Failover cơ bản.

## Các định nghĩa và các từ viết tắt được dùng

### Các định nghĩa

|  |  |
| --- | --- |
| **Thuật ngữ** | **Định nghĩa** |
| Hệ thống E-commerce Phân Tán | Ứng dụng web bán quần áo trực tuyến được xây dựng theo kiến trúc microservices. |
| Microservice | Một thành phần dịch vụ nhỏ, độc lập, chịu trách nhiệm cho một nghiệp vụ cụ thể (ví dụ: User Service, Product Service). |
| API Gateway | Điểm vào duy nhất cho tất cả các request từ client, thực hiện định tuyến đến các microservices backend. |
| JWT (JSON Web Token) | Một chuẩn mở (RFC 7519) định nghĩa một cách nhỏ gọn và khép kín để truyền thông tin giữa các bên dưới dạng một đối tượng JSON. |
| VNPay Sandbox | Môi trường thử nghiệm của cổng thanh toán VNPay, cho phép mô phỏng các giao dịch thanh toán. |
| Docker | Nền tảng container hóa ứng dụng. |
| Docker Compose | Công cụ để định nghĩa và chạy các ứng dụng Docker đa container. |
| RabbitMQ | Message broker mã nguồn mở, sử dụng cho giao tiếp bất đồng bộ. |
| Redis | Kho lưu trữ dữ liệu key-value trong bộ nhớ, sử dụng cho caching và quản lý giỏ hàng |
| PostgreSQL | Hệ quản trị cơ sở dữ liệu quan hệ đối tượng mã nguồn mở |
| Loki Stack (Loki, Promtail, Grafana) | Bộ công cụ dùng để thu thập, lưu trữ và trực quan hóa log tập trung. |
| E2E Test (End-to-End Test) | Loại kiểm thử phần mềm xác minh toàn bộ luồng ứng dụng từ đầu đến cuối, mô phỏng hành vi người dùng thực tế. |

### Các từ viết tắt

|  |  |
| --- | --- |
| **Từ viết tắt** | **Mô tả** |
| SRS | Software Requirement Specification (Đặc tả Yêu cầu Phần mềm) |
| QA | Quality Assurance (Đảm bảo Chất lượng) |
| UI | User Interface (Giao diện Người dùng) |
| API | Application Programming Interface (Giao diện Lập trình Ứng dụng |
| DB | Database (Cơ sở dữ liệu) |
| HTTP | Hypertext Transfer Protocol |
| HTTPS | Hypertext Transfer Protocol Secure |
| CRUD | Create, Read, Update, Delete |
| SPA | Single Page Application |

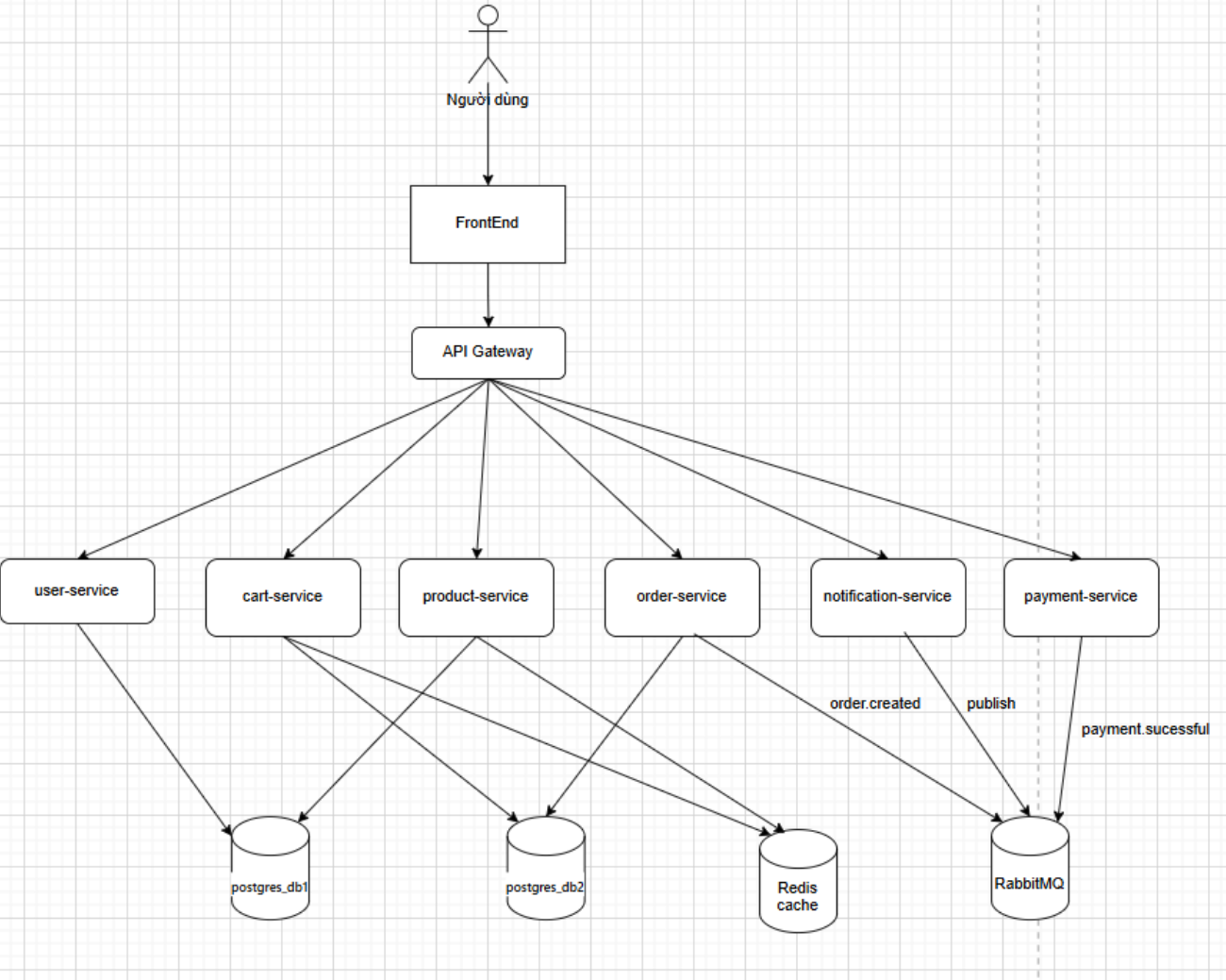
## Những người sử dụng tài liệu này

Tài liệu này được sử dụng chủ yếu bởi:

* **Người Kiểm thử (Tester)/Nhóm Kiểm thử:** Để hiểu phạm vi, chiến lược và các yêu cầu cần thiết cho việc kiểm thử.
* **Người Quản lý Dự án/Test Manager (nếu có):** Để theo dõi tiến độ và đảm bảo chất lượng kiểm thử.
* **Đội ngũ Phát triển (Developer):** Để hiểu các kịch bản kiểm thử và phối hợp sửa lỗi.
* **Giảng viên Hướng dẫn:** Để đánh giá kế hoạch và quy trình kiểm thử của dự án.

# Tài liệu tham khảo

* **Tài liệu Đặc tả Yêu cầu Phần mềm (SRS):** *SRS\_ECommerce\_PhanTan\_v1.0.docx* (Tài liệu này sẽ được xây dựng dựa trên các chức năng đã thống nhất).
* **Tài liệu Thiết kế Hệ thống:**

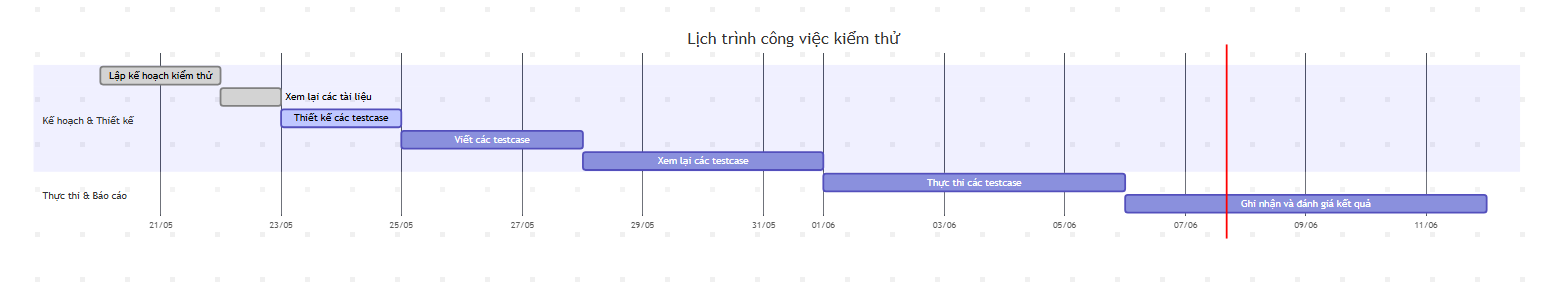


* **Mã nguồn dự án:** Toàn bộ mã nguồn của các microservices backend, frontend và API Gateway.
* **Tài liệu hướng dẫn của các công nghệ sử dụng:**
  + NestJS Documentation ([https://docs.nestjs.com/](https://www.google.com/url?sa=E&q=https%3A%2F%2Fdocs.nestjs.com%2F))
  + React Documentation ([https://react.dev/](https://www.google.com/url?sa=E&q=https%3A%2F%2Freact.dev%2F))
  + Vite Documentation ([https://vitejs.dev/](https://www.google.com/url?sa=E&q=https%3A%2F%2Fvitejs.dev%2F))
  + Axios Documentation ([https://axios-http.com/docs/intro](https://www.google.com/url?sa=E&q=https%3A%2F%2Faxios-http.com%2Fdocs%2Fintro))
  + React Router DOM Documentation ([https://reactrouter.com/](https://www.google.com/url?sa=E&q=https%3A%2F%2Freactrouter.com%2F))
  + TypeORM Documentation ([https://typeorm.io/](https://www.google.com/url?sa=E&q=https%3A%2F%2Ftypeorm.io%2F))
  + PostgreSQL Documentation ([https://www.postgresql.org/docs/](https://www.google.com/url?sa=E&q=https%3A%2F%2Fwww.postgresql.org%2Fdocs%2F))
  + Redis Documentation ([https://redis.io/docs/](https://www.google.com/url?sa=E&q=https%3A%2F%2Fredis.io%2Fdocs%2F))
  + RabbitMQ Documentation ([https://www.rabbitmq.com/documentation.html](https://www.google.com/url?sa=E&q=https%3A%2F%2Fwww.rabbitmq.com%2Fdocumentation.html))
  + Nginx Documentation ([https://nginx.org/en/docs/](https://www.google.com/url?sa=E&q=https%3A%2F%2Fnginx.org%2Fen%2Fdocs%2F))
  + Docker Documentation ([https://docs.docker.com/](https://www.google.com/url?sa=E&q=https%3A%2F%2Fdocs.docker.com%2F))
  + Docker Compose Documentation ([https://docs.docker.com/compose/](https://www.google.com/url?sa=E&q=https%3A%2F%2Fdocs.docker.com%2Fcompose%2F))
  + Loki & Promtail & Grafana Documentation.
  + VNPay Integration Documentation (Sandbox).
  + Jest Documentation ([https://jestjs.io/docs/getting-started](https://www.google.com/url?sa=E&q=https%3A%2F%2Fjestjs.io%2Fdocs%2Fgetting-started))
  + Selenium WebDriver JavaScript Documentation ([https://www.selenium.dev/documentation/webdriver/getting\_started/first\_script/](https://www.google.com/url?sa=E&q=https%3A%2F%2Fwww.selenium.dev%2Fdocumentation%2Fwebdriver%2Fgetting_started%2Ffirst_script%2F))
* **Slide bài giảng môn học:** Các slide liên quan đến quy trình kiểm thử, các loại kiểm thử, kỹ thuật thiết kế test case.

# Lịch trình công việc

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Milestone** | **Deliverables** | **Duration** | **Start Date** | **End Date** |
| Lập kế hoạch kiểm thử | Tài liệu Test Plan | 2 ngày | 20/05/2025 | 22/05/2025 |
| Xem lại các tài liệu | Tài liệu Test Plan | X1 ngày | 22/05/2025 | 23/05/2025 |
| Thiết kế các testcase | Tài liệu Testcase | X2 ngày | 23/05/2025 | 25/05/2025 |
| Viết các testcase | Tài liệu Testcase | X3 ngày | 25/05/2025 | 28/05/2025 |
| Xem lại các testcase | Tài liệu Testcase | X4 ngày | 28/05/2025 | 01/06/2025 |
| Thực thi các testcase | Tài liệu Testcase | X5 ngày | 01/06/2025 | 06/06/2025 |
| Ghi nhận và đánh giá kết quả kiểm thử | Tài liệu Testcase | X6 ngày | 06/06/2025 | 12/06/2025 |

Biểu đồ Gantt



Biểu đồ Grantt

# Những yêu cầu về tài nguyên

## Phần cứng

Máy tính cá nhân có kết nối mạng Internet (nếu có).

|  |  |  |  |
| --- | --- | --- | --- |
| **CPU** | **RAM** | **HDD** | **Architecture** |
| Intel Core i7, 2.3 GHz | 8 GB | 1TB | 64 bit |

## Phần mềm

|  |  |  |
| --- | --- | --- |
| **Tên phần mềm** | **Phiên bản** | **Loại** |
| Chrome | Phiên bản ổn định mới nhất | Trình duyệt web |
| Docker Desktop | Phiên bản ổn định mới nhất | Công cụ Container hóa |
| Postman (hoặc Insomnia) | Phiên bản ổn định mới nhất | Công cụ Kiểm thử API |
| VS Code (hoặc IDE/Text Editor khác) | Phiên bản ổn định mới nhất | Trình soạn thảo Mã nguồn |
| Node.js | LTS (ví dụ: 18.x hoặc 20.x) | Môi trường Runtime JavaScript |

## Công cụ kiểm thử

|  |  |  |  |
| --- | --- | --- | --- |
| **Hoạt động** | **Công cụ** | **Nhà cung cấp** | **Phiên bản** |
| Quản lý Test Case | Microsoft Office Excel | Microsoft | 2019 |
| Quản lý Configuration | Microsoft Office Word | Microsoft | 2019 |
| Defect Tracking | Microsoft Office Excel | Microsoft | 2019 |
| Quản lý tiến độ kiểm thử | Microsoft Project Professional | Microsoft | 2019 |

## Môi trường kiểm thử

* Môi trường chính: Máy tính cá nhân của người kiểm thử đã cài đặt Docker và các phần mềm cần thiết như mô tả ở mục 4.1 và 4.2.
* **Hệ thống được kiểm thử:** Toàn bộ hệ thống E-commerce (Frontend, API Gateway, các Microservices Backend, Databases, Redis, RabbitMQ, Loki Stack) sẽ được triển khai và chạy cục bộ thông qua Docker Compose từ mã nguồn mới nhất của dự án.
* **Mạng:** Sử dụng mạng nội bộ (localhost) do Docker tạo ra để các container giao tiếp với nhau. API Gateway sẽ được expose ra cổng 80 của máy host. Frontend dev server sẽ chạy trên cổng riêng (ví dụ: 5173).
* **Dữ liệu:**
  + Chuẩn bị một bộ dữ liệu mẫu (Test Data) bao gồm các tài khoản người dùng (user, admin), danh mục, sản phẩm với các trạng thái khác nhau để phục vụ cho các kịch bản kiểm thử.

## Nhân sự

### Vai trò và trách nhiệm

|  |  |
| --- | --- |
| **Thành viên** | **Vai trò** |
| Đinh Thị Ngọc Anh | Test Manager / Test Designer / Tester:  Chịu trách nhiệm chính cho việc lập Kế hoạch Kiểm thử (Test Plan).  - Thiết kế các Test Cases chính cho các luồng quan trọng (User & Admin).  - Thực thi kiểm thử chức năng (manual & automated), API testing.  - Thực hiện kiểm thử phi chức năng cơ bản (Performance, Security, Fault Tolerance).  - Ghi nhận, phân loại và theo dõi lỗi (Defect Tracking). |
| Nguyễn Thanh Phong | Test Designer / Tester:  - Thiết kế và viết các kịch bản E2E Tests tự động (Selenium/Jest).  - Tổng hợp và viết Báo cáo Kết quả Kiểm thử.  - Đảm bảo chất lượng tổng thể của sản phẩm trước khi demo/báo cáo. |

### Đào tạo

* Tất cả thành viên tham gia kiểm thử (nếu có nhiều người) cần:
  + Nắm vững Đặc tả Yêu cầu Hệ thống (SRS) và tài liệu thiết kế.
  + Hiểu rõ cách hoạt động của hệ thống E-commerce, luồng dữ liệu giữa các microservices.
  + Được hướng dẫn sử dụng các công cụ kiểm thử (Postman, Selenium/Jest, cách đọc log từ Grafana/Docker).
  + Tham gia các buổi họp nhóm để thảo luận về các kịch bản kiểm thử, các vấn đề phát sinh và thống nhất quy trình báo cáo lỗi.
* Tự nghiên cứu thêm về các kỹ thuật kiểm thử E2E, kiểm thử API, và các khái niệm kiểm thử phi chức năng cơ bản nếu cần.

# Phạm vi kiểm thử

## Những chức năng được kiểm thử

### Xác thực người dùng

* Kiểm tra chức năng Đăng ký tài khoản mới với các trường hợp dữ liệu hợp lệ và không hợp lệ.
* Kiểm tra chức năng Đăng nhập với tài khoản hợp lệ, tài khoản không tồn tại, sai mật khẩu.
* Kiểm tra chức năng Đăng xuất.
* Kiểm tra việc duy trì trạng thái đăng nhập qua localStorage (khi refresh trang).
* Kiểm tra việc tự động chuyển hướng đến trang login khi truy cập route private mà chưa đăng nhập.

### Quản lý Sản phẩm (Phía User)

* Kiểm tra chức năng Xem danh sách sản phẩm (hiển thị, phân trang nếu có).
* Kiểm tra chức năng Xem chi tiết một sản phẩm.
* Kiểm tra chức năng Tìm kiếm sản phẩm theo tên.
* Kiểm tra chức năng Lọc sản phẩm theo danh mục.

### Quản lý giỏ hàng

* Kiểm tra chức năng Thêm sản phẩm vào giỏ hàng từ trang danh sách và trang chi tiết.
* Kiểm tra chức năng Xem nội dung giỏ hàng (hiển thị đúng sản phẩm, số lượng, giá, tổng tiền tạm tính).
* Kiểm tra chức năng Cập nhật số lượng sản phẩm trong giỏ hàng.
* Kiểm tra chức năng Xóa một sản phẩm khỏi giỏ hàng.

### Quy trình Đặt hàng và Thanh toán

* Kiểm tra chức năng Tiến hành đặt hàng từ giỏ hàng (chuyển đến trang Checkout).
* Kiểm tra việc nhập/xác nhận thông tin giao hàng trên trang Checkout.
* Kiểm tra chức năng "Đặt hàng và Thanh toán VNPay", đảm bảo hệ thống gọi API tạo đơn hàng và sau đó gọi API tạo URL thanh toán.
* Kiểm tra việc **chuyển hướng thành công** sang trang sandbox của VNPay.
* Kiểm tra việc xử lý trang **Return URL** từ VNPay sau khi người dùng (mô phỏng) hoàn tất/hủy thanh toán trên sandbox VNPay (hiển thị thông báo thành công/thất bại tạm thời).

### Quản lý Đơn hàng Cá nhân

* Kiểm tra chức năng Xem lịch sử các đơn hàng đã đặt của người dùng.
* Kiểm tra chức năng Xem chi tiết một đơn hàng cụ thể trong lịch sử.

### Trang Thông tin Cá nhân

* Kiểm tra việc hiển thị thông tin cá nhân của người dùng đã đăng nhập.

### Xác thực Admin

* Kiểm tra chức năng Đăng nhập với tài khoản có vai trò "admin".
* Kiểm tra việc từ chối đăng nhập với tài khoản "user" thường vào giao diện Admin.
* Kiểm tra việc bảo vệ các route Admin (chỉ Admin mới truy cập được).

### Dashboard Admin

* Kiểm tra việc hiển thị các số liệu thống kê cơ bản (doanh thu, đơn hàng, user, sản phẩm) từ dữ liệu thật (API backend).
* Kiểm tra các biểu đồ (doanh thu theo tuần, phân bố danh mục) hiển thị đúng dữ liệu.

### Quản lý Người dùng (Admin)

* Kiểm tra chức năng Xem danh sách tất cả người dùng (hiển thị đúng thông tin, vai trò).

### Quản lý Danh mục (Admin)

* Kiểm tra chức năng Xem danh sách danh mục.
* Kiểm tra chức năng Thêm danh mục mới (validate input, lưu thành công).
* Kiểm tra chức năng Sửa thông tin danh mục.
* Kiểm tra chức năng Xóa danh mục (xử lý trường hợp danh mục còn sản phẩm)

### Quản lý Sản phẩm (Admin):

* Kiểm tra chức năng Xem danh sách tất cả sản phẩm.
* Kiểm tra chức năng Thêm sản phẩm mới (validate input, chọn danh mục, upload ảnh - nếu có, lưu thành công).
* Kiểm tra chức năng Sửa thông tin sản phẩm (bao gồm cập nhật tồn kho).
* Kiểm tra chức năng Xóa sản phẩm.

### Quản lý đơn hàng (Admin):

* Kiểm tra chức năng Xem danh sách tất cả đơn hàng (tìm kiếm, lọc theo trạng thái).
* Kiểm tra chức năng Xem chi tiết một đơn hàng.
* Kiểm tra chức năng Cập nhật trạng thái đơn hàng (ví dụ: Pending -> Processing, Processing -> Completed, Pending -> Cancelled).
* Kiểm tra logic hoàn kho khi đơn hàng được Admin chuyển sang trạng thái "Cancelled".
* Kiểm tra việc cập nhật địa chỉ giao hàng cho đơn hàng (nếu trạng thái cho phép).

## Những chức năng không được kiểm thử

Do giới hạn về thời gian và nguồn lực trong khuôn khổ đồ án, các chức năng/khía cạnh sau sẽ **không** nằm trong phạm vi kiểm thử chi tiết:

* **Chức năng nâng cao của E-commerce:** Quản lý khuyến mãi, mã giảm giá, chương trình khách hàng thân thiết, đánh giá sản phẩm, gợi ý sản phẩm, so sánh sản phẩm.
* **Tích hợp cổng thanh toán VNPay thực tế:** Chỉ kiểm thử với môi trường Sandbox và mô phỏng IPN.
* **Quản lý vận chuyển phức tạp:** Tích hợp với các đơn vị vận chuyển, tính phí ship động, theo dõi đơn hàng chi tiết.
* **Kiểm thử Tải (Load Testing) và Stress Testing ở quy mô lớn:** Chỉ thực hiện kiểm tra hiệu năng cơ bản.
* **Kiểm thử Khả năng mở rộng (Scalability Testing) chi tiết.**
* **Kiểm thử Bảo mật sâu rộng:** Ví dụ: Penetration testing, kiểm tra các lỗ hổng OWASP Top 10 chi tiết. Chỉ tập trung vào xác thực và phân quyền cơ bản.
* **Kiểm thử Giao diện trên nhiều thiết bị và trình duyệt phức tạp:** Tập trung vào một vài trình duyệt hiện đại phổ biến trên desktop.
* **Kiểm thử Tính dễ sử dụng (Usability Testing) theo quy trình chuẩn.**
* **Kiểm thử Toàn cầu hóa và Địa phương hóa (Globalization/Localization).**
* **Các chức năng Admin nâng cao chưa được implement:** Ví dụ: Quản lý nội dung website, SEO, marketing too

# Chiến lược kiểm thử

## Test Stages

Trong khuôn khổ dự án này, do giới hạn về thời gian và nguồn lực, chúng tôi sẽ tập trung chủ yếu vào cấp độ kiểm thử End-to-End để đảm bảo các luồng nghiệp vụ hoàn chỉnh hoạt động đúng như mong đợi từ góc độ người dùng. Các cấp độ kiểm thử chi tiết hơn như Unit Testing và Integration Testing cho từng microservice backend được xem xét là các cải tiến tiềm năng cho các giai đoạn phát triển sau này trong một dự án thực tế.

**Kiểm thử Chức năng End-to-End (E2E Functional Testing) - Frontend & Toàn hệ thống:**

* **Mục tiêu:** Xác minh các luồng nghiệp vụ hoàn chỉnh từ giao diện người dùng (cả User và Admin), mô phỏng hành vi người dùng thực tế, đảm bảo sự tương tác đúng đắn giữa Frontend, API Gateway, và các Microservices backend.
* **Cách thực hiện:**
  + **Tự động (Automated E2E):** Sử dụng Selenium WebDriver với Jest và TypeScript (theo cấu hình trong package.json của thư mục e2e-tests). Các script tự động hóa sẽ được phát triển cho các luồng người dùng quan trọng và các kịch bản hồi quy thường xuyên.

## Các loại kiểm thử

**Kiểm thử chức năng**

|  |  |
| --- | --- |
| Mục đích kiểm tra | Đảm bảo các chức năng được kiểm tra hoạt động chính xác theo đặc tả yêu cầu |
| Kỹ thuật | Thực thi tất cả các trường hợp có thể có cho mỗi nhóm chức năng, sử dụng dữ liệu hợp lệ và không hợp lệ để xác định:  - Kết quả mong đợi khi dữ liệu hợp lệ được sử dụng  - Cảnh báo phù hợp hiện ra khi dữ liệu không hợp lệ được sử dụng |
| Tiêu chuẩn dừng | Tất cả các testcase đã được thiết kế đều được thực thi.  Tất cả các lỗi tìm thấy đều được ghi nhận lý do rõ ràng để có thể giúp cho developer khắc phục. |
| Chịu trách nhiệm kiểm thử | Test Designer / Tester |
| Cách kiểm thử | Kiểm thử bằng tay thủ công, tuần tự theo các bước được định nghĩa trong testcase |
| Xử lý ngoại lệ | Liệt kê tất cả các vấn đề liên quan phát sinh trong quá trình thực thi kiểm thử. |

# Điều kiện chấp nhận

Quá trình kiểm thử sẽ được coi là hoàn thành và sản phẩm có thể được chấp nhận (cho mục đích demo/báo cáo đồ án) khi các điều kiện sau được thỏa mãn:

1. Hoàn thành Thực thi Test Case:
   * Tất cả các Test Case chức năng (E2E và API) có độ ưu tiên **Cao (High)** và **Trung bình (Medium)** đã được thực thi ít nhất một lần.
   * Tỷ lệ Test Case "Passed" đạt ít nhất **95%** đối với các chức năng cốt lõi và các luồng nghiệp vụ chính.
   * Các Test Case "Failed" còn lại phải có độ ưu tiên Thấp (Low) hoặc đã được đội ngũ chấp nhận rủi ro (với lý do rõ ràng, ví dụ: không đủ thời gian sửa, ảnh hưởng không lớn).
2. Xử lý Lỗi (Defects):
   * Tất cả các lỗi có mức độ nghiêm trọng **Nghiêm trọng (Critical)** và **Cao (High)** đã được ghi nhận, sửa chữa và kiểm thử lại (re-tested) thành công.
   * Không còn lỗi Critical nào chưa được giải quyết.
   * Số lượng lỗi Medium chưa được giải quyết nằm trong giới hạn chấp nhận được (ví dụ: dưới 5 lỗi) và không ảnh hưởng nghiêm trọng đến các luồng chính.
   * Tất cả các lỗi đã được ghi nhận đầy đủ trong Defect Log/Report.
3. Độ bao phủ Yêu cầu:
   * Các chức năng chính và quan trọng được đặc tả trong Tài liệu Yêu cầu Hệ thống (SRS) đã được kiểm thử và xác minh.
   * Có ma trận truy vết (Traceability Matrix) hoặc bằng chứng cho thấy Test Case đã bao phủ các yêu cầu chính (nếu có yêu cầu).
4. Hoàn thành Kiểm thử Phi Chức năng Cơ bản:
   * Đã thực hiện và ghi nhận kết quả cho các kịch bản kiểm thử hiệu năng API cơ bản (thời gian phản hồi).
   * Đã thực hiện và ghi nhận kết quả cho các kịch bản kiểm thử bảo mật cơ bản (xác thực JWT, phân quyền).
   * Đã thực hiện và ghi nhận kết quả cho các demo về tính chịu lỗi (Health Checks, Failover cơ bản, RabbitMQ message persistence).
5. Tài liệu Kiểm thử Hoàn chỉnh:
   * Tài liệu Kế hoạch Kiểm thử (Test Plan) đã được hoàn thiện và phê duyệt (nếu có quy trình).
   * Tài liệu Test Cases đã được hoàn thiện.
   * Báo cáo Kết quả Kiểm thử (Test Summary Report) đã được tạo, tóm tắt quá trình kiểm thử, các lỗi tìm thấy, trạng thái cuối cùng và đánh giá chất lượng tổng thể.
6. Sẵn sàng cho Demo:
   * Hệ thống chạy ổn định trên môi trường kiểm thử (local Docker Compose).
   * Các luồng demo chính hoạt động trơn tru.
   * Có thể trình bày được các bằng chứng về việc đáp ứng các yêu cầu chức năng và phi chức năng đã kiểm thử.
7. Tuân thủ Giới hạn Thời gian: (Thực tế cho đồ án)
   * Hoạt động kiểm thử được hoàn thành trong khung thời gian cho phép của đồ án.

# Defect Tracking (lỗi)

## Phân loại lỗi

|  |  |
| --- | --- |
| **Mức độ nghiêm trọng** | **Đặc tả lỗi** |
| Critical | * Hệ thống không thể hoạt động (sập toàn bộ, không khởi giao diện nghiêm trọng làm người dùng không thể hoàn thành tác vụ chính. |
| High | * Không sử dụng được chức năng * Không thể đăng ký tạo tài khoản mới * Không thể đăng nhập * Không thể mở thông tin * … |
| Medium | * Tìm kiếm sai * Hiển thị sai thông tin hoạt động của người dùng * Không thể tạo kết nối với Facebook, Linked in * Không thể chỉnh sửa thông tin cá nhân người dùng * Không thể chỉnh sửa thông tin tài khoản * … |
| Low | * Hiển thị kết quả tìm kiếm không đúng mẫu * Hiển thị sai thông tin cá nhân, thông tin tài khoản * Các chức năng chạy không ổn định * … |

## Quy trình xử lý lỗi

Quy trình xử lý lỗi mô tả các bước và trạng thái của một lỗi từ khi được phát hiện cho đến khi được giải quyết hoàn toàn. Quy trình này đảm bảo việc theo dõi lỗi được thực hiện một cách có hệ thống và hiệu quả.

1. **New (Mới):**
   * **Hành động:** Khi Tester phát hiện một lỗi trong quá trình thực thi kiểm thử.
   * **Mô tả:** Tester ghi nhận đầy đủ thông tin về lỗi vào Công cụ Theo dõi Lỗi (ví dụ: file Excel Defect Log, Jira, Trello). Thông tin bao gồm:
     + ID Lỗi (Defect ID - duy nhất)
     + Tiêu đề (Summary - mô tả ngắn gọn, rõ ràng)
     + Người báo cáo (Reported By)
     + Ngày báo cáo (Date Reported)
     + Phiên bản phần mềm/Build ID (Version/Build ID)
     + Môi trường kiểm thử (Environment - ví dụ: Local Docker, Trình duyệt Chrome vX)
     + Module/Chức năng bị ảnh hưởng
     + Các bước để tái hiện lỗi (Steps to Reproduce - chi tiết, rõ ràng)
     + Kết quả thực tế (Actual Result)
     + Kết quả mong đợi (Expected Result)
     + Mức độ nghiêm trọng (Severity - do Tester đề xuất)
     + Mức độ ưu tiên (Priority - do Tester đề xuất, có thể được Test Lead điều chỉnh)
     + Ảnh chụp màn hình/Video minh họa (Attachments)
   * **Trạng thái tiếp theo:** Open (Mở)
2. **Open (Mở):**
   * **Hành động:** Test Lead (hoặc người quản lý kiểm thử) xem xét lỗi vừa được báo cáo.
   * **Mô tả:**
     + Xác minh tính hợp lệ của lỗi (có phải là lỗi thực sự hay không, có trùng lặp với lỗi đã báo cáo trước đó không).
     + Nếu lỗi hợp lệ và không trùng lặp, Test Lead xác nhận Severity và Priority (có thể điều chỉnh nếu cần).
     + Lỗi được chấp nhận và sẵn sàng để gán cho đội ngũ phát triển.
   * **Trạng thái tiếp theo:** Assigned (Đã gán) hoặc Rejected (Từ chối - nếu không hợp lệ) hoặc Duplicate (Trùng lặp).
3. **Assigned (Đã gán):**
   * **Hành động:** Test Lead gán lỗi cho một Developer cụ thể (hoặc cho Development Team Lead để phân công).
   * **Mô tả:** Developer được chỉ định sẽ chịu trách nhiệm phân tích và sửa lỗi này.
   * **Trạng thái tiếp theo:** In Progress (Đang tiến hành) hoặc Resolved/Fixed (Đã sửa).
4. **In Progress (Đang tiến hành):**
   * **Hành động:** Developer bắt đầu quá trình phân tích nguyên nhân và sửa lỗi.
   * **Mô tả:** Developer có thể yêu cầu thêm thông tin từ Tester nếu cần.
   * **Trạng thái tiếp theo:** Resolved/Fixed (Đã sửa) hoặc Deferred (Hoãn lại) hoặc Cannot Reproduce (Không thể tái hiện).
5. **Resolved / Fixed (Đã sửa):**
   * **Hành động:** Developer đã hoàn thành việc sửa lỗi và deploy phiên bản đã sửa lên môi trường kiểm thử.
   * **Mô tả:** Developer thông báo cho Tester rằng lỗi đã được sửa và sẵn sàng để kiểm thử lại. Developer nên cung cấp thông tin về cách sửa lỗi hoặc các thay đổi liên quan.
   * **Trạng thái tiếp theo:** Ready for Retest (Sẵn sàng kiểm thử lại).
6. **Ready for Retest (Sẵn sàng kiểm thử lại):**
   * **Hành động:** Tester nhận được thông báo lỗi đã được sửa.
   * **Mô tả:** Tester chuẩn bị để thực hiện kiểm thử lại lỗi trên phiên bản phần mềm đã được cập nhật.
   * **Trạng thái tiếp theo:** Retesting (Đang kiểm thử lại).
7. **Retesting (Đang kiểm thử lại):**
   * **Hành động:** Tester thực hiện lại các bước đã gây ra lỗi ban đầu để xác minh việc sửa lỗi.
   * **Mô tả:** Tester cũng nên kiểm tra các khu vực liên quan để đảm bảo việc sửa lỗi không gây ra lỗi mới (Regression Testing cơ bản).
   * **Trạng thái tiếp theo:** Verified/Closed (Đã xác minh/Đóng) hoặc Reopened (Mở lại).
8. **Verified / Closed (Đã xác minh / Đóng):**
   * **Hành động:** Tester xác nhận lỗi đã được sửa hoàn toàn và không còn xuất hiện.
   * **Mô tả:** Lỗi được đóng lại và không cần theo dõi thêm.
   * **Trạng thái cuối cùng.**
9. **Reopened (Mở lại):**
   * **Hành động:** Tester phát hiện lỗi vẫn còn tồn tại hoặc việc sửa lỗi gây ra vấn đề mới.
   * **Mô tả:** Tester cung cấp thêm thông tin chi tiết, bằng chứng về việc lỗi vẫn còn và gán lại cho Developer.
   * **Trạng thái tiếp theo:** Assigned (Đã gán) hoặc In Progress (Đang tiến hành).
10. **Rejected (Từ chối):**
    * **Hành động:** Test Lead hoặc Developer xác định rằng báo cáo lỗi không hợp lệ (ví dụ: không phải lỗi của phần mềm, do hiểu sai yêu cầu, lỗi môi trường của Tester, hoặc không thể tái hiện được sau nhiều lần thử).
    * **Mô tả:** Cần có lý do từ chối rõ ràng.
    * **Trạng thái cuối cùng (cho lỗi đó).**
11. **Duplicate (Trùng lặp):**
    * **Hành động:** Test Lead phát hiện lỗi này đã được báo cáo trước đó.
    * **Mô tả:** Liên kết lỗi này với lỗi gốc và đóng lỗi trùng lặp.
    * **Trạng thái cuối cùng (cho lỗi đó).**
12. **Deferred / Postponed (Hoãn lại):**
    * **Hành động:** Sau khi xem xét, đội ngũ dự án (bao gồm Product Owner, Test Lead, Dev Lead) quyết định lỗi này sẽ không được sửa trong phiên bản hiện tại (ví dụ: do độ ưu tiên thấp, rủi ro sửa lỗi cao, không đủ thời gian).
    * **Mô tả:** Lỗi sẽ được xem xét để sửa trong các phiên bản hoặc giai đoạn sau.
    * **Trạng thái cuối cùng (cho phiên bản hiện tại).**

**Công cụ theo dõi lỗi:**

* Sử dụng file Excel hoặc Google Sheets với các cột tương ứng với thông tin lỗi và trạng thái trong quy trình trên.
* Hoặc các công cụ chuyên dụng như Jira, Trello, MantisBT (nếu có điều kiện).

# Test Deliverables

## Test Cases

Xem mẫu Test case

## Test Reports

Xem mẫu Test case