**BÁO CÁO BÀI TẬP**

**TEST PLAN**

**KIỂM THỬ PHẦN MỀM**

**GV hướng dẫn: TS. Đỗ Như Tài**

**Lớp: DCT122C3 – Mã học phần: 841408**

**Sinh viên: Trịnh Long Phát – 3122411150**

**Lê Hồng Phát – 3122411145**

**Trương Phú Kiệt – 3122411109**

**Trà Đức Toàn – 3122411217**

**Thành phố Hồ Chí Minh, tháng 9 năm 2025**

**ABC project**

**Test Plan**

**Version 1.0**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 29/09/2025 | 1.0 | Initial edition |  |
|  |  |  |  |

**Revision History**

**MỤC LỤC**

[1. Introduction 7](#_Toc210236841)

[1.1 Purpose 7](#_Toc210236842)

[1.2 Background 8](#_Toc210236843)

[1.3 Scope 9](#_Toc210236844)

[1.4 Project Identification 12](#_Toc210236845)

[1.5 References 15](#_Toc210236846)

[2. Feature to be tested 15](#_Toc210236847)

[2.1 Functionality 16](#_Toc210236848)

[2.1.1 Quản lý sản phẩm (Product Management) 16](#_Toc210236849)

[2.1.2 Quản lý người dùng (User Management) 16](#_Toc210236850)

[2.1.3 Giỏ hàng và đơn hàng (Cart & Order Management) 16](#_Toc210236851)

[2.1.4 Thanh toán (Payment) 17](#_Toc210236852)

[2.1.5 Vận chuyển và giao hàng (Delivery) 17](#_Toc210236853)

[2.1.6 Live Chat & Hỗ trợ khách hàng 17](#_Toc210236854)

[2.1.7 Bình luận & đánh giá sản phẩm (Comment & Rating) 17](#_Toc210236855)

[2.1.8 Quản trị hệ thống (Admin Management) 17](#_Toc210236856)

[2.2 Usability 18](#_Toc210236857)

[2.2.1 Responsive Design Support 18](#_Toc210236858)

[2.2.2 Multi-session / Login Support 18](#_Toc210236859)

[2.2.3 Font & Language Display 18](#_Toc210236860)

[2.2.4 Navigation & User Flow 19](#_Toc210236861)

[2.2.5 Accessibility & Error Messages 19](#_Toc210236862)

[2.2.6 Browser Compatibility 19](#_Toc210236863)

[2.3 Design Constraints 19](#_Toc210236864)

[2.3.1 Maximum message length 19](#_Toc210236865)

[2.3.2 Test environment 20](#_Toc210236866)

[2.3.3 Characters Sets & Localization 21](#_Toc210236867)

[2.4 Interfaces 21](#_Toc210236868)

[2.4.1 User Interfaces 21](#_Toc210236869)

[2.4.2 System Interfaces 22](#_Toc210236870)

[3. Feature not to be tested 24](#_Toc210236871)

[3.1 Setup a user account 24](#_Toc210236872)

[3.2 Remove user account 24](#_Toc210236873)

[4. Test Strategy 24](#_Toc210236874)

[4.1 Testing Types 25](#_Toc210236875)

[4.1.1 Function and Usability Testing 25](#_Toc210236876)

[4.1.2 User Interface Testing 25](#_Toc210236877)

[4.2 Tools 26](#_Toc210236878)

[5. Resources 26](#_Toc210236879)

[5.1 Roles 27](#_Toc210236880)

[6. Project Milestones 29](#_Toc210236881)

[7. Deliverables 30](#_Toc210236882)

[7.1 Test Model 30](#_Toc210236883)

[7.2 Test Logs 31](#_Toc210236884)

[7.3 Bug Reports 31](#_Toc210236885)

**Test Plan**

# 1. Introduction

## 1.1 Purpose

Mục tiêu của tài liệu này là xác định **chiến lược kiểm thử (Test Strategy)** và **quy trình đảm bảo chất lượng (QA process)** cho ứng dụng **Web Bán Quần Áo** có tích hợp **CI/CD pipeline**.

Tài liệu này thiết lập kế hoạch kiểm thử hệ thống (System Test Plan) nhằm đảm bảo:

* Ứng dụng web thương mại điện tử hoạt động ổn định, đáp ứng các yêu cầu nghiệp vụ về quản lý sản phẩm, giỏ hàng, thanh toán, quản lý người dùng, và khuyến mãi.
* Các thay đổi và cập nhật (deployments) trong quy trình CI/CD được triển khai tự động nhưng vẫn đảm bảo **chất lượng phần mềm** và **không ảnh hưởng đến chức năng hiện có**.
* Tích hợp kiểm thử tự động (automated testing) vào pipeline CI/CD để phát hiện lỗi sớm, giảm thiểu rủi ro khi phát hành.
* Các chức năng chính như đăng nhập, đăng ký, tìm kiếm sản phẩm, đặt hàng, thanh toán và quản lý đơn hàng được kiểm thử toàn diện.
* Đảm bảo ứng dụng có thể mở rộng, bảo mật và duy trì hiệu suất khi số lượng người dùng và đơn hàng tăng.

Đối tượng đọc tài liệu này bao gồm: **đội phát triển (Dev team), QA/Test team, Business Analyst, Quản lý dự án (PM), và các bên liên quan khác**.

## 1.2 Background

Ứng dụng **Website Bán Quần Áo** được xây dựng bằng **ReactJS (Frontend)** và **NodeJS (Backend – API, Socket)**, kết hợp **MongoDB Atlas** làm cơ sở dữ liệu chính. Hệ thống hỗ trợ mô hình **CI/CD pipeline**, cho phép tự động build, kiểm thử và triển khai nhằm đảm bảo chất lượng phần mềm và rút ngắn thời gian phát hành.

Ứng dụng được phát triển nhằm cung cấp một nền tảng thương mại điện tử hiện đại, nơi khách hàng có thể dễ dàng:

* Xem và tìm kiếm sản phẩm theo danh mục, giới tính, hoặc nhiều tiêu chí khác.
* Thêm sản phẩm vào giỏ hàng, đặt hàng trực tuyến.
* Thanh toán qua **PayPal** hoặc **ship COD**.
* Tra cứu trạng thái vận chuyển trên bản đồ thông qua **Google Map API**.
* Nhận tư vấn trực tiếp qua **live chat (Socket.IO)** và **email xác nhận đơn hàng**.

Các chức năng quản trị (Admin) bao gồm:

* CRUD sản phẩm, người dùng, đơn hàng, phân quyền.
* Quản lý bình luận, đánh giá (rating).
* Theo dõi số lượng hàng tồn kho, doanh thu và báo cáo.

Hệ thống được thiết kế theo mô hình **kết hợp API** với các công nghệ hỗ trợ:

* **NodeJS + Express**: xử lý backend API và WebSocket cho live chat.
* **MongoDB + Mongoose**: quản lý dữ liệu sản phẩm, người dùng, đơn hàng.
* **ReactJS**: giao diện người dùng responsive.
* **Nodemailer**: gửi email xác nhận đơn hàng.
* **Docker**: container hóa các thành phần, tích hợp CI/CD.
* **CI/CD Pipeline**: tự động hóa build, test, deploy.

## 1.3 Scope

**Testing Phases**

|  |  |  |
| --- | --- | --- |
| **Giai đoạn kiểm thử** | **Đội chịu trách nhiệm** | **Mô tả** |
| Kiểm thử đơn vị  (Unit Testing) | Development Team | Kiểm thử từng **module nhỏ** (ví dụ: API đăng nhập, API thêm sản phẩm, component hiển thị giỏ hàng).  Mục tiêu: đảm bảo mỗi hàm, mỗi đoạn code chạy đúng logic, đầu vào/ra hợp lệ. |
| Kiểm thử tích hợp (Integration Testing) | QA team | Kiểm thử sự **tương tác giữa các module** (ví dụ: khi đặt hàng → giỏ hàng cập nhật → tạo đơn hàng → thanh toán PayPal → gửi email).  Mục tiêu: đảm bảo các thành phần kết nối đúng luồng dữ liệu. |
| Kiểm thử hệ thống (System Testing) | QA team | Kiểm thử **toàn bộ hệ thống** trên môi trường gần giống production (frontend, backend, database, CI/CD pipeline).  Bao gồm: kiểm thử chức năng, bảo mật, giao diện, hiệu năng.  Mục tiêu: đảm bảo hệ thống hoạt động đúng yêu cầu nghiệp vụ ban đầu. |
| Kiểm thử chấp nhận người dùng (UAT) | Người dùng đại diện hoặc khách hàng | Kiểm thử các kịch bản thực tế (ví dụ: khách hàng đăng ký, chọn quần áo, đặt hàng, thanh toán, nhận email xác nhận).  Mục tiêu: xác nhận rằng hệ thống sẵn sàng để triển khai chính thức. |

**Testing Types**

|  |  |  |
| --- | --- | --- |
| **Loại kiểm thử** | **Đội chịu trách nhiệm** | **Mục tiêu** |
| Kiểm thử chức năng (Functionality Testing) | QA team | Đảm bảo các tính năng nghiệp vụ chạy đúng: CRUD sản phẩm, giỏ hàng, thanh toán, quản lý đơn hàng.  Dựa trên yêu cầu đã ghi trong Use Case và ERD. |
| Kiểm thử cơ sở dữ liệu (Database Testing) | QA team + DBA | Kiểm thử tính toàn vẹn dữ liệu (data integrity), truy vấn, và sự đúng đắn khi thêm/xóa/sửa dữ liệu.  Ví dụ: khi hủy đơn hàng, số lượng tồn kho phải được cập nhật lại chính xác. |
| Kiểm thử bảo mật (Security Testing) | QA team | Đảm bảo người dùng không thể truy cập trái phép (ví dụ: bypass login, SQL injection, XSS).  Kiểm thử phân quyền: Admin, Nhân viên, Khách hàng chỉ được truy cập chức năng phù hợp. |
| Kiểm thử giao diện và khả năng sử dụng (GUI and Usability Testing) | QA team | Kiểm tra bố cục, hiển thị sản phẩm, thông báo lỗi rõ ràng, tính dễ dùng.  Ví dụ: hiển thị sản phẩm theo phân loại, nút “Thanh toán” rõ ràng, không lỗi font. |
| Kiểm thử hiệu năng & tải (Performance & Load Testing) | QA team | Đánh giá tốc độ phản hồi khi có nhiều người dùng cùng lúc (ví dụ: 500 user truy cập giỏ hàng).  Kiểm thử CI/CD pipeline: build và test có chạy ổn định khi load cao không. |
| Kiểm thử khói (Smoke Testing) | Dev team + QA team | Kiểm thử nhanh các chức năng chính để đảm bảo build mới không bị “gãy” trước khi QA đi sâu hơn.  Ví dụ: kiểm tra login, thêm sản phẩm, checkout có chạy không. |
| Kiểm thử hồi quy (Regression Testing) | QA team | Thực hiện sau mỗi lần cập nhật, đảm bảo các tính năng cũ không bị lỗi do code mới.  Ví dụ: thêm tính năng khuyến mãi nhưng không làm lỗi giỏ hàng. |
| Kiểm thử tương thích (Compatible Testing) | QA team | Đảm bảo website chạy ổn định trên nhiều **trình duyệt (Chrome, Firefox, Edge)** và thiết bị (desktop, mobile). |
| Kiểm thử tự do (Adhoc Testing) | QA team | Tester kiểm thử theo kinh nghiệm và trực giác, không theo test case cứng nhắc.  Ví dụ: nhập dữ liệu bất thường (số điện thoại = chữ) để kiểm tra phản ứng của hệ thống. |

## 1.4 Project Identification

The table below identifies the documentation and availability used for developing the **Test Plan:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Document (and version /date)** | **Created or Available** | **Received or Reviewed** | **Author or Resource** | **Notes** |
| Requirements Specification | **** Yes  No | **** Yes  No | Nhóm phát triển (Team Members) | Yêu cầu hệ thống từ README và Use Case. |
| Functional Specification | **** Yes  No | **** Yes  No | Nhóm phát triển | Mô tả chức năng: CRUD sản phẩm, giỏ hàng, thanh toán, live chat… |
| Use-Case Reports | **** Yes  No | **** Yes  No | Nhóm phát triển | Bao gồm Use Case cho Khách hàng, Nhân viên, Admin. |
| Project Plan | **** Yes  No | **** Yes  No | Quản lý dự án (PM) | Timeline phát triển + tích hợp CI/CD |
| Design Specifications | **** Yes  No | **** Yes  No | Nhóm phát triển | Thiết kế database (ERD, bảng Product, Order, User…). |
| Prototype |  Yes  No |  Yes  No |  |  |
| User’s Manuals |  Yes No |  Yes  No | README.md trong repo | Hướng dẫn setup MongoDB Atlas, chạy server/client, Docker. |
| Business Model or Flow | Yes  No | Yes  No | README + Use Case | Mô tả quy trình mua hàng, giỏ hàng, thanh toán. |
| Data Model or Flow | Yes  No |  Yes  No | Database schema (MongoDB) | Bảng Product, Category, User, Order, Comment, Favorite, Payment. |
| Business Functions and Rules |  Yes  No |  Yes  No |  |  |
| Project or Business Risk  Assessment |  Yes  No |  Yes  No |  |  |

## 1.5 References

The following table contains references to external documents.

|  |  |  |
| --- | --- | --- |
| **No.** | **Document** | **Description** |
| 1 | README.md | Mô tả chung đề tài, các chức năng chính (CRUD sản phẩm, giỏ hàng, thanh toán PayPal, live chat, gửi email, vận chuyển qua Google Map API), hướng dẫn cài đặt và công nghệ sử dụng (ReactJS, NodeJS, MongoDB, Docker, CI/CD). |
| 2 | Database ERD & Thiết kế mô hình dữ liệu | Sơ đồ ERD và mô tả quan hệ các bảng (Product, Category, User, Order, Payment, Comment, Favorite). |
| 3 | Use Case Diagrams | Các biểu đồ Use Case cho Khách hàng, Nhân viên, Admin (tham khảo từ README) |

# 2. Feature to be tested

The listing below identifies those items − use cases, functional requirements, and non-functional requirements − that have been

identified as targets for testing. This list represents what is to be tested.

## 2.1 Functionality

### 2.1.1 Quản lý sản phẩm (Product Management)

* **CRUD sản phẩm**: Thêm, sửa, xóa, tìm kiếm sản phẩm.
* **Phân loại sản phẩm**: theo Category, Giới tính, Giá.
* **Phân trang và lọc sản phẩm** (paging, filtering, sorting).
* **Xem chi tiết sản phẩm**: tên, mô tả, hình ảnh, giá, số lượng tồn kho.

### 2.1.2 Quản lý người dùng (User Management)

* **Đăngký/Đăng nhập/Đăng xuất**.
* **Đặt lại mật khẩu (Reset password)**.
* **Phân quyền người dùng**: Khách hàng, Nhân viên bán hàng, Admin.
* **Quản lý hồ sơ cá nhân (Profile)**: cập nhật tên, email, mật khẩu.

### 2.1.3 Giỏ hàng và đơn hàng (Cart & Order Management)

* Thêm/Xóa sản phẩm trong giỏ hàng.
* Cập nhật số lượng sản phẩm trong giỏ hàng.
* Tạo đơn hàng (Order): nhập thông tin giao hàng, tính tổng tiền.
* Theo dõi trạng thái đơn hàng: Chưa xác nhận → Đã xác nhận → Đang giao → Hoàn thành/Hủy.
* Xem lịch sử mua hàng (Order history).

### 2.1.4 Thanh toán (Payment)

* Thanh toán PayPal (Sandbox).
* Thanh toán khi nhận hàng (COD).
* Xác nhận đơn hàng qua email (Nodemailer).
* Kiểm tra dữ liệu thanh toán được lưu chính xác vào MongoDB.

### 2.1.5 Vận chuyển và giao hàng (Delivery)

* **Tính phí vận chuyển** dựa trên khoảng cách bằng **Google Map API**.
* **Hiển thị tuyến đường từ cửa hàng → khách hàng**.
* **Quản lý phương thức giao hàng** (Delivery info trong Order).

### 2.1.6 Live Chat & Hỗ trợ khách hàng

* **Tính phí vận chuyển** dựa trên khoảng cách bằng **Google Map API**.
* **Hiển thị tuyến đường từ cửa hàng → khách hàng**.
* **Quản lý phương thức giao hàng** (Delivery info trong Order).

### 2.1.7 Bình luận & đánh giá sản phẩm (Comment & Rating)

* **Khách hàng thêm bình luận (Comment)**.
* **Đánh giá sản phẩm bằng sao (1–5)**.
* **Admin kiểm duyệt/xóa bình luận** nếu vi phạm.

### 2.1.8 Quản trị hệ thống (Admin Management)

* **Quản lý sản phẩm**: CRUD toàn bộ sản phẩm.
* **Quản lý người dùng**: tạo tài khoản nhân viên, phân quyền, khóa/mở khóa user.
* **Quản lý đơn hàng**: xác nhận, thay đổi trạng thái, hủy đơn hàng.
* **Quản lý bình luận, yêu thích (Favorite)**.
* **Xem báo cáo/thống kê** (doanh thu, số lượng bán ra, sản phẩm còn hàng/hết hàng).

## 2.2 Usability

Các kiểm thử về khả năng sử dụng (Usability) sẽ tập trung vào trải nghiệm người dùng của website bán quần áo. Mục tiêu là đảm bảo giao diện trực quan, dễ sử dụng, hiển thị đúng nội dung, và tương thích trên nhiều thiết bị.

### 2.2.1 Responsive Design Support

* Xác minh website hiển thị đúng trên **desktop, tablet, mobile**.
* Đảm bảo bố cục (layout), hình ảnh sản phẩm, nút chức năng (mua hàng, giỏ hàng, thanh toán) không bị vỡ layout.
* Kiểm tra hiển thị ở nhiều độ phân giải màn hình (FullHD, 2K, Mobile 360px).

### 2.2.2 Multi-session / Login Support

* Đảm bảo người dùng có thể **đăng nhập, đăng xuất nhiều lần** mà không mất dữ liệu giỏ hàng/đơn hàng.
* Kiểm tra việc duy trì **session** sau khi reload trang hoặc quay lại từ tab khác.
* Kiểm tra session timeout (người dùng không hoạt động trong 30 phút).

### 2.2.3 Font & Language Display

* Kiểm tra **font chữ** hiển thị rõ ràng, dễ đọc.
* Đảm bảo không bị lỗi font khi hiển thị **tiếng Việt có dấu** (Unicode).
* Kiểm tra hiển thị ký tự đặc biệt trong tên sản phẩm, bình luận, địa chỉ giao hàng.

### 2.2.4 Navigation & User Flow

* Đảm bảo người dùng có thể điều hướng dễ dàng giữa **Trang chủ → Danh mục → Sản phẩm chi tiết → Giỏ hàng → Thanh toán**.
* Kiểm tra link, button, menu, breadcrumbs hoạt động đúng.
* Đảm bảo luồng người dùng không bị “kẹt” (dead end).

### 2.2.5 Accessibility & Error Messages

* Đảm bảo form nhập liệu (đăng ký, đăng nhập, thanh toán) có thông báo lỗi rõ ràng khi nhập sai định dạng (ví dụ: email không hợp lệ, số điện thoại thiếu).
* Kiểm tra màu sắc, độ tương phản (contrast ratio) đảm bảo khả năng đọc.
* Các thông báo thành công/thất bại (toast, popup) phải hiển thị đúng và dễ hiểu.

### 2.2.6 Browser Compatibility

* Kiểm thử hiển thị và usability trên các trình duyệt chính: **Chrome, Firefox, Edge**.
* Kiểm tra một số trình duyệt mobile (Chrome Android, Safari iOS).

## 2.3 Design Constraints

### 2.3.1 Maximum message length

Xác minh các trường nhập liệu có giới hạn hợp lý:

* **Tên người dùng**: tối đa 50 ký tự.
* **Mật khẩu**: tối thiểu 8 ký tự, tối đa 64 ký tự, bao gồm chữ hoa, chữ thường, số, ký tự đặc biệt.
* **Tên sản phẩm**: tối đa 100 ký tự.
* **Mô tả sản phẩm**: tối đa 500 ký tự.
* **Địa chỉ giao hàng**: tối đa 200 ký tự.

Đảm bảo hệ thống hiển thị thông báo lỗi rõ ràng nếu người dùng vượt quá giới hạn.

### 2.3.2 Test environment

Kiểm thử sẽ được tiến hành trong các môi trường sau:

* **Frontend**: ReactJS chạy trên NodeJS, truy cập qua trình duyệt Chrome, Firefox, Edge.
* **Backend**: NodeJS + Express API, chạy trên Docker container.
* **Database**: MongoDB Atlas (cluster cloud).
* **CI/CD Pipeline**: GitHub Actions/GitLab CI/Jenkins (tùy thiết lập nhóm).
* **Payment Sandbox**: PayPal sandbox để kiểm thử thanh toán.
* **Email Service**: SMTP (Nodemailer, Gmail/SendGrid).
* **Map API**: Google Map API để hiển thị tuyến đường và tính phí giao hàng.

### 2.3.3 Characters Sets & Localization

* Hệ thống phải hỗ trợ **Unicode UTF-8** để hiển thị chính xác **tiếng Việt có dấu** và ký tự đặc biệt.
* Tất cả dữ liệu (tên sản phẩm, email, địa chỉ, bình luận) phải lưu trữ và truy xuất được trong MongoDB mà không mất ký tự.
* Website hiện chỉ hỗ trợ **tiếng Việt** và **tiền tệ VND/USD qua PayPal**.
* Các ngôn ngữ và tiền tệ khác chưa nằm trong phạm vi kiểm thử (Not in Scope).

## 2.4 Interfaces

### 2.4.1 User Interfaces

Các kiểm thử giao diện người dùng (UI) sẽ tập trung vào việc đảm bảo website hiển thị đúng và dễ sử dụng trên cả **giao diện khách hàng (User)** và **giao diện quản trị (Admin)**.

Giao diện Khách hàng

* **Trang chủ**: hiển thị sản phẩm nổi bật, danh mục sản phẩm.
* **Trang danh mục (Category Page)**: lọc, phân loại, phân trang sản phẩm.
* **Trang chi tiết sản phẩm**: hình ảnh, tên, mô tả, giá, tồn kho, nút “Thêm vào giỏ hàng”.
* **Giỏ hàng (Cart Page)**: danh sách sản phẩm đã chọn, số lượng, tổng tiền, nút “Thanh toán”.
* **Thanh toán (Checkout Page)**: nhập thông tin người nhận, chọn phương thức thanh toán (PayPal, COD).
* **Theo dõi đơn hàng (Order Tracking)**: hiển thị trạng thái đơn hàng và phí vận chuyển trên Google Map.
* **Live Chat**: khung chat với nhân viên tư vấn (Socket.IO).
* **Đăng ký/Đăng nhập/Quên mật khẩu**: form nhập liệu với xác thực dữ liệu đầu vào (validation).

Giao diện Quản trị (Admin Panel)

* **Quản lý sản phẩm**: CRUD sản phẩm (tên, giá, hình ảnh, mô tả, tồn kho).
* **Quản lý người dùng**: phân quyền (Khách hàng, Nhân viên, Admin).
* **Quản lý đơn hàng**: xác nhận, thay đổi trạng thái, hủy đơn hàng
* **Quản lý bình luận/đánh giá**: xem, xóa, xử lý các comment không hợp lệ.
* **Thống kê/Báo cáo**: doanh thu, sản phẩm bán chạy, hàng tồn kho.

### 2.4.2 System Interfaces

Các giao diện tích hợp giữa hệ thống và dịch vụ bên ngoài:

* **Payment Gateway (PayPal Sandbox)**
  + Kiểm thử tích hợp API PayPal ho thanh toán trực tuyến.
  + Đảm bảo dữ liệu giao dịch lưu vào hệ thống sau khi thanh toán thành công.
* **Email Service (Nodemailer, Gmail/SendGrid)**
  + Gửi email xác nhận đơn hàng cho khách hàng.
  + Kiểm thử nội dung email, độ trễ và độ tin cậy.
* **Google Map API**
  + Hiển thị tuyến đường vận chuyển từ cửa hàng đến khách hàng.
  + Tính toán phí vận chuyển dựa trên khoảng cách.
* **CI/CD Pipeline Integration**
  + Tích hợp GitHub Actions/GitLab CI/Jenkins.
  + Tự động chạy unit test, regression test trước khi deploy.

# 3. Feature not to be tested

## 3.1 Setup a user account

* + Kiểm thử khôi phục sau sự cố
  + Các tính năng nâng cấp tương lai như AI Recommendation, Chatbot
  + Hệ thống chỉ hỗ trợ chức năng **đăng ký tài khoản** theo form chuẩn (username, email, password).
  + Các chức năng nâng cao như thiết lập nhiều tài khoản, lưu “remember password” hay tích hợp tài khoản mạng xã hội (Google, Facebook) **không nằm trong phạm vi kiểm thử**.

## 3.2 Remove user account

* + - Chức năng **xóa tài khoản vĩnh viễn** (account deletion) hiện **chưa được phát triển** trong phiên bản này.
    - Do đó, kiểm thử sẽ **không bao gồm** việc xác minh quy trình xóa tài khoản, xử lý dữ liệu liên quan hay phục hồi dữ liệu sau khi xóa.

# 4. Test Strategy

## 4.1 Testing Types

### 4.1.1 Function and Usability Testing

|  |  |
| --- | --- |
| Test Objective: | Để đảm bảo chức năng mục tiêu thử nghiệm phù hợp, bao gồm điều hướng, nhập dữ liệu, xử lý và truy xuất |
| Technique: | Thực thi từng **use case**, luồng xử lý hoặc chức năng, với cả dữ liệu hợp lệ và dữ liệu không hợp lệ, để xác minh:   * Kết quả mong đợi xảy ra khi sử dụng dữ liệu hợp lệ (ví dụ: thêm sản phẩm vào giỏ hàng thành công). * Thông báo lỗi hoặc cảnh báo phù hợp hiển thị khi sử dụng dữ liệu không hợp lệ (ví dụ: nhập sai định dạng email khi đăng ký). |
| Completion Criteria: | • Tất cả các thử nghiệm theo kế hoạch đã được thực hiện.  • Tất cả các lỗi được xác định đã được khắc phục. |
| Special Considerations: | Xác định hoặc mô tả những vấn đề (nội bộ hoặc bên ngoài) có thể ảnh hưởng đến việc triển khai và thực hiện kiểm thử chức năng, ví dụ: môi trường kiểm thử chưa ổn định, dữ liệu mẫu chưa đầy đủ, hoặc tích hợp với hệ thống thanh toán bên thứ ba (Paypal) và API bản đồ (Google Maps). |

### 4.1.2 User Interface Testing

Kiểm thử khả dụng sẽ được thực hiện bằng cách xác minh rằng thông tin hiển thị trên từng màn hình (trang chủ, danh mục sản phẩm, chi tiết sản phẩm, giỏ hàng, thanh toán, hồ sơ người dùng) là chính xác và rõ ràng. Các menu, nút bấm, biểu tượng và liên kết điều hướng sẽ được kiểm tra để đảm bảo trải nghiệm duyệt và mua hàng diễn ra thuận lợi.

Tầm quan trọng đặc biệt được đặt vào:

* Hình ảnh sản phẩm (chất lượng, phóng to, carousel hiển thị nhiều ảnh).
* Nội dung mô tả, giá và khuyến mãi được trình bày chính xác.
* Trình bày danh mục, bộ lọc, sắp xếp sản phẩm (theo giới tính, kích thước, giá).
* Quy trình giỏ hàng và thanh toán rõ ràng, dễ sử dụng.
* Thông báo phản hồi (ví dụ: “Thêm sản phẩm vào giỏ thành công”, “Đặt hàng thành công”, “Thanh toán thất bại”).
* Tính nhất quán của bố cục, font chữ, màu sắc, nút và các thành phần giao diện.
* Các khía cạnh về khả năng truy cập như độ tương phản màu, độ dễ đọc của chữ, và gợi ý kiểm tra lỗi trong form (ví dụ: đăng nhập, đăng ký, thanh toán).

Kiểm thử giao diện và tính khả dụng sẽ được tiến hành song song với kiểm thử chức năng trong mọi bản build, bắt đầu từ bản build đầu tiên (chu kỳ kiểm thử 2 tuần) cho đến bản build cuối cùng trước khi phát hành.

## 4.2 Tools

Các tool được tích hợp trong dự án:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Tool** | **Vendor/In-house** | **Version** |
| Bug Tracking | Mantis |  | Lastest |
| Project Management | Github |  | Lastest |

# 5. Resources

## 5.1 Roles

Các vai trò được sử dụng trong dự án:

|  |  |  |
| --- | --- | --- |
| **Human Resources** | | |
| Employee | Minimum Resources Recommended  (number of full-time roles allocated) | Specific Responsibilities/Job Description |
| Test Developer | 1 | * Xác định, ưu tiên và triển khai các test case. * Xây dựng kế hoạch kiểm thử (Test Plan). * Đánh giá hiệu quả của hoạt động kiểm thử. * Hỗ trợ nhóm phát triển trong việc chuẩn bị dữ liệu kiểm thử. |
| Tester | 2 | * Thực hiện kiểm thử theo các test case. * Viết và cập nhật đặc tả test case. * Thực thi test case và ghi nhận kết quả. * Ghi log lỗi vào hệ thống (GitHub Issues). * Theo dõi việc sửa lỗi và xác nhận lại (re-test). * Đề xuất chỉnh sửa/tài liệu nếu phát hiện thiếu sót. |

# 6. Project Milestones

Hoạt động và mốc kiểm thử phụ thuộc vào tiến trình phát triển. Pha xây dựng được chia thành 3 build chính, sau đó là bản release candidate và bản release chính thức. Mỗi build bao gồm chu kỳ kiểm thử đầy đủ: lập kế hoạch, thiết kế, thực thi và đánh giá.

|  |  |  |  |
| --- | --- | --- | --- |
| **Milestone Task** | **Effort** | **Start Date** | **End Date** |
| **Build 1** – Plan Test, Design Test, Execute Test, Evaluate Test | 5 ngày | 01/10/2025 | 05/10/2025 |
| **Build 2** – Plan Test, Design Test, Execute Test, Evaluate Test | 5 ngày | 06/10/2025 | 10/10/2025 |
| **Build 3** – Plan Test, Design Test, Execute Test, Evaluate Test | 5 ngày | 11/10/2025 | 15/10/2025 |
| **Release Candidate** – Full System Test, Regression Test, UAT | 4 ngày | 16/10/2025 | 19/10/2025 |
| **Release Version –** Final Acceptance, Test Summary Report | 3 ngày | 10/10/2025 | 22/10/2025 |

# 7. Deliverables

## 7.1 Test Model

Mô hình kiểm thử bao gồm toàn bộ **test case, test scenario, quy trình kiểm thử và test script** được xây dựng cho từng chức năng của hệ thống. Các tài liệu này sẽ đảm bảo rằng mọi yêu cầu (requirement) đều được ánh xạ (traceability) với các trường hợp kiểm thử tương ứng.

## 7.2 Test Logs

Kết quả thực thi test case sẽ được ghi lại trong **Test Execution Log**. Nhật ký kiểm thử (test log) sẽ lưu trữ các thông tin: ngày kiểm thử, người thực hiện, môi trường kiểm thử, dữ liệu sử dụng và trạng thái (Pass/Fail). Tài liệu này sẽ được trình bày dưới dạng **Word/Excel** để tiện theo dõi và báo cáo tiến độ.

## 7.3 Bug Reports

Các lỗi (defects/bugs) được phát hiện trong quá trình kiểm thử sẽ được ghi nhận và theo dõi thông qua hệ thống **Jira hoặc Mantis**. Báo cáo lỗi sẽ bao gồm các thông tin: mô tả lỗi, bước tái hiện (steps to reproduce), môi trường xảy ra, mức độ nghiêm trọng (severity), mức độ ưu tiên (priority), trạng thái xử lý (open, in progress, fixed, closed).