**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***



**BÁO CÁO BÀI TẬP LỚN**

**HỌC PHẦN: KIỂM THỬ PHẦN MỀM**

Đề tài:

**TÌM HIỂU CÔNG CỤ KIỂM THỬ POSTMAN VÀ ỨNG DỤNG VÀO KIỂM THỬ API CHO WEBSITE BÁN ĐỒ ĐIỆN TỬ**

|  |  |  |
| --- | --- | --- |
| **GVHD:** | TS. Đỗ Ngọc Sơn | |
| **Nhóm:** | 4 |  |
| **Sinh viên:** | 1. Nguyễn Mạnh Hoàn | - MSV: 2022600679 |
|  | 2. Nguyễn Ngọc Hưng | - MSV: 2022601060 |
|  | 3. Lương Mạnh Hùng | - MSV: 2022600980 |
|  | 4. Hoàng Nguyễn Tùng Huy  5. Lê Toàn Thắng | - MSV: 2022601138  - MSV: 2022600284 |
| **Lớp:** | 20242IT6084001 |  |

*Hà Nội, năm 2025*

**LỜI CẢM ƠN**

Đầu tiên, chúng em xin gửi lời cảm ơn chân thành đến Trường Đại học Công nghiệp Hà Nội đã đưa môn học Kiểm thử phần mềm vào chương trình giảng dạy. Đặc biệt, chúng em xin gửi lời cảm ơn sâu sắc đến giảng viên bộ môn - TS. Đỗ Ngọc Sơn đã dạy dỗ, truyền đạt những kiến thức quý báu cho chúng em trong suốt thời gian học tập vừa qua. Trong thời gian tham gia lớp học của thầy, chúng em đã có thêm cho mình nhiều kiến thức bổ ích, tinh thần học tập hiệu quả, nghiêm túc. Đây chắc chắn sẽ là những kiến thức quý báu, là hành trang để chúng em có thể vững bước sau này.

Kiểm thử phần mềm là một môn học thú vị, vô cùng bổ ích và có tính thực tế cao. Đảm bảo cung cấp đủ kiến thức, gắn liền với nhu cầu thực tiễn của sinh viên. Tuy nhiên, do vốn kiến thức còn nhiều hạn chế và khả năng tiếp thu thực tế còn nhiều bỡ ngỡ. Mặc dù chúng em đã cố gắng hết sức nhưng chắc chắn bài báo cáo khó có thể tránh khỏi những thiếu sót và nhiều chỗ còn chưa chính xác, kính mong thầy xem xét và góp ý để bài báo cáo của chúng em được hoàn thiện hơn.

Chúng em xin chân thành cảm ơn!

**MỤC LỤC**

[DANH MỤC KÝ HIỆU, THUẬT NGỮ VIẾT TẮT 5](#_Toc199788241)

[DANH MỤC BẢNG BIỂU 6](#_Toc199788242)

[DANH MỤC HÌNH VẼ 7](#_Toc199788243)

[MỞ ĐẦU 10](#_Toc199788244)

[1. Lý do chọn đề tài 10](#_Toc199788245)

[2. Mục tiêu nghiên cứu 10](#_Toc199788246)

[3. Đối tượng và phạm vi nghiên cứu 11](#_Toc199788247)

[4. Bố cục của đề tài 11](#_Toc199788248)

[KẾT QUẢ NGHIÊN CỨU 12](#_Toc199788249)

[CHƯƠNG 1. TỔNG QUAN VỀ KIỂM THỬ PHẦN MỀM 12](#_Toc199788250)

[1.1. Giới thiệu về kiểm thử phần mềm 12](#_Toc199788251)

[1.1.1. Khái niệm kiểm thử phần mềm 12](#_Toc199788252)

[1.1.2. Mục tiêu của kiểm thử phần mềm 12](#_Toc199788253)

[1.1.3. Tầm quan trọng của kiểm thử phần mềm 12](#_Toc199788254)

[1.2. Các cấp độ kiểm thử phần mềm 13](#_Toc199788255)

[1.2.1. Kiểm thử đơn vị (Unit test) 13](#_Toc199788256)

[1.2.2. Kiểm thử tích hợp (Integration testing) 13](#_Toc199788257)

[1.2.3. Kiểm thử hệ thống (System testing) 14](#_Toc199788258)

[1.2.4. Kiểm thử chấp nhận (Acceptance testing) 15](#_Toc199788259)

[1.3. Các kỹ thuật kiểm thử phần mềm 15](#_Toc199788260)

[1.3.1. Kiểm thử hộp đen 15](#_Toc199788261)

[1.3.2. Kiểm thử hộp trắng 17](#_Toc199788262)

[CHƯƠNG 2. TÌM HIỂU VỀ CÔNG CỤ KIỂM THỬ POSTMAN 19](#_Toc199788263)

[2.1. Tổng quan về công cụ kiểm thử Postman 19](#_Toc199788264)

[2.1.1. Thông tin chung 19](#_Toc199788265)

[2.1.2. Ưu nhược điểm 19](#_Toc199788266)

[2.1.3. Giới thiệu về các thành phần và hoạt động của Postman 20](#_Toc199788267)

[2.1.3.1. Cấu trúc tổng quan: 20](#_Toc199788268)

[2.1.3.2. Cách hoạt động 21](#_Toc199788269)

[2.2. Cài đặt và sử dụng công cụ Postman 22](#_Toc199788270)

[2.2.1. Cài đặt công cụ Postman 22](#_Toc199788271)

[2.2.2. Sử dụng công cụ Postman 24](#_Toc199788272)

[CHƯƠNG 3. GIỚI THIỆU WEBSITE BÁN ĐỒ ĐIỆN TỬ 26](#_Toc199788273)

[3.1. Giới thiệu 26](#_Toc199788274)

[3.2. Đặc tả use case 27](#_Toc199788275)

[3.2.1. Mô tả use case Đăng ký 27](#_Toc199788276)

[3.2.2. Mô tả use case Đăng nhập 29](#_Toc199788277)

[3.2.3. Mô tả use case Tìm kiếm sản phẩm 31](#_Toc199788278)

[3.2.4. Mô tả use case Thanh toán 32](#_Toc199788279)

[3.2.5. Mô tả use case Cập nhật thông tin người dùng 34](#_Toc199788280)

[CHƯƠNG 4. KIỂM THỬ WEBSITE BÁN ĐỒ ĐIỆN TỬ 36](#_Toc199788281)

[4.1. Lập kế hoạch kiểm thử 36](#_Toc199788282)

[4.1.1. Giới thiệu 36](#_Toc199788283)

[4.1.1.1. Mục đích 36](#_Toc199788284)

[4.1.1.2. Tổng quan 36](#_Toc199788285)

[4.1.1.3. Phạm vi 36](#_Toc199788286)

[4.1.1.4. Các định nghĩa và các từ viết tắt được dùng 36](#_Toc199788287)

[4.1.1.5. Những người sử dụng tài liệu này 37](#_Toc199788288)

[4.1.2. Tài liệu tham khảo 37](#_Toc199788289)

[4.1.3. Lịch trình công việc 37](#_Toc199788290)

[4.1.4. Những yêu cầu về tài nguyên 38](#_Toc199788291)

[4.1.4.1. Phần cứng 38](#_Toc199788292)

[4.1.4.2. Phần mềm 38](#_Toc199788293)

[4.1.4.3. Công cụ kiểm thử 38](#_Toc199788294)

[4.1.4.4. Môi trường kiểm thử 38](#_Toc199788295)

[4.1.4.5. Nhân sự 39](#_Toc199788296)

[4.1.5. Phạm vi kiểm thử 39](#_Toc199788297)

[4.1.5.1. Những chức năng được kiểm thử 39](#_Toc199788298)

[4.1.5.2. Những chức năng không được kiểm thử 40](#_Toc199788299)

[4.1.6. Chiến lược kiểm thử 40](#_Toc199788300)

[4.1.6.1. Các giai đoạn kiểm thử 40](#_Toc199788301)

[4.1.6.2. Các loại kiểm thử 41](#_Toc199788302)

[4.1.7. Điều kiện chấp nhận 42](#_Toc199788303)

[4.1.8. Theo dõi lỗi 42](#_Toc199788304)

[4.1.8.1. Phân loại lỗi 42](#_Toc199788305)

[4.1.8.2. Quy trình xử lý lỗi 43](#_Toc199788306)

[4.2. Thực hiện kiểm thử 43](#_Toc199788307)

[4.2.1. Nguyễn Mạnh Hoàn - Đăng ký 43](#_Toc199788308)

[4.2.1.1. Phân tích thiết kế kiểm thử 43](#_Toc199788309)

[4.2.1.2. Thực hiện kiểm thử 49](#_Toc199788310)

[4.2.2. Nguyễn Ngọc Hưng - Đăng nhập 59](#_Toc199788311)

[4.2.2.1. Phân tích thiết kế kiểm thử 59](#_Toc199788312)

[4.2.2.2. Thực hiện kiểm thử 61](#_Toc199788313)

[4.2.3. Lương Mạnh Hùng - Tìm kiếm sản phẩm 66](#_Toc199788314)

[4.2.3.1. Phân tích thiết kế kiểm thử 66](#_Toc199788315)

[4.2.3.2. Thực hiện kiểm thử 68](#_Toc199788316)

[4.2.4. Hoàng Nguyễn Tùng Huy - Thanh toán 72](#_Toc199788317)

[4.2.4.1. Phân tích thiết kế kiểm thử 72](#_Toc199788318)

[4.2.4.2. Thực hiện kiểm thử 75](#_Toc199788319)

[4.2.5. Lê Toàn Thắng – Cập nhật thông tin người dùng 79](#_Toc199788320)

[4.2.5.1. Phân tích thiết kế kiểm thử 79](#_Toc199788321)

[4.2.5.2. Thực hiện kiểm thử 82](#_Toc199788322)

[KẾT LUẬN 88](#_Toc199788323)

[1. Kết quả đạt được 88](#_Toc199788324)

[2. Hướng phát triển 88](#_Toc199788325)

[TÀI LIỆU THAM KHẢO 89](#_Toc199788326)

# 

# DANH MỤC KÝ HIỆU, THUẬT NGỮ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| Thứ tự | Từ viết tắt | Mô tả |
| 1 | API | Application Programming Interface |
| 2 | SRS | Software Requirements Specification |
| 3 | URD | User Requirements Document |
| 4 | UAT | User Acceptance Testing |
| 5 | HTTP | HyperText Transfer Protocol |
| 6 | HTTPS | HyperText Transfer Protocol Secure |
| 7 | IEEE | Institute of Electrical and Electronics Engineers |
| 8 | UI | User Interface |
| 9 | CI/CD | Continuous Integration/Continuous Deployment |
| 10 | RAML | RESTful API Modeling Language |
| 11 | WSDL | Web Services Description Language |
| 12 | URL | Uniform Resource Locator |
| 13 | HTML | HyperText Markup Language |
| 14 | CSS | Cascading Style Sheets |
| 15 | EJS | Embedded JavaScript |
| 16 | CLI | Command Line Interface |

# DANH MỤC BẢNG BIỂU

[Bảng 4.1 Lịch trình công việc 37](#_Toc199721010)

[Bảng 4.2 Cấu hình tối thiểu của máy tính 38](#_Toc199721011)

[Bảng 4.3 Danh sách phần mềm phục vụ cho việc kiểm thử 38](#_Toc199721012)

[Bảng 4.4 Danh sách công cụ phục vụ cho quá trình kiểm thử 38](#_Toc199721013)

[Bảng 4.5 Danh sách nhân sự 39](#_Toc199721014)

[Bảng 4.6 Kiểm thử chức năng 41](#_Toc199721015)

[Bảng 4.7 Kiểm thử hiệu suất 41](#_Toc199721016)

[Bảng 4.8 Phân loại lỗi 42](#_Toc199721017)

[Bảng 4.9 *Bảng phân vùng tương đương (Chức năng đăng ký)* 44](#_Toc199721018)

[Bảng 4.10 Bảng thiết kế test case (Chức năng đăng ký) 45](#_Toc199721019)

[Bảng 4.11 *Bảng phân vùng tương đương (chức năng đăng nhập)* 59](#_Toc199721020)

[Bảng 4.12 *Bảng thiết kế test case (chức năng đăng nhập)* 60](#_Toc199721021)

[Bảng 4.13 *Bảng phân vùng tương đương (Chức năng tìm kiếm)* 67](#_Toc199721022)

[Bảng 4.14 *Bảng thiết kế test case (Chức năng tìm kiếm)* 67](#_Toc199721023)

[Bảng 4.15 Bảng phân vùng tương đương (Chức năng thanh toán) 73](#_Toc199721024)

[Bảng 4.16 *Bảng thiết kế test case (Chức năng thanh toán)* 73](#_Toc199721025)

[Bảng 4.17 Bảng phân vùng tương đương (Chức năng cập nhật tài khoản) 79](#_Toc199721026)

[Bảng 4.18 Bảng thiết kế test case (Chức năng cập nhật tài khoản) 80](#_Toc199721027)

# DANH MỤC HÌNH VẼ

[Hình 1.3.1 Kiểm thử hộp đen 16](#_Toc199720374)

[Hình 2.2.1 Giao diện của Postman 23](#_Toc199720375)

[Hình 2.2.2 Giao diện vùng làm việc của Postman 24](#_Toc199720376)

[Hình 3.2.1 Biểu đồ hoạt động của chức năng Đăng ký 27](#_Toc199720377)

[Hình 3.2.2 Biểu đồ hoạt động của chức năng Đăng nhập 29](#_Toc199720378)

[Hình 3.2.3 Biểu đồ hoạt động của chức năng Tìm kiếm sản phẩm 31](#_Toc199720379)

[Hình 3.2.4 Biểu đồ hoạt động của chức năng Thanh toán 32](#_Toc199720380)

[Hình 3.2.5 Biểu đồ hoạt động của chức năng Cập nhật thông tin người dùng 34](#_Toc199720381)

[Hình 4.2.1 Giao diện chức năng đăng ký 43](#_Toc199720382)

[Hình 4.2.2 Test case – Đăng ký 49](#_Toc199720383)

[Hình 4.2.3 Kịch bản kiểm thử đăng ký thành công 50](#_Toc199720384)

[Hình 4.2.4 Kịch bản kiểm thử họ tên để trống 50](#_Toc199720385)

[Hình 4.2.5 Kịch bản kiểm thử số điện thoại < 9 số 51](#_Toc199720386)

[Hình 4.2.6 Kịch bản kiểm thử số điện thoại > 11 số 51](#_Toc199720387)

[Hình 4.2.7 *Kịch bản kiểm thử số điện thoại chứa ký tự chữ* 52](#_Toc199720388)

[Hình 4.2.8 Kịch bản kiểm thử số điện thoại để trống 52](#_Toc199720389)

[Hình 4.2.9 Kịch bản kiểm thử số điện thoại đã tồn tại 53](#_Toc199720390)

[Hình 4.2.10 Kịch bản kiểm thử sai định dạng mail 54](#_Toc199720391)

[Hình 4.2.11 Kịch bản kiểm thử email để trống 54](#_Toc199720392)

[Hình 4.2.12 Kịch bản kiểm thử email đã tồn tạị 55](#_Toc199720393)

[Hình 4.2.13 Kịch bản kiểm thử mật khẩu < 6 ký tự 55](#_Toc199720394)

[Hình 4.2.14 Kịch bản kiểm thử mật khẩu để trống 56](#_Toc199720395)

[Hình 4.2.15 *Kịch bản kiểm thử mật khẩu không có ký tự đặc biệt* 56](#_Toc199720396)

[Hình 4.2.16 *Kịch bản kiểm thử mật khẩu không có ký tự chữ* 57](#_Toc199720397)

[Hình 4.2.17 *Kịch bản kiểm thử mật khẩu không có ký tự số* 57](#_Toc199720398)

[Hình 4.2.18 *: Kịch bản kiểm thử mật khẩu nhập lại không khớp* 58](#_Toc199720399)

[Hình 4.2.19 *Kịch bản kiểm thử không nhập lại mật khẩu* 58](#_Toc199720400)

[*Hình 4.2.20 Giao diện chức năng đăng nhập* 59](#_Toc199720401)

[Hình 4.2.21 Test case – Đăng nhập 61](#_Toc199720402)

[Hình 4.2.22 Kịch bản kiểm thử đăng nhập thành công 62](#_Toc199720403)

[Hình 4.2.23 Kịch bản kiểm thử tên đăng nhập không phải email 62](#_Toc199720404)

[Hình 4.2.24 Kịch bản kiểm thử bỏ trống tên đăng nhập 63](#_Toc199720405)

[Hình 4.2.25 Kịch bản kiểm thử mật khẩu < 6 ký tự 63](#_Toc199720406)

[Hình 4.2.26 Kịch bản kiểm thử để trống mật khẩu 64](#_Toc199720407)

[Hình 4.2.27 Kịch bản kiểm thử mật khẩu không có chữ cái 64](#_Toc199720408)

[Hình 4.2.28 Kịch bản kiểm thử mật khẩu không có số 65](#_Toc199720409)

[Hình 4.2.29 Kịch bản kiểm thử mật khẩu không có ký tự đặc biệt 65](#_Toc199720410)

[Hình 4.2.30 *Giao diện chức năng tìm kiếm* 66](#_Toc199720411)

[Hình 4.2.31 *Test case – Tìm kiếm* 68](#_Toc199720412)

[Hình 4.2.32 Kịch bản kiểm thử từ khóa hợp lệ, có kiểm thử 68](#_Toc199720413)

[Hình 4.2.33 *Kịch bản kiểm thử từ khóa hợp lệ, không có sản phẩm* 69](#_Toc199720414)

[Hình 4.2.34 *Kịch bản kiểm thử từ khóa hợp lệ, không dấu* 69](#_Toc199720415)

[Hình 4.2.35 *Kịch bản kiểm thử từ khóa hợp lệ, viết hoa thường đan xen* 70](#_Toc199720416)

[Hình 4.2.36 *Kịch bản kiểm thử không nhập từ khóa* 71](#_Toc199720417)

[Hình 4.2.37 *Kịch bản kiểm thử nhập quá dài* 71](#_Toc199720418)

[Hình 4.2.38 *Kịch bản kiểm thử có kí tự không hợp lệ* 72](#_Toc199720419)

[Hình 4.2.39 *Giao diện chức năng đăng ký* 72](#_Toc199720420)

[Hình 4.2.40 *Test case – Thanh toán* 75](#_Toc199720421)

[Hình 4.2.41 *Kịch bản kiểm thử thanh toán thành công* 75](#_Toc199720422)

[Hình 4.2.42 *Kịch bản kiểm thử số tiền để trống* 76](#_Toc199720423)

[Hình 4.2.43 Kịch bản kiểm thử số tiền nhỏ hơn 0,0001 76](#_Toc199720424)

[Hình 4.2.44 Kịch bản kiểm thử số tiền lớn hơn 10000000000 77](#_Toc199720425)

[Hình 4.2.45 Kịch bản kiểm thử số tiền là chuỗi ký tự 77](#_Toc199720426)

[Hình 4.2.46 *Kịch bản kiểm thử số tiền là số âm* 78](#_Toc199720427)

[Hình 4.2.47 *Kịch bản kiểm thử nội dung để trống* 78](#_Toc199720428)

[Hình 4.2.48 *Giao diện chức năng cập nhật tài khoản* 79](#_Toc199720429)

[Hình 4.2.49 *Kịch bản kiểm thử sửa thành công* 83](#_Toc199720430)

[Hình 4.2.50 *Kịch bản kiểm thử họ và tên để trống* 83](#_Toc199720431)

[Hình 4.2.51 *Kịch bản kiểm thử giới tính không phù hợp* 84](#_Toc199720432)

[Hình 4.2.52 Kịch bản kiểm thử lỗi khi cập nhật điện thoại đã dùng 84](#_Toc199720433)

[Hình 4.2.53 *Kịch bản kiểm thử ngày sinh không lớn hơn ngày hiện tại* 85](#_Toc199720434)

[Hình 4.2.54 *Kịch bản kiểm thử địa chỉ không được để trống* 85](#_Toc199720435)

[Hình 4.2.55 *Kịch bản kiểm thử số điện thoại có ít hơn 9* 86](#_Toc199720436)

[Hình 4.2.56 *Kịch bản kiểm thử số điện thoại chứa ký tự* 86](#_Toc199720437)

[Hình 4.2.57 Kịch bản kiểm thử số điện thoại phải từ 9-11 số, không chứa ký tự khác 87](#_Toc199720438)

# MỞ ĐẦU

## 1. Lý do chọn đề tài

Trong quá trình phát triển phần mềm, đặc biệt là các ứng dụng web hiện đại, API đóng vai trò quan trọng trong việc kết nối và trao đổi dữ liệu giữa các thành phần frontend và backend. Việc đảm bảo API hoạt động chính xác, ổn định và bảo mật là một yêu cầu thiết yếu trong quy trình kiểm thử. Postman là một trong những công cụ kiểm thử API phổ biến và mạnh mẽ nhất hiện nay, với giao diện trực quan, dễ sử dụng và tích hợp nhiều tính năng như kiểm thử tự động, mô phỏng server, quản lý môi trường và kiểm tra phản hồi.

Với đề tài "Tìm hiểu công cụ kiểm thử Postman và ứng dụng vào kiểm thử API cho website bán đồ điện tử", nhóm mong muốn tìm hiểu sâu hơn về cách sử dụng Postman trong thực tế, đồng thời áp dụng trực tiếp vào việc kiểm thử một hệ thống web cụ thể. Website bán đồ điện tử là một ứng dụng điển hình có nhiều chức năng liên quan đến API như: hiển thị danh sách sản phẩm, xử lý đăng nhập, đặt hàng, quản lý tài khoản… Do đó, việc kiểm thử API cho hệ thống này không chỉ giúp nâng cao chất lượng sản phẩm mà còn góp phần rèn luyện kỹ năng kiểm thử thực tế, đáp ứng nhu cầu học tập và ứng dụng nghề nghiệp sau này.

## 2. Mục tiêu nghiên cứu

Mục tiêu của đề tài là tìm hiểu và ứng dụng công cụ Postman trong quá trình kiểm thử API một cách hiệu quả. Trước tiên, đề tài hướng đến việc nghiên cứu tổng quan về Postman, bao gồm giao diện làm việc, các thành phần cơ bản như Collection, Request, Environment, cũng như cách sử dụng các tính năng như Pre-request Script, Test Script, và kiểm thử tự động. Qua đó, người thực hiện sẽ nắm bắt được cách thức hoạt động và quy trình kiểm thử API bằng Postman một cách bài bản.

Tiếp theo, đề tài tập trung vào việc áp dụng Postman để kiểm thử các chức năng API của một website bán đồ điện tử. Cụ thể, các chức năng như đăng ký tài khoản, đăng nhập, hiển thị sản phẩm, thêm vào giỏ hàng, đặt hàng và quản lý người dùng sẽ được kiểm thử thông qua Postman. Việc này giúp minh họa rõ ràng cách Postman được sử dụng trong thực tế, đồng thời làm nổi bật vai trò của kiểm thử API trong một hệ thống web hiện đại.

Ngoài ra, đề tài còn hướng đến mục tiêu phân tích hiệu quả của việc sử dụng Postman trong kiểm thử phần mềm, như khả năng phát hiện lỗi, tính tự động hóa và hỗ trợ kiểm thử hồi quy. Cuối cùng, quá trình thực hiện đề tài cũng là cơ hội để người học rèn luyện kỹ năng kiểm thử API, nâng cao năng lực chuyên môn và chuẩn bị tốt hơn cho công việc trong lĩnh vực phát triển phần mềm sau này.

## 3. Đối tượng và phạm vi nghiên cứu

* Đối tượng nghiên cứu:

Đối tượng nghiên cứu của đề tài là công cụ kiểm thử Postman và các API được sử dụng trong website bán đồ điện tử. Trong đó, Postman được xem xét dưới góc độ là một công cụ hỗ trợ kiểm thử phần mềm với các chức năng như gửi request, kiểm tra phản hồi, viết script kiểm thử và chạy kiểm thử tự động. Các API trong hệ thống bán hàng điện tử bao gồm các chức năng phổ biến như đăng ký, đăng nhập, xử lý sản phẩm, giỏ hàng, đặt hàng và quản lý người dùng, là các đối tượng trực tiếp được đưa vào quy trình kiểm thử.

* Phạm vi nghiên cứu:

Phạm vi nghiên cứu của đề tài tập trung vào việc kiểm thử **API chức năng** của một website bán đồ điện tử thông qua công cụ Postman. Đề tài không đi sâu vào kiểm thử hiệu năng (Performance Testing), bảo mật (Security Testing) hay kiểm thử giao diện người dùng (UI Testing). Hệ thống kiểm thử được xây dựng trên mô hình đơn giản gồm client giao diện người dùng, server xử lý backend bằng Node.js và cơ sở dữ liệu MySQL. Các kiểm thử được thực hiện trong môi trường giả lập hoặc môi trường cục bộ (localhost), không triển khai trên môi trường thực tế hoặc tích hợp CI/CD. Ngoài ra, đề tài chỉ sử dụng các tính năng có sẵn trong Postman, không tích hợp thêm công cụ kiểm thử bên ngoài như Newman trong pipeline DevOps.lai.

## 4. Bố cục của đề tài

Bên cạnh phần Mở đầu và Kết luận, quyển báo cáo bài tập lớn được xây dựng với 4 nội dung chính tương ứng với 4chương:

* Chương 1: Tổng quan về kiểm thử phần mềm
* Chương 2: Tìm hiểu về công cụ kiểm thử Postman
* Chương 3: Giới thiệu website bán đồ điện tử
* Chương 4: Kiểm thử website bán đồ điện tử

# KẾT QUẢ NGHIÊN CỨU

# CHƯƠNG 1. TỔNG QUAN VỀ KIỂM THỬ PHẦN MỀM

## Giới thiệu về kiểm thử phần mềm

### Khái niệm kiểm thử phần mềm

Kiểm thử phần mềm là một quá trình kiểm tra được tiến hành nhằm cung cấp thông tin cho các bên liên quan về chất lượng của sản phẩm hoặc dịch vụ phần mềm đang được kiểm thử. Hiểu theo cách đơn giản hơn, kiểm thử phần mềm là quá trình tìm ra thất bại (failure) hoặc chứng minh rằng phần mềm hoạt động đúng như mong đợi.

Theo Glenford Myers: Kiểm thử là quá trình vận hành chương trình để tìm ra lỗi.

Theo IEEE, Kiểm thử là:

* Là quá trình vận hành hệ thống hoặc thành phần dưới những điều kiện xác định, quan sát hoặc ghi nhận kết quả và đưa ra đánh giá về hệ thống hoặc thành phần đó.
* Là quá trình phân tích phần mềm để tìm ra sự khác biệt giữa điều kiện thực tế và điều kiện yêu cầu và dựa vào điểm khác biệt đó để đánh giá tính năng phần mềm

### Mục tiêu của kiểm thử phần mềm

Kiểm thử phần mềm là một bước quan trọng trong quy trình phát triển nhằm đảm bảo chất lượng sản phẩm trước khi đưa vào sử dụng. Mục tiêu chính của quá trình này bao gồm:

* Phát hiện càng nhiều lỗi phần mềm càng tốt trong phạm vi thời gian và nguồn lực cho phép.
* Đảm bảo rằng sản phẩm phần mềm đáp ứng đầy đủ các yêu cầu đã được đặc tả.
* Xác thực chất lượng của phần mềm, đảm bảo việc kiểm thử được thực hiện với chi phí và công sức tối thiểu nhưng vẫn đạt hiệu quả mong muốn.
* Thiết kế và thực hiện các tài liệu kiểm thử một cách có hệ thống, nhằm nâng cao hiệu quả kiểm thử và tiết kiệm thời gian cũng như công sức.

### Tầm quan trọng của kiểm thử phần mềm

Kiểm thử phần mềm đóng vai trò mấu chốt trong việc đảm bảo chất lượng, độ tin cậy và hiệu suất của sản phẩm phần mềm trước khi được đưa vào triển khai thực tế. Vì vậy nó không chỉ giúp phát hiện lỗi mà còn góp phần xác nhận rằng phần mềm đáp ứng đúng các yêu cầu kỹ thuật và nghiệp vụ một cách đầy đủ và chính xác. Một số điểm quan trọng cần chú ý của kiểm thử phần mềm bao gồm:

* Kiểm thử nhằm chứng minh rằng chương trình không có lỗi.
* Kiểm thử để xác nhận rằng chương trình đã thực hiện đúng các chức năng theo yêu cầu đặc tả.
* Kiểm thử giúp phát hiện lỗi và sửa chữa kịp thời, từ đó nâng cao độ tin cậy và chất lượng của phần mềm.

## Các cấp độ kiểm thử phần mềm

### Kiểm thử đơn vị (Unit test)

* Mục tiêu:

Unit test được áp dụng ở mức từng module. Các lập trình viên tự thực hiện công việc này trước khi tích hợp các modules với nhau.

Unit test đại diện cho kiểm thử mức thành phần nhỏ nhất, và thường được thực hiện bằng phương pháp kiểm thử hộp trắng.

Thông thường, các lỗi phát hiện trong quá trình Unit test chiếm trên 25% tổng số lỗi của dự án.

* Lập kế hoạch

Lập kế hoạch unit test: Unit test cần phải được lập kế hoạch cùng với các hoa động khác (như system test, integration test) trong tài liệu Test plan và trong schedule của dự án.

Yêu cầu kiểm thử: xác định các thuộc tính, phương thức cần kiểm thử.

Phương pháp thiết kế: kỹ thuật kiểm thử (kiểm thử hộp đen/kiểm thử hộp trắng); các công cụ được sử dụng trong unit test.

Thực thi unit test: để kiểm thử một đơn vị mã nguồn (một module, một lớp hay một hàm), chúng ta thường phải tạo các stubs và các drivers.

* Unit test case

Đội phát triển phải tạo unit test case để đảm bảo mỗi trạng thái, mỗi nhánh quyết định hoặc mỗi đường dẫn được kiểm thử ít nhất 1 lần.

### Kiểm thử tích hợp (Integration testing)

* Mục tiêu

Integration test là mức tiếp theo của unit test, do đội kiểm thử thực hiện. Integration test tập trung vào việc kiểm thử sự tích hợp các “đơn vị mã nguồn”, hay các thành phần (components).

Điều quan trọng là phải thực hiện unit test trước khi thực hiện Integration test.

Integration test kiểm tra xem các giao diện bên trong và bên ngoài hệ thống làm việc có đúng không.

* Lập kế hoạch

Kế hoạch Integration test có thể được đề cập đến trong tài liệu Test plan hoặc tài liệu Integration test plan riêng, do đội kiểm thử hoặc đội phát triển làm.

Với các dự án nhỏ và các hệ thống đơn giản, các giao diện bên trong và bên ngoài có thể được xác minh ngay trong giai đoạn unit test.

Với các dự án lớn và các hệ thống phức tạp, Integration test cần phải được đội phát triển lên kế hoạch riêng và thực thi khi “lắp ghép” các modules thành sản phẩm. Kỹ thuật Integration test dựa trên việc chọn thứ tự tích hợp sản phẩm. Integration test là một phần của hoạt động tích hợp sản phẩm. Integration test case có thể được xây dựng độc lập hoặc có trong kế hoạch tích hợp sản phẩm.

Giao diện của sản phẩm với các hệ thống ngoài có thể được xác minh bởi đội kiểm thử trong giai đoạn system test. Khi đó, các giao diện bên ngoài phải được mô tả chính xác trong URD hoặc SRS. Integration test case có thể có trong system test case.

* Integration test case

Integration tesst cases tập trung vào luồng dữ liệu/thông tin/điều khiển từ thành phần này đến thành phần khác. Do đó, Integration test cases cần phải chú trọng vào các scenarios mà thành phần này gọi đến thành phần khác.

### Kiểm thử hệ thống (System testing)

* Mục tiêu

System test xác minh các chức năng hệ thống đã đáp ứng được yêu cầu hệ thống chưa.

System test thường được thực hiện sau unit test và integration test thành công cho tất cả các thành phần.

Kỹ thuật system test phổ biến là kiểm thử hộp đen và do đội kiểm thử thực thi.

* Lập kế hoạch

Kế hoạch System test thường được nêu trong tài liệu Test plan.

Test plan xác định cấu trúc kiểm thử dự án, bao gồm các tổ chức tham gia, mối quan hệ với tổ chức khác; thẩm quyền và trách nhiệm của mỗi tổ chức.

Test plan cần chỉ rõ cái gì được kiểm thử, các hoạt động, tài nguyên, và lên kế hoạch cho các tiến trình cần thiết. Các cớ sở vật chất được sử dụng, các sản phẩm và các milestone trong quá trình kiểm thử phải được mô tả trong tài liệu này.

* System test case

System test case được tạo ra để xác minh ứng dụng có đáp ứng được những yêu cầu đã xác định trong tài liệu đặc tả chưa.

System test case cần mô tả chi tiết các hành vi hoặc các chức năng cần được kiểm thử, dữ liệu đầu vào thực tế và dữ liệu đầu ra mong muốn. Đầu ra mong muốn phải được nêu chi tiết đủ để thực thi. Dữ liệu cần chạy với phần mềm thực tế sẽ được bàn giao.

### Kiểm thử chấp nhận (Acceptance testing)

* Mục tiêu

Acceptance test chứng minh cho khách hàng thấy hệ thống đã đạt được những mục tiêu đề ra.

* Lập lế hoạch

Hướng dẫn khách hàng sử dụng sản phẩm dưới một số hình thức sau:

* Chạy sản phầm, và hướng dẫn bằng cách thực hiện trực tiếp trên sản phẩm.
* Tạo slide và trình chiếu để hướng dẫn khách hàng trực tiếp.
* Viết UAT test case cho khách hàng thực hiện.

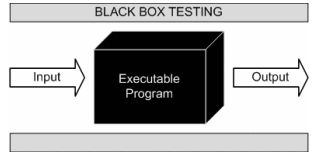
Tạo UAT Test Case hoặc tiêu chí nghiệm thu.

Khách hàng thực hiện dùng thử sản phẩm trên môi trường thật.

## Các kỹ thuật kiểm thử phần mềm

### Kiểm thử hộp đen

Kiểm thử hộp đen (Black box testing) là một phương pháp kiểm thử phần mềm mà việc kiểm tra các chức năng của một ứng dụng không cần quan tâm vào cấu trúc nội bộ hoặc hoạt động của nó. Mục đích chính của kiểm thử hộp đen chỉ là để xem phần mềm có hoạt động như dự kiến trong tài liệu yêu cầu và liệu nó có đáp ứng được sự mong đợi của người dùng hay không.



Hình 1.3.1 Kiểm thử hộp đen

* Đặc điểm:

Đây là kiểu kiểm thử thành phần phần mềm và chỉ dựa vào các thông tin đặc tả về yêu cầu, chức năng của các thành phần phần mềm tương ứng.

Việc kiểm thử được thực hiện bên ngoài, không liên quan đến lập trình viên hay các nhà phát triển phần mềm. Vì thế người kiểm thử cũng không cần thiết phải biết về cấu trúc bên trong của phần mềm cũng như các kiến thức về lập trình.

Mức test này thường yêu cầu các tester phải viết test case đầy đủ trước khi test. Các bước tiến hành test khá đơn giản, chỉ cần thực hiện theo các mô tả trong test case, thực hiện nhập dữ liệu vào, đợi kết quả trả về và so sánh với kết quả dự kiến trong test case.

* Có 8 kỹ thuật phổ biến:
* Phân vùng tương đương (Equivalence partitioning).
* Phân tích giá trị biên (Boundary value analysis).
* Bảng quyết định (Decision tables)
* Kiểm thử chuyển đổi trạng thái (State transition testing).
* Kỹ thuật kiểm thử các bộ n thần kỳ (Pairwise).
* Kỹ thuật phân tích vùng miền (Domain analysis).
* Kỹ thuật dựa trên đặc tả Use case (Use case).
* Kỹ thuật dùng lược đồ quan hệ nhân quả (Cause- Effect Diagram).
* Ưu điểm

Kỹ sư kiểm thử có thể không phải IT chuyên nghiệp.

Hệ thống thật sự với toàn bộ yêu cầu của nó được kiểm thử chính xác.

Thiết kế kịch bản kiểm thử khá nhanh, ngay khi mà các yêu cầu chức năng được xác định.

* Nhược điểm

Dữ liệu đầu vào yêu cầu một khối lượng mẫu (sample) khá lớn.

Khó viết kịch bản kiểm thử do cần xác định tất cả các yếu tố đầu vào, và thiếu cả thời gian cho việc tập hợp này.

Khả năng để bản thân kỹ sư lạc lối trong khi kiểm thử là khá cao.

### Kiểm thử hộp trắng

Kiểm thử hộp trắng là loại thử nghiệm được thực hiện để kiểm tra cấu trúc code. Loại thử nghiệm này đòi hỏi người test phải có kiến thức về code. Do đó, phần lớn là do các lập trình viên, nhà phát triển phần mềm thực hiện.

* Đặc điểm

Kiểm thử hộp trắng quan tâm đến việc hệ thống vận hành như thế nào chứ không phải chức năng của hệ thống. Vì nó dựa vào những thuật toán cụ thể, vào những cấu trúc dữ liệu bên trong của thành phần phần mềm.

Trong kỹ thuật kiểm thử này, đòi hỏi người tester phải có kiến thức và kỹ năng nhất định về ngôn ngữ lập trình được dùng, hiểu thuật toán trong phần mềm, để có thể hiểu được chi tiết về đoạn code cần kiểm thử.

Mức test này thường yêu cầu các tester phải viết test case đầy đủ các nhánh trong code, khi test sẽ đặt điều kiện và dữ liệu để chạy vào đủ tất cả các nhánh trong thuật toán, đảm bảo thực hiện đầy đủ.

Kiểm thử hộp trắng được dùng để đo độ bao phủ (coverage) và kiểm thử thiết kế (design test).

Bao phủ kiểm thử (test coverage) là tỷ lệ (tính theo %) testcase đã được thực hiện trên tổng số test case cần thiết cho phần mềm. Nếu tỉ lệ này càng cao thì phần mềm càng được test kỹ. Mặc dù việc đảm bảo phần mềm có test coverage là 100% nhưng không có nghĩa là 100% các trường hợp được kiểm thử.

* Ưu điểm

Test có thể bắt đầu ở giai đoạn sớm hơn, không cần phải chờ đợi cho GUI để có thể test.

Test kỹ càng hơn, có thể bao phủ hầu hết các đường dẫn.

Thích hợp trong việc tìm kiếm lỗi và các vấn đề trong mã lệnh.

Cho phép tìm kiếm các lỗi ẩn bên trong.

Các lập trình viên có thể tự kiểm tra.

Giúp tối ưu việc mã hoá.

Do yêu cầu kiến thức cấu trúc bên trong của phần mềm, nên việc kiểm soát lỗi tối đa nhất.

* Nhược điểm

Vì các bài kiểm tra rất phức tạp, đòi hỏi phải có các nguồn lực có tay nghề cao, với kiến thức sâu rộng về lập trình và thực hiện.

Maintenance test script có thể là một gánh nặng nếu thể hiện thay đổi quá thường xuyên.

Vì phương pháp thử nghiệm này liên quan chặt chẽ với ứng dụng đang được test, nên các công cụ để phục vụ cho mọi loại triển khai/ nền tảng có thể không sẵn có.

# TÌM HIỂU VỀ CÔNG CỤ KIỂM THỬ POSTMAN

## Tổng quan về công cụ kiểm thử Postman

### Thông tin chung

**Postman** là một công cụ mạnh mẽ để kiểm thử API (Application Programming Interface) được sử dụng rộng rãi trong phát triển phần mềm. Nó cho phép các nhà phát triển và kiểm thử viên gửi các yêu cầu HTTP tới các API và phân tích các phản hồi mà không cần viết mã.

Postman là một công cụ không thể thiếu cho các nhà phát triển và kiểm thử viên trong việc kiểm tra và đảm bảo chất lượng của các API.

### Ưu nhược điểm

* Ưu điểm:

- Giao diện người dùng thân thiện, dễ sử dụng cho cả người mới và lập trình viên có kinh nghiệm.

- Hỗ trợ đầy đủ các loại HTTP request như GET, POST, PUT, DELETE... cùng khả năng cấu hình chi tiết Header, Params, Body.

- Cho phép quản lý request theo Collection giúp tổ chức công việc rõ ràng hơn.

- Có thể tạo các môi trường (Environment) khác nhau với biến như {{base\_url}}, thuận tiện khi làm việc với nhiều môi trường (dev, staging, production).

- Hỗ trợ viết script kiểm thử bằng JavaScript để kiểm tra phản hồi API tự động.

- Cho phép chạy hàng loạt request thông qua Collection Runner để kiểm thử toàn bộ hệ thống API.

- Có tính năng tạo Mock Server để mô phỏng phản hồi từ API ngay cả khi backend chưa sẵn sàng.

- Hỗ trợ tạo Monitor theo dõi sức khỏe API định kỳ, gửi cảnh báo nếu có lỗi.

- Tích hợp tốt với CI/CD thông qua công cụ dòng lệnh Newman.

- Dễ dàng chia sẻ Collection, Environment, Mock Server thông qua link hoặc làm việc nhóm trong Workspace.

- Hỗ trợ import/export từ định dạng phổ biến như Swagger, OpenAPI, RAML, WSDL.

- Chạy được trên nhiều nền tảng: Windows, macOS, Linux và cả phiên bản web (Postman Web).

* Nhược điểm:

- Ứng dụng có thể bị chậm hoặc treo khi làm việc với số lượng lớn request hoặc response lớn.

- Một số tính năng nâng cao như quản lý team, giới hạn số lần gọi API, monitor nâng cao chỉ có trong bản trả phí.

- Không phù hợp cho kiểm thử UI hoặc kiểm thử tự động phức tạp, chỉ tập trung vào API.

- Script test bị giới hạn trong JavaScript, không hỗ trợ tốt cho test case phức tạp như các framework chuyên nghiệp (JUnit, Pytest...).

- Không hỗ trợ chạy kiểm thử song song (parallel testing), làm giảm hiệu suất với số lượng test lớn.

- Chưa có khả năng tích hợp trực tiếp với Git để kiểm soát version của Collection.

- Một số tính năng như monitor, chia sẻ workspace hoặc sync yêu cầu kết nối internet.

### Giới thiệu về các thành phần và hoạt động của Postman

#### Cấu trúc tổng quan:

**Workspace**: Môi trường làm việc, nơi người dùng tổ chức các collection, môi trường, request, test script... Có thể là cá nhân hoặc chia sẻ với nhóm.

**Collection**: Tập hợp các request API được tổ chức logic theo từng nhóm chức năng (ví dụ: User API, Product API). Có thể chứa folder con, script dùng chung và tài liệu mô tả.

**Request**: Một yêu cầu gửi đến hệ thống đích, gồm các thành phần như:

* Phương thức HTTP (GET, POST, PUT, DELETE,...),
* URL endpoint,
* Params, Headers, Body, Auth…

**Environment**: Tập hợp các biến (variable) dùng trong request, như {{base\_url}}, {{token}}, giúp dễ dàng chuyển đổi giữa môi trường Dev, Test, Prod.

**Pre-request Script**: Mã JavaScript chạy **trước khi gửi request**, dùng để thiết lập biến, tạo token, hoặc chuẩn bị dữ liệu.

**Tests**: Mã JavaScript chạy **sau khi nhận phản hồi**, dùng để kiểm tra mã phản hồi, nội dung dữ liệu, lưu giá trị vào biến toàn cục...

**Assertions**: Các đoạn kiểm thử được viết trong phần **Tests**, để xác nhận rằng phản hồi API đúng như mong đợi (ví dụ: pm.response.code === 200).

**Variables**: Biến có thể khai báo ở nhiều cấp độ: Global, Environment, Collection, Local, Data… Dùng để tái sử dụng dữ liệu trong request.

**Collection Runner**: Công cụ chạy tự động nhiều request trong một collection, có thể lặp lại và sử dụng dữ liệu từ file JSON hoặc CSV.

**Monitor**: Dịch vụ giám sát các collection theo lịch định sẵn (hằng giờ, hằng ngày), phát hiện lỗi và gửi cảnh báo.

**Mock Server**: Dùng để mô phỏng phản hồi từ API chưa có thật, hỗ trợ phát triển frontend song song với backend.

**Documentation**: Tài liệu API được tạo tự động từ collection, có thể chỉnh sửa và chia sẻ với người khác thông qua link.

**Newman**: Công cụ dòng lệnh cho phép chạy collection bên ngoài Postman, Hoạt động

#### Cách hoạt động

1. Khởi tạo và tạo request

Người dùng mở Postman, tạo mới hoặc chọn một request trong Collection. Cấu hình phương thức HTTP (GET, POST, PUT...), URL endpoint, thêm các tham số (Params), tiêu đề (Headers), và dữ liệu (Body) nếu cần.

1. Thực thi Pre-request Script

Trước khi gửi request, Postman chạy các đoạn mã JavaScript trong phần Pre-request Script (nếu có). Mục đích để chuẩn bị dữ liệu, tạo token, hoặc set biến môi trường cần thiết cho request.

1. Gửi request đến Server

Postman gửi yêu cầu HTTP đã cấu hình đến server đích thông qua mạng Internet.

1. Nhận phản hồi từ Server

Server trả về phản hồi (response), bao gồm mã trạng thái (status code), dữ liệu body, các header, cookie...

1. Thực thi Tests Script

Postman chạy đoạn mã JavaScript trong tab Tests để kiểm tra tính hợp lệ của phản hồi (ví dụ: status code, dữ liệu JSON, thời gian phản hồi). Các kết quả kiểm thử được hiển thị ở tab Test Results.

1. Hiển thị kết quả

Kết quả phản hồi được hiển thị trực quan trong phần Response Viewer, gồm nội dung body, mã trạng thái, thời gian và kích thước.

1. Lưu và tổ chức request

Người dùng có thể lưu request và kết quả vào Collection để tái sử dụng hoặc chia sẻ với nhóm.

1. Chạy nhiều request (tuỳ chọn)

Người dùng có thể dùng Collection Runner hoặc Newman để chạy tuần tự nhiều request, với dữ liệu đầu vào hoặc kịch bản kiểm thử phức tạp.

1. Giám sát và tự động hóa (tuỳ chọn)

Sử dụng Monitor để tự động chạy kiểm thử theo lịch và nhận cảnh báo khi có lỗi.

## Cài đặt và sử dụng công cụ Postman

### Cài đặt công cụ Postman

* + **Bước 1: Tải xuống Postman:** Mở trình duyệt (Edge hoặc Chorme) và nhập URL [http://postman.com/](http://seleniumhq.org/). Nó sẽ mở ra trang chủ của Postman.
  + **Bước 2:** Trong phần “Download the desktop app for”, chọn hệ điều hành tương ứng với thiết bị

A screenshot of a computer

AI-generated content may be incorrect.

Trang này bao gồm tất cả các bản phát hành mới nhất của tất cả các bản Postman.

* + **Bước 3:** Chọn mục Windows 64-bit để thực hiện tải xuống Postman

A screenshot of a computer

AI-generated content may be incorrect.

* Hình dưới là giao diện của Postman

A screenshot of a computer

AI-generated content may be incorrect.

Hình 2.2.1 Giao diện của Postman

### Sử dụng công cụ Postman

- Bước 1: Mở công cụ Postman

Dưới đây là giao diện vùng làm việc của Postman:

A screenshot of a computer

AI-generated content may be incorrect.

Hình 2.2.2 Giao diện vùng làm việc của Postman

- Bước 2: Tạo request đầu tiên (GET)

1. Nhấn nút + New Tab

2. Chọn phương thức GET từ dropdown

3. Nhập URL sau vào ô URL: <https://jsonplaceholder.typicode.com/posts>

4. Nhấn Send (Góc trên bên phải)

5. Danh sách kết quả trả về (dạng JSON) sẽ hiển thị ở khung kết quả trả về

- Bước 3: Tạo yêu cầu POST (Tạo dữ liệu)

1. Mở 1 tab mới

2. Chọn phương thức POST

3. Nhập URL: <https://jsonplaceholder.typicode.com/posts>

4. Chuyển sang tab body -> Chọn raw -> Chọn kiểu dữ liệu JSON

5. Dán nội dung sau vào ô nhập:

A black screen with green and blue text

AI-generated content may be incorrect.

6. Nhấn Send

7. Kết quả sẽ nhận được 1 object chứa id mới được tạo

- Bước 4: Viết kiểm thử đơn giản (Test Script)

1. Vẫn ở tab Post vừa dùng

2. Chuyển sang tab Tests

3. Dán đoạn mã sau:

A black screen with white text

AI-generated content may be incorrect.

4. Nhấn Send

5. Nếu response có status code 201 -> test sẽ PASS

- Bước 5: Lưu lại và tổ chức requests

1. Nhấn nút Save (Góc phải)

2. Tạo mới 1 Collection (Ví dụ: “API\_Test”)

3. Lưu lại các request vừa test vào đó để dùng lại sau này

# GIỚI THIỆU WEBSITE BÁN ĐỒ ĐIỆN TỬ

## Giới thiệu

Website bán đồ điện tử được xây dựng nhằm mang đến trải nghiệm mua sắm trực tuyến tiện lợi và hiện đại cho người dùng. Hệ thống cho phép khách hàng dễ dàng duyệt qua các sản phẩm điện tử đa dạng, từ điện thoại, laptop đến thiết bị gia dụng, với thông tin chi tiết, hình ảnh sinh động và giá cả minh bạch. Người dùng có thể đăng ký tài khoản, thêm sản phẩm vào giỏ hàng, thanh toán trực tuyến và theo dõi đơn hàng một cách nhanh chóng và thuận tiện.

Website được phát triển sử dụng các công nghệ hiện đại như HTML, CSS và JavaScript để tạo giao diện thân thiện, EJS giúp xây dựng các trang động hiệu quả, Node.js đảm nhận xử lý phía server với tốc độ nhanh và linh hoạt, cùng MySQL quản lý cơ sở dữ liệu sản phẩm, khách hàng, đơn hàng một cách chính xác và an toàn. Thiết kế responsive giúp website hoạt động mượt mà trên mọi thiết bị, từ máy tính bàn đến điện thoại di động, đáp ứng tốt nhu cầu mua sắm điện tử của khách hàng hiện đại.

## Đặc tả use case

### Mô tả use case Đăng ký

A diagram of a chemical process

Description automatically generated

Hình 3.2.1 Biểu đồ hoạt động của chức năng Đăng ký

|  |
| --- |
| 1. Tên Use Case: Đăng ký  2. Mô tả vắn tắt  Use case cho phép người dùng tạo tài khoản mới.  3. Luồng các sự kiện  3.1 Luồng cơ bản   1. Use case bắt đầu khi người dùng kích vào nút “Đăng ký”. 2. Hệ thống hiển thị form đăng ký yêu cầu nhập thông tin: email, mật khẩu, và xác nhận mật khẩu. 3. Người dùng nhập thông tin và nhấn nút “Tiếp theo”. 4. Hệ thống kiểm tra tính hợp lệ của dữ liệu (ví dụ: email hợp lệ, mật khẩu phải có ít nhất 6 ký tự). 5. Hệ thống lưu thông tin người dùng vào bảng CUSTOMERS và gửi email xác nhận đăng ký. 6. Người dùng nhận được thông báo xác nhận qua email. Use case kết thúc.   3.2. Luồng rẽ nhánh   1. Tại bước 4, nếu dữ liệu không hợp lệ, hệ thống sẽ hiển thị thông báo lỗi 2. Nếu hệ thông không kết nối được với cơ sở dữ liệu, hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc   4. Yêu cầu đặc biệt  Không có.  5. Tiền điều kiện  Không có  6. Hậu điều kiện  Nếu use case thực hiện thành công, hệ thống thêm bản ghi mới vào bảng CUSTOMERS trong cơ sở dữ liệu.  7. Điểm mở rộng  Không có. |

### Mô tả use case Đăng nhập

A diagram of a flowchart

Description automatically generated

Hình 3.2.2 Biểu đồ hoạt động của chức năng Đăng nhập

|  |
| --- |
| 1. Tên Use Case: Đăng nhập  2. Mô tả vắn tắt  Use case này cho phép người dùng đã có tài khoản đăng nhập hệ thống.  3. Luồng các sự kiện  3.1 Luồng cơ bản   1. Use case bắt đầu khi người dùng chọn trang đăng nhập phù hợp.    1. Khách hàng kích nút “Đăng nhập” trên giao diện trang chủ.    2. Admin kích nút “Đăng nhập” trên giao diện admin. 2. Hệ thống hiển thị form tương ứng cho từng loại người dùng:    1. Form đăng nhập khách hàng yêu câu nhập thông tin: email và mật khẩu.    2. Form đăng nhập admin yêu cầu nhập thông tin: email và mật khẩu. 3. Người dùng nhập thông tin vào form tương ứng và nhấn nút “Đăng nhập”. 4. Hệ thống kiểm tra thông tin đăng nhập:    1. Với khách hàng, hệ thống kiểm tra từ bảng CUSTOMERS.    2. Với admin, hệ thống kiểm tra từ bảng STAFFS 5. Người dùng nhận được thông báo đăng nhập thành công và chuyển hướng tới trang chính tương ứng:    1. Khách hàng được chuyển tới trang giao diện dành cho khách hàng.    2. Admin được chuyển hướng tới trang quản trị   Use case kết thúc  3.2. Luồng rẽ nhánh   1. Tại bước 4, nếu dữ liệu không hợp lệ, hệ thống sẽ hiển thị thông báo lỗi 2. Nếu hệ thống không kết nối được với cơ sở dữ liệu, hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc   4. Yêu cầu đặc biệt.  Email phải là duy nhất trong từng bảng (CUSTOMERS và STAFFS)  5. Tiền điều kiện.  Người dùng chọn đúng trang đăng nhập tương ứng với quyền truy cập của họ (khách hàng hoặc admin).  6. Hậu điều kiện  Không có  7. Điểm mở rộng  Không có. |

### Mô tả use case Tìm kiếm sản phẩm

A screenshot of a computer

Description automatically generated

Hình 3.2.3 Biểu đồ hoạt động của chức năng Tìm kiếm sản phẩm

|  |
| --- |
| 1. Tên Use Case: Tìm kiếm sản phẩm  2. Mô tả vắn tắt  Use case cho phép khách hàng tìm kiếm sản phẩm có trong hệ thống bán hàng online của cửa hàng.  3. Luồng các sự kiện  3.1 Luồng cơ bản  1) Use case này bắt đầu khi khách hàng nhập tên/loại sản phẩm muốn tra cứu vào ô tìm kiếm. Hệ thống lấy thông tin từ bảng PRODUCTS và hiển thị: ảnh minh họa, tiêu đề, giá bán của các lo  2) Khách hàng kích vào ảnh minh họa một loại sản phẩm trong danh sách. Hệ thống lấy thông tin chi tiết của sản phẩm được chọn gồm: ảnh minh họa, tên sản phẩm, mã sản phẩm, giá bán, số lượng, thông tin sản phẩm, số sao đánh giá về sản phẩm từ bảng PRODUCTS và hiển thị ra màn hình. Use case kết thúc.  3.2. Luồng rẽ nhánh   1. Tại bước 1 trong luồng cơ bản khi hệ thống không tìm thấy tên/loại sản phẩm nào thuộc danh mục khách hàng muốn tìm kiếm, thì sẽ hiển thị một thông báo “Không tìm thấy sản phẩm”. Use case kết thúc. 2. Tại thời điểm bất kì trong quá trình thực hiện use case nếu không kết nối được với hệ thống cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.   4. Yêu cầu đặc biệt  Không có  5. Tiền điều kiện  Không có  6. Hậu điều kiện  Không có  7. Điểm mở rộng  Không có. |

### Mô tả use case Thanh toán

A diagram of a flowchart

Description automatically generated

Hình 3.2.4 Biểu đồ hoạt động của chức năng Thanh toán

|  |
| --- |
| 1. Tên Use Case: Thanh toán  2. Mô tả vắn tắt  Use case cho phép người dùng thanh toán giỏ hàng và mua sản phẩm.  3. Luồng các sự kiện  3.1 Luồng cơ bản   1. Use case bắt đầu khi người dùng nhấn nút “Thanh toán” trong giỏ hàng. 2. Hệ thống hiển thị thông tin giỏ hàng từ bảng bao gồm: Tên sản phẩm, số lượng, đơn giá, thành tiền. 3. Hệ thống lấy thông tin mặc định từ bảng CUSTOMERS gồm: Tên khách hàng, số điện thoại, địa chỉ điền vào các trường tương ứng: Tên người nhận, số điện thoại người nhận, địa chỉ người nhận, và điền thêm trường: ghi chú. 4. Người dùng nhập thông tin và nhấn “Thanh toán”. 5. Hệ thống xác nhận thông tin thanh toán và thêm một đơn hàng vào bảng BILLS với thông tin: Mã khách hàng, tên người nhận, số điện thoại người nhận, địa chỉ người nhận, tổng giá trị, trạng thái (mặc định “chờ duyệt”), thời gian tạo 6. Hệ thống thêm số đơn hàng chi tiết tương ứng với số sản phẩm vào bảng DETAILS\_BILLS với thông tin từng bản ghi gồm: Mã đơn hàng, mã sản phẩm, số lượng mua. 7. Use case kết thúc khi đơn hàng được xác nhận thành công.   3.2. Luồng rẽ nhánh   1. Tại bước 2, nếu người dùng không nhập đầy đủ thông tin hoặc thông tin không hợp lệ, hệ thống thông báo lỗi và yêu cầu người dùng nhập lại. 2. Nếu hệ thống không kết nối được với cơ sở dữ liệu, hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.   4. Yêu cầu đặc biệt  Không có  5. Tiền điều kiện  Khách hàng đã đăng nhập và có sản phẩm trong giỏ hàng.  6. Hậu điều kiện  Nếu use case thực hiện thành công, đơn hàng mới được ghi vào bảng BILLS và chi tiết đơn hàng được ghi vào bảng DETAILS\_BILLS trong cơ sở dữ liệu.  7. Điểm mở rộng  Không có. |

### Mô tả use case Cập nhật thông tin người dùng

A diagram of a diagram

Description automatically generated

Hình 3.2.5 Biểu đồ hoạt động của chức năng Cập nhật thông tin người dùng

|  |
| --- |
| 1. Tên Use Case: Cập nhật thông tin người dùng  2. Mô tả vắn tắt  Use case cho phép khách hàng chỉnh sửa lại các thông tin cá nhân trong tài khoản của mình.  3. Luồng các sự kiện  3.1 Luồng cơ bản   1. Use case này bắt đầu khi khách hàng di vào biểu tượng người dùng bên góc phải. Hệ thống sẽ hiển thị ra một danh sách gồm có: “Tài khoản của tôi, Đơn mua, Đăng xuất” ra màn hình. 2. Khách hàng kích vào “tài khoản của tôi” trong danh sách vừa xuất hiện. Hệ thống hiển thị form thông tin tài khoản của khách hàng lên màn hình. 3. Chỉnh sửa thông tin đăng ký tài khoản: Khách hàng nhập thông tin gồm: email, mật khẩu, xác nhận mật khẩu và ấn “Lưu”. Hệ thống sẽ sửa thông tin đăng ký tài khoản của khách hàng và cập nhật vào bảng CUSTOMERS và hiển thị thông báo đã cập nhật. 4. Chỉnh sửa thông tin liên hệ: Khách hàng nhập thông tin gồm: họ và tên, số điện thoại, địa chỉ và ấn “Lưu”. Hệ thống sẽ sửa thông tin liên hệ của khách hàng và cập nhật vào bảng CUSTOMERS và hiển thị thông báo đã cập nhật. Use case kết thúc.   3.2. Luồng rẽ nhánh   1. Tại thời điểm bất kì trong quá trình thực hiện use case nếu không kết nối được với hệ thống cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.   4. Yêu cầu đặc biệt  Không có  5. Tiền điều kiện  Khách hàng phải thực hiện đăng nhập trước khi thực hiện use case này.  6. Hậu điều kiện  Nếu use case thực hiện thành công, hệ thống cập nhật bảng CUSTOMERS trong cơ sở dữ liệu.  7. Điểm mở rộng  Không có. |

# KIỂM THỬ WEBSITE BÁN ĐỒ ĐIỆN TỬ

## Lập kế hoạch kiểm thử

### Giới thiệu

#### Mục đích

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

* Xác định thông tin cơ bản về dự án và các thành phần chức năng được kiểm thử và không được kiểm thử.
* Liệt kê những yêu cầu cho việc kiểm thử.
* Những chiến lược kiểm thử nên được sử dụng.
* Ước lượng những yêu cầu về tài nguyên và chi phí cho việc kiểm thử.
* Những tài liệu được lập sau khi hoàn thành việc kiểm thử.

#### Tổng quan

Website bán đồ điện tử được xây dựng nhằm hỗ trợ các doanh nghiệp, cá nhân bán hàng qua dịch vụ internet, theo dõi, quản lý thông tin các loại sản phẩm, thu nhập, tài khoản, …

#### Phạm vi

Tài liệu kế hoạch kiểm thử này được áp dụng cho việc kiểm thử những chức năng của trang web Bán đồ điện tử được đặc tả trong Bài tập lớn.

Các dịch vụ, chức năng cần kiểm thử:

* Đăng ký
* Đăng nhập.
* Tìm kiếm sản phẩm
* Thanh toán
* Cập nhật tài khoản người dùng

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

* Các định nghĩa

|  |  |
| --- | --- |
| **Thuật ngữ** | **Định nghĩa** |
| user | Người dùng trang web |
| drivers | Đoạn mã giả lập mô-đun cấp trên |
| stubs | Đoạn mã giả lập mô-đun cấp dưới |
| modules | Đơn vị chức năng |
| scenarios | Tình huống sử dụng cụ thể mô tả cách người dùng tương tác với hệ thống |
| schedule | Lịch, kế hoạch |

* Danh sách các từ viết tắt

|  |  |
| --- | --- |
| **Từ viết tắt** | **Mô tả** |
| API | Application Programming Interface - Giao diện lập trình ứng dụng |
| SRS | Software Requirements Specification - Đặc tả Yêu cầu Phần mềm |
| URD | User Requirements Document - Yêu cầu Người dùng |
| UAT | User Acceptance Testing - Kiểm thử chấp nhận người dùng |
| HTTP | HyperText Transfer Protocol |
| HTTPS | HyperText Transfer Protocol Secure |
| TX | Thường xuyên |
| TG | Thời gian |
| IEEE | Institute of Electrical and Electronics Engineers - Viện Kỹ sư Điện và Điện tử |

#### Những người sử dụng tài liệu này

Tài liệu này được sử dụng cho Test Manager, Test Designer và Tester.

### Tài liệu tham khảo

* Slides môn học kiểm thử phần mềm
* Bài tập lớn dành cho sinh viên môn học Kiểm Thử Phần Mềm
* Tài liệu mẫu về kế hoạch kiểm thử của công ty CSCV

### Lịch trình công việc

Bảng 4.1 Lịch trình công việc

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Sản phẩm**  **bàn giao** | **Thời lượng** | **Ngày**  **bắt đầu** | **Ngày**  **kết thúc** |
| Lập kế hoạch kiểm thử | Tài liệu Test plan | 2 ngày | 22/5/2025 | 23/5/2025 |
| Xem lại các tài liệu | Tài liệu Test plan | 2 ngày | 25/5/2025 | 26/5/2025 |
| Thiết kế các test case | Tài liệu Test case | 1 ngày | 27/5/2025 | 27/5/2025 |
| Viết các test case | Tài liệu Test case | 2 ngày | 28/5/2025 | 29/5/2025 |
| Xem lại các test case | Tài liệu Test case | 1 ngày | 30/5/2025 | 30/5/2025 |
| Thực thi các test case | Tài liệu Test case | 1 ngày | 31/5/2025 | 31/5/2025 |
| Ghi nhận và đánh giá kết quả kiểm thử | Tài liệu Test case | 2 ngày | 31/5/2025 | 1/6/2025 |

### Những yêu cầu về tài nguyên

#### Phần cứng

Bảng 4.2 Cấu hình tối thiểu của máy tính

|  |  |  |  |
| --- | --- | --- | --- |
| **CPU** | **RAM** | **HDD** | **Architecture** |
| Inter core i5, 2.3 GHz | 8GB | 512GB | 64bit |

#### Phần mềm

Bảng 4.3 Danh sách phần mềm phục vụ cho việc kiểm thử

|  |  |  |
| --- | --- | --- |
| **Phần mềm** | **Phiên bản** | **Loại** |
| Google Chrome | 136.0.7103.114 | Trình duyệt Web |
| Microsoft Edge | 136.0.3240.76 | Trình duyệt Web |
| Cốc Cốc | 141.0.116 | Trình duyệt Web |
| Microsoft Windows | Windows 11 Enterprise | Hệ điều hành |

#### Công cụ kiểm thử

Bảng 4.4 Danh sách công cụ phục vụ cho quá trình kiểm thử

|  |  |  |  |
| --- | --- | --- | --- |
| **Hoạt động** | **Công cụ** | **Nhà cung cấp** | **Phiên bản** |
| Quản lý test case | Microsoft Office Excel | Microsoft | 2019 |
| Quản lý cấu hình | Microsoft Office Word | Microsoft | 2019 |
| Theo dõi lỗi | Microsoft Office Excel | Microsoft | 2019 |
| Quản lý tiến độ kiểm thử | Microsoft Office Excel | Microsoft | 2019 |

#### Môi trường kiểm thử

Máy tính cá nhân có kết nối mạng Internet để có thể truy cập vào trang web bán đồ điện tử bằng trình duyệt.

Các chức năng của trang web bán đồ điện tử được kiểm tra trên các trình duyệt Google Chrome, Microsoft Edge và Cốc Cốc. Hệ điều hành được sử dụng là Microsoft Windows 11 Enterprise.

#### Nhân sự

Bảng 4.5 Danh sách nhân sự

|  |  |  |
| --- | --- | --- |
| **Thành viên** | **Vai trò** | **Trách nhiệm** |
| Nguyễn Mạnh Hoàn | Test Manager  Test Designer  Tester | Lập kế hoạch kiểm thử, quản lý tiến độ hoạt động kiểm thử, thiết kế các testcase bổ sung và thực thi các testcase bổ sung.  Thiết kế và viết các testcase, thực thi các testcase cho chức năng Đăng ký |
| Nguyễn Ngọc Hưng | Test Designer  Tester | Thiết kế và viết các testcase, thực thi các testcase cho chức năng Đăng nhập bằng Postman, xem lại Test Plan |
| Lương Mạnh Hùng | Test Designer  Tester | Thiết kế và viết các testcase, thực thi các testcase cho chức năng Tìm kiếm sản phẩm bằng Postman, xem lại Test Plan |
| Hoàng Nguyễn Tùng Huy | Test Designer  Tester | Thiết kế và viết các testcase, thực thi các testcase cho chức năng Thanh toán bằng Postman, xem lại Test Plan |
| Lê Toàn Thắng | Test Designer  Tester | Thiết kế và viết các testcase, thực thi các testcase cho chức năng Cập nhật thông tin người dùng bằng Postman, xem lại Test Plan |

### Phạm vi kiểm thử

#### Những chức năng được kiểm thử

* Đăng ký

- Kiểm tra tính hợp lệ của dữ liệu nhập vào

- Kiểm tra chức năng đăng ký

* Đăng nhập

- Kiểm tra chức năng đăng nhập tài khoản

* Tìm kiếm sản phẩm

- Kiểm tra chức năng tìm kiếm sản phẩm

* Thanh toán

- Kiểm tra chức năng thanh toán

* Cập nhật thông tin người dùng

- Kiểm tra chức năng cập nhật thông tin người dùng

#### Những chức năng không được kiểm thử

Những chức năng không được đặc tả trong tài liệu:

* Xem sản phẩm
* Xem cửa hàng
* Xem đơn hàng
* Hủy đơn hàng
* Quản lý giỏ hàng
* Quản lý đơn hàng
* Quản lý tài khoản
* Quản lý sản phẩm
* Bảo trì danh mục

### Chiến lược kiểm thử

#### Các giai đoạn kiểm thử

* Dùng kiểm thử tự động (automatic) gồm kiểm thử chức năng, kiểm thử tải, kiểm thử hiệu suất
* Thiết kế test case theo phương pháp phân vùng tương đương
* Kiểm thử bằng Postman
  + Bật cổng dịch vụ Apache và MySql bằng XAMPP hoặc LARAGON
  + Chạy chương trình bằng lệnh: npm install và npm start
  + Mở trang web local thông qua cổng đã tạo (port:3000)
  + Mở Postman dán URL của chức năng cần kiểm thử
  + Nhập các trường dữ liệu theo từng test case đã tìm được qua phương pháp phân vùng tương đương -> Nhấn send để kiểm tra kết quả trả về có giống với kết quả mong đợi không

#### Các loại kiểm thử

Kiểm thử chức năng:

Bảng 4.6 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 test case 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 lập trình viên khắc phục. |
| Chịu trách nhiệm | Test Designer.  Tester. |
| Cách kiểm thử | Sử dụng công cụ và phần mềm để thực hiện các bài kiểm thử, kiểm thử chức năng trong cơ sở dữ liệu |
| Xử lý ngoại lệ | Liệt kê tất cả các vấn đề liên quan phát sinh trong quá trình thực thi kiểm thử. |

Kiểm thử hiệu suất:

Bảng 4.7 Kiểm thử hiệu suất

|  |  |
| --- | --- |
| Mục đích kiểm tra | Đảm bảo cơ sở dữ liệu hiển thị đúng, nội dung, thông tin trong điều kiện có nhiều người sử dụng |
| 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.  - Không có lỗi xuất hiện trong quá trình kiểm thử |
| Tiêu chuẩn dừng | Tất cả các test case 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 lập trình viên khắc phục. |
| Chịu trách nhiệm | Test Designer, Tester. |
| Cách kiểm thử | Sử dụng công cụ và phần mềm để thực hiện các bài kiểm thử, kiểm thử chức năng hoặc kiểm thử hiệu suất của cơ sở dữ liệu một cách tự động |
| 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

* Tỉ lệ test case đạt (passed): 100%
* Tỉ lệ test case không đạt (failed): 0%
* Cơ sở dữ liệu có hiệu suất không quá kém, đáp ứng nhu cầu cho điều kiện có nhiều users sử dụng
* Dữ liệu lưu trữ đúng điều kiện đầu vào, cho phép thao tác để sử dụng dữ liệu

### Theo dõi lỗi

#### Phân loại lỗi

Bảng 4.8 Phân loại lỗi

|  |  |
| --- | --- |
| **Mức độ nghiêm trọng** | **Đặc tả lỗi** |
| Cao | * Không thể đăng ký tài khoản mới * Không thể đăng nhập vào tài khoản * Không thể thực hiện thanh toán * Không thể tìm kiếm sản phẩm |
| Trung bình | * Hiển thị sai thông tin người dùng, sản phẩm hoặc đơn hàng * Không cập nhật thông tin cá nhân sau khi chỉnh sửa * Kết quả tìm kiếm không đúng với từ khoá |
| Thấp | * Cho phép đăng ký với dữ liệu sai định dạng (email không hợp lệ, thiếu mật khẩu, v.v.) * Hệ thống phản hồi chậm khi tìm kiếm sản phẩm hoặc xử lý thanh toán * Hiển thị thông báo lỗi không rõ ràng hoặc thiếu thông báo |

#### Quy trình xử lý lỗi

Ghi nhận lại các lỗi được tìm thấy trong quá trình kiểm thử.

Tiến hành sửa các lỗi, sửa chữa, bổ sung cơ sở dữ liệu cũ để trang web hoàn thiện hơn.

## Thực hiện kiểm thử

### Nguyễn Mạnh Hoàn - Đăng ký

#### Phân tích thiết kế kiểm thử

**- Giao diện chức năng:**

A screenshot of a phone

AI-generated content may be incorrect.

Hình 4.2.1 Giao diện chức năng đăng ký

**- Mô tả giao diện:** Nhập đầy đủ các thông tin cần thiết, sau đó Click nút “Tiếp theo”

**- Mô tả điều kiện:**

+ Tất cả các trường bắt buộc nhập.

+ Số điện thoại chỉ được chứa số và từ 9 – 11 số.

+ Email phải nhập đúng định dạng.

+ Mật khẩu phải có độ dài tối thiểu là 6 ký tự, phải chứa ít nhất 1 số, 1 chữ cái và một ký tự đặc biệt.

+ Số điện thoại và email phải không trùng với tài khoản đã tồn tại

**- Phương pháp tìm Test Case:** Dùng phương pháp phân vùng tương đương

**- Bảng phân vùng tương đương:**

Bảng 4.9 *Bảng phân vùng tương đương (Chức năng đăng ký)*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Đầu vào** | **Vùng hợp lệ** | **Kí hiệu** | **Vùng không hợp lệ** | **Kí hiệu** |
| Họ và tên | Không để trống | H1 | Để trống | K1 |
| Số điện thoại | Dài từ 9 – 11 số | H2 | < 9 số | K2 |
| > 11 số | K3 |
| Chỉ chứa số | H3 | Có ký tự | K4 |
| Để trống | K5 |
| Số điện thoại chưa được dùng để đăng ký | H4 | Số điện thoại đã được dùng để đăng ký | K6 |
| Email | Nhập đúng định dạng mail ([abc@gmail.com](mailto:abc@gmail.com)) | H5 | Nhập sai định dạng mail | K7 |
| Để trống | K8 |
| Email chưa được dùng để đăng ký | H6 | Email đã được dùng để đăng ký | K9 |
| Mật khẩu | Dài hơn 6 ký tự | H7 | < 6 ký tự | K10 |
| Để trống | K11 |
| Có > 1 ký tự đặc biệt | H8 | Không có ký tự đặc biệt | K12 |
| Có > 1 chữ cái | H9 | Không có ký tự chữ | K13 |
| Có > 1 ký tự số | H10 | Không có ký tự số | K14 |
| Nhập lại mật khẩu | Mật khẩu nhập lại phải khớp với mật khẩu đã đăng ký | H11 | Mật khẩu nhập lại không khớp | K15 |
| Để trống | K16 |

**- Bảng thiết kế TestCase:**

Bảng 4.10 Bảng thiết kế test case (Chức năng đăng ký)

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | **Ghi chú** | **Đầu vào (Input)** | **Đầu ra (Output)** |
| **a = Họ và tên**  **b = Số điện thoại**  **c = Email**  **d = Mật khẩu**  **e = Nhập lại mật khẩu** |
| **TC1** | H1, H2, H3, H4, H5, H6, H7, H8, H9, H10, H11 | a = an1  b = 0123456789  c = [an1@gmail.com](mailto:an1@gmail.com)  d = abc123!@#  e = abc123!@# | Đăng ký thành công |
| **TC2** | K1, H2, H3, H4, H5, H6, H7, H8, H9, H10, H11 | a =  b = 0123456789  c = [an1@gmail.com](mailto:an1@gmail.com)  d = abc123!@#  e = abc123!@# | Vui lòng nhập đầy đủ thông tin |
| **TC3** | H1, K2, H3, H4, H5, H6, H7, H8, H9, H10, H11 | a = an1  b = 01234567  c = [an1@gmail.com](mailto:an1@gmail.com)  d = abc123!@#  e = abc123!@# | Số điện thoại không hợp lệ |
| **TC4** | H1, K3, H3, H4, H5, H6, H7, H8, H9, H10, H11 | a = an1  b = 012345678910  c = [an1@gmail.com](mailto:an1@gmail.com)  d = abc123!@#  e = abc123!@# | Số điện thoại không hợp lệ |
| **TC5** | H1, H2, K4, H4, H5, H6, H7, H8, H9, H10, H11 | a = an1  b = 0123456789s  c = [an1@gmail.com](mailto:an1@gmail.com)  d = abc123!@#  e = abc123!@# | Số điện thoại không hợp lệ |
| **TC6** | H1, H2, K5, H4, H5, H6, H7, H8, H9, H10, H11 | a = an1  b =  c = [an1@gmail.com](mailto:an1@gmail.com)  d = abc123!@#  e = abc123!@# | Vui lòng nhập đầy đủ thông tin |
| **TC7** | H1, H2, H3, K6, H5, H6, H7, H8, H9, H10, H11 | a = an1  b = 0334456789  c = [an1@gmail.com](mailto:an1@gmail.com)  d = abc123!@#  e = abc123!@# | Email hoặc số điện thoại đã tồn tại |
| **TC8** | H1, H2, H3, H4, K7, H6, H7, H8, H9, H10, H11 | a = an1  b = 0123456789  c = [an1gmail.com](mailto:an1@gmail.com)  d = abc123!@#  e = abc123!@# | Email không hợp lệ |
| **TC9** | H1, H2, H3, H4, K8, H6, H7, H8, H9, H10, H11 | a = an1  b = 0123456789  c =  d = abc123!@#  e = abc123!@# | Vui lòng nhập đầy đủ thông tin |
| **TC10** | H1, H2, H3, H4, H5, K9, H7, H8, H9, H10, H11 | a = an1  b = 0123456789  c = abc@gmai.com  d = abc123!@#  e = abc123!@# | Email hoặc số điện thoại đã tồn tại |
| **TC11** | H1, H2, H3, H4, H5, H6, K10, H8, H9, H10, H11 | a = an1  b = 0123456789  c = [an1@gmail.com](mailto:an1@gmail.com)  d = a12!@  e = a12!@ | Mật khẩu phải có ít nhất 6 ký tự |
| **TC12** | H1, H2, H3, H4, H5, H6, K11, H8, H9, H10, H11 | a = an1  b = 0123456789  c = [an1@gmail.com](mailto:an1@gmail.com)  d =  e = abc123!@# | Vui lòng nhập đầy đủ thông tin |
| **TC13** | H1, H2, H3, H4, H5, H6, H7, K12, H9, H10, H11 | a = an1  b = 0123456789  c = [an1@gmail.com](mailto:an1@gmail.com)  d = abc123  e = abc123 | Mật khẩu phải chứa ít nhất 1 ký tự đặc biệt |
| **TC14** | H1, H2, H3, H4, H5, H6, H7, H8, K13, H10, H11 | a = an1  b = 0123456789  c = [an1@gmail.com](mailto:an1@gmail.com)  d = 123!@#  e = 123!@# | Mật khẩu phải chứa ít nhất 1 chữ cái |
| **TC15** | H1, H2, H3, H4, H5, H6, H7, H8, H9, K14, H11 | a = an1  b = 0123456789  c = [an1@gmail.com](mailto:an1@gmail.com)  d = abc!@#  e = abc!@# | Mật khẩu phải chứa ít nhất 1 số |
| **TC16** | H1, H2, H3, H4, H5, H6, H7, H8, H9, H10, K15 | a = an1  b = 0123456789  c = [an1@gmail.com](mailto:an1@gmail.com)  d = abc123!@#  e = 123!@# | Mật khẩu nhập lại không khớp |
| **TC17** | H1, H2, H3, H4, H5, H6, H7, H8, H9, H10, K16 | a = an1  b = 0123456789  c = [an1@gmail.com](mailto:an1@gmail.com)  d = abc123!@#  e = | Vui lòng nhập đầy đủ thông tin |

- **TestCase - DangKy.xlsx:**

A white rectangular box with black text

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.2.2 Test case – Đăng ký

#### Thực hiện kiểm thử

- Các bước thực hiện kiểm thử tự động: Công việc kiểm thử tự động được thực hiện bằng cách sử dụng Postman

- Ta có kịch bản kiểm thử:

+ Đăng ký thành công:

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.2.3 Kịch bản kiểm thử đăng ký thành công

+ Họ và tên để trống:

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.2.4 Kịch bản kiểm thử họ tên để trống

+ Số điện thoại < 9 số:

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.2.5 Kịch bản kiểm thử số điện thoại < 9 số

+ Số điện thoại > 11 số:

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.2.6 Kịch bản kiểm thử số điện thoại > 11 số

+ Số điện thoại chứa ký tự chữ:

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.2.7 *Kịch bản kiểm thử số điện thoại chứa ký tự chữ*

+ Số điện thoại để trống:

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.2.8 Kịch bản kiểm thử số điện thoại để trống

+ Số điện thoại đã tồn tại:

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.2.9 Kịch bản kiểm thử số điện thoại đã tồn tại

+ Sai định dạng mail:

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.2.10 Kịch bản kiểm thử sai định dạng mail

+ Email để trống:

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.2.11 Kịch bản kiểm thử email để trống

+ Email đã tồn tại:

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.2.12 Kịch bản kiểm thử email đã tồn tạị

+ Mật khẩu < 6 ký tự:

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.2.13 Kịch bản kiểm thử mật khẩu < 6 ký tự

+ Mật khẩu để trống:

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.2.14 Kịch bản kiểm thử mật khẩu để trống

+ Mật khẩu không có ký tự đặc biệt:

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.2.15 *Kịch bản kiểm thử mật khẩu không có ký tự đặc biệt*

+ Mật khẩu không có ký tự chữ:

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.2.16 *Kịch bản kiểm thử mật khẩu không có ký tự chữ*

+ Mật khẩu không có số:

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.2.17 *Kịch bản kiểm thử mật khẩu không có ký tự số*

+ Mật khẩu nhập lại không khớp:

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.2.18 *: Kịch bản kiểm thử mật khẩu nhập lại không khớp*

+ Không nhập lại mật khẩu:

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.2.19 *Kịch bản kiểm thử không nhập lại mật khẩu*

- Chọn “Send” trong Postman để tiến hành kiểm thử

=> Hệ thống trả về thông báo kiểm thử thành công

=> Tỉ lệ thành công (Passed) so với số TC đã lập: 100%

### Nguyễn Ngọc Hưng - Đăng nhập

#### Phân tích thiết kế kiểm thử

* **Giao diện chức năng:**

**A screenshot of a computer

AI-generated content may be incorrect.**

*Hình 4.2.20 Giao diện chức năng đăng nhập*

* **Mô tả giao diện:** Nhập các thông tin: Tên đăng nhập và Mật khẩu, sau đó Click nút “ ĐĂNG NHẬP ”.
* **Mô tả điều kiện:**
* Tất cả các trường đều bắt buộc nhập
* Tên đăng nhập: phải là email
* Mật khẩu: có ít nhất 6 ký tự (phải chứa ít nhất 1 chữ cái, phải chứa ít nhất 1 số, phải chứa ít nhất 1 ký tự đặc biệt).
* **Phương pháp tìm TestCase:** dùng phương pháp nhân vùng tương đương.
* **Bảng phân vùng tương đương:**

Bảng 4.11 *Bảng phân vùng tương đương (chức năng đăng nhập)*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Đầu vào** | **Vùng hợp lệ** | **Ký hiệu** | **Vùng không hợp lệ** | **Ký hiệu** |
| Tên đăng nhập | Là một email | H1 | Không là email | K1 |
| Để trống | K2 |
| Mật khẩu | Dải ít nhất 6 ký tự | H2 | < 6 ký tự | K3 |
| Để trống | K4 |
| Có ít nhất 1 chữ cái | H3 | Không có chữ cái | K5 |
| Có ít nhất 1 số | H4 | Không có số | K6 |
| Có ít nhất 1 ký tự đặc biệt | H5 | Không có ký tự đặc biệt | K7 |

* **Bảng thiết kế TestCase**

Bảng 4.12 *Bảng thiết kế test case (chức năng đăng nhập)*

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | **Ghi chú** | **Đầu vào (Input)** | **Đầu ra (Output)** |
| **a=tên đăng nhập; b=mật khẩu** |
| TC1 | H1,H2,H3,H4,H5,H6 | a=nghung2k4@gmail.com; b=Hung123@ | Đăng nhập thành công |
| TC2 | K1,H2,H3,H4,H5,H6 | a=hung; b=Hung123@ | Sai tên đăng nhập hoặc mật khẩu |
| TC3 | K2,H2,H3,H4,H5,H6 | a= ; b=Hung123@ | Sai tên đăng nhập hoặc mật khẩu |
| TC4 | H1,K3,H3,H4,H5,H6 | a=nghung2k4@gmail.com; b=Hung@ | Sai tên đăng nhập hoặc mật khẩu |
| TC5 | H1,K4,H3,H4,H5,H6 | a=nghung2k4@gmail.com; b= | Sai tên đăng nhập hoặc mật khẩu |
| TC6 | H1,H2,K5,H4,H5,H6 | a=nghung2k4@gmail.com; b=12345@ | Sai tên đăng nhập hoặc mật khẩu |
| TC7 | H1,H2,H3,K6,H5,H6 | a=nghung2k4@gmail.com; b=Hungg@ | Sai tên đăng nhập hoặc mật khẩu |
| TC8 | H1,H2,H3,H4,K7,H6 | a=nghung2k4@gmail.com; b=Hung123 | Sai tên đăng nhập hoặc mật khẩu |

**TestCase – DangNhap.xlsx**

**A white grid with black and red text

AI-generated content may be incorrect.**

Hình 4.2.21 Test case – Đăng nhập

#### Thực hiện kiểm thử

* Các bước thực hiện kiểm thử tự động: Công việc kiểm thử tự động được thực hiện bằng cách sử dụng Postman
* Ta có kịch bản kiểm thử:
* Đăng nhập thành công

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.2.22 Kịch bản kiểm thử đăng nhập thành công

* Tên đăng nhập không là email

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.2.23 Kịch bản kiểm thử tên đăng nhập không phải email

* Để trống tên đăng nhập

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.2.24 Kịch bản kiểm thử bỏ trống tên đăng nhập

* Mật khẩu < 6 ký tự

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.2.25 Kịch bản kiểm thử mật khẩu < 6 ký tự

* Để trống mật khẩu

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.2.26 Kịch bản kiểm thử để trống mật khẩu

* Mật khẩu không có chữ cái

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.2.27 Kịch bản kiểm thử mật khẩu không có chữ cái

* Mật khẩu không có số

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.2.28 Kịch bản kiểm thử mật khẩu không có số

* Mật khẩu không có ký tự đặc biệt

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.2.29 Kịch bản kiểm thử mật khẩu không có ký tự đặc biệt

* Chọn “Send” trong Postman để tiến hành kiểm thử

=> Hệ thống trả về thông báo kiểm thử thành công

=> Tỉ lệ thành công (Passed) so với số TC đã lập: 100%

### Lương Mạnh Hùng - Tìm kiếm sản phẩm

#### Phân tích thiết kế kiểm thử

**-Giao diện chức năng**

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.2.30 *Giao diện chức năng tìm kiếm*

**- Mô tả giao diện:** Nhập tên sản phẩm cần tìm kiếm, sau đó Click biểu tượng kính lúp hoặc nhấn “Enter”

**- Mô tả điều kiện:**

+ Từ khóa không được để trống.

+ Từ khóa phải hợp lệ.

+ Có thể phân biệt chữ hoa/thường

+Có thể tìm kiếm không dấu

+Không chứa các kí tự không hợp lệ (như @, #, $, …)

**Phương pháp tìm Test Case:** Dùng phương pháp phân vùng tương đương

**- Bảng phân vùng tương đương:**

Bảng 4.13 *Bảng phân vùng tương đương (Chức năng tìm kiếm)*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Đầu vào** | **Vùng hợp lệ** | **Kí hiệu** | **Vùng không hợp lệ** | **Kí hiệu** |
| Tìm kiếm | Chuỗi hợp lệ, có kết quả | H1 | Chuỗi rỗng | K1 |
|  | Chuỗi hợp lệ, không có kết quả | H2 | Chuỗi vượt quá 100 ký tự | K2 |
| Chuỗi hợp lệ, không dấu | H3 | Chuỗi chứa ký tự đặc biệt | K3 |
|  | Chuỗi không phân biệt hoa, thường | H4 |  |  |

**- Bảng thiết kế TestCase:**

Bảng 4.14 *Bảng thiết kế test case (Chức năng tìm kiếm)*

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | **Ghi chú** | **Đầu vào (Input)** | **Đầu ra (Output)** |
| **TC1** | H1 | “Chuột” | Hiển thị danh sách sản phẩm chuột |
| **TC2** | H2 | “Xe hơi” | Hiển thị thông báo “Không tìm thấy sản phẩm” |
| **TC3** | H3 | “Chuot” | Hiển thị danh sách sản phẩm chuột |
| **TC4** | H4 | “CHUộT” | Hiển thị danh sách sản phẩm chuột |
| **TC5** | K1 | “”(rỗng) | Hiển thị thông báo lỗi hoặc tất cả sản phẩm |
| **TC6** | K2 | 101 ký tự "aaaaaaaaaa..." | Hiển thị lỗi: “Từ khóa quá dài” |
| **TC7** | K3 | “@#$#@” | Hiển thị lỗi: “Từ khóa chứa ký tự không hợp lệ” |

- **TestCase - TimKiem.xlsx:**

A white sheet with black text

AI-generated content may be incorrect.

Hình 4.2.31 *Test case – Tìm kiếm*

#### Thực hiện kiểm thử

- Các bước thực hiện kiểm thử tự động: Công việc kiểm thử tự động được thực hiện bằng cách sử dụng Postman

- Ta có kịch bản kiểm thử:

+Từ khóa hợp lệ, có sản phẩm:

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.2.32 Kịch bản kiểm thử từ khóa hợp lệ, có kiểm thử

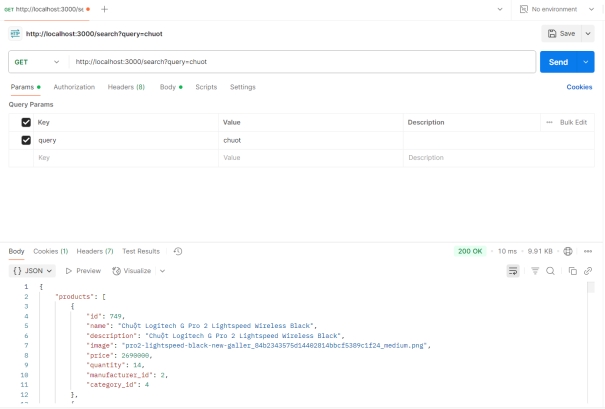
+Từ khóa hợp lệ, không có sản phẩm:

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.2.33 *Kịch bản kiểm thử từ khóa hợp lệ, không có sản phẩm*

+Từ khóa hợp lệ, không dấu:



Hình 4.2.34 *Kịch bản kiểm thử từ khóa hợp lệ, không dấu*

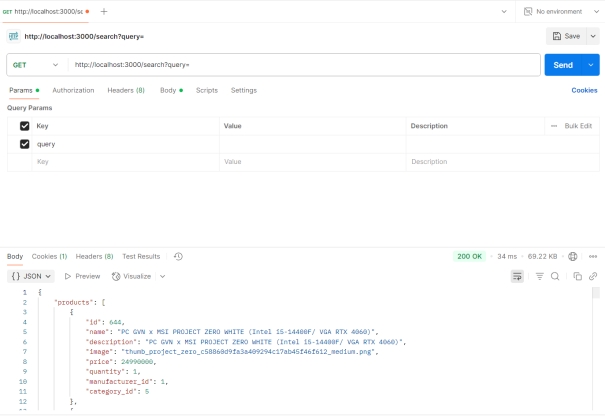
+Từ khóa hợp lệ, viết hoa và thường đan xen

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.2.35 *Kịch bản kiểm thử từ khóa hợp lệ, viết hoa thường đan xen*

+Không nhập từ khóa:



Hình 4.2.36 *Kịch bản kiểm thử không nhập từ khóa*

+Nhập quá dài:

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.2.37 *Kịch bản kiểm thử nhập quá dài*

**+**Có kí tự không hợp lệ:

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.2.38 *Kịch bản kiểm thử có kí tự không hợp lệ*

- Chọn “Send” trong Postman để tiến hành kiểm thử

=> Hệ thống trả về thông báo kiểm thử thành công

=> Tỉ lệ thành công (Passed) so với số TC đã lập: 100%

### Hoàng Nguyễn Tùng Huy - Thanh toán

#### Phân tích thiết kế kiểm thử

**- Giao diện kiểm thử**

A screenshot of a qr code

AI-generated content may be incorrect.

Hình 4.2.39 *Giao diện chức năng đăng ký*

**- Mô tả giao diện:** Quét mã QR để thanh toán hoặc chuyển khoản

**- Mô tả điều kiện:**

+ Tất cả các trường bắt buộc nhập.

+ Số tiền chỉ được phép lớn hơn 0,01 và nhỏ hơn 10000000000

+ Số tiền là chuỗi

+ Số tiền là số âm

**- Phương pháp tìm Test Case:** Dùng phương pháp phân vùng tương đương

**- Bảng phân vùng tương đương:**

Bảng 4.15 Bảng phân vùng tương đương (Chức năng thanh toán)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Đầu vào** | **Vùng hợp lệ** | **Kí hiệu** | **Vùng không hợp lệ** | **Kí hiệu** |
| Số tiền | Không để trống | H1 | Để trống | K1 |
| Lớn hơn 0,01 | H2 | < 0,01 | K2 |
| nhỏ hơn 10000000000 | H3 | > 10000000000 | K3 |
| Không viết chuỗi ký tự | H4 | Có chuỗi ký tự | kK4 |
|  |
| Không phải số âm | H5 | Số âm | K5 |  |
| Nội dung | Không để trống | H6 | Để trống | K6 |  |

**- Bảng thiết kế TestCase:**

Bảng 4.16 *Bảng thiết kế test case (Chức năng thanh toán)*

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | **Ghi chú** | **Đầu vào (Input)** | **Đầu ra (Output)** |
| **a = Số tiền**  **b = Nội dung** |
| **TC1** | H1, H2, H3, H4, H5, H6 | a = 100000  b = giao dich moi | Thanh toán thành công |
| **TC2** | K1, H2, H3, H4, H5, H6 | a =  b = giao dich moi | Vui lòng nhập số tiền |
| **TC3** | H1, K2, H3, H4, H5, H6 | a = 0,0001  b = giao dịch moi | Số tiền quá nhỏ để thanh toán |
| **TC4** | H1, H2, K3, H4, H5, H6 | a = 100000000000  b = giao dich moi | Số tiền quá lớn để thanh toán |
| **TC5** | H1, H2, H3, K4, H5, H6, | a = sotienla  b = giao dich moi | Số tiền không hợp lệ |
| **TC6** | H1, H2, H3, H4, K5, H6 | a = -10000  b = giao dich moi | Số tiền không nhập số âm |
| **TC7** | H1, H2, H3, K6, H5, K6 | a = an1  b = | Vui lòng không để trống nội dung |

- **TestCase - ThanhToan.xlsx:**

A white sheet with black text

AI-generated content may be incorrect.

Hình 4.2.40 *Test case – Thanh toán*

#### Thực hiện kiểm thử

- Các bước thực hiện kiểm thử tự động: Công việc kiểm thử tự động được thực hiện bằng cách sử dụng Postman

- Ta có kịch bản kiểm thử:

* Thanh toán thành công

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.2.41 *Kịch bản kiểm thử thanh toán thành công*

* Số tiền để trống

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.2.42 *Kịch bản kiểm thử số tiền để trống*

* Số tiền nhỏ hơn 0,0001

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.2.43 Kịch bản kiểm thử số tiền nhỏ hơn 0,0001

* Số tiền lớn hơn 10000000000

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.2.44 Kịch bản kiểm thử số tiền lớn hơn 10000000000

* Số tiền là chuỗi ký tự

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.2.45 Kịch bản kiểm thử số tiền là chuỗi ký tự

* Số tiền là số âm

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.2.46 *Kịch bản kiểm thử số tiền là số âm*

* Nội dung để trống

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.2.47 *Kịch bản kiểm thử nội dung để trống*

- Chọn “Send” trong Postman để tiến hành kiểm thử

=> Hệ thống trả về thông báo kiểm thử thành công

=> Tỉ lệ thành công (Passed) so với số TC đã lập: 100%

### Lê Toàn Thắng – Cập nhật thông tin người dùng

#### Phân tích thiết kế kiểm thử

**- Giao diện chức năng**

A white sheet with black lines

AI-generated content may be incorrect.

Hình 4.2.48 *Giao diện chức năng cập nhật tài khoản*

**- Mô tả giao diện:** Nhập đầy đủ các thông tin cần thiết, sau đó Click nút “Sửa”

**- Mô tả điều kiện:**

+ Tất cả các trường bắt buộc nhập.

+ Ngày sinh phải chọn ngày hợp lệ.

+ Giới tính bắt buộc chọn.

+ Số điện thoại phải chỉ có từ 9-11, không chứa ký tự.

+ Số điện thoại không trùng với tài khoản đã tồn tại

**- Phương pháp tìm Test Case:** Dùng phương pháp phân vùng tương đương - **Bảng phân vùng tương đương:**

Bảng 4.17 Bảng phân vùng tương đương (Chức năng cập nhật tài khoản)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Đầu vào** | **Vùng hợp lệ** | **Kí hiệu** | **Vùng không hợp lệ** | **Kí hiệu** |
| Họ và tên | Không để trống | H1 | Để trống | K1 |
| Giới tính | Chọn giá trị | H2 | Không chọn giá trị | K2 |
| Ngày sinh | Chọn ngày sinh hợp lệ | H3 | Chọn ngày sinh không hợp lệ | K3 |
| Số điện thoại | Số điện thoại hợp lệ | H4 | Số điện thoại đã được dùng để đăng ký | K4 |
| Có chứa ký tự | K5 |
| <9 số | K6 |
| >11 số | K7 |
| Địa chỉ | Không để trống | H5 | Để trống | K8 |

* **Bảng thiết kế Test case:**

Bảng 4.18 Bảng thiết kế test case (Chức năng cập nhật tài khoản)

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | **Ghi chú** | **Đầu vào** | **Đầu ra (Output)** |
| **a = Họ và tên**  **b = Giới tính**  **c = Ngày sinh**  **d = Số điện thoại**  **e = Địa chỉ** |
| TC1 | H1, H2, H3, H4, H5 | a = Nguyễn Văn A  b = 1  c = 1990-01-01  d = 0199192911  e = 133 | Sửa thành công |
| TC2 | K1, H2, H3, H4, H5 | a =  b = 1  c = 1990-01-01  d = 0199192911  e = 133 | Tên không được để trống |
| TC3 | H1, K2, H3, H4, H5 | a = Nguyễn Văn A  b =  c = 1990-01-01  d = 0199192911  e = 133 | Hãy chọn giới tính phù hợp |
| TC4 | H1, H2, K3, H4, H5 | a = Nguyễn Văn A  b = 1  c = **2222-01-01**  d = 0199192911  e = 133 | Ngày sinh không được lớn hơn ngày hiện tại |
| TC5 | H1, H2, H3, K4, H5 | a = Nguyễn Văn A  b = 1  c = 1990-01-01  d = 0327272297  e = 133 | Lỗi khi cập nhật người dùng |
| TC6 | H1, H2, H3, K5, H5 | a = Nguyễn Văn A  b = 1  c = 1990-01-01  d = 0199192911a  e = 133 | Số điện thoại phải từ 9-11 số và không chứa ký tự khác |
| TC7 | H1, H2, H3, K6, H5 | a = Nguyễn Văn A  b = 1  c = 1990-01-01  d = 01991929  e = 133 | Số điện thoại phải từ 9-11 số và không chứa ký tự khác |
| TC8 | H1, H2, H3, K7, H5 | a = Nguyễn Văn A  b = 1  c = 1990-01-01  d = 019919291111  e = 133 | Số điện thoại phải từ 9-11 số và không chứa ký tự khác |
| TC9 | H1, H2, H3, H4, K8 | a = Nguyễn Văn A  b = 1  c = 1990-01-01  d = 0199192911  e = | Địa chỉ không được để trống |

* [TestCase-CapNhatThongTinNguoiDung.xlsx](file:///C:\Users\thwng\OneDrive\Desktop\TestCase-CapNhatThongTinNguoiDung.xlsx):

A white sheet with black text

AI-generated content may be incorrect.

*Hình 4.49: Test case – Cập nhật tài khoản*

#### Thực hiện kiểm thử

- Các bước thực hiện kiểm thử tự động: Công việc kiểm thử tự động được thực hiện bằng cách sử dụng Postman

- Ta có kịch bản kiểm thử:

+ Sửa thành công:

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.2.49 *Kịch bản kiểm thử sửa thành công*

+ Họ và tên để trống:

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.2.50 *Kịch bản kiểm thử họ và tên để trống*

+ Giới tính để trống:

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.2.51 *Kịch bản kiểm thử giới tính không phù hợp*

+ Số điện thoại trùng

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.2.52 Kịch bản kiểm thử lỗi khi cập nhật điện thoại đã dùng

+ Ngày sinh lớn hơn ngày hiện tại:

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.2.53 *Kịch bản kiểm thử ngày sinh không lớn hơn ngày hiện tại*

+ Địa chỉ để trống:

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.2.54 *Kịch bản kiểm thử địa chỉ không được để trống*

+ Số điện thoại <9 số:

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.2.55 *Kịch bản kiểm thử số điện thoại có ít hơn 9*

+ Số điện thoại có chứa ký tự khác:

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.2.56 *Kịch bản kiểm thử số điện thoại chứa ký tự*

+ Số điện thoại lớn hơn 11 số:

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.2.57 Kịch bản kiểm thử số điện thoại phải từ 9-11 số, không chứa ký tự khác

- Chọn “Send” trong Postman để tiến hành kiểm thử

=> Hệ thống trả về thông báo kiểm thử thành công

=> Tỉ lệ thành công (Passed) so với số TC đã lập: 100%

# KẾT LUẬN

## 1. Kết quả đạt được

Qua quá trình thực hiện đề tài “Tìm hiểu công cụ kiểm thử Postman và ứng dụng vào kiểm thử API cho website bán đồ điện tử”, nhóm đã đạt được nhiều kết quả tích cực. Trước hết, nhóm đã hiểu rõ vai trò quan trọng của kiểm thử API trong phát triển phần mềm, đặc biệt là trong các hệ thống web hiện đại có kiến trúc phân lớp giữa frontend và backend. Đồng thời, nhóm đã nắm vững cách sử dụng công cụ Postman để thực hiện kiểm thử API, bao gồm việc tạo request, thiết lập môi trường kiểm thử, viết script để kiểm tra phản hồi cũng như thực hiện kiểm thử tự động.

Bên cạnh đó, nhóm đã áp dụng thành công Postman vào kiểm thử các chức năng chính của website bán đồ điện tử như đăng ký, đăng nhập, tìm kiếm sản phẩm, thanh toán và cập nhật thông tin cá nhân. Trong quá trình kiểm thử, các lỗi được phát hiện đã được ghi nhận, phân loại theo mức độ nghiêm trọng (cao, trung bình, thấp), từ đó hỗ trợ hiệu quả cho nhóm phát triển trong việc sửa lỗi và hoàn thiện hệ thống. Đề tài cũng giúp các thành viên rèn luyện kỹ năng thiết kế kịch bản kiểm thử, tư duy logic và sử dụng công cụ kiểm thử hiện đại, qua đó nâng cao kiến thức chuyên môn và chuẩn bị tốt hơn cho các dự án phần mềm trong tương lai.

## 2. Hướng phát triển

Trong thời gian tới, đề tài có thể được mở rộng theo nhiều hướng để tăng tính thực tiễn và chuyên sâu hơn. Trước hết, nhóm có thể tích hợp công cụ **Newman** – một CLI hỗ trợ chạy Collection của Postman – để thực hiện kiểm thử tự động trong các quy trình CI/CD, giúp kiểm thử API được diễn ra liên tục và đồng bộ với quá trình phát triển phần mềm. Bên cạnh đó, việc kết hợp Postman với các công cụ báo cáo nâng cao như **Allure Report** hoặc **ReportPortal** sẽ giúp hiển thị kết quả kiểm thử trực quan và dễ theo dõi hơn.

Ngoài ra, phạm vi kiểm thử cũng có thể được mở rộng từ chức năng sang kiểm thử hiệu năng (Performance Testing) bằng cách tích hợp với các công cụ như JMeter hoặc k6. Đồng thời, nhóm có thể phát triển thêm các tập dữ liệu kiểm thử với nhiều trường hợp đặc biệt để đảm bảo tính bao phủ cao và khả năng phát hiện lỗi toàn diện hơn. Trong tương lai, việc xây dựng một hệ thống kiểm thử tự động hoàn chỉnh kết hợp cả API, giao diện người dùng (UI), và bảo mật sẽ giúp nâng cao chất lượng sản phẩm phần mềm và tối ưu quy trình kiểm thử cho các hệ thống thực tế.

# TÀI LIỆU THAM KHẢO

[1]: https://www.guru99.com/end-to-end-testing.html

[2]: End To End Testcases In TestRigor:

<https://www.youtube.com/watch?v=9VPTyAQ-JLI&list=PL6flErFppaj3V4kreRE9x2M7MHHapXrJi&index=7>

<https://www.guru99.com/test-case.html>

<https://www.guru99.com/what-everybody-ought-to-know-about-test-planing.html>

<https://www.guru99.com/test-plan-for-project.html>

<https://www.guru99.com/web-application-testing.html>

Tài liệu: Kiểm thử và đảm bảo chất lượng phần mềm (Thạc Bình Cường)

Giáo trình Kiểm thử phần mềm (Đại học Công nghiệp Hà Nội)