

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á
KHOA: CÔNG NGHỆ THÔNG TIN



BÀI TẬP LỚN
HỌC PHẦN: KIỂM THỦ PHẦN MỀM

ĐỀ TÀI 67: KIỂM THỦ WEB E-COMMERCE

Sinh viên thực hiện	Lớp	Khóa
Nguyễn Văn Đức	DCCNTT12.10.7	K12
Hoàng Uyển Nhi	DCCNTT12.10.7	K12
Nguyễn Thị Hồng Thúy	DCCNTT12.10.7	K12

Bắc Ninh, năm 2024

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á
KHOA: CÔNG NGHỆ THÔNG TIN

BÀI TẬP LỚN

HỌC PHẦN: KIỂM THỦ PHẦN MỀM

ĐỀ TÀI 67: KIỂM THỦ WEB E-COMMERCE

STT	Sinh viên thực hiện	Mã sinh viên	Điểm bằng số	Điểm bằng chữ
1	Nguyễn Văn Đức	20211841		
2	Hoàng Uyển Nhi	20211920		
3	Nguyễn Thị Hồng Thúy	20212068		

CÁN BỘ CHẤM 1

(Ký và ghi rõ họ tên)

CÁN BỘ CHẤM 2

(Ký và ghi rõ họ tên)

MỤC LỤC

DANH MỤC BẢNG BIỂU	5
DANH MỤC HÌNH VẼ	7
LỜI MỞ ĐẦU.....	10
CHƯƠNG 1: TỔNG QUAN BÀI TOÁN.....	11
1. Giới thiệu đề tài.....	11
2. Đặc tả yêu cầu.....	12
2.1 Các luồng nghiệp vụ chính	12
2.2 Các màn hình chức năng chính	12
2.3 Các module chính.....	26
3. Phân công công việc	27
CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ TEST	28
1. Kiểm thử đơn vị (Unit Testing).....	28
1.1 Kiểm thử đơn vị là gì?	28
1.2 Phương pháp, kĩ thuật của kiểm thử đơn vị.....	28
1.3 Mô tả các đối tượng	30
1.4 Phương thức	32
1.5 Unit test case list.....	35
2. Kiểm thử tích hợp (Intergration Testing).....	39
2.1 Kiểm thử tích hợp là gì?	39
2.2 Phương pháp, kĩ thuật của kiểm thử tích hợp	40
2.3 Trang quản trị	41
2.4 Integration Testcase.....	48
3. Kiểm thử hệ thống (System testing)	54
3.1 Kiểm thử hệ thống là gì?	54
3.2 Phương pháp, kĩ thuật của kiểm thử hệ thống.....	55
3.3 Biểu đồ hoạt động (Activity Diagrams).....	56
3.4 System Test Case.....	63
CHƯƠNG 3: THỰC THI TEST VÀ BÁO CÁO KẾT QUẢ TEST.....	64
1. Unit Test.....	64
1.1 Viết script thực hiện Unit Test.....	64
1.2 Báo cáo kết quả	66
2. Integration Test.....	66
2.1 Viết script thực hiện Integration Test.....	66

2.2	Báo cáo kết quả	68
CHƯƠNG 4: AUTOMATION TEST		69
1.	Công cụ sử dụng.....	69
2.	Kết quả đạt được.....	69
KẾT LUẬN.....		70
TÀI LIỆU THAM KHẢO.....		71

DANH MỤC BẢNG BIỂU

Số hiệu	Tên	Trang
1	<i>Bảng 1: Mô tả phần tử NavBar user</i>	11
2	<i>Bảng 2: Mô tả trường danh mục sản phẩm</i>	12
3	<i>Bảng 3: Mô tả trường đăng ký</i>	13
4	<i>Bảng 4: Mô tả trường đăng nhập</i>	14
5	<i>Bảng 5: Mô tả trường thông tin cá nhân</i>	15
6	<i>Bảng 6: Mô tả trường địa chỉ</i>	16
7	<i>Bảng 7: Mô tả trường Product</i>	17
8	<i>Bảng 8: Mô tả trường đánh giá</i>	18
9	<i>Bảng 9: Mô tả trường phân loại sản phẩm</i>	19
10	<i>Bảng 10: Mô tả trường đặt hàng</i>	20
11	<i>Bảng 11: Mô tả trường quản lý người dùng của Admin</i>	22
12	<i>Bảng 12: Mô tả trường sản phẩm</i>	23
13	<i>Bảng 13: Mô tả trường danh sách sản phẩm</i>	24
14	<i>Bảng 14: Function User</i>	29
15	<i>Bảng 15: Function Product</i>	30
16	<i>Bảng 16: Các trường quản lý sản phẩm</i>	40
17	<i>Bảng 17: Các trường tạo sản phẩm mới</i>	41
18	<i>Bảng 18: Các trường đăng ký</i>	42
19	<i>Bảng 19: Các trường đăng nhập</i>	43
20	<i>Bảng 20: Các trường thêm địa chỉ</i>	44

21	<i>Bảng 21: Các trường cấp nhát địa chỉ</i>	45
22	<i>Bảng 22: Các trường đánh giá</i>	46

DANH MỤC HÌNH VẼ

Số hiệu	Tên	Trang
1	<i>Hình 1: Wireframe NavBar user</i>	11
2	<i>Hình 2. Wireframe page trang chủ</i>	12
3	<i>Hình 3: Wireframe đăng ký</i>	13
4	<i>Hình 4: Wireframe đăng nhập</i>	14
5	<i>Hình 5: Wireframe Trang thông tin cá nhân</i>	15
6	<i>Hình 6: Wireframe địa chỉ</i>	16
7	<i>Hình 7: Wireframe Product</i>	17
8	<i>Hình 8: Wireframe Ratings</i>	18
9	<i>Hình 9: Wireframe đặt hàng</i>	20
10	<i>Hình 10: Wireframe Trang danh sách tài khoản người dùng</i>	21
11	<i>Hình 11: Wireframe Trang chỉnh sửa tài khoản người dùng</i>	21
12	<i>Hình 12: Wireframe Trang tạo mới sản phẩm</i>	23
13	<i>Hình 13: Wireframe Trang hiển thị danh sách sản phẩm</i>	24
14	<i>Hình 14: Sơ đồ khôi Đăng ký (register)</i>	31
15	<i>Hình 15: Sơ đồ khôi Đăng nhập (login)</i>	31
16	<i>Hình 16: Sơ đồ khôi Cập nhật địa chỉ (updateUserAddress)</i>	32
17	<i>Hình 17: Sơ đồ khôi Thêm địa chỉ mới (addUserAddress)</i>	32
18	<i>Hình 18: Sơ đồ khôi Tạo sản phẩm (createProduct)</i>	33
19	<i>Hình 19: Sơ đồ khôi Chỉnh sửa sản phẩm (updateProduct)</i>	33
20	<i>Hình 20: Sơ đồ khôi Đánh giá (ratings)</i>	34

21	<i>Hình 21: Unit test case list</i>	34
22	<i>Hình 22: Testcase Đăng kí (register)</i>	35
23	<i>Hình 23: Testcase Đăng nhập (login)</i>	35
24	<i>Hình 24: Testcase Thêm địa chỉ mới (addUserAddress)</i>	36
25	<i>Hình 25: Testcase Cập nhật địa chỉ (updateUserAddress)</i>	36
26	<i>Hình 26: Testcase Tạo sản phẩm (createProduct)</i>	37
27	<i>Hình 27: Testcase Chính sửa sản phẩm (updateProduct)</i>	37
28	<i>Hình 28: Testcase Đánh giá (ratings)</i>	38
29	<i>Hình 29: Wireframe quản lý sản phẩm</i>	40
30	<i>Hình 30: Wireframe tạo mới sản phẩm</i>	41
31	<i>Hình 31: Wireframe đăng kí</i>	42
32	<i>Hình 32: Wireframe đăng nhập</i>	43
33	<i>Hình 33: Wireframe Thêm địa chỉ</i>	44
34	<i>Hình 34: Wireframe Cập nhật địa chỉ</i>	45
35	<i>Hình 35: Wireframe Ratings</i>	46
36	<i>Hình 36: Test case Đăng kí</i>	47
37	<i>Hình 37: Test case Đăng nhập</i>	48
38	<i>Hình 38: Test case Thêm địa chỉ</i>	49
39	<i>Hình 39: Test case Cập nhật địa chỉ</i>	50
40	<i>Hình 40: Test case Tạo sản phẩm</i>	51
41	<i>Hình 41: Test case Cập nhật sản phẩm</i>	52
42	<i>Hình 42: Test case Đánh giá</i>	53

43	<i>Hình 43: Biểu đồ hoạt động người dùng Đăng ký</i>	55
44	<i>Hình 44: Biểu đồ hoạt động người dùng Đăng nhập</i>	56
45	<i>Hình 45: Biểu đồ hoạt động cấp nhật địa chỉ</i>	57
46	<i>Hình 46: Biểu đồ hoạt động Thêm địa chỉ</i>	58
47	<i>Hình 47: Biểu đồ hoạt động người quản trị tạo sản phẩm</i>	59
48	<i>Hình 48: Biểu đồ hoạt động người quản trị chỉnh sửa sản phẩm</i>	60
49	<i>Hình 49: Biểu đồ hoạt động người dùng đánh giá sản phẩm</i>	61

LỜI MỞ ĐẦU

Trong thời đại công nghệ số phát triển mạnh mẽ như hiện nay, việc sử dụng các ứng dụng web để mua sắm trực tuyến không chỉ trở nên phổ biến mà còn là một phần không thể thiếu của cuộc sống hàng ngày của chúng ta. Cùng với sự gia tăng của nhu cầu này, việc đảm bảo chất lượng và tính ổn định của các trang web bán hàng trực tuyến là một thách thức đối với các nhà phát triển và nhà kiểm thử phần mềm.

Bài tập lớn môn kiểm thử phần mềm này tập trung vào đề tài kiểm thử một website bán đồ điện tử (e-commerce), một loại hình kinh doanh trực tuyến ngày càng phát triển và phổ biến trong thị trường hiện nay. Trong bối cảnh cạnh tranh khốc liệt và sự mong đợi cao độ từ phía người tiêu dùng, việc đảm bảo chất lượng, tính bảo mật và hiệu suất của các trang web e-commerce là vô cùng quan trọng.

Trong bài tập này, chúng tôi sẽ tiến hành một loạt các phương pháp kiểm thử phần mềm để đảm bảo rằng website bán đồ điện tử được phát triển đáp ứng các yêu cầu chất lượng cần thiết. Từ việc kiểm tra tính năng cơ bản đến kiểm thử bảo mật và hiệu suất, chúng tôi sẽ tập trung vào mọi khía cạnh để đảm bảo rằng trải nghiệm mua sắm trực tuyến của người dùng được tối ưu và an toàn.

Bài tập lớn này không chỉ là cơ hội để áp dụng các kiến thức lý thuyết đã học mà còn là một cơ hội để thử nghiệm và áp dụng các kỹ thuật kiểm thử thực tế trong một môi trường ứng dụng thực tế. Chúng tôi hy vọng rằng kết quả của bài tập này không chỉ mang lại những kiến thức mới mẻ mà còn là một đóng góp quan trọng vào việc nâng cao chất lượng của các ứng dụng web bán hàng trực tuyến trong tương lai.

CHƯƠNG 1: TỔNG QUAN BÀI TOÁN

1. Giới thiệu về tài

Thực trạng: Trong môi trường kinh doanh trực tuyến ngày nay, việc phát triển và duy trì một trang web bán hàng đòi hỏi sự chú ý đặc biệt đến các khía cạnh khác nhau của quản lý sản phẩm, thông tin khách hàng, quản lý đơn hàng và nhiều yếu tố khác. Để xây dựng một website thành công và đáp ứng được nhu cầu của người dùng, việc thu thập và quản lý dữ liệu từ các phần khác nhau của trang web là rất quan trọng.

Phát triển bài toán trong thực tế:

- **Product (Sản phẩm):** Đây là phần chứa thông tin chi tiết về các sản phẩm được bán trên trang web. Mỗi sản phẩm có thể bao gồm các thông tin như tên sản phẩm, mô tả, giá cả, hình ảnh, đánh giá từ người dùng, và các thuộc tính kỹ thuật khác.
- **Product Category (Danh mục sản phẩm):** Để tổ chức sản phẩm một cách có hệ thống, website thường chia sản phẩm thành các danh mục khác nhau. Mỗi danh mục có thể chứa một hoặc nhiều sản phẩm, giúp người dùng dễ dàng tìm kiếm và chọn lựa.
- **Blog:** Phần này cung cấp nội dung thông tin về các bài viết, hướng dẫn, tin tức hoặc những thông điệp quảng cáo liên quan đến các sản phẩm hoặc lĩnh vực kinh doanh của website. Các bài viết blog có thể giúp tăng cường sự quan tâm và tin cậy từ phía khách hàng.
- **Coupon (Phiếu giảm giá):** Thêm các sản phẩm được giảm giá để thu hút người tiêu dùng.
- **Order (Đơn hàng):** Phần này ghi lại thông tin về các đơn hàng được đặt từ người dùng trên trang web. Thông tin này có thể bao gồm các sản phẩm trong đơn hàng, địa chỉ giao hàng, phương thức thanh toán, trạng thái đơn hàng và các thông tin liên quan đến vận chuyển.
- **User (Người dùng):** Để quản lý các hoạt động của người dùng trên trang web, thông tin về người dùng cũng là một phần quan trọng. Điều này bao gồm thông tin đăng nhập, thông tin cá nhân, lịch sử đơn hàng và các hoạt động khác trên trang web.

Với sự phức tạp của các phần trên và mối quan hệ giữa chúng, việc kiểm thử phần mềm trở nên vô cùng quan trọng để đảm bảo tính ổn định, hiệu suất và bảo mật của website bán hàng trực tuyến.

2. Đặc tả yêu cầu

2.1 Các luồng nghiệp vụ chính

Đối với người dùng:

- Đăng kí, đăng nhập tài khoản truy cập website.
- Xem các danh mục sản phẩm và các thông tin cửa hàng chi tiết ở Trang chủ, có thể thêm sản phẩm vào giỏ hàng sau khi đăng nhập vào hệ thống.
- Tìm kiếm các sản phẩm theo từ khóa hoặc giá thành, màu sắc, brand.
- Xem các blog liên quan đến thông tin các sản phẩm điện tử đa dạng.
- Đặt hàng.

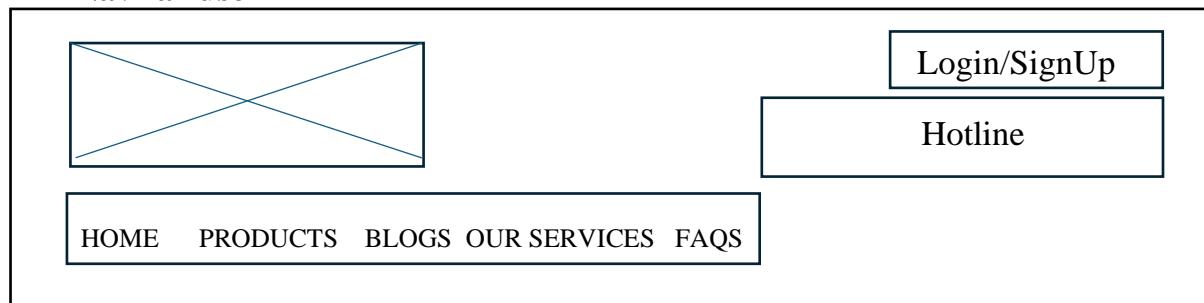
Đối với quản trị viên:

- Đăng nhập bằng tài khoản dành cho quản trị viên.
- Quản lý thông tin tài khoản (Thêm, sửa, xóa thông tin tài khoản).
- Quản lý sản phẩm, danh mục sản phẩm (Thêm, sửa, xóa sản phẩm).
- Quản lý bài viết.(Thêm, sửa, xóa bài viết).
- Quản lý các đơn đặt hàng(Thêm, sửa, xóa đơn đặt hàng).

2.2 Các màn hình chức năng chính

Wireframe

• NavBar user

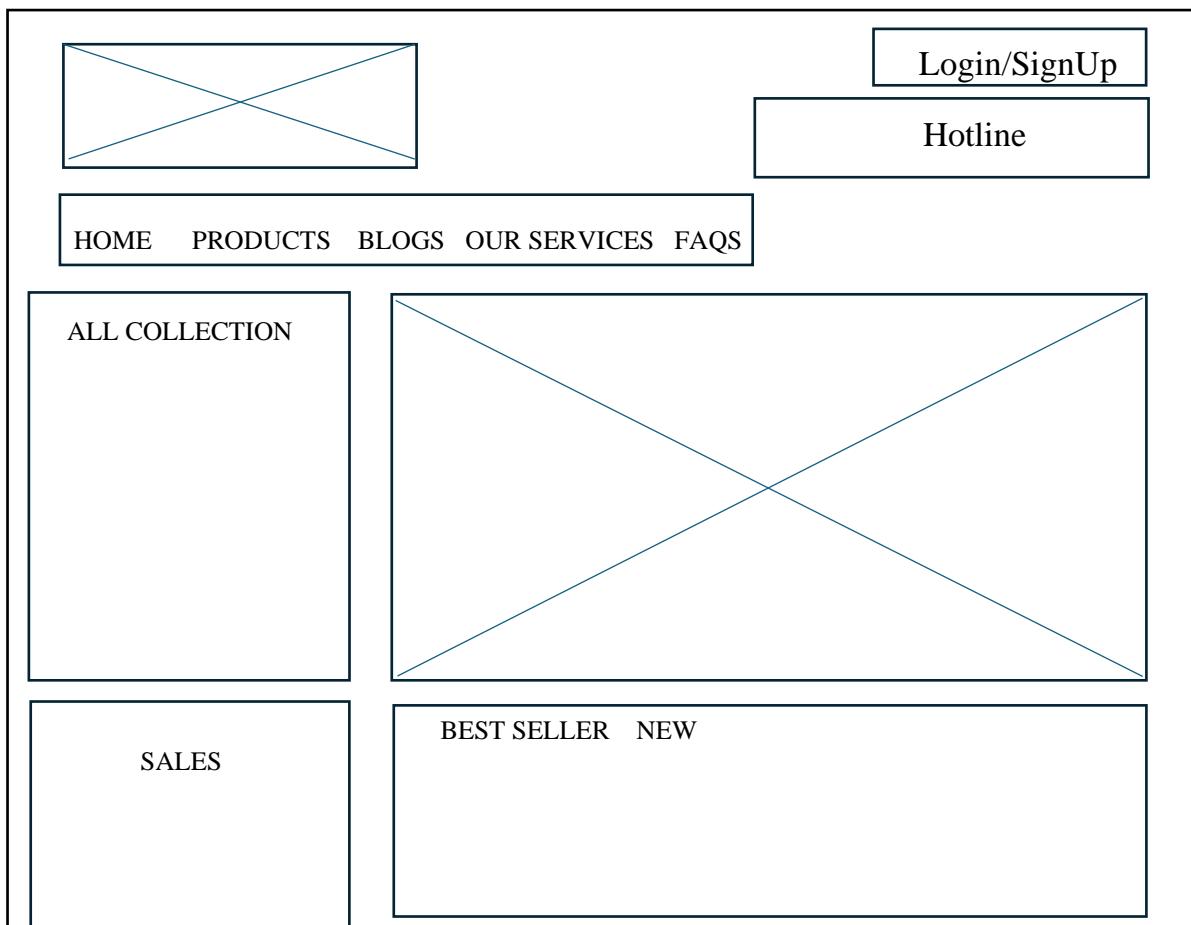


Hình 1: Wireframe NavBar user

Bảng 1: Mô tả phần tử NavBar user

Tên phần tử	Loại	Mô tả
Logo	Image	Logo nhận diện
Home	Button	Điều hướng đến trang chủ
Products	Button	Điều hướng đến trang sản phẩm
Blogs	Button	Điều hướng đến trang bài viết về sản phẩm
Our services	Button	Điều hướng đến chăm sóc khách hàng
Login/SignUp	Button	Hiển thị form Đăng nhập/ Đăng ký

- HOME (Trang chủ): Hiển thị các danh mục sản phẩm bán chạy, sản phẩm mới, collection, sản phẩm đang sales.



Hình 2. Wireframe page trang chủ

Bảng 2: Mô tả trường danh mục sản phẩm

Tên Trường	Loại	Kiểu Dữ Liệu	Bắt Buộc	Độ Dài Tối Đa	Mô tả
Title	Label	String	Có	100	Tên loại sản phẩm
Slug	Label	String	Có	100	URL
Brand	Label	Array	Có	50	Loại sản phẩm
Image	Label	String	Có	50	Ảnh sản phẩm
Icon	Label	String	Có	50	Icon minh họa
Timestamps	Label	Date	Có		Thời gian tạo

Điều kiện ràng buộc: Các trường title, slug, brand, image, icon, timestamps không được bỏ trống.

- Trang đăng kí

SIGN UP

Firstname Lastname

Email

Phone

Password

Sign Up

Hình 3: Wireframe đăng kí

Bảng 3: Mô tả trường đăng kí

Trường	Loại	Kiểu dữ liệu	Độ dài tối đa	Bắt buộc	Mô tả
Firstname	Input	String	30	Có	Tên
Lastname	Input	String	30	Có	Tên đệm
Email	Input	String	50	Có	Địa chỉ email
Mobile	Input	String	20	Có	Số điện thoại
Password	Input	String	50	Có	Mật khẩu

Điều kiện ràng buộc: Email phải được nhập đúng để nhận mã xác minh tài khoản. Các trường firstname, lastname, email, mobile, password không được trống. Mobile độ dài tối thiểu là 9.

- Trang đăng nhập

The wireframe shows a login form with the following components:

- A large square placeholder icon containing a large black 'X'.
- A title "XIN CHAO" centered above the input fields.
- An "Email" input field.
- An "Password" input field.
- A "Forgot account?" link.
- A "Create account" link.
- A "Login" button at the bottom right.

Hình 4: Wireframe đăng nhập

Bảng 4: Mô tả trường đăng nhập

Trường	Loại	Kiểu dữ liệu	Độ dài tối đa	Bắt buộc	Mô tả
Email	Input	String	50	Có	Địa chỉ email
Password	Input	String	50	Có	Mật khẩu

Điều kiện ràng buộc: Các trường email, password không được bỏ trống.

- Trang thông tin cá nhân: bao gồm thông tin cá nhân đã có khi tạo tài khoản, ngoài ra còn có thêm địa chỉ, lịch sử giao dịch, các sản phẩm yêu thích.

The wireframe illustrates the layout of the personal information page. On the left, there is a placeholder for an 'AVATAR'. On the right, under the heading 'Personal Info', there are several input fields: 'First Name' and 'Last Name' in separate boxes; 'Email' and 'Phone' in adjacent boxes; a long horizontal box for 'Address'; and a button labeled 'Upload AVT' at the bottom.

Hình 5: Wireframe Trang thông tin cá nhân

Bảng 5: Mô tả trường thông tin cá nhân

Trường	Loại	Kiểu dữ liệu	Độ dài tối đa	Bắt buộc	Mô tả
Firstname	Label	String	30	Có	Tên
Lastname	Label	String	30	Có	Tên đệm
Avatar	Input	String	30	Không	Ảnh người dùng
Email	Label	String	50	Có	Địa chỉ email
Mobile	Label	String	20	Có	Số điện thoại

Điều kiện ràng buộc: Các trường firstname, lastname, avatar, email, mobile không được bỏ trống.

- Form địa chỉ: người dùng có thể thêm địa chỉ khi tiến hành đặt 1 đơn hàng.

The wireframe shows a rectangular form titled "Add new shipping address". At the top are two rounded rectangular input fields: "Name" on the left and "Phone" on the right. Below them is a larger input field for "City/District/Town". Underneath that is another large input field for "Street Name, Building, House". At the bottom left is a checkbox labeled "Set as default address". On the bottom right is a rounded rectangular button labeled "Create".

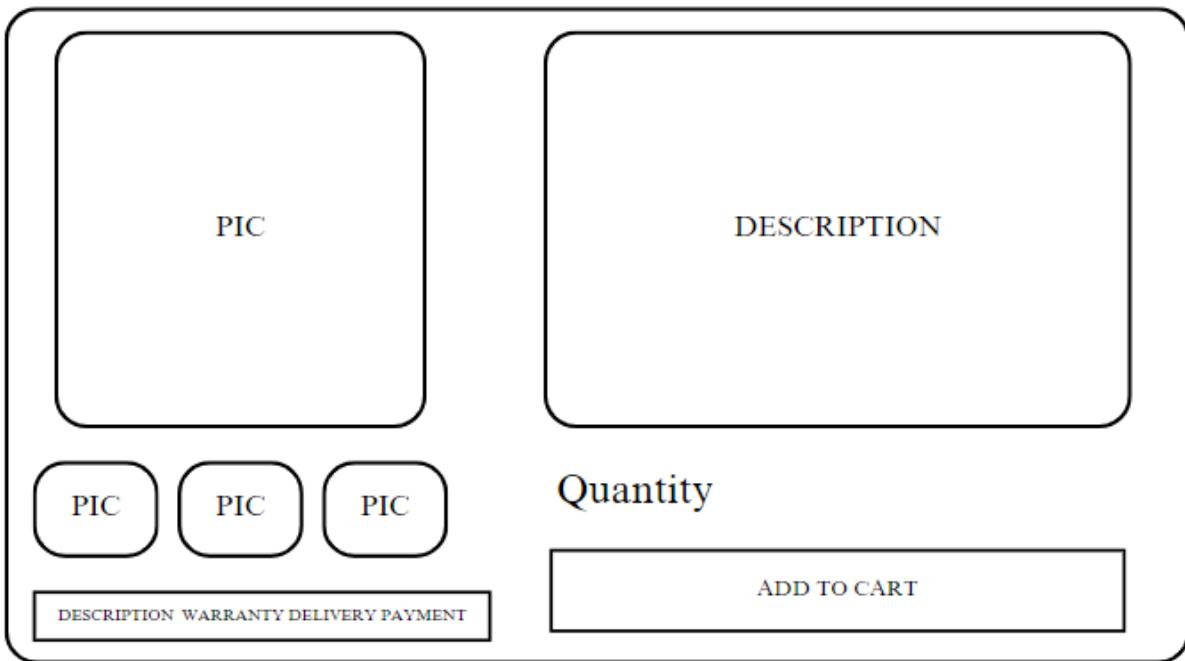
Hình 6: Wireframe địa chỉ

Bảng 6: Mô tả trường địa chỉ

Trường	Loại	Kiểu dữ liệu	Độ dài tối đa	Bắt buộc	Mô tả
Name	Input	String	30	Có	Tên
Phone	Input	String	20	Có	Số điện thoại
City	Input	String	50	Có	Thành phố
District	Input	String	50	Có	Huyện
Ward	Input	String	50	Có	Phường
AddressDetail	Input	String	100	Có	Địa chỉ chi tiết
defaultAddress	Input	Boolean		Không	Đặt làm địa chỉ mặc định

Điều kiện ràng buộc: Các trường name, phone, city, district, ward, addressdetail, defaultaddress không được bỏ trống. Phone tối thiểu 9 kí tự.

- Trang Product: Người dùng có thể tìm kiếm sản phẩm theo từ khóa hoặc giá thành theo nhu cầu.



Hình 7: Wireframe Product

Bảng 7: Mô tả trường Product

Trường	Loại	Kiểu dữ liệu	Độ dài tối đa	Bắt buộc	Mô tả
Title	Label	String	50	Có	Tên sản phẩm
Description	Label	Array		Có	Mô tả
Branch	Label	String	50	Có	Nhà sản xuất
Price	Label	Number	50	Có	Giá
Category	Label	String	100	Có	Phân loại
Quantity	Label	Number		Có	Số lượng sản phẩm
Thumb	Label	String	200	Có	Thông tin tổng quan về sản phẩm
Images	Label	Array		Có	Hình ảnh sản phẩm
Color	Label	String	50	Có	Màu sắc
TotalRating	Label	Number		Có	Tổng lượt đánh giá

Điều kiện ràng buộc: Các trường title, description, branch, price, category, quantity, thumb, images, color, totalrating không được bỏ trống.

- Trường Rating: Đánh giá sản phẩm.

The wireframe shows a rounded rectangular form with a title 'Đánh giá' at the top. Below the title is a large rectangular input field with the placeholder 'Comment'. Underneath the input field are five star icons arranged horizontally. The first four stars are solid black, while the fifth star is grayed out. At the bottom center of the form is a rectangular 'Submit' button.

Hình 8: Wireframe Ratings

Bảng 8: Mô tả trường đánh giá

Trường	Loại	Kiểu dữ liệu	Độ dài tối đa	Bắt buộc	Mô tả
Star	Input	Number	50	Có	Sao bình chọn
PostedBy	Label	String	20	Có	Đánh giá bởi người dùng
Comment	Input	String	100	Có	Bình luận
UpdatedAt	Label	Date	50	Có	Thời gian đánh giá

Điều kiện ràng buộc: Các trường star, postedby, comment, updatedat không được bỏ trống.

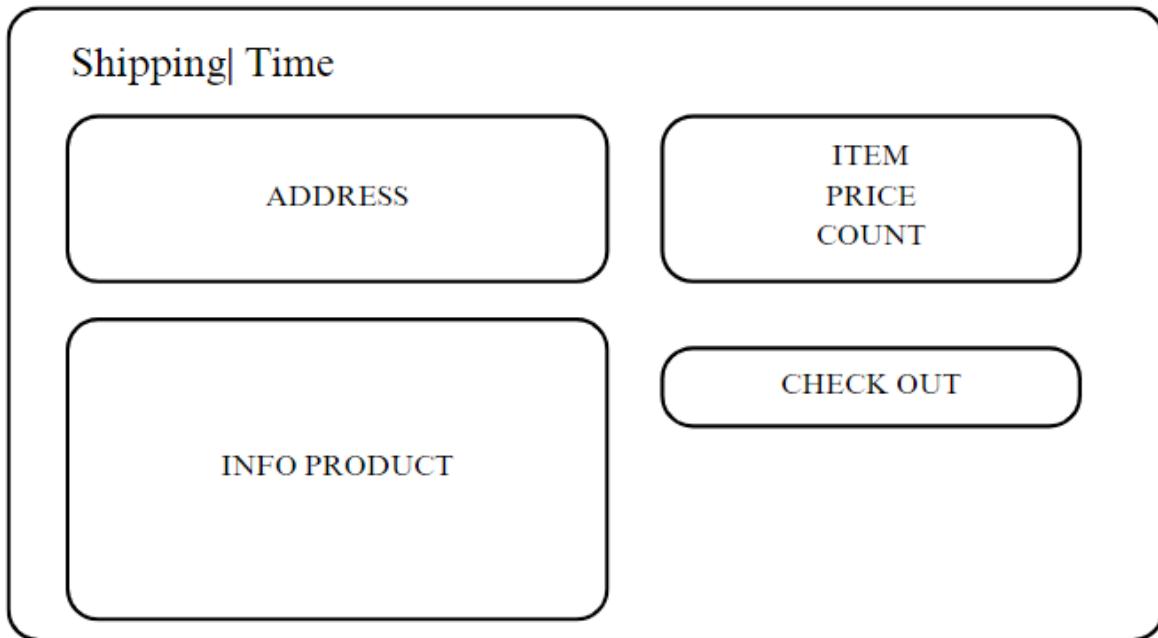
- Variants: Phân loại sản phẩm

Bảng 9: Mô tả trường phân loại sản phẩm

Trường	Loại	Kiểu dữ liệu	Độ dài tối đa	Bắt buộc	Mô tả
Title	Label	String	50	Có	Tên sản phẩm
Color	Label	String	20	Có	Màu
Price	Label	Number		Có	Giá
Thumb	Label	String	200	Có	Thông tin
Images	Label	Array		Có	Hình ảnh sản phẩm
SKU	Label	String	100	Có	Mã SKU riêng biệt của từng sản phẩm

Điều kiện ràng buộc: Các trường title, color, price, thumb, images không được bỏ trống.

- Trang đặt hàng: Tùy chỉnh địa chỉ, thêm hoặc xóa sản phẩm. Tính tổng giá đơn thanh toán.



Hình 9: Wireframe đặt hàng

Bảng 10: Mô tả trường đặt hàng

Trường	Loại	Kiểu dữ liệu	Độ dài tối đa	Bắt buộc	Mô tả
Products	Label			Có	Toàn bộ thông tin sản phẩm
Status	Label	String		Có	Trạng thái
OrderBy	Label			Có	Toàn bộ thông tin người dùng
Address	Label	String	200	Có	Địa chỉ
Timestamps	Label	Date		Có	Thời gian tạo

Điều kiện ràng buộc: Các trường products, status, orderby, address, timestamps không được bỏ trống.

- Trang quản lí người dùng của Admin.

The wireframe shows a user management interface. On the left, a vertical sidebar has the text "ADMIN WORKPLACE". The main area features a header with a logo placeholder and a search bar. Below the header is a table titled "Danh sách User" (User List) with the following columns: No, FullName, Email, Role, Phone, Status, CreateAt, and Action. The table contains three rows of data:

No	FullName	Email	Role	Phone	Status	CreateAt	Action
1	Hoàng Nhi	nhisinh@gmail.com	Admin	0964018832	Active	16/12/2023	
2	Văn Đức	duy@gmail.com	Admin	096406218	Active	16/12/2023	
3	Thúy Hồng	thuyne@gmail.com	User	0329323121	Block	16/12/2023	

Hình 10: Wireframe Trang danh sách tài khoản người dùng

The wireframe shows a form for editing a user account. It includes fields for "First Name" and "Last Name", "Email" and "Phone", and dropdown menus for "Role: Admin or User" and "Status: Blocked or Active". A central "Update" button is at the bottom.

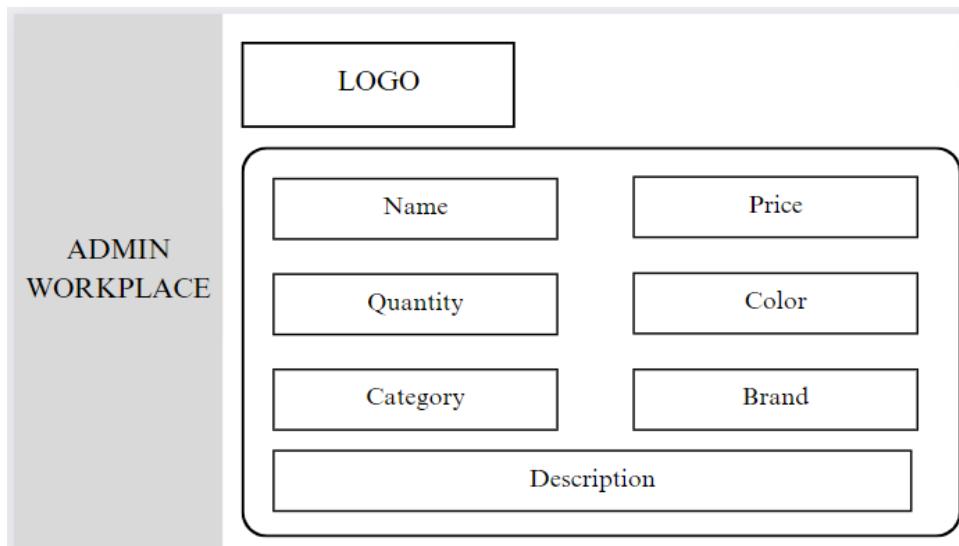
Hình 11: Wireframe Trang chỉnh sửa tài khoản người dùng

Bảng 11: Mô tả trường quản lý người dùng của Admin

Trường	Loại	Kiểu dữ liệu	Độ dài tối đa	Bắt buộc	Mô tả
Firstname	Label	String	30	Có	Tên
Lastname	Label	String	30	Có	Tên đệm
Email	Label	String	50	Có	Địa chỉ email
Phone	Label	String	20	Có	Số điện thoại
Status	Label	Integer		Có	Trạng thái
Role	Label	Integer		Có	Vị trí

Điều kiện ràng buộc: Các trường firstname, lastname, email, phone, status, role không được bỏ trống.

- Trang quản lý sản phẩm:

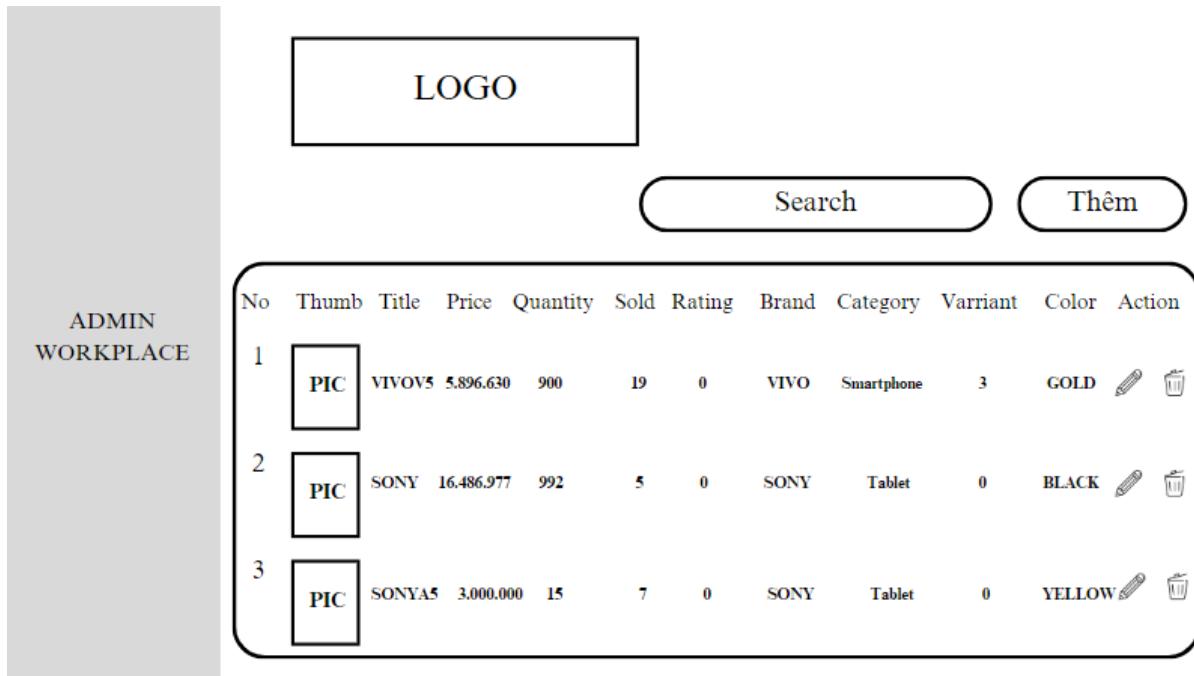


Hình 12: Wireframe Trang tạo mới sản phẩm

Bảng 12: Mô tả trường sản phẩm

Trường	Loại	Kiểu dữ liệu	Độ dài tối đa	Bắt buộc	Mô tả
Name	Input	String	50	Có	Tên sản phẩm
Slug	Input	String	20	Có	URL
Description	Input	Array		Có	Mô tả
Brand	Input	String	50	Có	Nhà sản xuất
Price	Input	Number	50	Có	Giá
Category	Input	String	100	Có	Phân loại
Quantity	Input	Number		Có	Số lượng sản phẩm
Thumbnails	Input	String	200	Có	Hình ảnh đại diện
Slider	Input	String	100	Có	Hình ảnh sản phẩm chi tiết
Color	Input	String	50	Có	Màu sắc

Điều kiện ràng buộc: Các trường name, slug, description, brand, price, category, quantity không được bỏ trống.



Hình 13: Wireframe Trang hiển thị danh sách sản phẩm

Bảng 13: Mô tả trường danh sách sản phẩm

Trường	Loại	Kiểu dữ liệu	Độ dài tối đa	Bắt buộc	Mô tả
Name	Label	String	50	Có	Tên sản phẩm
Slug	Label	String	20	Có	URL
Description	Label	Array		Có	Mô tả
Brand	Label	String	50	Có	Nhà sản xuất
Price	Label	Number	50	Có	Giá
Category	Label	String	100	Có	Phân loại
Quantity	Label	Number		Có	Số lượng sản phẩm
Sold	Label	Number		Không	Số sản phẩm sold out
Thumb	Label	String	200	Có	Thông tin tổng quan về sản phẩm
Images	Label	Array		Có	Hình ảnh sản phẩm
Color	Label	String	50	Có	Màu sắc
TotalRating	Label	Number		Có	Tổng lượt đánh giá
Timestamps	Label	Datetime		Có	Thời gian tạo

Điều kiện ràng buộc: Các trường name, slug, description, brand, price, category, quantity, sold, thumb, images, color, totalrating, timestamps không được bỏ trống.

2.3 Các module chính

- Đăng kí:

Input: Nhập First Name, Last Name, Phone, Email, Password.

Output: True/False

Mô tả chức năng: Đăng kí tài khoản người dùng.

- Đăng nhập:

Input: Nhập Name, Password

Output: True/False

Mô tả chức năng: Đăng nhập vào trang web để đặt hàng, thêm sản phẩm yêu thích, đánh giá sản phẩm (với người dùng). Quản lý sản phẩm, tài khoản, blogs, orders (với admin).

- User:

Input: Name

Output: Toàn bộ thông tin người dùng, sản phẩm yêu thích, địa chỉ.

Mô tả chức năng: Chính sửa thông tin, địa chỉ, xem sản phẩm yêu thích.

- Sản phẩm.

Input: Mã sản phẩm.

Output: Tên sản phẩm, hình ảnh, màu sắc, giá, thông tin sản phẩm chi tiết.

Mô tả chức năng: Tìm kiếm, thêm, sửa, xóa sản phẩm theo danh mục, giá thành, theo từ khóa.

- Đặt hàng và thanh toán.

Input: Thông tin user, địa chỉ, sản phẩm, tài khoản thanh toán.

Output: Hóa đơn.

Mô tả chức năng: Người dùng thực hiện thêm sản phẩm vào giỏ hàng, thiết lập các thông tin gửi hàng, số lượng mua,... sau đó thanh toán bằng thẻ.

- Module Mã giảm giá:

Input: Tên, Giá giảm, Thời hạn

Output: Thông tin mã giảm giá và thông tin sản phẩm hiển thị form riêng.

Mô tả chức năng: Thêm, sửa, xóa mã giảm giá.

3. Phân công công việc

Class Name	Function Name	Function Code(Optional)	Sheet Name	Description
user.js	Đăng kí	register	register	HongThuy
user.js	Đăng nhập	login	login	HongThuy
user.js	Cập nhật địa chỉ	updateUserAddress	updateUserAddress	HoangNhi
user.js	Thêm địa chỉ mới	addUserAddress	addUserAddress	HoangNhi
product.js	Tạo sản phẩm	createProduct	createProduct	Duc-Cute
product.js	Chỉnh sửa sản phẩm	updateProduct	updateProduct	Duc-Cute
product.js	Đánh giá	ratings	ratings	Duc-Cute

CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ TEST

1. Kiểm thử đơn vị (Unit Testing)

1.1 Kiểm thử đơn vị là gì?

- Kiểm thử đơn vị được định nghĩa là một loại kiểm thử phần mềm trong đó các đơn vị hay thành phần riêng lẻ của phần mềm được kiểm thử.
- Kiểm thử đơn vị được thực hiện trong quá trình phát triển (coding) ứng dụng. Mục tiêu của Kiểm thử đơn vị là cô lập một phần code và xác minh tính chính xác của đơn vị đó. Trong lập trình thủ tục, một đơn vị có thể là một chức năng hoặc thủ tục riêng lẻ. Kiểm thử đơn vị thường được thực hiện bởi developer.
- Trong quy trình phát triển phần mềm, quy trình kiểm thử phần mềm, Mô hình chữ V, Kiểm thử đơn vị là cấp độ kiểm thử đầu tiên được thực hiện trước khi kiểm thử tích hợp. Kiểm thử đơn vị là một kỹ thuật WhiteBox testing, thường được thực hiện bởi developer. Tuy nhiên trong một thực tế, do thời gian bị giới hạn hoặc yêu cầu của các developer, các kỹ sư QA cũng thực hiện kiểm thử đơn vị.

1.2 Phương pháp, kĩ thuật của kiểm thử đơn vị

- **Phương pháp**

- Kiểm thử hộp trắng (White-box testing): Trong phương pháp này, kiểm thử viên có kiến thức về cấu trúc nội bộ của mã nguồn. Họ tạo ra các ca kiểm thử dựa trên các đường dẫn, điều kiện và cấu trúc nội bộ của mã nguồn.
- Kiểm thử hộp đen (Black-box testing): Kiểm thử viên chỉ quan tâm đến các kết quả của chức năng hoặc phương thức, không quan tâm đến cấu trúc nội bộ của mã nguồn. Họ tạo ra các ca kiểm thử dựa trên các yêu cầu và mô tả chức năng.
- Kiểm thử hợp lý (Equivalence partitioning): Phương pháp này chia tập hợp các giá trị đầu vào thành các lớp tương đương và sau đó chỉ chọn một số giá trị đại diện từ mỗi lớp để kiểm tra.
- Kiểm thử giới hạn (Boundary testing): Kiểm tra biên của dải giá trị đầu vào, bao gồm giá trị biên và giá trị lớn hơn/giá trị nhỏ hơn giới hạn.
- Kiểm thử một lần (One-time testing): Kiểm tra mỗi chức năng, phương thức chỉ một lần để đảm bảo tính chính xác.
- Kiểm thử lặp lại (Repetition testing): Kiểm tra chức năng hoặc phương thức nhiều lần để đảm bảo tính ổn định và độ tin cậy của chúng.
- Kiểm thử đồng thời (Concurrent testing): Kiểm tra cùng lúc nhiều chức năng hoặc phương thức để kiểm tra tính tương tác giữa chúng.

- **Kỹ thuật**

- Kiểm thử dòng điều kiện (Branch Coverage Testing): Kiểm tra mỗi nhánh (branch) của mã nguồn ít nhất một lần. Điều này đảm bảo rằng tất cả các điều kiện đã được kiểm tra.
- Kiểm thử dòng điều kiện biên (Boundary Coverage Testing): Kiểm tra các giá trị biên của dữ liệu đầu vào để đảm bảo phần mềm hoạt động đúng trong các giới hạn cụ thể.
- Kiểm thử dòng dữ liệu (Data Flow Testing): Tập trung vào luồng dữ liệu qua các biến trong mã nguồn để đảm bảo rằng các giá trị được xử lý đúng cách.
- Kiểm thử bảng quyết định (Decision Table Testing): Sử dụng các bảng quyết định để kiểm tra tất cả các kịch bản có thể xảy ra và đảm bảo rằng phần mềm xử lý chúng đúng cách.
- Kiểm thử biến thể (Mutation Testing): Tạo ra các biến thể của mã nguồn bằng cách thay đổi các dòng mã và kiểm tra xem các ca kiểm thử có phát hiện ra lỗi hay không.
- Kiểm thử hướng điều kiện (Condition Coverage Testing): Đảm bảo rằng mỗi điều kiện logic trong các câu lệnh if hoặc loop đã được kiểm tra.
- Kiểm thử kiểm soát luồng (Control Flow Testing): Kiểm tra các luồng điều khiển trong chương trình để đảm bảo tất cả các điểm đích và các luồng con đã được kiểm tra.
- Kiểm thử gọi hàm (Stub Testing): Kiểm tra các hàm phụ thuộc bên ngoài bằng cách sử dụng các đối tượng giả lập (stubs) để kiểm tra các tương tác và kết quả trả về.

1.3 Mô tả các đối tượng

Bảng 14: Function User

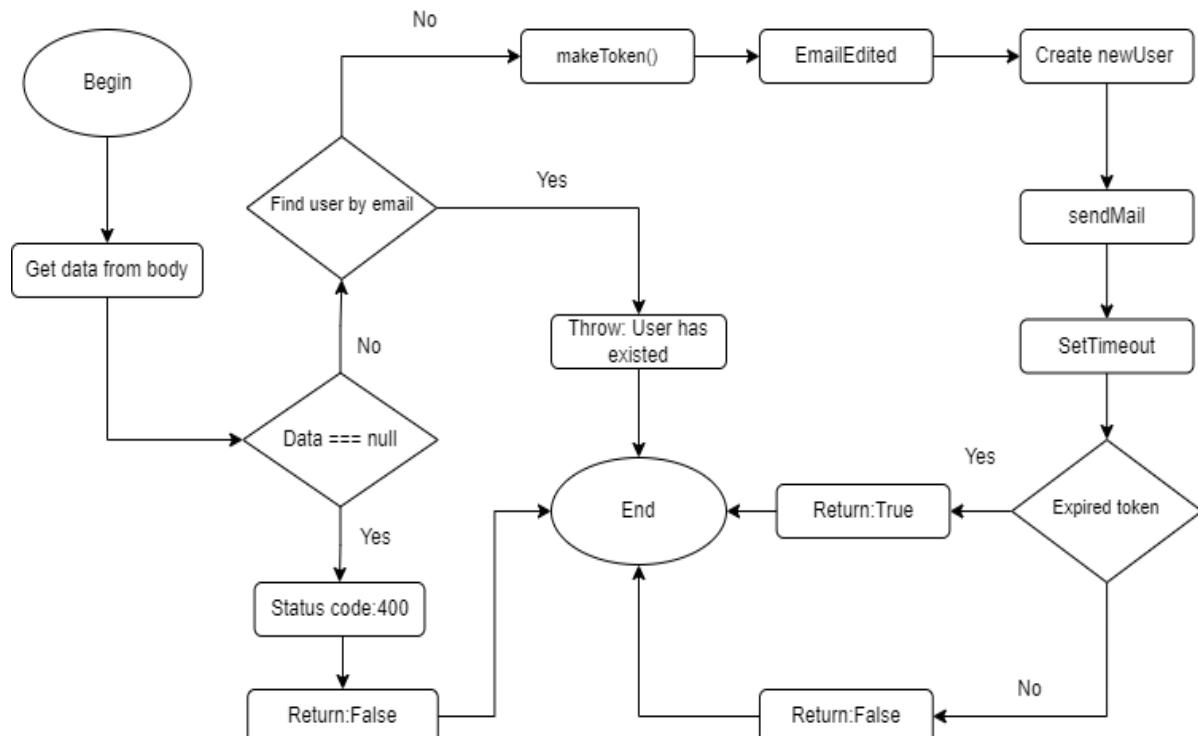
Function	Input	Output	Description
addUserAddress	<ul style="list-style-type: none"> • _id(string) • name(string) • phone(string) • city(string) • district(string) • ward(string) • addressDetail (string) • defaultAddress (boolean) 	Object	<p>Chức năng thêm địa chỉ User. Nếu các thông tin về địa chỉ không tồn tại sẽ hiển thị thông báo lỗi. Nếu không, nó sẽ tìm kiếm người dùng thông qua _id và cập nhật địa chỉ.</p> <p>Kết quả trả về là thông báo thêm thành công(True) hoặc lỗi(False)</p>
updateUserAddress	<ul style="list-style-type: none"> • _id(string) • city(string) • district(string) • ward(string) • addressDetail (string) • defaultAddress (boolean) 	Object	<p>Chức năng cập nhật, lưu trữ thông tin địa chỉ người dùng. Kết quả trả về thông báo cập nhật thành công(True) hoặc thông báo lỗi(False)</p>
login	email:string password:string	Object	<p>Hàm này xử lý quá trình đăng nhập của người dùng bằng cách kiểm tra thông tin đăng nhập, tạo và cập nhật token, trả về thông tin người dùng cùng với token truy cập mới cho việc xác thực.</p> <p>Kết quả trả về đúng(True) sẽ lưu vào userdata hoặc hiển thị thông báo lỗi (False)</p>
register	<ul style="list-style-type: none"> • email:string • password:string • firstname:string • lastname:string • mobile:string 	Object	<p>Hàm thực hiện đăng ký tài khoản người dùng. Nếu giá trị nhập vào lỗi sẽ hiển thị thông báo.</p> <p>Tìm kiếm trong FindOne thông qua email xem tài khoản đã được lập chưa. Nếu chưa sẽ tạo lập tài khoản mới. Trả về kết quả là thông báo kiểm tra code email(True) hoặc có lỗi xảy ra(False)</p>

Bảng 15: Function Product

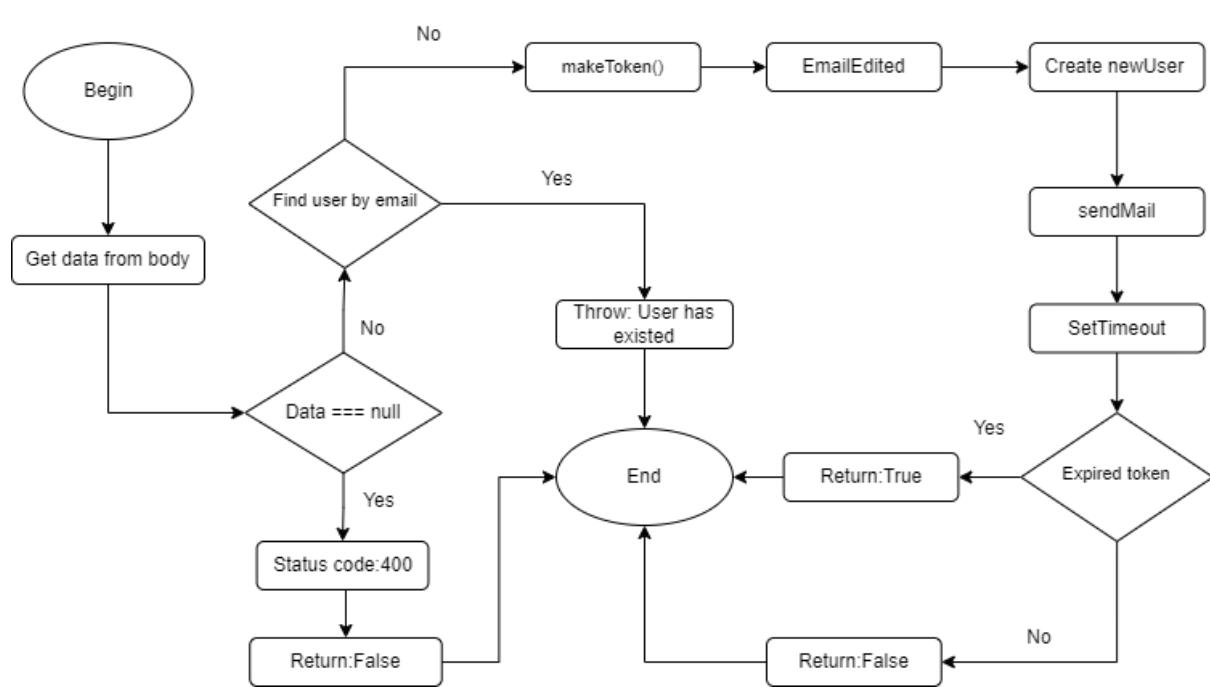
Function	Input	Output	Description
createProduct	<ul style="list-style-type: none"> • title:string • price:string • description:string • branch:string category:string • color:string • quantity:string • thumb:string • images:string 	Object	Chức năng tạo sản phẩm. Hàm này nhập vào thông tin để kiểm tra xem nó đã tồn tại chưa. Nếu chưa sẽ tiếp tục tạo mới và lưu trữ các dữ liệu. Kết quả trả ra đã thêm vào lưu trữ(True) hoặc hiển thị thông báo lỗi(False)
updateProduct	<ul style="list-style-type: none"> • pid:string 	Object	Chức năng cập nhật sản phẩm. từ id sản phẩm tìm ra thông tin sản phẩm, nếu sửa đổi thì dữ liệu được lưu trữ. Kết quả trả về thông tin được cập nhật thành công(True) hoặc hiển thị thông báo lỗi(False)
ratings	<ul style="list-style-type: none"> • _id:string • star:string • comment:text • pid:string • updateAt:NativeDate 	Object	Chức năng đánh giá sản phẩm. nếu star hoặc sản phẩm trống hiển thị thông báo lỗi. Nếu không sẽ tìm kiếm thông tin sản phẩm, lấy thông tin người đánh giá, thời gian. Hiển thị kết quả bên dưới thông tin sản phẩm trên trang web.

1.4 Phương thức

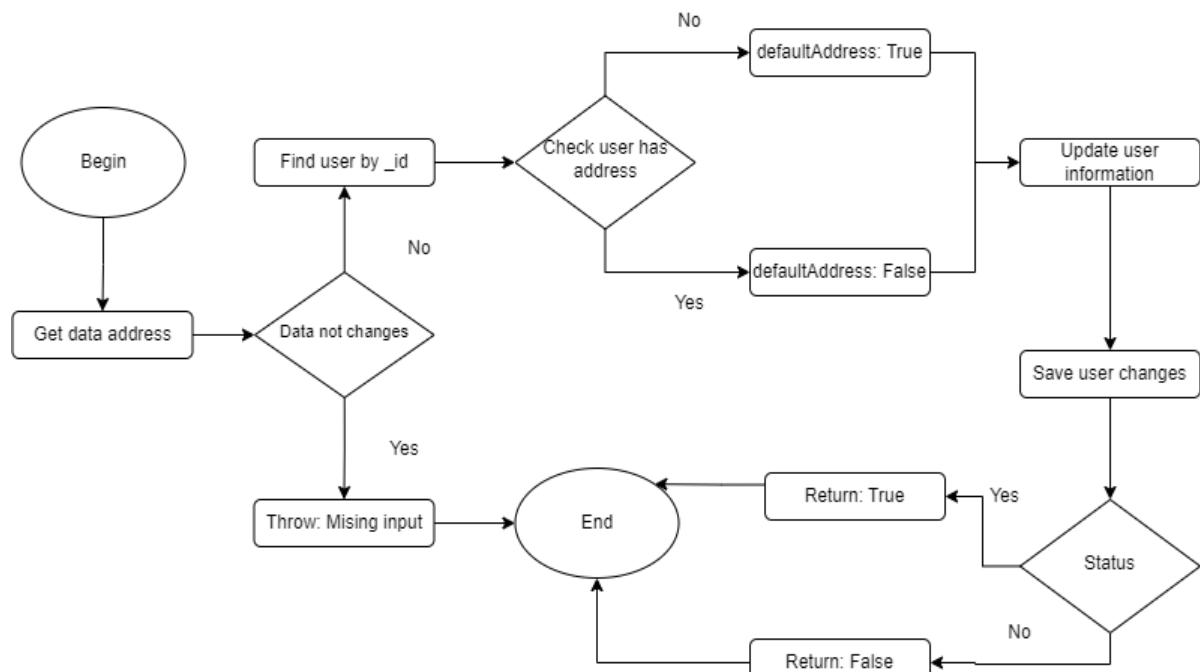
Sơ đồ khối Đăng kí (register)



Sơ đồ khối Đăng nhập (login)

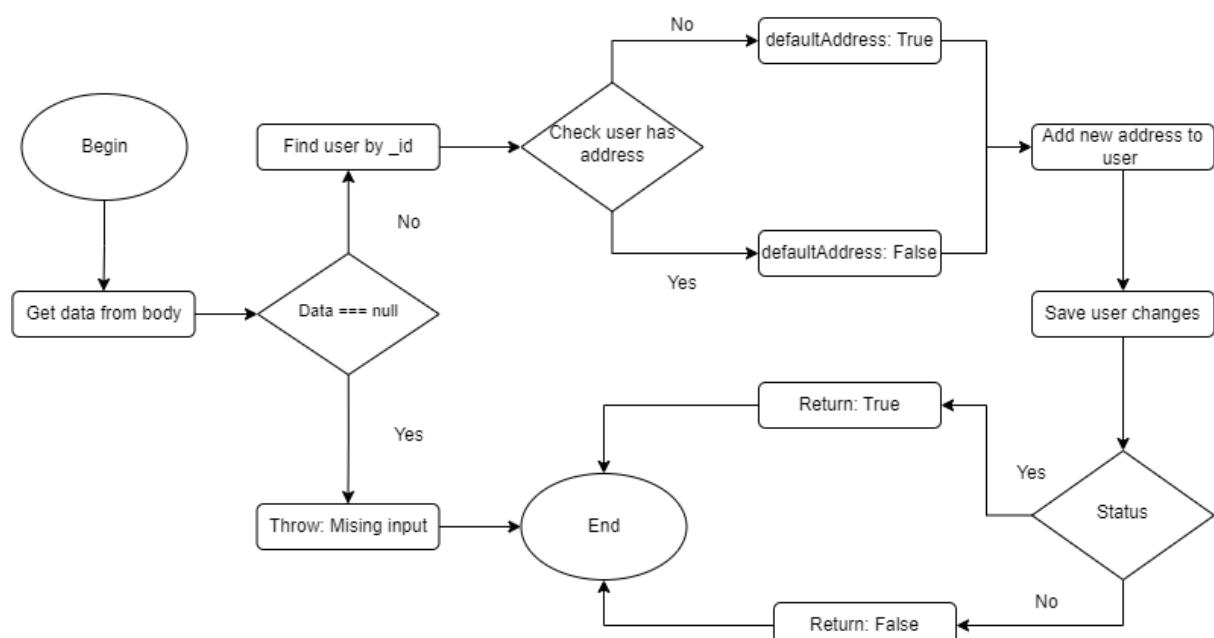


Sơ đồ khối Cập nhật địa chỉ (updateUserAddress)



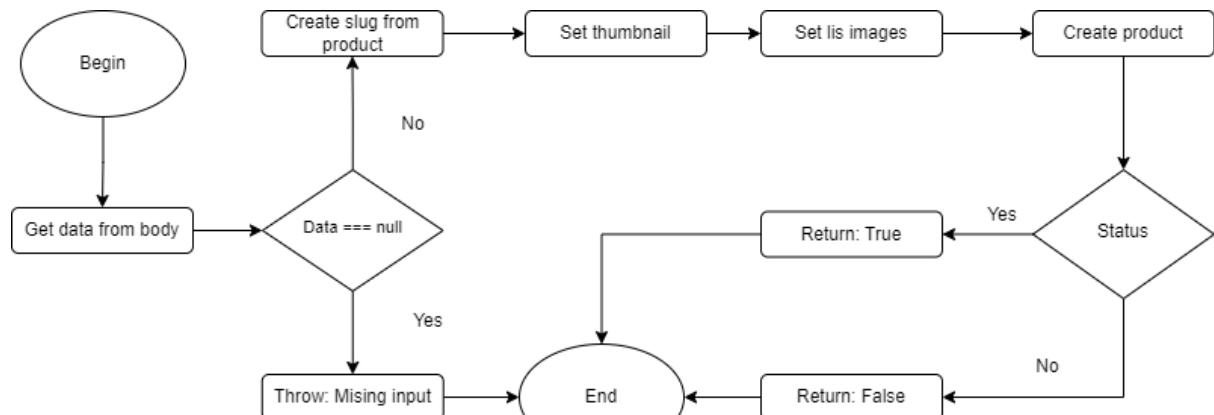
Hình 16: Sơ đồ khối Cập nhật địa chỉ (updateUserAddress)

Sơ đồ khối Thêm địa chỉ mới (addUserAddress)



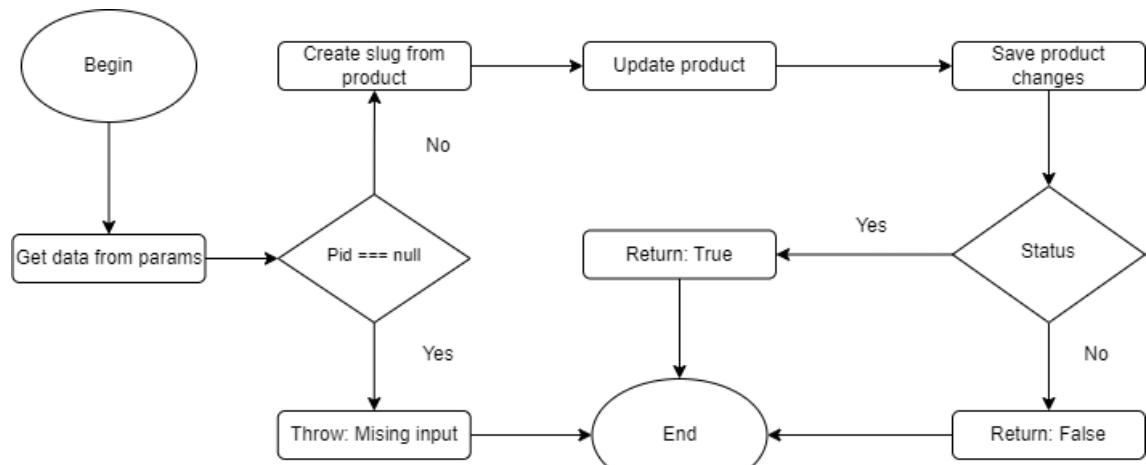
Hình 17: Sơ đồ khối Thêm địa chỉ mới (addUserAddress)

Sơ đồ khối Tạo sản phẩm (createProduct)



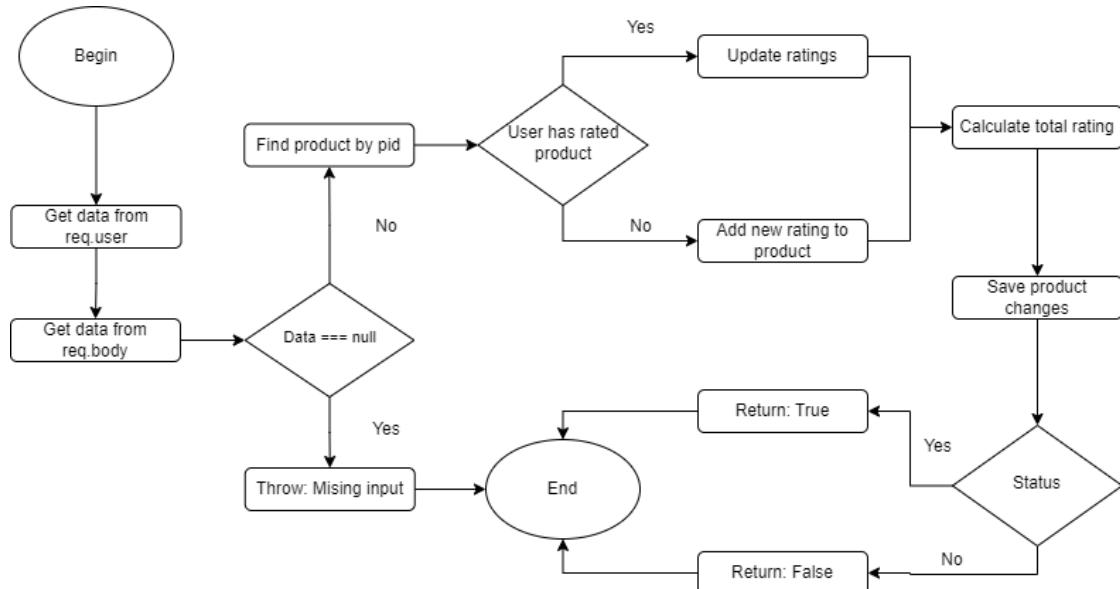
Hình 18: Sơ đồ khối Tạo sản phẩm (createProduct)

Sơ đồ khối Chính sửa sản phẩm (updateProduct)



Hình 19: Sơ đồ khối Chính sửa sản phẩm (updateProduct)

Sơ đồ khối Đánh giá (ratings)



Hình 20: Sơ đồ khối Đánh giá (ratings)

1.5 Unit test case list

- Unit test case list

UNIT TEST CASE LIST

Project Name	Unit Test Website E-Commerce						
Project Code	E-Commerce-server						
Normal number of Test cases/KLOC							
Test Environment Setup Description	<List environment requires in this system>						
No	Requirement Name	Class Name	Function Name	Function Code(Optional)	Sheet Name	Description	Pre-Condition
1	user.js	Đăng kí	register	register	HongThuy	N/A	
2	user.js	Đăng nhập	login	login	HongThuy	N/A	
3	user.js	Cập nhật địa chỉ	updateUserAddress	updateUserAddress	HoangNhi	N/A	
4	user.js	Thêm địa chỉ mới	addUserAddress	addUserAddress	HoangNhi	N/A	
5	product.js	Tạo sản phẩm	createProduct	createProduct	Duc-Cute	N/A	
6	product.js	Chỉnh sửa sản phẩm	updateProduct	updateProduct	Duc-Cute	N/A	
7	product.js	Đánh giá	ratings	ratings	Duc-Cute	N/A	

Hình 21: Unit test case list

▪ Testcase Đăng kí (register)

Function Code	register	Function Name	Đăng kí															
Created By	HongThuy	Executed By	HongThuy															
Lines of code	40	Lack of test cases																
Test requirement	<Brief description about requirements which are tested in this function>																	
Passed	33	Failed	0	Untested	33	N/A/B	1	32	0	Total Test Cases	33							
			URCID01	URCID02	URCID03	URCID04	URCID05	URCID06	URCID07	URCID08	URCID09	URCID10	URCID11	URCID12	URCID13	URCID14	URCID15	URCID16
Condition	Precondition																	
	req.body																	
	Lỗi email	null																
	Lỗi password	null																
	Lỗi firstname	null																
	Lỗi lastname	null																
	Lỗi mobile	null																
	Gửi email đã tồn tại	20211841@eput.edu.vn																
Confirm	Return																	
	TRUE	O																
	FALSE	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
	Something went wrong.																	
	Please check your email to active account	O																
Exception																		
Log message																		
	User has existed																	
	Missing input	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
Result	Type(N : Normal, A : Abnormal, B : Boundary)	N	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
	Passed/Failed	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
	Executed Date																	
	Defect ID																	

Hình 22: Testcase Đăng kí (register)

▪ Testcase Đăng nhập (login)

Function Code	login	Function Name	Đăng nhập															
Created By	HongThuy	Executed By	HongThuy															
Lines of code	31	Lack of test cases																
Test requirement	<Brief description about requirements which are tested in this function>																	
Passed	6	Failed	0	Untested	6	N/A/B	1	5	0	Total Test Cases	6							
			URCID01	URCID02	URCID03	URCID04	URCID05	URCID06										
Condition	Precondition																	
	req.body																	
	Email trống	null																
	Password trống	null																
	Email không tồn tại	nhisinhb123@gmail.com																
	Password không đúng	12345																
Confirm	Return																	
	TRUE	O																
	FALSE	O	O	O	O	O												
Exception																		
Log message																		
	Invalid credentials!																	
	Missing input	O	O	O														
Result	Type(N : Normal, A : Abnormal, B : Boundary)	N	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
	Passed/Failed	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
	Executed Date																	
	Defect ID																	

Hình 23: Testcase Đăng nhập (login)

▪ Testcase Thêm địa chỉ mới (addUserAddress)

Function Code	addUserAddress	Function Name	Thêm địa chỉ người dùng					
Created By	HoangNhi	Executed By	HoangNhi					
Lines of code	46	Lack of test cases						
Test requirement	<Brief description about requirements which are tested in this function>							
Passed		Failed	Untested	N/A/B	Total Test Cases			
64		0	64	1 63 0	64			

Condition	req.body	Chinh sửa Name	Văn Đức	O	req.body	Chinh sửa Phone	95656223	O
	Chinh sửa City	HCM	O		Chinh sửa District	Quận Bình Thành	O	
	Chinh sửa Ward	P11	O		Chinh sửa AddressDetail	Canh XX	O	
	Chinh sửa DefaulAddress	Tick	O		Không chỉnh sửa thông tin	null	O	
Confirm	Return	TRUE	O	O O O O O O	FALSE	O	O O O O O O	
	Update Address is successfully!				Something went wrong!			
Exception								
Log message	Missing inputs						O	
Result	Type(N : Normal, A : Abnormal, B : Boundary)	N	N	N	N	N	N	A
	Passed/Failed	P	P	P	P	P	P	P
	Executed Date							
	Defect ID							

Hình 24: Testcase Thêm địa chỉ mới (addUserAddress)

▪ Testcase Cập nhật địa chỉ (updateUserAddress)

Function Code	updateUserAddress	Function Name	Cập nhật địa chỉ					
Created By	HoangNhi	Executed By	HoangNhi					
Lines of code	18	Lack of test cases						
Test requirement	<Brief description about requirements which are tested in this function>							
Passed		Failed	Untested	N/A/B	Total Test Cases			
8		0	8	7 1 0	8			
Condition	req.body	Chinh sửa Name	Văn Đức	O	req.body	Chinh sửa Phone	95656223	O
	Chinh sửa City	HCM	O		Chinh sửa District	Quận Bình Thành	O	
	Chinh sửa Ward	P11	O		Chinh sửa AddressDetail	Canh XX	O	
	Chinh sửa DefaulAddress	Tick	O		Không chỉnh sửa thông tin	null	O	
Confirm	Return	TRUE	O	O O O O O O	FALSE	O	O O O O O O	
	Update Address is successfully!				Something went wrong!			
Exception								
Log message	Missing inputs						O	
Result	Type(N : Normal, A : Abnormal, B : Boundary)	N	N	N	N	N	N	A
	Passed/Failed	P	P	P	P	P	P	P
	Executed Date							
	Defect ID							

Hình 25: Testcase Cập nhật địa chỉ (updateUserAddress)

- Testcase Tao sản phẩm (createProduct)

Function Code	<i>createProduct</i>	Function Name	Tạo sản phẩm		
Created By	Duc-Cute	Executed By	Duc-Cute		
Lines of code	14	Lack of test cases			
Test requirement		<Brief description about requirements which are tested in this function>			
Passed		Failed	Untested	N/A/B	Total Test Cases
64		0	64	1 63 0	64

Condition	req.body	Chinh sửa Title	Chinh sửa Price	Chinh sửa Description	Chinh sửa Branch	Chinh sửa Category	Chinh sửa Color	Chinh sửa Quantity	Không chỉnh sửa thông tin	Missing input	Exception	Log message	Result
Return	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	Success - Normal, A : Abnormal, B : Boundary
Confirm	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	Passed/Failed
Exception													Defect ID
Log message													UTCID01 UTCID02 UTCID03 UTCID04 UTCID05 UTCID06 UTCID07 UTCID08
Result													

Hình 26: Testcase Tạo sản phẩm (createProduct)

- Testcase Chính sửa sản phẩm (updateProduct)

Function Code	<i>updateProduct</i>	Function Name	Cập nhật sản phẩm		
Created By	Duc-Cute	Executed By	Duc-Cute		
Lines of code	19	Lack of test cases			
Test requirement		<Brief description about requirements which are tested in this function>			
Passed		Failed	Untested	N/A/B	Total Test Cases
8		0	8	7 1 0	8

Condition	req.body	Chinh sửa Title	Chinh sửa Price	Chinh sửa Description	Chinh sửa Branch	Chinh sửa Category	Chinh sửa Color	Chinh sửa Quantity	Không chỉnh sửa thông tin	Missing inputs	Exception	Log message	Result
Return	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	Success - Normal, A : Abnormal, B : Boundary
Confirm	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	Passed/Failed
Exception													Defect ID
Log message													UTCID01 UTCID02 UTCID03 UTCID04 UTCID05 UTCID06 UTCID07 UTCID08
Result													

Hình 27: Testcase Chính sửa sản phẩm (updateProduct)

- Testcase Đánh giá (ratings)

Function Code	<i>ratings</i>	Function Name	Đánh giá					
Created By	Duc-Cute	Executed By	Duc-Cute					
Lines of code	45	Lack of test cases	Lack of test cases					
Test requirement	<Brief description about requirements which are tested in this function>							
Passed	8	Failed	0	Untested	8	N/A/B	1	Total Test Cases 8

			URC1D01	URC1D02	URC1D03	URC1D04	URC1D05	URC1D06	URC1D07	URC1D08
Condition	req.body									
	Thiếu star						0	0	0	0
	Thiếu comment					0	0		0	0
	Thiếu uid		0	0	0	0	0	0	0	
Confirm		TRUE	0							
		FALSE	0	0	0	0	0	0	0	
	Exception									
	Log message		Missing inputs	0	0	0	0	0	0	
Result	Type(N : Normal, A : Abnormal, B : Boundary)		N	A	A	A	A	A	A	
	Passed/Failed		P	P	P	P	P	P	P	
	Executed Date									
	Defect ID									

Hình 28: Testcase Đánh giá (ratings)

2. Kiểm thử tích hợp (Intergration Testing)

2.1 Kiểm thử tích hợp là gì?

- Kiểm thử tích hợp (Integration Testing) là quá trình kiểm tra và đảm bảo rằng các thành phần khác nhau của một hệ thống hoạt động chính xác khi được kết hợp với nhau. Trong kiểm thử tích hợp, các thành phần riêng lẻ đã được kiểm thử ở cấp độ đơn vị được kết hợp lại với nhau và được kiểm tra xem chúng hoạt động đúng cách khi tương tác với nhau.
- Mục tiêu của kiểm thử tích hợp là phát hiện và giải quyết các lỗi liên quan đến sự tương tác giữa các thành phần của hệ thống, như là việc truyền dữ liệu, gọi các chức năng từ các module khác nhau, và xử lý lỗi kết nối. Điều này đảm bảo rằng toàn bộ hệ thống hoạt động một cách tron tru và đáp ứng được các yêu cầu và kỳ vọng của người dùng cuối.
- Các phương pháp kiểm thử tích hợp có thể bao gồm kiểm thử hạng mục (top-down integration testing), kiểm thử hạng mục ngược (bottom-up integration testing), kiểm thử hạng mục kết hợp (combined integration testing), và kiểm thử hạng mục lớn (big bang integration testing), phụ thuộc vào cách các thành phần của hệ thống được kết hợp và kiểm tra.

2.2 Phương pháp, kỹ thuật của kiểm thử tích hợp

- **Phương pháp**

- **Top-down Testing (Kiểm thử từ trên xuống)**: Bắt đầu kiểm thử từ thành phần cấp cao nhất của hệ thống và tiếp tục xuống dần đến các thành phần cấp thấp hơn.
- **Bottom-up Testing (Kiểm thử từ dưới lên)**: Bắt đầu kiểm thử từ các thành phần cấp thấp nhất của hệ thống và tiếp tục lên trên.
- **Big Bang Integration Testing (Kiểm thử tích hợp Big Bang)**: Tất cả các thành phần của hệ thống được tích hợp lại với nhau cùng một lúc và sau đó kiểm thử.
- **Incremental Testing (Kiểm thử tăng dần)**: Hệ thống được xây dựng và kiểm thử theo từng phần nhỏ, sau đó các phần nhỏ này được tích hợp lại và kiểm thử lại.
- **Sandwich Testing (Kiểm thử Sandwich)**: Kết hợp cả kiểm thử từ trên xuống và từ dưới lên.
- **Thread Testing (Kiểm thử Luồng)**: Các luồng chức năng hoặc giao diện được kiểm thử độc lập, sau đó tích hợp lại với nhau để kiểm tra tích hợp.
- **Continuous Integration Testing (Kiểm thử tích hợp liên tục)**: Tích hợp và kiểm thử tự động liên tục trong quá trình phát triển, thường thông qua các công cụ và quy trình tự động hóa.

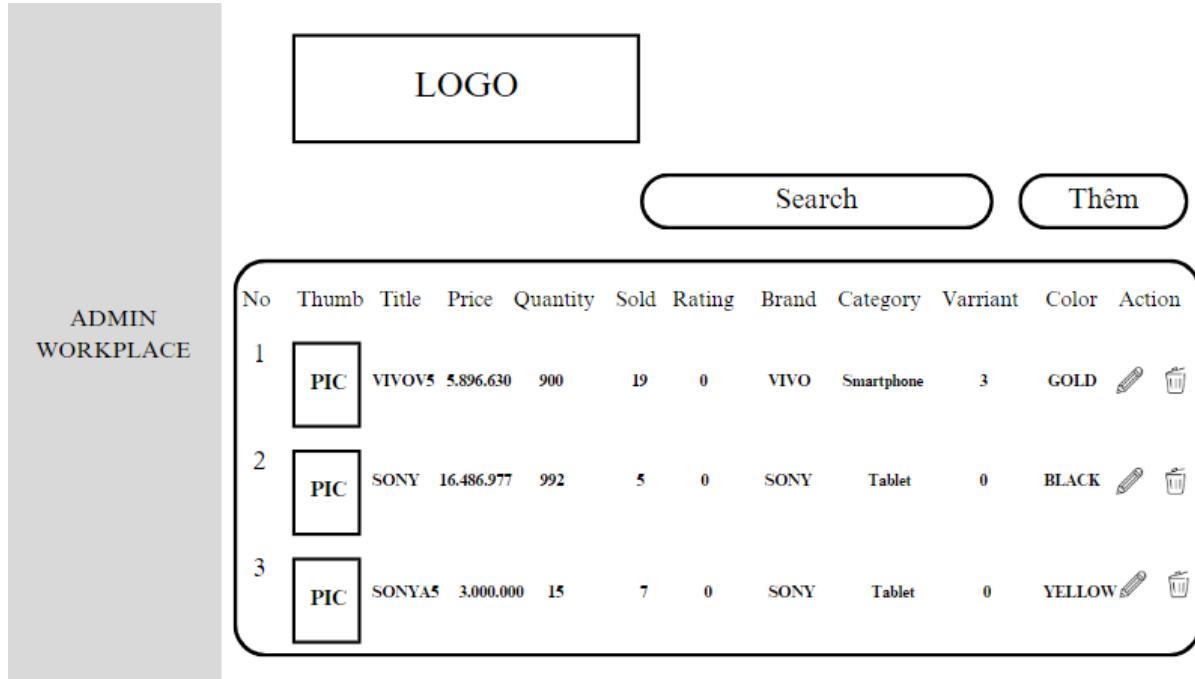
- **Kỹ thuật**

- **Kiểm thử giao diện (Interface Testing)**: Kiểm tra giao diện giữa các thành phần hoặc module để đảm bảo thông tin được truyền đúng cách và tương tác làm việc như mong đợi.
- **Kiểm thử dữ liệu (Data Testing)**: Kiểm tra dữ liệu được truyền giữa các thành phần để đảm bảo tính toàn vẹn và độ chính xác của dữ liệu.
- **Kiểm thử hoạt động (Functionality Testing)**: Kiểm tra tính hoạt động của các chức năng tích hợp trong hệ thống để đảm bảo chúng hoạt động như dự kiến khi tích hợp với nhau.
- **Kiểm thử hiệu suất (Performance Testing)**: Kiểm tra hiệu suất của hệ thống sau khi tích hợp để đảm bảo rằng nó đáp ứng được yêu cầu về tải và thời gian phản hồi.
- **Kiểm thử bảo mật (Security Testing)**: Kiểm tra các lỗ hổng bảo mật có thể tồn tại sau khi tích hợp các thành phần lại với nhau.

2.3 Trang quản trị

❖ Trang quản trị sản phẩm

- Trang quản lý sản phẩm

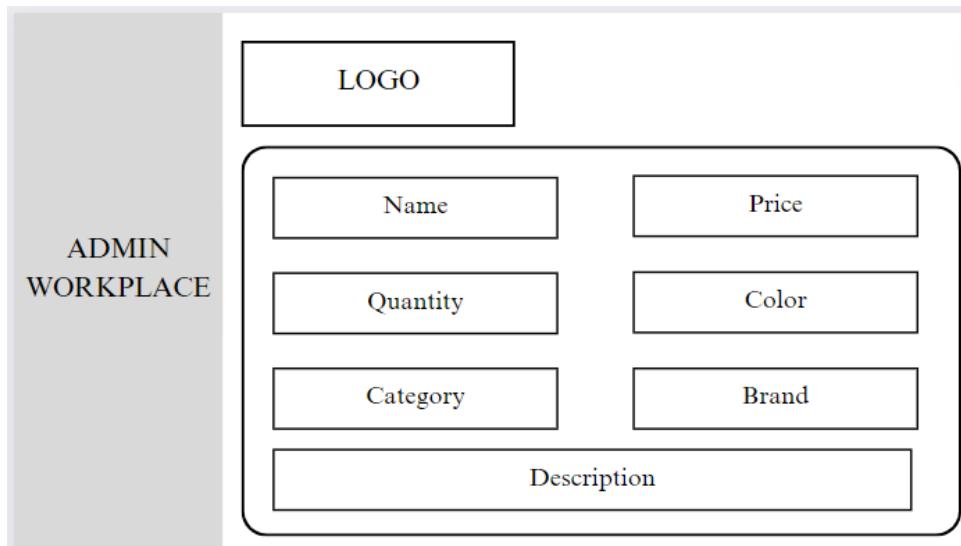


Hình 29: Wireframe quản lý sản phẩm

Bảng 16: Các trường quản lý sản phẩm

STT	Tên trường	Độ dài	Kiểu dữ liệu	Bắt buộc	Ràng buộc
1	No	1-225	String	Có	-
2	Thumb	1-225	String	Có	Không được để trống
3	Title	1-225	String	Có	Không được để trống
4	Price		Number	Có	Không được để trống
5	Quantity		Number	Có	Không được để trống
6	Sold		Number	Có	Không được để trống
7	Rating		Number	Không	-
8	Brand	1-225	String	Có	Không được để trống
9	Category		Integer	Có	Không được để trống
10	Varriant	1-225	String	Có	Không được để trống
11	Color	1-225	String	Có	Không được để trống

- Trang tạo mới sản phẩm



Hình 30: Wireframe tạo mới sản phẩm

Bảng 17: Các trường tạo sản phẩm mới

STT	Tên trường	Độ dài	Kiểu dữ liệu	Bắt buộc	Ràng buộc
1	Name	1-225	String	Có	Không được để trống
2	Price		Number	Có	Không được để trống
3	Quantity		Number	Có	Không được để trống
4	Brand	1-225	String	Có	Không được để trống
5	Category		Integer	Có	Không được để trống
6	Color	1-225	String	Có	Không được để trống
7	Description	1-225	String	Có	Không được để trống

❖ Trang quản trị người dùng

- Đăng ký

The wireframe shows a sign-up interface with the title "SIGN UP" at the top. On the left is a large square input field containing a large black "X". To the right are four smaller input fields labeled "Firstname", "Lastname", "Email", and "Phone" respectively. Below these is a larger input field labeled "Password". At the bottom is a "Sign Up" button.

Hình 31: Wireframe đăng ký

Bảng 18: Các trường đăng ký

STT	Tên trường	Độ dài	Kiểu dữ liệu	Bắt buộc	Ràng buộc
1	Firstname	1-225	String	Có	Không được để trống
2	Lastname	1-225	String	Có	Không được để trống
3	Email	1-50	String	Có	Không được để trống
5	Phone	1-10	Number	Có	Không được để trống
6	Password	1-100	String	Có	Không được để trống

- Đăng nhập

The wireframe shows a login interface with the following components:

- XIN CHAO**: The main title at the top center.
- Email**: A text input field for entering an email address.
- Password**: A text input field for entering a password.
- Forgot account?**: A link to reset a forgotten account.
- Create account**: A link to register a new account.
- Login**: A large button at the bottom right to submit the form.

Hình 32: Wireframe đăng nhập

Bảng 19: Các trường đăng nhập

Trường	Loại	Kiểu dữ liệu	Độ dài tối đa	Bắt buộc	Mô tả	Ràng buộc
Email	Input	String	50	Có	Địa chỉ email	Không được để trống
Password	Input	String	50	Có	Mật khẩu	Không được để trống

- Thêm địa chỉ

Add new shipping address

Name	Phone
<hr/>	
City/District/Town	
<hr/>	
Street Name, Building, House	
<hr/>	
<input type="checkbox"/> Set as default address	Create

Hình 33: Wireframe Thêm địa chỉ

Bảng 20: Các trường thêm địa chỉ

STT	Tên trường	Độ dài	Kiểu dữ liệu	Bắt buộc	Ràng buộc
1	Name	1-225	String	Có	Không được để trống
2	Phone	1-10	Number	Có	Không được để trống
3	City/District/Town		Integer	Có	Không được để trống
5	Description	1-255	String	Có	Không được để trống
6	Default address		Integer	Không	-

- Cập nhật địa chỉ

Add new shipping address

Hoàng Nhí	0926381331
Thành Phố Hà Nội, Quận Hoàn Kiếm, Đồng Xuân	
Ngõ 12	
<input type="checkbox"/> Set as default address	
Cancel	Update

Hình 34: Wireframe Cập nhật địa chỉ

Bảng 21: Các trường cập nhật địa chỉ

STT	Tên trường	Độ dài	Kiểu dữ liệu	Bắt buộc	Ràng buộc
1	Name	1-225	String	Có	Không được để trống
2	Phone	1-10	Number	Có	Không được để trống
3	City/District/Town		Integer	Có	Không được để trống
5	Description	1-255	String	Có	Không được để trống
6	Default address		Integer	Không	-

- Đánh giá

Đánh giá

Comment

Submit

Hình 35: Wireframe Ratings

Bảng 22: Các trường đánh giá

STT	Tên trường	Độ dài	Kiểu dữ liệu	Bắt buộc	Ràng buộc
1	Star		Integer	Có	Không được để trống
2	Cmt	1-10	String	Có	Không được để trống

2.4 Integration Testcase

- Đăng ký

Function code	register		
Created By	HongThuy		

No	Test Requirement ID	Test requirement description	Test Type	Importance	Number of TCs
1	TR register 01	Hệ thống cập nhật thành	Positive	High	1
2	TR register 02	Hệ thống sẽ thông báo "Missing input" khi người dùng để trống 1 hoặc nhiều trường dữ liệu bắt kỳ.	Negative	Medium	31
3	TR register 03	Hệ thống hiện thông báo "User has existed" khi dữ liệu email nhập vào đã tồn tại trong cơ sở dữ liệu.	Negative	Medium	1

No	Test Requirement ID	Test Objective	Pre-condition	Step	Step action	Test data	Expected result
1	TR register 01	Hệ thống sẽ thông báo "Please check your email to active account" khi các trường dữ liệu được nhập đầy đủ và đúng.	Truy cập thành công trang web Ecommerce-DevHouse	1	Truy cập trang chủ trang web Ecommerce-DevHouse	https://e-commerce-client-theta-seven.vercel.app/	Hiển thị giao diện trang chủ web Ecommerce-DevHouse
				2	Chọn Sign In or Create Account		Hiển thị trang đăng nhập bao gồm tùy chọn đăng kí (Create Account)
				3	Chọn Create Account		Màn hình hiển thị trang đăng kí
				4	Nhập đầy đủ và đúng các trường dữ liệu	firstname: Hồng; lastname: Thúy; email: 20211920@eaut.edu.vn; phone: 093256762; password:123456	
				5	Nhấn nút Sign Up		Hệ thống lưu thông tin đăng kí và hiển thị thông báo "Please check you email to active account"
2	TR register 02	Hệ thống sẽ thông báo "Missing input" khi người dùng để trống các trường dữ liệu bắt kỳ.	Truy cập thành công trang web Ecommerce-DevHouse	1	Truy cập trang chủ trang web Ecommerce-DevHouse	https://e-commerce-client-theta-seven.vercel.app/	Hiển thị giao diện trang chủ web Ecommerce-DevHouse
				2	Chọn Sign In or Create Account		Hiển thị trang đăng nhập bao gồm tùy chọn đăng kí (Create Account)
				3	Chọn Create Account		Màn hình hiển thị trang đăng kí
				4	Nhập đủ thông tin các trường firstname, lastname, email, phone, password	firstname: Hồng; lastname: Thúy; email: 20211920@eaut.edu.vn; phone: 093256762; password:123456	Trang hiển thị các trường đã nhập
				5	Xóa dữ liệu của một hoặc nhiều trường bắt kỳ		
				6	Nhấn nút Sign Up		Hệ thống hiển thị thông báo "Missing input"
3	TR register 03	Hệ thống hiện thông báo "User has existed" khi dữ liệu email nhập vào đã tồn tại trong cơ sở dữ liệu.	Truy cập thành công trang web Ecommerce-DevHouse	1	Truy cập trang chủ trang web Ecommerce-DevHouse	https://e-commerce-client-theta-seven.vercel.app/	Hiển thị giao diện trang chủ web Ecommerce-DevHouse
				2	Chọn Sign In or Create Account		Hiển thị trang đăng nhập bao gồm tùy chọn đăng kí (Create Account)
				3	Chọn Create Account		Màn hình hiển thị trang đăng kí
				4	Nhập đủ thông tin các trường firstname, lastname, email (trùng với email đã tồn tại trong cơ sở dữ liệu), phone, password	firstname: Hồng; lastname: Thúy; email: 20211841@eaut.edu.vn; phone: 093256762; password:123456	
				5	Nhấn nút Sign Up		Hệ thống hiển thị thông báo lỗi "User has existed"

Hình 36: Test case Đăng ký

▪ Đăng nhập

No	Test Requirement ID	Test requirement description	Test Type	Importance	Number of TCs
1	TR_login_01	Người dùng đăng nhập thành công vào trang web Ecommerce-DevHouse	Positive	High	1
2	TR_login_02	Hệ thống sẽ thông báo "Missing input" khi người dùng để trống các trường dữ liệu bất kỳ.	Negative	Medium	3
3	TR_login_03	Hệ thống hiển thị thông báo "Invalid credentials!" khi dữ liệu email nhập vào không tồn tại trong cơ sở dữ liệu	Negative	Medium	1
4	TR_login_04	Hệ thống hiển thị thông báo "Invalid credentials!" khi dữ liệu password nhập vào không đúng.	Negative	Medium	1

No	Test Requirement ID	Test Objective	Pre-condition	Step	Step action	Test data	Expected result
1	TR_login_01	Người dùng đăng nhập thành công vào trang web Ecommerce-DevHouse	Truy cập trang web Ecommerce-DevHouse	1	Truy cập trang chủ trang web Ecommerce-DevHouse	https://e-commerce-client-theta-seven.vercel.app/	Hiển thị giao diện trang chủ web Ecommerce-DevHouse
				2	Chọn Sign In or Create Account		Hiển thị trang đăng nhập
				4	Nhập đầy đủ và đúng các trường dữ liệu email, password	email: nhisinh@gmail.com; password: 123456	
				5	Nhấn nút Login		Hệ thống hiển thị trang chủ trang web đã đăng nhập tài khoản.
2	TR_login_02	Hệ thống sẽ thông báo "Missing input" khi người dùng để trống các trường dữ liệu bất kỳ.	Truy cập trang web Ecommerce-DevHouse	1	Truy cập trang chủ trang web Ecommerce-DevHouse	https://e-commerce-client-theta-seven.vercel.app/	Hiển thị giao diện trang chủ web Ecommerce-DevHouse
				2	Chọn Sign In or Create Account		Hiển thị trang đăng nhập
				4	Nhập đầy đủ và đúng các trường dữ liệu email, password	email: nhisinh@gmail.com; password: 12345678	
				5	Xóa dữ liệu của một trường bất kỳ hoặc cả hai trường.		
				6	Nhấn nút Login		Hiển thị thông báo lỗi "Missing input"
3	TR_login_03	Hệ thống hiển thị thông báo "Invalid credentials!" khi dữ liệu email nhập vào không tồn tại trong cơ sở dữ liệu.	Truy cập trang web Ecommerce-DevHouse	1	Truy cập trang chủ trang web Ecommerce-DevHouse	https://e-commerce-client-theta-seven.vercel.app/	Hiển thị giao diện trang chủ web Ecommerce-DevHouse
				2	Chọn Sign In or Create Account		Hiển thị trang đăng nhập
				4	Nhập email, password nhưng email không tồn tại trong cơ sở dữ liệu.	email: nhisinh123@gmail.com; password: 12345	
				5	Nhấn nút Login		Hiển thị thông báo lỗi "Invalid credentials!"
4	TR_login_04	Hệ thống hiển thị thông báo "Invalid credentials!" khi dữ liệu password nhập vào không đúng.	Truy cập trang web Ecommerce-DevHouse	1	Truy cập trang chủ trang web Ecommerce-DevHouse	https://e-commerce-client-theta-seven.vercel.app/	Hiển thị giao diện trang chủ web Ecommerce-DevHouse
				2	Chọn Sign In or Create Account		Hiển thị trang đăng nhập
				4	Nhập email đúng, password không đúng.	email: nhisinh@gmail.com; password: 12345	
				5	Nhấn nút Login		Hiển thị thông báo lỗi "Invalid credentials!"

Hình 37: Test case Đăng nhập

■ Thêm địa chỉ

Function code	<i>addUserAddress</i>				
Created By	HoangNhi				

No	Test Requirement ID	Test requirement description	Test Type	Importance	Number of TCs
1	TR_addUserAddress_01	Hệ thống sẽ thông báo "Update Address is successfully!" khi các trường dữ liệu được nhập đầy đủ và đúng.	Positive	High	1
2	TR_addUserAddress_02	Hệ thống sẽ thông báo "Missing input" khi người dùng để trống một hoặc nhiều trường dữ liệu bất kỳ.	Negative	Medium	63

No	Test Requirement ID	Test Objective	Pre-condition	Step	Step action	Test data	Expected result
1	TR_addUserAddress_01	Hệ thống sẽ thông báo "Update Address is successfully!" khi các trường dữ liệu được nhập đầy đủ và đúng.	Đăng nhập thành công vào trang web Ecommerce-DevHouse	1	Đăng nhập thành công trang web Ecommerce-DevHouse		
				2	Chọn Profile		Hiển thị trang thông tin cá nhân của người dùng bao gồm thông tin khi tạo tài khoản, địa chỉ, danh sách yêu thích.
				3	Chọn mục Addresses		Hiển thị trang danh sách địa chỉ người dùng bao gồm thêm, sửa, xóa địa chỉ.
				4	Chọn Add a new address		Hiển thị form điền thông tin địa chỉ.
				5	Nhập đầy đủ và đúng các trường dữ liệu.	Name: Nhi; Phone: 0987654321; City: Hà Nội; District: Ba Đình; Ward: Phúc Xá; AddressDetail: Ngõ 16	
				6	Nhấn Create		Hệ thống hiển thị thông báo "Update Address is
2	TR_addUserAddress_02	Hệ thống sẽ thông báo "Missing input" khi người dùng để trống một hoặc nhiều trường dữ liệu bất kỳ.	Đăng nhập thành công vào trang web Ecommerce-DevHouse	1	Đăng nhập thành công trang web Ecommerce-DevHouse		
				2	Chọn Profile		Hiển thị trang thông tin cá nhân của người dùng bao gồm thông tin khi tạo tài khoản, địa chỉ, danh sách yêu thích.
				3	Chọn mục Addresses		Hiển thị trang danh sách địa chỉ người dùng bao gồm thêm, sửa, xóa địa chỉ.
				4	Chọn Add a new address		Hiển thị form điền thông tin địa chỉ.
				5	Nhập đầy đủ và đúng các trường dữ liệu.	Name: Nhi; Phone: 0987654321; City: Hà Nội; District: Ba Đình; Ward: Phúc Xá; AddressDetail: Ngõ 16	
				6	Xóa một hoặc nhiều trường dữ liệu bất		
				7	Nhấn Create		Hệ thống hiển thị thông báo "Missing input"

Hình 38: Test case Thêm địa chỉ

▪ Cập nhật địa chỉ

Function code	<i>updateUserAddress</i>			
Created By	HoangNhi			

No	Test Requirement ID	Test requirement description	Test Type	Importance	Number of TCs
1	TR_updateUserAddress_01	Hệ thống sẽ thông báo "Update Address is successfully!" khi một trường dữ liệu bất kỳ được thay đổi.	Positive	High	7
2	TR_updateUserAddress_02	Hệ thống sẽ thông báo "Missing input" khi người dùng không chỉnh sửa thông tin.	Negative	Medium	1

No	Test Requirement ID	Test Objective	Pre-condition	Step	Step action	Test data	Expected result
1	TR_updateUserAddress_01	Hệ thống sẽ thông báo "Update Address is successfully!" khi một trường dữ liệu bất kỳ được thay đổi.	Đăng nhập thành công vào trang web Ecommerce-DevHouse	1	Đăng nhập thành công trang web Ecommerce-DevHouse		
				2	Chọn Profile		Hiển thị trang thông tin cá nhân của người dùng bao gồm thông tin khi tạo tài khoản, địa chỉ, danh sách yêu thích.
				3	Chọn mục Addresses		Hiển thị trang danh sách địa chỉ người dùng bao gồm thêm, sửa, xóa địa chỉ.
				4	Chọn Edit địa chỉ đầu tiên		Hiển thị form thông tin địa chỉ có thể chỉnh sửa.
				3	Chỉnh sửa một trường bất kỳ	Name: Văn Đức; Phone: 095656223; City: HCM; District: Bình Thạnh; Ward: Q11; AddressDetail: Cạnh XX; Tick defaultAddress	
				4	Nhấn Update		Hệ thống hiển thị thông báo "Update Address is successfully!"
1	TR_updateUserAddress_02	Hệ thống sẽ thông báo "Missing input" khi người dùng không chỉnh sửa thông tin.	Đăng nhập thành công vào trang web Ecommerce-DevHouse	1	Đăng nhập thành công trang web Ecommerce-DevHouse		
				2	Chọn Profile		Hiển thị trang thông tin cá nhân của người dùng bao gồm thông tin khi tạo tài khoản, địa chỉ, danh sách yêu thích.
				3	Chọn mục Addresses		Hiển thị trang danh sách địa chỉ người dùng bao gồm thêm, sửa, xóa địa chỉ.
				4	Chọn Edit địa chỉ đầu tiên		Hiển thị form thông tin địa chỉ có thể chỉnh sửa.
				3	Không chỉnh sửa thông tin.		
				4	Nhấn Update		Hệ thống hiển thị thông báo "Missing input"

Hình 39: Test case Cập nhật địa chỉ

▪ Tạo sản phẩm

Function code	<i>createProduct</i>			
Created By	Duc-Cute			

Test Requirement ID	Test requirement description	Test Type	Importance	Number of TCs
TR_createProduct_01	Tạo sản phẩm thành công khi người quản trị nhập đầy đủ và đúng các trường dữ liệu.	Positive	High	1
TR_createProduct_02	Hệ thống sẽ thông báo "Missing input" khi người quản trị để trống một hoặc nhiều trường dữ liệu bất kỳ.	Negative	Medium	63

Test Requirement ID	Test Objective	Pre-condition	Step	Step action	Test data	Expected result
TR_createProduct_01	Tạo sản phẩm thành công khi người quản trị nhập đầy đủ và đúng các trường dữ liệu.	Đăng nhập thành công vào trang web Ecommerce-DevHouse	1	Đăng nhập thành công trang web Ecommerce-DevHouse		
			2	Chọn Admin Workspace		Hiển thị trang quản trị hệ thống.
			3	Chọn mục Create Product		Hiển thị trang hiển thị thông tin sản phẩm.
			4	Nhập đầy đủ và đúng các trường dữ liệu	Title: S23Ultra; Price: 23000000; Color: Blue; Category: Smartphone; Brand:Samsung; Description: Good.	
			4	Nhấn Create		Tạo sản phẩm thành công
TR_createProduct_02	Hệ thống sẽ thông báo "Missing input" khi người quản trị để trống một hoặc nhiều trường dữ liệu bất kỳ.	Đăng nhập thành công vào trang web Ecommerce-DevHouse	1	Đăng nhập thành công trang web Ecommerce-DevHouse		
			2	Chọn Admin Workspace		Hiển thị trang quản trị hệ thống.
			3	Chọn mục Create Product		Hiển thị trang hiển thị thông tin sản phẩm.
			4	Nhập đầy đủ và đúng các trường dữ liệu	Title: S23Ultra; Price: 23000000; Color: Blue; Category: Smartphone; Brand:Samsung; Description: Good.	
			5	Xóa dữ liệu của một hoặc nhiều trường bất kỳ.		
			6	Nhấn Create		Hiển thị thông báo "Missing input"

Hình 40: Test case Tạo sản phẩm

- Cập nhật sản phẩm

Function code	<i>updateProduct</i>			
Created By	Duc-Cute			
Test Requirement ID	Test requirement description	Test Type	Importance	Number of TCs
TR_updateProduct_01	Hệ thống sẽ thông báo "Update product success" khi người quản trị chỉnh sửa 1 trường dữ liệu bất kỳ.	Positive	High	6
TR_updateProduct_02	Hệ thống sẽ thông báo "Missing input" khi người quản trị không chỉnh sửa thông tin.	Negative	Medium	1

Test Requirement ID	Test Objective	Pre-condition	Step	Step action	Test data	Expected result
TR_updateProduct_01	Hệ thống sẽ thông báo "Update product success" khi người quản trị chỉnh sửa 1 trường dữ liệu bất kỳ.	Đăng nhập thành công vào trang web Ecommerce-DevHouse	1	Đăng nhập thành công trang web Ecommerce-DevHouse		
			2	Chọn Admin Workspace		Hiển thị trang quản trị hệ thống.
			3	Chọn mục Manage Product		Hiển thị trang danh sách sản phẩm
			4	Chọn biểu tượng chỉnh sửa ở mục Action		Hiển thị thông tin sản phẩm
			4	Chỉnh sửa 1 trường dữ liệu bất kỳ	Title: S22Ultra; Price: 22000000; Color: Black; Category: Smartphone; Brand:Samsung; Description: Good.	
			4	Nhấn Update		Hiển thị thông báo "Update product success"
TR_updateProduct_02	Hệ thống sẽ thông báo "Missing input" khi người quản trị không chỉnh sửa thông tin.	Đăng nhập thành công vào trang web Ecommerce-DevHouse	1	Đăng nhập thành công trang web Ecommerce-DevHouse		
			2	Chọn Admin Workspace		Hiển thị trang quản trị hệ thống.
			3	Chọn mục Manage Product		Hiển thị trang danh sách sản phẩm
			4	Chọn biểu tượng chỉnh sửa ở mục Action		Hiển thị thông tin sản phẩm
			4	Không chỉnh sửa thông tin		
			4	Nhấn Update		Hiển thị thông báo "Missing input"

Hình 41: Test case Cập nhật sản phẩm

- Đánh giá

Function code	<i>ratings</i>			
Created By	<i>Duc-Cute</i>			
Test Requirement ID	Test requirement description	Test Type	Importance	Number of TCs
TR_ratings_01	Hệ thống cập nhật đánh giá thành công khi người dùng nhập đầy đủ các trường dữ liệu.	Positive	High	1
TR_ratings_02	Hệ thống sẽ thông báo "Missing input" khi người dùng bỏ trống 1 hoặc nhiều trường bất kỳ.	Negative	Medium	3

Test Requirement ID	Test Objective	Pre-condition	Step	Step action	Test data	Expected result
TR_ratings_01	Hệ thống cập nhật đánh giá thành công khi người dùng nhập đầy đủ các trường dữ liệu.	Đăng nhập thành công vào trang web Ecommerce-DevHouse	1	Đăng nhập thành công trang web Ecommerce-DevHouse		
			2	Chọn một sản phẩm bất kỳ trên trang web		Hiển thị trang thông tin sản phẩm bao gồm các đánh giá sản phẩm
			3	Chọn mục Vote Now		Hiển thị form điền thông tin đánh giá
			4	Nhập đầy đủ và đúng các trường dữ liệu	Cmt: Good; Star: 5	
			4	Nhấn Submit		Đánh giá thành công
TR_ratings_02	Hệ thống sẽ thông báo "Missing input" khi người dùng bỏ trống 1 hoặc nhiều trường bất kỳ.	Đăng nhập thành công vào trang web Ecommerce-DevHouse	1	Đăng nhập thành công trang web Ecommerce-DevHouse		
			2	Chọn một sản phẩm bất kỳ trên trang web		Hiển thị trang thông tin sản phẩm bao gồm các đánh giá sản phẩm
			3	Chọn mục Vote Now		Hiển thị form điền thông tin đánh giá
			4	Nhập đầy đủ và đúng các trường dữ liệu	Cmt: Good; Star: 5	
			4	Xóa một hoặc cả hai trường dữ liệu		
			4	Nhấn Submit		Hiển thông báo "Missing input"

Hình 42: Test case Đánh giá

3. Kiểm thử hệ thống (System testing)

3.1 Kiểm thử hệ thống là gì?

- **Kiểm thử hệ thống (System Testing)** là quá trình kiểm tra toàn bộ hệ thống hoạt động như thế nào trong môi trường thực tế, bao gồm cả các phần mềm, phần cứng, dữ liệu, và các yếu tố khác. Mục tiêu của kiểm thử hệ thống là đảm bảo rằng hệ thống hoạt động đúng như mong đợi và đáp ứng được các yêu cầu và tiêu chí chất lượng đã được đề ra.
- Trong quá trình này, các bài kiểm tra thường tập trung vào các tính năng tổng hợp của hệ thống và cách chúng tương tác với nhau. Kiểm thử hệ thống có thể bao gồm kiểm tra tính đúng đắn, hiệu suất, bảo mật, độ tin cậy và khả năng khôi phục của hệ thống trong các điều kiện khác nhau.
- Các phương pháp kiểm thử hệ thống có thể bao gồm kiểm thử đen (black-box testing), trong đó kiểm thử được thực hiện mà không cần biết về cấu trúc nội bộ của hệ thống, hoặc kiểm thử trắng (white-box testing), trong đó kiểm thử được thực hiện dựa trên kiến thức về cấu trúc nội bộ của hệ thống.

3.2 Phương pháp, kỹ thuật của kiểm thử hệ thống

• Phương pháp

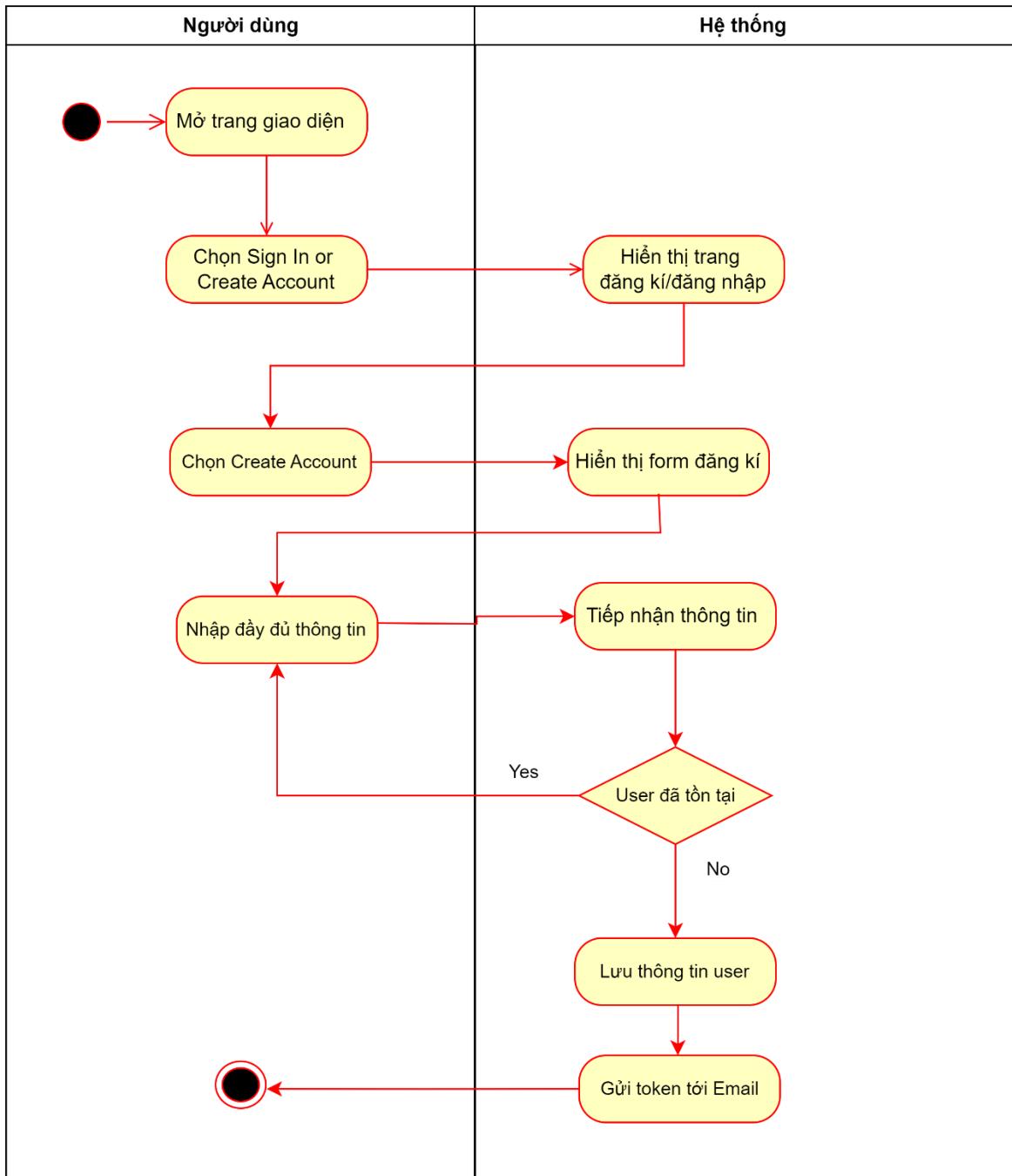
- **Kiểm thử hồi quy (Regression Testing)**: Kiểm tra lại các tính năng hoặc chức năng đã được kiểm thử trước đó sau khi có sự thay đổi trong hệ thống.
- **Kiểm thử hợp nhất (Integration Testing)**: Kiểm tra sự tương tác giữa các thành phần của hệ thống sau khi chúng đã được tích hợp lại với nhau.
- **Kiểm thử hạng nặng (Smoke Testing)**: Kiểm tra các tính năng cơ bản của hệ thống để đảm bảo rằng nó có thể hoạt động một cách cơ bản trước khi tiến hành kiểm thử chi tiết.
- **Kiểm thử hiệu suất (Performance Testing)**: Đánh giá hiệu suất của hệ thống dưới tải công việc cụ thể và điều kiện môi trường nhất định.
- **Kiểm thử bảo mật (Security Testing)**: Đánh giá cơ sở hạ tầng và ứng dụng của hệ thống để phát hiện các lỗ hổng bảo mật có thể bị khai thác.
- **Kiểm thử cấu hình (Configuration Testing)**: Đảm bảo rằng hệ thống hoạt động đúng cách dưới nhiều cấu hình khác nhau (hệ điều hành, phiên bản trình duyệt, v.v.).
- **Kiểm thử khôi phục (Recovery Testing)**: Kiểm tra khả năng của hệ thống để phục hồi sau khi xảy ra lỗi hoặc sự cố.

• Kỹ thuật

- **Kiểm thử đen (Black Box Testing)**: Kiểm thử dựa trên phân tích yêu cầu và không yêu cầu kiến thức về cấu trúc bên trong của hệ thống.
- **Kiểm thử trắng (White Box Testing)**: Kiểm thử dựa trên kiến thức về cấu trúc nội bộ của hệ thống, bao gồm mã nguồn và cấu trúc dữ liệu.
- **Kiểm thử hợp đồng (Contract Testing)**: Kiểm thử các giao diện hoặc giao thức giữa các thành phần để đảm bảo tính nhất quán và tuân thủ.
- **Kiểm thử tương tác (Exploratory Testing)**: Kiểm thử không có kế hoạch cụ thể, mà là việc thử nghiệm dựa trên sự tìm hiểu và khám phá của người kiểm thử về hệ thống.
- **Kiểm thử tự động (Automated Testing)**: Sử dụng các công cụ và kịch bản để thực hiện các bài kiểm tra một cách tự động.
- **Kiểm thử hướng đối tượng (Object-Oriented Testing)**: Kiểm thử tập trung vào các lớp và đối tượng của hệ thống để đảm bảo tính nhất quán và hiệu quả của mã nguồn.

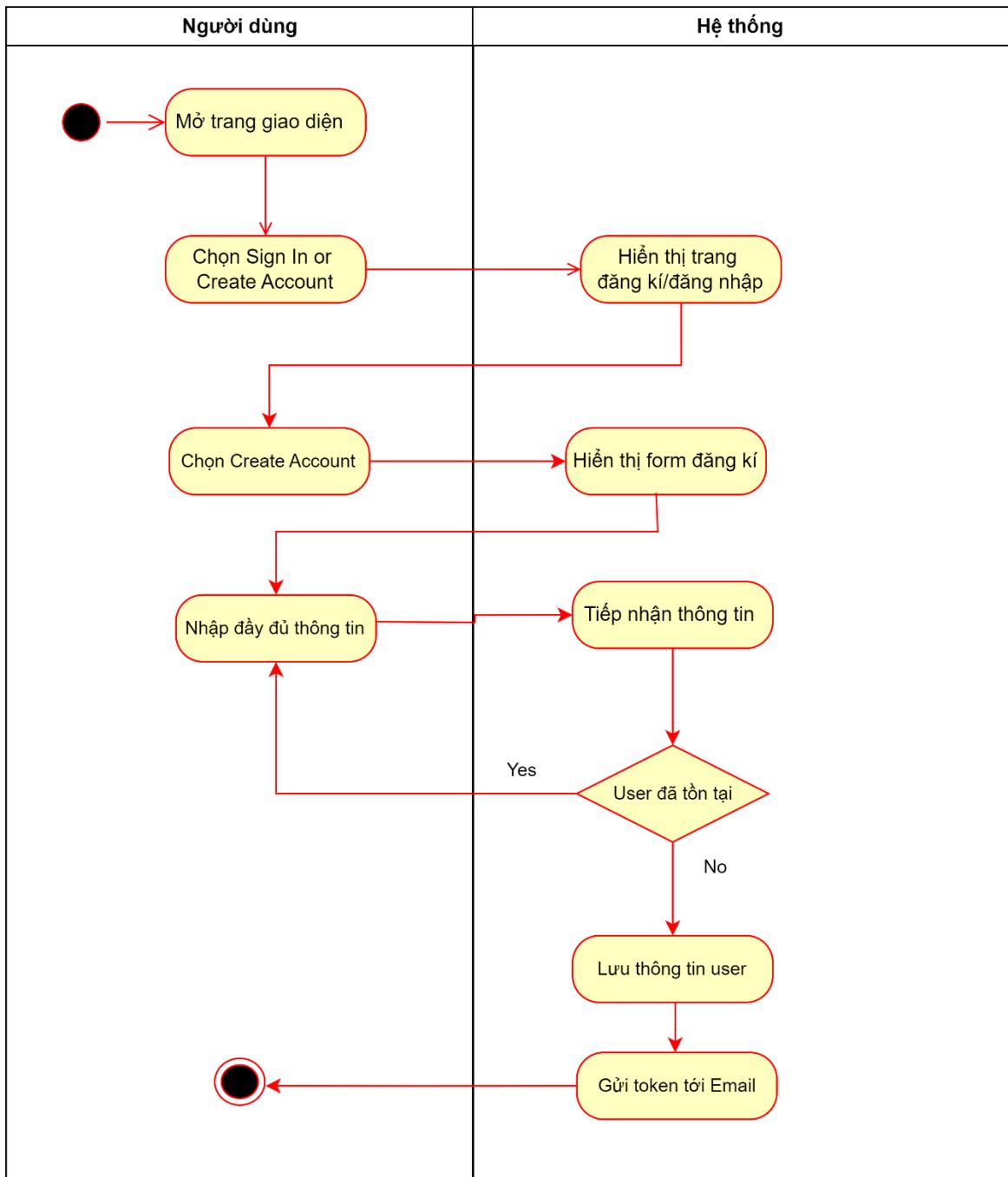
3.3 Biểu đồ hoạt động (Activity Diagrams)

- Biểu đồ hoạt động người dùng Đăng ký



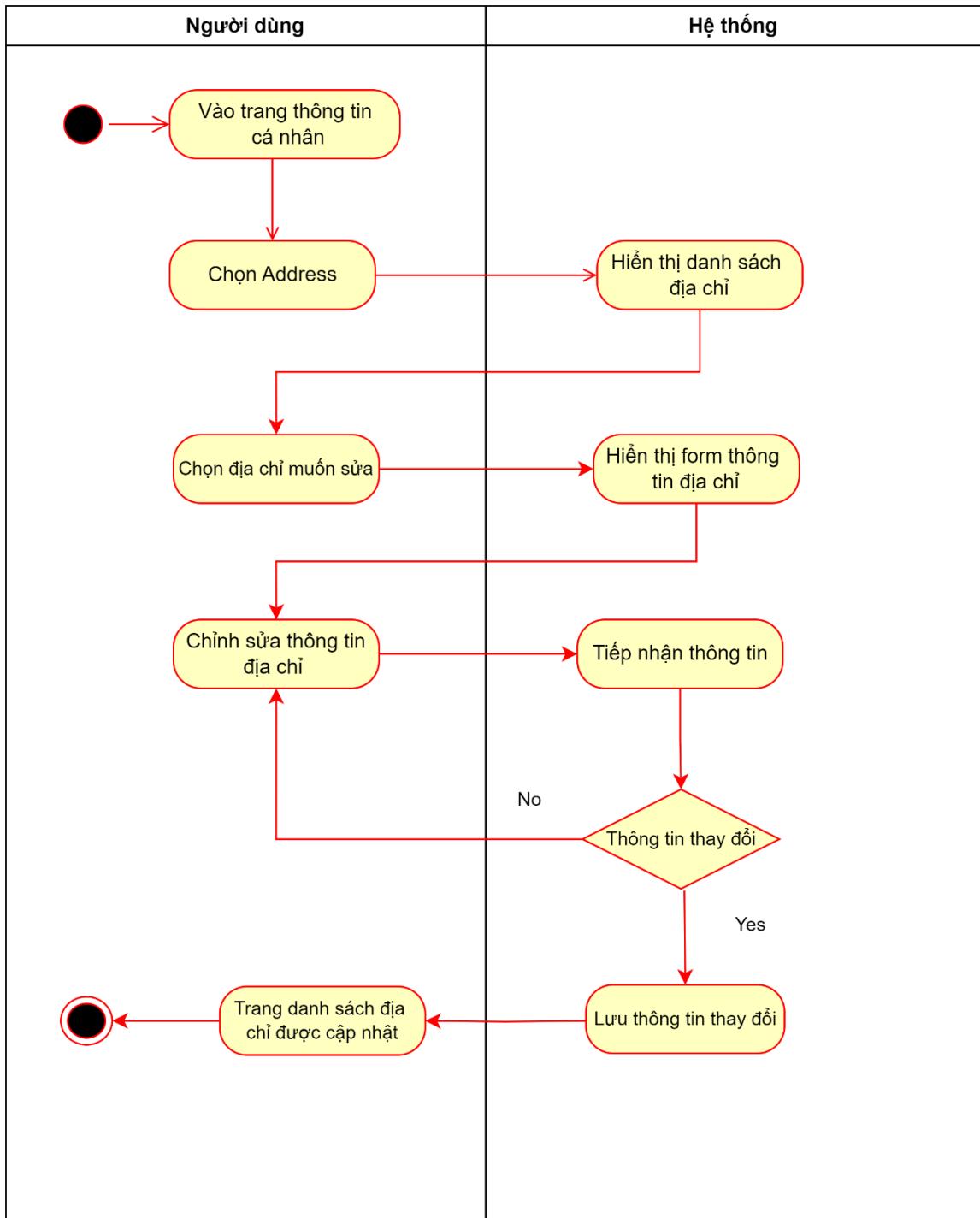
Hình 43: Biểu đồ hoạt động người dùng Đăng kí

▪ Biểu đồ hoạt động người dùng Đăng nhập



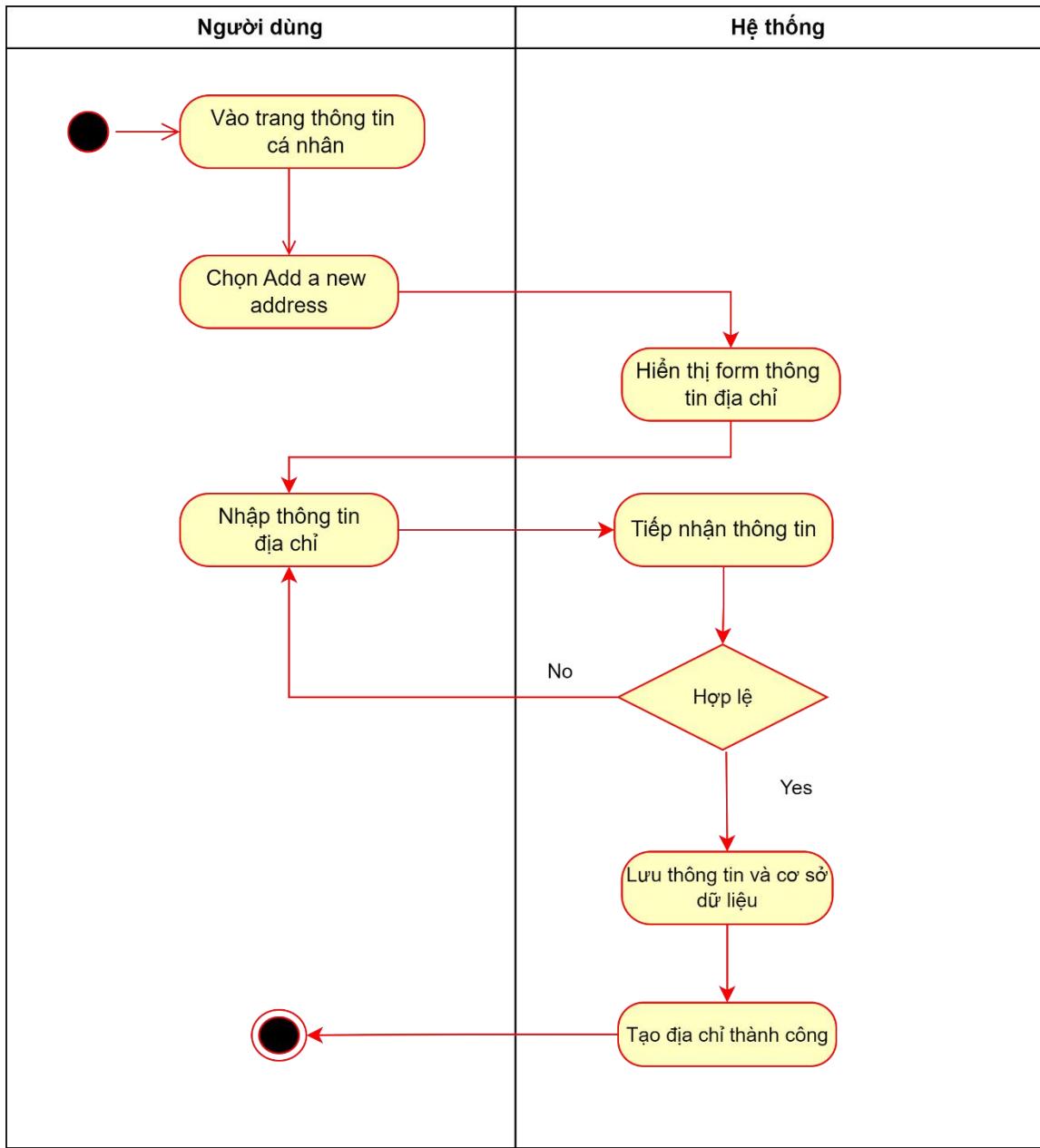
Hình 44: Biểu đồ hoạt động người dùng Đăng nhập

▪ Biểu đồ hoạt động cập nhật địa chỉ



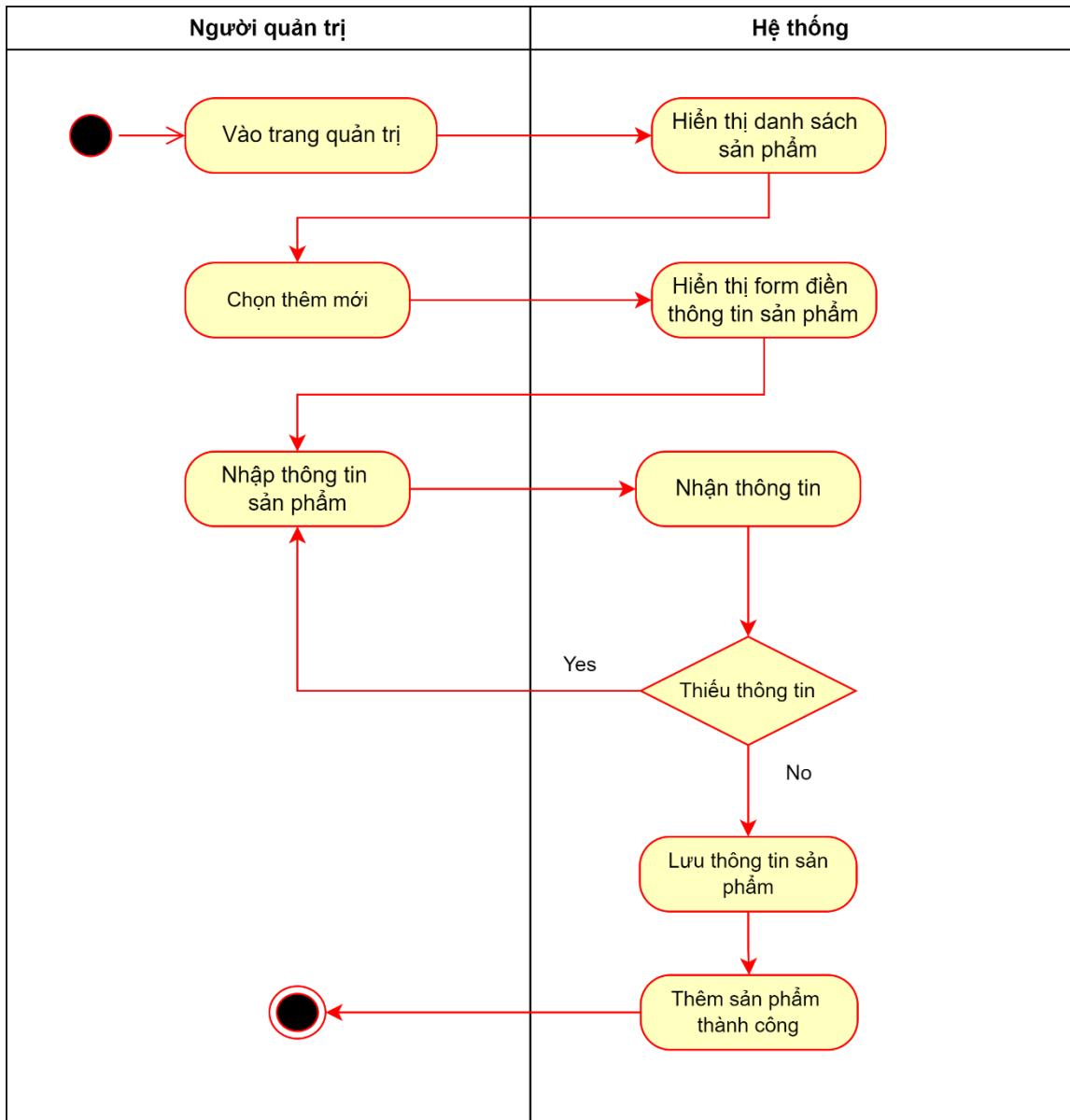
Hình 45: Biểu đồ hoạt động cập nhật địa chỉ

▪ Biểu đồ hoạt động Thêm địa chỉ



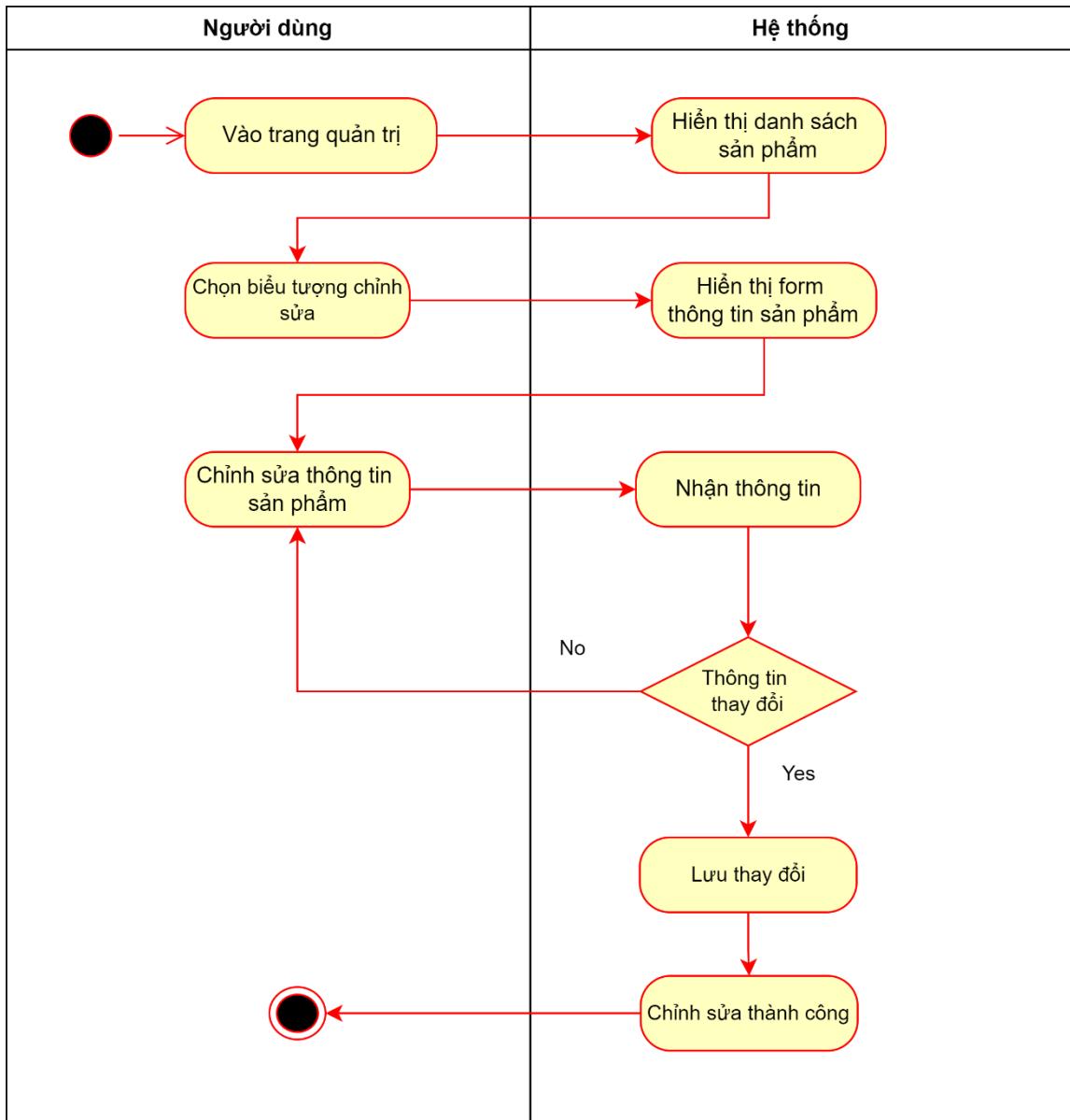
Hình 46: Biểu đồ hoạt động Thêm địa chỉ

▪ Biểu đồ hoạt động người quản trị tạo sản phẩm



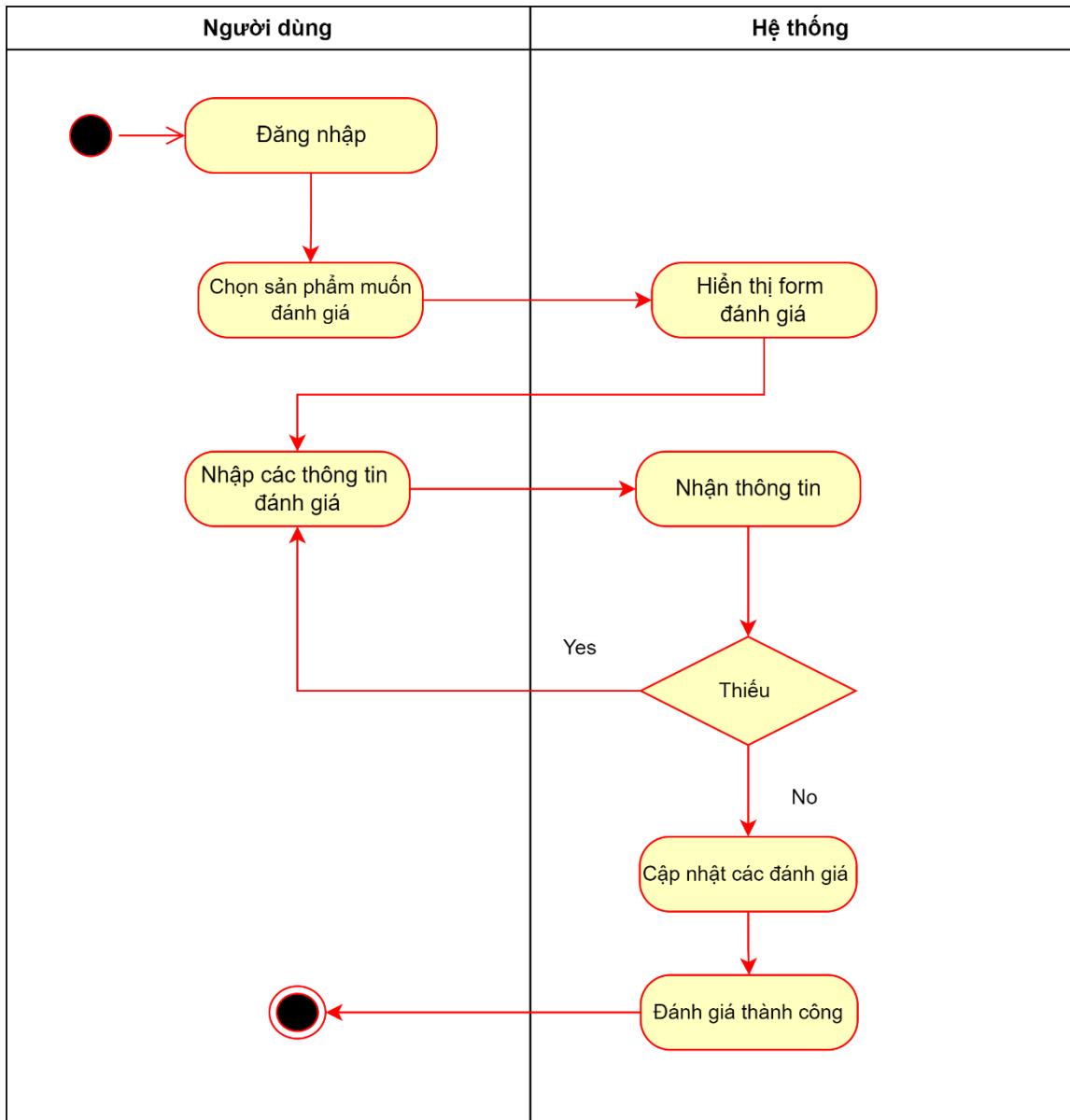
Hình 47: Biểu đồ hoạt động người quản trị tạo sản phẩm

▪ Biểu đồ hoạt động người quản trị chỉnh sửa sản phẩm



Hình 48: Biểu đồ hoạt động người quản trị chỉnh sửa sản phẩm

▪ Biểu đồ hoạt động người dùng đánh giá sản phẩm



Hình 49: Biểu đồ hoạt động người dùng đánh giá sản phẩm

3.4 System Test Case

System testcase list

TEST CASE LIST

Project Name	<i>SystemTestEcommerce</i>
Project Code	<i>Ecommerce-Server</i>
Test Environment Setup Description	

No	Function Name	Sheet Name	Description	Pre-Condition
1	CreateProduct	ST_CreateProduct		
2	Ratings	ST_Ratings		

Hình 50: System testcase list

Tạo sản phẩm

Module Code	<i>ST_CreateProduct</i>							
ID	Test Case Description	Pre-condition	Test Steps	Expected Output	Post-condition	Result	Test date	Note
CreateProduct_01	User với vai trò Admin truy cập phần Admin Workspace; Kiểm tra xem user có Role Admin có thể xem mục quản lý Product không	1. Đăng nhập vào hệ thống với vai trò Admin	1. Chọn Mục "Profile". 2. Chọn Mục "Admin Workspace" 3. Click Mục "Manage Product"	1. Hiển thị danh sách sản phẩm.				
CreateProduct_02	User không phải vai trò Admin truy cập phần Admin Workspace; Kiểm tra xem User không phải Role Admin có thể xem mục quản lý Product không	1. Đăng nhập vào hệ thống với vai trò không phải Admin	1. Chọn Mục "Profile". 2. Chọn Mục "Admin Workspace" 3. Click Mục "Manage Product"	1. Không hiển thị danh sách sản phẩm.				

Hình 51: System Testcase Tạo sản phẩm

Ratings

Module Code	<i>Ratings</i>							
ID	Test Case Description	Pre-condition	Test Steps	Expected Output	Post-condition	Result	Test date	Note
Ratings_01	User đã đăng nhập truy cập sản phẩm bất kỳ, chọn mục Vote Now; Kiểm tra xem hệ thống có hiển thị form đánh giá không	1. Đăng nhập vào hệ thống với vai trò Admin	1. Chọn một sản phẩm bất kỳ. 2. Chọn Mục "Vote Now"	1. Hiển thị form đánh giá.				
Ratings_02	User chưa đăng nhập tài khoản truy cập sản phẩm bất kỳ, chọn mục Vote Now; Kiểm tra xem hệ thống có hiển thị form đánh giá không	1. Truy cập vào trang web Ecommerce	1. Chọn một sản phẩm bất kỳ. 2. Chọn Mục "Vote Now"	1. Hiển thị thông báo Hủy đánh giá hoặc Login để được đánh giá.				

Hình 52: System Testcase Đánh giá

CHƯƠNG 3: THỰC THI TEST VÀ BÁO CÁO KẾT QUẢ TEST

1. Unit Test

1.1 Viết script thực hiện Unit Test

```
import mongoose from "mongoose";
import supertest from "supertest";
import createServer from "../utils/server";
const User = require("../models/user");

import * as UserController from "../controller/user";
// import * as SessionService from "../service/session.service";
// import { createUserSessionHandler } from "../controller/session.controller";

const app = createServer();

const userId = new mongoose.Types.ObjectId().toString();

const userPayload = {
};

const userInput = {

};

const userInputAddress = {

    // Thêm các thông tin khác của người dùng nếu cần
    address
};

const sessionPayload = {

};

escribe.only("user", () => {
    // user registration
    describe("user registration", () => {

        describe.only("given the email is empty", () => {
            });

        describe.only("given the email has existed", () => {
```

```
});

    describe("given the email is valid", () => {
});

    });

describe("user login", () => {
    describe.only("given the email or password is empty", () => {
});

    describe.only("given the email or password is incorrect", () => {
});

    describe("given the email or password true", () => {
});

    });

describe("add new address user", () => {
});

describe("updateUserAddress function", () => {

});

});
```

1.2 Báo cáo kết quả

User.test.js bao gồm Unit Test của các function: register, login, addUserAdress, updateUserAddress. Tổng có 10/10 trường hợp đã Passed.

```
user.test.js 1, M producttest.js M
E-Commerce-server > _tests_ > producttest.js > describe("Authentication") callback > beforeAll() callback > loginResponse > email
1 import mongoose from "mongoose";
2 import supertest from "supertest";
3 import createServer from "../utils/server";
4 const Product = require("../models/product");
5 const User = require("../models/user");
6
7 const app = createServer();
8
9 const userId = new mongoose.Types.ObjectId().toString();
10 const userInput = {
11   email: "test@example.com",
12   password: "password",
13   firstname: "John",
14   lastname: "Doe",
15   mobile: "123456729",
16   role: "admin",
17 };
18
19 describe("Authentication", () => {
20   it("should throw an error if required inputs are missing", () => {
21     const response = supertest(app)
22       .post("/api/auth/register")
23       .send({});
24     expect(response.status).toBe(400);
25     expect(response.body.message).toEqual("Email and Password are required");
26   });
27   it("should create a new product successfully", () => {
28     const response = supertest(app)
29       .post("/api/products")
30       .send({
31         name: "Laptop Dell XPS 15",
32         description: "A high-end laptop with Intel i7 processor and 16GB RAM.",
33         price: 1500,
34         stock: 10,
35         category: "Electronics",
36       });
37     expect(response.status).toBe(201);
38     expect(response.body.message).toEqual("Product created successfully");
39   });
40   it("should throw an error if no update data provided", () => {
41     const response = supertest(app)
42       .patch("/api/products/64d4f12e0000000000000001")
43       .send({});
44     expect(response.status).toBe(400);
45     expect(response.body.message).toEqual("Update data is required");
46   });
47   it("should update the product successfully", () => {
48     const response = supertest(app)
49       .patch("/api/products/64d4f12e0000000000000001")
50       .send({
51         name: "Laptop Dell XPS 15 (Updated)",
52         description: "A high-end laptop with Intel i7 processor and 16GB RAM (Updated).",
53         price: 1500,
54         stock: 10,
55         category: "Electronics",
56       });
57     expect(response.status).toBe(200);
58     expect(response.body.message).toEqual("Product updated successfully");
59   });
60   it("should throw an error if required inputs are missing", () => {
61     const response = supertest(app)
62       .patch("/api/products/64d4f12e0000000000000001")
63       .send({});
64     expect(response.status).toBe(400);
65     expect(response.body.message).toEqual("Email and Password are required");
66   });
67   it("should rate the product successfully", () => {
68     const response = supertest(app)
69       .patch("/api/products/64d4f12e0000000000000001")
70       .send({
71         rating: 4.5,
72       });
73     expect(response.status).toBe(200);
74     expect(response.body.message).toEqual("Rating updated successfully");
75   });
76   it("should throw an error if required inputs are missing", () => {
77     const response = supertest(app)
78       .patch("/api/products/64d4f12e0000000000000001")
79       .send({});
80     expect(response.status).toBe(400);
81     expect(response.body.message).toEqual("Email and Password are required");
82   });
83 });

Test Suites: 1 passed, 1 total
Tests:       6 passed, 6 total
Snapshots:  0 total
Time:        5.711 s
Ran all test suites matching /d:\\Admin\\Workspace\\main\\testing\\E-Commerce-server\\_tests_\\product\\.test\\.js/i with tests matching "Authentication".
Force exiting Jest: Have you considered using '--detectOpenHandles' to detect a sync operations that kept running after all tests finished?
```

PROBLEMS DEBUG CONSOLE TEST RESULTS COMMENTS OUTPUT TERMINAL

Test run at 4/30/2024, 1:35:13 AM
should throw an error if required inputs are ...
should create a new product successfully
should throw an error if no update data pro...
should update the product successfully
should throw an error if required inputs are ...
should rate the product successfully
Test run at 4/30/2024, 1:34:44 AM
should throw an error if required inputs are ...
Go Live tabnine starter Prettier

2. Integration Test

2.1 Viết script thực hiện Integration Test

```
import mongoose from "mongoose";
import supertest from "supertest";
import createServer from "../utils/server";
const Product = require("../models/product");
const User = require("../models/user");

const app = createServer();

const userId = new mongoose.Types.ObjectId().toString();
const userInput = {
  email: "test@example.com",
  password: "password",
  firstname: "John",
  lastname: "Doe",
  mobile: "123456729",
  role: "admin",
};
```

```
describe("Authentication", () => {
  let accessToken;
  let userId;

  beforeAll(async () => {
  });

  afterAll(async () => {
  });

  describe("createProduct function", () => {

  });
  describe("updateProduct function", () => {
    let productId;

    beforeAll(async () => {
    });

    afterAll(async () => {
    });

    it("should throw an error if no update data provided", async () => {
    });

    it("should update the product successfully", async () => {
    });
  });
  describe("ratings function", () => {
    let productId;

    beforeAll(async () => {
    });

    afterAll(async () => {
    });

    it("should throw an error if required inputs are missing", async () => {
    });

    it("should rate the product successfully", async () => {
```

});

2.2 Báo cáo kết quả

product.test.js bao gồm Integration Test của các function: createProduct, addProduct, Ratings. Tổng có 6/6 trường hợp đã Passed.

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure with files like user.test.js, product.test.js, config, controller, models, and various .js files for blog, brand, coupon, insertData, order, product, productCategory, and user.
- Code Editor:** The active file is product.test.js, which contains Jest test code for a MongoDB-based application. It imports mongoose, supertest, and createServer, and defines a user object with various properties and roles.
- Terminal:** Shows the output of the test run:

```
Test Suites: 1 passed, 1 total
Tests:       6 passed, 6 total
Snapshots:  0 total
Time:        5.711 s
Ran all test suites matching /d:\\Admin\\\\Workspace\\main\\testing\\E-Commerce-server\\_tests_\\product\\.test\\.js/i with tests matching "Authentication".
Force exiting Jest: Have you considered using `--detectOpenHandles` to detect a sync operations that kept running after all tests finished?
```
- Status Bar:** Shows the current file is product.test.js, line 30, column 20, and the file is saved (S). It also shows the date and time as 4/30/2024, 1:35:13 AM.

CHƯƠNG 4: AUTOMATION TEST

1. Công cụ sử dụng

Jest: Là framework thử nghiệm JavaScript sử dụng để viết và thực thi các test cases.

Visual Studio Code (VSCode): Là môi trường phát triển tích hợp (IDE) sử dụng để viết mã và chạy các test cases.

JavaScript: Ngôn ngữ lập trình chính bạn đang viết các test cases và mã nguồn cần được kiểm tra.

npm: Các công cụ quản lý gói phần mềm JavaScript/Node.js, cung cấp các gói như Jest và các thư viện hỗ trợ khác cần thiết cho việc viết và thực thi các test cases.

Supertest: kiem thử tự động API.

2. Kết quả đạt được

Sau khi nghiên cứu và hoàn thiện bài tập lớn này, chúng tôi nhận thấy quá trình kiểm thử phần mềm là vô cùng cần thiết. Việc tìm hiểu về Unit Test, Integration Test, System Test đã giúp nhóm nắm được một lượng lớn kiến thức liên quan tới kiểm thử phần mềm, đặc biệt là thực hành test hệ thống web bằng môi trường Jest trên VScode, sử dụng ngôn ngữ chính là JavaScript.

Ưu điểm: Kiểm thử đơn vị và kiểm thử tích hợp là các phương thức test khá phổ biến nên việc triển khai không quá phức tạp, việc viết và chạy các testcase đơn giản và tiện lợi. Việc Jest được tích hợp sẵn với các công cụ khác giúp việc viết các testcase trở nên hiệu quả hơn, tốc độ chạy nhanh, hỗ trợ async/await giúp việc kiểm tra các hàm và API trở nên dễ dàng hơn. Chúng tôi cũng đã lên kế hoạch phân công và hoàn thành các đầu việc tương đối chính chu để có thể xây dựng đề tài một cách tốt nhất.

Nhược điểm: Chúng tôi gặp khó khăn trong việc cấu hình Jest vì nó có cấu hình tương đối phức tạp và chỉ hỗ trợ ngôn ngữ JavaScript.

Hướng phát triển:

Nhóm mong muốn có thể thực hiện kiểm thử toàn bộ hệ thống website thay vì lựa chọn test một vài hàm như kế hoạch đề ra. System Testing sẽ được triển khai và nâng cao về kiểm thử hiệu suất.

KẾT LUẬN

Bài tập lớn kiểm thử phần mềm đã là một phần quan trọng trong quá trình phát triển sản phẩm. Qua việc áp dụng các phương pháp và công cụ kiểm thử phù hợp, chúng tôi đã có thể đạt được những kết quả quan trọng và đáng giá. Việc thực hiện các bài test unit, integration và system test đã giúp chúng tôi đảm bảo tính đúng đắn và hiệu suất của sản phẩm. Đồng thời, việc kiểm thử đã giúp chúng tôi phát hiện và sửa chữa các lỗi và vấn đề một cách kịp thời, giảm thiểu rủi ro trong quá trình triển khai và sử dụng sản phẩm.

Qua việc kiểm tra từ góc nhìn người dùng, chúng tôi đã đảm bảo rằng sản phẩm đáp ứng được các yêu cầu và mong đợi của người dùng cuối, từ giao diện đến tính năng. Điều này đã giúp nâng cao trải nghiệm người dùng và tạo ra một sản phẩm mà người dùng có thể tin cậy và sử dụng một cách dễ dàng.

Cuối cùng, bài tập lớn kiểm thử phần mềm đã chứng minh vai trò quan trọng của quy trình kiểm thử trong việc đảm bảo chất lượng và hiệu suất của sản phẩm phần mềm. Các kết quả và kinh nghiệm thu được từ dự án này sẽ là nền tảng để phát triển và cải thiện sản phẩm trong tương lai.

TÀI LIỆU THAM KHẢO

1. 04 Điều cần chú ý cho người mới làm Automation Test - TopDev.vn - <https://topdev.vn/blog/04-dieu-can-chu-y-cho-nguo-moi-lam-automation-test/>
2. System Testing là gì? – TopDev.vn - <https://topdev.vn/blog/system-testing-la-gi/>
3. Jest ·  Delightful JavaScript Testing - JestJS.io - <https://jestjs.io/>
4. Online API Testing Tool - Test API, Website, DB, and more - ReqBin.com - <https://reqbin.com/>
5. Integration Testing là gì? - TechAcademy.edu.vn - <https://techacademy.edu.vn/integration-testing-la-gi/>