



TRƯỜNG ĐẠI HỌC SÀI GÒN

Khoa Công nghệ thông tin

FINAL REPORT

COURSE: SOFTWARE TESTING

PROJECT : APPLYING TESTING TO A CAKE SELLING WEBSITE

Instructor: Do Nhu Tai

Member name:

- | | |
|------------------|------------------------|
| + Huynh Anh Tuan | Student ID: 3121411220 |
| + Phung Tan Nhut | Student ID: 3121411159 |

School year: 2024 - 2025

INDEX

CHAPTER 1: OVERVIEW OF BAKERY WEBSITE TESTING	7
1. Introduction	7
2. Business context	8
3. Business process	8
3.1. Import process row	8
3.2. Purchasing process	9
4. Problem solving subsystems	11
4.1. Import process	11
4.2. Purchasing process	12
5. Conceptual Model	13
6. User Stories	14
7. Business requirements	15
7.1. Functional requirements	15
7.2. Non-functional requirements	15
8. System participants	17
CHAPTER 2: SOFTWARE REQUIREMENTS SPECIFICATION.....	18
1. Deployment diagram	18
1.1. Import goods	18
1.2. Purchase	18
2. Overview diagram of system functions	19
3. General database diagram	20
4. Import goods	21
4.1. Focused UseCase	21
4.2. Activity Diagram	24
4.3. Sequence Diagram	25
4.4. Class Diagram	26
4.5. Interface illustration	26
5. Purchase	29
5.1. Focused UseCase	29
5.2. Activity Diagram	33
5.3. Sequence Diagram	34

5.4. Class Diagram	35
5.5. Illustrative interface	36
CHAPTER 3: TEST PLAN.....	39
1. Introduction	39
1.1. Purpose	39
1.2. Background	39
1.3. Scope	39
1.4. Project identification	39
1.5 References	40
2. Features to test.....	41
2.1. Function (Customer)	41
2.1.1. Add products to cart	41
2.1.2. Remove products from cart	41
2.1.3. Update products in cart	41
2.1.4. Shopping cart payment	41
2.2. Function (Manager)	42
2.2.1. Add products to inventory	42
2.2.2. Remove product from inventory	42
2.2.3. Update products in stock	42
2.3. Availability	42
2.4. Design limitations	42
2.5. Interface	43
2.5.1. User interface	43
3. Features that do not need testing	43
3.1. Function (Customer)	43
3.2. Function (Manager)	44
4. Testing strategy	47
4.1. Types of testing	47
4.2. Tools	47
5. Resources	47
6. Project milestones	48
7. Output products	48

7.1 Testing model	48
7.2 Test log	48
CHAPTER 4: TESTING METHODS.....	49
1. Unit Testing	49
1.1. TestPlan	50
1.2. Import process	53
1.3. Purchasing process	56
2. Integration Testing	58
2.1. Test Design	58
2.2. White box testing	62
2.2.1. Import process	64
2.2.2. Purchasing process	75
2.3. Black box testing	85
2.3.1. Import process	87
2.3.2. Purchasing process	94
3. System Testing	98
3.1. TestCase List	100
3.2. Import process	102
3.3. Purchasing process	105
3.4 . Non-functional	107
4. Acceptance Testing	108
CHAPTER 5: TEST RESULTS.....	109
1. Test Report	109
2. Bug Report	110
3. DefectList	111
4. Auto Test	112
CHAPTER 6. CONCLUSION.....	117

LIST OF CATEGORIES, IMAGES

Figure 1. Import process diagram	11
Figure 2. Purchasing process diagram	13
Figure 3. Conceptual model	13
Figure 4. System participants	17
Figure 5. Import Deployment Diagram	18
Figure 6. Purchasing Deployment Diagram	18
Figure 7. General UseCase	19
Figure 8. Database Diagram	20
Figure 9. Activity diagram of the Import process	24
Figure 10. Sequence diagram of the Import process	25
Figure 11. Class diagram of the Import process	26
Figure 12. Product deletion interface Import goods	26
Figure 13. Add product interface Import goods	27
Figure 14. Product update interface Import goods	28
Figure 15. Activity diagram of Purchasing process	33
Figure 16. Purchase Process Sequence Diagram	35
Figure 17. Purchase Process Class Diagram	35
Figure 18. Default interface for adding a product to cart when purchasing	36
Figure 19. Interface for adding products with optional quantity when purchasing	37
Figure 20. Interface for viewing and updating shopping cart	37
Figure 21. Interface to delete products from cart	38
Figure 22. Shopping cart payment interface	39
Figure 23. TestPlan Test Plan Table	50
Figure 24. Unit Test Importing goods	53
Figure 25. Purchase Unit Test	56
Figure 26. Test Design Import	59
Figure 27. Test Design Purchase	60
Figure 28. White Box CheckList	62
Figure 29. Control flow diagram Add product Import goods	64
Figure 30. Product Update Control Flow Diagram	69
Figure 31. Control flow diagram Delete product Import goods	73
Figure 32. Control flow diagram Add product Shopping cart	75
Figure 33. Control flow diagram of Update Shopping Cart	78
Figure 34. Control flow diagram Delete product Shopping cart	80
Figure 35. Control flow diagram of Checkout Shopping Cart	82
Figure 36. Black Box CheckList	85
Figure 37. Decision table Add product	87
Figure 38. Product Update Decision Table	90
Figure 39. Product Delete Decision Table	93
Figure 40. Decision table Add product to Cart	94
Figure 41. Decision table Update quantity in Shopping Cart	95
Figure 42. Decision table to Remove product from shopping cart	96

Figure 43. Cart Checkout Decision Table	97
Figure 44. List of test cases	100
Figure 45. Module and test results for the import process	102
Figure 46. Data Test for the import process	103
Figure 47. TestCase of import process	104
Figure 48. Module and test results for the purchasing process	105
Figure 49. Test Data for the purchasing process	105
Figure 50. Purchase Process TestCae	106
Figure 51. Test results for non-functionality	107
Figure 52. Test Data for Non-Functional	107
Figure 53. Non-functional TestCase	107
Figure 54. Q&A	108
Figure 55. TestReport of TestCases (UnitTest	109
Figure 56. TestReport of TestCases (white box and black box)	110
Figure 57. DefectList	111

CHAPTER 1: OVERVIEW OF BAKERY WEBSITE TESTING

1. Introduction

Software testing is an important step in the process of developing a Bakery Website, ensuring that the system operates correctly, stably, and provides the best experience for users. The Bakery Website is designed with many functions such as product management, import, and online shopping, meeting the operational needs of administrators and customers.

Website Bakery software testing focuses on verifying all system features, detecting errors, ensuring compatibility with different environments, and fully meeting business requirements.

Objective of testing

- **Ensure accuracy:** Test features like adding products, updating carts, and checkout to ensure they work as expected.
- **Error Detection and Correction:** Identify potential problems in the system before going live.
- **Enhance user experience:** Ensure the interface is easy to use and functions work smoothly across multiple devices.
- **Performance Evaluation:** Tests the system's ability to handle concurrency and responsiveness under high load conditions.

The Importance of Testing Bakery Website

- Testing helps ensure that the Bakery Website operates smoothly and without errors during use.
- Ensure product information, orders, and customer data are always stored accurately.
- Create trust and satisfaction for both administrators and customers.
- Minimize business risk by preventing critical errors before systems are officially deployed.

2. Business context

Bakery Website has basic features for Importing and Purchasing

With **Import Goods**, administrators can add, edit, and delete products in the system. After adding or editing products, the system will automatically update the product list with information such as product name, description, price, inventory quantity, and image. When necessary, administrators can delete products that are no longer for sale. Changes will be automatically saved and updated in the system, helping to ensure that the product list is always accurate and complete.

With **Buy**, customers can add cakes to their cart from the product list or from the product detail page. After adding products to their cart, customers can view their updated cart with information such as the total cost of the cake, and the total amount to be paid. Customers can adjust their cart by updating quantities or removing items. When ready, customers can proceed to checkout to complete their purchase.

Some initial setup work needs to be completed when the Bakery Website is launched, including creating an administrator account, two customer accounts, and sample data for bakery products, inventory, and reviews for a few cakes.

3. Business process

3.1. Import process row

Add product:

- **The manager** enters product information (name, price, image, description, quantity in stock, category, brand).
- Information is checked for validity.
- The product is added to the database and displayed in the product list.

Product Update :

- **The administrator** selects the product to edit.
- Update information (name, price, image, quantity in stock...).

- Update changes to the database and display the new information on the interface.

Delete product:

- **The administrator** selects the product to delete.
- The system checks if the product is in an unprocessed order:
 - If yes: No can be deleted.
 - If not: Delete the product from the database.

3.2. Purchasing process

Add products to cart:

- **User** selects product and quantity to buy.
- Product added to cart:
 - If the product is already in the cart: Update quantity.
 - If not: Add product to cart list.
- Show list of products in cart with total quantity
- The system automatically calculates the total price based on the quantity and price of the products in the shopping cart.

Update quantity of products in cart:

- **User** adjusts the quantity of products in the cart.
- Inventory checking system:
 - If sufficient: Update cart.
 - If not: Notify and keep the old quantity.

Delete product in Cart:

- **User** selects product to delete.
- The product is removed from the cart and the total is updated.

Bar shopping cart :

- **User** confirms order in cart.
- Testing system :
 - Delivery information (address, phone number).
- If eligible:
 - Payment successful.
 - The system reduces the quantity of products in stock accordingly.
 - Create order code and save to purchase history.
- If not eligible:
 - Report the error and keep the cart.

4. Problem solving subsystems

4.1. Import process

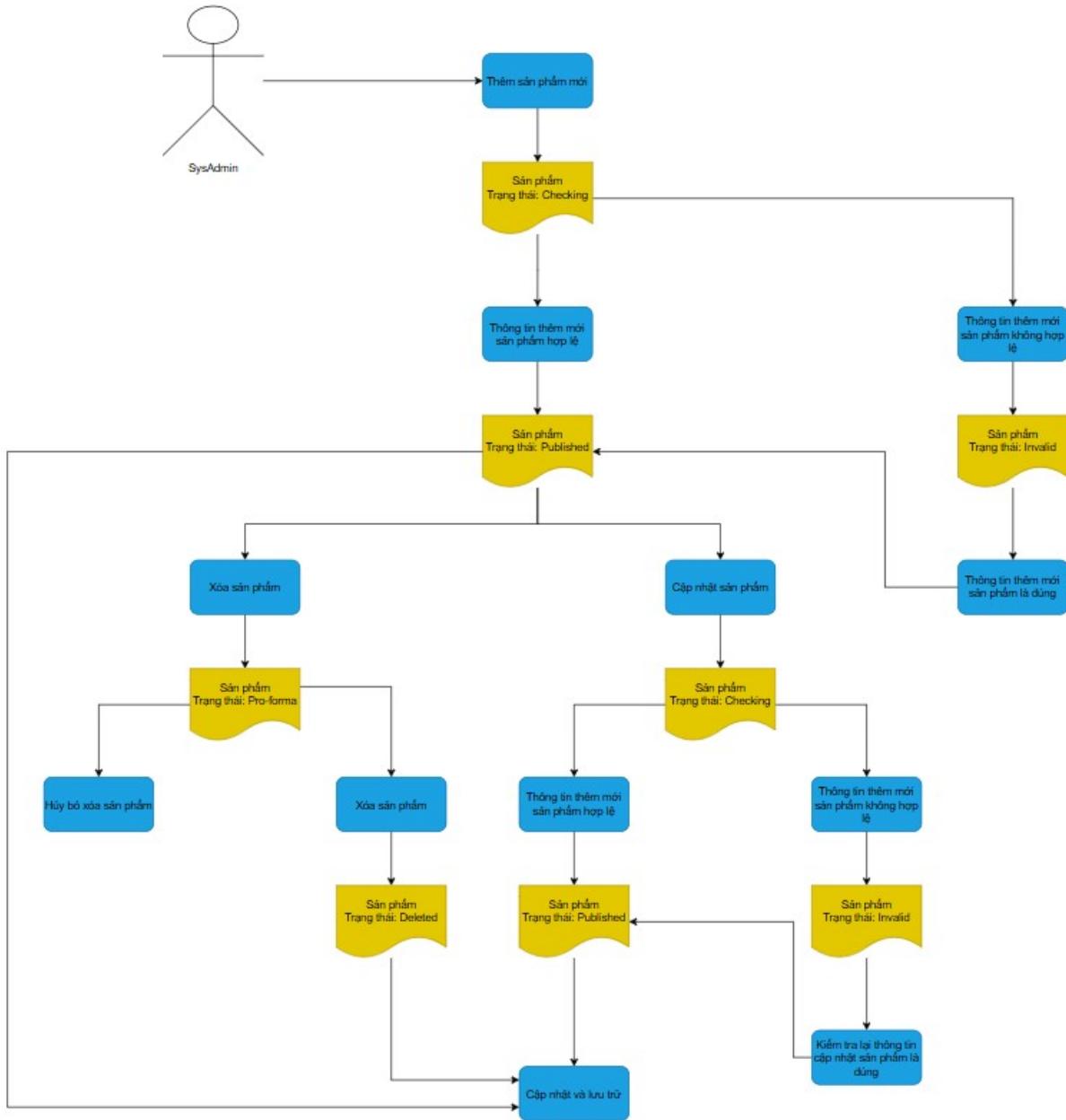


Figure 1. Import process diagram

4.2. Purchasing process

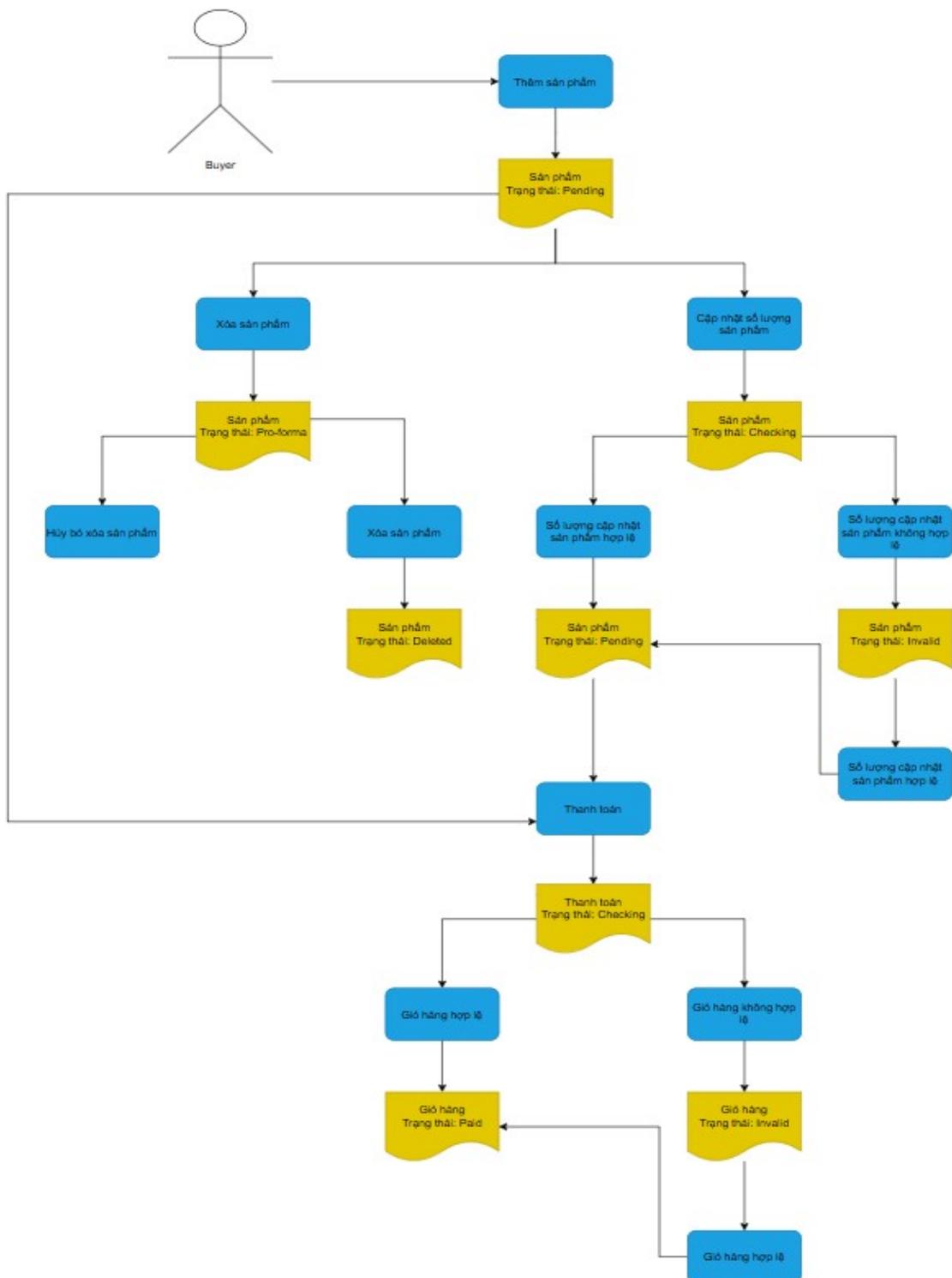


Figure 2. Purchasing process diagram

5. Conceptual Model

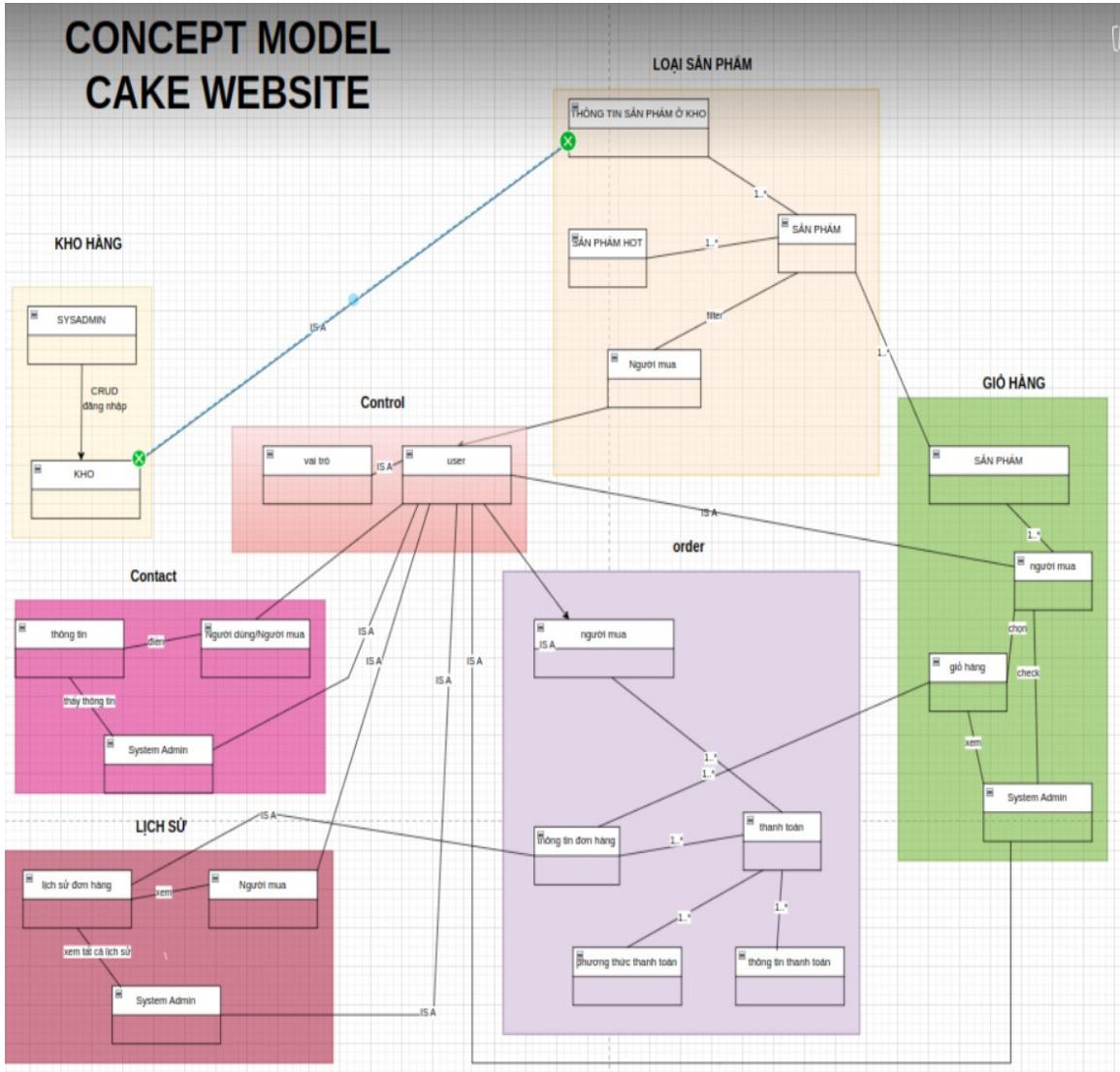


Figure 3. Conceptual model

6. User Stories

Import goods

- **As an administrator**, I want to add products to the system, including name, category, brand, selling price, import price, description, image and stock quantity.
- **As an administrator**, I want to delete a product from the system when it is no longer available for sale or out of stock.
- **As an admin**, I want to edit product information, including updating price, description, inventory quantity, images, and other attributes.
- **As a system**, I want to check the input data when adding or editing products to ensure the information is valid.
- **As a system**, I want to save product edit history so that I can look up and check past changes.

Purchase

- **As a buyer**, I want to buy any product on the product category page.
- **As a buyer**, I want to see product details to decide to buy the product. If I like the product and it is in stock, I can add it to my cart.
- **As a buyer**, I want to see the list of products in the cart and the total cart cost.
- **As a buyer**, I want to update the quantity of products in the shopping cart. If the product is out of stock The system will give an error message and not update the quantity . When I increase or decrease the quantity, the total cost of the cart will be automatically updated.
- **As a buyer**, I want to remove unnecessary products from the shopping cart. After deletion, the total cost and product list will be updated immediately.
- **As a buyer**, I want to check out my cart. If the cart is empty (product quantity is 0), I will get a message that I cannot check out.
- **As a buyer**, I want to save my shopping cart information so I can return to shopping after I have not paid.

- **As a system** , I will send a confirmation email to the buyer upon successful purchase .

7. Business requirements

7.1. Functional requirements

Request Code	Requestor	Request content	Essential level
WB1	Administrator	Import goods : Add, delete, edit products.	High
WB1.1	Administrator	Add products with information: name, category, brand, price, description, quantity, image.	High
WB1.2	Administrator	Delete product, only do this when there are no related orders.	High
WB1.3	Administrator	Edit product information, save change history.	High
WB2	User	Purchase : Add products, update quantity, delete products, checkout	High
WB2.1	User	More products in cart	High
WB2.2	User	Update quantity of products in cart, check inventory.	High
WB2.3	User	Remove product from cart .	High
WB2.4	User	Confirm information before payment .	High
WB2.5	User	After successful payment, reduce the quantity of products in stock accordingly.	High

7.2. Non-functional requirements

Request Code	Requestor	Description of requirements	Priority level

WB5	Administrator	Load Handling	High
WB5.1	Administrator	The system must be able to handle a large number of concurrent users without any loss of performance.	High
WB6	Administrator	Security	High
WB6.1	Administrator	The system must use strong authentication mechanisms (e.g. 2FA – 2-factor authentication) when users log in and perform important transactions.	High
WB6.2	Administrator	The system must have protections against attacks such as SQL Injection, XSS, CSRF, DoS/DDoS.	High
WB6.3	Administrator	All user connections and transactions must be encrypted and protected via HTTPS.	High

8. System participants

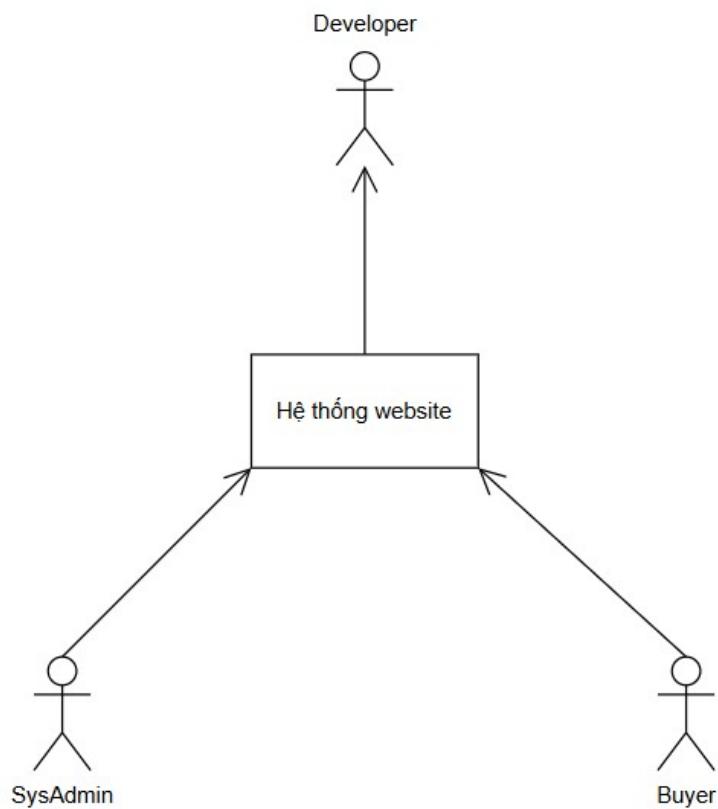


Figure 4. System participants

CHAPTER 2: SOFTWARE REQUIREMENTS SPECIFICATION

1. Deployment diagram

1.1. Import goods

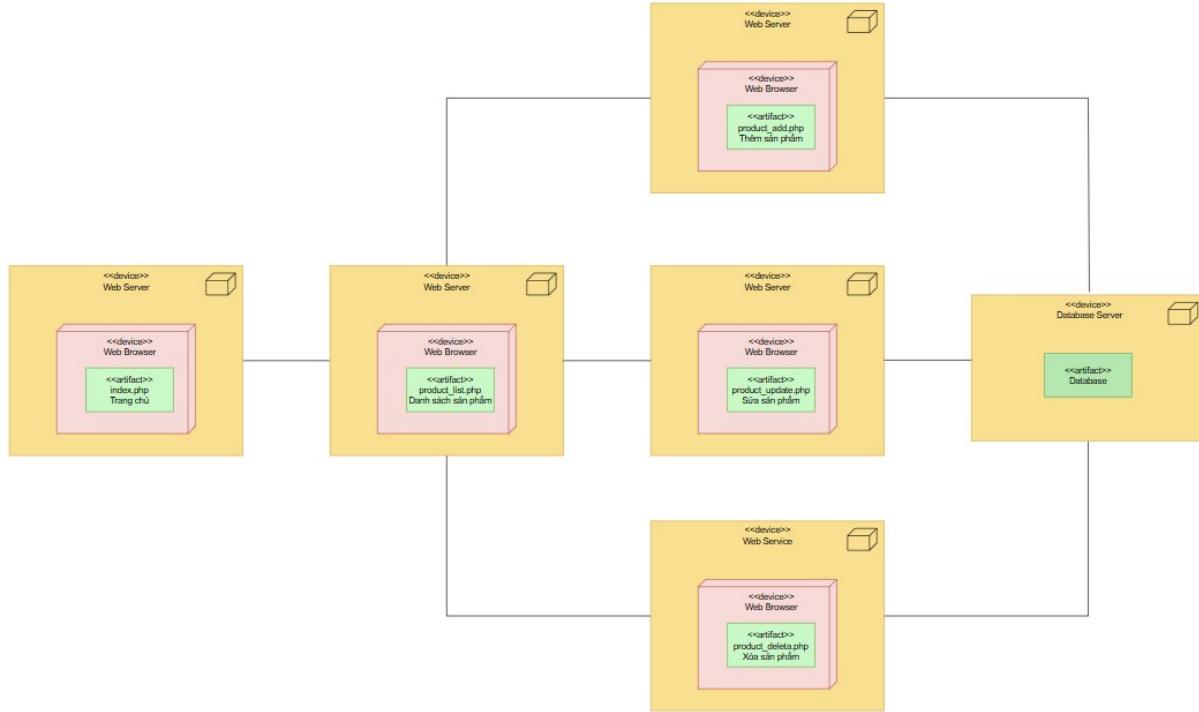


Figure 5. Import Deployment Diagram

1.2. Purchase

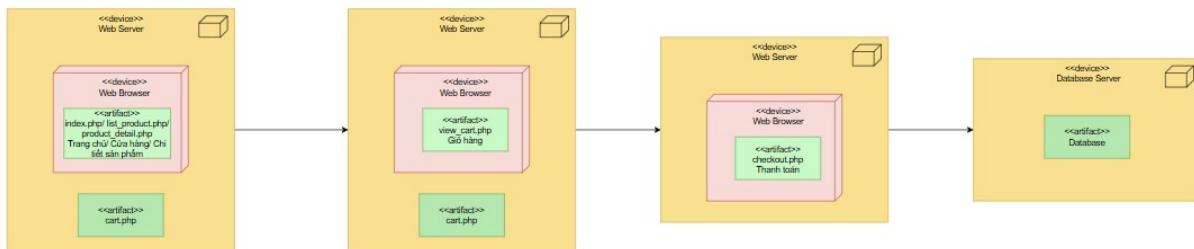


Figure 6. Purchasing Deployment Diagram

2. Overview diagram of system functions

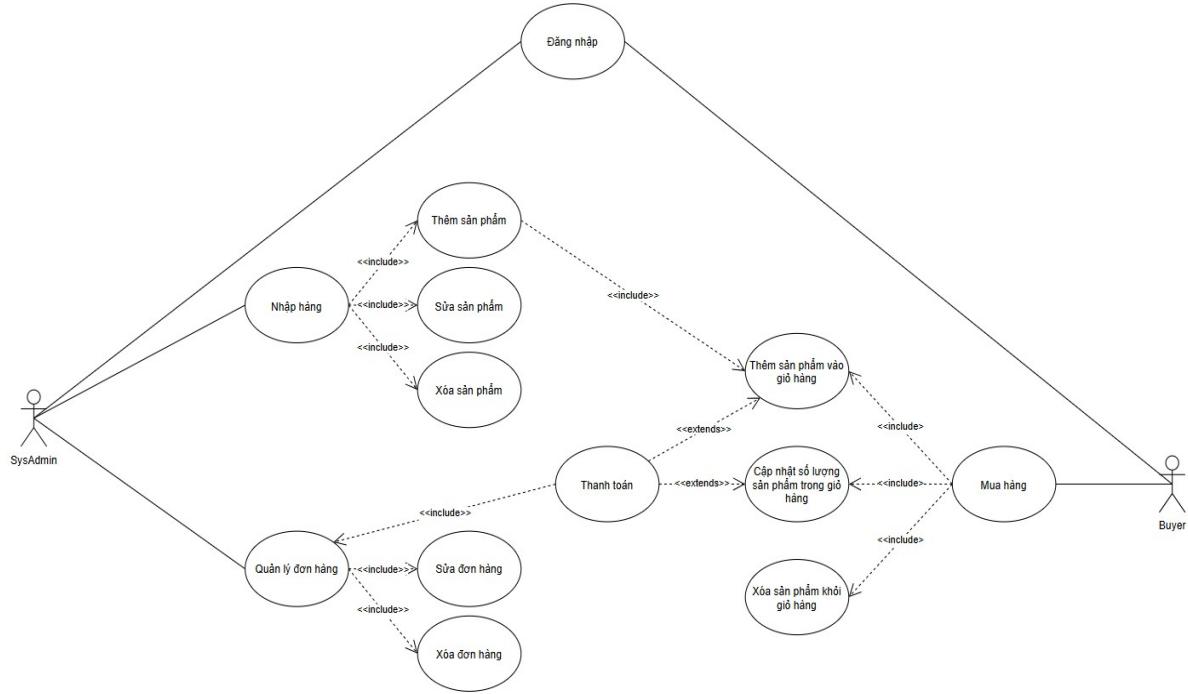


Figure 7. General UseCase

3. General database diagram

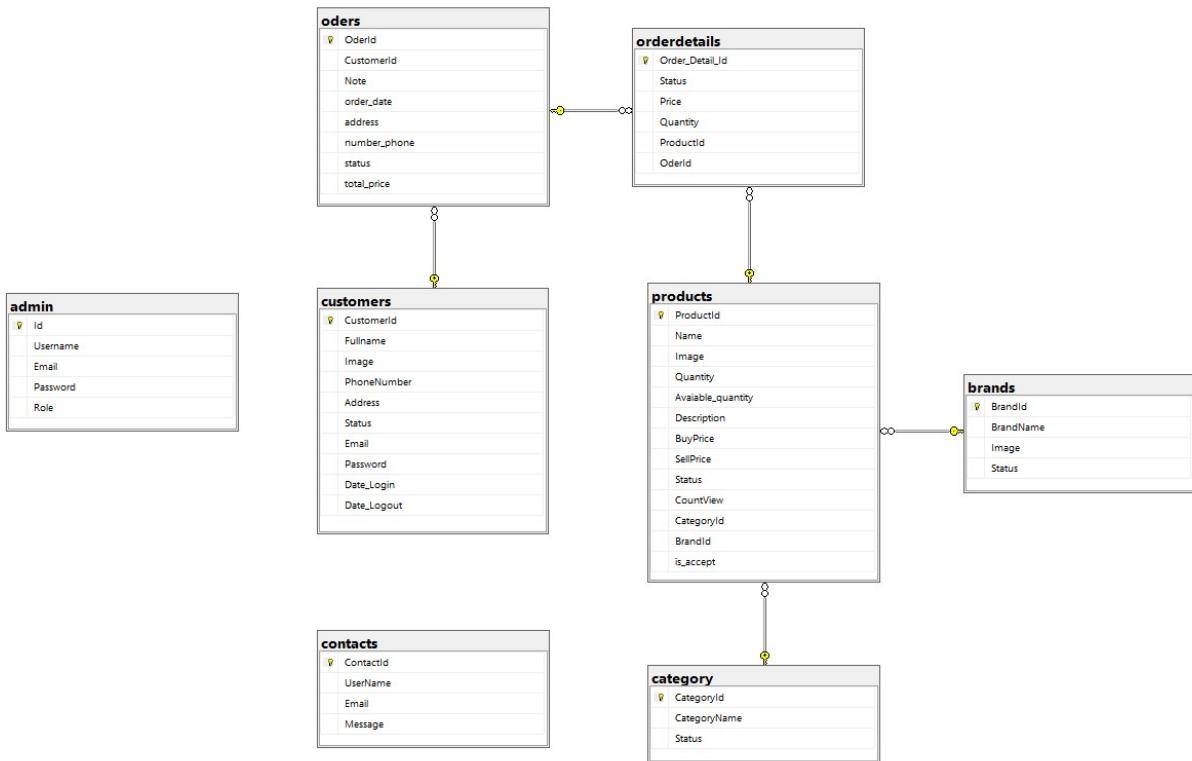
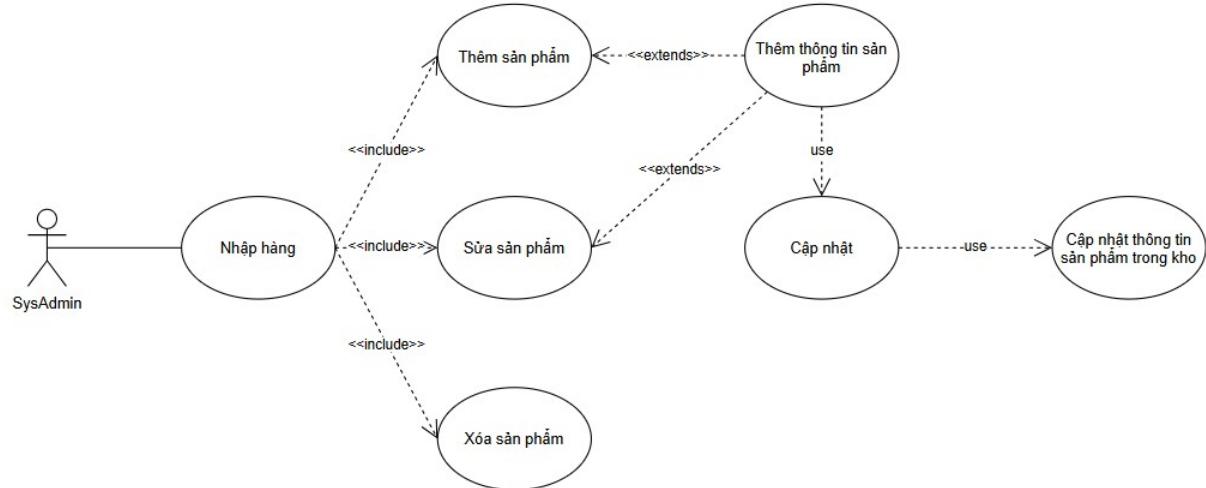


Figure 8. Database Diagram

4. Import goods

4.1. Focused UseCase



Use case number	UC1	
Use case name	Import and export	
Actor(s)	Manager	
Maturity	Focused	
Summary	<i>Use case describes the process of importing and exporting goods of the manager in the system.</i>	
Basic Course of events	Actor Action	System response
	1. Execute (Login Authentication)	
		2. Home page display system
	3. The Use Case begins when the Actor selects “ Product ”	
		4. The system displays a list of products.
	1.Actor clicks “Add new” A1 A2	
		2. The system displays the form

	to add new products.
3. Actor enters new information	
	4. Implementation system {Check text box} E 1
	5. System updates data into database
Alternative Paths	A1
	Actor Action
	System response
	1. Actor clicks “Edit”
	2. The system displays the product update form
	3. Actor enters updated information
	4. Implementation system {Check text box} E 1
	5. System updates data into database
	A2
	Actor Action
A2	System response
	1. Actor clicks “Delete”
	2. The system requires confirmation.
	3. Actor selects “OK” A4 confirmation
	6. The system updates data into the database.
A4	

	Actor Action	System response
	1. Actor can click “ Cancel” to operate	
		2 .The system deletes the confirmation request . Return to step 4 of Basic Course of Events.
Exception Path	E1 . At {Check text cell} , check if the text cell is empty, has the correct data type and constraint, if so, report the corresponding error message	
Triggers:	When actor wants to import or export goods	
Assumption:	None	
Preconditions:	None	
Post Conditions:	None	

4.2. Activity Diagram

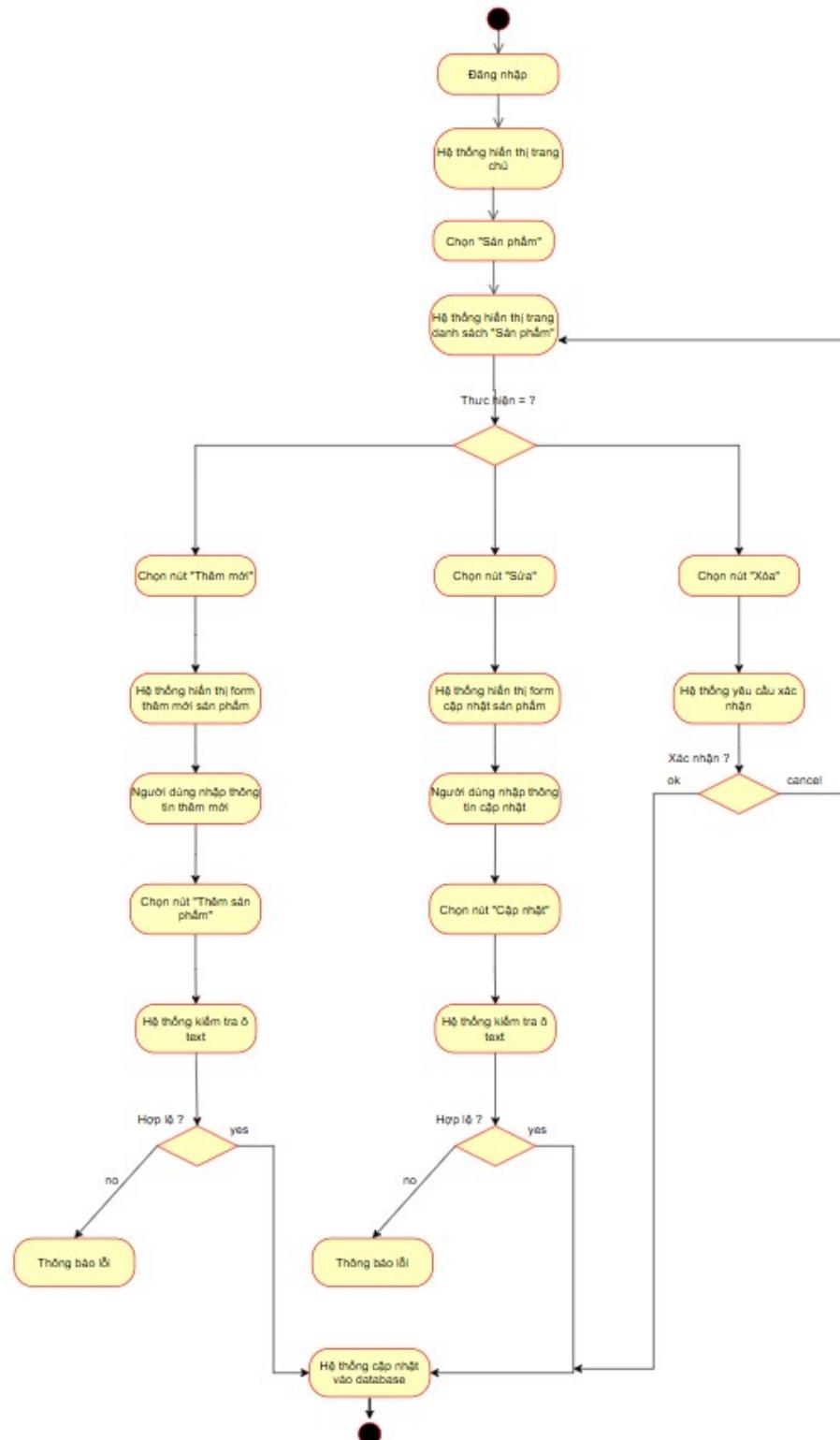


Figure 9. Activity diagram of the Import process

4.3. Sequence Diagram

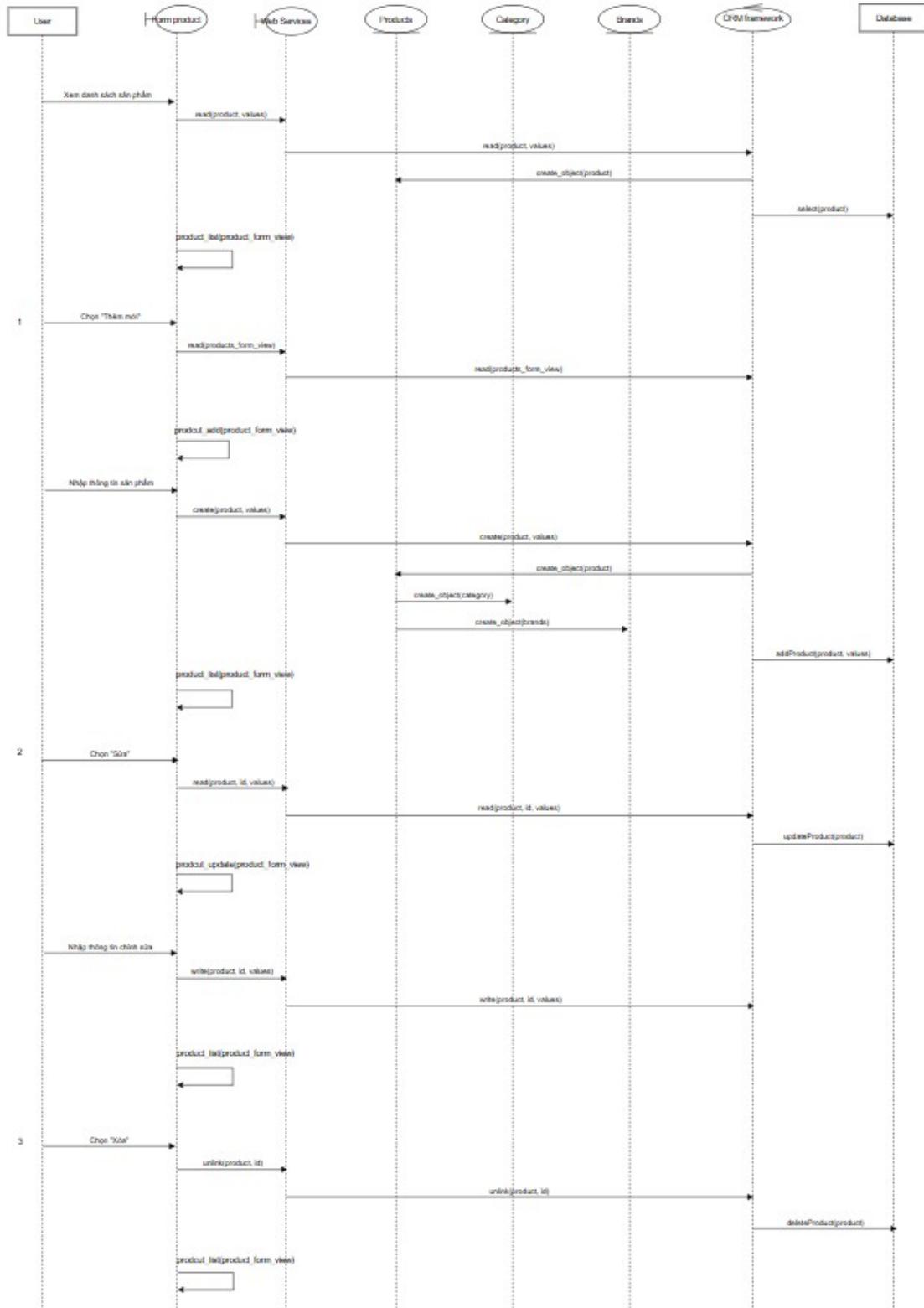


Figure 10. Sequence diagram of the Import process

4.4. Class Diagram

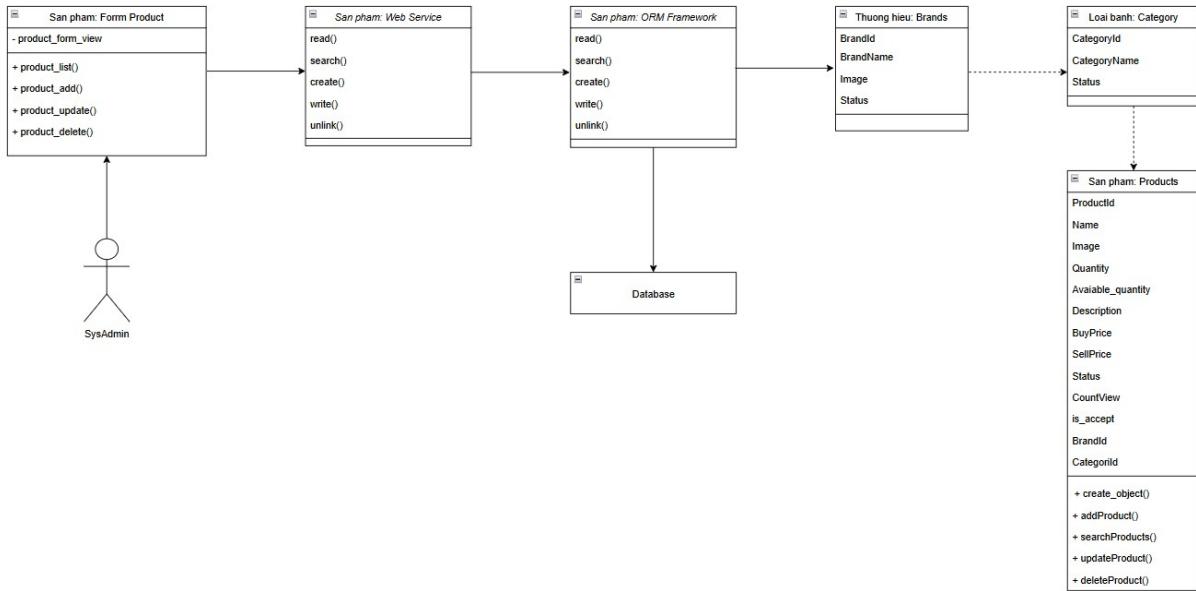


Figure 11. Class diagram of the Import process

4.5. Interface illustration

Delete product

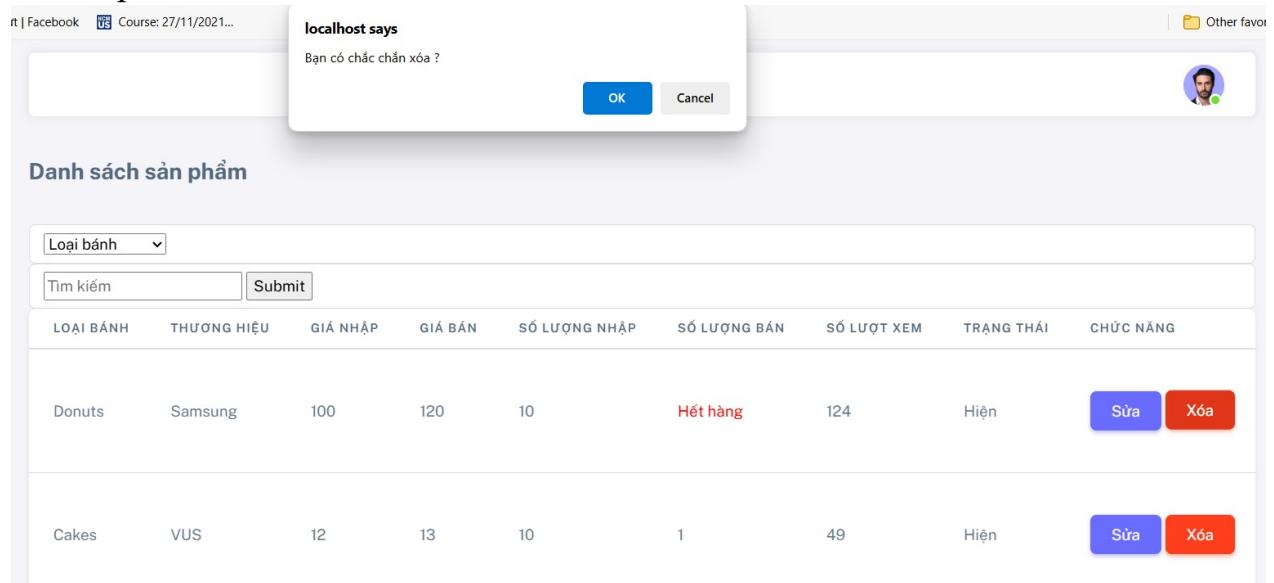


Figure 12. Product deletion interface Import goods

Add product

TÊN SẢN PHẨM
Bánh kem Le Castella

THƯƠNG HIỆU
-----Loại thương hiệu-----

LOẠI BÁNH
-----Loại bánh-----

HÌNH ẢNH SẢN PHẨM
Choose File No file chosen

GIÁ NHẬP

GIÁ BÁN

SỐ LƯỢNG NHẬP

SỐ LƯỢNG BÁN

MÔ TÀ

TRẠNG THÁI
 Hiện
 Ẩn

[Thêm sản phẩm](#)

Figure 13. Add product interface Import goods

Product Updates

TÊN SẢN PHẨM
DULCE DE LECHE

THƯƠNG HIỆU
Samsung

LOẠI BÁNH
Donuts

HÌNH ẢNH
Choose File No file chosen



GIÁ NHẬP
100

GIÁ BÁN
120

SỐ LƯỢNG NHẬP
10

SỐ LƯỢNG BÁN
0

MÔ TẢ
bánh kem ngon lâm, mua nhanh đi mọi người

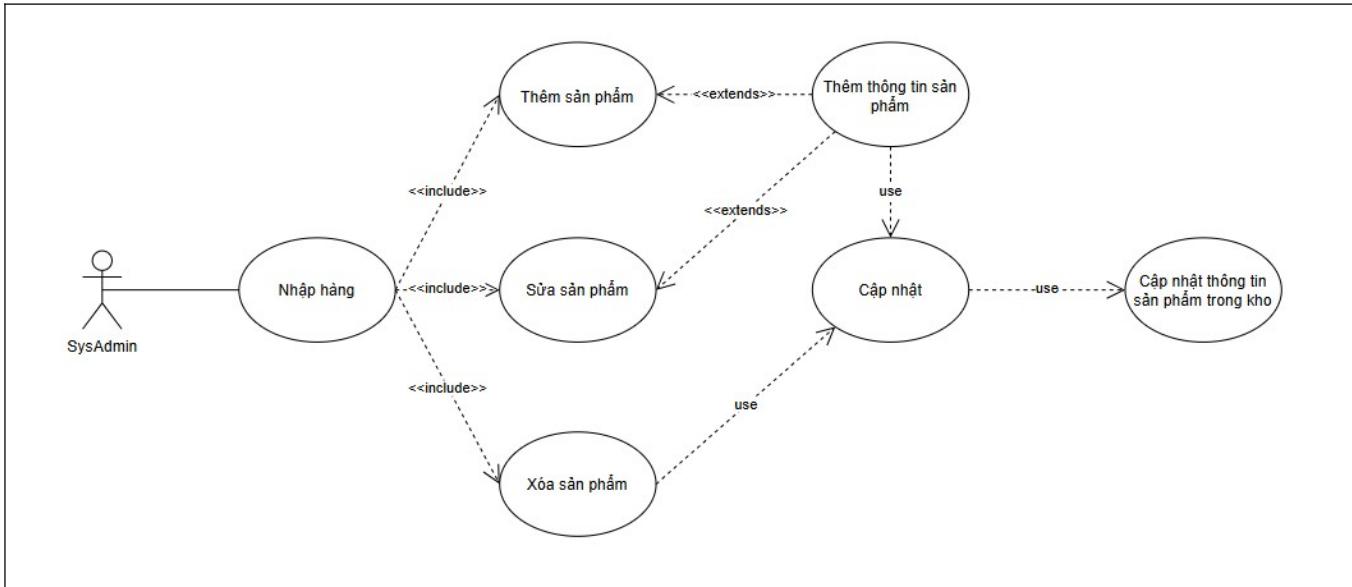
TRẠNG THÁI
 Hiện
 Ẩn

Cập nhật

Figure 14. Product update interface Import goods

5. Purchase

5.1. Focused UseCase



Use case number	UC2	
Use case name	Shopping Cart	
Actor(s)	Shopper	
Maturity	Focused	
Summary	<i>Use case describes the user's purchasing process in the system.</i>	
Basic Course of events	Actor Action	System response
	1. Use Case starts when Actor selects "Add to cart" with default product A1	
		2. Implementation system {Check quantity} E 1
		3. The system displays a list of products added to the cart.
	4. Actor can make payment A2 A3	

	5. The system displays the shopping cart payment form
6. Actor enters payment information	
	7. Implementation system {Check text box} E 1
	8. The system updates order information into the database.
	9. The quantity of products in stock will decrease according to the order quantity.
Alternative Paths	A1
	Actor Action
	1. Actor clicks on the product image to add
	2. The system displays detailed product information including: Name, Price, Quantity, Cake type, Brand
	3. Actor customizes the number of products to add to cart
	4. Implementation system {Check quantity} E 1
	5. Actor selects “Add to cart”
	6. The system displays a list of products added to the cart.

A2	
Actor Action	System response
1. Actor fills in the field the number of products to update	
2. Actor clicks “Update”	
	3. Implementation system {Check quantity} E 1
	4. The system displays the updated number of products in the shopping cart.
A3	
Actor Action	System response
1. Actor clicks “Delete”	
	2. The system requires confirmation.
3. Actor selects “OK” A4 confirmation	
	6. The system displays products removed from the cart.
A4	
Actor Action	System response
1. Actor can click “Cancel” to operate	
	2 .The system deletes the confirmation request . Return to step 2 of Basic Course of Events.

Exception Path	E1 . At {Check quantity} , check if the quantity is enough for the product in stock or not, if there is a corresponding error message
Triggers:	When the actor wants to buy the item
Assumption:	None
Preconditions:	None
Post Conditions:	None

5.2. Activity Diagram

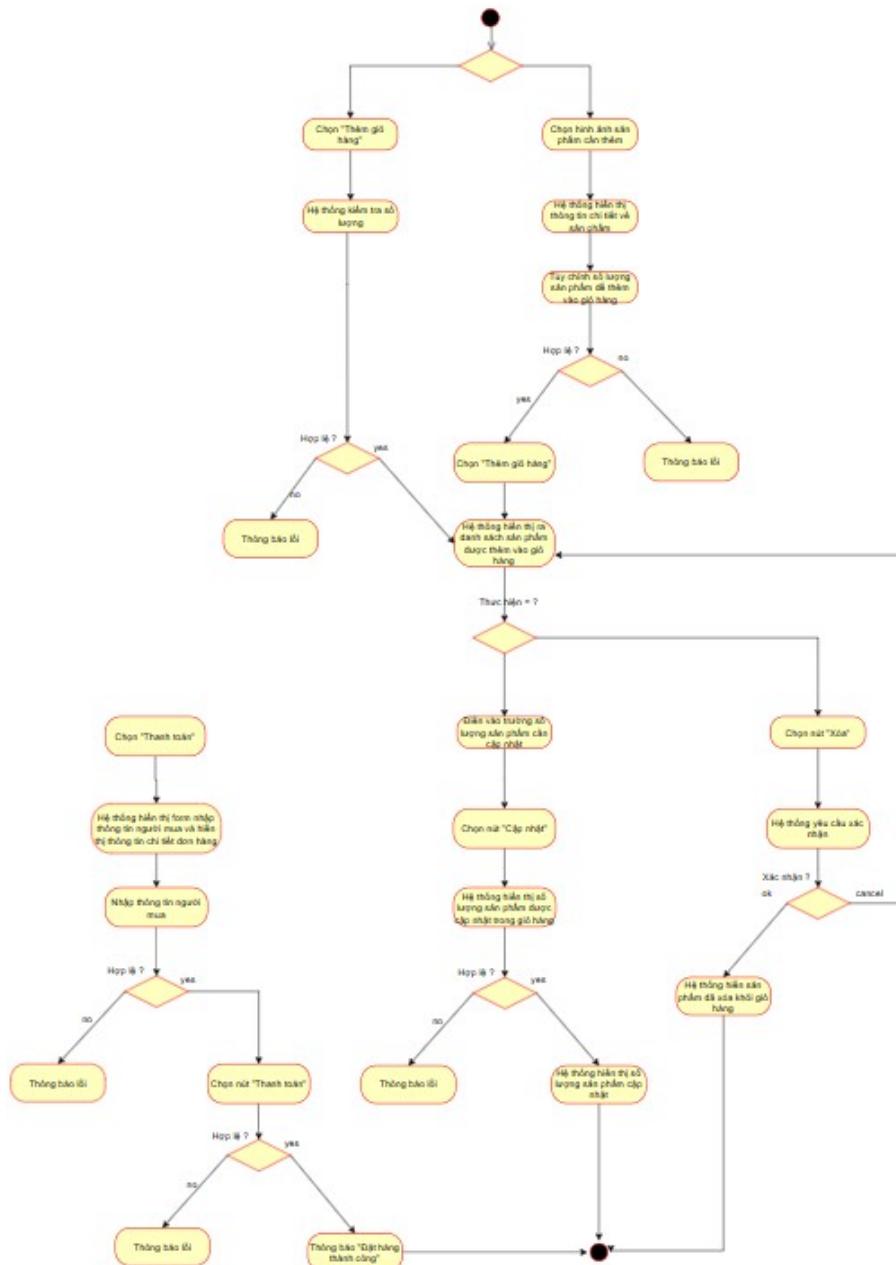


Figure 15. Activity diagram of Purchasing process

5.3. Sequence Diagram

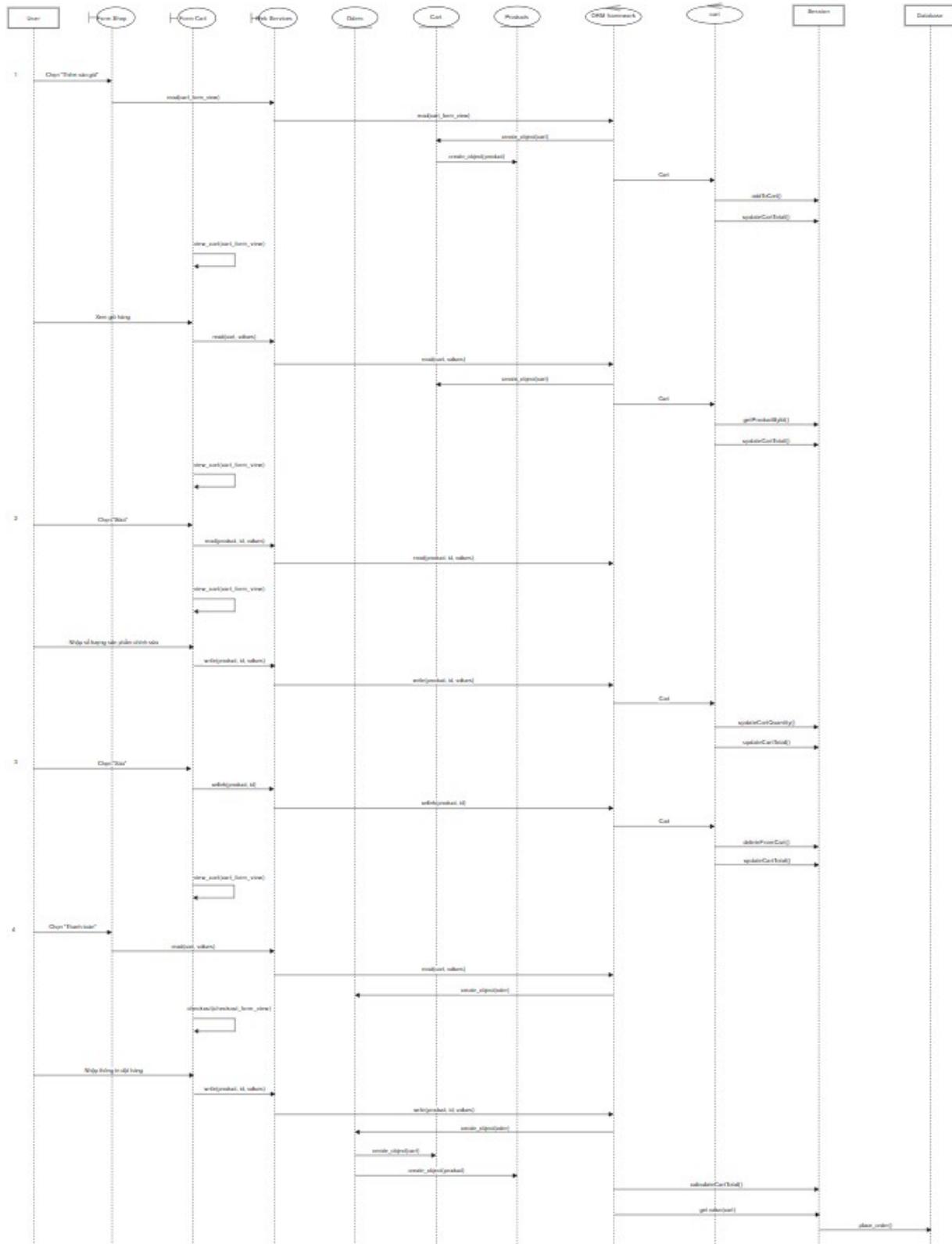


Figure 16. Purchase Process Sequence Diagram

5.4. Class Diagram

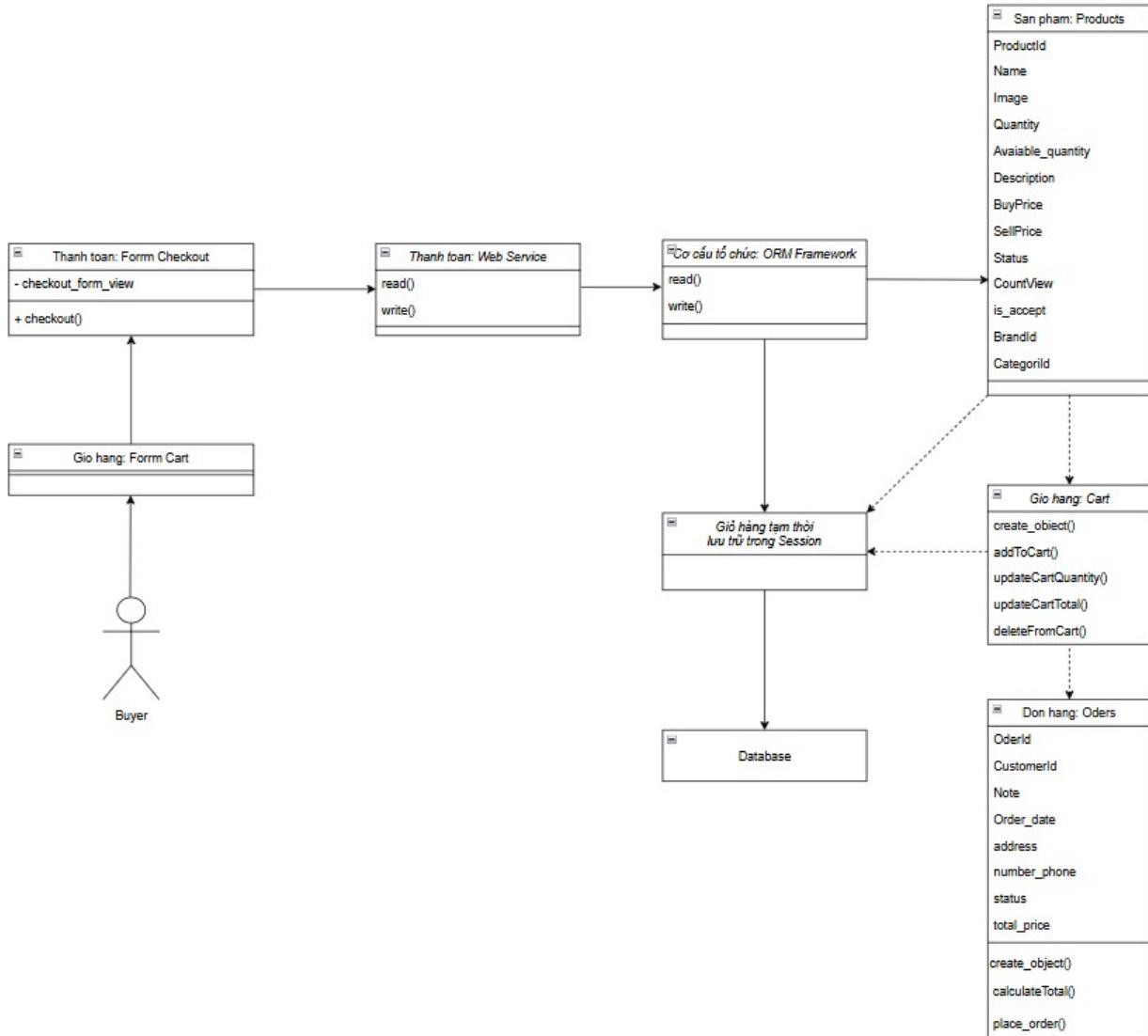


Figure 17. Purchase Process Class Diagram

5.5. Illustrative interface

Add product to cart

Cửa hàng

Trang chủ | Cửa hàng

Category ▾ | Search **Search** Default sorting ▾

Product	Price	Status	Action
DULCE DE LECHE	120USD	Hết hàng	Thêm giỏ hàng
DOZEN CUPCAKES	13USD		
DOZEN CUPCAKES	20USD	Hết hàng	
BÁNH KEM TOUS LES	3224USD	Hết hàng	

Figure 18. Default interface for adding a product to cart when purchasing

Chi tiết sản phẩm

Home | Shop |



DOZEN CUPCAKES

Giá: 13

hhjhj

Số lượng: 1

Loại bánh: Cakes

Thương hiệu: VUS

1

THÊM GIỎ HÀNG



Figure 19. Interface for adding products with optional quantity when purchasing

Update quantity of products in cart

The screenshot shows the homepage of a website for "cake". The top navigation bar includes links for "Trang chủ", "Cửa hàng", "Lịch sử", and "Liên hệ". On the right, there are links for "Tài Khoản" and a shopping cart icon with "(1)". Below the navigation is a banner with the text "Giỏ Hàng". The main content area displays a table of products in the cart, showing one item: "DOZEN CUPCAKES" with a quantity of "1". At the bottom, there is a green button labeled "Thanh toán".

Giỏ Hàng

Trang chủ | Cửa hàng |

STT	Tên sản phẩm	Hình ảnh	Số lượng	Đơn giá	Thành tiền	Chức năng
1	DOZEN CUPCAKES		<input type="text" value="1"/> Cập nhật	13	13 USD	Xóa

Tổng tiền

13 USD

Thanh toán

Figure 20. Interface for viewing and updating shopping cart

Remove product from cart

The screenshot shows a web browser window with a navigation bar at the top. A modal dialog box is centered, asking "Bạn có chắc chắn xóa ?" (Are you sure you want to delete?). Below the dialog, there are "OK" and "Cancel" buttons. The main content area shows a table of products in the cart. One row for "DOZEN CUPCAKES" is selected, with a quantity input field containing "1". A blue "Cập nhật" (Update) button is next to it. To the right of the quantity input is a red "Xóa" (Delete) button. At the bottom left of the page is a green "Thanh toán" (Checkout) button.

Figure 21. Interface to delete products from cart

Checkout cart

The screenshot shows a checkout page titled "Thanh Toán". On the left, there are input fields for "Họ và tên" (Name) with "nhut" entered, "Email" with "tannhut2111@gmail.com", "Địa chỉ" (Address) with "quận x, TP.HCM", and "Số điện thoại" (Phone Number) with "0123456789". Below these is a "Ghi chú" (Notes) text area with the placeholder "Hàng dễ vỡ xin nhẹ tay !!!". On the right, there is a summary table titled "Thông tin chi tiết đơn hàng" (Order details). It shows one item: "DOZEN CUPCAKES" with quantity "1" and unit price "13". The total amount "Tổng tiền" is "13 USD". At the bottom left is a green "Thanh toán" (Checkout) button.

Figure 22. Shopping cart payment interface

CHAPTER 3: TEST PLAN

1. Introduction

1.1. Purpose

This test plan aims to:

- Identify project information and software components to be tested.
- List the test requirements (at high level).
- Propose and describe a testing strategy.
- Identify resources and estimate testing effort.
- List the outputs of the testing project.

1.2. Background

The system developed is an online cake sales website, supporting users to choose and buy cakes with many different flavors and designs. The website provides services such as online ordering, online payment, shopping cart management, product search and filtering, as well as suggestions for featured cakes.

1.3. Scope

This test plan applies to all the requirements of the Cake-Sale-Website project defined in the Software Requirements Document. The testing scope includes all the main functions of the cake sales website, from product display, shopping cart management, ordering, to online payment and customer information management. In addition, the system must also be tested for the ability to search, filter products by category and support the suggestion of featured products. Factors related to user experience such as the interface , will also be tested. This plan also includes ensuring the website operates stably, testing normal use cases as well as exceptional cases, to ensure the system operates effectively and without errors during actual user use.

1.4. Project identification

The table below identifies the documents and resources used to develop the test plan, such as requirements documents, functional specification documents, and use-case statements.

The table below identifies the documents and availability used to develop the Test Plan:

Documentation (version / date)	Created or available	Received or reviewed	Author or Source	Note
Request	<input type="checkbox"/> Yes <input checked="" type="checkbox"/> No	<input type="checkbox"/> Yes <input checked="" type="checkbox"/> No		
Functional specification	<input type="checkbox"/> Yes <input checked="" type="checkbox"/> No	<input type="checkbox"/> Yes <input checked="" type="checkbox"/> No		
Use-Case Report	<input type="checkbox"/> Yes <input checked="" type="checkbox"/> No	<input type="checkbox"/> Yes <input checked="" type="checkbox"/> No		
Project plan	<input type="checkbox"/> Yes <input checked="" type="checkbox"/> No	<input type="checkbox"/> Yes <input checked="" type="checkbox"/> No		
Design	<input type="checkbox"/> Yes <input checked="" type="checkbox"/> No	<input type="checkbox"/> Yes <input checked="" type="checkbox"/> No		
Prototype	<input type="checkbox"/> Yes <input checked="" type="checkbox"/> No	<input type="checkbox"/> Yes <input checked="" type="checkbox"/> No		
User manual	<input type="checkbox"/> Yes <input checked="" type="checkbox"/> No	<input type="checkbox"/> Yes <input checked="" type="checkbox"/> No		
Business model or process	<input type="checkbox"/> Yes <input checked="" type="checkbox"/> No	<input type="checkbox"/> Yes <input checked="" type="checkbox"/> No		
Data model or process	<input type="checkbox"/> Yes <input checked="" type="checkbox"/> No	<input type="checkbox"/> Yes <input checked="" type="checkbox"/> No		
Business functions and rules	<input type="checkbox"/> Yes <input checked="" type="checkbox"/> No	<input type="checkbox"/> Yes <input checked="" type="checkbox"/> No		
Project or business risk assessment	<input type="checkbox"/> Yes <input checked="" type="checkbox"/> No	<input type="checkbox"/> Yes <input checked="" type="checkbox"/> No		

1.5 References

STT	Document	Describe
1	ABC Project Software Requirements Document, Version 1.0.3.3, Monday, August 25, 2010	The SRS document fully describes the external behavior of the application or subsystem being defined. It also describes non-functional requirements, design constraints, and other elements necessary to provide a complete and comprehensive description of the software requirements.
2	ABC Project Software Development Plan,	The purpose of the Software Development Plan is to define the ABC project tasks (known as IMs or

	Version 09/05/2010	1.0.5, projects) to be developed. It focuses on the capabilities that stakeholders, especially the target users, are looking for and the reasons for this search.
--	-----------------------	---

2. Features to test

2.1. Function (Customer)

2.1.1. Add products to cart

- Verify that the system allows customers to select products from the catalog and add them to the cart .
- Verify that each time a product is added, the system saves and updates the corresponding quantity and total in the shopping cart.

2.1.2. Remove products from cart

- Verify that the system provides functionality to remove unwanted products from the shopping cart.
- Verify that when you select delete, the product will be removed from the cart ..

2.1.3. Update products in cart

- Verify that the system allows changing the quantity of products in the shopping cart.
- Verify that when you press update, the system recalculates the product quantity and total price.

2.1.4. Shopping cart payment

- Verify that the system displays the selected products in the shopping cart and the total price.
- Verify that the system allows customers to checkout for the products selected in the shopping cart.

2.2. Function (Manager)

2.2.1. Add products to inventory

- Verify that the system allows the manager to add new products to inventory.
- Verify that product information including name, price, image, quantity, and other attributes are entered completely and stored accurately.

2.2.2. Remove product from inventory

- Verify that the system allows managers to remove out-of-stock products from inventory.
- Verify that the deleted product is no longer visible in the product catalog.

2.2.3. Update products in stock

- Verify that the system allows managers to update product information.
- Verify that information such as price, quantity, image, and description are updated correctly and display correctly after saving.

2.3. Availability

- Verify that the system supports displaying the pie image icon in the interface.
- Verify that the user interface is friendly, easy to use, and intuitive.

2.4. Design limitations

2.4.1. Message length

- Verify that the system does not allow messages or notifications to exceed the specified maximum length.

2.4.2. Testing environment

- Verify that the system works stably on popular browsers and on both mobile and desktop devices.

2.4.3. Language

- Verify that the system supports displaying Vietnamese correctly and without font errors.

2.4.4. Multi-device

- Verify that the system displays correctly and is compatible on both computers and phones with different screen sizes.

2.5. Interface

2.5.1. User interface

- Verify that the main page interface provides prominent product information, including images, prices, and an “Add to Cart” button.
- Verify that the shopping cart interface displays the selected products, allows for updating quantities, deleting products, and checking out.
- Verify that the product details view displays complete product information including image, description, price, quantity, cake type, and brand name.

2.5.2. Manager interface

- Verify that the cake type management interface allows adding, modifying, and deleting cake types.

3. Features that do not need testing

3.1. Function (Customer)

3.1 .1 . Contact the administrator

- Verify that the system allows customers to submit contact information or inquiries about products or services.
- Verify that contact information/inquiries are sent to the correct manager for response.

3.1 .2 . Search for products

- Verify that customers can search for products by name via the search bar.
- Verify that search results are returned accurately based on the entered keyword.

3.1 .3 . Product filter

- Verify that customers can filter products by cake type.
- Verify that the product list displays correctly according to the filter criteria.

3.1 .4 . View purchase history

- Verify that the system displays purchase history including product details, price, delivery status, and related information.
- Verify that customers can only view their own purchase history.

3.1 .5 . Track order status

- Verify that customers can track their order status as the manager updates.
- Verify that order statuses are displayed accurately and clearly.

3.1 .6 . View products in cart

- Verify that customers can see the list of products added to the cart.
- Verify that the system displays complete product information in the shopping cart (name, quantity, price, total).

3.1 .7 . View product details

- Verify that the system displays detailed information for each product, including product name, description, image, price, stock quantity, cake type, and brand.
- Verify that product details are fully updated and accurate.

3.2. Function (Manager)

3.2 . 1. Create a user account

- Verify that the system provides user account creation functionality for the bakery service.
- Verify that the administrator can create multiple user accounts.
- Verify that the system supports the "Remember password" option for each user account.

3.2 . 2. Delete user account

- Verify that the system allows deletion of user accounts from the system.
- Verify that the deleted account will not be able to access or use the service.

3.2 . 3. Create an administrator account

- Verify that the system provides the ability to create manager accounts to manage stores and products.

3.2 . 4 . Featured product suggestions

- Verify that the system automatically recommends featured products based on customer views.
- Verify that the featured product list changes dynamically based on real-world data.

3.2 .5 . Filter products in stock

- Verify that the system allows filtering of products in inventory by criteria such as cake type, brand, or stock status.

3.2.6. Add cake type

- Verify that the administrator can add new cakes to the system.
- Verify that the cake type is correctly categorized to display on the website.

3.2.7. Delete cake type

- Verify that the system allows deletion of out-of-sale cakes.
- Verify that the deleted cake type is no longer visible in the catalog.

3.2.8. Update cake type

- Verify that the manager can update the cake type information, including the cake type name.
- Verify that the updated cake type information is displayed correctly.

3.2.9. Add brand

- Verify that the system allows adding new brands, including brand names and images.

3.2.10. Remove brand

- Verify that the brand removal support system is no longer in business or partnership.
- Verify that the removed brand will no longer appear in the catalog.

3. 2.11. Brand Update

- Verify that the manager can edit brand information, including brand name and image.

3.2.12. View user information

- Verify that managers can view customer account information, including purchase history and contact information.
- Verify that only administrators have access to this information.

3.2.13. View user contact information

- Verify that the system allows managers to view and manage customer contact information.
- Verify that contact information is displayed completely and accurately.

3.2.14. Search for products in stock

- Verify that managers can search for products in inventory based on product name, product code, or other criteria.

3.2.15. View detailed order information

- Verify that the manager can view full details of each order, including buyer information and products purchased.

3.2.16. Update order status

- Verify that the system allows order status updates (pending, processing, processed, shipped).
- Verify that the order status displayed matches the actual situation.

3.2.17. Delete order

- Verify that the system supports deleting invalid or canceled orders.
- Verify that deleted orders are no longer visible in the system.

4. Testing strategy

4.1. Types of testing

Test Objective	Ensure that the target functionality under test works properly, including navigation, input, processing, and retrieval.
Technique	Execute each use case, use case flow, or function, using valid and invalid data, to verify the following: - Expected results occur when valid data is used. - Appropriate error or warning messages are displayed when invalid data is used.
Completion criteria	- All tests have been performed. - All identified bugs have been fixed.
Special Considerations	Identify or describe the items or issues (internal or external) that affect the implementation and execution of functional testing.

4.2. Tools

Testing tools include manual testing and automated testing

Tools	Supplier/Internal	Version
Manual Testing	Internal	
Auto Testing	Internal	

5. Resources

The table below shows the personnel required for the project.

Human resources	Describe specific responsibilities/jobs
Test Developer	Identify, prioritize and execute test cases.
	Responsibility: - Create test plan - Evaluate the effectiveness of testing efforts
Tester	Perform the tests.

	Responsibility: <ul style="list-style-type: none"> - Develop test case specifications - Develop test cases
	<ul style="list-style-type: none"> - Perform testing - Record the results - Monitor bug fixes - Request to change documents

6. Project milestones

The project consists of three development phases with full testing cycles from planning, design, development, implementation and evaluation.

Test Milestone Task	Effort	Start Date	End Date
Construction 1			
Test Plan	123.7	11/01/2024	11/01/2024
Test Design	300 .5	11/10/2024	11/15/2024
Test Execution	623 .3	11/20/2024	11/30/2024
Test Evaluation	216 .1	01/12/2024	02/12/2024

7. Output products

7.1 Testing model

The test model defines all the test cases, references, test procedures and test scenarios associated with each test case.

7.2 Test log

Use Microsoft Word to record and report test results, ensuring detailed and clear tracking of test activities and results.

CHAPTER 4: TESTING METHODS

1. Unit Testing

Unit testing is a software testing method in which the smallest units of source code (usually functions or methods) are tested independently to ensure that they function as expected. The goal of unit testing is to detect errors early in the basic components of the system while they are still simple, preventing errors from spreading into more complex parts of the program.

Purpose of Unit Testing:

- **Verify accuracy** : Ensure that the smallest units of the system operate as expected.
- **Early Error Detection** : Errors are detected early in the development process, reducing the cost of fixing them later.
- **Increased maintainability** : Unit testing makes it easier to maintain source code, as changes can be tested immediately.
- **Improve software quality** : Increase software reliability and quality by ensuring that code units function correctly.
- **Support for code refactoring** : With unit tests, programmers can refactor source code without fear of disrupting existing functionality.

1.1. TestPlan

No	Unit ID	Unit Description	Testing Environment	Owner	On build/iterations?	Comments
		Function addProduct()	PHPUnit	Phùng Tân Nhứt	Build 1.0	
1	UT-001	Kiểm tra khi giá nhập là số âm	PHPUnit	Phùng Tân Nhứt	Build 1.0	
2	UT-002	Kiểm tra khi giá bán là số âm	PHPUnit	Phùng Tân Nhứt	Build 1.0	
3	UT-003	Kiểm tra khi số lượng nhập là số âm	PHPUnit	Phùng Tân Nhứt	Build 1.0	
4	UT-004	Kiểm tra khi số lượng bán là số âm	PHPUnit	Phùng Tân Nhứt	Build 1.0	
5	UT-005	Kiểm tra khi số lượng bán lớn hơn số lượng nhập	PHPUnit	Phùng Tân Nhứt	Build 1.0	
6	UT-006	Kiểm tra khi thêm sản phẩm với tên bị trùng	PHPUnit	Phùng Tân Nhứt	Build 1.0	
7	UT-007	Kiểm tra khi dữ liệu đầy đủ và hợp lệ	PHPUnit	Phùng Tân Nhứt	Build 1.0	
		Function updateProduct()	PHPUnit	Trần Gia Phú	Build 1.0	
8	UT-008	Kiểm tra cập nhật sản phẩm với ID không tồn tại	PHPUnit	Trần Gia Phú	Build 1.0	
9	UT-009	Kiểm tra cập nhật sản phẩm với tên bị trùng	PHPUnit	Trần Gia Phú	Build 1.0	
10	UT-010	Kiểm tra cập nhật sản phẩm với tên hợp lệ	PHPUnit	Trần Gia Phú	Build 1.0	
11	UT-011	Kiểm tra cập nhật sản phẩm với hình ảnh hợp lệ	PHPUnit	Trần Gia Phú	Build 1.0	
12	UT-012	Kiểm tra khi số lượng nhập là số âm	PHPUnit	Trần Gia Phú	Build 1.0	
13	UT-013	Kiểm tra khi số lượng bán là số âm	PHPUnit	Trần Gia Phú	Build 1.0	
14	UT-014	Kiểm tra cập nhật số lượng bán hợp lệ	PHPUnit	Trần Gia Phú	Build 1.0	
15	UT-015	Kiểm tra khi số lượng bán lớn hơn số lượng nhập	PHPUnit	Trần Gia Phú	Build 1.0	
16	UT-016	Kiểm tra khi giá nhập là số âm	PHPUnit	Trần Gia Phú	Build 1.0	
17	UT-017	Kiểm tra cập nhật sản phẩm với giá nhập hợp lệ	PHPUnit	Trần Gia Phú	Build 1.0	
18	UT-018	Kiểm tra khi giá bán là số âm	PHPUnit	Trần Gia Phú	Build 1.0	
19	UT-019	Kiểm tra khi cập nhật sản phẩm với giá bán hợp lệ	PHPUnit	Trần Gia Phú	Build 1.0	
20	UT-020	Kiểm tra cập nhật sản phẩm với mã từ hợp lệ	PHPUnit	Trần Gia Phú	Build 1.0	
21	UT-021	Kiểm tra cập nhật sản phẩm với loại bánh hợp lệ	PHPUnit	Trần Gia Phú	Build 1.0	
22	UT-022	Kiểm tra cập nhật sản phẩm với thương hiệu hợp lệ	PHPUnit	Trần Gia Phú	Build 1.0	
		Function deleteProduct()	PHPUnit	Phùng Tân Nhứt	Build 1.0	
23	UT-023	Kiểm tra xóa sản phẩm có ID không tồn tại	PHPUnit	Phùng Tân Nhứt	Build 1.0	
24	UT-024	Kiểm tra xóa sản phẩm có ID hợp lệ	PHPUnit	Phùng Tân Nhứt	Build 1.0	
		Function addToCart()	PHPUnit	Huỳnh Anh Tuấn	Build 1.0	
25	UT-025	Kiểm tra thêm sản phẩm khi sản phẩm không tồn tại	PHPUnit	Huỳnh Anh Tuấn	Build 1.0	
26	UT-026	Kiểm tra thêm sản phẩm khi sản phẩm còn đủ số lượng trong kho và tổng tiền	PHPUnit	Huỳnh Anh Tuấn	Build 1.0	
27	UT-027	Kiểm tra thêm sản phẩm khi số lượng sản phẩm vượt quá số lượng trong kho và tổng tiền	PHPUnit	Huỳnh Anh Tuấn	Build 1.0	
28	UT-028	Kiểm tra thêm sản phẩm khi sản phẩm có sẵn trong giỏ hàng và còn đủ số lượng trong kho và tổng tiền	PHPUnit	Huỳnh Anh Tuấn	Build 1.0	
29	UT-029	Kiểm tra thêm sản phẩm khi sản phẩm có sẵn trong giỏ hàng và vượt quá số lượng trong kho và tổng tiền	PHPUnit	Huỳnh Anh Tuấn	Build 1.0	
		Function updateCartQuantity()	PHPUnit	Huỳnh Anh Tuấn	Build 1.0	
30	UT-030	Kiểm tra cập nhật sản phẩm khi sản phẩm không tồn tại	PHPUnit	Huỳnh Anh Tuấn	Build 1.0	
31	UT-031	Kiểm tra cập nhật số lượng sản phẩm khi sản phẩm còn đủ số lượng trong kho và tổng tiền	PHPUnit	Huỳnh Anh Tuấn	Build 1.0	
32	UT-032	Kiểm tra cập nhật số lượng sản phẩm vượt quá số lượng trong kho và tổng tiền	PHPUnit	Huỳnh Anh Tuấn	Build 1.0	
		Function removeFromCart()	PHPUnit	Huỳnh Anh Tuấn	Build 1.0	
33	UT-033	Kiểm tra xóa sản phẩm với ID không tồn tại	PHPUnit	Huỳnh Anh Tuấn	Build 1.0	
34	UT-034	Kiểm tra xóa sản phẩm với ID hợp lệ	PHPUnit	Huỳnh Anh Tuấn	Build 1.0	
		Function place_order()	PHPUnit	Phùng Tân Nhứt	Build 1.0	
35	UT-035	Thanh toán khi chưa đăng nhập	PHPUnit	Phùng Tân Nhứt	Build 1.0	
36	UT-036	Kiểm tra khi thông tin đơn hàng không hợp lệ	PHPUnit	Phùng Tân Nhứt	Build 1.0	
37	UT-037	Kiểm tra khi tổng tiền đơn hàng không hợp lệ	PHPUnit	Phùng Tân Nhứt	Build 1.0	
38	UT-038	Kiểm tra thanh toán khi ID giỏ hàng không tồn tại	PHPUnit	Phùng Tân Nhứt	Build 1.0	
39	UT-039	Kiểm tra khi dữ liệu thanh toán đầy đủ và hợp lệ	PHPUnit	Phùng Tân Nhứt	Build 1.0	

Figure 23. TestPlan Test Plan Table

Function addProduct() :

- Check if the cases when adding new products are valid.
 - Check if the input price is negative (Test when the input price is negative).
 - Test when the selling price is a negative value.
 - Check if the input quantity is negative (Test when the input quantity is negative).
 - Test when sales quantity is negative.
 - Test when sales quantity exceeds input quantity.
 - Check when product name is duplicated (Test when adding product with name duplicated with another product).
 - Check if input data is complete and valid.

Function updateProduct() :

- Check if product update cases are valid or not
 - Update product with non-existing ID (Test when product ID does not exist in database).
 - Update product with duplicate name (Test when updating product with existing name).
 - Test when the input quantity is negative (Test when updating the input quantity is negative).
 - Test when sales quantity is negative (Test when sales quantity update is negative).
 - Update valid sales quantity (Test when sales quantity is valid).
 - Update product name , image, import quantity, sale quantity, import price, sale price and description (Test when product name , image, import quantity, sale quantity, import price, sale price and description are valid).

Function deleteProduct() (Delete product):

- Check the cases when deleting products:
 - Delete product with non-existing ID (Test when deleting product with ID not in database).
 - Delete product with valid ID (Test when deleting product with valid ID).

Function addToCart() (Add product to cart):

- Check the cases of adding products to cart:
 - Add product when product does not exist (Test when adding product when product does not exist in system).
 - Add product when stock is sufficient and total price (Test when adding product to cart with stock is sufficient and total price).

- Add product when quantity exceeds stock quantity and total price (Test when adding product when quantity in cart exceeds stock quantity and total price).
- Add product when product is available in cart with sufficient quantity and total price (Test when product is already in cart and still has sufficient quantity in stock and total price).
- Add products when available products exceed stock quantity and total price (Test when the product is already in the cart exceed the quantity in stock and still have enough quantity in stock and total amount).

Function updateCartQuantity() (Update the quantity of products in the cart):

- Check the cases of updating the cart product quantity:
 - Update product quantity when product does not exist (Test when updating product quantity when product does not exist in shopping cart).
 - Update product quantity when there is enough stock and total price (Test when updating product quantity in cart when there is enough stock and total price).
 - Update product quantity when quantity exceeds stock quantity (Test when updating product where quantity in cart exceeds available quantity and total price).

Function deleteFromCart() (Delete product from cart):

- Check the cases of removing products from the cart:
 - Delete product with non-existent ID (Test when deleting product with ID not in cart).
 - Delete product with existing ID (Test when deleting product with ID in cart).

Function place_order () (Payment):

- Checkout in cart:

- Checkout while not logged in (Test when user is not logged in and tries to pay).
- Checkout when cart ID does not exist (Test when cart ID does not exist in the system).
- Check payment when order information is invalid (Test when order information is invalid when paying).
- Check payment when total order amount is invalid (Test when total order amount is invalid when paying).
- Pay when payment data is complete and valid (Test when payment data is complete and valid).

1.2. Import process

TestCase ID	Điều kiện cần để kiểm tra	Unit Under Test	Test Description	TestData	Expected Result	Passed	Failed	Result
TC_AddProduct_10	Yêu cầu đăng nhập	UT-001	Kiểm tra khi giá nhập là số âm	buy_price = -1 sell_price = -1	Hiển thị thông báo: "Giá sản phẩm không được là số âm"	Passed	0	
TC_AddProduct_11	Yêu cầu đăng nhập	UT-002	Kiểm tra khi số bán là số âm	buy_price = -1 sell_price = -1	Hiển thị thông báo: "Giá sản phẩm không được là số âm"	Passed	0	
TC_AddProduct_12	Yêu cầu đăng nhập	UT-003	Kiểm tra khi id hàng nhập không tồn tại	buy_price = -1 sell_price = -1	Hiển thị thông báo: "Không tìm thấy sản phẩm có id hàng nhập là số âm"	Passed	0	
TC_AddProduct_13	Yêu cầu đăng nhập	UT-004	Kiểm tra khi số lượng nhập là số âm	buy_price = -1 sell_price = -1 available_quantity = -1	Hiển thị thông báo: "Số lượng sản phẩm không là số âm"	Passed	0	
TC_AddProduct_14	Yêu cầu đăng nhập	UT-005	Kiểm tra khi số lượng bán lớn hơn số lượng nhập	buy_price = -1 sell_price = -1 available_quantity = 10, quantity = 5	Hiển thị thông báo: "Số lượng bán không được lớn hơn số lượng nhập"	Passed	0	
TC_AddProduct_15	Yêu cầu đăng nhập	UT-006	Kiểm tra khi số id hình ảnh không tồn tại	buy_price = -1 sell_price = -1 available_quantity = 10, quantity = 5 name = "DOLCE"	Hiển thị thông báo: "Tên sản phẩm đã tồn tại"	Passed	0	
TC_AddProduct_16	Yêu cầu đăng nhập	UT-007	Kiểm tra khi số id tên duy nhất và hợp lệ	buy_price = -1 sell_price = -1 available_quantity = 10, quantity = 5 name = "ABC", image = "1aba114354_product-big-4.jpg", buy_price = "100", sell_price = "120", id_categories = "Cakes", id_brands = "Samsung"	Hiển thị thông báo: "Thêm sản phẩm thành công"	Passed	0	
Function updateProduct()								
TC_UpdateProduct_01	Yêu cầu đăng nhập	UT-008	Kiểm tra cập nhật sản phẩm với ID không tồn tại	ID = 10	Hiển thị thông báo: "Sản phẩm không tồn tại"	Passed	0	
TC_UpdateProduct_09	Yêu cầu đăng nhập	UT-009	Kiểm tra cập nhật sản phẩm với tên bị trùng	ID = 1, name = "DOLCE"	Hiển thị thông báo: "Tên sản phẩm đã tồn tại"	Passed	0	
TC_UpdateProduct_10	Yêu cầu đăng nhập	UT-010	Kiểm tra cập nhật sản phẩm với id không tồn tại	ID = 1, name = "ABC"	Hiển thị thông báo: "Cập nhật sản phẩm thành công"	Passed	0	
TC_UpdateProduct_11	Yêu cầu đăng nhập	UT-011	Kiểm tra cập nhật sản phẩm với hình ảnh hợp lệ	ID = 1, name = "1aba114354_product-big-4.jpg"	Hiển thị thông báo: "Cập nhật sản phẩm thành công"	Passed	0	
TC_UpdateProduct_12	Yêu cầu đăng nhập	UT-012	Kiểm tra khi số lượng nhập là số âm	ID = 1, quantity = -1	Hiển thị thông báo: "Số lượng sản phẩm không được là số âm"	Passed	0	
TC_UpdateProduct_13	Yêu cầu đăng nhập	UT-013	Kiểm tra khi số id hình ảnh không tồn tại	ID = 1, quantity = -1	Hiển thị thông báo: "Không tìm thấy sản phẩm có id hình ảnh là số âm"	Passed	0	
TC_UpdateProduct_14	Yêu cầu đăng nhập	UT-014	Kiểm tra cập nhật số lượng bán hợp lệ	ID = 1, quantity = 10, available_quantity = 10	Hiển thị thông báo: "Cập nhật sản phẩm thành công"	Passed	0	
TC_UpdateProduct_15	Yêu cầu đăng nhập	UT-015	Kiểm tra khi số lượng bán lớn hơn số lượng nhập	ID = 1, quantity = 10, available_quantity = 20	Hiển thị thông báo: "Số lượng bán không được lớn hơn số lượng nhập"	Passed	0	
TC_UpdateProduct_16	Yêu cầu đăng nhập	UT-016	Kiểm tra khi số id tên không tồn tại	ID = 1, buy_price = -100	Hiển thị thông báo: "Giá sản phẩm không được là số âm"	Passed	0	
TC_UpdateProduct_17	Yêu cầu đăng nhập	UT-017	Kiểm tra khi số id tên không hợp lệ	ID = 1, buy_price = -100	Hiển thị thông báo: "Giá sản phẩm không hợp lệ"	Passed	0	
TC_UpdateProduct_18	Yêu cầu đăng nhập	UT-018	Kiểm tra khi số bán là số âm	ID = 1, sell_price = -100	Hiển thị thông báo: "Giá bán không được là số âm"	Passed	0	
TC_UpdateProduct_19	Yêu cầu đăng nhập	UT-019	Kiểm tra khi số id hình ảnh không hợp lệ	ID = 1, sell_price = -100	Hiển thị thông báo: "Cập nhật sản phẩm thành công"	Passed	0	
TC_UpdateProduct_20	Yêu cầu đăng nhập	UT-020	Kiểm tra cập nhật sản phẩm với id không hợp lệ	ID = 1, sell_price = -100	Hiển thị thông báo: "Không tìm thấy sản phẩm có id không hợp lệ"	Passed	0	
TC_UpdateProduct_21	Yêu cầu đăng nhập	UT-021	Kiểm tra cập nhật sản phẩm với loại bánh hợp lệ	ID = 1, id_categories = "Cakes"	Hiển thị thông báo: "Cập nhật sản phẩm thành công"	Passed	0	
TC_UpdateProduct_22	Yêu cầu đăng nhập	UT-022	Kiểm tra cập nhật sản phẩm với thương hiệu hợp lệ	ID = 1, id_brands = "Samsung"	Hiển thị thông báo: "Cập nhật sản phẩm thành công"	Passed	0	
Function deleteProduct()								
TC_DeleteProduct_01	Yêu cầu đăng nhập	UT-023	Kiểm tra xóa sản phẩm có ID không tồn tại	ID = 10	Hiển thị thông báo: "Không có ID sản phẩm đã tồn tại"	Passed	0	
TC_DeleteProduct_02	Yêu cầu đăng nhập	UT-024	Kiểm tra xóa sản phẩm có ID hợp lệ	ID = 10	Hiển thị thông báo: "Xóa sản phẩm thành công"	Passed	0	

Figure 24. Unit Test Importing goods

Passed 24 / Failed 0 :

- All tests in this table passed, none failed.

Function addProduct() :

TC_AddProduct_10: Test when the input price is negative (buy_price = -1)
– Test passed with error message: "Product price cannot be negative".

TC_AddProduct_11: Test when the selling price is negative (sell_price = -1)
– Test passed with error message: "Product price cannot be negative".

TC_AddProduct_12: Test when quantity entered is negative (quantity = -1)
– Test passed with error message: "Quantity cannot be negative".

TC_AddProduct_13: Test when quantity sold is negative (available_quantity = -1) – Test passed with error message: "Quantity cannot be negative".

TC_AddProduct_14: Test when sales quantity is greater than import quantity (available_quantity = 10, quantity = 5) – Test passed with error message: "Sales quantity cannot be greater than import quantity".

TC_AddProduct_15: Test when adding product with duplicate name (name = "DULCE") – Test passed with error message: "Product name already exists".

TC_AddProduct_16: Check when data is complete and valid – Test passed with message: "Added product successfully".

Function updateProduct() :

TC_UpdateProduct_01: Test to update product with non-existent ID (ID = 10) – Test passed with error message: "Product does not exist".

TC_UpdateProduct_09: Test to update product with duplicate name (name = "DULCE") – Test passed with error message: "Product name already exists".

TC_UpdateProduct_10: Test to update product with valid name (name = "ABC") – Test passed with message: "Product update successful".

TC_UpdateProduct_11: Test product update with valid image (image = "1aba114354.product-big-4.jpg") – Test passed with message: "Product update successful".

TC_UpdateProduct_12: Test when quantity entered is negative (quantity = -1) – Test passed with error message: "Quantity of product cannot be negative".

TC_UpdateProduct_13: Test when quantity sold is negative (available_quantity = -1) – Test passed with error message: "Product quantity cannot be negative".

TC_UpdateProduct_14: Check for valid sales quantity update (quantity = 10, available_quantity = 10) – Test passed with message: "Product update successful".

TC_UpdateProduct_15: Test when sales quantity is greater than input quantity (quantity = 10, available_quantity = 20) – Test passed with error message: "Sale quantity cannot be greater than input quantity".

TC_UpdateProduct_16: Test when the input price is negative (buy_price = -1) – Test passed with error message: "Product price cannot be negative".

TC_UpdateProduct_17: Test product update with valid input price (buy_price = 100) – Test passed with message: "Product update successful".

TC_UpdateProduct_18: Test when the selling price is negative (sell_price = -1) – Test passed with error message: "Product price cannot be negative".

TC_UpdateProduct_19: Test when updating product with valid selling price (sell_price = 100) – Test passed with message: "Product updated successfully".

TC_UpdateProduct_20: Test product update with valid description (description = "Banh ngon") – Test passed with message: "Product update successful".

TC_UpdateProduct_21: Test to update product with valid cake type (id_cagories = "Cakes") – Test passed with message: "Product updated successfully".

TC_UpdateProduct_22: Test for product update with valid brand (id_brands = "SamSung") – Test passed with message: "Product update successful".

Function deleteProduct() :

TC_DeleteProduct_01: Test to delete a product with a non-existent ID (ID = 10) – Test passed with error message: "No product ID to delete".

TC_DeleteProduct_02: Test to delete a product with a valid ID (ID = 10) – Test passed with message: "Product deleted successfully".

1.3. Purchasing process

Test Case ID	Điều kiện cần để kiểm tra	Unit Under Test	Test Description	TestData	Expected Result	Passed	Failed	Result
			Function addToCart()			15	0	
TC_AddCart_01		UT-015	Kiểm tra thêm sản phẩm khi sản phẩm không tồn tại	ID = 10	Hiển thị thông báo: "Sản phẩm không tồn tại"	Passed		
TC_AddCart_02		UT-016	Kiểm tra thêm sản phẩm khi sản phẩm có đủ số lượng trong kho và tồn tại	ID = 1, quantity = 2, sell price = 120	Hiển thị thông báo: "Thêm sản phẩm... vào giỏ hàng thành công", tổng tiền là 240USD	Passed		
TC_AddCart_03		UT-017	Kiểm tra thêm sản phẩm khi sản phẩm có tồn tại nhưng quá số lượng trong kho và tồn tại	ID = 1, quantity = 100, sell price = 120	Hiển thị thông báo: "Thêm sản phẩm... vào giỏ hàng thành công", Chi phí là: 12000đ, Chi phí là: 12000đ, tổng tiền là 12000USD	Passed		
TC_AddCart_04	Yêu cầu có sản phẩm trong giỏ trước	UT-018	Kiểm tra thêm sản phẩm khi sản phẩm có tồn trong giỏ hàng và còn số lượng trong kho và tồn tại	ID = 1, quantity = 2, sell price = 120 (cart has quantity = 1)	Hiển thị thông báo: "Thêm sản phẩm... vào giỏ hàng thành công", tổng tiền là 360USD	Passed		
TC_AddCart_05	Yêu cầu có sản phẩm trong giỏ trước	UT-019	Kiểm tra thêm sản phẩm khi sản phẩm có sẵn trong giỏ hàng và vượt quá số lượng trong kho và tồn tại	ID = 1, quantity = 120 (cart has quantity = 1)	Hiển thị thông báo: "Sản phẩm... không đủ số lượng. Chỉ còn lại: sản phẩm", tổng tiền 120USD	Passed		
			Function updateCartQuantity()					
TC_UpdateCart_01		UT-030	Kiểm tra cập nhật số lượng sản phẩm khi sản phẩm không tồn tại	ID = 10	Hiển thị thông báo: "Sản phẩm không tồn tại"	Passed		
TC_UpdateCart_02		UT-031	Kiểm tra cập nhật số lượng sản phẩm khi sản phẩm còn đủ số lượng trong kho và tồn tại	ID = 1, quantity = 5, sell price = 120	Hiển thị thông báo: "Cập nhật số lượng sản phẩm... thành công", tổng tiền 1100USD	Passed		
TC_UpdateCart_03		UT-032	Kiểm tra cập nhật số lượng sản phẩm vượt quá số lượng trong kho và tồn tại	ID = 1, quantity = 100, sell price = 120	Hiển thị thông báo: "Sản phẩm... không đủ số lượng. Chỉ còn lại: sản phẩm", tổng tiền 120USD	Passed		
			Function deleteCart()					
TC_DeleteCart_01		UT-033	Kiểm tra xóa sản phẩm với ID không tồn tại	ID = 10	Hiển thị thông báo: "Sản phẩm không tồn tại"	Passed		
TC_DeleteCart_02		UT-034	Kiểm tra xóa sản phẩm với ID hợp lệ	ID = 1	Hiển thị thông báo: "Xóa sản phẩm thành công"	Passed		
			Function placeOrder()					
TC_Checkout_01		UT-035	Thực hiện đơn hàng thành công		Hiển thị thông báo: "Vui lòng chờ nhận đồ mua hàng"	Passed		
TC_Checkout_02	Yêu cầu đăng nhập	UT-036	Kiểm tra khi thay đổi đơn hàng không hợp lệ	ID = 1	Không hiển thị thông báo đơn hàng	Passed		
TC_Checkout_03	Yêu cầu đăng nhập	UT-037	Kiểm tra khi thay đổi đơn hàng không hợp lệ	sell price = 120	Không hiển thị thông báo đơn hàng	Passed		
TC_Checkout_11	Yêu cầu đăng nhập	UT-038	Kiểm tra thành toán khi ID mua hàng không tồn tại	ID = 10	Hiển thị thông báo: "Đặt hàng không thành công"	Passed		
TC_Checkout_12	Yêu cầu đăng nhập	UT-039	Kiểm tra khi đã hủy thanh toán trước đó và hủy lại	image = "shut", email = "tienbut2111@gmail.com", address = "HCM", number phone = "0901641800", note = ...	Hiển thị thông báo: "Đặt hàng không thành công"	Passed		

Figure 25. Purchase Unit Test

Passed 15 / Failed 0 :

- All tests in this table passed, none failed.

Function addToCart() (Add product to cart):

- **TC_AddCart_01** : Test when adding a non-existent product to the cart. The test passed with the error message : "Product does not exist".
- **TC_AddCart_02** : Test when adding product to cart and product has enough quantity in stock. Test passed with message: "Added product to cart successfully".
- **TC_AddCart_03** : Test when the quantity of products to be added to the cart exceeds the quantity in stock. The test passed with the error message : "Product quantity is insufficient. Only ... products left".
- **TC_AddCart_04** : Test when the product is already in the cart and still has enough quantity in stock. The test passed with the message: "Product added to cart successfully".
- **TC_AddCart_05** : Test when the product is already in the cart but the quantity to be added exceeds the quantity in stock. The test passed with the error message : "Product quantity is insufficient. Only ... products left".
- **Function updateCartQuantity() (Update the quantity of products in the cart):**
- **TC_UpdateCart_01** : Test when updating quantity of product in cart with non-existent ID. Test passed with error message : "Product does not exist".

- **TC_UpdateCart_02** : Check when updating product quantity and product has enough quantity in stock. The test passed with the message: "Product quantity updated successfully!".
- **TC_UpdateCart_03** : Test when the quantity of products to be updated exceeds the quantity in stock . The test passed with the error message : "Product quantity is insufficient. Only ... products left".

Function deleteFormCart() (Delete product from cart):

- **TC_DeleteCart_01** : Test when deleting a product with a non-existent ID. The test passed with the error message : "Product does not exist".
- **TC_DeleteCart_02** : Test when deleting a product with a valid ID. The test passed with the message: "Product deleted successfully".

Function place_order () (Payment):

- **TC_Checkout_01** : Test when not logged in and try to checkout. Test passed with message: "Please login to purchase".
- **TC_Checkout_09** : Checkout when cart ID does not exist. Test passed with error message : "Order failed".
- **TC_Checkout_10** : Checkout when the required information (such as name, email, address, phone number) is complete and valid. The test passed with the message: "Order successful!".
- **TC_Checkout_11** : Check when order information is invalid. Test passed and order information is not displayed .
- **TC_Checkout_12** : Check when total amount is invalid. Test passed and total amount is not displayed .

2. Integration Testing

Integration Testing is a phase in the software testing process that focuses on testing the interactions between modules or components in a system. The goal is to ensure that the integrated modules function as expected when they work together.

2.1. Test Design

Test design is the process of creating scenarios, test cases, and test methods to test a software or system to ensure that the system operates correctly and meets technical and functional requirements. The goal of test design is to detect bugs and ensure software quality.

Purpose of Test Design:

- **Software quality assurance** : Test design helps identify the parts of the software that need to be tested to ensure they function as required.
- **Error and Problem Detection** : Test design aims to detect errors or problems in the software before it is released.
- **Risk Reduction** : Adequate testing helps to reduce quality risks when the software is deployed.
- **Ensuring user requirements are met** : Test design helps verify that the software works as per the requirements, thereby improving end user satisfaction.
- **Improve software performance** : Performance testing helps to check the system's ability to handle load and handling under extreme conditions.

Import process

Yêu cầu mức 1	Yêu cầu mức 2	Yêu cầu mức 3	Tiêu chí kiểm tra	Loại kiểm tra	TestCase ID	Ghi chú
Thêm sản phẩm	Thành công	Dữ liệu đầy đủ và hợp lệ	Kiểm tra thêm với dữ liệu đầy đủ và hợp lệ	Function addProduct()	TC AddProduct 016	
	Không thành công	Bó trống tên sản phẩm	Kiểm tra thêm sản phẩm khi để trống trường tên sản phẩm	Function addProduct()	TC AddProduct 01	
		Bó trống hình ảnh sản phẩm	Kiểm tra thêm sản phẩm khi không chọn file hình ảnh cho sản phẩm	Function addProduct()	TC AddProduct 02	
		Bó trống số lượng nhập	Kiểm tra thêm sản phẩm khi để trống trường số lượng nhập sản phẩm	Function addProduct()	TC AddProduct 03	
		Bó trống số lượng bán	Kiểm tra thêm sản phẩm khi để trống trường số lượng bán sản phẩm	Function addProduct()	TC AddProduct 04	
		Bó trống mô tả sản phẩm	Kiểm tra thêm sản phẩm khi để trống trường mô tả sản phẩm	Function addProduct()	TC AddProduct 05	
		Bó trống giá nhập	Kiểm tra thêm sản phẩm khi để trống trường giá nhập sản phẩm	Function addProduct()	TC AddProduct 06	
		Bó trống giá bán	Kiểm tra thêm sản phẩm khi để trống trường giá bán sản phẩm	Function addProduct()	TC AddProduct 07	
		Không chọn loại bánh	Kiểm tra thêm sản phẩm khi không chọn option loại bánh	Function addProduct()	TC AddProduct 08	
		Không chọn thương hiệu	Kiểm tra thêm sản phẩm khi không chọn option thương hiệu	Function addProduct()	TC AddProduct 09	
		Giá nhập là số âm	Kiểm tra thêm sản phẩm khi trường giá nhập là số âm	Function addProduct()	TC AddProduct 10	
		Giá bán là số âm	Kiểm tra thêm sản phẩm khi trường giá bán là số âm	Function addProduct()	TC AddProduct 11	
		Số lượng nhập là số âm	Kiểm tra thêm sản phẩm khi trường số lượng nhập là số âm	Function addProduct()	TC AddProduct 12	
		Số lượng bán là số âm	Kiểm tra thêm sản phẩm khi trường số lượng bán là số âm	Function addProduct()	TC AddProduct 13	
		Số lượng bán lớn hơn số lượng nhập	Kiểm tra thêm sản phẩm khi trường số lượng bán lớn hơn trường số lượng nhập	Function addProduct()	TC AddProduct 14	
		Thêm sản phẩm với tên bị trùng	Kiểm tra thêm sản phẩm khi tên sản phẩm đã tồn tại	Function addProduct()	TC AddProduct 15	
Cập nhật sản phẩm	Thành công	Tên sản phẩm hợp lệ	Kiểm tra cập nhật sản phẩm khi tên sản phẩm hợp lệ	Function updateProduct()	TC UpdateProduct 10	
		Hình ảnh sản phẩm hợp lệ	Kiểm tra cập nhật sản phẩm khi hình ảnh sản phẩm hợp lệ	Function updateProduct()	TC UpdateProduct 11	
		Số lượng bán hợp lệ	Kiểm tra cập nhật sản phẩm khi số lượng bán sản phẩm hợp lệ	Function updateProduct()	TC UpdateProduct 14	
		Giá nhập hợp lệ	Kiểm tra cập nhật sản phẩm khi giá nhập sản phẩm hợp lệ	Function updateProduct()	TC UpdateProduct 17	
		Giá bán hợp lệ	Kiểm tra cập nhật sản phẩm khi giá bán sản phẩm hợp lệ	Function updateProduct()	TC UpdateProduct 19	
		Mô tả hợp lệ	Kiểm tra cập nhật sản phẩm khi mô tả sản phẩm hợp lệ	Function updateProduct()	TC UpdateProduct 20	
		Loại bánh hợp lệ	Kiểm tra cập nhật sản phẩm khi loại bánh sản phẩm hợp lệ	Function updateProduct()	TC UpdateProduct 21	
		Thương hiệu hợp lệ	Kiểm tra cập nhật sản phẩm khi thương hiệu sản phẩm hợp lệ	Function updateProduct()	TC UpdateProduct 22	
	Không thành công	ID sản phẩm không tồn tại	Kiểm tra cập nhật sản phẩm khi ID sản phẩm không xuất hiện trên URL	Function updateProduct()	TC UpdateProduct 01	
		Bó trống tên sản phẩm	Kiểm tra cập nhật sản phẩm khi để trống trường tên sản phẩm	Function updateProduct()	TC UpdateProduct 02	
		Bó trống hình ảnh sản phẩm	Kiểm tra cập nhật sản phẩm khi không chọn file hình ảnh cho sản phẩm	Function updateProduct()	TC UpdateProduct 03	
		Bó trống số lượng nhập	Kiểm tra cập nhật sản phẩm khi để trống trường số lượng nhập sản phẩm	Function updateProduct()	TC UpdateProduct 04	
		Bó trống số lượng bán	Kiểm tra cập nhật sản phẩm khi để trống trường số lượng bán sản phẩm	Function updateProduct()	TC UpdateProduct 05	
		Bó trống giá nhập	Kiểm tra cập nhật sản phẩm khi để trống trường giá nhập sản phẩm	Function updateProduct()	TC UpdateProduct 06	
		Bó trống giá bán	Kiểm tra cập nhật sản phẩm khi để trống trường giá bán sản phẩm	Function updateProduct()	TC UpdateProduct 07	
		Bó trống mô tả sản phẩm	Kiểm tra cập nhật sản phẩm khi để trống trường mô tả sản phẩm	Function updateProduct()	TC UpdateProduct 08	
		Cập nhật sản phẩm với tên bị trùng	Kiểm tra cập nhật sản phẩm khi tên sản phẩm đã tồn tại	Function updateProduct()	TC UpdateProduct 09	
		Số lượng nhập là số âm	Kiểm tra cập nhật sản phẩm khi số lượng nhập là số âm	Function updateProduct()	TC UpdateProduct 12	
		Số lượng bán là số âm	Kiểm tra cập nhật sản phẩm khi số lượng bán là số âm	Function updateProduct()	TC UpdateProduct 13	
		Số lượng bán lớn hơn số lượng nhập	Kiểm tra cập nhật sản phẩm khi số lượng bán lớn hơn số lượng nhập	Function updateProduct()	TC UpdateProduct 15	
		Giá nhập là số âm	Kiểm tra cập nhật sản phẩm khi giá nhập là số âm	Function updateProduct()	TC UpdateProduct 16	
		Giá bán là số âm	Kiểm tra cập nhật sản phẩm khi giá bán là số âm	Function updateProduct()	TC UpdateProduct 18	
Xóa sản phẩm	Thành công	ID sản phẩm hợp lệ	Kiểm tra xóa sản phẩm với ID hợp lệ	Function deleteProduct()	TC DeleteProduct 02	
	Không thành công	Sản phẩm không tồn tại	Kiểm tra xóa sản phẩm khi ID sản phẩm không xuất hiện trên URL	Function deleteProduct()	TC DeleteProduct 01	

Figure 26. Test Design Import

Add product

Success : Test the add product functionality with valid data, including name, image, quantity, price, description, cake type, and brand.

Failed : Check for invalid cases like:

- Leave important fields blank (product name, image, quantity, price, description, cake type, brand).
- Negative values for quantity or price.
- Sales quantity is greater than import quantity.
- Duplicate product name.

Product Updates

Success : Test the product update functionality with valid fields like name, image, quantity, import price, sale price, description, cake type, and brand.

Failed : Check for invalid cases when updating:

- Product ID does not exist.
- Important fields are left blank (name, image, quantity, price, description).

- Negative values for quantity and price.
- Sales quantity is greater than import quantity.
- Duplicate product name.

Delete product

Success : Check the product deletion with valid ID.

Failed : Test the case of deleting a product when the product ID does not exist.

Purchasing process

Yêu cầu mức 1	Yêu cầu mức 2	Yêu cầu mức 3	Tiêu chí kiểm tra	Loại kiểm tra	TestCase ID	Ghi chú
Thêm Sản Phẩm vào Giỏ hàng	Thành công	Khai sản phẩm còn đủ số lượng trong kho	Kiểm tra thêm sản phẩm khi còn đủ số lượng	Function addToCart()	TC_AddCart_02	
		Khai sản phẩm có sẵn trong giỏ hàng và còn đủ số lượng trong kho	Kiểm tra thêm sản phẩm có sẵn trong giỏ hàng khi có đủ số lượng	Function addToCart()	TC_AddCart_04	
	Không thành công	Khai sản phẩm không tồn tại	Kiểm tra thêm sản phẩm có sẵn trong giỏ hàng khi không tồn tại	Function addToCart()	TC_AddCart_01	
		Khai sản phẩm vượt quá số lượng trong kho	Kiểm tra thêm sản phẩm vượt quá số lượng sản phẩm trong kho	Function addToCart()	TC_Checkout_03	
		Khai sản phẩm có sẵn trong giỏ hàng và vượt quá số lượng trong kho	Kiểm tra thêm sản phẩm có sẵn trong giỏ hàng vượt quá số lượng sản phẩm trong	Function addToCart()	TC_Checkout_05	
Cập Nhật Sản Phẩm	Thành công	Khai sản phẩm còn đủ số lượng trong kho	Kiểm tra cập nhật số lượng sản phẩm có sẵn trong	Function updateCartQuantity()	TC_UpdateCart_02	
	Không thành công	Khai sản phẩm không tồn tại	Kiểm tra cập nhật sản phẩm trong giỏ hàng khi sản phẩm không tồn tại	Function updateCartQuantity()	TC_UpdateCart_01	
Xóa Sản Phẩm	Thành công	ID hợp lệ	Kiểm tra xóa sản phẩm trong giỏ hàng không tồn tại ID	Function deleteFromCart()	TC_DeleteCart_02	
	Không tồn tại	ID không tồn tại	Kiểm tra xóa sản phẩm trong giỏ hàng xóa sản phẩm có ID hợp lệ	Function deleteFromCart()	TC_DeleteCart_01	
Thanh toán giỏ hàng	Thành công	Dữ liệu thanh toán đầy đủ và hợp lệ	Kiểm tra thanh toán giỏ hàng với dữ liệu đầy đủ và hợp lệ	Function place_order()	TC_Checkout_12	
	Không thành công	Thanh toán khi chưa đăng nhập	Kiểm tra phái đang nhập trước khi thanh toán	Function place_order()	TC_Checkout_01	
		Thanh toán khi thông tin đơn hàng không hợp lệ	Kiểm tra thanh toán khi thông tin đơn hàng không hợp lệ	Function place_order()	TC_Checkout_02	
		Bó trống tên và tên	Kiểm tra thanh toán khi trường tên và tên để trống	Function place_order()	TC_Checkout_03	
		Bó trống email	Kiểm tra thanh toán khi trường email để trống	Function place_order()	TC_Checkout_04	
		Bó trống địa chỉ	Kiểm tra thanh toán khi trường địa chỉ để trống	Function place_order()	TC_Checkout_05	
		Bó trống số điện thoại	Kiểm tra thanh toán khi trường số điện thoại để trống	Function place_order()	TC_Checkout_06	
		Họ và tên là số	Kiểm tra thanh toán khi trường họ và tên là số	Function place_order()	TC_Checkout_07	
		Email không đúng định dạng	Kiểm tra thanh toán khi trường email không đúng định dạng	Function place_order()	TC_Checkout_08	
		Số điện thoại ít hơn 10 và lớn hơn 10 số	Kiểm tra thanh toán khi số điện thoại ít hơn và lớn hơn 10 số	Function place_order()	TC_Checkout_09	
		Thanh toán khi ID giỏ hàng không tồn tại	Kiểm tra thanh toán khi giỏ hàng không tồn tại ID	Function place_order()	TC_Checkout_10	
				Function place_order()	TC_Checkout_11	

Figure 27. Test Design Purchase

Add product to cart

Success :

- Check for more products while supplies last
- Check more products available in cart when quantity is sufficient

Failed :

- Check for more products when product does not exist
- Check for additional products exceeding the quantity in stock
- Check more products available in cart exceeds the number of products in stock

Update product quantity in stock

Success : Check the updated quantity of products in the cart

Unsuccessful:

- Check update product in cart when product does not exist
- Check when updating quantity exceeds quantity in stock

Remove product from cart

Success : Check to delete products in cart that do not have an ID

Failed : Check to delete product in cart delete product with valid ID

Checkout cart

Success : Checkout cart with complete and valid data

Failed :

- Check must login before payment
- Leave important fields blank (full name , email, address , phone number).
- Check payment when first and last name field is number
- Check payment when email field is not in correct format
- Check payment when phone number is less than and greater than 10 digits
- Checkout when cart ID does not exist
- Check payment when order information is invalid
- Check payment when order total is invalid

2.2. White box testing

White-box Testing Checklist is a list of items to be checked to ensure the quality of the source code and internal logic of an application. This checklist is used by testers or developers to thoroughly check the technical aspects during white-box testing.

Dự án:	Quản lý trang web bán bánh		
Phiên bản:	1.0.		
Người đánh giá:	Phùng Tấn Nhựt, Huỳnh Anh Tuấn, Trần Gia Phú		
Tài liệu:	Báo cáo kiểm thử		
Ngày:	1-Dec-24		
STT	Tiêu chí	Có/Không/Không áp dụng	Ghi chú
A. TỔNG QUAN			
1	Mẫu kế hoạch kiểm thử có tuân thủ các tiêu chuẩn/hướng dẫn đã chỉ định không?	Có	
2	Ranh giới hoặc phạm vi của kiểm thử có thể xác định rõ không?	Có	
3	Tất cả các điều kiện tiên quyết và tài liệu liên quan có được xác định không?	Có	
4	Kế hoạch kiểm thử có thể được truy vết tới các yêu cầu không?	Có	
5	Kế hoạch kiểm thử có được kiểm soát phiên bản không?	Có	
B. NỘI DUNG			
1	Kế hoạch kiểm thử có đề cập rõ ràng đến các thông tin yêu cầu sau không?	Có	
2	Mục tiêu kiểm thử	Có	
3	Phạm vi kiểm thử	Có	
4	Chiến lược kiểm thử	Có	
5	Tài nguyên (chẳng hạn như vật liệu, công cụ và nhân lực)	Có	
6	Lịch trình kiểm thử	Có	
7	Môi trường kiểm thử	Có	
8	Các sản phẩm đầu ra của kiểm thử	Có	
9	Báo cáo kiểm thử	Có	
10	Kế hoạch có xác định rõ vai trò và trách nhiệm của các thành viên trong nhóm không?	Có	
11	Kế hoạch có bao gồm các đặc tả chi tiết, mô tả và thủ tục cho việc kiểm thử tất cả các hệ thống không?	Có	
12	Tài liệu có mô tả loại kiểm thử (test type) sẽ thực hiện không?	Có	
13	Đối với loại kiểm thử, tài liệu có liệt kê tất cả các kiểm thử cần áp dụng cho phần hệ thống đã chọn không?	Có	
14	Các kiểm thử tích hợp có được bao gồm không?	Có	
15	Phạm vi kiểm thử có đủ không?	Có	
16	Lịch trình có nằm trong khung thời gian đã thỏa thuận không?	Có	
17	Các phụ thuộc kiểm thử (nếu có) có được đề cập không?	Không	
18	Tiêu chí vào (entrance criteria) cho các hoạt động kiểm thử có được xác định không?	Có	
19	Tiêu chí ra (exit criteria), bao gồm ngưỡng cho sự sai lệch về hiệu suất, cho các hoạt động kiểm thử có được xác định không? Ví dụ: Tất cả các lỗi mức độ 1 và 2 phải được đóng và kế hoạch cho các lỗi mức độ 3 đã được phê duyệt trong Báo cáo Kiểm thử	Có	
20	Chiến lược kiểm thử hồi quy (regression testing) đã được định nghĩa khi có sự thay đổi hoặc cải tiến trong thiết kế hệ thống không?	Không	

Figure 28. White Box CheckList

White box testing (White Box Testing) is a software testing method that focuses on testing the internal logic and source code structure of an application to ensure that the functions are implemented as expected. This method is used to:

-Verify interface event handling: Ensure that events triggered from the interface (such as button presses or input) are correctly mapped to internal handler functions, and that processing steps (such as saving data, updating state) are performed logically.

-Test interface state transition flow: Ensure that interface state transition logic, based on user actions, is properly implemented, covering all possible states without creating error states.

Techniques used in the project:

- **Control Flow Testing** is a white box testing method that focuses on the logical execution flow of the application. It ensures that:

- All possible branches or paths in the program are tested.
- The conditions and decision points are verified.

- **Branch coverage testing:** Test decision branches like "Add successful" or "Add failed."

- **Path coverage testing:** Tests the various logical flows from start to finish, including error messages, validations, and database saves.

2.2.1. Import process

Add product

Control flow graph diagram

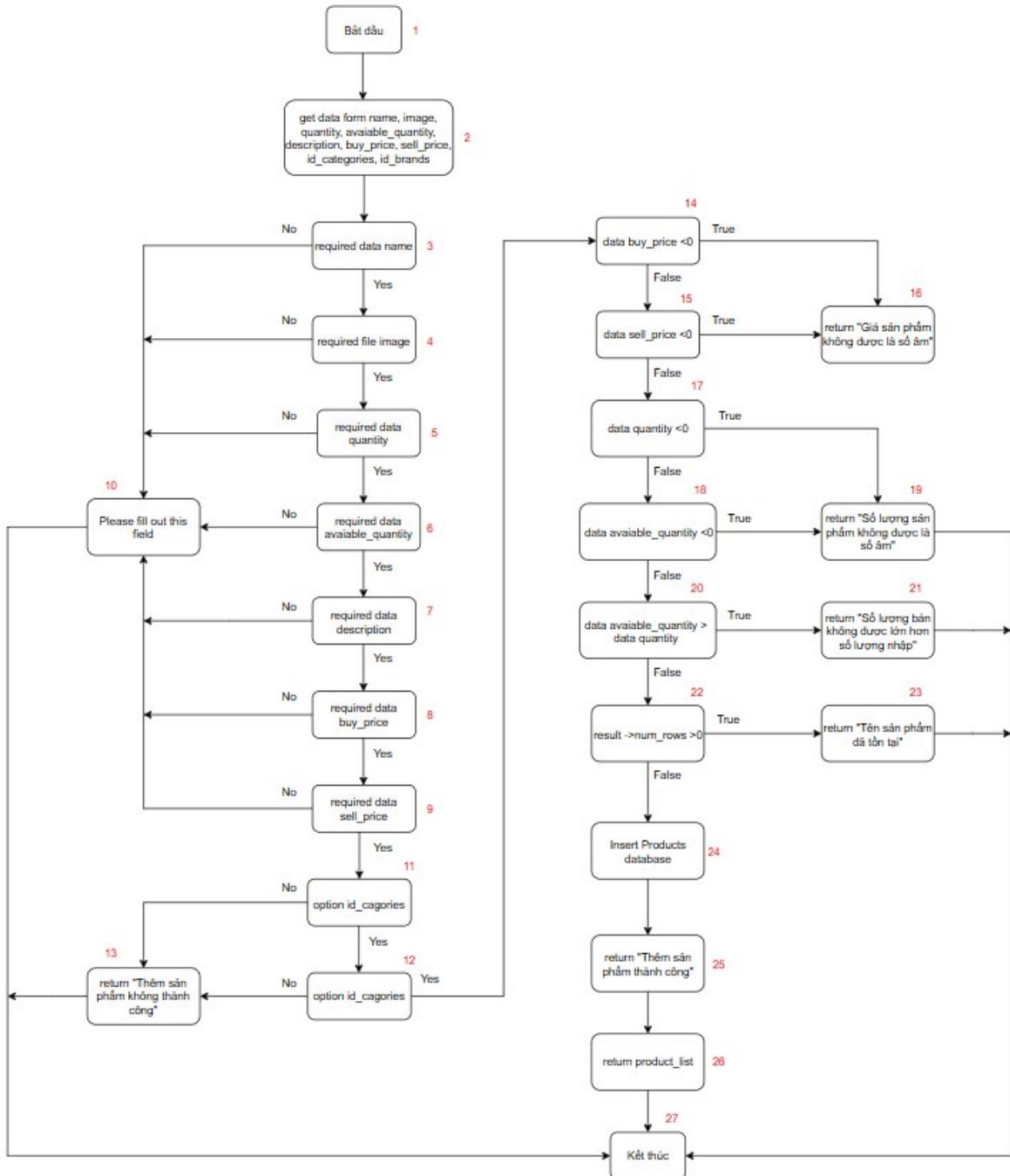


Figure 29. Control flow diagram Add product Import goods

Complexity

E = number of sides = 4 1

N = number of buttons = 2 7

P = number of connected components = 1

M = E - N + 2P = 4 1 - 2 7 + 2*1 = 16

Independent road

Independent line 1: 1-2 -3 -10-2 7

Independent line 2: 1-2-3 -4 -10-2 7

Independent line 3: 1-2-3-4 -5 -10-2 7

Independent line 4: 1-2-3-4-5 -6 -10-2 7

Independent line 5: 1-2-3-4-5-6 -7 -10-2 7

Independent line 6: 1-2-3-4-5-6-7 -8 -10-2 7

Independent line 7: 1-2-3-4-5-6-7-8 -9 -10-2 7

Independent line 8: 1-2-3-4-5-6-7-8-9-11-13-2 7

Independent line 9: 1-2-3-4-5-6-7-8-9 -11 -12-13-2 7

Independent line 10: 1-2-3-4-5-6-7-8-9- 11-12- 14-16-2 7

Independent line 11: 1-2-3-4-5-6-7-8-9-1 1-12-1 4-15-16-2 7

Independent line 12: 1-2-3-4-5-6-7-8-9- 11-12- 14-15-17-19-2 7

Independent line 13: 1-2-3-4-5-6-7-8-9- 11-12- 14-15-17-18-19-2 7

Independent line 14: 1-2-3-4-5-6-7-8-9- 11-12- 14-15-17-18-20-21-2 7

Independent line 15: 1-2-3-4-5-6-7-8-9- 11-12- 14-15-17-18-20-22-23-2 7

Independent line 16: 1-2-3-4-5-6-7-8-9 -11-12 -14-15-17-18-20-22-2 5 -2 6 -2 7

Test Suite

TC_AddProduct_01: 1-2 -3 -10-27

Input: name = “ ”

Expected result: Please fill out this field

TC_AddProduct_02: 1-2-3 -4 -10-27

Input: image = “ ”

Expected result: Please fill out this field

TC_AddProduct_03: 1-2-3-4 -5 -10-27

Input: quantity = “ ”

Expected result: Please fill out this field

TC_AddProduct_04: 1-2-3-4-5 -6 -10-27

Input : available_quantity = “ ”

Expected result: Please fill out this field

TC_AddProduct_05: 1-2-3-4-5-6 -7 -10-27

Input: description = “ ”

Expected result: Please fill out this field

TC_AddProduct_06: 1-2-3-4-5-6-7 -8 -10-27

Input: buy_price = “ ”

Expected result: Please fill out this field

TC_AddProduct_07: 1-2-3-4-5-6-7-8 -9 -10-27

Input: sell_price= “ ”

Expected result: Please fill out this field

TC_AddProduct_08 : 1-2-3-4-5-6-7-8-9-11-13-27

Input: id_cagories = “Cake Types”

Expected result: "Add product failed"

TC_AddProduct_09: 1-2-3-4-5-6-7-8-9- 11- 12-13-27

Input: id_brands = “Brand Type”

Expected result: "Add product failed"

TC_AddProduct_10: 1-2-3-4-5-6-7-8-9 -11-12 -14-16-27

Input: buy_price = -1

Expected result: "Product price cannot be negative"

TC_AddProduct_11: 1-2-3-4-5-6-7-8-9 -11-12 -14-15-16-27

Input: sell_price = - 1

Expected result: "Product price cannot be negative"

TC_AddProduct_12: 1-2-3-4-5-6-7-8-9 -11-12 -14-15-17-19-27

Input: quantity = -1

Expected result: "The number of products cannot be negative"

TC_AddProduct_13: 1-2-3-4-5-6-7-8-9 -11-12 -14-15-17-18-19-27

Input: available_quantity =-1

Expected result: "The number of products cannot be negative"

TC_AddProduct_14: 1-2-3-4-5-6-7-8-9 -11-12 -14-15-17-18-20-21-27

Input: available_quantity = 10, quantity = 5

Expected result: "The quantity sold must not be greater than the quantity imported"

TC_AddProduct_15: 1-2-3-4-5-6-7-8-9 -11-12 -14-15-17-18-20-22-23-27

Input: name = “ DULCE ”

Expected result: “Product name already exists”

TC_AddProduct_16: 1-2-3-4-5-6-7-8-9 -11-12 -14-15-17-18-20-22-25-26-27

Input: name = “ABC”, image = “1aba114354.product-big-4.jpg”, quantity = 10, available_quantity = 10, description = “ Delicious cakes”, buy_price = “100”, sell_price = “120”, id_cagories = “Cakes”, id_brands = “SamSung”

Expected result: “Added product successfully”

Product Updates

Control flow graph diagram

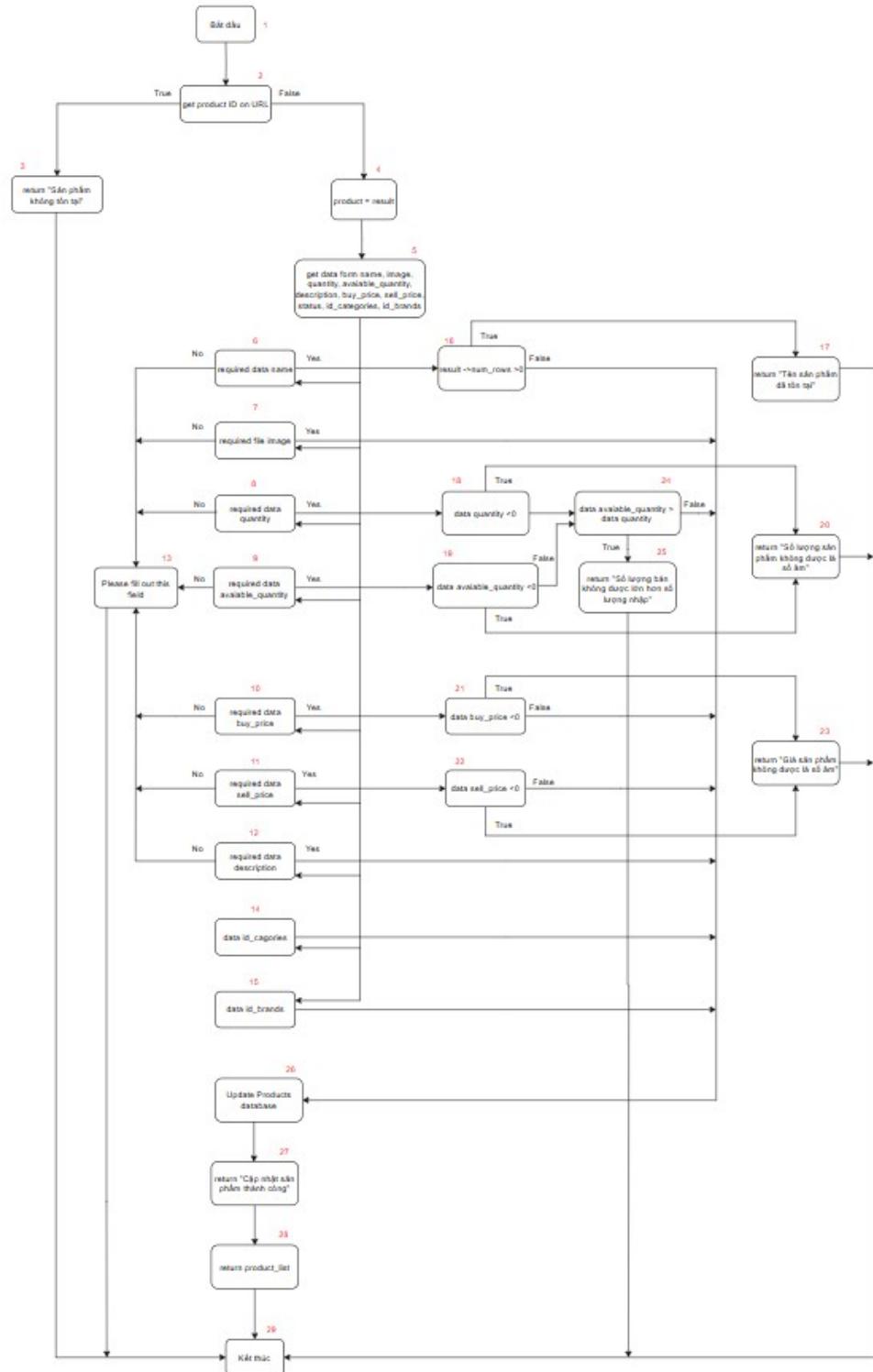


Figure 30. Product Update Control Flow Diagram

Complexity

E = number of sides = 49

N = number of buttons = 29

P = number of connected components = 1

$$M = E - N + 2P = 49 - 29 + 2*1 = 22$$

Independent road

Independent line 1: 1-2-3-29

Independent line 2: 1-2-4-5-6-13-29

Independent line 3: 1-2-4-5-7-13-29

Independent line 4: 1-2-4-5-8-13-29

Independent Line 5: 1-2-4-5-9-13-29

Independent line 6: 1-2-4-5-10-13-29

Independent line 7: 1-2-4-5-11-13-29

Independent line 8: 1-2-4-5-12-13-29

Independent line 9: 1-2-4-5-6-16-17-29

Independent Line 10: 1-2-4-5-6-16-27-28-29

Independent line 11: 1-2-4-5-7-27-28-29

Independent line 12: 1-2-3-4-5-8-18-20-29

Independent Line 13: 1-2-3-4-5-9-19-20-29

Independent Line 14: 1-2-3-4-5-9-19-24-27-28-29

Independent line 15: 1-2-3-4-5-9-19-24-25-29

Independent line 16: 1-2-3-4-5-10-21-23-29

Independent Line 17: 1-2-3-4-5-10-21-27-28-29

Independent Line 18: 1-2-3-4-5-11-22-23-29

Independent Line 19: 1-2-3-4-5-11-22-27-28-29

Independent line 20: 1-2-3-4-5-12-27-28-29

Independent Line 21: 1-2-3-4-5-14-27-28-29

Independent line 22: 1-2-3-4-5-15-27-28-29

Test Suite

TC_UpdateProduct_01: 1-2-3-29

Input: id = 10

Expected result: “Product does not exist”

TC_UpdateProduct_02: 1-2-4-5-6-13-29

Input: id = 1, name = “ ”

Expected result: “Please fill out this field”

TC_UpdateProduct_03: 1-2-4-5-7-13-29

Input: id = 1, image = “ ”

Expected result: “Please fill out this field”

TC_UpdateProduct_04: 1-2-4-5-8-13-29

Input: id = 1, quantity = “ ”

Expected result: “Please fill out this field”

TC_UpdateProduct_05: 1-2-4-5-9-13-29

Input: id = 1, available _quantity = “ ”

Expected result: “Please fill out this field”

TC_UpdateProduct_06: 1-2-4-5-10-13-29

Input: id = 1, buy _price = “ ”

Expected result: “Please fill out this field”

TC_UpdateProduct_07: 1-2-4-5-11-13-29

Input: id = 1, sell _price = “ ”

Expected result: “Please fill out this field”

TC_UpdateProduct_08: 1-2-4-5-12-13-29

Input: id = 1, description = “ ”

Expected result: “Please fill out this field”

TC_UpdateProduct_09: 1-2-4-5-6-16-17-29

Input: id = 1, name = “ DULCE ”

Expected result: “Product name already exists”

TC_UpdateProduct_10: 1-2-4-5-6-16-27-28-29

Input: id = 1, name = “ ABC ”

Expected result: “Product update successful”

TC_UpdateProduct_11: 1-2-4-5-7-27-28-29

Input : id = 1, image = “1aba114354.product-big-4.jpg”

Expected result: “Product update successful”

TC_UpdateProduct_12: 1-2-3-4-5-8-18-20-29

Input: id = 1, quantity = -1

Expected result: "The number of products cannot be negative"

TC_UpdateProduct_13: 1-2-3-4-5-9-19-20-29

Input: id = 1, available_quantity =-1

Expected result: "The number of products cannot be negative"

TC_UpdateProduct_14: 1-2-3-4-5-9-19-24-27-28-29

Input: id = 1, quantity = 10, available_quantity = 10

Expected result: “Product update successful”

TC_UpdateProduct_15: 1-2-3-4-5-9-19-24-25-29

Input: id = 1, quantity = 10, available_quantity = 20

Expected result: “The quantity sold must not be greater than the quantity imported”

TC_UpdateProduct_16: 1-2-3-4-5-10-21-23-29

Input: id = 1, buy_price = -1

Expected result: "Product price cannot be negative"

TC_UpdateProduct_17: 1-2-3-4-5-10-21-27-28-29

Input: id = 1, buy_price = 100

Expected result: “Product update successful”

TC_UpdateProduct_18: 1-2-3-4-5-11-22-23-29

Input: id = 1, sell_price = -1

Expected result: "Product price cannot be negative"

TC_UpdateProduct_19: 1-2-3-4-5-11-22-27-28-29

Input: id = 1, sell_price = 100

Expected result: “Product update successful”

TC_UpdateProduct_20: 1-2-3-4-5-12-27-28-29

Input: id = 1, description = “Delicious cake”

Expected result: “Product update successful”

TC_UpdateProduct_21: 1-2-3-4-5-14-27-28-29

Input: id = 1, id_cagories = “Cakes”

Expected result: “Product update successful”

TC_UpdateProduct_22: 1-2-3-4-5-15-27-28-29

Input: id = 1, id_brands = “SamSung”

Expected result: “Product update successful”

Delete product

Control flow graph diagram

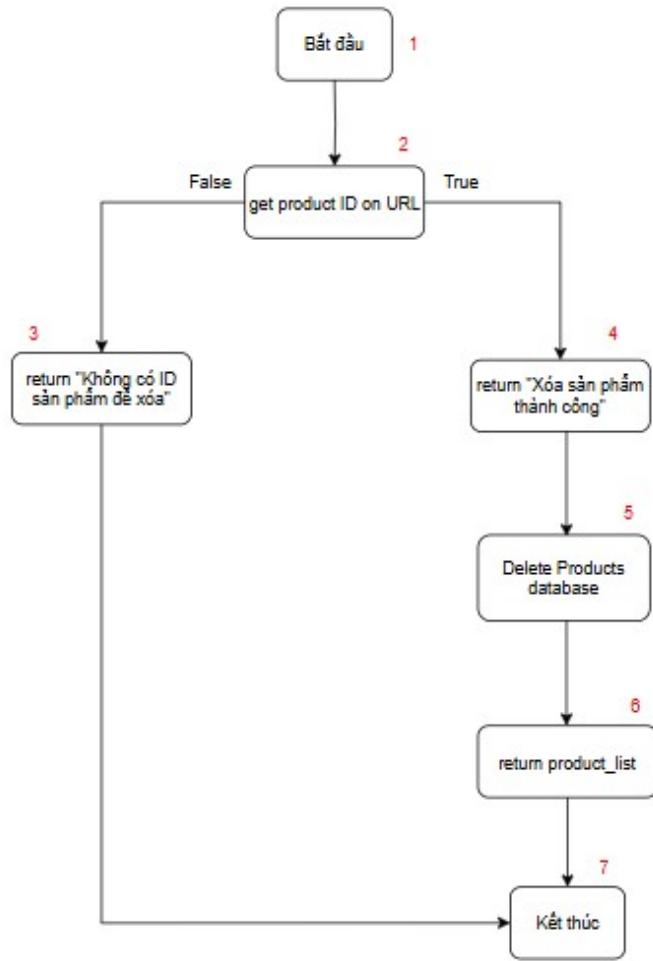


Figure 31. Control flow diagram Delete product Import goods

Complexity

$$E = \text{number of sides} = 7$$

$$N = \text{number of buttons} = 7$$

$$P = \text{number of connected components} = 1$$

$$M = E - N + 2P = 7 - 7 + 2 * 1 = 2$$

Independent road

Independent line 1: 1-2-3-7

Independent line 2: 1-2-4-5-6-7

Test Suite

TC_DeleteProduct_01: 1-2-3-7

Input: ID = 10

Expected result: “No product ID to delete”

TC_DeleteProduct_02: 1-2-4-5-6-7

Input: ID = 1

Expected result: “Product deleted successfully”

2.2.2. Purchasing process

Add product to cart

Control flow graph diagram

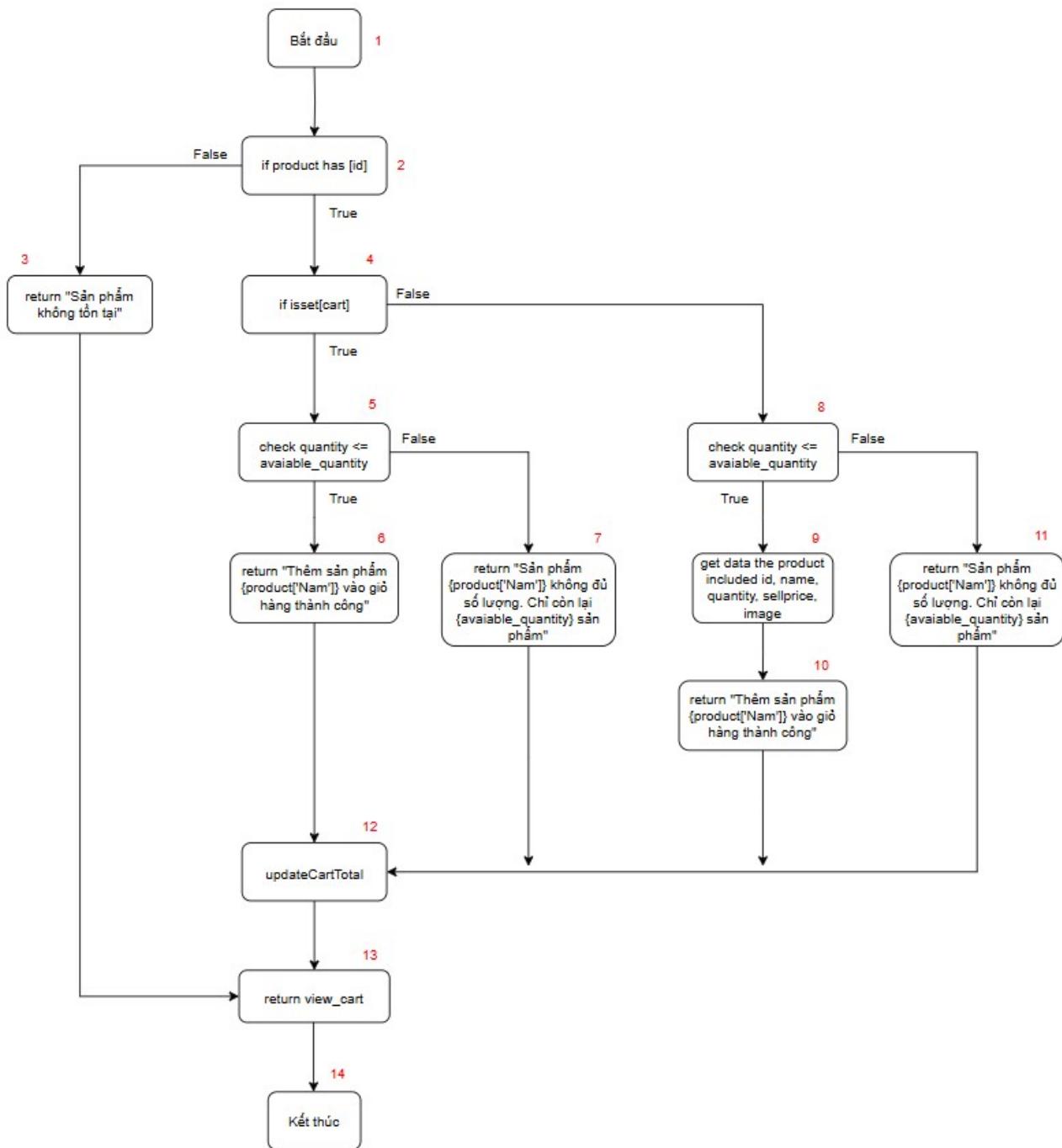


Figure 32. Control flow diagram Add product Shopping cart

Complexity

E = number of sides = 17

N = number of buttons = 14

P = number of connected components = 1

M = E - N + 2P = 18 - 15 + 2 * 1 = 5

Independent road

Independent line 1: 1-2-3-14-15

Independent line 2: 1-2-4-5-6-12-13-14

Independent line 3: 1-2-4-5-7-12-13-14

Independent line 4: 1-2-4-8-9-10-12-13-14

Independent Line 5: 1-2-4-8-11-12-13-14

Test Suite

TC_AddCart_01: 1-2-3-13-14

Input: ID = 10

Expected result: “Product does not exist”

TC_AddCart_02: 1-2-4-5-6-12-13-14

Input: ID = 1, quantity = 2, sell_price = 120

Expected result: “ Successfully added product {product['Male']} to cart ”, total amount 240USD

TC_AddCart_03: 1-2-4-5-7-12-13-14

Input: ID = 1, quantity = 100, sell_price = 120

Expected result: “ Product {product['Male']} is out of stock. Only {available_quantity} products left ”, total price 0USD

TC_AddCart_04: 1-2-4-8-9-10-12-13-14

Input: : ID = 1, quantity = 2, sell_price = 120, condition: cart has quatity = 1

Expected result: “ Successfully added product {product['Male']} to cart ”, total amount 360USD

TC_AddCart_05: 1-2-4-8-11-12-13-14

Input: ID = 1, quantity = 100, sell_price = 120, condition: cart has quatity = 1

Expected result: “ Product {product['Male']} is out of stock. Only {avaiable_quantity} products left ”, total price 120USD

Update quantity of products in cart

Control flow graph diagram

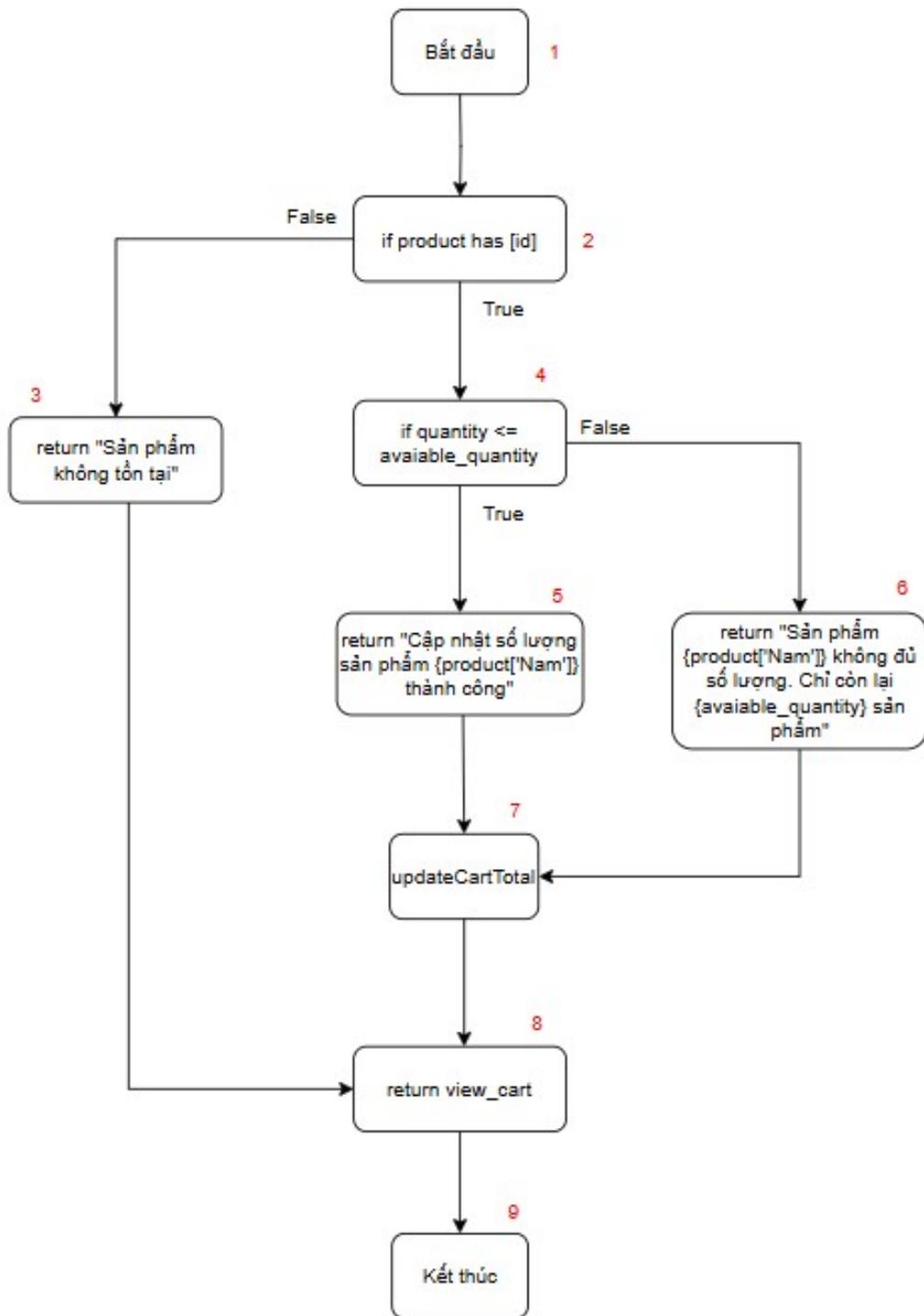


Figure 33. Control flow diagram of Update Shopping Cart

Complexity

E = number of sides = 10

N = number of buttons = 9

P = number of connected components = 1

M = E - N + 2P = 10 - 9 + 2 * 1 = 3

Independent road

Independent line 1: 1-2-3-8-9

Independent line 2: 1-2-4-5-7-8-9

Independent line 3: 1-2-4-6-7-8-9

Test Suite

TC_UpdateCart_01: 1-2-3-8-9

Input: ID = 10

Expected result: “Product does not exist”

TC_UpdateCart_02: 1-2-4-5-7-8-9

Input: ID = 1, quantity = 5, sell_price = 120, cart has quatity = 1

Expected result: “ Update product quantity {product['Male']} success ”, total 1100USD

TC_UpdateCart_03: 1-2-4-6-7-8-9

Input: ID = 1, quantity = 100, sell_price = 120, cart has quatity = 1

Expected result: “ Product {product['Male']} is out of stock. Only {available_quantity} products left ”, total price 120USD

Remove product from cart

Control flow graph diagram

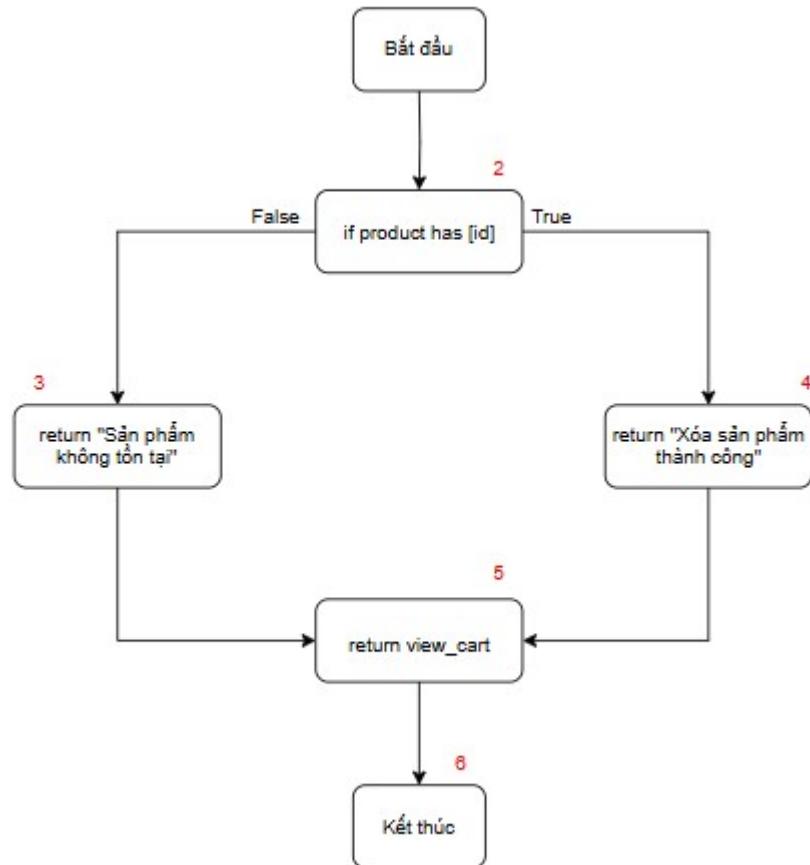


Figure 34. Control flow diagram Delete product Shopping cart

Complexity

E = number of sides = 6

N = number of buttons = 6

P = number of connected components = 1

$$M = E - N + 2P = 6 - 6 + 2 * 1 = 2$$

Independent road

Independent line 1: 1-2-3-5-6

Independent line 2: 1-2-4-5-6

Test Suite e

TC_DeleteCart_01: 1-2-3-6

Input: ID = 10

Expected result: “Product does not exist”

TC_DeleteCart_02: 1-2-4-5-6

Input: ID = 1

Expected result: “Product deleted successfully”

Checkout cart

Control flow graph diagram

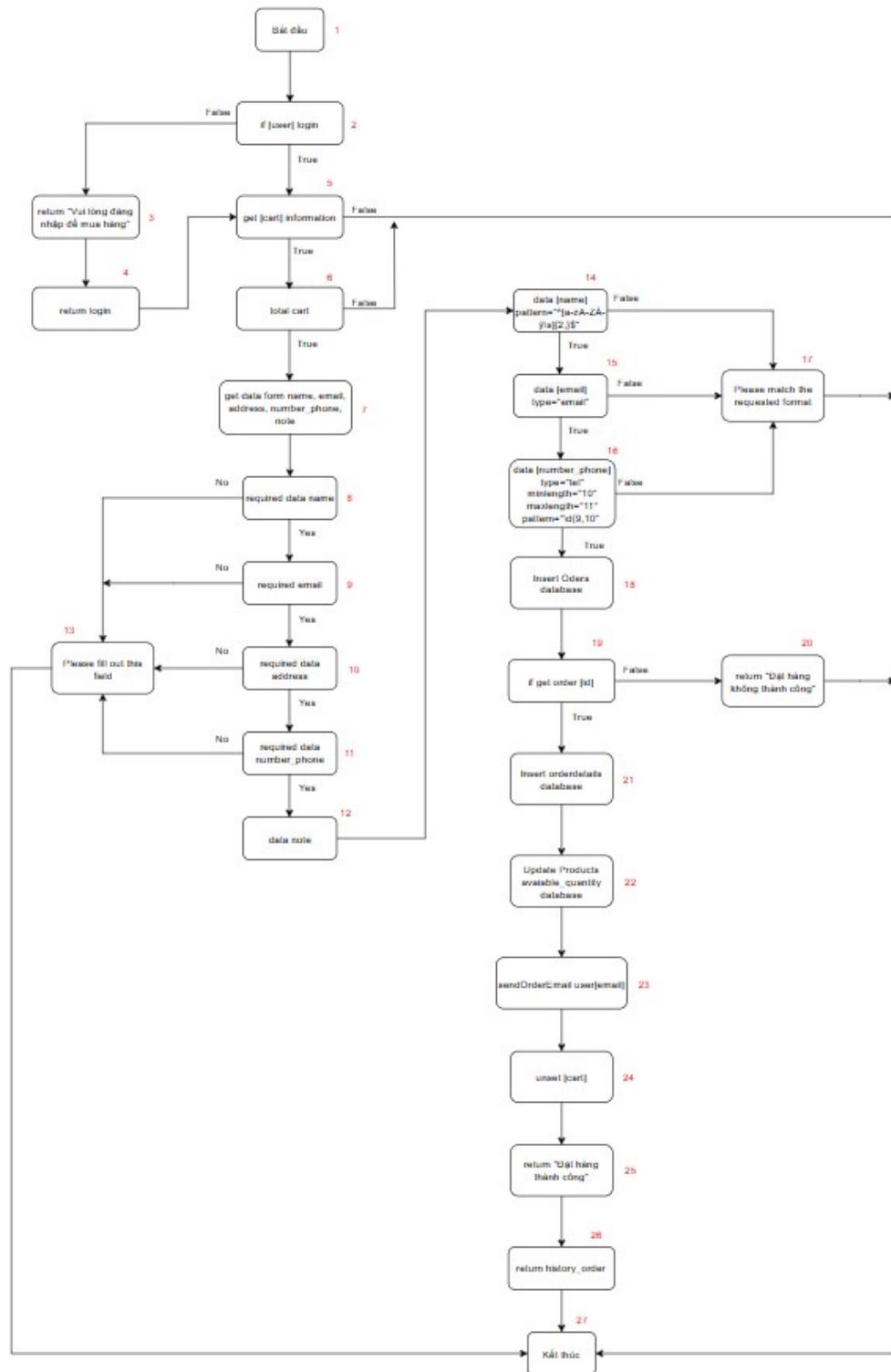


Figure 35. Control flow diagram of Checkout Shopping Cart

Complexity

E = number of sides = 37

N = number of buttons = 27

P = number of connected components = 1

M = E - N + 2P = 37-27 + 2*1 = 12

Independent road

Independent line 1: 1-2-3-4-5...

Independent Line 2: 1-2-5-27

Independent line 3: 1-2-5-6-27

Independent line 4: 1-2-5-6-7-8-13-27

Independent line 5: 1-2-5-6-7-8-9-13-27

Independent line 6: 1-2-5-6-7-8-9-10-13-27

Independent line 7: 1-2-5-6-7-8-9-10-11-13-27

Independent line 8: 1-2-5-6-7-8-9-10-11-12-14-17-27

Independent line 9: 1-2-5-6-7-8-9-10-11-12-14-15-17-27

Independent line 10: 1-2-5-6-7-8-9-10-11-12-14-15-16-17-27

Independent line 11: 1-2-5-6-7-8-9-10-11-12-14-15-16-18-19-20-27

Independent line 12: 1-2-5-6-7-8-9-10-11-12-14-15-16-18-19-21-22-23-24-25-26-27

Test Suite

TC_Checkout_01: 1-2-3-4-5...

Input: user no login

Expected result: “Please login to purchase”

TC_Checkout_02: 1-2-5-27

Input: ID = 1

Expected result: No order information displayed

TC_Checkout_03: 1-2-5-6-27

Input: sell_price = 120

Expected result: Order total not displayed

TC_Checkout_04: 1-2-5-6-7-8-13-27

Input: name = “ ”

Expected result: Please fill out this field

TC_Checkout_05: 1-2-5-6-7-8-9-13-27

Input: email = “ ”

Expected result: Please fill out this field

TC_Checkout_06: 1-2-5-6-7-8-9-10-13-27

Input: address = “ ”

Expected result: Please fill out this field

TC_Checkout_07: 1-2-5-6-7-8-9-10-11-13-27

Input: number _ phone = “ ”

Expected result: Please fill out this field

TC_Checkout_08: 1-2-5-6-7-8-9-10-11-12-14-17-27

Input: name = 123

Expected result: Please match the requested format

TC_Checkout_09: 1-2-5-6-7-8-9-10-11-12-14-15-17-27

Input: email = “ tannhut2111”

Expected result: Please match the requested format

TC_Checkout_10: 1-2-5-6-7-8-9-10-11-12-14-15-16-17-27

Input: number _ phone = “090164180000””

Expected result: Please match the requested format

TC_Checkout_11: 1-2-5-6-7-8-9-10-11-12-14-15-16-18-19-20-27

Input: ID = 10

Expected result: “Order failed”

TC_Checkout_12:1-2-5-6-7-8-9-10-11-12-14-15-16-18-19-21-22-23-24-25-26-27

Input: name = “nhut”, email = “ tannhut2111@gmail.com ”, address = “HCM”, number_phone = “0901641800”, note = “”

Expected result: “Order successfully”

2.3. Black box testing

Black-box Testing **Checklist** is a list of criteria or items to be tested to ensure that the software functions as per the requirements and expectations without regard to the source code structure or internal logic. This checklist focuses on testing the functionality and user interface of the application based on the inputs and outputs.

Mã dự án:	WBMS-001			
Phiên bản sản phẩm:	1.0.			
Người đánh giá:	Phùng Tân NhậtHuynh Anh Tuân, Phùng Tân Nhật			
Ngày đánh giá: <DD-MM-YY>	1-Dec-24			
Số lượng trường hợp kiểm thử:	119			
Thời gian dành cho việc đánh giá (giờ/người):	50 giờ			
Câu hỏi	Có	Không	Không áp dụng	Ghi chú
KIỂM SOÁT TÀI LIỆU				Ưu tiên
Kiểm tra xem các thủ tục kiểm soát tài liệu có được tuân thủ không bằng cách kiểm tra các yếu tố sau:	X			
Trang tiêu đề có chứa tên tài liệu, số phiên bản, ngày phát hành và ngày phát hành không?	X			
Phản ánh và chân trang có chí rõ tên và phiên bản tài liệu không?	X			
Số đếm số trang có chí rõ tổng số trang trong tài liệu không?	X			
Lịch sử có thể tra cứu được không?	X			Bắt buộc
Có bao gồm danh sách tài liệu tham khảo không?	X			
Tất cả tài liệu đã được kiểm tra chính tả và ngữ pháp bằng MS Word hoặc công cụ tương tự chưa?	X			
DANH SÁCH KIỂM TRA TRƯỞNG HỢP KIỂM THỬ				
Các trường hợp kiểm thử có bao phủ tất cả các yêu cầu không?	X			Bắt buộc
Các trường hợp kiểm thử có bao phủ tất cả các loại kiểm thử được mô tả trong kế hoạch kiểm thử không?	X			Bắt buộc
Mỗi trường hợp kiểm thử đã được gắn một mã trường hợp kiểm thử chưa?	X			Bắt buộc
Mỗi trường hợp kiểm thử có chí rõ:				
Hành động	X			Bắt buộc
Điều kiện kiểm thử	X			Bắt buộc
Kết quả mong đợi	X			Bắt buộc
Kết quả mong đợi đã được ghi lại chi tiết chưa?	X			Bắt buộc
Các trường hợp kiểm thử cho việc xác thực trường, xác thực bản ghi và cập nhật cơ sở dữ liệu có bao gồm các yếu tố sau không?				
Điều kiện hợp lệ	X			Bắt buộc
Điều kiện không hợp lệ	X			Bắt buộc
Điều kiện biến hoặc điều kiện bắt thường	X			Bắt buộc
Có chí rõ thông báo lỗi/mã thông báo trong kết quả mong đợi cho các trường hợp bắt thường không?	X			
Các trường hợp kiểm thử cho báo cáo có bao gồm dữ liệu kiểm thử cùng với đầu ra mong đợi không?	X			
Các phu thuộc giữa các trường hợp kiểm thử có được mô tả không?	X			
Tất cả môi trường yêu cầu có được chỉ rõ không?	X			Bắt buộc
Các điều kiện tiên quyết cho bài kiểm thử có được chỉ rõ không?	X			
Tất cả các chủ só của trường hợp kiểm thử có đạt tiêu chuẩn không?	X			

Figure 36. Black Box Checklist

Black Box Testing is a software testing method that focuses on testing the functionality of a system without regard to its internal structure. The main purpose of black box testing is to ensure that the software operates according to its specified requirements and handles all input scenarios correctly.

- Ensure software meets requirements:

- Check that the output matches expectations based on the inputs and functional requirements.
- Verify that business and technical requirements are met.

- Error detected in function:

- Incorrect input data handling.
- Miscalculation.
- Missing functionality or not handling as required.

- Focus on the external behavior of the system:

- No need to know the internal workings of the system, just care about the results.
- Suitable for testers without programming experience to perform.

- Ensure interoperability:

- Ensure that the system interacts properly with other systems or with users.

Techniques used in the project:

- **The Decision Table method** is a black box testing method used to test the combination of input conditions and output actions. This method is very effective when the system has complex conditions and needs to test all of their possibilities. Each input condition is combined with the corresponding output actions to form a rule table, which helps to determine the necessary test scenarios. It consists of 3 main parts:

Input conditions (Conditions): These are the factors that the system checks. Each condition can have a value of True or False.

Actions: Actions to be performed based on the values of input conditions. These actions can include error messages, success messages, or other system operations.

Rules: Each rule in a decision table describes a specific combination of conditions and actions. A decision table can have multiple rules, each rule testing a different scenario.

2.3.1. Import process

Add product

Thêm sản phẩm	TC_AddProduct_0	TC_AddProduct_01	TC_AddProduct_02	TC_AddProduct_03	TC_AddProduct_04	TC_AddProduct_05	TC_AddProduct_06	TC_AddProduct_07	TC_AddProduct_08	TC_AddProduct_09	TC_AddProduct_10	TC_AddProduct_11	TC_AddProduct_12	TC_AddProduct_13	TC_AddProduct_14	TC_AddProduct_15	TC_AddProduct_16
Conditions																	
Không có tên sản phẩm	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
Bố trống hình ảnh sản phẩm	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
Bố trống số lượng nhập	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N
Bố trống số lượng bán	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N
Bố trống mô tả sản phẩm	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N
Bố trống giá nhập	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N
Bố trống giá bán	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N
Không chọn loại hàng	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N
Không chọn thương hiệu	N	S	N	N	N	N	N	S	Y	N	N	N	N	N	N	N	N
Giá nhập <= số âm	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N
Giá bán <= số âm	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N
Số lượng nhập là số âm	N	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N
Số lượng bán là số âm	N	N	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N
Mã sản phẩm tồn kho số lượng nhập	N	N	N	N	N	N	N	N	N	N	N	N	N	Y	N	N	N
Mã sản phẩm tồn kho số lượng bán	N	N	N	N	N	N	N	N	N	N	N	N	N	Y	N	N	N
Triệt tiêu	N	N	N	N	N	N	N	N	N	N	N	N	N	N	Y	N	N
Actions																	
Thông báo lỗi: "Vui lòng điền vào trường này"	X		X	X	X	X	X										
Thông báo lỗi: "Vui lòng chọn 1 file"		X						X	X								
Thông báo lỗi: "Kết quả trả về tên thêm sản phẩm"									X	X							
Thông báo lỗi: "Số lượng sản phẩm không được để số âm"										X	X						
Thông báo lỗi: "Số lượng bán không được lớn hơn số lượng nhập"											X	X					
Thông báo lỗi: "Tên sản phẩm đã tồn tại"												X					
Thông báo: "Thêm sản phẩm thành công"													X				

Figure 37. Decision table Add product

Test Suite e

TC_AddProduct_01

Input: Product name = ""

Expected result: Please fill out this field

TC_AddProduct_02

Input: Image = ""

Expected result: Please fill out this field

TC_AddProduct_03

Input: Quantity entered = ""

Expected result: Please fill out this field

TC_AddProduct_04

Input : Quantity sold = ""

Expected result: Please fill out this field

TC_AddProduct_05

Input: Description = ""

Expected result: Please fill out this field

TC_AddProduct_06

Input: Input price = ""

Expected result: Please fill out this field

TC_AddProduct_07

Input: Selling price = ""

Expected result: Please fill out this field

TC_AddProduct_08

Input: Cake type = ""

Expected result: "Add product failed"

TC_AddProduct_09

Input: Brand = ""

Expected result: "Add product failed"

TC_AddProduct_10

Input: Input price = -1

Expected result: "Product price cannot be negative"

TC_AddProduct_11

Input: Selling price = -1

Expected result: "Product price cannot be negative"

TC_AddProduct_12

Input: Quantity entered = -1

Expected result: "The number of products cannot be negative"

TC_AddProduct_13

Input: Quantity sold = -1

Expected result: "The number of products cannot be negative"

TC_AddProduct_14

Input: Quantity imported = 100, Quantity sold = 120

Expected result: "The quantity sold must not be greater than the quantity imported"

TC_AddProduct_15

Input: Product name = "DULCE"

Expected result: "Product name already exists"

TC_AddProduct_16

Input: Product name = "ABC", Image = "1aba114354.product-big-4.jpg", Import quantity = 10, Sales quantity = 10, Description = "Delicious cakes", Import price = "100", Sales price = "120", Cake type = "Cakes", Brand = "SamSung"

Expected result: "Added product successfully"

Product Updates

Condition	WU_ValidateID	WU_ValidateName	WU_ValidateImage	WU_ValidatePrice	WU_ValidateQuantity	WU_ValidateUnit	WU_ValidateProduct	WU_ValidateImage	WU_ValidateName	WU_ValidateImage	WU_ValidatePrice	WU_ValidateQuantity	WU_ValidateUnit	WU_ValidateProduct	WU_ValidateImage	WU_ValidateName	WU_ValidateImage	WU_ValidatePrice	WU_ValidateQuantity	WU_ValidateUnit	WU_ValidateProduct	WU_ValidateImage	WU_ValidateName	WU_ValidateImage	WU_ValidatePrice	WU_ValidateQuantity	WU_ValidateUnit	WU_ValidateProduct	
Product ID is null	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
Product ID is blank	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
Product ID is not blank	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S
Product Name is null	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
Product Name is blank	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
Product Name is not blank	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S
Product Image is null	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
Product Image is blank	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
Product Image is not blank	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S
Product Price is null	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
Product Price is blank	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
Product Price is not blank	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S
Product Quantity is null	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
Product Quantity is blank	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
Product Quantity is not blank	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S
Product Unit is null	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
Product Unit is blank	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
Product Unit is not blank	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S
Product Product is null	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
Product Product is blank	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
Product Product is not blank	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S
Product Image is not null or blank	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S
Product Name is not null or blank	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S
Product Image is not null or blank and Product Name is not null or blank	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S
Product Image is null or blank and Product Name is null or blank	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
Product Image is not null or blank and Product Name is null or blank	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
Product Image is null or blank and Product Name is not null or blank	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
Product Image is not null or blank and Product Name is not null or blank	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S

Figure 38. Product Update Decision Table

Test Suite

TC_UpdateProduct_01

Input: ID = 10

Expected result: “Product does not exist”

TC_UpdateProduct_02

Input: ID = 1, Product name = ""

Expected result: “Please fill out this field”

TC_UpdateProduct_03

Input: ID = 1, Image = ""

Expected result: “Please fill out this field”

TC_UpdateProduct_04

Input: ID = 1, Quantity entered = ""

Expected result: “Please fill out this field”

TC_UpdateProduct_05

Input: ID = 1, Quantity sold = ""

Expected result: “Please fill out this field”

TC_UpdateProduct_06

Input: ID = 1, Input price = ""

Expected result: “Please fill out this field”

TC_UpdateProduct_07

Input: ID = 1, Price = ""

Expected result: "Please fill out this field"

TC_UpdateProduct_08

Input: ID = 1, Description = ""

Expected result: "Please fill out this field"

TC_UpdateProduct_09

Input: ID = 1, Product name = "DULCE"

Expected result: "Product name already exists"

TC_UpdateProduct_10

Input: ID 1, Product Name = "ABC"

Expected result: "Product update successful"

TC_UpdateProduct_11

Input : ID = 1, Image = "laba114354.product-big-4.jpg"

Expected result: "Product update successful"

TC_UpdateProduct_12

Input: ID = 1 , Quantity entered = -1

Expected result: "The number of products cannot be negative"

TC_UpdateProduct_13

Input: ID = 1, Quantity sold = -1

Expected result: "The number of products cannot be negative"

TC_UpdateProduct_14

Input: ID = 1, Quantity entered = 100, Quantity sold = 80

Expected result: "Product update successful"

TC_UpdateProduct_15

Input: ID = 1, Quantity entered = 100, Quantity sold = 120

Expected result: “The quantity sold must not be greater than the quantity imported”

TC_UpdateProduct_16

Input: ID = 1, Input price = -1

Expected result: "Product price cannot be negative"

TC_UpdateProduct_17

Input: ID = 1, Input price = 100

Expected result: “Product update successful”

TC_UpdateProduct_18

Input: ID = 1, Selling price = -1

Expected result: "Product price cannot be negative"

TC_UpdateProduct_19

Input: ID = 1, Price = 100

Expected result: “Product update successful”

TC_UpdateProduct_20

Input: ID = 1 , Description = “Delicious cake”

Expected result: “Product update successful”

TC_UpdateProduct_21

Input: ID = 1, Cake type = “Cakes”

Expected result: “Product update successful”

TC_UpdateProduct_22

Input: ID = 1, Brand = “SamSung”

Expected result: “Product update successful”

Delete product

Xóa sản phẩm	TC_DeleteProduct_01	TC_DeleteProduct_02
Conditions		
Sản phẩm không tồn tại	Y	N
Actions		
Thông báo lỗi "Không có ID sản phẩm để xóa"	X	
Thông báo "Xóa sản phẩm thành công"		X

Figure 39. Product Delete Decision Table

Test Suite e

TC_DeleteCart_01

Input: ID = 10

Expected result: “No product ID to delete”

TC_DeleteCart_02

Input: ID = 1

Expected result: “Product deleted successfully”

2.3.2. Purchasing process

Add product to cart

Thêm sản phẩm vào giỏ hàng	TC_AddCart_01	TC_AddCart_02	TC_AddCart_03	TC_AddCart_04	TC_AddCart_05
Conditions					
Sản phẩm không tồn tại	Y	N	N	N	N
Sản phẩm còn trong kho	N	Y	N	Y	N
Sản phẩm đã trong giỏ	N	N	N	Y	Y
Actions					
Thông báo lỗi: "Sản phẩm không tồn tại"	X				
Thông báo: "Thêm sản phẩm...vào giỏ hàng thành công"		X		X	
Thông báo lỗi: "Sản phẩm...không đủ số lượng. Chỉ còn lại...sản phẩm"			X		X

Figure 40. Decision table Add product to Cart

Test Suite e TC_AddCart_01

Input: ID = 10

Expected result: “Product does not exist”

TC_AddCart_02

Input: ID = 1, Quantity = 2, Price = 120USD

Expected result: “ Successfully added product ... to cart ”, total amount is 120USD

TC_AddCart_03

Input: ID = 1, Quantity = 100, Price = 120USD

Expected result: “ Product ... quantity is not enough. Only ... products left ”, total amount 0USD

TC_AddCart_04

Input: ID = 1, Quantity = 2, Price = 120USD, Condition: Cart quantity = 1

Expected result: “ Successfully added product ... to cart ”, total amount 360USD

TC_AddCart_05

Input: ID = 1, Quantity = 100, Price = 120USD, Condition: Cart quantity = 1

Expected result: “ Product ... quantity is not enough. Only ... product left ”, total price 120USD

Update products in cart

Cập nhật số lượng sản phẩm	TC_UpdateCart_01	TC_UpdateCart_02	TC_UpdateCart_03
Conditions			
Sản phẩm không tồn tại	Y	N	N
Sản phẩm còn trong kho	N	Y	N
Actions			
Thông báo lỗi: "Sản phẩm không tồn tại"	X		
Thông báo: "Cập nhật số lượng sản phẩm...thành công "		X	
Thông báo lỗi: "Sản phẩm...không đủ số lượng. Chỉ còn lại...sản phẩm"			X

Figure 41. Decision table Update quantity in Shopping Cart

Test Suite

TC_UpdateCart_01

Input: ID = 10

Expected result: “Product does not exist”

TC_UpdateCart_02

Input: Quantity = 5, Cart has quantity = 1

Expected result: “ Update product quantity... successfully ”, total amount 1100USD

TC_UpdateCart_03

Input: Quantity = 100, Cart has quantity = 1

Expected result: “ Product ... quantity is not enough. Only ... product left ”, total price 120USD

Remove product from cart

Xóa sản phẩm khỏi giỏ hàng	TC_DeleteCart_01	TC_DeleteCart_02
Conditions		
Sản phẩm không tồn tại	Y	N
Actions		
Thông báo lỗi: "Sản phẩm không tồn tại"	X	
Thông báo: "Xóa sản phẩm thành công "		X

Figure 42. Decision table to Remove product from shopping cart

Test Suite

TC_DeleteCart_01

Input: ID = 10

Condition: Product does not exist (N)

Expected result: “Product does not exist”

TC_DeleteCart_02

Input: ID = 1

Condition: Product does not exist (Y)

Expected result: “Product deleted successfully”

Checkout cart

Thanh toán giỏ hàng	TC1	TC2	TC3	TC4	TC5	TC6	TC7	TC8	TC9	TC10	TC11	TC12
Conditions												
Chưa đăng nhập	Y	N	N	N	N	N	N	N	N	N	N	N
Bỏ trống họ và tên	N	N	N	Y	N	N	N	N	N	N	N	N
Bỏ trống email	N	N	N	N	Y	N	N	N	N	N	N	N
Bỏ trống địa chỉ	N	N	N	N	N	Y	N	N	N	N	N	N
Bỏ trống số điện thoại	N	N	N	N	N	N	Y	N	N	N	N	N
Họ và tên là số	N	N	N	N	N	N	N	Y	N	N	N	N
Email không đúng định dạng	N	N	N	N	N	N	N	N	Y	N	N	N
Số điện thoại ít hơn 10 và lớn hơn 10 số	N	N	N	N	N	N	N	N	N	Y	N	N
Giỏ hàng không tồn tại	N	N	N	N	N	N	N	N	N	N	Y	N
Thông tin đơn hàng không hợp lệ	N	Y	N	N	N	N	N	N	N	N	N	N
Tổng tiền đơn hàng không hợp lệ	N	N	Y	N	N	N	N	N	N	N	N	N
Actions												
Thông báo: "Vui lòng đăng nhập để mua hàng"	X											
Thông báo: "Please fill out this field"			X	X	X	X						
Thông báo: "Please match the requested format"							X	X	X			
Thông báo: "Đặt hàng thành công"												X
Thông báo lỗi: "Đặt hàng không thành công"											X	
Không hiển thị		X	X								X	

Figure 43. Cart Checkout Decision Table

Test Suite

TC_Checkout_01

Input: user no login

Expected result: “Please login to purchase”

TC_Checkout_02

Input: ID = 1

Expected result: No order information displayed

TC_Checkout_03

Input: Price = 120USD

Expected result: Order total not displayed

TC_Checkout_04

Input: Full name = " "

Expected result: Please fill out this field

TC_Checkout_05

Input: Email = " "

Expected result: Please fill out this field

TC_Checkout_06

Input: Address = " "

Expected result: Please fill out this field

TC_Checkout_07

Input: Phone number = " "

Expected result: Please fill out this field

TC_Checkout_08

Input: Full name = "1"

Expected result: Please match the requested format

TC_Checkout_09

Input : Email = "tannhut2111"

Expected result: Please match the requested format

TC_Checkout_10

Input: Phone number = "090164180"

Expected result: Please match the requested format

TC_Checkout_11

Input: ID = 10

Expected result: “Order failed”

TC_Checkout_12

Input: Full name = “nhut”, Email = "tannhut2111@gmail.com", Address = “HCM”,
Phone number = “0901641800”, Note = “”

Expected result: “Order failed”

3. System Testing

Test Case is a set of conditions, inputs, actions to be taken, and expected results designed to test a specific functionality of a software or system. Each test case

describes in detail how to test and determine whether the software meets the requirements or test criteria.

Purpose of Test Case

Software quality assurance : Test cases help to check that the software works as required.

Error detected : Identify errors or defects in the system for repair.

Testing instructions : Test cases provide detailed instructions to testers to perform testing consistently.

Reuse : Test cases can be reused in subsequent tests or similar projects.

Documentation support : Test case is an important reference document in software development and maintenance process.

Verify request : Test cases help ensure that the software meets the specified technical and functional requirements.

3.1. TestCase List

No	Function Name	Module xử lý (thay đổi dữ liệu trong database)	Description	Pre-Condition
	Nhập hàng			
1	Function addProduct	Module product_add		
2	Function updateProduct	Module product_update		
3	Function deleteProduct	Module product_delete		
	Mua hàng			
4	Function addToCart	Module cart		
5	Function updateCartQuantity	Module cart		
6	Function deleteFromCart	Module cart		
7	Function place_order	Module checkout		
	Phi chức năng			

Figure 44. List of test cases

Import goods

- **Function addProduct**
 - **Processing module :** product_add
 - **Description :** This function allows the user to add a new product to the database. The information to be entered includes product name, image, quantity, price, description, cake type, and brand.
 - **Pre-Condition :** Access to the import module and valid product data is required.
- **Function updateProduct**
 - **Processing module :** product_update
 - **Description :** Function to update product information already in the system, including fields such as name, image, quantity, import price, selling price, description, cake type, and brand.
 - **Pre-Condition :** The product needs to be identified and have a valid ID before the update can be performed.
- **Function deleteProduct**
 - **Processing module :** product_delete
 - **Description :** Function to delete a product from the system using the product ID. After deletion, the product will no longer exist in the database.
 - **Pre-Condition :** The product must exist in the system and have a valid ID to be deleted.

Purchase

- **Function addToCart**
 - **Processing module :** cart
 - **Description :** Function to add products to the user's shopping cart. Users can add multiple products to the cart, and the system will store that product information in the cart.
 - **Pre-Condition :** User must be logged in and have permission to add products to cart. Product must be in stock.
- **Function updateCartQuantity**
 - **Processing module :** cart
 - **Description :** Function to update the quantity of products in the user's shopping cart. The user can change the quantity of products already in the cart.
 - **Pre-Condition :** The cart already contains a product whose quantity needs to be changed. The product must have sufficient quantity in stock.
- **Function deleteFromCart**
 - **Processing module :** cart
 - **Description :** Function to delete products from the shopping cart. Users can delete any product in their shopping cart.
 - **Pre-Condition :** The shopping cart must contain products to delete and the user must have permission to manipulate his/her shopping cart.
- **Function place_order**
 - **Processing module :** checkout
 - **Description :** Shopping cart payment function. The user will make payment when completing the shopping cart, and the system will process the order, deduct money and save transaction information.
 - **Pre-Condition :** User needs to be logged in, shopping cart cannot be empty, and payment information must be complete and valid.

Non-functional

Handle large number of users simultaneously

2-factor authentication when logging in

Protection against SQL Injection

Encrypt connections with HTTPS

3.2. Import process

Module Code	Module product_add, Module product_update, Module product_delete				Number of Test cases
Test requirement	Kiểm thử quy trình Nhập hàng				
Tester	Huỳnh Anh Tuấn	Fail	Untested	N/A	
Pass	40	0	0	0	40

Figure 45. Module and test results for the import process

Data ID	Data
Thêm sản phẩm	
Data_01	Tên sản phẩm = ""
Data_02	Hình ảnh = ""
Data_03	Số lượng nhập = ""
Data_04	Số lượng bán = ""
Data_05	Mô tả = ""
Data_06	Giá nhập = ""
Data_07	Giá bán = ""
Data_08	Loại bánh = ""
Data_09	Thương hiệu = ""
Data_10	Giá nhập = -1
Data_11	Giá bán = -1
Data_12	Số lượng nhập = -1
Data_13	Số lượng bán = -1
Data_14	Số lượng nhập = 100, Số lượng bán = 120
Data_15	Tên sản phẩm = "DULCE"
Data_16	Tên sản phẩm = "ABC", Hình ảnh = "label14354.product-big-4.jpg", Số lượng nhập = 10, Số lượng bán = 10, Mô tả = "Bánh ngon", Giá nhập = "100", Giá bán = "120", Loại bánh = "Cakes", Thương hiệu = "SamSung"
Cập nhật sản phẩm	
Data_17	ID = 10
Data_18	ID = 1
Data_19	Tên sản phẩm = ""
Data_20	Hình ảnh = ""
Data_21	Số lượng nhập = ""
Data_22	Số lượng bán = ""
Data_23	Giá nhập = ""
Data_24	Giá bán = ""
Data_25	Mô tả = ""
Data_26	Tên sản phẩm = "DULCE"
Data_27	Tên sản phẩm = "ABC"
Data_28	Hình ảnh = "label14354.product-big-4.jpg"
Data_29	Số lượng nhập = -1
Data_30	Số lượng bán = -1
Data_31	Số lượng nhập = 100, Số lượng bán: 80
Data_32	Số lượng nhập = 100, Số lượng bán: 120
Data_33	Giá nhập = -1
Data_34	Giá nhập = 100
Data_35	Giá bán = -1
Data_36	Giá bán = 100
Data_37	Mô tả = "Bánh ngon"
Data_38	Loại bánh = "Cakes"
Data_39	Thương hiệu = "SamSung"
Xóa sản phẩm	
Data_40	ID = 11
Data_41	ID = 1

Figure 46. Data Test for the import process

ID	TestCase ID	Mô tả trường hợp kiểm tra	Quy trình thực hiện kiểm tra	Kết quả mong đợi	Pha thuộc kiểm tra	Kết quả	Ngày kiểm tra	Ghi chú
Function addProduct								
Module product_add-1	TC_AddProduct_01	Kiểm tra khi bỏ trống tên sản phẩm	1. Đăng nhập vào hệ thống với vai trò SysAdmin 2. Trong danh mục SysAdmin, chọn vào mục Sản phẩm 3. Nhấn nút "Thêm mới" 4. Không nhập [Data_01] 5. Nhấn "Thêm sản phẩm"	Hiển thị thông báo lỗi "Vui lòng điền vào trường này"	Không có	Pass	11/26/2024	
Module product_add-2								
Module product_add-2	TC_AddProduct_02	Kiểm tra khi bỏ trống hình ảnh sản phẩm	1. Đăng nhập vào hệ thống với vai trò SysAdmin 2. Trong danh mục SysAdmin, chọn vào mục Sản phẩm 3. Nhấn nút "Thêm mới" 4. Không nhập [Data_02] 5. Nhấn "Thêm sản phẩm"	Hiển thị thông báo lỗi "Vui lòng chọn 1 file"	Không có	Pass	11/27/2024	
Module product_add-3								
Module product_add-3	TC_AddProduct_03	Kiểm tra khi bỏ trống số lượng nhập	1. Đăng nhập vào hệ thống với vai trò SysAdmin 2. Trong danh mục SysAdmin, chọn vào mục Sản phẩm 3. Nhấn nút "Thêm mới" 4. Không nhập [Data_03] 5. Nhấn "Thêm sản phẩm"	Hiển thị thông báo lỗi "Vui lòng điền vào trường này"	Không có	Pass	11/28/2024	
Module product_add-4								
Module product_add-4	TC_AddProduct_04	Kiểm tra khi bỏ trống số lượng bán	1. Đăng nhập vào hệ thống với vai trò SysAdmin 2. Trong danh mục SysAdmin, chọn vào mục Sản phẩm 3. Nhấn nút "Thêm mới" 4. Không nhập [Data_04] 5. Nhấn "Thêm sản phẩm"	Hiển thị thông báo lỗi "Vui lòng điền vào trường này"	Không có	Pass	11/29/2024	
Module product_add-5								
Module product_add-5	TC_AddProduct_05	Kiểm tra khi bỏ trống mô tả sản phẩm	1. Đăng nhập vào hệ thống với vai trò SysAdmin 2. Trong danh mục SysAdmin, chọn vào mục Sản phẩm 3. Nhấn nút "Thêm mới" 4. Không nhập [Data_05] 5. Nhấn "Thêm sản phẩm"	Hiển thị thông báo lỗi "Vui lòng điền vào trường này"	Không có	Pass	11/30/2024	
Module product_add-6								
Module product_add-6	TC_AddProduct_06	Kiểm tra khi bỏ trống giá nhập	1. Đăng nhập vào hệ thống với vai trò SysAdmin 2. Trong danh mục SysAdmin, chọn vào mục Sản phẩm 3. Nhấn nút "Thêm mới" 4. Không nhập [Data_06] 5. Nhấn "Thêm sản phẩm"	Hiển thị thông báo lỗi "Vui lòng điền vào trường này"	Không có	Pass	12/1/2024	
Module product_add-7								
Module product_add-7	TC_AddProduct_07	Kiểm tra khi bỏ trống giá bán	1. Đăng nhập vào hệ thống với vai trò SysAdmin 2. Trong danh mục SysAdmin, chọn vào mục Sản phẩm 3. Nhấn nút "Thêm mới" 4. Không nhập [Data_07] 5. Nhấn "Thêm sản phẩm"	Hiển thị thông báo lỗi "Vui lòng điền vào trường này"	Không có	Pass	12/2/2024	
Module product_add-8								
Module product_add-8	TC_AddProduct_08	Kiểm tra khi không chọn loại bánh	1. Đăng nhập vào hệ thống với vai trò SysAdmin 2. Trong danh mục SysAdmin, chọn vào mục Sản phẩm 3. Nhấn nút "Thêm mới" 4. Không nhập [Data_08] 5. Nhấn "Thêm sản phẩm"	Hiển thị thông báo lỗi "Xảy ra lỗi khi thêm sản phẩm"	Không có	Pass	12/3/2024	
Module product_add-9								
Module product_add-9	TC_AddProduct_09	Kiểm tra khi không chọn thương hiệu	1. Đăng nhập vào hệ thống với vai trò SysAdmin 2. Trong danh mục SysAdmin, chọn vào mục Sản phẩm 3. Nhấn nút "Thêm mới" 4. Không nhập [Data_09] 5. Nhấn "Thêm sản phẩm"	Hiển thị thông báo lỗi "Xảy ra lỗi khi thêm sản phẩm"	Không có	Pass	12/4/2024	
Module product_add-10								
Module product_add-10	TC_AddProduct_10	Kiểm tra khi giá nhập là số âm	1. Đăng nhập vào hệ thống với vai trò SysAdmin 2. Trong danh mục SysAdmin, chọn vào mục Sản phẩm 3. Nhấn nút "Thêm mới" 4. Nhập số âm [Data_10] 5. Nhấn "Thêm sản phẩm"	Hiển thị thông báo lỗi "Giá sản phẩm không được là số âm"	Không có	Pass	12/5/2024	
Module product_add-11								
Module product_add-11	TC_AddProduct_11	Kiểm tra khi giá bán là số âm	1. Đăng nhập vào hệ thống với vai trò SysAdmin 2. Trong danh mục SysAdmin, chọn vào mục Sản phẩm 3. Nhấn nút "Thêm mới" 4. Nhập số âm [Data_11] 5. Nhấn "Thêm sản phẩm"	Hiển thị thông báo lỗi "Giá sản phẩm không được là số âm"	Không có	Pass	12/6/2024	
Module product_add-12								
Module product_add-12	TC_AddProduct_12	Kiểm tra khi số lượng nhập là số âm	1. Đăng nhập vào hệ thống với vai trò SysAdmin 2. Trong danh mục SysAdmin, chọn vào mục Sản phẩm 3. Nhấn nút "Thêm mới" 4. Nhập số âm [Data_12] 5. Nhấn "Thêm sản phẩm"	Hiển thị thông báo lỗi "Số lượng sản phẩm không được là số âm"	Không có	Pass	12/7/2024	
Module product_add-13								
Module product_add-13	TC_AddProduct_13	Kiểm tra khi số lượng bán là số âm	1. Đăng nhập vào hệ thống với vai trò SysAdmin 2. Trong danh mục SysAdmin, chọn vào mục Sản phẩm 3. Nhấn nút "Thêm mới" 4. Nhập số âm [Data_13] 5. Nhấn "Thêm sản phẩm"	Hiển thị thông báo lỗi "Số lượng sản phẩm không được là số âm"	Không có	Pass	12/8/2024	
Module product_add-14								
Module product_add-14	TC_AddProduct_14	Kiểm tra khi số lượng bán lớn hơn số lượng nhập	1. Đăng nhập vào hệ thống với vai trò SysAdmin 2. Trong danh mục SysAdmin, chọn vào mục Sản phẩm 3. Nhấn nút "Thêm mới" 4. Nhập nội dung [Data_14] 5. Nhấn "Thêm sản phẩm"	Hiển thị thông báo lỗi "Số lượng bán không được lớn hơn số lượng nhập"	Không có	Pass	12/9/2024	
Module product_add-15								
Module product_add-15	TC_AddProduct_15	Kiểm tra khi thêm sản phẩm với tên bị trùng	1. Đăng nhập vào hệ thống với vai trò SysAdmin 2. Trong danh mục SysAdmin, chọn vào mục Sản phẩm 3. Nhấn nút "Thêm mới" 4. Nhập nội dung [Data_15] 5. Nhấn "Thêm sản phẩm"	Hiển thị thông báo lỗi "Tên sản phẩm đã tồn tại"	Không có	Pass	12/10/2024	
Module product_add-16								
Module product_add-16	TC_AddProduct_16	Kiểm tra khi dữ liệu đầy đủ và hợp lệ	1. Đăng nhập vào hệ thống với vai trò SysAdmin 2. Trong danh mục SysAdmin, chọn vào mục Sản phẩm 3. Nhấn nút "Thêm mới" 4. Nhập nội dung [Data_16] 5. Nhấn "Thêm sản phẩm"	Hiển thị thông báo "Thêm sản phẩm thành công"	Không có	Pass	12/11/2024	
Function updateProduct								
Module product_update-1	TC_UpdateProduct_01	Kiểm tra cập nhật sản phẩm với ID không tồn tại	1. Đăng nhập vào hệ thống với vai trò SysAdmin 2. Trong danh mục SysAdmin, chọn vào mục Sản phẩm 3. Chọn sản phẩm [Data_17] từ danh sách 4. Nhấn nút "Sửa" 5. Vào lại trang "Sửa" [Data_17] và nhấn "Cập nhật"	Hiển thị thông báo lỗi "Sản phẩm không tồn tại"	Không có	Pass	11/26/2024	
Module product_update-2								
Module product_update-2	TC_UpdateProduct_02	Kiểm tra khi bỏ trống tên sản phẩm	1. Đăng nhập vào hệ thống với vai trò SysAdmin 2. Trong danh mục SysAdmin, chọn vào mục Sản phẩm 3. Chọn sản phẩm [Data_18] từ danh sách 4. Nhấn nút "Sửa" 5. Không nhập [Data_19] 6. Nhấn "Cập nhật"	Hiển thị thông báo lỗi "Vui lòng điền vào trường này"	Module product_update-16	Pass	11/27/2024	
Module product_update-3								
Module product_update-3	TC_UpdateProduct_03	Kiểm tra khi bỏ trống hình ảnh sản phẩm	1. Đăng nhập vào hệ thống với vai trò SysAdmin 2. Trong danh mục SysAdmin, chọn vào mục Sản phẩm 3. Chọn sản phẩm [Data_19] từ danh sách 4. Nhấn nút "Sửa" 5. Không nhập [Data_20] 6. Nhấn "Cập nhật"	Hiển thị thông báo lỗi "Vui lòng chọn 1 file"	Module product_update-16	Pass	11/28/2024	
Module product_update-4								
Module product_update-4	TC_UpdateProduct_04	Kiểm tra khi bỏ trống số lượng nhập	1. Đăng nhập vào hệ thống với vai trò SysAdmin 2. Trong danh mục SysAdmin, chọn vào mục Sản phẩm 3. Chọn sản phẩm [Data_21] từ danh sách 4. Nhấn nút "Sửa" 5. Không nhập [Data_22] 6. Nhấn "Cập nhật"	Hiển thị thông báo lỗi "Vui lòng điền vào trường này"	Module product_update-16	Pass	11/29/2024	
Module product_update-5								
Module product_update-5	TC_UpdateProduct_05	Kiểm tra khi bỏ trống số lượng bán	1. Đăng nhập vào hệ thống với vai trò SysAdmin 2. Trong danh mục SysAdmin, chọn vào mục Sản phẩm 3. Chọn sản phẩm [Data_22] từ danh sách 4. Nhấn nút "Sửa" 5. Không nhập [Data_23] 6. Nhấn "Cập nhật"	Hiển thị thông báo lỗi "Vui lòng điền vào trường này"	Module product_update-16	Pass	11/30/2024	
Module product_update-6								
Module product_update-6	TC_UpdateProduct_06	Kiểm tra khi bỏ trống giá nhập	1. Đăng nhập vào hệ thống với vai trò SysAdmin 2. Trong danh mục SysAdmin, chọn vào mục Sản phẩm 3. Chọn sản phẩm [Data_23] từ danh sách 4. Nhấn nút "Sửa" 5. Không nhập [Data_24] 6. Nhấn "Cập nhật"	Hiển thị thông báo lỗi "Vui lòng điền vào trường này"	Module product_update-16	Pass	12/1/2024	

Module product_update-7	TC_UpdateProduct_07	Kiểm tra khi bỏ trống giá bán	1. Đăng nhập vào hệ thống với vai trò SysAdmin 2. Trong danh mục SysAdmin, chọn vào mục Sản phẩm 3. Chọn sản phẩm [Data_19] từ danh sách 4. Nhấn "Sửa" 5. Không nội dung [Data_24] 6. Nhấn "Cập nhật"	Hiển thị thông báo lỗi "Vui lòng điền vào trường này."	Module product_add-16	Pass	12/2/2024
Module product_update-8	TC_UpdateProduct_08	Kiểm tra khi bỏ trống mô tả sản phẩm	1. Đăng nhập vào hệ thống với vai trò SysAdmin 2. Trong danh mục SysAdmin, chọn vào mục Sản phẩm 3. Chọn sản phẩm [Data_19] từ danh sách 4. Nhấn "Sửa" 5. Không nội dung [Data_25] 6. Nhấn "Cập nhật"	Hiển thị thông báo lỗi "Vui lòng điền vào trường này."	Module product_add-16	Pass	12/3/2024
Module product_update-9	TC_UpdateProduct_09	Kiểm tra cập nhật sản phẩm với tên bị trùng	1. Đăng nhập vào hệ thống với vai trò SysAdmin 2. Trong danh mục SysAdmin, chọn vào mục Sản phẩm 3. Chọn sản phẩm [Data_19] từ danh sách 4. Nhấn "Sửa" 5. Không nội dung [Data_26] 6. Nhấn "Cập nhật"	Hiển thị thông báo lỗi "Tên sản phẩm đã tồn tại"	Module product_add-16	Pass	12/4/2024
Module product_update-10	TC_UpdateProduct_10	Kiểm tra cập nhật sản phẩm với tên hợp lệ	1. Đăng nhập vào hệ thống với vai trò SysAdmin 2. Trong danh mục SysAdmin, chọn vào mục Sản phẩm 3. Chọn sản phẩm [Data_19] từ danh sách 4. Nhấn "Sửa" 5. Không nội dung [Data_27] 6. Nhấn "Cập nhật"	Hiển thị thông báo "Cập nhật sản phẩm thành công"	Module product_add-16	Pass	12/5/2024
Module product_update-11	TC_UpdateProduct_11	Kiểm tra cập nhật sản phẩm với hình ảnh hợp lệ	1. Đăng nhập vào hệ thống với vai trò SysAdmin 2. Trong danh mục SysAdmin, chọn vào mục Sản phẩm 3. Chọn sản phẩm [Data_19] từ danh sách 4. Nhấn "Sửa" 5. Không nội dung [Data_28] 6. Nhấn "Cập nhật"	Hiển thị thông báo "Cập nhật sản phẩm thành công"	Module product_add-16	Pass	12/6/2024
Module product_update-12	TC_UpdateProduct_12	Kiểm tra khi số lượng nhập là số âm	1. Đăng nhập vào hệ thống với vai trò SysAdmin 2. Trong danh mục SysAdmin, chọn vào mục Sản phẩm 3. Chọn sản phẩm [Data_19] từ danh sách 4. Nhấn "Sửa" 5. Không nội dung [Data_29] 6. Nhấn "Cập nhật"	Hiển thị thông báo lỗi "Số lượng sản phẩm không được là số âm"	Module product_add-16	Pass	12/7/2024
Module product_update-13	TC_UpdateProduct_13	Kiểm tra khi số lượng bán là số âm	1. Đăng nhập vào hệ thống với vai trò SysAdmin 2. Trong danh mục SysAdmin, chọn vào mục Sản phẩm 3. Chọn sản phẩm [Data_19] từ danh sách 4. Nhấn "Sửa" 5. Không nội dung [Data_30] 6. Nhấn "Cập nhật"	Hiển thị thông báo lỗi "Số lượng sản phẩm không được là số âm"	Module product_add-16	Pass	12/8/2024
Module product_update-14	TC_UpdateProduct_14	Kiểm tra cập nhật số lượng bán hợp lệ	1. Đăng nhập vào hệ thống với vai trò SysAdmin 2. Trong danh mục SysAdmin, chọn vào mục Sản phẩm 3. Chọn sản phẩm [Data_19] từ danh sách 4. Nhấn "Sửa" 5. Không nội dung [Data_31] 6. Nhấn "Cập nhật"	Hiển thị thông báo "Cập nhật sản phẩm thành công"	Module product_add-16	Pass	12/9/2024
Module product_update-15	TC_UpdateProduct_15	Kiểm tra khi số lượng bán lớn hơn số lượng nhập	1. Đăng nhập vào hệ thống với vai trò SysAdmin 2. Trong danh mục SysAdmin, chọn vào mục Sản phẩm 3. Chọn sản phẩm [Data_19] từ danh sách 4. Nhấn "Sửa" 5. Không nội dung [Data_32] 6. Nhấn "Cập nhật"	Hiển thị thông báo lỗi "Số lượng bán không được lớn hơn số lượng nhập"	Module product_add-16	Pass	12/10/2024
Module product_update-16	TC_UpdateProduct_16	Kiểm tra khi giá nhập là số âm	1. Đăng nhập vào hệ thống với vai trò SysAdmin 2. Trong danh mục SysAdmin, chọn vào mục Sản phẩm 3. Chọn sản phẩm [Data_19] từ danh sách 4. Nhấn "Sửa" 5. Không nội dung [Data_33] 6. Nhấn "Cập nhật"	Hiển thị thông báo lỗi "Giá sản phẩm không được là số âm"	Module product_add-16	Pass	12/11/2024
Module product_update-17	TC_UpdateProduct_17	Kiểm tra cập nhật sản phẩm với giá nhập hợp lệ	1. Đăng nhập vào hệ thống với vai trò SysAdmin 2. Trong danh mục SysAdmin, chọn vào mục Sản phẩm 3. Chọn sản phẩm [Data_19] từ danh sách 4. Nhấn "Sửa" 5. Không nội dung [Data_34] 6. Nhấn "Cập nhật"	Hiển thị thông báo "Cập nhật sản phẩm thành công"	Module product_add-16	Pass	12/12/2024
Module product_update-18	TC_UpdateProduct_18	Kiểm tra khi giá bán là số âm	1. Đăng nhập vào hệ thống với vai trò SysAdmin 2. Trong danh mục SysAdmin, chọn vào mục Sản phẩm 3. Chọn sản phẩm [Data_19] từ danh sách 4. Nhấn "Sửa" 5. Không nội dung [Data_35] 6. Nhấn "Cập nhật"	Hiển thị thông báo lỗi "Giá sản phẩm không được là số âm"	Module product_add-16	Pass	12/13/2024
Module product_update-19	TC_UpdateProduct_19	Kiểm tra khi cập nhật sản phẩm với giá bán hợp lệ	1. Đăng nhập vào hệ thống với vai trò SysAdmin 2. Trong danh mục SysAdmin, chọn vào mục Sản phẩm 3. Chọn sản phẩm [Data_19] từ danh sách 4. Nhấn "Sửa" 5. Không nội dung [Data_36] 6. Nhấn "Cập nhật"	Hiển thị thông báo "Cập nhật sản phẩm thành công"	Module product_add-16	Pass	12/14/2024
Module product_update-20	TC_UpdateProduct_20	Kiểm tra cập nhật sản phẩm với mã là hợp lệ	1. Đăng nhập vào hệ thống với vai trò SysAdmin 2. Trong danh mục SysAdmin, chọn vào mục Sản phẩm 3. Chọn sản phẩm [Data_19] từ danh sách 4. Nhấn "Sửa" 5. Không nội dung [Data_37] 6. Nhấn "Cập nhật"	Hiển thị thông báo "Cập nhật sản phẩm thành công"	Module product_add-16	Pass	12/15/2024
Module product_update-21	TC_UpdateProduct_21	Kiểm tra cập nhật sản phẩm với loại bánh hợp lệ	1. Đăng nhập vào hệ thống với vai trò SysAdmin 2. Trong danh mục SysAdmin, chọn vào mục Sản phẩm 3. Chọn sản phẩm [Data_19] từ danh sách 4. Nhấn "Sửa" 5. Không nội dung [Data_38] 6. Nhấn "Cập nhật"	Hiển thị thông báo "Cập nhật sản phẩm thành công"	Module product_add-16	Pass	12/16/2024
Module product_update-22	TC_UpdateProduct_22	Kiểm tra cập nhật sản phẩm với thương hiệu hợp lệ	1. Đăng nhập vào hệ thống với vai trò SysAdmin 2. Trong danh mục SysAdmin, chọn vào mục Sản phẩm 3. Chọn sản phẩm [Data_19] từ danh sách 4. Nhấn "Sửa" 5. Không nội dung [Data_39] 6. Nhấn "Cập nhật"	Hiển thị thông báo "Cập nhật sản phẩm thành công"	Module product_add-16	Pass	12/17/2024
Function deleteProduct							
Module product_deleted-1	TC_DeleteProduct_01	Kiểm tra xóa sản phẩm có ID không tồn tại	1. Đăng nhập vào hệ thống với vai trò SysAdmin 2. Trong danh mục SysAdmin, chọn vào mục Sản phẩm 3. Chọn sản phẩm [Data_40] từ danh sách 4. Nhấn "Xóa" 5. Vào database xóa [Data_40]	Hiển thị thông báo lỗi "Không có id sản phẩm để xóa"	Không có	Pass	11/26/2024
Module product_deleted-2	TC_DeleteProduct_02	Kiểm tra xóa sản phẩm có ID hợp lệ	1. Đăng nhập vào hệ thống với vai trò SysAdmin 2. Trong danh mục SysAdmin, chọn vào mục Sản phẩm 3. Chọn sản phẩm [Data_19] từ danh sách 4. Nhấn "Xóa" 5. Vào database xóa [Data_40] và Nhấn OK để xác nhận xóa	Hiển thị thông báo "Xóa sản phẩm thành công"	Module product_add-16	Pass	11/26/2024

Figure 47. TestCase of import process

ID : Identifier for the test case.

TestCase ID : The ID of the test case, used for tracking and reference.

Test case description : A description of the purpose of the test case.

Test Execution Process : The detailed steps that the tester will perform to test the feature.

Expected result : The system will display an error message asking the user to fill in the product name field completely.

Inter-case Dependencies : If this test case depends on another test case, this information will be recorded.

Output : The actual result of the test case, which determines whether the system behaves as expected.

Test date : Date the test was performed.

Notes : Additional notes about the test, such as any issues that arose or points of attention.

3.3. Purchasing process

Module Code	Module name / Module Checklist
Test requirement	Ciền thẻ ghi minh Mua hàng
Tester	Phong Tân Ngọc
Pass	Fail

Figure 48. Module and test results for the purchasing process

Thêm sản phẩm vào giỏ hàng	
Data ID	Data
Data_01	ID = 12
Data_02	ID = 1
Data_03	Số lượng = 2
Data_04	Số lượng = 100
Data_05	Số lượng = 1
Data_06	Giá = 120USD
Cập nhật số lượng sản phẩm trong giỏ hàng	
Data_06	ID = 13
Data_07	ID = 1
Data_08	Số lượng = 1
Data_09	Số lượng = 5
Data_10	Số lượng = 100
Data_11	Giá = 120USD
Xóa sản phẩm khỏi giỏ hàng	
Data_12	ID = 13
Data_13	ID = 1
Thanh toán	
Data_14	
Data_15	Họ và tên = **
Data_16	Email = **
Data_17	Địa chỉ = **
Data_18	Số điện thoại = **
Data_19	Họ và tên = "L"
Data_20	Email = "tannhut2111"
Data_21	Số điện thoại = "090164180"
Data_22	Họ và tên = "như", Email = "tannhut2111@gmail.com", Địa chỉ = "HCM", Số điện thoại = "0901641800", Ghi chú = "l"
Data_23	ID = 1
Data_23	Giá = 120USD

Figure 49. Test Data for the purchasing process

ID	Test Case ID	Mô tả chi tiết của test case	Quy trình thực hiện kiểm tra	Đánh giá mong đợi	Phân tích kiểm thử	Độ phức tạp	Ngày hoàn thành	Phi tập
Module can-1	TC_A44Cart_01	Kiểm tra thêm sản phẩm khi sản phẩm không tồn tại	1. Tạo danh mục Buyer, chọn vào mục: Các hàng 2. Vào danh mục "Sản phẩm" và click "Thêm mới" [Danh mục] 3. Tạo danh mục Buyer, chọn vào mục: Các hàng 4. Chọn sản phẩm có [Danh mục] là "Các hàng" 5. Nhập số lượng mua: 1 6. Nhập số tiền mua: 1000 7. Tạo danh mục Buyer, chọn vào mục: Các hàng 8. Chọn sản phẩm có [Danh mục] là "Các hàng" 9. Nhập số lượng mua: 1 10. Nhập số tiền mua: 1000	Hiển thị thông báo: "Thêm sản phẩm không tồn tại"	Không có	Pass	11/26/2024	
Module can-2	TC_A44Cart_02	Kiểm tra thêm sản phẩm khi sản phẩm có sẵn và số lượng trong kho và số lượng mua	1. Tạo danh mục Buyer, chọn vào mục: Các hàng 2. Vào danh mục "Sản phẩm" và click "Thêm mới" [Danh mục] 3. Tạo danh mục Buyer, chọn vào mục: Các hàng 4. Chọn sản phẩm có [Danh mục] là "Các hàng" 5. Nhập số lượng mua: 1 6. Nhập số tiền mua: 1000 7. Tạo danh mục Buyer, chọn vào mục: Các hàng 8. Chọn sản phẩm có [Danh mục] là "Các hàng" 9. Nhập số lượng mua: 1 10. Nhập số tiền mua: 1000	Hiển thị thông báo: "Thêm sản phẩm..vào giỏ hàng thành công", tổng tiền là 2000USD	Không có	Pass	11/27/2024	
Module can-3	TC_A44Cart_03	Kiểm tra thêm sản phẩm khi số lượng sản phẩm vượt quá số lượng trong kho và số lượng mua	1. Tạo danh mục Buyer, chọn vào mục: Các hàng 2. Vào danh mục "Sản phẩm" và click "Thêm mới" [Danh mục] 3. Tạo danh mục Buyer, chọn vào mục: Các hàng 4. Chọn sản phẩm có [Danh mục] là "Các hàng" 5. Nhập số lượng mua: 2 6. Nhập số tiền mua: 2000 7. Tạo danh mục Buyer, chọn vào mục: Các hàng 8. Chọn sản phẩm có [Danh mục] là "Các hàng" 9. Nhập số lượng mua: 2 10. Nhập số tiền mua: 2000	Hiển thị thông báo: "Thêm sản phẩm..vô số lượng. Chỉ còn lại: sản phẩm", tổng tiền 0USD	Không có	Pass	11/28/2024	
Module can-4	TC_A44Cart_04	Kiểm tra thêm sản phẩm khi sản phẩm có sẵn trong giỏ hàng và có dù số lượng trong kho và tổng tiền	1. Vào giỏ hàng bằng cách nhấp vào biểu tượng giỏ hàng 2. Vào danh mục "Sản phẩm" và click "Thêm mới" [Danh mục] 3. Tạo danh mục Buyer, chọn vào mục: Các hàng 4. Chọn sản phẩm có [Danh mục] là "Các hàng" 5. Nhập số lượng mua: 1 6. Nhập số tiền mua: 1000 7. Nhập số lượng mua: 1 8. Nhập số tiền mua: 1000	Hiển thị thông báo: "Thêm sản phẩm..vào giỏ hàng thành công", tổng tiền là 2000USD	Module can-2	Pass	11/27/2024	
Module can-5	TC_A44Cart_05	Kiểm tra thêm sản phẩm khi sản phẩm có sẵn trong giỏ hàng và vượt quá số lượng trong kho và tổng tiền	1. Vào giỏ hàng bằng cách nhấp vào biểu tượng giỏ hàng 2. Vào danh mục "Sản phẩm" và click "Thêm mới" [Danh mục] 3. Tạo danh mục Buyer, chọn vào mục: Các hàng 4. Chọn sản phẩm có [Danh mục] là "Các hàng" 5. Nhập số lượng mua: 2 6. Nhập số tiền mua: 2000 7. Nhập số lượng mua: 2 8. Nhập số tiền mua: 2000	Hiển thị thông báo: "Thêm sản phẩm..vô số lượng. Chỉ còn lại: sản phẩm", tổng tiền 0USD	Module can-2	Pass	11/28/2024	
Function updateCartQuantity								
Module can-6	TC_UpdateCart_01	Kiểm tra cập nhật số lượng sản phẩm khi sản phẩm không tồn tại	1. Vào giỏ hàng bằng cách nhấp vào biểu tượng giỏ hàng 2. Vào danh mục "Sản phẩm" và click "Thêm mới" [Danh mục] 3. Tạo danh mục Buyer, chọn vào mục: Các hàng 4. Chọn sản phẩm có [Danh mục] là "Các hàng" 5. Nhập số lượng mua: 1 6. Nhập số tiền mua: 1000 7. Nhập số lượng mua: 1 8. Nhập số tiền mua: 1000	Hiển thị thông báo: "Sản phẩm không tồn tại"	Không có	Pass	11/26/2024	
Module can-7	TC_UpdateCart_02	Kiểm tra cập nhật số lượng sản phẩm khi sản phẩm có sẵn và số lượng trong kho và số lượng mua	1. Vào giỏ hàng bằng cách nhấp vào biểu tượng giỏ hàng 2. Vào danh mục "Sản phẩm" và click "Thêm mới" [Danh mục] 3. Tạo danh mục Buyer, chọn vào mục: Các hàng 4. Chọn sản phẩm có [Danh mục] là "Các hàng" 5. Nhập số lượng mua: 1 6. Nhập số tiền mua: 1000 7. Nhập số lượng mua: 1 8. Nhập số tiền mua: 1000	Hiển thị thông báo: "Cập nhật số lượng sản phẩm..thành công", tổng tiền 1000USD	Module can-2, Module can-4	Pass	11/26/2024	
Module can-8	TC_UpdateCart_03	Kiểm tra cập nhật số lượng sản phẩm vượt quá số lượng trong kho và số lượng mua	1. Vào giỏ hàng bằng cách nhấp vào biểu tượng giỏ hàng 2. Vào danh mục "Sản phẩm" và click "Thêm mới" [Danh mục] 3. Tạo danh mục Buyer, chọn vào mục: Các hàng 4. Chọn sản phẩm có [Danh mục] là "Các hàng" 5. Nhập số lượng mua: 2 6. Nhập số tiền mua: 2000 7. Nhập số lượng mua: 2 8. Nhập số tiền mua: 2000	Hiển thị thông báo: "Sản phẩm..vô số lượng. Chỉ còn lại: sản phẩm", tổng tiền 1000USD	Module can-2, Module can-4	Pass	11/27/2024	
Function deleteCart								
Module can-9	TC_DeleteCart_01	Kiểm tra xóa sản phẩm với ID không tồn tại	1. Vào giỏ hàng bằng cách nhấp vào biểu tượng giỏ hàng 2. Chọn sản phẩm với ID không tồn tại 3. Nhập số lượng mua: 1	Hiển thị thông báo: "Sản phẩm không tồn tại"	Module can-2, Module can-4	Pass	11/26/2024	
Module can-10	TC_DeleteCart_02	Kiểm tra xóa sản phẩm với ID tồn tại	1. Vào giỏ hàng bằng cách nhấp vào biểu tượng giỏ hàng 2. Chọn sản phẩm với ID tồn tại 3. Nhập số lượng mua: 1	Hiển thị thông báo: "Xóa sản phẩm thành công"	Module can-2, Module can-4	Pass	11/27/2024	
Function place_order								
Module Checkin-1	TC_Checkin_01	Kiểm tra lỗi khi nhập không đúng	1. Vào giỏ hàng và nhấn "Thanh toán" 2. Nhập số điện thoại không hợp lệ	Hiển thị thông báo: "Vui lòng nhập số điện thoại"	Module can-2, Module can-4	Pass	11/27/2024	
Module Checkin-2	TC_Checkin_02	Kiểm tra lỗi khi nhập tên và họ không hợp lệ	1. Vào giỏ hàng và nhấn "Thanh toán" 2. Nhập tên và họ không hợp lệ	Không hiển thị thông tin đơn hàng	Module can-2, Module can-4	Pass	12/2/2024	
Module Checkin-3	TC_Checkin_03	Kiểm tra lỗi khi nhập tên và họ không hợp lệ	1. Vào giỏ hàng và nhấn "Thanh toán" 2. Nhập tên và họ không hợp lệ	Không hiển thị thông tin đơn hàng	Module can-2, Module can-4	Pass	12/6/2024	
Module Checkin-4	TC_Checkin_04	Kiểm tra lỗi khi nhập họ và tên không đúng	1. Vào giỏ hàng và nhấn "Thanh toán" 2. Nhập họ và tên không đúng	Hiển thị thông báo: "Vui lòng chấp nhận tên và họ"	Module can-2, Module can-4	Pass	11/28/2024	
Module Checkin-5	TC_Checkin_05	Kiểm tra lỗi nhập email không đúng	1. Đăng nhập và điền số điện thoại 2. Nhập email không đúng	Hiển thị thông báo: "Vui lòng chấp nhận tên và họ"	Module can-2, Module can-4	Pass	11/29/2024	
Module Checkin-6	TC_Checkin_06	Kiểm tra lỗi nhập địa chỉ không đúng	1. Đăng nhập và điền số điện thoại 2. Nhập địa chỉ không đúng	Hiển thị thông báo: "Vui lòng chấp nhận tên và họ"	Module can-2, Module can-4	Pass	11/30/2024	
Module Checkin-7	TC_Checkin_07	Kiểm tra lỗi nhập số điện thoại không đúng	1. Đăng nhập và điền số điện thoại 2. Nhập số điện thoại không đúng	Hiển thị thông báo: "Vui lòng chấp nhận tên và họ"	Module can-2, Module can-4	Pass	12/1/2024	
Module Checkin-8	TC_Checkin_08	Kiểm tra lỗi nhập họ và tên là số	1. Đăng nhập và điền số điện thoại 2. Nhập họ và tên là số	Hiển thị thông báo: "Không đúng định dạng"	Module can-2, Module can-4	Pass	12/2/2024	
Module Checkin-9	TC_Checkin_09	Kiểm tra lỗi nhập email không đúng	1. Đăng nhập và điền số điện thoại 2. Nhập email không đúng	Hiển thị thông báo: "Không đúng định dạng"	Module can-2, Module can-4	Pass	12/3/2024	
Module Checkin-10	TC_Checkin_10	Kiểm tra lỗi số điện thoại từ 10 và lớn hơn 10 số	1. Đăng nhập và điền số điện thoại 2. Nhập số điện thoại từ 10 và lớn hơn 10 số	Hiển thị thông báo: "Không đúng định dạng"	Module can-2, Module can-4	Pass	12/4/2024	
Module Checkin-11	TC_Checkin_11	Kiểm tra thao tác khi ID giỏ hàng không tồn tại	1. Đăng nhập và điền số điện thoại 2. Nhập ID giỏ hàng không tồn tại	Hiển thị thông báo: "Đặt hàng thành công"	Module can-2, Module can-4	Pass	12/5/2024	
Module Checkin-12	TC_Checkin_12	Kiểm tra lỗi dữ liệu thao tác độ dày và hợp lệ	1. Đăng nhập và điền số điện thoại 2. Nhập ID giỏ hàng không tồn tại	Hiển thị thông báo: "Đặt hàng thành công"	Module can-2, Module can-4	Pass	12/6/2024	

Figure 50. Purchase Process TestCase

ID : Identifier for the test case.

TestCase ID : The ID of the test case, used for tracking and reference.

Test case description : A description of the purpose of the test case.

Test Execution Process : The detailed steps that the tester will perform to test the feature.

Expected result : The system will display an error message asking the user to fill in the product name field completely.

Inter-case Dependencies : If this test case depends on another test case, this information will be recorded.

Output : The actual result of the test case, which determines whether the system behaves as expected.

Test date : Date the test was performed.

Notes : Additional notes about the test, such as any issues that arose or points of attention.

3.4 . Non-functional

Module Code	Không
Test requirement	Kiểm tra phi chức năng
Tester	Trần Gia Phúc
Pass	Fail
	Untested
	N/A
	Number of test cases
	2

Figure 51. Test results for non-functionality

Data ID	Data
Data_01	Tên đăng nhập = "tannhuu2111@gmail.com"
Data_02	Mật khẩu = 12345
Data_03	Tên đăng nhập = 'OR '1'=1'

Figure 52. Test Data for Non-Functional

TestCase ID	Mô tả trường hợp kiểm tra	Quy trình thực hiện kiểm tra	Kết quả mong đợi	Phụ thuộc liên trường hợp	Kết quả	Ngày kiểm tra	Ghi chú
TC_Nofunction_01	Xử lý đồng thời số lượng lớn người dùng truy cập (1000 user)	1. Khởi động XAMPP 2. Mở ứng dụng JMeter Tạo Test Plan: 3. Mở JMeter và tạo mới Test Plan mới 4. Nhập chuỗi phím vào Test Plan → Add → Threads (Users) → Thread Group 5. Cấu hình Thread Group: Number of Threads: 1000, Ramp-Up Period (seconds): 5, Loop Count: 1 Thread Group Request Sampler 6. Nhập chuỗi phím vào Thread Group → Add → Sampler → HTTP Request 7. Cấu hình HTTP Request: Protocol: http, Server Name or IP: localhost, Port: 80, Path: /cake-mania/list_product.php 8. Nhập chuỗi phím vào Listener → Add → Listener → Summary Report 9. Nhập chuỗi phím vào Thread Group → Add → Listener → Summary Report 10. Thêm View Results Tree để xem chi tiết phản hồi từng yêu cầu 11. Nhấn nút Start để xem kết quả	Hết thời gian hoạt động mượt mà, không bị treo hoặc chậm. Thời gian phản hồi trung bình dưới 17 giây	Không có	Pass	11/26/2024	
TC_Nofunction_02	Xác thực 2 yếu tố khi đăng nhập	1. Truy cập vào trang đăng nhập của hệ thống 2. Nhập tên đăng nhập và mật khẩu [Data_01] 3. Nhập mã xác thực OTP và nhấn [Data_02] 4. Nhấn nút "Đăng nhập".	Người dùng phải nhập mã OTP để hoàn tất đăng nhập. Không thể đăng nhập nếu mã OTP không hợp lệ hoặc không nhập mã.	Không có	Fail	11/26/2024	Không có xác thực OTP
TC_Nofunction_03	Bảo vệ chống SQL Injection	1. Truy cập vào trang đăng nhập của hệ thống 2. Nhập tên đăng nhập và mật khẩu [Data_01] 3. Nhập bất kỳ mã khán cáo vào trường "Mật khẩu". 4. Nhấn nút "Đăng nhập".	Hệ thống không thực thi câu lệnh SQL và hiển thị lỗi "Đó là dữ liệu không hợp lệ".	Node-1	Pass	11/27/2024	
TC_Nofunction_04	Mã hóa kết nối bằng HTTPS	1. Mở trình duyệt và nhập địa chỉ URL cần kiểm tra 2. Kiểm tra rằng kết nối được mã hóa thành https://... 3. Xem liệu giao thức là HTTPS hay không. 4. Thử truy cập lại hệ thống bằng HTTP thay vì HTTPS.	Tất cả các kết nối được bảo vệ bằng HTTPS. Không thể truy cập hệ thống qua HTTP.	Không có	Fail	11/28/2024	

Figure 53. Non-functional TestCase

ID : Identifier for the test case.

TestCase ID : The ID of the test case, used for tracking and reference.

Test case description : A description of the purpose of the test case.

Test Execution Process : The detailed steps that the tester will perform to test the feature.

Expected result : The system will display an error message asking the user to fill in the product name field completely.

Inter-case Dependencies : If this test case depends on another test case, this information will be recorded.

Output : The actual result of the test case, which determines whether the system behaves as expected.

Test date : Date the test was performed.

Notes : Additional notes about the test, such as any issues that arose or points of attention.

4. Acceptance Testing

Acceptance Testing is the final testing phase before the software is deployed or released. Its main purpose is to ensure that the software meets all the business requirements and can function the way the customer or end user expects.

Objective of Acceptance Testing

- Verify that the software meets business requirements:
- Check that the software fulfills the functions and features that the customer expects.
- Ensuring Go-Live readiness:
- Ensure that the software is free of major bugs and can operate stably in a real environment.

No.	Module / Screen	Description	Creator	Create date	Answer	Answer date	Status
Nhập hàng							
1	Module product_add	Trong quá trình thêm sản phẩm, nếu người dùng không nhập đầy đủ thông tin, hệ thống có cần hiển thị thông báo lỗi chi tiết không?	Huynh Anh Tuấn	11/26/2024	Có, tôi muốn hệ thống hiển thị thông báo chi tiết để biết rõ cần nhập thông tin gì	11/27/2024	Open
2	Module product_update	Chức năng sửa sản phẩm có cần kiểm tra xem sản phẩm đang được sử dụng trong giỏ hàng hoặc đơn hàng trước khi cho phép chỉnh sửa không?	Huynh Anh Tuấn	11/26/2024	Có, tôi muốn hệ thống kiểm tra trước khi chỉnh sửa để tránh làm ảnh hưởng đến giỏ hàng và đơn hàng đã có.	11/27/2024	Open
3	Module product_update	Khi cập nhật thông tin sản phẩm, hệ thống có cần xác nhận lại số lượng nhập để tránh lỗi số lượng ban đầu vượt quá số lượng nhập?	Huynh Anh Tuấn	11/26/2024	Có, hệ thống cần kiểm tra số lượng nhập và hiển thị thông báo nếu số lượng ban đầu vượt quá số lượng nhập.	11/27/2024	Open
4	Module product_delete	Khi xóa sản phẩm, hệ thống có cần kiểm tra số lượng tồn kho trước trong giỏ hàng, hệ thống sẽ xử lý như thế nào?	Trần Gia Phú	11/26/2024	Tùy chọn PA1: Nếu sản phẩm đã tồn tại trong giỏ hàng thì không cho phép xóa và hiển thị thông báo lỗi. - PA1: Không cho phép xóa và hiển thị thông báo lỗi. - PA2: Xóa và không cần kiểm tra số lượng tồn kho.	11/27/2024	Open
Mua hàng							
5	Module cart	Trong quy trình thêm sản phẩm mặc định vào giỏ hàng, có cần kiểm tra số lượng tồn kho trước khi thêm không? Nếu số lượng trong kho không đủ, đề xuất hiển thị thông báo lỗi. Mong xác nhận.	Phùng Tân Nhật	11/26/2024	Có, cần kiểm tra tồn kho trước khi thêm và hiển thị thông báo nếu số lượng không đủ.	11/27/2024	Open
6	Module cart	Khi người dùng cập nhật số lượng sản phẩm trong giỏ hàng, nếu vượt quá số lượng tồn kho, đề xuất:	Phùng Tân Nhật	11/26/2024	Tùy chọn PA1: Nếu số lượng vượt tồn kho, hủy hiển thị thông báo lỗi và không cập nhật số lượng cho người dùng. - PA1: Hủy hiển thị thông báo lỗi và không cập nhật số lượng. - PA2: Tự động cập nhật số lượng tối đa có thể thêm. Mong chọn phương án.	11/27/2024	Open
7	Module checkout	Trong trường hợp thanh toán thất bại, đề xuất:	Trần Gia Phú	11/26/2024	Tùy chọn PA1: Hệ thống nên giữ lại giỏ hàng để tái có thể thanh toán lại. - PA1: Giữ lại giỏ hàng cho người dùng để thanh toán lại. - PA2: Xóa giỏ hàng nhưng hiển thị thông báo lỗi. Mong chọn phương án.	11/27/2024	Open
8	Module checkout	Trong trường hợp thanh toán thành công, hệ thống có cần xóa giỏ hàng của khách hàng không?	Trần Gia Phú	11/26/2024	Có, hệ thống sẽ xóa giỏ hàng của người mua khi thanh toán thành công	11/27/2024	Open

Figure 54. Q&A

CHAPTER 5: TEST RESULTS

1. Test Report

Test Report is a document that summarizes the results and findings from a software testing process. It is created to provide detailed information about the testing activities performed, the errors or issues found, and an overall assessment of the quality of the software.

Purpose of Test Report:

- Provide information about the state of the software: Reporting helps stakeholders (such as developers, managers, customers) understand the current state of the software, remaining bugs, and the level of meeting requirements.
- Release Decision Support: Test reports provide a basis for deciding whether the software is ready to be released.
- Documentation: Test reports help to keep a history of testing, which helps in future reference and comparison when needed.

Project Name	Quản lý trang web bán bánh	Creator	Phùng Tấn Nhựt, Huỳnh Anh Tuấn, Trần Gia Phú
Project Code	WBMS-001	Reviewer/Approver	Phùng Tấn Nhựt, Huỳnh Anh Tuấn, Trần Gia Phú
Document Code	WebCakeStore_Test Report_v1.0	Issue Date	1-Dec-24
Notes	Bao gồm 6 module: module product_model, module cart, module Checkout, module history_order, module order_model và 1 module: module Login kiểm tra phi chức năng		

No	Function Code	Pass	Fail	Untested	N/A	Number of test cases
1	Function addProduct()	7	0	0	0	7
2	Function updateProduct()	15	0	0	0	15
3	Function deleteProduct()	2	0	0	0	2
4	Function addToCart()	5	0	0	0	5
5	Function updateCartQuantity()	3	0	0	0	3
6	Function deleteFromCart()	2	0	0	0	2
7	Function place_order()	5	0	0	0	5
Sub total		39	0	0	0	39
Test coverage						50%
Test successful coverage						50%

Figure 55. TestReport of TestCases (UnitTest)

No	Module Code	Pass	Fail	Untested	N/A	Number of test cases
1	Module_product_add	16	0	0	0	16
2	Module_product_update	22	0	0	0	22
3	Module_product_delete	2	0	0	0	2
4	Module_cart	10	0	0	0	10
5	Module_checkout	10	0	0	0	10
6	Phi công năng	2	2	0	0	4
Sub total		62	2	0	0	64
Test coverage		84%				
Test successful coverage		83%				

Figure 56. TestReport of TestCases (white box and black box)

2. Bug Report

Error Report 1: Missing OTP Authentication

Title: Missing OTP authentication when logging in

Description:

When logging into the system, the OTP authentication step is missing. Users can log in without entering the OTP code, which reduces the security of the system.

Steps to follow :

1. Go to the system login page.
2. Enter a valid login name in the "Login Name" field.
3. Enter a valid password in the "Password" field.
4. Click the "Login" button.

Expected result:

The system should prompt the user to enter the OTP code to complete the login. Only if the OTP code is valid, the user can log in successfully.

Actual result:

The system does not display the OTP authentication step. The user can log in without entering the OTP code.

Environment:

- Test date: 11/26/2024
- Browser: [Google Chrome, Microsoft Edge]
- Operating System: [Windows]

Renewable: Always Happens

Severity: Severe

Priority: 1 (High)

Solution: New

Error Report 2: Missing HTTPS encryption

Title: HTTPS encryption not required by the system

Description:

The system does not require the use of HTTPS connections, allowing users to access through unencrypted HTTP connections. This poses a great risk as sensitive information can be stolen while being transmitted over the network.

Steps to follow :

1. Open your browser and enter the system URL
2. Observe the protocol displayed in the browser's address bar.
3. Try accessing the system using HTTP instead of HTTPS.

Expected result:

All connections to the system must be secured using HTTPS. If users access via HTTP, they must be automatically redirected to HTTPS.

Actual result:

The system allows access over HTTP without automatically redirecting to HTTPS, leaving the connection unencrypted.

Environment:

- Test date: 11/28/2024
- Browser: [Google Chrome, Microsoft Edge]]
- Operating System: [Windows]

Renewable: Always Happens

Severity: High

Priority Level: 2 (Medium)

Solution: New

3. DefectList

Defect ID	Mô tả Lỗi & Các bước tái tạo	Kết quả Thực tế	Kết quả Mong đợi	Uu tiên	Mức độ nghiêm trọng	Testcase ID
#1	Quản lý sản phẩm Cho phép cập nhật sản phẩm đang có trong giỏ hàng của người dùng 1. Truy cập trang "Sản phẩm" 2. Tìm và chọn một sản phẩm đang tồn tại trong giỏ hàng của người dùng 3. Click "Sửa" và nhập thông tin cần chỉnh sửa 4. Click "Cập nhật" và xác nhận để kiểm tra kết quả khi thao tác	Hệ thống vẫn cho phép cập nhật mới thông tin sản phẩm, thông tin cũ của sản phẩm vẫn hiển thị trong giỏ hàng của người dùng	Hệ thống NÊN kiểm tra trước khi cập nhật, nếu sản phẩm đang tồn tại trong giỏ hàng, hiển thị thông báo: "Không thể chỉnh sửa vì sản phẩm đang được sử dụng."	High	Critical	TC_Update_01
#2	Cho phép cập nhật sản phẩm đang có trong giỏ hàng của người dùng 1. Truy cập trang "Sản phẩm" 2. Tìm và chọn một sản phẩm đang tồn tại trong giỏ hàng của người dùng 3. Click "Xóa" 4. Click "Ok" để xác nhận và kiểm tra kết quả khi thao tác	Hệ thống vẫn cho phép xóa một sản phẩm, sản phẩm đó vẫn tồn tại trong giỏ hàng của người dùng	Hệ thống KHÔNG cho phép xóa sản phẩm, nếu sản phẩm đang tồn tại trong giỏ hàng. Thông báo: "Không thể xóa sản phẩm vì sản phẩm đang có trong giỏ hàng của người dùng"	High	Critical	TC_Delete_01

Figure 57. DefectList

4. Auto Test

Automated testing is the process of using tools or scripts to automate software testing to ensure product quality and performance. Instead of manual testing, automated testing helps to execute test cases quickly, accurately, and repeatably, thereby saving time and reducing human errors.

Automated testing is commonly used in software development phases, especially with testing types such as functional, regression, and performance. A popular tool used is Selenium.

Steps to perform automated testing using Selenium for product functionality

Step 1: Libraries used

```
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.chrome.options import Options
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.support.ui import Select
from webdriver_manager.chrome import ChromeDriverManager
import time
import openpyxl
```

Step 2: Create a function to write logs to excel

```
def log_to_excel(level, message):
    file_path = "log.xlsx"
    try:
        # Tạo hoặc mở file Excel
        workbook = openpyxl.load_workbook(file_path)
    except FileNotFoundError:
        workbook = openpyxl.Workbook()
        sheet = workbook.active
        sheet.append(["Level", "Message", "Timestamp"])

    sheet = workbook.active
    timestamp = time.strftime("%Y-%m-%d %H:%M:%S", time.gmtime())
    sheet.append([level, message, timestamp])
    workbook.save(file_path)
```

Step 3: Create a function to check if the product exists or not (browse the pages)

```
def check_product_exists(driver, product_name):
    # Duyệt qua các trang từ 1 đến 5
    for page in range(1, 6):
        driver.get(f"http://localhost/admin/html/product_list.php?page={page}")
        WebDriverWait(driver, 10).until(EC.presence_of_element_located((By.XPATH, "//table[contains(@class, 'table')]")))

        # Kiểm tra nếu tên sản phẩm đã có trong danh sách
        try:
            products = driver.find_elements(By.XPATH, "//table[contains(@class, 'table')]/tr/td[2]")
            for product in products:
                if product_name.lower() in product.text.lower():
                    log_to_excel("INFO", f"Sản phẩm '{product_name}' đã tồn tại trên trang {page}.")
                    return True
        except Exception as e:
            print(f"Error checking products on page {page}: {e}")

    return False
```

Step 4: Create a function to add products

```
def add_product():
    options = Options()
    options.add_argument("--no-sandbox")
    options.add_argument("--disable-dev-shm-usage")
    options.add_argument("--disable-gpu")
    driver = webdriver.Chrome(service=Service(ChromeDriverManager().install()), options=options)
    driver.maximize_window()

    try:
        driver.get("http://localhost/admin/html/product_add.php")
        log_to_excel("INFO", "Truy cập vào trang đăng nhập thành công.")

        email_input = driver.find_element(By.ID, "email")
        email_input.send_keys("tannhut2111@gmail.com")
        log_to_excel("INFO", "Đã nhập email: tannhut2111@gmail.com")

        password_input = driver.find_element(By.ID, "password")
        password_input.send_keys("12345")
        log_to_excel("INFO", "Đã nhập mật khẩu.")

        sign_in_button = driver.find_element(By.CSS_SELECTOR, "button[type='submit']")
        sign_in_button.click()
        log_to_excel("INFO", "Đã click vào nút Đăng nhập.")

        log_to_excel("INFO", "Đăng nhập thành công!")

        products_menu = driver.find_element(By.XPATH, "//a[@href='product_list.php?page=1']")
        driver.execute_script("arguments[0].scrollIntoView(true);", products_menu)
        WebDriverWait(driver, 10).until(EC.element_to_be_clickable(products_menu))
        driver.execute_script("arguments[0].click();", products_menu)
        log_to_excel("INFO", "Đã truy cập vào menu sản phẩm.")
```

```

# Nhập tên sản phẩm
product_name = "Nhựa"

# Kiểm tra xem sản phẩm đã tồn tại hay chưa
if check_product_exists(driver, product_name):
    log_to_excel("INFO", f"sản phẩm '{product_name}' đã tồn tại, không thêm mới.")
    return

add_product_button = driver.find_element(By.XPATH, "//a[@href='product_add.php' and @style='color: white']")
driver.execute_script("arguments[0].scrollIntoView(true);", add_product_button)
WebDriverWait(driver, 10).until(EC.element_to_be_clickable(add_product_button))
driver.execute_script("arguments[0].click();", add_product_button)
log_to_excel("INFO", "Đã click vào nút thêm sản phẩm.")

# Điền thông tin sản phẩm
try:
    name_input = driver.find_element(By.ID, "name")
    name_input.send_keys(product_name)
    log_to_excel("INFO", f"Đã nhập tên sản phẩm: {product_name}.")
except Exception as e:
    log_to_excel("ERROR", f"Lỗi khi nhập tên sản phẩm: {e}")

# Chọn thương hiệu
try:
    brand_select = driver.find_element(By.NAME, "id_brands")
    select = Select(brand_select)
    select.select_by_value("96") # Thay thế bằng giá trị đúng
    product_info_brand = select.first_selected_option.text
    log_to_excel("INFO", f"Đã chọn thương hiệu: {product_info_brand}")
except Exception as e:
    log_to_excel("ERROR", f"Lỗi khi chọn thương hiệu: {e}")

```

```

# Chọn danh mục
try:
    category_select = driver.find_element(By.NAME, "id_categories")
    select = Select(category_select)
    select.select_by_value("2") # Thay thế bằng giá trị đúng
    product_info_category = select.first_selected_option.text
    log_to_excel("INFO", f"Đã chọn danh mục: {product_info_category}")
except Exception as e:
    log_to_excel("ERROR", f"Lỗi khi chọn danh mục: {e}")

# Chọn hình ảnh
try:
    image_input = driver.find_element(By.ID, "image")
    image_input.send_keys("C:/xampp/htdocs/Cake-Sale-Website/admin/uploads/1aba114354.product-big-4.jpg")
    log_to_excel("INFO", "Đã chọn hình ảnh sản phẩm.")
except Exception as e:
    log_to_excel("ERROR", f"Lỗi khi chọn hình ảnh sản phẩm: {e}")

# Nhập số lượng nhập
try:
    quantity_in_input = driver.find_element(By.ID, "quantity")
    quantity_in_input.send_keys("100")
    log_to_excel("INFO", "Đã nhập số lượng nhập sản phẩm.")
except Exception as e:
    log_to_excel("ERROR", f"Lỗi khi điền số lượng nhập sản phẩm: {e}")

# Nhập số lượng bán
try:
    quantity_sold_input = driver.find_element(By.ID, "avaivable_quantity")
    quantity_sold_input.send_keys("80")
    log_to_excel("INFO", "Đã nhập số lượng bán sản phẩm.")
except Exception as e:
    log_to_excel("ERROR", f"Lỗi khi điền số lượng bán sản phẩm: {e}")

```

```

# Nhập giá nhập
try:
    buy_price_input = driver.find_element(By.ID, "buy_price")
    buy_price_input.send_keys("100000")
    log_to_excel("INFO", "Đã nhập giá nhập sản phẩm.")
except Exception as e:
    log_to_excel("ERROR", f'Lỗi khi điền giá nhập sản phẩm: {e}')

# Nhập giá bán
try:
    sell_price_input = driver.find_element(By.ID, "sell_price")
    sell_price_input.send_keys("120000") # Nhập giá bán
    log_to_excel("INFO", "Đã nhập giá bán sản phẩm.")
except Exception as e:
    log_to_excel("ERROR", f'Lỗi khi điền giá bán sản phẩm: {e}')

# Nhập mô tả
try:
    description_input = driver.find_element(By.ID, "description")
    description_input.send_keys("Bánh kem ngon.")
    log_to_excel("INFO", "Đã nhập mô tả sản phẩm.")
except Exception as e:
    log_to_excel("ERROR", f'Lỗi khi nhập mô tả sản phẩm: {e}')

# Nhấn Submit
submit_button = driver.find_element(By.NAME, "submit")
driver.execute_script("arguments[0].click();", submit_button)
log_to_excel("INFO", "Đã nhấn nút thêm sản phẩm.")

# Kiểm tra nếu sản phẩm được thêm thành công
WebDriverWait(driver, 10).until(EC.url_contains("product_list.php"))
log_to_excel("INFO", "Sản phẩm đã được thêm thành công!")

except Exception as e:
    log_to_excel("ERROR", f'Có lỗi xảy ra: {e}')

```

Step 5: Execute

```

if __name__ == "__main__":
    add_product()

```

CHAPTER 6. CONCLUSION

Software testing plays an extremely important role in ensuring the quality and stability of the system. When users make requests for a bakery website, the system designer needs to carefully analyze the requirements to build a suitable software architecture. The testing process will begin with writing test cases for each part of the system, including unit testing, integration testing, and system testing. Each of these types of testing helps detect errors at different levels, from the level of detail of each small part of the source code to the entire system. The results from this testing process will provide a clear view of the performance, security, and stability of the website, while ensuring that the system's functions meet user requirements, thereby improving the quality and reliability of the final product.