A black background with blue and purple text

Description automatically generated

**BÁO CÁO CUỐI KÌ**

**HỌC PHẦN: KIỂM THỬ PHẦN MỀM**

**ĐỒ ÁN: ỨNG DỤNG KIỂM THỬ VÀO TRANG WEBSITE BÁN BÁNH**

**Giảng viên hướng dẫn: Đỗ Như Tài**

**Họ và tên thành viên:**

+ Huỳnh Anh Tuấn MSSV: 3121411220

+ Trần Gia Phú MSSV: 3121411166

+ Phùng Tấn Nhựt MSSV: 3121411159

**Năm học: 2024 - 2025**

**MỤC LỤC**

[CHƯƠNG 1: TỔNG QUAN VỀ KIỂM THỬ WEBSITE TIỆM BÁNH 7](#_Toc185516822)

[1. Giới thiệu 7](#_Toc185516823)

[2. Ngữ cảnh nghiệp vụ 8](#_Toc185516824)

[3. Quy trình nghiệp vụ 8](#_Toc185516825)

[3.1. Quy trình nhập hàng 8](#_Toc185516826)

[3.2. Quy trình mua hàng 9](#_Toc185516827)

[4. Các phân hệ giải quyết bài toán 11](#_Toc185516828)

[4.1. Quy trình nhập hàng 11](#_Toc185516829)

[4.2. Quy trình mua hàng 12](#_Toc185516830)

[5. Conceptual Model 13](#_Toc185516831)

[6. User Stories 14](#_Toc185516832)

[7. Yêu cầu nghiệp vụ 15](#_Toc185516833)

[7.1. Yêu cầu chức năng 15](#_Toc185516834)

[7.2. Yêu cầu phi chức năng 16](#_Toc185516835)

[8. Các tác nhân tham gia hệ thống 17](#_Toc185516836)

[CHƯƠNG 2: ĐẶC TẢ YÊU CẦU PHẦN MỀM 18](#_Toc185516837)

[1. Sơ đồ triển khai 18](#_Toc185516838)

[1.1. Nhập hàng 18](#_Toc185516839)

[1.2. Mua hàng 18](#_Toc185516840)

[2. Sơ đồ tổng quan các chức năng của hệ thống 19](#_Toc185516841)

[3. Sơ đồ cơ sở dữ liệu tổng quát 20](#_Toc185516842)

[4. Nhập hàng 21](#_Toc185516843)

[4.1. Focused UseCase 21](#_Toc185516844)

[4.2. Sơ đồ Activity 24](#_Toc185516845)

[4.3. Sơ đồ Sequence 25](#_Toc185516846)

[4.4. Sơ đồ Class 26](#_Toc185516847)

[4.5. Giao diện minh họa 26](#_Toc185516848)

[5. Mua hàng 29](#_Toc185516849)

[5.1. Focused UseCase 29](#_Toc185516850)

[5.2. Sơ đồ Activity 32](#_Toc185516851)

[5.3. Sơ đồ Sequence 33](#_Toc185516852)

[5.4. Sơ đồ Class 34](#_Toc185516853)

[5.5. Giao diện minh họa 35](#_Toc185516854)

[CHƯƠNG 3: KẾ HOẠCH KIỂM THỬ 38](#_Toc185516855)

[1. Giới thiệu 38](#_Toc185516856)

[1.1. Mục đích 38](#_Toc185516857)

[1.2. Bối cảnh 38](#_Toc185516858)

[1.3. Phạm vi 38](#_Toc185516859)

[1.4. Nhận diện dự án 38](#_Toc185516860)

[1.5 Tài liệu tham khảo 39](#_Toc185516861)

[2. Các tính năng cần kiểm thử 40](#_Toc185516862)

[2.1. Chức năng (Khách hàng) 40](#_Toc185516863)

[2.1.1. Thêm sản phẩm vào giỏ hàng 40](#_Toc185516864)

[2.1.2. Xóa sản phẩm khỏi giỏ hàng 40](#_Toc185516865)

[2.1.3. Cập nhật sản phẩm trong giỏ hàng 40](#_Toc185516866)

[2.1.4. Thanh toán giỏ hàng 40](#_Toc185516867)

[2.2. Chức năng (Người quản lý) 41](#_Toc185516868)

[2.2.1. Thêm sản phẩm vào kho 41](#_Toc185516869)

[2.2.2. Xóa sản phẩm khỏi kho 41](#_Toc185516870)

[2.2.3. Cập nhật sản phẩm trong kho 41](#_Toc185516871)

[2.3. Tính khả dụng 41](#_Toc185516872)

[2.4. Các giới hạn thiết kế 41](#_Toc185516873)

[2.5. Giao diện 42](#_Toc185516874)

[2.5.1. Giao diện người dùng 42](#_Toc185516875)

[3. Các tính năng không cần kiểm thử 42](#_Toc185516876)

[3.1. Chức năng (Khách hàng) 42](#_Toc185516877)

[3.2. Chức năng (Người quản lý) 43](#_Toc185516878)

[4. Chiến lược kiểm thử 46](#_Toc185516879)

[4.1. Các loại kiểm thử 46](#_Toc185516880)

[4.2. Công cụ 46](#_Toc185516881)

[5. Nguồn lực: 46](#_Toc185516882)

[6. Các mốc của dự án 47](#_Toc185516883)

[7. Các sản phẩm đầu ra 47](#_Toc185516884)

[7.1 Mô hình kiểm thử 47](#_Toc185516885)

[7.2 Nhật ký kiểm thử 47](#_Toc185516886)

[CHƯƠNG 4: PHƯƠNG PHÁP KIỂM THỬ 48](#_Toc185516887)

[1. Unit Testing 48](#_Toc185516888)

[1.1. TestPlan 49](#_Toc185516889)

[1.2. Quy trình nhập hàng 52](#_Toc185516890)

[1.3. Quy trình mua hàng 55](#_Toc185516891)

[2. Integration Testing 57](#_Toc185516892)

[2.1. Test Design 57](#_Toc185516893)

[2.2. Kiểm thử hộp trắng 61](#_Toc185516894)

[2.2.1. Quy trình nhập hàng 63](#_Toc185516895)

[2.2.2. Quy trình mua hàng 74](#_Toc185516896)

[2.3. Kiểm thử hộp đen 84](#_Toc185516897)

[2.3.1. Quy trình nhập hàng 86](#_Toc185516898)

[2.3.2. Quy trình mua hàng 93](#_Toc185516899)

[3. System Testing 98](#_Toc185516900)

[3.1. TestCase List 99](#_Toc185516901)

[3.2. Quy trình nhập hàng 101](#_Toc185516902)

[3.3. Quy trình mua hàng 104](#_Toc185516903)

[3.4. Phi chức năng 106](#_Toc185516904)

[4. Acceptance Testing 106](#_Toc185516905)

[CHƯƠNG 5: KẾT QUẢ KIỂM THỬ 108](#_Toc185516906)

[1. Test Report 108](#_Toc185516907)

[2. Bug Report 109](#_Toc185516908)

[3. DefectList 110](#_Toc185516909)

[4. Auto Test 110](#_Toc185516910)

[CHƯƠNG 6. KẾT LUẬN 115](#_Toc185516911)

**DANH SÁCH DANH MỤC, HÌNH ẢNH**

[Hình 1. Sơ đồ quy trình nhập hàng 10](#_Toc185516912)

[Hình 2. Sơ đồ quy trình mua hàng 11](#_Toc185516913)

[Hình 3. Mô hình khái niệm 12](#_Toc185516914)

[Hình 4. Tác nhân tham gia hệ thống 16](#_Toc185516915)

[Hình 5. Sơ đồ triển khai Nhập hàng 17](#_Toc185516916)

[Hình 6. Sơ đồ triển khai Mua hàng 17](#_Toc185516917)

[Hình 7. UseCase Tổng quát 18](#_Toc185516918)

[Hình 8. Sơ đồ Database 19](#_Toc185516919)

[Hình 9. Sơ đồ Activity quy trình Nhập hàng 23](#_Toc185516920)

[Hình 10. Sơ đồ Sequence quy trình Nhập hàng 24](#_Toc185516921)

[Hình 11. Sơ đồ Class quy trình Nhập hàng 25](#_Toc185516922)

[Hình 12. Giao diện xóa sản phẩm Nhập hàng 25](#_Toc185516923)

[Hình 13. Giao diện thêm sản phẩm Nhập hàng 26](#_Toc185516924)

[Hình 14. Giao diện cập nhật sản phẩm Nhập hàng 27](#_Toc185516925)

[Hình 15. Sơ đồ Activity quy trình Mua hàng 31](#_Toc185516926)

[Hình 16. Sơ đồ Sequence quy trình Mua hàng 32](#_Toc185516927)

[Hình 17. Sơ đồ Class quy trình Mua hàng 33](#_Toc185516928)

[Hình 18. Giao diện thêm mặc định một sản phẩm vào giỏ hàng khi Mua hàng 34](#_Toc185516929)

[Hình 19. Giao diện thêm sản phẩm với số lượng tùy chọn khi Mua hàng 35](#_Toc185516930)

[Hình 20. Giao diện xem và cập nhật giỏ hàng 35](#_Toc185516931)

[Hình 21. Giao diện xóa sản phẩm khỏi giỏ hàng 36](#_Toc185516932)

[Hình 22. Giao diện thanh toán giỏ hàng 36](#_Toc185516933)

[Hình 23. Bảng Kế hoạch kiểm thử TestPlan 48](#_Toc185516934)

[Hình 24. Unit Test Nhập hàng 51](#_Toc185516935)

[Hình 25. Unit Test Mua hàng 54](#_Toc185516936)

[Hình 26. Test Design Nhập hàng 57](#_Toc185516937)

[Hình 27. Test Design Mua hàng 58](#_Toc185516938)

[Hình 28. CheckList Hộp Trắng 60](#_Toc185516939)

[Hình 29. Sơ đồ dòng điều khiển Thêm sản phẩm Nhập hàng 62](#_Toc185516940)

[Hình 30. Sơ đồ dòng điều khiển Cập nhật sản phẩm 66](#_Toc185516941)

[Hình 31. Sơ đồ dòng điều khiểm Xóa sản phẩm Nhập hàng 71](#_Toc185516942)

[Hình 32. Sơ đồ dòng điều khiểm Thêm sản phẩm Giỏ hàng 73](#_Toc185516943)

[Hình 33. Sơ đồ dòng điều khiển Cập nhật Giỏ hàng 76](#_Toc185516944)

[Hình 34. Sơ đồ dòng điều khiển Xóa sản phẩm Giỏ hàng 78](#_Toc185516945)

[Hình 35. Sơ đồ dòng điều khiển Thanh toán Giỏ hàng 80](#_Toc185516946)

[Hình 36.CheckList Hộp Đen 83](#_Toc185516947)

[Hình 37. Bảng quyết định Thêm sản phẩm 85](#_Toc185516948)

[Hình 38. Bảng quyết định Cập nhật sản phẩm 88](#_Toc185516949)

[Hình 39. Bảng quyết định Xóa sản phẩm 91](#_Toc185516950)

[Hình 40. Bảng quyết định Thêm sản phẩm vào Giỏ hàng 92](#_Toc185516951)

[Hình 41. Bảng quyết định Cập nhật số lượng trong Giỏ hàng 93](#_Toc185516952)

[Hình 42. Bảng quyết định Xóa sản phẩm khỏi Giỏ hàng 94](#_Toc185516953)

[Hình 43. Bảng quyết định Thanh toán Giỏ hàng 95](#_Toc185516954)

[Hình 44. Danh sách test case 98](#_Toc185516955)

[Hình 45. Module và kết quả kiểm thử cho quy trình nhập hàng 100](#_Toc185516956)

[Hình 46.DaTa Test cho quy trình nhập hàng 100](#_Toc185516957)

[Hình 47. TestCase quy trình nhập hàng 102](#_Toc185516958)

[Hình 48. Module và kết quả kiểm thử cho quy trình mua hàng 103](#_Toc185516959)

[Hình 49. Data Test cho quy trình mua hàng 103](#_Toc185516960)

[Hình 50. TestCae quy trình mua hàng 104](#_Toc185516961)

[Hình 51. Kết quả kiểm thử cho phi chức năng 105](#_Toc185516962)

[Hình 52. Data Test cho phi chức năng 105](#_Toc185516963)

[Hình 53. TestCase phi chức năng 105](#_Toc185516964)

[Hình 54. Q&A 106](#_Toc185516965)

[Hình 55. TestReport các TestCase (UnitTest 107](#_Toc185516966)

[Hình 56. TestReport các TestCase (hộp trắng và hộp đen) 107](#_Toc185516967)

[Hình 57. DefectList 109](#_Toc185516968)

# CHƯƠNG 1: TỔNG QUAN VỀ KIỂM THỬ WEBSITE TIỆM BÁNH

## 1. Giới thiệu

Kiểm thử phần mềm là một bước quan trọng trong quá trình phát triển Website Tiệm Bánh, đảm bảo rằng hệ thống hoạt động chính xác, ổn định, và mang lại trải nghiệm tốt nhất cho người dùng. Website Tiệm Bánh được thiết kế với nhiều chức năng như quản lý sản phẩm, nhập hàng, và mua hàng trực tuyến, đáp ứng nhu cầu vận hành của quản trị viên và khách hàng.

Kiểm thử phần mềm Website Tiệm Bánh tập trung vào việc xác minh toàn bộ các tính năng của hệ thống, phát hiện lỗi, đảm bảo sự tương thích với các môi trường khác nhau, và đáp ứng đầy đủ các yêu cầu nghiệp vụ.

**Mục tiêu của kiểm thử**

* **Đảm bảo tính chính xác:** Kiểm tra các tính năng như thêm sản phẩm, cập nhật giỏ hàng, thanh toán để đảm bảo hoạt động đúng như yêu cầu.
* **Phát hiện và sửa lỗi:** Xác định các vấn đề tiềm ẩn trong hệ thống trước khi đưa vào vận hành chính thức.
* **Tăng trải nghiệm người dùng:** Đảm bảo giao diện dễ sử dụng và các chức năng hoạt động mượt mà trên nhiều thiết bị.
* **Đánh giá hiệu năng:** Kiểm tra khả năng xử lý đồng thời và đáp ứng của hệ thống trong điều kiện tải cao.

**Tầm quan trọng của kiểm thử Website Tiệm Bánh**

* Kiểm thử giúp đảm bảo rằng Website Tiệm Bánh vận hành trơn tru, không xảy ra lỗi trong quá trình sử dụng.
* Đảm bảo thông tin sản phẩm, đơn hàng, và dữ liệu khách hàng luôn được lưu trữ chính xác.
* Tạo sự tin cậy và hài lòng cho cả quản trị viên và khách hàng.
* Giảm thiểu rủi ro kinh doanh bằng cách ngăn chặn các lỗi nghiêm trọng trước khi hệ thống được triển khai chính thức.

## 2. Ngữ cảnh nghiệp vụ

**Website Tiệm Bánh** có các tính năng cơ bản cho **Nhập Hàng, Mua Hàng**

Với **Nhập Hàng**, quản trị viên có thể thêm, sửa, xóa các sản phẩm trong hệ thống. Sau khi thêm hoặc chỉnh sửa sản phẩm, hệ thống sẽ tự động cập nhật danh sách sản phẩm với các thông tin như tên sản phẩm, mô tả, giá bán, số lượng tồn kho, và hình ảnh. Khi cần thiết, quản trị viên có thể xóa các sản phẩm không còn kinh doanh. Các thay đổi sẽ được tự động lưu và cập nhật trong hệ thống, giúp đảm bảo danh sách sản phẩm luôn chính xác và đầy đủ.

Với **Mua Hàng**, khách hàng có thể thêm các loại bánh vào giỏ hàng từ danh sách sản phẩm hoặc từ trang chi tiết sản phẩm. Sau khi thêm sản phẩm vào giỏ hàng, khách hàng có thể xem giỏ hàng được cập nhật với các thông tin như tổng chi phí bánh và tổng số tiền cần thanh toán. Khách hàng có thể điều chỉnh giỏ hàng bằng cách cập nhật số lượng hoặc xóa sản phẩm. Khi sẵn sàng, khách hàng có thể tiến hành thanh toán để hoàn tất mua hàng.

Một số công việc thiết lập ban đầu cần hoàn thành khi Website Tiệm Bánh được khởi chạy, bao gồm tạo tài khoản quản trị viên, hai tài khoản khách hàng và dữ liệu mẫu cho các sản phẩm bánh, kho hàng và đánh giá cho một vài loại bánh.

## 3. Quy trình nghiệp vụ

### 3.1. Quy trình nhập hàng

**Thêm sản phẩm:**

* **Người quản lý** nhập thông tin sản phẩm (tên, giá, hình ảnh, mô tả, số lượng trong kho, danh mục, thương hiệu).
* Thông tin được kiểm tra tính hợp lệ.
* Sản phẩm được thêm vào cơ sở dữ liệu và hiển thị trong danh sách sản phẩm.

**Cập nhật sản phẩm:**

* **Người quản lý** chọn sản phẩm cần chỉnh sửa.
* Thực hiện cập nhật thông tin (tên, giá, hình ảnh, số lượng trong kho...).
* Cập nhật thay đổi vào cơ sở dữ liệu và hiển thị thông tin mới trên giao diện.

**Xóa sản phẩm:**

* **Người quản lý** chọn sản phẩm cần xóa.
* Hệ thống kiểm tra xem sản phẩm có nằm trong đơn hàng chưa xử lý không:
  + Nếu có: Không thể xóa.
  + Nếu không: Xóa sản phẩm khỏi cơ sở dữ liệu.

### 3.2. Quy trình mua hàng

**Thêm sản phẩm vào giỏ hàng:**

* **Người dùng** chọn sản phẩm và số lượng cần mua.
* Sản phẩm được thêm vào giỏ hàng:
  + Nếu sản phẩm đã có trong giỏ hàng: Cập nhật số lượng.
  + Nếu chưa: Thêm sản phẩm vào danh sách giỏ hàng.
* Hiển thị danh sách sản phẩm trong giỏ hàng kèm tổng số lượng
* Hệ thống tự động tính tổng tiền dựa trên số lượng và giá của các sản phẩm trong giỏ hàng.

**Cập nhật số lượng sản phẩm trong giỏ hàng:**

* **Người dùng** điều chỉnh số lượng sản phẩm trong giỏ.
* Hệ thống kiểm tra số lượng trong kho:
  + Nếu đủ: Cập nhật giỏ hàng.
  + Nếu không: Thông báo và giữ nguyên số lượng cũ.

**Xóa sản phẩm trong giỏ hàng:**

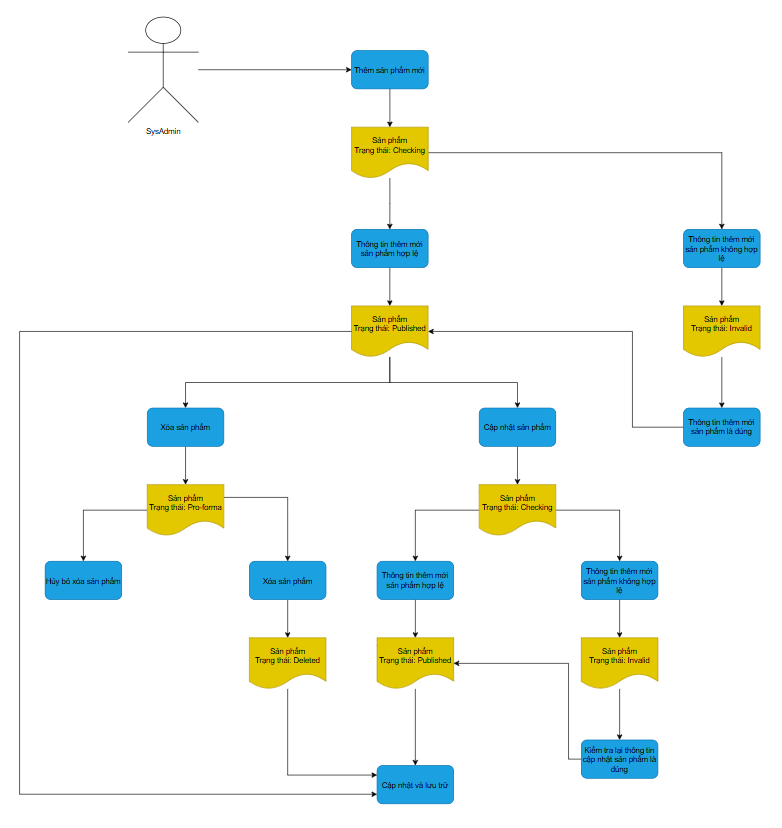
* **Người dùng** chọn sản phẩm cần xóa.
* Sản phẩm được loại bỏ khỏi giỏ hàng và tổng tiền được cập nhật.

**Thanh toán giỏ hàng:**

* **Người dùng** xác nhận đơn hàng trong giỏ.
* Hệ thống kiểm tra:
  + Thông tin giao hàng (địa chỉ, số điện thoại).
* Nếu đủ điều kiện:
  + Thanh toán thành công.
  + Hệ thống giảm số lượng sản phẩm trong kho tương ứng.
  + Tạo mã đơn hàng và lưu vào lịch sử mua hàng.
* Nếu không đủ điều kiện:
  + Thông báo lỗi và giữ nguyên giỏ hàng.

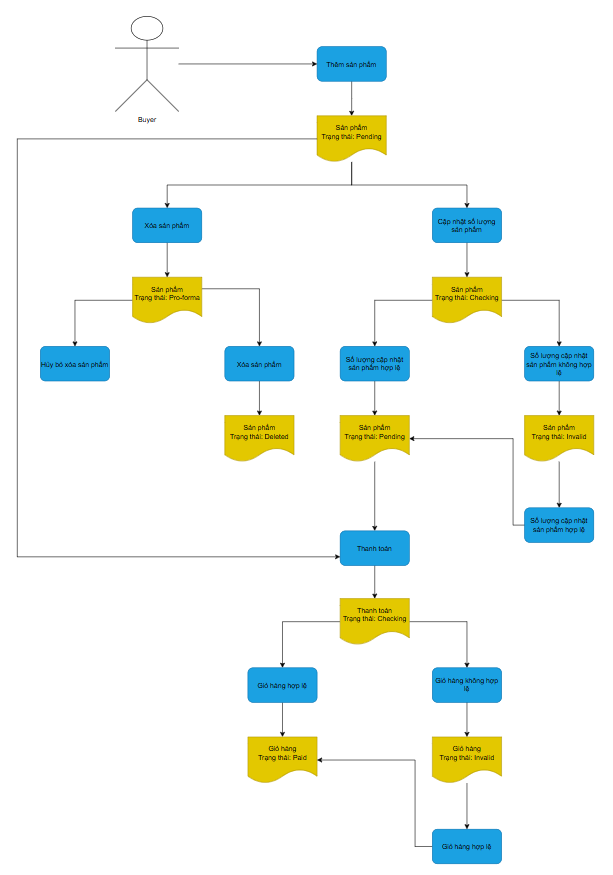
## 4. Các phân hệ giải quyết bài toán

### 4.1. Quy trình nhập hàng



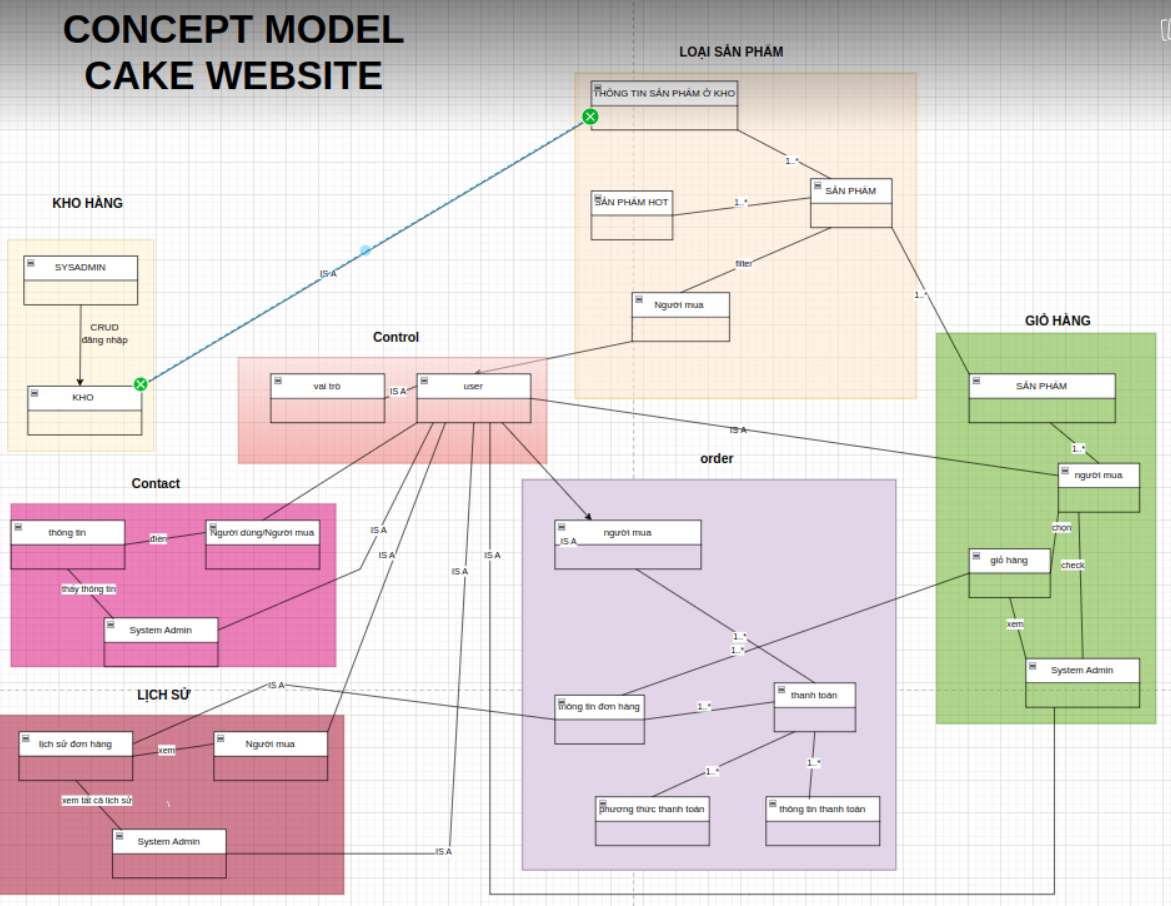
Hình 1. Sơ đồ quy trình nhập hàng

### 4.2. Quy trình mua hàng



Hình 2. Sơ đồ quy trình mua hàng

## 5. Conceptual Model



Hình 3. Mô hình khái niệm

## 6. User Stories

**Nhập hàng**

* **Là quản trị viên**, tôi muốn thêm sản phẩm vào hệ thống, bao gồm tên, danh mục, thương hiệu, giá bán, giá nhập, mô tả, hình ảnh và số lượng tồn kho.
* **Là quản trị viên**, tôi muốn xóa sản phẩm khỏi hệ thống khi không còn sản phẩm này bán hoặc không còn trong kho.
* **Là quản trị viên**, tôi muốn sửa thông tin sản phẩm, bao gồm cập nhật giá, mô tả, số lượng tồn kho, hình ảnh và các thuộc tính khác.
* **Là hệ thống**, tôi muốn kiểm tra dữ liệu đầu vào khi thêm hoặc sửa sản phẩm để đảm bảo thông tin hợp lệ.
* **Là hệ thống**, tôi muốn lưu lịch sử chỉnh sửa sản phẩm để có thể tra cứu và kiểm tra lại các thay đổi trong quá khứ.

**Mua hàng**

* **Là người mua**, tôi muốn mua bất kỳ sản phẩm nào trên trang danh mục sản phẩm.
* **Là người mua**, tôi muốn xem chi tiết sản phẩm để quyết định mua sản phẩm. Nếu tôi thích sản phẩm và số lượng còn trong kho, tôi có thể thêm sản phẩm đó vào giỏ hàng.
* **Là người mua**, tôi muốn xem danh sách các sản phẩm trong giỏ hàng và tổng chi phí giỏ hàng.
* **Là người mua**, tôi muốn cập nhật số lượng sản phẩm trong giỏ hàng. Nếu có sản phẩm hết hàng hệ thống sẽ gửi thông báo lỗi và không cho cập nhật số lượng. Khi tôi tăng hoặc giảm số lượng, tổng chi phí giỏ hàng sẽ tự động được cập nhật.
* **Là người mua**, tôi muốn xóa sản phẩm không cần thiết ra khỏi giỏ hàng. Sau khi xóa, tổng chi phí và danh sách sản phẩm sẽ được cập nhật ngay lập tức.
* **Là người mua**, tôi muốn thanh toán giỏ hàng của mình. Nếu giỏ hàng trống (số lượng sản phẩm là 0), tôi sẽ nhận được thông báo không thể thanh toán.
* **Là người mua**, tôi muốn lưu thông tin giỏ hàng để có thể quay lại mua sắm sau khi chưa thanh toán.
* **Là hệ thống**, tôi sẽ gửi thông báo xác nhận qua email cho người mua khi mua hàng thành công.

## 7. Yêu cầu nghiệp vụ

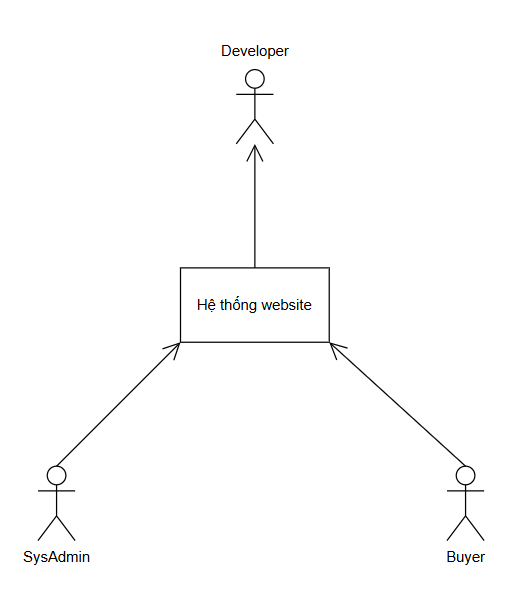
### 7.1. Yêu cầu chức năng

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã yêu cầu** | **Người yêu cầu** | **Nội dung yêu cầu** | **Mức thiết yếu** |
| **WB1** | Quản trị viên | **Nhập hàng:** Thêm, xóa, sửa sản phẩm. | High |
| **WB1.1** | Quản trị viên | Thêm sản phẩm với các thông tin: tên, danh mục, thương hiệu, giá, mô tả, số lượng, hình ảnh. | High |
| **WB1.2** | Quản trị viên | Xóa sản phẩm, chỉ thực hiện khi không có đơn hàng liên quan. | High |
| **WB1.3** | Quản trị viên | Sửa thông tin sản phẩm, lưu lịch sử thay đổi. | High |
| **WB2** | Người dùng | **Mua hàng:** Thêm sản phẩm, cập nhật số lượng, xóa sản phẩm, thanh toán | High |
| **WB2.1** | Người dùng | Thêm sản phẩm vào giỏ hàng | High |
| **WB2.2** | Người dùng | Cập nhật số lượng sản phẩm trong giỏ, kiểm tra tồn kho. | High |
| **WB2.3** | Người dùng | Xóa sản phẩm khỏi giỏ hàng. | High |
| **WB2.4** | Người dùng | Xác nhận thông tin trước thanh toán. | High |
| **WB2.5** | Người dùng | Sau thanh toán thành công, giảm số lượng sản phẩm trong kho tương ứng. | High |

### 7.2. Yêu cầu phi chức năng

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã yêu cầu** | **Người yêu cầu** | **Mô tả yêu cầu** | **Mức độ ưu tiên** |
| **WB5** | Quản trị viên | **Khả năng Xử lý Tải (Load Handling)** | High |
| **WB5.1** | Quản trị viên | Hệ thống phải có khả năng xử lý đồng thời một số lượng lớn người dùng truy cập mà không bị giảm hiệu suất. | High |
| **WB6** | Quản trị viên | **Bảo mật (Security)** | High |
| **WB6.1** | Quản trị viên | Hệ thống phải sử dụng cơ chế xác thực mạnh mẽ (ví dụ: 2FA – xác thực 2 yếu tố) khi người dùng đăng nhập và thực hiện giao dịch quan trọng. | High |
| **WB6.2** | Quản trị viên | Hệ thống phải có các biện pháp bảo vệ chống lại các cuộc tấn công như SQL Injection, XSS, CSRF, DoS/DDoS. | High |
| **WB6.3** | Quản trị viên | Tất cả các kết nối và giao dịch của người dùng phải được mã hóa và bảo vệ thông qua HTTPS. | High |

## **8. Các tác nhân tham gia hệ thống**



Hình 4. Tác nhân tham gia hệ thống

# CHƯƠNG 2: ĐẶC TẢ YÊU CẦU PHẦN MỀM

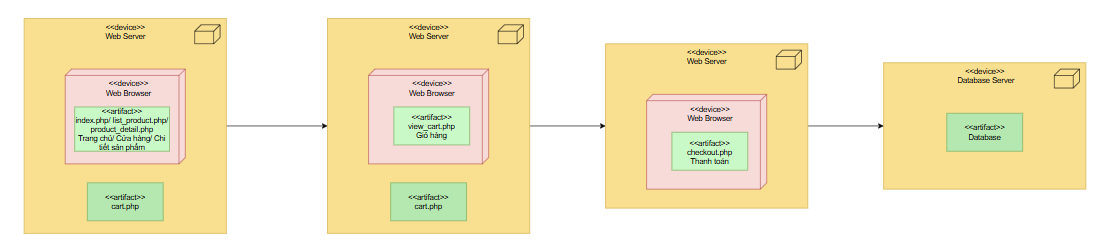
## 1. Sơ đồ triển khai

### 1.1. Nhập hàng



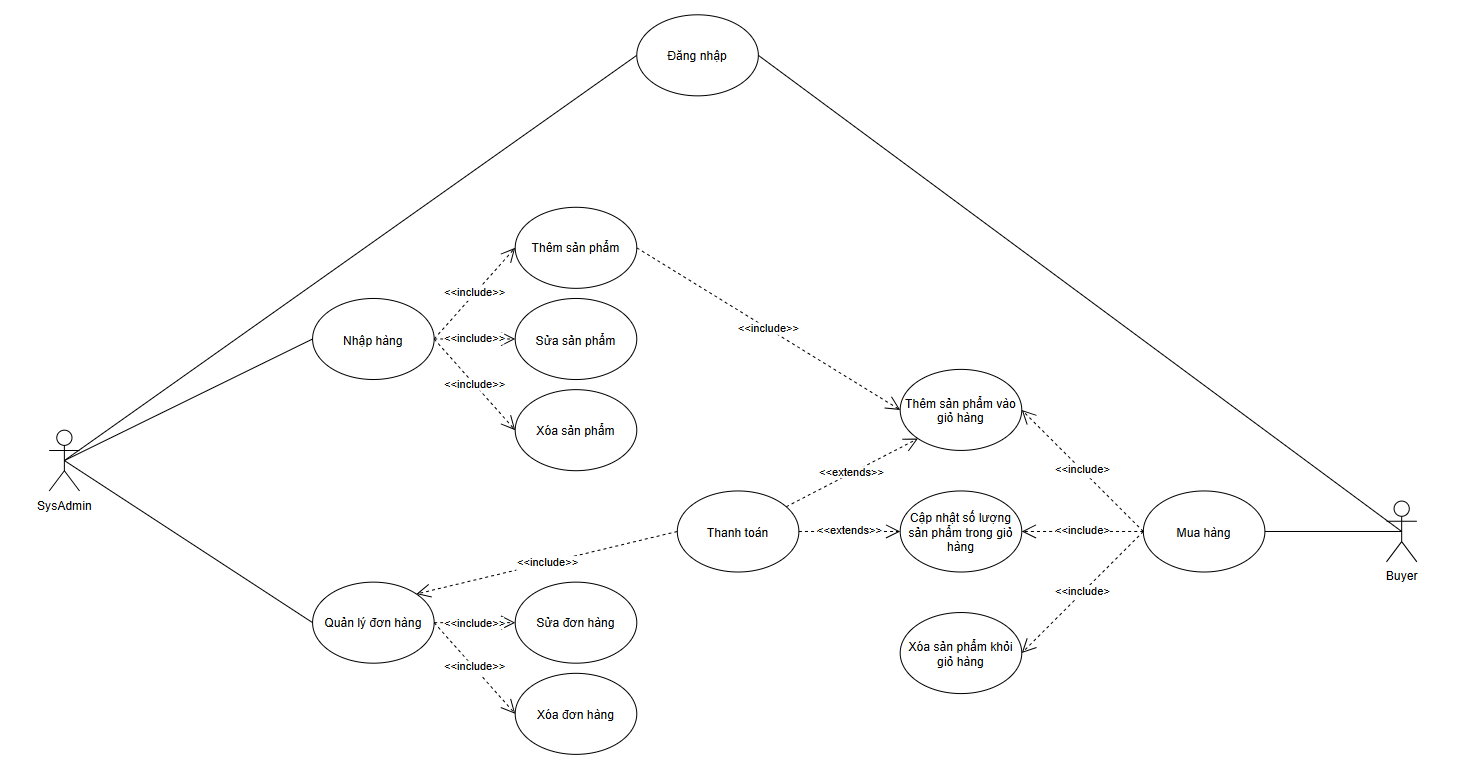
Hình 5. Sơ đồ triển khai Nhập hàng

### 1.2. Mua hàng

****

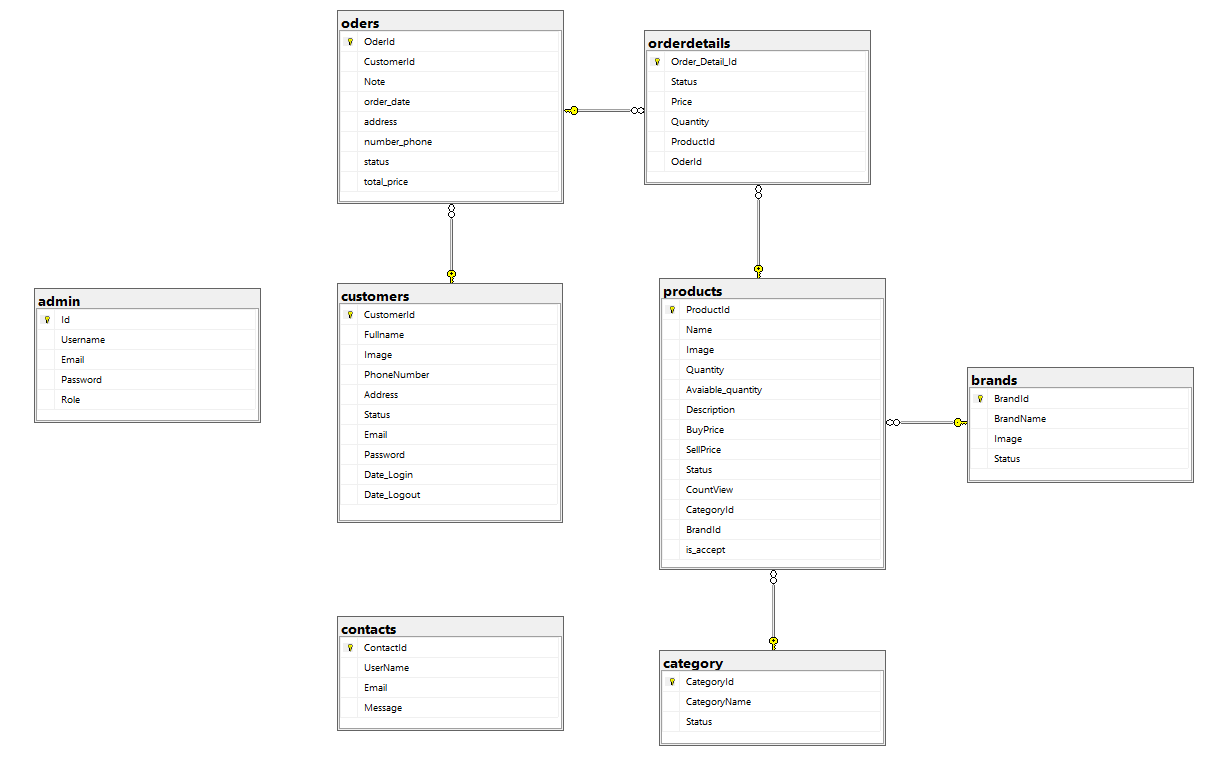
Hình 6. Sơ đồ triển khai Mua hàng

## 2. Sơ đồ tổng quan các chức năng của hệ thống



Hình 7. UseCase Tổng quát

## 3. Sơ đồ cơ sở dữ liệu tổng quát



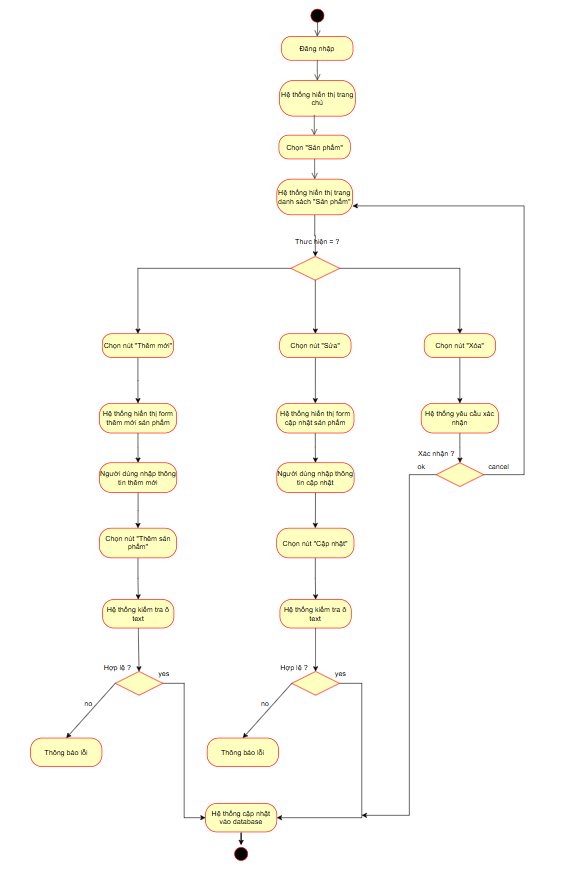
Hình 8. Sơ đồ Database

## 4. Nhập hàng

### 4.1. Focused UseCase

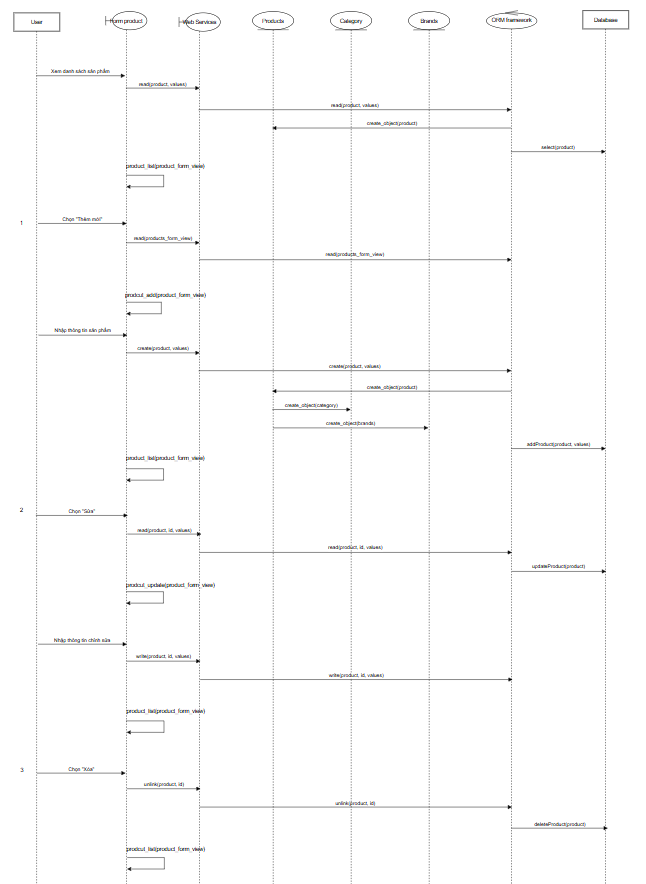
|  |  |  |
| --- | --- | --- |
|  | | |
| **Use case number** | **UC1** | |
| **Use case name** | **Nhập xuất** | |
| **Actor(s)** | **Người quản lý** | |
| **Maturity** | **Focused** | |
| **Summary** | *Use case mô tả quá trình nhập xuất hàng của người quản lý trong hệ thống* | |
| **Basic Course of events** | **Actor Action** | **System response** |
| 1. Thực thi (Xác thực đăng nhập) |  |
|  | 2. Hệ thống hiển thị trang chủ |
| 3. Use Case bắt đầu khi Actor chọn “Sản phẩm” |  |
|  | 4. Hệ thống hiển thị ra danh sách sản phẩm |
| 1.Actor bấm “Thêm mới” **A1 A2** |  |
|  | 2. Hệ thống hiển thị form thêm mới sản phẩm |
| 3. Actor nhập thông tin thêm mới |  |
|  | 4. Hệ thống thực hiện  **{Kiểm tra ô text} E1** |
|  | 5. Hệ thống cập nhật dữ liệu vào database |
| **Alternative Paths** | **A1** |  |
| **Actor Action** | **System response** |
| 1.Actor bấm “Sửa” |  |
|  | 2. Hệ thống hiển thị form cập nhật sản phẩm |
| 3. Actor nhập thông tin cập nhật |  |
|  | 4. Hệ thống thực hiện  **{Kiểm tra ô text} E1** |
|  | 5. Hệ thống cập nhật dữ liệu vào database |
| **A2** |  |
| **Actor Action** | **System response** |
| 1.Actor bấm “Xóa” |  |
|  | 2. Hệ thống yêu cầu xác nhận |
| 3. Actor chọn xác nhận “OK” **A4** |  |
|  | 6. Hệ thống cập nhật dữ liệu vào database |
| **A4** |  |
| **Actor Action** | **System response** |
| 1.Actor có thể bấm “Cancel” thao thác |  |
|  | 2.Hệ thống xóa yêu cầu xác nhận. Trở về bước 4 của Basic Course of Events. |
| **Exception Path** | **E1.** Tại **{Kiểm tra ô text}**, kiểm tra ô text có bị bỏ trống, đúng kiểu dữ liệu và ràng buộc hay không, nếu có thông báo lỗi tương ứng | |
| **Triggers:** | Khi actor muốn nhập xuất hàng | |
| **Assumption:** | None | |
| **Preconditions:** | None | |
| **Post Conditions:** | None | |

### 4.2. Sơ đồ Activity

****

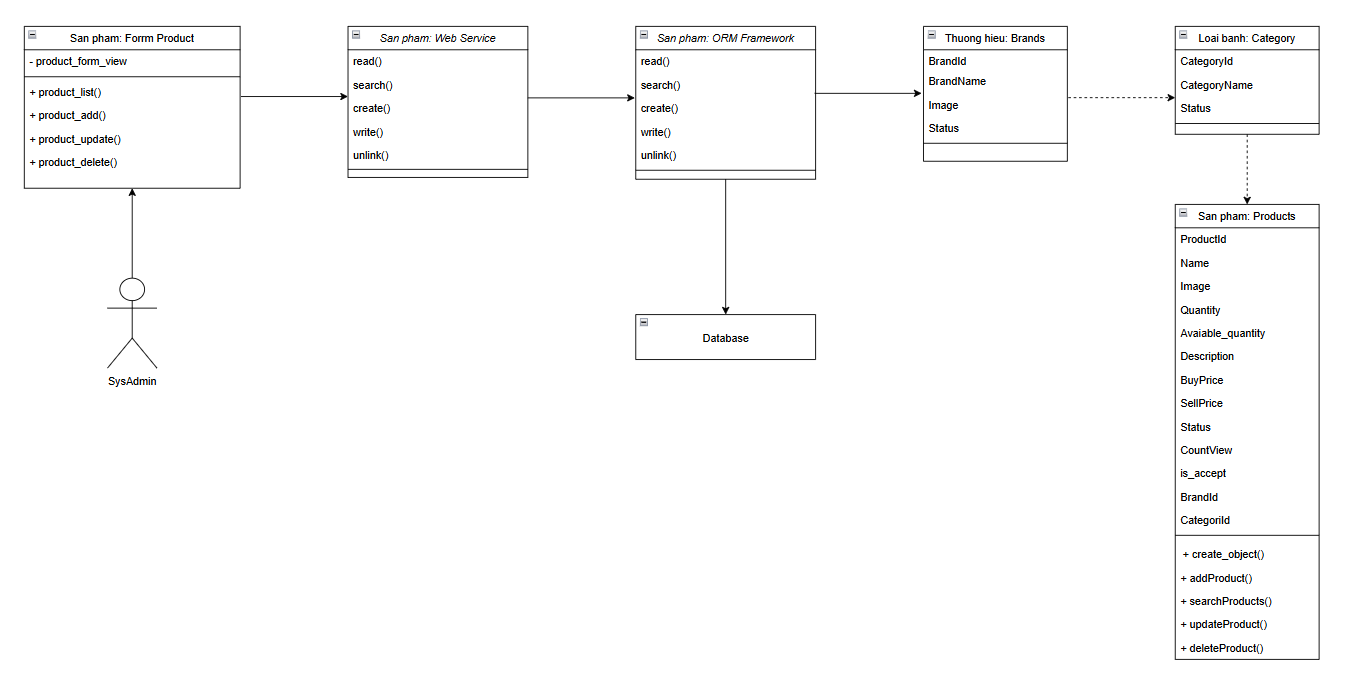
Hình 9. Sơ đồ Activity quy trình Nhập hàng

### 4.3. Sơ đồ Sequence



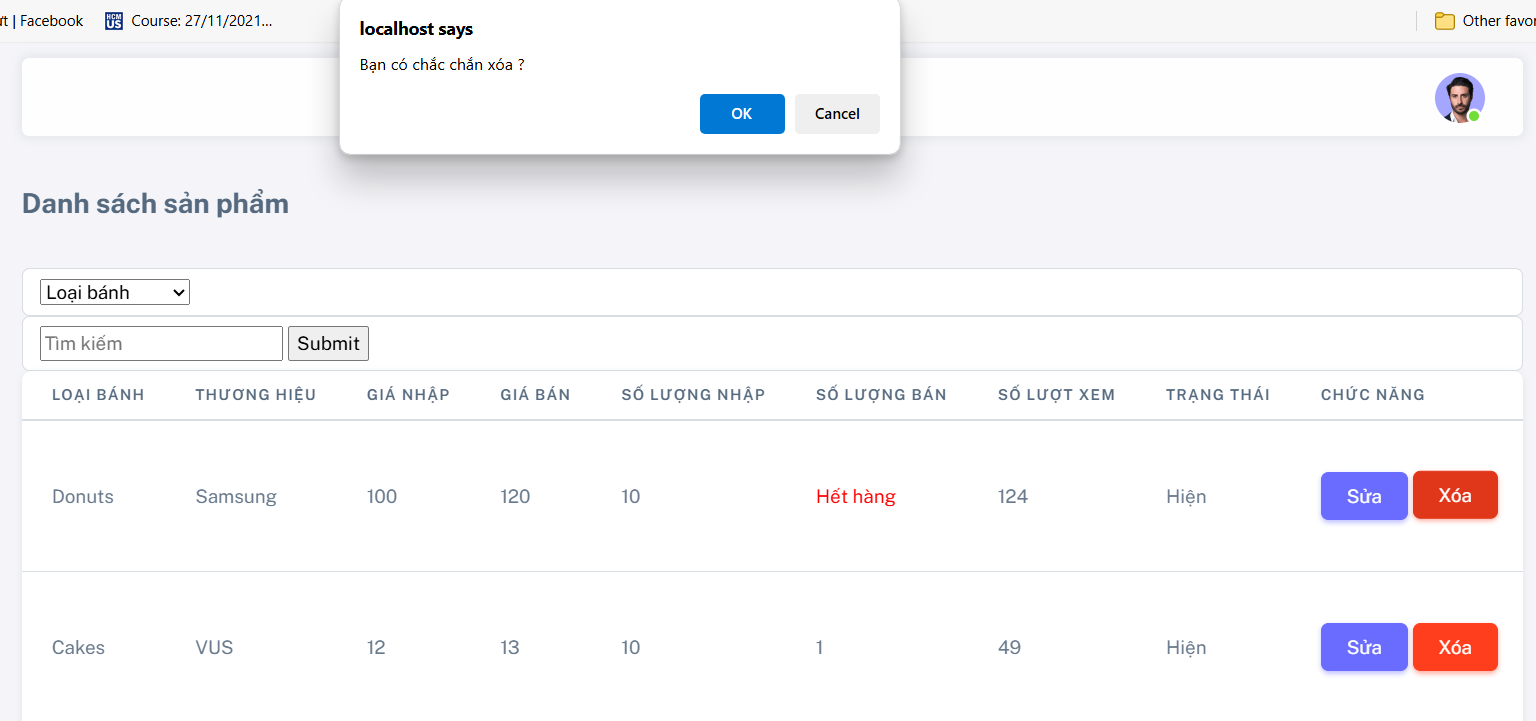
Hình 10. Sơ đồ Sequence quy trình Nhập hàng

### 4.4. Sơ đồ Class



Hình 11. Sơ đồ Class quy trình Nhập hàng

### 4.5. Giao diện minh họa

Xóa sản phẩm  


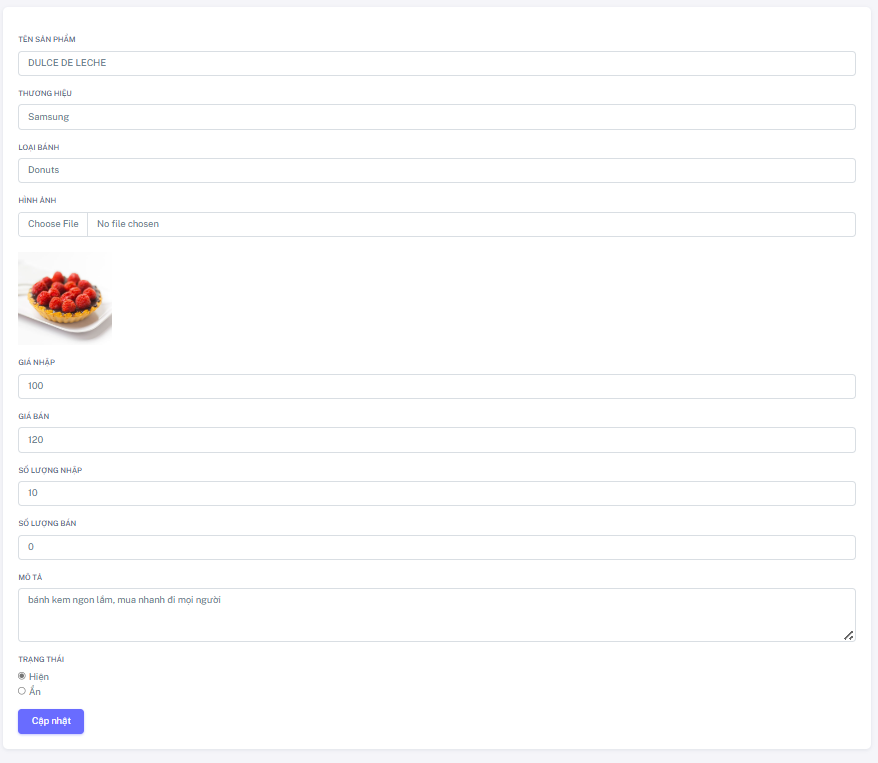
Hình 12. Giao diện xóa sản phẩm Nhập hàng

Thêm sản phẩm



Hình 13. Giao diện thêm sản phẩm Nhập hàng

Cập nhật sản phẩm



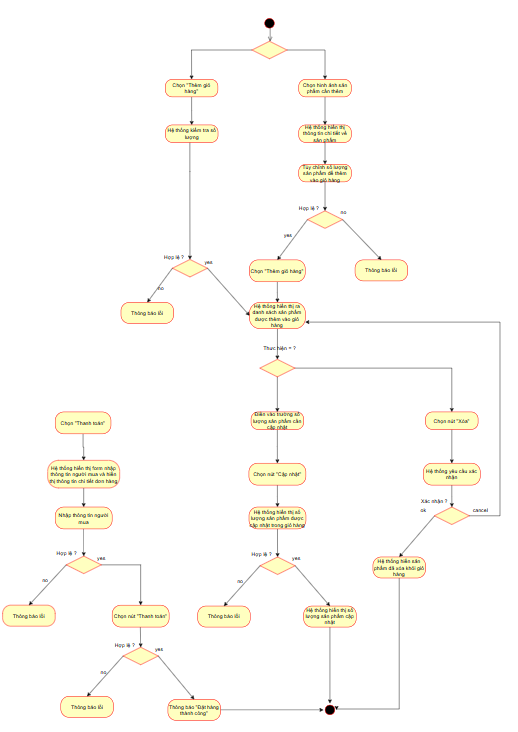
Hình 14. Giao diện cập nhật sản phẩm Nhập hàng

## 5. Mua hàng

### 5.1. Focused UseCase

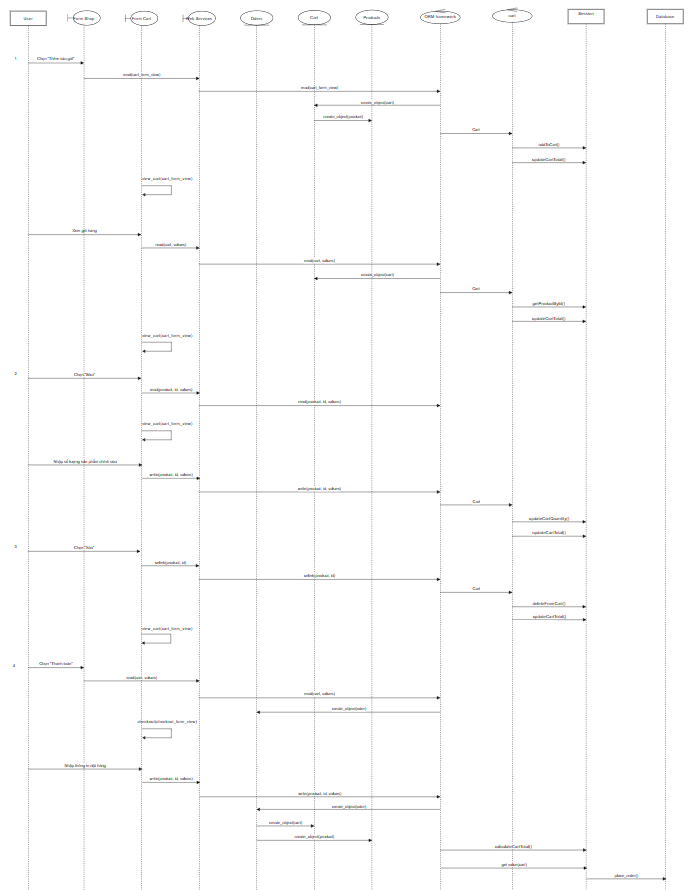
|  |  |  |
| --- | --- | --- |
|  | | |
| **Use case number** | **UC2** | |
| **Use case name** | **Giỏ hàng** | |
| **Actor(s)** | **Người mua hàng** | |
| **Maturity** | **Focused** | |
| **Summary** | *Use case mô tả quá trình mua hàng của người dùng trong hệ thống* | |
| **Basic Course of events** | **Actor Action** | **System response** |
| 1. Use Case bắt đầu khi Actor chọn “Thêm giỏ hàng” với mặc định là 1 sản phẩm **A1** |  |
|  | 2. Hệ thống thực hiện  **{Kiểm tra số lượng} E1** |
|  | 3. Hệ thống hiển thị ra danh sách sản phẩm được thêm vào giỏ hàng |
| 4. Actor có thể tiến hành thanh toán **A2 A3** |  |
|  | 5. Hệ thống hiển thị form thanh toán giỏ hàng |
| 6. Actor nhập thông tin thanh toán |  |
|  | 7. Hệ thống thực hiện  **{Kiểm tra ô text} E1** |
|  | 8. Hệ thống cập nhật thông tin đơn hàng vào database |
|  | 9. Số lượng sản phẩm trong kho sẽ giảm theo số lượng đặt hàng |
| **Alternative Paths** | **A1** |  |
| **Actor Action** | **System response** |
| 1.Actor bấm vào hình ảnh sản phẩm cần thêm |  |
|  | 2. Hệ thống hiển thị thông tin chi tiết về sản phẩm bao gồm: Tên, Giá, Số lượng, Loại bánh, Thương hiệu |
| 3. Actor tùy chỉnh số lượng sản phẩm để thêm vào giỏ hàng |  |
|  | 4. Hệ thống thực hiện  **{Kiểm tra số lượng} E1** |
| 5. Actor chọn “Thêm giỏ hàng” |  |
|  | 6. Hệ thống hiển thị ra danh sách sản phẩm được thêm vào giỏ hàng |
| **A2** |  |
| **Actor Action** | **System response** |
| 1. Actor điền vào trường số lượng sản phẩm cần cập nhật |  |
| 2. Actor bấm “Cập nhật” |  |
|  | 3. Hệ thống thực hiện  **{Kiểm tra số lượng} E1** |
|  | 4. Hệ thống hiển thị số lượng sản phẩm cập nhật trong giỏ hàng |
| **A3** |  |
| **Actor Action** | **System response** |
| 1.Actor bấm “Xóa” |  |
|  | 2. Hệ thống yêu cầu xác nhận |
| 3. Actor chọn xác nhận “OK” **A4** |  |
|  | 6. Hệ thống hiển thị sản phẩm đã xóa khỏi giỏ hàng |
| **A4** |  |
| **Actor Action** | **System response** |
| 1.Actor có thể bấm “Cancel” thao thác |  |
|  | 2.Hệ thống xóa yêu cầu xác nhận. Trở về bước 2 của Basic Course of Events. |
| **Exception Path** | **E1.** Tại **{Kiểm tra số lượng}**, kiểm tra số lượng có còn đủ sản phẩm trong kho hay không, nếu có thông báo lỗi tương ứng | |
| **Triggers:** | Khi actor muốn mua hàng | |
| **Assumption:** | None | |
| **Preconditions:** | None | |
| **Post Conditions:** | None | |

### 5.2. Sơ đồ Activity



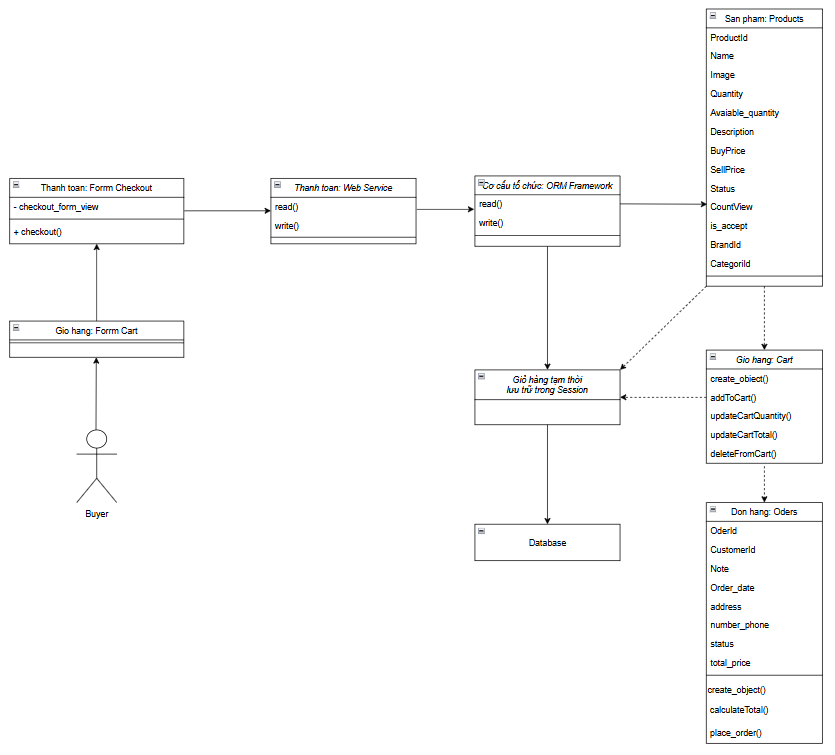
Hình 15. Sơ đồ Activity quy trình Mua hàng

### 5.3. Sơ đồ Sequence



Hình 16. Sơ đồ Sequence quy trình Mua hàng

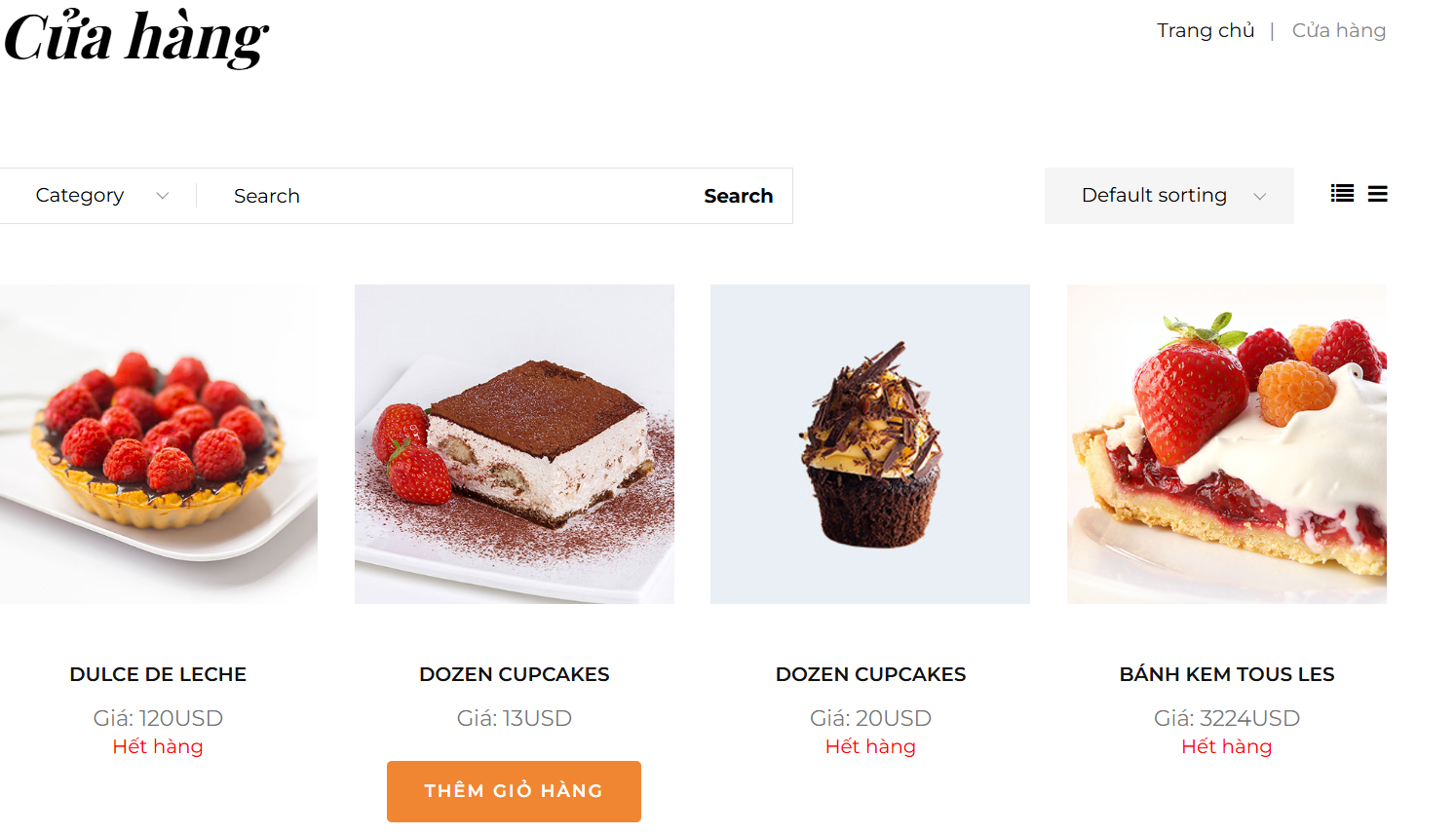
### 5.4. Sơ đồ Class



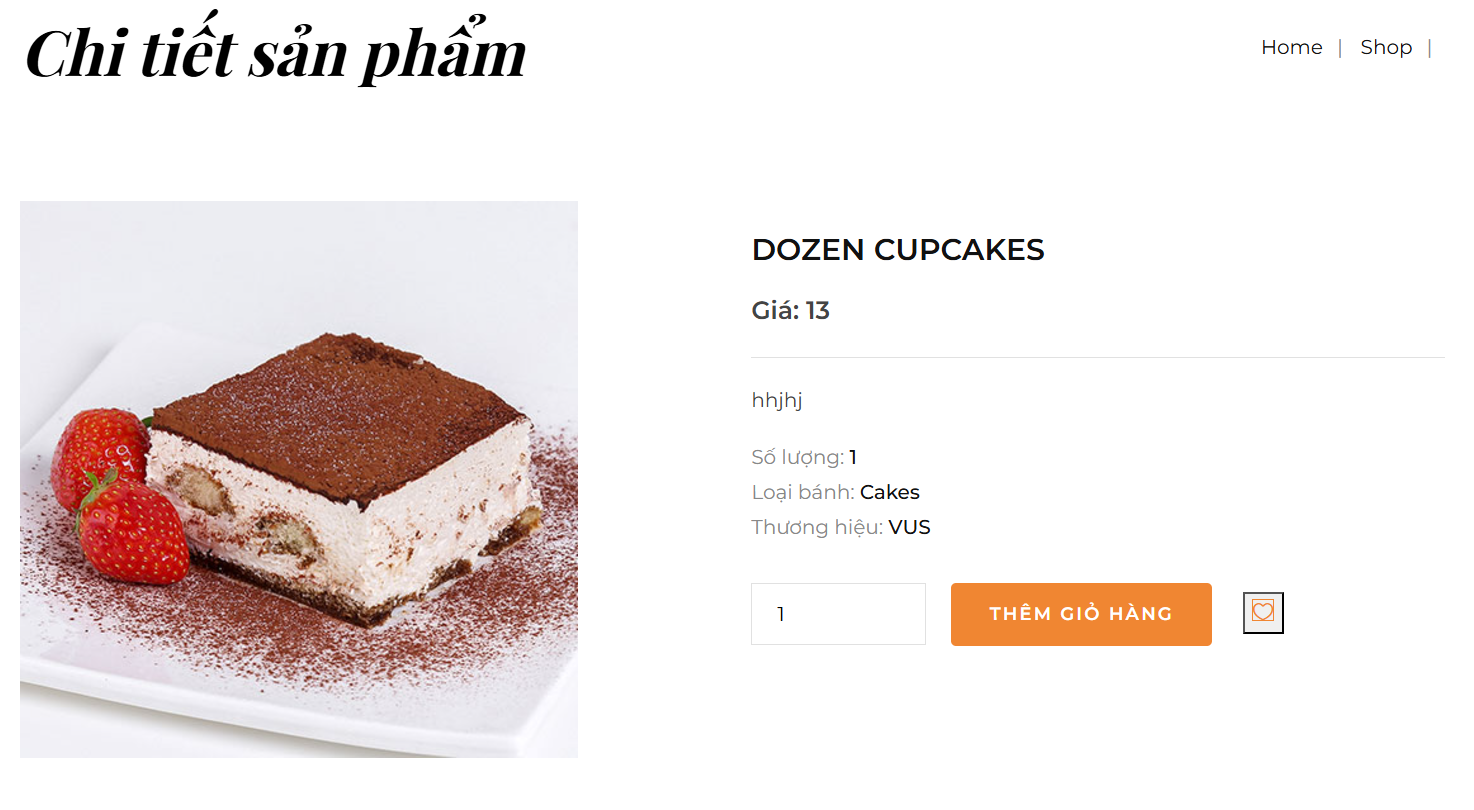
Hình 17. Sơ đồ Class quy trình Mua hàng

### 5.5. Giao diện minh họa

Thêm sản phẩm vào giỏ hàng

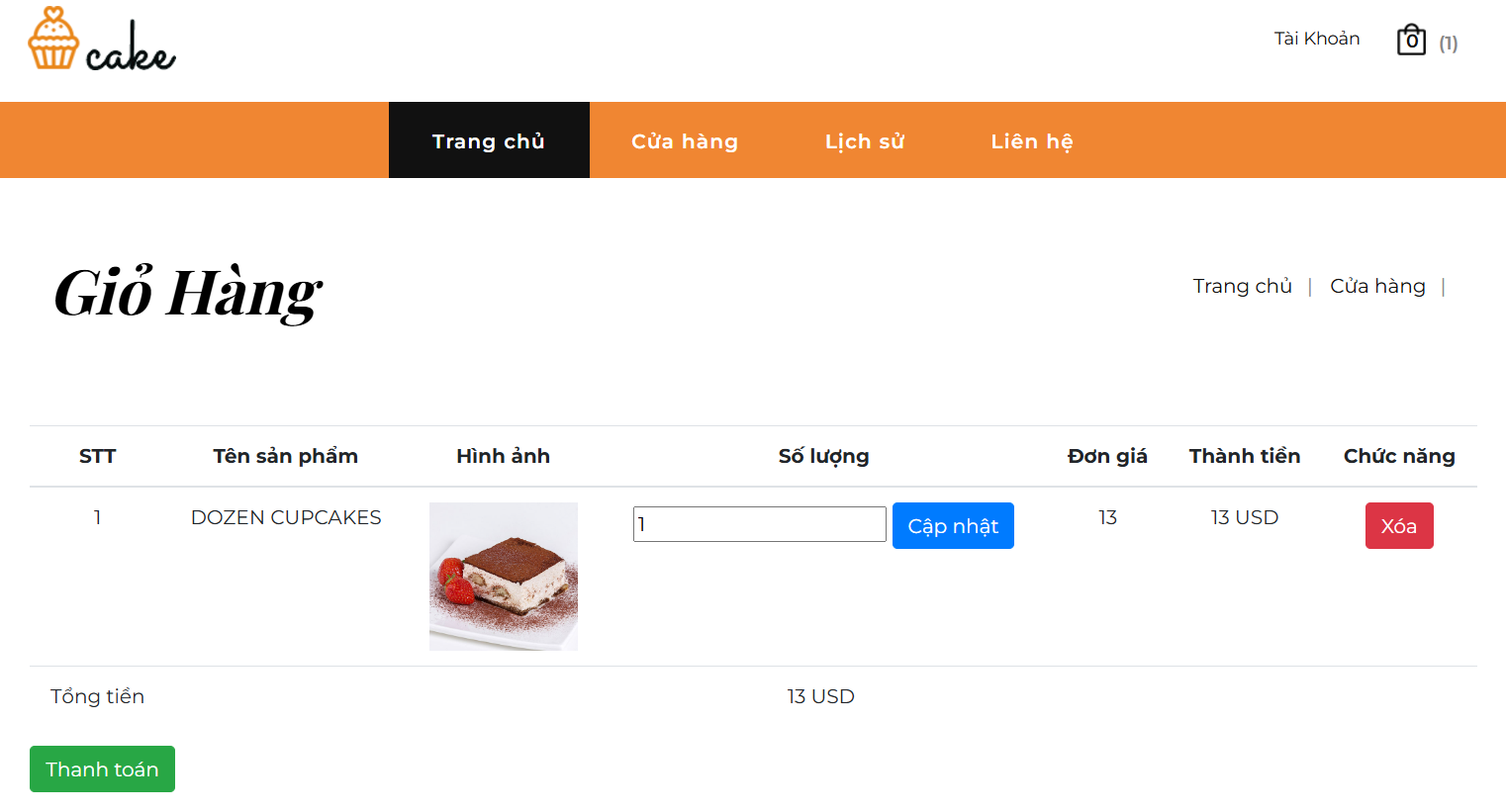


Hình 18. Giao diện thêm mặc định một sản phẩm vào giỏ hàng khi Mua hàng



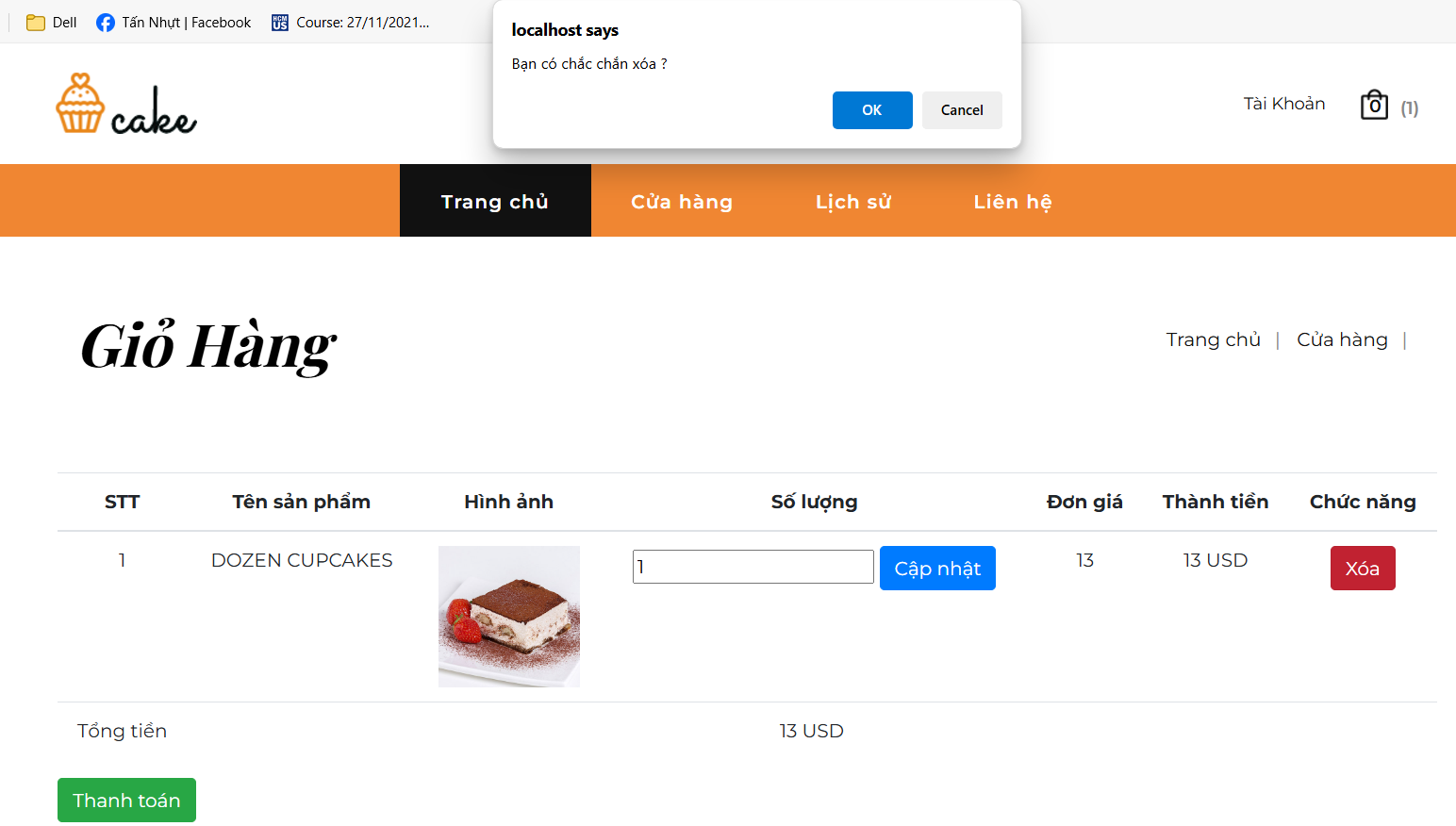
Hình 19. Giao diện thêm sản phẩm với số lượng tùy chọn khi Mua hàng

Cập nhật số lượng sản phẩm trong giỏ hàng



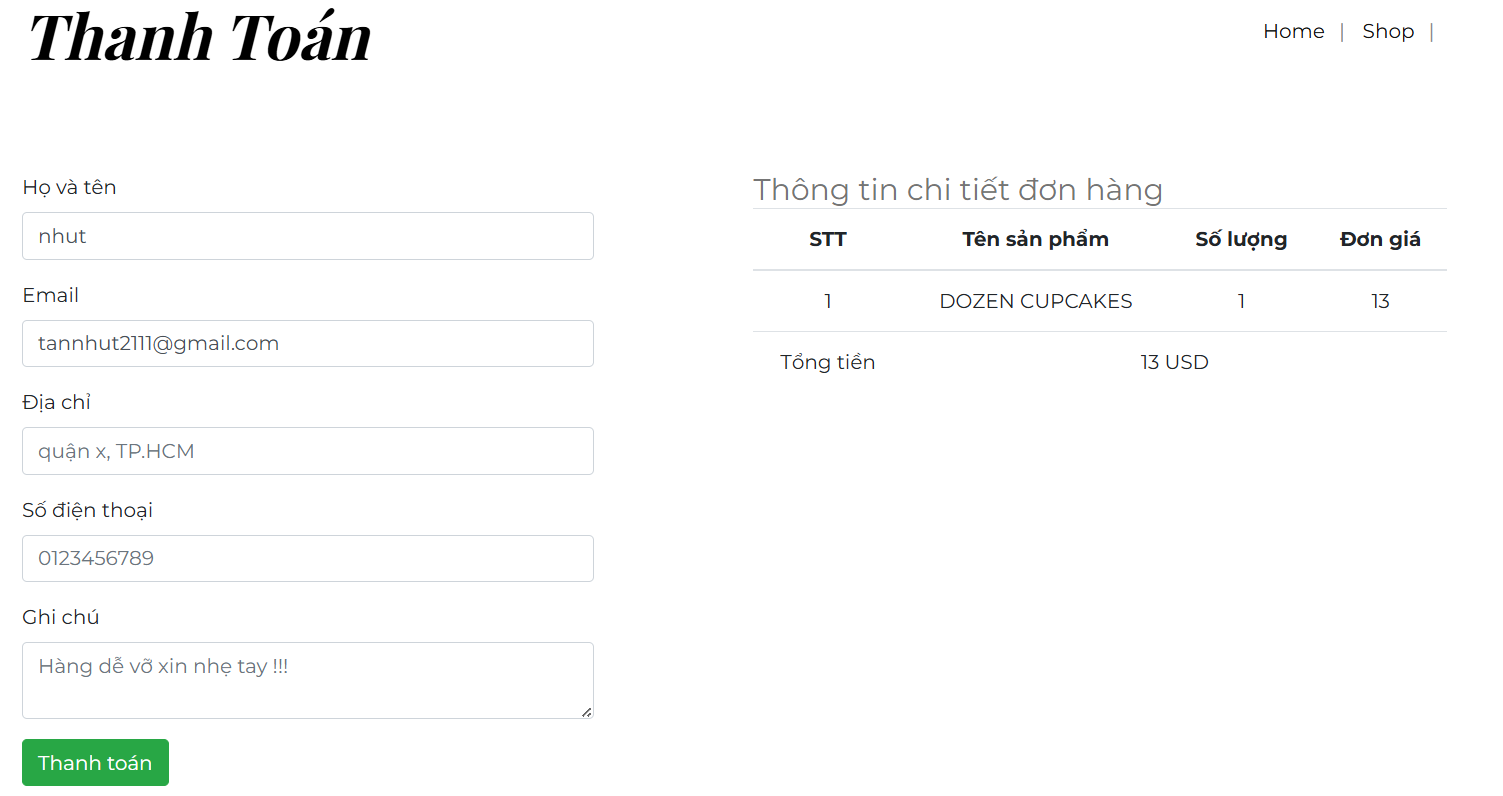
Hình 20. Giao diện xem và cập nhật giỏ hàng

Xóa sản phẩm khỏi giỏ hàng



Hình 21. Giao diện xóa sản phẩm khỏi giỏ hàng

Thanh toán giỏ hàng



Hình 22. Giao diện thanh toán giỏ hàng

# CHƯƠNG 3: KẾ HOẠCH KIỂM THỬ

## 1. Giới thiệu

### 1.1. Mục đích

Kế hoạch kiểm thử này nhằm mục đích:

Xác định thông tin dự án và các thành phần phần mềm cần kiểm thử.

Liệt kê các yêu cầu kiểm thử (ở cấp độ cao).

Đề xuất và mô tả chiến lược kiểm thử.

Xác định nguồn lực và ước tính nỗ lực kiểm thử.

Liệt kê các sản phẩm đầu ra của dự án kiểm thử.

### 1.2. Bối cảnh

Hệ thống được phát triển là một trang web bán bánh kem trực tuyến, hỗ trợ người dùng chọn mua các loại bánh kem với nhiều hương vị và mẫu mã khác nhau. Trang web cung cấp các dịch vụ như đặt hàng trực tuyến, thanh toán trực tuyến, quản lý giỏ hàng, tìm kiếm và lọc sản phẩm, cũng như gợi ý các loại bánh nổi bật.

### 1.3. Phạm vi

Kế hoạch kiểm thử này áp dụng cho tất cả các yêu cầu của dự án Cake-Sale-Website được định nghĩa trong Tài liệu Yêu cầu Phần mềm. Phạm vi kiểm thử bao gồm tất cả các chức năng chính của trang web bán bánh kem, từ việc hiển thị sản phẩm, quản lý giỏ hàng, đặt hàng, đến thanh toán trực tuyến và quản lý thông tin khách hàng. Ngoài ra, hệ thống cũng phải được kiểm thử về khả năng tìm kiếm, lọc sản phẩm theo danh mục và hỗ trợ gợi ý các sản phẩm nổi bật. Các yếu tố liên quan đến trải nghiệm người dùng như giao diện ,cũng sẽ được đưa vào kiểm thử. Kế hoạch này cũng bao gồm việc đảm bảo trang web hoạt động ổn định, kiểm tra các trường hợp sử dụng bình thường cũng như các trường hợp ngoại lệ, nhằm đảm bảo hệ thống vận hành hiệu quả và không gặp lỗi trong quá trình sử dụng thực tế của người dùng.

### 1.4. Nhận diện dự án

Bảng dưới đây xác định tài liệu và tài nguyên sử dụng để phát triển kế hoạch kiểm thử, như tài liệu yêu cầu, tài liệu đặc tả chức năng, và báo cáo use-case.

Bảng dưới đây xác định các tài liệu và tính khả dụng được sử dụng để phát triển Kế hoạch Kiểm thử:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tài liệu (phiên bản / ngày)** | **Đã tạo hoặc có sẵn** | **Đã nhận hoặc đã xem xét** | **Tác giả hoặc Nguồn** | **Ghi chú** |
| Yêu cầu | Có Không | Có Không |  |  |
| Đặc tả chức năng | Có Không | Có Không |  |  |
| Báo cáo Use-Case | Có Không | Có Không |  |  |
| Kế hoạch dự án | Có Không | Có Không |  |  |
| Thiết kế | Có Không | Có Không |  |  |
| Nguyên mẫu | Có Không | Có Không |  |  |
| Hướng dẫn sử dụng | Có Không | Có Không |  |  |
| Mô hình hoặc quy trình kinh doanh | Có Không | Có Không |  |  |
| Mô hình hoặc quy trình dữ liệu | Có Không | Có Không |  |  |
| Chức năng và quy tắc kinh doanh | Có Không | Có Không |  |  |
| Đánh giá rủi ro dự án hoặc kinh doanh | Có Không | Có Không |  |  |

### 1.5 Tài liệu tham khảo

|  |  |  |
| --- | --- | --- |
| **STT** | **Tài liệu** | **Mô tả** |
| 1 | Tài liệu yêu cầu phần mềm dự án ABC, Phiên bản 1.0.3.3, Thứ Hai, 25/08/2010 | Tài liệu SRS mô tả đầy đủ hành vi bên ngoài của ứng dụng hoặc hệ thống con được xác định. Nó cũng mô tả các yêu cầu phi chức năng, các giới hạn thiết kế, và các yếu tố khác cần thiết để cung cấp mô tả hoàn chỉnh và toàn diện về các yêu cầu phần mềm. |
| 2 | Kế hoạch phát triển phần mềm dự án ABC, Phiên bản 1.0.5, 05/09/2010 | Mục đích của Kế hoạch phát triển phần mềm là xác định các nhiệm vụ của dự án ABC (được gọi là IM hoặc dự án) để phát triển. Nó tập trung vào các khả năng mà các bên liên quan tìm kiếm, đặc biệt là người dùng mục tiêu, và lý do cho sự tìm kiếm này. |

## 2. Các tính năng cần kiểm thử

### 2.1. Chức năng (Khách hàng)

2.1.1. Thêm sản phẩm vào giỏ hàng

* Xác minh rằng hệ thống cho phép khách hàng chọn sản phẩm từ danh mục, thêm vào giỏ hàng.
* Xác minh rằng mỗi lần thêm sản phẩm, hệ thống sẽ lưu lại và cập nhật số lượng và tổng tiền tương ứng trong giỏ hàng.

2.1.2. Xóa sản phẩm khỏi giỏ hàng

* Xác minh rằng hệ thống cung cấp chức năng để xóa sản phẩm không mong muốn khỏi giỏ hàng.
* Xác minh rằng khi chọn xóa, sản phẩm sẽ được xóa khỏi giỏ hàng..

2.1.3. Cập nhật sản phẩm trong giỏ hàng

* Xác minh rằng hệ thống cho phép thay đổi số lượng sản phẩm trong giỏ hàng.
* Xác minh rằng khi nhấn cập nhật, hệ thống sẽ tính lại số lượng sản phẩm và tổng tiền.

2.1.4. Thanh toán giỏ hàng

* Xác minh rằng hệ thống hiển thị các sản phẩm đã chọn trong giỏ hàng và tổng tiền
* Xác minh rằng hệ thống cho phép khách hàng thanh toán các sản phẩm đã chọn trong giỏ hàng.

### 2.2. Chức năng (Người quản lý)

2.2.1. Thêm sản phẩm vào kho

* Xác minh rằng hệ thống cho phép người quản lý thêm sản phẩm mới vào kho.
* Xác minh rằng thông tin sản phẩm bao gồm tên, giá, hình ảnh, số lượng và các thuộc tính khác được nhập đầy đủ và lưu trữ chính xác.

2.2.2. Xóa sản phẩm khỏi kho

* Xác minh rằng hệ thống cho phép người quản lý xóa các sản phẩm không còn kinh doanh khỏi kho.
* Xác minh rằng sản phẩm sau khi xóa không còn hiển thị trong danh mục sản phẩm.

2.2.3. Cập nhật sản phẩm trong kho

* Xác minh rằng hệ thống cho phép người quản lý cập nhật thông tin sản phẩm.
* Xác minh rằng các thông tin như giá, số lượng, hình ảnh và mô tả được cập nhật đúng và hiển thị chính xác sau khi lưu.

### 2.3. Tính khả dụng

* Xác minh rằng hệ thống hỗ trợ hiển thị biểu tượng hình ảnh bánh trong giao diện.
* Xác minh rằng giao diện người dùng thân thiện, dễ sử dụng và trực quan.

### 2.4. Các giới hạn thiết kế

**2.4.1. Độ dài tin nhắn**

* Xác minh rằng hệ thống không cho phép tin nhắn hoặc thông báo vượt quá độ dài tối đa được quy định.

**2.4.2. Môi trường kiểm thử**

* Xác minh rằng hệ thống hoạt động ổn định trên các trình duyệt phổ biến và trên cả thiết bị di động lẫn máy tính.

**2.4.3. Ngôn ngữ**

* Xác minh rằng hệ thống hỗ trợ hiển thị tiếng Việt một cách chính xác và không lỗi font.

**2.4.4. Đa thiết bị**

* Xác minh rằng hệ thống hiển thị chính xác và tương thích trên cả máy tính và điện thoại với các kích thước màn hình khác nhau.

### 2.5. Giao diện

#### 2.5.1. Giao diện người dùng

* Xác minh rằng giao diện trang chính cung cấp thông tin sản phẩm nổi bật, bao gồm hình ảnh, giá và nút “Thêm vào giỏ hàng”.
* Xác minh rằng giao diện giỏ hàng hiển thị các sản phẩm đã chọn, cho phép cập nhật số lượng, xóa sản phẩm, và thanh toán.
* Xác minh rằng giao diện chi tiết sản phẩm hiển thị đầy đủ thông tin sản phẩm bao gồm hình ảnh, mô tả, giá cả, số lượng, loại bánh và tên thương hiệu.

**2.5.2. Giao diện người quản lý**

* Xác minh rằng giao diện quản lý loại bánh cho phép thêm, sửa đổi, và xóa loại bánh.

## 3. Các tính năng không cần kiểm thử

### 3.1. Chức năng (Khách hàng)

**3.1.1. Liên hệ với người quản lý**

* Xác minh rằng hệ thống cho phép khách hàng gửi thông tin liên hệ hoặc thắc mắc về sản phẩm, dịch vụ.
* Xác minh rằng thông tin liên hệ/thắc mắc được gửi đến người quản lý một cách chính xác để phản hồi.

**3.1.2. Tìm kiếm sản phẩm**

* Xác minh rằng khách hàng có thể tìm kiếm sản phẩm theo tên thông qua thanh tìm kiếm.
* Xác minh rằng kết quả tìm kiếm trả về chính xác dựa trên từ khóa đã nhập.

**3.1.3. Bộ lọc sản phẩm**

* Xác minh rằng khách hàng có thể lọc sản phẩm theo loại bánh.
* Xác minh rằng danh sách sản phẩm hiển thị đúng theo tiêu chí lọc.

**3.1.4. Xem lịch sử mua hàng**

* Xác minh rằng hệ thống hiển thị lịch sử mua hàng bao gồm chi tiết sản phẩm, giá cả, trạng thái giao hàng, và các thông tin liên quan.
* Xác minh rằng khách hàng chỉ có thể xem lịch sử mua hàng của chính mình.

**3.1.5. Theo dõi trạng thái đơn hàng**

* Xác minh rằng khách hàng có thể theo dõi trạng thái đơn hàng của mình khi người quản lý cập nhật.
* Xác minh rằng trạng thái đơn hàng được hiển thị chính xác và rõ ràng.

**3.1.6. Xem sản phẩm trong giỏ hàng**

* Xác minh rằng khách hàng có thể xem danh sách sản phẩm đã thêm vào giỏ hàng.
* Xác minh rằng hệ thống hiển thị đầy đủ thông tin sản phẩm trong giỏ hàng (tên, số lượng, giá, tổng tiền).

**3.1.7. Xem thông tin chi tiết sản phẩm**

* Xác minh rằng hệ thống hiển thị thông tin chi tiết của từng sản phẩm, bao gồm tên sản phẩm, mô tả, hình ảnh, giá cả, số lượng hàng còn, loại bánh, và thương hiệu.
* Xác minh rằng thông tin chi tiết sản phẩm được cập nhật đầy đủ và chính xác.

### 3.2. Chức năng (Người quản lý)

**3.2.1. Tạo tài khoản người dùng**

* Xác minh rằng hệ thống cung cấp chức năng tạo tài khoản người dùng cho dịch vụ bán bánh.
* Xác minh rằng người quản lý có thể tạo nhiều tài khoản người dùng khác nhau.
* Xác minh rằng hệ thống hỗ trợ tùy chọn "Nhớ mật khẩu" cho từng tài khoản người dùng.

**3.2.2. Xóa tài khoản người dùng**

* Xác minh rằng hệ thống cho phép xóa tài khoản người dùng khỏi hệ thống.
* Xác minh rằng tài khoản đã xóa sẽ không thể truy cập hoặc sử dụng dịch vụ.

**3.2.3. Tạo tài khoản người quản lý**

* Xác minh rằng hệ thống cung cấp khả năng tạo tài khoản người quản lý để quản lý cửa hàng và sản phẩm.

**3.2.4. Gợi ý sản phẩm nổi bật**

* Xác minh rằng hệ thống tự động đề xuất các sản phẩm nổi bật dựa trên lượt xem của khách hàng.
* Xác minh rằng danh sách sản phẩm nổi bật thay đổi linh hoạt dựa trên dữ liệu thực tế.

**3.2.5. Bộ lọc sản phẩm trong kho**

* Xác minh rằng hệ thống cho phép lọc sản phẩm trong kho theo các tiêu chí như loại bánh, thương hiệu, hoặc tình trạng kho.

**3.2.6. Thêm loại bánh**

* Xác minh rằng người quản lý có thể thêm các loại bánh mới vào hệ thống.
* Xác minh rằng loại bánh được phân loại chính xác để hiển thị trên website.

**3.2.7. Xóa loại bánh**

* Xác minh rằng hệ thống cho phép xóa các loại bánh không còn kinh doanh.
* Xác minh rằng loại bánh đã xóa không còn hiển thị trong danh mục.

**3.2.8. Cập nhật loại bánh**

* Xác minh rằng người quản lý có thể cập nhật thông tin loại bánh, bao gồm tên loại bánh.
* Xác minh rằng thông tin loại bánh sau khi cập nhật được hiển thị chính xác.

**3.2.9. Thêm thương hiệu**

* Xác minh rằng hệ thống cho phép thêm các thương hiệu mới, bao gồm tên thương hiệu và hình ảnh.

**3.2.10. Xóa thương hiệu**

* Xác minh rằng hệ thống hỗ trợ xóa thương hiệu không còn kinh doanh hoặc hợp tác.
* Xác minh rằng thương hiệu đã xóa sẽ không còn xuất hiện trong danh mục.

**3.2.11. Cập nhật thương hiệu**

* Xác minh rằng người quản lý có thể chỉnh sửa thông tin thương hiệu, bao gồm tên thương hiệu và hình ảnh.

**3.2.12. Xem thông tin người dùng**

* Xác minh rằng người quản lý có thể xem thông tin tài khoản khách hàng, bao gồm lịch sử mua hàng và thông tin liên hệ.
* Xác minh rằng chỉ người quản lý mới được quyền truy cập thông tin này.

**3.2.13. Xem thông tin liên hệ của người dùng**

* Xác minh rằng hệ thống cho phép người quản lý xem và quản lý thông tin liên hệ khách hàng.
* Xác minh rằng thông tin liên hệ được hiển thị đầy đủ và chính xác.

**3.2.14. Tìm kiếm sản phẩm trong kho**

* Xác minh rằng người quản lý có thể tìm kiếm sản phẩm trong kho dựa trên tên sản phẩm, mã sản phẩm, hoặc các tiêu chí khác.

**3.2.15. Xem chi tiết thông tin đơn hàng**

* Xác minh rằng người quản lý có thể xem toàn bộ thông tin chi tiết của từng đơn hàng, bao gồm thông tin người mua và sản phẩm đã mua.

**3.2.16. Cập nhật trạng thái đơn hàng**

* Xác minh rằng hệ thống cho phép cập nhật trạng thái đơn hàng (chưa xử lý, đang xử lý, đã xử lý, đã giao hàng).
* Xác minh rằng trạng thái đơn hàng hiển thị đúng với tình hình thực tế.

**3.2.17. Xóa đơn hàng**

* Xác minh rằng hệ thống hỗ trợ xóa đơn hàng không hợp lệ hoặc bị hủy.
* Xác minh rằng các đơn hàng đã xóa sẽ không còn hiển thị trong hệ thống.

## 4. Chiến lược kiểm thử

### 4.1. Các loại kiểm thử

|  |  |
| --- | --- |
| **Mục tiêu kiểm thử** | Đảm bảo chức năng mục tiêu được kiểm thử hoạt động đúng, bao gồm điều hướng, nhập liệu, xử lý và truy xuất. |
| **Kỹ thuật** | Thực hiện từng trường hợp sử dụng, luồng trường hợp sử dụng hoặc chức năng, sử dụng dữ liệu hợp lệ và không hợp lệ, để xác minh các điều sau:  - Kết quả mong đợi xảy ra khi dữ liệu hợp lệ được sử dụng.  - Thông báo lỗi hoặc cảnh báo phù hợp được hiển thị khi dữ liệu không hợp lệ được sử dụng. |
| **Tiêu chí hoàn thành** | - Tất cả các bài kiểm thử đã được thực hiện.  - Tất cả các lỗi đã được xác định đều đã được khắc phục. |
| **Cân nhắc đặc biệt** | Xác định hoặc mô tả những mục hoặc vấn đề (nội bộ hoặc bên ngoài) ảnh hưởng đến việc thực hiện và thực thi kiểm thử chức năng. |

### 4.2. Công cụ

Công cụ kiểm thử bao gồm kiểm thử thủ công và kiểm thử tự động

|  |  |  |
| --- | --- | --- |
| **Công cụ** | **Nhà cung cấp/Nội bộ** | **Phiên bản** |
| Kiểm thử thủ công (Manual Testing) | Nội bộ |  |
| Kiểm thử tự động (Auto Testing) | Nội bộ |  |

## 5. Nguồn lực:

Bảng dưới đây cho thấy nhân sự cần thiết của dự án

|  |  |
| --- | --- |
| **Nguồn nhân lực** | **Mô tả trách nhiệm/công việc cụ thể** |
| **Nhà phát triển kiểm thử** | Nhận diện, đặt ưu tiên và thực hiện các trường hợp kiểm thử. |
|  | **Trách nhiệm:**  - Tạo kế hoạch kiểm thử  - Đánh giá hiệu quả của nỗ lực kiểm thử |
| **Kiểm thử viên** | Thực hiện các bài kiểm thử. |
|  | **Trách nhiệm:**  - Phát triển đặc tả trường hợp kiểm thử  - Phát triển các trường hợp kiểm thử |
|  | - Thực hiện kiểm thử  - Ghi lại kết quả  - Giám sát việc sửa lỗi  - Yêu cầu thay đổi tài liệu |

## 6. Các mốc của dự án

Dự án bao gồm ba giai đoạn phát triển với các vòng kiểm thử đầy đủ từ lập kế hoạch, thiết kế, phát triển, thực thi và đánh giá.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nhiệm vụ Mốc Kiểm Thử** | **Công sức** | **Ngày Bắt Đầu** | **Ngày Kết Thúc** |
| **Xây dựng 1** |  |  |  |
| Kế hoạch Kiểm Thử | 123.7 | 01/11/2024 | 01/11/2024 |
| Thiết kế Kiểm Thử | 300.5 | 10/11/2024 | 15/11/2024 |
| Thực hiện Kiểm Thử | 623.3 | 20/11/2024 | 30/11/2024 |
| Đánh giá Kiểm Thử | 216.1 | 01/12/2024 | 02/12/2024 |

## 7. Các sản phẩm đầu ra

### 7.1 Mô hình kiểm thử

Mô hình kiểm thử định nghĩa tất cả các trường hợp kiểm thử, tài liệu tham khảo, quy trình kiểm thử và kịch bản kiểm thử liên quan đến từng trường hợp kiểm thử.

### 7.2 Nhật ký kiểm thử

Sử dụng Microsoft Word để ghi lại và báo cáo kết quả kiểm thử, đảm bảo việc theo dõi các hoạt động và kết quả kiểm thử một cách chi tiết và rõ ràng.

# CHƯƠNG 4: PHƯƠNG PHÁP KIỂM THỬ

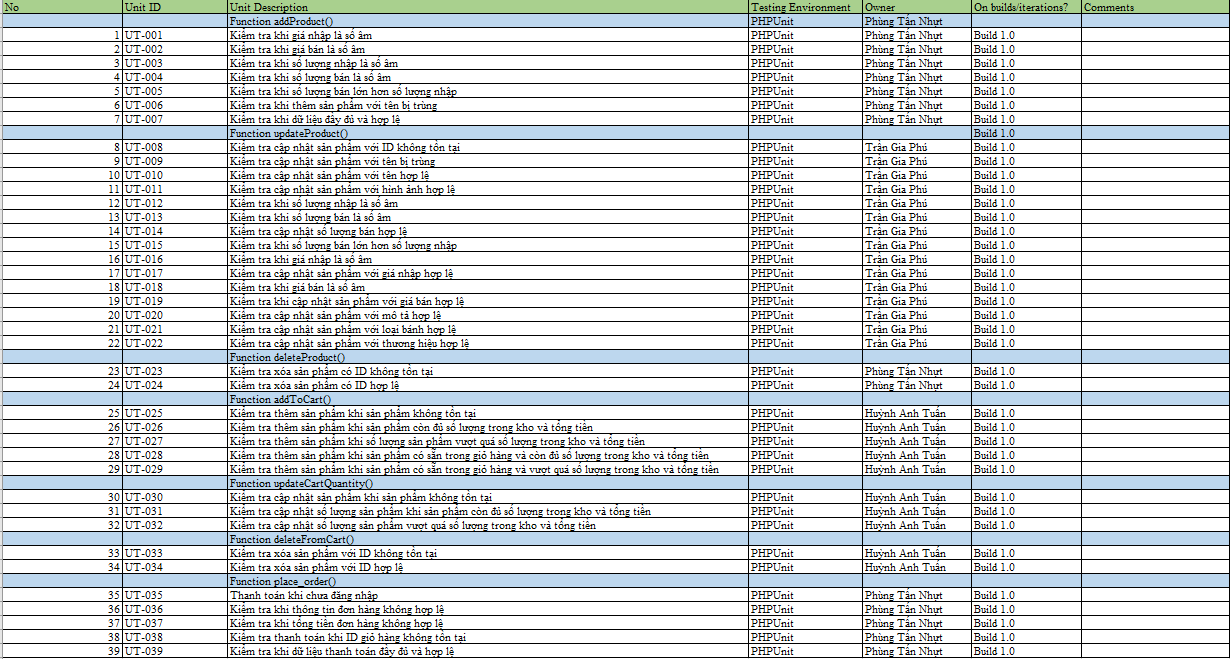
## 1. Unit Testing

**Unit testing** là một phương pháp kiểm thử phần mềm trong đó các đơn vị nhỏ nhất của mã nguồn (thường là các hàm hoặc phương thức) được kiểm tra độc lập để đảm bảo rằng chúng hoạt động đúng như mong đợi. Mục tiêu của unit testing là phát hiện lỗi ngay từ khi các thành phần cơ bản của hệ thống còn ở trạng thái đơn giản, tránh việc lỗi lan rộng vào các phần phức tạp hơn của chương trình.

**Mục đích của Unit Testing:**

* **Xác minh tính chính xác**: Đảm bảo rằng các đơn vị nhỏ nhất của hệ thống hoạt động đúng như mong đợi.
* **Phát hiện lỗi sớm**: Lỗi được phát hiện sớm trong quá trình phát triển, giúp giảm chi phí sửa lỗi sau này.
* **Tăng tính bảo trì**: Unit test giúp dễ dàng duy trì mã nguồn, vì các thay đổi có thể được kiểm tra ngay lập tức.
* **Cải thiện chất lượng phần mềm**: Tăng cường độ tin cậy và chất lượng của phần mềm thông qua việc đảm bảo các đơn vị mã hoạt động chính xác.
* **Hỗ trợ tái cấu trúc mã**: Khi có unit tests, lập trình viên có thể refactor (tái cấu trúc) mã nguồn mà không sợ làm gián đoạn tính năng đã có.

### 1.1. TestPlan



Hình 23. Bảng Kế hoạch kiểm thử TestPlan

**Function addProduct() (Thêm sản phẩm):**

* Kiểm tra các trường hợp khi thêm sản phẩm mới có hợp lệ hay không.
  + Kiểm tra giá nhập là số âm (Test khi giá nhập là giá trị âm).
  + Kiểm tra giá bán là số âm (Test khi giá bán là giá trị âm).
  + Kiểm tra số lượng nhập là số âm (Test khi số lượng nhập vào là số âm).
  + Kiểm tra số lượng bán là số âm (Test khi số lượng bán là số âm).
  + Kiểm tra khi số lượng bán lớn hơn số lượng nhập (Test khi số lượng bán vượt quá số lượng nhập vào).
  + Kiểm tra khi tên sản phẩm bị trùng (Test khi thêm sản phẩm với tên trùng với sản phẩm khác).
  + Kiểm tra dữ liệu đầy đủ và hợp lệ (Test khi dữ liệu đầu vào đầy đủ và hợp lệ).

**Function updateProduct() (Cập nhật sản phẩm):**

* Kiểm tra các trường hợp khi cập nhật sản phẩm có hợp lệ hay
  + Cập nhật sản phẩm với ID không tồn tại (Test khi ID sản phẩm không tồn tại trong cơ sở dữ liệu).
  + Cập nhật sản phẩm với tên bị trùng (Test khi cập nhật sản phẩm với tên đã tồn tại).
  + Kiểm tra khi số lượng nhập là số âm (Test khi cập nhật số lượng nhập vào là số âm).
  + Kiểm tra khi số lượng bán là số âm (Test khi cập nhật số lượng bán là số âm).
  + Cập nhật số lượng bán hợp lệ (Test khi số lượng bán hợp lệ).
  + Cập nhật tên, hình ảnh, số lượng nhập, số lượng bán, giá nhập, giá bán và mô tả sản phẩm (Test khi tên, hình ảnh, số lượng nhập, số lượng bán, giá nhập, giá bán và mô tả sản phẩm hợp lệ).

**Function deleteProduct() (Xóa sản phẩm):**

* Kiểm tra các trường hợp khi xóa sản phẩm:
  + Xóa sản phẩm với ID không tồn tại (Test khi xóa sản phẩm với ID không có trong cơ sở dữ liệu).
  + Xóa sản phẩm với ID hợp lệ (Test khi xóa sản phẩm với ID hợp lệ).

**Function addToCart() (Thêm sản phẩm vào giỏ hàng):**

* Kiểm tra các trường hợp thêm sản phẩm vào giỏ hàng:
  + Thêm sản phẩm khi sản phẩm không tồn tại (Test khi thêm sản phẩm mà sản phẩm đó không có trong hệ thống).
  + Thêm sản phẩm khi số lượng còn đủ trong kho và tổng tiền (Test khi thêm sản phẩm vào giỏ hàng với số lượng còn đủ trong kho và tổng tiền).
  + Thêm sản phẩm khi số lượng vượt quá số lượng trong kho và tổng tiền (Test khi thêm sản phẩm mà số lượng trong giỏ hàng vượt quá số lượng có sẵn và tổng tiền).
  + Thêm sản phẩm khi sản phẩm có sẵn trong giỏ hàng có đủ số lượng và tổng tiền (Test khi sản phẩm đã có trong giỏ hàng và vẫn đủ số lượng trong kho và tổng tiền).
  + Thêm sản phẩm khi sản phẩm có sẵn vượt quá số lượng trong kho và tổng tiền (Test khi sản phẩm đã có trong giỏ hàng vượt quá số lượng trong kho và vẫn đủ số lượng trong kho và tổng tiền).

**Function updateCartQuantity() (Cập nhật số lượng sản phẩm trong giỏ hàng):**

* Kiểm tra các trường hợp cập nhật số lượng sản phẩm giỏ hàng:
  + Cập nhật số lượng sản phẩm khi sản phẩm không tồn tại (Test khi cập nhật số lượng sản phẩm khi sản phẩm không tồn tại giỏ hàng).
  + Cập nhật số lượng sản phẩm khi có đủ số lượng trong kho và tổng tiền (Test khi cập nhật số lượng sản phẩm trong giỏ hàng mà kho còn đủ và tổng tiền).
  + Cập nhật số lượng sản phẩm khi số lượng vượt quá số lượng trong kho (Test khi cập nhật sản phẩm mà số lượng trong giỏ hàng vượt quá số lượng có sẵn và tổng tiền).

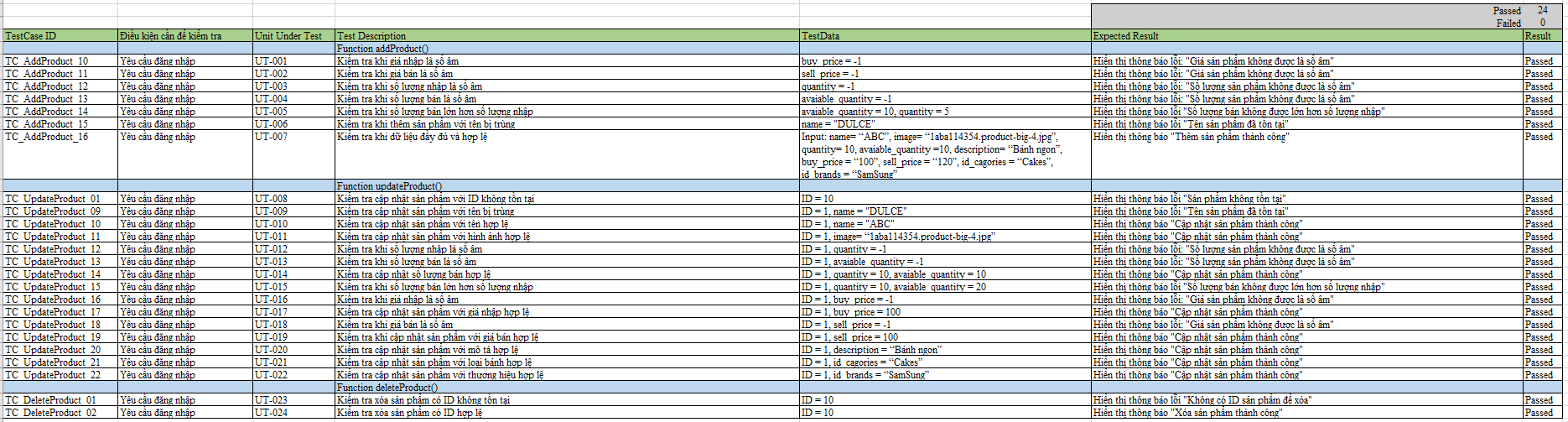
**Function deleteFormCart() (Xóa sản phẩm khỏi giỏ hàng):**

* Kiểm tra các trường hợp xóa sản phẩm khỏi giỏ hàng:
  + Xóa sản phẩm với ID không tồn tại (Test khi xóa sản phẩm với ID không có trong giỏ hàng).
  + Xóa sản phẩm với ID tồn tại (Test khi xóa sản phẩm với ID có trong giỏ hàng).

**Function place\_order() (Thanh toán):**

* Kiểm tra thanh toán trong giỏ hàng:
  + Thanh toán khi chưa đăng nhập (Test khi người dùng chưa đăng nhập và cố gắng thanh toán).
  + Kiểm tra thanh toán khi ID giỏ hàng không tồn tại (Test khi ID giỏ hàng không tồn tại trong hệ thống).
  + Kiểm tra thanh toán khi thông tin đơn hàng không hợp lệ (Test khi thông tin đơn hàng không hợp lệ khi thanh toán).
  + Kiểm tra thanh toán khi tổng tiền đơn hàng không hợp lệ (Test khi tổng tiền đơn hàng không hợp lệ khi thanh toán).
  + Thanh toán khi dữ liệu thanh toán đầy đủ và hợp lệ (Test khi dữ liệu thanh toán hoàn chỉnh và hợp lệ).

### 1.2. Quy trình nhập hàng



Hình 24. Unit Test Nhập hàng

**Passed 24 / Failed 0**:

* Tất cả các kiểm thử trong bảng này đều thành công, không có kiểm thử nào thất bại.

**Function addProduct() (Thêm sản phẩm):**

**TC\_AddProduct\_10:** Kiểm tra khi giá nhập là số âm (buy\_price = -1) – Kiểm thử đã pass với thông báo lỗi: "Giá sản phẩm không được là số âm".

**TC\_AddProduct\_11:** Kiểm tra khi giá bán là số âm (sell\_price = -1) – Kiểm thử đã pass với thông báo lỗi: "Giá sản phẩm không được là số âm".

**TC\_AddProduct\_12:** Kiểm tra khi số lượng nhập là số âm (quantity = -1) – Kiểm thử đã pass với thông báo lỗi: "Số lượng sản phẩm không được là số âm".

**TC\_AddProduct\_13:** Kiểm tra khi số lượng bán là số âm (avaiable\_quantity = -1) – Kiểm thử đã pass với thông báo lỗi: "Số lượng sản phẩm không được là số âm".

**TC\_AddProduct\_14:** Kiểm tra khi số lượng bán lớn hơn số lượng nhập (avaiable\_quantity = 10, quantity = 5) – Kiểm thử đã pass với thông báo lỗi: "Số lượng bán không được lớn hơn số lượng nhập".

**TC\_AddProduct\_15:** Kiểm tra khi thêm sản phẩm với tên bị trùng (name = "DULCE") – Kiểm thử đã pass với thông báo lỗi: "Tên sản phẩm đã tồn tại".

**TC\_AddProduct\_16:** Kiểm tra khi dữ liệu đầy đủ và hợp lệ – Kiểm thử đã pass với thông báo: "Thêm sản phẩm thành công".

**Function updateProduct() (Cập nhật sản phẩm):**

**TC\_UpdateProduct\_01:** Kiểm tra cập nhật sản phẩm với ID không tồn tại (ID = 10) – Kiểm thử đã pass với thông báo lỗi: "Sản phẩm không tồn tại".

**TC\_UpdateProduct\_09:** Kiểm tra cập nhật sản phẩm với tên bị trùng (name = "DULCE") – Kiểm thử đã pass với thông báo lỗi: "Tên sản phẩm đã tồn tại".

**TC\_UpdateProduct\_10:** Kiểm tra cập nhật sản phẩm với tên hợp lệ (name = "ABC") – Kiểm thử đã pass với thông báo: "Cập nhật sản phẩm thành công".

**TC\_UpdateProduct\_11:** Kiểm tra cập nhật sản phẩm với hình ảnh hợp lệ (image = “1aba114354.product-big-4.jpg”) – Kiểm thử đã pass với thông báo: "Cập nhật sản phẩm thành công".

**TC\_UpdateProduct\_12:** Kiểm tra khi số lượng nhập là số âm (quantity = -1) – Kiểm thử đã pass với thông báo lỗi: "Số lượng sản phẩm không được là số âm".

**TC\_UpdateProduct\_13:** Kiểm tra khi số lượng bán là số âm (avaiable\_quantity = -1) – Kiểm thử đã pass với thông báo lỗi: "Số lượng sản phẩm không được là số âm".

**TC\_UpdateProduct\_14:** Kiểm tra cập nhật số lượng bán hợp lệ (quantity = 10, avaiable\_quantity = 10) – Kiểm thử đã pass với thông báo: "Cập nhật sản phẩm thành công".

**TC\_UpdateProduct\_15:** Kiểm tra khi số lượng bán lớn hơn số lượng nhập (quantity = 10, avaiable\_quantity = 20) – Kiểm thử đã pass với thông báo lỗi: "Số lượng bán không được lớn hơn số lượng nhập".

**TC\_UpdateProduct\_16:** Kiểm tra khi giá nhập là số âm (buy\_price = -1) – Kiểm thử đã pass với thông báo lỗi: "Giá sản phẩm không được là số âm".

**TC\_UpdateProduct\_17:** Kiểm tra cập nhật sản phẩm với giá nhập hợp lệ (buy\_price = 100) – Kiểm thử đã pass với thông báo: "Cập nhật sản phẩm thành công".

**TC\_UpdateProduct\_18:** Kiểm tra khi giá bán là số âm (sell\_price = -1) – Kiểm thử đã pass với thông báo lỗi: "Giá sản phẩm không được là số âm".

**TC\_UpdateProduct\_19:** Kiểm tra khi cập nhật sản phẩm với giá bán hợp lệ (sell\_price = 100) – Kiểm thử đã pass với thông báo: "Cập nhật sản phẩm thành công".

**TC\_UpdateProduct\_20:** Kiểm tra cập nhật sản phẩm với mô tả hợp lệ (description = “Bánh ngon”) – Kiểm thử đã pass với thông báo: "Cập nhật sản phẩm thành công".

**TC\_UpdateProduct\_21:** Kiểm tra cập nhật sản phẩm với loại bánh hợp lệ (id\_cagories = “Cakes”) – Kiểm thử đã pass với thông báo: "Cập nhật sản phẩm thành công".

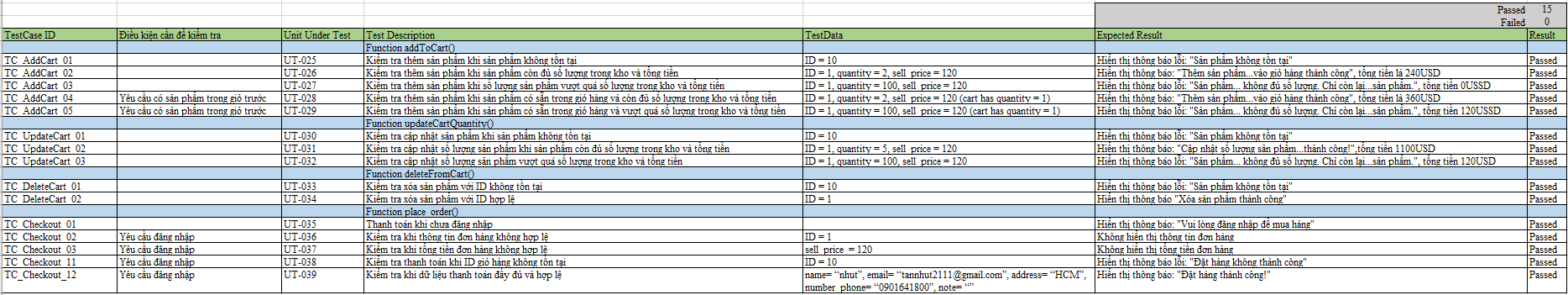
**TC\_UpdateProduct\_22:** Kiểm tra cập nhật sản phẩm với thương hiệu hợp lệ (id\_brands = “SamSung”) – Kiểm thử đã pass với thông báo: "Cập nhật sản phẩm thành công".

**Function deleteProduct() (Xóa sản phẩm):**

**TC\_DeleteProduct\_01:** Kiểm tra xóa sản phẩm có ID không tồn tại (ID = 10) – Kiểm thử đã pass với thông báo lỗi: "Không có ID sản phẩm để xóa".

**TC\_DeleteProduct\_02:** Kiểm tra xóa sản phẩm có ID hợp lệ (ID = 10) – Kiểm thử đã pass với thông báo: "Xóa sản phẩm thành công".

### 1.3. **Quy trình mua hàng**

****

Hình 25. Unit Test Mua hàng

**Passed 15 / Failed 0**:

* Tất cả các kiểm thử trong bảng này đều thành công, không có kiểm thử nào thất bại.

**Function addToCart() (Thêm sản phẩm vào giỏ hàng):**

* **TC\_AddCart\_01**: Kiểm tra khi thêm sản phẩm không tồn tại vào giỏ hàng. Kiểm thử đã pass với thông báo lỗi: "Sản phẩm không tồn tại".
* **TC\_AddCart\_02**: Kiểm tra khi thêm sản phẩm vào giỏ hàng và sản phẩm còn đủ số lượng trong kho. Kiểm thử đã pass với thông báo: "Thêm sản phẩm vào giỏ hàng thành công".
* **TC\_AddCart\_03**: Kiểm tra khi số lượng sản phẩm muốn thêm vào giỏ hàng vượt quá số lượng trong kho. Kiểm thử đã pass với thông báo lỗi: "Sản phẩm không đủ số lượng. Chỉ còn lại ... sản phẩm".
* **TC\_AddCart\_04**: Kiểm tra khi sản phẩm đã có sẵn trong giỏ hàng và vẫn còn đủ số lượng trong kho. Kiểm thử đã pass với thông báo: "Thêm sản phẩm vào giỏ hàng thành công".
* **TC\_AddCart\_05**: Kiểm tra khi sản phẩm đã có trong giỏ hàng nhưng số lượng muốn thêm vào vượt quá số lượng trong kho. Kiểm thử đã pass với thông báo lỗi: "Sản phẩm không đủ số lượng. Chỉ còn lại ... sản phẩm".
* **Function updateCartQuantity() (Cập nhật số lượng sản phẩm trong giỏ hàng):**
* **TC\_UpdateCart\_01**: Kiểm tra khi cập nhật số lượng sản phẩm trong giỏ hàng với ID không tồn tại. Kiểm thử đã pass với thông báo lỗi: "Sản phẩm không tồn tại".
* **TC\_UpdateCart\_02**: Kiểm tra khi cập nhật số lượng sản phẩm và sản phẩm còn đủ số lượng trong kho. Kiểm thử đã pass với thông báo: "Cập nhật số lượng sản phẩm thành công!".
* **TC\_UpdateCart\_03**: Kiểm tra khi số lượng sản phẩm muốn cập nhật vượt quá số lượng trong kho. Kiểm thử đã pass với thông báo lỗi: "Sản phẩm không đủ số lượng. Chỉ còn lại ... sản phẩm".

**Function deleteFormCart() (Xóa sản phẩm khỏi giỏ hàng):**

* **TC\_DeleteCart\_01**: Kiểm tra khi xóa sản phẩm có ID không tồn tại. Kiểm thử đã pass với thông báo lỗi: "Sản phẩm không tồn tại".
* **TC\_DeleteCart\_02**: Kiểm tra khi xóa sản phẩm có ID hợp lệ. Kiểm thử đã pass với thông báo: "Xóa sản phẩm thành công".

**Function place\_order() (Thanh toán):**

* **TC\_Checkout\_01**: Kiểm tra khi chưa đăng nhập và cố gắng thanh toán. Kiểm thử đã pass với thông báo: "Vui lòng đăng nhập để mua hàng".
* **TC\_Checkout\_09**: Kiểm tra thanh toán khi ID giỏ hàng không tồn tại. Kiểm thử đã pass với thông báo lỗi: "Đặt hàng không thành công".
* **TC\_Checkout\_10**: Kiểm tra thanh toán khi thông tin cần thiết (như tên, email, địa chỉ, số điện thoại) đầy đủ và hợp lệ. Kiểm thử đã pass với thông báo: "Đặt hàng thành công!".
* **TC\_Checkout\_11**: Kiểm tra khi thông tin đơn hàng không hợp lệ. Kiểm thử đã pass và không hiển thị thông tin đơn hàng.
* **TC\_Checkout\_12**: Kiểm tra khi tổng tiền không hợp lệ. Kiểm thử đã pass và không hiển thị tổng tiền.

## 2. Integration Testing

**Integration Testing** (kiểm thử tích hợp) là một giai đoạn trong quy trình kiểm thử phần mềm, tập trung vào việc kiểm tra sự tương tác giữa các module hoặc thành phần trong hệ thống. Mục tiêu là đảm bảo rằng các module được tích hợp hoạt động đúng như mong đợi khi chúng làm việc cùng nhau.

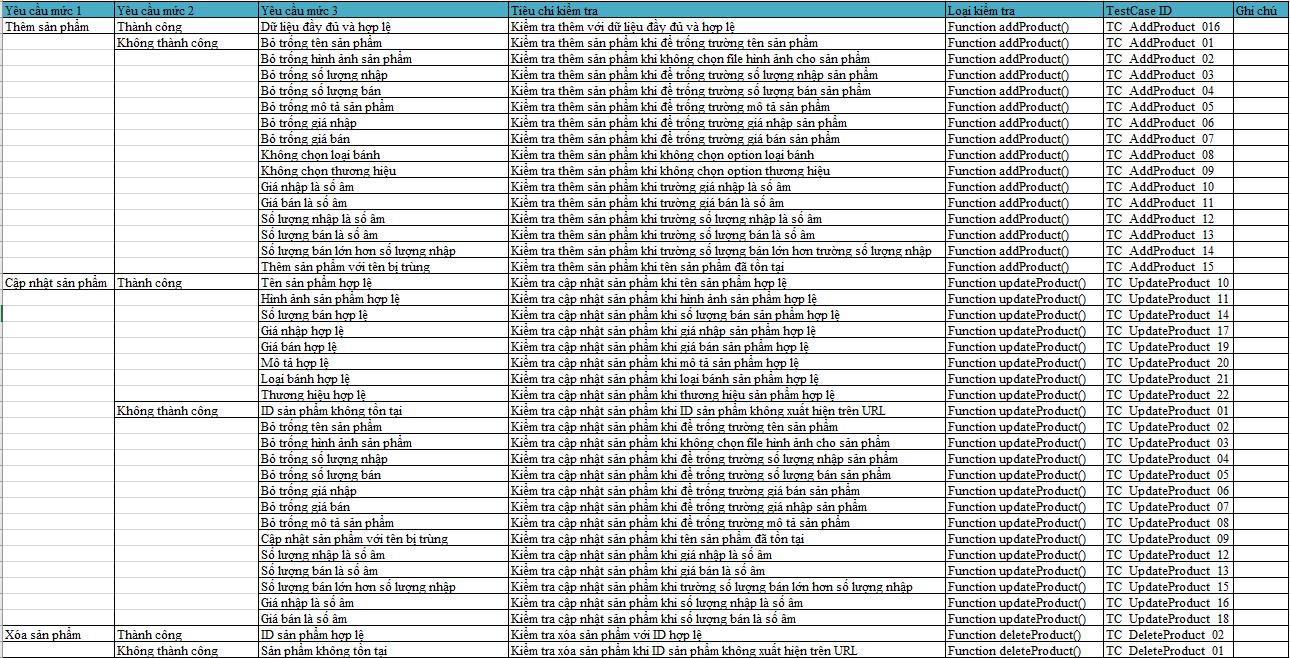
### 2.1. Test Design

**Test design** (Thiết kế kiểm thử) là quá trình tạo ra các kịch bản, trường hợp kiểm thử (test cases), và các phương pháp kiểm thử để kiểm tra một phần mềm hoặc hệ thống nhằm đảm bảo rằng hệ thống hoạt động đúng và đáp ứng các yêu cầu kỹ thuật, chức năng. Mục tiêu của test design là phát hiện lỗi (bugs) và đảm bảo chất lượng phần mềm.

**Mục Đích của Test Design:**

* **Đảm bảo chất lượng phần mềm**: Test design giúp xác định các phần của phần mềm cần kiểm tra để đảm bảo chúng hoạt động đúng như yêu cầu.
* **Phát hiện lỗi và sự cố**: Thiết kế kiểm thử có mục tiêu phát hiện các lỗi hoặc sự cố trong phần mềm trước khi nó được phát hành.
* **Giảm rủi ro**: Việc kiểm thử đầy đủ giúp giảm thiểu các rủi ro về chất lượng khi phần mềm được triển khai.
* **Đảm bảo đáp ứng yêu cầu người dùng**: Test design giúp xác minh rằng phần mềm hoạt động theo đúng các yêu cầu, từ đó nâng cao sự hài lòng của người dùng cuối.
* **Cải thiện hiệu suất phần mềm**: Các kiểm thử hiệu suất giúp kiểm tra khả năng chịu tải và xử lý của hệ thống dưới các điều kiện khắc nghiệt.

**Quy trình nhập hàng**



Hình 26. Test Design Nhập hàng

**Thêm sản phẩm**

**Thành công**: Kiểm tra chức năng thêm sản phẩm với dữ liệu hợp lệ, bao gồm tên, hình ảnh, số lượng, giá cả, mô tả, loại bánh, và thương hiệu.

**Không thành công**: Kiểm tra các trường hợp không hợp lệ như:

* Bỏ trống các trường quan trọng (tên sản phẩm, hình ảnh, số lượng, giá, mô tả, loại bánh, thương hiệu).
* Các giá trị âm cho số lượng hoặc giá.
* Số lượng bán lớn hơn số lượng nhập.
* Tên sản phẩm trùng lặp.

**Cập nhật sản phẩm**

**Thành công**: Kiểm tra chức năng cập nhật sản phẩm với các trường hợp hợp lệ như tên, hình ảnh, số lượng, giá nhập, giá bán, mô tả, loại bánh, và thương hiệu.

**Không thành công**: Kiểm tra các trường hợp không hợp lệ khi cập nhật:

* ID sản phẩm không tồn tại.
* Các trường quan trọng bị bỏ trống (tên, hình ảnh, số lượng, giá, mô tả).
* Các giá trị âm cho số lượng và giá.
* Số lượng bán lớn hơn số lượng nhập.
* Tên sản phẩm trùng lặp.

**Xóa sản phẩm**

**Thành công**: Kiểm tra việc xóa sản phẩm với ID hợp lệ.

**Không thành công**: Kiểm tra trường hợp xóa sản phẩm khi ID sản phẩm không tồn tại.

**Quy trình mua hàng**



Hình 27. Test Design Mua hàng

**Thêm sản phẩm vào giỏ hàng**

**Thành công**:

* Kiểm tra thêm sản phẩm khi còn đủ số lượng
* Kiểm tra thêm sản phẩm có sẵn trong giỏ hàng khi có đủ số lượng

**Không thành công:**

* Kiểm tra thêm sản phẩm khi sản phẩm không tồn tại
* Kiểm tra thêm sản phẩm vượt quá số lượng sản phẩm trong kho
* Kiểm tra thêm sản phẩm có sẵn trong giỏ hàng vượt quá số lượng sản phẩm trong kho

**Cập nhật sản phẩm số lượng trong kho**

**Thành công:** Kiểm tra cập nhật số lượng sản phẩm có trong giỏ

**Không thành công:**

* Kiểm tra cập nhật sản phẩm trong giỏ hàng khi sản phẩm không tồn tại
* Kiểm tra khi cập nhật số lượng vượt quá số lượng trong kho

**Xóa sản phẩm khỏi giỏ hàng**

**Thành công:** Kiểm tra xóa sản phẩm trong giỏ hàng không tồn tại ID

**Không thành công**: Kiểm tra xóa sản phẩm trong giỏ hàng xóa sản phẩm có ID hợp lệ

**Thanh toán giỏ hàng**

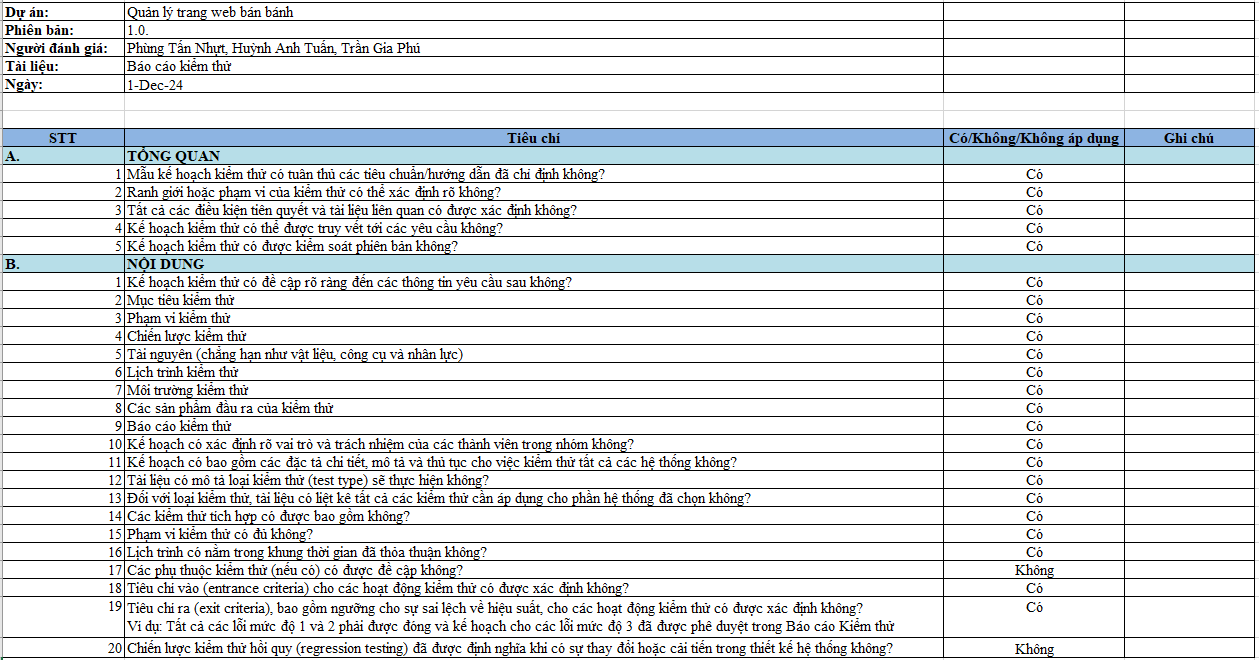
**Thành công**: Kiểm tra thanh toán giỏ hàng với dữ liệu đầy đủ và hợp lệ

**Không thành công:**

* Kiểm tra phải đăng nhập trước khi thanh toán
* Bỏ trống các trường quan trọng (họ và tên, email, địa chỉ, số điện thoại).
* Kiểm tra thanh toán khi trường họ và tên là số
* Kiểm tra thanh toán khi trường email không đúng định dạng
* Kiểm tra thanh toán khi số điện thoại ít hơn và lớn hơn 10 số
* Kiểm tra thanh toán khi giỏ hàng không tồn tại ID
* Kiểm tra thanh toán khi thông tin đơn hàng không hợp lệ
* Kiểm tra thanh toán khi tổng tiền đơn hàng không hợp lệ

### 2.2. Kiểm thử hộp trắng

**Checklist kiểm thử hộp trắng** (White-box Testing Checklist) là danh sách các mục cần kiểm tra nhằm đảm bảo chất lượng của mã nguồn và logic xử lý bên trong của một ứng dụng. Checklist này được sử dụng bởi người kiểm thử hoặc nhà phát triển để kiểm tra kỹ lưỡng các khía cạnh kỹ thuật trong quá trình kiểm thử hộp trắng.



Hình 28. CheckList Hộp Trắng

**Kiểm thử hộp trắng** (White Box Testing) là một phương pháp kiểm thử phần mềm tập trung vào việc kiểm tra logic bên trong và cấu trúc mã nguồn của ứng dụng để đảm bảo rằng các chức năng được triển khai đúng như mong đợi. Phương pháp này được sử dụng để:

-Xác minh xử lý sự kiện giao diện: Đảm bảo rằng các sự kiện được kích hoạt từ giao diện (như nhấn nút hoặc nhập liệu) được ánh xạ chính xác đến các hàm xử lý bên trong, và các bước xử lý (như lưu dữ liệu, cập nhật trạng thái) được thực hiện đúng logic.

-Kiểm tra luồng chuyển đổi trạng thái giao diện: Đảm bảo rằng logic chuyển đổi trạng thái giao diện, dựa trên các hành động của người dùng, được triển khai hợp lý, bao quát mọi trạng thái có thể xảy ra mà không tạo ra các trạng thái lỗi.

**Các kỹ thuật được sử dụng trong đồ án:**

-**Kiểm thử luồng điều khiển (Control Flow Testing)** là một phương pháp trong kiểm thử hộp trắng tập trung vào luồng thực thi logic của ứng dụng. Nó đảm bảo rằng:

* Tất cả các nhánh (branch) hoặc luồng (path) có thể xảy ra trong chương trình được kiểm tra.
* Các điều kiện và điểm quyết định (decision points) đều được xác minh.

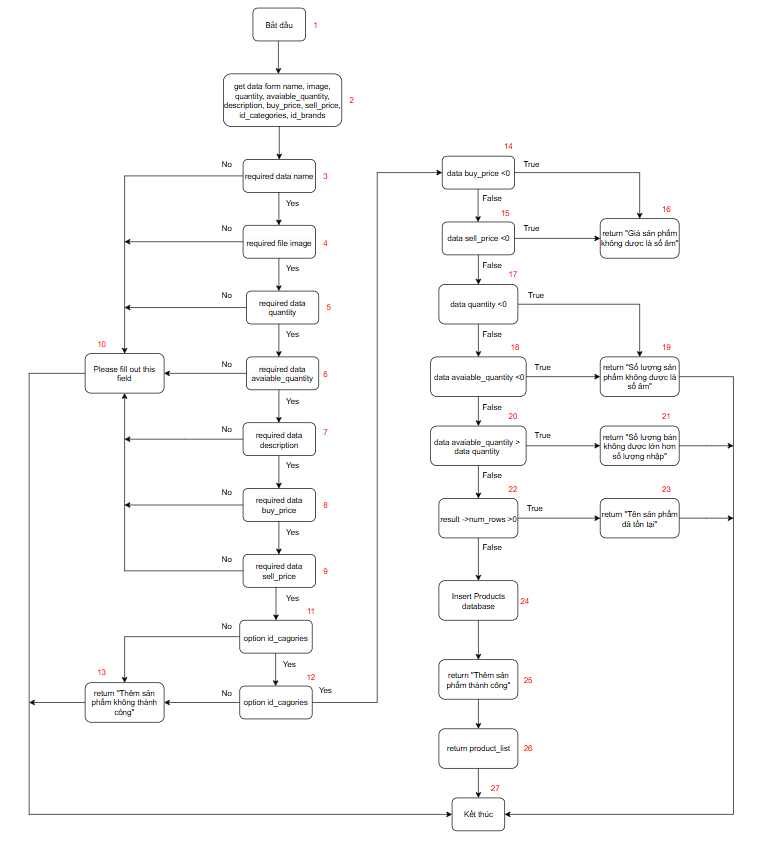
-**Kiểm thử branch coverage (bao phủ nhánh):** Kiểm tra các nhánh quyết định như "Thêm thành công" hay "Thêm không thành công."

-**Kiểm thử path coverage (bao phủ đường dẫn):** Kiểm tra các luồng logic khác nhau từ điểm bắt đầu đến điểm kết thúc, bao gồm các thông báo lỗi, xác nhận, và lưu cơ sở dữ liệu.

#### 2.2.1. Quy trình nhập hàng

**Thêm sản phẩm**

**Sơ đồ đồ thị dòng điều khiển**

****

Hình 29. Sơ đồ dòng điều khiển Thêm sản phẩm Nhập hàng

**Độ phức tạp**

E = số cạnh = 41

N = số nút = 27

P = số thành phần liên thông = 1

M = E – N + 2P = 41 – 27 + 2\*1 = 16

**Đường độc lập**

Đường độc lập 1: 1-2-3-10-27

Đường độc lập 2: 1-2-3-4-10-27

Đường độc lập 3: 1-2-3-4-5-10-27

Đường độc lập 4: 1-2-3-4-5-6-10-27

Đường độc lập 5: 1-2-3-4-5-6-7-10-27

Đường độc lập 6: 1-2-3-4-5-6-7-8-10-27

Đường độc lập 7: 1-2-3-4-5-6-7-8-9-10-27

Đường độc lập 8: 1-2-3-4-5-6-7-8-9-11-13-27

Đường độc lập 9: 1-2-3-4-5-6-7-8-9-11-12-13-27

Đường độc lập 10: 1-2-3-4-5-6-7-8-9-11-12-14-16-27

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

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

Đường độc lập 13: 1-2-3-4-5-6-7-8-9-11-12-14-15-17-18-19-27

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

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

Đường độc lập 16: 1-2-3-4-5-6-7-8-9-11-12-14-15-17-18-20-22-25-26-27

**Test Suite**

**TC\_AddProduct\_01: 1-2-3-10-27**

Input: name= “ ”

Kết quả kì vọng: Please fill out this field

**TC\_AddProduct\_02: 1-2-3-4-10-27**

Input: image= “ ”

Kết quả kì vọng: Please fill out this field

**TC\_AddProduct\_03: 1-2-3-4-5-10-27**

Input: quantity= “ ”

Kết quả kì vọng: Please fill out this field

**TC\_AddProduct\_04: 1-2-3-4-5-6-10-27**

Input: avaiable\_quantity= “ ”

Kết quả kì vọng: Please fill out this field

**TC\_AddProduct\_05: 1-2-3-4-5-6-7-10-27**

Input: description= “ ”

Kết quả kì vọng: Please fill out this field

**TC\_AddProduct\_06: 1-2-3-4-5-6-7-8-10-27**

Input: buy\_price= “ ”

Kết quả kì vọng: Please fill out this field

**TC\_AddProduct\_07: 1-2-3-4-5-6-7-8-9-10-27**

Input: sell\_price= “ ”

Kết quả kì vọng: Please fill out this field

**TC\_AddProduct\_08**: **1-2-3-4-5-6-7-8-9-11-13-27**

Input: id\_cagories= “Loại bánh”

Kết quả kì vọng: "Thêm sản phẩm không thành công"

**TC\_AddProduct\_09: 1-2-3-4-5-6-7-8-9-11-12-13-27**

Input: id\_brands= “Loại thương hiệu”

Kết quả kì vọng: "Thêm sản phẩm không thành công"

**TC\_AddProduct\_10:** **1-2-3-4-5-6-7-8-9-11-12-14-16-27**

Input: buy\_price = -1

Kết quả kì vọng: "Giá sản phẩm không được là số âm"

**TC\_AddProduct\_11: 1-2-3-4-5-6-7-8-9-11-12-14-15-16-27**

Input: sell\_price = -1

Kết quả kì vọng: "Giá sản phẩm không được là số âm"

**TC\_AddProduct\_12: 1-2-3-4-5-6-7-8-9-11-12-14-15-17-19-27**

Input: quantity = -1

Kết quả kì vọng: "Số lượng sản phẩm không được là số âm"

**TC\_AddProduct\_13: 1-2-3-4-5-6-7-8-9-11-12-14-15-17-18-19-27**

Input: avaiable\_quantity =-1

Kết quả kì vọng: "Số lượng sản phẩm không được là số âm"

**TC\_AddProduct\_14: 1-2-3-4-5-6-7-8-9-11-12-14-15-17-18-20-21-27**

Input: avaiable\_quantity = 10, quantity = 5

Kết quả kì vọng: "Số lượng bán không được lớn hơn số lượng nhập"

**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”

Kết quả kì vọng: “Tên sản phẩm đã tồn tại”

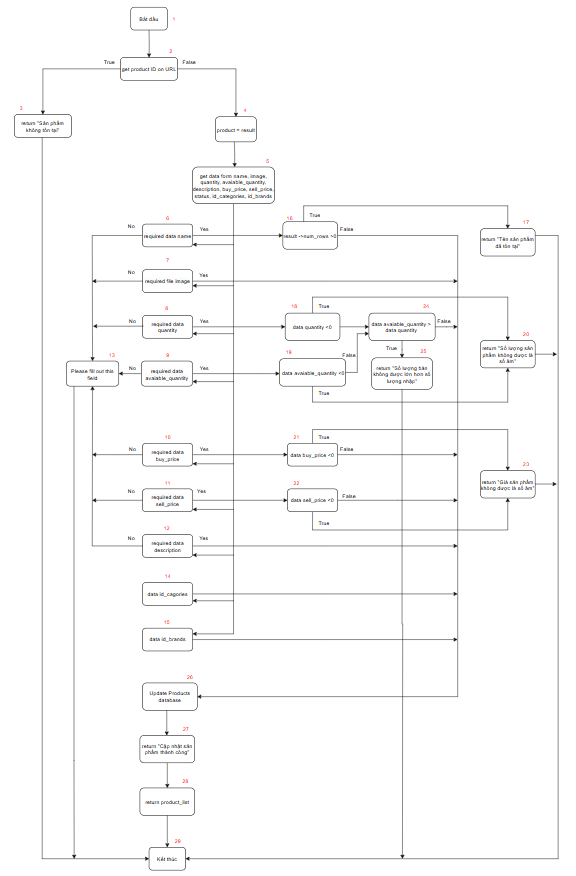
**TC\_AddProduct\_16: 1-2-3-4-5-6-7-8-9-11-12-14-15-17-18-20-22-25-26-27**

Input: name= “ABC”, image= “1aba114354.product-big-4.jpg”, quantity= 10, avaiable\_quantity =10, description= “Bánh ngon”, buy\_price = “100”, sell\_price = “120”, id\_cagories = “Cakes”, id\_brands = “SamSung”

Kết quả kì vọng: “Thêm sản phẩm thành công”

**Cập nhật sản phẩm**

**Sơ đồ đồ thị dòng điều khiển**

****

Hình 30. Sơ đồ dòng điều khiển Cập nhật sản phẩm

**Độ phức tạp**

E = số cạnh = 49

N = số nút = 29

P = số thành phần liên thông = 1

M = E – N + 2P = 49 – 29 + 2\*1 = 22

**Đường độc lập**

Đường độc lập 1: 1-2-3-29

Đường độc lập 2: 1-2-4-5-6-13-29

Đường độc lập 3: 1-2-4-5-7-13-29

Đường độc lập 4: 1-2-4-5-8-13-29

Đường độc lập 5: 1-2-4-5-9-13-29

Đường độc lập 6: 1-2-4-5-10-13-29

Đường độc lập 7: 1-2-4-5-11-13-29

Đường độc lập 8: 1-2-4-5-12-13-29

Đường độc lập 9: 1-2-4-5-6-16-17-29

Đường độc lập 10: 1-2-4-5-6-16-27-28-29

Đường độc lập 11: 1-2-4-5-7-27-28-29

Đường độc lập 12: 1-2-3-4-5-8-18-20-29

Đường độc lập 13: 1-2-3-4-5-9-19-20-29

Đường độc lập 14: 1-2-3-4-5-9-19-24-27-28-29

Đường độc lập 15: 1-2-3-4-5-9-19-24-25-29

Đường độc lập 16: 1-2-3-4-5-10-21-23-29

Đường độc lập 17: 1-2-3-4-5-10-21-27-28-29

Đường độc lập 18: 1-2-3-4-5-11-22-23-29

Đường độc lập 19: 1-2-3-4-5-11-22-27-28-29

Đường độc lập 20: 1-2-3-4-5-12-27-28-29

Đường độc lập 21: 1-2-3-4-5-14-27-28-29

Đường độc lập 22: 1-2-3-4-5-15-27-28-29

**Test Suite**

**TC\_UpdateProduct\_01: 1-2-3-29**

Input: id = 10

Kết quả kì vọng: “Sản phẩm không tồn tại”

**TC\_UpdateProduct\_02: 1-2-4-5-6-13-29**

Input: id = 1, name = “ ”

Kết quả kì vọng: “Please fill out this field”

**TC\_UpdateProduct\_03: 1-2-4-5-7-13-29**

Input: id = 1, image = “ ”

Kết quả kì vọng: “Please fill out this field”

**TC\_UpdateProduct\_04: 1-2-4-5-8-13-29**

Input: id = 1, quantity = “ ”

Kết quả kì vọng: “Please fill out this field”

**TC\_UpdateProduct\_05: 1-2-4-5-9-13-29**

Input: id = 1, avaiable\_quantity = “ ”

Kết quả kì vọng: “Please fill out this field”

**TC\_UpdateProduct\_06: 1-2-4-5-10-13-29**

Input: id = 1, buy\_price = “ ”

Kết quả kì vọng: “Please fill out this field”

**TC\_UpdateProduct\_07: 1-2-4-5-11-13-29**

Input: id = 1, sell\_price = “ ”

Kết quả kì vọng: “Please fill out this field”

**TC\_UpdateProduct\_08: 1-2-4-5-12-13-29**

Input: id = 1, description = “ ”

Kết quả kì vọng: “Please fill out this field”

**TC\_UpdateProduct\_09: 1-2-4-5-6-16-17-29**

Input: id = 1, name = “DULCE”

Kết quả kì vọng: “Tên sản phẩm đã tồn tại”

**TC\_UpdateProduct\_10: 1-2-4-5-6-16-27-28-29**

Input: id = 1, name = “ABC”

Kết quả kì vọng: “Cập nhật sản phẩm thành công”

**TC\_UpdateProduct\_11: 1-2-4-5-7-27-28-29**

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

Kết quả kì vọng: “Cập nhật sản phẩm thành công”

**TC\_UpdateProduct\_12: 1-2-3-4-5-8-18-20-29**

Input: id = 1, quantity = -1

Kết quả kì vọng: "Số lượng sản phẩm không được là số âm"

**TC\_UpdateProduct\_13: 1-2-3-4-5-9-19-20-29**

Input: id = 1, avaiable\_quantity =-1

Kết quả kì vọng: "Số lượng sản phẩm không được là số âm"

**TC\_UpdateProduct\_14: 1-2-3-4-5-9-19-24-27-28-29**

Input: id = 1, quantity = 10, avaiable\_quantity = 10

Kết quả kì vọng: “Cập nhật sản phẩm thành công”

**TC\_UpdateProduct\_15: 1-2-3-4-5-9-19-24-25-29**

Input: id = 1, quantity = 10, avaiable\_quantity = 20

Kết quả kì vọng: “Số lượng bán không được lớn hơn số lượng nhập”

**TC\_UpdateProduct\_16: 1-2-3-4-5-10-21-23-29**

Input: id = 1, buy\_price = -1

Kết quả kì vọng: "Giá sản phẩm không được là số âm"

**TC\_UpdateProduct\_17: 1-2-3-4-5-10-21-27-28-29**

Input: id = 1, buy\_price = 100

Kết quả kì vọng: “Cập nhật sản phẩm thành công”

**TC\_UpdateProduct\_18: 1-2-3-4-5-11-22-23-29**

Input: id = 1, sell\_price = -1

Kết quả kì vọng: "Giá sản phẩm không được là số âm"

**TC\_UpdateProduct\_19: 1-2-3-4-5-11-22-27-28-29**

Input: id = 1, sell\_price = 100

Kết quả kì vọng: “Cập nhật sản phẩm thành công”

**TC\_UpdateProduct\_20: 1-2-3-4-5-12-27-28-29**

Input: id = 1, description = “Bánh ngon”

Kết quả kì vọng: “Cập nhật sản phẩm thành công”

**TC\_UpdateProduct\_21: 1-2-3-4-5-14-27-28-29**

Input: id = 1, id\_cagories = “Cakes”

Kết quả kì vọng: “Cập nhật sản phẩm thành công”

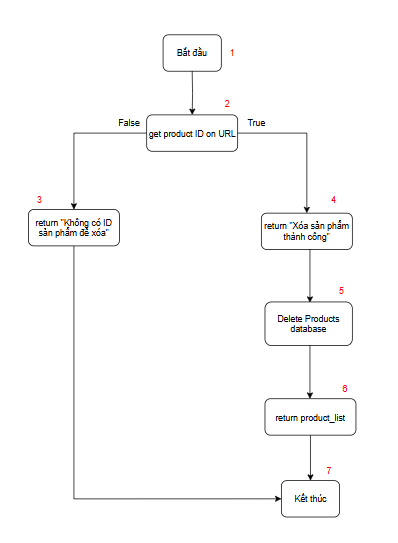
**TC\_UpdateProduct\_22: 1-2-3-4-5-15-27-28-29**

Input: id = 1¸id\_brands = “SamSung”

Kết quả kì vọng: “Cập nhật sản phẩm thành công”

**Xóa sản phẩm**

**Sơ đồ đồ thị dòng điều khiển**



Hình 31. Sơ đồ dòng điều khiểm Xóa sản phẩm Nhập hàng

**Độ phức tạp**

E = số cạnh = 7

N = số nút = 7

P = số thành phần liên thông = 1

M = E – N + 2P = 7-7 + 2\*1 = 2

**Đường độc lập**

Đường độc lập 1: 1-2-3-7

Đường độc lập 2: 1-2-4-5-6-7

**Test Suite**

**TC\_DeleteProduct\_01: 1-2-3-7**

Input: ID = 10

Kết quả kì vọng: “Không có ID sản phẩm để xóa”

**TC\_DeleteProduct\_02: 1-2-4-5-6-7**

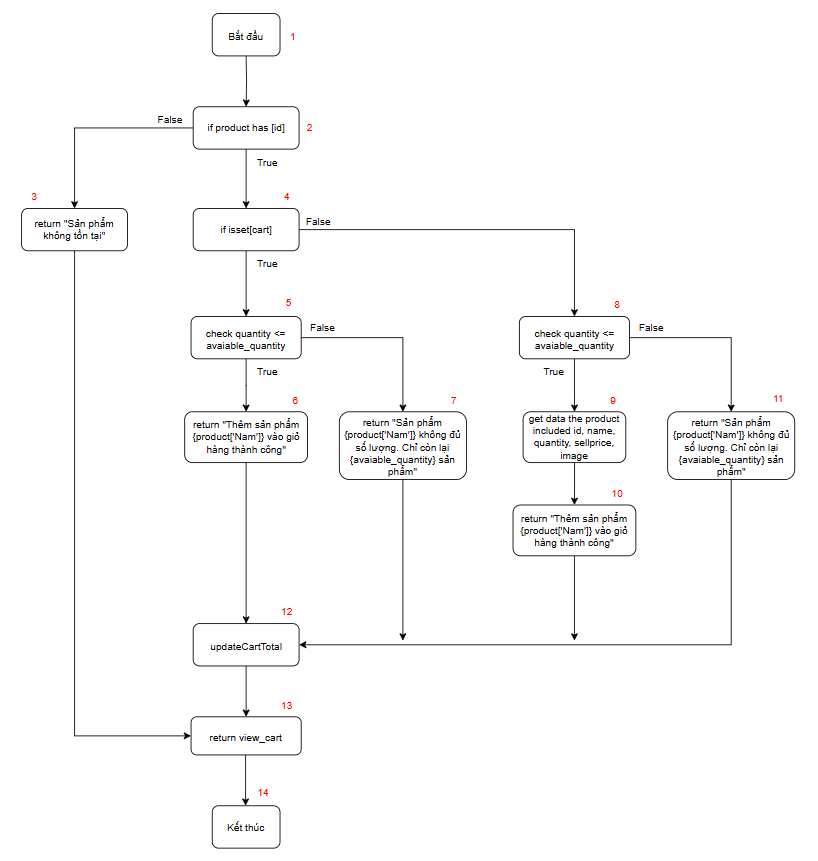
Input: ID = 1

Kết quả kì vọng: “Xóa sản phẩm thành công”

#### 2.2.2. Quy trình mua hàng

**Thêm sản phẩm vào giỏ hàng**

**Sơ đồ đồ thị dòng điều khiển**

****

Hình 32. Sơ đồ dòng điều khiểm Thêm sản phẩm Giỏ hàng

**Độ phức tạp**

E = số cạnh = 17

N = số nút = 14

P = số thành phần liên thông = 1

M = E – N + 2P = 18-15 + 2\*1 = 5

**Đường độc lập**

Đường độc lập 1: 1-2-3-14-15

Đường độc lập 2: 1-2-4-5-6-12-13-14

Đường độc lập 3: 1-2-4-5-7-12-13-14

Đường độc lập 4: 1-2-4-8-9-10-12-13-14

Đường độc lập 5: 1-2-4-8-11-12-13-14

**Test Suite**

**TC\_AddCart\_01: 1-2-3-13-14**

Input: ID = 10

Kết quả kì vọng: “Sản phẩm không tồn tại”

**TC\_AddCart\_02: 1-2-4-5-6-12-13-14**

Input: ID =1, quantity = 2, sell\_price = 120

Kết quả kì vọng: “Thêm sản phẩm {product['Nam']} vào giỏ hàng thành công”, tổng tiền 240USD

**TC\_AddCart\_03: 1-2-4-5-7-12-13-14**

Input: ID =1, quantity = 100, sell\_price = 120

Kết quả kì vọng: “Sản phẩm {product['Nam']} không đủ số lượng. Chỉ còn lại {avaiable\_quantity} sản phẩm”, tổng tiền 0USD

**TC\_AddCart\_04: 1-2-4-8-9-10-12-13-14**

Input: : ID =1, quantity = 2, sell\_price = 120, ĐK: cart has quatity = 1

Kết quả kì vọng: “Thêm sản phẩm {product['Nam']} vào giỏ hàng thành công”, tổng tiền 360USD

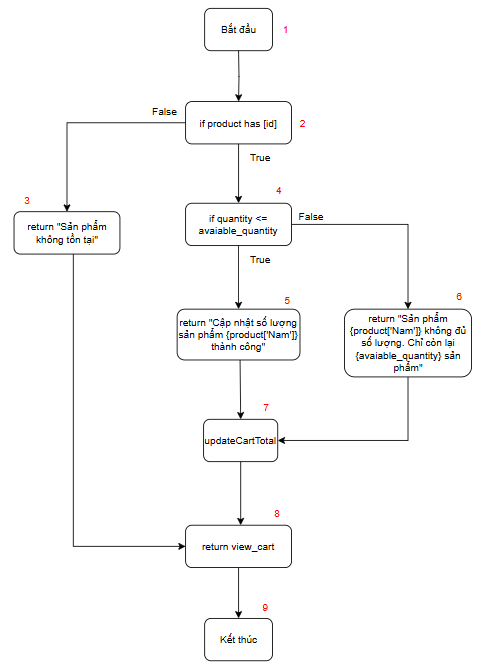
**TC\_AddCart\_05: 1-2-4-8-11-12-13-14**

Input: ID =1, quantity = 100, sell\_price = 120, ĐK: cart has quatity = 1

Kết quả kì vọng: “Sản phẩm {product['Nam']} không đủ số lượng. Chỉ còn lại {avaiable\_quantity} sản phẩm”, tổng tiền 120USD

**Cập nhật số lượng sản phẩm trong giỏ hàng**

**Sơ đồ đồ thị dòng điều khiển**

****

Hình 33. Sơ đồ dòng điều khiển Cập nhật Giỏ hàng

**Độ phức tạp**

E = số cạnh = 10

N = số nút = 9

P = số thành phần liên thông = 1

M = E – N + 2P = 10-19 + 2\*1 = 3

**Đường độc lập**

Đường độc lập 1: 1-2-3-8-9

Đường độc lập 2: 1-2-4-5-7-8-9

Đường độc lập 3: 1-2-4-6-7-8-9

**Test Suite**

**TC\_UpdateCart\_01: 1-2-3-8-9**

Input: ID = 10

Kết quả kì vọng: “Sản phẩm không tồn tại”

**TC\_UpdateCart\_02: 1-2-4-5-7-8-9**

Input: ID = 1, quantity = 5, sell\_price = 120, cart has quatity = 1

Kết quả kì vọng: “Cập nhật số lượng sản phẩm {product['Nam']} thành công”, tổng tiền 1100USD

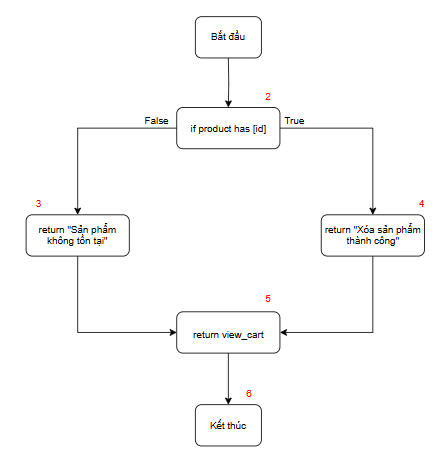
**TC\_UpdateCart\_03: 1-2-4-6-7-8-9**

Input: ID = 1, quantity = 100, sell\_price = 120, cart has quatity = 1

Kết quả kì vọng: “Sản phẩm {product['Nam']} không đủ số lượng. Chỉ còn lại {avaiable\_quantity} sản phẩm”, tổng tiền 120USD

**Xóa sản phẩm khỏi giỏ hàng**

**Sơ đồ đồ thị dòng điều khiển**



Hình 34. Sơ đồ dòng điều khiển Xóa sản phẩm Giỏ hàng

**Độ phức tạp**

E = số cạnh = 6

N = số nút = 6

P = số thành phần liên thông = 1

M = E – N + 2P = 6-6 + 2\*1 = 2

**Đường độc lập**

Đường độc lập 1: 1-2-3-5-6

Đường độc lập 2: 1-2-4-5-6

**Test Suite**

**TC\_DeleteCart\_01: 1-2-3-6**

Input: ID = 10

Kết quả kì vọng: “Sản phẩm không tồn tại”

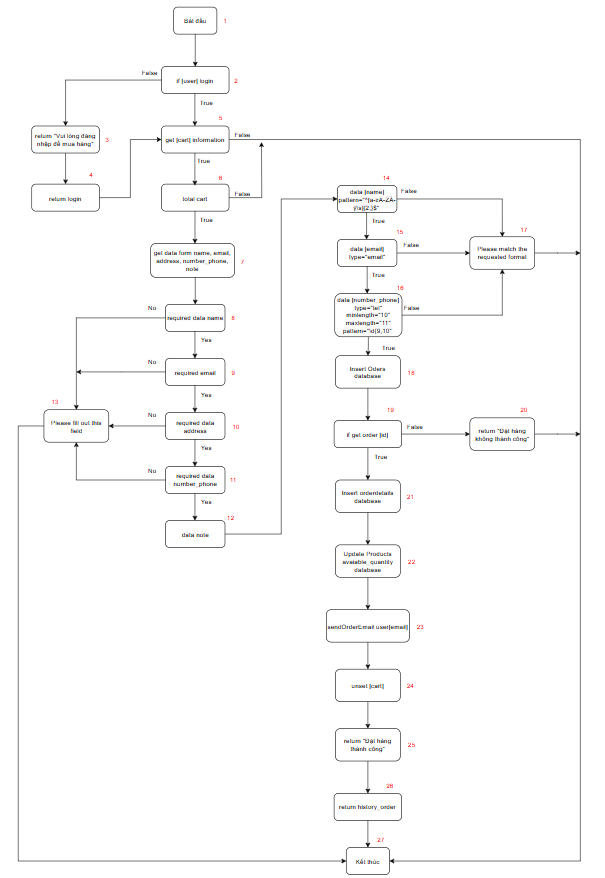
**TC\_DeleteCart\_02: 1-2-4-5-6**

Input: ID = 1

Kết quả kì vọng: “Xóa sản phẩm thành công”

**Thanh toán giỏ hàng**

**Sơ đồ đồ thị dòng điều khiển**



Hình 35. Sơ đồ dòng điều khiển Thanh toán Giỏ hàng

**Độ phức tạp**

E = số cạnh = 37

N = số nút = 27

P = số thành phần liên thông = 1

M = E – N + 2P = 35-27 + 2\*1 = 12

**Đường độc lập**

Đường độc lập 1: 1-2-3-4-5...

Đường độc lập 11: 1-2-5-27

Đường độc lập 12: 1-2-5-6-27

Đường độc lập 4: 1-2-5-6-7-8-13-27

Đường độc lập 5: 1-2-5-6-7-8-9-13-27

Đường độc lập 6: 1-2-5-6-7-8-9-10-13-27

Đường độc lập 7: 1-2-5-6-7-8-9-10-11-13-27

Đường độc lập 8: 1-2-5-6-7-8-9-10-11-12-14-17-27

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

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

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

Đường độc lập 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

Kết quả kì vọng: “Vui lòng đăng nhập để mua hàng”

**TC\_Checkout\_02: 1-2-5-27**

Input: ID = 1

Kết quả kì vọng: Không hiển thị thông tin đơn hàng

**TC\_Checkout\_03: 1-2-5-6-27**

Input: sell\_price = 120

Kết quả kì vọng: Không hiển thị tổng tiền đơn hàng

**TC\_Checkout\_04: 1-2-5-6-7-8-13-27**

Input: name= “ ”

Kết quả kì vọng: Please fill out this field

**TC\_Checkout\_05: 1-2-5-6-7-8-9-13-27**

Input: email= “ ”

Kết quả kì vọng: Please fill out this field

**TC\_Checkout\_06: 1-2-5-6-7-8-9-10-13-27**

Input: address= “ ”

Kết quả kì vọng: Please fill out this field

**TC\_Checkout\_07: 1-2-5-6-7-8-9-10-11-13-27**

Input: number\_phone= “ ”

Kết quả kì vọng: Please fill out this field

**TC\_Checkout\_08: 1-2-5-6-7-8-9-10-11-12-14-17-27**

Input: name= 123

Kết quả kì vọng: 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”

Kết quả kì vọng: 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"”

Kết quả kì vọng: 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

Kết quả kì vọng: “Đặt hàng không thành công”

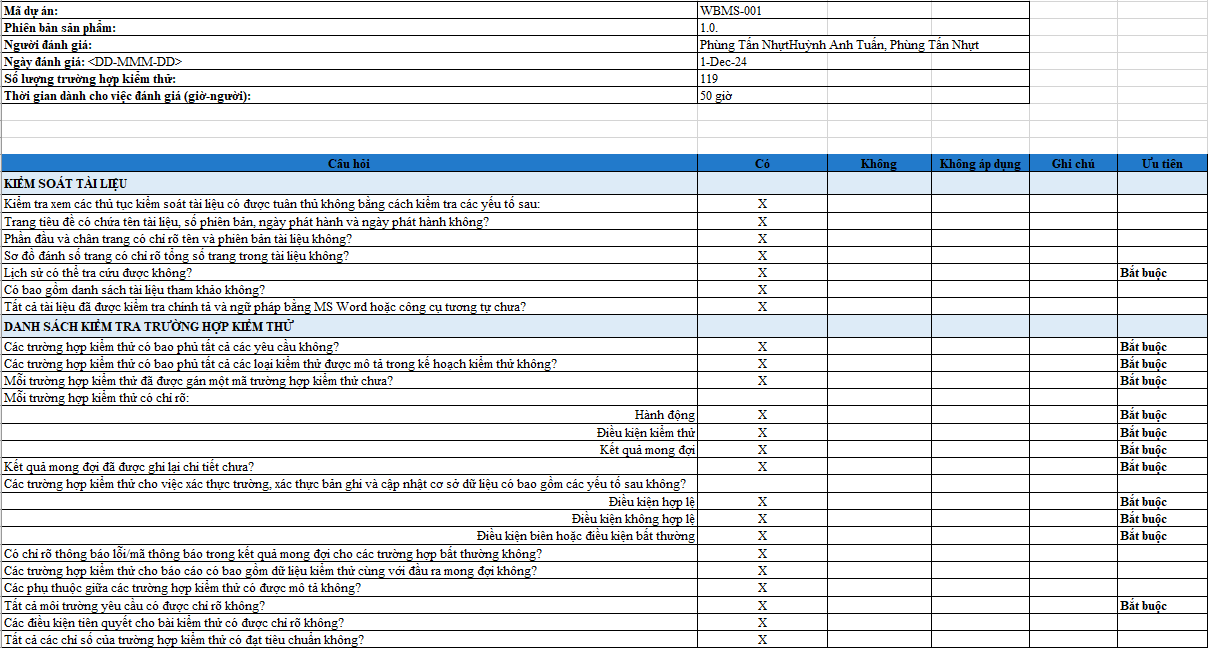
**TC\_Checkout\_12:1-2-5-6-7-8-9-10-11-12-14-15-16-18-19-21-22-23-24-25-26-27**

Input: name= “nhut”, email= “[tannhut2111@gmail.com](mailto:tannhut2111@gmail.com)”, address= “HCM”, number\_phone= “0901641800”, note= “”

Kết quả kì vọng: “Đặt hàng thành công”

### 2.3. Kiểm thử hộp đen

**Checklist kiểm thử hộp đen** (Black-box Testing Checklist)là danh sách các tiêu chí hoặc mục cần kiểm tra để đảm bảo rằng phần mềm hoạt động đúng theo yêu cầu và kỳ vọng mà không cần quan tâm đến cấu trúc mã nguồn hay logic bên trong. Checklist này tập trung vào kiểm thử các chức năng và giao diện người dùng của ứng dụng dựa trên các đầu vào và đầu ra.



Hình 36.CheckList Hộp Đen

**Kiểm thử hộp đen** (Black Box Testing) là một phương pháp kiểm thử phần mềm tập trung vào việc kiểm tra chức năng của hệ thống mà không quan tâm đến cấu trúc bên trong. Mục đích chính của kiểm thử hộp đen là đảm bảo rằng phần mềm hoạt động theo các yêu cầu đã định và xử lý đúng tất cả các tình huống đầu vào.

-Đảm bảo phần mềm đúng theo yêu cầu:

* Kiểm tra xem đầu ra có đúng với mong đợi dựa trên các đầu vào và yêu cầu chức năng không.
* Xác minh rằng các yêu cầu kinh doanh và kỹ thuật được đáp ứng.

-Phát hiện lỗi trong chức năng:

* Xử lý sai dữ liệu đầu vào.
* Tính toán sai.
* Thiếu chức năng hoặc xử lý không đúng theo yêu cầu.

-Tập trung vào hành vi bên ngoài của hệ thống:

* Không cần biết cách hoạt động bên trong của hệ thống, chỉ quan tâm đến kết quả.
* Phù hợp để người kiểm thử không có kinh nghiệm về lập trình thực hiện.

-Đảm bảo khả năng tương tác:

* Đảm bảo hệ thống tương tác đúng với các hệ thống khác hoặc với người dùng.

**Các kỹ thuật được sử dụng trong đồ án:**

-**Phương pháp Bảng quyết định** là một phương pháp kiểm thử hộp đen được sử dụng để kiểm tra sự kết hợp của các điều kiện đầu vào và hành động đầu ra. Phương pháp này rất hiệu quả khi hệ thống có các điều kiện phức tạp và cần kiểm tra tất cả các khả năng của chúng. Mỗi điều kiện đầu vào được kết hợp với các hành động đầu ra tương ứng để tạo thành một bảng quy tắc, từ đó giúp xác định các tình huống kiểm thử cần thiết. Gồm 3 phần chính:

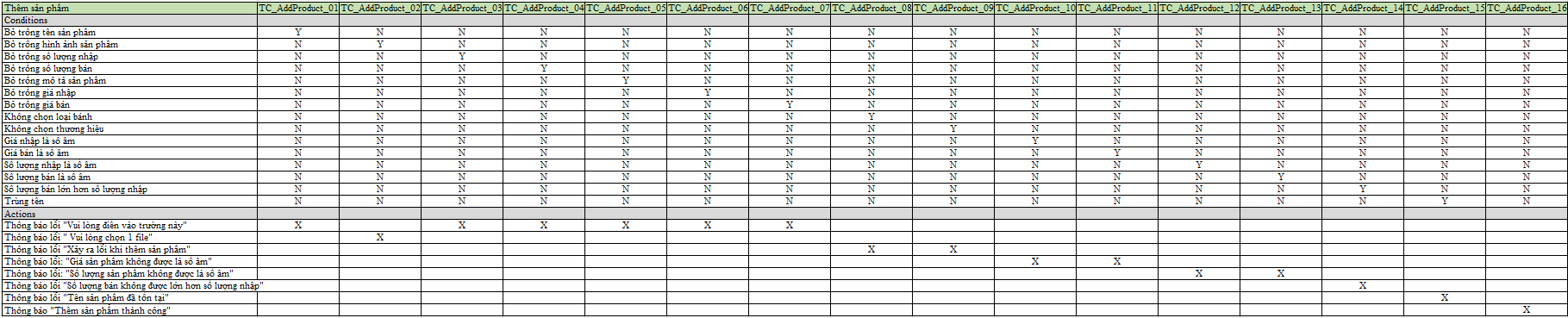
**Các điều kiện đầu vào (Conditions):** Đây là các yếu tố mà hệ thống kiểm tra. Mỗi điều kiện có thể có giá trị đúng (True) hoặc sai (False).

**Các hành động (Actions):** Các hành động cần thực hiện dựa trên các giá trị của điều kiện đầu vào. Các hành động này có thể bao gồm các thông báo lỗi, thành công, hoặc các thao tác khác trong hệ thống.

**Bảng quy tắc (Rules):** Mỗi quy tắc trong bảng quyết định mô tả một kết hợp cụ thể của điều kiện và hành động. Một bảng quyết định có thể có nhiều quy tắc, mỗi quy tắc kiểm tra một tình huống khác nhau.

#### 2.3.1. Quy trình nhập hàng

**Thêm sản phẩm**

****

Hình 37. Bảng quyết định Thêm sản phẩm

**Test Suite**

**TC\_AddProduct\_01**

Input: Tên sản phẩm = ""

Kết quả kì vọng: Please fill out this field

**TC\_AddProduct\_02**

Input: Hình ảnh = ""

Kết quả kì vọng: Please fill out this field

**TC\_AddProduct\_03**

Input: Số lượng nhập = ""

Kết quả kì vọng: Please fill out this field

**TC\_AddProduct\_04**

Input: Số lượng bán = ""

Kết quả kì vọng: Please fill out this field

**TC\_AddProduct\_05**

Input: Mô tả = ""

Kết quả kì vọng: Please fill out this field

**TC\_AddProduct\_06**

Input: Giá nhập = ""

Kết quả kì vọng: Please fill out this field

**TC\_AddProduct\_07**

Input: Giá bán = ""

Kết quả kì vọng: Please fill out this field

**TC\_AddProduct\_08**

Input: Loại bánh = ""

Kết quả kì vọng: "Thêm sản phẩm không thành công"

**TC\_AddProduct\_09**

Input: Thương hiệu = ""

Kết quả kì vọng: "Thêm sản phẩm không thành công"

**TC\_AddProduct\_10**

Input: Giá nhập = -1

Kết quả kì vọng: "Giá sản phẩm không được là số âm"

**TC\_AddProduct\_11**

Input: Giá bán = -1

Kết quả kì vọng: "Giá sản phẩm không được là số âm"

**TC\_AddProduct\_12**

Input: Số lượng nhập = -1

Kết quả kì vọng: "Số lượng sản phẩm không được là số âm"

**TC\_AddProduct\_13**

Input: Số lượng bán = -1

Kết quả kì vọng: "Số lượng sản phẩm không được là số âm"

**TC\_AddProduct\_14**

Input: Số lượng nhập = 100, Số lượng bán = 120

Kết quả kì vọng: "Số lượng bán không được lớn hơn số lượng nhập"

**TC\_AddProduct\_15**

Input: Tên sản phẩm = "DULCE"

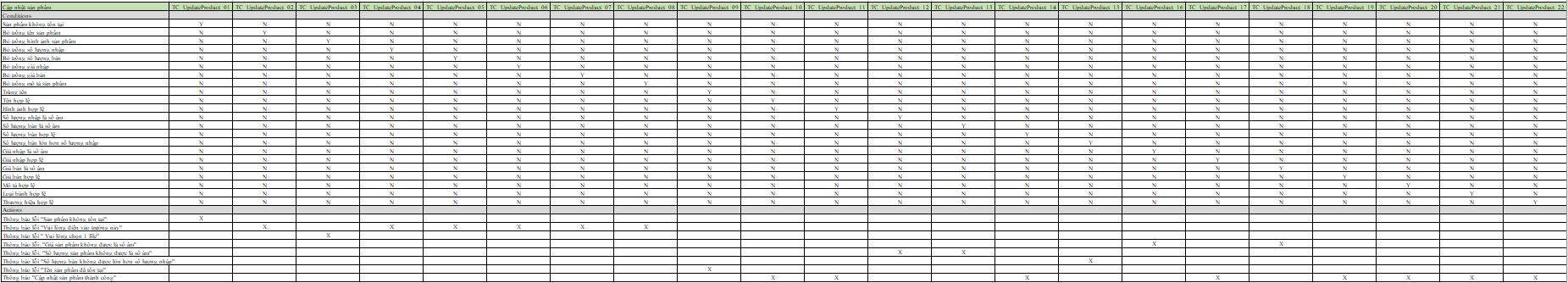
Kết quả kì vọng: “Tên sản phẩm đã tồn tại”

**TC\_AddProduct\_16**

Input: Tên sản phẩm = “ABC”, Hình ảnh = “1aba114354.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”

Kết quả kì vọng: “Thêm sản phẩm thành công”

**Cập nhật sản phẩm**



Hình 38. Bảng quyết định Cập nhật sản phẩm

**Test Suite**

**TC\_UpdateProduct\_01**

Input: ID = 10

Kết quả kì vọng: “Sản phẩm không tồn tại”

**TC\_UpdateProduct\_02**

Input: ID = 1, Tên sản phẩm = ""

Kết quả kì vọng: “Please fill out this field”

**TC\_UpdateProduct\_03**

Input: ID = 1, Hình ảnh = ""

Kết quả kì vọng: “Please fill out this field”

**TC\_UpdateProduct\_04**

Input: ID = 1, Số lượng nhập = ""

Kết quả kì vọng: “Please fill out this field”

**TC\_UpdateProduct\_05**

Input: ID = 1, Số lượng bán = ""

Kết quả kì vọng: “Please fill out this field”

**TC\_UpdateProduct\_06**

Input: ID = 1, Giá nhập = ""

Kết quả kì vọng: “Please fill out this field”

**TC\_UpdateProduct\_07**

Input: ID = 1, Giá bán = ""

Kết quả kì vọng: “Please fill out this field”

**TC\_UpdateProduct\_08**

Input: ID = 1, Mô tả = ""

Kết quả kì vọng: “Please fill out this field”

**TC\_UpdateProduct\_09**

Input: ID = 1, Tên sản phẩm = "DULCE"

Kết quả kì vọng: “Tên sản phẩm đã tồn tại”

**TC\_UpdateProduct\_10**

Input: ID 1, Tên sản phẩm = "ABC"

Kết quả kì vọng: “Cập nhật sản phẩm thành công”

**TC\_UpdateProduct\_11**

Input: ID = 1, Hình ảnh = “1aba114354.product-big-4.jpg”

Kết quả kì vọng: “Cập nhật sản phẩm thành công”

**TC\_UpdateProduct\_12**

Input: ID = 1, Số lượng nhập = -1

Kết quả kì vọng: "Số lượng sản phẩm không được là số âm"

**TC\_UpdateProduct\_13**

Input: ID = 1, Số lượng bán = -1

Kết quả kì vọng: "Số lượng sản phẩm không được là số âm"

**TC\_UpdateProduct\_14**

Input: ID = 1, Số lượng nhập = 100, Số lượng bán = 80

Kết quả kì vọng: “Cập nhật sản phẩm thành công”

**TC\_UpdateProduct\_15**

Input: ID = 1, Số lượng nhập = 100, Số lượng bán = 120

Kết quả kì vọng: “Số lượng bán không được lớn hơn số lượng nhập”

**TC\_UpdateProduct\_16**

Input: ID = 1, Giá nhập = -1

Kết quả kì vọng: "Giá sản phẩm không được là số âm"

**TC\_UpdateProduct\_17**

Input: ID = 1, Giá nhập = 100

Kết quả kì vọng: “Cập nhật sản phẩm thành công”

**TC\_UpdateProduct\_18**

Input: ID = 1, Giá bán = -1

Kết quả kì vọng: "Giá sản phẩm không được là số âm"

**TC\_UpdateProduct\_19**

Input: ID = 1, Giá bán = 100

Kết quả kì vọng: “Cập nhật sản phẩm thành công”

**TC\_UpdateProduct\_20**

Input: ID = 1, Mô tả = “Bánh ngon”

Kết quả kì vọng: “Cập nhật sản phẩm thành công”

**TC\_UpdateProduct\_21**

Input: ID = 1, Loại bánh = “Cakes”

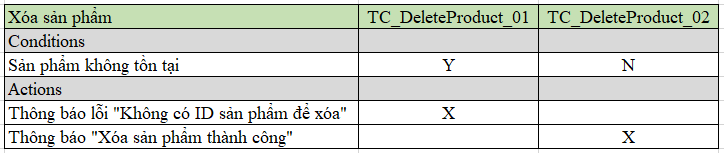
Kết quả kì vọng: “Cập nhật sản phẩm thành công”

**TC\_UpdateProduct\_22**

Input: ID = 1, Thương hiệu = “SamSung”

Kết quả kì vọng: “Cập nhật sản phẩm thành công”

**Xóa sản phẩm**



Hình 39. Bảng quyết định Xóa sản phẩm

**Test Suite**

**TC\_DeleteCart\_01**

Input: ID = 10

Kết quả kì vọng: “Không có ID sản phẩm để xóa”

**TC\_DeleteCart\_02**

Input: ID = 1

Kết quả kì vọng: “Xóa sản phẩm thành công”

#### 2.3.2. Quy trình mua hàng

**Thêm sản phẩm vào giỏ hàng**



Hình 40. Bảng quyết định Thêm sản phẩm vào Giỏ hàng

**Test Suite**

**TC\_AddCart\_01**

Input: ID = 10

Kết quả kì vọng: “Sản phẩm không tồn tại”

**TC\_AddCart\_02**

Input: ID =1, Số lượng = 2, Giá = 120USD

Kết quả kì vọng: “Thêm sản phẩm...vào giỏ hàng thành công”, tổng tiền là 120USD

**TC\_AddCart\_03**

Input: ID =1, Số lượng = 100, Giá = 120USD

Kết quả kì vọng: “Sản phẩm...không đủ số lượng. Chỉ còn lại...sản phẩm”, tổng tiền 0USD

**TC\_AddCart\_04**

Input: ID =1, Số lượng = 2, Giá = 120USD, ĐK: Giỏ hàng có số lượng = 1

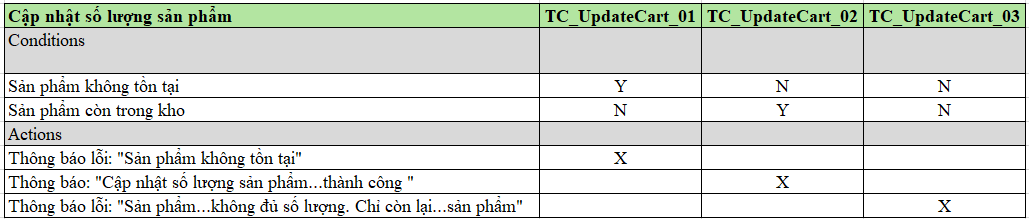
Kết quả kì vọng: “Thêm sản phẩm...vào giỏ hàng thành công”, tổng tiền 360USD

**TC\_AddCart\_05**

Input: ID =1, Số lượng = 100, Giá = 120USD, ĐK: Giỏ hàng có số lượng = 1

Kết quả kì vọng: “Sản phẩm...không đủ số lượng. Chỉ còn lại...sản phẩm”, tổng tiền 120USD

**Cập nhật sản phẩm trong giỏ hàng**



Hình 41. Bảng quyết định Cập nhật số lượng trong Giỏ hàng

**Test Suite**

**TC\_UpdateCart\_01**

Input: ID = 10

Kết quả kì vọng: “Sản phẩm không tồn tại”

**TC\_UpdateCart\_02**

Input: Số lượng = 5, Giỏ hàng có số lượng = 1

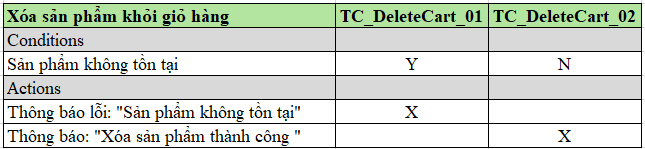
Kết quả kì vọng: “Cập nhật số lượng sản phẩm...thành công”, tổng tiền 1100USD

**TC\_UpdateCart\_03**

Input: Số lượng = 100, Giỏ hàng có số lượng = 1

Kết quả kì vọng: “Sản phẩm...không đủ số lượng. Chỉ còn lại...sản phẩm”, tổng tiền 120USD

**Xóa sản phẩm khỏi giỏ hàng**



Hình 42. Bảng quyết định Xóa sản phẩm khỏi Giỏ hàng

**Test Suite**

**TC\_DeleteCart\_01**

Input: ID = 10

Điều kiện: Sản phẩm không tồn tại (N)

Kết quả kì vọng: “Sản phẩm không tồn tại”

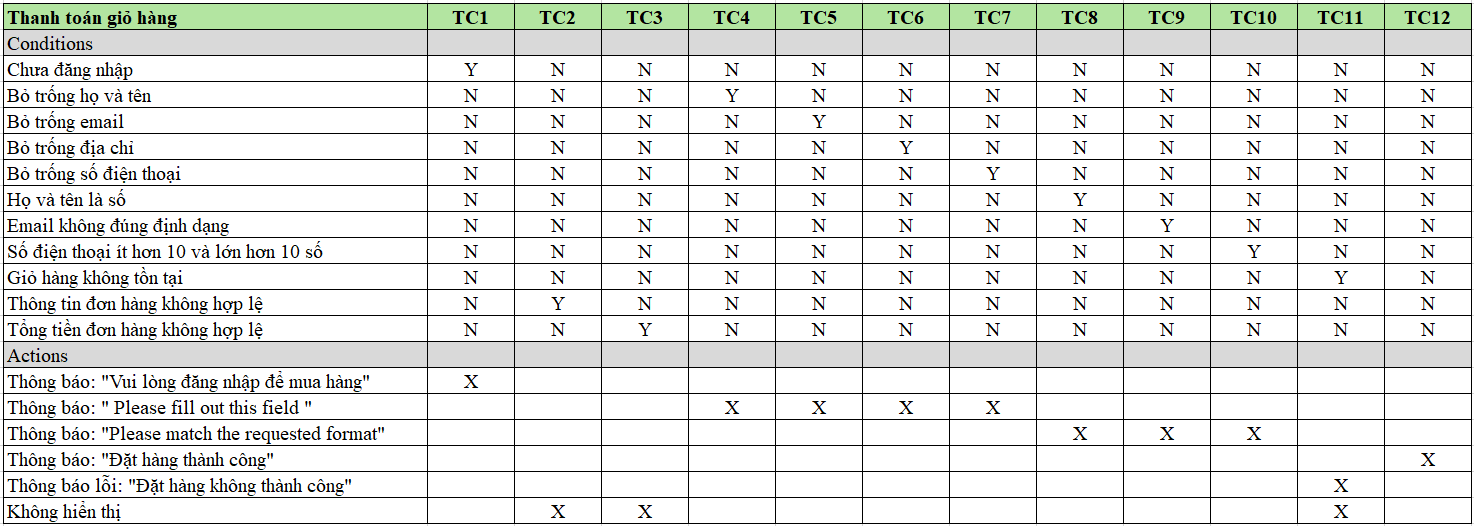
**TC\_DeleteCart\_02**

Input: ID = 1

Điều kiện: Sản phẩm không tồn tại (Y)

Kết quả kì vọng: “Xóa sản phẩm thành công”

**Thanh toán giỏ hàng**



Hình 43. Bảng quyết định Thanh toán Giỏ hàng

**Test Suite**

**TC\_Checkout\_01**

Input: user no login

Kết quả kì vọng: “Vui lòng đăng nhập để mua hàng”

**TC\_Checkout\_02**

Input: ID = 1

Kết quả kì vọng: Không hiển thị thông tin đơn hàng

**TC\_Checkout\_03**

Input: Giá = 120USD

Kết quả kì vọng: Không hiển thị tổng tiền đơn hàng

**TC\_Checkout\_04**

Input: Họ và tên = " "

Kết quả kì vọng: Please fill out this field

**TC\_Checkout\_05**

Input: Email = " "

Kết quả kì vọng: Please fill out this field

**TC\_Checkout\_06**

Input: Địa chỉ = " "

Kết quả kì vọng: Please fill out this field

**TC\_Checkout\_07**

Input: Số điện thoại = " "

Kết quả kì vọng: Please fill out this field

**TC\_Checkout\_08**

Input: Họ và tên = "1"

Kết quả kì vọng: Please match the requested format

**TC\_Checkout\_09**

Input: Email = "tannhut2111"

Kết quả kì vọng: Please match the requested format

**TC\_Checkout\_10**

Input: Số điện thoại = "090164180"

Kết quả kì vọng: Please match the requested format

**TC\_Checkout\_11**

Input: ID = 10

Kết quả kì vọng: “Đặt hàng không thành công”

**TC\_Checkout\_12**

Input: Họ và tên = “nhut”, Email = "tannhut2111@gmail.com", Địa chỉ = “HCM”, Số điện thoại = “0901641800”, Ghi chú = “”

Kết quả kì vọng: “Đặt hàng không thành công”

## 3. System Testing

**Test Case** (trường hợp kiểm thử) là một tập hợp các điều kiện, dữ liệu đầu vào, hành động thực hiện, và kết quả mong đợi được thiết kế để kiểm tra một chức năng cụ thể của phần mềm hoặc hệ thống. Mỗi test case mô tả chi tiết cách kiểm tra và xác định phần mềm có đáp ứng yêu cầu hoặc tiêu chí kiểm thử hay không.

**Mục đích của Test Case**

**Đảm bảo chất lượng phần mềm**: Test case giúp kiểm tra rằng phần mềm hoạt động đúng như yêu cầu.

**Phát hiện lỗi**: Xác định các lỗi hoặc khiếm khuyết trong hệ thống để sửa chữa.

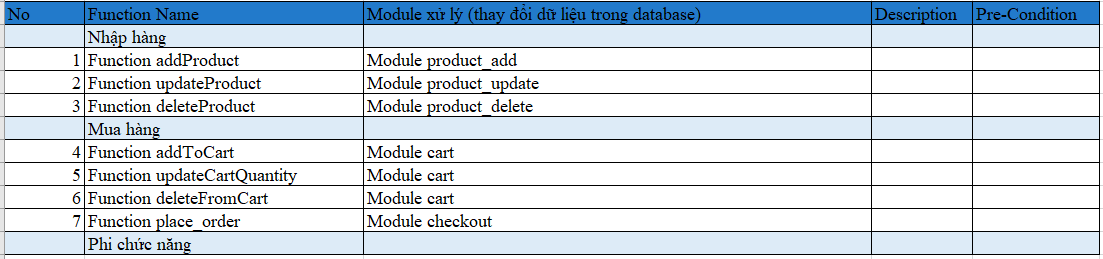
**Hướng dẫn kiểm thử**: Test case cung cấp hướng dẫn chi tiết cho người kiểm thử thực hiện kiểm tra một cách nhất quán.

**Tái sử dụng**: Các test case có thể được sử dụng lại trong các lần kiểm thử tiếp theo hoặc các dự án tương tự.

**Hỗ trợ lập tài liệu**: Test case là một tài liệu tham khảo quan trọng trong quá trình phát triển và bảo trì phần mềm.

**Xác minh yêu cầu**: Test case giúp đảm bảo rằng phần mềm đáp ứng các yêu cầu kỹ thuật và chức năng đã đề ra.

### 3.1. TestCase List



Hình 44. Danh sách test case

**Nhập hàng**

* **Function addProduct**
  + **Module xử lý**: product\_add
  + **Mô tả**: Chức năng này cho phép người dùng thêm một sản phẩm mới vào cơ sở dữ liệu. Thông tin cần nhập bao gồm tên sản phẩm, hình ảnh, số lượng, giá cả, mô tả, loại bánh, và thương hiệu.
  + **Pre-Condition**: Cần có quyền truy cập vào module nhập hàng và dữ liệu sản phẩm hợp lệ.
* **Function updateProduct**
  + **Module xử lý**: product\_update
  + **Mô tả**: Chức năng cập nhật thông tin sản phẩm đã có trong hệ thống, bao gồm các trường như tên, hình ảnh, số lượng, giá nhập, giá bán, mô tả, loại bánh, và thương hiệu.
  + **Pre-Condition**: Sản phẩm cần được xác định và có ID hợp lệ trước khi thực hiện cập nhật.
* **Function deleteProduct**
  + **Module xử lý**: product\_delete
  + **Mô tả**: Chức năng xóa một sản phẩm khỏi hệ thống bằng cách sử dụng ID sản phẩm. Sau khi xóa, sản phẩm sẽ không còn tồn tại trong cơ sở dữ liệu.
  + **Pre-Condition**: Sản phẩm phải tồn tại trong hệ thống và có ID hợp lệ để có thể xóa.

**Mua hàng**

* **Function addToCart**
  + **Module xử lý**: cart
  + **Mô tả**: Chức năng thêm sản phẩm vào giỏ hàng của người dùng. Người dùng có thể thêm nhiều sản phẩm vào giỏ, và hệ thống sẽ lưu trữ thông tin sản phẩm đó trong giỏ hàng.
  + **Pre-Condition**: Người dùng phải đăng nhập và có quyền thêm sản phẩm vào giỏ hàng. Sản phẩm cần có đủ số lượng trong kho.
* **Function updateCartQuantity**
  + **Module xử lý**: cart
  + **Mô tả**: Chức năng cập nhật số lượng sản phẩm trong giỏ hàng của người dùng. Người dùng có thể thay đổi số lượng các sản phẩm đã có trong giỏ.
  + **Pre-Condition**: Giỏ hàng đã có sản phẩm cần thay đổi số lượng. Sản phẩm phải có đủ số lượng trong kho.
* **Function deleteFromCart**
  + **Module xử lý**: cart
  + **Mô tả**: Chức năng xóa sản phẩm khỏi giỏ hàng. Người dùng có thể xóa bất kỳ sản phẩm nào trong giỏ hàng của mình.
  + **Pre-Condition**: Giỏ hàng phải có sản phẩm để xóa và người dùng phải có quyền thao tác trên giỏ hàng của mình.
* **Function place\_order**
  + **Module xử lý**: checkout
  + **Mô tả**: Chức năng thanh toán giỏ hàng. Người dùng sẽ thực hiện thanh toán khi đã hoàn tất giỏ hàng, và hệ thống sẽ xử lý đơn hàng, trừ tiền và lưu thông tin giao dịch.
  + **Pre-Condition**: Người dùng cần đăng nhập, giỏ hàng không được trống, và các thông tin thanh toán phải đầy đủ và hợp lệ.

**Phi chức năng**

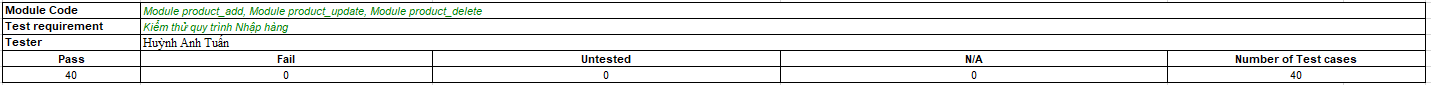
Xử lý đồng thời số lượng lớn người dùng truy cập

Xác thực 2 yếu tố khi đăng nhập

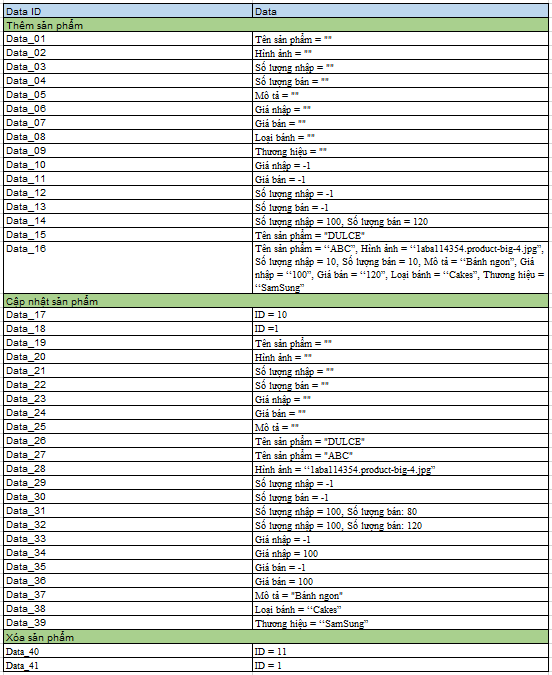
Bảo vệ chống SQL Injection

Mã hóa kết nối bằng HTTPS

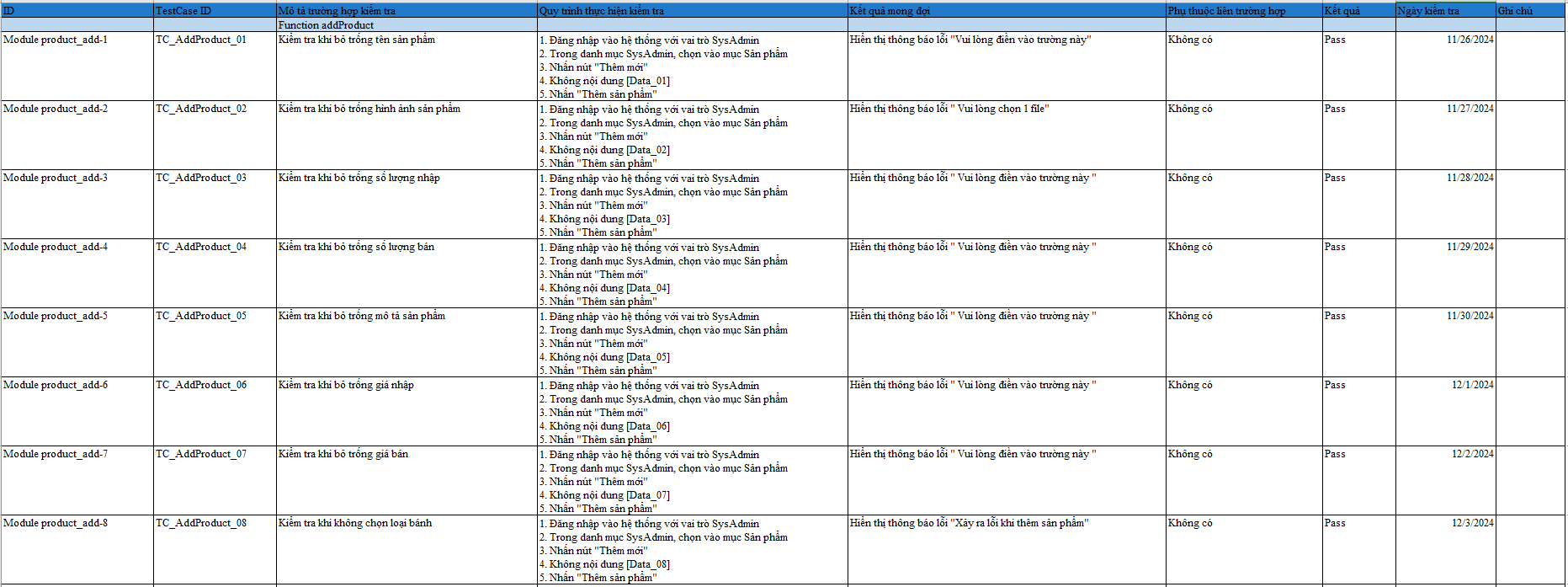
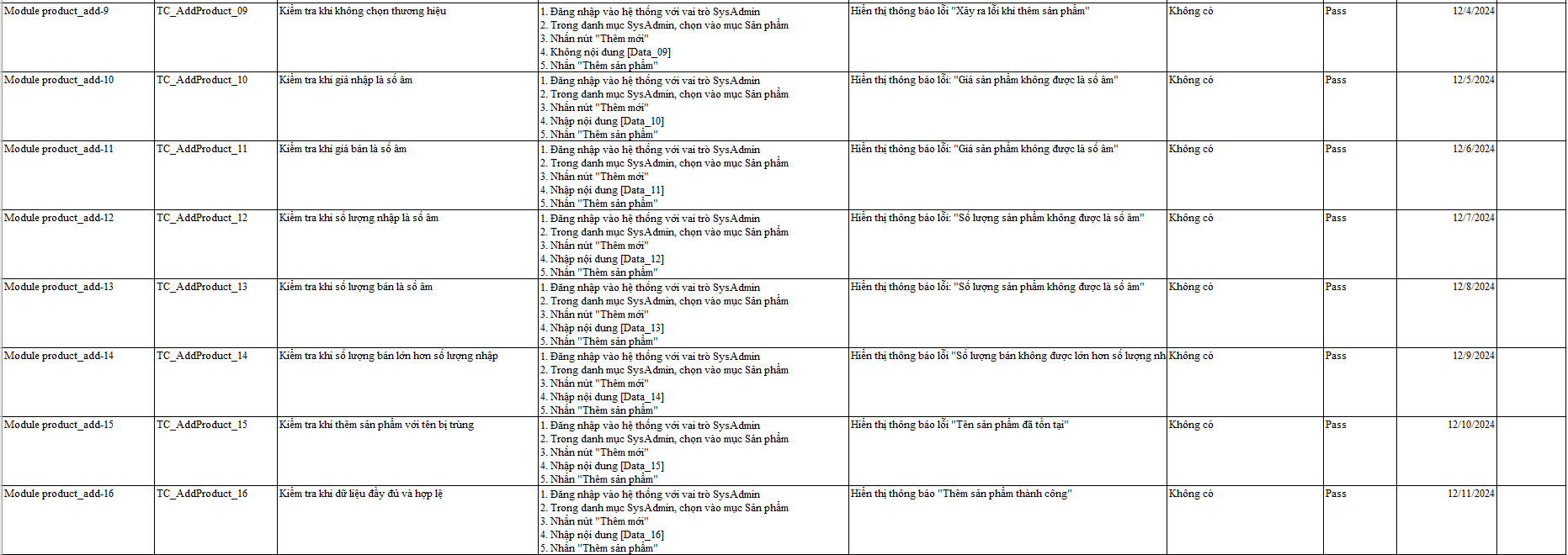
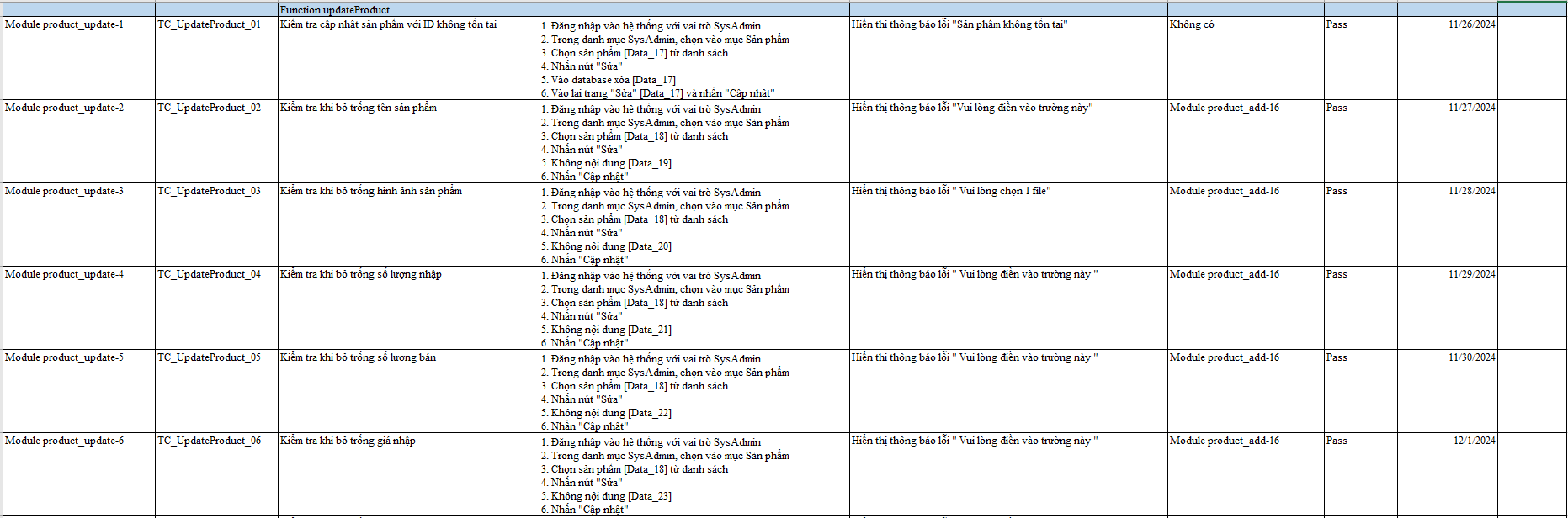
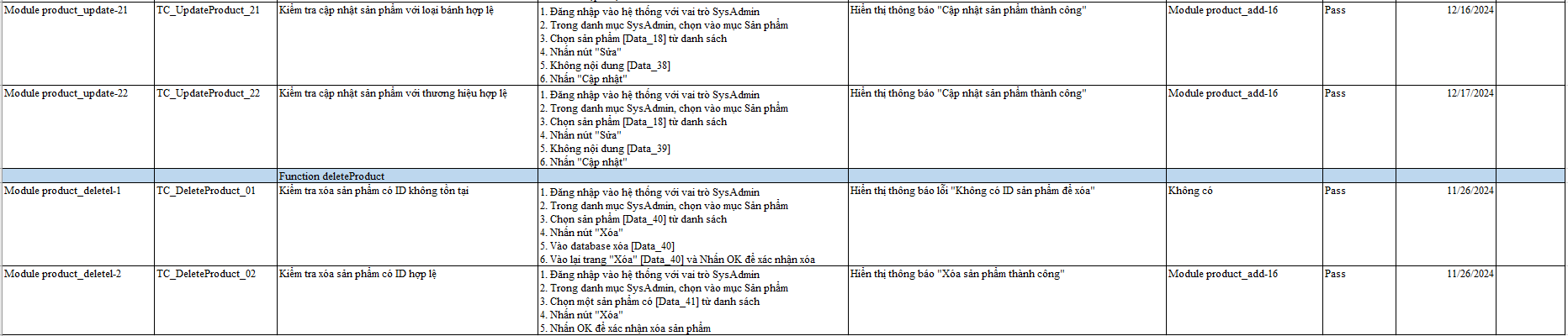
### 3.2. Quy trình nhập hàng



Hình 45. Module và kết quả kiểm thử cho quy trình nhập hàng



Hình 46.DaTa Test cho quy trình nhập hàng

Hình 47. TestCase quy trình nhập hàng

**ID**: Chỉ số định danh cho trường hợp kiểm tra.

**TestCase ID**: Mã ID của trường hợp kiểm tra, dùng để theo dõi và tham chiếu.

**Mô tả trường hợp kiểm tra**: Nội dung mô tả về mục đích của trường hợp kiểm tra.

**Quy trình thực hiện kiểm tra**: Các bước chi tiết mà người kiểm thử sẽ thực hiện để kiểm tra tính năng.

**Kết quả mong đợi**: Hệ thống sẽ hiển thị thông báo lỗi yêu cầu người dùng điền đầy đủ trường tên sản phẩm.

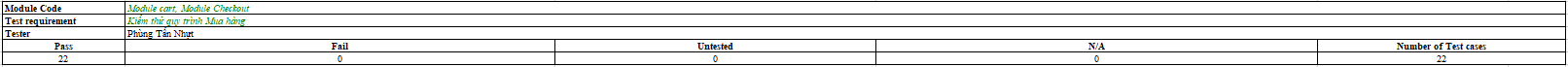
**Phụ thuộc liên trường hợp**: Nếu trường hợp kiểm tra này phụ thuộc vào trường hợp kiểm tra khác, thông tin này sẽ được ghi lại

**Kết quả**: Thực tế kết quả của trường hợp kiểm tra, xác định xem hệ thống có hành động đúng như mong đợi hay không.

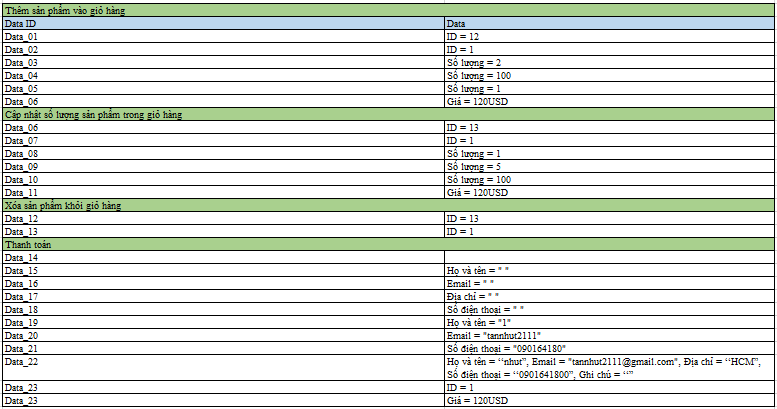
**Ngày kiểm tra**: Ngày thực hiện kiểm tra.

**Ghi chú**: Các ghi chú thêm về quá trình kiểm tra, chẳng hạn như các vấn đề phát sinh hoặc các điểm cần chú ý.

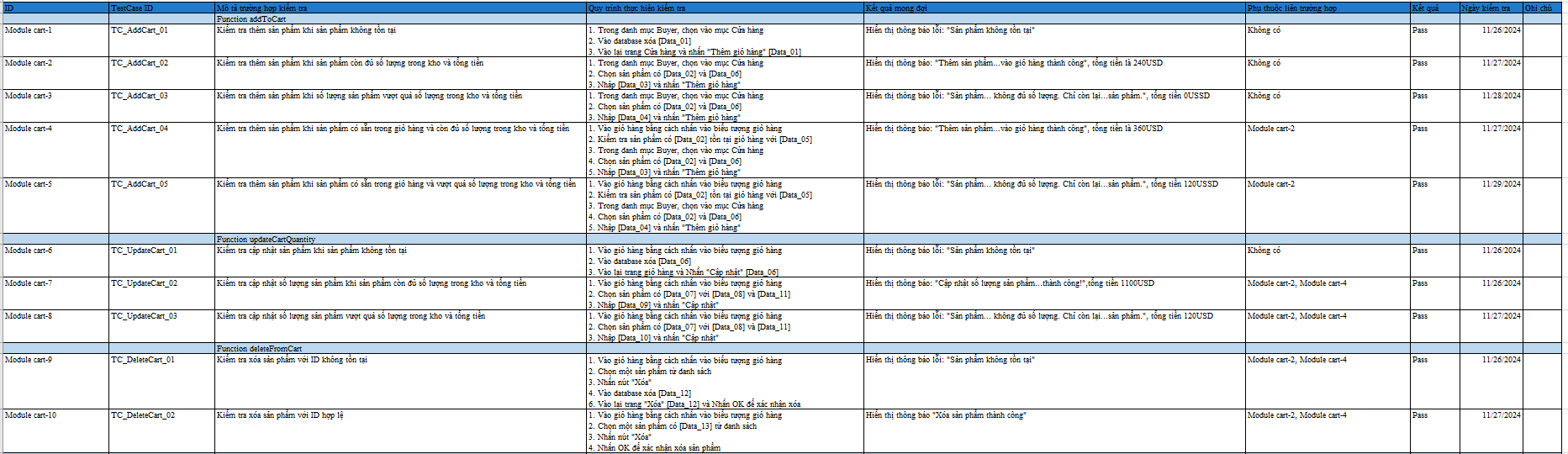
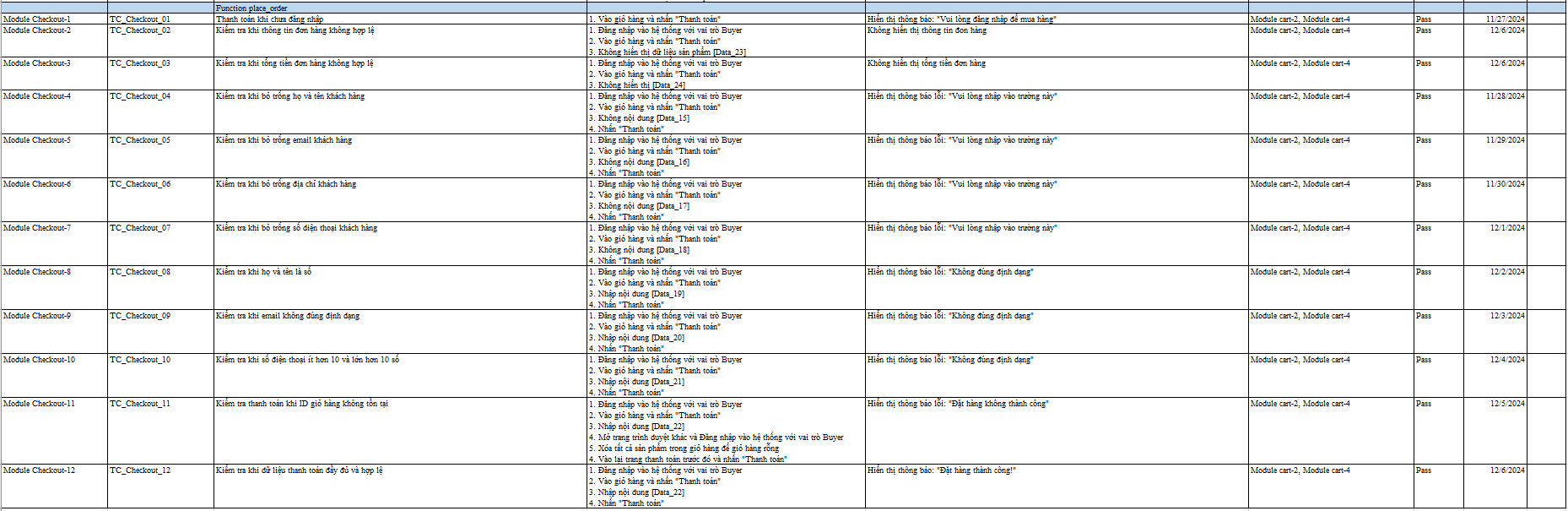
### 3.3. Quy trình mua hàng



Hình 48. Module và kết quả kiểm thử cho quy trình mua hàng



Hình 49. Data Test cho quy trình mua hàng

Hình 50. TestCae quy trình mua hàng

**ID**: Chỉ số định danh cho trường hợp kiểm tra.

**TestCase ID**: Mã ID của trường hợp kiểm tra, dùng để theo dõi và tham chiếu.

**Mô tả trường hợp kiểm tra**: Nội dung mô tả về mục đích của trường hợp kiểm tra.

**Quy trình thực hiện kiểm tra**: Các bước chi tiết mà người kiểm thử sẽ thực hiện để kiểm tra tính năng.

**Kết quả mong đợi**: Hệ thống sẽ hiển thị thông báo lỗi yêu cầu người dùng điền đầy đủ trường tên sản phẩm.

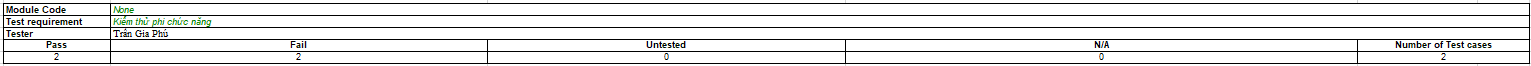
**Phụ thuộc liên trường hợp**: Nếu trường hợp kiểm tra này phụ thuộc vào trường hợp kiểm tra khác, thông tin này sẽ được ghi lại

**Kết quả**: Thực tế kết quả của trường hợp kiểm tra, xác định xem hệ thống có hành động đúng như mong đợi hay không.

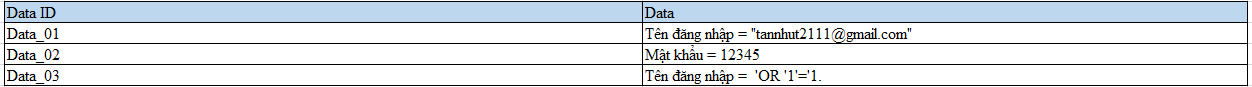
**Ngày kiểm tra**: Ngày thực hiện kiểm tra.

**Ghi chú**: Các ghi chú thêm về quá trình kiểm tra, chẳng hạn như các vấn đề phát sinh hoặc các điểm cần chú ý.

### 3.4. Phi chức năng



Hình 51. Kết quả kiểm thử cho phi chức năng



Hình 52. Data Test cho phi chức năng



Hình 53. TestCase phi chức năng

**ID**: Chỉ số định danh cho trường hợp kiểm tra.

**TestCase ID**: Mã ID của trường hợp kiểm tra, dùng để theo dõi và tham chiếu.

**Mô tả trường hợp kiểm tra**: Nội dung mô tả về mục đích của trường hợp kiểm tra.

**Quy trình thực hiện kiểm tra**: Các bước chi tiết mà người kiểm thử sẽ thực hiện để kiểm tra tính năng.

**Kết quả mong đợi**: Hệ thống sẽ hiển thị thông báo lỗi yêu cầu người dùng điền đầy đủ trường tên sản phẩm.

**Phụ thuộc liên trường hợp**: Nếu trường hợp kiểm tra này phụ thuộc vào trường hợp kiểm tra khác, thông tin này sẽ được ghi lại

**Kết quả**: Thực tế kết quả của trường hợp kiểm tra, xác định xem hệ thống có hành động đúng như mong đợi hay không.

**Ngày kiểm tra**: Ngày thực hiện kiểm tra.

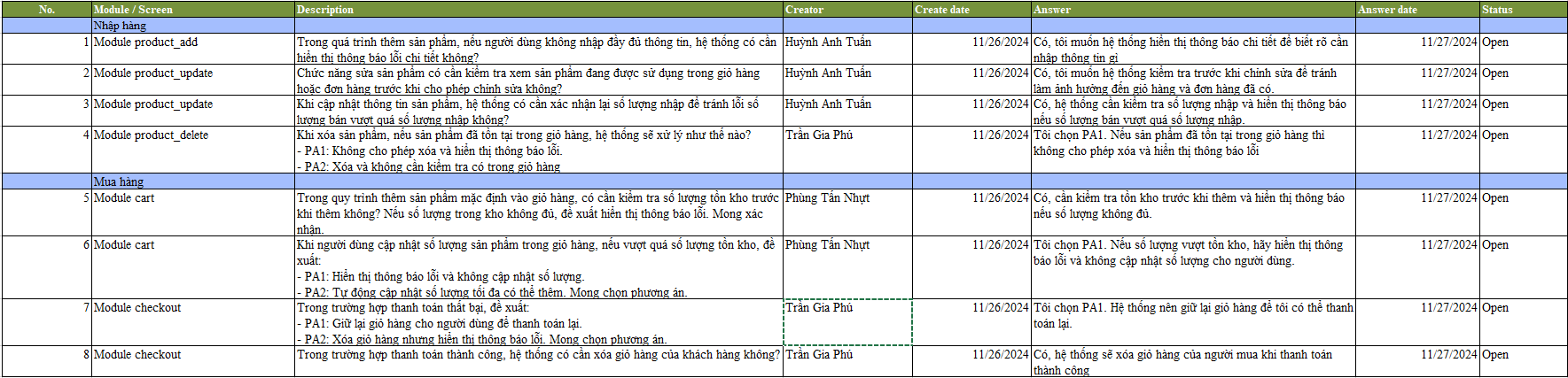
**Ghi chú**: Các ghi chú thêm về quá trình kiểm tra, chẳng hạn như các vấn đề phát sinh hoặc các điểm cần chú ý.

## 4. Acceptance Testing

**Acceptance Testing (Kiểm thử chấp nhận)** là giai đoạn kiểm thử cuối cùng trước khi phần mềm được triển khai hoặc phát hành. Mục đích chính của nó là để đảm bảo rằng phần mềm đáp ứng được tất cả các yêu cầu nghiệp vụ (business requirements) và có thể hoạt động theo cách mà khách hàng hoặc người dùng cuối mong muốn.

**Mục tiêu của Acceptance Testing**

* Xác minh rằng phần mềm đáp ứng yêu cầu nghiệp vụ:
* Kiểm tra xem phần mềm có hoàn thành các chức năng và tính năng mà khách hàng mong đợi không.
* Đảm bảo tính sẵn sàng để triển khai (Go-Live):
* Đảm bảo phần mềm không có lỗi lớn và có thể hoạt động ổn định trong môi trường thực tế.



Hình 54. Q&A

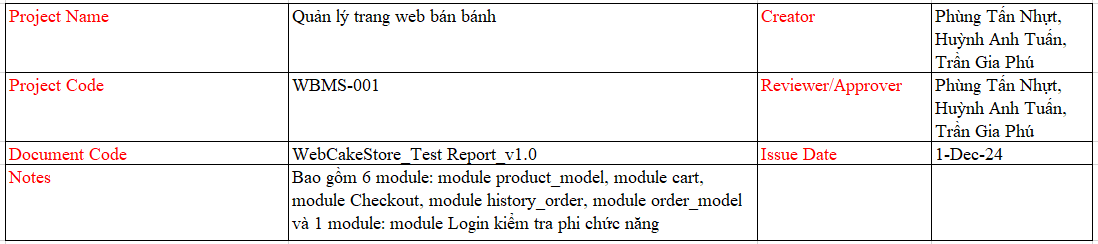
# CHƯƠNG 5: KẾT QUẢ KIỂM THỬ

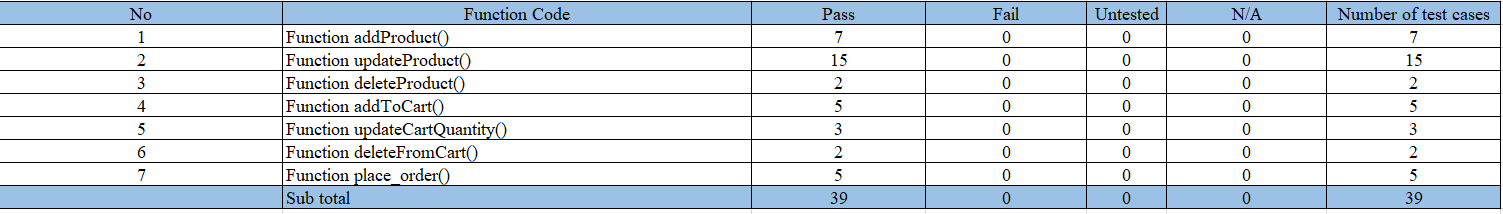
## 1. Test Report

**Test Report** (Báo cáo kiểm thử) là một tài liệu tổng hợp các kết quả và phát hiện từ quá trình kiểm thử phần mềm. Báo cáo này được tạo ra để cung cấp thông tin chi tiết về các hoạt động kiểm thử đã được thực hiện, những lỗi hoặc vấn đề phát hiện được, và đánh giá chung về chất lượng của phần mềm.

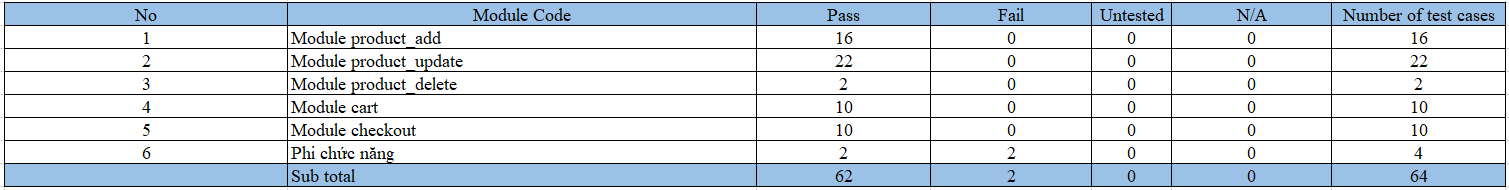
**Mục đích của Test Report:**

* Cung cấp thông tin về trạng thái của phần mềm: Báo cáo giúp các bên liên quan (như nhà phát triển, quản lý, khách hàng) hiểu được trạng thái hiện tại của phần mềm, các lỗi còn tồn tại, và mức độ đáp ứng yêu cầu.
* Hỗ trợ quyết định phát hành: Test report cung cấp cơ sở để quyết định xem phần mềm có sẵn sàng được phát hành hay không.
* Lưu trữ tài liệu: Báo cáo kiểm thử giúp lưu lại lịch sử các lần kiểm thử, giúp tham khảo trong tương lai và đối chiếu khi cần thiết





Hình 55. TestReport các TestCase (UnitTest

Hình 56. TestReport các TestCase (hộp trắng và hộp đen)

## 2. Bug Report

**Báo cáo lỗi 1: Thiếu xác thực OTP**

**Tiêu đề:** Thiếu xác thực OTP khi đăng nhập

**Mô tả:**  
Khi thực hiện đăng nhập vào hệ thống, bước xác thực OTP bị thiếu. Người dùng có thể đăng nhập mà không cần nhập mã OTP, điều này làm giảm mức độ an toàn của hệ thống.

**Các bước thực hiện:**

1. Truy cập trang đăng nhập của hệ thống.
2. Nhập tên đăng nhập hợp lệ vào trường "Tên đăng nhập".
3. Nhập mật khẩu hợp lệ vào trường "Mật khẩu".
4. Nhấn nút "Đăng nhập".

**Kết quả mong đợi:**  
Hệ thống phải yêu cầu người dùng nhập mã OTP để hoàn tất đăng nhập. Chỉ khi mã OTP hợp lệ, người dùng mới có thể đăng nhập thành công.

**Kết quả thực tế:**  
Hệ thống không hiển thị bước xác thực OTP. Người dùng có thể đăng nhập mà không cần nhập mã OTP.

**Môi trường:**

* Ngày kiểm thử: 26/11/2024
* Trình duyệt: [Google Chrome, Microsoft Edge]
* Hệ điều hành: [Windows]

**Tái tạo được:** Luôn xảy ra

**Mức độ nghiêm trọng:** Nghiêm trọng

**Mức độ ưu tiên:** 1 (Cao)

**Giải pháp:** Mới

**Báo cáo lỗi 2: Thiếu mã hóa HTTPS**

**Tiêu đề:** Hệ thống không bắt buộc mã hóa HTTPS

**Mô tả:**  
Hệ thống không bắt buộc sử dụng kết nối HTTPS, cho phép người dùng truy cập thông qua kết nối HTTP không được mã hóa. Điều này gây rủi ro lớn vì thông tin nhạy cảm có thể bị đánh cắp khi truyền qua mạng.

**Các bước thực hiện:**

1. Mở trình duyệt và nhập URL của hệ thống
2. Quan sát giao thức hiển thị trên thanh địa chỉ của trình duyệt.
3. Thử truy cập hệ thống bằng HTTP thay vì HTTPS.

**Kết quả mong đợi:**  
Tất cả các kết nối đến hệ thống phải được bảo mật bằng HTTPS. Nếu người dùng truy cập qua HTTP, họ phải được tự động chuyển hướng sang HTTPS.

**Kết quả thực tế:**  
Hệ thống cho phép truy cập qua HTTP mà không tự động chuyển hướng sang HTTPS, khiến kết nối không được mã hóa.

**Môi trường:**

* Ngày kiểm thử: 28/11/2024
* Trình duyệt: [Google Chrome, Microsoft Edge]]
* Hệ điều hành: [Windows]

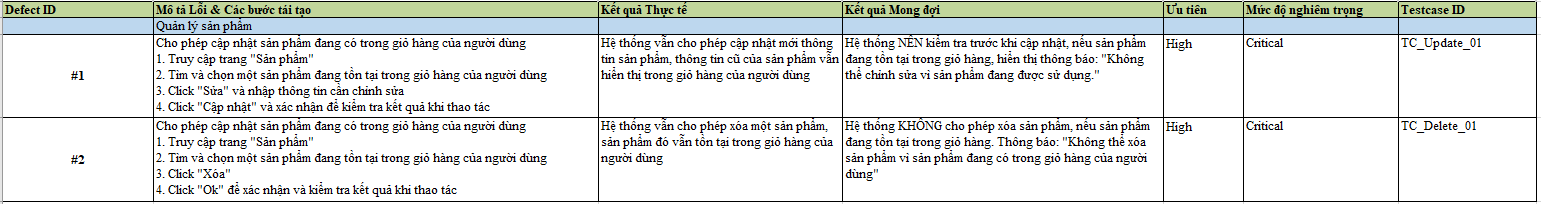
**Tái tạo được:** Luôn xảy ra

**Mức độ nghiêm trọng:** Cao

**Mức độ ưu tiên:** 2 (Trung bình)

**Giải pháp:** Mới

## 3. DefectList



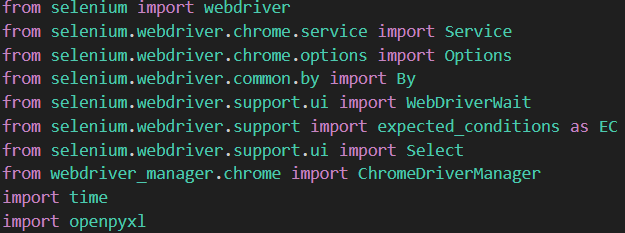
Hình 57. DefectList

## 4. Auto Test

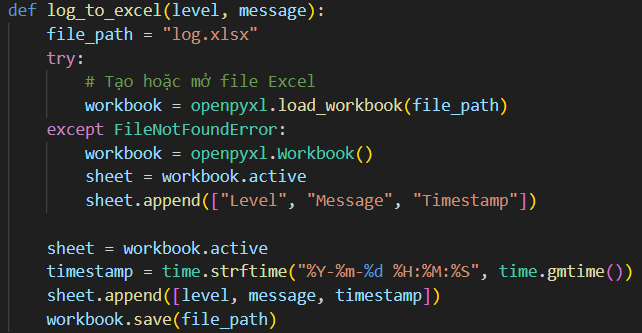
Auto test (kiểm thử tự động) là quá trình sử dụng các công cụ hoặc script để tự động hóa việc kiểm tra phần mềm nhằm đảm bảo chất lượng và hiệu suất của sản phẩm. Thay vì kiểm tra thủ công, auto test giúp thực hiện các trường hợp kiểm thử nhanh chóng, chính xác, và lặp lại được, từ đó tiết kiệm thời gian và giảm sai sót do con người gây ra.

Auto test thường được sử dụng trong các giai đoạn phát triển phần mềm, đặc biệt với các loại kiểm thử như chức năng, hồi quy, và hiệu năng. Công cụ được sử dụng phổ biến Selenium

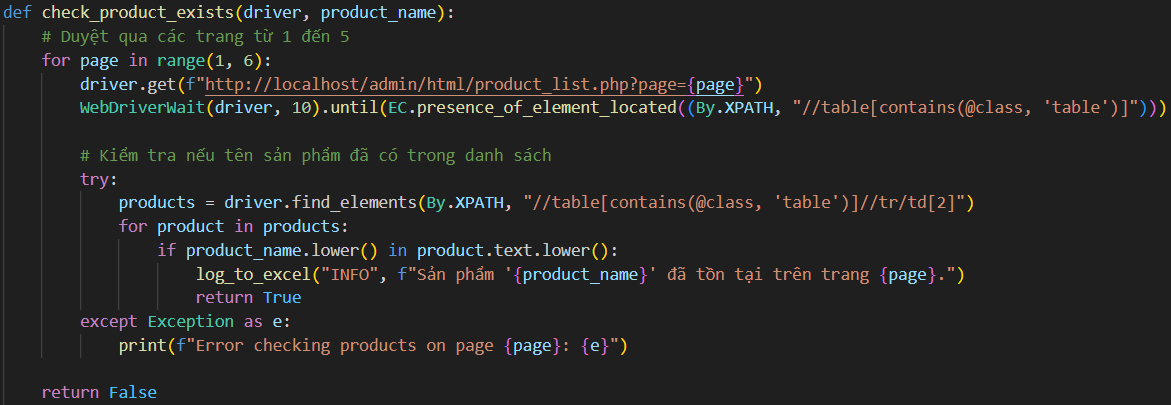
Các bước thực hiện test tự động bằng Selenium cho chức thêm sản phẩm

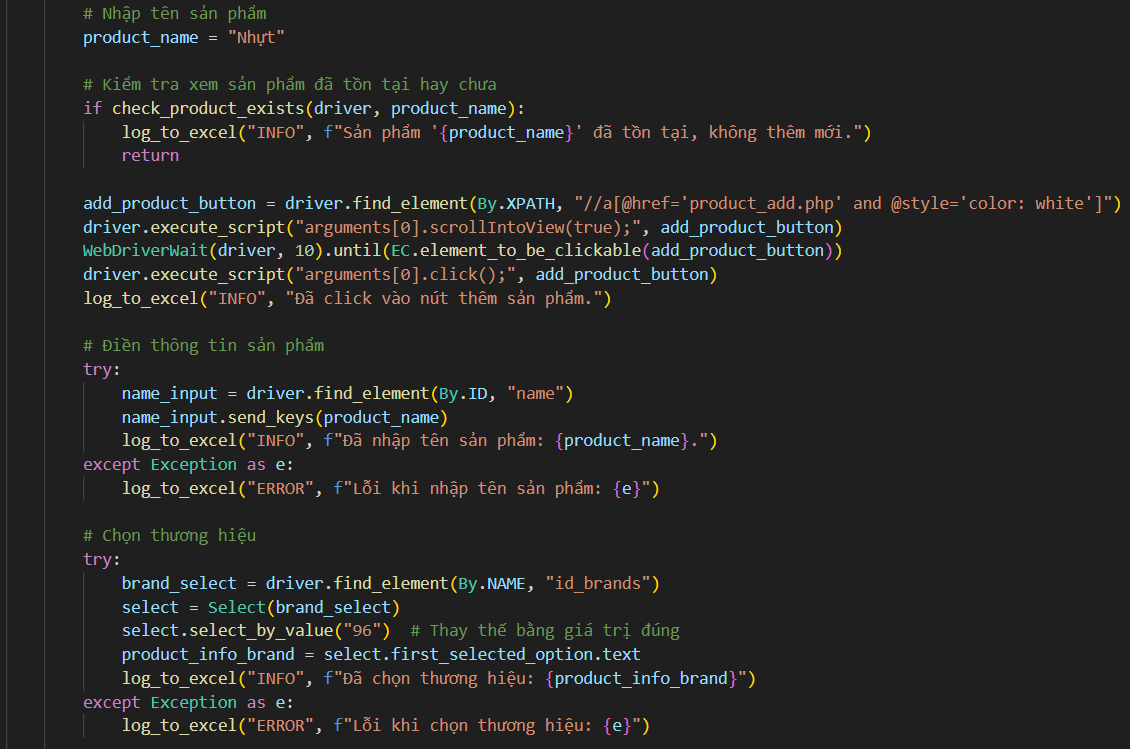
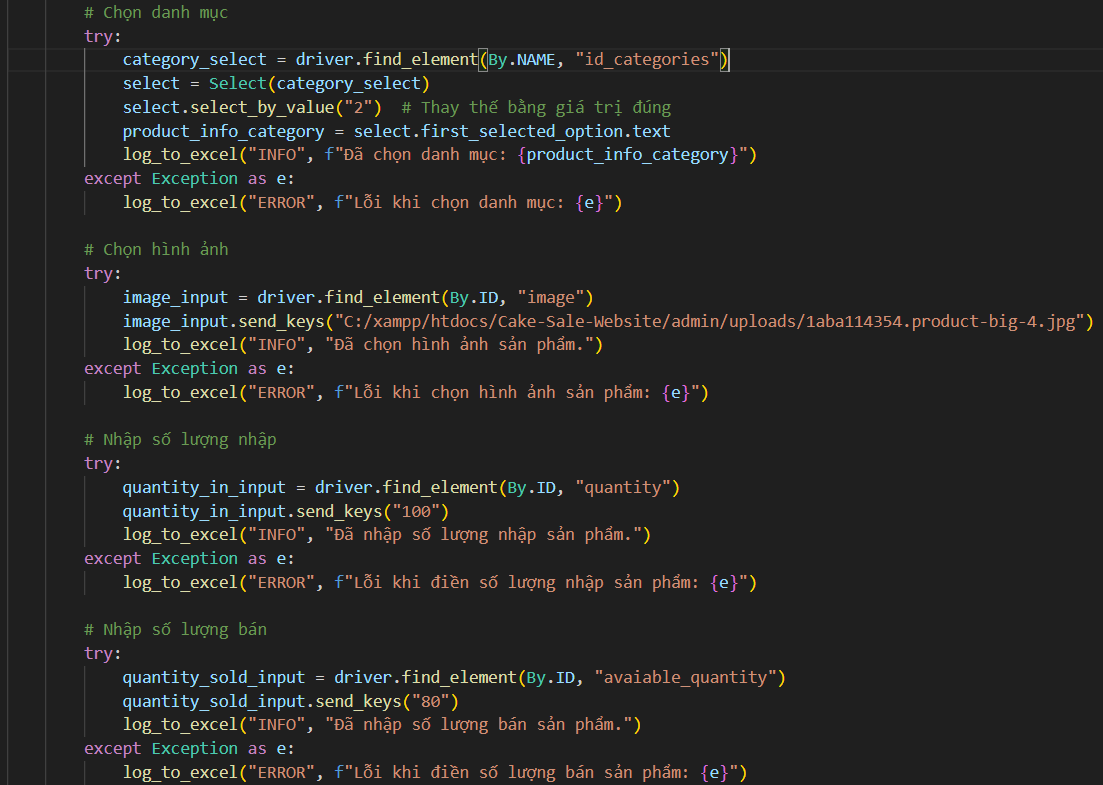
Bước1: Các thư viện đã sử dụng 

Bước 2: Tạo hàm ghi các log vào excel



Bước 3: Tạo hàm kiểm tra sản phẩm đã tồn tại hay chưa (duyệt qua các trang)



Bước 4: Tạo hàm thêm sản phẩm   

Bước 5: Thực thi



# CHƯƠNG 6. KẾT LUẬN

Kiểm thử phần mềm đóng vai trò cực kỳ quan trọng trong việc đảm bảo chất lượng và sự hoạt động ổn định của hệ thống. Khi người dùng đưa ra yêu cầu về một website bán bánh, người thiết kế hệ thống cần phải phân tích kỹ các yêu cầu để xây dựng kiến trúc phần mềm phù hợp. Quá trình kiểm thử sẽ bắt đầu với việc viết các test case cho từng phần của hệ thống, bao gồm kiểm thử đơn vị (unit testing), kiểm thử tích hợp (integration testing) và kiểm thử hệ thống (system testing). Mỗi loại kiểm thử này giúp phát hiện lỗi ở các mức độ khác nhau, từ mức độ chi tiết của từng phần nhỏ của mã nguồn cho đến toàn bộ hệ thống. Kết quả từ quá trình kiểm thử này sẽ cung cấp cái nhìn rõ ràng về hiệu suất, tính bảo mật và tính ổn định của website, đồng thời đảm bảo rằng các chức năng của hệ thống đáp ứng đúng yêu cầu người dùng, từ đó cải thiện chất lượng và độ tin cậy của sản phẩm cuối cùng.