

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA CÔNG NGHỆ THÔNG TIN I



ĐỒ ÁN
TỐT NGHIỆP ĐẠI HỌC

Đề tài:

**Nghiên cứu Selenium WebDriver
và áp dụng trong kiểm thử ứng dụng Web**

Giảng viên hướng dẫn	: TS. ĐỖ THỊ BÍCH NGỌC
Sinh viên thực hiện	: ĐỖ THỊ PHƯỢNG
Lớp	: D11CNPM1
Khóa	: 2011 – 2016
Hệ	: Đại học chính quy

Hà Nội, tháng 12 năm 2015

LỜI CẢM ƠN

Để có được kết quả như ngày hôm nay, trước tiên em xin gửi lời cảm ơn sâu sắc nhất tới TS. Đỗ Thị Bích Ngọc, giảng viên bộ môn Công nghệ phần mềm, khoa Công nghệ thông tin I, Học viện Công nghệ Bru chính Viễn thông đã tận tình hướng dẫn, giúp đỡ em trong suốt quá trình nghiên cứu và thực hiện đồ án tốt nghiệp.

Em cũng xin bày tỏ lòng biết ơn chân thành tới các thầy cô giáo trong Học viện Công nghệ Bru chính Viễn thông, đặc biệt là các thầy cô trong khoa Công nghệ thông tin I đã tận tụy dạy dỗ em trong quá trình học tập, rèn luyện suốt thời gian qua, giúp em có được những kiến thức không chỉ để hoàn thành đồ án mà còn học tập và làm việc sau này.

Đặc biệt, lời cảm ơn thiêng liêng nhất con xin được gửi tới bố mẹ. Con xin cảm ơn bố mẹ đã luôn yêu thương, tin tưởng con, dù có khó khăn thế nào cũng động viên, ủng hộ và tạo điều kiện để luôn là chỗ dựa tinh thần vững chắc cho con trong suốt thời gian ngồi trên ghế nhà trường.

Và cuối cùng tôi xin gửi lời cảm ơn tới những người bạn luôn ở bên, chia sẻ cùng tôi những lúc khó khăn, đóng góp ý kiến và giúp đỡ trong quá trình học tập, nghiên cứu và hoàn thành đồ án tốt nghiệp.

Trong quá trình thực hiện đồ án, mặc dù đã cố gắng để hoàn thiện đề tài tốt nhất có thể, nhưng chắc chắn không thể tránh khỏi những thiếu sót. Vì vậy, em rất mong nhận được sự góp ý của thầy cô và các bạn để đồ án của em được hoàn thiện hơn.

Em xin chân thành cảm ơn!

Hà Nội, ngày 5 tháng 12 năm 2015

Sinh viên

Đỗ Thị Phụng

NHẬN XÉT, ĐÁNH GIÁ, CHO ĐIỂM
(Của giảng viên hướng dẫn)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Điểm: (bằng chữ:)

Đồng ý/ Không đồng ý cho sinh viên bảo vệ trước hội đồng chấm đồ án tốt nghiệp?

.....

Hà Nội, ngày..... thángnăm 2015

CÁN BỘ - GIẢNG VIÊN HƯỚNG DẪN

(ký, họ tên)

NHẬN XÉT, ĐÁNH GIÁ, CHO ĐIỂM
(Của giảng viên phản biện)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Điểm: (bằng chữ:.....)

Đồng ý/ Không đồng ý cho sinh viên bảo vệ trước hội đồng chấm đồ án tốt nghiệp?

.....

Hà Nội, ngày..... thángnăm 2015
CÁN BỘ - GIẢNG VIÊN PHẢN BIỆN
(ký, họ tên)

MỤC LỤC

LỜI CẢM ƠN.....	i
MỤC LỤC	iv
DANH MỤC CÁC BẢNG THUẬT NGỮ VÀ CHỮ VIẾT TẮT	vi
DANH MỤC BẢNG BIỂU	vii
DANH MỤC SƠ ĐỒ, HÌNH ẢNH.....	viii
LỜI MỞ ĐẦU	1
CHƯƠNG 1: TỔNG QUAN VỀ KIỂM THỬ PHẦN MỀM VÀ SELENIUM WEBDRIVER	3
1.1. Kiểm thử phần mềm.....	4
1.1.1. Khái niệm kiểm thử phần mềm	4
1.1.2. Các cấp độ kiểm thử phần mềm	4
1.2. Kiểm thử tự động	5
1.2.1. Khái niệm kiểm thử tự động	6
1.2.2. Quy trình kiểm thử tự động	6
1.2.3. So sánh kiểm thử tự động và kiểm thử thủ công	7
1.2.4. Một số công cụ kiểm thử tự động phổ biến	8
1.3. Công cụ kiểm thử tự động Selenium và Selenium WebDriver	8
1.3.1. Khái quát về Selenium.....	8
1.3.2. Selenium WebDriver	10
1.3.3. Các câu lệnh sử dụng trong Selenium WebDriver	13
1.4. Kết luận chương.....	15
CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ WEBSITE BÁN HÀNG FASHIONISTA	16
2.1. Xác định yêu cầu hệ thống.....	17
2.1.1. Xác định và mô tả các tác nhân	17
2.1.2. Biểu đồ ca sử dụng	17
2.1.3. Xây dựng kịch bản và phác thảo giao diện.....	18
2.2. Phân tích và thiết kế.....	24
2.2.1. Phân tích yêu cầu	24
2.2.2. Thiết kế kiến trúc	25
2.2.3. Thiết kế chi tiết	26
2.2.4. Lược đồ cơ sở dữ liệu	29

2.3. Kết luận chương.....	30
CHƯƠNG 3: ỨNG DỤNG SELENIUM WEBDRIVER TRONG KIỂM THỬ WEBSITE FASHIONISTA	31
3. 1. Thiết kế test case cho các chức năng chính	32
3.1.1. Chức năng đăng ký	32
3.1.2. Chức năng thêm sản phẩm vào giỏ hàng	36
3.1.3. Chức năng sửa giỏ hàng	38
3.1.4. Chức năng đặt hàng	40
3. 2. Hướng giải quyết bài toán và thực thi kiểm thử tự động	43
3.2.1. Các yêu cầu về công nghệ	43
3.2.2. Hướng thực thi kiểm thử tự động	43
3.1.5. Báo cáo và phân tích lỗi	52
3. 3. Kết luận chương.....	53
CHƯƠNG 4: THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ.....	55
4. 1. Giới thiệu website Fashionista.....	56
4. 2. Kiểm thử tự động website với SeleniumWebDriver và TestNG	57
4.2.1. Chức năng đăng ký tài khoản	57
4.2.2. Chức năng thêm sản phẩm vào giỏ hàng	60
4.2.3. Chức năng sửa giỏ hàng	60
4.2.4. Chức năng đặt hàng	61
4. 3. Đánh giá kết quả kiểm thử	62
4.3.1. Kiểm thử chức năng	62
4.3.2. Kiểm thử phi chức năng	63
4. 4. Kết luận chương.....	63
KẾT LUẬN	64
TÀI LIỆU THAM KHẢO	65

DANH MỤC CÁC BẢNG THUẬT NGỮ VÀ CHỮ VIẾT TẮT

Thuật ngữ/ Từ viết tắt	Ý nghĩa
IEEE	Institute of Electrical and Electronic Engineers
Test case	Một tập các dữ liệu đầu vào, hành động hoặc sự kiện và các kết quả mong đợi
Test script	Một nhóm mã lệnh dạng đặc tả kịch bản dùng để tự động hóa một trình tự kiểm thử.
Selenium RC	Selenium Remote Control
Framework	Trong kiểm thử phần mềm, Framework cung cấp một cấu trúc cơ bản giúp cho việc kiểm thử trên ngôn ngữ lập trình bậc cao một cách nhanh chóng hơn, tiết kiệm thời gian và giảm thiểu số lần phải viết lại mã kiểm thử.
Validate	Một thuật ngữ trong kiểm thử phần mềm dùng để chỉ sự kiểm tra tính hợp lệ của dữ liệu trên một yếu tố của ứng dụng

DANH MỤC BẢNG BIỂU

Bảng 2.1. Kịch bản use case đăng nhập	18
Bảng 2.2. Kịch bản use case đăng ký tài khoản	19
Bảng 2.3. Kịch bản use case tìm kiếm sản phẩm	20
Bảng 2.4. Kịch bản use case xem thông tin sản phẩm từ form tìm kiếm	21
Bảng 2.5. Kịch bản use case xem thông tin sản phẩm từ form xem chi tiết giỏ hàng.....	21
Bảng 2.6. Kịch bản use case Thêm sản phẩm vào giỏ hàng.....	22
Bảng 2.7. Kịch bản use case Sửa số lượng sản phẩm.....	22
Bảng 2.8. Kịch bản use case xóa sản phẩm trong giỏ hàng.....	23
Bảng 2.9. Kịch bản use case Đặt hàng	24
Bảng 3.1: Mô tả yêu cầu chức năng đăng ký.....	32
Bảng 3.2: Các trường hợp test chức năng đăng ký.....	36
Bảng 3.4: Mô tả yêu cầu chức năng thêm sản phẩm vào giỏ hàng	36
Bảng 3.5: Các trường hợp test chức năng thêm sản phẩm vào giỏ hàng.....	38
Bảng 3.7: Mô tả yêu cầu chức năng sửa giỏ hàng	38
Bảng 3.8: Các trường hợp test chức năng sửa giỏ hàng	40
Bảng 3.10: Mô tả yêu cầu chức năng đăng ký.....	41
Bảng 3.11: Các trường hợp test chức năng đặt hàng.....	43
Bảng 3.13: Test script được export từ kịch bản kiểm thử Selenium IDE.....	46
Bảng 3.14: Kiểm thử khả năng tương thích.....	50
Bảng 4.1: Kết quả test chức năng đăng ký tài khoản mới	59
Bảng 4.2: Kết quả test chức năng thêm sản phẩm vào giỏ hàng	60
Bảng 4.3: Kết quả test chức năng sửa giỏ hàng.....	61
Bảng 4.4: Kết quả test chức năng đặt hàng	61
Bảng 4.5: Kết quả kiểm thử chức năng	62
Bảng 4.6: Kết quả kiểm thử phi chức năng	63

DANH MỤC SƠ ĐỒ, HÌNH ẢNH

Hình 1.1. Các cấp độ kiểm thử phần mềm	4
Hình 1.2: Selenium RC.....	12
Hình 2.1: Biểu đồ ca sử dụng	17
Hình 2.2: Biểu đồ lớp thiết kế	26
Hình 2.4: Biểu đồ tuần tự chức năng đăng ký tài khoản	27
Hình 2.5: Biểu đồ tuần tự chức năng thêm sản phẩm vào giỏ hàng	28
Hình 2.3: Biểu đồ tuần tự chức năng đăng nhập	29
Hình 2. 6: Lược đồ cơ sở dữ liệu.....	30
Hình 3.1: Quá trình kiểm thử tự động	45
Hình 3.2: Tạo class TestNG	47
Hình 3.3: Các Test case cho chức năng Sửa giỏ hàng.....	48
Hình 3.4: Kết quả chạy test case trên TestNG –All Tests	49
Hình 3.5: Kết quả chạy test case trên TestNG – Summary	49
Hình 3.6: Kết quả chụp lại màn hình khi đăng ký xảy ra lỗi.....	51
Hình 3.7: Kết quả thu được sau khi kiểm thử chức năng Sửa giỏ hàng	52
Hình 3.8: TestNG report (emailable-report.html)	53
Hình 3.9: TestNG report(index.html) hỗ trợ tìm nguyên nhân xảy ra lỗi	53
Hình 4.1: Kịch bản chức năng đăng ký tài khoản	57
Hình 4.2: Dữ liệu đầu vào cho chức năng đăng ký tài khoản.....	58
Hình 4.3: Kết quả đầu ra của chức năng đăng ký tài khoản	58
Hình 4.4: Báo cáo kết quả kiểm thử chức năng đăng ký tài khoản	59

LỜI MỞ ĐẦU

Các ứng dụng Web càng ngày càng phát triển mạnh mẽ nhằm đáp ứng tối đa những nhu cầu của người dùng và dần trở thành nền tảng kết nối thông tin thiết yếu trong nhiều doanh nghiệp, đóng vai trò quyết định của thương mại điện tử, trao đổi thông tin. Để có thể đạt được điều này, các ứng dụng Web cần phải có hiệu năng cao, đáng tin cậy,... Việc đưa ra một ứng dụng Web hoàn hảo cho những người đang và sẽ sử dụng ứng dụng đã trở thành một thách thức chính trong đảm bảo chất lượng phần mềm. Sự phức tạp, tính linh hoạt và sự phụ thuộc của các ứng dụng vào một dịch vụ với nhiều người sử dụng trên các nền tảng khác nhau (hệ điều hành, trình duyệt,...) là một thách thức mà các nhà phát triển website phải đối mặt. Điều này dẫn tới việc kiểm thử ứng dụng Web trở nên thiết yếu đối với các dự án phần mềm và cần phải có những phương pháp đặc biệt khác với phần mềm truyền thống.

Trong khi đó, để tạo ra sản phẩm công nghệ thông tin hay phần mềm có chất lượng thì hoạt động kiểm thử phần mềm đóng vai trò quan trọng. Tuy nhiên, kiểm thử thủ công lại chiếm đến hơn 40% thời gian, kinh phí và nguồn nhân lực phát triển dự án phần mềm[1]. Do vậy, một loạt các công cụ hỗ trợ kiểm thử đơn vị (như TestNG, JUnit, NUnit, TagUnit, JBehave,...), kiểm thử chức năng (như Selenium, Quicktest professional, IBM Rational Functional tester,...), kiểm thử hiệu năng (như Jmeter, OpenSTA, Load runner,...) đã ra đời. Trong đó, Selenium là một công cụ kiểm thử tự động cho các ứng dụng Web, có thể kiểm thử trên nhiều trình duyệt, hỗ trợ nhiều ngôn ngữ lập trình, giao tiếp được với các công cụ kiểm thử khác như JUnit, TestNG (với Java) hay NUnit (với C#) và đặc biệt công cụ này là một bộ mã nguồn mở, do đó các tổ chức không cần tốn kinh phí mua bản quyền.

Xuất phát từ thực tế đó, em đã chọn đề tài: **“Nghiên cứu Selenium WebDriver và áp dụng trong kiểm thử ứng dụng Web”** với mong muốn có được cái nhìn xác thực, rõ ràng hơn về kiểm thử phần mềm và tiếp cận được với công cụ kiểm thử tự động Selenium để làm tiền đề cho định hướng tương lai khi tốt nghiệp đại học sẽ trở thành một kỹ sư kiểm thử phần mềm.

Mục tiêu đồ án

Đồ án được thực hiện nhằm mục đích tìm hiểu về kiểm thử phần mềm, kiểm thử tự động phần mềm, đặc biệt là kiểm thử các ứng dụng Web và các công cụ kiểm thử tự động. Đồ án tập trung nghiên cứu vai trò, phương pháp cũng như ứng dụng của Selenium WebDriver trong việc kiểm thử ứng dụng Web. Từ đó, thực hiện phân tích, xây dựng các test case cho các chức năng của một website; sử dụng ngôn ngữ lập trình Java tạo các test

script từ các test case để kiểm tra khả năng thực hiện đúng và đủ của từng chức năng. Kết quả kiểm thử góp phần đánh giá được chất lượng của Website đó.

Với mục đặt ra như trên, những nội dung và kết quả chính của đồ án được trình bày trong bốn phần như sau:

Chương 1: Tổng quan về kiểm thử phần mềm và Selenium WebDriver

Chương 1 giới thiệu tổng quan về kiểm thử phần mềm, nêu lên vai trò kiểm thử trong quá trình phát triển dự án phần mềm và quy trình kiểm thử phần mềm tự động. Đồng thời tập trung tìm hiểu việc kiểm thử tự động ứng dụng Web, các bước kiểm thử và lựa chọn công cụ kiểm thử phù hợp – Selenium WebDriver. Từ đó đi sâu vào tìm hiểu công cụ Selenium WebDriver về vai trò, ưu nhược điểm cũng như các đặc trưng và các câu lệnh mà công cụ hỗ trợ.

Chương 2: Phân tích thiết kế website bán hàng Fashionista.

Chương 2 tập trung phân tích và thiết kế website, tạo kịch bản, biểu đồ tuần tự, các sơ đồ lớp và lược đồ cơ sở dữ liệu. Từ đó xây dựng lên một website hoàn chỉnh phục vụ cho việc thực thi việc kiểm thử tự động với công cụ Selenium WebDriver.

Chương 3: Ứng dụng Selenium WebDriver trong kiểm thử website Fashionista

Chương 3 đi vào xây dựng kịch bản kiểm thử cho các chức năng chính của website. Đồng thời, đưa ra giải pháp giải quyết bài toán kiểm thử tự động ứng dụng Web, quy trình thực thi kiểm thử tự động và đưa ra báo cáo cho kết quả kiểm thử.

Chương 4: Thực nghiệm và đánh giá kết quả

Áp dụng các kiến thức đã trình bày vào bài toán cụ thể: Sử dụng công cụ kiểm thử tự động Selenium WebDriver để kiểm thử tự động các chức năng đã phân tích.

CHƯƠNG 1: TỔNG QUAN VỀ KIỂM THỬ PHẦN MỀM VÀ SELENIUM WEBDRIVER

- Kiểm thử phần mềm
 - Kiểm thử tự động
 - Công cụ kiểm thử tự động Selenium và Selenium WebDriver
 - Kết luận chương
-

Chương đầu tiên của đồ án sẽ đưa ra cái nhìn tổng quan về kiểm thử phần mềm, nêu lên vai trò kiểm thử trong quá trình phát triển dự án phần mềm và quy trình kiểm thử phần mềm tự động. Đồng thời tập trung tìm hiểu việc kiểm thử tự động ứng dụng Web, các bước kiểm thử và lựa chọn công cụ kiểm thử phù hợp – Selenium WebDriver. Từ đó đi sâu vào tìm hiểu công cụ Selenium WebDriver về vai trò, ưu nhược điểm cũng như các đặc trưng và các câu lệnh mà công cụ hỗ trợ.

1.1. Kiểm thử phần mềm

1.1.1. Khái niệm kiểm thử phần mềm

Kiểm thử phần mềm có nhiều định nghĩa khác nhau đề xuất bởi nhiều tổ chức hay cá nhân khác nhau. Dưới đây là một số định nghĩa nổi bật:

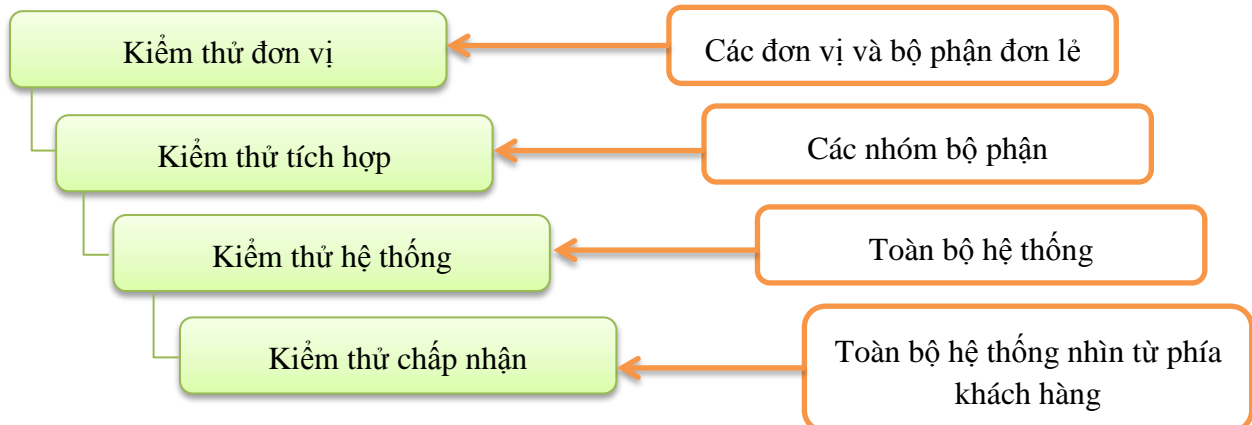
- Định nghĩa của Myer(1979):
“Kiểm thử phần mềm là quá trình thực thi một chương trình với mục đích tìm lỗi.”
- Hai định nghĩa của IEEE(1990):
 - Kiểm thử phần mềm là quá trình vận hành một hệ thống hoặc một thành phần của hệ thống với các điều kiện xác định, nhận xét và ghi lại các kết quả, tạo ra đánh giá về những khía cạnh của hệ thống hay thành phần đó.
 - Kiểm thử phần mềm là quá trình phân tích các yếu tố phần mềm để phát hiện những khác biệt giữa chương trình với các điều kiện yêu cầu và đánh giá các đặc điểm của các yếu tố phần mềm[6].

Mục tiêu của kiểm thử phần mềm:

- Các mục tiêu trực tiếp:
 - Xác định và phát hiện nhiều lỗi nhất có thể trong phần mềm được kiểm thử
 - Sau khi sửa chữa các lỗi đã xác định và kiểm tra lại, làm cho phần mềm đã được kiểm thử đến một mức độ chấp nhận được về chất lượng.
 - Thực hiện các yêu cầu kiểm thử cần thiết một cách hiệu quả và có hiệu quả, trong phạm vi ngân sách và thời gian cho phép.
- Các mục tiêu gián tiếp
 - Để biên dịch một tài liệu về các lỗi phần mềm thường gặp nhằm mục đích ngăn ngừa và sửa chữa lỗi.

1.1.2. Các cấp độ kiểm thử phần mềm

Một sản phẩm phần mềm từ khi bắt đầu phát triển đến khi hoàn thành và đưa đến tay người dùng cuối phải trải qua bốn mức kiểm thử:



Hình 1.1. Các cấp độ kiểm thử phần mềm

➤ **Kiểm thử đơn vị - Unit Test**

Kiểm thử đơn vị là mức thấp nhất trong các mức độ kiểm thử. Đơn vị thường có kích thước nhỏ, chức năng hoạt động đơn giản, không gây nhiều khó khăn trong việc kiểm thử, ghi nhận và phân tích kết quả. Do đó, nếu phát hiện lỗi thì việc tìm kiếm nguyên nhân và sửa lỗi cũng đơn giản và tốn ít chi phí hơn. Tuy nhiên, kiểm thử mức đơn vị lại tốn nhiều thời gian để thực hiện, chưa phát hiện được các lỗi xảy ra khi tích hợp[2].

- Kiểm thử thực hiện trên các hàm hay thành phần riêng lẻ
- Cần hiểu biết về thiết kế chương trình và code
- Thực hiện bởi lập trình viên để đảm bảo thời gian do việc phát hiện và sửa lỗi cần được thực hiện liên tục

➤ **Kiểm thử tích hợp - Integration Test**

Kiểm thử tích hợp kết hợp các thành phần của một ứng dụng và kiểm thử như một ứng dụng đã hoàn thành. Trong khi Unit Test kiểm tra các thành phần và Unit riêng lẻ thì Integration Test kết hợp chúng lại với nhau và kiểm tra sự giao tiếp giữa chúng.

Mục tiêu:

- Phát hiện lỗi giao tiếp xảy ra giữa các Unit cũng như lỗi của bản thân từng Unit
- Tích hợp các Unit đơn lẻ thành các hệ thống nhỏ và cuối cùng là nguyên hệ thống hoàn chỉnh chuẩn bị cho kiểm tra ở mức hệ thống

➤ **Kiểm thử hệ thống - System Test**

Kiểm thử hệ thống là một mức của tiến trình kiểm thử phần mềm khi các module và tích hợp các module đã được kiểm thử thành công. System Test thường được thực hiện bởi một nhóm kiểm thử viên hoàn toàn độc lập với nhóm phát triển dự án với mục tiêu là để đánh giá phần mềm có tuân thủ theo các yêu cầu đã đưa ra không.

System Test kiểm thử cả các hành vi chức năng của phần mềm lẫn các yêu cầu về chất lượng như độ tin cậy, tính tiện lợi khi sử dụng, hiệu năng và bảo mật. Mức kiểm thử này đặc biệt thích hợp cho việc phát hiện lỗi giao tiếp với phần mềm hoặc phần cứng bên ngoài, chẳng hạn các lỗi "tắc nghẽn" (deadlock) hoặc chiếm dụng bộ nhớ.

➤ **Kiểm thử chấp nhận - Acceptance Test**

Thông thường, sau giai đoạn System Test là Acceptance Test, được khách hàng thực hiện hoặc ủy quyền cho một nhóm thứ ba thực hiện. Mục đích của Acceptance Test là để chứng minh phần mềm thỏa mãn tất cả yêu cầu của khách hàng và khách hàng chấp nhận sản phẩm.

1.2. Kiểm thử tự động

Kiểm thử phần mềm tốn nhiều chi phí, nhân lực và thời gian. Trong một số dự án, chi phí kiểm thử phần mềm chiếm 40% tổng giá trị của dự án. Do đó một trong các mục

tiêu của kiểm thử là tự động hóa nhiều, nhờ đó mà giảm thiểu chi phí, giảm lỗi, đặc biệt giúp việc kiểm thử dễ dàng và nhanh chóng hơn.

1.2.1. Khái niệm kiểm thử tự động

Kiểm thử tự động là thực hiện kiểm thử phần mềm một cách tự động các bước trong một kịch bản kiểm thử bằng một chương trình đặc biệt với rất ít hoặc không có sự tương tác của con người, giúp cho người thực hiện việc kiểm thử phần mềm không phải lặp đi lặp lại các bước nhàm chán.

Công cụ kiểm thử tự động có thể lấy dữ liệu từ file bên ngoài (Excel, csv...) nhập vào ứng dụng, so sánh kết quả mong đợi (từ file Excel, csv...) với kết quả thực tế và xuất ra báo cáo kết quả kiểm thử.

Kiểm thử tự động sẽ được sử dụng khi dự án không đủ tài nguyên (thời gian, nhân lực và chi phí), phải thực hiện kiểm thử hồi quy khi sản phẩm được sửa đổi hoặc nâng cấp và cần kiểm thử lại các tính năng đã thực hiện tốt trước đó, kiểm tra khả năng vận hành của sản phẩm trong các môi trường đặc biệt (đo tốc độ xử lý trung bình ứng với mỗi yêu cầu, xác định khả năng chịu tải tối đa, kiểm tra các cơ chế an ninh và an toàn, ...).

1.2.2. Quy trình kiểm thử tự động

Quy trình kiểm thử tự động gồm các bước sau:

- Lập kế hoạch kiểm thử
- Thiết kế test case
- Phát triển test script
- Thực hiện kiểm thử
- Đánh giá kết quả kiểm thử

➤ Lập kế hoạch kiểm thử

Mục đích: Nhằm chỉ định và mô tả các loại kiểm thử sẽ được triển khai và thực hiện. Kết quả của bước lập kế hoạch là bản tài liệu kế hoạch kiểm thử phần mềm, bao gồm:

- Các giai đoạn kiểm thử áp dụng cho dự án phần mềm
- Chiến lược kiểm thử
- Các công cụ kiểm thử
- Nguồn lực kiểm thử
- Môi trường kiểm thử, bao gồm tài nguyên phần cứng và phần mềm
- Mốc bàn giao các tài liệu kiểm thử

➤ Thiết kế Test case

Mục đích: Nhằm xác định các test case và các bước kiểm tra chi tiết cho mỗi phiên bản phần mềm. Giai đoạn thiết kế test case là hết sức quan trọng, nó bảo đảm tất cả các tình huống kiểm thử “quét” hết tất cả yêu cầu cần kiểm tra.

➤ **Phát triển Test Script**

Mục đích: Bước này thường không bắt buộc trong các loại và mức kiểm thử, chỉ yêu cầu trong những trường hợp đặc thù cần thiết kể, tạo ra các test script có khả năng chạy trên máy tính giúp tự động hóa việc thực thi các bước kiểm tra đã định nghĩa ở bước thiết kế test case.

Trong đó, một test script được hiểu là một nhóm mã lệnh dạng đặc tả kịch bản dùng để tự động hóa một trình tự kiểm thử, giúp cho việc kiểm thử nhanh hơn, hoặc cho những trường hợp mà kiểm thử bằng tay sẽ rất khó khăn hoặc không khả thi. Các test script có thể tạo thủ công hoặc tạo tự động dùng công cụ kiểm thử tự động.

➤ **Thực hiện kiểm thử**

Mục đích: Thực hiện các bước kiểm tra đã thiết kế hoặc thực thi các test script nếu tiến hành kiểm tra tự động và ghi nhận kết quả.

➤ **Đánh giá kết quả kiểm thử**

Mục đích: Đánh giá toàn bộ quá trình kiểm thử, bao gồm xem xét và đánh giá kết quả kiểm thử, liệt kê lỗi, chỉ định các yêu cầu thay đổi, và tính toán các số liệu liên quan đến quá trình kiểm thử như số giờ, thời gian kiểm thử, số lượng lỗi, phân loại lỗi...

1.2.3. So sánh kiểm thử tự động và kiểm thử thủ công

❖ **Ưu điểm:**

- **Độ tin cậy cao (Reliability):** Nhờ sự ổn định vượt trội của công cụ kiểm thử tự động so với con người, đặc biệt trong trường hợp có quá nhiều test case cần được thực thi, nên độ tin cậy của kiểm thử tự động thường cao hơn so với kiểm thử thủ công.
- **Khả năng lặp (Repeatability):** công cụ kiểm thử tự động ra đời là để giúp cho các tester không phải lặp đi lặp lại các thao tác như: nhập dữ liệu, click, check kết quả... một cách nhàm chán với độ tin cậy và ổn định cao.
- **Khả năng tái sử dụng (Reusability):** với một bộ kiểm thử tự động, người ta có thể sử dụng cho nhiều phiên bản ứng dụng khác nhau, đây được gọi là tính tái sử dụng.
- **Tốc độ cao (Fast):** do thực thi bởi máy nên tốc độ của kiểm thử tự động nhanh hơn nhiều so với tốc độ của con người. Nếu cần 5 phút để thực thi một test case một cách thủ công thì có thể người ta chỉ cần khoảng 30 giây để thực thi một cách tự động.
- **Chi phí thấp (Cost Reduction):** nếu áp dụng kiểm thử tự động đúng cách, người ta có thể tiết kiệm được nhiều chi phí, thời gian và nhân lực, do kiểm thử tự động nhanh hơn nhiều so với kiểm thử thủ công, đồng thời nhân lực cần để thực thi và bảo trì scripts không nhiều.

❖ **Nhược điểm:**

- **Khó mở rộng, khó bảo trì (Poor scalability and maintainability):** Trong cùng một dự án, để mở rộng phạm vi cho kiểm thử tự động khó hơn nhiều so với kiểm thử thủ

công. Số lượng công việc phải làm để mở rộng phạm vi cho kiểm thử tự động là nhiều hơn và khó hơn kiểm thử thủ công. Cũng vậy, để cập nhật một test case thủ công, người ta chỉ cần mở ra và gõ, rất đơn giản. Nhưng kiểm thử tự động lại không đơn giản như vậy, cập nhật hay chỉnh sửa yêu cầu rất nhiều công việc như debug, thay đổi dữ liệu đầu vào, và cập nhật code mới.

- Khả năng bao phủ thấp (Low coverage): do khó ứng dụng, khó mở rộng và đòi hỏi nhiều kỹ năng lập trình nên độ bao phủ của kiểm thử tự động khá thấp.
- Vấn đề công cụ và nhân lực (Technology vs. people issues): hiện nay cũng có nhiều công cụ hỗ trợ kiểm thử tự động khá tốt nhưng chúng vẫn còn nhiều hạn chế. Ngoài ra nguồn nhân lực có thể sử dụng thành thạo các công cụ này cũng không nhiều.

1.2.4. Một số công cụ kiểm thử tự động phổ biến

- Selenium
- QuickTest Professional (QTP)
- Test Complete
- LoadTest
- Jmeter
- Visual Studio
- ...

1.3. Công cụ kiểm thử tự động Selenium và Selenium WebDriver

1.3.1. Khái quát về Selenium

Selenium là một công cụ hỗ trợ kiểm thử tự động cho các ứng dụng Web. Selenium hỗ trợ kiểm thử trên hầu hết các trình duyệt phổ biến hiện nay như Firefox, Internet Explorer, Safari, ... cũng như các hệ điều hành chủ yếu như Windows, Linux, Mac,... Selenium cũng hỗ trợ một số lớn các ngôn ngữ lập trình Web phổ biến hiện nay như C#, Java, Perl, PHP, Python, Ruby,... Công cụ này có thể kết hợp thêm với một số công cụ khác như Junit và TestNG nhưng với người dùng thông thường chỉ cần chạy tự động mà không cần cài thêm các công cụ bổ trợ.

a. Đặc điểm của Selenium

- Mã nguồn mở: Đây là điểm mạnh nhất của Selenium khi so sánh với các test tool khác. Vì là mã nguồn mở nên chúng ta có thể sử dụng mà không phải lo lắng về phí bản quyền hay thời hạn sử dụng.
- Cộng đồng hỗ trợ: vì là mã nguồn mở nên Selenium có một cộng đồng hỗ trợ khá mạnh mẽ. Bên cạnh đó, Google là nơi phát triển Selenium nên chúng ta hoàn toàn có thể yên tâm về sự hỗ trợ miễn phí khi có vấn đề về Selenium. Tuy nhiên, đây cũng là một điểm yếu của Selenium. Vì công cụ này hoàn toàn miễn phí, cộng đồng lại đông nên một vấn đề có thể nhiều giải pháp, và có thể một số giải pháp là không hữu ích. Mặc khác, chúng ta không thể hối thúc hay ra deadline cho sự hỗ trợ.

- Selenium hỗ trợ nhiều ngôn ngữ lập trình.
- Selenium hỗ trợ chạy trên nhiều hệ điều hành khác nhau với mức độ chỉnh sửa script hầu như là không có. Thực sự thì điều này phụ thuộc phần lớn vào khả năng viết script của người dùng.
- Chạy test case ở background. Khi chúng ta thực thi một test script, chúng ta hoàn toàn có thể làm việc khác trên cùng một máy tính. Điều này hỗ trợ chúng ta không cần tốn quá nhiều tài nguyên máy móc khi chạy test script.
- Không hỗ trợ Win app. Selenium thực sự chỉ hỗ trợ chúng ta tương tác với Browser mà không hỗ trợ chúng ta làm việc với các Win app, kể cả Win dialog như Download/Upload. Vậy nên, để xử lý các trường hợp cần tương tác với hệ thống hay một app thứ ba, chúng ta cần một hay nhiều thư viện khác như AutoIt hay Coded UI.

Là một công cụ hỗ trợ kiểm tra tính năng nên Selenium không có khả năng giả lập nhiều người dùng ảo cùng một lúc. Công việc của nó là chạy kiểm thử tự động dựa trên một kịch bản đã được thiết kế từ trước. Qua đó chúng ta có thể chắc chắn rằng đối tượng kiểm thử có hoạt động đúng như mong đợi hay không.

b. Các thành phần của Selenium

Selenium là một bộ công cụ hỗ trợ kiểm thử tự động các tính năng của ứng dụng trên nền Web, bao gồm 4 thành phần: Selenium IDE, Selenium Grid, Selenium 1.0 (hay Selenium Remote Control – Selenium RC) và Selenium 2.0 (hay Selenium WebDriver). Mỗi loại có một vai trò cụ thể trong việc hỗ trợ sự phát triển của tự động hóa kiểm thử ứng dụng web.

❖ Selenium IDE (Intergrated Development Environment)

Selenium IDE là được phát triển dưới hình thức add-on của Firefox. Chúng ta chỉ có thể Record trên trình duyệt FireFox, nhưng bù lại, chúng ta có thể Playback trên các trình duyệt khác như là IE, Chrome....

Selenium có thể sinh code tự động hoặc nạp các đoạn mã viết tay. Công cụ này cung cấp chức năng “thu và chạy lại” – Record and Playback. Sau đó chạy lại các câu lệnh này để kiểm thử. Chức năng này rất hữu dụng giúp tiết kiệm thời gian viết kịch bản kiểm thử. Selenium IDE còn cho phép lưu kịch bản đã thu dưới nhiều loại ngôn ngữ lập trình khác nhau như Java, PHP, C#, Ruby...[3].

Selenium Core: Đã được tích hợp trong Selenium IDE. Selenium Core là một công cụ chạy các test script viết bằng Selenese. Thế mạnh của công cụ này là có thể chạy test script trên gần như tất cả các trình duyệt, nhưng lại yêu cầu được cài đặt trên máy chủ của website cần kiểm tra. Điều này là không thể khi Tester không có quyền truy cập đến máy chủ đó.

❖ *Selenium RC (Remote Control)*

Selenium Remote Control là một framework kiểm thử cho phép thực hiện nhiều hơn và tuyến tính các hành động trên trình duyệt. Nó cho phép cho phép các nhà phát triển tự động hóa kiểm thử sử dụng một ngôn ngữ lập trình cho tính linh hoạt tối đa và mở rộng trong việc phát triển logic thử nghiệm.

Công cụ này có thể nhận các test script được thu bởi Selenium IDE, cho phép chỉnh sửa, cải tiến linh động bằng nhiều ngôn ngữ lập trình khác nhau. Sau đó khởi động một trong các trình duyệt Web được chỉ định để thực thi kiểm thử trực tiếp trên trình duyệt đó. Selenium RC còn cung cấp khả năng lưu lại kết quả kiểm thử; cung cấp một API (Application Programming Interface) và thư viện cho mỗi ngôn ngữ được hỗ trợ: HTML, Java, C#, Perl, PHP, Python và Ruby. Khả năng sử dụng Selenium RC với một ngôn ngữ lập trình bậc cao để phát triển các trường hợp kiểm thử cũng cho phép kiểm thử tự động được tích hợp với một dự án xây dựng môi trường tự động.

❖ *Selenium WebDriver*

Selenium WebDriver là sự kế thừa từ Selenium Remote Control, làm việc trực tiếp với trình duyệt ở mức hệ điều hành, cho phép gửi lệnh trực tiếp đến trình duyệt và xuất ra kết quả.

❖ *Selenium-Grid*

Selenium – Grid Là một hệ thống hỗ trợ người dùng thực thi test script trên nhiều trình duyệt một cách song song mà không cần phải chỉnh sửa test script.

Thực hiện phương pháp kiểm tra phân bổ, phối hợp nhiều Selenium RC để có thể thực thi trên nhiều trình duyệt Web khác nhau trong cùng một lúc nhằm giảm thiểu thời gian thực hiện.

1.3.2. Selenium WebDriver

❖ **Tiền thân của Selenium WebDriver**

Trước khi Selenium WebDriver ra đời và phát triển thì Selenium RC là công cụ chính trong suốt một thời gian dài. Hiện nay, Selenium RC không được sử dụng nhiều như Selenium WebDriver nữa, tuy nhiên người dùng vẫn có thể tiếp tục phát triển các kịch bản kiểm thử với Selenium RC.

Selenium RC là công cụ phục vụ cho các công việc kiểm thử đòi hỏi nhiều hơn việc thao tác với các website trên giao diện. Nó cho phép viết các kịch bản kiểm thử tự động ứng dụng Web với sự hỗ trợ của các ngôn ngữ lập trình như Java, C#, Python, Perl, PHP để tạo ra các trường hợp kiểm thử phức tạp hơn như đọc và viết các tệp tin, truy vấn cơ sở dữ liệu, gửi mail kết quả kiểm thử.

Các thành phần của Selenium RC gồm:

- **Máy chủ Selenium:** Thực hiện phân tích và chạy các lệnh được gửi đến từ ứng dụng cần kiểm thử và các thao tác như HTTP proxy, phân tích và xác minh các thông điệp HTTP, giữa trình duyệt và ứng dụng cần kiểm tra.
- **Các thư viện máy khách:** cung cấp sự hỗ trợ lập trình cho phép chạy lệnh Selenium từ chương trình. Các thư viện máy khách hỗ trợ cho các ngôn ngữ lập trình khác nhau thì khác nhau. Giao diện lập trình là tập các chức năng chạy lệnh Selenium, trong mỗi giao diện có một chức năng lập trình hỗ trợ Selenium.

❖ **Đặc trưng của Selenium WebDriver**

Selenium WebDriver (hay còn gọi là Selenium 2.0) kế thừa và phát triển từ Selenium IDE, Selenium RC, Selenium Grid. Selenium WebDriver tương tác trực tiếp với trình duyệt mà không cần thông qua bất kỳ trung gian, không giống như Selenium RC phụ thuộc vào một máy chủ.

Tính năng chính trong Selenium 2.0 là việc tích hợp WebDriver API. Ngoài việc giải quyết một số hạn chế trong Selenium RC API, Selenium WebDriver còn được thiết kế để mang đến một giao diện lập trình đơn giản hơn. Nó cho phép sử dụng một trong số các ngôn ngữ lập trình như HTML, Java, .Net, Perl, Ruby... để tạo kịch bản kiểm thử kết hợp với sử dụng các điều kiện, vòng lặp... khiến cho test script trở nên chính xác hơn.

Selenium WebDriver còn được phát triển tốt hơn để hỗ trợ cho các trang Web động do các phần tử trong một trang Web động có thể thay đổi bất cứ lúc nào, ngay cả khi trang đó không được tải lại. Vì vậy, Selenium WebDriver được phát triển để hỗ trợ quá trình kiểm thử mà không cần phải thực hiện lại khi có thay đổi xảy ra[4].

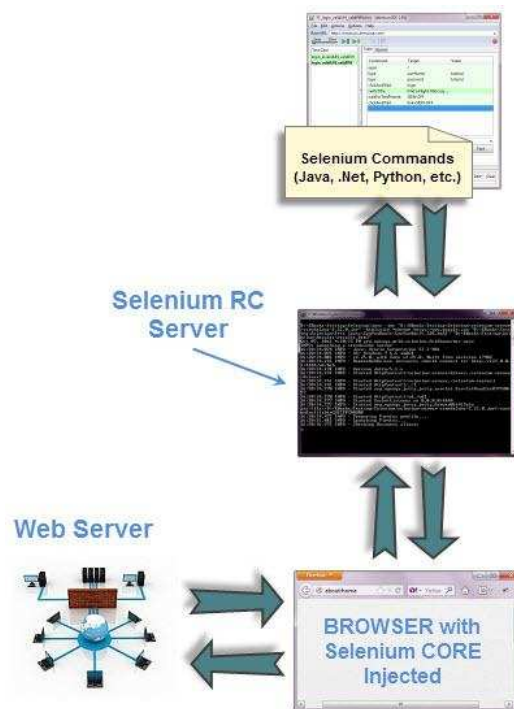
❖ **So sánh Selenium WebDriver và Selenium Remote Control**

• **Sự tương tác thực tế**

WebDriver có tính tương tác gần với thực tế hơn vì nó thực hiện các hành động trực tiếp với trình duyệt giống như các end-user trong thực tế. Còn Selenium RC thì không như vậy, khi sử dụng Selenium RC, cái trực tiếp tương tác với trình duyệt là Selenium Core và vì vậy sẽ có một số trường hợp kết quả trả về sai so với thực tế.

• **Cấu trúc:** Cấu trúc của WebDriver đơn giản hơn Selenium RC.

WebDriver kiểm soát trình duyệt từ cấp độ hệ điều hành. Tất cả những gì chúng ta cần khi sử dụng WebDriver cho công việc kiểm thử là tập lệnh Selenium và một trình duyệt để thực hiện các lệnh đó. Ngược lại, Selenium RC có cấu trúc phức tạp hơn rất nhiều. Thay vì việc trực tiếp thực hiện và nhận phản hồi từ trình duyệt như một end-user thông thường, khi sử dụng Selenium RC chúng ta phải qua rất nhiều bước trung gian như Hình 1.2.



Hình 1.2: Selenium RC

- **Tốc độ**

Từ cấu trúc hoạt động được mô tả như ở trên, có thể thấy tốc độ thực thi và nhận phản hồi từ trình duyệt khi sử dụng WebDriver sẽ nhanh hơn vì nó thực hiện mọi hành động trực tiếp với trình duyệt còn Selenium RC thì phải thông qua trạm trung gian và mất thời gian hơn cho việc chuyển đổi nên thời gian để nhận phản hồi từ trình duyệt sẽ dài hơn.

- **Giao diện lập trình ứng dụng API – Application Programming Interface**

Các API mà Selenium WebDriver cung cấp trong thư viện đơn giản hơn so với Selenium RC. Ví dụ như, với Selenium WebDriver, chúng ta chỉ có *click*; còn với Selenium RC, chúng ta có *click*, *mouseDown*, or *mouseDownAt*. Hơn nữa, khi sử dụng Selenium RC, chúng ta xác định hành động trước rồi mới đèn đối tượng UI; ngược lại, với Selenium WebDriver, chúng ta xác định đối tượng UI trước rồi mới nói đến làm gì trên nó.

Bên cạnh đó, Selenium WebDriver không phải không có điểm yếu:

Điểm yếu của Selenium WebDriver đó là mã nguồn của nó không chạy trên Selenium Grid được. Cho nên, khi muốn thực thi test case với Selenium WebDriver trên nhiều trình duyệt khác nhau, chúng ta phải cấu hình bằng tay. Và các API mà Selenium WebDriver cung cấp không hoạt động tốt với các thành phần Ajax cũng như các thành phần như hộp thoại hay cảnh báo. Thời gian thực thi ca kiểm thử chậm bởi Selenium WebDriver đòi hỏi phải bật và tắt trình duyệt Web mỗi lần thực hiện ca kiểm thử.

1.3.3. Các câu lệnh sử dụng trong Selenium WebDriver

a. Các câu lệnh trình duyệt

Selenium WebDriver có một số các câu lệnh thao tác với trình duyệt như mở, đóng, lấy tiêu đề của trang Web như dưới đây:

- Câu lệnh get

Mục đích: Câu lệnh này sử dụng để mở một trang Web mới trong trình duyệt hiện tại.

Cú pháp: *drive.get(URL);*

Trong đó: URL: Là url để tải trang, nên sử dụng một url đầy đủ

- Câu lệnh lấy tiêu đề getTitle

Mục đích: Câu lệnh này sử dụng để lấy tiêu đề của trang Web hiện tại

Cú pháp: *driver.getTitle();*

- Câu lệnh lấy URL hiện tại getCurrentUrl

Mục đích: Câu lệnh này dùng để lấy URL của trang hiện tại đã được tải trên trình duyệt.

Cú pháp: *driver.getCurrentUrl();*

- Câu lệnh lấy source của trang Web getPageSource

Mục đích: Câu lệnh này dùng để lấy source của trang được tải cuối cùng.

Cú pháp: *driver.getPageSource();*

b. Các câu lệnh WebElement

Để tương tác với một trang Web, cần phải xác định vị trí của phần tử trên trang Web, WebDriver cung cấp 2 phương thức “Find Element” and “Find Elements” để xác định vị trí của phần tử trên trang Web.

- Phương thức “Find Element” và “Find Elements”

Sự khác nhau giữa phương thức “Find Element” và “Find Elements” là trả về đối tượng WebElement, nếu không ném một ngoại lệ và trả về một danh sách WebElement, có thể sẽ trả về danh sách rỗng nếu không có phần tử DOM phù hợp với truy vấn. Phương thức “Find” lấy vị trí hoặc đối tượng truy vấn gọi bằng phương thức “By”

- Tìm phần tử bằng ID: By ID

Mục đích: Tìm vị trí của phần tử bằng ID, nếu tìm được id phù hợp sẽ trả về vị trí của phần tử, nếu không có phần tử phù hợp với id sẽ xuất hiện NoSuchElementException

Cú pháp: *driver.findElement(By.id(""));*

- Tìm phần tử bằng Name (By Name)

Mục đích: Tìm vị trí của phần tử bằng name, nếu tìm được giá trị thuộc tính name phù hợp sẽ trả về vị trí của phần tử, nếu không có phần tử phù hợp với thuộc tính name sẽ xuất hiện NoSuchElementException.

Cú pháp: *driver.findElement(By.name(""))*;

- Tìm phần tử bằng Class Name (className)

Mục đích: Tìm phần tử dựa trên giá trị của thuộc tính “class”.

Cú pháp: *driver.findElement(By.className(""))*;

- Tìm phần tử bằng Link Text

Mục đích: Tìm phần tử của thẻ a bằng tên của link ,

Cú pháp: *driver.findElement(By.tagName(""))*;

c. Các câu lệnh điều hướng trình duyệt

- Câu lệnh forward

Mục đích: Lệnh này dùng để đi đến trang tiếp theo, giống với nút forward trên trình duyệt.

Cú pháp: *driver.navigate().forward()*;

- Câu lệnh back

Mục đích: Lệnh này dùng để quay về trang trước, giống với nút back trên trình duyệt.

Cú pháp: *driver.navigate().back()*;

- Câu lệnh refresh

Mục đích: Lệnh này dùng để làm mới trang hiện tại .

Cú pháp: *driver.navigate().refresh()*;

d. Các lệnh switch

Một số trang Web có nhiều frames hoặc nhiều cửa sổ. Selenium WebDriver gán id cho mỗi cửa sổ ngay khi đối tượng WebDriver được khởi tạo. Id này được gọi là cửa sổ xử lý. Selenium sử dụng id duy nhất này để điều khiển nhiều cửa sổ. Trong đó, mỗi cửa sổ có một id duy nhất, do đó Selenium có thể phân biệt được khi nó được chuyển sang điều khiển một cửa sổ khác.

Dưới đây là một số câu lệnh switch:

- Câu lệnh getWindowHandle

Mục đích: Lệnh này dùng để lấy cửa sổ xử lý (window handle) của cửa sổ hiện tại.

Cú pháp: *driver.getWindowHandle()*;

- Câu lệnh getWindowHandles

Mục đích: Lệnh này dùng để lấy cửa sổ xử lý (window handle) của tất cả các cửa sổ hiện tại.

Cú pháp: *driver.getWindowHandles();*

- Câu lệnh Switch To Window

Mục đích: Lệnh này dùng hỗ trợ di chuyển giữa các cửa sổ khác nhau thông qua tên của chúng bằng cách sử dụng phương thức “switchTo”.

Cú pháp: *driver.switchTo().window("windowName");*

e. Các câu lệnh wait

- Câu lệnh implicitlyWait

Mục đích: Đợi một thời gian nhất định trước khi ném một ngoại lệ khi không thể tìm thấy các phần tử trên trang web.

Cú pháp: *drive.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);*

- Câu lệnh pageLoadTimeout

Mục đích: Thiết lập giá trị thời gian để chờ đợi cho trang Web hoàn thành tải(load) trước khi ném một lỗi.

Cú pháp: *driver.manage().timeouts().pageLoadTimeout(100, SECONDS);*

- Câu lệnh setScriptTimeout

Mục đích: Thiết lập giá trị thời gian chờ đợi một kịch bản(script) không đồng bộ để kết thúc việc thực hiện trước khi ném một lỗi. Nếu thời gian chờ là tiêu cực, sau đó kịch bản sẽ chạy vô hạn .

Cú pháp: *driver.manage().timeouts().setScriptTimeout(100,SECONDS);*

- Câu lệnh sleep

Mục đích: Câu lệnh này hiếm khi được sử dụng vì nó luôn luôn buộc các trình duyệt chờ đợi một thời gian cụ thể.

Cú pháp: *thread.sleep(1000);*

1.4. Kết luận chương

Chương 1 đã trình bày tổng quan về kiểm thử phần mềm, kiểm thử phần mềm tự động, quy trình kiểm thử tự động ứng dụng Web. Đồng thời, cũng đi sâu tìm hiểu công cụ kiểm thử tự động Selenium WebDriver và lí do lựa chọn Selenium làm công cụ thực hiện đồ án.

Trong chương 2, đồ án sẽ trình bày cụ thể phân tích thiết kế website và xây dựng ứng dụng Web để thực hiện kiểm thử.

CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ WEBSITE BÁN HÀNG FASHIONISTA

-
- Xác định yêu cầu bài toán
 - Phân tích yêu cầu
 - Thiết kế
 - Kết luận chương
-

Chương 2 của đồ án tập trung phân tích và thiết kế một website với đầy đủ các chức năng chính như đăng nhập, đăng ký, xem thông tin sản phẩm, thêm sản phẩm vào giỏ hàng, đặt hàng... Từ những phân tích và thiết kế này sẽ xây dựng lên một trang Web được dùng để thực thi việc kiểm thử tự động với công cụ Selenium WebDriver ở Chương 3.

2.1. Xác định yêu cầu hệ thống

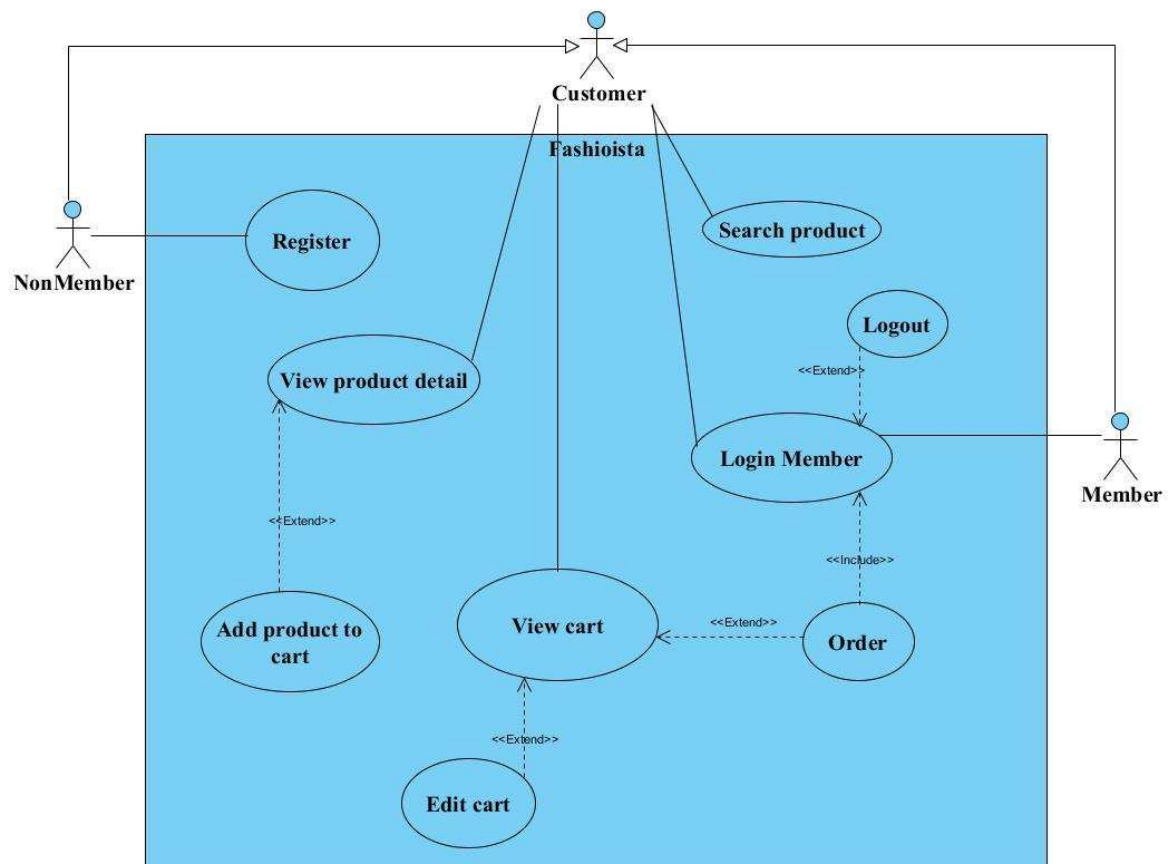
2.1.1. Xác định và mô tả các tác nhân

Hệ thống bán hàng thời trang online Fashionista bao gồm các tác nhân:

- Khách hàng (Customer): tìm kiếm sản phẩm, xem thông tin sản phẩm, xem danh sách sản phẩm sản phẩm theo thể loại, thêm sản phẩm vào giỏ hàng, xem giỏ hàng, thay đổi chi tiết giỏ hàng.
- Thành viên (Member): những khách hàng có thông tin được lưu trong cơ sở dữ liệu có thể đăng nhập hệ thống và đặt hàng.
- Người dùng (NonMember): người dùng có thể đăng ký tài khoản mới khi chưa có tài khoản truy cập hệ thống.

2.1.2. Biểu đồ ca sử dụng

❖ Biểu đồ ca sử dụng của hệ thống



Hình 2.1: Biểu đồ ca sử dụng

❖ Mô tả các ca sử dụng

- U1: Login: tác nhân đăng nhập vào hệ thống.
- U2: Logout: tác nhân đăng xuất khỏi hệ thống.
- U3: Register: người dùng đăng ký tài khoản truy cập hệ thống.

- U4: Search product: tác nhân tìm kiếm theo tên sản phẩm.
- U5: View product detail: tác nhân xem thông tin chi tiết của sản phẩm.
- U6: Add product to carts: tác nhân thêm sản phẩm vào giỏ hàng.
- U7: View carts: tác nhân xem chi tiết giỏ hàng.
- U8: Edit Carts: tác nhân thay đổi chi tiết giỏ hàng.
- U9: Order: thành viên đặt mua hàng trong giỏ hàng.

2.1.3. Xây dựng kịch bản và phác thảo giao diện

❖ Đăng nhập

Tên use case	Đăng nhập
Tác nhân chính	Thành viên
Tiền điều kiện	Thành viên đã có tài khoản để đăng nhập hệ thống
Đảm bảo thành công	Thành viên đăng nhập được vào hệ thống
Đảm bảo tối thiểu	Trở lại màn hình đăng nhập để thành viên có thể đăng nhập lại.
Kích hoạt	Thành viên chọn chức năng đăng nhập.
<p>Chuỗi sự kiện chính</p> <ol style="list-style-type: none"> 1. Thành viên kích hoạt form đăng nhập. 2. Hệ thống hiển thị form đăng nhập gồm các thông tin: tên tài khoản, mật khẩu, một link “Forgot Your Password?” để lấy lại mật khẩu và một nút “Login”. 3. Thành viên nhập tài khoản, mật khẩu và click vào nút “Login”. 4. Hệ thống kiểm tra tính hợp lệ của tài khoản và mật khẩu 5. Hệ thống hiển thị giao diện chính của hệ thống. 	
<p>Ngoại lệ:</p> <ol style="list-style-type: none"> 2.1. Thành viên nhập tài khoản và mật khẩu sai <ol style="list-style-type: none"> 2.1.1. Hệ thống thông báo lỗi và yêu cầu nhập lại. 2.1.2. Thành viên nhập lại thông tin tài khoản và thực hiện các bước sau. 2.2. Tài khoản thành viên đăng nhập không tồn tại <ol style="list-style-type: none"> 2.2.1. Hệ thống thông báo lỗi và yêu cầu người dùng nhập lại. 2.2.2. Thành viên nhập lại thông tin tài khoản và thực hiện các bước sau. 	

Bảng 2.1. Kịch bản use case đăng nhập

❖ Đăng ký tài khoản

Tên use case	Đăng ký tài khoản
Tác nhân chính	Người dùng

Tiền điều kiện	Người dùng kích hoạt giao diện của hệ thống.
Đảm bảo thành công	Người dùng đăng ký tài khoản thành công.
Đảm bảo tối thiểu	Trở lại màn hình đăng ký tài khoản để người dùng có thể tạo lại tài khoản.
Kích hoạt	Người dùng chọn chức năng đăng ký tài khoản.
Chuỗi sự kiện chính <ol style="list-style-type: none"> 1. Người dùng kích hoạt form đăng ký tài khoản. 2. Hệ thống hiển thị form đăng ký tài khoản gồm các thông tin như: tên tài khoản, mật khẩu, xác nhận mật khẩu, tên người dùng, địa chỉ, giới tính, email, số điện thoại và một nút “Register” 3. Khách hàng nhập thông tin tài khoản gồm: tên tài khoản, mật khẩu, xác nhận mật khẩu, tên người dùng, địa chỉ, giới tính, email, số điện thoại và click vào nút “Register”. 4. Hệ thống kiểm tra tính hợp lệ của các thông tin, lưu thông tin người dùng vào cơ sở dữ liệu. 5. Hệ thống hiển thị thông báo đăng ký thành công. 	
Ngoại lệ: <ol style="list-style-type: none"> 3.1. Hệ thống thông báo tài khoản đã tồn tại <ol style="list-style-type: none"> 3.1.1. Hệ thống yêu cầu nhập lại tên tài khoản 3.1.2. Người dùng nhập lại tên tài khoản và tiếp tục các bước sau. 3.2. Hệ thống thông báo thông tin người dùng không hợp lệ <ol style="list-style-type: none"> 3.2.1. Hệ thống yêu cầu nhập lại thông tin tài khoản 3.2.2. Người dùng nhập lại thông tin tài khoản và tiếp tục các bước sau 	

Bảng 2.2. Kịch bản use case đăng ký tài khoản**❖ Tìm kiếm sản phẩm**

Tên use case	Tìm kiếm sản phẩm
Tác nhân chính	Khách hàng
Tiền điều kiện	Khách hàng kích hoạt giao diện của hệ thống.
Đảm bảo thành công	Khách hàng tìm kiếm sản phẩm thành công.
Đảm bảo tối thiểu	Hiển thị màn hình chính để Khách hàng có thể tìm kiếm lại.
Kích hoạt	Khách hàng chọn chức năng tìm kiếm.

Chuỗi sự kiện chính

1. Khách hàng kích hoạt giao diện chính của hệ thống.
2. Hệ thống hiển thị giao diện chính với danh sản phẩm các sản phẩm của hệ thống và một ô nhập tên sản phẩm.
3. Khách hàng nhập vào ô tìm kiếm tên sản phẩm và chọn enter.
4. Hệ thống tìm kiếm sản phẩm trong cơ sở dữ liệu và trả về danh sản phẩm sản phẩm mà Khách hàng yêu cầu tìm kiếm.

Ngoại lệ:

- 3.1. Hệ thống thông báo không có sản phẩm với tên khách hàng yêu cầu tìm kiếm
 - 3.1.1. Hệ thống yêu cầu Khách hàng nhập lại tên sản phẩm
 - 3.1.2. Khách hàng nhập lại tên sản phẩm và tiếp tục các bước sau.

Bảng 2.3. Kịch bản use case tìm kiếm sản phẩm**❖ Xem thông tin sản phẩm****○ Xem thông tin sản phẩm từ form tìm kiếm**

Tên use case	Xem thông tin sản phẩm
Tác nhân chính	Khách hàng
Tiền điều kiện	Khách hàng tìm kiếm sản phẩm thành công công.
Đảm bảo thành công	Khách hàng xem thông tin sản phẩm thành công.
Đảm bảo tối thiểu	Trở lại màn hình chính.
Kích hoạt	Khách hàng chọn sản phẩm cần xem thông tin.

Chuỗi sự kiện chính

1. Khách hàng kích hoạt giao diện chính của hệ thống.
2. Hệ thống hiển thị giao diện chính với danh sách các sản phẩm của hệ thống và một ô nhập tên sản phẩm.
3. Khách hàng nhập vào ô tìm kiếm tên sản phẩm và chọn enter.
4. Hệ thống tìm kiếm sản phẩm trong cơ sở dữ liệu và trả về danh sách sản phẩm mà Khách hàng yêu cầu tìm kiếm gồm tên sản phẩm, hình ảnh và giá sản phẩm.
5. Khách hàng click chọn một sản phẩm trong danh sản phẩm sản phẩm vừa tìm kiếm.
6. Hệ thống hiển thị thông tin chi tiết sản phẩm đã chọn bao gồm: tên sản phẩm, thể loại, giá sản phẩm, trạng thái sản phẩm, một ô nhập số lượng và một nút “Add to carts”.

Ngoại lệ:

- 3.1. Hệ thống thông báo không có sản phẩm với tên khách hàng yêu cầu tìm kiếm

- 3.1.1. Hệ thống yêu cầu Khách hàng nhập lại tên sản phẩm
- 3.1.2. Khách hàng nhập lại tên sản phẩm và tiếp tục các bước sau.

Bảng 2.4. Kịch bản use case xem thông tin sản phẩm từ form tìm kiếm

○ **Xem thông tin sản phẩm từ form Xem chi tiết giỏ hàng**

Tên use case	Xem thông tin sản phẩm
Tác nhân chính	Khách hàng
Tiền điều kiện	Khách hàng thêm sản phẩm vào giỏ hàng thành công.
Đảm bảo thành công	Khách hàng xem thông tin sản phẩm thành công
Đảm bảo tối thiểu	Trở lại giao diện xem chi tiết giỏ hàng.
Kích hoạt	Khách hàng chọn sản phẩm cần xem thông tin.
Chuỗi sự kiện chính <ol style="list-style-type: none"> 1. Khách hàng kích hoạt form Xem chi tiết giỏ hàng 2. Hệ thống hiển thị danh sản phẩm sản phẩm gồm: hình ảnh, tên sản phẩm, giá sản phẩm, số lượng sản phẩm, tổng tiền, một link Delete sản phẩm, một nút “Continue shopping”, một nút “Update shopping carts” và một nút “Proceed to checkout”. 3. Khách hàng click vào ảnh của một sản phẩm trong danh sản phẩm của giỏ hàng. 4. Hệ thống hiển thị thông tin chi tiết sản phẩm đã chọn bao gồm: tên sản phẩm, thể loại, giá sản phẩm, trạng thái sản phẩm, một ô nhập số lượng và một nút “Add to carts”. 	

Bảng 2.5. Kịch bản use case xem thông tin sản phẩm từ form xem chi tiết giỏ hàng

❖ **Thêm sản phẩm vào giỏ hàng**

Tên use case	Thêm sản phẩm vào giỏ hàng
Tác nhân chính	Khách hàng
Tiền điều kiện	Khách hàng xem thông tin sản phẩm thành công.
Đảm bảo thành công	Hệ thống thêm thành công sản phẩm vào giỏ hàng của khách hàng
Đảm bảo tối thiểu	Trở lại màn hình xem thông tin sản phẩm
Kích hoạt	Khách hàng chọn “Add to Carts” từ Form Xem thông tin sản phẩm.
Chuỗi sự kiện chính: <ol style="list-style-type: none"> 1. Khách hàng kích hoạt form Xem thông tin sản phẩm 	

2. Hệ thống hiển thị form Xem thông tin sản phẩm bao gồm các thông tin: tên sản phẩm, thể loại, giá sản phẩm, trạng thái sản phẩm, một ô nhập số lượng và một nút “Add to carts”.
3. Khách hàng nhập số lượng sản phẩm và chọn “Add to carts”.
4. Hệ thống thêm sản phẩm với số lượng khách hàng đã chọn vào giỏ hàng
5. Hệ thống hiển thị giao diện xem chi tiết giỏ hàng.

Ngoại lệ:

3.1. Số lượng sản phẩm khách hàng nhập không hợp lệ.

3.1.1 Hệ thống yêu cầu khách hàng nhập lại số lượng sản phẩm.

3.1.2 Khách hàng nhập lại thông tin sản phẩm và tiếp tục thực hiện các bước sau.

Bảng 2.6. Kịch bản use case Thêm sản phẩm vào giỏ hàng

❖ **Sửa số lượng sản phẩm**

Tên use case	Sửa số lượng sách trong giỏ hàng
Tác nhân chính	Khách hàng
Tiền điều kiện	Khách hàng xem chi tiết giỏ hàng
Đảm bảo thành công	Số lượng sản phẩm trong giỏ hàng được cập nhật thành công.
Đảm bảo tối thiểu	Trở lại màn hình xem chi tiết giỏ hàng
Kích hoạt	Khách hàng chọn nút “Update shopping carts” trên Form Xem chi tiết giỏ hàng

Chuỗi sự kiện chính:

1. Khách hàng kích hoạt form Xem chi tiết giỏ hàng
2. Hệ thống hiển thị danh sản phẩm sản phẩm gồm: hình ảnh, tên sản phẩm, giá sản phẩm, số lượng sản phẩm, tổng tiền, một link Delete sản phẩm, một nút “Continue shopping”, một nút “Update shopping carts” và một nút “Proceed to checkout”.
3. Khách hàng sửa số lượng sản phẩm tương ứng với tên sản phẩm muốn sửa số lượng, và click chọn “Update shopping carts”.
4. Hệ thống kiểm tra số lượng sản phẩm, cập nhật lại số lượng sản phẩm trong giỏ hàng và hiển thị lại giao diện Xem chi tiết giỏ hàng.

Ngoại lệ:

3.1. Số lượng sản phẩm khách hàng nhập không hợp lệ.

3.1.1 Hệ thống yêu cầu khách hàng nhập lại số lượng sản phẩm.

3.1.2 Khách hàng nhập lại thông tin sản phẩm và tiếp tục thực hiện các bước sau.

Bảng 2.7. Kịch bản use case Sửa số lượng sản phẩm

❖ **Xóa sản phẩm trong giỏ**

Tên use case	Xóa sách trong giỏ hàng
Tác nhân chính	Khách hàng
Tiền điều kiện	Khách hàng xem chi tiết giỏ hàng
Đảm bảo thành công	Sản phẩm được xóa khỏi giỏ hàng
Đảm bảo tối thiểu	Trở lại màn hình xem chi tiết giỏ hàng
Kích hoạt	Khách hàng chọn “Delete” trên Form Xem chi tiết giỏ hàng
Chuỗi sự kiện chính: <ol style="list-style-type: none"> 1. Khách hàng kích hoạt form Xem chi tiết giỏ hàng 2. Hệ thống hiển thị danh sản phẩm sản phẩm gồm: hình ảnh, tên sản phẩm, giá sản phẩm, số lượng sản phẩm, tổng tiền, một link Delete sản phẩm, một nút “Continue shopping”, một nút “Update shopping carts” và một nút “Proceed to checkout”. 3. Khách hàng click chọn “Delete” tương ứng với sách cần xóa. 4. Hệ thống xóa sản phẩm trong giỏ hàng 5. Hệ thống hiển thị lại giao diện xem chi tiết giỏ hàng. 	

Bảng 2.8. Kịch bản use case xóa sản phẩm trong giỏ hàng❖ **Đặt hàng**

Tên use case	Đặt hàng
Tác nhân chính	Thành viên
Tiền điều kiện	Khách hàng đã đăng nhập thành và xem giỏ hàng thành công.
Đảm bảo thành công	Thành viên đặt hàng thành công.
Đảm bảo tối thiểu	Trở lại màn hình xem chi tiết giỏ hàng.
Kích hoạt	Khách hàng chọn “Proceed to checkout” từ Form Xem chi tiết giỏ hàng .
Chuỗi sự kiện chính <ol style="list-style-type: none"> 1. Khách hàng kích hoạt form Xem chi tiết giỏ hàng. 2. Hệ thống hiển thị danh sản phẩm sản phẩm gồm: hình ảnh, tên sản phẩm, giá sản phẩm, số lượng sản phẩm, tổng tiền theo từng sản phẩm, một link Delete sản phẩm, một nút “Continue shopping”, một nút “Update shopping carts”, tổng số sản phẩm, tổng tiền và một nút “Proceed to checkout”. 3. Thành viên click chọn “Proceed to checkout”. 	

4. Hệ thống hiển thị thông tin hóa đơn gồm có: thông tin của danh sách các sản phẩm thành viên muốn đặt hàng; form nhập thông tin giao hàng bao gồm: tên người nhận, địa chỉ người nhận, địa chỉ email, số điện thoại và tùy chọn “Order”, “Cancel”.
5. Khách hàng nhập tên người nhận, địa chỉ người nhận, địa chỉ email, số điện thoại và click chọn “Order”.
6. Hệ thống kiểm tra các thông tin: sản phẩm đặt hàng, thông tin người nhận hàng và cập nhật thông tin đặt hàng vào cơ sở dữ liệu.
7. Hệ thống hiển thị form hóa đơn gồm các thông tin: danh sách sản phẩm người dùng đặt mua, tổng tiền, thông tin giao hàng gồm: tên người nhận, địa chỉ người nhận, giới tính, email, số điện thoại và một nút “Continue shopping”.

Ngoại lệ:

- 5.1. Hệ thống thông báo số lượng sản phẩm thành viên đặt mua không hợp lệ.
 - 5.1.1 Hệ thống yêu cầu nhập lại số lượng sản phẩm.
 - 5.1.2 Thành viên nhập lại số lượng sản phẩm và tiếp tục các bước sau.
- 5.1. Hệ thống thông báo thông tin người nhận không hợp lệ.
 - 5.1.1 Hệ thống yêu cầu nhập lại thông tin người nhận.
 - 5.1.2 Thành viên nhập lại số lượng sản phẩm và tiếp tục các bước sau.

Bảng 2.9. Kịch bản use case Đặt hàng

2.2. Phân tích và thiết kế

2.2.1. Phân tích yêu cầu

a. Mô tả hệ thống

Khách hàng sau khi truy cập vào hệ thống thì có thể thực hiện các chức năng như tìm kiếm sản phẩm, xem thông tin sản phẩm, xem danh sách sản phẩm theo thể loại, thêm sản phẩm vào giỏ hàng, xem chi tiết giỏ hàng, thay đổi giỏ hàng mà không cần đăng nhập vào hệ thống. Sau khi xem giỏ hàng, khách hàng có thể thực hiện đặt hàng. Nếu thành viên đã đăng nhập thì thành viên sẽ nhập thông tin giao hàng để đặt hàng. Nếu khách hàng chưa đăng nhập thì hệ thống sẽ yêu cầu khách hàng đăng nhập vào hệ thống. Đối với những người dùng chưa có tài khoản để đăng nhập vào hệ thống thì người dùng có thể đăng ký tài khoản mới.

b. Trích các lớp – Trích danh từ

Khách hàng, Giỏ hàng, Sản phẩm, Sản phẩm, Hóa đơn, Thẻ loại, Hệ thống, Thành viên, người dùng.

❖ Trích lớp điều khiển

Mỗi module chính có một lớp điều khiển:

- BillDAO
- CartsDAO
- CustomerDAO
- ProductsDAO

❖ Trích lớp biên

- Giao diện chính: HomePages.jsp
- Giao diện đăng nhập: Login.jsp
- Giao diện đăng ký tài khoản: Register.jsp
- Giao diện xem chi tiết sản phẩm: Product_view.jsp
- Giao diện xem danh sản phẩm sản phẩm theo thể loại: CategoryGrid.jsp
- Giao diện đặt hàng: Checkout.jsp
- Giao diện yêu cầu lấy lại mật khẩu: PasswordRetrieval.jsp
- Giao diện xem giỏ hàng: ShoppingCart.jsp
- Giao diện xem kết quả tìm kiếm theo tên sản phẩm: ViewSearch.jsp

2.2.2. Thiết kế kiến trúc

a. Lựa chọn công nghệ sử dụng cho hệ thống

- Hệ thống: Sử dụng mô hình MVC2, công nghệ JSP và Servlet
- Giao diện Web: sử dụng HTML và Javascript
- Database: Microsoft SQL Server
- Server: Apache Tomcat

b. Tổng quan về cấu trúc phân tầng của hệ thống

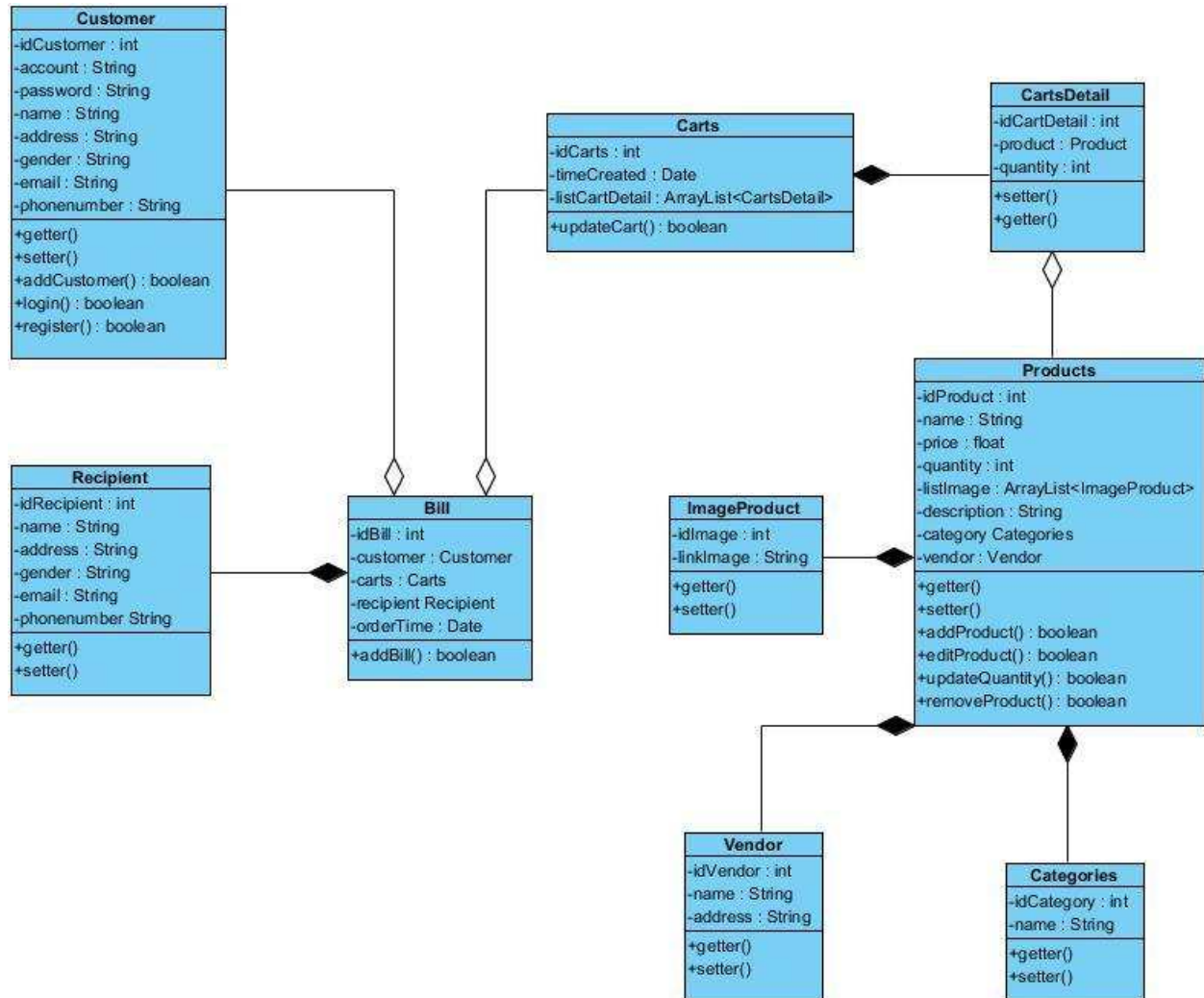
- Tầng trình diễn: Hiện thị giao diện tương tác với người sử dụng
- Tầng nghiệp vụ

Mỗi lớp trong tầng nghiệp vụ đều chứa một thể hiện của lớp DAO. Mục đích của việc xây dựng tầng nghiệp vụ là để tăng tính lỏng lẻo trong việc xây dựng hệ thống, đồng thời tránh việc truy cập trực tiếp với cơ sở dữ liệu.

- Tầng truy cập dữ liệu
 - Truy cập cơ sở dữ liệu với JDBC
 - Tầng này sẽ quản lý việc: đọc dữ liệu, ghi dữ liệu, cập nhật và xóa dữ liệu trong cơ sở dữ liệu.
 - Các lớp DAO sẽ thao tác trực tiếp với cơ sở dữ liệu và các lớp DAO sẽ được gọi đến từ các lớp kế thừa ở tầng nghiệp vụ.
- Các lớp dùng chung
 - Value Objects: lớp thực thể
 - Gói utility: chứa các lớp tĩnh, được sử dụng chung cho toàn bộ hệ thống

2.2.3. Thiết kế chi tiết

a. Biểu đồ lớp thiết kế



Hình 2.2: Biểu đồ lớp thiết kế

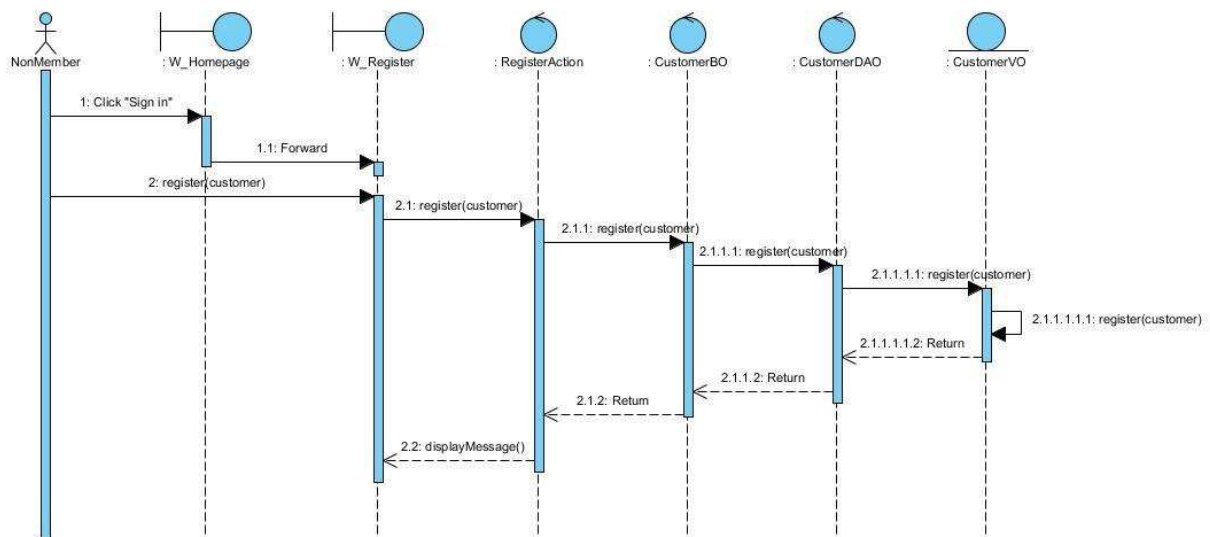
Trong đó:

- Customer: lớp chứa các thông tin cá nhân và thông tin tài khoản của thành viên
- Vendor: lớp chứa thông tin nhà cung cấp sản phẩm
- Categories: lớp chứa thông tin thể loại của sản phẩm
- ImageProduct: địa chỉ cục bộ của các hình ảnh của sản phẩm
- Products: lớp chứa các thông tin của sản phẩm. Quan hệ giữa Vendor, Categories, ImageProduct với Products là quan hệ hợp thành, khi đối tượng sản phẩm được tạo ra thì đối tượng Vendor, Categories, ImageProduct cũng đồng thời được tạo ra.

- CartsDetail: lớp chi tiết giỏ hàng chứa sản phẩm và số lượng sản phẩm trong giỏ hàng. Quan hệ giữa CartsDetail và Products là quan hệ kết hợp. Trong đó, products là thành phần của CartsDetail
- Carts: lớp liên kết chứa danh sách thông tin chi tiết sản phẩm và thời gian thêm vào giỏ hàng. Quan hệ giữa CartsDetail và Carts là quan hệ hợp thành.
- Recipient: lớp người nhận chứa thông tin người nhận hàng
- Bill: lớp hóa đơn chứa các thông tin đặt hàng. Quan hệ giữa Customer, Carts với Bill là quan hệ kết hợp. Trong đó, customer và carts là thành phần của Bill. Quan hệ giữa Recipient với Bill là quan hệ hợp thành, khi đối tượng Bill được tạo ra thì đối tượng người nhận cũng đồng thời được tạo ra.

b. Biểu đồ tuần tự

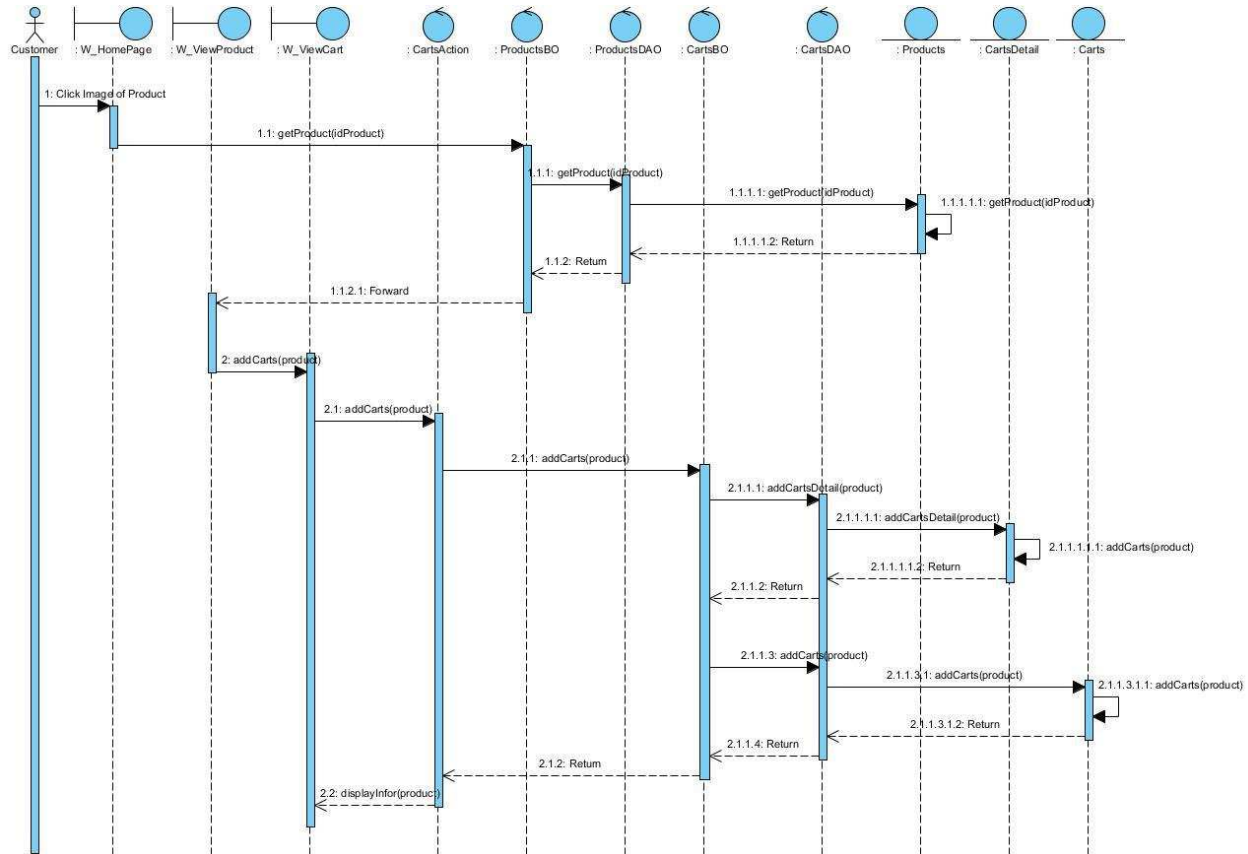
❖ Chức năng đăng ký tài khoản



Hình 2.4: Biểu đồ tuần tự chức năng đăng ký tài khoản

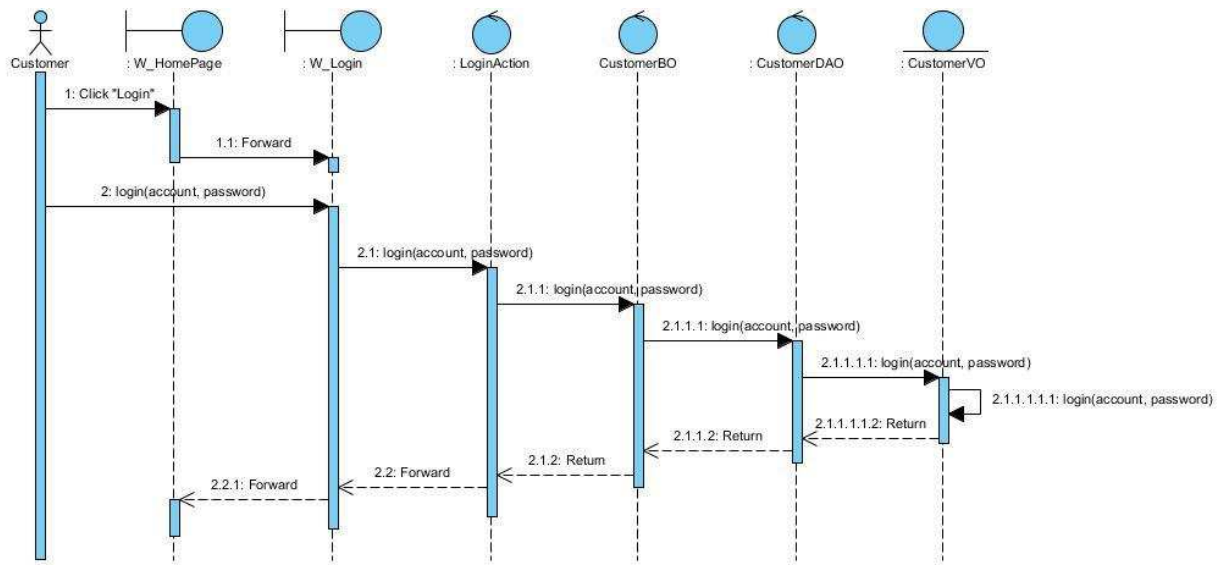
Khi người dùng chọn chức năng đăng ký trên giao diện chính, hệ thống sẽ gọi hiển thị giao diện đăng ký. Sau khi người dùng nhập các thông tin và chọn đăng ký, lớp RegisterAction sẽ nhận yêu cầu và gửi tới lớp CustomerDAO kiểm tra và xử lý thông qua lời gọi hàm trong lớp nghiệp vụ CustomerBO. Nếu thông tin người dùng hợp lệ, hệ thống sẽ thêm thông tin vào cơ sở dữ liệu của hệ thống và lớp RegisterAction gửi lại thông báo thành công. Ngược lại, lớp RegisterAction sẽ gửi lại một thông báo lỗi trên giao diện đăng ký.

❖ Chức năng thêm sản phẩm vào giỏ hàng

**Hình 2.5: Biểu đồ tuần tự chức năng thêm sản phẩm vào giỏ hàng**

Để có thể thêm sản phẩm vào giỏ hàng, trước tiên khách hàng cần phải xem thông tin chi tiết của sản phẩm đó. Tại giao diện chính, khi khách hàng chọn xem thông tin một sản phẩm, lớp nghiệp vụ ProductsBO sẽ gọi đến hàm tìm kiếm sản phẩm trong lớp ProductsDAO. Sau khi tìm kiếm, lớp ProductsDAO sẽ gửi trả về thông tin chi tiết của sản phẩm để hiển thị. Tại giao diện xem thông tin sản phẩm, khách hàng nhập số lượng và chọn thêm vào giỏ hàng. Lớp CartsAction sẽ nhận yêu cầu và gọi đến lớp nghiệp vụ CartsBO yêu cầu lớp CartsDAO kiểm tra và xử lý. Nếu số lượng sản phẩm là hợp lệ, CartsDAO sẽ lưu lại thông tin chi tiết của sản phẩm vào giỏ hàng. Ngược lại, nếu số lượng sản phẩm không hợp lệ, hệ thống sẽ gửi lại một thông báo lỗi tới giao diện xem thông tin sản phẩm.

❖ Chức năng đăng nhập

**Hình 2.3: Biểu đồ tuần tự chức năng đăng nhập**

Khi người dùng lựa chọn chức năng đăng nhập trên giao diện chính, hệ thống sẽ gọi hiển thị giao diện của chức năng đăng nhập. Sau khi người dùng nhập tài khoản, mật khẩu và xác nhận đăng nhập, lớp LoginAction sẽ nhận yêu cầu của người dùng và gọi đến hàm kiểm tra trong lớp CustomerDAO thông qua lớp nghiệp vụ CustomerBO. Sau khi kiểm tra, lớp LoginAction sẽ gửi lại thông báo tới giao diện đăng nhập nếu thông tin người dùng nhập không chính xác. Ngược lại, hệ thống sẽ chuyển sang trang giao diện chính để người dùng có thể mua hàng.

2.2.4. Lược đồ cơ sở dữ liệu**a. Mỗi lớp thực thể đề xuất thành một bảng tương ứng trong CSDL**

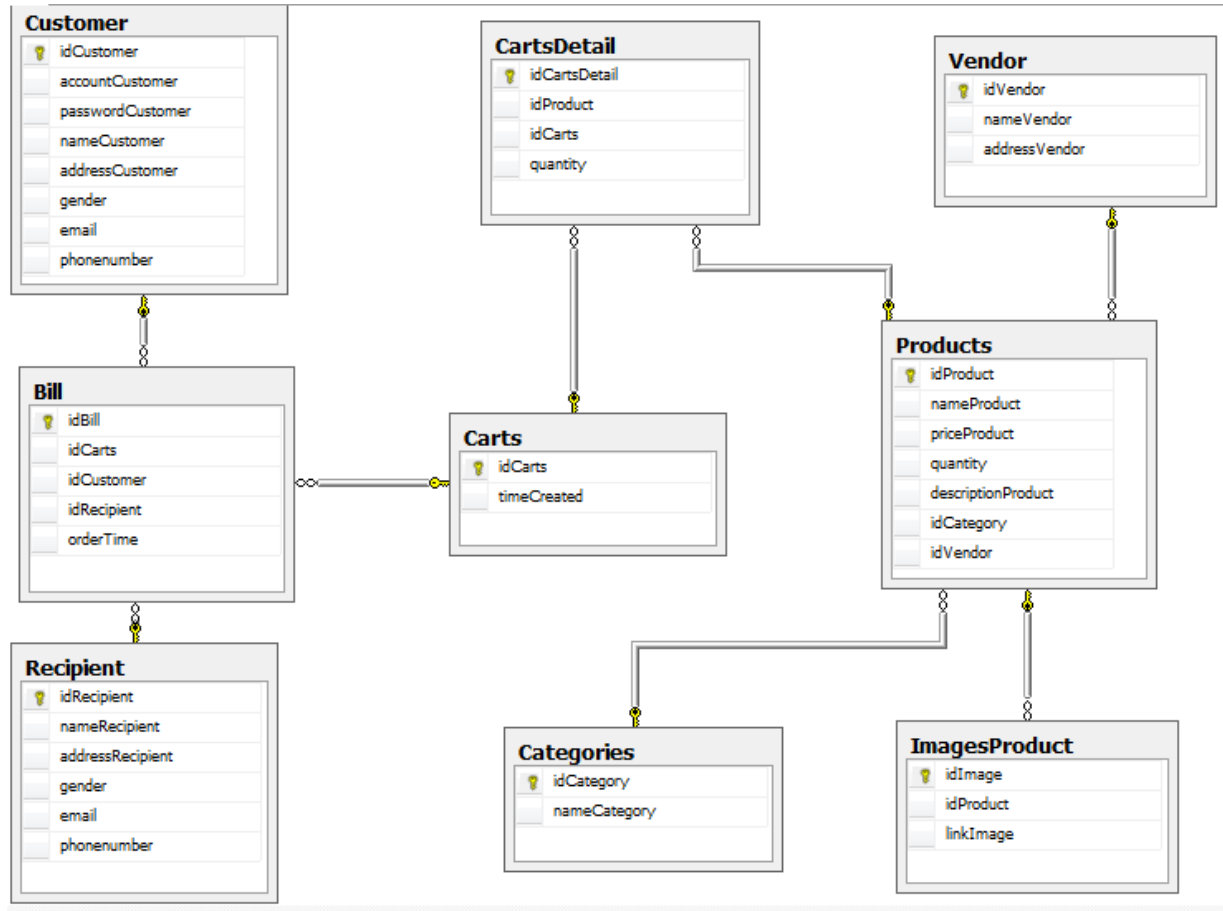
- bảng Customer
- bảng Bill
- bảng CartsDetail
- bảng Carts
- bảng Categories
- bảng Vendor
- bảng Products
- bảng ImagesProduct

b. Quan hệ giữa các bảng

- Products chứa Vendor, Categories và ImagesProduct; cho khóa chính của bảng Vendor, Categories và ImagesProduct thành các thuộc tính khóa ngoại của Products.

- CartsDetail chứa Products; cho khóa chính của bảng Products thành một thuộc tính khóa ngoại của bảng CartsDetail
- Carts chứa danh sách các CartsDetail; cho khóa chính của bảng Carts thành một thuộc tính khóa ngoại của bảng CartsDetail
- Bill chứa Carts, Customer và Recipient; cho khóa chính của bảng Carts, Customer, Recipient thành các thuộc tính khóa ngoại của bảng CartsDetail

c. Lược đồ cơ sở dữ liệu



Hình 2. 6: Lược đồ cơ sở dữ liệu

2.3. Kết luận chương

Chương 2 của đồ án đã phân tích và thiết kế website, vẽ biểu đồ tuần tự, các sơ đồ lớp và lược đồ cơ sở dữ liệu, tạo kịch bản cho các chức năng chính: đăng nhập, đăng ký, xem thông tin sản phẩm, thêm sản phẩm vào giỏ hàng, sửa giỏ hàng, đặt hàng. Từ những phân tích và thiết kế này xây dựng lên website bán hàng online.

Trong chương 3 sẽ thực hiện thiết kế các test case cho các chức năng chính của hệ thống và đưa ra phương hướng giải quyết cho bài toán kiểm thử tự động ứng dụng Web.

CHƯƠNG 3: ỨNG DỤNG SELENIUM WEBDRIVER TRONG KIỂM THỬ WEBSITE FASHIONISTA

- Thiết kế test case cho các chức năng chính
 - Hướng giải quyết bài toán và thực thi kiểm thử tự động
 - Kết luận chương
-

Chương 3 tập trung xây dựng kịch bản kiểm thử cho các chức năng chính của website. Đồng thời đưa ra quy trình thực thi kiểm thử tự động, phương hướng giải quyết bài toán và đưa ra báo cáo cho kết quả kiểm thử.

3.1. Thiết kế test case cho các chức năng chính

Thiết kế test case: Mục đích nhằm chỉ định các test case và các bước kiểm thử chi tiết cho mỗi phiên bản phần mềm. Giai đoạn thiết kế test case là hết sức quan trọng, nó bảo đảm tất cả các tình huống kiểm tra “quét” hết tất cả yêu cầu cần kiểm tra.

3.1.1. Chức năng đăng ký

a. Đặc tả kỹ thuật của chức năng đăng ký

Để sử dụng chức năng đặt hàng của hệ thống thì người dùng bắt buộc phải có tài khoản và đăng nhập vào hệ thống. Khi chưa có tài khoản truy cập hệ thống, khách hàng có thể đăng ký tài khoản mới.

❖ Trang đăng ký tài khoản mới gồm:

- Tên tài khoản, mật khẩu, xác nhận mật khẩu, tên người dùng, địa chỉ, giới tính, email, số điện thoại. Trong đó: giới tính có hai lựa chọn là “Male”, “Female” và giá trị được để mặc định “Male”.
- Một nút “Register”: Khi click vào button này thì hệ thống sẽ thông báo đăng ký thành công nếu các trường đăng ký đều được nhập các giá trị hợp lệ.

Tên trường	Bắt buộc	Khoảng hợp lệ	Khác
Tên tài khoản	Có	3 - 20 kí tự chữ	Không được bỏ trống Tên đăng nhập chưa tồn tại trong cơ sở dữ liệu
Mật khẩu	Có	6 - 20 kí tự chữ	Không được bỏ trống
Xác nhận mật khẩu	Có	6 - 20 kí tự chữ	Không được bỏ trống Phải trùng với Mật khẩu được nhập ở trường Password
Tên người dùng	Có	Không	Không được bỏ trống
Địa chỉ	Có	Không	Không được bỏ trống
Email	Có	Không	Không được bỏ trống Phải nhập đúng định dạng email
Số điện thoại	Có	Chỉ bao gồm 10 hoặc 11 kí tự số	Không được bỏ trống Chỉ bao gồm các kí tự số Phải là số nguyên dương Không chứa dấu cách Không chứa các kí tự

Bảng 3.1: Mô tả yêu cầu chức năng đăng ký

b. Phân tích các trường hợp kiểm thử**❖ Test Validate (Kiểm tra tính hợp lệ của các giá trị đầu vào)**

- Account:
 - Dựa vào khoảng hợp lệ đã được mô tả trong đặc tả kỹ thuật, sử dụng kỹ thuật phân vùng tương đương, chúng ta xác định được 3 vùng giá trị (số lượng) cần test như sau:
 - + Vùng hợp lệ: vùng từ 3 đến 20 kí tự chữ
 - + Vùng không hợp lệ: vùng ít hơn 3 kí tự, vùng từ 21 kí tự chữ trở lên
 - Dựa vào các yêu cầu khác, chúng ta có 3 vùng tương đương nữa là
 - + Vùng hợp lệ: tên chưa tồn tại
 - + Vùng không hợp lệ: vùng khoảng trắng, tên đã có trong cơ sở dữ liệu
- Password:
 - + Vùng hợp lệ: vùng từ 6 đến 20 kí tự chữ
 - + Vùng không hợp lệ: vùng ít hơn 6 kí tự, vùng từ 21 kí tự chữ trở lên
- Reenter password:
 - Dựa vào khoảng hợp lệ đã được mô tả trong đặc tả kỹ thuật, sử dụng kỹ thuật phân vùng tương đương, chúng ta xác định được 3 vùng giá trị (số lượng) cần test như sau:
 - + Vùng hợp lệ: vùng từ 6 đến 20 kí tự chữ
 - + Vùng không hợp lệ: vùng ít hơn 6 kí tự, vùng từ 21 kí tự chữ trở lên
 - Dựa vào các yêu cầu khác, chúng ta có 2 vùng tương đương nữa là:
 - + Vùng hợp lệ: Reenter password trùng với Password
 - + Vùng không hợp lệ: Reenter password không trùng với Password
- Name:
 - + Vùng hợp lệ: vùng kí tự chữ, kí tự số
 - + Vùng không hợp lệ: khoảng trắng
- Address:
 - + Vùng hợp lệ: vùng kí tự chữ, kí tự số
 - + Vùng không hợp lệ: khoảng trắng
- Email:
 - + Vùng hợp lệ: vùng kí tự chữ, kí tự số
 - + Vùng không hợp lệ: vùng khoảng trắng, vùng vi phạm định dạng email
- Phonenumber:
 - Dựa vào khoảng hợp lệ đã được mô tả trong đặc tả kỹ thuật, sử dụng kỹ thuật phân vùng tương đương, chúng ta xác định được 3 vùng giá trị (số lượng) cần test như sau:

- + Vùng hợp lệ: vùng từ 10 đến 11 kí tự số
- + Vùng không hợp lệ: vùng ít hơn 10 kí tự, vùng từ 12 kí tự chữ trở lên
- Dựa vào các yêu cầu khác, chúng ta có 2 vùng tương đương nữa là
 - + Vùng hợp lệ: vùng kí tự số
 - + Vùng không hợp lệ: vùng kí tự chữ, vùng khoảng trắng, vùng số nguyên âm.
- ❖ Thiết kế các test case

Nội dung chi tiết test case và kết quả test chức năng đăng ký tài khoản được trình bày ở định dạng Excel được đính kèm trong phần Phụ lục: Register.xlsx của đồ án.

STT	Hành động	Phản hồi từ hệ thống
Trường hợp đăng ký thành công		
	1. Nhập giá trị hợp lệ vào các trường: <ul style="list-style-type: none"> • Account • Password • Reenter password • Name • Address • Gender • Email • Phonenumber 2. Click nút: “Register”	Hệ thống hiển thị message tương ứng: “Register successfully!”
Nhập vào giá trị trống		
1	1. Để trống một trong các trường: <ul style="list-style-type: none"> • Account • Password • Reenter password • Name • Address • Email • Phonenumber 2. Click nút: “Register”	Hệ thống hiển thị message lỗi tương ứng: “These field must not be empty!”
Nhập giá trị không hợp lệ vào trường Account		

1	1. Nhập tên tài khoản ít hơn 3 ký tự 2. Click nút: “Register”	Hệ thống hiển thị message lỗi tương ứng: “Account have to ≥ 3 and ≤ 20 characters!”
2	1. Nhập tên tài khoản nhiều hơn 20 ký tự 2. Click nút: “Register”	Hệ thống hiển thị message lỗi tương ứng: “Account have to ≥ 3 and ≤ 20 characters!”
3	1. Nhập vào tên tài khoản có chứa thẻ HTML 2. Click nút: “Register”	Hệ thống hiển thị message lỗi tương ứng: “Error account!”
4	1. Nhập tên tài khoản đã tồn tại trong cơ sở dữ liệu 2. Click nút: “Register”	Hệ thống hiển thị message lỗi tương ứng: “Account is exists!”
Nhập giá trị không hợp lệ vào trường password		
1	1. Nhập mật khẩu ít hơn 6 ký tự 2. Click nút: “Register”	Hệ thống hiển thị message lỗi tương ứng: “Password have to ≥ 6 and ≤ 20 characters!”
2	1. Nhập mật khẩu nhiều hơn 20 ký tự 2. Click nút: “Register”	Hệ thống hiển thị message lỗi tương ứng: “Password have to ≥ 6 and ≤ 20 characters!”
Nhập sai xác nhận mật khẩu		
	1. Nhập sai xác nhận mật khẩu 2. Click nút: “Register”	Hệ thống hiển thị message lỗi tương ứng: “Error reenter password!”
Nhập sai định dạng email		
1	1. Nhập sai định dạng email 2. Click nút: “Register”	Hệ thống hiển thị message lỗi tương ứng: “Error email!”
Nhập giá trị không hợp lệ vào trường số điện thoại		
1	1. Nhập chữ cái vào trường số điện thoại 2. Click nút: “Register”	Hệ thống hiển thị message lỗi tương ứng: “Error phonenumber!”
2	1. Nhập số điện thoại ít hơn 10 số 2. Click nút: “Register”	Hệ thống hiển thị message lỗi tương ứng: “Phonenumber have to 10 or 11 number!”

3	1. Nhập số điện thoại nhiều hơn 11 số 2. Click nút: "Register"	Hệ thống hiển thị message lỗi tương ứng: "Phonenumber have to 10 or 11 number!"
4	1. Nhập số âm vào trường số điện thoại 2. Click nút: "Register"	Hệ thống hiển thị message lỗi tương ứng: "Error phonenumber!"

Bảng 3.2: Các trường hợp test chức năng đăng ký**3.1.2. Chức năng thêm sản phẩm vào giỏ hàng****a. Đặc tả kỹ thuật của chức năng Thêm sản phẩm vào giỏ hàng.**

Khi truy cập vào hệ thống, khách hàng có thể thêm sản phẩm vào giỏ hàng từ giao diện xem thông tin sản phẩm. Trang xem thông tin sản phẩm gồm:

- Thông tin chi tiết về sản phẩm bao gồm: tên sản phẩm, thể loại, giá sản phẩm, trạng thái sản phẩm
- Ô nhập số lượng sản phẩm thêm vào giỏ hàng
- Nút "Add to carts"

Nếu số lượng sản phẩm hợp lệ, khi click "Add to carts" sẽ chuyển đến trang xem chi tiết giỏ hàng với sản phẩm đã được thêm vào giỏ. Nếu số lượng sản phẩm không hợp lệ sẽ có thông báo lỗi hiển thị trên trang xem thông tin sản phẩm.

❖ Mô tả yêu cầu:

Tên trường	Bắt buộc	Khoảng hợp lệ	Khác
Quantity	Có	Giá trị tối thiểu là 1 và tối đa là 50	Phải là một số dương Không được bỏ trống Không chứa dấu cách Không được chứa các kí tự chữ

Bảng 3.4: Mô tả yêu cầu chức năng thêm sản phẩm vào giỏ hàng**b. Phân tích các trường hợp kiểm thử**

❖ Test Validate (Kiểm tra tính hợp lệ của các giá trị đầu vào)

- Dựa vào khoảng hợp lệ đã được mô tả trong đặc tả kỹ thuật, sử dụng kỹ thuật phân vùng tương đương, chúng ta xác định được 3 vùng giá trị (số lượng) cần test như sau:
 - + Vùng hợp lệ: vùng giá trị từ 1 đến 50
 - + Vùng không hợp lệ: vùng giá trị nhỏ hơn 1, vùng giá trị lớn hơn 50.
- Dựa vào các yêu cầu khác, chúng ta có 3 vùng tương đương nữa là
 - + Vùng hợp lệ: vùng kí tự số (0 - 9)
 - + Vùng không hợp lệ: vùng khoảng trống, vùng kí tự chữ

❖ Test chức năng màn hình

• Các trường hợp cần test:

1. Trong giỏ hàng đã có sản phẩm, số lượng sản phẩm: hợp lệ
2. Trong giỏ hàng đã có sản phẩm, số lượng sản phẩm: không hợp lệ
3. Trong giỏ hàng chưa có sản phẩm, số lượng sản phẩm: hợp lệ
4. Trong giỏ hàng chưa có sản phẩm, số lượng sản phẩm: không hợp lệ

Trong mỗi điều kiện “hợp lệ” và “không hợp lệ” của số lượng sản phẩm thì có nhiều trường hợp khác nhau tương ứng với các giá trị trong mỗi vùng tương đương.

• Nút “Add to carts”

Nếu số lượng sản phẩm hợp lệ, khi click “Add to carts” sẽ chuyển đến trang xem chi tiết giỏ hàng với sản phẩm đã được thêm vào giỏ. Nếu số lượng sản phẩm không hợp lệ sẽ có thông báo lỗi hiển thị trên trang xem thông tin sản phẩm.

❖ Thiết kế các test case

Nội dung chi tiết test case và kết quả test chức năng Thêm sản phẩm vào giỏ hàng được trình bày ở định dạng Excel được đính kèm trong phần Phụ lục: AddToCarts.xlsx của đồ án.

STT	Hành động	Phản hồi từ hệ thống
Nhập vào giá trị số lượng hợp lệ		
1	1. Nhập giá trị hợp lệ vào trường: Quantity 2. Click nút: “Add to carts”	Hệ thống hiển thị message tương ứng: “Add carts successfully!”
Nhập vào giá trị trống		
1	1. Để trống trường: Quantity 2. Click nút: “Add to carts”	Hệ thống hiển thị message tương ứng: “Quantity have to a number!”
Nhập sai định dạng		
1	1. Nhập giá trị âm vào trường: Quantity 2. Click nút: “Add to carts”	Hệ thống hiển thị message tương ứng: “Add product to carts is unsuccessfully!”
2	1. Nhập ký tự vào trường số lượng 2. Click nút: “Add to carts”	Hệ thống hiển thị message tương ứng: “Quantity have to a number!”
3	1. Nhập vào số lượng lớn hơn số lượng sản phẩm còn lại trong cơ sở	Hệ thống hiển thị message tương ứng: “Add product to carts is unsuccessfully!”

	dữ liệu. 2. Click nút: “Add to carts”	
4	1. Nhập số lượng giá trị 0 vào trường số lượng. 2. Click nút: “Add to carts”	Hệ thống hiển thị message tương ứng: “Add product to carts is unsuccessfully!”
5	1. Nhập số lượng lớn hơn 50. 2. Click nút: “Add to carts”	Hệ thống hiển thị message tương ứng: “Add product to carts is unsuccessfully!”

Bảng 3.5: Các trường hợp test chức năng thêm sản phẩm vào giỏ hàng**3.1.3. Chức năng sửa giỏ hàng****a. Đặc tả kỹ thuật của chức năng Sửa giỏ hàng.**

Khi truy cập vào hệ thống, khách hàng có thể thêm sản phẩm vào giỏ hàng từ giao diện xem thông tin sản phẩm. Khi trong giỏ hàng đã có sản phẩm, khách hàng có thể thực hiện sửa số lượng sản phẩm trong giỏ hàng. Trang xem chi tiết giỏ hàng gồm:

- Danh các sản phẩm sản phẩm, mỗi sản phẩm gồm các thông tin: hình ảnh, tên sản phẩm, giá một đơn vị sản phẩm, số lượng sản phẩm, tổng tiền, một link Delete sản phẩm. Trong đó: thông tin số lượng sản phẩm có thể sửa.
- Một nút “Continue shopping”: Khi click vào button này thì sẽ được chuyển đến trang giao diện chính của hệ thống để khách hàng tiếp tục xem các sản phẩm.
- Một nút “Update shopping carts”:

Nếu số lượng các sản phẩm trong giỏ hàng hợp lệ, khi click “Update shopping carts”, hệ thống sẽ cập nhật lại số lượng sản phẩm và hiển thị thông báo thành công: “Update shopping carts successfully!” trên trang xem chi tiết giỏ hàng.

Nếu trong giỏ hàng có số lượng tương ứng với sản phẩm không hợp lệ thì sẽ có thông báo lỗi hiển thị trên trang xem chi tiết giỏ hàng, tương ứng với tên sản phẩm có số lượng không hợp lệ.

- Một nút “Proceed to checkout”: khách hàng có thể đặt hàng nếu trong giỏ hàng có sản phẩm

❖ Mô tả yêu cầu:

Tên trường	Bắt buộc	Khoảng hợp lệ	Khác
Quantity	Có	Giá trị tối thiểu là 1 và tối đa là 50	<ul style="list-style-type: none"> • Phải là một số dương • Không được bỏ trống • Không chứa dấu cách • Không được chứa các ký tự chữ

Bảng 3.7: Mô tả yêu cầu chức năng sửa giỏ hàng

b. Phân tích các trường hợp kiểm thử**❖ Test Validate (Kiểm tra tính hợp lệ của các giá trị đầu vào)**

- Dựa vào khoảng hợp lệ đã được mô tả trong đặc tả kỹ thuật, sử dụng kỹ thuật phân vùng tương đương, chúng ta xác định được 3 vùng giá trị (số lượng) cần test như sau:

- + Vùng hợp lệ: vùng giá trị từ 1 đến 50
- + Vùng không hợp lệ: vùng giá trị nhỏ hơn 1, vùng giá trị lớn hơn 50.

- Dựa vào các yêu cầu khác, chúng ta có 3 vùng tương đương nữa là

- + Vùng hợp lệ: vùng kí tự số (0 - 9)
- + Vùng không hợp lệ: vùng khoảng trắng, vùng kí tự chữ

❖ Test chức năng màn hình

- Các trường hợp cần test:

1. Trong giỏ hàng đã có sản phẩm, số lượng sản phẩm: hợp lệ
2. Trong giỏ hàng đã có sản phẩm, số lượng sản phẩm: không hợp lệ

Trong mỗi điều kiện “hợp lệ” và “không hợp lệ” của số lượng sản phẩm thì có nhiều trường hợp khác nhau tương ứng với các giá trị trong mỗi vùng tương đương.

❖ Kiểm tra các link trên trang xem chi tiết giỏ hàng

- Link “Delete”: khi click vào “Delete” thì sản phẩm tương ứng sẽ bị xóa khỏi giỏ hàng.
- Một nút “Continue shopping”: Khi click vào button này thì sẽ được chuyển đến trang giao diện chính của hệ thống để khách hàng tiếp tục xem các sản phẩm.
- Một nút “Update shopping carts”:
 - Nếu số lượng các sản phẩm trong giỏ hàng hợp lệ, khi click “Update shopping carts”, hệ thống sẽ cập nhật lại số lượng sản phẩm và hiển thị thông báo thành công: “Update shopping carts successfully!” trên trang xem chi tiết giỏ hàng.
 - Nếu trong giỏ hàng có số lượng tương ứng với sản phẩm không hợp lệ thì sẽ có thông báo lỗi hiển thị trên trang xem chi tiết giỏ hàng, tương ứng với tên sản phẩm có số lượng không hợp lệ.
- Một nút “Proceed to checkout”: khách hàng có thể đặt hàng nếu trong giỏ hàng có sản phẩm

❖ Thiết kế các test case

Nội dung chi tiết test case và kết quả test chức năng Thêm sản phẩm vào giỏ hàng được trình bày ở định dạng Excel được đính kèm trong phần Phụ lục: EditCarts.xlsx của đồ án.

- Chức năng thêm sản phẩm vào giỏ hàng

STT	Hành động	Phản hồi từ hệ thống
Nhập vào giá trị số lượng hợp lệ		
1	1. Nhập giá trị hợp lệ vào các trường: Quantity 2. Click nút: “Update shopping carts”	Hệ thống hiển thị message tương ứng: “Add carts successfully!”
Nhập vào giá trị trống		
1	1. Để trống trường: Quantity 2. Click nút: “Update shopping carts”	Hệ thống hiển thị message tương ứng: “Quantity have to a number!”
Nhập sai định dạng		
1	1. Nhập giá trị âm vào trường: Quantity 2. Click nút: “Update shopping carts”	Hệ thống hiển thị message thông báo lỗi tương ứng với sản phẩm có số lượng không hợp lệ.
2	1. Nhập ký tự vào trường: Quantity 2. Click nút: “Update shopping carts”	Hệ thống hiển thị thông báo lỗi: "Formatting error!"
3	1. Nhập vào số lượng lớn hơn số lượng sản phẩm còn lại trong cơ sở dữ liệu. 2. Click nút: “Update shopping carts”	Hệ thống hiển thị message thông báo lỗi tương ứng với sản phẩm có số lượng không hợp lệ.
4	1. Nhập số lượng giá trị 0 vào trường: quantity. 2. Click nút: “Update shopping carts”	Hệ thống hiển thị message thông báo lỗi tương ứng với sản phẩm có số lượng không hợp lệ.
5	1. Nhập số lượng lớn hơn 50. 2. Click nút: “Update shopping carts”	Hệ thống hiển thị message thông báo lỗi tương ứng với sản phẩm có số lượng không hợp lệ.

Bảng 3.8: Các trường hợp test chức năng sửa giỏ hàng

3.1.4. Chức năng đặt hàng

a. Đặc tả kỹ thuật của chức năng Sửa giỏ hàng.

Khi đã đăng nhập vào hệ thống, khách hàng có thể đặt hàng khi trong giỏ hàng đã có sản phẩm. Trang xem đặt hàng gồm:

- Danh sách các sản phẩm thành viên muốn đặt hàng bao gồm: hình ảnh, tên sản phẩm, giá một đơn vị sản phẩm, số lượng sản phẩm, tổng tiền.

- Form nhập thông tin giao hàng bao gồm: tên người nhận, địa chỉ người nhận, địa chỉ email, số điện thoại
 - Một nút “Order”
 - Nếu số lượng các thông tin người nhận hợp lệ, khi click “Order”, hệ thống sẽ chuyển sang trang xem hóa đơn và hiển thị thông báo đặt hàng thành công: “Order successfully!”.
 - Nếu trong giỏ hàng có số lượng tương ứng với sản phẩm không hợp lệ thì sẽ có thông báo lỗi hiển thị trên trang xem chi tiết giỏ hàng, tương ứng với tên sản phẩm có số lượng không hợp lệ.
 - Một nút “Cancel”: Khi click nút “Cancel”, hệ thống sẽ quay trở về giao diện chính để khách hàng tiếp tục xem các sản phẩm.
- ❖ Mô tả yêu cầu:

Tên trường	Bắt buộc	Khoảng hợp lệ	Khác
Tên người dùng	Có	Không	Không được bỏ trống
Địa chỉ	Có	Không	Không được bỏ trống
Email	Có	Không	Không được bỏ trống Phải nhập đúng định dạng email
Số điện thoại	Có	Chỉ bao gồm 10 hoặc 11 kí tự số	Không được bỏ trống Chỉ bao gồm các kí tự số Phải là số nguyên dương Không chứa dấu cách Không chứa các kí tự chữ

Bảng 3.10: Mô tả yêu cầu chức năng đăng ký

b. Phân tích các trường hợp kiểm thử

- ❖ Test Validate (Kiểm tra tính hợp lệ của các giá trị đầu vào)
- Name:
 - + Vùng hợp lệ: vùng kí tự chữ, kí tự số
 - + Vùng không hợp lệ: khoảng trắng
 - Address:
 - + Vùng hợp lệ: vùng kí tự chữ, kí tự số
 - + Vùng không hợp lệ: khoảng trắng
 - Email:

- + Vùng hợp lệ: vùng kí tự chữ, kí tự số
- + Vùng không hợp lệ: vùng khoảng trắng, vùng vai định dạng email
- **Phonenumber:**
 - Dựa vào khoảng hợp lệ đã được mô tả trong đặc tả kỹ thuật, sử dụng kỹ thuật phân vùng tương đương, chúng ta xác định được 3 vùng giá trị (số lượng) cần test như sau:
 - + Vùng hợp lệ: vùng từ 10 đến 11 kí tự số
 - + Vùng không hợp lệ: vùng ít hơn 10 kí tự, vùng từ 12 kí tự chữ trở lên
 - Dựa vào các yêu cầu khác, chúng ta có 2 vùng tương đương nữa là
 - + Vùng hợp lệ: vùng kí tự số
 - + Vùng không hợp lệ: vùng kí tự chữ, vùng khoảng trắng, vùng số nguyên âm,

❖ Thiết kế các test case

Nội dung chi tiết test case và kết quả test chức năng đặt hàng được trình bày ở định dạng Excel được đính kèm trong phần Phụ lục: Order.xlsx của đồ án.

STT	Hành động	Phản hồi từ hệ thống
Trường hợp đặt hàng thành công		
1	1. Nhập giá trị hợp lệ vào các trường: <ul style="list-style-type: none"> • Name • Address • Gender • Email • Phonenumber 2. Click nút: "Order"	Hệ thống hiển thị message tương ứng: "Order successfully!"
Nhập vào giá trị trống		
1	1. Để trống một trong các trường: <ul style="list-style-type: none"> • Name • Address • Email • Phonenumber 2. Click nút: "Order"	Hệ thống hiển thị message lỗi tương ứng: "These field must not be empty!"
Nhập sai định dạng email		
1	1. Nhập sai định dạng email 2. Click nút: "Order"	Hệ thống hiển thị message lỗi tương ứng: "Error email!"

Nhập giá trị không hợp lệ vào trường số điện thoại		
1	1. Nhập chữ cái vào trường số điện thoại 2. Click nút: "Order"	Hệ thống hiển thị message lỗi tương ứng: "Error phonenumber!"
2	1. Nhập số điện thoại ít hơn 10 số 2. Click nút: "Order"	Hệ thống hiển thị message lỗi tương ứng: "Phonenumber have to 10 or 11 number!"
3	1. Nhập số điện thoại nhiều hơn 11 số 2. Click nút: "Order"	Hệ thống hiển thị message lỗi tương ứng: "Phonenumber have to 10 or 11 number!"
4	1. Nhập số âm vào trường số điện thoại 2. Click nút: "Order"	Hệ thống hiển thị message lỗi tương ứng: "Error phonenumber!"

Bảng 3.11: Các trường hợp test chức năng đặt hàng

3.2. Hướng giải quyết bài toán và thực thi kiểm thử tự động

3.2.1. Các yêu cầu về công nghệ

- Trình duyệt Mozilla firefox đã cài Add-on Selenium IDE
- Công cụ lập trình Eclipse đã được cấu hình với Selenium WebDriver và đã được cài đặt framework TestNG.
- Hệ quản trị cơ sở dữ liệu quan hệ SQL Server
- Bộ công cụ phát triển ứng dụng bằng ngôn ngữ lập trình Java: JDK (Java development Kit)
- Khác: Microsoft Excel 2003 trở lên.

3.2.2. Hướng thực thi kiểm thử tự động

a. Giải pháp

Để tạo ra được ứng dụng Web có hiệu năng cao, đáng tin cậy thì sau khi xây dựng website, cần phải kiểm thử ứng dụng đó một cách tỉ mỉ, cẩn thận và chặt chẽ. Về mặt bản chất, các ứng dụng Web cũng là phần mềm, nên các loại kiểm thử áp dụng cho phần mềm cũng được áp dụng khi kiểm thử ứng dụng Web. Tuy nhiên, một ứng dụng Web thường có rất nhiều nhóm người sử dụng với nhiều nền tảng khác nhau (hệ điều hành, trình duyệt...), chúng ta cũng rất khó có thể đoán được số lượng người sử dụng một ứng dụng Web là bao nhiêu, thời gian hồi đáp yêu cầu của người sử dụng đối với ứng dụng ...dẫn đến việc kiểm thử ứng dụng Web sẽ có những khác biệt nhất định so với kiểm thử phần

mềm truyền thống. Trong đó, kiểm thử giao diện người dùng, kiểm thử hiệu năng và kiểm thử bảo mật là những loại kiểm thử mà ứng dụng Web cần chú trọng.

Dưới đây là các bước kiểm thử ứng dụng Web:

- **Kiểm thử chức năng**

Kiểm tra form của các trang như: không nhập gì vào các trường bắt buộc, nhập dữ liệu đầu vào không đúng validate...Thực hiện test tất cả các link trong trang web, định dạng được sử dụng trong các trang Web để gửi và nhận các thông tin cần thiết từ người dùng.

- **Kiểm thử tính khả dụng**

Tính khả dụng của trang Web được định nghĩa là trang Web dễ sử dụng, mỗi trang đều có menu chính và menu này phải nhất quán. Ngoài ra, khi kiểm thử tính khả dụng cần thực hiện kiểm thử các điều khiển chuyển hướng như button, text box, link test..., nội dung của trang Web phải dễ hiểu và thân thiện với người sử dụng. Không có lỗi chính tả, không khó để đọc chữ, hình ảnh được sắp xếp gọn gàng, hợp lý.

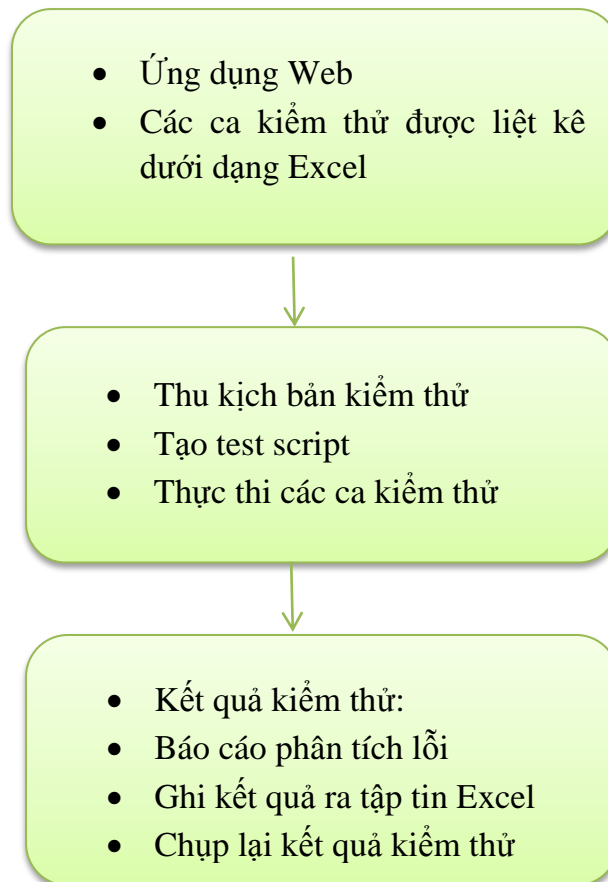
- **Kiểm thử khả năng tương thích**

Người dùng khác nhau có thể sử dụng trình duyệt khác nhau tùy theo nhu cầu, thói quen...của họ. Cần phải kiểm tra sự tương thích của ứng dụng Web trên nhiều trình duyệt như IE, Firefox, Chrome, Safari, Opera...

- **Kiểm thử bảo mật**

Đồ án thực hiện kiểm thử các tính năng bảo mật sau:

- Gõ trực tiếp url vào thanh địa chỉ của trình duyệt mà không qua thực hiện các yêu cầu bắt buộc trước khi truy cập vào trang đó.
- Thay đổi url trực tiếp bằng cách đổi tham số kiểm tra tính hợp lệ của trang đó để chuyển tới một trang khác.
- Thử các giá trị đầu vào không hợp lệ vào các trường của trang Web.
- Mô hình giải quyết vấn đề được mô tả tại Hình 3.1 với các tính năng cụ thể như sau:
- Phần đầu vào của chương trình sẽ là:
 - Một ứng dụng Web được sử dụng để kiểm thử với tập hợp các chức năng cơ bản của một website bán hàng online.
 - Tập hợp các ca kiểm thử với từng chức năng của hệ thống được liệt kê dưới dạng Excel.
- Phần tiếp theo của chương trình là thu lại kịch bản kiểm thử với Selenium IDE, tạo test script và thực thi các ca kiểm thử với Selenium WebDriver và framework TestNG
- Trong phần kết quả, đồ án sẽ đưa ra được kết quả kiểm thử các chức năng của ứng dụng, các chức năng lỗi so với ca kiểm thử và ghi vào file Excel. Đồng thời, chụp lại màn hình kết quả kiểm thử và xem báo cáo phân tích lỗi được tự động sinh ra bằng TestNG.



Hình 3.1: Quá trình kiểm thử tự động

Từ các phân tích ở trên, xây dựng chương trình thực hiện việc kiểm thử tự động qua các bước:

- Thu lại kịch bản kiểm thử với Selenium IDE
- Tạo test script để thực hiện kiểm thử tự động với Selenium WebDriver
- Viết hàm đọc dữ liệu các trường hợp kiểm thử có thể xảy ra từ tập tin Excel để truyền vào hàm test.
- Viết hàm ghi kết quả ra tập tin Excel và viết hàm chụp lại tất cả các kết quả kiểm thử xảy ra lỗi.
- Xem báo cáo và phân tích lỗi.

b. Kiểm thử chức năng

- ❖ Quá trình kiểm thử chức năng sử dụng dữ liệu động được thực hiện qua 3 bước sau:
 - Dùng Selenium IDE để tạo test script.
 - Chạy test script bằng ngôn ngữ Java trên Eclipse.
 - Đọc dữ liệu từ tập tin Excel để lấy dữ liệu chạy test script

❖ Bước 1: Tạo test script

Phần này của đồ án sẽ tạo một kịch bản kiểm thử bằng cách sử dụng công cụ Selenium IDE để thu lại các thao tác với website dựa trên các hành động được thực hiện trên trang Web. Sau đó, Export kịch bản kiểm thử vừa thu được sang ngôn ngữ Java(TestNG) - WebDriver Backed.

- Test script sau khi được export có dạng như sau:

```
@Test
public void testEditCarts() {
    selenium.open("/Fashionista/Product_view.jsp?idProduct=9");
    selenium.type("name=quantity", "2");
    selenium.click("css=input.bay");
    selenium.waitForPageToLoad("30000");
    assertEquals(selenium.getText("css=h3"), "Add carts successfully!");
    selenium.type("name=quantity0", "1");
    selenium.click("name=updateShopping");
    selenium.waitForPageToLoad("30000");
    assertEquals(selenium.getText("css=h3"), "Update shopping carts successfully!");
}
```

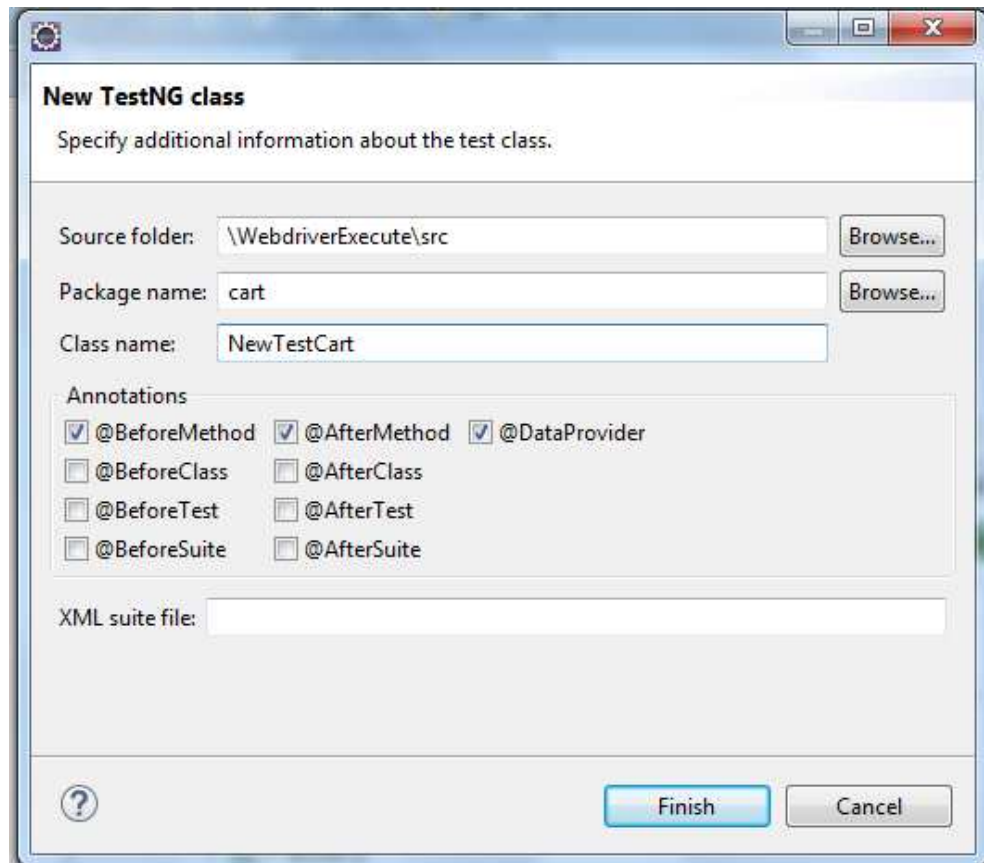
Bảng 3.13: Test script được export từ kịch bản kiểm thử Selenium IDE

Tuy nhiên, test script này chỉ có thể kiểm tra được duy nhất một test case với dữ liệu cố định đã được truyền trực tiếp vào trong hàm *@Test*. Khi muốn kiểm thử một chức năng của trang Web với lượng dữ liệu lớn hơn thì chúng ta sẽ phải thay đổi từng giá trị dữ liệu đã được truyền vào trong hàm *@Test* đó. Vì vậy, một giải pháp được đưa ra là đưa test script này vào Eclipse cùng với sự hỗ trợ của TestNG để tối ưu hóa việc thực thi nhiều test case trên cùng một kịch bản kiểm thử.

❖ Bước 2: Đưa test script vào Eclipse[11].

Để tránh việc phải sửa đổi mã code nhiều lần khi thay đổi dữ liệu kiểm thử, chúng ta sẽ tạo một Project mới để chứa class TestNG. Sau đó, test script đã được export từ Bước 1 sẽ được đưa vào Eclipse để thực thi.

Trong trường hợp có nhiều hơn một bộ dữ liệu kiểm thử, chúng ta cần thay đổi các giá trị cố định trong test script bằng các tham số. Khi đó, việc thực thi một lượng lớn test case sẽ trở nên dễ dàng hơn bằng cách truyền dữ liệu kiểm thử vào các tham số trong hàm *@Test*. Để có thể làm được việc này thì khi tạo mới class TestNG, chúng ta cần click chọn vào “*@DataProvider*” như Hình 3.2.



Hình 3.2: Tạo class TestNG

❖ Bước 3: Đọc dữ liệu từ đầu vào từ file Excel sử dụng data driven

Do Selenium IDE không hỗ trợ đưa dữ liệu kiểm thử vào tham số trong test script nên cần phải sử dụng framework TestNG. Trong đó, TestNG có cung cấp một đối tượng “DataProvider” dùng để lấy dữ liệu động, dữ liệu được khai báo trong “DataProvider” sẽ được truyền vào trong “@Test” để chạy test script với các dữ liệu đó. Tuy nhiên, “DataProvider” lại không có khả năng lấy dữ liệu từ một file bất kỳ nào đó. Do đó, khi muốn lấy dữ liệu trong một file Excel để đưa vào trong "DataProvider" thì chúng ta phải viết thêm hàm đọc dữ liệu từ file Excel bằng ngôn ngữ Java.

Trước khi viết hàm đọc dữ liệu đầu vào từ file Excel, dựa vào những phân tích ở mục 3.1. Thiết kế test case cho một số chức năng, chúng ta sẽ tạo một file Excel mới chứa các dữ liệu đầu vào như đã được mô tả dưới Hình 3.3:

	A	B	C	D	E
1	Test case Add To Carts				
2					
3	ID product	Quantity 1	Expect result add product	Quantity 2	Expect result edit product
4	9	1	Add carts successfully!	2	Update shopping carts successfully!
5	9	1	Add carts successfully!		Formatting error!
6	9	1	Add carts successfully!	-1	Error quantity of SKIRT FASHIONISTA 001
7	9	1	Add carts successfully!	a	Formatting error!
8	9	1	Add carts successfully!	1000	Error quantity of SKIRT FASHIONISTA 001
9	9	1	Add carts successfully!	0	Error quantity of SKIRT FASHIONISTA 001
10	9	1	Add carts successfully!	51	Error quantity of SKIRT FASHIONISTA 001

Hình 3.3: Các Test case cho chức năng Sửa giỏ hàng

Trong đó:

- ID product và Quantity 1 lần lượt là ID và số lượng sản phẩm thêm vào giỏ hàng.
 - Expect result add product: Kết quả mong muốn được hiển thị khi thêm sản phẩm vào giỏ hàng.
 - Quantity 2 là số lượng sản phẩm muốn sửa.
 - Expect result edit product : kết quả mong muốn được hiển thị sau khi sửa thành công.
- Viết hàm đọc dữ liệu đầu vào từ file Excel với thư viện JXL
Để đọc được dữ liệu từ file Excel, chúng ta cần dùng đến thư viện hỗ trợ “jxl.jar”. Trong thư viện này sẽ hỗ trợ một số câu lệnh như:
 - Tạo workbook để trỏ đến file Excel muốn đọc.
 - Lấy sheet muốn đọc theo vị trí sheet hoặc theo tên sheet.
 - Lấy số lượng hàng, số lượng cột trong sheet đó.
 - Đọc nội dung trong từng ô theo vị trí hàng và cột.
 - Gọi trực tiếp hàm đọc dữ liệu file Excel trong hàm @DataProvider của TestNG để gửi dữ liệu vào hàm @Test.

@DataProvider

```

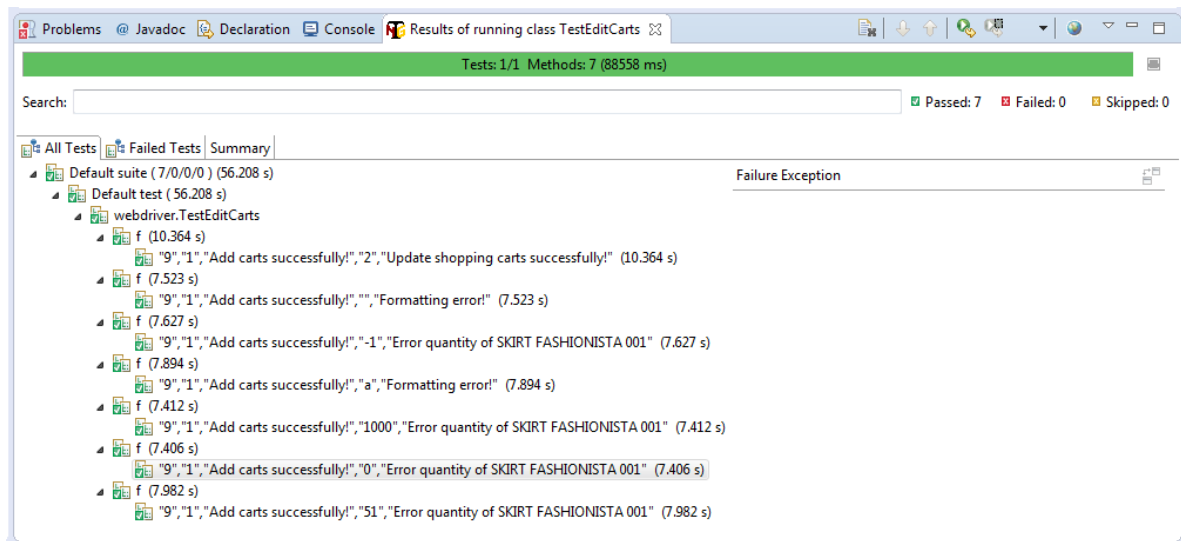
public Object[][] dp() {
    Object[][] listData = getData(inputFile);
    return listData;
}

```

Trong đó: **inputFile** là đường dẫn địa chỉ tới file Excel chứa dữ liệu.

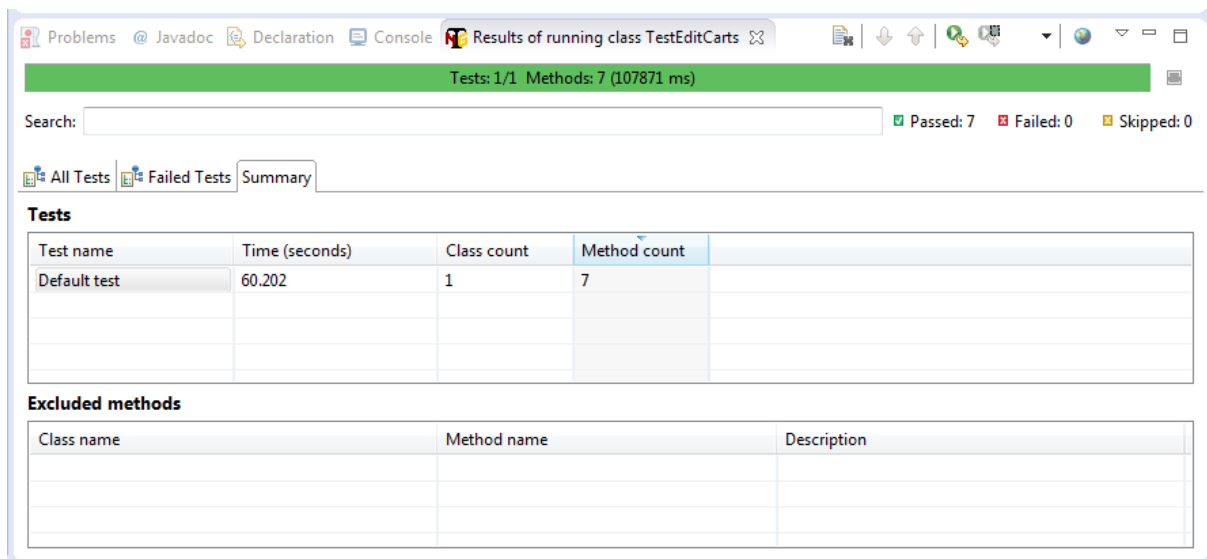
❖ Chạy test script và xem kết quả hiển thị

- Kết quả chạy test case được hiển thị như Hình 3.4:



Hình 3.4: Kết quả chạy test case trên TestNG –All Tests

Kết quả hiển thị có ba tab con: “All Tests”, “Failed Tests” and “Summary”. Khi click vào tab “Summary”, kết quả sẽ hiển thị như Hình 3.5:



Hình 3.5: Kết quả chạy test case trên TestNG – Summary

c. Kiểm thử trên đa trình duyệt

Một trong những tiêu chí đánh giá chất lượng của một website là khả năng sử dụng trên nhiều trình duyệt khác nhau, do những người dùng khác nhau có thể sử dụng trình duyệt khác nhau tùy theo nhu cầu, thói quen... của họ. Cần phải kiểm tra sự tương thích của ứng dụng Web trên nhiều trình duyệt như IE, Firefox, Chrome, Safari, Opera...

Selenium WebDriver hỗ trợ việc chạy các test case trên nhiều trình duyệt. Vì vậy để kiểm thử khả năng tương thích của website, ta thực hiện một test script trên 3 trình duyệt phổ biến nhất hiện nay là: Internet Explorer, Mozilla Firefox, Chrome. Tuy nhiên, để chạy

được trên hai trình duyệt Chrome và Internet Explorer cần tải về thư viện ChromeDriverServer và Internet Explorer Driver Server.

Để thực hiện kiểm thử trên đa trình duyệt, viết một hàm cho phép lựa chọn trình duyệt thực thi kiểm thử:

```
@BeforeMethod
public void beforeMethod() {
    Scanner sc = new Scanner(System.in);
    int n;
    boolean check = false;
    do {
        System.out.println("Please enter number: \n1. Mozilla Firefox\n2. Chrome\n3. Internet Explorer");
        try {
            n = Integer.parseInt(sc.nextLine());
            if (n == 1) {
                check = true;
                driver = new FirefoxDriver();
            } else if (n == 2) {
                check = true;
                File file = new File("D:/Seminar/Library/chromedriver.exe");
                System.setProperty("WebDriver.chrome.driver", file.getAbsolutePath());
                driver = new ChromeDriver();
            } else if (n == 3) {
                check = true;
                File file = new File("D:/Seminar/Library/IEDriverServer.exe");
                System.setProperty("WebDriver.ie.driver", file.getAbsolutePath());
                driver = new InternetExplorerDriver();
            }
        } catch (NumberFormatException e) {
            System.out.println("Please enter number!");
        }
    } while (!check);
}
```

Bảng 3.14: Kiểm thử khả năng tương thích

d. Lưu lại kết quả đầu ra

❖ Chụp lại kết quả kiểm thử với các test case lỗi[9]

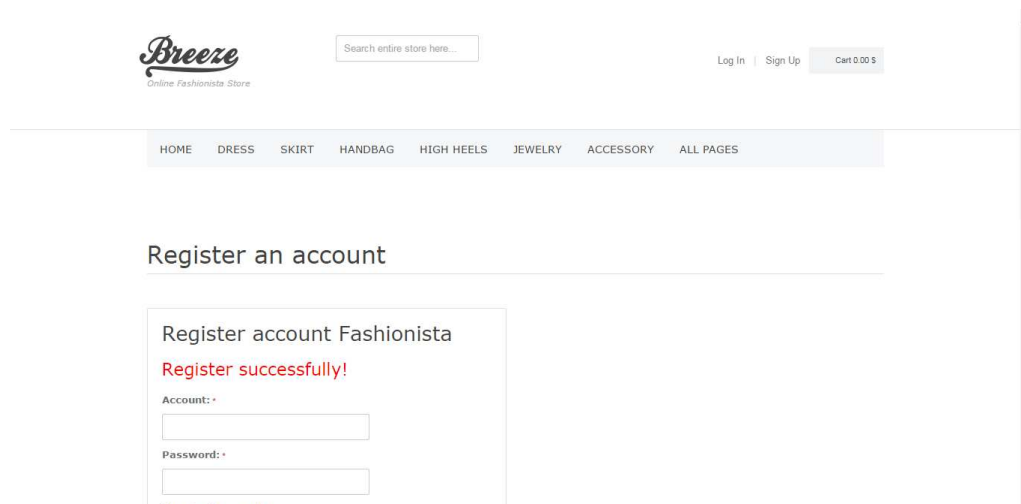
Một điều rất quan trọng trong khi thực thi kiểm thử tự động là chụp lại màn hình kết quả kiểm thử. Khi chúng ta thực thi kiểm thử tự động với một số lượng lớn các test script, và nếu có một vài test case trả về kết quả sai thì chúng ta cần kiểm tra lại các test case lỗi đó. Việc chụp lại màn hình kết quả hiển thị với các test case xảy ra lỗi sẽ giúp ích rất nhiều trong việc phân tích lỗi.

Trong Selenium WebDriver hỗ trợ câu lệnh dùng để chụp lại màn hình:

```
File screenshotFile = ((TakesScreenshot) driver).getScreenshotAs(OutputType.FILE);
```

Câu lệnh này sẽ chụp lại màn hình bất cứ khi nào được gọi đến. Nếu chúng ta chỉ muốn chụp lại các kết quả lỗi thì cần thêm một lệnh kiểm tra trước khi gọi đến hàm chụp màn hình:

```
ITestResult result;
if(result.getStatus() == ITestResult.FAILURE){
    //chụp lại màn hình và lưu kết quả vào thư mục mong muốn..
}
```



Hình 3.6: Kết quả chụp lại màn hình khi đăng ký xảy ra lỗi

❖ Ghi kết quả kiểm thử vào tệp tin Excel[12]

Tương tự như với chức năng đọc dữ liệu đầu vào từ file Excel, do Selenium IDE không hỗ trợ “data driven” nên khi muốn ghi dữ liệu kiểm thử vào một file Excel thì cần phải viết thêm hàm ghi dữ liệu ra file Excel bằng ngôn ngữ Java.

Để ghi dữ liệu từ file Excel, chúng ta cần dùng đến thư viện hỗ trợ “jxl.jar” để tạo tệp tin Excel mới nếu chưa tồn tại, hoặc khi tệp tin Excel đã tồn tại thì sẽ ghi chèn thêm dữ liệu vào cuối tệp tin Excel. Trong thư viện này sẽ hỗ trợ một số câu lệnh để ghi vào tệp tin Excel như:

- Tạo đường dẫn tới thư mục chứa tệp tin đầu ra
- Tạo tệp tin Excel mới và các sheet mới để ghi dữ liệu
- Ghi dữ liệu vào các ô trong file Excel

Dưới đây là kết quả thu được sau khi ghi dữ liệu vào file Excel:

	A	B	C	D	E	F	G
1			Output Edit Carts				
2							
3	ID product	Quantity 1	Expect result add product	Quantity 2	Expect result edit product	Actual result	Test report
4	9	1	Add carts successfully!	2	Update shopping carts successfully!	Update shopping carts successfully!	Pass
5	9	1	Add carts successfully!		Formatting error!	Formatting error!	Pass
6	9	1	Add carts successfully!	-1	Error quantity of SKIRT FASHIONISTA 001	Error quantity of SKIRT FASHIONISTA 001	Pass
7	9	1	Add carts successfully!	a	Formatting error!	Formatting error!	Pass
8	9	1	Add carts successfully!	1000	Error quantity of SKIRT FASHIONISTA 001	Error quantity of SKIRT FASHIONISTA 001	Pass
9	9	1	Add carts successfully!	0	Error quantity of SKIRT FASHIONISTA 001	Error quantity of SKIRT FASHIONISTA 001	Pass
10	9	1	Add carts successfully!	51	Error quantity of SKIRT FASHIONISTA 001	Error quantity of SKIRT FASHIONISTA 001	Pass

Hình 3.7: Kết quả thu được sau khi kiểm thử chức năng Sửa giỏ hàng

Trong đó:

- ID product và Quantity 1 lần lượt là ID và số lượng sản phẩm thêm vào giỏ hàng.
- Expect result add product: Kết quả mong muốn được hiển thị khi thêm sản phẩm vào giỏ hàng.
- Quantity 2 là số lượng sản phẩm muốn sửa.
- Expect result edit product : kết quả mong muốn được hiển thị sau khi sửa thành công.
- Actual result: kết quả thực tế thu được sau khi kiểm thử.
- Test report: kết quả test thành công hoặc thất bại.

3.1.5. Báo cáo và phân tích lỗi

Trong TestNG có hỗ trợ tạo báo cáo HTML sau mỗi lần chạy test script. Để xem báo cáo kết quả kiểm thử, vào workspace tìm đến project vừa thực thi, vào folder “test-output”. Trong folder có 2 file là index.html và emailable-report.html[10].

- emailable-report.html là file report chi tiết, cho phép xem chi tiết các test script, các giá trị đầu vào và hỗ trợ phân tích nguyên nhân xảy ra lỗi cho các test script Failed

Test	# Passed	# Skipped	# Failed	Time (ms)	Included Groups	Excluded Groups
Default suite						
Default test	7	0	0	106.949		

Class	Method	Start	Time (ms)
Default suite			
Default test — passed			
webdriver.TestEditCarts	f	1447081973403	11782
	f		
	f		
	f		
	f		
	f		

Default test

webdriver.TestEditCarts#f

Parameter #1	Parameter #2	Parameter #3	Parameter #4	Parameter #5
9	1	Add carts successfully!	2	Update shopping carts successfully!

[back to summary](#)

Hình 3.8: TestNG report (emailable-report.html)

- index.html là file report dạng tổng quát và có hỗ trợ tìm nguyên nhân xảy ra lỗi cho các script Failed.

Test results

1 suite, 1 failed test

All suites

Default suite

Info

- C:\Users\Phuong Do Thi\AppData\Local\Temp\testng-eclipse-1774495918\testng-customsuite.xml
- 1 test
- 0 groups
- Times
- Reporter output
- Ignored methods
- Chronological view

Results

- 19 methods, 1 failed, 18 passed
- Failed methods (hide)
- ❌ f()

phuongdothi

, phuongphuong, phuongphuong, ?? Th? Ph??ng, Hà N?i - Vi?t Nam, Male, phuongdothi@gmail.com, 1121409834, Error account!)

java.lang.AssertionError: expected [Error account!] but found [Register successfully!]

```

at org.testng.Assert.fail(Assert.java:94)
at org.testng.Assert.failNotEquals(Assert.java:494)
at org.testng.Assert.assertEquals(Assert.java:123)
at org.testng.Assert.assertEquals(Assert.java:176)
at org.testng.Assert.assertEquals(Assert.java:186)
at webdriver.Register.f(Register.java:100)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(Unknown Source)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(Unknown Source)
at java.lang.reflect.Method.invoke(Unknown Source)
at org.testng.internal.MethodInvocationHelper.invokeMethod(MethodInvocationHelper.java:84)
at org.testng.internal.Invoker.invokeMethod(Invoker.java:714)
at org.testng.internal.Invoker.invokeTestMethod(Invoker.java:901)
at org.testng.internal.Invoker.invokeTestMethods(Invoker.java:1231)
at org.testng.internal.TestMethodWorker.invokeTestMethods(TestMethodWorker.java:127)
at org.testng.internal.TestMethodWorker.run(TestMethodWorker.java:111)
at org.testng.TestRunner.privateRun(TestRunner.java:767)
at org.testng.TestRunner.run(TestRunner.java:617)
at org.testng.SuiteRunner.runTest(SuiteRunner.java:334)
at org.testng.SuiteRunner.runSequentially(SuiteRunner.java:329)
at org.testng.SuiteRunner.privateRun(SuiteRunner.java:291)
at org.testng.SuiteRunner.run(SuiteRunner.java:240)
at org.testng.SuiteRunnerWorker.runSuite(SuiteRunnerWorker.java:52)
at org.testng.SuiteRunnerWorker.run(SuiteRunnerWorker.java:86)
at org.testng.TestNG.runSuitesSequentially(TestNG.java:1224)
at org.testng.TestNG.runSuitesLocally(TestNG.java:1149)
at org.testng.TestNG.run(TestNG.java:1057)
at org.testng.remote.RemoteTestNG.run(RemoteTestNG.java:115)
at org.testng.remote.RemoteTestNG.initAndRun(RemoteTestNG.java:207)
at org.testng.remote.RemoteTestNG.main(RemoteTestNG.java:178)

```

Hình 3.9: TestNG report(index.html) hỗ trợ tìm nguyên nhân xảy ra lỗi

3.3. Kết luận chương

Chương 3 đã thiết kế kịch bản kiểm thử cho các chức năng chính của hệ thống Fashionista. Đồng thời, cũng đã đưa ra giải pháp giải quyết bài toán kiểm thử tự động một ứng dụng Web, đưa ra báo cáo kiểm thử và hỗ trợ tìm nguyên nhân xảy ra lỗi với các test case xảy ra lỗi.

Trong Chương 4 sẽ vận dụng các cơ sở lý thuyết về kiểm thử phần mềm và kiểm thử phần mềm tự động, kết hợp với việc sử dụng công cụ kiểm thử Selenium WebDriver và framework TestNG để kiểm thử các chức năng chính của ứng dụng Web Fashionista.

CHƯƠNG 4: THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ

- Giới thiệu hệ thống Fashionista
 - Kiểm thử tự động website với SeleniumWebDriver và TestNG
 - Đánh giá kết quả kiểm thử
 - Kết luận chương
-

Vận dụng các cơ sở lý thuyết về kiểm thử phần mềm và kiểm thử phần mềm tự động, kết hợp với việc sử dụng công cụ kiểm thử Selenium WebDriver và framework TestNG để kiểm thử các chức năng chính của ứng dụng Web Fashionista.

Với kiểm thử tự động, có hai phần chính cần quan tâm là kịch bản kiểm thử và dữ liệu kiểm thử. Trong đó:

- Kịch bản kiểm thử: Có thể là một lớp hoặc một hàm ghi lại một chuỗi sự kiện mà ta thao tác với ứng dụng cần kiểm thử.
- Dữ liệu kiểm thử: Là dữ liệu nhập vào để kiểm thử.

Từ các test case đã thiết kế và mô hình kiểm thử tự động đã xây dựng trong chương 3, phần này của đồ án tiến hành thực thi kiểm thử đối với các chức năng của website Fashionista.

Quy trình thực thi kiểm thử tự động sẽ tiến hành kiểm thử chức năng và kiểm thử khả năng tương thích:

- Phân tích dữ liệu bài toán
- Thiết kế test case
- Thu lại kịch bản kiểm thử bằng Selenium IDE
- Tạo test script trong Eclipse
- Phân tích báo cáo kiểm thử

4. 1. Giới thiệu website Fashionista

- Tên ứng dụng: Fashionista
- Người phát triển: Đỗ Thị Phụng
- Nền tảng: Hệ thống bán hàng online Fashionista
- Ngôn ngữ phát triển: Java
- Cơ sở dữ liệu: Microsoft SQL Server 2008

Hệ thống bán hàng online Fashionista được xây dựng nhằm cung cấp một hệ thống bán hàng trực tuyến, giúp cho khách hàng dù ở xa vẫn có thể mua được sản phẩm mà không cần trực tiếp tới cửa hàng.

Các chức năng chính của hệ thống:

- Login: khách hàng có thể đăng nhập vào hệ thống để thực hiện các chức năng như đặt hàng. Sau khi đặt hàng thành công, khách hàng có thể xem thông tin đặt hàng.
- Register: Người dùng khi chưa có tài khoản đăng nhập hệ thống, có thể thực hiện đăng ký tài khoản mới.
- AddProductToCart: Khách hàng thêm sản phẩm mới vào giỏ hàng. Khách hàng có thể sử dụng chức năng này khi đã đăng nhập vào hệ thống hoặc khi chưa đăng nhập vào hệ thống.
- EditCart: Khi muốn thay đổi số lượng sản phẩm trong giỏ hàng, khách hàng có thể sử dụng chức năng sửa giỏ hàng khi trong giỏ hàng đã có sản phẩm.
- Order: Thành viên thực hiện đặt hàng và nhập các thông tin người nhận để mua hàng.

Hướng dẫn cài đặt hệ thống:

- Hệ điều hành Windows
- Trình duyệt Mozilla Firefox đã cài Add-on Selenium IDE
- Công cụ lập trình Eclipse đã được cấu hình với Selenium WebDriver và đã được cài đặt framework TestNG.
- Hệ quản trị cơ sở dữ liệu quan hệ SQL Server
- Bộ công cụ phát triển ứng dụng bằng ngôn ngữ lập trình Java: JDK (Java development Kit)
- Khác: Microsoft Excel 2003 trở lên.
- Giải nén mã nguồn vào thư mục: D:\Seminar\Projects
- Tạo cơ sở dữ liệu mới: “FashionistaDB”.
- Import file “FashionistaDB.sql” vào cơ sở dữ liệu vừa tạo.
- Ứng dụng được kiểm thử trên 3 trình duyệt: Internet Explorer, Mozilla Firefox, Chrome

Đồ án kiểm thử ứng dụng Fashionista trên các chức năng chính: đăng ký tài khoản mới, thêm sản phẩm vào giỏ hàng, sửa giỏ hàng và đặt hàng.

4. 2. Kiểm thử tự động website với SeleniumWebDriver và TestNG

4.2.1. Chức năng đăng ký tài khoản

❖ Kịch bản kiểm thử chức năng đăng ký tài khoản mới

```
@Test(dataProvider = "dp")
public void f(String account, String password, String reenterPassword,
              String name, String address, String gender, String email,
              String phonenumber, String expectedResult)
    throws InterruptedException {
    selenium.open("/Fashionista/");
    selenium.click("link=Sign Up");
    selenium.waitForPageToLoad("30000");
    selenium.type("name=accountCustomer", account);
    selenium.type("name=passwordCustomer", password);
    selenium.type("name=reenterpassword", reenterPassword);
    selenium.type("name=nameCustomer", name);
    selenium.type("name=addressCustomer", address);
    if ("Female".equals(gender)) {
        selenium.click("xpath=//input[@name='gender'][2]");
    } else if ("Male".equals(gender)) {
        selenium.click("xpath=//input[@name='gender'][1]");
    }
    selenium.type("name=emailCustomer", email);
    selenium.type("name=phonenumber", phonenumber);
    selenium.click("css=input[type='submit']");
    selenium.waitForPageToLoad("30000");
    Assert.assertEquals(selenium.getText("css=h3"), expectedResult);
}
```

Hình 4.1: Kịch bản chức năng đăng ký tài khoản

Trong đó, tất cả các giá trị nhập giá trị nhập vào: Account, Password, Reenter Password, Name, Address, Gender, Email, Phonenumber đều đã được thể hiện dưới dạng tham số để có thể truyền các dữ liệu đầu vào khác nhau mà không cần thay đổi mã code.

❖ File Excel mô tả các trường hợp kiểm thử đầu vào:

	A	B	C	D	E	F	G	H	I
1	Test case register								
2									
3	Account	Password	Reenter password	Name	Address	Gender	Email	Phonenumber	Expect result
4	phuongphuong	phuongdothi	phuongdothi	Đỗ Thị Phương	Hà Nội - Việt Nam	Female	phuongdothi@gmail.com	1670192218	Register successfully!
5		phuongdothi	phuongdothi	Đỗ Thị Phương	Hà Nội - Việt Nam	Female	phuongdothi@gmail.com	1978270913	These field must not be empty!
6	phuongdothi		phuongdothi	Đỗ Thị Phương	Hà Nội - Việt Nam	Female	phuongdothi@gmail.com	1166902178	These field must not be empty!
7	phuongdothi	phuongdothi		Đỗ Thị Phương	Hà Nội - Việt Nam	Male	ivengirl@gmail.com	1912890371	These field must not be empty!
8	phuongdothi	phuongdothi	phuongdothi		Hà Nội - Việt Nam	Male	ivengirl@gmail.com	1789156280	These field must not be empty!
9	Alyssa	alysa	alysa	Alyssa Caesar		Female	alysacaesar@gmail.com	1189801283	These field must not be empty!
10	Shakespeare	shakespeare	shakespeare	William Shakespeare	England			1289123701	These field must not be empty!
11	Coco	cocochanel	cocochanel	Coco Chanel	France	Female	cocochanel@gmail.com		These field must not be empty!
12	va	valeria	valeria	Valeria Dylan	England	Male	valeriaDylan@gmail.com	1266098124	Account have to >=3 and <=20 characters!
13	nguyenhoangthuylinh12a9	thuylinh	thuylinh	Nguyễn Hoàng Thùy	Hà Nội - Việt Nam	Female	thuylinhnguyen@gmail.com	1978102478	Account have to >=3 and <=20 characters!
14	ivengirl	ivengirl	ivengirl	Đỗ Thị Phương	Hải Dương - Việt Nam	Female	ivengirl309@gmail.com	1987170102	Account exists!
15	<h1>phuongdothi</h1>	phuongphuong	phuongphuong	Đỗ Thị Phương	Hà Nội - Việt Nam	Male	phuongdothi@gmail.com	1121409834	Error account!
16	Valeria	va	va	Valeria Dylan	Korea	Male	valeriaDylan@gmail.com	1780123890	Password have to >=6 and <=20 character
17	thuylinh	nguyenhoangthuylinh	nguyenhoangthuylinh	Nguyễn Hoàng Thùy	Hải Dương - Việt Nam	Female	thuylinhnguyen@gmail.com	1780128230	Password have to >=6 and <=20 character
18	phuongdothi	phuongdothi	phuongphuong	Đỗ Thị Phương	Việt Nam	Female	phuongdothi@gmail.com	1912348091	Error reenter password!
19	Shakespeare	shakespeare	shakespeare	William Shakespeare	China	Male	williamShakespeare	1971292031	Error email!
20	Coco	cocochanel	cocochanel	Coco Chanel	France	Female	cocochanel@gmail.com		Phonenumber Error phonenumber!
21	Alyssa	alysa	alysa	Alyssa Caesar	Korea	Female	alysacaesar@gmail.com	89801283	Phonenumber have to 10 or 11 number!
22	Shakespeare	shakespeare	shakespeare	William Shakespeare	England	Male	shakespeare@gmail.com	100001237011	Phonenumber have to 10 or 11 number!

Hình 4.2: Dữ liệu đầu vào cho chức năng đăng ký tài khoản

Trong đó:

- Account, Password, Reenter password, Name, Gender, Email, Phonenumber là các trường thông tin của người dùng.
- Expect result là kết quả mong muốn được hiển thị sau khi đăng ký thành công.

❖ File Excel kết quả kiểm thử:

	D	E	F	G	H	I	J	K
1	Output register							
2								
3	Name	Address	Gender	Email	Phonenumber	Expect result	Actual result	Test report
4	Đỗ Thị Phương	Hà Nội - Việt Nam	Female	phuongdothi@gmail.com	1670192218	Register successfully!	Register successfully!	Pass
5	Đỗ Thị Phương	Hà Nội - Việt Nam	Female	phuongdothi@gmail.com	1978270913	These field must not be empty!	These field must not be empty!	Pass
6	Đỗ Thị Phương	Hà Nội - Việt Nam	Female	phuongdothi@gmail.com	1166902178	These field must not be empty!	These field must not be empty!	Pass
7	Đỗ Thị Phương	Hà Nội - Việt Nam	Male	ivengirl@gmail.com	1912890371	These field must not be empty!	These field must not be empty!	Pass
8		Hà Nội - Việt Nam	Male	ivengirl@gmail.com	1789156280	These field must not be empty!	These field must not be empty!	Pass
9	Alyssa Caesar		Female	alysacaesar@gmail.com	1189801283	These field must not be empty!	These field must not be empty!	Pass
10	William Shakespeare	England	Male		1289123701	These field must not be empty!	These field must not be empty!	Pass
11	Coco Chanel	France	Female	cocochanel@gmail.com		These field must not be empty!	These field must not be empty!	Pass
12	Valeria Dylan	England	Male	valeriaDylan@gmail.com	1266098124	Account have to >=3 and <=20 characters!	Account have to >=3 and <=20 characters!	Pass
13	Nguyễn Hoàng Thùy Linh	Hà Nội - Việt Nam	Female	thuylinhnguyen@gmail.com	1978102478	Account have to >=3 and <=20 characters!	Account have to >=3 and <=20 characters!	Pass
14	Đỗ Thị Phương	Hải Dương - Việt Nam	Female	ivengirl309@gmail.com	1987170102	Account exists!	Account exists!	Pass
15	Đỗ Thị Phương	Hà Nội - Việt Nam	Male	phuongdothi@gmail.com	1121409834	Error account!	Register successfully!	False
16	Valeria Dylan	Korea	Male	valeriaDylan@gmail.com	1780123890	Password have to >=6 and <=20 characters!	Password have to >=6 and <=20 characters!	Pass
17	Nguyễn Hoàng Thùy Linh	Hải Dương - Việt Nam	Female	thuylinhnguyen@gmail.com	1780128230	Password have to >=6 and <=20 characters!	Password have to >=6 and <=20 characters!	Pass
18	Đỗ Thị Phương	Việt Nam	Female	phuongdothi@gmail.com	1912348091	Error reenter password!	Error reenter password!	Pass
19	William Shakespeare	China	Male	williamShakespeare	1971292031	Error email!	Error email!	Pass
20	Coco Chanel	France	Female	cocochanel@gmail.com		Phonenumber Error phonenumber!	Error phonenumber!	Pass
21	Alyssa Caesar	Korea	Female	alysacaesar@gmail.com	89801283	Phonenumber have to 10 or 11 number!	Phonenumber have to 10 or 11 number!	Pass
22	William Shakespeare	England	Male	shakespeare@gmail.com	100001237011	Phonenumber have to 10 or 11 number!	Phonenumber have to 10 or 11 number!	Pass

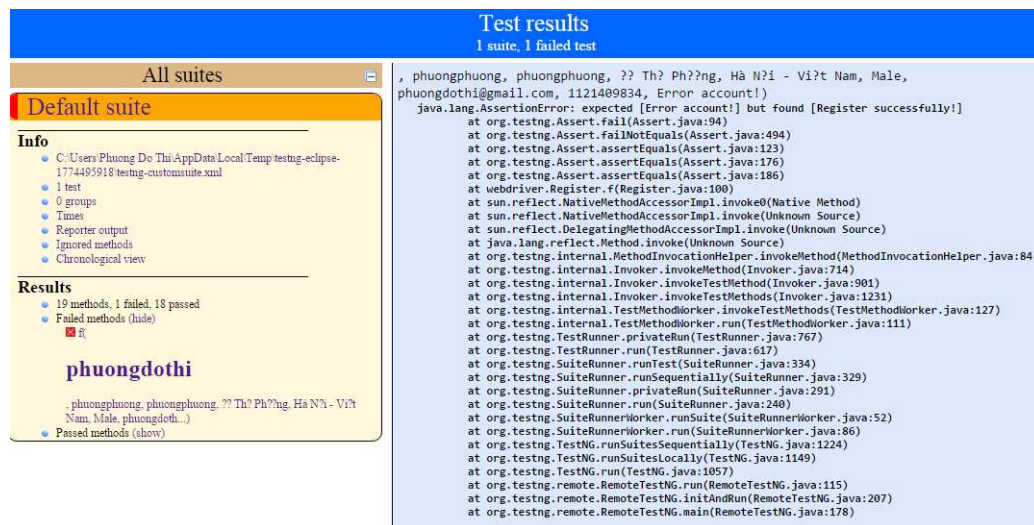
Hình 4.3: Kết quả đầu ra của chức năng đăng ký tài khoản

Trong đó:

- Account, Password, Reenter password, Name, Gender, Email, Phonenumber là các trường thông tin của người dùng.
- Expect result là kết quả mong muốn được hiển thị sau khi đăng ký thành công.
- Actual result là kết quả thực tế được hiển thị
- Test report là kết luận test case thành công hay thất bại

❖ Báo cáo kết quả chạy test script của chức năng đăng ký tài khoản

TestNG đưa ra báo cáo kết quả kiểm thử dưới dạng HTML và hỗ trợ tìm nguyên nhân xảy ra lỗi.



Hình 4.4: Báo cáo kết quả kiểm thử chức năng đăng ký tài khoản

❖ **Kết quả kiểm thử chức năng đăng ký tài khoản**

Số lượng test case	20
Số test case Pass	18
Số test case Failure	2
Số test case Error	0
Số test case chưa thực hiện	0
Tỉ lệ thành công	90%

Bảng 4.1: Kết quả test chức năng đăng ký tài khoản mới

❖ **Nhận xét:**

- Tính khả dụng
 - Giao diện đơn giản, dễ sử dụng, chức năng của phím tab, enter hoạt động tốt. Không có lỗi chính tả, không khó để đọc chữ, hình ảnh được sắp xếp gọn gàng, hợp lý.
 - Giao diện có đầy đủ có menu chính và menu này nhất quán với giao diện của các chức năng khác trong hệ thống.
- Khả năng tương thích
 - Hệ thống tương thích với cả ba trình duyệt được kiểm thử là Internet Explorer, Mozilla Firefox, Chrome.
- Về chức năng

Trong quá trình thực thi kiểm thử tự động chức năng đăng ký tài khoản mới, đã xảy ra lỗi ở trường hợp:

- Khi người dùng đăng ký tài khoản mới với tên tài khoản là một thẻ HTML “<h1>phuongdothi</h1>”, kết quả mong muốn hiển thị thông báo lỗi “Error account!” và đăng ký không thành công. Tuy nhiên, kết quả thực tế hiển thị thông báo đăng ký thành công với message “Register successfully!”.

4.2.2. Chức năng thêm sản phẩm vào giỏ hàng

❖ Kết quả kiểm thử chức năng thêm sản phẩm vào giỏ hàng

Số lượng test case	16
Số test case Pass	16
Số test case Failure	0
Số test case Error	0
Số test case chưa thực hiện	0
Tỉ lệ thành công	100%

Bảng 4.2: Kết quả test chức năng thêm sản phẩm vào giỏ hàng

❖ Nhận xét:

- Tính khả dụng
 - Giao diện đơn giản, dễ sử dụng, chức năng của phím tab, enter hoạt động tốt. Không có lỗi chính tả, không khó để đọc chữ, hình ảnh được sắp xếp gọn gàng, hợp lý.
 - Giao diện có đầy đủ có menu chính và menu này nhất quán với giao diện của các chức năng khác trong hệ thống.
- Khả năng tương thích
 - Hệ thống tương thích với cả ba trình duyệt được kiểm thử là Internet Explorer, Mozilla Firefox, Chrome
- Về chức năng
 - Thực hiện đúng và đầy đủ tất cả các trường hợp đã được mô tả trong bản thiết kế test case.

4.2.3. Chức năng sửa giỏ hàng

❖ Kết quả kiểm thử chức năng sửa giỏ hàng

Số lượng test case	8
Số test case Pass	8
Số test case Failure	0
Số test case Error	0
Số test case chưa thực hiện	0

Tỉ lệ thành công	100%
------------------	------

Bảng 4.3: Kết quả test chức năng sửa giỏ hàng**❖ Nhận xét:**

- Tính khả dụng

Giao diện đơn giản, dễ sử dụng, chức năng của phím tab, enter hoạt động tốt. Không có lỗi chính tả, không khó để đọc chữ, hình ảnh được sắp xếp gọn gàng, hợp lý.

Giao diện có đầy đủ có menu chính và menu này nhất quán với giao diện của các chức năng khác trong hệ thống.

Khi phóng to, thu nhỏ không bị vỡ giao diện

- Khả năng tương thích

Hệ thống tương thích với cả ba trình duyệt được kiểm thử là Internet Explorer, Mozilla Firefox, Chrome

- Về chức năng

Thực hiện đúng và đầy đủ tất cả các trường hợp đã được mô tả trong bản thiết kế test case.

4.2.4. Chức năng đặt hàng**❖ Kết quả kiểm thử chức năng đặt hàng**

Số lượng test case	12
Số test case Pass	12
Số test case Failure	0
Số test case Error	0
Số test case chưa thực hiện	0
Tỉ lệ thành công	100%

Bảng 4.4: Kết quả test chức năng đặt hàng**❖ Nhận xét:**

- Tính khả dụng

- Giao diện đơn giản, dễ sử dụng, chức năng của phím tab, enter hoạt động tốt. Không có lỗi chính tả, không khó để đọc chữ, hình ảnh được sắp xếp gọn gàng, hợp lý.
- Giao diện có đầy đủ có menu chính và menu này nhất quán với giao diện của các chức năng khác trong hệ thống.
- Khi phóng to, thu nhỏ không bị vỡ giao diện

- Khả năng tương thích
 - Hệ thống tương thích với cả ba trình duyệt được kiểm thử là Internet Explorer, Mozilla Firefox, Chrome.
- Về chức năng
 - Thực hiện đúng và đầy đủ tất cả các trường hợp đã được mô tả trong bản thiết kế test case.
- Tính bảo mật
 - Gõ trực tiếp url vào thanh địa chỉ của trình duyệt mà không qua thực hiện các yêu cầu bắt buộc trước khi truy cập vào trang đó.
 - Thay đổi url trực tiếp bằng cách đổi tham số kiểm tra tính hợp lệ của trang đó để chuyển tới một trang khác.
 - Thử các giá trị đầu vào không hợp lệ vào các trường của trang web.

4.3. Đánh giá kết quả kiểm thử

4.3.1. Kiểm thử chức năng

Đánh giá toàn bộ quá trình kiểm thử trong tổng số 56 test case của 4 chức năng chính của hệ thống:

Chức năng	Tổng số test	Số test tự động	Số test pass	Số test failure	Tỉ lệ thành công
Đăng ký tài khoản	20	20	18	2	90%
Thêm sản phẩm vào giỏ hàng	16	15	16	0	100%
Sửa giỏ hàng	8	7	8	0	100%
Đặt hàng	12	12	12	0	100%

Bảng 4.5: Kết quả kiểm thử chức năng

Trong đó:

- Các chức năng là Thêm sản phẩm vào giỏ hàng, sửa giỏ hàng và đặt hàng đã đảm bảo thực thi đúng và đầy đủ các test case. Tuy nhiên, do Selenium không hỗ trợ giả lập nhiều người dùng ảo cùng lúc, nên với chức năng Thêm sản phẩm vào giỏ hàng và chức năng sửa giỏ hàng đã có 2 test case không thể thực thi kiểm thử tự động với Selenium Webdriver và TestNG.
- Với chức năng đăng ký tài khoản mới thì có một test case đã xảy ra lỗi:
 Khi người dùng đăng ký tài khoản mới với tên tài khoản là một thẻ HTML “<h1>phuongdothi</h1>”, kết quả mong muốn hiển thị thông báo lỗi “Error account!” và đăng ký không thành công. Tuy nhiên, kết quả thực tế hiển thị thông báo đăng ký thành công với message “Register successfully!”.

Vì vậy cần bổ sung trường hợp kiểm tra tính hợp lệ của tên tài khoản khi đăng ký mới.

- Với các tính năng như: tính khả dụng, tính bảo mật và khả năng tương thích thì hệ thống đều đã đáp ứng tốt.
- Các báo cáo kiểm thử, gợi ý tìm và sửa lỗi đã được ghi lại đầy đủ trong báo cáo hỗ trợ bởi framework TestNG.

4.3.2. Kiểm thử phi chức năng

Đánh giá toàn bộ quá trình kiểm thử phi chức năng trong tổng số 86 test case của 4 chức năng chính của hệ thống:

Chức năng	Tổng số test	Số test pass	Số test failure	Tỉ lệ thành công
Đăng ký tài khoản	21	17	4	80.95%
Thêm sản phẩm vào giỏ hàng	21	18	3	85.71%
Sửa giỏ hàng	23	19	4	82.61%
Đặt hàng	21	19	3	90.48%

Bảng 4.6: Kết quả kiểm thử phi chức năng

Trong đó:

- Khi kiểm thử phi chức năng, hầu hết các chức năng của hệ thống đều bị lỗi timeout do website không được đặt thời gian timeout.
- Với chức năng đăng ký và chức năng đặt hàng, ngoài lỗi time out thì có 2 test case khác xảy ra lỗi khi kiểm thử là:
 - Lỗi khi mất kết nối mạng: khi kết nối mạng bị mất, hệ thống vẫn thực hiện đăng ký và lưu thông tin tài khoản vào cơ sở dữ liệu.
 - Lỗi khi mất kết nối database: khi kết nối database bị đóng, hệ thống vẫn thực hiện đăng ký và lưu thông tin tài khoản vào cơ sở dữ liệu.

4. 4. Kết luận chương

Chương 4 đã vận dụng các cơ sở lý thuyết về kiểm thử phần mềm và kiểm thử phần mềm tự động, kết hợp với việc sử dụng công cụ kiểm thử Selenium WebDriver và framework TestNG để kiểm thử các chức năng chính của ứng dụng Web Fashionista trên các trình duyệt Internet Explorer, Mozilla Firefox, Chrome. Đồng thời cũng đã đưa ra báo cáo kiểm thử và phân tích nguyên nhân gây ra lỗi.

KẾT LUẬN

Kiểm thử phần mềm hiện nay giữ vai trò hết sức quan trọng trong việc đảm bảo chất lượng của một sản phẩm phần mềm. Mặc dù việc sử dụng các công cụ kiểm thử tự động đã giúp giảm thời gian, kinh phí, nhân lực trong quá trình phát triển phần mềm. Nhưng cũng có những việc mà công cụ kiểm thử tự động không thể thay thế được kiểm thử viên, hoặc nếu có thì việc cấu hình cho nó khó khăn hơn rất nhiều so với kiểm thử thủ công. Do đó, kiểm thử viên giỏi phải là người nhận biết được khi nào nên sử dụng công cụ kiểm thử tự động và khi nào nên kiểm thử thủ công.

Sau thời gian thực hiện đồ án dưới sự hướng dẫn của TS. Đỗ Thị Bích Ngọc, kết quả mà em thu được cụ thể như sau:

Kết quả đạt được:

- Trình bày được các kiến thức cơ bản về kiểm thử phần mềm nói chung và kiểm thử phần mềm tự động cho các ứng dụng Web nói riêng
- Xây dựng ứng dụng Web bán hàng online Fashionista với công nghệ JSP và Servlet
- Giới thiệu được các đặc điểm, thành phần của công cụ kiểm thử tự động Selenium. Kết hợp công cụ Selenium WebDriver và framework TestNG trong kiểm thử tự động ứng dụng Web.
- Áp dụng các kiến thức đã tìm hiểu vào thực hiện kiểm thử tự động các chức năng chính của ứng dụng Web bán hàng online Fashionista.

Hạn chế:

- Chưa thể giả lập nhiều người dùng ảo chạy đồng thời để kiểm thử khả năng xử lý xung đột khi có nhiều khách hàng cùng đặt hàng vào một thời điểm.
- Chưa nghiên cứu được phương pháp kiểm thử hướng từ khóa với Selenium.
- Mới kiểm thử với ứng dụng nhỏ, chưa kiểm thử với ứng dụng thực tế.

Hướng phát triển đề tài:

Đồ án có thể tiếp tục tìm hiểu sâu hơn về các vấn đề của kiểm thử phần mềm và đặc biệt là công cụ kiểm thử tự động Selenium để có thể vận dụng kiểm thử các ứng dụng lớn hơn trong thực tiễn.

TÀI LIỆU THAM KHẢO

Tiếng Việt

- [1]. Phạm Ngọc Hùng, Trương Anh Hoàng, và Đặng Văn Hưng. *Giáo trình kiểm thử phần mềm* - tháng 1 năm 2014.
- [2]. Phan Thi Hoài Phương. *Đảm bảo chất lượng phần mềm* – 2010. Tr. 94-102

Tiếng Anh

- [3]. David Burns. *Selenium 1.0 Testing Tools: Beginner's Guide Paperback* – November 11, 2010. published by *Packt Publishing Ltd*
- [4]. David Burns. *Selenium 2 Testing Tools: Beginner's Guide Paperback* – October 19, 2012. published by *Packt Publishing Ltd*
- [5]. Hans-Erik Eriksson, Magnus Penker, Brian Lyons, and David Fado, *UML 2 Toolkit*, 2003, published by *Wiley Publishing*, pages 24-45 and 337-460
- [6]. Daniel Galin, “*Software Quality Assurance From Theory to Implementation*”, 2004, published by *Israel's leading publishers*, pages 205 – 235
- [7]. Mike O'Docherty, *Object Oriented Analysis & Design: Understanding System Development with UML 2.0*, pages 123-147, 430-452, 2005
- [8]. Alan John Richardson, *Selenium Simplified* – February 1, 2012, published by *Compendium Developments*, published by *Packt Publishing Ltd*, pages 283-287

Danh mục các Website tham khảo

- [9]. Taking Screenshot using Webdriver. <http://seleniumeasy.com/>. Ngày truy cập: 2 tháng 11 năm 2015
- [10]. Selenium and TestNG. <http://testng.org/> . Ngày truy cập: 14 tháng 9 năm 2015
- [11]. WebDriver: Advanced Usage. <http://www.seleniumhq.org/>. Ngày truy cập: 14 tháng 9 năm 2015
- [12]. TestNG Data Provider with Excel. <http://www.toolsqa.com/selenium-webdriver/> Ngày truy cập: 18 tháng 10 năm 2015