**TRƯỜNG ĐẠI HỌC KIẾN TRÚC HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

-----🙞🙜🕮🙞🙜-----

**ĐỒ ÁN KIỂM THỬ VÀ ĐẢM BẢO CHẤT LƯỢNG PHẦN MỀM**

**NGÀNH: CÔNG NGHỆ THÔNG TIN**

**MÃ NGÀNH: 7480201**

**KHÓA: 2021 – 2026**

***ĐỀ TÀI:***

**TÌM HIỂU MỘT SỐ KỸ THUẬT KIỂM THỬ TỰ ĐỘNG VÀ ỨNG DỤNG KIỂM THỬ CHO**

**WEBSITE BÁN HÀNG**

|  |  |
| --- | --- |
| **GIÁO VIÊN HƯỚNG DẪN:** | **ThS. NGUYỄN XUÂN THU** |
| **NHÓM SINH VIÊN THỰC HIỆN:** | **NGUYỄN HỮU THÀNH ĐẠT** |
|  | **TRẦN THANH HIỀN** |
|  | **TRẦN QUANG HƯNG** |

**LỚP: 21CN2**

**HÀ NỘI, 3-2025**

**TRƯỜNG ĐẠI HỌC KIẾN TRÚC HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

-----🙞🙜🕮🙞🙜-----

**ĐỒ ÁN KIỂM THỬ VÀ ĐẢM BẢO CHẤT LƯỢNG PHẦN MỀM**

**NGÀNH: CÔNG NGHỆ THÔNG TIN**

**MÃ NGÀNH: 7480201**

**KHÓA: 2021 – 2026**

***ĐỀ TÀI:***

**TÌM HIỂU MỘT SỐ KỸ THUẬT KIỂM THỬ TỰ ĐỘNG VÀ ỨNG DỤNG KIỂM THỬ CHO**

**WEBSITE BÁN HÀNG**

|  |  |
| --- | --- |
| **GIÁO VIÊN HƯỚNG DẪN:** | **ThS. NGUYỄN XUÂN THU** |
| **NHÓM SINH VIÊN THỰC HIỆN:** | **NGUYỄN HỮU THÀNH ĐẠT** |
|  | **TRẦN THANH HIỀN** |
|  | **TRẦN QUANG HƯNG** |

**LỚP: 21CN2**

**HÀ NỘI, 3-2025**

# LỜI CẢM ƠN

Trong thời đại số hóa ngày nay, việc kiểm thử chức năng trên các ứng dụng web đóng vai trò quan trọng trong việc đảm bảo chất lượng và tính ổn định của các sản phẩm công nghệ. Trong lĩnh vực thương mại điện tử, việc kiểm thử chức năng trên các trang web bán hàng là một phần không thể thiếu để đảm bảo trải nghiệm mua sắm trực tuyến mượt mà và hiệu quả cho người dùng.

Trong tiểu luận này, nhóm chúng em sẽ tập trung vào việc thực hiện kiểm thử chức năng trên một website bán máy tính bằng công cụ Selenium. Selenium là một công cụ mạnh mẽ, được sử dụng rộng rãi trong ngành công nghiệp phát triển phần mềm để tự động hóa quy trình kiểm thử trên các ứng dụng web. Bằng cách sử dụng Selenium, chúng ta có thể tạo ra các kịch bản kiểm thử tự động nhằm kiểm tra tính năng, tương tác và hiệu suất của các trang web.

Website mà chúng em sẽ thực hiện kiểm thử là một trang web bán máy tính, một trong những thị trường sôi động và cạnh tranh cao trong lĩnh vực thương mại điện tử hiện nay. Việc kiểm thử chức năng trên trang web này sẽ bao gồm việc kiểm tra đăng ký người dùng, đăng nhập, tính năng mua hàng, thanh toán, quản lý giỏ hàng và các tính năng khác mà người dùng thường xuyên gặp phải khi mua sắm trực tuyến.

Nhóm chúng em xin gửi lời cảm ơn chân thành và sâu sắc nhất đến ThS. Nguyễn Xuân Thu, giảng viên của Đại học Kỹ thuật - Hậu cần CAND, người đã tận tình hướng dẫn, dành rất nhiều thời gian, công sức để truyền đạt kiến thức, định hướng cũng như giải đáp rất nhiều thắc mắc cho nhóm chúng em, để nhóm có thể hoàn thành đồ án môn học một cách hoàn chỉnh và đúng thời hạn.

# LỜI CAM ĐOAN

Nhóm em xin cam đoan rằng đồ án môn học với đề tài **“Tìm hiểu một số kỹ thuật Kiểm thử tự động và Ứng dụng kiểm thử cho Website bán hàng”** là công trình nghiên cứu của riêng nhóm chúng em. Các thông tin, số liệu và kết quả trong báo cáo đều được nhóm thu thập và xử lý một cách trung thực, khách quan và không có sự sao chép, đạo văn từ bất kỳ nguồn nào khác.

Các số liệu trong báo cáo đều được trích dẫn từ các nguồn có uy tín và được tuân thủ đúng nguyên tắc. Kết quả thực nghiệm trình bày trong báo cáo thu thập được trong quá trình nghiên cứu là trung thực và chưa từng được công bố trước đây.

Nhóm em xin chịu trách nhiệm về những thông tin và kết quả trong báo cáo này. Nếu có bất kỳ sai sót nào, nhóm em xin chịu mọi hình thức kỷ luật theo quy định của nhà trường.

# LỜI NÓI ĐẦU

## Lý do chọn đề tài

Theo báo cáo của Statista, tính đến năm 2023, có hơn 5.18 tỷ người dùng Internet trên toàn cầu, chiếm khoảng 64.6% dân số thế giới. Các ứng dụng website đóng vai trò quan trọng trong mọi lĩnh vực, từ thương mại điện tử, giáo dục, tài chính đến giải trí và mạng xã hội. Sự phát triển nhanh chóng của công nghệ web kéo theo nhu cầu kiểm thử ứng dụng ngày càng cao nhằm đảm bảo chất lượng và trải nghiệm người dùng.

Kiểm thử phần mềm là một bước quan trọng trong vòng đời phát triển ứng dụng, giúp phát hiện và khắc phục lỗi trước khi sản phẩm đến tay người dùng. Tuy nhiên, kiểm thử thủ công thường tốn nhiều thời gian, công sức và dễ xảy ra sai sót do yếu tố con người. Do đó, kiểm thử tự động trở thành một giải pháp tối ưu, giúp tăng tốc quá trình kiểm thử, giảm thiểu sai sót và đảm bảo ứng dụng hoạt động ổn định hơn.

Một trong những thách thức lớn trong kiểm thử tự động là việc lựa chọn và áp dụng các công cụ kiểm thử phù hợp để đảm bảo độ chính xác và tính toàn diện của quá trình kiểm tra. Các công cụ kiểm thử tự động cần có khả năng mô phỏng hành vi người dùng, kiểm tra các tính năng quan trọng và đảm bảo hiệu suất hoạt động của ứng dụng. Việc thiết lập và tối ưu hóa kịch bản kiểm thử cũng đóng vai trò quan trọng trong việc nâng cao hiệu quả của quá trình kiểm thử.

Xuất phát từ thực tế này, nhóm em lựa chọn đề tài: "**Tìm hiểu một số kỹ thuật Kiểm thử tự động và Ứng dụng kiểm thử cho Website bán hàng**" nhằm tìm hiểu các phương pháp kiểm thử tự động, đánh giá hiệu quả của các công cụ hỗ trợ như Selenium, Cypress, Playwright,... và đề xuất giải pháp tối ưu giúp nâng cao chất lượng kiểm thử phần mềm. Những kết quả nghiên cứu này có thể được ứng dụng để cải thiện hiệu suất kiểm thử, tiết kiệm thời gian và chi phí cho các tổ chức, cá nhân trong lĩnh vực phát triển phần mềm.

## Tình hình nghiên cứu trong nước và trên thế giới

***Tình hình nghiên cứu trên thế giới***

Trong lĩnh vực kiểm thử tự động cho ứng dụng website, nhiều tổ chức và công ty phần mềm lớn đã nghiên cứu và phát triển các phương pháp, công cụ nhằm hỗ trợ quá trình kiểm thử phần mềm một cách toàn diện và hiệu quả hơn. Một số công cụ và kỹ thuật phổ biến có thể kể đến:

* **Selenium [1]:** Đây là một trong những công cụ kiểm thử tự động phổ biến nhất, hỗ trợ kiểm thử trên nhiều trình duyệt và hệ điều hành. Selenium cho phép tự động hóa các thao tác trên giao diện web, giúp phát hiện lỗi trong quá trình phát triển phần mềm.
* **Cypress [2]:** Công cụ kiểm thử hiện đại, tập trung vào kiểm thử giao diện người dùng với tốc độ nhanh và tích hợp tốt với các framework JavaScript. Cypress hỗ trợ kiểm thử theo thời gian thực và dễ dàng ghi lại kết quả kiểm thử.
* **Playwright [3]:** Một công cụ kiểm thử tự động mạnh mẽ do Microsoft phát triển, hỗ trợ đa trình duyệt và cung cấp nhiều tính năng tiên tiến như kiểm thử song song, chụp ảnh màn hình, ghi lại video quá trình kiểm thử.
* **JMeter [4]:** Một công cụ kiểm thử hiệu năng phổ biến, giúp đánh giá khả năng chịu tải của hệ thống web, đo lường tốc độ phản hồi và kiểm tra khả năng mở rộng của ứng dụng.
* **Appium [5]:** Công cụ kiểm thử tự động dành cho ứng dụng di động, hỗ trợ cả Android và iOS, cho phép kiểm thử trên thiết bị thực và trình giả lập.

Ngoài ra, nhiều nền tảng CI/CD như Jenkins, GitHub Actions, GitLab CI/CD cũng tích hợp các công cụ kiểm thử tự động nhằm hỗ trợ kiểm tra phần mềm liên tục, giúp phát hiện lỗi sớm trong quá trình phát triển.

***Tình hình nghiên cứu trong nước***

Tại Việt Nam, các doanh nghiệp phần mềm chủ yếu áp dụng các công cụ kiểm thử tự động từ nước ngoài vào quy trình phát triển phần mềm. Hiện nay, chưa có nhiều nghiên cứu chuyên sâu về việc phát triển công cụ kiểm thử tự động nội địa, hầu hết các nhóm phát triển phần mềm vẫn dựa vào các công cụ phổ biến như Selenium, Cypress hay JMeter.

Tuy nhiên, với sự phát triển mạnh mẽ của ngành công nghệ thông tin, nhu cầu về kiểm thử tự động ngày càng trở nên quan trọng. Các công ty phần mềm lớn tại Việt Nam đang bắt đầu ứng dụng các nền tảng kiểm thử liên tục (CI/CD) để nâng cao hiệu suất và độ tin cậy của phần mềm. Việc nghiên cứu và phát triển các công cụ kiểm thử tự động nội địa có thể giúp giảm bớt sự phụ thuộc vào công nghệ nước ngoài, tối ưu hóa quy trình kiểm thử phù hợp với đặc thù của các doanh nghiệp Việt Nam.

Kiểm thử tự động không chỉ giúp tiết kiệm thời gian, chi phí mà còn nâng cao chất lượng phần mềm, đảm bảo ứng dụng web hoạt động ổn định trong nhiều môi trường khác nhau. Do đó, đây là một lĩnh vực có tiềm năng phát triển lớn và cần được đầu tư nghiên cứu sâu hơn trong tương lai.

## Mục đích nghiên cứu của đề tài

Nghiên cứu, tìm hiểu các kỹ thuật kiểm thử tự động cho website; khảo sát các công cụ kiểm thử mã nguồn mở đã được triển khai sử dụng trên thế giới và trong nước, từ đó lựa chọn ra các công cụ phù hợp xây dựng thành kịch bản thử nghiệm để phân tích, đánh giá hiệu quả của chúng.

## Nhiệm vụ nghiên cứu

Để đạt được mục đích nói trên, đề tài tập trung thực hiện những nhiệm vụ chính như sau:

* Nghiên cứu, tìm hiểu tổng quan về công nghệ Web
* Nghiên cứu, tìm hiểu về các phương pháp kiểm thử tự động
* Nghiên cứu, khảo sát, tìm hiểu một số công cụ kiểm thử tự động mã nguồn mở cho ứng dụng web
* Xây dựng kịch bản thử nghiệm để phân tích, đánh giá hiệu quả của công cụ kiểm thử tự động

## Đối tượng nghiên cứu

* Website bán hàng nói riêng và đa số các website dịch vụ khác nói chung
* Công cụ mã nguồn mở sinh dữ liệu kiểm thử tự động cho website

## Phạm vi nghiên cứu

Các phương pháp kiểm thử tự động cho Website bán hàng

## Phương pháp nghiên cứu

* Phương pháp nghiên cứu tài liệu: Thu thập, nghiên cứu, phân tích tổng hợp các tài liệu, báo cáo khoa học về kỹ thuật và công cụ kiểm thử tự động.
* Phương pháp tham khảo ý kiến chuyên gia: Tham khảo ý kiến của giáo viên hướng dẫn, các thầy cô giáo có nhiều kinh nghiệm trong lĩnh vực kiểm thử phần mềm.
* Phương pháp thực nghiệm: Lựa chọn các công cụ mã nguồn mở kiểm thử tự động cho website, cài đặt, xây dựng kịch bản thử nghiệm để phân tích, đánh giá hiệu quả của công cụ.

## Dự kiến kết quả đạt được

Kết quả của đồ án dự kiến bao gồm các sản phẩm chính như sau:

* Nghiên cứu về các kỹ thuật kiểm thử tự động cho ứng dụng Web
* Quy trình kiểm thử tự động ứng dụng Web dựa trên các công cụ mã nguồn mở có sẵn
* Kịch bản thử nghiệm kiểm thử tự động ứng dụng Web dựa trên các công cụ mã nguồn mở và các kết quả phân tích, đánh giá hiệu quả của công cụ

## Bố cục

Đồ án được bố cục thành 4 chương với nội dung như sau:

Chương 1. Cơ sở lý thuyết

Chương 2. Các kỹ thuật và công cụ kiểm thử tự động

Chương 3. Các yêu cầu và tiêu chuẩn kiểm thử trên web

Chương 4. Ứng dụng kiểm thử tự động cho website bán hàng

# MỤC LỤC

[LỜI CẢM ƠN 1](#_Toc192098313)

[LỜI CAM ĐOAN 2](#_Toc192098314)

[LỜI NÓI ĐẦU 3](#_Toc192098315)

[1. Lý do chọn đề tài 3](#_Toc192098316)

[2. Tình hình nghiên cứu trong nước và trên thế giới 3](#_Toc192098317)

[3. Mục đích nghiên cứu của đề tài 4](#_Toc192098318)

[4. Nhiệm vụ nghiên cứu 5](#_Toc192098319)

[5. Đối tượng nghiên cứu 5](#_Toc192098320)

[6. Phạm vi nghiên cứu 5](#_Toc192098321)

[7. Phương pháp nghiên cứu 5](#_Toc192098322)

[8. Dự kiến kết quả đạt được 5](#_Toc192098323)

[9. Bố cục 6](#_Toc192098324)

[MỤC LỤC 7](#_Toc192098325)

[DANH MỤC CÁC BẢNG 10](#_Toc192098326)

[CHƯƠNG 1. CƠ SỞ LÝ THUYẾT 11](#_Toc192098327)

[1.1. Kiểm thử phần mềm và các khái niệm liên quan 11](#_Toc192098328)

[1.1.1. Khái niệm kiểm thử phần mềm 11](#_Toc192098329)

[1.1.2. Vai trò của kiểm thử phần mềm 11](#_Toc192098330)

[1.1.3. Phân loại kiểm thử phần mềm 11](#_Toc192098331)

[1.2. Kiểm thử tự động 13](#_Toc192098332)

[1.2.1. Khái niệm kiểm thử tự động 13](#_Toc192098333)

[1.2.2. Quy trình kiểm thử tự động 14](#_Toc192098334)

[1.2.3. Ưu, nhược điểm của kiểm thử tự động 14](#_Toc192098335)

[1.2.4. Các nguyên tắc trong kiểm thử tự động 15](#_Toc192098336)

[1.2.5. Các phương pháp kiểm thử tự động 16](#_Toc192098337)

[1.2.6. Các công cụ hỗ trợ kiểm thử tự động 19](#_Toc192098338)

[CHƯƠNG 2. CÁC KỸ THUẬT VÀ CÔNG CỤ KIỂM THỬ TỰ ĐỘNG 21](#_Toc192098339)

[2.1. Công cụ kiểm thử Selenium – Selenium Automated Testing 21](#_Toc192098340)

[2.1.1. Tổng quan về Selenium 21](#_Toc192098341)

[2.1.2. Selenium Webdriver 22](#_Toc192098342)

[2.2. Robot Framework 23](#_Toc192098343)

[2.2.1. Các tính năng của Robot Framework 23](#_Toc192098344)

[2.2.2. Cài đặt Robot Framework 24](#_Toc192098345)

[2.2.3. Các câu lệnh trong Robot Framework 24](#_Toc192098346)

[CHƯƠNG 3. CÁC YÊU CẦU VÀ TIÊU CHUẨN KIỂM THỬ TRÊN WEB 26](#_Toc192098347)

[3.1. Khái quát về kiểm thử ứng dụng trên Web 26](#_Toc192098348)

[3.1.1. Khái quát 26](#_Toc192098349)

[3.1.2. Các loại ứng dụng Web 26](#_Toc192098350)

[3.2. Công việc chính khi kiểm thử ứng dụng Web 27](#_Toc192098351)

[3.2.1. Kiểm thử chức năng 27](#_Toc192098352)

[3.2.2. Kiểm thử khả năng 28](#_Toc192098353)

[3.2.3. Kiểm thử sự tương thích 28](#_Toc192098354)

[3.2.4. Kiểm thử hiệu suất 29](#_Toc192098355)

[3.2.5. Kiểm thử bảo mật 29](#_Toc192098356)

[3.3. Một số công cụ hỗ trợ 30](#_Toc192098357)

[3.3.1. Công cụ kiểm thử hiệu năng 30](#_Toc192098358)

[3.3.2. Công cụ kiểm thử bảo mật 31](#_Toc192098359)

[3.3.3. Công cụ kiểm thử chức năng 31](#_Toc192098360)

[CHƯƠNG 4. ỨNG DỤNG KIỂM THỬ TỰ ĐỘNG CHO WEBSITE BÁN HÀNG 33](#_Toc192098361)

[4.1. Mô tả hệ thống thử nghiệm 33](#_Toc192098362)

[4.1.1. Tổng quan về hệ thống thử nghiệm 33](#_Toc192098363)

[4.1.2. Công nghệ sử dụng trong hệ thống 33](#_Toc192098364)

[4.1.3. Môi trường thử nghiệm 33](#_Toc192098365)

[4.1.4. Mục tiêu kiểm thử 34](#_Toc192098366)

[4.2. Xây dựng kịch bản kiểm thử cho từng trường hợp 35](#_Toc192098367)

[4.2.1. Kiểm thử chức năng đăng ký tài khoản 35](#_Toc192098368)

[4.2.2. Kiểm thử chức năng đăng nhập 36](#_Toc192098369)

[4.2.3. Kiểm thử chức năng đăng xuất 39](#_Toc192098370)

[4.2.4. Kiểm thử chức năng thêm sản phẩm vào giỏ hàng 39](#_Toc192098371)

[4.2.5. Kiểm thử chức năng thanh toán 40](#_Toc192098372)

[4.2.6. Kiểm thử chức năng tìm kiếm 41](#_Toc192098373)

[4.2.7. Kiểm thử chức năng quản lý sản phẩm 43](#_Toc192098374)

[4.2.8. Kiểm thử chức năng quản lý danh mục 45](#_Toc192098375)

[4.2.9. Kiểm thử chức năng quản lý nhãn hiệu 46](#_Toc192098376)

[4.2.10. Kiểm thử chức năng quản lý tài khoản 48](#_Toc192098377)

[4.2.11. Kiểm thử chức năng trang chủ 51](#_Toc192098378)

[4.3. Thực thi các ca kiểm thử trên Selenium Webdriver 52](#_Toc192098379)

[4.3.1. Kiểm thử chức năng đăng ký tài khoản 52](#_Toc192098380)

[4.3.2. Kiểm thử chức năng đăng nhập 54](#_Toc192098381)

[4.3.3. Kiểm thử chức năng đăng xuất 56](#_Toc192098382)

[4.3.4. Kiểm thử chức năng thêm sản phẩm vào giỏ hàng 57](#_Toc192098383)

[4.3.5. Kiểm thử chức năng thanh toán 59](#_Toc192098384)

[4.3.6. Kiểm thử chức năng tìm kiếm 61](#_Toc192098385)

[4.3.7. Kiểm thử chức năng quản lý sản phẩm 63](#_Toc192098386)

[4.3.8. Kiểm thử chức năng quản lý danh mục 65](#_Toc192098387)

[4.3.9. Kiểm thử chức năng quản lý nhãn hiệu 67](#_Toc192098388)

[4.3.10. Kiểm thử chức năng quản lý tài khoản 69](#_Toc192098389)

[4.3.11. Kiểm thử giao diện trang chủ 71](#_Toc192098390)

[KẾT LUẬN 73](#_Toc192098391)

[TÀI LIỆU THAM KHẢO 74](#_Toc192098392)

# DANH MỤC CÁC BẢNG

[Bảng 1 Một số lệnh thường dùng trong Robot Framework 25](#_Toc192098299)

[Bảng 2 Kịch bản kiểm thử chức năng đăng ký tài khoản 36](#_Toc192098300)

[Bảng 3 Kịch bản kiểm thử chức năng đăng nhập 39](#_Toc192098301)

[Bảng 4 Kịch bản kiểm thử chức năng đăng xuất 39](#_Toc192098302)

[Bảng 5 Kịch bản kiểm thử chức năng thêm sản phẩm vào giỏ hàng 40](#_Toc192098303)

[Bảng 6 Kịch bản kiểm thử chức năng thanh toán 41](#_Toc192098304)

[Bảng 7 Kịch bản kiểm thử chức năng tìm kiếm 43](#_Toc192098305)

[Bảng 8 Kịch bản kiểm thử chức năng quản lý sản phẩm 45](#_Toc192098306)

[Bảng 9 Kịch bản kiểm thử chức năng quản lý danh mục 46](#_Toc192098307)

[Bảng 10 Kịch bản kiểm thử chức năng quản lý nhãn hiệu 48](#_Toc192098308)

[Bảng 11 Kịch bản kiểm thử chức năng quản lý tài khoản 51](#_Toc192098309)

[Bảng 12 Kịch bản kiểm thử chức năng trang chủ 51](#_Toc192098310)

# CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

## Kiểm thử phần mềm và các khái niệm liên quan

### Khái niệm kiểm thử phần mềm

Kiểm thử phần mềm (software testing) là hoạt động nhằm tìm kiếm và phát hiện ra các lỗi của phần mềm, đảm bảo phần mềm chính xác, đúng và đầy đủ theo yêu cầu của khách hàng, yêu cầu của sản phẩm đã đặt ra. Software testing cũng cung cấp mục tiêu, cái nhìn độc lập về phần mềm điều này cho phép đánh giá và hiểu rõ các rủi ro khi thực thi phần mềm.

### Vai trò của kiểm thử phần mềm

Kiểm thử phần mềm đóng vai trò thiết yếu trong quy trình phát triển phần mềm, giúp đảm bảo chất lượng và hiệu suất của sản phẩm trước khi ra mắt. Trong đó, những vai trò chính của kiểm thử phần mềm bao gồm:

* **Đảm bảo chất lượng sản phẩm:** Kiểm thử phần mềm đảm bảo rằng phần mềm hoàn thành đáp ứng được các mục tiêu về kỹ thuật và chức năng. Qua quá trình kiểm tra toàn diện, các lỗi hoặc khiếm khuyết trong phần mềm sẽ được phát hiện và sửa chữa, từ đó cải thiện tính ổn định của sản phẩm, ngăn ngừa các vấn đề có thể phát sinh sau khi phần mềm được triển khai thực tế.
* **Tiết kiệm chi phí sửa chữa:** Việc phát hiện lỗi và sự cố sớm trong quá trình phát triển sẽ giúp tiết kiệm chi phí và thời gian đáng kể so với việc khắc phục lỗi khi phần mềm đã đi vào hoạt động
* **Nâng cao trải nghiệm của người dùng:** Phần mềm được kiểm thử kĩ lưỡng sẽ mang lại trải nghiệm người dùng mượt mà, tránh các sự cố gián đoạn trong quá trình sử dụng. Hơn nữa, trải nghiệm tốt cũng gia tăng khả năng giữ chân khách hàng và thu hút người dùng mới cho doanh nghiệp.
* **Tăng cường bảo mật:** Quá trình kiểm thử giúp phát hiện các lỗ hổng bảo mật và các điểm yếu trong phần mềm, từ đó khắc phục và tăng cường khả năng phòng vệ trước các cuộc tấn công mạng.

### Phân loại kiểm thử phần mềm

#### Kiểm thử chức năng (Functional Testing)

Kiểm thử chức năng là quá trình xác minh xem hệ thống có hoạt động đúng theo các yêu cầu nghiệp vụ đã được đặt ra hay không. Phương pháp này thường được thực hiện theo hai hướng: dựa trên yêu cầu (requirements-based) và dựa trên quy trình nghiệp vụ (business – process – based).

Đối với kiểm thử dựa trên yêu cầu, các yêu cầu được sắp xếp theo mức độ ưu tiên dựa trên tiêu chỉ rủi ro, nhằm đảm bảo các phần quan trọng nhất của hệ thống sẽ được kiểm thử kỹ lưỡng.

Bên cạnh đó, kiểm thử dựa trên quy trình nghiệp vụ sẽ hoạt động dựa vào kiến thức về quy trình nghiệp vụ để tạo ra các kịch bản kiểm thử. Các quy trình nghiệp vụ thường mô tả các hoạt động hàng ngày của hệ thống. Ví dụ, trong một hệ thống quản lý đơn hàng, quy trình nghiệp vụ có thể là: khách hàng đặt hàng, hệ thống xử lý đơn hàng, sau đó đơn hàng được giao, và cuối cùng khách hàng xác nhận nhận hàng. Từ những quy trình này, ta có thể tạo ra các kịch bản kiểm thử mô phỏng các tình huống thực tế mà người dùng sẽ trải qua.

Ngoài ra kiểm thử chức năng còn được chia nhỏ thành các loại:

* Kiểm thử đơn vị (Unit Testing)
* Kiểm thử giao diện (Interface Testing)
* Kiểm thử hệ thống (System Testing)
* Kiểm thử hồi quy (Regression Testing)
* Kiểm thử chấp nhận (Acceptance Testing)

#### Kiểm thử phi chức năng (Non-Functional Testing)

Kiểm thử phi chức năng tập trung vào các yếu tố như hiệu suất, tính bảo mật, khả năng mở rộng, khả năng tương thích và tính ổn định của phần mềm. Loại kiểm thử này giúp đánh giá các thuộc tính quan trọng nhưng không liên quan trực tiếp đến chức năng phần mềm.

Ngoài ra, kiểm thử phi chức năng cũng bao gồm nhiều loại như sau:

* **Kiểm thử độ ổn định (Stability testing):** Xác định liệu phần mềm có thể duy trì hoạt động ổn định trong khoảng thời gian dài hay không.
* **Kiểm thử khả năng chịu tải (Load testing):** Đánh giá cách hệ thống hoạt động khi phải xử lý khối lượng công việc ngày càng tăng, giúp xác định ngưỡng mà phần mềm có thể hoạt động mà không gặp vấn đề.
* **Kiểm thử áp lực (Stress testing):** Xem xét hiệu suất của hệ thống khi phải làm việc dưới áp lực hoặc vượt quá mức tải dự kiến.
* **Kiểm thử tính khả dụng (Usability testing):** Kiểm tra mức độ thân thiện với người dùng của sản phẩm, đảm bảo người dùng có trải nghiệm dễ dàng và hiệu quả khi sử dụng phần mềm.
* **Kiểm thử bảo trì (Maintainability testing):** Đánh giá khả năng phần mềm có thể được sửa chữa, cải tiến và kiểm tra lại sau khi được thay đổi hoặc cập nhật.
* **Kiểm thử độ tin cậy (Reliability testing):** Sử dụng các công cụ để phát hiện và loại bỏ lỗi, đảm bảo hệ thống hoạt động ổn định trước khi triển khai.
* **Kiểm thử tính tương thích (Portability testing):** Đánh giá khả năng của phần mềm trong việc chuyển đổi hoặc triển khai trên các môi trường khác nhau, đảm bảo tính linh hoạt và hiệu quả.

#### Kiểm thử cấu trúc (Structural Testing)

Kiểm thử cấu trúc là phương pháp kiểm thử dựa trên cấu trúc bên trong của phần mềm, thường Liên quan đến việc kiểm tra mã nguồn hoặc kiến trúc hệ thống.

Các mục tiêu của kiểm thử cấu trúc bao gồm:

* **Nhận ra những điểm bất cập:** Kiểm thử cấu trúc giúp xác định các điểm yếu trong mã nguồn hoặc cấu trúc hệ thống mà có thể gây ra sự cổ trong quá trình vận hành, bao gồm việc phát hiện các đoạn mã chưa tối ưu, lỗi logic....
* **Test chức năng bổ sung:** Kiểm thử cấu trúc kiểm tra các chức năng có thể đã bị bỏ qua trong quá trình phát triển. Điều này giúp đảm bảo rằng tất cả các đoạn mã trong hệ thống đều được kiểm thử đầy đủ, không có đoạn mã nào không được kiểm tra.
* **Xác định những phần bị thiếu trong bộ kiểm thử:** Khi thực hiện kiểm thử cấu trúc, người kiểm thử có thể phát hiện ra những phần của hệ thống chưa được bao phủ bởi bộ kiểm thử hiện tại, từ đó bổ sung các trường hợp kiểm thử còn thiếu.

#### Kiểm thử liên quan đến các thay đổi (Change Related Testing)

Khi phần mềm được cập nhật hoặc sửa đổi, việc kiểm thử lại các tính năng bị ảnh hưởng bởi sự thay đổi là cực kỳ quan trọng. Kiểm thử liên quan đến thay đổi bao gồm hai phương pháp chính:

* **Kiểm thử hồi quy (Regression Testing):** Đảm bảo rằng các tính năng cũ vẫn hoạt động bình thường sau khi phần mềm có thay đổi. Phương pháp này đặc biệt quan trọng khi phần mềm trải qua các bản cập nhật, vì một thay đổi nhỏ có thể gây ra lỗi không mong muốn trong vận hành hệ thống.
* **Kiểm thử xác nhận (Confirmation Testing):** Đảm bảo rằng lỗi hoặc vấn đề trước đó đã được khắc phục đúng cách và không còn xuất hiện trong phiên bản mới của phần mềm.

Tóm lại, kiểm thử liên quan đến thay đổi là phương pháp bảo đảm rằng các cải tiến hoặc sửa đổi không làm gián đoạn hoạt động của hệ thống.

## Kiểm thử tự động

### Khái niệm kiểm thử tự động

Kiểm thử tự động là quá trình xử lý một cách tự động các bước thực hiện các testcase, kiểm thử tự động bằng một công cụ nhằm rút ngắn thời gian kiểm thử.

Kiểm thử tự động là một kỹ thuật tự động trong đó người kiểm thử tự viết các tập lệnh và sử dụng phần mềm phù hợp để kiểm thử phần mềm. Về cơ bản là một quá trình tự động hóa của một quy trình kiểm thử thủ công. Giống như kiểm thử hồi quy, kiểm thử tự động cũng được sử dụng để kiểm thử ứng dụng theo quan điểm tải, hiệu năng và ứng suất.

Kiểm thử tự động giúp giảm chi phí kiểm thử bằng cách hỗ trợ quá trình kiểm thử thông qua các công cụ phần mềm.

Kiểm thử tự động hay sử dụng phần mềm để kiểm thử với các ưu điểm:

* Có thể thực hiện các kiểm thử một cách liên tục, lặp lại và giảm chi phí cho nhân lực kiểm thử.
* Luôn đảm bảo hoạt động theo một kịch bản duy nhất – không bị ảnh hưởng như với kiểm thử viên.

### Quy trình kiểm thử tự động

Quy trình kiểm thử tự động bao gồm: tester sử dụng các kịch bản tự động (automation scripts) và thực thi các script để chạy ứng dụng với sự giúp sức của các automation tool. Một khi script đã sẵn sàng thì việc thực thi kiểm thử có thể diễn ra nhanh chóng và hiệu quả.

Các hoạt động của kiểm thử tự đông:

* Phân tích yêu cầu/Xác định môi trường/công cụ
* Xác định tiêu chí đầu ra
* Lên kế hoạch và kiểm soát
* Thiết lập môi trường kiểm thử
* Triển khai thiết kế kiểm thử
* Thực thi kiểm thử
* Phân tích, báo cáo

### Ưu, nhược điểm của kiểm thử tự động

***Ưu điểm***

* Độ tin cậy: Công cụ kiểm thử tự động mang lại tính ổn định cao hơn so với con người. Đặc biệt, trong những trường hợp nhiều test case nên độ tin cậy của kiểm thử sẽ đạt mức tối ưu hơn so với kiểm thử thủ công.
* Khả năng lặp lại: Với những công cụ kiểm thử tự động sẽ ra đời nhiều các tester không phải trải qua quá trình lặp lại nhiều thao tác giảm được sự nhàm chán cũng như độ tin cậy cao và ổn định.
* Khả năng tái sử dụng: Đây chính là một bộ kiểm thử tự động được nhiều người sử dụng với nhiều những phiên bản khác nhau và được gọi là tái sử dụng.
* Tốc độ cao: với tốc độ kiểm thử nhanh hơn so với tốc độ của con người. Những thực thi của một test case một cách thủ công có thể hoàn thành hay thực thi trong thời gian ngắn nhất một cách tự động.
* Chi phí thấp: chi phí sử dụng kiểm thử tự động khá hấp dẫn và phù hợp có thể tiết kiệm được nhiều chi phí cũng như thời gian nhân lực. Do quá trình kiểm thử nhanh hơn thử thủ công nên nhân lực sẽ được thực thi và bảo trì không nhiều.

***Nhược điểm***

* Khó bảo trì và mở rộng: Trong một dự án để có thể mở rộng được phạm vi kiểm thử tự động sẽ khá khó so với những kiểm thử thủ công vì quá trình cập nhật và chỉnh sửa theo những yêu cầu của Debug, thay đổi dữ liệu, cập nhật code mới.
* Khả năng bao phủ kém: Do mở rộng nên sự đòi hỏi kỹ năng lập trình là rất cần thiết và khá cầu kỳ để phủ sóng được những kiểm thử tự động thấp xét trên nhiều góc nhìn toàn diện.
* Vấn đề công cụ và nhân lực: Hiện nay có nhiều công cụ hỗ trợ kiểm thử tự động cũng khá tốt như QTP, Selenium, Test Complete, Visual Studio, ... nhưng nhìn chung vẫn còn rất nhiều mặt hạn chế, nguồn nhân lực đạt yêu cầu cũng không nhiều.

### Các nguyên tắc trong kiểm thử tự động

* **Nguyên tắc 1:** Kiểm thử chứng minh sự hiện diện của lỗi

Kiểm thử có thể cho thấy rằng phần mềm có lỗi, nhưng không thể chứng minh rằng phần mềm không có lỗi. Kiểm thử làm giảm xác suất lỗi chưa tìm thấy vẫn còn trong phần mềm, thậm chí khi kiểm thử không phát hiện lỗi nào thì việc kiểm thử đó cũng không thể để chứng minh rằng phần mềm không có lỗi

* **Nguyên tắc 2:** Kiểm thử toàn bộ là không thể

Kiểm thử với tất cả các kết hợp đầu vào và đầu ra, với tất cả các kịch bản là không thể trừ khi kiểm thử chỉ bao gồm một số ít trường hợp thì có thể kiểm thử toàn bộ.Thay vì kiểm thử toàn bộ, việc phân tích rủi ro và dựa trên sự mức độ ưu tiên người kiểm thử có thể tập trung việc kiểm thử vào một số điểm cần thiết, có nguy cơ lỗi cao hơn. Nghĩa là phải lên kế hoạch kiểm thử, thiết kế trường hợp kiểm thử sao cho có độ bao phủ nhiều nhất và giảm thiểu rủi ro sót lỗi khi đến tay người dùng.

* **Nguyên tắc 3:** Kiểm thử sớm tiết kiệm thời gian và tiền bạc

Các khiếm khuyết được loại bỏ sớm trong quá trình sẽ không gây ra các khiếm khuyết tiếp theo trong các sản phẩm công việc phái sinh. Chi phí chất lượng sẽ giảm vì ít lỗi xảy ra hơn sau này trong SDLC (Boehm 1981). Để tìm ra các khiếm khuyết sớm, cả thử nghiệm tĩnh và thử nghiệm động nên được bắt đầu càng sớm càng tốt.

* **Nguyên tắc 4:** Lỗi gom thành nhóm

Một số lượng nhỏ các mô-đun (cụm chức năng) thường chứa hầu hết lỗi được phát hiện trong quá trình kiểm thử phần mềm trước khi phát hành, đưa nó đến tay người dùng. Hoặc các mô-đun này gây ra phần lớn các hỏng hóc trong quá trình vận hành. Vì vậy, các cụm lỗi được dự đoán hay được quan sát thực tế trong quá trình kiểm thử hoặc vận hành là thông tin quan trọng để phân tích rủi ro. Từ đó giúp bạn tập trung kiểm thử một cách phù hợp dựa vào mật độ lỗi của các mô-đun

* **Nguyên tắc 5:** Đề phòng nghịch lý thuốc trừ sâu

Nếu các bài kiểm tra giống nhau được lặp lại nhiều lần, chúng sẽ ngày càng kém hiệu quả trong việc phát hiện các lỗi mới (Beizer 1990). Để khắc phục hiệu ứng này, các bài kiểm tra và dữ liệu kiểm tra hiện có có thể cần được sửa đổi và các bài kiểm tra mới có thể cần được viết. Tuy nhiên, trong một số trường hợp, việc lặp lại các bài kiểm tra giống nhau có thể mang lại kết quả có lợi, ví dụ như trong thử nghiệm hồi quy tự động.

* **Nguyên tắc 6:** Kiểm thử phụ thuộc vào ngữ cảnh

Không có cách tiếp cận duy nhất có thể áp dụng chung cho việc thử nghiệm. Việc thử nghiệm được thực hiện khác nhau trong các bối cảnh khác nhau (Kaner 2011).

* **Nguyên tắc 7:** Quan niệm sai về phần mềm không có lỗi

Thật là sai lầm (tức là một quan niệm sai lầm) khi mong đợi rằng xác minh phần mềm sẽ đảm bảo sự thành công của một hệ thống. Việc kiểm tra kỹ lưỡng tất cả các yêu cầu đã chỉ định và sửa tất cả các lỗi được tìm thấy vẫn có thể tạo ra một hệ thống không đáp ứng được nhu cầu và kỳ vọng của người dùng, không giúp đạt được mục tiêu kinh doanh của khách hàng và kém hơn so với các hệ thống cạnh tranh khác. Ngoài việc xác minh, cũng nên thực hiện xác thực (Boehm 1981).

### Các phương pháp kiểm thử tự động

#### Phương pháp kiểm thử Fuzz Testing

Fuzz Testing (hay còn gọi là Fuzzing) là một kỹ thuật kiểm thử phần mềm nhằm phát hiện lỗi bảo mật hoặc lỗi logic bằng cách cung cấp các dữ liệu ngẫu nhiên, không mong đợi hoặc không hợp lệ vào hệ thống.

##### Khái niệm Fuzz Testing

Fuzz Testing là một phương pháp kiểm thử tự động trong đó phần mềm sẽ được cung cấp các dữ liệu đầu vào ngẫu nhiên để kiểm tra xem hệ thống có bị lỗi hoặc gặp sự cố nào không.

##### Các giai đoạn của kiểm thử Fuzz

Quá trình Fuzz Testing thường bao gồm các bước:

* Chuẩn bị dữ liệu đầu vào: Xây dựng hoặc tạo ra dữ liệu ngẫu nhiên hoặc có chủ đích.
* Thực hiện kiểm thử: Cung cấp dữ liệu cho hệ thống và quan sát phản hồi.
* Giám sát hành vi hệ thống: Kiểm tra xem phần mềm có bị crash, leak bộ nhớ hoặc có lỗi bảo mật nào không.
* Phân tích kết quả: Xác định nguyên nhân gây lỗi và đề xuất các bản vá.

##### Phân loại kiểm thử Fuzz

Fuzz Testing có thể được chia thành các loại:

* Black-box Fuzzing: Không cần biết về cấu trúc bên trong của ứng dụng, chỉ dựa vào đầu vào và đầu ra.
* White-box Fuzzing: Dựa vào hiểu biết về mã nguồn để tạo ra dữ liệu kiểm thử phù hợp.
* Grey-box Fuzzing: Kết hợp cả hai phương pháp trên, sử dụng một phần thông tin nội bộ để tối ưu kiểm thử.

##### Các lỗ hổng được phát hiện bởi kiểm thử Fuzz

Một số lỗi phổ biến có thể được phát hiện:

* Buffer Overflow: Khi chương trình ghi quá nhiều dữ liệu vào bộ nhớ.
* SQL Injection/XSS: Tấn công bằng cách chèn mã độc vào đầu vào của hệ thống.
* Crash ứng dụng: Khi chương trình không xử lý đúng dữ liệu đầu vào, gây ra lỗi nghiêm trọng.

##### Ưu, nhược điểm của kiểm thử Fuzz

***Ưu điểm:***

* Tự động hóa cao, giảm bớt gánh nặng cho tester.
* Phát hiện nhiều lỗi bảo mật quan trọng.
* Có thể kiểm thử trên quy mô lớn và với tốc độ cao.

***Nhược điểm:***

* Khó xác định nguyên nhân chính xác gây ra lỗi.
* Có thể tạo ra quá nhiều false positive (cảnh báo sai).
* Cần kết hợp với các phương pháp kiểm thử khác để đạt hiệu quả tối đa.

#### Phương pháp dựa trên mô hình (Model Based Testing)

Model Based Testing là một phương pháp kiểm thử trong đó các test case được tạo ra từ một mô hình trừu tượng của hệ thống.

##### Khái niệm kiểm thử dựa trên mô hình

MBT sử dụng mô hình hệ thống để tạo test case tự động. Mô hình có thể là dạng đồ thị trạng thái, bảng chuyển trạng thái hoặc mô hình toán học.

##### Các loại kiểm thử dựa trên mô hình

* Finite State Machine (FSM): Dựa trên mô hình trạng thái hữu hạn.
* Decision Table Testing: Sử dụng bảng quyết định để kiểm tra nhiều kịch bản khác nhau.
* Data Flow Testing: Dựa trên luồng dữ liệu trong chương trình.

##### Các mô hình khác nhau trong kiểm thử

* State Transition Models: Dùng cho hệ thống có nhiều trạng thái và chuyển đổi giữa chúng.
* Control Flow Models: Dùng trong kiểm thử luồng điều khiển của chương trình.
* Data Flow Models: Theo dõi cách dữ liệu di chuyển qua hệ thống.

##### Ưu, nhược điểm của kiểm thử dựa trên mô hình

***Ưu điểm:***

* Giúp phát hiện lỗi từ giai đoạn thiết kế.
* Tạo test case tự động, giảm công sức kiểm thử thủ công.
* Cải thiện độ bao phủ kiểm thử.

***Nhược điểm:***

* Cần thời gian và công sức để xây dựng mô hình.
* Không phù hợp cho hệ thống nhỏ và đơn giản.

#### Các phương pháp kiểm thử tự động khác

Ngoài Fuzz Testing và Model Based Testing, còn có các phương pháp kiểm thử tự động phổ biến khác như:

* **Unit Testing (Kiểm thử đơn vị):** Kiểm thử từng thành phần nhỏ của phần mềm, thường được thực hiện bởi các lập trình viên.

Công cụ: JUnit, NUnit, PyTest.

* **Integration Testing (Kiểm thử tích hợp):** Kiểm thử sự kết hợp giữa các module trong hệ thống.

Công cụ: TestNG, JUnit.

* **Regression Testing (Kiểm thử hồi quy):** Đảm bảo các thay đổi mới không làm ảnh hưởng đến chức năng cũ.

Công cụ: Selenium, TestComplete.

* **Performance Testing (Kiểm thử hiệu năng):** Kiểm tra tốc độ, khả năng mở rộng của hệ thống.

Công cụ: JMeter, LoadRunner.

* **Security Testing (Kiểm thử bảo mật):** Tìm lỗ hổng bảo mật trong hệ thống.

Công cụ: OWASP ZAP, Burp Suite.

* **UI Testing (Kiểm thử giao diện người dùng):** Kiểm tra xem giao diện có hoạt động đúng không.

Công cụ: Selenium, Cypress.

* **Mutation Testing (Kiểm thử đột biến):** Thay đổi mã nguồn để kiểm tra độ mạnh của test case.

### Các công cụ hỗ trợ kiểm thử tự động

**Selenium:** được sử dụng để kiểm tra trình duyệt chéo ( cross-browser testing) và tự động hóa kiểm tra trình duyệt web (web-browser test automation). Để sử dụng công cụ này, người kiểm tra phải có kỹ năng lập trình nâng cao để viết các kịch bản kiểm tra phức tạp và nâng cao. Những kỹ năng này là cần thiết để xây dựng các khung và thư viện tự động hóa cho các nhu cầu kiểm thử cụ thể. Selenium là một công cụ nguồn mở thường được sử dụng bởi các nhà phát triển và người thử nghiệm, những người thông thạo các ngôn ngữ lập trình như Java, C #, Perl, Python, Scala, Groovy, PHP & Ruby. Selenium được trang bị Selenium WebDriver mạnh mẽ cho phép bạn tạo các bộ kiểm tra và tự động hồi quy dựa trên trình duyệt. Một trong những lợi thế chính của công cụ này là nó cho phép bạn chia tỷ lệ và phân phối các tập lệnh trên nhiều môi trường.

**IBM Rational Functional Tester (RFT):** IBM RFT rất phù hợp để kiểm thử hồi quy (regression testing) và kiểm thử chức năng (functional testing). Đây là một nền tảng kiểm thử (testing platform) dựa trên dữ liệu hỗ trợ các ứng dụng như .Net, Java, SAP, Flex và Ajax. Các ngôn ngữ kịch bản được sử dụng bởi RFT là .Net và Java. Một trong những tính năng độc đáo của IBM RFT là kiểm thử Storyboard (Storyboard testing) giúp đơn giản hóa kiểm thử trực quan bằng cách ghi lại và trực quan hóa các hành động của người dùng với sự trợ giúp của các ảnh chụp màn hình ứng dụng theo định dạng bảng phân cảnh. Nó cũng cho phép chỉnh sửa bằng ngôn ngữ tự nhiên. Nó cũng cung cấp tích hợp với quản lý vòng đời ứng dụng IBM Jazz như IBM Rational Team Concert và Rational Quality Manager.

**Cucumber:** là khung phát triển hướng hành vi (BDD) cho phép xác thực chức năng theo định dạng dễ hiểu và dễ đọc. BDD là một cách tiếp cận mở rộng của Phát triển hướng thử nghiệm (Test Driven Development) và nó được sử dụng để kiểm tra toàn bộ hệ thống thay vì kiểm tra một đoạn mã cụ thể. Cucumber là một công cụ để làm việc với các thông số kỹ thuật thực thi. Các thông số kỹ thuật thực thi được sử dụng cho sự hợp tác lớn hơn giữa các nhóm CNTT và doanh nghiệp. Công cụ này hữu ích để viết acceptance tests cho ứng dụng web. Cucumber cung cấp các tệp tính năng có thể được sử dụng làm tài liệu của các Nhà phân tích, Nhà phát triển và Người kiểm tra, v.v. Cucumber hỗ trợ các ngôn ngữ như Perl, PHP, Python, Net, v.v.

**TestComplete:** cho phép bạn xây dựng và chạy các kiểm thử giao diện người dùng chức năng (functional UI tests). Đây là một công cụ kiểm tra tự động hóa rất phù hợp để kiểm tra các ứng dụng máy tính để bàn, thiết bị di động và web. Công cụ cho phép bạn tạo ra các trường hợp kiểm thử bằng hầu hết các ngôn ngữ phổ biến như Python, JavaScript và VBScript, v.v ... Nó cho phép bạn ghi lại và phát lại các bài kiểm thử. Nó cung cấp các khả năng nhận dạng đối tượng GUI UI tự động phát hiện và cập nhật các đối tượng UI. Nó giúp giảm bớt những nỗ lực cần thiết để duy trì các kịch bản kiểm thử (test scripts) . Với TestComplete, việc kiểm thử quy mô trên 1500+ môi trường thử nghiệm thực tế là khá dễ dàng để cung cấp phạm vi kiểm tra hoàn chỉnh.

# CHƯƠNG 2. CÁC KỸ THUẬT VÀ CÔNG CỤ KIỂM THỬ TỰ ĐỘNG

## 2.1. Công cụ kiểm thử Selenium – Selenium Automated Testing

### 2.1.1. Tổng quan về Selenium

Selenium là một phần mềm mã nguồn mở - một công cụ kiểm thử phần mềm tự động để kiểm thử các ứng dụng trên nền Web. Selenium chỉ tập trung vào việc tự động hóa các ứng dụng dựa trên web. Việc kiểm thử được sử dụng bằng công cụ Selenium thường được gọi là Automation Selenium Testing. Selenium không chỉ là một công cụ duy nhất mà nó là một bộ phần mềm, mỗi bộ nó cung cấp các nhu cầu thử nghiệm khác nhau của một tổ chức. Selenium có 4 thành phần là Selenium Integrated Development Environment (Selenium IDE), Selenium Remote Control (Selenium RC), Selenium WebDriver, Selenium Grid.

**Selenium IDE** là một công cụ kiểm thử tự động trên trình duyệt, giúp đơn giản hóa quá trình kiểm thử cho người dùng nhờ khả năng ghi và phát lại thao tác mà không cần lập trình phức tạp. Là tiện ích mở rộng trên trình duyệt, Selenium IDE cho phép ghi lại từng bước tương tác của người dùng với ứng dụng web, từ việc nhấp chuột, nhập liệu đến điều hướng trang. Công cụ này phù hợp với các kiểm thử đơn giản và cung cấp tính năng báo cáo chi tiết, hỗ trợ người dùng nhận diện nhanh các lỗi tiềm ẩn.

**Selenium Remote Control (RC)** là một công cụ trong bộ Selenium hỗ trợ tự động hóa kiểm thử trên trình duyệt web thông qua mã lệnh. Khác với Selenium IDE, Selenium RC có kiến trúc client-server, với Selenium Server đóng vai trò như một proxy trung gian, giúp gửi lệnh kiểm thử từ mã của người dùng đến trình duyệt. Nhờ đó, Selenium RC vượt qua được các hạn chế cùng nguồn gốc, cho phép thực hiện kiểm thử trên nhiều miền khác nhau. Công cụ này hỗ trợ đa ngôn ngữ lập trình như Java, C#, Ruby, Perl và Python, mang lại tính linh hoạt cao trong việc xây dựng kịch bản kiểm thử. Với khả năng kiểm thử phân tán, Selenium RC có thể chạy trên nhiều trình duyệt và hệ điều hành khác nhau, phù hợp với các dự án lớn yêu cầu kiểm thử trên nhiều môi trường.

**Selenium WebDriver** là một công cụ mạnh mẽ trong bộ Selenium, được thiết kế để tự động hóa kiểm thử ứng dụng web trên nhiều trình duyệt khác nhau. Khác với Selenium RC, WebDriver tương tác trực tiếp với trình duyệt thông qua API gốc, cho phép thực hiện các thao tác kiểm thử một cách nhanh chóng và chính xác hơn. WebDriver hỗ trợ đa nền tảng, có thể hoạt động trên các trình duyệt phổ biến như Chrome, Firefox, Safari và Edge, cũng như trên nhiều hệ điều hành như Windows, macOS và Linux. Bên cạnh đó, Selenium WebDriver cho phép người dùng viết kịch bản kiểm thử bằng nhiều ngôn ngữ lập trình như Java, Python, C#, và JavaScript, mang lại tính linh hoạt và khả năng tích hợp cao trong các dự án phát triển phần mềm.

**Selenium Grid** là một công cụ trong bộ Selenium cho phép thực hiện kiểm thử tự động trên nhiều máy tính và trình duyệt khác nhau cùng một lúc, giúp tiết kiệm thời gian và nâng cao hiệu quả kiểm thử. Công cụ này sử dụng kiến trúc client-server, trong đó có một máy chủ (hub) điều phối các yêu cầu kiểm thử và nhiều máy khách (node) thực hiện kiểm thử trên các trình duyệt hoặc hệ điều hành khác nhau. Selenium Grid hỗ trợ chạy các kịch bản kiểm thử song song, cho phép kiểm tra tính tương thích của ứng dụng trên nhiều nền tảng và trình duyệt khác nhau một cách đồng thời. Điều này đặc biệt hữu ích cho các dự án lớn yêu cầu tính hiệu quả và độ chính xác cao trong quy trình kiểm thử. Selenium Grid giúp các đội ngũ phát triển phần mềm nhanh chóng phát hiện lỗi và cải thiện chất lượng sản phẩm cuối cùng.

### 2.1.2. Selenium Webdriver

#### 2.1.2.1. Khái niệm

WebDriver là một giao diện lập trình ứng dụng (API) cho phép thực hiện thao tác trên các trình duyệt web như mở trang web, nhập dữ liệu, nhấn nút, kéo thả, chụp ảnh màn hình... WebDriver được ra đời vào năm 2006 bởi Simon Stewart, một kỹ sư phần mềm. Năm 2008, WebDriver được kết hợp với Selenium RC để tạo ra Selenium 2.0, phiên bản hiện tại của Selenium.

Selenium WebDriver là một thành phần của Selenium, bao gồm cả Selenium IDE, Selenium Grid và Selenium RC. Selenium WebDriver là phiên bản nâng cấp của Selenium RC, với khả năng tương tác trực tiếp với trình duyệt và thực hiện các thao tác như click, nhập liệu, kéo thả, chụp ảnh màn hình,... Selenium WebDriver hỗ trợ nhiều ngôn ngữ lập trình và nhiều trình duyệt khác nhau rất thân thiện và dễ sử dụng.

#### 2.1.2.2. Ưu, nhược điểm của Selenium Webdriver

***Ưu điểm:***

WebDriver có nhiều ưu điểm so với các công cụ kiểm thử tự động khác, đặc biệt là so với Selenium RC, phiên bản trước đó của Selenium. Một số ưu điểm chính của WebDriver là

Mã nguồn mở: Selenium WebDriver là một dự án mã nguồn mở, có nghĩa là bất kỳ ai cũng có thể sử dụng và đóng góp cho dự án. Điều này giúp Selenium WebDriver được phát triển và cập nhật thường xuyên, đảm bảo luôn đáp ứng được các nhu cầu của người dùng.

Tương thích nhiều trình duyệt: Selenium WebDriver hỗ trợ nhiều trình duyệt phổ biến, bao gồm Chrome, Firefox, Edge, Safari,... Điều này giúp bạn có thể sử dụng Selenium WebDriver để kiểm thử các ứng dụng web trên nhiều trình duyệt khác nhau.

Hỗ trợ đa ngôn ngữ lập trình: Selenium WebDriver hỗ trợ nhiều ngôn ngữ lập trình phổ biến, bao gồm Java, Python, C#... Điều này giúp bạn có thể sử dụng ngôn ngữ lập trình mà mình mong muốn để tạo các kịch bản kiểm thử tự động

Tốc độ thực thi nhanh: Selenium Webdriver có tốc độ thực thi nhanh và hiệu quả.

Nâng cao chất lượng ứng dụng: Selenium WebDriver có thể giúp bạn nâng cao chất lượng ứng dụng web của mình bằng cách phát hiện và khắc phục các lỗi sớm.

***Nhược điểm:***

Bên cạnh những ưu điểm nổi bật, Selenium WebDriver cũng có một số nhược điểm, như:

Chỉ hỗ trợ ứng dụng web: Selenium WebDriver chỉ hỗ trợ kiểm thử tự động các ứng dụng web. Điều này có nghĩa là nếu bạn muốn kiểm tra tốc độ ứng dụng trên Windows, bạn sẽ gặp nhiều khó khăn vì Selenium WebDriver không hỗ trợ đầy đủ các tính năng và công cụ tiện ích trên hệ điều hành này. Đây là một nhược điểm lớn của Selenium WebDriver khi làm việc với Windows.

Đòi hỏi kinh nghiệm lập trình: Để sử dụng Selenium WebDriver, bạn cần có kiến thức về lập trình. Điều này có thể là một trở ngại lớn đối với những bạn không có kinh nghiệm lập trình.

## 2.2. Robot Framework

Robot Framework là một testing framework. Nó cung cấp nền tảng kiểm thử cho tester dựa trên ngôn ngữ lập trình Python. Cách tiếp cận của nền tảng kiểm thử này là hướng từ khoá (keyword driven) và hướng dữ liệu (data driven) dành cho việc kiểm thử để nghiệm thu sản phẩm ngay từ đầu (end-to-end acceptance testing).... Điểm hấp dẫn của Robot Framework với các tester chính là tester chỉ cần viết kịch bản kiểm thử theo hướng từ khóa (keyword driven) và hướng dữ liệu (data driven). Tester có thể tạo các từ khóa cấp cao mới từ những cái hiện có bằng cách sử dụng cú pháp tương tự được sử dụng để tạo ra các trường hợp thử nghiệm và mọi thứ chúng ta cần làm chính là viết ra một kịch bản kiểm thử dựa trên các từ khóa (keyword).

### 2.2.1. Các tính năng của Robot Framework

Dễ dàng thực hiện kiểm thử từ động với các kịch bản ở dạng bảng.

Robot Framework đưa ra kết quả thực thi các kịch bản kiểm thử và các log ở dạng html. Điều này giúp chúng ta đọc và phân tích kết quả dễ dàng và nhanh chóng.

Robot Framework hỗ trợ chức năng đánh dấu các kịch bản kiểm thử nên cho phép chúng ta lựa chọn kịch bản để thực thi một cách dễ dàng và tiện lợi.

Cuối cùng, thế mạnh nhất của Robot Framework so với các nền tảng kiểm thử khác là khả năng chạy trên nhiều hệ điều hành khác nhau mà không cần chỉnh sửa kịch bản kiểm thử hay các từ khoá ở tầng dưới.

### 2.2.2. Cài đặt Robot Framework

* **Bước 1:** Cài đặt Python.

Cài đặt python ở web chính thức: <https://www.python.org/> và tải xuống phiên bản phù hợp với hệ điều hành của bạn (Windows, macOS, Linux).

Sau khi cài đặt xong thì dùng lệnh sau để kiểm tra xem đã cài thành công chưa: *python --version*. Nếu thấy trả về phiên bản thì đã tải về thành công.

* **Bước 2:** Cài đặt PIP (Python Package Manager).

Mở cửa sổ dòng lệnh và gõ dòng lệnh sau: *pip --version*. Nếu trả về phiên bản thì PIP đã được cài đặt. Nếu chưa thì có thể cài theo hướng dẫn trên web: <https://stackoverflow.com/questions/64684571/how-do-you-download-pip>

* **Bước 3:** Cài đặt **Robot Framework**

**Cài đặt Robot Framework:** Mở cửa sổ dòng lệnh và gõ lệnh sau: *pip install robotframework*. Lệnh này sẽ tự động tải và cài đặt Robot Framework cùng với các thư viện và công cụ cần thiết.

Kiểm tra cài đặt Robot Framework: Gõ lệnh sau trong cửa sổ dòng lệnh: *robot --version*. Nếu thấy thông tin về phiên bản của Robot Framework, nghĩa là bạn đã cài đặt thành công.

* **Bước 4:** Cài đặt thư viện **SeleniumLibrary**

Mở cửa sổ dòng lệnh và gõ lệnh sau:

*pip install robotframework-seleniumlibrary*

* **Bước 5:** Cài đặt **IDE ( RIDE )**

Mở cửa sổ dòng lệnh và gõ lệnh sau để cài đặt RIDE:

*pip install robotframework-ride*

Lệnh này sẽ tự động tải và cài đặt RIDE cùng với các thư viện và công cụ cần thiết.

### 2.2.3. Các câu lệnh trong Robot Framework

Robot Framework cung cấp rất nhiều lệnh hữu ích để thực hiện các tác vụ tự động hóa, đặc biệt khi kết hợp với thư viện SeleniumLibrary để tương tác với giao diện web. Dưới đây là một số lệnh thường dùng trong Robot Framework chia theo các nhóm chức năng chính:

Bảng sau đây cung cấp 1 số lệnh thường dùng trong Robot Framework kết hợp với thư viện SeleniumLibrary

|  |  |
| --- | --- |
| **Tên lệnh** | **Ý nghĩa** |
| Open Browser | Mở một trình duyệt cụ thể với URL cho trước |
| Close Browser | Đóng trình duyệt hiện tại |
| Click Element | Nhấn vào một thành phần dựa trên Xpath, ID hoặc tên |
| Input Text | Nhập văn bản vào trường đầu vào |
| Select From List By Value/Label | Chọn một mục trong danh sách dựa trên giá trị hoặc nhãn |
| Check Checkbox | Đánh dấu hộp kiểm |
| Uncheck Checkbox | Bỏ đánh dấu hộp kiểm |
| Element Should Be Visible | Kiểm tra xem một phần tử có hiển thị trên trang không |
| Text Should Be | Kiểm tra văn bản của một phần tử cụ thể |
| Element Should Contain | Kiểm tra phần tử có chứa một đoạn văn bản cụ thể |
| Get Text | Lấy văn bản từ phần tử |
| Get Title | Lấy tiêu đề trang hiện tại |
| Wait Until Element Is Visible | Chờ cho đến khi phần tử hiển thị |
| Wait Until Page Contains Element | Chờ cho đến khi trang chứa một phần tử cụ thể |
| Maximize Browser Window | Mở rộng cửa sổ trình duyệt |
| Scroll Element Into View | Cuộn trang để một phần tử cụ thể hiển thị trên màn hình |
| getWindowHandle() | Sử dụng để giải quyết tình huống khi có nhiều cửa sổ cần xử lý. Command này giúp chuyển sang cửa sổ mới và thực hiện các thao tác trên cửa sổ mới này. Người dùng cũng có thể chuyển trở lại cửa sổ trước đó nếu muốn. |

Bảng Một số lệnh thường dùng trong Robot Framework

# CHƯƠNG 3. CÁC YÊU CẦU VÀ TIÊU CHUẨN KIỂM THỬ TRÊN WEB

## 3.1. Khái quát về kiểm thử ứng dụng trên Web

### 3.1.1. Khái quát

Khi mạng Internet ngày càng phát triển, môi trường mạng đem đến nhiều cơ hội kinh doanh, tiếp cận khách hàng thì hiển nhiên việc thiết kế website và các ứng dụng chạy trên nền Web là cần thiết để chiếm lĩnh thị trường. Các ứng dụng Web phát triển và đóng vai trò to lớn trong việc kết nối, trao đổi thông tin của nhiều doanh nghiệp.

Muốn có được sự thành công kể trên, trước hết các ứng dụng chạy trên nền Web phải có chất lượng tốt, hiệu năng cao, chưa kể tới các yếu tố về giao diện, trải nghiệm người dùng, v.v. Ngoài ra, chúng ta đều biết ứng dụng trên nền Web có những đặc thù khác biệt hoàn toàn so với ứng dụng di động, ứng dụng desktop, v.v. Ứng dụng trên nền Web không giới hạn chỉ ở điện thoại thông minh, máy vi tính hay máy tính bảng, mà được thiết kế để chạy trên nhiều nền tảng khác nhau. Mỗi nền tảng lại có những yêu cầu riêng về cấu hình, độ phân giải, đặc thù thao tác, v.v. Đó chính là những vấn đề lớn đặt ra cho các nhà phát triển phần mềm trong việc đảm bảo chất lượng cho các ứng dụng trên nền Web khi phải chạy trên đa nền tảng. Vì thế cần phải đưa ra một chiến lược hiệu quả cho kiểm thử, tránh những rủi ro, nâng cao chất lượng cho ứng dụng Web.

### 3.1.2. Các loại ứng dụng Web

**Ứng dụng Web tĩnh:** Là loại ứng dụng Web hiển thị ít nội dung và không có tính linh hoạt. Ứng dụng Web tĩnh thường chỉ được xây dựng từ HTML, CSS và Javascript. Do không có cơ sở dữ liệu và công cụ điều khiển nội dung gián tiếp nên người quản trị không thể tùy ý thay đổi nội dung mà cần có kiến thức về HTML, CSS cơ bản để chỉnh sửa. Điểm cộng của loại website này là nội dung đơn giản, không mất nhiều thời gian, công sức để xây dựng do không phải xử lý những câu lệnh phức tạp. Tuy nhiên, do không có hệ thống hỗ trợ thay đổi nội dung nên việc cập nhật thông tin cho website gặp rất nhiều khó khăn, thậm chí phải bỏ ra chi phí lớn khi thay đổi nhiều lần. Một số ví dụ cho Web tĩnh là những trang giới thiệu công ty, tổ chức, sản phẩm, khóa học ít có nhu cầu cập nhật thông tin, v.v

**Ứng dụng Web động:** So với web tĩnh thì Web động phức tạp hơn về mặt kỹ thuật khi xây dựng. Web động sử dụng cơ sở dữ liệu để hiển thị nội dung cũng như cho phép người dùng tương tác được với nội dung đó. Web động được chia làm 2 phần là back-end (dành cho người quản trị Web thay đổi, cập nhật nội dung) và front-end (dành cho người dùng truy cập). Hiện nay có rất nhiều ngôn ngữ lập trình được sử dụng để xây dựng Web động như Java, PHP, ASP.NET, VB.NET, Ruby, v.v. Đối với Web động, việc cập nhật nội dung là rất đơn giản và dễ dàng. Không những thế, một số hệ thống lớn hiện nay còn cho phép người quản trị có thể thay đổi giao diện Web trên trang quản trị mà không cần phải can thiệp trực tiếp vào mã nguồn. Đó là những lý do khiến cho Web động được sử dụng phổ biến hơn Web tĩnh.

## 3.2. Công việc chính khi kiểm thử ứng dụng Web

### 3.2.1. Kiểm thử chức năng

Kiểm thử chức năng yêu cầu kiểm thử viên thực hiện kiểm thử 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. Ngoài ra còn có kết nối cơ sở dữ liệu, kiểm tra cookie và xác minh HTML/CSS, v.v.

* **Kiểm thử giao diện:** Trước khi lập trình viên bắt tay vào xây dựng mã nguồn sẽ luôn có một bản thiết kế UI quy định giao diện của ứng dụng Web. Mỗi thành phần textbox, button, image, link và bố cục trên ứng dụng Web đều được chỉ ra một cách cụ thể trong tài liệu này. Bản thiết kế UI thường sẽ kèm theo file .PSD – bản vẽ giao diện của ứng dụng Web sử dụng phần mềm Photoshop để lập trình viên dễ dàng xây dựng cũng như khách hàng có thể biết trước ứng dụng của mình sẽ hiển thị ra sao. Đây cũng là tài liệu không thể thiếu cho kiểm thử viên so sánh, đối chiếu giữa thiết kế và nội dung thực tế của ứng dụng hiển thị trên trình duyệt.
* **Kiểm thử các liên kết và menu:** Trong một ứng dụng Web có 2 loại liên kết: liên kết nội bộ (internal link) và liên kết ngoại bộ (external link). Cả 2 loại liên kết trên đều cần được kiểm tra xem chúng có hoạt động không? Có trỏ đến địa chỉ mong muốn không? Cần đảm bảo rằng các liên kết không tự trỏ đến vị trí của chính nó. Ngoài ra cũng cần xem xét thuộc tính “target” của các liên kết xem chúng có hoạt động đúng như bản thiết kế yêu cầu hay không.
* **Kiểm thử các form nhập dữ liệu:** Cần đảm bảo các trường nhập liệu được thiết kế đúng kiểu loại, có bộ lọc kiểm tra tính đúng đắn của dữ liệu nhập vào (validation) trước khi gửi đi (submit) tránh việc hacker có thể tận dụng lỗ hổng SQL Injection từ chính các form nhập liệu trên ứng dụng Web. Ngoài ra cũng cần đảm bảo sự toàn vẹn dữ liệu trong quá trình truyền tải thông tin từ trình duyệt tới server, nhất là đối với các ứng dụng thương mại điện tử, ngân hàng, v.v.
* **Kiểm thử lỗi cú pháp HTML/CSS:** Ở bước tiếp theo, người kiểm thử cần xác định các thẻ CSS bị lỗi hoặc các thuộc tính, id, class được viết trong thẻ HTML không hợp lệ hoặc không thuộc bất kỳ thẻ CSS nào.
* **Kiểm thử cookie và session:** Kiểm thử các ứng dụng đăng nhập trong phiên bằng cách cho phép và vô hiệu hóa các tập tin cookie. Có thể thử đưa lỗi vào ứng dụng Web bằng cách sử dụng một tên miền không phù hợp như cố tình truyền sai, thiếu tham số, v.v. Ngoài ra, cần kiểm tra khả năng bảo mật của ứng dụng Web bằng cách xóa các tập tin cookie có chọn lọc khi kiểm thử.
* **Kiểm thử nội dung đa ngôn ngữ:** Bước kiểm thử này đặc biệt cần thiết với những ứng dụng Web hỗ trợ đa ngôn ngữ để đảm bảo thông tin khi dịch sang các ngôn ngữ khác nhau luôn được sát nghĩa, không bị tràn dòng khi dịch, các yếu tố về chính tả được tuân thủ.
* **Kiểm thử cơ sở dữ liệu (database):** Kiểm tra kết nối tới cơ sở dữ liệu và các lỗi truy vấn có thể gặp phải, đảm bảo dữ liệu được cung cấp chính xác khi các chức năng xem thông tin, thêm, sửa, xóa, v.v. hoạt động

### 3.2.2. Kiểm thử khả năng

* **Kiểm thử nội dung:** Chúng ta cần đảm bảo nội dung trong ứng dụng được sắp xếp hợp lý và dễ hiểu với người dùng, không mắc các lỗi chính tả, các hình ảnh hiển thị chính xác về vị trí, kích thước. Ngoài ra cũng cần chú trọng tới màu sắc, font chữ phù hợp với mọi đối tượng sử dụng.
* **Kiểm thử logic các liên kết và hướng dẫn:** Đối với người dùng lần đầu tiên truy cập một ứng dụng Web, họ luôn gặp những khó khăn nhất định trong việc sử dụng. Vì vậy cần kiểm tra xem các hướng dẫn, liên kết, thông báo đã được bố trí đầy đủ trên ứng dụng hay chưa? Tuy nhiên, việc xuất hiện quá nhiều hướng dẫn, thông báo ở mọi nơi trên ứng dụng cũng khiến người dùng rối mắt, không thoải mái khi sử dụng. Tốt nhất nên đảm bảo các hướng dẫn, thông báo đưa ra hết sức ngắn gọn nhưng đủ ý ngay tại nơi người dùng có thể gặp khó khăn khi sử dụng.
* **Kiểm thử văn hoá khu vực và đối tượng sử dụng:**

Điều này bắt nguồn từ đặc điểm riêng của từng lĩnh vực (ví dụ y khoa thường dùng màu sáng để thể hiện sự sạch sẽ), hoặc văn hóa riêng từng khu vực (người châu Á thường chuộng tông màu nóng và thiết kế cầu kỳ hơn châu Âu).

Thêm vào đó, trong quá trình kiểm thử phải luôn bảo đảm rằng chuẩn thiết kế ứng dụng Web của mình có thể được tìm thấy phổ biến ở nhiều ứng dụng Web khác cùng loại. Ví dụ như button Đăng nhập, Đăng xuất thường nằm ở góc trên bên phải và menu chính luôn nằm ở trên cho tất cả trang Web con. Nếu một ứng dụng Web trong lĩnh vực khoa học lại trình bày bằng font chữ cách điệu lòe loẹt, tiêu đề chạy ngang dọc, hoặc một ứng dụng Web dành cho trẻ em lại chỉ dùng 2 tông màu đen trắng buồn tẻ thì nên góp ý với bộ phận thiết kế

### 3.2.3. Kiểm thử sự tương thích

Một ứng dụng Web thường hỗ trợ nhiều thiết bị, môi trường khác nhau. Vì vậy kiểm thử độ tương thích của ứng dụng Web là một điều không dễ dàng khi công nghệ của các nền tảng thay đổi quá nhanh chóng.

* **Kiểm thử tương thích theo thiết bị, hệ điều hành:** Khó có ứng dụng Web nào chạy hoàn hảo trên tất cả các môi trường, vì vậy người kiểm thử cần đặt ưu tiên cho những môi trường cần hỗ trợ để tiết kiệm thời gian cho việc kiểm thử. Có hai điều cần lưu tâm nhất khi kiểm thử khả năng tương thích của ứng dụng với thiết bị, đó là: khung hình và khả năng hỗ trợ của thiết bị với các phiên bản HTML. Người kiểm thử cần truy cập tất cả các nội dung trên từng loại thiết bị, có thể xoay ngang, dọc màn hình (đối với thiết bị di động, máy tính bảng) để xem ứng dụng Web được hiển thị như thế nào, chạy thử từng chức năng trên ứng dụng để đảm bảo chúng hoạt động như mong muốn.
* Kiểm thử tương thích với trình duyệt:

+ Cần chạy thử ứng dụng trên một số trình duyệt phổ biến hiện nay như IE, Chrome, Firefox, Opera, Safari, v.v. để đảm bảo hoạt động chính xác trên các trình duyệt khác nhau.

+ Kiểm tra hoạt động các chức năng của ứng dụng khi thực hiện cài đặt, cấu hình bảo mật cho trình duyệt.

+ Mỗi trình duyệt lại có nhiều phiên bản cập nhật khác nhau, cần kiểm tra sự nhất quán của ứng dụng khi chạy trên các phiên bản đó.

+ Kiểm tra hoạt động của ứng dụng khi bật/tắt flash, cookie, java, v.v.

### 3.2.4. Kiểm thử hiệu suất

* **Kiểm thử khả năng tải (Load test):** Ở bước này cần xác định thời gian thực thi cho các hành động tương ứng với các chức năng trên ứng dụng. Công việc này cần được thực hiện ở nhiều thời điểm khác nhau (giờ cao điểm/thấp điểm) để có những đánh giá khách quan nhất về khả năng tải của ứng dụng.
* **Kiểm thử độ chịu lỗi (Stress test):** Công việc này chính là kiểm tra sức chịu đựng của ứng dụng Web khi có lượng truy cập cao từ phía người dùng. Trong thực tế đó có thể là nhu cầu sử dụng thực sự của người dùng đối với ứng dụng hoặc khi máy chủ bị tấn công dưới dạng Ddos. Nói cách khác, người kiểm thử cần trả lời câu hỏi: Số lượng người truy cập cùng lúc là bao nhiêu sẽ đánh sập hệ thống? Hay đơn giản hơn là khi lượng người truy cập tăng lên ở các mức khác nhau, ứng dụng còn hoạt động ổn định hay không? Trả lời được các câu hỏi trên sẽ giúp cho ứng dụng Web khi đưa vào hoạt động tránh được những rủi ro không đáng có và lường trước những nguy cơ có thể xảy ra.

### 3.2.5. Kiểm thử bảo mật

Ứng dụng Web là một trong những loại ứng dụng có nguy cơ bị tấn công cao nhất. Vì vậy, ngoài việc đảm bảo ứng dụng chạy đúng, ổn định cần phải kiểm tra nghiêm ngặt khả năng bảo mật của ứng dụng. Các công việc cần làm có thể kể đến như:

* Kiểm tra độ tin cậy của việc phân quyền sử dụng trên ứng dụng.
* Đưa lỗi vào bằng cách truyền các tham số không hợp lệ trên URL hay trong các form nhập liệu. Lỗ hổng SQL Injection được khai thác mạnh nhất thông qua các thành phần trên.
* Kiểm tra khả năng truy cập trái phép đối với những thư mục bị cấm trên máy chủ của ứng dụng.
* Kiểm tra hoạt động các bộ lọc (validation) khi sử dụng chức năng upload tệp tin, thư mục của ứng dụng (nếu có).
* Kiểm tra độ xác thực khi nhập CAPTCHA trong ứng dụng (nếu có).

## 3.3. Một số công cụ hỗ trợ

Công việc cần làm đối với một kiểm thử viên kiểm thử ứng dụng trên nền Web như đã nói ở phần trước là rất nhiều. Những công cụ kiểm thử ra đời để hỗ trợ cho các kiểm thử viên thực hiện công việc một cách nhanh chóng, bớt nhàm chán và giảm thiểu chi phí kiểm thử. Đồ án này sẽ giới thiệu một số công cụ hỗ trợ kiểm thử ứng dụng trên nền Web, phân loại dựa trên mục đích sử dụng.

### 3.3.1. Công cụ kiểm thử hiệu năng

Dưới đây là danh sách một số công cụ kiểm thử hiệu năng được sử dụng rộng rãi nhất để đo hiệu suất ứng dụng Web và khả năng chịu tải của chúng. Các công cụ kiểm tra tải này sẽ đưa ra đánh giá về hiệu suất của ứng dụng trong thời gian có lưu lượng truy cập cao điểm.

* **WebLoad:** Cho phép thực hiện kiểm thử khả năng chịu tải và độ chịu lỗi của ứng dụng Web bằng cách sử dụng Ajax, Adobe Flex, .NET, Oracle Forms, HTML5 và nhiều công nghệ khác. Điểm mạnh của WebLoad là là dễ sử dụng với các tính năng như cho phép ghi/phát lại dựa trên DOM, tương quan tự động và ngôn ngữ kịch bản Javascript. Công cụ này hỗ trợ thử nghiệm hiệu suất quy mô lớn với các kịch bản phức tạp và đưa ra những phân tích rõ ràng.
* **Apache JMeter:** Đây là một công cụ phát triển trên mã nguồn mở. Apache Jmeter được coi như một công cụ kiểm thử hiệu năng, có khả năng tích hợp với kế hoạch kiểm thử. Ngoài việc kiểm thử hiệu năng, Apache JMeter còn có thể sử dụng để kiểm tra các chức năng của ứng dụng Web.
* **NeoLoad:** Công cụ sử dụng để đo và phân tích hiệu suất của ứng dụng Web. NeoLoad phân tích hiệu suất của ứng dụng Web bằng cách tăng lưu lượng truy cập vào ứng dụng. Nhờ đó, kiểm thử viên có thể biết được năng lực chịu tải của ứng dụng. Công cụ này được viết trên nền Java, tương thích với nhiều hệ điều hành khác nhau và hỗ trợ hai ngôn ngữ: Tiếng Anh và tiếng Pháp.
* **LoadStorm:** Là một công cụ kiểm thử cho các ứng dụng Web và mobile. Điểm mạnh là nó có thể kiểm tra hiệu năng của ứng dụng dựa trên số lượng người dùng và lưu lượng truy cập. LoadStorm cũng có khả năng chịu tải rất tốt khi mà nó có thể giả lập hàng trăm nghìn đến hàng triệu user để tìm kiếm các breaking point (điểm dừng) trong ứng dụng. Các kịch bản kiểm thử của LoadStorm có thể được chỉnh sửa bởi kiểm thử viên.

### 3.3.2. Công cụ kiểm thử bảo mật

* **Burp Suite:** Là một công cụ kiểm tra lỗ hổng bảo mật cho ứng dụng Web. Nó có nhiều công cụ tích hợp trong đó hai công cụ chính trong phiên bản miễn phí là Spider and Intruder. Spider được sử dụng để thu thập thông tin các trang của ứng dụng và Intruder được sử dụng để thực hiện các cuộc tấn công tự động trên ứng dụng Web. Burp có một công cụ bổ sung hiện nay được gọi là Burp Scanner được dùng trong việc quét các lỗ hổng có trong ứng dụng.
* **OWASP Zed Attack Proxy:** Tương tự như Burp Suite, OWASP Zed Attack Proxy là công cụ để thâm nhập, đánh giá an ninh mạng, bảo mật của các ứng dụng Web.
* **Nikto:** Công cụ đánh giá hệ thống Nikto là một máy quét lỗ hổng máy chủ Web mã nguồn mở. Nó phát hiện việc cài đặt phần mềm và cấu hình đã lỗi thời, các tệp tin có khả năng nguy hiểm, v.v
* **Exploit-Me:** Là một công cụ kiểm tra bảo mật ứng dụng Web có thể tích hợp trên trình duyệt Firefox được thiết kế nhỏ gọn, dễ sử dụng. Exploit- Me bao gồm các gói: XSS-Me và SQL Inject-Me. Cross-Site Scripting (XSS) là một lỗ hổng được tìm thấy trong nhiều ứng dụng Web hiện nay. Lỗ hổng XSS có thể gây ra thiệt hại nghiêm trọng cho một ứng dụng Web. XSS-Me là công cụ giúp phát hiện ra các lỗ hổng XSS này. Trong khi đó, SQL Inject-Me được sử dụng để kiểm tra các lỗ hổng SQL Injection trong ứng dụng Web.

### 3.3.3. Công cụ kiểm thử chức năng

* **BrowserStack:** Đây là một công cụ giúp kiểm thử hoạt động của các chức năng trên ứng dụng Web trên nhiều trình duyệt khác nhau. Ứng dụng Web có thể được kiểm tra bằng thao tác của người dùng hoặc tự động thông qua Selenium. Ngoài ra, BrowserStack còn cung cấp tính năng chụp ảnh màn hình ứng dụng Web trên 650 trình duyệt khác nhau và kiểm tra khả năng hiển thị responsive trên các loại màn hình.
* **Ranorex:** Công cụ kiểm thử tự động cho các ứng dụng Web, desktop và di động. Chỉ với một tài khoản, người dùng có thể sử dụng Ranorex để kiểm thử cho 3 loại ứng dụng kể trên. Việc tích hợp này sẽ giúp rút ngắn thời gian khi kiểm thử ứng dụng được thiết kế chạy trên nhiều nền tảng khác nhau. Tuy nhiên, bản trả phí của Ranorex khá đắt, lên tới 3500$/năm.
* **Selenium:** Là một trong những công cụ kiểm thử tự động ứng dụng Web mạnh mẽ nhất hiện nay. Selenium script có thể chạy trên hầu hết các trình duyệt hiện nay như IE, Chrome, Firefox, Safari, Opera và các hệ điều hành phổ biến như Windows, Mac, Linux. Trong thực tế, người ta thường sử dụng Selenium dưới dạng Add-on tích hợp trong trình duyệt Firefox, kết hợp cùng với Firebug để kiểm thử ứng dụng Web một cách hiệu quả nhất. Tuy chỉ có thể ghi lại (Record) hành động trên trình duyệt Firefox, nhưng có thể phát lại (Playback) trên nhiều trình duyệt phổ biến khác. Vì là công cụ mã nguồn mở nên Selenium có ưu thế lớn so với các công cụ kiểm thử tự động khác: có cộng đồng hỗ trợ mạnh mẽ và không phải trả phí bản quyền. Công cụ này hỗ trợ khá nhiều ngôn ngữ lập trình Web phổ biến hiện nay. Ngoài ra, Selenium được phát triển bởi Selenium team từ Google nên người dùng hoàn toàn yên tâm về chất lượng và độ tin cậy của Selenium.

# CHƯƠNG 4. ỨNG DỤNG KIỂM THỬ TỰ ĐỘNG CHO WEBSITE BÁN HÀNG

## 4.1. Mô tả hệ thống thử nghiệm

### 4.1.1. Tổng quan về hệ thống thử nghiệm

Hệ thống thử nghiệm là website bán máy tính được phát triển bằng Java Springboot với mục tiêu cung cấp nền tảng bán lẻ máy tính trực tuyến. Website hỗ trợ các chức năng quan trọng sau:

**Quản lý tài khoản:**

* Đăng ký, đăng nhập bằng email
* Cập nhật thông tin cá nhân, thay đổi mật khẩu, xem lịch sử mua hàng

**Quản lý giỏ hàng và đặt hàng:**

* Thêm/xóa sản phẩm vào giỏ hàng, cập nhật số lượng sản phẩm
* Xác nhận đơn hàng, xem chi tiết đơn hàng đã đặt

**Quản lý sản phẩm:**

* Danh sách sản phẩm theo hãng
* Tìm kiếm sản phẩm theo danh mục, hãng, khoảng giá

**Quản lý đơn hàng:**

* Theo dõi trạng thái đơn hàng (đang xử lý, đã giao, đã hủy)
* Xác nhận giao hàng thành công

**Liên hệ:**

* Hỗ trợ liên hệ tư vấn với cửa hàng

### 4.1.2. Công nghệ sử dụng trong hệ thống

Hệ thống được xây dựng bằng công nghệ Java Spring Boot và MySQL.

### 4.1.3. Môi trường thử nghiệm

Để kiểm thử hệ thống, một môi trường thử nghiệm được thiết lập như sau:

* Hệ điều hành: Windows 10
* Trình duyệt kiểm thử: Google Chrome (v.110 trở lên)
* Thiết bị kiểm thử: Laptop (Windows)
* Công cụ hỗ trợ kiểm thử: Robot Framework, Selenium

### 4.1.4. Mục tiêu kiểm thử

**Kiểm thử chức năng (Functional Testing)**

Kiểm thử chức năng được thực hiện nhằm đảm bảo các tính năng trên website bán hàng máy tính hoạt động chính xác theo yêu cầu. Các trường hợp kiểm thử bao gồm:

* Quản lý tài khoản:

+ Đăng ký: Kiểm tra khả năng đăng ký tài khoản mới với các trường hợp hợp lệ và không hợp lệ (trùng email, thiếu thông tin, mật khẩu không khớp,...).

+ Đăng nhập: Xác thực thông tin đăng nhập, bao gồm đăng nhập đúng, sai, và các trường hợp để trống trường thông tin.

+ Đăng xuất: Kiểm tra chức năng đăng xuất đảm bảo người dùng được đưa về trang đăng nhập và không thể truy cập trang quản trị hay giỏ hàng sau khi đăng xuất.

* Quản lý sản phẩm, nhãn hiệu, danh mục:

+ Thêm, cập nhật, xóa sản phẩm và đảm bảo thông tin sản phẩm được cập nhật chính xác trên giao diện.

+ Kiểm tra các chức năng thêm, sửa, xóa danh mục và nhãn hiệu, đảm bảo sự thay đổi được phản ánh đầy đủ trong hệ thống.

* Giỏ hàng:

+ Thêm sản phẩm vào giỏ hàng:

- Thêm sản phẩm khi còn hàng, xác nhận sản phẩm và số lượng chính xác trong giỏ.

- Xử lý tình huống sản phẩm hết hàng, đảm bảo hệ thống ngăn không cho thêm vào giỏ hàng.

+ Cập nhật giỏ hàng:

+ Chỉnh sửa số lượng sản phẩm trong giỏ hàng, đảm bảo không vượt quá giới hạn kho.

+ Xóa sản phẩm khỏi giỏ hàng, đảm bảo tổng tiền và số lượng sản phẩm được cập nhật chính xác.

* Thanh toán (Checkout):

+ Nhập thông tin giao hàng, kiểm tra tính hợp lệ của các trường thông tin như tên, số điện thoại, địa chỉ.

+ Xử lý các tình huống nhập thiếu thông tin, hoặc nhập sai định dạng, đảm bảo hệ thống hiển thị thông báo lỗi rõ ràng.

* Tìm kiếm sản phẩm:

+ Kiểm tra chức năng tìm kiếm, đảm bảo tìm đúng sản phẩm theo từ khóa.

+ Xử lý các trường hợp tìm kiếm sản phẩm không tồn tại, đảm bảo hệ thống hiển thị thông báo phù hợp.

**Kiểm thử giao diện người dùng (UI Testing)**

Kiểm thử giao diện nhằm đảm bảo tính nhất quán và thân thiện với người dùng trên nhiều thiết bị khác nhau. Các tiêu chí kiểm tra bao gồm:

* Hiển thị trên các trình duyệt phổ biến: Chrome, Edge.
* Bố cục và hiển thị các thành phần giao diện:

+ Đảm bảo nút "Thanh toán", "Xác nhận đơn hàng" được hiển thị đầy đủ, không bị ẩn hoặc trùng lặp trên các kích thước màn hình khác nhau.

## 4.2. Xây dựng kịch bản kiểm thử cho từng trường hợp

### 4.2.1. Kiểm thử chức năng đăng ký tài khoản

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TC\_ID** | **Testcase Name** | **Description** | **Pre-condition** | **Test Steps** | **Expected Result** | **Pass/Fail** |
| TC\_REG\_001 | Register with Empty Fields | Kiểm tra đăng ký khi bỏ trống các trường | Truy cập trang đăng ký | 1.Để trống tất cả các trường  2.Nhấn “Đăng ký” | Hiển thị lỗi “Please fill out this field” | Pass |
| TC\_REG\_002 | Register with Existing Email | Kiểm tra đăng ký với email đã tồn tại | Truy cập trang đăng ký | 1.Nhập email đã đăng ký  2.Nhập mật khẩu hợp lệ  3.Nhập thông tin cá nhân  4.Nhấn “Đăng ký” | Hiển thị lỗi “Địa chỉ email đã được sử dụng” | Pass |
| TC\_REG\_003 | Register with Invalid Email Format | Kiểm tra đăng ký với email không hợp lệ | Truy cập trang đăng ký | 1.Nhập email sai định dạng (admin@gmai.com<>)  2.Nhập mật khẩu hợp lệ  3.Nhấn "Đăng ký" | Hiển thị lỗi “A part following '@' should not contain the symbol '<'. “ | Pass |
| TC\_REG\_004 | Register with Short Password | Kiểm tra đăng ký với mật khẩu quá ngắn | Truy cập trang đăng ký | 1.Nhập email hợp lệ  2.Nhập mật khẩu ít hơn 6 ký tự  3.Nhấn "Đăng ký" | Hiển thị lỗi “Mật khẩu phải dài 8-32 ký tự” | Pass |
| TC\_REG\_005 | Register with Mismatched Passwords | Kiểm tra đăng ký khi mật khẩu và xác nhận mật khẩu không khớp | Truy cập trang đăng ký | 1.Nhập email hợp lệ  2.Nhập mật khẩu hợp lệ  3.Nhập lại mật khẩu khác  4.Nhấn "Đăng ký" | Hiển thị lỗi "Nhắc lại mật khẩu không chính xác" | Pass |
| TC\_REG\_006 | Register with Valid Data | Kiểm tra đăng ký với dữ liệu hợp lệ | Truy cập trang đăng ký | 1.Nhập email hợp lệ  2.Nhập mật khẩu hợp lệ  3.Nhập thông tin cá nhân  4.Nhấn "Đăng ký" | Đăng ký thành công, chuyển đến trang chủ | Pass |

Bảng Kịch bản kiểm thử chức năng đăng ký tài khoản

### 4.2.2. Kiểm thử chức năng đăng nhập

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TC\_ID** | **Testcase Name** | **Description** | **Pre-condition** | **Test Steps** | **Expected Result** | **Pass/Fail** |
| TC\_LG\_001 | Login as admin with a valid credential | Xác minh rằng người dùng quản trị có thể đăng nhập vào hệ thống bằng thông tin xác thực hợp lệ | Tài khoản quản trị đã được tạo | 1.Điều hướng đến trang đăng nhập  2.Điền biểu mẫu đăng nhập  3.Nhấn "Đăng nhập" | Chuyển hướng đến trang Admin | Pass |
| TC\_LG\_002 | Login as normal user with a valid credential | Xác minh rằng người dùng phổ thông có thể đăng nhập vào hệ thống bằng thông tin xác thực hợp lệ | Tài khoản người dùng đã được tạo | 1.Điều hướng đến trang đăng nhập  2.Điền biểu mẫu đăng nhập  3.Nhấn "Đăng nhập" | Chuyển hướng đến trang chủ | Pass |
| TC\_LG\_003 | Login after failed attempt | Xác minh rằng người dùng đăng nhập thành công vào hệ thống sau một lần đăng nhập không thành công | Mở trình duyệt | 1. Nhập sai thông tin đăng nhập  2.Nhấn "Đăng nhập"  3. Nhập lại đúng thông tin đăng nhập  4.Nhấn "Đăng nhập" | Chuyển hướng đến trang chủ | Pass |
| TC\_LG\_004 | Login with an invalid credential | Xác minh rằng đăng nhập bằng thông tin không hợp lệ | Mở trình duyệt | 1.Điều hướng đến trang đăng nhập  2.Nhập tên người dùng hoặc mật khẩu sai  3.Nhấn "Đăng nhập" | Hiển thị thông báo lỗi | Pass |
| TC\_LG\_005 | Login with a non-existing account | Xác minh rằng người dùng không thể đăng nhập bằng tài khoản không tồn tại | Mở trình duyệt | 1.Điều hướng đến trang đăng nhập  2. Nhập tài khoản không tồn tại (tên đăng nhập)  3.Nhấn "Đăng nhập" | Hiển thị thông báo lỗi | Pass |
| TC\_LG\_006 | Login Without Username | Xác minh rằng người dùng không thể đăng nhập nếu không có Tên người dùng | Mở trình duyệt | 1.Điều hướng đến trang đăng nhập  2.Nhập mật khẩu  3.Nhấn "Đăng nhập" | Hiển thị lỗi "Vui lòng nhập tên người dùng" | Pass |
| TC\_LG\_007 | Login Without Password | Xác minh rằng người dùng không thể đăng nhập nếu không có Mật khẩu | Mở trình duyệt | 1.Điều hướng đến trang đăng nhập  2.Nhập tên người dùng  3.Nhấn "Đăng nhập" | Hiển thị lỗi "Vui lòng nhập mật khẩu" | Pass |
| TC\_LG\_008 | Login Without Username And Password | Xác minh rằng người dùng không thể đăng nhập nếu không có Tên người dùng và Mật khẩu | Mở trình duyệt | 1.Điều hướng đến trang đăng nhập  2.Nhấn "Đăng nhập" | Hiển thị lỗi "Vui lòng nhập thông tin đăng nhập" | Pass |
| TC\_LG\_009 | Login with Special characters | Xác minh rằng người dùng không thể đăng nhập nếu tên đăng nhập là kí tự đặc biệt | Mở trình duyệt | 1. Điều hướng đến trang đăng nhập  2. Nhập tên người dùng với ký tự đặc biệt  3. Nhấn "Đăng nhập" | Hiển thị thông báo lỗi | Pass |
| TC\_LG\_010 | Login with Numeric characters | Xác minh rằng người dùng không thể đăng nhập nếu tên đăng nhập là kí tự số | Mở trình duyệt | 1. Điều hướng đến trang đăng nhập  2. Nhập tên người dùng với ký tự số  3. Nhấn "Đăng nhập" | Hiển thị thông báo lỗi | Pass |

Bảng Kịch bản kiểm thử chức năng đăng nhập

### 4.2.3. Kiểm thử chức năng đăng xuất

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TC\_ID** | **Testcase Name** | **Description** | **Pre-condition** | **Test Steps** | **Expected Result** | **Pass/Fail** |
| TC\_01 | Logout Successfully | Xác nhận người dùng đăng xuất thành công | Người dùng đã đăng nhập | Nhấn "Đăng xuất" | Đăng xuất thành công | Pass |

Bảng Kịch bản kiểm thử chức năng đăng xuất

### 4.2.4. Kiểm thử chức năng thêm sản phẩm vào giỏ hàng

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TC\_ID** | **Testcase Name** | **Description** | **Pre-condition** | **Test Steps** | **Expected Result** | **Pass/Fail** |
| TC\_CART\_001 | Add Single Product to Cart | Kiểm tra thêm một sản phẩm vào giỏ hàng | Người dùng đang ở trang chủ | Chọn một sản phẩm → Nhấn "Giỏ hàng" | Sản phẩm được thêm vào giỏ hàng thành công | Pass |
| TC\_CART\_002 | Add Multiple Products to Cart | Kiểm tra thêm nhiều sản phẩm vào giỏ hàng | Người dùng đang ở trang chủ | Chọn nhiều sản phẩm → Nhấn "Giỏ hàng" cho từng sản phẩm | Tất cả sản phẩm được thêm vào giỏ hàng | Pass |
| TC\_CART\_003 | Add Same Product Multiple Times | Kiểm tra thêm cùng một sản phẩm nhiều lần | Người dùng đang ở trang chủ | Chọn một sản phẩm → Nhấn "Giỏ hàng" nhiều lần | Số lượng sản phẩm trong giỏ tăng lên 2 | Pass |
| TC\_CART\_004 | Add Out-of-Stock Product | Kiểm tra thêm sản phẩm hết hàng | Sản phẩm hết hàng | Chọn một sản phẩm hết hàng→ Nhấn "Giỏ hàng" nhiều lần | Hiển thị thông báo "Sản phẩm đã hết hàng" | Pass |
| TC\_CART\_005 | Increase/Decrease Product Quantity | Kiểm tra tăng/giảm số lượng sản phẩm trong giỏ hàng | Người dùng đã thêm sản phẩm vào giỏ | 1. Vào giỏ hàng  2. Nhấn nút "+"/”-” | Số lượng sản phẩm tăng/giảm và tổng giá cập nhật | Pass |
| TC\_CART\_006 | Remove Product from Cart | Kiểm tra xóa sản phẩm khỏi giỏ hàng | Người dùng đã thêm sản phẩm vào giỏ | Nhấn biểu tượng giỏ hàng → Nhấn vào biểu tượng thùng rác để xóa với sản phẩm cụ thể | Sản phẩm bị xóa khỏi giỏ hàng | Pass |
| TC\_CART\_007 | Add Product Without Login | Kiểm tra thêm sản phẩm khi chưa đăng nhập | Người dùng chưa đăng nhập | Chọn một sản phẩm → Nhấn "Giỏ hàng" | Hiển thị thông báo”Vui lòng đăng nhập” | Fail |

Bảng Kịch bản kiểm thử chức năng thêm sản phẩm vào giỏ hàng

### 4.2.5. Kiểm thử chức năng thanh toán

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TC\_ID** | **Testcase Name** | **Description** | **Pre-condition** | **Test Steps** | **Expected Result** | **Pass/Fail** |
| TC\_CHECKOUT\_001 | Checkout with Empty Cart | Kiểm tra thanh toán khi giỏ hàng không có sản phẩm | Đã đăng nhập và giỏ hàng trống | 1.Truy cập giỏ hàng trống  2.Nhấn “Thanh toán” | Nút thanh toán không hiển thị | Pass |
| TC\_CHECKOUT\_002 | Checkout without Name | Kiểm tra bỏ trống “Họ tên” khi thanh toán | - Người dùng đã đăng nhập  - Giỏ hàng có sản phẩm | 1.Thêm sản phẩm vào giỏ hàng  2.Nhấn “Thanh toán”  3.Bỏ trống “Họ tên”  4.Nhấn “Gửi đơn hàng” | Hiển thị lỗi “Vui lòng nhập vào trường này” | Pass |
| TC\_CHECKOUT\_003 | Checkout with Invalid Phone Number | Kiểm tra nhập số điện thoại sai định dạng | - Người dùng đã đăng nhập  - Giỏ hàng có sản phẩm | 1.Nhập SĐT "hello1234"  2.Nhấn "Thanh toán"” | Hiển thị lỗi "Số điện thoại không hợp lệ" | Fail |
| TC\_CHECKOUT\_004 | Checkout with Short Address | Kiểm tra nhập địa chỉ dưới 5 ký tự | - Người dùng đã đăng nhập  - Giỏ hàng có sản phẩm | 1. Nhập "HN"  2. Nhấn "Thanh toán" | Hiển thị lỗi "Địa chỉ quá ngắn" | Fail |
| TC\_CHECKOUT\_005 | Checkout with Valid Information | Kiểm tra hoàn tất thanh toán với đầy đủ thông tin | - Người dùng đã đăng nhập  - Giỏ hàng có sản phẩm | 1.Thêm sản phẩm vào giỏ  2.Nhập tên, SĐT, địa chỉ  3.Nhấn "Thanh toán" | Hiển thị thông báo "Đặt hàng thành công" | Pass |

Bảng Kịch bản kiểm thử chức năng thanh toán

### 4.2.6. Kiểm thử chức năng tìm kiếm

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TC\_ID** | **Testcase Name** | **Description** | **Pre-condition** | **Test Steps** | **Expected Result** | **Pass/Fail** |
| TC\_SR\_001 | Search with Valid Keyword | Kiểm tra tìm kiếm với từ khóa hợp lệ | Trang chủ đã load thành công | 1. Nhập từ khóa hợp lệ (VD: "laptop") vào thanh tìm kiếm.  2. Click nút tìm kiếm | Danh sách sản phẩm liên quan đến từ khóa được hiển thị đúng | Pass |
| TC\_SR\_002 | Search with Invalid Keyword | Kiểm tra tìm kiếm với từ khóa không hợp lệ | Trang chủ đã load thành công | 1. Nhập từ khóa không hợp lệ (VD: "xyz123") vào thanh tìm kiếm  2. Click nút tìm kiếm | Thông báo "Không tìm thấy sản phẩm" được hiển thị | Pass |
| TC\_SR\_003 | Search with Empty Keyword | Kiểm tra tìm kiếm khi không nhập gì | Trang chủ đã load thành công | 1. Để trống thanh tìm kiếm  2. Click nút tìm kiếm | Hiển thị thông báo lỗi "Vui lòng nhập từ khóa" | Fail |
| TC\_SR\_004 | Search with Special Characters | Kiểm tra tìm kiếm với ký tự đặc biệt | Trang chủ đã load thành công | 1. Nhập ký tự đặc biệt (VD: "@#$%^") vào thanh tìm kiếm  2. Click nút tìm kiếm | Thông báo "Không tìm thấy sản phẩm" được hiển thị | Pass |
| TC\_SR\_005 | Search with Leading/Trailing Spaces | Kiểm tra nhập từ khóa có khoảng trắng ở đầu và cuối | Trang chủ đã load thành công | 1. Nhập từ khóa " laptop " (thừa khoảng trắng)  2. Click nút tìm kiếm | Hệ thống loại bỏ khoảng trắng thừa và tìm kiếm đúng từ khóa | Pass |
| TC\_SR\_006 | Case-Insensitive Search | Kiểm tra tìm kiếm không phân biệt hoa thường | Trang chủ đã load thành công | 1. Nhập từ khóa "LAPTOP"  2. Click nút tìm kiếm | Kết quả tìm kiếm giống như khi nhập "laptop" | Pass |
| TC\_SR\_007 | Filter Products by Category | Kiểm tra lọc sản phẩm theo danh mục cụ thể | Trang chủ hiển thị sản phẩm | 1. Chọn danh mục "Laptop" từ sidebar.  2. Quan sát sản phẩm hiển thị. | Chỉ sản phẩm thuộc danh mục "Laptop" được hiển thị. | Pass |
| TC\_SR\_008 | Filter Products by Price Range | Kiểm tra lọc sản phẩm theo khoảng giá | Trang chủ hiển thị sản phẩm | 1. Chọn khoảng giá "5.000.000 - 10.000.000 VND".  2. Quan sát sản phẩm hiển thị. | Chỉ sản phẩm có giá trong khoảng được hiển thị. | Pass |
| TC\_SR\_009 | Filter Products by Brand | Kiểm tra lọc sản phẩm theo thương hiệu | Trang chủ hiển thị sản phẩm | 1. Chọn thương hiệu "Dell".  2. Quan sát sản phẩm hiển thị. | Chỉ sản phẩm thương hiệu "Dell" được hiển thị. | Pass |
| TC\_SR\_010 | Combine Multiple Filters | Kiểm tra lọc sản phẩm với nhiều điều kiện kết hợp | Trang chủ hiển thị sản phẩm | 1. Chọn danh mục "Laptop".   2. Chọn khoảng giá "5.000.000 - 10.000.000 VND".  3. Chọn "Dell". | Chỉ sản phẩm vừa là Laptop, vừa có giá hợp lệ và thương hiệu "Dell" được hiển thị. | Pass |
| TC\_SR\_011 | Filter with No Matching Results | Kiểm tra khi không có sản phẩm nào phù hợp với bộ lọc | Trang chủ hiển thị sản phẩm | Chọn khoảng giá 2.000.000 | Thông báo "Không tìm thấy sản phẩm phù hợp". | Pass |

Bảng Kịch bản kiểm thử chức năng tìm kiếm

### 4.2.7. Kiểm thử chức năng quản lý sản phẩm

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TC\_ID** | **Testcase Name** | **Description** | **Pre-condition** | **Test Steps** | **Expected Result** | **Pass/Fail** |
| TC\_01 | Upload Product with Valid Data | Kiểm tra upload sản phẩm với dữ liệu hợp lệ | Đã đăng nhập, mở trang quản lý sản phẩm | 1.Click "Add Product" 2.Nhập thông tin hợp lệ (tên, giá, hình ảnh) 3.Click "Save" | Sản phẩm được thêm mới thành công | Pass |
| TC\_02 | Upload Product without Images | Kiểm tra upload sản phẩm không có hình ảnh | Đã đăng nhập, mở trang quản lý sản phẩm | 1.Click "Add Product" 2.Nhập thông tin nhưng không chọn ảnh 3.Click "Save" | Hiển thị lỗi: Hiển thị lỗi "Trường bắt buộc không được để trống | Fail |
| TC\_03 | Upload Product with Missing Fields | Kiểm tra upload sản phẩm thiếu trường thông tin | Đã đăng nhập, mở trang quản lý sản phẩm | 1.Click "Add Product" 2.Bỏ trống tên hoặc giá 3.Click "Save" | Sản phẩm được cập nhật thành công | Pass |
| TC\_04 | Update Product Name | Kiểm tra cập nhật tên sản phẩm | Sản phẩm đã tồn tại | 1. Tìm sản phẩm cần cập nhật 2.Nhập tên mới 3.Click "Save" | Không có thay đổi nào, hiển thị thông báo cập nhật thành công | Pass |
| TC\_05 | Update Without Any Change | Kiểm tra cập nhật sản phẩm mà không thay đổi gì | Sản phẩm đã tồn tại | 1. Tìm sản phẩm 2.Click "Save" mà không thay đổi thông tin | Sản phẩm bị xóa khỏi danh sách | Pass |
| TC\_06 | Delete Product | Kiểm tra xóa sản phẩm | Sản phẩm đã tồn tại | 1. Tìm sản phẩm 2.Click "Delete" 3.Xác nhận xóa | Hiển thị lỗi "Hình ảnh không được để trống" | Fail |
| TC\_07 | Search Product by Category | Kiểm tra tìm kiếm sản phẩm theo danh mục | Danh mục đã tồn tại | 1. Nhập mã không tồn tại vào ô tìm kiếm  2. Click "Search" | Hiển thị thông báo "Không tìm thấy sản phẩm" | Pass |
| TC\_08 | Search Product by Code | Kiểm tra tìm kiếm sản phẩm theo mã sản phẩm | Mã sản phẩm đã tồn tại | 1.Nhập mã sản phẩm vào ô tìm kiếm 2.Click "Search" | Hiển thị thông tin chi tiết của sản phẩm | Pass |
| TC\_09 | Search Non-existent Product | Kiểm tra tìm kiếm sản phẩm không tồn tại | Không có sản phẩm đó | 1.Nhập mã không tồn tại vào ô tìm kiếm 2.Click "Search" | Hiển thị thông báo "Không tìm thấy sản phẩm" | Pass |
| TC\_10 | View Product Details | Kiểm tra xem chi tiết sản phẩm | Sản phẩm đã tồn tại | 1. Tìm sản phẩm 2.Click "View Details" | Hiển thị thông tin chi tiết của sản phẩm | Pass |

Bảng Kịch bản kiểm thử chức năng quản lý sản phẩm

### 4.2.8. Kiểm thử chức năng quản lý danh mục

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TC\_ID** | **Testcase Name** | **Description** | **Pre-condition** | **Test Steps** | **Expected Result** | **Pass/Fail** |
| TC\_CAT\_001 | Add Category with Valid Data | Kiểm tra thêm danh mục mới với dữ liệu hợp lệ | Đã đăng nhập với quyền admin | 1.Nhấn "Thêm danh mục". 2.Nhập tên danh mục 3.Nhấn "Xác nhận". | Hiển thị thông báo "Thêm danh mục thành công". Danh mục mới xuất hiện trong bảng. | Pass |
| TC\_CAT\_002 | Add Category with Empty Name | Kiểm tra lỗi khi thêm danh mục mà để trống tên | Đã đăng nhập với quyền admin | 1.Nhấn "Thêm danh mục". 2.Để trống trường tên danh mục. 3.Nhấn "Xác nhận". | Hiển thị lỗi "Tên danh mục không được để trống". | Pass |
| TC\_CAT\_003 | Add Duplicate Category | Kiểm tra lỗi khi thêm danh mục trùng tên | - Đã đăng nhập với quyền admin - Đã có danh mục "Máy tính bảng" | 1.Nhấn "Thêm danh mục". 2.Nhập tên danh mục 3.Nhấn "Xác nhận". | Hiển thị lỗi "Danh mục đã tồn tại". | Fail |
| TC\_CAT\_004 | Update Category with Valid Data | Kiểm tra chỉnh sửa tên danh mục thành công | - Đã đăng nhập với quyền admin - Đã có danh mục "Máy tính bảng" | 1.Chọn danh mục "Máy tính bảng". 2.Nhập tên mới: "Máy tính để bàn". 3.Nhấn "Xác nhận". | Hệ thống cập nhật tên danh mục mới và thông báo "Cập nhật danh mục thành công" | Pass |
| TC\_CAT\_005 | Update Category with Empty Name | Kiểm tra lỗi khi cập nhật danh mục mà để trống tên | - Đã đăng nhập với quyền admin - Đã có danh mục "Máy tính bảng" | 1.Chọn danh mục "Laptop". 2. Xóa tên danh mục. 3.Nhấn "Xác nhận". | Hiển thị lỗi "Tên danh mục không được để trống" | Pass |
| TC\_CAT\_006 | Delete Category | Kiểm tra xóa danh mục thành công | - Đã đăng nhập với quyền admin - Đã có danh mục "Máy tính bảng" | 1.Chọn danh mục "Máy tính để bàn". 2.Nhấn "Xóa". 3.Xác nhận xóa. | Hiển thị thông báo "Xóa danh mục thành công". Danh mục biến mất khỏi bảng. | Pass |

Bảng Kịch bản kiểm thử chức năng quản lý danh mục

### 4.2.9. Kiểm thử chức năng quản lý nhãn hiệu

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TC\_ID** | **Testcase Name** | **Description** | **Pre-condition** | **Test Steps** | **Expected Result** | **Pass/Fail** |
| TC\_BRAND\_001 | Add Brand with Valid Data | Kiểm tra thêm nhãn hàng mới với dữ liệu hợp lệ | Đã đăng nhập với quyền admin | 1.Nhấn "Thêm nhãn hàng". 2.Nhập tên nhãn hàng 3.Nhấn "Xác nhận". | Hiển thị thông báo "Thêm nhãn hàng thành công". Nhãn hàng mới xuất hiện trong bảng | Pass |
| TC\_BRAND\_002 | Add Brand with Empty Name | Kiểm tra lỗi khi thêm nhãn hàng mà để trống tên | Đã đăng nhập với quyền admin | 1.Nhấn "Thêm nhãn hàng". 2.Để trống trường tên nhãn hàng. 3.Nhấn "Xác nhận". | Hiển thị lỗi "Tên hãng sản xuất không được trống". | Pass |
| TC\_BRAND\_003 | Add Duplicate Brand | Kiểm tra lỗi khi thêm nhãn hàng trùng tên | Đã đăng nhập với quyền admin | 1.Nhấn "Thêm nhãn hàng". 2.Nhập tên nhãn hàng 3.Nhấn "Xác nhận". | Hiển thị lỗi "Nhãn hàng đã tồn tại". | Fail |
| TC\_BRAND\_004 | Update Brand with Valid Data | Kiểm tra chỉnh sửa tên nhãn hàng thành công | - Đã đăng nhập với quyền admin - Đã có hãng sản xuất: LG | 1.Chọn nhãn hàng 2.Nhập tên mới 3.Nhấn "Xác nhận". | Hiển thị thông báo "Cập nhật thành công". Nhãn hàng cập nhật tên mới. | Pass |
| TC\_BRAND\_005 | Update Brand with Empty Name | Kiểm tra lỗi khi cập nhật nhãn hàng mà để trống tên | - Đã đăng nhập với quyền admin - Đã có hãng sản xuất: LG | 1.Chọn nhãn hàng. 2. Xóa tên nhãn hàng. 3.Nhấn "Xác nhận". | Hiển thị lỗi "Tên hãng sản xuất không được để trống". | Pass |
| TC\_BRAND\_006 | Delete Brand | Kiểm tra xóa nhãn hàng thành công | - Đã đăng nhập với quyền admin - Đã có hãng sản xuất: LG | 1.Chọn nhãn hàng. 2.Nhấn "Xóa". 3.Nhấn "Xác nhận". | Hiển thị thông báo "Xóa nhãn hàng thành công". Nhãn hàng biến mất khỏi bảng. | Pass |

Bảng Kịch bản kiểm thử chức năng quản lý nhãn hiệu

### 4.2.10. Kiểm thử chức năng quản lý tài khoản

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TC\_ID** | **Testcase Name** | **Description** | **Pre-condition** | **Test Steps** | **Expected Result** | **Pass/Fail** |
| TC\_ACC\_001 | Add Account with Valid Data | Kiểm tra thêm tài khoản mới với dữ liệu hợp lệ | Đã đăng nhập với quyền admin | 1.Nhấn "Thêm tài khoản". 2.Nhập Email. 3.Nhập mật khẩu. 4: Nhập lại mật khẩu 5.Chọn vai trò: "Admin" 6.Nhập tên. 7. Nhập số điện thoại. 8.Nhập địa chỉ 9.Nhấn "Xác nhận" | Hiển thị thông báo "Thêm tài khoản thành công". Tài khoản mới xuất hiện trong bảng. | Pass |
| TC\_ACC\_002 | Add Account with Empty Fields | Kiểm tra lỗi khi thêm tài khoản mà để trống các trường bắt buộc | Đã đăng nhập với quyền admin | 1.Nhấn "Thêm tài khoản". 2.Để trống tất cả các trường: Email, Mật khẩu, Nhập lại mật khẩu, Tên, SĐT, Địa chỉ, Vai trò. 3.Nhấn "Xác nhận". | - Hiển thị lỗi "Phải nhập địa chỉ email". - Hiển thị lỗi "mật khẩu phải dài 8-32 ký tự" - Hiển thị lỗi " Họ tên không được trống" - Hiển thị lỗi "Số điện thoại không được trống" -Hiển thị lỗi "Địa chỉ không được trống" | Pass |
| TC\_ACC\_003 | Add Account with Short Password | Kiểm tra lỗi khi thêm tài khoản với mật khẩu ngắn hơn mức cho phép | Đã đăng nhập với quyền admin | 1.Nhấn "Thêm tài khoản". 2.Nhập Email. 3.Nhập mật khẩu. 4: Nhập lại mật khẩu 5.Chọn vai trò: "Admin" 6.Nhập tên. 7. Nhập số điện thoại. 8.Nhập địa chỉ 9.Nhấn "Xác nhận" | Hiển thị lỗi "mật khẩu phải dài 8-32 ký tự". | Pass |
| TC\_ACC\_004 | Add Account with Mismatched Passwords | Kiểm tra lỗi khi mật khẩu và xác nhận mật khẩu không khớp | Đã đăng nhập với quyền admin | 1.Nhấn "Thêm tài khoản". 2.Nhập Email. 3.Nhập mật khẩu. 4: Nhập lại mật khẩu 5.Chọn vai trò: "Admin" 6.Nhập tên. 7. Nhập số điện thoại. 8.Nhập địa chỉ 9.Nhấn "Xác nhận" | Hiển thị lỗi "Nhắc lại mật khẩu không đúng". | Pass |
| TC\_ACC\_005 | Add Account with Invalid Email | Kiểm tra lỗi khi thêm tài khoản với định dạng email không hợp lệ | Đã đăng nhập với quyền admin | 1.Nhấn "Thêm tài khoản". 2.Nhập Email. 3.Nhập mật khẩu. 4: Nhập lại mật khẩu 5.Chọn vai trò: "Admin" 6.Nhập tên. 7. Nhập số điện thoại. 8.Nhập địa chỉ 9.Nhấn "Xác nhận" | Hiển thị lỗi "Phải nhập đúng địa chỉ email". | Pass |
| TC\_ACC\_006 | Add Existing Email Account | Kiểm tra lỗi khi thêm tài khoản đã tồn tại | - Đã đăng nhập với quyền admin - Đã có tài khoản "admin@gmail.com" | 1.Nhấn "Thêm tài khoản". 2.Nhập Email. 3.Nhập mật khẩu. 4: Nhập lại mật khẩu 5.Chọn vai trò: "Admin" 6.Nhập tên. 7. Nhập số điện thoại. 8.Nhập địa chỉ 9.Nhấn "Xác nhận" | Hiển thị lỗi "Email đã được đăng ký". | Pass |
| TC\_ACC\_007 | Delete Account | Kiểm tra xóa tài khoản thành công | - Đã đăng nhập với quyền admin - Đã có tài khoản " nguyenvana@gmail.com" | 1. Chọn tài khoản "nguyenvana@gmail.com". 2.Nhấn "Xóa". 3.Xác nhận xóa. | Hiển thị thông báo "Xóa tài khoản thành công". Tài khoản không còn trong danh sách. | Pass |
| TC\_ACC\_008 | Access Account Page without Login | Kiểm tra thông báo lỗi khi truy cập trang quản lý tài khoản mà chưa đăng nhập | Đóng trình duyệt, chưa đăng nhập | Truy cập vào đường dẫn "/accounts". | Hiển thị thông báo "Truy cập bị từ chối" | Fail |

Bảng Kịch bản kiểm thử chức năng quản lý tài khoản

### 4.2.11. Kiểm thử chức năng trang chủ

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TC\_ID** | **Testcase Name** | **Description** | **Pre-condition** | **Test Steps** | **Expected Result** | **Pass/Fail** |
| TC\_HP\_001 | Verify Homepage Loads Successfully | Kiểm tra trang chủ tải lên thành công | Truy cập website | 1.Mở trình duyệt. 2.Nhập URL trang chủ. 3.Quan sát quá trình tải trang. | Trang chủ hiển thị đầy đủ nội dung và hình ảnh. | Pass |
| TC\_HP\_002 | Verify Product List Display | Kiểm tra danh sách sản phẩm có hiển thị đúng | Truy cập trang chủ | 1.Truy cập trang chủ. 2.Kiểm tra mục "Danh sách laptop mới nhất". | Danh sách sản phẩm hiển thị đầy đủ hình ảnh, tên, giá. | Pass |
| TC\_HP\_003 | Verify Price Format | Kiểm tra giá sản phẩm hiển thị đúng định dạng | Truy cập trang chủ | 1.Truy cập trang chủ. 2.Kiểm tra định dạng giá | Giá sản phẩm hiển thị đúng định dạng và rõ ràng. | Pass |
| TC\_HP\_004 | Verify "Add to Cart" Buttons | Kiểm tra nút "Giỏ hàng" có hoạt động đúng không | Truy cập trang chủ | 1.Truy cập trang chủ. 2.Nhấn vào nút "Giỏ hàng" dưới mỗi sản phẩm. | Nút "Giỏ hàng" hoạt động, thêm sản phẩm vào giỏ hàng. | Pass |

Bảng Kịch bản kiểm thử chức năng trang chủ

## 4.3. Thực thi các ca kiểm thử trên Selenium Webdriver

### 4.3.1. Kiểm thử chức năng đăng ký tài khoản



**Kết quả:** Sau khi thực hiện kiểm thử 6 test case cho tính năng đăng ký tài khoản, kết quả như sau:

**6 test case Passed** — các chức năng hoạt động đúng như mong đợi:

* **TC\_REG\_001: Đăng ký với dữ liệu hợp lệ** — hệ thống chuyển đến trang chủ và hiển thị thông báo "Đăng ký thành công".
* **TC\_REG\_002: Đăng ký với các trường để trống** — hệ thống hiển thị thông báo lỗi tương ứng cho từng trường bắt buộc.
* **TC\_REG\_003: Đăng ký với email đã tồn tại** — hệ thống cảnh báo "Email đã được sử dụng".
* **TC\_REG\_004: Đăng ký với email không hợp lệ** — hệ thống thống báo Email không hợp lệ
* **TC\_REG\_005: Đăng ký và các nhận mật khẩu không khớp** — trình duyệt hiển thị thông báo "Nhắc lại mật khẩu không chính xác".
* **TC\_REG\_006: Đăng ký khi để trống trường email** — trình duyệt hiển thị thông báo "Please fill out this field".

### 4.3.2. Kiểm thử chức năng đăng nhập

****

**Kết quả:** Sau khi thực hiện kiểm thử 10 test case cho tính năng đăng nhập, kết quả như sau:

**10 test case Passed** — tất cả chức năng hoạt động đúng như mong đợi:

* **TC\_LG\_001: Đăng nhập với tài khoản Admin hợp lệ** — hệ thống chuyển hướng đến trang quản trị thành công.
* **TC\_LG\_002: Đăng nhập với tài khoản User hợp lệ** — hệ thống chuyển hướng đến trang chủ dành cho khách hàng.
* **TC\_LG\_003: Đăng nhập lại sau khi thất bại** — hệ thống hiển thị thông báo lỗi rồi cho phép đăng nhập lại thành công.
* **TC\_LG\_004: Đăng nhập với tài khoản không hợp lệ** — hiển thị thông báo lỗi "Tên đăng nhập hoặc mật khẩu không đúng".
* **TC\_LG\_005: Đăng nhập với tài khoản không tồn tại** — hệ thống thông báo "Tài khoản không tồn tại".
* **TC\_LG\_006: Đăng nhập không điền tên đăng nhập** — hiển thị thông báo "Vui lòng nhập tên đăng nhập".
* **TC\_LG\_007: Đăng nhập không điền mật khẩu** — hiển thị thông báo "Vui lòng nhập mật khẩu".
* **TC\_LG\_008: Đăng nhập không điền tên đăng nhập và mật khẩu** — thông báo "Vui lòng nhập thông tin đăng nhập".
* **TC\_LG\_009: Đăng nhập với ký tự đặc biệt** — hệ thống từ chối và hiển thị thông báo lỗi.
* **TC\_LG\_010: Đăng nhập với ký tự số** — hệ thống từ chối và hiển thị thông báo lỗi.

**Kết luận:** Các trường hợp kiểm thử cho tính năng đăng nhập đều hoạt động theo đúng yêu cầu.

### 4.3.3. Kiểm thử chức năng đăng xuất

****

Chức năng đăng xuất hoạt động chính xác theo yêu cầu:

* Admin đăng xuất thành công.
* Hệ thống chuyển hướng về trang đăng nhập.

Trạng thái: Passed

### 4.3.4. Kiểm thử chức năng thêm sản phẩm vào giỏ hàng



**Kết quả:** Sau khi thực hiện kiểm thử 7 test case cho tính năng giỏ hàng, kết quả như sau:

**6 test case Passed** — các chức năng hoạt động đúng như mong đợi:

* **TC\_CART\_001: Thêm một sản phẩm vào giỏ hàng** — sản phẩm hiển thị đúng thông tin trong giỏ hàng.
* **TC\_CART\_002: Thêm nhiều sản phẩm vào giỏ hàng** — tất cả sản phẩm được thêm vào giỏ hàng và hiển thị chính xác.
* **TC\_CART\_003: Thêm cùng một sản phẩm nhiều lần** — số lượng sản phẩm trong giỏ hàng tăng tương ứng.
* **TC\_CART\_004: Thêm sản phẩm hết hàng** — hệ thống cảnh báo "Sản phẩm đã hết hàng".
* **TC\_CART\_005: Tăng/giảm số lượng sản phẩm** — tổng giá trị trong giỏ hàng cập nhật chính xác theo số lượng.
* **TC\_CART\_006: Xóa sản phẩm khỏi giỏ hàng** — sản phẩm bị loại bỏ thành công, giỏ hàng cập nhật lại.

**1 test case Failed**:

* **TC\_CART\_007: Thêm sản phẩm vào giỏ hàng khi chưa đăng nhập** — hiển thị thông báo “Vui lòng đăng nhập”

**+** Kết quả mong đợi: hệ thống hiển thị thông báo "Vui lòng đăng nhập để thêm sản phẩm vào giỏ hàng".

**+** Kết quả thực tế: sản phẩm vẫn được thêm vào giỏ hàng, không có cảnh báo đăng nhập.

### 4.3.5. Kiểm thử chức năng thanh toán



**Kết quả:** Sau khi thực hiện kiểm thử 5 test case cho tính năng thanh toán, kết quả như sau:

**3 test case Passed** — các chức năng hoạt động đúng như mong đợi:

* **TC\_CHECKOUT\_001: Thanh toán khi giỏ hàng trống** — nút "Thanh toán" ẩn đi, người dùng không thể tiếp tục.
* **TC\_CHECKOUT\_002: Thanh toán với để trống trường Họ tên** — hệ thống hiển thị thông báo lỗi "Họ tên không được để trống".
* **TC\_CHECKOUT\_005: Thanh toán với thông tin hợp lệ** — hệ thống xử lý thanh toán và chuyển đến trang xác nhận đơn hàng thành công.

**2 test case Failed** — lỗi phát sinh trong quá trình kiểm thử:

* **TC\_CHECKOUT\_003: Thanh toán với số điện thoại không hợp lệ** — Kết quả mong đợi: hiển thị thông báo "Số điện thoại không hợp lệ".

**+** Kết quả thực tế: hệ thống vẫn tiếp tục quá trình thanh toán mà không kiểm tra tính hợp lệ của số điện thoại.

* **TC\_CHECKOUT\_004: Nhập địa chỉ dưới 5 ký tự** — Kết quả mong đợi: hiển thị thông báo "Địa chỉ quá ngắn".

**+** Kết quả thực tế: hệ thống cho phép tiếp tục thanh toán mà không kiểm tra độ dài địa chỉ.

### 4.3.6. Kiểm thử chức năng tìm kiếm

****

**Kết quả:**

**10 test case Passed** — các chức năng hoạt động đúng như mong đợi:

* **TC\_SR\_001: Search with Valid Keyword** — Hệ thống trả về kết quả phù hợp khi tìm kiếm với từ khóa hợp lệ.
* **TC\_SR\_002: Search with Invalid Keyword** — Hệ thống hiển thị thông báo "Không có kết quả" khi tìm kiếm với từ khóa không hợp lệ.
* **TC\_SR\_004: Search with Special Characters** — Hệ thống xử lý ký tự đặc biệt và thông báo không tìm thấy kết quả.
* **TC\_SR\_005: Search with Leading/Trailing Spaces** — Hệ thống tự động loại bỏ khoảng trắng dư thừa và tìm kiếm chính xác.
* **TC\_SR\_006: Case-Insensitive Search** — Tìm kiếm không phân biệt hoa thường hoạt động tốt, kết quả hiển thị chính xác.
* **TC\_SR\_007: Filter Products by Price Range** — Bộ lọc theo mức giá hoạt động đúng, sản phẩm phù hợp được hiển thị.
* **TC\_SR\_008: Filter by Category** — Lọc sản phẩm theo danh mục chính xác.
* **TC\_SR\_009: Filter by Brand** — Bộ lọc theo thương hiệu hoạt động đúng, kết quả hiển thị theo đúng thương hiệu được chọn.
* **TC\_SR\_010: Combine Multiple Filters** — Hệ thống xử lý đúng khi kết hợp nhiều bộ lọc (giá, danh mục, thương hiệu).
* **TC\_SR\_011: No Matching Results** — Khi không có sản phẩm phù hợp, hệ thống hiển thị thông báo "Không có kết quả".

**1 test case Failed:**

* **TC\_SR\_003: Search with Empty Keyword**

+ Mô tả: Kiểm tra tìm kiếm khi không nhập từ khóa.

+ Kết quả mong đợi: Hệ thống hiển thị thông báo lỗi "Vui lòng nhập từ khóa"

+ Kết quả thực tế: Hệ thống không hiển thị bất kỳ thông báo lỗi nào và vẫn thực hiện tìm kiếm, trả về toàn bộ sản phẩm.

+ Ảnh hưởng: Người dùng có thể nhầm lẫn rằng tìm kiếm hoạt động bình thường, trong khi thực tế là hệ thống đang bỏ qua logic kiểm tra đầu vào.

### 4.3.7. Kiểm thử chức năng quản lý sản phẩm

****

**Kết quả:**

**9 test case Passed** — các chức năng hoạt động đúng như mong đợi:

* **TC01: Upload Product with Valid Data** — Hệ thống cho phép tải lên sản phẩm mới với đầy đủ dữ liệu hợp lệ, sản phẩm hiển thị chính xác trong danh sách.
* **TC03: Upload Product with Missing Fields** — Hệ thống hiển thị thông báo lỗi khi người dùng bỏ trống các trường bắt buộc.
* **TC04: Update Product Name** — Hệ thống cho phép cập nhật tên sản phẩm và lưu thay đổi thành công.
* **TC05: Update Without Any Change** — Hệ thống xử lý đúng khi người dùng nhấn "Submit" mà không thay đổi bất kỳ thông tin nào.
* **TC06: Delete Product** — Hệ thống xóa sản phẩm thành công và xác nhận sản phẩm đã biến mất khỏi danh sách.
* **TC07: Search Product by Category** — Người dùng có thể tìm kiếm sản phẩm theo danh mục, và sản phẩm hiển thị chính xác.
* **TC08: Search Product by Code** — Hệ thống cho phép tìm kiếm sản phẩm theo mã, và sản phẩm được hiển thị đúng.
* **TC09: Search Non-existent Product** — Khi tìm kiếm sản phẩm không tồn tại, hệ thống phản hồi chính xác rằng không tìm thấy sản phẩm.
* **TC10: View Product Details** — Hệ thống hiển thị đầy đủ thông tin chi tiết sản phẩm khi người dùng nhấn "Details".

**1 test case Failed:**

* **TC02: Upload Product without Images**

+ Mô tả: Hệ thống cho phép tải lên sản phẩm nhưng không kèm hình ảnh.

+ Kết quả mong đợi: Hệ thống hiển thị thông báo lỗi yêu cầu hình ảnh (tùy vào quy định của hệ thống).

+ Kết quả thực tế: Hệ thống báo "Upload thành công", nhưng sản phẩm không xuất hiện trong danh sách.

+ Ảnh hưởng: Người dùng nghĩ sản phẩm đã được thêm vào, nhưng thực tế sản phẩm không tồn tại trong hệ thống, gây nhầm lẫn về dữ liệu.

### 4.3.8. Kiểm thử chức năng quản lý danh mục

****

**Kết quả:** Sau khi thực hiện kiểm thử 6 test case cho tính năng đăng ký tài khoản, kết quả như sau:

**5 test case Passed** — các chức năng hoạt động đúng như mong đợi:

* **TC\_CAT\_001:** Thêm danh mục hợp lệ — Hệ thống cho phép thêm danh mục mới khi nhập tên danh mục hợp lệ và hiển thị thông báo "Thêm danh mục thành công".
* **TC\_CAT\_002: Thêm danh mục nhưng để trống tên** — Hệ thống hiển thị thông báo lỗi "Tên danh mục không được để trống".
* **TC\_CAT\_004: Cập nhật danh mục hợp lệ** — Hệ thống cập nhật tên danh mục mới và thông báo "Cập nhật danh mục thành công".
* **TC\_CAT\_005: Cập nhật danh mục với tên trống** — Hệ thống hiển thị thông báo lỗi "Tên danh mục không được để trống".
* **TC\_CAT\_006: Xóa danh mục** — Danh mục được xóa thành công và hệ thống hiển thị "Xóa danh mục thành công".

**1 test case Failed:**

* **TC\_CAT\_003: Thêm danh mục trùng tên** — Khi nhập tên danh mục đã tồn tại, hệ thống không hiển thị thông báo lỗi "Tên danh mục đã tồn tại". Thay vào đó, danh mục trùng lặp vẫn được thêm vào.

### 4.3.9. Kiểm thử chức năng quản lý nhãn hiệu



**Kết quả:** Sau khi thực hiện kiểm thử 6 test case cho tính năng quản lý danh mục, kết quả như sau:

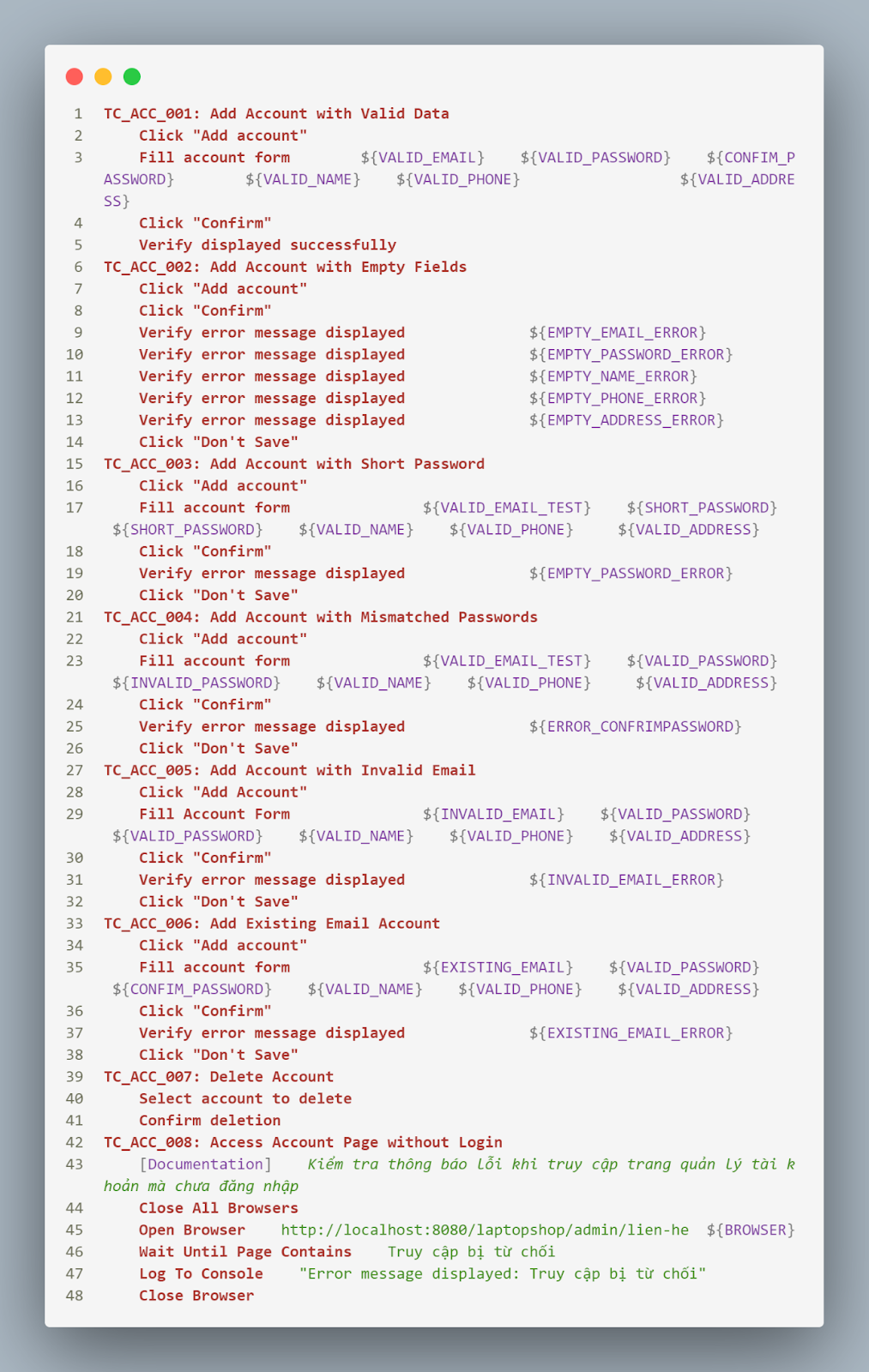
**5 test case Passed** — các chức năng hoạt động đúng như mong đợi:

* **TC\_BRAND\_001: Thêm thương hiệu hợp lệ** — Hệ thống cho phép thêm thương hiệu mới khi nhập tên hợp lệ và hiển thị thông báo "Thêm thương hiệu thành công".
* **TC\_BRAND\_002: Thêm thương hiệu nhưng để trống tên** — Hệ thống hiển thị thông báo lỗi "Tên thương hiệu không được để trống".
* **TC\_BRAND\_004: Cập nhật thương hiệu hợp lệ** — Hệ thống cập nhật tên thương hiệu mới và thông báo "Cập nhật thương hiệu thành công".
* **TC\_BRAND\_005: Cập nhật thương hiệu với tên trống** — Hệ thống hiển thị thông báo lỗi "Tên thương hiệu không được để trống".
* **TC\_BRAND\_006: Xóa thương hiệu** — Thương hiệu được xóa thành công và hệ thống hiển thị "Xóa thương hiệu thành công".

**1 test case Failed:**

* **TC\_BRAND\_003: Thêm thương hiệu trùng tên** — Khi nhập tên thương hiệu đã tồn tại, hệ thống không hiển thị thông báo lỗi "Tên thương hiệu đã tồn tại". Thay vào đó, thương hiệu trùng lặp vẫn được thêm vào.

### 4.3.10. Kiểm thử chức năng quản lý tài khoản



**Kết quả**

**7 test case Passed** — tất cả các chức năng hoạt động đúng như mong đợi:

* **TC\_ACC\_001: Thêm tài khoản hợp lệ** — Hệ thống cho phép thêm tài khoản mới khi nhập đầy đủ và hợp lệ các trường thông tin, hiển thị thông báo "Thêm tài khoản thành công".
* **TC\_ACC\_002: Thêm tài khoản với các trường trống** — Hệ thống kiểm tra và hiển thị các thông báo lỗi phù hợp cho từng trường trống, bao gồm:

"Email không được để trống"

"Mật khẩu không được để trống"

"Họ tên không được để trống"

"Số điện thoại không được để trống"

"Địa chỉ không được để trống"

* **TC\_ACC\_003: Thêm tài khoản với mật khẩu ngắn** — Khi nhập mật khẩu không đủ độ dài yêu cầu, hệ thống hiển thị thông báo lỗi "Mật khẩu quá ngắn".
* **TC\_ACC\_004: Thêm tài khoản với mật khẩu không khớp** — Khi nhập mật khẩu và mật khẩu xác nhận không khớp, hệ thống báo lỗi "Mật khẩu xác nhận không trùng khớp".
* **TC\_ACC\_005: Thêm tài khoản với email không hợp lệ** — Hệ thống kiểm tra định dạng email và hiển thị thông báo lỗi "Email không hợp lệ".
* **TC\_ACC\_006: Thêm tài khoản với email đã tồn tại** — Khi nhập email đã được sử dụng, hệ thống hiển thị thông báo lỗi "Email đã tồn tại".
* **TC\_ACC\_007: Xóa tài khoản** — Tài khoản được xóa thành công và hệ thống hiển thị thông báo "Xóa tài khoản thành công".

**1 testcase Failed**:

* **TC\_ACC\_008: Truy cập trang quản lý tài khoản mà chưa đăng nhập** — Khi truy cập trực tiếp vào đường dẫn "/accounts" mà không đăng nhập, hệ thống không hiển thị thông báo lỗi "Truy cập bị từ chối". Thay vào đó, người dùng vẫn có thể truy cập trang quản lý tài khoản.

### 4.3.11. Kiểm thử giao diện trang chủ



**Kết quả:** Sau khi thực hiện kiểm thử 5 test case cho tính năng trang chủ, kết quả như sau:

**5 test case Passed** — giao diện hiển thị đúng như mong đợi:

* **TC\_HP\_001: Kiểm tra trang chủ tải lên thành công** — tiêu đề trang chủ hiển thị chính xác, trang tải hoàn tất.
* **TC\_HP\_002: Kiểm tra danh sách sản phẩm hiển thị đúng** — danh sách sản phẩm được tải và hiển thị đầy đủ trên giao diện.
* **TC\_HP\_003: Kiểm tra định dạng giá sản phẩm** — giá sản phẩm hiển thị đúng định dạng, không có lỗi về font hoặc ký tự lạ.
* **TC\_HP\_004: Kiểm tra nút "Giỏ hàng" có hoạt động** — khi nhấn vào nút "Add to Cart", sản phẩm được thêm vào giỏ hàng thành công.
* **TC\_HP\_005: Kiểm tra menu "Danh mục"** — menu điều hướng hoạt động chính xác, chuyển đúng đến trang danh mục tương ứng.

# KẾT LUẬN

Kiểm thử phần mềm nói chung và kiểm thử ứng dụng trên nền Web nói riêng là một vấn đề hết sức quan trọng đối với các tổ chức phát triển phần mềm hiện này. Trong quá trình thực hiện đề tài do kinh nghiệm còn hạn chế nên một số phần nhóm em nghiên cứu chưa được sâu.

Kiểm thử chức năng được thực hiện nhằm đảm bảo các tính năng cốt lõi của hệ thống hoạt động chính xác theo yêu cầu. Nhóm chúng em đã kiểm tra các quy trình quan trọng như quản lý tài khoản, quản lý sản phẩm và danh mục, giỏ hàng, thanh toán và tìm kiếm sản phẩm. Kết quả cho thấy việc áp dụng kiểm thử tự động giúp phát hiện lỗi nhanh chóng, đảm bảo các thao tác như đăng ký, đăng nhập, thêm sản phẩm vào giỏ hàng, thanh toán diễn ra đúng với yêu cầu của hệ thống.

Bên cạnh đó, kiểm thử giao diện người dùng đóng vai trò quan trọng trong việc đánh giá trải nghiệm người dùng trên nhiều trình duyệt và thiết bị khác nhau. Nhóm đã kiểm tra khả năng hiển thị trên trình duyệt phổ biến như Chrome đảm bảo bố cục giao diện không bị lỗi, các nút quan trọng như "Thanh toán", "Xác nhận đơn hàng" luôn hiển thị rõ ràng và không bị trùng lặp trên các kích thước màn hình khác nhau.

Qua quá trình kiểm thử, nhóm chúng em nhận thấy rằng mặc dù kiểm thử tự động giúp tiết kiệm thời gian và giảm thiểu sai sót, nhưng vẫn cần kết hợp với kiểm thử thủ công trong một số trường hợp nhất định, đặc biệt khi kiểm tra các yếu tố liên quan đến trực quan và trải nghiệm người dùng.

Kết quả của đề tài không chỉ giúp nâng cao chất lượng website bán hàng mà còn tạo tiền đề cho việc áp dụng kiểm thử tự động vào các dự án phần mềm khác. Trong tương lai, có thể mở rộng nghiên cứu về kiểm thử hiệu năng (Performance Testing) và kiểm thử bảo mật (Security Testing) để đảm bảo hệ thống hoạt động ổn định và an toàn hơn.

# TÀI LIỆU THAM KHẢO

[1]. Co-Well, *Kiểm thử phần mềm là gì & Quy trình thực hiện*, 05/09/2021, <https://co-well.vn/nhat-ky-cong-nghe/kiem-thu-phan-mem-la-gi-quy-trinh-thuc-hien/>

[2]. Nhóm Sinh viên Trường Đại học Kiến Trúc Hà Nội, Báo cáo Nghiên cứu khoa học: *Nghiên cứu kiểm thử tự động với công cụ Selenium ứng dụng Kiểm thử hệ thống Website Trường Đại học Kiến Trúc Hà Nội*, 10/02/2023,

<https://www.studocu.vn/vn/document/truong-dai-hoc-kien-truc-ha-noi/nghien-cuu-khoa-hoc-kien-truc-ha-noi/nhom-3-bao-cao-kiem-thu-tu-dong-selenium-thanh-trung-1-2-1-2-1/81135476>

[3]. Viettel IDC, *Kiểm thử phần mềm là gì? Quy trình Kiểm thử phần mềm*, 16/09/2024, <https://viettelidc.com.vn/tin-tuc/kiem-thu-phan-mem-la-gi>

**BẢNG PHÂN CÔNG CÔNG VIỆC**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **MSV** | **Họ tên** | **Công việc** | **Mức độ đóng góp (%)** |
| 1 | 2155010067 | Nguyễn Hữu Thành Đạt | Nghiên cứu, xây dựng kịch bản kiểm thử | 30% |
| 2 | 2155100092 | Trần Thanh Hiền | Nghiên cứu, sử dụng công cụ kiểm thử, thực thi kịch bản kiểm thử | 40% |
| 3 | 2155010132 | Trần Quang Hưng | Nghiên cứu, xây dựng kịch bản kiểm thử | 30% |