

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT HƯNG YÊN

HOÀNG THỊ MẾN

**KIỂM THỬ TỰ ĐỘNG CHỨC NĂNG CHO WEBSITE
VINAGIAY.VN SỬ DỤNG KATALON STUDIO**

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC

HƯNG YÊN - 2021

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT HƯNG YÊN

HOÀNG THỊ MẾN

KIỂM THỬ TỰ ĐỘNG CHỨC NĂNG CHO WEBSITE
VINAGIAY.VN SỬ DỤNG KATALON STUDIO

NGÀNH: CÔNG NGHỆ THÔNG TIN

CHUYÊN NGÀNH: KỸ THUẬT PHẦN MỀM

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC

NGƯỜI HƯỚNG DẪN: ĐỖ THU TRANG

HƯNG YÊN - 2021

NHẬN XÉT

Nhận xét của giảng viên hướng dẫn:

[illegible]

GIẢNG VIÊN HƯỚNG DẪN

(Ký và ghi rõ họ tên)

LỜI CAM ĐOAN

Em xin cam đoan đồ án tốt nghiệp “Kiểm thử tự động chức năng cho website vinagiay.vn sử dụng Katalon Studio” là công trình nghiên cứu của bản thân. Những phần sử dụng tài liệu tham khảo trong đồ án đã được nêu rõ trong phần tài liệu tham khảo. Các số liệu, kết quả trình bày trong đồ án là hoàn toàn trung thực, nếu sai em xin chịu hoàn toàn trách nhiệm và chịu mọi kỷ luật của Bộ môn và Nhà trường đề ra.

Hưng Yên, ngày 20 tháng 06 năm 2021.

Sinh viên

.....

MỤC LỤC

DANH SÁCH BẢNG BIỂU	8
DANH SÁCH HÌNH VẼ	9
CHƯƠNG 1: MỞ ĐẦU	11
1.1 Lý do chọn đề án	11
1.2 Mục tiêu của đề án.....	12
1.2.1 Mục tiêu tổng quát	12
1.2.2 Mục tiêu cụ thể.....	13
1.3 Giới hạn và phạm vi của đề án	14
1.3.1 Đối tượng nghiên cứu	14
1.3.2 Phạm vi nghiên cứu.....	14
1.4 Nội dung thực hiện	14
1.5 Phương pháp tiếp cận.....	14
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT	15
2.1 Tổng quan về kiểm thử phần mềm	15
2.1.1 Kiểm thử phần mềm là gì?.....	15
2.1.2 Các loại kiểm thử phần mềm	19
2.2 Kiểm thử tự động phần mềm.....	25
2.2.1 Kiểm thử tự động là gì?	25
2.2.2 Ưu điểm của kiểm thử tự động:	26
2.2.3 Quy trình kiểm thử tự động.....	29
2.2.4 Một số công cụ hỗ trợ kiểm thử tự động.....	30
2.2.5 Các hướng kiểm thử tự động.....	33

2.3 Công cụ Katalon Studio	38
CHƯƠNG 3: ĐẶC TẢ HỆ THỐNG PHẦN MỀM.....	45
3.1 Giới thiệu về hệ thống phần mềm.....	45
3.2 Các yêu cầu chức năng	46
3.2.1 Chức năng đăng nhập	46
3.2.2 Chức năng đăng ký.....	48
3.2.3 Màn hình chức năng tìm kiếm.....	49
3.2.4 Quản lý giỏ hàng	51
3.3 Các yêu cầu phi chức năng	52
CHƯƠNG 4: TRIỂN KHAI KIỂM THỬ TỰ ĐỘNG.....	53
4.1 Thiết kế các yêu cầu kiểm thử	53
4.2 Xây dựng ca kiểm thử.....	57
4.2.1 Tạo tài khoản.....	57
4.2.2 Đăng nhập	57
4.2.3 Thông tin cá nhân.....	58
4.2.4 Đổi mật khẩu.....	59
4.2.5 Giỏ hàng	59
4.3 Xây dựng dữ liệu kiểm thử	60
4.3.1 Data đăng nhập.....	60
4.3.2 Data tạo tài khoản	61
4.3.3 Data quản lý thông tin cá nhân	65
4.3.4 Data đổi mật khẩu	69
4.3.5 Data Tìm kiếm	69
4.3.6 Data Giỏ hàng	70

4.4	Xây dựng kịch bản kiểm thử.....	71
4.4.1	Test scripts Đăng nhập.....	71
4.4.2	Test scripts Tạo tài khoản	71
4.4.3	Test scripts Thông tin cá nhân.....	72
4.4.4	Test scripts Giỏ hàng.....	74
4.5	Thực thi và báo cáo kiểm thử	76
CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN		79
Kết quả đạt được.....		79
Hạn chế của đề tài.....		79
Hướng phát triển của đề tài		80

DANH SÁCH BẢNG BIỂU

Bảng 1-1: Danh mục ví dụ phạm vi nghiên cứu.....	14
Bảng 2-1: So sánh tính năng Katalon Studio với công cụ khác	41
Bảng 2-2: So sánh ưu nhược điểm của Katalon Studio với một số công cụ	43
Bảng 4-1: Test case tạo tài khoản.....	57
Bảng 4-2: Test case đăng nhập.....	57
Bảng 4-3: Test case Thông tin cá nhân	58
Bảng 4-4: Test case Đổi mật khẩu.....	59
Bảng 4-5: Test case Giỏ hàng.....	59
Bảng 4-6: Data đăng nhập	60
Bảng 4-7: Data tạo tài khoản	61
Bảng 4-8: Data quản lý thông tin cá nhân	65
Bảng 4-9: Data đổi mật khẩu.....	69
Bảng 4-10: Data Tìm kiếm.....	69
Bảng 4-11: Data Giỏ hàng.....	70

DANH SÁCH HÌNH VẼ

Hình 2-1: Kiểm thử phần mềm là gì	15
Hình 2-2: Tester có vai trò quan trọng đối với doanh nghiệp	18
Hình 2-3: Điều kiện để trở thành 1 tester chuyên nghiệp.	24
Hình 2-4: Tại sao cần sử dụng kiểm thử tự động	26
Hình 2-5: Ưu điểm và nhược điểm của Automation test	28
Hình 2-6: Quy trình kiểm thử tự động.....	29
Hình 2-7: Cấu trúc Selenium.....	31
Hình 3-1: Màn hình “Đăng nhập”	46
Hình 3-2: Màn hình sau khi [Đăng nhập] thành công.....	47
Hình 3-3: Màn hình “Đăng ký”	48
Hình 3-4: Màn hình sau khi [Đăng ký] thành công.....	49
Hình 3-5: Màn hình [Tìm kiếm].....	49
Hình 3-6: Ảnh minh họa phân trang.....	50
Hình 3-7: Màn hình chức năng quản lý giỏ hàng.....	51
Hình 4-1: Test design chức năng Đăng ký	53
Hình 4-2: Test design chức năng Đăng nhập	54
Hình 4-3: Test design Giỏ hàng	55
Hình 4-4: Test design chức năng Quản lý tài khoản	56

DANH SÁCH CÁC THUẬT NGỮ

Từ viết tắt	Từ đầy đủ	Giải thích
SRS	Software Requirement Specification	Đặc tả yêu cầu phần mềm
ANSI/IEEE	Tiêu chuẩn ANSI/IEEE 1059	Tiêu chuẩn ANSI/IEEE 1059 về kiểm thử phần mềm
QA	Quality Assurance	Đảm bảo chất lượng

CHƯƠNG 1: MỞ ĐẦU

1.1 Lý do chọn đề án

Ngày nay, các sản phẩm phần mềm đã có mặt trong tất cả các lĩnh vực đời sống kinh tế, xã hội của con người. Sản xuất phần mềm trở thành một ngành công nghiệp mang lại giá trị lớn, có tốc độ phát triển nhanh như vũ bão. Đặc biệt với sự bùng nổ của Internet đã thúc đẩy nhu cầu cộng tác làm việc qua mạng và sử dụng dịch vụ trực tuyến. Các ứng dụng Web trở thành xu hướng tất yếu của ngành công nghiệp phần mềm. Nó giúp người dùng cộng tác với nhau trên những hệ thống linh hoạt không phụ thuộc vào nền tảng cố định mọi lúc mọi nơi. Trong quy trình phát triển phần mềm, việc đảm bảo chất lượng phần mềm đang là một bài toán khó tiêu tốn 50% công sức và chi phí của các doanh nghiệp phần mềm. Kiểm thử là khâu không thể thiếu trong đảm bảo chất lượng phần mềm. Tùy vào từng giai đoạn phát triển phần mềm mà có kiểm thử đơn vị, kiểm thử tích hợp, kiểm thử hệ thống, kiểm thử chấp nhận.v.v. Kiểm thử bằng tay tốn kém về thời gian, công sức, không kinh tế và dễ gây nhầm lẫn cho người thực hiện kiểm thử. Vì vậy, nhiều công cụ kiểm thử tự động ra đời nhằm giải quyết được những vấn đề trên. Do đó, đề tài nghiên cứu mong muốn đưa ra giải pháp tự động hóa kiểm thử nhằm giảm thiểu chi phí kiểm thử cả về thời gian, tiền bạc, công sức của con người và làm giảm sự nhầm lẫn cho kiểm thử viên nhưng vẫn đảm bảo được chất lượng của sản phẩm phần mềm. Với sự phát triển mạnh mẽ của các phần mềm cũng như kiểm thử phần mềm hiện nay, có rất nhiều công cụ hỗ trợ cho việc kiểm thử tự động, mỗi công cụ đều có một hoặc một số phương pháp luận khác nhau. Nhưng điều đó không đồng nghĩa với việc lựa chọn bất kỳ công cụ nào cũng tốt, hay cứ áp dụng kiểm thử tự động là có thể tiết kiệm chi phí và đảm bảo dự án sẽ thành công.

Vì vậy, việc xây dựng được 1 framework để áp dụng vào kiểm thử tự động là cần thiết, phù hợp với điều kiện thực tế, trong phạm vi đề án này tập trung nghiên cứu xây dựng framework kiểm thử tự động hướng lai và áp dụng vào kiểm thử một ứng dụng cụ thể.

1.2 Mục tiêu của đồ án

1.2.1 Mục tiêu tổng quát

Trên cơ sở về thực tiễn của kiểm thử trong việc đảm bảo chất lượng phần mềm, tầm quan trọng của kiểm thử chấp nhận, tôi đã nghiên cứu, tìm hiểu và chọn đề tài: “Kiểm thử tự động chức năng cho website vinagiay.vn sử dụng Katalon Studio”. Mục đích là vừa nghiên cứu lý thuyết, vừa xây dựng kịch bản kiểm thử đối với một ứng dụng thực tế. Sau khi hoàn thành, hiểu nắm được công cụ Katalon Studio với các thư viện chuẩn. Chương trình kiểm thử tự động được triển khai cho trang Website vinagiay.vn

Áp dụng được những kiến thức học trên lớp môn chuyên đề và đồ án trước về kiểm thử và đảm bảo chất lượng phần mềm. Biết cách sử dụng kiểm thử tự động cho các chức năng, kiểm thử hiệu năng, bảo mật cho ứng dụng. Sử dụng thành thạo các công cụ hỗ trợ kiểm thử như: Katalon Studio

Đồ án tập trung nghiên cứu vai trò, phương pháp kiểm thử hướng dữ liệu cũng như ứng dụng trong việc kiểm thử ứng dụng Website. Sử dụng phương pháp kiểm thử hộp đen, áp dụng thành thạo các kỹ thuật kiểm thử hộp đen: Kỹ thuật phân tích giá trị biên và phân vùng tương đương, kỹ thuật bảng quyết định, kỹ thuật chuyển đổi trạng thái. Nắm được các bước và hoàn thành đầy đủ việc kiểm thử. Từ đó, thực hiện phân tích, xây dựng các test case cho các chức năng của một website; sử dụng ngôn ngữ lập trình Java tạo các test case, test script để kiểm tra khả năng thực hiện đúng và đủ của từng chức năng. Kết quả kiểm thử góp phần đánh giá được chất lượng của Website đó.

Giúp tiết kiệm chi phí, thời gian phát triển về sau cho đơn vị phát triển phần mềm triển khai phần mềm, nâng cao hiệu quả làm việc, nâng cao kiến thức, tránh gây nhầm lẫn cho những người trực tiếp làm công việc kiểm thử phần mềm.

1.2.2 Mục tiêu cụ thể

Mục tiêu cụ thể của đề tài là Nghiên cứu và xây dựng một chương trình kiểm thử tự động hướng dữ liệu, hướng từ khóa... thường là điều gì đó hoặc hoạt động nào đó cụ thể, rõ ràng mà người học sẽ hoàn thành theo kế hoạch đã đề ra. Mục tiêu có thể đo lường hay định lượng được và là cơ sở cho việc đánh giá kế hoạch thực hiện đã đưa ra. Xác định các mục tiêu cụ thể đề tài cần đạt được để nêu bật mục đích tổng quát. Mục tiêu của đề tài cụ thể chỉ ra một cách hệ thống các khía cạnh khác nhau của vấn đề thiết kế giải pháp và triển khai, là những mục tiêu mà ta phải đạt được khi kết thúc quá trình thực hiện đề tài. Giúp tiết kiệm chi phí phát triển về sau cho đơn vị phát triển phần mềm triển khai phần mềm, nâng cao hiệu quả làm việc, nâng cao kiến thức, tránh gây nhầm chán cho những người trực tiếp làm công việc kiểm thử phần mềm.

- Mục tiêu chung:
 - Đưa ra một cái nhìn tổng quan trong việc nghiên cứu, lựa chọn công cụ kiểm thử tự động áp dụng trong từng trường hợp kiểm thử phần mềm tự động hiện nay.
- Mục tiêu cụ thể:
 - Nghiên cứu công cụ kiểm thử tự động phần mềm trong ngành công nghiệp phần mềm: Tập trung nghiên cứu công cụ Katalon Studio. Xây dựng bộ thư viện, tài liệu kiểm thử hỗ trợ việc tự động hóa kiểm thử phần mềm.

1.3 Giới hạn và phạm vi của đề án

1.3.1 Đối tượng nghiên cứu

- Công cụ Katalon Studio và áp dụng kiểm thử tự động cho website
- Xây dựng từ khóa, data...dựa trên ngôn ngữ java
- Khách thể nghiên cứu: Đề tài sử dụng website bán giấy để làm đối tượng thực hiện kiểm thử tự động

1.3.2 Phạm vi nghiên cứu

Phạm vi không gian: Websie vinagiay.vn

Phạm vi thời gian: 01/2021 -06/2021

Bảng 1-1: Danh mục ví dụ phạm vi nghiên cứu

Tiêu đề 1	Mô tả
Phạm vi không gian	Websie vinagiay.vn
Phạm vi thời gian	01/2021 -06/2021

1.4 Nội dung thực hiện

- Xây dựng đặc tả của trang web, test plan, test design, testcase.
- Xây dựng từ khóa, bộ dữ liệu.
- Áp dụng công cụ Katalon Studio vào kiểm thử tự động.

1.5 Phương pháp tiếp cận

- Phương pháp phân tích tài liệu
- Phương pháp phân tích mẫu
- Phương pháp thực nghiệm

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1 Tổng quan về kiểm thử phần mềm

2.1.1 Kiểm thử phần mềm là gì?

Kiểm thử là một quá trình đánh giá một hệ thống hay là các thành phần của nó với mục đích là xác định xem nó có thỏa mãn những yêu cầu được đưa ra hay không. Hiểu một cách đơn giản, kiểm thử - test là chạy một chương trình để xác nhận bất kỳ lỗi hỏng, lỗi sai hay những yêu cầu bị bỏ quên, những yêu cầu không đúng so với yêu cầu thực tế đề ra. Mục đích của kiểm thử phần mềm là tìm ra lỗi chưa được phát hiện, tìm một cách sớm nhất và bảo đảm rằng lỗi sẽ được sửa.

Mục tiêu của kiểm thử phần mềm là thiết kế tài liệu kiểm thử một cách có hệ thống và thực hiện nó sao cho có hiệu quả, nhưng tiết kiệm được thời gian, công sức và chi phí.

Theo tiêu chuẩn ANSI/IEEE 1059, kiểm thử có thể được định nghĩa như một quá trình của việc phân tích các thành phần của phần mềm để dò tìm sự khác biệt giữa phần mềm thực tế đang tồn tại và những điều kiện được yêu cầu - requirement (đó là thiếu sót - defect, sai sót - error, lỗi - bug). Từ đó đánh giá được chất lượng của sản phẩm phần mềm.



Hình 2-1: Kiểm thử phần mềm là gì

Ai sẽ là người test?

Điều này phụ thuộc vào quy trình và các bên liên quan đến dự án. Trong ngành công nghiệp phần mềm, những công ty lớn có một team chuyên chịu trách nhiệm việc đánh giá phần mềm đã phát triển với những yêu cầu được chỉ định từ trước. Hơn thế nữa, nhân viên phát triển phần mềm - developer cũng tiến hành kiểm thử sản phẩm, và việc đó được gọi là unit testing. Trong hầu hết các trường hợp, người kiểm thử - tester có thể là:

- Software Tester – Nhân viên kiểm thử phần mềm
- Software Developer – Nhân viên phát triển phần mềm
- Leader hoặc Manager của dự án
- User - Người sử dụng cuối cùng

Các công ty khác nhau sẽ có các quy định về tên gọi của nhân viên kiểm thử dựa trên kinh nghiệm và kiến thức về kiểm thử phần mềm, như là: software tester – nhân viên kiểm thử phần mềm, software quality assurance engineer - kỹ sư đảm bảo chất lượng phần, QA analyst – nhân viên phân tích chất lượng phần mềm ...

Thời điểm bắt đầu test

Việc kiểm thử sớm sẽ giúp giảm chi phí và thời gian để xây dựng lại và sửa lỗi để bàn giao sản phẩm phần mềm. Tuy nhiên trong một vòng đời của phần mềm, việc kiểm thử nên được bắt đầu từ khi có những yêu cầu từ phía khách hàng và được kéo dài đến cho đến khi triển khai phần mềm. Thời điểm bắt đầu kiểm thử còn phụ thuộc vào mô hình phát triển phần mềm đang được sử dụng. Ví dụ như: trong mô hình thác nước – waterfall model, kiểm thử chính thức được thực hiện ở giai đoạn kiểm thử - testing phase. Nhưng ở trong mô hình gia tăng – incremental model, kiểm thử được thực hiện ở cuối mỗi chu kỳ con và kiểm thử cho toàn bộ sản phẩm được thực hiện ở giai đoạn cuối khi hoàn thiện sản phẩm.

Việc kiểm thử được thể hiện theo nhiều dạng công việc khác nhau ở các giai đoạn khác trong suốt vòng đời phát triển phần mềm:

Trong quá trình tập hợp yêu cầu – requirement gathering phase, phân tích và xác minh yêu cầu cũng được coi là kiểm thử - test requirement.

Trong giai đoạn thiết kế - design phase, kiểm tra lại thiết kế với mục đích cải thiện thiết kế cũng được tính là kiểm thử.

Trong giai đoạn phát triển phần mềm – Implement phase, kiểm thử được thực hiện bởi lập trình viên – unit testing cũng được tính là kiểm thử.

Khi nào thì dừng việc test:

Rất khó để xác định khi nào thì dừng kiểm thử, vì test là quá trình ko bao giờ kết thúc và không ai có thể yêu cầu phần mềm được test 100%. Các khía cạnh dưới đây có thể được cân nhắc cho việc dừng lại quy trình test:

- Hạn chót phát triển phần mềm – deadline.
- Đã thực hiện hết các testcase được đề ra.
- Hoàn thiện các chức năng và toàn bộ code đã đảm bảo được các yêu cầu đề ra.
- Tỷ lệ bug phải trong giới hạn mong muốn.
- Không có những bug nghiêm trọng
- Quyết định của quản lý dự án

Vai trò của Test là gì?

Tester là những người có khả năng nắm vững mọi công cụ cũng như kỹ thuật và kiến thức liên quan đến quá trình phát triển phần mềm. Trong giai đoạn phát triển cho sản phẩm thì Tester đóng góp vai trò kiểm tra, đánh giá để đưa ra các thông số kỹ thuật phù hợp nhất.

Tester thường tham gia và chịu hoàn toàn trách nhiệm trong việc xác định điều kiện thử nghiệm và tạo ra được các thiết kế thử nghiệm. Một số trường hợp thực hiện thử nghiệm và đưa ra các đặc tả cho quy trình cũng như dữ liệu thử nghiệm.

Ngoài ra, mỗi Tester đều sẽ thực hiện quá trình tự động hóa cho các bài kiểm tra và đảm bảo cho việc thiết lập môi trường thử nghiệm, quản trị hệ thống và những nhân viên quản lý mạng làm việc đó.

Trong quá trình kiểm thử, thông thường các Tester đều được yêu cầu ghi lại những kết quả đánh giá và kiểm tra khi tìm thấy lỗi. Họ sẽ thực hiện các giám sát và đảm bảo việc thu thập các số liệu liên quan đến hiệu suất công việc.

Những yêu cầu cần phải có đối với một Tester:

Là Tester cần phải hiểu rõ được các sản phẩm cần phải kiểm tra. Cần nắm rõ các lập kế hoạch các chiến lược thử nghiệm để tìm ra được những vấn đề cần giải quyết và thực hiện các thử nghiệm. Thực hiện phân tích rõ ràng các ưu điểm, nhược điểm hoặc giải quyết dễ dàng hơn các rủi ro liên quan đến từng thành phần cũng như giao diện của sản phẩm.



Hình 2-2: Tester có vai trò quan trọng đối với doanh nghiệp

Cần hiểu rõ về code để có thể dễ dàng check cũng như kiểm tra lại chúng. Cần phải có các kỹ năng làm việc với các tập lệnh cũng như công cụ tự động hóa. Cần thường xuyên cập nhật cho mọi khía cạnh kỹ thuật của các cơ sở hạ tầng sản phẩm. Chú ý phân tích, ghi nhận được các vấn đề và có trách nhiệm cung cấp các phản hồi thích hợp nhất.

Lợi ích của Tester là gì?

- Các tester phải đảm bảo được chất lượng cho các sản phẩm phần mềm được sản xuất. Việc thực hiện Tester sẽ đảm bảo cho các sản phẩm luôn chất lượng trước khi cung cấp đến tay người tiêu dùng.
- Tester là người giữ vai trò loại bỏ các rủi ro cũng như các vấn đề có thể xảy ra đối với sản phẩm phần mềm.
- Việc thực hiện kiểm tra đối với dự án CNTT bất kỳ sẽ giúp doanh nghiệp tiết kiệm được số lượng tiền dài hạn. Một số trường hợp nếu phát hiện được lỗi trong các giai đoạn testing trước đó thì chi phí sửa chữa thường sẽ thấp hơn.
- Tester có nhiệm vụ loại bỏ được mọi rủi ro để đảm bảo cho chất lượng sản phẩm nhằm mang lại sự hài lòng triệt để cho khách hàng.

2.1.2 Các loại kiểm thử phần mềm

Testing of function (Functional testing)

Kiểm thử chức năng là kiểm tra xem hệ thống có hoạt động theo đúng theo các yêu cầu nghiệp vụ không? Kiểm thử chức năng được thực hiện ở tất cả các mức kiểm thử.

Kiểm thử chức năng có thể thực hiện theo 2 quan điểm: requirements-based và business - process – based.

Requirements - based: Sử dụng các đặc tả yêu cầu của hệ thống làm cơ sở để design test. Một cách tốt để bắt đầu là sử dụng bảng nội dung của đặc tả yêu cầu như một danh sách các mục kiểm thử và không kiểm thử. Chúng ta nên xét độ ưu tiên của

yêu cầu dựa trên các tiêu chí rủi ro và sử dụng độ ưu tiên để kiểm thử. Điều này sẽ đảm bảo những phần quan trọng nhất sẽ được kiểm thử.

Business - process - based: sử dụng các kiến thức về quy trình nghiệp vụ. Quy trình nghiệp vụ mô tả các kịch bản liên quan đến nghiệp vụ hằng ngày của hệ thống.

Ví dụ: Một hệ thống quản lý nhân sự và bảng lương có thể có quy trình nghiệp vụ như sau: người nào đó gia nhập vào công ty, họ được trả lương thường xuyên và cuối cùng họ rời khỏi công ty. Các use case bắt nguồn từ hướng đối tượng, nhưng ngày nay phổ biến nhiều là dựa trên vòng đời phát triển phần mềm. Chúng ta có thể lấy quy trình nghiệp vụ là điểm bắt đầu. Các use case là một cơ sở rất hữu ích tạo ra các trường hợp kiểm thử từ góc nhìn về nghiệp vụ.

Kiểm thử chức năng bao gồm 5 bước:

- Xác định các chức năng mà phần mềm mong muốn sẽ thực hiện.
- Tạo các dữ liệu đầu vào dựa trên các tài liệu đặc tả kỹ thuật của các chức năng.
- Xác định các kết quả đầu ra dựa trên các tài liệu đặc tả kỹ thuật của các chức năng.
- Thực hiện các trường hợp kiểm thử.
- So sánh kết quả thực tế và kết quả mong muốn.

Các loại kiểm thử chức năng bao gồm:

- Kiểm thử đơn vị (Unit Testing)
- Smoke Testing
- Sanity Testing
- Kiểm thử giao diện (Interface Testing)
- Kiểm thử tích hợp (Integration Testing)
- Kiểm thử hệ thống (System Testing)
- Kiểm thử hồi quy (Regression Testing)

- Kiểm thử chấp nhận (Acceptance Testing)

Testing of software product characteristics (Non - Functional testing)

Kiểm thử phi chức năng là các đặc tính chất lượng của hệ thống sẽ được kiểm tra. Chúng ta quan tâm đến việc mọi thứ hoạt động tốt không? Hay nhanh như thế nào? Chúng ta sẽ kiểm tra những thứ cần phải đo như thời gian phản hồi, hay bao nhiêu người có thể đăng nhập cùng một lúc? Kiểm thử phi chức năng cũng giống như kiểm thử chức năng được thực hiện ở tất cả các cấp độ kiểm thử.

Kiểm thử phi chức năng bao gồm:

- Kiểm thử hiệu năng (Performance testing)
- Kiểm thử khả năng chịu tải (Load testing)
- Kiểm thử áp lực (Stress testing)
- Kiểm thử tính khả dụng (Usability testing)
- Kiểm thử bảo trì (Maintainability testing)
- Kiểm thử độ tin cậy (Reliability testing)
- Kiểm thử tính tương thích (Portability testing)

Các đặc điểm và các đặc điểm phụ tương ứng:

- Chức năng (Functionality) gồm 5 đặc điểm phụ: sự phù hợp, chính xác, bảo mật, khả năng tương tác và tuân thủ.
- Độ tin cậy (Reliability) gồm 4 đặc điểm phụ: độ bền, khả năng chịu lỗi, khả năng phục hồi và tuân thủ.
- Khả năng sử dụng (Usability) gồm 5 đặc điểm phụ: dễ hiểu, khả năng học hỏi, khả năng hoạt động, sự thu hút và tính tuân thủ.
- Tính hiệu quả (Efficiency) gồm 3 đặc điểm phụ: thời gian (hiệu suất), sử dụng tài nguyên và tuân thủ.

- Khả năng bảo trì (Maintainability) gồm 5 đặc điểm phụ: khả năng phân tích, khả năng thay đổi, tính ổn định, khả năng kiểm tra và tuân thủ.
- Tính tương thích (Portability) gồm 5 đặc điểm phụ: khả năng thích ứng, khả năng cài đặt, cùng tồn tại, khả năng thay thế và tuân thủ.

Testing of software structure/architecture (Structural testing)

Kiểm thử cấu trúc thường được gọi là "hộp trắng" hoặc "hộp thủy tinh" vì chúng quan tâm đến những gì đang xảy ra bên trong hộp.

Kiểm thử cấu trúc thường được sử dụng như một cách đo lường của kiểm thử thông qua độ bao phủ của một tập hợp các yếu tố cấu trúc hoặc các mục bao phủ.

Nó có thể xảy ra ở bất kỳ mức độ kiểm thử nào chủ yếu ở kiểm thử thành phần, tích hợp.

Ở cấp độ thành phần, và mức thấp hơn trong kiểm thử tích hợp thành phần có hỗ trợ công cụ tốt để đo mức độ bao phủ của mã. Các công cụ đo lường độ bao phủ đánh giá tỉ lệ phần trăm thực thi đã được thực hiện bởi một bộ kiểm thử. Nếu độ bao phủ không phải là 100% thì các kiểm thử bổ sung có thể cần phải được viết và chạy để bao phủ những phần chưa được thực hiện.

Các kỹ thuật được sử dụng để kiểm tra cấu trúc là kỹ thuật kiểm thử hộp trắng, các mô hình luồng điều khiển thường sử dụng để hỗ trợ kiểm thử cấu trúc.

Testing related to changes (Confirmation and regression testing)

Confirmation testing (Kiểm thử xác nhận)

Khi kiểm thử bị lỗi, và chúng ta xác định nguyên nhân lỗi là do lỗi phần mềm, lỗi được báo cáo, khi một phiên bản mới của phần mềm đã sửa lỗi. Trong trường hợp này chúng ta cần thực hiện kiểm tra một lần nữa để xác định rằng lỗi thực sự đã được sửa.

Khi thực hiện kiểm tra xác nhận điều quan trọng là phải đảm bảo rằng thử nghiệm được thực hiện chính xác giống như lần đầu tiên, sử dụng cùng một đầu vào, dữ liệu và môi trường. Nếu bây giờ đúng có nghĩa là phần mềm chính xác. Chúng ta

biết rằng ít nhất một phần của phần mềm là chính xác, nhưng điều đó là không đủ. Sửa lỗi có thể gây ra một lỗi khác trong phần mềm. Cách phát hiện các bất lợi ngoài ý muốn của việc sửa lỗi là thực hiện kiểm thử hồi quy.

Regression testing (Kiểm thử hồi quy)

Giống như kiểm thử xác nhận kiểm thử hồi quy liên quan đến việc thực hiện các trường hợp kiểm thử đã được thực hiện trước đó. Sự khác biệt là đối với kiểm thử hồi quy, các trường hợp kiểm thử có thể đúng ở lần cuối cùng chúng được thực thi.

Mục đích của kiểm thử hồi quy để xác minh rằng các sửa đổi trong phần mềm hoặc môi trường không gây ra bất lợi ngoài ý muốn và hệ thống vẫn đáp ứng các yêu cầu của nó.

Bộ kiểm thử hồi quy hoặc gói kiểm tra hồi quy là một tập hợp các trường hợp kiểm thử được sử dụng đặc biệt để kiểm tra hồi quy. Chúng được thiết kế để thực hiện hầu hết các chức năng trong một hệ thống nhưng không chi tiết bất kỳ chức năng nào. Tất cả các trường hợp trong bộ kiểm thử hồi quy sẽ được thực thi mỗi khi một phiên bản mới của phần mềm được phát hành và điều này làm cho chúng trở lên lý tưởng cho tự động hóa.

Kiểm thử hồi quy được thực hiện khi phần mềm thay đổi, do sửa lỗi, chức năng mới. Nó cũng là một ý tưởng tốt để thực thi chúng khi một vài khía cạnh của môi trường thay đổi.

Tester cần những kiến thức gì?

Để trở thành một Tester thì bạn cần phải có kiến thức căn bản về máy tính, hệ điều hành, cơ sở dữ liệu database, lập trình, mạng máy tính. Tiếp theo, cần phải hiểu rõ những kỹ năng testing cơ bản, tư duy logic, thuật toán lập trình. Từ đó, rút ngắn quá trình lý luận, tìm tòi và xử lý những lỗi fix bug nhanh nhất có thể. Cuối cùng, một Tester cần phải có khả năng tiếng anh tốt bởi các tài liệu chủ yếu là tiếng anh. Việc đọc hiểu và dịch được các loại tài liệu từ nước ngoài sẽ giúp ích rất nhiều cho công việc của bạn sau này. Vậy là bạn cũng đã bắt đầu chuẩn bị những kiến thức cho

mình khi đọc tới đây rồi phải không? Ở phần tiếp theo của bài viết này, trước khi đi đến phần cuối đó chính là những kỹ năng cần thiết mà một tester cần phải có?

Cùng tham khảo một số kỹ năng quan trọng cần phải có để có thể trở thành những nhà tester chuyên nghiệp.



Hình 2-3: Điều kiện để trở thành 1 tester chuyên nghiệp.

Kỹ năng sáng tạo: Tìm kiếm những lỗi bug mà người khác chưa nghĩ ra và những lỗi chưa ai nghĩ tới sẽ giúp cho bạn nâng cao được tay nghề.

Kỹ năng phân tích: Trang bị cho mình kỹ năng phân tích nhanh chóng, nhạy bén và có độ chính xác cực cao. Chú ý, xem xét một vấn đề ở nhiều khía cạnh khác nhau, phân tích từng function để tìm ra từng lỗi nhỏ nhất.

Có tính cẩn thận, tỉ mỉ: Là tester bắt buộc không thể cầu thả, điều này sẽ rất nguy hiểm cho công việc. Có thể sẽ khiến bạn bỏ qua khá nhiều lỗi nghiêm trọng, điều này dẫn đến phần mềm khi đến tay người dùng sẽ không được hoàn thiện, nhận được những phản nản cũng như feedback xấu từ khách hàng.

Luôn luôn đổi mới: Kỹ năng này bạn cần phải cập nhật hàng ngày mới có thể nâng cao được tay nghề của bản thân.

Kỹ năng giao tiếp: Quá trình làm việc của một tester thường sẽ tiếp xúc với Developer liên tục và khả năng xung đột có thể xảy ra liên tục. Việc giao tiếp sẽ đảm bảo cho quá trình trình bày lý do lỗi được minh bạch và dễ hiểu hơn.

2.2 Kiểm thử tự động phần mềm

2.2.1 Kiểm thử tự động là gì?

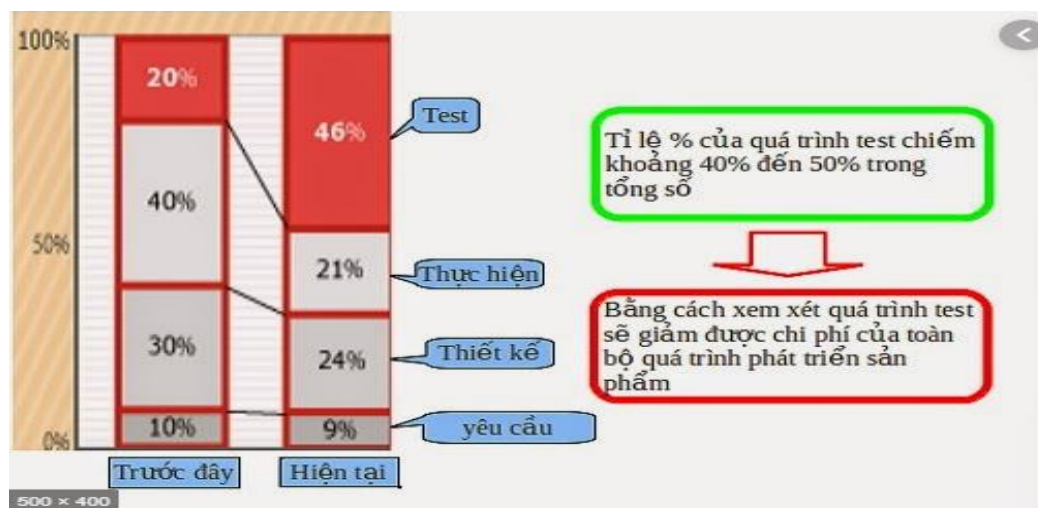
Kiểm thử tự động: Là xử lý một cách tự động các bước thực hiện các testcase, kiểm thử tự động bằng một công cụ nhằm rút ngắn thời gian kiểm thử.

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. Nó 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 là rất quan trọng cho phân phối liên tục và kiểm thử liên tục.

Tại sao cần sử dụng kiểm thử tự động ?

- Giảm bớt công sức và thời gian thực hiện quá trình kiểm thử
- Tăng độ tin cậy.
- Giảm sự nhầm lẫn cho con người
- Rèn luyện kỹ năng lập trình cho kiểm thử viên
- Giảm chi phí cho tổng quá trình kiểm thử



Hình 2-4: Tại sao cần sử dụng kiểm thử tự động

Kiểm thử tự động được thực hiện khi:

- Các trường hợp kiểm thử được thực hiện lặp đi lặp lại để đảm bảo tính năng của phần mềm/ sản phẩm.
- Thực hiện ở các trường hợp mà kiểm thử thủ công khó thực hiện. Các trường hợp kiểm thử cần tốn nhiều thời gian.
- Không đủ tài nguyên: Khi số lượng TestCase quá nhiều mà kiểm thử viên không thể hoàn tất trong thời gian cụ thể.
- Kiểm tra hồi quy: Nâng cấp phần mềm, kiểm tra lại các tính năng đã chạy tốt và những tính năng đã sửa. Tuy nhiên, việc này khó đảm bảo về mặt thời gian.
- Kiểm tra khả năng vận hành phần mềm trong môi trường đặc biệt (Đo tốc độ trung bình xử lý một yêu cầu của Web server, xác định cấu hình máy thấp nhất mà phần mềm vẫn có thể hoạt động tốt).

2.2.2 Ưu điểm- nhược điểm của kiểm thử tự động:

Ưu điểm:

Giúp tiết kiệm tiền bạc và thời gian: nhất là trong giai đoạn bảo trì của các dự án lớn. Mỗi tuần chúng ta phải thực hiện regression test từ 1 đến 2 lần với số lượng test case rất lớn trong 1 đến 2 ngày. Điều này gần như không thể thực hiện bằng cách

thủ công, trong khi với kiểm thử tự động chúng ta hoàn toàn có thể với nguồn nhân lực vô cùng khiêm tốn.

Chính xác hơn: Nhờ độ ổn định cao, kiểm thử tự động có thể thực thi các test case với độ chính xác cao hơn.

Độ bao phủ cao: Như đã nói ở trên, khi sử dụng kiểm thử tự động, chúng ta có thể thực thi số lượng lớn test case trong một thời gian ngắn. Nên độ bao phủ của nó rất cao. Điều này giúp chúng ta tăng độ bao phủ trong giai đoạn regression test.

Hoàn thành các công việc mà con người không thể làm được: Nếu chúng ta muốn thực thi load test, performance test, thì kiểm thử tự động là cách duy nhất.

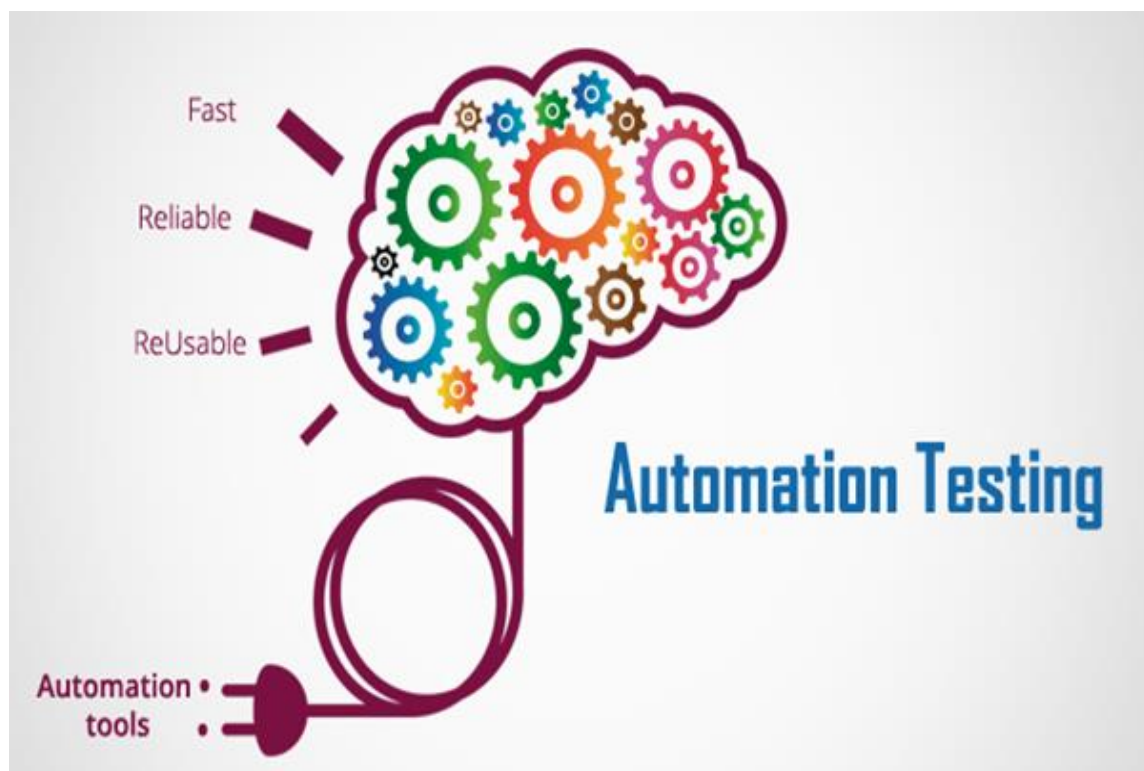
Độ tin cậy: Công cụ kiểm thử tự động mang tính ổn định cao hơn so với con người. Đặc biệt, trong những trường hợp nhiều test case nên độ tin cậy của kiểm thử sẽ đạt mức tối ưu hơn so với kiểm thử thủ công.

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 tính sử dụng.

Tốc độ cao: Với tốc độ kiểm thử nhanh hơn nhiều 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: Đặc biệt, 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.



Hình 2-5: Ưu điểm và nhược điểm của Automation test

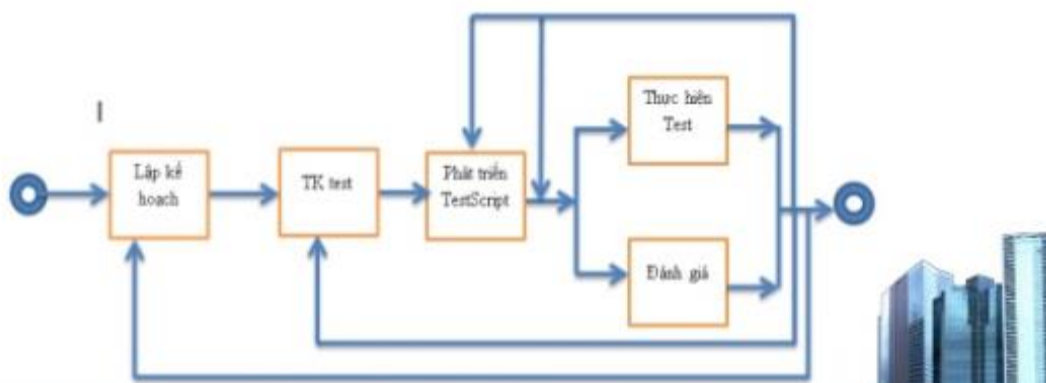
Nhược điểm:

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 nhưng đem lại nhiều hạn chế và nhân lực chưa đạt yêu cầu và có thể sử dụng một cách thành thạo cũng không khá khó.

2.2.3 Quy trình kiểm thử tự động



Hình 2-6: Quy trình kiểm thử tự động

Quy trình kiểm thử tự động phần mềm cũng giống như quy trình thực hiện kiểm thử thủ công chỉ khác ở chỗ kiểm thử tự động có hỗ trợ của công cụ ít hoặc nhiều như tạo test script (có thể bằng tay hoặc công cụ), công cụ hỗ trợ về ghi lại kết quả và lưu trữ kết quả trong máy tính. Quy trình này cũng gần tương tự với quy trình phát triển phần mềm, được thực hiện qua nhiều bước, được tiến hành rất sớm trong quy trình phát triển phần mềm và đội kiểm thử tiến hành gần như song song cùng đội phát triển phần mềm.

Để kiểm thử tự động thì công cụ là thành phần không thể thiếu trong tiến trình này, việc kiểm thử viên thành thạo các công cụ kiểm thử đảm bảo cho quy trình kiểm thử tự động được hiệu quả.

Một công cụ kiểm thử phần mềm tự động yêu cầu phải làm được những công việc sau:

- Hiểu các mã assembly được kiểm tra một cách tự động.
- Tiến hành các nhiệm vụ đơn giản và lặp đi lặp lại một cách tự động.
- Tạo ra các kịch bản và chạy các kịch bản trong những lô lệnh theo một lịch trình đã vạch ra.
- Kiểm tra giao diện của các đối tượng COM và các thành phần phần mềm khác với tập dữ liệu được thiết lập sẵn.

- Truy cập vào dữ liệu để xác minh lại các kết quả.

2.2.4 Một số công cụ hỗ trợ kiểm thử tự động

- Selenium

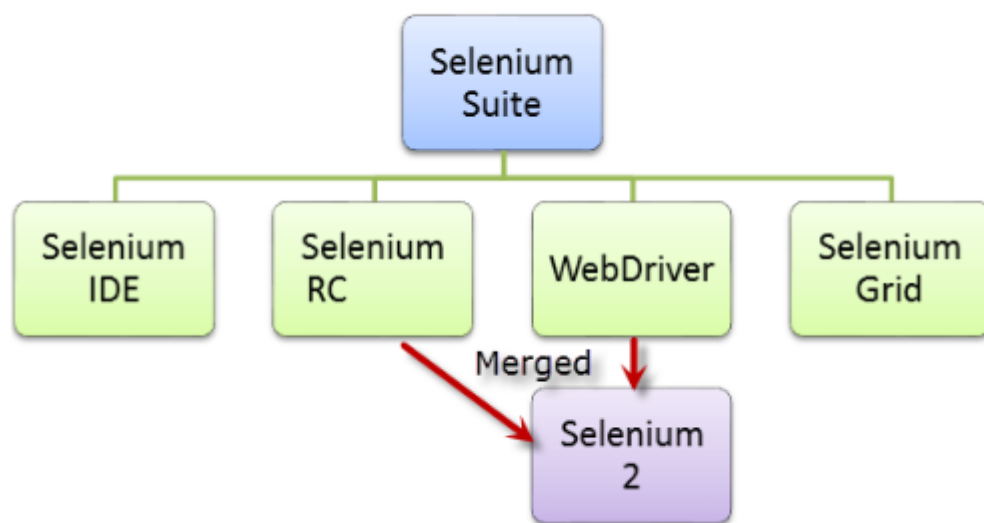
Giới thiệu về Selenium

Selenium (thường được viết tắt là SE) là một công cụ kiểm thử phần mềm tự động, được phát triển bởi ThoughtWorks từ năm 2004 với tên ban đầu là JavaScriptTestRunner. Đến năm 2007, tác giả Jason Huggins rời ThoughtWorks và gia nhập Selenium team, một phần của Google và phát triển thành Selenium như hiện nay. Selenium là một công cụ hỗ trợ kiểm tra tự động cho các ứng dụng chạy trên nền web. Selenium hỗ trợ kiểm tra hầu hết trên các trình duyệt phổ biến hiện nay như Firefox, Internet Explorer, Safari,...cũng như các hệ điều hành chủ yếu như Windows, Linux, Mac,...Selenium hỗ trợ một số lớn các ngôn ngữ lập trình như C#, Java, Perl, PHP, Python, Ruby...

Selenium có thể kết hợp thêm một số công cụ khác như Bromien, Junit nhưng với người dùng thông thường chỉ cần chạy tự động mà không cần cài thêm các công cụ hỗ trợ.

Các thành phần của Selenium

Selenium bao gồm một bộ các công cụ hỗ trợ kiểm tra tự động tính năng của ứng dụng web bao gồm: Selenium IDE, Selenium Remote Control (RC), Selenium Web Driver và Selenium Grid.



Hình 2-7: Cấu trúc Selenium

Selenium Integrated Development Environment - Selenium IDE: Đây là công cụ tích hợp trên trình duyệt và khá đơn giản với người dùng, hỗ trợ Record các thao tác để tạo thành các kịch bản kiểm thử và Playback trên các trình duyệt khác.

Selenium Remote Control - Selenium RC: Đối với các kịch bản thực tiễn, cần phải thực hiện các công việc kiểm tra phức tạp với các câu lệnh, Selenium RC hỗ trợ tối đa các công việc này.

Selenium Grid: Hỗ trợ thực hiện kiểm thử trên các trình duyệt song song mà không cần chỉnh sửa kịch bản.

Selenium WebDriver: Là bộ thư viện API nhằm giúp xây dựng ca kiểm thử trên nền tảng của ngôn ngữ lập trình

Một số ưu điểm và hạn chế của Selenium

Ưu điểm:

Mã nguồn mở: Đây là điểm mạnh nhất của Selenium khi so sánh với các test tool khác. Selenium không mất phí bản quyền hay thời hạn sử dụng.

Cộng đồng hỗ trợ khá mạnh mẽ, đặc biệt là nhà phát triển là Google. Đơn giản, dễ cài đặt, dễ làm việc

Selenium hỗ trợ nhiều ngôn ngữ lập trình và hỗ trợ chạy trên nhiều OS khác nhau. Dễ dàng điều chỉnh thông qua các plugin.

Hỗ trợ các tệp tin selenium user-extensions.js.

Tự động hoàn chỉnh cho tất cả các lệnh Selenium thường gặp.

Hạn chế:

Selenium chỉ hỗ trợ những ứng dụng web

Các ứng dụng trên mobile không thể sử dụng Selenium

Selenium là một tool miễn phí, do đó không có hỗ trợ từ nhà cung cấp mặc dù có thể tìm thấy giúp đỡ từ đông đảo cộng đồng sử dụng Selenium

Người sử dụng cần có kiến thức về ngôn ngữ lập trình

- Eclipse

Eclipse là 1 công cụ hỗ trợ lập trình mã nguồn mở được phát triển bởi IBM.

Eclipse như một môi trường phát triển Java tích hợp (IDE), với Eclipse chúng ta có thể mở rộng hơn mã nguồn bằng cách chèn thêm các plugins cho project (PDE- Plugin Development Environment). Mặc dù Eclipse được viết bằng ngôn ngữ lập trình Java, nhưng việc sử dụng nó không hạn chế chỉ cho ngôn ngữ Java. Ví dụ, Eclipse hỗ trợ sẵn hoặc có thể cài thêm các plugins để hỗ trợ cho các ngôn ngữ lập trình như C/C++ và COBOL. Ngoài ra, còn rất nhiều ngôn ngữ khác như PHP, Groovy...

Eclipse còn hỗ trợ cho lập trình viên code theo các mô hình phát triển như MVC, tạo thêm các lib hỗ trợ phát triển phần mềm.

Ưu điểm

- Tạo thuận lợi cho tích hợp liền mạch các công cụ bên trong mỗi một và xuyên qua nhiều kiểu nội dung và các nhà cung cấp công cụ khác nhau.
- Hỗ trợ việc xây dựng nhiều công cụ.
- Hỗ trợ một tập hợp không hạn chế các nhà cung cấp công cụ, bao gồm cả các nhà cung cấp phần mềm độc lập (ISV).

- Hỗ trợ các công cụ thao tác các kiểu nội dung bất kỳ (bao gồm cả HTML, Java, C, JSP, EJB, XML, và GIF).
- Hỗ trợ cả môi trường phát triển ứng dụng GUI lẫn không dựa trên GUI.
- Chạy trên nhiều hệ điều hành, bao gồm cả Windows và Linux.
- Lợi dụng tính phổ biến của ngôn ngữ Java để viết các công cụ.
- Do sử dụng SWT/JFace nên có lẽ load nhanh hơn

Nhược điểm

- Cài đặt phức tạp
- Tốn phần cứng máy Eclipse cũng khá tốn bộ nhớ máy để cài đặt nên hiện nay eclipse đã khắc phục bằng phiên bản Eclipse Luna, không cần cài đặt
- Tốn bộ nhớ máy: chạy eclipse cũng khá tốn bộ nhớ máy tính nên để sử dụng nó máy của bạn cũng cần có cấu hình tương đối cao
- Thiếu tính nhất quán vì có nhiều plugins quá

2.2.5 Các hướng kiểm thử tự động

Những Mô hình phổ biến hiện nay

- Tuyến tính
- Hướng module
- Hướng dữ liệu
- Hướng từ khóa
- Hỗn hợp

a) Tuyến tính – Linear Framework

Đặc điểm cơ bản

- Mọi thứ liên quan đến mã đều được định nghĩa bên trong phương thức kiểm thử

- Không quan tâm đến việc lãng phí và trùng lặp các câu lệnh
- Việc record/playback thường xuyên sinh ra mã tuyến tính
- Dễ dàng để bắt đầu
- Khó khăn trong việc chỉnh sửa

Mô hình này có thể được dùng trong dự án nhỏ, nơi mà không có quá nhiều màn hình giao diện cũng như chức năng. Mặc khác, chúng ta cũng hay dùng Mô hình này khi sử dụng một công cụ kiểm thử tự động lần đầu tiên. Thông qua việc record/playback và phát sinh mã, chúng ta có thể học được cách công cụ tương tác với ứng dụng như thế nào. Ngoài hai lý do trên, Mô hình này không được khuyến khích sử dụng trong kiểm thử tự động.

b) Hướng modul – Modularity

Đặc điểm cơ bản

- Các đối tượng được định nghĩa một lần và tái sử dụng trong các phương thức kiểm thử.
- Các phương thức nhỏ và có mục đích được tạo ra cho những chức năng riêng biệt
- Kịch bản kiểm thử tự động là một tập hợp các phương thức nhỏ và các đối tượng được định nghĩa từ trước
- Cho phép chúng ta dễ dàng viết các mã dễ dàng được chỉnh sửa

Mục đích chính của Mô hình này là việc chỉnh sửa dễ dàng. Nếu có bất kỳ thay đổi nào trên giao diện, chúng ta chỉ cần chỉnh sửa trong các phương thức và đối tượng. Mã kiểm thử chính của chúng ta vẫn hoạt động chính xác. Mô hình POM (Page Object Model) – thường được dùng với Selenium – là một dạng của việc ứng dụng Mô hình hướng modul. Toàn bộ trang web sẽ được chia nhỏ thành các trang. Các đối tượng UI của từng trang được định nghĩa bên trong từng lớp của trang đó. Nếu có bất kỳ thay đổi nào trên ứng dụng web, chúng ta chỉ cần chỉnh sửa lớp của trang đó,

những lớp của trang khác vẫn giữ nguyên. Kết quả cuối cùng chúng ta sẽ có những đoạn mã được bảo trì tốt hơn và dễ đọc hơn.

Điểm yếu của mô hình này là nó yêu cầu một mức độ kỹ năng lập trình và hiểu sâu về hướng đối tượng. Nếu bạn có nó, khuôn mẫu này được khuyến khích sử dụng.

c) Hướng dữ liệu – Data driven

Đặc điểm cơ bản

Dữ liệu kiểm thử (giá trị đầu vào và đầu ra) được tách khỏi mã nguồn và lưu trong một tập tin bên ngoài. Nó có thể là một tập tin CSV, một bảng Excel hay một cơ sở dữ liệu.

Khi mã kiểm thử thực thi, các giá trị này được lấy ra từ tập tin, chứa vào biến và thay thế các giá trị cứng (nếu có) trong mã nguồn.

Thực sự hữu ích khi mà cùng một kịch bản kiểm thử cần thực thi với nhiều dữ liệu đầu vào khác nhau.

Có vài ưu điểm khi áp dụng mô hình này. Tất cả các giá trị kiểm thử được lưu bên ngoài mã nguồn, do đó, bất kỳ thay đổi nào xảy ra trong quá trình phát triển ứng dụng, chúng ta chỉ cần thay đổi dữ liệu trong tập tin bên ngoài, và mã kiểm thử tự động của chúng ta vẫn được giữ nguyên.

Một ưu điểm khác là, khả năng sử dụng một kịch bản kiểm thử cho nhiều dữ liệu khác nhau. Ví dụ như, bạn đang làm một kịch bản đăng nhập hệ thống với 100 user. Bạn có thể viết 1 đoạn mã và một tập tin lưu trữ thông tin của 100 user. Sau đó, bạn chỉ cần thực thi 1 lần, và đi qua cả 100 bộ dữ liệu. Bạn dễ dàng phát hiện, với kiểu dữ liệu nào thì đoạn mã Fail. Đây cũng là một thế mạnh khi bạn đang làm kiểm thử phủ định – Negative Test.

d) Hướng từ khóa – Keyword driven

Đặc điểm cơ bản

- Cả dữ liệu và chức năng được định nghĩa bên ngoài mã nguồn
- Cần phát triển các từ khóa cho nhiều chức năng khác nhau

- Mã kiểm thử tự động đôi khi được lưu trữ ở một tập tin bên ngoài mã nguồn giống như mô hình hướng dữ liệu. Các bước của kịch bản kiểm thử được viết từng bước với định dạng bảng, nơi mà sử dụng các từ khóa và dữ liệu kiểm thử
- Mã nguồn chính sẽ đọc các bước trong định dạng bảng và thực thi các chức năng tương ứng
- Cho phép các kỹ sư kiểm thử thủ công, nhưng người không biết về lập trình, có thể là một phần, ở một mức độ, của nhóm kiểm thử tự động

Ưu điểm của mô hình hướng từ khóa

Mô hình này rất hữu dụng trong những trường hợp mà kịch bản kiểm thử có quá nhiều thay đổi. Nếu bất kỳ bước nào trong kịch bản kiểm thử bị thay đổi, chúng ta không cần phải chỉnh sửa mã nguồn. Chúng ta chỉ cần chỉnh sửa tập tin bên ngoài và như vậy, kịch bản kiểm thử tự động sẽ được chỉnh sửa theo.

Chúng ta định nghĩa toàn bộ kịch bản ở tập tin và đưa cho kỹ sư kiểm thử thủ công, họ sẽ thêm các đoạn văn bản (text) hoặc chỉnh sửa cái có sẵn. Bằng cách này, kỹ sư kiểm thử tự động cũng có thể trở thành một phần của nhóm kiểm thử tự động bởi vì họ không cần phải lập trình gì cả. Họ chỉ cần chỉnh sửa tập tin ở những vị trí cần thiết và kịch bản kiểm thử tự động sẽ được chỉnh sửa một cách tự động.

Một lợi ích khác của mô hình này là, kịch bản kiểm thử của bạn trở thành một công cụ độc lập. Bạn chỉ cần bảo trì kịch bản kiểm thử trong một tập tin và nếu bạn cần thay đổi công cụ kiểm thử tự động ở điểm nào đó, bạn có thể dễ dàng chỉnh sửa bằng cách viết lại cách đọc và thực thi tập tin với công cụ mới.

Mặc khác, khuyết điểm của mô hình này là, bạn cần phát triển các từ khóa cho các chức năng khác nhau. Trong một dự án lớn, có thể có rất nhiều từ khóa mà bạn cần phải nhớ và tổ chức nó hợp lý. Bản thân việc này có thể sẽ làm một công việc nặng nhọc cho quá trình phát triển kiểm thử tự động.

Ở vài trường hợp phức tạp, khi mà các đối tượng UI không thể được xác định dễ dàng, chúng ta phải sử dụng nhiều kỹ thuật khác nhau để xử lý, mô hình này không hữu dụng cho lắm.

Mô hình hướng từ khóa là một mô hình ưa thích của nhiều kỹ sư kiểm thử tự động. Robot Framework – công cụ kiểm thử tự động được phát triển bởi Google – là một công cụ phổ biến đi theo hướng từ khóa. Những công cụ đi theo hướng từ khóa này còn có Test Architect hay Katalon Studio

e) Hướng hỗn hợp – Hybrid

Đặc điểm cơ bản

- Là sự kết hợp của hai hoặc nhiều kỹ thuật ở trên, kế thừa thế mạnh và loại bỏ những điểm yếu của các mô hình khác.
- Mô hình này sử dụng cách tiếp cận theo hướng modul, kết hợp với hướng dữ liệu hoặc hướng từ khóa
- Mô hình này có thể dùng mã nguồn để xử lý những công việc đặc biệt mà quá khó để tạo ra với cách làm từ khóa

Một cách đơn giản, mô hình này sử dụng kết hợp nhiều kỹ thuật với nhau. Chúng ta có thể sử dụng hướng dữ liệu đồng thời với hướng modul. Trong vài trường hợp, chúng ta có thể dùng từ khóa song song với modul. Cơ bản, khi nào chúng ta sử dụng nhiều hơn một mô hình, đó là lúc chúng ta sử dụng hỗn hợp – Hybrid.

2.3 Công cụ Katalon Studio

Giới thiệu về Katalon Studio

Katalon Studio là một bộ công cụ toàn diện để kiểm thử tự động ứng dụng Web và Mobile. Công cụ này bao gồm một gói đầy đủ các tính năng mạnh mẽ giúp vượt qua các thách thức phổ biến trong kiểm thử tự động giao diện người dùng web, ví dụ: cửa sổ bật lên, iFrame và thời gian chờ. Giải pháp thân thiện và linh hoạt này giúp cho người kiểm tra tốt hơn, làm việc nhanh hơn và khởi chạy phần mềm chất lượng cao nhờ vào sự thông minh mà nó cung cấp cho toàn bộ quy trình kiểm thử tự động.

a) Các tính năng chính của Katalon Studio

Triển khai đơn giản (***Simple deployment***): gói triển khai duy nhất, gắn kết chứa mọi thứ cần để triển khai một công cụ kiểm thử tự động mạnh mẽ.

Cài đặt nhanh chóng và dễ dàng (***Quick & easy set-up***): không chỉ cung cấp cài đặt đơn giản, Katalon Studio còn giúp tester dễ dàng thiết lập môi trường. Người kiểm thử có thể chạy kịch bản kiểm thử đầu tiên của họ khá nhanh chóng bằng cách sử dụng các mẫu và tập lệnh kiểm thử dựng sẵn của nó, chẳng hạn như kho đối tượng và thư viện từ khóa.

Kết quả nhanh hơn và tốt hơn (***Faster & Better results***): các mẫu dựng sẵn với hướng dẫn rõ ràng giúp người kiểm tra nhanh chóng xây dựng và chạy các kịch bản kiểm thử tự động. Có thể thực hiện từng bước một tốc độ và hiệu quả, từ thiết lập dự án, tạo kiểm thử, thực hiện, tạo báo cáo và bảo trì.

Chế độ linh hoạt (***Flexible modes***): có thể sử dụng bản ghi và từ khóa để xây dựng bài kiểm thử tự động, trong khi có IDE đầy đủ để xây dựng tập lệnh nâng cao.

Dễ sử dụng (***Ease of use***): ngay cả thủ công với kinh nghiệm lập trình tối thiểu cũng có thể khai thác lợi ích của nó một cách dễ dàng.

Ứng dụng đa trình duyệt (***Cross-browser application***): Katalon Studio hỗ trợ nhiều nền tảng: Windows 32 và 64 (7, 8, và 10) và OS X 10.5+.

b) Làm việc với Katalon Studio

Katalon Studio là một giải pháp tự động hóa sâu sắc, thân thiện với người dùng, được đặc trưng bởi sự đơn giản và tốc độ. Nó rất hữu ích cho các nhóm và cá nhân người kiểm thử tự động dành ít nỗ lực nhất từ việc thiết lập một dự án mới đến thực hiện các kiểm thử và sau đó giám sát kết quả thực hiện. Mỗi quy trình công việc được cung cấp rất nhiều khả năng và tùy chỉnh để bảo trì dễ dàng và mở rộng dự án:

- Cấu trúc được xác định trước (Pre-defined structure): test cases, test suites, test objects, reports. Người kiểm tra không cần phải dành hàng giờ để xác định và duy trì chúng sau này.
- Từ khóa (Custom keywords) tùy chỉnh cung cấp tính linh hoạt trong việc thêm các từ khóa bổ sung để kiểm tra AUT hiệu quả cho các mục đích thử nghiệm cụ thể và phức tạp.
- Hỗ trợ các nhu cầu kiểm thử chính: web, mobile và API.
- Thực hiện nhiều bộ kiểm thử cùng một lúc với bộ sưu tập bộ kiểm thử.
- Mở rộng dòng CI hiện tại một cách dễ dàng với việc thực hiện chế độ bàn điều khiển mà không cần nỗ lực. Thực hiện dòng lệnh có thể được tạo ra nhanh chóng bằng cách sử dụng tính năng 'Generate Command Line for console mode'
- Giám sát kết quả thực hiện dễ dàng với chế độ xem bảng hoặc chế độ xem cây trong/ sau khi thực hiện.
- Báo cáo chi tiết bộ kiểm thử giảm thời gian phân tích kết quả. Có thể xuất nó sang các định dạng khác nhau như CSV, PDF, HTML và lưu trữ để sử dụng sau.

c) Quy trình làm việc của Katalon Studio

Điều hành một luồng công việc tuyến tính:

- Khởi tạo (INITIATE):

Các mẫu dự án tích hợp: bằng cách cung cấp các mẫu dựng sẵn để tổ chức các trường hợp kiểm thử, kho lưu trữ đối tượng và từ khóa, Katalon Studio giúp việc kiểm thử dễ dàng hơn cho người kiểm tra.

Nhiều khả năng: hỗ trợ đầy đủ kiểm thử cho Web, Android, iOS, và API trên tất cả các hệ điều hành.

Tích hợp trên công cụ không rắc rối: dễ dàng kết hợp với Jenkins, GIT và JIRA với các plug-in gốc.

- Triển khai (CREATE):

Tạo kiểm thử tự động: ghi lại các hành động và tạo các kịch bản tự động bằng các từ khóa tích hợp.

Tập lệnh Hi-end: cho phép xây dựng kịch bản kiểm thử nâng cao hoặc từ khóa tùy chỉnh dễ dàng và hiệu quả.

Thu thập thông tin đối tượng: máy ghi âm tiên tiến phát hiện các thuộc tính đối tượng một cách hiệu quả để tối đa hóa nhận dạng.

- Hoạt động (OPERATE):

Thực hiện kiểm thử (Test execution) mạnh mẽ: chạy các trường hợp kiểm thử hoặc bộ kiểm thử bằng cách sử dụng nhiều cấu hình và bộ dữ liệu.

Tính linh hoạt trong thực thi: cung cấp bảng điều khiển tích hợp CI với các tham số khác nhau để thực hiện từ xa. Chạy thử nghiệm trên nhiều trình duyệt và hệ điều hành OS cục bộ hoặc với Sauce Labs và BrowserStack.

Xử lý lỗi linh hoạt và thực hiện lại tự động: bao gồm các quy tắc thời gian chạy để tự động xử lý các luồng thực thi phức tạp.

- Báo cáo (REPORT):

Báo cáo có sẵn trong một số định dạng: với ghi nhật kí nâng cao, dữ liệu gỡ lỗi và ảnh chụp màn hình.

Báo cáo thực hiện Bespoke: được tích hợp với quy trình thông báo của bạn.

Nhật kí Selenium và Appium cải tiến: với các tính năng phân tích được cải tiến để cải thiện chiến lược tự động hóa.

- Bảo trì (MAINTAIN):

Bảo trì đối tượng kiểm thử thông minh: tự động cập nhật tất cả các trường hợp và bộ kiểm thử liên quan khi các đối tượng được thay đổi.

Tổ chức kiểm thử hiệu quả: cho phép dễ dàng quản lý và bảo trì các bài kiểm thử, dữ liệu và từ khóa.

d) So sánh các công cụ kiểm thử tự động

Katalon Studio ẩn tất cả các sự phức tạp về mặt kỹ thuật đằng sau hiện trường và cung cấp giao diện thân thiện với chế độ thủ công (người dùng có thể kéo-thả, chọn từ khóa và thử nghiệm các đối tượng để tạo thành các bước thử nghiệm) nhưng vẫn giữ vũ khí cần thiết cho những người dùng kỹ thuật mạnh mẽ sâu hơn vào mã hóa với chế độ kịch bản hỗ trợ đầy đủ các tiện ích phát triển như làm nổi bật cú pháp, đề xuất mã và gỡ lỗi.

Với Selenium, người dùng phải đi qua các hướng dẫn kỹ thuật cho API của họ và tích hợp chúng vào với một IDE và một ngôn ngữ lập trình ưa thích. Ngược lại, Katalon Studio được xây dựng như là một gói thống nhất bao gồm hầu như tất cả những thứ cần thiết như Java, Android SDK, trình điều khiển Web để điều khiển trình duyệt và yêu cầu phụ thuộc. Tất cả bạn phải làm là tải về và cài đặt

Bảng 2-1: So sánh tính năng Katalon Studio với công cụ khác

Tính năng	Selenium	QTP/UFT	Katalon Studio
Nền tảng phát triển kiểm thử	Cross-platform	Window	Cross-platform
Application under test	Web apps	Windows desktop, Web, Mobile apps, API/ Web services	Web, Mobile, API/ Web services

Ngôn ngữ kịch bản	Java, C#, Perl, Python, JavaScript, Ruby, PHP	VBScript	Java/ Groovy
Kỹ năng lập trình	Kỹ năng nâng cao cần thiết để tích hợp cho các công cụ khác nhau.	Không yêu cầu. Đề xuất cho các kịch bản nâng cao.	Không yêu cầu. Đề xuất cho các kịch bản nâng cao.
Dễ cài đặt và sử dụng	Yêu cầu cài đặt và tích hợp các công cụ khác nhau	Dễ dàng cài đặt và chạy	Dễ dàng cài đặt và chạy
Thời gian tạo kịch bản	Chậm	Nhanh	Nhanh Hỗ trợ Katalon Recorder
Lưu trữ và bảo trì đối tượng	XPath, bản đồ UI	Kho lưu trữ đối tượng tích hợp, phát hiện và điều chỉnh	Kho lưu trữ đối tượng tích hợp, XPath, nhận
Tính năng	Selenium	QTP/UFT	Katalon Studio
Kiểm thử dựa trên hình ảnh	Yêu cầu cài đặt thư viện bổ sung	Hỗ trợ tích hợp, nhận dạng đối tượng dựa trên hình ảnh	Hỗ trợ tích hợp
Kiểm thử phân tích	Không	Không	Katalon Analytics
Loại giấy phép	Mã nguồn mở (Apache 2.0)	Bản quyền	Phần mềm miễn phí
Phí	Miễn phí	Phí giấy phép và bảo trì	Miễn phí

Bảng so sánh trên chủ yếu tập trung vào các tính năng phổ biến của một công cụ kiểm thử tự động. Bảng sau sẽ trình bày một quan điểm khác bằng cách chọn và so sánh các điểm mạnh và hạn chế chính của các công cụ.

Bảng 2-2: So sánh ưu nhược điểm của Katalon Studio với một số công cụ

Công cụ	Điểm mạnh	Hạn chế
Katalon Studio	<ul style="list-style-type: none"> - Không có phí cấp phép và bảo trì cần thiết (dịch vụ hỗ trợ chuyên dụng có trả phí có sẵn nếu cần). - Tích hợp các framework và tính năng cần thiết để tạo và thực hiện các trường hợp kiểm thử nhanh. - Được xây dựng dựa trên Selenium nhưng loại bỏ nhu cầu về các kỹ năng lập trình nâng cao cần thiết cho Selenium. 	<ul style="list-style-type: none"> - Giải pháp mới nổi với một cộng đồng phát triển nhanh chóng. - Bộ tính năng vẫn đang phát triển. - Thiếu các lựa chọn cho các ngôn ngữ kịch bản: chỉ hỗ trợ Java/ Groovy.
Công cụ	Điểm mạnh	Hạn chế
Selenium	<p>Mã nguồn mở, không có phí cấp phép và bảo trì.</p> <p>Cộng đồng người dùng và phát triển lớn và tích cực để kịp với các công nghệ phần mềm.</p> <p>Mở để tích hợp các công cụ và các framework khác để tăng cường khả năng của nó.</p>	<p>Các nhóm kiểm thử cần phải có kỹ năng và kinh nghiệm lập trình tốt để thiết lập và tích hợp với Selenium với các công cụ và framework khác.</p> <p>Hỗ trợ chậm từ cộng đồng.</p>
QTP/UFT	<ul style="list-style-type: none"> - Các tính năng kiểm thử tự động toàn diện được tích hợp vào một hệ thống duy nhất. - Hỗ trợ người dùng chuyên dụng với một cộng đồng người dùng lớn được thành lập. - Chỉ yêu cầu các kỹ năng lập trình cơ bản để bắt đầu với việc tạo và thực thi kiểm thử. 	<ul style="list-style-type: none"> - Giải pháp tốn kém: giấy phép và phí bảo trì cao đáng kể. - Chi phí cao để nâng cấp và các module bổ sung. - Chỉ hỗ trợ VBScript

Đánh giá:

Không có công cụ nào phù hợp cho tất cả để kiểm thử tự động. Rất khuyến khích người dùng kiểm thử đánh giá các công cụ khác nhau để chọn ra những gì sẽ đáp ứng tốt nhất nhu cầu kiểm thử tự động của họ. Các ngôn ngữ lập trình và công nghệ được sử dụng để phát triển phần mềm tiếp tục phát triển, cũng như các công cụ kiểm thử tự động, khiến chi phí trở thành một yếu tố quan trọng trong việc lựa chọn công cụ. Các nhà cung cấp thương mại thường tính phí nâng cấp công cụ, có thể là đáng kể nếu phần mềm của bạn sử dụng các công nghệ mới nổi và thường xuyên thay đổi.

Mặt khác, các công cụ mã nguồn mở và phi thương mại không phải chịu phí bổ sung mà đòi hỏi nỗ lực và chuyên môn để tích hợp các nâng cấp mới. Thật khó để tìm thấy sự hỗ trợ và chuyên môn cần thiết để tích hợp các công cụ và framework khác nhau vào các giải pháp mã nguồn mở. Các công cụ mới nổi tích hợp với các framework mã nguồn mở, như Katalon, cung cấp một giải pháp thay thế khả thi cho cả các giải pháp kiểm thử tự động mã nguồn mở và thương mại.

CHƯƠNG 3: ĐẶC TẢ HỆ THỐNG PHẦN MỀM

3.1 Giới thiệu về hệ thống phần mềm

Trong những năm gần đây với sự phát triển vượt trội của khoa học kỹ thuật đặt biệt là công nghệ thông tin, với những ứng dụng của công nghệ thông tin vào các lĩnh vực đã đóng góp phần to lớn cho sự nghiệp phát triển của con người. Trong các lĩnh vực đó thì lĩnh vực quản lý là thật sự giúp ích được rất nhiều cho con người, việc áp dụng quản lý và mua bán bằng máy tính thay cho quản lý và mua bán bằng thủ công ở các doanh nghiệp, công ty, cá nhân... là rất cần thiết và thật sự cần thiết. Do đó, việc “quản lý mua bán hàng qua mạng” là không thể thiếu được trong mọi doanh nghiệp, cửa hàng vừa và nhỏ.

Năm 2008 là năm đầu tiên Việt Nam có sự phát triển lớn mạnh về lĩnh vực thương mại khi chính thức trở thành thành viên thứ 150 của tổ chức thương mại thế giới WTO.

Với sự phát triển mạnh mẽ đó không thể phủ nhận sự đóng góp của thương mại điện tử, một lĩnh vực nóng bỏng hiện nay! Một đất nước đang phát triển mạnh mẽ, cuộc sống con người càng ngày càng nâng cao, mức tiêu thụ sản phẩm ngày càng tăng...Tóm lại nhu cầu con người ngày càng cao. Vì thế việc trao đổi mua bán cũng như quản lý hàng hóa cần phải có sự thay đổi từ thủ công sang máy móc.

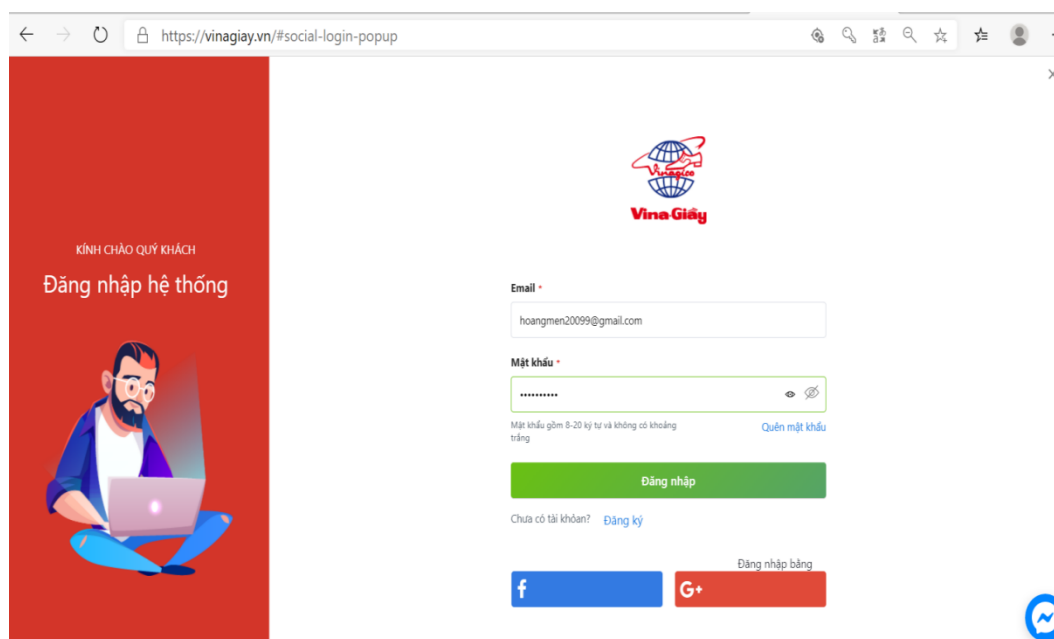
Như chúng ta cũng thấy trong thị trường hiện nay thì việc cạnh tranh về kinh doanh ngày càng trở nên quyết liệt và hầu hết những nhà kinh doanh, những công ty lớn đều rất chú tâm đến việc làm thỏa mãn khách hàng một cách tốt nhất. So với kinh doanh truyền thống thì thương mại điện tử không những giúp giảm thiểu chi phí mà còn đạt hiệu quả cao hơn.

Hơn thế nữa, với lợi thế của công nghệ Internet nên việc truyền tải thông tin về sản phẩm nhanh chóng, thuận tiện. Kết hợp với bộ phận giao hàng tận nơi, là thông qua bưu điện và ngân hàng để thanh toán tiền, càng tăng thêm thuận lợi để loại hình này phát triển.

Và đối tượng kinh doanh mà em đề cập ở đây là website bán giày. Nắm bắt được tình hình hiện tại cũng như nhu cầu của con người và sự phát triển của công nghệ, dịch vụ kinh doanh online hiện nay nên em đã chọn đề tài: “Kiểm thử tự động chức năng cho website vinagiay.vn sử dụng Katalon Studio.

3.2 Các yêu cầu chức năng

3.2.1 Chức năng đăng nhập



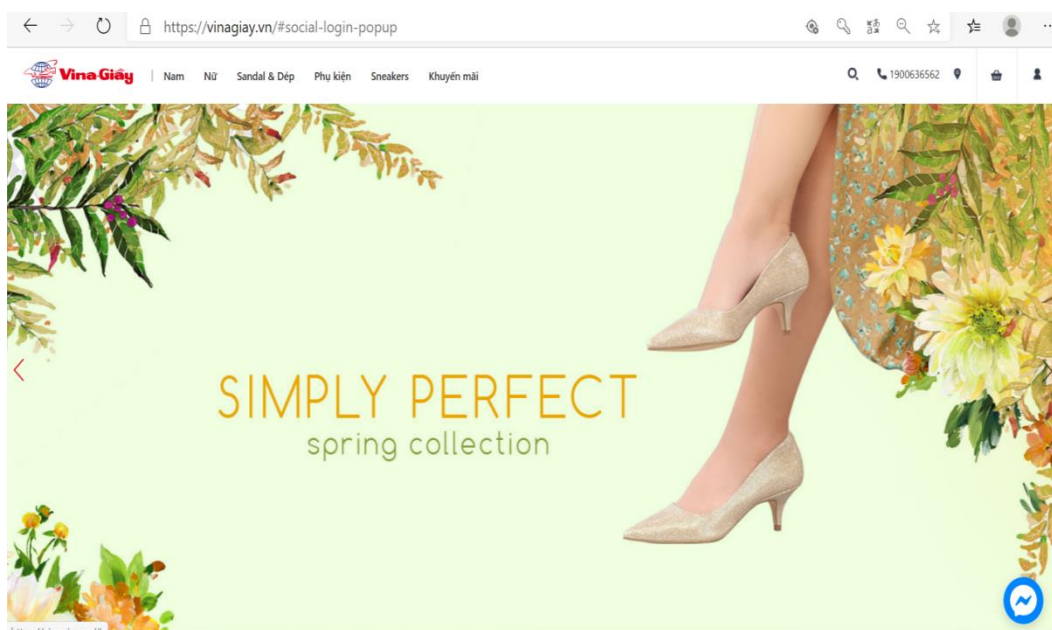
Hình 3-1: Màn hình “Đăng nhập”

- Chức năng [Đăng nhập]: cho phép người dùng đăng nhập tài khoản đã đăng ký vào trang web.
 - Validate:
 - + Email: Là tên đăng nhập đã đăng ký, có độ dài từ 8-20 ký tự, không được bỏ trống
 - + Mật khẩu: Mật khẩu được mã hóa dưới dạng (“*****”), có độ dài từ 8-20 ký tự, không được bỏ trống
 - Chức năng: Người dùng nhập thông tin vào các ô Email, mật khẩu. Sau đó ấn [Đăng nhập]
 - + Nếu [Email] được nhập không đúng định dạng thì sẽ hiển thị 1 label thông báo ngay dưới email với nội dung “Email bạn nhập không đúng”

+ Nếu [Mật khẩu] không đúng sẽ hiển thị một label màu đỏ ngay trên Email với nội dung: “Bạn đã đăng nhập không thành công. Vui lòng thử lại!”

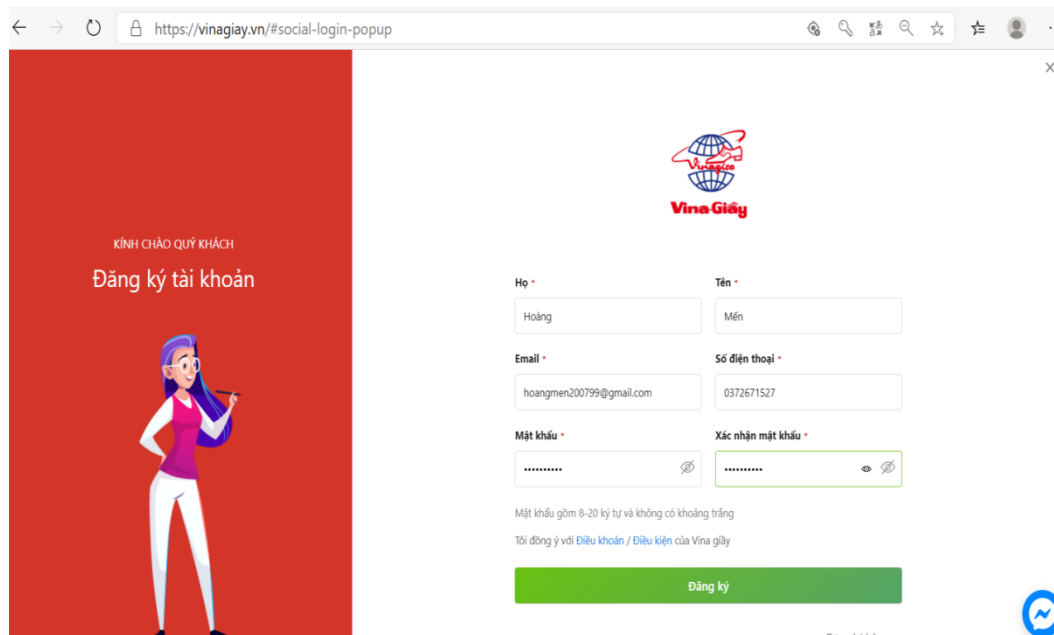
+ Nếu [Đăng nhập] thành công, chuyển người dùng đến trang quản lý thông tin của họ.

- Chức năng [Quên mật khẩu]: Khi nhấn vào quên mật khẩu, người dùng sẽ được chuyển đến trang [Phục hồi mật khẩu].
- Chức năng [Quay về]: Khi nhấn vào quay về, người dùng sẽ được chuyển đến [Trang chủ].



Hình 3-2: Màn hình sau khi [Đăng nhập] thành công

3.2.2 Chức năng đăng ký



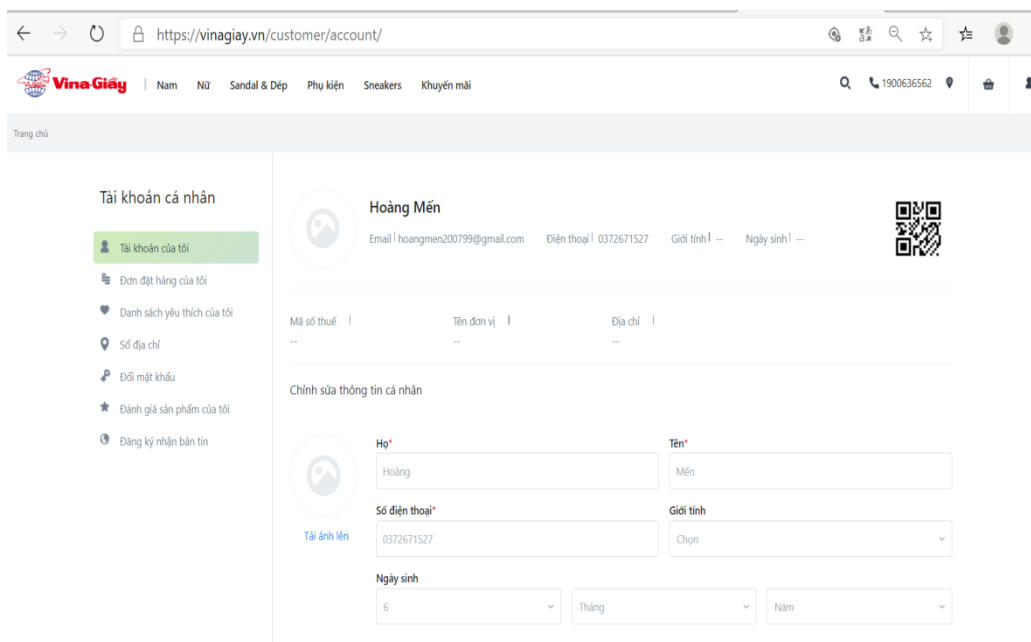
Hình 3-3: Màn hình “Đăng ký”

Chức năng [Đăng ký]: Cho phép người dùng đăng ký tài khoản để mua hàng, có thể đăng nhập vào hệ thống.

- Validate:
 - + Họ: không bỏ trống
 - + Tên: Không được bỏ trống.
 - + Email: Là tên đăng nhập đã đăng ký, có độ dài từ 8-20 ký tự, không được bỏ trống
 - + Mật khẩu: Mật khẩu được mã hóa dưới dạng (“*****”), có độ dài từ 8-20 ký tự, không được bỏ trống
 - + Số điện thoại
 - + Xác nhận mật khẩu
- Chức năng: Người dùng nhập dữ liệu vào các ô. Sau đó nhấn [Đăng ký].
 - + Các trường không hợp lệ đều sẽ hiển thị 1 label trên cùng với nội dung “Yêu cầu không hợp lệ, hoặc quá hạn, phiền bạn thử lại”.
 - + Nếu Email bị trùng: Hiển thị 1 label thông báo trên cùng với nội dung: “Email đã tồn tại. Nếu bạn quên mật khẩu, bạn có thể thiết lập mật

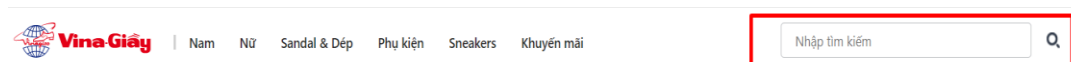
khẩu tại đây.”. Nếu ấn vào thiết lập lại mật khẩu, người dùng sẽ được chuyển đến trang [Phục hồi mật khẩu.]

- Chức năng [Đăng ký]: Sau khi đăng ký thành công, người dùng sẽ được chuyển đến trang thông tin tài khoản của mình.
- Chức năng [Quay về]: Khi nhấn vào quay về, người dùng sẽ được chuyển đến [Trang chủ].



Hình 3-4: Màn hình sau khi [Đăng ký] thành công

3.2.3 Màn hình chức năng tìm kiếm

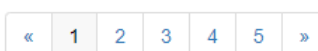


Hình 3-5: Màn hình [Tìm kiếm]

Chức năng [Tìm kiếm]: Cho phép người dùng tìm kiếm gần đúng theo tên sản phẩm. Ngoài ra, người dùng còn có thể tìm kiếm theo danh mục như hình trên.

- Validate:
 - + Chuỗi tìm kiếm: minlength(0), maxlength(250)
- Chức năng:
 - + Nếu không nhập gì, sẽ không hiển thị ra sản phẩm nào.

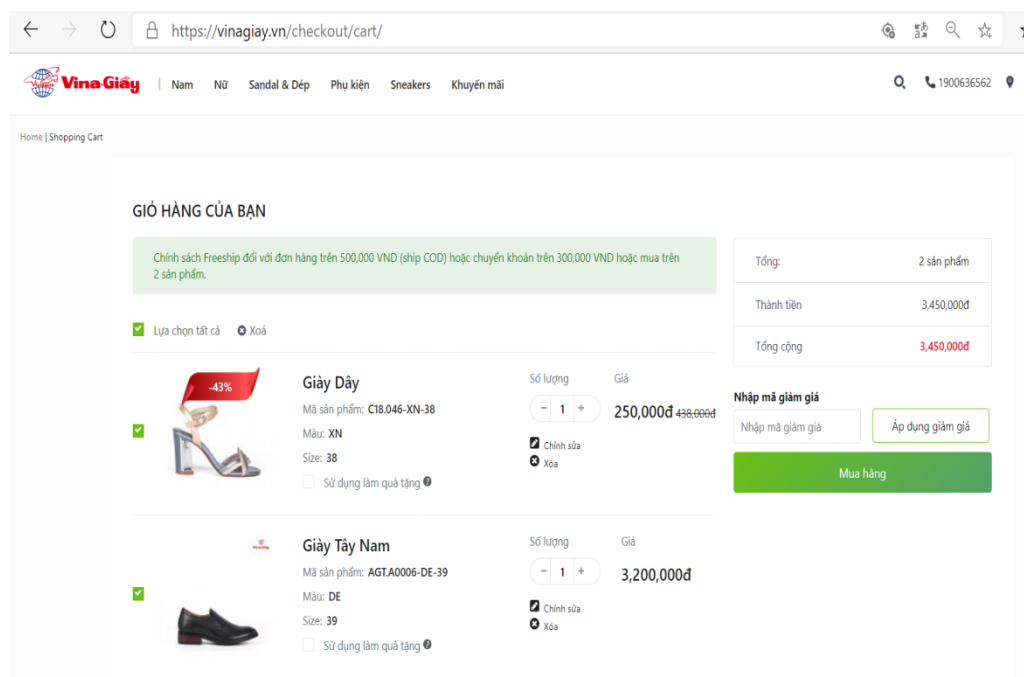
- + Nếu nhập một chuỗi ký tự, hệ thống sẽ so sánh các sản phẩm có chuỗi ký tự đó và hiển thị theo danh sách.
- + Kết hợp kiểm thử chức năng tìm kiếm với phân trang.
- Chức năng [phân trang]: (kết hợp kiểm tra với chức năng tìm kiếm): cho phép hiển thị danh sách đầu sách dưới dạng các trang khi số bản ghi vượt quá 10 (mỗi trang sẽ hiển thị 10 bản ghi, nếu vượt quá 10 sẽ phân trang dữ liệu).



Hình 3-6: Ảnh minh họa phân trang

- TH1: số bản ghi < 10 bản ghi (tức có 1 trang): chỉ hiển thị số trang: 1 dưới thanh phân trang, các nút “Trang trước”, “Trang sau”.
- TH2: số bản ghi > 10 bản ghi (tức có nhiều trang):
 - + số trang tối đa sẽ hiển thị là: (số bản ghi/số bản ghi 1 trang) lấy phần nguyên thêm 1.
 - + Tại trang 1: nút “Trang trước” bị vô hiệu hóa, các nút khác trên thanh phân trang sử dụng bình thường, khi chọn “Trang sau” sẽ chuyển tới trang 2, hoặc muốn chọn trang nào trực tiếp nhấn vào số trang đó.
 - + Tại trang $n > 1$: tất cả các nút tên thanh phân trang sử dụng bình thường, khi nhấn nút “Trang trước” sẽ chuyển về trang 1, khi chọn “Trang sau” sẽ chuyển tới sau 1 trang.
 - + Tại trang cuối cùng: nút “Trang sau” bị vô hiệu hóa, khi chọn “Trang trước” sẽ về trang $n-1$ (tức trang trước trang cuối).

3.2.4 Quản lý giỏ hàng



Hình 3-7: Màn hình chức năng quản lý giỏ hàng

Chức năng quản lý [Giỏ hàng]: Giúp người dùng quản lý số sản phẩm mình đã chọn để mua. Trong giỏ hàng, người dùng có thể thay đổi số lượng, ghi chú, cập nhật giỏ hàng, xóa 1 sản phẩm, thanh toán.

- Người dùng chọn sản phẩm:
 - + Khi thêm 1 sản phẩm bất kỳ vào giỏ hàng. Số lượng của sản phẩm mặc định là 1.
 - + Khi thêm 1 sản phẩm đã có trong giỏ hàng. Số lượng của sản phẩm sẽ tăng lên 1.
- Thay đổi số lượng sản phẩm:
 - Validate: Số lượng >0 , nhỏ hơn số sản phẩm hiện có trong kho.
 - Chức năng: Thay đổi số lượng mà người dùng cần mua. Nếu sản phẩm vượt quá số lượng trong kho sẽ hiển thị thông báo với nội dung “Số lượng sản phẩm không đủ!”. Nếu nhập số lượng <0 sẽ hiển thị thông báo với nội dung “Số lượng sai. Mời bạn nhập lại số lượng >0 ”. Người dùng nhập số lượng là 0 thì sản phẩm đó sẽ bị xóa khỏi danh sách sản phẩm trong giỏ hàng.

- Ghi chú:
 - Validate: maxleight
 - Chức năng: Dừng để ghi lại lưu ý của người dùng (nếu có).
- Cập nhật giỏ hàng: Sau khi thay đổi số lượng sản phẩm. Người dùng ấn cập nhật thì thành tiền của sản phẩm, tổng tiền sẽ được cập nhật lại.
- Xóa sản phẩm khỏi giỏ hàng: Khi người dùng ấn xóa 1 sản phẩm trong giỏ. Sản phẩm đó sẽ được xóa khỏi giỏ hàng.
- Đặt hàng: Sau khi người dùng ấn thanh toán, màn hình sẽ chuyển đến trang [Đặt hàng]

3.3 Các yêu cầu phi chức năng

- Tính ứng dụng

Hệ thống cần đáp ứng được cho 10 đến 20 user cùng sử dụng. Thiết kế website cần đơn giản, phù hợp với các chuẩn màn hình thông dụng và thói quen sử dụng. Có thể chạy trên các trình duyệt google chrome, internet explorer, microsoft edge Tính toàn vẹn dữ liệu/ performcance

- Tính toàn vẹn dữ liệu

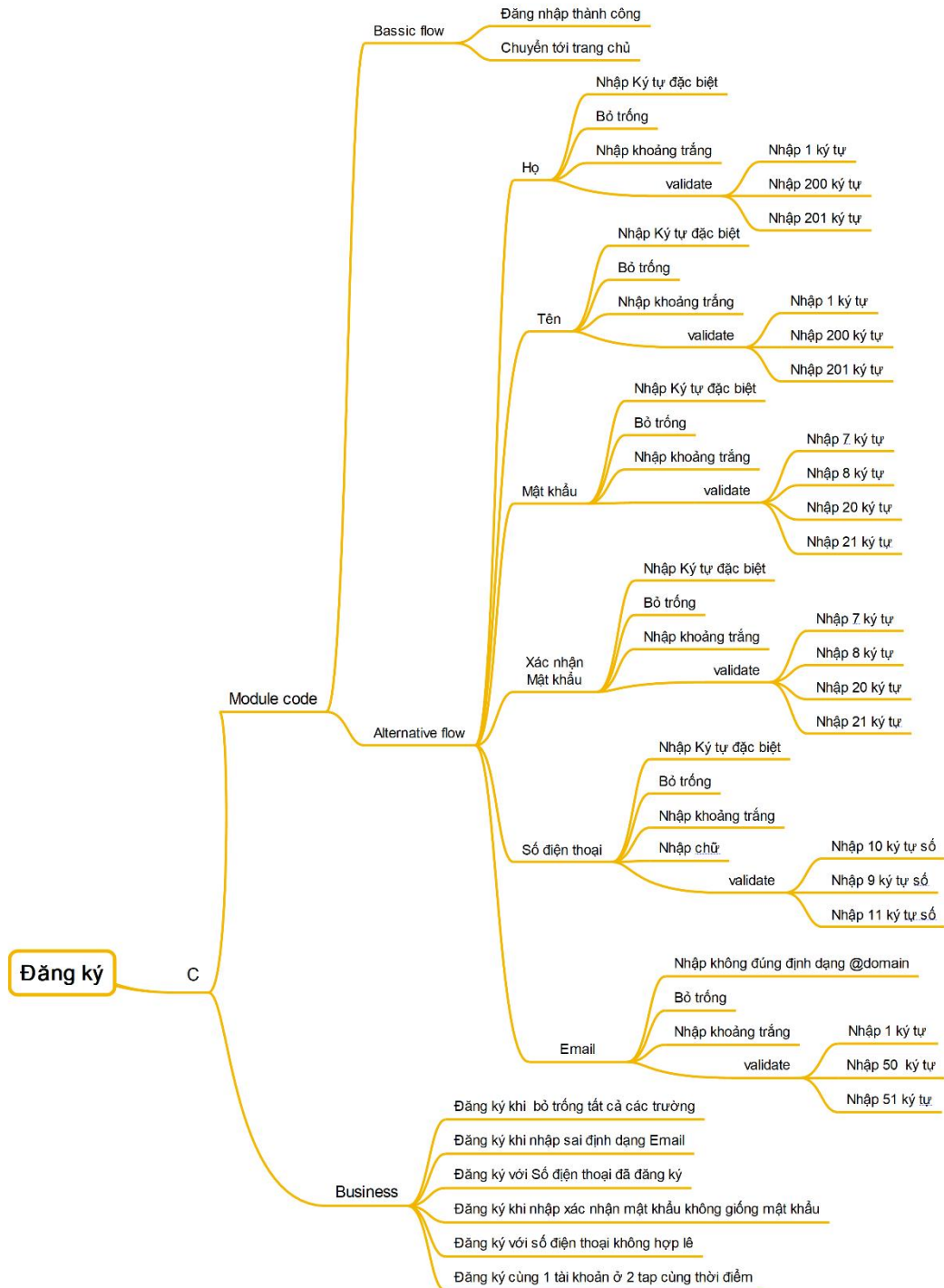
Khi có nhiều hơn 1 user cùng thay đổi 1 data trong DB thì chỉ user đầu tiên thực hiện được bình thường. Đối với các user khác sẽ không lưu lại thay đổi mà hiển thị thông báo và yêu cầu thực hiện lại xử lý.

- Performcance

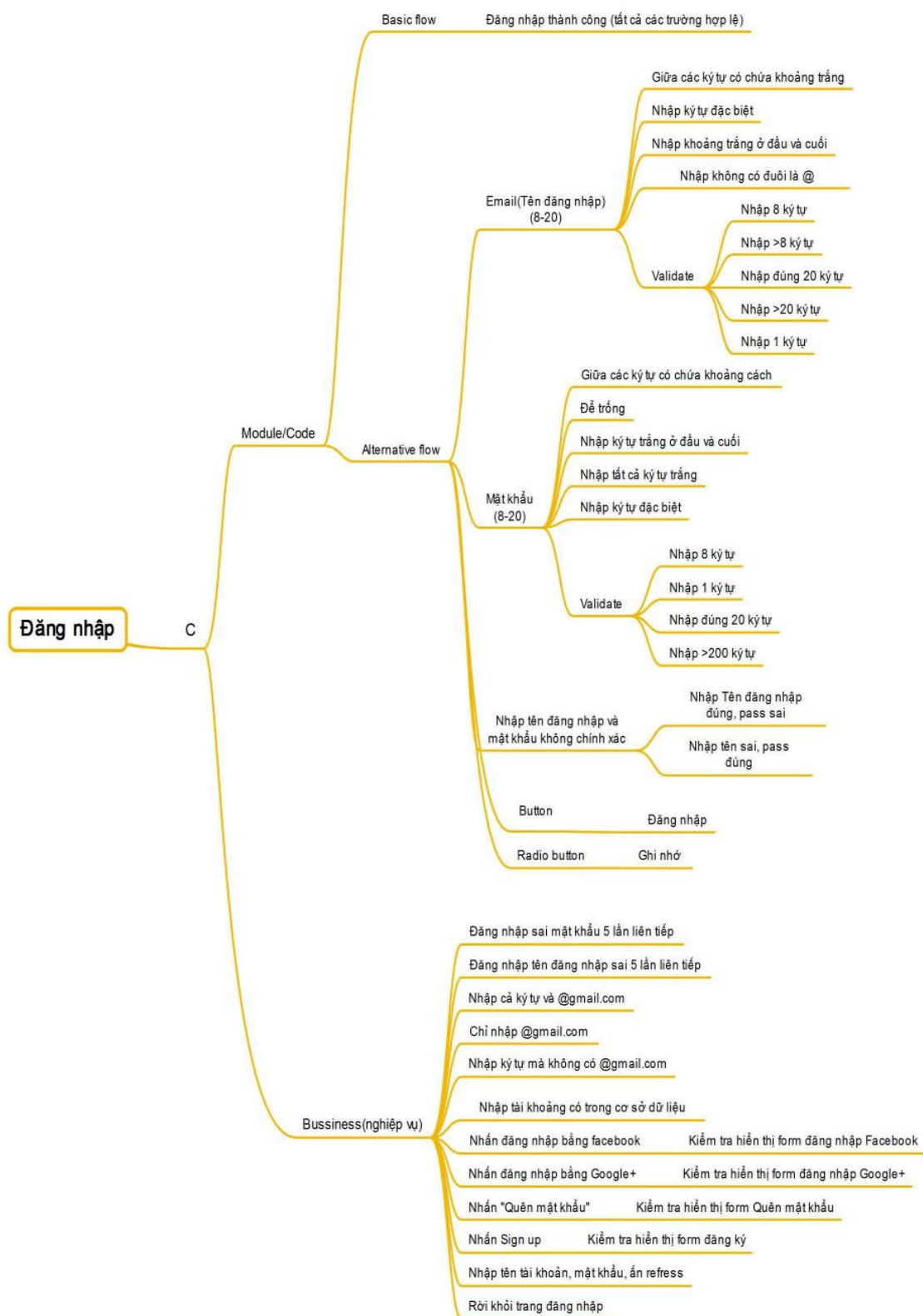
Yêu cầu thời gian mở website con bất kỳ không được chậm hơn 3 giây, và xử lý lưu thông tin không được chậm hơn 4 giây.

CHƯƠNG 4: TRIỂN KHAI KIỂM THỬ TỰ ĐỘNG

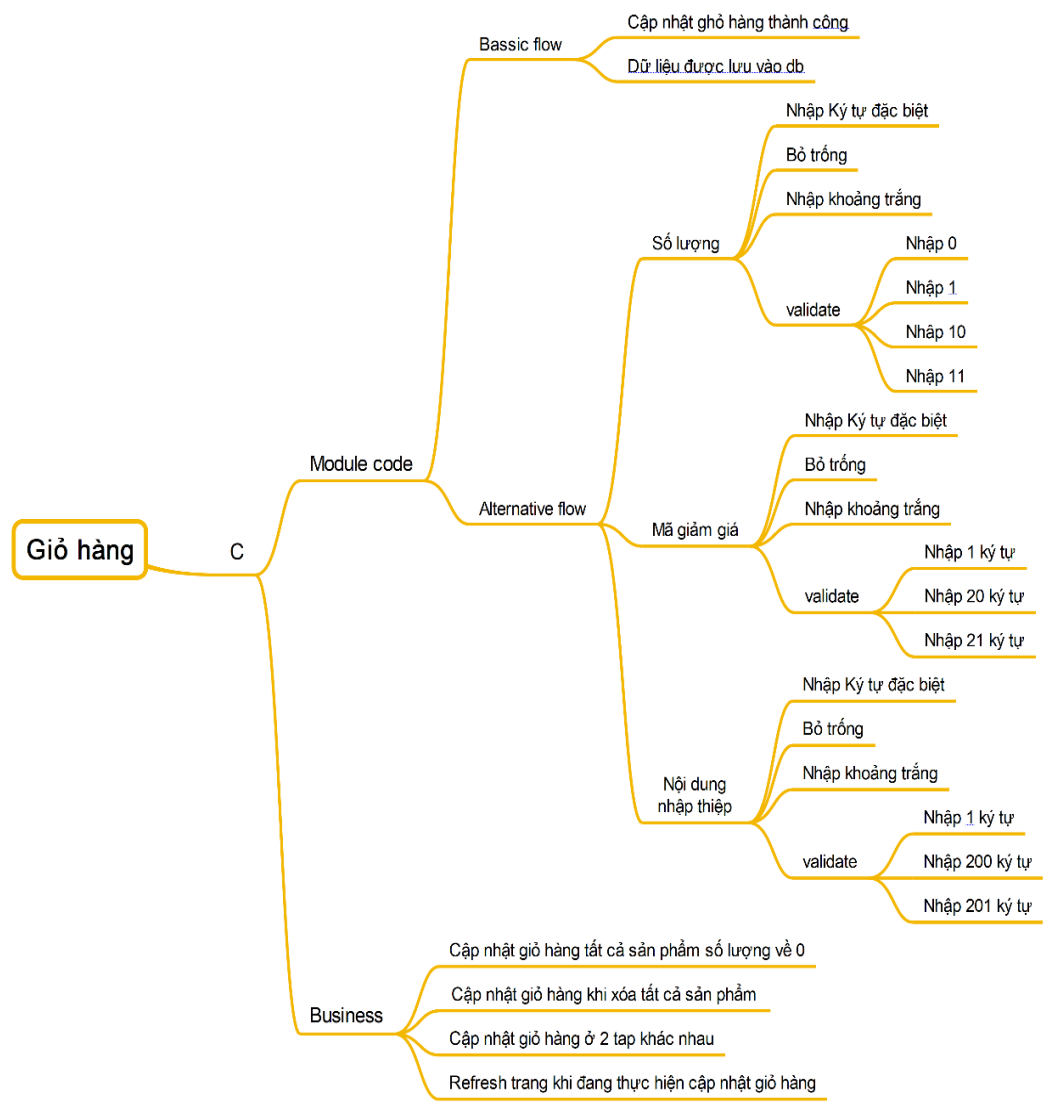
4.1 Thiết kế các yêu cầu kiểm thử



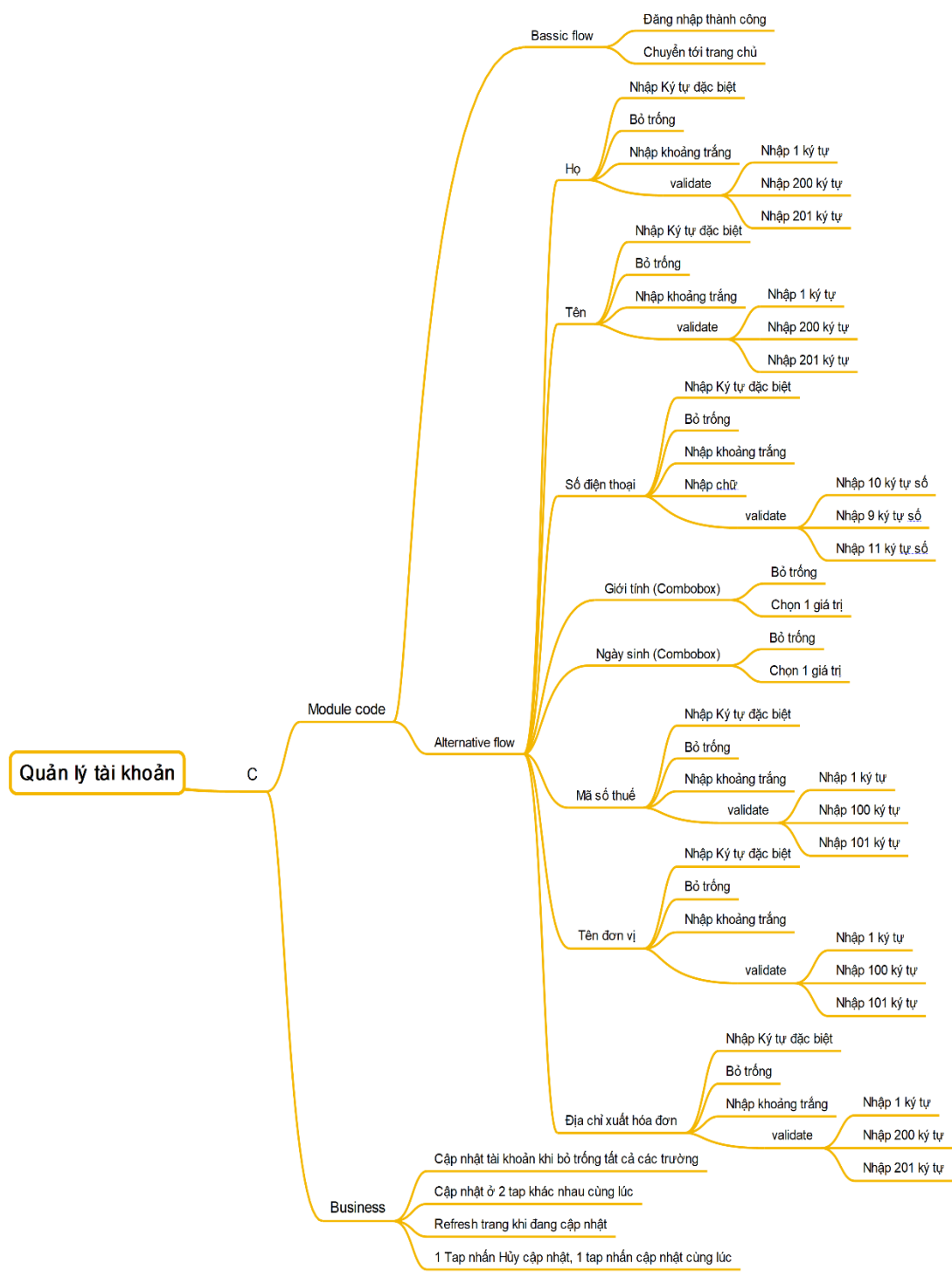
Hình 4-1: Test design chức năng Đăng ký



Hình 4-2: Test design chức năng Đăng nhập



Hình 4-3: Test design Giỏ hàng



Hình 4-4: Test design chức năng Quản lý tài khoản

4.2 Xây dựng ca kiểm thử

4.2.1 Tạo tài khoản

Bảng 4-1: Test case tạo tài khoản

Steps	Data	Expected Result
Truy cập trang sign up	Truy cập website thành công	Trang đăng ký mở
Nhập Họ	Họ	
Nhập Tên	Tên	
Nhập Email	Bỏ trống	
Nhập Số điện thoại	Số điện thoại	
Nhập Mật Khẩu	mật khẩu	
Xác Nhận mật khẩu	Xác nhận mật khẩu	
Click đăng ký	Click Đăng nhập	Kết quả

4.2.2 Đăng nhập

Bảng 4-2: Test case đăng nhập

Steps	Data	Expected Result
Truy cập trang đăng nhập	Truy cập website thành công	Trang đăng nhập mở
Nhập Email	Username	Hiển thị email
Nhập mật khẩu	Password	Hiển thị mật khẩu
Click Quên mật khẩu	Button Quên mật khẩu	Chuyển đến trang quên mật khẩu
Click nút đăng nhập	Button Đăng nhập	Chuyển sang trang Home
		Message

4.2.3 Thông tin cá nhân

Bảng 4-3: Test case Thông tin cá nhân

Steps	Data	Result
Đăng nhập tài khoản		Trang đăng nhập mở
Vào Thông tin cá nhân		
Nhập Họ	Họ	
Nhập Tên	Tên	
Nhập Số điện thoại	Số điện thoại	
Nhập Giới tính	Giới tính	
Nhập Ngày sinh	Ngày sinh	
Nhập Mã số thuế	Mã số thuế	
Nhập Tên đơn vị	Tên đơn vị	
Nhập địa chỉ xuất hóa đơn VAT	Địa chỉ xuất hóa đơn	
Click nút Hủy		Hủy hết dữ liệu vừa nhập
Click nút Cập nhật		Kết quả
Vào Số địa chỉ		
Họ	Họ	
Tên	Tên	
Mobile Number	Thay đổi số điện thoại khác	
Tỉnh/ Thành phố	Tỉnh/ Thành phố	
Phường /Xã	Phường /Xã	
Quốc gia	Quốc gia	
Mã Bưu chính	Mã bưu chính	
Click nút Hủy		Hủy hết dữ liệu vừa nhập
Click nút Cập nhật		
Kiểm tra kết quả		Kết quả

4.2.4 Đổi mật khẩu

Bảng 4-4: Test case Đổi mật khẩu

Steps	Data	Result
Mật khẩu hiện tại	Mật khẩu hiện tại	
Mật khẩu mới	Mật khẩu mới	
Nhập lại mật khẩu mới	Nhập lại mật khẩu mới	
Click Cập nhật mật khẩu		
Kiểm tra kết quả		Kết quả

4.2.5 Giỏ hàng

Bảng 4-5: Test case Giỏ hàng

Steps	Data	Expected Result
Đăng nhập		Trang đăng nhập mở
Vào giỏ hàng		
Kiểm tra giỏ hàng		Giỏ hàng trống
Thêm sản phẩm vào giỏ hàng	Sản phẩm	
Kiểm tra sản phẩm trong giỏ hàng		Sản phẩm có trong giỏ hàng số lượng =1
Thay đổi số lượng sản phẩm	Số lượng	
Kiểm tra thay đổi giỏ hàng		Số lượng thay đổi, giá thay đổi chính xác
Click Xóa sản phẩm trong giỏ hàng		
Kiểm tra thay đổi giỏ hàng		Sản phẩm xóa khỏi giỏ hàng
Thêm sản phẩm vào giỏ hàng	Sản phẩm	
Nhấn chỉnh sửa sản phẩm		
Kiểm tra sản phẩm trong giỏ hàng		Sản phẩm bị thay đổi
Nhấn thanh toán		chuyển tới trang thanh toán

4.3 Xây dựng dữ liệu kiểm thử

4.3.1 Data đăng nhập

Bảng 4-6: Data đăng nhập

Username	Password	Message
hoangmen200799@gmail.com	hoangmen99	Đăng xuất
hoangmen200799@gmail.com		Email hoặc/và mật khẩu không đúng
	hoangmen99	Email hoặc/và mật khẩu không đúng
hoangmen200799@gmail.com	hoangmen	Email hoặc/và mật khẩu không đúng
men	hoangmen99	Email hoặc/và mật khẩu không đúng
hoangmen200799@	hoangmen99	Email hoặc/và mật khẩu không đúng
men@gmail.com	hoangmen100	Email hoặc/và mật khẩu không đúng
		Email hoặc/và mật khẩu không đúng

4.3.2 Data tạo tài khoản

Bảng 4-7: Data tạo tài khoản

Họ	Tên	Email	Số điện thoại	Mật khẩu	Xác nhận mật khẩu	Kết quả
Hoàng	Mến	hoangmen200799@gmail.com	334218638	hoangmen99	hoangmen99	Đăng xuất
	Mến	hoangmen200799@gmail.com	334218638	hoangmen99	hoangmen99	Đây là trường bắt buộc
hoanghaojdck dnfrklfejdnkcm njkdsfls	Mến	hoangmen200799@gmail.com	334218638	hoangmen99	hoangmen99	Đăng xuất
h	Mến	hoangmen200799@gmail.com	334218638	hoangmen99	hoangmen99	Đăng xuất
hoang	Mến	hoangmen200799@gmail.com	334218638	hoangmen99	hoangmen99	Đăng xuất
!@#\$%^&*()	Mến	hoangmen200799@gmail.com	334218638	hoangmen99	hoangmen99	Đăng xuất
Hoàng		hoangmen200799@gmail.com	334218638	hoangmen99	hoangmen99	Đây là trường bắt buộc
Hoàng	M	hoangmen200799@gmail.com	334218638	hoangmen99	hoangmen99	

Họ	Tên	Email	Số điện thoại	Mật khẩu	Xác nhận mật khẩu	Kết quả
Hoàng	hoangthimen xinhgailawm docoaibietkh onfgdkefjlefjf erf	hoangmen200799@gmail.com	334218638	hoangmen99	hoangmen99	Đăng xuất
Hoàng	Mến	hoangmen200799@gmail.com	334218638	hoangmen99	hoangmen99	Đăng xuất
Hoàng	Mến	nguyenquocbao	334218638	hoangmen99	hoangmen99	Đăng xuất
Hoàng	Mến	bao@gmail.com	334218638	hoangmen99	hoangmen99	Đây là trường bắt buộc
Hoàng	Mến	@	334218638	hoangmen99	hoangmen99	Email không hợp lệ
Hoàng	Mến	@gmail.com	334218638	hoangmen99	hoangmen99	Email không hợp lệ
Hoàng	Mến	nguyenquocbao 199.443@gmail.com	334218638	hoangmen99	hoangmen99	Đăng xuất
Hoàng	Mến	hoangmen200799@gmail.com	334218638	hoangmen99	hoangmen99	Email đã tồn tại
Hoàng	Mến	hoangmen200799.443@gmail.com	334218638	hoangmen99	hoangmen99	Email không hợp lệ

Họ	Tên	Email	Số điện thoại	Mật khẩu	Xác nhận mật khẩu	Kết quả
Hoàng	Mến	hoangmen200799.443@gmail.com		hoangmen99	hoangmen99	Đây là trường bắt buộc
Hoàng	Mến	hoangmen200799.443@gmail.com	0	hoangmen99	hoangmen99	Vui lòng nhập số hợp lệ trong lĩnh vực này.
Hoàng	Mến	hoangmen200799.443@gmail.com	037267152777	0	hoangmen99	Vui lòng nhập số hợp lệ trong lĩnh vực này.
Hoàng	Mến	hoangmen200799.443@gmail.com	33421683	hoangmen99	hoangmen99	Vui lòng nhập số hợp lệ trong lĩnh vực này.
Hoàng	Mến	hoangmen200799.443@gmail.com	đi chơi đê	hoangmen99	hoangmen99	Vui lòng nhập số hợp lệ trong lĩnh vực này.
Hoàng	Mến	hoangmen200799.443@gmail.com	!@#\$\$%^	hoangmen99	hoangmen99	Vui lòng nhập số hợp lệ trong lĩnh vực này.
Hoàng	Mến	hoangmen200799.443@gmail.com	334216838	hoangmen99	hoangmen99	Đăng xuất
Hoàng	Mến	hoangmen200799.443@gmail.com	334218638		hoangmen99	Đây là trường bắt buộc

Họ	Tên	Email	Số điện thoại	Mật khẩu	Xác nhận mật khẩu	Kết quả
Hoàng	Mến	hoangmen200799.443@gmail.com	334218638	hoangmen	hoangmen99	Mật khẩu đã nhập không trùng nhau
Hoàng	Mến	hoangmen200799.443@gmail.com	334218638	hoangmen99	hoangmen99	Mật khẩu gồm 8-20 ký tự và không có khoảng trắng
Hoàng	Mến	hoangmen200799.443@gmail.com	334218638	hoangme	hoangmen99	Mật khẩu đã nhập không trùng nhau
Hoàng	Mến	hoangmen200799.443@gmail.com	334218638	hoangmen99		Đây là trường bắt buộc
Hoàng	Mến	hoangmen200799.443@gmail.com	334218638	hoangmen99	hoangmen	Mật khẩu đã nhập không trùng nhau
Hoàng	Mến	hoangmen200799.443@gmail.com	334218638	hoangmen99	hoangthimenxinhgailamdo	Mật khẩu gồm 8-20 ký tự và không có khoảng trắng
Hoàng	Mến	hoangmen200799.443@gmail.com	334218638	hoangmen99	hoangmen99	Mật khẩu gồm 8-20 ký tự và không có khoảng trắng

4.3.3 Data quản lý thông tin cá nhân

Bảng 4-8: Data quản lý thông tin cá nhân

Họ	Tên	Số điện thoại	Giới tính	Ngày sinh	Tháng sinh	Năm sinh	Mã số thuế	Tên đơn vị	Địa chỉ xuất hóa đơn VAT	Kết quả
Hoàng	Mến	0372671527	Nam	20	7	1999	12345	GHTK	12/38 Khúc Thừa Dụ-Cầu Giấy - Hà Nội	Cập nhật thành công
	Mến	0372671527	Nữ	20	7	1999	12346	GHTK	12/38 Khúc Thừa Dụ-Cầu Giấy - Hà Nội	Cập nhật thành công
hoanghaojd cfkdnfrklfe jdkncmnjn kdsfls	Mến	0372671527	Nữ	20	7	1999	12347	GHTK	12/38 Khúc Thừa Dụ-Cầu Giấy - Hà Nội	Cập nhật thành công
h	Mến	0372671527	Nữ	20	7	1999	12348	GHTK	12/38 Khúc Thừa Dụ-Cầu Giấy - Hà Nội	Cập nhật thành công
Hoàng		0372671527	Nữ	20	7	1999	12349	GHTK	12/38 Khúc Thừa Dụ-Cầu Giấy - Hà Nội	Cập nhật thành công
Hoàng	M	0372671527	Nữ	20	7	1999	12350	GHTK	12/38 Khúc Thừa Dụ-Cầu Giấy - Hà Nội	Cập nhật thành công

Họ	Tên	Số điện thoại	Giới tính	Ngày sinh	Tháng sinh	Năm sinh	Mã số thuế	Tên đơn vị	Địa chỉ xuất hóa đơn VAT	Kết quả
Hoàng	hoangt himen xinhga ilawm docoai bietkh onfgdk efjlefj erf	0372671527	Nữ	20	7	1999	12351	GHTK	12/38 Khúc Thừa Dụ- Cầu Giấy - Hà Nội	Cập nhật thành công
Hoàng	Mến		Nữ	20	7	1999	12352	GHTK	12/38 Khúc Thừa Dụ- Cầu Giấy - Hà Nội	Cập nhật thành công
Hoàng	Mến	0	Nữ	20	7	1999	12353	GHTK	12/38 Khúc Thừa Dụ- Cầu Giấy - Hà Nội	Cập nhật thành công
Hoàng	Mến	0372671527 77	Nữ	20	7	1999	12354	GHTK	12/38 Khúc Thừa Dụ- Cầu Giấy - Hà Nội	Cập nhật thành công
Hoàng	Mến	33421683	Nữ	20	7	1999	12355	GHTK	12/38 Khúc Thừa Dụ- Cầu Giấy - Hà Nội	Cập nhật thành công

Họ	Tên	Số điện thoại	Giới tính	Ngày sinh	Tháng sinh	Năm sinh	Mã số thuế	Tên đơn vị	Địa chỉ xuất hóa đơn VAT	Kết quả
Hoàng	Mến	đi chơi đê	Nữ	20	7	1999	12356	GHTK	12/38 Khúc Thừa Dụ-Cầu Giấy - Hà Nội	Cập nhật thành công
Hoàng	Mến	!@#\$\$%^	Nữ	20	7	1999	12357	GHTK	12/38 Khúc Thừa Dụ-Cầu Giấy - Hà Nội	Cập nhật thành công
Hoàng	Mến	0372671527		20	7	1999	12358	GHTK	12/38 Khúc Thừa Dụ-Cầu Giấy - Hà Nội	Cập nhật thành công
Hoàng	Mến	0372671527	Nam		7	1999	12359	GHTK	12/38 Khúc Thừa Dụ-Cầu Giấy - Hà Nội	Cập nhật thành công
Hoàng	Mến	0372671527	Nam	1		1999	12360	GHTK	12/38 Khúc Thừa Dụ-Cầu Giấy - Hà Nội	Cập nhật thành công
Hoàng	Mến	0372671527	Nam	20	10		12361	GHTK	12/38 Khúc Thừa Dụ-Cầu Giấy - Hà Nội	Cập nhật thành công
Hoàng	Mến	0372671527	Nam	20	7	2000		GHTK	12/38 Khúc Thừa Dụ-Cầu Giấy - Hà Nội	Cập nhật thành công
Hoàng	Mến	0372671527	Nam	20	7	1999	ABC	GHTK	12/38 Khúc Thừa Dụ-Cầu Giấy - Hà Nội	Cập nhật thành công
Hoàng	Mến	0372671527	Nam	20	7	1999	12345 ABCD hfeoi fherfer	Viettel	12/38 Khúc Thừa Dụ-Cầu Giấy - Hà Nội	Cập nhật thành công

Họ	Tên	Số điện thoại	Giới tính	Ngày sinh	Tháng sinh	Năm sinh	Mã số thuế	Tên đơn vị	Địa chỉ xuất hóa đơn VAT	Kết quả
Hoàng	Mến	0372671527	Nam	20	7	1999	12345 ABCD		12/38 Khúc Thừa Dụ- Cầu Giấy - Hà Nội	Cập nhật thành công
Hoàng	Mến	0372671527	Nam	20	7	1999	12345 ABCD	GHTK ghsndi foefens dncscs sd		Cập nhật thành công
Hoàng	Mến	0372671527	Nam	20	7	1999	12345 ABCD	GHTK	12	Cập nhật thành công
Hoàng	Mến	0372671527	Nam	20	7	1999	12345 ABCD	GHTK	12-38	Cập nhật thành công
Hoàng	Mến	0372671527	Nam	20	7	1999	12345 ABCD	GHTK	Khúc Thừa Dụ Khúc Thừa Dụ Khúc Thừa Dụ Khúc Thừa Dụ	Cập nhật thành công
Hoàng	Mến	0372671527	Nam	20	7	1999	12345 ABCD	GHTK	!@#\$\$%	Cập nhật thành công
Hoàng	Mến	0372671527	Nam	20	7	1999	12345 ABCD	GHTK	Hà	Cập nhật thành công

4.3.4 Data đổi mật khẩu

Bảng 4-9: Data đổi mật khẩu

Mật khẩu hiện tại	Mật khẩu mới	Nhập lại mật khẩu mới	Kết quả
hoangmen99	hoangmen	hoangmen	Cập nhật thành công
	hoang	hoang	Cập nhật thành công
hoangmen99		hoang	Cập nhật thành công
hoangmen99	hoangmen99		Cập nhật thành công
hoangmen99	hoangmen	hoangmen	Cập nhật thành công

4.3.5 Data Tìm kiếm

Bảng 4-10: Data Tìm kiếm

Dữ liệu	Kết quả
de	Dép nam
cao	Giày cao gót
san	Sandal Nam
Túi	Túi xách
túi	Túi xách nữ
Ví	Ví nữ
vi	Ví Nam Da Cá Sấu
sneake	Giày Sneaker

4.3.6 Data Giỏ hàng

Bảng 4-11: Data Giỏ hàng

Loại	Tên sản phẩm	Mã sản phẩm	Số lượng
Nữ	Giày Búp Bê	C18.049-35-HK	1
Nữ	Giày Cao Gót		
Nữ	Giày Boot Nữ	UES.A0014	1
Nữ	Dép Comfort		1
Nam	Giày Tăng Chiều Cao	ADE.I0017	1
Nam	Giày Boot Nam		
Nam	Giày Tây		
Sneaker	Giày Sneaker Nam	C19.072	
Sneaker	Giày Sneaker Nữ	C19.069	
Phụ kiện	Túi Xách	C.VI.0749	
Phụ kiện	Ví Nữ	C.VI.0599	
Phụ kiện	Ví Nam	C27.NI.3A1600	
Sandal & Dép	Dép Nam		
Sandal & Dép	Dép Nữ		
Sandal & Dép	Sandal Nam		
Sandal & Dép	Sandal Nữ		

4.4 Xây dựng kịch bản kiểm thử

4.4.1 Test scripts Đăng nhập

```
WebUI.openBrowser('')
WebUI.navigateToUrl('https://vinagiay.vn/#social-login-popup')
WebUI.maximizeWindow()
WebUI.click(findTestObject('Page_Giay/Icon_XemGioHang'))
WebUI.click(findTestObject('Page_Giay/a_DangNhap'))
WebUI.waitForElementVisible(findTestObject("Page_Login/input_Email_username"), 5)
WebUI.setText(findTestObject('Page_Login/input_Email_username'), username)
WebUI.setText(findTestObject('Page_Login/input_MatKhai_pass'), password)
WebUI.click(findTestObject('Page_Login/button_DangNhap'))
if(WebUI.waitForElementPresent(findTestObject('Page_Giay/Icon_XemGioHang'), 5)) {
    WebUI.verifyElementPresent(findTestObject('Page_Giay/Icon_XemGioHang'), 3)
}
else {
    WebUI.verifyTextPresent(thongBao, false)
}
```

4.4.2 Test scripts Tạo tài khoản

```
WebUI.openBrowser('')
WebUI.maximizeWindow()
WebUI.navigateToUrl('https://vinagiay.vn/#social-login-popup')
WebUI.click(findTestObject('Page_Giay/Icon_XemGioHang'))
WebUI.waitForPageLoad(10)
WebUI.click(findTestObject('Page_Giay/a_Tao tai khoan'))
WebUI.waitForElementPresent(findTestObject('Page_DangKy/input_H_lastname'), 5)
```

```
WebUI.setText(findTestObject('Page_DangKy/input_H_lastname'), 'hoang')
WebUI.setText(findTestObject('Page_DangKy/input_Tn_firstname'), 'Men')
WebUI.setText(findTestObject('Page_DangKy/input_Email_email'),
'hoangmen200799@gmail.com')
WebUI.setText(findTestObject('Page_DangKy/input_telephone'), '0372671527')
WebUI.click(findTestObject('Page_DangKy/input_Mt_khu_password_dk'))
WebUI.setText(findTestObject('Page_DangKy/input_Mt_khu_password_dk'),
'hoangmen99')
WebUI.click(findTestObject('Page_DangKy/input_password_confirmation'))
WebUI.setText(findTestObject('Page_DangKy/input_password_confirmation'),
'hoangmen99')
WebUI.click(findTestObject('Page_DangKy/button_dangky'))
WebUI.waitForPageLoad(10)
WebUI.verifyElementText(findTestObject('Page_Giay/a_Tao_tai_khoan'), 'Đăng ký
thành công')
WebUI.closeBrowser()
```

4.4.3 Test scripts Thông tin cá nhân

```
CustomKeywords.'keyword.Login'('hoangmen200799@gmail.com', 'hoangmen99')
WebUI.click(findTestObject('Page_TrangChu/icon_User'))
WebUI.click(findTestObject('Page_TrangChu/span_ThongTinCaNhan'))
'Dợi đối tượng input họ xuất hiện mới làm bước tiếp theo\r\n\r\n'
WebUI.waitForElementPresent(findTestObject('Page_TaiKhoan/input_Ho'), 3)
WebUI.setText(findTestObject('Page_TaiKhoan/input_Ho'), Ho)
WebUI.setText(findTestObject('Page_TaiKhoan/input_Ten'), Ten)
WebUI.setText(findTestObject('Page_TaiKhoan/input_SoDT'), Sodt)
WebUI.selectOptionByLabel(findTestObject('Page_TaiKhoan/select_GioiTinh'),
GioiTinh, true)
WebUI.selectOptionByValue(findTestObject('Page_TaiKhoan/select_NgaySinh'),
NgaySinh, true)
WebUI.selectOptionByValue(findTestObject('Page_TaiKhoan/select_ThangSinh'),
ThangSinh, true)
```



```
WebUI.selectOptionByValue(findTestObject('Page_TaiKhoan/select_NamSinh'),  
NamSinh, true)  
  
WebUI.setText(findTestObject('Page_TaiKhoan/input_Masothue'), MaSoThue)  
  
WebUI.setText(findTestObject('Page_TaiKhoan/input_TenDonVi'), TenDonVi)  
  
WebUI.setText(findTestObject('Page_TaiKhoan/input_DiaChi'), DiaChi)  
  
WebUI.click(findTestObject('Page_TaiKhoan/span_CapNhat'))  
  
if (WebUI.waitForElementPresent(findTestObject('Page_TaiKhoan/popup_error'),  
1)) {  
    'Kiểm tra text có hiển thị trên màn hình không\r\n'  
    WebUI.verifyTextPresent(KetQua, false)  
} else {  
    'Kiểm tra text có hiển thị trên màn hình không\r\n'  
    WebUI.verifyTextPresent(KetQua, false)  
  
    WebUI.refresh()  
  
    'Kiểm tra value của attribute của 1 element'  
  
WebUI.verifyElementAttributeValue(findTestObject('Page_TaiKhoan/input_Ho'),  
'value', Ho, 1)  
  
WebUI.verifyElementAttributeValue(findTestObject('Page_TaiKhoan/input_Ten'),  
'value', Ten, 0)  
  
WebUI.verifyElementAttributeValue(findTestObject('Page_TaiKhoan/input_DiaChi'),  
'value', DiaChi, 0)  
  
WebUI.verifyElementAttributeValue(findTestObject('Page_TaiKhoan/input_SoDT'),  
'value', Sodt.trim(), 0)  
  
WebUI.verifyElementAttributeValue(findTestObject('Page_TaiKhoan/input_MaSoThu  
e'), 'value', MaSoThue, 0)  
  
WebUI.verifyElementAttributeValue(findTestObject('Page_TaiKhoan/input_TenDonV  
i'), 'value', TenDonVi, 0)  
  
WebUI.verifyOptionSelectedByLabel(findTestObject('Page_TaiKhoan/select_GioiT  
inh'), GioiTinh, false, 0)  
  
WebUI.verifyElementAttributeValue(findTestObject('Page_TaiKhoan/select_NgayS  
inh'), 'value', NgaySinh, 0)  
  
WebUI.verifyElementAttributeValue(findTestObject('Page_TaiKhoan/select_ThangS  
inh'), 'value', ThangSinh, 0)  
  
WebUI.verifyElementAttributeValue(findTestObject('Page_TaiKhoan/select_NamSin  
h'), 'value', NamSinh, 0)  
}
```

4.4.4 Test scripts Giỏ hàng

```
CustomKeywords.'keyword.Login'('hoangmen200799@gmail.com', 'hoangmen99')

WebUI.navigateToUrl('https://vinagiay.vn/checkout/cart/')

WebUI.waitForPageLoad(30)

WebUI.waitForElementPresent(findTestObject('Page_TrangChu/icon_User'), 5)

WebDriver driver = DriverFactory.getWebDriver()

List<WebElement> listSp =
driver.findElements(ByXPath.xpath('//a[@class=\'action action-delete\']'))

println(listSp.size())

// xóa lần lượt các sản phẩm trong giỏ hàng từ dưới lên trên
for (int i = listSp.size(); i >= 1; i--) {
    WebUI.click(WebUI.convertWebElementToTestObject(listSp.get(i - 1)))

WebUI.waitForElementNotPresent(WebUI.convertWebElementToTestObject(listSp.get
(i - 1)), 5)
}

CustomKeywords.'keyword.MouseOverByText'(Loai)

CustomKeywords.'keyword.ClickByText'(TenSP)

WebUI.waitForPageLoad(30)

WebUI.click(findTestObject('Page_TrangChu/first_product'))

WebUI.waitForPageLoad(30)

masp = WebUI.getText(findTestObject('Page_TrangChu/maSP'))

tensp = WebUI.getText(findTestObject('Page_TrangChu/tenSP'))

giasp = WebUI.getText(findTestObject('Page_TrangChu/giaSP'))

while (WebUI.waitForElementNotPresent(findTestObject('Page_TrangChu/load'),
3)) {
    WebUI.click(findTestObject('Page_TrangChu/Button_MuaNgay'))
    //WebUI.click(findTestObject('Page_TrangChu/Button_MuaNgay'))
}

WebUI.waitForPageLoad(30)

WebUI.verifyElementText(findTestObject('Page_GioHang/maSP'), masp)

WebUI.verifyElementText(findTestObject('Page_GioHang/tenSP'), tensp)

WebUI.verifyElementText(findTestObject('Page_GioHang/giaSP'), giasp)
```

```
soLuong1 = WebUI.getAttribute(findTestObject('Page_GioHang/txtSoLuong'),
'value' //lấy ra số lượng mặc định
)

WebUI.verifyMatch(soLuong1, '1', false)

WebUI.click(findTestObject('Page_GioHang/TangSl'))

WebUI.waitForElementNotPresent(findTestObject('Page_TrangChu/load'), 10)

soLuong2 = WebUI.getAttribute(findTestObject('Page_GioHang/txtSoLuong'),
'value' //lấy ra số lượng mặc định
)

WebUI.verifyMatch(soLuong2, '2', false)

giaExpected = CustomKeywords.'keyword.TinhGiaTien'(soLuong2, giasp)

println(giaExpected)

giaActure = WebUI.getText(findTestObject('Page_GioHang/giaSP'))

println(giaActure)

giaActureConverted = CustomKeywords.'keyword.ChuyenTienVeDangSo'(giaActure)

WebUI.verifyMatch(giaActureConverted, giaExpected, false)

WebUI.click(findTestObject('Page_GioHang/GiamSl'))

WebUI.waitForElementNotPresent(findTestObject('Page_TrangChu/load'), 10)

soLuong3 = WebUI.getAttribute(findTestObject('Page_GioHang/txtSoLuong'),
'value' //lấy ra số lượng mặc định
)

WebUI.verifyMatch(soLuong3, '1', false)

WebUI.verifyElementText(findTestObject('Page_GioHang/giaSP'), giasp)

WebUI.click(findTestObject('Page_GioHang/txtSoLuong'))

WebUI.setText(findTestObject('Page_GioHang/txtSoLuong'), '3')

WebUI.click(findTestObject('Page_GioHang/Space'),
FailureHandling.CONTINUE_ON_FAILURE)

WebUI.waitForElementNotPresent(findTestObject('Page_TrangChu/load'), 10)

giaExpected2 = CustomKeywords.'keyword.TinhGiaTien'('3', giasp)

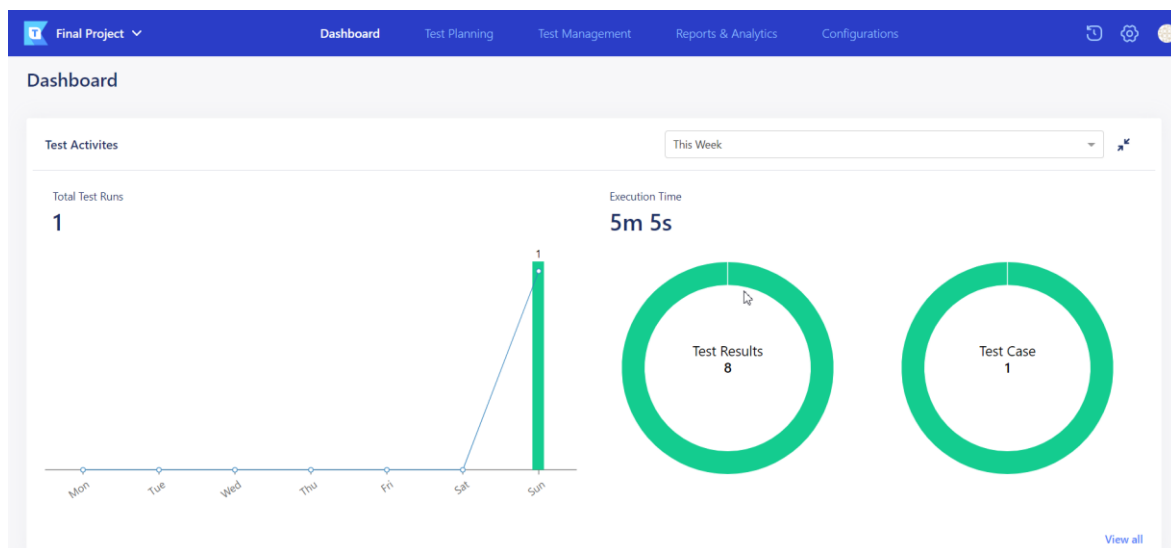
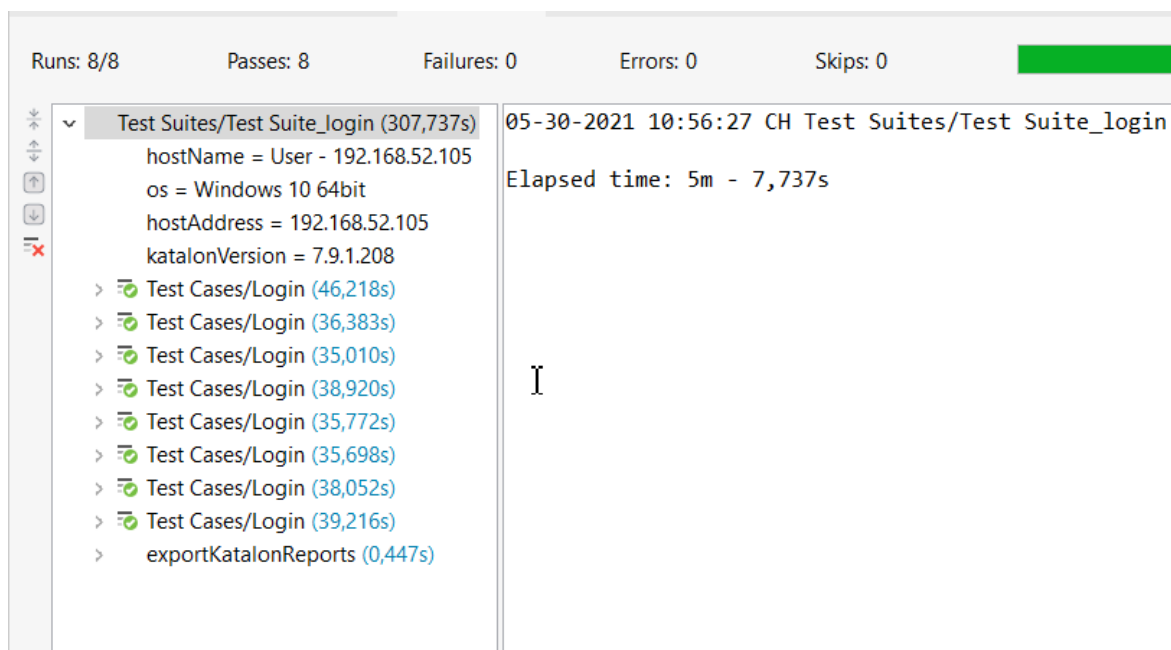
giaActure2 = WebUI.getText(findTestObject('Page_GioHang/giaSP'))

giaActureConverted2 = CustomKeywords.'keyword.ChuyenTienVeDangSo'(giaActure2)

WebUI.verifyMatch(giaActureConverted2, giaExpected2, false)
```

4.5 Thực thi và báo cáo kiểm thử

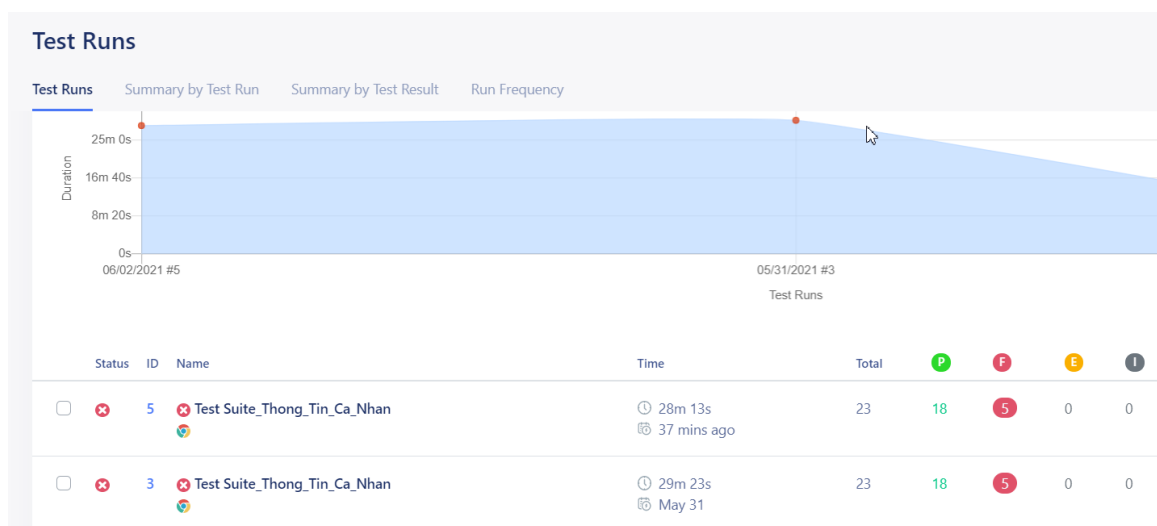
Kết quả chạy Test suite đăng nhập



Kết quả chạy test suite Thông tin cá nhân

Runs: 23/23 Passes: 18 Failures: 5 Errors: 0 Skips: 0

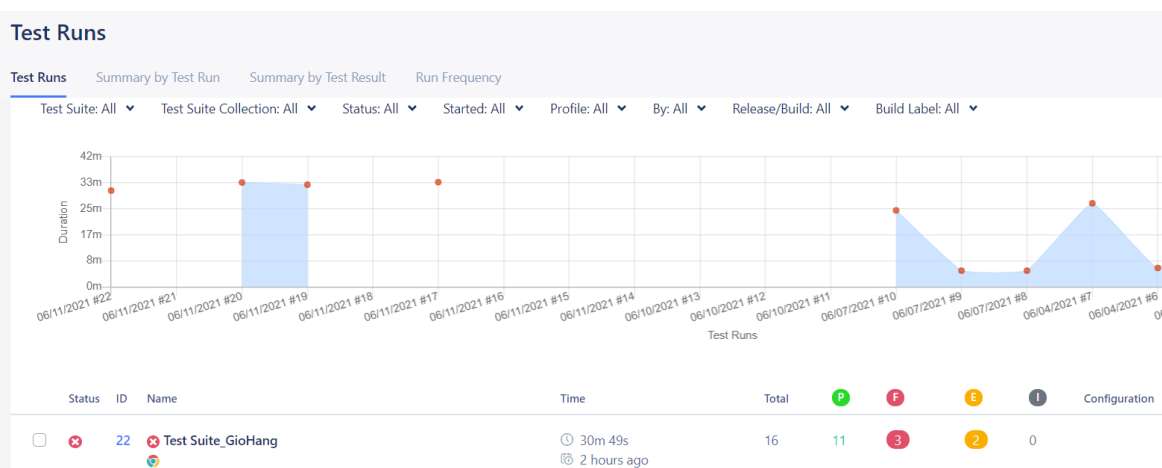
- Test Suites/Test Suite_Thong_Tin_Ca_Nhan (1696,487s)
 - hostName = User - 192.168.52.105
 - os = Windows 10 64bit
 - hostAddress = 192.168.52.105
 - katalonVersion = 7.9.1.208
 - Test Cases/TC_ThongTinCaNhan (77,934s)
 - Test Cases/TC_ThongTinCaNhan (65,309s)
 - Test Cases/TC_ThongTinCaNhan (74,803s)
 - Test Cases/TC_ThongTinCaNhan (72,621s)
 - Test Cases/TC_ThongTinCaNhan (60,536s)
 - Test Cases/TC_ThongTinCaNhan (74,869s)
 - Test Cases/TC_ThongTinCaNhan (74,798s)
 - Test Cases/TC_ThongTinCaNhan (63,096s)
 - Test Cases/TC_ThongTinCaNhan (72,350s)
 - Test Cases/TC_ThongTinCaNhan (76,319s)
 - Test Cases/TC_ThongTinCaNhan (74,553s)
 - Test Cases/TC_ThongTinCaNhan (74,062s)
 - Test Cases/TC_ThongTinCaNhan (73,263s)
 - Test Cases/TC_ThongTinCaNhan (75,799s)
 - Test Cases/TC_ThongTinCaNhan (75,161s)
 - Test Cases/TC_ThongTinCaNhan (75,279s)
 - Test Cases/TC_ThongTinCaNhan (73,829s)
 - Test Cases/TC_ThongTinCaNhan (74,723s)
 - Test Cases/TC_ThongTinCaNhan (77,053s)
 - Test Cases/TC_ThongTinCaNhan (78,155s)
 - Test Cases/TC_ThongTinCaNhan (77,430s)
 - Test Cases/TC_ThongTinCaNhan (77,978s)
 - Test Cases/TC_ThongTinCaNhan (69,864s)
 - exportKatalonReports (1,469s)



Kết quả chạy test suite Giỏ hàng

Runs: 16/16 Passes: 11 Failures: 3 Errors: 2 Skips: 0

- Test Suites/Test Suite_GioHang (1853,747s)
 - hostName = User - 192.168.52.101
 - os = Windows 10 64bit
 - hostAddress = 192.168.52.101
 - katalonVersion = 7.9.1.208
 - > sampleBeforeTestSuite (0,026s)
 - > Test Cases/TC_Giohang1 (97,780s)
 - > Test Cases/TC_Giohang1 (108,474s)
 - > Test Cases/TC_Giohang1 (377,801s)
 - > Test Cases/TC_Giohang1 (60,118s)
 - > Test Cases/TC_Giohang1 (121,486s)
 - > Test Cases/TC_Giohang1 (116,431s)
 - > Test Cases/TC_Giohang1 (112,952s)
 - > Test Cases/TC_Giohang1 (66,258s)
 - > Test Cases/TC_Giohang1 (51,583s)
 - > Test Cases/TC_Giohang1 (95,554s)
 - > Test Cases/TC_Giohang1 (108,382s)
 - > Test Cases/TC_Giohang1 (97,042s)
 - > Test Cases/TC_Giohang1 (104,854s)
 - > Test Cases/TC_Giohang1 (110,408s)
 - > Test Cases/TC_Giohang1 (109,461s)
 - > Test Cases/TC_Giohang1 (109,632s)
 - > sampleAfterTestSuite (0,000s)
 - > exportKatalonReports (2,920s)



CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Kết quả đạt được

Sau quá trình làm việc, đề tài đã kiểm thử tự động được một số chức năng cơ bản của trang web, sử dụng được công cụ kiểm thử tự động vào kiểm tra. Vận dụng các kiến thức đã học để thực thi kiểm thử 1 trang web bao gồm tất cả các giai đoạn.

Các chức năng đã chạy được kiểm thử tự động:

- Đăng nhập
- Đăng ký
- Tìm kiếm
- Giỏ hàng
- Thông tin cá nhân

Hạn chế của đề tài

Do trình độ, khả năng và thời gian còn hạn chế nên báo cáo còn tồn tại một số điểm hạn chế:

- Dễ nảy sinh lỗi trong code, và rất khó sửa.
- Cần phải ghi nhớ nhiều keyword.
- Mặc dù đã tìm tòi, cũng như chủ động trong việc tìm kiếm nguồn tài liệu, báo cáo nước ngoài bằng tiếng Anh để nghiên cứu, nhưng việc thể hiện lại bằng tiếng Việt còn nhiều bối rối khi chưa truyền tải chính xác các thuật ngữ chuyên ngành.
- Trong khuôn khổ báo cáo chưa thể trình bày đầy đủ các tính năng, và nghiên cứu các kỹ thuật nâng cao được sử dụng trong Katalon Studio như test API.
- Nguồn tài liệu bằng tiếng Việt không nhiều, đặc biệt là các tài liệu chuyên sâu.
- Chỉ áp dụng kiểm thử được chức năng đăng ký, đăng nhập tài khoản của người dùng.

Hướng phát triển của đề tài

Hướng phát triển tiếp theo em sẽ xây dựng một công cụ kiểm thử đầy đủ để thực hiện việc kiểm thử toàn bộ ứng dụng web một cách tổng thể không chỉ là kiểm thử chức năng mà còn kiểm thử hiệu năng, kiểm thử giao diện.

Xây dựng framework kiểm thử tự động hướng từ khóa áp dụng cho kiểm thử website vinagiay.vn

Cần nghiên cứu sâu hơn về Katalon Studio trong lĩnh vực an toàn thông tin - Kết hợp công cụ kiểm thử tự động Katalon Studio với công cụ kiểm thử khác để có thể cải tiến được quy trình kiểm thử. Đảm bảo ứng dụng web có thể được kiểm tra về chức năng.

TÀI LIỆU THAM KHẢO

1. Bộ môn CNPM, Đề cương “Kiểm thử phần mềm tự động”, ĐHSPTK Hưng Yên, 2016.
2. Bộ môn CNPM, Đề cương “Kiểm thử phần mềm”, ĐHSPTK Hưng Yên, 2016.
3. Bộ môn CNPM, Đề cương “Đảm bảo chất lượng phần mềm”, ĐHSPTK Hưng Yên, 2016.
4. <https://viblo.asia/p/cac-mo-hinh-phat-trien-phan-mem-GrLZDwbGKk0>
5. <https://viblo.asia/p/tim-hieu-ve-framework-gioi-thieu-mot-so-framework-pho-bien-m68Z0p2NZkG>
6. <https://viblo.asia/p/kiem-thu-tu-dong-va-cac-cong-cu-duoc-su-dung-trong-kiem-thu-tu-dong-aWj531qbZ6m>
7. <https://www.guru99.com/keyword-driven-testing.html>
8. <https://www.softwaretestinghelp.com/keyword-driven-framework-in-selenium/>
9. <https://www.devpro.edu.vn/kiem-thu-tu-dong-la-gi>
10. <https://vntesters.com/tu-dong-kiem-thu-thanh-cong-2>
11. Nguyễn Văn Vy - Nguyễn Việt Hà, Giáo trình Kỹ nghệ phần mềm, Nhà xuất bản Giáo dục Việt Nam, 2009
12. <https://viblo.asia/p/kiem-thu-tu-dong-va-kiem-thu-thu-cong-nen-sudung-khi-nao-EoDkQqEqkbV>
13. <https://www.slideshare.net/qnv96/n-kim-th-phn-mm>
14. <https://docs.katalon.com/katalon-studio/docs/index.html>
15. <https://www.katalon.com/>
16. <https://www.altexsoft.com/blog/engineering/the-good-and-the-bad-ofkatalon-studio-automation-testing-tool/>

PHỤ LỤC

Link source code: https://github.com/hoangmen99/Do_An_Tot_Nghiep