ĐẠI HỌC HUẾ TRƯỜNG ĐẠI HỌC KHOA HỌC KHOA ĐIỆN, ĐIỆN TỬ & CÔNG NGHỆ VẬT LIỆU

....**%**@**&**.....

HỒ VĂN NHẬT

ÚNG DỤNG ROBOT FRAMEWORK VÀ APPIUM TRONG KIỂM THỬ TỰ ĐỘNG WEBSITE VÀ ỨNG DỤNG ANDROID

ĐÒ ÁN TỐT NGHIỆP ĐẠI HỌC

Huế - 2023

ĐẠI HỌC HUẾ TRƯỜNG ĐẠI HỌC KHOA HỌC KHOA ĐIỆN, ĐIỆN TỬ & CÔNG NGHỆ VẬT LIỆU

....**%**[]_&....

HỒ VĂN NHẬT

ÚNG DỤNG ROBOT FRAMEWORK VÀ APPIUM TRONG KIỂM THỬ TỰ ĐỘNG WEBSITE VÀ ỨNG DỤNG ANDROID

ĐÒ ÁN TỐT NGHIỆP ĐẠI HỌC

Ngành: Công nghệ Kỹ thuật Điện tử - Viễn thông

NGƯỜI HƯỚNG DẪN: ThS. VƯƠNG QUANG PHƯỚC

Huế - 2023

LÒI CẨM ƠN

Trong hành trình thực hiện đề tài tốt nghiệp " Úng dụng Robot Framework và Appium trong kiểm thử tự động website và ứng dụng Android", tôi xin gửi lời cảm ơn sâu sắc và chân thành đến những người đã đồng hành và hỗ trợ tôi suốt quãng thời gian nghiên cứu và thực hiện đề tài này.

Đầu tiên và trước hết, tôi xin gửi lời cảm ơn chân thành đến thầy hướng dẫn của tôi, thầy **Vương Quang Phước** đã dành thời gian, kiến thức và sự hướng dẫn tận tâm để tôi có thể hoàn thành đề tài này. Sự hỗ trợ và định hướng từ thầy không chỉ giúp tôi nắm bắt được kiến thức mới mẻ mà còn giúp tôi phát triển kỹ năng nghiên cứu và thực hiện dự án.

Tôi cũng muốn gửi lời cảm ơn đến bạn bè, gia đình và những người thân yêu đã luôn ở bên cạnh tôi, động viên và cổ vũ trong những khoảnh khắc khó khăn và thách thức. Sự ủng hộ của các bạn đã là động lực quý báu giúp tôi vượt qua mọi khó khăn.

Không thể không đề cập đến các giảng viên, bạn bè và đồng nghiệp đã đóng góp ý kiến, gợi ý và hỗ trợ tôi trong quá trình nghiên cứu và thực hiện dự án. Những chia sẻ và kinh nghiệm từ mọi người đã góp phần làm cho đề tài của tôi trở nên phong phú và đa dạng hơn.

Cuối cùng, tôi xin bày tỏ lòng biết ơn chân thành đến Trường Đại học Khoa học – Đại học Huế đã cung cấp môi trường học tập và nghiên cứu tốt nhất, giúp tôi phát triển bản thân và hoàn thành đề tài tốt nghiệp này.

Xin chân thành cảm ơn!

LÒI CAM ĐOAN

Tôi – tác giả của đề tài tốt nghiệp mang tựa đề "Úng dụng Robot Framework và Appium trong kiểm thử tự động website và ứng dụng Android", cam kết rằng tất cả các thông tin và nội dung được trình bày trong báo cáo này là hoàn toàn chân thực và đã được nghiên cứu, thu thập và trình bày một cách trung thực.

Tôi xác nhận rằng mọi nguồn tài liệu, số liệu, thông tin, và kết quả nghiên cứu từ các nguồn khác nhau đã được chúng tôi trích dẫn và ghi rõ trong phần tài liệu tham khảo. Chúng tôi cam kết không sao chép hay sử dụng bất kỳ nội dung nào mà không được trích dẫn một cách đúng đắn.

Tôi hiểu rằng việc vi phạm quy tắc về sự trích dẫn và sử dụng tài liệu của người khác có thể dẫn đến việc mất đạo đức học thuật và bị xem là hành vi gian lận. Tôi cam kết tuân thủ các quy tắc này một cách nghiêm ngặt và tôn trọng quyền sở hữu trí tuệ của người khác.

Bản báo cáo này được viết dưới sự chỉ dẫn và giám sát của thầy **Vương Quang Phước**, người đã hướng dẫn và cung cấp hướng dẫn chi tiết trong quá trình nghiên cứu và viết báo cáo. Tôi xác nhận rằng toàn bộ nội dung được trình bày trong báo cáo này phản ánh kiến thức, quan điểm, và công trình nghiên cứu của chúng tôi.

Xin cam đoan tính chân thực và trung thực của bản báo cáo này.

Xác nhận của người hướng dẫnThừa Thiên Huế, ngàytháng	
	Tác giả
	Hồ Văn Nhật

MỤC LỤC

LÒI CA	ÅM ON	i
LỜI CA	AM ĐOAN	. ii
MỤC I	LŲC	iii
DANH	MỤC CÁC HÌNH VỄ	vi
DANH	MỤC CÁC BẢNG	iii
DANH	MỤC CÁC TỪ VIẾT TẮT	ix
MỞ ĐÃ	ÀU	. 1
1. LÝ	DO CHỌN ĐỀ TÀI	. 1
2. MU	ŲC TIÊU VÀ PHẠM VI ĐỀ TÀI	. 1
3. Ý 1	NGHĨA ĐỀ TÀI	. 2
4. BĆ	CỤC NỘI DUNG CỦA ĐỒ ÁN	. 3
CHƯƠ	NG I. TỔNG QUAN VỀ KIỂM THỬ TỰ ĐỘNG	. 4
1.1.	KIỂM THỬ PHẦN MỀM	. 4
1.1.1.	Khái niệm về kiểm thử	. 4
1.1.2.	Vai trò của kiểm thử phần mềm	. 5
1.1.3.	Quy trình kiểm thử phần mềm	. 5
1.1.4.	Các kỹ thuật kiểm thử phần mềm	10
1.1.5.	Các giai đoạn hay cấp độ kiểm thử phần mềm	11
1.1.6.	Một số loại hình kiểm thử phổ biến	14
1.2.	KIỂM THỬ TỰ ĐỘNG (AUTOMATION TEST)	15
1.2.1.	Khái niệm	15
1.2.2.	Quy trình kiểm thử tự động	15
1.2.3.	Các loại kiểm thử	16
1.2.4.	Ưu và nhược điểm của kiểm thử tự động	17
1.3.	TỔNG KẾT CHƯƠNG	18
CHƯƠ	NG II. CÔNG CỤ KIỂM THỬ TỰ ĐỘNG ROBOT FRAMEWORK	_

APPIU	U M	. 19
2.1.	GIỚI THIỆU VỀ ROBOT FRAMEWORK	. 19
2.2.	CÁC THÀNH PHẦN CHÍNH CỦA ROBOT FRAMEWORK	. 19
2.3.	THƯ VIỆN DÙNG TRONG ROBOTFRAMEWORK	. 20
2.3.1.	Thư viện "SeleniumLibrary"	. 20
2.3.2.	Thư viện "Browser"	. 21
2.3.3.	Một số keywords được dùng phổ biến	. 21
2.4.	CÚ PHÁP CƠ BẢN CỦA TEST CASE	. 23
2.5.	TẠO TEST SUITE VÀ TEST CASE HIERACHY	. 24
2.6.	CÁCH BẮT PHẦN TỬ GIAO DIỆN	. 26
2.6.1.	Tổng quan XPath	. 26
2.6.2.	Cú pháp của XPath	. 27
2.6.3.	Cách bắt XPath bằng ChroPath	. 27
2.7.	TỔNG KẾT CHƯƠNG	. 28
	DNG III. TIẾN HÀNH CÀI ĐẶT VÀ TRIỂN KHAI THỰC NGHIỆM V G CỤ ROBOT FRAMEWORK	
3.1.	TIẾN HÀNH CÀI ĐẶT	. 29
3.1.1.	Cài đặt python và pip	. 29
3.1.2.	Cài đặt robot framework và các thư viện liên quan	. 32
3.1.3.	Robot Framework và Appium - Cài đặt môi trường trên Windows	. 35
3.2.	TRIỂN KHAI THỰC NGHIỆM	. 38
3.2.1.	Giới thiệu về website mẫu	. 38
3.2.2.	Xây dựng các Testcases	. 40
3.2.3. xây dụ	Xây dựng kịch bản kiểm thử bằng Robot Framework dựa trên Testcase	
3.3.	KẾT QUẢ KIỂM THỬ	. 59
3.4.	BÁO CÁO KẾT QUẢ	. 61
3.5.	BÁO CÁO LÕI (BUG REPORT)	. 63

3.6.	ĐÁNH GIÁ HIỆU QUẢ KIỂM THỦ	69
3.7.	KÉT LUẬN CHƯƠNG	70
TÔNG	KẾT VÀ ĐÁNH GIÁ	71
	SÁNH ROBOT FRAMEWORK VỚI CÁC CÔNG CỤ KIỂM THỦ KHA	
•••••		71
4.2. TÔ	NG KẾT NỘI DUNG ĐỒ ÁN	71
4.3. ĐÁ	NH GIÁ ĐẠT ĐƯỢC MỤC TIÊU ĐỀ RA BAN ĐẦU	72
4.4. HU	ƯỚNG PHÁT TRIỂN VÀ NÂNG CAO TRONG TƯƠNG LAI	72
DANH	MỤC TÀI LIỆU THAM KHẢO	73

DANH MỤC CÁC HÌNH VỄ

Hình 1.1. Tương quan giữa các chặng trong PTPM và KTPM	5
Hình 1.2. Quy trinh test phần mềm	6
Hình 1.3. Tiếp nhận và phân tích yêu cầu – Requirement Analysis	6
Hình 1.4. Các loại kỹ thuật kiểm thử	. 10
Hình 1.5. Các mức độ cơ bản của kiểm thử phần mềm	. 12
Hình 1.6. Quy trình kiểm thử tự động	. 15
Hình 1.7. Tương quan giữa kiểm thử tự động và toàn bộ chu trình kiểm tra phần m	
Hình 1.8. Kiểm thử hồi quy	. 16
Hình 1.9. Kiểm thử chức năng ở góc nhìn tổng quan (High level view)	. 17
Hình 2.1. Logo Robotframework	. 19
Hình 2.2. Ví dụ về 1 test case bằng robotframework	. 20
Hình 2.3. Ví dụ về việc sử dụng test suite và test case hierarchy	. 25
Hình 2.4. Cài đặt tiện ích mở rộng ChroPath cho trình duyệt Chrome	. 28
Hình 2.5. Cách bắt xpath bằng ChroPath	. 28
Hình 3.1. Kiểm tra xem Python đã cài hay chưa	. 29
Hình 3.2. Chọn Python 3.8.5 để gỡ cài đặt	. 30
Hình 3.3. Quá trình gỡ cài đặt Python bắt đầu	. 30
Hình 3.4. Download phiên bản Python cần cài đặt	. 30
Hình 3.5. Chạy phần mềm python vừa tải về	. 31
Hình 3.6. Chọn "Next" để tiếp tục cài đặt	. 31
Hình 3.7. Chọn đường dẫn lưu file cài đặt Python và chọn "Install"	. 31
Hình 3.8. Bắt đầu cài đặt python	. 32
Hình 3.9. Kiểm tra phiên bản Python và pip đã cài đặt	. 32
Hình 3.10. Update phiên bản pip	. 32
Hình 3.11. Giao diện Command Prompt	. 33

Hình 3.12.	Cài đặt thành công Robotframework	33
Hình 3.13.	Kiểm tra xem phiên bản robotframework đã cài đặt	33
Hình 3.14.	Cài đặt thành công thư viện FakerLibrary	34
Hình 3.15.	Cài đặt thành công thư viện Selenium	34
Hình 3.16.	Cài đặt thành công thư viện Browser	35
Hình 3.17.	Chọn bản Java SDK cần cài đặt	35
Hình 3.18.	Cài đặt biến môi trường cho Java JDK	36
Hình 3.19.	Chọn phiên bản Android Studio cần cài đặt	36
Hình 3.20.	Thiết lập biến môi trường cho Android Studio	38
Hình 3.21.	Giao diện Appium	38
Hình 3.22.	Giao diện Website mẫu trang đăng nhập	39
Hình 3.23.	Giao diện và menu trang chủ tài khoản Admin	39
Hình 3.24.	Giao diện và menu trang chủ tài khoản Store	39
Hình 3.25.	Giao diện và menu trang chủ tài khoản Supplier	40
Hình 3.26.	Cấu trúc cây thư mục	45
Hình 3.27.	Báo cáo kết quả kiểm thử Login trên Windows	59
Hình 3.28.	Báo cáo kết quả kiểm thử Login trên Android	60
Hình 3.29.	Báo cáo kết quả kiểm thử update profile trên Windows	60
Hình 3.30.	Báo cáo kết quả kiểm thử update profile trên Android	61

DANH MỤC CÁC BẢNG

Bảng 1.1. Tỉ lệ công việc của các giai đoạn phát triển phần mềm	5
Bảng 1.2. Biểu mẫu Test Case	8
Bảng 1.3. Ưu và nhược điểm của kiểm thử tự động	17
Bảng 1.4. So sánh kiểm thử thủ công và kiểm thử tự động	18
Bảng 2.1. Một số keywords được sử dụng phổ biến trong robotframework	22
Bảng 3.1. Kết quả sau khi kiểm thử website Balance trên Windows	61
Bảng 3.2. Kết quả sau khi kiểm thử website Balance trên Android	63
Bång 3.3. Bång so sánh giữa Robot Framework và Selenium WebDriver	71

DANH MỤC CÁC TỪ VIẾT TẮT

STT	TỪ VIẾT TẮT	DIỄN GIẢI	DỊCH NGHĨA
1	PTPM	Phát triển phần mềm	
2	KTPM	Kiểm thử phần mềm	
3	QA	Quality Analyst	Nhà phân tích chất lượng: là người kiểm tra phần mềm được phát triển để đảm bảo phần mềm đáp ứng các yêu cầu cuối cùng của khách hàng.
4	BA	Business Analysis / Business Analyst	Phân tích kinh doanh / Chuyên gia phân tích kinh doanh: chịu trách nhiệm xác minh phần mềm được xây dựng và phân phối có đáp ứng các yêu cầu cuối cùng của khách hàng hay không.
5	PM	Project Management/ Project Manager	Quản lý dự án/ Trưởng dự án: chịu trách nhiệm chính về việc quản lý dự án và đảm bảo rằng dự án thành công.
6	UI	User Interface	Giao diện người dùng
7	API	Application Programming Interface	Giao diện lập trình ứng dụng
8	HTML	HyperText Markup Language	Ngôn ngữ đánh dấu siêu văn bản
9	XML	eXtensible Markup Language	Ngôn ngữ đánh dấu mở rộng
10	QA Environment	Quality Assurance Environment	Môi trường kiểm thử

MỞ ĐẦU

1. LÝ DO CHỌN ĐỀ TÀI

Trong môi trường phát triển phần mềm ngày càng phức tạp, việc đảm bảo chất lượng sản phẩm là một thách thức quan trọng. Kiểm thử phần mềm đóng một vai trò quan trọng trong việc đảm bảo rằng sản phẩm đáp ứng các yêu cầu và chức năng của nó một cách đáng tin cậy. Trong bối cảnh này, kiểm thử tự động đã trở thành một xu hướng thiết thực giúp cải thiện hiệu suất và đảm bảo tính nhất quán trong quá trình kiểm thử.

Kiểm thử phần mềm là một quá trình diễn ra liên tục trong tất cả các giai đoạn phát triển phần mềm để đảm bảo rằng phần mềm đáp ứng các yêu cầu thiết kế và các yêu cầu này đáp ứng nhu cầu của người dùng. Công nghệ kiểm thử phần mềm đã và đang được nghiên cứu và kiểm thử phần mềm đã trở thành một quy trình bắt buộc đối với các dự án phát triển phần mềm trên toàn thế giới. Kiểm thử phần mềm là một hoạt động rất kém, tốn thời gian và khó phát hiện hết lỗi.

Sở dĩ tôi chọn chủ đề "Úng dụng Robot Framework và Appium trong kiểm thử tự động website và ứng dụng Android" là để tìm hiểu sâu hơn về một trong những công cụ kiểm thử tự động hóa hàng đầu - Robot Framework. Tính linh hoạt và khả năng tương tác của Robot Framework với nhiều thư viện hỗ trợ khác nhau đã thu hút sự quan tâm và áp dụng rộng rãi trong cộng đồng phát triển phần mềm. Tôi muốn tìm hiểu cách sử dụng Robot Framework để kiểm tra chức năng của một trang web, đặc biệt trong bối cảnh xu hướng phát triển ứng dụng web ngày càng tăng.

Dự án này không chỉ giúp tôi có cái nhìn tổng quan về Robot Framework mà còn giúp chúng tôi áp dụng kiến thức thực tế để xây dựng các trường hợp thử nghiệm tự động cho các trang web thực. Mình mong rằng thông qua dự án này mình có thể chia sẻ những kiến thức cơ bản về Robot Framework cũng như những trải nghiệm thực tế khi sử dụng công cụ này để tự động kiểm thử các ứng dụng web và android.

2. MỤC TIÊU VÀ PHẠM VI ĐỀ TÀI

Mục tiêu của đề tài:

- ✓ Nắm vững kiến thức về cách sử dụng Robot Framework trong kiểm thử tự động các trang web và ứng dụng trên Android.
- ✓ Hiểu rõ cách xây dựng các test cases và sử dụng các keywords để kiểm thử tự động chức năng của website và ứng dụng trên Android.

- ✓ Áp dụng Robot Framework để thực hiện các kịch bản kiểm thử trên các trang
 web mẫu và ứng dụng trên Android.
- ✓ Hiểu được cách tự động hóa quy trình kiểm thử và tạo báo cáo kết quả kiểm thử tự động.

Phạm vi của đề tài:

- ✓ Tìm hiểu tổng quan về Robot Framework, bao gồm cách cài đặt, cấu hình môi trường và cú pháp cơ bản.
- ✓ Xây dựng các test cases kiểm thử tự động cho một trang web mẫu và ứng dụng trên Android.
- ✓ Sử dụng các keywords và thư viện mở rộng để thực hiện các kịch bản kiểm thử cụ thể.
- ✓ Tự động hóa quy trình kiểm thử bằng cách tạo và thực thi test suite.
- ✓ So sánh Robot Framework với một số công cụ kiểm thử tự động khác để nhận biết ưu điểm và han chế của nó.
- ✓ Phạm vi của đề tài sẽ tập trung vào việc áp dụng Robot Framework trong kiểm thử tự động các trang web và ứng dụng trên Android. Đồ án sẽ không đi sâu vào các khía cạnh phức tạp khác như kiểm thử API, mà tập trung vào việc hiểu và thực hành việc sử dụng Robot Framework trong kiểm thử web và ứng dụng trên Android.

3. Ý NGHĨA ĐỀ TÀI

- ➡ Việc nghiên cứu và thực hiện đề tài "Úng dụng Robot Framework và Appium trong kiểm thử tự động website và ứng dụng Android" mang lại nhiều ý nghĩa quan trọng trong lĩnh vực phát triển phần mềm và kiểm thử:
 - ✓ Nâng cao hiệu quả kiểm thử tự động: Sự tăng cường sự nhất quán và khả năng tái sử dụng của Robot Framework giúp cải thiện hiệu quả kiểm thử tự động. Điều này giúp tiết kiệm thời gian và tài nguyên trong việc kiểm thử các ứng dụng web.
 - ✓ Tích hợp kiểm thử sớm trong quy trình phát triển: Hiểu rõ cách sử dụng Robot Framework giúp đưa quy trình kiểm thử vào giai đoạn phát triển sớm hơn, từ đó giúp phát hiện và sửa lỗi một cách nhanh chóng và tiết kiệm chi phí.
 - ✓ Phát triển kỹ năng và kiến thức: Việc nắm vững cách sử dụng một công cụ kiểm thử phổ biến như Robot Framework không chỉ giúp nâng cao kỹ năng

- thực hiện kiểm thử tự động mà còn giúp mở rộng hiểu biết về quy trình phát triển phần mềm.
- ✓ Chia sẻ kiến thức và kinh nghiệm: Kết quả nghiên cứu và thực hiện trong đề tài có thể được chia sẻ và đóng góp vào cộng đồng phát triển phần mềm. Bài viết, tài liệu và trải nghiệm cá nhân có thể hỗ trợ người khác hiểu rõ hơn về Robot Framework và kiểm thử tự động.
- ✓ Đóng góp vào sự phát triển của ngành công nghiệp phần mềm: Nắm vững kiến thức về Robot Framework và kiểm thử tự động giúp bạn đóng góp vào việc cải thiện chất lượng phần mềm và nâng cao hiệu suất của các dự án phát triển.
- ♣ Tóm lại, việc thực hiện đề tài này không chỉ mang lại lợi ích cho người thực hiện mà còn góp phần cải thiện quy trình phát triển phần mềm và kiểm thử tự động trên toàn ngành công nghiệp.

4. BỐ CỰC NỘI DUNG CỦA ĐỒ ÁN

Đồ án được chia thành 5 chương với nội dung như sau:

- **Mở đầu:** Chương này trình bày về lý do chọn đề tài, mục tiêu và phạm vi đề tài, ý nghĩa đề tài và bố cục nội dung của đồ án.
- Chương I: Tổng quan về kiểm thử tự động: Chương này trình bày những kiến thức cơ bản về kiểm thử phần mềm và kiểm thử tự động như khái niệm, các phương pháp kiểm thử, các giai đoạn kiểm thử phần mềm.
- Chương II: Công cụ kiếm thử tự động Robot Framework: Chương này trình bày tổng quan về bộ công cụ Robot Framework, giới thiệu các thành phần chính của Robot Framework, phần mềm dùng viết code và thư viện hỗ trợ cho Robot Framework. Cú pháp cơ bản của test case và cách tạo test suite, test hierarchy.
- Chương III: Tiến hành cài đặt và triển khai thực nghiệm: Chương này trình bày cách cài đặt Robot Framework, Appium, giới thiệu về website mẫu thực hiện kiểm thử, xây dựng kịch bản kiểm thử qua đó viết các module test case tương ứng, viết code automation test bằng Robot Framework dựa trên các test case đã xây dựng đồng thời chạy chương trình đó và kiểm tra kết quả.
- **Tổng kết và đánh giá:** Chương này đưa ra những kết quả đồ án đạt được, những thiếu sót chưa thực hiện được và hướng phát triển đề tài trong tương lai.

CHƯƠNG I. TỔNG QUAN VỀ KIỂM THỬ TỰ ĐỘNG

1.1. KIỂM THỬ PHẦN MỀM

1.1.1. Khái niệm về kiểm thử

Một chương trình mới được tạo ra thường chứa vài lỗi trong 100 dòng bao gồm lỗi từ quá trình lập trình và lỗi từ quá trình thiết kế. Nếu một chương trình chứa lỗi được dung để vận hành một hệ thống trực tuyến thì những hư hỏng nghiêm trọng phát sinh ra không chỉ ảnh hưởng tới công ty vận hành hệ thống đó mà còn ảnh hưởng cả tới công chúng lớn bên ngoài.

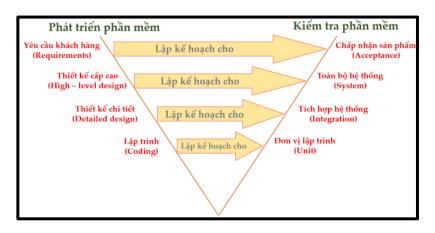
Do đó việc kiểm thử phần mềm phải được tiến hành trước khi chuyển giao sản phẩm công nghiệp. Việc kiểm thử phần mềm cũng phải được tiến hành theo một trình tự kiểm thử đặc biệt để kiểm chứng rằng chương trình và hệ thống mà nó điều khiển có thể vận hành tương ứng với các đặc tả. Mặc dù chúng ta không thể đảm bảo loại bỏ hoàn toàn hết lỗi trong chương trình nhưng chúng ta vẫn có thể giảm số lỗi đó tới mức tối thiểu nhất nếu chúng ta kiểm thử chương trình theo cách chính xác, hiệu quả.

Một hoạt động mang tính sống còn trong các dự án sản xuất hoặc gia công phần mềm, đó là kiểm tra (Testing). Người làm phần mềm chắc hẳn không ai nghi ngờ vai trò quan trọng của nó, tuy nhiên không phải ai cũng hiểu rõ hoạt động này. Bản than công việc kiểm thử phần mềm cũng là một lĩnh vực hoạt động độc lập và khá "hấp dẫn". Cùng với các dự án gia công sản xuất phần mềm, hiện cũng có khá nhiều dự án mà nội dung công việc chỉ là kiểm tra những phần đã được khách hang phát triển sẵn.

Thực tế cho thấy kiểm thử phần mềm là công việc mà bất cứ người nào từng tham gia phát triển phần mềm đều biết và từng làm. Kiểm thử phần mềm bao gồm việc "chạy thử" phần mềm hay một chức năng của phần mềm, xem nó "chạy" đúng như mong muốn hay không. Việc kiểm tra này có thể thực hiện từng chặng, sau mỗi chức năng hoặc module được phát triển, hoặc thực hiện sau cùng, khi phần mềm đã được phát triển hoàn tất.

Do đó kiểm thử một sản phẩm phần mềm là xây dựng một cách có chủ đích những tập dữ liệu và dãy thao tác nhằm đánh giá một số hoặc toàn bộ các tiêu chuẩn của sản phẩm phần mềm đó.

Kiểm tra phần mềm có nhiều mức độ khác nhau và có mối tương quan với các chặng phát triển trong dự án phát triển phần mềm.



Hình 1.1. Tương quan giữa các chặng trong PTPM và KTPM

1.1.2. Vai trò của kiểm thử phần mềm

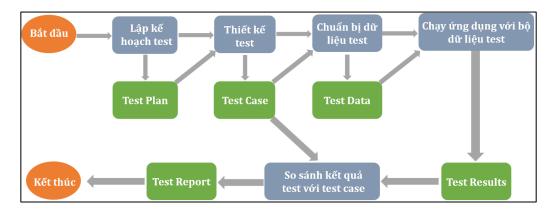
➡ Việc tạo ra một sản phẩm phần mềm phải trải qua nhiều giai đoạn, người ta gọi là quy trình phát triển phần mềm, bắt đầu từ khi bắt đầu có ý tưởng cho đến khi đưa ra sản phẩm phần mềm thực thi. Khối lượng công việc trong từng giai đoạn của quá trình sản xuất phần mềm cũng thay đổi theo thời gian. Bảng 1.1 minh hoa cụ thể hơn về điều này.

Bảng 1.1. Tỉ lệ công việc của các giai đoạn phát triển phần mềm

Giai đoạn	Phân tích yêu cầu	Thiết kế sơ bộ	Thiết kế chi tiết	Lập trình và kiểm thử đơn vị	Tích hợp và kiểm thử tích hợp	Kiểm thử hệ thống
Hai thập kỉ 1960 - 1970	10%		80%	109	%	
Thập kỉ 1980	20%		50%	200	%	
Thập kỉ 1990	40%	30%		30%		
Hiện nay		40%		30%	300	%

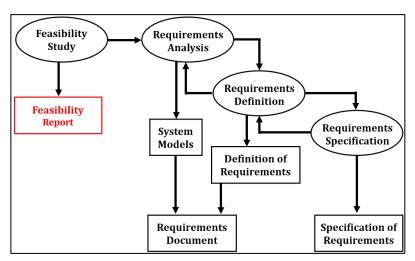
Như vậy, một sản phẩm phần mềm không chỉ đơn giản là các đoạn mã chương trình mà còn rất nhiều phần ẩn đằng sau nó. Vì vậy, việc mắc lỗi không chỉ xảy ra trong khi lập trình mà còn xảy ra cao hơn trong các công đoạn khác của quy trình phát triển một sản phẩm phần mềm. Việc kiểm thử cũng vì thế phải được tiến hành trong tất cả các phần tạo nên một sản phẩm phần mềm.

1.1.3. Quy trình kiểm thử phần mềm



Hình 1.2. Quy trinh test phần mềm

🖶 Tiếp nhận và phân tích yêu cầu – Requirement Analysis:



Hình 1.3. Tiếp nhận và phân tích yêu cầu – Requirement Analysis

- ✓ Đầu tiên, nhóm kiểm thử sẽ tiếp nhận các thông tin, yêu cầu từ khách hàng: tài liệu, mẫu thử của khách hàng, tài liệu đặc tả yêu cầu phần mềm, tài liệu thiết kế hệ thống...
- ✓ Sau khi nhận được yêu cầu, nhóm QA có trách nhiệm phân tích và đưa ra các câu hỏi cho các bên liên quan như: BA (Phân tích kinh doanh), PM (Quản lý dự án), trưởng nhóm và khách hàng để hiểu rõ hơn về yêu cầu sản phẩm và đưa ra các yêu cầu phù hợp hỗ trợ giải pháp cho khách hàng. Những câu hỏi này sẽ được lưu trữ trong file Q&A (câu hỏi và câu trả lời).
- ✓ Kết thúc bước này, bạn sẽ nhận được kết quả như sau: một tài liệu chứa các câu hỏi và câu trả lời liên quan đến hoạt động của hệ thống và một tài liệu báo cáo tính khả thi và phân tích rủi ro của việc kiểm thử phần mềm.
- ✓ Giai đoạn này rất quan trọng vì nó đánh giá liệu yêu cầu có thể kiểm chứng được hay không? Nếu vậy, chúng ta bắt đầu phần tiếp theo của quy trình kiểm thử phần mềm.

Lập kế hoạch kiểm thử Test Planning:

- ✓ Dựa vào tài liệu nhận được trong giai đoạn đầu, Test Leader hoặc Test Manager sẽ lên kế hoạch kiểm thử phần mềm cho QA team để xác định một số yếu tố:
- ✓ Phạm vi dự án: Thời gian thực hiện dự án bao lâu? Trong từng khoảng thời gian sẽ có những công việc gì?
- ✓ Phương pháp tiếp cận: Dựa vào yêu cầu chất lượng của khách hàng, thời gian test, kỹ thuật phát triển ứng dụng, lĩnh vực của sản phẩm...

✓ Các bước lập kế hoạch:

- Xác định yêu cầu kiểm tra: Xác định những gì cần kiểm tra và xác định
 phạm vi cũng như rủi ro dựa trên yêu cầu của khách hàng và thông số kỹ
 thuật yêu cầu của người dùng.
- O Xác định chiến lược kiểm thử: Xác định phương pháp, loại hình, mục tiêu kiểm thử, tổng thời gian, nguồn lực cần thiết và tiêu chuẩn đầu ra.
- Xác định nguồn lực và môi trường: Xác định nguồn nhân lực và môi trường kiểm thử (số lượng nhân sự, yêu cầu về phần mềm và phần cứng, công cụ hỗ trợ...).
- Sắp xếp giai đoạn thử nghiệm.
- Đánh giá kế hoạch: Sau khi kế hoạch được hoàn thành, test lead và các thành viên đánh giá tính hiệu quả của kế hoạch như độ bao phủ của test case, số lỗi tìm được, mức độ nghiêm trọng, tầm quan trọng của lỗi tìm thấy. Nếu chưa đạt yêu cầu thì phải điều chỉnh theo yêu cầu.
- Thông báo cho các bên liên quan: Trưởng nhóm test sẽ gửi thông báo công việc đến các thành viên thực hiện dự án và các bộ phận liên quan.

↓ Thiết kế test case cho quy trình kiểm thử phần mềm – Test Case Development:

- ✓ Bước này nhằm xác định các ca kiểm thử và các bước kiểm thử chi tiết cho từng phiên PM. Cây kiểm tra phải thể hiện đầy đủ các điều kiện lỗi có thể xảy ra để đáp ứng các yêu cầu của phần mềm. Giai đoạn này bao gồm các bước sau:
 - Review tài liệu: Đầu tiên, dựa trên tài liệu của khách hàng, người kiểm thử xác định tính năng nào cần test và tính năng nào không cần test lại để tiết kiệm thời gian mà vẫn hiệu quả.
 - Viết Test Case/Checklist: Viết test case theo kế hoạch đưa ra và áp dụng các kỹ thuật thiết kế kịch bản kiểm thử.

- Chuẩn bị dữ liệu thử nghiệm: Ngoài việc tạo các trường hợp thử nghiệm chi tiết, nhóm thử nghiệm cũng cần chuẩn bị dữ liệu thử nghiệm cho các trường hợp sử dụng cần thiết (dữ liệu, tập lệnh).
- O Review Test Case/Checklist: Sau khi hoàn thành cây test, các thành viên trong nhóm test xem lại các test case đã tạo để bổ sung, hỗ trợ lẫn nhau tránh sai sót, bổ sót test case và rủi ro trong tương lai.

Bảng 1.2. Biểu mẫu Test Case

Testcase ID	Summary	Test Step	Expected Result	Test Result
LG – 01	Test username and password = empty	 Username and Password textbox = empty Click "Login" button 	Message display 'Username and Password are required'	Passed
LG – 02	<test 2="" case=""></test>			Failed

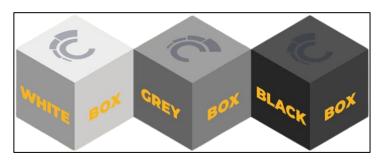
Thực hiện kiểm thử:

- ✓ Thiết lập và cài đặt môi trường: Trước khi thử nghiệm, môi trường thử nghiệm cần được thiết lập và khởi động. Đảm bảo tất cả các thành phần liên quan (phần cứng, phần mềm, máy chủ, mạng, dữ liệu...) đã được cài đặt và sẵn sàng để thử nghiệm.
- ✓ Phân loại các ca kiểm thử dựa trên mức độ ưu tiên của từng ca kiểm thử.
- ✓ Chạy test case theo các bước đã xác định trước đó.
- ✓ Báo cáo lỗi cho công cụ quản lý lỗi và so sánh với kết quả mong đợi, đồng thời theo dõi trạng thái lỗi cho đến khi được sửa chữa thành công.
- ✓ Tiến hành kiểm tra lại để xác minh các lỗi đã được sửa và tiến hành kiểm tra
 hồi quy khi có những thay đổi liên quan.
- ✓ Đánh giá kết quả test (Pass/Fail) của các test case.
- ✓ Viết báo cáo lỗi cho các tình huống mà kết quả ghi được không nhất quán với kết quả mong đợi.
- ✓ Trưởng nhóm kiểm tra cẩn thận về thời gian và thực hiện công việc bằng cách so sánh thời gian thực hiện thực tế với thời gian dự kiến để điều chỉnh cho phù hợp.

- ✓ Thường xuyên báo cáo, cập nhật cho khách hàng về quy trình kiểm thử phần mềm, chức năng nào đã thực hiện, chức năng nào đang thực hiện, tỷ lệ hoàn thành công việc.
- ✓ Kiểm tra kết quả kiểm tra: Sau khi tiến hành kiểm tra, kết quả kiểm tra cần được xem xét lại để đảm bảo độ tin cậy của kết quả. Tìm ra nguyên nhân phát hiện lỗi, nếu lỗi xảy ra do kiểm tra thì cần khắc phục lại từ đầu và kiểm tra lai.
- ✓ Kết quả của giai đoạn này là kết quả kiểm tra và báo cáo lỗi (danh sách các lỗi được tìm thấy).
- **◆ Đánh giá quy trình test phần mềm:** Bao gồm xem xét và đánh giá kết quả kiểm thử lỗi, chỉ định các yêu cầu thay đổi và tính toán số liệu liên quan đến quá trình kiểm thử.
 - ✓ Các bước đánh giá quá trình kiểm thử:
 - o Thống kê số lượng lỗi.
 - O Từ kết quả kiểm thử đánh giá chất lượng của sản phẩm.
 - Đối chiếu kết quả thực thi test case so với các tiêu chí kết thúc kiểm thử được định ra trong lúc lập kế hoạch kiểm thử.
 - Tổng hợp kết quả gửi thông tin yêu cầu sửa đến người có liên quan đến dự án.
 - Viết báo cáo tóm tắt hoạt động kiểm thử cũng như kết quả kiểm thử cho các bên liên quan.
- **Đóng quy trình kiểm thử phần mềm:** Đây là giai đoạn kết thúc hoạt động kiểm thử và phần mềm sẵn sàng được giao cho khách hàng.
 - ✓ Kiểm tra lại các lỗi nghiêm trọng đã được fix tương ứng.
 - √ Đánh giá mức độ hoàn thành tuyệt đối đối với các chỉ tiêu đặt ra từ đầu phạm vi kiểm tra, chất lượng, chi phí, thời gian và mục tiêu kinh doanh quan trọng.
 - ✓ Lưu trữ các tài liệu kiểm thử, kịch bản kiểm thử, môi trường test v.v để dùng cho những mục đích /dự án sau này.
 - ✓ Kiểm tra lại đã giao đầy đủ cho khách hàng những phần đã cam kết từ đầu.
 - ✓ Đánh giá quá trình kiểm thử cũng như rút ra bài học kinh nghiệm cho những dự án trong tương lai.

✓ Quy trình test phần mềm là một hệ thống bài bản mà tester team áp dụng để xác định công việc cần thực hiện, phân chia công việc, phản hồi thông tin để đem tới sản phẩm tốt nhất đến tay khách hàng hay thỏa mãn được nhu cầu của khách hàng.

1.1.4. Các kỹ thuật kiểm thử phần mềm



Hình 1.4. Các loại kỹ thuật kiểm thử

♣ Kiểm thử hộp đen (Black box testing): dùng để kiểm tra chức năng mà không xem xét mã nguồn cũng như cấu trúc chương trình bên trong. Thường kiểm thử hộp đen quan tâm nhiều đến các bộ dữ liệu kiểm thử đầu vào.

Ưu điểm:

- Các tester khi dùng phương pháp này sẽ không cần liên quan đến code.
- Có thể tìm được nhiều bug hơn.
- Việc kiểm thử được thực hiện bởi một cách độc lập với các Dev, cho phép quan điểm khách quan và tránh sự thiên vị.

Nhược điểm:

- Chỉ có một số lượng nhỏ các đầu vào có thể được kiểm thử và nhiều đường dẫn chương trình hoặc 1 vài phần cuối sẽ không được kiểm thử.
- Các kiểm thử có thể thừa nếu nhà thiết kế/ nhà phát triển phần mềm đã chạy kiểm thử.

Vì vậy, black box testing có ưu điểm là sản phẩm phần mềm được Kiểm thử theo một quan điểm độc lập tuy vậy vẫn còn khá nhiều nhược điểm đáng lưu ý.

♣ Kiểm thử hộp trắng (White box testing): khác với kiểm thử hộp đen, kiểm thử hộp trắng xem xét mọi module trong chương trình, các luồng thực hiện công việc để từ đó đưa ra các chiến lược kế hoạch cụ thể cho việc kiểm thử.

Ưu điểm:

- Dễ dàng tự động.
- Cung cấp các quy tắc dựa trên kỹ thuật rõ ràng cho thời điểm ngừng kiểm thử.
- Buộc các chuyên gia kiểm thử phải suy luận cẩn thận về việc test lỗi vì vậy lỗi sẽ được triệt để.

Nhược điểm:

- Khá tốn thời gian và công sức.
- Vẫn sẽ tồn tại lỗi.
- Để kiểm thử được bằng phương pháp này cần có kinh nghiệm và trình độ chuyên sâu về kiểm thử.
- ♣ Kiểm thử hộp xám (Grey box testing): là một trong các phương pháp kiểm thử phần mềm phổ biến nhất hiện nay, đây là một kĩ thuật kiểm thử mới dựa trên những đặc tính của cả kiểm thử hộp đen và hộp trắng. Mục tiêu chính của kiểm thử hộp xám là kiểm thử các ứng dụng trên nền web (web based).

Ưu điểm:

- Là sự kết hợp của kiểm thử hộp đen và hộp trắng nên sẽ tối ưu hơn.
- Kiểm thử bằng phương pháp hộp màu xám có thể thiết kế kịch bản kiểm thử phức tạp một cách thông minh hơn.

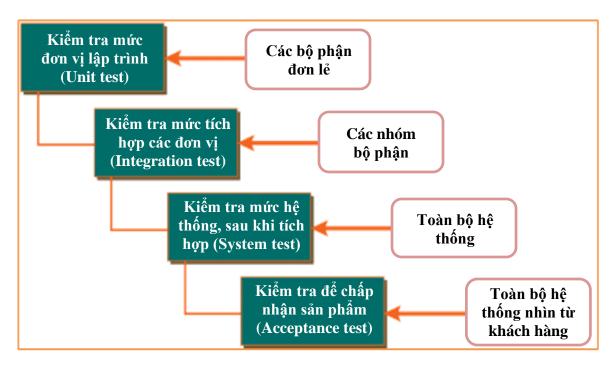
Nhược điểm:

Rất khó để liên kết lỗi khi thực hiện kiểm thử hộp xám cho một ứng dụng có hệ thống phân tán.

Trên đây là 3 phương pháp kiểm thử phần mềm cơ bản nhất mà bất cứ một lập trình viên nào cũng cần nắm. Việc lựa chọn phương pháp nào phụ thuộc vào khả năng cũng như dự án mà bạn đang thực hiện.

1.1.5. Các giai đoạn hay cấp độ kiểm thử phần mềm

Kiểm thử phần mềm nói chung có 4 mức độ sau đây:



Hình 1.5. Các mức độ cơ bản của kiểm thử phần mềm

- ♣ Kiểm thử mức đơn vị Unit Test: là quá trình kiểm thử những thành phần nhỏ nhất của phần mềm, như hàm, thủ tục, lớp, hay phương thức. Thường do lập trình viên thực hiện, Unit Test cần được thực hiện sớm trong quá trình viết code và xuyên suốt quá trình phát triển. Mục tiêu của Unit Test là đảm bảo thông tin được xử lý và xuất chính xác, kiểm tra tất cả các nhánh bên trong Unit để phát hiện lỗi. Đồng thời, chuẩn bị trước các ca kiểm thử và kịch bản kiểm thử giúp tái sử dụng các Test case và Test script.
- ♣ Kiểm thử tích hợp Integration Test: kết hợp và kiểm thử các thành phần của ứng dụng như một ứng dụng đã hoàn thành. Trái ngược với Unit Test, Integration Test tập trung vào sự giao tiếp giữa các thành phần đã kiểm thử. Hai mục tiêu chính của nó là phát hiện lỗi giao tiếp giữa các Unit và tích hợp các Unit thành các hệ thống nhỏ và cuối cùng là hệ thống hoàn chỉnh để chuẩn bị cho kiểm thử hệ thống (System Test).
 - ✓ Integration Test nên chỉ thực hiện trên những Unit đã qua kiểm thử cẩn thận bằng Unit Test, và mọi lỗi ở mức Unit cần được sửa chữa trước. Chiến lược quan trọng là tích hợp dần từng Unit, giảm số lượng kiểm thử và sai sót khi Unit được thêm vào hệ thống đã tích hợp trước đó.
 - ✓ Có 4 loại kiểm thử trong Integration Test:
 - o Kiểm thử cấu trúc (Structure Test): Đảm bảo thành phần bên trong chương trình chạy đúng.

- o **Kiểm thử chức năng (Function Test):** Chú trọng đến chức năng của chương trình theo yêu cầu kỹ thuật.
- Kiểm thử hiệu năng (Performance Test): Kiểm thử vận hành của hệ thống.
- O Kiểm thử khả năng chịu tải (Stress Test): Kiểm thử giới hạn của hệ thống.
- ♣ Kiểm thử hệ thống System Test: Kiểm thử toàn bộ hệ thống để đảm bảo thiết kế và tích hợp đáp ứng yêu cầu. Bắt đầu sau khi tích hợp xong tất cả các thành phần, loại kiểm thử này tập trung vào đánh giá hành vi và lỗi trên toàn hệ thống. System Test đòi hỏi sự tập trung vào chất lượng và hoạt động của hệ thống, có thể yêu cầu các thiết bị, phần mềm, hoặc phần cứng đặc thù.
 - ✓ Khác biệt chính giữa Integration Test và System Test là System Test tập trung vào hành vi và lỗi trên toàn hệ thống, trong khi Integration Test chú trọng vào sự giao tiếp giữa các đơn thể hoặc đối tượng. Để đảm bảo System Test hiệu quả, cần thực hiện Unit Test và Integration Test trước đó.
 - ✓ Sau Integration Test, System Test kiểm thử tất cả các khía cạnh của phần mềm, bao gồm chức năng, hiệu năng, và bảo mật. Loại kiểm thử này là bước quan trọng trước khi sản phẩm sẵn sàng cho người dùng hoặc kiểm thử chấp nhận sản phẩm. System Test thường được thực hiện độc lập với nhóm phát triển và bao gồm nhiều loại kiểm thử như Function Test, Performance Test, Stress Test, Configuration Test, Security Test, và Recovery Test:
 - Kiểm thử chức năng (Function Test): Bảo đảm các hành vi cảu hệ thống thỏa màn đúng yêu cầu thiết kế.
 - Kiểm thử hiệu năng (Performance Test): Bảo đảm tối ưu việc phân bố tài nguyên hệ thống (ví dụ bộ nhớ) nhằm đạt các chỉ tiêu như thời gian xử lý hay đáp ứng yêu cầu truy vấn...
 - Kiểm thử khả năng chịu tải (Stress Test hay Load Test): Bảo đảm hệ thống vận hành đúng dưới áp lực cao (ví dụ nhiều người truy xuất cùng lúc). Stress Test tập trung vào các trạng thái tới hạn, các "điểm chết", các tình huống bất thường như đang giao dịch thì ngắt kết nối (xuất hiện nhiều trong kiểm tra thiết bị như POS, ATM...)...
 - o Kiểm thứ cấu hình (Configuration Test)
 - Kiểm thử bảo mật (Security Test): Bảo đảm tính toàn vẹn, bảo mật của dữ liệu và của hệ thống.

- Kiểm thử khả năng phục hồi (Recovery Test): Bảo đảm hệ thống có khả năng khôi phục trạng thái ổn định trước đó trong tình huống mất tài nguyên hoặc dữ liệu; đặc biệt quan trọng đối với các hệ thống giao dịch như ngân hàng trực tuyến...
- ♣ Kiểm thử chấp nhận sản phẩm Acceptance Test: là bước cuối cùng sau System Test, được thực hiện bởi khách hàng hoặc nhóm thứ ba được ủy quyền. Mục tiêu của Acceptance Test là chứng minh rằng phần mềm đáp ứng đầy đủ yêu cầu của khách hàng và được khách hàng chấp nhận.
 - ✓ Loại kiểm thử này rất quan trọng, mặc dù System Test và Acceptance Test thường giống nhau, nhưng cách thức thực hiện và bản chất lại khác biệt. Đối với sản phẩm rộng rãi, có thể thực hiện kiểm thử Alpha Test và Beta Test. Trong Alpha Test, người dùng kiểm thử tại nơi phát triển phần mềm và ghi nhận lỗi hoặc phản hồi. Trong Beta Test, phần mềm được gửi đến người dùng để kiểm thử trong môi trường thực, với lỗi hoặc phản hồi được gửi ngược lại cho lập trình viên để sửa chữa.
 - ✓ Sự quan tâm và tham gia của khách hàng trong quá trình phát triển là quan trọng để đảm bảo kết quả Acceptance Test chính xác, tránh sai lệch lớn do hiểu sai yêu cầu hoặc mong đợi của khách hàng. Đồng thời, quan trọng là cập nhật và kiểm thử chặt chẽ tất cả tài liệu đi kèm, bao gồm hướng dẫn cài đặt và sử dụng.

1.1.6. Một số loại hình kiểm thử phổ biến

Hiên nay, do sự phát triển mạnh mẽ của công nghệ phần mềm nên có một số loại hình kiểm thử tiêu biểu như:

- ✓ Kiểm thử các phần mềm trên Desktop: đây là các ứng dụng được cài đặt trực tiếp trên máy tính cá nhân nhằm thực hiện các chức năng theo yêu cầu của người dung. Đây vẫn đang là những ứng dụng phổ biến nhất.
- ✓ Kiểm thử Web hay kiểm thử trên đám mây: với sự lớn mạnh của Internet thì các ứng dụng web cũng ngày càng phát triển và đang dần thay thế các ứng dụng trên Desktop truyền thống như Google Document, Microsoft web, app, ...
- ✓ Kiểm thử trên Mobile: ngày nay xã hội với sự phát triển nhanh chóng, các thiết bị di động (điện thoại thông minh, máy tính bảng, ...) có số lượng người dung cũng đang tăng lên chóng mặt, cùng với đó là số lượng phần mềm phục vụ cho nhu cầu cũng tăng cao, vì vậy đây là một lĩnh vực đầy tiềm năng và thách thức

trong công nghệ phần mềm.

1.2. KIỂM THỬ TỰ ĐỘNG (AUTOMATION TEST)

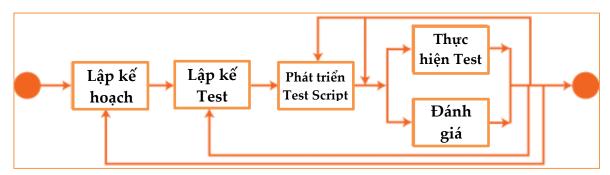
1.2.1. Khái niệm

Kiểm thử tự động là phương pháp sử dụng phần mềm hay các công cụ để xử lý tự động các bước thực hiện test case mà không cần sự can thiệp của con người. Trong kiểm thử tự động, phần mềm phải được biên dịch và chạy thực sự.

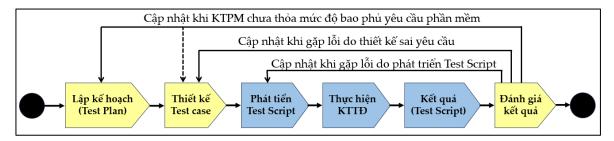
Muc tiêu:

- ✓ Giảm bót công sức và thời gian thực hiện quá trình kiểm thử cho cả một kế hoach kiểm thử.
- ✓ Tăng độ tin cậy.
- ✓ Rèn luyên kỹ năng lập trình cho tester.
- ✓ Giảm chi phí cho tổng quá trình kiểm thử.

1.2.2. Quy trình kiểm thử tự động



Hình 1.6. Quy trình kiểm thử tự động



Hình 1.7. Tương quan giữa kiểm thử tự động và toàn bộ chu trình kiểm tra phần mềm

Việc phát triển kiểm thử tự động cũng tuân theo các bước phát triển phần mềm, ta phải xem việc phát triển phần mềm giống như phát triển một dự án. Giống như phát triển phần mềm, chung ta thực hiện các bước cơ bản sau:

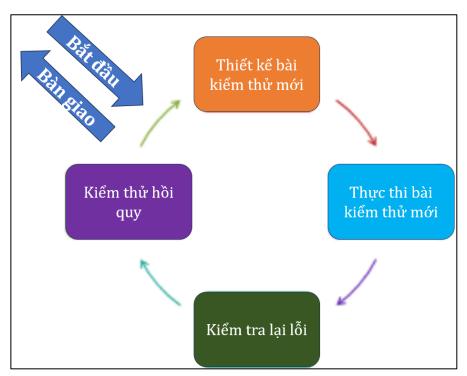
- ✓ Xây dựng yêu cầu: thu thập các đặc tả yêu cầu, lựa chọn những phần cần thực hiện kiểm thử tự động, lập kế hoạch kiểm thử.
- ✓ Phân tích thiết kế mô hình kiểm thử tự động: xây dựng mô hình phát triển kiểm thử tự động, thiết kế và xây dựng các test case để thực thi.

✓ Phát tiển Test Script:

- Tạo Test Script: giai đoạn này chúng ta sẽ sử dụng test tool để ghi lại các thao tác lên phần mềm cần kiểm tra và tự động sinh ra Test Script.
- Chỉnh sửa Test Script: chỉnh sửa để Test Script thực hiện kiểm tra theo đúng yêu cầu đặt ra, cụ thể là làm theo test case cần thực hiện.
- ✓ Chạy Test Script: giám sát các hoạt động kiểm thử phần mềm của Test Script.
- ✓ Kiểm tra kết quả: kiểm tra kết quả thông báo ngay sau khi thực hiện kiểm thử tự động.
- ✓ Đánh giá kết quả kiểm thử: thông qua báo cáo kết quả kiểm thử, bổ sung, chỉnh sửa những sai sót.

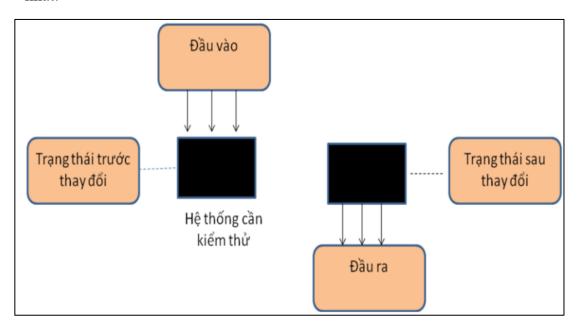
1.2.3. Các loai kiểm thử

✓ Kiểm thử hồi quy: kiểm thử tự động rất hiệu quả khi áp dụng cho giai đoạn re – test và test hồi quy. Hoặc cần chạy một lượng testcases trong một thời gian ngắn.



Hình 1.8. Kiểm thử hồi quy

- ✓ Thực hiện lặp đi lặp lại: khi chúng ta chạy test mỗi ngày cần phải add Data hoặc chạy sẵn các bước nền để test các khâu quan trọng. Khi chạy full flow mà chỉ test một phần nhỏ trong đó thì các bước đã chạy qua rồi sẽ lặp đi lặp lại thì khi đó Automation sẽ thấy tác dụng rõ nhất.
- ✓ Thử nghiệm tính năng: kiểm tra sự kết hợp giữa nhiều giá trị đầu vào ở một bước nào đó. Kiểm tra nhiều màn hình của dữ liệu đầu vào. Khi muốn thực hiện kiểm thử hiệu năng hoặc load test thì kiểm thử tự động gần như là biện pháp duy nhất.



Hình 1.9. Kiểm thử chức năng ở góc nhìn tổng quan (High level view)

1.2.4. Ưu và nhược điểm của kiểm thử tự động

Bảng 1.3. Ưu và nhược điểm của kiểm thử tự động

ƯU ĐIỂM NHƯỢC ĐIỂM ✓ Không cần đến sự can thiệp của kiểm ✓ Mất chi phí tạo các script để thực hiện kiểm thử tự động. thử viên. ✓ Giảm chi phí khi thực hiện kiểm tra Tốn chi phí cho bảo trì các script. số lương lớn test case hoặc test case ✓ Đòi hỏi kiểm thử viên phải có kỹ lặp lại nhiều lần. năng tạo các script kiểm thử tự động. ✓ Giả lập tình huống khó có thể thực ✓ Không áp dụng được trong việc tìm hiên bằng tay. lỗi mới của phần mềm.

Bảng 1.4. So sánh kiểm thử thủ công và kiểm thử tự động

MANUAL TEST	AUTOMATION TEST
Được thực hiện bởi con người, không chính xác tại mọi thời điểm do đó nó ít tin cậy.	Được thực hiện bởi các công cụ hoặc các kịch bản. Đáng tin cậy hơn.
Tốn nhiều thời gian khi thực hiện Test.	Được thực hiện bởi các công cụ phần mềm, vì vậy nó nhanh hơn.
Cần đầu tư nguồn nhân lực nhiều.	Đầu tư cho công cụ kiểm thử.
Chỉ phù hợp cho các trường hợp thử nghiệm ít lặp đi lặp lại trong thời gian dài.	Thích hợp cho các trường hợp thử nghiệm được chạy liên tục trong một thời gian dài.
Phù hợp khi cần có sự quan sát của con người hoặc trải nghiệm của khách hàng.	Không đòi hỏi sự quan sát của con người, nhưng không thể đảm bảo thân thiện với người dùng trải nghiệm thực tế.

1.3. TỔNG KẾT CHƯƠNG

Trong chương này, chúng ta đã xem xét tổng quan về kiểm thử tự động trong phát triển phần mềm. Chúng ta đã thấy rằng kiểm thử tự động đóng vai trò quan trọng trong việc đảm bảo chất lượng sản phẩm phần mềm. Bằng cách sử dụng các công cụ và kịch bản kiểm thử tự động, chúng ta có thể tăng cường tốc độ kiểm thử, tiết kiệm tài nguyên và đạt được độ chính xác cao hơn. Sự tích hợp của kiểm thử tự động trong quy trình phát triển liên tục cũng đã được nhấn mạnh, đồng thời giúp phát hiện và khắc phục lỗi sớm hơn. Tuy nhiên, cần lưu ý rằng việc triển khai kiểm thử tự động đòi hỏi sự đầu tư ban đầu và sự lựa chọn phù hợp về phương pháp kiểm thử.

Chương 2 sẽ trình bày chi tiết về Công cụ kiểm thử tự động Robot Framework để bạn đọc có cái nhìn sâu hơn về công cụ và thư viện được sử dụng trong quá trình kiểm thử tự động.

CHƯƠNG II. CÔNG CỤ KIỂM THỬ TỰ ĐỘNG ROBOT FRAMEWORK - APPIUM

2.1. GIỚI THIỆU VỀ ROBOT FRAMEWORK

♣ Robot Framework là một công cụ kiểm thử tự động mã nguồn mở và mạnh mẽ, được phát triển nhằm giúp tự động hóa quy trình kiểm thử và kiểm tra chất lượng phần mềm. Đặc biệt, Robot Framework là một công cụ linh hoạt, hỗ trợ nhiều loại kiểm thử, bao gồm kiểm thử giao diện người dùng (UI), kiểm thử API và nhiều loại kiểm thử khác.



Hình 2.1. Logo Robotframework

4 Tính năng chính của Robot Framework:

- ✓ Ngôn ngữ đơn giản và dễ đọc: Robot Framework sử dụng cú pháp gần với ngôn ngữ tự nhiên, giúp người sử dụng dễ dàng viết và đọc mã kiểm thử.
- ✓ Tích hợp dễ dàng: Robot Framework cho phép tích hợp với nhiều thư viện hỗ trợ kiểm thử khác nhau như Selenium, Appium, API testing libraries, và nhiều thư viện tự tạo khác.
- ✓ Cấu trúc tổ chức linh hoạt: Test cases và test suites trong Robot Framework có thể được tổ chức thành cấu trúc phân cấp, giúp dễ dàng quản lý và chạy các kiểm thử khác nhau.
- ✓ Tùy chỉnh keywords: Bạn có thể tạo ra các keywords tùy chỉnh để mô tả các hành động cụ thể trong quy trình kiểm thử và tái sử dụng chúng trong nhiều test cases khác nhau.

2.2. CÁC THÀNH PHẦN CHÍNH CỦA ROBOT FRAMEWORK

Test Case: Một test case trong Robot Framework là một chuỗi các bước kiểm thử, bao gồm cả việc chuẩn bị dữ liệu và kiểm tra kết quả. Test cases được viết bằng ngôn ngữ có cấu trúc rõ ràng, giúp người viết dễ dàng biểu diễn các bước kiểm thử một cách chi tiết.

♣ Keyword: Keywords là các hành động hoặc chức năng có sẵn hoặc tạo ra bởi người sử dụng để thực hiện các bước kiểm thử cụ thể. Các keywords có thể là các keywords có sẵn trong Robot Framework hoặc là các keywords tùy chỉnh được tao ra bởi người dùng.

```
*** Settings ***
                     ./keywords/common.robot
Test Setup
                    Setup
Test Teardown
                   Tear Down
*** Test Cases ***
  -----VERIEY VALIDATION TEXT-
LG_01: Verify that validation text of "Tên đăng nhập" field display when logging in with "Tên đăng nhập" field empty
   [Tags] @smoketest @regression
When Enter "text" in "Mật khấu" with "_RANDOM_"
   And Click "Đăng nhập" button
   Then Required message "Tên đăng nhập" displayed under "Đây là trường bắt buộc!" field
*** Keywords ***
Enter "${type}" in "${name}" with "${text}"
                           Get Random Text
                                                              ${type}
 ${element}=
                           Get Element Form Item By Name
                                                                                            //input[contains(@class, "ant-input")]
 Clear Text
                           ${element}
 Fill Text
                           ${element}
                                                              ${text}
 ${cnt}=
                          Get Length
                                                              ${text}
 IF ${cnt} > 0
   Set Global Variable ${STATE["${name}"]}
                                                              ${text}
Click "${text}" button
                             xpath=//button[@id='submit-btn' and text()='${text}']
 # Click
 Click
                          xpath=//button[@title = "${text}"]
 Click Confirm To Action
```

Hình 2.2. Ví dụ về 1 test case bằng robotframework

- **Test Suite:** Test suite là tập hợp các test cases có thể được tổ chức thành các nhóm logic. Test suites giúp bạn tổ chức và quản lý các test cases dễ dàng hơn, đồng thời cũng cho phép bạn thực hiện kiểm thử theo nhóm.
- ♣ Thư viện (Library): Thư viện là tập hợp các keywords đã được định nghĩa sẵn để hỗ trợ kiểm thử trong các lĩnh vực cụ thể. Robot Framework cung cấp một số thư viện sẵn có như SeleniumLibrary (cho kiểm thử UI), RequestsLibrary (cho kiểm thử API), và nhiều thư viện khác.

Thông qua những thành phần chính này, Robot Framework mang đến một cách tiếp cận linh hoạt và mạnh mẽ cho việc thực hiện kiểm thử tự động và đóng góp vào quá trình tối ưu hóa chất lượng phần mềm.

2.3. THƯ VIỆN DÙNG TRONG ROBOTFRAMEWORK

2.3.1. Thư viện "SeleniumLibrary"

♣ Thư viện "SeleniumLibrary" là một trong những thư viện quan trọng trong Robot Framework, chuyên được sử dụng để kiểm thử giao diện người dùng (UI) của các ứng dụng web. Thư viện này giúp kết nối trực tiếp với Selenium WebDriver, cho phép kiểm thử viên tương tác với trình duyệt web như Chrome, Firefox, Edge, và Safari để thực hiên các bước kiểm thử.

Tính năng và chức năng chính của thư viện "SeleniumLibrary":

- ✓ Tương tác với trình duyệt: SeleniumLibrary cho phép bạn mở trình duyệt web, nhập dữ liệu, nhấn nút, và thực hiện nhiều hành động khác trực tiếp trên trình duyệt.
- ✓ Xác nhận và kiểm tra: Bạn có thể kiểm tra các yếu tố trên trang web như văn bản, giá trị của các trường, sự hiện diện của yếu tố, và nhiều khía cạnh khác.
- ✓ Xử lý cửa sổ và tab: Thư viện cho phép bạn mở và đóng cửa sổ hoặc tab trình duyệt, chuyển đổi giữa chúng và làm việc với nhiều cửa sổ/tab cùng lúc.

2.3.2. Thư viện "Browser"

♣ Thư viện "Browser" là một thư viện mở rộng dựa trên "SeleniumLibrary", giúp tối ưu hóa việc kiểm thử các ứng dụng web bằng cách cung cấp các keywords linh hoạt hơn cho việc tương tác với trình duyệt và các yếu tố trong trang web.

♣ Tính năng và chức năng chính của thư viện "Browser":

- ✓ Quản lý trình duyệt: Thư viện "Browser" cung cấp các keywords để mở, đóng và xác định trình duyệt mặc định hoặc nhiều trình duyệt cùng lúc.
- ✓ Xác định yếu tố: Bạn có thể tìm kiếm và tương tác với các yếu tố trong trang web bằng cách sử dụng các keywords dễ đọc và gần gũi với ngôn ngữ tự nhiên.
- ✓ Xử lý popup và iframe: Thư viện cho phép bạn xử lý các cửa sổ popup và iframe một cách dễ dàng, giúp kiểm thử giao diện người dùng một cách toàn diên hơn.
- ♣ Việc sử dụng "SeleniumLibrary" và "Browser" không chỉ giúp tự động hóa kiểm thử UI một cách hiệu quả, mà còn tạo điều kiện thuận lợi cho việc quản lý và tương tác với các yếu tố trong trang web.

2.3.3. Một số keywords được dùng phổ biến

Bảng 2.1. Một số keywords được sử dụng phổ biến

Keyword	Mô tả	
Keywords liên quan đến giao diện người dùng (UI)		
Click	Nhấp vào một phần tử trên giao diện.	
Fill Text	Nhập dữ liệu vào trường văn bản.	
Element Text Should Be	Xác nhận văn bản trên giao diện.	
Keywords liên quan đến l	xiểm tra	
Should Be Equal	So sánh hai giá trị để đảm bảo chúng bằng nhau.	
Should Be True	Kiểm tra xem một biểu thức logic có đúng không.	
Should Be False	Kiểm tra xem một biểu thức logic có sai không.	
Keywords liên quan đến c	lữ liệu	
Set Variable	Thiết lập giá trị cho biến.	
Append To List	Thêm giá trị vào danh sách.	
Keywords liên quan đến c	chuỗi văn bản	
Should Contain	Kiểm tra xem một chuỗi có chứa một phần tử con không.	
Should Not Contain	Kiểm tra xem một chuỗi có không chứa một phần tử con.	
Keywords liên quan đến t	hời gian và ngày tháng	
Sleep	Tạm dừng thực thi trong một khoảng thời gian.	
Get Current Date	Lấy ngày hiện tại.	
Keywords liên quan đến vòng lặp và điều kiện		
For	Vòng lặp for.	
Run Keyword If	Thực hiện keyword dựa trên điều kiện.	

Đây chỉ là một số keywords phổ biến trong Robot Framework. Mỗi thư viện hay framework cụ thể sẽ có những keywords riêng phù hợp với tác vụ cụ thể mà bạn đang thực hiện.

2.4. CÚ PHÁP CƠ BẢN CỦA TEST CASE

♣ Định nghĩa cấu trúc cơ bản của một test case trong Robot Framework:

✓ Trong Robot Framework, mỗi test case được viết dưới dạng một tệp .robot riêng biệt. Một test case chứa các thông tin cơ bản như tên, mô tả, thẻ, và các bước kiểm thử cụ thể. Dưới đây là cấu trúc cơ bản của một test case:

```
*** Test Cases ***

Tên Test Case

[Documentation] Mô tả về mục tiêu của test case

[Tags] Thể liên quan đến test case (tùy chọn)

Bước 1

Bước 2
....
```

✓ Cú pháp chi tiết của một bước kiểm thử: Mỗi bước kiểm thử trong test case sẽ được biểu diễn bởi một keyword theo sau bởi các tham số. Cấu trúc chi tiết của một bước kiểm thử như sau:

Tên Keyword Tham số 1 Tham số 2 ...

- Các keywords có sẵn trong Robot Framework: Các keywords có sẵn trong Robot Framework là những khối xây dựng sẵn cho các thao tác kiểm thử. Các keywords này được cung cấp bởi thư viện "SeleniumLibrary" và thư viện "Browser" để giúp bạn tương tác với trình duyệt và các yếu tố trong trang web. Có hàng trăm keywords có sẵn cho các thao tác khác nhau như tìm kiếm, nhập liệu, nhấn nút, và kiểm tra kết quả.
- ✓ Sử dụng các keywords liên quan đến trình duyệt: Các keywords có sẵn liên quan đến trình duyệt cho phép bạn thực hiện các thao tác trên trình duyệt như mở trang web, nhập liệu, click, và kiểm tra kết quả. Phần này đã được giới thiệu ở mục 2.3.3 của chương 2.
- √ Ví dụ thực tế về việc sử dụng cú pháp trong test case:

```
Settings ***
           SeleniumLibrary
Library
*** Test Cases ***
Kiểm tra đẳng nhập hợp lệ
   [Documentation]
                        Kiểm tra việc đăng nhập với thông tin hợp lệ
   Open Browser
                       http://localhost:5173/
                                                  firefox
   Maximize Browser Window
   Input Text
                       //input[@id='username']
                                                  admin_balan@getnada.com
   Input Password
                       //input[@id='password']
                                                    Ari123456#
   Click Button
                       //button[@id='idSubmit']
   Element Text Should Be
                             id=swal2-html-container
                                                        Đăng nhập thành công
```

✓ Mô tả các phần cú pháp:

Tên Test Case: Tên của test case, thể hiện mục đích và chức năng cần kiểm thử.

[**Documentation**]: Mô tả chi tiết về mục đích của test case, giúp người đọc hiểu rõ hơn về mục tiêu kiểm thử.

[Tags]: Các thẻ liên quan đến test case, giúp phân loại và nhận biết test case thuộc nhóm nào (ví dụ: UI, API).

Tên Keyword: Tên của keyword được sử dụng để thực hiện bước kiểm thử cụ thể.

Tham số: Các giá trị hoặc dữ liệu cần thiết để thực hiện bước kiểm thử.

[Tham số tùy chọn]: Các tham số có thể đi kèm với keyword để tùy chỉnh hành động.

✓ Lợi ích của cú pháp cơ bản trong test case:

- Rõ ràng và dễ đọc: Cú pháp cơ bản giúp mã kiểm thử trở nên dễ đọc, dễ hiểu và dễ theo dõi.
- Tổ chức logic: Với cú pháp cơ bản, bạn có thể tổ chức logic kiểm thử thành các bước rõ ràng và có cấu trúc.
- Dễ bảo trì: Cấu trúc chuẩn giúp bạn dễ dàng bổ sung, sửa đổi và duyệt test case trong tương lai.

2.5. TẠO TEST SUITE VÀ TEST CASE HIERACHY

♣ Tổ chức kiểm thử với test suite và test case hierarchy:

✓ Trong quá trình thực hiện kiếm thử tự động bằng Robot Framework, việc tổ chức test suite và test case hierarchy là một phần quan trọng giúp bạn tổ chức,

quản lý và thực hiện kiểm thử một cách hiệu quả. Test suite là một tập hợp các test case có liên quan đến nhau, còn test case hierarchy là cách bạn tổ chức các test suite vào các mức khác nhau, tạo nên một cấu trúc kiểm thử có hệ thống.

4 Tạo test suite và test case hierarchy trong Robot Framework:

- ✓ Test Suite: Một test suite thường tương ứng với một chức năng hoặc một phần trong ứng dụng. Nó có thể chứa một hoặc nhiều test case liên quan đến chức năng đó. Test suite giúp bạn tổ chức các test case có cùng mục tiêu kiểm thử vào cùng một nhóm.
 - Một test suite là một bộ sưu tập các test cases và/hoặc các test suites khác.
 Bạn có thể tạo một test suite bằng cách sử dụng câu lệnh *** Test Cases
 *** trong têp .robot.
- ✓ **Test Case Hierarchy:** Robot Framework cho phép bạn xây dựng test case hierarchy bằng cách tổ chức test suite vào các mức khác nhau. Điều này giúp bạn kiểm thử từng mức độ khác nhau của ứng dụng, từ kiểm thử chức năng cơ bản đến kiểm thử hệ thống tổng thể. Test case hierarchy giúp bạn phân chia và tập trung vào từng khía cạnh cụ thể của ứng dụng.
- ✓ Ví dụ về việc sử dụng test suite và test case hierarchy:

```
Test Setup
Test Teardown
                  Tear Down
*** Variables ***
${email_reset_password}
LG_01: Kiểm tra đăng nhập không thành công khi bỏ trống trường "Tên đăng nhập"
    [Tags] @Test_Login @Test_by_Ho^Văn_Nhật
    When Enter "text" in "Mật khẩu" with "_RANDOM_"
    And Click "Đăng Nhập" button
    Then Required message "Tên đăng nhập" displayed under "This is a required field!" field
    [Tags] @Test_Login @Test_by_Ho^Văn_Nhật
    When Enter "email" in "Tên đăng nhập" with "_RANDOM_"
    And Click "Đăng Nhập" button
    Then Required message "Mật khẩu" displayed under "This is a required field!" field
    Set Browser Timeout
    New Browser
```

Hình 2.3. Ví dụ về việc sử dụng test suite và test case hierarchy

🖶 Lợi ích của test suite và test case hierarchy:

- ✓ *Tổ chức rõ ràng*: Việc sử dụng test suite và test case hierarchy giúp bạn tổ chức kiểm thử một cách có cấu trúc, dễ quản lý và tìm kiếm khi cần.
- ✓ Tái sử dụng dễ dàng: Test suite và test case hierarchy giúp bạn dễ dàng tái sử dụng các test case trong các dự án khác nhau, giảm thiểu công việc viết lại mã kiểm thử.
- ✓ Kiểm thử từng mức độ: Với test case hierarchy, bạn có thể kiểm thử từng mức độ khác nhau của ứng dụng, từ kiểm thử chức năng cơ bản đến kiểm thử hệ thống tổng thể, giúp đảm bảo tính ổn định và chất lượng của ứng dụng.
- ♣ Chạy Test Cases và Test Suites: Bạn có thể chạy các test cases và test suites bằng cách sử dụng lệnh robot từ dòng lệnh hoặc sử dụng các công cụ tích hợp.
 - ✓ Để chạy một test case cụ thể: robot -t "Test Case 1" name_file.robot
 - ✓ Để chạy một test suite cụ thể: robot -t "Test Suite Name" name_file.robot
 - ✓ Để chạy các test cases dựa trên tags: robot -i Smoke name_file.robot

Như vậy, bạn có thể sắp xếp và quản lý bài kiểm tra của mình trong Robot Framework bằng cách tạo test suite và test case hierarchy, sử dụng tags để nhóm chúng, và thực hiện các bài kiểm tra dựa trên các tags và tên của chúng.

2.6. CÁCH BẮT PHẦN TỬ GIAO DIỆN

2.6.1. Tổng quan XPath

- ✓ XPath là một cách để phân tích mã HTML nhằm xác định các yếu tố của một web driver.
- ✓ Là ngôn ngữ hỗ trợ tìm kiếm thông tin trong tài liệu XML qua việc sử dụng biểu thức XPath để định hướng tìm kiếm dữ liệu trên XML thay vì phải thực hiện tìm kiếm đệ qui để duyệt cây XML.
- ✓ Xpath định nghĩa 7 loại nodes theo mô hình thể hiện bên dưới từ root, element, attribute, text, namespace, processing-instruction và comment.
- ✓ Ngoài ra, Xpath còn định nghĩa một số node đặc biệt để thể hiện mối quan hệ giữa các node trong mô hình trong quá trình xử lý như sau:
 - o Parent Node: node trên trực tiếp của node hiện hành.
 - O Child Node: tập node trực tiếp của node hiện hành cấp thấp hơn.
 - O Sibling: node ngang hàng hay cùng cha với node hiện hành.

- O Ancestors: tất cả node con bên trên node hiện hành cùng nhánh.
- O Descendants: tất cả node con bên dưới của node hiện hành cùng nhánh.

2.6.2. Cú pháp của XPath

- ✓ Để truy vấn với đường dẫn tuyệt đối nghĩa là đi từ root của tài liệu XML đến các thành phần cần truy cập, XPath qui định với cú pháp bắt đầu bằng dấu /
- ✓ Để truy vấn với đường dận tương đối để có thể truy cập đến thành phần bất kỳ thỏa điều kiện, XPath qui định cú pháp sử dụng với dấu //
- ✓ Để truy vấn đến một thành phần bất kỳ mà không cần biết tên của nó là gì, XPath
 qui định ký tự sử dụng là *.
- √ Để truy cập thuộc tính của một node, XPath qui định thuộc tính truy vấn phải có cú pháp bắt đầu là @.Ví dụ @tênThuộcTính.
- ✓ Điều kiện khi truy vấn được đặt trong dấu []
- ✓ Truy vấn lựa chọn nodes

Biểu thức	Định nghĩa
tênNode	Chọn tất cả các node con của tênNode.
1	Chọn tất cả các node tính từ root.
//	Chọn tất cả node tính từ node hiện hành.
•	Chọn node hiện hành.
••	Chọn node cha của node hiện hành.

Các phép toán được sử dụng trong XPath:

- √ Đại số: +, -, * (nhân), div (chia thập phân), mod (chia lấy dư)
- ✓ So Sánh hay quan hệ: =, != (khác), <, <=, >, >=
- ✓ Luận lý: true, false, and, or, not
- √ Kết hợp: | (hội)

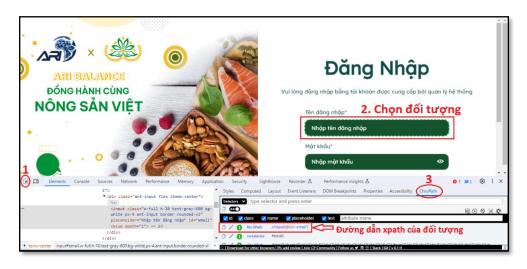
2.6.3. Cách bắt XPath bằng ChroPath

✓ Tải xuống và cài đặt tiện ích mở rộng ChroPath cho trình duyệt Chrome:



Hình 2.4. Cài đặt tiện ích mở rộng ChroPath cho trình duyệt Chrome

✓ Các bước bắt xpath bằng ChroPath:



Hình 2.5. Cách bắt xpath bằng ChroPath

2.7. TỔNG KẾT CHƯƠNG

Chương 2 đã giới thiệu về Robot Framework, một công cụ kiểm thử tự động mã nguồn mở, linh hoạt, và dễ sử dụng. Chương này đề cập đến cách tạo và chạy các test case. Chương này cũng đã trình bày về cấu trúc của một test case, bao gồm khai báo biến, sử dụng keywords, và kiểm tra kết quả. Với cấu trúc đơn giản và dễ đọc, Robot Framework cho phép các nhà phát triển và kiểm thử viên dễ dàng triển khai kiểm thử tự động mà không cần có kiến thức sâu về lập trình.

Thư viện Browser đã được giới thiệu như một công cụ hỗ trợ mạnh mẽ để thực hiện kiểm thử tự động các ứng dụng web. Chương 2 đã trình bày ví dụ về việc sử dụng các keywords của thư viện, qua đó cho thấy rõ sự tiện lợi và hiệu quả của thư viện Browser trong việc kiểm thử tự động.

Chương 3 sẽ tiếp tục trình bày về cách cài đặt và triển khai thực nghiệm công cụ Robot Framework và Appium.

CHƯƠNG III. TIẾN HÀNH CÀI ĐẶT VÀ TRIỂN KHAI THỰC NGHIỆM VỚI CÔNG CỤ ROBOT FRAMEWORK

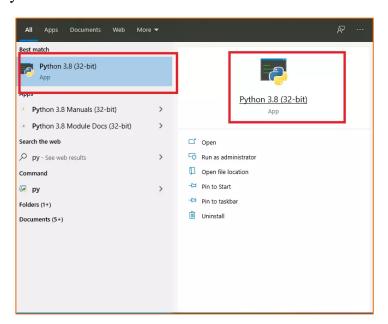
3.1. TIẾN HÀNH CÀI ĐẶT

3.1.1. Cài đặt python và pip

- **♣** Giới thiệu về Python và vai trò của nó trong phát triển và kiểm thử:
 - ✓ Python là một ngôn ngữ lập trình đa năng, mạnh mẽ và dễ học, được sử dụng rộng rãi trong lĩnh vực phát triển phần mềm và kiểm thử.
 - ✓ Python cung cấp cú pháp đơn giản, hỗ trợ nhiều thư viện mạnh mẽ và có khả
 năng tích hợp dễ dàng với các công cụ khác.
 - √ Với khả năng này, Python đã trở thành một trong những ngôn ngữ phổ biến trong việc thực hiện kiểm thử tự động.
- ♣ Cài đặt Python: Trước hết, vì Robot Framework là một nền tảng kiểm thử dựa trên nền tảng Python, nên trước tiên cần cài đặt Python (nên cài Python 2.5 hoặc mới hơn khuyến cáo cài đặt Python 2.7).

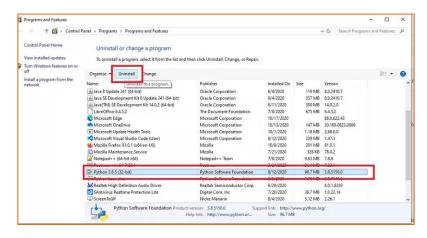
Bước 1. Kiểm tra xem trong máy tính của bạn đã cài đặt Python hay chưa?

Bạn hãy nhấp vào biểu tượng kính lúp/vòng tròn cạnh biểu tượng Windows, hoặc nhấn \boxplus Win+S. Nếu tìm thấy kết quả Python thì chứng tỏ là máy của bạn đã sẵn sàng cài đặt Python rồi đó.

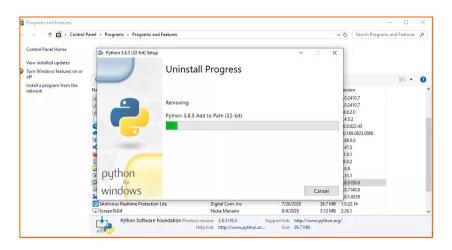


Hình 3.1. Kiểm tra xem Python đã cài hay chưa

Bước 2. Gỡ cài đặt và cài lai

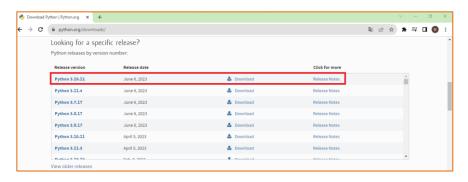


Hình 3.2. Chọn Python 3.8.5 để gỡ cài đặt



Hình 3.3. Quá trình gỡ cài đặt Python bắt đầu

Bước 3. Cài đặt lại. Mở trình duyệt, truy cập vào website có đường dẫn sau để tải phần mềm python: https://www.python.org/



Hình 3.4. Download phiên bản Python cần cài đặt

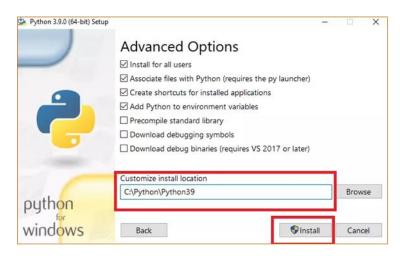
Bắt đầu cài đặt



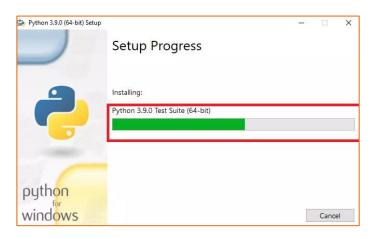
Hình 3.5. Chạy phần mềm python vừa tải về



Hình 3.6. Chọn "Next" để tiếp tục cài đặt



Hình 3.7. Chọn đường dẫn lưu file cài đặt Python và chọn "Install"



Hình 3.8. Bắt đầu cài đặt python

Bước 4. Kiểm tra xem Python và Pip cài đặt xong chưa

```
Administrator Command Prompt

Microsoft Windows [Version 10.0.19044.2006]

(c) MrThuan Diskless 2022. All rights reserved.

C:\Users\Administrator>python --version

Python 3.11.4

C:\Users\Administrator>pip --version

pip 23.1.2 from C:\Program Files\Python311\Lib\site-packages\pip (python 3.11)
```

Hình 3.9. Kiểm tra phiên bản Python và pip đã cài đặt

Hình 3.10. Update phiên bản pip

3.1.2. Cài đặt robot framework và các thư viện liên quan

♣ Giới thiệu về cách cài đặt Robot Framework và các bước cần thiết:

- ✓ Robot Framework là một công cụ kiểm thử tự động mạnh mẽ và linh hoạt.
- ✓ Để sử dụng Robot Framework, bạn cần cài đặt nó cùng với các thư viện hỗ
 trợ.
- ✓ Trong phần này, chúng ta sẽ cài đặt Robot Framework cùng với hai thư viện quan trọng là "FakerLibrary" và "Browser".

♣ Cài đặt Robot Framework và thư viện "FakerLibrary":

Bước 1. Mở Command Prompt hoặc Terminal.

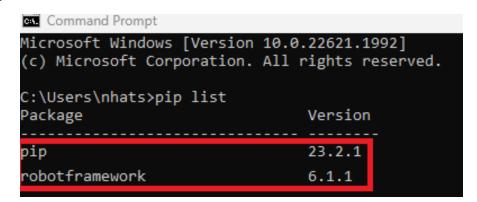


Hình 3.11. Giao diện Command Prompt

Bước 2. Sử dụng pip để cài đặt Robot Framework bằng cách chạy lệnh: "pip install -U robotframework".

Hình 3.12. Cài đặt thành công Robotframework

Bước 3. Kiểm tra xem robotframework đã được cài đặt đúng chưa bằng câu lệnh: "pip list"



Hình 3.13. Kiểm tra xem phiên bản robotframework đã cài đặt

Bước 4. Sau khi cài đặt xong Robot Framework, bạn cần cài đặt thư viện "FakerLibrary" để làm việc với kiểm thử giao diện người dùng (UI) của các trang web. Sử dụng pip để cài đặt thư viện này bằng lệnh: "pip install -U robotframework-faker".

Hình 3.14. Cài đặt thành công thư viện FakerLibrary

```
| Comparison | Com
```

Hình 3.15. Cài đặt thành công thư viện Selenium

♣ Cài đặt thư viện "Browser" cho việc kiểm thử trực tiếp trình duyệt: Thư viện "Browser" là một thư viện mở rộng dựa trên "SeleniumLibrary", cung cấp các keywords linh hoạt hơn để tương tác với trình duyệt và các yếu tố trong trang web. Để cài đặt thư viện "Browser", sử dụng pip bằng lệnh: "pip install -U robotframework-browser".

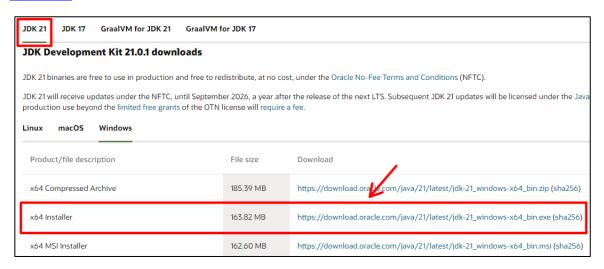
```
(C) Perhan Diskles 2022. All rights reserved,
(C) Perhan Diskles 202
```

Hình 3.16. Cài đặt thành công thư viện Browser

3.1.3. Robot Framework và Appium - Cài đặt môi trường trên Windows

Cài đặt Java JDK và thiết lập môi trường trên Windows:

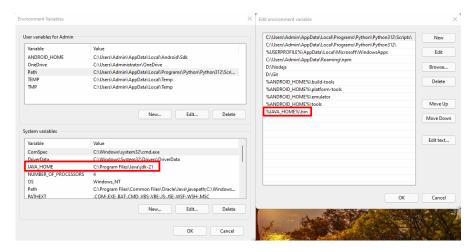
Buốc 1: Truy cập https://www.oracle.com/java/technologies/downloads/#jdk21-windows, tải và cài đặt.



Hình 3.17. Chọn bản Java SDK cần cài đặt

Bước 2. Làm theo hướng dẫn trên màn hình để cài đặt Java JDK mới nhất.

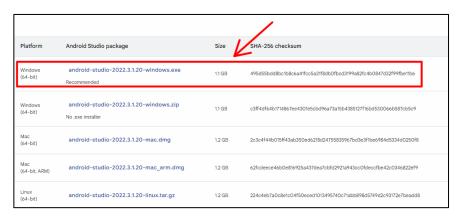
Bước 3. Thiết lập môi trường Java.



Hình 3.18. Cài đặt biến môi trường cho Java JDK

Cài đặt Android Studio:

Bước 1. Truy cập https://developer.android.com/studio, tải và cài đặt.



Hình 3.19. Chọn phiên bản Android Studio cần cài đặt

Bước 2. Mở file đã tải xuống để cài đặt Android Studio.

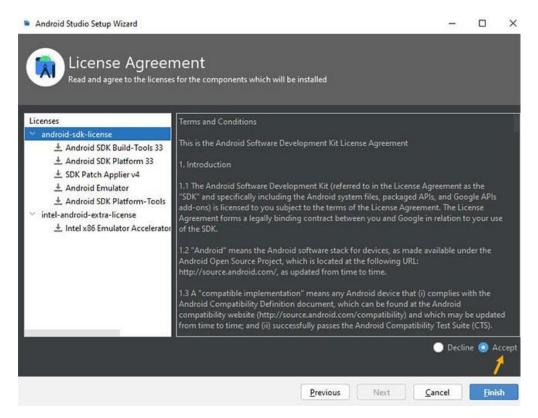
✓ Nhấp vào Next để tiếp tục cài đặt Android Studio.



- ✓ Ngoài ra, hãy nhấp vào Install và đợi trong khi Android Studio được cài đặt trên PC của ban.
- ✓ Sau khi hoàn tất, hãy nhấp vào Finish để đóng trình cài đặt và khởi động Android Studio.
- ✓ Sau khi hoàn tất, hãy nhấp vào Finish để đóng trình cài đặt và khởi động Android Studio.

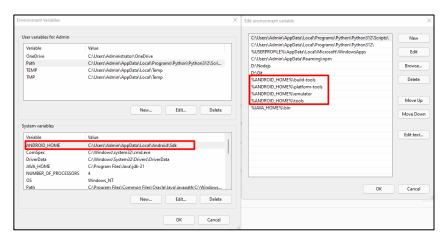
Bước 3. Thiết lập Android Studio trên Windows

- ✓ Trong cửa sổ Setup Wizard, chọn tùy chọn Standard để áp dụng các cài đặt được đề xuất và nhấp vào Next.
- ✓ Tiếp theo, chọn một theme bạn muốn và nhấp vào Next một lần nữa để xác minh cài đặt ban đã cấu hình.
- ✓ Trên màn hình tiếp theo, chọn Accept và nhấp vào Finish để áp dụng các thay đổi.



- ✓ Chờ cho đến khi Setup Wizard tải xuống một số file cần thiết. Có thể mất khoảng 15 đến 30 phút, tùy thuộc vào tốc độ mạng của bạn.
- ✓ Sau khi quá trình tải xuống hoàn tất, hãy nhấp vào Finish để hoàn tất quá trình thiết lập.

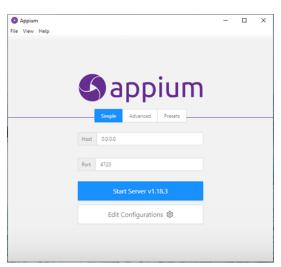
Bước 4. Cài đặt biến môi trường cho Android Studio



Hình 3.20. Thiết lập biến môi trường cho Android Studio

Cài đặt Appium:

- ✓ Chuyển đến trang http://appium.io
- ✓ Click nút "Download" và chọn tải file phù hợp với máy của mình.
- ✓ Sau khi tải hoàn thành tiến hành cài đặt bình thường, và đây là giao diện Appium

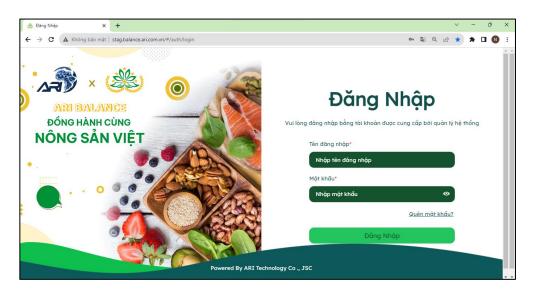


Hình 3.21. Giao diện Appium

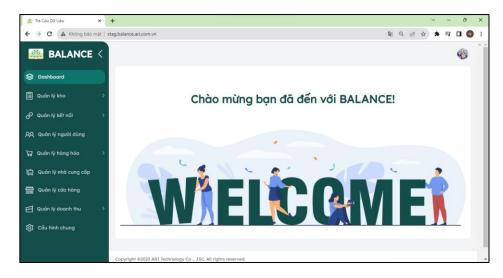
3.2. TRIỂN KHAI THỰC NGHIỆM

3.2.1. Giới thiệu về website mẫu

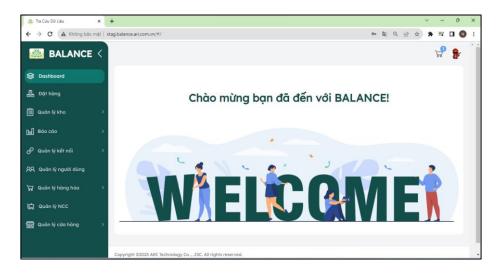
Trong phần này, tôi sẽ giới thiệu về website mẫu mà tôi sẽ sử dụng để thực hiện kiểm thử tự động bằng Robot Framework. Website mẫu này sẽ đóng vai trò là một ứng dụng web cơ bản để chúng tôi thực hiện các bài kiểm thử và trình bày cách sử dụng Robot Framework để kiểm thử tự động.



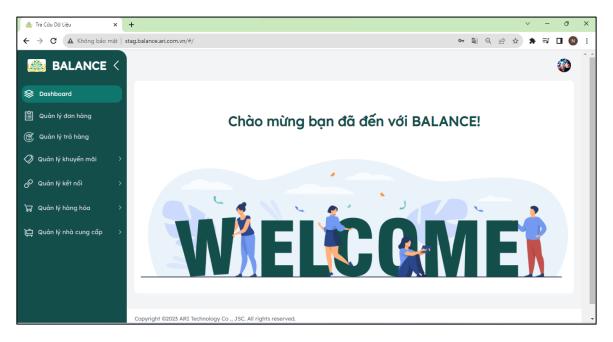
Hình 3.22. Giao diện Website mẫu trang đăng nhập



Hình 3.23. Giao diện và menu trang chủ tài khoản Admin



Hình 3.24. Giao diện và menu trang chủ tài khoản Store



Hình 3.25. Giao diện và menu trang chủ tài khoản Supplier

♣ Mục tiêu của website mẫu:

- ✓ Mô phỏng ứng dụng web cơ bản: Website mẫu được thiết kế để mô phỏng một ứng dụng web cơ bản với các tính năng cơ bản như đăng nhập, tìm kiếm sản phẩm, và hiển thị thông tin.
- ✓ Để dàng sử dụng: Giao diện của website mẫu được thiết kế đơn giản và dễ sử dụng, giúp người dùng nhanh chóng làm quen và tương tác với các tính năng.
- ✓ Đa dạng chức năng: Website mẫu cung cấp các chức năng chính như đăng nhập, tìm kiếm sản phẩm, và hiển thị thông tin sản phẩm, cho phép chúng tôi thực hiện các kiểm thử cơ bản và nâng cao.

♣ Cấu trúc và tính năng của website mẫu:

- ✓ Trang chủ: Trang chủ hiển thị thông tin tổng quan về ứng dụng và các tính năng chính khi đăng nhập thành công. Úng với mỗi tài khoản khác nhau, sẽ có giao diện trang chủ với các mục menu khác nhau.
- ✓ Đăng nhập: Trang đăng nhập cho phép người dùng nhập tên người dùng và mật khẩu để đăng nhập vào hệ thống.
- ✓ Quên mật khẩu: Ngoài ra, website mẫu cũng có chức năng cho phép người dùng lấy lại mật khẩu.

3.2.2. Xây dựng các Testcases

♣ Testcase chức năng đăng nhập:

Testcase ID	Summary	Preconditions	Test Step	Expected Result
LOGIN				
erify vali	idation text			
LG_01	Kiểm tra đăng nhập không thành công khi bỏ trống trường " Tên đăng nhập "		1. Enter " text " in " Mật khẩu " with " _RANDOM_ " 2. Click " Đăng nhập" button	Show validation text at "Tên đăng nhập" field: "This is a required field!"
LG_02	Kiểm tra đăng nhập không thành công khi bỏ trống trường " Mật khẩu "	Go to:	Enter "email" in "Tên đăng nhập" with "_RANDOM_" Click "Đăng nhập" button	Show validation text at "Mậ khẩu" field: "This is a required field!"
LG_03	Kiểm tra đẳng nhập không thành công khi bỏ trống cả "Tên đẳng nhập" và "Mật khẩu"	http://stag.balance.ari.c om.vn/#/	1. Click "Đăng nhập" button	Show validation text at "Têr đăng nhập" and "Mật khẩu field: "This is a required field!"
LG_04	Kiểm tra đăng nhập không thành công khi nhập " Tên đăng nhập " sai định dạng email		1. Enter "text" in "Tên đăng nhập" with "_RANDOM_" 2. Enter "text" in "Mật khẩu" with "\${empty}"	Show validation text at "Têr đăng nhập" field: "Please enter a valid email address!"
erify erro	or message			
LG_05	Kiểm tra đăng nhập không thành công khi nhập sai " Tên đăng nhập"		1. Enter "email" in "Tên đăng nhập" with "_RANDOM_" 2. Enter "text" in "Mật khẩu" with "Ari123456#" 3. Click "Đăng nhập" button	- Display pop-up with error message "Thất bại" - Show error message: "Tài khoản chưa được tạo hoặc chưa được kích hoạt."
LG_06	Kiểm tra đăng nhập không thành công khi nhập sai "Mật khẩu"	Go to:	 Enter "email" in "Tên đăng nhập" with "admin_balan@getnada.com" Enter "test name" in "Mật khẩu" with "_RANDOM_" Click "Đăng nhập" button 	- Display pop-up with error message "Thất bại" - Show error message: "Tài khoản hoặc mật khẩu khôn, đúng, vui lòng thử lại"
LG_07	Kiểm tra đăng nhập không thành công khi nhập sai cả "Tên đăng nhập" và "Mật khẩu"	http://stag.balance.ari.c om.vn/#/	1. Enter "email" in "Tên đăng nhập" with "_RANDOM_" 2. Enter "text" in "Mật khẩu" with "_RANDOM_" 3. Click "Đăng nhập" button	- Display pop-up with error message "Thất bại" - Show error message: ""Tà khoản chưa được tạo hoặc chưa được kích hoạt.""
LG_08	Kiểm tra đăng nhập không thành công khi tài khoản đã bị hủy kích hoạt		1. Enter "email" in "Tên đăng nhập" with "lehuyen@getnada.com" 2. Enter "text" in "Mật khẩu" with "_RANDOM_" 3. Click "Đăng nhập" button	- Display pop-up with error message "Thất bại" - Show error message: "Tài khoản chưa được tạo hoặc chưa được kích hoạt."
Verify Sign	n in successfuly			
LG_09	Xác minh đăng nhập thành công tài khoản "Admin"		1. Enter "email" in "Tên đăng nhập" with "test_admin@gmail.com" 2. Enter "text" in "Mật khẩu" with "Nhat@01101999" 3. Click "Đăng nhập" button	- Đăng nhập thành công tài khoản Admin
LG_10	Xác minh đăng nhập thành công tài khoản "Store"	Go to: http://stag.balance.ari.c	1. Enter "email" in "Tên đăng nhập" with "test_store@gmail.com" 2. Enter "text" in "Mật khẩu" with "Nhat@01101999" 3. Click "Đăng nhập" button	- Đăng nhập thành công tài khoản Store
LG_11	Xác minh đăng nhập thành công tài khoản "Supplier"	om.vn/#/	1. Enter "email" in "Tên đăng nhập" with "test_supplier@gmail.com" 2. Enter "text" in "Mật khẩu" with "Nhat@01101999" 3. Click "Đăng nhập" button	- Đăng nhập thành công tài khoản Supplier
LG_12	Xác minh đăng nhập thành công khi dùng phím Enter		1. Enter "email" in "Tên đăng nhập" with "test_admin@gmail.com" 2. Enter "text" in "Mật khẩu" with "Nhat@01101999" 3. Click "Đăng nhập" button	- Đăng nhập thành công tài khoản Admin
erify disp	plays the password			
LG_13	Xác minh rằng có thể hiển thị mật khẩu dưới dạng text	Go to: http://stag.balance.ari.c om.vn/#/	1. Enter " text " in "Mật khẩu" with "_RANDOM_" 2. Click "Eye" icon to display password	Passwords are displayed as characters
erify refr	resh page			
LG_14	Xác minh rằng thông tin đã nhập sẽ không còn khi reload page	Go to: http://stag.balance.ari.c om.vn/#/	1. Enter "email" in "Tên đăng nhập" with "_RANDOM_" 2. Enter "text" in "Mật khấu" with "_RANDOM_" 3. Reload page with F5 key	The page is refreshed with empty fields
	i cioau page	OHI.VIIIπ/	J. Neloda page with 1 3 key	I

Testcase ID	Summary	Preconditions	Test Step	Expected Result
FORGOT P	ASSWORD			
Verify nav	igate to Forgot password pa	ge		
LG_15	Xác minh rằng có thể điều hướng đến trang "Quên mật khẩu" từ trang "Đăng nhập"	Go to: http://stag.balance.ari.c om.vn/#/	1. Click "Quên mật khẩu?" link	Navigate to Forgot Password page
Verify Vali	idation Text Forgot password	l page		
LG_16	Xác minh Validation text khi bỏ trống trường "Email"	Go to: http://stag.balance.ari.c	1. Click "Lấy mã OTP" button	Show validation text at "Email khôi phục" field: "This is a required field!"
LG_17	Xác minh Validation text khi nhập "Email" sai định dạng	om.vn/#/ Navigate to Forgot Password page	1. Enter "text" in "Email khôi phục" with "_RANDOM_" 2. Click "Lấy mã OTP" button	Show validation text at "Email khôi phục" field: "Please enter a valid email address!"
erify Erro	or Message Forgot password	page		
LG_18	Xác minh thông báo lỗi khi nhập email không tồn tại trên hệ thống	Go to: http://stag.balance.ari.c om.vn/#/ Navigate to Forgot Password page	1. Enter "email" in "Email khôi phục" with "_RANDOM_" 2. Click "Lấy mã OTP" button	- Display pop-up with error message "Thất bại" - Show error message: "Địa chỉ email không tồn tại, vui lòng thử địa chỉ email khác."
Forgot Pas	ssword Successfull			
LG_19	Xác minh quên mật khẩu thành công	Go to: http://stag.balance.ari.com.vn/#/ Navigate to Forgot Password page	1. Enter "email" in "Email khôi phục" with "hoangdieu1810@gmail.com" 2. Click "Lấy mã OTP" button	Navigate to Enter OTP page
ОТР				
Validation	text			
LG_20	Xác minh Validation text trường "Mã OTP" khi không nhập mã OTP	Go to: http://stag.balance.ari.c om.vn/#/ Navigate to enter OTP page	1. Click "Gửi OTP" button	Show validation text at "Mã OTP" field: "This is a required field!"
Error mess	sage			
LG_21	Xác minh hiển thị thông báo lỗi khi nhập sai mã OTP	Go to: http://stag.balance.ari.c om.vn/#/ Navigate to enter OTP page	1. Enter " otp " in " Mã OTP" with "_ RANDOM_ " 2. Click " Gửi OTP " button	- Display pop-up with error message "Thất bại" - Show error message: "Mã OTP sai hoặc hết thời hạn sử dụng."
Success M	lessage			
LG_22	Xác minh điều hướng đến trang Reset password khi nhập đúng mã OTP	Go to: http://stag.balance.ari.c om.vn/#/ Navigate to enter OTP page	1. Enter " otp " in " Mã OTP" with " 123456 " 2. Click " Gửi OTP" button	Navigate to Change password
Reset pas	word page	. 5-		
/alidation	text			
LG_23	Xác minh Validation text khi không nhập cả " Mật khẩu " và " Xác nhận mật khẩu"		1. Click "Đổi mật khẩu" button	Show validation text at "Mậ khẩu" field and "Xác nhận mật khẩu" field: "This is a required field!"
LG_24	Xác minh Validation text khi không nhập "Mật khẩu"		Enter "password" in "Mật khẩu" with "\${empty}" Click in "Xác nhận mật khẩu" field	Show validation text at "Mật khẩu" field: "This is a required field!"
LG_25	Xác minh Validation text khi nhập "Mật khẩu" nhỏ hơn 8 ký tự		Enter "7 character" in "Mật khẩu" with "_RANDOM_" Click in "Xác nhận mật khẩu" field	Show validation text at "Mật khẩu" field: "Mật khẩu yêu cầu có 8 ký tự trở lên"

Testcase ID	Summary	Preconditions	Test Step	Expected Result
LG_26	Xác minh Validation text khi nhập "Mật khẩu" chứa khoảng trắng	Go to: http://stag.balance.ari.c om.vn/#/	1. Enter "text" in "Mật khẩu" with "_RANDOM_" 2. Click in "Xác nhận mật khẩu" field	Show validation text at "Mật khẩu" field: "Mật khẩu không được có khoảng trắng"
LG_27	Xác minh Validation text khi nhập "Mật khẩu" chưa đúng yêu cầu về bảo mật	Navigate to Reset Password page	Enter "number" in "Mật khẩu" with "_RANDOM_" Click in "Xác nhận mật khẩu" field	Show validation text at "Mật khẩu" field: "Mật khẩu yêu cầu có 8 ký tự trở lên, có ít nhất 1 chữ hoa, 1 chữ thường, 1 chữ số và 1 kí tự đặc biệt"
LG_28	Xác minh Validation text khi không nhập "Xác nhận mật khẩu"		Enter "retype password" in "Xác nhận mật khẩu" with "\${empty}" Click in "Mật khẩu" field	Show validation text at "Xác nhận mật khẩu" field: "This is a required field!"
LG_29	Xác minh Validation text khi nhập "Xác nhận mật khẩu" không chính xác		1. Enter "password" in "Mật khẩu" with "_RANDOM_" 2. Enter "password" in "Xác nhận mật khẩu" with "_RANDOM_" 3. Click "Đổi mật khẩu" button	Show validation text at "Mật khẩu" field and "Xác nhận mật khẩu" field: "Xác nhận mật khẩu không chính xác"
Change pa	ssword successfull			
LG_30	Xác minh message successfull khi thay đổi mật khẩu thành công	Go to: http://stag.balance.ari.c om.vn/#/ Navigate to Reset Password page	1. Enter "password" in "Mật khẩu" with "Nhat@01101999" 2. Enter "password" in "Xác nhận mật khẩu" with "Nhat@01101999" 3. Click "Đổi mật khẩu" button	- Display pop-up with successful message "Thành công" - Show error message: "Mã OTP sai hoặc hết thời hạn sử dụng." - Navigate to Login page
Display pa	ssword and retype password	I		
LG_31	Xác minh có thể hiển thị mật khẩu và xác nhận mật khẩu dưới dạng text	Go to: http://stag.balance.ari.com.vn/#/ Navigate to Reset Password page	Enter "password" in "Mật khẩu" with "_RANDOM_" Enter "password" in "Xác nhận mật khẩu" with "_RANDOM_" Glick "Eye" icon to display password when reset password	User look "Mật khẩu" field and "Xác nhận mật khẩu" field with type "text"
Back to Lo	gin page			
LG_32	Xác minh quay trở lại thành công từ trang "Quên mật khẩu"	Go to: http://stag.balance.ari.c om.vn/#/ Navigate to Forgot Password page	1. Click link "Quay trở lại Đăng nhập"	Navigate to Login page
LG_33	Xác minh quay trở lại thành công từ trang "Nhập mã OTP"	Go to: http://stag.balance.ari.c om.vn/#/ Navigate to enter OTP page	1. Click link "Quay trở lại Đăng nhập"	Navigate to Login page

Testcases chức năng cập nhập thông tin cá nhân:

Testcase ID	Summary	Preconditions	Test Step	Expected Result				
Update Profile								
Navigate to update profile page								
PRO_01	Xác minh rằng có thể điều hướng tông thành công thiệt thông tin cá nhân thành công Gô to: http://stag.balance.ari.com.vn/#/1 Login Account need to update profile		1. Click icon Avatar 2. Click "Thông tin cá nhân" button	- Navigate to update profile page successfully				
Verify Vali	idation Text							
PRO_02	Xác minh Validation text khi bỏ trống trường " Họ và tên "		1.Enter "text" in "Ho và tên" with "S{empty}" 2. Click in "Email" field	- Show validation text at "Họ và tên" field: "Xin vui lòng nhập tên người dùng"				
PRO_03	Xác minh Validation text trường " Họ và tên" khi nhập họ và tên không hợp lệ		1. Enter "text" in "Ho và tên" with "_RANDOM_" 2. Click in "Email" field	- Show validation text at "Họ và tên" field: "Xin vui lòng chỉ nhập chữ"				
PRO_04	Xác minh Validation text trường "Email" khi không nhập Email		1. Enter "email" in "Email" with "\${empty}" 2. Click in "Họ và tên" field	- Show validation text at "Email" field: "Xin vui lòng nhập email"				
PRO_05	Xác minh Validation text trường "Email" khi nhập sai định dạng Email		1. Enter "email" in "Email" with "_RANDOM_" 2. Click in "Họ và tên" field	- Show validation text at "Email" field: "Vui lòng nhập địa chỉ email hợp lệ!"				
PRO_06	Xác minh Validation text trường "Số điện thoại" khi không nhập Số điện thoại		1. Enter "phone" in "Số điện thoại" with "\${empty}" 2. Click in "Họ và tên" field	- Show validation text at "Số điện thoại" field: "Xin vui lòng nhập số điện thoại người dùng"				

Testcase ID	Summary	Preconditions	Test Step	Expected Result		
PRO_07	Xác minh Validation text trường "Số điện thoại" khi nhập sai định dạng số điện thoại và nhỏ hơn 8 ký tự	Go to: http://stag.balance.ari.com.vn/#/	1. Enter "text" in "Số điện thoại" with "abcdef" 2. Click in "Họ và tên" field	- Show validation text at "Sổ điện thoại" field: "Xin vui lòng chỉ nhập số" - Show validation text at "Sổ điện thoại" field: "Xin vui lòng nhập tối thiễu 8 ký tự số"		
PRO_08	Xác minh Validation text trường "Số điện thoại" khi nhập sai định dạng số điện thoại và lớn hơn 12 ký tự	Navigate to update profile page	1. Enter "text" in "Số điện thoại" with "_RANDOM_" 2. Click in "Họ và tên" field	- Show validation text at "Số điện thoại" field: "Xin vui lòng chỉ nhập số" - Show validation text at "Số điện thoại" field: "Xin vui lòng nhập tối đa 12 ký tự số"		
PRO_09	Xác minh Validation text trường "Số điện thoại" khi nhập Số điện thoại nhỏ hơn 8 ký tự		1. Enter "phone_7" in "Số điện thoại" with "\${empty}" 2. Click in "Họ và tên" field	- Show validation text at "Số điện thoại" field: "Xin vui lòng nhập tối thiểu 8 ký tự số"		
PRO_10	Xác minh Validation text trường "Số điện thoại" khi nhập Số điện thoại lớn hơn 12 ký tự		1. Enter "phone_invalid" in "Số điện thoại" with "\${empty}" 2. Click in "Họ và tên" field	- Show validation text at "Số điện thoại" field: "Xin vui lòng nhập tối đa 12 ký tự số"		
PRO_11	Xác minh Validation text turờng "Số điện thoại" khi nhập sai định dạng Số điện thoại và lớn hơn 8 ký tự nhưng nhỏ hơn 12 ký tự		1. Enter "full name" in "Số điện thoại" with "_RANDOM_" 2. Click in "Họ và tên" field	- Show validation text at "Số điện thoại" field: "Xin vui lòng chỉ nhập số"		
PRO_12	Xác minh Validation text trường "Ghi chú" khi nhập quá 500 kí tự		1. Enter "word" in textarea "Ghi chú" with "_RANDOM_" 2. Click in "Họ và tên" field	- Show validation text at "Ghi chú" field: "Xin vui lòng chỉ nhập số"		
Verify erro	or message					
PRO_13	Xác minh hiển thị thông báo lỗi khi chinh sửa thông tin cá nhân với email đã tồn tại trên hệ thống	Go to: http://stag.balance.ari.com.vn/#/	1. Enter "email" in "Email" with "conheohong@mailinator.com" 3. Click "Luu" button	- Show error message: "Số điện thoại hoặc email đã tồn tại trong hệ thống."		
PRO_14	Xác minh hiển thị thông báo lỗi khi chính sửa thông tin cá nhân với email đã tồn tại trên hệ thống	Navigate to update profile page				
Verify Upda	nte Profile Successfuly					
PRO_15	Xác minh cập nhật thông tin cá nhân thành công khi thay đổi " Họ và tên "		1. Enter "fullname" in "Họ và tên" with "_RANDOM_"	- Show message successfull: "" popup		
PRO_16	Xác minh cập nhật thông tin cá nhân thành công khi thay đổi "Email"		1. Enter "email" in "Email" with "_RANDOM_"	- Show message successfull: "Bạn đã thay đổi thông tin thành công, Vui lòng đăng nhập lại !" popup		
PRO_17	Xác minh cập nhật thông tin cá nhân thành công khi thay đổi "Số điện thoại"	Go to:	1. Enter "phone" in "Số điện thoại" with "_RANDOM_"	- Show message successfull: "" popup		
PRO_18	Xác minh cập nhật thông tin cá nhân thành công khi thay đổi " Ghi chú "	http://stag.balance.ari.com.vn/#/ Navigate to update profile page	1. Enter "text" in textarea "Ghi chú" with "_RANDOM_"	- Show message successfull: "" popup		
PRO_19	Xác minh có thể hủy thao tác thành công khi thay đổi thông tin cá nhân		1. Enter "fullname" in "Ho và tên" with "_RANDOM_" 2. Enter "email" in "Email" with "_RANDOM_" 3. Enter "phone" in "Số điện thoại" with "_RANDOM_" 4. Enter "text" in textarea "Ghi chú" with "_RANDOM_" 5. Click on "Hủy thao tác" button	- Value text in "Họ và tên" field: "User" - Value text in "Email" field: "user_test_3@gmail.com" - Value text in "Số điện thoại" field: "68686868" - Value text in "Ghi chú" field: "Đây là user"		
CHANGE P	ASSWORD					
Verify Vali	dation Text					
PRO_20	Xác minh Validation text khí không nhập thông tin cả		1. Click "Đổi mật khẩu" button	- Show validation text at "Mật khẩu hiện tại" field: "Đây là trường bắt buộc!" - Show validation text at "Mật khẩu mới" field: "Đây là trưởng bắt buộc!" - Show validation text at "Xác nhận mật khẩu" field: "Đây là trường bắt buộc!"		
	Xác minh Validation text trường " Mật khẩu mới" khi nhập mật khẩu mới nhỏ hơn 8 ký tự		1. Enter "phone_7" in "Mật khẩu mới" with "_RANDOM_" 2. Click in "Mật khẩu hiện tại" field	- Show validation text at "Mật khẩu mới" field: "Mật khẩu yêu cầu có 8 ký tự trở lên"		
PRO_22	Xác minh Validation text trường " Mật khẩu mới" khi nhập mật khẩu mới chứa khoảng trắng	Go to:	1. Enter "text" in "Mật khẩu mới" with "_RANDOM_" 2. Click in "Mật khẩu hiện tại" field	- Show validation text at "Mật khẩu mới" field: "Mật khẩu không được có khoảng trắng"		
PRO_23	Xác minh Validation text trường "Mật khẩu mới" khi nhập mật khẩu mới không đủ độ bảo mật	http://stag.balance.ari.com.vn/#/ Navigate to change password page	1. Enter "text" in "Mật khẩu mới" with "hovannhat" 2. Click in "Mật khẩu hiện tại" field	- Show validation text at "Mật khẩu mới" field: "Mật khẩu yêu cầu có 8 ký tự trở lên, có ít nhất 1 chữ hoa, 1 chữ thường, 1 chữ số và 1 ki tự đặc biệt"		
PRO_24	Xác minh Validation text trường "Xác nhận mật khẩu" khi nhập xác nhận mật khẩu chứa khoảng trắng		1. Enter "text" in "Xác nhận mật khẩu" with "_RANDOM_" 2. Click in "Mật khẩu hiện tại" field	- Show validation text at "Xác nhận mật khẩu" field: "Mật khẩu không được có khoảng trắng"		

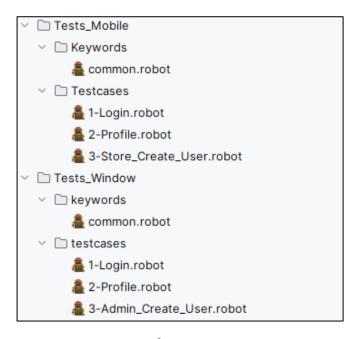
Testcase ID	Summary	Preconditions	Test Step	Expected Result
PRO_25	Xác minh Validation text trường "Xác nhận mật khẩu" khi nhập xác nhận mật khẩu không đủ độ bảo mật		1. Enter "text" in "Xác nhận mật khẩu" with "hovannhat" k 2. Click in "Mật khẩu hiện tại" field th th	
PRO_26	Xác minh Validation text trường "Xác nhận mật khẩu" khi nhập xác nhận mật khẩu không chính xác		1. Enter "text" in "Mật khẩu mới" with "Nhat@01101999" 2. Enter "text" in "Xác nhận mật khẩu" with "Hovannhat@01101999" 3. Click in "Mật khẩu hiện tại" field	- Show validation text at "Xác nhận mật khẩu" field: "Xác nhận mật khẩu không chính xác"
Verify Erro	or Message			
PRO_27	Xác minh hiển thị thông báo lỗi khi nhập "Mật khẩu hiện tại" không chính xác	p "Mật khấu hiện tại" không 2. Enter "password" in "Mặt khau mơn" with "Nhat@011011999" 3. Enter "passward" in "Nặt nhận mặt thiệm with "Nhat@011011900"		- Show display popup: "Thất bại" - Show error message: "Mật khẩu nhập không đúng."
PRO_28	Xác minh hiển thị thông báo lỗi khi nhập mật khẩu hiện tại và mật khẩu mới trùng nhau	Navigate to change password page	1. Enter "password" in "Mật khẩu hiện tại" with "Nhat@01101999" 2. Enter "password" in "Mật khẩu mới" with "Nhat@01101999" 3. Enter "password" in "Xác nhận mật khẩu" with "Nhat@01101999" 4. Click "Đổi mật khẩu" button	- Show display popup: "Thất bại" - Show error message: "Mật khẩu trùng với mật khẩu hiện tại."
Verify Cha	nge Password Successfully			
PRO_29	Xác minh hiển thị thông báo thay đổi mật khẩu thành công	Go to: http://stag.balance.ari.com.vn/#/	1. Enter "password" in "Mật khẩu hiện tại" with "Nhat@01101999" 2. Enter "password" in "Mật khẩu mới" with "Nhat@011019999" 3. Enter "password" in "Xác nhận mật khẩu" with "Nhat@011019999" 4. Click "Đổi mật khẩu" button	- Show display popup: "Thành công" - Show successfull message popup: "Đổi mật khẩu thành công!"
PRO_30	Xác minh có thể hủy thao tác thành công khi thay đổi mật khẩu	Navigate to change password page	1. Enter "password" in "Mật khẩu hiện tại" with "_RANDOM_" 2. Enter "password" in "Mật khẩu mới" with "_RANDOM_" 3. Enter "password" in "Xác nhận mật khẩu" with "_RANDOM_" 4. Click "Hủy thao tác" button	- All field empty
Display Al	l Password			
PRO_31	Xác minh có thể hển thị mật khẩu cả 3 trường đưới dạng text	Go to: http://stag.balance.ari.com.vn/#/ Navigate to Change Password page	1. Enter "password" in "Mật khẩu hiện tại" with "_RANDOM_" 2. Enter "password" in "Mật khẩu mới" with "_RANDOM_" 3. Enter "password" in "Xác nhận mật khẩu" with "_RANDOM_" 4. Click all icon "eye" to display all password	- All password display with "text"

♣ Xem đầy đủ và chi tiết từng module testcase tại đường link sau:

https://github.com/hovannhatautotest/DO-AN-TOT-NGHIEP.git

3.2.3. Xây dựng kịch bản kiểm thử bằng Robot Framework dựa trên Testcase đã xây dựng

> Cấu trúc thư mục kiểm thử tự động:



Hình 3.26. Cấu trúc cây thư mục

> Kịch bản kiểm thử chức năng đăng nhập trên Windows:

```
*** Settings ***
Resource
              ../keywords/common.robot
Test Setup
               Setup
Test Teardown
                 Tear Down
*** Variables ***
${email_reset_password} hoangdieu1810@gmail.com
${email khôi phục}
                       routes.auth.reset-password.Recovery Email
*** Test Cases ***
LG 01: Kiểm tra validation text khi bỏ trống trường "Tên đăng nhập"
    [Tags] @Test Login @Test by Hồ Văn Nhật
    [Documentation] Kiểm tra validation text khi bỏ trống trường "Tên đăng nhập"
    When Enter "text" in "Mật khẩu" with "_RANDOM_"
    And Click "Đăng Nhập" button
    Then Required message "Tên đăng nhập" displayed under "This is a required field!" field
LG_02: Kiểm tra validation text khi bỏ trống trường "Mật khẩu"
    [Tags] @Test_Login @Test by Hồ Văn Nhât
    [Documentation] Kiểm tra validation text bỏ trống trường "Mật khẩu"
    When Enter "email" in "Tên đăng nhập" with "_RANDOM_"
    And Click "Đăng Nhập" button
    Then Required message "Mật khẩu" displayed under "This is a required field!" field
LG 03: Kiểm tra validation text khi bỏ trống cả "Tên đăng nhập" và "Mật khẩu"
    [Tags] @Test Login @Test by Hồ Văn Nhật
    [Documentation] Kiểm tra validation text khi bỏ trống cả "Tên đăng nhập" và "Mật khẩu"
    When Click "Đăng Nhập" button
    Then Required message "Tên đăng nhập" displayed under "This is a required field!" field
    And Required message "Mật khẩu" displayed under "This is a required field!" field
LG 04: Kiểm tra validation text khi nhập "Tên đăng nhập" sai định dạng email
    [Tags] @Test Login @Test by Hồ Văn Nhật
    [Documentation] Kiểm tra validation text khi nhập "Tên đăng nhập" sai định dạng email
    When Enter "text" in "Tên đăng nhập" with "_RANDOM_"
    And Enter "text" in "Mật khẩu" with "${empty}"
    Then Required message "Tên đăng nhập" displayed under "Please enter a valid email address!" field
LG 05: Kiểm tra hiển thị thông báo lỗi khi nhập sai "Tên đăng nhập"
    [Tags] @Test Login @Test by Hồ Văn Nhật
    [Documentation] Kiểm tra hiển thị thông báo lỗi khi nhập sai "Tên đăng nhập"
    When Enter "email" in "Tên đăng nhập" with "_RANDOM_"
    And Enter "text" in "Mât khẩu" with "Ari123456#"
    And Click "Đăng Nhập" button
    Then User look message "Tài khoản chưa được tạo hoặc chưa được kích hoạt." popup
LG 06: Kiểm tra hiển thị thông báo lỗi khi nhập sai "Mật khẩu"
```

[Tags] @Test Login @Test by Hồ Văn Nhật [Documentation] Kiểm tra đăng nhập không thành công khi nhập sai "Mật khẩu" When Enter "email" in "Tên đăng nhập" with "admin_balan@getnada.com" And Enter "password" in "Mât khẩu" with " RANDOM " And Click "Đăng Nhập" button Then User look message "Tài khoản hoặc mật khẩu không đúng, vui lòng thử lại" popup LG 07: Kiểm tra hiển thị thông báo lỗi khi nhập sai cả Tên đăng nhập và Mật khẩu [Tags] @Test Login @Test by Hồ Văn Nhât [Documentation] Kiểm tra hiển thị thông báo lỗi khi nhập sai cả Tên đăng nhập và Mật khẩu When Enter "email" in "Tên đăng nhập" with "_RANDOM_" And Enter "text" in "Mật khẩu" with "_RANDOM_" And Click "Đăng Nhâp" button Then User look message "Tài khoản chưa được tạo hoặc chưa được kích hoạt." popup LG 08: Kiểm tra hiển thị thông báo lỗi khi tài khoản đã bị hủy kích hoạt [Tags] @Test Login @Test by Hồ Văn Nhật [Documentation] Kiểm tra hiển thị thông báo lỗi khi tài khoản đã bị hủy kích hoạt When Enter "email" in "Tên đăng nhập" with "lehuyen@getnada.com" And Enter "text" in "Mât khẩu" with " RANDOM " And Click "Đăng Nhâp" button Then User look message "Tài khoản bi hủy kích hoat, vui lòng liên hệ quản tri viên để giải quyết. Xin lỗi vì sư bất tiên này." popup LG 09: Kiểm tra đăng nhập thành công tài khoản Admin [Tags] @Test Login @Test by Hồ Văn Nhật [Documentation] Kiểm tra đăng nhập thành công tài khoản Admin When Login to Admin Then User look title "Tra Cứu Dữ Liêu" And User look dashboard "BALANCE" And Check Welcome Message Text Is "Chào mừng bạn đã đến với BALANCE!" LG 10: Kiểm tra đăng nhập thành công tài khoản Store [Tags] @Test Login @Test by Hồ Văn Nhật [Documentation] Kiểm tra đăng nhập thành công tài khoản Store When Login to Store Then User look title "Tra Cứu Dữ Liêu" And User look dashboard "BALANCE" And Check Welcome Message Text Is "Chào mừng bạn đã đến với BALANCE!" LG 11: Kiểm tra đăng nhập thành công tài khoản Supplier [Tags] @Test Login @Test by Hồ Văn Nhật [Documentation] Kiểm tra đăng nhập thành công tài khoản Supplier When Login to Supplier Then User look title "Tra Cứu Dữ Liêu" And User look dashboard "BALANCE" And Check Welcome Message Text Is "Chào mừng bạn đã đến với BALANCE!"

```
LG 12: Kiểm tra đăng nhập thành công bằng phím Enter
    [Tags] @Test Login @Test by Hồ Văn Nhật
    [Documentation] Kiểm tra đăng nhập thành công bằng phím Enter
    Enter "email" in "Tên đăng nhập" with "test admin@gmail.com"
    Enter "text" in "Mât khẩu" with "Nhat@01101999"
    Enter at "Mât khẩu" field to Login
    User look title "Tra Cứu Dữ Liêu"
LG 13: Kiểm tra có thể hiển thi mật khẩu dưới dang text
    [Tags] @Test_Login @Test by Hồ Văn Nhật
    [Documentation] Kiểm tra có thể hiển thị mật khẩu dưới dạng text
    When Enter "text" in "Mật khẩu" with "_RANDOM_"
    And Click "Eye" icon to display password
    Then User look "Mật khẩu" field with type "text"
LG 14: Kiểm tra thông tin nhập sẽ bị xóa sau khi reload trang web
    [Tags] @Test Login @Test by Hồ Văn Nhật
    [Documentation] Kiểm tra thông tin nhập sẽ bị xóa sau khi reload trang web
    When Enter "email" in "Tên đăng nhập" with "_RANDOM_"
    And Enter "text" in "Mât khẩu" with " RANDOM "
    And Reload Page
    Then User look "Tên đăng nhập" field empty
    And User look "Mật khẩu" field empty
LG 15: Kiểm tra chuyển hướng đến trang "Quên mật khẩu" thành công
    [Tags] @Test Login @Test by Hồ Văn Nhật
    [Documentation] Kiểm tra chuyển hướng đến trang "Quên mật khẩu" thành công
    When Click "Quên mât khẩu?" link
    Then User look title "titles.Forgot Password"
    And User look contains title is "Quên Mật Khẩu"
LG 16: Kiểm tra validation text khi không nhập email
    [Tags] @Test Login @Test by Hồ Văn Nhật
    [Documentation] Kiểm tra validation text khi không nhập email
    When Navigate to forgot password page
    And Click "Lấy mã OTP" button
    Then Required message "${email khôi phục}" displayed under "This is a required field!" field
LG 17: Kiểm tra validation text khi nhập sai định dạng email
    [Tags] @Test Login @Test by Hồ Văn Nhật
    [Documentation] Kiểm tra validation text khi nhập sai đinh dang email
    When Navigate to forgot password page
    And Enter "text" in "${email khôi phục}" with "_RANDOM_"
    And Click "Lấy mã OTP" button
    Then Required message "${email khôi phục}" displayed under "Please enter a valid email address!" field
LG 18: Kiểm tra hiển thị thông báo lỗi khi email không tồn tại trên hệ thống
    [Tags] @Test_Login @Test by Hồ Văn Nhật
```

[Documentation] Kiểm tra hiển thị thông báo lỗi khi email không tồn tại trên hệ thống When Navigate to forgot password page And Enter "email" in "\${email khôi phục}" with "_RANDOM_" And Click "Lấy mã OTP" button Then User look message "Địa chỉ email không tồn tại, vui lòng thử địa chỉ email khác." popup LG 19: Kiểm tra có thể điều hướng thành công đến trang nhập mã OTP [Tags] @Test Login @Test by Hồ Văn Nhật [Documentation] Kiểm tra có thể điều hướng thành công đến trang nhập mã OTP When Navigate to forgot password page And Enter "email" in "\${email khôi phục}" with "\${email reset password}" And Click "Lấy mã OTP" button Then User lok button with "Gửi OTP" And User look text with "Vui lòng nhập mã OTP đã gửi đến email của bạn" LG 20: Kiểm tra Validation text trường "Mã OTP" khi bỏ trống mã OTP [Tags] @Test Login @Test by Hồ Văn Nhật [Documentation] Kiểm tra Validation text trường "Mã OTP" khi bỏ trống mã OTP When Navigate to enter OTP page When Click "Gửi OTP" button Then Required message "Mã OTP" displayed under "This is a required field!" field LG 21: Kiểm tra hiển thị thông báo lỗi khi nhập sai mã OTP [Tags] @Test Login @Test by Hồ Văn Nhật [Documentation] Kiểm tra error message khi nhập sai mã OTP When Navigate to enter OTP page And Enter "otp" in "Mã OTP" with "_RANDOM_" And Click "Gửi OTP" button Then User look message "Mã OTP sai hoặc hết thời han sử dung." popup LG 22: Kiểm tra có thể điều hướng đến trang thay đổi mật khẩu thành công [Tags] @Test Change Password @Test by Hồ Văn Nhật [Documentation] Kiểm tra có thể điều hướng đến trang thay đổi mật khẩu thành công When Navigate to enter OTP page And Enter "otp" in "Mã OTP" with "123456" And Click "Gửi OTP" button Then User look title "Reset Password" And User look contains title is "Đặt Lai Mật Khẩu" LG_23: Kiểm tra Validation text khi không nhập "Mật khẩu" và "Xác nhận mật khẩu" [Tags] @Test Change Password @Test by Hồ Văn Nhật [Documentation] Kiểm tra Validation text khi bỏ trống "Mật khẩu" và "Xác nhận mật khẩu" When Go to change password page And Click "Đổi mật khẩu" button Then Required message "Mật khẩu" displayed under "This is a required field!" field And Required message "Xác nhận mật khẩu" displayed under "This is a required field!" field LG 24: Kiểm tra Validation text khi không nhập "Mật khẩu"

```
[Tags] @Test Change Password @Test by Hồ Văn Nhật
    [Documentation] Kiểm tra Validation text khi không nhập "Mật khẩu"
    When Go to change password page
    And Enter "password" in "Mât khẩu" with "${empty}"
    And Click in "Xác nhân mật khẩu" field
    Then Required message "Mât khẩu" displayed under "This is a required field!" field
LG 25: Kiểm tra Validation text khi nhập "Mật khẩu" nhỏ hơn 8 ký tư
    [Tags] @Test Change Password @Test by Hồ Văn Nhật
    [Documentation] Kiểm tra Validation text khi nhập "Mật khẩu" nhỏ hơn 8 ký tự
    When Go to change password page
    And Enter "7 character" in "Mât khẩu" with " RANDOM "
    And Click in "Xác nhân mật khẩu" field
    Then Required message "Mật khẩu" displayed under "Mật khẩu yêu cầu có 8 ký tự trở lên" field
LG 26: Kiểm tra Validation text khi nhập "Mật khẩu" chứa khoảng trắng
    [Tags] @Test Change Password @Test by Hồ Văn Nhật
    [Documentation] Kiểm tra Validation text khi nhập "Mật khẩu" chứa khoảng trắng
    When Go to change password page
    And Enter "text" in "Mật khẩu" with " RANDOM "
    And Click in "Xác nhân mật khẩu" field
    Then Required message "Mât khẩu" displayed under "Mât khẩu không được có khoảng trắng" field
LG 27: Kiểm tra Validation text khi nhập "Mật khẩu" chưa đủ độ bảo mật
    [Tags] @Test Change Password @Test by Hồ Văn Nhật
    [Documentation] Kiểm tra Validation text khi nhập "Mật khẩu" chưa đủ độ bảo mật
    When Go to change password page
    And Enter "number" in "Mật khẩu" with "_RANDOM_"
    And Click in "Xác nhân mật khẩu" field
    Then Required message "Mật khẩu" displayed under "Mật khẩu yêu cầu có 8 ký tự trở lên, có ít nhất 1 chữ
hoa, 1 chữ thường, 1 chữ số và 1 kí tự đặc biệt" field
LG 28: Kiểm tra Validation text khi không nhập "Xác nhận mật khẩu"
    [Tags] @Test Change Password @Test by Hồ Văn Nhật
    [Documentation] Kiểm tra Validation text khi không nhập "Xác nhận mật khẩu"
    When Go to change password page
    And Enter "retype password" in "Xác nhận mật khẩu" with "${empty}"
    And Click in "Mât khẩu" field
    Then Required message "Xác nhận mật khẩu" displayed under "This is a required field!" field
LG 29: Kiểm tra Validation text khi nhập "Xác nhân mật khẩu" không chính xác
    [Tags] @Test Change Password @Test by Hồ Văn Nhật
    [Documentation] Kiểm tra Validation text khi nhập "Xác nhận mật khẩu" không chính xác
    When Go to change password page
    And Enter "password" in "Mật khẩu" with "_RANDOM_"
    And Enter "password" in "Xác nhận mật khẩu" with "_RANDOM_"
    And Click "Đổi mật khẩu" button
```

```
Then Required message "Mật khẩu" displayed under "Xác nhận mật khẩu không chính xác" field
    And Required message "Xác nhận mật khẩu" displayed under "Xác nhận mật khẩu không chính xác" field
LG 30: Kiểm tra successfull message khi thay đổi mật khẩu thành công
    [Tags] @Test Change Password @Test by Hồ Văn Nhật
    [Documentation] Kiểm tra successfull message khi thay đổi mật khẩu thành công
    When Go to change password page
    And Enter "password" in "Mât khẩu" with "Nhat@01101999"
    And Enter "retype password" in "Xác nhân mật khẩu" with "Nhat@01101999"
    And Click "Đổi mật khẩu" button
    Then User look message "Đổi mật khẩu thành công." popup
    And User look title "Login"
LG 31: Kiểm tra có thể hiển thị "Mật khẩu" và "Xác nhận mật khẩu" dưới dạng text
    [Tags] @Test Change Password @Test by Hồ Văn Nhật
    [Documentation] Kiểm tra có thể hiển thị "Mật khẩu" và "Xác nhận mật khẩu" dưới dạng text
    When Go to change password page
    And Enter "password" in "Mật khẩu" with "_RANDOM_"
    And Enter "password" in "Xác nhận mật khẩu" with "_RANDOM_"
    And Click "Eye" icon to display password when reset password
    Then User look "Mật khẩu" field with type "text"
    And User look "Xác nhân mật khẩu" field with type "text"
LG 32: Kiểm tra quay trở lại trang đăng nhập từ trang Quên mật khẩu
    [Tags] @Test Login @Test by Hồ Văn Nhật
    [Documentation] Kiểm tra quay trở lại trang đăng nhập từ trang Quên mật khẩu
    When Navigate to forgot password page
    And Click "Quay trở lại Đăng nhập" link
    Then User look title "Login"
LG 33: Kiểm tra quay trở lại trang đăng nhập từ trang Nhập mã OTP
    [Tags] @Test Login @Test by Hồ Văn Nhật
                       Kiểm tra quay trở lại trang đăng nhập từ trang Nhập mã OTP
    [Documentation]
    When Navigate to enter OTP page
    And Click "Quay trở lại Đăng nhập" link
    Then User look title "Login"
*** Keywords ***
Go to change password page
    When Click "Quên mật khẩu?" link
    And Enter "email" in "${email khôi phuc}" with "${email reset password}"
    And Click "Lấy mã OTP" button
    And Enter "otp" in "Mã OTP" with "123456"
    And Click "Gửi OTP" button
Click "Eye" icon to display password when reset password
    ${elements}
                                Get Elements xpath=//*[contains(@class, 'text-lg')][1]
    FOR ${element} IN @{elements}
```

```
Click
                                  ${element}
    END
Navigate to forgot password page
    When Click "Quên mât khẩu?" link
Navigate to enter OTP page
    When Navigate to forgot password page
    And Enter "email" in "${email khôi phue}" with "${email reset password}"
    And Click "Lấy mã OTP" button
User lok button with "${text}"
    ${button}
                Get Text xpath=//button[contains(text(), '$\{text\}')]
    Should Be Equal "$\{text\}" $\{button\}
User look text with "${text}"
    Wait Until Element Is Visible
                                     //h5[contains(text(),'Vui lòng nhập mã OTP đã gửi đến email của b')]
    ${text_actual}
                                    //h5[contains(text(),'Vui lòng nhập mã OTP đã gửi đến email của b')]
                       Get Text
                         "${text}"
                                      ${text_actual}
    Should Be Equal
```

➤ Kịch bản kiểm thử chức năng đăng nhập trên Android:

```
*** Settings ***
Library
              AppiumLibrary
Resource
               ../Keywords/common.robot
Test Setup
               Setup
Test Teardown
                  Tear Down
*** Test Cases ***
LG 01: Kiểm tra validation text khi bỏ trống trường "Tên đăng nhập"
    [Tags] Test_Login Validation Text
    [Documentation] Kiểm tra validation text khi bỏ trống trường "Tên đặng nhập"
    When Enter "password" in "Mật khẩu" with "_RANDOM_"
    And Click "Đăng Nhâp" button
    Then Show validation text in "Tên đăng nhập" field with "Đây là trường bắt buộc!"
LG 02: Kiểm tra validation text khi bỏ trống trường "Mật khẩu"
    [Tags] Test_Login Validation Text
    [Documentation] Kiểm tra validation text khi bỏ trống trường "Mật khẩu"
    When Enter "email" in "Tên đăng nhập" with "_RANDOM_"
    And Click "Đăng Nhập" button
    Then Show validation text in "Mật khẩu" field with "Đây là trường bắt buộc!"
LG 03: Kiểm tra validation text khi bỏ trống trường "Tên đăng nhập và "Mật khẩu"
    [Tags] Test_Login Validation Text
    [Documentation] Kiểm tra validation text khi bỏ trống trường "Tên đăng nhập và "Mật khẩu"
    When Click "Đăng Nhập" button
    Then Show validation text in "Tên đăng nhập" field with "Đây là trường bắt buộc!"
    And Show validation text in "Mật khẩu" field with "Đây là trường bắt buộc!"
LG 04: Kiểm tra validation text khi nhập "Tên đăng nhập" sai định dạng email
```

```
[Tags] Test_Login Validation Text
    [Documentation] Kiểm tra validation text khi nhập "Tên đăng nhập" sai định dạng email
    When Enter "text" in "Tên đăng nhập" with "_RANDOM_"
    And Enter "password" in "Mât khẩu" with " RANDOM "
    Then Show validation text in "Tên đăng nhập" field with "Vui lòng nhập địa chỉ email hợp lệ!"
LG 05: Kiểm tra hiển thi thông báo lỗi khi nhập sai "Tên đăng nhập"
    [Tags] Test Login Error Message
    [Documentation] Kiểm tra hiển thi thông báo lỗi khi nhập sai "Tên đặng nhập"
    When Enter "email" in "Tên đăng nhập" with "_RANDOM_"
    And Enter "password" in "Mật khẩu" with "Nhat@01101999"
    And Click "Đăng Nhập" button
    Then Display popup with "Thất bại"
    And User look message "Tài khoản chưa được tạo hoặc chưa được kích hoạt." popup
LG 06: Kiểm tra hiển thị thông báo lỗi khi nhập sai "Mật khẩu"
    [Tags] Test_Login Error Message
    [Documentation] Kiểm tra hiển thị thông báo lỗi khi nhập sai "Mật khẩu"
    When Enter "email" in "Tên đăng nhập" with "minhminh@getnada.com"
    And Enter "password" in "Mật khẩu" with "_RANDOM_"
    And Click "Đăng Nhập" button
    Then Display popup with "Thất bai"
    And User look message "Tài khoản hoặc mật khẩu không đúng, vui lòng thử lại" popup
LG 07: Kiểm tra hiển thi thông báo lỗi khi nhập sai "Tên đặng nhập" và "Mật khẩu"
    [Tags] Test_Login Error Message
    [Documentation] Kiểm tra error message khi nhập sai "Tên đăng nhập" và "Mật khẩu"
    When Enter "email" in "Tên đăng nhập" with "_RANDOM_"
    And Enter "password" in "Mât khẩu" with " RANDOM "
    And Click "Đăng Nhập" button
    Then Display popup with "Thất bại"
    And User look message "Tài khoản chưa được tạo hoặc chưa được kích hoạt." popup
LG 08: Kiểm tra đăng nhập thành công tài khoản "Store"
    [Tags] Test_Login Successful Message
    [Documentation] Kiểm tra đăng nhập thành công tài khoản "Store"
    When Enter "email" in "Tên đăng nhập" with "minhminh@getnada.com"
    And Enter "password" in "Mật khẩu" with "Nhat@01101999"
    And Click "Đăng Nhập" button
    Then Verify login successful user "Store" with "Chào mùng ban đã đến với BALANCE!"
LG 09: Kiểm tra đăng nhập thành công tài khoản "Supplier"
    [Tags] Test_Login Successful Message
    [Documentation] Kiểm tra đăng nhập thành công tài khoản "Supplier"
    When Enter "email" in "Tên đăng nhập" with "ha@getnada.com"
    And Enter "password" in "Mật khẩu" with "Nhat@01101999"
    And Click "Đăng Nhập" button
```

```
Then Verify login successful user "Store" with "Chào mừng bạn đã đến với BALANCE!"
LG 10: Kiểm tra có thể hiển thị mật khẩu dưới dạng text
    [Tags] Test_Login Display Password
    [Documentation] Kiểm tra có thể hiển thi mật khẩu dưới dang text
    When Enter "email" in "Tên đăng nhập" with "_RANDOM_"
    And Enter "password" in "Mât khẩu" with " RANDOM "
    Then User look "Mât khẩu" field with type password is "true"
    And Show password in "Mât khẩu"
    Then User look "Mât khẩu" field with type password is "false"
#FORGOT PASSWORD#
*** Test Cases ***
ForgotPassword 01: Kiểm tra có thể điều hướng đến trang "Quên mật khẩu"
    [Tags] Test_ForgotPassword
    [Documentation] Kiểm tra thông tin đặng nhập sẽ không còn khi làm mới trang web
    When Click link "Quên mât khẩu?"
    Then User look title with "Quên Mât Khẩu"
    And User look content with "Vui long nhập e-mail của bạn. Mã xác minh OTP sẽ được gửi cho bạn"
ForgotPassword 02: Kiểm tra validation text khi không nhập "Email khôi phục"
    [Tags] Test ForgotPassword Validation Text
    [Documentation] Kiểm tra validation text khi không nhập "Email khôi phục"
    When Click link "Quên mât khẩu?"
    And Enter "email" in "Email khôi phuc" with "${empty}"
    And Click "Lấy mã OTP" button
    Then Show validation text in "Email khôi phục" field with "Đây là trường bắt buôc!"
ForgotPassword 03: Kiểm tra validation text khi nhập "Email khôi phục" sai định dạng email
    [Tags] Test ForgotPassword Validation Text
    [Documentation] Kiểm tra validation text khi nhập "Email khôi phục" sai định dạng email
    When Click link "Quên mât khẩu?"
    And Enter "text" in "Email khôi phục" with "_RANDOM_"
    And Click "Lấy mã OTP" button
    Then Show validation text in "Email khôi phục" field with "Vui lòng nhập địa chỉ email hợp lệ!"
ForgotPassword 04: Kiểm tra hiển thị thông báo lỗi khi nhập "Email khôi phục" không tồn tại trên hệ thống
    [Tags] Test_ForgotPassword Error Message
    [Documentation] Kiểm tra error message khi nhập email không tồn tại trên hệ thống
    When Click link "Quên mât khẩu?"
    And Enter "email" in "Email khôi phuc" with " RANDOM "
    And Click "Lấy mã OTP" button
    Then Display popup with "Thất bai"
    And User look message "Địa chỉ email không tồn tại, vui lòng thử địa chỉ email khác." popup
ForgotPassword 05: Kiểm tra có thể điều hướng đến trang nhập mã OTP thành công
    [Tags] Test_ForgotPassword Success Message
    [Documentation] Kiểm tra có thể điều hướng đến trang nhập mã OTP thành công
```

```
When Click link "Quên mât khẩu?"
    And Enter "email" in "Email khôi phục" with "92t4709016@husc.edu.vn"
    And Click "Lấy mã OTP" button
    Then User look content with "Vui lòng nhập e-mail của bạn. Mã xác minh OTP sẽ được gửi cho bạn"
ForgotPassword 06: Kiểm tra validation text khi không nhập "Mã OTP"
    [Tags] Test ForgotPassword Validation Text
    [Documentation] Kiểm tra validation text khi không nhập "Mã OTP"
    When Click link "Quên mât khẩu?"
    And Enter "email" in "Email khôi phục" with "92t4709016@husc.edu.vn"
    And Click "Lấy mã OTP" button
    And Enter "OTP" in "Mã OTP" with "${empty}"
    And Click "Gửi OTP" button
    Then Show validation text in "Mã OTP" field with "Đây là trường bắt buộc!"
ForgotPassword 07: Kiểm tra hiển thị thông báo lỗi khi nhập "Mã OTP" không hợp lệ
    [Tags] Test_ForgotPassword Error Message
    [Documentation] Kiểm tra hiển thị thông báo lỗi khi nhập "Mã OTP" không hợp lệ
    When Click link "Quên mât khẩu?"
    And Enter "email" in "Email khôi phuc" with "92t4709016@husc.edu.vn"
    And Click "Lấy mã OTP" button
    And Enter "phone" in "Mã OTP" with " RANDOM "
    And Click "Gửi OTP" button
    Then Display popup with "Thất bai"
    And User look message "Mã OTP sai hoặc hết thời hạn sử dụng." popup
ForgotPassword 08: Kiểm tra có thể điều hướng đến trang thay đổi mật khẩu thành công
    [Tags] Test_ForgotPassword Success Message
    [Documentation] Kiểm tra có thể điều hướng đến trang thay đổi mật khẩu thành công
    When Click link "Quên mât khẩu?"
    And Enter "email" in "Email khôi phuc" with "92t4709016@husc.edu.vn"
    And Click "Lấy mã OTP" button
    And Enter "OTP" in "Mã OTP" with "123456"
    And Click "Gửi OTP" button
    Then User look title with "Đặt Lai Mật Khẩu"
    And User look content with "Mật khẩu yêu cầu có 8 ký tự trở lên, có ít nhất 1 chữ hoa, 1 chữ thường, 1 chữ
số và 1 kí tư đặc biệt."
#CHANGE PASSWORD#
*** Test Cases ***
ChangePassword 01: Kiểm tra Validation text khi bỏ trống cả "Mật khẩu" và "Xác nhận mật khẩu"
    [Tags] Test ChangePassword Validation Text
    [Documentation] Kiểm tra Validation text khi bỏ trống cả "Mật khẩu" và "Xác nhận mật khẩu"
    When Go to change password page
    And Click "Đổi mật khẩu" button
    Then Show validation text in "Mật khẩu" field with "Đây là trường bắt buộc!"
```

And Show validation text in "Xác nhận mật khẩu" field with "Đây là trường bắt buộc!" ChangePassword 02: Kiểm tra Validation text khi nhập "Mật khẩu" nhỏ hơn 8 ký tự [Tags] Test_ChangePassword Validation Text [Documentation] Kiểm tra Validation text khi nhập "Mật khẩu" nhỏ hơn 8 ký tư When Go to change password page And Enter "7 character" in "Mât khẩu" with "1234567" And Enter "password" in "Xác nhân mật khẩu" with "1234567" And Click "Đổi mật khẩu" button Then Show validation text in "Mật khẩu" field with "Mật khẩu yêu cầu có 8 ký tự trở lên" ChangePassword 03: Kiểm tra Validation text khi nhập "Mật khẩu" chứa khoảng trắng [Tags] Test_ChangePassword Validation Text [Documentation] Kiểm tra Validation text khi nhập "Mật khẩu" chứa khoảng trắng When Go to change password page And Enter "text" in "Mât khẩu" with "Hồ Văn Nhât" And Enter "password" in "Xác nhận mật khẩu" with "Hồ Văn Nhật" And Click "Đổi mật khẩu" button Then Show validation text in "Mật khẩu" field with "Mật khẩu không được có khoảng trắng" ChangePassword 04: Kiểm tra Validation text khi nhập "Mật khẩu" chưa đúng yêu cầu [Tags] Test_ChangePassword Validation Text [Documentation] Kiểm tra Validation text khi nhập "Mật khẩu" chứa khoảng trắng When Go to change password page And Enter "number" in "Mât khẩu" with "hovannhat" And Enter "password" in "Xác nhân mật khẩu" with "hovannhat" And Click "Đổi mật khẩu" button Then Show validation text in "Mật khẩu" field with "Mật khẩu yêu cầu có 8 ký tự trở lên, có ít nhất 1 chữ hoa, 1 chữ thường, 1 chữ số và 1 kí tư đặc biệt" ChangePassword 05: Kiểm tra Validation text khi nhập "Xác nhận mật khẩu" không trùng với "Mật khẩu" [Tags] Test_ChangePassword Validation Text [Documentation] Kiểm tra Validation text khi nhập "Xác nhận mật khẩu" không trùng với "Mật khẩu" When Go to change password page And Enter "password" in "Mật khẩu" with "_RANDOM_" And Enter "password" in "Xác nhận mật khẩu" with "_RANDOM_" And Click "Đổi mật khẩu" button Then Show validation text in "Mât khẩu" field with "Xác nhân mật khẩu không chính xác" And Show validation text in "Xác nhận mật khẩu" field with "Xác nhận mật khẩu không chính xác" ChangePassword 06: Kiểm tra hiển thi thông báo khi thay đổi mật khẩu thành công [Tags] Test_ChangePassword Successful Message [Documentation] Kiểm tra hiển thị thông báo khi thay đổi mật khẩu thành công When Go to change password page And Enter "password" in "Mật khẩu" with "Nhat@01101999" And Enter "password" in "Xác nhận mật khẩu" with "Nhat@01101999" And Click "Đổi mật khẩu" button

```
Then Display popup with "Thanh Công"
    And User look message "Đổi mật khẩu thành công." popup
ChangePassword 07: Kiểm tra hiển thị "Mật khẩu" và "Xác nhận mật khẩu" dưới dạng text
    [Tags] Test_ChangePassword Display Password
    [Documentation] Kiểm tra hiển thi "Mật khẩu" và "Xác nhân mật khẩu" dưới dang text
    When Go to change password page
    And Enter "password" in "Mât khẩu" with "Nhat@01101999"
    And Show password in "Mât khẩu"
    Then User look "Mât khẩu" field with type password is "false"
    And Enter "password" in "Xác nhận mật khẩu" with "Nhat@01101999"
    And Show password in "Xác nhận mật khẩu"
    Then User look "Xác nhận mật khẩu" field with type password is "false"
*** Keywords ***
Go to change password page
    When Click link "Quên mật khẩu?"
    And Enter "email" in "Email khôi phục" with "92t4709016@husc.edu.vn"
    And Click "Lấy mã OTP" button
    And Enter "OTP" in "Mã OTP" with "123456"
    And Click "Gửi OTP" button
# Nhập giá tri vào trường dữ liêu trên form.
Enter "${type}" in "${name}" with "${text}"
                      Get Random Text
                                                  "${type}"
                                                                         "${text}"
    \{\text{text}\}=
    IF '${name' == 'Tên đăng nhập' or '${name' == 'Email khôi phục'
                                          set variable
                              ${id}
                                                                  email
    ELSE IF '${name' == 'Mật khẩu'
                       set variable
           ${id}
                                              password
    ELSE IF '${name' == 'Mã OTP'
                       set variable
           ${id}
                                              otp
    ELSE IF '${name' == 'Xác nhận mật khẩu'
          ${id}
                     set variable
                                             passwordComfirm
    END
                                        //*[@resource-id="${id}"]
    ${element}=
                        set variable
    wait until page contains element
                                          ${element}
                                                              ${BROWSER_TIMEOUT}
    clear text
                      ${element}
    input text
                      ${element}
                                               ${text}
Show validation text in "${name}" field with "${text}"
    IF '${name}' == 'Tên đăng nhập' or '${name}' == 'Email khôi phục'
                              ${id}
                                          set variable
                                                                  email
    ELSE IF '${name}' == 'Mật khẩu'
                       set variable
           ${id}
                                              password
    ELSE IF '${name}' == 'M\tilde{a} OTP'
                       set variable
           ${id}
                                              otp
```

```
ELSE IF '${name}' == 'Xác nhận mật khẩu'
          ${id}
                      set variable
                                             passwordComfirm
    END
    IF '${name}' == 'Mât khẩu' or '${name}' == 'Xác nhân mât khẩu'
        ${element} = Get Element Form Item By Name ${name}
//android.widget.EditText[@resource-id="${id}"]/../../android.view.View[@text="${text}"]
    ELSE IF '${name}' != 'Mât khẩu'
         ${element}= Get Element Form Item By Name ${name}
//android.widget.EditText[@resource-id="${id}"]/../..//android.view.View[@text="${text}"]
    END
    wait until page contains element
                                       ${element}
                                                      ${BROWSER_TIMEOUT}
    element text should be
                              ${element}
# Kiểm tra mật khẩu có hiển thị hay không khi click icon "eye"
User look "${name}" field with type password is "${type}"
    IF '${name}' == 'Mât khẩu'
        ${id}
                    set variable
                                          password
    ELSE IF '${name}' == 'M\tilde{a} OTP'
                    set variable
        ${id}
    ELSE IF '${name}' == 'Xác nhân mât khẩu'
        ${id}
                    set variable
                                          passwordComfirm
    END
    ${element}=
                    Get Element Form Item By Name
                                                        ${name}
                                                                    //android.widget.EditText[@resource-
id="${id}"]
    wait until page contains element
                                       ${element} ${BROWSER TIMEOUT}
    ${password_field_type}=
                                get element attribute ${element}
                                                                    password
    Should Be Equal As Strings
                                  ${password field type}
                                                                ${type}
User look in "${name}" field with "${text}"
    IF '${name}' == 'Tên đăng nhập'
       ${type-text}
                         set variable
                                                false
    ELSE IF '\$\{name\}' == 'M\hat{a}t \, kh\hat{a}u'
       ${type-text}
                          set variable
                                                 true
    END
                      Get Element Form Item By Name
    ${element}=
                                                        ${name}
//android.widget.EditText[@password="${type-text}"]
    wait until page contains element
                                       ${element}
                                                      ${BROWSER_TIMEOUT}
                            ${element}
    element text should be
                                          ${text}
#FORGOT PASSWORD#
User look button with "${text}"
    ${element}=
                        set variable
                                        //android.widget.Button[@text="${text}"]
    wait until page contains element
                                        ${element}
                                                       ${BROWSER TIMEOUT}
    element text should be
                                   ${element}
                                                   ${text}
Show password in "${name}"
```

```
IF '${name}' == 'Mật khẩu'

${id} set variable password

ELSE IF '${name}' == 'Xác nhận mật khẩu'

${id} set variable passwordComfirm

END

${element} = Get Element Form Item By Name ${name}

//android.widget.EditText[@resource-id="${id}"]/..//android.widget.TextView[@password="false"]

click element ${element}
```

★ Xem đầy đủ code tất cả module tại link githud sau: https://bom.so/BkXdY2

3.3. KẾT QUẢ KIỂM THỬ

♣ Kết quả kiểm thử chức năng đăng nhập:

1-Login R	eport							202308	Generate 21 23:47:40 UTC+07:0 5 days 7 hours ag
Summary Inf	ormation								
Status: Start Time: End Time: Elapsed Time: Log File:	All tests passed 20230821 23:44:25.450 20230821 23:47:38.428 00:03:12.978 log.html								
Test Statistic	s								
	Total Statistics	¢	Total ≑	Pass +	Fail		Skip ¢	Elapsed \$	D (E-il (Ol-i-
All Tests			33	33	_				Pass / Fail / Skip
				55	0		0	00:03:10	Pass / Fail / Skip
	Statistics by Tag	÷	Total	Pass \$	Fail	\$	0 Skip \$		
@Test_by_Hồ_Văr	• •	¢	Total \$			\$			
	_Nhật	\$		Pass ÷	Fail	\$	Skip \$	Elapsed \$	
@Test_Change_Pa	_Nhật	\$	33	Pass \$	Fail	\$	Skip ÷	Elapsed \$ 00:03:10	
@Test_by_Hồ_Văr @Test_Change_Pa @Test_Login	_Nhật	\$	33 10	Pass \$ 33 10	Fail 0 0 0	\$	Skip * 0 0	Elapsed \$ 00:03:10 00:01:07	Pass / Fail / Skip Pass / Fail / Skip

Hình 3.27. Báo cáo kết quả kiểm thử Login trên Windows

✓ Tổng số lượng testcase: 33

✓ Số lượng testcase Pass: 33

✓ Số lượng testcase Fail: 0

Login Rep	oort						202310	Generated 30 09:27:24 UTC+07:00 8 days 3 hours ago
Summary Inf	ormation							
Status: Start Time: End Time: Elapsed Time: Log File:	All tests passed 20231030 09:12:53.929 20231030 09:27:23.977 00:14:30.048 log.html							
Test Statistic								
	Total Statistics	\$	Total +	Pass \$	Fail +	Skip \$	Elapsed \$	Pass / Fail / Skip
All Tests			25	25	0	0	00:14:29	
	Statistics by Tag	\$	Total \$	Pass \$	Fail +	Skip \$	Elapsed \$	Pass / Fail / Skip
Display Password			2	2	0	0	00:01:10	
Error Message			5	5	0	0	00:02:49	
Successful Messa	ge		3	3	0	0	00:01:53	
Success Message			2	2	0	0	00:01:06	
Test_ChangePass	word		7	7	0	0	00:04:37	
Test_ForgotPassw	vord		8	8	0	0	00:04:22	
Test_Login			10	10	0	0	00:05:30	
Validation Text			12	12	0	0	00:07:00	
	Statistics by Suite		Total	Pass \$	Fail \$	Skip \$	Elapsed \$	Pass / Fail / Skip

Hình 3.28. Báo cáo kết quả kiểm thử Login trên Android

✓ Tổng số lượng testcase: 25

✓ Số lượng testcase Pass: 25

✓ Số lượng testcase Fail: 0



Hình 3.29. Báo cáo kết quả kiểm thử update profile trên Windows

✓ Tổng số lượng testcase: 31

✓ Số lượng testcase Pass: 31

✓ Số lượng testcase Fail: 0

2-Profile Report

Generated 20231031 14:17:15 UTC+07:00 6 days 22 hours ago

Summary Information

 Status:
 All tests passed

 Start Time:
 20231031 13:50:12.840

 End Time:
 20231031 14:17:10.427

 Elapsed Time:
 00:26:57.587

 Log File:
 log.html

Test Statistics

		_				
Total Statistics	\$ Total	Pass ♦	Fail ♦	Skip \$	Elapsed \$	Pass / Fail / Skip
All Tests	27	27	0	0	00:26:49	
Statistics by Tag	\$ Total \$	Pass \$	Fail \$	Skip \$	Elapsed *	Pass / Fail / Skip
Error Message	4	4	0	0	00:04:05	
Hủy thao tác	2	2	0	0	00:02:06	
Navigate_to_update_profile	1	1	0	0	00:00:49	
Show password	1	1	0	0	00:01:06	
Successful Message	2	2	0	0	00:02:50	
Test_Change_Password	11	11	0	0	00:11:03	
Test_Update_Profile	16	16	0	0	00:15:46	
Validation Text	17	17	0	0	00:15:52	
Statistics by Suite	\$ Total \$	Pass ♦	Fail \$	Skip \$	Elapsed \$	Pass / Fail / Skip
2-Profile	27	27	0	0	00:26:58	

Hình 3.30. Báo cáo kết quả kiểm thử update profile trên Android

✓ Tổng số lượng testcase: 27

✓ Số lượng testcase Pass: 27

✓ Số lượng testcase Fail: 0

Link githud: https://bom.so/BkXdY2

3.4. BÁO CÁO KÉT QUẢ

Bảng 3.1. Kết quả sau khi kiểm thử website Balance trên Windows

NO	Feature	Testcase Total	Testcase Passed	Testcase Failed
1	Sign in	33	33	0
2	Update Profile	31	31	0
3	Admin Create User	46	46	0
4	Admin Create Supplier	68	67	1
5	Admin Create Main Store	86	85	1

NO	Feature	Testcase Total	Testcase Passed	Testcase Failed
6	Admin Create Branch Store	78	77	1
7	Admin Create Category	23	22	1
8	Admin Create Tax	28	28	0
9	Supplier Create Product	84	84	0
10	Store Create Non-BAL Supplier	66	65	1
11	Store Create Non-BAL Product	55	55	0
12	Store & Supplier connect	48	48	0
13	Supplier Create Voucher	54	54	0
14	Supplier Import Product From Excel	24	24	0
15	Supplier Edit Information	29	28	1
16	Order flow	51	51	0
17	Store Nhập Hàng Trực Tiếp	33	33	0
18	Store Nhập Hàng Non-Balance	28	27	1
19	Store Creare Product Disposal (Hủy hàng)	25	24	1
20	Store Creare Good return (Trå hàng)	36	32	4
21	CRUD Issueing store Goods transfer (Chuyển hàng)	28	23	5
	TOTAL	954	937	17

Kết quả sau khi thực hiện kiểm thử website balance trên Windows, tổng cộng có 954 test case, bao gồm 937 test case Pass và 17 test case Fail (17 lỗi được kiểm thử viên phát hiện).

Bảng 3.2. Kết quả sau khi kiểm thử website Balance trên Android

NO	Feature	Testcase Total	Testcase Passed	Testcase Failed
1	Sign in	25	25	0
2 Update Profile		27	27	0
TOTAL		52	52	0

Kết quả sau khi thực hiện kiểm thử website balance trên Android, tổng cộng có 52 test case, bao gồm 52 test case Pass và 0 test case Fail (0 lỗi được kiểm thử viên phát hiện).

3.5. BÁO CÁO LÕI (BUG REPORT)

Theo báo cáo kết quả ở mục 3.4 có 17 test case Fail.

Ta tiến hành viết Bug Report như sau:

> Thông tin chung:

✓ Ngày tạo bảng báo cáo: 10/11/2023

✓ Người tạo: Hồ Văn Nhật

✓ Trình duyệt/Thiết bị kiểm thử: Chromium, Firefox, Webkit / Windows 10

✓ Môi trường kiểm thử: QA Environment

> Danh sách bug:

STT	Tên Bug	Mô tả	Ưu tiên	Trạng thái
1	Admin – Lỗi chỉnh sửa nhà cung cấp	Hiển thị sai thông báo khi chỉnh sửa nhà cung cấp với "Số điện thoại" đã tồn tại	Trung bình	Đang chờ sửa
2	Admin – Lỗi chỉnh sửa cửa hàng chính	Hiển thị sai thông báo khi chỉnh sửa cửa hàng chính với "Số điện thoại" đã tồn tại	Trung bình	Đang chờ sửa
3	Admin – Lỗi chỉnh sửa cửa hàng chi nhánh	Hiển thị sai thông báo khi chỉnh sửa cửa hàng chi nhánh với "Số điện thoại" đã tồn tại	Trung bình	Đang chờ sửa

STT	Tên Bug	Mô tả	Ưu tiên	Trạng thái
4	Admin – Lỗi tạo mới danh mục cấp 2	Hiện thị sai thông báo khi tạo mới		Đang chờ sửa
5	Store – Lỗi chỉnh nhà cung cấp Non_Balance	nhà cung cấp sửa nhà cung cấp Non-Balance Trung bình		Đang chờ sửa
6	Supplier – Lỗi chỉnh sửa thông tin nhà cung cấp	Hiển thị sai thông báo khi chỉnh sửa nhà cung cấp với "Số điện thoại" đã tồn tại	Trung bình	Đang chờ sửa
7	Store – Lỗi tạo mới nhập hàng Non_Balance	Hiển thị sai thông báo khi chỉnh sửa "Số lượng nhập hàng" lớn hơn 999999	Trung bình	Đang chờ sửa
8	Store – Lỗi tạo mới huỷ hàng	Hiển thị sai thông báo khi chỉnh sửa "Số lượng huỷ" lớn hơn 999999	Trung bình	Đang chờ sửa
9	Store – Lỗi tạo mới trả hàng	Hiển thị sai Validation text khi bỏ trống trường "Thời gian trả hàng"	Trung bình	Đang chờ sửa
10	Store – Lỗi tạo mới trả hàng	Hiển thị sai Validation text khi bỏ trống trường "Thời gian trả hàng" – Trả hàng không theo phiếu nhập hàng	Trung bình	Đang chờ sửa
11	Store – Lỗi tạo mới trả hàng	Hiển thị sai Validation text khi bỏ trống trường "Lý do trả hàng"	Trung bình	Đang chờ sửa
12	Store – Lỗi chỉnh sửa phiếu trả hàng	Hiển thị sai thông báo khi chỉnh sửa "SL trả" lớn hơn 999999	Trung bình	Đang chờ sửa
13	Store – Lỗi tạo mới chuyển hàng	Hiển thị sai Validation text khi bỏ trống trường "Thời gian chuyển hàng"	Trung bình	Đang chờ sửa

STT	Tên Bug	Mô tả	Ưu tiên	Trạng thái
14	Store – Lỗi tạo mới chuyển hàng	Hiển thị sai Validation text khi bỏ trống trường "Lý do"	Trung bình	Đang chờ sửa
15	Store – Lỗi chỉnh sửa phiếu chuyển hàng	Hiển thị sai Validation text khi bỏ trống trường "Thời gian chuyển hàng"	Trung bình	Đang chờ sửa
16	Store – Lỗi chỉnh sửa phiếu chuyển hàng	Hiển thị sai Validation text khi bỏ trống trường "Lý do"	Trung bình	Đang chờ sửa
17	Store – Lỗi chỉnh sửa phiếu chuyển hàng	Hiển thị sai thông báo khi nhập "Số lượng chuyển" lớn hơn 999999	Trung bình	Đang chờ sửa

> Chi tiết bug:

✓ Bug – 01: Admin – Lỗi chỉnh sửa nhà cung cấp:

Mô tả: Khi Admin thực hiện thao tác chỉnh sửa thông tin nhà cung cấp và thay đổi "Số điện thoại" thành một số đã tồn tại trong hệ thống, hệ thống hiển thị thông báo không chính xác hoặc không đúng về tình trạng của thao tác chỉnh sửa.

Bước để tái tạo:

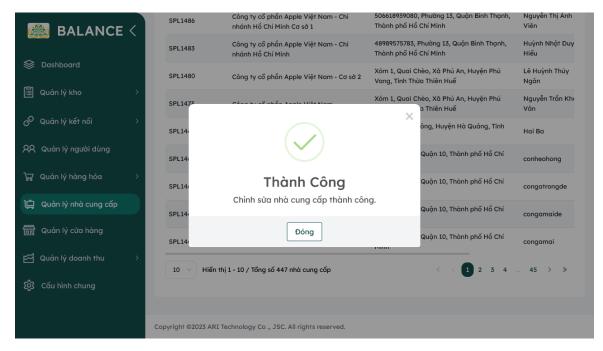
- 1. Login to Admin
- 2. Go to edit information supplier
- 3. Enter "phone" in "Số điện thoại đại diện" with "0977855888"
- 4. Click "Luu" button

Kỳ vọng:

- o Hiển thị popup "Thất bại"
- o Hiển thị message "Số điện thoại đã được đăng ký trước đó."

Kết quả thực tế:

- o Hiển thị popup "Thành công"
- O Hiển thị message "Chỉnh sửa nhà cung cấp thành công."



Hình 3.31. Bug - 01

✓ Bug – 02: Admin – Lỗi chỉnh sửa thông tin cửa hàng chính:

Mô tả: Khi Admin thực hiện thao tác chỉnh sửa thông tin cửa hàng chính và thay đổi "Số điện thoại" thành một số đã tồn tại trong hệ thống, hệ thống hiển thị thông báo không chính xác hoặc không đúng về tình trạng của thao tác chỉnh sửa.

Bước để tái tạo:

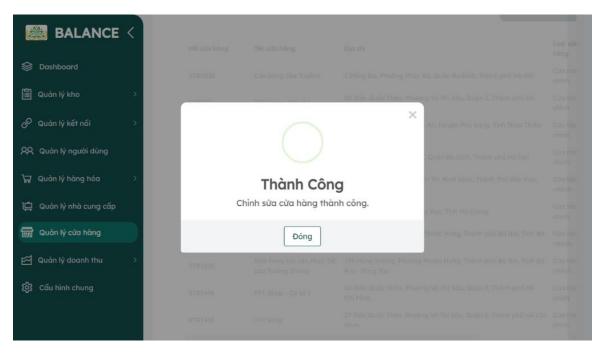
- 1. Login to Admin
- 2. Go to edit store page
- 3. Enter "phone" in "Số điện thoại đại diện" with " 0869667249"
- 4. Click "Luu" button

Kỳ vọng:

- Hiển thị popup "Thất bại"
- Hiển thị message "Số điện thoại đã được đăng ký trước đó."

Kết quả thực tế:

- Hiển thị popup "Thành công"
- Hiển thị message "Chỉnh sửa cửa hàng thành công."



Hình 3.32. Bug – 02

✓ Bug – 03: Admin – Lỗi chỉnh sửa cửa hàng chi nhánh:

Mô tả: Khi Admin thực hiện thao tác chỉnh sửa thông tin cửa hàng chi nhánh và thay đổi "Số điện thoại" thành một số đã tồn tại trong hệ thống, hệ thống hiển thị thông báo không chính xác hoặc không đúng về tình trạng của thao tác chỉnh sửa.

Bước để tái tạo:

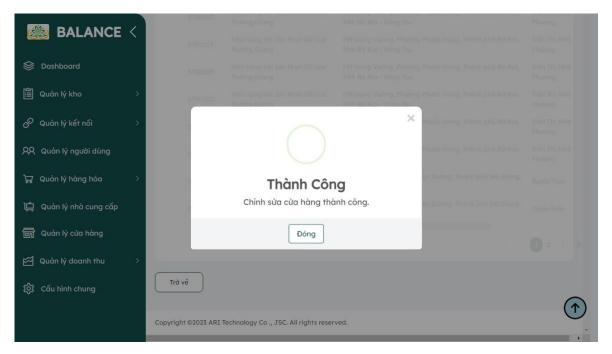
- 1. Login to Admin
- 2. Go to edit Branch store page
- 3. Enter "phone" in "Số điện thoại đại diện" with " 0947284210"
- 4. Click "Luu" button

Kỳ vọng:

- o Hiển thị popup "Thất bại"
- Hiển thị message "Số điện thoại đã được đăng ký trước đó."

Kết quả thực tế:

- o Hiển thị popup "Thành công"
- Hiển thị message "Chỉnh sửa cửa hàng thành công."



Hình 3.33. Bug -03

✓ Bug – 04: Admin – Lỗi tạo mới danh mục cấp 2:

Mô tả: Khi Admin thực hiện thao tác tạo mới "Danh mục cấp 2" và "Danh mục cấp 2" đã được sản phẩm áp dụng, hệ thống hiển thị thông báo với message không chính xác hoặc không đúng về tình trạng của thao tác tạo mới.

Bước để tái tạo:

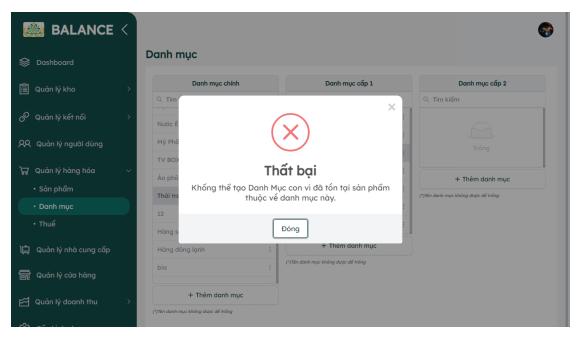
- 1. Login to Admin
- 2. Go to create category page
- 3. Select category in "Danh muc chính" with "Thời trang nữ"
- 4. Select category in "Danh mục cấp 1" with "Váy"
- 5. Click on "Thêm danh mục" button in "Danh mục cấp 2"
- 6. Enter category "text" in "Danh mục cấp 2" with "_RANDOM_"
- 7. Click on "Luu" button in "Danh mục cấp 2"

Kỳ vọng:

- o Hiển thị popup "Thất bại"
- Hiển thị message "Không thể tạo Danh Mục con vì đã tồn tại sản phẩm thuộc về danh mục này."

Kết quả thực tế:

- Hiển thị popup "Thất bại"
- Hiển thị message "Khống thể tạo Danh Mục con vì đã tồn tại sản phẩm thuộc về danh mục này."



Hình 3.34. Bug – 04

Link quản lý bug: https://bom.so/XAs0lx

3.6. ĐÁNH GIÁ HIỆU QUẢ KIỂM THỬ

Dựa vào quy tình kiển thử 6 bước:

- Phân tích yêu cầu khách hàng.
- Lập kế hoạch kiểm thử.
- Thiết kế kịch bản kiểm thử.
- Chuẩn bị môi trường kiểm thử.
- Thực thi kiểm thử.
- Đóng chu trình kiểm thử.

Đã kiểm thử xong website trên nền tản Windows và Android với công việc chính là kiểm thử chức năng. Qua đó tìm thấy các lỗi (bug) sau nhiều lần kiểm thử và đưa ra báo cáo lỗi (Bug Report). Tuy nhiên, việc kiểm thử tự động không thể tránh khỏi thiếu sót. Nhưng nhìn chung, cũng đã đảm bảo được chất lượng tốt cho phần mềm.

3.7. KẾT LUẬN CHƯƠNG

Chương 3 đã trình bày quá trình cài đặt và triển khai thực nghiệm sử dụng công cụ kiểm thử tự động Robot Framework. Việc triển khai thực nghiệm này nhằm mục đích đánh giá hiệu quả của Robot Framework và Appium trong việc kiểm thử tự động trên cả website và ứng dụng Android.

- ✓ Quá trình cài đặt Robot Framework và Appium đã được thực hiện theo các bước chi tiết và hướng dẫn cụ thể. Việc này giúp đảm bảo sự chuẩn bị kỹ lưỡng trước khi bắt đầu thực hiện các test case.
- ✓ Các test case đã được xây dựng và triển khai trên cả website và ứng dụng Android, đảm bảo sự đa dạng trong việc kiểm thử chức năng và tương tác trên nhiều nền tảng.
- ✓ Kết quả kiểm thử đã được đánh giá và tóm tắt một cách chi tiết. Tỷ lệ thành công và thất bại, số lượng lỗi phát hiện, và thời gian chạy test case là những yếu tố chính được xem xét.

TỔNG KẾT VÀ ĐÁNH GIÁ

4.1. SO SÁNH ROBOT FRAMEWORK VỚI CÁC CÔNG CỤ KIỂM THỬ KHÁC

Bång 3.3. Bång so sánh giữa Robot Framework và Selenium WebDriver

Công cụ Tiêu chí	Robot Framework	Selenium WebDriver
Cú pháp và Đọc hiểu	 ✓ Cú pháp tabular (bảng) dễ đọc và viết. ✓ Dễ hiểu đối với người không có kinh nghiệm lập trình. 	Sử dụng API của Java hoặc các ngôn ngữ khác, có thể phức tạp hơn đối với người mới học.
Tính Linh hoạt	 ✓ Hỗ trợ nhiều thư viện và plugins cho kiểm thử Web, API, mobile. ✓ Có thể mở rộng dễ dàng. 	Chủ yếu tập trung vào kiểm thử Web, có thể sử dụng cùng với các framework khác để kiểm thử API hoặc mobile.
Hiệu suất	 ✓ Phù hợp cho dự án vừa và nhỏ. ✓ Hiệu suất có thể giảm khi số lượng test case lớn. 	Phù hợp cho các dự án lớn với số lượng test case nhiều.
Hỗ trợ Môi trường và Ngôn ngữ	 ✓ Hỗ trợ nhiều ngôn ngữ lập trình (Python, Java, .NET). ✓ Tích hợp tốt với nhiều môi trường thử nghiệm. 	Thường sử dụng với Java, nhưng cũng hỗ trợ các ngôn ngữ khác như Python, C#, Ruby.

4.2. TỔNG KẾT NỘI DUNG ĐỒ ÁN

Đồ án đã tập trung vào nghiên cứu và đánh giá hiệu quả của Robot Framework và Appium trong kiểm thử tự động trên website Balance trên nền tản Windows và ứng dụng Android. Dưới đây là những điểm chính của đồ án:

- ✓ Cài Đặt và Triển Khai: Quá trình cài đặt và triển khai được thực hiện một cách chi tiết, chuẩn bị cho quá trình kiểm thử.
- ✓ Kiểm Thử Tự Động: Kiểm thử đã được thực hiện trên cả website trên nền tảng Windows và ứng dụng Android, đảm bảo sự đa dạng và toàn diện.
- ✓ Hiệu Quả và So Sánh: Kết quả kiểm thử đã được đánh giá, và so sánh với các công cụ kiểm thử khác đã được thực hiện để hiểu rõ về ưu điểm và nhược điểm.

4.3. ĐÁNH GIÁ ĐẠT ĐƯỢC MỤC TIÊU ĐỀ RA BAN ĐẦU

- √ Đồ án không chỉ mở rộng kiến thức về kiểm thử tự động mà còn đặt ra những thách thức thực tế. Kết quả thu được sẽ đóng góp vào quá trình phát triển và cải thiện chất lượng sản phẩm phần mềm.
- ✓ Cuối cùng, đồ án mở ra những cơ hội mới cho sự tiếp tục nghiên cứu và áp dụng Robot Framework và Appium trong các dự án thực tế.

4.4. HƯỚNG PHÁT TRIỂN VÀ NÂNG CAO TRONG TƯƠNG LAI

Để tối ưu hóa sự sử dụng của Robot Framework và Appium trong kiểm thử tự động, có một số hướng phát triển và nâng cao có thể được xem xét trong tương lai:

- ✓ Tối Ưu Hóa Hiệu Năng: Nghiên cứu và thực hiện các chiến lược tối ưu hóa hiệu năng cho cả Robot Framework và Appium để đảm bảo chúng hoạt động mượt mà và hiệu quả trên các dự án lớn và phức tạp.
- ✓ Tích Hợp Mở Rộng Thêm Thư Viện và Plugin: Mở rộng sự tích hợp với các thư viện và plugin mới để mở rộng khả năng kiểm thử, đặc biệt là đối với các công nghệ và framework mới xuất hiện trong cộng đồng phần mềm.
- ✓ Tăng Cường Tính Linh Hoạt Cho Kiểm Thử Trên Nền Tảng Khác: Nghiên cứu và phát triển tính năng để tăng cường tính linh hoạt của Robot Framework và Appium trong việc kiểm thử trên các nền tảng khác như iOS, Windows, và các loại thiết bị di động khác.
- ✓ Hỗ Trợ Cộng Đồng: Tiếp tục đóng góp vào cộng đồng Robot Framework và
 Appium thông qua việc chia sẻ kiến thức, góp phần vào tài liệu, và tham gia vào
 việc giải quyết vấn đề để làm cho cộng đồng ngày càng mạnh mẽ.
- ✓ **Nghiên Cứu và Áp Dụng Các Xu Hướng Mới:** Liên tục nghiên cứu và áp dụng các xu hướng mới trong lĩnh vực kiểm thử tự động, bao gồm cả trí tuệ nhân tạo, machine learning, và automation testing trong môi trường DevOps.
- ✓ **Tích Hợp Với Công Cụ Quản Lý Dự Án và CI/CD:** Tăng tính tương tác và tích hợp với các công cụ quản lý dự án như Jira, và các hệ thống CI/CD để tối ưu hóa quy trình kiểm thử trong quy trình phát triển.

DANH MỤC TÀI LIỆU THAM KHẢO

- [1]. Pressman, R. S. (2014). Software Engineering: A Practitioner's Approach. McGraw-Hill Education.
- [2]. Kang, M. S., & Kim, S. H. (2013). Mobile Application Testing with Robot Framework. Packt Publishing.
- [3]. Richardson, M., & Riley, D. (2013). Appium Essentials. Packt Publishing.
- [4]. Burns, D., & Copeland, G. (2014). Appium: Mobile Automation Made Awesome. Appium contributors.
- [5]. Nguyen, V. H., & Tran, T. N. (2017). Robot Framework Test Automation. Packt Publishing.
- [6]. Marick, B. (2003). Everyday Scripting with Ruby: For Teams, Testers, and You. Pragmatic Bookshelf.
- [7]. Liew, S. (2018). Appium Mobile Automation Testing from Scratch + Framework. Udemy Online Course.
- [8]. Roger, S. P. (2015). Selenium WebDriver: From Foundations to Framework. Apress.
- [9]. Tài liệu trực tuyến:
 - ✓ Robot Framework Official Documentation
 - ✓ Appium Documentation
 - ✓ Browser Library
- [10]. Bài báo và Nguồn Thông Tin Trực Tuyến:
 - ✓ Smith, J. (2019). Comparing Automated Testing Tools: Selenium, Appium, and Espresso. DZone.
 - ✓ Johnson, M. (2020). Introduction to Robot Framework. TestProject Blog.