

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



LƯƠNG QUANG CƯỜNG

**NGHIÊN CỨU PHƯƠNG PHÁP KIỂM THỬ CHỨC
NĂNG TỰ ĐỘNG VÀ ÁP DỤNG CHO HỆ THỐNG
VNEDU**

ĐỀ ÁN TỐT NGHIỆP THẠC SĨ KỸ THUẬT
(Theo định hướng ứng dụng)

HÀ NỘI - NĂM 2024

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



Lương Quang Cường

**NGHIÊN CỨU PHƯƠNG PHÁP KIỂM THỬ CHỨC
NĂNG TỰ ĐỘNG VÀ ÁP DỤNG CHO HỆ THỐNG
VNEDU**

Chuyên ngành: Hệ thống thông tin

Mã số: 8.48.01.04 (Hệ thống thông tin)

ĐỀ ÁN THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

NGƯỜI HƯỚNG DẪN: TS. Đỗ Thị Bích Ngọc

HÀ NỘI - 2024

LỜI CAM ĐOAN

Em xin cam đoan đề án là thành quả công trình nghiên cứu độc lập với sự giúp đỡ của giáo viên hướng dẫn. Kết quả nghiên cứu chưa từng được công bố trong bất kỳ công trình nào khác. Tất cả sự giúp đỡ cho việc thực hiện đề án đã được cảm ơn và các thông tin trích dẫn đều có nguồn gốc rõ ràng.

TÁC GIẢ ĐỀ ÁN



LƯƠNG QUANG CƯỜNG

LỜI CẢM ƠN

Trước tiên, em xin cảm ơn Học viện Công nghệ Bưu chính viễn thông, Khoa Đào tạo sau đại học, các thầy cô giáo của Học viện đã tạo điều kiện và trực tiếp giảng dạy, giúp đỡ em hoàn thành chương trình đào tạo Thạc sĩ kỹ thuật.

Em xin gửi lời cảm ơn sâu sắc tới Tiến sĩ Đỗ Thị Bích Ngọc đã dành nhiều thời gian, tâm huyết hướng dẫn tận tình và đóng góp các ý kiến để em hoàn thiện đề án này.

Em xin chân thành cảm ơn tới Ban lãnh đạo, anh chị em đồng nghiệp tại Trung tâm Giải pháp Giáo dục điện tử - Công ty Công Nghệ thông Tin VNPT đã nhiệt tình góp ý trong quá trình em thực hiện khảo sát và viết bài.

Cuối cùng, em muốn bày tỏ sự biết ơn đối với gia đình, người thân, bạn bè đã luôn bên cạnh khích lệ, động viên, giúp đỡ em trong suốt thời gian học tập và hoàn thiện luận văn.

Em xin trân trọng cảm ơn.

Hà Nội, ngày tháng năm 2024

Người viết



Lương Quang Cường

MỤC LỤC

LỜI CAM ĐOAN	i
LỜI CẢM ƠN	ii
MỤC LỤC	iii
DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT.....	v
DANH MỤC CÁC BẢNG	vi
DANH MỤC CÁC HÌNH	vii
MỞ ĐẦU	1
1. Tính cấp thiết của đề án	1
2. Tổng quan đề án.....	2
3. Mục tiêu của đề án	3
4. Đối tượng và phạm vi đề án.....	3
5. Nội dung đề án	3
CHƯƠNG 1. KIỂM THỬ CHỨC NĂNG VÀ HIỆN TRẠNG CỦA VNEDU	4
1.1. Kiểm thử Chức năng.....	4
1.1.1. Kiểm thử chức năng là gì	4
1.1.2. Khó khăn gặp phải khi kiểm thử chức năng.....	5
1.2. Giới thiệu và phân tích hiện trạng kiểm thử chức năng của vnEdu.....	6
1.2.1. Hệ sinh thái giáo dục vnEdu.....	6
1.2.2. Các điểm nổi bật của hệ sinh thái vnEdu so với các giải pháp khác trên thị trường	6
1.2.3. Tính năng của hệ sinh thái vnEdu	7
1.2.4. Một số lợi ích mang lại từ hệ sinh thái vnEdu	8
1.2.5. Hiện trạng kiểm thử chức năng của hệ sinh thái vnEdu.....	9
1.3. Kiểm thử chức năng tự động:	11
1.3.1. Giới thiệu	11
1.3.2. Một số công cụ kiểm thử chức năng tự động phổ biến hiện nay	12
1.3.2. Khó khăn khi sử dụng các công cụ kiểm thử tự động.....	18
1.3.3. Mô hình tổ chức dự án kiểm thử tự động thông dụng hiện nay:.....	19
KẾT LUẬN CHƯƠNG	21
CHƯƠNG 2. ĐỀ XUẤT PHƯƠNG PHÁP KIỂM THỬ TỰ ĐỘNG CHO HỆ	

SINH THÁI VNEDU.....	22
2.1. Đánh giá về việc áp dụng công cụ kiểm thử tự động trong thực tế.....	22
2.2. Các bước xây dựng đề xuất kiểm thử chức năng tự động cho hệ thống vnEdu:	23
2.3. Đề xuất giải pháp kiểm thử tự động cho hệ thống vnEdu:	25
2.3.1. Đánh giá thực trạng, yêu cầu và chọn lựa cách thức triển khai:	25
2.3.2. Chọn mô hình triển khai:.....	25
2.3.3. Phân tích thuận lợi, khó khăn và chọn ra những chức năng có thể kiểm thử tự động và những chức năng không kiểm thử tự động:	26
2.3.4. Phân loại những tính năng có thể kiểm thử tự động và những tính năng không áp dụng kiểm thử tự động:.....	28
2.3.5. Nghiên cứu triển khai bộ template kiểm thử tự động:	31
Cách tổ chức đề xuất	31
Template kiểm thử tự động đề xuất:.....	32
2.3.6. Xây dựng các thư viện, các hàm, các keyword và báo cáo hỗ trợ cho việc kiểm thử chức năng tự động:	33
2.3.7. Đánh giá và xây dựng mô hình báo cáo:	35
KẾT LUẬN CHƯƠNG	37
CHƯƠNG 3. THỰC NGHIỆM VÀ ĐÁNH GIÁ	38
3.1. Triển khai kiểm thử tự động cho hệ sinh thái vnEdu	38
3.1.1. Xây dựng kịch bản thủ công các tính năng có thể viết kịch bản kiểm thử tự động.....	38
3.1.2. Xây dựng kịch bản kiểm thử chức năng tự động theo các kịch bản thủ công đã xây dựng.....	42
3.1.3. Đề xuất áp dụng những mở rộng.....	44
3.2. Kết quả thực nghiệm được đề xuất	47
3.3. Đánh giá kết quả thực nghiệm	48
KẾT LUẬN CHƯƠNG	50
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	51
DANH MỤC TÀI LIỆU THAM KHẢO.....	53

DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT

Từ viết tắt	Giải thích	Tên tiếng anh
VNPT IT	Công ty Công Nghệ thông tin VNPT	VNPT IT
SGD	Sở giáo dục	
PGD	Phòng Giáo dục	
TCGD	Tổ chức giáo dục	
CSGD	Cơ sở Giáo dục	
HTML	Ngôn ngữ đánh dấu siêu văn bản	Hypertext Markup Language
XML	Ngôn ngữ đánh dấu mở rộng	Extensible Markup Language
POM	Mô hình đối tượng trang	Page Object Model
PM	Người quản lý dự án	Project Manager
BA	Đội phân tích kinh doanh	Business Analysis
QC/Tester	Kiểm thử viên	Quality Controller/Tester

DANH MỤC CÁC BẢNG

Bảng 1.1. Các phân hệ tính năng của sinh thái vn Edu.....	7
Bảng 1.2. So sánh các phần mềm kiểm thử tự động hiện nay	18
Bảng 3.1. Kịch bản kiểm thử chi tiết	40

DANH MỤC CÁC HÌNH

Hình 1.1. Tình hình triển khai thực tế của hệ sinh thái vnEdu	7
Hình 1.2. Các lợi ích khi sử dụng hệ sinh thái vnEdu	8
Hình 1.3. Các trình duyệt được hỗ trợ bởi Webdriver	13
Hình 1.4. Các điểm mạnh của Katalon Studio	14
Hình 1.5. Mô hình hoạt động của RobotFramework	16
Hình 1.6. Mô hình Page Object Model (POM)	20
Hình 2.1. Mô hình các thành phần trong dự án kiểm thử tự động được đề xuất	31
Hình 2.2. Mô hình tổ chức dự án kiểm thử tự động cho vnEdu	33
Hình 2.3. Keyword Readexcel	35
Hình 2.4. Báo cáo HTM của Robot Framework	36
Hình 2.5. Các keyword để tính kịch bản Pass/Failed.....	37
Hình 2.6. Báo cáo Excel được học viên xây dựng	37
Hình 2.7. Gửi báo cáo qua Telegram	37
Hình 3.1. Tiêu đề các trường hợp kiểm tra	39
Hình 3.2. File Resource (Nơi chứa các keyword cho từng màn hình).....	43
Hình 3.3. Một kịch bản kiểm thử trong file Testcases	43
Hình 3.4. Cách gọi kịch bản trong file Testsuite	44
Hình 3.5. Báo cáo HTML sau khi chạy	44
Hình 3.6. Keyword để chạy lại testcases	45
Hình 3.7. Gọi đến keyword Rerun_Testcases dựa vào biến checkstt	45
Hình 3.8. Tin nhắn thông báo kết quả kiểm thử chức năng	46
Hình 3.9. Tin nhắn gửi báo cáo excel trong nhóm chat Telegram.....	46
Hình 3.11. Các công cụ tích hợp liên tục sử dụng trong DevOps.....	47
Hình 3.12. Quy trình DevOps triển khai cho vnEdu.....	47

MỞ ĐẦU

1. Tính cấp thiết của đề án

Công nghệ thông tin đang trải qua một sự phát triển nhanh chóng trong thời đại hiện tại. Điều này đi kèm với sự gia tăng vượt bậc về số lượng và quy mô của hệ thống mạng và phần mềm. Tuy nhiên, sự tăng trưởng này cũng đã kéo theo nhiều vấn đề liên quan đến lỗi và sự cố của phần mềm, ảnh hưởng đến kinh tế, xã hội và uy tín của các tổ chức.

Các sự cố này không chỉ xuất phát từ những vấn đề khách quan, mà còn do sự thiếu sót trong quá trình kiểm thử và thẩm định trước khi sản phẩm hoặc dịch vụ được đưa ra thị trường.

Như vậy, mặc dù sự phát triển của phần mềm và công nghệ ngày càng phức tạp và tiên tiến, việc đảm bảo chất lượng vẫn là một vấn đề cấp thiết. Đó là lý do tại sao kiểm thử phần mềm đóng một vai trò quan trọng trong quá trình cung cấp sản phẩm hoặc dịch vụ cho khách hàng, đảm bảo rằng chúng đã được kiểm tra kỹ lưỡng và đáng tin cậy.

Vì vậy, việc đảm bảo chất lượng vẫn là một vấn đề cấp thiết. Đó là lý do tại sao kiểm thử phần mềm đóng một vai trò quan trọng trong quá trình cung cấp sản phẩm hoặc dịch vụ cho khách hàng, đảm bảo rằng phần mềm đã được kiểm tra kỹ lưỡng và đáng tin cậy.

Việc sử dụng các hệ thống kiểm thử tự động là hết sức cần thiết. Chúng cho phép thực hiện kiểm thử phần mềm một cách nhanh chóng, hiệu quả và chính xác, đồng thời giảm bớt sự phụ thuộc vào công sức và thời gian của con người.

Trung tâm Giáo dục điện tử trực thuộc công ty Công nghệ thông tin VNPT thực hiện các chức năng nhiệm vụ cho công cuộc chuyển đổi số trong ngành giáo dục với 3 khối chính là khối Quản lý số chuyên về quản lý nhà trường, Khối học tập số chuyên cung cấp các giải pháp học và thi trực tuyến và khối tích hợp số chuyên cung cấp các hạng mục chuyển đổi dữ liệu với các hệ thống bộ ban ngành khác. Trung tâm

giải pháp Giáo dục điện tử cung cấp các sản phẩm dịch vụ liên quan đến ngành giáo dục theo quy định, thông tư của bộ Giáo dục đào tạo và tùy biến phù hợp với yêu cầu của 63 tỉnh thành trên toàn quốc.

Hệ sinh thái Giáo dục của trung tâm Giải pháp Giáo dục điện tử gồm rất nhiều phần mềm được sử dụng 63/63 tỉnh thành với số lượng sở liên lạc là hơn 8.000.000 sở liên lạc, hơn 29.000 trường học, hơn 800.000 giáo viên trên toàn quốc cung cấp đầy đủ các nghiệp vụ về giáo dục cho toàn trình công tác chuyển đổi số giáo dục trên diện rộng. Hệ thống chỉ cần downtime trong một khoản thời gian ngắn sẽ gây thiệt hại rất lớn.

Với lượng sử dụng lớn, thay đổi cập nhật nhiều, số lượng tính năng lớn, các hệ thống trong hệ sinh thái cần một chất lượng và độ ổn định cao. Với lượng nhân sự kiểm thử hạn chế và định hướng phát triển theo mô hình Agile/Scrum khiến hệ thống cải tiến liên tục nên việc triển khai kiểm thử tự động là một việc rất quan trọng để đảm bảo chất lượng cho các hệ thống.

Vì những lý do trên nên em đã lựa chọn đề án “**NGHIÊN CỨU PHƯƠNG PHÁP KIỂM THỬ CHỨC NĂNG TỰ ĐỘNG VÀ ÁP DỤNG CHO HỆ THỐNG VNEDU**” nhằm xem xét các vấn đề về kiểm thử tự động và áp dụng kiểm thử tự động trong các hệ thống công nghệ thông tin.

2. Tổng quan đề án

Kiểm thử chương trình và phần mềm là một quá trình đòi hỏi sự đóng góp của nhiều chuyên gia kiểm thử khác nhau. Trong những năm 1980, việc sử dụng kiểm thử viên đã trở nên phổ biến trong các dự án phần mềm, nhưng cho đến sau này, nghề nghiệp này mới thực sự được công nhận và coi là một lĩnh vực chuyên biệt.

Trong kiểm thử phần mềm, kiểm thử tự động (test automation) là việc sử dụng công cụ (tool) để thực hiện các kiểm tra và so kết quả thực tế với kết quả dự đoán. Kiểm thử tự động là có thể tự động hóa một số nhiệm vụ kiểm thử lặp đi lặp lại, hay là các kiểm thử bổ sung nhưng sẽ khó thực hiện thủ công. Kiểm thử tự động rất quan trọng kiểm thử liên tục.

3. Mục tiêu của đề án

Đề án xem xét và tìm hiểu về cơ sở lý thuyết về kiểm thử và triển khai kiểm thử phần mềm tự động để giảm nguồn nhân lực kiểm thử và đảm bảo chất lượng phần mềm so với công việc kiểm thử thủ công.

Mục tiêu chính của đề án là nghiên cứu cách áp dụng công cụ kiểm thử tự động vào các phần mềm, đảm bảo phần mềm hạn chế lỗi đến mức tối thiểu.

4. Đối tượng và phạm vi đề án

Đề án bao gồm đối tượng và phạm vi như sau:

- Giới thiệu và phân tích hiện trạng của vnEdu.
- Tìm hiểu về kiểm thử chức năng tự động.
- Đề xuất và xây dựng phương pháp Kiểm thử chức năng tự động cho các chức năng chính trên Web của vnEdu.
- Đánh giá hiệu quả của việc áp dụng phương pháp.

5. Nội dung đề án

Ngoài mở đầu và kết luận, đề tài có kết cấu gồm 4 chương sau:

CHƯƠNG I: KIỂM THỬ CHỨC NĂNG VÀ HIỆN TRẠNG CỦA VNEDU

CHƯƠNG 2. ĐỀ XUẤT PHƯƠNG PHÁP KIỂM THỬ TỰ ĐỘNG CHO HỆ SINH THÁI VNEDU

CHƯƠNG 3. THỰC NGHIỆM VÀ ĐÁNH GIÁ

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

CHƯƠNG 1. KIỂM THỬ CHỨC NĂNG VÀ HIỆN TRẠNG CỦA VNEDU

Chương này chủ yếu tìm hiểu về kiểm thử chức năng và phân tích hiện trạng kiểm thử chức năng của vnEdu.

1.1. Kiểm thử Chức năng

1.1.1. Kiểm thử chức năng là gì

Kiểm thử chức năng là một loại kiểm thử phần mềm nhằm xác minh chức năng của hệ thống phần mềm hoặc ứng dụng. Kiểm thử chức năng tập trung vào việc đảm bảo rằng hệ thống hoạt động theo các yêu cầu chức năng đã được chỉ định và đáp ứng các nhu cầu kinh doanh đã định.

Kiểm thử chức năng có thể giúp đảm bảo chức năng của hệ thống được xây dựng đúng yêu cầu và các chức năng hoạt động ổn định bình thường qua các lần nâng cấp.

Kiểm thử chức năng thường liên quan đến việc kiểm tra đầu vào và đầu ra của phần mềm, thao tác dữ liệu tương tác của người dùng và phản hồi của hệ thống với các tình huống và điều kiện khác nhau

Kiểm thử chức năng thường được xây dựng và thực hiện bởi các kiểm thử viên (tester). Bằng các phương pháp như kiểm thử hệ thống (system testing) hay kiểm thử hồi quy (Regression testing)....

Kiểm thử chức năng được thực hiện bằng 2 phương pháp sau:

- **Kiểm thử dựa trên yêu cầu:** Kiểm thử viên sẽ dựa vào các đặc tả yêu cầu của người dùng, khách hàng để thiết kế các kịch bản kiểm thử nhằm đảm bảo kịch bản bao phủ hết các yêu cầu, ngoài ra tester cũng dựa vào yêu cầu để xác định những phạm vi của các kịch bản kiểm thử hay những phần không phải kiểm thử
- **Kiểm thử dựa trên bối cảnh thực tế:** Kiểm thử viên sẽ dựa vào bối cảnh thực tế, các ngữ cảnh thực sự mà người sử dụng sẽ dùng sản phẩm phần mềm để thiết kế các kịch bản kiểm thử để có thể bao phủ hết các ngữ cảnh người dùng trong môi trường thực tế. Qua đó đảm bảo chất lượng và nghiệp vụ của sản phẩm phần mềm

đáp ứng tốt nhu cầu của người sử dụng.

1.1.2. Khó khăn gặp phải khi kiểm thử chức năng

Kiểm thử chức năng có thể giúp chúng ta xác nhận xem hệ thống có hoạt động đúng như yêu cầu đặc tả hay không. Có thể xác nhận phần mềm hoạt động đúng không và kết quả của kiểm định chức năng là một yếu tố quan trọng để xác định chất lượng phần mềm. Nhưng để kiểm định chức năng thủ công hiệu quả chúng ta thường gặp phải những khó khăn như sau:

Khó khăn do lượng chức năng lớn và bộ kịch bản lớn: Với các hệ thống lớn và có khối lượng lớn chức năng bộ kịch bản kiểm thử chức năng sẽ rất lớn, số lượng nghiệp vụ cần phải kiểm thử sẽ rất lớn

Nghệ vụ thay đổi mỗi lần cập nhật phần mềm yêu cầu phải kiểm thử lại chức năng của phần mềm trong một khoảng thời gian ngắn: Với các hệ thống lớn trong quá trình sử dụng các nghiệp vụ luôn được thêm mới và cập nhật khiến việc kiểm thử lại chức năng là một việc vô cùng quan trọng trong quá trình đảm bảo chất lượng phần mềm nhưng yêu cầu phát triển của khách hàng luôn rất cấp thiết cần phải hoàn thành trong một khoảng thời gian ngắn nên kéo theo áp lực về mặt thời gian của kiểm thử chức năng là rất lớn

Kiểm thử hồi quy tốn rất nhiều thời gian và nguồn lực: Việc kiểm thử hồi quy (Kiểm thử những tính năng không thay rồi để đảm bảo những thay đổi không làm ảnh hưởng đến những tính năng khác của hệ thống) tốn rất nhiều nguồn lực và thời gian của kiểm thử viên

Tốn nguồn lực kiểm thử chức năng trên nhiều môi trường khác nhau để đảm bảo chất lượng cho hệ thống: Để đảm bảo độ ổn định của hệ thống nhiều hệ thống lớn sẽ có những cơ chế chạy song song hay chia tải cho cả máy chủ ứng dụng và máy chủ cơ sở dữ liệu dẫn đến việc đảm bảo kiểm thử chức năng trên từng máy chủ trở nên rất tốn thời gian và nguồn lực.

Kiểm thử chức năng bị ảnh hưởng bởi các yếu tố con người: Kiểm thử chức năng thường được thực hiện bởi các nhân sự kiểm thử. Và các yếu tố con người như tâm lý, áp lực, chủ quan có thể ảnh hưởng lớn đến hiệu quả và kết quả của kiểm

thử chức năng.

Chi phí kiểm thử chức năng lớn: Với các hệ thống lớn cần kiểm thử chức năng trên nhiều môi trường, nhiều máy chủ cần một lượng lớn nhân viên kiểm thử cùng các thiết bị cần thiết để tiến hành kiểm thử chức năng như máy tính, điện thoại, máy tính bảngkhiến chi phí để kiểm thử chức năng rất lớn

Những hạn chế này có thể khắc phục được nếu áp dụng kiểm thử tự động để hỗ trợ cho việc kiểm thử chức năng của hệ thống.

1.2. Giới thiệu và phân tích hiện trạng kiểm thử chức năng của vnEdu

1.2.1. Hệ sinh thái giáo dục vnEdu

vnEdu là một giải pháp xây dựng trên nền tảng web công nghệ điện toán đám mây nhằm tin học hoá toàn diện công tác quản lý, điều hành trong giáo dục, kết nối gia đình, nhà trường và xã hội, góp phần nâng cao chất lượng quản lý, chất lượng dạy và học.

vnEdu là hệ sinh thái giáo dục số 1 Việt Nam, hiện đang cung cấp hơn 20 sản phẩm dịch vụ trên 29.000 trường học trên toàn quốc. Hệ sinh thái được xây dựng tuân thủ khung kiến trúc chính phủ điện tử Việt Nam 2.0 do Bộ Thông tin & Truyền thông ban hành, trong đó ứng dụng mạnh mẽ các xu hướng giáo dục hiện đại và các công nghệ 4.0. Đây cũng là hệ sinh thái giáo dục chiếm thị phần lớn nhất Việt Nam.

1.2.2. Các điểm nổi bật của hệ sinh thái vnEdu so với các giải pháp khác trên thị trường

- Dữ liệu hệ thống được liên thông đồng bộ với mã định danh của hệ thống CSDL ngành giáo dục của Bộ Giáo dục và đào tạo.
- Thị phần số 1 tại Việt Nam
- Ứng dụng Mobile có trên 1 triệu lượt tải
- Hệ sinh thái đầy đủ và toàn diện nhất với nhiều sản phẩm dịch vụ
- Đáp ứng tất cả các thông tư, nghị định của Bộ Giáo dục và đào tạo
- Triển khai theo mô hình điện toán đám mây SaaS: Các cơ sở giáo dục có thể dễ dàng khởi tạo 1 site riêng cho mình mà không phải trang bị, đầu tư máy chủ, hạ

tăng, đội ngũ vận hành....

- Lấy người học làm trung tâm, hỗ trợ tương tác giữa học sinh và giáo viên thông qua các công nghệ Conference, Bảng tương tác Whiteboard, Livestream...

THỰC TẾ TRIỂN KHAI



Hình 1.1. Tình hình triển khai thực tế của hệ sinh thái vnEdu

1.2.3. Tính năng của hệ sinh thái vnEdu

VNEDU hướng tới mục tiêu tin học hoá một cách toàn diện trong giáo dục, phục vụ nhu cầu quản lý cho các cấp quản lý giáo dục. VNEDU có thể áp dụng cho các cấp học từ mầm non tới trung học phổ thông. Các đối tượng VNEDU phục vụ bao gồm:

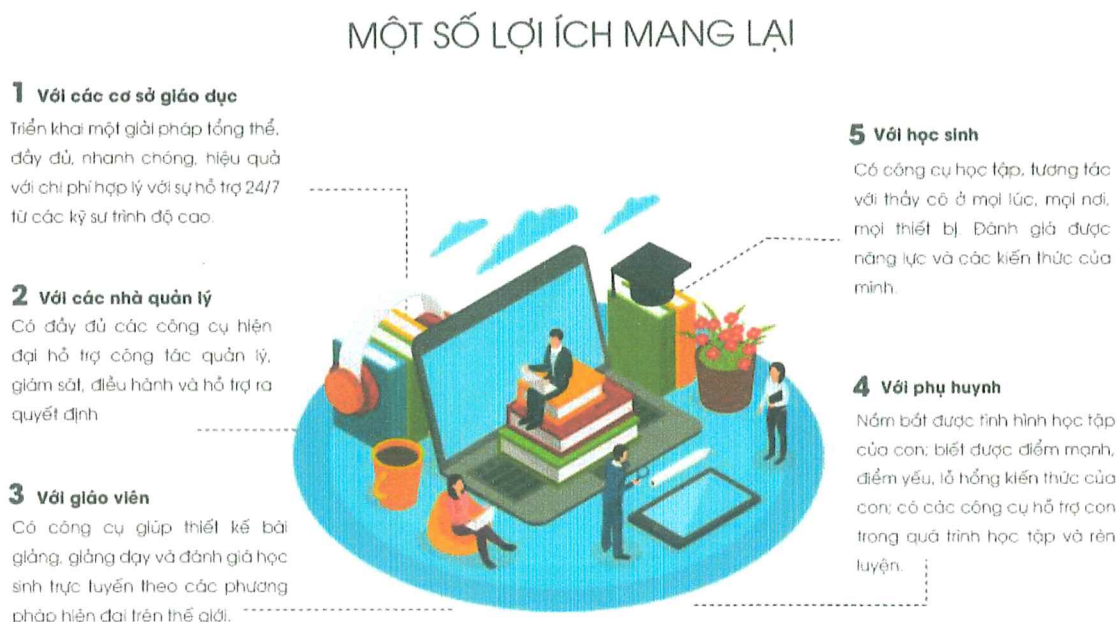
- Cấp cơ sở giáo dục :
 - Các giáo viên làm việc tại cơ sở giáo dục
 - Các nhân viên làm việc tại cơ sở giáo dục
 - Các cấp quản lý tại cơ sở giáo dục
- Cấp Phòng Giáo dục :
 - Nhân viên Phòng giáo dục
 - Cấp quản lý phòng giáo dục
- Cấp Sở giáo dục :
 - Nhân viên Sở giáo dục
 - Cấp quản lý của Sở Giáo dục

Bảng 1.1. Các phân hệ tính năng của sinh thái vn Edu

Nhóm chức năng	Mô tả	Đối tượng thực hiện
Quản trị hệ thống	Khởi tạo người dùng, phân quyền, Cấu hình các tham số quản lý...	Quản trị hệ thống
Danh mục dùng chung	Khai báo các danh mục dùng chung như tỉnh thành, quận huyện...	Nhà cung cấp ứng dụng
Hồ sơ học sinh	Quản lý thông tin hồ sơ học sinh	Quản trị hệ nhà trường, giáo viên chủ nhiệm

Nhóm chức năng	Mô tả	Đối tượng thực hiện
Quá trình học tập, rèn luyện	Các thông tin liên quan đến quá trình học tập của học sinh	Giáo viên bộ môn, giáo viên chủ nhiệm
Quản lý nhân sự/giáo viên	Quản lý hồ sơ giáo viên, phân công giảng dạy...	Quản trị nhà trường, ban giám hiệu
Thống kê báo cáo	Nơi thống kê các báo cáo chuyên môn của nhà trường	Tất cả người dùng đã đăng nhập vào hệ thống
Sổ liên lạc điện tử	Cho phép phụ huynh học sinh tra cứu kết quả học tập, rèn luyện qua trang web hoặc qua các dịch vụ tin nhắn SMS, Voice	Phụ huynh, học sinh
Điều hành	Cho phép thông báo điều hành qua tin nhắn SMS; công văn đến, công văn đi	Lãnh đạo các cấp và nhà trường
Quản lý tiện ích khác	Quản lý văn bản, bảng tin, dịch vụ, tài sản	Quản trị nhà trường

1.2.4. Một số lợi ích mang lại từ hệ sinh thái vnEdu



Hình 1.2. Các lợi ích khi sử dụng hệ sinh thái vnEdu

- Hình thành một cách thức quản lý mới, khoa học cho nhà trường.

- Giảm bớt công việc thủ công, các thủ tục hành chính trong quản lý, tiết kiệm thời gian, chi phí, nâng cao hiệu quả công việc.

- vnEdu là một kênh liên lạc giúp phụ huynh nắm bắt dễ dàng, trực tiếp và nhanh chóng kết quả học tập, rèn luyện của con em để kịp thời khích lệ, uốn nắn con em mình.

- Việc tính toán, tổng hợp số học lực, hạnh kiểm... được hệ thống thực hiện từ động tuân theo các quy định của bộ giáo dục ban hành nên tránh được sai sót do việc tính toán thủ công.

- Các số liệu báo cáo được hệ thống cung cấp một cách nhanh chóng và chính xác.

- Hệ thống tích hợp các mẫu báo cáo thống kê EMIS theo các giai đoạn của năm học, các mẫu báo cáo về hồ sơ và điểm của học sinh tương thích với phân hệ quản lý học sinh VEMIS và quản lý điểm.

- Nhà trường có thể chủ động trong việc in ấn một số mẫu sổ gọi tên và ghi điểm, mẫu sổ theo dõi và đánh giá học sinh tiểu học.

- Tổ chức và quản lý kỳ thi một cách nhanh chóng hiệu quả.

1.2.5. Hiện trạng kiểm thử chức năng của hệ sinh thái vnEdu

Hiện nay hệ thống vnEdu đang được phát triển theo mô hình Agile và được triển khai trên quy mô toàn quốc. Lượng chức năng triển khai trên một sprint của vnEdu là rất lớn (trên 300 yêu cầu được xử lý và thay đổi trong mỗi sprint). Mỗi sprint kéo dài 2 tuần cho cả phát triển chức năng và kiểm thử vì vậy thời gian dành cho kiểm thử chức năng rất hạn chế.

Hệ thống vnEdu được sử dụng cho các sở giáo dục toàn quốc nên sẽ có các kỳ cao điểm ví dụ như khi thi học kỳ, khi tổng kết điểm, khi thay đổi thông tư với những kỳ cao điểm số lượng yêu cầu phải cập nhật sẽ tăng lên rất nhiều (gấp 2, gấp 3 lần bình thường) dẫn đến áp lực của kiểm thử viên để đảm bảo chất lượng hệ thống sẽ tăng lên rất lớn.

Với khối lượng công việc vô cùng lớn (cả khi bình thường và khi cao điểm) số lượng kiểm thử viên của trung tâm Giải pháp Giáo dục điện tử rất hạn chế (5 kiểm thử viên) nên khối lượng công việc của mỗi kiểm thử viên khi kiểm thử chức năng là

vô cùng lớn.

Hệ thống vnEdu được phát triển đến nay đã hơn 10 năm, số lượng chức năng và nghiệp vụ rất lớn nên khi thay đổi cần phải kiểm thử hồi quy để đảm bảo chất lượng và độ ổn định của hệ thống. vnEdu được phát triển theo mô hình phát triển phần mềm Agile nên thay đổi được cập nhật liên tục khiến việc kiểm thử hồi quy cũng cần được thực hiện liên tục.

Hạ tầng của vnEdu bao gồm: 11 máy chủ ứng dụng và 10 máy chủ cơ sở dữ liệu. Với hạ tầng vô cùng lớn mỗi lần cập nhật thay đổi cần phải cập nhật toàn bộ trên các máy chủ ứng dụng và máy chủ cơ sở dữ liệu. Rủi ro khi cập nhật có thể mất đồng bộ hoặc thiếu cập nhật trên một số máy chủ. Công việc của kiểm thử viên là phải kiểm tra và đảm bảo hệ thống trên toàn bộ các máy chủ cả ứng dụng lẫn cơ sở dữ liệu. Dẫn đến công việc của kiểm thử viên là vô cùng lớn.

Số lượng người dùng của vnEdu rất lớn (Như học viên đã trình bày ở phần trước bao gồm gần 9 triệu học sinh và hơn 900.000 giáo viên) nên độ ổn định của hệ thống rất quan trọng. Ngoài ra với lượng người sử dụng lớn nên nếu phát hiện ra lỗi muộn (hoặc để người dùng phát hiện ra lỗi) sẽ dẫn đến mất uy tín và tổn thất nghiêm trọng.

Hiện nay vnEdu chưa áp dụng công cụ kiểm thử tự động nào dẫn đến rủi ro không đảm bảo chất lượng. Ví dụ như sau:

Ngày **15-06-2021**: Bị mắc lỗi bắt buộc nhập mã MOET ID cho học sinh khi tạo và sửa học sinh do ảnh hưởng của ticket mầm non

Ngày **21-11-2021**: Bị mắc lỗi không tạo được lớp sau khi sửa thêm một số thông tin của lớp học

Ngày **16-12-2021**: Bị mắc lỗi không duyệt được giáo án giáo viên của chức năng quản lý giáo án.

.....

Các lỗi trên đều được phát hiện ra muộn sau khi triển khai lên môi trường thật của khách hàng và làm ảnh hưởng đến chất lượng của sản phẩm dịch vụ của VNPT với khách hàng.

Trong quá trình tổng hợp khi làm việc tại Trung tâm Giáo dục điện tử, học viên nhận thấy chủ yếu các lỗi phát sinh là ở trên nền ứng dụng Web tại các chức năng liên quan đến học sinh, giáo viên, lớp học và thường phát hiện ra muộn do các yếu tố về con người như: chủ quan, không đủ thời gian regression test.....

Vì vậy qua những điều trên, học viên thấy việc áp dụng kiểm thử tự động vào kiểm thử chức năng cho các chức năng chính của vnEdu liên quan đến Giáo viên, lớp học và học sinh là vô cùng cần thiết. Qua đó để giải quyết bài toán này học viên đã đề xuất đề án “**NGHIÊN CỨU PHƯƠNG PHÁP KIỂM THỬ CHỨC NĂNG TỰ ĐỘNG VÀ ÁP DỤNG CHO HỆ THỐNG VNEDU**”

1.3. Kiểm thử chức năng tự động:

1.3.1. Giới thiệu

Kiểm thử tự động là quá trình sử dụng các công cụ hay phần mềm cần thiết để thực hiện các kịch bản kiểm thử một cách tự động hóa thay vì làm thủ công bằng sức người. Nó nhằm mục đích tăng cường hiệu suất và hiệu quả của quá trình kiểm thử phần mềm bằng cách tự động hóa các bước kiểm thử, từ việc chạy kiểm thử đến tổng hợp kết quả.

Kiểm thử tự động thường được thực hiện bằng cách sử dụng các công cụ và kỹ thuật và công nghệ tự động hóa. Các kịch bản kiểm thử có thể bao gồm kiểm thử đơn vị, Kiểm thử chức năng, kiểm thử tích hợp, kiểm thử hành vi người dùng và kiểm thử hệ thống.

Lợi ích của kiểm thử tự động bao gồm tăng cường sự nhất quán, tiết kiệm thời gian, chi phí giảm sai sót do con người và tăng cường khả năng tái sử dụng của các kịch bản kiểm thử.

Vì các kịch bản kiểm thử được cài đặt trước trên các công cụ tự động hóa nên kiểm thử tự động thích hợp với những loại kiểm thử cần được lặp đi lặp lại nhiều lần: như kiểm thử hồi quy, kiểm thử trên các môi trường khác nhau, các máy chủ khác nhau...

Với các khó khăn của kiểm thử chức năng thủ công mà học viên đã trình bày trong 1.1.2. Các khó khăn đó đều được giải quyết khi áp dụng kiểm thử chức năng tự

động nói cách khác chính những khó khăn khi kiểm thử chức năng thủ công nêu trên là ưu điểm của kiểm thử chức năng tự động. Cụ thể như:

+ **Nghịệp vụ thay đổi mỗi lần cập nhật phần mềm yêu cầu phải kiểm thử lại chức năng của phần mềm trong một khoảng thời gian ngắn:** Với khó khăn này khi áp dụng kiểm thử tự động với những kịch bản thông thường chúng ta có thể áp dụng để cho các công cụ tự động hóa chạy, có thể kiểm chạy kiểm thử chức năng tự động vào ngoài giờ làm việc, như vậy kiểm thử viên có thể rút ngắn thời gian kiểm thử chức năng lại khá nhiều.

+ **Kiểm thử hồi quy tốn rất nhiều thời gian và nguồn lực:** Với kiểm thử tự động việc kiểm thử hồi quy có thể được hỗ trợ bằng kiểm thử tự động. Việc tự động hóa kiểm thử hồi quy sẽ giảm đáng kể thời gian và nguồn lực so với kiểm thử chức năng thủ công.

+ **Tốn nguồn lực kiểm thử chức năng trên nhiều môi trường khác nhau để đảm bảo chất lượng cho hệ thống:** Với các môi trường khác nhau, máy chủ khác nhau công cụ tự động hóa có thể chạy lại, chạy liên tục để đảm bảo chất lượng của hệ thống.

+ **Kiểm thử chức năng bị ảnh hưởng bởi các yếu tố con người:** Kiểm thử chức năng tự động không hề bị ảnh hưởng bởi yếu tố con người, hệ thống tự động hóa sẽ chạy lại chính xác 100% những thao tác và kịch bản do kiểm thử viên đã quy định sẵn.

+ **Chi phí kiểm thử chức năng lớn:** Có thể việc triển khai kiểm thử tự động ban đầu sẽ tốn chi phí và nguồn lực lớn nhưng tính trên một khoảng thời gian dài với ưu thế giảm thời gian và nguồn lực kiểm thử thì càng về sau chi phí kiểm thử khi áp dụng kiểm thử tự động sẽ càng giảm và tạo ra hiệu quả kinh tế lớn hơn so với kiểm thử chức năng thủ công.

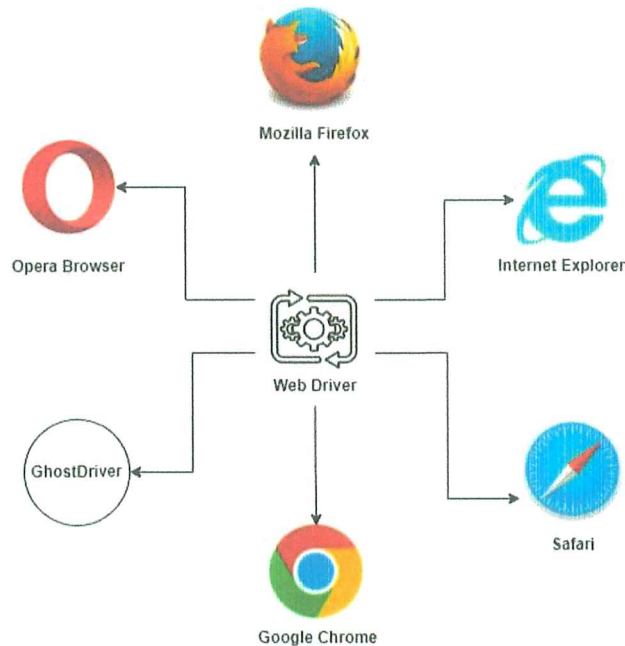
1.3.2. Một số công cụ kiểm thử chức năng tự động phổ biến hiện nay

Hiện nay trên thị trường có rất nhiều công cụ hỗ trợ việc kiểm thử tự động. Mỗi công cụ có ưu và nhược điểm riêng qua quá trình nghiên cứu học viên sẽ giới thiệu một số công cụ đang phổ biến hiện nay.

1. Selenium Webdriver:

Là một automation framework dành cho web với WebDriver bạn có thể thực hiện các phiên kiểm thử của mình trên các trình duyệt khác nhau mà không giới hạn trên trình duyệt nào.

WebDriver sẽ cung cấp công cụ hỗ trợ tốt hơn trong việc coding để tạo các kịch bản kiểm thử. Người dùng có thể sử dụng ngay các phương thức điều kiện như if-then-other, hoặc các vòng lặp như do-while



Hình 1.3. Các trình duyệt được hỗ trợ bởi Webdriver

Ưu điểm của Selenium Webdriver:

- Hỗ trợ được đa trình duyệt, đa ngôn ngữ
- Độ tùy biến cao
- Có thể tích hợp với các công cụ và framework kiểm thử khác như TestNG, Junit, Cucumber.

- Miễn phí vì là mã nguồn mở

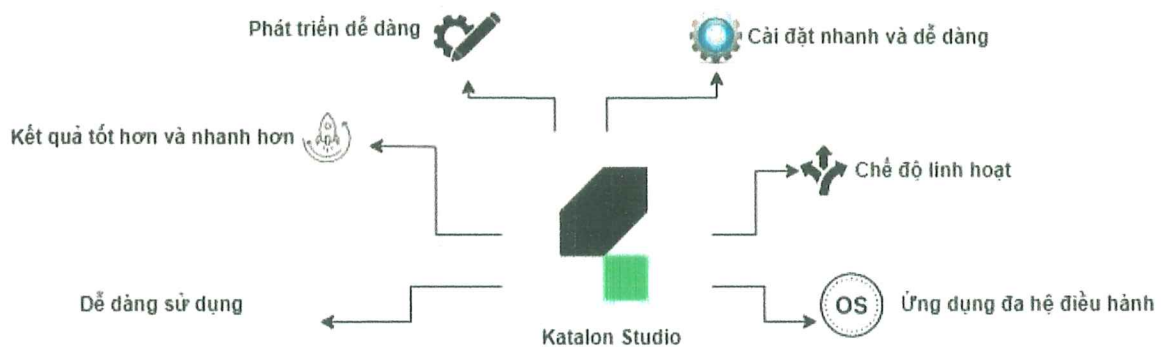
Nhược điểm của Selenium Webdriver:

- Người sử dụng phải biết kỹ thuật XPath hoặc CSS selector để xác định vị trí của các phần tử trong DOM, điều này gây khó khăn với những người không có kỹ năng lập trình.

- Yêu cầu người sử dụng phải có kỹ năng lập trình nhất định.
- Không cung cấp sẵn các báo cáo trực quan điều này làm giảm tính thẩm mỹ và hiệu quả sử dụng selenium trong hoạt động kiểm thử
- Yêu cầu trình duyệt và webDriver phải tương thích: Selenium WebDriver tương thích với phiên bản cụ thể của trình duyệt, điều này có thể gây ra các lỗi với các phiên bản mới của trình duyệt.
- Khả năng quản lý các cửa sổ và các tab của trình duyệt không tốt nên thường gặp khó khăn với các kịch bản kiểm thử phức tạp.

2. Katalon Studio:

Katalon là một sản phẩm của công ty KMS Technology. Dành cho việc kiểm thử tự động các ứng dụng Web và Mobile. Katalon Studio được đánh giá là một công cụ kiểm thử tự động đa năng và mạnh mẽ được thiết kế để hỗ trợ việc kiểm thử tự động cho các ứng dụng Web, di động và API một cách hiệu quả. Với giao diện người dùng thân thiện và dễ sử dụng, Katalon Studio là một lựa chọn lý tưởng cho các nhân viên kiểm thử cả chuyên và không chuyên



Hình 1.4. Các điểm mạnh của Katalon Studio

Katalon Studio cung cấp nhiều tính năng hữu ích như ghi và chạy lại các kịch bản kiểm thử tự động một cách nhanh chóng và dễ dàng, quản lý dự án kiểm thử, tạo báo cáo chi tiết về kết quả kiểm thử, tích hợp với các công cụ quản lý phổ biến như Jira, Jenkin hay Git.

Với sự tích hợp với Selenium và Appium, Katalon Studio cho phép người sử dụng kiểm thử ứng dụng trên nhiều trình duyệt Web và cả các thiết bị di động một

cách linh hoạt. Đồng thời việc hỗ trợ kiểm thử API cũng là một điểm mạnh của công cụ này, giúp cho người kiểm thử có thể kiểm thử toàn diện từ giao diện người dùng đến các dịch vụ phía sau.

Ưu điểm của Katalon:

- Giao diện người dùng thân thiện: Katalon Studio có một giao diện người dùng trực quan và dễ sử dụng, giúp kiểm thử viên sử dụng dễ dàng tạo và quản lý các kịch bản kiểm thử.

- Hỗ trợ đa nền tảng và đa ngôn ngữ lập trình: Katalon Studio hỗ trợ kiểm thử trên nhiều nền tảng và ngôn ngữ lập trình khác nhau như: Windows, macOS, Linux và các ngôn ngữ lập trình như Java, Groovy.

- Tích hợp với các công cụ quản lý dự án và phiên bản: Katalon có thể dễ dàng tích hợp với các công cụ quản lý như Jira, Jenkins, Git để tự động hóa quy trình kiểm thử

- Ghi và chạy lại kịch bản kiểm thử (Record and Replay): Katalon Studio cung cấp công cụ ghi và chạy lại kịch bản kiểm thử một cách dễ dàng và nhanh chóng giúp giảm thời gian xây dựng kịch bản kiểm thử tự động.

- Dễ dàng cài đặt và sử dụng: Katalon đóng gói sản phẩm và tổ chức theo mô hình sẵn thuận tiện cho người sử dụng.

- Cộng đồng người sử dụng đông đảo trên toàn thế giới.

- Cập nhật thường xuyên và hỗ trợ chuyên nghiệp: Vì là sản phẩm thu phí nên Katalon thường xuyên được cập nhật với các tính năng mới và sửa lỗi, cùng với sự hỗ trợ chuyên nghiệp từ đội ngũ phát triển.

- Nhược điểm của Katalon:

- Giá thành cao với phiên bản trả phí: Phiên bản trả phí của Katalon có giá cao nên đây là một hạn chế đối với các công ty có mức kinh phí hạn chế (Giá bán của Katalon Studio hiện tại là 167\$/tháng và 1999\$/năm).

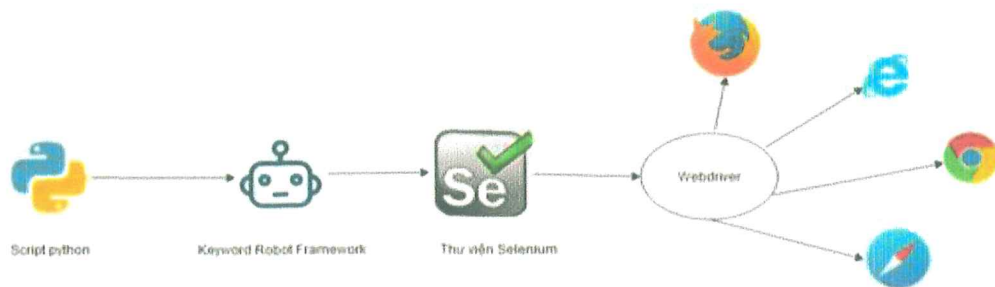
- Yêu cầu tài nguyên hệ thống tương đối mạnh: Katalon Studio yêu cầu tài nguyên hệ thống tương đối mạnh để chạy và quản lý các kịch bản kiểm thử phức tạp.

- Khó khăn trong việc tùy chỉnh sâu và phức tạp: Mặc dù Katalon linh hoạt và có độ mở rộng cao, nhưng việc tùy chỉnh sẽ phải theo mô hình và quy tắc của Katalon

nên gây khó khăn cho những người sử dụng ít kinh nghiệm.

- Các gói hay các chức năng bổ sung đều phải mua riêng gây tốn kém kinh phí cho người sử dụng ví dụ như muốn chạy Command line để có thể chạy CI/CD thì phải mua thêm Katalon RunTime Engine hay muốn xem báo cáo online phải mua thêm gọi Katalon TestOps

3. RobotFramework:



Hình 1.5. Mô hình hoạt động của RobotFramework

RobotFramework là một công cụ kiểm thử tự động mã nguồn mở được phát triển và cho ra mắt từ năm 2005 bởi Pekka Klarck. Ý tưởng ban đầu của Robot Framework là một dự án nghiên cứu về tự động hóa kiểm thử và tự động hóa việc viết tài liệu kiểm thử.

Ban đầu RobotFramework được viết bằng ngôn ngữ lập trình Python và được phát triển như một công cụ giả lập việc tự động với các tác vụ kiểm thử phức tạp. Từ khi được giới thiệu lần đầu tiên, Robot Framework đã dần trở nên phổ biến trong cộng đồng kiểm thử và phát triển phần mềm, được nhiều tổ chức và doanh nghiệp lựa chọn để tự động hóa quy trình kiểm thử của họ.

Với sự linh hoạt, dễ dàng sử dụng và tính mở rộng cao, Robot Framework đã ngày càng cải tiến và mở rộng tính năng qua các phiên bản mới. Cộng đồng người dùng của Robot Framework cũng đã phát triển mạnh mẽ, cung cấp hỗ trợ, chia sẻ kiến thức và tạo ra các thư viện mở rộng để tối ưu hóa việc sử dụng công cụ này trong quá trình kiểm thử.

Hiện nay, Robot Framework là một trong những công cụ kiểm thử tự động phổ biến nhất trên thị trường, được sử dụng rộng rãi trong các dự án phần mềm ở nhiều

lĩnh vực khác nhau. Điều này nói lên sức mạnh và tính ổn định của Robot Framework trong việc hỗ trợ quy trình kiểm thử tự động cho ứng dụng web và mobile.

Ưu điểm của RobotFramework:

- Cú pháp đơn giản và dễ đọc: RobotFramework sử dụng cú pháp dễ hiểu, gần gũi với ngôn ngữ tự nhiên, giúp người dùng dễ dàng viết và hiểu mã kiểm thử
- Hỗ trợ đa nền tảng và đa ngôn ngữ lập trình: RobotFramework không bị ràng buộc bởi một ngôn ngữ lập trình cụ thể, cho phép sử dụng Python, Java hoặc cả JavaScript.
- Tích hợp linh hoạt: RobotFramework có khả năng tích hợp nhiều công cụ và Framework kiểm thử khác nhau như Selenium, Appium hay các thư viện mở rộng khác
- Cộng đồng lớn và hỗ trợ tốt: RobotFramework với sự phát triển lâu dài đã có một cộng đồng người sử dụng lớn và hỗ trợ tốt.
- Hỗ trợ kiểm thử đa dạng các loại ứng dụng: RobotFramework cùng các thư viện hỗ trợ có thể hỗ trợ kiểm thử đa dạng các loại ứng dụng như Web, Mobile application hay thậm chí cả ứng dụng window.
- Báo cáo chi tiết và đồng nhất: RobotFramework cung cấp các báo cáo kiểm thử chi tiết và dễ đọc, giúp người đọc hiểu rõ kết quả kiểm thử.
- Miễn phí: RobotFramework là một framework mã nguồn mở nên các thư viện và mã nguồn đều được cung cấp miễn phí

Nhược điểm của RobotFramework:

- Cài đặt phức tạp hơn các phần mềm trả phí: RobotFramework cài đặt bằng các câu lệnh của python nên nhiều bước và phức tạp hơn các phần mềm trả phí đã được đóng gói.
- Yêu cầu có kiến thức lập trình để sử dụng chuyên sâu: RobotFramework có độ tùy biến cao nên hỗ trợ người dùng có thể tùy biến cách tổ chức dự án hoặc viết thêm các thư viện. Nhưng để làm được thì yêu cầu người sử dụng cần có kiến thức lập trình nhất định.
- Độ tin cậy chưa được đảm bảo: Vì là phần mềm mã nguồn mở nên thư viện

chưa được kiểm soát và đánh giá chất lượng toàn diện trước khi đưa lên web. Người dùng cần có kỹ năng tìm kiếm và hiểu biết để tìm ra các thư viện tốt và có độ ổn định cao.

Dưới đây là bảng so sánh một số mặt của các nền tảng kiểm thử tự động thịnh hành hiện nay

Bảng 1.2. So sánh các phần mềm kiểm thử tự động hiện nay

Đặc tính	RobotFramework	Selenium	Katalon Studio
Cài đặt	Cài đặt đơn giản	Cài đặt phức tạp	Cài đặt đơn giản
Khả năng triển khai	Triển khai tốt và đơn giản	Triển khai khó vì hoàn toàn phải tự code	Triển khai tốt và đơn giản
Khả năng bảo trì	Bảo trì tốt, nhanh	Bảo trì tốt nhưng yêu cầu người sử dụng phải biết code	Triển khai tốt và đơn giản
Khả năng nâng cấp	Có thể nâng cấp theo 2 cách là thêm keyword hoặc thêm thư viện	Chỉ có thể code thêm tính năng	Có thể nâng cấp theo 2 cách là thêm keyword hoặc thêm thư viện
Chi phí	Miễn phí	Miễn phí	Có phí
Cộng đồng	Cộng đồng lớn nhiều người sử dụng	Cộng đồng lớn nhiều người sử dụng	Không bằng các cộng đồng mã nguồn mở
Báo cáo	Hỗ trợ báo cáo dễ đọc, dễ hiểu	Không có sẵn báo cáo, cần dùng báo cáo của bên thứ 3 hoặc tự phát triển	Hỗ trợ báo cáo dễ đọc, dễ hiểu

1.3.2. Khó khăn khi sử dụng các công cụ kiểm thử tự động

- Yếu tố con người: Phần lớn kiểm thử viên là những người có kỹ năng lập

trình hạn chế. Có cả những kiểm thử viên là người trái ngành họ sẽ gặp nhiều khó khăn nếu sử dụng các phần mềm yêu cầu khả năng lập trình để sử dụng.

- Với mô hình Agile cần tích hợp liên tục hệ thống thay đổi nhiều tốn công sức maintain: Khi thay đổi nhằm mục đích sửa lỗi hoặc nâng cấp tính năng mới theo yêu cầu của người sử dụng có thể dẫn đến các kịch bản kiểm thử tự động không còn có thể hoạt động nữa. Khi đó kiểm thử viên sẽ phải tiến hành chỉnh sửa, cập nhật để các kịch bản kiểm thử tự động phù hợp với mã nguồn mới được cập nhật. Điều này trong mô hình phát triển phần mềm Agile rất khó khăn vì tần suất cập nhật phần mềm cao, các thay đổi được cập nhật liên tục dẫn đến cần tốn một nguồn lực rất lớn của kiểm thử viên để cập nhật, điều chỉnh kịch bản kiểm thử tự động.

- Hệ thống lớn cần kiểm soát khi lượng testcases tăng nhiều theo thời gian: Các dự án cần dùng kiểm thử tự động thường là các dự án lớn và phát triển trong một thời gian dài điều này khiến sau một thời gian khối lượng các kịch bản kiểm thử tự động ngày càng lớn. Nếu không có cách tổ chức quản lý kịch bản kiểm thử tự động tốt sẽ khiến kịch bản kiểm thử bị chồng chéo và sử dụng kém hiệu quả.

- Báo cáo có sẵn không đầy đủ, khó khăn trong việc đọc hiểu và debug script kiểm thử tự động: Một số công cụ kiểm thử tự động không được tích hợp báo cáo hoặc báo cáo khó sử dụng khiến người dùng gặp khó khăn trong việc thống kê hoặc phân tích lỗi khi kịch bản chạy có vấn đề xảy ra.

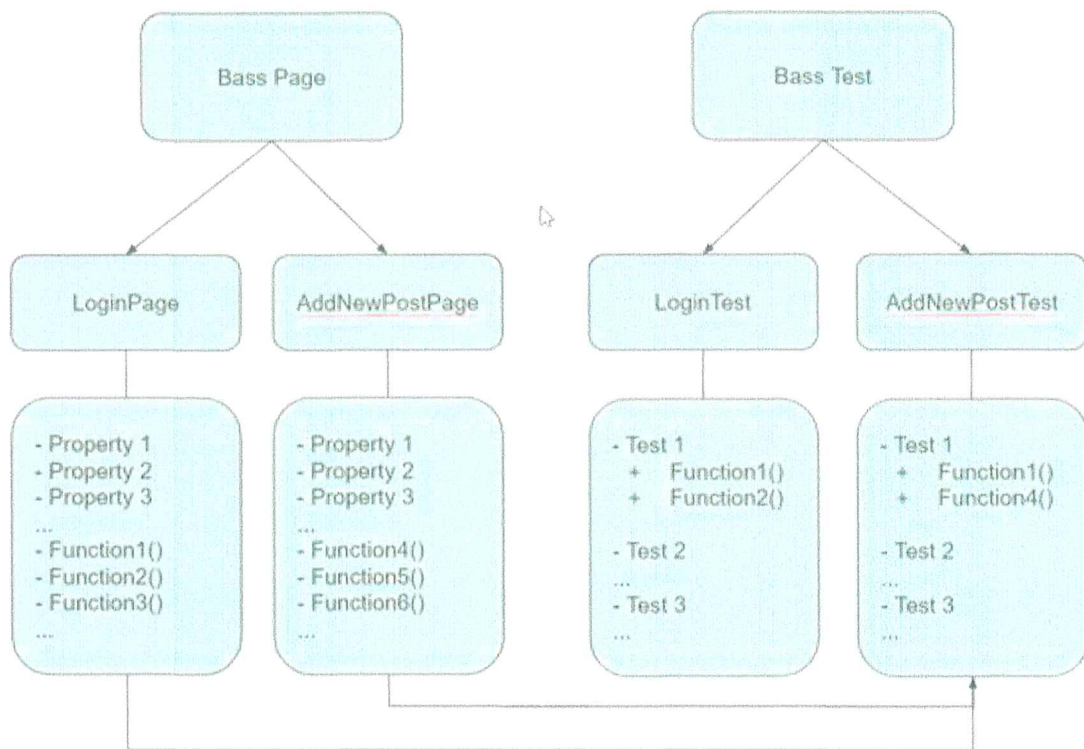
1.3.3. Mô hình tổ chức dự án kiểm thử tự động thông dụng hiện nay:

Để giải quyết khó khăn trong việc quản lý kịch bản kiểm thử tự động để tiện cho việc duy trì và bảo dưỡng cần có một mô hình xây dựng và quản lý kịch bản kiểm thử tự động hợp lý. Mô hình **Page Object Model (POM)** là một mô hình phổ biến trong các dự án triển khai kiểm thử chức năng tự động. Được cộng đồng thừa nhận và sử dụng rộng rãi.

Page Object Model (POM): là 1 design pattern giúp mô hình hóa các pages, hoặc các thành phần trong các màn hình của trang web thành mỗi đối tượng riêng biệt. Mỗi thành phần sẽ gói gọn tất cả các hành động và thuộc tính của thành phần đó.

Để phù hợp với hệ thống cần Page Object Model (POM) cần tách biệt ra 2 phần riêng biệt một là thành phần đối tượng trên màn hình và hành động của đối tượng đó.

Ví dụ về Page Object Model:



Hình 1.6. Mô hình Page Object Model (POM)

Theo Page object model (POM) có nghĩa là sẽ tổ chức kịch bản kiểm thử chức năng tự động một màn hình của hệ thống sẽ bao gồm 2 thành phần: Một đối tượng đại diện cho các thành phần của màn hình và một đại diện cho các hành động có thể thực hiện của màn hình đó.

Ưu điểm lớn nhất là khi cần thay đổi các kịch bản kiểm thử đã xây dựng nếu có sự thay đổi về giao diện của các màn hình, kiểm thử viên sẽ không phải thay đổi nhiều nơi hoặc tìm kiếm quá phức tạp tiềm ẩn nguy cơ nhầm lẫn và thiếu sót mà chỉ cần thay đổi một đối tượng đã được phân tách theo cấu trúc.

Hiện nay mô hình POM là mô hình phổ biến và được sử dụng rộng rãi trong việc thiết kế và triển khai kiểm thử tự động

KẾT LUẬN CHƯƠNG

Trong Chương 1, đề án trình bày và giới thiệu tổng quan các nghiên cứu về kiểm thử phần mềm, kiểm thử chức năng cũng như các công cụ kiểm thử chức năng tự động đang được sử dụng trên thị trường cùng điểm mạnh và yếu của chúng.

CHƯƠNG 2. ĐỀ XUẤT PHƯƠNG PHÁP KIỂM THỬ TỰ ĐỘNG CHO HỆ SINH THÁI VNEDU

Chương này sẽ trình bày về hướng đề xuất để xây dựng bộ template script kiểm thử tự động để triển khai trên diện rộng cho các hệ thống và áp dụng để xây dựng bộ script kiểm thử cho hệ sinh thái vnEdu

2.1. Đánh giá về việc áp dụng công cụ kiểm thử tự động trong thực tế

- Hiện nay trên thị trường các công ty phần mềm đang phát triển phần mềm theo 2 hướng sau:

- **Tự phát triển công cụ kiểm thử tự động:** Cách này tốn cần đầu tư thời gian, có độ tùy biến cao vì tất cả mã nguồn, kịch bản kiểm thử do người dùng chuẩn bị. Được tự chọn ngôn ngữ lập trình mà mình mạnh để phát triển. Nhưng bất cập của phương pháp này là yêu cầu có nhóm kiểm thử tự động có nền tảng về lập trình tương đối tốt, không có cộng đồng hỗ trợ. Khó khăn trong việc mở rộng, chuyển giao và bảo trì các kịch bản kiểm thử. Vì các kịch bản kiểm thử được lập trình bởi các nhà phát triển nên mang tư duy và luồng suy nghĩ của người lập trình nên khi người khác tiếp quản bảo trì hay nhận chuyển giao sẽ cần một trình độ nhất định và cần một thời gian để làm quen và sử dụng.

- **Sử dụng công cụ hay nền tảng có sẵn:** Cách này có chi phí rẻ hơn các công cụ và nền tảng có sẵn có những công cụ có thể hỗ trợ những người không có thể mạnh về lập trình vẫn có thể viết kịch bản kiểm thử tự động, dễ dàng cài đặt, dễ dàng chuyển giao và bảo trì. Vì là nền tảng có sẵn nên một số công cụ thực sự nổi trội sẽ có nhiều người dùng và cộng đồng hỗ trợ đông đảo. Nhược điểm của cách phát triển này là một số công cụ và nền tảng hỗ trợ cách tổ chức kịch bản kiểm thử nên người dùng phải làm quen và sử dụng theo cách công cụ hỗ trợ. Một số công cụ chỉ hỗ trợ một số ngôn ngữ lập trình nhất định.

Cả hai cách đều có ưu và nhược điểm riêng tùy thuộc vào công ty và đặc thù của dự án để chọn phát triển kiểm thử tự động chức năng theo hướng nào. Với các dự

án yêu cầu tính bảo mật cao, chi phí dành cho kiểm thử tự động dồi dào, đội ngũ nhân sự có khả năng lập trình tốt thì có thể cân nhắc đi theo hướng tự phát triển công cụ kiểm thử tự động. Ngược lại với những dự án kinh phí hạn chế, đội ngũ kiểm thử viên có kiến thức lập trình hạn chế, cần triển khai kiểm thử tự động trong thời gian ngắn, không yêu cầu các cases quá phức tạp thì để đi theo hướng sử dụng công cụ hay nền tảng có sẵn là một lựa chọn tốt.

2.2. Các bước xây dựng đề xuất kiểm thử chức năng tự động cho hệ thống vnEdu:

Áp dụng kiểm thử chức năng tự động là một công việc tốn nhiều công sức và nguồn lực để áp dụng một cách hiệu quả và tiết kiệm công sức cần nghiên cứu đánh giá và có một quy trình triển khai chính xác. Qua thời gian làm việc và đánh giá điều kiện thực tế, học viên đề xuất sẽ áp dụng kiểm thử theo các bước:

- 1) **Đánh giá thực trạng, yêu cầu và chọn lựa cách thức triển khai:** Quá trình này học viên sẽ dựa vào nghiên cứu thực trạng của hệ thống vnEdu, các xu hướng áp dụng kiểm thử chức năng tự động trên thị trường, các công cụ kiểm thử chức năng tự động để chọn ra hướng đi cụ thể như: Tự phát triển hay dùng công cụ có sẵn, chọn công cụ gì, ngôn ngữ nào.....
- 2) **Chọn mô hình triển khai:** Kiểm thử chức năng tự động sẽ vô cùng hiệu quả khi dùng cho trong thời gian dài. Thời gian càng dài thì khối lượng kịch bản sẽ càng lớn nếu không được tổ chức một cách khoa học sẽ khó có thể quản lý. Ngoài ra khi dự án lớn yêu cầu cập nhật nhiều không tổ chức khoa học sẽ dẫn đến việc khó chỉnh sửa kịch bản tốn nguồn lực để duy trì kịch bản làm cho việc triển khai kiểm thử chức năng tự động không hiệu quả.
- 3) **Phân tích thuận lợi, khó khăn khi áp dụng kiểm thử tự động cho dự án:** Sau khi chọn mô hình triển khai chúng ta cần đánh giá lại những thuận lợi và khó khăn trong thực tế khi áp dụng công cụ kiểm thử tự động.
- 4) **Phân loại những tính năng có thể kiểm thử tự động và những tính năng không áp dụng kiểm thử tự động:** Trong các hệ thống có một số nhóm

chức năng, một số màn hình rất phù hợp để triển khai kiểm thử tự động nhưng có một số tính năng rất khó để xây dựng và triển khai kiểm thử tự động. Cần xác định và phân loại các tính năng để đưa ra chiến lược triển khai kiểm thử chức năng tự động một cách hợp lý.

- 5) **Xây dựng triển khai bộ template kiểm thử tự động:** Dựa vào hướng phát triển, công cụ, ngôn ngữ, mô hình và các chức năng đã chọn ở các bước trên học viên sẽ nghiên cứu tổ chức xây dựng một bộ template để có thể dễ dàng triển khai kiểm thử chức năng tự động. Thuận tiện cho việc chuyển giao, triển khai, bảo trì kịch bản kiểm thử tự động. Đặc biệt với bộ template mẫu kiểm thử viên có thể dễ dàng áp dụng để triển khai kiểm thử tự động cho các dự án.
- 6) **Xây dựng các thư viện, các hàm, các keyword và báo cáo hỗ trợ cho việc kiểm thử chức năng tự động:** Sau khi đã xây dựng một bộ template có thể triển khai áp dụng cho đa dạng dự án học viên sẽ đề xuất mở rộng viết một số thư viện, hàm, từ khóa (keyword) để hỗ trợ tốt hơn cho kiểm thử viên và người sử dụng như đọc ghi file excel, thông báo hay báo cáo excel
- 7) **Đánh giá và xây dựng báo cáo:** Trong việc triển khai kiểm thử chức năng tự động, báo cáo là một thành phần rất quan trọng. Báo cáo thể hiện các kết quả khi chạy có thể cho người dùng cái nhìn tổng quan nhất về quá trình cũng như kết quả của kiểm thử tự động ở mỗi lần chạy. Báo cáo cũng hỗ trợ cho việc debug. Nếu báo cáo không tốt, khó hiểu thì sẽ ảnh hưởng lớn đến hiệu quả của việc sử dụng công cụ kiểm thử tự động.

Qua đánh giá thử nghiệm và áp dụng 7 bước kể trên vào xây dựng kiểm thử tự động cho một số chức năng trong hệ thống và thu được kết quả tích cực. Học viên nhận thấy 7 bước kể trên thích hợp để triển khai xây dựng kiểm thử tự động cho hệ thống vnEdu vì các lý do sau :

- 7 bước đã xây dựng được một quy trình bài bản để triển khai kiểm thử tự động
- Qua mô hình đề xuất đã triển khai được kiểm thử tự động bài bản dễ dàng mở rộng

- Xây dựng đc các chức năng cốt lõi cần thiết để dùng chung cũng như mô hình báo cáo hiệu quả
 - Bài bản xuyên suốt dễ tiếp cận và sử dụng cho các kiểm thử viên
- Phù hợp để triển khai cho hệ sinh thái vnEdu

2.3. Đề xuất giải pháp kiểm thử tự động cho hệ thống vnEdu:

2.3.1. Đánh giá thực trạng, yêu cầu và chọn lựa cách thức triển khai:

Với tình hình hiện trạng của hệ thống vnEdu đã phân tích tại chương 1. Việc triển khai kiểm thử tự động chức năng là một nhu cầu cấp thiết.

Các chuyên viên kiểm thử của phòng Quản lý chất lượng trung tâm Giáo dục điện tử đều là các chuyên viên chuyên về kiểm thử thủ công có kiến thức lập trình hạn chế. Nên hướng sử dụng công cụ và nền tảng có sẵn là lựa chọn tốt nhất.

Các tiêu chí để chọn công cụ và nền tảng dựa trên hiện trạng của vnEdu:

- Kinh phí hạn chế
- Cài đặt dễ dàng
- Không quá phức tạp trong việc triển khai và chuyển giao
- Dễ dàng mở rộng
- Vì hệ thống vnEdu là hệ thống phát triển từ rất lâu (> 10 năm) nên logic rất phức tạp nên sẽ có nhiều trường hợp khó nên công cụ được chọn cần hoạt động ổn định và có cộng đồng hỗ trợ tốt.
- Báo cáo xúc tích dễ hiểu không qua sâu về kỹ thuật.

Với các tiêu chí trên sau khi so sánh các công cụ thì học viên thấy rằng công cụ **RobotFramework** là phù hợp nhất để triển khai kiểm thử chức năng tự động cho hệ thống vnEdu.

2.3.2. Chọn mô hình triển khai:

Với một sự án lớn như vnEdu mô hình triển khai kiểm thử tự động cần đáp ứng những yếu tố sau:

- **Tính duy trì:** Chúng ta cần một mô hình có thể tách biệt logic ra khỏi giao diện người dùng, giúp dễ dàng duy trì khi có sự thay đổi trong giao diện người

dùng. Khi giao diện thay đổi chúng ta chỉ cần sửa trên một nơi duy nhất thay vì phải sửa từng kịch bản.

- **Tính tái sử dụng:** Chúng ta cần một mô hình có thể tái sử dụng các phần tử và hành động trên các hệ thống khác nhau mà không cần viết lại. Điều này giúp tiết kiệm thời gian và công sức khi phát triển và duy trì các kịch bản.
- **Tính tách biệt:** Mô hình được sử dụng phải tách biệt rõ ràng giữa logic của kiểm thử và các chi tiết cụ thể của giao diện người dùng. Điều này giúp code trở nên dễ hiểu và dễ bảo trì hơn.
- **Tính độc lập:** Mô hình được chọn cần có tính độc lập cao không quá phụ thuộc và cấu trúc của giao diện người dùng. Điều này giúp dễ dàng chuyển đổi giữa các công nghệ giao diện người dùng khác nhau mà không làm ảnh hưởng quá nhiều đến kịch bản

Với những đặc điểm trên, học viên nhận thấy mô hình Page Object Model (POM) là thích hợp nhất, Mô hình POM đáp ứng đủ các đặc tính kể trên và là một mô hình được sử dụng rộng rãi cho việc áp dụng kiểm thử chức năng tự động hiện nay.

2.3.3. Phân tích thuận lợi, khó khăn và chọn ra những chức năng có thể kiểm thử tự động và những chức năng không kiểm thử tự động:

- **Thuận lợi:**

Hệ thống vnEdu là một hệ thống lớn, khối lượng chức năng đa dạng từ đơn giản đến phức tạp phù hợp để triển khai kiểm thử tự động chức năng từ đầu và phát triển lên dần theo thời gian

Có nhân lực để triển khai kiểm thử tự động: Mặc dù hạn chế về số lượng nhân sự nhưng nguồn lực để triển khai kiểm thử chức năng tự động của vnEdu là có. Các nhân sự kiểm thử thủ công sẵn sàng học và triển khai kiểm thử tự động

RobotFramework là một công cụ kiểm thử tự động được viết bằng ngôn ngữ lập trình Python, Đi theo định hướng Keyword Driven.

Với định hướng Keyword Driven các hàm tương tác với phần tử web sẽ được đóng gói lại thành các keyword khi sử dụng người dùng sẽ gọi hàm thông qua các

keyword và tham số. Điều đó thuận tiện hơn cho những kiểm thử viên triển khai kiểm thử tự động vì họ có hạn chế về kỹ năng lập trình

Robot Framework là phần mềm mã nguồn mở với cộng đồng sử dụng đông đảo trên toàn thế giới nên thích hợp triển khai thử nghiệm kiểm thử chức năng tự động cho vnEdu vì sẽ giảm được chi phí đầu tư ban đầu và tận dụng được sự hỗ trợ của cộng đồng sử dụng đông đảo.

Robot Framework được viết bằng ngôn ngữ lập trình Python một ngôn ngữ lập trình không quá phức tạp và dễ dàng tiếp cận cho những kiểm thử viên không có thể mạnh về lập trình.

Tuy nhiên không như các phần mềm có phí như Katalon khi tạo dự án kiểm thử tự động công cụ sẽ giúp người dùng tạo ra một template dự án theo mô hình hợp lý được cấu hình sẵn. Robot Framework là một phần mềm mã nguồn mở miễn phí độ tùy biến cao khi tạo dự án sẽ chưa có mô hình hay cấu trúc tổ chức các thành phần trong dự án. Vì vậy cần phải tự xây dựng tổ chức mô hình cấu trúc dự án và kịch bản kiểm thử hợp lý để dễ phát triển mở rộng, chuyển giao và bảo trì.

- **Khó khăn:**

- Các phần tử trên giao diện của hệ thống không có ID cố định, mỗi lần truy cập hệ thống lại sinh ra các id khác nhau cho từng phần tử
- Hệ thống có khối lượng chức năng phức tạp các kịch bản muốn chạy hiệu quả cần có độ ổn định cao.
- Nhân sự kiểm thử có kỹ năng lập trình hạn chế.
- Khối lượng server app và server database lớn dẫn đến phải chạy kiểm thử chức năng trên nhiều site nhiều môi trường
- RobotFramework với các tính năng có sẵn chưa đáp ứng được các yêu cầu kiểm thử tự động (như chức năng báo cáo, gửi tin)

Từ những thuận lợi và khó khăn kể trên học viên nhận thấy để áp dụng được kiểm thử tự động vào kiểm thử cho hệ thống vnEdu trong thực tế cần phải phân loại cụ thể những tính năng nào có thể áp dụng kiểm thử tự động, tính năng nào không áp dụng. cũng như xây dựng một bộ template (biểu mẫu) chung theo mô hình POM để triển

khai kiểm thử tự động cho hệ thống vnEdu

2.3.4. Phân loại những tính năng có thể kiểm thử tự động và những tính năng không áp dụng kiểm thử tự động:

Qua thời gian thu thập và phân tích số liệu những lỗi thường xảy ra học viên thấy lỗi chủ yếu tập trung ở ứng dụng Web của vnEdu, các lỗi thường xảy ra ở các chức năng cơ bản liên quan đến nhân sự, lớp, học sinh gây hậu quả nghiêm trọng và lỗi thường sinh ra do quá trình kiểm thử hồi quy thủ công gặp sai sót.

Vì thế học viên đề xuất tập trung xây dựng kiểm thử tự động cho các chức năng chính hay xảy ra lỗi của hệ thống vnEdu.

Nhưng trong hệ thống vnEdu khi sử dụng RobotFramework cho mục đích kiểm thử chức năng tự động học viên thấy có một số tính năng phù hợp để làm kiểm thử chức năng tự động một số chức năng rất khó để có thể triển khai kiểm thử chức năng tự động. Cụ thể như sau:

Nhóm tính năng có thể kiểm thử tự động: Các tính năng có thể kiểm thử tự động là nhóm tính năng cơ bản của hệ thống. Các nhóm tính năng này có những đặc tính phù hợp để triển khai kiểm thử tự động như:

- Là nhóm tính năng quan trọng của hệ thống, có độ quan trọng cao nếu gặp lỗi sẽ gây tổn thất nghiêm trọng đến uy tín của đơn vị.
- Các tính năng đã được xây dựng từ lâu có độ ổn định cao ít thay đổi về mặt giao diện cũng như nghiệp vụ: Những tính năng này phù hợp để kiểm thử tự động vì khi ít thay đổi kiểm thử viên sẽ thường chủ quan hoặc bỏ qua nghĩ rằng sẽ không có lỗi xảy ra. Điều đó tiềm ẩn rủi ro rất lớn về mặt chất lượng. Kiểm thử tự động sẽ đảm bảo tính năng đó vẫn hoạt động tốt sau mỗi lần release.
- Là nhóm tính năng cốt lõi dễ bị ảnh hưởng khi sửa lỗi hay thay đổi tính năng khác. Các tính năng cốt lõi rất dễ bị ảnh hưởng khi sửa đổi các tính năng khác, kiểm thử viên thường thiếu sót không lường hết được độ ảnh hưởng khi nhóm phát triển cập nhật, sửa đổi code. Nên khi áp dụng kiểm thử tự động để kiểm tra lại các tính năng cốt lõi mỗi lần release mang lại hiệu quả to lớn.
- Là các nhóm tính năng phải chạy lặp đi lặp lại trong các lần kiểm thử hồi

quy. Với các tính năng phải chạy lặp đi lặp lại với mỗi lần kiểm thử hồi quy sẽ tốn rất nhiều nguồn lực. Khi áp dụng kiểm thử tự động sẽ giảm bớt được nguồn lực khi chạy kiểm thử hồi quy một cách đáng kể.

Với các tiêu chí kể trên học viên đề xuất nhóm tính năng có thể kiểm thử tự động bao gồm:

- **Đăng nhập:** Phù hợp để xây dựng kịch bản kiểm thử tự động vì có nhiều trường hợp như: Đăng nhập không nhập dữ liệu, Đăng nhập không nhập trên đăng nhập, Đăng nhập không nhập mật khẩu

- **Quản lý nhân sự:** Màn hình thông tin nhân sự có nhiều field nhập gồm cả danh sách, field nhập nội dung, upload ảnh thích hợp để xây dựng kịch bản kiểm thử tự động bao gồm cả chức năng thêm, sửa, xóa nhân sự.

- **Quản lý lớp:** Màn hình thông tin lớp có nhiều field nhập gồm cả danh sách, field nhập nội dung, upload ảnh thích hợp để xây dựng kịch bản kiểm thử tự động bao gồm cả chức năng thêm, sửa, xóa.

- **Quản lý học sinh:** Màn hình thông tin nhân sự có nhiều field nhập gồm cả danh sách, field nhập nội dung, upload ảnh thích hợp để xây dựng kịch bản kiểm thử tự động bao gồm cả chức năng thêm, sửa, xóa học sinh.

- **Nhóm chức năng kết quả học tập:** Màn hình nhập kết quả học tập với dạng nhập luôn trên bảng và tính toán tổng kết dựa vào dữ liệu đầu vào người dùng nhập và công thức tính dựa theo các thông tư nên rất thích hợp để xây dựng kịch bản kiểm thử tự động cho chức năng kết quả học tập.

Nhóm tính năng không áp dụng kiểm thử tự động: Là các tính năng không phù hợp để triển khai kiểm thử chức năng tự động trong giai đoạn đầu. Những tính năng không phù hợp để áp dụng kiểm thử tự động trong giai đoạn này có đặc điểm như sau:

- Những tính năng ngăn chặn xâm nhập hay tự động hóa: Những tính năng ngăn chặn xâm nhập hay tự động hóa được phát triển để không cho người sử dụng dùng script để thực hiện thao tác. Những chức năng này không phù hợp để kiểm thử tự động.

- Những tính năng yêu cầu xác nhận của một thiết bị bên ngoài Một số tính năng yêu cầu xác nhận từ một hoặc nhiều thiết bị bên ngoài hoặc phần mềm bên ngoài, trong giai đoạn đầu triển khai với nguồn lực và trình độ còn hạn chế sẽ rất khó trong việc liên kết và lấy dữ liệu từ thiết bị hay phần mềm bên ngoài. Nên trong giai đoạn này với các tính năng có liên kết với thiết bị hoặc phần mềm bên ngoài học viên đề xuất sẽ không áp dụng kiểm thử tự động.

- Những tính năng liên quan đến số liệu được xuất ra trên file hoặc một phần mềm offline: Vì đề án tập trung vào thực hiện kiểm thử tự động trên Web nên với những tính năng liên quan đến xử lý số liệu trên file offline và các phần mềm offline không thích hợp để kiểm thử tự động trong giai đoạn này.

Dựa vào các đặc điểm trên học viên đề xuất những tính năng không áp dụng kiểm thử tự động trong giai đoạn này bao gồm:

- **Tính năng Captcha:** Tính năng Captcha được thiết kế để phát hiện những truy cập không phải người thật (robot, hoặc các đoạn mã tự động thực thi). Chức năng được thiết kế để ngăn cản tự động hóa nên những chức năng có Capcha học viên đề xuất sẽ không áp dụng kiểm thử tự động

- **Tính năng gửi tin nhắn:** Gửi tin nhắn tin nhắn sẽ được gửi đến số điện thoại đã đăng ký. Rất khó để thực hiện kiểm thử tự động để kiểm tra được tin nhắn đã được gửi đến điện thoại thành công hay kiểm tra nội dung tin nhắn tại điện thoại di động. Nên tính năng gửi tin nhắn học viên cũng đề xuất không áp dụng kiểm thử tự động.

- **Tính năng ký số:** Tính năng ký số với ký số qua token Usb hệ thống yêu cầu xác nhận trên một popup window khi cắm usb vào máy. Với cửa sổ window không thể lấy được xpath. Vì vậy học viên đề xuất trọng nội dung đề án sẽ không áp dụng kiểm thử chức năng cho chức năng ký số điện tử.

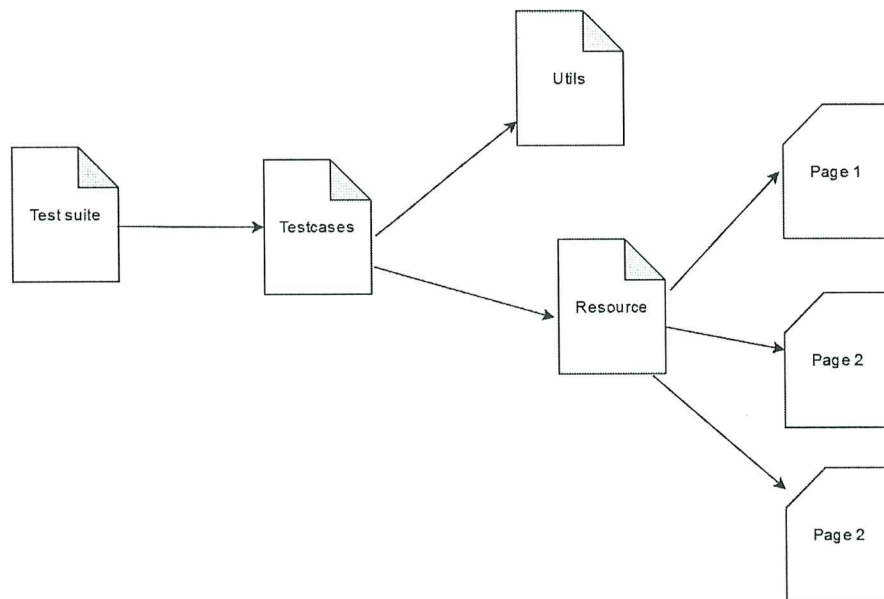
- **Tính năng thông kê báo cáo:** Tính năng báo cáo với báo cáo dữ liệu cần kiểm tra là rất lớn và cần độ chính xác cao, có thể xây dựng kịch bản kiểm thử từng giá trị trong báo cáo nhưng việc đó sẽ tốn rất nhiều công sức và nguồn lực. Do đó trong đề án học viên đề xuất không áp dụng kiểm thử chức năng tự

động cho chức năng báo cáo.

2.3.5. Nghiên cứu triển khai bộ template kiểm thử tự động:

Với mô hình POM được đề xuất ở phần 1.3.3 để triển khai hiệu quả và thống nhất trong việc quản lý cần cụ thể hóa mô hình thành một bộ template (biểu mẫu) bao gồm các thành phần, cấu trúc của dự án kiểm thử tự động. Quy định rõ chức năng các thành phần sẽ làm nhiệm vụ gì, giao tiếp với nhau như thế nào. Để tránh việc các kiểm thử viên viết code một cách tự do, không nhất quán dẫn đến khó khăn trong việc ghép các code thành phần lại với nhau, hoặc khó quản lý, duy trì và nâng cấp. Ngoài ra việc xây dựng bộ mẫu chung sẽ giúp thuận lợi hơn khi áp dụng cho các hệ thống khác trong tương lai.

Cách tổ chức đề xuất



Hình 2.1. Mô hình các thành phần trong dự án kiểm thử tự động được đề xuất

Trong đó gồm:

- **Testsuite:** Là một file có định dạng.robot và khi chạy kịch bản kiểm thử ta sẽ chạy file này. Lưu tên các kịch bản kiểm thử (Testcases) và gọi đến file Testcase.robot

- **Testcases:** Là một file có định dạng.robot chứa những Custom keywords để thực hiện các thao tác của chức năng. Mỗi keyword trong file Testcases.robot sẽ tượng trưng cho một Testcases trong đó chứa các Custom Keyword được gọi ra từ file Ultils hoặc Resource

- **Ultils** là một file có định dạng.robot là nơi định nghĩa các keywords và các biến dùng chung cho tất cả các kịch bản kiểm thử tự động như: Mở trang, Đăng nhập, xóa dữ liệu

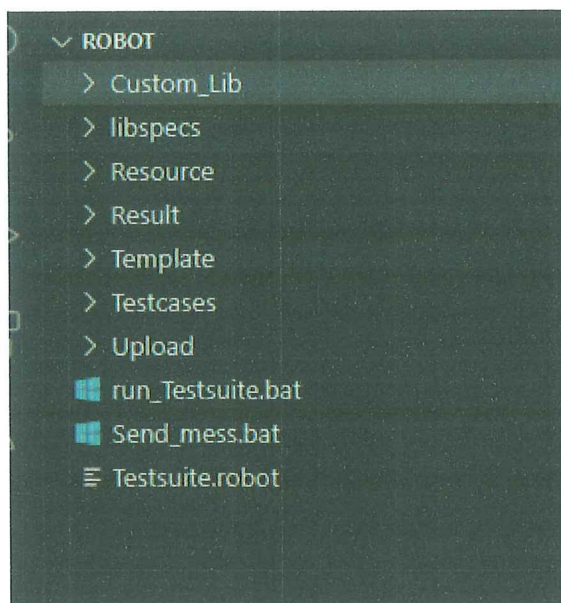
- **Resource:** Là tập hợp các file robot chứa các keyword và biến thực hiện cụ thể một chức năng nào đó trên màn hình, mỗi màn hình là một file.robot trong Resource chứa các keywords được sử dụng cho màn hình hoặc đối tượng nghiệp vụ đó.

Trên đây là đề xuất tổ chức kịch bản kiểm thử tự động theo mô hình POM của học viên qua đó khi nếu có lỗi do sửa UI chúng ta có thể không mất công tìm kiếm mà không biết cần phải sửa gì và sửa ở đâu. Với cách tổ chức này khi màn hình nào thay đổi ta chỉ cần tìm đến chính xác testcases và resource của màn hình đó. Tiết kiệm nhiều thời gian cho việc bảo trì.

Đối với mở rộng cũng dễ dàng hơn, chúng ta chỉ cần thêm một file đại diện cho màn hình đó trong resource chứa các keyword và biến thao tác với màn hình đó. Sau đó triển khai mã kịch bản kiểm thử tự động ở File Testcases.robot. Việc cuối cùng là thêm keyword với tên của testcases đã tạo. Thuận tiện cho việc mở rộng.

Template kiểm thử tự động đề xuất:

Với mô hình học viên đề xuất ở trên mô hình tổ chức kịch bản kiểm thử tự động sẽ gồm các thành phần chính như hình



Hình 2.2. Mô hình tổ chức dự án kiểm thử tự động cho vnEdu

Trong đó các gồm các thành phần

- Custom_Lib: Chứa các thư viện python được học viên viết thêm
- Libspecs: Các thư viện được cài thêm vào dự án
- Resource: Nơi chứa các file Resource được chia theo từng màn hình chức

năng

- Result: Nơi lưu kết quả sau mỗi lần chạy
- Template: Nơi lưu các biểu mẫu
- Testcases: Nơi lưu các file kịch bản kiểm thử
- Upload: Nơi lưu trữ các file để kịch bản upload lên hệ thống với các kịch bản

upload file

- Run_Testsuite.bat: File bat chạy Testsuite
- Send_message.bat: Là file bat dùng để gửi tin nhắn qua tele
- Testsuite.robot: Là danh sách các kịch bản sẽ chạy

2.3.6. Xây dựng các thư viện, các hàm, các keyword và báo cáo hỗ trợ cho việc kiểm thử chức năng tự động:

Trong quá trình phát triển và triển khai kiểm thử tự động việc tương tác với file excel và gửi thông báo qua các nền tảng như Telegram đang trở nên ngày càng phổ

biến. Việt hệ thống có khả năng gửi thông điệp tự động qua Telegram mang lại nhiều lợi ích đáng kể nhưng các keyword sẵn có của robot Framework chưa hỗ trợ hai tính năng này nên cần xây dựng thư viện riêng cho 2 tính năng này:

- Thư viện đọc ghi Excel:

Một số chức năng cần đọc ghi file excel các thư viện có sẵn không phù hợp nên học viên đã viết một thư viện python để đọc ghi file excel.

Thư viện gồm 3 keyword:

+ **def make_excel_file(file_name):** Keyword này được sử dụng để tạo một file excel với tên được chỉ định có thể sử dụng khi tạo file report với tên là biến truyền vào theo ngày tháng hoặc phiên bản

+ **def write_excel_value(data, sheetname, cell, result):** Để ghi vào file Excel với tham số là dữ liệu data, sheetname là tên sheet, cell là vị trí cần ghi và result là một tham số dữ liệu, Keyword được sử dụng khi muốn truyền một dữ liệu vào chính xác vị trí nào đó trong một file excel được chỉ định

+ **def read_excel_data_into_dict(filename, sheetname):** Để đọc dữ liệu từ file Excel ra với tham số là tên file excel và tên sheet. Keyword được dùng để đọc dữ liệu từ file excel và trả lại nội dung với kiểu dữ liệu từ điển dict (dictionary)

- Thư viện gửi message qua Telegram:

Để gửi tin nhắn qua Telegram (phục vụ cho mục đích gửi báo cáo, gửi file...) ta cần viết thư viện thực hiện việc gửi tin nhắn qua telegram

Thư viện được xây dựng gồm 2 keywords:

+**def send_file_to_bot(self, bot_token, chat_id, file_path message=None):** Keyword này dùng để gửi file qua một nhóm chat của telegram với các tham số bot token, group chat id, file path để chỉ ra đường dẫn địa chỉ của file cần gửi, message: tin nhắn gửi cùng với file.

+**def send_message_to_bot(self, bot_token, chat_id, message):** Keyword này dùng để gửi tin nhắn qua telegram bao gồm các tham số bot_token để chỉ ra bot gửi tin là bot nào, chat_id để xác định phòng chat gửi đến và, message là tin nhắn cần gửi.

Ngoài 2 thư viện nêu trên học viên có xây dựng một số keyword dùng chung có thể áp dụng cho nhiều dự án được đặt trong file Utils bao gồm:

CreateNewFile: Keyword dùng để sinh ra một file ở đây học viên dùng cho việc sinh file kết quả excel bằng cách copy một file mẫu có sẵn và lưu vào thư mục result.

GhiDulieu: Keyword dùng để ghi dữ liệu vào file excel với tham số là tên file, tên sheet, vị trí cần ghi dữ liệu và dữ liệu cần ghi

Generate name: Keyword dùng để sinh tự động các tên ngẫu nhiên tránh cho việc trùng lặp tên dẫn đến kịch bản kiểm thử không chạy được. Keyword trả về một biến có tên là giá trị truyền vào nối với một chuỗi String.

Upload File: Keyword dùng chung cho các chức năng upload file lấy file có tên nhất định từ đường dẫn trên máy tính đang chạy upload lên field có type= file

Readexcel: Hàm gọi tới thư viện đọc ghi excel học viên đã nói ở trên và sử dụng keyword “read_excel_data_into_dict” được lập trình trong thư viện và giá trị trả về là một biến data gồm các giá trị đọc được trong file excel

```

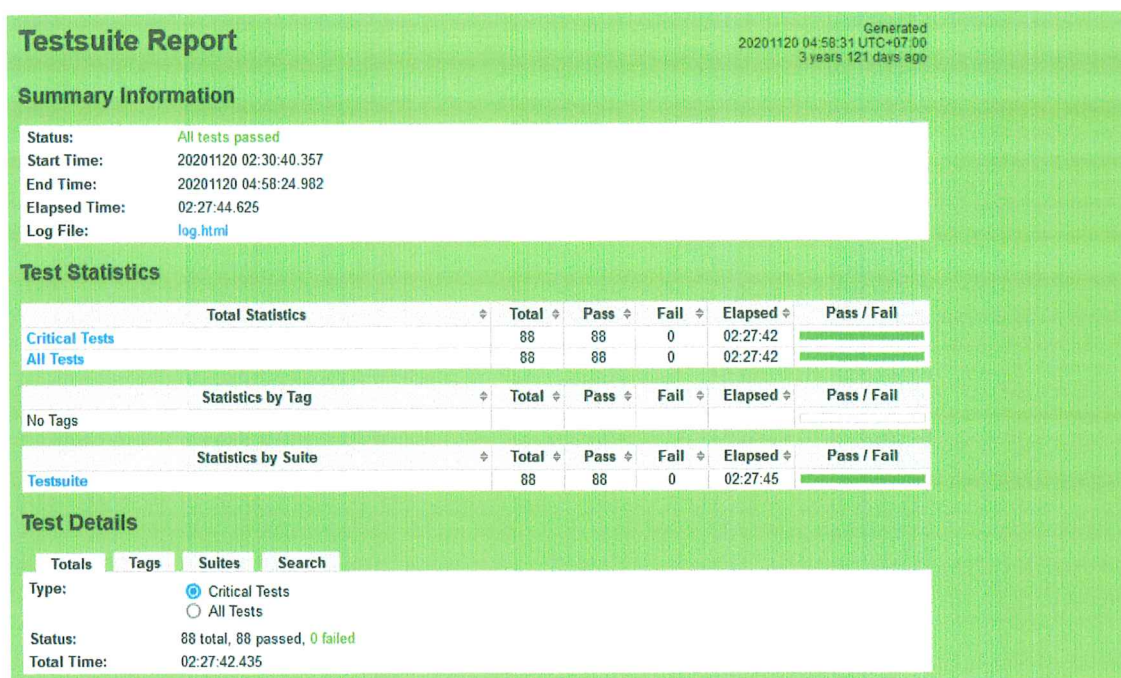
Readexcel
[Arguments]    ${FilePath}    ${filename}
${data} =      read_excel_data_into_dict    ${FilePath}    ${filename}
[Return]       ${data}

```

Hình 2.3. Keyword Readexcel

2.3.7. Đánh giá và xây dựng mô hình báo cáo:

Trong RobotFramework có cung cấp dạng báo cáo HTML lưu trên máy chạy kịch bản, nhưng hơi bất cập vì muốn xem phải remote vào máy, báo cáo lại bằng tiếng Anh gây khó khăn cho người sử dụng.



Hình 2.4. Báo cáo HTM của Robot Framework

Học viên đề xuất sinh ra một báo cáo rút gọn bằng file excel với các thông tin chính như hệ thống (chức năng), tổng số lượng kịch bản đã chạy, danh sách kịch bản đã chạy cùng thời gian chạy mỗi kịch bản, số lượng kịch bản chạy thành công (Pass) số lượng kịch bản chạy thất bại (Failed).

Xây dựng keyword để đếm số lượng kịch bản Pass và Failed. Mỗi lần chạy testcases sẽ dùng keyword **“Run Keyword And Return Status”** keyword sẽ trả về biến **checkstt** là True nếu testcases pass và false nếu testcases Failed. Dựa vào biến **checkstt** ta xây dựng các keyword để tăng số lượng kịch bản dựa theo trạng thái

Nếu biến **checkstt** trả về giá trị **True** chúng ta sẽ tiến hành gọi keyword **Cong_mot_don_vi_pass_Sub_feature** và **Cong_mot_don_vi_pass** để tính thêm một đơn vị vào số lượng kịch bản Pass

Nếu biến **checkstt** trả về giá trị **False** chúng ta sẽ tiến hành gọi keyword **Cong_mot_don_vi_failed_Sub_feature** và **Cong_mot_don_vi_failed** để tính thêm một đơn vị vào số lượng kịch bản Failed.

```

Cong_mot_donvi_pass
[Arguments]    ${Pass_tcs}
${Pass_tcs}    Evaluate    ${Pass_tcs}+1
Set Global Variable    ${Pass_tcs}

Cong_mot_donvi_faield
[Arguments]    ${Failed_tcs}
${Failed_tcs}    Evaluate    ${Failed_tcs}+1
Set Global Variable    ${Failed_tcs}

```

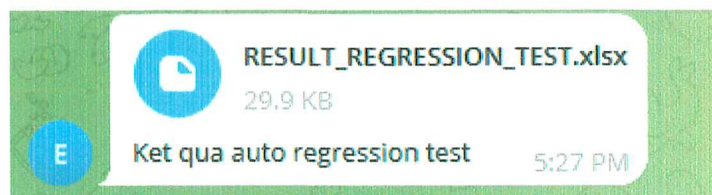
Hình 2.5. Các keyword để tính kịch bản Pass/Failed

File kết quả sau khi ghi

Kết quả chạy testcases				Kết quả	
STT	Tên Testcases	Ngày chạy	Trạng thái	Số trường hợp testcases passes	99
1	TC-1 QLY DANGNHAP Dang nhap bo trong ten dang nhap va mat khau	19-03-24 11:20:55	PASSED	Số trường hợp testcases failed	1
2	TC-2 QLY DANGNHAP Dang nhap bo trong ten dang nhap va nhap mat kh	19-03-24 11:20:59	PASSED	Tổng số trường testcases đã chạy	100
3	TC-3 QLY DANGNHAP Dang nhap nhap ten dang nhap va bo trong mat kh	19-03-24 11:21:03	PASSED		
4	TC-4 QLY DANGNHAP Dang nhap voi nguoi dung sai mat khau	19-03-24 11:21:11	PASSED		
5	TC-5 QLY DANGNHAP Dang nhap voi ten dang nhap khong ton tai	19-03-24 11:21:16	PASSED		
6	TC-6 QLY DANGNHAP Dang nhap thanh cong	19-03-24 11:21:23	PASSED		
7	TC-7 QLY DANGNHAP Vao trang quan ly nha truong	19-03-24 11:22:05	PASSED		
8	TC-8 QLY DANGNHAP Kiem tra trang quan ly nha truong	19-03-24 11:22:58	PASSED		

Hình 2.6. Báo cáo Excel được học viên xây dựng

Để thuận tiện hơn cho người sử dụng không phải remote vào máy để đọc báo cáo này học viên đã đề xuất sử dụng thư viện dùng bot gửi file và tin nhắn qua telegram được học viên xây dựng ở phần trên để có thể gửi file báo cáo qua group chat Telegram.



Hình 2.7. Gửi báo cáo qua Telegram

KẾT LUẬN CHƯƠNG

Từ các kết quả của chương 2, học viên đã đề xuất xây dựng mô hình template kiểm thử tự động để có thể triển khai trên diện rộng cho các dự án. Và sử dụng template đó để áp dụng kiểm thử tự động chức năng cho hệ sinh thái vnEdu.

CHƯƠNG 3. THỰC NGHIỆM VÀ ĐÁNH GIÁ

Chương này sẽ trình bày cách triển thực nghiệm và đánh giá của giải pháp đề xuất sau một thời gian triển khai thực tế.

3.1. Triển khai kiểm thử tự động cho hệ sinh thái vnEdu

Sau khi đã xây dựng xong tempale đề xuất cho kiểm thử chức năng tự động. Việc tiếp theo là sẽ áp dụng theo template mẫu để xây dựng kịch bản kiểm thử tự động cho hệ thống vnEdu.

Việc xây dựng kịch bản kiểm thử chức năng tự động cho hệ thống vnEdu gồm các bước: Xác định và viết các kịch bản kiểm thử, xây dựng script kiểm thử tự động, đề xuất áp dụng những mở rộng


3.1.1. Xây dựng kịch bản thủ công các tính năng có thể viết kịch bản kiểm thử tự động

Như tại chương 2 phần 2.1 học viên đã đề xuất xây dựng kịch bản kiểm thử tự động gồm các chức năng liên quan tới Đăng nhập, Quản lý nhân sự, Quản lý lớp, Quản lý học sinh

Dưới đây là một số kịch bản thủ công học viên đã xây dựng để kiểm thử một phần các chức năng được nhắc đến ở trên.

Tính đúng đắn của bộ testcases:

- Testcases được xây dựng dựa trên các phương pháp như Phân vùng tương đương, Phân tích giá trị biên theo chuẩn chứng chỉ ISTQB được công nhận trên thế giới.
- Testcases tuân theo tiêu chuẩn IEEE 829-2008
- Testcases đã được kiểm định và xác nhận đạt tiêu chuẩn của trung tâm Kiểm định thuộc công ty Công nghệ thông tin VNPT (Đã được cấp phép đăng ký hoạt động thử nghiệm)

 TEST CASE (Trường hợp kiểm tra)			
CÔNG TY CÔNG NGHỆ THÔNG TIN VNPT			
Tên dự án:	Hệ thống quản lý thông tin	Người tạo:	Lương Quang Cường, Nguyễn Quỳnh Anh, Bùi Thị
Mã dự án:	VNPT_eEDU_v1.0.0	Người kiểm tra:	Lương Quang Cường
Mã tài liệu:	TC_vnEdu	Ngày kiểm tra:	

Hình 3.1. Tiêu đề các trường hợp kiểm tra

Bảng 3.1. Mẫu kịch bản kiểm thử

Mã kịch bản	Tên kịch bản	Các bước thực hiện	Kết quả mong đợi
TC01	Đăng nhập bỏ trống tên đăng nhập và mật khẩu	1. Truy cập vào hệ thống 2. Bỏ trống tên đăng nhập và mật khẩu 3. Click nút “Đăng nhập”	Hệ thống hiển thị thông báo “Bạn chưa nhập tài khoản”
TC02	Đăng nhập thành công	1. Truy cập vào hệ thống 2. Nhập chính xác tên đăng nhập và mật khẩu 3. Click nút “Đăng nhập”	Đăng nhập thành công
TC03	Vào trang quản lý nhân sự thành công	1. Đăng nhập vào hệ thống 2. Vào quản lý nhà trường 3. Vào quản lý nhân sự	Hệ thống đăng nhập thành công
TC04	Thêm nhân sự thành công	1. Đăng nhập vào hệ thống 2. Vào quản lý nhà trường 3. Vào quản lý nhân sự 4. Click thêm mới > Nhập liệu > Click Lưu	Hệ thống lưu nhân sự thành công nhân sự mới hiển thị thành công trong danh sách
TC05	Sửa nhân sự thành công	1. Đăng nhập vào hệ thống 2. Vào quản lý nhà trường 3. Vào quản lý nhân sự 4. Chọn một nhân sự > Sửa thông tin > Click Lưu	Hệ thống hiển thị thông tin nhân sự chính xác tại bước 4 Sau khi lưu thông tin nhân sự sau khi sửa hiển thị chính xác
TC06	Xóa nhân sự thành công	1. Đăng nhập vào hệ thống 2. Vào quản lý nhà trường 3. Vào quản lý nhân sự 4. Chọn một nhân sự > Click Xóa > Click Yes trong confirm popup	Hệ thống xóa nhân sự thành công. Nhân sự bị xóa sẽ không hiển thị trong danh sách
TC07	Vào trang quản	1. Đăng nhập vào hệ thống	Trang danh sách

Mã kịch bản	Tên kịch bản	Các bước thực hiện	Kết quả mong đợi
	lý lớp thành công	2. Vào quản lý nhà trường 3. Vào Quản lý lớp	lớp hiển thị chính xác
TC08	Thêm lớp thành công	1. Đăng nhập vào hệ thống 2. Vào quản lý nhà trường 3. Vào quản lý lớp 4. Click thêm mới > Nhập liệu > Click Lưu	Hệ thống lưu lớp thành công lớp mới hiển thị thành công trong danh sách
TC09	Sửa lớp thành công	1. Đăng nhập vào hệ thống 2. Vào quản lý nhà trường 3. Vào quản lý lớp 4. Chọn một lớp > Click Sửa > Sửa thông tin lớp > Click Lưu	Hệ thống hiển thị thông tin lớp chính xác Sau khi sửa dữ liệu hệ thống lưu dữ liệu thành công. Dữ liệu sau khi sửa sẽ hiển thị thành công
TC10	Xóa lớp thành công	1. Đăng nhập vào hệ thống 2. Vào quản lý nhà trường 3. Vào quản lý lớp 4. Chọn một lớp > Click Xóa > Click Yes trong confirm popup	Hệ thống xóa lớp thành công. Nhân sự bị xóa sẽ không hiển thị trong danh sách
TC11	Hệ thống hiển thị danh sách học sinh thành công	1. Đăng nhập vào hệ thống 2. Vào quản lý nhà trường 3. Vào quản lý lớp > Chọn một lớp > Click biểu tượng danh sách học sinh	Hệ thống hiển thị danh sách học sinh thành công
TC12	Thêm học sinh thành công	1. Đăng nhập vào hệ thống 2. Vào quản lý nhà trường 3. Vào quản lý lớp > Chọn một lớp > Click biểu tượng	Hệ thống lưu học sinh thành công học sinh mới hiển thị thành công trong

Mã kịch bản	Tên kịch bản	Các bước thực hiện	Kết quả mong đợi
		danh sách học sinh 4. Click thêm mới > Nhập liệu > Click Lưu	danh sách
TC13	Sửa học sinh thành công	1. Đăng nhập vào hệ thống 2. Vào quản lý nhà trường 3. Vào quản lý lớp > Chọn một lớp > Click biểu tượng danh sách học sinh 4. Chọn một học sinh > Click Sửa > Sửa thông tin > Click Lưu	Hệ thống hiển thị thông tin học sinh chính xác Sau khi sửa dữ liệu hệ thống lưu dữ liệu thành công. Dữ liệu sau khi sửa sẽ hiển thị thành công
TC14	Xóa học sinh thành công	1. Đăng nhập vào hệ thống 2. Vào quản lý nhà trường 3. Vào quản lý lớp > Chọn một lớp > Click biểu tượng danh sách học sinh 4. Chọn một học sinh > Click Xóa > Click Lưu trong confirm popup	Hệ thống xóa học sinh thành công. Học sinh bị xóa sẽ không hiển thị trong danh sách

3.1.2. Xây dựng kịch bản kiểm thử chức năng tự động theo các kịch bản thủ công đã xây dựng

Với mô hình đã trình bày ở phần này học viên sẽ trình bày hướng xây dựng các kịch bản kiểm thử chức năng tự động từ kịch bản kiểm thử chức năng thủ công đã được liệt kê ở trên.

Bước 1: Xây dựng file resource gồm những keyword cần thiết của một màn hình

```

close_popup_nhansu [Arguments] ${tenpopup}
    Maximize Browser Window
    Wait Until Element Is Visible //span[contains(text(), '${tenpopup}')]//...//img[@class='x-tool-close']/...
    Click Element //span[contains(text(), '${tenpopup}')]//...//img[@class='x-tool-close']/...
    Wait Until Element Is Not Visible //span[contains(text(), '${tenpopup}')]//...//img[@class='x-tool-close']/... 10s
    Element Should Not Be Visible //span[contains(text(), '${tenpopup}')]//...//img[@class='x-tool-close']/...
    Set Window Size 1036 754

nhapthongtin_giao_vien
    [Arguments] ${hoten} ${Ngaysinh}
    Wait Until Element Is Visible //label[text()='Họ và tên:']/div/input[@name='ho_ten'] 10s
    Click Element //label[text()='Họ và tên:']/div/input[@name='ho_ten']
    Input Text //label[text()='Họ và tên:']/div/input[@name='ho_ten'] ${hoten}
    Click Element //label[text()='Ngày sinh:']/div/input[@name='ngay_sinh_text']
    Input Text //label[text()='Ngày sinh:']/div/input[@name='ngay_sinh_text'] ${Ngaysinh}
    Click Element //label[text()='Điện thoại DD:']/div/input
    Input Text //label[text()='Điện thoại DD:']/div/input 0949830087

Luu_giao_vien
    [Arguments] ${thanhcong} ${Yes}
    Click Element //span[text()='Lưu']
    Run Keyword If ${thanhcong}==0 Wait Until Element Is Visible //span[text()='Thông tin không hợp lệ!']
    Run Keyword If ${thanhcong}==0 Click Element //span[text()='OK']
    Run Keyword If ${thanhcong}==0 Element Should Not Be Visible //span[text()='Thông tin không hợp lệ!']
    Run Keyword If ${thanhcong}==1 Element Should Be Visible //span[text()='Bạn có chắc chắn lưu thông tin?']
    Run Keyword If ${thanhcong}==1 and ${Yes}==0 Click Element //span[text()='No']/...
    Run Keyword If ${thanhcong}==1 and ${Yes}==0 Wait Until Element Is Not Visible //span[text()='Bạn có chắc chắn lưu thông tin?']
    Run Keyword If ${thanhcong}==1 and ${Yes}==1 Click Element //span[text()='Yes']/...
    Run Keyword If ${thanhcong}==1 and ${Yes}==1 Wait Until Element Is Not Visible //span[text()='Bạn có chắc chắn lưu thông tin?']
    Run Keyword If ${thanhcong}==1 and ${Yes}==1 Wait Until Element Is Visible //span[text()='Thành công!' or text()='Thành công!'] 30s
    Run Keyword If ${thanhcong}==1 and ${Yes}==1 Click Element //span[text()='OK']/...
    Run Keyword If ${thanhcong}==1 and ${Yes}==1 Wait Until Element Is Not Visible //span[text()='OK']/... 10s

```

Hình 3.2. File Resource (Nơi chứa các keyword cho từng màn hình)

Bước 2: Sử dụng các keyword đã được xây dựng để tạo ra các kịch bản kiểm thử trong file testcases

```

TC-04_QLY_NHANSU_Them_nhan_su_thanh_cong
[Tags]    Dangnhap
#dang_nhap_bo_trong_email_mat_khau
Open
Dangnhap_thanhcong    ${tentruong}
Vao_trang_quan_ly_nha_truong
sleep 2s
Vao_mot_shorcut    ${nhansugiaoven}
sleep 5s
Click Element //span[text()='Thêm mới']/...
${tennhansu}= Generate_name    Tennhansu_1
nhapthongtin_giao_vien    ${tennhansu}    22/12/2000
Luu_giao_vien    1    1
Tim_kiem_nhan_su    ${tennhansu}    1
Xoa_nhan_su    1    1
Dangxuat_quanlynhatruong
Close Browser

```

Hình 3.3. Một kịch bản kiểm thử trong file Testcases

Bước 3: Trong file Testsuite.robot sẽ gọi đến keyword testcases vừa xây dựng

```

60  ✓ TC-04_QLY_NHANSU_Them_nhan_su_thanh_cong
61      ${Tcs_name}    Set Variable    TC-11_QLY_NHANSU_Them_nhan_su_thanh_cong
62      ${checkstt}=    Run Keyword And Return Status    TC-11_QLY_NHANSU_Them_nhan_su_thanh_cong
63      GhiResult_Sheet    ${a}    ${b}    CORE    ${checkstt}    ${Tcs_name}
64
65  ✓ TC-05_QLY_NHANSU_Sua_nhan_su_thanh_cong
66      ${Tcs_name}    Set Variable    TC-11_QLY_NHANSU_Them_nhan_su_thanh_cong
67      ${checkstt}=    Run Keyword And Return Status    TC-11_QLY_NHANSU_Them_nhan_su_thanh_cong
68      ###GhiResult    CORE    ${checkstt}    ${Tcs_name}
69      GhiResult_Sheet    ${a}    ${b}    CORE    ${checkstt}    ${Tcs_name}
70

```

Hình 3.4. Cách gọi kịch bản trong file Testsuite

Các kịch bản còn lại cũng được xây dựng với phương pháp tương tự

Sau khi xây dựng xong kịch bản khi chạy hệ thống hiển thị kết quả như sau:

Nhập dấu trang...
 Hệ thống E-Club
 Nghiên cứu và ứng dụng...
 Run a WordCount exa...
 Nguyen Duong Hai - ...

SUITE TestSuite 1

Full Name: TestSuite 1
Source: D:\PMP\Caohoc\Luanvan\Robot\TestSuite_1.robot
Start / End / Elapsed: 20240322 23:40:25.174 / 20240322 23:49:01.196 / 00:08:36.022
Status: 16 tests total, 16 passed, 0 failed, 0 skipped

- + **TEST** TC0_SETUP
- + **TEST** TC-01_QLY_DANGNHAP_Dang_nhap_bo_trong_ten_dang_nhap_va_mat_khau
- + **TEST** TC-02_QLY_DANGNHAP_Dang_nhap_thanh_cong
- + **TEST** TC-03_QLY_NHATRUONG_Vao_trang_quan_ly_nhan_su
- + **TEST** TC-04_QLY_NHANSU_Them_nhan_su_thanh_cong
- + **TEST** TC-05_QLY_NHANSU_Sua_nhan_su_thanh_cong
- + **TEST** TC-06_QLY_NHANSU_Xoa_thanh_cong
- + **TEST** TC-07_QLY_LOP_Vao_trang_quan_ly_lop_hoc_thanh_cong
- + **TEST** TC-08_QLY_LOP_Them_lop_cap_3_thanh_cong
- + **TEST** TC-09_QLY_LOP_Sua_lop_thanh_cong
- + **TEST** TC-10_QLY_LOP_Xoa_lop_thanh_cong
- + **TEST** TC-11_QLY_HOCSINH_Check_vao_trang_quan_ly_hoc_sinh_thanh_cong
- + **TEST** TC-12_QLY_HOCSINH_Tao_hoc_cap_2_sinh_thanh_cong
- + **TEST** TC-13_QLY_HOCSINH_Sua_hoc_sinh_thanh_cong
- + **TEST** TC-14_QLY_HOCSINH_Xoa_hoc_sinh_thanh_cong

Hình 3.5. Báo cáo HTML sau khi chạy

3.1.3. Đề xuất áp dụng những mở rộng

- Áp dụng cơ chế tự động chạy lại kịch bản chạy thất bại:

Khi chạy kịch bản kiểm thử có thể do một nguyên nhân khách quan nào đó khiến một trong các kịch bản chạy thất bại (nguyên nhân khách quan có thể là do mạng, do phần tử chưa kịp hiển thị ...). Nếu đã tích hợp vào chuỗi devOps một kịch bản chạy không ổn định có thể làm thất bại cả chuỗi và phiên đó sẽ không được đưa lên môi trường thật gây hậu quả là chậm trễ việc upcode.

Để hạn chế việc này học viên đề xuất xây dựng cơ chế mỗi kịch bản thất bại sẽ được chạy lại 3 lần nếu cả ba lần đều là thất bại thì kết quả của kịch bản đó trong báo cáo mới được xác định là thất bại, nếu một trong 3 lần mà chạy thành công thì sẽ không chạy lần tiếp theo và được đánh dấu là thành công

Sử dụng keyword Rerun_Testcases và kiểm tra theo checkstt

```
Rerun_Testcases    [Arguments]    ${checkstt}    ${name_tcs}
    ${checkstt}=    Run Keyword And Return Status    ${name_tcs}
    Set Global Variable    ${checkstt}
```

Hình 3.6. Keyword để chạy lại testcases

Khi gọi keyword Testcases cho các kịch bản kiểm thử kiểm tra checkstt và gọi đến hàm Rerun_Testcase

```
TC-9_QLY_NHATRUONG_Vao_trang_quan_ly_nhan_su
    ${Tcs_name}    Set Variable    TC-9_QLY_NHATRUONG_Vao_trang_quan_ly_nhan_su
    ${checkstt}=    Run Keyword And Return Status    TC-9_QLY_NHATRUONG_Vao_trang_quan_ly_nhan_su
    Run Keyword If    '${checkstt}'== 'False'    Rerun_Testcases    ${checkstt}    ${Tcs_name}
    Run Keyword If    '${checkstt}'== 'False'    Rerun_Testcases    ${checkstt}    ${Tcs_name}
    ###GhiResult    CORE    ${checkstt}    ${Tcs_name}
    GhiResult_Sheet    ${a}    ${b}    CORE    ${checkstt}    ${Tcs_name}
```

Hình 3.7. Gọi đến keyword Rerun_Testcases dựa vào biến checkstt

- Áp dụng tạo báo cáo Excel và gửi báo cáo qua Telegram:

Như đã nói ở phần trên báo cáo của RobotFramework là dạng HTML muốn xem cần phải vào máy test nên học viên đề xuất dùng hàm đọc ghi excel để tạo ra report vào file excel và gửi qua telegram bằng thư viện gửi report telegram đã tạo ở phía trên

Kết quả báo cáo:

eEdu_DevOps_bot

Reply

Ngày 22-03-2024 chạy regression test trên server_s37:

Tổng số lượng Testcases đã chạy là : 14 testcases

Số lượng case PASSED là : 8 testcases

Số lượng testcases FAILED là : 6

Trong đó Hệ thống vnEdu-Core có :

Tổng số testcases là :14

Số testcases Pass là : 8

Số testcases Failed là:6

Hình 3.8. Tin nhắn thông báo kết quả kiểm thử chức năng



Hình 3.9. Tin nhắn gửi báo cáo excel trong nhóm chat Telegram

- Áp dụng vào quy trình devOps:

Quy trình DevOps là quy trình khép kín tự động hóa từ việc kiểm thử tĩnh, lưu giữ bản build, triển khai code, chạy kiểm thử chức năng tự động và cuối cùng là lưu giữ phiên bản release

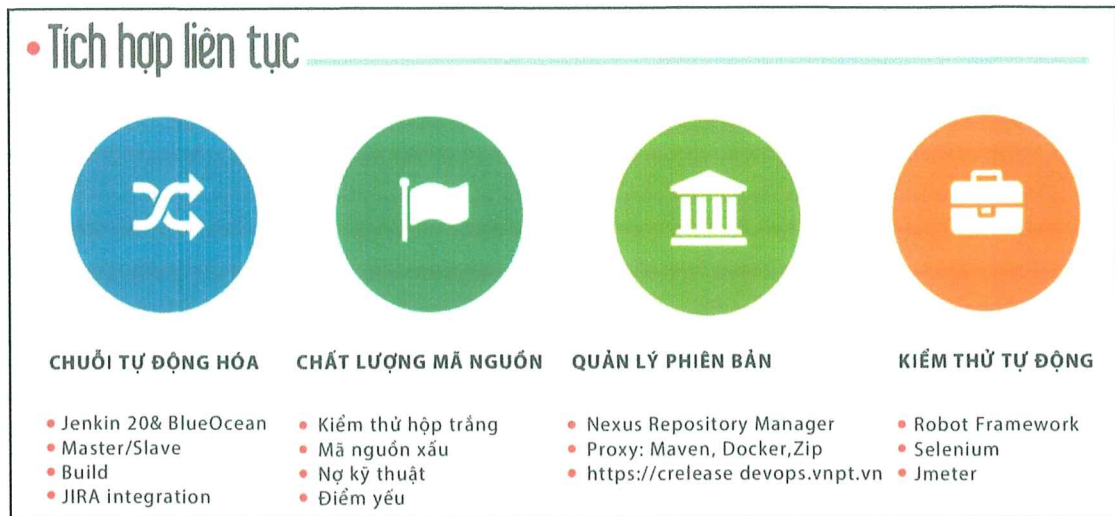
Được sử dụng trên ứng dụng Jenkin_Blue Ocean với chuỗi pipeline. Học viên đề nghị sẽ sử dụng kịch bản kiểm thử chức năng tự động sau khi triển khai ở môi trường kiểm thử hệ thống (System test).

RobotFramework cung cấp cho chúng ta câu lệnh để chạy ở cửa sổ dòng lệnh (Command line) như sau:

Robot -d <Đường dẫn của các file báo cáo HTML> <Đường dẫn của file Testsuite>

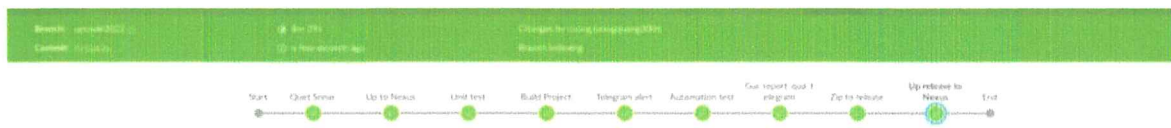
Trong câu lệnh này cú pháp -d là để cấu hình nơi lưu báo cáo HTML

Sau khi triển khai xong chuỗi pipeline DevOps sẽ như sau:



Hình 3.11. Các công cụ tích hợp liên tục sử dụng trong DevOps

Mỗi lần dev commit code lên theo như cấu hình của project trên Jenkin sau 1 phút Jenkin sẽ kiểm tra trên server chứa code (bitbucket) xem có bản build mới không nếu có sẽ lấy code về Jenkin agent và chạy phân tích tĩnh bằng phần mềm SonarQuebe sau khi phân tích tĩnh đạt yêu cầu hệ thống sẽ nén và đẩy một bản code lên lưu trữ ở server Nexus. Sau đó lấy bản build vừa lưu tại Nexus triển khai trên môi trường dev/Test. Sau khi triển khai trên môi trường dev/Test sẽ gọi đến Testsuite của Robot Framework và chạy các kịch bản kiểm thử chức năng. Nếu kịch bản kiểm thử chức năng đều đạt yêu cầu thì sẽ gửi kết quả lên nhóm chat và lưu giữ một bản trên server Nexus tại folder release



Hình 3.12. Quy trình DevOps triển khai cho vnEdu

3.2. Kết quả thực nghiệm được đề xuất

Đề án đã được học viên đề xuất và áp dụng hơn 1 năm, bắt đầu triển khai sử dụng từ ngày **10.12.2022**.

Đưa các kịch bản kiểm thử chức năng tự động cho hệ thống vnEdu để kiểm

thử trước và sau mỗi lần triển khai Upcode.

Hiện tại vnEdu upcode định kỳ vào ngày thứ 3 và thứ 5 hàng tuần, trước khi upcode hệ thống sẽ chạy bộ testcases trên site staging trước khi tạo package release nếu thành công các kịch bản đều được thông qua không có vấn đề nào xảy ra thì sẽ tạo package release. Sau đó sau khi package release được triển khai trên môi trường production thì hệ thống sẽ chạy lại bộ kịch bản kiểm thử chức năng tự động trên môi trường production và thông báo kết quả ngay sau khi chạy xong.

Sau một năm đưa đề xuất đề án vào sử dụng với vnEdu đã thu được một số kết quả tích cực như sau:

- Số lượng kịch bản kiểm thử chức năng đã tăng lên đáng kể từ **14** kịch bản thử nghiệm ban đầu đã tăng lên **160** kịch bản bao phủ hầu như toàn bộ các chức năng chính của vnEdu
- **100%** kiểm thử viên của trung tâm Giải pháp Giáo dục điện tử đã sử dụng thành thạo RobotFramework và có thể tự áp dụng RobotFramework cho các dự án mới
- Độ ổn định của kịch bản kiểm thử tăng lên nhờ có các cơ chế chạy lại kịch bản khi thất bại.
- Tích hợp vào quy trình DevOPs chạy kiểm thử chức năng tự động trên cả môi trường dev/test, Staging và Production (môi trường thật)
- Chạy một tuần 2 lần đều đặn vào ngày thứ 3 và thứ 5 trước và sau khi upcode của nhánh release và production
- Áp dụng tốt kiểm thử chức năng tự động trong các hoạt động như đổi pass server, chuyển hạ tầng hệ thống, thêm hạ tầng (scale) hệ thống.
- Kiểm tra độ ổn định của hệ thống hàng ngày.
- Thời gian chỉnh sửa kịch bản khi có thay đổi giảm đi đáng kể, đáp ứng nhanh mỗi thay đổi.

3.3. Đánh giá kết quả thực nghiệm

- Số lượng kịch bản tăng **800%** (từ **20** kịch bản thử nghiệm lên **160** kịch bản).

Link devOPs:

<http://cbuild.devops.vnpt.vn:8080/blue/organizations/jenkins/EEDU.VNED>

U/activity

- Số lượng issues được tìm ra sớm tăng dần theo thời gian (theo số lượng kịch bản tăng thêm). Tổng số lượng issue được tìm ra là **56** issues. Nguyên nhân chính của các issues này là do thay đổi ở các phần khác gây ảnh hưởng (thay đổi) tạo nên lỗi. Tăng đáng kể hiệu quả của kiểm thử hồi quy (regression test).

- Thời gian kiểm thử hồi quy của kiểm thử viên giảm đi đáng kể tầm **40%**. Trước khi áp dụng kiểm thử chức năng tự động cần **3** kiểm thử viên chạy kiểm thử hồi quy trong vòng **2h**. Sau khi áp dụng kiểm thử tự động chỉ cần một kiểm thử viên kiểm thử hồi quy với những kịch bản không kiểm thử tự động được trong vòng **1h** (tốc độ kiểm thử tăng lên **6** lần). Có nhiều thời gian hơn để thực hiện kiểm thử các kịch bản phức tạp hơn.

- Số lượng issue sau khi đưa lên môi trường production khi có kịch bản kiểm thử chức năng tự động giảm đi **25%** do các lỗi kiểm thử hồi quy đã được phát hiện từ sớm

- Về hiệu quả khi sử dụng kiểm thử tự động cho kiểm thử hồi quy của vnEdu:

+ Mỗi lần release trước khi có kiểm thử chức năng tự động sẽ tốn mất thời gian **2h** của **3** kiểm thử viên là **6h** để đảm bảo hệ thống trên môi trường staging và **2h** để kiểm thử hồi quy trên môi trường Production. Như vậy mỗi lần release trung bình sẽ mất **8h** để kiểm thử hồi quy. Một tuần có **2** lần update code lên production (Vào thứ 3 và thứ 5 hàng tuần) nên sẽ tốn **16h** cho việc kiểm thử hồi quy để đảm bảo hệ thống chạy ổn định. Một tháng trung bình có **8** lần release: **128h** và một năm số lượng thời gian dành cho kiểm thử hồi quy của vnEdu là: **1536h**

+ Sau khi có kiểm thử tự động mỗi lần upcode chỉ cần 1 nhân viên kiểm thử dùng **1h** để kiểm thử các kịch bản không kiểm thử được tự động trên môi trường realease và **1h** để kiểm thử trên môi trường production => Như vậy mỗi lần release nhân viên kiểm thử sẽ tốn khoảng **2h** để kiểm thử hồi quy cho các chức năng không áp dụng kiểm thử tự động và các tính năng mới chỉnh sửa. Thời gian tiết kiệm được

75% so với thời gian khi không dùng kiểm thử tự động. Thời gian dành cho kiểm thử hồi quy một tuần là: **4h** cũng giảm được **75%** thời gian so với kiểm thử hồi quy thủ công.

+ Thời gian xây dựng kịch bản kiểm thử trong giai đoạn mới bắt đầu việc đào tạo, cài đặt và làm quen với công cụ tốn khá nhiều thời gian trong 1 tháng đầu tiên cả team chỉ hoàn thành cài đặt, đào tạo và triển khai được **13** kịch bản kiểm thử đơn giản ở các chức năng như đăng nhập, vào quản lý nhà trường

+ Nhưng ở các tháng tiếp theo sau khi đã quen thời gian xây dựng kịch bản kiểm thử đã giảm đi nhiều cụ thể là tháng thứ 2 đã xây dựng được **35** kịch bản kiểm thử và tháng thứ 3 đã xây dựng **57** kịch bản kiểm thử.

+ Tương tự thời gian bảo trì kịch bản ở giai đoạn đầu là khá nhiều nhưng giảm dần theo thời gian do độ ổn định của hệ thống và độ thành thạo của các kiểm thử viên. Cụ thể tháng 3 năm 2023 thời gian bảo trì kịch bản là tầm **6h/tháng** cho **35** kịch bản đến tháng 12 năm 2023 thời gian bảo trì kịch bản vẫn là **5h/tháng** cho **160** kịch bản (Thời gian bảo trì 1 kịch bản giảm đi gần **5 lần**).

- Đề án xây dựng kiểm thử chức năng tự động đã được giải nhì phong trào triển khai DevOPs cấp công ty

- Các ứng dụng của RobotFramework đều được giải sáng kiến cấp công ty ví dụ như:

+ Sáng kiến “Xây dựng hệ thống healthcheck đảm bảo khả năng sử dụng của các hệ thống nâng cao trải nghiệm khách hàng, chủ động phòng ngừa, phát hiện kịp thời xử lý sự cố” đoạt giải Khuyến khích phong trào sáng kiến đợt 2 cấp công ty VNPT IT năm 2023

KẾT LUẬN CHƯƠNG

Trong chương này học viên đã trình bày thực tế việc triển khai kiểm thử tự động cho hệ sinh thái giáo dục vnEdu theo giải pháp đề xuất. Cũng như đánh giá những giá trị thu được cũng như những điểm còn tồn tại của giải pháp sau một thời gian thực nghiệm thực tế

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Trong quá trình xây dựng và triển khai thử nghiệm đề án, nhiều kết luận đã được học viên rút ra từ các kinh nghiệm thực tế, Những kết luận này không chỉ phản ánh những thành công mà còn có những thách thức cần học hỏi cải tiến từ quá trình triển khai kiểm thử tự động. Dưới đây là một số các kết luận được học viên rút ra

- **Hiệu quả của việc áp dụng Robot Framework:** Việc sử dụng RobotFramework đã mang lại những lợi ích rõ ràng trong việc tối ưu hóa quy trình kiểm thử. Việc sử dụng các thư viện và keywords có sẵn giúp giảm thiểu thời gian và công sức cần thiết cho việc phát triển và duy trì của các bộ kịch bản kiểm thử

- **Tính linh hoạt và mở rộng:** Mô hình kiểm thử tự động được đề xuất cùng việc sử dụng Robot Framework đã cho thấy tính linh hoạt cao trong việc mở rộng và điều chỉnh các kịch bản theo yêu cầu của dự án. Dễ dàng áp dụng triển khai cho nhiều dự án khác nhau. Việc sử dụng các thư viện và custom keywords đã giúp thuận tiện cho việc tùy chỉnh kịch bản để phù hợp với nhu cầu của từng dự án.

- **Sự tương thích:** Qua quá trình triển khai đề án học viên rút ra kết luận RobotFramework rất thích hợp để áp dụng kiểm thử cho các tính năng hay màn hình có danh sách, các form dữ liệu, các chức năng thêm, sửa, xóa dữ liệu. Và robot Framework không phù hợp để check các tính năng có gửi tin nhắn xác nhận qua điện thoại, các tính năng thiên về phòng chống tự động hóa như captcha hay các phần mềm nhúng hoặc phần cứng những ứng dụng phần mềm không có giao diện người dùng hay API.

- Hướng phát triển tương lai:

Mặc dù đề án mang lại hiệu quả mạnh mẽ và linh hoạt để kiểm thử chức năng tự động, nhưng vẫn có một số hạn chế cần khắc phục trong tương lai, có thể kể đến:

- Với số lượng kịch bản kiểm thử lớn sẽ tốn rất nhiều thời gian để chạy bộ kịch bản kiểm thử.
- Với các màn hình hay thao tác cần tương tác với các phần tử winform (popup của trình duyệt hay hệ điều hành) những phần tử này không có Xpath trong

khuôn khổ của đề án học viên chưa xử lý được.

- Chưa có nơi để quản lý tập trung kết quả chạy kiểm thử tự động

Với những hạn chế kể trên hướng phát triển của đề án sẽ nâng cấp để hỗ trợ giải quyết những vấn đề hạn chế kể trên:

- **Phát triển chạy song song:** Khi số lượng kịch bản lớn có thể nghiên cứu theo hướng để có thể chạy song song các kịch bản cùng lúc để tiết kiệm thời gian. Có thể nghiên cứu chạy đa luồng (parallel) hoặc triển khai máy ảo, docker để có thể chạy cùng lúc các kịch bản khác nhau để tiết kiệm thời gian chạy kiểm thử tự động.
- **Phát triển tương tác các phần tử không có Xpath:** Có thể sử dụng những công cụ thư viện xử lý phần tử theo tọa độ hay hình ảnh để có thể tương tác với các phần tử không có Xpath. Các thư viện hay ứng dụng hỗ trợ có thể kể đến là AutoIT, Sikuli
- **Phát triển hệ thống quản lý tập trung kết quả chạy kiểm thử tự động:** Trong khuôn khổ đề án kết quả kiểm thử tự động chỉ được lưu tại máy chạy kiểm thử tự động dưới dạng HTML và Excel sau đó gửi qua nhóm chat Telegram. Chưa có nơi quản lý tập trung các kết quả kiểm thử tự động. Trong tương lai cần xây dựng hệ thống để có thể quản lý tập trung kết quả kiểm thử tự động, từ đó thể hiện trực quan kết quả dưới dạng biểu đồ, dashboard hoặc so sánh kết quả kiểm thử tự động với các lần chạy trước đó.

DANH MỤC TÀI LIỆU THAM KHẢO

Tiếng Việt:

[1] <https://vnedu.vn/>

[2] TS. Đỗ Thị Bích Ngọc và TS. Phan Hoài Phương (2020) “Bài giảng đảm bảo chất lượng phần mềm”

Tiếng Anh:

[3] <https://robotframework.org/>

[4] Harsha T and B A Sujatha Kumari (2017) Software Test Automation with Robot Framework

[5] Stanislav Stresnjak and Zeljko Hocenski (2011) Usage of Robot Framework in Automation of Functional Test Regression

[6] Neha S Batni, Jyoti Shetty (2018) A Comprehensive Study on Automation using Robot Framework

[7] Mark Collin (2015), Mastering Selenium WebDriver, Packt Publishing

[8] David Burns (2012), Selenium 2 Testing tools Beginner's Guide book.


BẢN CAM ĐOAN

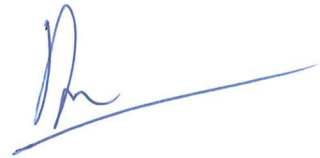
Tôi cam đoan đã thực hiện việc kiểm tra mức độ tương đồng nội dung đề án qua phần mềm DoIT một cách trung thực và đạt kết quả mức độ tương đồng ...% toàn bộ nội dung đề án. Bản đề án kiểm tra qua phần mềm là bản cứng đề án đã nộp để bảo vệ trước hội đồng. Nếu sai tôi xin chịu các hình thức kỷ luật theo quy định hiện hành của Học viện.

....., ngày tháng năm

HỌC VIÊN CAO HỌC/NCS

(Ký và ghi rõ họ tên)


Lương Quang Cường



Kết quả kiểm tra trùng lặp #15169319

KiểmTraTàiLieu



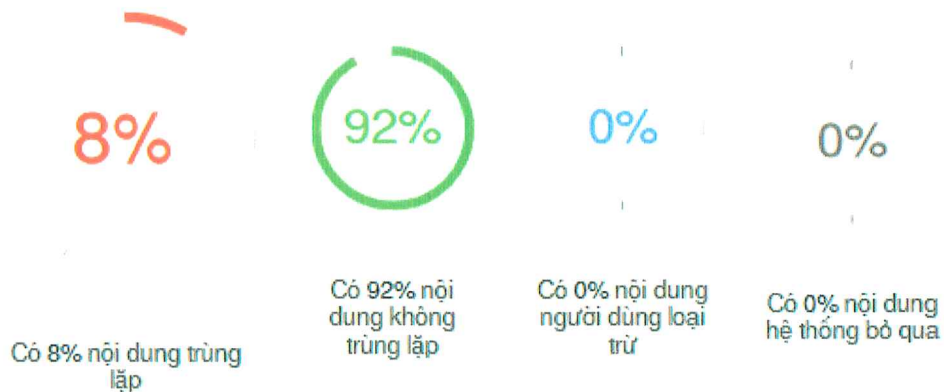
BÁO CÁO KIỂM TRA TRÙNG LẬP

Thông tin tài liệu

Tên tài liệu:	CS_SAU_BAOVE_LuongQuangCuong_Dean_KIEMTHUTUDO NGCHO_VNEDU_v1.0.4
Tác giả:	Cuong Luong Quang
Điểm trùng lặp:	8
Thời gian tải lên:	06:52 18/06/2024
Thời gian sinh báo cáo:	06:54 18/06/2024
Các trang kiểm tra:	65/65 trang



Kết quả kiểm tra trùng lặp



Nguồn trùng lặp tiêu biểu

123docz.net tailieu.vn nghean.vnpt.vn

BIÊN BẢN
HỌP HỘI ĐỒNG CHẤM ĐỀ ÁN TỐT NGHIỆP THẠC SĨ

Căn cứ quyết định số Quyết định số 871/QĐ-HV ngày 30 tháng 05 năm 2024 của Giám đốc Học viện Công nghệ Bưu chính Viễn thông về việc thành lập Hội đồng chấm đề án tốt nghiệp thạc sĩ. Hội đồng đã họp vào hồi 09 giờ 15 phút, ngày 12 tháng 6 năm 2024 tại Học viện Công nghệ Bưu chính Viễn thông để chấm đề án tốt nghiệp thạc sĩ cho:

Học viên: **Lương Quang Cường**

Tên đề án tốt nghiệp: **Nghiên cứu phương pháp kiểm thử chức năng tự động và áp dụng cho hệ thống VNEDU**

Chuyên ngành: **Hệ thống thông tin**

Mã số: **8.48.01.04**

Các thành viên của Hội đồng chấm đề án tốt nghiệp có mặt: 5 / 05

TT	HỌ VÀ TÊN	TRÁCH NHIỆM TRONG HỌP	GHI CHÚ
1	PGS.TS. Trần Quang Anh	Chủ tịch	
2	TS. Đinh Trường Duy	Thư ký	
3	TS. Vũ Văn Thỏa	Phản biện 1	
4	TS. Nguyễn Xuân Thắng	Phản biện 2	
5	TS. Nguyễn Thanh Tuyên	Ủy viên	

Các nội dung thực hiện:

1. Chủ tịch Hội đồng điều khiển buổi họp. Công bố quyết định của Giám đốc Học viện Công nghệ Bưu chính Viễn thông về việc thành lập Hội đồng chấm đề án tốt nghiệp thạc sĩ.
2. Người hướng dẫn khoa học hoặc thư ký đọc lý lịch khoa học và các điều kiện bảo vệ đề án tốt nghiệp của học viên (Có bản lý lịch khoa học và kết quả các môn học cao học của học viên kèm theo).
3. Học viên trình bày tóm tắt đề án tốt nghiệp.
4. Phản biện 1 đọc nhận xét (có văn bản kèm theo)
5. Phản biện 2 đọc nhận xét (có văn bản kèm theo)
6. Các câu hỏi của thành viên Hội đồng:

- Giới thiệu chi tiết hơn về hệ thống
- Giới thiệu rõ hơn về các bước xây dựng quy trình kiểm thử hệ thống?
- Giới thiệu các tiên đề định nghĩa hệ testcase phân lớp?
- Trình bày giải pháp xử lý nếu quy trình kiểm thử thất bại do ngoại cảnh, khi sử dụng testing tool thì sao?
- Giới thiệu tại sao sử dụng robot Framework?

7. Trả lời của học viên:

- Học viên thảo luận và trả lời các câu hỏi của hội đồng.
- Học viên tiếp thu các ý kiến và chỉnh sửa trong đề án.

8. Thư ký đọc nhận xét về quá trình thực hiện đề án tốt nghiệp của học viên (có văn bản kèm theo).

9. Hội đồng họp riêng:

- Bầu Ban kiểm phiếu:

1. Trưởng Ban kiểm phiếu: Nguyễn Thanh Tuyền
2. Ủy viên Ban kiểm phiếu: Nguyễn Xuân Thắng
3. Ủy viên Ban kiểm phiếu: Đinh Trường Duy

- Hội đồng chấm đề án tốt nghiệp bằng bỏ phiếu kín.

- Ban kiểm phiếu làm việc:

- Trưởng Ban kiểm phiếu báo cáo kết quả kiểm phiếu (có Biên bản họp Ban kiểm phiếu kèm theo)

- Điểm trung bình của đề án tốt nghiệp: 8,1

Kết luận:

1. Các nội dung cần chỉnh sửa, hoàn thiện sau bảo vệ đề án tốt nghiệp:

- Bổ sung nội dung các bước xây dựng giải pháp kiểm thử tự động cho hệ thống VNEDU.
- Bổ sung tham chiếu đến tài liệu các tiêu chuẩn quốc tế với nội dung trong test-case và minh chứng cho bất luận tăng số kích bản.
- Giải thích chi tiết các đối tượng phục vụ và các chức năng chính của hệ thống VNEDU tại bảng 1.1 (tr 7)
- So sánh thêm các công cụ khác với công cụ đề án chọn

2. Đề nghị Học viện công nhận (hoặc không) và cấp bằng (hoặc không) thạc sĩ cho học viên: Vương Quang Cường

3. Đề án tốt nghiệp có thể phát triển thành đề tài nghiên cứu cho

NCS.....

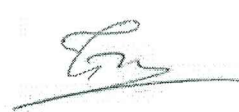
Buổi làm việc kết thúc vào 10g.00 cùng ngày.

Chủ tịch



PGS.TS. Trần Quang Anh

Thư ký



TS. Đinh Trường Duy

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập – Tự do – Hạnh phúc

BẢN NHẬN XÉT LUẬN VĂN TỐT NGHIỆP THẠC SĨ

(Dùng cho người phản biện)

Tên đề tài luận văn: Nghiên cứu phương pháp kiểm thử chức năng tự động và áp dụng cho hệ thống VNEDU

Chuyên ngành: Hệ thống thông tin

Mã chuyên ngành: 8.48.01.04

Họ và tên học viên: Lương Quang Cường

Họ và tên người nhận xét: Nguyễn Xuân Thắng

Học hàm, học vị: Tiến sĩ

Chuyên ngành: Khoa học máy tính

Cơ quan công tác: Khoa Công nghệ thông tin - Trường Đại học Hà Nội

Số điện thoại: 0912543560.

E-mail: nxthang@hanu.edu.vn

NỘI DUNG NHẬN XÉT

I/ Cơ sở khoa học và thực tiễn, tính cấp thiết của đề tài:

Đề án nghiên cứu cơ sở lý thuyết về kiểm thử tự động và triển khai áp dụng các công cụ kiểm thử tự động cho các chức năng chính của ứng dụng Web vnEdu. Tác giả đề án đã luận giải được nhu cầu cấp thiết của đơn vị quản trị hệ thống vnEdu đối với việc xây dựng quy trình kiểm thử tự động và áp dụng công cụ kiểm thử tự động trong quá trình quản lý, nâng cấp hệ thống. Kết quả của đề án có tính ứng dụng thực tiễn tại Trung tâm Giải pháp giáo dục điện tử - VNPT IT và các đơn vị đang quản trị các ứng dụng web có quy mô tương tự.

II/ Nội dung của luận văn, các kết quả đã đạt được:

Nội dung đề án gồm 03 chương, trong chương 1, tác giả trình bày tổng quan về kiểm thử chức năng và phân tích hiện trạng kiểm thử chức năng cho hệ thống vnEdu. Một số kịch bản và công cụ kiểm thử chức năng tự động tiêu biểu cũng được trình bày trong chương này. Chương 2 đề xuất ứng dụng công cụ kiểm thử tự động RobotFramework và quy trình triển khai kiểm thử tự động một số chức năng của hệ thống vnEdu. Tác giả đã xây dựng được các biểu mẫu mô tả thành phần và cấu trúc của dự án kiểm thử tự động và dựa vào đó để xây dựng các kịch bản kiểm thử chức năng tự động cho hệ thống. Bên cạnh đó, tác giả đã xây dựng bổ sung tính năng xuất báo cáo kết quả kiểm thử ra file excel và gửi file báo cáo qua Telegram.

Trong Chương 3, tác giả trình bày về các kết quả thực nghiệm và đánh giá giải pháp. Phần này tác giả mô tả 14 kịch bản kiểm thử tự động cho các chức năng khác nhau của hệ thống và kết quả khi thực hiện kiểm thử tự động. Việc tích hợp kiểm thử tự động vào quy trình DevOps (kiểm thử trước và sau mỗi lần triển khai Upcode) cũng được tác giả mô tả trong phần này. Các kết quả thử nghiệm và đánh giá được trình bày cụ thể, rõ ràng.

III/ Những vấn đề cần giải thích thêm:

1. Tại sao tác giả đề xuất sử dụng RobotFramework mà không sử dụng các công cụ khác như Selenium hoặc Puppeteer?
2. Giải quyết vấn đề kiểm thử chức năng thất bại do các nguyên nhân khách quan như lỗi mạng, lag mạng nên không kịp hiện các phần tử giao diện ... như thế nào? Nếu số lượng kịch bản kiểm thử lớn dẫn đến thời gian chạy kiểm thử tự động kéo dài thì giải quyết như thế nào?

IV/ Kết luận:

Đồng ý cho phép học viên bảo vệ luận văn tốt nghiệp

Ngày 12 tháng 6 năm 2024

NGƯỜI NHẬN XÉT



Nguyễn Xuân Thắng

BẢN NHẬN XÉT ĐỀ ÁN TỐT NGHIỆP THẠC SĨ

Tên đề tài đề án: Nghiên cứu phương pháp kiểm thử chức năng tự động và áp dụng cho hệ thống VNEDU

Chuyên ngành: Hệ thống thông tin

Mã số: 8.48.01.04

Tên học viên: Lương Quang Cường

Họ tên người nhận xét: Vũ Văn Thò

Học hàm, học vị: Tiến sĩ **Chuyên ngành:** Toán học tính toán

Cơ quan công tác: Học viện Công nghệ Bưu chính Viễn thông

NỘI DUNG NHẬN XÉT

I/ Cơ sở khoa học và thực tiễn, tính cấp thiết của đề tài

Hiện nay, quá trình chuyển đổi số đang diễn ra rộng khắp trên thế giới cũng như tại Việt Nam. Qua trình đó dẫn đến sự bùng nổ các sản phẩm, dịch vụ đáp ứng các yêu cầu đa dạng của người dùng được triển khai trên nền tảng Internet. Các dịch vụ thường có phạm vi địa lý rộng trong một quốc gia hoặc toàn cầu, với số lượng người dùng đông đảo. Từ đó, vấn đề đảm bảo chất lượng dịch vụ có ý nghĩa sống còn đối với mọi nhà cung cấp dịch vụ. Vì vậy, vấn đề kiểm thử các hệ thống phần mềm đóng vai trò quan trọng trong suốt quá trình xây dựng, triển khai, vận hành và cập nhật hệ thống nhằm đảm bảo các hệ thống được kiểm tra kỹ lưỡng và đáng tin cậy. Quá trình kiểm thử phải đảm bảo các yêu cầu về tính hiệu quả, chính xác và nhanh chóng. Do đó, phương pháp kiểm thử tự động có vai trò quan trọng nhằm giảm bớt sự phụ thuộc của quá trình kiểm thử vào công sức và thời gian của con người.

Đề án với tiêu đề “Nghiên cứu phương pháp kiểm thử chức năng tự động và áp dụng cho hệ thống VNEDU” của học viên Lương Quang Cường tập trung nghiên cứu bộ test-case và triển khai kiểm thử chức năng tự động cho hệ thống VNEDU của tập đoàn Bưu chính Viễn thông Việt Nam. Vì vậy, đề tài đề án có tính cấp thiết, có ý nghĩa khoa học và thực tiễn.

II/ Về nội dung chất lượng của đề án, những kết quả đạt được

Đề án được trình bày trong 53 trang bao gồm phần mở đầu, 3 chương nội dung, phần kết luận và danh mục các tài liệu tham khảo.

Chương 1 của đề án gồm 18 trang, trình bày tổng quan các kiến thức cơ sở về kiểm thử nói chung và kiểm thử tự động các chức năng của hệ thống, giới thiệu hệ thống VNEDU và các vấn đề liên quan. Đề án cũng thực hiện khảo sát và đánh giá một số công cụ kiểm thử tự động chức năng phổ biến hiện nay. Tuy nhiên, một số nội dung trình bày về kiểm thử chức năng và kiểm thử chức năng tự động còn trùng lặp cần trình bày rõ ràng hơn. Phần giới thiệu về hệ thống VNEDU cần chi tiết hơn các giải pháp công nghệ của hệ thống, làm rõ các đối tượng người dùng, qui trình nghiệp vụ và các chức năng của hệ thống. Phần kết luận chương cần bổ sung số thứ tự và các nội dung liên kết với chương 2.

Nội dung chính của chương 2 là nghiên cứu đề xuất phương pháp kiểm thử tự động các chức năng cho hệ thống VNEDU được trình bày trong 16 trang. Đề án đã trình bày đánh giá về việc áp dụng công cụ kiểm thử tự động chức năng trong thực tế và nội dung các bước xây dựng giải pháp kiểm thử tự động chức năng cho hệ thống VNEDU. Trên cơ sở đó, đề án đề xuất giải pháp kiểm thử tự động chức năng cho hệ thống VNEDU. Tuy nhiên, các nội dung đánh giá về việc áp dụng công cụ kiểm thử tự động chức năng trong thực tế và nội dung các bước xây dựng giải pháp kiểm thử tự động chức năng cho hệ thống VNEDU còn sơ sài chưa gắn với các đặc điểm của hệ thống đã trình bày trong chương 1. Phần kết luận chương cần bổ sung số thứ tự và các nội dung liên kết với chương 3.

Chương 3 của đề án gồm 13 trang, trình bày các nội dung triển khai kiểm thử chức năng tự động cho hệ thống VNEDU và đánh giá kết quả thử nghiệm. Đề án đã xây dựng được 14 test-case và triển khai đánh giá thực tế hệ thống VNEDU theo bộ test-case đã xây dựng. Tuy nhiên, trong nội dung mỗi test-case không có tham chiếu đến các tài liệu của các tổ chức tiêu chuẩn uy tín quốc tế (chẳng hạn tổ chức ISO) nên độ tin cậy chưa cao. Đề án nên bổ sung những nội dung này và trình bày thêm minh chứng cho kết luận về tăng số lượng kịch bản thử nghiệm từ 20 lên 160. Phần kết luận chương cần bổ sung số thứ tự và các kết quả chính đã được trình bày trong chương 3.

Đề án được chia thành các chương mục rõ ràng và cơ bản bám sát các nội dung theo đề cương đã được phê duyệt. Tuy nhiên, một số nội dung trong chương 2 đề án cần được trình bày logic hơn. Trong đề án còn có khá nhiều lỗi in ấn phải sửa chữa tại các trang 2, 3, 5, 6, 9, 11, 12, 15, 22, 24, 29, 30, 38, 50, ... Phần Danh mục tài liệu tham khảo cần được trình bày theo đúng qui định của Học viện, bổ sung thêm tài liệu về hệ thống VNEDU và trích dẫn phù hợp trong các chương của đề án.

III/ Những vấn đề cần giải thích thêm

- Giải thích chi tiết hơn các đối tượng phục vụ và các chức năng chính của hệ thống VNEDU được trình bày tại bảng 1.1 (trang 7)?
- Giải thích rõ thêm các bước xây dựng giải pháp kiểm thử tự động chức năng cho hệ thống VNEDU trình bày tại mục 2.2 (trang 23, 24)?
- Giải thích thêm về tiêu chí đánh giá bộ test-case kiểm thử tự động hệ thống VNEDU và một số kịch bản kiểm thử mở rộng đã thực hiện của học viên?

IV/ Kết luận

Tôi đồng ý cho học viên Lương Quang Cường được bảo vệ đề án với đề tài “Nghiên cứu phương pháp kiểm thử chức năng tự động và áp dụng cho hệ thống VNEDU” trước Hội đồng chấm đề án tốt nghiệp thạc sĩ của Học viện. Tuy nhiên, để đảm bảo chất lượng đề án tốt nghiệp thạc sĩ chuyên ngành Hệ thống thông tin, học viên cần chỉnh sửa đề án và giải trình rõ trước Hội đồng các vấn đề được góp ý trên đây.

Ngày 06 tháng 06 năm 2024

Người nhận xét



TS Vũ Văn Thỏa

BÁO CÁO GIẢI TRÌNH
SỬA CHỮA, HOÀN THIỆN ĐỀ ÁN TỐT NGHIỆP THẠC SĨ

Họ và tên học viên: Lương Quang Cường
Chuyên ngành: Hệ thống thông tin
Khóa: Khóa 2022 đợt 2
Tên đề tài: Nghiên cứu phương pháp kiểm thử chức năng tự động và áp dụng cho hệ thống VNEDU
Người hướng dẫn khoa học: TS. Đỗ Thị Bích Ngọc
Ngày bảo vệ: 12/06/2024

Các nội dung học viên đã sửa chữa, bổ sung trong đề án tốt nghiệp theo ý kiến đóng góp của Hội đồng chấm đề án tốt nghiệp:

TT	Ý kiến hội đồng	Sửa chữa của học viên
1	Bổ sung nội dung các bước xây dựng giải pháp kiểm thử tự động chức năng cho hệ thống VN EDU	Bổ sung chi tiết việc đánh giá mục 2.2 (Trang 24, Trang 25)
2	Bổ sung tham chiếu đến tài liệu các tiêu chuẩn quốc tế với nội dung trong test-case và minh chứng cho kết luận tăng số kịch bản	Bổ sung sở cứ và tham chiếu về tính đúng đắn của bộ testcases (Trang 38). Bổ sung link tham chiếu việc tăng testcases (Trang 49)
3	Giải thích chi tiết các đối tượng phục vụ và các chức năng chính của hệ thống VN EDU tại bảng 1.1	Bổ sung các đối tượng phục vụ và chức năng chính của hệ sinh thái vnEdu (Trang 7)
4	So sánh thêm các công cụ khác với công cụ đề án chọn	Bổ sung phân tích so sánh và bảng so sánh các công cụ kiểm thử tự động (Trang 17)

Hà Nội, ngày..... tháng năm 20...

Ký xác nhận của

CHỦ TỊCH HỘI ĐỒNG
CHẤM LUẬN VĂN

THƯ KÝ HỘI ĐỒNG

NGƯỜI HƯỚNG DẪN
KHOA HỌC

HỌC VIÊN



PGS.TS. Trần Quang Anh TS. Đinh Trường Duy TS. Đỗ Thị Bích Ngọc Lương Quang Cường