



**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM
KHOA CÔNG NGHỆ THÔNG TIN**

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

Đề tài: Kiểm thử website xem phim online

Giảng viên hướng dẫn: **ThS. Dương Thành Phết**

Sinh viên thực hiện:

1. Trần Lâm Anh	2080600160	20DTHE2
2. Lý Huỳnh Minh Triết	1911062102	19DTHC2
3. Nguyễn Quốc Duy	2011060151	20DTHC5
4. Trần Văn Dự	2080601110	20DTHE2

TP. Hồ Chí Minh, 2024

TRANG PHỤC LỤC

BÁO CÁO ĐỒ ÁN KIỂM THỬ VÀ ĐẢM BẢO CHẤT LƯỢNG PHẦN MỀM

Đề tài: **Kiểm thử website xem phim online**

Giảng viên hướng dẫn: **ThS. Dương Thành Phết**

Sinh viên thực hiện:

1. Trần Lâm Anh	2080600160	20DTHE2
2. Lý Huỳnh Minh Triết	1911062102	19DTHC2
3. Nguyễn Quốc Duy	2011060151	20DTHC5
4. Trần Văn Dự	2080601110	20DTHE2

TP. Hồ Chí Minh, 2024

BẢNG PHÂN CÔNG NHIỆM VỤ

Bảng đánh giá đóng góp của thành viên nhóm 7

STT	Thành viên	Công việc	Đánh giá (%)	Điểm
1	Trần Lâm Anh	<ul style="list-style-type: none"> - Kiểm thử hộp trắng – đường dẫn cơ sở - Kiểm thử hộp trắng – bao phủ - Kiểm thử tự động (Katalon Studio) - Giới thiệu kiểm thử phần mềm - Sửa và hoàn thiện báo cáo - Viết báo cáo tuần 3 	30%	8
2	Nguyễn Quốc Duy	<ul style="list-style-type: none"> - Kiểm thử tự động (TestComplete) - Kiểm thử hộp đen (Kiểm thử vùng tương đương) - Giới thiệu các phương pháp thiết kế test-case và mô tả chương trình - Sửa và hoàn thiện báo cáo - Viết báo cáo tuần 4 	30%	8
3	Lý Huỳnh Minh Triết	<ul style="list-style-type: none"> - Kiểm thử tự động (Selenium) - Giới thiệu phần mềm và lỗi - phần mềm - Sửa và hoàn thiện báo cáo - Làm video demo 	20%	7.5
4	Trần Văn Dự	<ul style="list-style-type: none"> - Kiểm thử tự động (Telerik) - Giới thiệu các phương pháp thiết kế test-case - Sửa và hoàn thiện báo cáo - Viết báo cáo tuần 4 	20%	7.5

LỜI CAM ĐOAN

Chúng em cam đoan rằng báo cáo đồ án này được trình bày dưới đây là kết quả nghiên cứu và thực hiện của chính chúng em dưới sự hướng dẫn của giảng viên **ThS.Dương Thành Phết**. Chúng em xin cam đoan rằng:

1. Báo cáo đồ án này được thực hiện dựa trên kiến thức và kỹ năng của nhóm em, và không sao chép hoặc sử dụng bất kỳ nguồn thông tin nào mà không được trích dẫn.
 2. Các kết quả và ý kiến trong báo cáo đồ án này là chân thực và chưa từng được công bố trước đây.
 3. Tất cả các tài liệu, nguồn thông tin, và phần mềm được sử dụng trong quá trình nghiên cứu và phát triển đã được trích dẫn và đề cập đầy đủ trong phần tham khảo của báo cáo.
 4. Mọi sự hỗ trợ và tư vấn từ giảng viên hướng dẫn đã được ghi nhận và biểu đạt sự biết ơn trong phần cảm ơn của báo cáo.
- Nhóm em hiểu rằng báo cáo đồ án này có thể được sử dụng để đánh giá và kiểm tra năng lực của chúng em trong quá trình học tập. chúng em xin chịu trách nhiệm hoàn toàn về nội dung và kết quả của đồ án này.

LỜI CẢM ƠN

Chúng em xin gửi lời cảm ơn chân thành đến thầy **ThS. Dương Thành Phết** - giảng viên dạy chúng em môn kiểm thử và đảm bảo chất lượng phần mềm trong Khoa Công Nghệ Thông Tin đã đồng hành cùng chúng em trong suốt quá trình làm đồ án, đã trang bị cho chúng em những kiến thức, kỹ năng cơ bản và chuyên sâu để chúng em có thể hoàn thành đồ án này.

Trong suốt thời gian thực hiện đồ án, chúng em cũng đã cố gắng hết sức hoàn thành đồ án. Nhưng do kiến thức còn hạn hẹp và chưa có những kinh nghiệm thực tế, nên không tránh khỏi những sai sót. Kính mong nhận được sự góp ý và giúp đỡ từ quý thầy để em có thể khắc phục những hạn chế và cải thiện sản phẩm một cách tốt nhất.

Em xin chân thành cảm ơn!

MỤC LỤC

TRANG PHỤC LỤC.....	1
BẢNG PHÂN CÔNG NHIỆM VỤ	2
LỜI CAM ĐOAN	3
LỜI CẢM ƠN	4
DANH MỤC KÝ HIỆU, CÁC CHỮ VIẾT TẮT	7
DANH MỤC CÁC BẢNG	8
DANH MỤC CÁC ẢNH.....	9
CHƯƠNG 1: GIỚI THIỆU	11
1.1. Phần mềm và lỗi phần mềm.....	11
1.1.1. Phần mềm	11
1.1.2. Lỗi phần mềm	11
1.1.3. Yêu cầu của khách hàng	12
1.2. Kiểm thử phần mềm	12
1.2.1. Khái niệm	12
1.2.2. Vai trò của kiểm thử phần mềm.....	12
1.2.3. Các quy tắc trong kiểm thử phần mềm.....	13
1.2.4. Các phương pháp trong kiểm thử	15
1.2.5. Kết luận.....	19
CHƯƠNG 2: THIẾT KẾ TEST-CASE	20
2.1. Các phương pháp thiết kế test-case	20
2.1.1. Kỹ thuật phân vùng tương đương (Equivalence Partitioning)	20
2.1.2. Kỹ thuật phân tích giá trị biên (Boundary-value Analysis)	20
2.1.3. Bảng quyết định – Decision Table Exercise	21
2.1.4. Dịch chuyển trạng thái.....	22
2.2. Mô tả chương trình	23
CHƯƠNG 3: TRIỂN KHAI KIỂM THỬ	24
3.1. Kiểm thử hộp đen.....	24
3.2. Kiểm thử hộp trắng.....	25
3.2.1. Kiểm thử đường dẫn cơ sở	25
3.2.2. Kiểm thử bao phủ	27
CHƯƠNG 4: KIỂM THỬ TỰ ĐỘNG	29
4.1. Áp dụng công cụ kiểm thử Katalon	29
4.1.1. Giới thiệu	29
4.1.2. Các tính năng chính	29

4.1.3. Triển khai kiểm thử.....	30
4.1.4. Các test-case thực hiện	32
4.2. Áp dụng công cụ kiểm thử Test-Complete.....	40
4.2.1. Giới thiệu	40
4.2.2. Các tính năng chính	41
4.2.3. Triển khai kiểm thử.....	42
4.3. Áp dụng công cụ kiểm thử Selenium.....	46
4.3.1. Giới thiệu	46
4.3.2. Các chức năng chính:	47
4.3.3. Triển khai kiểm thử:	48
4.4. Công cụ kiểm thử tự động Telerik	54
4.4.1. Giới thiệu	54
4.4.2. Các tính năng chính	55
4.4.3. Triển khai kiểm thử.....	55
CHƯƠNG 5: KẾT LUẬN	60
5.1. Kết luận.....	60
5.2. Kiến nghị.....	60
TÀI LIỆU THAM KHẢO	61

DANH MỤC KÝ HIỆU, CÁC CHỮ VIẾT TẮT

STT	Chữ viết tắt	Chữ đầy đủ
1	Pass	Kiểm thử đạt
2	Fail	Kiểm thử không đạt
3	Test	Kiểm thử
4	Tester	Người kiểm thử
5	Test Case	Bản kiểm thử
6	NULL	Rỗng
7	Bug	Lỗi

DANH MỤC CÁC BẢNG

Bảng 1. 1. Ưu điểm và nhược điểm của kiểm thử hộp đen	15
Bảng 1. 2. Ưu điểm và nhược điểm của kiểm thử hộp trắng.....	16
Bảng 1. 3. Ưu điểm và nhược điểm của kiểm thử hộp xám	17
Bảng 1. 4. So sánh các loại kiểm thử	19
Bảng 3. 1. Test case đăng nhập	27
Bảng 3. 2. Test case tìm kiếm	28
Bảng 4. 1. Test case Katalon Studio	40

DANH MỤC CÁC ẢNH

Hình 1. 1. Vòng đời của quá trình kiểm thử.....	13
Hình 3. 1. Vùng tương đương.....	24
Hình 3. 2. Phân tích biên	24
Hình 3. 3. Xây dựng vùng tương đương và phân tích viên	25
Hình 3. 4. Code chức năng đăng nhập.....	26
Hình 3. 5. Code chức năng tìm kiếm.....	27
Hình 4. 1. Katalon Studio	29
Hình 4. 2. Giao diện Katalon Studio	30
Hình 4. 3: Record test case	31
Hình 4. 4. Kịch bản test case	31
Hình 4. 5. Test thành công.....	32
Hình 4. 6. TestComplete	40
Hình 4. 7. Giao diện TestComplete	42
Hình 4. 8. Tạo project.....	43
Hình 4. 9. Record test steps	43
Hình 4. 10. Bắt đầu record.....	44
Hình 4. 11. Áp dụng kiểm thử	44
Hình 4. 12. Áp dụng kiểm thử	45
Hình 4. 13. Áp dụng kiểm thử	45
Hình 4. 14. Kiểm thử thành công	46
Hình 4. 15. Selenium	46
Hình 4. 16. Selenium	47
Hình 4. 17. Tạo project.....	48
Hình 4. 18. Đặt tên Project	49
Hình 4. 19. Gắn link web tiến hành test	49
Hình 4. 20. Giao diện Selenium	49
Hình 4. 21. Giao diện web bắt đầu ghi test	50
Hình 4. 22. Thực hiện quá trình kiểm thử	50
Hình 4. 23. Chạy kết quả test	51

Hình 4. 24. Cho kết quả test	51
Hình 4. 25. Cho kết quả test	52
Hình 4. 26. Thực hiện quá trình test.....	52
Hình 4. 27. Cho kết quả test	53
Hình 4. 28. Thực hiện quá trình test.....	53
Hình 4. 29. Cho kết quả test	54
Hình 4. 30. Telerik Test Studio	54
Hình 4. 31. Xuất file CSV	55
Hình 4. 32. Nhập link url.....	56
Hình 4. 33. Kịch bản kiểm thử	57
Hình 4. 34. Áp dụng kiểm thử.....	57
Hình 4. 35. Áp dụng kiểm thử.....	58
Hình 4. 36. Áp dụng kiểm thử.....	59
Hình 4. 37. Áp dụng kiểm thử.....	59

CHƯƠNG 1: GIỚI THIỆU

1.1. Phần mềm và lỗi phần mềm

1.1.1. Phần mềm

Phần mềm thường được mô tả với ba bộ phận cấu thành:

- Tập các lệnh (chương trình máy tính) trên máy tính khi được thực hiện sẽ tạo ra các dịch vụ và đem lại những kết quả mong muốn cho người dùng.
- Các cấu trúc dữ liệu (lưu giữ trên các bộ nhớ) làm cho chương trình thao tác hiệu quả với các thông tin thích hợp và nội dung thông tin được số hoá.
- Các tài liệu để mô tả các thao tác, cách sử dụng và bảo trì phần mềm (hướng dẫn sử dụng, tài liệu kỹ thuật, tài liệu phân tích, thiết kế, kiểm thử,...).

1.1.2. Lỗi phần mềm

Lỗi phần mềm nhìn chung là sự không khớp giữa chương trình và đặc tả của nó, kéo theo những vấn đề xuất hiện trong các giai đoạn phát triển phần mềm.

- Lỗi phần mềm thường xuất hiện ở các hình thức sau đây:
 - Sai (Fault): Khi phần mềm gặp lỗi sẽ đưa đến những sai sót. Tuy nhiên, không dễ để phát hiện ra sai sót trong quá trình phát triển phần mềm. Sai lầm có thể xuất hiện ở ngay đầu quy trình phát triển phần mềm khi người phân tích, thiết kế bỏ sót thông tin dẫn tới thiếu chức năng mà lẽ ra cần phải có.
 - Thất bại (Failure): Thất bại dễ nhận thấy nhất khi một lỗi được thực thi. Chúng thường xuất hiện dưới 2 dạng: thất bại có thể chạy được (ví dụ như mã nguồn) và thất bại chỉ liên kết với các lỗi về nhiệm vụ. Ngoài ra, có thể kể đến các thất bại liên quan tới các lỗi do bỏ quên. Chúng ta có thể hạn chế thất bại ngay tại bước đầu tiên của quy trình phát triển phần mềm nếu việc khảo sát được thực hiện tốt.
 - Sự cố (Incident): *Sự cố* thường được liên kết với một thất bại. Tuy nhiên nó khác với thất bại ở chỗ *sự cố* luôn hiển thị cho người dùng hoặc kiểm thử viên biết về sự tồn tại của nó.

- Thừa: 1 số chức năng không có trong bản đặc tả yêu cầu phần mềm nhưng lại xuất hiện trong phần mềm được xây dựng.

Ngoài ra, còn xuất hiện 1 số lỗi phi chức năng như phần mềm khó sử dụng, tốc độ không đáp ứng yêu cầu (vấn đề hiệu năng) hay giao diện khó nhìn cũng dễ khiến cho người sử dụng nghĩ rằng phần mềm đang hoạt động không đúng.

1.1.3. Yêu cầu của khách hàng

Phần mềm được phát triển dựa trên nhu cầu của khách hàng. Chính vì lẽ đó, các chức năng của phần mềm được xây dựng dựa trên việc thu thập, phân tích, khảo sát nhu cầu của khách hàng thông qua những yêu cầu cụ thể. Đối với phần mềm, yêu cầu thường được tổng hợp từ nhiều người, nhiều tổ chức có mức độ chuyên môn và mức độ tham gia cũng như tương tác với phần mềm khác nhau trong môi trường hoạt động của nó.

1.2. Kiểm thử phần mềm

1.2.1. Khái niệm

Kiểm thử phần mềm là một cuộc kiểm tra được tiến hành để cung cấp cho các bên liên quan thông tin về chất lượng của sản phẩm hoặc dịch vụ được kiểm thử. Hiểu theo cách đơn giản hơn, kiểm thử phần mềm là quá trình tìm thất bại hoặc chứng tỏ việc tiến hành của phần mềm là đúng đắn.

1.2.2. Vai trò của kiểm thử phần mềm

Kiểm thử phần mềm chiếm một vị trí quan trọng trong việc nâng cao chất lượng cũng như độ tin cậy của phần mềm trong quá trình phát triển. Hoàn thành vòng quay “đưa lỗi vào - tìm lỗi - khử lỗi đi” của quy trình kiểm thử phần mềm sẽ thu lại được những cải tiến đáng kể cho chất lượng sản phẩm phần mềm. Việc biết được sản phẩm phần mềm tốt tới mức nào trước khi đưa vào sử dụng sẽ hạn chế tối đa những rủi ro gặp phải trong quá trình phát triển phần mềm.

QUY TRÌNH KIỂM THỬ PHẦN MỀM



Hình 1. 1.Vòng đời của quá trình kiểm thử.

1.2.3. Các quy tắc trong kiểm thử phần mềm

Trong kiểm thử phần mềm có 7 nguyên tắc kiểm thử. Tìm hiểu chúng là 1 điều rất quan trọng bởi vì nó giúp tiết kiệm thời gian cũng như công sức truy lùng bug ẩn trong các ứng dụng của bạn. Có một câu hỏi đặt ra như thế này: “Bạn có thật sự hiểu thế nào là nguyên tắc?”. Về cơ bản, nguyên tắc chính là những quy định hoặc là luật mà chúng ta phải tuân theo. Hoặc bạn cũng có thể hiểu nguyên tắc là điều thiết yếu cần phải theo đuổi để phát triển hệ thống tốt nhất. Việc bỏ qua bất kỳ nguyên tắc nào cũng có thể làm giảm hiệu quả của việc thực hiện hệ thống. Vậy chúng ta sẽ nói về 7 nguyên tắc trong kiểm thử phần mềm:

1.2.3.1. Kiểm thử đưa ra lỗi

Kiểm thử có thể cho thấy rằng phần mềm đang có lỗi, nhưng không thể chứng minh rằng phần mềm không có lỗi. Kiểm thử được thực hiện bằng những kỹ thuật khác nhau. Kiểm thử làm giảm xác suất lỗi chưa tìm thấy vẫn còn trong phần mềm, ngay cả khi đã kiểm thử nghiêm ngặt phần mềm vẫn có thể còn lỗi. Vì vậy chúng ta phải tìm được càng nhiều lỗi càng tốt.

1.2.3.2. Kiểm thử cạn kiệt là không thể

Nguyên tắc này nói rằng kiểm tra mọi thứ trong phần mềm một cách trọn vẹn là không thể. Kiểm thử với tất cả các kết hợp đầu vào và đầu ra, với tất cả các kịch bản là không thể trừ phi nó chỉ bao gồm ít trường hợp thì có thể kiểm thử toàn bộ. Thay vì kiểm thử toàn bộ, việc phân tích rủi ro và dựa trên sự mức độ ưu tiên chúng ta có thể tập trung việc kiểm thử vào một số điểm cần thiết, có nguy cơ lỗi cao hơn.

1.2.3.3. Kiểm thử càng sớm càng tốt

Thông thường, lỗi tập trung vào những module, thành phần chức năng chính của hệ thống. Nếu xác định được điều này bạn sẽ tập trung vào tìm kiếm lỗi quanh khu vực được xác định. Nó được coi là một trong những cách hiệu quả nhất để thực hiện kiểm tra hiệu quả.

1.2.3.4. Sự tập trung của lỗi

Thông thường, lỗi tập trung vào những module, thành phần chức năng chính của hệ thống. Nếu xác định được điều này bạn sẽ tập trung vào tìm kiếm lỗi quanh khu vực được xác định. Nó được coi là một trong những cách hiệu quả nhất để thực hiện kiểm tra hiệu quả.

1.2.3.5. Nghịch lí thuốc trừ sâu

Nếu bạn sử dụng cùng một tập hợp các trường hợp kiểm thử liên tục, sau đó một thời gian các trường hợp kiểm thử không tìm thấy lỗi nào mới. Hiệu quả của các trường hợp kiểm thử bắt đầu giảm xuống sau một số lần thực hiện, vì vậy luôn luôn chúng ta phải luôn xem xét và sửa đổi các trường hợp kiểm thử trên một khoảng thời gian thường xuyên.

1.2.3.6. Kiểm thử phụ thuộc vào ngữ cảnh

Theo nguyên tắc này thì việc kiểm thử phụ thuộc vào ngữ cảnh và chúng ta phải tiếp cận kiểm thử theo nhiều ngữ cảnh khác nhau. Nếu bạn đang kiểm thử ứng dụng web và ứng dụng di động bằng cách sử dụng chiến lược kiểm thử giống nhau, thì đó là sai. Chiến lược để kiểm thử ứng dụng web sẽ khác với kiểm thử ứng dụng cho thiết bị di động của Android.

1.2.3.7. Không có lỗi - Sai lầm

Việc không tìm thấy lỗi trên sản phẩm không đồng nghĩa với việc sản phẩm đã sẵn sàng để tung ra thị trường. Việc không tìm thấy lỗi cũng có thể là do bộ trường hợp kiểm thử được tạo ra chỉ nhằm kiểm tra những tính năng được làm đúng theo yêu cầu thay vì nhằm tìm kiếm lỗi mới.

1.2.4. Các phương pháp trong kiểm thử

1.2.4.1. Kiểm thử hộp đen

Kiểm thử hộp đen là một phương pháp kiểm thử mà các tester không cần quan tâm đến các hoạt động bên trong hệ thống chạy ra sao, không cần quan tâm đến các dòng lệnh bên trong hệ thống hệ thống như thế nào. mà chỉ cần tập trung vào các giá trị đầu vào và các giá trị đầu ra của hệ thống có đúng với kết quả mong đợi của các trường hợp kiểm thử không để từ đó đánh giá chất lượng hệ thống.

Chính vì cơ chế như vậy nên phương pháp này có các ưu nhược điểm như sau:

Ưu điểm	Nhược điểm
- Rất phù hợp và hiệu quả khi mà số lượng các dòng lệnh của hệ thống là lớn.	- Bị giới hạn ở độ bao phủ của các trường hợp kiểm thử.
- Không cần truy cập vào các dòng lệnh.	- Sẽ không hiệu quả bởi thực tế các tester bị giới hạn kiến thức về hệ thống.
- Phân biệt được rõ ràng quan điểm của người dùng với quan điểm của nhà phát triển.	- Độ bao phủ sẽ bị thiếu vì tester không kiểm tra được các đoạn lệnh của hệ thống hoặc tập trung vào các dòng lệnh để xảy ra lỗi.
- Không cần đòi hỏi những kiến thức về ngôn ngữ lập trình ở các tester để có thể kiểm thử hệ thống.	- Sẽ khó để có thể thiết kế đầy đủ các trường hợp kiểm thử.

Bảng 1. 1. Ưu điểm và nhược điểm của kiểm thử hộp đen

1.2.4.2. Kiểm thử hộp trắng

Kiểm thử hộp trắng là việc nghiên cứu cụ thể chi tiết từng luồng hoạt động cũng như các dòng lệnh bên trong hệ thống. Kiểm thử hộp trắng cũng được gọi dưới các cái tên khác như: Glass testing hay open-box testing. Kiểm thử hộp trắng đòi hỏi tester phải có kiến thức về ngôn ngữ lập trình. Tester sẽ cần phải nghiên cứu vào bên trong hệ thống cụ thể là các dòng lệnh để tìm hiểu chúng có chạy đúng hay không.

Dưới đây là các ưu nhược điểm của phương pháp này:

Ưu điểm	Nhược điểm
- Đối với những tester có kiến thức về ngôn ngữ lập trình sẽ rất dễ dàng để phát hiện ra những lỗi ở trong các dòng lệnh.	- Trên thực tế việc sử dụng các tester có kiến thức về ngôn ngữ lập trình sẽ làm gia tăng giá thành để phát triển phần mềm.
- Giúp tối ưu hóa các dòng lệnh của hệ thống.	- Đôi lúc sẽ là không khả thi khi kiểm tra chi tiết từng dòng lệnh để có thể từ đó phát hiện ra các lỗi tiềm ẩn của hệ thống, có rất nhiều các luồng không thể kiểm tra được.
- Các dòng lệnh không cần thiết hoặc các dòng lệnh có khả năng mang đến các lỗi tiềm ẩn sẽ bị loại bỏ.	- Rất khó để duy trì phương pháp này liên tục, cần phải có những tool chuyên biệt như tool về phân tích code hay tool về phát hiện lỗi và sửa lỗi.
- Các tester có kiến thức về ngôn ngữ lập trình sau khi đã thực hiện phương pháp này thì sẽ dễ dàng đạt được độ bao phủ lớn nhất khi thực hiện thiết kế các trường hợp kiểm thử sau này.	

Bảng 1. 2. Ưu điểm và nhược điểm của kiểm thử hộp trắng

1.2.4.3. Kiểm thử hộp xám

Kiểm thử hộp xám là một phương pháp kiểm thử mà đòi hỏi tester phải có một lượng kiến thức nhất định về các luồng hoạt động ở bên trong hệ thống. Khác với kiểm thử hộp đen, phương pháp mà tester chỉ quan tâm duy nhất để việc kiểm thử thông qua giao diện người dùng, kiểm thử hộp xám đòi hỏi tester phải truy cập vào các tài liệu thiết kế hệ thống cũng như hệ thống cơ sở dữ liệu của hệ thống. Do đó mà tester có thể chuẩn bị tốt hơn những dữ liệu cho việc kiểm thử cũng như các trường hợp kiểm thử trong quá trình lên kế hoạch kiểm thử hệ thống.

Ưu điểm	Nhược điểm
- Vì là sự kết hợp giữa kiểm thử hộp trắng và kiểm thử hộp đen nên có được ưu điểm của cả hai phương pháp này.	- Vì phương pháp này không dựa trên việc truy cập code của hệ thống nên sẽ không tránh được việc độ bao phủ của các trường hợp kiểm thử bị giới hạn.
- Các tester sử dụng phương pháp này không dựa vào các dòng lệnh của hệ thống mà chủ yếu dựa trên các tài liệu định nghĩa giao diện cũng như các tài liệu đặc tả chức năng.	- Khi sử dụng phương pháp này thì nhiều trường hợp kiểm thử có thể bị dư thừa nếu mà những nhà thiết kế phần mềm đã chạy các trường hợp kiểm thử này trước đó.
- Trong phương pháp này các tester có thể thiết kế nên những trường hợp kiểm thử đặc biệt xung quanh các giao thức kết nối và các loại dữ liệu khác nhau.	- Việc kiểm tra tất cả các luồng đầu vào của hệ thống là không thể thực hiện được vì bị giới hạn về mặt thời gian kiểm thử và sẽ dẫn đến có rất nhiều các luồng hoạt động của hệ thống không được kiểm tra.
- Việc kiểm thử được hoàn thành từ góc nhìn của người dùng chứ không phải từ nhà thiết kế.	

Bảng 1. 3. Ưu điểm và nhược điểm của kiểm thử hộp xám

1.2.4.4. Bảng so sánh giữa các phương pháp kiểm thử

Kiểm thử hộp đen.	Kiểm thử hộp xám	Kiểm thử hộp trắng
Không cần quan tâm đến các luồng hoạt động trong hệ thống.	Cần có kiến thức nhất định về các luồng hoạt động bên trong hệ thống.	Cần nắm được toàn bộ các luồng hoạt động bên trong hệ thống.
Được biết đến với các tên gọi khác như: closed-box testing, datadriven testing hoặc functional testing.	Được biết đến với các tên gọi khác như: translucent testing.	Được biết đến với các tên gọi khác như: clear-box testing hoặc code-based testing.
Được thực hiện bởi người dùng cuối, kiểm thử viên và lập trình viên.	Được thực hiện bởi người dùng cuối, kiểm thử viên và lập trình viên.	Thường thì được hoàn thành bởi kiểm thử viên và lập trình viên.
Việc kiểm thử dựa trên kết quả mong muốn và kết quả thực tế mà hệ thống trả về.	Việc kiểm thử dựa trên các sơ đồ về cơ sở dữ liệu và sơ đồ về các luồng dữ liệu.	Dựa trên toàn bộ kiến thức về các luồng hoạt động bên trong hệ thống và các bộ dữ liệu kiểm thử phù hợp mà các kiểm thử viên tự thiết kế.
Vì chỉ quan tâm đến các giá trị đầu vào, kết quả đầu ra và kết quả mong đợi nên đây là phương pháp tốn ít thời gian nhất cũng như bao phủ các trường hợp không đầy đủ nhất.	Mức độ đầy đủ của các trường hợp kiểm thử ở mức vừa phải và mức độ tốn thời gian là vừa phải.	Đầy đủ nhất và tốn nhiều thời gian nhất

Không thích hợp để kiểm tra các thuật toán trong hệ thống.	Không thích hợp để kiểm tra các thuật toán trong hệ thống.	Thích hợp để kiểm tra các thuật toán trong hệ thống.
Phương pháp này sẽ được hoàn thành bởi cơ chế phát hiện lỗi.	Các miền dữ liệu và các giới hạn có thể sẽ được test nếu các tester có kiến thức về nó.	Các miền dữ liệu và các giới hạn sẽ được test.

Bảng 1. 4. So sánh các loại kiểm thử

1.2.5. Kết luận

Kiểm thử không phải chỉ đơn thuần là một hoạt động riêng lẻ mà là một loạt các hoạt động liên quan và bổ sung cho nhau và phức tạp. Tuy nhiên, việc dựa theo 7 nguyên tắc trên sẽ giúp cho chúng ta có cái nhìn tổng quát hơn về kiểm thử cũng như giúp chúng ta đánh giá được tính hiệu quả của hoạt động kiểm thử được thực thi.

CHƯƠNG 2: THIẾT KẾ TEST-CASE

2.1. Các phương pháp thiết kế test-case

2.1.1. Kỹ thuật phân vùng tương đương (*Equivalence Partitioning*)

2.1.1.1. Ý tưởng

Phân vùng tương đương là phương pháp chia các điều kiện đầu vào thành những vùng tương đương nhau. Tất cả các giá trị trong một vùng tương đương sẽ cho một kết quả đầu ra giống nhau. Vì vậy chúng ta có thể test một giá trị đại diện trong vùng tương đương.

2.1.1.2. Các bước thực hiện

Thiết kế test case bằng kỹ thuật phân vùng tương đương tiến hành theo 2 bước:

- (1) Xác định các lớp tương đương
- (2) Xác định các ca kiểm thử

2.1.1.3. Ưu điểm

Vì mỗi vùng tương đương ta chỉ cần test trên các phần tử đại diện nên số lượng test case được giảm đi khá nhiều nhờ đó mà thời gian thực hiện test cũng giảm đáng kể.

2.1.1.4. Nhược điểm

Không phải với bất kỳ bài toán nào đều có thể áp dụng kỹ thuật này. Có thể bị lack lỗi ở biên nếu chỉ chọn giá trị ở khoảng giữa của miền tương đương.

Vì vậy việc kết hợp linh hoạt giữa kỹ thuật phân vùng tương đương và phân tích giá trị biên dưới đây sẽ mang lại hiệu quả cao hơn để vừa tối ưu số lượng test case và vẫn đảm bảo được chất lượng phần mềm.

2.1.2. Kỹ thuật phân tích giá trị biên (*Boundary-value Analysis*)

2.1.2.1. Ý tưởng

Hầu hết các lỗi được tìm thấy khi kiểm tra ở các giá trị biên. Vì vậy phương pháp này tập trung vào việc kiểm thử các giá trị biên này.

Phân tích giá trị biên là trường hợp đặc biệt của phân vùng tương đương, dựa trên những phân vùng tương đương tester sẽ xác định giá trị biên giữa những phân vùng này và lựa chọn test case phù hợp.

2.1.2.2. Các bước thực hiện

Các case chuẩn được lựa chọn dựa vào quy tắc sau:

- Giá trị biên nhỏ nhất – 1
- Giá trị biên lớn nhất + 1

Nhưng nếu bạn muốn kiểm tra sâu hơn thì bạn cũng có thể lựa chọn theo quy tắc:

- Giá trị biên nhỏ nhất – 1
- Giá trị biên nhỏ nhất + 1
- Giá trị biên lớn nhất – 1
- Giá trị biên lớn nhất + 1

2.1.2.3. Ưu điểm

Thay vì phải test hết toàn bộ các giá trị trong từng vùng tương đương, kỹ thuật phân tích giá trị biên tập trung vào việc kiểm thử các giá trị biên của miền giá trị inputs để thiết kế test case do “lỗi thường tiềm ẩn tại các ngõ ngách và tập hợp tại biên”. Tiết kiệm thời gian thiết kế test case và thực hiện test.

2.1.2.4. Nhược điểm

Phương pháp này chỉ hiệu quả trong trường hợp các đối số đầu vào (input variables) độc lập với nhau và mỗi đối số đều có một miền giá trị hữu hạn.

2.1.3. Bảng quyết định – Decision Table Exercise

2.1.3.1. Ý tưởng

Sử dụng trong trường hợp: Hành động được lựa chọn phù hợp vào một lượng lớn các điều kiện

Bảng quyết định chia làm 2 phần:

- Điều kiện: diễn tả mọi điều kiện có thể có
- Hành động: là những hành động khác nhau có thể xảy ra tùy thuộc vào điều kiện

Số cột của bảng diễn tả mọi điều kiện bằng 2 lũy thừa của số điều kiện (Ví dụ có 3 điều kiện thì có $2 \text{ mũ } 3 = 8$ cột).

2.1.4. Dịch chuyển trạng thái

2.1.4.1. Ý tưởng

Kiểm thử chuyển đổi trạng thái được định nghĩa là kỹ thuật kiểm thử phần mềm trong đó thay đổi điều kiện đầu vào gây ra thay đổi trạng thái trong Ứng dụng được kiểm thử (Application under Test - AUT).

Đây là một kỹ thuật kiểm thử hộp đen trong đó tester phân tích cách xử lý của một ứng dụng được kiểm thử để đưa ra các điều kiện đầu vào khác nhau trong một trình tự. Trong kỹ thuật này, tester cung cấp cả giá trị kiểm thử đầu vào hợp lệ và không hợp lệ, sau đó xác định cách xử lý của hệ thống.

Kỹ thuật kiểm thử chuyển đổi trạng thái rất hữu ích khi bạn cần kiểm thử các cách chuyển đổi khác nhau trong hệ thống.

2.1.4.2. Ưu điểm

Kỹ thuật kiểm thử này sẽ cung cấp sự diễn tả bằng hình ảnh hoặc dạng bảng cách xử lý của hệ thống, điều này sẽ khiến tester bao quát và hiểu cách xử lý của hệ thống một cách hiệu quả.

Bằng cách sử dụng kiểm thử này, tester có thể xác minh rằng tất cả các điều kiện được bao phủ và kết quả được ghi lại

2.1.4.3. Nhược điểm

Nhược điểm chính của kỹ thuật kiểm thử này là chúng ta không thể sử dụng kỹ thuật này trong mọi lúc. Ví dụ, nếu hệ thống không theo thứ tự tuần tự, kỹ thuật này không thể được sử dụng.

Một nhược điểm khác là bạn phải xác định tất cả các trạng thái có thể có của một hệ thống. Mặc dù điều này hoàn toàn đúng đối với các hệ thống nhỏ, nhưng nó

sẽ bị phá vỡ tuần tự ở các hệ thống lớn hơn vì có sự tiến triển theo cấp số nhân trong số lượng các trạng thái.

2.2. Mô tả chương trình

Ngày nay, áp lực cuộc sống ngày một lớn hơn do sự vận động và phát triển không ngừng của khoa học kỹ thuật, sự cạnh tranh ngày càng khốc liệt và đòi hỏi nhiều sự nỗ lực và phấn đấu hơn. Như một sự cân bằng tất yếu, ngành dịch vụ giải trí, chăm sóc khách hàng, nghỉ dưỡng cũng phát triển chóng mặt. Nền công nghiệp phim ảnh là một phần không thể thiếu.

Các rạp chiếu phim ngày càng hiện đại hơn với đầy đủ những tiện nghi và dịch vụ đi kèm nhằm hài lòng cả những khách hàng khó tính nhất. Tuy nhiên, không phải ai cũng muốn đứng xếp hàng chờ đợi để lấy vé, không phải ai cũng thích đến những nơi công cộng với số lượng khách ra vào lớn do vấn đề vệ sinh, an ninh, yên tĩnh và quyền riêng tư.

Vì vậy, một giải pháp thay thế được đặt ra là xây dựng những nền tảng xem phim cá nhân, xem phim gia đình vừa có thể đảm bảo sự riêng tư, an toàn. Hơn nữa, việc cá nhân hóa trải nghiệm xem phim còn giúp người dùng thoải mái chọn lựa phim mình yêu thích trong mọi khung giờ rảnh rỗi, người dùng còn có một dữ liệu phim khổng lồ gồm nhiều thể loại phim khác nhau trong khi rạp chiếu phim chỉ công chiếu những phim mới nhất trong thời gian cố định.

Đặc biệt, về mặt chi phí, xem phim online giúp người dùng có thể tiết kiệm rất nhiều so với trải nghiệm đến rạp chiếu phim vì chi phí đăng ký thành viên xem phim online một tháng với vô vàn tiện ích kể trên chỉ có chi phí tương đương một vé xem trực tiếp ở rạp.

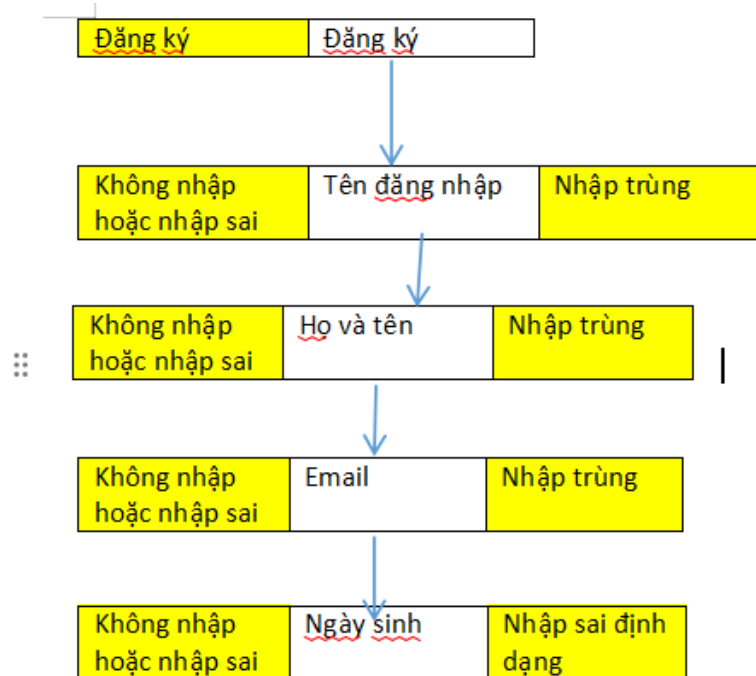
Tình trạng website xem phim ở Việt Nam rất nhiều nhưng lại là những trang web xem phim vi phạm bản quyền và chèn rất nhiều quảng cáo cùng chất lượng phim thấp làm trải nghiệm xem phim người dùng rất tệ và cũng làm mất đi giá trị, bản sắc và những ý nghĩa mà một bộ phim gửi gắm. Do đó, em đã quyết định thực hiện xây dựng trang website xem phim có bản quyền, cung cấp cho khách hàng kho phim khổng lồ với chi phí phải chăng cùng chất lượng cao.

CHƯƠNG 3: TRIỂN KHAI KIỂM THỬ

3.1. Kiểm thử hộp đen

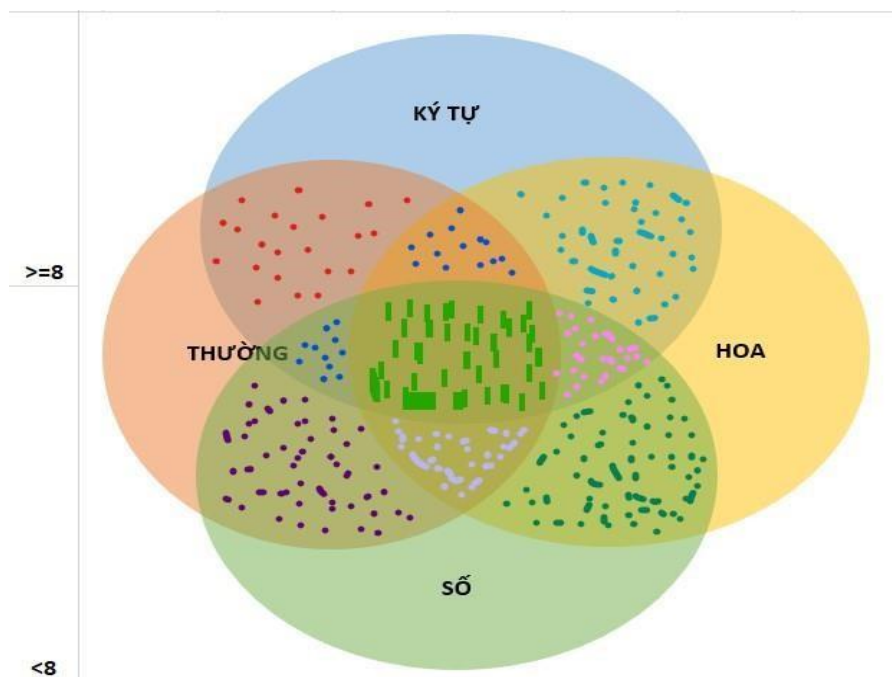
❖ Kiểm thử vùng tương đương

Xây dựng vùng tương đương và phân tích biên cho chức năng **đăng kí**.



Hình 3. 1. Vùng tương đương

Password:

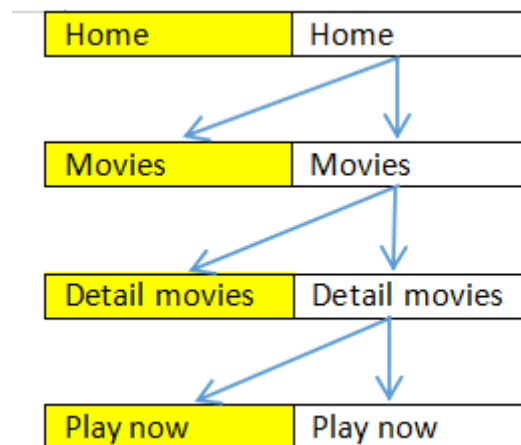


Hình 3. 2. Phân tích biên

Test case cho password có thể sử dụng được:

- Kí tự, Thường , Hoa, Số.
- Kí tự + thường, kí tự + hoa, kí tự + số, thường + hoa, thường + số, hoa + số.
- Kí tự + thường + hoa, kí tự + thường + số, thường + hoa + số.
- Kí tự + thường + hoa + số.

Xây dựng vùng tương đương và phân tích biên cho chức năng Xem Phim.

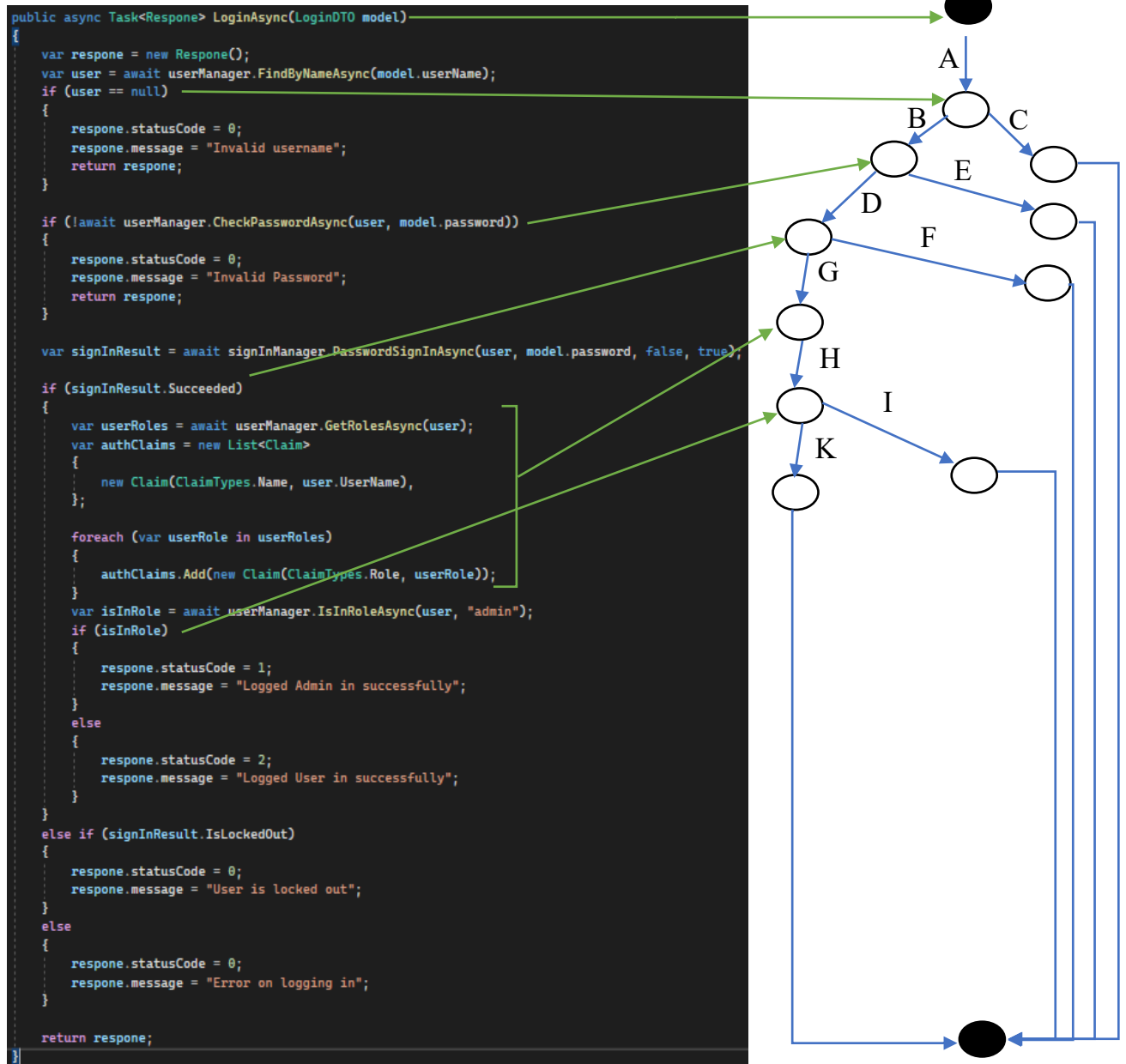


Hình 3. 3. Xây dựng vùng tương đương và phân tích biên

3.2. Kiểm thử hộp trắng

3.2.1. Kiểm thử đường dẫn cơ sở

Chức năng Login



Hình 3. 4. Code chức năng đăng nhập

- Xác định độ phức tạp Cyclomatic
 - Số lượng các vùng của G: 5
 - $V(G) = E - N + 2 = 15 - 12 + 2 = 5$
 - $V(G) = P + 1 = 4 + 1 = 5$
- Xác định tập đường dẫn cơ sở
 - A,C
 - A,B,E
 - A,B,D,F
 - A,B,D,G,H,K

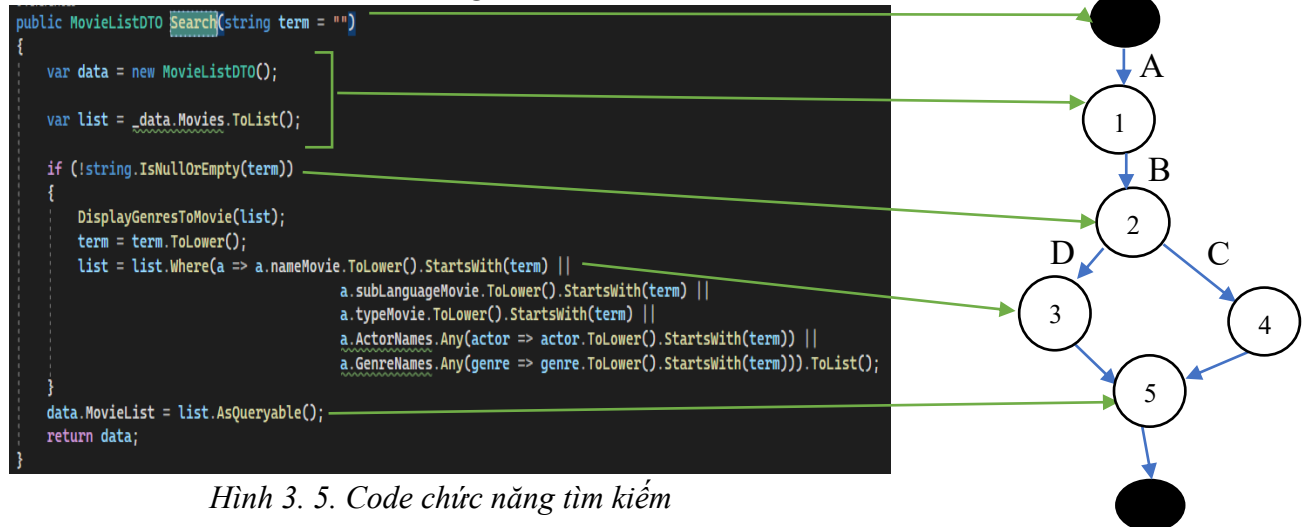
- A,B,D,G,H,I

Đường dẫn	Đầu vào	Đầu ra mong muốn
A,C	Username: cxcvxcv Password:****	Tài khoản không tồn tại
A,B,E	Username: admin Password:****	Mật khẩu không đúng
A,B,D,F	Username: addmin Password:****	Tài khoản đang bị khóa
A,B,D,G,H,K	Username: admin Password:****	Đăng nhập thành công với role Admin
A,B,D,G,H,I	Username: customer Password:*****	Đăng nhập thành công với role User

Bảng 3. 1. Tesst case đăng nhập

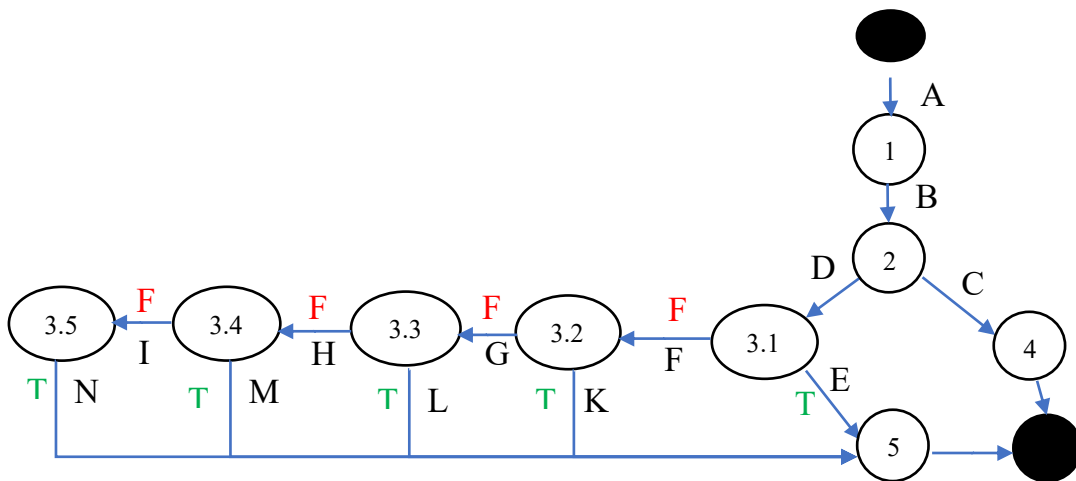
3.2.2. Kiểm thử bao phủ

Phủ nhánh và điều kiện chức năng tìm kiếm:



Hình 3. 5. Code chức năng tìm kiếm

- Chú thích
 - 3.1: a.nameMovie.ToLower().StartsWith(term)
 - 3.2: a.subLanguageMovie.ToLower().StartsWith(term)
 - 3.3: a.typeMovie.ToLower().StartsWith(term)
 - 3.4: a.ActorNames.Any(actor => actor.ToLower().StartsWith(term))
 - 3.5: a.GenreNames.Any(genre=> genre.ToLower().StartsWith(term))).ToList()



- Độ phức tạp Cyclomatic:
 - Số lượng các vùng của G: 6
 - $V(G) = E - N + 2 = 15 - 11 + 2 = 6$
 - $V(G) = P + 1 = 5 + 1 = 6$
- Các đường dẫn sau đảm bảo phủ nhánh và điều kiện:
 - A,B,C
 - A,B,D,E
 - A,B,D,F,K
 - A,B,D,F,G,L
 - A,B,D,F,G,H,M
 - A,B,D,F,G,H,I,N

Đường dẫn	Đầu vào	Đầu ra mong muốn
A,B,C	NULL	Tất cả phim
A,B,D,E	Tên phim	Tất cả phim có chứa tên phim
A,B,D,F,K	Đất nước	Tất cả phim của đất nước đó
A,B,D,F,G,L	Phim bộ/phim lẻ	Tất cả phim bộ/phim lẻ
A,B,D,F,G,H,M	Tên diễn viên	Tất cả phim có diễn viên đó
A,B,D,F,G,H,I,N	Thẻ loại	Tất cả phim thuộc thẻ loại đó

Bảng 3. 2. Test case tìm kiếm

CHƯƠNG 4: KIỂM THỬ TỰ ĐỘNG

4.1. Áp dụng công cụ kiểm thử Katalon

4.1.1. Giới thiệu



Hình 4. 1. Katalon Studio

Katalon Studio là một bộ công cụ toàn diện cho kiểm thử tự động hóa ứng dụng trên web và điện thoại di động. Công cụ này bao gồm một gói đầy đủ các tính năng mạnh mẽ giúp vượt qua những thách thức phổ biến trong tự động hóa thử nghiệm giao diện web, ví dụ như: pop-up, iFrame và wait-time. Giải pháp thân thiện và linh hoạt này giúp tester thực hiện công tác kiểm tra tốt hơn, làm việc nhanh hơn và khởi chạy phần mềm chất lượng cao nhờ vào sự thông minh mà nó cung cấp cho toàn bộ quá trình tự động hóa kiểm thử.

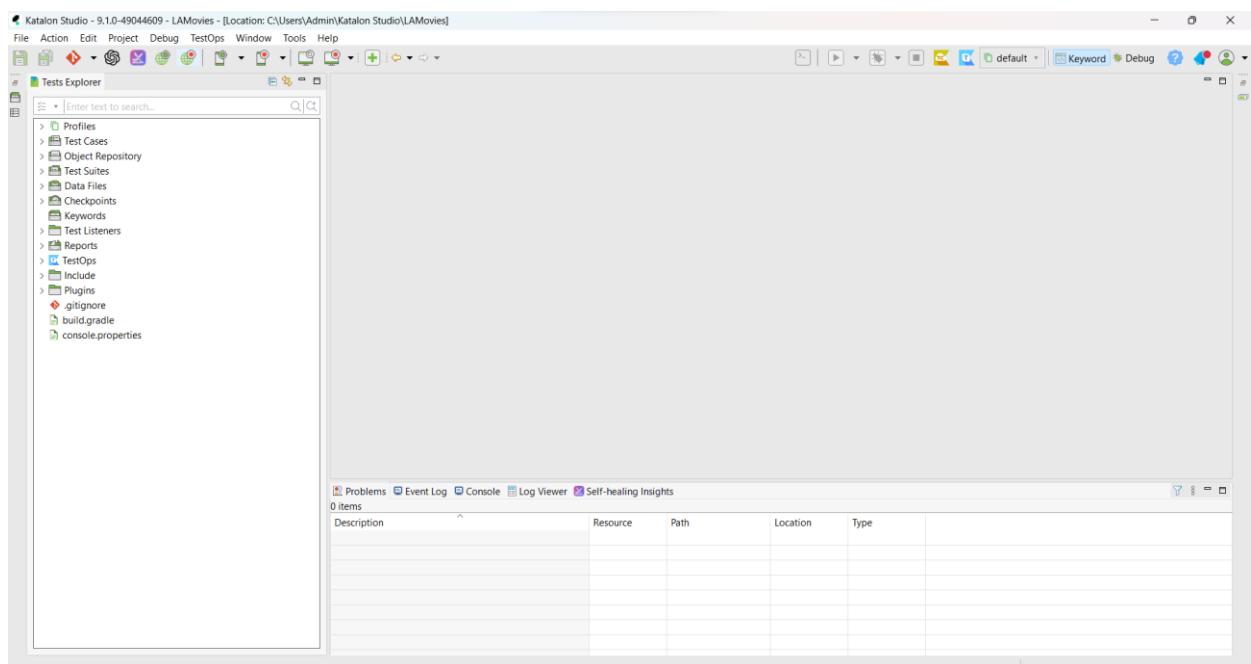
Katalon Studio được xây dựng từ Selenium/Appium.

4.1.2. Các tính năng chính

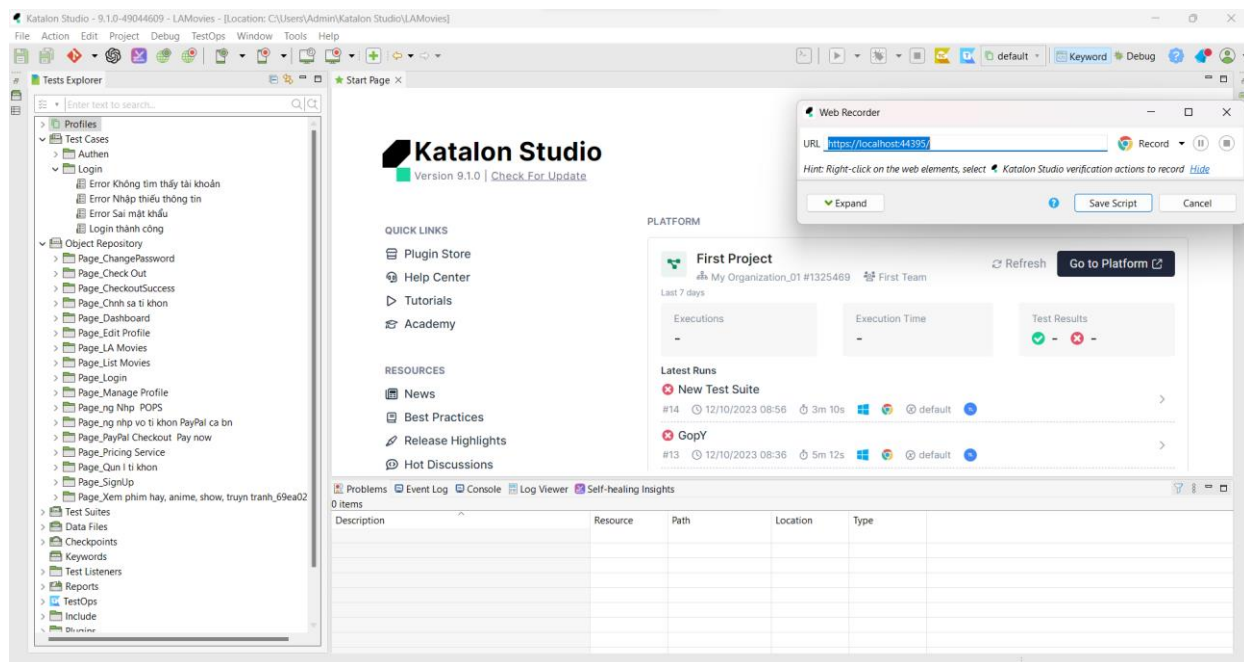
- Simple deployment: Một gói triển khai duy nhất, gắn kết chứa mọi thứ bạn cần để triển khai một công cụ kiểm tra tự động mạnh mẽ.
- Quick and easy set up: Không chỉ cung cấp sự cài đặt đơn giản, Katalon Studio cũng giúp bạn dễ dàng thiết lập môi trường. Tester có thể chạy test script đầu tiên của họ khá nhanh bằng cách sử dụng mẫu được xây dựng trước và test scripts, chẳng hạn như object repositories và keyword libraries.
- Faster and better results: Tích hợp sẵn mẫu với hướng dẫn rõ ràng giúp tester nhanh chóng xây dựng và chạy các test scripts tự động hóa. Họ có thể thực hiện từng bước với tốc độ và hiệu quả, từ thiết lập dự án, tạo ra thử nghiệm, thực hiện, tạo báo cáo và bảo trì.

- Flexible modes: Một tester mới có thể sử dụng recording và keywords để xây dựng các bài kiểm tra tự động hóa, trong khi các chuyên gia kiểm tra có một IDE hoàn chỉnh để xây dựng các kịch bản nâng cao.
- Ease of use: Nó không thể được dễ dàng hơn, ngay cả hướng dẫn sử dụng với kinh nghiệm lập trình tối thiểu cũng có thể khai thác lợi ích của nó một cách dễ dàng.
- Cross-browser application: Hỗ trợ nhiều nền tảng (Windows 32 và 64 (7, 8 và 10) và OS X 10.5+).

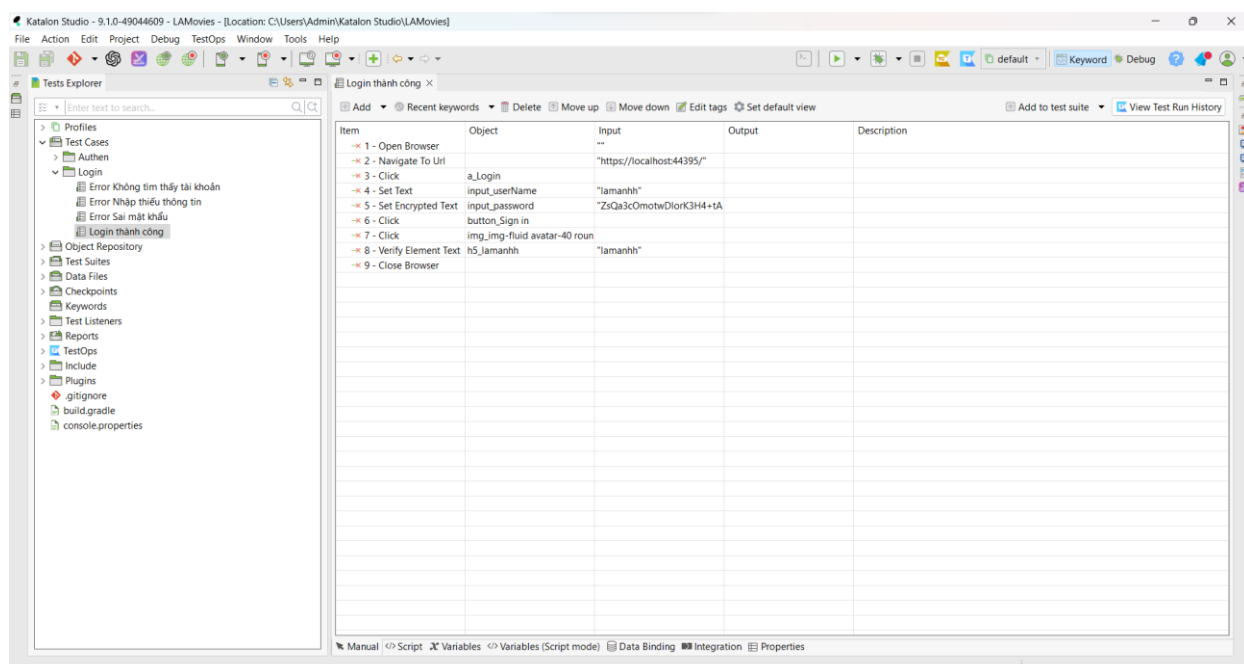
4.1.3. Triển khai kiểm thử



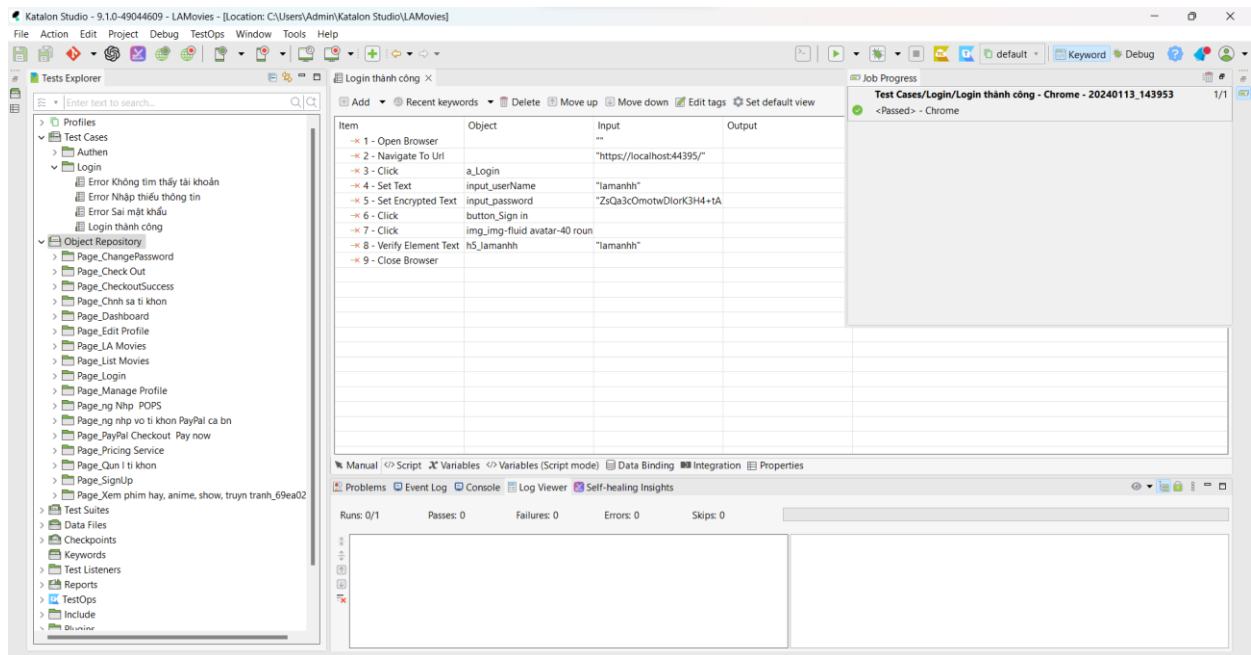
Hình 4. 2. Giao diện Katalon Studio



Hình 4. 3: Record test case



Hình 4. 4. Kịch bản test case



Hình 4. 5. Test thành công

4.1.4. Các test-case thực hiện

TC ID	TC Description	Automation	Author	Status
1	Test case chức năng đăng nhập thành công cho User 1. Mở trình duyệt 2. Truy cập đường dẫn: https://localhost:44395/ 3. Bấm vào nút đăng nhập 4. Nhập username 5. Nhập password 6. Bấm đăng nhập 7. Bấm vào avatar 8. Verify (xác nhận bằng username) 9. Đóng trình duyệt	<input checked="" type="checkbox"/>	Lâm Anh	Pass
2	Test case chức năng đăng nhập thành công cho Admin 1. Mở trình duyệt 2. Truy cập đường dẫn: https://localhost:44395/ 3. Bấm vào nút đăng nhập 4. Nhập username 5. Nhập password 6. Bấm đăng nhập 7. Verify (xác nhận bằng text Admin)	<input checked="" type="checkbox"/>	Lâm Anh	Pass

	8. Đóng trình duyệt			
3	Test case chức năng đăng nhập: Không tìm thấy tài khoản 1. Mở trình duyệt 2. Truy cập đường dẫn: https://localhost:44395/ 3. Bấm vào nút đăng nhập 4. Nhập username 5. Nhập password 6. Bấm đăng nhập 7. Verify (“Invalid username”) 8. Đóng trình duyệt	<input checked="" type="checkbox"/>	Lâm Anh	Pass
4	Test case chức năng đăng nhập: Sai mật khẩu 1. Mở trình duyệt 2. Truy cập đường dẫn: https://localhost:44395/ 3. Bấm vào nút đăng nhập 4. Nhập username 5. Nhập password 6. Bấm đăng nhập 7. Verify (“Invalid Password”) 8. Đóng trình duyệt	<input checked="" type="checkbox"/>	Lâm Anh	Pass
5	Test case chức năng đăng nhập: Tài khoản bị khoá 1. Mở trình duyệt 2. Truy cập đường dẫn: https://localhost:44395/ 3. Bấm vào nút đăng nhập 4. Nhập username 5. Nhập password 6. Bấm đăng nhập 7. Verify (“User is locked out”) 8. Đóng trình duyệt	<input checked="" type="checkbox"/>	Lâm Anh	Pass
6	Test case chức năng đăng ký thành công 1. Mở trình duyệt 2. Truy cập đường dẫn: https://localhost:44395/	<input checked="" type="checkbox"/>	Lâm Anh	Pass

	3. Bấm vào nút đăng nhập 4. Bấm vào nút đăng ký 5. Nhập username 6. Nhập password 7. Nhập lại password 8. Nhập email 9. Nhập fullname 10. Nhập ngày sinh 11. Bấm đăng ký 12. Bấm vào avatar 13. Verify (xác nhận bằng username) 14. Đóng trình duyệt			
7	Test case chức năng đăng ký: Tài khoản đã tồn tại 1. Mở trình duyệt 2. Truy cập đường dẫn: https://localhost:44395/ 3. Bấm vào nút đăng nhập 4. Bấm vào nút đăng ký 5. Nhập username 6. Nhập password 7. Nhập lại password 8. Nhập email 9. Nhập fullname 10. Nhập ngày sinh 11. Bấm đăng ký 12. Verify (text: “User already exist”) 13. Đóng trình duyệt	<input checked="" type="checkbox"/>	Lâm Anh	Pass
8	Test case chức năng đăng ký: Mật khẩu yếu 1. Mở trình duyệt 2. Truy cập đường dẫn: https://localhost:44395/ 3. Bấm vào nút đăng nhập 4. Bấm vào nút đăng ký 5. Nhập username 6. Nhập password 7. Nhập lại password 8. Nhập email 9. Nhập fullname 10. Nhập ngày sinh	<input checked="" type="checkbox"/>	Lâm Anh	Pass

	11. Bấm đăng ký 12. Verify (text: “Minimum length 6 and must contain 1 Uppercase,1 lowercase, 1 special character and 1 digit”) 13. Đóng trình duyệt			
9	Test case chức năng đăng ký: Mật khẩu không khớp 1. Mở trình duyệt 2. Truy cập đường dẫn: https://localhost:44395/ 3. Bấm vào nút đăng nhập 4. Bấm vào nút đăng ký 5. Nhập username 6. Nhập password 7. Nhập lại password 8. Nhập email 9. Nhập fullname 10. Nhập ngày sinh 11. Bấm đăng ký 12. Verify (text: “'passwordConfirm' and 'password' do not match.”) 13. Đóng trình duyệt	<input checked="" type="checkbox"/>	Lâm Anh	Pass
10	Test case chức năng đăng ký: Tài khoản chưa đủ 8 ký tự 1. Mở trình duyệt 2. Truy cập đường dẫn: https://localhost:44395/ 3. Bấm vào nút đăng nhập 4. Bấm vào nút đăng ký 5. Nhập username 6. Nhập password 7. Nhập lại password 8. Nhập email 9. Nhập fullname 10. Nhập ngày sinh 11. Bấm đăng ký 12. Verify (text: “Minimum length 8 for username”) 13. Đóng trình duyệt	<input checked="" type="checkbox"/>	Lâm Anh	Fail

11	<p>Test case chức năng đăng ký: Ngày sinh < ngày hiện tại</p> <ol style="list-style-type: none"> 1. Mở trình duyệt 2. Truy cập đường dẫn: https://localhost:44395/ 3. Bấm vào nút đăng nhập 4. Bấm vào nút đăng ký 5. Nhập username 6. Nhập password 7. Nhập lại password 8. Nhập email 9. Nhập fullname 10. Nhập ngày sinh 11. Bấm đăng ký 12. Verify (text: “Date Birthday < Date Now”) <p>Đóng trình duyệt</p>	<input checked="" type="checkbox"/>	Lâm Anh	Fail
12	<p>Test case chức năng đăng xuất</p> <ol style="list-style-type: none"> 1. Mở trình duyệt 2. Truy cập đường dẫn: https://localhost:44395/ 3. Bấm vào nút đăng nhập 4. Nhập username 5. Nhập password 6. Bấm đăng nhập 7. Bấm vào avatar 8. Bấm vào đăng xuất 9. Verify (button Login) 10. Đóng trình duyệt 	<input checked="" type="checkbox"/>	Lâm Anh	Pass
13	<p>Test case chức năng thanh toán paypal</p> <ol style="list-style-type: none"> 1. Mở trình duyệt 2. Truy cập đường dẫn: https://localhost:44395/ 3. Bấm vào nút đăng nhập 4. Nhập username 5. Nhập password 6. Bấm đăng nhập 7. Bấm vào chọn dịch vụ (Pricing) 8. Chọn gói cước đăng ký 9. Bấm thanh toán bằng Paypal 10. Nhập email 	<input checked="" type="checkbox"/>	Lâm Anh	Pass

	11. Bấm tiếp theo 12. Nhập password 13. Bấm đăng nhập 14. Bấm xác nhận thanh toán 15. Verify (xác nhận thanh toán thành công) 16. Đóng trình duyệt			
14	Test case chức năng thay đổi thông tin cá nhân thành công 1. Mở trình duyệt 2. Truy cập đường dẫn: https://localhost:44395/ 3. Bấm vào nút đăng nhập 4. Nhập username 5. Nhập password 6. Bấm đăng nhập 7. Bấm vào avatar 8. Bấm Edit Profile 9. Nhập fullname 10. Nhập email 11. Nhập ngày sinh 12. Bấm save 13. Verify (chuyển hướng đến trang quản lý thông tin) 14. Đóng trình duyệt	<input checked="" type="checkbox"/>	Lâm Anh	Pass
15	Test case chức năng đổi mật khẩu 1. Mở trình duyệt 2. Truy cập đường dẫn: https://localhost:44395/ 3. Bấm vào nút đăng nhập 4. Nhập username 5. Nhập password 6. Bấm đăng nhập 7. Bấm vào avatar 8. Bấm Change Password 9. Nhập mật khẩu cũ 10. Nhập mật khẩu mới 11. Nhập xác nhận mật khẩu 12. Bấm save	<input checked="" type="checkbox"/>	Lâm Anh	Pass

	13. Verify (thông báo thay đổi mật khẩu thành công) 14. Đóng trình duyệt			
16	Test case chức năng sửa thông tin tài khoản 1. Mở trình duyệt 2. Truy cập đường dẫn: https://localhost:44395/ 3. Bấm vào nút đăng nhập 4. Nhập username 5. Nhập password 6. Bấm đăng nhập 7. Bấm quản lý tài khoản 8. Bấm icon Edit 9. Nhập fullname 10. Nhập email 11. Nhập ngày sinh 12. Bấm nút lưu 13. Verify (Hiển thị kết quả) 14. Đóng trình duyệt	<input checked="" type="checkbox"/>	Lâm Anh	Pass
17	Test case chức năng xoá tài khoản 1. Mở trình duyệt 2. Truy cập đường dẫn: https://localhost:44395/ 3. Bấm vào nút đăng nhập 4. Nhập username 5. Nhập password 6. Bấm đăng nhập 7. Bấm quản lý tài khoản 8. Bấm icon Delete 9. Bấm oke (xác nhận delete) 10. Verify (Hiển thị kết quả) Đóng trình duyệt	<input checked="" type="checkbox"/>	Lâm Anh	Pass
18	Test case chức năng tìm kiếm: Rỗng 1. Mở trình duyệt 2. Truy cập đường dẫn: https://localhost:44395/ 3. Bấm nút Search 4. Nhập null 5. Enter	<input checked="" type="checkbox"/>	Lâm Anh	Pass

	6. Verify (Hiển thị kết quả) 7. Đóng trình duyệt			
19	Test case chức năng tìm kiếm phim theo tên phim 1. Mở trình duyệt 2. Truy cập đường dẫn: https://localhost:44395/ 3. Bấm nút Search 4. Nhập tên phim 5. Enter 6. Verify (Hiển thị kết quả) Đóng trình duyệt	<input checked="" type="checkbox"/>		Pass
20	Test case chức năng tìm kiếm phim theo quốc gia 1. Mở trình duyệt 2. Truy cập đường dẫn: https://localhost:44395/ 3. Bấm nút Search 4. Nhập tên quốc gia 5. Enter 6. Verify (Hiển thị kết quả) Đóng trình duyệt	<input checked="" type="checkbox"/>		Pass
21	Test case chức năng tìm kiếm phim theo thể loại 1. Mở trình duyệt 2. Truy cập đường dẫn: https://localhost:44395/ 3. Bấm nút Search 4. Nhập thể loại 5. Enter 6. Verify (Hiển thị kết quả) Đóng trình duyệt	<input checked="" type="checkbox"/>		Pass
22	Test case chức năng tìm kiếm phim theo diễn viên 1. Mở trình duyệt 2. Truy cập đường dẫn: https://localhost:44395/ 3. Bấm nút Search 4. Nhập tên diễn viên 5. Enter	<input checked="" type="checkbox"/>		Pass

	6. Verify (Hiển thị kết quả) Đóng trình duyệt			
23	Test case chức năng tìm kiếm phim theo loại phim 1. Mở trình duyệt 2. Truy cập đường dẫn: https://localhost:44395/ 3. Bấm nút Search 4. Nhập null 5. Enter 6. Verify (Hiển thị kết quả) Đóng trình duyệt	<input checked="" type="checkbox"/>		Pass

Bảng 4. 1. Test case Katalon Studio

- Trong 23 test case có 21 test case pass và 2 test case fail => 2 bug: Username bé hơn 8 ký tự, ngày sinh lớn hơn ngày hiện tại

4.2. Áp dụng công cụ kiểm thử Test-Complete

4.2.1. Giới thiệu



Hình 4. 6. TestComplete

- TestComplete là một môi trường kiểm thử tự động cho một loạt các loại ứng dụng và công nghệ, bao gồm (nhưng không giới hạn) Windows, .NET, WPF, Visual C++, Visual Basic, Delphi, C++ Builder, Java và các ứng dụng Web và dịch vụ.

- TestComplete được định hướng như nhau đối với chức năng kiểm thử, đơn vị. Nó cung cấp hỗ trợ cho các thử nghiệm hồi quy hàng ngày và hỗ trợ nhiều loại thử nghiệm: thử nghiệm dữ liệu điều khiển, kiểm thử đối tượng điều khiển, và những người khác.

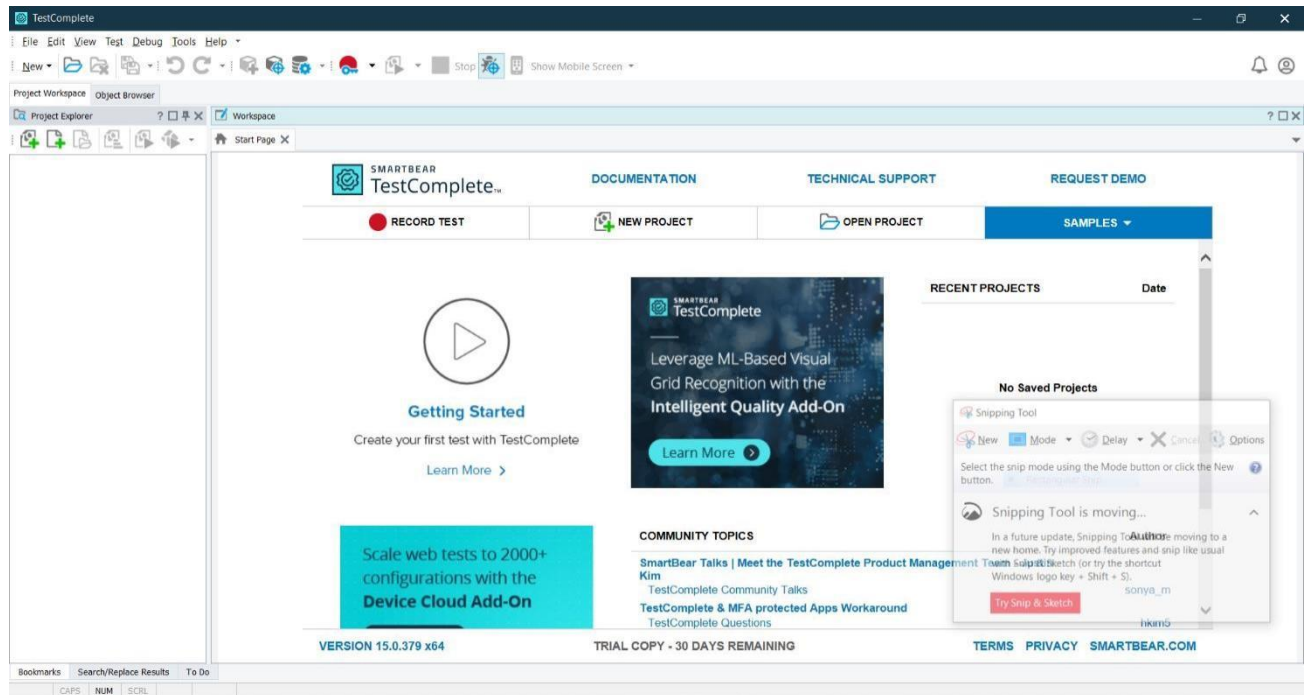
- Bạn tạo ra các bài kiểm thử bằng cách ghi lại chúng hoặc lệnh kiểm thử chỉnh sửa trong bảng và biên tập viên của TestComplete . Kiểm thử có thể được chạy từ bên trong TestComplete hoặc họ có thể được xuất khẩu sang một ứng dụng bên ngoài và chạy đó.

- TestComplete nhận đối tượng và điều khiển trong các ứng dụng thử nghiệm và cung cấp các lệnh đặc biệt để mô phỏng hành động sử dụng với họ. Nó cũng cung cấp các trạm kiểm soát cụ thể , cho phép bạn dễ dàng kiểm thử trạng thái ứng dụng trong thời gian chạy thử nghiệm.

4.2.2. Các tính năng chính

- Keyword Testing: Kiểm tra từ khóa
- Full-Featured Script Editor: Chỉnh sửa đầy đủ các kịch bản
- Test Record and Playback: Cho phép ghi và chạy lại quá trình test
- Script Debugging Features: Gỡ lỗi
- Access to Methods and Properties of Internal Objects : Truy cập đến các phương thức và thuộc tính của bên trong đối tượng
- Unicode Support: Hỗ trợ bộ gõ Unicode
- Issue-Tracking Support

4.2.3. Triển khai kiểm thử

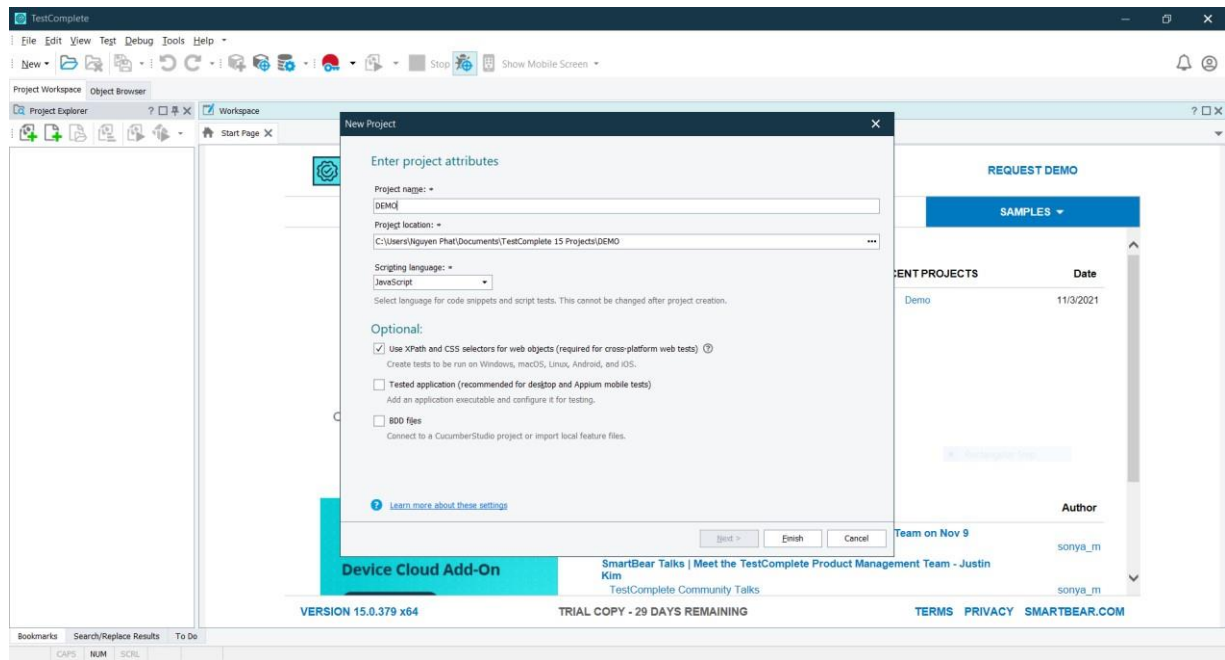


Hình 4. 7. Giao diện TestComplete

ÁP DỤNG KIỂM THỬ

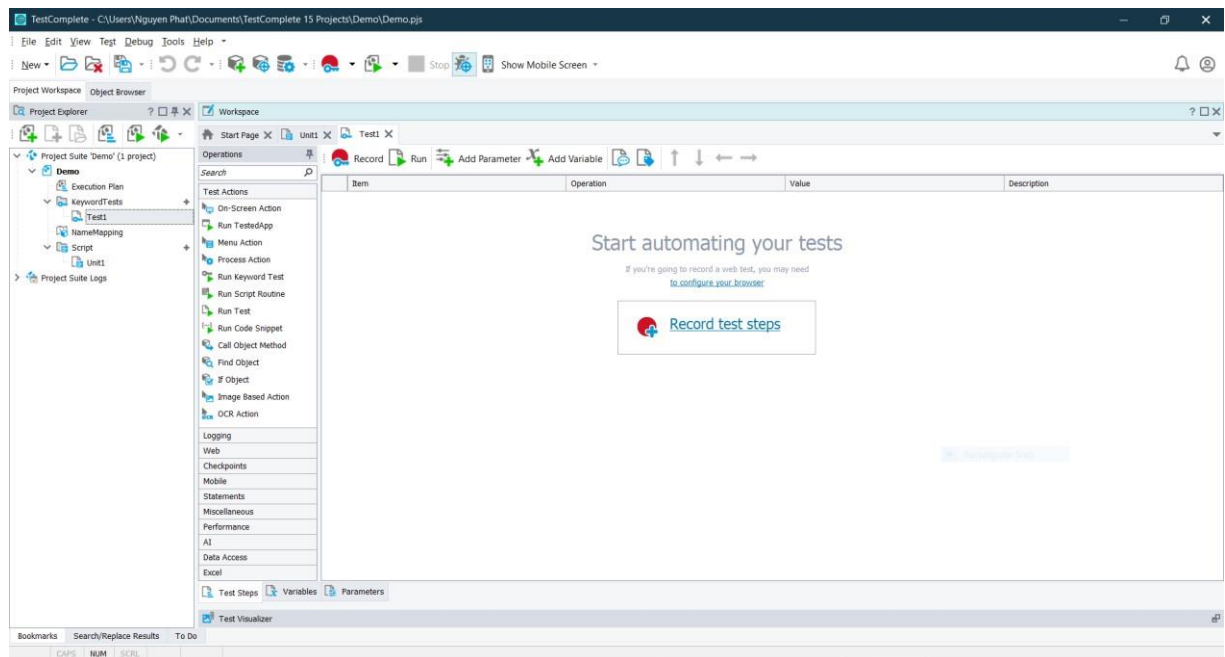
- Với công cụ kiểm thử TestComplete này nhóm em sẽ áp dụng kiểm thử chức năng thêm mới thể loại của Website xem phim online. Đến với công cụ kiểm thử TestComplete ta gồm các bước như sau:

B1: Từ màn hình giao diện chính ta ấn chọn File -> New -> New Project -> Đổi tên như mong muốn sau đó -> Finish



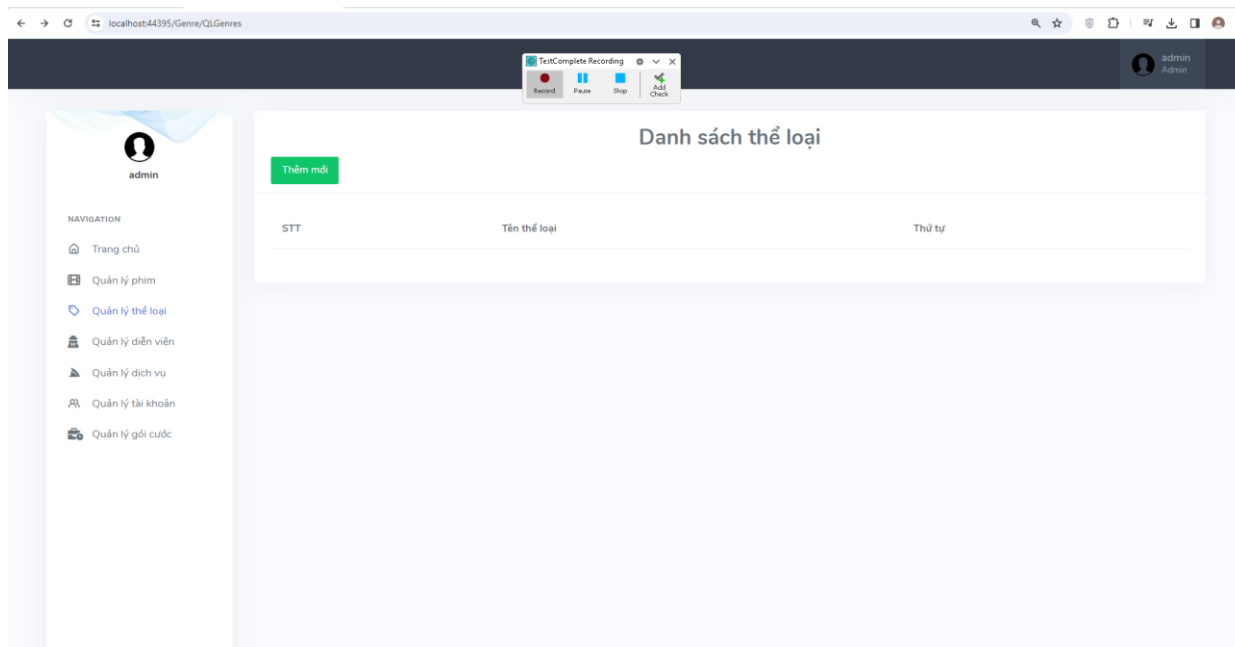
Hình 4. 8. Tạo project

B2: Sau đó ấn vào “Record test steps” chính giữa màn hình để ghi lại một bài Test trên TestComplete



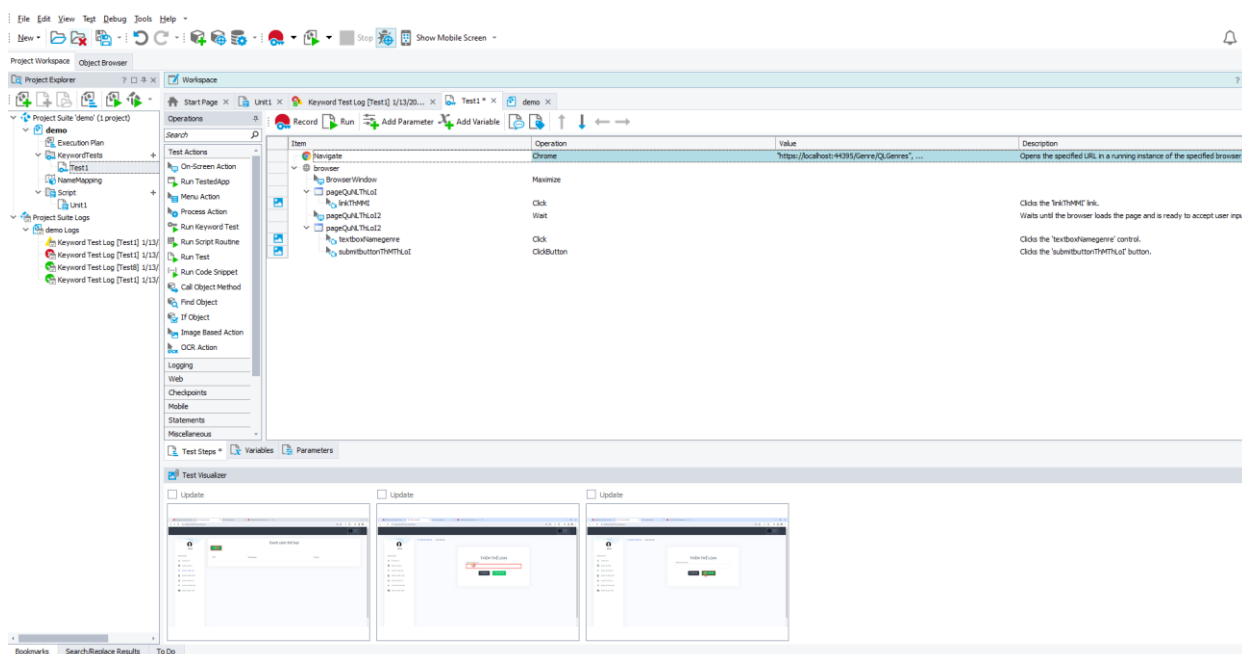
Hình 4. 9. Record test steps

B3: Sau khi kết thúc một bài test chọn vào item Stop trên thanh công cụ để kết thúc quá trình ghi lại bài test.



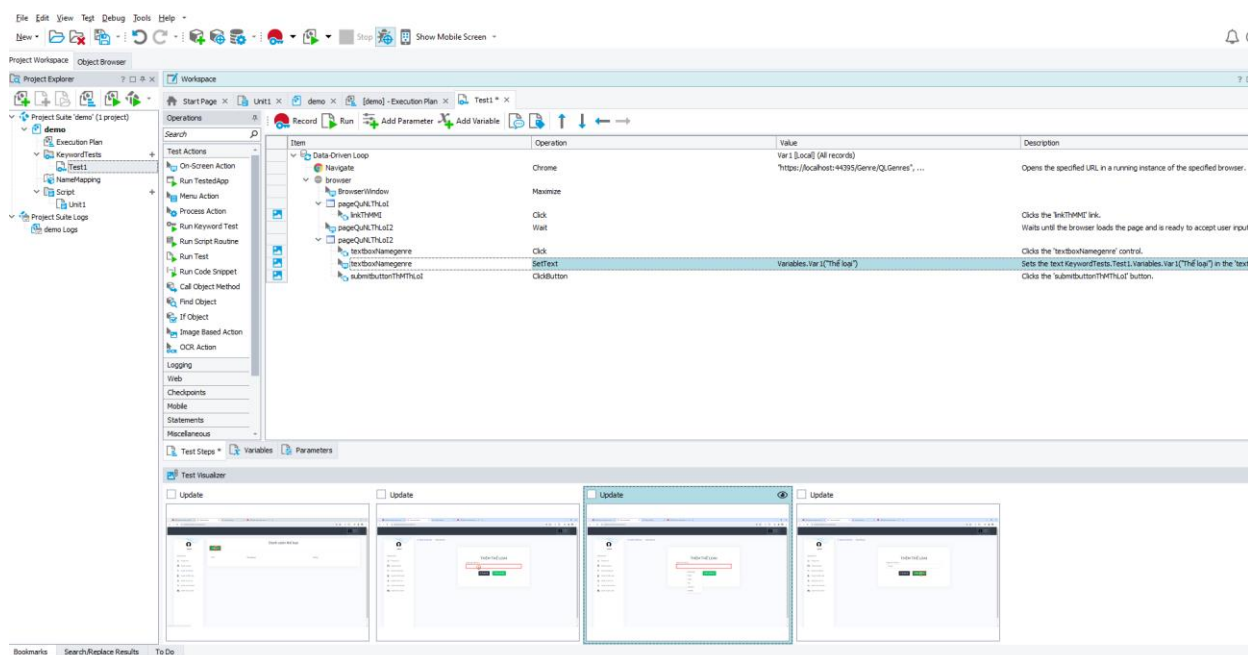
Hình 4. 10. Bắt đầu record

Đây là một ví dụ mà kết quả hiển thị sau khi Record

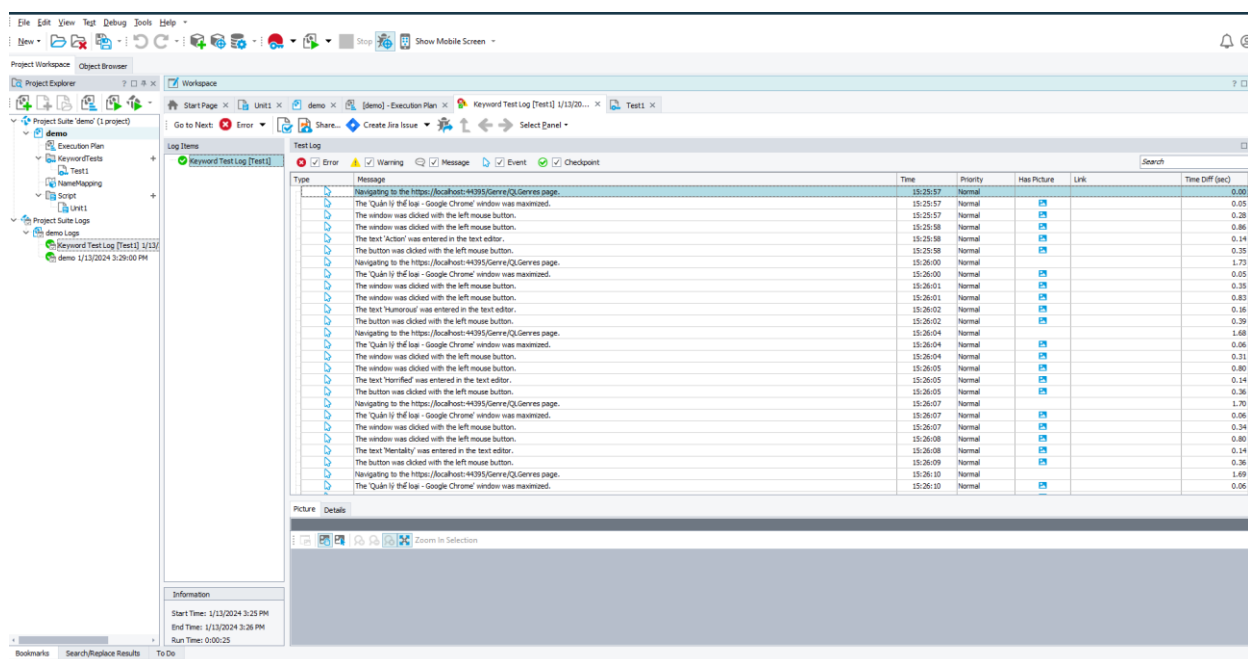


Hình 4. 11. Áp dụng kiểm thử

B4: Thêm data vào testcase để tự động kiểm thử chức năng thêm sử dụng data Loop

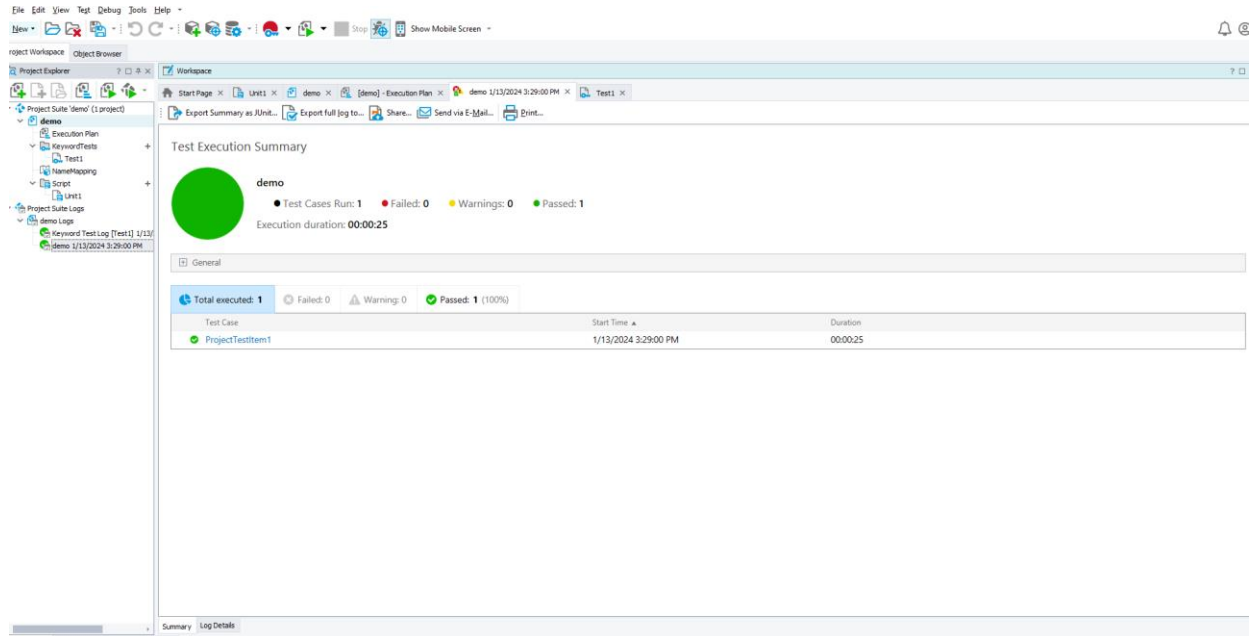


Hình 4. 12. Áp dụng kiểm thử



Hình 4. 13. Áp dụng kiểm thử

Quá trình kiểm tra thành công và không xảy ra lỗi



Hình 4. 14. Kiểm thử thành công

4.3. Áp dụng công cụ kiểm thử Selenium

4.3.1. Giới thiệu



Hình 4. 15. Selenium

Selenium là bộ kiểm thử tự động miễn phí (mã nguồn mở) dành cho các ứng dụng web trên các trình duyệt và nền tảng khác nhau. Nó khá là giống với HP Quick Test Pro (QTP bây giờ là UFT) chỉ khác là Selenium thì tập trung vào việc tự động hoá các ứng dụng dựa trên nền tảng web. Kiểm thử được thực hiện bằng

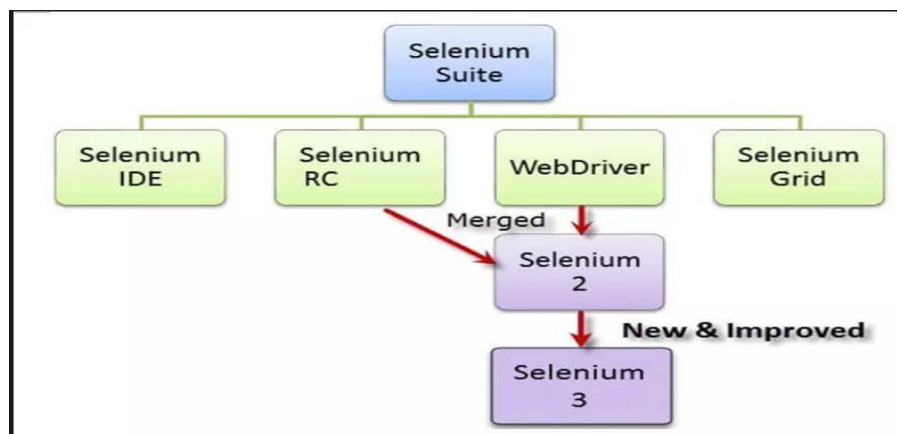
cách sử dụng công cụ Selenium thường được gọi là Kiểm thử Selenium. Selenium không chỉ là 1 công cụ độc lập mà là 1 bộ công cụ của phần mềm, mỗi bộ đều đáp ứng được nhu cầu kiểm thử khác nhau của 1 tổ chức. Nó có 4 thành phần.

Selenium Integrated Development Environment (IDE)

Selenium Remote Control (RC)

- WebDriver
- Selenium Grid

Hiện tại, Selenium RC và WebDriver được hợp nhất thành một framework duy nhất để tạo ra Selenium 2. Còn Selenium 1 thì tham chiếu đến Selenium RC.



Hình 4. 16. Selenium

4.3.2. Các chức năng chính:

- **New Test Case:** Điều này được sử dụng để tạo ra một Test Case mới được sử dụng để mở và load một Test Case đã được lưu từ trước.
- **Save Test Case:** Điều này cho phép lưu Testcase hiện tại.
- **Save Test Case As:** Điều này cho phép lưu các Test Case hiện tại, chỉ định một tên cho tập tin.

- **Export Test Case As:** Điều này cho phép xuất các Testcase trong một loạt các ngôn ngữ (Ruby / Python / Java / C #) mà trong cả Selenium Remote Control và Selenium WebDriver Format
- **Recent Test Cases:** Danh sách các testcase lưu cuối cùng (Nhấp vào bất kỳ testcase hiển thị sẽ tải các testcase đó).
- **Add Test Case:** Tìm kiếm và tải một testcase vào các testcase đang mở, để hợp nhất chúng.
- **Properties:** cung cấp thuộc tính của các testcase hiện tại mà người dùng đang làm việc.
- **New Test Suite:** Điều này được sử dụng để tạo ra một Test Suite mới
- **Open Test Suite:** Điều này được sử dụng để mở và load 1 Test Suite đã được lưu từ trước.
- **Save Test Suite:** Điều này cho phép lưu các Test Suite hiện hành.
- **Save Test Suite As:** Điều này cho phép lưu các Test Suite hiện tại, chỉ định một tên cho tập tin.
- **Export Test Suite As:** Điều này cho phép xuất khẩu các Test Suite vào một loạt các ngôn ngữ (Ruby / Python / Java / C #) mà là trong cả hai Selenium Remote Control và Format Selenium WebDriver.
- **Recent Test Suites:** Danh sách Test Suites lưu lại lần gần đây nhất

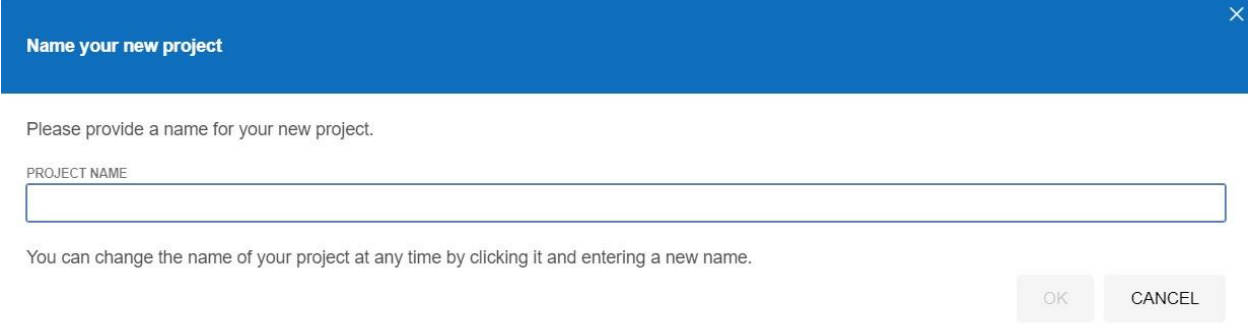
4.3.3. Triển khai kiểm thử:

4.3.3.1. Áp dụng kiểm thử chức năng đăng nhập với công cụ Selenium.



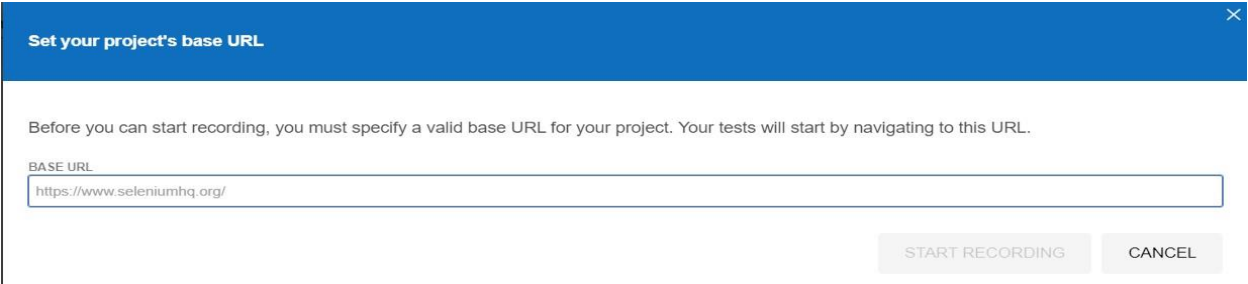
Hình 4. 17. Tạo project

- Đầu tiên khi mở công cụ selenium ta chọn vào New project.



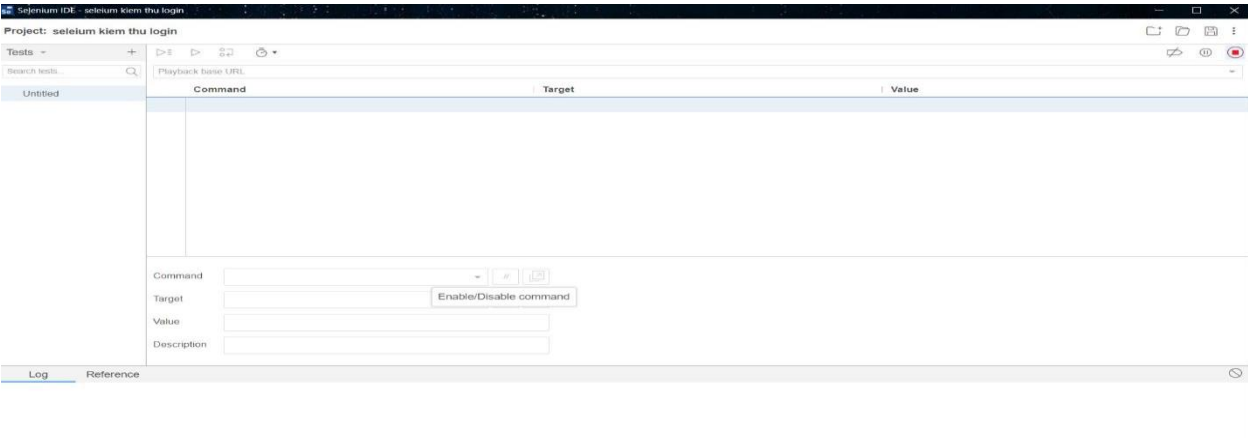
Hình 4. 18. Đặt tên Project

- Tiếp theo chúng ta đặt tên cho project.

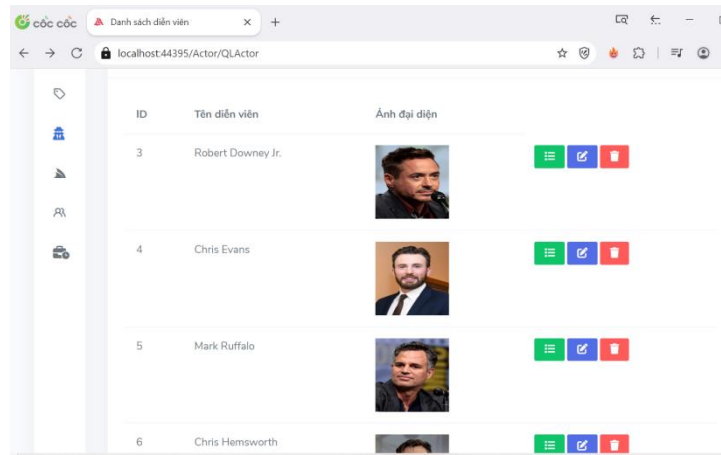


Hình 4. 19. Gắn link web tiến hành test

- Chúng ta gắn đường link trang web muốn kiểm thử.

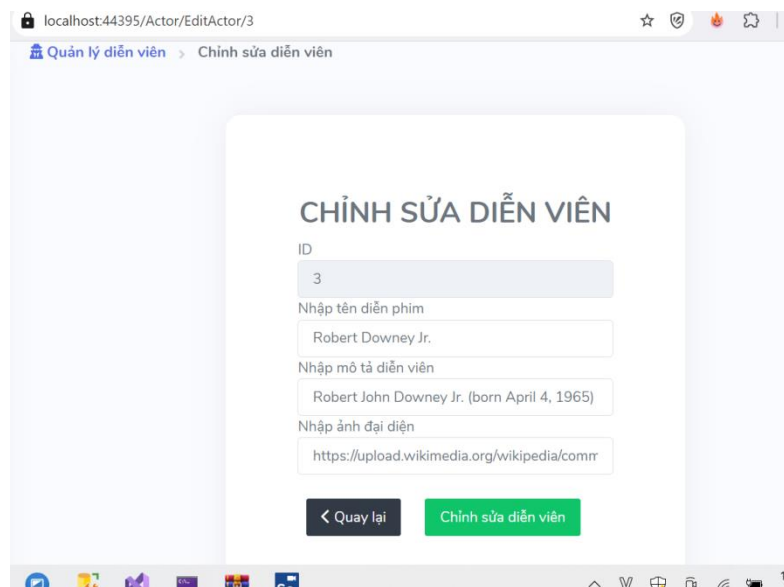


Hình 4. 20. Giao diện Selenium



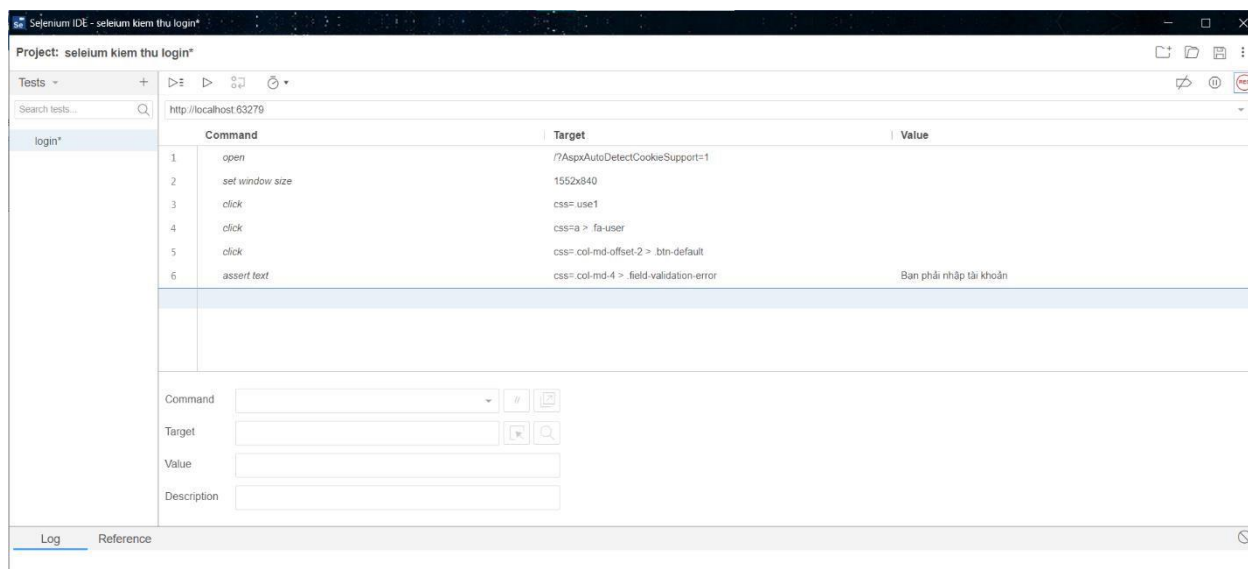
Hình 4. 21. Giao diện web bắt đầu ghi test

- Trang web sau khi mở bằng selenium, và bắt đầu ghi test.



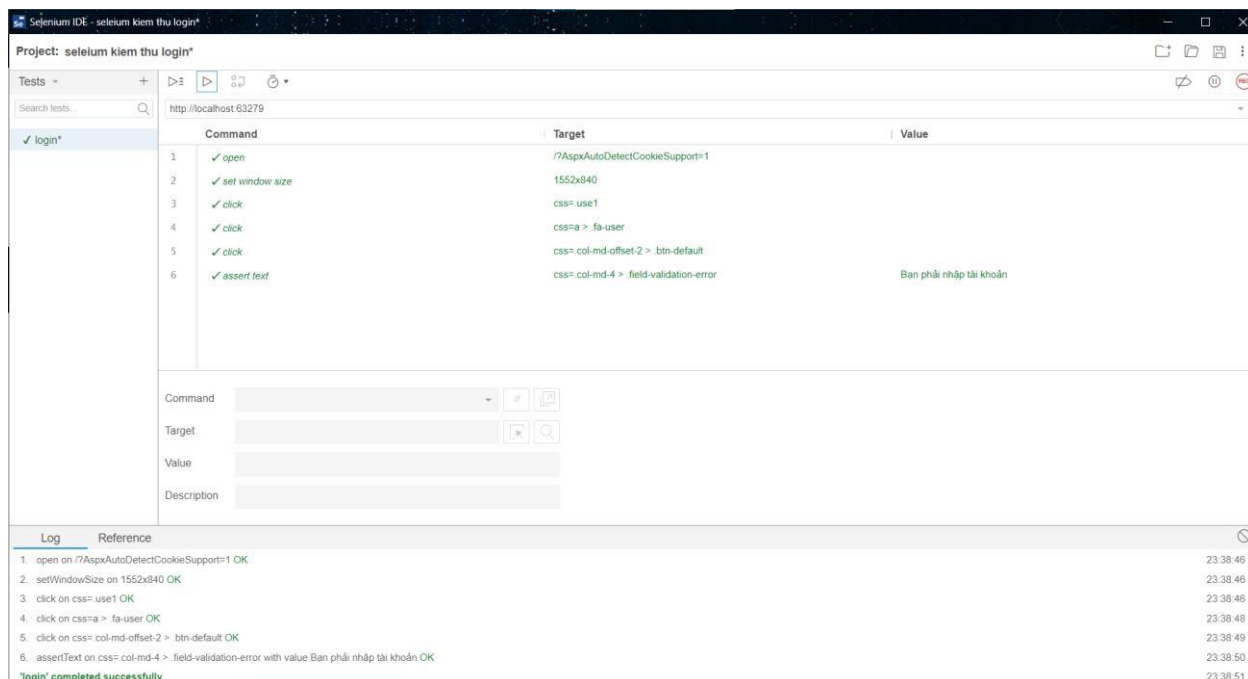
Hình 4. 22. Thực hiện quá trình kiểm thử

- Sau khi thao tác chỉnh sửa diễn viên xong



Hình 4. 23. Chạy kết quả test

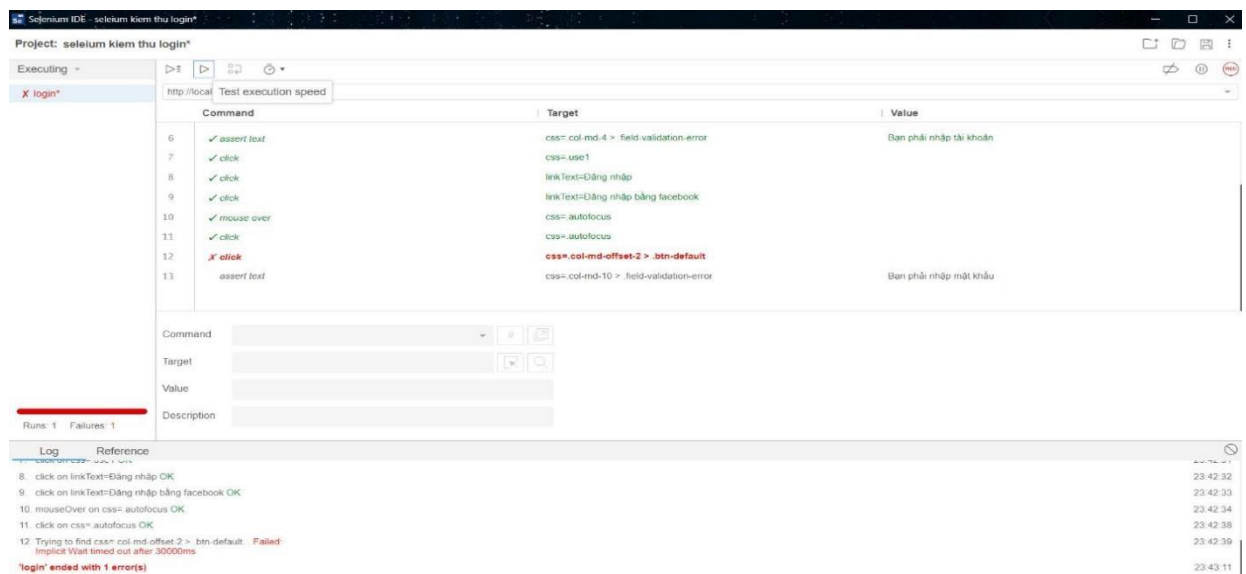
- Ta vào lại Selenium và ấn stop(Ctrl+U) và đặt tên cho new test.



Hình 4. 24. Cho kết quả test

- Sau khi Run test thì Selenium sẽ chạy lại quá trình thao tác đăng nhập một lần nữa vào cho ra kết quả test.

- Theo kết quả trên thì quá trình test không xảy ra lỗi.



Hình 4. 25. Cho kết quả test

- Đây là 1 ví dụ khi Selenium test phát hiện ra lỗi.

4.3.3.2. Áp dụng kiểm thử chức năng thêm diễn viên với công cụ Selenium.

- Chúng ta thực hiện các thao tác thêm diễn viên.

THÊM DIỄN VIÊN

Nhập tên diễn phim

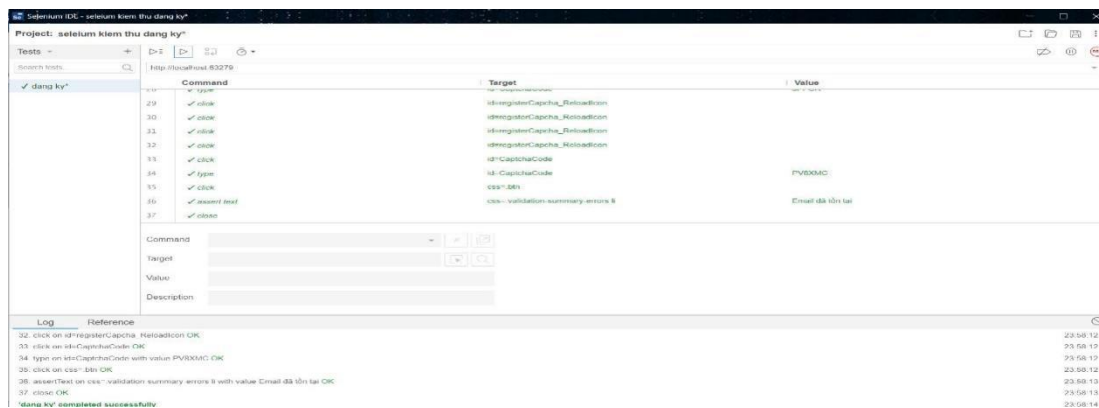
Nhập mô tả diễn viên

Nhập ảnh đại diện

← Quay lại
Thêm diễn viên

Hình 4. 26. Thực hiện quá trình test

- Khi chúng ta thêm diễn viên thành công



Hình 4. 27. Cho kết quả test

- Quay về Selenium Run test thì không phát hiện ra lỗi.

THÊM DIỄN VIÊN

Nhập tên diễn phim

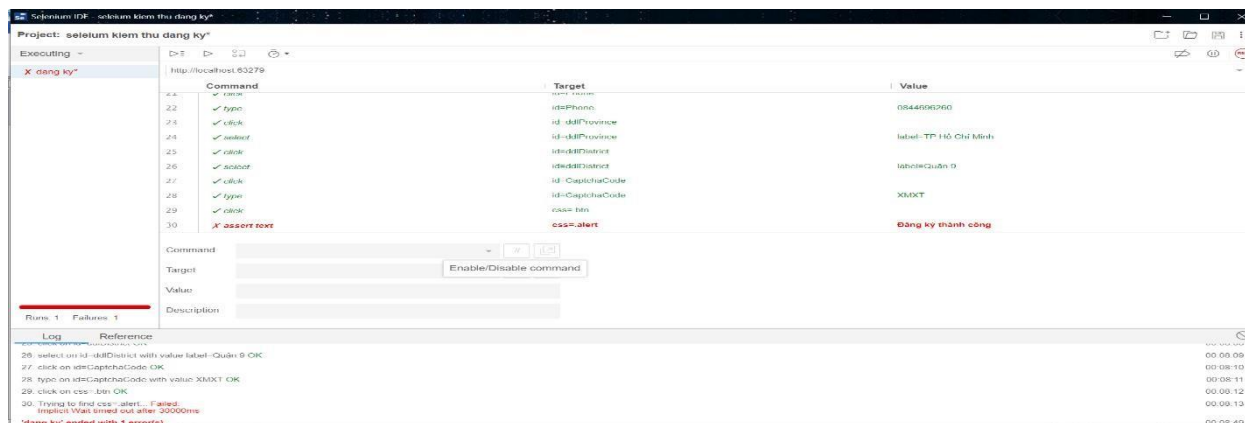
Nhập mô tả diễn viên

Nhập ảnh đại diện

← Quay lại
Thêm diễn viên

Hình 4. 28. Thực hiện quá trình test

- Khi chúng ta thêm diễn viên thành công.



Hình 4. 29. Cho kết quả test

- Lần này phát hiện lỗi là do khi chúng ta chạy test thì Selenium sẽ thực hiện lại quá trình đổi tên diễn viên lần nữa nên có thể xảy ra lỗi đó là trùng tên diễn viên.

4.4. Công cụ kiểm thử tự động Telerik

4.4.1. Giới thiệu



Hình 4. 30. Telerik Test Studio

Telerik Test Studio là một công cụ tự động hóa kiểm thử phần mềm được phát triển bởi Telerik, giúp các nhà phát triển và nhóm kiểm thử xây dựng và thực hiện các bộ kiểm thử tự động cho ứng dụng web và desktop. Được thiết kế để tối ưu hóa quá trình kiểm thử và giảm thiểu công sức cần thiết để đảm bảo chất lượng phần mềm, Telerik Test Studio cung cấp một loạt các tính năng và công cụ linh hoạt.

4.4.2. Các tính năng chính

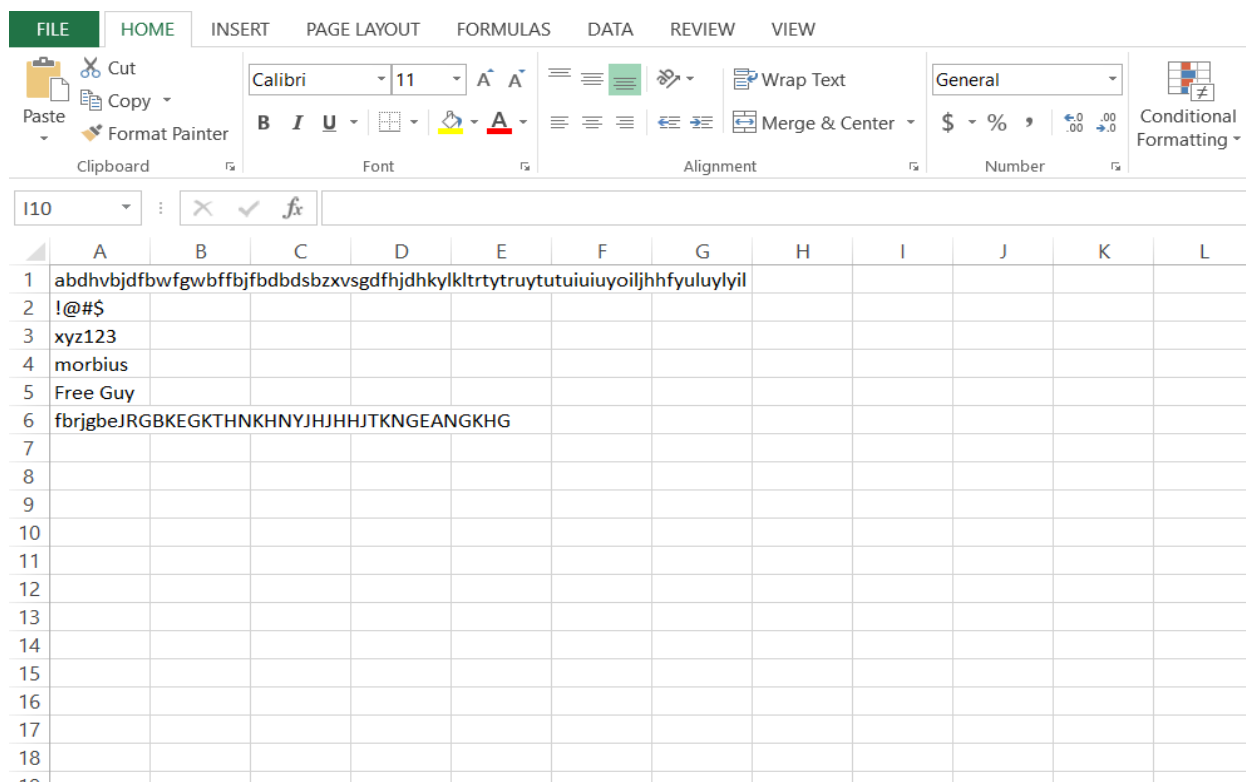
- Giao diện đồ họa thân thiện
- Hỗ trợ nhiều loại ứng dụng
- Ghi và chạy kiểm thử dễ dàng
- Kiểm thử chức năng và hiệu suất
- Tích hợp với các công cụ phần mềm phổ biến
- Hỗ trợ đa nền tảng và đa trình duyệt

4.4.3. Triển khai kiểm thử

ÁP DỤNG KIỂM THỬ

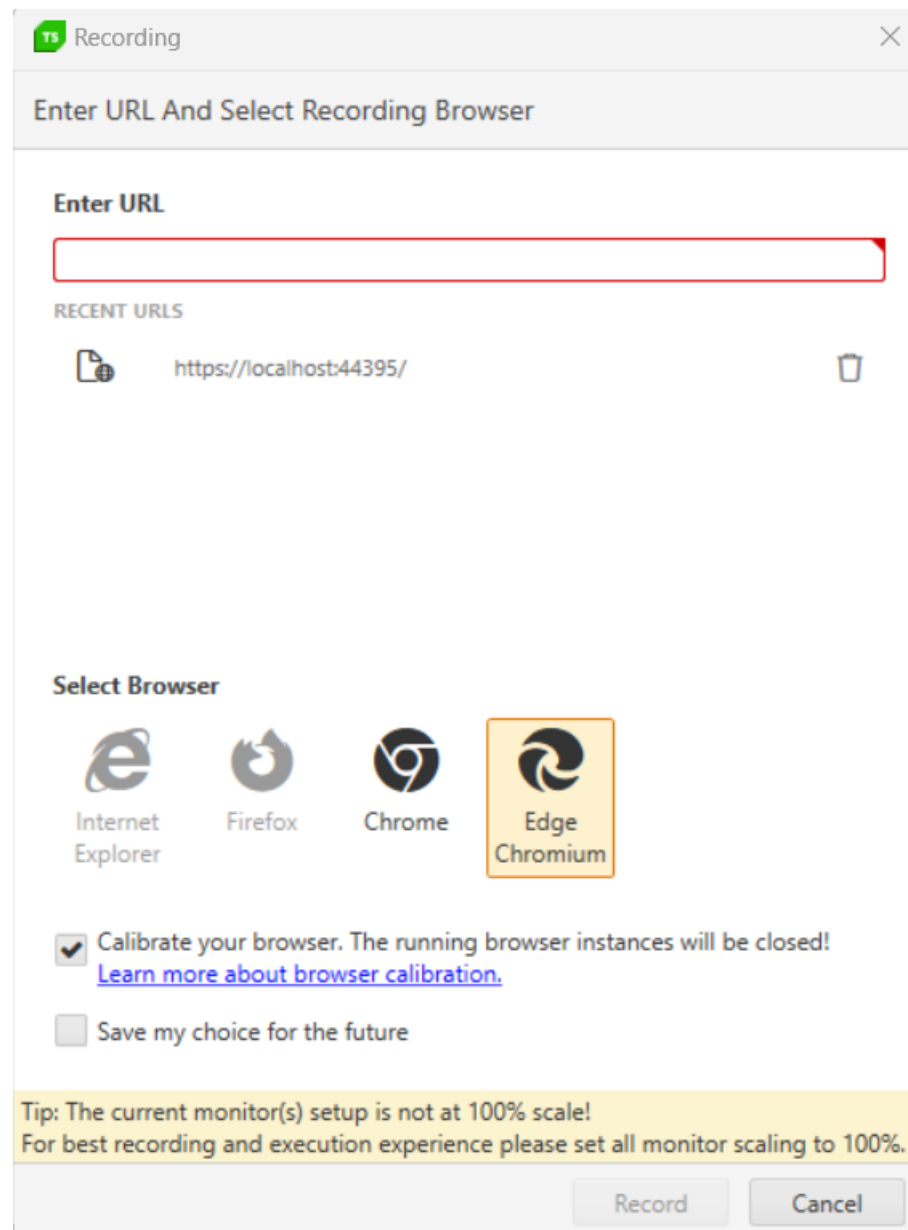
- Nhóm em sẽ áp dụng kiểm thử chức năng Tìm kiếm của Website xem phim bằng công cụ kiểm thử Telerik. Đến với công cụ kiểm thử Telerik ta gồm các bước như sau:

B1: Ta xuất file CSV các testcase để kiểm thử.

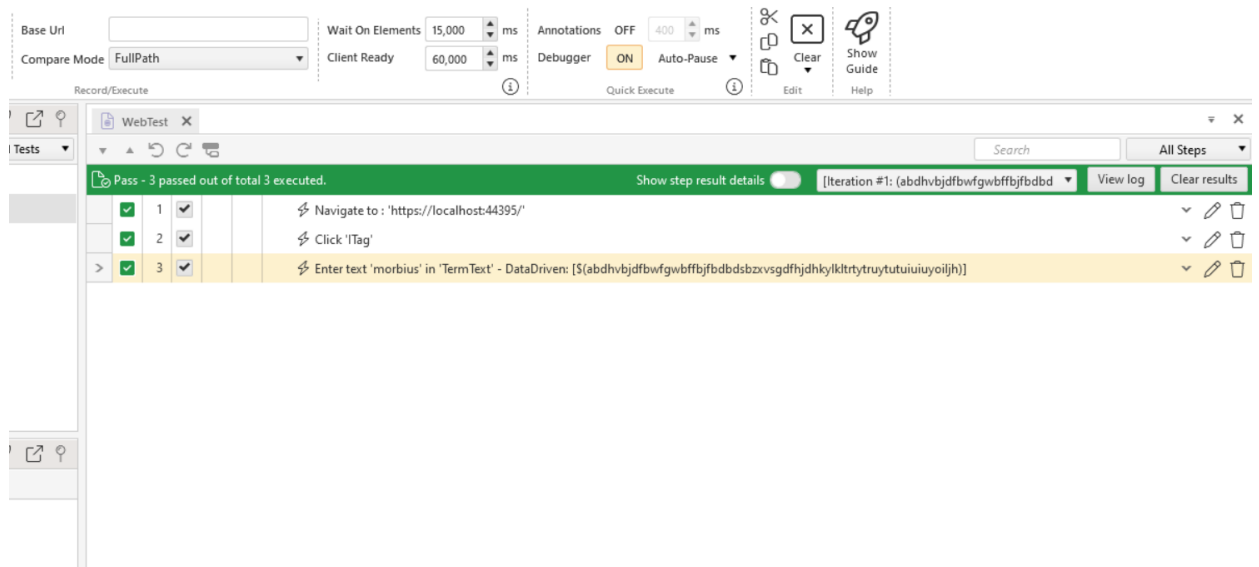


Hình 4. 31. Xuất file CSV

B2: Vào công cụ Telerik add link Url để chạy web xây dựng kịch bản kiểm thử

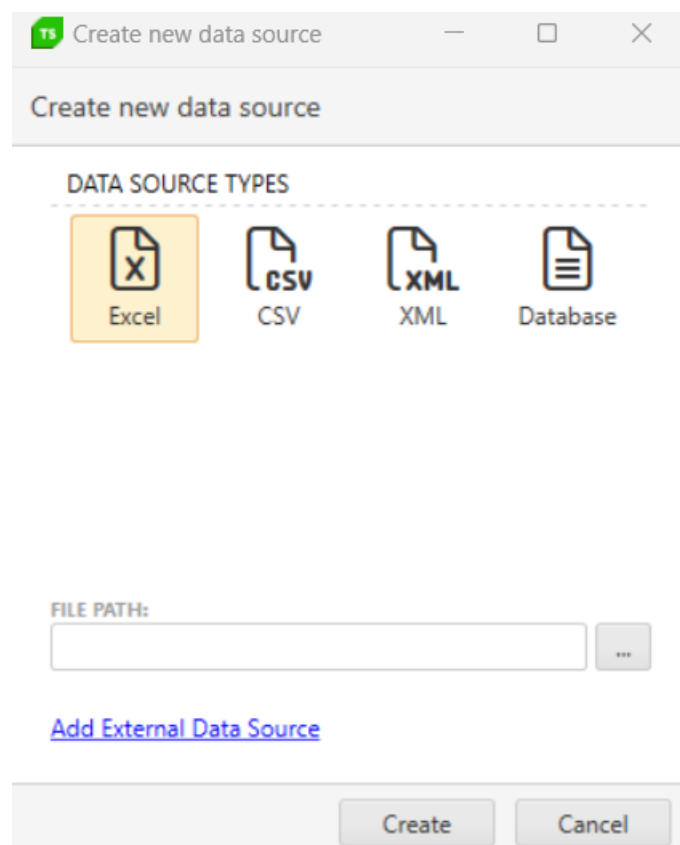


Hình 4. 32. Nhập link url



Hình 4. 33. Kịch bản kiểm thử

B3: Upload file CSV vừa xuất



Hình 4. 34. Áp dụng kiểm thử

B4: Sau khi upload file thành công ta vào “Data binding” để tiến hành kiểm tra Scripts trên file của mình. Nhập “Test Name”

Bind test to a data source

SELECT DATA SOURCE:
test2 - (CSVFile)

SELECT TABLE:
Select Table

☐ Filter data between rows: 1 - 1 (i.e 2-4) Update

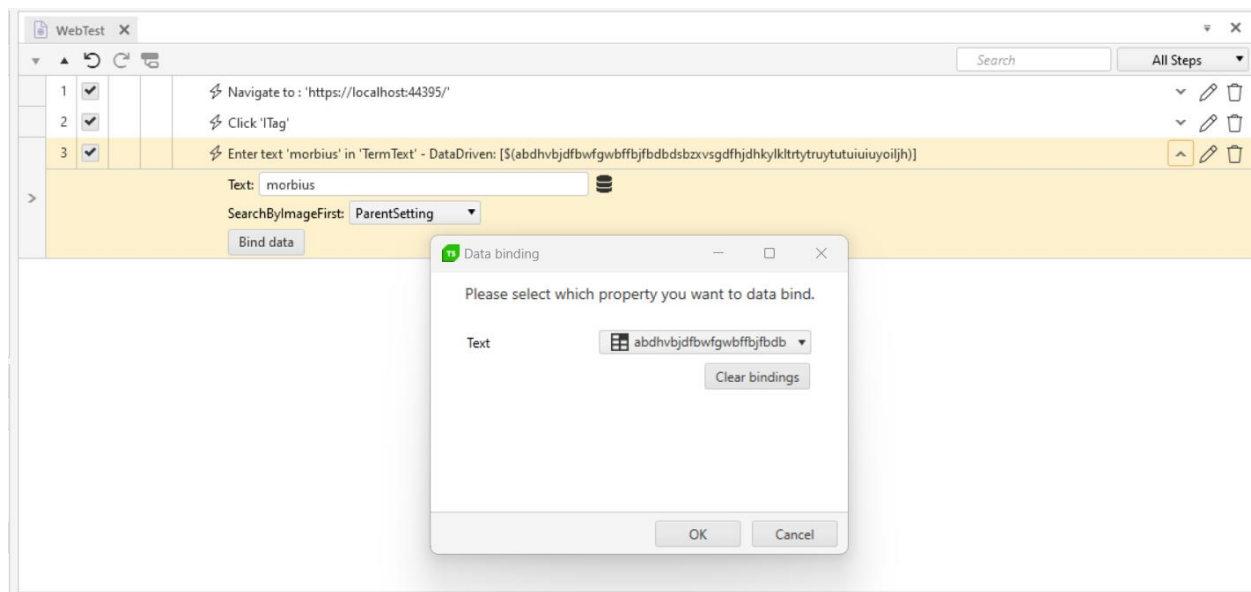
PREVIEW:

#	ABDHVB/DFBWFGWBFFBJFBDSDS8ZXVSGDFHJDHKYLLKTRTYTRUYTUTUIUIUYOILJH
1	!@#\$
2	xyz123
3	morbius
4	Free Guy
5	fbrjgbeJRGBKEGKTHNKHNYJHJHHJTKNGEANGKHG

Bind Cancel

Hình 4. 35. Áp dụng kiểm thử

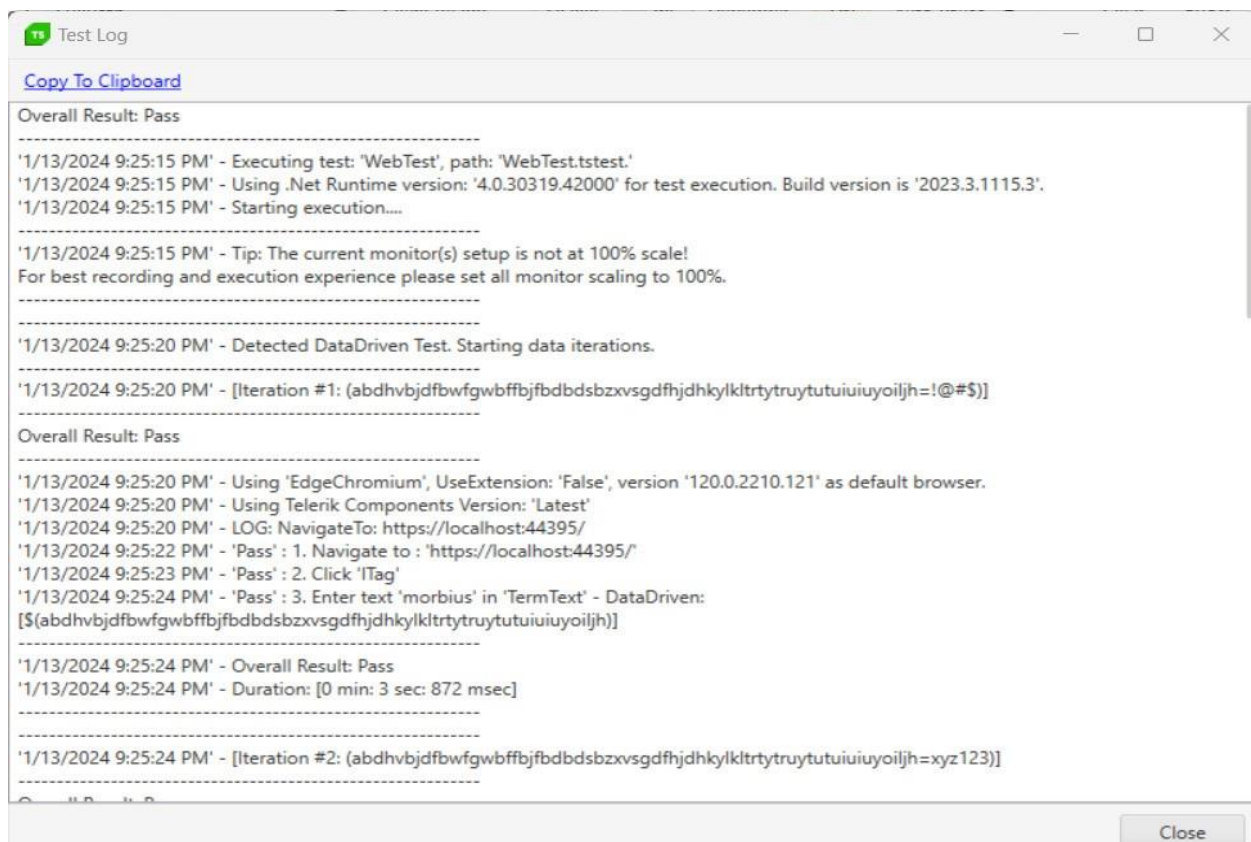
B5: “ADD” Scripts của file CSV vào để tiến hành viết kịch bản cho chức năng mình muốn kiểm thử



Hình 4. 36. Áp dụng kiểm thử

B6: Sau khi add mình bấm “Run test”

B7: Sau khi Run ta vào “Show log” để xem kết quả kiểm thử



Hình 4. 37. Áp dụng kiểm thử

Quá trình kiểm thử thành công và trang web không xảy ra lỗi

CHƯƠNG 5: KẾT LUẬN

5.1. Kết luận

- Kiểm thử phần mềm là một quá trình quan trọng trong quá trình phát triển phần mềm. Nó giúp đảm bảo rằng phần mềm đáp ứng các yêu cầu của khách hàng và không có lỗi.
- Trong báo cáo này, chúng em đã tìm hiểu về các khái niệm cơ bản của kiểm thử phần mềm, bao gồm các loại kiểm thử, các giai đoạn kiểm thử và các phương pháp kiểm thử. Chúng ta cũng đã tìm hiểu về một số công cụ kiểm thử phần mềm phổ biến, bao gồm Katalon Studio, TestComplete, Telerik Test Studio và Selenium IDE.
- Từ những gì đã học được, chúng em có thể kết luận rằng kiểm thử phần mềm là một quá trình phức tạp và cần được thực hiện bởi các kỹ sư kiểm thử có kinh nghiệm. Kiểm thử phần mềm phải được thực hiện trong suốt quá trình phát triển phần mềm để đảm bảo chất lượng của sản phẩm.

5.2. Kiến nghị

- Lập kế hoạch kiểm thử: Các nhóm kiểm thử cần lập kế hoạch kiểm thử trước khi bắt đầu quá trình kiểm thử. Kế hoạch kiểm thử nên xác định các loại kiểm thử sẽ được thực hiện, các giai đoạn kiểm thử và các tài nguyên cần thiết.
- Tự động hóa kiểm thử: Tự động hóa kiểm thử có thể giúp tiết kiệm thời gian và công sức. Các nhóm kiểm thử nên xem xét tự động hóa các thử nghiệm đơn giản và lặp đi lặp lại.
- Trao đổi thông tin: Các nhóm kiểm thử cần trao đổi thông tin với các nhóm phát triển và khách hàng. Điều này sẽ giúp đảm bảo rằng các thử nghiệm được thực hiện chính xác và đáp ứng các yêu cầu của khách hàng.

Kiểm thử phần mềm là một quá trình quan trọng cần được thực hiện nghiêm túc. Bằng cách thực hiện các lời khuyên trên, các nhóm kiểm thử có thể giúp đảm bảo rằng phần mềm được phát hành đáp ứng các yêu cầu của khách hàng và không có lỗi.

TÀI LIỆU THAM KHẢO

- [1] Tài liệu môn Kiểm thử và đảm bảo chất lượng phần mềm (Đại học Hutech)
- [2] <https://tenten.vn/tin-tuc/selenium-la-gi>
- [3] <https://docs.katalon.com/docs>
- [4] <https://docs.telerik.com/teststudio/welcome>
- [5] <https://support.smartbear.com/testcomplete/docs/>