TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

KHOA CÔNG NGHỆ THÔNG TIN

------\*\*\*------



BÁO CÁO BÀI TẬP LỚN HỌC PHẦN:

KIỂM THỬ PHẦN MỀM

TÌM HIỂU VÀ SỬ DỤNG NHỮNG KỸ THUẬT KIỂM THỬ ĐƯỢC HỖ TRỢ BỞI CÔNG CỤ TEST COMPLETE

|  |  |  |
| --- | --- | --- |
| Giảng viên hướng dẫn | : | Ths Nguyễn Thái Cường |
| Nhóm – lớp | : | Nhóm 16 - 20232IT6084007 |
| Sinh viên | : | Nguyễn Tiến Dũng |
|  | : | Nguyễn Đức Huy |
|  | : | Nguyễn Kim Hiếu |
|  | : | Hồ Hải Hà |

Hà Nội, Năm 2024

LỜI CẢM ƠN

Đầu tiên, em xin gửi lời cảm ơn chân thành đến Trường Đại học Công Nghiệp Hà Nội đã đưa môn học Kiểm thử phần mềm vào chương trình giảng dạy. Đặc biệt, em xin gửi lời cảm ơn sâu sắc đến giảng viên bộ môn Thầy Nguyễn Thái Cường đã dạy dỗ, truyền đạt những kiến thức quý báu cho em trong suốt thời gian học tập vừa qua. Trong thời gian tham gia lớp học Kiểm thử phần mềm của thầy, em đã có thêm cho mình nhiều kiến thức bổ ích, tinh thần học tập hiệu quả, nghiêm túc. Đây chắc chắn sẽ là những kiến thức quý báu, là hành trang để em có thể vững bước sau này.

Bộ môn kiểm thử phần mềm là môn học thú vị, vô cùng bổ ích và có tính thực tế cao. Đảm bảo cung cấp đủ kiến thức, gắn liền với nhu cầu thực tiễn của sinh viên. Tuy nhiên, do vốn kiến thức còn nhiều hạn chế và khả năng tiếp thu thực tế còn nhiều bỡ ngỡ. Mặc dù em đã cố gắng hết sức nhưng chắc chắn bài tiểu luận khó có thể tránh khỏi những thiếu sót và nhiều chỗ còn chưa chính xác, kính mong thầy, cô xem xét và góp ý để bài tiểu luận của em được hoàn thiện hơn.

Em xin chân thành cảm ơn!

MỤC LỤC

[LỜI CẢM ƠN 2](#_Toc167325765)

[Chương 1. Giới thiệu kiểm thử phần mềm 5](#_Toc167325766)

[1.1 Tổng quan về kiểm thử phần mềm 5](#_Toc167325767)

[1.2 Quy trình kiểm thử phần mềm 6](#_Toc167325768)

[1.3 Các cấp độ kiểm thử 7](#_Toc167325769)

[1.4 Các loại hình kiểm thử 8](#_Toc167325770)

[1.5 Các kỹ thuật kiểm thử 10](#_Toc167325771)

[Chương 2. Tìm hiểu công cụ kiểm thử phần mềm 14](#_Toc167325772)

[2.1 Giới thiệu công cụ Test Complete 14](#_Toc167325773)

[2.2 Đặc điểm 15](#_Toc167325774)

[2.3 Cài đặt công cụ 16](#_Toc167325775)

[Chương 3. Kiểm thử website newshop.vn 18](#_Toc167325776)

[3.1 Lập kế hoạch kiểm thử 18](#_Toc167325777)

[3.2 Thực hiện kiểm thử 22](#_Toc167325778)

[3.2.1 Nguyễn Tiến Dũng – Quản lý tài khoản 23](#_Toc167325779)

[3.2.2 Nguyễn Kim Hiếu – Tìm kiếm 32](#_Toc167325780)

[3.2.3 Nguyễn Đức Huy – Quản lý tài khoản người dùng 36](#_Toc167325781)

[3.3.4 Hồ Hải Hà - Đăng kí 49](#_Toc167325782)

[KẾT LUẬN 54](#_Toc167325783)

[TÀI LIỆU THAM KHẢO 55](#_Toc167325784)

**DANH MỤC HÌNH ẢNH**

[Hinh 1. 1 Hình ảnh quy trình phần mềm 6](#_Toc167325585)

[Hinh 1. 2 Hỉnh ảnh các cấp độ kiểm thử 8](#_Toc167325586)

[Hinh 1. 3 Hình ảnh các loại hình kiểm thử 9](#_Toc167325587)

[Hinh 2. 1 Công cụ TestComplete 14](#_Toc167325595)

[Hình 3. 1 Giao diện quản lý tài khoản 23](#_Toc167325884)

[Hình 3. 2 TestCase quản lý tài khoản 32](#_Toc167325885)

[Hình 3. 3 Giao diện tìm kiếm 32](#_Toc167325886)

[Hình 3. 4 TestCase tìm kiếm 36](#_Toc167325887)

[Hình 3. 5 Giao diên quản lý tài khoản người dùng 36](#_Toc167325888)

[Hình 3. 6 TestCase quản lý tài khoản người dùng 39](#_Toc167325889)

[Hình 3. 7Giao diện đăng kí 49](#_Toc167325890)

[Hình 3. 8 TestCase đăng kí 51](#_Toc167325891)

Chương 1. Giới thiệu kiểm thử phần mềm

1.1 Tổng quan về kiểm thử phần mềm

Theo IEEE và ISO, phần mềm không chỉ bao gồm mã nguồn (chương trình máy tính) mà còn bao gồm các thủ tục, tài liệu hỗ trợ và dữ liệu cần thiết để vận hành. Điều này nghĩa là, ngoài mã nguồn, phần mềm còn bao gồm các quy trình hoạt động, tài liệu hướng dẫn và dữ liệu cần thiết để chương trình hoạt động một cách hiệu quả.

Phần mềm, được kỹ nghệ chứ không chế tạo theo nghĩa cổ điển, bao gồm mã nguồn, quy trình, tài liệu hỗ trợ và dữ liệu cần thiết để vận hành. Nó có tính phức tạp cao, luôn thay đổi theo nhu cầu người dùng và môi trường vận hành. Mặc dù không thể nhìn thấy được, phần mềm có thể được mô tả qua các khía cạnh như sơ đồ điều khiển, mô hình luồng dữ liệu và mô hình tương tác. Quá trình phát triển phần mềm, quyết định giá thành và chất lượng của nó, đòi hỏi sự kiểm tra và kiểm soát chặt chẽ để tìm ra và sửa lỗi, đảm bảo chất lượng và hiệu suất.

Chất lượng phần mềm được ảnh hưởng bởi con người, quy trình và công cụ. Có ba khoảng cách chính có thể xuất hiện trong quá trình phát triển phần mềm. Đầu tiên, giữa yêu cầu người dùng và bản đặc tả yêu cầu hệ thống, có thể xảy ra khi không hiểu rõ yêu cầu của người dùng hoặc bỏ qua yêu cầu. Thứ hai, giữa bản đặc tả và sản phẩm, có thể xảy ra khi hiểu sai yêu cầu đặc tả hoặc có các yêu cầu được thêm vào trong quá trình phát triển nhưng không được thêm vào bản đặc tả. Cuối cùng, giữa yêu cầu người dùng và sản phẩm, xảy ra khi sản phẩm không thỏa mãn yêu cầu người dùng. Độ lệch này phụ thuộc vào hai cạnh còn lại của tam giác chất lượng và là độ lệch gây tốn kém nhất để sửa chữa. Những khoảng cách này cần được quản lý và giảm thiểu để đảm bảo chất lượng phần mềm.

Mục tiêu của kiểm thử phần mềm là tìm ra càng nhiều lỗi càng tốt trong điều kiện về thời gian đã định và nguồn lực sẵn có. Nó cũng nhằm chứng minh rằng sản phẩm phần mềm phù hợp với các đặc tả của nó. Quá trình này cần xác thực chất lượng kiểm thử phần mềm đã dùng chi phí và nỗ lực tối thiểu. Đồng thời, việc thiết kế tài liệu kiểm thử một cách có hệ thống và thực hiện nó sao cho có hiệu quả, tiết kiệm được thời gian công sức cũng là một mục tiêu quan trọng trong kiểm thử phần mềm.

Kiểm thử phần mềm đóng vai trò quan trọng trong việc đảm bảo chất lượng sản phẩm. Mặc dù một số người phát triển phần mềm cho rằng mục đích của kiểm thử là để chứng minh chương trình không có lỗi và đã thực hiện đúng các chức năng đã đưa ra, nhưng thực tế, kiểm thử còn giúp tìm ra lỗi và sửa chữa các lỗi đó nhằm tăng độ tin cậy cho phần mềm. Ví dụ, trong một thử nghiệm rút tiền từ máy ATM, người kiểm thử thử nghiệm với các yêu cầu rút tiền khác nhau và phát hiện ra lỗi khi máy cho phép rút quá số tiền tối đa cho phép. Như vậy, kiểm thử không chỉ giúp xem xét chất lượng sản phẩm mà còn giúp phát hiện và khắc phục lỗi, đảm bảo chức năng hoạt động đúng như mong đợi.

1.2 Quy trình kiểm thử phần mềm

A blue rectangular arrows with white text

Description automatically generated

Hinh 1. 1 Hình ảnh quy trình phần mềm

Quy trình kiểm thử phần mềm tổng quát thường bao gồm bốn giai đoạn chính:

Lập kế hoạch kiểm thử (Test Planning): Giai đoạn này bao gồm việc xác định mục tiêu kiểm thử, phương pháp kiểm thử, và tài nguyên cần thiết. Kế hoạch kiểm thử sẽ 2 được tạo ra để hướng dẫn toàn bộ quá trình kiểm thử.

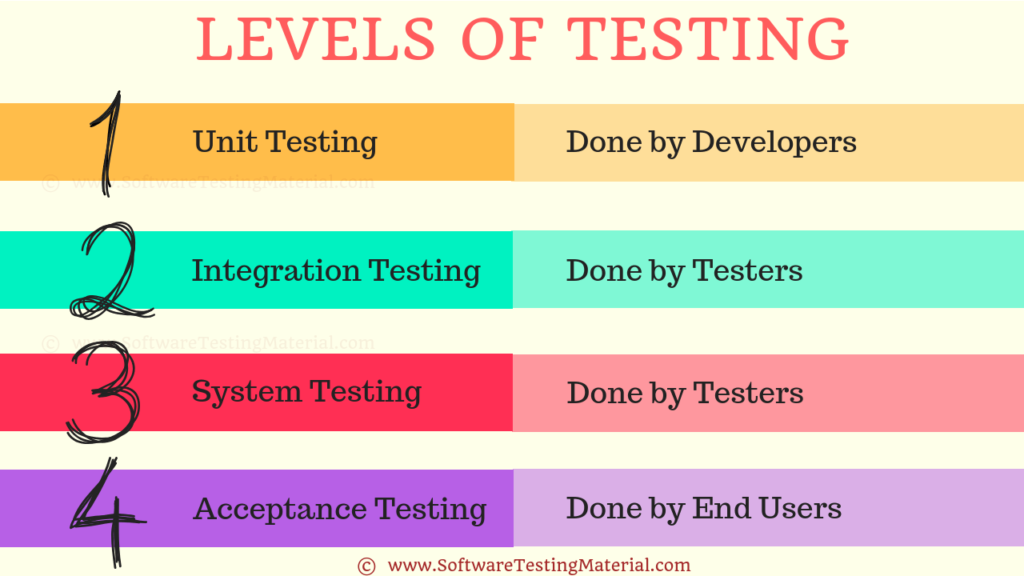
Phân tích và thiết kế kiểm thử (Test Analysis and Design): Trong giai đoạn này, đội ngũ kiểm thử sẽ phân tích yêu cầu và thiết kế các trường hợp kiểm thử dựa trên yêu cầu đó. Mục tiêu là để đảm bảo rằng tất cả các chức năng của phần mềm đều được kiểm thử đầy đủ.

Thực hiện kiểm thử (Test Execution): Giai đoạn này bao gồm việc thực hiện các trường hợp kiểm thử đã được thiết kế, ghi lại kết quả, và so sánh kết quả với kết quả mong đợi.

Báo cáo và đánh giá kiểm thử (Test Report and Evaluation): Cuối cùng, sau khi tất cả các trường hợp kiểm thử đã được thực hiện, kết quả kiểm thử sẽ được tổng hợp và đánh giá. Báo cáo kiểm thử sẽ được tạo ra để mô tả quá trình kiểm thử, kết quả, và các khuyến nghị cho các bước tiếp theo.

Bản kế hoạch kiểm thử phần mềm cơ bản bao gồm việc lập ra mục tiêu, tiêu chí chấp nhận, yêu cầu kiểm thử, và chiến lược kiểm thử. Nó cũng xác định các tài nguyên cần thiết cho việc kiểm thử, bao gồm con người, hệ thống phần cứng và phần mềm. Các mốc kiểm thử được xác định để theo dõi tiến trình và đảm bảo rằng tất cả các yêu cầu kiểm thử đều được thực hiện. Kết quả của quá trình kiểm thử được tổng hợp và đánh giá trong báo cáo kiểm thử, giúp đảm bảo chất lượng sản phẩm và tìm ra các lỗi có thể xảy ra.

1.3 Các cấp độ kiểm thử



Hinh 1. 2 Hỉnh ảnh các cấp độ kiểm thử

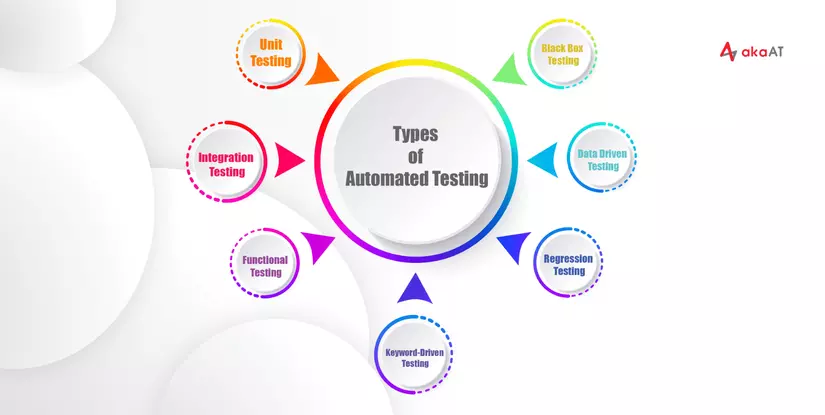
Kiểm thử đơn vị (Unit Testing): Kiểm thử trên các hàm hay thành phần riêng lẻ, cần hiểu biết về thiết kế chương trình và code.

Kiểm thử tích hợp (Integration Testing): Kiểm thử trên các nhóm các thành phần được kết hợp với nhau, cần hiểu biết về kiến trúc hệ thống và giao diện.

Kiểm thử hệ thống (System Testing): Kiểm thử trên toàn bộ hệ thống, cần hiểu biết về yêu cầu và chức năng của hệ thống.

Kiểm thử chấp nhận (Acceptance Testing): Kiểm thử trên hệ thống hoàn thiện, cần hiểu biết về nhu cầu và mong đợi của khách hàng.

1.4 Các loại hình kiểm thử



Hinh 1. 3 Hình ảnh các loại hình kiểm thử

***Kiểm thử chức năng***

Qui trình cố gắng tìm ra các khác biệt giữa đặc tả bên ngoài của phần mềm và thực tế mà phần mềm cung cấp. Đặc tả bên ngoài của phần mềm là đặc tả chính xác về hành vi của phần mềm theo góc nhìn của người dùng thấy.

Các loại kiểm thử chức năng:

- Kiểm thử chức năng của hệ thống: Mục tiêu của loại kiểm thử này là đảm bảo đúng mục tiêu của kiểm thử chức năng: nhập dữ liệu- xử lý- lấy và kiểm tra kết quả trả về. Kiểm tra sản phẩm phần mềm và các hoạt động của các chức năng bên trong sản phẩm đó bằng cách tương tác thông qua giao diện người dùng của sản phẩm, phân tích kết quả trả về.

- Kiểm thử tích hợp dữ liệu và cơ sở dữ liệu: Kiểm tra các chức năng của một sản phẩm hay hệ thống phần mềm hoạt động đúng không sau khi sản phẩm pm đó đã có sự tích hợp hay đưa dữ liệu cũ, dữ liệu đã có sẵn từ bên ngoài vào sản phẩm. Đảm bảo các chức năng của hệ thống mới sử dụng được các dữ liệu cũ.

- Kiểm thử vòng lặp công việc: Đảm bảo hoạt động của các công việc được chạy tự động theo lịch đã đặt trước không do người dùng tác động.

- Kiểm thử kiểm soát truy cập: Đảm bảo các tác nhân, người sử dụng chỉ có thể truy cập vào đúng chức năng họ được phép truy cập. Đảm bảo chỉ những người dùng được phân quyền truy cập hệ thống mới có thể truy cập vào hệ thống và thông qua các gateway thích hợp.

- Kiểm thử giao diện: Mục tiêu kiểm tra giao diện của các chức năng trong một sản phẩm hệ thống hoạt động so với thiết kế. Kiểm thử giao diện cần kiểm thử: Liên kết hay chuyển tiếp, cách thức truy cập ( Sử dụng phím tab, chuột…), kiểm tra các đối tượng trên màn hình bao gồm: Màu sắc, Vị trí, Kích thước chữ, kiểu đối tượng.

***Kiểm thử phi chức năng***

Tập trung vào kiểm thử sản phẩm, hệ thống phần mềm cần kiểm thử có những đặc tính tốt như thế nào. Kiểm thử phi chức năng có thể được sử dụng ở mọi cấp độ kiểm thử nhưng thường được sử dụng hiệu quả nhất trong cấp độ kiểm thử hệ thống và kiểm thử chấp nhận sản phẩm.

Các loại kiểm thử phi chức năng bao gồm:

- Kiểm thử hiệu năng (performance testing): Bảo đảm tối ưu việc phân bổ tài nguyên hệ thống nhằm đạt các chỉ tiêu như thời gian xử lý hay đáp ứng câu truy vấn.

- Kiểm thử tải trọng (load testing): Tập trung vào xác định đặc tính hiệu suất của hệ thống hay sản phẩm phần mềm trong điều kiện tải hay upload cụ thể. Bảo đảm hệ thống vận hành đúng dưới áp lực cao.

- Kiểm thử tập trung (stress testing): Tập trung vào xác định đặc tính hiệu suất của hệ thống hay sản phẩm phần mềm trong điều kiện tải hay upload cụ thể. Bảo đảm hệ thống vận hành đúng dưới áp lực cao.

- Kiểm thử với lượng dữ liệu lớn (volume testing): Kiểm thử tập trung vào việc xác định hoặc xác nhận đặc tính hiệu suất của hệ thống hoặc ứng dụng được kiểm thử trong điều kiện hệ thống có lượng dữ liệu rất lớn. Dữ liệu lớn có thể là cơ sở dữ liệu lớn hoặc dữ liệu trong file upload lên hệ thống có dung lượng lớn.

***Kiểm thử liên quan đến sự thay đổi***

Thực hiện hoạt động kiểm thử khi có sự thay đổi trên hoặc trong sản phẩm phần mềm. Sự thay đổi của sản phẩm phần mềm có thể là: Sửa chữa các lỗi tìm được, sản phẩm được nâng cấp, được thay đổi về chức năng.

Các loại kiểm thử liên quan đến sự thay đổi gồm:

- Kiểm thử lại: Chỉ kiểm thử những test case chưa pass. Khi thực hiện kiểm thử xác nhận cần chú ý thực hiện kiểm thử đúng các bước như trong mô tả trường hợp kiểm thử gồm đúng các tập đầu vào, đúng các dữ liệu, đúng môi trường kiểm thử

- Kiểm thử hồi quy: Kiểm thử hồi quy để đảm bảo rằng những thay đổi mới không làm ảnh hưởng đến những phần đã hoàn thiện trước đó. Kiểm thử hồi quy thường được thực hiện tự động. Các phép thử hồi quy được chia làm 3 loại gồm: Các phép thử đại diện: thực hiện tất cả chức năng của pm, các phép thử bổ sung: tập trung vào chức năng dễ bị ảnh hưởng nhất khi có thay đổi, các phép thử tập trung: tập trung vào thành phần pm bị thay đổi

1.5 Các kỹ thuật kiểm thử

***Kiểm thử hộp trắng***

Là phương pháp kiểm nghiệm dựa vào cấu trúc/mã lệnh chương trình. Phương pháp white-box kiểm nghiệm một chương trình (một phần chương trình, hay một hệ thống, một phần của hệ thống) đáp ứng tốt tất cả các giá trị input bao gồm cả các giá trị không đúng hay không theo dự định của chương trình.

*Phương pháp kiểm nghiệm white-box dựa trên:*

- Các câu lệnh (statement)

- Đường dẫn (path)

- Các điều kiện (condition)

- Vòng lặp (loop)

- Ngã rẽ (branch)

*Các loại kiểm thử hộp trắng :*

- Kiểm thử đơn vị: Đây thường là loại kiểm thử đầu tiên được thực hiện trên một ứng dụng. Kiểm tra đơn vị được thực hiện trên từng đơn vị hoặc khối mã khi nó được phát triển. Kiểm tra đơn vị giúp xác định phần lớn lỗi, sớm trong vòng đời phát triển phần mềm. Lỗi được xác định trong giai đoạn này rẻ hơn và dễ sửa.

- Kiểm tra rò rỉ bộ nhớ : Rò rỉ bộ nhớ là nguyên nhân hàng đầu khiến các ứng dụng chạy chậm hơn.

*Ưu điểm của kiểm thử hộp trắng*

- Tối ưu hóa mã bằng cách tìm các lỗi ẩn.

- Các trường hợp kiểm thử hộp trắng có thể dễ dàng được tự động hóa.

- Thử nghiệm kỹ lưỡng hơn vì tất cả các đường dẫn mã thường được bao phủ.

- Thử nghiệm có thể bắt đầu sớm trong SDLC ngay cả khi GUI không khả dụng.

*Nhược điểm của kiểm thử White-Box:*

- Kiểm thử hộp trắng có thể khá phức tạp và tốn kém.

- Các nhà phát triển thường thực hiện các ca kiểm thử hộp trắng. Thử nghiệm hộp trắng của nhà phát triển không chi tiết và có thể dẫn đến lỗi sản xuất.

- Kiểm thử hộp trắng yêu cầu nguồn lực chuyên nghiệp với sự hiểu biết chi tiết về lập trình và triển khai.

- Kiểm thử hộp trắng tốn nhiều thời gian, các ứng dụng lập trình lớn hơn cần thời gian để kiểm tra đầy đủ.

***Kiểm thử hộp đen***

Kiểm thử hộp đen là một phương pháp kiểm thử phần mềm trong đó các chức năng của ứng dụng phần mềm được kiểm tra mà không cần biết về cấu trúc mã nội bộ, chi tiết triển khai và đường dẫn nội bộ.

Kiểm thử hộp đen chủ yếu tập trung vào đầu vào và đầu ra của các ứng dụng phần mềm và nó hoàn toàn dựa trên các yêu cầu và thông số kỹ thuật của phần mềm.

*Các loại kiểm thử hộp đen :*

- Kiểm thử chức năng: Loại kiểm thử hộp đen này liên quan đến các yêu cầu chức năng của hệ thống; nó được thực hiện bởi những người kiểm thử phần mềm.

- Kiểm thử phi chức năng: Loại kiểm thử hộp đen này không liên quan đến kiểm thử chức năng cụ thể, mà là các yêu cầu phi chức năng như hiệu suất, khả năng mở rộng, khả năng sử dụng.

- Kiểm tra hồi quy: Kiểm tra hồi quy được thực hiện sau khi sửa mã, nâng cấp hoặc bất kỳ bảo trì hệ thống nào khác để kiểm tra mã mới không ảnh hưởng đến mã hiện có.

*Kỹ thuật kiểm thử hộp đen :*

- Kiểm thử lớp tương đương: Nó được sử dụng để giảm thiểu số lượng các trường hợp kiểm thử có thể đến mức tối ưu trong khi vẫn duy trì phạm vi kiểm thử hợp lý.

- Kiểm tra giá trị ranh giới: Kiểm tra giá trị ranh giới tập trung vào các giá trị tại các ranh giới. Kỹ thuật này xác định xem một phạm vi giá trị nhất định có được hệ thống chấp nhận hay không. Nó rất hữu ích trong việc giảm số lượng các trường hợp thử nghiệm. Nó phù hợp nhất cho các hệ thống có đầu vào nằm trong phạm vi nhất định.

- Kiểm tra bảng quyết định: Bảng quyết định đặt các nguyên nhân và tác động của chúng trong một ma trận. Có một sự kết hợp độc đáo trong mỗi cột.

*Ưu điểm của kiểm thử hộp đen :*

- Người kiểm thử không cần phải có thêm kiến thức chức năng hoặc kỹ năng lập trình để thực hiện Kiểm thử hộp đen.

- Nó hiệu quả để thực hiện các bài kiểm tra trong hệ thống lớn hơn.

- Các thử nghiệm được thực hiện từ quan điểm của người dùng hoặc khách hàng.

- Các trường hợp thử nghiệm có thể dễ dàng tái sản xuất.

- Nó được sử dụng để tìm ra sự mơ hồ và mâu thuẫn trong các đặc tả chức năng.

*Nhược điểm của kiểm thử hộp đen*

- Có khả năng lặp lại các bài kiểm tra tương tự trong khi thực hiện quy trình kiểm tra.

- Nếu không có đặc tả chức năng rõ ràng, các trường hợp thử nghiệm rất khó thực hiện.

- Rất khó để thực hiện các trường hợp thử nghiệm vì đầu vào phức tạp ở các giai đoạn thử nghiệm khác nhau.

- Đôi khi, lý do cho sự thất bại kiểm tra không thể được phát hiện.

- Một số chương trình trong ứng dụng không được thử nghiệm.

- Nó không tiết lộ các lỗi trong cấu trúc điều khiển.

- Làm việc với một không gian mẫu lớn của đầu vào có thể mệt mỏi và tiêu tốn nhiều thời gian.

Chương 2. Tìm hiểu công cụ kiểm thử phần mềm

2.1 Giới thiệu công cụ Test Complete



Hinh 2. 1 Công cụ TestComplete

TestComplete là một công cụ kiểm thử tự động phổ biến và mạnh mẽ được phát triển bởi công ty SmartBear Software. Nó được sử dụng để kiểm tra ứng dụng desktop, web và di động trên nhiều nền tảng khác nhau như Windows, macOS, iOS và Android.

TestComplete cung cấp một môi trường kiểm thử đồ họa dễ sử dụng, cho phép người dùng ghi và tái tạo các bước kiểm thử một cách tự động. Điều này giúp giảm thiểu sự phụ thuộc vào kiến thức kỹ thuật và mã lập trình. Bạn có thể ghi lại hoặc viết các kịch bản kiểm thử bằng ngôn ngữ kịch bản bên trong TestComplete như VBScript, JScript, C++ Script, hoặc sử dụng các công cụ kiểm thử khác như Python hoặc JavaScript.

***Lịch sử phát triển:***

TestComplete được phát triển bởi công ty AutomatedQA Corporation, được thành lập vào năm 1999 bởi Sergey Korolev và Denis Zubarev. Ban đầu, công ty tập trung vào việc phát triển các công cụ kiểm thử và kiểm tra tự động cho phần mềm. Trong quá trình phát triển, công ty đã phát triển một số sản phẩm chính như AutomatedQA TestComplete, AQtime, và TestExecute.

AutomatedQA TestComplete ban đầu ra mắt vào năm 2001 và đã trở thành một trong những công cụ kiểm thử tự động phổ biến trong ngành phần mềm. Công cụ này đã được thiết kế để hỗ trợ kiểm thử tự động cho các ứng dụng desktop và web trên nền tảng Windows.

Vào năm 2008, AutomatedQA Corporation được sáp nhập với công ty SmartBear Software, một công ty phát triển phần mềm chuyên về các công cụ hỗ trợ phát triển và kiểm thử phần mềm. Việc sáp nhập này đã định hình lại tương lai của TestComplete và mang lại sự bổ sung và phát triển liên tục cho công cụ.

***Các phiên bản:***

- Aqtest 1.x

- TestComplete 2.x

- TestComplete 3.x

- TestComplete 4.x

- TestComplete 5.x

- TestComplete 6.x

- TestComplete 7.x

- TestComplete 8.x

- TestComplete 9.x

- TestComplete 10.x

- TestComplete 11.x

- TestComplete 12.x

- TestComplete 14.x

- TestComplete 15.x

2.2 Đặc điểm

TestComplete, là một công cụ kiểm thử phần mềm đóng, hiện đang ở phiên bản mới nhất là 15.52. Đặc biệt, TestComplete hỗ trợ kiểm thử đa nền tảng, giúp kiểm tra hiệu suất và tính năng của phần mềm trên nhiều hệ điều hành và môi trường khác nhau. Điều này giúp đảm bảo rằng phần mềm hoạt động một cách ổn định và hiệu quả trên tất cả các nền tảng mà nó hỗ trợ.

***TestComplete hỗ trợ các loại kiểm thử như:***

- Kiểm thử giao diện người dùng (UI testing): TestComplete cho phép ghi lại và tái tạo các hành động của người dùng trên giao diện người dùng của ứng dụng để kiểm tra tính tương thích và độ tin cậy.

- Kiểm thử chức năng (Functional testing): TestComplete cho phép kiểm tra các chức năng của ứng dụng bằng cách kiểm tra các phương thức, hàm, lớp, và module riêng lẻ của mã nguồn.

- Kiểm thử hiệu suất (Performance testing): TestComplete cung cấp các tính năng để đo và kiểm tra hiệu suất của ứng dụng, bao gồm thời gian phản hồi, tải trọng, và sự ổn định.

***TestComplete cũng hỗ trợ nhiều ngôn ngữ lập trình như:***

- JavaScript và Python tích hợp với TestComplete

- VBScript

- JScript

- DelphiScript

- C++Script

- C#Script

2.3 Cài đặt công cụ

***Các hệ điều hành hỗ trợ:***

- Windows 8.1 and Windows 10 (64-bit).

- Windows Server 2012 - 2019.

- macOS 10.13 (High Sierra) - 10.15 (Catalina).

- 64-bit Oracle Linux 6 - 8.

- 64-bit Red Hat Enterprise Linux 6 - 8.

- 64-bit Suse Linux Enterprise Server 12 - 15.

- Ubuntu Linux 20.04.

***Phiên bản java hỗ trợ:***

- Từ java 11 trở lên là sẽ được hỗ trợ

***Yêu cầu hệ thống(Khuyến nghị):***

- Intel Core i5 hoặc i7 (thế hệ thứ 3 trở lên) hoặc là vi xử lý AMD tương đương

- 4 GB Ram hoặc hơn.

- 500 MB ổ đĩa trống cho việc cài đặt.

- Hệ điều hành 64 bit.

***Yêu cầu hệ thống(tối thiểu):***

- Intel core I3(từ thế hệ thứ 3 trở lên)

- 2 GB hoặc hơn

- 500 Mb ổ đĩa trống cho việc cài đặt

Chương 3. Kiểm thử website newshop.vn

3.1 Lập kế hoạch kiểm thử

***Mục đích lập kế hoạch:***

Test Plan được thiết kế để quy định phạm vi, cách tiếp cận, nguồn lực và lịch trình của tất cả các hoạt động thử nghiệm của dự án

Kế hoạch xác định các hạng mục cần kiểm tra, các tính năng cần kiểm tra, các loại thử nghiệm sẽ được thực hiện, nhân viên chịu trách nhiệm thử nghiệm, các nguồn lực và lịch trình cần thiết để hoàn thành thử nghiệm và các rủi ro liên quan đến kế hoạch.

***Các tài liệu liên quan:***

|  |  |  |  |
| --- | --- | --- | --- |
| Tên tài liệu | Tác giả | Phiên bản | Ngày duyệt |
| Tài liệu mô tả yêu cầu hệ thống | Phạm Văn Hùng | 02 | 5/5/2024 |
| Mẫu kế hoạch kiểm thử | Phạm Văn Hà | 01 | 5/5/2024 |
| Mẫu test case | Phạm Văn Hưng | 02 | 5/5/2024 |
| Mẫu test report | Phạm Văn Huy | 01 | 5/5/2024 |

***Giới thiệu chung dự án:***

newshop.vn là một trang web dành cho việc mua sách trực tuyến. Như vậy, nó có thể là một nền tảng thương mại điện tử chuyên về việc bán sách từ nhiều thể loại khác nhau, bao gồm tiểu thuyết, sách giáo khoa, sách về kỹ năng và kiến thức chuyên sâu, và nhiều hơn nữa. Trang web có thể cung cấp thông tin về các tác giả, nhận xét từ khách hàng, cũng như các tính năng khác để giúp người dùng tìm kiếm và mua sách dễ dàng.

***Phạm vi kiểm thử:***

Tính năng cần kiểm tra: Tính năng của dự án đã được xác định trong thông số kỹ thuật yêu cầu phần mềm là cần phải tốt nhất.

Tính năng không được kiểm tra: Các tính năng này không được kiểm tra vì chúng không được bao gồm trong thông số kỹ thuật yêu cầu phần mềm bao gồm:

- Giao diện người dùng

- Giao diện phần cứng

***Các rủi ro***

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Rủi ro | Cách khắc phục | Mức độ rủi ro |
| 1 | Thành viên trong nhóm thiếu các kỹ năng cần thiết để kiểm tra trang web | Lập kế hoạch khóa đào tạo để nâng cao kỹ năng cho các thành viên của bạn | Cao |
| 2 | Tiến độ dự án quá chặt chẽ, thật khó để hoàn thành dự án này đúng hạn | Đặt mức độ ưu tiên kiểm tra cho từng hoạt động kiểm tra . | Cao |
| 3 | Test Manager có kỹ năng quản lý kém | Lập kế hoạch đào tạo lãnh đạo cho người quản lý | Cao |
| 4 | Thiếu hợp tác ảnh hưởng tiêu cực đến năng suất của nhân viên | Khuyến khích mỗi thành viên trong nhóm trong nhiệm vụ của mình và truyền cảm hứng cho họ để nỗ lực lớn hơn . | Cao |
| 5 | Dự toán ngân sách sai và vượt chi phí | Thiết lập phạm vi trước khi bắt đầu công việc, chú ý nhiều đến việc lập kế hoạch dự án và liên tục theo dõi và đo lường tiến độ | Cao |

***Kiểm tra Logistics***

Dự án được các thành viên nhóm 1 kiểm thử.

Thử nghiệm sẽ xảy ra khi :

- Người kiểm thử sẽ bắt đầu thực hiện kiểm thử khi tất cả các đầu vào sau đã sẵn sàng

- Phần mềm có sẵn để thử nghiệm

- Test Specification được tạo

- Môi trường thử nghiệm được xây dựng

- Đủ nguồn nhân lực để thử nghiệm

***Mục tiêu kiểm tra***

Mục tiêu thử nghiệm là kiểm chứng chức năng của dự án, dự án nên tập trung thử nghiệm chức năng của trang để đảm bảo tất cả các hoạt động này có thể hoạt động bình thường trong môi trường thực tế .

***Các tiêu chí kiểm tra***

Tiêu chí đình chỉ: Nếu các thành viên trong nhóm báo cáo rằng có 40% trường hợp kiểm thử không thành công, hãy tạm dừng kiểm thử cho đến khi nhóm phát triển khắc phục tất cả các trường hợp thất bại .

Tiêu chí thoát: Chỉ định các tiêu chí biểu thị sự hoàn thành thành công của giai đoạn thử nghiệm

- Tỷ lệ chạy là bắt buộc phải là 100% trừ khi có lý do rõ ràng .

- Tỷ lệ vượt qua là 80%, đạt được tỷ lệ vượt qua là bắt buộc.

***Lập kế hoạch tài nguyên***

*Tài nguyên hệ thống:*

|  |  |  |
| --- | --- | --- |
| No | Tài nguyêm | Mô tả |
| 1 | Công cụ kiểm tra | TestComplete |
| 2 | Mạng lưới | Có kết nối internet |
| 3 | Máy tính | Ít nhất 1 máy tính chạy Windows 10, Ram 8GB |

*Nguồn nhân lực:*

|  |  |  |
| --- | --- | --- |
| No | Thành viên | Nhiệm vụ |
| 1. | Cả nhóm | Quản lý toàn bộ dự án Xác định hướng dự án |
| 2. | Cả nhóm | Xác định và mô tả các kỹ thuật / công cụ kiểm tra / kiến trúc tự động hóa thích hợp  Thực hiện các bài kiểm tra, Ghi nhật ký kết quả, Báo cáo lỗi. Thành viên thuê ngoài |
| 3. | Cả nhóm | Thực hiện các trường hợp kiểm thử, chương trình kiểm thử, bộ kiểm thử , v.v. |
| 4. | Cả nhóm | Xây dựng và đảm bảo môi trường thử nghiệm  Hỗ trợ Tester sử dụng môi trường kiểm thử để thực hiện kiểm thử |
| 5. | Cả nhóm | Phụ trách đảm bảo chất lượng |

***Môi trường thử nghiệm***

Trên nhiều thiết bị khác nhau cũng như nhiều loại trình duyệt

***Lịch trình và ước tính***

*Tất cả các nhiệm vụ và dự toán dự án :*

|  |  |  |
| --- | --- | --- |
| Nhiệm vụ | Thành viên | Ước tính nỗ lực |
| Tạo đặc tả kiểm thử | Cả nhóm | 10 ngày |
| Thực hiện kiểm thử | Cả nhóm | 5 ngày |
| Báo cáo thử nghiệm | Cả nhóm | 12 ngày |
| Kết thúc kiểm thử | Cả nhóm | 1 ngày |
| Tất cả |  | 28 ngày |

***Sản phẩm thử nghiệm:***

*Trước giai đoạn thử nghiệm :*

- Tài liệu kế hoạch kiểm tra .

- Tài liệu trường hợp kiểm thử

- Thông số kỹ thuật thiết kế thử nghiệm .

*Trong quá trình thử nghiệm :*

- Công cụ kiểm tra

- Mô phỏng.

- Dữ liệu thử nghiệm

- Nhật ký lỗi và nhật ký thực thi.

*Sau khi chu kỳ thử nghiệm kết thúc :*

- Kết quả/báo cáo kiểm tra

- Báo cáo lỗi

- Hướng dẫn quy trình cài đặt / kiểm tra

- Ghi chú phát hành

3.2 Thực hiện kiểm thử

3.2.1 Nguyễn Tiến Dũng – Quản lý tài khoản



Hình 3. 1 Giao diện quản lý tài khoản

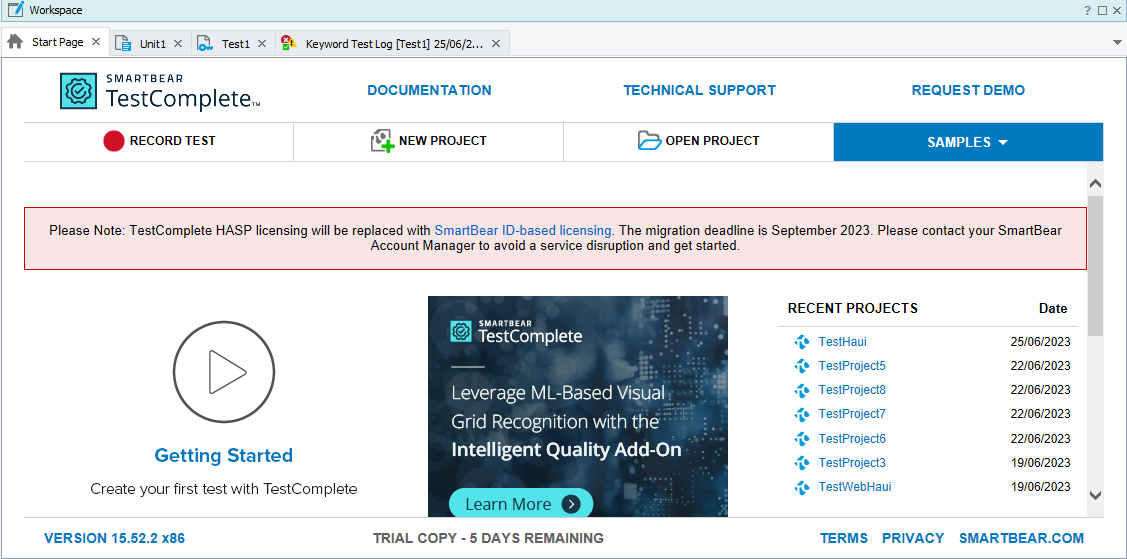
***Phân tích thiết kế kiểm thử***

* Lập danh sách các TC (theo template của cả nhóm) và ghi chú rõ các điều kiện bao phủ hay cơ chế tìm TC,
* Chức năng này dùng để chỉnh sửa các thông tin cá nhân của chủ sỡ hữu tài khoản
* Giao diện chức năng gồm các thông tin động ( lấy từ cơ sở dữ liệu của hệ thống) bao gồm các thông tin như:
  + Id
  + Họ và tên
  + Ngày sinh
  + Điện thoại
  + Địa chỉ
  + Tỉnh thành
  + CMND
  + Giới thiệu
* Điều kiện ràng buộc: Phải đăng nhập thành công trước đót trong 7 nguyên tắc kiểm thử phần mềm, phần chức năng này nằm trong nguyên tắc “ Lỗi đi theo cụm”
* Kỹ thuật kiểm thử sử dụng: “Phân tích giá trị biên” và “Bảng quyết định”

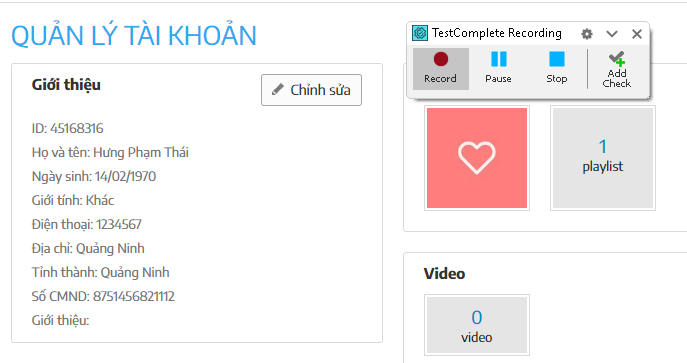
***Thực hiện kiểm thử***

Các bước kiểm thử tự động với công cụ TestComplete:

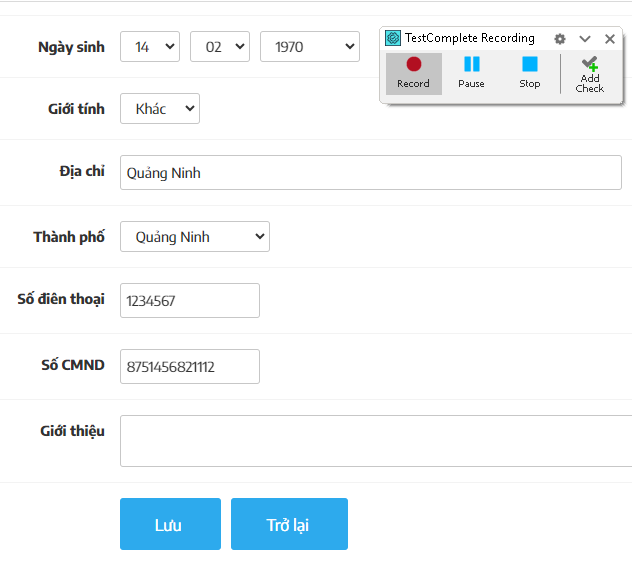
Ở Start Page chúng ta có thể chọn Record Test luôn Hoặc New Project đều sẽ tạo một project mới (Record Test sau khi tạo project xong thì sẽ quay luôn ) hoặc lựa chọn các project đã tạo sẵn rồi lại chọn Record Test để bắt đầu quay



Sau khi Đã Record Test Complete sẽ hiện lên 1 màn hình Pop Up các thao tác cho chúng ta lựa chọn

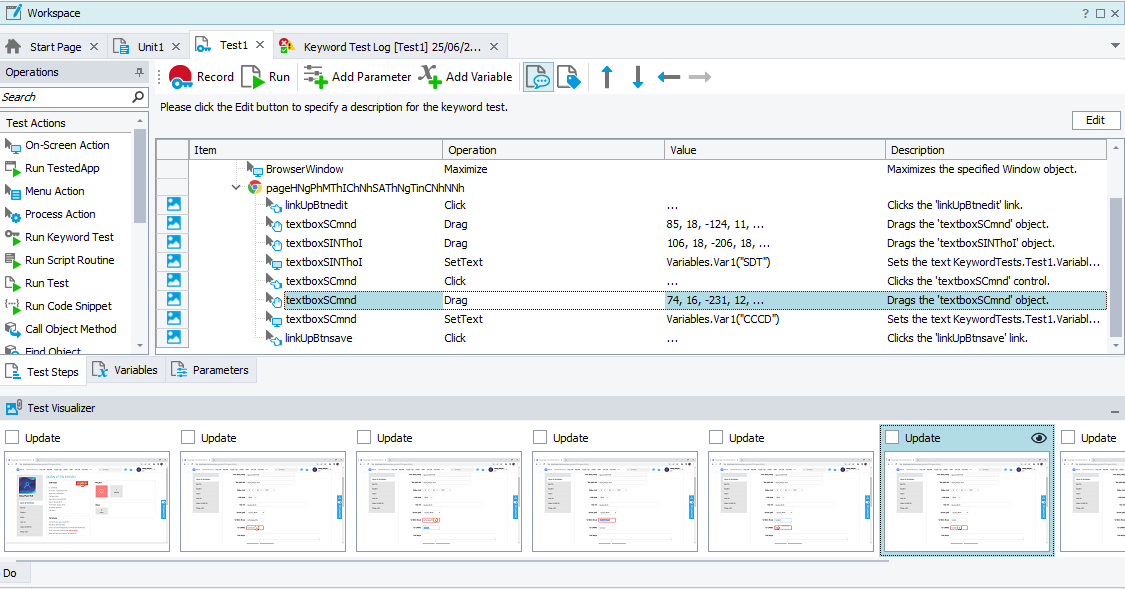


Chúng ta sẽ bắt đầu thực hiện các thao tác mô phỏng người dùng sử dụng như bình thường



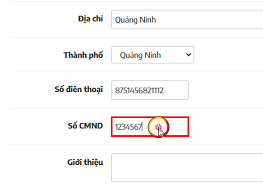
Sau khi hoàn thành xong chúng ta ấn Stop trên pop up của Test Complete

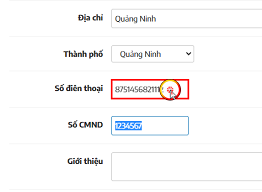
Phần mềm sẽ đưa chúng ta đến 1 màn hình hiển thị toàn bộ các thao tác chúng ta đã sử dụng cũng như tự động chụp lại màn hình chúng ta đã đi qua

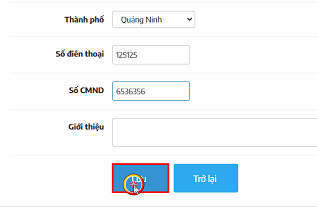


Các màn hình đã được Test Complete tự động chụp lại (Phần khoanh đỏ chính là nơi con trỏ chuột dc trỏ đến và được hệ thống nhận diện như nơi cần test)

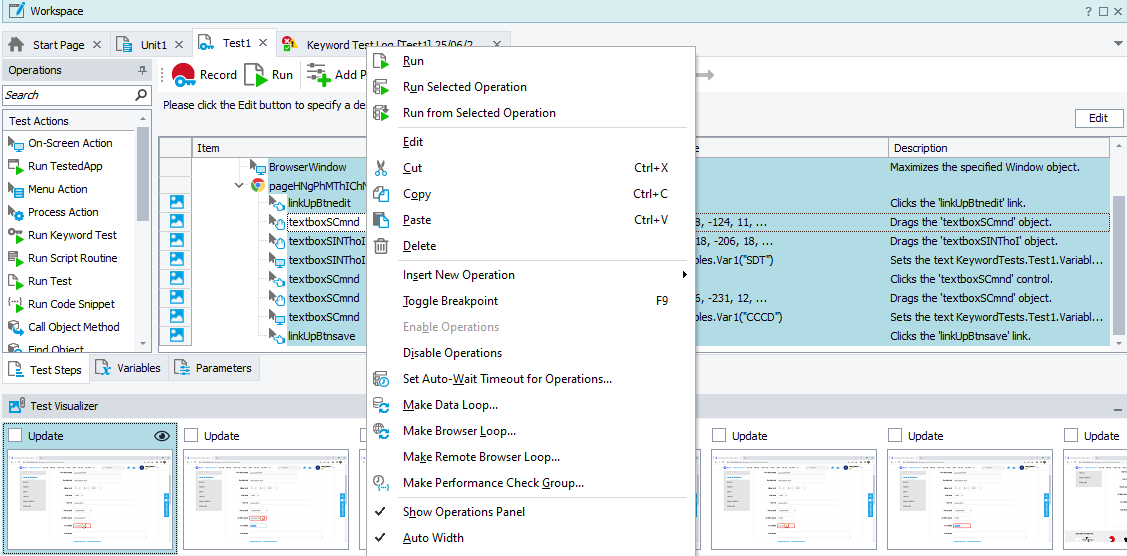




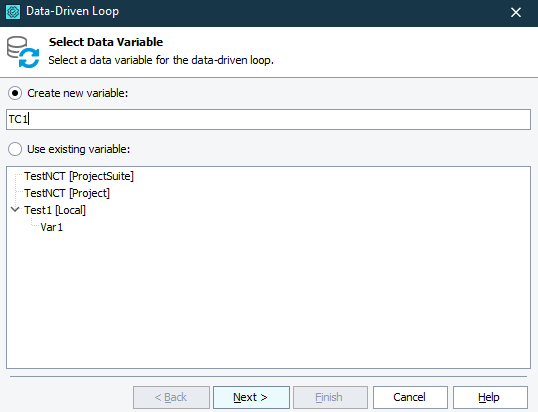




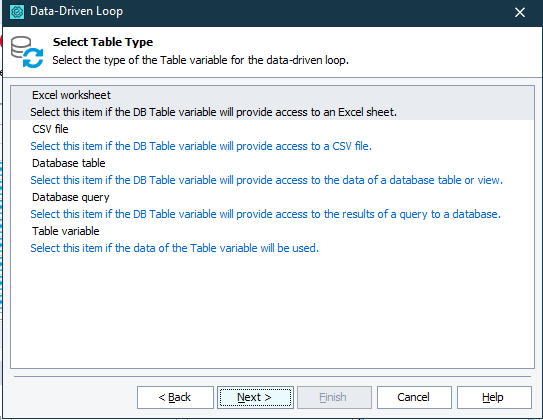
Để công cụ có thể thực hiện tự động lặp lại nhiều lần các bước chúng ta vừa thực hiện ở trên ở H1 ta bôi đen toàn bộ các thao tác rồi chuột phải nhấn “Make data loop”



Bước tiếp theo ta chọn 1 tên biến sau đó nhấn ‘Next”



Tiếp theo ta chọn 1 loại dữ liệu để Test Complete nhập vào như test case trên thì dùng excel

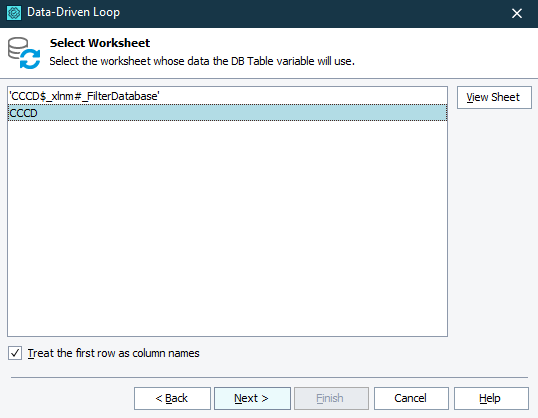


Lưu ý: khi dùng excel phiên bản 32bit thì TestComplete cũng phải dùng 32bit và excel 64bit thì TestComplete cũng là 64bit

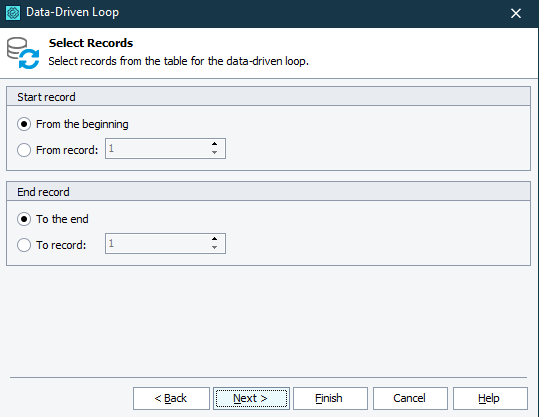
Bước tiếp theo chúng ta có thể chọn 1 trường dữ liệu đã có sẵn hoặc tạo luôn mới



Chọn Sheet muốn lấy dữ liệu

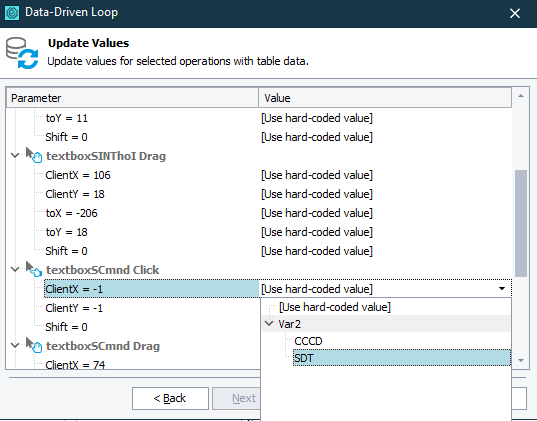


Sau đó thiết lập các bước lặp

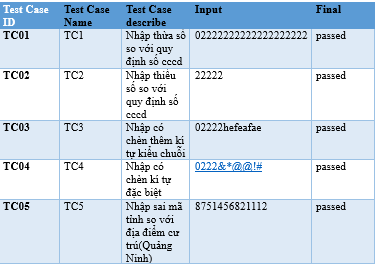


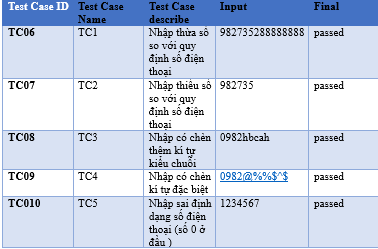
Có thể để mặc định là 1:1 để mô phỏng y nguyên thao tác người dùng với các dữ liệu khác nhau

Cuối cùng chọn tham chiếu cho các thao tác đã dùng là ở Test Box dùng tham chiếu “SDT” và “CCCD”



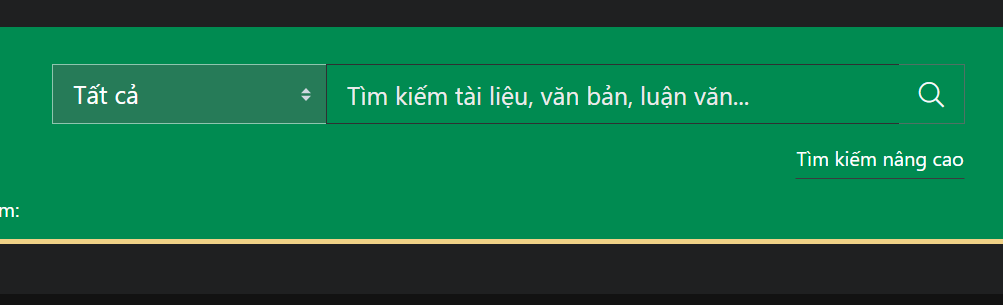
Nhấn Finish và Run để công cụ thực hiện việc tự động





Hình 3. 2 TestCase quản lý tài khoản

3.2.2 Nguyễn Kim Hiếu – Tìm kiếm

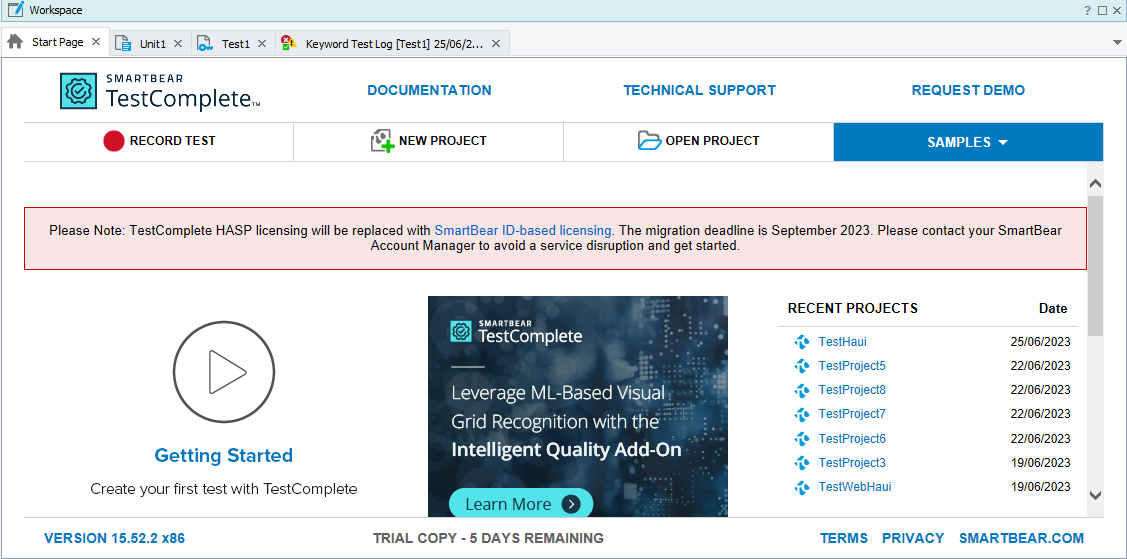


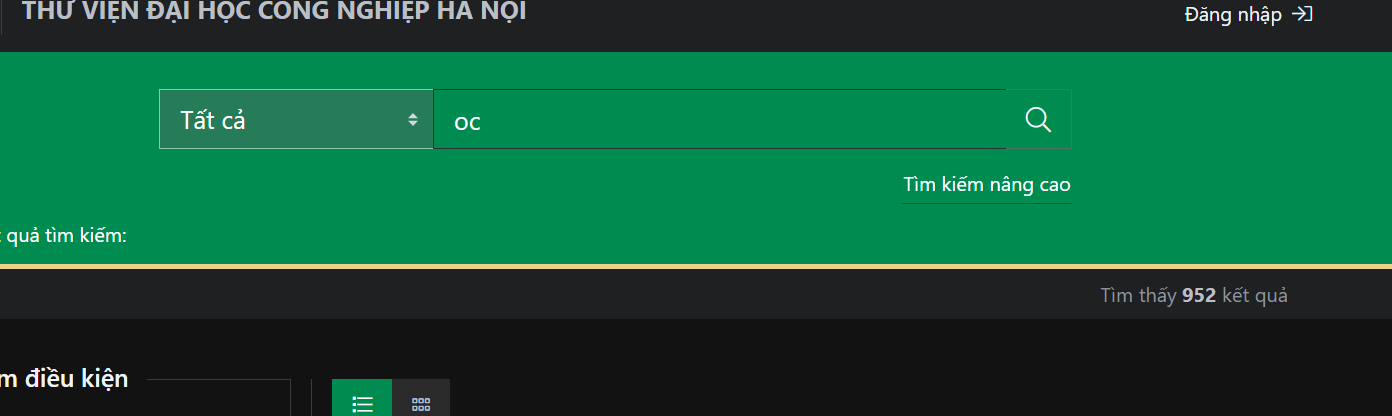
Hình 3. 3 Giao diện tìm kiếm

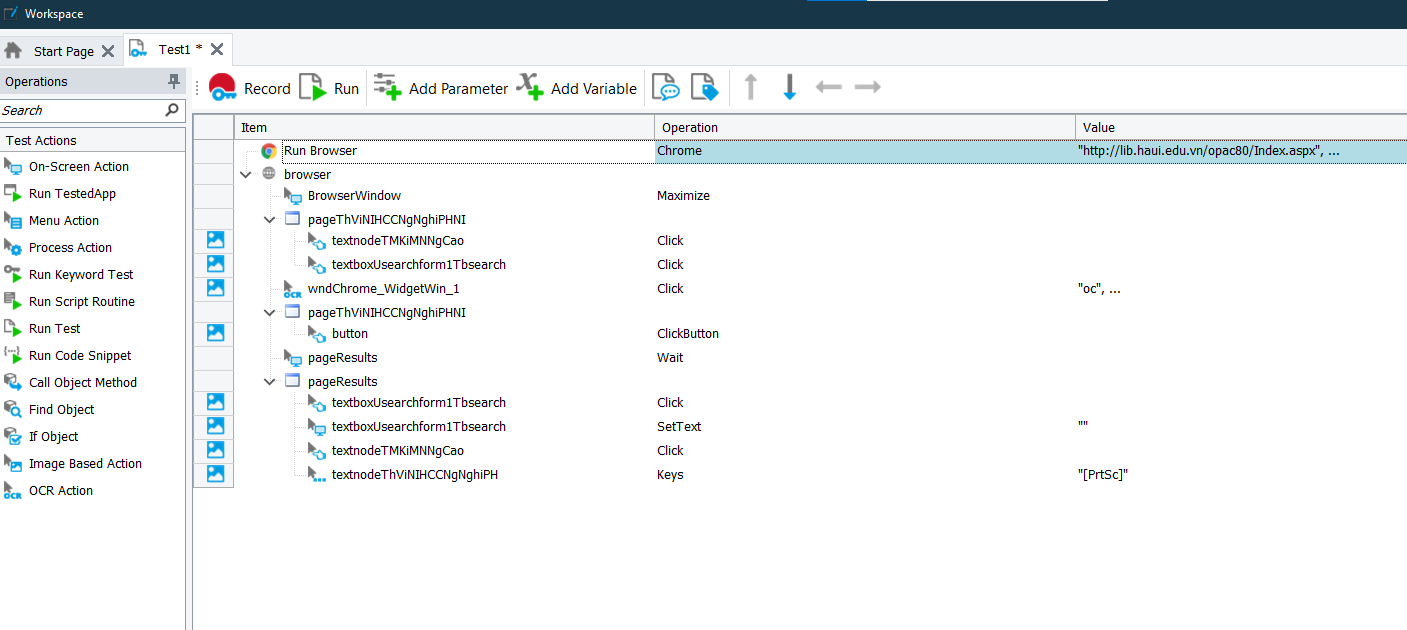
***Phân tích thiết kế kiểm thử***

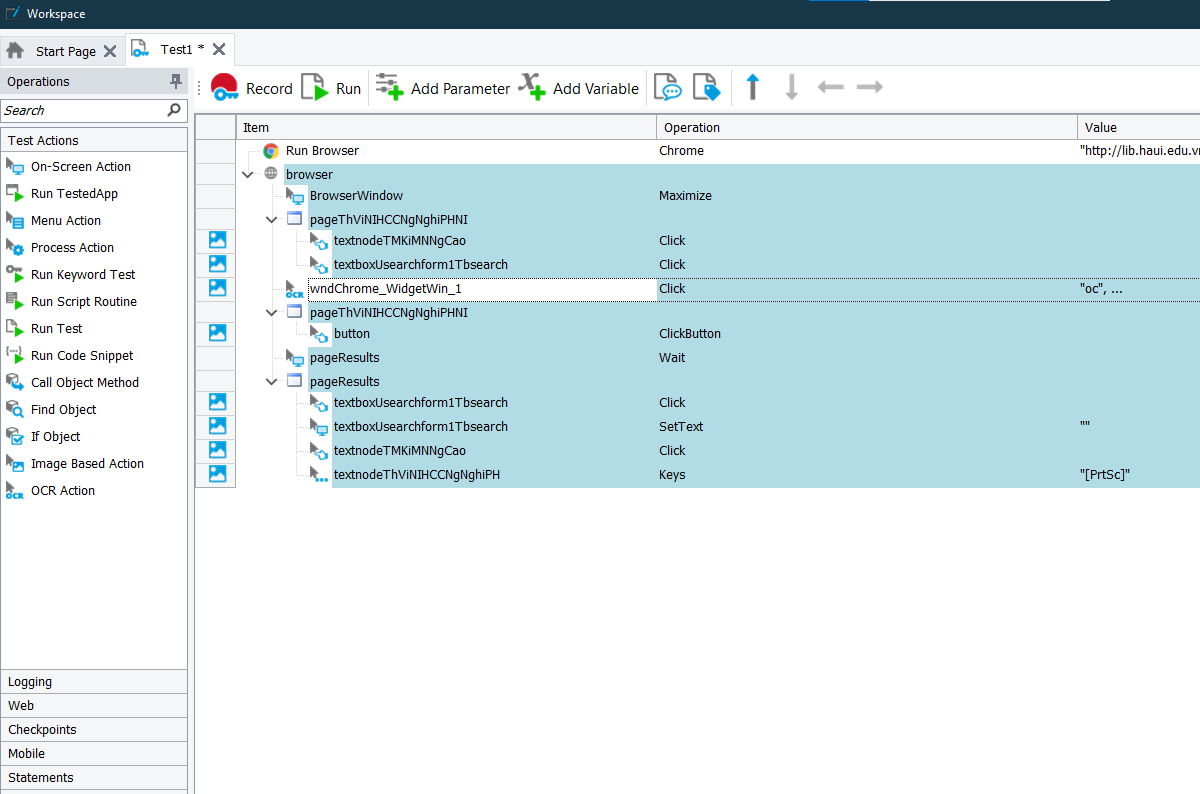
* Lập danh sách các TC (theo template của cả nhóm) và ghi chú rõ các điều kiện bao phủ hay cơ chế tìm TC,
* Chức năng này dùng để tìm kiếm những tài liệu có sẵn.
* Điều kiện ràng buộc: Phải đăng nhập thành công trước đó
* Trong 7 nguyên tắc kiểm thử phần mềm, phần chức năng này nằm trong nguyên tắc “Lỗi đi theo cụm”
* Kỹ thuật kiểm thử sử dụng: “Phân tích giá trị biên” và “Bảng quyết định”

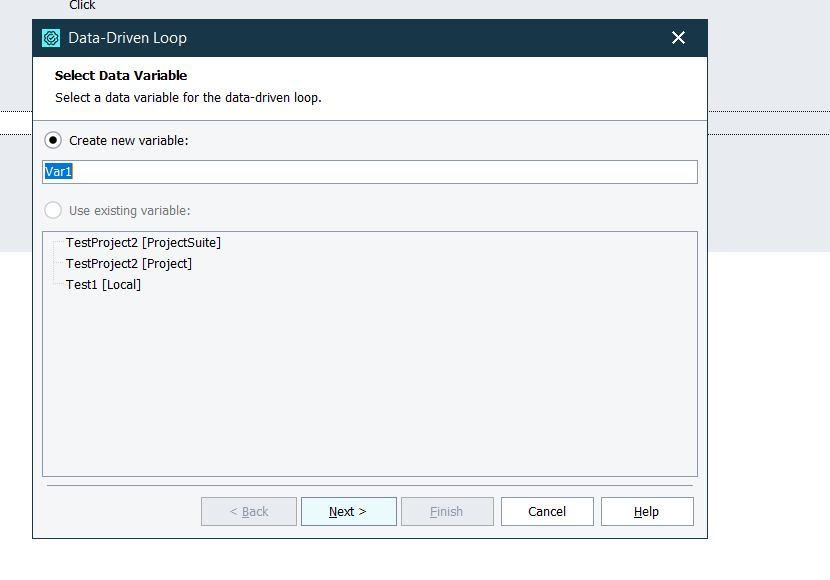
***Thực hiện kiểm thử***

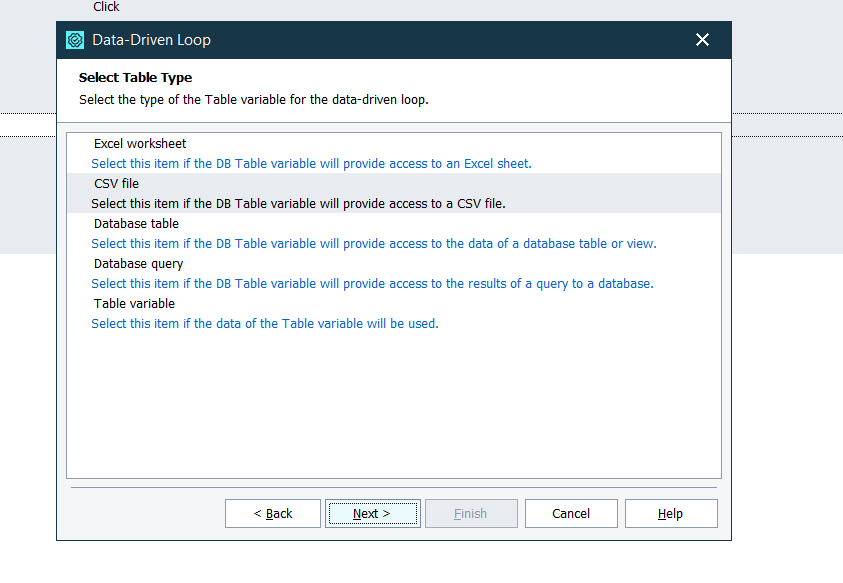


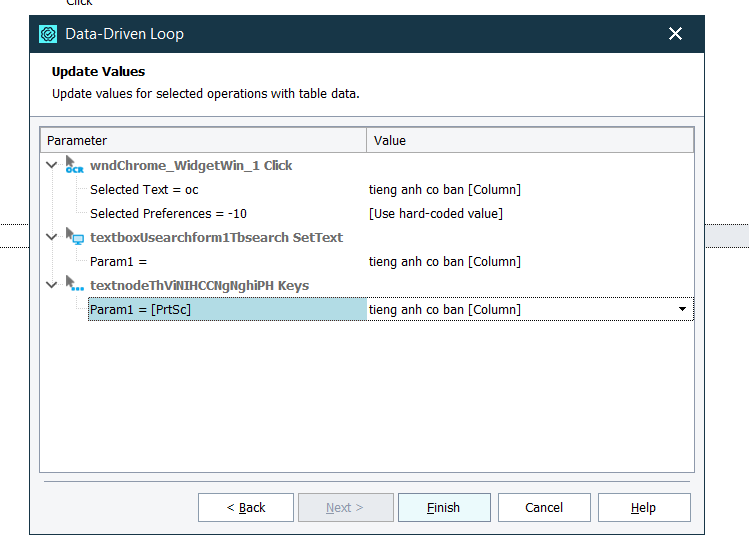
****

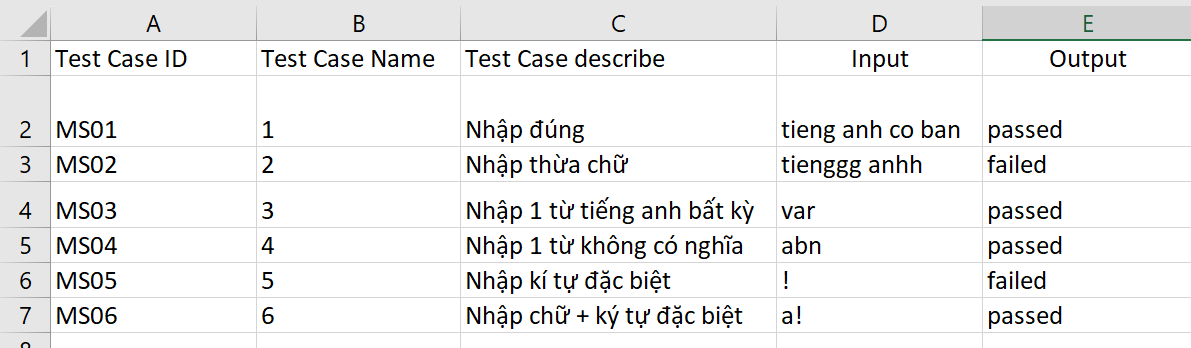
****

****

****

****

****

****

Hình 3. 4 TestCase tìm kiếm

3.2.3 Nguyễn Đức Huy – Quản lý tài khoản người dùng

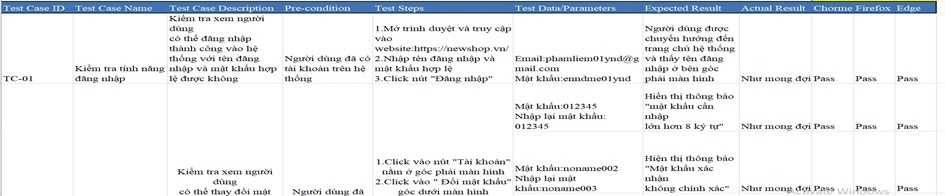


Hình 3. 5 Giao diên quản lý tài khoản người dùng

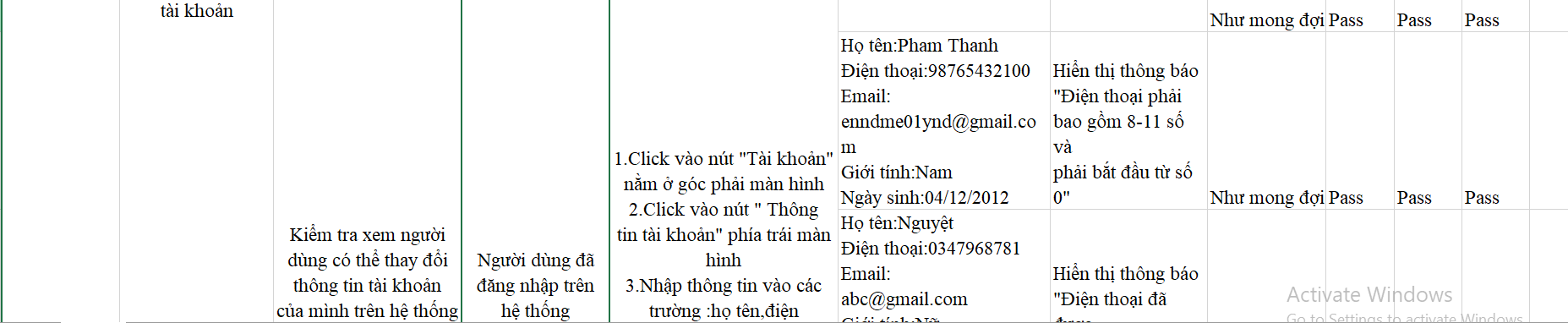
***Phân tích thiết kế kiểm thử***

* Chức năng này dùng để chỉnh sửa các thông tin cá nhân của chủ sỡ hữu tài khoản cũng như thay đổi mật khẩu
* Giao diện chức năng gồm các thông tin động (lấy từ cơ sở dữ liệu của hệ thống) bao gồm các thông tin như:
  + Họ và tên
  + Điên thoại
  + Ngày sinh
  + Email
  + Giới tính
  + Mật khẩu
  + Xác nhận mật khẩu
* Điều kiện ràng buộc: Phải đăng nhập thành công trước đó
* Trong 7 nguyên tắc kiểm thử phần mềm, phần chức năng này nằm trong nguyên tắc “Lỗi đi theo cụm”
* Kỹ thuật kiểm thử sử dụng: “Phân tích giá trị biên” và “Bảng quyết định” ngoài ra còn sử dụng kỹ thuật experience-based
* Cơ chế tìm TC dựa trên kỹ thuật phân vùng tương đương và tìm giá trị biên
* VD:Như trường nhập SĐT: phải bao gồm từ “8-11 số và bắt đầu bằng số 0” thì chúng ta có thể chia là các vùng
* Phân vùng 1:Chọn 10 số bất kỳ và bắt đầu bằng số 0 “Hợp lệ”
* Phân vùng 2:Chọn 12 số bất kỳ và bắt đầu bằng số 0 “Không hợp lệ”
* Phân vùng 3:Viết 10 chữ và bắt đầu bằng số 0 “Không hợp lệ”
* Phân vùng 4:Nhập 7 sos bất kỳ “Không hợp lệ”
* Phân vùng 5:Nhập 9 số và bắt đầu bằng số 0 nhưng lại có ký tự đặc biệt “Không hợp lệ”
* Phân vùng 6:Nhập 10 số và không bắt đầu bằng số 0 “Không hợp lệ”
* Như vậy chỉ dựa vào kỹ thuật phân vùng ta đã có thể tìm được nhiều trường hợp có thể kiểm thử và còn rất nhiều kỹ thuật cơ chế tìm TC khác nữa.

***Lập danh sách các test case***

****

****

****

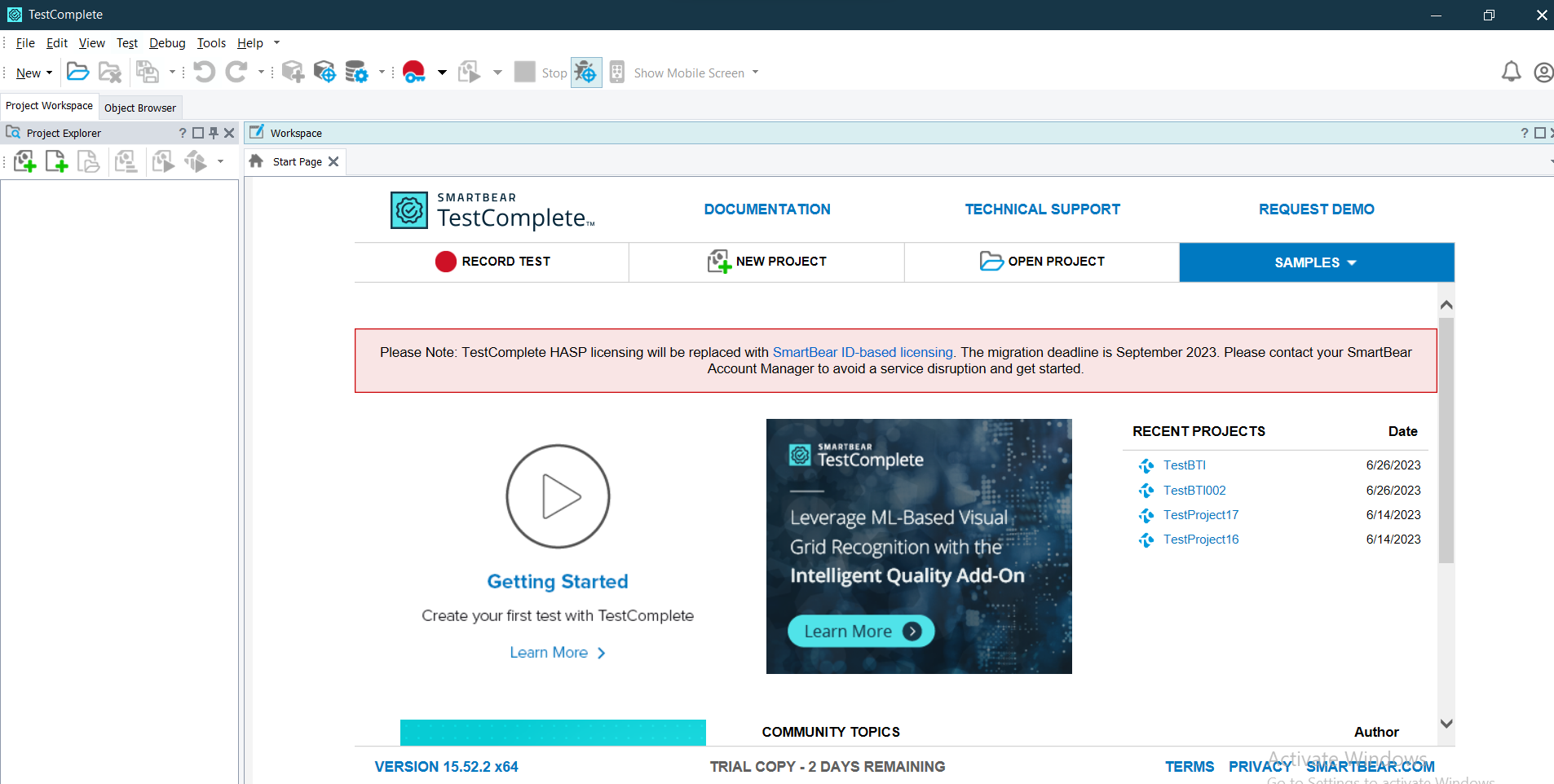
****

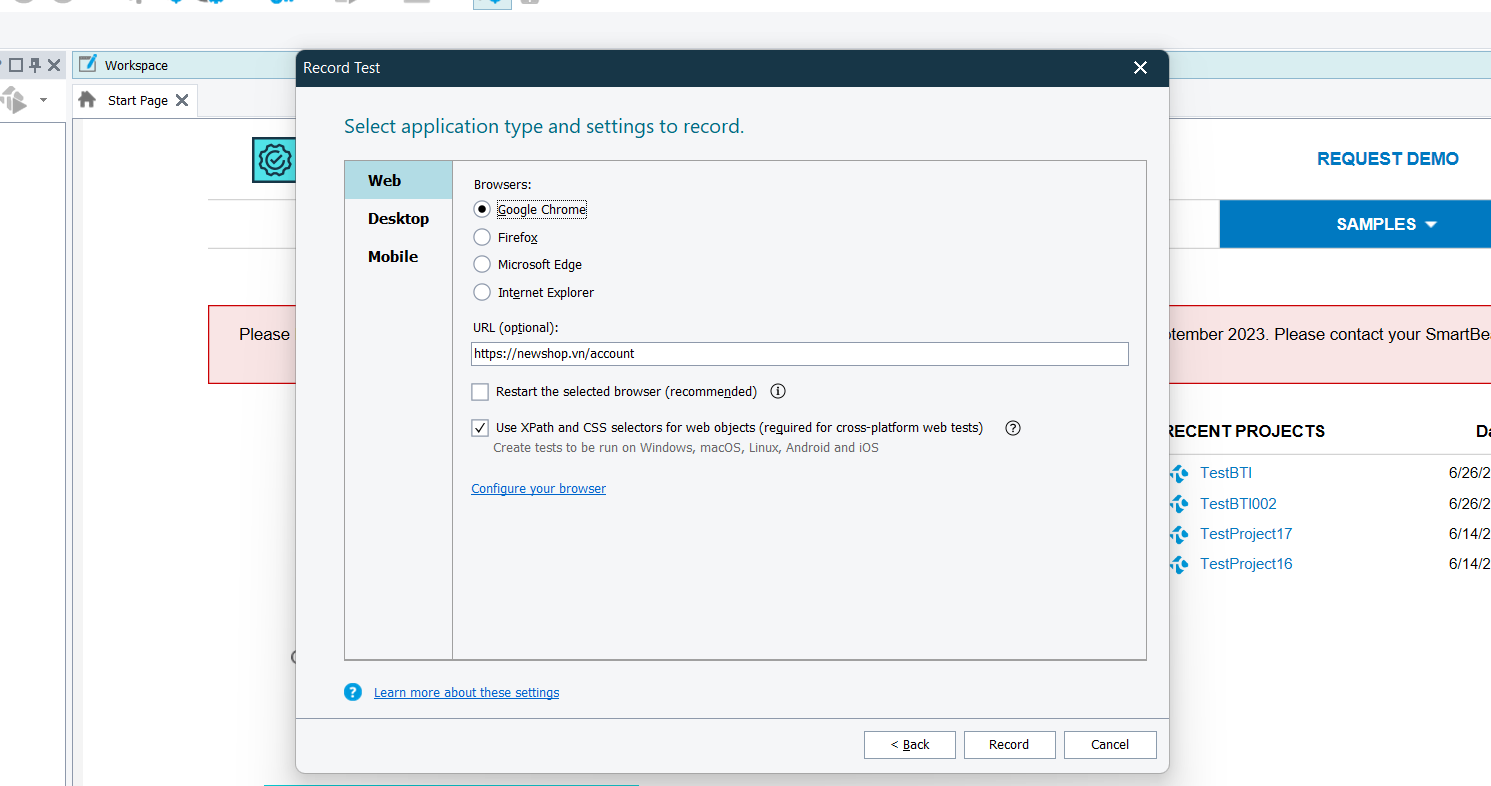
Hình 3. 6 TestCase quản lý tài khoản người dùng

***Kiểm thử phần mềm***

Các bước kiểm thử tự động với công cụ TestComplete:

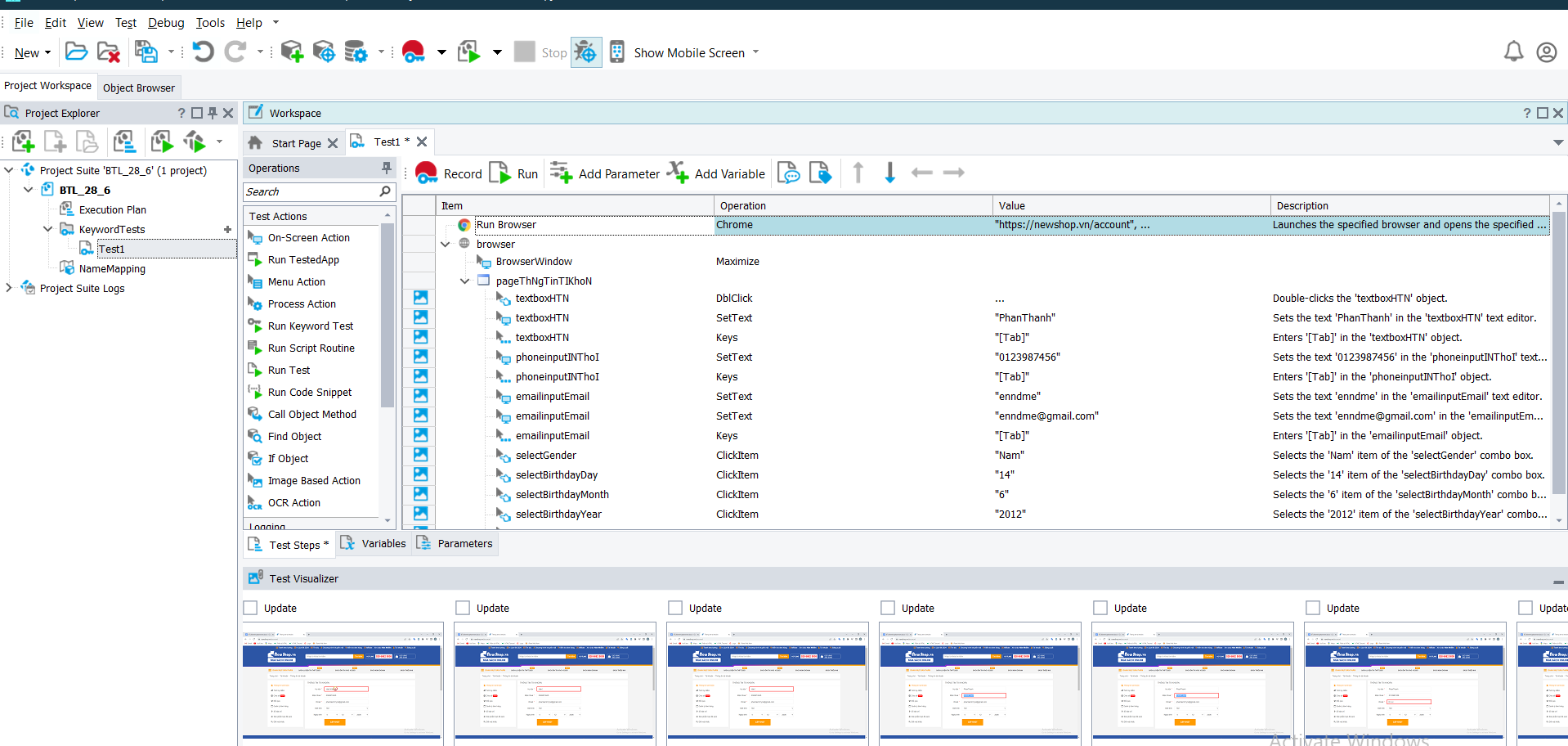
Ở Start Page chúng ta có thể chọn Record Test luôn Hoặc New Project đều sẽ tạo một project mới (Record Test sau khi tạo project xong thì sẽ quay luôn ) hoặc lựa chọn các project đã tạo sẵn rồi lại chọn Record Test để bắt đầu quay



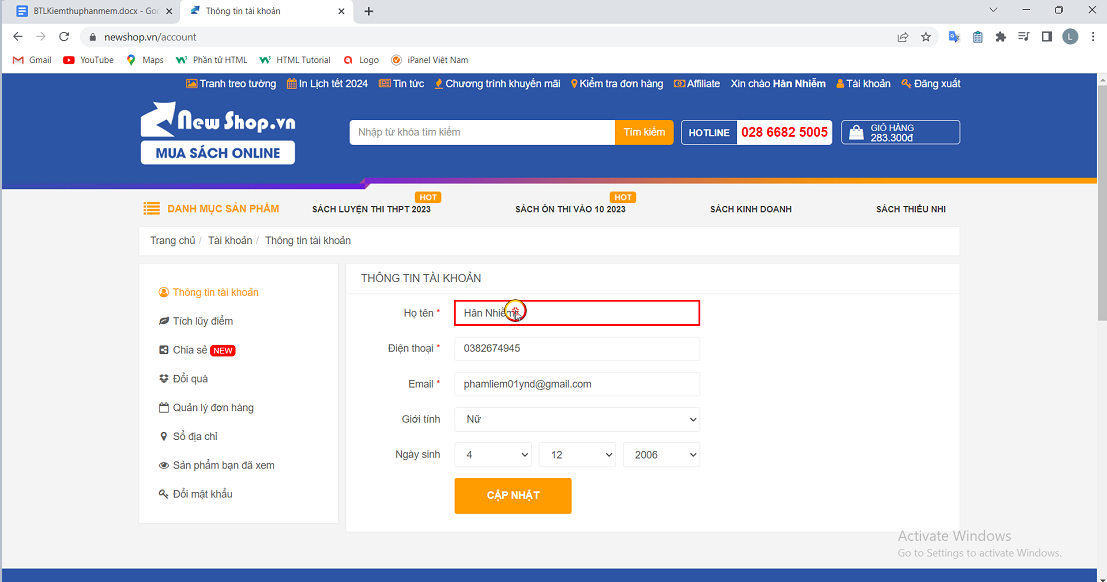


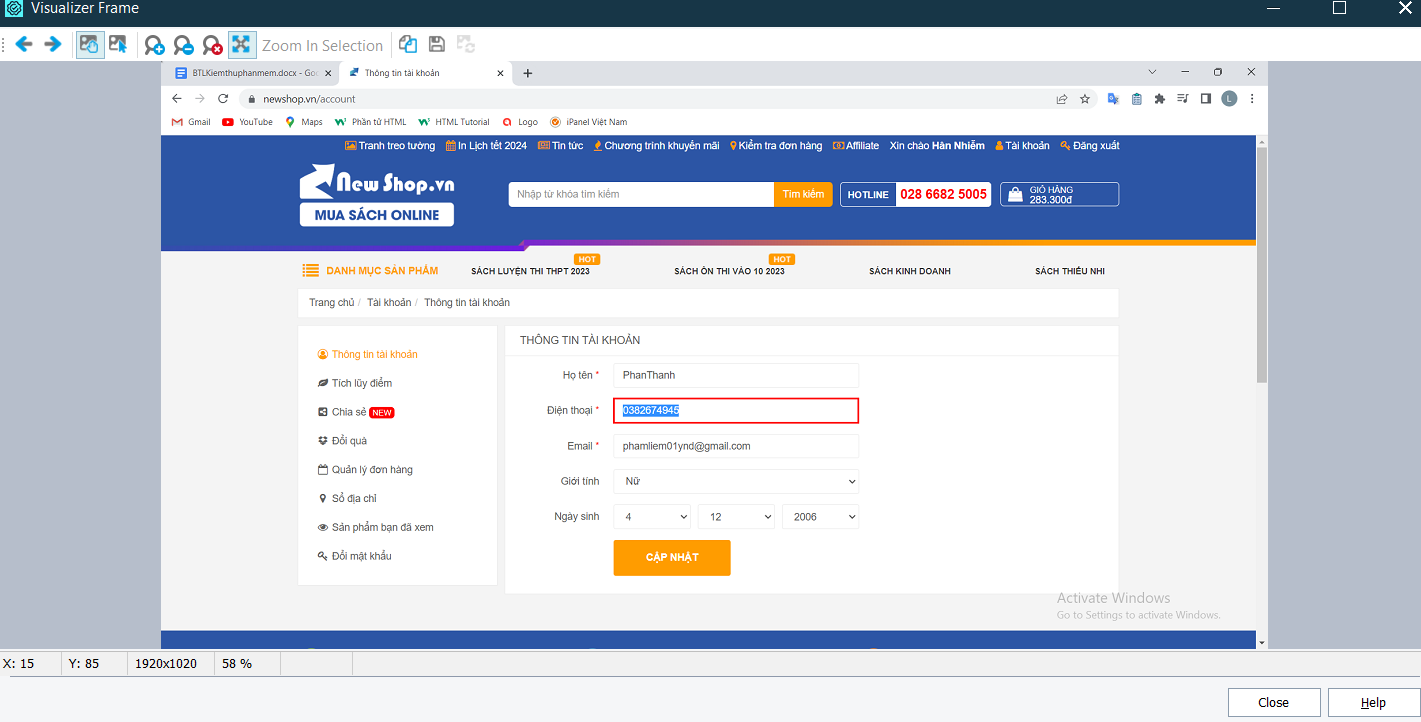
Sau khi hoàn thành xong chúng ta ấn Stop trên pop up của Test Complete

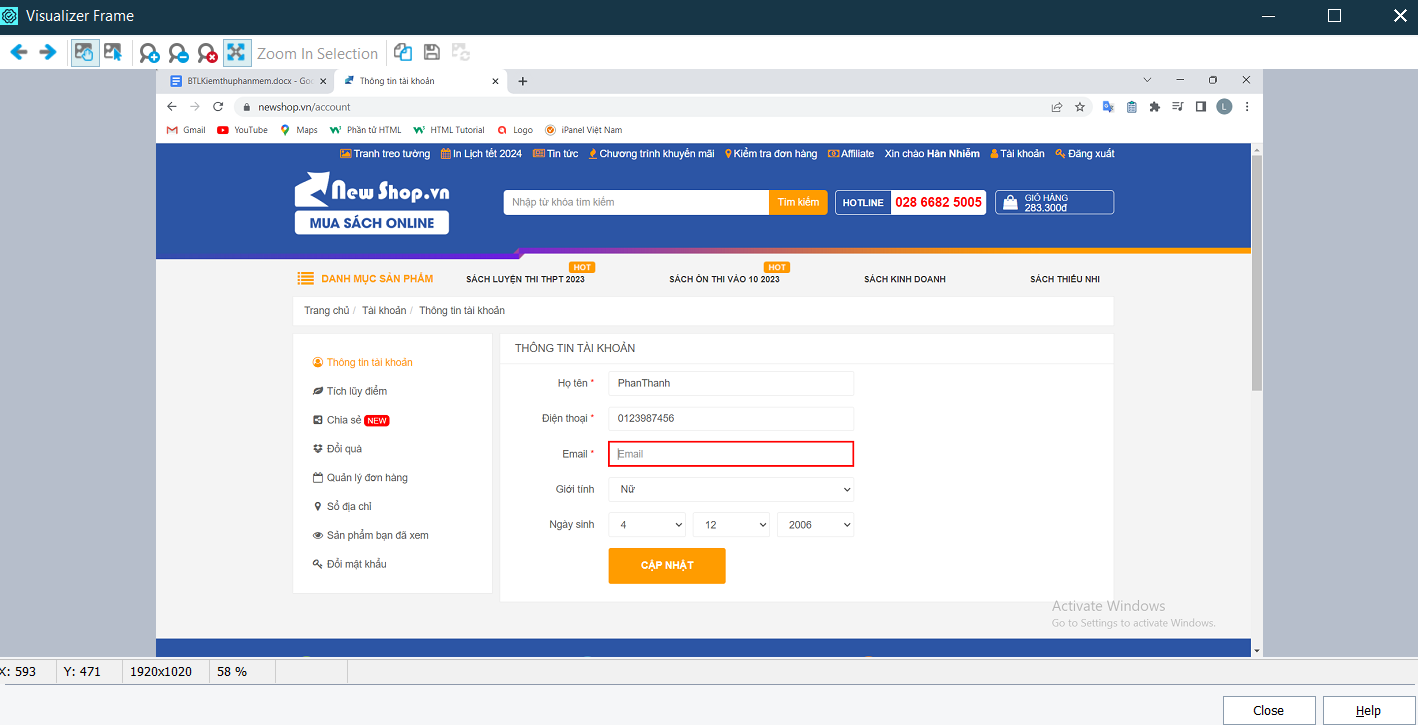
Phần mềm sẽ đưa chúng ta đến 1 màn hình hiển thị toàn bộ các thao tác chúng ta đã sử dụng cũng như tự động chụp lại màn hình chúng ta đã đi qua

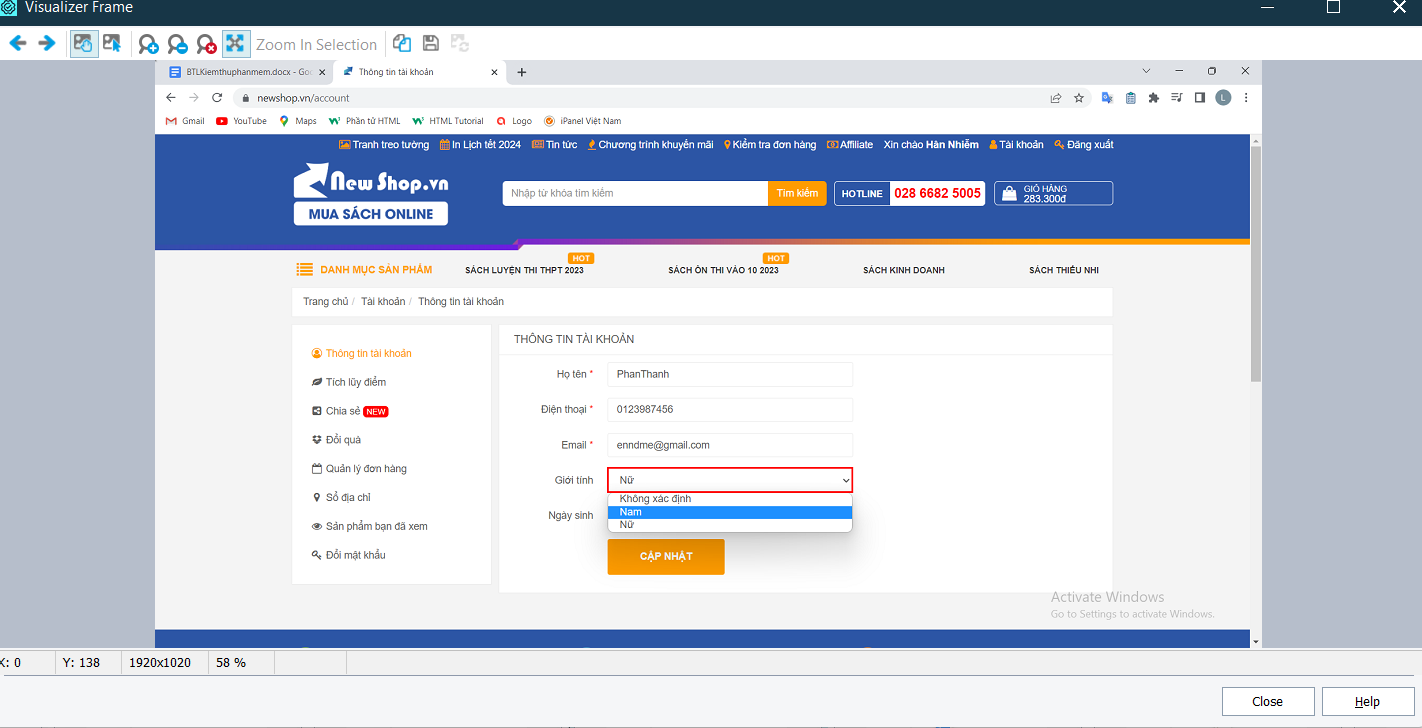


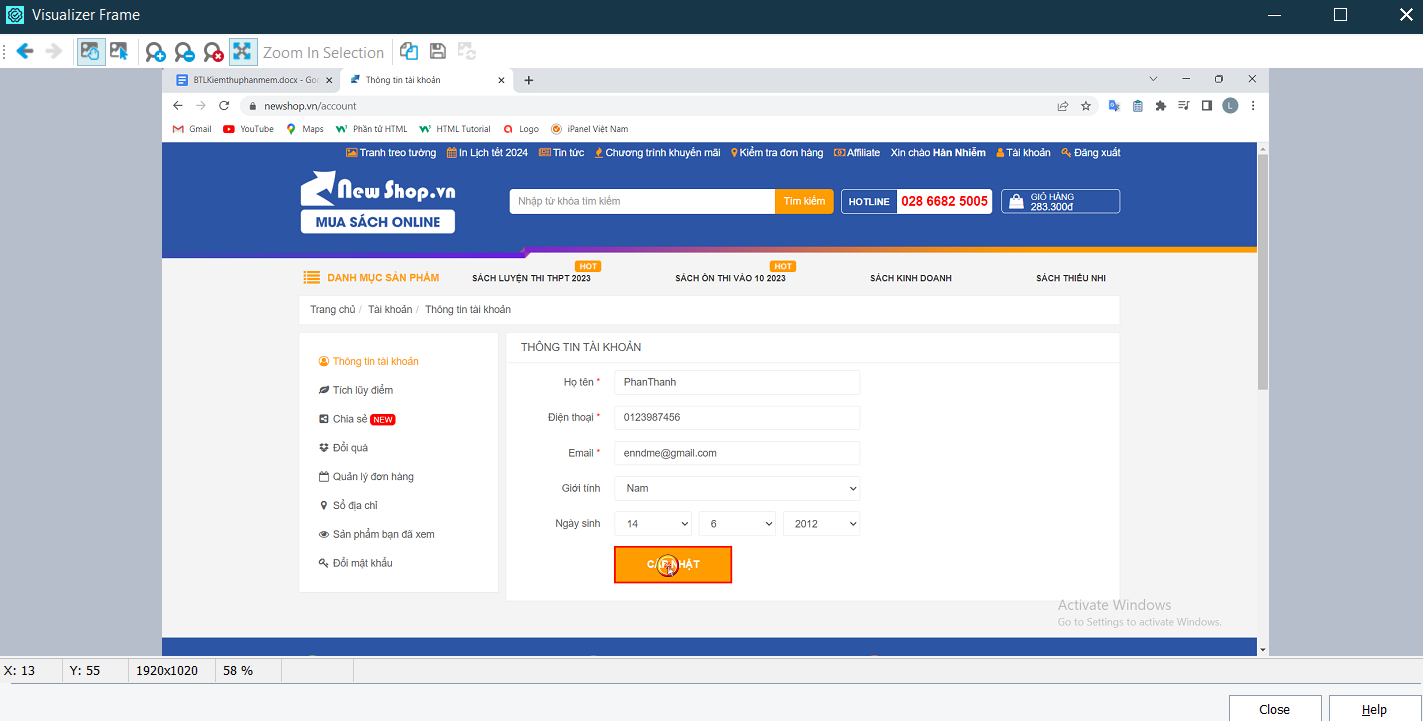
Màn hình được TestComplete ghi lại tự động ghi lại



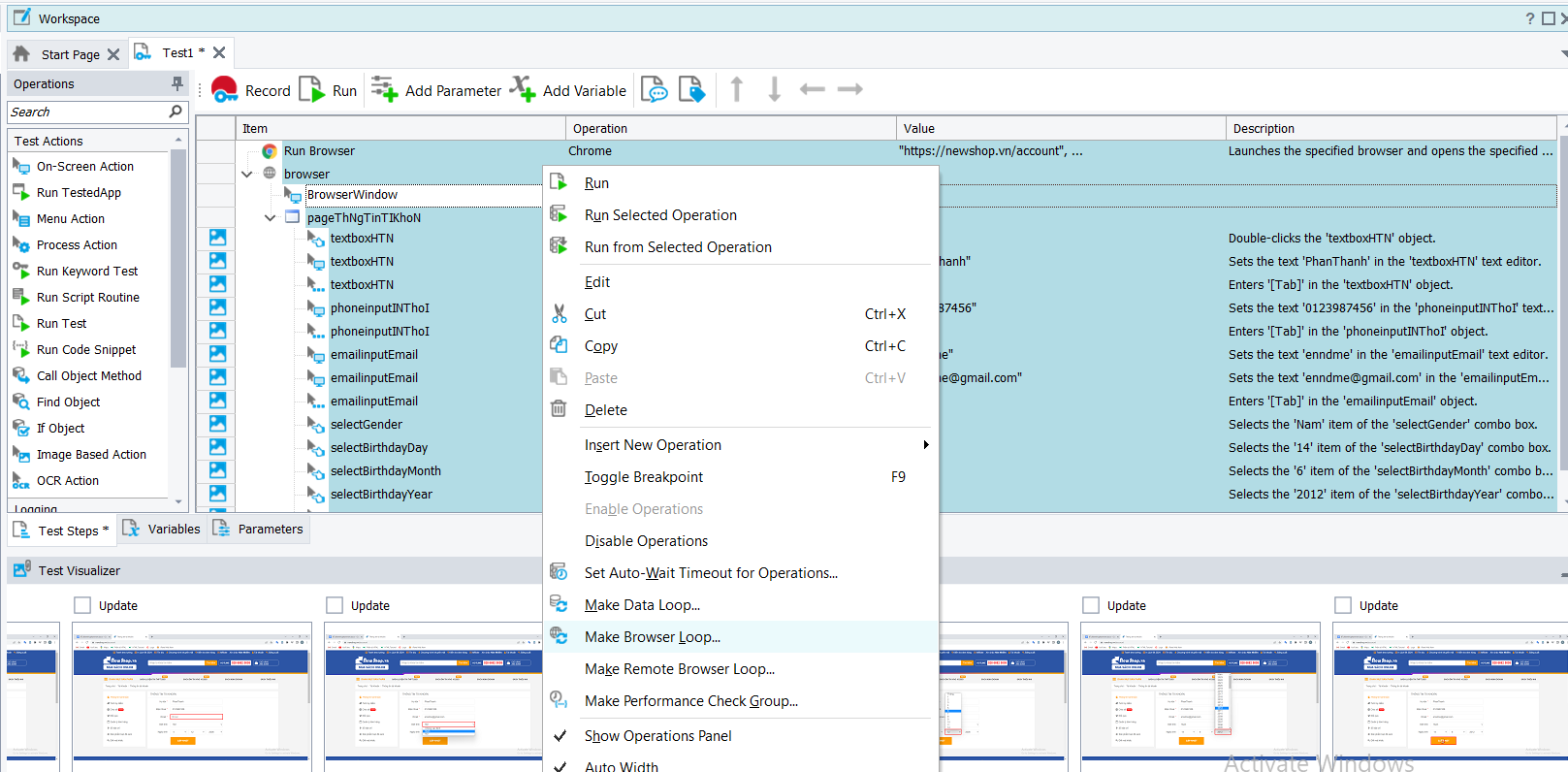
**

**

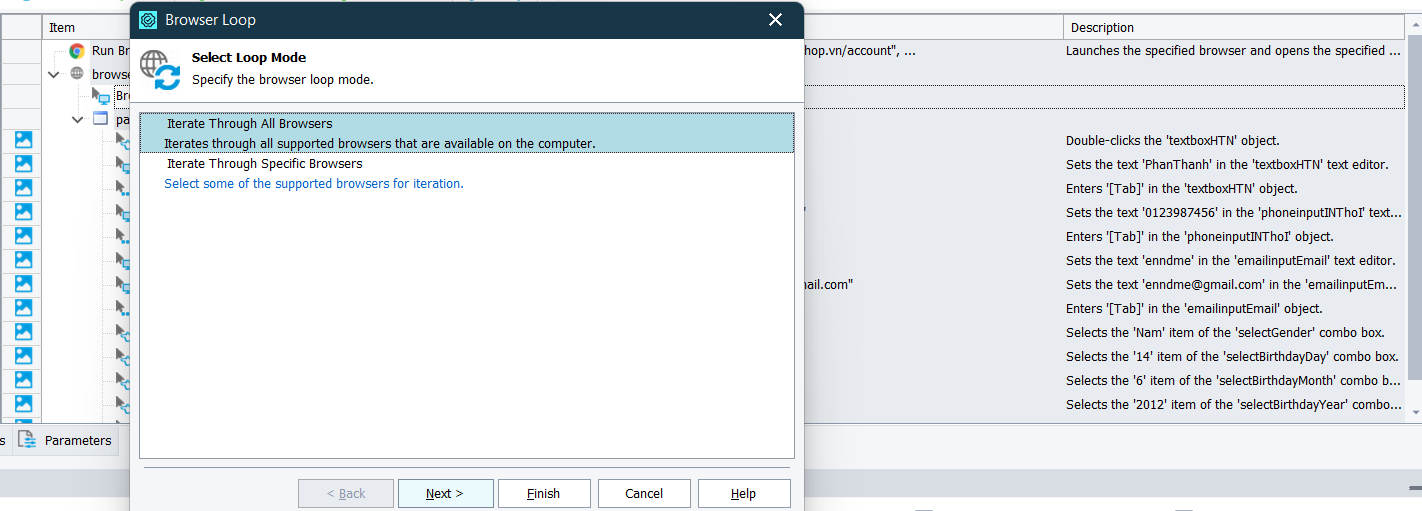
**

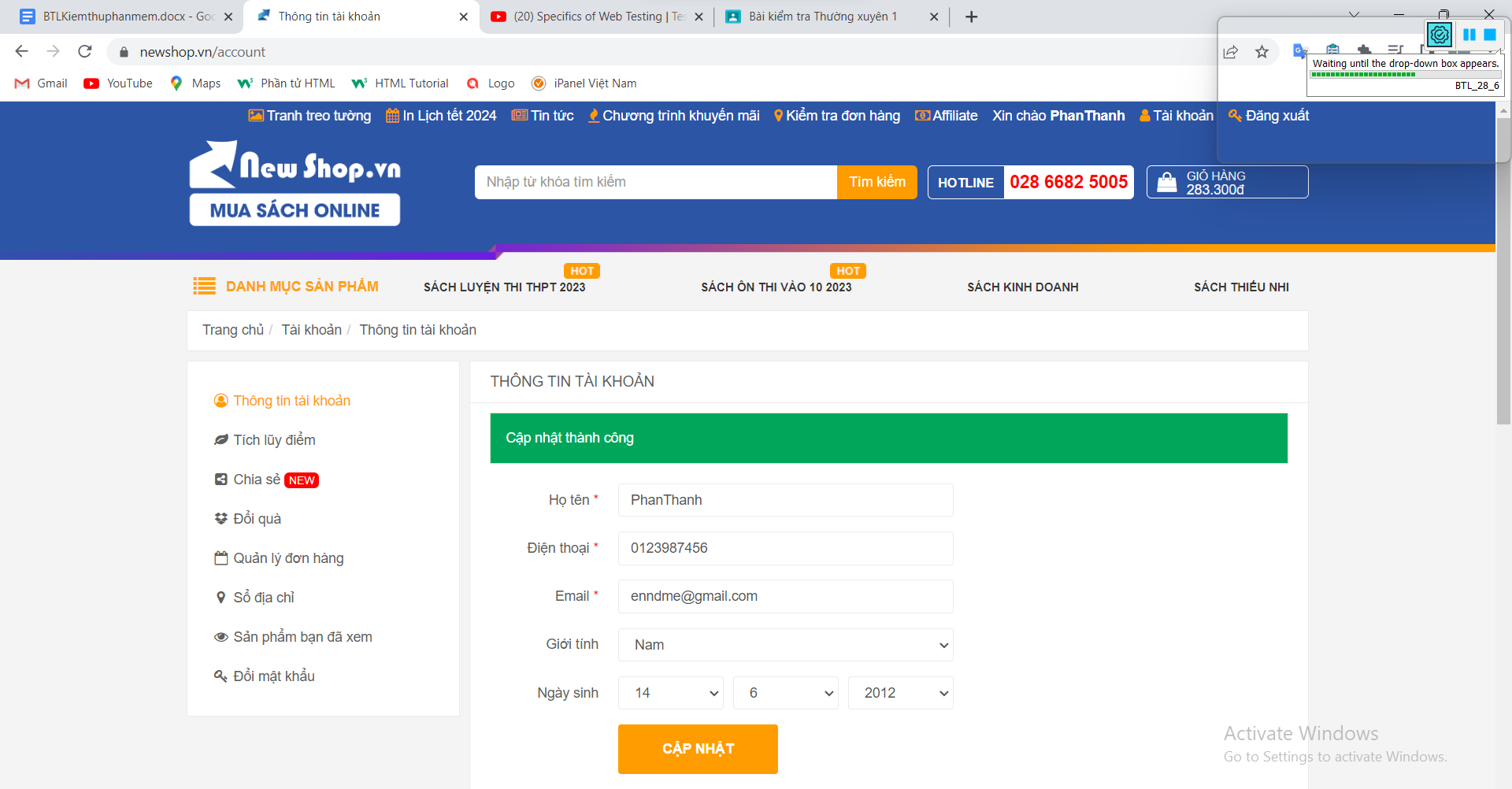
**

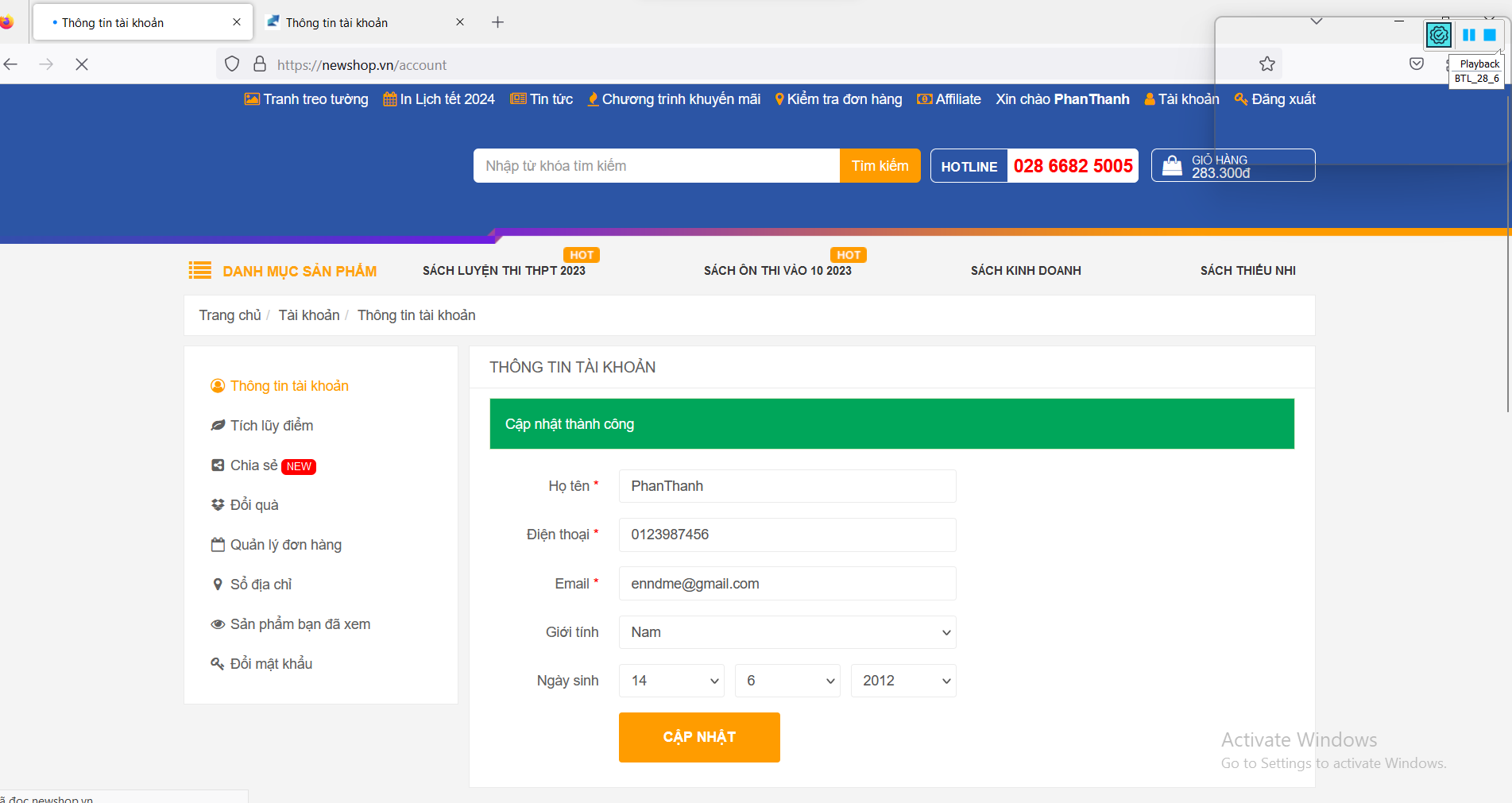
Để công cụ có thể thực hiện tự động vòng lặp trên nhiều trình duyệt ta thực hiện bội đen hình dưới và chọn Make Browser Loop

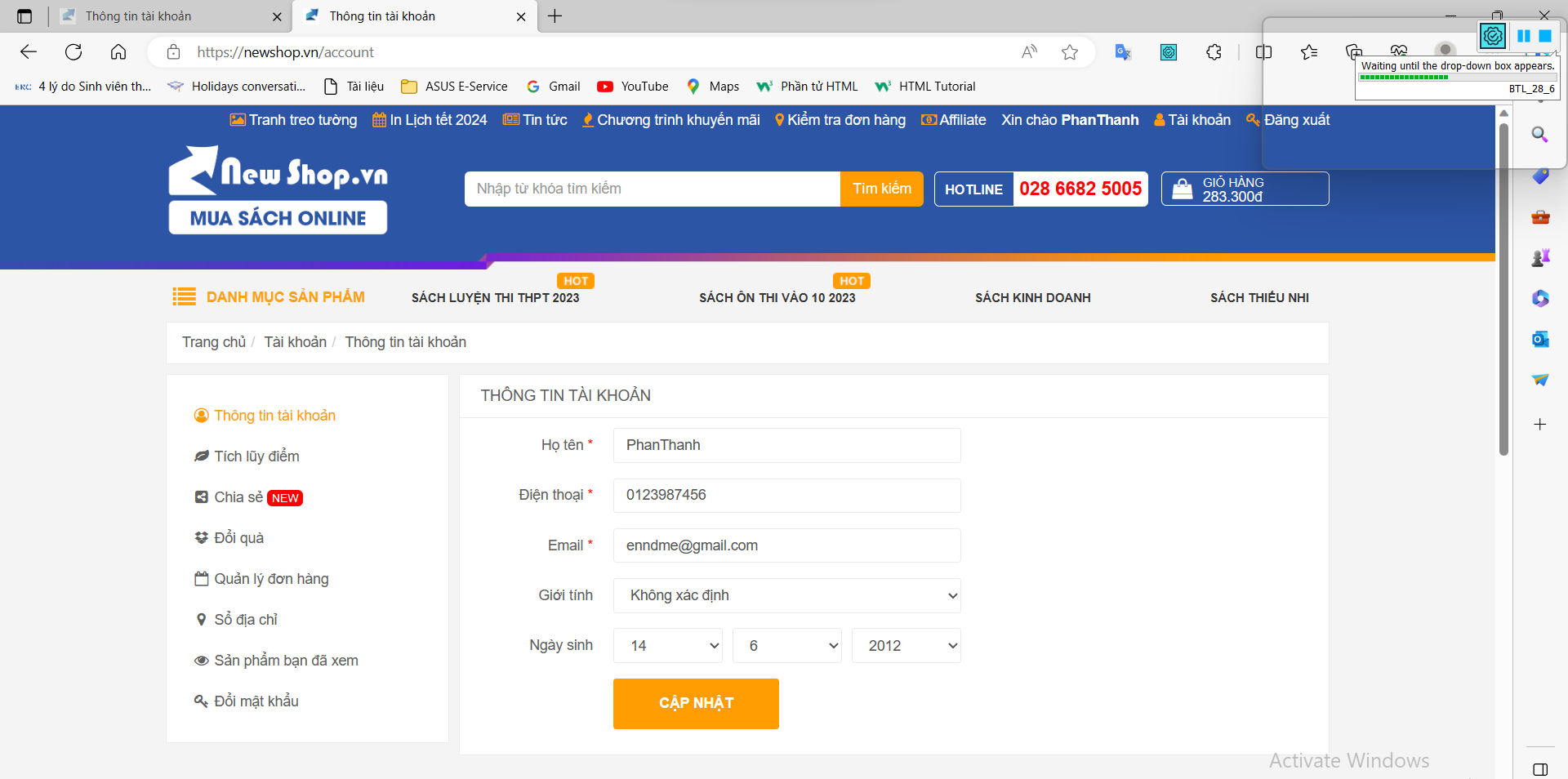


Sau khi chọn Make browser loop ta sẽ chọn dòng đầu tiên rồi click nút “finish”.

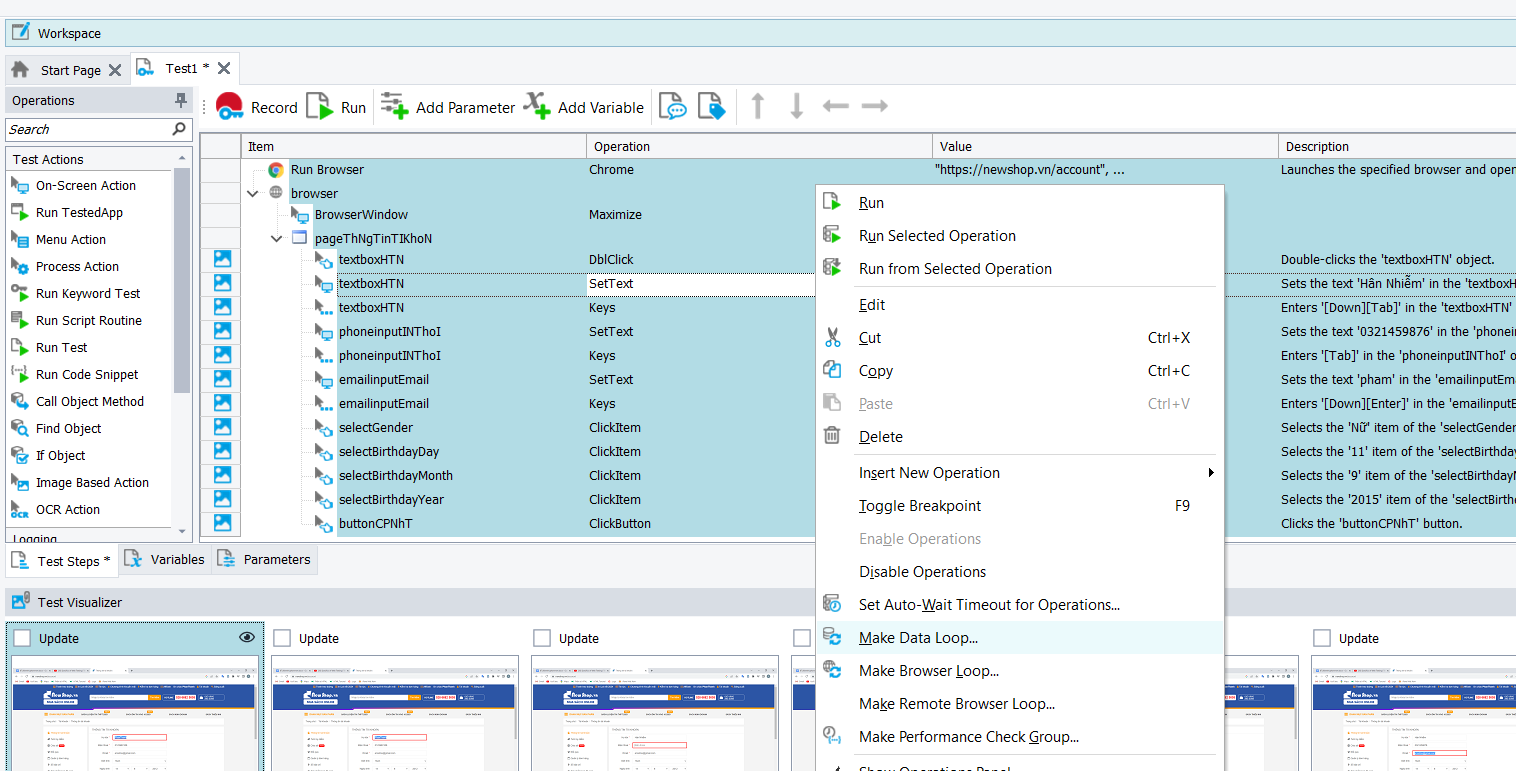


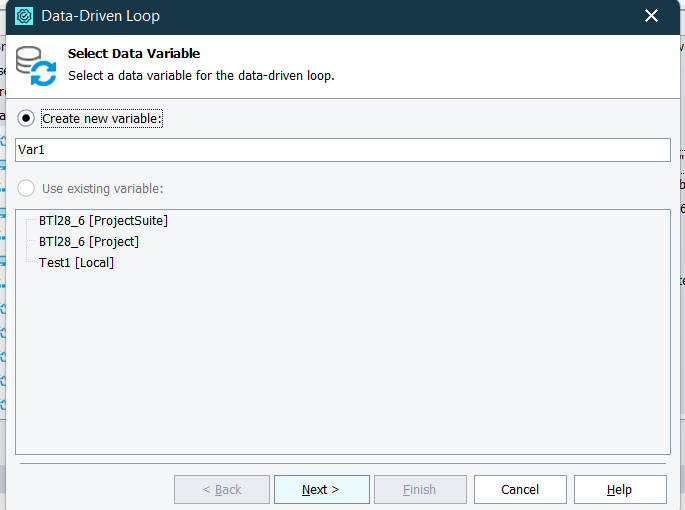
**



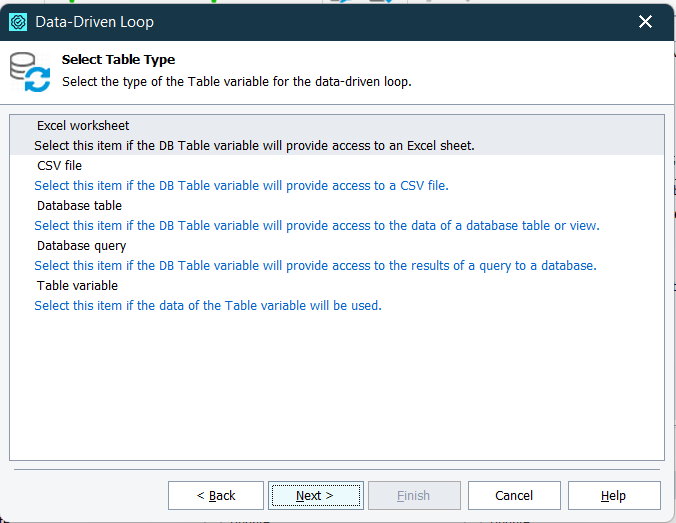
**

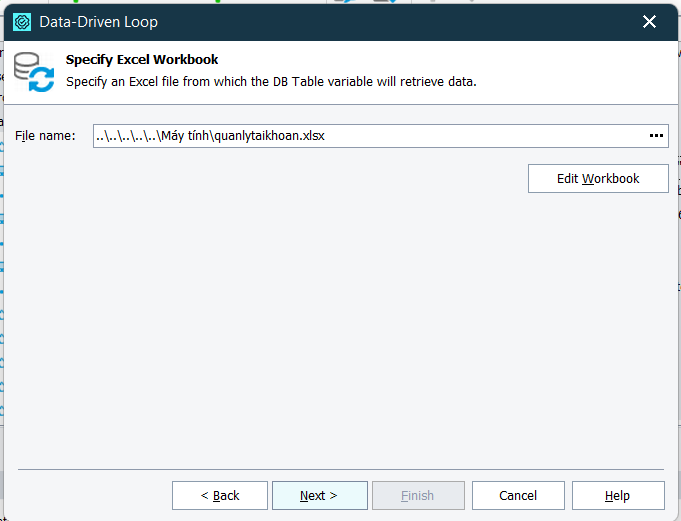
Để chèn dữ liệu cho công cụ để thực hiện vòng lặp chúng ta bôi đèn toàn bộ các thao tác rồi nhấp chuột phải chọn “Make data loop”



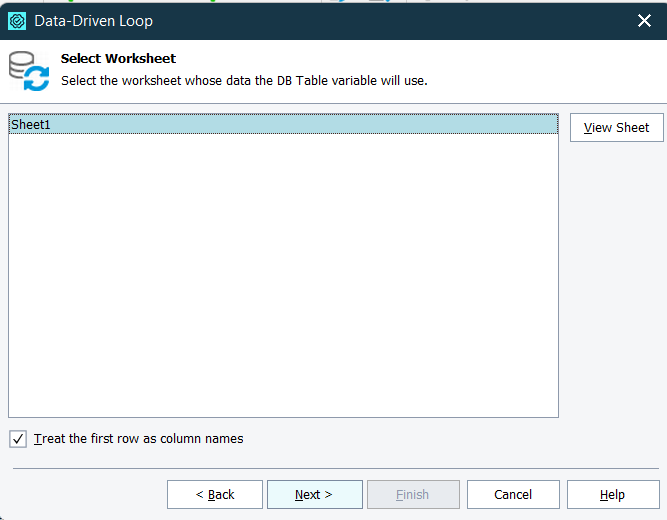
**

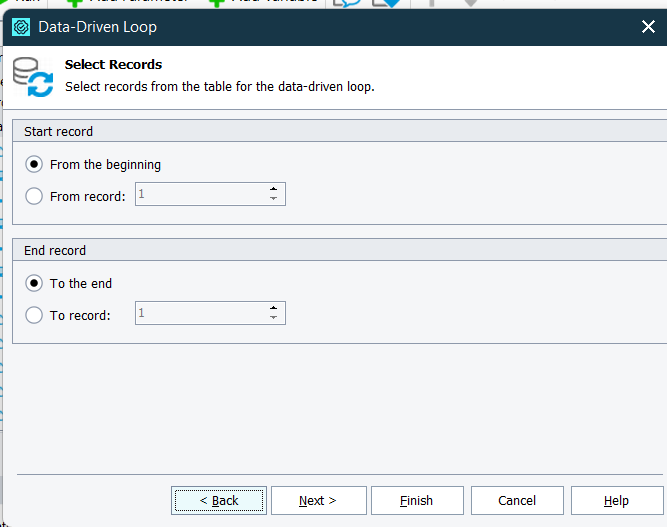
Tiếp theo ta chọn 1 loại dữ liệu để Test Complete nhập vào ( như test case trên thì dùng excel )



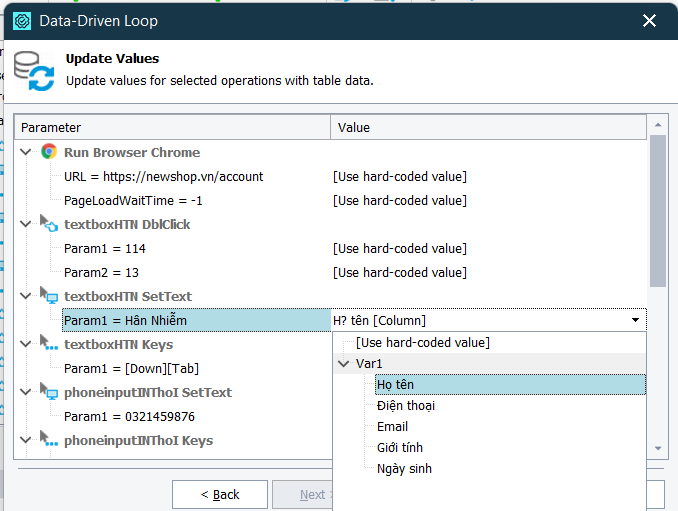
**

Chọn sheet1

**

**

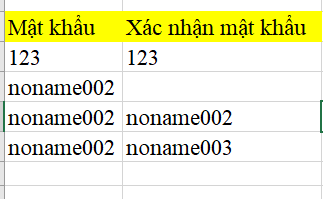
Chọn tham chiếu cho các thao tác

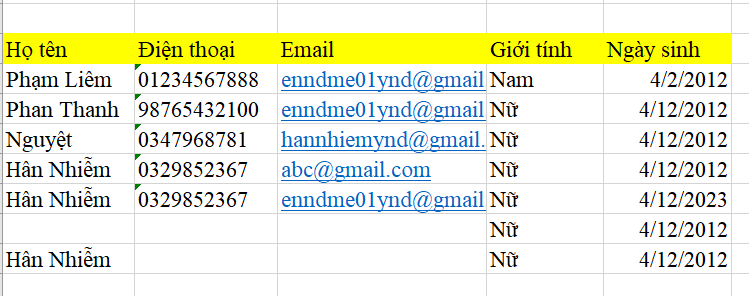


Sau đó click nút “finish” và run công cụ sẽ tự động thực hiện

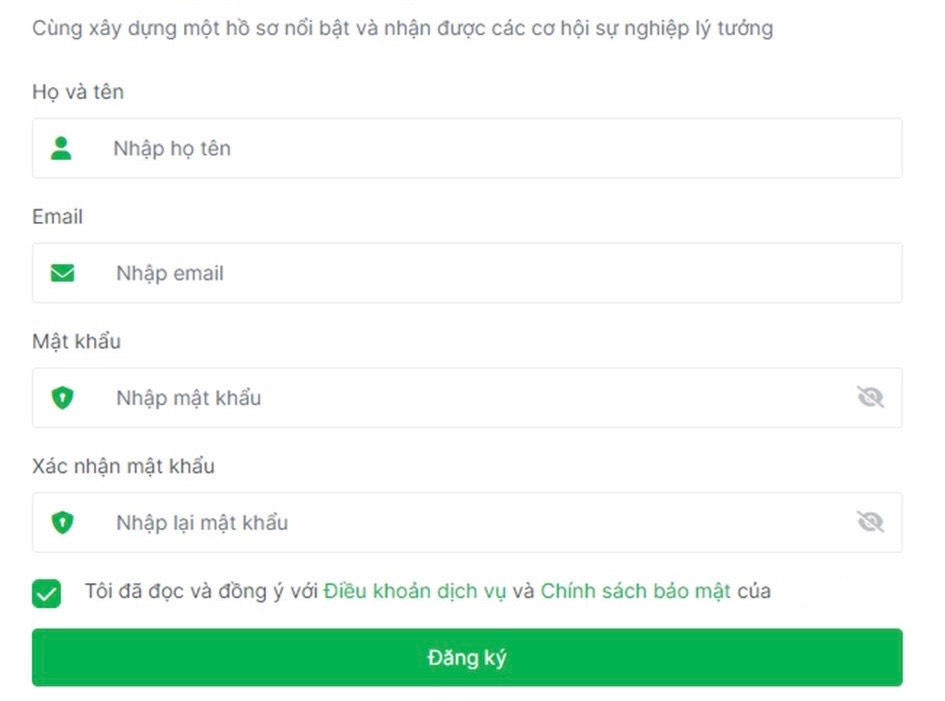
Phần thay đổi mật khẩu thực hiện tương tự.

Dữ liệu phần thay đổi mật khẩu và thay đổi thông tin tài khoản





3.3.4 Hồ Hải Hà - Đăng kí

**

Hình 3. 7Giao diện đăng kí

Mô tả giao diện và các điều kiện ràng buộc:

Đăng ký bao gồm Họ tên, Email, Mật khẩu, Nhập lại mật khẩu.

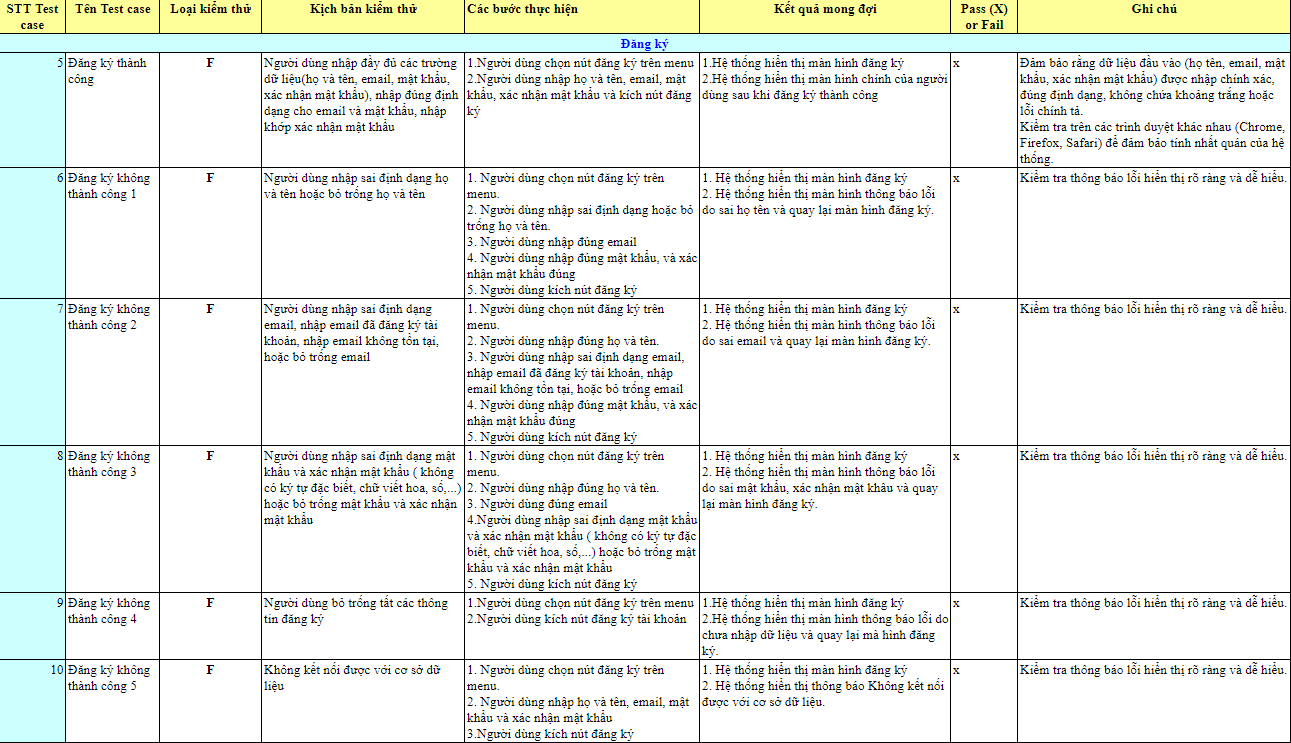
Điều kiện ràng buộc: Họ tên, Email, Mật khẩu, Nhập lại mật khẩu không được để trống và tối đa 255 ký tự.

Sử dụng các phương pháp kỹ thuật đã học để lấy cơ sở tìm testcase.

***Lưu đồ thuật toán:***

Sử dụng phương pháp phân hoạch tương đương

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Điều kiện đầu vào | Các lớp tương đương hợp lệ | ký hiệu đánh dấu | Các lớp tương đương không hợp lệ | Đánh dấu |
| Họ tên | 1.> 0 và <= 255 | v | 2.Để trống  3.> 255 | ✗ Độ dài không hợp lệ |
| Email | 4.> 0 và <= 255  5.Nhập đúng định dạng | v | 6.Để trống  7.Nhập sai định dạng | ✗ Độ dài không hợp lệ  ✗ Ký tự không hợp lệ |
| Mật khẩu | 8.> 0 và <= 255  9.Nhập đúng định dạng ( viết hoa, ký tự đặc biệt,...) | v | 10..Để trống  11.Nhập sai định dạng | ✗ Độ dài không hợp lệ  ✗ Ký tự không hợp lệ |
| Nhập lại mật khẩu | 12. Nhập lại đúng mật khẩu | v | 13. Để trống  14. Nhập lại sai mật khẩu | ✗ Ký tự không hợp lệ |

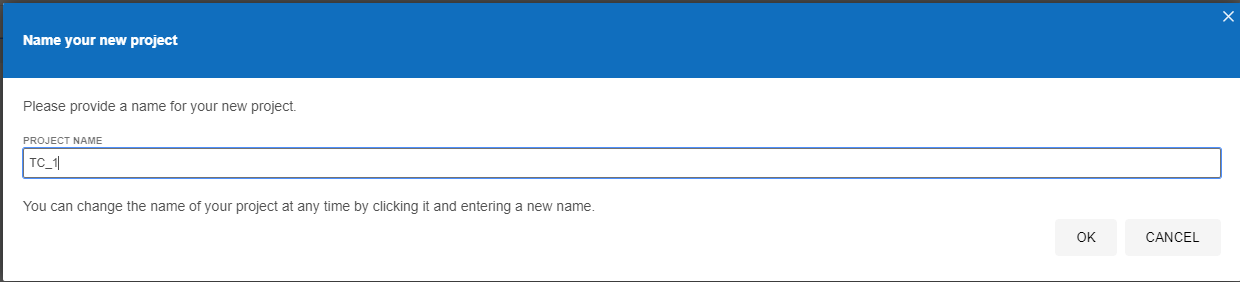
Danh sách test case:

Hình 3. 8 TestCase đăng kí

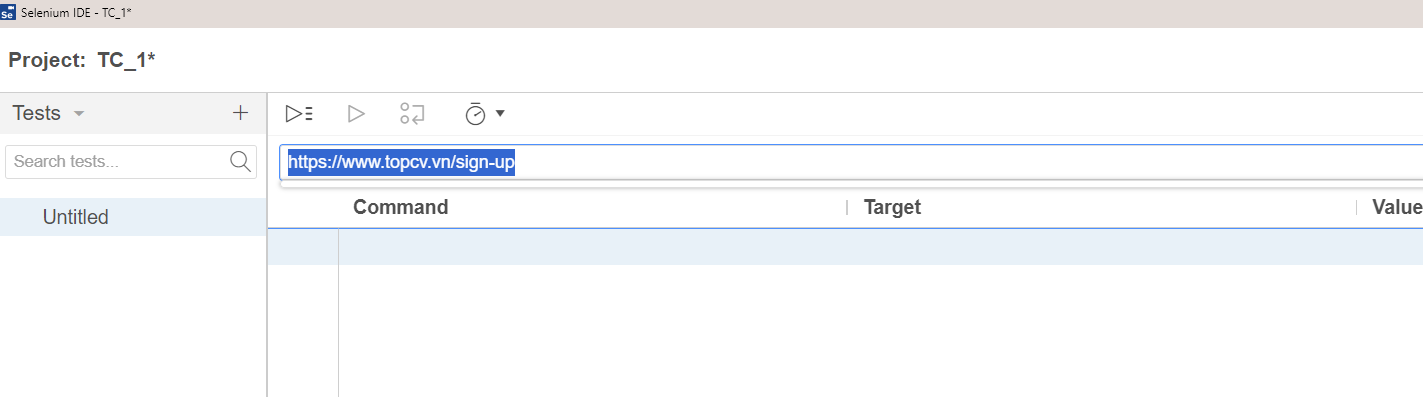
***Thực hiện kiểm thử***

Các bước thực hiện kiểm thử tự động: Công việc kiểm thử tự động được thực hiện bằng cách sử dụng Selenium IDE

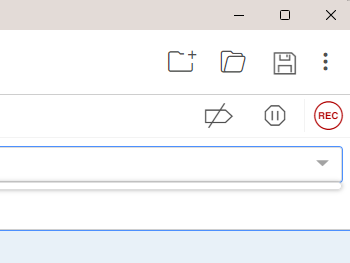
Bước 1: Tạo test mới



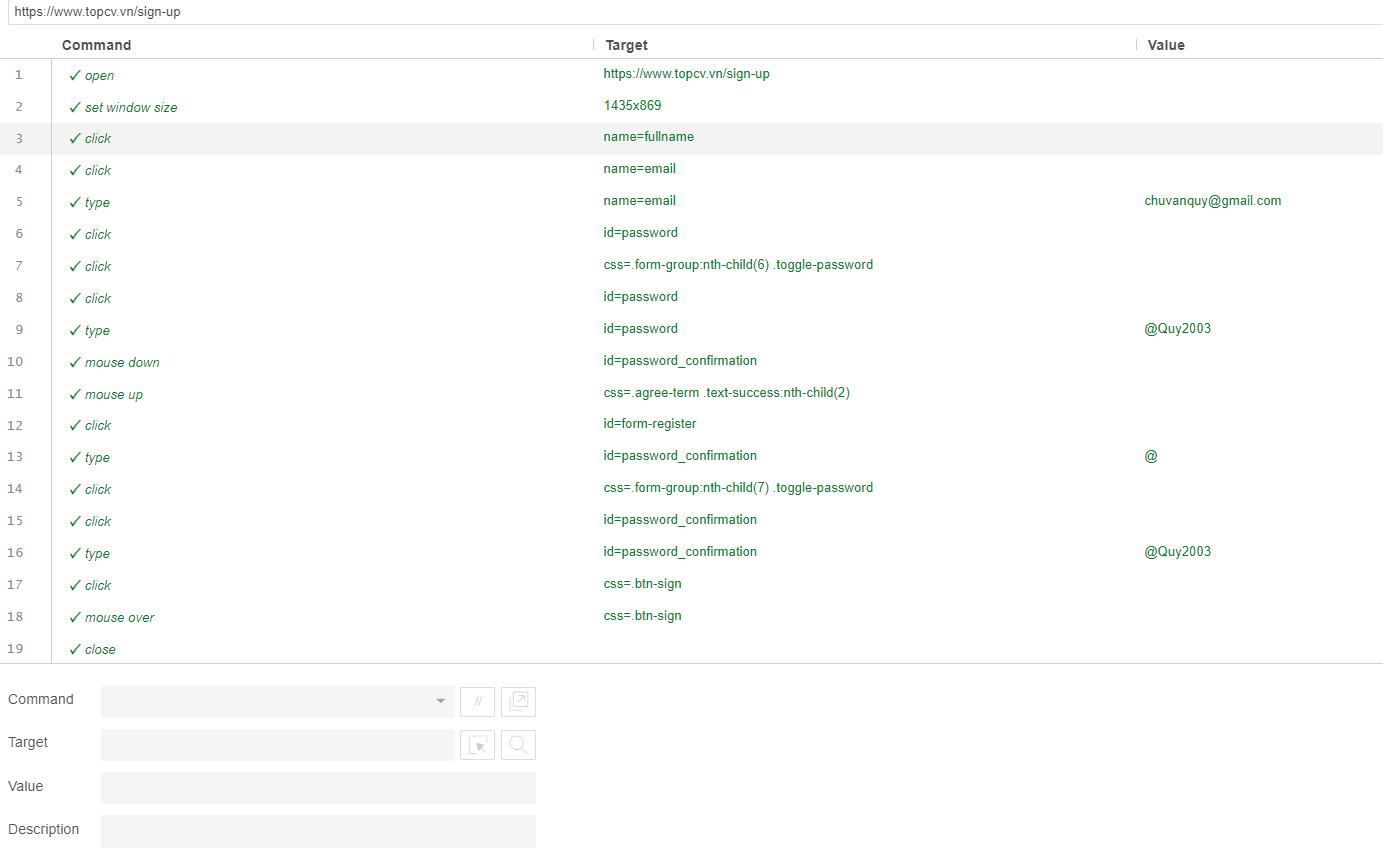
Bước 2: Nhập đường dẫn của trang web

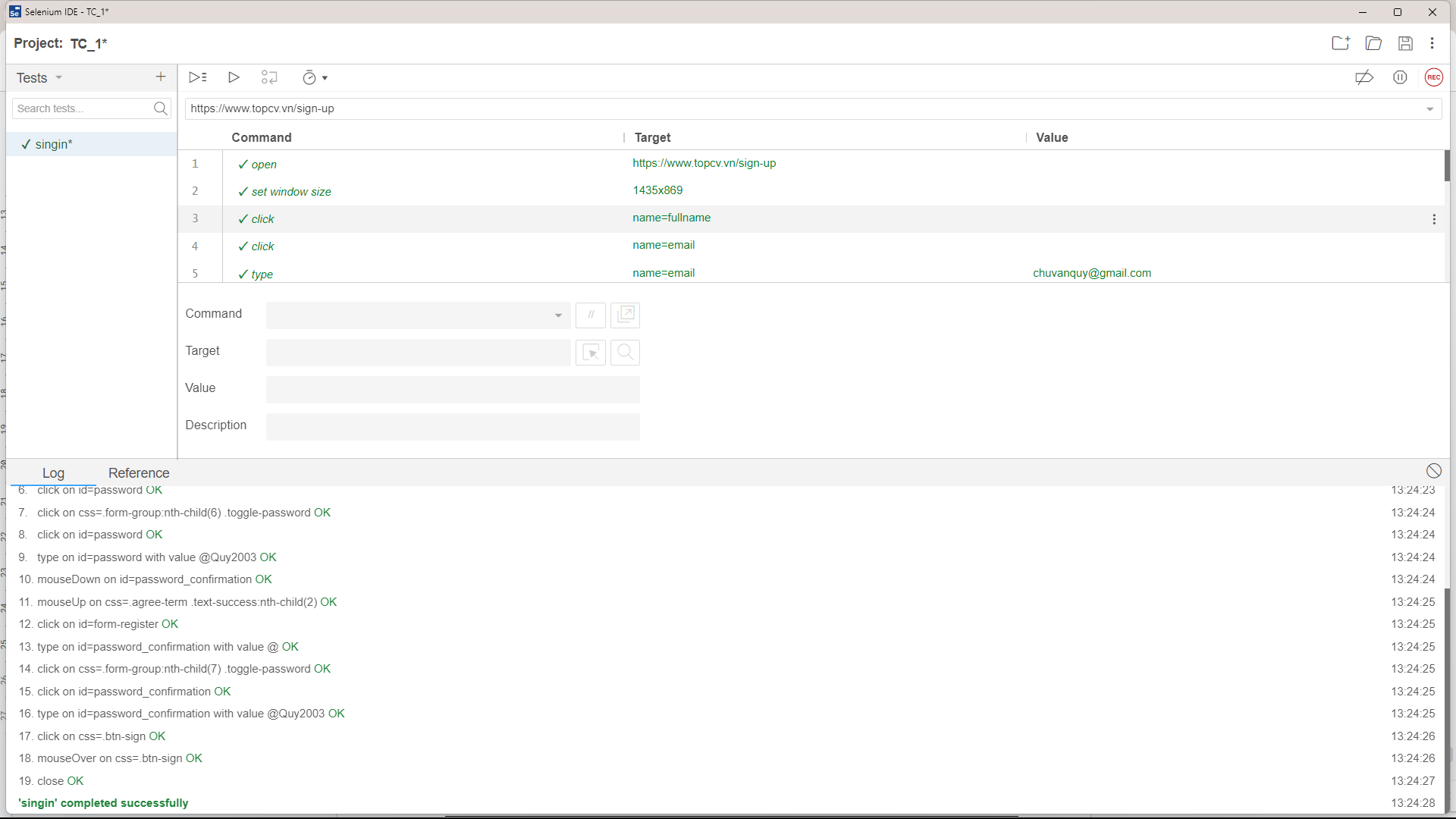


Bước 3: Chọn “REC” để bắt đầu viết kịch bản:



Ta có kịch bản kiểm thử:

Bước 4: Chọn “Run” trong Selenium để tiến hành kiểm thử.



=> Hệ thống thông báo kiểm thử thành công.

KẾT LUẬN

**Kết quả đạt được**

* Nắm rõ được khái niệm cơ bản về kiểm thử phần mềm.
* Biết về các loại kiểm thử, các phương pháp kiểm thử.

**Phương hướng phát triển**

* Tìm hiểu và lựa chọn các công cụ kiểm thử khác để so sánh và tìm ra công cụ tốt nhất để phục vụ cho mục đích của cá nhân hay doanh nghiệp.

TÀI LIỆU THAM KHẢO

[1]: <https://www.guru99.com/end-to-end-testing.html>

[2]: End To End Testcases In TestRigor:

<https://www.youtube.com/watch?v=9VPTyAQ-JLI&list=PL6flErFppaj3V4kreRE9x2M7MHHapXrJi&index=7>