

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI
KHOA CÔNG NGHỆ THÔNG TIN

=====***=====



BÁO CÁO BTL THUỘC HỌC PHẦN:
KIỂM THỬ PHẦN MỀM

Nghiên cứu công cụ kiểm thử tự động QTP
và ứng dụng

GVHD: Ths Hà Mạnh Đào
Nhóm - Lớp: 12 – KTPM2K12
Thành viên: Phạm Tuấn Sơn
Nguyễn Thanh Thiên
Hoàng Đình Thiện
Trần Lê Thịnh
Vũ Đức Vượng

Hà nội, Năm 2020

LỜI MỞ ĐẦU

Trong những năm gần đây, cùng với sự bùng nổ của ngành công nghệ thông tin, các sản phẩm của ngành công nghệ thông tin cũng phát triển không ngừng. Đặc biệt các sản phẩm phần mềm là nổi bật nhất, các sản phẩm phần mềm cũng cần có được chất lượng cao để đáp ứng nhu cầu. Kiểm thử phần mềm là một phần không thể thiếu để hoàn thiện sản phẩm phần mềm. Trong lĩnh vực kiểm thử phần mềm hiện có khá nhiều công cụ kiểm thử tự động thương mại nổi tiếng, phổ biến như QuickTest Professional, WinRunner, Rational Robot, SilkTest, JTest, ... Trong số đó, QuickTest Professional (QTP) khá tốt và mạnh, bao gồm nhiều chức năng điển hình của một công cụ kiểm thử tự động. QTP được dùng để kiểm thử chức năng (functional test) và cho phép thực hiện kiểm thử hồi qui (regression test) một cách tự động. Đây cũng là công cụ áp dụng phương pháp Keyword-Driven, một kỹ thuật scripting (lập trình trong kiểm thử tự động) hiện đại, cho phép kỹ sư kiểm thử phần mềm bổ sung test case bằng cách tạo tập tin (file) mô tả cho nó mà không cần phải chỉnh sửa hay bổ sung bất cứ script nào cả. Nó cũng phù hợp trong tình huống chuyển giao công việc mà người mới tiếp nhận chưa có thời gian hoặc không hiểu script vẫn có thể thực hiện kiểm thử phần mềm theo đúng yêu cầu.

Qua sự tìm hiểu và sự hướng dẫn của giảng viên chúng em đã nghiên cứu và thực hiện các kịch bản kiểm thử với công cụ QTP.

Nhóm chúng em xin cảm ơn giảng viên bộ môn “Kiểm thử phần mềm” – Hà Mạnh Đào - đã giao cho nhóm chúng em đề tài tìm hiểu này! Bài tập lớn của nhóm chúng em cũng không tránh khỏi những thiếu sót, mong thầy đưa ra cho chúng em những lời góp ý để chúng em rút kinh nghiệm cho những sản phẩm trong tương lai.

Chúng em xin trân thành cảm ơn!

Nhóm 12

MỤC LỤC

Chương 1. TỔNG QUAN VỀ KIỂM THỬ PHẦN MỀM.....	7
1.1 KIẾN THỨC KIỂM THỬ PHẦN MỀM CƠ SỞ	7
1.1.1 Chất lượng phần mềm và đảm bảo chất lượng phần mềm	7
1.1.2 Lỗi phần mềm.....	8
1.1.3 Khái niệm kiểm thử phần mềm.....	12
1.1.4 Mục tiêu của kiểm thử.....	12
1.1.5 Tầm quan trọng của kiểm thử phần mềm.....	14
1.1.6 Các nguyên tắc trong kiểm thử	14
1.1.7 Phân loại kiểm thử.....	15
1.1.8 Đối tượng thực hiện kiểm thử	16
1.1.9 Các hạn chế của kiểm thử	16
1.2 KỸ THUẬT KIỂM THỬ THỦ CÔNG VÀ TỰ ĐỘNG	17
1.2.1 Kỹ thuật kiểm thử thủ công (Manual testing).....	17
1.2.2 Kỹ thuật kiểm thử tự động (Automation testing).....	19
1.3 MỘT SỐ CÔNG CỤ KIỂM THỬ TỰ ĐỘNG	23
1.3.1 Selenium IDE	23
1.3.2 Selenium RC	23
1.3.3 Selenium WebDriver.....	23
1.3.4 Selenium Grid.....	24
1.3.5 Công cụ kiểm thử Jmeter.....	24
1.3.6 Katalon (katalon studio)	24
1.3.7 Junit	24
1.3.8 Appium.....	24

1.3.9	Công cụ kiểm thử LoadStorm	25
1.3.10	Công cụ kiểm thử QTP	25
1.3.11	Công cụ kiểm thử Robotium.....	25
1.3.12	Công cụ kiểm thử SOASTA CloudTest.....	25
CHƯƠNG 2. CÔNG CỤ KIỂM THỬ QTP (QUICK TEST PROFESSIONAL)		26
2.1	GIỚI THIỆU.....	26
2.2	ĐẶC ĐIỂM CỦA QTP	29
2.3	CÁC THÀNH PHẦN QUAN TRỌNG QTP.....	30
2.3.1	Action	30
2.3.2	DataTable	31
2.3.3	Object Repository (OR).....	31
2.3.4	Checkpoint.....	31
2.4	ƯU NHƯỢC ĐIỂM.....	31
2.4.1	Ưu điểm.....	31
2.4.2	Nhược điểm.....	32
2.5	QUÁ TRÌNH CÀI ĐẶT	32
2.6	MÔ TẢ MÔI TRƯỜNG LÀM VIỆC	36
2.7	CÁC KỸ THUẬT THỰC HIỆN KIỂM THỬ CHI TIẾT	38
Chương 3. MỘT SỐ KỊCH BẢN KIỂM THỬ VÀ ĐÁNH GIÁ		40
3.1	KỊCH BẢN KIỂM THỬ SỐ 1	40
3.2	KỊCH BẢN TEST SỐ 2	46
3.3	KỊCH BẢN TEST SỐ 3	57

Chương 1. TỔNG QUAN VỀ KIỂM THỬ PHẦN MỀM

1.1 KIẾN THỨC KIỂM THỬ PHẦN MỀM CƠ SỞ

1.1.1 Chất lượng phần mềm và đảm bảo chất lượng phần mềm

1.1.1.1 Định nghĩa chất lượng phần mềm

- Có rất nhiều định nghĩa về chất lượng phần mềm được đưa ra bởi các tổ chức, cá nhân khác nhau. Trong phạm vi của bài viết này trình bày một số định nghĩa tiêu biểu.

*Định nghĩa theo IEEE (1991):

- **Định nghĩa 1:** Chất lượng phần mềm là một mức độ mà một hệ thống, thành phần hệ thống hay tiến trình đáp ứng được yêu cầu đã được đặc tả.
- **Định nghĩa 2:** Chất lượng phần mềm là mức độ mà một hệ thống, thành phần hệ thống hay tiến trình đáp ứng được yêu cầu và sự mong đợi của khách hàng hay người sử dụng.

*Phân tích hai quan điểm của IEEE:

- Theo quan điểm thứ nhất của IEEE: chúng ta sẽ bị phụ thuộc quá nhiều vào tài liệu đặc tả của yêu cầu, dẫn đến nếu việc xác định yêu cầu bị sai, thiếu thì một phần mềm được làm đúng với đặc tả chưa chắc đã là một phần mềm có chất lượng.
- Theo quan điểm thứ hai của IEEE: khách hàng đôi khi không có kiến thức về công nghệ, họ có thể đưa ra những mong muốn hết sức vô lý và có thể thay đổi yêu cầu với phần mềm nhiều lần, thậm chí thay đổi ngay

trong giai đoạn cuối. Điều này gây nhiều khó khăn cho việc phát triển phần mềm.

***Định nghĩa theo Pressman:** Chất lượng phần mềm là sự phù hợp của các yêu cầu cụ thể về hiệu năng và chức năng, các tiêu chuẩn phát triển phần mềm được ghi lại rõ ràng bằng tài liệu với các đặc tính ngầm định của tất cả các phần mềm được phát triển chuyên nghiệp.

Định nghĩa của Pressman đề xuất ba yêu cầu với chất lượng phần mềm phải được đáp ứng khi phát triển phần mềm:

- Các yêu cầu chức năng rõ ràng là nhân tố chính quyết định chất lượng đầu ra của phần mềm.
- Các tiêu chuẩn chất lượng phần mềm sẽ được nói đến trong hợp đồng.
- Các đặc tính ngầm định cần được đáp ứng trong quá trình phát triển cho dù không được nói đến rõ ràng trong hợp đồng.

1.1.1.2 Định nghĩa đảm bảo chất lượng phần mềm

- Định nghĩa theo Daniel Galin: Đảm bảo chất lượng phần mềm (Software Quality Assure) là một tập hợp các hành động cần thiết được lên kế hoạch một cách hệ thống để cung cấp đầy đủ niềm tin rằng quá trình phát triển phần mềm phù hợp để thành lập các yêu cầu chức năng kỹ thuật cũng như các yêu cầu quản lý theo lịch trình và hoạt động trong giới hạn ngân sách.

1.1.2 Lỗi phần mềm

1.1.2.1 Định nghĩa lỗi phần mềm và phân loại lỗi phần mềm

- Định nghĩa lỗi phần mềm: Có rất nhiều định nghĩa về lỗi phần mềm nhưng có thể hiểu và phát biểu một cách đơn giản thì "Lỗi phần mềm là sự không khớp giữa chương trình và đặc tả của nó".

- Dựa vào định nghĩa, ta có thể phân loại lỗi phần mềm thành 3 dạng:
 - Lỗi sai: Sản phẩm phần mềm được xây dựng khác với đặc tả.
 - Lỗi thiếu: Các yêu cầu của sản phẩm phần mềm đã có trong đặc tả nhưng lại không có trong sản phẩm thực tế.
 - Lỗi thừa: Sản phẩm thực tế có những tính năng không có trong tài liệu đặc tả.

1.1.2.2 Các nguyên nhân gây lỗi phần mềm

- Lỗi phần mềm có thể đến từ nhiều nguyên nhân khác nhau, trong đó có cả các nguyên nhân chủ quan và các nguyên nhân khách quan. Dưới đây là chín nguyên nhân chủ yếu gây ra lỗi phần mềm:

***Định nghĩa các yêu cầu bị lỗi:** Những lỗi trong việc xác định yêu cầu thường nằm ở phía khách hàng. Một số lỗi thường gặp là: định nghĩa sai yêu cầu, lỗi không hoàn chỉnh, thiếu các yêu cầu quan trọng hoặc là quá chú trọng các yêu cầu không thật sự cần thiết.

***Các lỗi trong giao tiếp giữa khách hàng và nhà phát triển:** Hiểu lầm trong giao tiếp giữa khách hàng và nhà phát triển cũng là nguyên nhân gây lỗi. Những lỗi này thường xuất hiện trong giai đoạn đầu của dự án. Một số lỗi hay gặp phải: hiểu sai chỉ dẫn trong tài liệu yêu cầu, hiểu sai thay đổi khi khách hàng trình bày bằng lời nói và tài liệu, hiểu sai về phản hồi và thiếu quan tâm đến những đề cập của khách hàng.

- Giải pháp khắc phục: Cần có ủy ban liên kết giữa khách hàng và nhà cung cấp để tránh lỗi trong giao tiếp. Ủy ban do quản trị dự án đứng đầu và khách hàng phải giới thiệu những người hiểu về mặt nghiệp vụ vào ủy ban đó.

***Sai lệch có chủ ý với các yêu cầu phần mềm:** Trong một số trường hợp các nhà phát triển cố tình làm sai lệch các yêu cầu trong tài liệu đặc tả. Nguyên nhân của việc này đến từ các áp lực thời gian, ngân sách, hay cố tình sử dụng lại các mô-đun từ các dự án trước mà chưa phân tích đầy đủ những thay đổi để thích nghi với các yêu cầu mới.

- Giải pháp khắc phục: Dựa trên những thống kê để quyết định xem giải pháp như thế nào, sắp xếp ưu tiên xem bỏ được yêu cầu nào hay sử dụng lại được mô-đun nào.

***Các lỗi thiết kế logic:** Lỗi phần mềm xảy ra trong quá trình các chuyên gia thiết kế hệ thống, các kiến trúc sư hệ thống, kỹ sư phần mềm, các nhà phân tích xây dựng phần mềm theo yêu cầu. Các lỗi điển hình bao gồm:

- Định nghĩa các yêu cầu phần mềm bằng các thuật toán sai.
- Quy trình định nghĩa có chứa trình tự lỗi.
- Sai sót trong các định nghĩa biên như > 3 hay ≥ 3 .
- Thiếu sót các trạng thái hệ thống phần mềm được yêu cầu.

***Các lỗi lập trình:** Có rất nhiều lý do dẫn đến việc các lập trình viên gây ra các lỗi lập trình. Những lý do này bao gồm: sự hiểu sai các tài liệu thiết kế, ngôn ngữ; sai sót trong ngôn ngữ lập trình; sai sót trong việc áp dụng các công cụ phát triển; sai sót trong lựa chọn dữ liệu...

***Không tuân thủ theo các tài liệu hướng dẫn và tiêu chuẩn lập trình:** Các lỗi phần mềm có thể đến từ việc không tuân thủ các tài liệu và tiêu chuẩn lập trình của các tổ chức phát triển phần mềm.

***Thiếu sót trong quá trình kiểm thử:** Lỗi phần mềm có thể đến từ chính quá trình kiểm thử khi mà người kiểm thử để lọt lỗi.

Những lỗi này đến từ các nguyên nhân sau đây:

- Kế hoạch kiểm thử chưa hoàn chỉnh, để sót yêu cầu cần kiểm thử.
- Lỗi trong tài liệu và báo cáo kiểm thử.
- Việc sửa chữa các lỗi được phát hiện không hoàn chỉnh do áp lực thời gian hay do thiếu cẩn thận.

Giải pháp: Lên kế hoạch kiểm thử cụ thể tại giai đoạn đầu của dự án.

***Các lỗi thủ tục:** Các thủ tục hướng dẫn cho người sử dụng tại từng bước của tiến trình. Chúng có tầm quan trọng đặc biệt trong các hệ thống phần mềm phức tạp mà các tiến trình được thực bằng nhiều bước, mỗi bước có thể có nhiều kiểu dữ liệu và cho phép kiểm tra các kết quả trung gian. Các lỗi có thể đến từ việc viết các thủ tục.

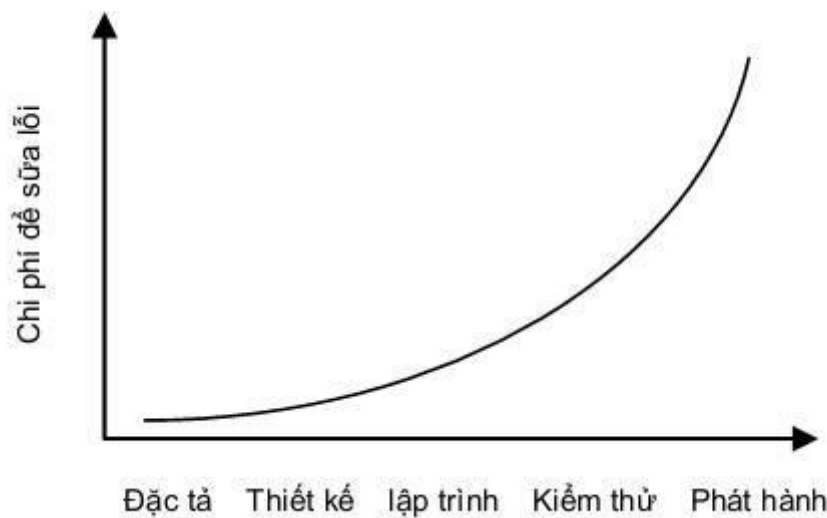
***Các lỗi về tài liệu:** Các lỗi về tài liệu là vấn đề của các đội phát triển và bảo trì khi có những sai sót trong các tài liệu liên quan. Những lỗi này có thể là nguyên nhân gây ra lỗi trong giai đoạn phát triển kế tiếp và giai đoạn bảo trì.

1.1.2.3 Chi phí cho việc sửa lỗi phần mềm

- Việc kiểm thử và sửa lỗi phần mềm có thể thực hiện trong bất cứ giai đoạn nào của vòng đời phần mềm. Tuy nhiên công việc này nên được thực hiện càng sớm càng tốt vì càng về giai đoạn sau của vòng đời phần mềm, chi phí cho việc tìm và sửa lỗi càng tăng, đặc biệt là đến giai đoạn đã triển khai phần mềm thì chi phí cho sửa lỗi sẽ trở nên rất lớn và ảnh hưởng trực tiếp đến uy tín của tổ chức phát triển phần mềm.

Theo tài liệu của Boehm, chi phí cho việc tìm và sửa lỗi phần mềm sẽ tăng

theo cấp độ phát triển phần mềm.



Sơ đồ 1: Chi phí cho việc sửa lỗi phần mềm

1.1.3 Khái niệm kiểm thử phần mềm

- Theo Glenford Myers: Kiểm thử là quá trình vận hành chương trình để tìm ra lỗi
- Theo IEEE: Kiểm thử là:
 - (1) Là quá trình vận hành hệ thống hoặc thành phần dưới những điều kiện xác định, quan sát hoặc ghi nhận kết quả đánh giá về hệ thống hoặc thành phần đó.
 - (2) Là quá trình phân tích phần mềm để tìm ra sự khác biệt giữa điều kiện thực tế và điều kiện yêu cầu và dựa vào điểm khác biệt đó để đánh giá tính năng phần mềm.

1.1.4 Mục tiêu của kiểm thử

- Tìm các bug phát sinh do dev tạo ra khi code.
- Đạt được sự tự tin và cung cấp thông tin về mức độ chất lượng.

- Đề ngăn ngừa lỗi.
- Đảm bảo rằng kết quả cuối cùng đáp ứng các yêu cầu kinh doanh và người sử dụng.
- Đề đạt được sự tín nhiệm của khách hàng bằng cách cung cấp cho họ một sản phẩm chất lượng.

Kiểm thử phần mềm sẽ giúp hoàn thiện các ứng dụng phần mềm hoặc sản phẩm so với yêu cầu kinh doanh và người sử dụng. Nó là rất quan trọng để đảm bảo kiểm thử tốt để kiểm thử các ứng dụng phần mềm hoàn toàn và chắc chắn rằng nó hoạt động tốt và theo các thông số kỹ thuật.

Việc xác định phạm vi kiểm tra các trường hợp kiểm thử nên được thiết kế tốt với khả năng tối đa của việc tìm kiếm các lỗi hiệu quả và được tính toán là số bug báo cáo cho mỗi trường hợp kiểm thử.

Kiểm tra phần mềm để chắc chắn kiểm thử đang thực hiện đúng cách và hệ thống đã sẵn sàng để sử dụng. Kiểm thử bao phủ các lĩnh vực khác nhau như: chức năng của các ứng dụng, khả năng tương thích của các ứng dụng với các hệ điều hành, phần cứng và các loại khác nhau của các trình duyệt, thực hiện kiểm thử để kiểm tra hiệu năng của các ứng dụng để đảm bảo rằng hệ thống đáng tin cậy và không có trục trặc hay không nên có bất kỳ vấn đề cản trở. Xác định rằng các ứng dụng có thể được triển khai một cách dễ dàng với máy tính và không có bất kỳ sự cố. Do đó các ứng dụng rất dễ dàng để cài đặt, tìm hiểu và sử dụng.

Kiểm thử phần mềm cho phép tạo ra những đánh giá khách quan về mức độ phù hợp của hệ thống các yêu cầu đã nêu và thông số kỹ thuật.

Kiểm tra xác nhận rằng hệ thống đáp ứng các yêu cầu khác nhau bao gồm: chức năng, hiệu suất, độ tin cậy, an toàn, khả năng sử dụng và như vậy. Việc

xác nhận này được thực hiện để đảm bảo rằng chúng tôi đang xây dựng hệ thống phù hợp.

Xác nhận để đảm bảo đang xây dựng hệ thống phù hợp. Ngoài việc giúp đưa ra quyết định, các thông tin từ các kiểm thử phần mềm giúp quản lý rủi ro.

1.1.5 Tầm quan trọng của kiểm thử phần mềm

Người phát triển phần mềm cho rằng:

- Kiểm thử để chứng minh chương trình không có lỗi
- Mục đích của kiểm thử là chỉ ra rằng chương trình đã thực hiện đúng các chức năng đã đưa ra.
- Kiểm thử là quy trình thực hiện để chứng tỏ chương trình đã làm được các chức năng cần có
- Kiểm thử còn để tìm lỗi và sửa chữa các lỗi đó nhằm tăng thêm độ tin cậy cho phần mềm

Vai trò của kiểm thử tròn suốt quá trình sống của phần mềm:

- Kiểm thử không tồn tại độc lập
- Các hoạt động của kiểm thử luôn gắn liền với các hoạt động phát triển phần mềm

Các mô hình phát triển phần mềm khác nhau cần các cách tiếp cận kiểm thử khác nhau

1.1.6 Các nguyên tắc trong kiểm thử

Trong kiểm thử các 7 nguyên tắc cơ bản:

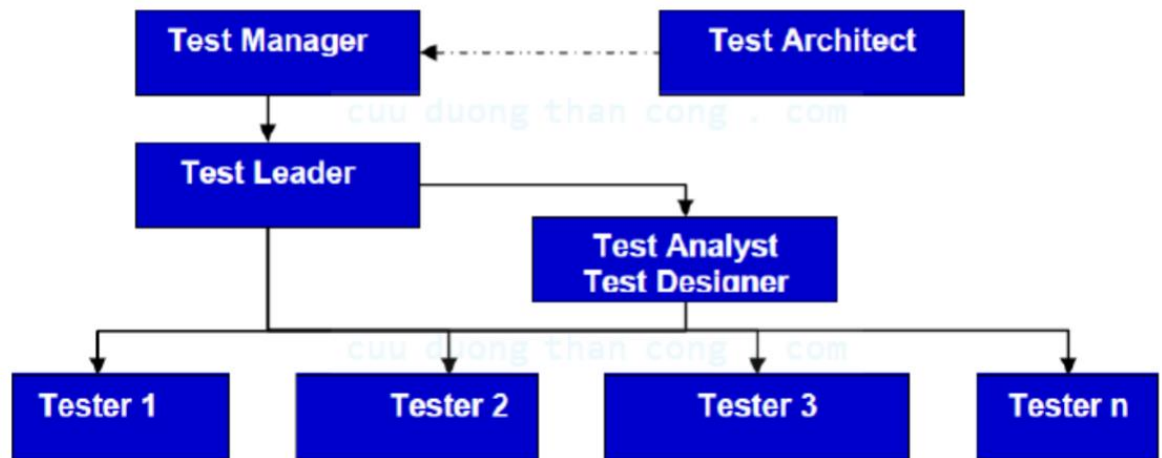
- 1, Kiểm thử chỉ ra sự hiện diện của lỗi trong phần mềm
- 2, Kiểm thử tất cả các trường hợp là không thể
- 3, Nên thực hiện kiểm thử càng sớm càng tốt
- 4, Sự phân cụm của các lỗi
- 5, Ngịch lý thuốc trừ sâu
- 6, Kiểm thử theo các ngữ cảnh độc lập

7, Sự sai lầm về việc không có lỗi

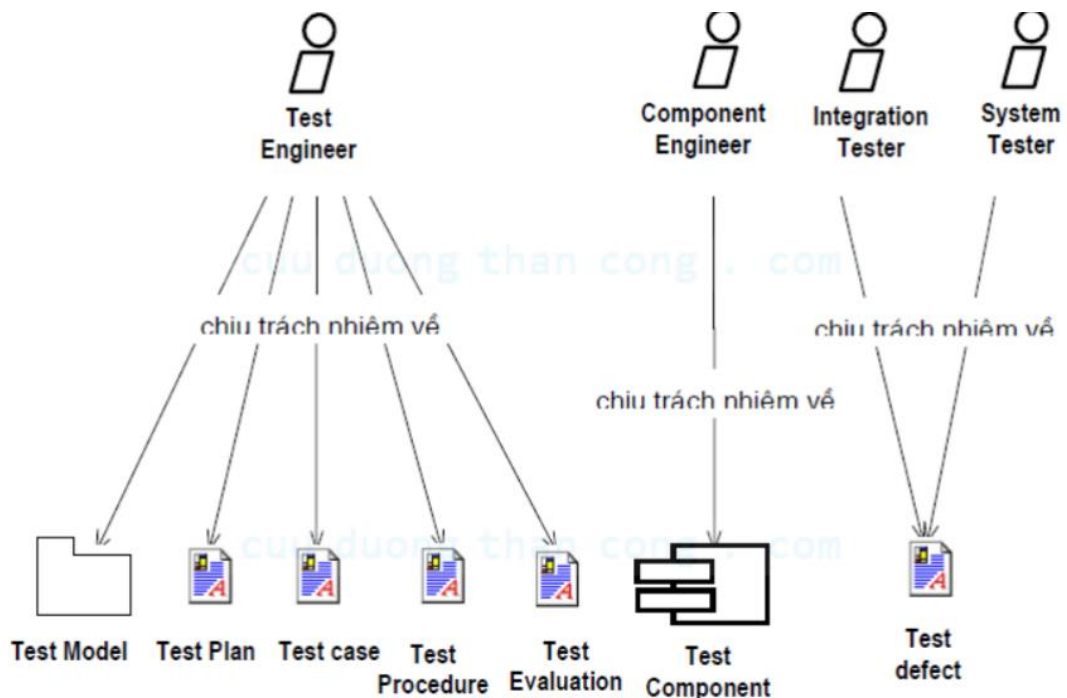
1.1.7 Phân loại kiểm thử

- Dựa vào mục đích kiểm thử
 - Kiểm thử đơn vị, module
 - Kiểm thử cấu hình
 - Kiểm thử sơ lược
 - Kiểm thử tích hợp
 - Kiểm thử chức năng
 - Kiểm thử hồi quy
 - Kiểm thử hệ thống
 - Kiểm thử tải dữ liệu
 - Kiểm thử tải trọng
 - Kiểm thử hiệu suất
 - Kiểm thử chấp nhận
 - Kiểm thử bảo mật
- Dựa vào chiến lược kiểm thử
 - Kiểm thử thủ công: Thực hiện kiểm thử mọi thứ bằng tay
 - Kiểm thử tự động: Kiểm thử một cách tự động nhờ công cụ kiểm thử
- Dựa vào phương pháp tiến hành
 - Kiểm thử tĩnh: thường không kiểm thử chi tiết mà kiểm tra tính đúng đắn của thuật toán, code, tài liệu
 - Kiểm thử động: chạy mã lập trình thực tế trong các tình huống, diễn ra khi bản thân chương trình đó đang được sử dụng
- Dựa vào kỹ thuật kiểm thử
 - Kiểm thử hộp trắng
 - Kiểm thử hộp đen

1.1.8 Đối tượng thực hiện kiểm thử



Sơ đồ tổ chức của đội kiểm thử



Các worker và quy trình

1.1.9 Các hạn chế của kiểm thử

- Không thể chắc chắn đặc tả phần mềm đúng hoàn toàn
- Không thể chắc chắn hệ thống hay tool kiểm thử là đúng
- Không có tool kiểm thử nào phù hợp với mọi phần mềm
- Kỹ sư kiểm thử không thể chắc chắn hiểu hết về phần mềm
- Không có tài nguyên để kiểm thử tất cả

- Không thể tìm ra được tất cả lỗi

1.2 KỸ THUẬT KIỂM THỬ THỦ CÔNG VÀ TỰ ĐỘNG

1.2.1 Kỹ thuật kiểm thử thủ công (Manual testing)

1.2.1.1 Khái niệm:

- Kiểm thử thủ công là kiểm thử một phần mềm một cách thủ công (không sử dụng bất kỳ công cụ tự động hoặc bất kỳ đoạn mã nào). Với loại kiểm thử này, tester như người sử dụng cuối sẽ kiểm tra phần mềm để xác định bất kỳ hành vi không mong muốn hoặc lỗi. Có rất nhiều giai đoạn để kiểm thử bằng tay như Kiểm thử đơn vị (Unit testing), Kiểm thử tích hợp (Integration testing), Kiểm thử hệ thống (System testing) và Kiểm thử chấp nhận (User Acceptance testing).

1.2.1.2 Mục tiêu của kiểm thử tự động

- Kiểm thử thủ công là đảm bảo rằng ứng dụng hoạt động phù hợp với các yêu cầu chức năng được chỉ định.
- Kiểm thử thủ công cũng đảm bảo rằng các lỗi đã tìm thấy được sửa chữa bởi các developer và được kiểm thử lại bởi những tester sau khi các lỗi được khắc phục.

1.2.1.3 Các loại kiểm thử thủ công

- Black Box Testing
- White Box Testing
- Unit Testing
- System Testing
- Integration Testing
- Acceptance Testing

1.2.1.4 Cách thực hiện kiểm thử thủ công

1. Đọc và hiểu tài liệu của dự án phần mềm. Ngoài ra, nghiên cứu Ứng dụng khi thực hiện kiểm thử (AUT) nếu có.
2. Dự thảo kiểm thử bao gồm tất cả các yêu cầu được đề cập trong tài liệu.
3. Xem xét và vạch ra các trường hợp thử nghiệm với Trưởng nhóm, Khách hàng (nếu có)
4. Thực hiện các trường hợp kiểm thử trên AUT
5. Báo cáo lỗi.
6. Khi các lỗi đã được sửa sẽ được tester thực hiện một lần nữa các trường hợp kiểm thử thất bại để xác minh rằng lỗi đã được khắc phục.

1.2.1.5 Ưu nhược điểm

Ưu điểm:

- Nhận phản hồi trực quan nhanh và chính xác
- Ít tốn kém hơn vì không cần phải chi ngân sách cho các công cụ và quy trình tự động hóa
- Sự phán đoán và trực giác của con người luôn có lợi cho yếu tố thử công
- Trong khi kiểm thử một thay đổi nhỏ, một kiểm thử tự động hóa sẽ yêu cầu coding có thể tốn thời gian. Trong khi kiểm thử thủ công lại không tốn thời gian.

Nhược điểm:

- Phương pháp kiểm thử thủ công ít đáng tin cậy hơn vì được thực thi bởi con người. Do đó, dễ mắc sai lầm & không tìm thấy lỗi.
- Quá trình kiểm thử thủ công không thể được ghi lại, do đó không thể sử dụng lại.

- Một số phần nhất định khó thực hiện thủ công, có thể cần thêm thời gian.

1.2.2 Kỹ thuật kiểm thử tự động (Automation testing)

1.2.2.1 Khái niệm:

- Kiểm thử tự động (Automation testing) là khi tester viết kịch bản và sử dụng những công cụ hỗ trợ để kiểm thử phần mềm. Quá trình này là chuỗi tự động hóa quá trình kiểm thử thủ công. Kiểm thử tự động được sử dụng để chạy lại các kịch bản kiểm thử đã được kiểm thử thủ công, nhanh chóng, lặp lại nhiều lần.

- Ngoài kiểm thử hồi quy, kiểm thử tự động cũng được sử dụng để kiểm tra ứng dụng từ load, performance, và stress. Nó tăng thêm độ bao phủ, cải thiện độ chính xác, tiết kiệm thời gian và tiền bạc so với kiểm thử thủ công

1.2.2.2 Tại sao cần kiểm thử tự động

Kiểm thử tự động rất quan trọng vì những lý do sau:

- Kiểm thử thủ công tất cả các kịch bản nhiều lần sẽ gây tốn thời gian và tiền bạc
- Rất khó để kiểm thử thủ công các trang web hay phần mềm đa ngôn ngữ
- Kiểm thử tự động hóa không cần sự can thiệp của con người. Có thể chạy kiểm thử tự động qua đêm mà không cần giám sát.
- Kiểm thử tự động hóa làm tăng tốc độ thực hiện kiểm thử
- Kiểm thử tự động hóa giúp tăng phạm vi kiểm thử

Kiểm thử thủ công có thể trở nên nhàm chán, do đó dễ bỏ sót lỗi.

1.2.2.3 Tự động hóa là gì?

- Không phải tự động hóa lúc nào cũng khả thi với mọi phần mềm. Các lĩnh vực như người dùng có thể thực hiện giao dịch cũng như form đăng nhập hoặc form đăng ký, bất kỳ lĩnh vực nào có số lượng người dùng lớn có thể truy cập đồng thời vào phần mềm thì nên được tự động hóa.

1.2.2.4 Tự động hóa như thế nào?

Tự động hóa được thực hiện bằng cách sử dụng một ngôn ngữ máy tính được hỗ trợ như VB và ứng dụng phần mềm tự động. Có rất nhiều công cụ có sẵn có thể được sử dụng để viết những đoạn kịch bản tự động. Trước khi nhắc đến các công cụ đó, hãy xác định kế hoạch để tự động hóa quá trình kiểm thử:

- Xác định các lĩnh vực trong phần mềm để tự động hoá.
- Lựa chọn những công cụ thích hợp cho kiểm thử tự động hóa.
- Viết mã tự động hóa (test scripts).
- Phát triển các bộ kiểm tra (test suits).
- Thực thi các đoạn mã.
- Tạo các báo cáo kết quả kiểm thu.
- Phát hiện các lỗi tiềm ẩn hoặc vấn đề hiệu suất.

1.2.2.5 Quy trình kiểm thử tự động

- Lựa chọn công cụ kiểm thử: Lựa chọn Công cụ kiểm thử phần lớn phụ thuộc vào công nghệ mà Ứng dụng Kiểm thử được xây dựng. Chẳng hạn, QTP không hỗ trợ Informatica. Vì vậy, QTP không thể được sử dụng để kiểm thử các ứng dụng của Informatica.

- Xác định phạm vi tự động: Phạm vi tự động là phạm vi Ứng dụng được Kiểm thử tự động hóa. Các điểm sau giúp xác định phạm vi tự động:

- Các tính năng quan trọng

- Kịch bản có lượng dữ liệu lớn
- Các chức năng chung trên ứng dụng
- Tính khả thi về mặt kỹ thuật
- Mức độ các thành phần được sử dụng lại
- Độ phức tạp của test cases
- Khả năng sử dụng các test cases tương tự để kiểm thử trên nhiều trình duyệt

- Lập kế hoạch, thiết kế và phát triển: Tester tạo một chiến lược & kế hoạch kiểm thử tự động, chi tiết như sau:

- Công cụ kiểm thử tự động được chọn
- Thiết kế Framework và các tính năng của Framework
- Các mục trong phạm vi và ngoài phạm vi tự động hóa
- Chuẩn bị kiểm thử tự động hóa
- Lịch trình và mốc thời gian thực thi kịch bản kiểm thử
- Bàn giao sản phẩm được kiểm thử tự động

- Thực thi kiểm thử: Kịch bản kiểm thử tự động hóa được thực thi trong giai đoạn này. Các Scripts cần dữ liệu kiểm thử đầu vào trước khi được thiết lập để chạy. Sau khi thực hiện sẽ trả về các báo cáo kiểm thử chi tiết.

- Bảo trì: Khi các chức năng mới được thêm vào Hệ thống được kiểm thử với các chu kỳ liên tiếp, Scripts tự động hóa cũng cần được thêm, xem xét và bảo trì cho mỗi chu kỳ phát hành. Bảo trì trở nên cần thiết để cải thiện hiệu quả của Scripts trong kiểm thử tự động hóa.

- Các công cụ kiểm thử tự động hóa

- Tricentis
- Mabl
- Selenium

- QTP(MicroF Focus UFT)
- Rational Functional Tester
- Witr
- Skill test

- Ưu nhược điểm

Ưu điểm:

- Vì hầu hết các phần của quy trình kiểm thử là tự động, bạn có thể có một quy trình nhanh chóng và hiệu quả
- Quá trình tự động có thể được ghi lại. Điều này cho phép bạn sử dụng lại và thực hiện cùng loại hoạt động kiểm thử
- Kiểm thử tự động được thực hiện bằng các công cụ phần mềm, do đó, có thể hoạt động liên tục mà không biết mệt mỏi.
- Có thể dễ dàng tăng năng suất, cung cấp kết quả kiểm thử nhanh & chính xác
- Kiểm thử tự động giúp bạn tìm thấy nhiều lỗi hơn so với tester

Nhược điểm:

- Không có yếu tố con người, thật khó để hiểu sâu hơn về các khía cạnh trực quan của giao diện người dùng như màu sắc, phông chữ, kích thước, độ tương phản hoặc kích thước nút bấm, layout...
- Các công cụ để chạy kiểm thử tự động có thể tốn kém, có thể làm tăng chi phí kiểm thử của dự án.
- Công cụ kiểm thử tự động chưa phải là bằng chứng đầy đủ. Mỗi công cụ tự động hóa đều có những hạn chế của chúng làm, giảm phạm vi tự động hóa.

1.3 MỘT SỐ CÔNG CỤ KIỂM THỬ TỰ ĐỘNG

Các công cụ kiểm thử của Selenium: Selenium là một bộ kiểm thử tự động mã nguồn mở miễn phí cho các ứng dụng web trên các trình duyệt và nền tảng khác nhau. Nó khá giống với HP Quick Test Pro (QTP) nhưng Selenium chỉ tập trung vào việc tự động hóa các ứng dụng dựa trên web. Kiểm thử được thực hiện bằng công cụ Selenium thường được gọi là Selenium Testing.

Trong Selenium có 4 phần:

1. Selenium IDE
2. Selenium RC
3. Selenium WebDriver
4. Selenium Grid

1.3.1 Selenium IDE

Môi trường phát triển tích hợp Selenium (IDE) là framework đơn giản nhất trong bộ Selenium. Nó là một plugin Firefox mà bạn có thể cài đặt dễ dàng như các plugin khác. Tuy nhiên, vì sự đơn giản của nó, nên chỉ sử dụng Selenium IDE làm công cụ tạo mẫu. Nếu bạn muốn tạo thêm các trường hợp kiểm thử nâng cao, bạn sẽ cần phải sử dụng hoặc là Selenium RC hoặc WebDriver.

1.3.2 Selenium RC

Selenium RC là **framework kiểm thử hàng đầu** của toàn bộ dự án Selenium trong một thời gian dài. Đây là công cụ kiểm thử web tự động đầu tiên **cho phép người dùng sử dụng đa dạng ngôn ngữ lập trình để làm việc với nó**. Kể từ phiên bản 2.25.0, RC có thể hỗ trợ các ngôn ngữ lập trình sau: Java, C#, PHP, Python, Perl, Ruby

1.3.3 Selenium WebDriver

WebDriver tốt hơn cả Selenium IDE và Selenium RC ở nhiều khía cạnh. Nó là một cách tiếp cận hiện đại và ổn định hơn trong việc tự động hóa các hành động của trình duyệt. WebDriver, không giống như Selenium RC, không dựa vào

JavaScript cho tự động hóa. Nó kiểm soát trình duyệt bằng cách giao tiếp trực tiếp với nó. Các ngôn ngữ được hỗ trợ giống như ngôn ngữ trong Selenium RC: Java, C#, PHP, Python, Perl, Ruby

1.3.4 Selenium Grid

Selenium Grid là một công cụ **được sử dụng cùng với Selenium RC để chạy kiểm thử song song** trên các máy khác nhau và các trình duyệt khác nhau cùng một lúc. Thực thi song song có nghĩa là chạy nhiều kiểm thử cùng một lúc.

1.3.5 Công cụ kiểm thử Jmeter

JMeter là một phần mềm có thể được sử dụng để thực hiện kiểm tra hiệu năng (**performance testing**), kiểm tra tải (**load testing**) và kiểm tra chức năng (**functional testing**) của các ứng dụng web của bạn. JMeter cũng có thể mô phỏng tốc độ tải trên một máy chủ bằng cách tạo ra hàng ngàn người dùng ảo đồng thời truy cập máy chủ web.

1.3.6 Katalon (katalon studio)

- Là một bộ công cụ toàn diện cho kiểm thử tự động hoá ứng dụng trên web và điện thoại di động. Nó có đầy đủ các tính năng mạnh mẽ để vượt qua được những vấn đề phổ biến trong tự động hoá thử nghiệm giao diện web (ví dụ: pop-up, iFrame, wait-time). Nhờ đó, tester thực hiện công việc tốt hơn, nhanh hơn, tăng cao chất lượng phần mềm qua các giải pháp thông minh, thân thiện và linh hoạt.

1.3.7 Junit

Là một framework mã nguồn mở, miễn phí, quan trọng trong sự phát triển của test – driven của lập trình viên về hướng đối tượng java.

1.3.8 Appium

Là một công cụ kiểm thử tự động mã nguồn mở để tự động hoá các ứng dụng di động native và hybrid. Nó được biết đến như một công cụ kiểm thử

1.3.9 Công cụ kiểm thử LoadStorm

LoadStorm là một công cụ kiểm thử cho các ứng dụng web và mobile.

LoadStorm là một công cụ kiểm tra tải SaaS. Nó thử nghiệm hiệu suất theo yêu cầu, kiểm tra tải và thử nghiệm ứng suất cho các ứng dụng web và trang web. Nó tạo giúp cho việc tìm kiếm các dữ liệu có vấn đề của trang web của bạn bằng cách cung cấp báo cáo phân tích sâu rộng trên máy chủ, từng trang hoặc theo loại yêu cầu cho mọi chỉ số hiệu suất.

1.3.10 Công cụ kiểm thử QTP

Quick Test Professional (QTP) là một công cụ kiểm thử tự động được thiết kế bởi Mercury Interactive và sau đó được mua lại bởi HP. QTP giúp tester tiến hành các kiểm tra một cách tự động để xác định errors, defects khác với kết quả mong muốn của ứng dụng, phần mềm hay chức năng... mà ta đang kiểm tra. QTP được dùng để kiểm thử chức năng (functional test) và cho phép thực hiện kiểm thử hồi qui (regression test) một cách tự động

1.3.11 Công cụ kiểm thử Robotium

Robotium là một công cụ viết bằng mã nguồn mở dùng để kiểm thử hộp xám tự động dành cho các ứng dụng trên điện thoại hệ điều hành android.

Robotium giúp viết các test case về test chức năng, hệ thống, và acceptance test scenarios, bao phủ các tính năng của Android.

1.3.12 Công cụ kiểm thử SOASTA CloudTest

SOASTA CloudTest hỗ trợ chức năng quản lý người dùng (RUM - Real User Monitoring) và cho phép những người dùng xây dựng những bài test sử dụng những dữ liệu dựa trên cơ chế quản lý người dùng (RUM based data) và những kịch bản ứng dụng cụ thể. SOASTA CloudTest có thể kiểm tra khả năng chịu tải của các ứng dụng theo vị trí địa lý khác nhau, đặc biệt 2 khâu integration và phân tích thời gian thực giữa các monitoring, test design, reporting đều được tiến hành một cách liền mạch.

CHƯƠNG 2. CÔNG CỤ KIỂM THỬ QTP (QUICK TEST PROFESSIONAL)

2.1 GIỚI THIỆU



HP Quick Test Professional viết tắt là QTP, là một phần mềm hỗ trợ Kiểm tra tự động hóa cho các ứng dụng phần mềm. QTP đi kèm với một giao diện người dùng có thể được coi là một môi trường phát triển tích hợp (IDE) cho bản thân bài kiểm tra. IDE có các tính năng khác nhau giúp người thử nghiệm phát triển một kịch bản toàn diện để xác thực thành công mục đích của thử nghiệm. Các kiến thức cơ bản về QTP cụ thể như sau:

- Nó sử dụng VB Script làm ngôn ngữ kịch bản của nó. (Một ngôn ngữ kịch bản là một ngôn ngữ được diễn giải trong thời gian chạy)
- QTP chỉ chạy trong môi trường windows.
- Phiên bản hiện tại của QTP là 11.0 (Phiên bản mới nhất Unified Functional Testing - UFT 11.5)
- Các công nghệ mà nó hỗ trợ là Web, Java.Net, SAP, Oracle, Siebel, PeopleSoft, Dịch vụ Web và nhiều ngôn ngữ chính. Mặc dù một số phiên bản cũ hơn không hỗ trợ tất cả các công nghệ được liệt kê.

Quick Test Professional (QTP) là một công cụ kiểm thử tự động được thiết kế bởi Mercury Interactive và sau đó được mua lại bởi HP. QTP giúp tester tiến hành các kiểm tra một cách tự động để xác định errors, defects khác với kết quả mong muốn của ứng dụng, phần mềm hay chức năng... mà ta đang kiểm tra. Nếu ta có một phần mềm Quản Lý Nhân Sự. Ví dụ, khi mở phần

mềm Quản Lý Nhân sự lên, thì người dùng sẽ gặp form Đăng nhập(login) để nhập vào “Tên tài khoản”, và “Mật khẩu”, rồi nhấn nút “OK” hoặc “Cancel” để vào. Ta lập trình ra lệnh cho QTP tự động điền thông tin vào 2 ô “Tên tài khoản” và “Mật khẩu”, và rồi tự động nhấn nút “OK” hoặc “Cancel” giúp ta luôn. Công việc bày gọi là viết script cho QuickTest Pro. Viết script để thực hiện nhiều trường hợp nhập dữ liệu khác nhau, để thử xem chức năng của form “Đăng nhập” có hoạt động đúng hay không. QTP sau khi chạy script xong, sẽ thực hiện ghi nhận kết quả việc test tự động, và có thể xuất report . Nếu có đủ một hệ thống Mercury Quality Center thì ít ra phải có thêm phần mềm Mercury Test Director đóng vai trò là phần mềm chủ(serving software) đảm nhận việc tổng hợp các kết quả test, các báo cáo, các phát sinh,... của QTP, từ đó phục vụ cho công việc quản trị chất lượng sản phẩm của ta (Software Quality Assurance). Đây là chương trình dùng để kiểm tra chức năng (functional test) và cho phép thực hiện kiểm tra hồi quy(regression test) một cách tự động.

Lịch sử phát hành

- 5.5: Bản phát hành đầu tiên: Được phát hành vào năm 2001
- 6.0: Phát hành năm 2002
- 6.5: Phát hành năm 2003
- 7.0: Chưa bao giờ phát hành
- 8.0: Phát hành năm 2004
- 8.2: Phát hành năm 2005
- 9.0: Phát hành năm 2007
- 9.1: Phát hành năm 2007
- 9.2: Phát hành năm 2007
- 9.5: Phát hành năm 2008
- 10.0: Phát hành năm 2009
- 11.0: Phát hành năm 2010
- 11,5: Được phát hành vào năm 2012 [đổi tên thành Unified Functional Testing] [\[26\]](#)
- 11,52: Phát hành vào tháng 6 năm 2013
- 11,53: Phát hành vào tháng 11 năm 2013
- 12.0: Phát hành tháng 3 năm 2014
- 12,01: Phát hành tháng 7 năm 2014
- 12.02: Phát hành tháng 12 năm 2014
- 12,50: Phát hành tháng 7 năm 2015
- 12,54: Phát hành tháng 10 năm 2016
- 14:00: Phát hành ngày 27 tháng 1 năm 2017
- 14.01: Phát hành ngày 18 tháng 8 năm 2017
- 14.02: Phát hành tháng 12 năm 2017

- 14.03: Phát hành tháng 3 năm 2018
- 14.50: Phát hành tháng 8 năm 2018
- 14.51: Phát hành tháng 11 năm 2018
- 14.52: Phát hành tháng 3 năm 2019
- 14.53: Phát hành tháng 7 năm 2019

QTP hỗ trợ sẵn một số loại chương trình thông dụng như:

- Ứng dụng Windows chuẩn/Win32.
- Ứng dụng web theo chuẩn HTML, XML chạy trong trình duyệt Internet Explorer, Netscape hoặc AOL.
- Visual Basic.
- ActiveX.
- QTP hỗ trợ Unicode (UTF-8, UTF-16).

Một số loại chương trình khác đòi hỏi chúng ta phải cài đặt thêm thành phần bổ sung của QTP (plug-in) thì mới thực hiện kiểm thử được. Các loại chương trình đó là:

.NET	<ul style="list-style-type: none"> • NET Framework 1.0, 1.1, 2.0 beta • Các đối tượng chuẩn của .NET và các đối tượng khác thừa kế từ các đối tượng chuẩn.
Java	<ul style="list-style-type: none"> • Sun JDK 1.1 – 1.4.2 • IBM JDK 1.2 – 1.4
Oracle	<ul style="list-style-type: none"> • Oracle Applications 11.5.7, 11.5.8, 11.5.9
People Soft	<ul style="list-style-type: none"> • PeopleSoft Enterprise 8.0 – 8.8
SAP	<ul style="list-style-type: none"> • SAP GUI HMTL 4.6D, 6.10, 6.20 • SAP Workplace 2.11 • SAP Enterprise Portal 5.0
Siebel	<ul style="list-style-type: none"> • Siebel 7.0, 7.5, 7.7
Terminal Emulators	<ul style="list-style-type: none"> • Attachmate EXTRA! 6.7, 7.1 • Attachmate EXTRA! Terminal Viewer 3.1 Java sessions • IBM Personal Communications • ...

2.2 ĐẶC ĐIỂM CỦA QTP

- Dễ sử dụng, bảo trì, tạo test script nhanh. Cung cấp dữ liệu kiểm thử rõ ràng và dễ hiểu.
- Kiểm thử phiên bản mới của ứng dụng với rất ít sự thay đổi. Ví dụ khi ứng dụng thay đổi nút tên “Login” thành “Đăng nhập”, thì chỉ cần cập nhật lại Object Repository để QTP nhận ra sự thay đổi đó mà không cần thay đổi bất cứ test script nào.
- Hỗ trợ làm việc theo nhóm thông qua sự chia sẻ thư viện, thống nhất quản lý Object Repository.
- Thực tế cho thấy, QTP thực hiện kiểm thử tự động trên nhiều trình duyệt cùng lúc tốt hơn những công cụ kiểm thử tự động khác.
- Với chức năng Recovery Scenarios, QTP cho phép xử lý những sự kiện hoặc lỗi không thể đoán trước có thể làm script bị dừng trong khi đang chạy.
- QTP có khả năng hiểu test script của Mercury Winrunner (một công cụ kiểm thử khác của Mercury).
- Một số tính năng nổi bật

Quản trị Object Repository	Phối hợp giữa các kỹ sư kiểm thử qua việc đồng bộ hóa dữ liệu, khả năng trộn, nhập/xuất ra file XML
Thư viện hàm mới	Chia sẻ các thư viện hàm giữa các nhóm kỹ sư kiểm thử
Kiểm thử tài nguyên	Kiểm thử tài nguyên cần thiết trước khi thực thi lệnh kiểm thử tự động.
Nâng cấp khả năng kéo thả	Kéo thả các bước kiểm thử trong môi trường ngôn ngữ tự nhiên.
Hỗ trợ XML cho báo cáo	Lưu trữ kết quả kiểm thử dưới dạng XML, HTML, từ đó cho phép tùy biến báo cáo.
Trình phát triển mới (IDE)	Môi trường soạn thảo mới, mềm dẻo cho tùy biến và sử dụng.

Trình dò lỗi mới	Cho phép kỹ sư kiểm thử kiểm soát lỗi khi viết test case.
Quản trị từ khóa	Quản lý từ khóa trong quá trình sử dụng
Hỗ trợ đa giao tiếp	Cho phép người dùng mở và soạn thảo đồng thời nhiều hàm thư viện và Object Repository.
Hỗ trợ Unicode	Hỗ trợ Unicode với các ứng dụng đa ngôn ngữ (multi-language).
Menu bar	Cấu hình thao tác với QTP và script
File toolbar	Hỗ trợ quản lý script
Debug toolbar	Hỗ trợ kiểm thử lỗi trong test script (debug)
Testing toolbar	Hỗ trợ quá trình tạo test script hoặc thực hiện KTTĐ
Action toolbar	Xem một Action (thủ tục, hàm) hoặc toàn bộ chu trình của test script
Test pane	Soạn thảo script ở một trong 2 chế độ Keyword View hoặc Expert View
Data Table	Nơi lưu trữ dữ liệu cho test script
Active Screen	Xem lại giao diện PM được kiểm thử

2.3 CÁC THÀNH PHẦN QUAN TRỌNG QTP

2.3.1 Action

Giống như thủ tục hay hàm trong các ngôn ngữ lập trình khác, Action ghi lại các bước thực hiện kiểm thử và nó có thể được sử dụng lại nhiều lần. Trong một test script có thể có nhiều action.

2.3.2 DataTable

Nơi lưu trữ dữ liệu phục vụ cho kiểm thử. Một test script sẽ có một DataTable được dùng chung cho tất cả các Action. Bên cạnh đó mỗi Action cũng có một DataTable riêng cho mình.

2.3.3 Object Repository (OR)

- Cấu trúc theo dạng cây, mô tả các đối tượng trong phần mềm được kiểm tra. Đây được xem là cầu nối để test script tương tác với phần mềm được kiểm tra.
- Khi ra lệnh cho QTP ghi lại thao tác người dùng lên phần mềm thì trong OR sẽ tự động phát sinh thành phần đại diện cho những đối tượng trên phần mềm vừa được thao tác.
- OR có thể tổ chức thành 2 loại, một loại dùng chung trong nhiều test script, loại khác dùng theo từng nhóm Action.

2.3.4 Checkpoint

Có thể hiểu là nơi kiểm tra trong test script, khi chạy nó sẽ thực hiện so sánh kết quả thực tế khi kiểm tra phần mềm với kết quả mong đợi. Sau khi tiến hành so sánh QTO sẽ tự động ghi lại kết quả vào Test Results.

2.4 ƯU NHƯỢC ĐIỂM

2.4.1 Ưu điểm

- Nó hỗ trợ record và playback
- Nó sử dụng màn hình active để record scripts và giúp tester tham khảo các thuộc tính đối tượng
- Nó hỗ trợ add-in khác nhau như Oracle, Java, SAP, NET, Web Forms, People soft, etc..
- Nó cho phép bạn nâng cao khả năng test hiện có ngay cả khi không có AUT thông qua một màn hình hoạt động
- Nó hỗ trợ hầu hết automation frameworks- keyword driven testing approach, data driven testing approach, etc..
- Nó có thể kết hợp với Test management tools như Quality Center, Test Director, and Winrunner

- Một số loại khác như Smoke, Regression, Sanity có thể dễ dàng duy trì Nó hỗ trợ XML

2.4.2 Nhược điểm

- QTP không thể nhận ra các đối tượng đã được người dùng tùy chỉnh và một số đối tượng phức tạp khác.
- QTP chỉ chạy trên môi trường windowx.
QTP mất phí nên công cụ này không được sử dụng rộng rãi.

2.5 QUÁ TRÌNH CÀI ĐẶT

Các yêu cầu tối thiểu để có thể cài đặt QTP từ 12.0 trở lên như sau

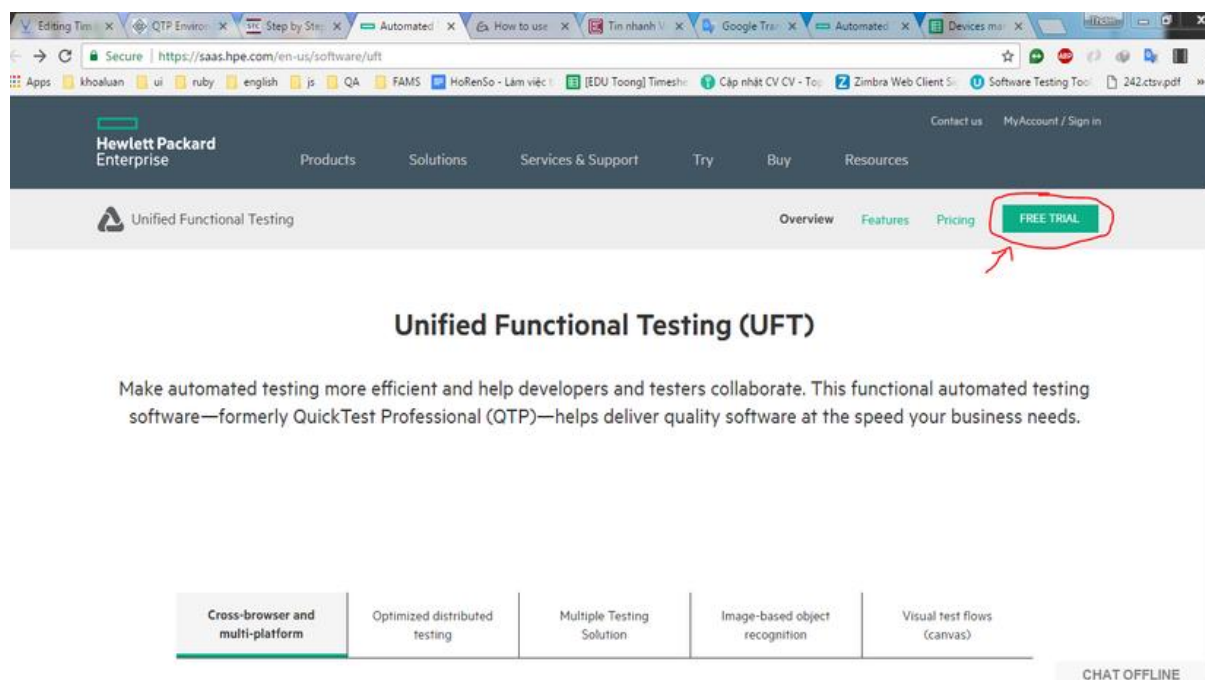
Chỗ trống trong ổ cứng	1GB vùng trống để files và folder ứng dụng và 120MB chỗ trống để cài đặt hệ điều hành
Bộ vi xử lý máy tính	Pentium IV (1.6 GHz) hoặc cao hơn
Hệ điều hành	Windows XP Service Pack 3, Windows Vista, Windows 7, Windows 8/8.1, Windows 10
Color Settings	High Color (16 bit)
Card Đồ họa	64MB video memory
Memory	Cần tối thiểu 2 GB bộ nhớ để có thể thêm ba add-in, đồng thời và nếu tải thêm add-in thì bộ nhớ cũng cần nhiều hơn. Ví dụ khi sử dụng tùy chọn lưu phim hay tùy chọn kết quả để chụp phim trong các phiên chạy
Hard Disk Driver	5400 RPM

(Hình 2.5.1)

Các bước cài đặt như sau:

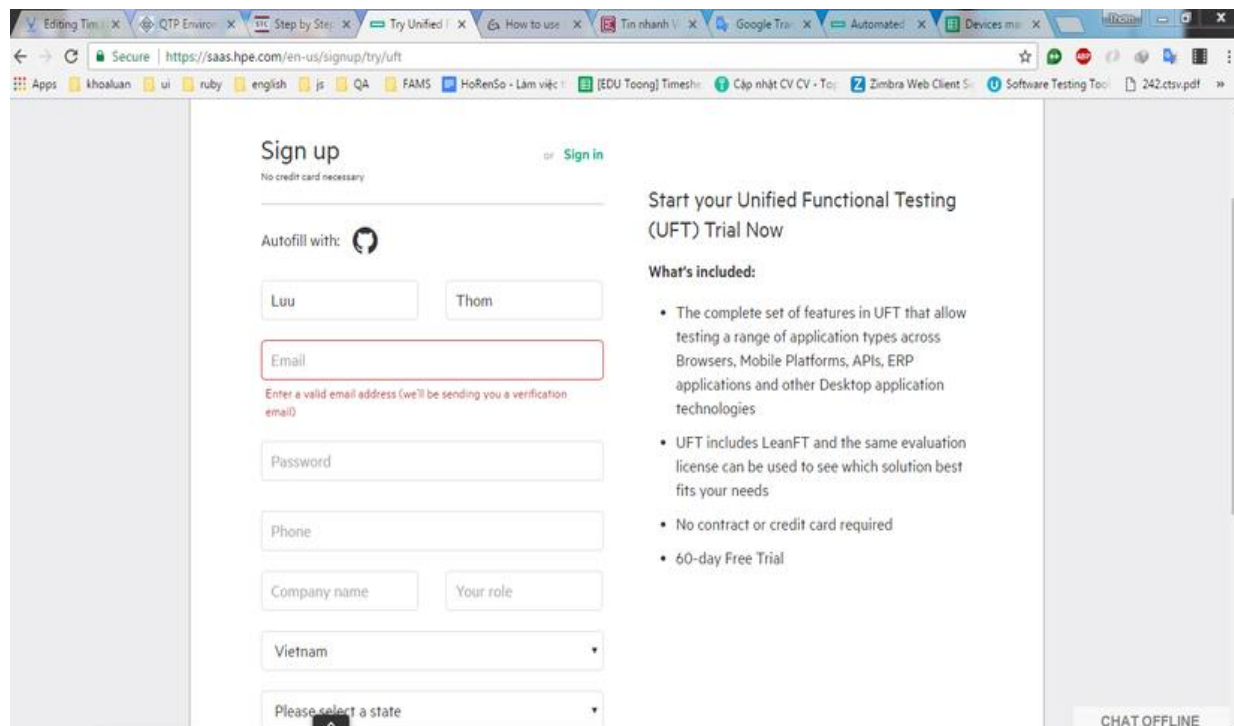
B1: Vào đường link <https://saas.hpe.com/enus/software/uft>

B2: Click vào free trial



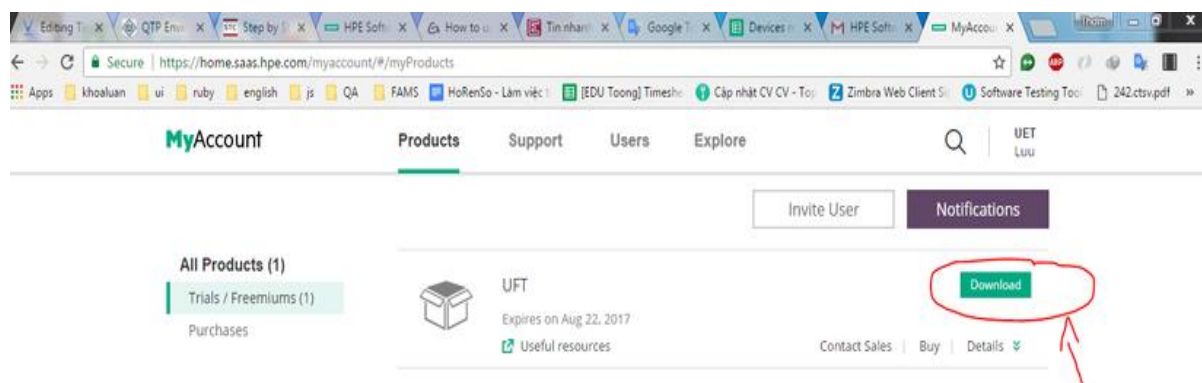
(Hình 2.5.2)

B3: Điền thông tin tạo tài khoản, sau đó xác nhận email được gửi vào mail mà bạn đăng kí



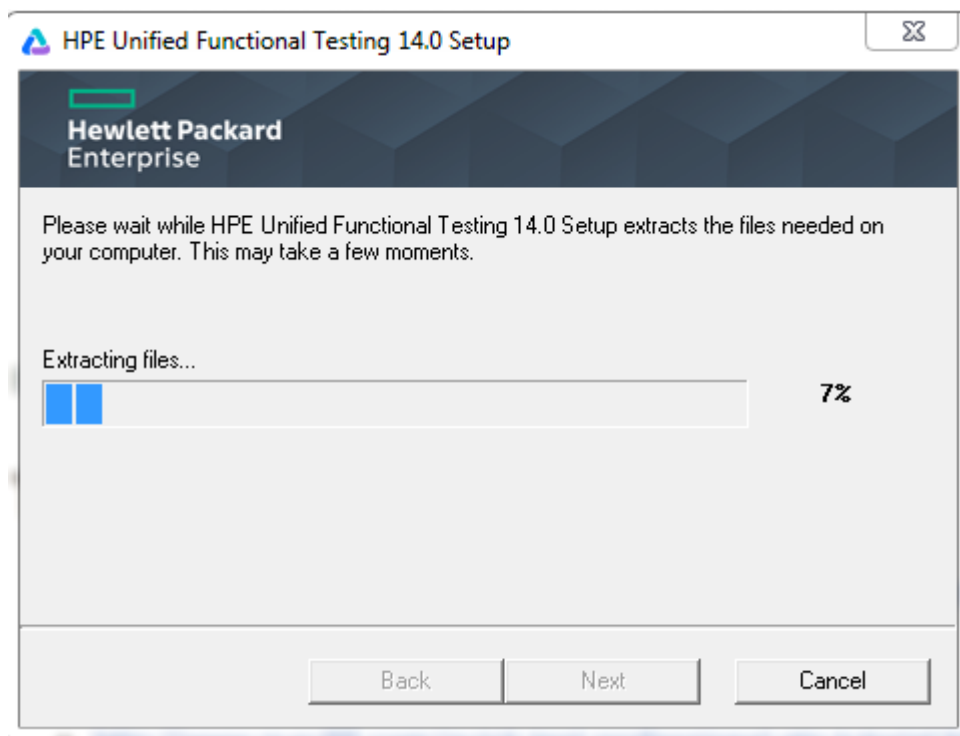
(Hình 2.5.3)

B4: Đăng nhập vào hệ thống download version QTP



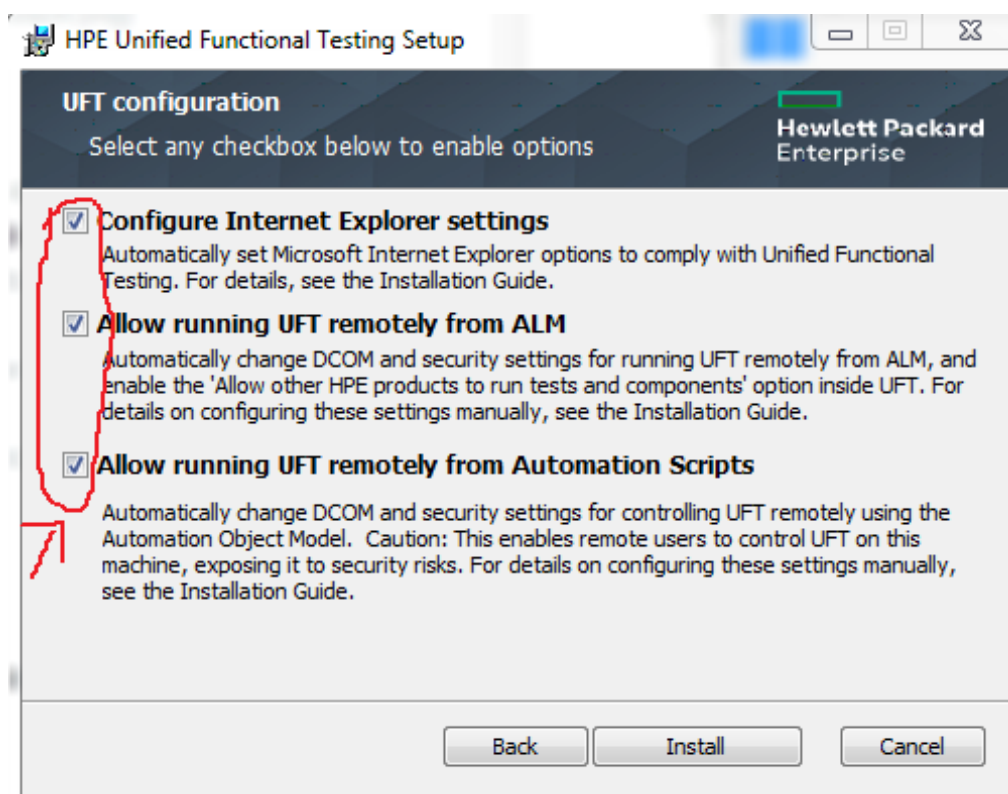
(Hình 2.5.4)

B5: Chạy file setup exe



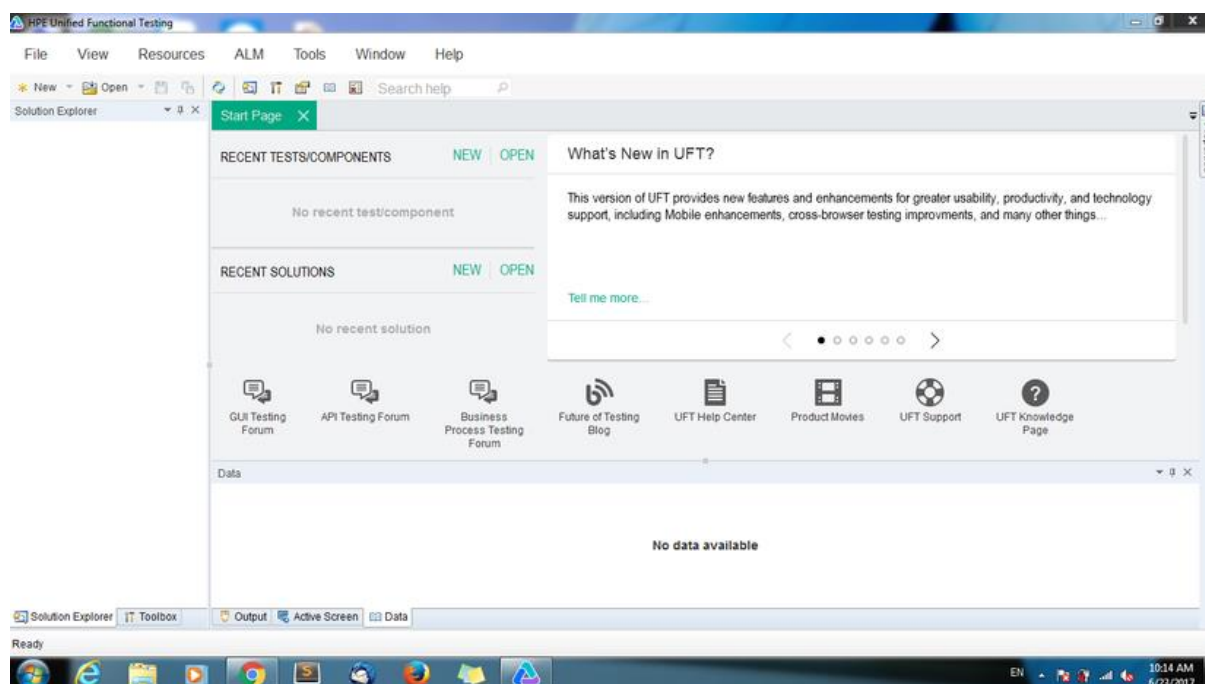
(Hình 2.5.5)

Trong quá trình chạy, chú ý tích enable các option như trong hình



(Hình 2.5.6)

B6: Mở IDE lên và sử dụng



(Hình 2.5.7)

2.6 MÔ TẢ MÔI TRƯỜNG LÀM VIỆC

QTP sử dụng ngôn ngữ VBScript để viết test script. Đây là ngôn ngữ dễ học, rất giống ngôn ngữ VBA – Visual Basic for Applications. Chế độ Expert View của QTP là chế độ soạn thảo dành cho VBScript. Ngoài việc dùng VBScript để tương tác với phần mềm được kiểm.

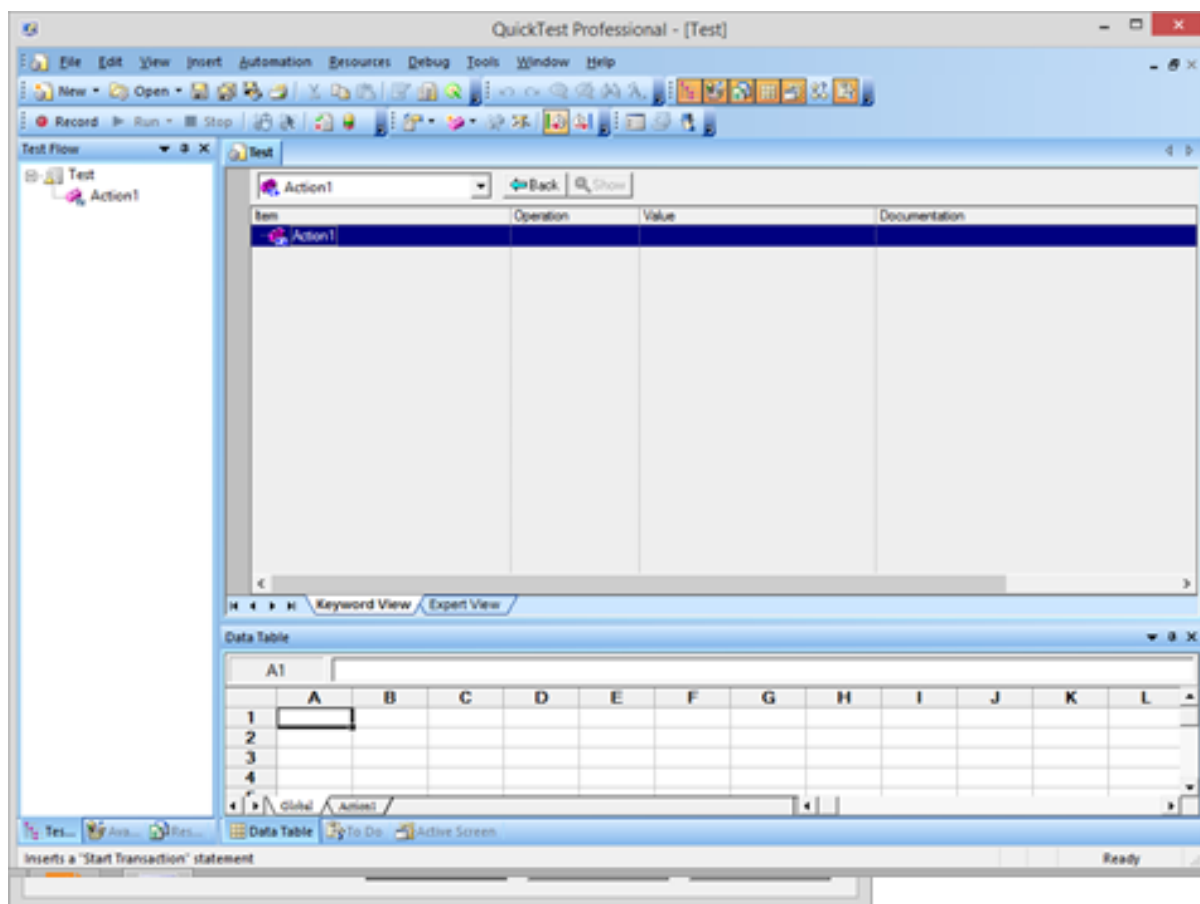
Một số màn hình làm việc của công cụ Quick Test Pro

- Add-in Manager:



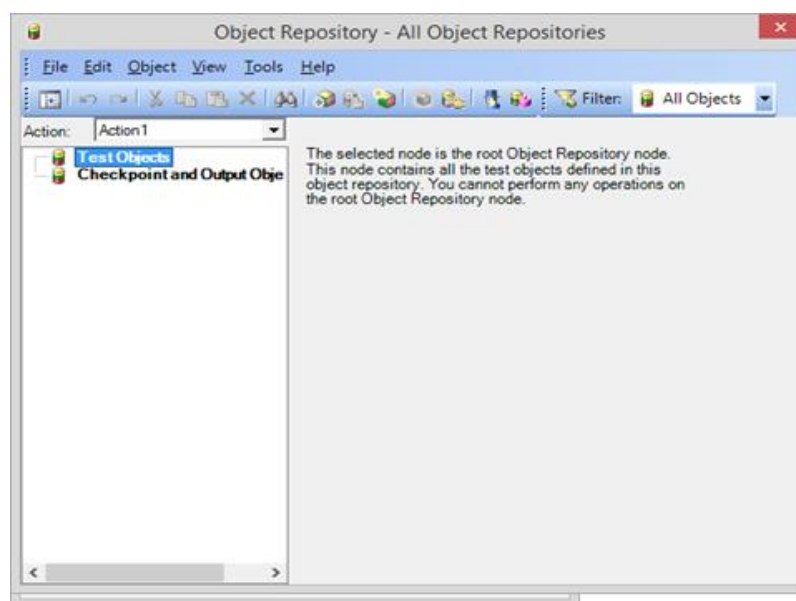
(Hình 2.6.1)

- Giao diện chính của chương trình



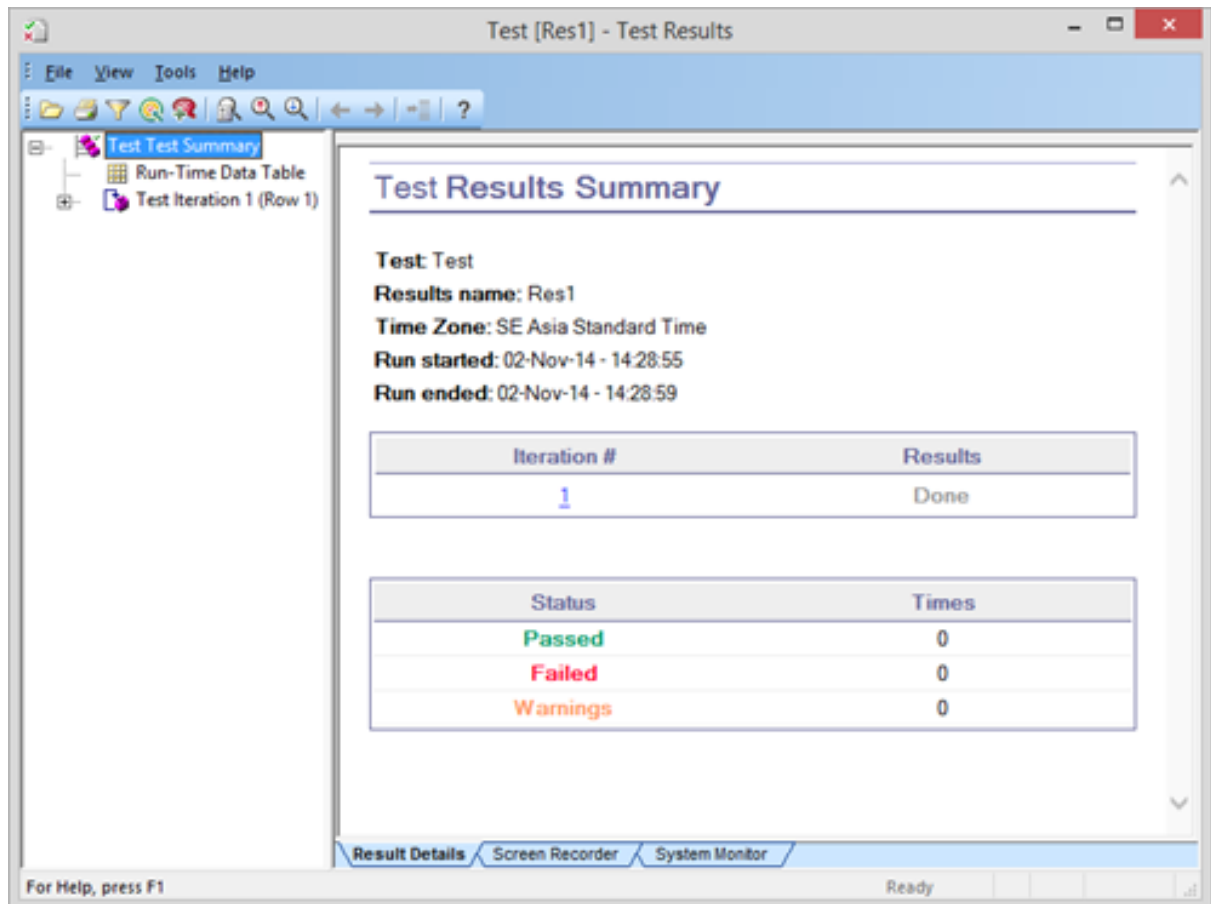
(Hình 2.6.2)

- Cửa sổ Object Repository Manager: dùng để quản lý các Object Repository



(Hình 2.6.3)

- Giao diện kết quả test



(Hình 2.6.4)

2.7 CÁC KỸ THUẬT THỰC HIỆN KIỂM THỬ CHI TIẾT

Phương pháp kiểm thử cơ bản vẫn là **record** và **playback**. Ý nghĩa thực sự của điều này là gì?

Trước tiên, hãy nói về ‘Record’. Khi một thử nghiệm khởi chạy QTP và thực hiện một loạt các hoạt động trên AUT (Ứng dụng được kiểm tra) QTP tạo ra các dòng mã tương ứng với từng hoạt động được thực hiện. Đây sẽ là kịch bản thử nghiệm cơ bản.

Chuyển sang ‘Playback’.

Khi tập lệnh thử nghiệm được tạo chạy, nó thực hiện các hoạt động chính xác giống nhau trên AUT, do đó phát lại chuỗi các bước đã được ghi lại.

Ví dụ: Khi khởi chạy QTP và cố gắng truy cập vào trang đăng nhập của một chương trình email dựa trên web, bạn nhập ID đăng nhập, Mật khẩu và nhấn nút OK. Những hành động này được ghi lại trong dòng mã VBScript tương đương của chúng trong IDE QTP. Khi lưu tập lệnh thử nghiệm này và phát lại, QTP sẽ nhập cùng một ID đăng nhập, Mật khẩu mà bạn đã nhập trước đó và nhấp vào nút OK.

Đây chính là kỹ thuật tạo thử nghiệm cơ bản, việc record và playback không phải lúc nào cũng đủ cho một kịch bản thử nghiệm mạnh mẽ. Người thử nghiệm sẽ phải sử dụng các tính năng khác trong IDE cùng với một chút kỹ năng lập trình của mình để đạt được kết quả cần thiết.

Có nhiều khía cạnh kỹ thuật khác nhau được cân nhắc trước khi **record** và **playback**, như trạng thái của AUT, đồng bộ hóa, các tùy chọn menu chính xác, mã chính xác mà QTP tạo ra. Những điều này sẽ được sáng tỏ sau khi các khái niệm cơ bản của QTP được hiểu chính xác và căn kẽ.

QTP xác định các đối tượng khác nhau trong AUT theo tên hoặc ID trình xử lý hoặc bất kỳ thuộc tính / thuộc tính duy nhất khác mà đối tượng sở hữu. Trong giai đoạn record, nó nắm bắt tất cả các thuộc tính này và trong khi phát lại, nó thực hiện các thao tác mong muốn như nhấp chuột, kiểm tra hộp kiểm, v.v... trên các đối tượng này.

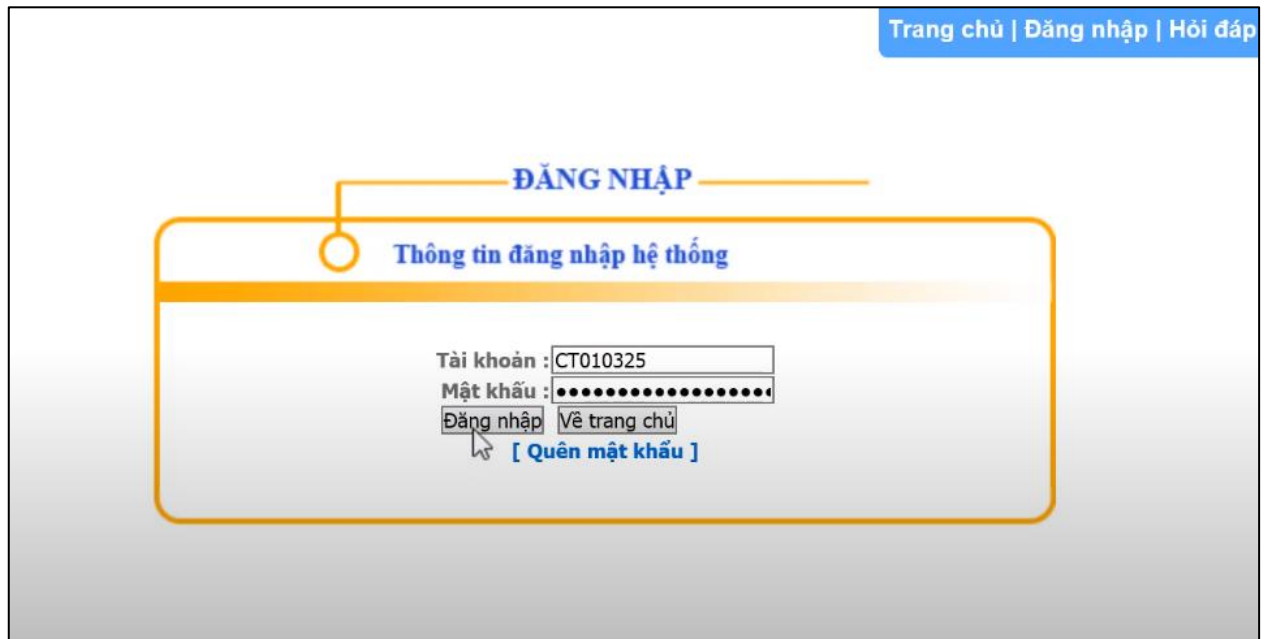
Đây cũng là công cụ áp dụng phương pháp Keyword-Driven, một kỹ thuật scripting hiện đại, cho phép kỹ thuật viên bổ sung test case bằng cách tạo file mô tả cho nó mà không cần phải chỉnh sửa hay bổ sung bất cứ script nào cả. Nó cũng phù hợp cho tình huống chuyển giao công việc mà người mới tiếp nhận chưa có thời gian hoặc không hiểu script vẫn có thể thực hiện kiểm tra phần mềm theo đúng yêu cầu.

Chương 3. MỘT SỐ KỊCH BẢN KIỂM THỬ VÀ ĐÁNH GIÁ

3.1 KỊCH BẢN KIỂM THỬ SỐ 1

+ Kịch bản số 1:

- Kịch bản kiểm thử form đăng nhập của 1 trang web:



(Hình 3.1.1)

Kiểm thử với công cụ Quick Test Professional

+ Mô tả các test case

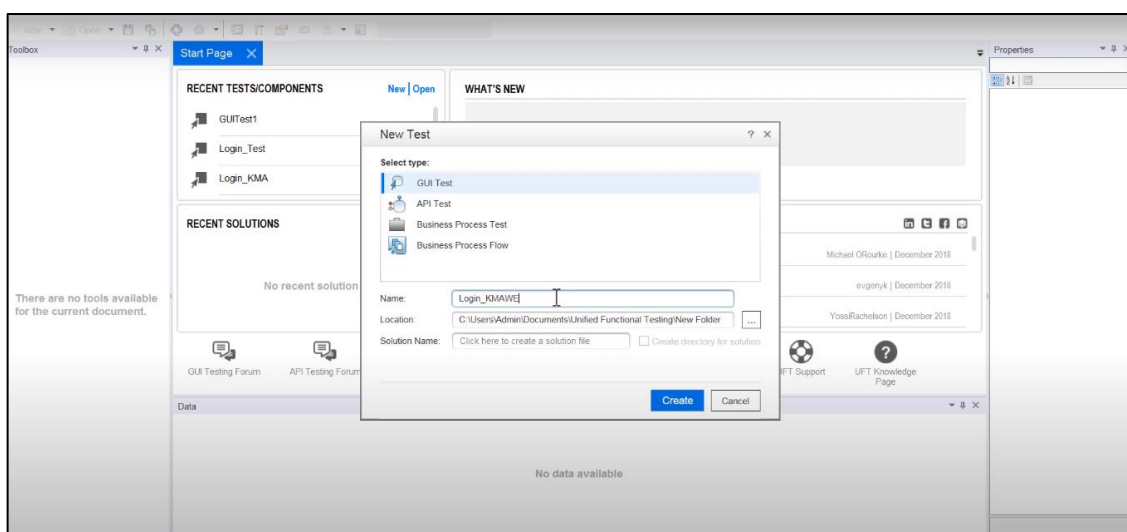
- Nhập sai tên tài khoản : thông báo tên đăng nhập không đúng
- Nhập sai mật khẩu : thông báo nhập sai tên đăng nhập hoặc mật khẩu
- Nhập sai cả tên tài khoản và mật khẩu : thông báo tên đăng nhập không đúng
- Nhập tên tài khoản bằng chữ thường : thông báo tên đăng nhập không đúng

1	UserName	Password	Thong_bao
2	CT0103254	liem1993	Tên đăng nhập không đúng!
3	CT010325	liem19934	Bạn đã nhập sai tên hoặc mật khẩu!
4	CT0103254	liem19934	Tên đăng nhập không đúng!
5	cT010325	liem1993	Tên đăng nhập không đúng!
6	Ct010325	liem1993	Tên đăng nhập không đúng!
7	ct010325	liem1993	Tên đăng nhập không đúng!
8			

(Hình 3.1.2)

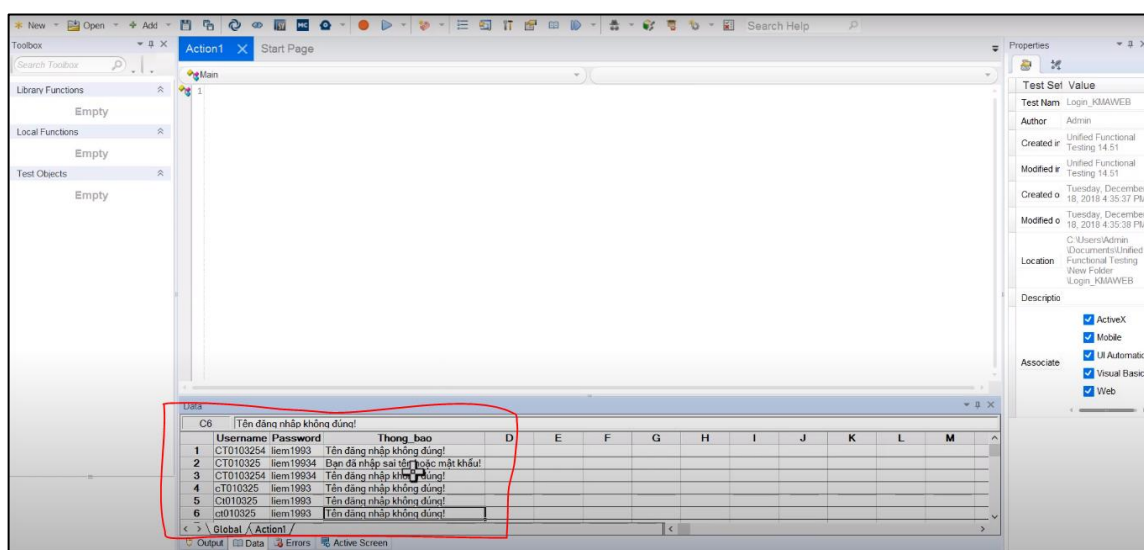
+ Các bước kiểm thử

- Tạo 1 file new Test



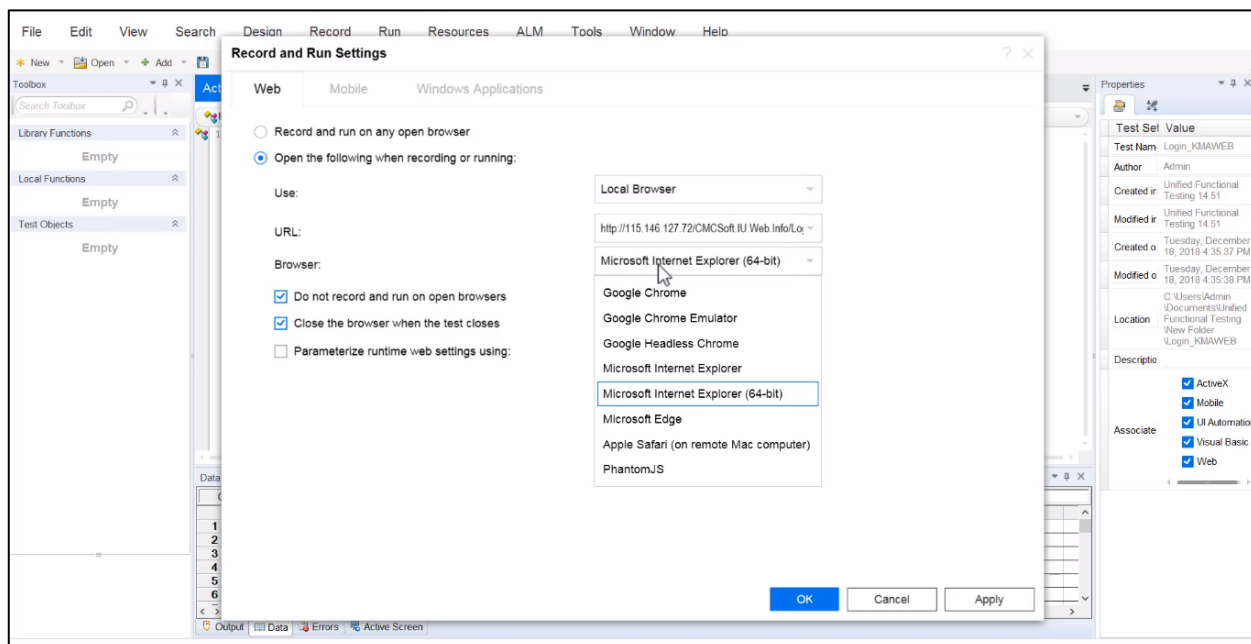
(Hình 3.1.3)

- Nhập các dữ liệu và thông báo để tiện theo dõi quá trình kiểm thử



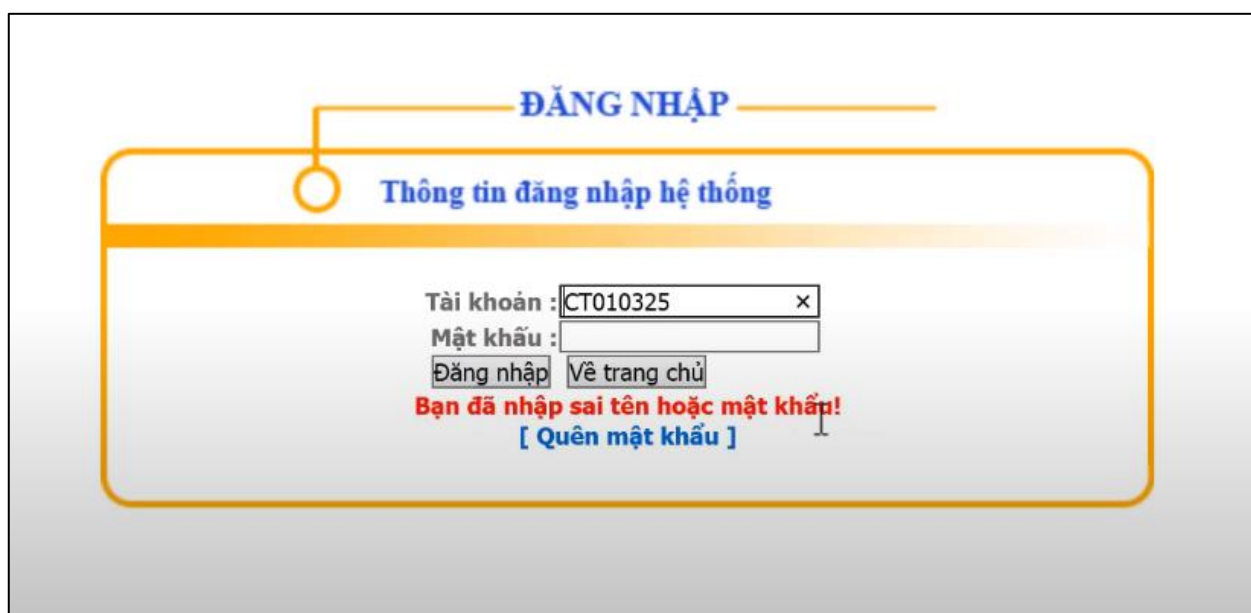
(Hình 3.1.4)

- Bấm Record -> Record and Run Setting... để cài đặt đường link website, chọn Browser muốn làm việc



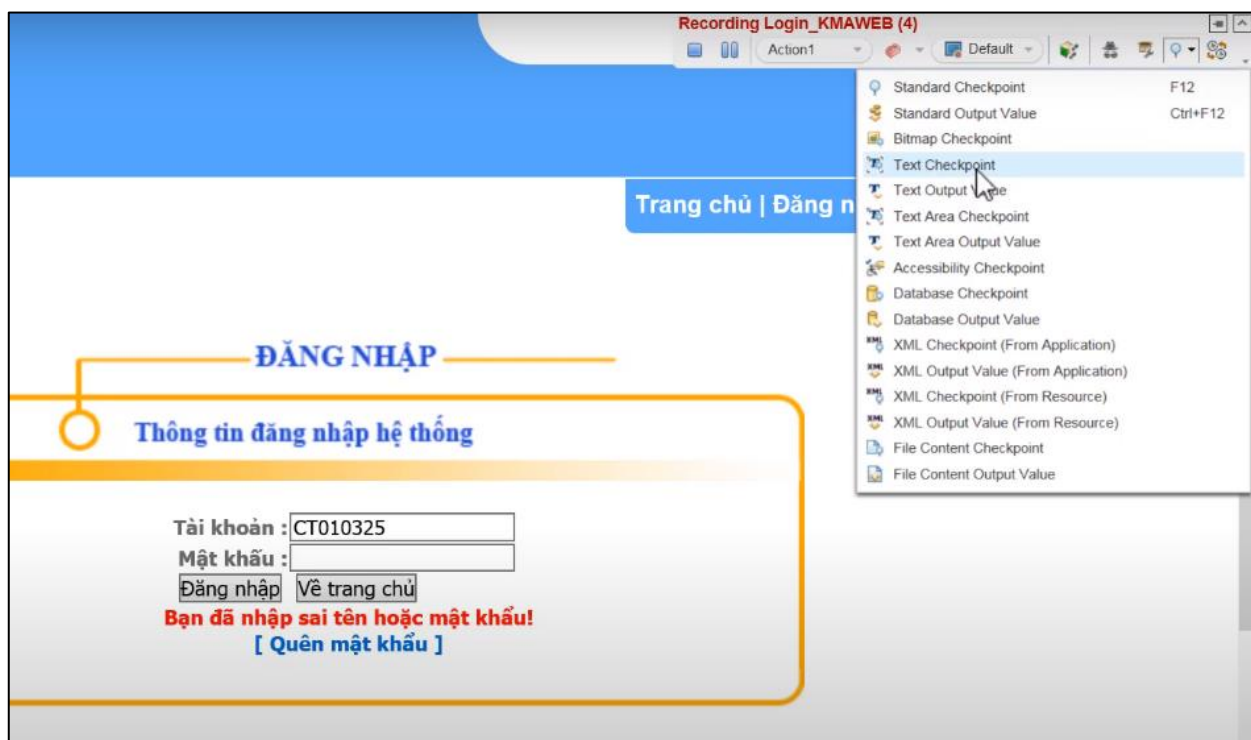
(Hình 3.1.5)

- Bấm F6 để ghi lại quá trình kiểm thử
- Tiến hành nhập tên tài khoản và mật khẩu để kiểm thử



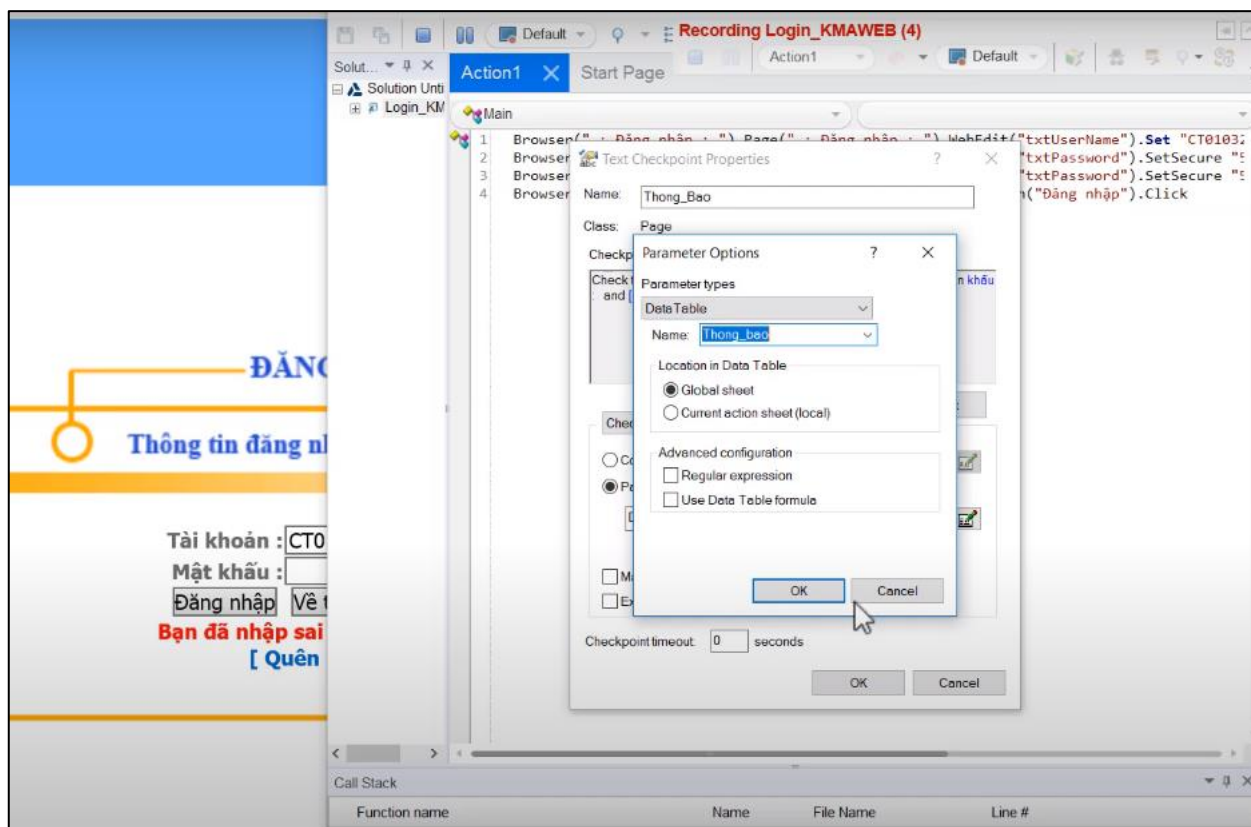
(Hình 3.1.6)

- Bôi đen dòng thông báo -> Text checkpoint để tạo so sánh



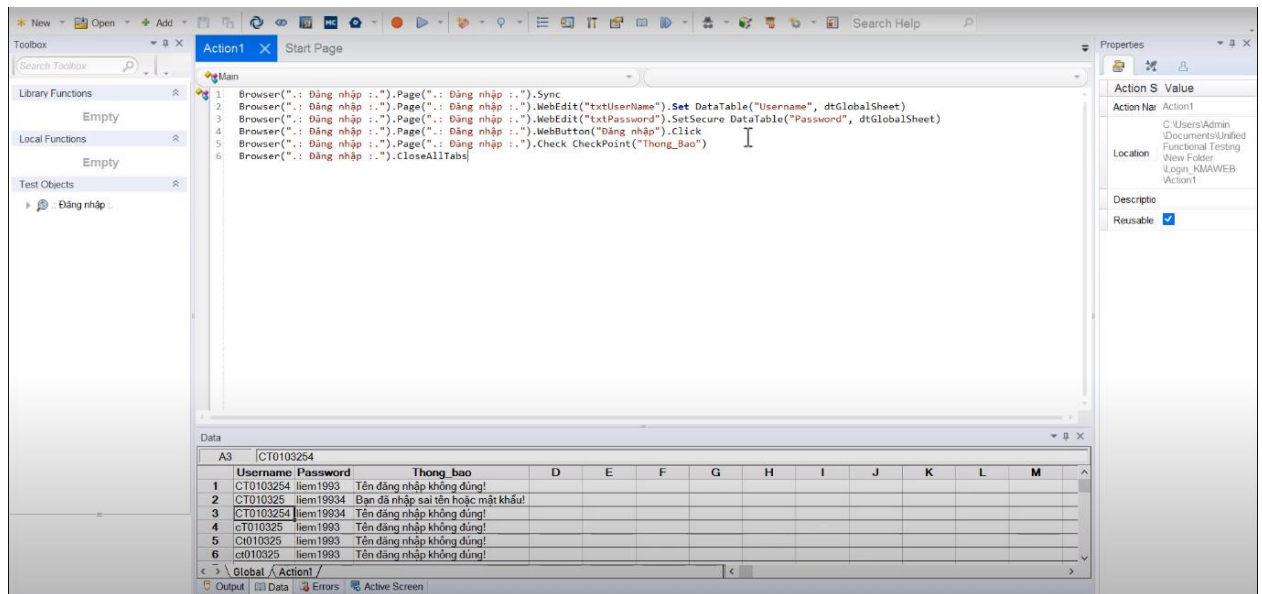
(Hình 3.1.7)

- Kích chuột vào thông báo chọn Parameter và làm như hình rồi ấn OK



(Hình 3.1.8)

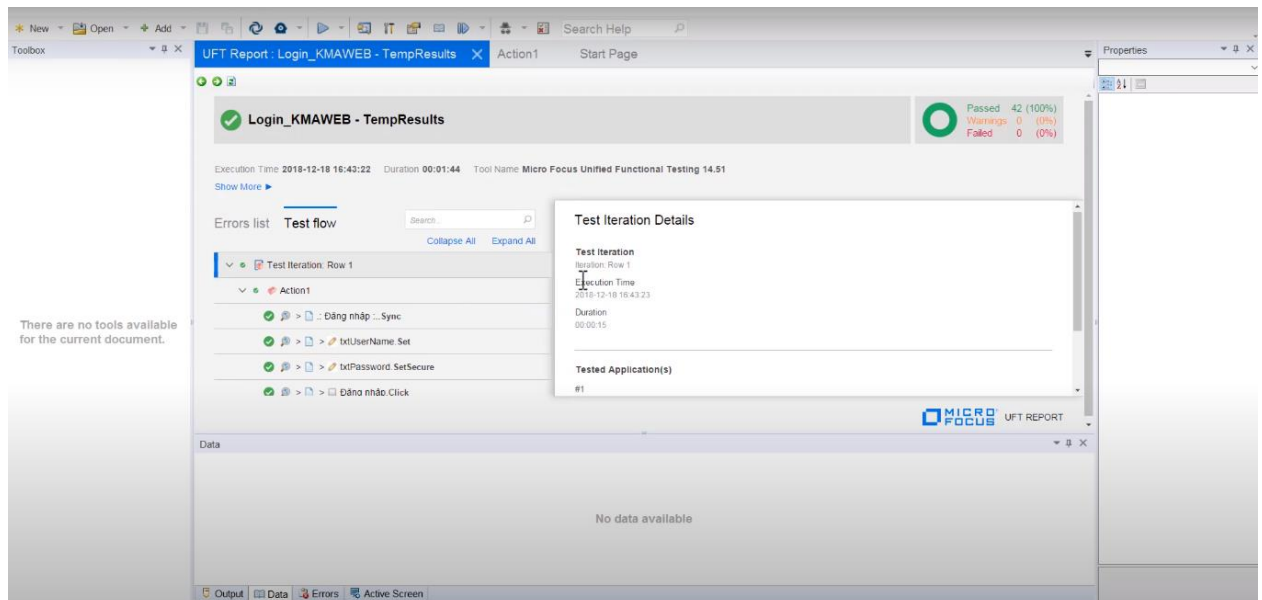
- Sau khi bấm ok màn hình trả về các công việc, test case



(Hình 3.1.9)

- Bây giờ chúng ta sẽ bấm Run hoặc Ctrl + F5 để phần mềm tự động chạy quá trình kiểm thử và các bước kiểm thử.

- Sau khi đã chạy xong nếu các test case đều đúng sẽ trả về 1 màn hình thông báo như sau, thông số được ghi bên phải gồm (Passed,Warning,Failed):



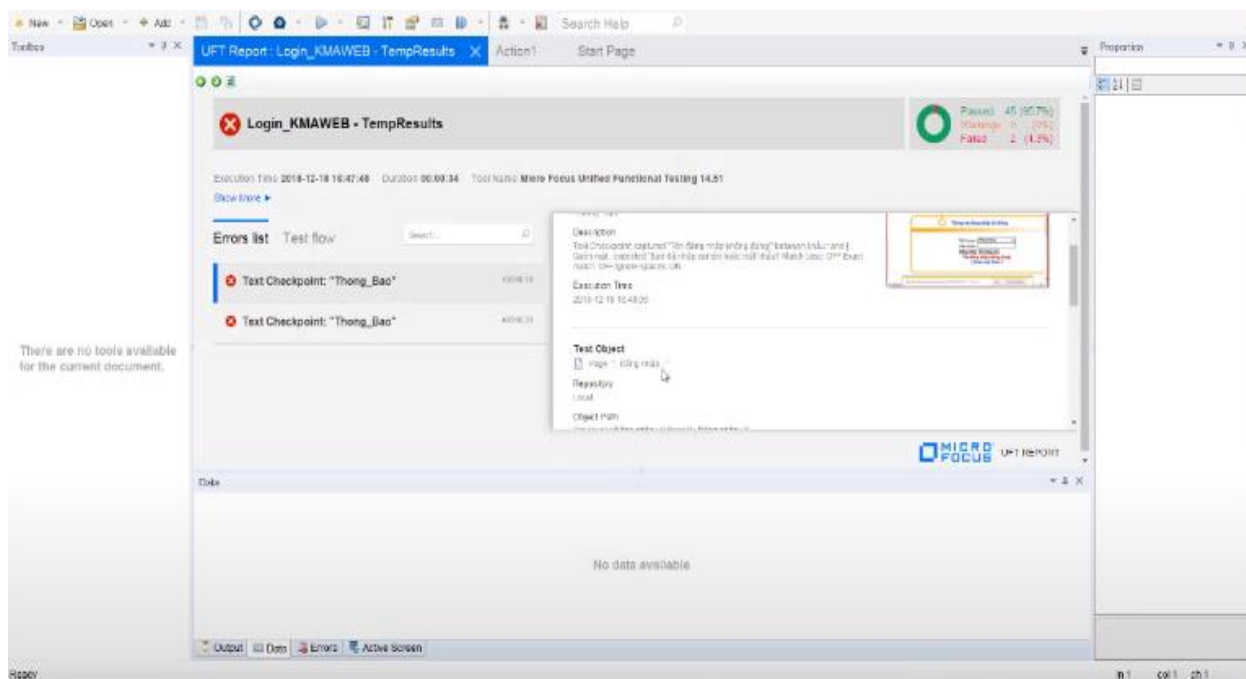
(Hình 3.1.10)

- Nếu như ta cố tình sửa lại dữ liệu thành sai để chắc chắn chính xác hơn phần mềm hoạt động tốt:

1	UserName	Password	Thong_bao	
2	CT0103254	liem1993	Bạn đã nhập sai tên hoặc mật khẩu!	
3	CT010325	liem19934	Tên đăng nhập không đúng!	
4				

(Hình 3.1.11)

- Và sau khi test ta có kết quả lỗi:



(Hình 3.1.12)

⇒ Kết Luận: Sau khi đổi dữ liệu thì phần mềm báo lỗi vậy ta phần mềm đã thực hiện test thành công.

3.2 KỊCH BẢN TEST SỐ 2

- Kịch bản số 2: Kiểm thử “Phần mềm quản lý quán café”



(Hình 3.2.1. Phần mềm quản lý quán cafe)

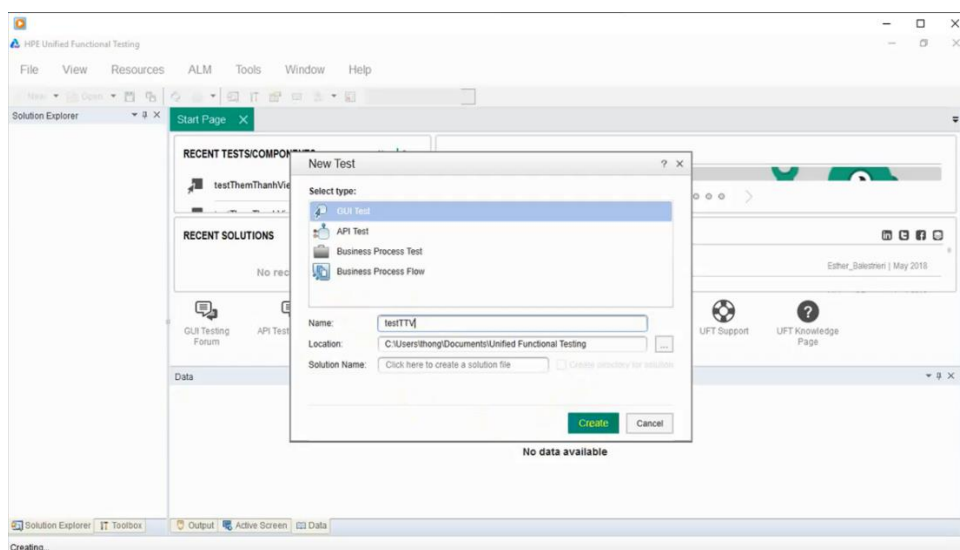
Kiểm thử với công cụ Quick Test Professional

+ Mô tả các test case

- Nhập sai loại dữ liệu dành cho chức vụ nhưng hiển thị message đúng.
- Nhập trùng dữ liệu đã có sẵn trong database và hiển thị message đúng.
- Nhập sai loại dữ liệu dành cho chức vụ và hiển thị message sai.

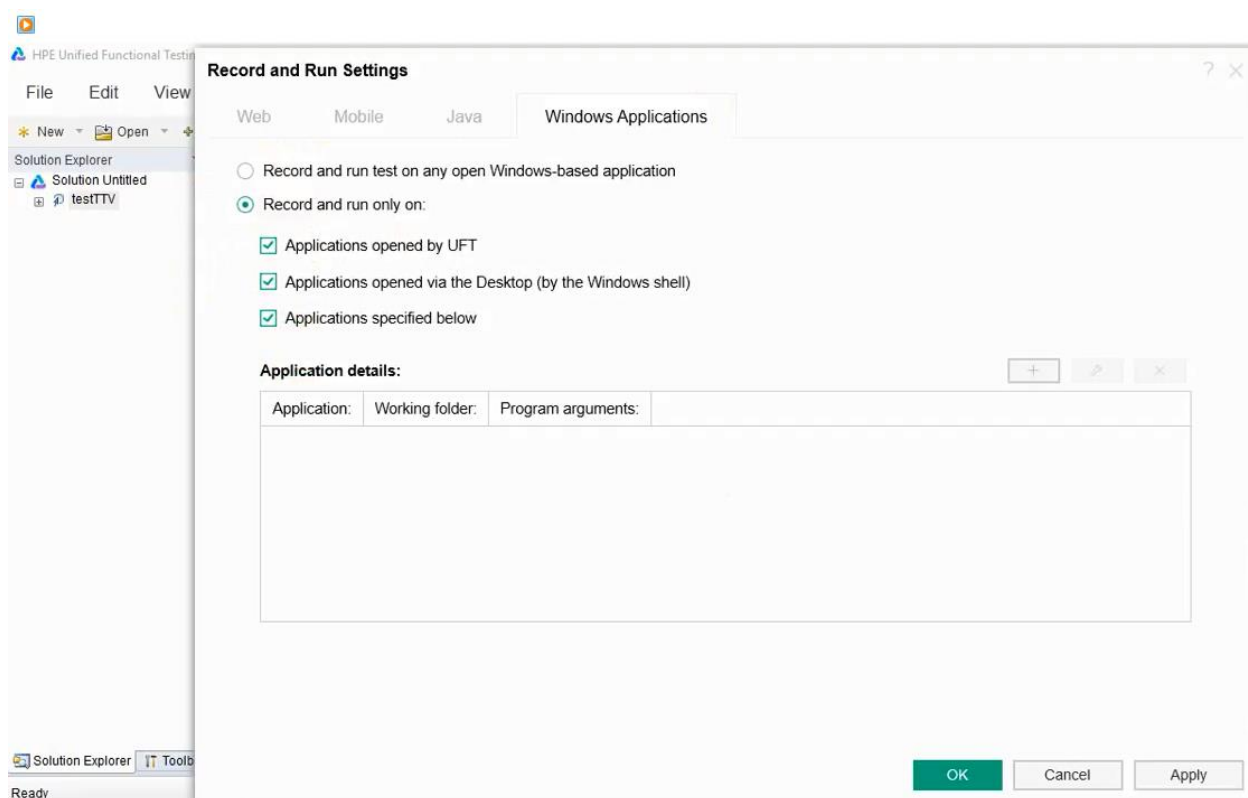
+ Các bước kiểm thử

- Tạo 1 file mới với tên testTTV



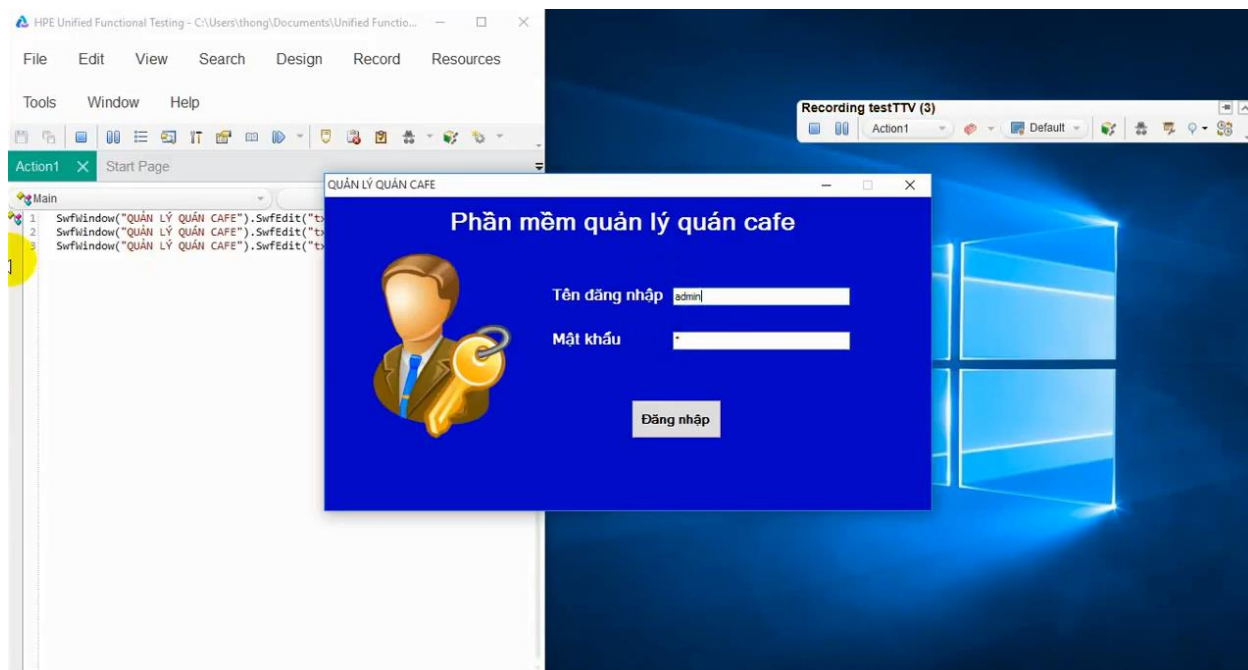
(Hình 3.2.2)

- Tạo bản ghi Windows Application:



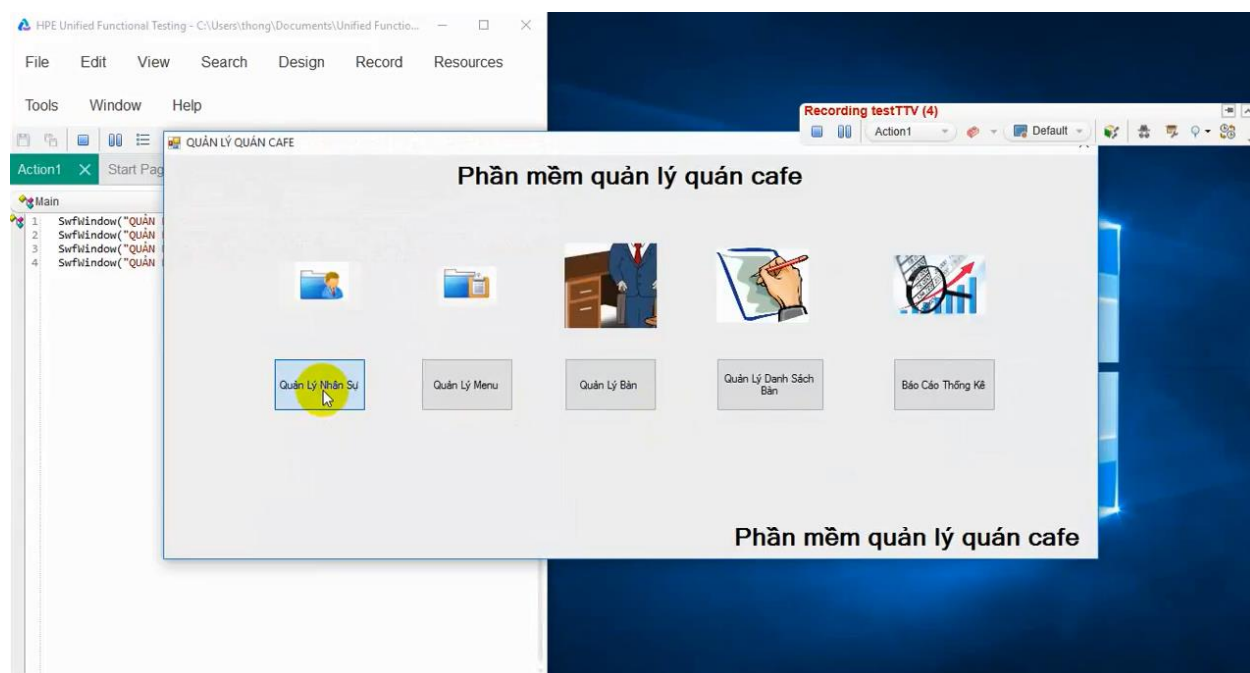
(Hình 3.2.3)

- Mở phần mềm “Quản lý quán café” để thực hiện kiểm thử:



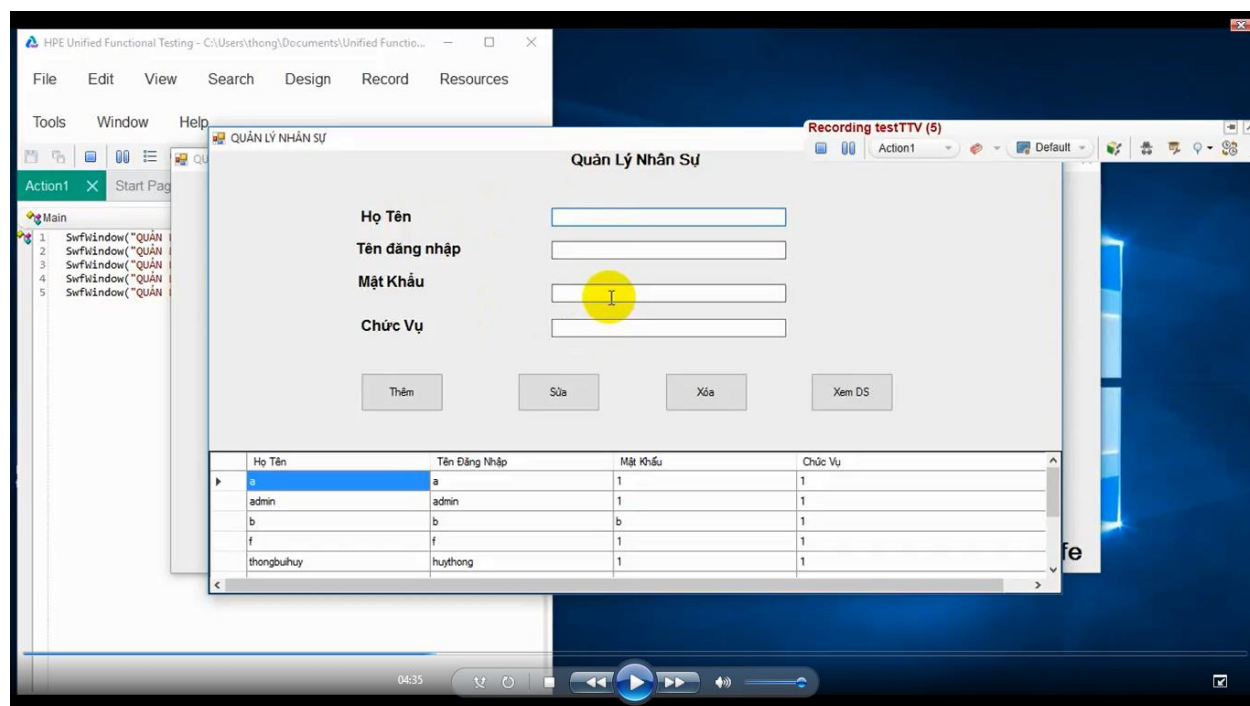
(Hình 3.2.4)

- Ấn vào chức năng quản lý nhân sự để thêm nhân sự



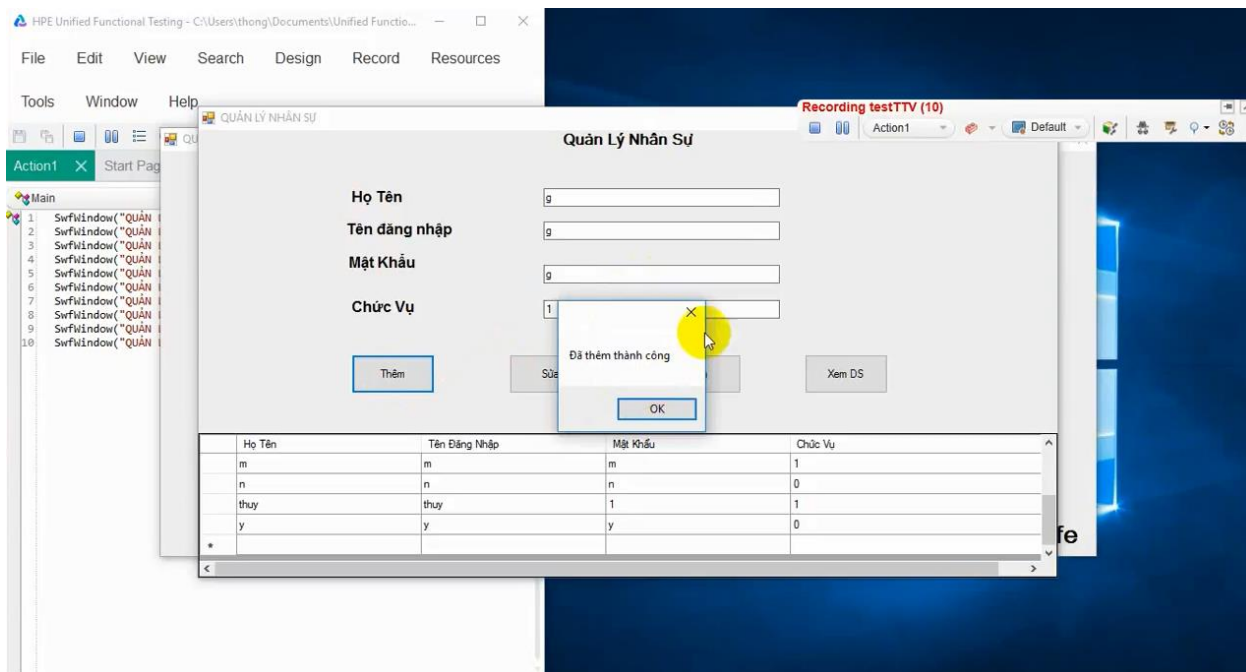
(Hình 3.2.5)

- Chức năng “Thêm” yêu cầu điều kiện các trường phải được điền đầy đủ, dữ liệu của “Chức vụ” phải là 0 hoặc 1.



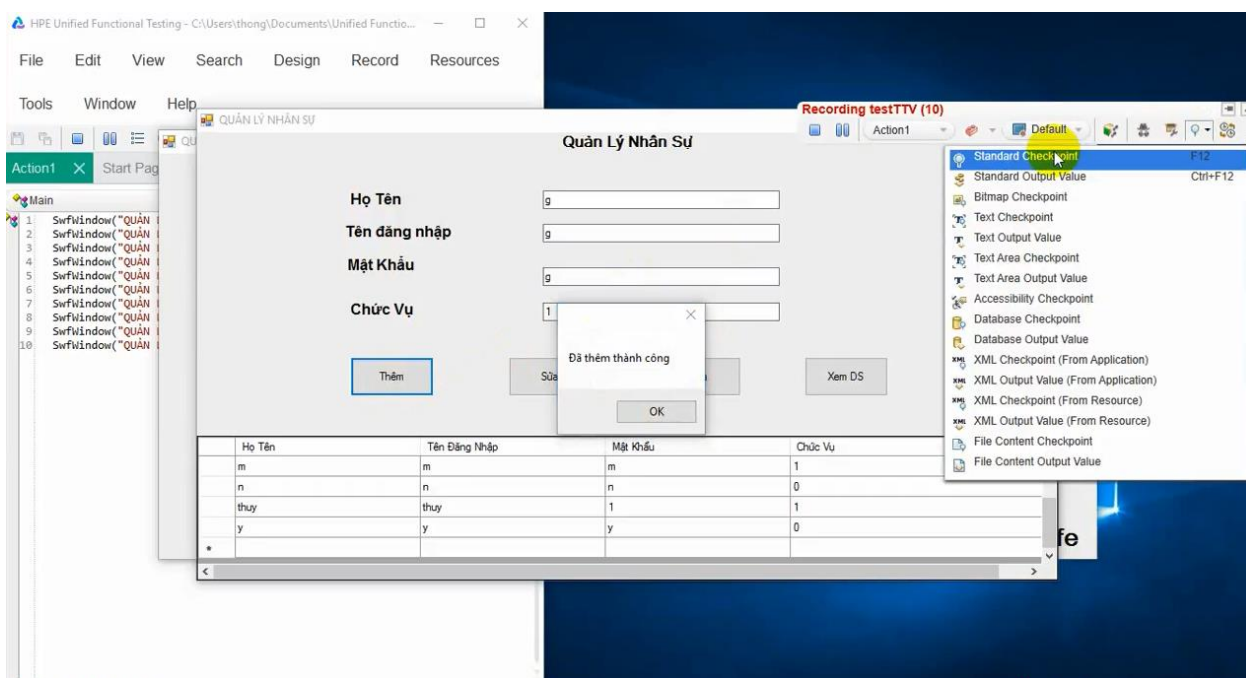
(Hình 3.2.6)

- Kiểm tra phần mềm: Thêm 1 bản ghi mới với các giá trị đúng, phần mềm ổn định với message thông báo thêm giá trị đúng là “Đã thêm thành công”:



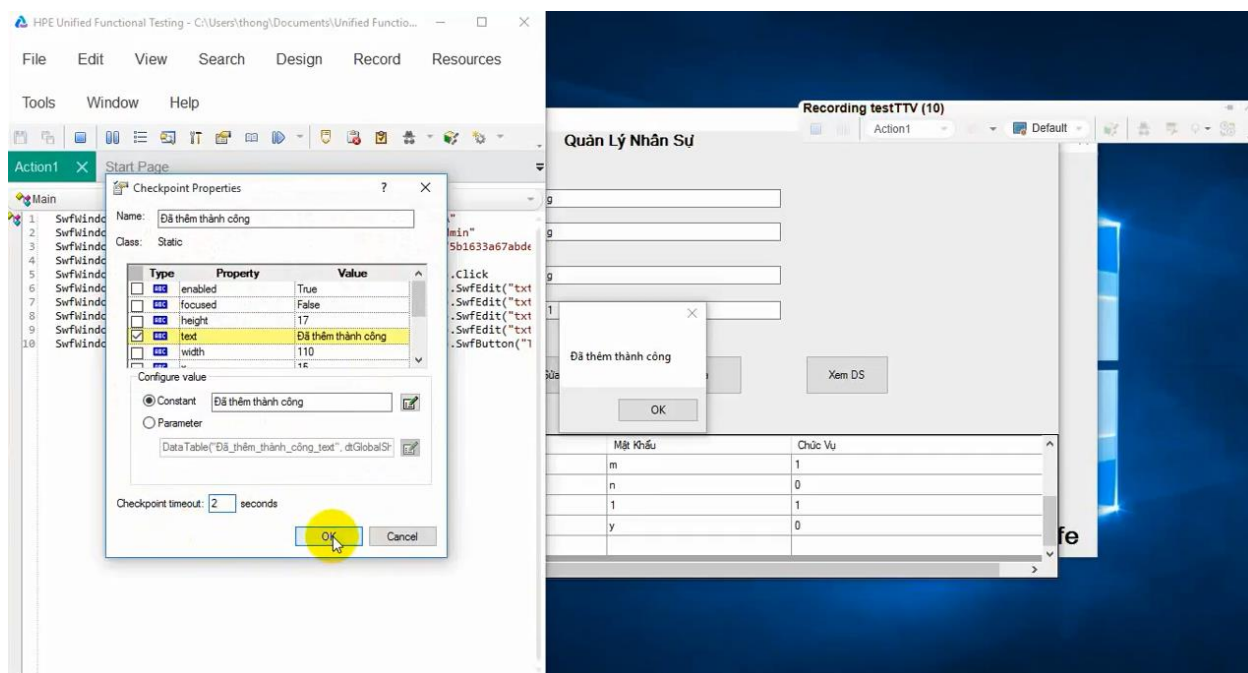
(Hình 3.2.7)

- Sử dụng Checkpoint để kiểm tra kết quả đưa ra là đúng hay sai:



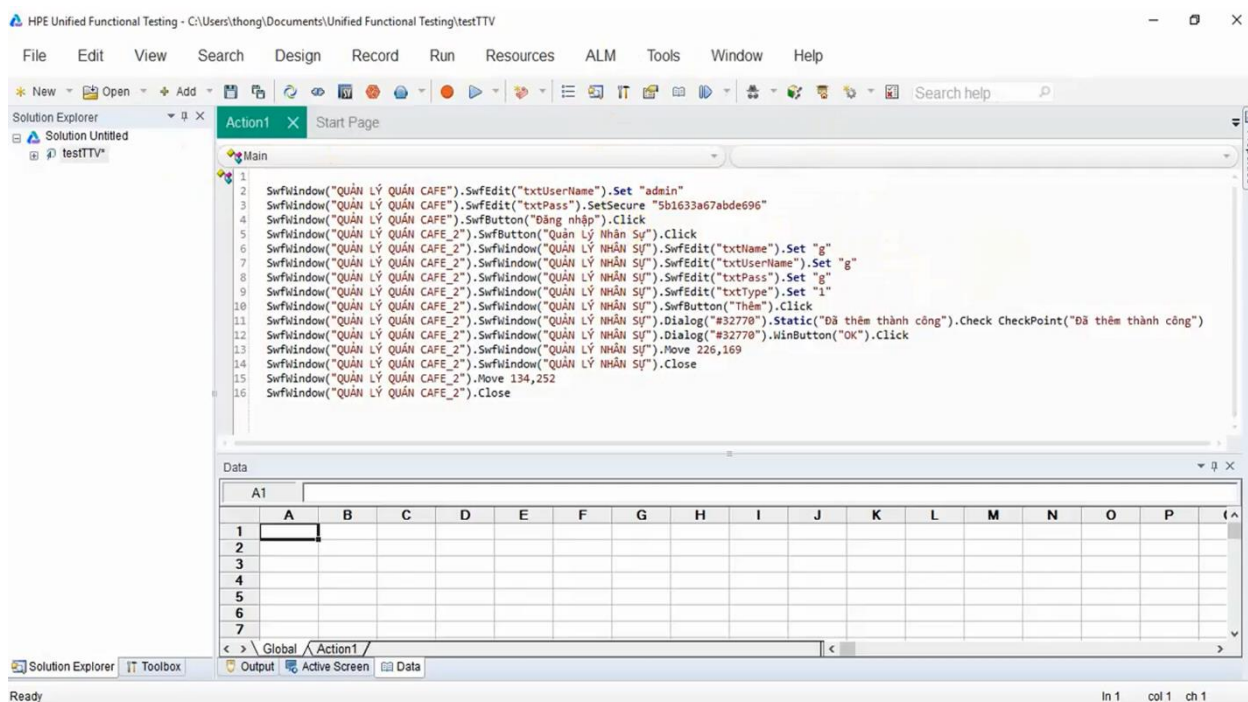
(Hình 3.2.8)

Ấn vào thông báo đã thêm thành công, bỏ tích ô đầu, chỉ để so sánh text và để thời gian checkpoint là 2s:



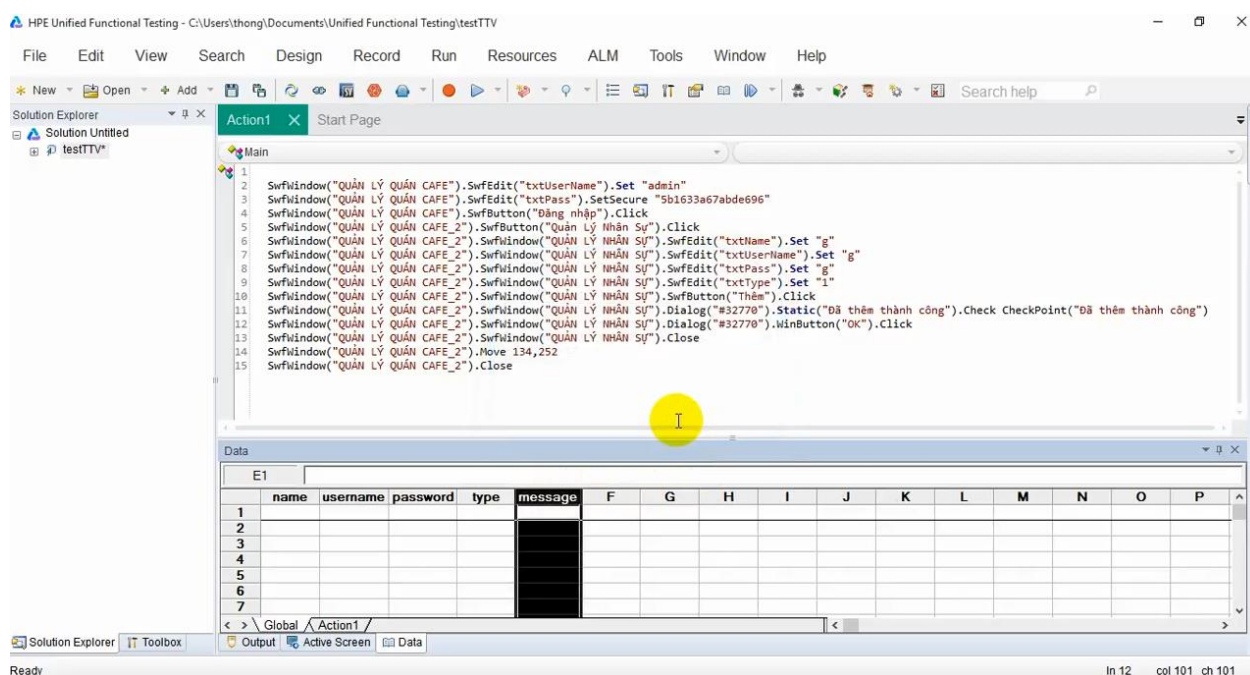
(Hình 3.2.9)

- Tắt phần mềm “Quản lý quán café” và record đi, show bản ghi của phiên làm việc lên toàn màn hình:



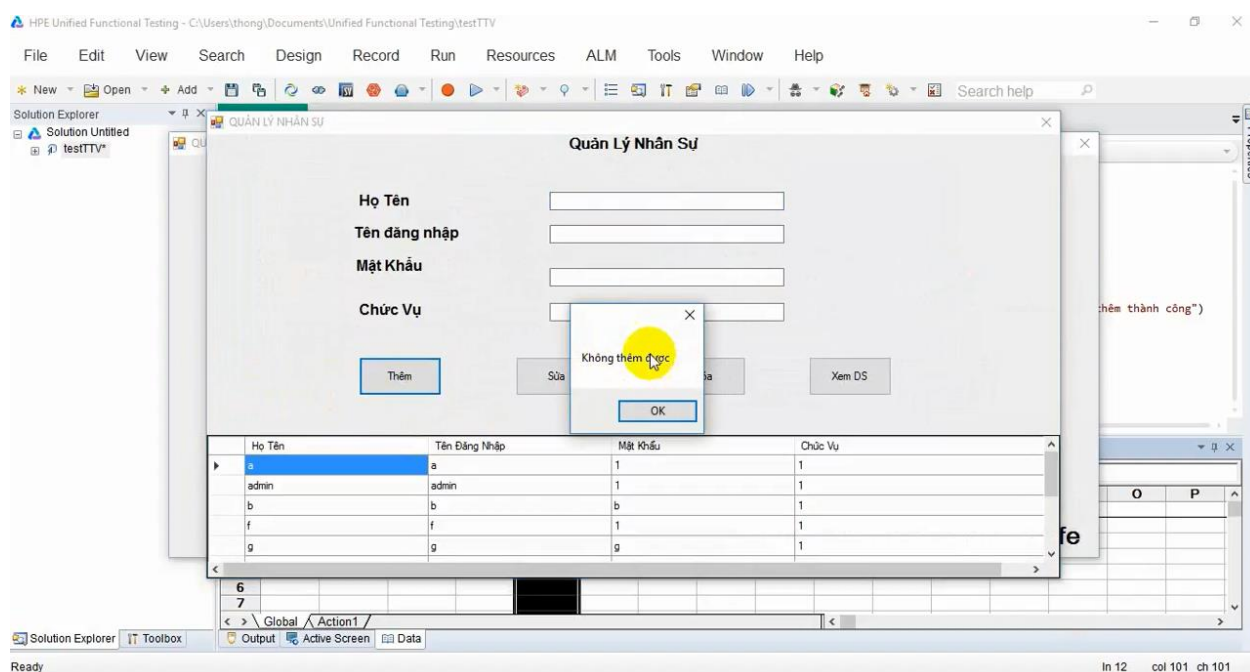
(Hình 3.2.10)

Thêm các Data Table column:



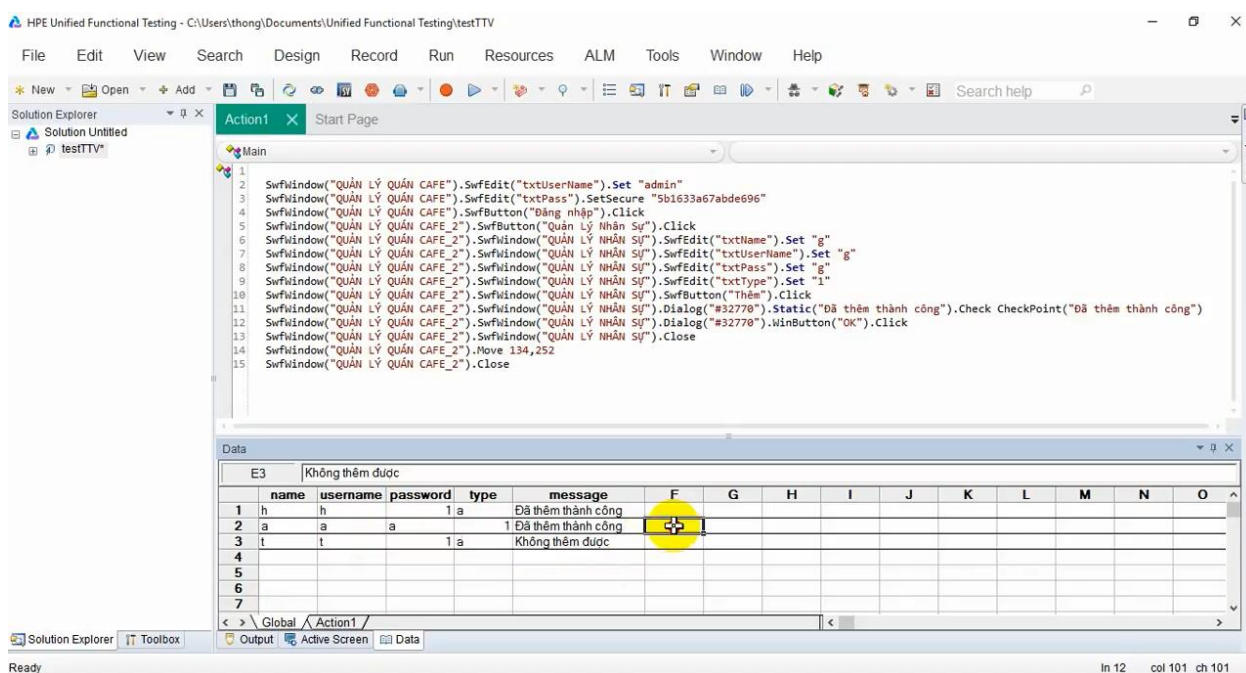
(Hình 3.2.11)

- Kiểm tra phần mềm: Thêm 1 bản ghi mới với các giá trị sai, phần mềm ổn định với message thông báo thêm giá trị sai là “Không thêm được”:



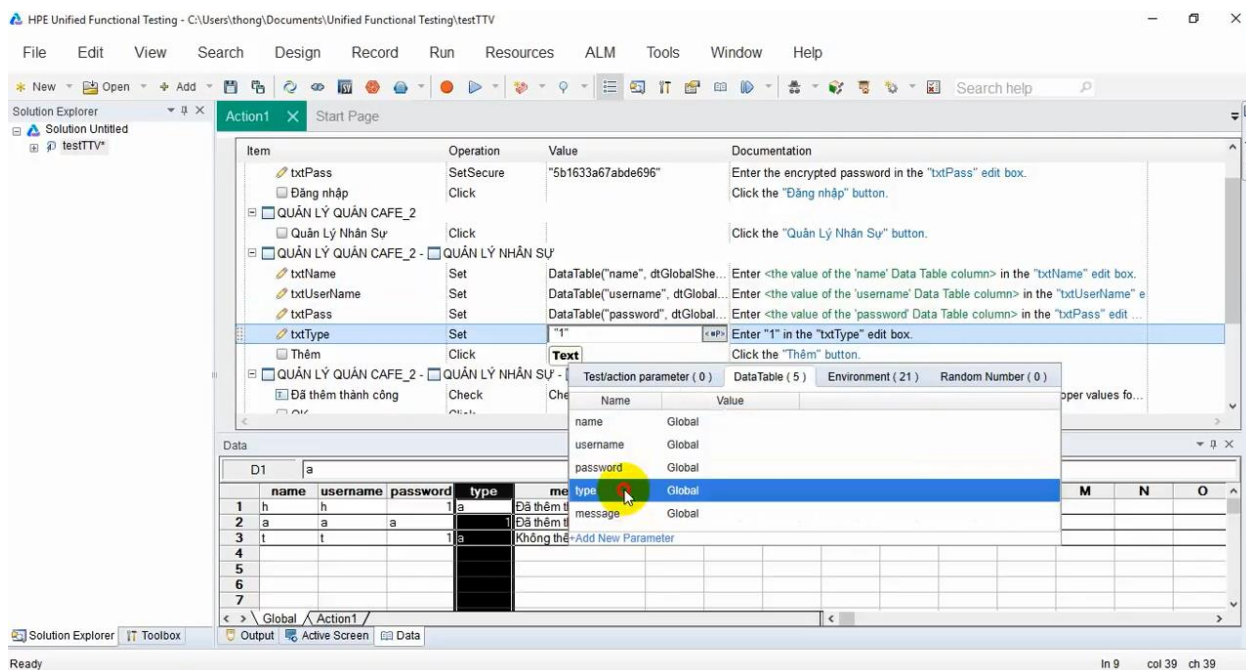
(Hình 3.2.12)

- Thêm các giá trị để kiểm thử ở Data Column Table:



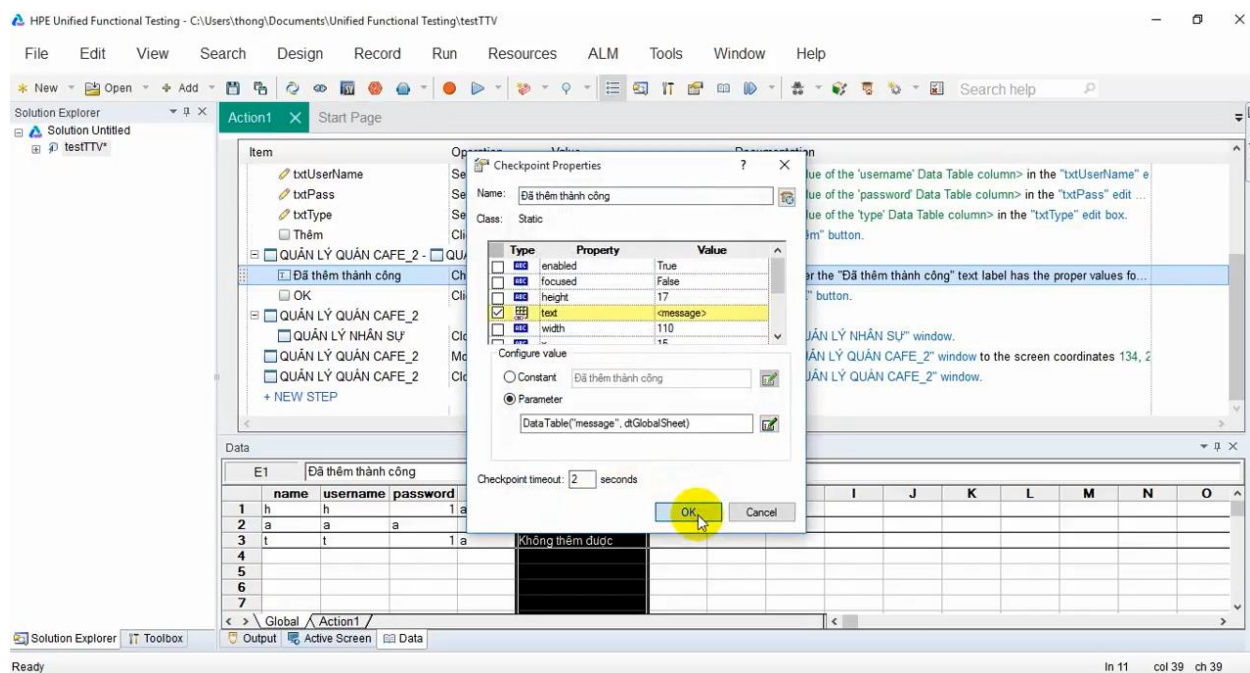
(Hình 3.2.13)

- Sau đó set giá trị của Column Table tương ứng với giá trị trong thêm thành viên của phần mềm “Quản lý quán café”:



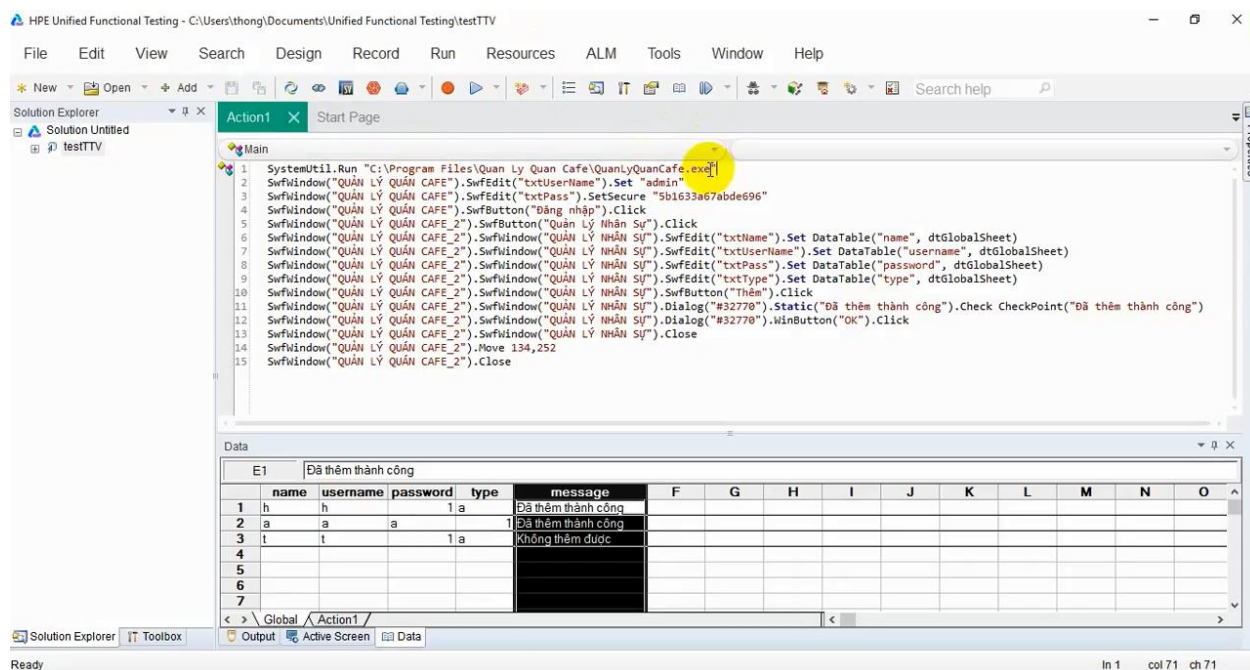
(Hình 3.2.14)

- Set checkpoint để so sánh nội dung hiển thị của message ta cài và message của “phần mềm quản lý quán cafe” :



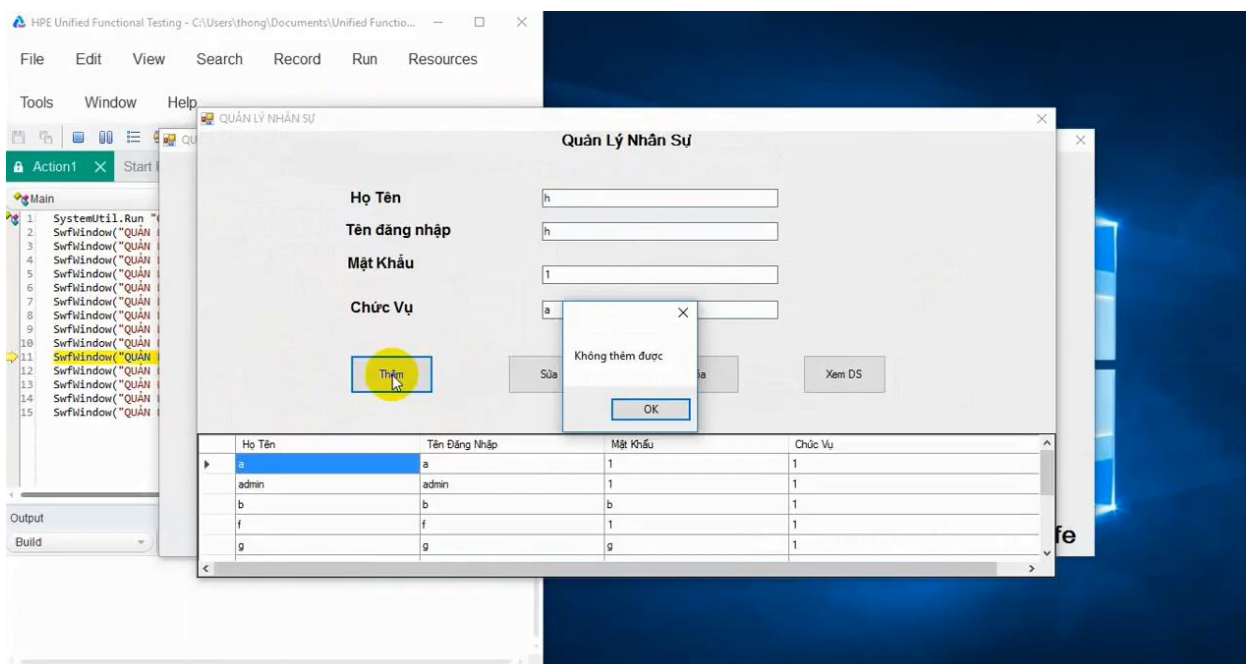
(Hình 3.2.15)

- Set thêm câu lệnh SystemUtil.Run để tự động khởi chạy phần mềm “Quản lý quán cafe” và thực hiện test case:

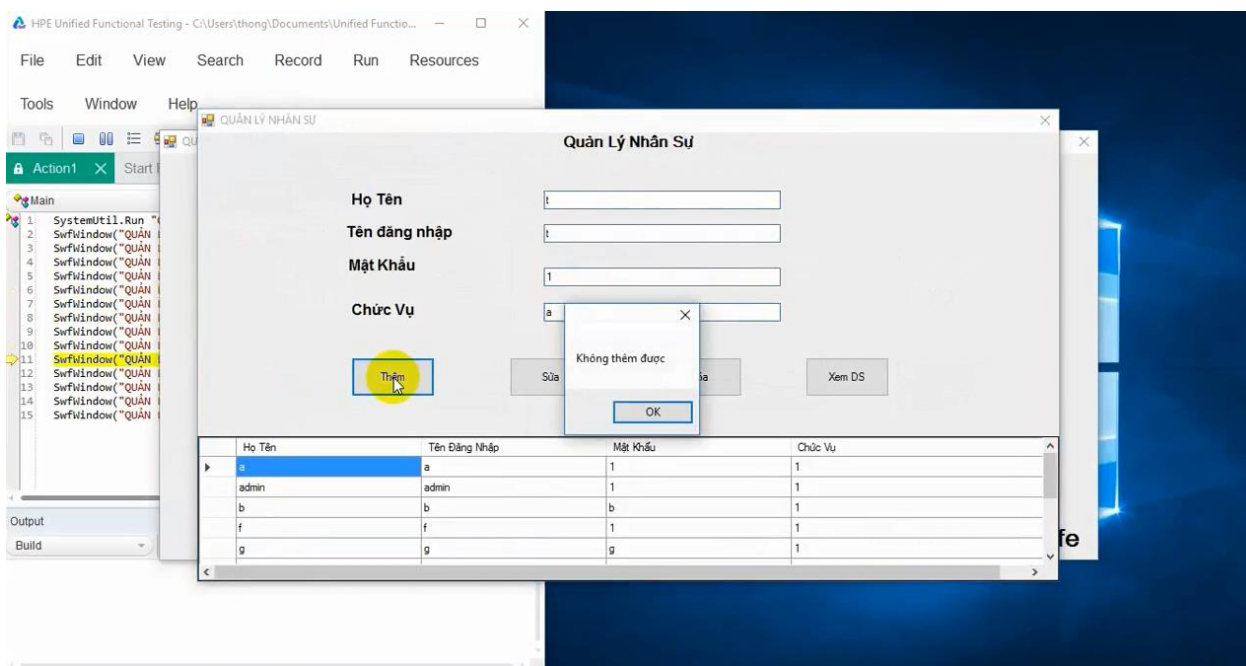


(Hình 3.2.16)

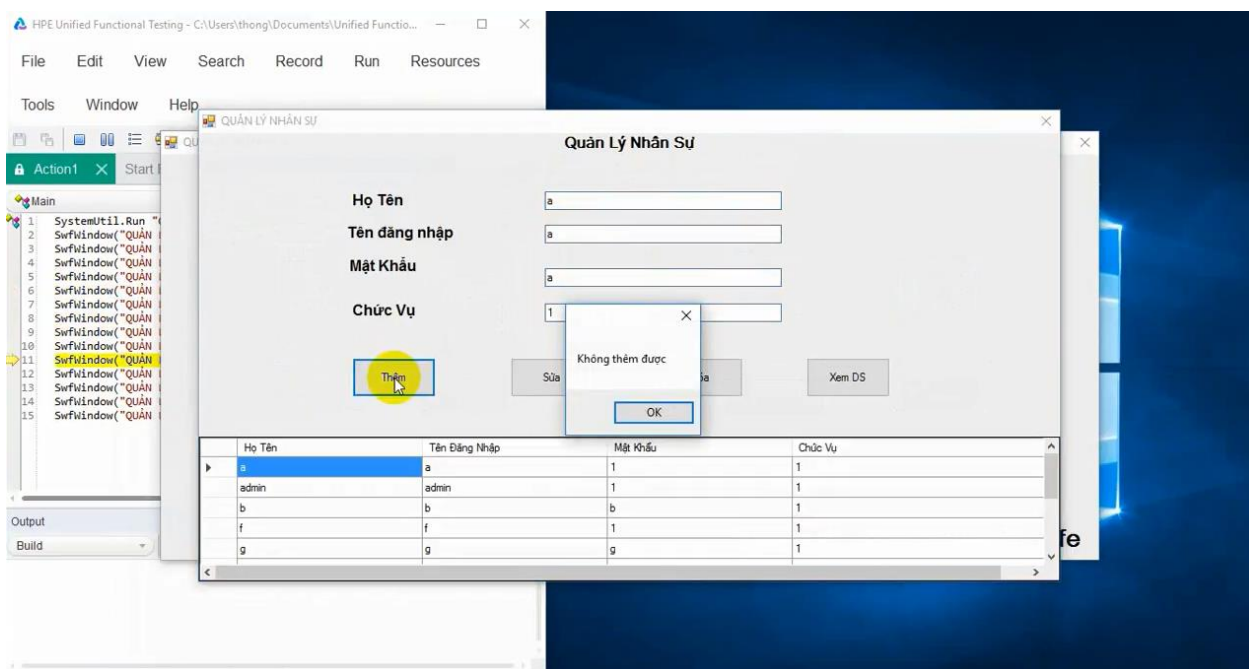
- Sau đó ấn F5 để tự động chạy, kết quả chạy test case:



(Hình 3.2.17)

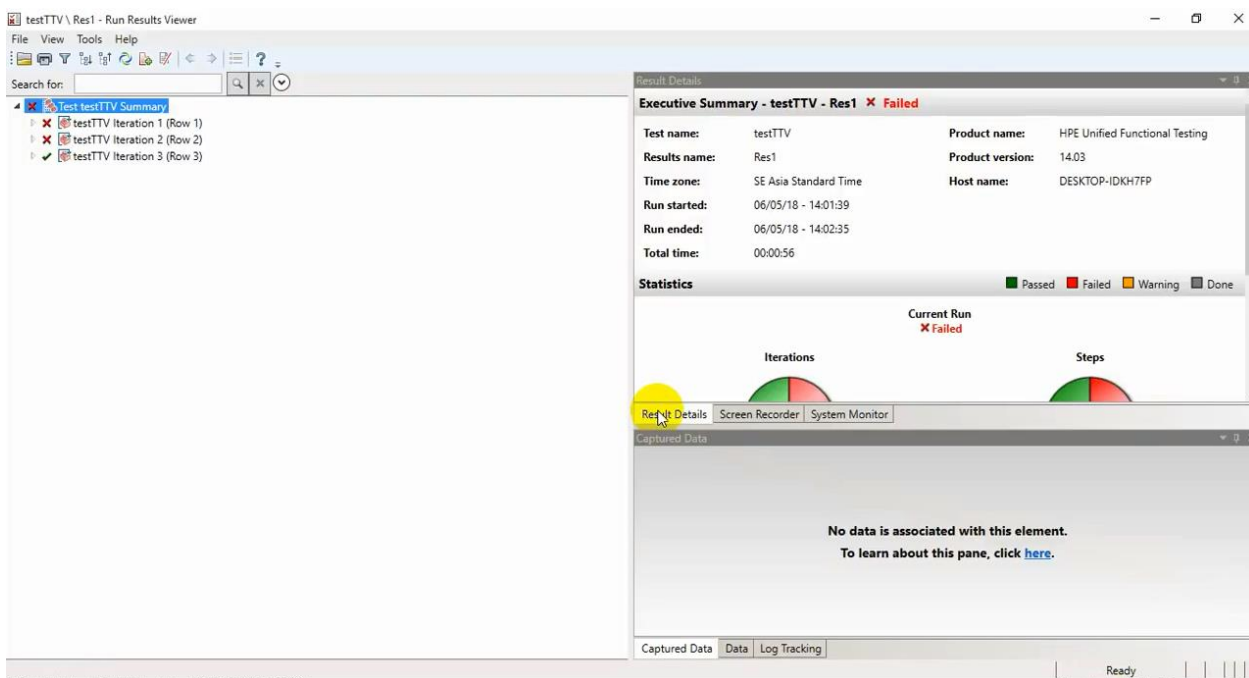


(Hình 3.2.18)

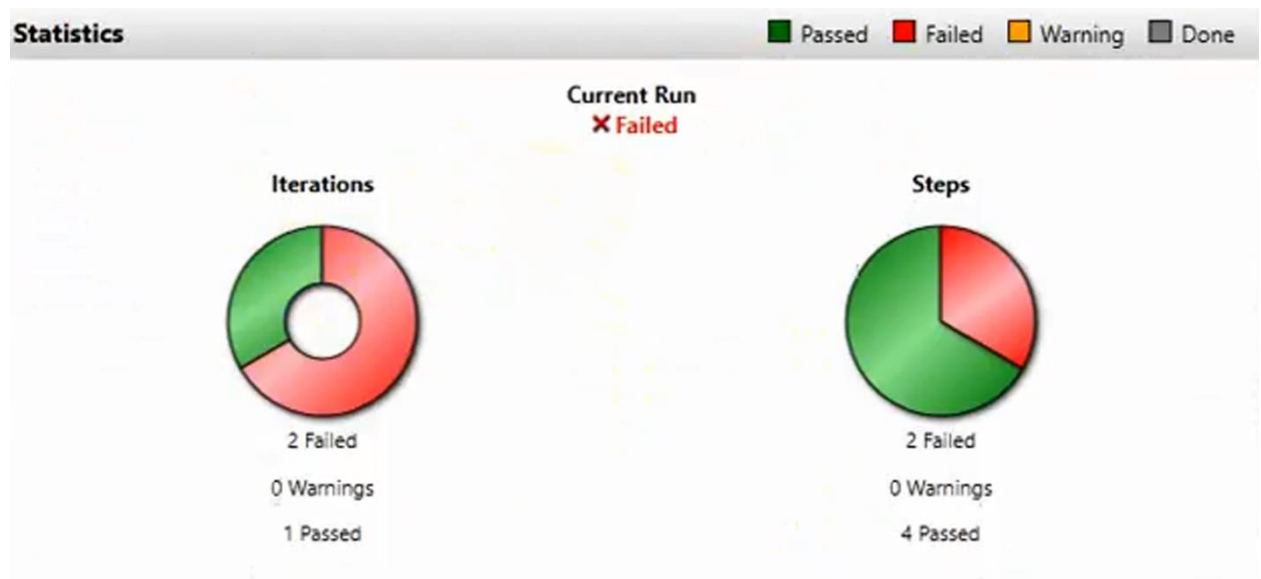


(Hình 3.2.19)

- Sau khi đã chạy xong sẽ trả về kết quả của 3 test case, trong đó có 2 test case sai là của dòng 1 và 2, test case đúng là của dòng 3:



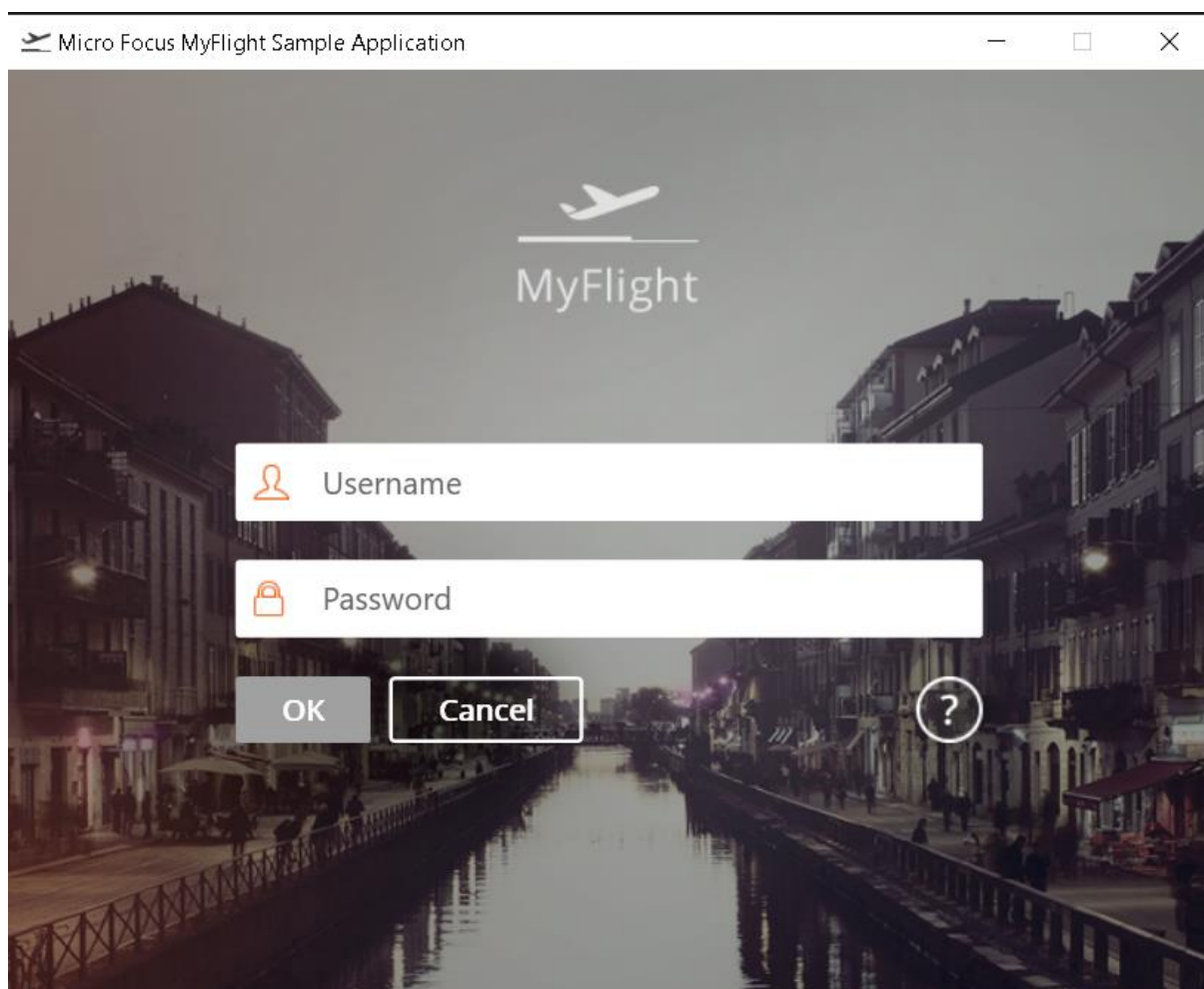
(Hình 3.2.20)



(Hình 3.2.21)

- Test case 1 sai vì giá trị thêm type – tương ứng với “Chức vụ” là a - sai với quy định của “Phần mềm quản lý quán café” là 0 hoặc 1 trong khi đó message ta chọn là “Đã thêm thành công” => Test case sai
 - Test case 2 sai vì user name a đã tồn tại trong cơ sở dữ liệu, trong khi đó message ta chọn là “Đã thêm thành công” => Test case sai
 - Test case 3 giá trị của type – tương ứng với “Chức vụ” ta chọn là a – sai với quy định của “Phần mềm quản lý quán café” là 0 hoặc 1 – message ta chọn là “Không thêm được”, đúng với message của “Phần mềm quản lý quán café” => Test case đúng.
- ⇒ **Kết luận:** Ta đã thực hiện test phần mềm thành công.

3.3 KỊCH BẢN TEST SỐ 3



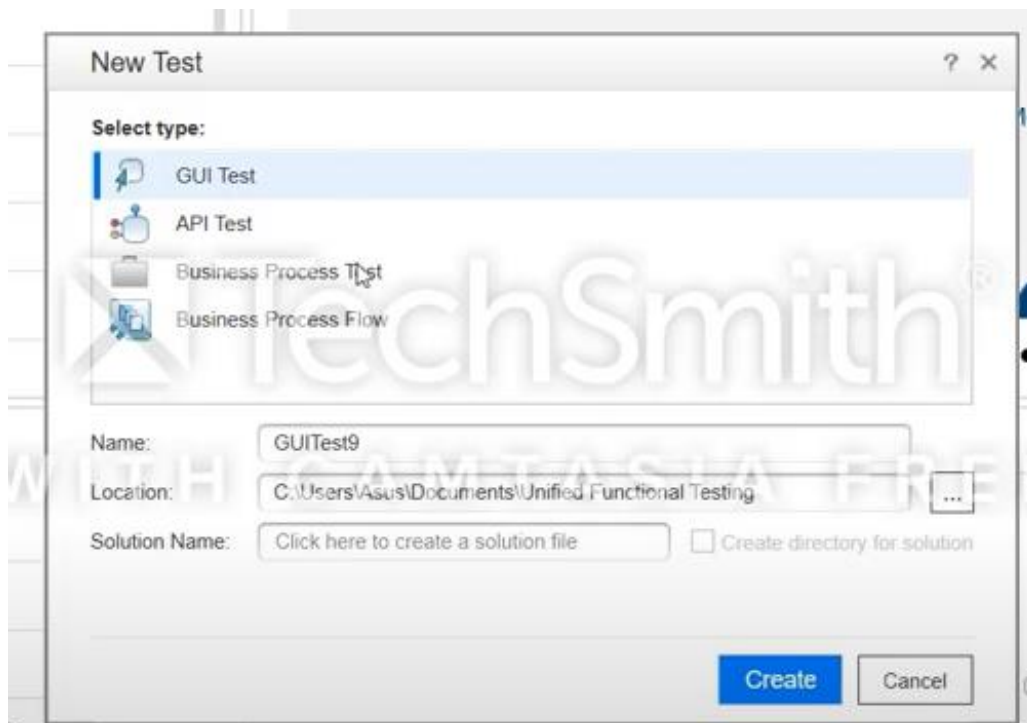
(Hình 3.3.1. Giao diện đăng nhập của phần mềm)

+ Mô tả các trường hợp để kiểm thử

- Sai tài khoản ⇔ Báo lỗi đăng nhập!
- Sai mật khẩu ⇔ Báo lỗi đăng nhập!
- Nhập sai toàn bộ ⇔ Báo lỗi đăng nhập!

+ Các thao tác thực hiện kiểm thử

➤ Tạo 1 file mới với tên Login_Test



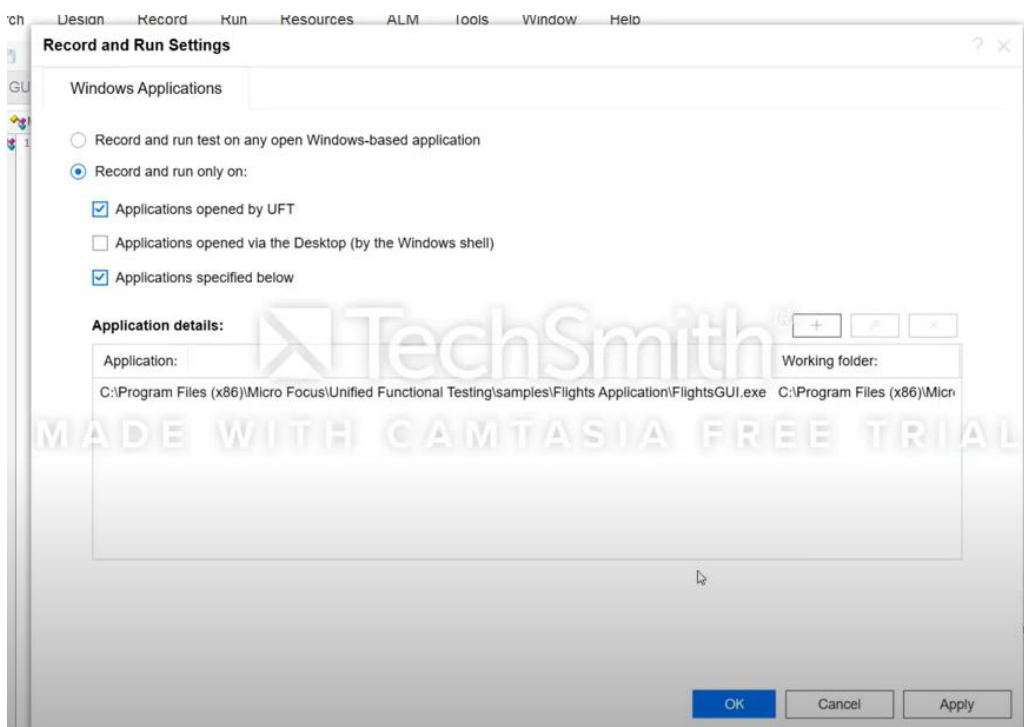
(Hình 3.3.2)

➤ Tạo bảng dữ liệu để thực hiện các ca kiểm thử (có thêm cả dữ liệu bên trong):

B3	qtp			
	agentName	password	C	D
1	abcd	abcd		
2	aaaa	aaaa		
3	uft	qtp		
4				
5				
6				
7				
8				

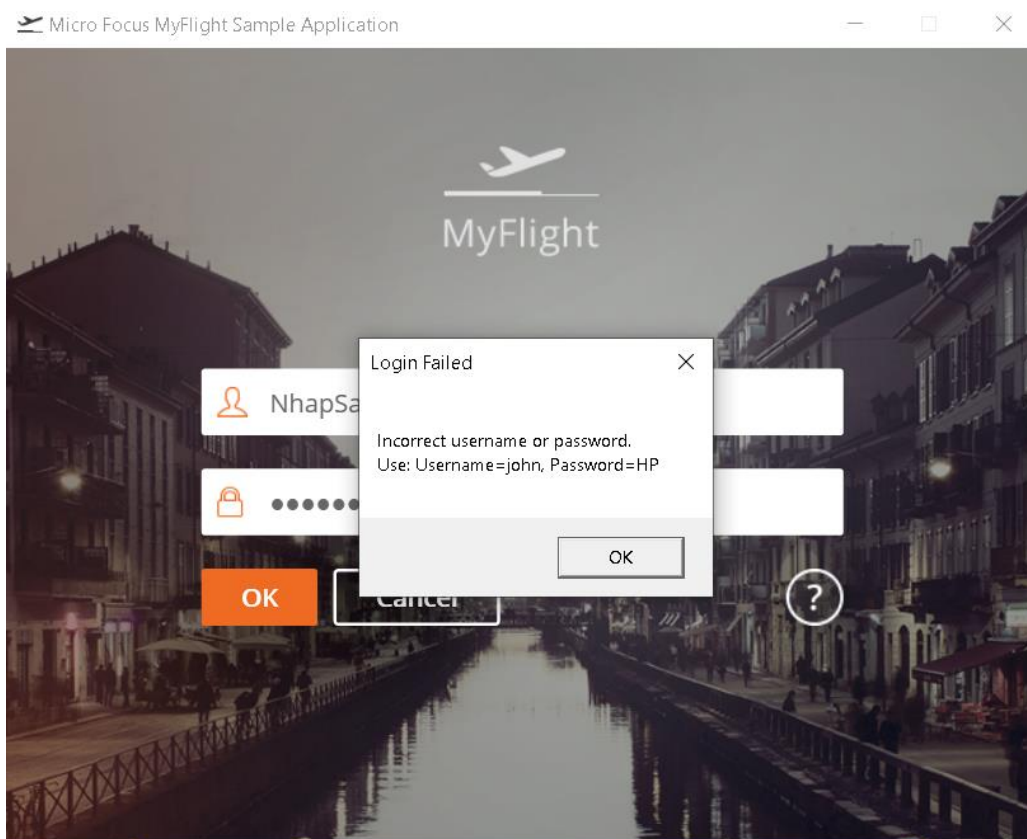
(Hình 3.3.3. Ảnh bảng dữ liệu kiểm thử)

➤ Cài đặt record để ghi lại quá trình kiểm thử:



(Hình 3.3.4)

➤ Kiểm tra: đăng nhập thử với trường hợp sai password



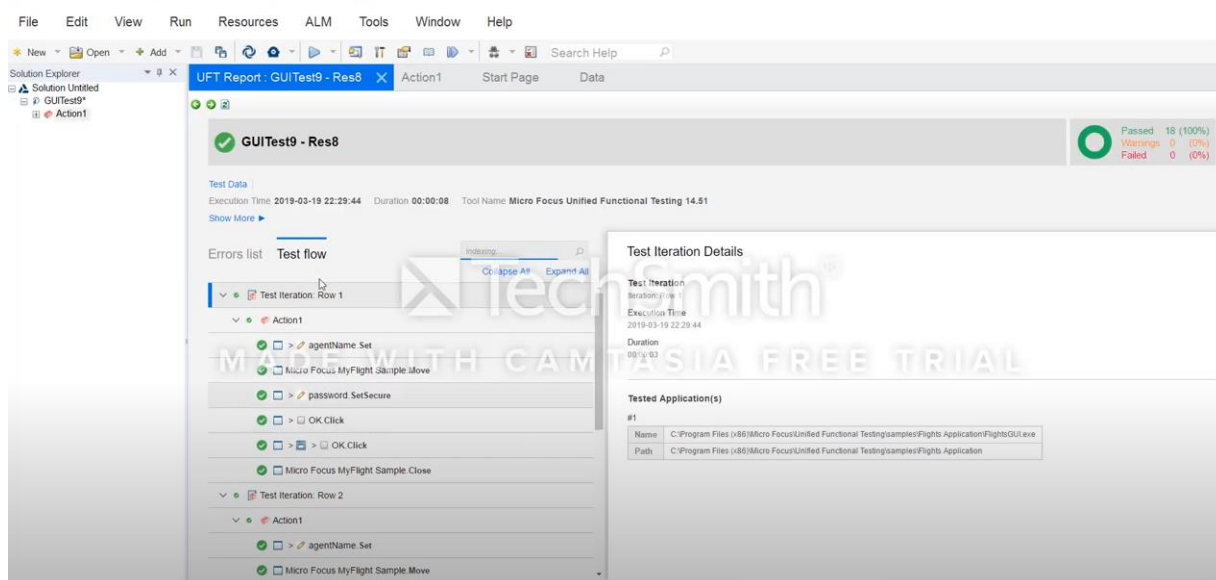
(Hình 3.3.5)

- Sau khi thực hiện thao tác xong thì sẽ thu được 1 tập các câu lệnh tương ứng với các thao tác chúng ta vừa thực hiện:

```
WpfWindow("Micro Focus MyFlight Sample").WpfEdit("agentName").Set "abcd"
WpfWindow("Micro Focus MyFlight Sample").Move 790,266 I
WpfWindow("Micro Focus MyFlight Sample").WpfEdit("password").SetSecure "5c910833f39adbcdbcc187bfc22"
WpfWindow("Micro Focus MyFlight Sample").WpfButton("OK").Click
WpfWindow("Micro Focus MyFlight Sample").Dialog("Login Failed").WinButton("OK").Click
WpfWindow("Micro Focus MyFlight Sample").WpfEdit("password").SetSecure "5c910845d504"
WpfWindow("Micro Focus MyFlight Sample").Close
```

(Hình 3.3.6)

- Tiến hành chạy kiểm thử, phần mềm sẽ trả về cho kết quả các test case đi kèm với lý do (nếu có lỗi):



(Hình 3.3.7)

Vậy là đã xong các bước thực hiện kiểm thử chức năng đăng nhập của 1 phần mềm bằng phần mềm QTP (UFT).

-----**HẾT**-----