TRƯỜNG ĐẠI HỌC KINH TẾ

**KHOA THỐNG KÊ – TIN HỌC**



**BÁO CÁO THỰC TẬP NGHỀ NGHIỆP**

**NGÀNH HỆ THỐNG THÔNG TIN QUẢN LÝ**

**CHUYÊN NGÀNH TIN HỌC QUẢN LÝ**

**THỰC HIỆN KIỂM THỬ ỨNG DỤNG PHẦN MỀM MUA BÁN BẤT ĐỘNG SẢN CỦA MỸ**

Sinh viên thực hiện : **Huỳnh Thị Anh**

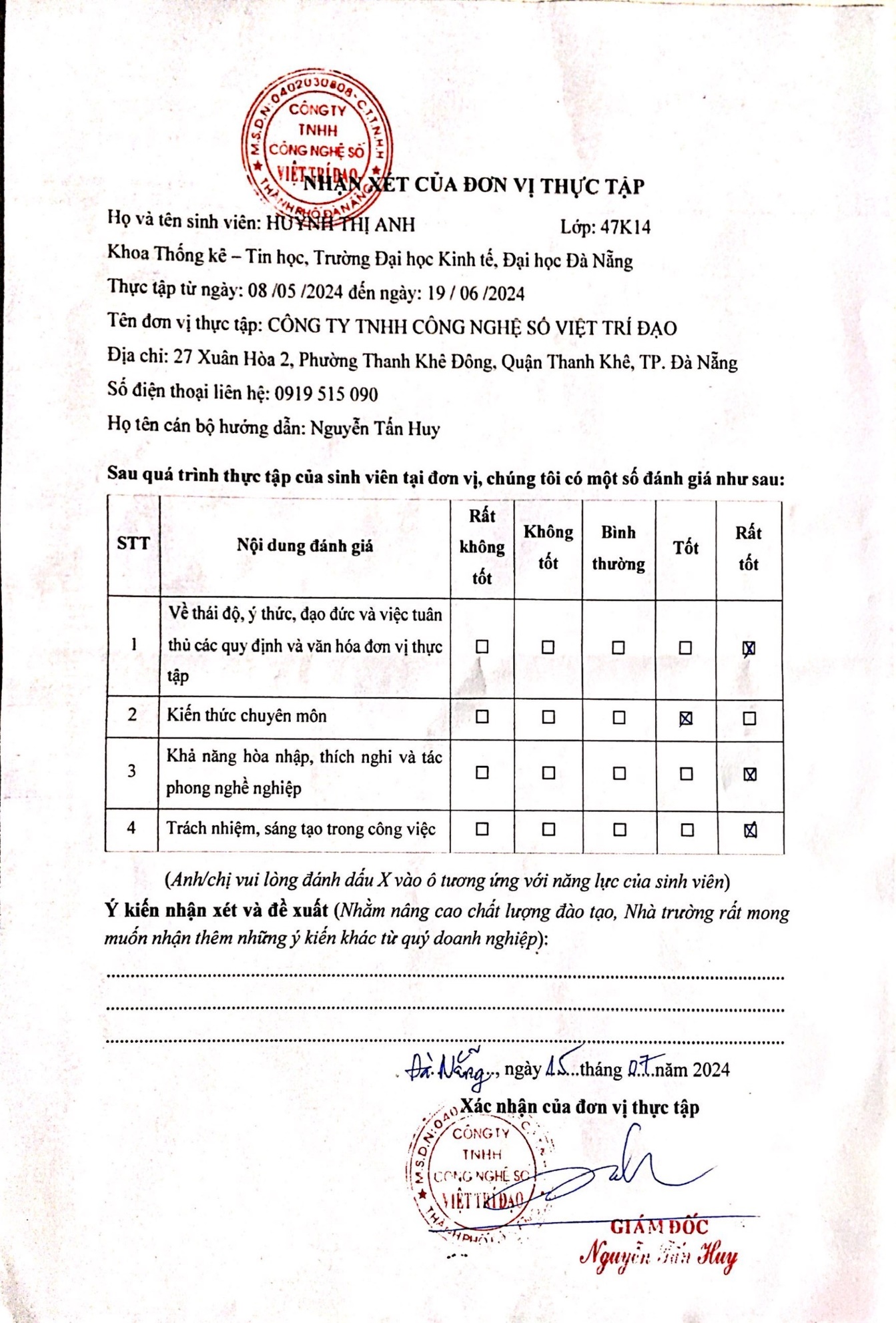
Lớp : **47K14**

Đơn vị thực tập : **Công ty TNHH Công Nghệ Số Việt Trí Đạo**

Cán bộ hướng dẫn : **Nguyễn Tấn Huy**

Giảng viên hướng dẫn : **Ths. Nguyễn Thành Thủy**

***Đà Nẵng, 8/******2024***

**

# **LỜI CẢM ƠN**

Trước tiên với tình cảm sâu sắc và chân thành nhất, cho phép em được bày tỏ lòng biết ơn đến tất cả các cá nhân và tổ chức đã tạo điều kiện hỗ trợ, giúp đỡ em trong suốt quá trình học tập và nghiên cứu đề tài này. Trong suốt thời gian từ khi bắt đầu học tập tại trường đến nay, em đã nhận được rất nhiều sự quan tâm, giúp đỡ của của nhiều tập thể và cá nhân.

Trước hết, em xin gửi tới toàn thể các thầy, cô giáo trong Khoa Thống kê –Tin học, cùng thầy cô trong trường Đại học Kinh tế lời cảm ơn chân thành nhất. Em xin trân trọng gửi đến thầy Nguyễn Thành Thủy - Người đã trực tiếp tận tình hướng dẫn cũng như cung cấp tài liệu cần thiết, chỉ bảo, góp ý để em có thể hoàn thành được báo cáo một cách tốt nhất lời cảm ơn chân thành và sâu sắc nhất.

Em xin gửi lời cảm ơn tới CÔNG TY TNHH CÔNG NGHỆ SỐ VIỆT TRÍ ĐẠO đã có sự hỗ trợ nhiệt tình để em có thể hoàn thành tốt đợt thực tập vừa qua. Với những gì đã tích lũy được, giờ đây em đã xác nhận được hướng đi đúng đắn nhất cho bản thân để có thể vươn tới thành công trong tương lai.

Cuối cùng em kính chúc quý thầy, cô dồi dào sức khỏe và thành công trong sự nghiệp cao quý. Đồng kính chúc các anh, chị trong Công Ty TNHH Công Nghệ Số Việt Trí Đạo luôn dồi dào sức khỏe, đạt được nhiều thành công tốt đẹp trong công việc.

Em xin chân thành cảm ơn!

# **LỜI CAM ĐOAN**

Đề tài "Thực hiện kiểm thử ứng dụng phần mềm mua bán bất động sản của Mỹ" đang được thực hiện trong khuôn khổ dự án của Công ty TNHH Công Nghệ Số Việt Trí Đạo, với sự phê duyệt và hướng dẫn từ Giám đốc Công ty, anh Nguyễn Tấn Huy. Em đã được phép và đóng góp công sức của mình vào việc nghiên cứu và thực hiện đề tài này.

Các kết quả và kết luận trong đề tài là kết quả của quá trình tự tìm hiểu, phân tích và thực hiện một cách trung thực và phù hợp với thực tế. Em cam đoan rằng tất cả các thông tin và kết quả được trình bày không sao chép từ bất kỳ tài liệu nào khác và chưa từng được công bố trong bất kỳ nghiên cứu nào trước đây. Em đã tuân thủ nguyên tắc đạo đức nghiên cứu và bảo mật thông tin trong quá trình thực hiện đề tài này.

Em xin khẳng định rằng đề tài "Thực hiện kiểm thử ứng dụng phần mềm mua bán bất động sản của Mỹ" là một công trình nghiên cứu và cống hiến của em, với mục đích cung cấp thông tin chính xác và hữu ích về việc kiểm thử ứng dụng phần mềm trong lĩnh vực mua bán bất động sản.

**MỤC LỤC**

[LỜI CẢM ƠN ii](#_Toc173599833)

[LỜI CAM ĐOAN iii](#_Toc173599834)

[DANH MỤC HÌNH ẢNH vii](#_Toc173599835)

[LỜI MỞ ĐẦU 1](#_Toc173599836)

[CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI 2](#_Toc173599837)

[**1.1 Giới thiệu tổng quát về Công ty TNHH Công Nghệ Số Việt Trí Đạo** 2](#_Toc173599838)

[1.1.1 Giới thiệu về Công ty TNHH Công Nghệ Số Việt Trí Đạo 2](#_Toc173599839)

[**1.2 Tổng quan về vị trí việc làm** 2](#_Toc173599840)

[1.2.1 Khái niệm về kiểm thử phần mềm – Tester? 3](#_Toc173599841)

[1.2.1 Mô tả về công việc Tester 3](#_Toc173599842)

[1.2.3 Nhiệm vụ của một Tester. 3](#_Toc173599843)

[1.2.4 Cơ hội nghề nghiệp 4](#_Toc173599844)

[**1.3 Cơ sở lý thuyết về kiểm thử phần mềm** 5](#_Toc173599845)

[1.3.1 Các nguyên tắc trong kiểm thử phần mềm 5](#_Toc173599846)

[1.3.3. Các loại kiểm thử phần mềm 9](#_Toc173599847)

[1.3.4 Phương pháp kiểm thử phần mềm 10](#_Toc173599848)

[1.3.5 Các cấp độ kiểm thử phần mềm 10](#_Toc173599849)

[**1.4 Phương pháp quản lý dự án.** 12](#_Toc173599850)

[1.4.1 Quy trình Agile – Mô hình Scrum 12](#_Toc173599851)

[1.4.2 Đặc điểm của quy trình Agile 13](#_Toc173599852)

[1.4.3 Các đối tượng, phương tiện của Scrum 13](#_Toc173599853)

[1.4.4 Các mức độ của lỗi khi fix bug 15](#_Toc173599854)

[CHƯƠNG 2: PHÂN TÍCH HỆ THỐNG PHẦN MỀM 17](#_Toc173599855)

[**2.1 Giới thiệu tổng quát về phần mềm mua bán bất động sản của Mỹ** 17](#_Toc173599856)

[2.1.1 Giới thiệu về phần mềm 17](#_Toc173599857)

[2.1.2 Sơ đồ use case tổng quát 18](#_Toc173599858)

[2.1.3 Workflow của hệ thống 18](#_Toc173599859)

[2.1.4 Phân tích các giao diện – chức năng 19](#_Toc173599860)

[CHƯƠNG 3: TRIỂN KHAI THỰC NGHIỆM 24](#_Toc173599861)

[**3.1 Thực hiện kiểm thử Manual Testing trên phần mềm** 24](#_Toc173599862)

[3.1.1 Testcase màn hình HomPage 24](#_Toc173599863)

[3.1.2 Testcase màn hình Login 25](#_Toc173599864)

[3.1.3 Testcase màn hình “Tạo thông tin BĐS bán” 26](#_Toc173599865)

[3.1.4 Kết quả của kiểm thử manual testing 27](#_Toc173599866)

[**3.2 Nghiên cứu Automation Testing trên phần mềm bằng Selenium Framework** 28](#_Toc173599867)

[3.2.1 Giới thiệu công cụ kiểm thử tự động Selenium 28](#_Toc173599868)

[3.2.2 Đặc điểm của Selenium 29](#_Toc173599869)

[3.2.3 Môi trường Selenium 29](#_Toc173599870)

[3.2.4 Cài đặt selenium 29](#_Toc173599871)

[3.2.5 Website thực hiện kiểm thử Automation 32](#_Toc173599872)

[3.2.6 Testcase cho Automation 34](#_Toc173599873)

[3.2.7 Kết quả run code Test Case Automation 37](#_Toc173599874)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 38](#_Toc173599875)

[**1. Kết quả đạt được** 38](#_Toc173599876)

[**2. Hạn chế và hướng phát triển** 38](#_Toc173599877)

[CHECK LIST CỦA BÁO CÁO 39](#_Toc173599878)

[PHỤ LỤC 40](#_Toc173599879)

[TÀI LIỆU THAM KHẢO 41](#_Toc173599880)

# **DANH MỤC HÌNH ẢNH**

[Hình 1. 1 Logo của công ty TNHH Công Nghệ Số Việt Trí Đạo 2](#_Toc173599952)

[Hình 2.1 Quy trình kiểm thử phần mềm 7](#_Toc173599953)

[Hình 3.1 Cấp độ kiểm thử phần mềm 10](#_Toc173599954)

[Hình 4.1 Mô hình Agile/ Scrum 12](#_Toc173599955)

[Hình 5.1 DailyScrum 14](#_Toc173599956)

[Hình 6.1 . Ví dụ quản lý tiến độ cá nhân trong từng sprint 14](#_Toc173599957)

[Hình 7.1 Giao diện trang chủ của phần mềm 17](#_Toc173599958)

[Hình 8.1 Use case tổng quát 18](#_Toc173599959)

[Hình 9.1 Workflow của hệ thống 18](#_Toc173599960)

[Hình 10.1 Giao diện đăng nhập 19](#_Toc173599961)

[Hình 11.1 Click vào button " Sign In" 20](#_Toc173599962)

[Hình 12.1 User story về giao diện đăng nhập 21](#_Toc173599963)

[Hình 13.1 Giao diện màn hình “Tạo thông tin BĐS bán” 22](#_Toc173599964)

[Hình 14.1 Màn hình kết quả testcase 28](#_Toc173599965)

[Hình 15.1 Màn hình download Chrome Driver 30](#_Toc173599966)

[Hình 16.1Tải Version 115.x.xxxx.xx của Chrome. 30](#_Toc173599967)

[Hình 17.1 Tập tin chromedriver.exe 30](#_Toc173599968)

[Hình 18.1 Intellij IDEA 31](#_Toc173599969)

[Hình 19.1 Tạo project mới 31](#_Toc173599970)

[Hình 20.1 Web thực hiện automation 33](#_Toc173599971)

[Hình 21.1 Giao diện màn hình Login 33](#_Toc173599972)

[Hình 22.1 Giao diện màn hình Register 34](#_Toc173599973)

[Hình 23.1 Giao diện Lost your Password 34](#_Toc173599974)

[Hình 24.1 Testcase automation 35](#_Toc173599975)

[Hình 25.1 Màn hình code testcase 37](#_Toc173599976)

[Hình 26.1 Mành hình kết quả code Automation 37](#_Toc173599977)

# **LỜI MỞ ĐẦU**

1. **Mục tiêu của đề tài**

Mục tiêu chính của đề tài là đánh giá toàn diện chất lượng và độ tin cậy của ứng dụng phần mềm giao dịch bất động sản Mỹ.

Cụ thể, nghiên cứu tập trung vào việc xác định các lỗ hổng bảo mật, lỗi chức năng, thiếu sót trong yêu cầu và các bất hợp lý so với quy trình thực tế của giao dịch bất động sản.

1. **Mục tiêu nghiên cứu**

Nghiên cứu về kiểm thử phần mềm bao gồm việc tìm hiểu các lý thuyết và phương pháp liên quan, cũng như hiểu về các mô hình phát triển phần mềm như Agile/Scrum. Đồng thời, nghiên cứu này cũng bao gồm việc học cách ghi nhật ký lỗi và xây dựng testcase một cách hệ thống trong quá trình kiểm thử.

Mục tiêu của việc áp dụng kiến thức như vậy là để tìm ra các lỗi trong hệ thống và đảm bảo chất lượng sản phẩm, giảm thiểu các lỗi nghiêm trọng trước khi đưa sản phẩm đến tay khách hàng.

1. **Đối tượng và phạm vi nghiên cứu**

* Đối tượng nghiên cứu: Trang web mua bán bât động sản của Mỹ
* Phạm vi nghiên cứu: Thực hiện kiểm thử thủ công và tự động với một số chức năng nhất định trên trang web.

1. **Kết cấu của đề tài**

Đề tài được tổ chức gồm phần mở đầu, 3 chương nội dung và phần kết luận.

- Mở đầu

- Chương 1: Tổng quan về đề tài

- Chương 2: Phân tích hệ thống phần mềm

- Chương 3: Triển khai thực nghiệm

- Kết luận và hướng phát triển

# **CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI**

## **1.1 Giới thiệu tổng quát về Công ty TNHH Công Nghệ Số Việt Trí Đạo**

### **1.1.1 Giới thiệu về Công ty TNHH Công Nghệ Số Việt Trí Đạo**

VIỆT TRÍ ĐẠO đã được thành lập chính thức từ ngày 05/05/2019



Hình 1. 1 Logo của công ty TNHH Công Nghệ Số Việt Trí Đạo

Việt Trí Đạo là một công ty tư vấn chiến lược và công nghệ, và tất cả các cán bộ, nhân viên luôn tuân thủ một triết lý làm việc: Đặt khách hàng lên hàng đầu, như chúng ta đang phục vụ bản thân mình! Chúng tôi cam kết hỗ trợ khách hàng của mình trong việc tận dụng những công nghệ phần mềm và phần cứng mới nhất để tái thiết kế hoạt động kinh doanh, tạo ra những sản phẩm mới đột phá hoặc thực hiện sự chuyển đổi tổ chức để tìm kiếm cơ hội tăng trưởng và tối ưu hóa chi phí. Chúng tôi luôn nỗ lực để mang lại giá trị cao nhất cho khách hàng và giúp họ thành công.

## **1.2 Tổng quan về vị trí việc làm**

Kiểm thử phần mềm là quá trình giám định dựa trên chất lượng sản phẩm hoặc dịch vụ sau khi tiến hành kiểm tra. Kiểm thử độc lập cho phép doanh nghiệp đánh giá mức độ rủi ro khi phát triển phần mềm.

Kỹ thuật kiểm thử bao gồm quá trình kiểm thử phần mềm và tìm lỗi. Kiểm thử phần mềm bao gồm quá trình kiểm tra và kiểm định để đảm bảo phần mềm đáp ứng nhu cầu người dùng, phần mềm chạy đúng chức năng.

### **1.2.1 Khái niệm về kiểm thử phần mềm – Tester?**

Kiểm thử phần mềm (Software Testing) là một quá trình thực thi một chương trình hoặc ứng dụng với mục đích tìm ra lỗi của phần mềm. Và được sử dụng để xác định tính đúng đắn, đầy đủ và chất lượng của phần mềm máy tính được phát triển.

### **1.2.1 Mô tả về công việc Tester**

Kiểm thử viên (tester) phải làm những công việc sau đây trong dự án:

1. Đọc hiểu tài liệu mô tả nghiệp vụ của phần mềm (thường được gọi là tài liệu SRS- Software requirements specification)

2. Đọc hiểu bản thiết kế cơ sở dữ liệu (tiếng Anh gọi là database) của hệ thống

3. Viết tài liệu testcase (tức là các kịch bản kiểm thử)

4. Mô tả lỗi (tiếng Anh gọi là log bug)

5. Sau khi lập trình viên sửa lỗi xong thì test lại xem có còn lỗi hay không

6. Sau khi lập trình viên sửa xong hết lỗi thì tester sẽ làm test report (bản báo cáo về quá trình kiểm thử) để gửi cho người quản trị dự án hoặc khách hàng xem.

### **1.2.3 Nhiệm vụ của một Tester.**

**Nghiên cứu, phân tích yêu cầu**

Trong quá trình xây dựng và phát triển dự án phần mềm, tester đóng vai trò quan trọng trong việc nghiên cứu và phân tích yêu cầu. Tester sẽ cùng lập trình viên làm việc để phân tích và xác định những yêu cầu kỹ thuật liên quan trong dự án. Qua việc hợp tác chặt chẽ này, chúng ta có thể đảm bảo rằng các yêu cầu kỹ thuật được hiểu rõ và áp dụng một cách chính xác trong quá trình xây dựng phần mềm. Điều này đóng góp vào việc tạo ra sản phẩm chất lượng cao và đáng tin cậy.

**Đánh giá, phát hiện các vấn đề của phần mềm**

Nhiệm vụ quan trọng nhất của tester là thực hiện kiểm thử để phát hiện lỗi và các vấn đề kỹ thuật có thể ảnh hưởng đến chất lượng phần mềm, sau đó đề xuất các giải pháp khắc phục. Việc kiểm thử được tiến hành dựa trên danh sách kiểm tra được yêu cầu hoặc các kịch bản đã được chuẩn bị sẵn. Qua quá trình này, tester giúp đảm bảo rằng phần mềm đáp ứng các yêu cầu và tiêu chuẩn chất lượng, đồng thời đảm bảo tính ổn định và đáng tin cậy của hệ thống.

**Ngăn ngừa lỗi phát sinh của phần mềm**

Ngoài việc phát hiện và đánh giá vấn đề trong phần mềm, tester cũng phải thực hiện các nhiệm vụ chuyên môn nhằm ngăn chặn sự phát sinh lỗi. Điều này đòi hỏi tester có khả năng quan sát, đánh giá và kiểm tra mọi quy trình trong quá trình phát triển phần mềm. Hơn nữa, tester cần tương tác trực tiếp với khách hàng để hiểu rõ yêu cầu và mong muốn của họ, từ đó xây dựng danh mục kiểm tra khi thử nghiệm phần mềm. Tester cũng phải chuẩn bị các báo cáo liên quan đến kiểm thử phần mềm và hỗ trợ cho các nhà phát triển trong quá trình phát triển sản phẩm. Điều này giúp đảm bảo rằng quá trình kiểm thử được thực hiện một cách hiệu quả và hỗ trợ cho việc cải thiện chất lượng phần mềm.

### **1.2.4 Cơ hội nghề nghiệp**

Nghề tester hiện nay không những rất triển vọng tại Việt Nam mà còn rất “hot” tại các quốc gia có ngành Công Nghệ Thông Tin phát triển. Ngoài ra, Tester có thể phát triển sự nghiệp của mình trong nhiều lĩnh vực liên quan đến công nghệ thông tin. Dưới đây là một số nghề nghiệp liên quan đến cơ hội nghề nghiệp của tester:

+ ***Test Automation Engineer***: Tester có thể trở thành một Test Automation Engineer, người có trách nhiệm tạo ra các kịch bản tự động hóa để kiểm thử phần mềm.

+ ***Quality Assurance Manager***: Tester có thể trở thành một Quality Assurance Manager, người quản lý hoạt động kiểm thử phần mềm và đảm bảo chất lượng sản phẩm. Quality Assurance Manager cần có các kỹ năng quản lý dự án và kinh nghiệm trong việc đào tạo và lãnh đạo các nhân viên kiểm thử phần mềm

+ ***Business Analyst***: Tester có thể trở thành một Business Analyst, người phân tích yêu cầu khách hàng và đưa ra giải pháp kỹ thuật để giải quyết các vấn đề kinh doanh. Business Analyst cần có các kỹ năng phân tích và giao tiếp tốt.

+ ***Technical Support Engineer***: Tester có thể trở thành một Technical Support Engineer, người hỗ trợ khách hàng trong việc sử dụng sản phẩm phần mềm. Technical Support Engineer cần có kiến thức về phần mềm và kỹ năng giao tiếp tốt để giải quyết các vấn đề của khách hàng.

+ ***Software Developer***: Tester có thể học thêm về lập trình và trở thành một Software Developer, người phát triển phần mềm. Tester có kiến thức về kiểm thử phần mềm và có thể áp dụng kiến thức này trong việc phát triển phần mềm.

## **1.3 Cơ sở lý thuyết về kiểm thử phần mềm**

### **1.3.1 Các nguyên tắc trong kiểm thử phần mềm**

***Kiểm thử càng sớm càng tiết kiệm thời gian và chi phí***

Nguyên tắc này yêu cầu bắt đầu thử nghiệm phần mềm trong giai đoạn đầu của vòng đời phát triển phần mềm. Các hoạt động kiểm thử phần mềm từ giai đoạn đầu sẽ giúp phát hiện bug sớm hơn. Nó cho phép chuyển giao phần mềm theo yêu cầu đúng thời gian với chất lượng dự kiến. Kiểm thử nên được thực hiện càng sớm, từ giai đoạn đầu của dự án thì cũng đỡ mất thời gian và chi phí, nếu mà lỗi phát hiện muộn thì càng mất thời gian và sẽ tốn nhiều chi phí hơn.

***Kiểm thử toàn bộ là không thể***

Kiểm thử mọi thứ là không khả thi trừ trường hợp nhỏ. Thay vì kiểm thử toàn bộ, chúng ta có thể dựa vào phân tích rủi ro (risk analysis), độ ưu tiên (priorities), sử dụng kỹ thuật test (test techniques) để lựa chọn các ca kiểm thử phù hợp nhất

***Lỗi thường được phân bố tập trung***

Thông thường, phần lớn lỗi tập trung vào những module, thành phần chức năng chính của hệ thống. Điều này cũng thuận theo nguyên lý Pareto: 80% số lượng lỗi được tìm thấy trong 20% tính năng của hệ thống. Nếu bạn thành công xác định được điều này, bạn sẽ tập trung vào tìm kiếm lỗi quanh khu vực được xác định. Nó được coi là một trong những cách hiệu quả nhất để thực hiện kiểm tra hiệu quả

***Nghịch lý thuốc trừ sâu***

Trong kiểm thử phần mềm, nếu bạn cứ thực thi lặp đi lặp lại một bộ test case thì có khả năng rất thấp bạn sẽ tìm được lỗi từ những trường hợp kiểm thử này. Nguyên nhân là do khi hệ thống ngày càng hoàn thiện, những lỗi được tìm thấy lúc trước đã được sửa trong khi những trường hợp kiểm thử đã cũ. Do đó, khi một lỗi được sửa hay một tính năng mới được thêm vào, chúng ta nên tiến hành làm regression (kiểm thử hồi qui) nhằm mục đích đảm bảo những thay đổi này không ảnh hưởng đến những vùng khác của sản phẩm.

***Kiểm thử phụ thuộc vào ngữ cảnh***

khi test một app đặt nước thì suy nghĩ hay tiến trình test sẽ khác với khi chúng ta test một app quản lý bệnh nhân hay quản lý môn học. Mỗi app sẽ có một cái cách suy nghĩ, một tiến trình test khác nhau. Ví dụ, chúng ta không thể áp dụng bộ kiểm thử trên môi trường Mobile app để thực hiện kiểm thử lên một Web trên PC được

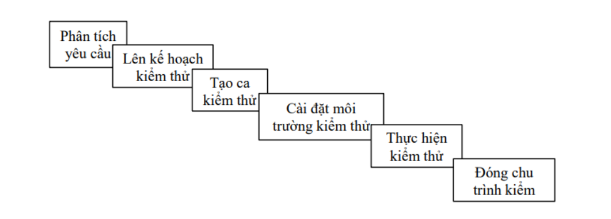
***Quan niệm sai lầm về việc “hết lỗi”***

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

***Kiểm thử chứng mình sự hiện diện của lỗi***

Mục tiêu chính của kiểm thử là xác định các lỗi trong phần mềm. Kiểm thử giúp phát hiện và loại bỏ các lỗi để đảm bảo phần mềm ổn định, đáng tin cậy và an toàn. (Test có thể cho thấy phần mềm không có lỗi nhưng không thể chứng minh rằng phần mềm không có lỗi.

**1.3.2 Quy trình kiểm thử phần mềm**



Hình 2.1 Quy trình kiểm thử phần mềm

***Phân tích yêu cầu***

Phân tích yêu cầu là giai đoạn đầu tiên trong quy trình kiểm thử phần mềm. QA team sẽ thực hiện đọc hiểu, nghiên cứu và phân tích cụ thể các yêu cầu trong tài liệu đặc tả của dự án hoặc tài liệu khách hàng. Qua hoạt động này, QA team sẽ được các yêu cầu mà dự án đưa ra bao gồm yêu cầu kiểm thử chức năng/ phi chức năng nào.

***Lập kế hoạch kiểm thử***

Test plan là tài liệu tổng qua về kiểm thử của một dự án. Thường được sử dụng đặt tả phạm vu dự án, hướng tiếp cận, tài nguyên và nhân lực cần có, các tính năng cần được test và không cần phải test, các công cụ và môi trường test cần có. Trong giai đoạn này, nhà quản lý kiểm thử sẽ tiến hành lập kế hoạch kiểm thử.

***Tạo ca kiểm thử***

Nội dung của kịch bản kiểm thử bao gồm các hoạt động sau:

- Review tài liệu: Đầu tiên, các kiểm thử viên cần review lại tất cả các tài liệu để xác định công việc cần làm, các công việc có khác gì so với dự án trước khách hàng đưa cho, chức năng nào cần test, chức năng nào không cần test lại nữa. Từ đó, vừa có thể tiết kiệm thời gian mà vẫn đưa ra được một kịch bản kiểm thử đầy đủ và hiệu quả.

- Viết test case/ check list: Sau đó, tester bắt tay vào việc viết test case chi tiết dựa vào kế hoạch đã đưa ra và vận dụng các kỹ thuật thiết kế kịch bản kiểm thử. Test case cần bao phủ được tất cả các trường hợp kiểm thử có thể xảy ra cũng như đáp ứng đầy đủ các tiêu chí của sản phẩm. Đồng thời tester cũng cần đánh giá mức độ ưu tiên cho từng test case.

- Chuẩn bị dữ liệu kiểm thử: Cùng với việc tạo ra các test case chi tiết, đội kiểm thử cũng cần chuẩn bị trước các dữ liệu kiểm thử cho các trường hợp cần thiết như test data, test script.

- Review test case/ check list: Sau khi hoàn thành, các thành viên trong đội kiểm thử hoặc test leader cũng cần review lại test case đã tạo để có thể bổ sung, hỗ trợ lẫn nhau nhằm tránh những sai sót trong thiết kế test case và rủi ro về sau.

***Cài đặt môi trường kiểm thử***

Việc cài đặt môi trường kiểm thử là giai đoạn cũng rất quan trọng trong vòng đời phát triển phần mềm. Môi trường kiểm thử sẽ được quyết định dựa trên những yêu cầu của khách hàng, hay đặc thù của sản phẩm ví dụ như server/ client/ network ,...

Tester cần chuẩn bị một vài test case để kiểm tra xem môi trường cài đặt đã sẵn sàng cho việc kiểm thử hay chưa. Đây chính là việc thực thi các smoke test case.

***Thực hiện kiểm thử***

˗ Ghi lại ID và phiên bản của (các) hoặc đối tượng kiểm thử, công cụ kiểm tra và phần mềm kiểm thử

˗ Thực hiện các bài kiểm tra theo cách thủ công hoặc bằng cách sử dụng các công cụ thực thi kiểm thử

˗ So sánh kết quả thực tế với kết quả dự kiến

˗ Phân tích các trường hợp fail và có nguy cơ fail để xác định nguyên nhân của nó

˗ Báo cáo lỗi dựa trên những trường hợp fail

˗ Ghi lại kết quả thực hiện kiểm thử

˗ Ghi lại các kết quả vào quy trình để tạo ra các kết quả đó

˗ Xác minh và cập nhật khả năng xác định nguồn gốc hai chiều giữa cơ sở thử nghiệm, điều kiện thử nghiệm, trường hợp thử nghiệm, quy trình thử nghiệm và bộ thử nghiệm

***Đóng chu trình kiểm thử***

Đây là giai đoạn cuối cùng trong quy trình kiểm thử phần mềm. Ở giai đoạn này, QA team thực hiện tổng kết, báo cáo kết quả về việc thực thi test case, bao nhiêu case pass/ fail, bao nhiêu case đã được fix, mức độ nghiêm trọng của lỗi, bao nhiêu lỗi cao/ thấp, lỗi còn nhiều ở chức năng nào, dev nào nhiều lỗi. Chức năng nào đã hoàn thành test/ chưa hoàn thành test/ trễ tiến độ bàn giao.

Đánh giá các tiêu chí hoàn thành như phạm vi kiểm tra, chất lượng, chi phí, thời gian, mục tiêu kinh doanh quan trọng.

### **1.3.3. Các loại kiểm thử phần mềm**

**Manual testing**

Kiểm thử thủ công là loại kiểm thử phần mềm không sử dụng bất kỳ công cụ tự động nào. Trong loại này, người kiểm thử đóng vai trò là người dùng cuối và kiểm tra phần mềm để xác định lỗi không mong muốn. Các loại kiểm thử thủ công như unit testing, integration testing, system testing, user acceptance testing.

**Automation testing**

Automation Test là một quá trình xử lý tự động các bước thực hiện một test case và được thực hiện bởi phần mềm là Automation Testing Tool. Automation Test có thể hiểu rất đơn giản là thay vì test bằng tay, ta để máy thực hiện việc testing mà Tester phải làm (Khởi động hệ thống, nhập dữ liệu đầu vào, kiểm tra so sánh với dữ liệu đầu ra và ghi kết quả). Automation Testing đóng một vai trò quan trọng góp phần nâng cao năng suất kiểm thử, giảm thiểu lỗi cũng như giảm việc kiểm thử bằng tay lặp đi lặp lại.

### **1.3.4 Phương pháp kiểm thử phần mềm**

**Black Box Testing**

Kiểm thử hộp đen là phương pháp kiểm tra dựa vào đầu vào và đầu ra của chương trình để test, không quan tâm đến code bên trong được viết như thế nào. Tester sẽ xem phần mềm như một cái hộp đen.

Input: thông tin về các yêu cầu chức năng và giao diện của phần mềm. Tester sẽ không có thông tin về cấu trúc nội bộ và hoạt động bên trong của phần mềm. Tester sử dụng các bộ test case dựa trên đặc tả chức năng và yêu cầu của phần mềm để kiểm tra xem phần mềm có thực hiện chính xác các chức năng yêu cầu hay không.

Output: Xác định các lỗi: lỗi chức năng, lỗi giao diện, lỗi hoạt động của phần mềm.

**White Box Testing**

Kiểm thử hộp trắng sẽ xem xét cấu trúc code của phần mềm. Các test case sẽ được viết dựa vào cấu trúc code này. Người kiểm thử truy cập vào code và có thể kiểm tra nó. Input: Mã nguồn, cấu trúc của phần mềm. Tester sẽ xem xét mã nguồn và tìm hiểu hoạt động phần mềm bên trong. Dựa vào cấu trúc của phần mềm để thực hiện viết các bộ test case.

Output: Xác định lỗi, kiểm tra tính đúng đắn của mã nguồn, cấu trúc code của phần mềm.

### **1.3.5 Các cấp độ kiểm thử phần mềm**



Hình 3.1 Cấp độ kiểm thử phần mềm

**Unit Testing (Kiểm thử đơn vị)**

***Khái niệm:*** Kiểm thử đơn vị là cấp độ kiểm thử cơ bản, thực hiện test từng module nhỏ trong hệ thống trong đó các thành phần đơn lẻ của phần mềm được kiểm tra như: Hàm (Function), Lớp (Class), Phương thức (Method). Kiểm thử đơn vị được thực hiện trong quá trình phát triển ứng dụng. Lỗi xảy ra ở giai đoạn này này thường được sửa ngay sau khi chúng được tìm ra mà không cần lưu lại và quản lý như các cấp độ kiểm thử khác.

***Người thực hiện:*** Thường là do lập trình viên thực hiện

***Mục đích:*** Kiểm thử đơn vị dùng để xác nhận mỗi thành phần của phần mềm thực hiện đúng với thiết kế ban đầu.

**Integration Testing (Kiểm thử tích hợp)**

***Khái niệm:*** Kiểm thử tích hợp hay còn gọi là kiểm thử kết hợp. Thông thường, khi phát triển một dự án phần mềm, các chức năng (hay các module riêng lẻ) được code bởi nhiều lập trình viên khác nhau. Kiểm thử tích hợp nghĩa là cùng một lúc chúng ta sẽ test cho nhiều chức năng (hay module) cùng lúc. Nói một cách, kiểm thử tích hợp sẽ kiểm tra việc truyền dữ liệu giữa các chức năng, các màn hình với nhau. Xem thử các chức năng có liên lạc với nhau đúng như yêu cầu hay không.

***Người thực hiện:*** Integration testing được thực hiện bởi một tester hoặc một nhóm các tester trong dự án. Lập trình viên không tham gia vào giai đoạn kiểm thử này

***Mục đích:*** để đảm bảo rằng hệ thống tích hợp đã sẵn sàng để thử nghiệm hệ thống. Kiểm thử tích hợp được thực hiện sau khi kiểm tra đơn vị và trước khi kiểm tra hệ thống

**System Testing (Kiểm thử hệ thống)**

***Khái niệm:*** System Testing là thực hiện kiểm thử nhằm đưa hệ thống vào vận hành thử nghiệm tại các môi trường khác nhau nhằm tìm ra lỗi về hệ thống tương thích.

***Người thực hiện:*** Kiểm thử hệ thống được thực hiện bởi tester.

***Mục đích:*** Kiểm thử hệ thống thường là thử nghiệm cuối cùng để xác minh rằng hệ thống được phân phối đáp ứng theo đúng yêu cầu nghiệp vụ, yêu cầu về chức năng đưa ra hay không.

**Acceptance Testing (Kiểm thử chấp nhận)**

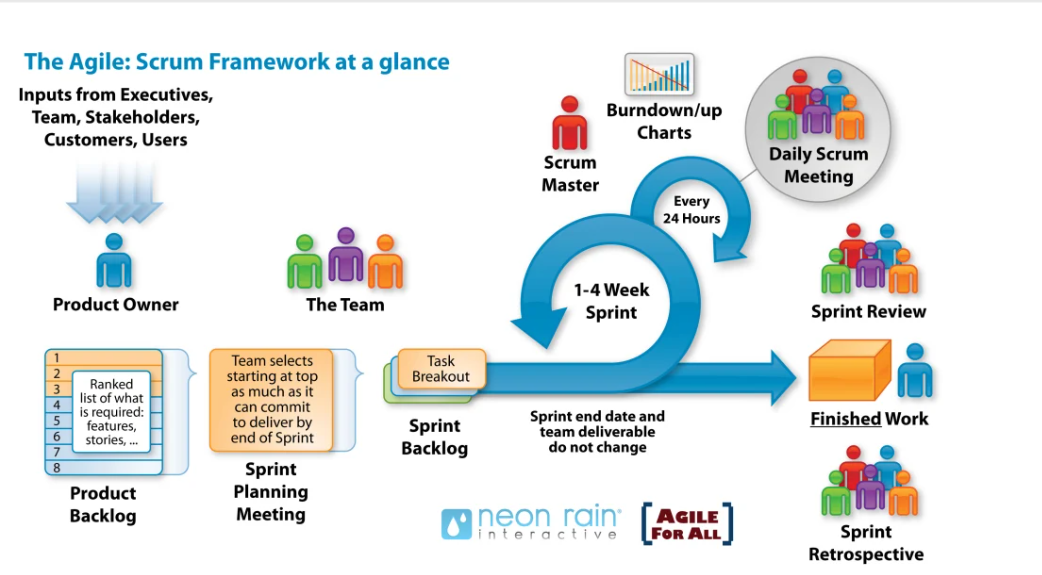
***Khái niệm:*** Kiểm thử chấp nhận là kiểm tra xem phần mềm đã thỏa mãn tất cả yêu cầu của khách hàng chưa và khách hàng có chấp nhận sản phẩm hay không.

***Người thực hiện:*** Khách hàng hoặc bên thứ 3

***Mục đích:*** đảm bảo phần mềm đáp ứng đúng yêu cầu của khách hàng. Sản phẩm nhận được sự chấp nhận từ khách hàng/ người dùng cuối.

## **1.4 Phương pháp quản lý dự án.**

### **1.4.1 Quy trình Agile – Mô hình Scrum**



Hình 4.1 Mô hình Agile/ Scrum

***Khái niệm***

Quy trình Agile với mục tiêu là phần mềm phải có khả năng biến đổi, phát triển và tiến hóa theo thời gian mà không phải phát triển lại từ đầu. Mục tiêu là làm ra những sản phẩm đơn giản đáp ứng đúng và đủ yêu cầu của khách hàng ở thời điểm hiện tại, sẵn sàng cho sự thay đổi trong tương lai.

Mô hình Scrum là 1 dạng của mô hình Agile và là framework phổ biến nhất khi thực hiện mô hình Agile. Scrum là mô hình phát triển phần mềm lặp đi lặp lại. Những khoảng lặp cố định, thường kéo dài 1, 2 tuần được gọi là sprint. Cuối mỗi sprint, các stakeholder và team member sẽ cùng ngồi lại với nhau và lên kế hoạch cho giai đoạn tiếp theo.

### **1.4.2 Đặc điểm của quy trình Agile**

- Đội dự án đông đúc ngay từ ngày đầu. Tức là ngay từ đầu thường dự án đã có luôn dev, tester, PM…

- Không chia dự án thành các giai đoạn như Waterfall. Mà chia dự án thành các sprint. 1 Sprint là 1 block về thời gian. Thường 1 sprint kéo dài từ 1 đến 4 tuần. Trong mỗi sprint thì làm đầy đủ các công việc như: phân tích yêu cầu, coding, testing, deploy…

- Đội ngũ tham gia phần mềm chia thành 3 vai trò với trách nhiệm rõ ràng như sau:

• Product Owner (GĐ dự án): Là có trách nhiệm vừa phải biết phân tích nghiệp vụ, vừa phải kiểm soát được tiến độ dự án và quan trọng nhất là phải biết chọn task nào để giao cho các member làm tưng ứng với từng sprint. PO gần như là người có quyền lực nhất trong dự án. Vì PO vừa nắm cả yêu cầu của KH vừa nắm cả quản lý nguồn lực. Thành bại của dự án phụ thuộc rất nhiều vào PO. Cái tên Product Owner nó cũng phản ánh được trách nhiệm quan trọng của PO.

• Scrum Master (Quản lý dự án): họ phải đảm bảo các sprint được hoàn thành đúng mục đích, bảo vệ đội làm việc và loại bỏ các trở ngại.

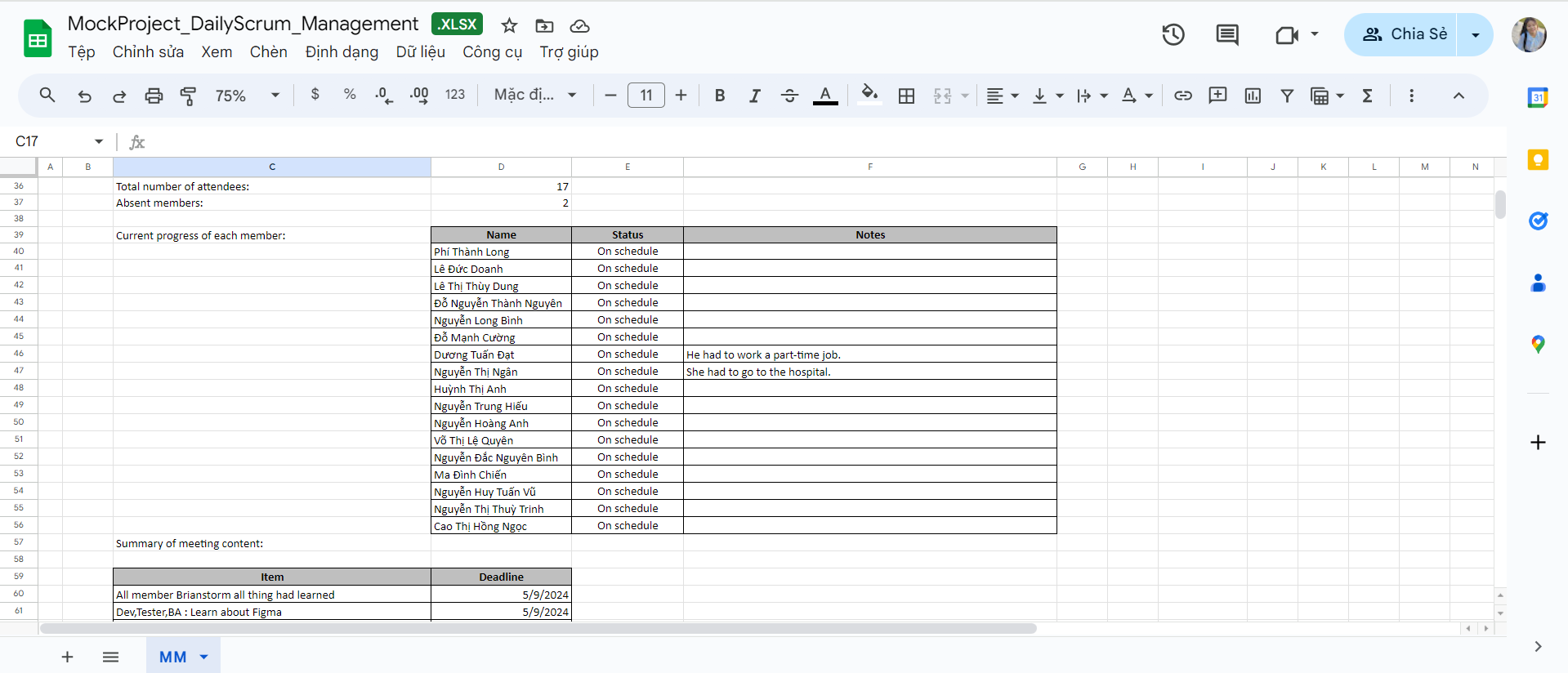
• Development Team (Dev, Test...): thường 5-9 người, tùy theo quy mô dự án nó có thể có rất nhiều, nhiều người tham gia.

### **1.4.3 Các đối tượng, phương tiện của Scrum**

- **Product backlog:** Toàn bộ danh sách các yêu cầu, tính năng của sản phẩm. Danh sách này sẽ được cập nhật thường xuyên bởi PO và bất kỳ ai trong team Độ ưu tiên thứ tự của các tính năng sẽ được PO quyết định.

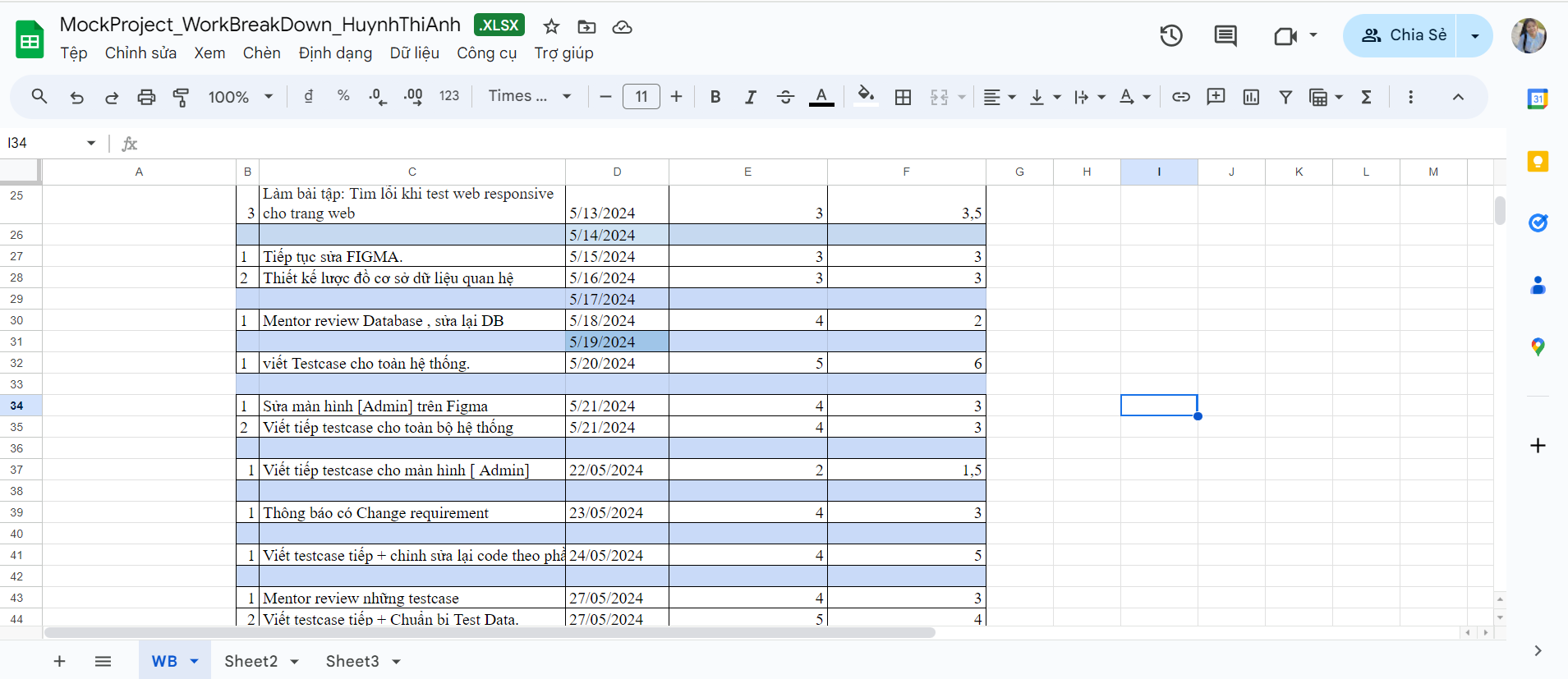
- **Daily Scrum:** Họp hằng ngày khoảng 15 phút để xem tiến độ dự án như thế nào và có gặp vấn đề gì không?

Ví dụ: đây là biên bản họp DailyScrum



Hình 5.1 DailyScrum

**- Sprint:** 1 giai đoạn của dự án với thời gian cố định. Độ dài của 1 sprint sẽ được team và PO quyết định. Thông thường là từ 1 - 4 tuần.



Hình 6.1 . Ví dụ quản lý tiến độ cá nhân trong từng sprint

*-* **Sprint Plan:** Là họp để estimate các user story và chọn các user story sẽ làm cho Sprint tiếp theo. Thường diễn ra vào đầu mỗi Sprint. Nhóm phát triển họp với product owner để lên kế hoạch làm việc cho 1 sprint. Công việc lập kế hoạch bao gồm việc chọnlựa các yêu cầu cần phải phát triển, phân tích và nhận biết các công việc phải làm kèm theo các ước lượng thời gian cần thiết để hoàn tất các nhiệm vụ.

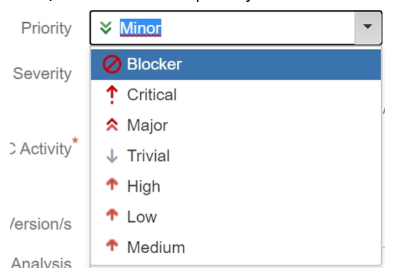
- **Sprint Review:** Sau mỗi Sprint thì tổ chức họp và demo sản phẩm của Sprint vừa rồi cho khách hành xem và nhận phản hồi. Mục đích chính là vậy. Tức là khách hàng liên tục xuất hiện trọng toàn bộ thời gian làm dự án, chứ không phải đến dự án khách hàng mới comment về sản phẩm.

- **Sprint Retrospective:** Sau 1 hoặc vài sprint thì đội dự án tổ chức retro 1 lần xem thử có cần phải cải tiến quy trình dự án hoặc các vấn đề gì khác hay không. Sprint Review thì tập trung vào họp demo sản phẩm, còn Sprint Retrospective thì tập trung vào việc phân tích đánh giá quy trình, cách làm việc trong team.

### **1.4.4 Các mức độ của lỗi khi fix bug**

**Priority:** là mức độ ưu tiên (của việc fix bug). Tức là bug có cần phải sửa gấp hay không thì mình chọn mức độ ưu tiên cho phù hợp. Cái này là bắt buộc phải chọn. Chú ý là tùy vào thời điểm mà mức độ ưu tiên sẽ khác nhau.

Ví dụ danh sách của priority sẽ là:

******

• ***High:*** Sửa ngay lập tức đây là Main function, Flow quan trọng của hệ thống

***• Medium:*** Sửa càng nhanh càng tốt chức năng lỗi function

***• Low:*** có thể khắc phục khi thời gian cho phép.

***• Severity:*** điền mức độ nghiêm trọng của bug. Cái này là bắt buộc phải chọn.

Danh sách của severity thường sẽ là:



***• Cosmetic***: nghĩa là lỗi nhỏ nhặt (Lỗi rất ít nghiêm trọng). Thường là các lỗi về GUI ta sẽ chọn cái này. Nói chung là những lỗi nào chỉ ảnh hưởng đến case đang test và không ảnh hưởng nhiều đến những chức năng khác của phần mềm thì mình chọn cái này.

***• Medium***: là lỗi ở mức độ trung bình (Lỗi hơi nghiêm trọng rồi). Thường là các lỗi liên quan đến chức năng (function) thì mình chọn cái này. Những lỗi medium thường chỉ ảnh hưởng đến 1 màn hình, 1 chức năng, hoặc 1 nhóm chức năng thôi.

***• Serious***: là lỗi ở mức độ nguy hiểm. Thường những lỗi liên quan đến chức năng làm cho cả website hay phần mềm bị treo, tạm thời không thể sử dụng tiếp được; hoặc lỗi sai về nghiệp vụ một cách rất nguy hiểm, có thể gây ảnh hưởng lớn đến nghiệp vụ chung của hệ thống… thì ta mới chọn mức độ này.

***• Fatal***: là ở cực kỳ nguy hiểm (cực kỳ nghiêm trọng). Thường những lỗi như hệ thống bị sập, hoặc buộc phải khởi động lại server… hoặc làm mất dữ liệu của người dùng (ví dụ như xóa sạch dữ liệu trong database hay xóa mất file dữ liệu của user...) thì nên chọn mức độ này.

# **CHƯƠNG 2: PHÂN TÍCH HỆ THỐNG PHẦN MỀM**

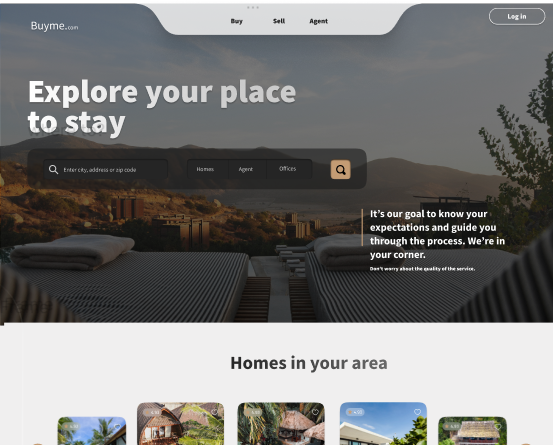
## **2.1 Giới thiệu tổng quát về phần mềm mua bán bất động sản của Mỹ**

### **2.1.1 Giới thiệu về phần mềm**

Phần mềm mua bán bất động sản của Mỹ là một nền tảng trực tuyến cung cấp một trải nghiệm tuyệt vời cho người dùng trong việc tìm kiếm và giao dịch bất động sản. Với hàng ngàn thông tin về nhà đất và căn hộ được tổng hợp, sàng lọc và xử lý một cách kỹ lưỡng, người dùng có thể dễ dàng tìm kiếm và khám phá các cơ hội đầu tư hấp dẫn.

Với giao diện trực quan và dễ sử dụng, phần mềm mua bán bất động sản giúp người dùng tiết kiệm thời gian và công sức trong việc tìm kiếm thông tin.

Không chỉ là một công cụ tìm kiếm, phần mềm còn cung cấp các công cụ quản lý thông tin mạnh mẽ cho các đại lý bất động sản. Đại lý có thể dễ dàng quản lý danh sách BĐS, theo dõi khách hàng tiềm năng, và quản lý các giao dịch hiện tại và trước đó. Họ có thể tạo ra các báo cáo tổng quan, theo dõi tiến độ giao dịch, và tương tác với khách hàng một cách thuận tiện và nhanh chóng.



Hình 7.1 Giao diện trang chủ của phần mềm

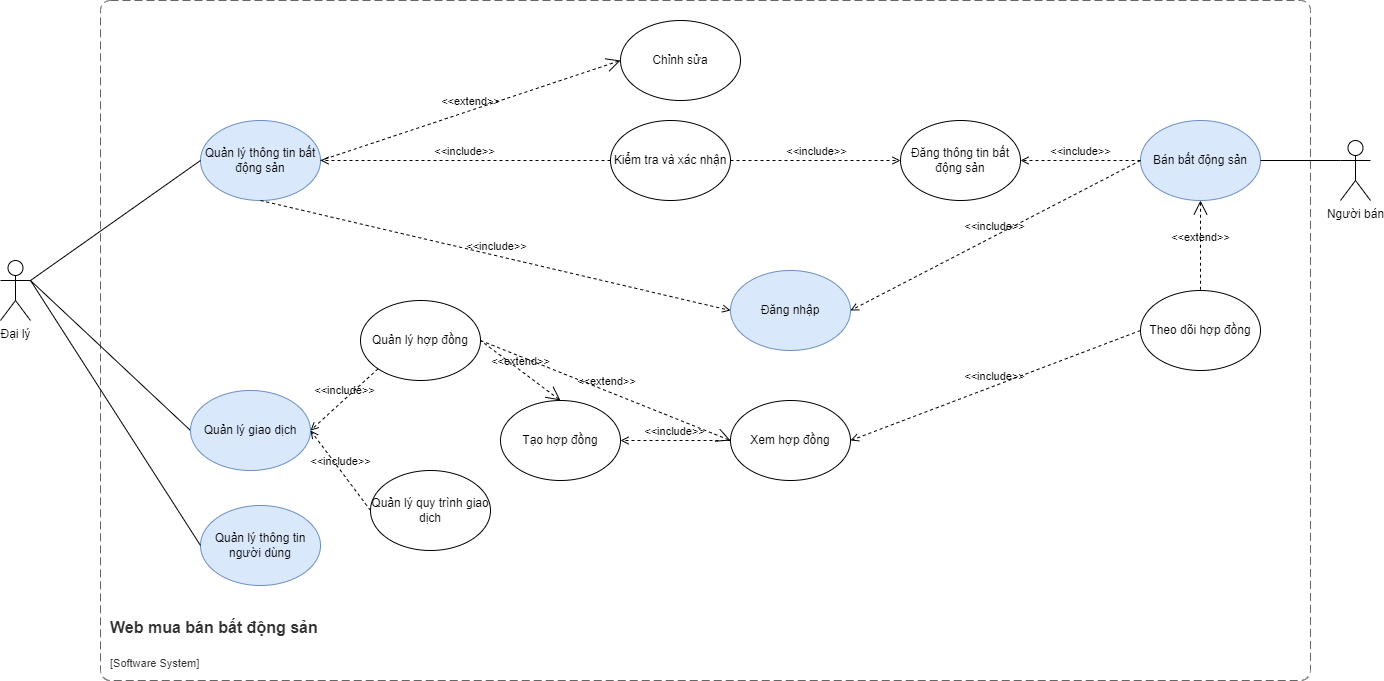
Trong phạm vi báo cáo em sẽ tiến hành phân tích và thực hiện kiểm thử cho các use case sau:

***- Use case “ Đăng nhập”***

***- Use case “Tạo thông tin bất động sản bán”***

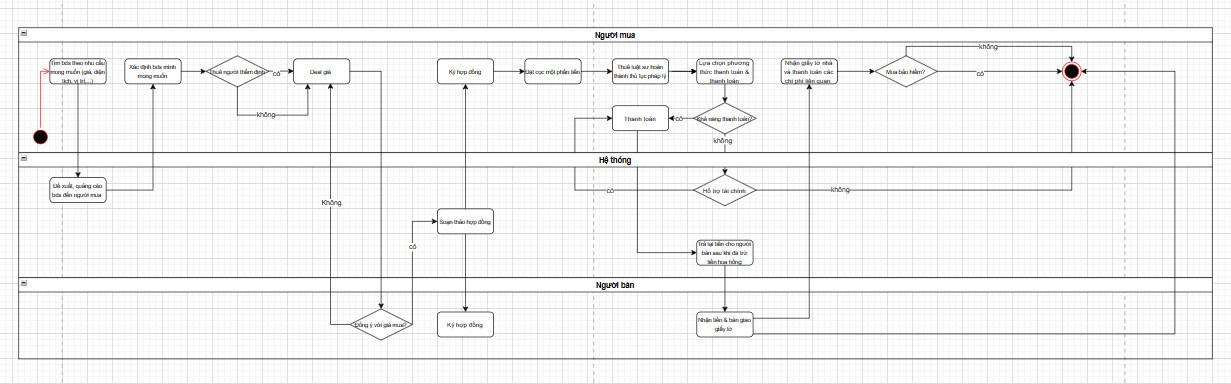
***🡪 Dưới dây là nội dung phân tích nghiệp vụ chi tiết cho các use case đã hoàn thành kiểm thử.***

### **2.1.2 Sơ đồ use case tổng quát**



Hình 8.1 Use case tổng quát

### **2.1.3 Workflow của hệ thống**

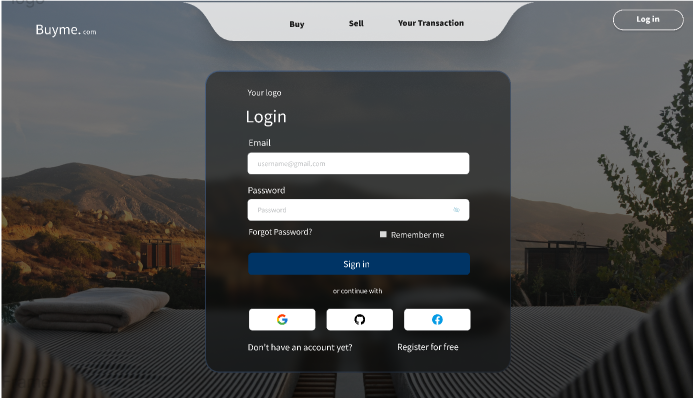


Hình 9.1 Workflow của hệ thống

### **2.1.4 Phân tích các giao diện – chức năng**

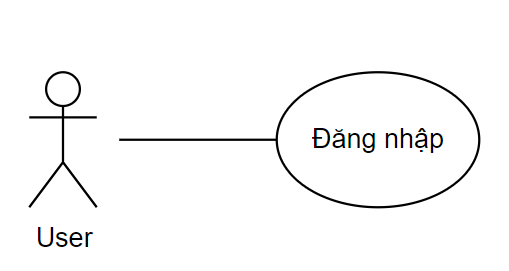
**Đặc tả chức năng “Đăng nhập”:**

1. ***Giao diện màn hình “Đăng nhập”***



Hình 10.1 Giao diện đăng nhập

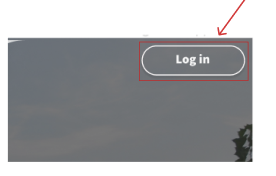
1. ***Đặc tả Use case “ Đăng nhập”***

******

|  |  |
| --- | --- |
| **Tên sơ đồ chức năng** | Login |
| **Yêu cầu mức cao** | Cho phép user đăng nhập vào hệ thống website |
| **Người sử dụng** | admin |
| **Mô tả** | Đăng nhập hệ thống website Bất Động Sản |
| **Kích hoạt** | Link của website Bất Động Sản |
| **Điều kiện tiên quyết** | N/A |
| **Xử lý sau** | N/A |

**Mô tả chi tiết :**

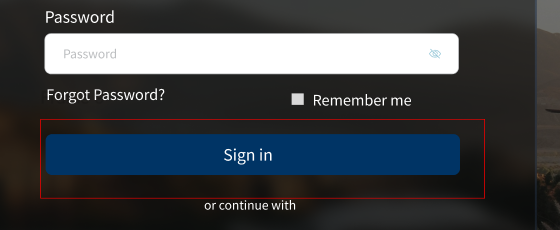
**Bước 1 :** Người dùng click vào “Login”

****

**Bước 2:** Hệ thống sẽ hiển thị màn hình đăng nhập, yêu cầu người dùng nhập các thông tin vào.

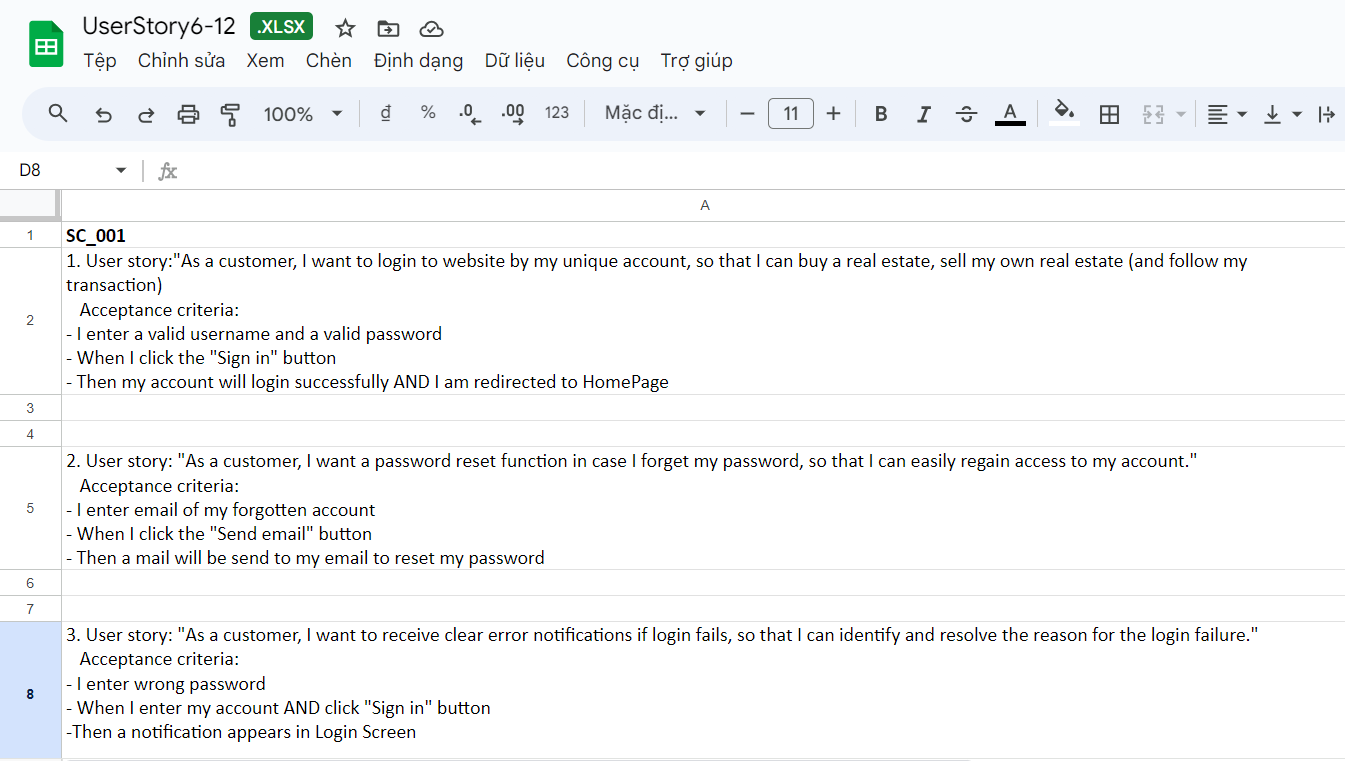
|  |  |  |  |
| --- | --- | --- | --- |
| **Mục** | **Kiểu** | **Required?** | **Mô tả** |
| Mail | TextField – String (50) | Yes | Nhập Mail của admin |
| Password | TextField – String (50) | Yes | Nhập password của admin. Data sẽ được che bởi dấu “.” hoặc “\*” |
| Remember me | Checkbox |  | Khi tick vào ô Remember me thì thông tin đăng nhập sẽ được giữ lại trong 2 textbox trên sau khi logout.  Mặc định là unchecked |
| Sign in | Button |  | Đăng nhập hệ thống |

**Bước 3**: Sau khi nhập đủ các trường của hệ thống thì người dùng tiến hành click vào button “Sign In ” để tiến hành đăng nhập tài khoản. Và hoàn tất đăng nhập .



Hình 11.1 Click vào button " Sign In"

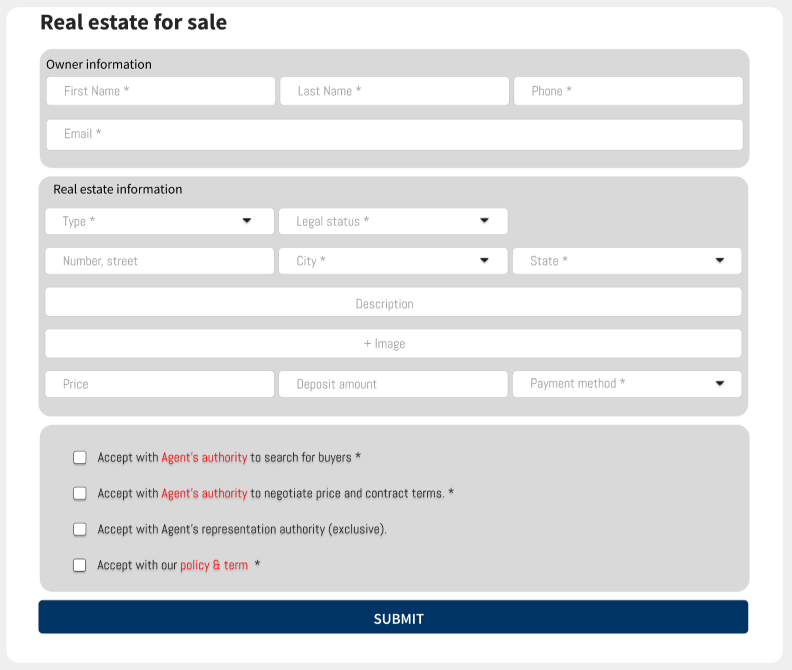
|  |  |  |  |
| --- | --- | --- | --- |
| **Admin** | | **System** | |
| **Main Flow: Login system** | | | |
| **1** | **Admin nhập Mail và Password. Click vào button Sign in** |  |  |
|  |  | **2** | **Kiểm tra Mail và Password tồn tại trong DB table Account chưa** |
|  |  | **2.1** | **Nếu đã tồn tại và người dùng có role là Admin thì hiển thị Home page - Admin** |
|  |  | **2.3** | **Nếu không tồn tại thì báo message tương ứng** |



Hình 12.1 User story về giao diện đăng nhập

**Đặc tả chức năng “Tạo thông tin BĐS bán ”:**

1. ***Giao diện màn hình “Tạo thông tin BĐS bán”***

****

Hình 13.1 Giao diện màn hình “Tạo thông tin BĐS bán”

1. ***Đặt tả Use case “ Tạo thông tin* BĐS *bán”***

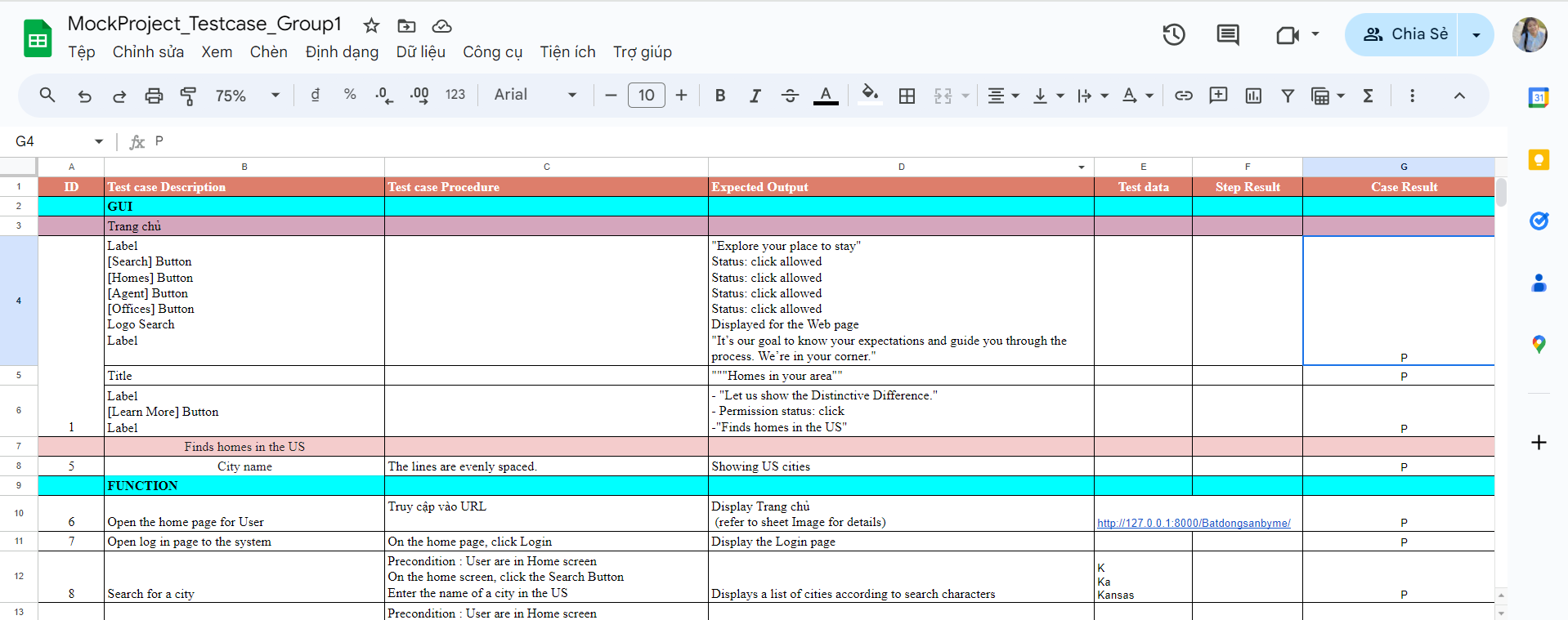
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tên sơ đồ chức năng** | | **Menu SELL** | | | |
| **Yêu cầu mức cao** | | **Hiển thị danh sách Project có trong hệ thống** | | | |
| **Người sử dụng** | | **Account có role là ‘Admin’** | | | |
| **Mô tả** | | **Sơ đồ chức năng này cho phép admin tìm/xem các Project có trong hệ thống** | | | |
| **Kích hoạt** | | **NA** | | | |
| **Điều kiện tiên quyết** | | **Người dùng phải đăng nhập thành công vào hệ thống** | | | |
| **Mục** | **Kiểu** | | **Dữ liệu** | **Bắt buộc** | **Mô tả** |
| **First Name** | **TextField – String (50)** | | **Dữ liệu mặc định là trống** | **Yes** | **First Name của chủ sở hữu được tạo mới.** |
| **Last Name** | **TextField – String (50)** | | **Dữ liệu mặc định là trống** | **Yes** | **Last Name của chủ sở hữu được tạo** |
| **Phone** |  | | **Dữ liệu mặc định là trống** | **Yes** | **Phone của chủ sở hữu được tạo** |
| **Email** | **Combobox** | | **Dữ liệu mặc định là trống** | **Yes** | **Email của chử sở hữu được tạo.** |
| **Type** | **TextField – String (50)** | | **Dữ liệu mặc định là trống.** | **Yes** | **Click vào sẽ hiện những dạng bất động sản (Nhà ở, văn phòng, mặt bằng…)** |
| **Legat status** | **Combobox** | | **Dữ liệu mặc định** | **Yes** | **Cho biết tình trạng pháp lý của bất động sản** |
| **Number,Stree** | **Combobo** | | **Dữ liệu mặc định là trống.** | **Yes** | **Cho biết chính xác địa chỉ bất động sản tại đường nào** |
| **City** | **TextField – String (50)** | | **Dữ liệu mặc định là trống.** | **Yes** | **CHo biết Bất động sản nằm ở thành phố** |
| **State** | **TextField – String (100)** | | **Dữ liệu mặc định là trống.** | **Yes** | **Cho biết tình trạng hư hại, hao mòn của bất động sản** |
| **Description** | **TextField – String (100)** | | **Dữ liệu mặc định là trống.** | **Yes** | **Miêu tả rõ bất động sản( diện tích, những tiện ích…)** |
| **Image** |  | | **Dữ liệu mặc định là trống.** | **Yes** | **Hình ảnh cụ thể về bất động sản** |
| **Price** | **Combobo** | | **Dữ liệu mặc định là trống.** | **Yes** | **Giá tiền của bất động sản** |
| **Deposit amount** | **Combobo** | | **Dữ liệu mặc định là trống.** | **Yes** | **Số tiền cọc khi muốn mua bất động sản** |
| **Payment method** | **Combobo** | | **Dữ liệu mặc định là trống.** | **Yes** | **Phương thức thanh toán bất động sản** |

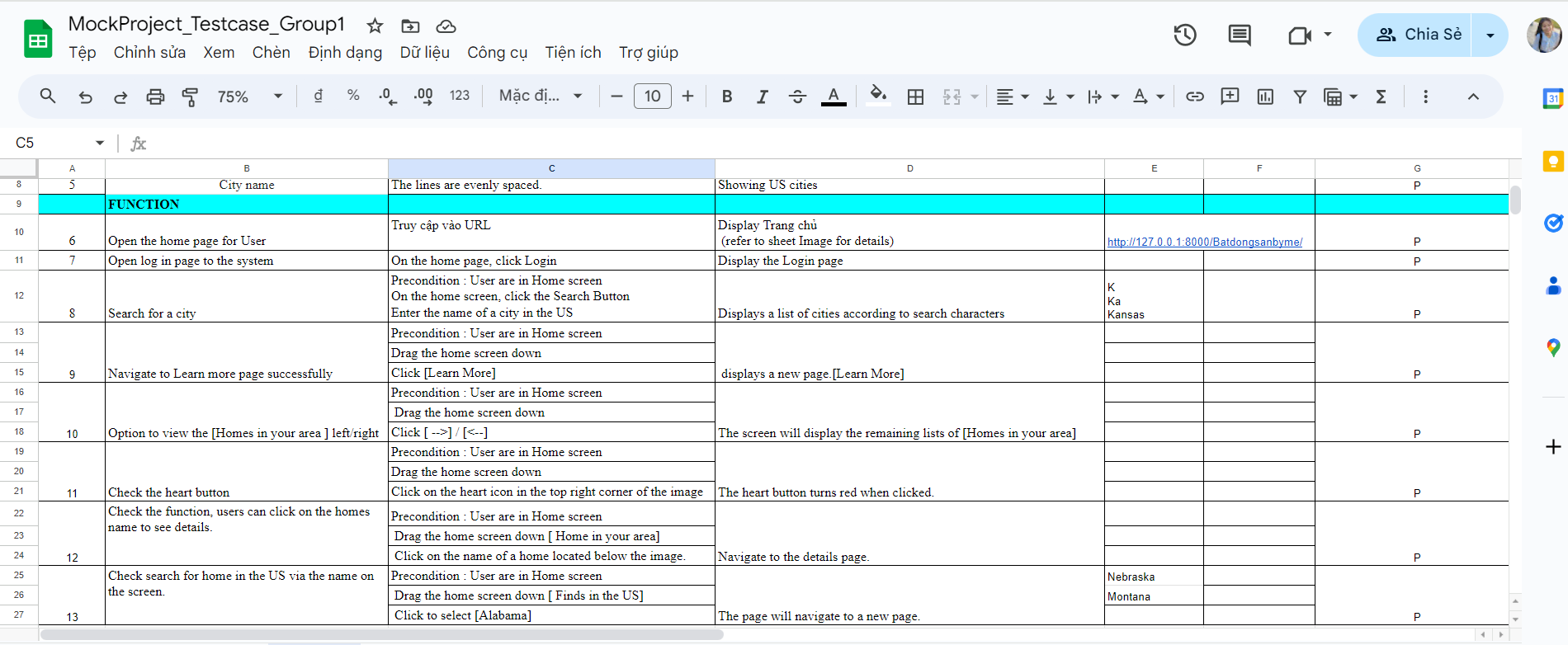
|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Nhập thông tin của chủ sở hữu và bất động sản** | | | |
| **1** | **Nhập thông tin chủ sở hữu** |  |  |
|  |  | **2** | **lưu thông tin chủ sở hữu** |
| **3** | **Nhập thông tin bất động sản** |  |  |
|  |  | **4** | **lưu thông tin bất động sản** |

# **CHƯƠNG 3: TRIỂN KHAI THỰC NGHIỆM**

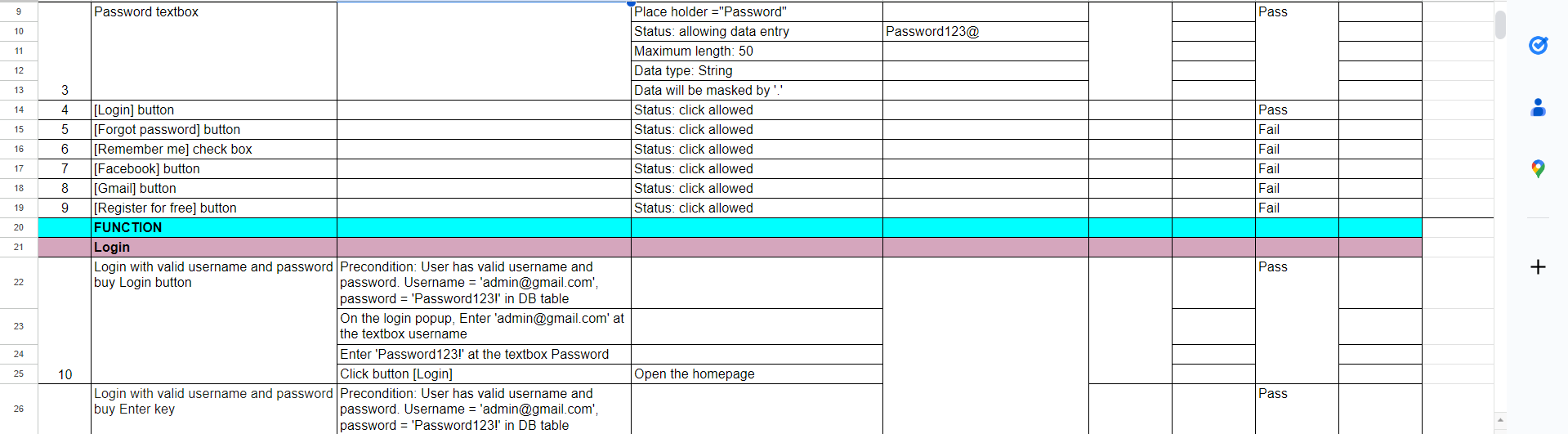
## **3.1 Thực hiện kiểm thử Manual Testing trên phần mềm**

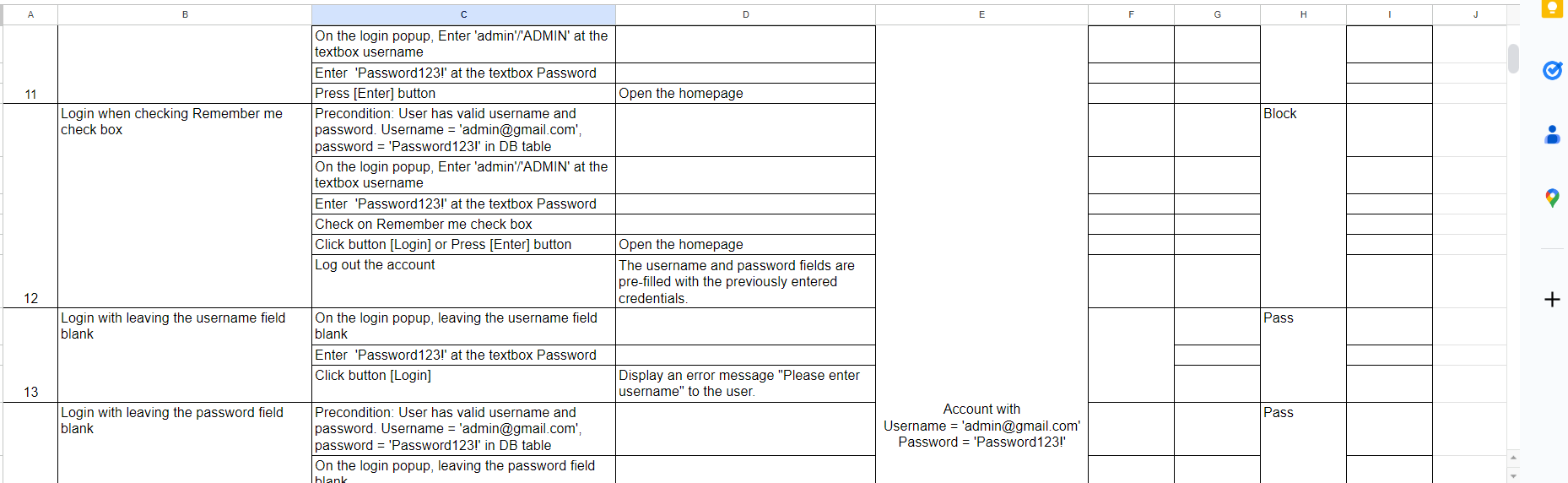
### **3.1.1 Testcase màn hình HomPage**

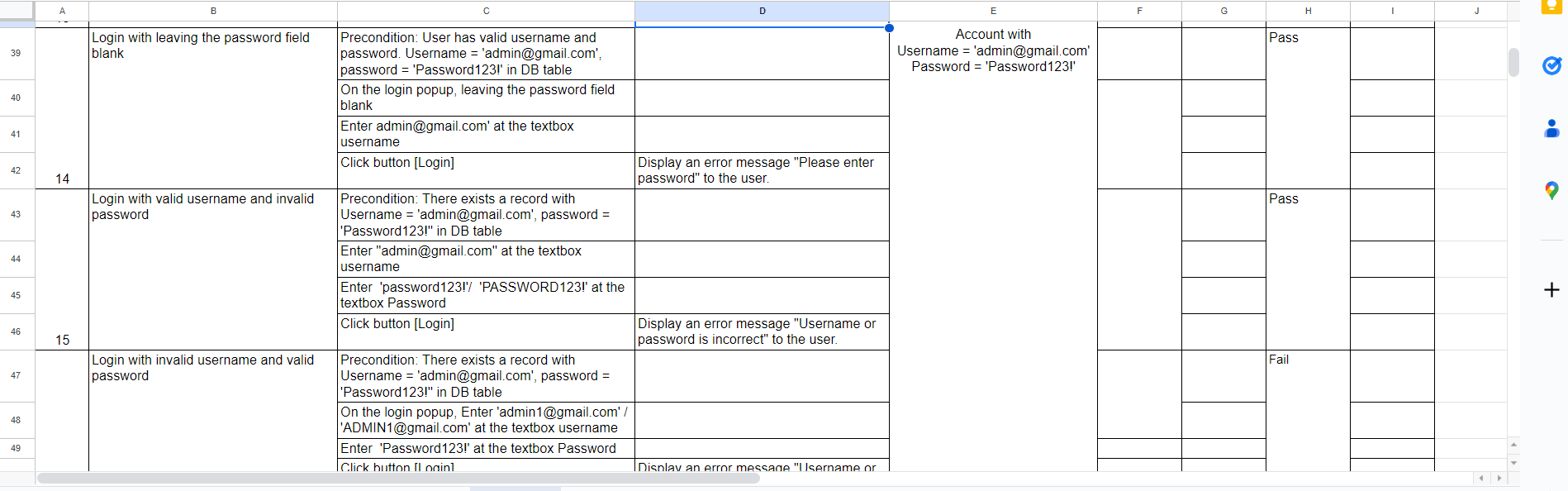


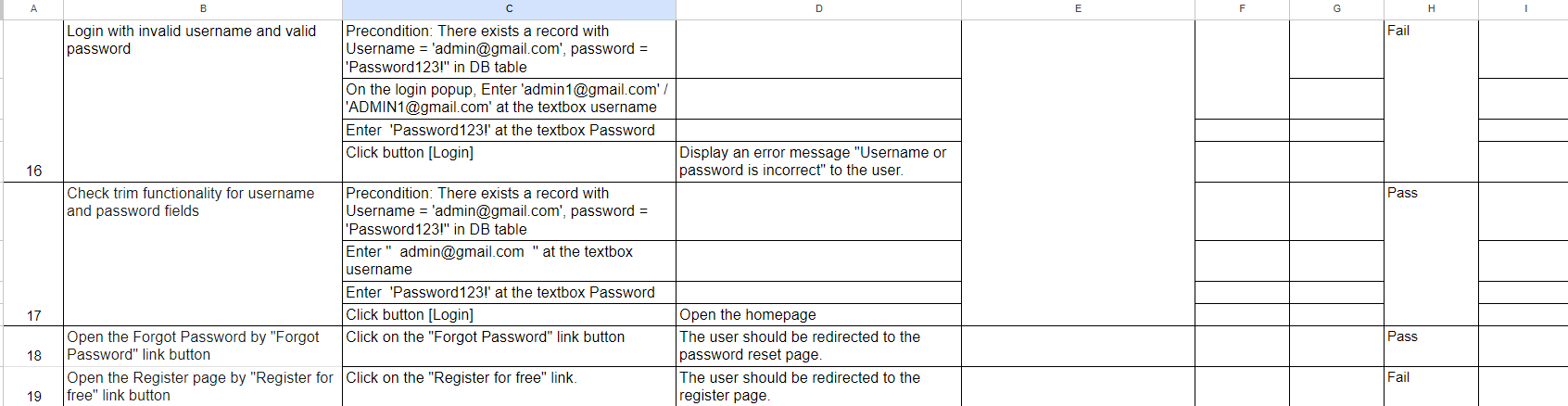


### **3.1.2 Testcase màn hình Login**

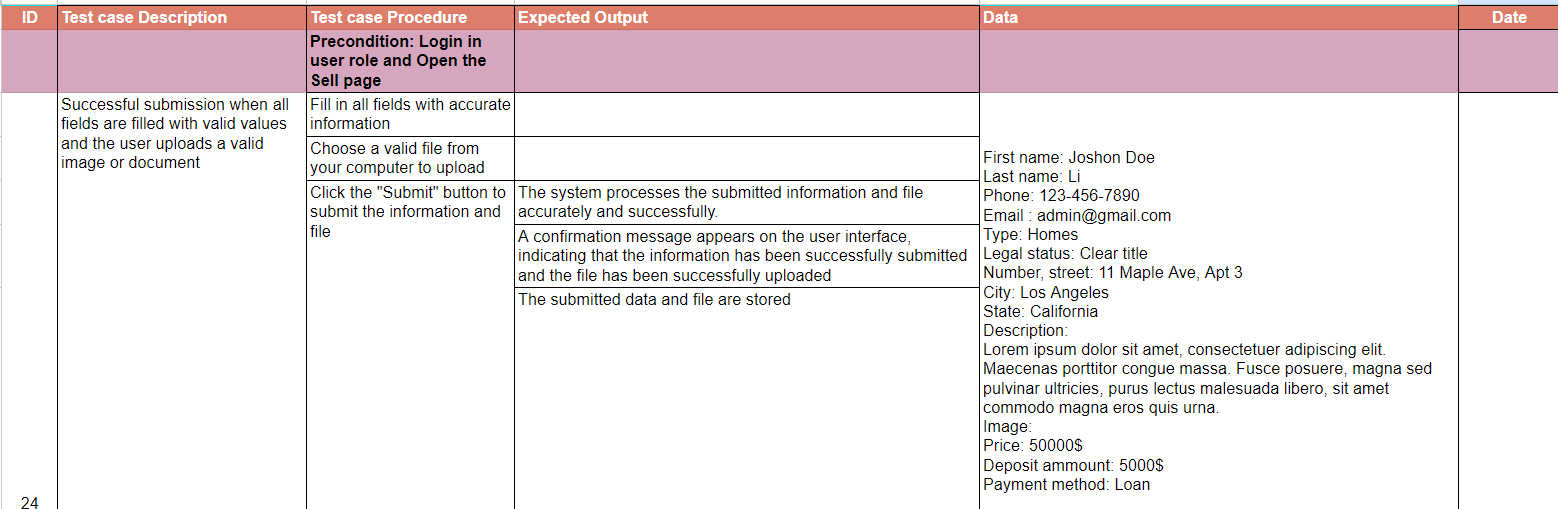


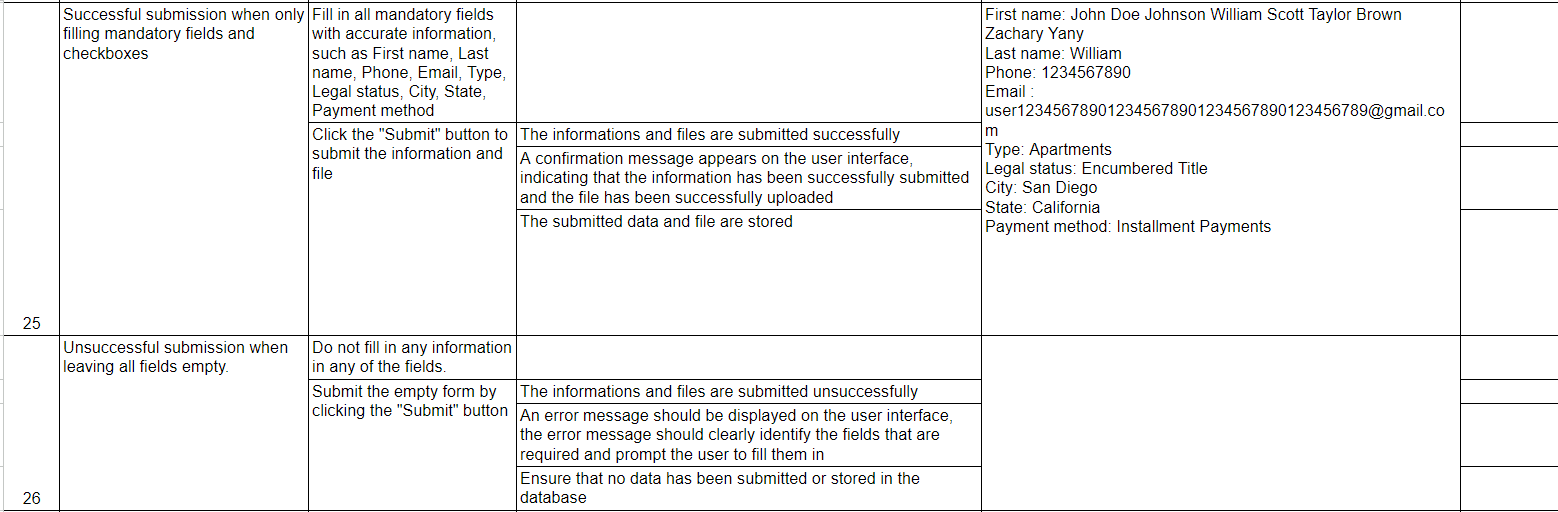


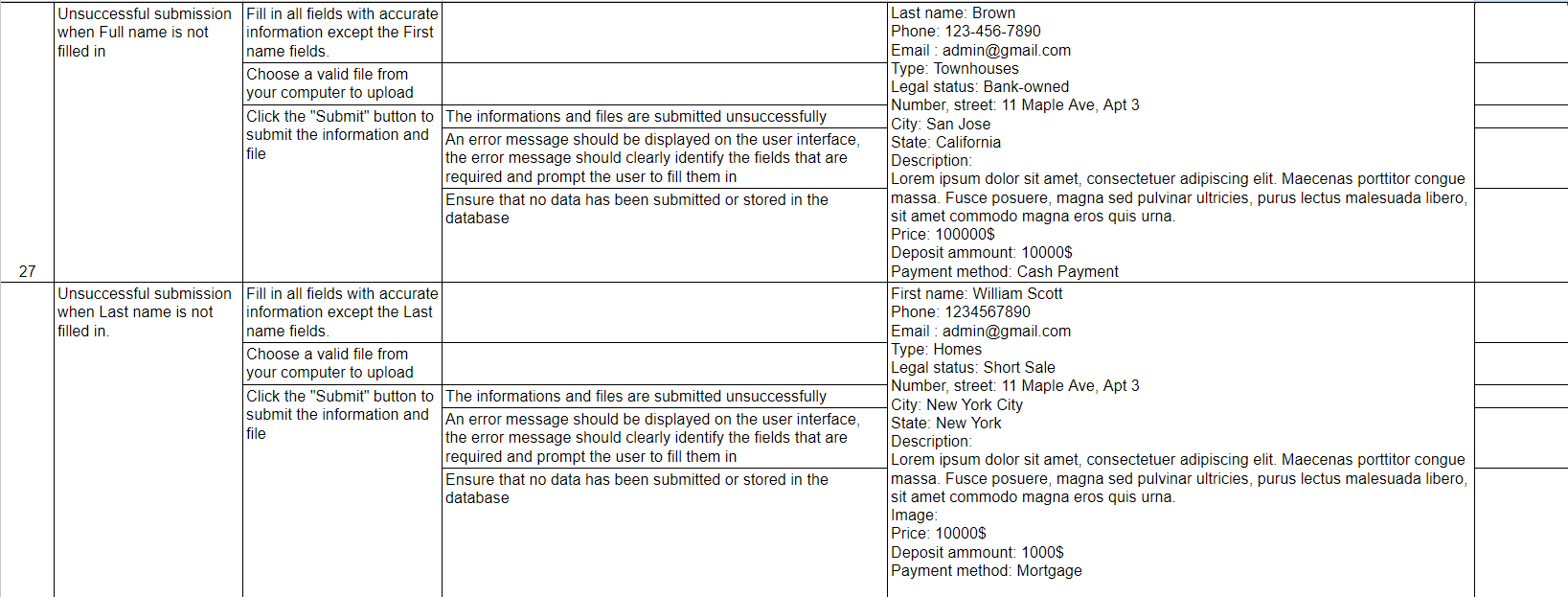


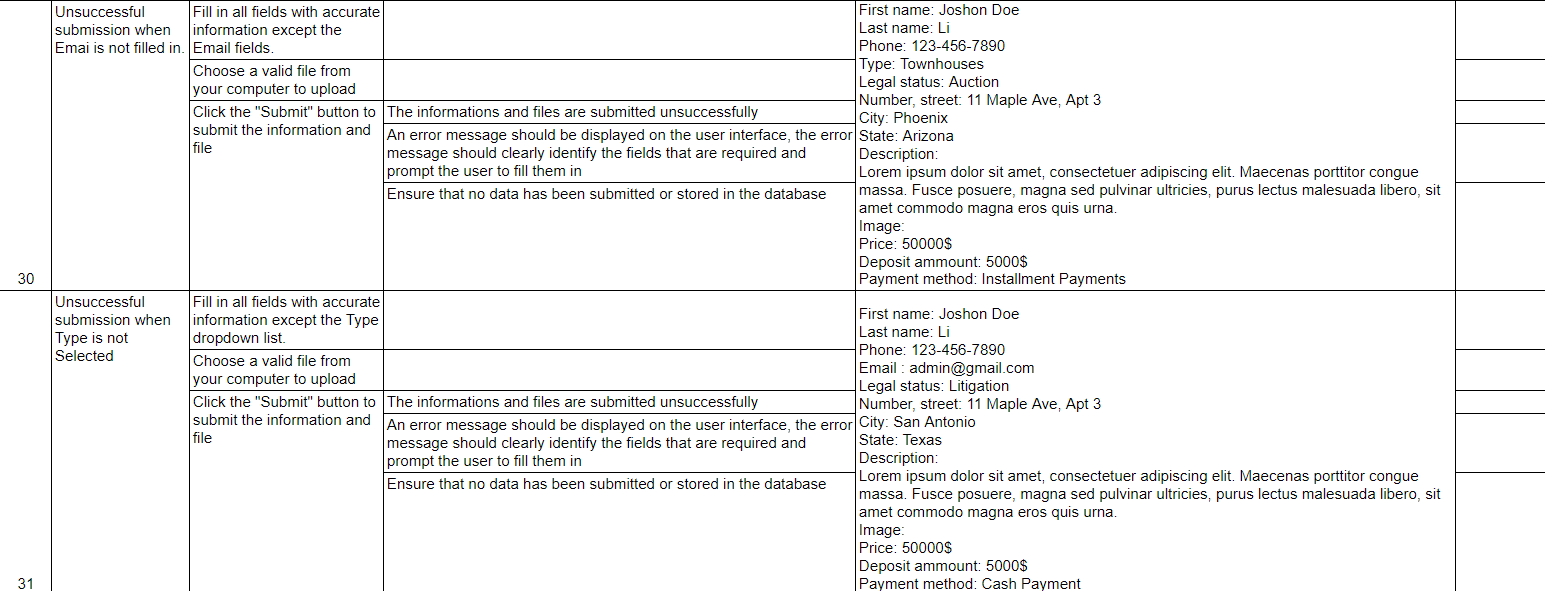


### **3.1.3 Testcase màn hình “Tạo thông tin BĐS bán”**

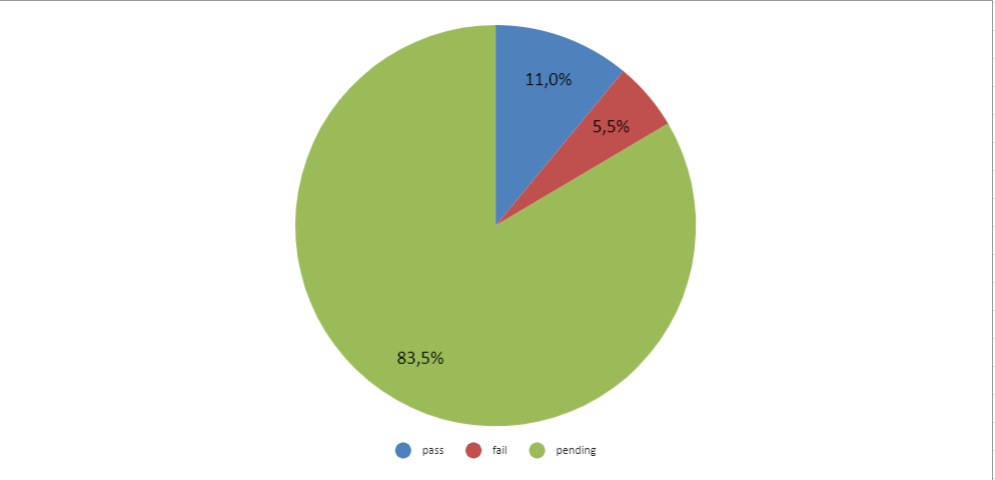


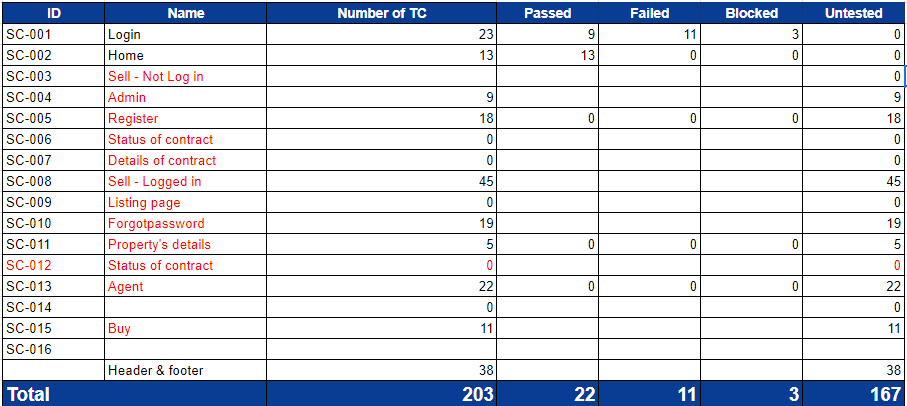






### **3.1.4 Kết quả của kiểm thử manual testing**





Hình 14.1 Màn hình kết quả testcase

## **3.2 Nghiên cứu Automation Testing trên phần mềm bằng Selenium Framework**

### **3.2.1 Giới thiệu công cụ kiểm thử tự động Selenium**

**Selenium** là một phần mềm mã nguồn mở - một công cụ kiểm thử phần mềm tự động để kiểm thử các ứng dụng trên nền Web. Năm 2004, Selenium được phát triển bởi ThoughtWorks với cái tên ban đầu JavaScript Test Runner. Đến năm 2007, tác giả Jason Huggins rời ThoughtWorks và gia nhập Selenium Team (thuộc Google), từ đó tiếp tục phát triển Selenium như hiện nay.

**Selenium IDE** là một công cụ cho phép chúng ta ghi lại một kịch bản và tái sử dụng kịch bản đó. Nó hoạt động như một Add-on của trình duyệt Mozilla Firefox với giao diện trực quan, dễ sử dụng ngay cả với những kiểm thử viên không biết về code. Với Selenium IDE, chúng ta chỉ có thể ghi lại kịch bản (Record) trên trình duyệt Mozilla Firefox nhưng có thể tái sử dụng kịch bản này trên nhiều trình duyệt khác như Internet Explorer, Google Chrome,v.v

**Selenium WebDriver** là phiên bản kế nhiệm của Selenium RC. Cũng giống như Selenium RC, Selenium WebDriver hỗ trợ viết kịch bản kiểm thử bằng các ngôn ngữ khác nhau như Java, .NET, PHP, Python, Perl, Ruby và kiểm thử viên có thể sử dụng các điều kiện if, else hay các vòng lặp để tăng tính chính xác cho kịch bản kiểm thử. Selenium WebDriver có kiến trúc khá đơn giản, điều khiển trình duyệt trực tiếp từ hệ điều hành.

### **3.2.2 Đặc điểm của Selenium**

- Selenium là một công cụ mã nguồn mở: Vì là mã nguồn mở nên chúng ta có thể sử dụng mà không phải lo lắng về phí bản quyền hay thời hạn sử dụng.

- Các thử nghiệm sau đó có thể được chạy trên các trình duyệt web hiện đại nhất.

- Selenium hỗ trợ chạy trên nhiều OS khác nhau, Selenium triển khai trên nền tảng Windows, Linux và Mac.

- Kiểm thử có thể được exported ở hầu hết các ngôn ngữ ví dụ: HTML, Java, .Net, perl, ruby, ...

- Selenium có sự hỗ trợ của một số nhà cung cấp trình duyệt lớn hỗ trợ.

### **3.2.3 Môi trường Selenium**

- Môi trường phát triển: sử dụng công cụ hỗ trợ Eclipse.

- Ngôn ngữ lập trình: Java.

- Trình duyệt web: sử dụng trình duyệt Chrome.

- Driver của trình duyệt: ChromeDriver.

### **3.2.4 Cài đặt selenium**

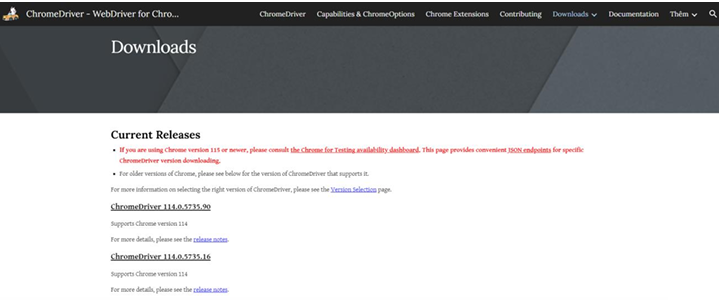
***a. Cài đặt Chrome Browser***

1. Cài đặt Chrome Browser ở trang <https://www.google.com/chrome/>

2. Tiến hành check version của Chrome, gõ vào chrome://settings/help để tiến hành kiểm tra Chrome Version.

***b. Download Chrome Driver***

1. Tải Chrome Driver tại trang <https://chromedriver.chromium.org/downloads>

****

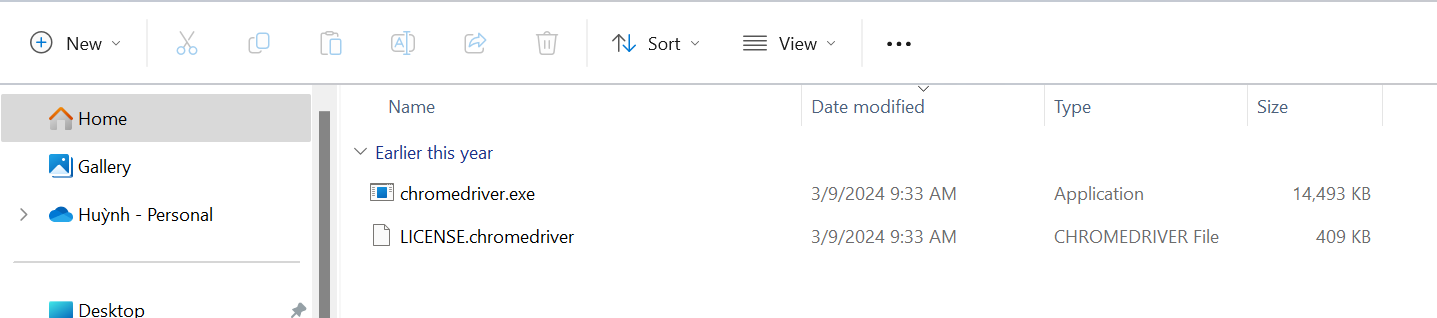
Hình 15.1 Màn hình download Chrome Driver

Ở trên dùng Chrome version 115.0.5790.171 nên phải tải về đúng với version 115.x.xxxx.xx của Chrome.



Hình 16.1Tải Version 115.x.xxxx.xx của Chrome.

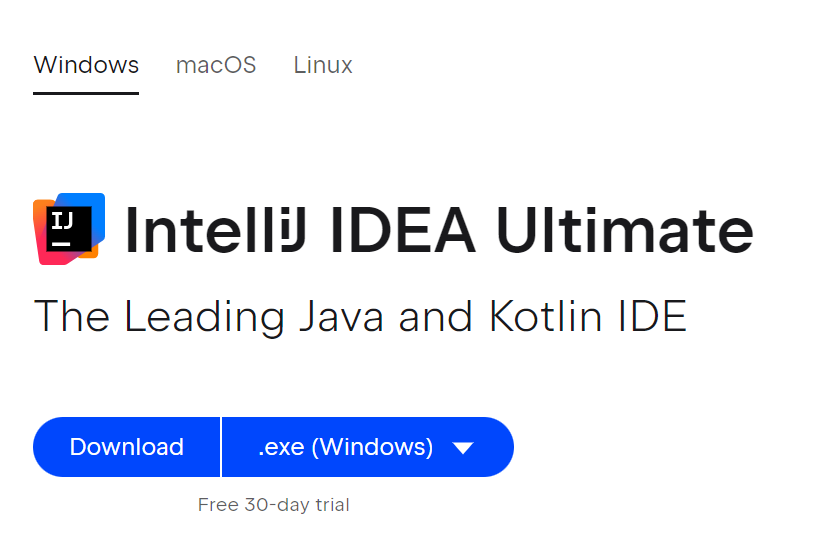
2. Sau khi tải về, các bạn giải nén ra, sẽ ra một tập tin tên là chromedriver.exe (trên Window)



Hình 17.1 Tập tin chromedriver.exe

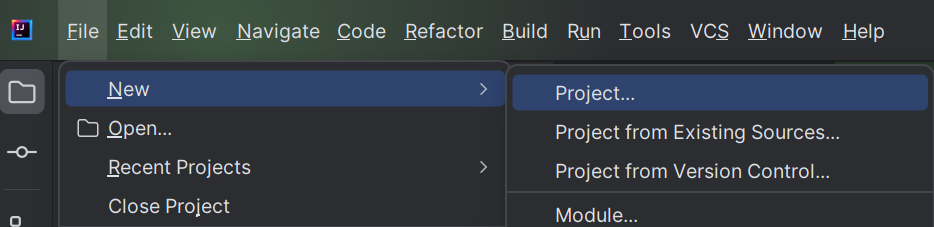
C. Bước cuối cùng tạo một project và add Selenium Webdriver vào

1.Tải Intellij IDEA từ trang web <https://www.jetbrains.com/idea/download/?section=windows>



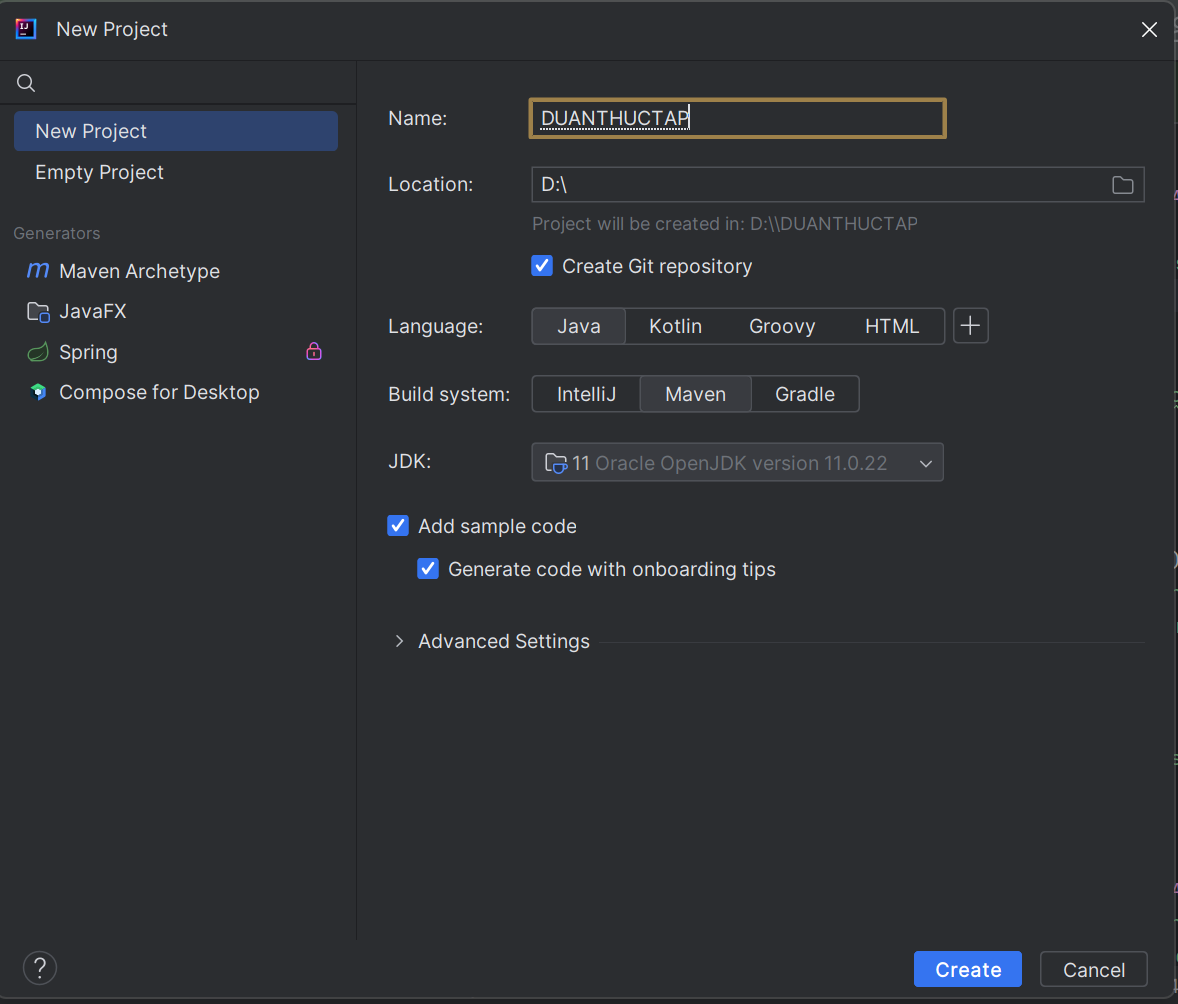
Hình 18.1 Intellij IDEA

2. Sau đó mở Intellij lên, tạo một project mới

**

Hình 19.1 Tạo project mới

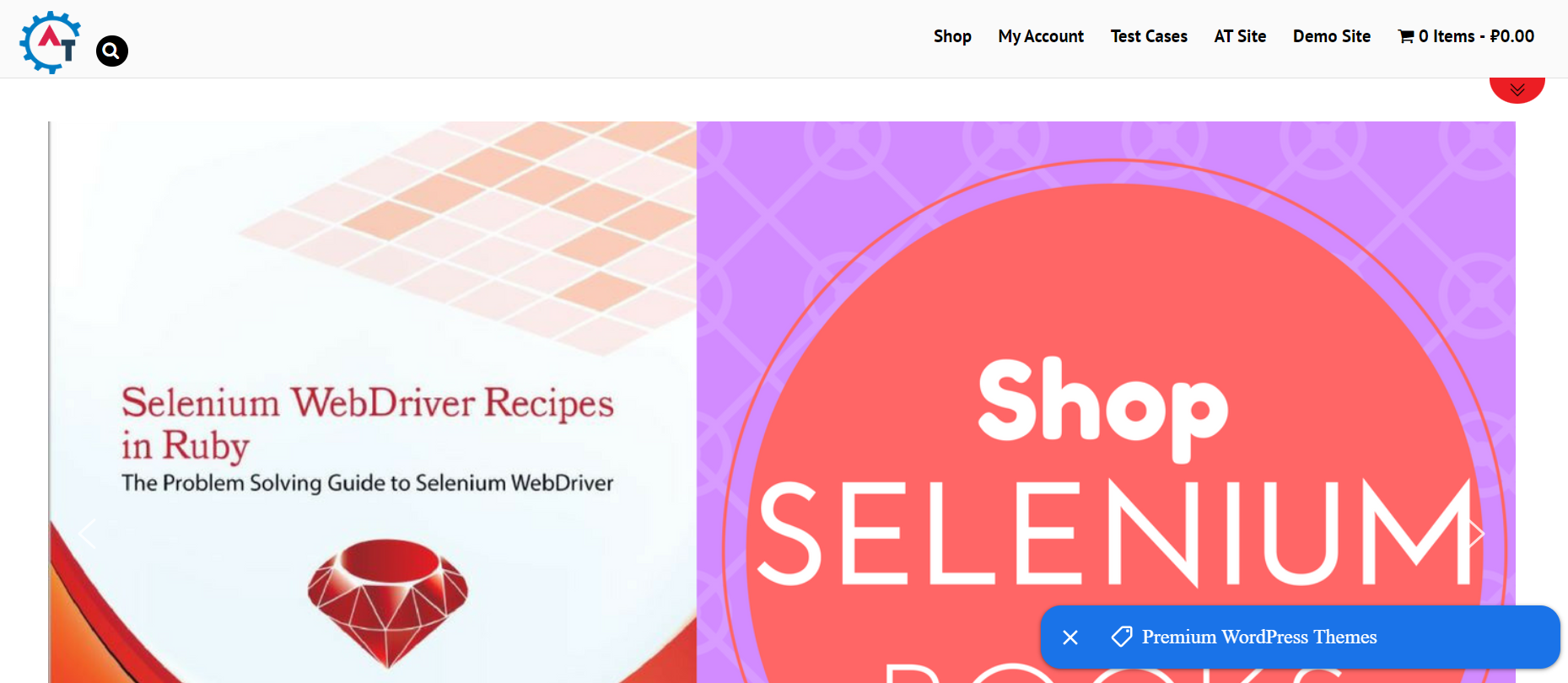
Đặt tên cho project, chọn JDK đúng với Version Java, click Create.



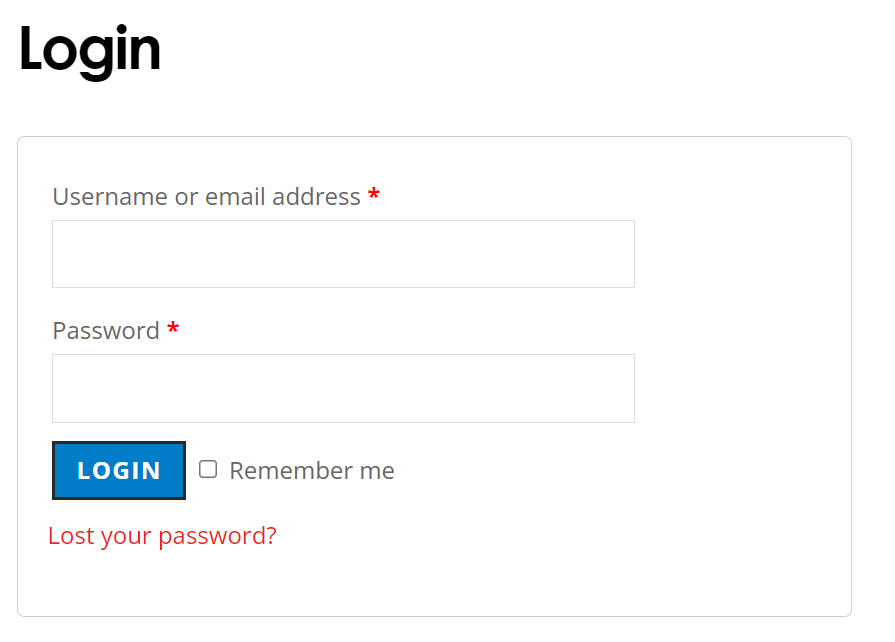
3. Add library của Selenium Webdriver vào Project, để nó có thể chạy được các command của Webdriver

### **3.2.5 Website thực hiện kiểm thử Automation**

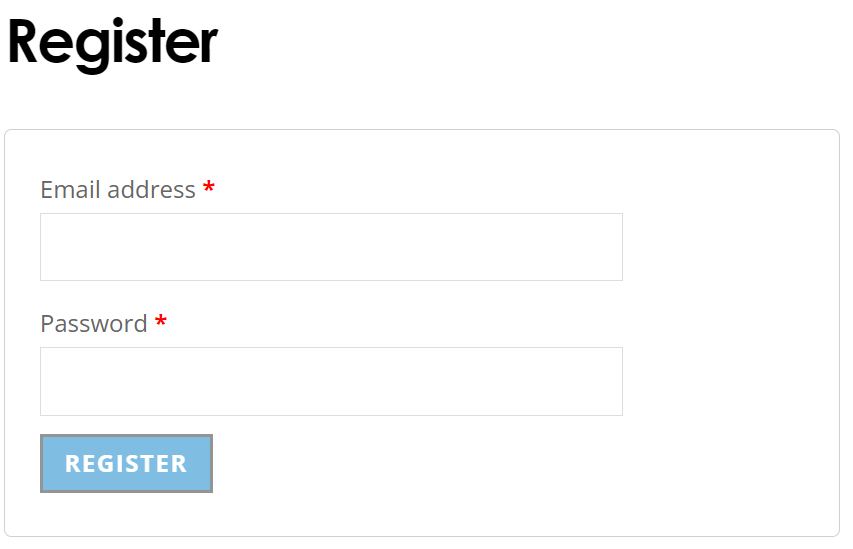
Link website: <https://practice.automationtesting.in/>



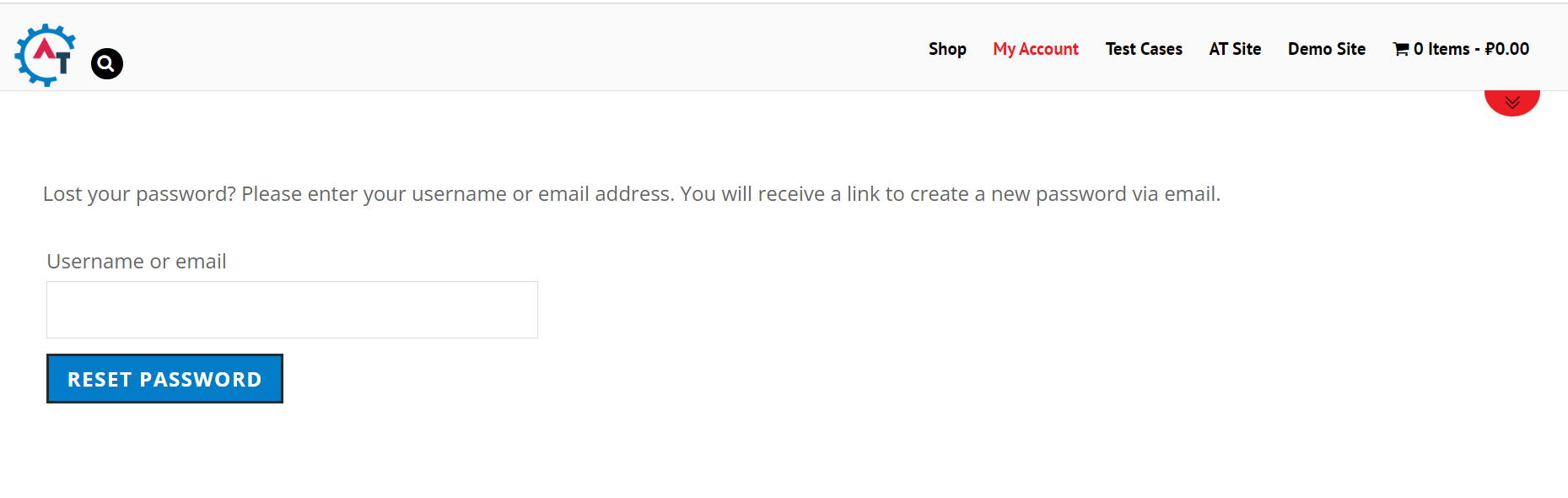
Hình 20.1 Web thực hiện automation

****

Hình 21.1 Giao diện màn hình Login



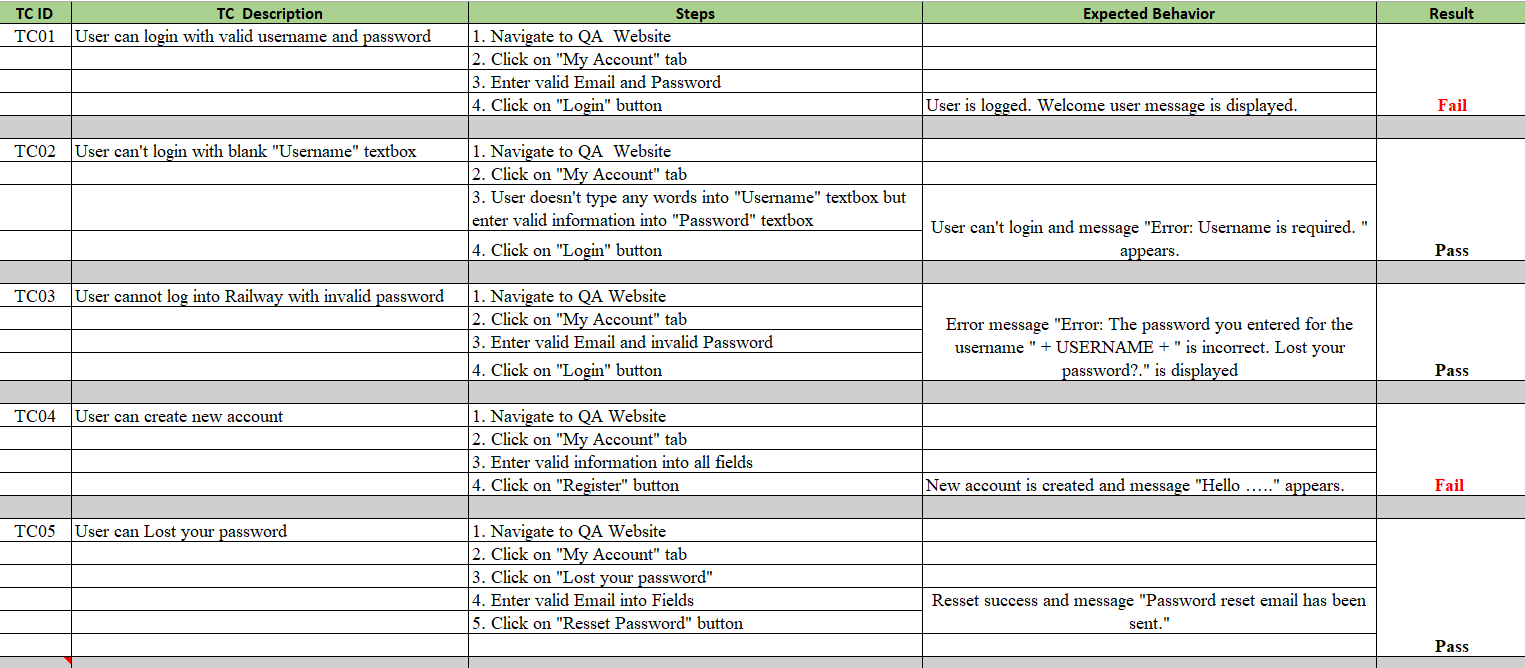
Hình 22.1 Giao diện màn hình Register



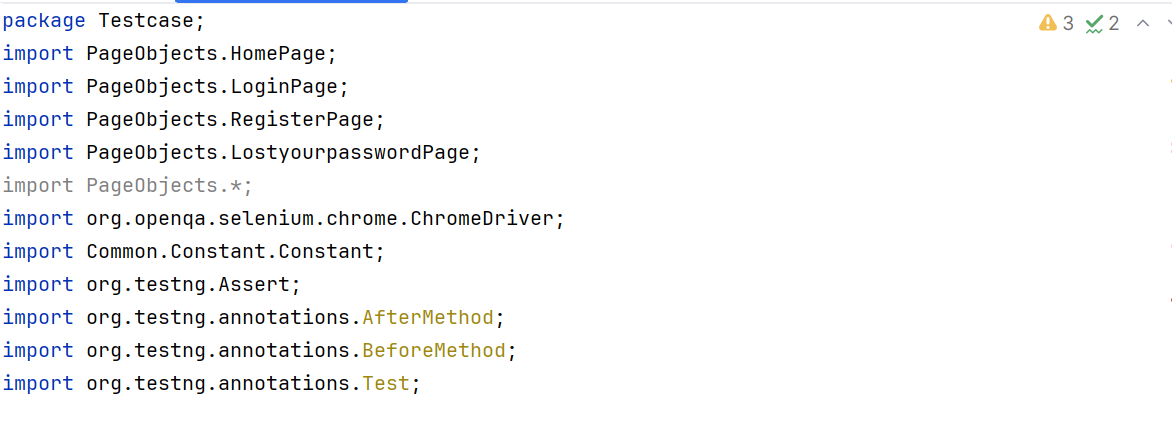
Hình 23.1 Giao diện Lost your Password

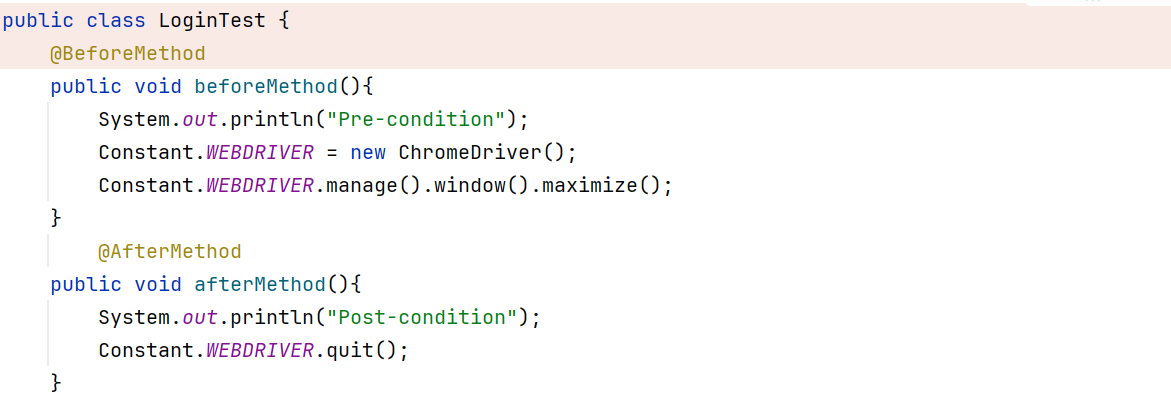
### **3.2.6 Testcase cho Automation**

Dưới dây là bảng mô tả testcase cho automation và em đã thực hiện 5 testcase như sau:



Hình 24.1 Testcase automation





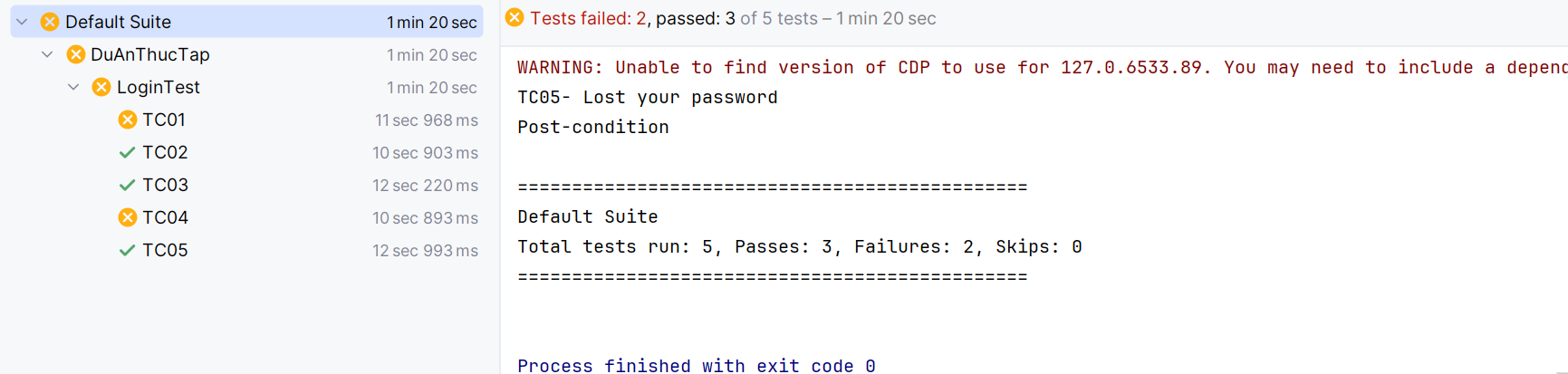






Hình 25.1 Màn hình code testcase

### **3.2.7 Kết quả run code Test Case Automation**



Hình 26.1 Mành hình kết quả code Automation

# **KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

## **1. Kết quả đạt được**

Sau khoảng thời gian thực tập tại công ty TNHH Công Nghệ Số Việt Trí Đạo em đã đạt được những kiến thức và kết quả tích cực như sau:

• Thiết kế testcase hoàn chỉnh và chuyên nghiệp.

• Cải thiện khả năng sử dụng tiếng Anh, áp dụng vào công việc khi thực tập.

• Áp dụng được những kiến thức đã học vào thực tế.

• Có thêm nhiều kiến thức mới mẻ chưa được học, kiến thức liên quan về mảng Tester cũng như các công việc liên quan được nâng cao.

## **2. Hạn chế và hướng phát triển**

**• Hạn chế:**

- Thời gian thực hiện đề tài còn hạn chế, kiến thức lại quá nhiều nên bản thân còn chưa tìm hiểu sâu rộng, điều này khiến việc viết test case,test design còn nhiều thiếu sót.

**• Hướng phát triển:**

*- Về phía đề dự án:* Tiếp tục cập nhật và hoàn thiện các ca kiểm thử cho những phần còn lại của hệ thống

+ Mở rộng phạm vi bài báo cáo với việc phân tích hết các tính năng của hệ thống.

*- Về phía bản thân:*

+ Nâng cao kiến thức của bản thân về lĩnh vực kiểm thử

+ Tìm hiểu thêm kiểm tra tự động, các công cụ khác để linh hoạt công việc, áp dụng vào hệ thống nhằm nâng cao hiệu quả

+ Cải thiện kỹ năng tiếng anh để phục vụ cho công việc

# **CHECK LIST CỦA BÁO CÁO**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Nội dung công việc** | **Có** | **Không** | **Ghi chú** |
| 1 | Báo cáo được trình bày (định dạng) đúng với yêu cầu. | X |  |  |
| 2 | Báo cáo có số lượng trang đáp ứng đúng yêu cầu (30-50 trang) | X |  |  |
| 3 | Báo cáo trình bày được phần mở đầu bao gồm: Mục tiêu, Phạm vi và đối tượng, kết cấu … | X |  |  |
| 4 | Báo cáo trình bày về công ty, vị trí việc làm (công việc đó làm gì, kiến thức và kỹ năng cần thiết là gì, con đường phát triển sự nghiệp (career path)), cơ sở lý thuyết phù hợp với nội dung của đề tài (Tối đa 10-12 trang) | X |  |  |
| 5 | Báo cáo có sản phẩm cụ thể phù hợp với mục tiêu đặt ra của đề tài | X |  |  |
| 6 | Báo cáo có phần kết luận và hướng phát triển của đề tài | X |  |  |

# **PHỤ LỤC**

# **TÀI LIỆU THAM KHẢO**

1. Tài liệu hướng dẫn sử dụng của công ty.
2. [https://viblo.asia/p/quy-trinh-va-ke-hoach-kiem-thu-phan-mem L4x5xyRaKBM](https://viblo.asia/p/quy-trinh-va-ke-hoach-kiem-thu-phan-mem%20L4x5xyRaKBM%20)

3. <https://viblo.asia/p/what-is-test-data-QpmleAv9lrd>

4. [https://viblo.asia/p/bai-dich-test-data-la-gi-huong-dan-thiet-ke-test-data dWrvwWjbvw38](https://viblo.asia/p/bai-dich-test-data-la-gi-huong-dan-thiet-ke-test-data%20dWrvwWjbvw38)

5. <https://www.youtube.com/watch?v=qXgo4cyR3n8>