**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

**======\*\*\*======**



**BÁO CÁO BÀI TẬP LỚN**

**KIỂM THỬ PHẦN MỀM**

**ĐỀ TÀI:**

**TÌM HIỂU VỀ CÔNG CỤ KIỂM THỬ SELENIUM IDE**

**VÀ KIỂM THỬ WEBSITE ĐIỆN MÁY CHỢ LỚN**

|  |  |
| --- | --- |
| GVHD | : Th.S Nguyễn Ngọc Quang |
| Lớp | : 20241IT6084006 |
| Nhóm | : 9 |
| Sinh viên thực hiện: | |
| 1. Nguyễn Đặng Hải Quân 2022604407 2. Lê Thế Sơn 2022605688 3. Nguyễn Thanh Tùng 2022607244 4. Cù Minh Hải 2022604136 5. Giáp Văn Duy 2022607127 | |

*Hà Nội - Năm 2024*

# *LỜI CẢM ƠN*

*Nhóm chúng em xin chân thành gửi lời cảm ơn đến thầy ThS. Nguyễn Ngọc Quang đã tận tâm và hướng dẫn giúp đỡ trong quá trình thực hiện đề tài này. Chúng em rất trân trọng những kiến thức mà thầy đã chia sẻ với chúng em, không chỉ trong lĩnh vực học tập mà còn cả những lời khuyên, chia sẻ thực tế quý giá. Phương pháp dạy học của thầy đã mang lại cho chúng em cơ hội khám phá và phát triển tiềm năng của bản thân. Những buổi thuyết trình, giao lưu với thầy và các bạn trong lớp cũng đã giúp chúng em rèn luyện tự tin, kỹ năng giao tiếp và làm việc nhóm, đóng vai trò quan trọng trong việc chuẩn bị cho môi trường làm việc thực tế.*

*Để hoàn thành báo cáo này, chúng em đã cùng nhau nghiên cứu, thảo luận và áp dụng kiến thức từ lớp học cũng như từ các nguồn tài liệu trên Internet, cùng với những trải nghiệm cá nhân của mỗi thành viên trong nhóm. Chúng em rất mong nhận được những ý kiến, đề xuất từ thầy cô và bạn đọc, để đề tài này càng trở nên hoàn thiện hơn.*

*Chúng em xin chân thành cảm ơn thầy ThS. Nguyễn Ngọc Quang vì sự đồng hành và hỗ trợ quý báu của thầy. Chúng em sẽ luôn ghi nhớ những kiến thức và kinh nghiệm mà thầy đã truyền đạt và hy vọng có thể gặp lại thầy trong những dịp tiếp theo.*

***Nhóm sinh viên thực hiện.***

MỤC LỤC

[*LỜI CẢM ƠN* 1](#_Toc185953208)

[DANH MỤC CÁC THUẬT NGỮ, KÝ HIỆU VÀ CÁC CHỮ VIẾT TẮT 5](#_Toc185953209)

[DANH MỤC HÌNH ẢNH 6](#_Toc185953210)

[DANH MỤC BẢNG BIỂU 8](#_Toc185953211)

[PHẦN I. MỞ ĐẦU 9](#_Toc185953212)

[1. Tên đề tài 9](#_Toc185953213)

[2. Lý do chọn đề tài 9](#_Toc185953214)

[3. Mục đích đề tài 9](#_Toc185953215)

[4. Mục tiêu đề tài 10](#_Toc185953216)

[5. Bố cục đề tài 10](#_Toc185953217)

[PHẦN II. KẾT QUẢ NGHIÊN CỨU 11](#_Toc185953218)

[Chương 1: Tổng quan về kiểm thử phần mềm 11](#_Toc185953219)

[1.1 Khái niệm 11](#_Toc185953220)

[1.2 Vai trò của kiểm thử phần mềm 11](#_Toc185953221)

[1.3 Phân loại kiểm thử 13](#_Toc185953222)

[**1.3.1 Kiểm thử chức năng (Functional testing)** 13](#_Toc185953223)

[**1.3.2 Kiểm thử phi chức năng (Non-functional testing)** 14](#_Toc185953224)

[**1.3.3 Kiểm thử cấu trúc (Structural testing)** 15](#_Toc185953225)

[**1.3.4 Kiểm thử liên quan đến các thay đổi (Change related testing)** 16](#_Toc185953226)

[**1.3.5 Giới thiệu về kiểm thử Frontend** 17](#_Toc185953227)

[**1.3.6 Giới thiệu công cụ kiểm thử Selenium** 20](#_Toc185953228)

[Chương 2: Tìm hiểu công cụ Selenium 22](#_Toc185953229)

[2.1 Tìm hiểu công cụ kiểm thử Selenium 22](#_Toc185953230)

[**2.1.1 Selenium là gì** 22](#_Toc185953231)

[**2.1.2 Các đặc điểm của Selenium** 23](#_Toc185953232)

[**2.1.3 Selenium IDE** 24](#_Toc185953233)

[**2.1.4 Selenium Remote Control (Selenium RC)** 25](#_Toc185953234)

[**2.1.5 Selenium WebDriver** 26](#_Toc185953235)

[**2.1.6 Selenium Grid** 26](#_Toc185953236)

[2.2 Cài đặt và sử dụng công cụ Selenium 27](#_Toc185953237)

[**2.2.1 Cài đặt Selenium IDE trên Chrome** 27](#_Toc185953238)

[**2.2.2 Tạo project** 27](#_Toc185953239)

[**2.2.3 Record một test case** 28](#_Toc185953240)

[**2.2.4 Playback test case** 31](#_Toc185953241)

[**2.2.5 Lưu project** 32](#_Toc185953242)

[Chương 3. Tiến hành kiểm thử 33](#_Toc185953243)

[3.1 Giới thiệu trang web. 33](#_Toc185953244)

[3.2 Đặc tả use case 34](#_Toc185953245)

[**3.2.1 Mô tả use case Đăng ký** 34](#_Toc185953246)

[**3.2.2 Mô tả use case Tìm kiếm** 35](#_Toc185953247)

[**3.2.3 Mô tả use case Đăng nhập** 36](#_Toc185953248)

[**3.2.4 Mô tả use case Đổi mật khẩu** 37](#_Toc185953249)

[**3.2.5 Mô tả use case Thêm địa chỉ mới** 38](#_Toc185953250)

[3.3 Kiểm thử Website (Công cụ Selenium). 39](#_Toc185953251)

[**3.3.1 Lập kế hoạch kiểm thử (test plan)** 39](#_Toc185953252)

[**3.3.2 Đăng ký** 47](#_Toc185953253)

[**3.3.3 Đăng nhập** 54](#_Toc185953254)

[**3.3.4 Tìm kiếm sản phẩm** 58](#_Toc185953255)

**3.3.5 Đổi mật khẩu** ………………………………………………………………..61

[**3.3.6 Thêm mới địa chỉ** 70](#_Toc185953256)

[PHẦN 3. KIẾN THỨC LĨNH HỘI VÀ BÀI HỌC KINH NGHIỆM 77](#_Toc185953257)

[1. Kết quả đạt được 77](#_Toc185953258)

[2. Hướng phát triển 77](#_Toc185953259)

[TÀI LIỆU THAM KHẢO 78](#_Toc185953260)

# 

# DANH MỤC CÁC THUẬT NGỮ, KÝ HIỆU VÀ CÁC CHỮ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **STT** | **Thuật ngữ** | **Định nghĩa** |
| 1 | https://dienmaycholon.vn/ | Địa chỉ của trang Web Siêu thị điện máy nội thất Chợ Lớn |
| 2 | URL | Cách gọi chung địa chỉ trang Web |
| 3 | Test Designer/ Tester | Người thiết kế kiểm thử/ Kiểm thử viên |

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Từ viết tắt** | **Phiên âm tiếng anh** | **Ý nghĩa** |
| 1 | UC | Use case | Use case |
| 2 | TC | Test case | Trường hợp kiểm thử |
| 3 | GUI | Graphical user interface | Giao diện người dùng |
| 4 | ST | System test | Hệ thống kiểm thử |
| 5 | UAT | User acceptance test | Kiểm tra sự chấp nhận của người dùng |
| 6 | TR | Test report | Báo cáo kiểm thử |

# DANH MỤC HÌNH ẢNH

[*Hình 2. 1: Phân loại công cụ Selenium 21*](#_heading=h.3cqmetx)

[*Hình 2. 2: Tạo project với Selenium 27*](#_heading=h.1jlao46)

[*Hình 2. 3: Đặt tên project 27*](#_heading=h.43ky6rz)

[*Hình 2. 4: Thực hiện record một kịch bản 28*](#_heading=h.xvir7l)

[*Hình 2. 5: Gắn link trang web cần kiểm thử 28*](#_heading=h.3hv69ve)

[*Hình 2. 6: Thực hiện thao tác trên web 29*](#_heading=h.1x0gk37)

[*Hình 2. 7: Thực hiện dừng record 29*](#_heading=h.4h042r0)

[*Hình 2. 8: Tạo test case 30*](#_heading=h.2w5ecyt)

[*Hình 2. 9: Run test case 30*](#_heading=h.3vac5uf)

[*Hình 2. 10: Lưu lại project 31*](#_heading=h.pkwqa1)

[*Hình 3. 1: Giao diện trang chủ của trang web DIENMAYCHOLON.VN 31*](#_heading=h.48pi1tg)

[*Hình 3. 2: Giao diện chức năng đăng nhập 46*](#_heading=h.1d96cc0)

[*Hình 3. 3: Kịch bản kiểm thử bỏ trống họ và tên 48*](#_heading=h.rjefff)

[*Hình 3. 4: Kịch bản kiểm thử bỏ trống email 49*](#_heading=h.3bj1y38)

[*Hình 3. 5: Kịch bản kiểm thử bỏ trống ô nhập username 49*](#_heading=h.1qoc8b1)

[*Hình 3. 6: Kịch bản kiểm thử bỏ trống mật khẩu 49*](#_heading=h.4anzqyu)

[*Hình 3. 7: Kịch bản kiểm thử nhập sai mật khẩu 50*](#_heading=h.2pta16n)

[*Hình 3. 8: Kịch bản kiểm thử bỏ trống ô nhập địa chỉ 50*](#_heading=h.14ykbeg)

[*Hình 3. 9: Tiến hành kiểm thử 51*](#_heading=h.3oy7u29)

[*Hình 3. 10: Giao diện chức năng đăng nhập 51*](#_heading=h.j8sehv)

[*Hình 3. 11: Kịch bản kiểm thử đăng nhập thành công 53*](#_heading=h.42ddq1a)

[*Hình 3. 12: Kịch bản kiểm thử sai mật khẩu 53*](#_heading=h.2hio093)

[*Hình 3. 13: Kịch bản kiểm thử mật khẩu để trống 53*](#_heading=h.wnyagw)

[*Hình 3. 14: Kịch bản kiểm thử đăng nhập không đúng 53*](#_heading=h.3gnlt4p)

[*Hình 3. 15: Kịch bản kiểm thử đăng nhập để trống 54*](#_heading=h.1vsw3ci)

[*Hình 3. 16: Tiến hành kiểm thử 54*](#_heading=h.4fsjm0b)

[*Hình 3. 17: Giao diện chức năng tìm kiếm 55*](#_heading=h.1a346fx)

[*Hình 3. 18: Kịch bản kiểm thử ô tìm kiếm khi dữ liệu trống 56*](#_heading=h.2981zbj)

[*Hình 3. 19: Kịch bản kiểm thử ô tìm kiếm khi nhập dữ liệu hợp lệ 56*](#_heading=h.odc9jc)

[*Hình 3. 20: Kịch bản kiểm thử ô tìm kiếm khi nhập dữ liệu có ký tự đặc biệt 56*](#_heading=h.38czs75)

[*Hình 3. 21: Kịch bản kiểm thử tìm kiếm với từ khóa vô nghĩa hoặc không có trong cơ sở dữ liệu 56*](#_heading=h.1nia2ey)

[*Hình 3. 22: Kịch bản kiểm thử tìm kiếm với từ khóa tiếng Việt viết hoa, thường, có dấu, không dấu xen kẽ 57*](#_heading=h.47hxl2r)

[*Hình 3. 23: Tiến hành kiểm thử 57*](#_heading=h.2mn7vak)

[*Hình 3. 24: Giao diện chức năng đổi mật khẩu 58*](#_heading=h.3ls5o66)

[*Hình 3. 25: Kịch bản kiểm thử nhập sai mật khẩu hiện tại 60*](#_heading=h.302dr9l)

[*Hình 3. 26: Kịch bản kiểm thử bỏ trống ô nhập ô mật khẩu hiện tại 60*](#_heading=h.1f7o1he)

[*Hình 3. 27: Kịch bản kiểm thử bỏ trống ô nhập mật khẩu mới 60*](#_heading=h.3z7bk57)

[*Hình 3. 28: Kịch bản kiểm thử nhập sai xác nhận mật khẩu 61*](#_heading=h.2eclud0)

[*Hình 3. 29: Tiến hành kiểm thử 61*](#_heading=h.thw4kt)

# DANH MỤC BẢNG BIỂU

[*Bảng 1. 1: Tỉ lệ công việc của các giai đoạn phát triển phần mềm 9*](#_heading=h.4f1mdlm)

[*Bảng 3. 1: Các định nghĩa 37*](#_heading=h.2rrrqc1)

[*Bảng 3. 2: Các từ viết tắt 37*](#_heading=h.16x20ju)

[*Bảng 3. 3: Tài liệu tham khảo 38*](#_heading=h.184mhaj)

[*Bảng 3. 4: Lịch trình công việc 38*](#_heading=h.3s49zyc)

[*Bảng 3. 5: Phần cứng máy tính 39*](#_heading=h.meukdy)

[*Bảng 3. 6: Danh sách phần mềm 39*](#_heading=h.36ei31r)

[*Bảng 3. 7: Công cụ kiểm thử 39*](#_heading=h.45jfvxd)

[*Bảng 3. 8: Danh sách nhân sự 40*](#_heading=h.2koq656)

[*Bảng 3. 9: Kiểm thử chức năng 41*](#_heading=h.zu0gcz)

[*Bảng 3. 10: Kiểm thử giao diện 42*](#_heading=h.3jtnz0s)

[*Bảng 3. 11: Bảng rủi ro 43*](#_heading=h.1yyy98l)

[*Bảng 3. 12: Phân loại lỗi 44*](#_heading=h.4iylrwe)

[*Bảng 3. 13: Điều kiện đầu vào chức năng đăng ký 46*](#_heading=h.3x8tuzt)

[*Bảng 3. 14: Danh sách test case chức năng Đăng ký 47*](#_heading=h.2ce457m)

[*Bảng 3. 15: Đầu vào chức năng đăng nhập 52*](#_heading=h.338fx5o)

[*Bảng 3. 16: Danh sách các test case 52*](#_heading=h.1idq7dh)

[*Bảng 3. 17: Điều kiện đầu vào chức năng đăng nhập 55*](#_heading=h.3u2rp3q)

[*Bảng 3. 18: Danh sách các TC 55*](#_heading=h.ye015yiyzs9z)

[*Bảng 3. 19: Điều kiên ràng buộc chức năng đổi mật khẩu 58*](#_heading=h.20xfydz)

[*Bảng 3. 20: Danh sách test case 59*](#_heading=h.4kx3h1s)

# PHẦN I. MỞ ĐẦU

## Tên đề tài

***Tìm hiểu công cụ kiểm thử Selenium và ứng dụng kỹ thuật kiểm thử Website dienmaycholon.vn***

## Lý do chọn đề tài

Hiện nay, công nghệ thông tin càng ngày càng phát triển đi cùng với đó là sự phát triển của phần mềm. Có rất nhiều loại phần mềm, chủ yếu là dành cho các thiết bị như máy tính, điện thoại, máy tính bảng. Chúng phục vụ cho nhiều mục đích khác nhau của đời sống, ví dụ: Phần mềm kế toán, phần mềm vẽ kỹ thuật, phần mềm quản lý công việc, phần mềm quản lý hàng hóa,…. Nhưng cũng từ đó nảy sinh ra nhiều vấn đề về lỗi hỏng hóc phần mềm không đáng có gây ra các ảnh hưởng nghiêm trọng đến xã hội, kinh tế,…Để đảm bảo chất lượng phần mềm trước khi đưa ra sử dụng, tránh các rủi ro cho khách hàng khi đưa phần mềm vào sử dụng, giảm thời gian và chi phí phát sinh do bảo trì (fix lỗi) cho người viết phần mềm thì kiểm thử là một tiến trình không thể thiếu trong quy trình phát triển phần mềm.

Thật vậy, ngày nay càng ngày các chương trình (các phần mềm) càng trở lên phức tạp và đồ sộ. Việc tạo ra một sản phẩm có thể bán được trên thị trường đòi hỏi sự nỗ lực của hàng chục, hàng trăm thậm chí hàng ngàn nhân viên. Số lượng dòng mã lên đến hàng triệu. Và để tạo ra một sản phẩm thì không phải chỉ do một tổ chức đứng ra làm từ đầu đến cuối, mà đòi hỏi sự liên kết, tích hợp của rất nhiều sản phẩm, thư viện lập trình, … của nhiều tổ chức khác nhau… Từ đó đòi hỏi việc kiểm nghiệm phần mềm càng ngày càng trở nên rất quan trọng và rất phức tạp.

Nhằm củng cố kiến thức môn Kiểm thử phần mềm, nhóm chúng em đã nghiên cứu các kĩ thuật kiểm thử phần mềm cụ thể là kỹ thuật kiểm thử Fontend.

## Mục đích đề tài

Đề tài ***“Tìm hiểu công cụ kiểm thử Selenium và ứng dụng kỹ thuật kiểm thử Website dienmaycholon.com”*** nhằm áp dụng những công cụ, kiến thức, kỹ thuật để định nghĩa, lập kế hoạch, tiến hành triển khai kiểm thử trong học phần *Kiểm thử phần mềm* vào việc kiểm thử một website thực tế.

## Mục tiêu đề tài

* + Nắm bắt quy trình kiểm thử
  + Nắm bắt định nghĩa, ưu, nhược điểm của kiểm thử Fontend.
  + Sử dụng công cụ Selenium để tiến hành kiểm thử

## Bố cục đề tài

*Nội dung chính của đề tài gồm 3 chương:*

**Chương 1. Tổng quan về kiểm thử phần mềm:** Giới thiệu về vai trò của kiểm thử.Phân loại kiểm thử.Giới thiệu về kiểm thử frontend, công cụ selenium.

**Chương 2. Tìm hiểu các công cụ kiểm thử:** Tìm hiểu về công cụ sẽ áp dụng, cách cài đặt và sử dụng công cụ.

**Chương 3. Tiến hành kiểm thử**: Áp dụng công cụ đã tìm hiểu để kiểm thử các chức năng đăng ký, đăng nhập, đổi mật khẩu, tìm kiếm sản phẩm.

# PHẦN II. KẾT QUẢ NGHIÊN CỨU

## Chương 1: Tổng quan về kiểm thử phần mềm

## Khái niệm

Kiểm thử phần mềm là hoạt động thực tiễn sản phẩm hay dịch vụ phần mềm trong đúng môi trường chúng dự định sẽ được triển khai nhằm cung cấp cho người có lợi ích liên quan những thông tin về chất lượng của sản phẩm hay dịch vụ phần mềm ấy. Mục đích của kiểm thử phần mềm là tìm ra các lỗi hay khiếm khuyết phần mềm nhằm đảm bảo hiệu quả hoạt động tối ưu của phần mềm trong nhiều ngành khác nhau.

Kiểm thử phần mềm là phương pháp kiểm tra xem sản phẩm phần mềm đó trên thực tế có phù hợp với các yêu cầu đã đặt ra hay không, và đảm bảo rằng không có lỗi hay khiếm khuyết. Nó bao gồm việc kiểm tra, phân tích, quan sát và đánh giá các khía cạnh khác nhau của sản phẩm. Người kiểm thử phần mềm (Tester) sử dụng kết hợp các công cụ thủ công và tự động. Sau khi tiến hành kiểm thử, Tester báo cáo kết quả cho team phát triển. Mục đích là xác định các lỗi, khiếm khuyết hoặc các yêu cầu còn thiếu so với yêu cầu thực tế.

Nội dung của hoạt động kiểm thử bao gồm:

* Kế hoạch kiểm thử
* Phương pháp kiểm thử
* Chiến lược kiểm thử và kỹ thuật sử dụng

## Vai trò của kiểm thử phần mềm

Việc tạo ra một sản phẩm phần mềm phải trải qua nhiều giai đoạn, người ta gọi là quy trình phát triển phần mềm, bắt đầu từ khi có ý tưởng cho đến khi đưa ra sản phẩm phần mềm thực thi. Khối lượng công việc trong từng giai đoạn của quá trình sản xuất phần mềm cũng thay đổi theo thời gian.

*Bảng 1. 1: Tỉ lệ công việc của các giai đoạn phát triển phần mềm*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Giai đoạn | Phân tích yêu cầu | Thiết kế sơ bộ | Thiết kế chi tiết | Lập trình và kiểm thử đơn vị | Tích hợp và kiểm thử tích hợp | Kiểm thử hệ thống |
| Hai thập kỷ 1960-  1970 | 10% | | | 80% | 10% | |
| Thập kỷ 1980 | 20% | 60% | | 20% | | |
| Thập kỷ 1990 | 40% | 30% | | 30% | | |

Như vậy, một sản phẩm phần mềm không chỉ đơn giản là tập hợp các

đoạn mã chương trình. Trong quá trình phát triển phần mềm, ngoài việc lập trình, còn nhiều công đoạn khác quan trọng không kém, và những công đoạn này có thể đóng góp nhiều hơn vào sự thành công của sản phẩm phần mềm. Việc kiểm thử cũng vì thế phải được tiến hành trong tất cả các sản phần tạo nên một sản phẩm phần mềm.

Hiệu quả về chi phí**:** Đây là một trong những lợi ích quan trọng của kiểm thử phần mềm. Thực tế cho thấy rằng các lỗi thiết kế khó có thể được loại trừ hoàn toàn đối với bất kỳ hệ thống nào. Đó không phải là lỗi bất cẩn của Developer mà đôi khi do sự phức tạp của hệ thống. Nếu các vấn đề về thiết kế không được phát hiện, thì việc tìm ra và sửa các lỗi/khiếm khuyết sẽ trở nên khó khăn và tốn kém hơn. Kiểm thử bất kỳ dự án IT nào cũng sẽ giúp công ty tiết kiệm, việc xác định lỗi trong giai đoạn đầu sẽ giúp quá trình sửa chữa tốn ít chi phí hơn.

Bảo mật:Đây là điểm nhạy cảm và dễ bị tấn công nhất của kiểm thử phần mềm. Kiểm thử giúp loại bỏ các rủi ro và vấn đề trong sản phẩm. Cùng với đó, tất cả khách hàng đều đang tìm kiếm những sản phẩm đáng tin cậy.

Chất lượng sản phẩm**:** Đây là yêu cầu thiết yếu của bất kỳ sản phẩm phần mềm nào. Kiểm thử phần mềm giống như việc củng cố danh tiếng công ty bằng cách cung cấp các sản phẩm chất lượng cho khách hàng.

Sự hài lòng của khách hàng:Trong bất kỳ hoạt động kinh doanh sản phẩm nào, mục tiêu cuối cùng đều là mang đến cho khách hàng trải nghiệm tốt nhất. Sự hài lòng của khách hàng rất quan trọng trong quá trình hợp tác lâu dài.

## Phân loại kiểm thử

### **1.3.1 Kiểm thử chức năng (Functional testing)**

Kiểm thử chức năng là xác minh hệ thống hoạt động theo đúng theo các yêu cầu nghiệp vụ. Hình thức kiểm thử này có thể được thực hiện từ hai khía cạnh: dựa trên yêu cầu (requirements-based) và dựa trên quy trình nghiệp vụ (business – process – based).

Trong kiểm thử dựa trên yêu cầu, các yêu cầu được ưu tiên tùy thuộc vào tiêu chí rủi ro. Điều này sẽ đảm bảo những phần quan trọng nhất sẽ được test đầy đủ. Mặt khác, kiểm thử dựa trên quy trình nghiệp vụ sẽ sử dụng những kiến thức tương ứng. Quy trình nghiệp vụ mô tả các việc liên quan đến nghiệp vụ hằng ngày của hệ thống.

Kiểm thử chức năng bao gồm 5 bước:

* Xác định các chức năng mà phần mềm sẽ thực hiện.
* Tạo các dữ liệu đầu vào dựa trên các tài liệu đặc tả kỹ thuật của các chức năng.
* Xác định các kết quả đầu ra dựa trên các tài liệu đặc tả kỹ thuật của các chức năng.
* Thực hiện các trường hợp kiểm thử.
* So sánh kết quả thực tế và kết quả mong muốn.

Trong đó, kiểm thử chức năng còn được chia nhỏ ra thành các loại:

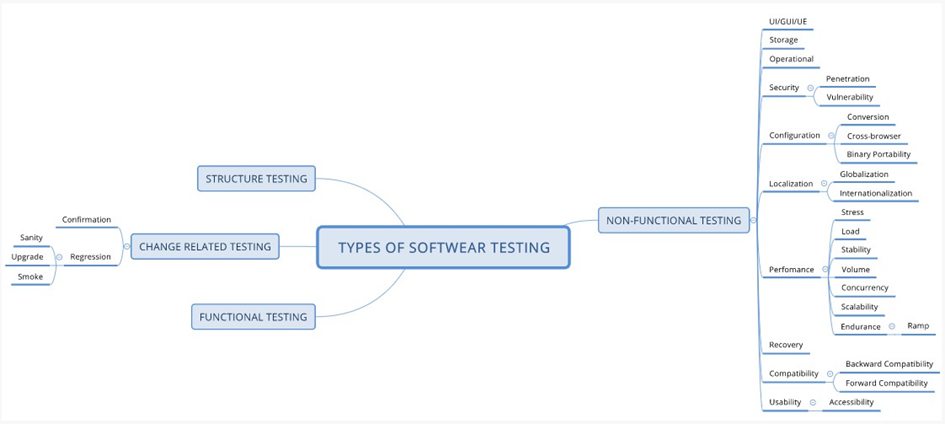
* Kiểm thử đơn vị (Unit testing)
* Smoke Testing
* Sanity Testing
* Kiểm thử giao diện (Interface testing)
* Kiểm thử tích hợp (Integration testing)
* Kiểm thử hệ thống (System testing)
* Kiểm thử hồi quy (Regression testing)
* Kiểm thử chấp nhận (Acceptance testing)

Ưu điểm:

* Hình thức kiểm thử này mô phỏng việc sử dụng hệ thống thực tế
* Được thực hiện trong các điều kiện gần với điều kiện của khách hàng
* Không có giả định nào về cấu trúc hệ thống được đưa ra trong khi kiểm thử chức năng
* Rất dễ dàng để thực hiện test thủ công

Nhược điểm:

* Khả năng cao xảy ra tình trạng test dư thừa
* Các lỗi logic trong phần mềm có thể bị bỏ sót trong khi kiểm thử chức năng



*Hình 1. 1: Kiểm thử phần mềm có nhiều hình thức khác nhau và được phân loại theo một số tiêu chí*

### **1.3.2 Kiểm thử phi chức năng (Non-functional testing)**

Kiểm thử phi chức năng là kiểm tra các đặc tính chất lượng của hệ thống. Ví dụ, kiểm tra xem bao nhiêu người có thể đăng nhập đồng thời vào một phần mềm. Kiểm tra phi chức năng cũng quan trọng không kém như kiểm tra chức năng và ảnh hưởng đến sự hài lòng của khách hàng.

Tương tự, kiểm thử phi chức năng cũng được chia thành các loại:

***Kiểm thử độ ổn định (Stability testing)***: đánh giá phần mềm có thể liên tục hoạt động tốt trong hoặc ngay trên khoảng thời gian có thể chấp nhận hay không.

***Kiểm thử khả năng chịu tải (Load testing)***: đánh giá hoạt động của hệ thống khi khối lượng công việc ngày càng tăng.

***Kiểm thử áp lực (Stress testing)***: ước tính hoạt động của hệ thống ở trong hoặc vượt quá giới hạn khối lượng công việc dự kiến.

***Kiểm thử tính khả dụng (Usability testing)***: sản phẩm được test về tính thân thiện với người dùng.

***Kiểm thử bảo trì (Maintainability testing)***: kiểm tra mức độ đánh giá, thay đổi và test sản phẩm.

***Kiểm thử độ tin cậy (Reliability testing)***: sử dụng công cụ để tìm, ngăn chặn và loại bỏ lỗi trước khi hệ thống được triển khai.

***Kiểm thử tính tương thích (Portability testing)***: xác định mức độ dễ dàng hoặc khó khăn mà phần mềm có thể di chuyển từ môi trường này sang môi trường khác.

### **1.3.3 Kiểm thử cấu trúc (Structural testing)**

Kiểm thử cấu trúc thường được gọi là “hộp trắng” hoặc “hộp thủy tinh” bởi vì phương pháp này quan tâm đến việc tìm kiếm những gì đang xảy ra bên trong, kiểm tra dựa trên phân tích cấu trúc bên trong của thành phần hoặc hệ thống. Nó thường được sử dụng như một cách đo lường của kiểm thử, thông qua độ bao phủ của một tập hợp các yếu tố cấu trúc. Kiểm thử cấu trúc chủ yếu được áp dụng ở kiểm thử thành phần, kiểm thử tích hợp.

Các mục tiêu chính của kiểm thử cấu trúc bao gồm:

* Nhận ra những điểm bất cập.
* Test chức năng bổ sung.
* Xác định những phần bị thiếu trong bộ kiểm thử.

Ưu điểm:

* Loại bỏ code chết.
* Có khả năng tìm ra lỗi ở giai đoạn đầu.
* Đảm bảo kiểm tra phần mềm kỹ lưỡng hơn.
* Tiết kiệm thời gian.

Nhược điểm:

* Kiểm tra kết cấu khá tốn kém.
* Yêu cầu kiến thức về code.
* Đòi hỏi kiến thức vững chắc về công cụ được sử dụng để test.



*Hình 1. 2: Kiểm thử cấu trúc đảm bảo kiểm tra kỹ lưỡng hơn và tiết kiệm thời gian*

### **1.3.4 Kiểm thử liên quan đến các thay đổi (Change related testing)**

***Kiểm thử xác nhận***

Khi kiểm thử gặp lỗi, Tester phải xác định nguyên nhân lỗi là do lỗi phần mềm. Sau khi Tester phát hiện lỗi và báo cho Developer để sửa thì phần mềm sau đó sẽ cập nhật phiên bản vá lỗi. Cuối cùng, Tester cần thực hiện kiểm tra thêm một lần nữa để xác định rằng lỗi thực sự đã được giải quyết.

Khi thực hiện kiểm tra xác nhận, điều quan trọng nhất là phải đảm bảo rằng các trường hợp kiểm thử phải được thực hiện chính xác giống như lần đầu tiên, sử dụng cùng một đầu vào, dữ liệu và môi trường kiểm thử để đảm bảo rằng các lỗi đã được sửa. Tester cần phải biết rằng trong lần kiểm thử sau khi vá lỗi khả năng sinh ra lỗi khác trong phần mềm là điều hoàn toàn có thể xảy ra. Vì vậy kiểm thử chính xác ở phiên bản hiện tại của phần mềm là chưa đủ. Cách phát hiện các điểm ngoài ý muốn của việc kiểm lỗi là thực hiện kiểm thử hồi quy.

***Kiểm thử hồi quy (Regression testing)***

Tương tự như kiểm thử xác nhận thì kiểm thử hồi quy liên quan đến việc lặp lại các trường hợp kiểm thử đã được thực hiện trước đó. Kiểm thử hồi quy được thực hiện khi phần mềm thay đổi do sửa lỗi, chức năng mới.

Mục đích của kiểm thử hồi quy để xác minh rằng các sửa đổi trong phần mềm hoặc môi trường không gây ra bất lợi ngoài ý muốn, ảnh hưởng hoặc làm hư các chức năng và hệ thống vẫn đáp ứng các yêu cầu của phần mềm. Tất cả các trường hợp trong quá trình kiểm thử hồi quy sẽ được thực hiện mỗi khi một phiên bản vá lỗi của phần mềm được release, và điều này khiến chúng trở nên lý tưởng cho tự động hóa.

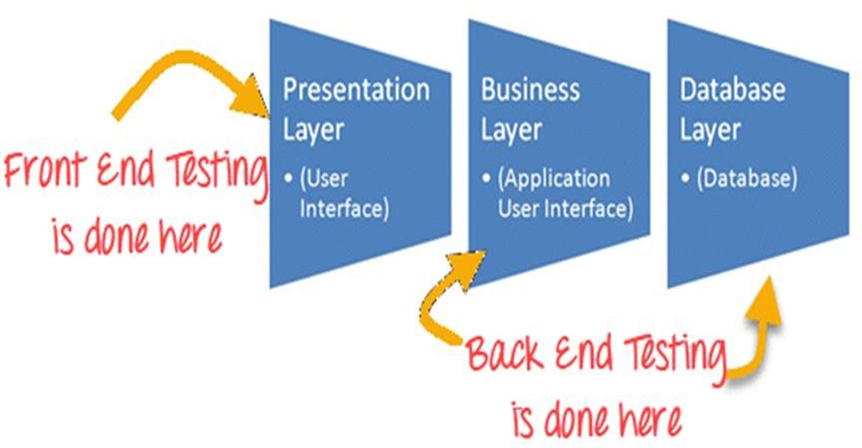
### **1.3.5 Giới thiệu về kiểm thử Frontend**

#### 1.3.5.1 Khái niệm

Kiểm thử Front-end là kiểm thử giao diện đồ họa người dùng (GUI), tính năng và khả năng sử dụng của trang web hoặc ứng dụng.

Ví dụ: Nếu chúng ta nhập tên vào phần đầu của ứng dụng, số không được chấp nhận. Một ví dụ khác là kiểm tra sự liên kết của các yếu tố GUI.

Frontend Testing là một kiểm thử Presentation layer của Kiến trúc 3 lớp. Tức là, chúng ta đang kiểm tra GUI - mọi thứ hiển thị trên màn hình, phía client. Đối với một ứng dụng web, kiểm thử front-end sẽ bao gồm kiểm tra các chức năng như biểu mẫu, biểu đồ, menu, báo cáo, v.v. cũng như Javascript liên quan. Frontend tests là một thuật ngữ bao gồm nhiều chiến lược thử nghiệm khác nhau. Tester cần có sự hiểu biết tốt, có kinh nghiệm về các yêu cầu nghiệp vụ để thực hiện dạng test này.



*Hình 1. 3: Tìm hiểu về kiểm thử Frontend*

#### 1.3.5.2 Mục đích chính của kiểm thử Frontend

Mục đích chính của kiểm thử Frontend là đảm bảo lối vào không còn lỗi với các bản cập nhật liên tiếp. Ngoài ra việc kiểm thử Frontend này được tiến hành cho:

- Kiểm thử hồi quy CSS: Các thay đổi CSS nhỏ có thể làm phá vỡ bố cục.

- Thay đổi đối với tệp JS làm cho giao diện người dùng không hoạt động

- Kiểm thử Performance.

#### 1.3.5.3 Những thách thức của kiểm thử Frontend

Giao diện người dùng luôn phát triển: Trong không gian phần mềm hiện đại, các thư viện lõi và các thành phần của bên thứ ba phải được nâng cấp vài tháng một lần. Nâng cấp một thư viện yêu cầu những thay đổi tương xứng đối với tất cả các thành phần cần thiết khác. Với mỗi lần nâng cấp, tất cả các thành phần cần được kiểm tra lại, bao gồm cả các công cụ tự động hóa và công cụ kiểm tra. Các API và chức năng mới nhất cũng phải được tích hợp, xây dựng và xử lý trong các mốc thời gian ngày càng ngắn.

Liên tục thay đổi theo sở thích của người dùng: Với các thiết bị, trình duyệt và phiên bản hệ điều hành mới được giới thiệu vài tháng một lần, nhu cầu và sở thích của người dùng liên tục thay đổi. Ví dụ: đại dịch gây ra sự gia tăng bùng nổ nhu cầu, mong muốn của người dùng đối với hội nghị truyền hình và phát trực tuyến. Tất nhiên, mỗi khi các tính năng mới hơn được thêm vào, các bài kiểm tra mới cần được tạo và thực thi. Người ta cũng phải tiếp tục thử nghiệm các khía cạnh hiện có như tốc độ tải trang web (các tính năng mới có làm chậm trang không?) Hoặc sự hấp dẫn trực quan (nút mới có che hết menu hiện có không?) Về bản chất, các nhà phát triển và người thử nghiệm làm việc trên một phần mềm không bao giờ thực sự kết thúc.

Chọn công cụ tự động hóa phù hợp: Kiểm tra hiệu quả, định kỳ front-end đòi hỏi phải tự động hóa. Người kiểm tra thủ công không thể tiếp tục chạy kiểm tra mỗi khi nâng cấp được đẩy. Tuy nhiên, việc chọn một công cụ tự động hóa có thể được thiết lập hiệu quả và được cấp quyền với các tập lệnh thử nghiệm để chạy các kiểm tra và xác minh cần thiết. Tuy nhiên, với rất nhiều công cụ kiểm tra tự động hóa có sẵn, có thể hơi khó khăn để chọn những gì sẽ hoạt động tốt nhất cho nhóm của chúng ta, dựa trên các bộ kỹ năng và yêu cầu dự án của họ. Nó sẽ đòi hỏi sự nghiên cứu chuyên dụng, đây sẽ là một sự đầu tư về thời gian và công sức.

Phát hiện các vấn đề về trình duyệt và thiết bị chéo: Với hàng nghìn phiên bản trình duyệt và thiết bị được sử dụng trên khắp thế giới để truy cập internet, người thử nghiệm phải bao gồm một phạm vi lớn để trang bị một trang web hoặc ứng dụng cho việc sử dụng trong thế giới thực. Điều này có thể là một thách thức vì các thiết bị và phiên bản trình duyệt mới liên tục được phát hành. Để theo kịp, các nhóm cần có quyền truy cập vào các trình duyệt và thiết bị thực. Phòng thí nghiệm thiết bị nội bộ cần nguồn nhân lực và tài chính đáng kể để thiết lập, bảo trì và nâng cấp. Nó chỉ dễ dàng hơn khi sử dụng hạ tầng thử nghiệm được lưu trữ trên đám mây, như trong BrowserStack hoặc LambdaTest.

### **1.3.6 Giới thiệu công cụ kiểm thử Selenium**

Selenium là một công cụ kiểm thử phần mềm tự động, được phát triển bởi ThoughtWorks từ năm 2004 với tên ban đầu là JavaScript Test Runner. Đến năm 2007, tác giả Jason Huggins rời ThoughtWorks và gia nhập Selenium team, một phần của Google và phát triển thành Selenium như hiện nay. Selenium là một công cụ rất phổ biến và mạnh mẽ trong việc kiểm thử frontend (giao diện người dùng) của các ứng dụng web. Dưới đây là một số lý do vì sao nên sử dụng Selenium trong kiểm thử frontend:

***Tự động hóa các tác vụ kiểm* thử**: Selenium cho phép tự động hóa các tác vụ kiểm thử lặp đi lặp lại, giúp tiết kiệm thời gian và công sức so với việc kiểm thử thủ công.Tự động hóa giúp tích hợp kiểm thử vào quy trình CI/CD, đảm bảo rằng các thay đổi mới được kiểm thử ngay lập tức.

***Hỗ trợ đa trình duyệt***: Selenium hỗ trợ kiểm thử trên nhiều trình duyệt như Chrome, Firefox, Safari, Edge và Internet Explorer, đảm bảo tính tương thích và hoạt động đúng đắn trên các môi trường khác nhau.

***Hỗ trợ đa nền tảng***: Selenium có thể chạy trên các hệ điều hành khác nhau như Windows, macOS và Linux, giúp dễ dàng tích hợp vào nhiều môi trường phát triển và kiểm thử khác nhau.

***Ngôn ngữ lập trình linh hoạt***: Selenium hỗ trợ nhiều ngôn ngữ lập trình như Java, C#, Python, Ruby, JavaScript, và PHP, cho phép các đội ngũ phát triển lựa chọn ngôn ngữ mà họ thành thạo.

***Cộng đồng lớn và tài liệu phong phú***: Selenium có một cộng đồng lớn và hoạt động mạnh mẽ, cung cấp nhiều tài liệu, diễn đàn, và tài nguyên hỗ trợ, giúp giải quyết các vấn đề gặp phải trong quá trình sử dụng.

***Thư viện và plugin phong phú***: Có nhiều thư viện và plugin mở rộng giúp tăng cường chức năng của Selenium, như Selenium Grid để chạy kiểm thử song song, hoặc WebDriver để tương tác với các thành phần của trang web.

***Tích hợp tốt với các công cụ khác***: Selenium dễ dàng tích hợp với các framework kiểm thử như TestNG, JUnit, và NUnit, giúp quản lý và thực thi các kịch bản kiểm thử một cách hiệu quả.

***Báo cáo kiểm thử chi tiết***: Khi kết hợp với các công cụ như Allure hoặc ExtentReports, Selenium có thể cung cấp các báo cáo kiểm thử chi tiết và dễ hiểu.

***Kiểm thử mô phỏng hành vi người dùng***: Selenium có thể mô phỏng các hành vi của người dùng như nhấp chuột, điền biểu mẫu, cuộn trang, và điều hướng, giúp kiểm thử toàn diện hơn về trải nghiệm người dùng.

Selenium là một công cụ mạnh mẽ và linh hoạt giúp tự động hóa quy trình kiểm thử frontend, đảm bảo ứng dụng web hoạt động ổn định và đúng đắn trên nhiều môi trường và trình duyệt khác nhau. Việc sử dụng Selenium không chỉ giúp tiết kiệm thời gian và công sức mà còn nâng cao chất lượng sản phẩm phần mềm. Chính vì vậy, nhóm chúng em chọn công cụ Selenium cho đề tài này.

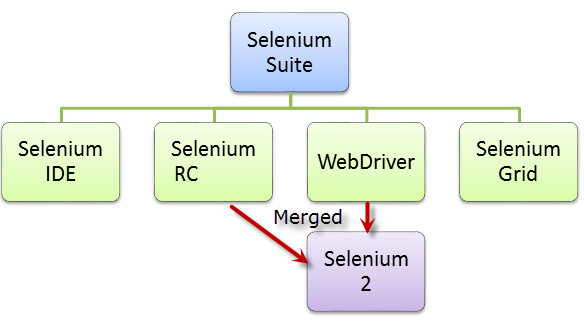
# Chương 2: Tìm hiểu công cụ Selenium

## 2.1 Tìm hiểu công cụ kiểm thử Selenium

### **2.1.1 Selenium là gì**

Selenium là bộ kiểm thử tự động miễn phí (mã nguồn mở) dành cho các ứng dụng web trên các trình duyệt và nền tảng khác nhau. Nó khá là giống với HP Quick Test Pro (QTP bây giờ là UFT) chỉ khác là Selenium thì tập trung vào việc tự động hoá các ứng dụng dựa trên nền tảng web. Kiểm thử được thực hiện bằng cách sử dụng công cụ Selenium thường được gọi là Kiểm thử Selenium. Selenium không chỉ là 1 công cụ độc lập mà là 1 bộ công cụ của phần mềm, mỗi bộ đều đáp ứng được nhu cầu kiểm thử khác nhau của 1 tổ chức. Nó có 4 thành phần:

* [Selenium IDE](http://vntesters.com/selenium-ide-record-and-playback/)
* Selenium RC (Selenium 1 – Selenium Remote Control)
* [Selenium Grid](http://vntesters.com/kiem-thu-tu-dong-selenium-grid/)
* [Selenium WebDriver](http://vntesters.com/gioi-thieu-selenium-webdriver/) (Selenium 2)



*Hình 2. 1: Phân loại công cụ Selenium*

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

Selenium IDE (Integrated Development Environment) là một công cụ cho phép chúng ta Record/Playback một test script. Đây là một add-on hỗ trợ cho FireFox. Chúng ta chỉ có thể Record trên trình duyệt FireFox, nhưng bù lại, chúng ta có thể Playback trên các trình duyệt khác như là IE, Chrome….

Selenium Grid là một hệ thống hỗ trợ người dùng thực thi test script trên nhiều trình duyệt một cách song song mà không cần phải chỉnh sửa test script.

Selenium RC, Selenium WebDriver là một thư viện cho phép chúng ta lập trình (scripting) test script trên các ngôn ngữ lập trình khác nhau như Python, Java, C#, Ruby.

### **2.1.2 Các đặc điểm của Selenium**

***Mã nguồn mở***: là điểm mạnh nhất của Selenium khi so sánh với các test tool khác. 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.

***Cộng đồng hỗ trợ***: Vì là mã nguồn mở nên Selenium có một cộng đồng hỗ trợ khá mạnh mẽ. Bên cạnh đó, Google là nơi phát triển Selenium nên chúng ta hoàn toàn có thể yên tâm về sự hỗ trợ miễn phí khi có vấn đề về Selenium. Tuy nhiên, đây cũng là một điểm yếu của Selenium. Cơ bản vì là hàng miễn phí, cộng đồng lại đông nên một vấn đề có thể nhiều giải pháp, và có thể một số giải pháp là không hữu ích. Mặc khác, chúng ta không thể hối thúc hay ra deadline cho sự hỗ trợ.

***Selenium hỗ trợ nhiều ngôn ngữ lập trình.***

***Selenium hỗ trợ chạy trên nhiều OS khác nhau***: với mức độ chỉnh sửa script hầu như là không có. Thực sự thì điều này phụ thuộc phần lớn vào khả năng viết script của chúng ta.

***Chạy test case ở background***: Khi chúng ta thực thi một test script, chúng ta hoàn toàn có thể làm việc khác trên cùng một PC. Điều này hỗ trợ chúng ta không cần tốn quá nhiều tài nguyên máy móc khi chạy test script.

***Không hỗ trợ Win app***: Selenium thực sự chỉ hỗ trợ chúng ta tương tác với Browser mà không hỗ trợ chúng ta làm việc với các Win app, kể cả Win dialog như Download/Upload – ngoại trừ Browser Alarm. Vậy nên, để xử lý các trường hợp cần tương tác với hệ thống hay một app thứ ba, chúng ta cần một hay nhiều thư viện khác như AutoIt hay Coded UI.

### **2.1.3 Selenium IDE**

Selenium Integrated Development Environment (IDE) là framework đơn giản nhất trong bộ Selenium và nó là phần dễ nhất để học. Nó là một Browser plugin mà bạn có thể cài đặt dễ dàng như các plugin khác. Tuy nhiên, vì sự đơn giản của nó, nên chỉ sử dụng công cụ IDE làm công cụ tạo mẫu. Nếu bạn muốn các trường hợp kiểm thử nâng cao thì bạn có thể sử dụng Katalon Studio thay thế tool hoặc là Selenium WebDriver kết hợp code với ngôn ngữ lập trình.  
 ***Ưu điểm*:**

* Quá trình cài đặt và sử dụng đều dễ dàng.
* Không yêu cầu người sử dụng phải có kỹ năng lập trình mà họ chỉ cần hiểu biết một chút về HTML và DOM là đã có thể sử dụng.
* Có thể thực hiện export cho các test đã được tạo ra để sử dụng bên trong Selenium Webdriver hoặc Selenium RC.
* Bạn sẽ được cung cấp các chức năng để có thể thực hiện report kết quả hoặc những hỗ trợ cần thiết khi sử dụng.
* Sử dụng được tích hợp cùng với các extension khác nhau.

***Nhược điểm*:**

* Bởi nó là một Extension nên bạn chỉ có thể cài đặt nó trên trình duyệt.
* Nó được dùng để thiết kế nhằm tạo ra các test đơn giản hoặc prototype test.
* Với IDE thì bạn không có khả năng thực hiện được các tính toán cũng như câu lệnh phức tạp hoặc có điều kiện.
* Hiệu năng hoạt động có thể sẽ chậm hơn nhiều so với Webdriver và Selenium RC.

### **2.1.4 Selenium Remote Control (Selenium RC)**

Selenium RC là framework kiểm thử hàng đầu của dự án Selenium trong một thời gian dài. Đây là công cụ kiểm tra web tự động đầu tiên cho phép người dung sử dụng ngôn ngữ lập trình mà họ thích. Kể từ phiên bản 2.25.0, RC có thể hỗ trợ các ngôn ngữ lập trình sau:

* Java
* C#
* PHP
* Python
* Perl
* Ruby

**Ưu điểm:**

* Chạy trên các trình duyệt và các hệ điều hành khác nhau.
* Hỗ trợ các hoạt động lặp lại và điều kiện.
* Hỗ trợ các trình duyệt mới.
* Thực thi nhanh hơn Selenium IDE.
* Có thể hỗ trợ data-driven testing.
* Có API hoàn thiện.

**Nhược điểm:**

* Cài đặt phức tạp hơn Selenium IDE.
* Yêu cầu Selenium RC Server để thực run.
* Sự tương tác trình duyệt ít thực tế hơn.
* Kết quả không phù hợp và sử dụng JavaScript.
* Thời gian thực thi chậm hơn WebDriver.
* Phải có kiến thức về lập trình.

### **2.1.5 Selenium WebDriver**

WebDriver chứng minh rằng nó tốt hơn cả Selenium IDE và Selenium RC ở nhiều khía cạnh. Nó thực hiện 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.

**Ưu điểm:**

* Cài đặt đơn giản hơn Selenium RC
* Có thể thực hiện giao tiếp trực tiếp với trình duyệt.
* Tương tác với trình duyệt tương tự như thao tác của một người dùng thật.
* Tốc độ thực thi nhanh hơn so với Selenium IDE và RC
* Kiểm tra được hầu hết các component trên web. Ngoại trừ một số cái không nên làm auto An sẽ chia sẻ sau trong quá trình học.

**Nhược điểm:**

* Quá trình cài đặt phức tạp hơn so với Selenium IDE
* Đòi hỏi người dùng cần phải có các kỹ năng lập trình

### **2.1.6 Selenium Grid**

Selenium Grid là một công cụ được sử dụng cùng với Selenium RC để chạy thử nghiệm 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 thử nghiệm cùng một lúc.

***Đặc điểm:***

* Cho phép chạy đồng thời các thử nghiệm trong nhiều trình duyệt và môi trường.
* Tiết kiệm thời gian rất nhiều.
* Sử dụng khái niệm hub-and-nodes. Hub hoạt động như một nguồn trung tâm của các lệnh Selenium cho mỗi nút được kết nối với nó.

Đây là cách ta xây dựng một Selenium hub để khởi chạy nhiều hơn các test thông qua các máy cũng như trình duyệt khác nhau trong cùng một thời điểm. Selenium Grid cho phép bạn có thể chạy các trường hợp kiểm tra song song có nghĩa là các trường hợp kiểm tra khác nhau sẽ có thể được chạy tại một thời điểm ở trên các máy từ xa khác nhau.

***Điều này sẽ mang lại nhiều lợi thế như:***

Nếu như bạn có bộ dữ liệu kiểm tra đủ lớn hoặc có bộ dữ liệu kiểm tra chạy chậm thì bạn có thể tăng hiệu suất của nó một cách đáng kể bằng cách sử dụng Selenium Grid. Nó sẽ phân chia các trường hợp kiểm tra để chạy những trường hợp kiểm tra khác nhau tại cùng một thời điểm ở trên nhiều máy khác nhau. Như vậy, thì trong nhiều trường hợp bạn sẽ có thể hỗ trợ từ xa nhau và thực hiện chúng trong cùng một thời điểm.

Ở một số trường hợp thì Selenium Grid có thể cải thiện được thời gian mà nó chạy cho các trường hợp song song. Nếu như bạn không có nhiều kinh nghiệm với các chương trình hoặc ngôn ngữ kịch bạn thì bạn đều có thể sử dụng Selenium Grid để có thể làm quen với câu lệnh Selenium.

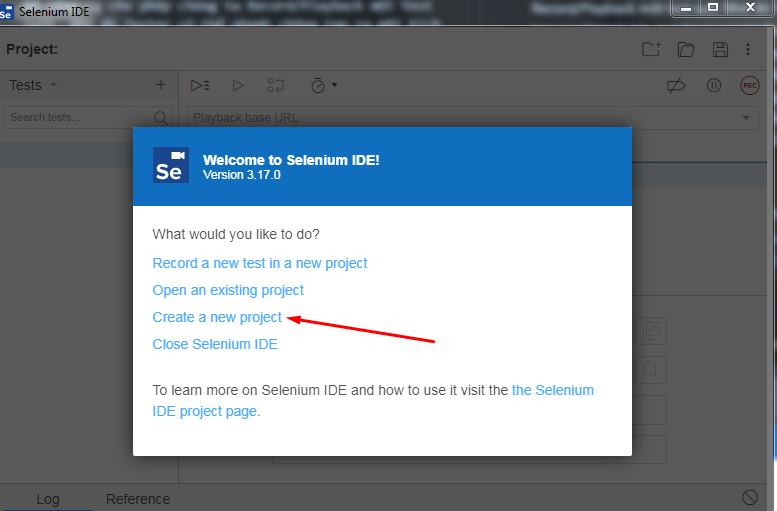
## 2.2 Cài đặt và sử dụng công cụ Selenium

### **2.2.1 Cài đặt Selenium IDE trên Chrome**

Truy cập vào: <https://chrome.google.com/webstore/detail/selenium-ide/mooikfkahbdckldjjndioackbalphokd> để cài dặt “Selenium IDE” trên trình duyệt Chrome.

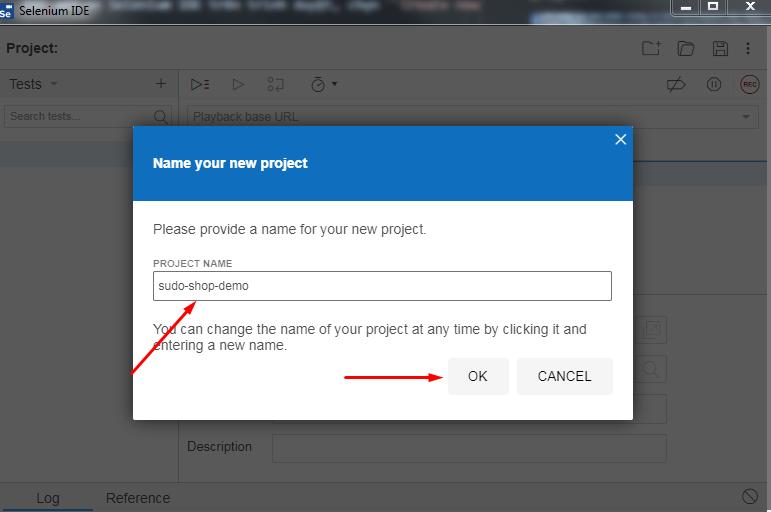
### **2.2.2 Tạo project**

Bật add-on Selenium IDE trên trình duyệt, chọn Create new project.



*Hình 2. 2: Tạo project với Selenium*

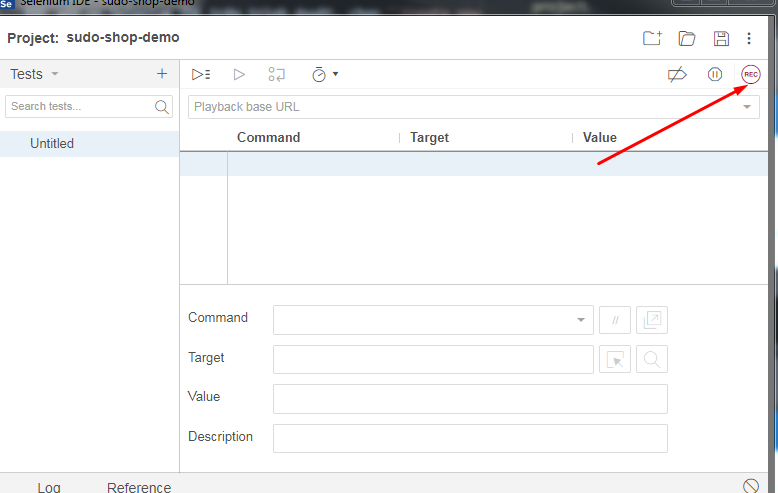
Nhập tên project và nhấn OK



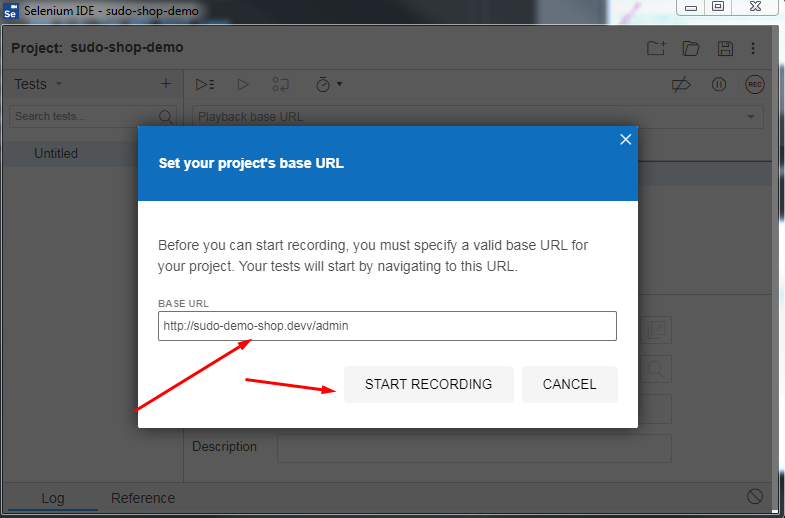
*Hình 2. 3: Đặt tên project*

### **2.2.3 Record một test case**

Tại cửa sổ project, Bấm nút REC ở góc phải và nhập tên miền sẽ test và ấn nút START RECORD để tiến hành ghi lại Test case

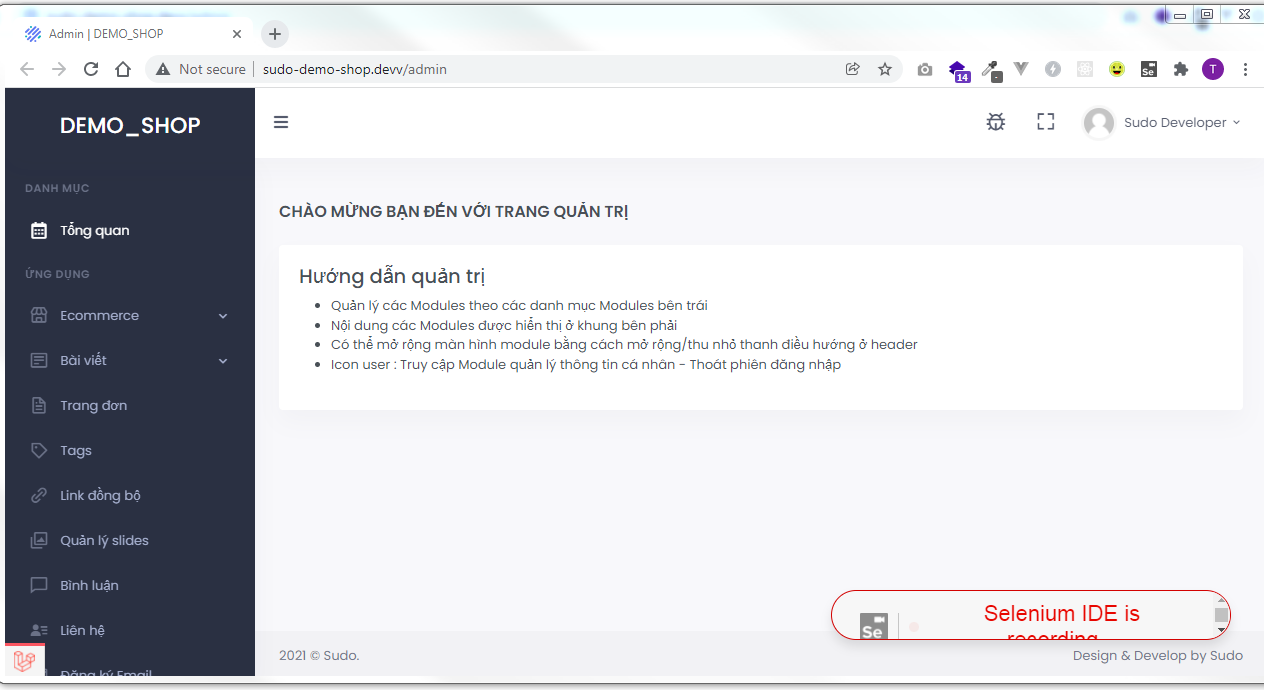


*Hình 2. 4: Thực hiện record một kịch bản*



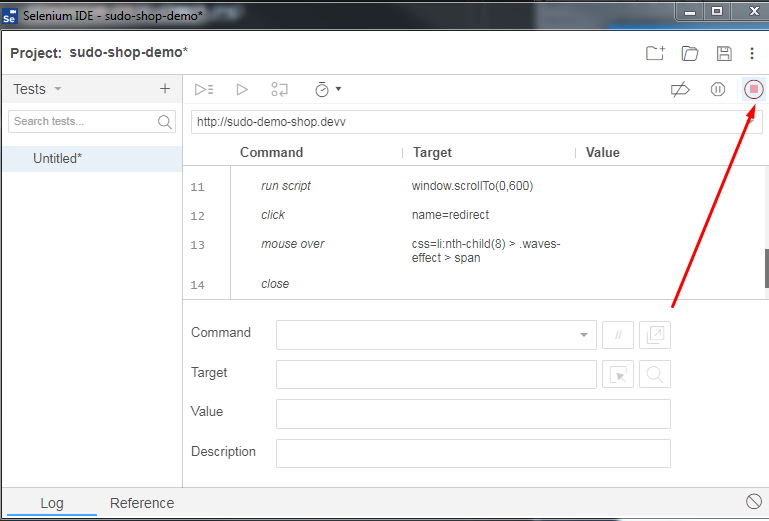
*Hình 2. 5: Gắn link trang web cần kiểm thử*

Cửa sổ mới hiện ra với tên miền đã chọn, bạn tiến hành các thao tác, Selenium IDE sẽ ghi lại các thao tác click, nhập,... thành các command



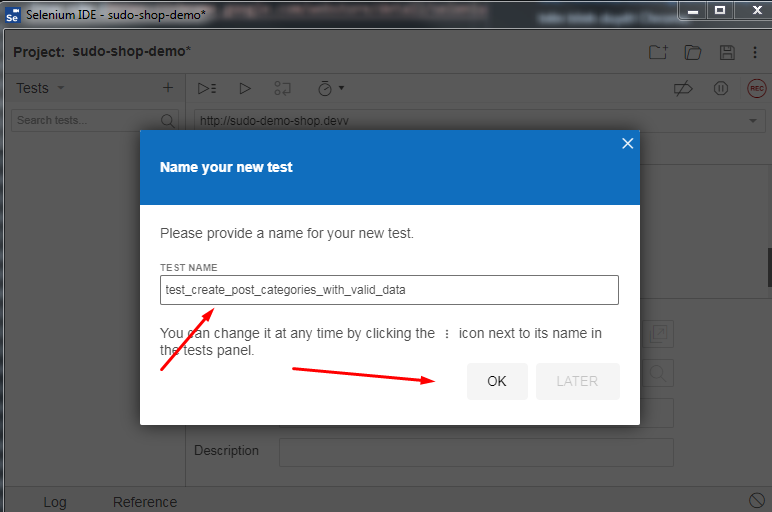
*Hình 2. 6: Thực hiện thao tác trên web*

Sau khi thực hiện xong, đóng cửa sổ test, quay lại cửa số Selenium IDE và ấn nút có icon Stop.



*Hình 2. 7: Thực hiện dừng record*

Nhập tên Test case, chú ý tên phải rõ ràng hoặc là mã Test Case. Nhấn OK



*Hình 2. 8: Tạo test case*

### **2.2.4 Playback test case**

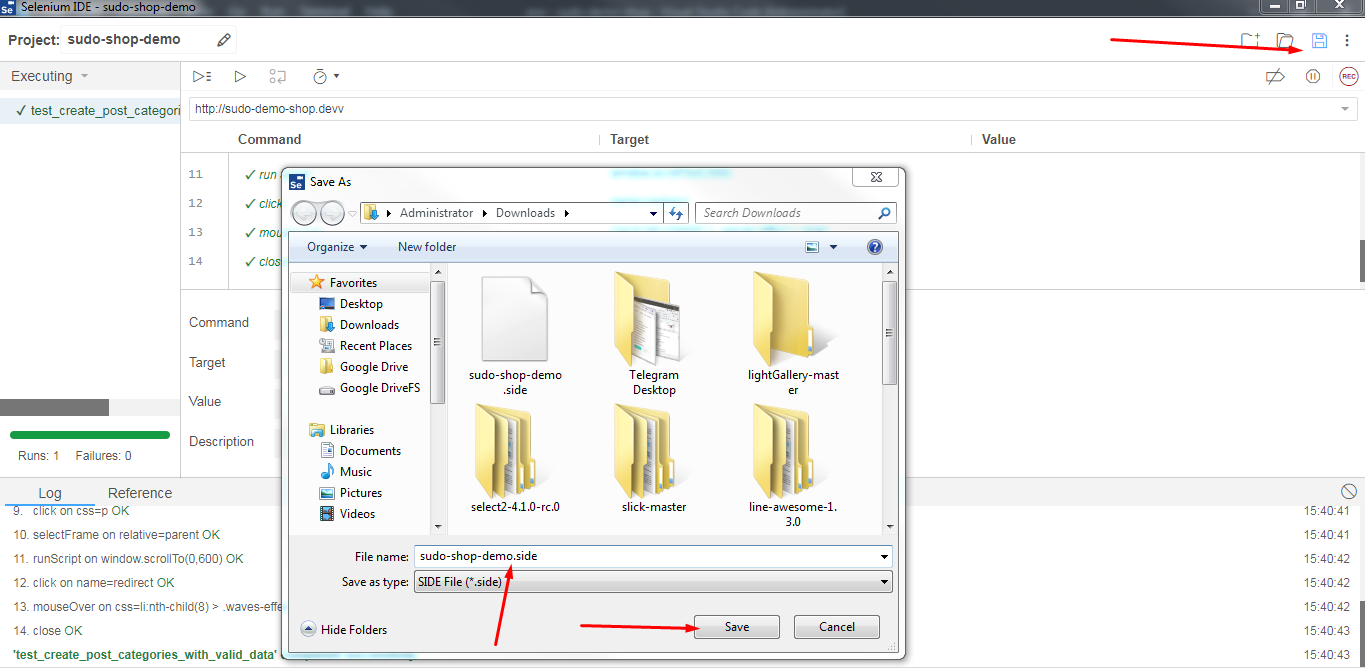
Chọn test case và ấn vào icon Run all test, Selenium IDE sẽ thực hiện lại các thao tác theo những command đã record.



*Hình 2. 9: Run test case*

### **2.2.5 Lưu project**

Nhấn vào icon Save project trên góc phải. Nhập tên và lưu.



*Hình 2. 10: Lưu lại project*

# Chương 3. Tiến hành kiểm thử

## 3.1 Giới thiệu trang web.

Địa chỉ trang web: https://dienmaycholon.vn/

Thông tin về trang web DIENMAYCHOLON.VN

Đây là một trang bán hàng online được phát triển với mục đích cung cấp một nền tảng mua sắm trực tuyến về các mặt hàng gia dụng. Được xây dựng và phát triển để cung cấp cho người dùng một trải nghiệm mua sắm trực tuyến thuận tiện và đa dạng.

Trang web DIENMAYCHOLON.VN cung cấp một bộ sưu tập đa dạng các loại hàng gia dụng như: tivi, tủ lạnh, máy giặt, máy sấy, điện thoại, máy lọc nước..., các mặt hàng này thuộc vào các hãng sản xuất nổi tiếng trong và ngoài nước như SamSung, Sunhouse, Kangaroo, Panasonic,... Người dùng có thể dễ dàng tìm kiếm và chọn lựa sản phẩm theo giá thành, mẫu mã, màu sắc, kích cỡ và hãng sản xuất phù hợp với nhu cầu của mình.



*Hình 3. 1: Giao diện trang chủ của trang web DIENMAYCHOLON.VN*

## 3.2 Đặc tả use case

### **3.2.1 Mô tả use case Đăng ký**

1. Tên use case: Đăng ký

2. Mô tả vắn tắt: Use case này cho phép người dùng đăng ký tài khoản hệ thống.

3. Luồng các sự kiện:

3.1. Luồng cơ bản:

1. Use case này bắt đầu khi khách hàng kích vào nút “Đăng ký” trên hệ thống. Hệ thống sẽ hiển thị màn hình đăng ký.
2. Khách hàng điền Tên đăng nhập, họ và tên, email, mật khẩu, địa chỉ và số điện thoại, ngày sinh sau đó kích vào nút “Đăng ký”. Hệ thống sẽ lưu thông tin vừa nhập vào bảng KHACHHANG và TAIKHOAN và hiển thị 1 thông báo đăng ký thành công.

Use case kết thúc.

3.2. Luồng rẽ nhánh:

1. Tại bước 2 của luồng cơ bản nếu khách hàng nhập thiếu thông tin thì hệ thống sẽ hiển thị thông báo “Chưa điền đủ thông tin”. Khách hàng thực hiện lại bước 2 hoặc use case kết thúc.
2. Tại bước 2 của luồng cơ bản nếu khách hàng nhập Email không đúng định dạng thì hệ thống sẽ hiện ra một dòng thông báo “Email trống hoặc sai định dạng Email”. Người dùng thực hiện lại bước 2 của use case hoặc use case kết thúc.
3. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sử dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

4. Yêu cầu đặc biệt:

Không có.

5. Tiền điều kiện:

Không có.

6. Hậu điều kiện:

Không có.

7. Các điểm mở rộng:

Không có.

### **3.2.2 Mô tả use case Tìm kiếm**

1. Tên Use Case: Tìm Kiếm

2. Mô tả vắn tắt: Use case này cho phép Khách Hàng tìm kiếm sản phẩm.

3. Luồng các sự kiện

3.1. Luồng cơ bản

1. Use case này bắt đầu khi khách hàng nhập tên sản phẩm và kích vào nút “Tìm kiếm” trên menu. Hệ thống sẽ lấy ra danh sách thông tin các sản phẩm trùng khớp gồm: tiêu đề, đơn giá, ảnh sản phẩm, số lượng từ bảng SANPHAM, và hiển thị ra màn hình.

Use case kết thúc.

3.2. Các luồng rẽ nhánh

1. Trong luồng cơ bản nếu trong bảng SANPHAM không có dữ liệu sản phẩm cần tìm thì hệ thống sẽ hiển thị thông báo: “Tìm thấy **0** kết quả phù hợp”. Use case kết thúc.
2. Tại bất kỳ thời điểm nào trong quá trình thực hiện Use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ thông báo lỗi và Use case kết thúc.

4. Các yêu cầu đặc biệt:

Không có.

5. Tiền điều kiện:

Không Có

6. Hậu điều kiện:

Không Có

7. Điểm mở rộng:

Không có.

### **3.2.3 Mô tả use case Đăng nhập**

1. Tên use case: Đăng nhập

2. Mô tả vắn tắt:

Use case cho phép khách hàng đăng nhập vào hệ thống dựa trên quyền truy cập.

3. Luồng các sự kiện:

3.1. Luồng cơ bản

* 1. Use case này bắt đầu khi khách hàng chọn “Đăng nhập” trên hệ thống. Hệ thống sẽ hiển thị form Đăng nhập.
  2. Khách hàng nhập tên tài khoản và mật khẩu rồi kích nút “Đăng nhập”. Hệ thống sẽ kiểm tra dữ liệu về tên tài khoản và mật khẩu mà khách hàng nhập ở bảng TAIKHOAN và sau đó sẽ hiển thị giao diện màn hình tương ứng với quyền truy cập của tài khoản.

Use case kết thúc.

3.2. Luồng rẽ nhánh

* 1. Tại bước 2 của luồng cơ bản, nếu để trống trường tên đăng nhập hệ thống sẽ hiển thị thông báo “Vui lòng nhập tên đăng nhập của bạn”. Người dùng sẽ phải thực hiện lại bước 2 của luồng cơ bản hoặc use case kết thúc.
  2. Tại bước 2 của luồng cơ bản, nếu để trống trường mật khẩu thì hệ thống sẽ hiển thị thông báo “ Vui lòng nhập mật khẩu của bạn”. Người dùng sẽ phải thực hiện lại bước 2 của luồng cơ bản hoặc use case kết thúc.
  3. Tại bước 2 của luồng cơ bản, nếu người dùng nhập sai tên đăng nhập hoặc mật khẩu thì hệ thống sẽ hiển thị thông báo “Tên đăng nhập hay mật khẩu không chính xác.” và người dùng sẽ phải thực hiện lại bước 2 của luồng cơ bản hoặc use case sẽ kết thúc.
  4. Tại bất kì bước nào trong luồng cơ bản, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

4. Các yêu cầu đặc biệt:

Không có.

5. Tiền điều kiện:

Không có.

6. Hậu điều kiện:

Hệ thống sẽ hiển thị menu chức năng tương ứng với quyền người dùng.

7. Điểm mở rộng:

Không có.

### **3.2.4 Mô tả use case Đổi mật khẩu**

1. Tên use case: Đổi mật khẩu

2. Mô tả vắn tắt:

Use case cho phép khách hàng đổi mật khẩu tài khoản.

3. Luồng các sự kiện:

3.1. Luồng cơ bản

1. Use case này bắt đầu khi khách hàng chọn “Đổi mật khẩu” trên hệ thống. Hệ thống sẽ hiển thị form Đổi mật khẩu.
2. Khách hàng nhập mật khẩu hiện tại, mật khẩu mới và xác nhận mật khẩu rồi kích nút “Xác nhận”. Hệ thống sẽ kiểm tra dữ liệu tài khoản của khách hàng ở bảng TAIKHOAN và sau đó sẽ cập nhật lại mật khẩu.

Use case kết thúc.

3.2. Luồng rẽ nhánh

* 1. Tại bước 2 của luồng cơ bản, nếu nhập sai trường mật khẩu hiện tại hệ thống sẽ hiển thị thông báo “Mật khẩu cũ không chính xác, Vui lòng nhập lại”. Người dùng sẽ phải thực hiện lại bước 2 của luồng cơ bản hoặc use case kết thúc.
  2. Tại bước 2 của luồng cơ bản, nếu để trống trường mật khẩu hiện tại, mật khẩu mới và xác nhận mật khẩu thì hệ thống sẽ hiển thị thông báo “ Vui lòng nhập mật khẩu của bạn”. Người dùng sẽ phải thực hiện lại bước 2 của luồng cơ bản hoặc use case kết thúc.
  3. Tại bước 2 của luồng cơ bản, nếu người dùng nhập xác nhận mật khẩu không trùng với mật khẩu mới thì hệ thống sẽ hiển thị thông báo “Nhập lại mật khẩu không chính xác” và người dùng sẽ phải thực hiện lại bước 2 của luồng cơ bản hoặc use case sẽ kết thúc.
  4. Tại bất kì bước nào trong luồng cơ bản, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

4. Các yêu cầu đặc biệt:

Không có.

5. Tiền điều kiện:

Khách hàng phải đăng nhập vào hệ thống.

6. Hậu điều kiện:

Không có.

7. Điểm mở rộng:

Không có.

### **3.2.5 Mô tả use case Thêm địa chỉ mới**

1. Tên use case: Thêm mới địa chỉ

2. Mô tả vắn tắt:

Use case cho phép khách hàng thêm mới địa chỉ.

3. Luồng các sự kiện:

3.1. Luồng cơ bản

1. Use case này bắt đầu khi khách hàng chọn “THÊM ĐỊA CHỈ MỚI” trên hệ thống. Hệ thống sẽ hiển thị form Thêm mới địa chỉ.
2. Khách hàng nhập Họ tên, Số điện thoại, Địa chỉ, Tỉnh/Thành phố rồi kích nút “Thêm địa chỉ”. Hệ thống sẽ tạo 1 bảng địa chỉ mới với những thông tin vừa nhập trong bảng TAIKHOAN.

Use case kết thúc.

3.2. Luồng rẽ nhánh

* 1. Tại bước 2 của luồng cơ bản, nếu bỏ trống ô Họ tên hệ thống sẽ hiển thị thông báo “Vui lòng nhập tên của bạn”. Người dùng sẽ phải thực hiện lại bước 2 của luồng cơ bản hoặc use case kết thúc.
  2. Tại bước 2 của luồng cơ bản, nếu bỏ trống ô Số điện thoại hệ thống sẽ hiển thị thông báo “Vui lòng nhập số điện thoại”. Người dùng sẽ phải thực hiện lại bước 2 của luồng cơ bản hoặc use case kết thúc.
  3. Tại bước 2 của luồng cơ bản, nếu bỏ trống ô Địa chỉ hệ thống sẽ hiển thị thông báo “Vui lòng nhập địa chỉ”. Người dùng sẽ phải thực hiện lại bước 2 của luồng cơ bản hoặc use case kết thúc.
  4. Tại bất kì bước nào trong luồng cơ bản, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

4. Các yêu cầu đặc biệt:

Không có.

5. Tiền điều kiện:

Khách hàng phải đăng nhập vào hệ thống.

6. Hậu điều kiện:

Không có.

7. Điểm mở rộng:

Không có.

## 3.3 Kiểm thử Website (Công cụ Selenium).

### **3.3.1 Lập kế hoạch kiểm thử (test plan)**

#### 3.3.1.1 Giới thiệu

**Mục đích**

* Xác định thông tin cơ bản về dự án và các thành phần chức năng được kiểm thử và không được kiểm thử.
* Liệt kê những yêu cầu cho việc kiểm thử (Test Requirements).
* Những chiến lược kiểm thử nên được sử dụng.
* Ước lượng những yêu cầu về tài nguyên và chi phí cho việc kiểm thử.
* Những tài liệu được lập sau khi hoàn thành việc kiểm thử.

**Phạm vi kiểm thử**

Kiểm thử hệ thống: tiến hành thực hiện trên một hệ thống hoàn thành và hội nhập để đánh giá điểm tương thích của hệ thống với các yêu cầu cụ thể.

` Các kiểu kiểm thử có trong kế hoạch:

* + Kiểm thử chức năng: kiểm thử các chức năng của hệ thống website bán phụ kiện thủy sinh thỏa mãn yêu cầu khách hàng.
  + Kiểm thử giao diện người dùng: giao diện đúng như yêu cầu đặc tả khách hàng cung cấp.

**Người sử dụng tài liệu**

Tài liệu này được sử dụng cho Test Manager, Test Designer và Tester.

#### 3.3.1.2 Tài liệu tham khảo

*Bảng 3. 1: Tài liệu tham khảo*

| **Stt** | **Tên tài liệu** | **Tác giả** | **Ngày duyệt** |
| --- | --- | --- | --- |
| 1 | Tài liệu mô tả yêu cầu hệ thống | Nguyễn Đặng Hải Quân | 15/12/2024 |
| 2 | Mẫu kế hoạch kiểm thử | Lê Thế Sơn | 15/12/2024 |
| 3 | Mẫu test case | Nguyễn Thanh Tùng | 15/12/2024 |
| 4 | Mẫu test report | Cù Minh Hải | 15/12/2024 |
| 5 | Mẫu test website | Giáp Văn Duy | 15/12/2024 |

#### 

#### 3.3.1.3 Lịch trình công việc

*Bảng 3. 2: Lịch trình công việc*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mốc công việc** | **Sản phẩm** | **Thời gian** | **Bắt đầu** | **Kết thúc** |
| Lập kế hoạch kiểm thử | Test plan | 1 ngày | 08/06/2024 | 09/12/2024 |
| Xem lại các tài liệu | Test plan | 2 ngày | 09/06/2024 | 11/12/2024 |
| Thiết kế các test case | Test case | 1 ngày | 11/06/2024 | 12/12/2024 |
| Viết các test case | Test case | 2 ngày | 12/06/2024 | 13/12/2024 |
| Xem lại các testcase | Test case | 2 ngày | 13/06/2024 | 14/12/2024 |
| Thực thi các testcase | Test case | 1 ngày | 14/06/2024 | 14/12/2024 |
| Ghi nhận và đánh giá kết quả kiểm thử | Test report | 2 ngày | 15/06/2024 | 16/12/2024 |

#### 3.3.1.4 Những yêu cầu về tài nguyên

**Phần cứng:** Máy tính cá nhân có kết nối mạng Internet.

*Bảng 3. 3: Phần cứng máy tính*

|  |  |  |  |
| --- | --- | --- | --- |
| **CPU** | **RAM** | **SSD** | **Architecture** |
| CPU Intel Core i3 trở lên | 4GB trở lên | 256 GB trở lên | 32/ 64 bit |

**Phần mềm**

*Bảng 3. 4: Danh sách phần mềm*

|  |  |  |
| --- | --- | --- |
| **Tên phần mềm** | **Phiên bản** | **Loại** |
| Microsoft Edge | 120.0.2210.61 | Trình duyệt web |
| Mozilla Firefox | 12.0 | Trình duyệt web |
| Google Chrome | 19.0.1084.56 | Trình duyệt web |
| Microsoft Windows 7 Professional | Final | Hệ điều hành |

**Công cụ kiểm thử**

*Bảng 3. 5: Công cụ kiểm thử*

|  |  |  |  |
| --- | --- | --- | --- |
| **Hoạt động** | **Công cụ** | **Nhà cung cấp** | **Phiên bản** |
| Quản lý Test Case | Microsoft Office Excel | Microsoft | 2013 |
| Quản lý phiên bản | Microsoft Office Word | Microsoft | 2013 |
| Theo dõi lỗi | Microsoft Office Excel | Microsoft | 2013 |
| Thực thi kiểm thử | Selenium IDE | seleniumhq.org | 3.17.2 |

**Môi trường kiểm thử**

* Máy tính cá nhân có kết nối mạng Internet để có thể truy cập vào trang web bằng trình duyệt.
* Các chức năng của trang web được kiểm tra trên các trình duyệt Microsoft Edge, Mozilla Firefox, Google Chrome, Internet Explorer.
* Hệ điều hành được sử dụng là Microsoft Windows 10 Professional.

**Nhân sự**

*Bảng 3. 6: Danh sách nhân sự*

|  |  |
| --- | --- |
| **Thành viên** | **Vai trò** |
| Giáp Văn Duy | Test Manager / Test Designer / Tester: Lập kế hoạch kiểm thử, quản lý tiến độ hoạt động kiểm thử, thiết kế test case cho chức năng Đăng nhập |
| Lê Thế Sơn | Test Designer / Tester: Thiết kế và viết các test case, thực thi các test case cho chức năng Đăng ký. |
| Nguyễn Đặng Hải Quân | Test Designer / Tester: Thiết kế và viết các test case, thực thi các test case cho chức năng Tìm kiếm sản phẩm. |
| Nguyễn Thanh Tùng | Test Designer / Tester: Thiết kế và viết các test case, thực thi các test case cho chức năng Đổi mật khẩu. |
| Cù Minh Hải | Test Designer / Tester: Thiết kế và viết các test case, thực thi các test case cho chức năng Thêm mới địa chỉ. |

#### 3.3.1.5 Phạm vi kiểm thử

**Những chức năng được kiểm thử**

* Đăng ký: Kiểm tra chức năng tạo tài khoản của hệ thống.
* Đăng nhập: Kiểm tra chức năng đăng nhập của hệ thống.
* Tìm kiếm sản phẩm: Kiểm tra chức năng tìm kiếm muốn mua.
* Đổi mật khẩu: Kiểm tra chức năng đổi mật khẩu của người dùng.

**Những giao diện được kiểm thử**

* Đăng ký
* Đăng nhập
* Tìm kiếm sản phẩm
* Đổi mật khẩu
* Thêm mới địa chỉ

#### 3.3.1.6 Chiến lược kiểm thử

**Các giai đoạn kiểm thử**

* Dùng kiểu kiểm thử tự động (automatic test) bao gồm kiểm thử GUI và từng chức năng.
* Việc kiểm thử chỉ bắt đầu khi đã hoàn thiện bộ test case để kiểm thử GUI và chức năng.
* Thiết kế test case theo phương pháp phân vùng tương đương.
* Chỉ thực hiện kiểm thử hồi quy, không thực hiện kiểm thử lại.
* Các yêu cầu phi chức năng khác: tải trọng, hiệu năng…không được kiểm thử.

**Các loại kiểm thử**

***Kiểm thử chức năng***

*Bảng 3. 7: Kiểm thử chức năng*

|  |  |
| --- | --- |
| Mục đích kiểm tra | Đảm bảo các chức năng được kiểm tra hoạt động chính xác theo đặc tả yêu cầu |
| Kỹ thuật | Thực thi tất cả các trường hợp có thể có cho mỗi nhóm chức năng, sử dụng dữ liệu hợp lệ và không hợp lệ để xác định:  - Kết quả mong đợi khi dữ liệu hợp lệ được sử dụng  - Cảnh báo phù hợp hiện ra khi dữ liệu không hợp lệ được sử dụng |
| Tiêu chuẩn dừng | Tất cả các test case đã được thiết kế đều được thực thi.  Tất cả các lỗi tìm thấy đều được ghi nhận lý do rõ ràng để có thể giúp cho lập trình viên khắc phục. |
| Chịu trách nhiệm kiểm thử | Test Designer / Tester |
| Cách kiểm thử | Sử dụng công cụ và phần mềm để thực hiện các bài kiểm tra, kiểm tra chức năng hoặc kiểm tra tính năng của phần mềm một cách tự động. |
| Xử lý ngoại lệ | Liệt kê tất cả các vấn đề liên quan phát sinh trong quá trình thực thi kiểm thử. |

***Kiểm thử giao diện***

*Bảng 3. 8: Kiểm thử giao diện*

|  |  |
| --- | --- |
| Mục đích kiểm tra | Đảm bảo các giao diện được hiển thị đúng với thiết kế về kiểu hiển thị, tính hợp lệ về dữ liệu, tính thích ứng khi chạy trên các trình duyệt khác nhau |
| Kỹ thuật | Thực thi tất cả các trường hợp có thể cho các trình duyệt khác nhau, sử dụng dữ liệu hợp lệ và không hợp lệ để xác định:  - Kết quả mong đợi khi dữ liệu hợp lệ được sử dụng.  - Cảnh báo phù hợp hiện ra khi dữ liệu không hợp lệ được sử dụng. |
| Tiêu chuẩn dừng | Tất cả các test case đã được thiết kế đều được thực thi.  Tất cả các lỗi tìm thấy đều được ghi nhận lý do rõ ràng để có thể giúp cho lập trình viên khắc phục. |
| Chịu trách nhiệm kiểm thử | Test Designer / Tester |
| Cách kiểm thử | Sử dụng công cụ và phần mềm để thực hiện các bài kiểm tra, kiểm tra chức năng hoặc kiểm tra tính năng của phần mềm một cách tự động. |
| Xử lý ngoại lệ | Liệt kê tất cả các vấn đề liên quan phát sinh trong quá trình thực thi kiểm thử. |

#### 3.3.1.7 Đánh giá rủi ro

*Bảng 3. 9: Bảng rủi ro*

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Rủi ro** | **Cách khắc phục** | **Mức độ rủi ro** |
| 1 | Thay đổi yêu cầu làm ảnh hưởng đến nguồn nhân lực và chiến lược test. | Lập lại plan sao cho phù hợp với lịch trình thực tế khi thay đổi yêu cầu, có thể chọn cách tăng thêm nguồn nhân lực cho dự án, hoặc tăng thời gian làm việc ngoài giờ cho nhân viên. | Cao |
| 2 | Sản phẩm mà lập trình viên thực hiện không kịp theo thời gian như lịch trình đề ra. | Yêu cầu cập nhật tiến độ công việc thường xuyên để quản lý kịp thời các thay đổi về thời gian và kỹ thuật. | Cao |
| 3 | Thiếu nguồn nhân lực do có thành viên đột ngột xin nghỉ. | Bổ sung thêm nguồn nhân lực mới hoặc tăng giờ làm việc ngoài giờ. | Thấp |

#### 3.3.1.8 Điều kiện chấp nhận kiểm thử

Tỉ lệ test case đạt( passed): 100%

Tỉ lệ test case không đạt (failed): 0%

Hệ thống chạy ổn định trên các trình duyệt web khác nhau (Mozilla Firefox, Google Chrome, Microsoft Edge, Internet Explorer).

#### 3.3.1.9 Theo dõi lỗi

**Phân loại lỗi**

*Bảng 3. 10: Phân loại lỗi*

|  |  |  |
| --- | --- | --- |
| **Mức độ nghiêm trọng** | | **Đặc tả lỗi** |
| Cao | * Không thể đăng nhập * Không thể đăng ký * Không đổi được mật khẩu * Không tìm kiếm được hoặc tìm kiếm sai sản phẩm | |
| Trung bình | * Đăng nhập sai không báo lỗi, không trả về kết quả * Lưu dữ liệu vào csdl nhưng không hiển thị lại lên màn hình khi thêm, sửa, xóa sản phẩm | |
| Thấp | * Hiển thị kết quả tìm kiếm nhưng không đúng mẫu * Xuất hiện các lỗi chính tả trên màn hình * Các chức năng chạy không ổn định trên một số trình duyệt | |

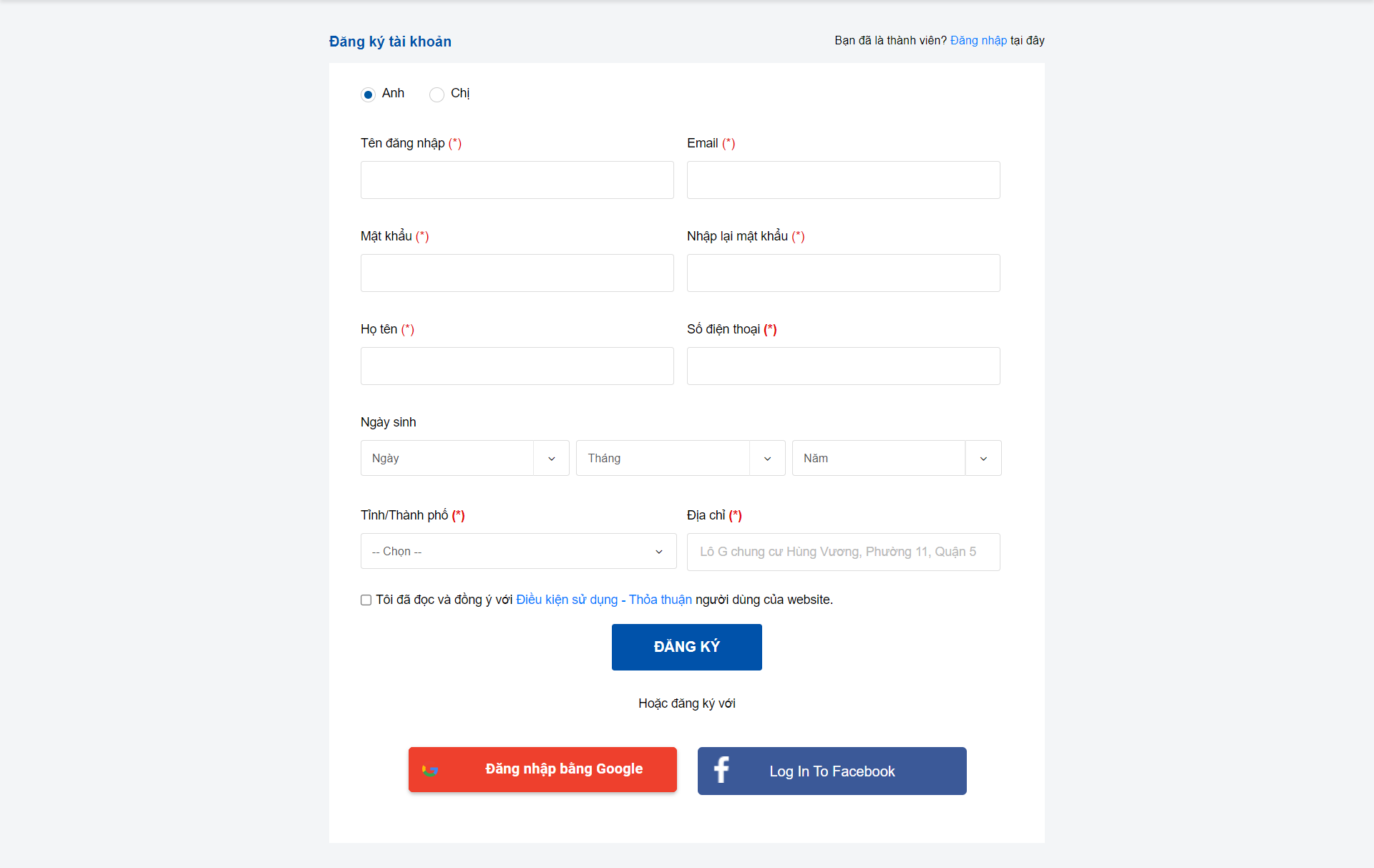
**Quy trình xử lý lỗi**

* Ghi nhận lại các lỗi được tìm thấy trong quá trình kiểm thử.
* Xác nhận lại lỗi
* Sửa lỗi
* Kiểm thử hồi quy.
* Cập nhật trạng thái lỗi.

### **3.3.2 Đăng ký**

**3.3.2.1. Phân tích thiết kế kiểm thử.**

**Giao diện chức năng:**

****

*Hình 3. 2: Giao diện chức năng đăng nhập*

**Mô tả giao diện**

Đăng ký bao gồm Tên đăng nhập, Mật khẩu, Nhập lại mật khẩu, Họ tên, Giới tính, Email, Số điện thoại, Ngày sinh, Tỉnh/thành phố, Địa chỉ.

**Các điều kiện ràng buộc:**

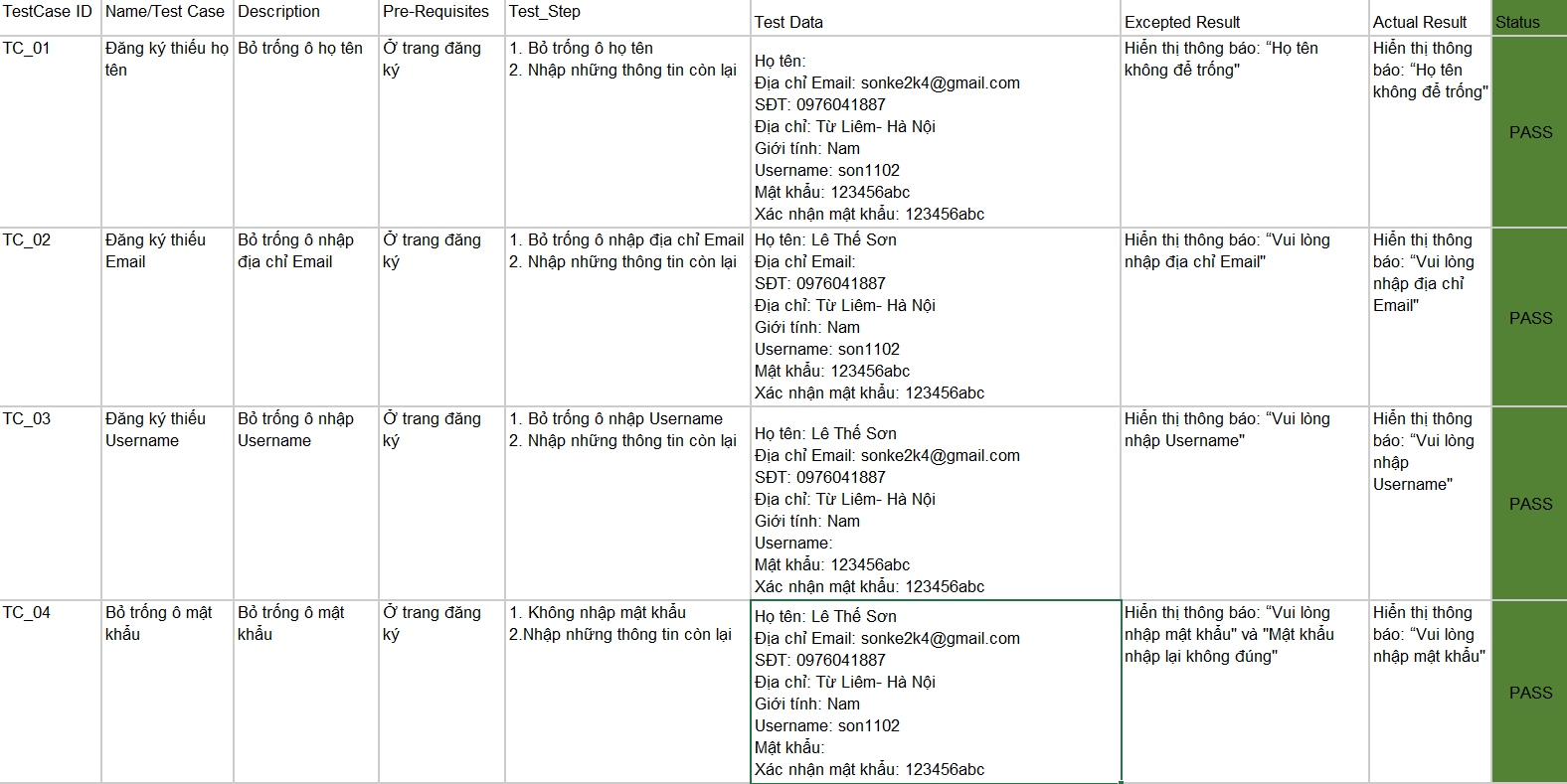
Điều kiện ràng buộc: Tên đăng nhập, Mật khẩu, Nhập lại mật khẩu, Họ tên, Email, Số điện thoại, Tỉnh/thành phố, Địa chỉ không được để trống, tối đa 255 ký tự và số điện thoại là số

*Bảng 3. 11: Điều kiện đầu vào chức năng đăng ký*

|  |  |  |
| --- | --- | --- |
| Điều kiện đầu vào | Các lớp tương đương hợp lệ | Các lớp tương đương không hợp lệ |
| Tên đăng nhập | > 0 và <= 255 | Để trống  > 255 |
| Mật khẩu | > 0 và <= 255 | Để trống  > 255 |
| Nhập lại mật khẩu | > 0 và <= 255 | Để trống  > 255 |
| Họ tên | > 0 và <= 255 | Để trống  > 255 |
| Email | > 0 và <= 255  Nhập đúng định dạng | Để trống  Nhập sai định dạng |
| Số điện thoại | > 0 và <= 10  Nhập đúng định dạng | Để trống  Nhập sai định dạng |
| Địa chỉ | > 0 và <= 255 | Để trống |

Danh sách test case:

*Bảng 3. 12: Danh sách test case chức năng Đăng ký*



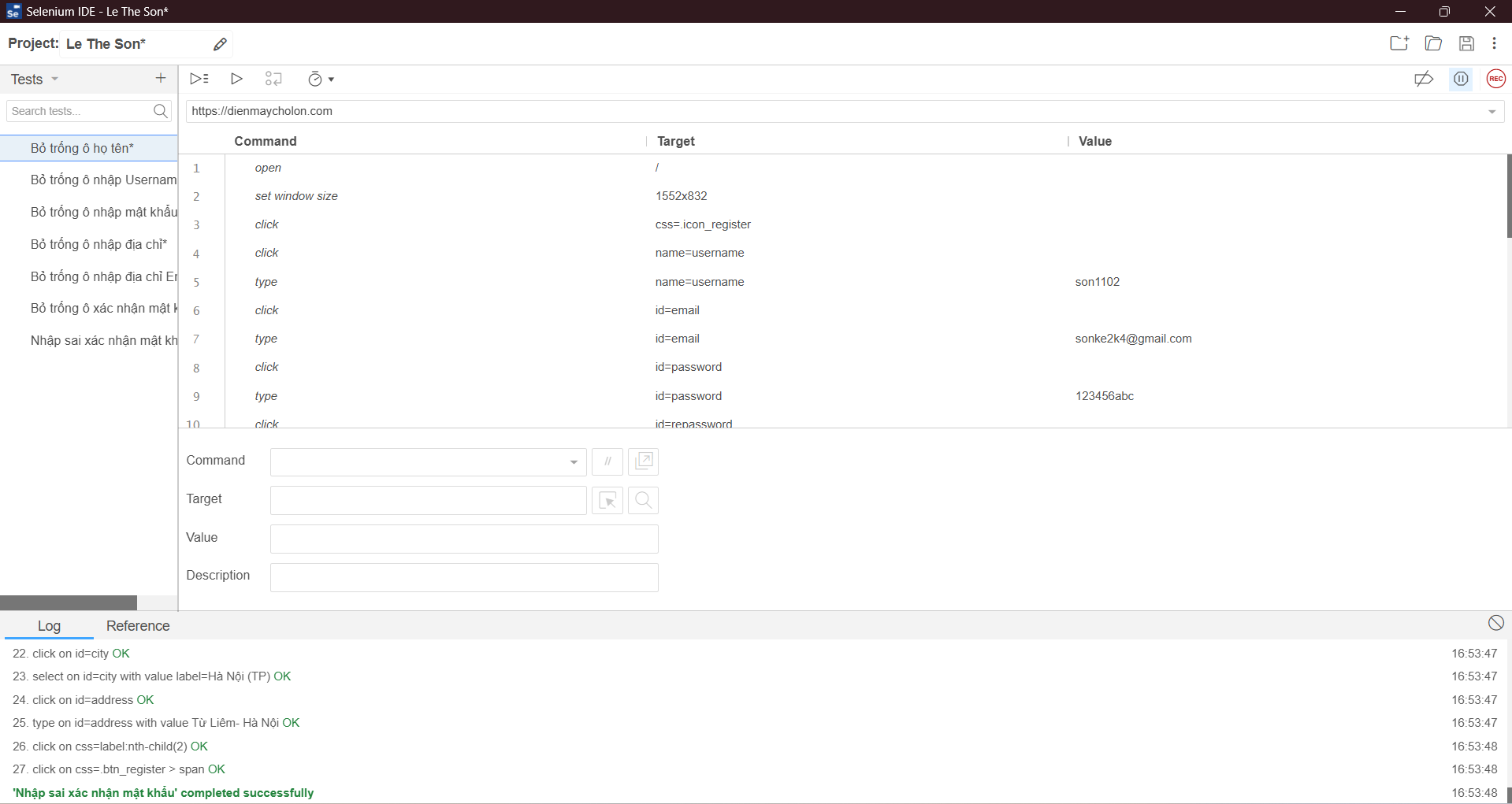


**3.3.2.2 Thực hiện kiểm thử.**

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

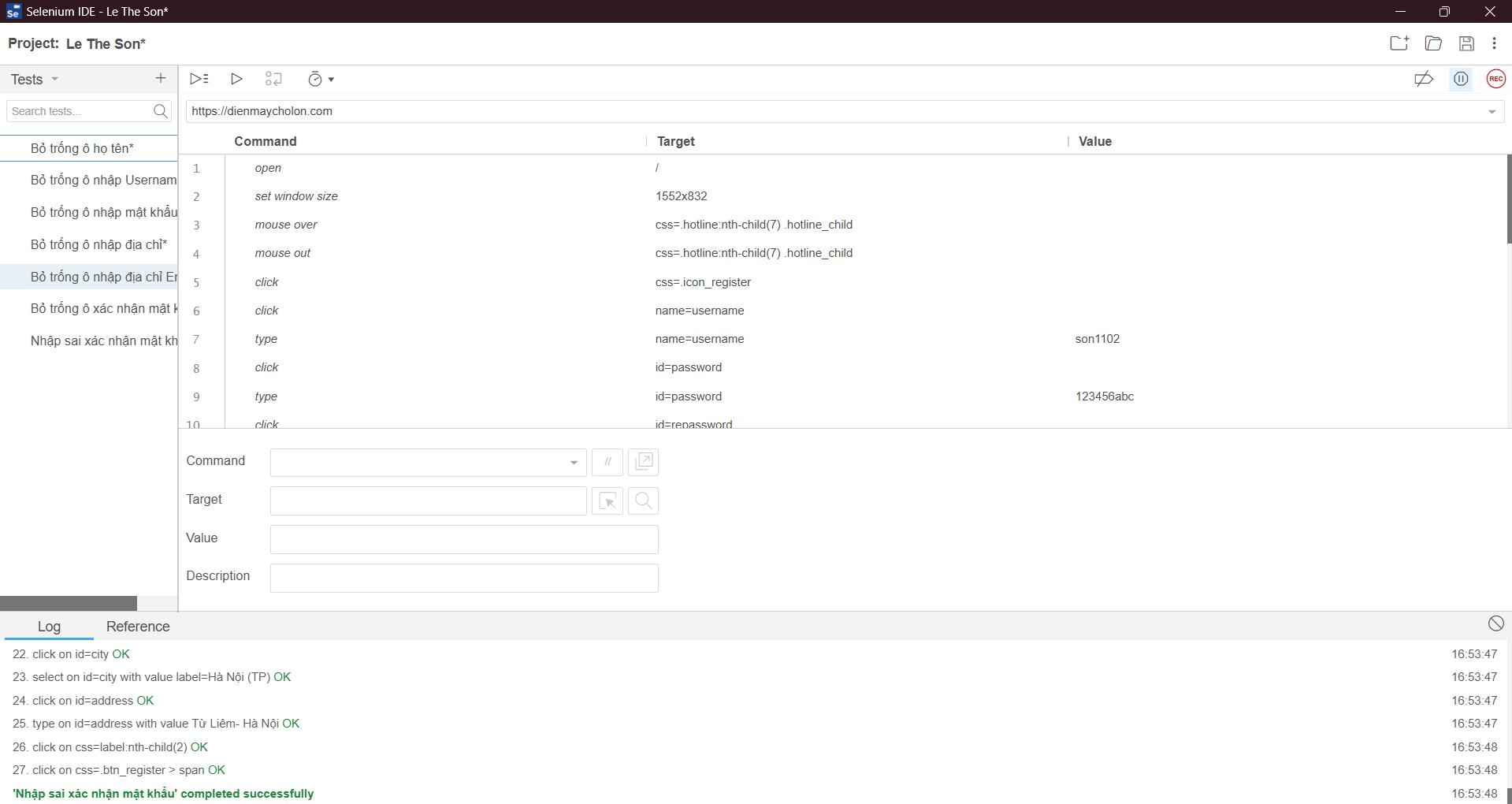
Ta có kịch bản kiểm thử:

* Bỏ trống họ tên:



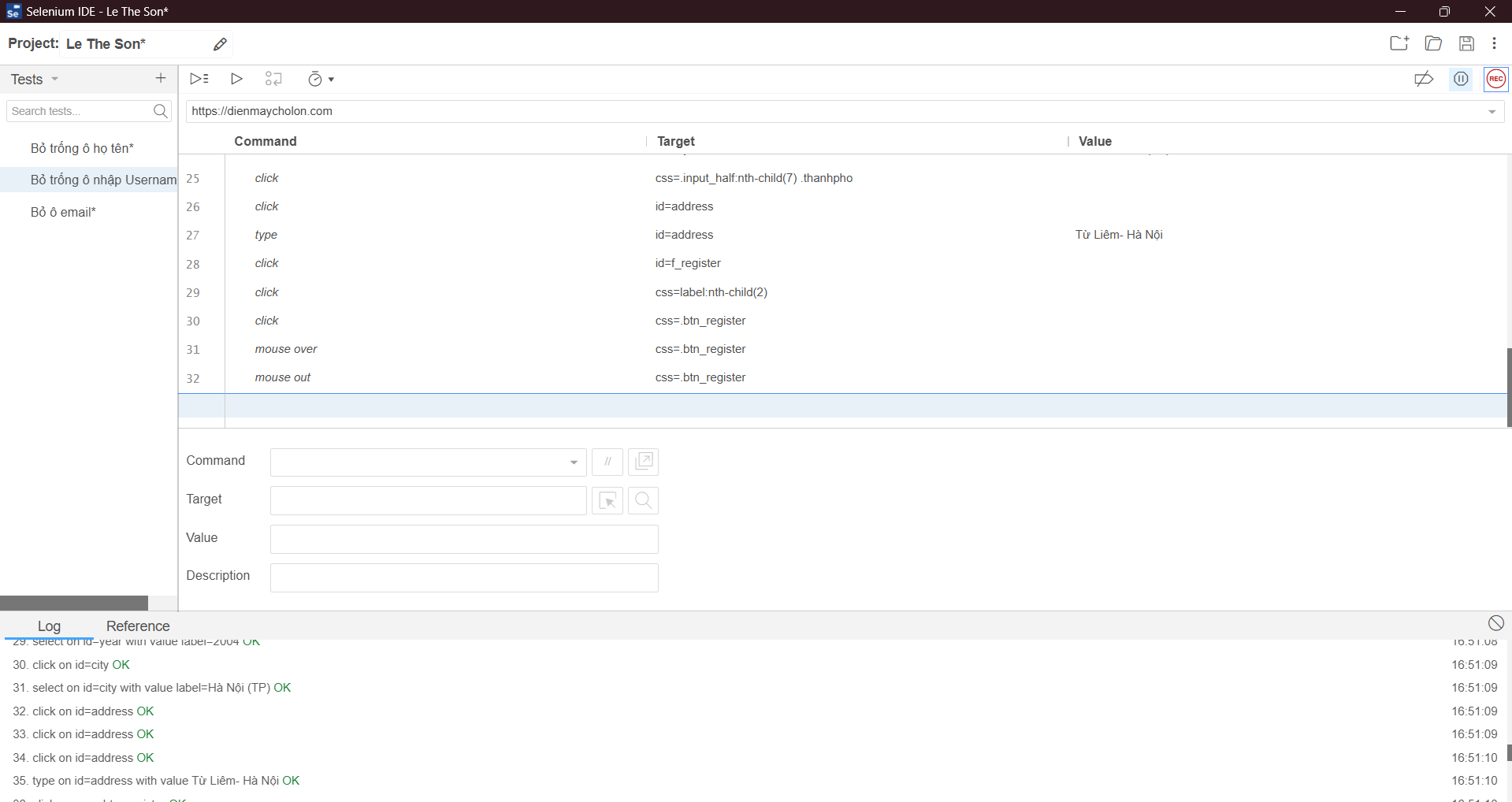
*Hình 3. 3: Kịch bản kiểm thử bỏ trống họ và tên*

* Bỏ trống ô nhập địa chỉ Email



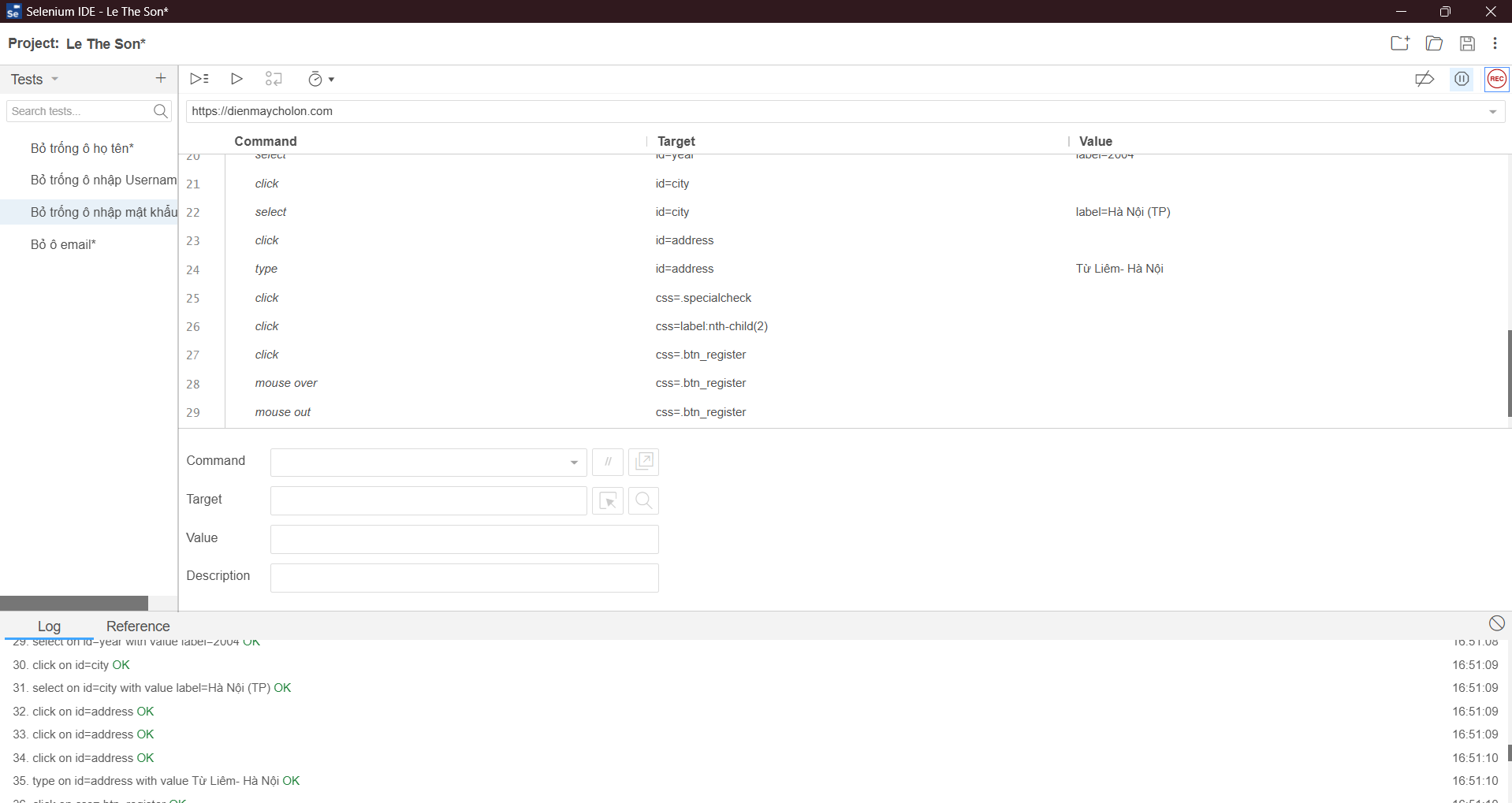
*Hình 3. 4: Kịch bản kiểm thử bỏ trống email*

* Bỏ trống ô nhập Username



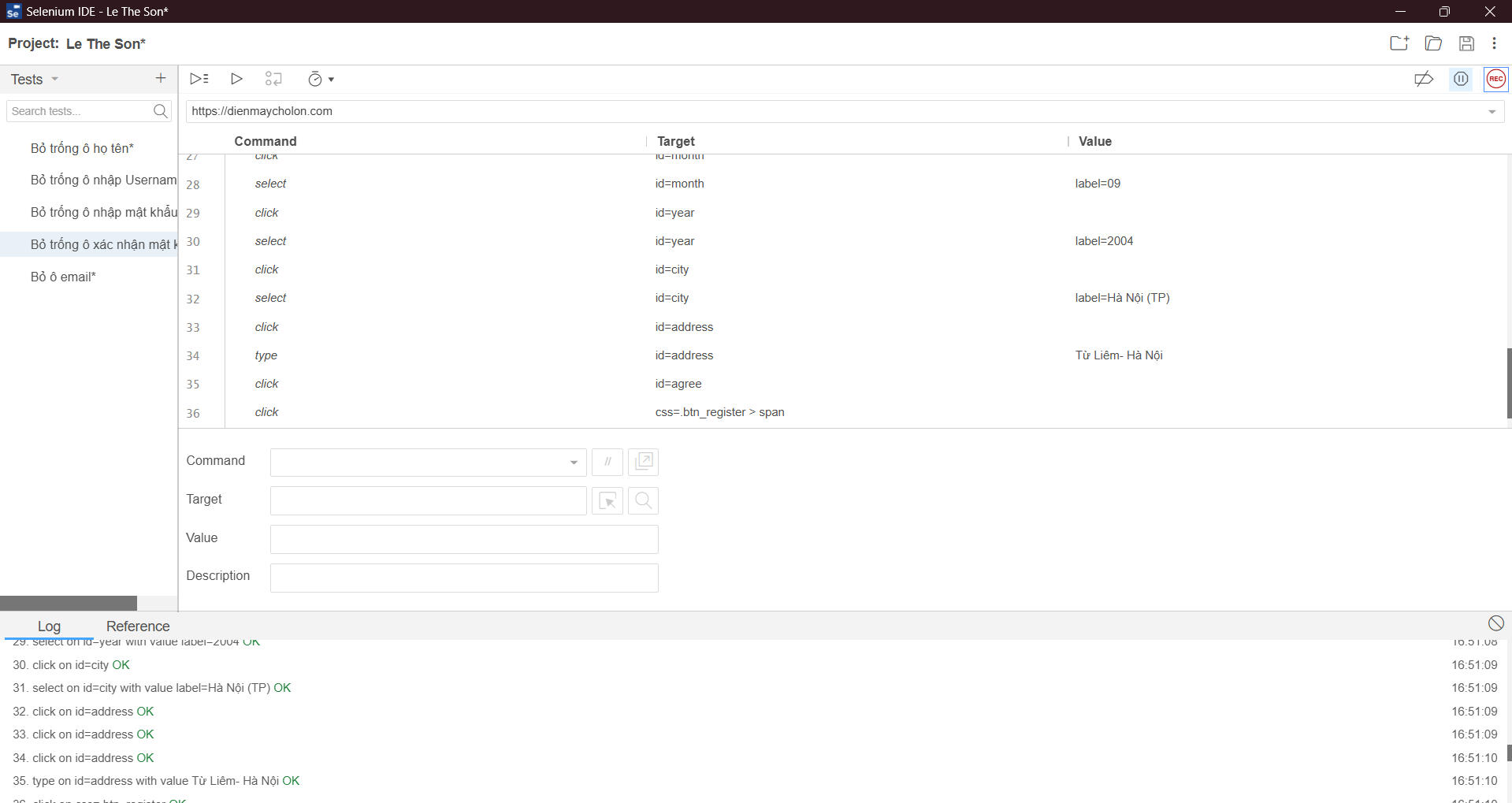
*Hình 3. 5: Kịch bản kiểm thử bỏ trống ô nhập username*

* Bỏ trống ô nhập mật khẩu



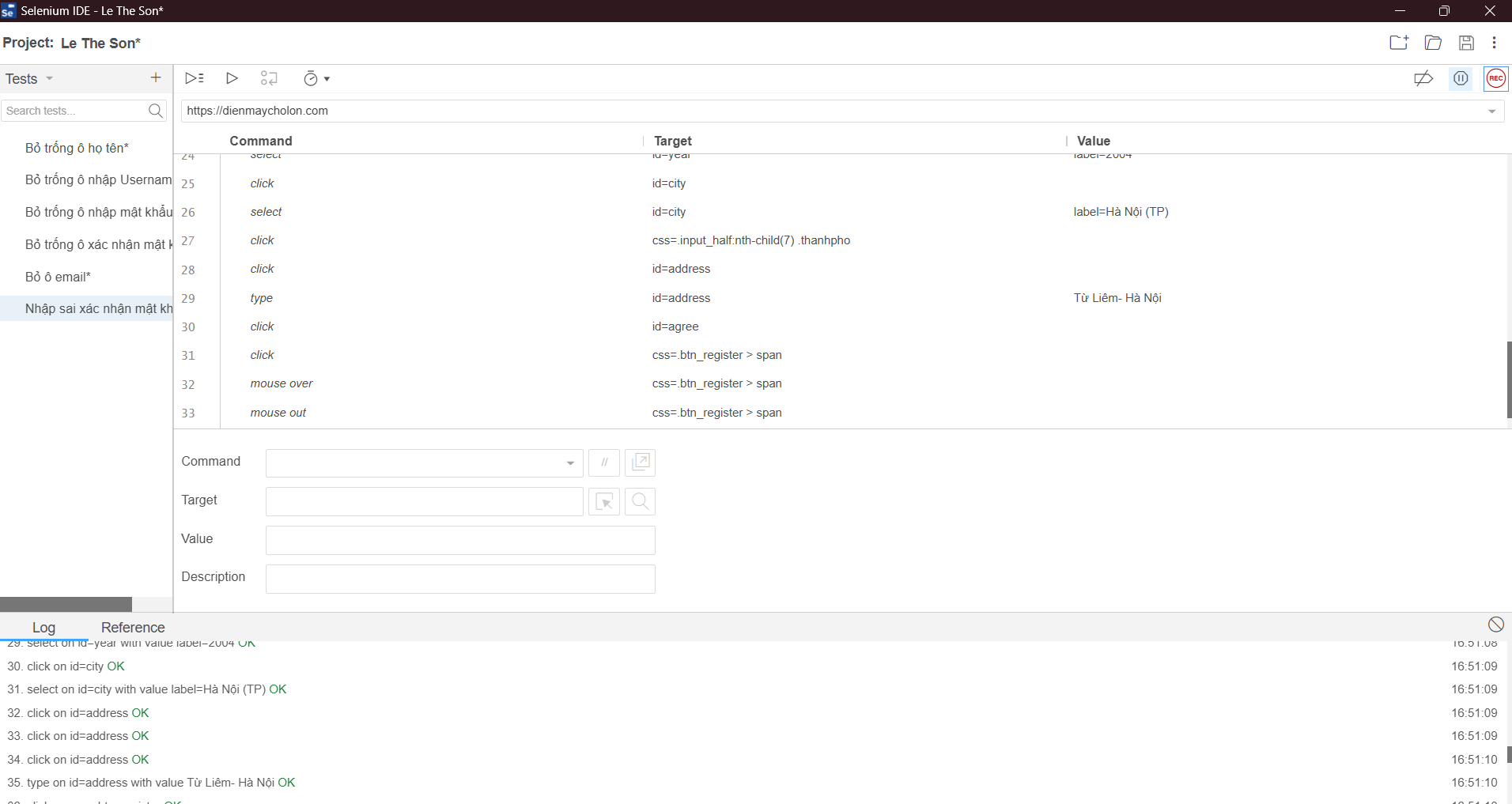
*Hình 3. 6: Kịch bản kiểm thử bỏ trống mật khẩu*

* Bỏ trống ô nhập lại mật khẩu



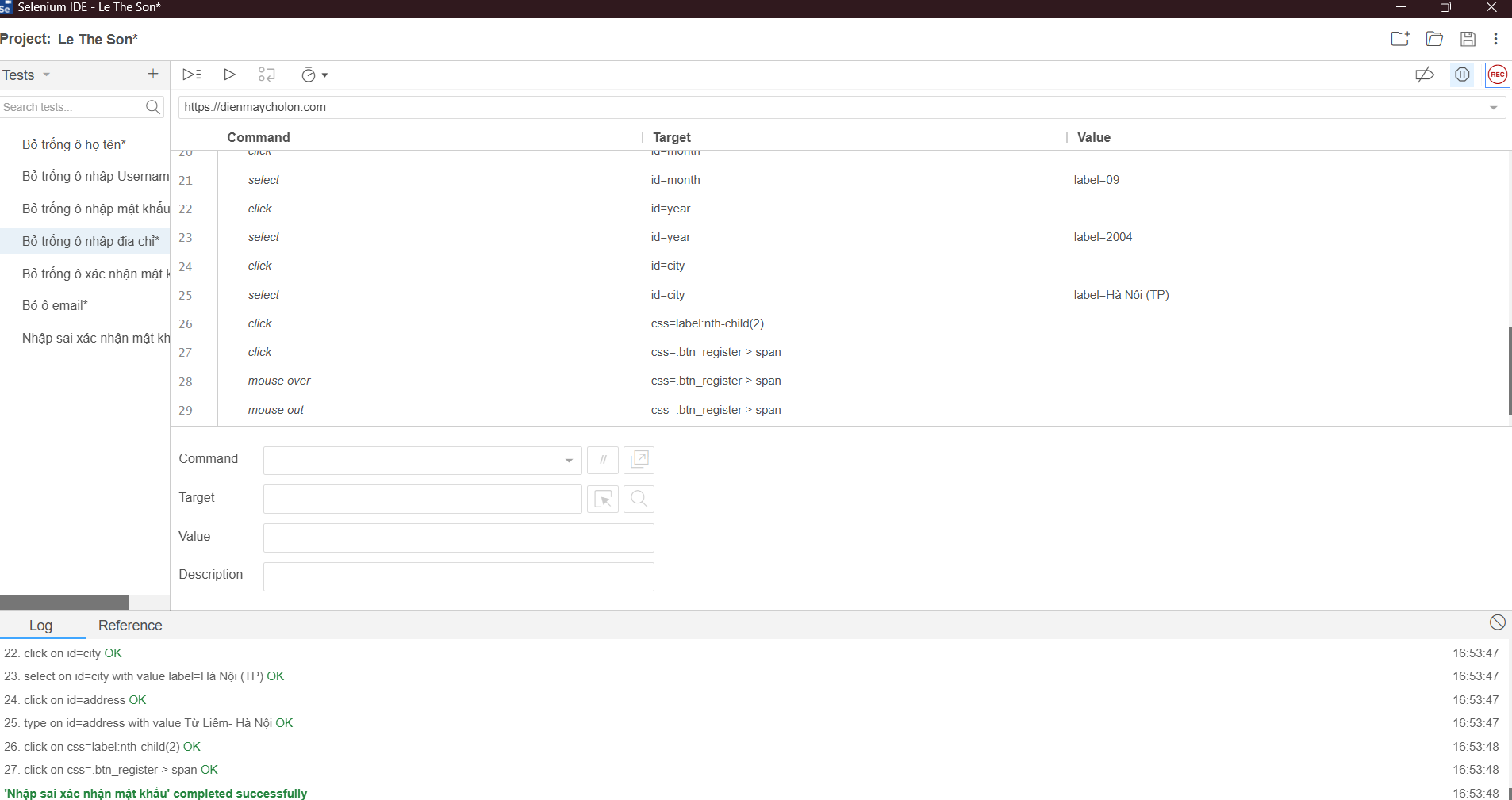
*Hình 3. 7: Kịch bản kiểm thử bỏ trống nhập lại mật khẩu*

* Nhập sai xác nhận mật khẩu



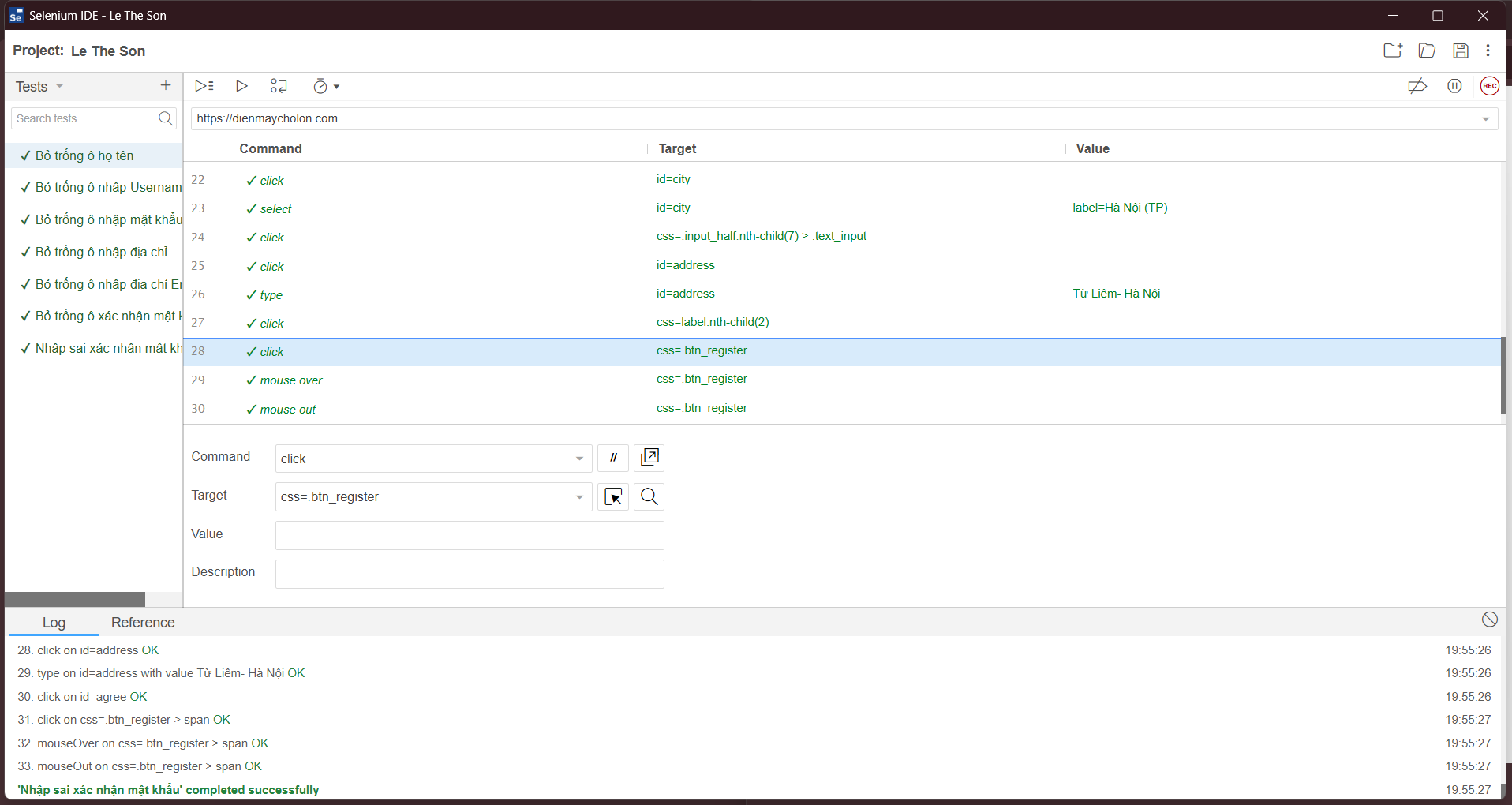
*Hình 3. 8: Kịch bản kiểm thử nhập sai mật khẩu*

* Bỏ trống ô nhập địa chỉ



*Hình 3. 9: Kịch bản kiểm thử bỏ trống ô nhập địa chỉ*

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



*Hình 3. 10: Tiến hành kiểm thử*

=> Hệ thống thông báo kiểm thử thành công.

Tương tự lặp lại các bước với các test case trên

=>Tỉ lệ thành công(PASSED) so với số TC đã lập: 100%.

### **3.3.3 Đăng nhập**

#### 3.3.3.1. Phân tích thiết kế kiểm thử.

**Giao diện chức năng**:

**A screenshot of a computer

Description automatically generated**

*Hình 3. 11: Giao diện chức năng đăng nhập*

**Mô tả giao diện:** Đăng nhập bao gồm Tên đăng nhập(Email) , Mật khẩu.

**Các điều kiện ràng buộc**: Tên đăng nhập (Email), Mật khẩu không được để trống.

*Bảng 3. 13: Đầu vào chức năng đăng nhập*

|  |  |  |
| --- | --- | --- |
| Điều kiện đầu vào | Các lớp tương đương hợp lệ | Các lớp tương đương không hợp lệ |
| Tên đăng nhập | Có trong CSDL | Để trống |
| Mật khẩu | Có trong CSDL | Để trống |

*Bảng 3. 14: Danh sách các test case*

A screenshot of a computer

Description automatically generated

#### 3.3.3.2 Thực hiện kiểm thử

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

Ta có kịch bản kiểm thử:

* Đăng nhập thành công

A screenshot of a computer

Description automatically generated

*Hình 3. 12: Kịch bản kiểm thử đăng nhập thành công*

* Sai mật khẩu

A screenshot of a computer

Description automatically generated

*Hình 3. 13: Kịch bản kiểm thử sai mật khẩu*

* Mật khẩu để trống

A screenshot of a computer

Description automatically generated

*Hình 3. 14: Kịch bản kiểm thử mật khẩu để trống*

* A close-up of a computer screen

  Description automatically generatedTên đăng nhập không đúng

*Hình 3. 15: Kịch bản kiểm thử tên đăng nhập không đúng*

* Tên đăng nhập để trống

A screenshot of a computer

Description automatically generated

*Hình 3. 16: Kịch bản kiểm thử đăng nhập để trống*

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

A screenshot of a computer

Description automatically generated

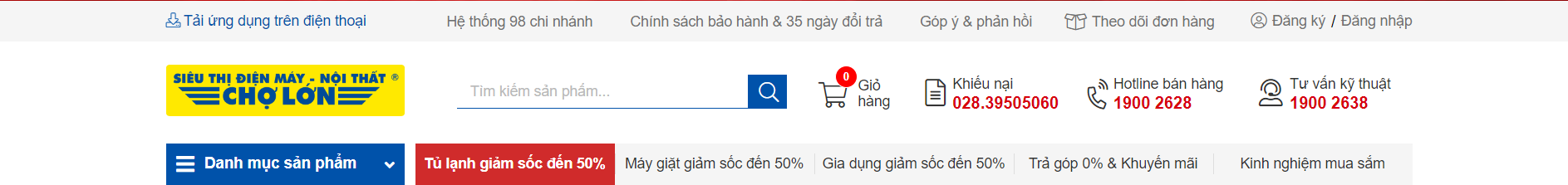
*Hình 3. 17: Tiến hành kiểm thử*

=> Hệ thống thông báo kiểm thử thành công.

=> Tỉ lệ thành công (PASSED) so với số TC đã lập: 100%.

### **3.3.4 Tìm kiếm sản phẩm**

#### 3.3.4.1 Phân tích thiệt kế kiểm thử

**Giao diện chức năng** : 

*Hình 3. 18: Giao diện chức năng tìm kiếm*

**Mô tả giao diện**: Giao diện tìm kiếm gồm Thanh tìm kiếm và biểu tượng tìm kiếm, Người dùng nhập từ khóa muốn tìm vào Thanh tìm kiếm và kích biểu tượng “Kính lúp” có nền vàng ở bên trái thanh tìm kiếm.

**Các điều kiện ràng buộc**: từ khóa không được để trống

*Bảng 3. 15: Điều kiện đầu vào chức năng đăng nhập*

|  |  |  |
| --- | --- | --- |
| Điều kiện đầu vào | Các lớp tương đương hợp lệ | Các lớp tương đương không hợp lệ |
| Từ khóa nhập vào thanh tìm kiếm | Có trong cơ sở dữ liệu | Kí tự đặc biệt |

*Bảng 3. 16: Danh sách các TC*

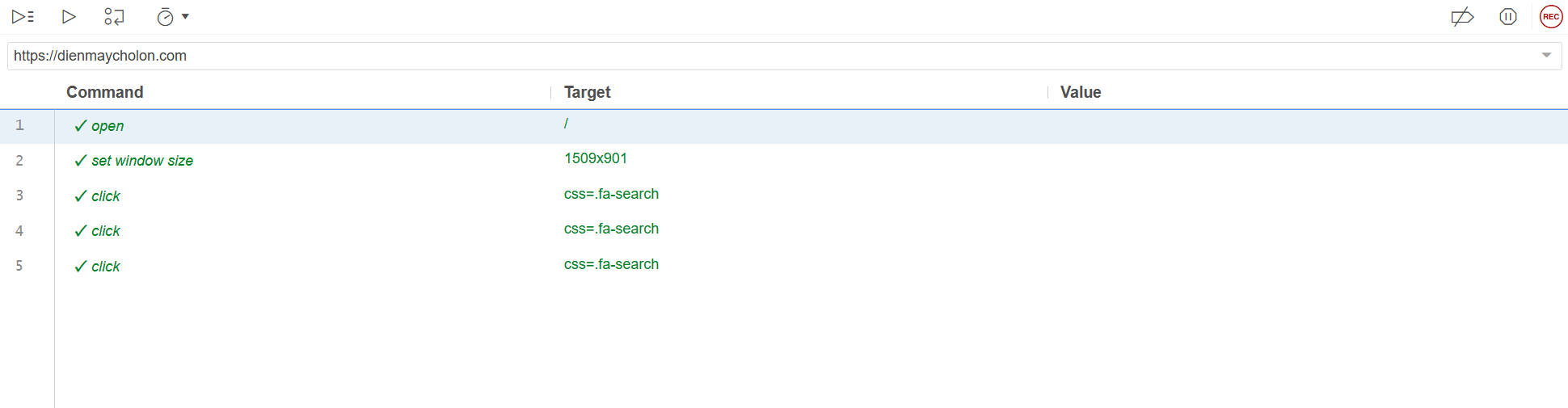


#### 3.3.4.2 Thực hiện kiểm thử

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

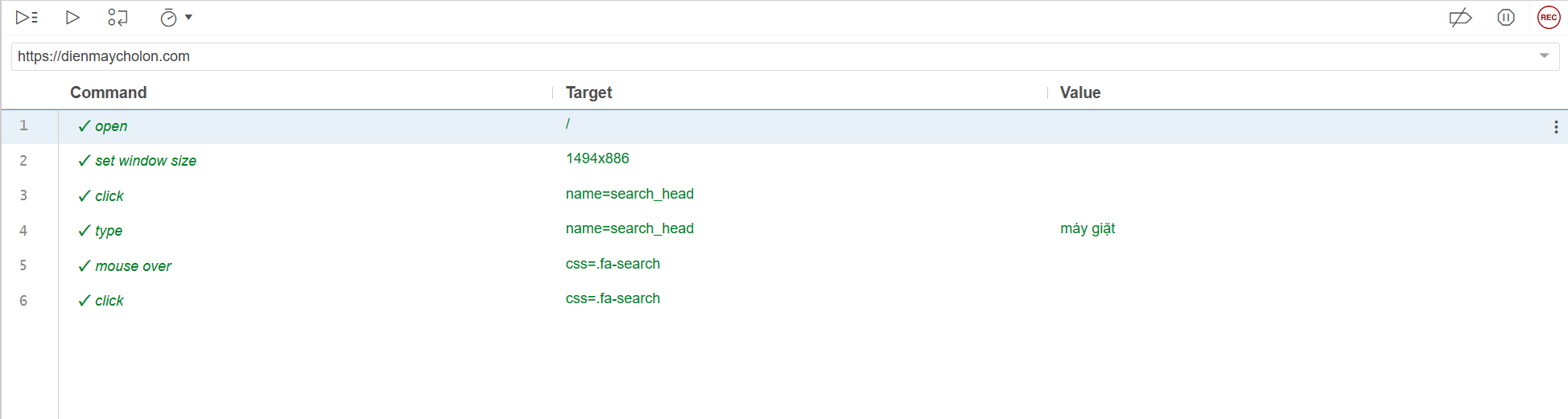
Ta có kịch bản kiểm thử:

* Kịch bản kiểm thử ô tìm kiếm khi dữ liệu trống



*Hình 3. 19: Kịch bản kiểm thử ô tìm kiếm khi dữ liệu trống*

* Kịch bản kiểm thử ô tìm kiếm khi nhập dữ liệu hợp lệ



*Hình 3. 20: Kịch bản kiểm thử ô tìm kiếm khi nhập dữ liệu hợp lệ*

* Kịch bản kiểm thử ô tìm kiếm khi nhập dữ liệu có ký tự đặc biệt



*Hình 3. 21: Kịch bản kiểm thử ô tìm kiếm khi nhập dữ liệu có ký tự đặc biệt*

* Kịch bản kiểm thử tìm kiếm với từ khóa vô nghĩa hoặc không có trong cơ sở dữ liệu



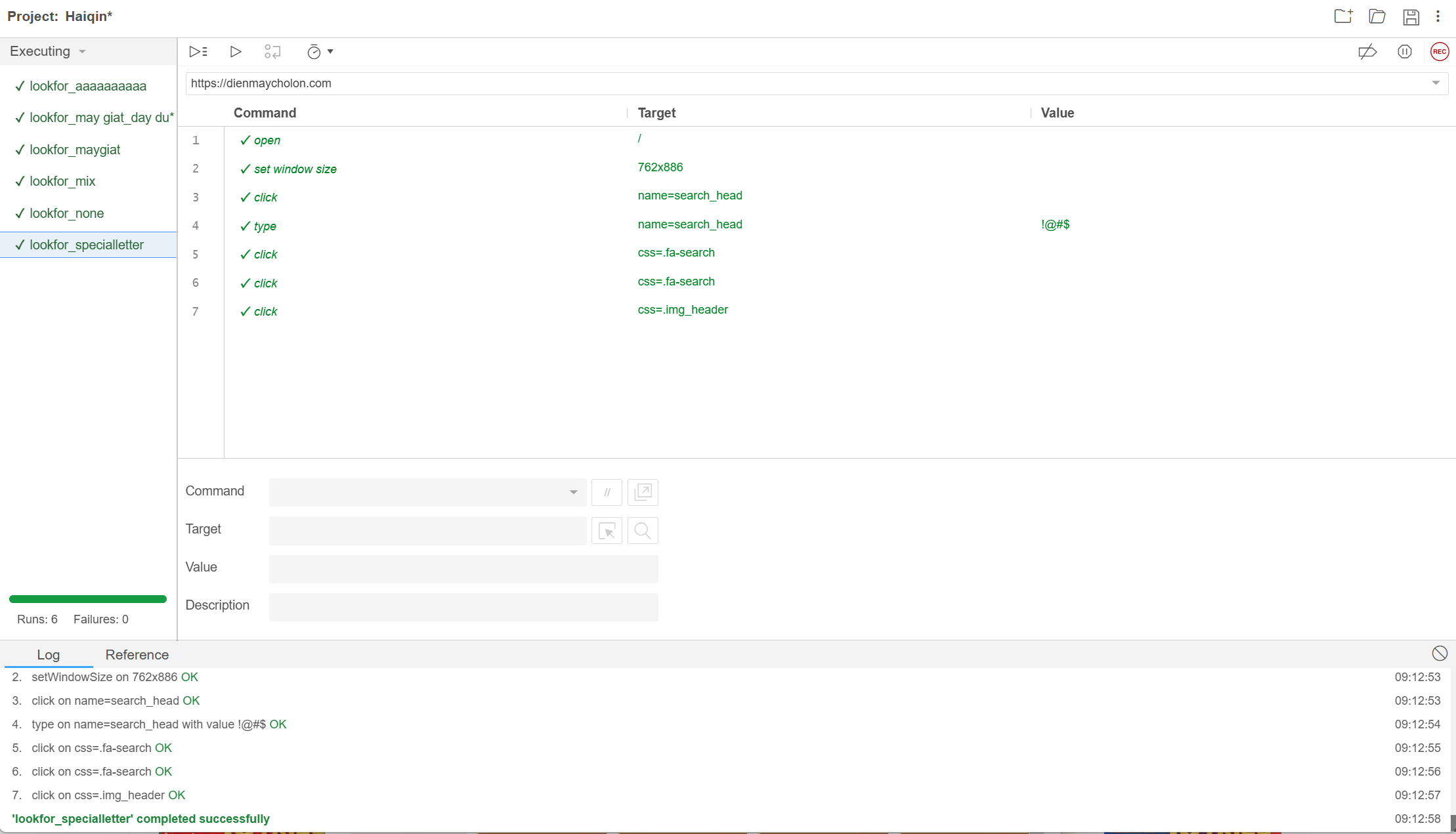
*Hình 3. 22: Kịch bản kiểm thử tìm kiếm với từ khóa vô nghĩa hoặc không có trong cơ sở dữ liệu*

* Kịch bản kiểm thử tìm kiếm với từ khóa tiếng Việt viết hoa, thường, có dấu, không dấu xen kẽ



*Hình 3. 23: Kịch bản kiểm thử tìm kiếm với từ khóa tiếng Việt viết hoa, thường, có dấu, không dấu xen kẽ*

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



*Hình 3. 24: Tiến hành kiểm thử*

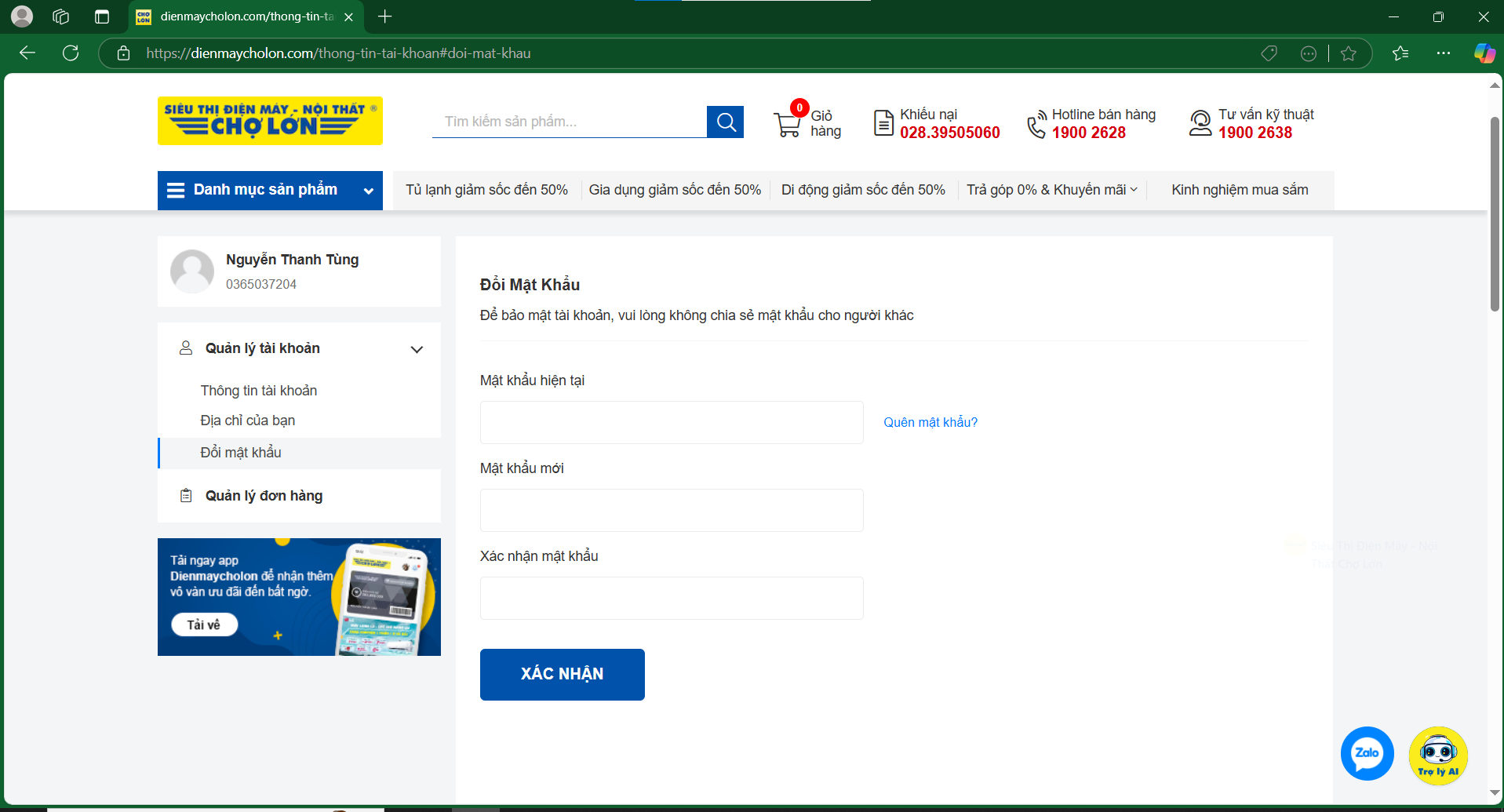
=> Hệ thống thông báo kiểm thử thành công

*=>Tỉ lệ thành công(PASSED) so với số TC đã lập: 100%.*

### **3.3.5 Đổi mật khẩu**

**3.3.5.1 Phân tích thiết kế kiểm thử**

**Giao diện chức năng**



*Hình 3.25: Giao diện chức năng đổi mật khẩu*

**Mô Tả Giao Diện**

* **Header:** Logo, thanh tìm kiếm, biểu tượng giỏ hàng, số điện thoại hỗ trợ.
* **Sidebar**: Thông tin người dùng, các tùy chọn quản lý tài khoản và đơn hàng.
* **Nội dung chính:** Tiêu đề "Đổi Mật Khẩu", hướng dẫn bảo mật, các trường nhập liệu (mật khẩu hiện tại, mật khẩu mới, xác nhận mật khẩu), liên kết "Quên mật khẩu?", nút "XÁC NHẬN".

Giao diện này giúp người dùng dễ dàng thay đổi mật khẩu tài khoản một cách an toàn và thuận tiện.

**Điều kiện ràng buộc**: tồn tại tài khoản trong CSDL

|  |  |  |  |
| --- | --- | --- | --- |
| Các trường | Điều kiện | Hợp lệ | Không hợp lệ |
| Mật khẩu cũ | Nhập đúng mật khẩu đã đăng nhập | Đã tồn tại trong CSDL | Nhập sai mật khẩu |
| Để trống |
| Mật khẩu mới | Không để trống, không vượt quá lớn hơn 6 ký tự | >= 6 ký tự và không để trống bất kỳ ô nào | < 6 ký tự |
| Để Trống |
| Nhập lại mới | Phải đúng với mật khẩu mới | Trùng mật khẩu vừa nhập | Xác nhận sai |
| Để trống |

*Bảng 3.17: Điều kiện ràng buộc chức năng đổi mật khẩu*

**Danh sách test case**

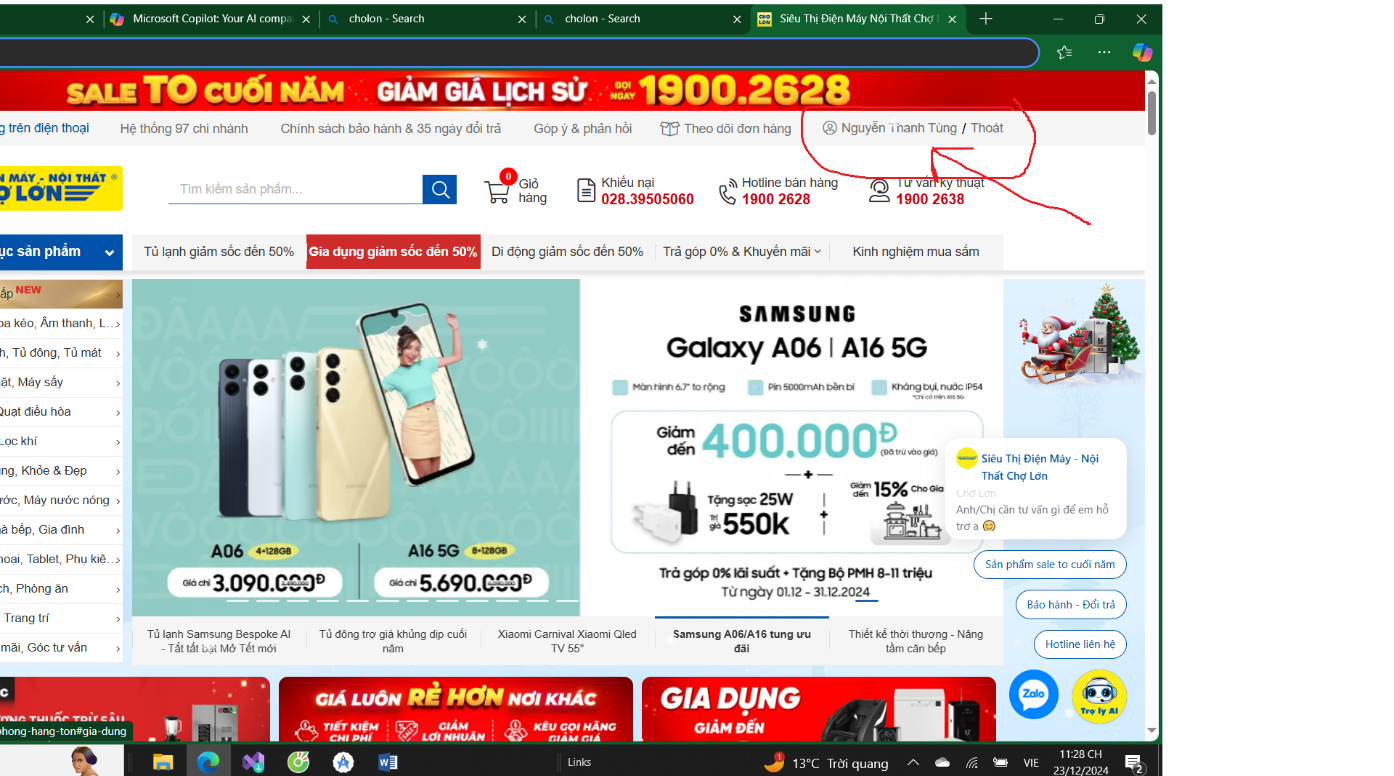
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Mô tả test case | INPUT | Dữ Liệu | OUTPUT | NOTICE |
| 1 | TC\_01 : Mật khẩu cũ không chính xác | 1.Nhập sai mật khẩu hiện tại  2.Nhập hoàn thiện những thông tin còn lại | Mật khẩu hiện tại : 123456789  Mật khẩu mới : thanhtung123  Xác nhận mật khẩu : thanhtung123 | Thông báo mật khẩu cũ không chính xác.Hiển thị thông báo “Mật khẩu cũ không chính xác . Vui lòng nhập lại ” |  |
| 2 | TC\_02:Mật khẩu mới ngắn hơn 6 ký tự | 1.Nhập mật khẩu với 5 ký tự  2.Nhập hoàn hiện những thông tin còn lại | Mật khẩu hiện tại : thanhtung11  Mật khẩu mới : thanh  Xác nhận mật khẩu : thanh | Thông báo : “Vui lòng nhập giá trị lớn hơn 6 ký tự “ |  |
| 3 | TC\_03: Xác nhận mật khẩu không khớp | 1.Nhập mật khẩu với hiện tại  2.Nhập mật khẩu mới  3.Nhập xác nhận mật khẩu | Mật khẩu hiện tại : thanhtung11  Mật khẩu mới : thanhtung123  Xác nhận mật khẩu : thanhtung12 | Thông báo : “Nhập lại mật khẩu không chính xác “ |  |
| 4 | TC\_04: Bỏ trống mật khẩu cũ | 1.Nhập mật khẩu mới  2.Nhập xác nhận mật khẩu | Mật khẩu mới : thanhtung123  Xác nhận mật khẩu : thanhtung123 | Thông báo:”Mật khẩu cũ không chính xác ,Vui lòng nhập lại “ |  |
| 5 | TC\_05: Bỏ trống mật khẩu mới | 1.Nhập mật khẩu hiện tại  2.Nhập xác nhận mật khẩu | Mật khẩu hiện tại : thanhtung11  Xác nhận mật khẩu : thanhtung123 | Thông báo 1 :”Vui lòng nhập mật khẩu của bạn ”  Thông báo 2 :”Nhập lại mật khẩu không chính xác “ |  |
| 6 | TC\_06: Bỏ trống xác nhận mật khẩu | 1.Nhập mật khẩu với hiện tại  2.Nhập mật khẩu mới | Mật khẩu hiện tại : thanhtung11  Mật khẩu mới : thanhtung123 | Thông báo : “Nhập lại mật khẩu không chính xác “ |  |
| 7 | TC\_07: Đổi mật khẩu thành công | 1.Nhập mật khẩu với hiện tại  2.Nhập mật khẩu mới  3.Nhập xác nhận mật khẩu | Mật khẩu hiện tại : thanhtung11  Mật khẩu mới : thanhtung123  Xác nhận mật khẩu : thanhtung123 | Thông báo :Đổi mật khẩu thành công “ |  |

*Bảng 3.18: Danh sách các test case*

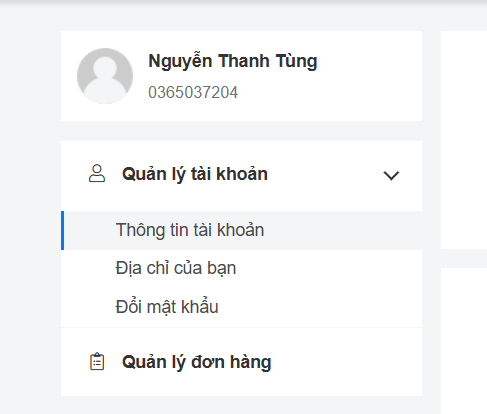
**3.3.5.2 Các bước thực hiện đổi mật khẩu**

Bước 1: Truy cập đường link : <https://dienmaycholon.com>

Bước2: Click vào hướng mũi tên theo hình ảnh

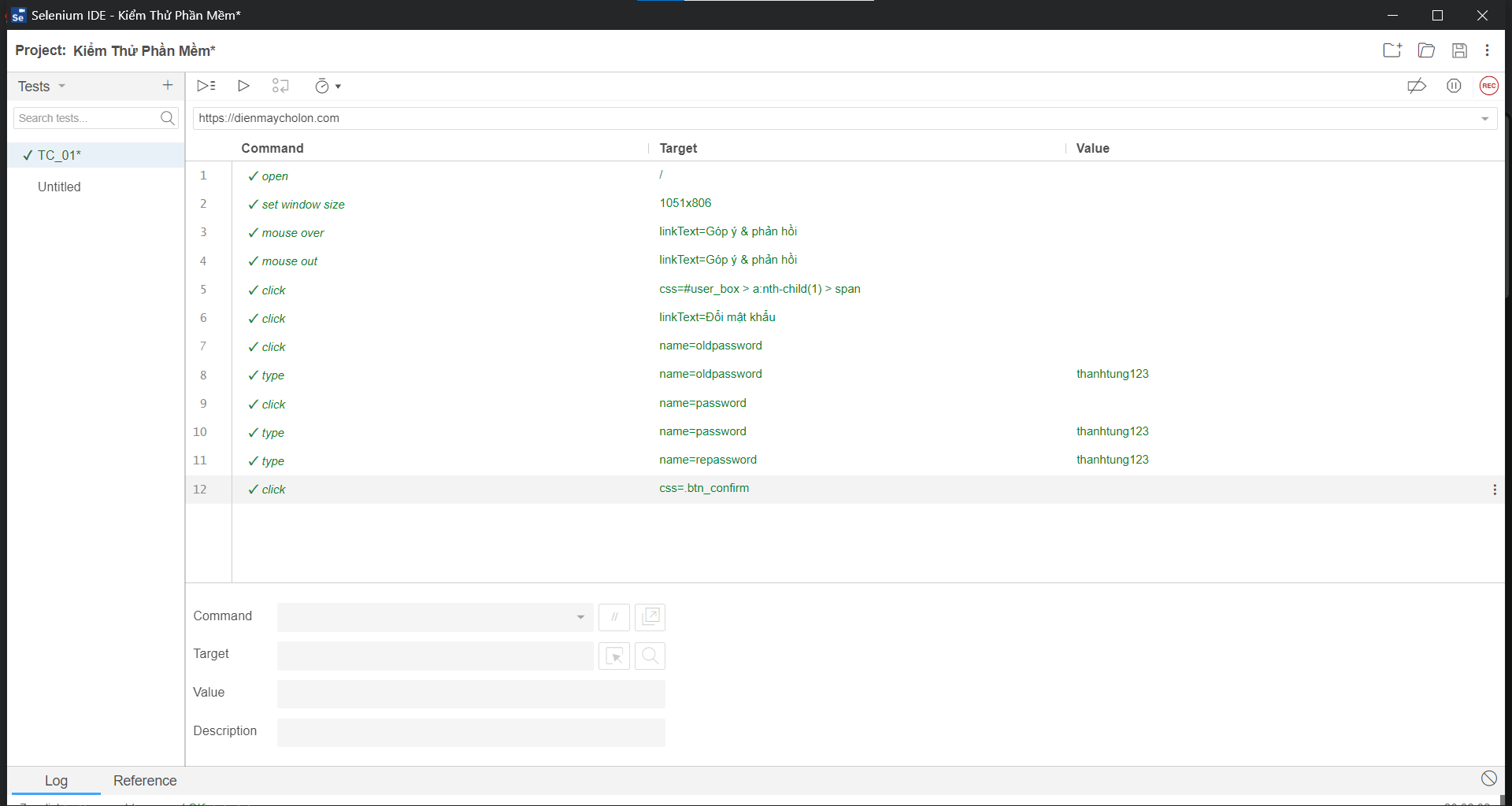


Bước 3 : Click đổi mật khẩu



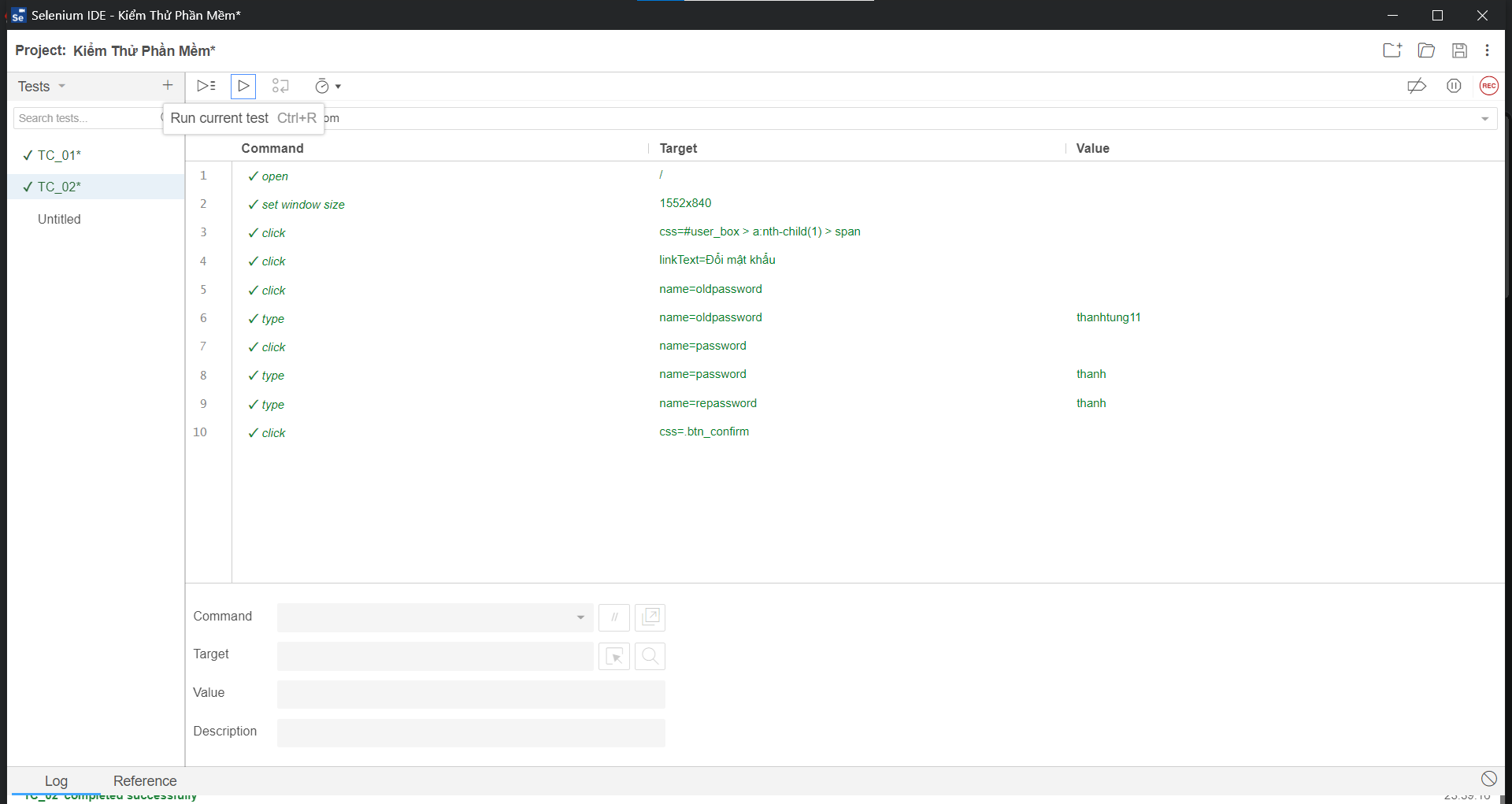
**3.3.5.3.Kết quả các test case**

**Nhập mật khẩu hiện tại không chính xác**



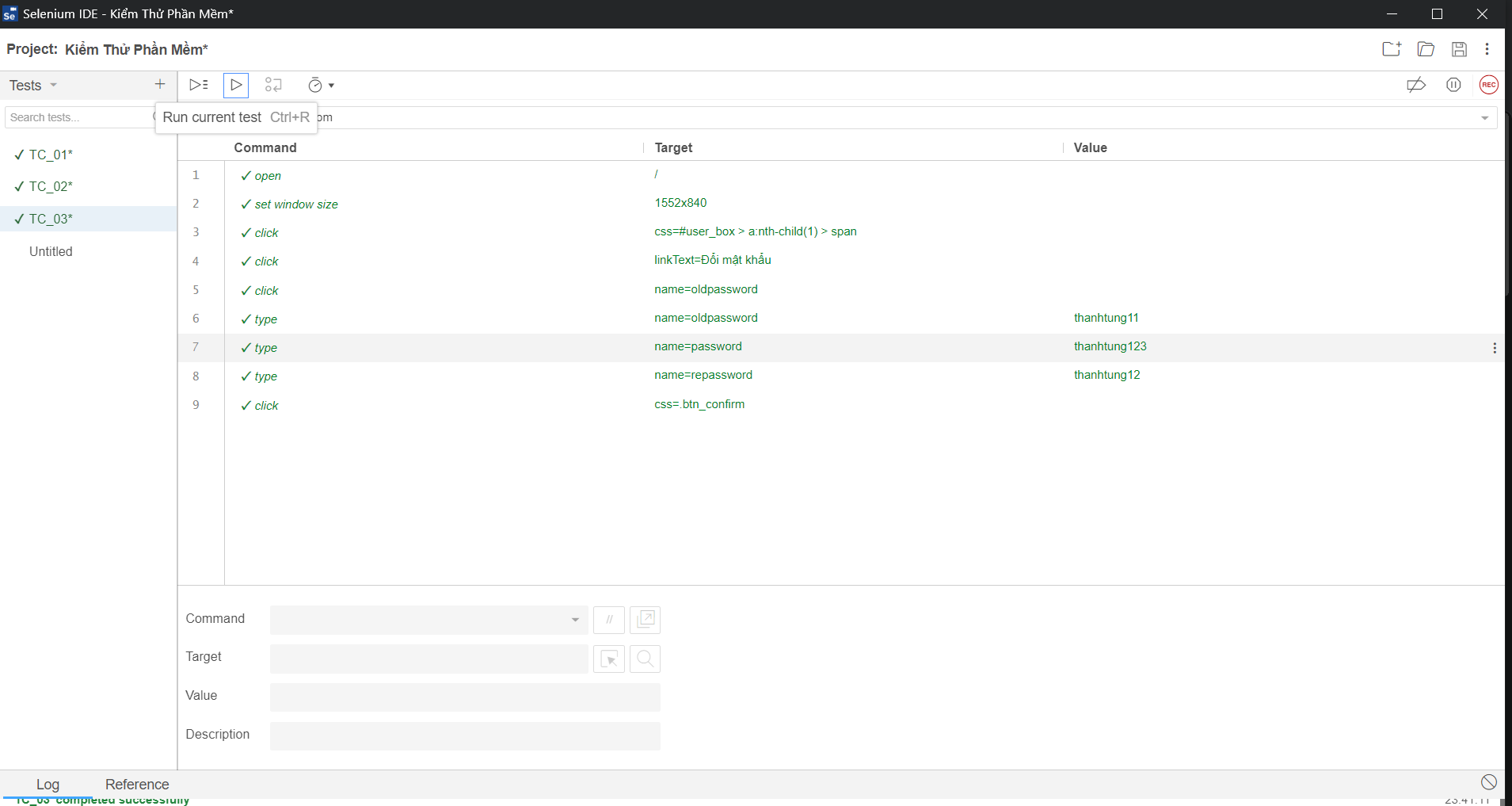
*Hình 3.26: Kịch bản kiểm thử nhập sai mật khẩu hiện tại*

**Mật khẩu mới ngắn hơn 6 ký tự**



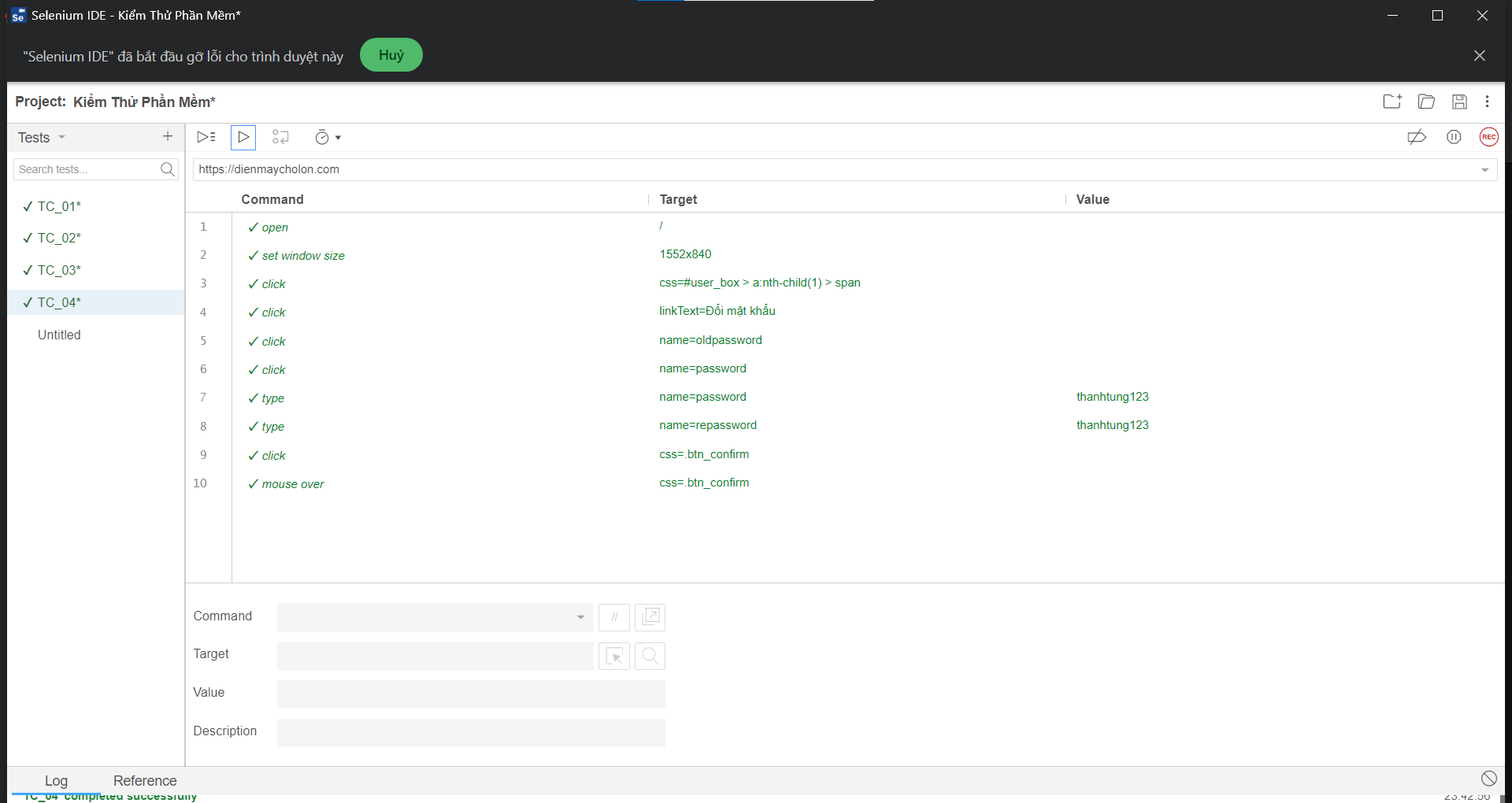
*Hình 3.27: Kịch bản kiểm thử nhập mật khẩu mới ngắn hơn 6 ký tự*

**Xác nhận mật khẩu không khớp**



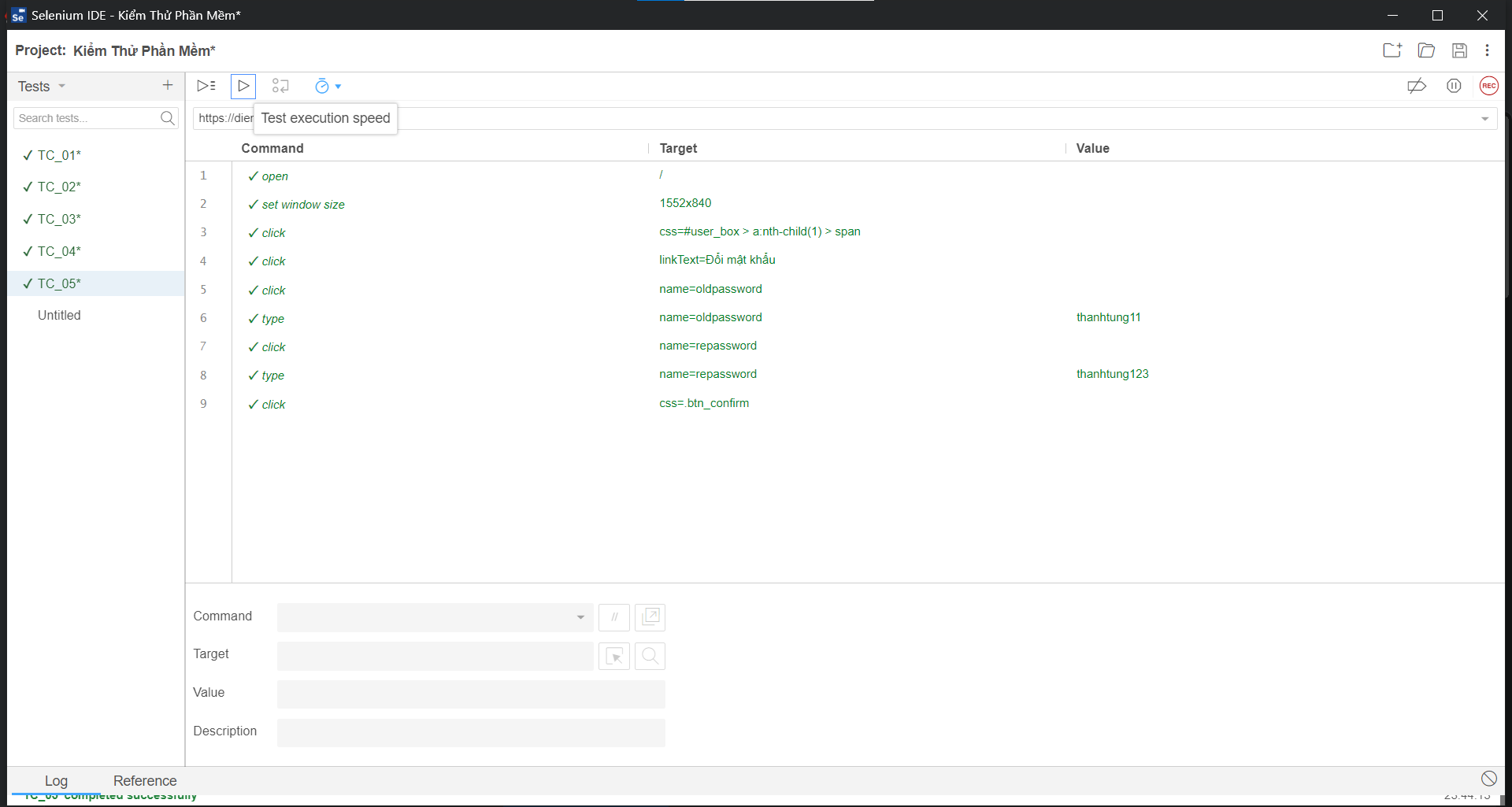
*Hình 3.28: Kịch bản kiểm thử nhập xác nhận mật khẩu*

Bỏ Trống Mật Khẩu Cũ



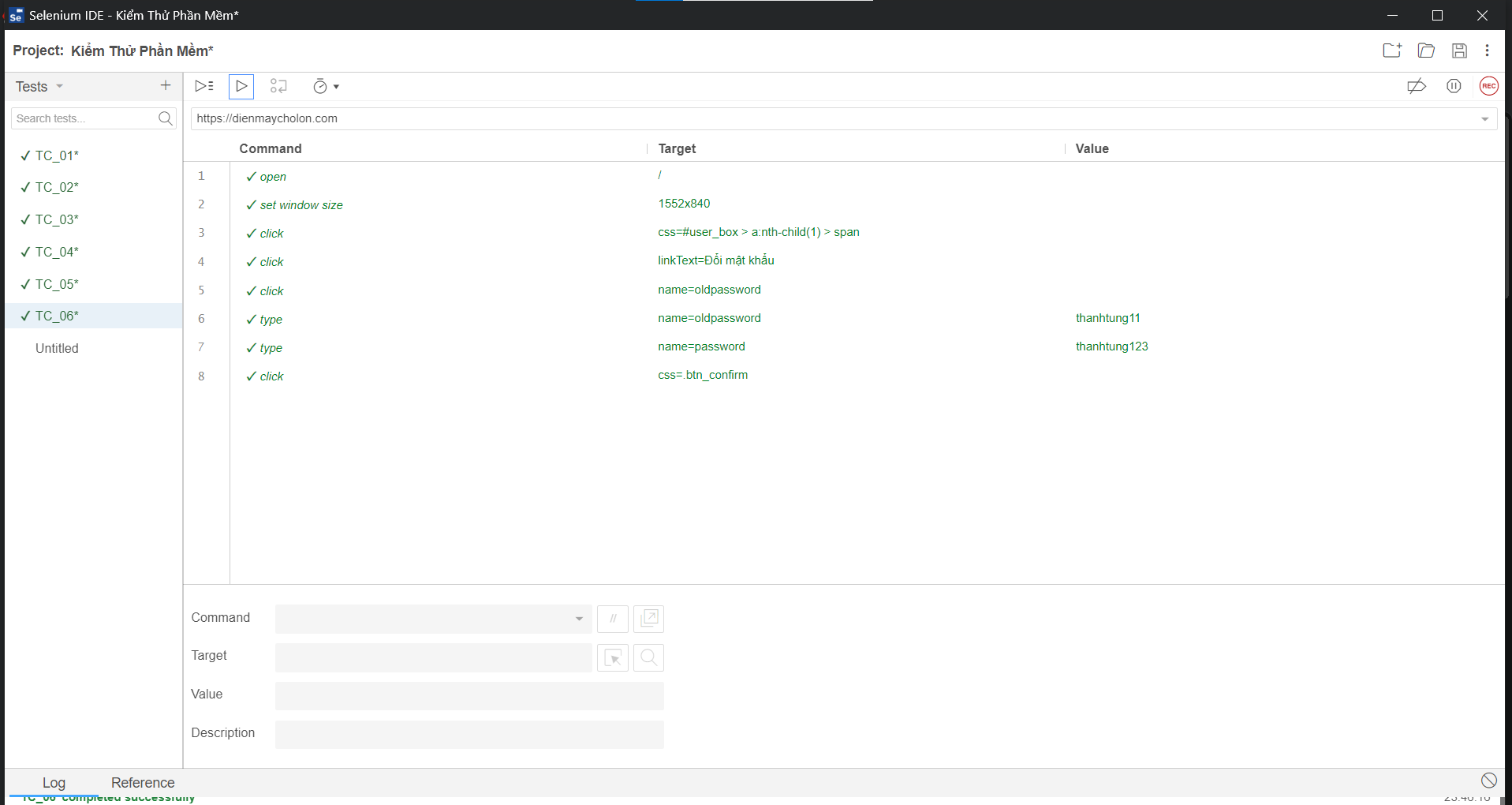
*Hình 3.29: Kịch bản kiểm thử bỏ trống mật khẩu cũ*

**Bỏ trống mật khẩu mới**



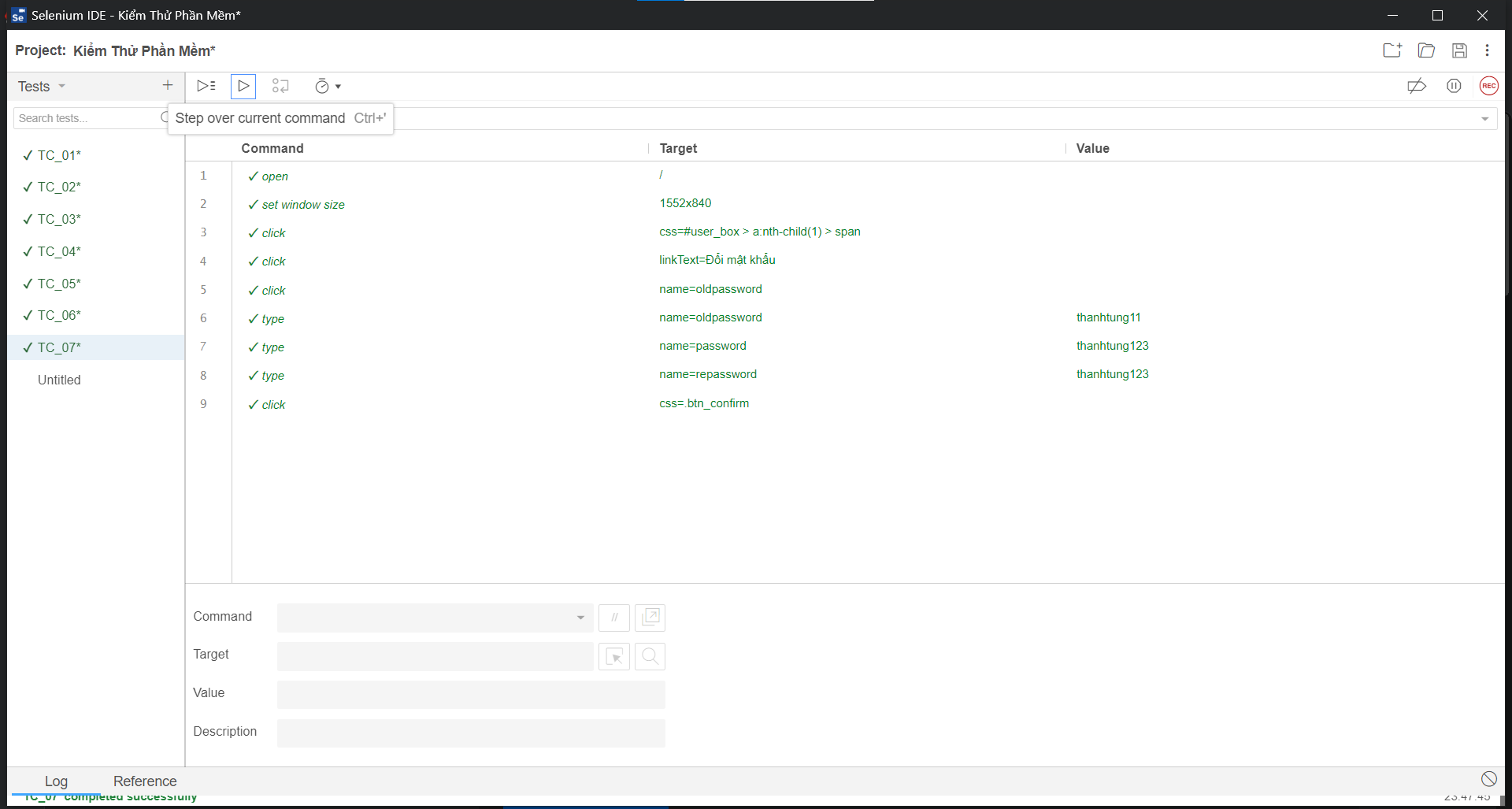
*Hình 3.30: Kịch bản kiểm thử bỏ trống mật khẩu mới*

**Bỏ Trống xác nhận mật khẩu**



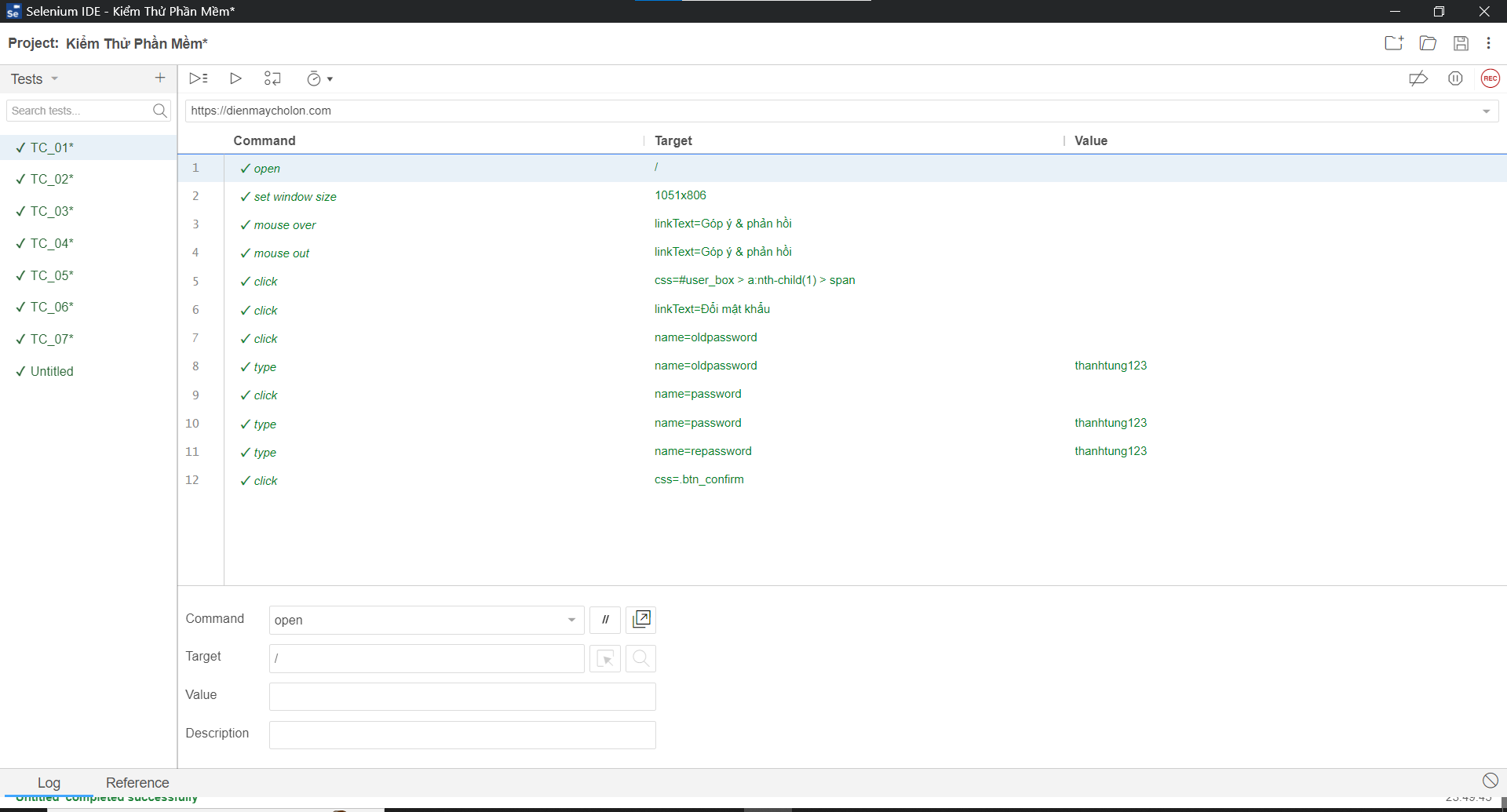
*Hình 3.31: Kịch bản kiểm thử bỏ trống xác nhận mật khẩu cũ*

**Đổi mật khẩu thành công**



*Hình 3.32: Kịch bản kiểm thử đổi mật khẩu thành công*

**Thông báo kiểm thử thành công**



*Hình 3.33: Chạy tất cả các test case*

=> Hệ thống thông báo kiểm thử thành công.

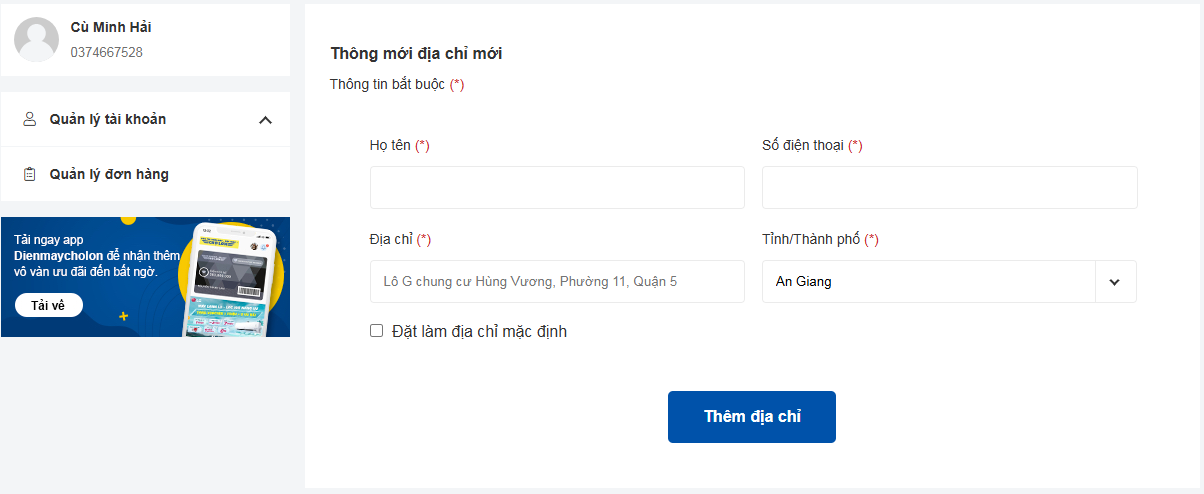
Tương tự lặp lại các bước với các test case trên

=> Tỉ lệ thành công(PASSED) so với số TC đã lập: 100%.

### **3.3.6 Thêm mới địa chỉ**

#### 3.3.6.1. Phân tích thiết kế kiểm thử.

**Giao diện chức năng:**



*Hình 3. 34: Giao diện chức năng thêm mới địa chỉ*

**Mô tả giao diện**

* + Có các trường: Họ tên, Số điện thoại, Địa chỉ, Tỉnh/Thành phố
* Có các ô input, label nhập dữ liệu
* Có nút Thêm địa chỉ

**Điều kiện ràng buộc**: tồn tại tài khoản trong CSDL

*Bảng 3. 19: Điều kiên ràng buộc chức năng thêm mới địa chỉ*

|  |  |  |  |
| --- | --- | --- | --- |
| Các trường | Điều kiện | Hợp lệ | Không hợp lệ |
| Họ tên | Không để trống | Không để trống | Để trống |
| Số điện thoại | Không để trống | Không để trống | Để trống |
| Địa chỉ | Không để trống | Không để trống | Để trống |

Danh sách test case:

*Bảng 3. 20: Danh sách test case*

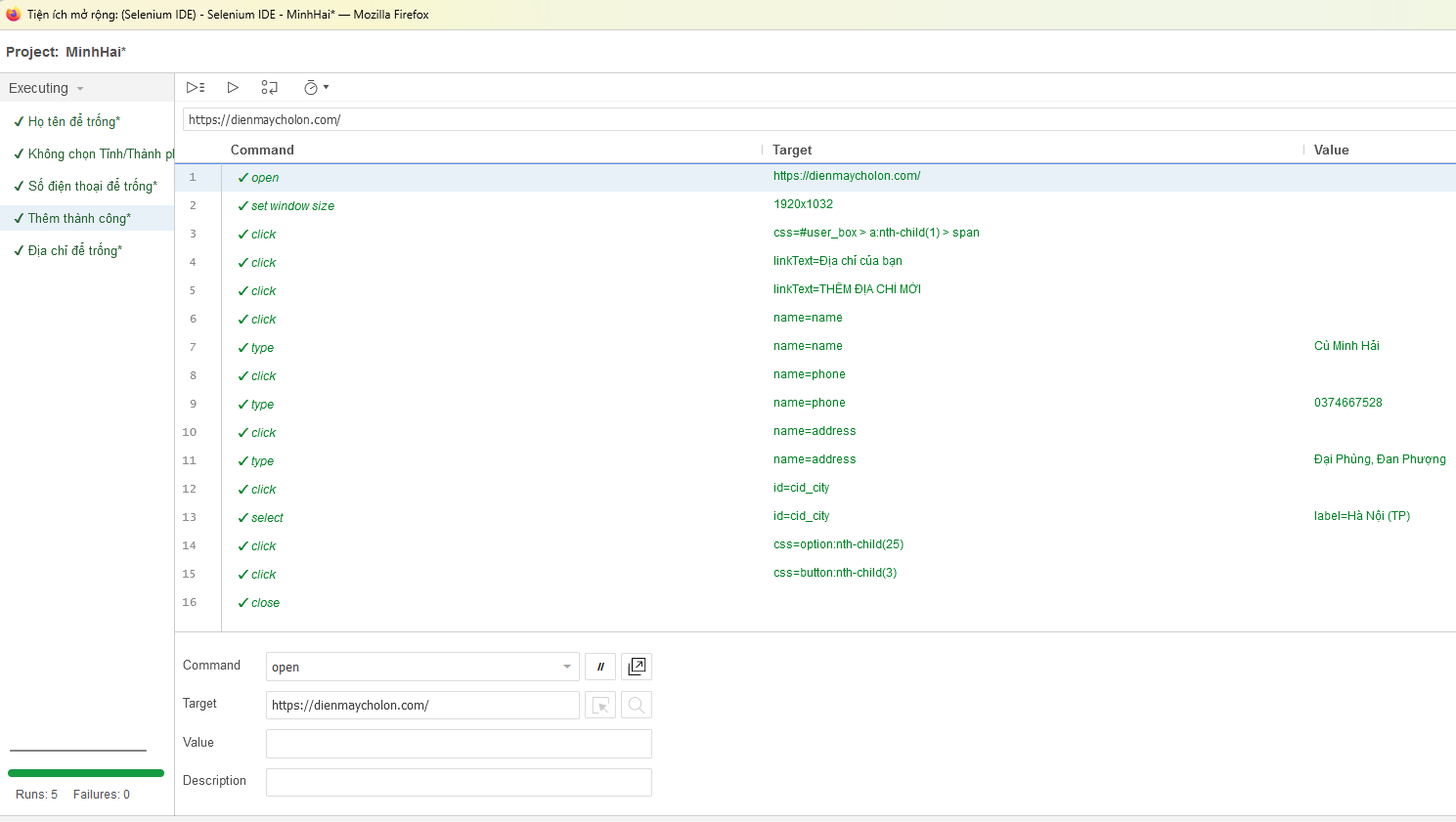


#### 3.3.6.2 Thực hiện kiểm thử

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

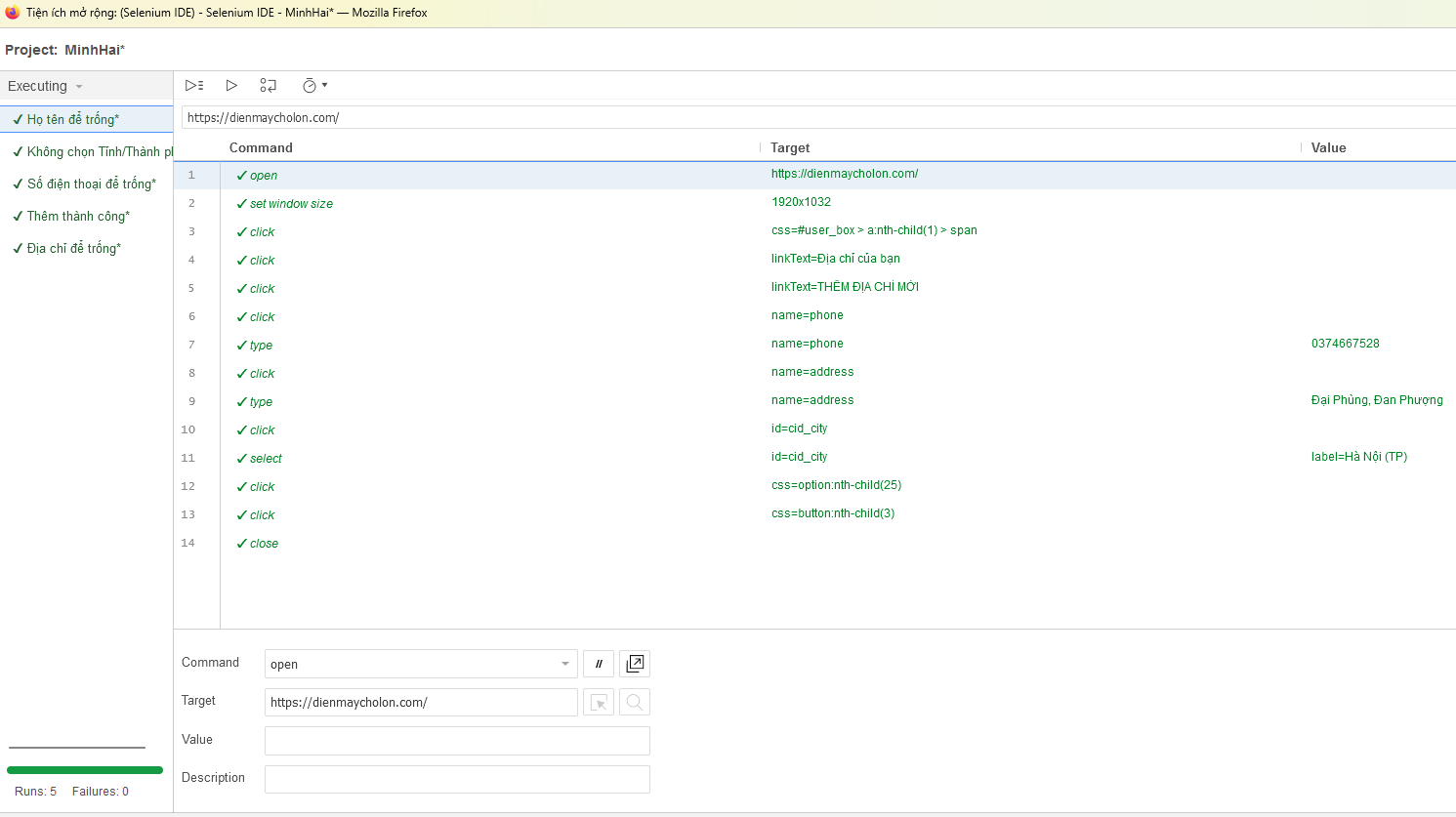
Ta có kịch bản kiểm thử:

* Thêm thành công



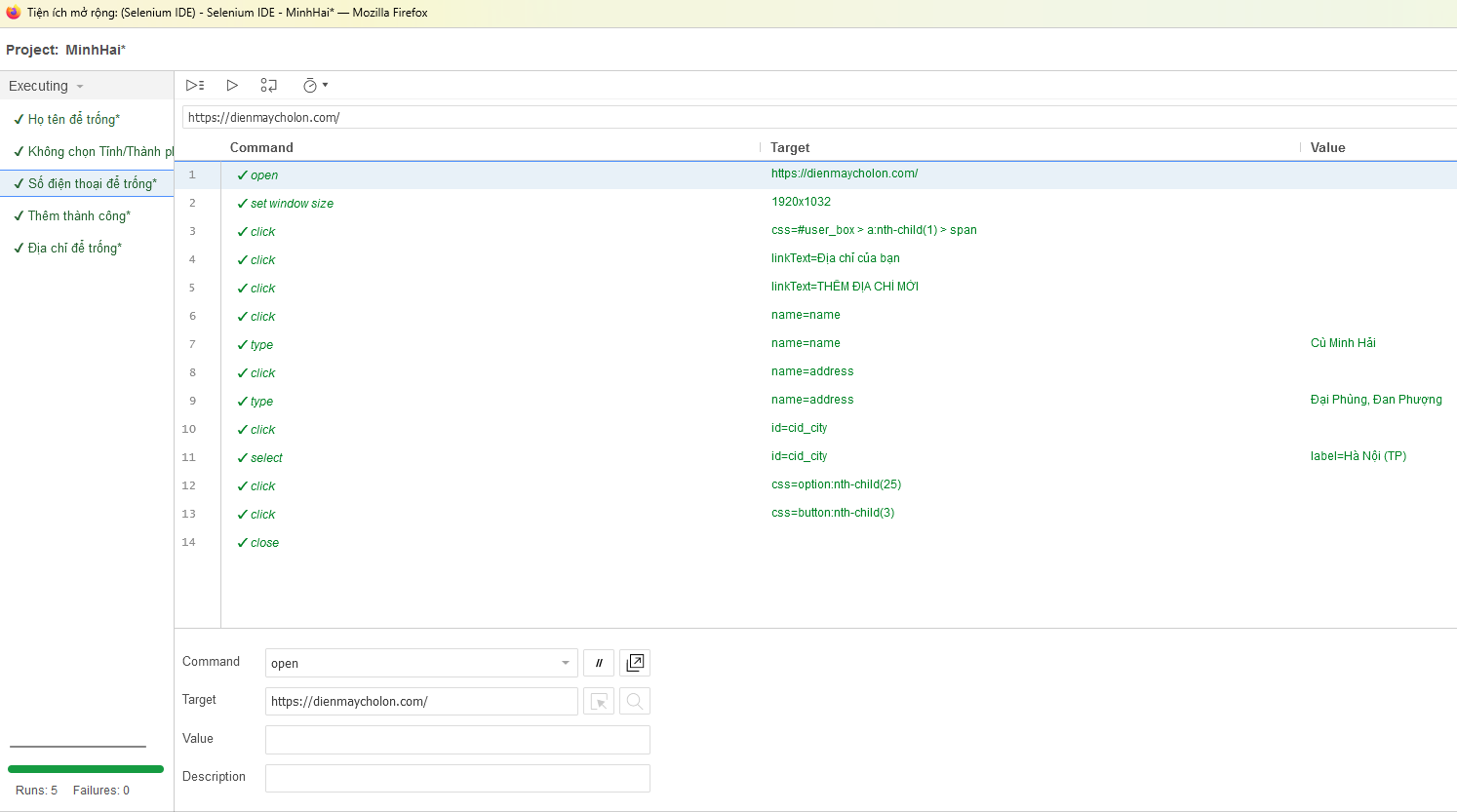
*Hình 3. 35: Kịch bản kiểm thử thêm thành công*

* Họ tên để trống



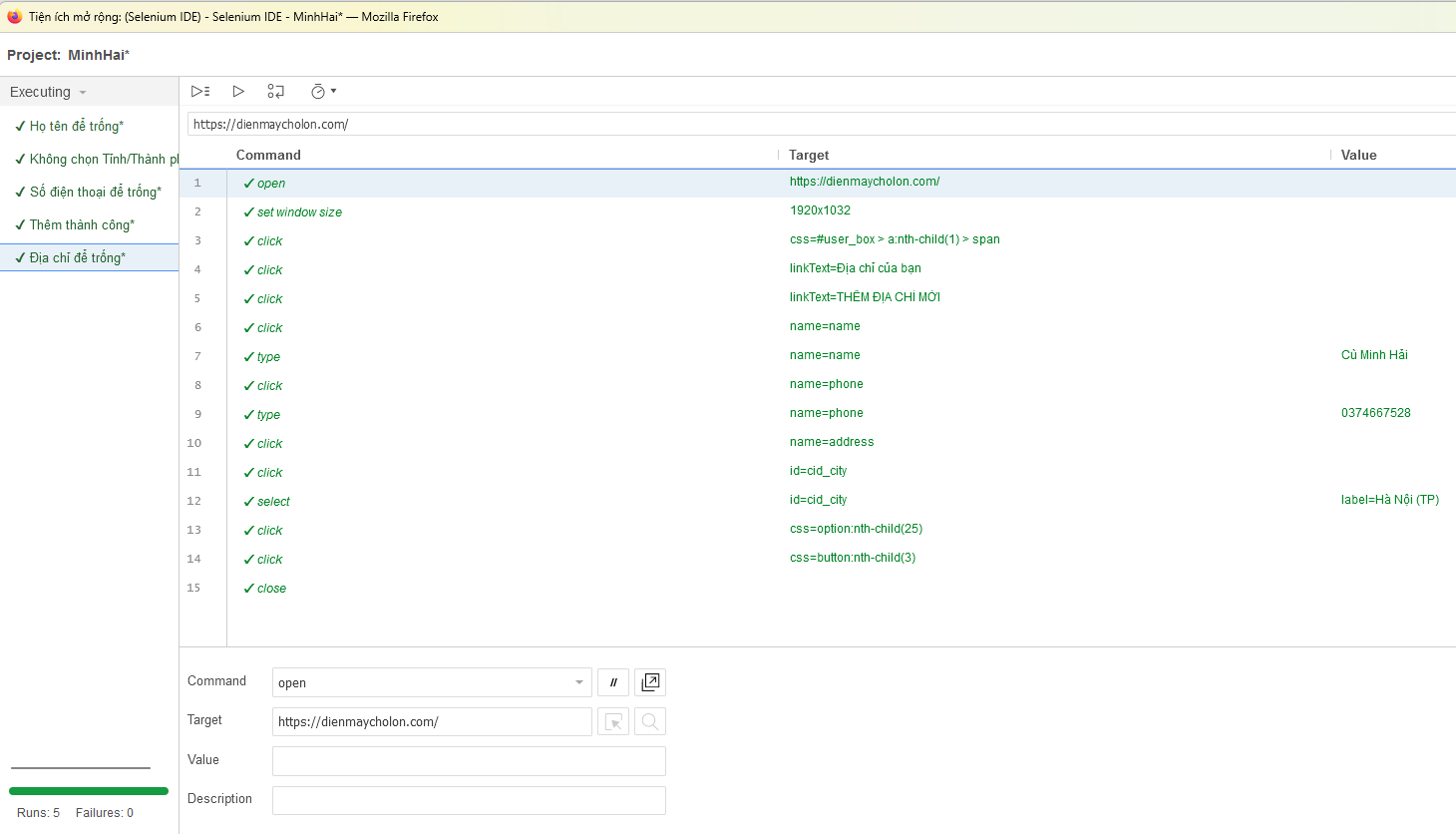
*Hình 3. 36: Kịch bản kiểm thử họ tên để trống*

* Số điện thoại để trống



*Hình 3. 37: Kịch bản kiểm thử số điện thoại để trống*

* Địa chỉ để trống



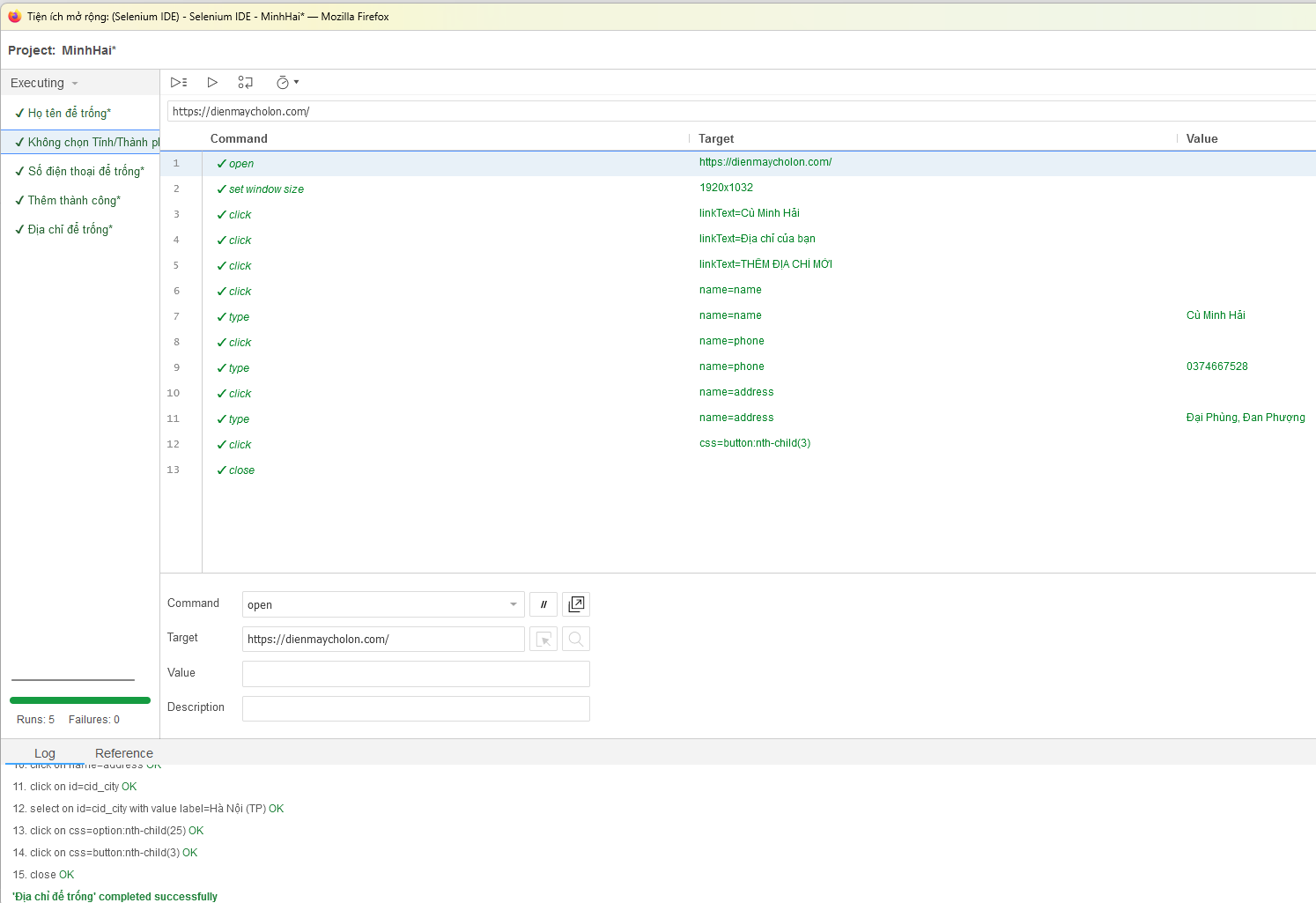
*Hình 3. 38: Kịch bản kiểm thử địa chỉ để trống*

* Không chọn Tỉnh/Thành phố



*Hình 3. 39: Kịch bản kiểm thử không chọn Tỉnh/Thành phố*

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



*Hình 3. 40: Tiến hành kiểm thử*

=> Hệ thống thông báo kiểm thử thành công.

Tương tự lặp lại các bước với các test case trên

=> Tỉ lệ thành công(PASSED) so với số TC đã lập: 100%.

# 

# PHẦN 3. KIẾN THỨC LĨNH HỘI VÀ BÀI HỌC KINH NGHIỆM

## 1. Kết quả đạt được

Trong quá trình thực hiện báo cáo, chúng em đã cố gắng hết sức tìm hiểu các kỹ thuật kiểm thử và sử dụng chúng để thực hiện kiểm thử. Bài báo cáo đã hoàn thành các mục đặt ra đó là: Tổng quan về kiểm thử phần mềm, Nghiên cứu các kỹ thuật kiểm thử, Tìm hiểu công cụ kiểm thử. Nhưng do thời gian có hạn nên bài báo cáo không thể tránh khỏi những thiếu sót. Rất mong nhận được sự thông cảm của thầy. Chúng em xin chân thành cảm ơn.

Những kết quả đạt được:

**Về kiểm thử phần mềm:**

* + Hiểu được kiểm thử phần mềm là gì?
  + Phân loại được kiểm thử phần mềm
  + Vai trò của kiểm thử phần mềm

**Về các kỹ thuật kiểm thử:**

Hiểu được kiểm thử Selenium?, cách áp dụng, mục tiêu, quy trình thực hiện, ưu và nhược điểm của chúng

**Về công cụ kiểm thử:**

* + Sử dụng công cụ kiểm thử Selenium tìm được các lỗi trong bài
  + Hoàn thiện sửa lỗi và chạy được chương trình 100%

## 2. Hướng phát triển

Nội dung trong báo cáo đã nêu lên được các nội dung của kỹ thuật kiểm thử. Chính vì vậy mà hướng nghiên cứu tiếp theo của chúng tôi đi sâu vào chương trình code và sử dụng công cụ kiểm thử tìm ra lỗi hệ thống. Việc kiểm thử được tiến hành dựa vào việc kiểm xem giải thuật, mã lệnh đã làm có đúng không. Mức test này thường yêu cầu các tester phải viết test case đầy đủ các nhánh trong code; khi test, sẽ set điều kiện và data để chạy vào đủ tất cả các nhánh trong giải thuật, đảm bảo thực hiện đầy đủ.

# TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | https://[www.guru99.com/fontend-testing.html.](http://www.guru99.com/fontend-testing.html) |
| [2] | javaTpoint, "Front End testing," [Online].  Available:https://[www.javatpoint.com/component-testin.](http://www.javatpoint.com/component-testing) |
| [3] | javaTpoint, "Selenium Tutorial," [Online]. Available: https://[www.javatpoint.com/selenium-tutorial.](http://www.javatpoint.com/selenium-tutorial) |