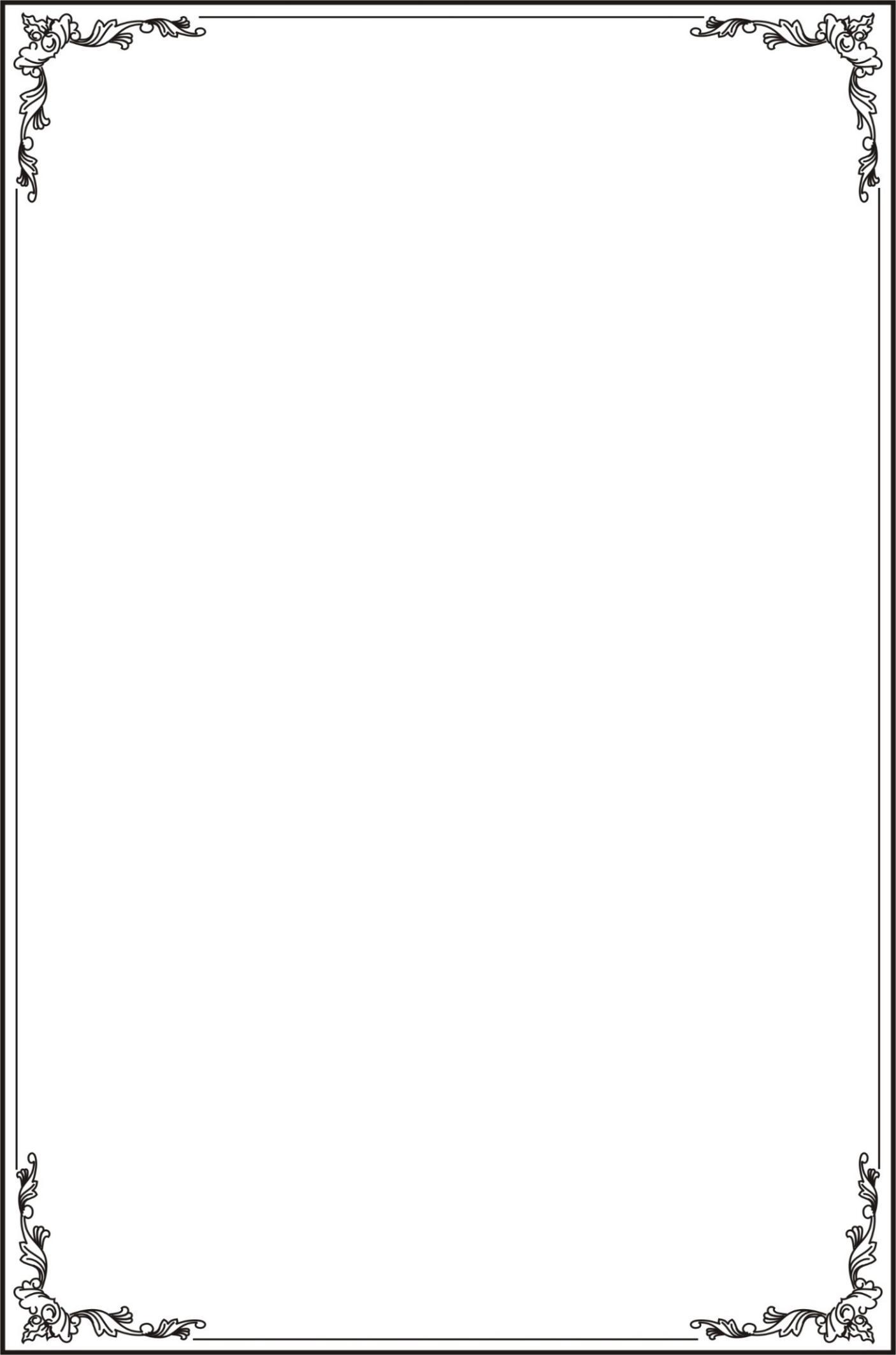
**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

-----



**BÀI TẬP LỚN**

**MÔN: KIỂM THỬ PHẦN MỀM**

**ĐỀ TÀI: TRÌNH BÀY VÀ ỨNG DỤNG KỸ THUẬT KIỂM THỬ CHỨC NĂNG CỦA ỨNG DỤNG WEB**

Giảng viên hướng dẫn : Ths. Hoàng Quang Huy

Lớp : 20221IT6084002

Nhóm : 3

Sinh viên thực hiện : Phạm Văn Cương - 2020600079

Đoàn Thị Thuý - 2020603329

Nguyễn Thị Minh - 2020603396

**Hà Nội, 2022**

MỤC LỤC

[**DANH MỤC HÌNH ẢNH** 4](#_heading=h.haapch)

[**1.**](#_heading=h.1gf8i83) **Tổng quan về kiểm thử(Minh)** 5

[1.1. Khái niệm về kiểm thử, các loại kiểm thử 5](#_heading=h.40ew0vw)

[1.1.1. Khái niệm kiểm thử 5](#_heading=h.2fk6b3p)

[1.1.2. Các loại kiểm thử 6](#_heading=h.upglbi)

[1.2. Các quy trình kiểm thử 11](#_heading=h.3ep43zb)

[1.3. Các kĩ thuật kiểm thử 14](#_heading=h.1tuee74)

[1.3.1. Kỹ thuật hộp đen 14](#_heading=h.4du1wux)

[1.3.2. Kỹ thuật hộp trắng 15](#_heading=h.2szc72q)

[**2.**](#_heading=h.3s49zyc) **Kiểm thử chức năng vào một trang web(Thuý)** 19

[2.1. Kiểm thử chức năng là gì 19](#_heading=h.279ka65)

[2.2. Các hoạt động cần làm trong kiểm thử chức năng của ứng dụng web 19](#_heading=h.meukdy)

[2.3. Cách thực hiện kiểm thử chức năng 21](#_heading=h.36ei31r)

[**3.**](#_heading=h.45jfvxd) **Tìm hiểu về công cụ Selenium (Cương)** 22

[3.1. Giới thiệu về công cụ Selenium 22](#_heading=h.2koq656)

[3.2. Cách cài đặt công cụ Selenium 22](#_heading=h.zu0gcz)

[**4.**](#_heading=h.2y3w247) **Ứng dụng công cụ kiểm thử Selenium IDE vào website và đánh giá** 27

[4.1. Lập kế hoạch(cả nhóm) 27](#_heading=h.1d96cc0)

[4.1.1. Giới thiệu 27](#_heading=h.3x8tuzt)

[4.1.1.1. Phạm vi 27](#_heading=h.2ce457m)

[4.1.1.2. Mục tiêu chất lượng 27](#_heading=h.rjefff)

[4.1.1.3. Nhân sự thực hiện 27](#_heading=h.3bj1y38)

[4.1.2. Phương pháp kiểm tra 28](#_heading=h.1qoc8b1)

[4.1.2.1. Tổng quan 28](#_heading=h.4anzqyu)

[4.1.2.2. Các quy trình kiểm thử 28](#_heading=h.2pta16n)

[4.1.2.3. Phân loại Lỗi 29](#_heading=h.14ykbeg)

[4.1.2.4. Tiêu chí dừng lại và tiếp tục kiểm thử 29](#_heading=h.3oy7u29)

[4.1.3. Nhu cầu tài nguyên & môi trường 29](#_heading=h.243i4a2)

[4.1.3.1. Công cụ kiểm thử 29](#_heading=h.j8sehv)

[4.1.3.2.](#_heading=h.338fx5o) Môi trường thực hiện test: 29

[4.2. Kịch bản kiểm thử và test case 29](#_heading=h.1idq7dh)

[4.2.1. Kiểm thử đăng ký(Minh) 29](#_heading=h.42ddq1a)

[4.2.2. Kiểm thử đăng nhập(Thuý) 31](#_heading=h.2hio093)

[4.2.3. Kiểm thử thêm tài liệu(Thuý) 33](#_heading=h.wnyagw)

[4.2.4. Kiểm thử tạo project(Minh) 34](#_heading=h.3gnlt4p)

[4.2.5. Kiểm thử sửa tài liệu(Cương) 36](#_heading=h.1vsw3ci)

[4.2.6. Kiểm thử xoá tài liệu(Cương) 37](#_heading=h.4fsjm0b)

[4.3. Tiến hành kiểm thử 38](#_heading=h.2uxtw84)

[4.3.1. Kiểm thử đăng ký(Minh) 38](#_heading=h.1a346fx)

[4.3.2. Kiểm thử đăng nhập(Thuý) 41](#_heading=h.1jlao46)

[4.3.3. Kiểm thử thêm tài liệu (Thuý) 43](#_heading=h.4kx3h1s)

[4.3.4. Kiểm thử tạo project(Minh) 44](#_heading=h.2eclud0)

[4.3.5. Kiểm thử sửa tài liệu(Cương) 46](#_heading=h.4cmhg48)

[4.3.6. Kiểm thử xoá tài liệu (Cương) 47](#_heading=h.16x20ju)

[**5.**](#_heading=h.l7a3n9) **Báo cáo, đánh giá kết quả kiểm thử** 48

[**6.**](#_heading=h.356xmb2) **Tài liệu tham khảo** 49

# **DANH MỤC HÌNH ẢNH**

[Hình 1.3.2.1: Các loại code coverage 17](#_heading=h.184mhaj)

[Hình 1.3.2.1: Các bước thức hiện kiểm thử chức năng 21](#_heading=h.3j2qqm3)

[Hình 1.3.2.1: Giao diện Selenium IDE 24](#_heading=h.3whwml4)

[Hình 1.3.2.2: Giao diện vùng làm việc 25](#_heading=h.2bn6wsx)

[Hình 1.3.2.3: Các thanh công cụ của Selenium IDE 26](#_heading=h.qsh70q)

[Hình 4.3.1.1: Kết quả test trường hợp dữ liệu cung cấp khi đăng ký là đủ và đúng 38](#_heading=h.sqyw64)

[Hình 4.3.1.2: Kết quả test trường hợp dữ liệu cung cấp để đăng ký thiếu tên user 39](#_heading=h.2981zbj)

[Hình 4.3.1.3: Kết quả test trường hợp dữ liệu cung cấp khi đăng ký thiếu email 39](#_heading=h.odc9jc)

[Hình 4.3.1.4: Kết quả test trường hợp dữ liệu cung cấp khi đăng ký thiếu mật khẩu 40](#_heading=h.38czs75)

[Hình 4.3.1.5: Kết quả test trường hợp dữ liệu cung cấp khi đăng ký không click vào box Agree the terms and policy 40](#_heading=h.kgcv8k)

[Hình 4.3.1.6: Kết quả test trường hợp dữ liệu cung cấp khi đăng ký nhập trùng Username đã đăng ký 41](#_heading=h.47hxl2r)

[Hình 4.3.2.1: Kết quả test trường hợp dữ liệu cung cấp để đăng nhập là đủ và đúng 41](#_heading=h.2mn7vak)

[Hình 4.3.2.2: 2 Kết quả test trường hợp dữ liệu cung cấp để đăng nhập thiếu password 42](#_heading=h.11si5id)

[Hình 4.3.2.3: Kết quả test trường hợp dữ liệu cung cấp để đăng nhập thiếu username 42](#_heading=h.3ls5o66)

[Hình 4.3.2.4: Kết quả test trường hợp dữ liệu cung cấp để đăng nhập sai password 43](#_heading=h.20xfydz)

[Hình 4.3.3.1: Kết quả test trường hợp dữ liệu cung cấp để thêm tài liệu file hợp lệ 43](#_heading=h.302dr9l)

[Hình 4.3.3.2: Kết quả test trường hợp dữ liệu cung cấp để thêm tài liệu file không hợp lệ 44](#_heading=h.1f7o1he)

[Hình 4.3.3.3: Kết quả test trường hợp dữ liệu cung cấp để thêm tài liệu không thêm file 44](#_heading=h.3z7bk57)

[Hình 4.3.4.1: Kết quả test trường hợp dữ liệu cung cấp khi tạo project là đủ và đúng 45](#_heading=h.thw4kt)

[Hình 4.3.4.2: Kết quả test trường hợp dữ liệu cung cấp khi tạo project thiếu tên project 45](#_heading=h.3dhjn8m)

[Hình 4.3.4.3: Kết quả test trường hợp dữ liệu cung cấp khi tạo project mà tên project bị trùng 46](#_heading=h.1smtxgf)

[Hình 4.3.5.1: Kết quả test trường hợp dữ liệu cung cấp khi sửa tài liệu 46](#_heading=h.2rrrqc1)

[Hình 4.3.6.1: Kết quả test trường hợp xoá tài liệu 47](#_heading=h.3qwpj7n)

[Hình 4.3.6.2: Kết quả test trường hợp bấm huỷ không xoá tài liệu 47](#_heading=h.261ztfg)

# **Tổng quan về kiểm thử(Minh)**

## Khái niệm về kiểm thử, các loại kiểm thử

### Khái niệm kiểm thử

**\*** **Kiểm thử là gì?**

- Kiểm thử tồn tại trước và sau khi quá trình thực thi kiểm thử được bắt đầu. Hoạt động bao gồm việc lên kế hoạch (planning) và kiểm soát (control), lựa chọn điều kiện kiểm thử, thiết kế và thực thi test case, kiểm tra kết quả, xác định khi nào thì dừng quá trình kiểm thử, báo cáo trong quá trình kiểm thử và tổng hợp hoặc hoàn thành việc kiểm thử trước khi giai đoạn kiểm thử (test phase) được hoàn thành. Kiểm thử cũng bao gồm kiểm tra (review) tài liệu (có thể gồm source code) và tiền hành phân tích.

- Kiểm thử động (dynamic testing) và kiểm thử tĩnh (static testing) thường được sử dụng để đạt được mục đích, cung cấp thông tin để nâng cao hệ thống trong quá trình kiểm thử và phát triển.

**\*** **Kiểm thử phải tuân theo các mục đích:**

- Tìm kiếm lỗi/ Finding defects

- Đảm bảo được một mức độ chất lượng/ Gaining confidence about the level of quality

- Cung cấp thông tin để đưa ra quyết định/ Providing information for decision-making

- Ngăn ngừa lỗi/ Preventing defects

**\*** **Các nguyên tắc kiểm thử (7 testing principles):**

- Kiểm tra sự hiện diện của lỗi/ Testing shows presence of defect

- Kiểm tra toàn bộ là không thể/ Exhaustive testing is impossible

- Thực hiện kiểm thử sớm/ Early testing

- Phân loại lỗi/ Defect clustering

- Phương pháp thuốc trừ sâu/ Pesticide paradox

- Kiểm thử phụ thuộc vào bối cảnh/ Testing is context dependent

- Sự vắng mặt của lỗi/ Absence of error fallacy

**\*** **Tại sao kiểm thử phần mềm là cần thiết?**

- **Vai trò của kiểm thử trong phát triển phần mềm, bảo trì và vận hành:**

+ Kiểm thử nghiêm ngặt hệ thống và tài liệu có thể giúp giảm thiểu những vấn đề rủi ro xảy ra trong quá trình vận hành và góp phần nâng cao chất lượng của hệ thống phần mềm, nếu như các lỗi được tìm thấy và sửa chữa trước khi hệ thống được vận hành thực tế.

+ Kiểm thử phần mềm có được yêu cầu để đạt được tính hợp pháp và đảm bảo hợp đồng, hoặc những tiêu chuẩn trong ngành công nghiệp đặc biệt.

- **Kiểm thử và chất lượng:**

+ Kiểm thử mang lại sự tự tin về chất lượng của phần mềm nếu nó tìm thấy một vài hoặc không tìm thấy lỗi. Kiểm thử đúng sẽ giảm thiểu được tổng thể mức độ rủi ro của hệ thống. Khi quá trình kiểm thử tìm thấy lỗi, chất lượng của hệ thống phần mềm được nâng cao sau khi những lỗi đó được sửa chữa.

- **Kiểm thử bao nhiêu là đủ ?**

+ Việc quyết định kiểm thử bao nhiêu là đủ phụ thuộc vào mức độ của rủi ro, bao gồm kỹ thuật, độ an toàn, rủi ro trong kinh doanh, và hạn mức của dự án như là thời gian và ngân sách.

+ Kiểm thử nên cung cấp đủ thông tin để các bên liên quan có thể quyết định về việc bàn giao phần mềm hoặc hệ thống đã qua kiểm thử, cho các bước phát triển tiếp theo hay bàn giao cho khách hàng

### Các loại kiểm thử

**\*** **Kiểm thử chức năng (functional testing)**

- Kiểm thử chức năng là một loại kiểm thử hộp đen (black box) và test case của nó được dựa trên đặc tả của ứng dụng phần mềm/thành phần đang test. Các chức năng được test bằng cách nhập vào các giá trị và kiểm tra kết quả đầu ra, ít quan tâm đến cấu trúc bên trong của ứng dụng.

- Kiểm thử chức năng có thể được thực hiện từ 2 góc nhìn: dựa trên yêu cầu và dựa trên quy trình nghiệp vụ.

+ Dựa trên yêu cầu:

* Sử dụng các đặc tả kỹ thuật của các yêu cầu chức năng để làm cơ sở cho việc test các thiết kế.
* Nội dung của các yêu cầu có thể làm các mục kiểm thử ban đầu hoặc sử dụng nó như là một danh sách các mục kiểm thử hoặc không kiểm thử.
* Dựa theo yêu cầu để phân mức độ ưu tiên trong quá trình kiểm thử. Cần ưu tiên các yêu cầu có mức độ rủi ro cao.

+ Dựa trên quy trình nghiệp vụ:

* Các quy trình nghiệp vụ mô tả các kịch bản scenarios liên quan đến các nghiệp vụ hằng ngày của hệ thống
* các usecase được bắt nguồn phát triển theo hướng đối tượng nhưng hiện tại phổ biến trong nhiều trong các vòng đời phát triển.
* Lấy các quy trình nghiệp vụ làm điểm khởi đầu, các quy trình nghiệp vụ xuất phát từ các nhiệm vụ được thực hiện bởi người dùng.
* Các use case là một cơ sở hữu ích cho các testcase từ góc độ nghiệp vụ.

- Kiểm thử chức năng bao gồm 5 bước:

+ Xác định các chức năng mà phần mềm mong muốn sẽ thực hiện.

+ Tạo ra các dữ liệu đầu vào dựa trên các tài liệu đặc tả kỹ thuật của các chức năng.

+ Xác định 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.

- Các loại kiểm thử chức năng:

+ Kiểm thử đơn vị (Unit Testing)

+ Kiểm thử khói (Smoke Testing - check nhanh xem hệ thống có khởi động được hay không)

+ Kiểm thử độ tỉnh táo (Sanity Testing - check nhanh xem sau khi sửa đổi thì function có hoạt động như mong muốn hay không)

+ Kiểm thử giao diện (Interface Testing)

+ Kiểm thử tích hợp (Integration Testing)

+ Kiểm thử hệ thống (Systems Testing)

+ Kiểm thử hồi quy (Regression Testing)

+ Kiểm thử chấp nhận (Acceptance testing)

**\*** **Kiểm thử phi chức năng (non-functional testing).**

- Kiểm thử phi chức năng cùng giống kiểm thử chức năng ở chỗ là thực hiện được ở mọi cấp độ kiểm thử, Kiểm thử phi chức năng xem xét các hành vi bên ngoài của phần mềm.

- Kiểm thử phi chức năng bao gồm:

+ Kiểm thử hiệu năng (performance testing).

+ Kiểm thử khả năng chịu tải (load testing).

+ Kiểm thử áp lực (stress testing).

+ Kiểm thử khả năng sử dụng (usability testing).

+ Kiểm thử bảo trì (maintainability testing).

+ Kiểm thử độ tin cậy (reliability testing)

+ Kiểm thử tính tương thích (portability testing)

- Những đặc điểm phụ tương ứng:

+ Độ tin cậy (reliability): được xác định rõ hơn từ các đặc trưng phụ đã được tính toán cẩn thận (độ bền), khả năng chịu lỗi (fault tolerance), phục hồi (recoverability) và tuân thủ (compliance).

+ Khả năng sử dụng (usability): được chia thành các đặc trưng dễ hiểu, khả năng học hỏi (learnability), khả năng hoạt động (operability), sự thu hút (attractiveness) và tính tuân thủ (compliance).

+ Tính hiệu quả (efficiency): được chia thành hành vi về thời gian (hiệu suất), sử dụng tài nguyên (resource utilization) và tuân thủ (compliance).

+ Khả năng bảo trì (maintainability): bao gồm 5 đặc điểm phụ: phân tích, khả năng thay đổi, tính ổn định, khả năng kiểm tra và tuân thủ.

+ Tính tương thích (portability): bao gồm 5 đặc điểm phụ: khả năng thích ứng, khả năng cài đặt, cùng tồn tại, khả năng thay thế và tuân thủ.

**\*** **Kiểm thử cấu trúc/kiến trúc phần mềm (structural testing).**

- Kiểm thử cấu trúc có thể xảy ra ở bất kỳ mức độ kiểm thử nào, được áp dụng chủ yếu ở kiểm thử thành phần, tích hợp.

- Phương pháp kiểm thử cấu trúc cũng có thể áp dụng ở các mức độ như kiểm thử tích hợp hệ thống hoặc kiểm thử chấp nhận.

- Kỹ thuật kiểm thử cấu trúc được sử dụng tốt nhất sau các kỹ thuật dựa trên các đặc điểm kỹ thuật (specification-based). Giúp đo lường kỹ lưỡng kiểm thử thông qua đánh giá độ bao phủ của loại cấu trúc.

- Độ bao phủ là phạm vi mà một cấu trúc đã được thực hiện bởi một bộ kiểm thử, tính theo phần trăm của các mục đã được bao phủ. Nếu độ bao phủ không phải là 100% các kiểm thử sẽ được thiết kế để kiểm tra các mục đã bị bỏ lỡ để tăng độ bao phủ.

- Các kỹ thuật được sử dụng để kiểm thử cấu trúc là: các kỹ thuật hộp trắng và các mô hình luồng điều khiển (Control flow models).

**\*** **Kiểm thử xác nhận (confirmation testing) và kiểm thử hồi quy (regression testing)**

- Kiểm thử xác nhận.

+ Sau khi một lỗi được phát hiện và sửa chữa, phần mềm được kiểm thử lại để xác nhận lỗi ban đầu đã được khắc phục gọi là kiểm thử xác nhận (Confirmation testing).

+ Khi thực hiện kiểm thử xác nhận phải đảm bảo rằng các thử nghiệm được thực hiện giống như lần đầu tiên sử dụng, sử dụng các inputs, dữ liệu và môi trường giống nhau.

- Kiểm thử hồi quy:

+ Mục đích của kiểm thử hồi quy là xác minh rằng sửa đổi trong phần mềm hoặc môi trường không gây ra các phản ứng phụ không mong muốn và hệ thống vẫn đáp ứng các yêu cầu.

+ Kiểm thử hồi quy là các kiểm thử lặp đi lặp lại của một chương trình đã được kiểm thử, sau khi sửa đổi.

+ Kiểm thử hồi quy được thực hiện bất cứ khi nào trong phần mềm hoặc là kết quả của các bản sửa lỗi, chức năng mới được thay đổi

+ Kiểm thử hồi quy dựa vào các bộ test case. Khi thêm chức năng mới thì phải thêm các testcase mới hoặc là các chức năng cũ được thay đổi hay xóa bỏ thì test case cũng phải được thay đổi hoặc xóa bỏ.

+ Kiểm thử hồi quy có thể được thực hiện tại tất cả mức độ kiểm thử, bao gồm kiểm thử chức năng, phi chức năng và kiểm thử cấu trúc.

**\*** **Kiểm thử bảo trì (Maintenance testing).**

- Phân tích tác động và kiểm thử hồi quy:

+ Thông thường kiểm thử bảo trì gồm 2 phần: kiểm thử các thay đổi và Kiểm thử hồi quy để cho thấy phần còn lại của hệ thống không bị ảnh hưởng bởi công việc bảo trì.

+ Hoạt động chính và quan trọng trong việc kiểm thử bảo trì là việc phân tích các tác động. Từ việc phân tích sẽ quyết định được những phần nào của hệ thống có thể bị ảnh hưởng không mong muốn.

+ Phân tích rủi ro sẽ giúp quyết định được nơi cần tập trung kiểm thử hồi quy.

- Khởi động cho kiểm thử bảo trì:

+ Kiểm thử bảo trì được thực hiện trên hệ thống đã tồn tại và được thực hiện khi có sự thay đổi, di chuyển hoặc rút lui của phần mềm hoặc hệ thống.

+ Kiểm thử bảo trì cho việc thay đổi: Các cải tiến bao gồm thay đổi tăng theo kế hoạch, khắc phục những thay đổi khẩn cấp và thay đổi môi trường.

+ Kiểm thử bảo trì cho sự chuyển đổi: Bao gồm kiểm tra hoạt động của môi trường mới, các phần mềm đã thay đổi. Kiểm thử di chuyển (kiểm thử chuyển đổi) cũng cần thiết khi dữ liệu từ một ứng dụng khác sẽ được di chuyển vào hệ thống đang được bảo trì.

+ Kiểm thử bảo trì đối với hệ thống đã ngừng hoạt động: bao gồm kiểm thử việc chuyển đổi dữ liệu hoặc lưu trữ, nếu cần lưu trữ dữ liệu lâu dài.

+ Từ quan điểm của việc chuyển đổi thì có 2 loại:

* Chuyển đổi theo kế hoạch bao gồm: Chuyển đổi hoàn thiện (phần mềm thích nghi được với mong muốn người dùng), Chuyển đổi thích nghi (phần mềm thích nghi được với sự thay đổi của môi trường như phần cứng mới, phần mềm hệ thống mới), Chuyển đổi điều chỉnh theo kế hoạch (sửa chữa lỗi).
* Những chuyển đổi bột phát không thể lên kế hoạch được: đối với những lỗi như thế này cần phân tích rủi ro của hệ thống hoạt động để xác định chức năng hoặc chương trình gây lỗi.

## Các quy trình kiểm thử

**\*** **Bước 1: Lập kế hoạch và kiểm soát việc kiểm thử**

Mục đích: Nhằm chỉ định và mô tả các loại kiểm tra sẽ được triển khai và thực hiện. Được chia làm 2 hoạt động:

**Thứ nhất: Lập kế hoạch kiểm thử:**

- Xác định phạm vi, rủi ro cũng như mục đích của hoạt động kiểm thử

- Xác định các cách tiếp cận kiểm thử

- Xác định chiến lược kiểm thử. Chiến lược kiểm thử mô tả các thành phần kiểm thử cần có trong một chu kỳ phát triển phần mềm chẳng hạn như: các mục tiêu kiểm thử, các phương pháp kiểm thử, tổng thời gian và nguồn lực yêu cầu cho các dự án cũng như các môi trường test. (Lưu ý: Chiến lược kiểm thử thường được tạo ra bởi PM, TL)

- Xác định các nguồn lực cần có cho kiểm thử như: nhân lực, phần cứng, phần mềm, môi trường test v.v

- Lên lịch cho các hoạt động phân tích và thiết kế các trường hợp kiểm thử, thực thi kiểm thử cũng như đánh giá kết quả kiểm thử.

- Xác định các tiêu chí kết thúc việc kiểm thử (exit criteria) chẳng hạn như tỉ lệ độ bao phủ của test case, số lượng bug tìm được, độ nghiêm trọng của những con bug tìm được.

**Thứ hai: Kiểm soát kiểm thử:**

- Đo lường và phân tích các kết quả của hoạt động kiểm thử

- Theo dõi và ghi lại tiến độ, độ bao phủ cũng như các tiêu chí kết thúc kiểm thử

- Cung cấp thông tin về kiểm thử

- Tiến hành các hành động khắc phục nếu cần thiết.

- Đưa ra quyết định.

**\*** **Bước 2: Phân tích và Thiết kế**

Mục đích: Nhằm chỉ định các test case và các bước kiểm tra chi tiết cho mỗi phiên PM.

Hoạt động phân tích và thiết kế kiểm thử có các nhiệm vụ chủ yếu sau đây:

- Rà soát các yêu cầu cần thiết trước khi tiến hành kiểm thử như tài liệu đặc tả, tài liệu thiết kế, tài liệu giao diện, v.v

- Xác định các điều kiện kiểm thử

- Thiết kế test case

- Đánh giá tính khả thi trong việc kiểm thử của yêu cầu cũng như của hệ thống.

- Chuẩn bị môi trường test cũng như xác định các yêu cầu về cơ sở hạ tầng và các công cụ kiểm thử tương ứng.

- Giai đoạn thiết kế test là hết sức quan trọng, nó đảm bảo tất cả các tình huống kiểm tra hết tất cả các yêu cầu

**\*** **Bước 3: Thực thi test**

Mục đích: Thực hiện các bước kiểm tra đã thiết kế và ghi nhận kết quả. Chia thành 2 hoạt động chính là: thực hiện test và chạy test

Thứ nhất: Việc thực hiện test có nhiệm vụ chủ yếu sau đây:

- Chuẩn bị test data

- Thiết kế và phân loại các trường hợp kiểm thử dựa theo độ ưu tiên của từng trường hợp kiểm thử

- Tự động hóa cho các trường hợp kiểm thử nếu thấy cần thiết

Thứ hai: Hoạt động chạy test có nhiệm vụ chủ yếu sau đây:

- Chạy các test case theo các bước đã định ra trước đó

- Chạy lại các case bị failed trước đó để xác nhận là case đó đã được sửa

- So sánh kết quả ghi nhận được khi thực thi với kết quả mong đợi

- Đánh giá kết quả kiểm thử (Passed/Failed) cho các trường hợp kiểm thử

- Viết báo cáo lỗi cho những trường hợp kết quả ghi nhận được và kết quả mong đợi không giống nhau

- Bước này thường không bắt buộc trong các loại và mức kiểm tra, chỉ yêu cầu trong những trường hợp đặc thù cần thiết kế, tạo ra các test script có khả năng chạy trên máy tính giúp tự động hóa việc thực thi các bước kiểm tra đã định nghĩa ở các bước thiết kế test

**\*** **Bước 4: Đánh giá kết quả thực thi và báo cáo kết quả**

Mục đích: Đánh giá toàn bộ quá trình kiểm tra bao gồm xem xét và đánh giá kết quả kiểm tra lỗi, chỉ định các yêu cầu thay đổi và tính toán số liệu liên quan, đến quá trình kiểm tra.

Các tiêu chí đánh giá kết quả thực thi này bao gồm:

- Số lượng test case tối đa được thực thi Passed

- Tỷ lệ lỗi giảm xuống dưới mức nhất định

- Khi đến deadline

Việc kiểm thử chỉ kết thúc khi:

- Đối chiếu kết quả thực thi test case so với các tiêu chí kết thúc kiểm thử được định ra trong lúc lập kế hoạch kiểm thử

- Từ đó, đánh giá xem liệu có cần phải test thêm hay điều chỉnh các tiêu chí kết thúc kiểm thử trong bản kế hoạch

- Viết báo cáo tóm tắt hoạt động kiểm thử cũng như kết quả kiểm thử cho các bên liên quan.

**\***  **Bước 5: Đóng hoạt động kiểm thử**

Mục đích: Kết thúc hoạt động kiểm thử và phần mềm sẵn sàng được giao cho khách hàng.

Ngoài ra, chúng ta cũng thường kết thúc kiểm thử với một trong những lí do sau:

- Khi tất cả các thông tin đã được thu thập đầy đủ cho hoạt động kiểm thử

- Khi dự án bị hủy bỏ.

- Khi một mục tiêu nào đó đạt được.

- Khi hoạt động bảo trì hay cập nhật hệ thống hoàn tất.

Hoạt động đóng kiểm thử bao gồm:

- Kiểm tra lại đã giao đầy đủ cho khách hàng những phần đã cam kết từ đầu

- Kiểm tra lại các lỗi nghiêm trọng đã được fix tương ứng

- Đóng gói các tài liệu kiểm thử, kịch bản kiểm thử, môi trường test v.v để dùng cho những mục đích /dự án sau này

- Đánh giá quá trình kiểm thử cũng như rút ra bài học kinh nghiệm cho những dự án trong tương lai.

## Các kĩ thuật kiểm thử

### Kỹ thuật hộp đen

Kiểm thử hộp đen (Black box testing) là một phương pháp kiểm thử phần mềm mà việc kiểm tra các chức năng của một ứng dụng không cần quan tâm vào cấu trúc nội bộ hoặc hoạt động của nó. Mục đích chính của kiểm thử hộp đen chỉ là để xem phần mềm có hoạt động như dự kiến trong tài liệu yêu cầu và liệu nó có đáp ứng được sự mong đợi của người dùng hay không.

Quy trình kiểm thử hộp đen:

- Phân tích đặc tả về các yêu cầu chức năng mà TPPM cần thực hiện.

- Dùng 1 kỹ thuật định nghĩa các testcase xác định (sẽ giới thiệu sau) để định nghĩa các testcase. Định nghĩa mỗi testcase là xác định 3 thông tin sau :

+ Giá trị dữ liệu nhập để TPPM xử lý (hoặc hợp lệ hoặc không hợp lệ).

+ Trạng thái của TPPM cần có để thực hiện testcase.

+ Giá trị dữ liệu xuất mà TPPM phải tạo được.

- Kiểm thử các testcase đã định nghĩa.

- So sánh kết quả thu được với kết quả kỳ vọng trong từng testcase, từ đó lập báo cáo về kết quả kiểm thử.

- Đặc điểm

+ Đây là kiểu kiểm thử thành phần phần mềm và chỉ dựa vào các thông tin đặc tả về yêu cầu, chức năng của các thành phần phần mềm tương ứng.

+ Việc kiểm thử được thực hiện bên ngoài, không liên quan đến lập trình viên hay các nhà phát triển phần mềm. Vì thế người kiểm thử cũng không cần thiết phải biết về cấu trúc bên trong của phần mềm cũng như các kiến thức về lập trình.

+ Mức test này thường yêu cầu các tester phải viết test case đầy đủ trước khi test; Các bước tiến hành test khá đơn giản, chỉ cần thực hiện theo các mô tả trong test case, thực hiện nhập dữ liệu vào, đợi kết quả trả về và so sánh với kết quả dự kiến trong test case.

- Có 8 kỹ thuật phổ biến và có nhiều ưu điểm nhất:

+ Phân vùng tương đương (equivalence partitioning)

+ Phân tích giá trị biên(boundary value analysis)

+ Bảng quyết định (decision tables)

+ Kiểm thử chuyển đổi trạng thái (state transition testing).

+ Kỹ thuật kiểm thử các bộ n thần kỳ (Pairwise)

+ Kỹ thuật phân tích vùng miền (Domain analysis)

+ Kỹ thuật dựa trên đặc tả Use case ( Use case)

+ Kỹ thuật dùng lược đồ quan hệ nhân quả ( Cause- Effect Diagram)

### Kỹ thuật hộp trắng

Kiểm thử hộp trắng là loại thử nghiệm được thực hiện để kiểm tra cấu trúc code. Loại thử nghiệm này đòi hỏi người test phải có kiến thức về code. Do đó, phần lớn là do các lập trình viên, nhà phát triển phần mềm thực hiện.

- Đặc điểm:

+ Kiểm thử hộp trắng quan tâm đến việc hệ thống vận hành như thế nào chứ không phải chức năng của hệ thống. Vì nó dựa vào những thuật toán cụ thể, vào những cấu trúc dữ liệu bên trong của thành phần phần mềm.

+ Trong kỹ thuật kiểm thử này, đòi hỏi người tester phải có kiến thức và kỹ năng nhất định về ngôn ngữ lập trình được dùng, hiểu thuật toán trong phần mềm, để có thể hiểu được chi tiết về đoạn code cần kiểm thử .

+ 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 thuật toán, đảm bảo thực hiện đầy đủ.

+ Kiểm thử hộp trắng được dùng để đo độ bao phủ( coverage ) và kiểm thử thiết kế(Design test).

+ Bao phủ kiểm thử(test coverage ) là tỷ lệ (tính theo %) testcase đã được thực hiện trên tổng số test case cần thiết cho phần mềm. Nếu tỉ lệ này càng cao thì phần mềm càng được test kỹ. Mặc dù việc đảm bảo phần mềm có test coverage là 100% nhưng không có nghĩa là 100% các trường hợp được kiểm thử. Độ bao phủ được tính bằng công thức sau:

**Coverage = ( Number of coverage items exercised / Total number of coverage items) \* 100%**

Có nghĩa là:

**Độ bao phủ = Số item bao phủ được thực hiện/ Tổng số item bao phủ \* 100%**

- Có 2 loại điều kiện bao phủ cơ bản: Bao phủ code(Code coverage), Bao phủ chức năng(function coverage).

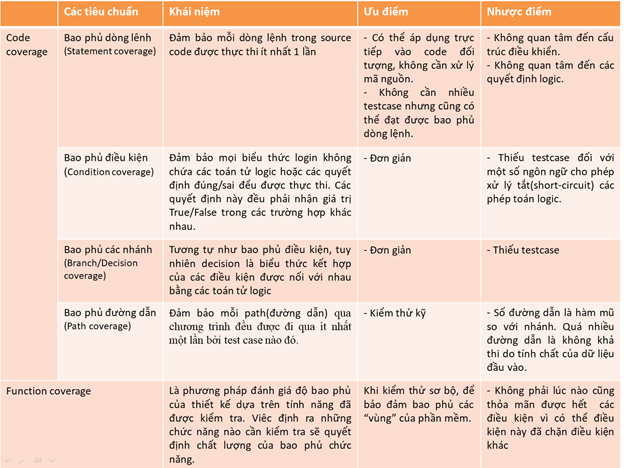
- Code coverage là phương pháp thống kê dựa vào số lượng code được kiểm tra. Trong code coverage có thể phân chia thành các loại sau:

+ Bao phủ dòng lệnh(Statement coverage)

+ Bao phủ các nhánh (Branch/Decision coverage)

+ Bao phủ điều kiện (Condition coverage)

+ Bao phủ đường dẫn (Path coverage)



*Hình 1.3.2.1: Các loại code coverage*

- Kiểm thử hộp trắng có 2 hoạt động chính: kiểm thử luồng điều khiển(control flow testing) và kiểm thử luồng dữ liệu(data flow testing).

Kiểm thử luồng điều khiển: là một chiến lược trong kiểm thử cấu trúc nó sử dụng luồng điều khiển của chương trình như một mô hình. Nó dựa vào việc lựa chọn một tập các đường dẫn kiểm tra thông qua chương trình. Tập các đường dẫn đã chọn được sử dụng để đạt được một số đo nhất định của việc kiểm thử cẩn thận.

Kiểm thử luồng dữ liệu: là một nhóm các chiến lược kiểm thử dựa trên việc chọn các đường dẫn thông qua luồng điều khiển của chương trình để khám phá chuỗi các sự kiện liên quan đến trạng thái của các biến hoặc các đối tượng dữ liệu. Kiểm tra Dataflow tập trung vào các điểm mà tại đó các biến nhận giá trị và các điểm mà tại đó các giá trị này được sử dụng.

- Ưu điểm

+Test có thể bắt đầu ở giai đoạn sớm hơn, không cần phải chờ đợi cho GUI để có thể test

+ Test kỹ càng hơn, có thể bao phủ hầu hết các đường dẫn

+ Thích hợp trong việc tìm kiếm lỗi và các vấn đề trong mã lệnh

+ Cho phép tìm kiếm các lỗi ẩn bên trong

+ Các lập trình viên có thể tự kiểm tra

+ Giúp tối ưu việc mã hoá

+ Do yêu cầu kiến thức cấu trúc bên trong của phần mềm, nên việc kiểm soát lỗi tối đa nhất.

- Nhược điểm

+ Vì các bài kiểm tra rất phức tạp, đòi hỏi phải có các nguồn lực có tay nghề cao, với kiến thức sâu rộng về lập trình và thực hiện.

+ Maintenance test script có thể là một gánh nặng nếu thể hiện thay đổi quá thường xuyên.

+ Vì phương pháp thử nghiệm này liên quan chặt chẽ với ứng dụng đang được test, nên các công cụ để phục vụ cho mọi loại triển khai / nền tảng có thể không sẵn có.

# **Kiểm thử chức năng vào một trang web(Thuý)**

## Kiểm thử chức năng là gì

Kiểm tra chức năng là một loại kiểm thử phần mềm nhằm xác nhận hệ thống phần mềm theo các yêu cầu / thông số kỹ thuật chức năng. Mục đích của Kiểm thử chức năng là kiểm tra từng chức năng của ứng dụng phần mềm, bằng cách cung cấp đầu vào thích hợp, xác minh đầu ra so với các yêu cầu Chức năng.

Kiểm tra chức năng chủ yếu liên quan đến kiểm tra hộp đen và nó không quan tâm đến mã nguồn của ứng dụng.

Kiểm tra chức năng của một trang web là một quá trình bao gồm một số tham số kiểm tra như giao diện người dùng, API, kiểm tra cơ sở dữ liệu, kiểm tra bảo mật, kiểm tra máy khách và máy chủ và các chức năng cơ bản của trang web.

## Các hoạt động cần làm trong kiểm thử chức năng của ứng dụng web

1. Kiểm tra tất cả các liên kết trong các trang web của bạn đang hoạt động bình thường và đảm bảo rằng không có liên kết nào bị hỏng. Các liên kết được kiểm tra sẽ bao gồm:

* Outgoing links
* Internal links
* Anchor Links
* MailTo Links

1. Các biểu mẫu kiểm tra đang hoạt động như mong đợi. Điều này sẽ bao gồm:

* Kiểm thử kịch bản trên biểu mẫu đang hoạt động như mong đợi. Ví dụ - nếu người dùng không điền vào trường bắt buộc trong biểu mẫu, thông báo lỗi sẽ được hiển thị.
* Kiểm thử các giá trị mặc định đang được điền
* Sau khi được gửi, dữ liệu trong các biểu mẫu được gửi đến cơ sở dữ liệu trực tiếp hoặc được liên kết với một địa chỉ email đang hoạt động
* Biểu mẫu được định dạng tối ưu để dễ đọc

1. Cookie thử nghiệm đang hoạt động như mong đợi. Cookie là các tệp nhỏ được các trang web sử dụng để chủ yếu ghi nhớ các phiên hoạt động của người dùng, do đó bạn không cần phải đăng nhập mỗi khi truy cập một trang web. Kiểm tra cookie sẽ bao gồm:

* Các cookie (phiên) kiểm tra sẽ bị xóa khi bộ nhớ cache bị xóa hoặc khi chúng hết hạn.
* Xóa cookie (phiên) và kiểm tra xem thông tin đăng nhập được yêu cầu khi bạn truy cập trang web lần sau.

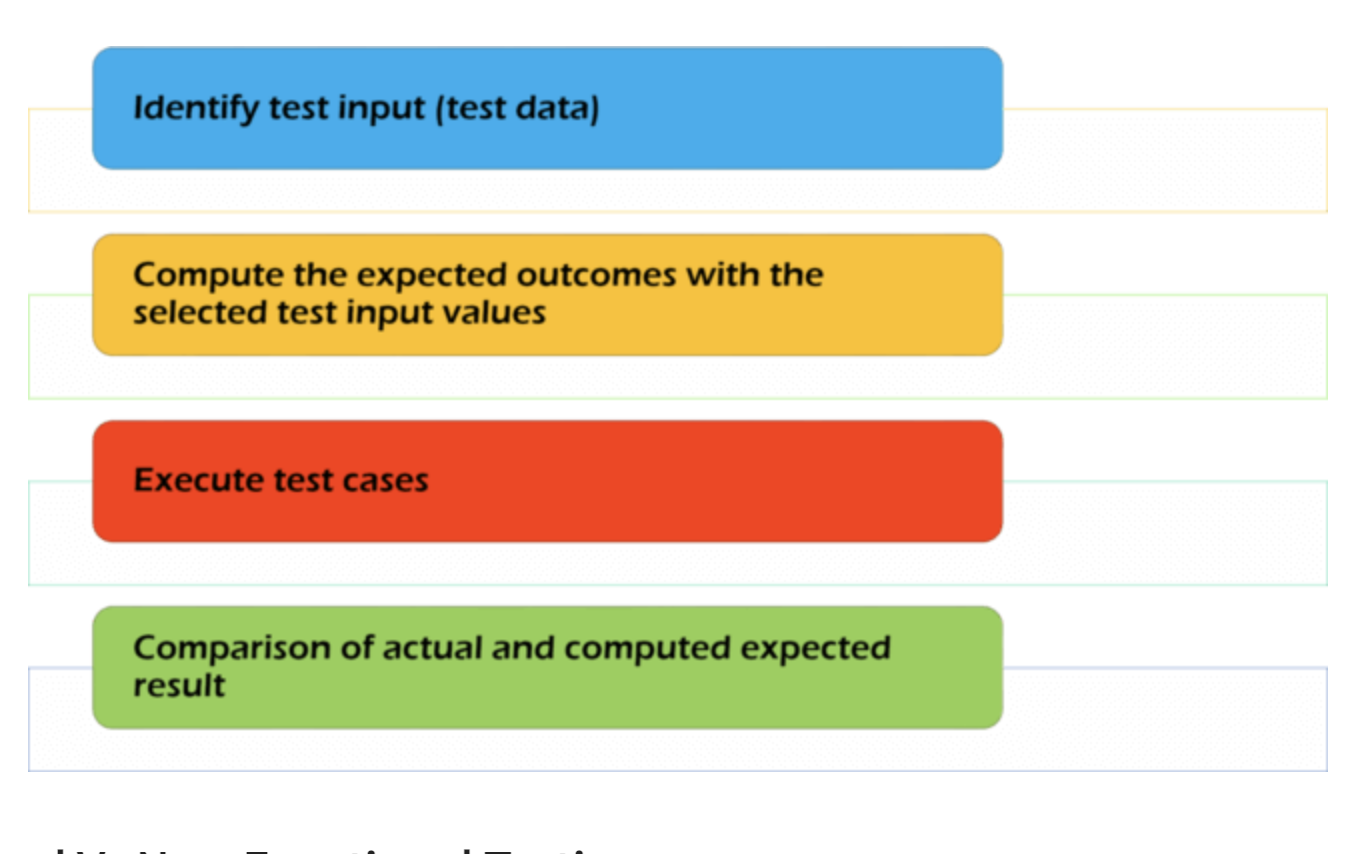
1. Kiểm tra HTML và CSS để đảm bảo rằng các công cụ tìm kiếm có thể thu thập dữ liệu trang web của bạn một cách dễ dàng. Điều này sẽ bao gồm:

* Kiểm tra lỗi cú pháp
* Các giản đồ màu có thể đọc được
* Tuân thủ tiêu chuẩn. Đảm bảo tuân thủ các tiêu chuẩn như W3C, OASIS, IETF, ISO, ECMA hoặc WS-I.

1. Kiểm tra quy trình kinh doanh– Điều này sẽ bao gồm:

* Kiểm tra các kịch bản quy trình làm việc / kinh doanh từ đầu đến cuối của bạn, điều này sẽ đưa người dùng đi qua một loạt các trang web để hoàn thành
* Kiểm tra cả các tình huống tiêu cực, chẳng hạn như khi người dùng thực hiện một bước không mong muốn, thông báo lỗi hoặc trợ giúp thích hợp sẽ hiển thị trong ứng dụng web của bạn

## Cách thực hiện kiểm thử chức năng



*Hình 1.3.2.1:* Các bước thức hiện kiểm thử chức năng

Thực hiện các bước sau :

1. Xác định chức năng nào của sản phẩm cần được kiểm tra. Điều này có thể thay đổi từ việc kiểm tra các chức năng chính, thông báo, điều kiện lỗi và / hoặc khả năng sử dụng của sản phẩm.

2. Tạo dữ liệu đầu vào cho các chức năng cần kiểm tra theo các yêu cầu cụ thể

3. Xác định các thông số đầu ra có thể chấp nhận được theo các yêu cầu quy định.

4. Thực thi các trường hợp kiểm thử.

5. So sánh sản lượng thực tế từ thử nghiệm với các giá trị đầu ra được xác định trước. Điều này cho biết hệ thống có hoạt động như mong đợi hay không.

# **Tìm hiểu về công cụ Selenium (Cương)**

## Giới thiệu về công cụ Selenium

- Selenium IDE là viết tắt của cụm từ Môi trường phát triển tích hợp Selenium, là một công cụ kiểm thử tự động. Đây là một trong những công cụ đơn giản và dễ dàng sử dụng nhất để cài đặt, tìm hiểu và tạo các kịch bản test. Công cụ này được đặt trên một bản ghi, có thể phát lại tập lệnh và cũng cho phép chỉnh sửa các tập lệnh được ghi (chỉnh sửa kịch bản kiểm thử).

- Điểm ấn tượng nhất của việc sử dụng selenium IDE là người dùng không bắt buộc phải có kiến thức lập trình nào trước đó. Tối thiểu mà người dùng cần là biết về HTML, DOM và JavaScript để tạo ra các tập lệnh thử nghiệm (kịch bản test) bằng công cụ này.

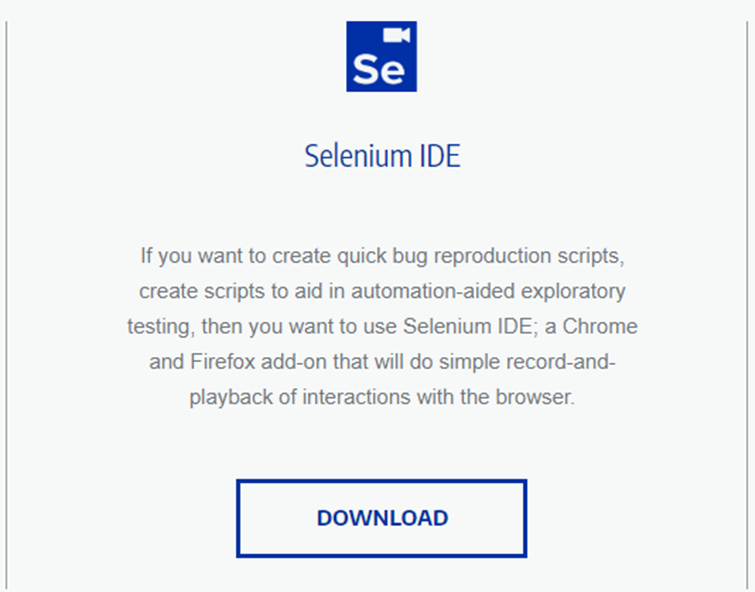
- Selenium IDE trước đây chỉ hỗ trợ trên Firefox nhưng giờ đã có trên trình duyệt Chrome, do đó các tập lệnh kiểm thử được tạo chỉ có thể được thực thi trên Firefox và Chrome. Một vài sơ hở nữa làm cho công cụ này không phù hợp để sử dụng cho các kịch bản thử nghiệm phức tạp. Khi đó sẽ cần đến các công cụ khác như Selenium RC, WebDriver.

## Cách cài đặt công cụ Selenium

· Cài đặt :

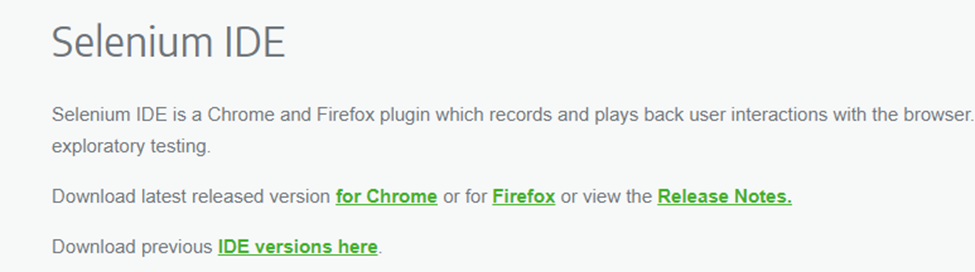
- **Bước 1: Tải xuống Selenium IDE:** Mở trình duyệt (Firefox hoặc Chorme) và nhập URL <http://seleniumhq.org/>. Nó sẽ mở ra trang chủ của Selenium.

- **Bước 2:** Click vào button "Download", nó sẽ điều hướng đến trang Tải xuống (như hình dưới).

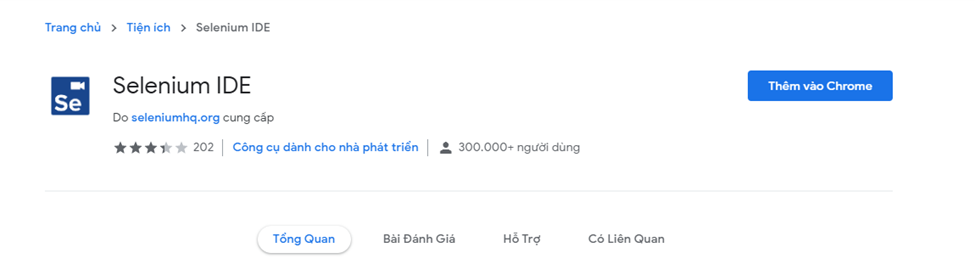


Trang này bao gồm tất cả các bản phát hành mới nhất của tất cả các bản Selenium.

- **Bước 3:** Kéo xuống mục Selenium IDE và chọn trình duyệt cần thêm



- **Bước 4:** Sau đó chúng ta sẽ Add công cụ này vào trình duyệt

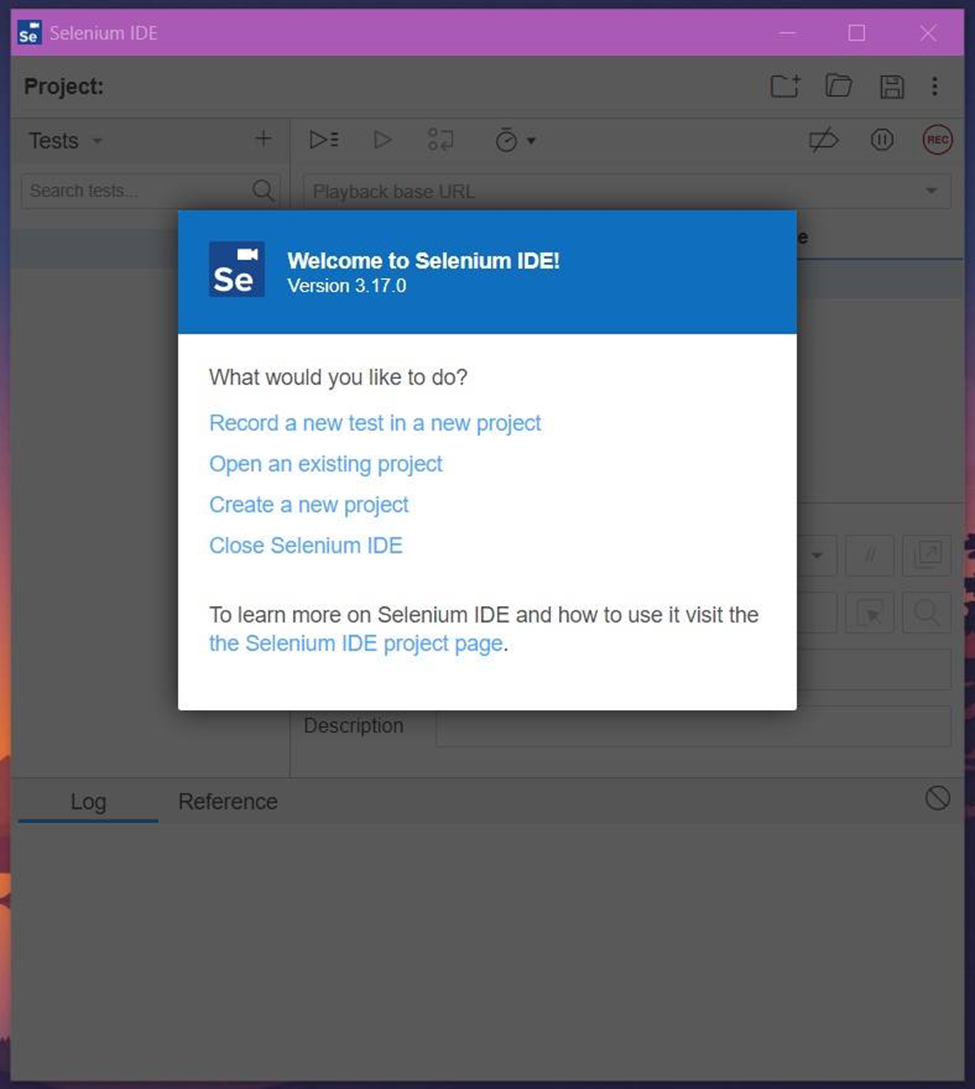


· Các chỉ dẫn cơ bản về giao diện Selenium IDE :

- Selenium IDE (Integrated Development Environment) được phát hành dưới dạng phần mềm bổ trợ (add-on) của Firefox,Google Chrome , cho phép test , edit và debug code . Selenium có thể sinh code tự động hoặc nạp các đoạn mã viết tay.

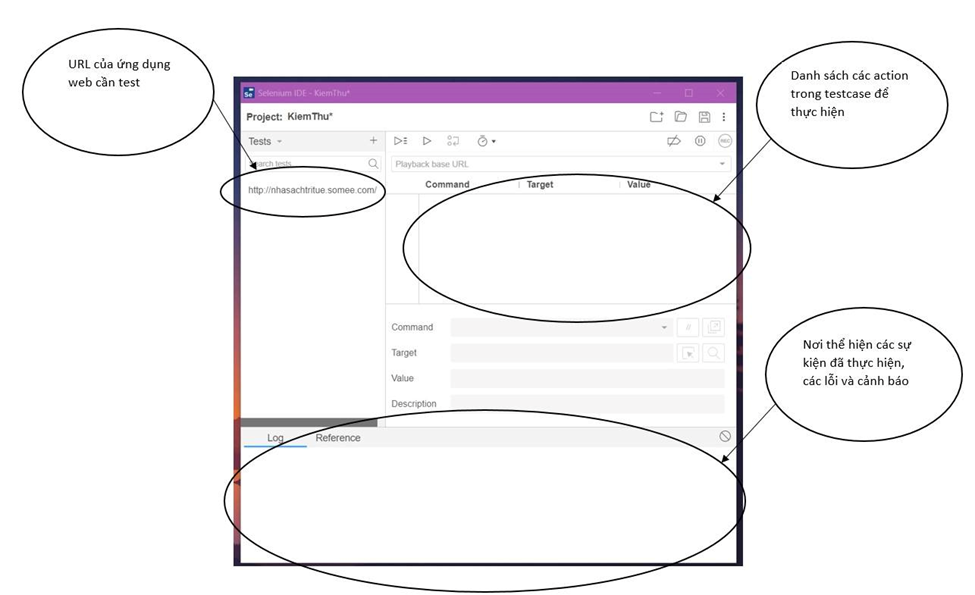
- Đề cài đặt Selenium IDE , vào tab Get Extension của Tool/Add-ons, trong phần tìm kiếm gõ từ khóa “Selenium IDE ” và sau đó tiến hành cài đặt . Khởi động lại trình duyệt , nếu cài đặt thành công thì trong mục Tool sẽ có thêm dòng Selenium IDE .

- Hình dưới là giao diện của Selenium IDE :

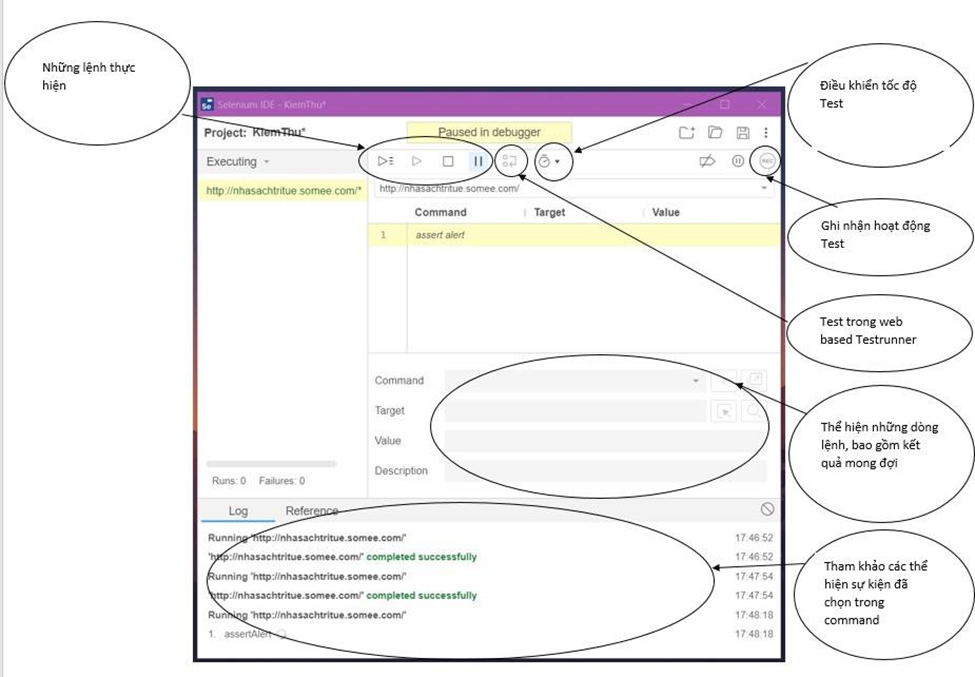


*Hình 1.3.2.1: Giao diện Selenium IDE*

Các chức năng trong File :

* Tạo test-case và test-suite mới.
* Mở test-case và test-suite đã lưu.
* Lưu test-case và test-suite theo định dạng html.
* Export test-case và test-suite theo các định dạng mà Selenium hỗ trợ.
* Thêm test-case.
* Thoát khỏi chương trình.
* Các chức năng trong Edit :
* Undo , Redo : Thực hiện lại , thực hiện tới các command.
* Cut , Copy , Paste , Delete : Cắt , copy , dán , xóa
* Select All : Chọn tất cả các command
* Các chức năng trong Options :
* Options : Lựa chọn một số tính năng : như encoding của file , chọn lựa phần mở rộng của Selenium IDE
* Format : Chọn dạng của nguồn test-script
* Clipboard Format : Chọn dạng của Clipboard
* Các chức năng trong Help :
* Các thông tin và tài liệu về Selenium IDE

*Hình 1.3.2.2: Giao diện vùng làm việc*



*Hình 1.3.2.3: Các thanh công cụ của Selenium IDE*

# **Ứng dụng công cụ kiểm thử Selenium IDE vào website và đánh giá**

## Lập kế hoạch(cả nhóm)

### Giới thiệu

Kế hoạch kiểm tra được thiết kế để quy định phạm vi, cách tiếp cận, tài nguyên và lịch trình của tất cả các hoạt động kiểm tra của dự án “MMQ”

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

#### Phạm vi

* Trong phạm vi

+ Tất cả các tính năng của trang “MMQ” đã được xác định trong thông số kỹ thuật yêu cầu phần mềm đều cần được kiểm tra

+ Kiểm tra giao diện

* Ngoài phạm vi

Các tính năng này không được kiểm tra:

+ Bảo mật và hiệu suất trang web

+ Kiểm tra tương thích

#### Mục tiêu chất lượng

Mục tiêu thử nghiệm là để xác minh Chức năng của trang web “MMQ”, dự án nên tập trung vào thử nghiệm hoạt động trang web như tạo project, thêm tài liệu, xem tài liệu, sửa tài liệu, xoá tài liệu…. để đảm bảo tất cả các chức năng này có hoạt động bình thường

#### Nhân sự thực hiện

| **Họ và Tên** | **MSSV** | **Trách nhiệm** |
| --- | --- | --- |
| Đoàn Thị Thuý | 2020603329 | Viết test, viết testcase cho tính năng Đăng nhập, thêm tài liệu |
| Nguyễn Thị Minh | 2020603396 | Viết test, viết testcase cho tính năng Đăng ký, tạo project |
| Phạm Văn Cương | 2020600079 | Viết test, viết testcase cho tính năng sửa tài liệu, xoá tài liệu |

### Phương pháp kiểm tra

#### Tổng quan

Quy trình kiểm thử website MMQ được dựa trên Mô hình thác nước(Waterfall) vì những lý do:

- Đây là mô hình đơn giản, dễ áp dụng, quy trình rõ ràng theo từng bước.

- Dễ quản lý và bảo trì bởi cách tiếp cận tuyến tính và cố định theo từng bước.

- Các tiêu chí đầu vào và đầu ra được xác định rõ ràng nên dễ dàng trong công tác kiểm tra chất lượng.

- Kiểm thử website MMQ là một dự án kiểm thử nhỏ. Mô hình thác nước hoạt động hiệu quả trong các dự án nhỏ, với các yêu cầu rõ ràng.

- Có nhiều tài liệu cung cấp cho khách hàng.

#### Các quy trình kiểm thử

Chạy thử các chức năng của sản phẩm

+ Đăng nhập

+ Đăng ký

+ Tạo project

+ Thêm tài liệu

+ Xem tài liệu

+ Sửa tài liệu

+ Kiểm thử giao diện của sản phẩm

#### Phân loại Lỗi

Mục tiêu của việc phân loại là để

+ Xác định loại giải pháp cho từng lỗi

+ Sắp xếp thứ tự ưu tiên các lỗi và xác định lịch trình cho tất cả dùng để “Sửa lỗi”.

#### Tiêu chí dừng lại và tiếp tục kiểm thử

Khi thực hiện kiểm thử chức năng, kiểm thử các chức năng chính của trang web thì dừng kiểm thử.

### Nhu cầu tài nguyên & môi trường

#### Công cụ kiểm thử

Bài tập kiểm thử trang web MMQ sử dụng công cụ Selenium IDE

#### Môi trường thực hiện test:

Môi trường thực hiện kiểm thử trang web MMQ là web server [https://mmq.hithaui.com](https://mmq.hithaui.com/register)

## Kịch bản kiểm thử và test case

### Kiểm thử đăng ký(Minh)

| **Project Name:** [**https://mmq.hithaui.com**](https://mmq.hithaui.com/) | |
| --- | --- |
| **Test Case** | |
| **Test Priority: High** | Test Designed by: Nguyễn Thị Minh |
| **Module Name: Test\_Register** | Test Designed date: 25/11/2022 |
| **Test Title: Đăng ký** | Test Executed by: Nguyễn Thị Minh |
| **Description: Test chức năng đăng ký** | Test Execution date: 20/05/2020 |

| **Test Case ID** | **Testcase** | **Step #** | **Step Details** | **Test data** | **Expected Results** | **Actual Results** | **Pass / Fail** |
| --- | --- | --- | --- | --- | --- | --- | --- |
|
| TR\_1 | Nhập đầy đủ tên user, email, password | 1 | Điều hướng đến https://mmq.hithaui.com/register | Tên user: minh1  Email: a1@gmail.com  Mật khẩu: 123 | Điều hướng đến trang chủ | Đúng như mong đợi | Pass |
| 2 | Nhập Username, Email & Password |
| 3 | Click vào box Agree the terms and policy |
| 4 | Click vào Button Register |
| TR\_2 | Nhập thiếu tên user | 1 | Điều hướng đến [https://mmq.hithaui.com/register](https://mmq.hithaui.com/login) | Email: a2@gmail.com  Mật khẩu: 123 | Thông báo lỗi thiếu tên user | Không thông báo thiếu thông tin | Fail |
| 2 | Nhập Email & Password |
| 3 | Click vào box Agree the terms and policy |
| 4 | Click vào Button Register |
| TR\_3 | Nhập thiếu email | 1 | Điều hướng đến [https://mmq.hithaui.com/register](https://mmq.hithaui.com/login) | Tên user: minh2  Mật khẩu: 123 | Thông báo lỗi thiếu email | Không thông báo thiếu thông tin | Fail |
| 2 | Nhập Username & Password |
| 3 | Click vào box Agree the terms and policy |
| 4 | Click vào Button Register |
| TR\_4 | Nhập thiếu mật khẩu | 1 | Điều hướng đến [https://mmq.hithaui.com/register](https://mmq.hithaui.com/login) | Tên user: minh3  Email: a3@gmail.com | Thông báo lỗi thiếu mật khẩu | Không thông báo thiếu thông tin | Fail |
| 2 | Nhập Username & Email |
| 3 | Click vào box Agree the terms and policy |
| 4 | Click vào Button Register |
| TR\_5 | Không click vào box Agree the terms and policy | 1 | Điều hướng đến [https://mmq.hithaui.com/register](https://mmq.hithaui.com/login) | Tên user: minh4  Email: a5@gmail.com  Mật khẩu: 123 | Thông báo lỗi chưa đồng ý điều khoản sử dụng | Không thông báo thiếu thông tin | Fail |
| 2 | Nhập Username, Email & Password |
| 3 | Không click vào box Agree the terms and policy |
| 4 | Click vào Button Register |
| TR\_6 | Nhập trùng Username đã đăng ký | 1 | Điều hướng đến [https://mmq.hithaui.com/register](https://mmq.hithaui.com/login) | Tên user: minh1  Email: a6@gmail.com  Mật khẩu: 123 | Thông báo lỗi trùng Username | Không thông báo trùng tên user | Fail |
| 2 | Nhập Username, Email & Password |
| 3 | Click vào box Agree the terms and policy |
| 4 | Click vào Button Register |

### Kiểm thử đăng nhập(Thuý)

| **Project Name:** [**https://mmq.hithaui.com**](https://mmq.hithaui.com/) | |
| --- | --- |
| **Test Case** | |
| **Test Priority: High** | Test Designed by: Đoàn Thị Thuý |
| **Module Name: Test\_Login** | Test Designed date: 25/11/2022 |
| **Test Title: Đăng nhập** | Test Executed by: Đoàn Thị Thuý |
| **Description: Test chức năng đăng nhập** | Test Execution date: 20/05/2020 |
| **Prerequisites** | Tài khoản đã được đăng ký |

| **Test Case ID** | **Testcase** | **Step #** | **Step Details** | **Test data** | **Expected Results** | **Actual Results** | **Pass / Fail** |
| --- | --- | --- | --- | --- | --- | --- | --- |
|
| LG\_1 | Nhập đầy đủ username, password | 1 | Điều hướng đến<https://mmq.hithaui.com/login> | username=test  password=123 | Điều hướng đến trang chủ | Như mong đợi | Pass |
| 2 | Nhập Username & Password |
| 3 | Click Login |
| LG\_2 | Nhập thiếu tên | 1 | Điều hướng đến<https://mmq.hithaui.com/login> | password=12 | Thông báo “Thiếu tên hoặc mật khẩu. Vui lòng nhập lại" | Không hiện thông báo | fail |
| 2 | Nhập Username & Password |
| 3 | Click Login |
| LG\_3 | Nhập thiếu password | 1 | Điều hướng đến<https://mmq.hithaui.com/login> | username=test | Thông báo “Thiếu tên hoặc mật khẩu. Vui lòng nhập lại" | Không hiện thông báo | fail |
| 2 | Nhập Username & Password |
| 3 | Click Login |
| LG\_4 | Nhập sai password | 1 | Điều hướng đến<https://mmq.hithaui.com/login> | username=test  password=12456789 | Thông báo “Nhập sai mật khẩu hoặc tên người dùng” | Không hiện thông báo | fail |
| 2 | Nhập Username & Password |
| 3 | Click Login |

### Kiểm thử thêm tài liệu(Thuý)

| **Project Name:** [**https://mmq.hithaui.com**](https://mmq.hithaui.com/) | |
| --- | --- |
| **Test Case** | |
| **Test Case ID: TC\_2** | Test Designed by: Đoàn Thị Thuý |
| **Test Priority: High** | Test Designed date: 25/11/2022 |
| **Module Name: Test\_Add\_Project** | Test Executed by: Đoàn Thị Thuý |
| **Test Title: Thêm tài liệu** | Test Execution date: 20/05/2020 |
| **Description: Test chức năng thêm tài liệu** |  |
| **Prerequisites** | Đã đăng nhập và tạo project |

| **Test Case ID** | **Test**  **case** | **Step #** | **Step Details** | **Test data** | **Expected Results** | **Actual Results** | **Pass / Fail** |
| --- | --- | --- | --- | --- | --- | --- | --- |
|
| AD\_1 | File hợp lệ | 1 | Click vào icon xem tài liệu ở project muốn xem |  | Thêm thành công | As Expected | Pass |
| 2 | Click vào button "Add document" |  |
| 3 | Click vào "Upload file" | unit4-2.docx |
| 4 | Click button "Add" |  |
| AD\_2 | File không hợp lệ | 1 | Click vào icon xem tài liệu ở project muốn xem |  | Không add được file | As Expected | pass |
| 2 | Click vào button "Add document" |  |
| 3 | Click vào "Upload file" | unit4-2.rar |
| 4 | Click button "Add" |  |
| AD\_3 | Huỷ không thêm file | 1 | Click vào icon xem tài liệu ở project muốn xem |  | Không thêm file nào ở bảng | As Expected | pass |
| 2 | Click vào button "Add document" |  |
| 3 | Click button "Cancel" |  |

### Kiểm thử tạo project(Minh)

| **Project Name:** [**https://mmq.hithaui.com**](https://mmq.hithaui.com/) | |
| --- | --- |
| **Test Case** | |
| **Test Case ID: TC\_4** | Test Designed by: Nguyễn Thị Minh |
| **Test Priority: High** | Test Designed date: 25/11/2022 |
| **Module Name: Test\_AddProject** | Test Executed by: Nguyễn Thị Minh |
| **Test Title:Thêm tài liệu** | Test Execution date: 20/05/2020 |
| **Description: Test chức năng tạo Project** |  |
| **Prerequisites** | Đã đăng nhập tài khoản |

| **Test Case ID** | **Test**  **case** | **Step #** | **Step Details** | **Test data** | **Expected Results** | **Actual Results** | **Pass / Fail** |
| --- | --- | --- | --- | --- | --- | --- | --- |
|
| AP\_1 | Thêm thành công project với đầy đủ tên, mô tả | 1 | Điều hướng đến [https://mmq.hithaui.com](https://mmq.hithaui.com/login) | Tên tài liệu: Test 1  Mô tả: Mô tả của test1  Ngôn ngữ nguồn: Vietnamese  Ngôn ngữ mục tiêu: Vietnamese | Tài liệu được thêm vào CSDL và hiển thị lên màn hình | Như mong đợi | Pass |
| 2 | Kích vào Project trên thanh Menu |
| 3 | Click vào button Add |
| 4 | Nhập tên tài liệu, mô tả |
| 5 | Chọn Ngôn ngữ nguồn và ngôn ngữ mục tiêu |
| 6 | Click vào Save changes |
| AP\_2 | Thêm tài liệu nhưng thiếu tên project | 1 | Điều hướng đến [https://mmq.hithaui.com](https://mmq.hithaui.com/login) | Mô tả: Mô tả của test2  Ngôn ngữ nguồn: Vietnamese  Ngôn ngữ mục tiêu: Vietnamese | Thông báo lỗi thiếu tên tài liệu | Không thông báo lỗi thiếu tên tài liệu | Fail |
| 2 | Kích vào Project trên thanh Menu |
| 3 | Click vào button Add |
| 4 | Nhập tên tài mô tả |
| 5 | Chọn Ngôn ngữ nguồn và ngôn ngữ mục tiêu |
| 6 | Click vào Save changes |
| AP\_3 | Tên project bị trùng | 1 | Điều hướng đến [https://mmq.hithaui.com](https://mmq.hithaui.com/login) | Tên tài liệu: Test1  Mô tả: Mô tả của test3  Ngôn ngữ nguồn: Vietnamese  Ngôn ngữ mục tiêu: Vietnamese | Thông báo lỗi tên tài liệu bị trùng | Không thông báo tên tài liệu bị trùng | Fail |
| 2 | Kích vào Project trên thanh Menu |
| 3 | Click vào button Add |
| 4 | Nhập tên tài mô tả |
| 5 | Chọn Ngôn ngữ nguồn và ngôn ngữ mục tiêu |
| 6 | Click vào Save changes |

### Kiểm thử sửa tài liệu(Cương)

| **Project Name:** [**https://mmq.hithaui.com**](https://mmq.hithaui.com/) | |
| --- | --- |
| **Test Case** | |
| **Test Priority: High** | Test Designed by: Phạm Văn Cương |
| **Module Name: Test\_EditDocument** | Test Designed date: 25/11/2022 |
| **Test Title: Đăng nhập** | Test Executed by: Phạm Văn Cương |
| **Description: Test chức năng sửa tài liệu** | Test Execution date: 20/05/2020 |
| **Prerequisites** | Đã đăng nhập tài khoản, tài liệu đã được thêm |

| **Test Case ID** | **Test**  **case** | **Step #** | **Step Details** | **Test data** | **Expected Results** | **Actual Results** | **Pass / Fail** |
| --- | --- | --- | --- | --- | --- | --- | --- |
|
| ED\_1 | Sửa tài liệu | 1 | Điều hướng đến<https://mmq.hithaui.com/login> | Ô target: "Nội dung đã sửa đổi" | Sửa nội dung của tài liệu trong Project |  |  |
| 2 | Kích vào Project trong thanh Menu |
| 3 | Kích vào Action xem Tài liệu |
| 4 | Kích vào Action Edit |
| 5 | Sửa nội dung Target |
| 6 | Kích button Save changes |
| 7 | Kích Export Document để lấy tài liệu được sửa đổi |

### Kiểm thử xoá tài liệu(Cương)

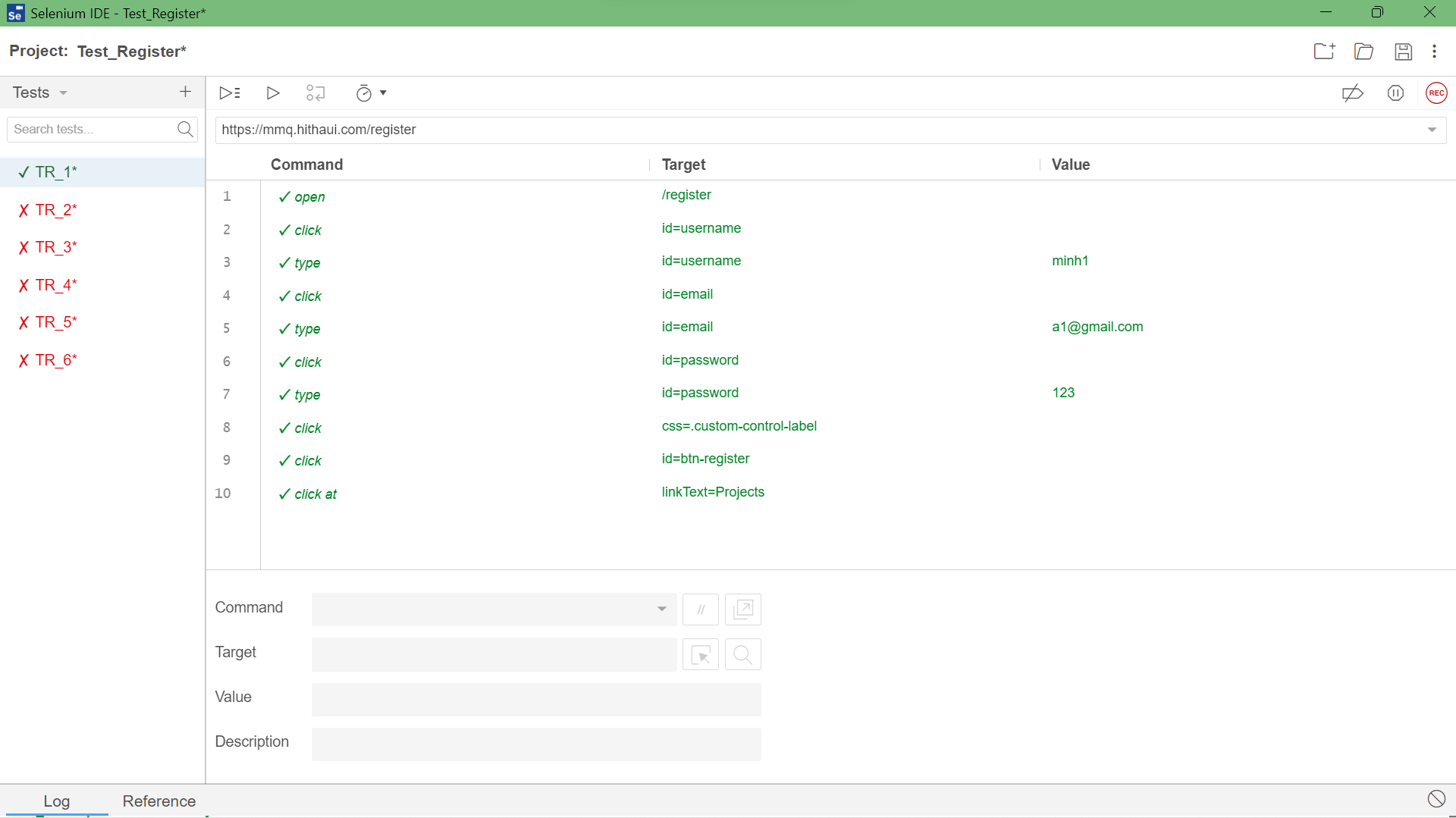
| **Project Name:** [**https://mmq.hithaui.com**](https://mmq.hithaui.com/) | |
| --- | --- |
| **Test Case** | |
| **Test Priority: High** | Test Designed by: Phạm Văn Cương |
| **Module Name: Test\_DeleteProfile** | Test Designed date: 25/11/2022 |
| **Test Title:Xóa tài liệu** | Test Executed by: Phạm Văn Cương |
| **Description: Test chức năng xóa Tài liệu** | Test Execution date: 20/05/2020 |
| **Prerequisites** | Đã đăng nhập tài khoản, tài liệu đã được thêm |

| **Test Case ID** | **Test**  **case** | **Step #** | **Step Details** | **Test data** | **Expected Results** | **Actual Results** | **Pass / Fail** |
| --- | --- | --- | --- | --- | --- | --- | --- |
|
| DD\_1 | Xóa 1 tài liệu | 1 | Điều hướng đến [https://mmq.hithaui.com/projects](https://mmq.hithaui.com/login) |  | Tài liệu xóa khỏi CSDL và hiển thị lên màn hình danh sách project |  |  |
| 2 | Kích vào nút Project trong thanh Menu |  |
| 3 | Kích vào Action Delete |  |
| 4 | Kích vào button Delete |  |
| DD\_2 | Bấm nút huỷ không xoá tài liệu | 1 | Điều hướng đến [https://mmq.hithaui.com/projects](https://mmq.hithaui.com/login) |  | Tài liệu chưa được xoá khỏi CSDL và hiển thị lên màn hình danh sách project |  |  |
| 2 | Kích vào nút Project trong thanh Menu |  |
| 3 | Kích vào Action Delete |  |
| 4 | Kích vào button Close |  |

## Tiến hành kiểm thử

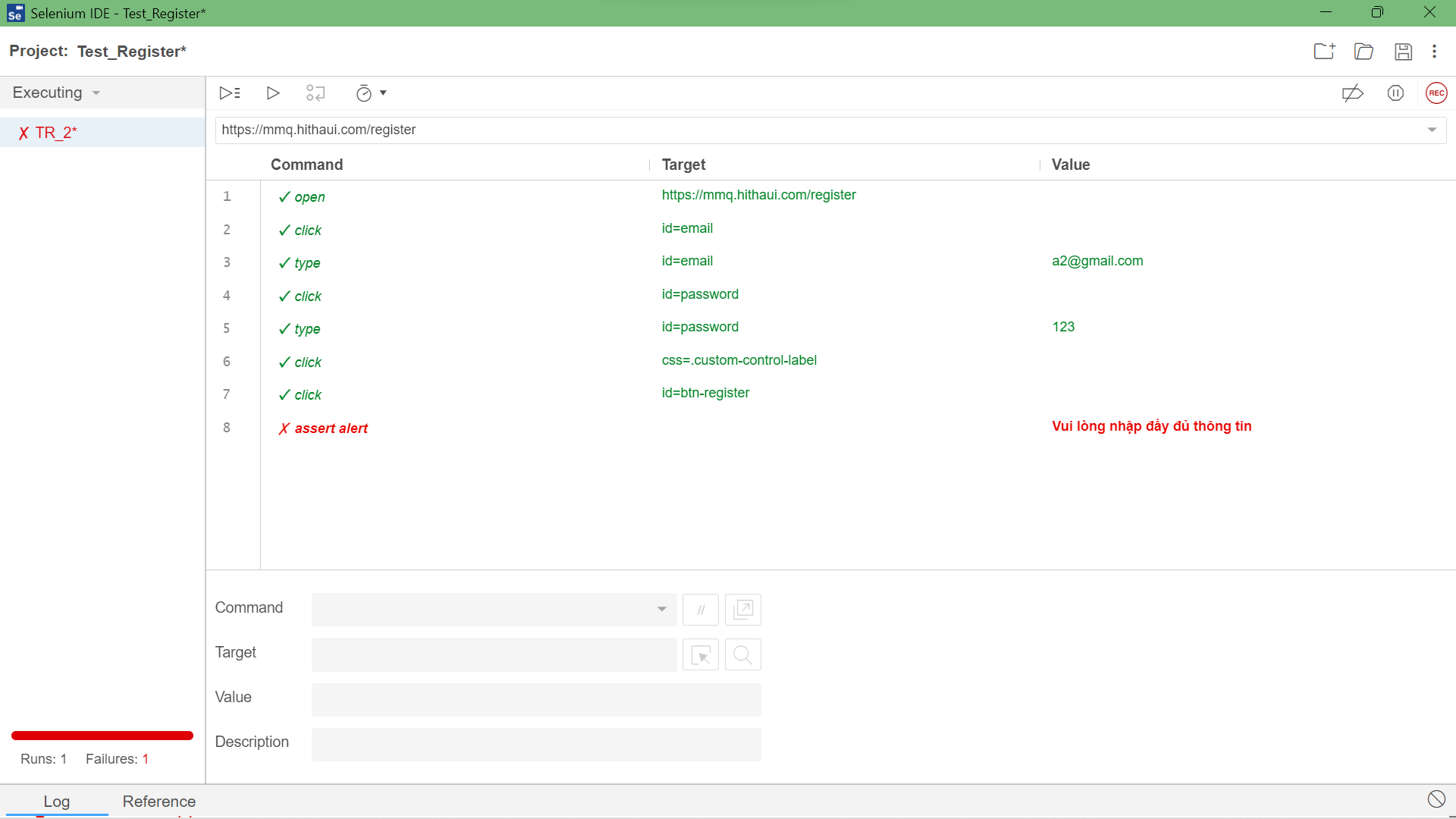
### Kiểm thử đăng ký(Minh)

- TH1: Nhập đầy đủ và đúng username, email, password



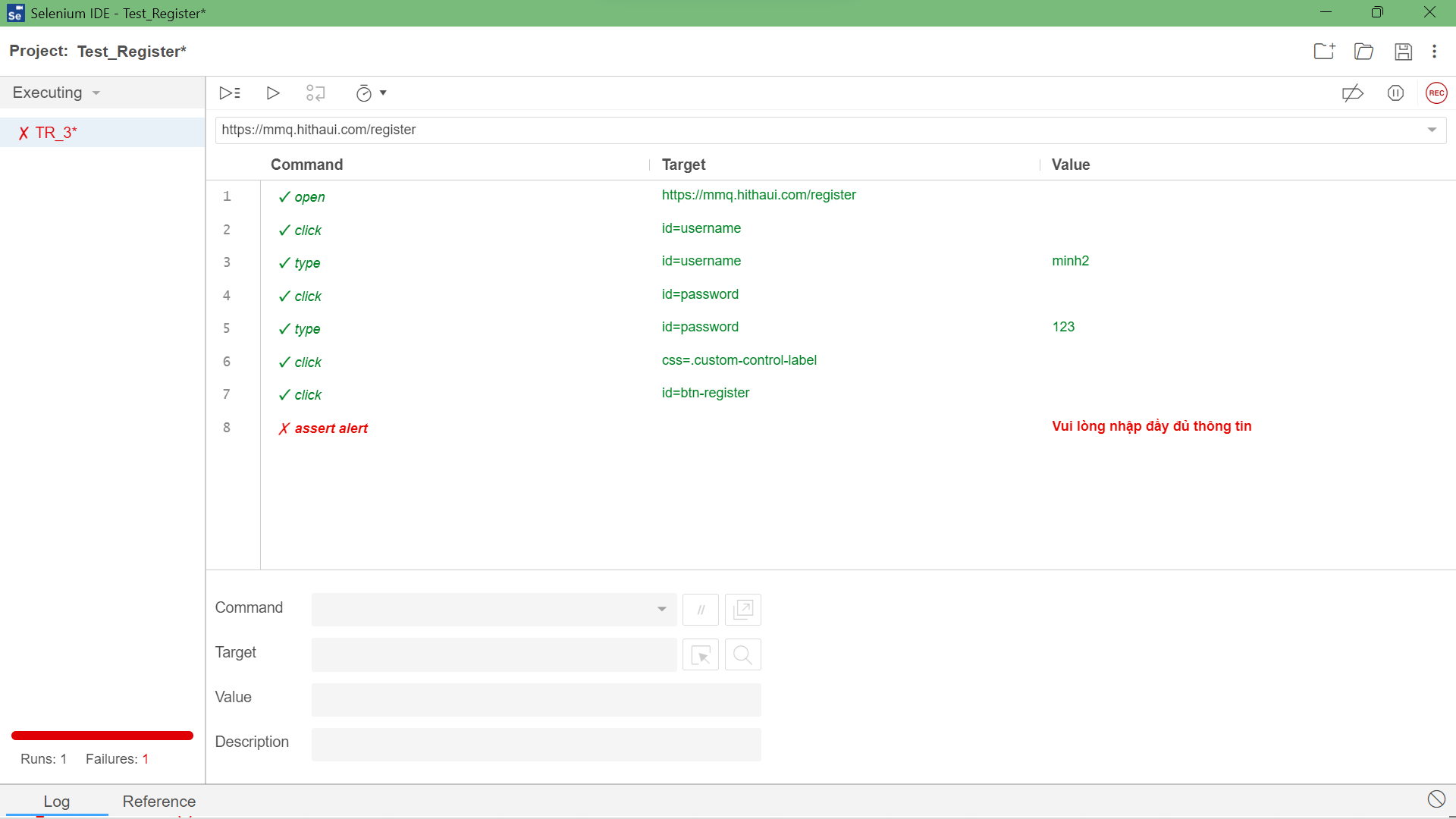
*Hình 4.3.1.1:* Kết quả test trường hợp dữ liệu cung cấp khi đăng ký là đủ và đúng

- TH2: Nhập thiếu tên user



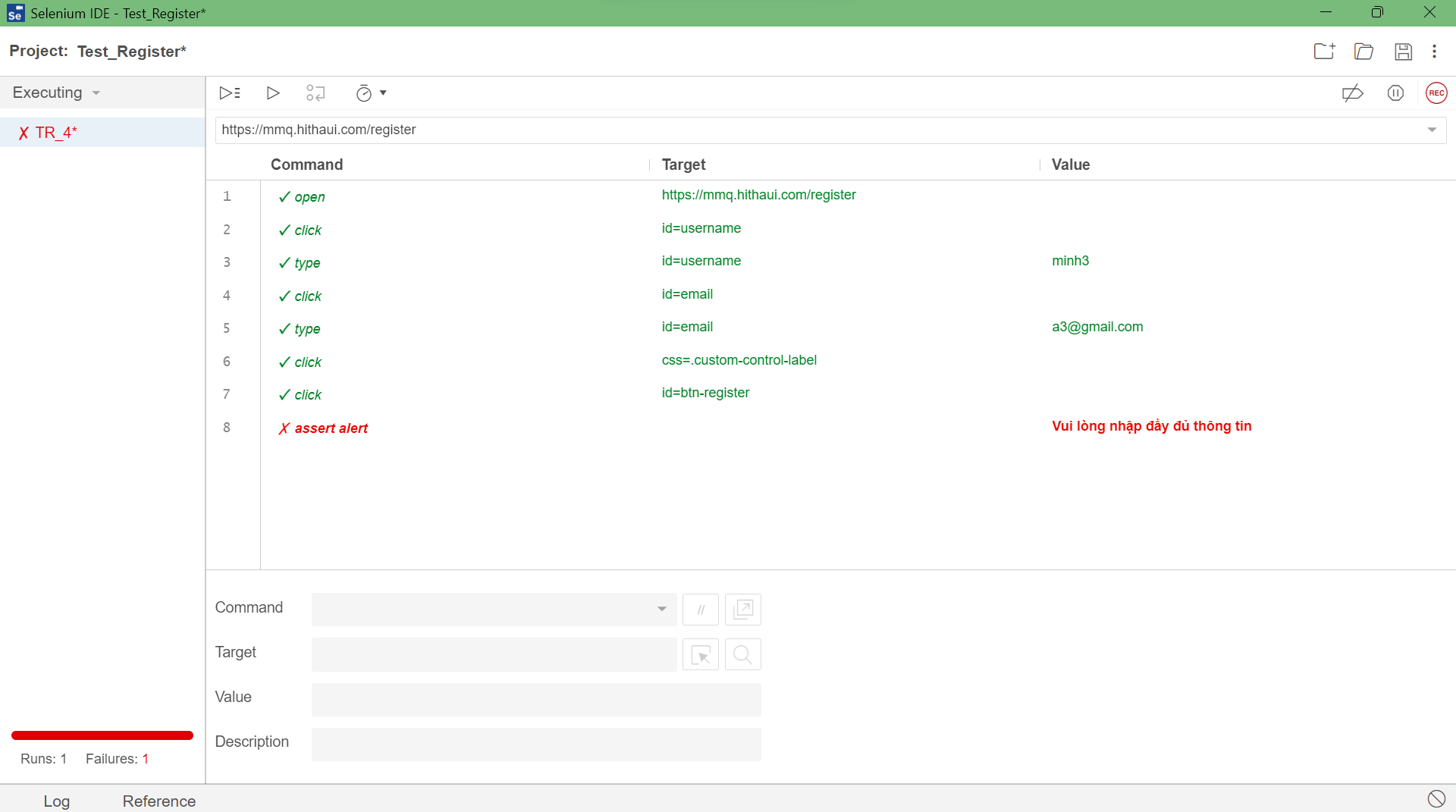
*Hình 4.3.1.2:* Kết quả test trường hợp dữ liệu cung cấp để đăng ký thiếu tên user

- TH3: Nhập thiếu email



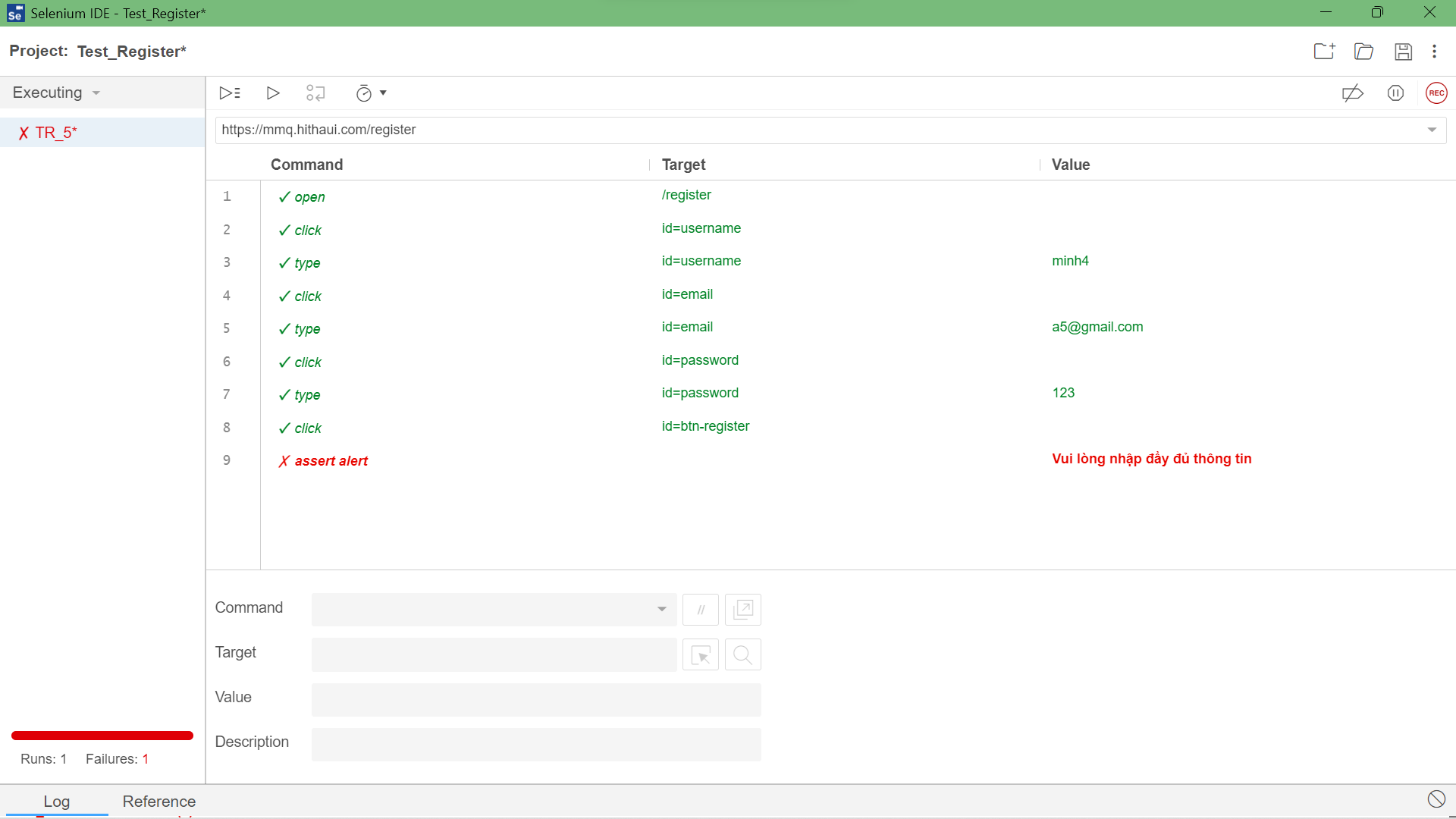
*Hình 4.3.1.3:* Kết quả test trường hợp dữ liệu cung cấp khi đăng ký thiếu email

*-* TH4: Nhập thiếu mật khẩu



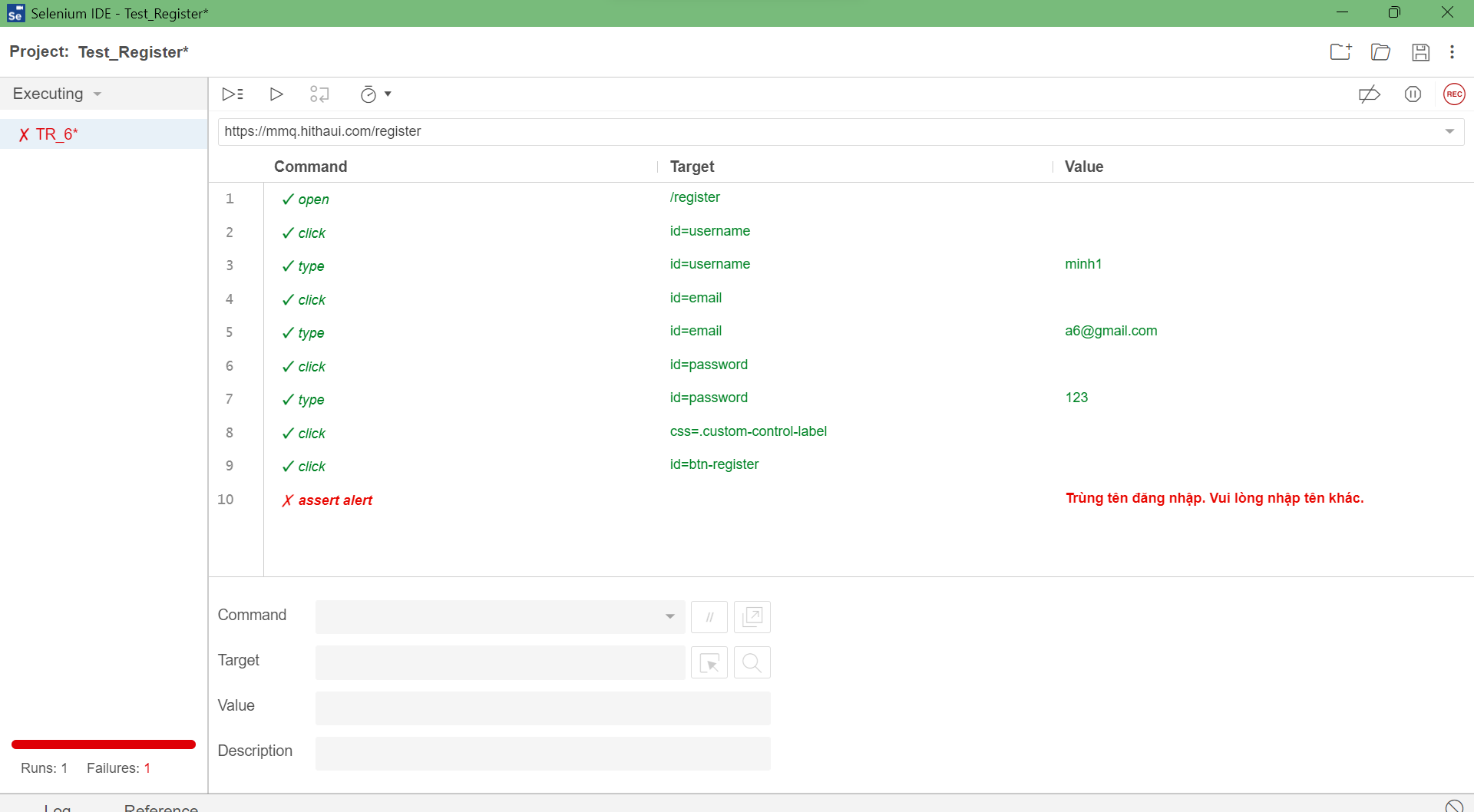
*Hình 4.3.1.4:* Kết quả test trường hợp dữ liệu cung cấp khi đăng ký thiếu mật khẩu

- TH5: Không click vào box Agree the terms and policy



*Hình 4.3.1.5:* Kết quả test trường hợp dữ liệu cung cấp khi đăng ký không click vào box Agree the terms and policy

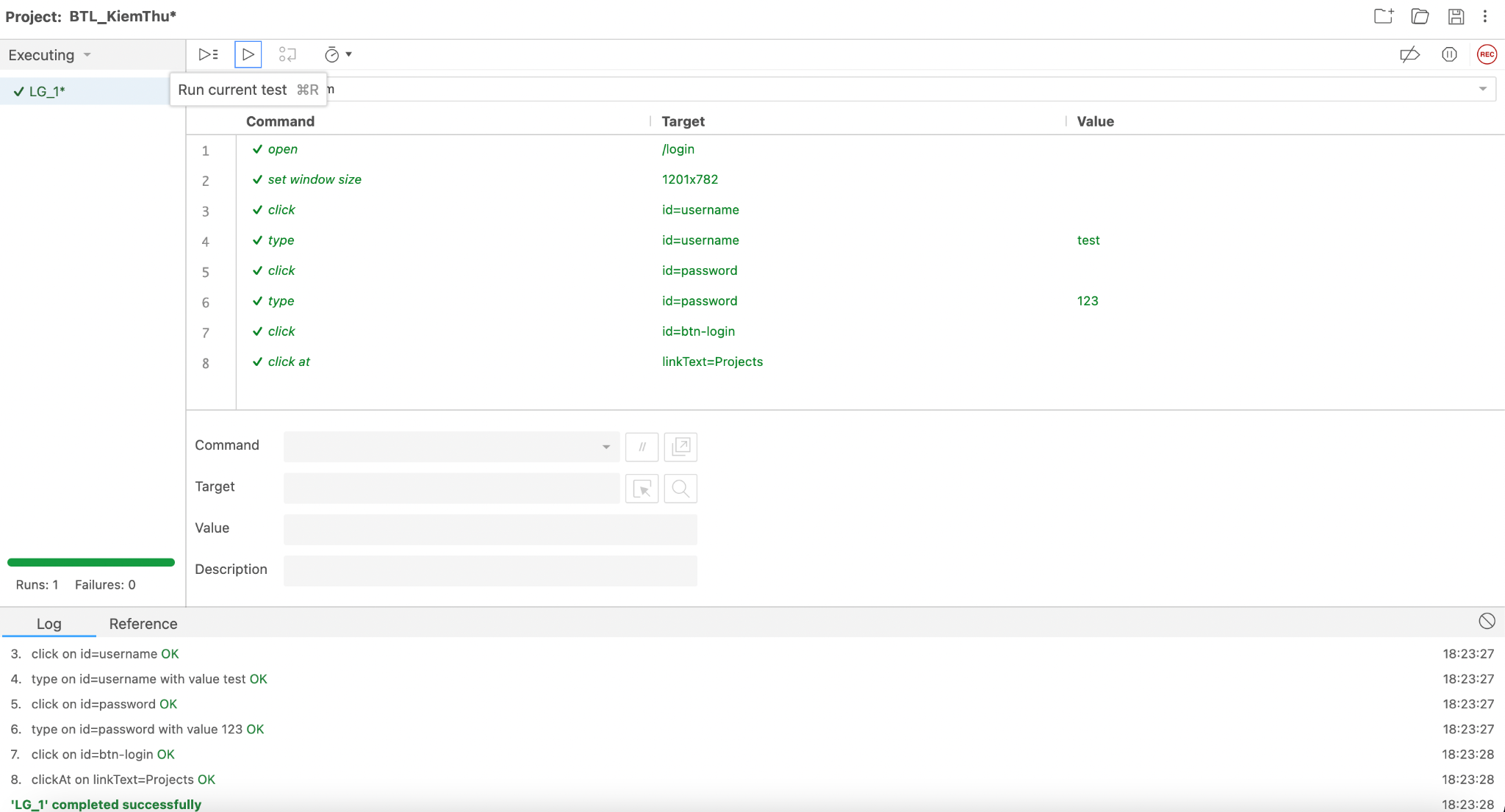
- TH6: Nhập trùng Username đã đăng ký



*Hình 4.3.1.6:* Kết quả test trường hợp dữ liệu cung cấp khi đăng ký nhập trùng Username đã đăng ký

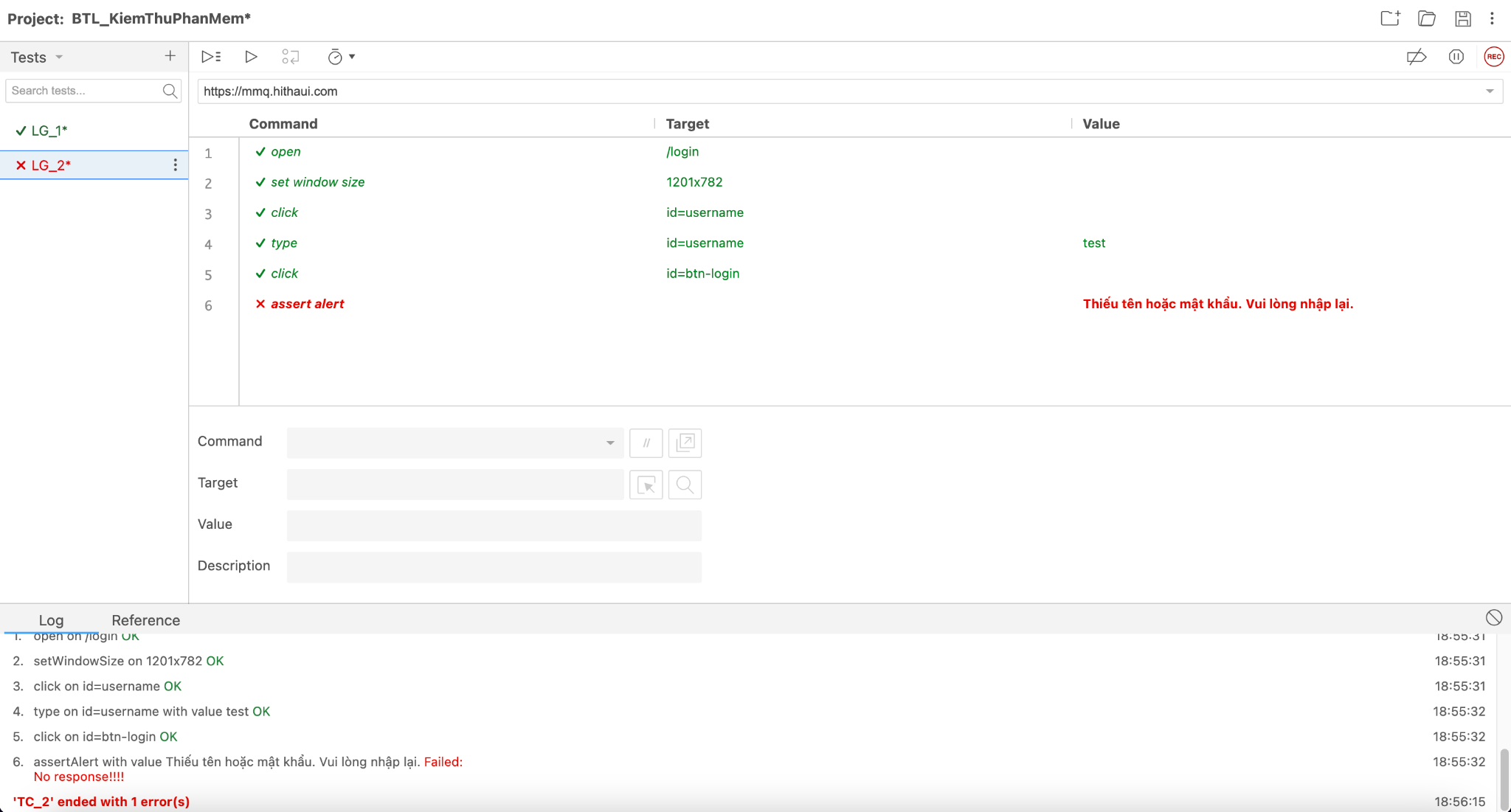
### Kiểm thử đăng nhập(Thuý)

- TH1: Nhập đầy đủ và đúng username, password

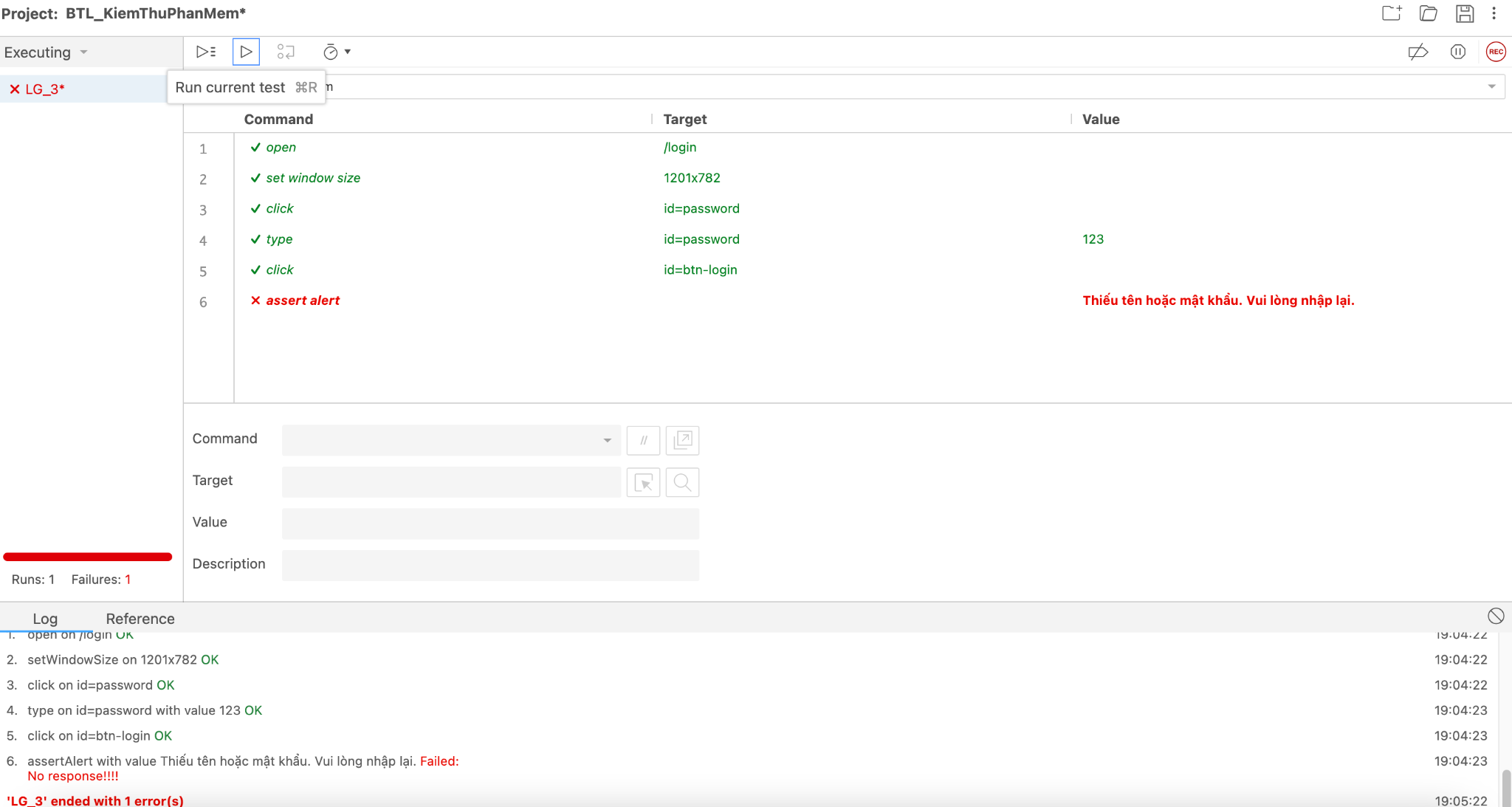


*Hình 4.3.2.1:* Kết quả test trường hợp dữ liệu cung cấp để đăng nhập là đủ và đúng

- TH2: Nhập thiếu password



*Hình 4.3.2.2:* 2 Kết quả test trường hợp dữ liệu cung cấp để đăng nhập thiếu password

- TH3: Nhập thiếu username

*Hình 4.3.2.3:* Kết quả test trường hợp dữ liệu cung cấp để đăng nhập thiếu username

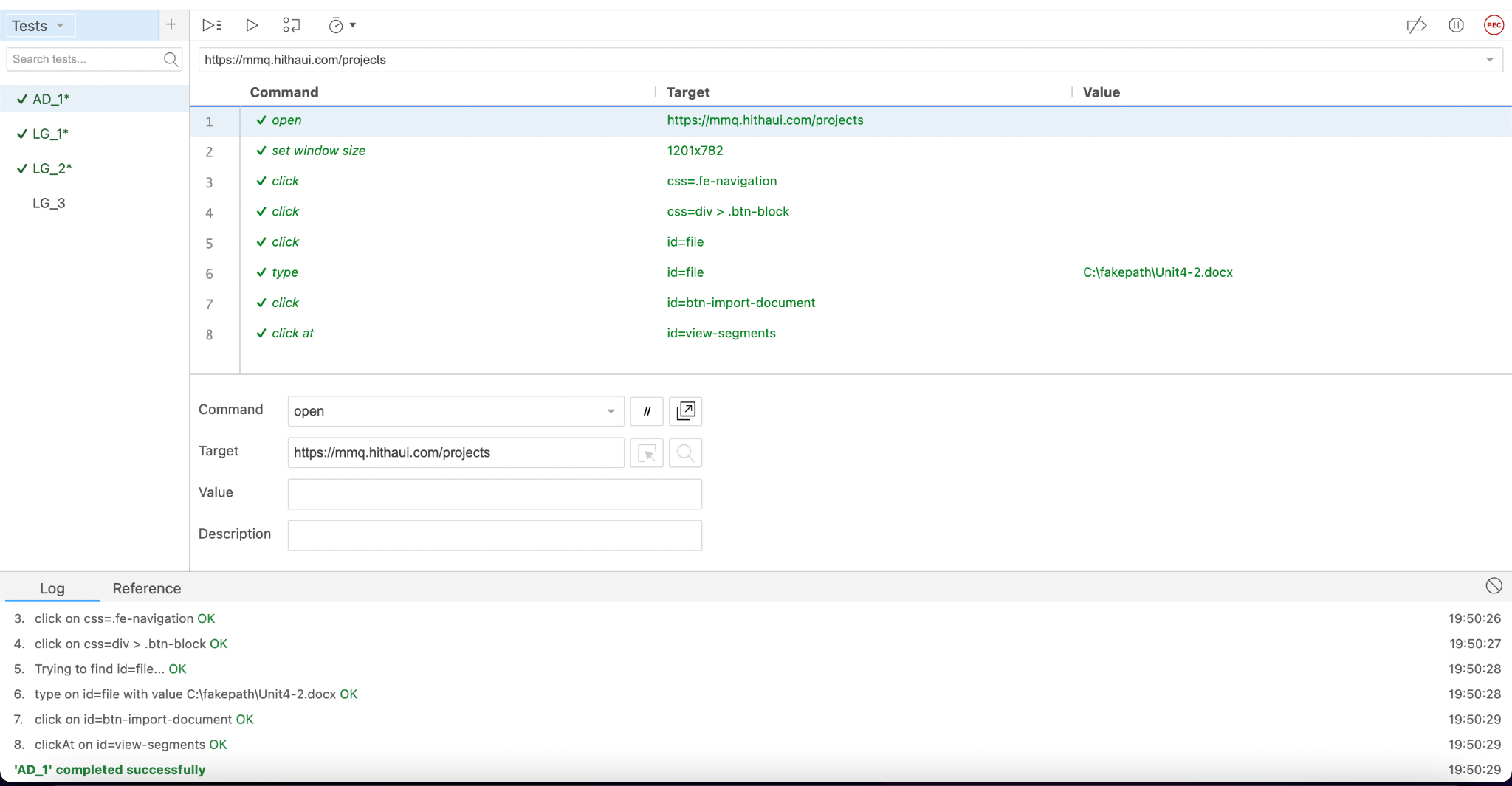
- TH4: Nhập sai password



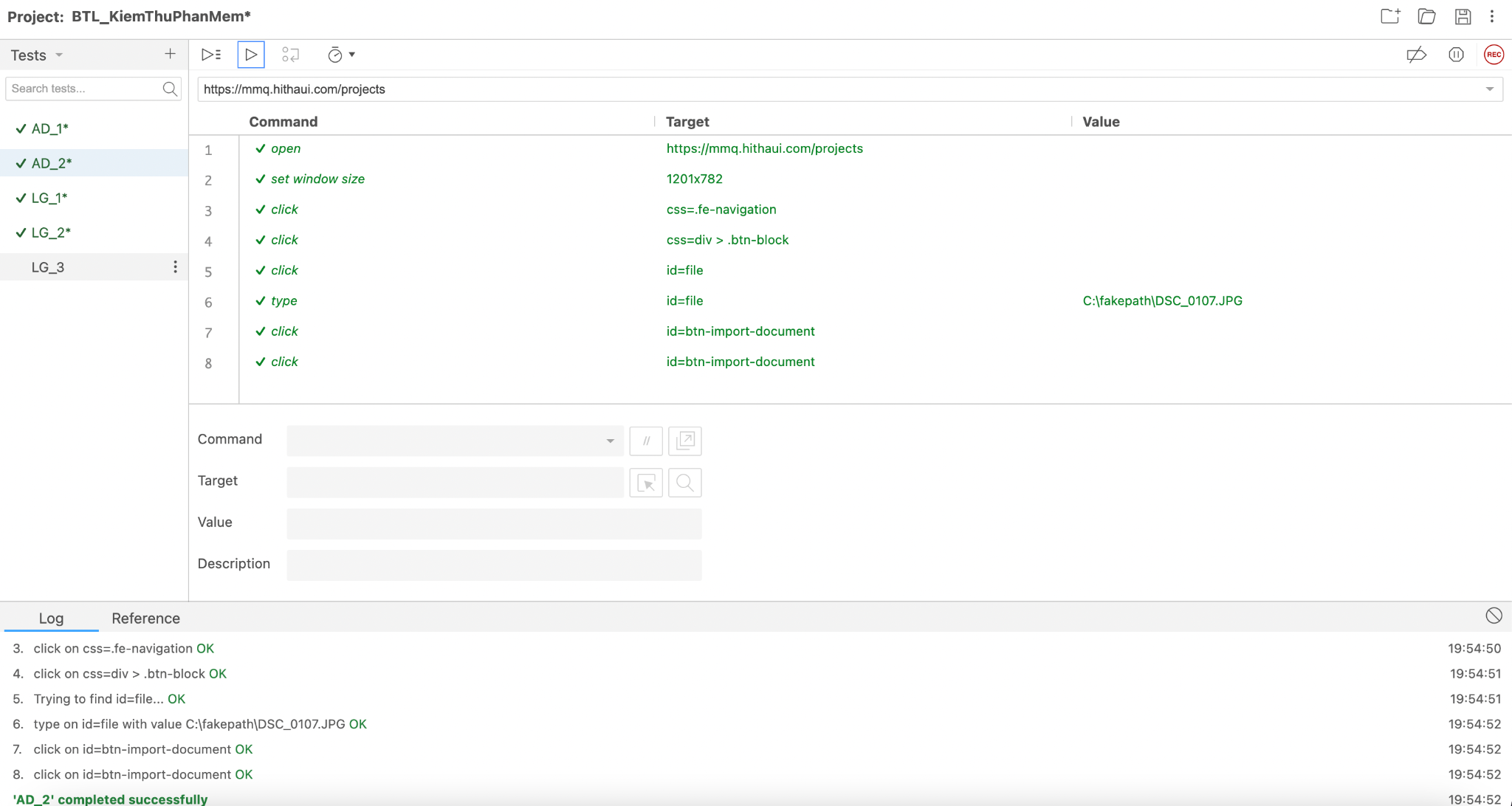
Hình 4.3.2.4: Kết quả test trường hợp dữ liệu cung cấp để đăng nhập sai password

### Kiểm thử thêm tài liệu (Thuý)

- TH1: Thêm file hợp lệ

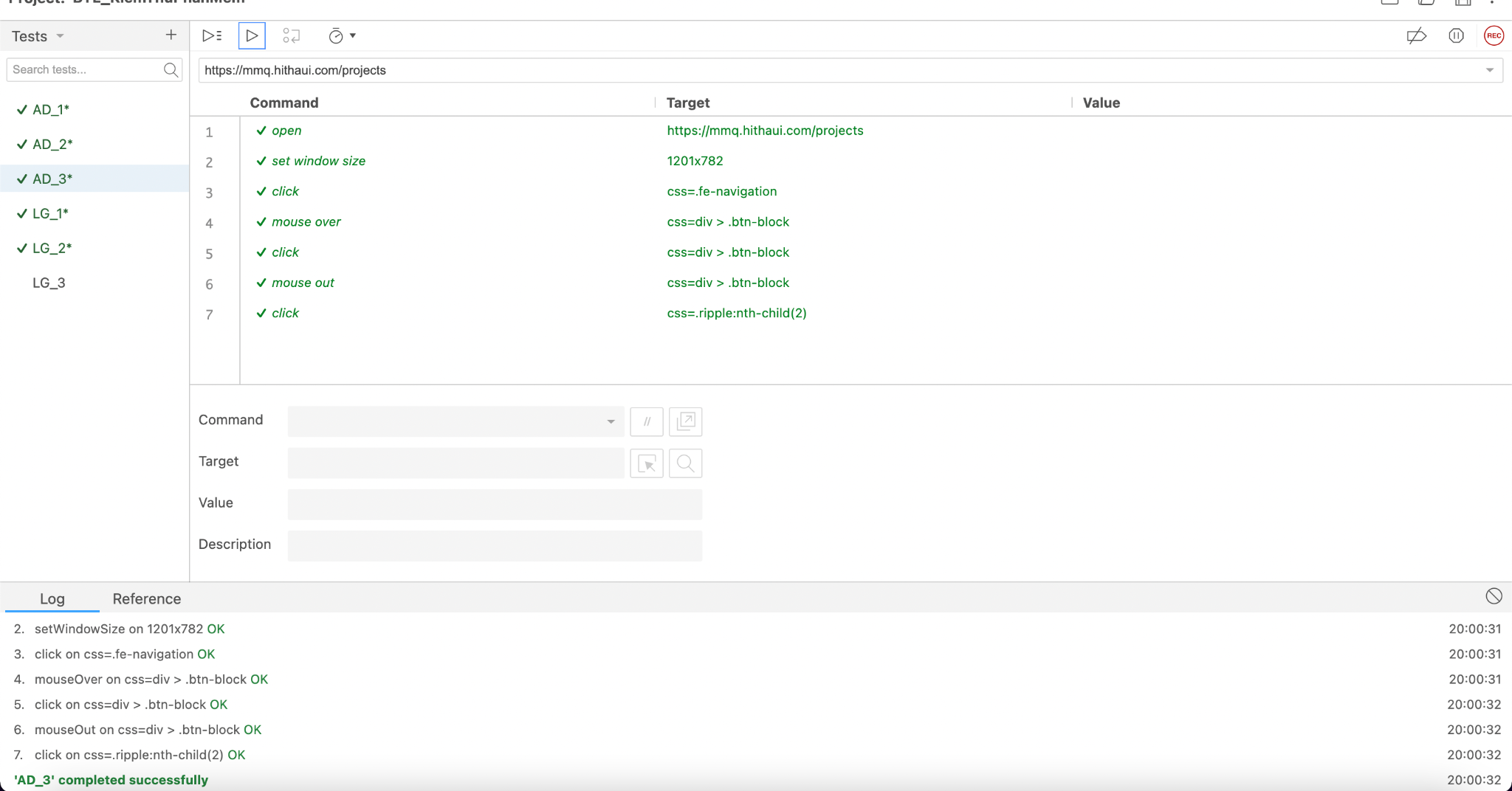


*Hình 4.3.3.1:* Kết quả test trường hợp dữ liệu cung cấp để thêm tài liệu file hợp lệ

- TH2: Thêm file không hợp lệ

*Hình 4.3.3.2:* Kết quả test trường hợp dữ liệu cung cấp để thêm tài liệu file không hợp lệ

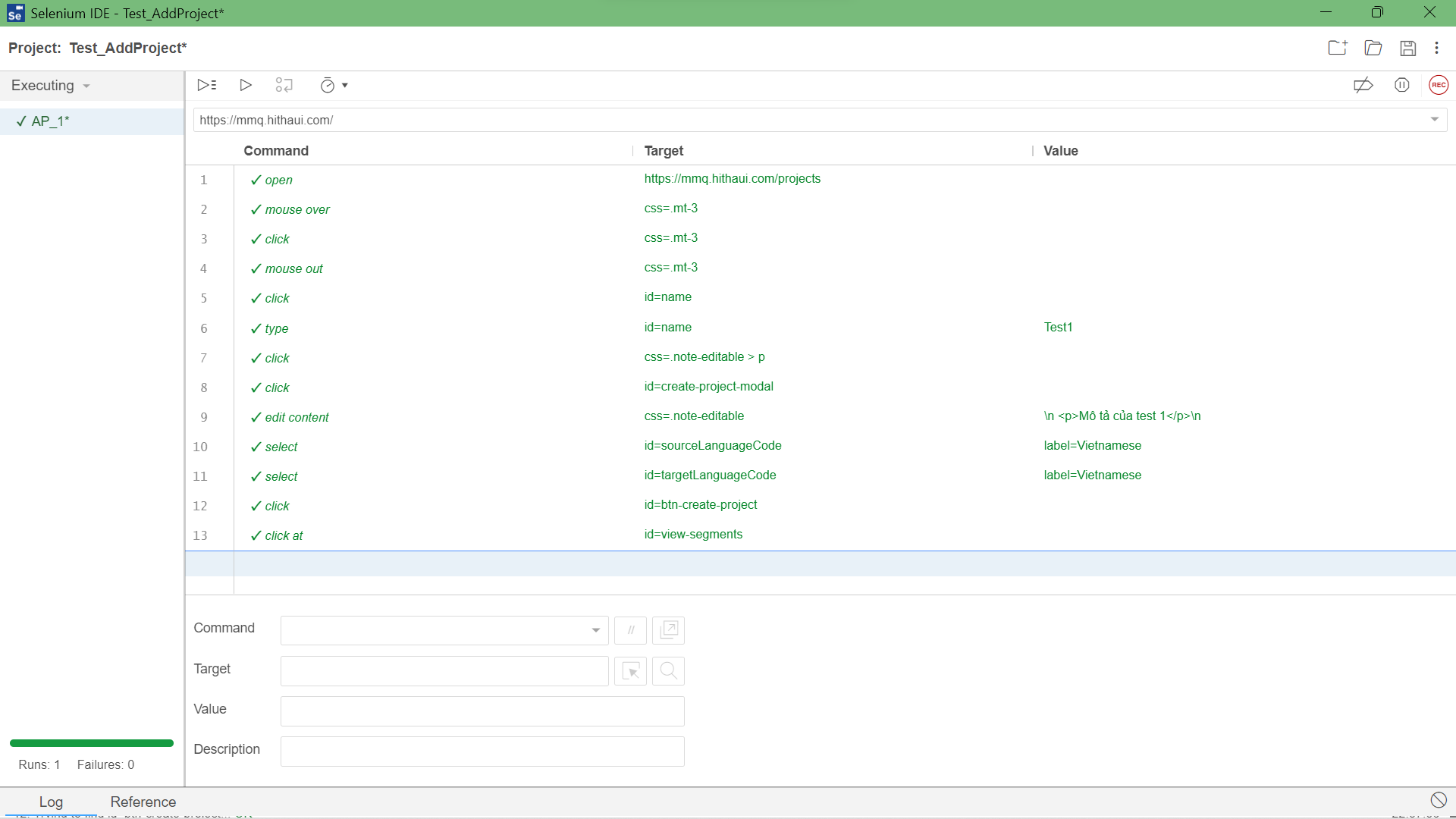
- TH3: Huỷ không thêm file



*Hình 4.3.3.3:* Kết quả test trường hợp dữ liệu cung cấp để thêm tài liệu không thêm file

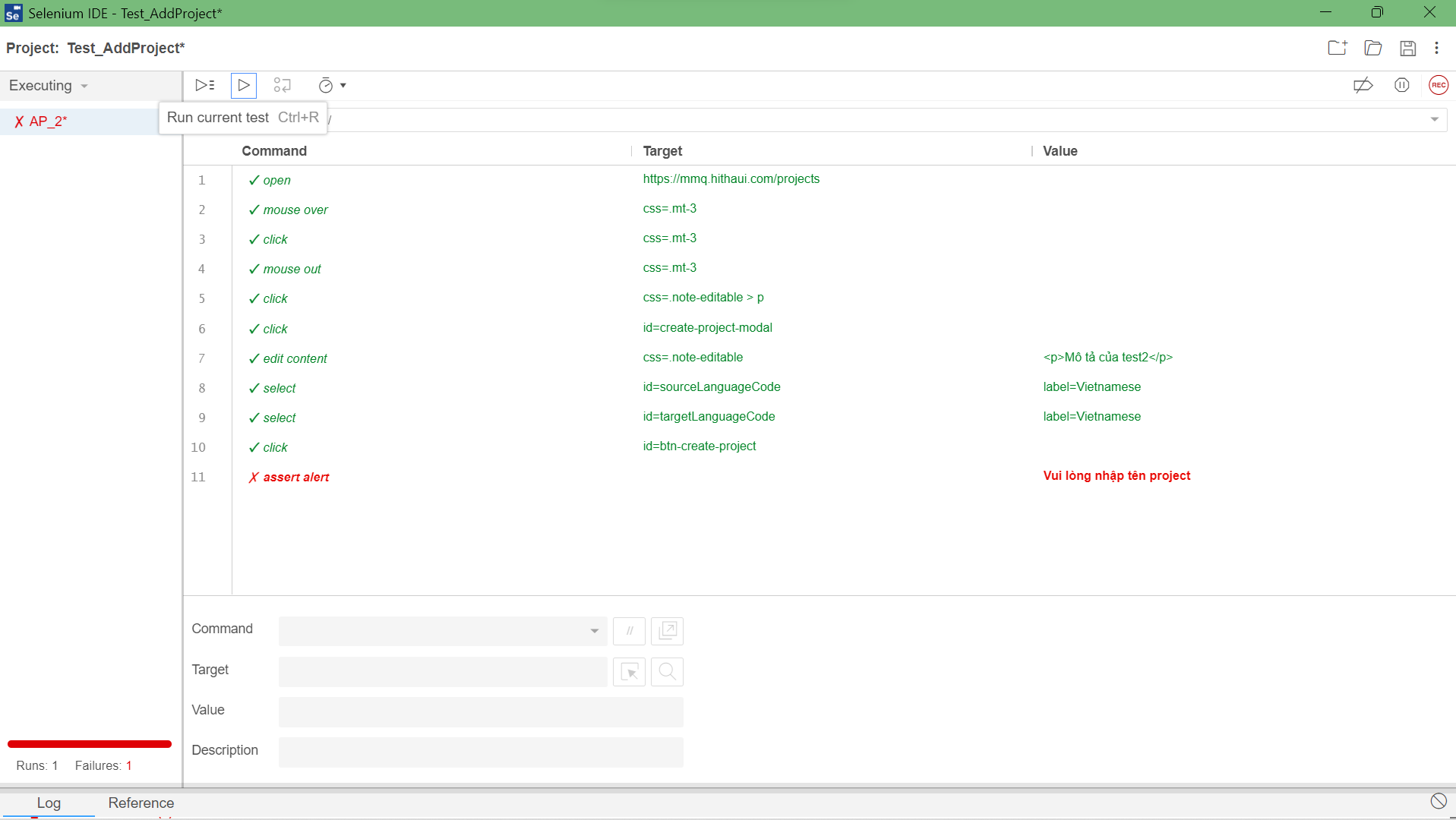
### Kiểm thử tạo project(Minh)

- TH1: Thêm thành công project với đầy đủ tên, mô tả



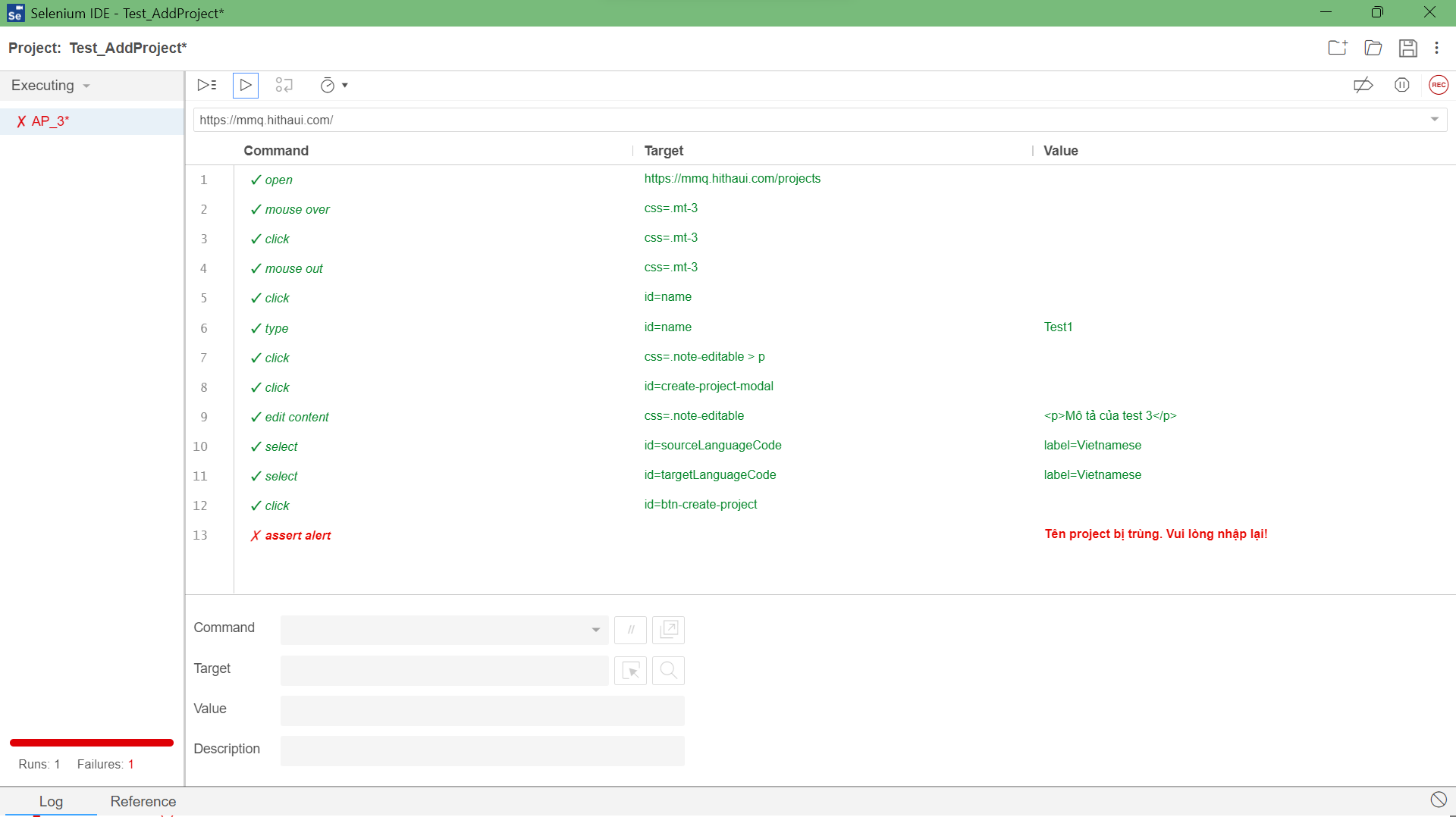
*Hình 4.3.4.1:* Kết quả test trường hợp dữ liệu cung cấp khi tạo project là đủ và đúng

- TH2: Thêm tài liệu nhưng thiếu tên project



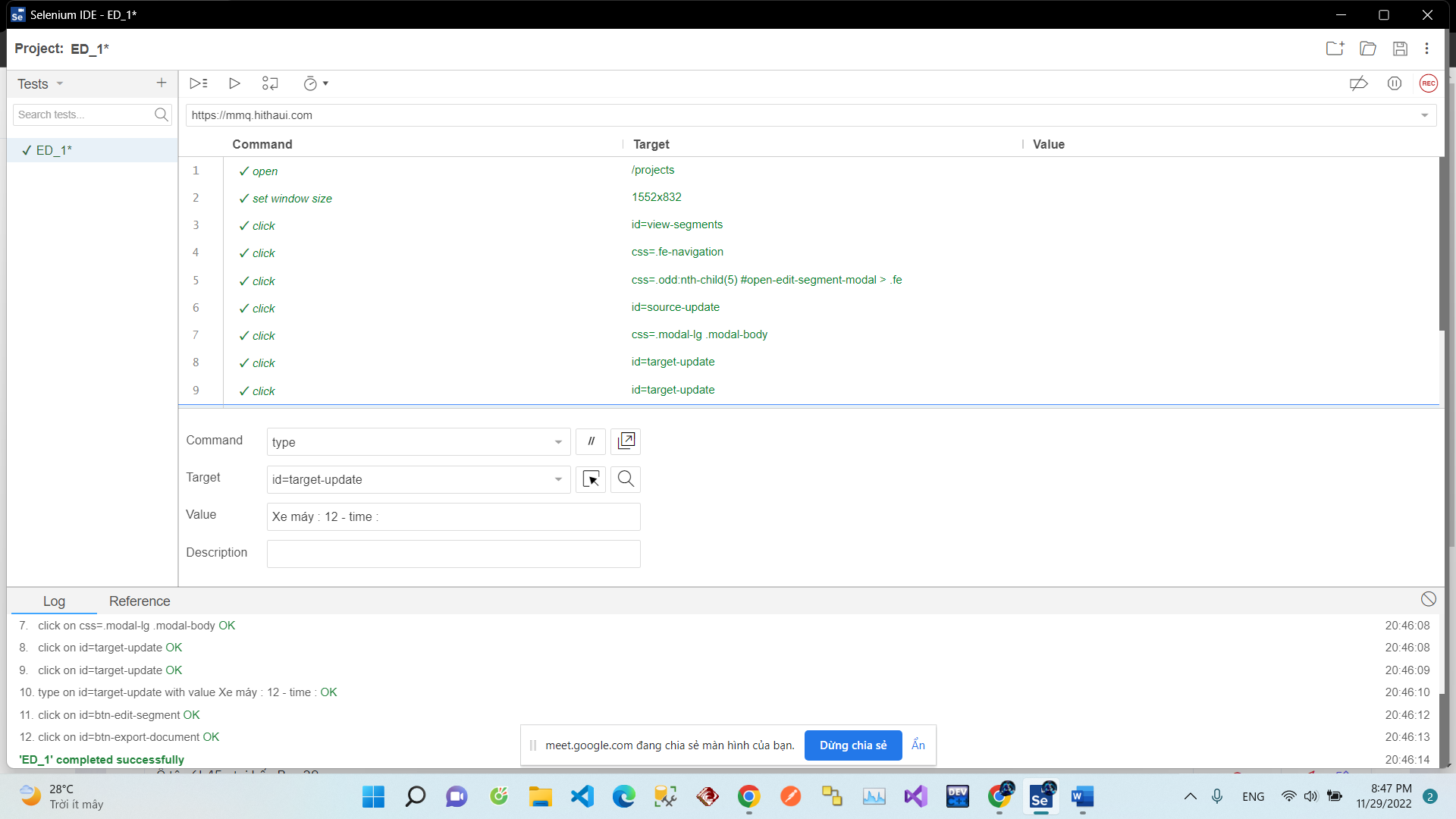
*Hình 4.3.4.2:* Kết quả test trường hợp dữ liệu cung cấp khi tạo project thiếu tên project

- TH3: Tên project bị trùng



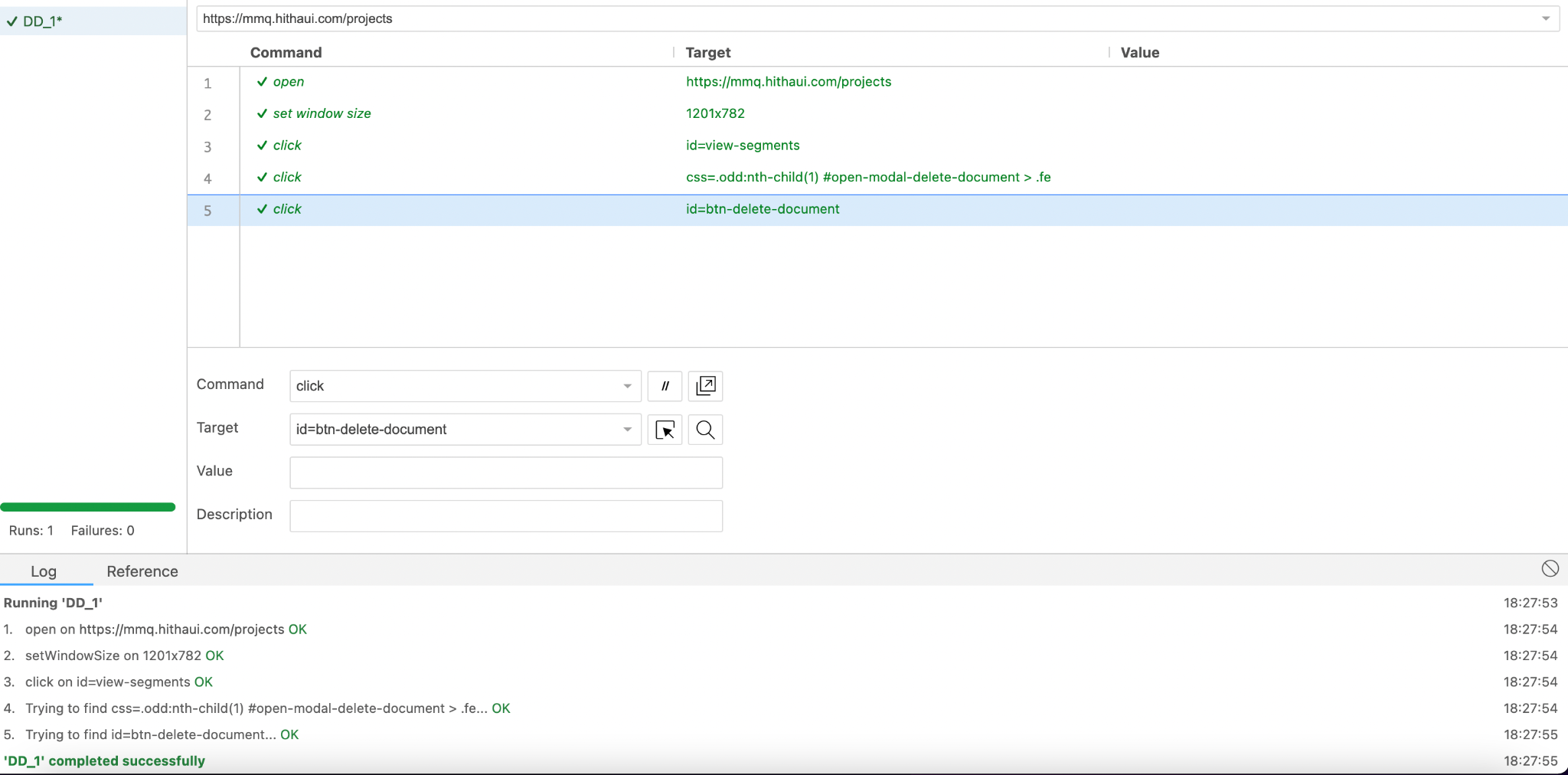
Hình 4.3.4.3: Kết quả test trường hợp dữ liệu cung cấp khi tạo project mà tên project bị trùng

### Kiểm thử sửa tài liệu(Cương)



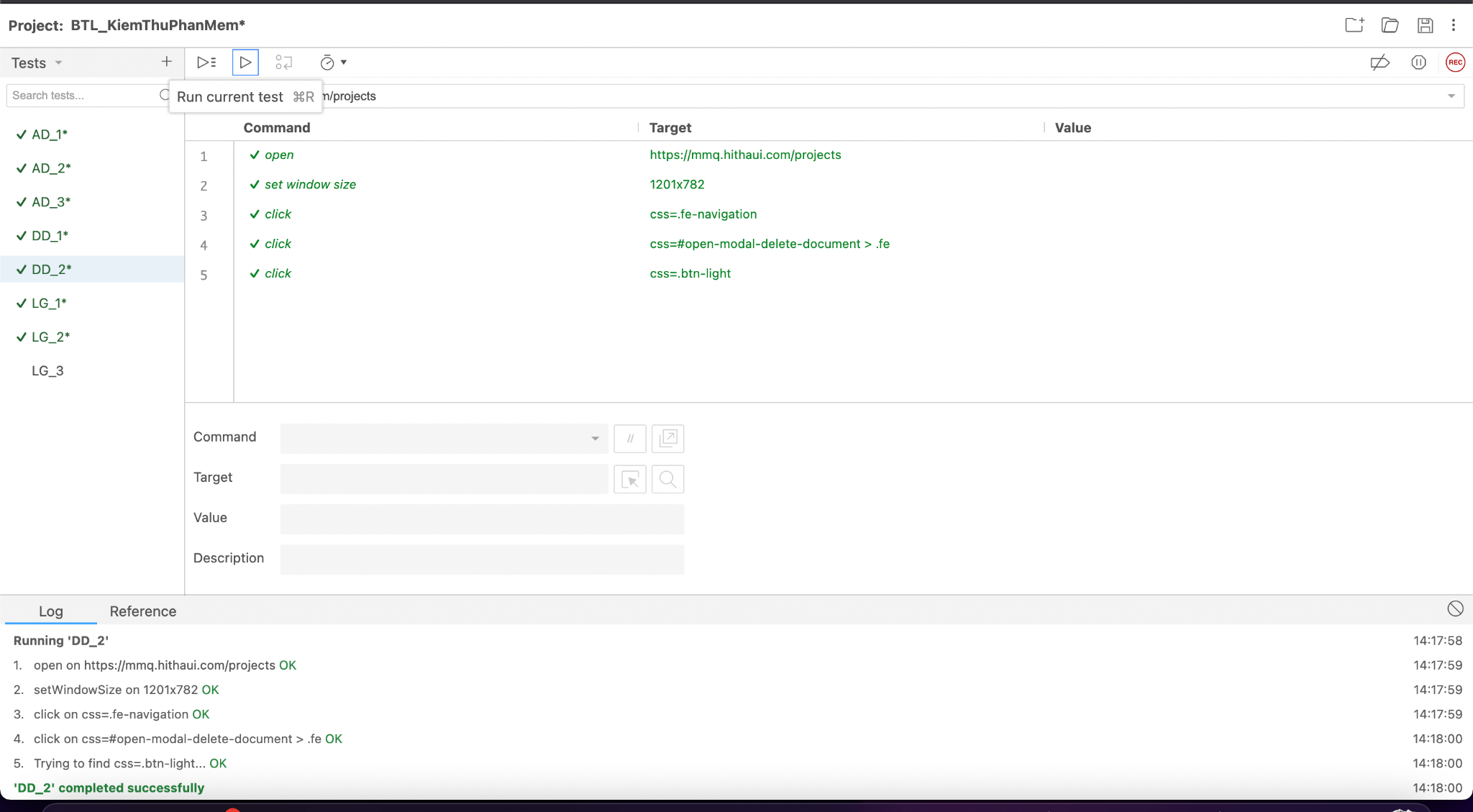
*Hình 4.3.5.1:* Kết quả test trường hợp dữ liệu cung cấp khi sửa tài liệu

### Kiểm thử xoá tài liệu (Cương)

- TH1: Xóa 1 tài liệu

Hình 4.3.6.1: Kết quả test trường hợp xoá tài liệu

- TH2: Bấm nút huỷ không xoá tài liệu



*Hình 4.3.6.2:* Kết quả test trường hợp bấm huỷ không xoá tài liệu

# **Báo cáo, đánh giá kết quả kiểm thử**

- Kiểm thử phần mềm là một hoạt động quan trọng trong đảm bảo chất lượng phần mềm. Việc nghiên cứu lựa chọn các kĩ thuật và chiến lược kiểm thử phần mềm phù hợp giúp cho việc kiểm thử có hiệu quả, giảm chi phí, thời gian. Việc xây dựng tài liệu kiểm thử phần mềm hợp lý sẽ giúp cho việc tổ chức, quản lý việc kiểm thử một cách hiệu quả.

- Thông qua việc kiểm thử chức năng của trang web cơ bản với Selenium IDE thì ta nhận thấy được sự cần thiết của việc kiểm thử cũng như đáng giá được sự phức tạp của hệ thống trang web do người lập trình tạo nên. Kiểm thử tất cả các trường hợp của chức năng hệ thống không phải là đơn giản khi ta không chỉ kiểm tra các trường hợp đúng mà còn phải thử cả các trường hợp có thể dẫn đến các lỗi , ngoại lệ.

- Những vấn đề đạt được

+ Nắm được tổng quan về kiểm thử phần mềm như các phương pháp, kĩ thuật kiểm thử phần mềm nói chung và công dụng, chức năng của kiểm thử Selenium IDE nói riêng.

+ Tìm hiểu và nắm được phương pháp thiết kế các test case trong kiểm thử phần mềm và áp dụng phương pháp đó vào bài toán cụ thể.

+ Nghiên cứu những chức năng và cách hoạt động kiểm thử của Selenium IDE và sử dụng nó để kiểm thử cho trang web hoàn chỉnh.

+ Đưa ra các trường hợp dẫn đến ngoại lệ và đưa ra giải pháp cho các ngoại lệ đó

- Đánh giá kết quả:

+ Cài đặt và triển khai thành công phiên bản mới nhất của Selenium IDE lên trang web cần kiểm thử

+ Lập được các Testcase cần thiết cho việc kiểm thử các chức năng quan trọng cho trang web

+ Hoàn thành việc kiểm thử các ngoại lệ trong các chức năng quan trọng của trang web.

# **Tài liệu tham khảo**

<https://www.guru99.com/test-case.html>

<https://www.guru99.com/what-everybody-ought-to-know-about-test-planing.html>

<https://www.guru99.com/test-plan-for-project.html>

<https://www.guru99.com/web-application-testing.html>

<https://www.selenium.dev/selenium-ide/docs/en/introduction/getting-started>

Tài liệu: Kiểm thử và đảm bảo chất lượng phần mềm (Thạc Bình Cường)