

KHOA KỸ THUẬT VÀ CÔNG NGHỆ
BỘ MÔN CÔNG NGHỆ THÔNG TIN



THỰC TẬP ĐỒ ÁN CƠ SỞ NGÀNH
HỌC KỲ I, NĂM HỌC 2023-2024

TÌM HIỂU CÁC LOẠI TEST CASES TRONG KIỂM THỬ PHẦN MỀM

Giáo viên hướng dẫn:

ThS.Nguyễn Khắc Quốc

Sinh viên thực hiện:

Phan Nguyễn Cảnh Thịnh

MSSV: 110121235

Lớp: DA21TTB

Trà Vinh, tháng 1 năm 2024

KHOA KỸ THUẬT VÀ CÔNG NGHỆ
BỘ MÔN CÔNG NGHỆ THÔNG TIN



THỰC TẬP ĐỒ ÁN CƠ SỞ NGÀNH

HỌC KỲ I, NĂM HỌC 2023-2024

TÌM HIỂU CÁC LOẠI TEST CASES TRONG KIỂM THỬ PHẦN MỀM

Giáo viên hướng dẫn:

ThS.Nguyễn Khắc Quốc

Sinh viên thực hiện:

Phan Nguyễn Cảnh Thịnh

MSSV: 110121235

Lớp: DA21TTB

Trà Vinh, tháng 1 năm 2024

[illegible]

Giáo viên hướng dẫn

This image shows a full page of white paper with horizontal dotted lines, typical of primary school writing paper. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

Thành viên hội đồng

LỜI CẢM ƠN

Xin chân thành cảm ơn thầy ThS.Nguyễn Khắc Quốc đã dành khoảng thời gian vô cùng quý giá để hỗ trợ em trong suốt quá trình thực hiện đồ án này. Thầy đã truyền đạt và cung cấp rất nhiều những kiến thức quan trọng để bản thân em học tập cũng như hoàn thành đồ án này một cách tốt nhất.

Bản thân em luôn cố gắng hết mình trong quá trình thực hiện đồ án nhưng sẽ không tránh được những sai sót xảy ra cũng như những vấn đề chưa được giải quyết. Em mong có thể được quý thầy cô thông cảm và chia sẻ những ý kiến góp ý quý báu.

Em xin chân thành cảm ơn.

MỤC LỤC

MỞ ĐẦU.....	10
1. Lý do chọn đề tài	10
2. Mục tiêu.....	10
3. Đối tượng nghiên cứu	11
4. Phạm vi nghiên cứu	11
CHƯƠNG 1: TỔNG QUAN	12
1.1. TỔNG QUAN VỀ TEST CASES.....	12
1.1.1. Khái niệm Test cases	12
1.1.2. Các thành phần của Test case	12
1.1.3. Các đặc trưng của Test case	12
1.2. PHÂN LOẠI.....	13
CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT	14
2.1. KIỂM THỬ CHỨC NĂNG VÀ KIỂM THỬ CẤU TRÚC.....	14
2.1.1. Kiểm thử hộp đen (Black box Testing) – kiểm thử chức năng.....	14
2.1.2. Kiểm thử hộp trắng (White box Testing) – kiểm thử cấu trúc.....	15
2.1.3. So sánh kiểm thử hộp đen và kiểm thử hộp trắng	17
2.1.4. Kiểm thử dựa vào đặc điểm kỹ thuật.....	17
2.1.5. Kiểm thử trực quan.....	18
2.1.6. Kiểm thử đơn vị.....	18
2.1.7. Kiểm thử tích hợp.....	19
2.1.8. Kiểm thử hệ thống	19
2.2. KIỂM THỬ TĨNH VÀ KIỂM THỬ ĐỘNG.....	19
2.2.1. Kiểm thử tĩnh.....	19
2.2.2. Kiểm thử động.....	20
2.2.3. So sánh kiểm thử tĩnh và kiểm thử động	20
2.3. MỘT SỐ KỸ THUẬT THIẾT KẾ TEST CASES	21

2.4. CHU KỲ KIỂM THỬ MẪU PHẦN MỀM.....	21
CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU	23
3.1. ĐẶT VẤN ĐỀ	23
3.2. TRIỂN KHAI THIẾT KẾ	23
3.2.1. Phân tích	23
3.2.2. Công cụ sử dụng	23
CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU	32
CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	33
DANH MỤC TÀI LIỆU THAM KHẢO	34
PHỤ LỤC	35

DANH MỤC HÌNH ẢNH – BẢNG BIỂU

Hình 2. 1. Kiểm thử hộp đen	14
Hình 2. 2. Kiểm thử hộp trắng.....	15
Hình 3. 1. Các thành phần khi viết test case.....	24
Hình 3. 2. Kịch bản trường hợp kiểm thử thứ nhất	24
Hình 3. 3. Kịch bản trường hợp kiểm thử thứ hai	24
Hình 3. 4. Kịch bản trường hợp kiểm thử thứ ba	24
Hình 3. 5. Kịch bản trường hợp kiểm thử thứ tư.....	24
Hình 3. 6. Kịch bản trường hợp kiểm thử thứ năm	25
Hình 3. 7. Kịch bản trường hợp kiểm thử thứ sáu.....	25
Hình 3. 8. Kịch bản trường hợp kiểm thử thứ bảy	25
Hình 3. 9. Kịch bản trường hợp kiểm thử thứ tám	25
Hình 3. 10. Kịch bản trường hợp kiểm thử thứ chín	25
Hình 3. 11. Giao diện của Selenium IDE khi bắt đầu kiểm thử	26
Hình 3. 12. Thực hiện trường hợp kiểm thử thứ nhất.....	27
Hình 3. 13. Thực hiện trường hợp kiểm thử thứ hai.....	27
Hình 3. 14. Thực hiện trường hợp kiểm thử thứ ba.....	28
Hình 3. 15. Thực hiện trường hợp kiểm thử thứ tư	28
Hình 3. 16. Thực hiện trường hợp kiểm thử thứ năm.....	29
Hình 3. 17. Thực hiện trường hợp kiểm thử thứ sáu	29
Hình 3. 18. Thực hiện trường hợp kiểm thử thứ bảy.....	30
Hình 3. 19. Thực hiện trường hợp kiểm thử thứ tám	30
Hình 3. 20. Trường hợp kiểm thử thứ chín.....	31
Hình 3. 21. Kết quả kiểm thử	32
Bảng 2. 1. So sánh kiểm thử hộp đen và kiểm thử hộp trắng.....	17
Bảng 2. 2. So sánh kiểm thử tĩnh và kiểm thử động	21

TÓM TẮT ĐỒ ÁN CƠ SỞ NGÀNH

Kiểm thử phần mềm là công việc nhằm bảo đảm chất lượng sản phẩm trong quá trình phát triển phần mềm mà trong đó test cases góp một phần quan trọng không kém, nó là một loại tài liệu thể hiện dữ liệu đầu vào, đầu ra và mô tả các bước thực hiện khi kiểm thử phần mềm. Thiết kế test case tốt góp phần đảm bảo phần mềm sẽ được kiểm tra đầy đủ và hiệu quả.

Một số vấn đề nghiên cứu khi tìm hiểu các loại Test cases trong kiểm thử phần mềm:

- Phân loại test case: dựa trên nhiều tiêu chí khác nhau. Cách phân loại phổ biến là phân loại theo phạm vi kiểm thử.
- Kỹ thuật thiết kế test case: có bốn kỹ thuật cơ bản là: phân vùng tương đương (Equivalence Partitioning), phân tích giá trị biên (Boundary Value Analysis), bảng quyết định (Decision Tables) và đoán lỗi (Error Guessing).

Các hướng tiếp cận chính:

- Test case chức năng: kiểm tra chức năng của phần mềm xem có đáp ứng được các yêu cầu được đề ra hay không.
- Test case cấu trúc: kiểm tra các yếu tố cấu trúc như các nhánh, vòng lặp, điều kiện và các thao tác.
- Test case giao diện người dùng (UI): kiểm tra giao diện phần mềm xem có dễ nhìn, dễ sử dụng, đáp ứng nhu cầu thẩm mỹ hay không.
- Test case hiệu năng: kiểm tra khả năng về tốc độ xử lý, độ trễ, khả năng chịu tải.
- Test case tích hợp: kiểm tra sự tương tác giữa các phần của phần mềm.
- Test case hệ thống: kiểm tra tổng thể hệ thống có thể đáp ứng nhiều yếu tố như chức năng, hiệu năng, khả năng sử dụng, ...

MỞ ĐẦU

1. Lý do chọn đề tài

Ngày nay, lĩnh vực Công nghệ thông tin có tiềm năng rất lớn với sự phát triển của chuyển đổi số, các công nghệ mới và nhu cầu nhân lực Công nghệ thông tin ngày càng tăng từ đó kéo theo sự phát triển của rất nhiều lĩnh vực mà điển hình trong số đó là sự phát triển mạnh mẽ về phần mềm.

Phần mềm là một hệ thống được tạo thành từ các chương trình máy tính, dữ liệu và tài liệu liên quan. Phần mềm được sử dụng để thực hiện các nhiệm vụ cụ thể, chẳng hạn như xử lý dữ liệu, điều khiển máy móc hoặc cung cấp thông tin, ...

Kiểm thử phần mềm là một quá trình thực hiện các hoạt động nhằm tìm kiếm và xác nhận các lỗi trong phần mềm. Kiểm thử phần mềm góp một phần quan trọng của quá trình phát triển phần mềm, giúp đảm bảo rằng phần mềm đáp ứng các yêu cầu và không có lỗi.

Phần mềm và kiểm thử phần mềm có mối quan hệ chặt chẽ với nhau. Kiểm thử phần mềm là một phần quan trọng của quá trình phát triển phần mềm, giúp đảm bảo chất lượng phần mềm, giảm thiểu rủi ro và nâng cao hiệu quả hoạt động của phần mềm. Trong đó, Test cases là một trong những công cụ quan trọng nhất trong kiểm thử phần mềm giúp xác định các trường hợp cần kiểm tra, xác định dữ liệu đầu vào và kết quả mong đợi, từ đó giúp đảm bảo chất lượng phần mềm. Mỗi loại test cases có mục đích và phạm vi kiểm tra khác nhau. Việc lựa chọn loại test cases phù hợp phụ thuộc vào các yếu tố như yêu cầu của phần mềm, mục tiêu kiểm thử, ..

Đó cũng chính là lý do mà đề tài “Tìm hiểu các loại Test cases trong kiểm thử phần mềm” được chọn để nghiên cứu và thực hiện trong đồ án này.

2. Mục tiêu

Việc nghiên cứu đề tài này sẽ giúp đạt được những mục tiêu sau:

- Hiểu rõ các loại test cases, từ đó có thể lựa chọn test cases phù hợp với yêu cầu của phần mềm.
- Thiết kế được các test cases hiệu quả, giúp đảm bảo chất lượng phần mềm.

- Tăng cường kỹ năng phân tích và giải quyết vấn đề dữ liệu, thuật toán, logic,...
- Triển khai áp dụng được test case (chức năng hoặc phi chức năng) để kiểm tra chức năng của một phần mềm.

3. Đối tượng nghiên cứu

Các đối tượng nghiên cứu bao gồm:

- Các loại test cases: được phân loại theo các tiêu chí khác nhau.
- Các thông tin cần thiết của một test case: ID, dữ liệu đầu vào, hành động, kết quả.
- Các kỹ thuật thiết kế test cases: phân tích yêu cầu, trường hợp sử dụng, luồng dữ liệu, ...

4. Phạm vi nghiên cứu

- Các kiến thức cơ bản về kiểm thử phần mềm như khái niệm, mục đích, các loại kiểm thử, các giai đoạn kiểm thử.
- Các kiến thức về các loại test cases.

CHƯƠNG 1: TỔNG QUAN

1.1. TỔNG QUAN VỀ TEST CASES

1.1.1. Khái niệm Test cases

- Test cases (bộ kiểm thử) là một tập hợp các hành động được thực thi để xác minh một chức năng, một hệ thống phần mềm có hoạt động đúng hay không. Test cases mô tả dữ liệu đầu vào (input), hành động (action) hoặc sự kiện (event) và một kết quả mong đợi (expected result).

Test case được chia làm 3 trường hợp sau đây:

- Normal case: kiểm thử thông thường.
- Abnormal case: kiểm thử những lỗi xuất hiện một cách bất thường.
- Boundary case: kiểm thử theo mỗi giá trị lỗi tương ứng.

1.1.2. Các thành phần của Test case

- Một Test case bao gồm những thành phần sau:
- Test case ID: mã định danh của test case.
 - Function: chức năng cần kiểm thử.
 - Test Objective: mục tiêu kiểm thử.
 - Test data: số lượng dữ liệu dùng cho quá trình kiểm thử.
 - Pre-condition: điều kiện tiên quyết.
 - Test Steps: các bước thực hiện kiểm thử.
 - Expected Result: kết quả mong đợi.
 - Actual Result: kết quả cuối cùng sau khi kiểm thử, có hai giá trị tham chiếu: pass, fail.
 - Date Execute Test: thời gian thực hiện kiểm thử.

1.1.3. Các đặc trưng của Test case

- Đặc trưng của Test case là những thuộc tính cần có để đảm bảo chất lượng và hiệu quả của quá trình kiểm thử, bao gồm:
- Độ đầy đủ: Test case phải bao phủ tất cả các chức năng và yêu cầu của phần mềm. Điều này giúp đảm bảo rằng phần mềm đã được kiểm tra đầy đủ và không sót lỗi.
 - Độ chính xác: phải mô tả chính xác các bước thực hiện kiểm thử và kết quả mong đợi.

- Độ khả thi: phải thực hiện được trong thực tế.
- Độ rõ ràng: Test case phải được viết rõ ràng và dễ hiểu để bất kì ai cũng có thể hiểu và thực hiện được.
- Độ ngắn gọn: không nên quá dài dòng hoặc rườm rà. Điều này giúp dễ quản lý và theo dõi test case.
- Độ hiệu quả: có thể phát hiện lỗi một cách nhanh chóng và hiệu quả.

1.2. PHÂN LOẠI

- Function Test Cases: kiểm thử chức năng.
- Unit Test Cases: kiểm thử đơn vị.
- User Interface Test Cases: kiểm thử giao diện người dùng.
- Performance Test Cases: kiểm thử hiệu năng.
- Integration Test Cases: kiểm thử tích hợp.
- System Test Cases: kiểm thử hệ thống.
- Usability Test Cases: kiểm thử khả năng sử dụng.
- Database Test Cases: kiểm thử cơ sở dữ liệu.
- Security Test Cases: kiểm thử bảo mật.
- User Acceptance Cases: kiểm thử khả năng chấp nhận người dùng.
- Static Test Cases: kiểm thử tĩnh.
- Dynamic Test Cases: kiểm thử động.

CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT

Test cases có thể được phân loại theo nhiều cách, dựa trên nhiều tiêu chí khác nhau. Mà một trong những cách phổ biến nhất là phân loại dựa trên phạm vi kiểm thử.

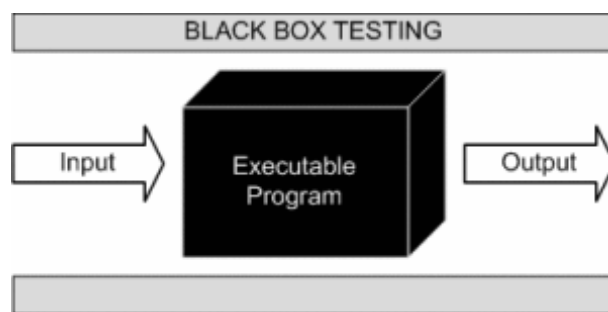
Dựa theo cách phân loại này, sẽ trình bày các loại test cases sau đây:

2.1. KIỂM THỬ CHỨC NĂNG VÀ KIỂM THỬ CẤU TRÚC

- Có 2 loại kiểm thử phần mềm phổ biến bao gồm: kiểm thử hộp đen (hay còn được gọi là Black box Testing) và kiểm thử hộp trắng (hay còn được gọi là White box Testing).

2.1.1. Kiểm thử hộp đen (Black box Testing) – kiểm thử chức năng

- Kiểm thử hộp đen là một phương pháp kiểm thử phần mềm mà người kiểm thử (tester) không biết được cấu tạo bên trong của phần mềm, không quan tâm về cấu trúc bên trong của nó mà chỉ tập trung vào dữ liệu đầu vào và đầu ra của phần mềm.



Hình 2. 1. Kiểm thử hộp đen

- Một số kỹ thuật kiểm thử hộp đen gồm:
 - Kỹ thuật đường dẫn kiểm thử: sử dụng phân tích đường dẫn để có thể xác định được các dòng mã cần được kiểm tra.
 - Kỹ thuật trường hợp sử dụng: sử dụng các trường hợp sử dụng để xác định đầu vào, đầu ra cần kiểm tra.
 - Kỹ thuật kiểm thử hồi quy: là kiểm tra lại phần mềm sau khi đã có các thay đổi để có thể đảm bảo rằng các thay đổi đó không gây ra lỗi ảnh hưởng đến phần mềm.
- Phương pháp kiểm thử hộp đen có thể tìm ra được một số lỗi sau đây:
 - Chức năng của phần mềm không đúng hoặc bị thiếu.
 - Lỗi giao diện.
 - Lỗi bên trong cấu trúc dữ liệu hoặc lỗi truy cập cơ sở dữ liệu.

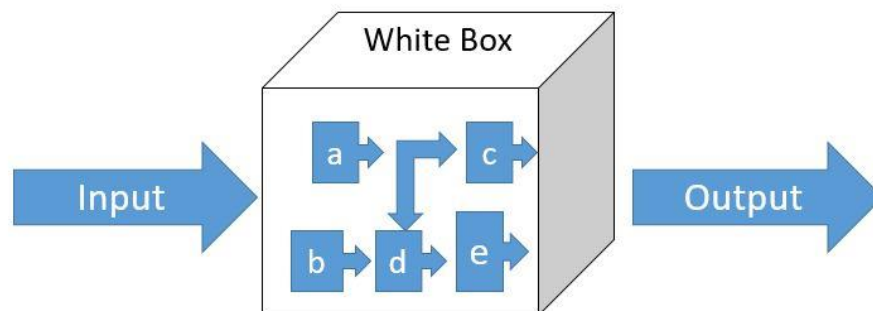
- Lỗi hành vi hoặc lỗi hiệu suất.

Ưu điểm, nhược điểm của kiểm thử hộp đen

- Kiểm thử hộp đen có những ưu điểm như sau:
 - Dễ dàng sử dụng và không cần hiểu biết rõ về mã nguồn bên trong.
 - Tester không cần phải biết lập trình, không bị ràng buộc bởi code.
 - Có ưu thế về kiểm tra chức năng của phần mềm.
 - Toàn bộ yêu cầu về hệ thống được kiểm thử chính xác.
 - Khi đã xác định được yêu cầu, có thể thiết kế kịch bản kiểm thử rất nhanh.
- Bên cạnh đó, vẫn còn tồn tại một số hạn chế như sau:
 - Yêu cầu về khối lượng mẫu dữ liệu đầu vào rất lớn.
 - Khó thiết kế test case khi không có thông số cụ thể.
 - Có thể bỏ sót lỗi do không nắm rõ cấu trúc bên trong phần mềm.

2.1.2. Kiểm thử hộp trắng (White box Testing) – kiểm thử cấu trúc

- Kiểm thử hộp trắng là một phương pháp kiểm thử phần mềm mà người kiểm thử có quyền truy cập vào cấu trúc bên trong, thiết kế và các mã nguồn của phần mềm để các nhà phát triển có thể cải thiện thiết kế cũng như khả năng sử dụng và bảo mật phần mềm.



Hình 2. 2. Kiểm thử hộp trắng

- Một số kỹ thuật kiểm thử hộp trắng gồm:
 - Kỹ thuật kiểm soát luồng: kiểm tra cách mà mã kiểm soát luồng của chương trình.
 - Kỹ thuật phân tích dữ liệu: kiểm tra cách mà mã xử lý dữ liệu.
 - Kỹ thuật kiểm tra lỗi: cố gắng tìm các lỗi có trong mã.

Kiểm thử hộp trắng có thể tìm ra nhiều loại lỗi, ví dụ như:

- **Lỗi logic:** là những sai sót thuật toán hoặc logic của chương trình, phần mềm. Có thể tìm ra lỗi này bằng cách kiểm tra các đường dẫn khác nhau của chương trình và phải đảm bảo chúng vẫn cho các kết quả là chính xác.

Ví dụ: hàm cộng hai số nhưng trả KQ sai; vòng lặp không kết thúc; ...

- **Lỗi dòng mã:** là những lỗi đánh máy, lỗi cú pháp hoặc lỗi logic nhỏ trong mã của phần mềm. Có thể tìm ra những lỗi này bằng cách kiểm tra từng dòng mã một và phải đảm bảo chúng vẫn hoạt động một cách chính xác.

Ví dụ: biến được khai báo nhưng không dùng; toán tử được sử dụng sai; câu lệnh được viết không chính xác, ...

- **Lỗi không thực thi:** lỗi xảy ra khi các phần của chương trình không thể thực thi. Có thể tìm ra lỗi bằng cách kiểm tra các đường dẫn khác nhau của chương trình và phải đảm bảo tất cả các đường dẫn đó đều được thực thi tối thiểu là một lần.

Ví dụ: một phần chương trình được viết nhưng không hề được gọi và sử dụng; một phần của chương trình chỉ được gọi và sử dụng trong trường hợp rất hiếm, ...

- **Lỗi ẩn:** những lỗi này sẽ thường khó phát hiện nếu sử dụng phương pháp kiểm thử hộp đen nhưng kiểm tra hộp trắng có thể phát hiện bằng cách sử dụng các kiến thức liên quan đến cấu trúc bên trong của mã nguồn.

Ví dụ: lỗi trong một hàm phụ gây ảnh hưởng đến hàm chính; lỗi trong cách xử lý ngoại lệ dẫn đến gây ra lỗi nghiêm trọng, ...

Ưu điểm, nhược điểm của kiểm thử hộp trắng

➤ Kiểm thử hộp trắng có những ưu điểm như sau:

- Có khả năng kiểm tra kỹ lưỡng và bao phủ hầu hết các đường dẫn.
- Phát hiện được hầu hết các lỗi hoặc thiếu sót xảy ra bên trong phần mềm.
- Có thể được sử dụng để kiểm tra các yêu cầu cả chức năng và phi chức năng của phần mềm.
- Quá trình kiểm tra đơn giản và mang lại hiệu quả cao.

➤ Bên cạnh đó vẫn còn tồn tại một số hạn chế như sau:

- Tester phải có năng lực cao và kiến thức chuyên sâu về lập trình.
- Khó khăn trong việc bảo trì các tập lệnh thử nghiệm.
- Tốn nhiều thời gian để tạo nhiều trường hợp kiểm thử đầy đủ và hiệu quả.

2.1.3. So sánh kiểm thử hộp đen và kiểm thử hộp trắng

Đặc điểm	Kiểm thử hộp đen	Kiểm thử hộp trắng
Kiến thức về mã nguồn	Không cần thiết	Cần thiết
Người thực hiện	Chỉ cần là người kiểm thử không nhất thiết phải là lập trình viên.	Lập trình viên
Kỹ thuật kiểm thử thường được sử dụng	Đường dẫn kiểm thử, trường hợp sử dụng, kiểm thử hồi quy.	Kiểm soát luồng, phân tích dữ liệu, kiểm tra lỗi.
Ưu điểm	Dễ dàng sử dụng, không cần hiểu biết về mã nguồn và có thể được thực hiện bởi các tester không phải lập trình viên.	Phát hiện được nhiều lỗi hơn, có thể được sử dụng để kiểm tra các yêu cầu cả về chức năng và phi chức năng của phần mềm.
Nhược điểm	Khó thiết kế kịch bản kiểm thử do không có thông số cụ thể. Có thể bỏ sót lỗi do không hiểu rõ về cấu trúc bên trong phần mềm.	Yêu cầu tester có kiến thức hiểu biết cao về mã nguồn. Tốn kém thời gian để tạo các trường hợp kiểm thử hiệu quả.
Thời điểm sử dụng	Thường sử dụng trong giai đoạn đầu của quá trình phát triển chương trình, phần mềm.	Thường sử dụng trong các giai đoạn sau của quá trình phát triển chương trình, phần mềm.

Bảng 2. 1. So sánh kiểm thử hộp đen và kiểm thử hộp trắng

2.1.4. Kiểm thử dựa vào đặc điểm kỹ thuật

Kiểm thử dựa trên đặc điểm kỹ thuật là phương pháp dùng để có thể xác minh được rằng phần mềm đáp ứng được các yêu cầu được nêu trong đặc tả yêu cầu. Đây là một phương pháp kiểm thử khá quan trọng vì có giúp bảo đảm rằng phần mềm có thể đáp ứng được các nhu cầu của người dùng nhưng lại là một rủi ro vì nó không đủ để giúp bảo vệ phần mềm cũng như có khả năng chống chọi lại các trường hợp lỗi phức tạp nghiêm trọng.

- Một số ưu điểm của phương pháp kiểm thử dựa trên đặc điểm kỹ thuật:
 - Bảo đảm cho phần mềm đáp ứng các yêu cầu được đặc tả và yêu cầu của người dùng.
 - Có ưu thế lớn khi kiểm thử cho từng đơn vị.
 - Được sử dụng để kiểm tra cho nhiều loại phần mềm khác nhau.
- Bên cạnh đó là một số nhược điểm như:
 - Không thể phát hiện ra những lỗi không được đề cập trong đặc tả yêu cầu.
 - Không thể thực hiện hoàn toàn các kiểm thử có mức độ cao hơn.

2.1.5. Kiểm thử trực quan

Kiểm thử trực quan là phương pháp thường tập trung vào UI (giao diện người dùng), sử dụng cảm giác, giác quan của người kiểm thử để xác minh các lỗi có thể có, từ đó đảm bảo sản phẩm cuối cùng đáp ứng được các tiêu chuẩn, yêu cầu và mong muốn của người dùng.

Kiểm thử trực quan được hỗ trợ tối đa từ các công cụ lưu trữ dạng video để có thể cung cấp những minh chứng rõ ràng nhất cho việc kiểm tra các nguyên nhân dẫn đến lỗi.

Một số công cụ có thể hỗ trợ cho kiểm thử trực quan:

- Các công cụ ghi và phát lại: người kiểm thử có thể ghi lại quá trình tương tác giữa người dùng với UI rồi sau đó phát lại để dễ dàng kiểm tra.
- Các công cụ phân tích ảnh chụp màn hình: người kiểm thử có thể chụp màn hình của UI rồi dựa trên các ảnh đó để tìm lỗi.
- Các công cụ kiểm tra về khả năng tiếp cận: người kiểm thử có thể kiểm tra UI có đáp ứng được về khả năng tiếp cận của người dùng hay không.

2.1.6. Kiểm thử đơn vị

Kiểm thử đơn vị là phương pháp tập trung vào việc kiểm tra các đơn vị nhỏ nhất của hệ thống ví dụ như các hàm, thủ tục hoặc các lớp, ... thường được sử dụng để bảo đảm các đơn vị của hệ thống hoạt động chính xác.

Ví dụ: nếu một hệ thống có hàm “Thêm tài khoản” thì bộ test case đơn vị cho hàm này sẽ phải kiểm tra các trường hợp sau đây:

- Thêm tài khoản không thành công vì tên người dùng đã tồn tại.
- Thêm tài khoản không thành công vì nhập dữ liệu tên người dùng không hợp lệ.
- Thêm tài khoản thành công.

2.1.7. Kiểm thử tích hợp

Kiểm thử tích hợp là phương pháp tập trung vào kiểm tra sự tương tác giữa các giao diện hoặc giữa các thành phần của hệ thống, thường được sử dụng để bảo đảm các giao diện hoặc thành phần đó có thể giao tiếp, tương tác và hoạt động cùng nhau một cách trơn tru nhất.

Ví dụ: khi một hệ thống bao gồm hai thành phần “Đăng nhập” và “Quản lý tài khoản” thì bộ test case tích hợp cho việc này sẽ kiểm tra các trường hợp sau đây:

- Đăng nhập vào thành công vào hệ thống bằng thành phần “Đăng nhập”
- Sử dụng thành phần “Quản lý tài khoản” để quản lý các tài khoản được đăng nhập thành công.
- Thoát khỏi hệ thống.

2.1.8. Kiểm thử hệ thống

Kiểm thử hệ thống là phương pháp được thực hiện để kiểm tra toàn bộ hệ thống như một khối tổng thể về chức năng, hiệu suất, khả năng sử dụng, mở rộng, ... bảo đảm hệ thống đáp ứng các yêu cầu đã được thiết lập.

Tương tự như ví dụ trên, nhưng bộ test case hệ thống sẽ kiểm tra các trường hợp như: truy cập trình duyệt web, đăng nhập vào hệ thống, tạo tài khoản mới, xóa tài khoản, ...

2.2. KIỂM THỬ TĨNH VÀ KIỂM THỬ ĐỘNG

2.2.1. Kiểm thử tĩnh

- *Kiểm thử tĩnh*: là loại test case dùng để kiểm tra cú pháp code, xem xét thiết kế, tài liệu, phân tích yêu cầu chức năng mà không cần phải thực thi chương trình hay phần mềm.
- Có 3 loại kiểm thử tĩnh, bao gồm:
 - Informal review: là kiểm thử không chính thức, chỉ thực hiện xem xét tài liệu.
 - Walkthroughs: là chia sẻ thông tin cũng như hướng dẫn và giải thích cho người kiểm thử được hiểu rõ về chương trình hay phần mềm.
 - Technical review: là đánh giá và thảo luận về các phần liên quan đến kỹ thuật.
- Một số vấn đề cần lưu ý về kiểm thử tĩnh:
 - Cần phân tích kỹ lưỡng các yêu cầu chức năng.
 - Xác định rõ các trường hợp cần kiểm thử.
 - Lựa chọn dữ liệu đầu vào thích hợp.
 - Cần mô tả test case một cách rõ ràng và chuẩn xác.

2.2.2. Kiểm thử động

- *Kiểm thử động*: là loại test case dùng để kiểm tra dựa vào cấu trúc bên trong thông qua thực thi chương trình hay phần mềm.
- Có thể chia kiểm thử động thành các nhóm sau:
 - Nhóm kiểm thử những yếu tố bên ngoài: các kỹ thuật như phân vùng tương đương, phân tích giá trị biên, chuyển đổi trạng thái, trường hợp sử dụng.
 - Nhóm kiểm tra cấu trúc bên trong và cách thức hoạt động: các kỹ thuật như kiểm thử câu lệnh, kiểm thử quyết định, điều kiện hoặc đa điều kiện.
 - Nhóm kiểm tra năng lực của người kiểm thử như khả năng thăm dò hoặc phỏng đoán lỗi.
- Một số vấn đề cần lưu ý về kiểm thử động:
 - Phân tích và nắm vững về cấu trúc bên trong của phần mềm.
 - Xác định được lỗi cần kiểm tra.
 - Lựa chọn các giá trị cho dữ liệu đầu vào phù hợp.
 - Mô tả test case rõ ràng, thực hiện đúng quy trình và chính xác.

2.2.3. So sánh kiểm thử tĩnh và kiểm thử động

- ❖ Giống nhau: hai phương pháp trên đều dùng để thẩm định và xác minh rằng phần mềm hoạt động đúng như mong đợi.

❖ Khác nhau:

Tiêu chí	Kiểm thử tĩnh	Kiểm thử động
	Yêu cầu chức năng	Cấu trúc bên trong
Kiến thức cần có	Không yêu cầu hiểu biết về cấu trúc bên trong phần mềm.	Yêu cầu hiểu biết rõ về cấu trúc bên trong phần mềm.
Độ khó	Dễ hiểu và viết	Khó hiểu và viết
Thời gian	Thường được sử dụng trong các dự án kiểm thử quy mô nhỏ, thời gian ngắn.	Chỉ được sử dụng ở các dự án kiểm thử quy mô lớn và có thời gian phù hợp.
Độ bao phủ	Không thể kiểm tra được hết tất cả trường hợp.	Có thể kiểm tra được tất cả trường hợp lỗi ngay cả lỗi thiết kế.

Bảng 2. 2. So sánh kiểm thử tĩnh và kiểm thử động

2.3. MỘT SỐ KỸ THUẬT THIẾT KẾ TEST CASES

Hiện nay, có rất nhiều kỹ thuật thiết kế test cases được sử dụng, nhìn chung vẫn có thể phân loại các kỹ thuật này theo những nhóm khác nhau. Tuy nhiên, có 4 kỹ thuật thiết kế test cases phổ biến nhất, đó là:

- Phân vùng tương đương (Equivalence Partitioning): có nghĩa là phân chia dữ liệu đầu vào thành các nhóm tương đương nhau rồi sau đó sẽ thiết kế test case riêng cho từng nhóm.
- Phân tích giá trị biên (Boundary Value Analysis): thiết kế test case cho các giá trị biên của dữ liệu đầu vào. Giá trị biên là giá trị nhỏ nhất hoặc lớn nhất mà một đầu vào có thể nhận được trước khi nó vượt qua ranh giới của miền giá trị hợp lệ.
- Bảng quyết định (Decision Tables): sẽ sử dụng bảng quyết định để thiết kế test case cho các trường hợp phức tạp với nhiều trường dữ liệu.
- Đoán lỗi (Error Guessing): sẽ dựa vào kiến thức và kinh nghiệm của người kiểm thử để đoán các lỗi có thể có trong phần mềm và thiết kế test case để kiểm tra các lỗi đó.

2.4. CHU KỲ KIỂM THỬ MẪU PHẦN MỀM

Chu kỳ kiểm thử mẫu được sử dụng ở giai đoạn phát triển phần mềm ngay sau khi đã xác định rõ những yêu cầu của phần mềm.

- Phân tích yêu cầu: phân tích và hiểu rõ các yêu cầu của phần mềm, người kiểm thử cần xác định những trường hợp của thiết kế cần được kiểm tra cũng như thông số cụ thể.
- Lập kế hoạch kiểm thử: cần lên kế hoạch chi tiết về những hoạt động sẽ làm như mục tiêu kiểm thử, phạm vi kiểm thử, phương pháp kiểm thử, ...
- Phát triển kịch bản kiểm thử: có thể có nhiều kịch bản, nhưng kịch bản kiểm thử tốt nhất cần có đầy đủ các hướng dẫn, các quy trình, test case, dữ liệu sẽ được sử dụng.
- Thực thi kiểm thử: căn cứ vào kế hoạch để thực hiện kiểm thử, người kiểm thử sẽ sử dụng nhiều phương pháp kiểm thử khác nhau để tìm lỗi trong phần mềm.
- Đánh giá kết quả: người kiểm thử sẽ phân tích và đánh giá các lỗi nếu được phát hiện trong quá trình kiểm thử.
- Đóng chu kỳ kiểm thử: sau khi đã hoàn thành kiểm thử, người kiểm thử sẽ tổng hợp số liệu cụ thể để báo cáo đến các bên liên quan cũng như cho biết rằng phần mềm có sẵn sàng đến tay người dùng hay không.

Các giai đoạn này được người kiểm thử thực hiện lặp đi lặp lại, tức là cho phép người kiểm thử quay lại các giai đoạn trước khi cần thiết.

Chu kỳ kiểm thử mẫu là một quy trình điển hình để kiểm thử, là một bộ khung chung để các nhà kiểm thử dựa vào đó thực hiện kiểm thử phần mềm. Họ hoàn toàn có thể điều chỉnh chu kỳ kiểm thử mẫu này sao cho phù hợp với nhu cầu và thực tiễn của dự án.

CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU

3.1. ĐẶT VẤN ĐỀ

Với vai trò của một người kiểm thử, việc lựa chọn vấn đề cần kiểm tra để viết và thực hiện test case trở nên quen thuộc. Chương này sẽ trình bày quá trình thực hiện kiểm thử các trường hợp khi người dùng đăng nhập vào hệ thống Facebook.

Ví dụ: nếu hệ thống có một nút chức năng “Đăng nhập”, thì một bộ test case kiểm thử chức năng này sẽ bao gồm các test case kiểm tra các trường hợp sau:

- Nhập đúng tên người dùng và mật khẩu
- Nhập sai tên người dùng
- Nhập sai mật khẩu
- Nhập tên người dùng và mật khẩu bỏ trống

3.2. TRIỂN KHAI THIẾT KẾ

3.2.1. Phân tích

Để thực hiện được vấn đề đã nêu, cần phải xác định rõ các trường hợp có thể xảy ra khi người dùng nhập tài khoản và mật khẩu để truy cập vào Facebook. Từ đó có thể suy ra, sẽ xuất hiện 9 trường hợp cụ thể cho việc đăng nhập này bao gồm:

- Trường hợp một: người dùng nhập đúng tài khoản, nhưng nhập sai mật khẩu.
- Trường hợp hai: người dùng nhập đúng tài khoản, nhưng không nhập mật khẩu.
- Trường hợp ba: người dùng nhập sai tài khoản, nhưng nhập đúng mật khẩu.
- Trường hợp bốn: người dùng nhập sai tài khoản và mật khẩu.
- Trường hợp năm: người dùng nhập sai tài khoản và không nhập mật khẩu.
- Trường hợp sáu: người dùng không nhập tài khoản, nhưng nhập đúng mật khẩu.
- Trường hợp bảy: người dùng không nhập tài khoản và nhập sai mật khẩu.
- Trường hợp tám: người dùng không nhập tài khoản và mật khẩu.
- Trường hợp chín: người dùng nhập đúng cả tài khoản và mật khẩu.

3.2.2. Công cụ sử dụng

- Excel: dùng để xây dựng một kịch bản kiểm thử đầy đủ cả 9 trường hợp khi người dùng sử dụng tài khoản và mật khẩu để đăng nhập vào Facebook.

Test ID	Test Objective	Test Pre-conditions	Test Steps	Expected Result
---------	----------------	---------------------	------------	-----------------

Hình 3. 1. Các thành phần khi viết test case

Trong đó:

- Test ID: mã định danh cho mỗi trường hợp test case.
- Test Objective: mục tiêu khi kiểm thử.
- Test Pre-conditions: điều kiện tiên quyết phải có.
- Test Steps: các bước thực hiện.
- Expected Result: kết quả mong đợi.

❖ **Thiết kế kịch bản cho từng trường hợp kiểm thử:**

Test ID	Test Objective	Test Pre-conditions	Test Steps	Expected Result
TC01	Kiểm tra trường hợp đăng nhập vào Facebook thất bại khi nhập chính xác email/phone number và với password không chính xác khi click vào nút "Log in"	Người dùng đã có tài khoản trên hệ thống	1.Vào trang "https://www.facebook.com/" 2.Nhập đúng email/phone number 3.Nhập sai password 4.Click vào nút "Log in"	Thông báo "Mật khẩu bạn đã nhập không chính xác. Đã quên mật khẩu?". Hệ thống tiếp tục yêu cầu nhập password.

Hình 3. 2. Kịch bản trường hợp kiểm thử thứ nhất

Test ID	Test Objective	Test Pre-conditions	Test Steps	Expected Result
TC02	Kiểm tra trường hợp đăng nhập vào Facebook thất bại khi nhập chính xác email/phone number và với password để trống khi click vào nút "Log in"	Người dùng đã có tài khoản trên hệ thống	1.Vào trang "https://www.facebook.com/" 2.Nhập đúng email/phone number 3.Bỏ trống password 4.Click vào nút "Log in"	Thông báo "Mật khẩu bạn đã nhập không chính xác. Đã quên mật khẩu?". Hệ thống tiếp tục yêu cầu nhập password.

Hình 3. 3. Kịch bản trường hợp kiểm thử thứ hai

Test ID	Test Objective	Test Pre-conditions	Test Steps	Expected Result
TC03	Kiểm tra trường hợp đăng nhập vào Facebook thất bại khi nhập không chính xác email/phone number nhưng nhập chính xác password khi click vào nút "Log in"	Người dùng đã có tài khoản hợp lệ trên hệ thống	1.Vào trang "https://www.facebook.com/" 2.Nhập sai email/phone number 3.Nhập đúng password 4.Click vào nút "Log in"	Thông báo "Địa chỉ email bạn đã nhập không được kết nối với tài khoản. Tạo một tài khoản Facebook mới.". Hệ thống tiếp tục yêu cầu nhập email/phone number và password.

Hình 3. 4. Kịch bản trường hợp kiểm thử thứ ba

Test ID	Test Objective	Test Pre-conditions	Test Steps	Expected Result
TC04	Kiểm tra trường hợp đăng nhập vào Facebook thất bại khi nhập không chính xác email/phone và password khi click vào nút "Log in"	Người dùng đã có tài khoản hợp lệ trên hệ thống	1.Vào trang "https://www.facebook.com/" 2.Nhập sai email/phone number 3.Nhập sai password 4.Click vào nút "Log in"	Thông báo "Địa chỉ email bạn đã nhập không được kết nối với tài khoản. Tạo một tài khoản Facebook mới.". Hệ thống tiếp tục yêu cầu nhập email/phone number và password.

Hình 3. 5. Kịch bản trường hợp kiểm thử thứ tư

Test ID	Test Objective	Test Pre-conditions	Test Steps	Expected Result
TC05	Kiểm tra trường hợp đăng nhập vào Facebook thất bại khi nhập không chính xác email/phone number và bỏ trống mật khẩu khi click vào nút "Log in"	Người dùng đã có tài khoản hợp lệ trên hệ thống	1.Vào trang "https://www.facebook.com/" 2.Nhập sai email/phone number 3.Bỏ trống password 4.Click vào nút "Log in"	Thông báo "Địa chỉ email bạn đã nhập không được kết nối với tài khoản. Tạo một tài khoản Facebook mới.". Hệ thống tiếp tục yêu cầu nhập email/phone number và password.

Hình 3. 6. Kịch bản trường hợp kiểm thử thứ năm

Test ID	Test Objective	Test Pre-conditions	Test Steps	Expected Result
TC06	Kiểm tra trường hợp đăng nhập vào Facebook thất bại khi bỏ trống email/phone number nhưng nhập chính xác mật khẩu khi click vào nút "Log in"	Người dùng đã có tài khoản trên hệ thống	1.Vào trang "https://www.facebook.com/" 2.Bỏ trống email/phone number 3.Nhập đúng password 4.Click vào nút "Log in"	Thông báo "Địa chỉ email hoặc số điện thoại di động bạn đã nhập không được kết nối với tài khoản. Tìm tài khoản của bạn và đăng nhập.". Hệ thống tiếp tục yêu cầu nhập email/phone number và password.

Hình 3. 7. Kịch bản trường hợp kiểm thử thứ sáu

Test ID	Test Objective	Test Pre-conditions	Test Steps	Expected Result
TC07	Kiểm tra trường hợp đăng nhập vào Facebook thất bại khi bỏ trống email/phone number và nhập không chính xác mật khẩu khi click vào nút "Log in"	Người dùng đã có tài khoản trên hệ thống	1.Vào trang "https://www.facebook.com/" 2.Bỏ trống email/phone number 3.Nhập sai password 4.Click vào nút "Log in"	Thông báo "Địa chỉ email hoặc số điện thoại di động bạn đã nhập không được kết nối với tài khoản. Tìm tài khoản của bạn và đăng nhập.". Hệ thống tiếp tục yêu cầu nhập email/phone number và password.

Hình 3. 8. Kịch bản trường hợp kiểm thử thứ bảy

Test ID	Test Objective	Test Pre-conditions	Test Steps	Expected Result
TC08	Kiểm tra trường hợp đăng nhập vào Facebook thất bại khi bỏ trống email/phone number và mật khẩu khi click vào nút "Log in"	Người dùng đã có tài khoản trên hệ thống	1.Vào trang "https://www.facebook.com/" 2.Bỏ trống email/phone number 3.Bỏ trống password 4.Click vào nút "Log in"	Thông báo "Địa chỉ email hoặc số điện thoại di động bạn đã nhập không được kết nối với tài khoản. Tìm tài khoản của bạn và đăng nhập.". Hệ thống tiếp tục yêu cầu nhập email/phone number và password.

Hình 3. 9. Kịch bản trường hợp kiểm thử thứ tám

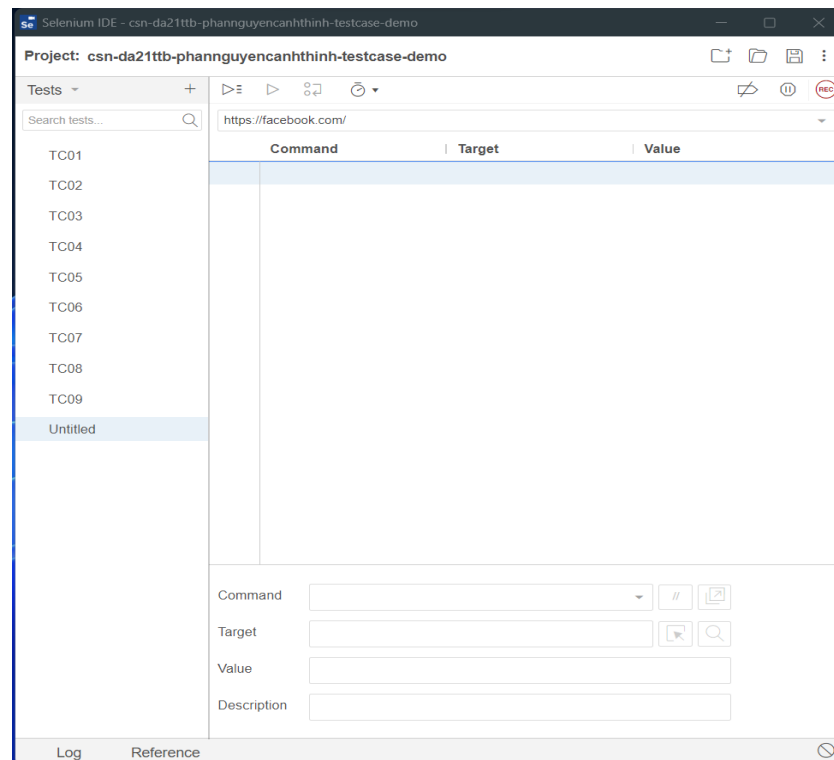
Test ID	Test Objective	Test Pre-conditions	Test Steps	Expected Result
TC09	Kiểm tra trường hợp đăng nhập thành công vào Facebook với việc nhập chính xác cả email/phone number và password khi click vào nút "Log in"	Người dùng đã có tài khoản hợp lệ trên hệ thống	1.Vào trang "https://www.facebook.com/" 2.Nhập đúng email/phone number 3.Nhập đúng password 4.Click vào nút "Log in"	Người dùng được chuyển đến trang chủ

Hình 3. 10. Kịch bản trường hợp kiểm thử thứ chín

- Selenium IDE: công cụ kiểm thử tự động hóa mã nguồn mở dùng để kiểm tra các ứng dụng web, cho phép ghi lại và phát lại những tương tác của người dùng/người kiểm thử với một web bất kỳ nào đó.

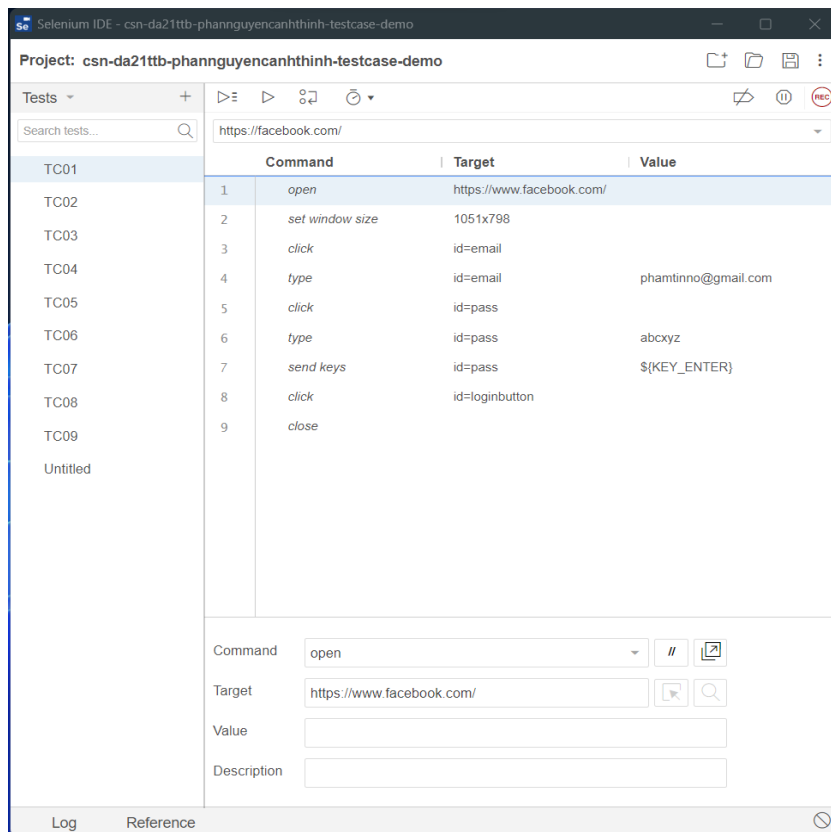
Trong trường hợp này, sử dụng Selenium IDE để kiểm tra chức năng đăng nhập Facebook.

Giao diện chính của Selenium IDE:

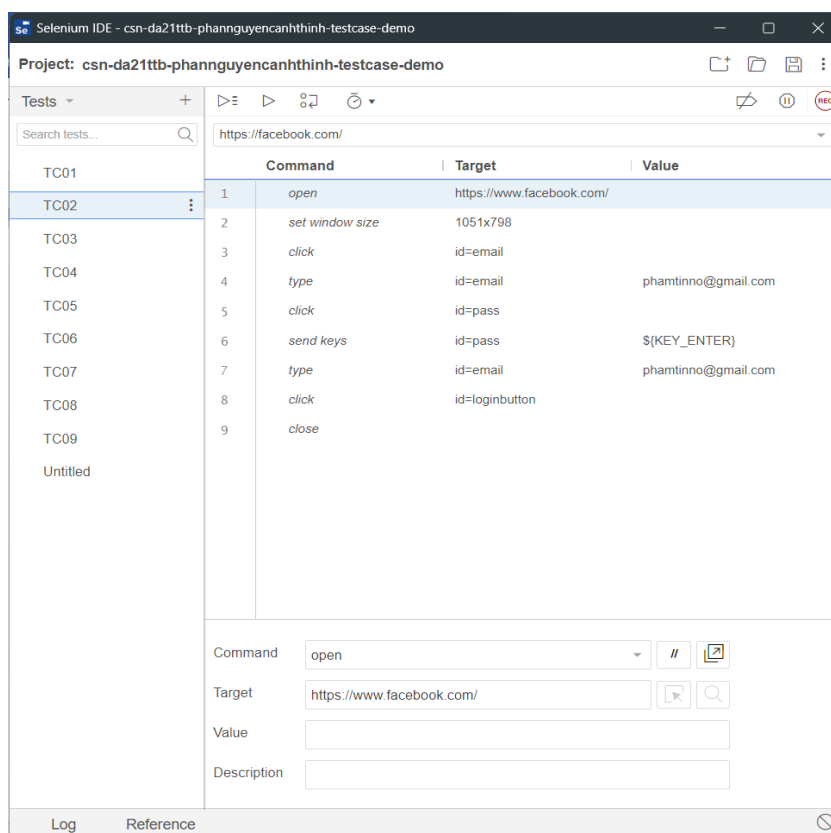


Hình 3. 11. Giao diện của Selenium IDE khi bắt đầu kiểm thử

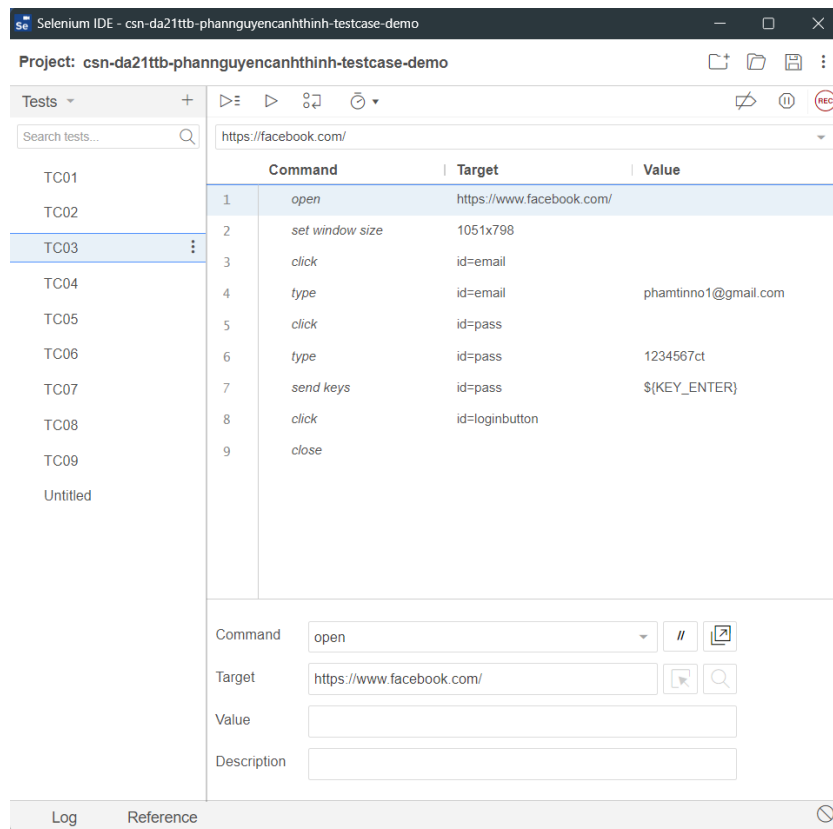
❖ Thiết kế các test case cho từng trường hợp kiểm thử:



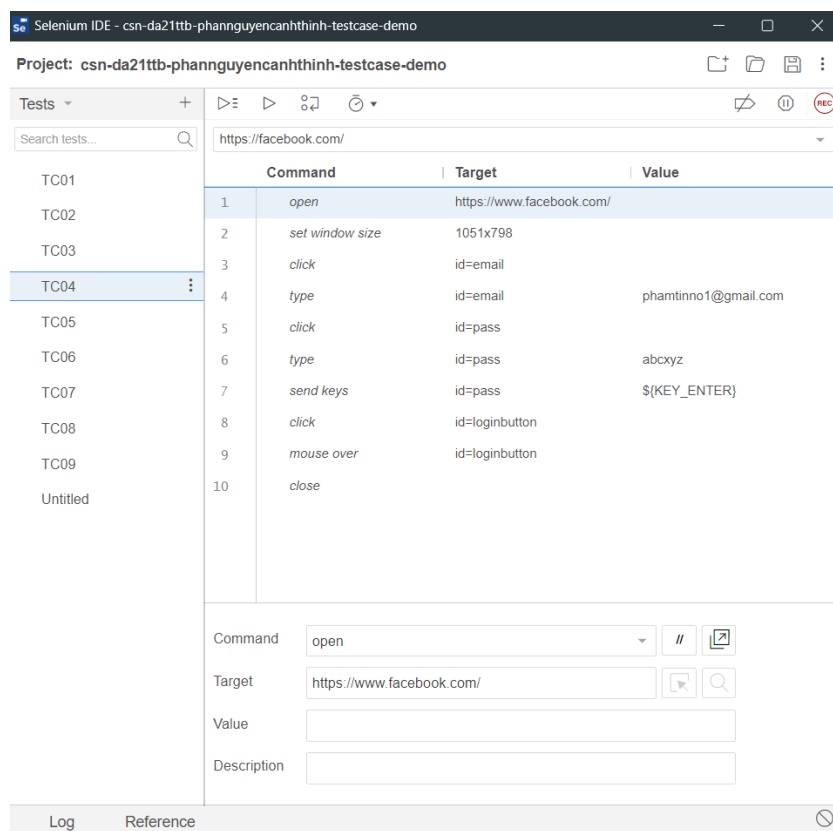
Hình 3. 12. Thực hiện trường hợp kiểm thử thứ nhất



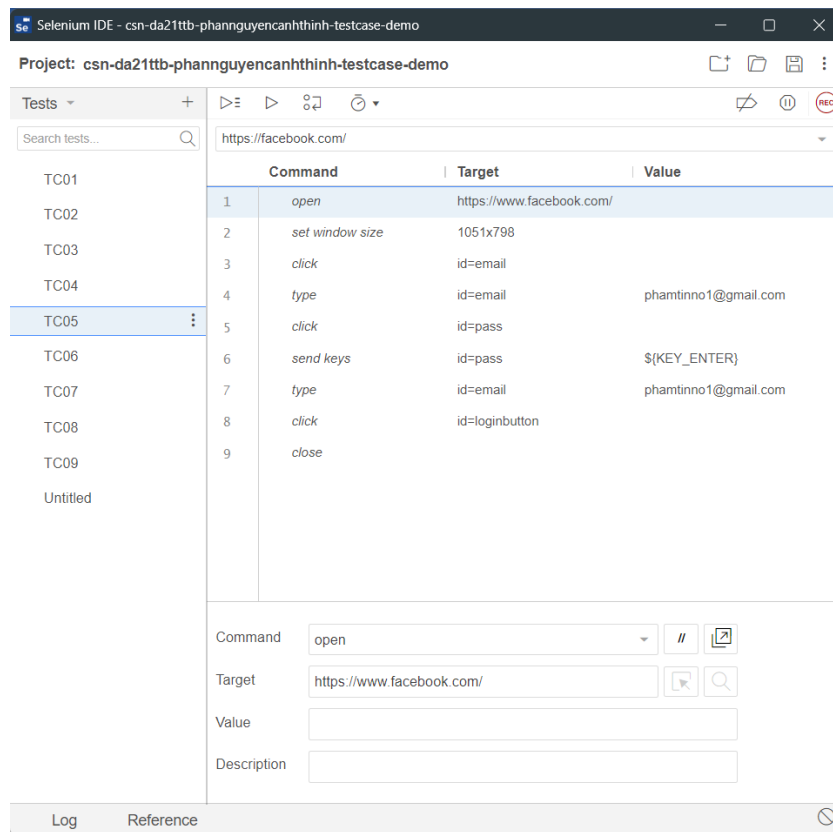
Hình 3. 13. Thực hiện trường hợp kiểm thử thứ hai



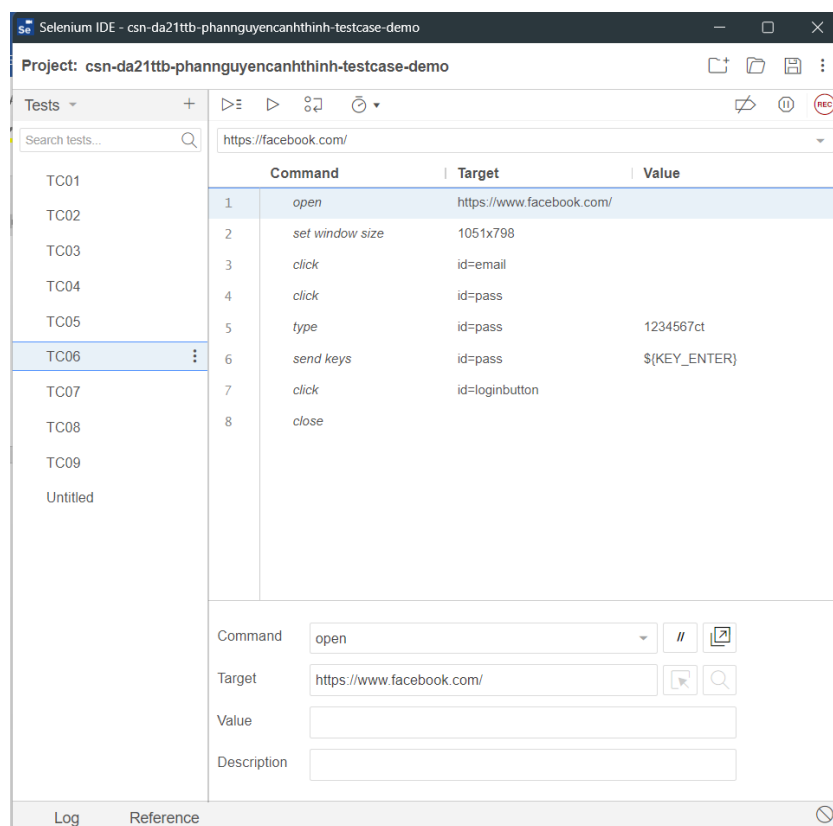
Hình 3. 14. Thực hiện trường hợp kiểm thử thứ ba



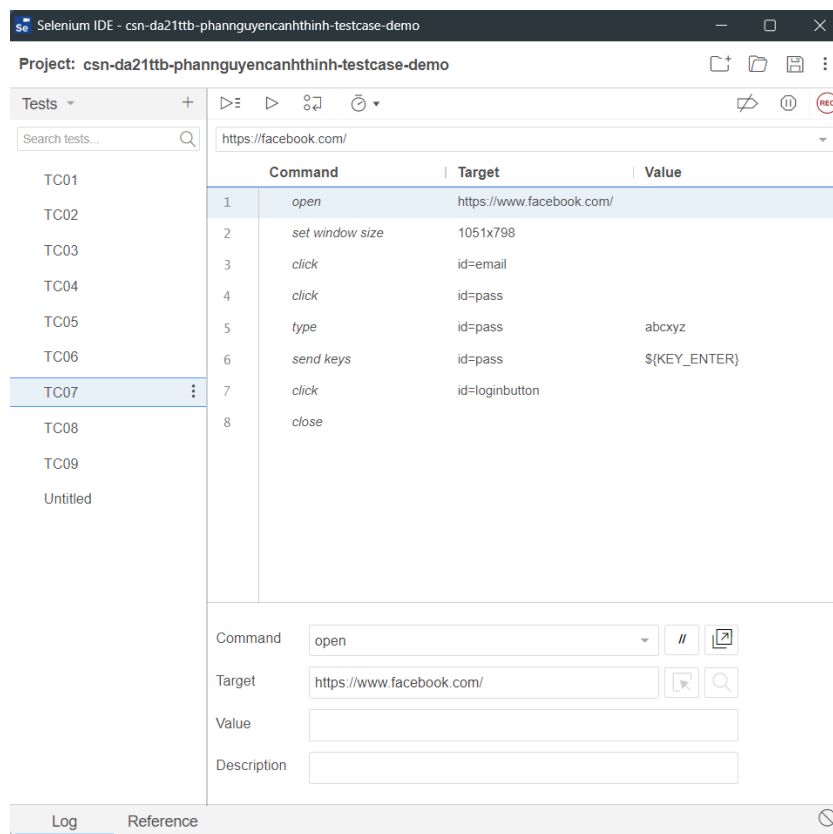
Hình 3. 15. Thực hiện trường hợp kiểm thử thứ tư



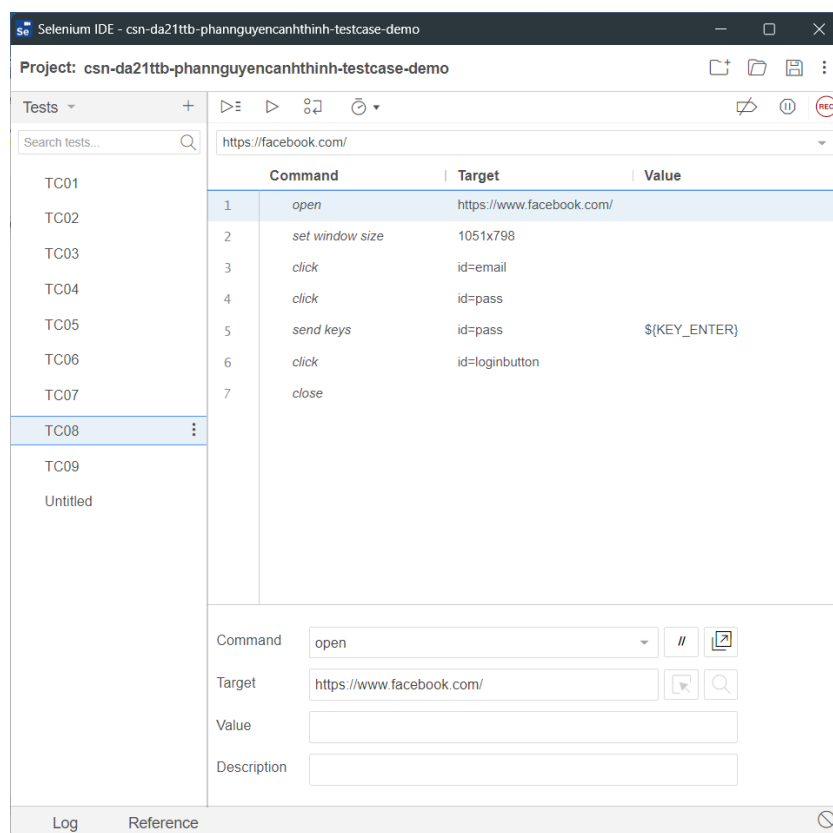
Hình 3. 16. Thực hiện trường hợp kiểm thử thứ năm



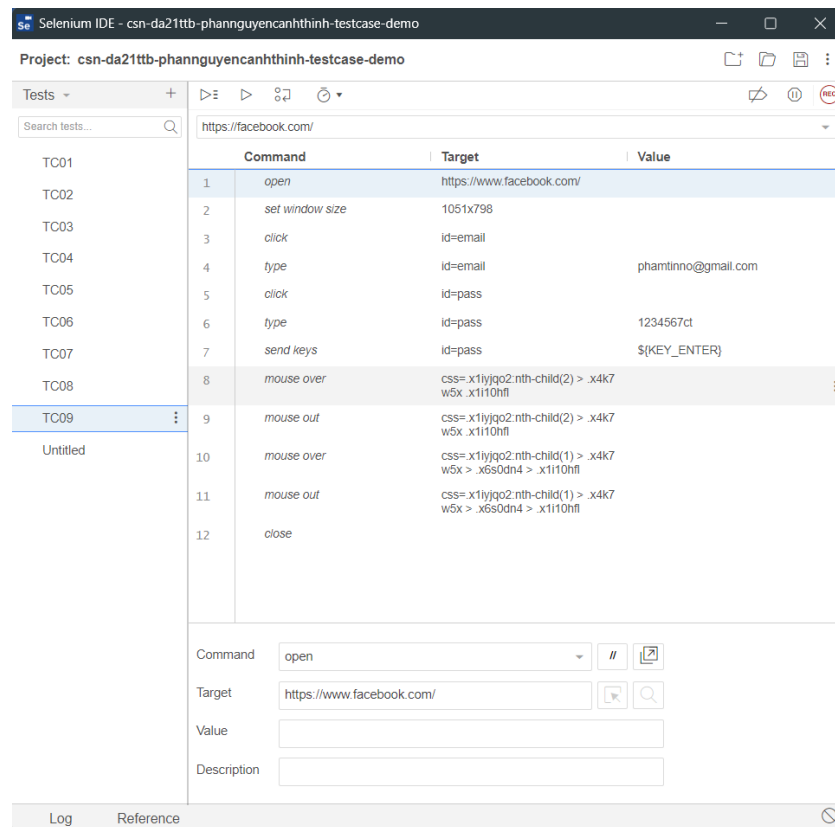
Hình 3. 17. Thực hiện trường hợp kiểm thử thứ sáu



Hình 3. 18. Thực hiện trường hợp kiểm thử thứ bảy



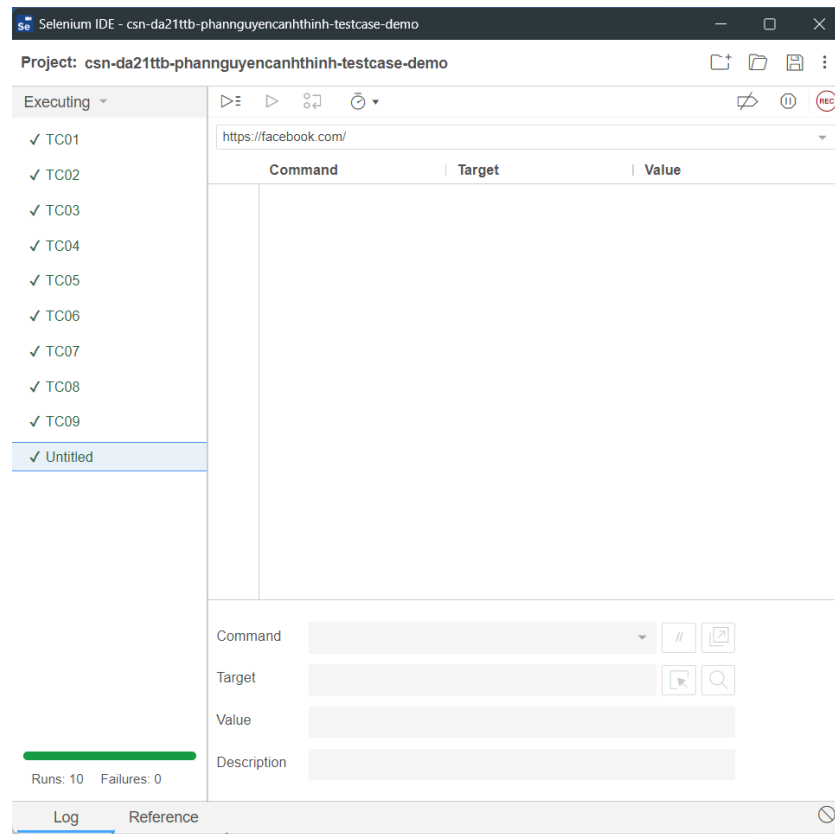
Hình 3. 19. Thực hiện trường hợp kiểm thử thứ tám



Hình 3. 20. Trường hợp kiểm thử thứ chín

CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU

- Nghiên cứu bao gồm việc xây dựng kịch bản và test case kiểm tra chức năng đăng nhập vào Facebook.
- Kết quả là khi thực hiện kiểm thử thì test case của cả 9 trường hợp đều chạy thành công.



Hình 3. 21. Kết quả kiểm thử

- Hoàn thành kiểm thử.

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

➤ Kết luận:

Các nghiên cứu và tìm hiểu về các loại test case trong kiểm thử phần mềm đã được triển khai theo đúng dự định và kế hoạch đề ra. Tuy gặp nhiều khó khăn trong việc tìm kiếm nguồn tài liệu và thông tin cũng như tiếp cận với một số công cụ để thực hiện sản phẩm nhưng cuối cùng đồ án vẫn được bảo đảm tiến độ và hoàn thành.

➤ Hướng phát triển:

Một số hướng phát triển có thể được áp dụng cho việc thực hiện sản phẩm demo tìm hiểu các loại test cases trong kiểm thử phần mềm:

- Sử dụng nhiều phương pháp kiểm thử khác nhau.
- Sử dụng công cụ chuyên về kiểm thử.
- Lựa chọn trường hợp kiểm thử có mức độ khó cao hơn.

DANH MỤC TÀI LIỆU THAM KHẢO

<https://careerbuilder.vn/vi/talentcommunity/test-case-la-gi-huong-dan-cach-viet-test-case-don-gian-chat-luong.35A520D0.html#:~:text=K%E1%BB%B9%20thu%E1%BA%ADt%20Test%20case%20t%C4%A9nh,x%C3%A9t%20t%C3%A0i%20li%E1%BB%87u%20k%E1%BB%B9%20thu%E1%BA%ADt.>

<https://viblo.asia/p/quy-trinh-kiem-thu-phan-mem-software-testing-life-cycle-stlc-Qbq5QLvmlD8>

<https://itnavi.com.vn/blog/cac-ky-thuat-kiem-thu-phan-mem>

PHỤ LỤC