

**BỘ CÔNG THƯƠNG**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

-----

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC**  
**NGÀNH KỸ THUẬT PHẦN MỀM**

**XÂY DỰNG KỊCH BẢN VÀ KIỂM THỬ**  
**ỨNG DỤNG WEBSITE SỬ DỤNG SELENIUM.**

**CBHD: ThS. Phạm Bá Tuấn Chung**

**Sinh viên: Đỗ Thị Huệ**

**Mã sinh viên: 2020606709**

**Hà Nội – Năm 2024**

## LỜI CẢM ƠN

Trên thực tế không có sự thành công nào mà không gắn liền với những sự hỗ trợ, sự giúp đỡ dù ít hay nhiều, dù là trực tiếp hay gián tiếp của người khác. Trong suốt thời gian từ khi bắt đầu học tập, em đã nhận được rất nhiều sự quan tâm, giúp đỡ của Thầy Cô, gia đình và bạn bè.

Trước hết em xin gửi tới các thầy các cô khoa Công Nghệ Thông Tin Trường Đại học Công Nghiệp Hà Nội lời chào trân trọng, lời chúc sức khỏe và lời cảm ơn sâu sắc. Với sự quan tâm, dạy dỗ, chỉ bảo tận tình chu đáo của thầy cô, đến nay em đã có thể hoàn thành đồ án tốt nghiệp, đề tài: ***“Xây dựng kịch bản và kiểm thử ứng dụng website sử dụng Selenium.”***.

Đặc biệt em xin chân thành cảm ơn thầy giáo ThS.Phạm Bá Tuấn Chung, người đã hướng dẫn, dìu dắt em tận tình để em hoàn thành tốt đề tài đồ án.

Sau cùng, em xin kính chúc các thầy cô dồi dào sức khỏe, niềm tin để tiếp tục thực hiện sứ mệnh cao đẹp của mình là truyền đạt kiến thức cho thế hệ mai sau.

Trong quá trình hoàn thành đồ án tốt nghiệp không thể tránh khỏi thiếu sót, kính mong có sự góp ý từ thầy cô.

Em xin chân thành cảm ơn!

**Sinh viên thực hiện**

**Đỗ Thị Huệ**

# MỤC LỤC

DANH MỤC BẢNG BIỂU	5
DANH MỤC HÌNH ẢNH	6
MỞ ĐẦU	1
1.Lý do chọn đề tài	1
2.Đối tượng, mục đích và phạm vi nghiên cứu	1
3.Ý nghĩa khoa học và thực tiễn của đề tài	2
CHƯƠNG 1: TỔNG QUAN VỀ KIỂM THỬ PHẦN MỀM	3
1.1 . Các khái niệm cơ bản về kiểm thử phần mềm	3
1.1.1.Kiểm thử phần mềm	3
1.1.2. Mục tiêu của kiểm thử phần mềm	3
1.1.3. Tầm quan trọng của kiểm thử phần mềm:	3
1.1.4. Ai tham gia vào kiểm thử phần mềm	4
1.1.5. Quy trình kiểm thử phần mềm	4
1.2 . Kỹ thuật kiểm thử phần mềm	6
1.2.1. Kiểm thử hộp trắng (White box testing)	6
1.2.2. Kiểm thử hộp đen (Black box testing)	7
1.3. Các phương pháp kiểm thử hộp đen:	9
1.3.1 Phân vùng tương đương (Equivalence Partitioning)	9
1.3.2 Phân tích giá trị biên (Boundary Value Analysis)	11
1.3.3 Bảng quyết định (Decision Table Exercise)	13
1.3.4 Phương pháp State transition diagrams	15
1.3.5 Phương pháp đoán lỗi (Error Guessing)	16
1.4. Test Case	16
1.5. Các cấp độ kiểm thử phần mềm	19
1.6. Phân loại kiểm thử	22
1.6.1. Khái niệm	22
1.6.2. Điểm khác nhau giữa kiểm thử thủ công và kiểm thử tự động	23
1.6.3. Ưu điểm và nhược điểm	25
1.6.4.Khi nào thì sử dụng kiểm thử thủ công và khi nào thì sử dụng kiểm thử tự động?	26
1.7. Quy trình kiểm thử tự động	26
1.7.1. Một số Test automation framework khác	29
1.7.2. Tại sao nên lựa chọn Automation testing	30
CHƯƠNG 2: TÌM HIỂU VỀ CÔNG CỤ SELENIUM	31

2.1. Giới thiệu công cụ Selenium	31
2.1.1 Lịch sử phát triển	31
2.1.2 Khái niệm	31
2.1.3 Những tính năng của Selenium	31
2.1.4 Selenium bao gồm những công cụ nào?	32
2.1.5 Ưu điểm và nhược điểm của Selenium	33
2.2 Giới thiệu framework Mocha	34
2.3. Cài đặt Selenium Webdriver và môi trường kiểm thử	36
2.3.1. Cài đặt Node.js và npm	36
2.3.2. Cài đặt Selenium webdriver	39
2.3.3. Cài đặt framework Mocha	39
CHƯƠNG 3. THỰC HIỆN KIỂM THỬ WEBSITE	40
3.1. Giới thiệu chương trình	40
3.1.1 Giới thiệu về website	40
3.1.2 Chức năng của hệ thống	40
3.1.3 Một số màn hình giao diện của website	46
3.2. Sự khác nhau giữa kịch bản kiểm thử và kịch bản tự động	48
3.3. Kịch bản kiểm thử thủ công trong website	48
3.3.1. Testcase kiểm thử chức năng Đăng ký	49
3.3.2 Testcase kiểm thử chức năng Đăng nhập	50
3.3.3. Testcase kiểm thử chức năng Đăng bài	51
3.3.4 Test case kiểm thử chức năng Tìm kiếm bạn bè	52
3.3.5 Test case chức năng Nhắn tin	52
3.3.6 Trường hợp kiểm thử chức năng Bình luận	53
3.3.7 Báo cáo kết quả kiểm thử	53
3.4 Kiểm thử tự động các chức năng	54
3.4.1. Kiểm thử tự động chức năng Đăng ký	54
3.4.2. Kiểm thử tự động chức năng Đăng nhập	54
3.4.3 Kiểm thử tự động chức năng Đăng bài	55
3.4.4 Kiểm thử tự động chức năng Nhắn tin	55
3.4.5 Kiểm thử tự động chức năng Tìm kiếm	56
3.4.6 Kiểm thử tự động chức năng Bình luận	56
KẾT LUẬN	57
TÀI LIỆU THAM KHẢO	58

## **DANH MỤC BẢNG BIỂU**

Bảng 1.1 Điểm khác nhau giữa kiểm thử thủ công và kiểm thử tự động .....	23
Bảng 1.2 Ưu điểm và nhược điểm của kiểm thử thủ công và kiểm thử tự động .....	25
Bảng 3.1 Chức năng hệ thống về phía người dùng.....	40
Bảng 3.2 Chức năng hệ thống về phía người quản trị .....	45

## DANH MỤC HÌNH ẢNH

Hình 1.1 Quy trình kiểm thử phần mềm .....	4
Hình 1.2 Kiểm thử hộp trắng (Nguồn: fptshop.com).....	6
Hình 1.3 Kiểm thử hộp đen (Nguồn: viblo.asia).....	8
Hình 1.4 Phương pháp kiểm thử phân vùng tương đương.....	9
Hình 1.5 Xác định các lớp tương đương.....	10
Hình 1.6 Xác định các ca kiểm thử và lựa chọn dữ liệu kiểm thử.....	11
Hình 1.7 Phương pháp kiểm thử Phân tích giá trị biên.....	12
Hình 1.8 Thiết kế các ca kiểm thử dựa vào các điểm biên .....	12
Hình 1.9 Tổng hợp các trường hợp kiểm thử sử dụng 2 phương pháp phân vùng tương đương và phân tích giá trị biên.....	13
Hình 1.10 Bảng quyết định .....	14
Hình 1. 11 Sơ đồ chuyển trạng thái.....	15
Hình 1.12 Các cấp độ kiểm thử phần mềm.....	20
Hình 1.13 Quy trình kiểm thử tự động.....	27
Hình 2. 1 Những công cụ trong Selenium .....	32
Hình 2.2 Cài đặt Nodejs(1) .....	36
Hình 2.3: Cài đặt Nodejs(2) .....	36
Hình 2.4 Cài đặt Nodejs(3) .....	36
Hình 2.5 Cài đặt Nodejs(4) .....	37
Hình 2.6 Cài đặt Nodejs(5) .....	37
Hình 2.7 Cài đặt Nodejs(6) .....	37
Hình 2.8 Cài đặt Nodejs(7) .....	38
Hình 2. 9 Cài đặt Nodejs(8) .....	38
Hình 2.10 Kiểm tra version của nodejs và npm .....	39
Hình 2.11 Folder node_module sau khi cài đặt .....	39
Hình 2.12 Cài đặt Selenium webdriver.....	39
Hình 2.13 Cài đặt framework Mocha.....	39

Hình 3.1 Giao diện màn hình Đăng ký .....	46
Hình 3.2 Giao diện màn hình Đăng nhập.....	46
Hình 3.3 Giao diện màn hình Trang chủ.....	47
Hình 3.4 Giao diện màn hình Trang cá nhân .....	47
Hình 3.5 Giao diện màn hình Trang cá nhân bạn bè.....	48
Hình 3.6 Trường hợp kiểm thử chức năng Đăng ký .....	49
Hình 3.7 Trường hợp kiểm thử chức năng Đăng nhập .....	50
Hình 3.8 Trường hợp kiểm thử chức năng Đăng bài .....	51
Hình 3.9 Trường hợp kiểm thử chức năng Tìm kiếm bạn bè .....	52
Hình 3.10 Trường hợp kiểm thử chức năng Nhắn tin.....	52
Hình 3.12 Trường hợp kiểm thử chức năng Bình luận .....	53
Hình 3.13 Kết quả kiểm thử tự động chức năng Đăng ký .....	54
Hình 3.14 Kết quả kiểm thử tự động chức năng Đăng nhập.....	54
Hình 3.15 Kết quả kiểm thử tự động chức năng Đăng bài .....	55
Hình 3.16 Kết quả kiểm thử tự động chức năng Nhắn tin .....	55
Hình 3.17 Kết quả kiểm thử tự động chức năng Tìm kiếm bạn bè.....	56
Hình 3.18 Kết quả kiểm thử tự động chức năng Bình luận .....	56

## MỞ ĐẦU

### 1.Lý do chọn đề tài

Trong những năm gần đây với sự phát triển rất mạnh mẽ của công nghệ thông tin, ngành công nghệ phần mềm đang chiếm một vị trí hết sức quan trọng trong xu hướng phát triển kinh tế công nghiệp hóa, hiện đại hóa của nước ta. Cùng với sự phát triển ấy các chương trình phần mềm ra đời ngày càng nhiều, đòi hỏi các nhà sản xuất phần mềm phải có một phương pháp để nâng cao chất lượng sản phẩm cũng như tối ưu hiệu suất làm việc để có thể cạnh tranh. Vì vậy kiểm thử phần mềm đang ngày càng đóng vai trò quan trọng trong ngành công nghiệp phát triển phần mềm không chỉ ở Việt Nam và trên thế giới. Với mong muốn có cái nhìn xác thực, rõ ràng hơn về quy trình kiểm thử phần mềm, đảm bảo chất lượng phần mềm và tiếp cận với các công cụ hỗ trợ kiểm thử, giải quyết phần nào vấn đề về thời gian, kinh phí trong việc tìm kiếm lỗi, quản lý lỗi khi tiến hành kiểm thử; đồng thời rèn kỹ năng làm việc, tạo tiền đề định hướng cho tương lai sau khi ra trường. Với những lý do như trên nên em đã lựa chọn đề tài “***Xây dựng kịch bản và kiểm thử ứng dụng website sử dụng Selenium***”.

### 2.Đối tượng, mục đích và phạm vi nghiên cứu

#### a) Đối tượng nghiên cứu

- Công nghệ: Ngôn ngữ Javascript, Selenium Webdriver, Mocha
- Phần mềm lập trình: Visual Studio Code

#### b) Mục đích nghiên cứu

- Xác định và làm rõ các vấn đề liên quan đến đề tài.
- Tìm hiểu khái niệm, thành phần, tính năng của công cụ Selenium
- Tìm hiểu framework để hỗ trợ trình bày và quản lý test cases
- Xây dựng kịch bản và kiểm thử ứng dụng website sử dụng các phương pháp kiểm thử hộp đen.



- Cài đặt công cụ và môi trường kiểm thử trên thiết bị.
- Sử dụng framework Mocha để trình bày và quản lý test cases.
- Sử dụng ngôn ngữ lập trình và công cụ Selenium để xây dựng kịch bản và kiểm thử tự động ứng dụng website.

#### c) Phạm vi nghiên cứu

- Tìm hiểu những yếu tố và điều kiện cần thiết phải có để có thể hỗ trợ xây dựng kịch bản và kiểm thử một cách tối ưu nhất.
- Thực hiện kiểm thử các chức năng chính cần thiết cho hệ thống phần mềm.
- Lựa chọn các phương pháp và công nghệ phù hợp để xây dựng kịch bản và kiểm thử ứng dụng website.

### 3. Ý nghĩa khoa học và thực tiễn của đề tài

Đề tài em muốn thực hiện là ***“Xây dựng kịch bản và kiểm thử ứng dụng website sử dụng Selenium”***. Với đề tài này, chương trình kiểm thử tự động sẽ cho phép người kiểm thử thực hiện bộ test suite một cách nhanh chóng, chính xác và tự động, giúp người kiểm thử có kết quả kiểm thử 1 chức năng chỉ sau vài giây.

*Một số kết quả thực nghiệm cần đạt được:*

- Chương trình thực hiện đúng, đủ và cho ra kết quả chính xác.
- Chương trình được trình bày gọn gàng và dễ hiểu
- Chương trình có tính tái sử dụng cao

# CHƯƠNG 1: TỔNG QUAN VỀ KIỂM THỬ PHẦN MỀM

## 1.1. Các khái niệm cơ bản về kiểm thử phần mềm

### 1.1.1. Kiểm thử phần mềm

Kiểm thử phần mềm có nhiều định nghĩa khác nhau đề xuất bởi nhiều tổ chức hay cá nhân khác nhau nhưng tổng quan thì:

Kiểm thử phần mềm là quá trình thực thi một chương trình với mục đích tìm ra lỗi và đảm bảo sản phẩm phần mềm đáp ứng chính xác, đầy đủ đúng theo yêu cầu của khách hàng, yêu cầu của sản phẩm đã đặt ra. Kiểm thử phần mềm cũng cung cấp mục tiêu, cái nhìn độc lập về phần mềm, điều này cho phép việc đánh giá và hiểu rõ các rủi ro khi thực thi phần mềm.

### 1.1.2. Mục tiêu của kiểm thử phần mềm

#### Mục tiêu trực tiếp:

- Xác định và phát hiện nhiều lỗi nhất có thể trong phần mềm được kiểm thử. Chứng minh rằng sản phẩm phần mềm phù hợp với đặc tả yêu cầu của nó
- Sau khi sửa chữa các lỗi đã xác định và kiểm tra lại, làm cho phần mềm đã được kiểm thử đến một mức độ chấp nhận được về chất lượng.
- Tạo các testcase chất lượng cao, thực hiện kiểm thử hiệu quả và tạo ra các báo cáo vấn đề đúng và hữu dụng.

#### Mục tiêu gián tiếp:

- Biên dịch một tài liệu về các lỗi phần mềm thường gặp nhằm mục đích ngăn ngừa và sửa chữa lỗi.

### 1.1.3. Tầm quan trọng của kiểm thử phần mềm:

#### Ý nghĩa:

- Kiểm thử phần mềm không chỉ cần tìm lỗi phần mềm, mà còn là quá trình kiểm tra và xác minh một phần mềm đã đáp ứng được yêu cầu và mong đợi của khách hàng. Chi phí kiểm thử chiếm: 40% tổng công sức phát triển,  $\geq$

30% tổng thời gian phát triển. Với các phần mềm quan trọng chi phí có thể gấp từ 3 đến 5 lần tổng các chi phí khác cộng lại.

- Giảm chi phí điều chỉnh và thời gian phát triển.
- Tăng độ tin cậy của phần mềm

#### 1.1.4. Ai tham gia vào kiểm thử phần mềm

Tất cả các thành viên của dự án đều tham gia vào quá trình kiểm thử phần mềm ở góc độ và vai trò của họ trong dự án phần mềm:

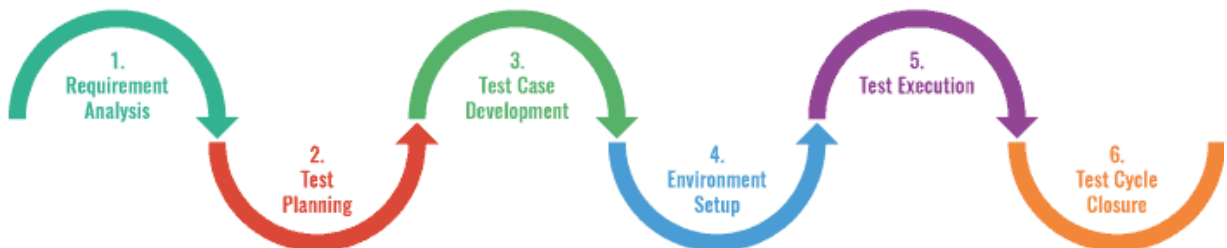
Software Tester:

- Test leader: Lên kế hoạch kiểm thử, tạo báo cáo, phân công tester;
- Test designer: Thiết kế Testcase, sắp xếp thứ tự kiểm thử;
- Test worker: Thực hiện kiểm thử, làm báo cáo kết quả kiểm thử.
- Software Developer: Tham gia vào quá trình kiểm thử đơn vị;
- Project Lead/Manager: Lên kế hoạch, thời gian kiểm thử để đảm bảo duy trì đúng tiến độ
- End User: Những người kiểm thử cuối cùng trong vai trò người sử dụng.

#### 1.1.5. Quy trình kiểm thử phần mềm

Quy trình kiểm thử phần mềm xác định các giai đoạn/ pha trong kiểm thử phần mềm. Tuy nhiên, không có STLC tiêu chuẩn cố định nào trên thế giới, nhưng về cơ bản quy trình kiểm thử bao gồm những giai đoạn sau:

### Software Testing Life Cycle (STLC)



Hình 1.1 Quy trình kiểm thử phần mềm

## **Requirement analysis – Phân tích yêu cầu:**

QA team (Quality Assurance team) có nhiệm vụ phân tích và xác định những yêu cầu của khách hàng, trong đó có yêu cầu về kiểm thử chức năng/phi chức năng của phần mềm. Trong quá trình phân tích, QA team có thể đặt ra câu hỏi để hiểu chính xác hơn về yêu cầu của sản phẩm, đồng thời hỗ trợ đưa ra giải pháp thích hợp cho khách hàng.

## **Test planning – Lập kế hoạch kiểm thử**

Dựa vào tài liệu nhận được trong giai đoạn đầu, Test Lead hoặc Test Manager sẽ lên kế hoạch kiểm thử phần mềm cho QA team để xác định một số yếu tố

Phạm vi dự án: Thời gian thực hiện dự án bao lâu? Trong từng khoảng thời gian sẽ có những công việc gì?

## **Test case development – Thiết kế kịch bản cho quy trình kiểm thử**

Trong giai đoạn này, các Tester sẽ đọc hiểu tất cả các tài liệu, từ đó xác định những việc cần làm, chức năng nào cần test hoặc không. Sau đó, dựa vào kế hoạch và kỹ thuật thiết kế kịch bản kiểm thử, Tester sẽ bắt đầu viết test case. Yêu cầu của test case: Thể hiện tất cả các trường hợp kiểm thử có thể phát sinh để đáp ứng yêu cầu sản phẩm. Ngoài test case, Tester cũng cần chuẩn bị các dữ liệu cần thiết khác như test data, test script, test design, test automation script.

## **Test environment set up – Thiết lập môi trường kiểm thử**

Đây là một trong những giai đoạn đóng vai trò rất quan trọng trong Software Testing Life Cycle (vòng đời phát triển phần mềm). Dựa trên yêu cầu khách hàng và đặc thù của sản phẩm, môi trường kiểm thử sẽ được xác định. Tester cần chuẩn bị smoke test case để kiểm tra môi trường cài đặt đã đáp ứng yêu cầu và sẵn sàng cho giai đoạn kiểm thử tiếp theo hay chưa.

## **Test execution – Thực hiện kiểm thử**

Theo test case đã thiết kế và môi trường kiểm thử đã hoàn tất cài đặt, Tester sẽ báo cáo bug lên tool quản lý lỗi và theo dõi đến khi fix bug thành công. Tiếp đó,

Tester thực hiện retest để verify các fix bug và regression test trong trường hợp có sự thay đổi. Sau khi hoàn tất giai đoạn này, các chuyên viên kiểm thử cần có được test results (kết quả kiểm thử) và defect reports (danh sách các lỗi tìm được).

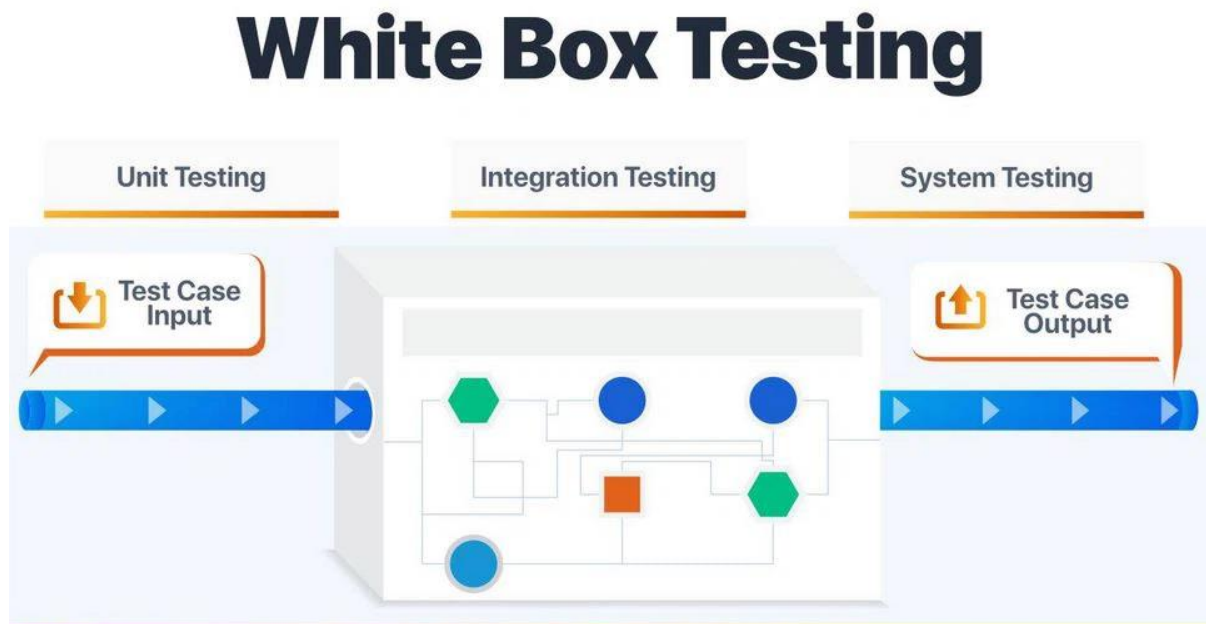
### Test cycle closure – Đóng chu trình kiểm thử

Để đóng chu trình kiểm thử, QA team cần có được những tài liệu đã được tổng hợp và hoàn thiện từ những giai đoạn trước: tài liệu phân tích đặc tả yêu cầu, test plan, defect reports, test results... Tiếp đó, QA team sẽ tổng kết, báo cáo về quá trình kiểm thử, có bao nhiêu bug đã được fix, bug có nghiêm trọng hay không, chức năng nào còn lỗi, chức năng nào đã hoàn thành...

## 1.2. Kỹ thuật kiểm thử phần mềm

### 1.2.1. Kiểm thử hộp trắng (White box testing)

Kiểm thử hộp trắng dựa vào thuật toán, cấu trúc code bên trong của chương trình với mục đích đảm bảo rằng tất cả các câu lệnh sẽ được thực hiện ít nhất một lần.



Hình 1.2 Kiểm thử hộp trắng (Nguồn: fptshop.com)

Kiểm thử hộp trắng là kỹ thuật kiểm thử dựa trên đặc tả bên trong của chương trình, dựa vào mã nguồn, cấu trúc chương trình.

Kiểm thử hộp trắng thường phát hiện các lỗi lập trình, để thực hiện chi phí không cao. Với các module quan trọng, thực thi việc tính toán chính của hệ thống, phương pháp này là cần thiết.

**Ưu điểm:**

- Tối ưu hóa mã bằng cách tìm các lỗi ẩn.
- Các trường hợp kiểm thử hộp trắng có thể dễ dàng được tự động hóa.
- Thử nghiệm kỹ lưỡng hơn vì tất cả các đường dẫn mã thường được bao phủ
- Thử nghiệm có thể bắt đầu sớm trong SDLC ngay cả khi GUI không khả dụng.

**Nhược điểm:**

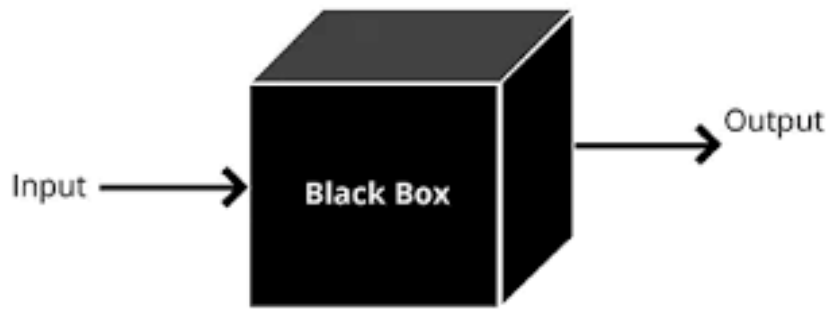
- Kiểm thử hộp trắng có thể khá phức tạp và tốn kém.
- Các nhà phát triển thường thực hiện các ca kiểm thử hộp trắng. Thử nghiệm hộp trắng của nhà phát triển không chi tiết và có thể dẫn đến lỗi sản xuất.
- Kiểm thử hộp trắng yêu cầu nguồn lực chuyên nghiệp với sự hiểu biết chi tiết về lập trình và triển khai.
- Kiểm thử hộp trắng tốn nhiều thời gian, các ứng dụng lập trình lớn hơn cần thời gian để kiểm tra đầy đủ.

**Các kỹ thuật kiểm thử dùng trong kiểm thử hộp trắng:**

- Bao phủ câu lệnh - Statement Coverage (SC).
- Bao phủ rẽ nhánh - Bao phủ quyết định Branch Coverage.
- Bao phủ đường dẫn - Path Throught Coverage

**1.2.2. Kiểm thử hộp đen (Black box testing)**

Kiểm thử hộp đen là 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 tra hộp đen chỉ là kiểm tra xem phần mềm hoạt động có đúng như dự kiến trong tài liệu yêu cầu và liệu nó có đáp ứng được sự mong đợi của người dùng hay không.



Hình 1.3 Kiểm thử hộp đen (Nguồn: viblo.asia)

Nó còn được gọi là kiểm thử hướng dữ liệu hay là kiểm thử hướng in/out. Người kiểm thử nên xây dựng các nhóm giá trị đầu vào mà sẽ thực thi đầy đủ tất cả các yêu cầu chức năng của chương trình.

Cách tiếp cận của các tester đối với hệ thống là không dùng bất kỳ một kiến thức về cấu trúc lập trình bên trong hệ thống, xem hệ thống là một cấu trúc hoàn chỉnh, không thể can thiệp vào bên trong.

#### **Ưu điểm:**

- Thông tin duy nhất là đặc tả về yêu cầu chức năng.
- Người kiểm thử không cần thiết phải có kiến thức về việc mã hóa, cấu trúc bên trong của thành phần phần mềm, cũng như không yêu cầu phải biết lập trình phần mềm.
- Việc kiểm thử được tiến hành dựa vào việc kiểm thử chức năng làm được gì, có phù hợp với yêu cầu của người dùng hay không. Các tester nhập số liệu vào phần mềm và chỉ cần xem kết quả của phần mềm và các mục tiêu kiểm tra.
- Mức test này thường yêu cầu các tester phải viết test case đầy đủ trước khi test; khi test, đơn giản chỉ cần thực hiện theo các bước mô tả trong test case thao tác và nhập data vào, sau đó xem kết quả trả về hoặc hành vi của phần mềm, rồi so sánh với kết quả được viết trong test case.

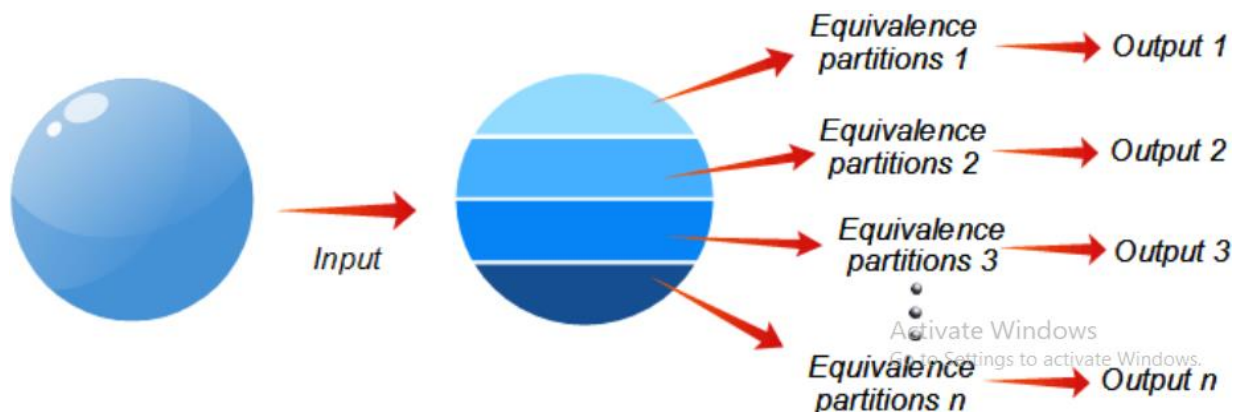
### Nhược điểm:

- Dữ liệu đầu vào yêu cầu một khối lượng mẫu khá lớn.
- Nhiều dự án không có tham số rõ ràng thì việc thiết kế test case rất khó, và do đó khó viết kịch bản kiểm thử do cần xác định tất cả các yếu tố đầu vào, và thiếu cả thời gian cho việc tập hợp này.
- Khả năng để bản thân kỹ sư lạc lối trong khi kiểm thử là khá cao.
- Chỉ có một số nhỏ các đầu vào có thể được kiểm tra và nhiều đường dẫn chương trình sẽ được để lại chưa được kiểm tra.

### 1.3. Các phương pháp kiểm thử hộp đen:

#### 1.3.1 Phân vùng tương đương (Equivalence Partitioning)

Phân vùng tương đương là phương pháp chia các điều kiện đầu vào thành những vùng tương đương nhau. Tất cả các giá trị trong một vùng tương đương sẽ cho một kết quả đầu ra giống nhau. Vì vậy chúng ta có thể kiểm tra một giá trị đại diện trong vùng tương đương.



Hình 1.4 Phương pháp kiểm thử phân vùng tương đương

Thiết kế ca kiểm thử cho phân lớp tương đương dựa trên sự đánh giá về các lớp tương đương với một điều kiện vào. Lớp tương đương biểu thị cho tập các trạng thái hợp lệ hay không hợp lệ đối với điều kiện vào.

Thiết kế các ca kiểm thử bằng kỹ thuật phân vùng tương đương tiến hành theo hai bước :



### Bước 1: Xác định các lớp tương đương

Các lớp tương đương được xác định bằng cách lấy mỗi trạng thái đầu vào (thường là một câu hay một cụm từ trong đặc tả) và phân chia nó thành 2 hay nhiều nhóm.

Mẫu liệt kê các lớp tương đương:

- Điều kiện đầu vào là một giá trị đặc biệt mảng số, mảng chuỗi hay tập hợp điều kiện đúng sai.
- Các lớp tương đương hợp lệ là mô tả các đầu vào hợp lệ của chương trình.
- Các lớp tương đương không hợp lệ là mô tả các trạng thái khác nhau của chương trình như: sai, thiếu, không đúng,...

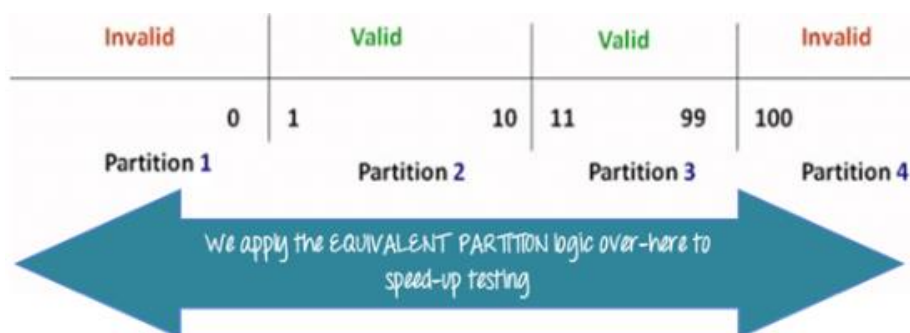
### Bước 2: Xác định các ca kiểm thử và lựa chọn dữ liệu kiểm thử

Với các lớp tương đương được xác định ở bước trước, bước thứ hai là sử dụng các lớp tương đương đó để xác định các ca kiểm thử sau:

- Gán một số duy nhất cho mỗi lớp tương đương.
- Hợp nhất các lớp tương đương hợp lệ được bao phủ bởi các ca kiểm thử.
- Hợp nhất các lớp tương đương không hợp lệ được bao phủ bởi các ca kiểm thử.

*Ví dụ: Trường Quantity chỉ cho phép nhập số nguyên từ 1 đến 99. Trong đó từ 1-10 dành cho khách hàng mua lẻ (không được chiết khấu), từ 11 đến 99 dành cho khách hàng mua sỉ (có chiết khấu).*

*B1: Xác định các lớp tương đương như Hình 1.5*



Hình 1.5 Xác định các lớp tương đương

*B2: Xác định các ca kiểm thử và lựa chọn dữ liệu kiểm thử như Hình 1.6*

TC	Giá trị	Kết quả mong đợi
1	Nhập giá trị < 1 (Nhập -1)	Hiển thị thông báo “Số lượng nhập vào phải thuộc khoảng 1 đến 99”
2	Nhập giá trị trong khoảng từ 1 đến 10 (nhập 5)	Hệ thống tính toán lại Thành tiền = Số lượng * Đơn giá
3	Nhập giá trị trong khoảng từ 11 đến 99 (nhập 50)	Hệ thống tính toán lại Thành tiền = Số lượng * Đơn giá - Số tiền giảm giá
4	Nhập giá trị $\geq 100$ (nhập 100)	Hiển thị thông báo “Số lượng nhập vào phải thuộc khoảng 1 đến 99”

*Hình 1.6 Xác định các ca kiểm thử và lựa chọn dữ liệu kiểm thử*

#### **Ưu điểm :**

- Toàn bộ yêu cầu của hệ thống được kiểm thử chính xác.
- Do mỗi vùng tương đương chỉ cần test trên các phần tử đại diện nên số lượng testcase được giảm đi khá nhiều nhờ đó mà thời gian thực hiện test cũng giảm đáng kể.

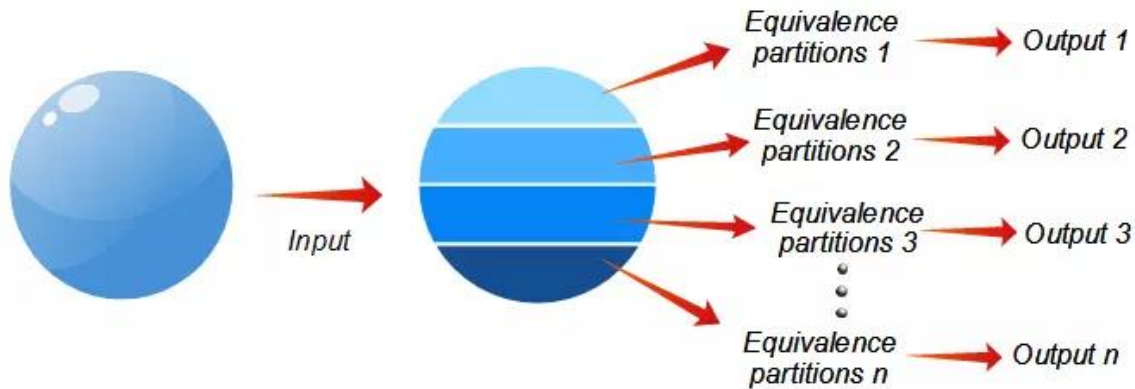
#### **Nhược điểm:**

- Không phải với bất kỳ bài toán nào đều có thể áp dụng kỹ thuật này.
- Có thể bị sót các trường hợp nếu không chọn dữ liệu kiểm thử hợp lý.

### **1.3.2 Phân tích giá trị biên (Boundary Value Analysis)**

Hầu hết các lỗi được tìm ra khi kiểm tra ở các giá trị biên. Vì vậy phương pháp này tập trung vào việc kiểm thử các giá trị biên này.

Phân tích giá trị biên là trường hợp đặc biệt của phân vùng tương đương, dựa trên những phân vùng tương đương tester sẽ xác định giá trị biên giữa những phân vùng này và lựa chọn test case phù hợp.



Hình 1.7 Phương pháp kiểm thử Phân tích giá trị biên

Thiết kế testcase theo Phân tích giá trị biên tiến hành theo 3 bước:

Bước 1: Xác định các điểm biên  $\{a; b\}$

Bước 2: Thiết kế các ca kiểm thử dựa vào các điểm biên như Hình 1.8

Test case	Giá trị	Kết quả mong đợi
1	a-1	Không hợp lệ
2	a	Hợp lệ
3	b	Hợp lệ
4	b+1	Không hợp lệ

Hình 1.8 Thiết kế các ca kiểm thử dựa vào các điểm biên

### Ưu điểm:

- Bù được những nhược điểm của phương pháp phân vùng tương đương
- Bắt được lỗi tiềm ẩn tại các ngõ ngách và tập hợp tại biên
- Tiết kiệm thời gian thiết kế testcase và thực hiện test.

### Nhược điểm:

- Không phải với bất kỳ bài toán nào cũng có thể áp dụng kỹ thuật này. Phương pháp này chỉ hiệu quả trong trường hợp các đối số đầu vào (input variables) độc lập với nhau, và mỗi đối số đều có một miền giá trị hữu hạn.

Vì vậy, phương pháp phân vùng tương đương và phân tích giá trị biên luôn

được sử dụng đồng thời.

Các trường hợp kiểm thử được tổng hợp lại khi sử dụng hai phương pháp phân vùng tương đương và phân tích giá trị biên trong ví dụ trên như Hình 1.9

Test case	Giá trị	Kết quả mong đợi
1	-1	Không hợp lệ
2	0	Không hợp lệ
3	1	Giá mua lẻ
4	5	Giá mua lẻ
5	10	Giá mua lẻ
6	11	Giá mua sỉ
7	15	Không hợp lệ
8	99	Không hợp lệ
9	100	Không hợp lệ
10	100.5	Không hợp lệ

*Hình 1.9 Tổng hợp các trường hợp kiểm thử sử dụng 2 phương pháp phân vùng tương đương và phân tích giá trị biên*

### 1.3.3 Bảng quyết định (Decision Table Exercise)

Trong kỹ thuật viết kịch bản kiểm thử, đối với trường dữ liệu đơn như textbox, chúng ta thường sử dụng các phương pháp như phân lớp tương đương, hay phân tích giá trị biên. Đối với kiểm thử hành vi của hệ thống với nhiều trường dữ liệu, bảng quyết định sẽ giúp chúng ta phân loại và định hình được kịch bản kiểm thử một cách chính xác và rõ ràng hơn.

Bảng quyết định là một kỹ thuật test được sử dụng để kiểm tra các hành vi hệ thống với các cách kết hợp đầu vào khác nhau. Đây là một cách tiếp cận có hệ thống, kết quả của các kết hợp và hành vi hệ thống tương ứng của chúng sẽ được ghi lại dưới dạng bảng. Bảng quyết định thường áp dụng để kiểm thử những chức năng có logic phức tạp và có sự kết hợp của nhiều điều kiện.

Thiết kế bảng quyết định theo 4 bước:

Bước 1: Liệt kê toàn bộ các điều kiện đầu vào

Bước 2: Tính toán số lượng các kết hợp (Rules):

**Rules = 2 lũy thừa của số điều kiện**

Bước 3: Đặt toàn bộ các kết hợp vào bảng

Bước 4: Giảm số lượng các kết hợp và quyết định các trường hợp kiểm thử

		Rules								
		R1	R2	R3	R4	R5	R6	R7	R8	
Conditions	C1	T	T	T	T	F	F	F	F	Values of Conditions
	C2	T	T	F	F	T	T	F	F	
	C3	T	F	T	F	T	F	T	F	
Actions	a1	x			x	x		x		Actions taken
	a2	x						x		
	a3		x				x			
	a4			x	x		x	x		

Hình 1.10 Bảng quyết định

Bảng quyết định chia làm 2 phần:

- Điều kiện: diễn tả mọi điều kiện có thể có
- Hành động: là những hành động khác nhau có thể xảy ra tùy thuộc vào điều kiện.

Số cột của bảng là số lượng các kết hợp điều kiện (Rules).

**Ưu điểm:**

- Bù được những nhược điểm của phương pháp phân vùng tương đương và phân tích giá trị biên.
- Mô tả các quy tắc nghiệp vụ phức tạp dưới dạng dễ đọc và dễ kiểm soát.
- Xác định số trường hợp kiểm thử tối thiểu với độ bao phủ tối đa.
- Tiết kiệm thời gian thiết kế các trường hợp kiểm thử và thực hiện kiểm thử.

**Nhược điểm:**

- Không phải với bất kỳ bài toán nào cũng có thể áp dụng kỹ thuật này. Phương pháp này chỉ hiệu quả trong trường hợp các đối số đầu vào chỉ có giá trị giới hạn là True/False hoặc Yes/No.
- Nếu có quá nhiều kết hợp, chúng ta dễ bỏ sót trường hợp kiểm thử.

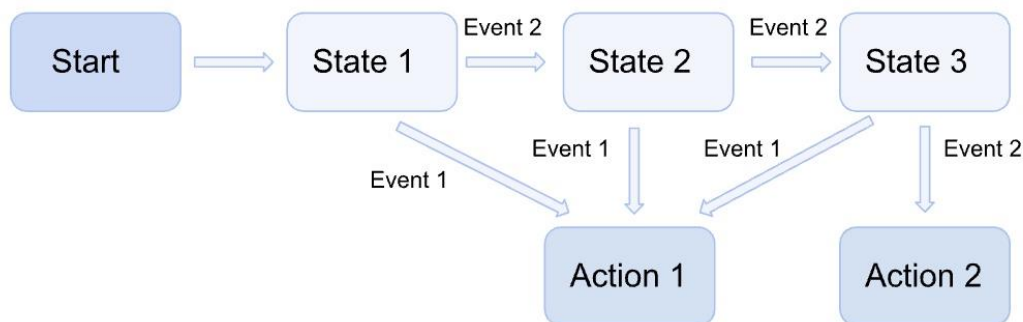
**1.3.4 Phương pháp State transition diagrams**

State transition diagrams là một loại sơ đồ được sử dụng để biểu thị các trạng thái chuyển tiếp (thay đổi) khác nhau của hệ thống. Nó thường được sử dụng để thể hiện bằng đồ họa tất cả các trạng thái chuyển tiếp có thể có mà một hệ thống có thể có và mô hình hóa các hệ thống đó.

Kỹ thuật này giúp bạn thiết kế các trường hợp kiểm thử dựa vào mô hình hoặc lược đồ mô tả các trạng thái và chuyển đổi của chúng. State transition diagrams được áp dụng cho những tính năng của hệ thống được biểu diễn dưới các trạng thái chuyển đổi thành một hệ thống khác.

Mô hình một sơ đồ chuyển trạng thái gồm 4 phần cơ bản:

- Các trạng thái (hữu hạn)
- Các sự dịch chuyển trạng thái
- Các sự kiện kích hoạt sự dịch chuyển trạng thái
- Các hành động là kết quả của sự dịch chuyển



*Hình 1. 11 Sơ đồ chuyển trạng thái*

**Ưu điểm:**

- Sơ đồ cung cấp hình ảnh tổng quan hành vi của hệ thống, giúp người kiểm thử bao quát và hiểu hành vi của hệ thống một cách hiệu quả.
- Giúp người kiểm thử phủ được tất cả các trường hợp theo các điều kiện.

**Nhược điểm:**

- Không áp dụng cho những bài toán có sự di chuyển lặp lại liên tục, không có điểm kết thúc.

**1.3.5 Phương pháp đoán lỗi (Error Guessing)**

Phương pháp này không có quy trình cụ thể vì có tính trực giác cao và không thể dự đoán trước.

Phương pháp này phù hợp với những tester có kinh nghiệm. Họ được đưa cho một chương trình, họ phỏng đoán dựa vào trực giác, dựa vào kinh nghiệm, dữ liệu lịch sử về các lỗi đã từng xảy ra với chương trình trước đó...và sau đó viết các ca kiểm thử để đưa ra các lỗi đó.

**Ưu điểm:**

- Sử dụng phương pháp này có thể giúp tester tìm ra những lỗi điển hình thường xảy ra trong phần mềm hoặc những lỗi không thể tìm thấy khi thiết kế testcase theo hình thức formal

**Nhược điểm:**

- Kỹ thuật này thường được thực hiện bởi các tester có kinh nghiệm và không theo một quy tắc nhất định, thiết kế test case dựa nhiều vào cảm tính.

**1.4. Test Case**

Test case (trường hợp kiểm thử/ ca kiểm thử) là một trường hợp cần kiểm thử, nó bao gồm các thao tác/ hành động trên hệ thống, điều kiện cần (tiên quyết), các giá trị đầu vào, và kết quả mong đợi và các điều kiện kết thúc, được xây dựng cho mục đích hoặc điều kiện kiểm thử riêng biệt, như thực hiện một đường dẫn chương trình riêng hoặc để kiểm tra lại đúng với yêu cầu của spec. Một test case thì nên

chỉ kiểm tra một trường hợp, một khía cạnh cụ thể nào đó.

Test case là đảm bảo rằng các tính năng của ứng dụng hoặc phần mềm được thiết kế hoạt động chính xác và đúng như kỳ vọng. Ngoài ra, test case còn giúp xác định phạm vi kiểm thử; phát hiện lỗi để cải thiện chất lượng phần mềm; hỗ trợ quá trình vận hành, bảo trì và cập nhật; ...

### **Cấu trúc:**

1. Test case ID: Nó là duy nhất, là các chuỗi ký tự để định danh cho test case.
2. Function: Dựa theo chức năng của hệ thống có thể chia nhỏ các functions ra để tạo bộ test case rõ ràng hơn.
3. Description: Mô tả tóm tắt testcase sẽ được viết.
4. Pre-Condition: Bất cứ điều gì cần phải thiết lập bên ngoài của ứng dụng trước khi chúng ta có thể test được test case này. Ví dụ chúng ta cần đăng ký tài khoản trước khi đăng nhập
5. Test Case Procedure/ Execution Steps: Đây là mô tả chi tiết từng bước thực hiện và là danh mục quan trọng nhất trong một test case.
6. Test data: Những dữ liệu cần chuẩn bị để kiểm thử.
7. Expected Results: Mô tả kết quả mong đợi của một test case.
8. Result: thông thường sẽ là Pass, Fail hoặc Pending. Đây là kết quả thực tế khi thực hiện kiểm thử theo test case trên môi trường hệ thống.

Ngoài ra có thể thêm một số cột như Ngày thực hiện kiểm thử, Người thực hiện kiểm thử, ghi chú, ...

Cấu trúc của test case có thể thay đổi tùy theo dự án và công ty. Quan trọng nhất là test case phải đảm bảo có đủ các thông tin cần thiết để hiểu, thực hiện và đánh giá kết quả kiểm thử.

### **Phân loại:**

Test case thường được chia thành 4 nhóm chính:

- GUI test case: test case về giao diện



- Positive test case: test case về việc nhập giá trị đúng
- Negative test case: test case về việc nhập giá trị sai
- Combination Test case: test case có nhiều bước đúng sai đan xen nhưng bước cuối cùng luôn đúng

**Update Testcase:** Khi tài liệu yêu cầu hoặc thiết kế hệ thống đã thay đổi, mình phải cập nhật lại test case dựa trên tài liệu mới nhất đó. Hoặc có khi phải dựa trên tài liệu của bản build hiện tại (không phải là bản mới nhất) nếu khách hàng yêu cầu:

- Thêm mới test case
- Sửa testcase theo tài liệu yêu cầu mới (latest requirements)
- Xóa testcase không còn liên quan đến tài liệu yêu cầu mới

#### **Execute testcase:**

- Nếu kết quả thực tế giống với kết quả mong đợi: ghi “Pass” vào cột Status của test case đó.
- Nếu kết quả thực tế không giống với kết quả mong đợi: ghi “False” vào cột Status của test case đó rồi post bug và ghi BugID vào cột BugID.
- Nếu có lý do nào đó làm cho test case nào đó không chạy được (ví dụ như không thực hiện được step nào đó, chức năng cần test chưa làm xong, hoặc có bug làm cho không test được nữa,..v...v...): Ghi “Pending” vào cột Status của test case đó và ghi chú lý do tại sao không test được vào cột Ghi chú.

#### **Log Bug**

Nội dung gồm các phần chính sau:

- 1) Title Bug: nên viết ngắn gọn mô tả đầy đủ ý nghĩa của lỗi, tốt nhất khoảng 60 từ trở lại. Tránh để các tiêu đề chung chung
- 2) Description:
  - Pre-condition: điều kiện tiền đề để thực hiện các steps

- Procedure/Steps: mô tả step tái hiện lỗi. Nên đánh số cho từng bước để dev có thể dễ dàng thực hiện lại.
- Expected result: kết quả mong đợi
- Actual result: kết quả thực tế
- Evidence: Image, video, logfile
- Môi trường xảy ra lỗi: IE, Chrome, bản build,...
- Mức độ thường xuyên xảy ra lỗi: thỉnh thoảng, luôn luôn...

3) Priority: Set mức độ ưu tiên High/Medium/Low

4) Severity: Mức độ nghiêm trọng Critical/High/Medium/Low

5) Assignee: Assign cho dev fix lỗi

### 1.5. Các cấp độ kiểm thử phần mềm

Cấp độ kiểm thử phần mềm hay mức độ kiểm thử phần mềm là một quá trình trong đó mọi thành phần của phần mềm hệ thống đều được kiểm tra. Thông qua các mức độ kiểm thử phần mềm sẽ giúp chúng ta đánh giá được chức năng của ứng dụng được những yêu cầu đã chỉ định hay chưa. Đồng thời thông qua mức độ kiểm thử phần mềm sẽ giúp chúng ta tìm và sửa lỗi nhằm đảm bảo sản phẩm tạo ra có chất lượng tốt nhất.

Tất cả các giai đoạn của quá trình phát triển phần mềm đều trải qua bốn mức kiểm thử phần mềm là:

- Kiểm thử đơn vị (Unit Testing) - Dev
- Kiểm thử tích hợp (Integration Testing) - Dev
- Kiểm thử hệ thống (System Testing) – Tester
- Kiểm thử chấp nhận (Acceptance Testing) - Tester



Hình 1.12 Các cấp độ kiểm thử phần mềm

### Mức độ 1: Kiểm thử đơn vị - Unit Test

Kiểm thử đơn vị - Unit Testing là giai đoạn đầu tiên trong kiểm thử phần mềm. Với chức năng hoạt động đơn giản, không gây nhiều khó khăn trong việc kiểm thử, ghi nhận và phân tích kết quả do đó nếu phát hiện lỗi thì việc tìm kiếm nguyên nhân và sửa lỗi cũng đơn giản và tốn ít chi phí hơn. Tuy nhiên, kiểm thử mức đơn vị lại tốn nhiều thời gian để thực hiện, chưa phát hiện được các lỗi xảy ra khi tích hợp.

#### Mục tiêu:

- Xác định mỗi đơn vị phần mềm có đang thực hiện theo đúng thiết kế ban đầu hay không.
- Thông qua thử nghiệm sẽ giúp khắc phục những phát sinh do việc thay đổi hay bảo trì code.
- Unit Test giúp tiết kiệm chi phí, thời gian và thể diện khi phát hiện ra lỗi.

### Mức độ 2: Integration Testing – Kiểm thử tích hợp

Mỗi dự án phần mềm được hoàn thành bởi rất nhiều module do nhiều người code khác nhau. Integration Testing là cấp độ kiểm thử phần mềm tích hợp của các đơn vị riêng lẻ được kết hợp và thử nghiệm thành một nhóm thông qua việc tập trung vào kiểm tra truyền dữ liệu giữa các module.

**Mục tiêu:**

- Phát hiện lỗi giao tiếp xảy ra giữa các Unit cũng như lỗi của từng Unit
- Tích hợp các Unit đơn lẻ thành các hệ thống nhỏ và cuối cùng là nguyên hệ thống hoàn chỉnh chuẩn bị cho kiểm tra ở mức hệ thống

**Một số phương pháp kiểm thử tích hợp:**

- Phương pháp kiểm thử Bigbang
- Phương pháp kiểm thử Topdown
- Phương pháp kiểm thử Bottom up
- Phương pháp kiểm thử Sandwich

**Mức độ 3: System Testing – Kiểm thử hệ thống**

System Testing là giai đoạn thứ 3 của kiểm thử phần mềm cho phép phần mềm hoàn chỉnh và tích hợp được kiểm tra. System Testing tập trung nhiều hơn vào các chức năng của toàn bộ hệ thống. Kiểm thử hệ thống bao gồm kiểm thử chức và kiểm thử phi chức năng.

**Mục tiêu:**

- System Test kiểm tra thiết kế và toàn bộ hệ thống sau khi tích hợp có tuân thủ những yêu cầu đã được định sẵn trước đó .
- System Test kiểm thử cả các hành vi chức năng của phần mềm lẫn các yêu cầu về chất lượng như độ tin cậy, tính tiện lợi khi sử dụng, hiệu năng và bảo mật.

## **Mức độ 4: Acceptance Testing – Kiểm thử chấp nhận**

Sau khi kiểm tra hệ thống đã sửa tất cả hoặc hầu hết các lỗi, hệ thống sẽ được gửi đến người dùng hoặc khách hàng để kiểm tra chấp nhận. Về cơ bản kiểm thử chấp nhận cũng khá giống kiểm thử hệ thống nhưng được thực hiện bởi khách hàng.

Mục đích của Acceptance Testing đó là xác nhận lại sự tin tưởng vào hệ thống, các đặc tính thuộc về chức năng hoặc phi chức năng của hệ thống.

Có 2 loại kiểm thử chấp nhận đó là Alpha Testing và Beta Testing.

- Kiểm thử alpha: được thực hiện tại nơi phát triển phần mềm bởi những người trong tổ chức nhưng không tham gia phát triển phần mềm.
- Kiểm thử beta: được thực hiện tại bởi khách hàng/ người dùng cuối tại địa điểm của người dùng cuối.

### **1.6. Phân loại kiểm thử**

Có hai phương thức kiểm thử chính là Manual testing (kiểm thử thủ công) và Automation Testing (kiểm thử tự động). Kiểm thử tự động và kiểm thử thủ công là những bước vô cùng quan trọng để đảm bảo một dự án hoạt động trơn tru và hiệu quả. Do vậy vai trò của Tester trong cả hai lĩnh vực đều quan trọng như nhau.

#### **1.6.1. Khái niệm**

Kiểm thử thủ công: tester làm mọi công việc hoàn toàn bằng tay, từ viết test case đến thực hiện test, mọi thao tác như nhập điều kiện đầu vào, thực hiện một số sự kiện khác như click nút và quan sát kết quả thực tế, sau đó so sánh kết quả thực tế với kết quả mong muốn trong test case, điền kết quả test. Hiện nay, phần lớn các tổ chức, các công ty phần mềm, hoặc các nhóm làm phần mềm đều thực hiện kiểm thử thủ công là chủ yếu.

Kiểm thử tự động: thực hiện kiểm thử phần mềm bằng một chương trình đặc biệt với rất ít hoặc không có sự tương tác của con người, giúp cho người thực hiện

việc kiểm thử phần mềm (tester) không phải lặp đi lặp lại các bước nhàm chán. Công cụ kiểm thử tự động có thể lấy dữ liệu từ file bên ngoài (excel, csv...) nhập vào ứng dụng, so sánh kết quả mong đợi (từ file excel, csv...) với kết quả thực tế và xuất ra báo cáo kết quả kiểm thử.

### 1.6.2. Điểm khác nhau giữa kiểm thử thủ công và kiểm thử tự động

*Bảng 1.1 Điểm khác nhau giữa kiểm thử thủ công và kiểm thử tự động*

Thông số	Kiểm thử tự động	Kiểm thử thủ công
Định nghĩa	Kiểm thử tự động sử dụng các công cụ tự động để thực hiện các trường hợp kiểm thử.	Các trường hợp kiểm thử được thực hiện bởi tester trên phần mềm
Thời gian xử lý	Kiểm thử tự động nhanh hơn đáng kể so với phương pháp kiểm thử thủ công.	Các trường hợp kiểm thử được thực hiện bởi tester trên phần mềm
Đầu tư ban đầu	Đầu tư ban đầu trong kiểm thử tự động cao hơn.	Đầu tư ban đầu trong kiểm thử thủ công là tương đối thấp hơn kiểm thử tự động.
Độ tin cậy	Kiểm thử tự động là một phương pháp đáng tin cậy, vì được thực hiện bởi các công cụ và scripts nên chính xác và không gây nhầm lẫn	Kiểm thử thủ công có thể bị nhầm lẫn và dễ bị lỗi.
Đầu tư	Đầu tư là cần thiết cho các công cụ kiểm thử cũng như các kỹ sư kiểm thử tự động hóa	Đầu tư là cần thiết cho nguồn nhân lực.
Hiệu quả chi phí	Không hiệu quả nếu kiểm thử hồi quy với số lượng nhỏ	Kết quả của kiểm thử thủ công thường được ghi lại trong Excel hoặc Word

Báo cáo Kiểm thử	Với kiểm thử tự động, tất cả các bên liên quan có thể đăng nhập vào hệ thống tự động và xem được kết quả kiểm thử	Kết quả của kiểm thử thủ công thường được ghi lại trong Excel hoặc Word
Thay đổi giao diện người dùng	Đối với ngay cả một thay đổi nhỏ trong giao diện người dùng của AUT (Application Under Test), scripts kiểm thử tự động cần phải được sửa đổi để hoạt động như mong đợi	Những thay đổi nhỏ như thay đổi trong id, class, v.v. của một nút sẽ không cản trở việc thực hiện một trình kiểm thử thủ công.
Giao diện đối với người dùng	Kiểm thử tự động không liên quan đến hành động của con người. Vì vậy, không phù hợp với người dùng và trải nghiệm tích cực khách hàng.	Phương pháp kiểm thử thủ công cho phép người dùng quan sát, có thể hữu ích để cung cấp hệ thống thân thiện với người dùng.
Kiểm thử hàng loạt	Có thể chạy hàng loạt các Script.	Kiểm thử thủ công không thể thực hiện hàng loạt.
Kiến thức lập trình	Kiến thức lập trình là bắt buộc để thực hiện kiểm thử tự động.	Không cần kiến thức lập trình trong Kiểm thử thủ công.
Thời hạn (Deadlines)	Các kiểm thử tự động không có rủi ro trễ Deadlines	Kiểm thử thủ công có nguy cơ trễ Deadlines cao.
Framework	Kiểm thử tự động hóa sử dụng các Framework như Data Drive, Keyword, Hybrid để tăng tốc quá trình tự động hóa.	Kiểm thử thủ công không sử dụng Framework nhưng có thể sử dụng guidelines, checklists, quy trình nghiêm ngặt để tạo ra một số test case nhất định.

Tài liệu	Kiểm thử tự động như một tài liệu để training về những test cases unit tự động. Một developer mới có thể tham khảo và nhanh chóng hiểu yêu cầu.	Các trường hợp kiểm thử thủ công không cung cấp giá trị training
Test Design	Kiểm thử đơn vị Tự động bắt buộc xây dựng thiết kế dựa trên kiểm thử	Kiểm thử đơn vị thủ công không bắt buộc thiết kế trong quá trình coding

### 1.6.3. Ưu điểm và nhược điểm

*Bảng 1.2 Ưu điểm và nhược điểm của kiểm thử thủ công và kiểm thử tự động*

Kiểm thử	Ưu điểm	Nhược điểm
Thủ công	<ul style="list-style-type: none"> <li>- Cho phép tester thực hiện việc kiểm thử khám phá</li> <li>- Thích hợp kiểm tra sản phẩm lần đầu tiên, các test case chỉ phải thực hiện một số ít lần</li> <li>- Giảm được chi phí ngắn hạn.</li> </ul>	<ul style="list-style-type: none"> <li>- Tốn thời gian.</li> <li>- Đối với mỗi lần release, người kiểm thử vẫn phải thực hiện lại một tập hợp các test case đã chạy dẫn đến sự mệt mỏi và lãng phí effort.</li> </ul>
Tự động	<ul style="list-style-type: none"> <li>- Thích hợp với trường hợp phải test nhiều lần cho một case, có tính ổn định và tin cậy cao hơn so với kiểm thử thủ công</li> <li>- Có thể thực hiện các thao tác lặp đi lặp lại (nhập dữ liệu, click, check kết quả...) giúp tester không phải làm những việc gây nhàm chán và dễ nhầm lẫn như vậy.</li> </ul>	<ul style="list-style-type: none"> <li>- Tốn kém hơn kiểm thử thủ công, chi phí đầu tư ban đầu lớn</li> <li>- Không thể test được hình ảnh hay các case phức tạp</li> </ul>



#### **1.6.4. Khi nào thì sử dụng kiểm thử thủ công và khi nào thì sử dụng kiểm thử tự động?**

- Kiểm thử thủ công sẽ được sử dụng khi:

Triển khai tính năng mới của sản phẩm. Rất khó để hoàn thành cả việc hiện thực cho sản phẩm và viết test script tự động để test tính tăng mới. Để thời gian release được đảm bảo, manual testing vẫn được ưu tiên sử dụng hơn.

Cần thực hiện Exploratory Testing (Kiểm thử thăm dò): Đây là loại kiểm thử đòi hỏi phải thử nghiệm của kiến thức, kinh nghiệm, phân tích / logic kỹ năng, sáng tạo và trực giác. Xét nghiệm này được đặc trưng bởi các tài liệu ở đây kém bằng văn bản kỹ thuật, hoặc một thời gian ngắn để thực hiện. Chúng ta cần những kỹ năng của con người để thực hiện quá trình kiểm thử.

Đối với những dự án ngắn hạn và không nhiều kinh phí hoặc những dự án tần suất thực hiện test thấp, khoảng 2-3 lần/năm, không nên đầu tư vào automation testing. Manual testing là sự lựa chọn phù hợp nhất.

- Kiểm thử tự động được sử dụng khi:

Test case thường xuyên được yêu cầu thực hiện như: acceptance testing, regression testing chẳng hạn (thay đổi mã thường xuyên và khả năng chạy các hồi quy một cách kịp thời.). Việc thực hiện automation testing cho các test case này giúp giảm chi phí của QA cho những test case lặp đi lặp lại, giúp họ có thời gian đầu tư nghiên cứu yêu cầu của phần mềm và thực hiện các test case khác.

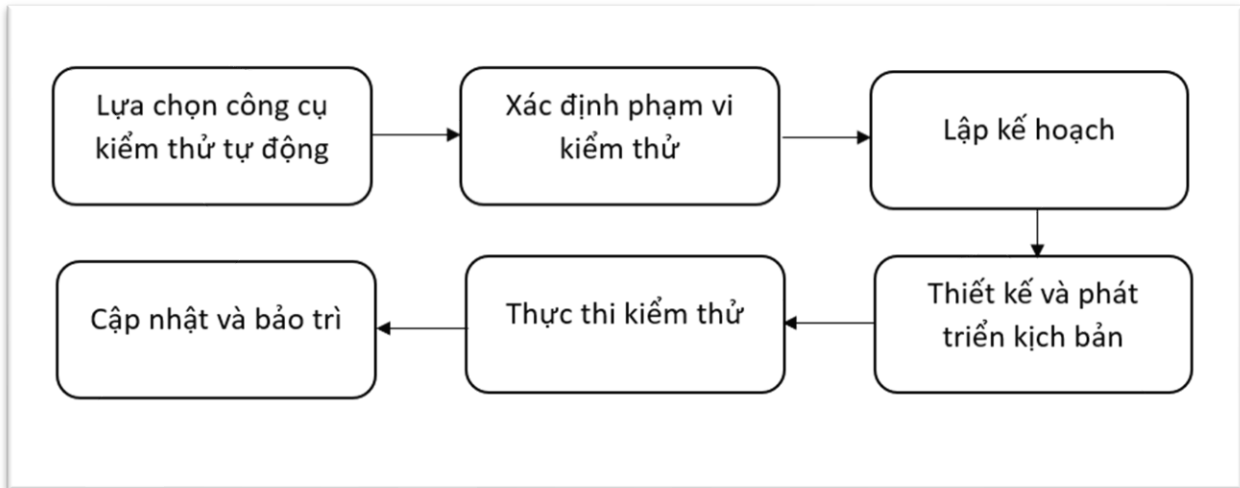
Test case cần thực hiện trên nhiều môi trường khác nhau: trên nhiều browser khác nhau chẳng hạn. Phù hợp với dự án lớn, liên tục phát triển trong thời gian dài hạn

#### **1.7. Quy trình kiểm thử tự động**

Việc kiểm thử tự động thì tốn kém hơn kiểm thử thủ công rất nhiều nếu chỉ thực hiện một lần. Để đạt được lợi ích, kiểm thử tự động cần phải được lựa chọn và thực thi một cách cẩn thận theo một tiến trình cụ thể.

Trước khi thực hiện kiểm thử tự động, kỹ sư kiểm thử và người quản lý phải có hiểu biết rõ ràng về kiểm thử tự động, bao gồm nhu cầu, mục tiêu, lợi ích, các vấn đề và thách thức.

Một tiến trình hiệu quả để thực hiện tự động hóa kiểm thử bao gồm các bước như ở hình 1.13



Hình 1.13 Quy trình kiểm thử tự động

#### **Bước 1:** Lựa chọn công cụ kiểm thử.

Việc lựa chọn công cụ kiểm thử tự động phù hợp thường phụ thuộc vào loại của ứng dụng đang được kiểm tra và môi trường mà ứng dụng đó sẽ chạy.

#### **Bước 2:** Xác định phạm vi kiểm thử tự động

Trong bước này, cần xem xét khoanh vùng các luồng nghiệp vụ kiểm thử phù hợp cho việc tự động hóa nó, chuẩn bị dữ liệu và môi trường diễn ra kiểm thử. Dưới đây là một số yếu tố cần xem xét khi xác định phạm vi thử nghiệm tự động

- Các tính năng chính, quan trọng của ứng dụng
- Các trường hợp kiểm thử có nhiều dữ liệu
- Các tính năng dùng chung (common) trên ứng dụng
- Những vùng khả thi về kỹ thuật mà công cụ đáp ứng được
- Những nghiệp vụ hay được tái sử dụng

- Mức độ phức tạp của các test cases
- Khả năng sử dụng các trường hợp kiểm thử giống nhau để test trên nhiều trình duyệt (với trường hợp kiểm thử ứng dụng web)

### **Bước 3:** Lập kế hoạch.

Mục đích chính của bước này là lập kế hoạch xác định các đối tượng cần phải tự động, mục đích, chiến lược, yêu cầu, lịch trình, kinh phí. Trong thực tế, kế hoạch tự động hóa thường được tạo cho một sản phẩm, hoặc một dòng sản phẩm ngay ở giai đoạn đầu của tiến trình phát triển phần mềm.

### **Bước 4:** Thiết kế và phát triển kịch bản kiểm thử tự động.

Ở giai đoạn này, người kiểm thử bắt tay vào thiết kế các kịch bản tự động và dùng công cụ được chọn để tạo ra các script tự động hóa. Các công việc điển hình như: Thiết kế Framework và các tính năng, thiết kế các kịch bản tự động, viết script và kiểm tra tính ổn định của script, xem xét tính đúng đắn của script so với thiết kế

### **Bước 5:** Thực thi kiểm thử

Đây là giai đoạn áp chót trong quy trình kiểm thử tự động hóa. Khi đã tạo xong các script tự động, đã đến lúc chạy các script này để thực hiện kiểm thử ngay trên ứng dụng. Kết quả của việc thực thi các script thường sẽ được tổng hợp vào một báo cáo cho biết số lượng test cases PASS/FAIL và kèm theo các bằng chứng ghi lại hình ảnh/ trạng thái tại thời điểm xảy ra các lỗi được tìm thấy.

### **Bước 6:** Cập nhật và bảo trì

Khi phần mềm được cập nhật thêm tính năng mới hoặc chỉnh sửa, thì bộ script sẽ được chạy để kiểm tra xem mức độ ảnh hưởng của các tính năng mới tới các tính năng hiện tại, liệu chúng có còn hoạt động đúng hay không. Khi phần mềm có cập nhật, những thay đổi trên phần mềm có thể làm cho bộ script không còn đúng, đây là lúc xem xét chỉnh sửa lại script cho phù hợp với những thay đổi.

### 1.7.1. Một số Test automation framework khác

#### *Google EarlGrey*

EarlGrey là một khung kiểm tra tự động hóa giao diện người dùng iOS gốc cho phép các nhà phát triển viết các bài kiểm tra ngắn gọn và rõ ràng. Các bài kiểm tra dễ dàng hơn để viết và duy trì. Nó có tính năng đồng bộ hóa tích hợp mạnh mẽ với giao diện người dùng, hình ảnh động, yêu cầu mạng, v.v.

#### *Cucumber*

Cucumber là một công cụ hướng hành vi chủ yếu được sử dụng để viết các bài kiểm tra chấp nhận cho các ứng dụng web. Nó cung cấp cho người dùng một thiết lập nhanh chóng và dễ dàng để bắt đầu và cũng cho phép họ sử dụng lại mã trong các thử nghiệm khác nhau. Khung công tác ban đầu được thực hiện trong Ruby và bây giờ được mở rộng sang khung công tác Java.

#### *Appium*

Appium chủ yếu được thiết kế để kiểm tra các ứng dụng di động. Nó được thiết kế theo cách mà chúng ta không phải biên dịch lại ứng dụng của mình hoặc sửa đổi nó theo bất kỳ cách nào để chạy thử nghiệm. Nó là một khuôn khổ đa nền tảng có thể được sử dụng để chạy trên các nền tảng khác nhau bằng cách sử dụng cùng một API.

#### *Robot Framework*

Đây là một trong những khuôn khổ tự động hóa kiểm tra chung nhất được sử dụng để phát triển theo hướng kiểm tra chấp nhận và kiểm tra chấp nhận. Robot Framework có thể được sử dụng trong các môi trường phân tán, không đồng nhất, nơi yêu cầu sử dụng các công nghệ và giao diện khác nhau. Đây là một khuôn mẫu đa nền tảng cung cấp cú pháp dữ liệu kiểm tra dạng bảng dễ sử dụng.

Khung cho phép tích hợp dễ dàng, ghi nhật ký chi tiết và báo cáo thử nghiệm rõ ràng. Điểm mạnh của Robot Framework chính là được viết trên nền tảng Python

và được hỗ trợ bởi số lượng thư viện dành cho tester, Robot Framework rất dễ sử dụng cũng như viết test script và có thể chạy được trên mọi nền tảng khác nhau mà không cần chỉnh sửa test script.

### *Gauge*

Gauge là một trong những công cụ tự động hóa thử nghiệm nhẹ tiên tiến cung cấp các tính năng đa nền tảng. Nó giới thiệu một cú pháp đơn giản, phong phú và linh hoạt và thực hiện thực thi theo hướng dữ liệu. Các trường hợp thử nghiệm trong khuôn khổ này có thể dễ dàng hiểu và duy trì, đồng thời nó có kiến trúc mô-đun cung cấp các plugin có khả năng mở rộng cao.

#### **1.7.2. Tại sao nên lựa chọn Automation testing**

+ Độ tin cậy cao: Công cụ kiểm thử tự động có sự ổn định cao vì hoạt động theo quy trình định sẵn, đặc biệt trong trường hợp nhiều test case, các bài kiểm tra tiêu chuẩn lặp đi lặp lại nhằm tránh không thể bỏ.

+ Khả năng lặp: Mình có thể test cách phần mềm xử lý (tính năng/hiệu năng) khi gặp tình huống chạy lặp đi lặp lại nhiều lần trên cùng script test giúp các Tester xử lý trường hợp lặp đi lặp lại các thao tác như: click, nhập dữ liệu, check kết quả,...) Đây còn gọi là performance/load testing.

+ Khả năng tái sử dụng: Các script có thể sử dụng lại và không cần script mới mọi lúc. Ngoài ra, các script có thể thực hiện lại các bước chính xác như những gì đã diễn ra trước đó.

+ Tiết kiệm thời gian: Automation test giúp chạy test nhanh hơn với tốc độ nhanh hơn ít nhất 10 lần so với tốc độ kiểm thử thủ công. Nếu cần 5 phút để thực thi một test case cách thủ công thì chỉ cần khoảng 30s để thực thi tự động.

+ Chi phí thấp: nếu áp dụng kiểm thử tự động đúng cách, chúng ta có thể tiết kiệm được nhiều chi phí về thời gian và nhân lực.

## CHƯƠNG 2: TÌM HIỂU VỀ CÔNG CỤ SELENIUM

### 2.1. Giới thiệu công cụ Selenium

#### 2.1.1 Lịch sử phát triển

Selenium được phát triển bởi Jason Huggins làm việc tại ThoughtWorks có trụ sở tại Chicago có tên ban đầu là JavaScript Test Runner. Tự động kiểm tra bất kỳ ứng dụng nào là cốt lõi đối với phong cách của ThoughtWork, dựa trên khuynh hướng Agile của công ty này. Đây là tiền đề của Selenium IDE và Selenium RC.

Selenium WebDriver được phát triển Simon Stewart năm 2007, Nó không dựa vào JavaScript để thực hiện công việc nặng nhọc mà thay vào đó có một ứng dụng khách cho mỗi trình duyệt được mã hóa từ đầu. Nó cũng có API “cấp cao hơn” so với Selenium-RC và cho thấy nhiều hứa hẹn. Simon đã trình bày công cụ này tại GTAC và bắt đầu nghiên cứu khả năng tương thích với Selenium-RC, điều này dẫn đến kết luận rõ ràng rằng hai dự án nên hợp nhất.

#### 2.1.2 Khái niệm

Selenium là bộ kiểm thử tự động miễn phí (mã nguồn mở) dành cho các ứng dụng web trên các trình duyệt và nền tảng khác nhau như: Firefox, Google Chrome, Microsoft edge. Selenium không chỉ là 1 công cụ độc lập mà là 1 bộ công cụ của phần mềm, mỗi bộ đều đáp ứng được nhu cầu kiểm thử khác nhau của 1 tổ chức. Selenium hoạt động bằng cách mô phỏng các thao tác của người dùng trên web page hoặc web element. Khi viết test bằng Selenium, tester chỉ định các hành động mà họ muốn người dùng thực hiện, Selenium sẽ tự động thực thi các hành động đó trên trình duyệt.

#### 2.1.3 Những tính năng của Selenium

- Selenium IDE hỗ trợ tính năng playback giúp chúng ta có thể sử dụng các bài test của người khác và không cần phải biết ngôn ngữ script
- Selenium là một nền tảng kiểm thử dựa trên cloud giúp tester có thể lưu lại

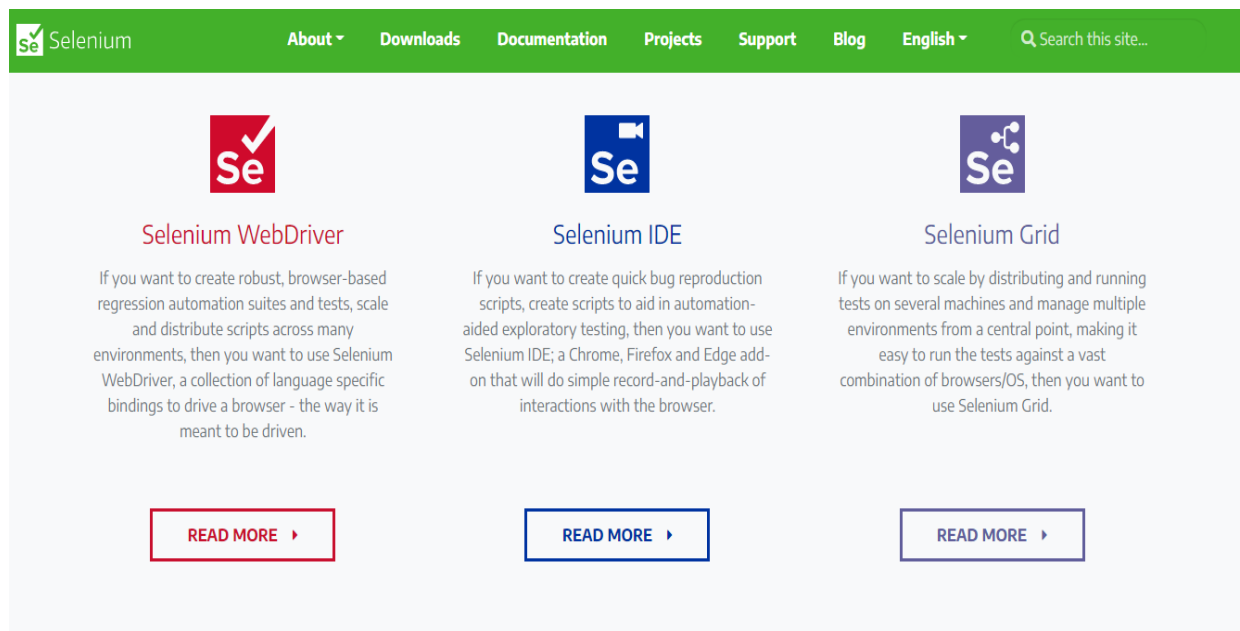
thao tác và xuất ra dưới dạng script đơn giản, dễ hiểu.

- Selenium hỗ trợ nhiều hệ điều hành, ngôn ngữ và trình duyệt khác nhau.
- Giúp chúng ta có thể chạy cùng lúc nhiều bài test để giảm thời gian và tăng hiệu quả
- Quá trình kiểm thử của Selenium hao tốn ít tài nguyên và yêu cầu cấu hình thiết bị thấp hơn các công cụ khác.
- Selenium WebDriver không yêu cầu cài đặt server, test script của chúng ta sẽ trực tiếp tương tác với trình duyệt.

#### 2.1.4 Selenium bao gồm những công cụ nào?

Tính đến thời điểm hiện tại, Selenium không phải là một công cụ duy nhất. Selenium có đến 4 công cụ để chúng ta có thể tìm ra mục đích sử dụng và lựa chọn thích hợp.

- Selenium Integrated Development Environment (IDE)
- Selenium Remote Control
- Selenium WebDriver
- Selenium Grid



Hình 2. 1 Những công cụ trong Selenium

## **Selenium IDE**

Sử dụng để học và tìm hiểu về các khái niệm kiểm tra tự động và Selenium như:

- Học và chọn các lệnh như: type, open, clickAndWait, assert, verify,...
- Học cách sử dụng bộ định vị như: id, name, xpath, css selector
- Tùy chỉnh JavaScript bằng cách sử dụng runScript

## **Selenium Remote Control**

- Tạo test với ngôn ngữ dễ hiểu hơn Selenese
- Sử dụng để chạy thử nghiệm test trên nhiều trình duyệt khác nhau trên nhiều hệ điều hành khác nhau.
- Triển khai thêm nhiều môi trường thử nghiệm khác bằng cách kết hợp sử dụng với Selenium Grid.

## **Selenium WebDriver**

- Để dùng một ngôn ngữ nhất định trong quá trình kiểm thử
- Kiểm thử nhiều ứng dụng dựa trên nền Ajax
- Tạo bài test trên trình duyệt HtmlUnit
- Tạo kết quả kiểm tra tùy chỉnh

## **Selenium Grid**

- Dùng để chạy các script của Selenium Remote Control trên nhiều hệ điều hành, nhiều trình duyệt khác nhau cùng lúc
- Giúp chúng ta có thể tạo ra một bộ test với số lượng lớn bài test nhỏ cùng lúc để tăng tốc hoàn thành việc kiểm tra.

### **2.1.5 Ưu điểm và nhược điểm của Selenium**

#### **Ưu điểm:**

- Quá trình cài đặt và sử dụng rất đơn giản
- Yêu cầu cấu hình phần cứng thấp
- Hỗ trợ đa dạng hệ điều hành, trình duyệt và hỗ trợ nhiều ngôn ngữ lập trình
- Có bộ API hoàn thiện



- Tạo ra bộ test lớn với nhiều bài test nhỏ để tiết kiệm thời gian
- Hỗ trợ kiểm thử tự động thay cho người dùng thật giúp tiết kiệm nhiều nhân lực

### **Nhược điểm:**

- Selenium IDE không hỗ trợ thực hiện tính toán hoặc câu lệnh phức tạp được
- Quá trình cài đặt Selenium Webdriver tốn thời gian và cần có nhiều kinh nghiệm
- Nếu chúng ta chạy quá nhiều test so với khả năng đáp ứng của phần cứng, chắc chắn quá trình test của chúng ta sẽ không có kết quả tốt và đôi khi sẽ đứng máy
- Không có bộ phận hỗ trợ kỹ thuật chuyên dụng. Vì thế, khi triển khai các dự án thương mại và bị lỗi, chúng ta sẽ cần phải tìm đơn vị thứ 3 hỗ trợ
- Chỉ hỗ trợ ứng dụng web.

## **2.2 Giới thiệu framework Mocha**

Mocha là một khung kiểm tra JavaScript giàu tính năng chạy trên Node.js và trong trình duyệt, giúp việc kiểm tra không đồng bộ trở nên đơn giản và thú vị. Các thử nghiệm Mocha chạy tuần tự, cho phép báo cáo linh hoạt và chính xác, đồng thời ánh xạ các ngoại lệ chưa được phát hiện vào các trường hợp kiểm thử chính xác. Cấu trúc cơ bản của một tệp kiểm thử:

```
describe("hooks", function() => {
  //Những điều kiện không bắt buộc
  before(function() { // Chạy trước tất cả các bài kiểm thử});
  after(function() { // Chạy sau tất cả các bài kiểm thử});
  beforeEach(function() { //Chạy trước mỗi bài kiểm thử });
  afterEach(function() { //Chạy sau mỗi bài kiểm thử});
  // Test cases
  it("name", () => { // Các câu lệnh thực thi test case;});
});
```

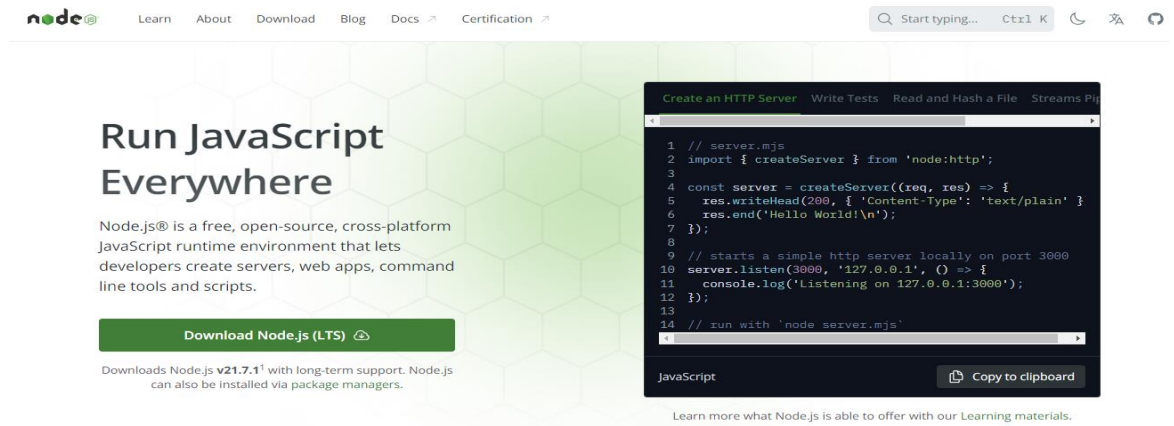
Trong trình duyệt, các tệp kiểm tra được tải theo từng thẻ `<script>` và quá trình gọi `mocha.run()` bắt đầu ở bước 9 theo chế độ nối tiếp dưới đây:

1. Người dùng thực thi mocha
2. Tải các tùy chọn từ tập tin cấu hình (nếu có)
3. Mocha xử lý mọi tùy chọn dòng lệnh được cung cấp
4. Nếu tìm thấy các flags để thực thi tệp node, mocha sẽ sinh ra node trong một tiến trình con, tự thực thi với các flags và không sinh ra các tiến trình con.
5. Mocha tải các mô-đun được chỉ định bởi `--require`.
6. Mocha xác thực mọi trình báo cáo hoặc giao diện được tải qua `--require` hoặc bằng cách khác.
7. Mocha phát hiện ra các tập tin kiểm thử, khi không có tệp hoặc thư mục nào, nó sẽ tìm các tệp có phần mở rộng `.js`, `.mjs` hoặc `.cjs` trong mục test (không tệp con của nó), liên quan đến thư mục làm việc hiện tại.
8. Giao diện bdd sẽ tải các tệp không theo thứ tự cụ thể. Khi các tệp được tải, Mocha sẽ tìm nhưng không thực thi phần nào trong bộ test suite đó.
9. Mocha chạy các thiết lập được cài đặt (nếu có)
10. Mocha bắt đầu với bộ “root” để thực thi
11. Mocha thực thi một lần “before all” cho cả bộ trong mỗi lần chạy
12. Với mỗi trường hợp kiểm thử, Mocha sẽ thực thi “before each”, sau đó thực thi các câu lệnh bên trong (bao gồm cả báo cáo) và cuối cùng là thực thi “after each”.
13. Nếu trong bộ test có bộ con, sẽ lặp lại các bước 10,11,12 cho mỗi bộ con. Mỗi bộ con đều kế thừa “before each” và “after each” được xác định trong phần cha của nó.
14. Mocha thực thi “after all”
15. Mocha in phần kết quả

## 2.3. Cài đặt Selenium Webdriver và môi trường kiểm thử

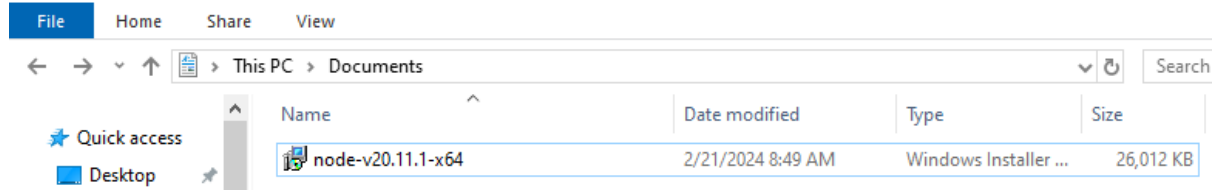
### 2.3.1. Cài đặt Node.js và npm

Bước 1: Truy cập “<https://nodejs.org/en/download>” và tải xuống Node.js phiên bản mới nhất phù hợp với thiết bị.

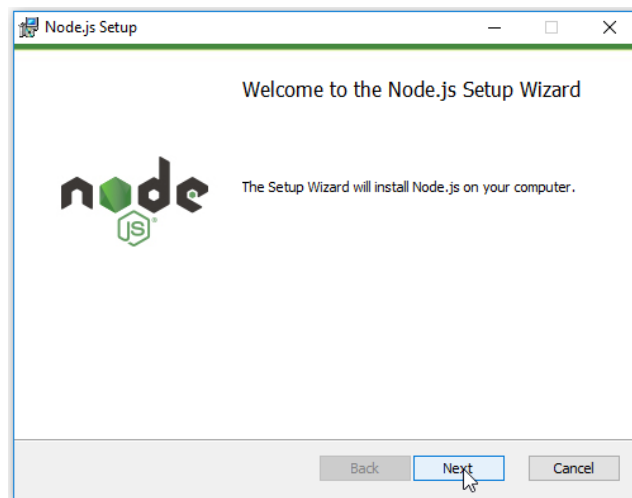


Hình 2.2 Cài đặt Nodejs(1)

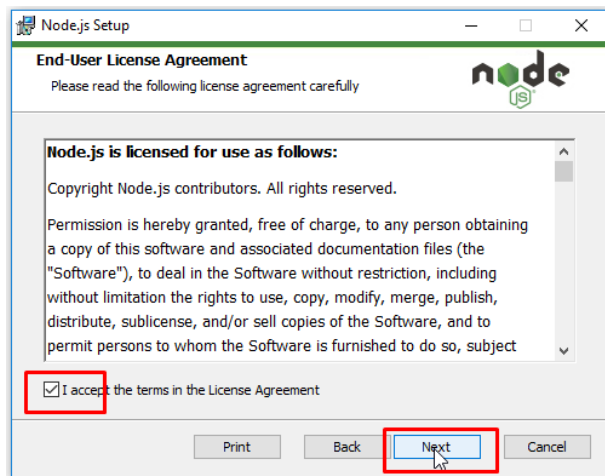
Bước 2: Click đúp vào file .msi vừa tải về để tiến hành cài đặt.



Hình 2.3: Cài đặt Nodejs(2)

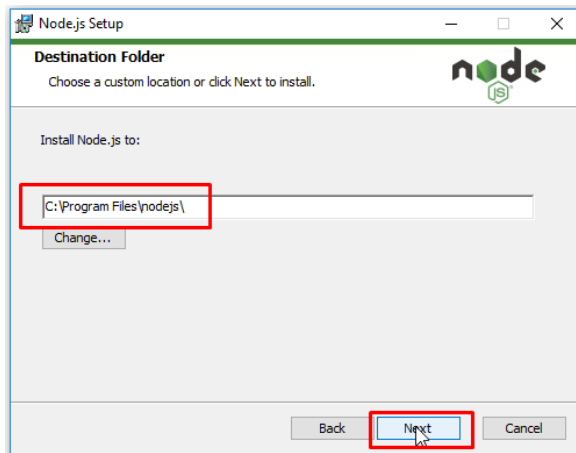


Hình 2.4 Cài đặt Nodejs(3)

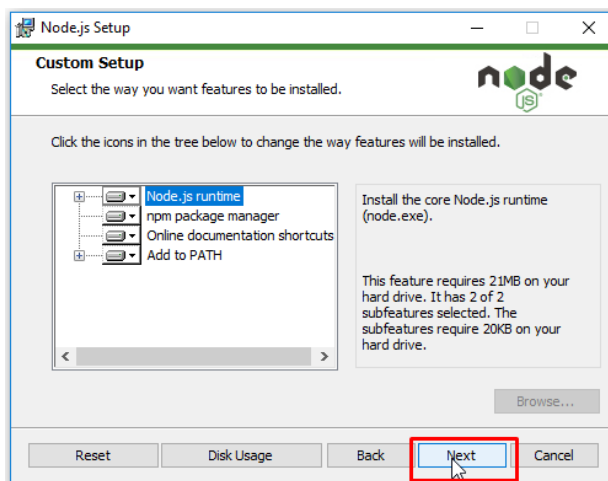


Hình 2.5 Cài đặt Nodejs(4)

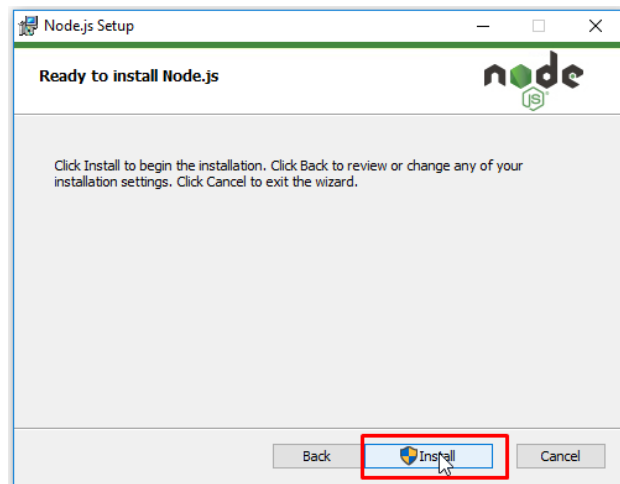
Bước 3: Chọn thư mục để cài đặt nodejs



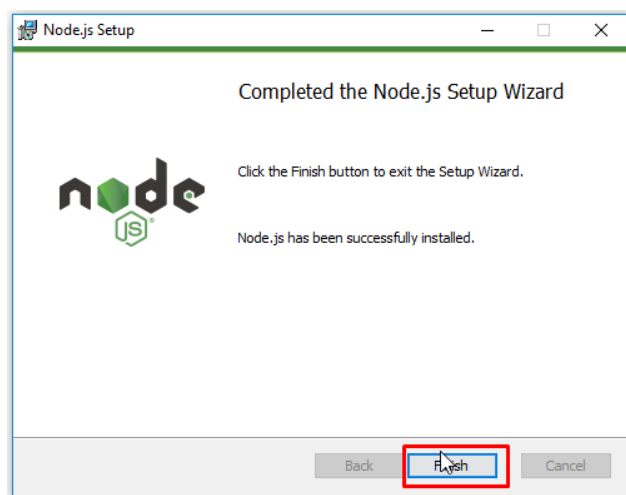
Hình 2.6 Cài đặt Nodejs(5)



Hình 2.7 Cài đặt Nodejs(6)



Hình 2.8 Cài đặt Nodejs(7)



Hình 2. 9 Cài đặt Nodejs(8)

Bước 4: Kiểm tra version của nodejs và npm

Để kiểm tra version của nodejs và npm, ta mở cửa sổ terminal của Visual Studio Code và nhập lệnh.

- Lệnh “`node -v`” để kiểm tra version nodejs

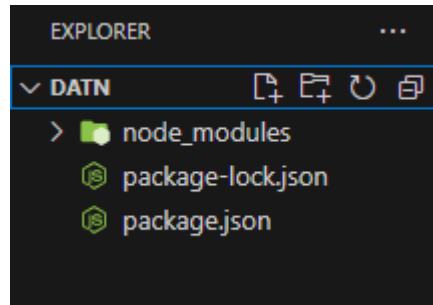
```

PROBLEMS  OUTPUT  TERMINAL  ...
PS D:\STUDY\DATN> node -v
v20.11.1
PS D:\STUDY\DATN> npm -v
10.2.4
PS D:\STUDY\DATN>
  
```

Hình 2.10 Kiểm tra version của nodejs và npm

- Lệnh “`npm -v`” để kiểm tra version npm

Sau khi cài đặt xong sẽ xuất hiện folder “`node_module`” chứa thông tin của dự án



Hình 2.11 Folder `node_module` sau khi cài đặt

### 2.3.2. Cài đặt Selenium webdriver

Điền lệnh “`npm install selenium-webdriver`” vào terminal trong Visual Studio Code

```
PS D:\STUDY\DATN> npm install selenium-webdriver

added 16 packages in 3s
PS D:\STUDY\DATN> 
```

Hình 2.12 Cài đặt Selenium webdriver

### 2.3.3. Cài đặt framework Mocha

Điền lệnh “`npm install --save-dev mocha`” vào terminal trong Visual Studio Code

```
PS D:\STUDY\123\DoThiHue> npm install --save-dev mocha

up to date, audited 664 packages in 2s

144 packages are looking for funding
  run `npm fund` for details

1 moderate severity vulnerability

To address all issues, run:
  npm audit fix
```

Hình 2.13 Cài đặt framework Mocha

## CHƯƠNG 3. THỰC HIỆN KIỂM THỬ WEBSITE

### 3.1. Giới thiệu chương trình

#### 3.1.1 Giới thiệu về website

Website Divimenti được phát triển để người dùng có thể chia sẻ thông tin và kết nối với nhiều người dùng khác ở mọi lúc, mọi nơi, mọi lứa tuổi mà chỉ cần có kết nối Internet. Với giao diện đơn giản và dễ sử dụng, Divimenti cho phép người dùng chia sẻ văn bản, hình ảnh và nhiều nội dung đa dạng khác chỉ trong vài cú nhấp chuột. Ngoài ra, Divimenti còn cung cấp tính năng nhắn tin để người dùng có thể trò chuyện cùng với bạn bè một cách nhanh chóng, thuận tiện và miễn phí.

#### 3.1.2 Chức năng của hệ thống

Mỗi người dùng, tùy thuộc vào là người quản trị hay người dùng sẽ có duy nhất một tên đăng nhập và mật khẩu để sử dụng cho việc đăng nhập tài khoản trên hệ thống.

#### Về phía người dùng:

*Bảng 3.1 Chức năng hệ thống về phía người dùng*

Chức năng	Mô tả
Đăng ký	<ul style="list-style-type: none"> <li>- <b>Tài khoản:</b> bắt buộc nhập, nhập tối đa 100 kí tự. Hiển thị mặc định: “Tài khoản” Nếu bỏ trống cảnh báo: Vui lòng điền vào trường này! Không nhập được quá 100 ký tự</li> <li>- <b>Email:</b> bắt buộc nhập, nhập tối đa 254 kí tự. Hiển thị mặc định: “Email” Có @. Nếu không có @, báo lỗi "Enter a valid email address." Nếu bỏ trống cảnh báo: Vui lòng điền vào trường này!</li> </ul>

	<p>Không nhập được quá 254 ký tự</p> <p>- <b>Số điện thoại:</b> bắt buộc nhập, là số, nhập tối đa 15 ký tự.          Hiện thị mặc định: “Số điện thoại”</p> <p>Nếu bỏ trống cảnh báo: Vui lòng điền vào trường này!</p> <p>- <b>Mật khẩu:</b> bắt buộc nhập, có ít nhất 8 ký tự bao gồm chữ hoa, chữ thường, số và ký tự đặc biệt. Nếu không đủ báo lỗi: "Passwords must have at least 8 characters, including uppercase letters, lowercase letters, numbers and special characters to ensure security!"</p> <p>Nếu bỏ trống cảnh báo: Vui lòng điền vào trường này!</p> <p>Các ký tự chuyển thành chấm tròn sau khi nhập.          Hiện thị mặc định: “Mật khẩu”</p> <p>- <b>Button "Hiện":</b> ban đầu button hiển thị "Hiện"</p> <p>Click lần 1 button chuyển "Ẩn", các ký tự đã nhập hiển thị.</p> <p>Click lần 2 button chuyển thành "Hiện", các ký tự chuyển thành chấm tròn.</p> <p>Nếu textbox không có ký tự nào thì hiển thị mặc định "Mật khẩu"</p> <p>- <b>Tên:</b> bắt buộc nhập, nhập tối đa 100 ký tự.          Hiện thị mặc định: “Tên”</p> <p>Nếu bỏ trống cảnh báo: Vui lòng điền vào trường này!</p> <p>Không nhập được quá 100 ký tự</p> <p>- <b>Họ:</b> bắt buộc nhập, nhập tối đa 100 ký tự.          Hiện thị mặc định: “Họ”</p> <p>Nếu bỏ trống cảnh báo: Vui lòng điền vào trường này!</p> <p>Không nhập được quá 100 ký tự</p>
--	--



	<p>- <b>Ngày sinh:</b> điền hoặc chọn . Nếu nhập ngày sinh lỗi thì thông báo "Enter a valid date."</p> <p>- <b>Giới tính:</b> mặc định chọn “Male”, người dùng có thể chọn 1 trong 3 lựa chọn Male, Female, Other.</p> <p>- <b>Button "Đăng ký":</b></p> <p>Nếu bỏ trống tất cả các trường và click “Đăng ký” thì báo lỗi "Vui lòng điền vào trường này" trường đầu tiên chưa nhập từ trên xuống.</p> <p>Nếu người dùng đăng ký tài khoản với có tên trùng với tài khoản khác trong cơ sở dữ liệu thì hiển thị thông báo "User with this Username already exists.".</p> <p>Nếu người dùng đăng ký tài khoản với có Email trùng với tài khoản khác trong cơ sở dữ liệu thì hiển thị thông báo "User with this Email already exists.".</p> <p>Nếu người dùng đăng ký tài khoản với có tên trùng với tài khoản khác trong cơ sở dữ liệu thì hiển thị thông báo "User with this Phone already exists.".</p> <p>- <b>Link "Đăng nhập":</b> click vào link thì chuyển sang trang đăng nhập</p>
Đăng nhập	<p>- <b>Tài khoản:</b> bắt buộc nhập, nhập tối đa 100 ký tự.</p> <p>Hiển thị mặc định: “Tài khoản”</p> <p>Nếu bỏ trống báo lỗi: “Vui lòng điền vào trường này”</p> <p>Không nhập được quá 100 ký tự.</p> <p>- <b>Mật khẩu:</b> bắt buộc nhập tối thiểu 8 ký tự có phân biệt chữ hoa chữ thường.</p> <p>Hiển thị mặc định: “Mật khẩu”</p>

	<p>Sau khi nhập, kí tự chuyển thành chấm tròn.</p> <p>Nếu bỏ trống báo lỗi: “Vui lòng điền vào trường này”</p> <p>- <b>Button “Login”</b>: báo lỗi tại trường đầu tiên từ trên xuống khi người dùng chưa điền đầy đủ thông tin. Hiện thị thông báo “Please enter a correct username and password. Note that both fields may be case-sensitive!” khi tài khoản người dùng không tồn hoặc mật khẩu nhập không đúng</p> <p>- <b>Link “Đăng ký</b>: khi người dùng click vào link sẽ chuyển sang trang đăng ký.</p>
Xem trang cá nhân	<p>Người dùng click vào ảnh đại diện ở góc trên bên phải màn hình tại màn hình trang chủ, hiển thị màn hình trang cá nhân của người dùng bao gồm ảnh đại diện, ảnh bìa, Tên tài khoản, btn "Interacs", button “Rechange”, hiển thị các bài viết đã đăng(nếu có)</p>
Xem trang chủ	<p>Người dùng click vào logo "Divimenti" hoặc btn "New feed" ở góc trên bên trái màn hình, hiển thị màn hình trang chủ</p>
Đăng bài	<p>Người dùng có thể nhập liệu và chọn ảnh và click btn "Share" để đăng bài</p> <p>-<b>Caption</b>: nhập nội dung bài viết. Hiện thị mặc định: “What’s on your mind?”</p> <p>- <b>Chọn ảnh</b>: chọn 1 ảnh từ thiết bị</p> <p>- <b>Button “Đăng”</b>: đăng bài. Nếu không có dữ liệu nào thì không đăng được bài viết mới.</p>

	<p>Người dùng cũng có thể xóa bài viết đã đăng bằng cách click vào icon ba chấm ở góc trên bên phải bài viết và chọn “Delete”</p>
Bình luận	<p>Người dùng có thể bình luận/ xem bình luận trên những bài đăng của mình hoặc bạn bè</p> <ul style="list-style-type: none"> <li>- <b>Comment:</b> nhập nội dung bình luận.</li> <li>- <b>Button “Post”:</b> bình luận. Nếu không có ký tự nào thì không bình luận được</li> </ul>
Tìm kiếm bạn bè	<p>Hiển thị mặc định : Start typing to search...</p> <p>-<b>Textbox:</b> nhập tối đa 100 ký tự</p> <p>Thanh search hiển thị trên màn hình trang chủ, người dùng nhập liệu vào thanh search và nhấn Enter trên bàn phím để tìm kiếm.</p> <p>Sau khi tìm kiếm, nếu có tài khoản người dùng khớp với nội dung tìm kiếm thì hệ thống sẽ hiển thị lên màn hình "Search result for: + nội dung trên ô search" và danh sách người dùng.</p> <p>Nếu không nhập ký tự nào và Enter thì hiển thị tất cả người dùng</p>
Nhắn tin	<p>Tại màn hình trang cá nhân của người dùng khác, người dùng hover vào button "Interacs", màn hình hiển thị danh sách các actions để chọn bao gồm: Follow, Add friend, Become a fan, Send message. Người dùng chọn "Send Message" để nhắn tin.</p> <p>Trường hợp người dùng và người dùng khác đã có tin nhắn, click vào icon tin nhắn ở góc trên bên phải hoặc button</p>

	<p>"Chat" ở sidebar bên trái tại bất kì màn hình nào, sau đó chọn cuộc hội thoại</p> <p>Hiển thị mặc định: "Type your message"</p> <p>- <b>InputMessage</b>: nhập tin nhắn</p> <p>- <b>Button "Send"</b>: gửi tin nhắn.</p> <p>Nếu không có ký tự nào thì không gửi được</p>
Kết bạn	<p>Tại màn hình trang cá nhân của người dùng khác, người dùng hover vào button "Interacs", màn hình hiển thị danh sách các action để chọn. Người dùng chọn "Add friend", hệ thống sẽ gửi thông báo đến người dùng đó. Nếu người dùng đó chấp nhận kết bạn thì 2 người dùng trở thành bạn bè.</p>
Theo dõi	<p>Tại màn hình trang cá nhân của người dùng khác, người dùng hover vào button "Interacs", màn hình hiển thị danh sách các action để chọn. Người dùng chọn "Follow", action Follow chuyển thành Unfollow, hệ thống lưu mối quan hệ của 2 người dùng</p>
Trở thành fan	<p>Tại màn hình trang cá nhân của người dùng khác, người dùng hover vào button "Interacs", màn hình hiển thị danh sách các action để chọn. Người dùng chọn "Become a Fan", action chuyển thành Unfollow, hệ thống lưu mối quan hệ của 2 người dùng</p>

#### Về phía người quản trị:

*Bảng 3.2 Chức năng hệ thống về phía người quản trị*

Chức năng	Mô tả
Quản lý tài khoản người dùng	Người quản trị có thể thêm, sửa, xóa tài khoản người dùng
Quản lý bài viết	Người quản trị có thể thêm, sửa, xóa bài viết

Quản lý tin nhắn	Người quản trị có thể thêm, sửa, xóa tin nhắn
Quản lý bình luận	Người quản trị có thể thêm, sửa, xóa bình luận
Quản lý túi	Người quản trị có thể thêm, sửa, xóa túi
Quản lý thẻ nạp	Người quản trị có thể thêm, sửa, xóa thẻ nạp
Tất cả các chức năng của người dùng.	

### 3.1.3 Một số màn hình giao diện của website

- Giao diện màn hình Đăng ký:

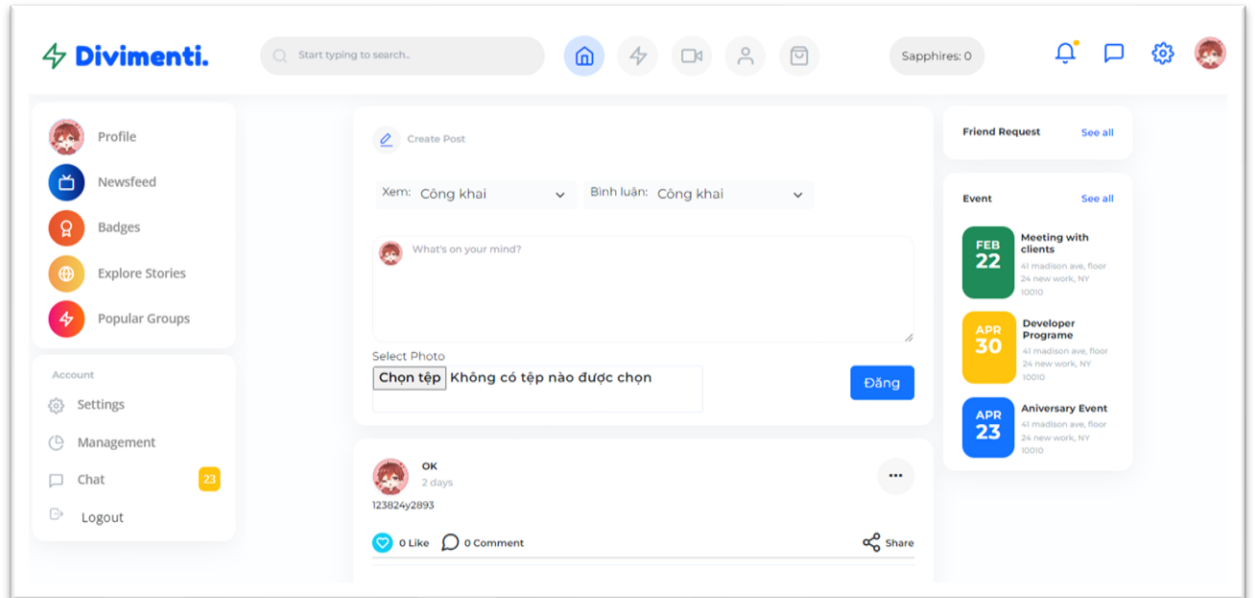
Hình 3.1 Giao diện màn hình Đăng ký

- Giao diện màn hình Đăng nhập

Click vào đây' (Forgot password? Click here) and 'Chưa có tài khoản? [Đăng ký](#)' (Don't have an account? Register)."/>

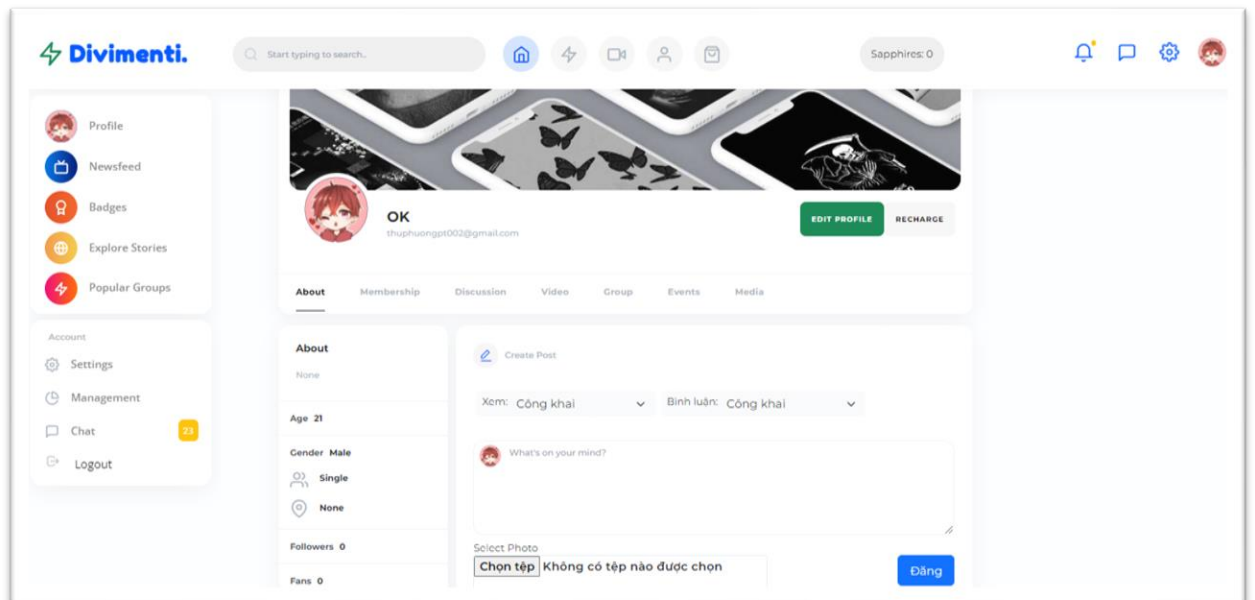
Hình 3.2 Giao diện màn hình Đăng nhập

- Giao diện màn hình trang chủ



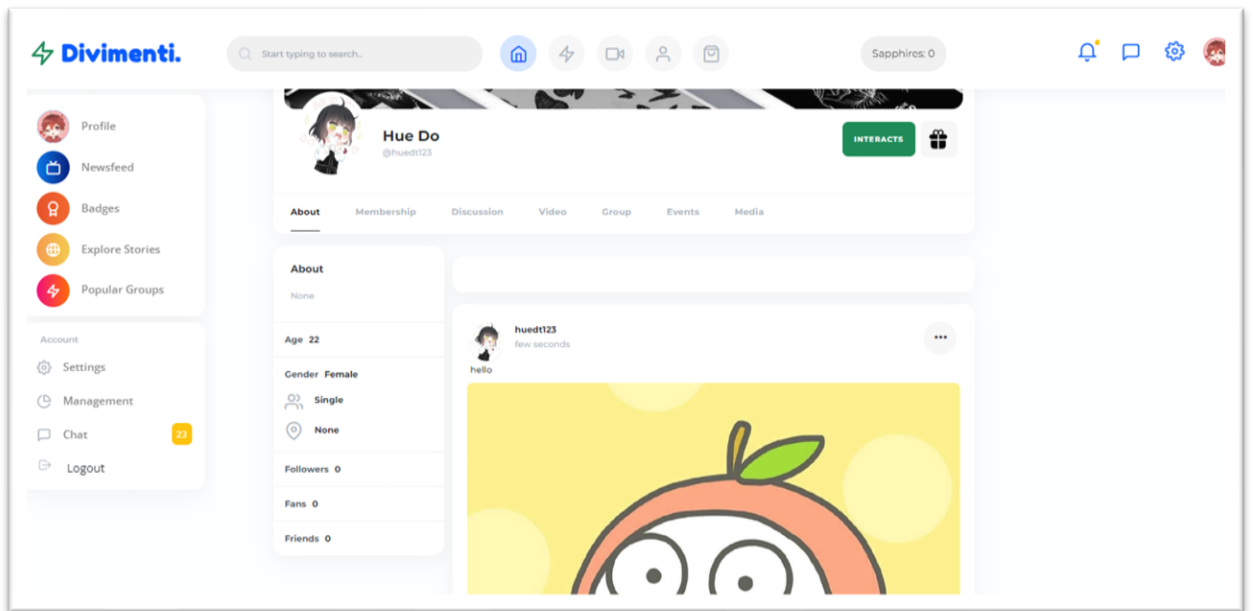
Hình 3.3 Giao diện màn hình Trang chủ

- Giao diện màn hình Trang cá nhân



Hình 3.4 Giao diện màn hình Trang cá nhân

- Giao diện màn hình trang cá nhân bạn bè



Hình 3.5 Giao diện màn hình Trang cá nhân bạn bè

### 3.2. Sự khác nhau giữa kịch bản kiểm thử thủ công và kịch bản tự động

- Đối với kịch bản kiểm thử thủ công thì việc kiểm thử thường được chia thành ba phần chính:

- Kiểm tra giao diện(UI)
- Kiểm tra validate
- Kiểm tra chức năng

- Còn đối với kịch bản kiểm thử tự động thì việc kiểm thử thường được chú trọng, đề ý tới là dữ liệu và trường hợp thử nghiệm:

- Trường hợp thử nghiệm: có thể ghi lại chuỗi sự kiện khi ta thao tác với phần mềm cần kiểm thử. Khác với khái niệm các trường hợp kiểm thử (Test case) khi thực hiện kiểm thử thủ công các giá trị đầu vào khác nhau thì sẽ thành một testcase.

### 3.3. Kịch bản kiểm thử thủ công trong website

Sử dụng kỹ thuật phân vùng tương đương, giá trị biên, đoán lỗi, kiểm thử chuyển trạng thái, bảng quyết định

### 3.3.1. Testcase kiểm thử chức năng Đăng ký

Module Code	Register				
Test requirement	Đăng ký thành công				
Tester	Đỗ Thị Huệ				
Result	Pass	Fail	Untested	N/A	Number of Test cases
	64	0	0	0	64

ID	Group Name	Sub Group Name	Test Case Description	Precondition	Test Case Procedure	Test data	Expected Output	Result
RE_001	Trigger	Link	Truy cập màn hình Đăng ký bằng link		1. Click vào link	<a href="http://127.0.0.1:8000/user/register">http://127.0.0.1:8000/user/register</a>	1. hiển thị trang Đăng ký	Pass
RE_002		Button	Truy cập màn hình Đăng ký bằng cách click button Đăng ký trên màn hình		1. Click button "Register" trên màn hình đăng nhập		1. hiển thị trang Đăng ký	Pass
RE_003	UI/Design		Kiểm tra số lượng các fields trên màn hình		1. Kiểm tra số lượng các fields		1. Số lượng các fields trên màn hình như sau: - text box: Tài khoản, Email, Số điện thoại, Mật khẩu, Tên, Họ - button: Đăng ký, Hiện - Link: Đăng nhập - Datepicker: Ngày sinh - Giới tính	Pass
RE_004			Kiểm tra vị trí, font chữ, size chữ, màu sắc		1. Kiểm tra vị trí, font chữ, size chữ, màu sắc		1. Hiện thị đúng theo design Design	Pass
RE_005			Kiểm tra trạng thái		1. Kiểm tra trạng thái các item		1. Các item đều ở đúng trạng thái enable/disable	Pass
RE_006	Validation		Kiểm tra giá trị default		1. Kiểm tra các giá trị default		1. Các text box đều hiển thị trắng, có tên trường: Tài khoản, Email, Mật khẩu, Số Điện Thoại, Tên, Họ	Pass
RE_007		All	Kiểm tra behavior của hệ thống khi bỏ trống toàn bộ thông tin		1. Click "Đăng ký"		1. Báo lỗi: "Vui lòng điền vào trường này."	Pass
RE_008		Tài khoản	Bỏ trống thông tin Tài khoản	Các trường khác được điền đầy đủ	1. Bỏ trống thông tin trường Tài khoản 2. Click "Đăng ký"		2. Báo lỗi: "Vui lòng điền vào trường này."	Pass
RE_009			Nhập 1 kí tự vào trường Tài khoản	Các trường khác được điền đầy đủ	1. Nhập 1 kí tự vào trường Tài khoản 2. Click "Đăng ký"	h	1. Hiện thị kí tự đã nhập trên textbox Tài khoản 2. Login thành công	Pass
RE_010			Nhập 10 kí tự vào trường Tài khoản	Các trường khác được điền đầy đủ	1. Nhập 10 kí tự vào trường Tài khoản 2. Click "Đăng ký"	1234567890	1. Hiện thị kí tự đã nhập trên textbox Tài khoản 2. Login thành công	Pass
RE_011			Nhập 100 kí tự vào trường Tài khoản	Các trường khác được điền đầy đủ	1. Nhập 100 kí tự vào trường Tài khoản 2. Click "Đăng ký"	sfgdgdff... (100 ký tự)	1. Hiện thị kí tự đã nhập trên textbox Tài khoản 2. Login thành công	Pass
RE_012			Nhập 101 kí tự vào trường Tài khoản	Các trường khác được điền đầy đủ	1. Nhập 101 kí tự vào trường Tài khoản 2. Click "Đăng ký"	sfgdgdff... (101 ký tự)	2. Không nhập được ký tự 101	Pass
RE_013		Email	Bỏ trống thông tin Email	Các trường khác được điền đầy đủ	1. Bỏ trống thông tin trường Email 2. Click "Đăng ký"		2. Báo lỗi: "Vui lòng điền vào trường này."	Pass
RE_054		Giới tính	Chọn Male	Các trường khác được điền đầy đủ	1. Chọn Male 2. Click "Đăng ký"		1. Hiện thị chọn Male 2. Đăng ký thành công	Pass
RE_055			Chọn Female	Các trường khác được điền đầy đủ	1. Chọn Female 2. Click "Đăng ký"		1. Hiện thị chọn Female 2. Đăng ký thành công	Pass
RE_056			Chọn Other	Các trường khác được điền đầy đủ	1. Chọn Other 2. Click "Đăng ký"		1. Hiện thị chọn Other 2. Đăng ký thành công	Pass
RE_057	Business	button "Hiện"	Click 1 lần khi trường Mật khẩu bị bỏ trống	Trường Mật khẩu bỏ trống	1. Click 1 lần button		1. Button "Hiện" đổi thành "Ẩn", textbox Mật khẩu hiển thị mặc định "Mật khẩu"	Pass
RE_060			Click 2 lần khi trường Mật khẩu có dữ liệu	Trường Mật khẩu có dữ liệu	1. Click lần 1 button 2. Click lần 2 button		1. Button "Hiện" đổi thành "Ẩn", textbox Mật khẩu hiển thị các ký tự đã nhập 2. Button "Ẩn" đổi thành "Hiện", textbox Mật khẩu hiển thị chấm tròn	Pass
RE_061		button "Đăng ký"	Kiểm tra behavior của Đăng ký btn khi nhập đầy đủ các thông tin hợp lệ		1. Nhập đầy đủ thông tin hợp lệ cho tất cả các trường 2. Click "Đăng ký"		2. Lưu thông tin user vào DB và Hiện thị trang "Đăng nhập"	Pass
RE_062			Kiểm tra behavior của Đăng ký btn khi nhập các thông tin không hợp lệ		1. Nhập thông tin không hợp lệ cho 1 trường bắt buộc bất kỳ 2. Click "Đăng ký"		2. Báo lỗi tại trường không hợp lệ, đăng ký không thành công	Pass
RE_063		Link "Đăng nhập"	Kiểm tra behavior của link "Đăng nhập" khi nhập đầy đủ các thông tin hợp lệ		1. Nhập đầy đủ thông tin hợp lệ cho tất cả các trường 2. Click "Đăng nhập"		2. Chuyển sang trang Đăng nhập	Pass
RE_064			Kiểm tra behavior của link "Đăng nhập" khi nhập các thông tin không hợp lệ		1. Nhập thông tin không hợp lệ cho 1 trường bắt buộc bất kỳ 2. Click "Đăng nhập"		2. Chuyển sang trang Đăng nhập	Pass

Hình 3.6 Trường hợp kiểm thử chức năng Đăng ký



### 3.3.2 Testcase kiểm thử chức năng Đăng nhập

Module Code	Login				
Test requirement	Đăng nhập thành công, chuyển đến trang chủ				
Tester	Đỗ Thị Huệ				
Result	Pass	Fail	Untested	N/A	Number of Test cases
	21	2	0	0	23

ID	Group Name	Sub Group Name	Test Case Description	Precondition	Test Case Procedure	Test data	Expected Output	Result
Login_001	Trigger	Link	Kiểm tra khi truy cập trang bằng link		1. Click link	<a href="http://127.0.0.1:8000/user/login">http://127.0.0.1:8000/user/login</a>	1. hiển thị trang đăng nhập	Pass
Login_002		Button	Kiểm tra khi click button "Login"		1. Click button "Login" trên màn hình đăng ký		1. hiển thị trang đăng nhập	Pass
Login_003	UI/Design		Kiểm tra số lượng các fields trên màn hình		1. Kiểm tra số lượng các fields		1. Số lượng các fields trên màn hình như sau: - text box: Tài khoản, Mật khẩu - button: Login - Link: Quên mật khẩu, Đăng ký	Pass
Login_004			Kiểm tra vị trí, font chữ, size chữ, màu sắc		1. Kiểm tra vị trí, font chữ, size chữ, màu sắc		1. Hiện thị đúng theo design <a href="#">Design</a>	Pass
Login_005			Kiểm tra trạng thái		1. Kiểm tra trạng thái các item		1. Các item đều ở đúng trạng thái enable/disable	Pass
Login_006			Kiểm tra giá trị default		1. Kiểm tra các giá trị default		1. Các text box đều hiển thị trắng, có tên trường Tài khoản, mật khẩu	Pass
Login_007	Validation	All	Kiểm tra behavior của hệ thống khi bỏ trống toàn bộ thông tin		1. Click "Đăng nhập"		1. Báo lỗi: "Vui lòng nhập vào trường này"	Pass
Login_008		Tài khoản	Bỏ trống thông tin Tài khoản		1. Bỏ trống thông tin trường Tài khoản		1. Báo lỗi: "Vui lòng nhập vào trường này"	Pass
Login_009			Nhập 1 kí tự vào trường Tài khoản		1. Nhập 1 ký tự vào trường Tài khoản	h	1. Hiện thị kí tự đã nhập trên textbox Tài khoản	Pass
Login_010			Nhập 10 kí tự vào trường Tài khoản		1. Nhập 10 ký tự vào trường Tài khoản	huehue1234	1. Hiện thị kí tự đã nhập trên textbox Tài khoản	Pass
Login_011			Nhập 100 kí tự vào trường Tài khoản		1. Nhập 100 ký tự vào trường Tài khoản	hue123.... (100 ký tự)	1. Hiện thị kí tự đã nhập trên textbox Tài khoản	Pass
Login_012			Nhập 101 kí tự vào trường Tài khoản		1. Nhập 101 ký tự vào trường Tài khoản	hue123.... (101 ký tự)	1. Không nhập được ký tự thứ 101	Pass
Login_013		Mật khẩu	Bỏ trống thông tin Mật khẩu		1. Bỏ trống thông tin trường Mật khẩu		2. Báo lỗi: "Vui lòng nhập vào trường này"	Pass
Login_014			Nhập 1 kí tự vào trường Mật khẩu		1. Nhập 1 kí tự vào trường Mật khẩu	h	1. Hiện thị 1 chấm tròn	Pass
Login_015			Nhập 5 kí tự vào trường Password		1. Nhập 5 kí tự vào trường Password 2. Click ra khỏi textbox Password	12345	1. Hiện thị 5 chấm tròn	Pass
Login_016			Nhập 8 kí tự vào trường Password		1. Nhập 8 kí tự vào trường Password 2. Click ra khỏi textbox Password	12345678	1. Hiện thị 8 chấm tròn	Pass
Login_017			Nhập 15 kí tự vào trường Password		1. Nhập 15 kí tự vào trường Password 2. Click ra khỏi textbox Password	huehue123456789	2. Hiện thị 15 chấm trên textbox	Pass
Login_018	Business	btn "Login"	Kiểm tra behavior của Đăng ký btn khi nhập tài khoản, mật khẩu hợp lệ		1. Nhập đầy đủ thông tin hợp lệ cho tất cả các trường 2. Click "Đăng nhập"		2. Đăng nhập thành công, chuyển đến trang chủ	Pass
Login_019			Kiểm tra behavior của Đăng nhập btn khi nhập Tài khoản hoặc mật khẩu không đúng trong DB		1. Nhập thông tin không hợp lệ cho 1 trường bắt buộc bất kỳ 2. Click "Đăng nhập"		2. Báo lỗi "Please enter a correct username and password. Note that both fields may be case-sensitive."	Pass
Login_020		link Đăng ký	Kiểm tra behavior của link "Đăng ký" khi nhập đầy đủ các thông tin hợp lệ		1. Nhập đầy đủ thông tin hợp lệ cho tất cả các trường 2. Click "Đăng ký"		2. Chuyển sang trang Đăng ký	Pass
Login_021			Kiểm tra behavior của link "Đăng ký" khi nhập các thông tin không hợp lệ		1. Nhập thông tin không hợp lệ cho 1 trường bắt buộc bất kỳ 2. Click "Đăng nhập"		2. Chuyển sang trang Đăng ký	Pass
Login_022		link Quên mật khẩu	Kiểm tra behavior của link "Đăng ký" khi nhập đầy đủ các thông tin hợp lệ		1. Nhập đầy đủ thông tin hợp lệ cho tất cả các trường 2. Click "Quên mật khẩu"		2. Hệ thống gửi mã OTP về số điện thoại đã đăng ký và chuyển sang trang nhập mã OTP	Fail
Login_023			Kiểm tra behavior của link "Đăng ký" khi nhập các thông tin không hợp lệ		1. Nhập thông tin không hợp lệ cho 1 trường bắt buộc bất kỳ 2. Click "Quên mật khẩu"		2. Hệ thống gửi mã OTP về số điện thoại đã đăng ký và chuyển sang trang nhập mã OTP	Fail

Hình 3.7 Trường hợp kiểm thử chức năng Đăng nhập

### 3.3.3. Testcase kiểm thử chức năng Đăng bài

Module Code	Post				
Test requirement	Đăng bài thành công				
Tester	Đỗ Thị Huệ				
Result	Pass	Fail	Untested	N/A	Number of Test cases
	28	0	0	0	28

ID	Group Name	Sub Group Name	Test Case Description	Precondition	Test Case Procedure	Test data	Expected Output	Result
POST_001	Trigger	Paste link	Truy cập màn hình bằng cách copy link trên cùng trình duyệt	Đã login vào hệ thống	1. Click link	<a href="http://127.0.0.1:8000/">http://127.0.0.1:8000/</a>	1. hiển thị trang chủ	Pass
POST_002			Truy cập vào màn hình bằng cách copy paste link khác trình duyệt	Đã login vào hệ thống	1. Click link	<a href="http://127.0.0.1:8000/">http://127.0.0.1:8000/</a>	1. hiển thị trang chủ	Pass
POST_003	UI/Design		Kiểm tra số lượng các fields trên màn hình		1. Kiểm tra số lượng các fields		1. Số lượng các fields trên màn hình như sau: - text box: caption - items: Chọn ảnh - button: Share, - Combobox: Chế độ xem, chế độ bình luận	Pass
POST_004			Kiểm tra vị trí, font chữ, size chữ, màu sắc		1. Kiểm tra vị trí, font chữ, size chữ, màu sắc		1. Hiển thị đúng theo design <a href="#">Design</a>	Pass
POST_005			Kiểm tra trạng thái		1. Kiểm tra trạng thái các item		1. Các item đều ở đúng trạng thái enable/disable	Pass
POST_006			Kiểm tra giá trị default		1. Kiểm tra các giá trị default		1. Textbox caption hiển thị "What's on your mind + <name>?" Chế độ xem "Công khai" Chế độ bình luận "Công khai"	Pass
POST_007	Validation		Bỏ trống các thông tin		1. Click "Đăng"		Không đăng bài được	Pass
POST_008			Bỏ trống textbox + chọn ảnh		1. Điền tắt cả các thông tin 2. Click "Đăng"		2. Hiện thị bài đăng mới ngay bên dưới Thời gian hiển thị "a few seconds ago" Nội dung là kí tự vừa nhập Ảnh là ảnh vừa chọn	Pass
POST_009			Nhập 1 kí tự vào textbox + không chọn ảnh		1. Nhập 1 kí tự vào textbox 2. Click "Đăng"	h	2. Hiện thị bài đăng mới ngay bên dưới Thời gian hiển thị "few seconds ago" Nội dung là kí tự vừa nhập	Pass
POST_015	Bussiness	Combo box "Chế độ xem"	Không đổi		1. Không đổi		1. Hiện thị "Công khai", tắt cả mọi người đều xem được bài viết	Pass
POST_016			Chọn "Người theo dõi"		1. Chọn "Người theo dõi"		1. Hiện thị "Người theo dõi", chỉ người theo dõi xem được bài viết	Pass
POST_017			Chọn "Người ủng hộ"		1. Chọn "Người ủng hộ"		1. Hiện thị "Người ủng hộ", chỉ người ủng hộ xem được bài viết	Pass
POST_018			Chọn "Bạn bè"		1. Chọn "Bạn bè"		1. Hiện thị "Bạn bè", chỉ bạn bè xem được bài viết	Pass
POST_019			Chọn "Chỉ mình tôi"		1. Chọn "Chỉ mình tôi"		1. Hiện thị "Chỉ mình tôi", chỉ chủ tài khoản xem được bài viết	Pass
POST_020			Chọn "Công khai"		1. Chọn "Công khai"		1. Hiện thị "Công khai", tắt cả mọi người đều xem được bài viết	Pass
POST_021		Combo box "Chế độ bình luận"	Không đổi		1. Không đổi		1. Hiện thị "Công khai", tắt cả mọi người đều bình luận được bài viết	Pass
POST_022			Chọn "Người theo dõi"		1. Chọn "Người theo dõi"		1. Hiện thị "Người theo dõi", chỉ người theo dõi bình luận được bài viết	Pass
POST_023			Chọn "Người ủng hộ"		1. Chọn "Người ủng hộ"		1. Hiện thị "Người ủng hộ", chỉ người ủng hộ bình luận được bài viết	Pass
POST_024			Chọn "Bạn bè"		1. Chọn "Bạn bè"		1. Hiện thị "Bạn bè", chỉ bạn bè bình luận được bài viết	Pass
POST_025			Chọn "Chỉ mình tôi"		1. Chọn "Chỉ mình tôi"		1. Hiện thị "Chỉ mình tôi", chỉ chủ tài khoản bình luận được bài viết	Pass
POST_026			Chọn "Công khai"		1. Chọn "Công khai"		1. Hiện thị "Công khai", tắt cả mọi người đều bình luận được bài viết	Pass
POST_027		button "Đăng"	Kiểm tra behavior của Đăng btn khi nhập caption hoặc ảnh		1. Nhập aption hoặc chọn ảnh 2. Click "Đăng"	dfgdgdgdge	2. Đăng bài thành công	Pass
POST_028			Kiểm tra behavior của Đăng btn khi không nhập caption và ảnh		1. Bỏ trống caption và ảnh 2. Click "Đăng"		Không đăng bài được	Pass

Hình 3.8 Trường hợp kiểm thử chức năng Đăng bài

### 3.3.4 Test case kiểm thử chức năng Tìm kiếm bạn bè

Module Code	Search				
Test requirement	Tìm kiếm bạn bè				
Tester	Đỗ Thị Huệ				
Result	Pass	Fail	Untested	N/A	Number of Test cases
	11	0	0	0	11

ID	Group Name	Sub Group Name	Test Case Description	Precondition	Test Case Procedure	Test data	Expected Output	Result
Search_001	Trigger	Paste link	Truy cập màn hình bằng cách copy link trên cùng trình duyệt	Đã login vào hệ thống	1. Click link	<a href="http://127.0.0.1:8000/">http://127.0.0.1:8000/</a>	1. hiển thị trang chủ	Pass
Search_002			Truy cập vào màn hình bằng cách copy paste link khác trình duyệt	Đã login vào hệ thống	1. Click link	<a href="http://127.0.0.1:8000/">http://127.0.0.1:8000/</a>	1. hiển thị trang chủ	Pass
Search_003	UI/Design		Kiểm tra số lượng các fields trên màn hình		1. Kiểm tra số lượng các fields		1. Số lượng các fields trên màn hình như sau: - text box: Searchbox - icon Search	Pass
Search_004			Kiểm tra vị trí, font chữ, size chữ, màu sắc		1. Kiểm tra vị trí, font chữ, size chữ, màu sắc		1. Hiển thị đúng theo design <a href="#">Design</a>	Pass
Search_005			Kiểm tra trạng thái		1. Kiểm tra trạng thái các item		1. Các item đều ở đúng trạng thái enable/disable	Pass
Search_006			Kiểm tra giá trị default		1. Kiểm tra các giá trị default		1. Textbox hiển thị "Start typing to search..."	Pass
Search_007	Validation	Search	Bỏ trống ô Search		1. Bỏ trống ô Search 2. Nhấn phím Enter		2. Hiện thị tất cả người dùng	Pass
Search_008			Nhập 1 kí tự vào ô Search và có kết quả tìm kiếm		1. Nhập 1 ký tự vào ô Search 2. Nhấn phím Enter	h	1. Hiện thị nội dung nhập trên ô Search 2. Hiện thị "Search result for: h" và danh sách người dùng	Pass
Search_009			Nhập 10 kí tự vào ô Search và có kết quả tìm kiếm		1. Nhập 10 ký tự vào ô Search 2. Nhấn phím Enter	huehue1234	1. Hiện thị nội dung nhập trên ô Search 2. Hiện thị "Search result for: huehue1234" và danh sách người dùng	Pass
Search_010			Nhập 100 kí tự vào ô Search và không có kết quả tìm kiếm		1. Nhập 100 ký tự vào ô Search 2. Nhấn phím Enter	hue123.... (100 ký tự)	1. Hiện thị nội dung nhập trên ô Search 2. Hiện thị "Search result for: hđghđghđ..." và danh sách người	Pass
Search_011			Nhập 101 kí tự vào ô Search		1. Nhập 101 ký tự vào ô Search	hue123.... (101 ký tự)	Không nhập dc ký tự 101	Pass

Hình 3.9 Trường hợp kiểm thử chức năng Tìm kiếm bạn bè

### 3.3.5 Test case chức năng Nhắn tin

Module Code	Send_Message				
Test requirement	Nhắn tin				
Tester	Đỗ Thị Huệ				
Result	Pass	Fail	Untested	N/A	Number of Test cases
	13	0	0	0	13

ID	Group Name	Sub Group Name	Test Case Description	Precondition	Test Case Procedure	Test data	Expected Output	Result
SM_001	Trigger	Paste link	Truy cập màn hình bằng cách copy link trên cùng trình duyệt	Đã login vào hệ thống	1. Click link	<a href="http://127.0.0.1:8000/chat/quanghuy/">http://127.0.0.1:8000/chat/quanghuy/</a>	1. hiển thị cuộc trò chuyện	Pass
SM_002			Truy cập vào màn hình bằng cách copy paste link khác trình duyệt	Đã login vào hệ thống	1. Click link	<a href="http://127.0.0.1:8000/chat/quanghuy/">http://127.0.0.1:8000/chat/quanghuy/</a>	1. hiển thị cuộc trò chuyện	Pass
SM_003		icon Chat	Truy cập màn hình bằng cách click icon Chat ở góc trên bên phải màn	Đã login vào hệ thống	1. Click icon		1. hiển thị cuộc trò chuyện	Pass
SM_004		button Chat	Truy cập màn hình bằng cách click button Chat ở sidebar bên trái màn hình	Đã login vào hệ thống	1. Click button		1. hiển thị cuộc trò chuyện	Pass
SM_005	UI/Design		Kiểm tra số lượng các fields trên màn hình		1. Kiểm tra số lượng các fields		1. Số lượng các fields trên màn hình như sau: - textbox: InputMessage - button: Send	Pass
SM_006			Kiểm tra vị trí, font chữ, size chữ, màu sắc		1. Kiểm tra vị trí, font chữ, size chữ, màu sắc		1. Hiển thị đúng theo design <a href="#">Design</a>	Pass
SM_007			Kiểm tra trạng thái		1. Kiểm tra trạng thái các item		1. Các item đều ở đúng trạng thái enable/disable	Pass
SM_008			Kiểm tra giá trị default		1. Kiểm tra các giá trị default		1. Textbox hiển thị "Type your message"	Pass

Hình 3.10 Trường hợp kiểm thử chức năng Nhắn tin

### 3.3.6 Trường hợp kiểm thử chức năng Bình luận

Module Code	Comment				
Test requirement	Bình luận				
Tester	Đỗ Thị Huệ				
Result	Pass	Fail	Untested	N/A	Number of Test cases
	12	0	0	0	12

ID	Group Name	Sub Group Name	Test Case Description	Precondition	Test Case Procedure	Test data	Expected Output	Result
CM_001	Trigger	Paste link	Truy cập màn hình bằng cách copy link trên cùng trình duyệt	Đã login vào hệ thống	1. Click link	<a href="http://127.0.0.1:8000">http://127.0.0.1:8000</a>	1. Hiện thị cuộc trò chuyện	Pass
CM_002			Truy cập vào màn hình bằng cách copy paste link khác trình duyệt	Đã login vào hệ thống	1. Click link	<a href="http://127.0.0.1:8001">http://127.0.0.1:8001</a>	1. Hiện thị cuộc trò chuyện	Pass
CM_005	UI/Design		Kiểm tra số lượng các fields trên màn hình		1. Kiểm tra số lượng các fields		1. Số lượng các fields trên màn hình như sau: - textbox: InputComment - button: Post	Pass
CM_006			Kiểm tra vị trí, font chữ, size chữ, màu sắc		1. Kiểm tra vị trí, font chữ, size chữ, màu sắc		1. Hiện thị đúng theo design Design	Pass
CM_007			Kiểm tra trạng thái		1. Kiểm tra trạng thái các item		1. Các item đều ở đúng trạng thái enable/disable	Pass
CM_008			Kiểm tra giá trị default		1. Kiểm tra các giá trị default		1. Textbox hiển thị trắng	Pass
CM_009	Validation	InputComment	Bỏ trống textbox		1. Bỏ trống textbox 2. Click "Post"		2. Không bình luận được	Pass
CM_010			Nhập 1 ký tự vào textbox		1. Nhập 1 ký tự vào textbox 2. Click "Post"	a	1. Hiện thị các ký tự đã nhập trên box 2. Bình luận hiển thị với nội dung như đã nhập	Pass
CM_011			Nhập 10 ký tự vào textbox		1. Nhập 10 ký tự vào textbox 2. Click "Post"	1234567890	1. Hiện thị các ký tự đã nhập trên box 2. Bình luận hiển thị với nội dung như đã nhập	Pass
CM_012			Nhập 100 ký tự vào textbox		1. Nhập 100 ký tự vào textbox 2. Click "Post"	sdsdsfsff.. (100 ký tự)	1. Hiện thị các ký tự đã nhập trên box 2. Bình luận hiển thị với nội dung như đã nhập	Pass
CM_013	Bussiness	button "Post"	Kiểm tra behavior của Đăng btn khi nhập caption hoặc ảnh		1. Nhập nội dung bình luận 2. Click "Post"	dfgdgdgdge	2. Bình luận thành công	Pass
CM_014			Kiểm tra behavior của Đăng btn khi không nhập caption và ảnh		1. Bỏ trống nội dung bình luận 2. Click "Post"		Không bình luận được	Pass

Hình 3.11 Trường hợp kiểm thử chức năng Bình luận

### 3.3.7 Báo cáo kết quả kiểm thử

TEST REPORT						
Project Name	Diviment					
Creator	Đỗ Thị Huệ					
No	Module code	Pass	Fail	Untested	N/A	Number of test cases
1	Đăng ký	64	0	0	0	64
2	Đăng nhập	21	2	0	0	23
3	Đăng bài	28	0	0	0	28
4	Tìm kiếm bạn bè	11	0	0	0	11
5	Nhắn tin	13	0	0	0	13
6	Bình luận	12	0	0	0	12
Sub total		85	2	0	0	87
Test coverage			100,00 %			
Test successful coverage			97,70 %			

Hình 3.12 Test report

### 3.4 Kiểm thử tự động các chức năng

Sử dụng Selenium WebDriver áp dụng test một số chức năng trên website

#### 3.4.1. Kiểm thử tự động chức năng Đăng ký

Thực hiện kiểm thử chức năng Đăng ký và cho ra kết quả

```
[chrome 124.0.6367.208 windows #0-0] » \test\specs\TC_Register.js
[chrome 124.0.6367.208 windows #0-0] Register an account
[chrome 124.0.6367.208 windows #0-0] ✓ Hiển thị mặc định trường tài khoản
[chrome 124.0.6367.208 windows #0-0] ✓ Hiển thị mặc định trường Email
[chrome 124.0.6367.208 windows #0-0] ✓ Hiển thị mặc định trường số điện thoại
[chrome 124.0.6367.208 windows #0-0] ✓ Hiển thị mặc định trường mật khẩu
[chrome 124.0.6367.208 windows #0-0] ✓ Hiển thị mặc định trường tên
[chrome 124.0.6367.208 windows #0-0] ✓ Hiển thị mặc định trường họ
[chrome 124.0.6367.208 windows #0-0] ✓ Hiển thị mặc định trường Ngày sinh
[chrome 124.0.6367.208 windows #0-0] ✓ Button Đăng ký có thể click được
[chrome 124.0.6367.208 windows #0-0] ✓ Nhập 1 ký tự vào Email- không chứa @
[chrome 124.0.6367.208 windows #0-0] ✓ Nhập 9 ký tự vào Email- không chứa @
[chrome 124.0.6367.208 windows #0-0] ✓ Nhập 254 ký tự vào Email- không chứa @
[chrome 124.0.6367.208 windows #0-0] ✓ Bỏ trống số điện thoại điền các trường khác
[chrome 124.0.6367.208 windows #0-0] ✓ Nhập chữ vào Số điện thoại
[chrome 124.0.6367.208 windows #0-0] ✓ Nhập ký tự đặc biệt vào Số điện thoại
[chrome 124.0.6367.208 windows #0-0] ✓ Nhập chữ hoa vào Mật khẩu
[chrome 124.0.6367.208 windows #0-0] ✓ Nhập chữ thường vào Mật khẩu
[chrome 124.0.6367.208 windows #0-0] ✓ Nhập số vào Mật khẩu
[chrome 124.0.6367.208 windows #0-0] ✓ Nhập ký tự đặc biệt vào Mật khẩu
[chrome 124.0.6367.208 windows #0-0] ✓ Nhập 7 ký tự đủ loại vào Mật khẩu
[chrome 124.0.6367.208 windows #0-0] ✓ Nhập >= 8 ký tự thiếu chữ hoa vào Mật khẩu
[chrome 124.0.6367.208 windows #0-0] ✓ Nhập >= 8 ký tự thiếu chữ thường vào Mật khẩu
[chrome 124.0.6367.208 windows #0-0] ✓ Nhập >= 8 ký tự thiếu số vào Mật khẩu
[chrome 124.0.6367.208 windows #0-0] ✓ Nhập >=8 ký tự thiếu ký tự đặc biệt vào Mật khẩu
[chrome 124.0.6367.208 windows #0-0] ✓ Bỏ trống ngày sinh
[chrome 124.0.6367.208 windows #0-0] ✓ Nhập Ngày sinh không hợp lệ
[chrome 124.0.6367.208 windows #0-0] ✓ Đăng ký thành công
[chrome 124.0.6367.208 windows #0-0] ✓ Click link Đăng nhập - Điền đúng đủ thông tin
[chrome 124.0.6367.208 windows #0-0] ✓ Click link Đăng nhập - Không điền thông tin
[chrome 124.0.6367.208 windows #0-0] 28 passing (52.4s)
```

Hình 3.13 Kết quả kiểm thử tự động chức năng Đăng ký

#### 3.4.2. Kiểm thử tự động chức năng Đăng nhập

Thực hiện kiểm thử chức năng Đăng nhập và cho ra kết quả kiểm thử

```
Login application
✓ Hiển thị mặc định trường Tài khoản
✓ Hiển thị mặc định trường Mật khẩu
✓ Button Đăng nhập có thể click được
✓ Username không chính xác
✓ Username chính xác - Password không chính xác
✓ Password không chính xác
✓ Password chính xác chữ thường
✓ Login OK
✓ Click link Đăng ký - Điền đúng đủ thông tin
✓ Click link Đăng ký - Không điền thông tin
✗ Click link Quên mật khẩu - Điền đúng đủ thông tin
✗ Click link Quên mật khẩu - Không điền thông tin

10 passing (48.6s)
2 failing
```

Hình 3.14 Kết quả kiểm thử tự động chức năng Đăng nhập

### 3.4.3 Kiểm thử tự động chức năng Đăng bài

Thực hiện kiểm thử chức năng Đăng bài và cho ra kết quả kiểm thử

```
"spec" Reporter:
-----
[chrome 124.0.6367.208 windows #0-0] Running: chrome (v124.0.6367.208) on windows
[chrome 124.0.6367.208 windows #0-0] Session ID: 86203ceea0f2732a423bd92482eccaa8
[chrome 124.0.6367.208 windows #0-0]
[chrome 124.0.6367.208 windows #0-0] » \test\specs\TC_Post.js
[chrome 124.0.6367.208 windows #0-0] Post
[chrome 124.0.6367.208 windows #0-0]     ✓ Hiển thị mặc định
[chrome 124.0.6367.208 windows #0-0]     ✓ Post 1 ký tự
[chrome 124.0.6367.208 windows #0-0]     ✓ Post 10 ký tự
[chrome 124.0.6367.208 windows #0-0]     ✓ Post 200 ký tự
[chrome 124.0.6367.208 windows #0-0]     ✓ Post ảnh
[chrome 124.0.6367.208 windows #0-0]     ✓ Post 1 ký tự + ảnh
[chrome 124.0.6367.208 windows #0-0]     ✓ Post 10 ký tự + ảnh
[chrome 124.0.6367.208 windows #0-0]     ✓ Post 1 ký tự + ảnh
[chrome 124.0.6367.208 windows #0-0]
[chrome 124.0.6367.208 windows #0-0] 8 passing (50.7s)

Spec Files:      1 passed, 1 total (100% completed) in 00:00:55
```

Hình 3.15 Kết quả kiểm thử tự động chức năng Đăng bài

### 3.4.4 Kiểm thử tự động chức năng Nhắn tin

Thực hiện kiểm thử tự động chức năng Nhắn tin và cho ra kết quả kiểm thử

```
"spec" Reporter:
-----
[chrome 124.0.6367.208 windows #0-0] Running: chrome (v124.0.6367.208) on windows
[chrome 124.0.6367.208 windows #0-0] Session ID: aff7f82576c5f3512d142669f0947d56
[chrome 124.0.6367.208 windows #0-0]
[chrome 124.0.6367.208 windows #0-0] » \test\specs\TC_Chat.js
[chrome 124.0.6367.208 windows #0-0] Nhắn tin cho bạn bè
[chrome 124.0.6367.208 windows #0-0]     ✓ Hiển thị mặc định
[chrome 124.0.6367.208 windows #0-0]     ✓ Nhập 1 ký tự vào Message box
[chrome 124.0.6367.208 windows #0-0]     ✓ Nhập 100 ký tự vào Message box
[chrome 124.0.6367.208 windows #0-0]     ✓ Gửi bằng phím Enter
[chrome 124.0.6367.208 windows #0-0]
[chrome 124.0.6367.208 windows #0-0] 4 passing (8.7s)

Spec Files:      1 passed, 1 total (100% completed) in 00:00:12
```

Hình 3.16 Kết quả kiểm thử tự động chức năng Nhắn tin



### 3.4.5 Kiểm thử tự động chức năng Tìm kiếm

Thực hiện kiểm thử và cho ra kết quả kiểm thử

```
"spec" Reporter:
-----
[chrome 124.0.6367.208 windows #0-0] Running: chrome (v124.0.6367.208) on windows
[chrome 124.0.6367.208 windows #0-0] Session ID: b7a84a139aa2d122fe4461124e1a29f8
[chrome 124.0.6367.208 windows #0-0]
[chrome 124.0.6367.208 windows #0-0] » \test\specs\TC_Search.js
[chrome 124.0.6367.208 windows #0-0]   Tìm kiếm bạn bè
[chrome 124.0.6367.208 windows #0-0]     ✓ Bỏ trống ô search
[chrome 124.0.6367.208 windows #0-0]     ✓ Nhập 1 ký tự vào ô search ô search
[chrome 124.0.6367.208 windows #0-0]     ✓ Nhập 10 ký tự vào ô search ô search
[chrome 124.0.6367.208 windows #0-0]     ✓ Nhập 100 ký tự vào ô search
[chrome 124.0.6367.208 windows #0-0]
[chrome 124.0.6367.208 windows #0-0] 4 passing (12.2s)

Spec Files:      1 passed, 1 total (100% completed) in 00:00:17
```

Hình 3.17 Kết quả kiểm thử tự động chức năng Tìm kiếm bạn bè

### 3.4.6 Kiểm thử tự động chức năng Bình luận

Thực hiện kiểm thử tự động chức năng Bình luận và cho ra kết quả kiểm thử

```
"spec" Reporter:
-----
[chrome 124.0.6367.208 windows #0-0] Running: chrome (v124.0.6367.208) on windows
[chrome 124.0.6367.208 windows #0-0] Session ID: 15dd7a2dd8941ba26170b8cd5d9182d1
[chrome 124.0.6367.208 windows #0-0]
[chrome 124.0.6367.208 windows #0-0] » \test\specs\TC_Comment.js
[chrome 124.0.6367.208 windows #0-0]   Post
[chrome 124.0.6367.208 windows #0-0]     ✓ Hiện thị mặc định
[chrome 124.0.6367.208 windows #0-0]     ✓ Bình luận 1 ký tự
[chrome 124.0.6367.208 windows #0-0]     ✓ Bình luận 10 ký tự
[chrome 124.0.6367.208 windows #0-0]     ✓ Bình luận 100 ký tự
[chrome 124.0.6367.208 windows #0-0]
[chrome 124.0.6367.208 windows #0-0] 4 passing (17.7s)

Spec Files:      1 passed, 1 total (100% completed) in 00:00:22
```

Hình 3.18 Kết quả kiểm thử tự động chức năng Bình luận

## KẾT LUẬN

### Kết luận:

Sau thời gian tìm hiểu và ứng dụng kiểm thử tự động em nhận thấy những vấn đề đã đạt được:

- Kiểm thử phần mềm rất quan trọng trong việc đảm bảo chất lượng của phần mềm. Việc nghiên cứu lựa chọn các kỹ thuật và chiến lược kiểm thử nhằm giúp việc thực hiện kiểm thử hiệu quả, giảm chi phí và thời gian.
- Ứng dụng thành công để truyền dữ liệu đầu vào cho kiểm thử tự động chức năng đăng ký, thêm mới sinh viên trên nền tảng Web. Tạo được bản kế hoạch kiểm thử (Testcases) và chạy được bản kế hoạch.
- Có thêm định hướng về ngành nghề kiểm thử phần mềm.

### Hạn chế:

Tuy nhiên, thời gian có hạn và kinh nghiệm thực tế của bản thân em chưa nhiều nên đề tài không tránh khỏi những thiếu sót:

- Chưa nghiên cứu lập trình nâng cao với Selenium
- Chưa kiểm thử được về mặt giao diện, hiệu năng,...
- Chưa kiểm thử những chức năng có nhiều các loại control sử dụng nền tảng web.

### Hướng phát triển đề tài:

- Những nhược điểm trên cũng là hướng phát triển tương lai của đồ án.
- Có thêm định hướng về ngành nghề kiểm thử phần mềm.

Trong thời gian tới em sẽ tiếp tục và nghiên cứu sâu hơn về kiểm thử để kiểm thử tự động sử dụng Selenium hỗ trợ vào công việc tương lai sau này.



## **TÀI LIỆU THAM KHẢO**

- [1] Hoàng Quang Huy (2016), Giáo trình kiểm thử phần mềm, Nhà xuất bản Thống kê, Hà Nội
- [2] Mocha – the fun, simple, flexible JavaScript test framework, <https://mochajs.org/>
- [3] Selenium, [selenium.dev](https://selenium.dev)
- [4] WebdriverIO - Next-gen browser and mobile automation test framework for Node.js, <http://webdriver.io/>
- [5] Kiểm thử tự động là gì, <https://hocjavascript.net/kiem-thu/>