

**TRƯỜNG ĐẠI HỌC KIẾN TRÚC HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**



Ảnh có chứa biểu tượng, Phông chữ, Đồ họa, Xanh điện

Mô tả được tạo tự động

**BÀI TẬP LỚN**

**Môn học: Kiểm thử và đảm bảo chất lượng phần mềm**

**Đề tài: Kiểm Thử Qua Bảng Quyết Định, Sử dụng phần mềm Katalon Kiểm thử Tính Năng của website Facebook**

Giảng viên hướng dẫn: **Ts. Nguyễn Đăng Khoa**

Nhóm thực hiện: **Nhóm 1,** Lớp: **20CN5**

Thành viên:

**Hoàng Kông Minh – 2055010168**

**Trịnh Đình Đại - 2055010054,**

**Bùi Đức Năng – 2055010186,**

**Nguyễn Thị Ngọc – 2055010191,**

**Tống Thị Vân – 2055010287**

HÀ NỘI, THÁNG 03 NĂM 2024

**MỤC LỤC**

[**MỞ ĐẦU** 3](#_Toc162731816)

[**CHƯƠNG 1: TỔNG QUAN VỀ KIỂM THỬ CHỨC NĂNG** **PHẦN MỀM** 4](#_Toc162731817)

[**1.1.  Các khái niệm kiểm thử, kiểm thử chức năng phần mềm** 4](#_Toc162731818)

[1.1.1. Kiểm thử phần mềm là gì? 4](#_Toc162731819)

[1.1.2. Kiểm thử chức năng phần mềm là gì? 4](#_Toc162731820)

[**1.2. Quy trình kiểm thử chức năng** 4](#_Toc162731821)

[1.2.1. Các quan điểm kiểm thử chức năng 4](#_Toc162731822)

[1.2.2. Các bước tiến hành 5](#_Toc162731823)

[1.2.3. Các cấp độ kiểm thử 5](#_Toc162731824)

[1.2.4. Một số kĩ thuật kiểm thử chức năng 6](#_Toc162731825)

[1.2.5. So sánh kiểm thử chức năng và kiểm thử phi chức năng 8](#_Toc162731826)

[**1.3. Kết chương** 9](#_Toc162731827)

[**CHƯƠNG 2: TÌM HIỂU KIỂM THỬ BẢNG TÍNH QUYẾT ĐỊNH** 10](#_Toc162731828)

[**2.1. Kiểm thử bảng tính quyết định** 10](#_Toc162731829)

[**2.2. Cách dùng bảng quyết định** 10](#_Toc162731830)

[**2.3. Ví dụ về bảng quyết định** 11](#_Toc162731831)

[**CHƯƠNG 3: TÌM HIỂU KATALON STUDIO** 15](#_Toc162731832)

[**3.1. Tổng quan về KATALON STUDIO** 15](#_Toc162731833)

[**3.2. Các tính năng của KATALON STUDIO** 16](#_Toc162731834)

[**3.3. Quy trình làm việc của KATALON STUDIO** 17](#_Toc162731835)

[3.3.1. Khởi tạo dự án test (INITIATE) 17](#_Toc162731836)

[3.3.2. Tạo nội dung test (CREATE) 17](#_Toc162731837)

[3.3.3. Vận hành - chạy (OPERATE) 17](#_Toc162731838)

[3.3.4. Báo cáo (REPORT) 18](#_Toc162731839)

[3.3.5. Bảo trì (MAINTAIN) 18](#_Toc162731840)

[**3.4. Kết chương** 18](#_Toc162731841)

[**CHƯƠNG 4: SỬ DỤNG KATALON STUDIO KIỂM THỬ BẢNG TÍNH QUYẾT ĐỊNH MẠNG XÃ HỘI FACEBOOK** 19](#_Toc162731842)

[**4.1. Tổng quan phần mềm sử dụng để kiểm thử** 19](#_Toc162731843)

[4.1.1. Kiểm thử trang đăng nhập 19](#_Toc162731844)

[4.1.2. Kiểm thử Upload avatar facebook 19](#_Toc162731845)

[**4.2. Sử dụng Katalon Studio kiểm thử phần mềm** 19](#_Toc162731846)

[4.2.1. Kiểm thử bảng tính quyết định với chức năng đăng nhập 19](#_Toc162731847)

[4.2.2. Kiểm thử bảng tính quyết định với chức năng Upload avatar 23](#_Toc162731848)

[**TỔNG KẾT** 29](#_Toc162731849)

[**TÀI LIỆU THAM KHẢO** 29](#_Toc162731850)

**MỞ ĐẦU**

**1. Tổng quan tình hình nghiên cứu thuộc lĩnh vực của đề tài**

Ngày nay, công nghệ thông tin nói chung và công nghệ phần mềm nói riêng đang chiếm một vị trí quan trọng trong tiến trình công nghiệp hoá, hiện đại hoá đất nước. Cùng với việc phát triển công nghệ phần mềm luôn tiềm ẩn những thách thức cho dành các doanh nghiệp, nhà phát triển phần mềm trong việc kiểm chất lượng phần mềm, đặc biệt là vấn đề soát lỗi. Một phần mềm chứa lỗi sau khi phát hành sẽ gây thiệt hại vô cùng lớn không chỉ cho doanh nghiệp phát triển phần mềm, mà tác hại hơn là thiệt hại cho người sử dụng thậm chí cho cả xã hội. Chính vì vậy việc kiểm soát, phát hiện lỗi phần mềm luôn được đặt song hành với việc tạo ra phần mềm mới. Lỗi càng sớm phát hiện càng giảm thiểu thiệt hại về cả chi phí nguồn nhân lực, vật lực và thời gian. Kiểm thử là một quá trình liên tục, xuyên suốt mọi giai đoạn phát triển phần mềm để đảm bảo rằng phần mềm thỏa mãn yêu cầu thiết kế và yêu cầu đó đáp ứng được nhu cầu của người sử dụng. Các kỹ thuật kiểm thử phần mềm đang được nghiên cứu và việc kiểm thử phần mềm trở thành quy trình bắt buộc trong các dự án phát triển phần mềm trên thế giới. Xu hướng áp dụng tự động hoá đang được triển khai rộng rãi ở nhiều lĩnh vực, trong đó có kiểm thử phần mềm. Đặc biệt, khi kiểm thử phần mềm là công đoạn chiếm phần lớn thời gian trong quá trình phát triển dự án phần mềm thì sự ra đời của các công cụ kiểm thử tự động càng có ý nghĩa hơn bao giờ hết, giúp tiết kiệm thời gian, công sức và tiền bạc.

**2. Tính cấp thiết, ý nghĩa khoa học và thực tiễn của đề tài**

Từ những phân tích trên kết hợp với những kiến thức đã học trong môn kiểm thử phần mềm nhóm chúng em sẽ trình bày bài thu hoạch của mình dựa trên việc kiểm thử tự động trên trang web “Học trực tuyến cùng OLM” chúng em đang xây dựng. Công cụ để kiểm thử tự động được chúng em sử dụng là Katalon Studio. Đây là công cụ kiểm thử tự động được đánh giá cao cả về chất lượng lẫn tốc độ. Điều đó chúng em sẽ trình bày cụ thể ở các phần bên dưới qua đề tài:

*“****Kiểm thử bảng quyết định mạng xã hội Facebook qua phần mềm Katalon****”*

# **CHƯƠNG 1: TỔNG QUAN VỀ KIỂM THỬ CHỨC NĂNG** **PHẦN MỀM**

## **1.1.  Các khái niệm kiểm thử, kiểm thử chức năng phần mềm**

### **1.1.1. Kiểm thử phần mềm là gì?**

Kiểm thử phần mềm[1] là quá trình thực thi 1 chương trình với mục đích tìm

ra lỗi.

Kiểm thử phần mềm đảm bảo sản phẩm phần mềm đáp ứng chính xác, đầy đủ và đúng theo yêu cầu của khách hàng, yêu cầu của sản phẩm đề 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.

Kiểm thử phần mềm tạo điều kiện cho bạn tận dụng tối đa tư duy đánh giá và sáng tạo để bạn có thể phát hiện ra những điểm mà người khác chưa nhìn thấy.

### **1.1.2. Kiểm thử chức năng phần mềm là gì?**

Kiểm thử chức năng[1] là một loại kiểm thử hộp đen (black box) và các trường hợp kiểm thử của nó được dựa trên đặc tả của ứng dụng phần mềm/thành phần đang test. Các chức năng được test bằng cách nhập vào các giá trị nhập và kiểm tra kết quả đầu ra, và ít quan tâm đến cấu trúc bên trong của ứng dụng (không giống như kiểm thử hộp trắng - white-box testing).

Có thể hiểu 1 cách đơn giản, kiểm thử chức năng là xác nhận tất cả các chức năng của hệ thống. Nó đánh giá ứng dụng và xác nhận liệu ứng dụng có đang hoạt động theo yêu cầu hay không.

Trong kiểm thử phần mềm có nhiều quy trình, mỗi cái sẽ có một nhiệm vụ khác nhau. Tại đây kiểm thử chức năng sẽ đảm nhận nhiệm vụ kiểm tra xem các chức năng của hệ thống, chương trình có đảm nhận hoản chỉnh nhiệm vụ ở đặc tả không khi đưa các giá trị đầu vào. Việc kiểm thử chức năng đem lại khá nhiều lợi ích, chẳng hạn như tránh được việc kiểm thử dư thừa với các chức năng không cần thiết, hay ngăn chặn sự đa dạng lỗi tại cùng một thời điểm.

## **1.2. Quy trình kiểm thử chức năng**

### **1.2.1. Các quan điểm kiểm thử chức năng**

Kiểm thử chức năng[1]  có thể thực hiện theo 2 quan điểm:

**Requiments - based:** sử dụng các đặc tả yêu cầu của hệ thống làm cơ sở để thiết kế test. Một cách tốt để bắt đầu là sử dụng bảng nội dung của đặc tả yêu cầu như một danh sách các mục kiểm thử và không kiểm thử. Chúng ta nên xét độ ưu tiên của yêu cầu dựa trên các tiêu chí rủi ro và sử dụng độ ưu tiên để kiểm thử. Điều này sẽ đảm bảo những phần quan trọng nhất sẽ được kiểm thử.

**Business – process – based:** Sử dụng các kiến thức về quy trình nghiệp vụ. Quy trình nghiệp vụ mô tả các kịch bản liên quan đến nghiệp vụ hằng ngày của hệ thống.

Các use case bắt nguồn từ hướng đối tượng, nhưng ngày nay phổ biến nhiều là dựa trên vòng đời phát triển phần mềm. chúng ta có thể lấy quy trình nghiệp vụ là điểm bắt đầu. Các use case là một cơ sở rất hữu ích tạo ra các trường hợp kiểm thử từ góc nhìn về nghiệp vụ.

### **1.2.2. Các bước tiến hành**

Thông thường kiểm thử chức năng sẽ được tiến hành theo 6 bước sau[1]:

* **Bước 1:** Xác định các chức năng mà phần mềm dự kiến sẽ làm (dực vào phần đặc tả của phần mềm).
* **Bước 2:** Xác định bộ dữ liệu đầu vào dựa trên các thông số kỹ thuật của chức năng.
* **Bước 3:** Xác định bộ dữ liệu đầu ra dựa trên các thông số kỹ thuật của chức năng.
* **Bước 4:** Viết và thực thi các trường hợp kiểm thử (Test Case).
* **Bước 5:** So sánh kết quả đàu ra chuẩn bị ở bước 3 và kết quả thực tế.
* **Bước 6:** Dựa vào nhu cầu của khách hàng để đánh giá xem kết quả ở bước 5 có phù hợp hay không?

Bước 1: xác định các chức năng mà phần mềm dự kiến sẽ làm (dựa vào phần đặc tả  của phần mềm).

Bước 2: xác định bộ dữ liệu đầu vào dựa trên các thông số kỹ thuật của

### **1.2.3. Các cấp độ kiểm thử**

**Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động**

*Hình 1.1: Các cấp độ kiểm thử*

Giống như những loại kiểm thử khác, chúng ta sẽ đi từ đơn giản cho đến

phức tạp:

* Đầu tiên và quan trọng nhất, chúng ta cần hiểu được luồng dữ liệu hoạt

động, các quy trình vận hành của sản phẩm phần mềm.

* Sau đó chúng ta cần xác định được các nghiệp vụ sử dụng của nó.
* Chúng ta sẽ kiểm thử các chức năng ở từng màn hình riêng biệt, chẳng

hạn như màn hình đăng nhập, màn hình profile…

* Sau khi kiểm thử từng màn hình chúng ra sẽ kiểm thử một module chứa

nhiều màn hình cùng nhóm.

* Tiếp đến là kiểm thử một vòng hoàn chỉnh của nghiệp vụ và kiểm tra tất

cả các vòng.

* Cuối cùng chúng ta sẽ giả định như mình là người dùng thực tế, định ra

các kịch bản đặc biệt và tiến hành kiểm tra nó.

### **1.2.4. Một số kĩ thuật kiểm thử chức năng**

* Kiểm thử điều hướng người dùng[1]  (user navigation testing):

Trong kỹ thuật này, chúng ta sẽ cần kiểm tra một số thành phần, các liên kết

giữa màn hình này và màn hình khác để đảm bảo cho sự hoạt động liên tục của

nghiệp vụ đang thực hiện. Có thể kể đến một số thành phần như:

* Hệ thống đăng nhập, đăng xuất
* Hệ thống thanh điều hướng.
* Hệ thống thanh công cụ (toolbar).
* Hệ thống cây phân cấp chức năng.

Tại kiểm thử này, khả năng điều hướng sẽ tập trung trên 2 vấn đề:

Người dùng đăng nhập vào hệ thống với một số quyền hạn nhất định.

Người dùng thao tác qua các tính năng một cách tự nhiên không bị đứt

quãng và sau đó sẽ đăng xuất.

* Kiểm thử thao tác trên màn hình[1]  (transaction screen testing)

Thông thường kiểm thử thao tác sẽ thực hiện ở các form nhập dữ liệu, hoặc

một số các nút chức năng, các lựa chọn. Có 2 loại thường thấy:

* Kết quả sẽ được thực hiện ở một trang khác.
* Kết quả sẽ được hiển thị ngay trang hiện tại.

**Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động**

*Hình 1.2: Kiểm thử các thao tác*

Với mỗi loại ô nhập dữ liệu, các nút chức năng, các lựa chọn, người kiểm thử cần xác định các trường hợp thao tác trên màn hình theo các yêu cầu đặc tả, tài liệu người dùng hay tài liệu quản trị viên tương ứng.

Nếu kết quả hiển thị ngay trên màn hình thao tác hiện tại thì ta có thể dựa vào bộ dữ liệu đầu vào và đầu ra dự kiến để kiểm tra nó bằng phương pháp để kiểm tra hộp đen. Tương tự nếu kết quả được hiển thị ở trang khác nếu là click vào các nút.

Công việc của người kiểm thử:

* Thiết kế test case để xác thực hoạt động của mỗi file dữ liệu, list,

option và button trên màn hình thao tác theo các yêu cầu nghiệp vụ,

tài liệu người dùng và tài liệu người quản trị.

* Nếu kết quả được hiển thị trên màn hình thao tác, thì kỹ thuật kiểm

thử hộp đen với test case gồm (data input, output kỳ vọng) sẽ được

dùng để xác thực kết quả hiển thị.

* Kiểm thử luồng dữ liệu[1]  (transaction flow testing)

Với kỹ thuật này, chúng ta cần thực hiện liền mạch một số thao tác qua nhiều bước với nhiều màn hình khác nhau để đánh giá xem có phù hợp với luồng nghiệp vụ không.

Chúng ta cần xác định 2 trường hợp:

* Nếu người dùng thao tác đúng quy trình sẽ cho ra đúng như đặc tả yêu
* cầu.
* Nếu người dùng thao tác không đúng dù chỉ một bước thì sẽ không ra
* được kết quả.
* Kiểm thử màn hình báo cáo[1]  (report screen testing).

Không như màn hình thao tác ở phần kiểm thử thao tác trên màn hình, màn hình báo cáo sẽ không yêu cầu phải nhập dữ liệu. ở đây chúng ta cần kiểm tra cách hiển thị hay tìm kiếm dữ liệu. Có rất nhiều cách người dùng có thể đặc tả các dữ liệu cần tìm (như sử dụng lọc thông tin hay nhập các điều kiện tìm kiếm) hoặc cách mà dữ liệu được hiển thị (danh sách, bảng biểu, biểu đồ,…)

Chúng ta cần căn cứ vào đặc tả của ứng dụng để phân tích các trường hợp kiểm thử, có thể có một số trường hợp như lựa chọn giá trị không phù hợp cho bộ lọc (VD: năm tìm kiếm lớn hơn năm hiện tại) hay dữ liệu không được hiển thị đúng.

* Kiểm thử luồng báo cáo[1]  (report flow testing).

Kiểm thử các khác biệt giữa kết quả hiển thị trong quá trình report và các phương thức report khác (như: máy in, file,…). Nhiệm vụ của người kiểm thử

* Xác định xem phần mềm gửi cùng kết quả ra màn hình report và máy in?
* Xác thực kết quả report trên tất cả các phương thức báo cáo khác nhau được hỗ trợ bởi phần mềm,
* Xác định xem khả năng máy in có hỗ trợ font, vùng chọn được người dùng xác định trong màn hình report?

### **1.2.5. So sánh kiểm thử chức năng và kiểm thử phi chức năng**

|  |  |
| --- | --- |
| **Kiểm thử chức năng** | **Kiểm thử phi chức năng** |
| Được thực hiện bằng cách sử dụng đặc tả chức năng do khách hàng cung cấp và xác minh hệ thống theo các yêu cầu chức năng | Dùng đề kiểm tra hiệu năng, độ tin cậy, khả năng mở rộng và các nghiệp vụ phi chức năng khác của hệ thống |
| Được thực hiện đầu tiên | Nên được thực hiện ngay sau khi kiểm thử chức năng kết thúc |
| Có thể sử dụng tool để kiểm thử hoặc có thể kiểm thử bằng tay | Sử dụng tool để kiểm thử sẽ chính xác hơn |
| Đầu vào là các yêu cầu nghiệp vụ theo các tài liệu đặc tả | Đầu vào là các thông số như tốc độ tải, độ bảo mật cũng như khả năng mở rộng |
| Mô tả những thứ sản phẩm phần mềm sẽ làm | Mô tả sản phẩm phần mềm hoạt động như thế nào là tốt |
| Dễ dàng thực hiện kiểm thử bằng tay | Khó khăn khi kiểm thử bằng tay |
| Các loại kiểm thử chức năng  thường gặp:   * Unit testing (kiểm thử đơn vị). * Smoke Testing * Sanity Testing * Integration Testing (kiểm thử tích hợp) * White Box Testing (kiểm thử hộp trắng) * Black Box Testing (kiểm thử hộp đen) * User Acceptance Testing (kiểm thử chấp nhận) * Regression Testing (kiểm thử hồi quy) | Các loại kiểm thử phi chức năng  thường gặp:   * Performance Testing (kiểm thử hiệu năng) * Load Testing (kiểm thử tải) * Volume Testing (kiểm thử khối lượng) * Stress Testing * Security Testing (kiểm thử bảo   mật)   * Installation Testing (kiểm thử cài đặt) * Penetration Testing (kiểm thử   xâm nhập)   * Compatibility Testing (kiểm thử tương thích) * Migration Testing |

*Bảng 1.1: So sánh kiểm thử các chức năng và phi chức năng*

## **1.3. Kết chương**

Trong chương này nhóm chúng em đã trình bày khái quát nhất về các khái niệm kiểm thử, kiểm thử chức năng, quy trình, mức độ kiểm thử chức năng. Trong chương tiếp theo chúng em sẽ giới thiệu cụ thể về công cụ sẽ dùng để kiểm thử tự động trong bài tập lớn này.

# **CHƯƠNG 2: TÌM HIỂU KIỂM THỬ BẢNG TÍNH QUYẾT ĐỊNH**

## **2.1. Kiểm thử bảng tính quyết định**

Bảng tính quyết định là một kỹ thuật Test được sử dụng để kiểm tra các hành vi hệ thống (system behavior) với các cách kết hợp input đầu vào khác nhau. Đây là một cách tiếp cận có hệ thống, kết quả của các kết hợp đó là hành vi hệ thống tương ứng của chúng (output) sẽ được ghi lại dưới dạng bảng. Đó cũng là lý do tại sao bảng quyết định còn gọi là “ Nguyên nhân – Kết quả”.

Số Case xảy ra là 2^n (n là số điều kiện đầu vào).

Một Decision Table thường có cấu trúc như bên dưới.

Ảnh có chứa văn bản, biểu đồ, Phông chữ, hàng

Mô tả được tạo tự động

*Hình 2.1 Cấu trúc Decision Table*

## **2.2. Cách dùng bảng quyết định**

Bảng quyết định bao gồm các cột đại diện cho các điều kiện và hành động kết hợp, và các hàng đại diện cho các quy tắc quyết định. Mỗi ô trong bảng quyết định đại diện cho một trạng thái kết quả dự kiến (có thể là True/False, Đúng/Sai, Yes/No,… hoặc giá trị cụ thể khác).

Ảnh có chứa văn bản, ảnh chụp màn hình, số, Phông chữ

Mô tả được tạo tự động

*Hình 2.2 Ví dụ về bảng quyết định*

Có thể thấy rằng, chúng ta tạo một bảng trong đó các hàng trên cùng là các điều kiện đầu vào(Input/Condition), các hàng dưới cùng là các hành động kết quả (Output/Action).

Thứ đề bài đưa cho ta giải quyết ta biến thành Input, còn thứ đề bài mong muốn là Output.

## **2.3. Ví dụ về bảng quyết định**

**Ví dụ 1: Form đăng nhập**

**Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình, thiết kế

Mô tả được tạo tự động**

*Hình 2.3 Ví dụ Form đăng nhập*

Với các yêu cầu của Form đăng nhập:

* + Người dùng nhập đúng email và mật khẩu khi đăng nhập thành công sẽ dược điều hướng sang trang chủ của website.
  + Nếu nhập email hoặc mật khẩu không đúng khi đăng nhập hệ thống sẽ hiển thị lỗi thông báo tương ứng.

Ở đây có 2 điều kiện đầu vào. Vậy ở đây ta sẽ có số Case xảy ra là 2^2 = 4.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Điều kiện | Quy tắc 1 | Quy tắc 2 | Quy tắc 3 | Quy tắc 4 |
| Email ( T/F) | T | T | F | F |
| Mật khẩu (T/F) | T | F | T | F |
| Kết quả (E/H) | H | E | E | E |

*Bảng 2.1 Bảng Case ví dụ đăng nhập*

**Chú thích:**

T - True: Nhập đúng email và mật khẩu.

F - False: Email hoặc mật khẩu bị sai.

E - Error: Hiển thị lỗi.

H - Home: Hiển thị trang chủ.

Diễn giải:

Trường hợp 1: Email và mật khẩu đúng, người dùng sẽ được chuyển hướng đến trang chủ.

Trường hợp 2: Email đúng, mật khẩu sai; người dùng sẽ nhận được thông báo lỗi.

Trường hợp 3: Email sai, mật khẩu đúng; người dùng sẽ nhận được thông báo lỗi.

Trường hợp 4: Email và mật khẩu sai, người dùng sẽ nhận được thông báo lỗi.

Khi chuyển các trường hợp trên thành testcase, chúng ta có thể tạo ra hai kịch bản,

Một là điền email đúng - mật khẩu đúng -> click nút đăng nhập => Kết quả mong muốn là hệ thống sẽ chuyển hướng người dùng đến trang chủ.

Khi chuyển các trường hợp trên thành testcase, chúng ta có thể tạo ra hai kịch bản:

**Kịch bản 1:**

Là điền email đúng - mật khẩu đúng -> click nút đăng nhập => Kết quả mong muốn là hệ thống sẽ chuyển hướng người dùng đến trang chủ.

**Kịch bản 2 :**

Nhập email sai - mật khẩu sai -> click nút đăng nhập => Kết quả mong muốn là hệ thống sẽ hiển thị lỗi cho người dùng.

Nhập email đúng - mật khẩu sai -> click nút đăng nhập => Kết quả mong muốn là hệ thống sẽ hiển thị lỗi cho người dùng.

Nhập email sai - mật khẩu đúng -> click nút đăng nhập => Kết quả mong muốn là hệ thống sẽ hiển thị lỗi cho người dùng.

**Test Case: Kịch bản 1**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TT** | **Tên Kiểm Thử** | **ĐK Kiểm Thử (Đầu Vào)** | **Kiểm Thử Cái Gì (Thao Tác Kiểm Thử)** | **Đầu Ra** | **Pass/Fail** |
| 1 | Đăng nhập thành công | Email: correct@example.com | Nhập email và mật khẩu đúng | Chuyển hướng đến trang chủ | Pass |
| 2 | Đăng nhập không thành công | Email: incorrect@example.com | Nhập email đúng và mật khẩu sai | Thông báo lỗi đăng nhập không thành công | K Pass |
| 3 | Đăng nhập không thành công | Email: correct@example.com, Mật khẩu: | Nhập email đúng và không nhập mật khẩu | Thông báo lỗi đăng nhập không thành công | K Pass |
| 4 | Đăng nhập không thành công | Email: , Mật khẩu: correctpassword | Không nhập email và nhập mật khẩu đúng | Thông báo lỗi đăng nhập không thành công | K |

*Bảng 2.2 Bảng Case ví dụ đăng nhập kịch bản* *1*

**Test Case: Kịch bản 2:** Nhập email sai – Mật khẩu sai -> Click nút đăng nhập => Kết quả mong muốn là hệ thống sẽ hiển thị lỗi cho người dùng.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên kiểm thử** | **Điều kiện kiểm thử (đầu vào)** | **Kiểm thử cái gì (thao tác kiểm thử)** | **Đầu ra** | **Pass/Fail** |
| 1 | Nhập email sai - mật khẩu sai | Email không tồn tại | Nhập email sai, mật khẩu sai và click nút đăng nhập | Hệ thống hiển thị thông báo lỗi "Email hoặc mật khẩu không chính xác." | Pass |
| 2 | Nhập email sai - mật khẩu sai | Email đúng, mật khẩu sai | Nhập email đúng, mật khẩu sai và click nút đăng nhập | Hệ thống hiển thị thông báo lỗi "Mật khẩu không chính xác." | Pass |
| 3 | Nhập email sai - mật khẩu sai | Email và mật khẩu đều sai | Nhập email và mật khẩu đều sai và click nút đăng nhập | Hệ thống hiển thị thông báo lỗi "Email hoặc mật khẩu không chính xác." | Pass |
| 4 | Nhập email sai - mật khẩu sai | Email và mật khẩu đều rỗng | Nhập email và mật khẩu đều rỗng và click nút đăng nhập | Hệ thống hiển thị thông báo lỗi "Vui lòng nhập email và mật khẩu." | Pass |
| 5 | Nhập email sai - mật khẩu sai | Email sai định dạng | Nhập email sai định dạng (ví dụ: abc@xyz) và mật khẩu và click nút đăng nhập | Hệ thống hiển thị thông báo lỗi "Email không hợp lệ." | Pass |

*Bảng 2.3 Bảng Case ví dụ đăng nhập kịch bản 2*

Nhập email đúng - mật khẩu sai -> click nút đăng nhập => Kết quả mong muốn là hệ thống sẽ hiển thị lỗi cho người dùng.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên kiểm thử** | **Điều kiện kiểm thử (đầu vào)** | **Kiểm thử cái gì (thao tác kiểm thử)** | **Đầu ra** | **Pass/Fail** |
| 1 | Nhập email đúng - mật khẩu sai | Email đúng, mật khẩu sai | Nhập email đúng, mật khẩu sai và click nút đăng nhập | Hệ thống hiển thị thông báo lỗi "Mật khẩu không chính xác." | Pass |

Nhập email sai - mật khẩu đúng -> click nút đăng nhập => Kết quả mong muốn là hệ thống sẽ hiển thị lỗi cho người dùng.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên kiểm thử** | **Điều kiện kiểm thử (đầu vào)** | **Kiểm thử cái gì (thao tác kiểm thử)** | **Đầu ra** | **Pass/Fail** |
| 1 | Nhập email sai - mật khẩu đúng | Email không tồn tại | Nhập email sai, mật khẩu đúng và click nút đăng nhập | Hệ thống hiển thị thông báo lỗi "Email hoặc mật khẩu không chính xác." | Pass |

**Ví dụ 2:** Một web khóa học có chương trình khuyến mãi như sau:

* + Khách mới giảm 20%,
  + Khách cũ giảm 10%,
  + Khách có mã KM giảm 25%.
* Ở đây có 3 điều kiện. Nên sẽ có tổng 2^3 = 8 trường hợp:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Điều kiện | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Khách mời | T | T | T | T | F | F | F | F |
| Khách cũ | T | T | F | F | T | T | F | F |
| Khác có mã khuyến mại | T | F | T | F | T | F | T | F |
| Kết quả | X | X | 45% | 20% | 35% | 10% | X | X |

*Bảng 2.4 Bảng Case ví dụ web khóa học*

# **CHƯƠNG 3: TÌM HIỂU KATALON STUDIO**

## **3.1. Tổng quan về KATALON STUDIO**

**KATALON STUDIO[2]** là một bộ công cụ toàn diện cho kiểm thử tự động

hóa ứng dụng trên web và điện thoại di động. Công cụ này bao gồm một gói đầy đủ các tính năng mạnh mẽ giúp vượt qua những thách thức phổ biến trong tự động hóa thử nghiệm giao diện web, ví dụ như pop-up, iFrame và wait-time. Giải pháp thân thiện và linh hoạt này giúp tester thực hiện công tác kiểm tra tốt hơn, làm việc nhanh hơn và khởi chạy phần mềm chất lượng cao nhờ vào sự thông minh mà nó cung cấp cho toàn bộ quá trình tự động hóa kiểm thử.

Công cụ này có sẵn miễn phí và có thể tải xuống và cài đặt dễ dàng.

**Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

Mô tả được tạo tự động**

*Hình 3.1: Giao diện Katalon Studio*

Một số công cụ chính trong Katalon Studio:

|  |  |
| --- | --- |
| 1. Nhóm công cụ thực hiện khởi tạo, ghi lại trường hợp kiểm thử mới | |
|  | Khởi tạo, ghi lại trường hợp kiểm thử trên web |
|  | Khởi tạo, ghi lại trường hợp kiểm thử trên mobile |
|  | Khởi tạo, ghi lại trường hợp kiểm thử trên window |
| 2. Nhóm vận hành | |
|  | Build CMD |
|  | Chạy trường hợp kiểm thử |
|  | Sửa lỗi trong trường hợp kiểm thử |
|  | Tạm dừng chạy kiểm thử |
| 3. Nhóm xem và tùy chỉnh nhật ký thực thi kiểm thử | |
|  | Hiển thị trạng thái các bước thực thi trong quá trình vận hành kiểm thử |
|  | Hiển thị chi tiết các bước bắt đầu, kết thúc, thời gian như thế nào |
| 4. Nhóm quản lý các mục trong dự án | |
|  | Danh sách các trường hợp kiểm thử |
|  | Danh sách các nút, đối tượng sử dụng kiểm thử |
|  | Tập hợp các trường hợp kiểm thử với dữ liệu tương ứng |
|  | Danh sách dữ liệu dùng để kiểm thử |

*Bảng 3.1: Công cụ thường dùng trong Katalon Studio*

## **3.2. Các tính năng của KATALON STUDIO**

* **Triển khai đơn giản[2] (Simple deployment):** gói triển khai duy nhất, gắn kết chứa mọi thứ cần để triển khai một công cụ kiểm thử tự động mạnh mẽ.
* Cài đặt **nhanh chóng và dễ dàng[2] (Quick & easy set-up):** không chỉ cung cấp cài đặt đơn giản, Katalon Studio còn giúp tester dễ dàng thiết lập môi trường.
* Người kiểm thử có thể chạy kịch bản kiểm thử đầu tiên của họ khá nhanh chóng bằng cách sử dụng các mẫu và tập lệnh kiểm thử dựng sẵn của nó.

**Kết quả nhanh hơn và tốt hơn[2] (Faster & Better results):** các mẫu dựng sẵn với hướng dẫn rõ ràng giúp người kiểm tra nhanh chóng xây dựng và chạy các kịch bản kiểm thử tự động. Có thể thực hiện từng bước một tốc độ và hiệu quả, từ thiết lập dự án, tạo kiểm thử, thực hiện, tạo báo cáo và bảo trì.

**Chế độ linh hoạt (Flexible modes) [2]:** có thể sử dụng bản ghi và từ khóa để xây dựng bài kiểm thử tự động, trong khi có IDE đầy đủ để xây dựng tập lệnh nâng cao.

**Dễ sử dụng (Ease of use) [2]:** ngay cả thủ công với kinh nghiệm lập trình tối thiểu cũng có thể khai thác lợi ích của nó một cách dễ dàng.

**Ứng dụng đa trình duyệt (Cross-browser application) [2]:** Katalon Studio hỗ trợ nhiều nền tảng: Windows 32 và 64 (7, 8, và 10) và OS X 10.5+

## **3.3. Quy trình làm việc của KATALON STUDIO**

**Katalon Studio[2]** là một giải pháp tự động hóa thân thiện với người dùng, được đặc trưng bởi sự đơn giản và tốc độ. Nó rất hữu ích cho các đội thử nghiệm tự động và cá nhân để dành ít nhất những nỗ lực từ việc thiết lập một dự án mới để thực hiện các kiểm tra và sau đó giám sát kết quả thực hiện. Mỗi luồng công việc cung cấp rất nhiều khả năng và tùy chỉnh cho việc bảo trì dễ dàng và quy mô dự án:

Quy trình làm việc với Katalon Studio:

### **3.3.1. Khởi tạo dự án test (INITIATE)**

Các mẫu dự án tích hợp[2]  bằng cách cung cấp các mẫu dựng sẵn để tổ chức các trường hợp kiểm thử, kho lưu trữ đối tượng và từ khóa, Katalon Studio giúp việc kiểm thử dễ dàng hơn cho người kiểm tra.

**Nhiều khả năng:** hỗ trợ đầy đủ kiểm thử cho Web, Android, iOS, và API trên tất cả các hệ điều hành.

**Tích hợp trên công cụ không rắc rối**: dễ dàng kết hợp với Jenkins, GIT và JIRA với các plug-in gốc

### **3.3.2. Tạo nội dung test (CREATE)**

**Tạo kiểm thử tự động[2]:** ghi lại các hành động và tạo các kịch bản tự động bằng các từ khóa tích hợp.

**Tập lệnh Hi-end:** cho phép xây dựng kịch bản kiểm thử nâng cao hoặc từ khóa tùy chỉnh dễ dàng và hiệu quả.

**Thu thập thông tin đối tượng:** máy ghi âm tiên tiến phát hiện các thuộc tính đối tượng một cách hiệu quả để tối đa hóa nhận dạng.

### **3.3.3. Vận hành - chạy (OPERATE)**

**Thực hiện kiểm thử[2]  (Test execution) mạnh mẽ:** chạy các trường hợp kiểm thử hoặc bộ kiểm thử bằng cách sử dụng nhiều cấu hình và bộ dữ liệu.

**Tính linh hoạt trong thực thi:** cung cấp bảng điều khiển tích hợp CI với các tham số khác nhau để thực hiện từ xa. Chạy thử nghiệm trên nhiều trình duyệt và hệ điều hành OS cục bộ hoặc với Sauce Labs và BrowserStack.

**Xử lý lỗi linh hoạt và thực hiện lại tự động:** bao gồm các quy tắc thời gian chạy để tự động xử lý các luồng thực thi phức tạp.

### **3.3.4. Báo cáo (REPORT)**

**Báo cáo có sẵn[2]  trong một số định dạng:** với ghi nhật kí nâng cao, dữ liệu gỡ lỗi và ảnh chụp màn hình.

**Báo cáo thực hiện Bespoke:** được tích hợp với quy trình thông báo của bạn.

**Nhật ký Selenium và Appium cải tiến:** với các tính năng phân tích được cải tiến để cải thiện chiến lược tự động hóa.

### **3.3.5. Bảo trì (MAINTAIN)**

**Bảo trì đối tượng kiểm thử[2]  thông minh:** tự động cập nhật tất cả các trường hợp và bộ kiểm thử liên quan khi các đối tượng được thay đổi.

**Tổ chức kiểm thử hiệu quả:** cho phép dễ dàng quản lý và bảo trì các bài kiểm thử, dữ liệu và từ khóa.

## **3.4. Kết chương**

Trong chương này chúng em đã giới thiệu tổng quan về công cụ kiểm thử tự động mà nhóm đã dùng để làm bài tập lớn đó là Katalon Studio. Đây là công cụ tương đối dễ dùng, cho kết quả nhanh chóng. Trong chương tiếp theo nhóm sẽ trình bày cụ thể từng công đoạn kiểm thử các chức năng với công cụ Katalon Studio này như thế nào.

# **CHƯƠNG 4: SỬ DỤNG KATALON STUDIO KIỂM THỬ BẢNG TÍNH QUYẾT ĐỊNH MẠNG XÃ HỘI FACEBOOK**

## **4.1. Tổng quan phần mềm sử dụng để kiểm thử**

Phần mềm được sử dụng trong quá trình kiểm thử của nhóm 1 l kiểm thử bảng tính quyết đinh với các chức năng trên facebook.

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, phần mềm

Mô tả được tạo tự động

*Hình 4.1 Trang đăng nhập Facebook*

### **4.1.1. Kiểm thử trang đăng nhập**

Nhóm sẽ triển khai kiểm thử việc đăng nhập vào hệ thống từ trang đăng nhập. Tức là khi nhập đúng mật khẩu, tên đăng nhập thì người dùng đến Trang chủ. Ngược lại người dùng sẽ không vào được trang chủ và sẽ có thông báo liên quan.

### **4.1.2. Kiểm thử Upload avatar facebook**

Nhóm sẽ triển khai việc thay đổi ảnh facebook.

## **4.2. Sử dụng Katalon Studio kiểm thử phần mềm**

### **4.2.1. Kiểm thử bảng tính quyết định với chức năng đăng nhập**

Với các yêu cầu của Form đăng nhập:

* + Người dùng nhập đúng email và mật khẩu khi đăng nhập thành công sẽ được điều hướng sang trang chủ của website.
  + Nếu nhập email hoặc mật khẩu không đúng khi đăng nhập hệ thống sẽ hiển thị thông báo lỗi tương ứng

Ở đây có 2 điều kiện đầu vào. Vậy ở đây ta sẽ có số Case xảy ra là 2^2 = 4:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Điều kiện | Quy tắc 1 | Quy tắc 2 | Quy tắc 3 | Quy tắc 4 |
| Email ( T/F) | T | T | F | F |
| Mật khẩu (T/F) | T | F | T | F |
| Kết quả (E/H) | H | E | E | E |

*Bảng 4.1 Bảng Testcase đăng nhập*

**Chú thích:**

T - True: Nhập đúng email và mật khẩu.

F - False: Email hoặc mật khẩu bị sai.

E - Error: Hiển thị lỗi.

H - Home: Hiển thị trang chủ.

**Diễn giải:**

**Trường hợp 1:** Email và mật khẩu đúng, người dùng sẽ được chuyển hướng đến trang chủ.

**Trường hợp** **2:** Email đúng, mật khẩu sai; người dùng sẽ nhận được thông báo lỗi.

**Trường hợp 3:** Email sai, mật khẩu đúng; người dùng sẽ nhận được thông báo lỗi.

**Trường hợp 4:** Email và mật khẩu sai, người dùng sẽ nhận được thông báo lỗi.

Khi chuyển các trường hợp trên thành testcase, chúng ta có thể tạo ra hai kịch bản,

Một là điền email đúng - mật khẩu đúng -> click nút đăng nhập => Kết quả mong muốn là hệ thống sẽ chuyển hướng người dùng đến trang chủ.

Khi chuyển các trường hợp trên thành testcase, chúng ta có thể tạo ra hai kịch bản:

**Kịch bản 1:**

Là điền email đúng - mật khẩu đúng -> click nút đăng nhập => Kết quả mong muốn là hệ thống sẽ chuyển hướng người dùng đến trang chủ.

**Kịch bản 2 :**

Nhập email sai - mật khẩu sai -> click nút đăng nhập => Kết quả mong muốn là hệ thống sẽ hiển thị lỗi cho người dùng.

Nhập email đúng - mật khẩu sai -> click nút đăng nhập => Kết quả mong muốn là hệ thống sẽ hiển thị lỗi cho người dùng.

Nhập email sai - mật khẩu đúng -> click nút đăng nhập => Kết quả mong muốn là hệ thống sẽ hiển thị lỗi cho người dùng.

**Test Case: Kịch bản 1**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TT** | **Tên Kiểm Thử** | **ĐK Kiểm Thử (Đầu Vào)** | **Kiểm Thử Cái Gì (Thao Tác Kiểm Thử)** | **Đầu Ra** | **Pass/K Pass** |
| 1 | Đăng nhập thành công | Email: correct@example.com | Nhập email và mật khẩu đúng | Chuyển hướng đến trang chủ | Pass |
| 2 | Đăng nhập không thành công | Email: incorrect@example.com | Nhập email đúng và mật khẩu sai | Thông báo lỗi đăng nhập không thành công | K Pass |
| 3 | Đăng nhập không thành công | Email: correct@example.com, Mật khẩu: | Nhập email đúng và không nhập mật khẩu | Thông báo lỗi đăng nhập không thành công | K Pass |
| 4 | Đăng nhập không thành công | Email: , Mật khẩu: correctpassword | Không nhập email và nhập mật khẩu đúng | Thông báo lỗi đăng nhập không thành công | K |

*Bảng 4.2 Bảng TestCase đăng nhập kịch bản1*

**Test case kịch bản 2:** Nhập email sai - mật khẩu sai -> click nút đăng nhập => Kết quả mong muốn là hệ thống sẽ hiển thị lỗi cho người dùng.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên kiểm thử** | **Điều kiện kiểm thử (đầu vào)** | **Kiểm thử cái gì (thao tác kiểm thử)** | **Đầu ra** | **Pass/Fail** |
| 1 | Nhập email sai - mật khẩu sai | Email không tồn tại | Nhập email sai, mật khẩu sai và click nút đăng nhập | Hệ thống hiển thị thông báo lỗi "Email hoặc mật khẩu không chính xác." | Pass |
| 2 | Nhập email sai - mật khẩu sai | Email đúng, mật khẩu sai | Nhập email đúng, mật khẩu sai và click nút đăng nhập | Hệ thống hiển thị thông báo lỗi "Mật khẩu không chính xác." | Pass |
| 3 | Nhập email sai - mật khẩu sai | Email và mật khẩu đều sai | Nhập email và mật khẩu đều sai và click nút đăng nhập | Hệ thống hiển thị thông báo lỗi "Email hoặc mật khẩu không chính xác." | Pass |
| 4 | Nhập email sai - mật khẩu sai | Email và mật khẩu đều rỗng | Nhập email và mật khẩu đều rỗng và click nút đăng nhập | Hệ thống hiển thị thông báo lỗi "Vui lòng nhập email và mật khẩu." | Pass |
| 5 | Nhập email sai - mật khẩu sai | Email sai định dạng | Nhập email sai định dạng (ví dụ: abc@xyz) và mật khẩu và click nút đăng nhập | Hệ thống hiển thị thông báo lỗi "Email không hợp lệ." | Pass |

*Bảng 4.3 Bảng Case ví dụ đăng nhập kịch bản 2*

Nhập email đúng - mật khẩu sai -> click nút đăng nhập => Kết quả mong muốn là hệ thống sẽ hiển thị lỗi cho người dùng.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên kiểm thử** | **Điều kiện kiểm thử (đầu vào)** | **Kiểm thử cái gì (thao tác kiểm thử)** | **Đầu ra** | **Pass/Fail** |
| 1 | Nhập email đúng - mật khẩu sai | Email đúng, mật khẩu sai | Nhập email đúng, mật khẩu sai và click nút đăng nhập | Hệ thống hiển thị thông báo lỗi "Mật khẩu không chính xác." | Pass |

Nhập email sai - mật khẩu đúng -> click nút đăng nhập => Kết quả mong muốn là hệ thống sẽ hiển thị lỗi cho người dùng.

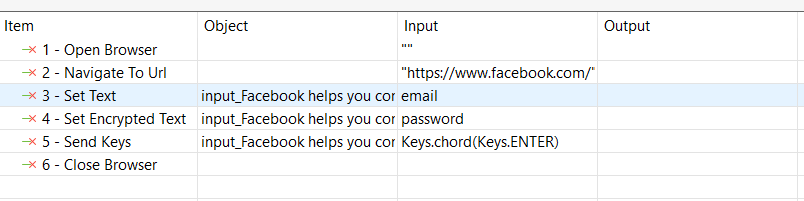
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên kiểm thử** | **Điều kiện kiểm thử (đầu vào)** | **Kiểm thử cái gì (thao tác kiểm thử)** | **Đầu ra** | **Pass/Fail** |
| 1 | Nhập email sai - mật khẩu đúng | Email không tồn tại | Nhập email sai, mật khẩu đúng và click nút đăng nhập | Hệ thống hiển thị thông báo lỗi "Email hoặc mật khẩu không chính xác." | Pass |

Kết quả test thực tế qua phần mềm Katalon

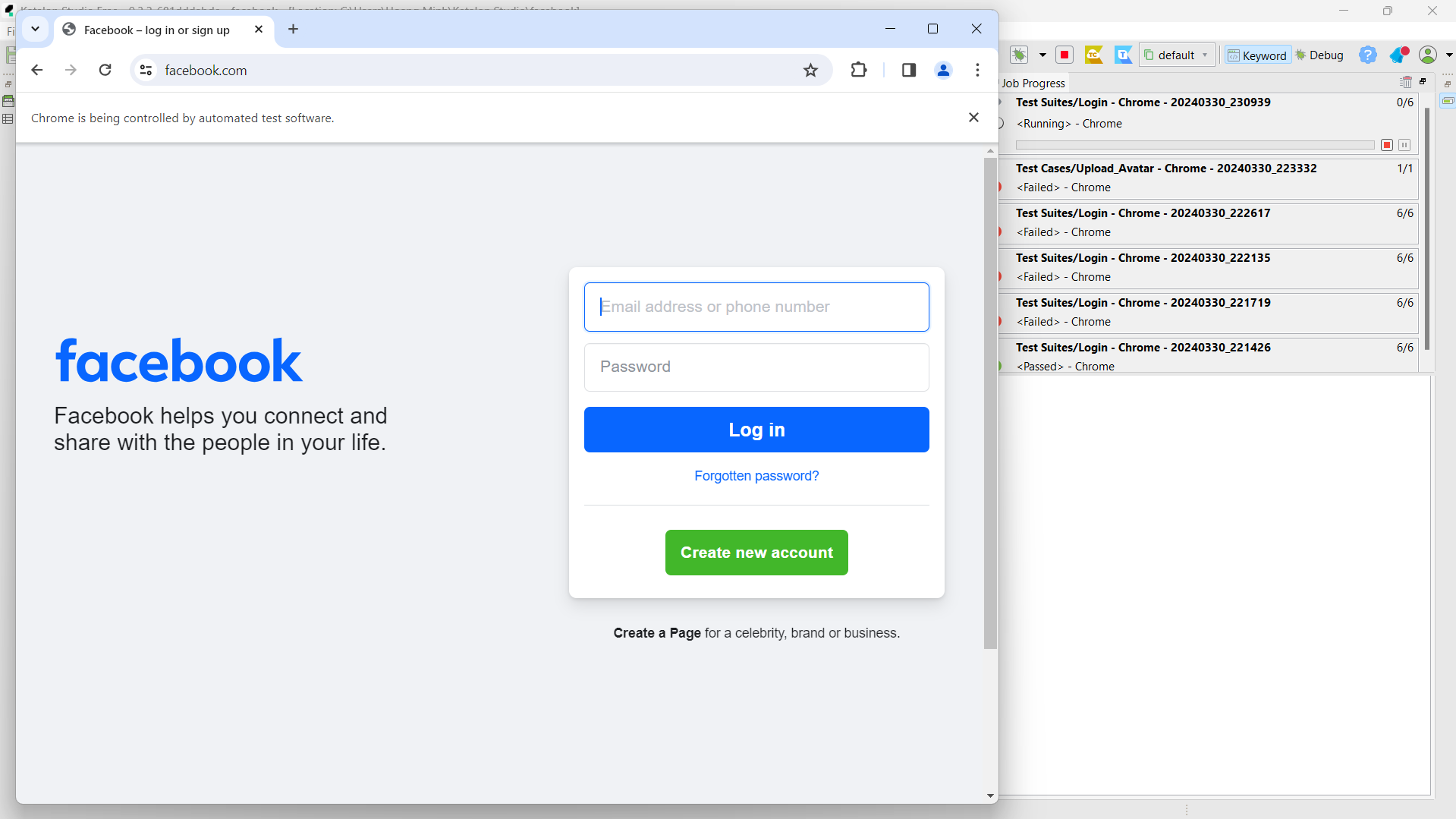
Bảng testcase chạy thực hiện phần mềm Katalon

|  |  |
| --- | --- |
| email | password |
| [kongminh@imail.edu.vn](mailto:kongminh@imail.edu.vn) | Test123456 |
| [kongminh@imail.edu.vn](mailto:kongminh@imail.edu.vn) | Test123457 |
| [kongminh1123@imail.edu.vn](mailto:kongminh1123@imail.edu.vn) | Test123456 |
| [kongminh1123@imail.edu.vn](mailto:kongminh1123@imail.edu.vn) | Test1234578 |

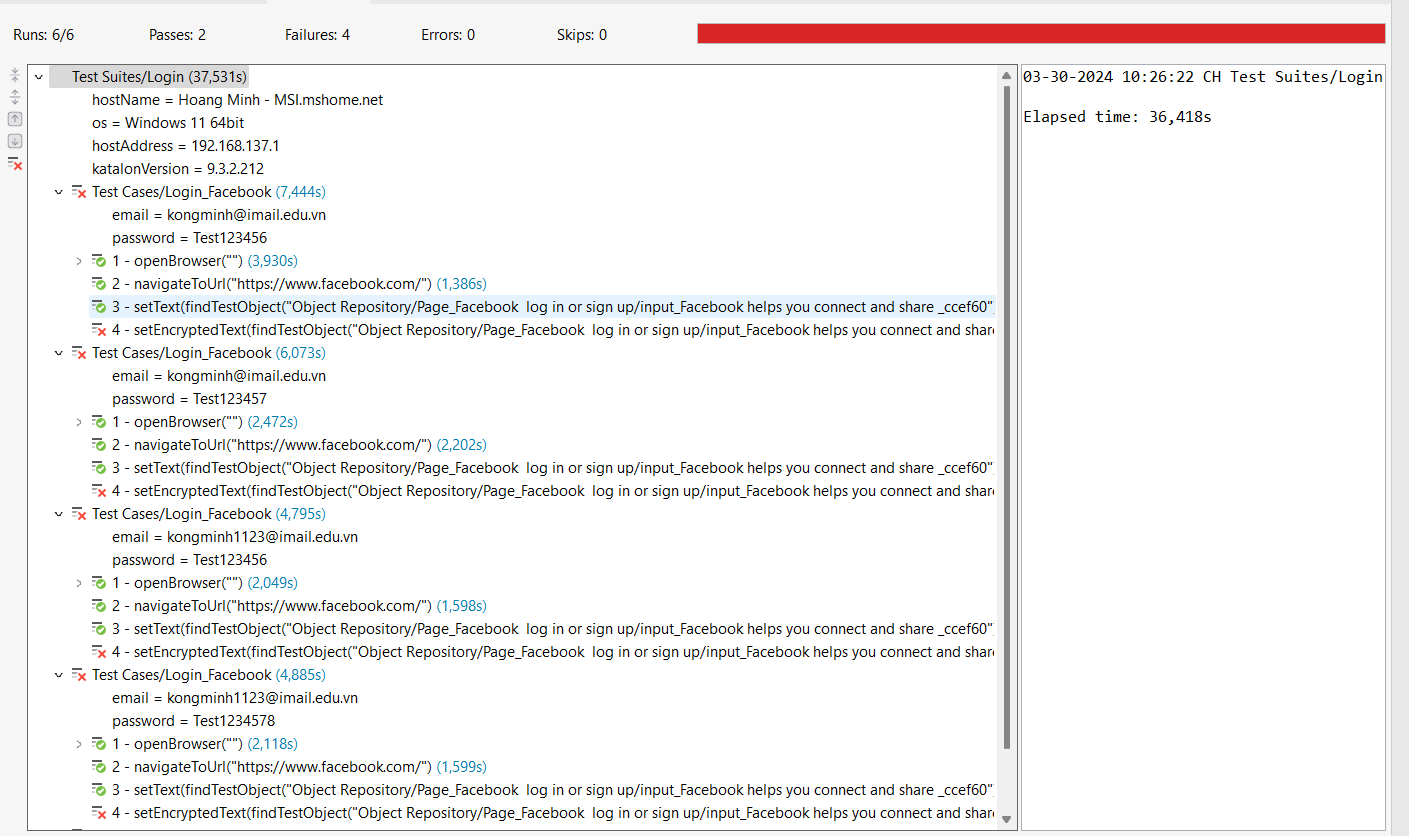
*Bảng 4.4 Bảng testcase chạy thực hiện phần mềm Katalon*

Thao tác thực hiện chức năng login Facebook qua phần mềm Katalon

*Hình 4.2 Thao tác thực hiện chức năng login Facebook*

*Hình 4.3 Giao diện Login trên phần mềm Katalon*

Kết quả chạy thực thi trên phần mềm Katalog

*Hình 4.4 Kết quả chạy thực thi trên phần mềm Katalog*

### **4.2.2. Kiểm thử bảng tính quyết định với chức năng Upload avatar**

Điều kiện Upload thành công là:

* JPG, PNG, GIF, TIFF, HEIF hoặc WebP.
* Kích thước của file hình ảnh từ 4mb trở xuống.
* Độ phân giải: 18 Pixel.

Bảng quyết định:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Điều kiện** | **TH 1** | **TH 2** | **TH 3** | **TH 4** | **TH 5** | **TH 6** | **TH 7** | **TH 8** |
| Định dạng | PNG, GIF, TIFF, HEIF hoặc WebP. | PNG, GIF, TIFF, HEIF hoặc WebP | PNG, GIF, TIFF, HEIF hoặc WebP | PNG, GIF, TIFF, HEIF hoặc WebP | Không phải  PNG, GIF, TIFF, HEIF hoặc WebP | Không phải PNG, GIF, TIFF, HEIF hoặc WebP | Không phải .PNG, GIF, TIFF, HEIF hoặc WebP | Không phải PNG, GIF, TIFF, HEIF hoặc WebP |
| Kích thước | < 4mb | < 4mb | >=  4mb | >= 4mb | < 4mb | < 4mb | >= 4mb | >=4mb |
| Độ phân giải | 18pixel | Không phải 18pixel | 18pixel | Không phải 18pixel | 18pixel | 18pixel | 18pixel | 18pixel |
| Kết quả | Upload ảnh thành công | Thông báo lỗi "Độ phân giải chưa đúng" | Thông báo lỗi "Kích thước chưa đúng" | Thông báo lỗi "Kích thước và Độ phân giải chưa đúng" | Thông báo lỗi "Định dạng chưa đúng" | Thông báo lỗi "Định dạng và Độ phân giải chưa đúng" | Thông báo lỗi "Định dạng và Kích thước chưa đúng" | Thông báo lỗi "Định dạng, Kích thước và Độ phân giải chưa đúng" |

*Bảng 4.4 bảng tính quyết định với chức năng Upload avatar*

Với những điều kiện trên, chúng ta có thể tạo ra 8 case test:

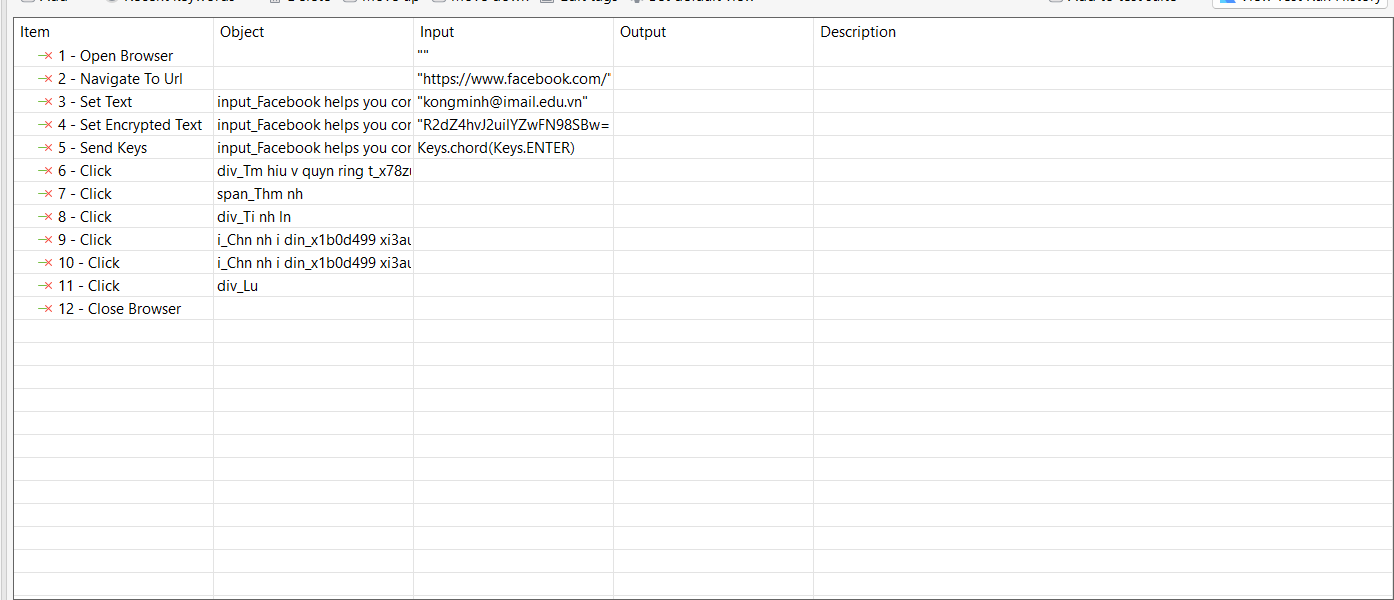
1. Upload hình ảnh với định dạng .PNG, GIF, TIFF, HEIF hoặc WebP., kích thước bé hơn 4mb và độ phân giải là 18pixel -> click nút upload. Kết quả mong muốn là hình ảnh được upload thành công.
2. Upload hình ảnh với định dạng PNG, GIF, TIFF, HEIF hoặc WebP., kích thước bé hơn 4mb và độ phân giải không phải là 18pixel  -> click nút upload. Kết quả mong muốn là hiển thị thông báo lỗi "Độ phân giải chưa đúng".
3. Upload hình ảnh với định dạng .PNG, GIF, TIFF, HEIF hoặc WebP., kích thước lớn hơn hoặc bằng  4mb và độ phân giải là 18pixel  -> click nút upload. Kết quả mong muốn là hiển thị thông báo lỗi "Kích thước chưa đúng".
4. Upload hình ảnh với định dạng .PNG, GIF, TIFF, HEIF hoặc WebP., kích thước lớn hơn hoặc bằng  4mb và độ phân giải không phải là 18pixel  -> click nút upload. Kết quả mong muốn là hiển thị thông báo lỗi "Kích thước và Độ phân giải chưa đúng".
5. Upload hình ảnh với định dạng không phải là .jpg, kích thước bé hơn  4mb và độ phân giải là 18pixel  -> click nút upload. Kết quả mong muốn là hiển thị thông báo lỗi "Định dạng chưa đúng".
6. Upload hình ảnh với định dạng không phải là .jpg, kích thước bé hơn  4mb và độ phân giải không phải là 18pixel  -> click nút upload. Kết quả mong muốn là hiển thị thông báo lỗi "Định dạng và Độ phân giải chưa đúng".
7. Upload hình ảnh với định dạng không phải là .PNG, GIF, TIFF, HEIF hoặc WebP., kích thước lớn hơn hoặc bằng  4mb và độ phân giải là 18pixel  -> click nút upload. Kết quả mong muốn là hiển thị thông báo lỗi "Định dạng và kích thước chưa đúng".
8. Upload hình ảnh với định dạng không phải là .PNG, GIF, TIFF, HEIF hoặc WebP., kích thước lớn hơn hoặc bằng  4mb và độ phân giải không phải là 18pixel  -> click nút upload. Kết quả mong muốn là hiển thị thông báo lỗi "Định dạng, Kích thước và Độ phân giải chưa đúng".

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên kiểm thử** | **Điều kiện kiểm thử (Đầu vào)** | **Kiểm thử cái gì (Thao tác kiểm thử)** | **Đầu ra** | **Pass/Fail** |
| 1 | Upload ảnh đúng định dạng, kích thước và độ phân giải | - Định dạng: .jpg - Kích thước: < 4MB - Độ phân giải: 18pixel | Click nút Upload | Ảnh được upload thành công | Pass |
| 2 | Upload ảnh đúng định dạng, kích thước nhưng sai độ phân giải | - Định dạng: .jpg - Kích thước: < 4MB - Độ phân giải: Không phải 18pixel | Click nút Upload | Hiển thị thông báo lỗi "Độ phân giải chưa đúng" | Pass |
| 3 | Upload ảnh đúng định dạng, độ phân giải nhưng sai kích thước | - Định dạng: .jpg - Kích thước: ≥ 4MB - Độ phân giải: 18pixel | Click nút Upload | Hiển thị thông báo lỗi "Kích thước chưa đúng" | Pass |
| 4 | Upload ảnh đúng định dạng nhưng sai kích thước và độ phân giải | - Định dạng: .jpg - Kích thước: ≥ 4MB - Độ phân giải: Không phải 18pixel | Click nút Upload | Hiển thị thông báo lỗi "Kích thước và Độ phân giải chưa đúng" | Pass |
| 5 | Upload ảnh sai định dạng, kích thước nhưng đúng độ phân giải | - Định dạng: Không phải .jpg - Kích thước: < 4MB - Độ phân giải: 18pixel | Click nút Upload | Hiển thị thông báo lỗi "Định dạng chưa đúng" | Pass |
| 6 | Upload ảnh sai định dạng, kích thước và độ phân giải | - Định dạng: Không phải .jpg - Kích thước: < 4MB - Độ phân giải: Không phải 18pixel | Click nút Upload | Hiển thị thông báo lỗi "Định dạng và Độ phân giải chưa đúng" | Pass |
| 7 | Upload ảnh sai định dạng, kích thước nhưng đúng độ phân giải | - Định dạng: Không phải .jpg - Kích thước: ≥ 4MB - Độ phân giải: 18pixel | Click nút Upload | Hiển thị thông báo lỗi "Định dạng và Kích thước chưa đúng" | Pass |
| 8 | Upload ảnh sai định dạng, kích thước và độ phân giải | - Định dạng: Không phải .jpg - Kích thước: ≥ 4MB - Độ phân giải: Không phải 18pixel | Click nút Upload | Hiển thị thông báo lỗi "Định dạng, Kích thước và Độ phân giải chưa đúng" | Pass |

*Bảng 4.5 Bảng testcase với chức năng Upload avatar*

Kết quả chạy test phần mềm qua Katalog

Thao tác các bước thực hiện để upload ảnh



*Hình 4.5 Thao tác các bước thực hiện để upload ảnh*

Giao diện kiểm thử trên phần mềm Katalon **Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Biểu tượng máy tính

Mô tả được tạo tự động**

*Hình 4.6 Giao diện kiểm thử trên phần mềm Katalon*

Kết quả thực tế khi thực hiện chức năng upload ảnh   
Ảnh có chứa văn bản, Phông chữ, số, ảnh chụp màn hình

Mô tả được tạo tự động

*Hình 4.7 Kết quả thực tế khi thực hiện chức năng upload ảnh*

# **TỔNG KẾT**

- Đề tài đã giúp nhóm em hiểu rõ hơn về phương pháp kiểm thử bảng tính quyết định và cách áp dụng nó vào phần mềm Katalon.

- Qua đề tài đã giúp nhóm em có cơ hội thực hành nâng cao kỹ năng sử dụng phần mềm Katalon để thực hiện tử động trên website.

- Việc kiểm thử tính năng trên 1 trạng mạng xã hội Facebook đã giúp chúng em hiểu thêm cách hoạt động của 1 website.

- Trong quá trình thực hiện dự án, có thể bạn đã phải đối mặt với các vấn đề và thách thức trong việc thiết kế bảng quyết định hoặc triển khai kiểm thử. Việc này giúp bạn học hỏi và cải thiện kỹ năng giải quyết vấn đề.

- Qua đề tài đã giúp nhóm em có cái nhìn tổng quát về quá trình kiểm thử phần mềm và cách thực hiện cải thiện chất lượng phần mềm thông qua kiểm thử.

- Qua đề tài này cũng có thể giúp bạn xác định được sở thích, kỹ năng, và mục tiêu nghề nghiệp trong lĩnh vực kiểm thử và đảm bảo chất lượng phần mềm. Nếu bạn thấy mình thích thú và thành thạo trong việc kiểm thử phần mềm, đây có thể là một bước khởi đầu quan trọng cho sự nghiệp của bạn.

# **TÀI LIỆU THAM KHẢO**

[**https://katalon.com/download-next-steps**](https://katalon.com/download-next-steps)

[**https://viblo.asia/p/huong-dan-ky-thuat-bang-quyet-dinh-decision-table-y37LdD5mLov**](https://viblo.asia/p/huong-dan-ky-thuat-bang-quyet-dinh-decision-table-y37LdD5mLov)

[**https://viblo.asia/p/decision-table-bang-quyet-dinh-trong-testing-GrLZDvzn5k0**](https://viblo.asia/p/decision-table-bang-quyet-dinh-trong-testing-GrLZDvzn5k0)