TRƯỜNG ĐẠI HỌC ĐIỆN LỰC

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN TỐT NGHIỆP**

**ĐỀ TÀI:** **TÌM HIỂU KIỂM THỬ TỰ ĐỘNG VÀ ỨNG DỤNG**

**KIỂM THỬ HỆ THỐNG QUẢN LÝ SINH VIÊN TRONG KÝ TÚC XÁ**

**Sinh viên thực hiện : LÒ HOÀNG LINH**

**Giảng viên hướng dẫn :Ths. TRẦN THỊ MINH THU**

**Ngành : CÔNG NGHỆ THÔNG TIN**

**Chuyên ngành : CÔNG NGHỆ PHẦN MỀM**

**Lớp : D13CNPM6**

**Khóa : 2018-2023**

***Hà nội, 15 tháng 01 năm 2022***

**PHIẾU CHẤM ĐIỂM**

Sinh viên thực hiện:

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **Điểm** | **Chữ ký** |
| 1 | Lò Hoàng Linh |  |  |

Giảng viên chấm:

|  |  |  |
| --- | --- | --- |
| **Họ và tên** | **Chữ ký** | **Ghi chú** |
| Giảng viên chấm 1: |  |  |
| Giảng viên chấm 2: |  |  |

**MỤC LỤC**

[LỜI MỞ ĐẦU 1](#_Toc126053116)

[CHƯƠNG 1 TỔNG QUAN VỀ KIỂM THỬ PHẦN MỀM 2](#_Toc126053117)

[**1.1. Khái niệm cơ bản về kiểm thử phần mềm** 2](#_Toc126053118)

[1.1.1. Kiểm thử phần mềm 2](#_Toc126053119)

[1.1.2. Mục tiêu của kiểm thử phần mềm 2](#_Toc126053120)

[1.1.3. Tầm quan trọng của kiểm thử phần mềm 2](#_Toc126053121)

[1.1.4. Ai tham gia vào kiểm thử phẩn mềm 3](#_Toc126053122)

[1.1.5. Quy trình kiểm thử phần mềm 3](#_Toc126053123)

[1.2. Mô hình kiểm thử phần mềm 5](#_Toc126053124)

[1.3. Các cấp độ kiểm thử phần mềm 9](#_Toc126053125)

[1.4. Phân loại kiểm thử 11](#_Toc126053126)

[1.4.1. Khái niệm 11](#_Toc126053127)

[1.4.2. Điểm khác nhau giữa kiểm thử thủ công và kiểm thử tự động 11](#_Toc126053128)

[1.4.3. Ưu điểm và nhược điểm 15](#_Toc126053129)

[1.4.4 Khi nào thì sử dụng kiểm thử thủ công và khi nào thì sử dụng kiểm thử tự động? 15](#_Toc126053130)

[1.5. Quy trình kiểm thử tự động 16](#_Toc126053131)

[1.5.1. Một số Test automation framework 18](#_Toc126053132)

[1.5.2. Tại sao nên lựa chọn Automation testing 19](#_Toc126053133)

[CHƯƠNG 2 TÌM HIỂU VỀ SELENIUM WEBDRIVER 20](#_Toc126053134)

[2.1 Lịch sử phát triển 20](#_Toc126053135)

[2.2 Giới thiệu chung 20](#_Toc126053136)

[2.2.1 Khái niệm 20](#_Toc126053137)

[2.2.2 Ưu điểm và nhược điểm 21](#_Toc126053138)

[2.2.3 Cấu trúc 21](#_Toc126053139)

[CHƯƠNG 3: THỰC HIỆN THỬ NGHIỆM KIỂM THỬ TRONG HỆ THỐNG QUẢN LÝ SINH VIÊN TRONG KÝ TÚC XÁ 23](#_Toc126053140)

[3.1. Giới thiệu về hệ thống Quản lý sinh viên trong ký túc xá 23](#_Toc126053141)

[3.1.1. Giới thiệu về hệ thống 23](#_Toc126053142)

[3.1.2. Quy trình quản lý ký túc xá sinh viên 23](#_Toc126053143)

[3.1.2.1. Về phía admin 23](#_Toc126053144)

[3.1.2.2 . Về phía sinh viên 25](#_Toc126053145)

[3.1.3. Chức năng của hệ thống 25](#_Toc126053146)

[3.1.3.1 Chức năng chung 25](#_Toc126053147)

[3.1.3.2. Đối với admin 25](#_Toc126053148)

[3.1.3.3. Đối với sinh viên 26](#_Toc126053149)

[3.1.4. Một số màn hình giao diện chính của hệ thống quản lý sinh viên ký túc xá 26](#_Toc126053150)

[3.3. Sự khác nhau giữ kịch bản kiểm thử và kịch bản tự động 27](#_Toc126053151)

[3.4. Kịch bản kiểm thử thủ công trong hệ thống quản lý sinh viên ký túc xá 27](#_Toc126053152)

[3.4.1. Kịch bản kiểm thử chức năng Đăng nhập 27](#_Toc126053153)

[3.5. Kịch bản kiểm thử tự động trong hệ thống quản lý sinh viên ký túc xá 48](#_Toc126053154)

[3.5.1. Kịch bản kiểm thử tự động chức năng đăng nhập 48](#_Toc126053155)

[3.5.2. Kịch bản kiểm thử tự động chức năng thêm mới sinh viên 52](#_Toc126053156)

[KẾT LUẬN 58](#_Toc126053157)

[TÀI LIỆU THAM KHẢO 59](#_Toc126053167)

# DANH MỤC CÁC HÌNH ẢNH

[Hình ảnh 1‑0‑1 Mô hình thác nước 10](#_Toc110178369)

[Hình ảnh 1‑0‑2 Mô hình chữ V 13](#_Toc110178370)

[Hình ảnh 1‑0‑3 Mô hình chữ W 14](#_Toc110178371)

[Hình ảnh 1‑0‑4 Các cấp độ kiểm thử phần mềm 15](#_Toc110178372)

[Hình ảnh 1‑0‑5 Tiến trình tự động hóa 23](#_Toc110178373)

**Lời cảm ơn**

Đầu tiên em xin chân thành gửi lời cảm ơn đến tất cả các thầy cô khoa Công nghệ thông tin của trường Đại học Điện Lực Hà Nội, đã nhiệt tình hướng dẫn, truyền đạt kiến thức trong suốt thời gian học tập tại trường, là nền tảng giúp em thực hiện đồ án tốt nghiệp này.

Em xin gửi lời cảm ơn sâu sắc đến cô ThS. Trần Thị Minh Thu đã tận tình hướng dẫn, chỉ bảo trong suốt thời gian làm đồ án, giúp em có những hướng đi đúng đắn để có thể hoàn thành bản đồ án tốt nghiệp này.

Một lần nữa, em xin chân thành cảm ơn!

# LỜI MỞ ĐẦU

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 suất phần mềm phải có một phương pháp để nâng cao chất lượng sản phẩm cũng như tối tru 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.

Kiểm thử phần mềm là một khâu rất quan trọng trong quá trình phát triển phần mềm. Kiểm thử phần mềm để kiểm tra phần mềm có đúng với đặc tả và thiết kế hệ thống không, có đáp ứng yêu cầu người dùng không, có lỗi lập trình không, hoạt động có hiệu quả không,...Như vậy, kiểm thử phần mềm là để đáp ứng yêu cầu người dùng, phát triển lỗi để từ đó nâng cao chất lượng phần mềm. Vậy làm thế nào để có thể kiểm tra dự án phần mềm của ta chạy ổn định, đạt được tính hiệu quả cao, nhưng lại tiết kiệm được thời gian cũng như kinh phí trong quả trình kiểm thử là một điều thiết yếu đối với các nhà kiểm thử.

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 tiết kiệm 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 “Tìm hiểu kiểm thử tự động và ứng dụng kiểm thử hệ thống quản lý sinh viên trong ký túc xá”.

# CHƯƠNG 1 TỔNG QUAN VỀ KIỂM THỬ PHẦN MỀM

**1**.1. 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 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 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 hiển 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, ≥ 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
* Chi phí của kiểm thử chiếm: 40% tổng công sức phát triển ≥ 30% tổng thời gian phát triển
* Với phần mềm ảnh hưởng tới sinh mạng chi phí có thể gấp từ 3 đến 5 lần tổng chi phí khác cộng lại.
* Mục tiêu:
* Phát hiện càng nhiều lỗi càng tốt trong thời gian xác định trước.
* Và phải làm thế nào nghĩ ra cách ngăn chặn lỗi phát sinh.
* Chứng minh rằng sản phẩm phần mềm phù hợp với các đặc tả của nó.
* Chứng tỏ được rằng các testcase và các báo cáo tạo ra đúng và hữu dụng.

### 1.1.4. Ai tham gia vào kiểm thử phẩn 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

Kế hoạch kiểm thử:

* Requirement analysis – Phân tích yêu cầu:
* QA 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. Mô hình kiểm thử phần mềm

* **Mô hình thác nước**

*Hình ảnh 1.1: Mô hình thác nước*

|  |  |  |
| --- | --- | --- |
| **TT** | **Giai đoạn** | **Các hoạt động đã thực hiện** |
| 1 | Phân tích yêu cầu | 1. Nắm được tất cả các yêu cầu.  2. Thảo luận và hướng dẫn để hiểu các yêu cầu.  3. Thực hiện thử nghiệm các yêu cầu để đảm bảo rằng các yêu cầu có thể kiểm chứng được hay không. |
| 2 | Thiết kế hệ thống | 1. Theo yêu cầu, tạo ra thiết kế  2. Thảo luận về yêu cầu phần cứng /phần mềm.  3. Tài liệu thiết kế |
| 3 | Thực hiện | 1. Theo thiết kế tạo ra các chương trình / code  2. Tích hợp code cho giai đoạn tiếp theo.  3. Unit testing |
| 4 | Thử nghiệ:m hệ thống | 1. Tích hợp unit tested code và kiểm tra nó để đảm bảo nó hoạt động như mong đợi.  2. Thực hiện tất cả các hoạt động thử nghiệm (Functional and non functional) để đảm bảo hệ thống đáp ứng các yêu cầu.  3. Trong trường hợp bất thường, báo cáo.  4. Theo dõi tiến độ về kiểm tra thông qua các công cụ như số liệu truy xuất nguồn gốc, ALM  5. Báo cáo hoạt động thử nghiệm |
| 5 | Triển khai Hệ thống | 1. Môi trường đang hoạt động  2. Không có lỗi mở server.  3. Đảm bảo rằng các tiêu chí test được đáp ứng.  4. Triển khai ứng dụng trong môi trường tương ứng.  5. Thực hiện kiểm tra về môi trường sau khi ứng dụng được triển khai để đảm bảo ứng dụng không gặp vấn đề |
| 6 | Bảo trì hệ thống | 1. Ứng dụng đang chạy và chạy trong môi trường tương ứng.  2. Trong trường hợp người dùng gặp lỗi, chắc chắn giải quyết và khắc phục các vấn đề.  3. Trong trường hợp bất kỳ vấn đề fixed; code cập nhật được triển khai trong môi trường.  4. Ứng dụng luôn được tăng cường để kết hợp nhiều tính năng, cập nhật môi trường với các tính năng mới nhất |

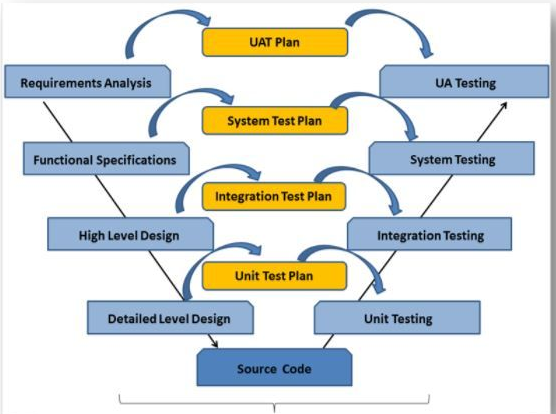
**Ưu điểm:**

Dễ sử dụng, dễ tiếp cận, dễ quản lý. Sản phẩm phát triển theo các giai đoạn được xác định rõ ràng. Xác nhận ở từng giai đoạn, đảm bảo phát hiện sớm các lỗi.

**Nhược điểm:**

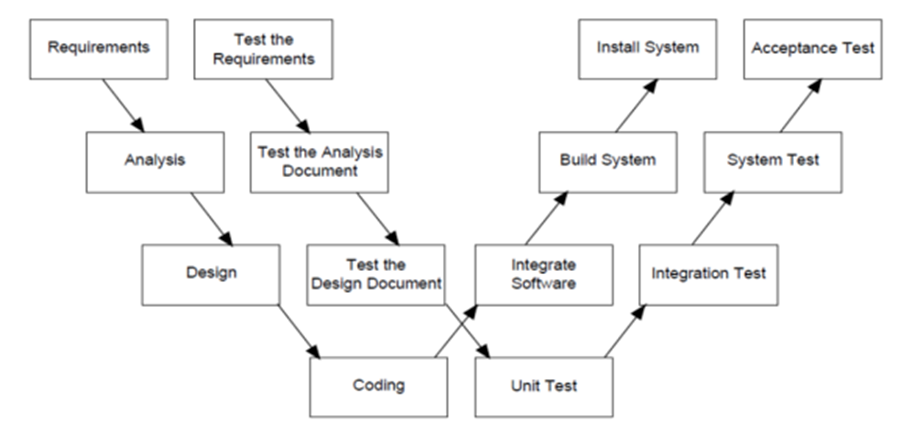
Ít linh hoạt, phạm vi điều chỉnh hạn chế. Rất khó để đo lường sự phát triển trong từng giai đoạn. Mô hình không thích hợp với những dự án dài, đang diễn ra, hay những dự án phức tạp, có nhiều thay đổi về yêu cầu trong vòng đời phát triển. Khó quay lại khi giai đoạn nào đó đã kết thúc.

* **Mô hình chữ V (V-model).**



*Hình ảnh 1‑0‑2 Mô hình chữ V*

* Mô hình chữ V có 4 mức độ kiểm thử:
* Kiểm thử thành phần (Component testing).
* Kiểm thử tích hợp (integration testing).
* Kiểm thử hệ thống (system testing).
* Kiểm thử chấp nhận (acceptance testing).
* **Ưu điểm:**
* Đơn giản dễ sử dụng.
* Có hoạt động, kế hoạch cụ thể cho quá trình test.
* Tiết kiệm được thời gian, và có cơ hội thành công cao hơn waterfall.
* Chủ động trong việc phát hiện bug, sớm tìm ra bug ngay từ những bước đầu.
* **Nhược điểm:**
* Khó quản lý kiểm soát rủi ro, rủi ro cao.
* Không phải là một mô hình tốt cho các dự án phức tạp và hướng đối tượng.
* Mô hình hoạt động không hiệu quả đối với các dự án dài và đang diễn ra.
* Không thích hợp cho các dự án có nguy cơ thay đổi yêu cầu trung bình đến cao.
* **Mô hình chữ W (W-model).**



*Hình ảnh 1‑0‑3 Mô hình chữ W*

Mô hình chữ W (W-model) là mô hình được phát triển dựa trên mô hình chữ V. Nếu như trong mô hình V chỉ tập trung vào kiểm thử động (dynamic testing), không chú trọng đến lợi ích và hiệu quả của kiểm thử tĩnh (static testing) như: đánh giá, kiểm tra và phân tích code thì mô hình W nhằm giải quyết những thiếu sót trong mô hình V.

Trong mô hình W kỹ thuật kiểm thử tĩnh được áp dụng trong đoạn đầu của sự phát triển. Tất cả các hoạt động của quá trình phát triển phần mềm ánh xạ đến một hoạt động kiểm thử. Các mô hình W đặt quá trình kiểm thử bình đẳng với quá trình phát triển phần mềm.

### 1.3. Các cấp độ kiểm thử phần mềm

Cấp độ kiểm thử phần mềm hay mức độ kiểm thử phầm 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.

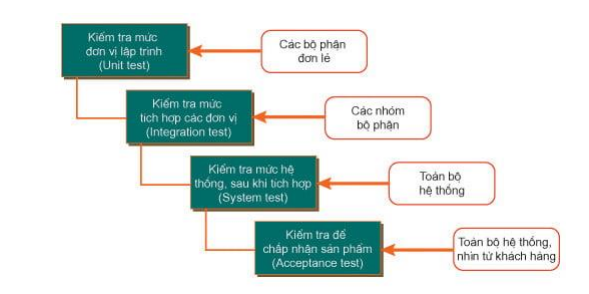
Tất cả các giai đoạn của quá trình phát triển phầm mềm đều trải qua bốn mức kiểm thử phần mềm là:

Kiểm thử đơn vị (Unit Testing)

Kiểm thử tích hợp (Integration Testing)

Kiểm thử hệ thống (System Testing)

Kiểm thử chấp nhận (Acceptance Testing)



*Hình ảnh 1‑0‑4 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 Testing 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 bản thân 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 thăng 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.

### 1.4. 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.

### 1.4.1. 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.

### 1.4.2. 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 |
| Kiểm thử ngẫu nhiên (Exploratory testing) | Tự động hóa không cho phép kiểm thử ngẫu nhiên | Kiểm thử ngẫu nhiên có thể thực hiện trong Kiểm thử thủ công |
| Đầu tư ban đầu | Đầu tư ban đầu trong kiểm thử tự động cao hơn. Mặc dù ROI tốt hơn về lâu dài. | Đầu tư ban đầu trong kiểm thử thủ công là tương đối thấp hơn kiểm thử tự động. ROI thấp hơn so với kiểm thử Tự động trong thời gian dài. |
| 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 nhàm chán | Kiểm thử thủ công có thể bị nhàm chán 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, 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 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 dung quan sát, có thể hữu ích để cung cấp hệ thống thân thiện với người dùng. |
| Kiểm thử hiệu suất | Các loại Kiểm thử hiệu suất như Load Testing, Stress Testing, Spike Testing, v.v. bắt buộc phải được kiểm thử bằng công cụ tự động. | Kiểm thử hiệu suất là không khả thi |
| Thực hiện kiểm thử tương đồng | Có thể được thực hiện tương đồng trên các nền tảng vận hành khác nhau và giảm thời gian thực hiện kiểm thử. | Kiểm thử thủ công có thể được thực hiện tương đồng nhưng sẽ cần tăng nguồn nhân lực |
| Kiểm thử hàng loạt | Có thể chạy hàng loạt các Script, có thể thực thi vào ban đêm. | 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. |
| Tiếp cận lý tưởng | Kiểm thử tự động rất hữu ích khi thường xuyên thực hiện cùng một bộ test cases | Kiểm thử thủ công chứng minh hữu ích khi bộ test cases chỉ cần chạy một hoặc hai lần. |
| Kiểm thử xác minh build (Build Verification Testing - BVT) | Kiểm thử tự động rất hữu ích cho BVT. | Thực hiện BVT là rất khó khăn và tốn thời gian 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 |
| Devops | Kiểm thử tự động giúp Kiểm thử xác minh build và là một phần không thể thiếu của Chu kỳ DevOps | Kiểm thử thủ công loại bỏ nguyên lý build tự động của DevOps |

### 1.4.3. Ưu điểm và nhược điểm

|  |  |  |
| --- | --- | --- |
| Kiểm thử | Điểm mạnh | Điểm yếu |
| Thủ công | Cho phép tester thực hiện việc kiểm thử khám phá  Thích hợp kiểm tra sản phẩm lần đầu tiên  Thích hợp kiểm thử trong trường hợp các test case chỉ phải thực hiện một số ít lần  Giảm được chi phí ngắn hạn. | Tốn thời gian.  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. |
| Tự động | - 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 - 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. | - Tốn kém hơn kiểm thử tự động, chi phí đầu tư ban đầu lớn  - Kiểm thử thủ công là không thể thay thế vì người ta không thể tự động hóa mọi thứ. |

### 1.4.4 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 trong trường hợp này.

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 trong kịch bản này

Đối với những dự án ngắn hạn và không nhiều kinh phí, không nên đầu tư vào automation testing. Manual testing là sự lựa chọn phù hợp cho trường hợp này.

Đối với những dự án tần suất thực hiện test thấp, khoảng 2-3 lần/năm, thì không nên đầu tư vào việc automation testing.

* 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

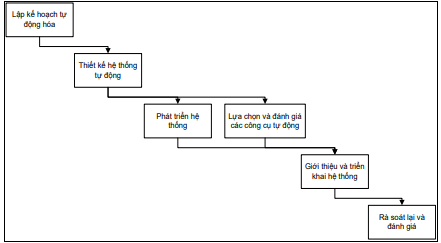
### 1.5. Quy trình kiểm thử tự động

Kiểm thử tự động sẽ được sử dụng khi dự án không đủ tài nguyên (thời gian, nhân lực và chi phí), như sau khi sản phẩm được sửa đổi, nâng cấp cần phải thực hiện kiểm thử hồi qui để chắc chắn rằng việc sửa đổi nâng cấp không làm ảnh hưởng đến các chức năng đã có của sản phẩm. Ngoài ra, kiểm thử tự động cũng được lựa chọn khi phải kiểm tra khả năng vận hành của sản phẩm trong các môi trường đặc biệt mà nếu dùng kiểm thử thủ công sẽ rất khó khăn để thực hiện như là 9 đo tốc độ xử lý, xác định khả năng chịu tải tối đa, kiểm tra cơ chế an ninh, an toàn, lặp đi lặp lại một số thao tác trong thời gian dài…

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.

Theo Douglas Hoffman, một tiến trình hiệu quả để thực hiện tự động hóa kiểm thử bao gồm các bước như ở hình dưới đây:



*Hình ảnh 1.5 Tiến trình tự động hóa*

**Bước 1:** Lập kế hoạch tự động hóa. Đây là bước khởi tạo. 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 phầm mềm.

**Bước 2:** Thiết kế kiểm thử tự động. Mục tiêu chính của bước này là tạo ra một giải pháp tự động hóa chi tiết, phù hợp với yêu cầu và mục tiêu trong kế hoạch đã vạch ra. Bao gồm 2 nhiệm vụ chính. Thứ nhất là xác định và lựa chọn các công cụ có sẵn (các sản phẩm thương mại hoặc tự làm) để hỗ trợ tiến trình tự động hóa. 10 Để thực hiện nhiệm vụ này, người tự động hóa cần phải có thông tin hướng dẫn và đánh giá về công cụ được lựa chọn. Thứ hai là thiết kế giải pháp tự động hóa.

**Bước 3:** Phát triển công cụ kiểm thử tự động. Ở bước này, sẽ thực hiện phát triển công cụ dựa theo giải pháp tự động hóa đã đưa ra. Điểm mấu chốt ở bước này là phải chắc chắn rằng công cụ đã được phát triển là đáng tin cậy và dễ sử dụng, có tài liệu hướng dẫn tốt. Rất nhiều dự án tự động kiểm thử đã thất bại bởi vì chất lượng kém và tài liệu không tốt.

**Bước 4:** Triển khai công cụ tự động. Giống như một sản phẩm thương mại, công cụ tự động và các lợi ích của nó cần phải được giới thiệu và triển khai sử dụng cho một sản phẩm nào đó. Ở bước này, việc đào tạo cho người dùng và hỗ trợ họ là thiết yếu.

**Bước 5:** Rà soát lại (review) và đánh giá. Bất cứ khi nào một công cụ mới được triển khai, thì đều phải tiến hành rà soát lại để xác định các vấn đề và hạn chế của công cụ, đánh giá các tính năng mà nó cung cấp. Kết quả của việc rà soát này là bài học cho các lần triển khai về sau.

### 1.5.1. Một số Test automation framework

***Selenium Framework***

Khung công tác Selenium là cấu trúc khung công tác tự động hóa được sử dụng rộng rãi nhất để tăng khả năng đọc mã và do đó cải thiện năng suất kiểm tra. 'Trường hợp thử nghiệm' và 'dữ liệu' được giữ riêng biệt với nhau để kiểm tra mức độ hiệu quả của trang web có thể chạy. Các khuôn khổ tự động hóa Selenium cũng có thể được sử dụng bằng cách thực thi các trường hợp thử nghiệm từ một nguồn bên ngoài.

***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 bạn 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.

### 1.5.2. 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 nhàm chán 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 SELENIUM** **WEBDRIVER**

## **2.1** **Lịch sử phát triển**

Selenium được phát triển bởi Jason Huggins làm việc tại ThoughtWorks có trụ sợ 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 đươc 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.

### 2.2 Giới thiệu chung

### 2.2.1 Khái niệm

Selenium hỗ trợ tự động hóa trên hầu hết các trình duyệt chính như firefox, google chrome, microsoft edge thông qua việc sử dụng WebDriver. WebDriver là một API và giao thức xác định giao diện ngôn ngữ trung lập để kiểm soát hành vi của trình duyệt web.

Từng trình duyệt được hỗ trợ bởi một WebDriver cụ thể, được gọi là trình điều khiển. Trình điều khiển là thành phần chịu trách nhiệm ủy quyền cho trình duyệt và xử lý giao tiếp đến và từ Selenium và trình duyệt. Sự tách biệt này là một phần trong nỗ lực có ý thức nhằm yêu cầu các nhà cung cấp trình duyệt chịu trách nhiệm triển khai trình duyệt của họ. Selenium sử dụng các trình điều khiển của bên thứ ba này nếu có thể, nhưng cũng cung cấp các trình điều khiển riêng do dự án duy trì cho các trường hợp khi điều này không thực tế.

Khung Selenium liên kết tất cả các phần này lại với nhau thông qua giao diện hướng tới người dùng cho phép sử dụng các phụ trợ trình duyệt khác nhau một cách minh bạch, cho phép tự động hóa nhiều trình duyệt và đa nền tảng. Thiết lập Selenium khá khác so với thiết lập các công cụ khác. Trước khi bạn có thể bắt đầu code Selenium, bạn phải cài đặt các thư viện ràng buộc ngôn ngữ cho ngôn ngữ bạn chọn, trình duyệt bạn muốn sử dụng và trình điều khiển cho trình duyệt đó.

### 2.2.2 Ưu điểm và nhược điểm

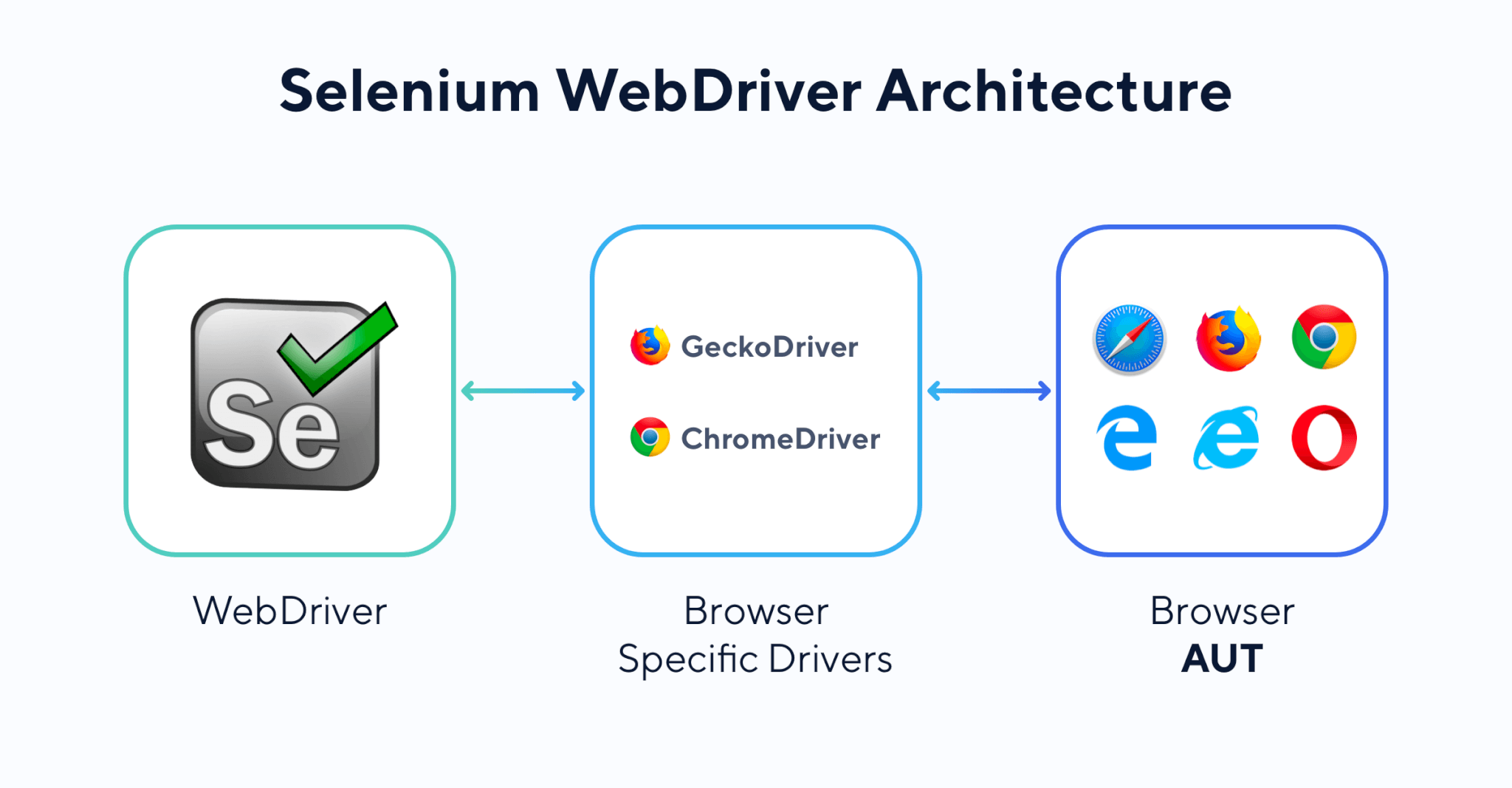
**- Ưu điểm:**

* Cài đặt, sử dụng đơn giản
* Linh hoạt có thể mở rộng các chức năng thông qua các module Python hoặc Java.
* Không yêu cầu phần cứng
* Hoàn toàn miễn phí
* Hỗ trợ nhiều trình duyệt khác nhau
* Có khả năng chạy trên nhiều hệ điều hành khác nhau mà không cần chỉnh sửa kịch bản kiểm thử hoặc các từ khóa ở tầng dưới.

**- Nhược điểm:**

* Chỉ hỗ trợ websites
* Đòi hỏi có kinh nghiệm lập trình
* Thiếu tính năng báo độngcá tự động

### 2.2.3 Cấu trúc

 *Hình 2.1: Cấu trúc Selenium Webdriver*

Kiến trúc Selenium WebDriver bao gồm ba thành phần: WebDriver, trình điều tkhiển dành riêng cho trình duyệt hay còn gọi trình điều khiển và trình duyệt. Giao tiếp giữa các thành phần này là hai chiều, gửi yêu cầu HTTP và nhận phản hồi HTTP dưới dạng dịch vụ RESTful.

WebDriver chứa các thư viện máy khách dành riêng cho ngôn ngữ, hay còn gọi là các ràng buộc ngôn ngữ để viết tập lệnh tự động hóa và thực thi hướng dẫn tập lệnh trong trình duyệt thông qua Trình điều khiển. Các ràng buộc ngôn ngữ này cho phép Selenium hỗ trợ viết tập lệnh bằng nhiều ngôn ngữ khác nhau như Java, C#, Python, Ruby và JavaScript. Bên thứ ba có thể mở rộng Selenium bằng cách triển khai thư viện máy khách cho một ngôn ngữ mới.

Trình điều khiển dành riêng cho trình duyệt hay còn gọi trình điều khiển chịu trách nhiệm kiểm soát trình duyệt thực tế bằng cách chuyển tiếp các yêu cầu từ WebDriver. Mỗi trình điều khiển dành riêng cho từng trình duyệt và thường chạy trên cùng một hệ thống với trình duyệt. Trình điều khiển có thể được triển khai và mở rộng bởi bên thứ ba. GeckoDriver dành cho Firefox và ChromeDriver dành cho Chrome là hai triển khai trình điều khiển dành cho các trình duyệt này.

Trình duyệt là trình duyệt thực tế nhận yêu cầu từ thành phần Trình điều khiển để thúc đẩy các tương tác trên ứng dụng đang được thử nghiệm. Selenium hỗ trợ tất cả các trình duyệt chính, bao gồm Chrome, Firefox, Internet Explorer, Opera và Safari.

WebDriver có thể chạy trên một hệ thống khác với các thành phần trình điều khiển và Trình duyệt. Trong cài đặt này, WebDriver giao tiếp từ xa với các thành phần này thông qua Selenium Server hoặc Remote WebDriver. Điều này cho phép Selenium chạy thử nghiệm tự động hóa trên nhiều trình duyệt và máy cùng một lúc.

# CHƯƠNG 3: THỰC HIỆN THỬ NGHIỆM KIỂM THỬ TRONG HỆ THỐNG QUẢN LÝ SINH VIÊN TRONG KÝ TÚC XÁ

## 3.1. Giới thiệu về hệ thống Quản lý sinh viên trong ký túc xá

### 3.1.1. Giới thiệu về hệ thống

Hệ thống quản lý sinh viên ký túc xá cho 1 trường đại học có quy mô nhỏ.

Ngày nay, công nghệ thông tin càng một phát triển và ứng dụng công nghệ thông tin vào các lĩnh vực xã hội nhằm cải thiện thời gian, công sức và chất lượng rất được chú ý đến

Tại nước ta, ngành tin học đang ngày một khẳng định vị trí, vai trò trong mọi lĩnh vực kinh tế, chính trị, xã hội. Cụ thể hơn là đã có rất nhiều cơ quan Nhà nước, công sở, xí nghiệp, bệnh viện, trường học... đưa tin học vào ứng dụng, áp dụng thành công các hệ thống thông tin. Điều này giúp việc quản lý tại các cơ sở đó trở nên hiệu quả hơn rất nhiều thông qua việc giảm thiểu thời gian, tiền bạc, công sức và đặc biệt có tính logic và độ chính xác rất cao.

Trước đây, việc quản lý chỉ dựa trên việc ghi chép tay, quản lý qua sổ sách và giấy tờ. Việc làm này có thể dẫn đến rất nhiều rủi ro như bị thất lạc giấy tờ, giấy tờ bị mối mọt do lâu ngày, hơn nữa là thông tin bị sai lệch mâu thuãn với nhau, tìm kiếm một thông tin nào đó cũng mất khá nhiều thời gian... Vì vậy, áp dụng tin học vào trong quản lý ký túc xá sinh viên là một việc làm hợp lý, giải quyết những vấn đề và khó khăn đã gặp phải trước kia. Không chỉ giúp nhân viên, nhà quản lý thu hẹp không gian lưu trữ, tránh được thất lạc dữ liệu, xử lý nhanh một khối lượng lơn về thông tin, tra cứu tìm kiếm một cách nhanh chóng vơi độ chính xác cao, tiết kiệm được nhiều thời gian và công sức mà điều này cũng chính là giúp cả người thân, bạn bè hay chính bản thân sinh viên.

Hệ thống quản lý ký túc xá sinh viên chuyển giao quản lý thủ công sang quản lý tin học hiện đại mà không làm biến đổi quy trình quản lý công việc hiện tại, giúp cho việc quản lý hệ thống trở nên dễ dàng thực hiện hơn, tiện lợi hơn trong việc quản lý thông tin về đăng ký của sinh viên, cơ sở vật chất, phản ánh của sinh viên về ký túc xá và báo cáo thống kê.

### 3.1.2. Quy trình quản lý ký túc xá sinh viên

Mỗi người dùng, tùy thuộc vào là quản lý (admin) hay sinh viên (student) sẽ có duy nhất một tên đăng nhập và mật khẩu để sử dụng cho việc đăng nhập tài khoản trên hệ thống quản lý ký túc xá.

### 3.1.2.1. Về phía admin

Admin thực hiện đăng nhập vào hệ thống bằng tài khoản email được cấp, admin có thể sửa thông tin mật khẩu.

Sau khi sinh viên tới văn phòng hoàn tất thủ tục đóng phí và ký hợp trên giấy thì công việc đầu tiên của admin là thêm mới sinh viên vào hệ thống.

Khi được thêm mới thành công, lúc này sinh chưa được thêm vào phòng trên trên hệ thống nên có thạng thái "Không có phòng".

Sau khi được xếp phòng tại chức năng "Quản lý phòng", lúc này sinh viên chuyển trạng thái "Đã có phòng".

Ở trạng thái "Đã có phòng", nếu vi phạm nội quy ký túc xá, sinh viên sẽ bị kỷ luật. Admin sẽ nhập vào lý do bị kỷ luật và chuyển sinh viên sang trạng thái "Kỷ luật". Ở trạng thái này, hệ thống thực hiện "Huỷ" hợp đồng và xoá sinh viên khỏi phòng đang ở. Sau này, hệ thống cũng không cho phép sinh viên đăng ký.

Từ danh sách sinh viên đã được lập, admin bắt đầu thực hiện thêm hợp đồng cho các sinh viên theo mã số sinh viên. Hợp đồng sẽ có 3 trạng thái: "Hoạt động" - "Gia hạn" - "Hết hạn"

* Trường hợp 1: sau khi thêm mới hợp đồng thành công, hợp đồng sẽ có trạng thái "Hoạt động".
* Trường hợp 2:
* Sinh viên không thực hiện gia hạn khi hợp đồng đang có trạng thái "Hoạt động". Hệ thống tự động chuyển trạng thái "Huỷ".
* Sau này khi sinh viên muốn đăng ký ở nội trú ký túc xá sẽ phải thực hiện thêm mới hoạt đồng tại trường hợp 1.
* Trường hợp 3:
* Khi bản ghi hợp đồng hiện taị có trạng thái "Hoạt động", sinh viên có thể đến văn phòng nộp thêm phí để gia hạn hợp đồng và tiếp tục sinh hoạt nội trú tại ký túc xá.
* Lúc này, admin sẽ gia hạn bản ghi hợp đồng hiện tại cho sinh viên bằng cách thêm một bản ghi hợp đồng có trạng thái "Gia hạn" vào cơ sở dữ liệu.
* Khi bản ghi hợp đồng hiện tại hết hạn, hệ thống tự động chuyển trạng thái "Hết hạn" và bản ghi hợp đồng có trạng thái "Gia hạn" sẽ được chuyển thành "Hoạt động".
* Trường hợp 4: sinh viên vi phạm kỷ luật, hệ thống tự động chuyển trạng thái hợp đồng sang "Huỷ" và sau này không cho phép sinh viên đó đăng ký ở nội trú ký túc xá.

Admin sẽ thực hiện thêm toà nhà sau đó thêm phòng cho các toà nhà. Trong một phòng, admin có thể thêm hoặc sinh viên tại phòng đó. Quá trình cập nhật sinh viên trong một phòng sẽ được admin thực hiện như sau: khi có điều chỉnh về sinh viên trong phòng (xóa, thêm sinh viên), ví dụ trường hợp sinh viên chuyển phòng, admin sẽ tìm kiếm mã phòng cần cập nhật và thực hiện xóa sinh viên tại phòng cũ sau đó thêm vào phòng mới. Số lượng sinh viên trong các phòng tự động điều chỉnh giảm/tăng.

Ngoài ra, hệ thống còn cung cấp chức năng quản lý hóa đơn điện nước: hàng tháng, nhân viên sẽ ghi nhận số lượng điện nước tiêu thụ tại công tơ riêng của mỗi phòng. Mỗi phòng có nghĩa vụ đóng tiền điện nước, admin thực hiện nhập thông tin, tạo lập hóa đơn trên hệ thống, thanh toán và in hóa đơn cho phòng.

Admin tiếp nhận phản ánh tứinh viên gửi lên, họ có thể từ chối hoặc nhận xử lý các phản ánh. hHi nhận xử lý, admin phải vào cập nhật trạng thái của phản ánh đó sau khi xử lý xong.

### 3.1.2.2 . Về phía sinh viên

Sau khi sinh viên được thêm mới thành công vào hệ thống, sinh viên sẽ được hệ thống cung cấp tài khoản đăng nhập bằng mã số sinh viên và có thể thay đổi mật khẩu tài khoản.

Sinh viên có thể theo dõi tất cả các đăng ký hợp đồng của mình và thấy được thông tin các bạn trong phòng đang ở.

Bên cạnh đó, sinh viên cũng thể xem hoá đơn điện nước tháng hiện tại của phòng mình.

Khi gặp vấn đề cần ban quản lý giải quyết, sinh viên có thể tạo phản ánh và gửi lên hệ thống.

### 3.1.3. Chức năng của hệ thống

Cụ thể như sau:

### 3.1.3.1 Chức năng chung

* Có thể đăng nhập và đăng xuất hệ thống bằng tài khoản cá nhân.
* Thực hiện đổi mật khẩu tài khoản cá nhân, sửa thông tin hồ sơ của sinh viên.
* Tìm kiếm, tra cứu thông tin.

### 3.1.3.2. Đối với admin

* Quản lý danh sách sinh viên, hợp đồng và các loại trạng thái đi kèm.
* Có thể quản lý dễ dàng thông tin, số lượng toà nhà và số lượng phòng trong toà.
* Dễ dàng trong khâu xử lý sinh viên chuyển phòng.
* Quản lý, nắm bắt và xử lý kịp thời các phản ánh của sinh viên.
* Tiện lợi trong quản lý, in ấn hoá đơn điện nước.

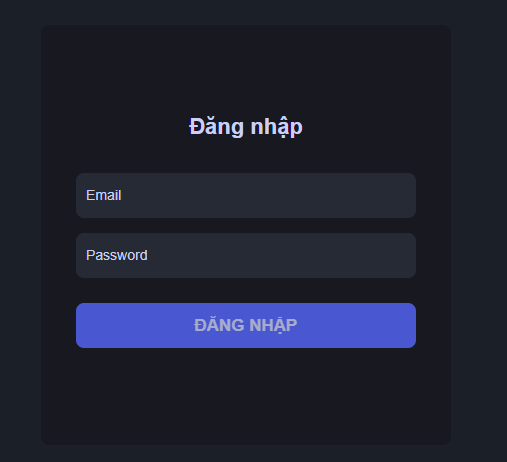
### 3.1.3.3. Đối với sinh viên

* Dễ dàng theo dõi lịch sử đăng ký hợp đồng của mình.
* Có thể xem thông tin các bạn cùng phòng và thông tin hoá đơn điện nước hàng tháng của phòng.

### 3.1.4. Một số màn hình giao diện chính của hệ thống quản lý sinh viên ký túc xá

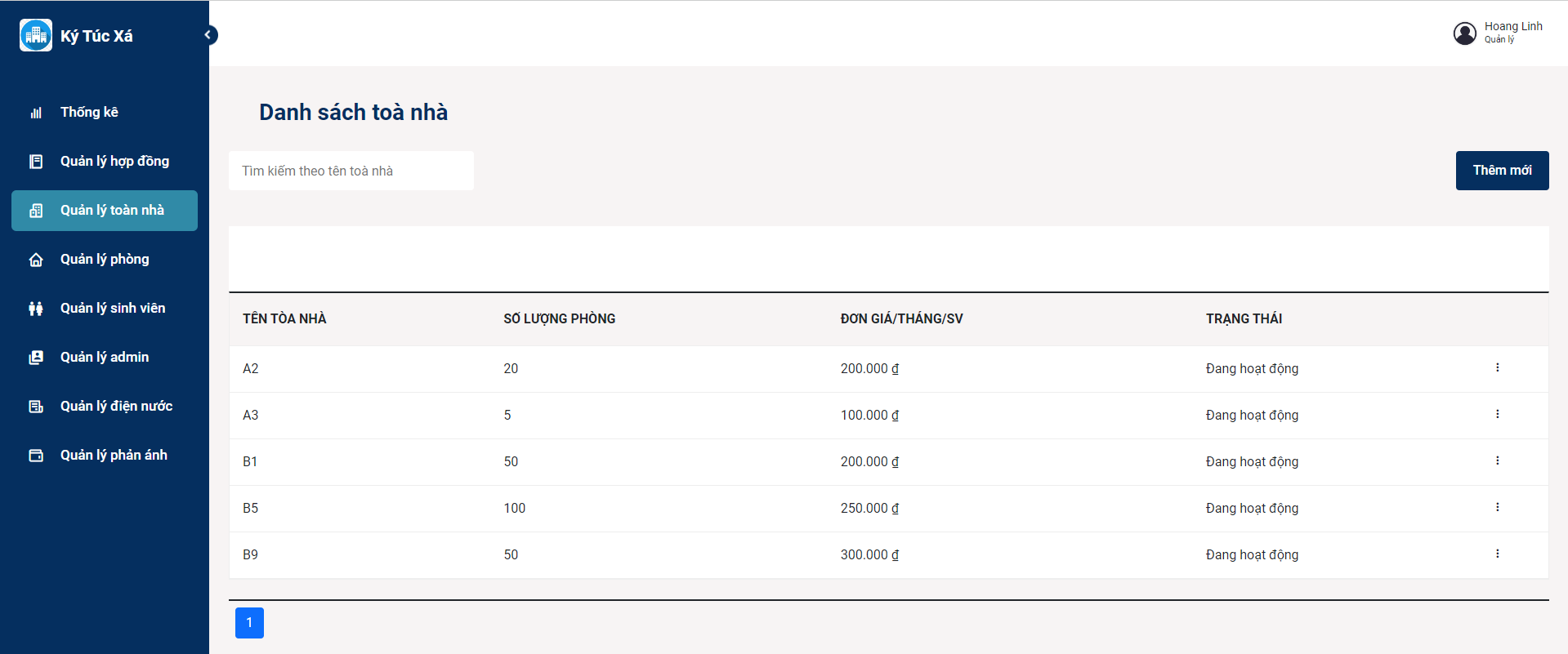
Màn hình đăng nhập: Cho phép người dùng có thể đăng nhập vào hệ thống.

Người dùng nhập email, mật khẩu đã database và nhấn nút Đăng nhập để tiến hành đăng nhập.



*Hình 3.1: Màn hình đăng nhập*

* Màn hình danh sách tòa nhà: Thông tin hiển bao gồm
* Danh sách menu quản lý
* Tài khoản admin
* Input Tìm kiếm theo tên tòa nhà
* Button Thêm mới
* Danh sách tòa nhà
* Phân trang



*Hình 3.2: Giao diện danh sách tòa nhà*

### 3.3. Sự khác nhau giữ kịch bản kiểm thử 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, để 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.

### 3.4. Kịch bản kiểm thử thủ công trong hệ thống quản lý sinh viên ký túc xá

### 3.4.1. Kịch bản kiểm thử chức năng Đăng nhập

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **KỊCH BẢN KIỂM THỬ** | |  |  |  |  |
|  |  | **Tên màn hình/Tên chức năng** | Đăng nhập |  |  |  |  |
|  |  | **Mã testcase** | ĐN |  |  |  |  |
|  |  | **Số testcase đạt (P)** | 31 |  |  |  |  |
|  |  | **Số testcase không đạt (F)** | 0 |  |  |  |  |
|  |  | **Số testcase đang xem xét (PE)** | 0 |  |  |  |  |
|  |  | **Số testcase chưa test** | 0 |  |  |  |  |
|  |  | **Tổng số testcase** | 31 |  |  |  |  |
|  |  |  |  |  |  |  |  |
| **Mã trường hợp kiểm thử** | **Mục đích kiểm thử** | **Các bước thực hiện** | **Kết quả mong muốn** | **Lần Test** | | | **Kết quả hiện tại** |
| **Lần 1** | **Lần 2** | **Lần 3** |
|  | **Kiểm tra giao diện màn Đăng nhập** | | | | | | |
| ***ĐN\_1*** | Kiểm tra giao diện chung | 1. Kiểm tra giao diện màn hình đăng nhập | 2. Giao diện hiển thị gồm các thành phần: - Fix Text: Đăng nhập - Textbox: Email - Textbox: Mật khẩu - Button: [Đăng nhập] |  |  |  | ***P*** |
| ***ĐN\_2*** | Check bố cục,căn chỉnh các field, font,size,color …  ( icon,logo,hint… nếu có ) | 1. Check bố cục,căn chỉnh các field, font,size,color .. của from "Đăng nhập" | 1.  - Căn chỉnh các field hợp lý, thẳng hàng  - Bố cục ,font,size,color hiển thị giống như trong ảnh đính kèm ở trên |  |  |  | ***P*** |
| ***ĐN\_3*** | Check xử lý của hệ thống khi nhấm "Tab' từ bàn phím | 1. Focus chuột vào text box đầu tiên của form "Đăng nhập" 2. Nhấn phím "tab" từ bán phím | 2. Con trỏ chuột sẽ dịch chuyển qua các textbox từ trái sang phải, từ trên xuống dưới |  |  |  | ***P*** |
| ***ĐN\_4*** | Check xử lý của hệ thống khi nhấm "Shift+ Tab' từ bàn phím | 1. Focus chuột vào text box cuối cùng của form "Đăng nhập" 2. Nhấn phím "shift+tab" từ bán phím | 2. Con trỏ chuột sẽ dịch chuyển qua các textbox từ phải sang trái, từ dưới lên trên |  |  |  | ***P*** |
|  | **Kiểm tra validate thông tin** | | | | | | |
|  | **Kiểm tra validate Email** | | | | | | |
| ***ĐN\_5*** | Nhập đúng địa chỉ Email hợp lệ | 1.Nhập đúng địa chỉ Email hợp lệ 2.Nhập giá trị hợp lệ vào tất cả các trường còn lại. 3. Nhấn vào nút [Đăng nhập]. | 3. Login thành công chuyển đến màn hình Quản lý tòa nhà |  |  |  | ***P*** |
| ***ĐN\_6*** | Nhập sai định dạng email | 1. Tiến hành nhập email thiếu tên miền vào field "Email " *Ví dụ:anh@gmail* | 3. Hiển thị error message dưới field "Email " với nội dung: "Sai định dạng Email " |  |  |  | ***P*** |
| ***ĐN\_7*** | 1. Tiến hành nhập email thiếu @ vào field "Email " Ví dụ*:* [*anhabc.com*](http://anhabc.com/) | 3. Hiển thị error message dưới field "Email " với nội dung: "Sai định dạng Email " |  |  |  | ***P*** |
| ***ĐN\_8*** | 1. Tiến hành nhập Email thiếu tên miền sau dấu chấm vào field "Email " *vidu : anh@abc.* | 3. Hiển thị error message dưới field "Email " với nội dung: "Sai định dạng Email " |  |  |  | ***P*** |
| ***ĐN\_9*** | 1. Tiến hành nhập Email chứa ký tự đăc biệt ( loại trừ @, \_ , . ) vào field "Email " *Ví dụ:anh$%^^^@gmail.com* | 3. Hiển thị error message dưới field "Email " với nội dung: "Sai định dạng Email " |  |  |  | ***P*** |
| ***ĐN\_10*** | 1. Tiến hành nhập Email chứa ký tự đăc biệt ( loại trừ @, \_ , . ) vào field "Email " Ví dụ:anh$%^^^@gmail.com | 3. Hiển thị error message dưới field "Email " với nội dung: "Sai định dạng Email " |  |  |  | ***P*** |
| ***ĐN\_11*** | Để trống field "Email" | 1. Tiến hành để trống field "Email" 2.Nhập giá trị hợp lệ vào tất cả các trường còn lại. 3. Nhấn vào nút [Đăng nhập]. | 3. Show thông báo: "Vui lòng nhập Email" |  |  |  | ***P*** |
| ***ĐN\_12*** | Nhập space vào trước/ sau "Email" | 1. Nnhập space vào trước và sau text trên field "Email Address" . Ví dụ : " hoangnguyen@gmail.com " 2.Nhập giá trị hợp lệ vào tất cả các trường còn lại. 3. Nhấn vào nút [Đăng nhập]. | 3. Login thành công chuyển đến màn hình Home page *>> Hệ thống sẽ tự động loại bỏ khoảng trắng trước /sau data vừa nhập* |  |  |  | ***P*** |
| ***ĐN\_13*** | Nhập Email đúng định dạng nhưng không có trong database | 1. Nhập đúng định dạng Email. Ví dụ : "hoangnguyen@gmail.com" 2.Nhập giá trị hợp lệ vào tất cả các trường còn lại. 3. Nhấn vào nút [Đăng nhập]. | 3. Hiển thị error message dưới field "Email Address" với nội dung : " Email không tồn tại" - Highlight và focus vào field bị lỗi |  |  |  | ***P*** |
|  | **Kiểm tra validate Mật Khẩu** | | | | | | |
| ***ĐN\_14*** | Check hiển thị ở field " Password" | 1. Nhập data vào field " Password" 2. Quan sát hiển thị data ở field đó | 2. Password vừa nhập có định dạng: •••••• |  |  |  | ***P*** |
| ***ĐN\_15*** | Nhập đúng giá trị Password hợp lệ | 1.Nhập đúng Password. 2.Nhập giá trị hợp lệ vào tất cả các trường còn lại. 3. Nhấn vào nút [Đăng nhập]. | 3. Login thành công chuyển đến màn hình Quản lý tòa nhà |  |  |  | ***P*** |
| ***ĐN\_16*** | Kiểm tra ký tự nhập | 1. Tiến hành nhập ký tự đặt biệt vào Password | 1. Cho phép nhập dữ liệu ký tự đặc biệt |  |  |  | ***P*** |
| ***ĐN\_17*** | 1. Tiến hành nhập ký tự dạng chữ cái vào Password | 1. Cho phép nhập dữ liệu dạng chữ |  |  |  | ***P*** |
| ***ĐN\_18*** | 1. Tiến hành nhập ký tự dạng số vào Password | 1. Cho phép nhập dữ liệu dạng số |  |  |  | ***P*** |
| ***ĐN\_19*** | Để trống field "Password" | 1. Tiến hành để trống field "Password" 2.Nhập giá trị hợp lệ vào tất cả các trường còn lại. 3. Nhấn vào nút [Đăng nhập]. | 3. Hiển thị thông báo: "Vui lòng nhập mật khẩu " |  |  |  | ***P*** |
| ***ĐN\_20*** | Nhập space vào trước/sau "Password" | 1. Nhập space vào trước và sau text trên field "Password" . Ví dụ: " hoangnguyen " 2.Nhập giá trị hợp lệ vào tất cả các trường còn lại. 3. Nhấn vào nút [Đăng nhập]. | 3.. Login thành công chuyển đến màn hình Home page. *>> Hệ thống sẽ tự động loại bỏ khoảng trắng trước /sau data vừa nhập* |  |  |  | ***P*** |
| ***ĐN\_21*** | Nhập sai Password | 1. Nhập sai Password. 2. Nhập giá trị hợp lệ vào tất cả các trường còn lại. 3. Nhấn vào nút [Đăng nhập]. | 3. Hiển thị thông báo lỗi : "Bạn đã nhập sai Email hoặc Password" -Highlight và focus vào field bị lỗi |  |  |  | ***P*** |
|  | **Kiểm tra button** | | | | | | |
| ***ĐN\_22*** | Kiêm tra trạng thái Button | 1. Kiểm tra trạng thái button | 1. Button luôn ở trạng thái enable 2. Click button thực hiện validate dữ liệu để đăng nhập |  |  |  | ***P*** |
| ***ĐN\_23*** | Kiểm tra click button [Đăng nhập] | 1. Kiểm tra click button [Đăng nhập]] | 1. Click button thực hiện validate dữ liệu để đăng nhập |  |  |  | ***P*** |
|  | **Kiểm tra chức năng đăng nhập** | | | | | | |
| ĐN\_24 | Kiểm tra đăng nhập không thành công | **TH sai Email** 1. Tại màn hình đăng nhập, nhập Email không tồn tại trong db 2. Nhập mật khẩu hợp lệ 3. Click "Đăng nhập" | 2. Show thông báo: "Email hoặc mật khẩu không đúng" |  |  |  | ***P*** |
| ĐN\_25 | **TH sai Password** 1. Tại màn hình đăng nhập, nhập Email tồn tại trong db 2. Nhập sai mật khẩu 3. Click "Đăng nhập" | 2. Show thông báo: "Email hoặc mật khẩu không đúng" |  |  |  | ***P*** |
| ĐN\_26 | **TH sai Email và passwword** 1. Tại màn hình đăng nhập, nhập Email và mật khẩu không tồn tại trong database 2. Click "Đăng nhập" | 2. Show thông báo: "Đăng nhập thất bại" |  |  |  | ***P*** |
| ĐN\_27 | **TH bỏ trống Email** 1. Tại màn hình đăng nhập, bỏ trống Email  2. Nhập mật khẩu hợp lệ 3. Click "Đăng nhập" | 2. Show cả 2 thông báo: "Đăng nhập thất bại" và "Vui lòng nhập Email" |  |  |  | ***P*** |
| ĐN\_28 | **TH bỏ trống Password** 1. Tại màn hình đăng nhập, nhập Email tồn tại trong db 2. Bỏ trống mật khẩu 3. Click "Đăng nhập" | 2. Show cả 2 thông báo: "Đăng nhập thất bại" và "Vui lòng nhập Password" |  |  |  | ***P*** |
| ĐN\_29 | **TH bỏ trống Email và Password** 1. Tại màn hình đăng nhập, bỏ trống Email  2. Nhập mật khẩu hợp lệ 3. Click "Đăng nhập" | 2. Show cả 3 thông báo: "Đăng nhập thất bại", "Vui lòng nhập Password","Vui lòng nhập Email" |  |  |  | ***P*** |
| ĐN\_30 | **TH server lỗi** 1. Tại màn hình đăng nhập, nhập Tên đăng nhập và mật khẩu tồn tại trong db 2. Click "Đăng nhập" | 2. Show thông báo: “Đăng nhập thất bại. Vui lòng kiểm tra lại kết nối” |  |  |  | ***P*** |
| ĐN\_31 | Kiểm tra đăng nhập thành công | **TH Email và passwword là một cặp tồn tại trong database** 1. Tại màn hình đăng nhập, nhập Email và mật khẩu không tồn tại trong database 2. Click "Đăng nhập" | 2. Chuyển đến màn hình Danh sách tòa nhà |  |  |  | ***P*** |

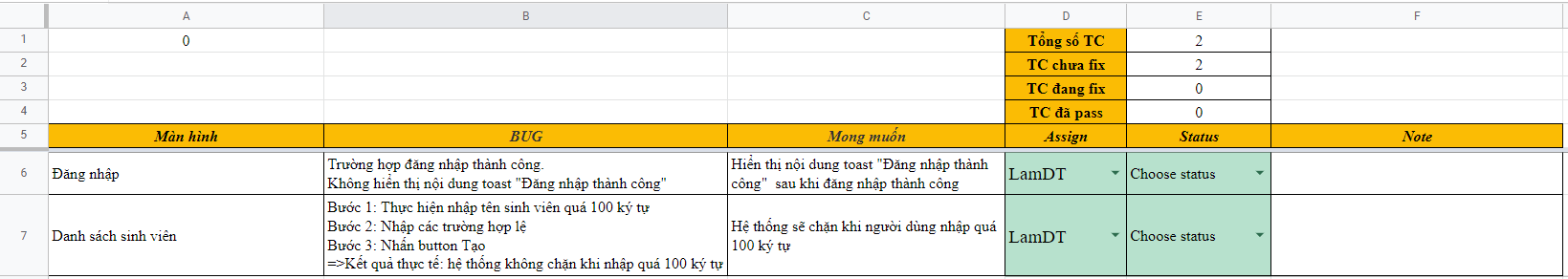
**3.4.2. Kịch bản kiểm thử chức năng thêm mới sinh viên**

|  |  |
| --- | --- |
| **KỊCH BẢN KIỂM THỬ** | |
| **Tên màn hình/Tên chức năng** | Thêm mới sinh viên |
| **Mã testcase** | QLSV |
| **Số testcase đạt (P)** | 62 |
| **Số testcase không đạt (F)** | 17 |
| **Số testcase đang xem xét (PE)** | 1 |
| **Số testcase chưa test** | 0 |
| **Tổng số testcase** | 80 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Mã trường hợp kiểm thử** | **Mục đích kiểm thử** | **Các bước thực hiện** | **Kết quả mong muốn** | **Lần Test** | | | **Kết quả hiện tại** |  |
| **Lần 1** | **Lần 2** | **Lần 3** |  |
|  | **Kiểm tra giao diện màn Phát hành vé** | | | | | | |  |
| ***QLSV\_1*** | Kiểm tra giao diện chung | 1. Kiểm tra giao diện màn hình quản lý sinh viên | 2. Giao diện hiển thị gồm các thành phần: - Danh sách quản lý - Tài khoản admin - Fix Text Title: Danh sách sinh viên - Button: Thêm mới - Bảng Thông tin danh sách sinh viên - Phân trang |  |  |  | ***P*** |  |
| ***QLSV\_2*** | Check bố cục,căn chỉnh các field, font,size,color …  ( icon,logo,hint… nếu có ) | 1. Check bố cục,căn chỉnh các field, font,size,color .. của from "quản lý sinh viên" | 1.  - Căn chỉnh các field hợp lý, thẳng hàng  - Bố cục ,font,size,color hiển thị giống như trong ảnh đính kèm ở trên |  |  |  | ***P*** |  |
| ***QLSV\_3*** | Check xử lý của hệ thống khi nhấm "Tab' từ bàn phím | 1. Focus chuột vào text box đầu tiên của form "quản lý sinh viên" 2. Nhấn phím "tab" từ bán phím | 2. Con trỏ chuột sẽ dịch chuyển qua các textbox từ trái sang phải, từ trên xuống dưới |  |  |  | ***P*** |  |
| ***QLSV\_4*** | Check xử lý của hệ thống khi nhấm "Shift+ Tab' từ bàn phím | 1. Focus chuột vào text box cuối cùng của form "quản lý sinh viên" 2. Nhấn phím "shift+tab" từ bán phím | 2. Con trỏ chuột sẽ dịch chuyển qua các textbox từ phải sang trái, từ dưới lên trên |  |  |  | ***P*** |  |
|  | Kiểm tra validate thông tin | | | | | | |  |
|  | **Kiểm tra validate Mã sinh viên** | | | | | | |  |
|  | Kiểm tra hiển thị mặc định trường Mã sinh viên | 1. Kiểm tra hiển thị mặc định trường Mã sinh viên | 1. Mặc định hiển thị rỗng. Hiển thị hintext: Nhập mã sinh viên |  |  |  | ***P*** |  |
| ***QLSV\_6*** | Kiểm tra bỏ trống mã sinh viên | 1. Thực hiện bỏ trống mã sinh viên 2. Thực hiện nhập hợp lệ các trường khác 3. Click Button "Tạo" | 1. Show thông báo "Vui lòng nhập mã sinh viên" |  |  |  | ***P*** |  |
| ***QLSV\_7*** | Kiểm tra dữ liệu cho phép nhập | 1. Nhập dữ liệu chữ cái 2. Thực hiện nhập hợp lệ các trường khác  3. Click Button "Tạo" | 1. Show thông báo "Thêm mới thành công" |  |  |  | ***P*** |  |
| ***QLSV\_8*** | 1. Nhập dữ liệu số 2. Thực hiện nhập hợp lệ các trường khác  3. Click Button "Tạo" | 1. Show thông báo "Thêm mới thành công" |  |  |  | ***P*** |  |
| ***QLSV\_9*** | 1. Nhập ký tự đặc biệt 2. Thực hiện nhập hợp lệ các trường khác  3. Click Button "Tạo" | 1. Show thông báo "Thêm mới thành công" |  |  |  | ***P*** |  |
| ***QLSV\_10*** | Kiểm tra maxlength (Max 100 ký tự) | Trường hợp nhập 101 ký tự 1. Nhập 101 ký tự | 1. Chặn không cho nhập quá 100 ký tự |  |  |  | ***F*** |  |
| ***QLSV\_11*** | Trường hợp nhập 99 ký tự 1. Nhập 99 ký tự 2. Thực hiện nhập hợp lệ các trường khác  3. Click Button "Tạo" | 1. Show thông báo "Thêm mới thành công" |  |  |  | ***P*** |  |
|  | **Kiểm tra validate Họ tên sinh viên** | | | | | | |  |
| ***QLSV\_12*** | Kiểm tra hiển thị mặc định trường Họ tên | 1. Kiểm tra hiển thị mặc định trường Họ tên | 1. Mặc định hiển thị rỗng. Hiển thị hintext: Nhập họ tên sinh viên |  |  |  | ***P*** |  |
| ***QLSV\_13*** | Kiểm tra bỏ trống Họ tên sinh viên | 1. Thực hiện bỏ trống tên sinh viên 2. Thực hiện nhập hợp lệ các trường khác 3. Click Button "Tạo" | 1. Show thông báo "Vui lòng nhập nhập sinh viên" |  |  |  | ***P*** |  |
| ***QLSV\_14*** | Kiểm tra dữ liệu cho phép nhập | 1. Nhập dữ liệu chữ cái 2. Thực hiện nhập hợp lệ các trường khác  3. Click Button "Tạo" | 1. Show thông báo "Thêm mới thành công" |  |  |  | ***P*** |  |
| ***QLSV\_15*** | 1. Nhập dữ liệu số 2. Thực hiện nhập hợp lệ các trường khác  3. Click Button "Tạo" | 1. Show thông báo "Thêm mới thành công" |  |  |  | ***P*** |  |
| ***QLSV\_16*** | 1. Nhập ký tự đặc biệt 2. Thực hiện nhập hợp lệ các trường khác  3. Click Button "Tạo" | 1. Show thông báo "Thêm mới thành công" |  |  |  | ***P*** |  |
| ***QLSV\_17*** | Kiểm tra maxlength (Max 100 ký tự) | Trường hợp nhập 101 ký tự 1. Nhập 101 ký tự | 1. Chặn không cho nhập quá 100 ký tự |  |  |  | ***F*** |  |
| ***QLSV\_18*** | Trường hợp nhập 99 ký tự 1. Nhập 99 ký tự 2. Thực hiện nhập hợp lệ các trường khác  3. Click Button "Tạo" | 1. Show thông báo "Thêm mới thành công" |  |  |  | ***P*** |  |
|  | **Kiểm tra validate Lớp** | | | | | | |  |
| ***QLSV\_19*** | Kiểm tra hiển thị mặc định trường Lớp sinh viên | 1. Kiểm tra hiển thị mặc định trường lớp | 1. Mặc định hiển thị rỗng. Hiển thị hintext: Nhập lớp sinh viên |  |  |  | ***P*** |  |
| ***QLSV\_20*** | Kiểm tra bỏ trống lớp sinh viên | 1. Thực hiện bỏ trống lớp sinh viên 2. Thực hiện nhập hợp lệ các trường khác 3. Click Button "Tạo" | 1. Show thông báo "Vui lòng nhập nhập sinh viên" |  |  |  | ***P*** |  |
| ***QLSV\_21*** | Kiểm tra dữ liệu cho phép nhập | 1. Nhập dữ liệu chữ cái 2. Thực hiện nhập hợp lệ các trường khác  3. Click Button "Tạo" | 1. Show thông báo "Thêm mới thành công" |  |  |  | ***P*** |  |
| ***QLSV\_22*** | 1. Nhập dữ liệu số 2. Thực hiện nhập hợp lệ các trường khác  3. Click Button "Tạo" | 1. Show thông báo "Thêm mới thành công" |  |  |  | ***P*** |  |
| ***QLSV\_23*** | 1. Nhập ký tự đặc biệt 2. Thực hiện nhập hợp lệ các trường khác  3. Click Button "Tạo" | 1. Show thông báo "Thêm mới thành công" |  |  |  | ***P*** |  |
| ***QLSV\_24*** | Kiểm tra maxlength (Max 100 ký tự) | Trường hợp nhập 101 ký tự 1. Nhập 101 ký tự | 1. Chặn không cho nhập quá 100 ký tự |  |  |  | ***F*** |  |
| ***QLSV\_25*** | Trường hợp nhập 99 ký tự 1. Nhập 99 ký tự 2. Thực hiện nhập hợp lệ các trường khác 3. Click Button "Tạo" | 1. Show thông báo "Thêm mới thành công" |  |  |  | ***P*** |  |
|  | **Kiểm tra validate Khóa** | | | | | | |  |
| ***QLSV\_26*** | Kiểm tra hiển thị mặc định trường khóa | 1. Kiểm tra hiển thị mặc định trường khóa | 1. Mặc định hiển thị rỗng. Hiển thị hintext: Nhập khóa sinh viên |  |  |  | ***P*** |  |
| ***QLSV\_27*** | Kiểm tra bỏ trống Khóa sinh viên | 1. Thực hiện bỏ trống khóa sinh viên 2. Thực hiện nhập hợp lệ các trường khác 3. Click Button "Tạo" | 1. Show thông báo "Vui lòng nhập khóa sinh viên" |  |  |  | ***P*** |  |
| ***QLSV\_28*** | Kiểm tra dữ liệu cho phép nhập | 1. Nhập dữ liệu chữ cái 2. Thực hiện nhập hợp lệ các trường khác  3. Click Button "Tạo" | 1. Show thông báo "Khóa không được phép nhập chữ cái, ký tự đặc biệt" |  |  |  | ***F*** |  |
| ***QLSV\_29*** | 1. Nhập dữ liệu số 2. Thực hiện nhập hợp lệ các trường khác  3. Click Button "Tạo" | 1. Show thông báo "Thêm mới thành công" |  |  |  | ***P*** |  |
| ***QLSV\_30*** | 1. Nhập ký tự đặc biệt 2. Thực hiện nhập hợp lệ các trường khác  3. Click Button "Tạo" | 1. Show thông báo "Khóa không được phép nhập chữ cái, ký tự đặc biệt" |  |  |  | ***F*** |  |
| ***QLSV\_31*** | Kiểm tra maxlength (Max 50 ký tự) | Trường hợp nhập 51 ký tự 1. Nhập 51 ký tự | 1. Chặn không cho nhập quá 50 ký tự |  |  |  | ***F*** |  |
| ***QLSV\_32*** | Trường hợp nhập 50 ký tự 1. Nhập 50 ký tự 2. Thực hiện nhập hợp lệ các trường khác 3. Click Button "Tạo" | 1. Show thông báo "Thêm mới thành công" |  |  |  | ***P*** |  |
|  | **Kiểm tra validate Địa chỉ** | | | | | | |  |
| ***QLSV\_33*** | Kiểm tra hiển thị mặc định trường Địa chỉ | 1. Kiểm tra hiển thị mặc định trường Địa chỉ | 1. Mặc định hiển thị rỗng. Hiển thị hintext: Nhập địa chỉ sinh viên |  |  |  | ***P*** |  |
| ***QLSV\_34*** | Kiểm tra bỏ trống Địa chỉ sinh viên | 1. Thực hiện bỏ trống Địa chỉ sinh viên 2. Thực hiện nhập hợp lệ các trường khác 3. Click Button "Tạo" | 1. Show thông báo "Vui lòng nhập địa chỉ sinh viên" |  |  |  | ***P*** |  |
| ***QLSV\_35*** | Kiểm tra dữ liệu cho phép nhập | 1. Nhập dữ liệu chữ cái 2. Thực hiện nhập hợp lệ các trường khác  3. Click Button "Tạo" | 1. Show thông báo "Thêm mới thành công" |  |  |  | ***P*** |  |
| ***QLSV\_36*** | 1. Nhập dữ liệu số 2. Thực hiện nhập hợp lệ các trường khác 3. Click Button "Tạo" | 1. Show thông báo "Thêm mới thành công" |  |  |  | ***P*** |  |
| ***QLSV\_37*** | 1. Nhập ký tự đặc biệt 2. Thực hiện nhập hợp lệ các trường khác  3. Click Button "Tạo" | 1. Show thông báo "Thêm mới thành công" |  |  |  | ***P*** |  |
| ***QLSV\_38*** | Kiểm tra maxlength (Max 200 ký tự) | Trường hợp nhập 201 ký tự 1. Nhập 201 ký tự | 1. Chặn không cho nhập quá 200 ký tự |  |  |  | ***F*** |  |
| ***QLSV\_39*** | Trường hợp nhập 200 ký tự 1. Nhập 200 ký tự 2. Thực hiện nhập hợp lệ các trường khác 3. Click Button "Tạo" | 1. Show thông báo "Thêm mới thành công" |  |  |  | ***P*** |  |
|  | **Kiểm tra validate Ngày sinh** | | | | | | |  |
| ***QLSV\_40*** | Kiểm tra hiển thị mặc định trường Ngày sinh | 1. Kiểm tra hiển thị mặc định trường Ngày sinh | 1. Hiển thị ngày sinh ctheo định dạng mm/dd/yyyy |  |  |  | ***P*** |  |
| ***QLSV\_41*** | Kiểm tra hiển thị khi click calendar Ngày sinh | 1. click calendar Ngày sinh | 1. Hiển thị calendar cho phép người dùng chọn |  |  |  | ***P*** |  |
| ***QLSV\_42*** | 1. Kiểm tra hiển thị ngày được chọn | 1. chọn ngày tháng năm trong calendar | 1. Hiển thị ngày được chọn theo định dạng mm/dd/yyyy |  |  |  | ***P*** |  |
| ***QLSV\_43*** | Kiểm tra nhập ký tự chữ cái | 1.nhập ký tự chữ cái vào trường ngày sinh | 1. Không cho phép nhâp ký tự chữ cái |  |  |  | ***P*** |  |
| ***QLSV\_44*** | Kiểm tra nhập ký tự đặc biệt | 1.nhập ký tự đặc biệt vào trường ngày sinh | 1. Không cho phép nhâp ký tự đặc biệt |  |  |  | ***P*** |  |
| ***QLSV\_45*** | Kiểm tra nhập sai định dạng ngày sinh | 1. Nhập sai định dạng ngày sinh Vd:dd/mm/yyyy - 25/01/2000 2. Thực hiện nhập hợp lệ các trường khác 3. Click Button "Tạo" | 1. Show thông báo "Sai định dạng ngày sinh" |  |  |  | ***F*** |  |
| ***QLSV\_46*** | Kiểm tra chọn ngày tháng năm lớn hơn ngày hiện tại tương lại | 1. chọn ngày tháng năm lớn hơn ngày hiện tại tương lại 2. Thực hiện nhập hợp lệ các trường khác  3. Click Button "Tạo" | 1.Show thông báo "Ngày tháng năm không được phép lớn hơn ngày hiện tại" |  |  |  | ***F*** |  |
| ***QLSV\_47*** | Kiểm tra hiển thị ngày sinh khi chọn lại ngày mới | 1. Thực hiện chọn ngày VD: 01/01/2000 2. Thực hiện chọn lại ngày VD: 01/02/2002 | 1. Hiển thị ngày được chọn theo định dạng mm/dd/yyyy |  |  |  | ***P*** |  |
|  | **Kiểm tra validate Số điện thoại** | | | | | | |  |
| ***QLSV\_48*** | Kiểm tra hiển thị mặc định trường SĐT | 1. Kiểm tra hiển thị mặc định trường SĐT | 1. Mặc định hiển thị rỗng. Hiển thị hintext: Nhập số điện thoại |  |  |  | ***P*** |  |
| ***QLSV\_49*** | Kiểm tra bỏ trống số điện thoại | 1. Thực hiện bỏ trống sđt 2. Thực hiện nhập hợp lệ các trường khác 3. Click Button "Tạo" | 1. Show thông báo "Vui lòng nhập số điện thoại" |  |  |  | ***P*** |  |
| ***QLSV\_50*** | Kiểm tra dữ liệu cho phép nhập | 1. Nhập dữ liệu chữ cái 2. Thực hiện nhập hợp lệ các trường khác 3. Click Button "Tạo" | 1. Show thông báo"Số điện thoại không được phép nhập chữ cái, ký tự đặc biệt" |  |  |  | ***F*** |  |
| ***QLSV\_51*** | 1. Nhập dữ liệu số 2. Thực hiện nhập hợp lệ các trường khác 3. Click Button "Tạo" | 1. Show thông báo "Thêm mới thành công" |  |  |  | ***P*** |  |
| ***QLSV\_52*** | 1. Nhập ký tự đặc biệt 2. Thực hiện nhập hợp lệ các trường khác  3. Click Button "Tạo" | 1. Show thông báo"Số điện thoại không được phép nhập chữ cái, ký tự đặc biệt" |  |  |  | ***F*** |  |
| ***QLSV\_53*** | Kiểm tra maxlength (Max 20 ký tự) | Trường hợp nhập 21 ký tự 1. Nhập 21 ký tự 2. Thực hiện nhập hợp lệ các trường khác  3. Click Button "Tạo" | 1. Show thông báo "Số điện thoại phải nhập bé hơn hoặc bằng 20 ký tự" |  |  |  | ***F*** |  |
| ***QLSV\_54*** | Trường hợp nhập 20 ký tự 1. Nhập 20 ký tự 2. Thực hiện nhập hợp lệ các trường khác 3. Click Button "Tạo" | 1. Show thông báo "Thêm mới thành công" |  |  |  | ***P*** |  |
| ***QLSV\_55*** | Kiểm tra độ dài tối thiểu (Min 9 ký tự) | Trường hợp nhập 8 ký tự 1. Nhập 8 ký tự 2. Thực hiện nhập hợp lệ các trường khác  3. Click Button "Tạo" | 1. Show thông báo "Số điện thoại phải nhập lớn hơn hoặc bằng 9 ký tự" |  |  |  | ***P*** |  |
| ***QLSV\_56*** | Trường hợp nhập 9 ký tự 1. Nhập 9 ký tự 2. Thực hiện nhập hợp lệ các trường khác 3. Click Button "Tạo" | 1. Show thông báo "Thêm mới thành công" |  |  |  | ***P*** |  |
|  | **Kiểm tra radiobutton Giới tính** | | | | | | |  |
| ***QLSV\_57*** | Kiểm tra mặc định tích chọn radiobutton | 1. Kiểm tra mặc định tích chọn radiobutton | 1. Mặc định tích chọn radiobutton giới tính Nam |  |  |  | ***P*** |  |
| ***QLSV\_58*** | Kiểm tra hiển khi click tích chọn radiobutton giới tính nữ | 1. Kiểm tra hiển khi click tích chọn radiobutton giới tính nữ | 1. Giới tính nữ được tích chọn. radiobutton giới tính Nam bỏ tích chọn |  |  |  | ***P*** |  |
|  | **Kiểm tra validate thông tin bố** | | | | | | |  |
| ***QLSV\_59*** | Kiểm tra hiển thị mặc định trường thông tin bố | 1. Kiểm tra hiển thị mặc định trường thông tin bố sinh viên | 1. Mặc định hiển thị rỗng. Hiển thị hintext: Nhập thông tin bố sinh viên" |  |  |  | ***F*** |  |
| ***QLSV\_60*** | Kiểm tra bỏ trống thông tin bố | 1. Thực hiện bỏ trống thông tin bố sinh viên 2. Thực hiện nhập hợp lệ các trường khác 3. Click Button "Tạo" | 1. Show thông báo "Vui lòng nhập thông tin bố sinh viên" |  |  |  | ***P*** |  |
| ***QLSV\_61*** | Kiểm tra dữ liệu cho phép nhập | 1. Nhập dữ liệu chữ cái 2. Thực hiện nhập hợp lệ các trường khác  3. Click Button "Tạo" | 1. Show thông báo "Thêm mới thành công" |  |  |  | ***P*** |  |
| ***QLSV\_62*** | 1. Nhập dữ liệu số 2. Thực hiện nhập hợp lệ các trường khác  3. Click Button "Tạo" | 1. Show thông báo "Thêm mới thành công" |  |  |  | ***P*** |  |
| ***QLSV\_63*** | 1. Nhập ký tự đặc biệt 2. Thực hiện nhập hợp lệ các trường khác  3. Click Button "Tạo" | 1. Show thông báo "Thêm mới thành công" |  |  |  | ***P*** |  |
| ***QLSV\_64*** | Kiểm tra maxlength (Max 100 ký tự) | Trường hợp nhập 101 ký tự 1. Nhập 101 ký tự | 1. Chặn không cho nhập quá 100 ký tự |  |  |  | ***F*** |  |
| ***QLSV\_65*** | Trường hợp nhập 99 ký tự 1. Nhập 99 ký tự 2. Thực hiện nhập hợp lệ các trường khác  3. Click Button "Tạo" | 1. Show thông báo "Thêm mới thành công" |  |  |  | ***P*** |  |
|  | **Kiểm tra validate thông tin mẹ** | | | | | | |  |
| ***QLSV\_66*** | Kiểm tra hiển thị mặc định trường thông tin mẹ | 1. Kiểm tra hiển thị mặc định trường thông tin mẹ sinh viên | 1. Mặc định hiển thị rỗng. Hiển thị hintext: Nhập thông tin mẹ sinh viên" |  |  |  | ***F*** |  |
| ***QLSV\_67*** | Kiểm tra bỏ trống thông tin mẹ | 1. Thực hiện bỏ trống thông tin mẹ sinh viên 2. Thực hiện nhập hợp lệ các trường khác 3. Click Button "Tạo" | 1. Show thông báo "Vui lòng nhập thông tin mẹ sinh viên" |  |  |  | ***P*** |  |
| ***QLSV\_68*** | Kiểm tra dữ liệu cho phép nhập | 1. Nhập dữ liệu chữ cái 2. Thực hiện nhập hợp lệ các trường khác  3. Click Button "Tạo" | 1. Show thông báo "Thêm mới thành công" |  |  |  | ***P*** |  |
| ***QLSV\_69*** | 1. Nhập dữ liệu số 2. Thực hiện nhập hợp lệ các trường khác  3. Click Button "Tạo" | 1. Show thông báo "Thêm mới thành công" |  |  |  | ***P*** |  |
| ***QLSV\_70*** | 1. Nhập ký tự đặc biệt 2. Thực hiện nhập hợp lệ các trường khác  3. Click Button "Tạo" | 1. Show thông báo "Thêm mới thành công" |  |  |  | ***P*** |  |
| ***QLSV\_71*** | Kiểm tra maxlength (Max 100 ký tự) | Trường hợp nhập 101 ký tự 1. Nhập 101 ký tự | 1. Chặn không cho nhập quá 100 ký tự |  |  |  | ***F*** |  |
| ***QLSV\_72*** | Trường hợp nhập 99 ký tự 1. Nhập 99 ký tự 2. Thực hiện nhập hợp lệ các trường khác 3. Click Button "Tạo" | 1. Show thông báo "Thêm mới thành công" |  |  |  | ***P*** |  |
|  | **Kiểm tra button** | | | | | | |  |
| ***QLSV\_73*** | Kiểm tra button [Tạo] | 1. Kiểm tra trạng thái button [Tạo] | 1. Luôn enable button [Tạo] |  |  |  | ***P*** |  |
| ***QLSV\_74*** | Kiểm tra button [Đóng] | 1. Kiểm tra trạng thái button [Đóng] | 1. Luôn enable button [Đóng] |  |  |  | ***P*** |  |
|  | **Kiểm tra chức năng thêm mới sinh viên** | | | | | | |  |
| ***QLSV\_75*** | Kiểm tra thêm mới sinh viên thất bại | 1. Bỏ trống 1 trường bắt buộc 2. Thực hiện nhập hợp lệ các trường khác  3. Click Button "Tạo" | 3. Show toast thông báo theo trường bị bỏ trống |  |  |  | ***P*** |  |
| ***QLSV\_76*** | 1. Nhập sai định dạng ngày sinh 2. Thực hiện nhập hợp lệ các trường khác 3. Click Button "Tạo" | 1. Show thông báo "Nhập sai định dạng ngày sinh" |  |  |  | ***P*** |  |
|  | 1. Nhập ký tự đặc biệt text input khóa sih viên 2. Thực hiện nhập hợp lệ các trường khác 3. Click Button "Tạo" | 1. Show thông báo "Nhập sai định dạng ngày sinh" |  |  |  | ***F*** |  |
|  | 1. Nhập ký tự chữ cái vào text input Khóa 2. Thực hiện nhập hợp lệ các trường khác 3. Click Button "Tạo" | 1. Show thông báo "Nhập sai định dạng ngày sinh" |  |  |  | ***P*** |  |
| ***QLSV\_79*** | Pre: Do lỗi server Step: Tại màn hình Thiết lập thông tin: 1. Nhập đầy đủ các thông tin bắt buộc  2. Click [Thiết lập vé] 3. Kiểm tra hiển thị | 3. Show popup thông báo gồm các thành phần: - Title popup: Thông báo - Nội dung thông báo: “Vui lòng kiểm tra kết nối” - Button: [Đóng]. Click [Đóng] tắt popup và ở màn hình hiện tại |  |  |  | ***PE*** |  |
| ***QLSV\_80*** | Kiểm tra thêm mới sinh viên thành công | Step: Tại màn hình Thiết lập thông tin: 1. Nhập đầy đủ các thông tin bắt buộc hợp lệ 2. Click [Tạo] 3. Kiểm tra hiển thị khi gọi api thành công | 3. - Nội dung thông báo: "Thêm mới thành thành công" - |  |  |  | ***P*** |  |

**3.4.3 Tiến hành kiểm thử và Logbug**

Khi thực hiện kiểm thử thì test luồng chính trước, test các chức năng chính trước, các luồng phụ và giao diện test sau luồng chính. Trong quá trình test, tìm thấy bug thì log lên tool quản lý lỗi. Lỗi của lập trình viên nào thì assign cho người đó. Khi nào lập trình viên fix xong thì test lại. Bug chưa hoàn thành mở lại trạng thái bug thành Re-open. Bug sửa hoàn thành thì Close bug



Hình 3.3: Ảnh file loge bug

### 3.5. Kịch bản kiểm thử tự động trong hệ thống quản lý sinh viên ký túc xá

Cách chạy tập lệnh kiểm tra Selenium WebDriver bằng ngôn ngữ Lập trình Python.

### 3.5.1. Kịch bản kiểm thử tự động chức năng đăng nhập

TestCase và data test

TestCase: thực hiện viết script trên trang đăng nhập

“http://localhost:3000/admin/login” và sử dụng trình duyệt Chrome để chạy test:

TestCase: thực hiện viết script trên trang đăng nhập và sử dụng trình duyệt Chrome để chạy test:

Step 1: Truy cập trang http://localhost:3000/admin/login

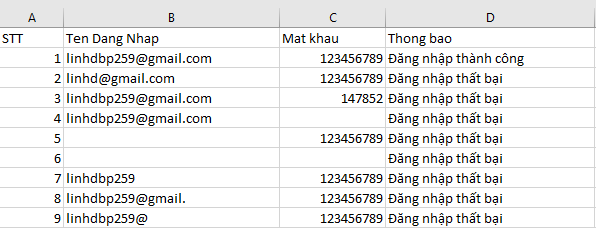
Step 2: Nhập Email vào textbox Email

Step 3: Nhập Password vào textbox Password

Step 4: Nhấn button Đăng nhập

Step 5: Kiểm tra kết quả

Data test: Chuẩn bị data test trong file CSV theo mẫu sau:



Kịch bản thử nghiệm:

from selenium import webdriver

from selenium.webdriver.common.by import By

import time

import random

import pandas as pd

import unittest

import HTMLTestRunner

def Setup\_driver():

    global driver

    driver = webdriver.Chrome()

    #driver = webdriver.Edge()

    driver.get("http://localhost:3000/admin/login")

    driver.maximize\_window()

def Sleep():

    time.sleep(0.5)

def Test():

    #Setup Data

    countpass = countfail = 0

    result = "pass";

    file = open('./Data/DataLogin11.csv', encoding='utf')

    file.readline().strip()

    type = file.readline().strip()

    #Test

    while type != '':

        list = type.split(',')

        z = 0

        a = 1

        b = a + 1

        c = b + 1

        email = driver.find\_element(By.XPATH, '//\*[@id="email"]')

        email.clear()

        email.send\_keys(list[a])

        time.sleep(0.3)

        passw = driver.find\_element(By.XPATH, '//\*[@id="password"]')

        passw.clear()

        passw.send\_keys(list[b])

        time.sleep(0.3)

        driver.find\_element(By.XPATH, '//\*[@id="submit"]').click()

        time.sleep(1.3)

        url = driver.current\_url

        if url == "http://localhost:3000/admin/login":

            notification = driver.find\_element(By.XPATH, '/html/body/div').text

        time.sleep(1.3)

        #Report

        if url == "http://localhost:3000/admin/building":

            countpass += 1

            print(list[z],"pass")

            #driver.find\_element(by=By.XPATH, value=list[c]).click()

            #time.sleep(1)

            driver.find\_element(by=By.XPATH, value="/html/body/section/div[1]/div[1]/div").click()

            Sleep()

            driver.find\_element(by=By.XPATH, value="/html/body/section/div[1]/div[2]").click()

            time.sleep(1)

        elif str(notification) == list[c] and url == "http://localhost:3000/admin/login":

            countpass += 1

            print(list[z], "pass")

            # driver.find\_element(by=By.XPATH, value=list[c]).click()

            time.sleep(1)

        else:

            countfail += 1

            print(list[z], "fail")

            result = "fail"

            #print("Mô tả: ", list[description])

            print("username: ", list[a])

            print("password: ", list[b])

            Sleep()

            print("Thông báo thực tế: ", notification)

            Sleep()

            print("Thông báo mong muốn: ", list[c])

            Sleep()

        type = file.readline().strip()

        a +1

        z +1

    print('Tổng TestCase:',countfail+countpass,'-', 'Pass:',countpass,'-', 'Fail:',countfail)

    file = open('./Data/DataLogin11.csv', mode='r',encoding='utf-8-sig')

    file\_new = open('./Data/DataLogin11\_new.csv', mode='w', encoding='utf-8-sig')

    header = file.readline()

    file\_new.write(header.strip() + ",Thông báo thực tế, Kết quả\n")

    row = file.readline()

    while row != "":

        row\_list = row.split(",") # mảng ,  split chia cách

        row\_new = row.strip() + "," + notification + "," + result+ "\n"

        file\_new.write(row\_new)

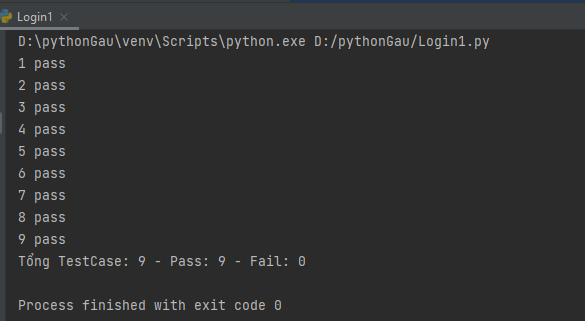
        row = file.readline()

Setup\_driver()

Test()

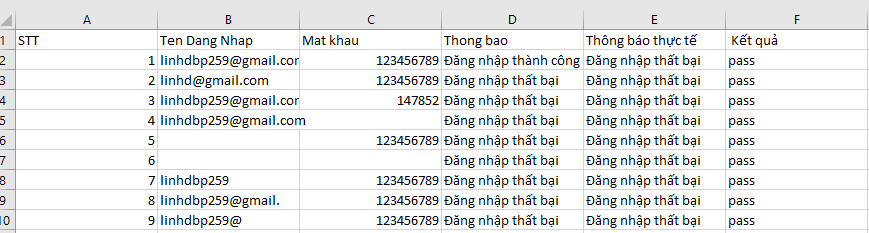
driver.quit()

**Kết quả sau khi thực hiện chạy test case:**



* Tổng số test case chạy: 9
* Tổng số trường hợp kiểm thử thành công: 9
* Tổng số trường hợp kiểm thử không thành công: 0

**Kết quả export qua file csv sau khi đã có kết quả test:**

****

### 3.5.2. Kịch bản kiểm thử tự động chức năng thêm mới sinh viên

TestCase: thực hiện viết script trên trang đăng nhập

“http://localhost:3000/admin/login” và sử dụng trình duyệt Chrome để chạy test:

TestCase: thực hiện viết script trên trang đăng nhập và sử dụng trình duyệt Chrome để chạy test:

Step 1: Truy cập trang http://localhost:3000/admin/login

Step 2: Nhập Mã sinh viên vào textbox Mã sinh viên

Step 3: Nhập Tên sinh viên vào textbox Họ tên

Step 4: Nhập Lớp vào textbox Lớp

Step 5: Nhập Khóa vào textbox Khóa

Step 6: Nhập Ngày sinh vào textbox Ngày sinh

Step 7: Nhập Thông tin bố của sinh viên vào textbox Thông tin bố

Step 8: Nhập Thông tin mẹ của sinh viên vào textbox Thông tin mẹ

Step 9: Nhấn button Tạo

Step 10: Kiểm tra kết quả

Data test: Chuẩn bị data test trong file CSV theo mẫu sau:

Kịch bản thử nghiệm:

from selenium import webdriver

from selenium.webdriver.common.by import By

import time

def Setup\_driver():

    global driver

    driver = webdriver.Chrome()

    # driver = webdriver.Edge()

    driver.get("http://localhost:3000/admin/login")

    driver.maximize\_window()

    time.sleep(1.2)

    username = "linhdbp259@gmail.com"

    password = "123456789"

    driver.maximize\_window()

    driver.find\_element("id", "email").send\_keys(username)

    # find password input field and insert password as well

    driver.find\_element("id", "password").send\_keys(password)

    # click login button

    driver.find\_element("id", "submit").click()

    time.sleep(1)

    driver.find\_element(By.XPATH, '/html/body/nav/div/div/ul/li[5]').click()

    time.sleep(1.5)

    driver.find\_element(By.XPATH, '/html/body/section/div[2]/div[2]/button').click()

    time.sleep(1.5)

def Test():

    # mở file dữ liệu

    file = open('./Data/themmoisinhvien.csv', encoding='utf')

    type1 = file.readline().strip()

    type = file.readline().strip()

    cousetype = 1

    countpass = countfail = 0

    while type!='':

        list=type.split(',')

        z = 0

        a = 1

        b = a+1

        c = b+1

        d = c+1

        e = d+1

        f = e+1

        g = f+1

        h = g+1

        i = h+1

        k = i+1

        masv = driver.find\_element(By.XPATH, '//\*[@id="studentID"]')

        masv.clear()

        masv.send\_keys(list[a])

        time.sleep(0.3)

        hoten = driver.find\_element(By.XPATH, '//\*[@id="name"]')

        hoten.clear()

        hoten.send\_keys(list[b])

        time.sleep(0.3)

        lophoc = driver.find\_element(By.XPATH, '//\*[@id="class"]')

        lophoc.clear()

        lophoc.send\_keys(list[c])

        khoas = driver.find\_element(By.XPATH, '//\*[@id="academic"]')

        khoas.clear()

        khoas.send\_keys(list[d])

        time.sleep(0.3)

        Diachi = driver.find\_element(By.XPATH, '//\*[@id="address"]')

        Diachi.clear()

        Diachi.send\_keys(list[e])

        time.sleep(0.3)

        Ngaysinh = driver.find\_element(By.XPATH, '//\*[@id="dateOfBirth"]')

        Ngaysinh.clear()

        Ngaysinh.send\_keys(list[f])

        time.sleep(0.3)

        Sodienthoai = driver.find\_element(By.XPATH, '//\*[@id="phone"]')

        Sodienthoai.clear()

        Sodienthoai.send\_keys(list[g])

        time.sleep(0.3)

        thongtinbo = driver.find\_element(By.XPATH, '//\*[@id="father"]')

        thongtinbo.clear()

        thongtinbo.send\_keys(list[h])

        time.sleep(0.3)

        thongtinme = driver.find\_element(By.XPATH, '//\*[@id="mother"]')

        thongtinme.clear()

        thongtinme.send\_keys(list[i])

        time.sleep(0.3)

        driver.find\_element(By.XPATH, '//\*[@id="addNewModal"]/div/div/div[3]/button[1]').click()

        time.sleep(1.3)

        notification = driver.find\_element(By.XPATH, '/html/body/div[1]').text

        time.sleep(1.3)

        if str(notification) == list[k]:

            if str(notification) == "Thêm mới thành công":

                print(list[z], 'Pass')

                countpass += 1

                time.sleep(1)

                driver.find\_element(By.XPATH, '/html/body/section/div[2]/div[2]/button').click()

                time.sleep(1)

            else:

                print(list[z], 'Pass')

                countpass += 1

                time.sleep(1)

        else:

            if str(notification) == "Thêm mới thành công":

                countfail += 1

                print(list[z], 'Fail')

                print('Thông báo thực tế:   ', notification)

                print('Thông báo mong muốn: ', list[k])

                print('Thông tin thêm mới:')

                print('Mã sinh viên:    ', list[a])

                print('Họ tên sinh viên: ', list[b])

                print('Lớp: ', list[c])

                print('Khóa:    ', list[d])

                print('Địa chỉ: ', list[e])

                print('Ngày sinh: ', list[f])

                print('Số điện thoại:    ', list[g])

                print('Thông tin bố: ', list[h])

                print('Thông tin mẹ: ', list[i])

                driver.find\_element(By.XPATH, '/html/body/section/div[2]/div[2]/button').click()

                time.sleep(1)

            else:

                countfail += 1

                print(list[z], 'Fail')

                print('Thông báo thực tế:   ', notification)

                print('Thông báo mong muốn: ', list[k])

                print('Thông tin thêm mới:')

                print('Mã sinh viên:    ', list[a])

                print('Họ tên sinh viên: ', list[b])

                print('Lớp: ', list[c])

                print('Khóa:    ', list[d])

                print('Địa chỉ: ', list[e])

                print('Ngày sinh: ', list[f])

                print('Số điện thoại:    ', list[g])

                print('Thông tin bố: ', list[h])

                print('Thông tin mẹ: ', list[i])

        time.sleep(1.2)

        type = file.readline().strip()

        a += 1

        z += 1

        cousetype += 1

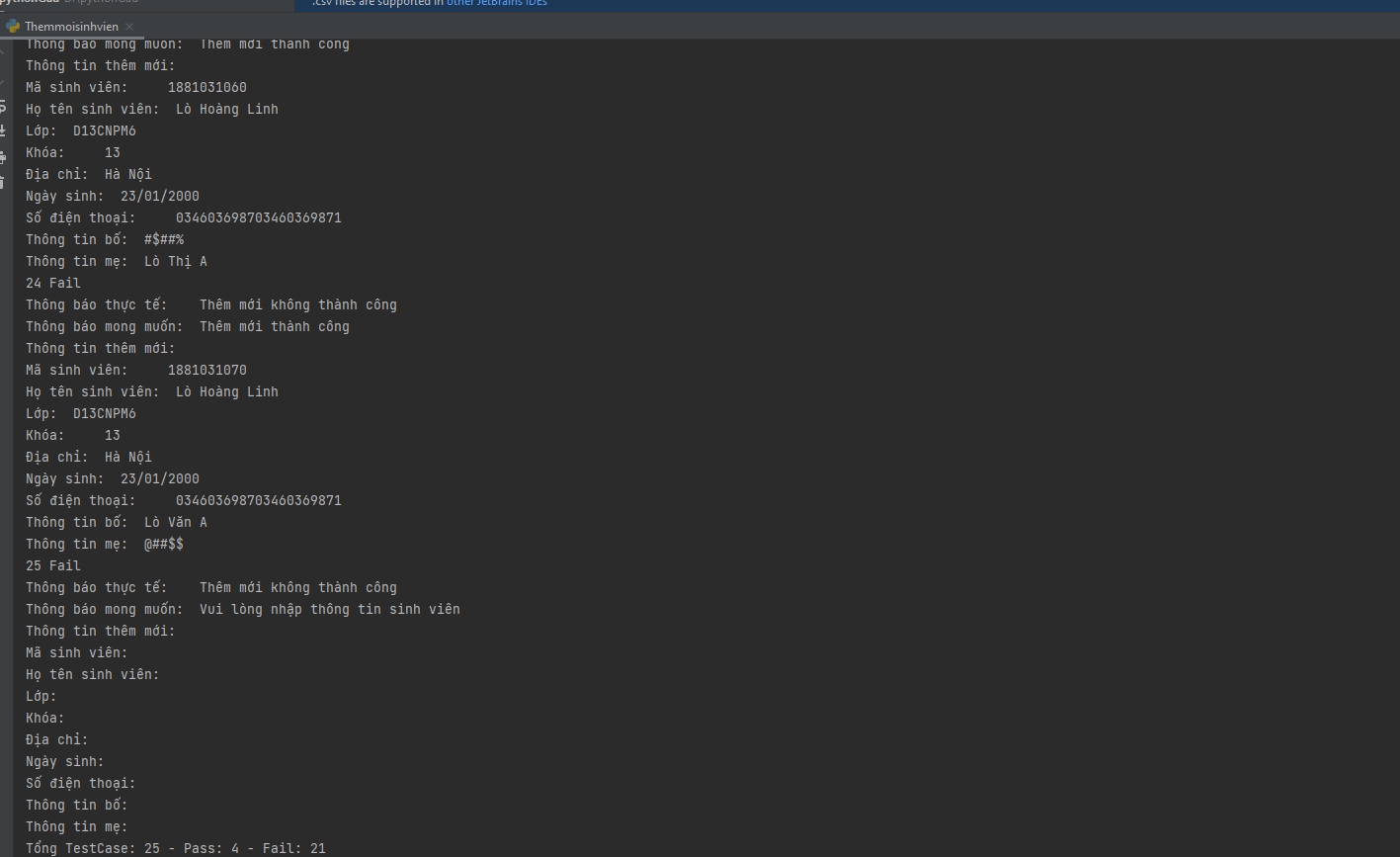
    print('Tổng TestCase:',countfail+countpass,'-', 'Pass:',countpass,'-', 'Fail:',countfail)

Setup\_driver()

Test()

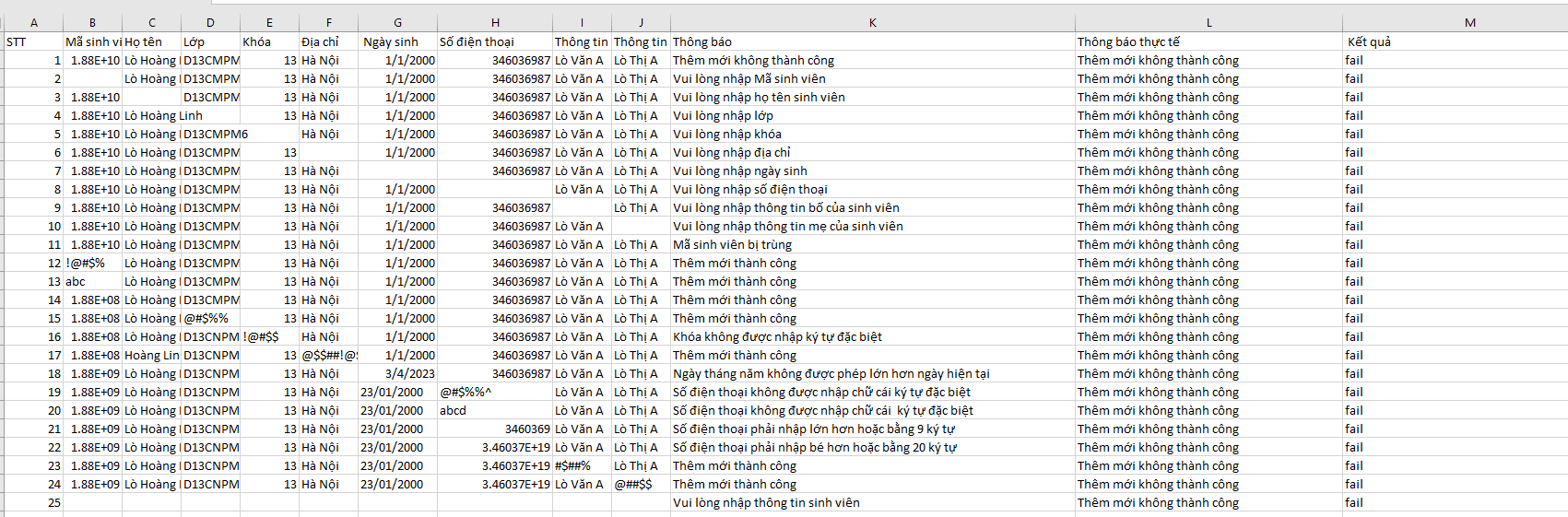
driver.quit()

**Kết quả sau khi thực hiện chạy test case:**



* Tổng số test case chạy: 25
* Tổng số trường hợp kiểm thử thành công: 4
* Tổng số trường hợp kiểm thử không thành công: 21

**Kết quả export qua file csv sau khi đã có kết quả test:**

****

# 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 phẩn 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 nên 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 WebDriver hỗ trợ vào công việc tương lai sau này.

# TÀI LIỆU THAM KHẢO

[1] Glenford J. Myers, Corey Sandler, Tom Badgett : The Art of Software Testing, 3rd Edition.

[2] Elfriede Dustin (1999), Automated Software Testing, Addison Wesley, 1999, ISBN 0-20143-287-0.

[3] Pekka Laukkanen: Data-Driven and Keyword-Driven Test Automation Frameworks.

[4] Douglas Hoffman (1999), Test Automation Architectures: Planning for Test Automation, Software Quality Methods, LLC.

[5] Mark Fewster and Dorothy Graham (1999), Software Test Automation: Effective use of test execution tools, ACM Press Books.

[6] Tuomas Pajunen, Tommi Takala, and Mika Katara, Model-Based Testing with a General Purpose Keyword-Driven Test Automation Framework.

* Các website tham khảo:

[1] <https://www.guru99.com/automation-testing.html>

[2] <https://itnavi.com.vn/blog/ky-thuat-manual-testing-la-gi/>

[3]<https://viblo.asia/p/su-khac-nhau-giua-manual-testing-va-automation-testing-1VgZvoRmlAw>

[4] <http://robotframework.org/>

[5] <https://github.com/alans09/robotframework-RanorexLibrary/wiki/Installation>

[6] <http://en.wikipedia.org/wiki/Test_automation>