TRƯỜNG ĐẠI HỌC KINH TẾ

**KHOA THỐNG KÊ – TIN HỌC**

–––––––––––––––––––––––––––––––



**BÁO CÁO THỰC TẬP NGHỀ NGHIỆP**

**NGÀNH HỆ THỐNG THÔNG TIN QUẢN LÝ**

**CHUYÊN NGÀNH QUẢN TRỊ HỆ THỐNG THÔNG TIN**

**ĐỀ TÀI**

**NGHIÊN CỨU VÀ ỨNG DỤNG KIỂM THỬ THỦ CÔNG CHO WEBSITE SWAGLABS**

Sinh viên thực hiện : Lê Thị Vân

Lớp : 45K21.1

Đơn vị thực tập : TMA Solutions Bình Định

Cán bộ hướng dẫn : Phạm Minh Tuấn

Giảng viên hướng dẫn : ThS Cao Thị Nhâm

**Đà Nẵng, 8/2022**

**NHẬN XÉT CỦA ĐƠN VỊ THỰC TẬP**

Họ và tên sinh viên:

Lớp: Khoa: Trường:

Thực tập từ ngày: …./……/ 2022 đến ngày: ........./ .……./ 2022

Tại:

Địa chỉ:

Sau quá trình thực tập tại đơn vị của sinh viên, chúng tôi có một số nhận xét, đánh giá như sau:

**1. Về thái độ, ý thức, đạo đức, kỷ luật**

**2. Kiến thức chuyên môn**

**3. Khả năng hòa nhập và thích nghi với công việc**

**4. Trách nhiệm, sáng tạo trong công việc**

**5. Các nhận xét khác**

**Đánh giá chung:**

**Điểm:**

……….., ngày .......tháng ......năm 2022

**Xác nhận của đơn vị thực tập**

# LỜI CẢM ƠN

Lời cám ơn đầu tiên em xin gửi đến Công Ty TMA Solution Bình Định – đơn vị đã tiếp nhận, hỗ trợ nhiệt tình và tạo mọi điều kiện thuận lợi cho em tiếp cận thực tế, được trau dồi kiến thức và kinh nghiệm.

Em xin cảm ơn đến quý Thầy Cô Khoa Thống kê- Tin học trường Đại Học Kinh Tế Đà Nẵng đã tạo điều kiện để em có điều kiện thực tập. Đặc biệt, em xin cảm ơn cô Cao Thị Nhâm, người đã tận tình hướng dẫn em hoàn thành bài báo cáo này.

Cuối cùng em xin cảm ơn anh Phạm Minh Tuấn người hướng dẫn em tại công ty TMA Solution Bình Định đã giúp đỡ, tạo điều kiện, cung cấp những tài liệu cho em trong quá trình thực tập về mảng Tester để hoàn thành được kỳ thực tập tốt hơn.

Vì thời gian và kiến thức còn hạn hẹp nên bài báo cáo không thể tránh khỏi những thiếu sót, Em rất mong nhận được những ý kiến đóng góp và phản hồi từ quý thầy cô để em có thể khắc phục được những sai sót cũng như rút ra được những bài học cho mình và trau dồi thêm những kiến thức mới. Em xin chân thành cảm ơn!

# LỜI CAM ĐOAN

Em xin cam dự án là kết quả nghiên cứu của em dưới sự hướng dẫn của giáo viên hướng dẫn: ThS. Cao Thị Nhâm và mentor Phạm Minh Tuấn ngoài ra không có bất kì sự sao chép nào của người khác.

Dự án, nội dung báo cáo là sản phẩm mà chúng em đã nổ lực nghiên cứu trong quá trình học tại nhà và bên thực tập. Các dữ liệu, kết quả trình bày trong báo cáo là hoàn toàn trung thực, chúng em xin chịu hoàn toàn trách nhiệm, kỷ luật bộ môn và nhà trường đề ra nếu có vấn đề nào xảy ra.

MỤC LỤC

[LỜI CẢM ƠN 3](#_Toc110974883)

[LỜI CAM ĐOAN 4](#_Toc110974884)

[MỤC LỤC 5](#_Toc110974885)

[DANH MỤC HÌNH ẢNH 7](#_Toc110974886)

[DANH MỤC BẢNG BIỂU 8](#_Toc110974887)

[DANH MỤC CÁC TỪ VIẾT TẮT 9](#_Toc110974888)

[LỜI MỞ ĐẦU 10](#_Toc110974889)

[CHƯƠNG 1. CƠ SỞ LÝ THUYẾT 11](#_Toc110974890)

[1.1. Tổng quan về kiểm thử phần mềm 11](#_Toc110974891)

[1.1.1. Khái niệm kiểm thử phần mềm 11](#_Toc110974892)

[1.1.2. Bảy nguyên tắc trong kiểm thử 11](#_Toc110974893)

[1.1.3. Quy trình kiểm thử cơ bản 11](#_Toc110974894)

[1.2. Các mô hình phát triển phần mềm 13](#_Toc110974895)

[1.2.1. Mô hình Waterfall 13](#_Toc110974896)

[1.2.2. Mô hình V 15](#_Toc110974897)

[1.3. Các cấp độ kiểm thử 17](#_Toc110974898)

[1.3.1. Unit testing (Kiểm thử đơn vị) 17](#_Toc110974899)

[1.3.2. Intergration testing (Kiểm thử tích hợp) 17](#_Toc110974900)

[1.3.4. Acceptance Testing (Kiểm thử chấp nhận) 18](#_Toc110974901)

[1.4. Các loại kiểm thử 19](#_Toc110974902)

[1.4.1. Kiểm thử chức năng 19](#_Toc110974903)

[1.4.2. Kiểm thử phi chức năng 19](#_Toc110974904)

[1.4.3. Kiểm thử cấu trúc 19](#_Toc110974905)

[1.4.4. Kiểm thử thay đổi 19](#_Toc110974906)

[1.5. Các kĩ thuật kiểm thử 20](#_Toc110974907)

[1.5.1. Thử nghiệm tĩnh 20](#_Toc110974908)

[1.5.2. Kiểm thử động 20](#_Toc110974909)

[CHƯƠNG 2. PHÂN TÍCH WEBSITE SWAGLABS 22](#_Toc110974910)

[2.1. Giới thiệu về website SwagLabs 22](#_Toc110974911)

[2.2. Sơ đồ Use case tổng quát 22](#_Toc110974912)

[2.3. Phân tích use case Đăng nhập 23](#_Toc110974913)

[2.3.1. Sơ đồ Use case cho chức năng Login 23](#_Toc110974914)

[2.3.2. Đặc tả yêu cầu cho chức năng Đăng nhập 23](#_Toc110974915)

[2.3.3. Sơ đồ hoạt động 24](#_Toc110974916)

[CHƯƠNG 3. THIẾT KẾ USE CASE VÀ THỰC THI KIỂM THỬ 25](#_Toc110974917)

[3.1. Thiết kế Test case cho màn hình Đăng nhập 25](#_Toc110974918)

[3.2. Thực thi kiểm thử 31](#_Toc110974919)

[CHƯƠNG 4. KẾT QUẢ 36](#_Toc110974920)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 37](#_Toc110974921)

[TÀI LIỆU THAM KHẢO 38](#_Toc110974922)

# DANH MỤC HÌNH ẢNH

[***Hình 1. Mô hình Waterfall*** 13](#_Toc110974945)

[***Hình 2. Mô Hình V*** 15](#_Toc110974946)

[***Hình 3. Mô hình phát triển Iterative-incremental*** 16](#_Toc110974947)

[***Hình 4. Usecase tổng quát hệ thống*** 22](#_Toc110974948)

[***Hình 5. Màn hình Đăng nhập*** 23](#_Toc110974949)

[***Hình 6. use case Đăng nhập*** 23](#_Toc110974950)

[***Hình 7. Sơ đồ hoạt động của use case Đăng nhập*** 24](#_Toc110974951)

[***Hình 8. Màn hình thực thi TC1*** 31](#_Toc110974952)

[***Hình 9. Màn hình thực thi TC1*** 31](#_Toc110974953)

[***Hình 10. Màn hình thực thi TC2*** 32](#_Toc110974954)

[***Hình 11. Màn hình thực thi TC3*** 32](#_Toc110974955)

[***Hình 12.Màn hình thực thi TC4*** 33](#_Toc110974956)

[***Hình 13. Màn hình thực thi TC5*** 33](#_Toc110974957)

[***Hình 14. Màn hình thực thi TC6*** 34](#_Toc110974958)

[***Hình 15. Màn hình thực thi TC7*** 34](#_Toc110974959)

[***Hình 16. Màn hình thực thi TC8*** 35](#_Toc110974960)

[***Hình 17. Màn hình thực thi TC9*** 35](#_Toc110974961)

# DANH MỤC BẢNG BIỂU

[***Bảng 1. Thiết kế test case*** 25](#_Toc110974962)

[***Bảng 2. Kết quả thực thi kiểm thử*** 36](#_Toc110974963)

# DANH MỤC CÁC TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| STT | **Từ viết tắt** | **Từ viết đầy đủ** |
| 1 | TC | Test case |
| 2 | UI | User interface |
| 3 | N/A | Not available |

# LỜI MỞ ĐẦU

1. **Mục tiêu nghiên cứu của đề tài**

Đề tài này nghiên cứu và tìm hiểu các kiến thức cơ bản về kiểm thử phần mềm, đặc biệt áp dụng thực hành kiểm thử tự động cho website.

1. **Nhiệm vụ của đề tài**

Tìm hiểu, áp dụng các kiến thức về kiểm thử phần mềm và kết hợp với các công cụ kiểm thử tự động để kiểm thử website SwagLabs

1. **Phương pháp nghiên cứu**

* Nghiên cứu lý thuyết từ tài liệu sẵn có.
* Sử dụng các công cụ kiểm thử tự động để thực hành kiểm thử

1. **Đối tượng và phạm vi nghiên cứu**

Đối tượng nghiên cứu: Website Swaglabs

Phạm vi nghiên cứu: Nghiên cứu lý thuyết cơ bản về kiểm thử phần mềm và các công cụ kiểm thử phần mềm tự động, sau đó thực hiện kiểm thử.

1. **Kết cấu của đề tài**

Đề tài được tổ chức gồm phần mở đầu, 4 chương nội dung và phần kết luận.

* Mở đầu
* **Chương 1**: Cơ sở lý thuyết
* **Chương 2**: Phân tích website SwagLabs
* **Chương 3**: Thiết kế use case và thực thi kiểm thử
* **Chương 4**: Kết quả
* Kết luận và hướng phát triển

# CƠ SỞ LÝ THUYẾT

* 1. Tổng quan về kiểm thử phần mềm

### Khái niệm kiểm thử phần mềm

Kiểm thử phần mềm là một quá trình kiểm tra để đưa ra những đánh giá về chức năng của một ứng dụng phần mềm với mục đích xem xét liệu phần mềm đó đã được phát triển theo đúng tiêu chuẩn hay không, có đáp ứng được các yêu cầu cụ thể không. Bên cạnh đó, việc xác định được chi tiết lỗi đảm bảo rằng sản phẩm khi cho ra đời sẽ không có khuyết điểm, hướng đến mục tiêu tạo ra một sản phẩm phần mềm hoàn chỉnh và chất lượng cao.

### Bảy nguyên tắc trong kiểm thử

Nguyên tắc 1: Kiểm thử cho thấy sự hiện diện của các khiếm khuyết

Nguyên tắc 2: Kiểm thử toàn bộ là không thể

Nguyên tắc 3: Kiểm thử sớm

Nguyên tắc 4: Sự tập trung của lỗi

Nguyên tắc 5: Nghịch lý thuốc trừ sâu

Nguyên tắc 6: Kiểm thử theo các ngữ cảnh độc lập

Nguyên tắc 7: Sự sai lầm về việc không có lỗi

### Quy trình kiểm thử cơ bản

Về cơ bản thì quy trình kiểm thử phần mềm gồm 5 giai đoạn:

* ***Giai đoạn 1:*** Lập kế hoạch và kiểm soát kiểm thử (Test plan).

Kế hoạch kiểm thử là một tài liệu mô tả các mục tiêu, phạm vi, phương pháp tiếp cận, và tập trung vào nỗ lực kiểm thử phần mềm. Cấu trúc chung của một test plan:

* Tên project.
* Danh sách các module cần test.
* Ngày bắt đầu, ngày kết thúc.
* Danh sách các test case.
* Nhân sự tham gia.
* Tài nguyên sử dụng (Severs, Workstations, Printers).
* Kế hoạch thực hiện.
* **Giai đoạn 2:** Phân tích và thiết kế kiểm thử.

Ở giai đoạn này thực hiện các nhiệm vụ chủ yếu:

* Rà soát các yêu cầu cần thiết trước khi tiến hành kiểm thử như: tài liệu đặc tả, tài liệu đặc tả, tài liệu thiết kế.
* Xác định các điều kiện kiểm thử
* Thiết kế test case
* Đánh giá tính khả thi trong việc kiểm thử cũng như yêu cầu của hệ thống.
* Chuẩn bị môi trường test cũng như xác định các yêu cầu về cơ sở hạ tầng cũng như các công cụ kiểm thử tương ứng.
* **Giai đoạn 3:** Thực thi và chạy kiểm thử.

Ở giai đoạn này thực hiện các nhiệm vụ chủ yếu:

* Thực hiện test dựa trên test case đã viết.
* Chạy lại các case bị lỗi trước đó để xác nhận là case đó đã được sửa.
* So sánh kết quả khi thực thi với kết quả mong đợi.
* Đánh giá kết quả kiểm thử cho các trường hợp kiểm thử.
* Biết báo cáo lỗi khi có bug
* **Giai đoạn 4:** Đánh giá và báo cáo:

Ở giai đoạn này thực hiện các nhiệm vụ chủ yếu:

* Đối chiếu các kết quả thực thi các test case cho với các tiêu chí đã đề ra ở giai đoạn 1.
* Đánh giá xe liệu có cần phải test thêm hay điều chỉnh các tiêu chí kết thúc kiểm thử trong bản kế hoạch.
* Viết báo cáo tóm tắt hoạt động kiểm thử cũng như kết quả kiểm thử gửi cho các bên liên quan.
* **Giai đoạn 5:** Đóng kiểm thử

Ở giai đoạn này thực hiện các nhiệm vụ chủ yếu:

* Kiểm tra khách hàng được nhận sản phẩm theo dự kiến từ đầu và đảm bảo rằng tất cả sự cố đã được giải quyết.
* Hoàn thiện và lưu trữ phần mềm testware như script, môi trường kiểm thử để sau này sử dụng lại.
* Bàn giao lại testware cho đội bảo trì.
* Đánh giá cách kiểm thử và đưa ra bài học cho lần phát hành và các dự án trong tương lai.
  1. **Các mô hình phát triển phần mềm**
     1. **Mô hình Waterfall**

Text

Description automatically generated

***Hình 1. Mô hình Waterfall***

* ***Mô tả***
* Đây được coi như là mô hình phát triển phần mềm đầu tiên được sử dụng.
* Mô hình này áp dụng tuần tự các giai đoạn của phát triển phần mềm.
* Đầu ra của giai đoạn trước là đầu vào của giai đoạn sau. Giai đoạn sau chỉ được thực hiện khi giai đoạn trước đã kết thúc. Đặc biệt không được quay lại giai đoạn trước để xử lý các yêu cầu khi muốn thay đổi.
* ***Phân tích mô hình***
* Requirement gathering: Thu thập và phân tích yêu cầu được ghi lại vào tài liệu đặc tả yêu cầu trong giai đoạn này.
* System Analysis: Phân tích thiết kế hệ thống phần mềm, xác định kiến trúc hệ thống tổng thể của phần mềm.
* Coding: Hệ thống được phát triển theo từng unit và được tích hợp trong giai đoạn tiếp theo. Mỗi Unit được phát triển và kiểm thử bởi dev được gọi là Unit Test.
* Testing: Cài đặt và kiểm thử phần mềm. Công việc chính của giai đoạn này là kiểm tra và sửa tất cả những lỗi tìm được sao cho phần mềm hoạt động chính xác và đúng theo tài liệu đặc tả yêu cầu.
* Implementation: Triển khai hệ thống trong môi trường khách hàng và đưa ra thị trường.
* Operations and Maintenance: Bảo trì hệ thống khi có bất kỳ thay đổi nào từ phía khách hàng, người sử dụng.
* ***Ứng dụng***

Mô hình thường được áp dụng cho các dự án phần mềm như sau:

* Các dự án nhỏ , ngắn hạn.
* Các dự án có ít thay đổi về yêu cầu và không có những yêu cầu không rõ ràng.
* ***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
  + 1. **Mô hình V**

Diagram

Description automatically generated

***Hình 2. Mô Hình V***

* ***Mô tả***
* Mô hình chữ V là một phần mở rộng của mô hình thác nước và được dựa trên sự kết hợp của một giai đoạn thử nghiệm cho từng giai đoạn phát triển tương ứng. Đây là một mô hình có tính kỷ luật cao và giai đoạn tiếp theo chỉ bắt đầu sau khi hoàn thành giai đoạn trước.
* Với V model thì công việc test được tham gia ngay từ đầu.
* ***Ứng dụng***
* Yêu cầu được xác định rõ ràng.
* Xác định sản phẩm ổn định.
* Công nghệ không thay đổi và được hiểu rõ bởi nhóm dự án.
* Không có yêu cầu không rõ ràng hoặc không xác định.
* Dự án ngắn.
* ***Ưu điểm***
* Đây là một mô hình có tính kỷ luật cao và các giai đoạn được hoàn thành cùng một lúc.
* Hoạt động tốt cho các dự án nhỏ, khi các yêu cầu được hiểu rất rõ.
* Đơn giản và dễ hiểu và dễ sử dụng, dễ quản lý.
* ***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 kém cho 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.
  + 1. **Mô hình phát triển Iterative-incremental**

Diagram

Description automatically generated

***Hình 3. Mô hình phát triển Iterative-incremental***

* Mô hình phát triển Iterative-incremental là quá trình xây dựng các yêu cầu, thiết kế, xậy dựng và kiểm thử một hệ thống trong một loạt các chu kỳ phát triển ngắn.
* Ví dụ như : nguyên mẫu (prototyping), Phát triển ứng dụng nhanh (RAD), Rational Unified Process (RUP) và mô hình phát triển agile.
* ***Đặc điểm***
* Phân chia thành số gia (increments ) hoặc xây dựng (builds).
* Các số gia ban đầu sẽ có các cơ sở hạ tầng cần thiết
* Các số gia có thể được kiểm thử với nhiều mức trong mỗi lần lặp.
* Các công việc thực hiện với số gia tiếp theo (Subsequent increments):
* Kiểm thử các chức năng mới
* Kiểm thử hồi quy các chức năng hiện có
* Kiểm thử tích hợp cả phần mới và phần cũ
* Xác minh và xác nhận hợp lệ có thể được thực hiện trong mỗi số gia.
* ***Ưu điểm:***
* Sớm cung cấp ra thị trường
* Đơn giản để quản lý
* Giảm đầu tư ban đầu
* Nhận được thông tin phản hồi sớm
  1. **Các cấp độ kiểm thử**
     1. **Unit testing (Kiểm thử đơn vị)**
* Loại kiểm thử này được chạy bởi developer trước khi cài đặt để chuyển giao cho đội kiểm thử thực hiện những test case chính thống. Unit test được developer chạy tương ứng cho từng đơn vị mã nguồn. Developer sử dụng bộ dữ liệu kiểm thử từ test case để đảm bảo chất lượng.
* Mục đích của Unit test là cô lập từng phần của chương trình và kiểm tra các bộ phận độc lập đấy đã hoạt động chính xác theo yêu cầu hay chưa.
* Kiểm thử theo phương pháp này không thể tìm được mọi lỗi của ứng dụng.
* Không thể đánh giá mọi tình huống có thể xảy ra trong chương trình. Nó bị giới hạn bởi các tình huống và bộ dữ liệu test mà developer sử dụng để kiểm thử chương trình.
  + 1. **Intergration testing (Kiểm thử tích hợp)**
* Kiểm thử tích hợp kiểm tra sự tích hợp hoặc giao diện giữa các thành phần, tương tác với các phần khác nhau của hệ thống như hệ điều hành, hệ thống tệp và phần cứng hoặc giao diện giữa các hệ thống.
* Các hình thức Test:
* Bottom up
* Top down
  + 1. **System Testing (Kiểm thử hệ thống)**
* Kiểm thử hệ thống là kiểm thử một hệ thống đã hoàn thành, đã tích hợp đầy đủ các chức năng nhằm kiểm tra xem hệ thống phần mềm đó có đáp ứng đầy đủ các yêu cầu chức năng theo bản đặc tả yêu cầu phần mềm (SRS) hay không. Người thực hiện test level này thường là Tester.
* Kiểm thử hệ thống thuộc phạm vi Kiểm thử hộp đen (tức là Tester chỉ quan tâm đầu vào và kết quả mong đợi ở đầu ra mà không cần kiểm tra code bên trong được viết như thế nào).
* Kiểm thử hệ thống theo kiểm thử chức năng và phi chức năng:
* Kiểm thử chức năng:
* Requirement-based testing: Thử nghiệm dựa trên yêu cầu
* Business-process-based testing: Thử nghiệm dựa trên quy trình kinh doanh
* Kiểm thử phi chức năng:
* Sanity Testing
* Re-Test
* Regression Test
* Security Test
* Recovery Test
* Usability Test
* Performance Test
* Load Test
* Stress Test
  + 1. **Acceptance Testing (Kiểm thử chấp nhận)**
* Acceptance Testing là kiểm thử được tiến hành để tìm xem các yêu cầu của đặc điểm kỹ thuật hoặc hợp đồng có được đáp ứng với những yêu cầu của khách hàng hay không. Kiểm thử chấp nhận về cơ bản được thực hiện bởi người dùng hoặc khách hàng.Acceptance test gồm 2 loại kiểm thử là:
* **Alpha Test**
* **Beta Test**
  1. **Các loại kiểm thử**

### Kiểm thử chức năng

Kiểm thử chức năng được các Tester(người kiểm thử) thực hiện để kiểm tra hoạt động của một chức năng nào đó có được hoạt động ổn định cũng như tốc độ phản hồi của chức năng có ổn định hay không.

### Kiểm thử phi chức năng

Kiểm thử phi chức năng được thực hiện nhằm trả lời câu hỏi: “Phần mềm có hoạt động tốt không?”. Kiểm thử phi chức năng chú trọng nhiều hơn vào những khía cạnh khác của phần mềm, như là độ bảo mật và khả năng tải của phần mềm đó.

### Kiểm thử cấu trúc

* Kiểm thử cấu trúc thường được gọi là “kiểm thử hộp trắng“(white-box) hoặc “hộp thủy tinh” (glass-box) vì chúng quan tâm đến những gì đang xảy ra bên trong hộp.
* Kiểm thử cấu trúc là kiểm thử dựa trên phân tích cấu trúc bên trong của thành phần hoặc hệ thống. Nó thường được sử dụng như một cách đo lường của kiểm thử, thông qua độ bao phủ của một tập hợp các yếu tố cấu trúc.

### Kiểm thử thay đổi

* + Mục đích của kiểm thử thay đổi là để kiểm tra xem phần mềm có vận hành trơn tru sau những lần sửa lỗi hay không.
* Kiểm thử thay đổi gồm 2 loại chính:
* Kiểm thử xác nhận: Thường sẽ diễn ra sau khi lỗi trong phần mềm đã được xác nhận và được sửa. Lúc này, vai trò của Kiểm thử xác nhận là để xem lỗi đã thực sự được sửa hay chưa. Các tester sẽ tiến hành bằng cách cho một input giống hệt ban đầu và test xem output có ra được như mong muốn hay không.
* Regression Testing (Kiểm thử hồi quy): Mục đích của kiểm thử hồi quy để xác nhận rằng các thay đổi trong phần mềm hoặc môi trường không gây ra bất lợi ngoài mong muốn và hệ thống vẫn đáp ứng các yêu cầu.
  1. **Các kĩ thuật kiểm thử**

### Kiểm thử tĩnh

Thử nghiệm tĩnh là loại kiểm tra trong đó code không được thực hiện. Nó có thể được thực hiện bằng tay hoặc bằng một bộ công cụ. Loại kiểm tra này thực hiện kiểm tra code, tài liệu yêu cầu và tài liệu thiết kế và đưa ra nhận xét, lưu nhận xét vào tài liệu công việc.

### Kiểm thử động

Kiểm thử tĩnh là sẽ soi vào code còn kiểm thử động thì sẽ thực hiện dựng code đó lên rồi thực thi việc test trên môi trường và thiết bị tương tự như môi trường thực tế mà phần mềm/hệ thống sẽ được sử dụng.

* + - 1. **Kiểm thử hộp đen**

Kiểm thử hôp đen hay còn gọi là Black Box Testing là phương pháp kiểm thử phần mềm được sử dụng để kiểm tra phần mềm mà không biết cấu trúc nội bộ của mã hoặc chương trình. Tất cả các thử nghiệm được thực hiện dựa trên quan điểm của Người dùng. Tester chỉ đưa ra các đầu vào hợp lệ và không hợp lệ để xác định các kết quả đầu ra mong đợi chính xác. Mục đích của Kiểm thử hộp đen là kiểm tra xem phần mềm có hoạt động như mong đợi trong tài liệu yêu cầu hay không và liệu có đáp ứng được tài liệu kỳ vọng của khách hàng hay không.

* + - 1. **Kiểm thủ hộp trắng**
  + Kiểm thử hộp trắng là loại thử nghiệm được thực hiện để kiểm tra cấu trúc code. Loại thử nghiệm này đòi hỏi người test phải có kiến thức về code. Do đó, phần lớn là do các lập trình viên, nhà phát triển phần mềm thực hiện.
* Đặc điểm:
* Kiểm thử hộp trắng quan tâm đến việc hệ thống vận hành như thế nào chứ không phải chứ năng của hệ thống. Vì nó dựa vào những thuật toán cụ thể, vào những cấu trúc dữ liệu bên trong của thành phần phần mềm.
* Trong kỹ thuật kiểm thử này, đòi hỏi người tester phải có kiến thức và kỹ năng nhất định về ngôn ngữ lập trình được dùng, hiểu thuật toán trong phần mềm, để có thể hiểu được chi tiết về đoạn code cần kiểm thử .
* Mức test này thường yêu cầu các tester phải viết test case đầy đủ các nhánh trong code; khi test, sẽ set điều kiện và data để chạy vào đủ tất cả các nhánh trong thuật toán, đảm bảo thực hiện đầy đủ.
  + - 1. **Kỹ thuật dựa trên kinh nghiệm**
* Kỹ thuật này rất phụ thuộc vào kiến thức, kỹ năng và kinh nghiệm của người kiểm thử.
* Nó thường tập trung vào:
* Lỗi điển hình
* Giá trị đặc biệt
* Nội dung chưa rõ ràng
* Kết nối đặc biệt và bối cảnh
* Các yếu tố rủi ro
* Sử dụng các chương trình và môi trường của nó

# PHÂN TÍCH WEBSITE SWAGLABS

## Giới thiệu về website SwagLabs

Website Swaglabs là trang web cho phép khách hàng sử dụng để có thể tiến hành việc mua sắm sản phẩm trực tuyến. Cụ thể, khi truy cập vào website này, khách hàng tiến hành đăng nhập, xem thông tin sản phẩm, thêm sản phẩm vào giỏ hàng, đặt hàng nhanh chóng.

* **Chức năng của hệ thống:**
* Đăng nhập
* Xem thông tin chi tiết sản phẩm
* Chọn sản phẩm
* Quản lí giỏ hàng
* Thanh toán đơn hàng
* Đăng xuất
* Vì website được xây dựng để kiểm thử quá trình mua hàng của người dùng nên hệ thống sẽ không có quản lí bán hàng.
* **Tác nhân liên quan đến hệ thống:**

**User**: Là những khách hàng sử dụng website Swaglabs để mua hàng. Người dùng muốn sử dụng trang web này bắt buộc phải có tài khoản thì mới thực hiện được chức năng bên trong. Người dùng được thực hiện những chức năng sau:

* Đăng nhập, đăng xuất.
* Xem thông tin chi tiết sản phẩm
* Thêm sản phẩm vào giỏ hàng
* Thêm, xóa sản phẩm trong giỏ hàng
* Thanh toán đơn hàng
* Đặt mua sản phẩm

## Sơ đồ Use case tổng quát



***Hình 4. Usecase tổng quát hệ thống***

## Phân tích use case Đăng nhập

Graphical user interface, application

Description automatically generated

***Hình 5. Màn hình Đăng nhập***

### Sơ đồ Use case cho chức năng Login

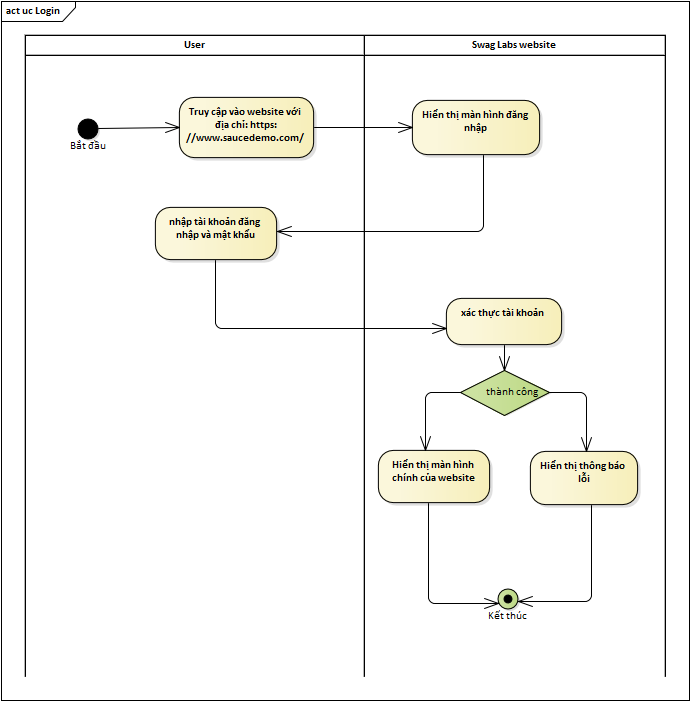


***Hình 6. use case Đăng nhập***

### Đặc tả yêu cầu cho chức năng Đăng nhập

* Tác nhân: Người dùng
* Điều kiện tiên quyết: Người dùng đã có tài khoản để đăng nhập.
* Mô tả khái quát: Usecase cho phép người dùng đăng nhập vào Website để mua hàng.
* Mô tả chi tiết:
* Bước 1: Người dùng truy cập vào website với địa chỉ: [https://www.saucedemo.com/](https://www.saucedemo.com/ )
* Bước 2: Hệ thống hiển thị màn hình đăng nhập
* Bước 3: Người dùng nhập tài khoản đăng nhập và mật khẩu, sau đó chọn nút Login
* Bước 4: Hệ thống thực hiện xác thực tài khoản
* Bước 5: Nếu xác thực thành công, hệ thống hiển thị màn hình chính của website. Ngược lại, hiển thị thông báo lỗi.

### Sơ đồ hoạt động



***Hình 7. Sơ đồ hoạt động của use case Đăng nhập***

# THIẾT KẾ USE CASE VÀ THỰC THI KIỂM THỬ

## Thiết kế Test case cho màn hình Đăng nhập

***Bảng 1. Thiết kế test case***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID Test case** | **Categoy** | **Steps** | **Input data** | **Excepted results** |
| 1 | Verify UI | * + - 1. Go to: https://www.saucedemo.com/ |  |  |
| * + - 1. Observe all items in login page |  | Items in the login page include:   * “Swag Labs” Logo * "Username" input textbox * “Password” input text box * "LOGIN" button with white text color and red background. * “Swag lab” icon to the right of the login form. * "Acceptable username is" text is bold. * “Password for all users” is bold. * "standard\_user, locked\_out\_user, problem\_user, performance\_glitch\_user" just below the text "Accepted usernames are" * "secret\_sauce" just below the text “Password for all users” |
| * + - 1. Click on "Username" textbox |  | * The placeholder in the Username text box is disappear |
| * + - 1. Input value into 'Username' text box | Vanle123 | * Username value is 'Vanle123 ' and password value is empty |
| * + - 1. Click on "Password" textbox |  | * The placeholder in 'Password' text box disappear |
| 1. Input value into 'Password' text box | secret\_sauce | * Password value is displaying by 'Password' type and Username value is empty |
| 1. Delete input value of the"Username","Password" textbox and click outside the Password box. |  | * Placeholder "Username" and "Password" is displayed |
| 2 | Verify that Login is successful when the user enters the correct username, password | * + - 1. Go to: https://www.saucedemo.com/ |  |  |
| * + - 1. Enter username, password | Username: standard\_user  Password: secret\_sauce | * The entered username can be seen but the password is encrypted with an asterisk |
| * + - 1. Click “Login” button |  | * Successfully logged into the website homepage |
| 3 | Verify that when the user enter invalid username and valid password | 1. Go to: https://www.saucedemo.com/ |  |  |
| 1. Enter invalid username and valid password | Username: Vanle  Password: secret\_sauce | * The entered username can be seen but the password is encrypted with an asterisk |
| 1. Click 'Login' button |  | * Error message:” Username or password do not match any user in this service” |
| 4 | Verify that when the user enter valid username and invalid password | 1. Go to: https://www.saucedemo.com/ |  |  |
| Enter valid username and invalid password | Username: standard\_user  Password: Vanle | * The entered username can be seen but the password is encrypted with an asterisk |
| Click 'Login' button |  | * Error message:” Username or password do not match any user in this service” |
| 5 | Verify that when the user enter invalid username and password | 1. Go to: https://www.saucedemo.com/ |  |  |
| 2. Enter invalid username and password | Username: Vanle  Password: Vanle123 | * The entered username can be seen but the password is encrypted with an asterisk |
| 3. Click 'Login' button |  | * Error message:”Username and password do not match any user in this service” |
| 6 | Verify that when the user enter empty username and valid password | 1. Go to: https://www.saucedemo.com/ |  |  |
| 2. Enter empty username and valid password | Username:  Password: secret\_sauce | * The entered username can be seen but the password is encrypted with an asterisk |
| 3.Click 'Login' button |  | * Error message:”Username is required” |
| 7 | Verify that when the user enter valid username and empty password | 1. Go to: https://www.saucedemo.com/ |  |  |
| 2. Enter valid username and empty password | Username: standard\_user  Password: | * The entered username can be seen but the password is encrypted with an asterisk |
| 3. Click 'Login' button |  | * Error message:”Password is required” |
| 8 | Verify that when the user enter empty username and password. | 1. Go to: https://www.saucedemo.com/ |  |  |
| 2. Enter empty username and password | Username:  Password: | * The entered username can be seen but the password is encrypted with an asterisk |
| 3. Click 'Login' button |  | * Error message:”Username is required” |
| 9 | Verify that the user has been locked out. | 1. Go to: https://www.saucedemo.com/ |  |  |
| 2. Enter username and password | Username: locked\_out\_user  Password: secret\_sauce | * The entered username can be seen but the password is encrypted with an asterisk |
| 3. Click 'Login' button |  | * Error message:” Sorry, this user has been locked out.” |

* 1. **Thực thi kiểm thử**
* **Nhiệm vụ của thực thi kiểm thử:**
* Thực hiện kiểm thử theo Test case
* Ghi chú lại các testcase thực hiện không thành công
* **TC1:** Verify UI

Graphical user interface

Description automatically generated

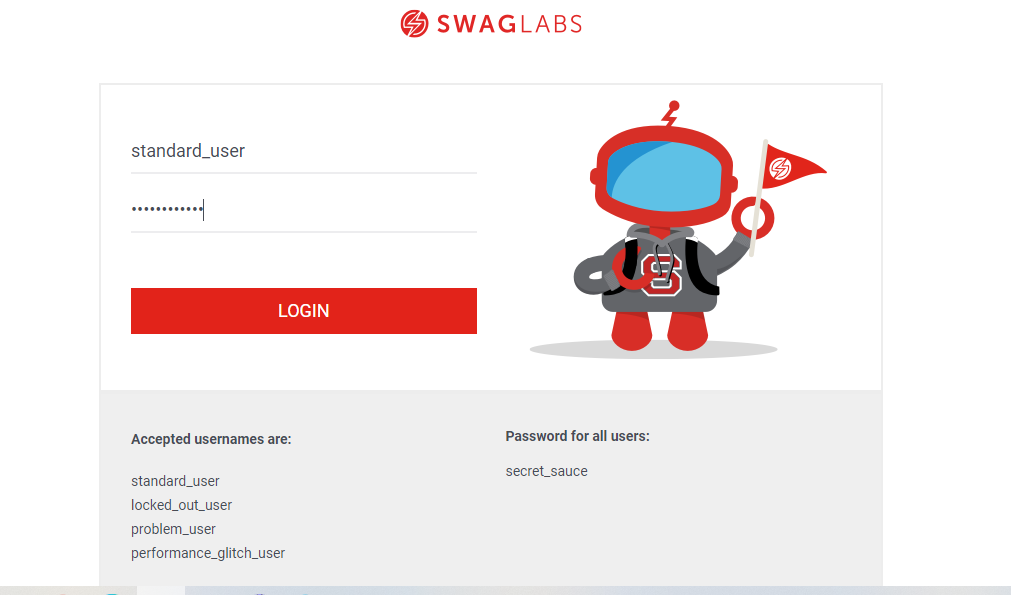
***Hình 8. Màn hình thực thi TC1***

Graphical user interface, website

Description automatically generated

***Hình 9. Màn hình thực thi TC1***

* **TC2:** Verify that Login is successful when the user enters the correct username, password.



***Hình 10. Màn hình thực thi TC2***

* **TC3:** Verify that when the user enter invalid username and valid password

Graphical user interface

Description automatically generated

***Hình 11. Màn hình thực thi TC3***

* **TC4:** Verify that when the user enter valid username and invalid password.

Graphical user interface, Teams

Description automatically generated

***Hình 12.Màn hình thực thi TC4***

* **TC5:** Verify that when the user enter invalid username and password

Graphical user interface

Description automatically generated

***Hình 13. Màn hình thực thi TC5***

* **TC6:** Verify that when the user enter empty username and valid password

Graphical user interface, website

Description automatically generated

***Hình 14. Màn hình thực thi TC6***

* **TC7:** Verify that when the user enter valid username and empty password

Graphical user interface

Description automatically generated

***Hình 15. Màn hình thực thi TC7***

* **TC8:** Verify that when the user enter empty username and password

Graphical user interface, application, Teams

Description automatically generated

***Hình 16. Màn hình thực thi TC8***

* **TC9:** Verify that the user has been locked out.

Graphical user interface

Description automatically generated

***Hình 17. Màn hình thực thi TC9***

# KẾT QUẢ

***Bảng 2. Kết quả thực thi kiểm thử***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Total Test case | Pass | Fail | N/A | Execute Date |
| Login | 9 | 6 | 3 | 0 | 08/08/2022 |
| Tester: VanLT | | | | | |

**Trạng thái của Test case**:

* Passed: Nhập nội dung này nếu testcase đã được thực hiện và kết quả đã kiểm thử đúng như kết quả mong đợi
* Failed: Nhập nội dung này nếu testcase đã được thực hiện và kết quả đã kiểm thử không đúng như kết quả mong đợi
* N/A: Nhập nội dung này nếu test case thuộc đối tượng cần phải thực hiện trong đợt kiểm thử, tuy nhiên vì lý do thiếu môi trường kiểm thử, thiếu thiết bị ... nên đã không thể thực hiện được.

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

* **Kết quả:**
* Đã tiếp thu, học hỏi được nhiều kinh nghiệm và kiến thức mà từ trước đến nay em mới có cơ hội được tiếp cận.
* Nâng cao khả năng tự nghiên cứu, chủ động tiếp thu kiến thức.
* Cải thiện khả năng giao tiếp, đọc hiểu bằng tiếng anh.
* Được đào tạo kiến thức, kỹ năng của 1 Tester và phát triển kĩ năng mềm thông qua việc thuyết trình tại công ty.
* Phát  triển và hoàn thiện bản thân cả trong tính cách lẫn công việc như tỉ mỉ, cẩn thận và chủ động hơn trong công việc.
* Nắm rõ quy trình phát triển phần mềm và quản lý dự án.
* **Kết luận:**

Kiểm thử phần mềm hiện nay vẫn là vấn đề hết sức quan trọng với các tổ chức phát triển phần mềm. Trong khuôn khổ báo cáo thực tập tốt nghiệp của mình do thời gian và kinh nghiệm còn hạn chế nên có những phần của báo cáo thực tập chưa được đào sâu nghiên cứu.

* **Hướng phát triển:**

Từ những kiến thức có được sau quá trình học hỏi và thực tập tại doanh nghiệp, bản thân em đã có những định hướng tốt hơn và có được những định hình rõ ràng hơn về công việc trong tương lai. Sau thời gian thực tập nghề nghiệp, bản thân em sẽ trau dồi thêm những kĩ năng và tìm hiểu sâu hơn về Tester để tìm kiếm cơ hội nghề nghiệp trong tương lai.

Vì kiến thức liên quan trực tiếp tới công việc Tester chỉ mới dừng lại ở những kiến thức căn bản, do đó em sẽ cố gắng phát triển nhiều hơn trong tương lai. Đồng thời, hiện tại kiến thức và kĩ năng mà em có chỉ mới dừng lại ở việc Manual Testing. Vì thế, thời gian tới em rất mong muốn được học hỏi và tìm hiểu thêm về Automation Testing. Việc tự khám phá và tìm hiểu về Automation Testing sẽ giúp ích cho công việc sau này của bản thân.

# TÀI LIỆU THAM KHẢO

Dorothy Graham, Erik van Veenendaal - Foundations of software testing\_ISTQB Certification

<https://viblo.asia/p/use-case-va-use-case-testing-Az45bgmzKxY>

<https://viblo.asia/p/cach-viet-test-case-YWOZrNpYZQ0>