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 CAM ĐOAN

Quy định:

Soạn thảo trên trang A4 (trang dọc, lề trái: 3.5cm; trên, phải, dưới: 2.5 cm), Font Times New Roman, canh đều 2 bên, size 13, cách dòng 1.5, cách đoạn trên 6pt, cách đoạn dưới 3pt, hàng đâu tiên lùi vào 1.27 cm; hình và bảng soạn theo caption, chèn trích dẫn chéo (Cross-reference) cho bảng và hình; các danh mục hình, bảng, mục lục làm tự động; đánh số trang như file mẫu (bìa không có số trang, danh mục + mục lục số trang theo i, ii, iii…, nội dung chính theo 1,2,3…)

# MỤC LỤC

[LỜI CẢM ƠN i](#_Toc110844453)

[LỜI CAM ĐOAN ii](#_Toc110844454)

[MỤC LỤC iv](#_Toc110844455)

[DANH MỤC HÌNH ẢNH vi](#_Toc110844456)

[DANH MỤC BẢNG BIỂU vii](#_Toc110844457)

[DANH MỤC CÁC TỪ VIẾT TẮT viii](#_Toc110844458)

[LỜI MỞ ĐẦU 1](#_Toc110844459)

[CHƯƠNG 1. Cơ sở lý thuyết 2](#_Toc110844460)

[1.1.1. Khái Niệm kiểm thử phần mềm 2](#_Toc110844461)

[1.1.2. Bảy nguyên tắc trong kiểm thử 2](#_Toc110844462)

[1.1.3. Quy trình kiểm thử cơ bản 2](#_Toc110844463)

[1.2.1. Các mô hình phát triển phần mềm 4](#_Toc110844464)

[1.3. Các cấp độ kiểm thử 8](#_Toc110844465)

[1.3.1. Unit testing(kiểm thử đơn vị) 8](#_Toc110844466)

[1.3.2. Intergration testing(Kiểm thử tích hợp) 9](#_Toc110844467)

[1.3.3. System Testing (Kiểm thử hệ thống) 9](#_Toc110844468)

[1.3.4. Acceptance Testing (Kiểm thử chấp nhận) 10](#_Toc110844469)

[1.4. Các loại kiểm thử 10](#_Toc110844470)

[1.4.3. Kiểm thử chức năng 10](#_Toc110844471)

[1.4.2. Kiểm thử phi chức năng 10](#_Toc110844472)

[1.4.3. Kiểm thử cấu trúc 11](#_Toc110844473)

[1.4.4. Kiểm thử thay đổi 11](#_Toc110844474)

[1.5. Các kĩ thuật kiểm thử 11](#_Toc110844475)

[1.5.2. Thử nghiệm tĩnh 11](#_Toc110844476)

[1.5.3. Kiểm thử động 11](#_Toc110844477)

[CHƯƠNG 2. Phân tích website SwagLabs 14](#_Toc110844478)

[2.1. Giới thiệu về website SwagLabs 14](#_Toc110844479)

[2.1.1. Chức năng của hệ thống 14](#_Toc110844480)

[2.2. Sơ đồ Use case tổng quát 14](#_Toc110844481)

[2.3. Phân tích use case Login 15](#_Toc110844482)

[2.3.1. Sơ đồ Use case cho chức năng Login 15](#_Toc110844483)

[2.3.2. Đặc tả yêu cầu cho chức năng Login 15](#_Toc110844484)

[2.3.3. Sơ đồ hoạt động 16](#_Toc110844485)

[CHƯƠNG 3. Thiết kế use case và thực thi kiểm thử 1](#_Toc110844486)

[CHƯƠNG 4. 1](#_Toc110844487)

[CHƯƠNG 5. KẾT QUẢ 1](#_Toc110844488)

[5.1. Mục 4.1… 1](#_Toc110844489)

[5.2. Mục 4.2… 1](#_Toc110844490)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 2](#_Toc110844491)

[TÀI LIỆU THAM KHẢO 3](#_Toc110844492)

[PHỤ LỤC 4](#_Toc110844493)

*(Mục lục này chỉ là ví dụ)*

# DANH MỤC HÌNH ẢNH

[***Hình 1. Mô hình Waterfall*** 4](#_Toc110849626)

[***Hình 2. Mô Hình V*** 6](#_Toc110849627)

[***Hình 3. Mô hình phát triển Iterative-incremental*** 7](#_Toc110849628)

[***Hình 4. Usecase tổng quát hệ thống*** 14](#_Toc110849629)

[***Hình 5. Màn hình Login*** 15](#_Toc110849630)

[***Hình 6. use case Login*** 15](#_Toc110849631)

[***Hình 7. Sơ đồ hoạt đông của use case Login*** 16](#_Toc110849632)

[Hình 8 Kiến trúc của mô hình 21](#_Toc110849633)

# DANH MỤC BẢNG BIỂU

[Bảng 4.1 Kiến trúc 5](#_Toc74235471)

# DANH MỤC CÁC TỪ VIẾT TẮT

**AI** : Artificial Intelligence

**…**

# LỜI MỞ ĐẦU

1. **Mục tiêu nghiên cứu của đề tài**

* Đề tài này nghiên cứu …

1. **Nhiệm vụ của đề tài**

* Nghiên cứu ...
* ...

1. **Phương pháp nghiên cứu**

* ...

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

* ...

1. **Kết cấu của đề tài**

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

* Mở đầu
* **Chương 1**:
* **Chương 2**:
* **Chương 3**:
* **Chương 4**:
* 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. Kiểm thử thông qua vòng đời phát triển phần mềm

### Các mô hình phát triển phần mềm

#### 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.

#### ****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

## Các cấp độ kiểm thử

### 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.

### 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

### 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:

### 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**

## 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.

## Các kĩ thuật kiểm thử

### Thử nghiệm 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

**1.5.2.1. 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.

**1.5.2.2. 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.

**1.5.2.3. Kỹ thuật dựa trên kinh nghiệm**

Kỹ thuật này rất phụ thuộc vào kiến thức, kỹ năng và kinh nghiệm 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

Swag Labs là trang Web giúp người dùng có thể tiến hành việc mua bán sản phẩm trực tuyến. Cụ thể, khi truy cập vào website này, người dùng có thể xem thông tin sản phẩm, tìm kiếm các sản phẩm cần mua, đặt hàng nhanh chóng.

### Chức năng của hệ thống

Website có những chức năng sau:

* Đăng nhập
* Chọn sản phẩm
* Xóa sản phẩm
* Kiểm tra thông tin
* Đăng xuất

## Sơ đồ Use case tổng quát

Diagram

Description automatically generated

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

## Phân tích use case Login

Graphical user interface, application

Description automatically generated

***Hình 5. Màn hình Login***

### Sơ đồ Use case cho chức năng Login

Diagram

Description automatically generated

***Hình 6. use case Login***

### Đặc tả yêu cầu cho chức năng Login

* 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/ )

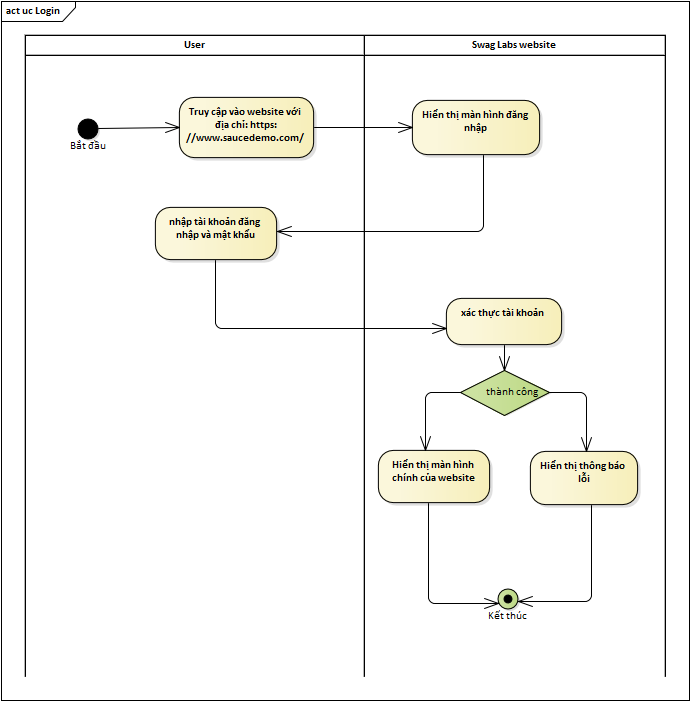
Bước 2: Hệ thống hiển thị màn hình đăng nhập

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 đông của use case Login***

# Thiết kế use case và thực thi kiểm thử

## Thiết kế Test case

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **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 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/ |  |  |
| 1. 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 |
| 1. 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**

# Kết quả kiểm thử

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID Test case | Pass | Fail | N/A | Execute Date |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |
| 4 |  |  |  |  |
| 5 |  |  |  |  |
| 6 |  |  |  |  |
| 7 |  |  |  |  |
| 8 |  |  |  |  |
| 9 |  |  |  |  |

# KẾT QUẢ

# 

## Mục 4.1…

## Mục 4.2…

Kết quả được xây dựng dựa trên ngôn ngữ lập trình thể hiện như (Bảng 4.1).

Bảng 4.1 Kiến trúc

|  |  |  |
| --- | --- | --- |
| Lớp | Ý nghĩa | Tham số |
|  |  |  |
|  |  |  |
|  | | |

Hình 8 Kiến trúc của mô hình

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Đề tài đã thực hiện được …

# TÀI LIỆU THAM KHẢO

* + - 1. Tên tác giả, Tên tác giả… *Tên công trình*. Nơi xuất bản, năm, trang
      2. Tên tác giả, Tên tác giả… *Tên công trình*. Nơi xuất bản, năm, trang

# PHỤ LỤC