**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 TỰ ĐỘNG CHO WEBSITE GURU**

Sinh viên thực hiện : Nguyễn Thị Tường Vi

Lớp : 45K21.1

Đơn vị thực tập : TMA Solutions Bình Định

Cán bộ hướng dẫn : Huỳnh Thanh Nhã

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: ........./ .……./ 202

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

Để hoàn thành bài báo cáo thực tập nghề nghiệp này, trước tiên, em xin bày tỏ sự biết ơn chân thành đến quý thầy cô giảng viên khoa Thống kê - Tin học trường Đại học Kinh tế - Đại học Đà Nẵng đã tạo điều kiện cho em có cơ hội thực tập. Em cũng xin cảm ơn công ty TMA Solutions đã tạo điều kiện cho em có cơ hội được trải nghiệm thực tập tại công ty. Đặc biệt, em xin chân thành gửi lời cảm ơn đến anh Huỳnh Thanh Nhã – Người hướng dẫn trực tiếp, luôn tận tình và giúp đỡ em trong quá trình thực tập.

Trong suốt quá trình tìm hiểu và thực hành, với điều kiện kiến thức và kinh nghiệm chuyên môn còn hạn chế, bài báo cáo này không thể tránh được những thiếu sót. Em rất mong nhận được sự nhận xét, đóng góp ý kiến của các quý thầy cô để em rút ra kinh nghiệm và nâng cao kiến thức bản thân.

Em xin chân thành cảm ơn.

# LỜI CAM ĐOAN

Em xin cam đoan đề tài: “Nghiên cứu và ứng dụng kiểm thử tự động cho website Guru” là một công trình nghiên cứu độc lập không có sự sao chép. Đề tài là một sản phẩm mà em đã hoàn thành sau quá trình nỗ lực nghiên cứu dưới sự hướng dẫn của anh Huỳnh Thanh Nhã và cô Cao Thị Nhâm.

Trong quá trình làm báo cáo có tham khảo một số tài liệu có nguồn gốc rõ ràng.

Các số liệu và kết quả thực hành trong đề tài là trung thực và hoàn toàn không sao chép hay sử dụng kết quả nghiên cứu đề tài của người khác. Nếu phát hiện có gì sai sót trong đề tài, em xin chịu hoàn toàn trách nhiệm.

# MỤC LỤC

[LỜI CẢM ƠN 3](#_Toc110946624)

[LỜI CAM ĐOAN 4](#_Toc110946625)

[MỤC LỤC 5](#_Toc110946626)

[DANH MỤC HÌNH ẢNH 8](#_Toc110946627)

[DANH MỤC BẢNG BIỂU 9](#_Toc110946628)

[DANH MỤC CÁC TỪ VIẾT TẮT 10](#_Toc110946629)

[LỜI MỞ ĐẦU 11](#_Toc110946630)

[CHƯƠNG 1. TỔNG QUAN VỀ CÔNG TY TMA SOLUTIONS BÌNH ĐỊNH 12](#_Toc110946631)

[CHƯƠNG 2. TỔNG QUAN VỀ CƠ SỞ LÝ THUYẾT 13](#_Toc110946632)

[2.1. Tổng quan về kiểm thử phần mềm 13](#_Toc110946633)

[2.1.1. Kiểm thử phần mềm là gì? 13](#_Toc110946634)

[2.1.2. Mục tiêu của kiểm thử phần mềm 13](#_Toc110946635)

[2.1.3. Vòng đời kiểm thử phần mềm là gì? 13](#_Toc110946636)

[2.1.4. Nguyên tắc kiểm thử 14](#_Toc110946637)

[2.1.5. Error/ Fault/ Failure/ Bug/ Defect 14](#_Toc110946638)

[2.1.6. Xác minh – Xác thực 15](#_Toc110946639)

[2.1.7. QA – QC 15](#_Toc110946640)

[2.2. Vòng đời phát triển phần mềm (SDLC) 15](#_Toc110946641)

[2.2.1. SDLC là gì 15](#_Toc110946642)

[2.2.2. Các mô hình của SDLC 16](#_Toc110946643)

[2.3. Hình thức và phương pháp kiểm thử phần mềm 18](#_Toc110946644)

[2.3.1. Hình thức kiểm thử phần mềm 18](#_Toc110946645)

[2.3.2. Phương pháp kiểm thử phần mềm 19](#_Toc110946646)

[2.4. Cấp độ của kiểm thử phần mềm 19](#_Toc110946647)

[2.4.1. Kiểm thử đơn vị 19](#_Toc110946648)

[2.4.2. Kiểm thử tích hợp 20](#_Toc110946649)

[2.4.3. Kiểm thử hệ thống 20](#_Toc110946650)

[2.4.4. Kiểm thử chấp nhận 20](#_Toc110946651)

[2.5. Test case 20](#_Toc110946652)

[2.5.1. Khái niệm 20](#_Toc110946653)

[2.5.2. Kỹ thuật thiết kế Test case 21](#_Toc110946654)

[2.6. Robot Framework 22](#_Toc110946655)

[2.6.1. Giới thiệu về Robot Framework 22](#_Toc110946656)

[2.6.2. Các tính năng nổi bật của Robot Framework 22](#_Toc110946657)

[2.6.3. Các thư viện hỗ trợ trong Robot Framwork 23](#_Toc110946658)

[2.7. Selenium 23](#_Toc110946659)

[2.7.1. Giới thiệu về Selenium 23](#_Toc110946660)

[2.7.2. Các thành phần của Selenium 24](#_Toc110946661)

[2.8. Website hỗ trợ kiểm thử tự động 26](#_Toc110946662)

[CHƯƠNG 3. PHÂN TÍCH WEBSITE GURU99 BANK 27](#_Toc110946663)

[3.1. Tổng quan về webstite Guru99 Bank 27](#_Toc110946664)

[3.1.1. Sơ đồ Use case tổng quát 27](#_Toc110946665)

[3.1.2. Vai trò của từng tác nhân 28](#_Toc110946666)

[3.2. Phân tích Use case “Đăng nhập” 28](#_Toc110946667)

[3.2.1. Sơ đồ Use case chi tiết cho chức năng “Đăng nhập” 28](#_Toc110946668)

[3.2.2. Sơ đồ hoạt động của chức năng “Đăng nhập” 29](#_Toc110946669)

[3.2.3. Đặc tả yêu cầu cho chức năng “Đăng nhập” 29](#_Toc110946670)

[CHƯƠNG 4. THIẾT KẾ TEST CASE VÀ THỰC HIỆN KIỂM THỬ 31](#_Toc110946671)

[4.1. Thiết kế Test case 31](#_Toc110946672)

[4.2. Thực hiện kiểm thử 33](#_Toc110946673)

[4.3. Kết quả kiểm thử 33](#_Toc110946674)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 35](#_Toc110946675)

[TÀI LIỆU THAM KHẢO 36](#_Toc110946676)

# DANH MỤC HÌNH ẢNH

[*Hình 1. Mô hình Waterfall* 16](#_Toc110928147)

[*Hình 2. Mô hình chữ V* 17](#_Toc110928148)

[*Hình 3. Mô hình Agile* 17](#_Toc110928149)

[*Hình 4. Phương pháp Scrum* 18](#_Toc110928150)

[*Hình 5. Giao diện website Guru 99* 26](#_Toc110928151)

[*Hình 6. Use case tổng quát hệ thống* 27](#_Toc110928152)

[*Hình 7. Use case “Đăng nhập”* 28](#_Toc110928153)

[*Hình 8. Sơ đồ hoạt động của Usecase “Đăng nhập”* 29](#_Toc110928154)

[*Hình 9. Code thực hiện kiểm thử* 33](#_Toc110928155)

[*Hình 10. Giao diện sau khi thực hiện kiểm thử* 33](#_Toc110928156)

[*Hình 11. Thống kê kiểm thử* 34](#_Toc110928157)

[*Hình 12. Nhật ký thực thi kiểm thử* 34](#_Toc110928158)

[*Hình 13. Báo cáo kiểm thử* 34](#_Toc110928159)

# DANH MỤC BẢNG BIỂU

[*Bảng 1. Xác minh và Xác thực* 15](#_Toc110928160)

[*Bảng 2. Đảm bảo chất lượng và Kiểm soát chất lượng* 15](#_Toc110928161)

[*Bảng 3. Ưu và nhược điểm của Selenium IDE* 24](#_Toc110928162)

[*Bảng 4. Ưu và nhược điểm của Selenium RC* 24](#_Toc110928163)

[*Bảng 5. Ưu và nhược điểm của Selenium WebDriver* 25](#_Toc110928164)

[*Bảng 6. Vai trò của từng tác nhân* 28](#_Toc110928165)

[*Bảng 7. Thiết kế Test case “Đăng nhập”* 31](#_Toc110928166)

# DANH MỤC CÁC TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **STT** | **Chữ viết tắt** | **Chữ viết đầy đủ** |
| **1** | STLC | Software Testing Life Cycle |
| **2** | QA | Quanlity Assurance |
| **3** | QC | Quanlity Control |
| **4** | SDLC | Software Development Life Cycle |
| **5** | API | Application Programming Interface |
| **6** | EP | Equivalence Partitioning |
| **7** | BVA | Boundary Value Analysis |
| **8** | RPA | Robotic Process Automation |
| **9** | IDE | Integrated Development Environment |
| **10** | HTML | HyperText Markup Language |
| **11** | DOM | Document Object Model |

# 

# 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 Guru.

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 Guru
* 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: Tổng quan về công ty TMA Solutions Bình Định
* Chương 2: Tổng quan về cơ sở lý thuyết
* Chương 3: Phân tích website Guru99 Bank
* Chương 4: Thiết kế test case và thực hiện kiểm thử
* Kết luận và hướng phát triển

# TỔNG QUAN VỀ CÔNG TY TMA SOLUTIONS BÌNH ĐỊNH

Được thành lập năm 1997, TMA là tập đoàn công nghệ hàng đầu Việt Nam với 3500 kỹ sư và khách hàng là những tập đoạn công nghệ cao hàng đầu thế giới từ 30 quốc gia. TMA hiện có 7 chi nhanh tại Việt Nam (6 tại Tp.HCM và 1 ở Tp.Quy Nhơn) cùng 6 chi nhanh ở nước ngoài (Mỹ, Úc, Canada, Đức, Nhật, Singapore)

Tháng 6 năm 2018, TMA đã mở chi nhánh tại Bình Định. Sau 4 năm, TMA Bình Định đã phát triển nhanh chóng với 400 kỹ sư, trong đó có nhiều kỹ sư đang làm việc tại TP.HCM đã trở về làm việc tại quê hương.

Tháng 8 năm 2018, TMA đã khởi công xây dựng Công viên Sáng tạo TMA Bình Định (TMA Innovation Park – TIP) trên 10 hecta tại Thung lũng Sáng Tạo Quy Nhơn (Quy Nhon Innovation Park – QNIVY) với vốn đầu tư hàng trăm tỷ đồng.

Là trung tâm phần mềm đầu tiên tại Thung lũng Sáng tạo Quy Nhơn, Công viên Sáng tạo TMA mang sứ mệnh trở thành trung tâm phát triển phần mềm và công nghệ cao hàng đầu tại miền trung, góp phần quan trọng đưa Thung lũng sáng tạo Quy Nhơn trở thành một điểm đến của công nghệ 4.0 tại Việt Nam.

# TỔNG QUAN VỀ CƠ SỞ LÝ THUYẾT

## Tổng quan về kiểm thử phần mềm

### Kiểm thử phần mềm là gì?

Kiểm thử phần mềm là hoạt động nhằm tìm kiếm và phát hiện ra các lỗi của phần mềm, đảm bảo phần mềm chính xác, đúng và đầy đủ theo yêu cầu của khách hàng, yêu cầu của sản phẩm đặt ra.

### Mục tiêu của kiểm thử phần mềm

* Tìm các lỗi phát sinh do người phát triển tạo ra khi code.
* Đạt được sự tin tưởng và cung cấp thông tin về mức độ chất lượng.
* Ngăn ngừa lỗi.
* Đảm bảo kết quả cuối cùng đáp ứng các yêu cầu kinh doanh và người sử dụng.
* Đạt được tự tín nhiệm của khách hàng bằng cách cung cấp cho họ một sản phẩm chất lượng.

### Vòng đời kiểm thử phần mềm là gì?

* Vòng đời kiểm thử phần mềm (STLC) là quá trình kiểm thử được thực hiện một cách có hệ thống và có kế hoạch. Trong quá trình STLC, các hoạt động khác nhau được thực hiện để cải thiện chất lượng sản phầm.
* Các giai đoạn của STLC: có 6 giai đoạn
  + - * Phân tích yêu cầu
      * Lập kế hoạch phân tích
      * Phát triển trường hợp kiểm thử
      * Thiết lập môi trường
      * Thực hiện kiểm thử
      * Đóng chu kì kiểm thử

### Nguyên tắc kiểm thử

Có 7 nguyên tắc kiểm thử

* Kiểm thử chứng minh sự hiện diện của lỗi
* Kiểm thử toàn diện là không thể
* Kiểm tra càng sớm càng tốt
* Lỗi thường được phân bố tập trung
* Nguyên lý thuốc trừ sâu
* Kiểm thử phụ thuộc vào ngữ cảnh
* Sự sai lầm về việc không có lỗi

### Error/ Fault/ Failure/ Bug/ Defect

* ***Error:*** Là hành động của con người dẫn đến kết quả sai.
* ***Fault:*** Là trạng thái phần mềm gây ra bởi Error.
* ***Failure:*** Lỗi khi có kết quả sai lệch so với yêu cầu đặc tả, là sự khác biệt giữa kết quả thực tế trên màn hình và kết quả mong đợi của một thành phần hoặc một hệ thống nào đó.
* ***Bug:*** Là một khiếm khuyết trong một thành phần hoặc hệ thống mà nó có thể làm cho thành phần hoặc hệ thống này không thực hiện đúng chức năng yêu cầu của nó.
* ***Defect:*** Là lỗi trong quá trình phát triển hoặc lỗi logic làm cho chương trình hoạt động sai yêu cầu đề ra.

### ­Xác minh – Xác thực

*Bảng 1. Xác minh và Xác thực*

|  |  |
| --- | --- |
| **Xác minh (Verification)** | **Xác thực (Validation)** |
| Là quá trình xác nhận rằng phần mềm đáp ứng đặc điểm kỹ thuật của nó, được thực hiện thông qua kiểm tra và hướng dẫn.  Trả lời cho câu hỏi: Sản phầm có đúng hay không? | Là quá trình xác nhận rằng phần mềm đáp ứng yêu cầu người dùng. Đó là thử nghiệm thực tế.  Trả lời cho câu hỏi: đó có phải là sản phẩm phù hợp hay không? |

### QA – QC

*Bảng 2. Đảm bảo chất lượng và Kiểm soát chất lượng*

|  |  |
| --- | --- |
| **Đảm bảo chất lượng (QA)** | **Kiểm soát chất lượng (AC)** |
| Là một quy trình được cân nhắc thận trọng nằm cung cấp sự đảm bảo rằn phần mềm sẽ vượt qua được những yêu cầu về chất lượng. | Là quy trình kiểm tra sự hoàn thành của các yêu cầu về chất lượng phần mềm. |
| Mục tiêu: Ngăn ngừa khiếm khuyết. | Mục tiêu: Xác định và cải thiện các khiếm khuyết |
| Đảm bảo rằng những điều đang làm là đúng những điều phài làm. | Đảm bảo kết quả của những gì đã làm là những gì mong đợi. |
| Là quy trình để tạo ra phần mềm. | Là quy trình để xác minh phần mềm. |

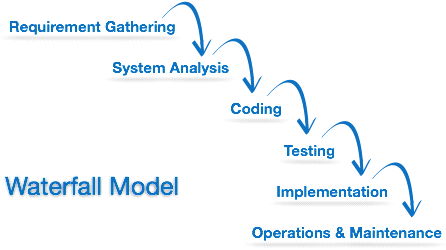
## Vòng đời phát triển phần mềm (SDLC)

### SDLC là gì

* SDLC là một quy trình được ngành công nghiệp phần mềm sử dụng để thiết kế, phát triển và kiểm tra phần mềm chất lượng cao. Mục đích tạo ra một phần mềm chất lượng cao đáp ứng hoặc vượt quá mong đợi của khách hàng, hoàn thành trong thời gian và chi phí ước tính.
* Các giai đoạn của SDLC: 6 giai đoạn
  + - * Thu thập và phân tích yêu cầu
      * Thiết kế
      * Thực hiện hoặc coding
      * Kiểm thử
      * Triển khai
      * Duy trì

### Các mô hình của SDLC

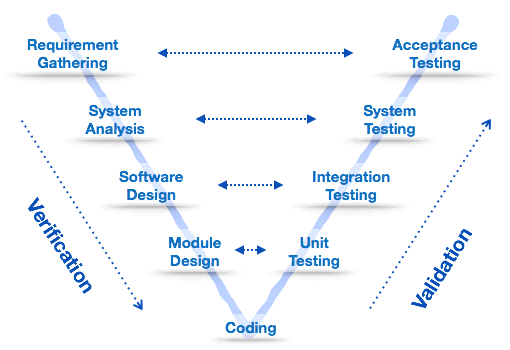
##### Mô hình Waterfall



*Hình 1. Mô hình Waterfall*

* Đây là phương pháp lâu đời nhất và đơn giản nhất của phương pháp SDLC. Các giai đoạn sẽ được thực hiện tuần tự nối tiếp nhau. Giai đoạn sau chỉ được thực hiện khi giai đoạn trước đã kết thúc và không được quay lại giai đoạn trước để xử lý các yêu cầu khi muốn thay đổi.
* Thường được áp dụng cho các dự án không thường xuyên bị thay đổi về yêu cầu.

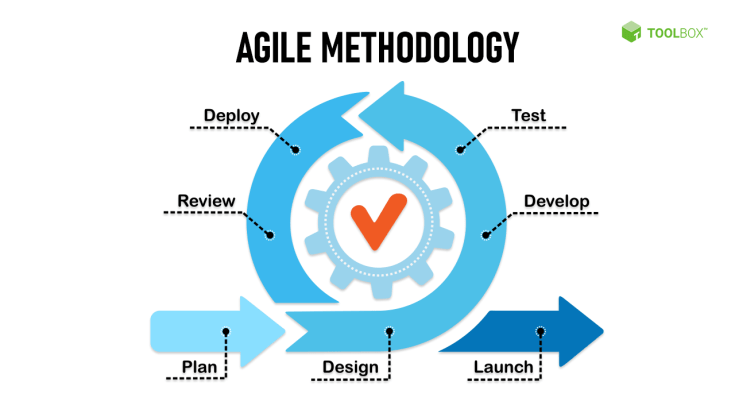
##### Mô hình chữ V



*Hình 2. Mô hình chữ V*

* Đây là mô hình mở rộng từ mô hình thác nước, toàn bộ quy trình được chia làm hai nhánh: Phát triển và Kiểm thử. Mỗi giai đoạn phát triển sẽ tiến hành song song với một giai đoạn kiểm thử tương ứng, do đó, các lỗi sẽ được phát hiện sớm ngay từ đầu.
* Thường được áp dụng cho các dự án ngắn và có yêu cầu rõ ràng ít có sự thay đổi, công nghệ sử dụng không thay đổi và được hiểu rõ hởi nhóm dự án.

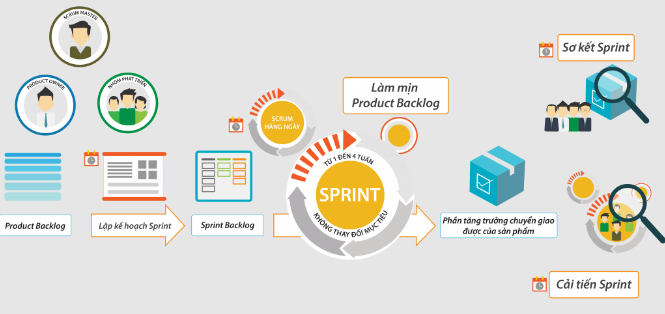
##### Mô hình Agile



*Hình 3. Mô hình Agile*

* Agile là một phương pháp phát triển phần mềm linh hoạt để làm sao đưa sản phẩm đến tay người dùng càng nhanh càng tốt và được xem như là sự cải tiến so với những mô hình cũ.
* Mô hình này được ứng dụng với bất kỳ loại hình dự án nào, nhưng cần sự tham gia và tính tương tác của khách hàng. Được sử dụng khi khách hàng yêu cầu chức năng sẵn sàng trong khoảng thời gian ngắn.

##### Phương pháp Scrum



*Hình 4. Phương pháp Scrum*

* Scrum là 1 dạng của mô hình Agile và là framework phổ biến nhất khi thực hiện mô hình Agile. Scrum là mô hình phát triển lặp đi lặp lại, những khoảng lặp cố định thường kéo dài 1 đến 2 tuần được gọi là Sprint.
* Thường phù hợp với những dự án có sự thay đổi về yêu cầu hoặc nghiệp vụ của khách hàng, làm theo giai đoạn ngắn, có thể nhìn thấy những rủi ro, hay những điểm chưa phù hợp để thay đổi.

## Hình thức và phương pháp kiểm thử phần mềm

### Hình thức kiểm thử phần mềm

* ***Kiểm thử thủ công:*** Là hình thức kiểm thử phần mềm trong đó các trường hợp kiểm thử được người kiểm thử thực hiện theo cách thủ công mà không sử dụng bất kỳ công cụ tự động nào. Có các giai đoạn khác nhau để kiểm thử thủ công như kiểm thử đơn vị, kiểm thử tích hợp, kiểm thử hệ thông và kiểm thử chấp nhận người dùng.
* ***Kiểm thử tự động:*** Là hình thức kiểm thử phần mềm thực hiện bằng cách sử dụng các công cụ phần mềm kiểm thử tự động đặc biệt để thực thi một trường hợp kiểm thử. Kiểm thử tự động được dùng để chạy lại các kịch bản kiểm thử mà đã được thực hiện một cách thủ công, nhanh chóng và lặp đi lặp lại.
* ***Kiểm thử bảo mật:*** Là hình thức kiểm thử phần mềm nhằm phát hiện ra các lỗ hổng, mối đe dọa, rủi ro trong một ứng dụng phần mềm và ngăn chặn các cuộc tấn công nguy hiểm từ những kẻ xâm nhập.
* ***Kiểm thử API:*** Là hình thức kiểm thử phần mềm liên quan đến việc kiểm thử các giao diện lập trình ứng dụng một cách trực tiếp và là một phần của kiểm thử tích hợp để xác định xem hệ thống có đáp ứng các yêu cầu về tính năng, độ tin cậy, hiệu suất và bảo mật.

### Phương pháp kiểm thử phần mềm

* ***Kiểm thử hộp đen:*** Là phương pháp kiểm thử mà người kiểm thử không cần quan tâm đến các hoạt động bên trong hệ thống chạy ra sao, không cần quan tâm đến các dòng lệnh bên trong hệ thống như thế nào mà chỉ cần tập trung và các giá trị đầu vào và các giá trị đầu ra của hệ thống có đúng với kết quả mong đợi của các trường hợp kiểm thử không để từ đó đánh giá chất lượng hệ thống.
* ***Kiểm thử hộp trắng:*** Là việc nghiên cứu từng chi tiết luồng hoạt động cũng như các dòng lệnh bên trong hệ thống. Kiểm thử hộp trắng cũng được gọi là Kiểm thử kính hay Kiểm thử hộp mở.
* ***Kiểm thử hộp xám:*** Là phương pháp kiểm thử đòi hỏi người kiểm thử có một lượng kiến thức nhất định về các luồng hoạt động bên trong hệ thống.

## Cấp độ của kiểm thử phần mềm

### Kiểm thử đơn vị

* Là cấp độ kiểm thử cơ bản, thực hiện kiểm thử từng module nhỏ trong hệ thống.
* Mục đích: Để xác nhận mỗi thành phần của phần mềm thực hiện đúng với thiết kế.
* Thường do lập trình viên thực hiện.

### Kiểm thử tích hợp

* Là tích hợp kiểm tra các module riêng lẻ với nhau thành một nhóm.
* Mục đích: Để đảm bảo rằng hệ thống tích hợp đã sẵn sàng để kiểm thử hệ thống.
* Được thực hiện sau khi kiểm thử đơn vị và trước khi kiểm thử hệ thống.
* Một số phương pháp:
  + - * Phương pháp kiểm thử Bigbag
      * Phương pháp kiểm thử Topdown
      * Phương pháp kiểm thử Bottom up
      * Phương pháp kiểm thử Sandwich

### Kiểm thử hệ thống

* Là thực hiện kiểm thử một hệ thống đã được tích hợp hoàn chỉnh để xác minh rằng nó đúng với yêu cầu phần mềm.
* Thường là thử nghiệm cuối cùng để xác minh rằng hệ thống phân phối đáp ứng được các đặc điểm kỹ thuật và mục đích của nó.

### Kiểm thử chấp nhận

* Sau khi kiểm thử hệ thống đã sửa tất cả hoặc hầu hết các lỗi, hệ thống sẽ được gửi đến người dùng hoặc khác hàng để kiểm thử chấp nhận.
* Mục đích: Đảm bảo phần mềm đáp ứng đúng yêu cầu của khách hàng.
* Được chia thành hai mức: Kiểm thử Alpha và Kiểm thử Beta.

## Test case

### Khái niệm

Test case (Kịch bản kiểm thử) là một tập hợp các hành động được thực thi để xác minh một chức năng, một hệ thống phần mềm có hoạt động đúng hay không.

### Kỹ thuật thiết kế Test case

* ***Kiểm thử tĩnh:*** Là phương pháp kiểm thử phần mềm trong đó ứng dụng phần mềm được kiểm tra mà không cần thực thi code. Mục tiêu là cải thiện chất lượng của các ứng dụng phần mềm bằng cách tìm ra các lỗ hổng trong giai đoạn đầu của quá trình phát triển. Kỹ thuật kiểm thử tĩnh:
  + - * Đánh giá không chính thức (Informal reviews)
      * Đánh giá kỹ thuật (Technical reviews)
      * Hướng dẫn (Walkthrough)
      * Kiểm tra (Inspection)
      * Đánh giá code tính (Static code review)
* ***Kiểm thử động:*** Là một phương pháp kiểm thử phần mềm được sử dụng để kiểm tra hành vi động của code phần mềm. Mục tiêu là xác nhận sản phẩm phần mềm hoạt động phù hợp với yêu cầu kinh doanh. Kỹ thuật kiểm thử động:
  + - * *Kỹ thuật kiểm thử dựa trên đặc điểm kỹ thuật:*
* Phân vùng tương đương (EP)
* Phân tích giá trị biên (BE)
* Bảng quyết định
* Chuyển đổi trạng thái
  + - * *Kỹ thuật kiểm thử dựa trên cấu trúc:*
* Kiểm thử câu lệnh
* Kiểm thử quyết định
* Kiểm thử điều kiện
* Kiểm thử đa điều kiện
  + - * *Kỹ thuật kiểm thử dựa trên kinh nghiệm:*
* Kiểm thử thăm dò
* Phỏng đoán lỗi

## Robot Framework

### Giới thiệu về Robot Framework

* Robot Framework là một khung tự động mã nguồn mở chung, nó có thể được sử dụng để tự động hóa kiểm thử và tự động hóa quy trình bằng robot (RPA).
* Robot Framework có thể được tích hợp với hầu như bất kỳ công cụ nào khác để tạo ra các giải pháp tự động hóa mạnh mẽ và linh hoạt.
* Robot Framework tuân theo các kiểu trường hợp kiểm thử khác nhau – theo hướng từ khóa (keyword – driven), hướng hành vi (behaviour-driven) và hướng dữ liệu (data-driven) để viết các trường hợp kiểm thử. Khả năng của nó có thể được mở rộng bởi các thư viện được triển khai bằng Python, Java hoặc nhiều ngôn ngữ lập trình khác.
* Thư viện phổ biến nhất được sử dụng với Robot Frameword là Selenium Libarary được sử dụng để phát triển web và kiểm tra giao diện người dùng.

### Các tính năng nổi bật của Robot Framework

* Robot Framework giúp chúng ta thực hiện kiểm thử tự động với kịch bản ở dạng bảng một cách dễ dàng. Robot Framework đưa ra kết quả thực thi các kịch bản kiểm thử và các log ở dạng html, giúp chúng ta đọc và phân tích kết quả nhanh chóng và dễ dàng hơn.
* Hỗ trợ chức năng đánh dấu các kịch bản kiểm thử, cho phép chúng ta lựa chọn kịch bản kiểm thử tiện lợi và nhanh chóng.
* Thế mạnh của Robot Framework là khả năng chạy trên nhiều hệ điều hành khác nhau mà không cần chỉnh sửa kịch bản kiểm thử hay các từ khóa tầng dưới.

### Các thư viện hỗ trợ trong Robot Framwork

* ***Selenium2Library:***
* Được sử dụng để kiểm thử trên nền tàng Web, và được fork từ SeleniumLibrary và được bổ sung để sử dụng Selnium 2 và WebDriver.
* Hoạt động ở hầu hết các trình duyệt hay dùng như IE, FireFox, Chrome,.. và có thể được dùng với các Python và cả Jython.
* ***Calculator Library:*** Là một thư viện được có sẵn trong Robot Framework, một thư viện về tính toán đơn giản, nó chỉ chưa logic nghiệp vụ chứ không bao gồm phần UI.

## Selenium

### Giới thiệu về Selenium

* Selenium là một bộ công cụ phần mềm chuyên dụng được dùng để kiểm thử tự động các ứng dụng web và có khả năng hỗ trợ chạy trên trình duyệt với nhiều nền tảng như Windowns, Linux, Mac,..
* Các tính năng của Selenium:
* Selenium là mootjc ông cụ mã nguồn mở/ frameworkk để kiểm tra web, webiste phiên bản tự động.
* Selenium IDE hỗ trợ tính năng playback giúp bạn có thể sử dụng các bài test của người khác và không cần phải biết ngôn ngữ script.
* Là một nền tảng kiểm thử dựa trên cloud giúp tester có thể lưu lại thao tác và xuất ra dưới dạng script đơn giản, dễ hiểu.
* Hỗ trợ nhiều hệ điều hành, ngôn ngữ và trình duyệt khác nhau.
* Quá trình kiểm thử của Selenium hao tốn ít tài nguyên và yêu cầu cấu hình thiết bị thấp hơn so với các công cụ khác.

### Các thành phần của Selenium

* ***Selenium IDE:*** Là một framework đơn giản nhất trong bộ Selenium và là cách đơn giản nhất để học. Đây là một plugin của Firefox mà có thể cài đặt dễ dàng như các plugin khác. Tuy nhiên, vì tính đơn giản của nó, Selenium IDE chỉ nên sử dụng như một công cụ tạo mẫu.

*Bảng 3. Ưu và nhược điểm của Selenium IDE*

|  |  |
| --- | --- |
| ***Ưu điểm*** | ***Nhược điểm*** |
| Dễ dàng cái đặ và sử dụng | Chỉ có trong Firefox |
| Không yêu cầu kinh nghiệm lập trình, chỉ cần có kiến thức HTML và DOM | Thiết kế chỉ để tạo mẫu của kiểm thử |
| Có thể export test sang định dạng được sử dụng trong Selenium RC và Webdriver. | Không cung cấp sự lặp lại hoặc câu lệnh có điều kiện cho tập lệnh kiểm tra. |
| Được xây dựng để có thể báo cáo kết quả | Thực hiện kiểm thử chậm so với Selnium RC và Webdriver. |

* ***Selenium Romote Control (Selenium RC):*** là một framwork kiểm thử hàng đầu của toàn bộ dự án Selenium trong một thời gian dài. Đây là công cụ kiểm tra web tự động đầu tiên cho phép người dùng sử dụng ngôn ngữ lập trình mà họ thích.

*Bảng 4. Ưu và nhược điểm của Selenium RC*

|  |  |
| --- | --- |
| ***Ưu điểm*** | ***Nhược điểm*** |
| Nhiều trình duyệt và đa nền tảng | Cài đặt phức tạp hơn IDE |
| Có thể hỗ trợ cho việc lặp lại hoặc câu lệnh có điều kiện cho tập lệnh kiểm tra | Yêu cầu phải có kiến thức lập trình |
| Có thể hỗ trợ kiểm tra theo dữ liệu | Cấn có Seleium RC Server để chạy |
| Thực hiện nhanh hơn IDE | Thời gian thực thi chậm hơn Web Driver |
| Có thể sẵn sàng hỗ trợ các trình duyệt mới | Tương tác với trình duyệt ít hơn thực tế |

* ***Selenium WebDriver:*** Được thực hiện một cách hiện đại và ổn định hơn trong tự động hóa các hành động của trình duyetek. WebDriver không giống như Selenium RC, không dựa vào JavaScript dành cho tự động hóa. Nó điều khiển trình duyệt bằng cách trực tiếp liên lạc với nó

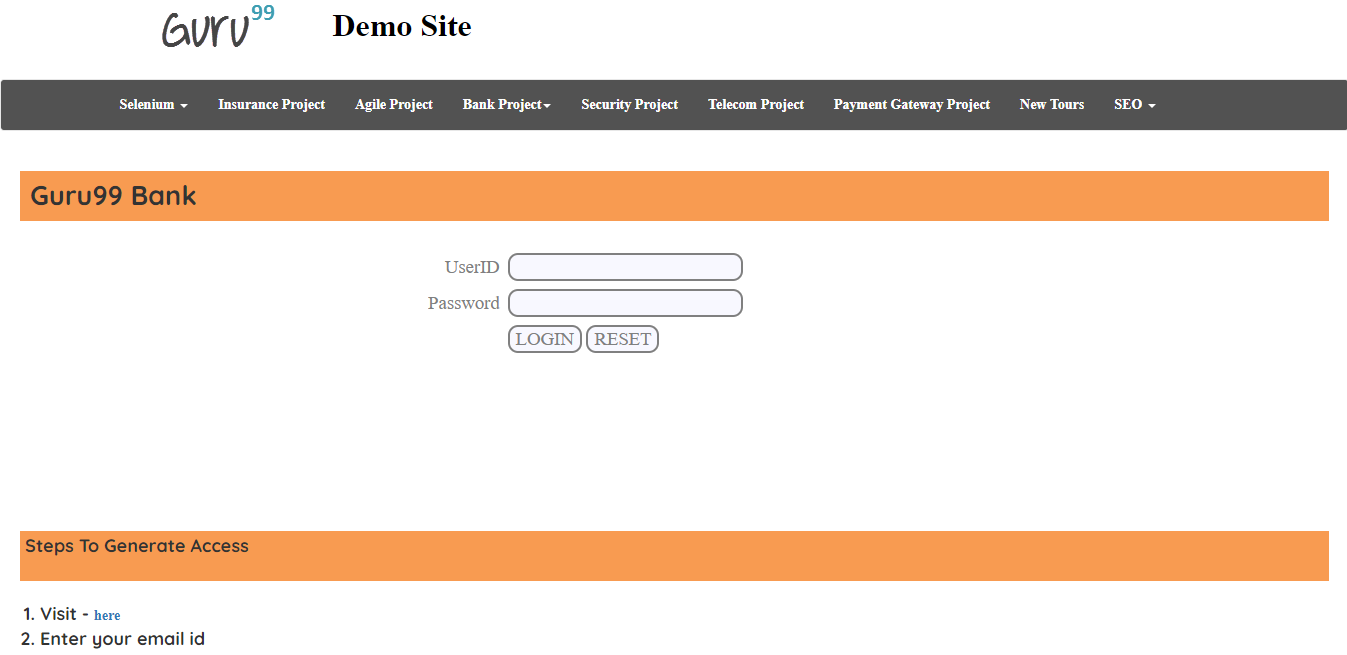
*Bảng 5. Ưu và nhược điểm của Selenium WebDriver*

|  |  |
| --- | --- |
| ***Ưu điểm*** | ***Nhược điểm*** |
| Cài đặt đơn giản hơn Selenium RC | Lắp đặt phức tạo hơn Selenium IDE |
| Giao tiếp trực tiếp với trình duyệt | Yêu cầu kiến thức lập trình |
| Sự tương tác của trình duyệt thực tế hơn | Không thể sẵn sàng hỗ trợ các trình duyệt mới |
| Không cần một thành phần riêng biệt như RC Server | Không có cơ chế tích hợp để ghi lại các thông điệp thời gian chạy và tạo ra các kết quả kiểm tra |

* ***Selenium Grid:*** Là một công cụ được sử dụng cùng với Selenium RC để chạy thử nghiệm song song trên các máy khác nhau và các trình duyệt khác nhau cùng một lúc.

## Website hỗ trợ kiểm thử tự động

Website Guru99



*Hình 5. Giao diện website Guru 99*

# PHÂN TÍCH WEBSITE GURU99 BANK

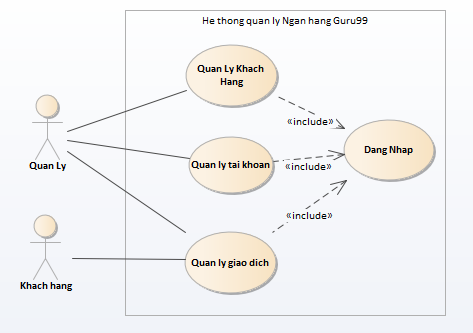
## Tổng quan về webstite Guru99 Bank

Website Guru99 Bank được xây dựng nhằm tối ưu hóa việc thực hiện các giao dịch tại các ngân hàng. Hệ thống cung cấp cho các quản lý ngân hàng một nền tảng quản lý tài khoản, giao dịch, khách hàng một cách chuyên nghiệp, bên cạnh đó, khách hàng cũng dễ dàng kiểm soát các giao dịch của mình.

### Sơ đồ Use case tổng quát

Các tác nhân chính của hệ thống bao gồm:

* Quản lý
* Khách hàng



*Hình 6. Use case tổng quát hệ thống*

### Vai trò của từng tác nhân

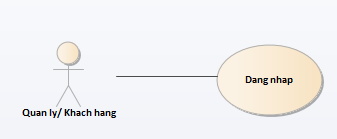
*Bảng 6. Vai trò của từng tác nhân*

|  |  |  |
| --- | --- | --- |
| STT | Tác nhân | Chức năng |
| 1 | Quản lý | Quản lý, theo dõi và tiếp nhận các yêu cầu từ phía khách hàng. Quản lý có thể thực hiện các chức năng:   * Quản lý Khách hàng * Quản lý Tài khoản khách hàng * Quản lý Giao dịch * Đăng nhập/ Đăng xuất |
| 2 | Khách hàng | Khách hàng là người sử dụng hệ thống, thực hiện quản lý các giao dịch của mình như:   * Kiểm tra số dư * Chuyển tiền * Xuất báo cáo sao kê * Đăng nhập/ Đăng xuất |

Trong bài báo cáo này, em tiến hành lập kế hoạch và thực hiện kiểm thử Use case “Đăng nhập”.

## Phân tích Use case “Đăng nhập”

### Sơ đồ Use case chi tiết cho chức năng “Đăng nhập”



*Hình 7. Use case “Đăng nhập”*

### Sơ đồ hoạt động của chức năng “Đăng nhập”



*Hình 8. Sơ đồ hoạt động của Usecase “Đăng nhập”*

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

* ***Tổng quan:*** Thực hiện xác thực người dùng.
* ***Tác nhân:*** Quản lý/ Khách hàng
* ***Sự kiện kích hoạt:*** Khi người dùng bắt đầu sử dụng hệ thống
* ***Luồng xử lý:***
* Bước 1: Người dùng mở website
* Bước 2: Hệ thống hiển thị màn hình đăng nhập
* Bước 3: Người dùng nhập tên truy cập và mật khẩu, sau đó chọn nút Đăng nhập
* Bước 4: Hệ thống thực hiện xác thực tài khoản
* Bước 5: Nếu xác thực thành công, hệ thống hiển thị màn hình chính. Ngược lại:
* Nếu Tên đăng nhập để trống, thì hiển thị thông báo lỗi “User-ID must not be blank”
* Nếu Mật khẩu để trống, thì hiển thị thông báo lỗi “Password must not be blank”
* Nếu Tên đăng nhập hoặc Mật khẩu không đúng, thì hiển thị lỗi “User or Password is not valid”

# THIẾT KẾ TEST CASE VÀ THỰC HIỆN KIỂM THỬ

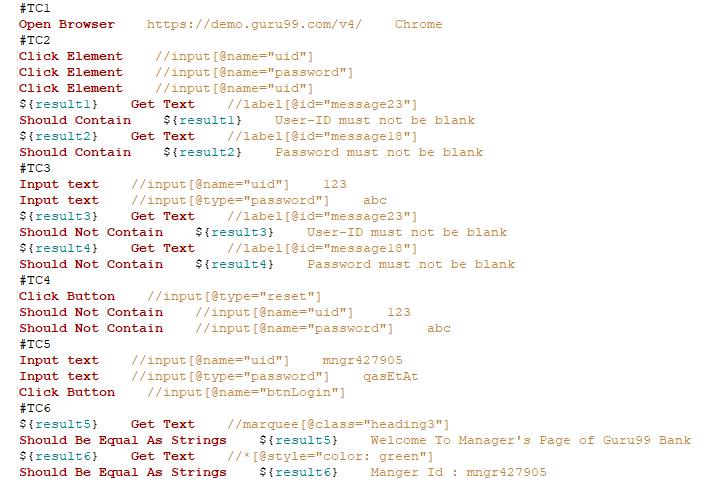
## Thiết kế Test case

*Bảng 7. Thiết kế Test case “Đăng nhập”*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test scenario** | **Test case descriptions** | **Test steps** | **Test data** | **Expected Result** | **Result** |
| **Verify the Login Section** | **TC1:** Go to  <https://demo.guru99.com/v4/> | Go to  <https://demo.guru99.com/v4/> |  | Display the login interface of the Guru99 Bank website | Pass |
| **TC2:** Verify text 'User-ID must not be blank' and 'Password must not be blank' if UserID or Password field is empty | * Click Element User ID * Click Element Password * Click Element UserID | UserID = empty  Password = empty | Show message ‘User-ID must not be blank’ and ‘Password must not be blank’ | Pass |
| **TC3:** Verify text 'User-ID must not be blank' and 'Password must not be blank' are disappeared | * Input User ID * Input Password | UserID = 123  Password = abc | Input UserID and Password successful  Message ‘User-ID must not be blank’ and ‘Password must not be blank’ are disappeared | Pass |
| **TC4:** Verify userID and password is cleared | Click button Reset |  | UserID and Password empty | Pass |
| **TC5:** Verify login to home page successful | * Input valid User ID * Input valid Password * Click button Login | UserID = mngr427905  Password = qasEtAt | Display successful login screen | Pass |
| **TC6:** Verify dynamic userID in home page |  |  | The UserID on the homepage is the UserID used for login. | Pass |

## Thực hiện kiểm thử

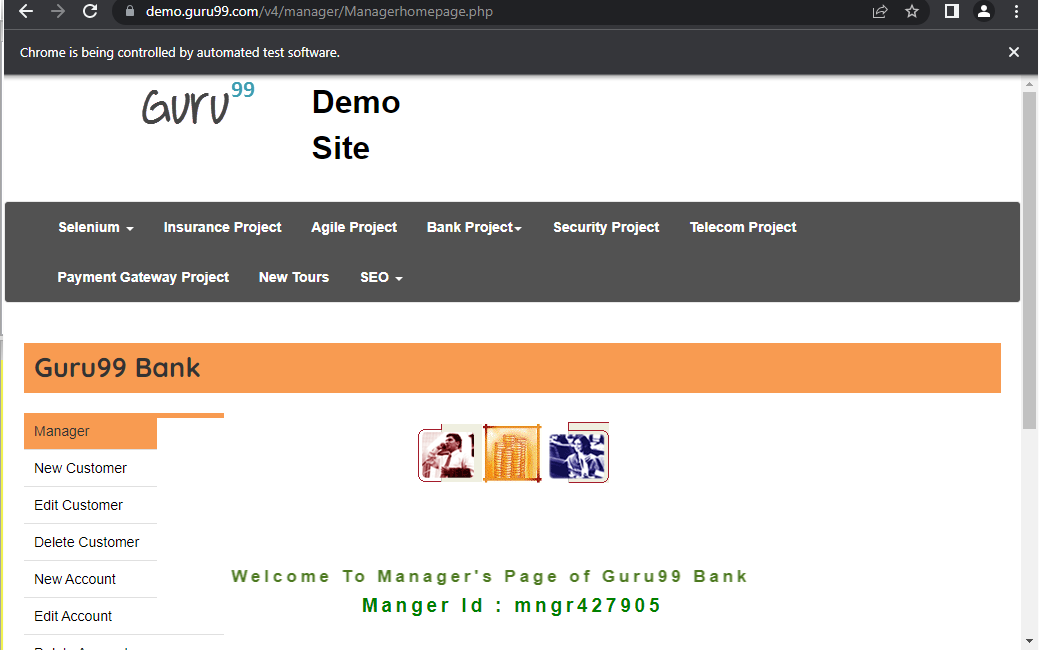
Viết code trên công cụ RIDE – Robot Framework để thực hiện kiểm thử.



*Hình 9. Code thực hiện kiểm thử*

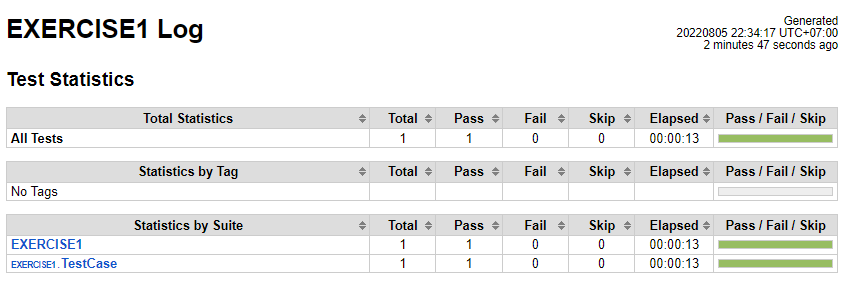
## Kết quả kiểm thử

* Giao diện sau khi thực hiện kiểm thử



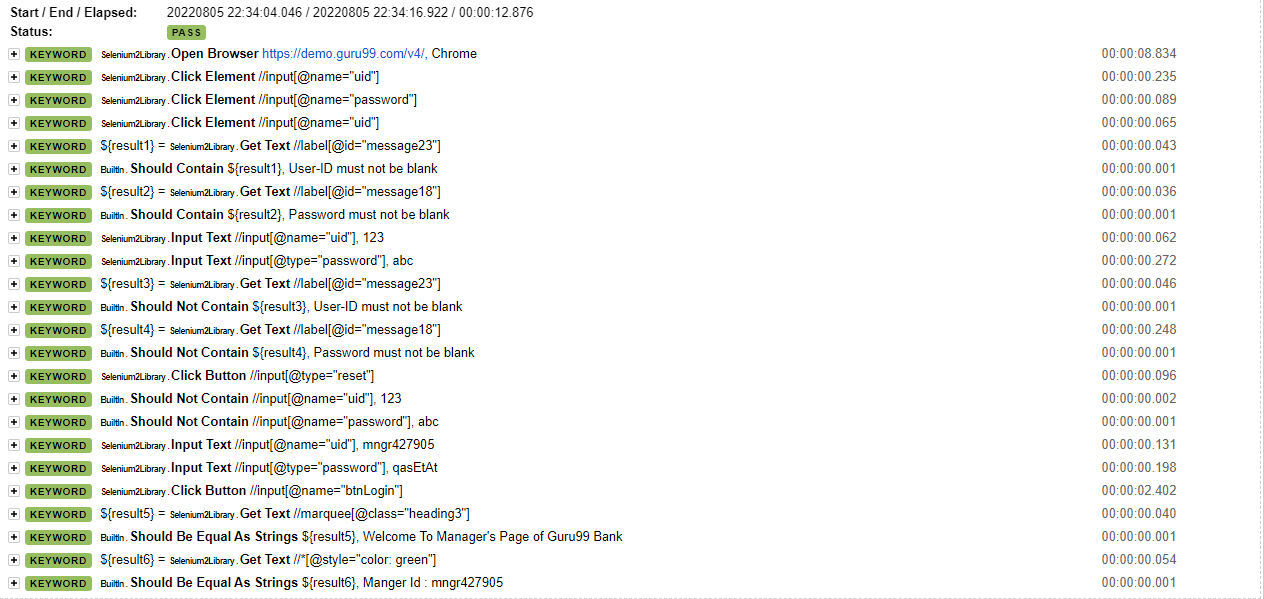
*Hình 10. Giao diện sau khi thực hiện kiểm thử*

* Thống kê kiểm thử (Test Statistics)



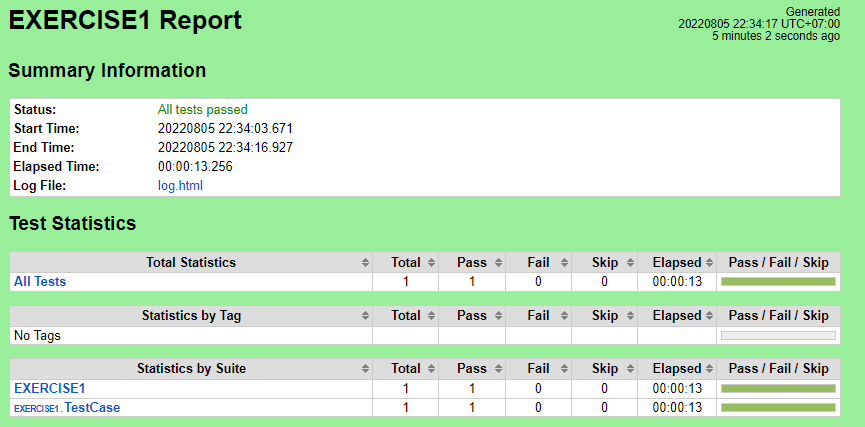
*Hình 11. Thống kê kiểm thử*

* Nhật ký thực thi kiểm thử (Test Execution Log)



*Hình 12. Nhật ký thực thi kiểm thử*

* Báo cáo



*Hình 13. Báo cáo kiểm thử*

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

* ***Kết quả đạt được:***
* Cải thiện khả năng đọc hiểu tài liệu tiếng anh.
* Nắm được kiến thức cơ bản về kiểm thử phần mềm.
* Áp dụng thành công kiến thức đã tìm hiểu và sử dụng các công cụ kiểm thử tự động để kiểm thử website Guru99.
* ***Kết luận:***
* Việc thực hiện kiểm thử được thực hiện theo một quy trình chặt chẽ và có sự liên kết giữa các bước thực hiện.
* Thiết kế test case phù hợp sẽ nâng cao được hiệu quả của kiểm thử tự động.
* ***Hạn chế:*** Thời gian thực hiện đề tài tương đối hạn chế nên khi thực hiện kiểm thử không thể tránh được những thiếu sót nhất định. Bên cạnh đó, báo cáo mới chỉ tập trung nghiên cứu sâu một tiện ích trong bộ công cụ kiểm thử tự động Selenium và chưa áp dụng được Robot framework trong Visual Studio vào triển khai code Python hỗ trợ cho kiểm thử tự động.
* ***Hướng phát triển:*** Từ những kiến thức có được sau quá trình học hỏi và thực tập tại doanh nghiệp, bản thân sẽ cố gắng tiếp tục rèn luyện để củng cố lại những kiến thức đã có. Đồng thời, trau dồi thêm những kĩ năng và tìm hiểu sâu hơn về Tester để tìm kiếm cơ hội nghề nghiệp trong tương lai.

# TÀI LIỆU THAM KHẢO

1. Rex Black, Dorothy Graham - Foundations of software testing\_ISTQB Certification.
2. Tutorialspoint - <https://www.tutorialspoint.com/index.htm>