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**

**NGHIÊN CỨU KIỂM THỬ TỰ ĐỘNG**

**CHO ỨNG DỤNG WEB GURU99**

**BẰNG CÔNG CỤ ROBOT FRAMEWORK**

Sinh viên thực hiện : Lê Nhật Phi

Lớp : 45K21.1

Đơn vị thực tập : TMA Solution 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: ........./ .……./ 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

Để hoàn thành bài báo cáo thực tập này, em xin bày tỏ lòng biết ơn sâu sắc đến cô Cao Thị Nhâm đã tận tình hướng dẫn cũng như nhận xét, góp ý trong suốt quá trình viết Báo cáo thực tập nghề nghiệp.

Em chân thành cảm ơn tất cả thầy cô giáo trong khoa Thông kê – Tin học, trường đại học Kinh tế - Đại học Đà Nẵng đã nhiệt tình giảng dạy và truyền đạt những kiến thức trong suốt 10 tuần thực tập. Những kiến thức này không chỉ là nền tảng cho quá trình thực tập mà còn giúp em đã phần nào biết được một số phương thức phục vụ cho ngành Tester mà em đang theo học và cũng chính là công việc sau này của bản thân.

Em cũng xin gửi lời cảm ơn đến ban lãnh đạo ở công ty TMA Solutions đã cho phép và tạo điều kiện thuận lợi cho em được thực tập tại công ty. Em xin chân thành cảm ơn các anh chị trong công ty, đặc biệt là anh Huỳnh Thanh Nhã đã trực tiếp hướng dẫn em trong suốt 10 tuần thực tập vừa qua. Em xin gửi lời chúc sức khỏe đến toàn thể các anh chị trong công ty. Chúc công ty ngày càng thành công và phát triển.

Em xin chân thành cảm ơn!

# LỜI CAM ĐOAN

Em xin cam đoan đây là bài báo cáo nghề nghiệp về đề tài “Nghiên cứu và kiểm thử tự động cho ứng dụng web Guru99 bằng công cụ Robot Framework” của em trong thời gian qua. Những số liệu và kết quả nghiên cứu là trung thực, hoàn toàn được thực hiện tại Phòng DG5 Tester của Công ty TMA solutions, không sao chép bất kỳ nguồn nào khác. Ngoài ra, trong bài báo cáo có sử dụng một số nguồn tài liệu tham khảo đã được trích dẫn nguồn và chú thích rõ ràng. Em xin hoàn toàn chịu trách nhiệm trước bộ môn, khoa và nhà trường về sự cam đoan này.

# MỤC LỤC

[LỜI CẢM ƠN i](#_Toc110935398)

[LỜI CAM ĐOAN ii](#_Toc110935399)

[MỤC LỤC iii](#_Toc110935400)

[DANH MỤC HÌNH ẢNH vi](#_Toc110935401)

[DANH MỤC BẢNG BIỂU vii](#_Toc110935402)

[DANH MỤC CÁC TỪ VIẾT TẮT viii](#_Toc110935403)

[LỜI MỞ ĐẦU 1](#_Toc110935404)

[CHƯƠNG 1. CƠ SỞ LÝ THUYẾT 3](#_Toc110935405)

[1.1. Tổng quan về kiểm thử phần mềm. 3](#_Toc110935406)

[1.1.1. Kiểm thử phần mềm là gì? 3](#_Toc110935407)

[1.1.2. Lợi ích của kiểm thử phần mềm. 3](#_Toc110935408)

[1.1.3. Vòng đời kiểm thử phần mềm (Software Testing Life Cycle). 3](#_Toc110935409)

[1.1.4. 7 nguyên tắc kiểm thử phần mềm. 4](#_Toc110935410)

[1.1.5. Error, Fault, Failure, Bug và Defect là gì? 4](#_Toc110935411)

[1.1.6. Verification và Validation là gì? 5](#_Toc110935412)

[1.1.7. Phân biệt giữa QA và QC. 6](#_Toc110935413)

[1.2. Vòng đời phát triển phần mềm (Software Development Life Cycle). 6](#_Toc110935414)

[1.2.1. SDLC là gì? 6](#_Toc110935415)

[1.2.2. Các mô hình phổ biến của SDLC. 7](#_Toc110935416)

[1.2.3. Các phương pháp của SDLC. 9](#_Toc110935417)

[1.3. Loại kiểm thử phần mềm. 10](#_Toc110935418)

[1.3.1. Kiểm thử thủ công. 10](#_Toc110935419)

[1.3.2. Kiểm thử tự động. 10](#_Toc110935420)

[1.3.3. Kiểm thử bảo mật. 10](#_Toc110935421)

[1.3.4. Kiểm thử API. 10](#_Toc110935422)

[1.4. Kỹ thuật kiểm thử phần mềm. 11](#_Toc110935423)

[1.4.1. Kiểm thử tĩnh. 11](#_Toc110935424)

[1.4.2. Kiểm thử động. 11](#_Toc110935425)

[1.4.3. Kiểm thử hộp trắng. 11](#_Toc110935426)

[1.4.4. Kiểm thử hộp đen. 11](#_Toc110935427)

[1.4.5. Kiểm thử hộp xám 11](#_Toc110935428)

[1.5. Cấp độ kiểm thử phần mềm. 12](#_Toc110935429)

[1.5.1. Kiểm thử đơn vị. 12](#_Toc110935430)

[1.5.2. Kiểm thử tích hợp. 12](#_Toc110935431)

[1.5.3. Kiểm thử hệ thống. 12](#_Toc110935432)

[1.5.4. Kiểm thử chấp nhận 12](#_Toc110935433)

[CHƯƠNG 2. TỔNG QUAN VÀ THIẾT LẬP MÔI TRƯỜNG KIỂM THỬ ROBOT FRAMEWORK 13](#_Toc110935434)

[2.1. Công cụ kiểm thử tự động Selenium 13](#_Toc110935435)

[2.2. Tổng quan về Robot FrameWork. 13](#_Toc110935436)

[2.3. Tổng quan về Xpath. 13](#_Toc110935437)

[2.4. Thiết lập môi trường thực hiện kiểm thử tự động. 14](#_Toc110935438)

[2.4.1. Cài đặt Python. 14](#_Toc110935439)

[2.4.2. Cài đặt Robot FrameWork bằng pip. 16](#_Toc110935440)

[2.4.3. Cài đặt thư viện Selenium2Library. 17](#_Toc110935441)

[2.4.4. Cài đặt Webdriver Manager. 18](#_Toc110935442)

[2.4.5. Cài đặt ChromeDriver 20](#_Toc110935443)

[CHƯƠNG 3. PHÂN TÍCH HỆ THỐNG ỨNG DỤNG WEB GURU99 21](#_Toc110935444)

[3.1. Tổng quan về hệ thống GURU99 Bank. 21](#_Toc110935445)

[3.1.1. Sơ đồ Use case tổng quát. 21](#_Toc110935446)

[3.1.2. Vai trò của từng tác nhân 21](#_Toc110935447)

[3.2. Phân tích Use Case “Đăng nhập” 22](#_Toc110935448)

[3.2.1. Sơ đồ use case chi tiết cho chức năng “Đăng nhập” 22](#_Toc110935449)

[3.2.2. Sơ đồ hoạt động chức năng đăng nhập. 23](#_Toc110935450)

[3.2.3. Đặc tả yêu cầu cho chức năng “Đăng nhập” 23](#_Toc110935451)

[3.3. Phân tích use case “thêm khách hàng” của Manager. 25](#_Toc110935452)

[3.3.1. Sơ đồ use case chi tiết cho chức năng “thêm khách hàng” 25](#_Toc110935453)

[3.3.2. Sơ đồ hoạt động chức năng “thêm khách hàng” 26](#_Toc110935454)

[3.3.3. Đặc tả yêu cầu cho chức năng “thêm khách hàng” 26](#_Toc110935455)

[CHƯƠNG 4. THIẾT KẾ TEST CASE VÀ TRIỂN KHAI KIỂM THỬ TRÊN MÔI TRƯỜNG ROBOT FRAMEWORK 30](#_Toc110935456)

[4.1. Thiết kế test case 30](#_Toc110935457)

[4.2. Thực hiện kiểm thử verify the Login Section. 32](#_Toc110935458)

[4.2.1. Thực hiện kiểm thử trên RIDE – Robot Frameword 32](#_Toc110935459)

[4.2.2. Kết quả đạt được 34](#_Toc110935460)

[4.3. Kiểm thử chức năng “Add New Customer” 36](#_Toc110935461)

[4.3.1. Thực hiện kiểm thử trên RIDE. 36](#_Toc110935462)

[4.3.2. Kết quả đạt được. 37](#_Toc110935463)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 40](#_Toc110935464)

[TÀI LIỆU THAM KHẢO 42](#_Toc110935465)

[PHỤ LỤC 43](#_Toc110935466)

# DANH MỤC HÌNH ẢNH

[Hình 1.1.3‑1 Hình ảnh vòng đời kiểm thử 4](#_Toc110935467)

[Hình 1.2.1‑1 Hình ảnh giai đoạn vòng đời phát triển phần mềm. 7](#_Toc110935468)

[Hình 1.5.4‑1 Cú pháp của XPath 13](#_Toc110935469)

[Hình 2.4.1‑1 Cài đặt Python 3.9.1 14](#_Toc110935470)

[Hình 2.4.1‑2 Thiết lập PYTHON\_HOME 14](#_Toc110935471)

[Hình 2.4.1‑3 Thiết lập Path 15](#_Toc110935472)

[Hình 2.4.1‑4 Cài đặt Python thành công 16](#_Toc110935473)

[Hình 2.4.2‑1 Cài đặt robot framework thành công 16](#_Toc110935474)

[Hình 2.4.2‑2 Cài đặt wxPython==4.1.1 thành công 16](#_Toc110935475)

[Hình 2.4.2‑3 Cài đặt RIDE thành công 17](#_Toc110935476)

[Hình 2.4.2‑4 Giao diện RIDE khi mở sau cài đặt thành công 17](#_Toc110935477)

[Hình 2.4.3‑1 Cài đặt Selenium2Library thành công 18](#_Toc110935478)

[Hình 2.4.4‑1 Cài đặt Webdriver – Manager thành công 18](#_Toc110935479)

[Hình 2.4.4‑2 Cài đặt webdrivermanager 19](#_Toc110935480)

[Hình 2.4.4‑3 cài đặt webdrivermanager chrome thành công 19](#_Toc110935481)

[Hình 2.4.4‑4 Thiết lập đường dẫn sau khi cài Webdrivermanager Chrome 19](#_Toc110935482)

[Hình 2.4.5‑1 Cài đặt Chromedriver 20](#_Toc110935483)

[Hình 3.1.1‑1 Sơ đồ usecase tổng quát 21](#_Toc110935484)

[Hình 3.2.1‑1Usecase chi tiết chức năng đăng nhập 22](#_Toc110935485)

[Hình 3.2.2‑1Sơ đồ hoạt động chức năng đăng nhập 23](#_Toc110935486)

[Hình 3.3.1‑1 Sơ đồ usecase chi tiết cho chức năng “thêm khách hàng” 25](#_Toc110935487)

[Hình 3.3.2‑1 Sơ đồ hoạt động chức năng thêm khách hàng 26](#_Toc110935488)

[Hình 4.2.1‑1 Import thư viện Selenium2Library 32](#_Toc110935489)

[Hình 4.2.1‑2 Nhập keyword và xpath cho kiểm thử chức năng đăng nhập(1) 33](#_Toc110935490)

[Hình 4.2.1‑3 Nhập keyword và xpath cho kiểm thử chức năng đăng nhập(2) 33](#_Toc110935491)

[Hình 4.2.1‑4 Code test chức năng đăng nhập 34](#_Toc110935492)

[Hình 4.2.2‑1 Giao diện sau khi thực hiện kiểm thử 34](#_Toc110935493)

[Hình 4.2.2‑2 Test Report chức năng đăng nhập 35](#_Toc110935494)

[Hình 4.2.2‑3 Test Log chức năng đăng nhập 35](#_Toc110935495)

[Hình 4.3.1‑1Import thư viện Selenium2Library cho chức năng thêm khách hàng 36](#_Toc110935496)

[Hình 4.3.1‑2 Nhập keyword và xpath cho kiểm thử chức năng thêm khách hàng(1) 36](#_Toc110935497)

[Hình 4.3.1‑3 Nhập keyword và xpath cho kiểm thử chức năng thêm khách hàng(2) 36](#_Toc110935498)

[Hình 4.3.1‑4 Code test chức năng thêm mới khách hàng 37](#_Toc110935499)

[Hình 4.3.2‑1Giao diện sau khi kiểm thử 37](#_Toc110935500)

[Hình 4.3.2‑2 Test Report chức năng thêm khách hàng 38](#_Toc110935501)

[Hình 4.3.2‑3 Test log chức năng thêm khách hàng(1) 39](#_Toc110935502)

[Hình 4.3.2‑4 Test log chức năng thêm khách hàng (2) 39](#_Toc110935503)

# DANH MỤC BẢNG BIỂU

[Bảng 1.1.6‑1 Bảng phân biệt giữa Verification và Validation 5](#_Toc110935504)

[Bảng 1.1.7‑1 Bảng phân biệt giữa QA và QC. 6](#_Toc110935505)

[Bảng 1.2.2‑1 Ưu – nhược điểm của mô hình thác nước. 8](#_Toc110935506)

[Bảng 1.2.2‑2 Ưu – nhược điểm của mô hình chữ V. 8](#_Toc110935507)

[Bảng 1.2.2‑3 Ưu - nhược điểm của mô hình xoắn ốc. 9](#_Toc110935508)

[Bảng 1.2.2‑4 Ưu – nhược điểm của mô hình Agile. 9](#_Toc110935509)

[Bảng 3.1.2‑1 Vai trò của từng tác nhân 22](#_Toc110935510)

[Bảng 3.2.3‑1Bảng mô tả chi tiết User-ID và password 24](#_Toc110935511)

[Bảng 3.2.3‑2Hệ thống kiểm tra dữ liệu 24](#_Toc110935512)

[Bảng 3.2.3‑3 Hệ thống xác nhận 25](#_Toc110935513)

[Bảng 3.3.3‑1Mô tả chi tiết các thông tin cần nhập 27](#_Toc110935514)

[Bảng 3.3.3‑2 Hệ thống kiểm tra dữ liệu nhập vào 29](#_Toc110935515)

[Bảng 3.3.3‑1Thiết kế test case cho 2 chức năng 32](#_Toc110935516)

# DANH MỤC CÁC TỪ VIẾT TẮT

**STLC**: Software Testing Life Cycle

**QA**: Quality Assurance

**QC**: Quality Control

**SDLC**: Software Development Life Cycle

**HP**: Hewlett Packard

**QTP**: Quick Test Professional

**ATDD**: Acceptance Test Driven Development

**XML**: eXtensible Markup Language

# LỜI MỞ ĐẦU

1. **Mục tiêu nghiên cứu của đề tài**

* Đề tài này nghiên cứu về kiểm thử lỗi của chức năng đăng nhập và thêm mới khách hàng, từ đó tìm ra lỗi và đưa ra giải pháp khắc phụ lỗi của ứng dụng web GURU99 Bank.

1. **Nhiệm vụ của đề tài**

* Nắm vững các kiến thức về Kiểm thử phần mềm, đặc biệt là Kiểm thử tự động
* Nghiên cứu và phân tích tổng quát ứng dụng web GURU99 Bank và chi tiết về hai chức năng: Đăng nhập và Thêm mới khách hàng.
* Thực hiện kiểm thử tự động cho ứng dụng web GURU99 Bank về hai chức năng: Đăng nhập và Thêm mới khách hàng.

1. **Phương pháp nghiên cứu**

* Tìm hiểu quy trình phát triển phần mềm.
* Tìm hiểu quy trình kiểm thử.
* Tìm hiểu về các phương pháp kiểm thử, cấp độ kiểm thử, phương pháp thiết kế kiểm thử.
* Tìm hiểu về cách thiết lập môi trường kiểm thử
* Phân tích ứng dụng web GURU99 Bank.
* Thực hiện kiểm thử tự động trên Robot Framwork.

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

* Đối tượng và phạm vi nghiên cứu của đề tài lần này là toàn bộ nhân viên quản lý của GURU99 Bank và khách hàng của GURU99 Bank

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**: Tổng quan và thiết lập môi trường kiểm thử Robot Framwork
* **Chương 3**: Phân tích hệ thống ứng dụng web GURU99 Bank
* **Chương 4**: Thiết kế testcase và triển khai kiểm thử trên môi trường Robot Framework
* Kết luận và hướng phát triển

# CƠ SỞ LÝ THUYẾT

## Tổng quan về kiểm thử phần mềm.

### Kiểm thử phần mềm là gì?

Kiểm thử phần mềm là phương pháp kiểm tra xem sản phẩm phần mềm đó trên thực thế có phù hợp với các yêu cầu đặt ra hay không, và đảm bảo rằng không có lỗi hay khiếm khuyết.

Mục đích của kiểm thử phần mềm là xác định các lỗi, khiếm khuyết hoặc các yêu cầu còn thiếu của phần mềm so với yêu cầu thực tế.

### Lợi ích của kiểm thử phần mềm.

* **Hiệu quả về chi phí:** việc tìm ra các lỗi/khiếm khuyết sẽ giúp công ty tiết kiệm, việc xác định lỗi trong giai đoạn đầu sẽ giúp quá trình sửa chữa tốn ít chi phí hơn.
* **Bảo mật:** kiểm thử giúp loại bỏ các rủi ro và vấn đề trong sản phẩm.
* **Chất lượng sản phẩm:** giúp công ty cung cấp các sản phẩm chất lượng cho khách hàng.
* **Sự hài lòng của khách hàng:** kiểm thử giúp sản phẩm đầy đủ và phù hợp với các yêu cầu của khách hàng.

### Vòng đời kiểm thử phần mềm (Software Testing Life Cycle).

Vòng đời kiểm thử phần mềm (STLC) là quy trình kiểm thử được thực hiện theo hệ thống và có kế hoạch rõ ràng. Trong quá trình kiểm thử, rất nhiều giai đoạn khác nhau được thực hiện một cách tuần tự. Mỗi giai đoạn đều có đầu vào và đầu ra khác nhau nhưng đều hướng tới mục tiêu cuối cùng là đảm bảo chất lượng sản phẩm.

Vòng đời kiểm thử phần mềm (STLC) gồm 6 giai đoạn:

* Phân tích yêu cầu.
* Lập kế hoạch kiểm thử.
* Thiết kế kịch bản kiểm thử.
* Thiết lập môi trường kiểm thử.
* Thực hiện kiểm thử.
* Đóng chu trình kiểm thử.



Hình 1.1.3‑1 Hình ảnh vòng đời kiểm thử

### 7 nguyên tắc kiểm thử phần mềm.

* Kiểm thử đưa ra lỗi.
* Kiểm thử cạn kiệt là không thể.
* Kiểm thử càng sớm càng tốt.
* Sự tập trung của lỗi.
* Nghịch lí thuốc trừ sâu.
* Kiểm thử phụ thuộc vào ngữ cảnh.
* Không có lỗi – Sai lầm.

### Error, Fault, Failure, Bug và Defect là gì?

**Error** là hành động của con người dẫn đến kết quả sai.

**Fault** là lỗi xảy ra khi làm sai các bước, quy trình hoặc chuẩn bị dữ liệu.

**Failure** là 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, hệ thống hoặc service 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.

### Verification và Validation là gì?

Verification là một quá trình đánh giá các sản phẩm làm việc trung gian của một vòng đời phát triển phần mềm để kiểm tra xem liệu rằng chúng ta có đi đúng hướng để tạo ra sản phẩm cuối cùng. Cũng có thể hiểu, Verification là một quá trình để đánh giá các sản phẩm trung gian của phần mềm để kiểm tra xem sản phẩm có đáp ứng các điều kiện đặt ra trong khi bắt đầu một giai đoạn.

Validation là quá trình đánh giá sản phẩm cuối cùng để kiểm tra xem phần mềm có đáp ứng được yêu cầu nghiệp vụ không.

|  |  |
| --- | --- |
| Verification | Validation |
| Đánh giá sản phẩm trung gian để kiểm tra xem nó có đáp ứng các yêu cầu cụ thể của từng giai đoạn không. | Đánh giá sản phẩm cuối cùng để kiểm tra xem nó có đáp ứng yêu cầu nghiệp vụ không. |
| Kiểm tra xem sản phẩm có được xây dựng đúng yêu cầu và đặc điểm kỹ thuật thiết kế không. | Xác định xem phần mềm có phù hợp với nhu cầu sử dụng và đáp ứng yêu cầu nghiệp vụ không. |
| Kiểm tra “Chúng tôi xây dựng sản phẩm đúng không?”. | Kiểm tra “Chúng tôi xây dựng đúng sản phẩm”. |
| Thực hiện mà không cần chạy phần mềm. | Thực hiện cùng với việc chạy phần mềm. |
| Bao gồm tất cả các kỹ thuật kiểm thử tĩnh. | Bao gồm tất cả các kỹ thuật kiểm thử động. |

Bảng 1.1.6‑1 Bảng phân biệt giữa Verification và Validation

### Phân biệt giữa QA và QC.

Quality Assurance (QA) là một tập hợp các hoạt động có kế hoạch và có hệ thống cần thiết để cung cấp sự tin tưởng đầy đủ rằng các sản phẩm và dịch vụ sẽ phù hợp với các yêu cầu cụ thể và đáp ứng nhu cầu của người sử dụng.

Quality Control (QC) là quá trình mà chất lượng sản phẩm được so sánh với các tiêu chuẩn áp dụng và hành động được thực hiện khi phát hiện thấy sự không phù hợp.

|  |  |
| --- | --- |
| QA | QC |
| Mục tiêu: ngăn ngừa khiếm khuyết. | Mục tiêu: xác định và cải thiện các khiếm khuyết. |
| Không liên quan đến thực hiện chương trình. | Luôn luôn liên quan đến việc thực hiện chương trình. |
| Là quy trình để tạo ra phần mềm. | Là quá trình để xác minh phần mềm. |
| Tất cả thành viên nhóm chịu trách nhiệm. | Đội kiểm thử chịu trách nhiệm. |

Bảng 1.1.7‑1 Bảng phân biệt giữa QA và QC.

## Vòng đời phát triển phần mềm (Software Development Life Cycle).

### SDLC là gì?

Vòng đời phát triển phần mềm (SDLC) là một quy trình phát triển phần mềm, được chia nhỏ thành nhiều giai đoạn từ ý tưởng đến triển khai. SDLC được sử dụng để giúp các nhóm phát triển tối ưu quy trình làm việc của họ và làm cho nó có thể mở rộng mà vẫn đảm bảo chất lượng.



Hình 1.2.1‑1 Hình ảnh giai đoạn vòng đời phát triển phần mềm.

SDLC gồm có 6 giai đoạn:

* Thu thập và phân tích yêu cầu.
* Thiết kế.
* Phát triển.
* Kiểm thử.
* Triển khai.
* Bảo trì.

### Các mô hình phổ biến của SDLC.

#### Mô hình thác nước (Water Model)

Là mô hình SDLC được chấp nhận rộng rãi. Theo cách tiếp cận này, toàn bộ quá trình phát triển phần mềm được chia thành các giai đoạn khác nhau, kết quả của một giai đoạn đóng vai trò là đầu vào cho giai đoạn tiếp theo.

|  |  |
| --- | --- |
| **Ưu điểm** | **Nhược điểm** |
| * Dễ hiểu, dễ quản lý. * Nhóm dự án hiểu rõ các yêu cầu của khách hàng * Có nhiều tài liệu cung cấp thông tin cho khách hàng. * Thích hợp cho các dự án đơn giản và nhỏ. | * Không cho phép thay đổi phạm vi dự dán. * Không cho phép thay đổi yêu cầu của dự án. * Chỉ nhìn thấy sản phẩm sau khi hoàn thành dự án. * Không dễ xử lý rủi ro bất ngờ. |

Bảng 1.2.2‑1 Ưu – nhược điểm của mô hình thác nước.

#### Mô hình chữ V (V-Model)

Là mô hình mở rộng của mô hình thác nước, trong mô hình này, các giai đoạn được lập kế hoạch song song. Vì vậy, có giai đoạn xác minh của SDLC ở phía bên kia và giai đoạn xác nhận ở phía bên kia. V-Model tham gia theo gia đoạn lập trình.

|  |  |
| --- | --- |
| **Ưu điểm** | **Nhược điểm** |
| * Dễ dàng quản lý. * Có tính tổ chức và hệ thống. * Phù hợp với các dự án có quy mô vừa và nhỏ. | * Không dễ thay đổi các yêu cầu thường xuyên. * Có sự không chắc chắc và có tính rủi ro. |

Bảng 1.2.2‑2 Ưu – nhược điểm của mô hình chữ V.

#### Mô hình xoắn ốc.

Là quy trình phát triển định hướng rủi ro cho các dự án phần mềm. Kết hợp thế mạnh của các mô hình khác và giải quyết của các mô hình trước còn tồn tại.

|  |  |
| --- | --- |
| **Ưu điểm** | **Nhược điểm** |
| * Các yêu cầu được thay đổi linh hoạt. * Hạn chế được rủi ro. * Phần mềm được sản xuất sớm trong vòng đời của nó. | * Không thích hợp cho dự án có quy mô nhỏ. * Đòi hỏi quản lý phải có chuyên môn cao. * Không thể xác định trước thời gian và chi phí của dự án. |

Bảng 1.2.2‑3 Ưu - nhược điểm của mô hình xoắn ốc.

#### Mô hình Agile

Là phương pháp thực hành nhằm thúc đẩy sự tương tác liên tục của quá trình phát triển và kiểm thử trong quá trình SDLC của bất kỳ dự án nào. Trong Agile, toàn bộ dự án được chia thành các bản xây dựng nhỏ tăng dần.

|  |  |
| --- | --- |
| **Ưu điểm** | **Nhược điểm** |
| * Phù hợp với những dự án thường xuyên thay đổi yêu cầu hoặc không có yêu cầu rõ ràng. * Khách hàng có thể xem trước từng phần dự án trong suốt quá trình phát triển. * Tỉ lệ thành công cao. | * Khó lên được kế hoạch dự án. * Cần sự hợp tác của các thành viên cao. * Chi phí cao so với các mô hình khác. * Ít tài liệu hướng dẫn cho nhân viên. |

Bảng 1.2.2‑4 Ưu – nhược điểm của mô hình Agile.

### Các phương pháp của SDLC.

#### Phương pháp Scrum

Scrum là bộ khung làm việc (framework) giúp các công ty, tổ chức chia nhỏ công việc thành những phần nhỏ hơn, để quản lý dễ dàng hơn và được hoàn thành bởi một nhóm liên chức năng trong khoảng thời gian quy định từ 2-4 tuần.

Lợi ích của phương pháp Scrum:

* Cải thiện chất lượng phần mềm.
* Rút ngắn thời gian phát hành phần mềm.
* Nâng cao tinh thần đồng đội.
* Tăng mức độ hài long khách hàng, …

#### Phương pháp Kanban.

Kanban là công cụ được sử dụng để giúp ác công ty tổ chức đạt hiệu quả cao trong công việc, là công cụ kiểm soát sản xuất, dung nhiều màu để chỉ định nguyên liệu và các công đoạn khác nhau. Kanban giới hạn số lượng công việc cho phép trong một điều kiện nhất định.

## Loại kiểm thử phần mềm.

### Kiểm thử thủ công.

Kiểm thử thủ công là một loại 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. Mục đích là xác định các lỗi, sự cố và khiếm khuyết trong ứng dụng phần mềm.

### Kiểm thử tự động.

Kiểm thử tự động là quá trình sử dụng các công cụ, script và phần mềm để thực hiện các trường hợp kiểm thử, bằng cách lặp lại những hành động được xác định trước để thực hiện các test case.

### Kiểm thử bảo mật.

Kiểm thử bảo mật là một loại kiểm thử phần mềm nhằm khám phá các lỗ hổng, mối đe dọa, rủi ro trong một ứng dụng phần mềm và ngăn chặn các cuộc tấn công độc hại từ những kẻ xâm nhập. Mục đích của kiểm thử bảo mật là xác định tất cả các lỗ hổng và điểm yếu có thể có của hệ thống phần mềm có thể dẫn đến việc mất thông tin, doanh thu, danh tiếng dưới tay của nhân viên hoặc người ngoài của tổ chức.

### Kiểm thử API.

Kiểm thử API là một loại kiểm thử nhằm kiểm tra trực tiếp server mà không cần các giao diện lập trình ứng dụng. Kiểm thử API cũng là một phần của kiểm thử tích hợp có vai trò xác định xem phần mềm có đáp ứng mong đợi về chức năng, độ tin cậy, hiệu suất và bảo mật hay không.

## Kỹ thuật kiểm thử phần mềm.

### Kiểm thử tĩnh.

Kiểm thử tĩnh (Static testing): Kiểm thử một thành phần hoặc hệ thống ở mức độ đặc tả hoặc thực nghiệm mà không chạy phần mềm, ví dụ: nhận xét hoặc phân tích code tĩnh.

### Kiểm thử động.

Kiểm thử động (Dynamic testing): Là kiểm thử có tiến hành chạy phần mềm, thực hiện các bước để kiểm thử xem các chức năng đã thực hiện có chạy đúng với kết quả mong đợi hay không.

### Kiểm thử hộp trắng.

Kiểm thử hộp trắng (White Box Testing) là kỹ thuật kiểm thử phần mềm trong đó cấu trúc bên trong, thiết kế và mã hóa của phần mềm được kiểm tra để xác minh luồng đầu vào – đầu ra và cải thiện thiết kế, khả năng sử dụng và bảo mật.

### Kiểm thử hộp đen.

Kiểm thử hộp đen (Black Box Testing) là 1 phương pháp kiểm thử phần mềm được thực hiện mà không biết được cấu tạo bên trong của phần mềm, là cách mà các tester kiểm tra xem hệ thống như một chiếc hộp đen, không có cách nào nhìn thấy bên trong của cái hộp.

### Kiểm thử hộp xám

Kiểm thử hộp xám (Gray box testing) là sự kết hợp giữa kiểm thử hộp đen và kiểm thử hộp trắng. Với kỹ thuật này, cấu trúc bên trong sản phẩm được biết một phần.

## Cấp độ kiểm thử phần mềm.

### Kiểm thử đơn vị.

Kiểm thử đơn vị (Unit testing), mức độ kiểm thử phần mềm này do các developer phụ trách, họ sẽ kiểm tra các module, các hàm, các phương thức,… mà họ viết ra nhằm gia tăng sự tin cậy cho các chức năng mà mình viết.

Kiểm thử đơn vị nằm trong phạm vi của kiểm thử hộp trắng.

### Kiểm thử tích hợp.

Kiểm thử tích hợp (Integration testing) là kiểm thử sự tương tác giữa các chức năng với nhau trong hệ thống và được thực hiện bởi các Tester.

Kiểm thử tích hợp có 2 hướng tiếp cận phổ biển: tích hợp từ dưới đi lên và tích hợp từ trên đi xuống.

### Kiểm thử hệ thống.

Kiểm thử hệ thống (System testing) 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 và được thực hiện bởi các Tester.

Kiểm thử hệ thống nằm trong phạm vi kiểm thử hộp đen.

Một số kiểm thử thường được thực hiện trong kiểm thử hệ thống: kiểm thử chức năng, kiểm thử hiệu năng, kiểm thử cơ sở dữ liệu, kiểm thử bảo mật, …

### Kiểm thử chấp nhận

Kiểm thử chấp nhận (Acceptance testing) là kiểm tra xem hệ thống có đá ứng đúng nhu cầu và mong đợi của khách hàng hay không và được thực hiện bởi người dung hoặc khách hàng.

Kiểm thử chấp nhận gồm 2 loại:

* Kiểm thử Alpha.
* Kiểm thử Beta.

# TỔNG QUAN VÀ THIẾT LẬP MÔI TRƯỜNG KIỂM THỬ ROBOT FRAMEWORK

## Công cụ kiểm thử tự động Selenium

Selenium là bộ kiểm thử tự động miễn phí (mã nguồn mở) dành cho các ứng dụng web trên các trình duyệt và nền tảng khác nhau. Nó khá giống với HP Quick Test Pro (QTP bây giờ là UFT) chỉ khác là Selenium thì tập trung vào việc tự động hóa các ứng dụng dựa trên nền tảng web.

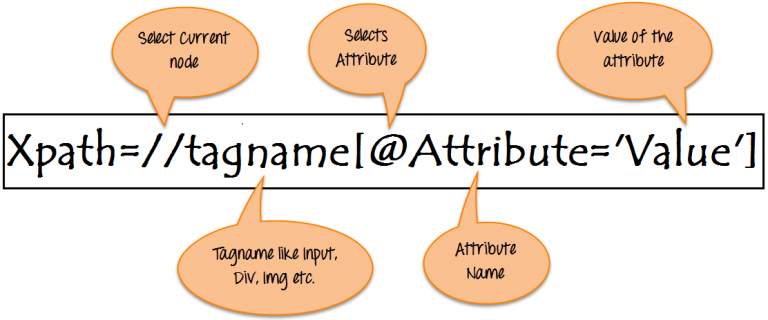
## Tổng quan về Robot FrameWork.

Là một khung kiểm thử tự động hóa hướng từ khóa và hướng dữ liệu dựa trên Python, có thể mở rộng để kiểm thử chấp nhận, phát triển theo hướng kiểm thử chấp nhận (ATDD). Nó cung cấp mọi thứ cần thiết để xây dựng và phát triển một kịch bản kiểm thử, gồm điều kiện đầu vào/kết thúc, báo cáo kết quả, …. Các tester không cần quan tâm đến các thuật toán lập trình cơ bản nhất.

## Tổng quan về Xpath.

Là một ngôn ngữ được thiết kế ra với mục đích giúp cho ứng dụng có thể “di chuyển” bên trong các tệp XML để truy xuất các giá trị, thuộc tính của các element.

XPath chứa đường dẫn của element nằm trên trang web, sau đây là cấu trúc tiêu chuẩn để tạo XPath:

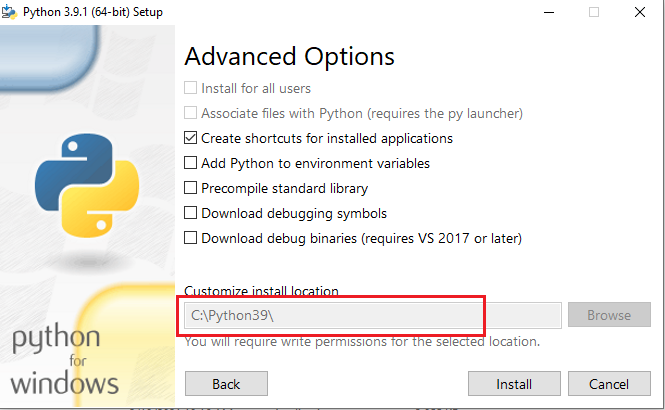


Hình 1.5.4‑1 Cú pháp của XPath

## Thiết lập môi trường thực hiện kiểm thử tự động.

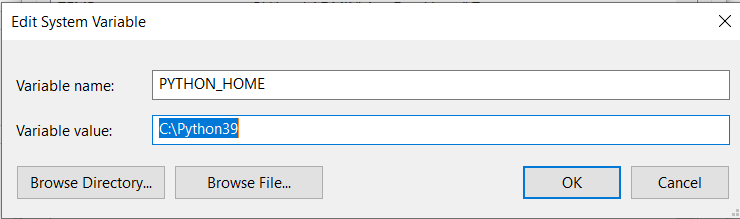
### Cài đặt Python.

Đây là bước bắt buộc cần có để sử dụng được Robot FrameWork. Cài đặt Python phiên bản 3.9.1 về máy tính, thư mục Python được lưu ở ổ đĩa C.



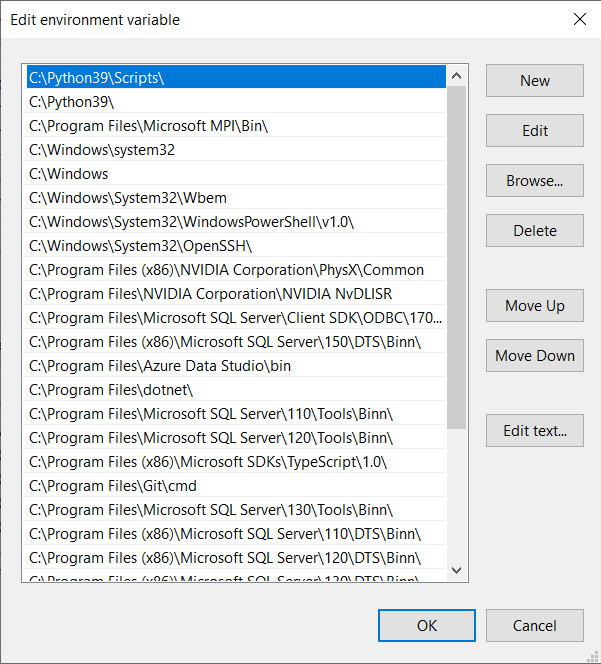
Hình 2.4.1‑1 Cài đặt Python 3.9.1

Thiết lập môi trường PYTHON\_HOME tại Advanced symtem setting.



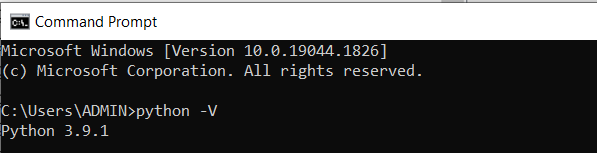
Hình 2.4.1‑2 Thiết lập PYTHON\_HOME

Thiết lập Path của Python tại Advanced symtem setting.



Hình 2.4.1‑3 Thiết lập Path

Kiểm tra Python đã cài đặt thành công.

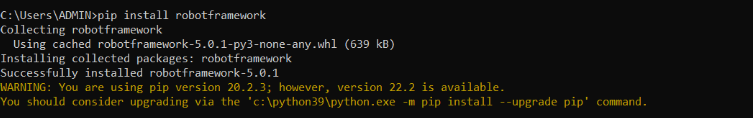


Hình 2.4.1‑4 Cài đặt Python thành công

### Cài đặt Robot FrameWork bằng pip.

Cài đặt Robot FrameWork sử dụng cmd trong C:\Python39\Script với lệnh:

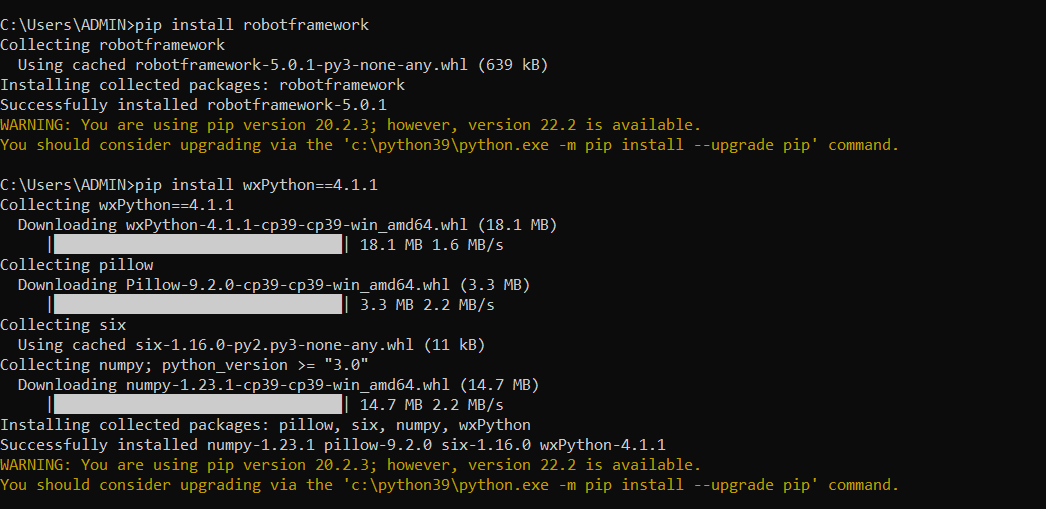
**pip install robot framework**

****

Hình 2.4.2‑1 Cài đặt robot framework thành công

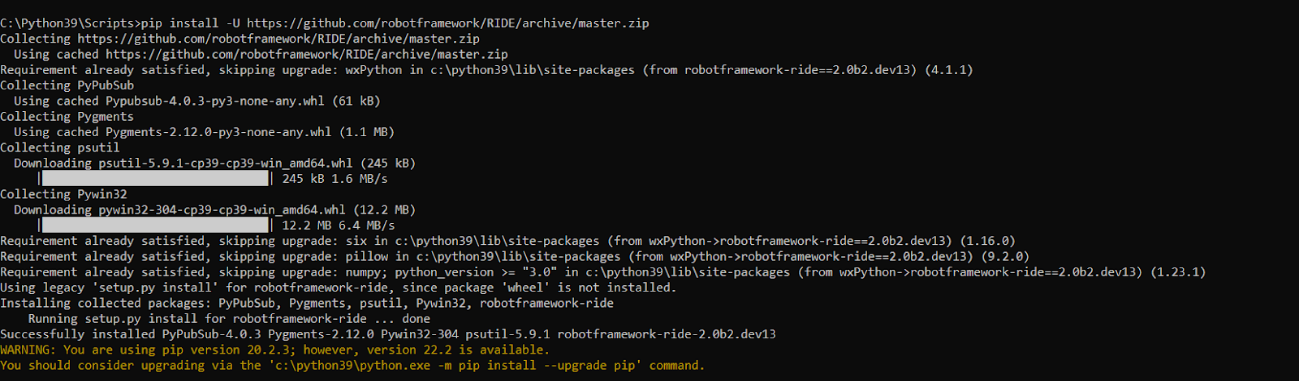
Cài đặt wxPython sử dụng cmd trong C:\Python39\Script với lệnh:

**pip install wxPython==4.1.1**



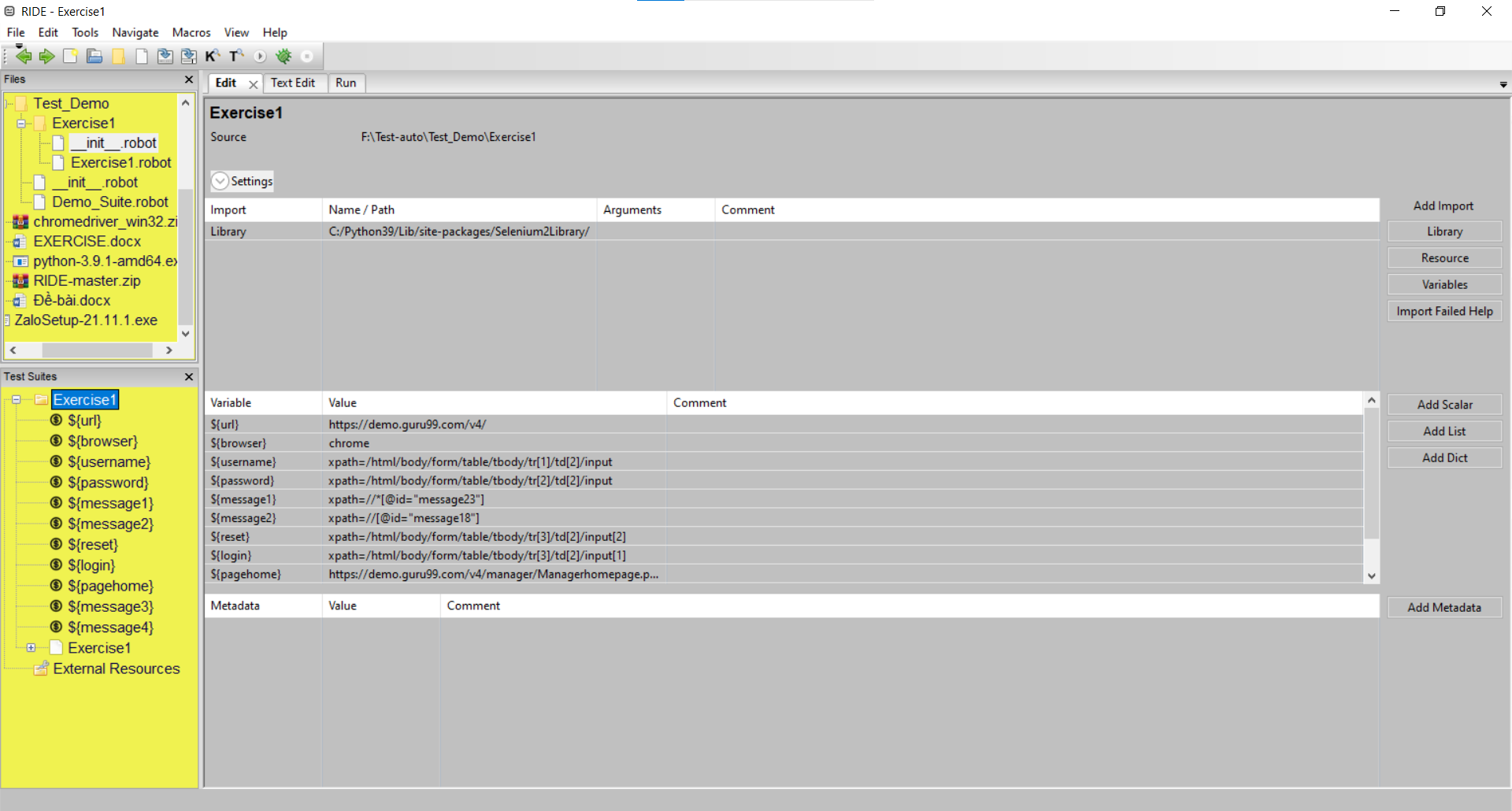
Hình 2.4.2‑2 Cài đặt wxPython==4.1.1 thành công

Cài đặt RIDE (tool editor) sử dụng cmd trong C:\Python39\Script với lệnh:



Hình 2.4.2‑3 Cài đặt RIDE thành công

Mở RIDE bằng file Ride.py trong C:\Python39\Script

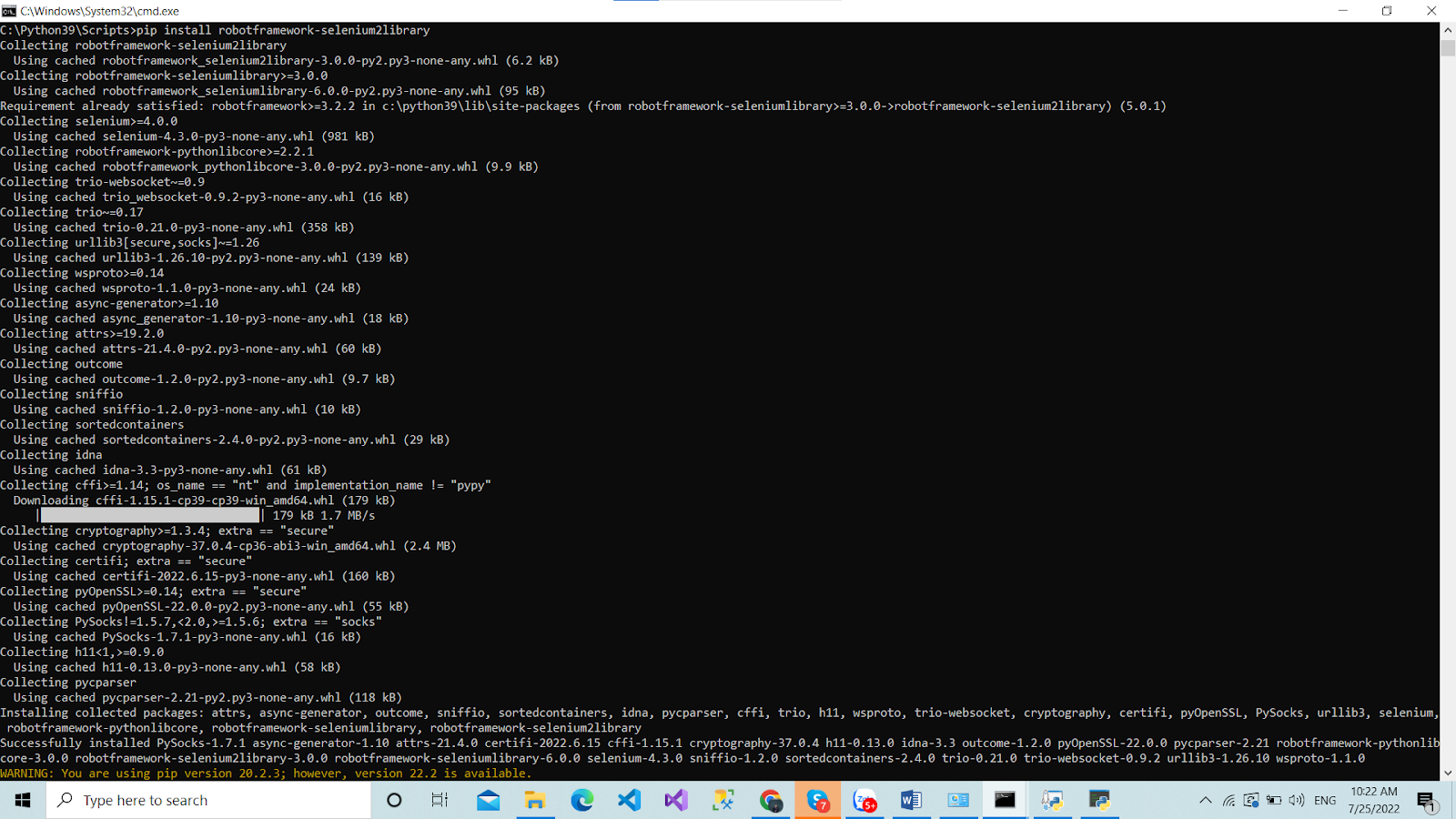


Hình 2.4.2‑4 Giao diện RIDE khi mở sau cài đặt thành công

### Cài đặt thư viện Selenium2Library.

Cài đặt Selenium2Library sử dụng cmd trong C:\Python39\Script với lệnh:

**pip install robotframework-selenium2library**

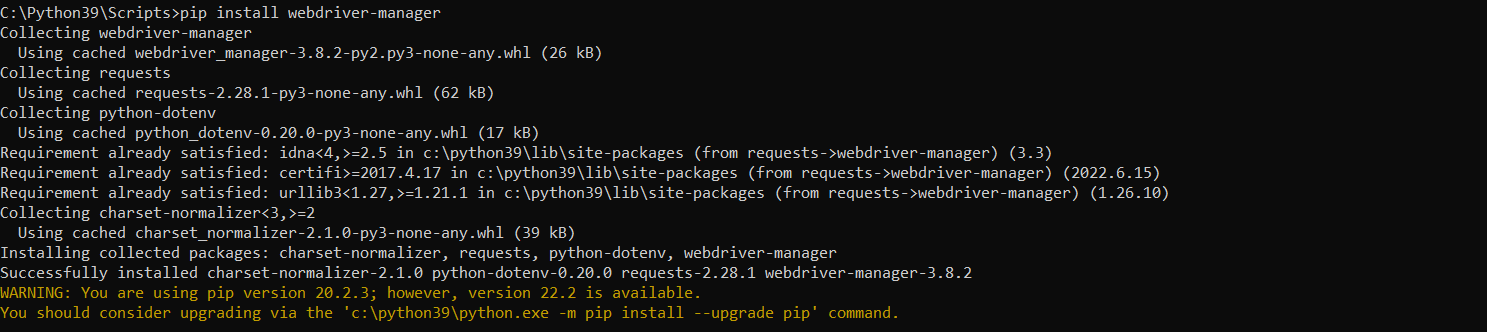


Hình 2.4.3‑1 Cài đặt Selenium2Library thành công

### Cài đặt Webdriver Manager.

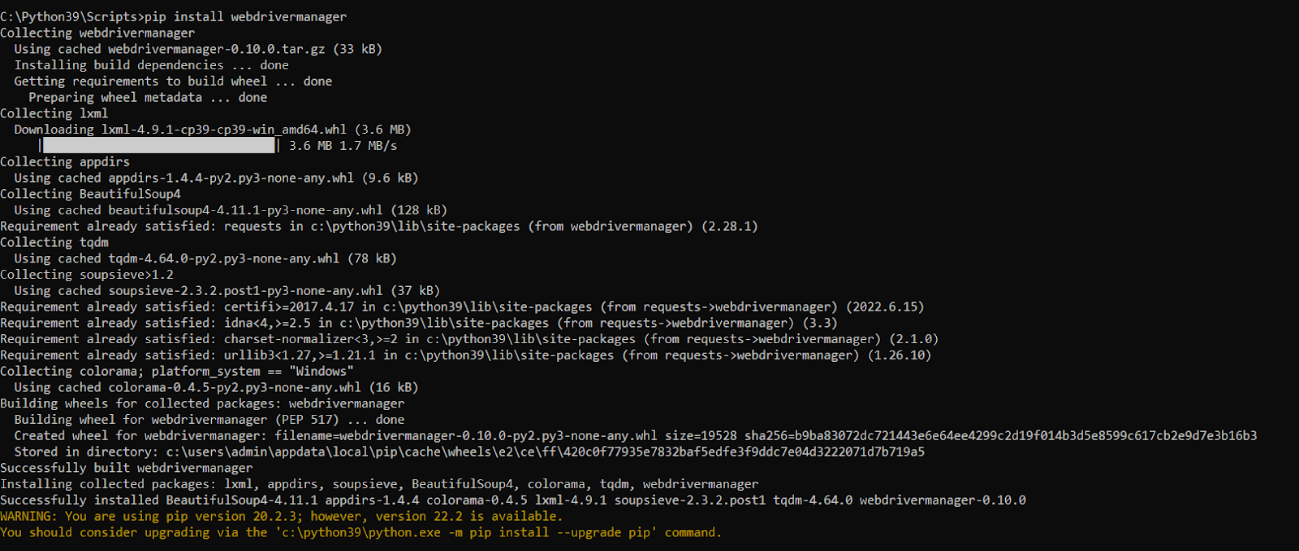
Cài đặt Webdriver Manager sử dụng cmd trong C:\Python39\Script với lần lượt từng lệnh:

**pip install webdriver-manager**



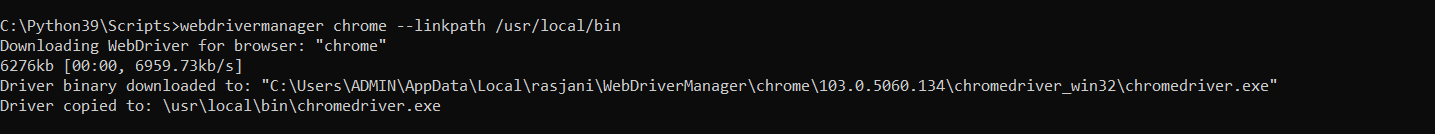
Hình 2.4.4‑1 Cài đặt Webdriver – Manager thành công

**pip install webdrivermanager**



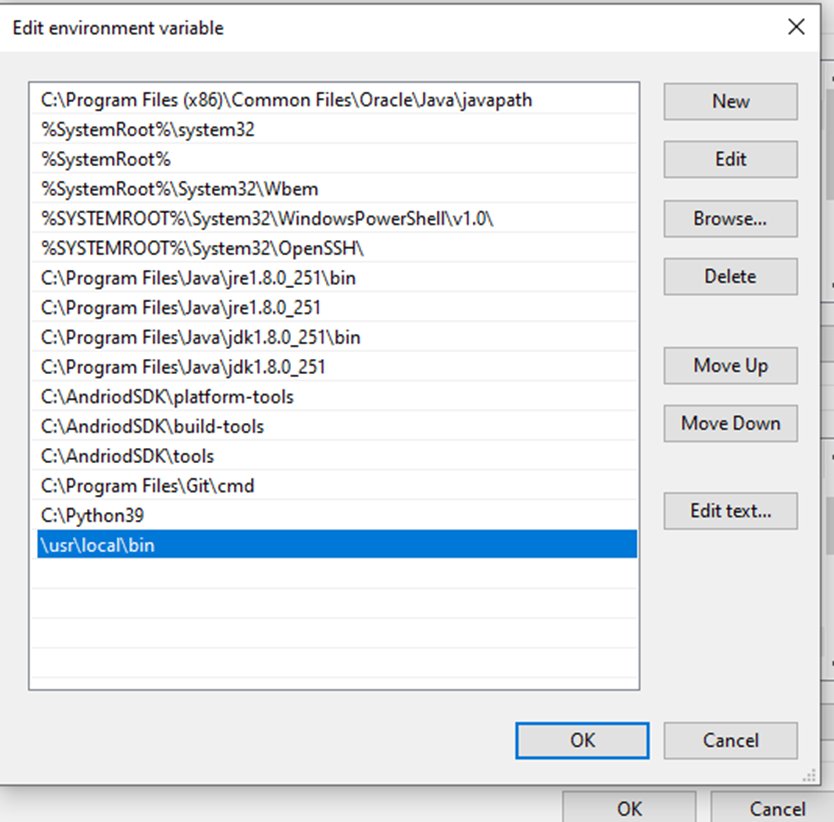
Hình 2.4.4‑2 Cài đặt webdrivermanager

**webdrivermanager chrome --linkpath /usr/local/bin**



Hình 2.4.4‑3 cài đặt webdrivermanager chrome thành công

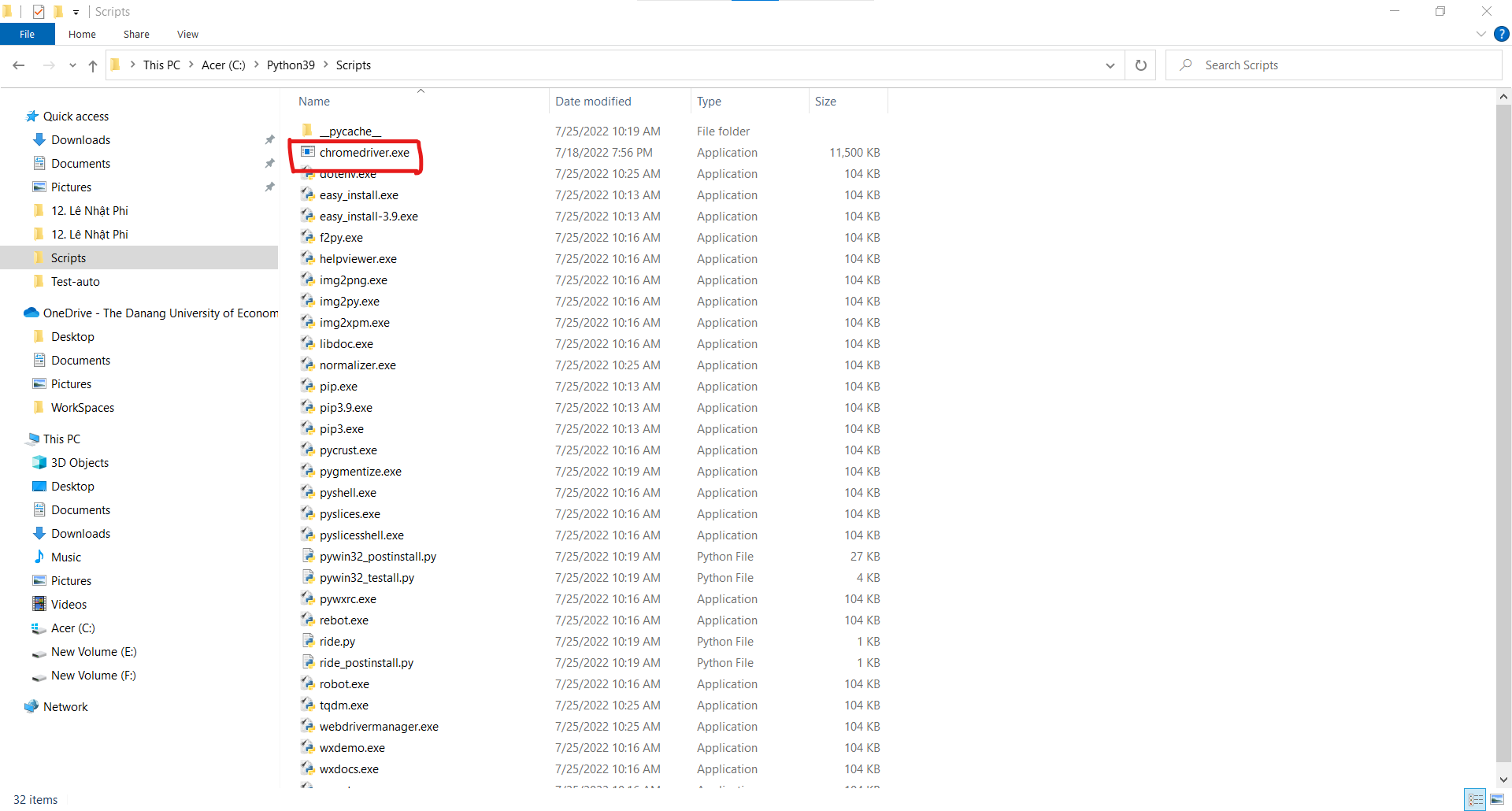
Thiết lập đường dẫn **\usr\local\bin**



Hình 2.4.4‑4 Thiết lập đường dẫn sau khi cài Webdrivermanager Chrome

### Cài đặt ChromeDriver

Tải ChromeDriver phù hợp với phiên bản Google Chrome ở máy tính và sao chép vào C:\Python39\Script



Hình 2.4.5‑1 Cài đặt Chromedriver

# PHÂN TÍCH HỆ THỐNG ỨNG DỤNG WEB GURU99

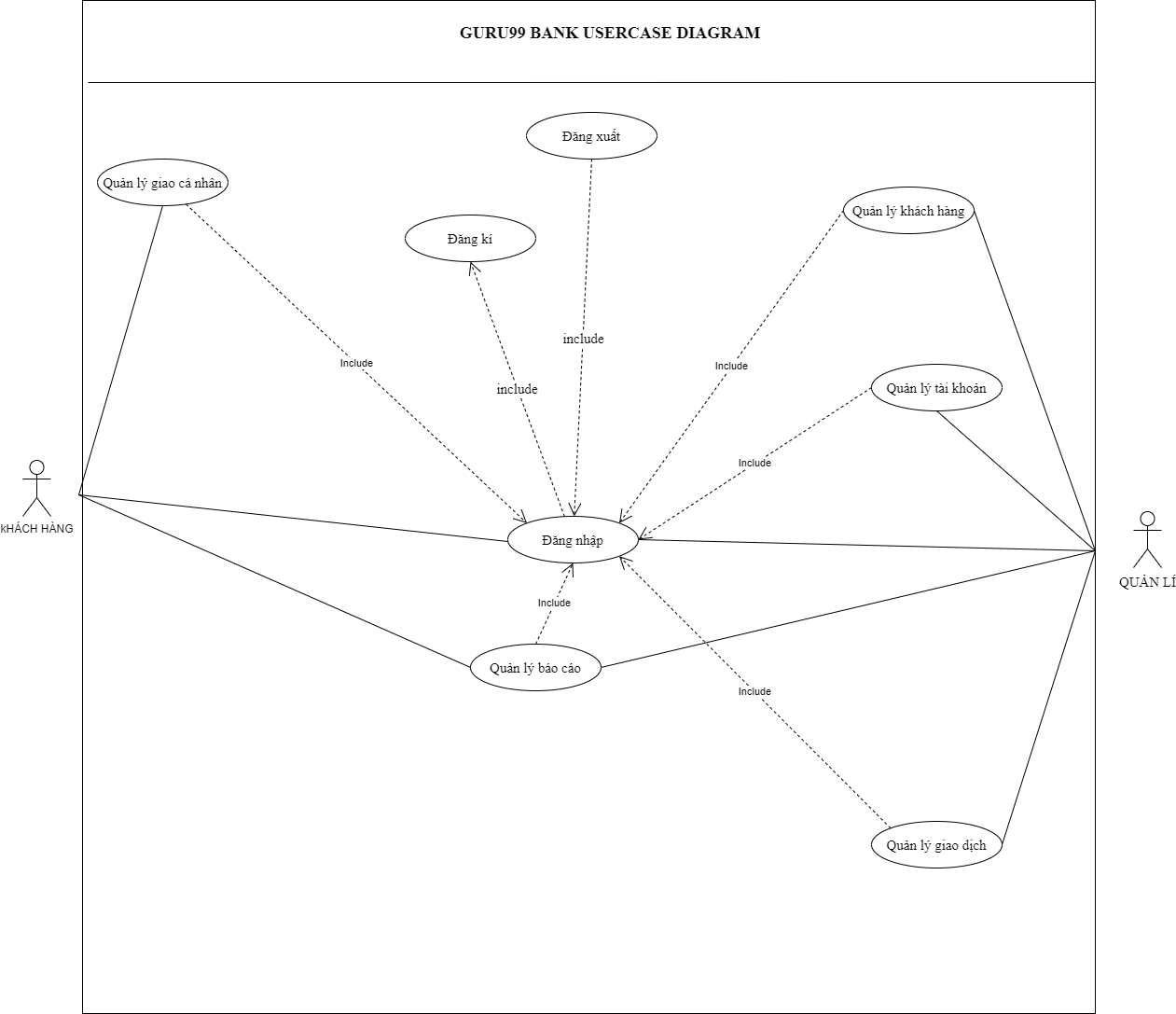
## Tổng quan về hệ thống GURU99 Bank.

Ứng dụng web GURU99 Bank được tạo ra nhằm làm môi trường giao dịch trực tuyến của ngân hàng GURU99, ứng dụng web này tập trung vào các nhân viên quản lý và khách hàng của ngân hàng. Điều này giúp ngân hàng và cả khách hàng giảm thiểu được các chi phí và đỡ tốn thời gian.

### Sơ đồ Use case tổng quát.

Các tác nhân của ứng dụng web Guru99 Bank bao gồm:

* Customer.
* Manager



Hình 3.1.1‑1 Sơ đồ usecase tổng quát

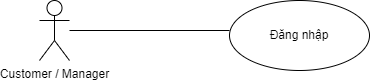
### Vai trò của từng tác nhân

|  |  |  |
| --- | --- | --- |
| No. | Tác nhân | Chức năng |
| 1 | Manager | Tác nhân thực hiện quản lý, giám sát và nắm bắt những yêu cầu từ người dùng:  Quản lý có thể thực hiện các chức năng của mình trong hệ thống bao gồm:   * Đăng nhập vào hệ thống. * Quản lý khách hàng (tạo, chỉnh sửa, xóa thông tin khách hàng). * Quản lý tài khoản (tạo, chỉnh sửa, xóa tài khoản của khách hàng) * Quản lý giao dịch của khách hàng. * Quản lý báo cáo giao dịch của khách hàng. |
| 2 | Customer | Là những người dùng tham gia vào việc sử dụng các dịch vụ của ngân hàng GURU99 Bank. Khách hàng được thực hiện các chức năng sau:   * Đăng nhập vào hệ thống. * Quản lý giao dịch của chính mình. * Quản lý báo cáo giao dịch của chính mình. |

Bảng 3.1.2‑1 Vai trò của từng tác nhân

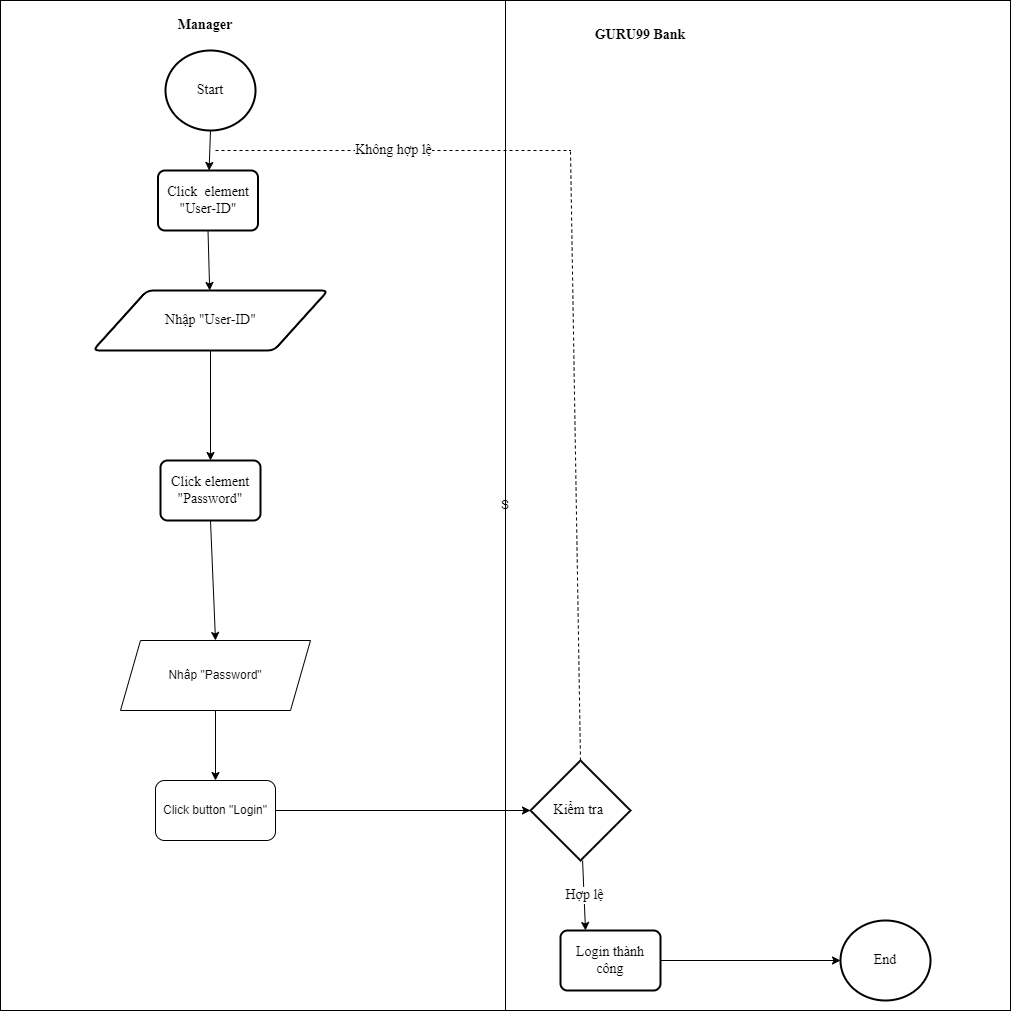
## Phân tích Use Case “Đăng nhập”

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



Hình 3.2.1‑1Usecase chi tiết chức năng đăng nhập

### Sơ đồ hoạt động chức năng đăng nhập.



Hình 3.2.2‑1Sơ đồ hoạt động chức năng đăng nhập

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

Tác nhân: Customer/Manager

Điều kiện tiên quyết: Đã có tài khoản (UserID và password), tài khoản đã Logout khỏi hệ thống

Mô tả khái quát: Use case được thực hiện khi khách hàng hoặc quản lý muốn đăng nhập vào hệ thống để thực hiện các chức năng khác.

Mô tả chi tiết:

* Bước 1: Customer/Manager muốn đăng nhập vào hệ thống đầu tiên phải nhập “User-ID” và “Password” ở 2 element “User-ID” và “Password”.

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Bắt buộc** | **Mục đích** |
| User-ID | x | Nhập User-ID của tài khoản cần đăng nhập |
| Password | x | Nhập Password của tài khoản cần đăng nhập |

Bảng 3.2.3‑1Bảng mô tả chi tiết User-ID và password

* Bước 2: Hệ thống kiểm tra dữ liệu nhập vào:

|  |  |
| --- | --- |
| **Tên trường** | **Yêu cầu kiểm tra** |
| User-ID | * Nếu User-ID để trống thì sẽ hiển thị thông báo lỗi “User-ID must not be blank” * Nếu User-ID nhập vào hơn 10 ký tự thì hệ thống chỉ hiển thị 10 ký tự nhập đầu tiên. |
| Password | * Nếu Password để trống thì sẽ hiển thị thông báo lỗi “Password must not be blank” |

Bảng 3.2.3‑2Hệ thống kiểm tra dữ liệu

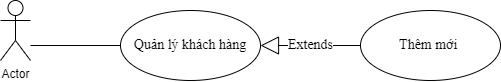
* Bước 3: Click vào Button “Login” để đăng nhập vào hệ thống.
* Bước 4: Hệ thống xác nhận:

|  |  |
| --- | --- |
| **Tên trường** | **Yêu cầu kiểm tra** |
| User-ID và Password | * Nếu User-ID và Password sai thì sẽ hiển thị thông báo “User or Password is not valid” * Nếu 1 trong 2 trường User-ID và Password sai thì sẽ hiển thị thông báo “User or Password is not valid” * Nếu User-ID và Password đúng thì login thành công và xuất hiện HomePage |

Bảng 3.2.3‑3 Hệ thống xác nhận

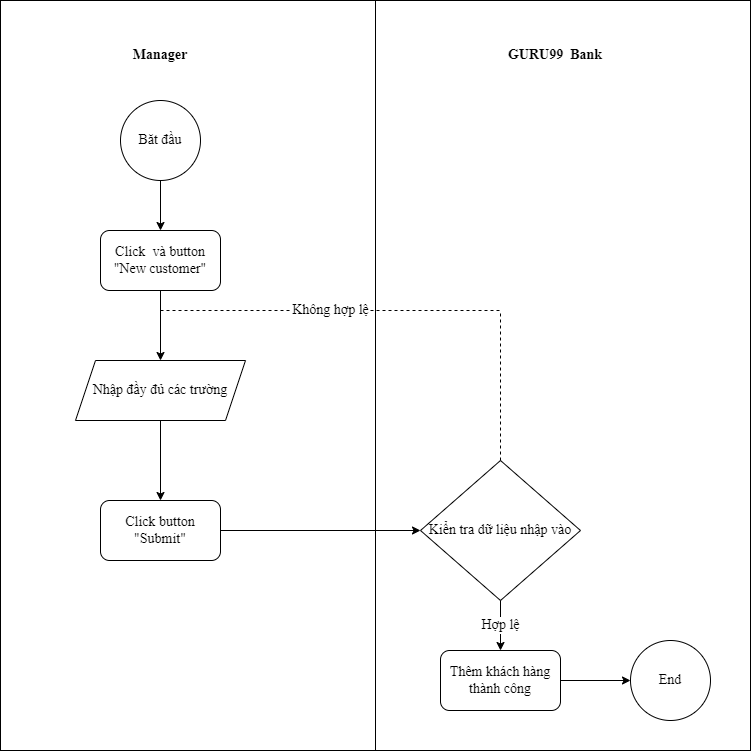
## Phân tích use case “thêm khách hàng” của Manager.

### Sơ đồ use case chi tiết cho chức năng “thêm khách hàng”



Hình 3.3.1‑1 Sơ đồ usecase chi tiết cho chức năng “thêm khách hàng”

### Sơ đồ hoạt động chức năng “thêm khách hàng”





Hình 3.3.2‑1 Sơ đồ hoạt động chức năng thêm khách hàng

### Đặc tả yêu cầu cho chức năng “thêm khách hàng”

Tác nhân: Manager

Điều kiện tiên quyết: Đăng nhập vào hệ thống thành công.

Mô tả khái quát: Use case thực hiện việc thêm mới một khách hàng và quản lý khách hàng đó.

Mô tả chi tiết:

* Bước 1: Manager muốn thêm mới khách hàng, thực hiện click vào button “New customer”.
* Bước 2: Hệ thống sẽ điều hướng đến Form “Add New Customer”, tại đây Manager sẽ cập nhập các thông tin của khách hàng bao gồm:

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Trường BB** | **Mục đích** |
| Customer Name | X | Đặt tên cho khách hàng |
| Gender | X | Giới tính của khách hàng. |
| Date of Birth | X | Ngày sinh của khách hàng. |
| Address | X | Địa chỉ cụ thể của khách hàng. |
| City | X | Thành phố đang ở của khách hàng. |
| State | X | Tiểu bang của khách hàng. |
| PIN | X | Mã Pin tài khoản của khách hàng |
| Mobile Number | X | Số điện thại liên lạc của khách hàng. |
| E-mail | X | E-mail của khách hàng |
| Password | X | Mật khẩu đăng nhập vào hệ thống của khách hàng. |

Bảng 3.3.3‑1Mô tả chi tiết các thông tin cần nhập

* Bước 3: Hệ thống kiểm tra các dữ liệu nhập vào:

|  |  |
| --- | --- |
| **Tên trường** | **Kiểm tra** |
| Customer Name | * Nếu để trống thì hệ thống xuất hiện thông báo lỗi “Customer name must not be blank” * Nhập không quá 25 ký tự |
| Gender | * Hợp lệ nếu chọn vào male hoặc female |
| Date of Birth | * Nếu để trống hoặc chỉ nhập thiếu một trong ba thuộc tính (dd, mm, yyyy) thì hệ thống xuất hiện thông báo lỗi “Date Field must not be blank” |
| Address | * Nếu để trống thì hệ thống xuất hiện thông báo lỗi “Date Field must not be blank” * Nhập không quá 50 ký tự. |
| City | * Nếu để trống thì hệ thống xuất hiện thông báo lỗi “City Field must not be blank” * Nếu nhập số vào thì hệ thống xuất hiện thông báo lỗi “Numbers are not allowed” * Nhập không quá 25 ký tự |
| State | * Nếu để trống thì hệ thống xuất hiện thông báo lỗi “State must not be blank” * Nếu nhập số vào thì hệ thống xuất hiện thông báo lỗi “Numbers are not allowed” * Nhập không quá 25 ký tự |
| PIN | * Nếu để trống thì hệ thống xuất hiện thông báo lỗi “PIN Code must not be blank” * Nếu nhập ký tự vào thì hệ thống xuất hiện thông báo lỗi “Characters are not allowed” |
| Mobile Number | * Nếu để trống thì hệ thống xuất hiện thông báo lỗi “Mobile no must not be blank” * Nếu nhập ký tự vào thì hệ thống xuất hiện thông báo lỗi “Characters are not allowed” * Nhập không đủ 6 số thì hệ thống xuất hiện thông báo lỗi “PIN Code must have 6 Digits” |
| E-mail | * Nếu để trống thì hệ thống xuất hiện thông báo lỗi “Email-ID must not be blank” * Nếu nhập e-mail sai vào thì hệ thống xuất hiện thông báo lỗi “Email-ID is not valid” |
| Password | * Nếu để trống thì hệ thống xuất hiện thông báo lỗi “Password must not be blank” |

Bảng 3.3.3‑2 Hệ thống kiểm tra dữ liệu nhập vào

* Bước 4: Manager click button “Submit” để thêm khách hàng.

# THIẾT KẾ TEST CASE VÀ TRIỂN KHAI KIỂM THỬ TRÊN MÔI TRƯỜNG ROBOT FRAMEWORK

## Thiết kế test case

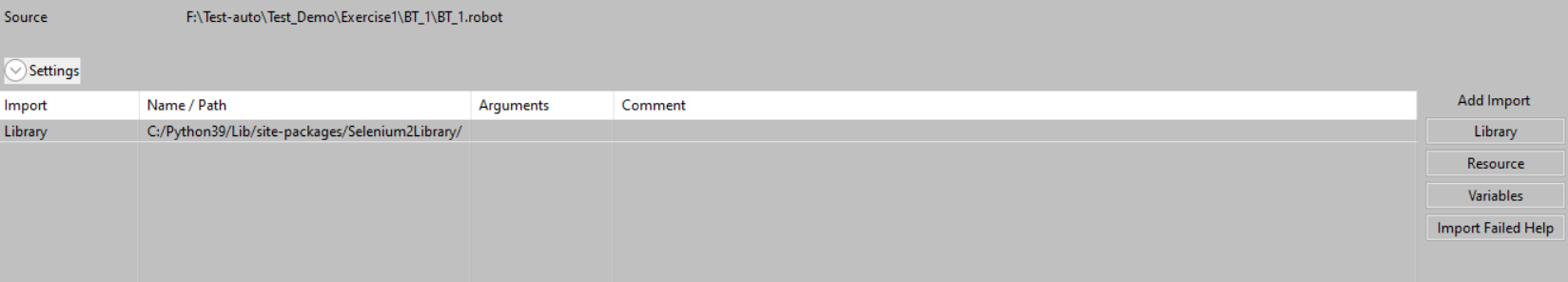
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Feature** | **Test case Description** | **Test data** | **Expected Result** | **Result** |
| **1** | **Verify the Login Section** | **TC1:** Go to https://demo.guru99.com/v4/ |  | Browser Chorme  Guru99 Bank Home | **Pass** |
| **TC2:** Verify text 'User-ID must not be blank' and 'Password must not be blank' if userID or password field is empty |  | Guru99 Bank Home  Message1: “User-ID must not be blank'”  Message2: “Password must not be blank'” | **Pass** |
| **TC3:** Verify text 'User-ID must not be blank' and 'Password must not be blank' are disappeared | User-ID: abcd  Password: 1234 | Guru99 Bank Home  Input User-ID  Input Password | **Pass** |
| **TC4:** Verify userID and password is cleared |  | Guru99 Bank Home  User-ID empty  Password empty | **Pass** |
| **TC5:**Verify login to home page successful | User-ID: mngr427649  Password:  rAsYrep | Managerhomepage | **Pass** |
| **TC6:** Verify dynamic userID in home page |  | Managerhomepage  Dynamic userID | **Pass** |
| 2 | Verify add new customer | **TC7:** Go to https://demo.guru99.com/v4/ |  | Browser Chorme  Guru99 Bank Home | **Pass** |
| **TC8:** Login Home | User-ID  Password | Managerhomepage | **Pass** |
| **TC9:** Click new customer |  | Managerhomepage  Table “Add New Custome” | **Pass** |
| **TC10:** Fill all fields | Customer name | Show customer name | **Pass** |
| Gender | Show gender | **Pass** |
| Date of Birth | Show date of birth | **Fail** |
| Address | Show address | **Pass** |
| City | Show city | **Pass** |
| State | Show state | **Pass** |
| Mobile Number | Show mobile number | **Pass** |
| E-mail | Show e-mail | **Pass** |
| Password | Show hidden password | **Pass** |
| **TC11:** Verify add new customer's information correct with value input above |  | Managerhomepage  Message “Customer Registered Successfully!!!” | **Fail** |

Bảng 3.3.3‑1Thiết kế test case cho 2 chức năng

## Thực hiện kiểm thử verify the Login Section.

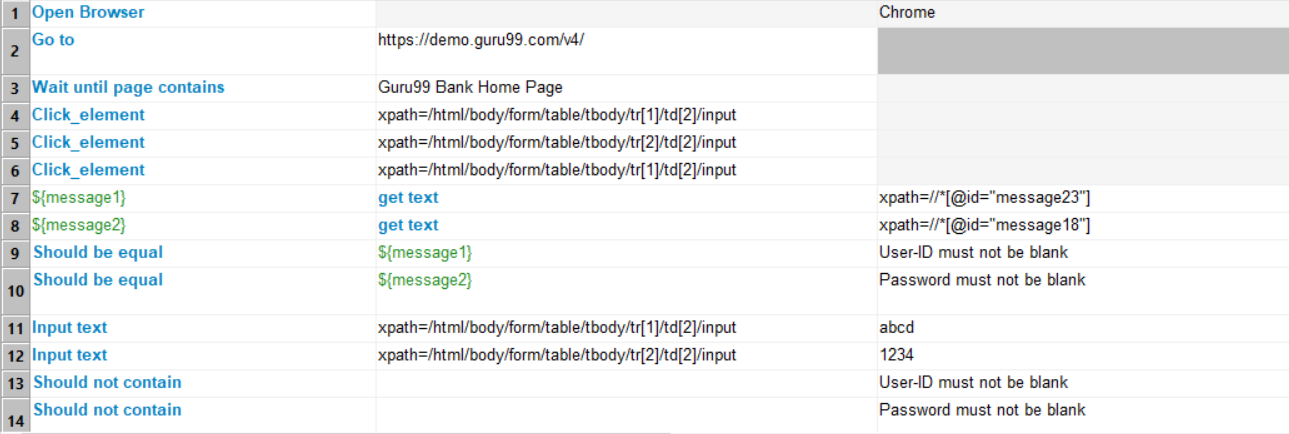
### Thực hiện kiểm thử trên RIDE – Robot Frameword

* Kết nối thư viện Selenium2Library vào file .robot thực hiện kiểm thử

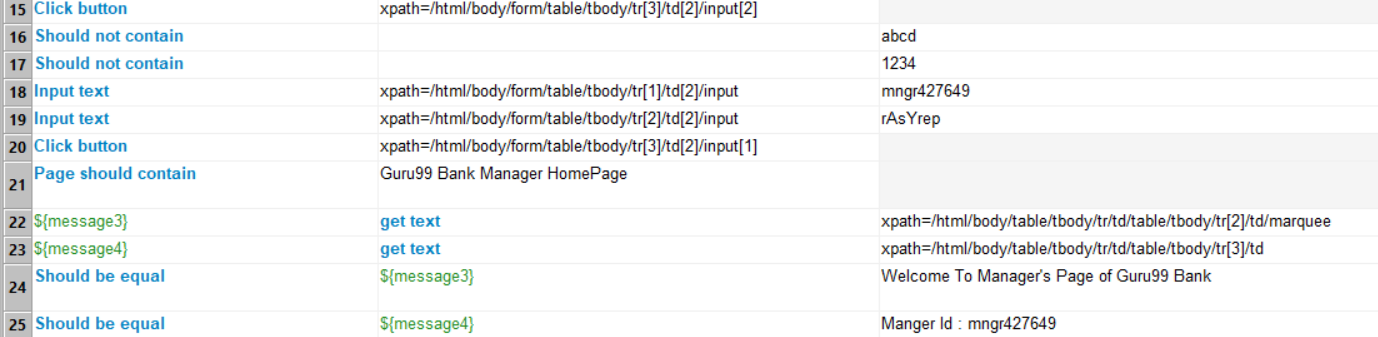


Hình 4.2.1‑1 Import thư viện Selenium2Library

* Nhập các keyword của thư viện Selenium2Library cho các steps dựa trên testcase phân tích ở trên và lấy Xpath cho phù hợp.



Hình 4.2.1‑2 Nhập keyword và xpath cho kiểm thử chức năng đăng nhập(1)



Hình 4.2.1‑3 Nhập keyword và xpath cho kiểm thử chức năng đăng nhập(2)

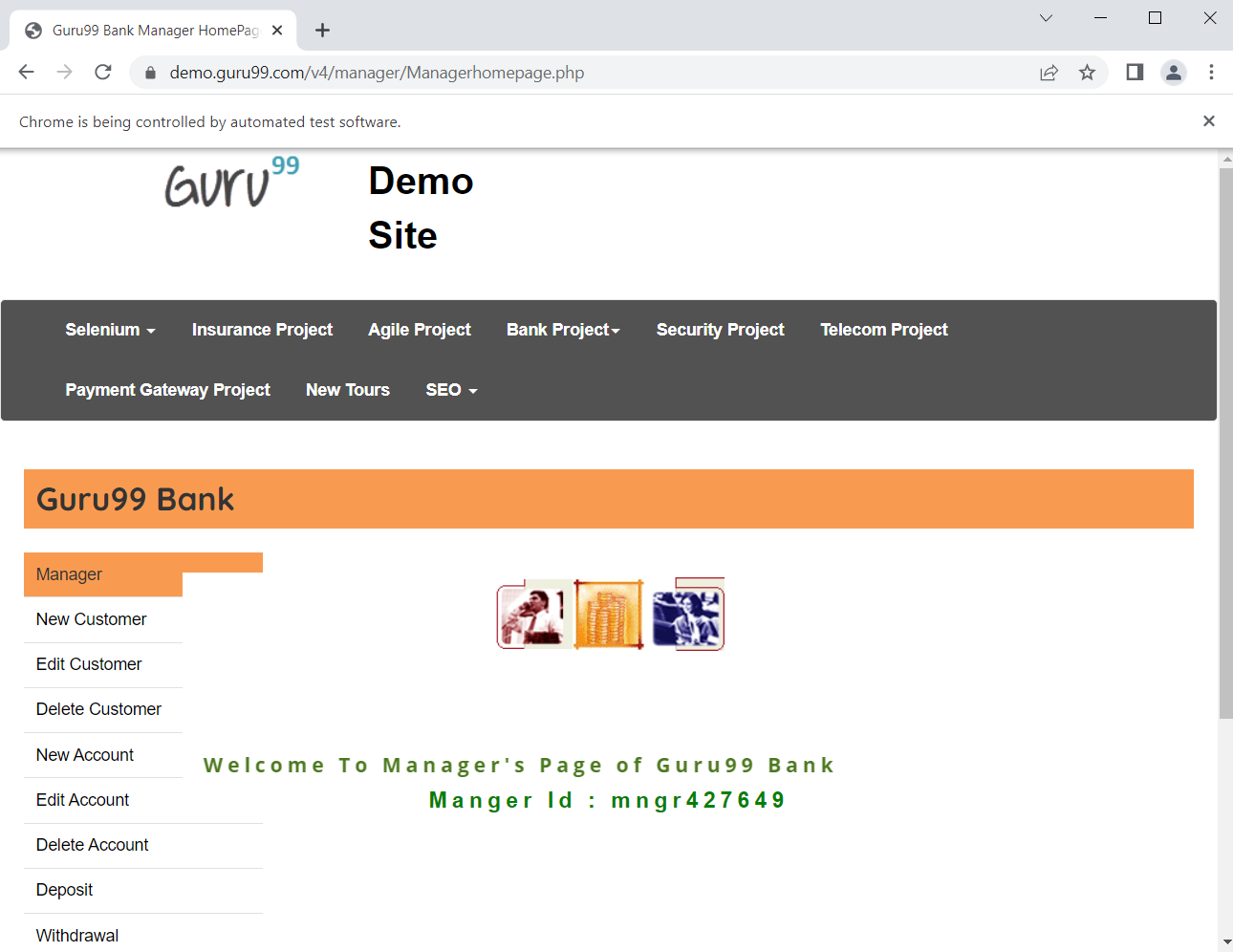
* Code test ***verify the Login Section***



Hình 4.2.1‑4 Code test chức năng đăng nhập

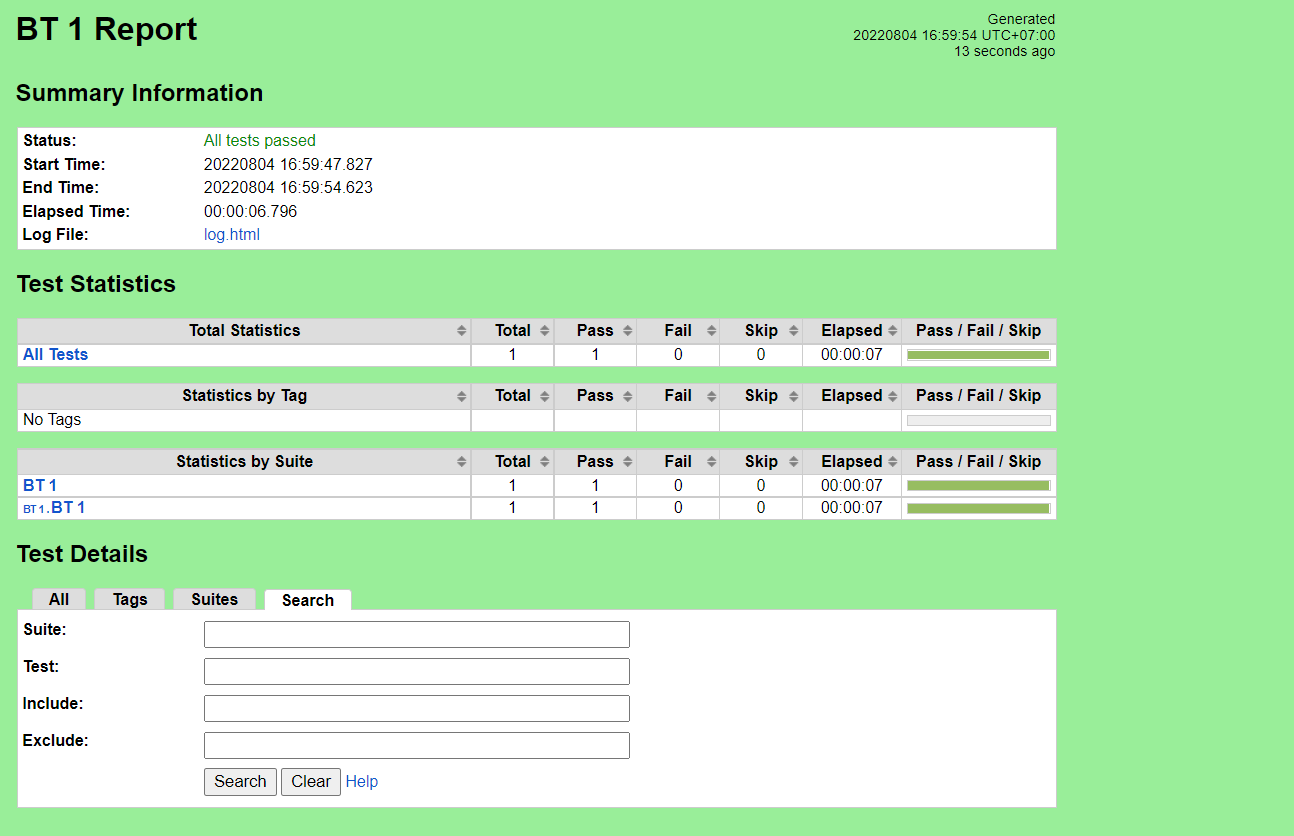
### Kết quả đạt được

* **Giao diện sau khi thực hiện kiểm thử**



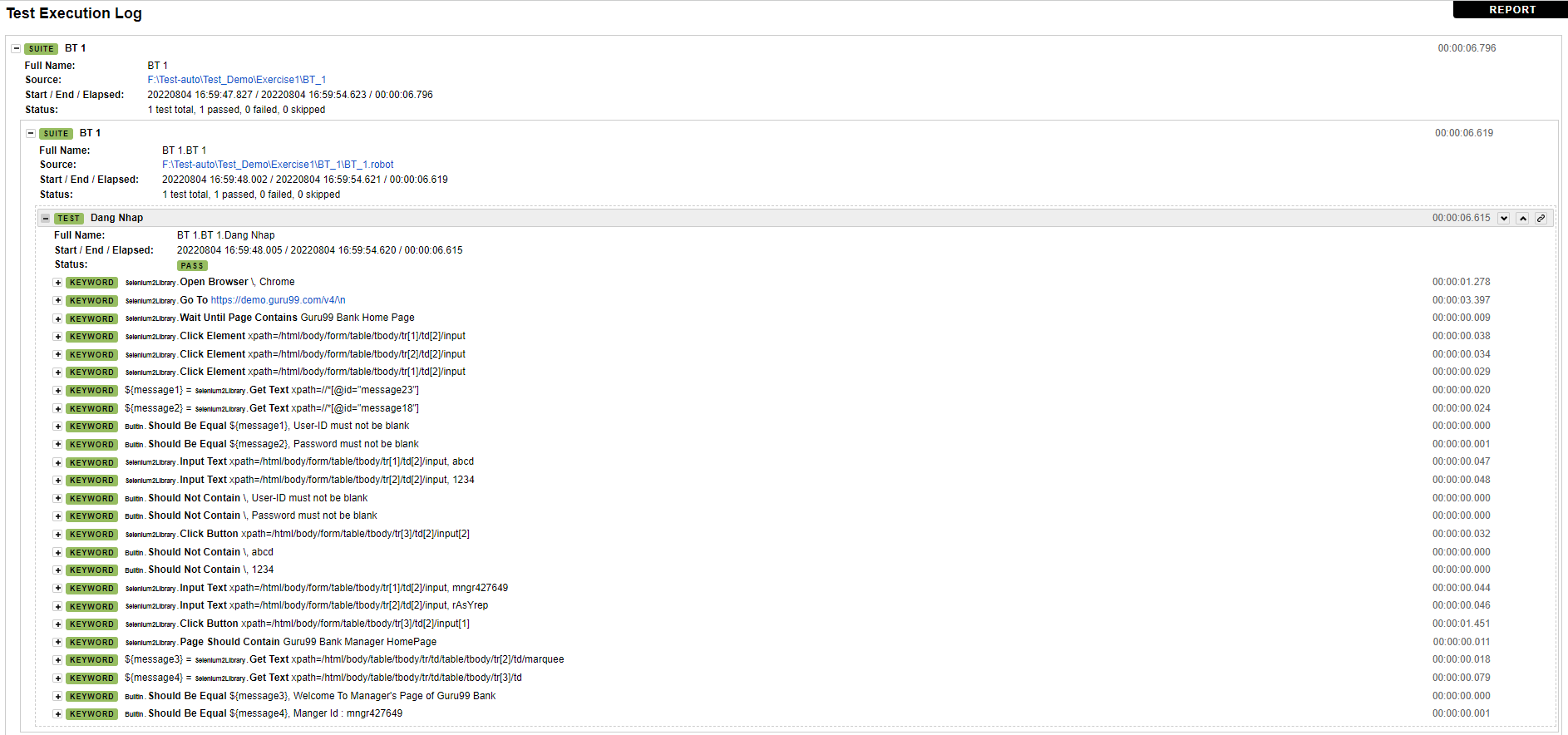
Hình 4.2.2‑1 Giao diện sau khi thực hiện kiểm thử

* **Test Report**



Hình 4.2.2‑2 Test Report chức năng đăng nhập

* **Test Log**

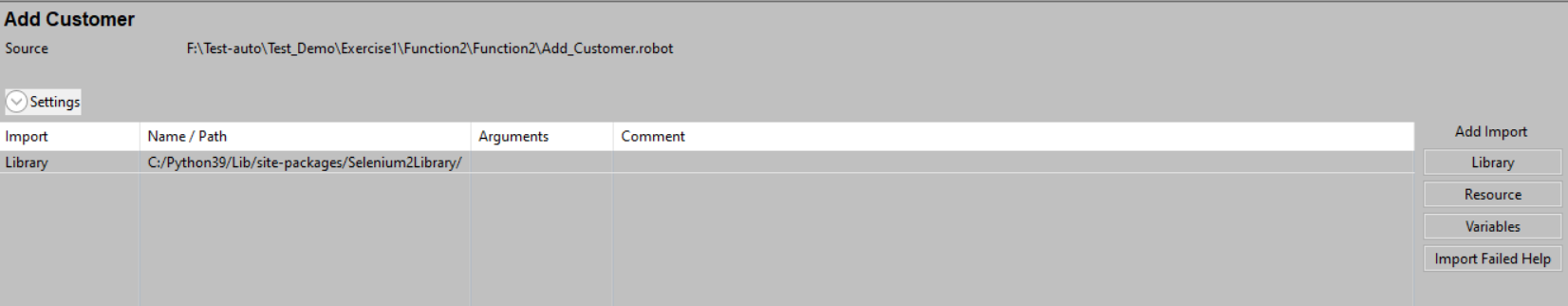


Hình 4.2.2‑3 Test Log chức năng đăng nhập

## Kiểm thử chức năng “Add New Customer”

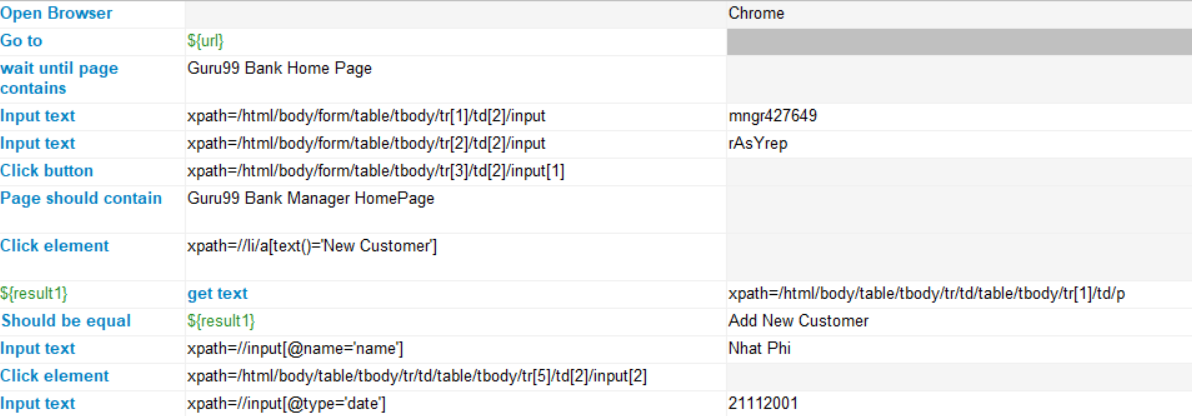
### Thực hiện kiểm thử trên RIDE.

* Kết nối thư viện Selenium2Library vào file .robot thực hiện kiểm thử

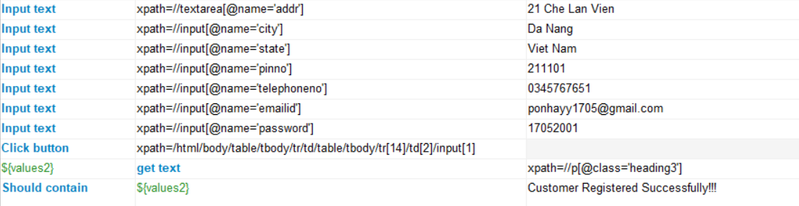


Hình 4.3.1‑1Import thư viện Selenium2Library cho chức năng thêm khách hàng

* Nhập các keyword của thư viện Selenium2Library cho các steps dựa trên testcase phân tích ở trên và lấy Xpath cho phù hợp

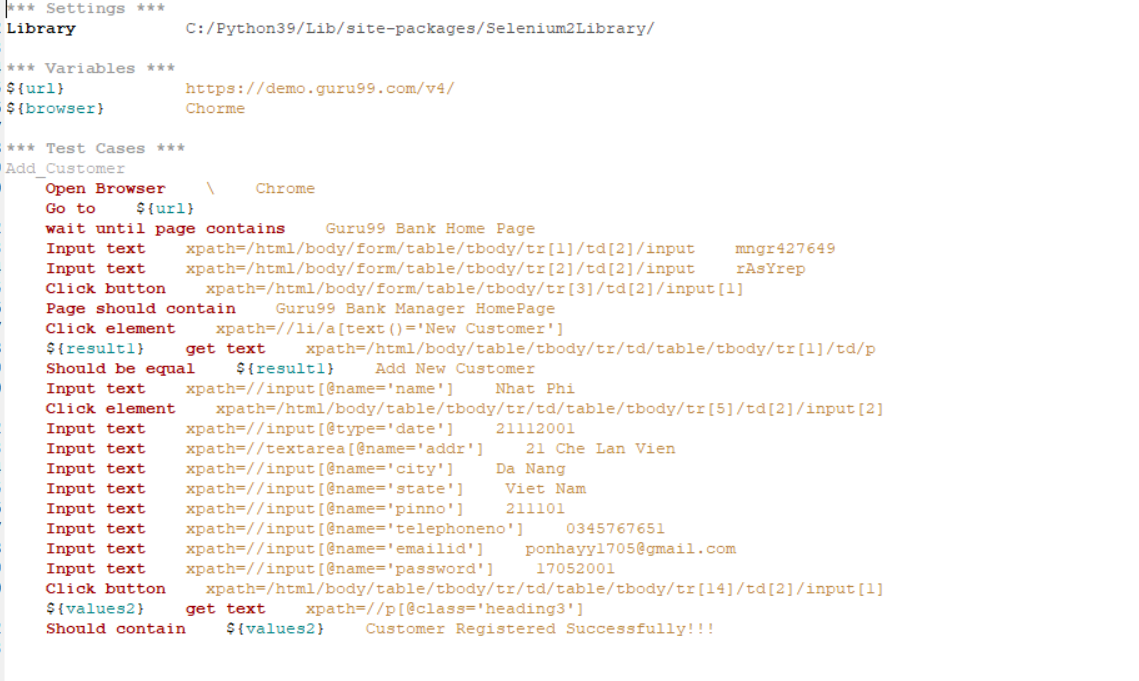


Hình 4.3.1‑2 Nhập keyword và xpath cho kiểm thử chức năng thêm khách hàng(1)



Hình 4.3.1‑3 Nhập keyword và xpath cho kiểm thử chức năng thêm khách hàng(2)

* Code test “Add New Customer”



Hình 4.3.1‑4 Code test chức năng thêm mới khách hàng

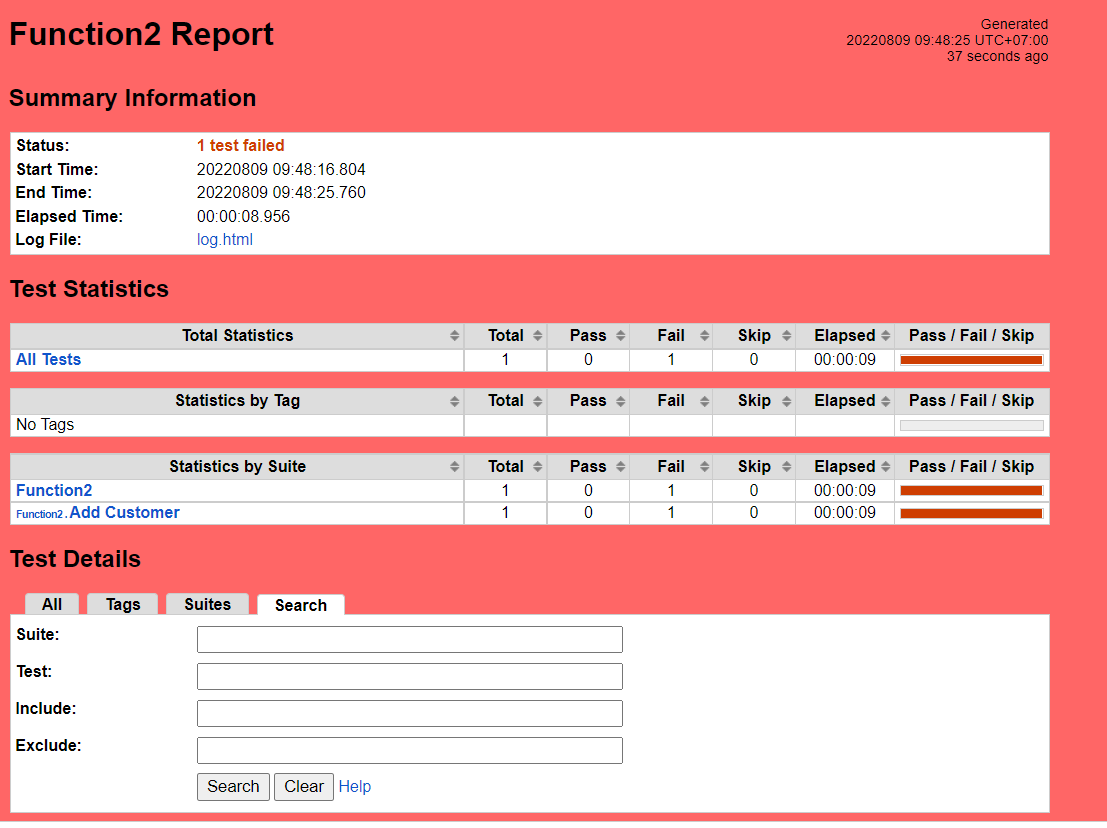
### Kết quả đạt được.

* **Giao diện sau khi thực kiểm thử**

****

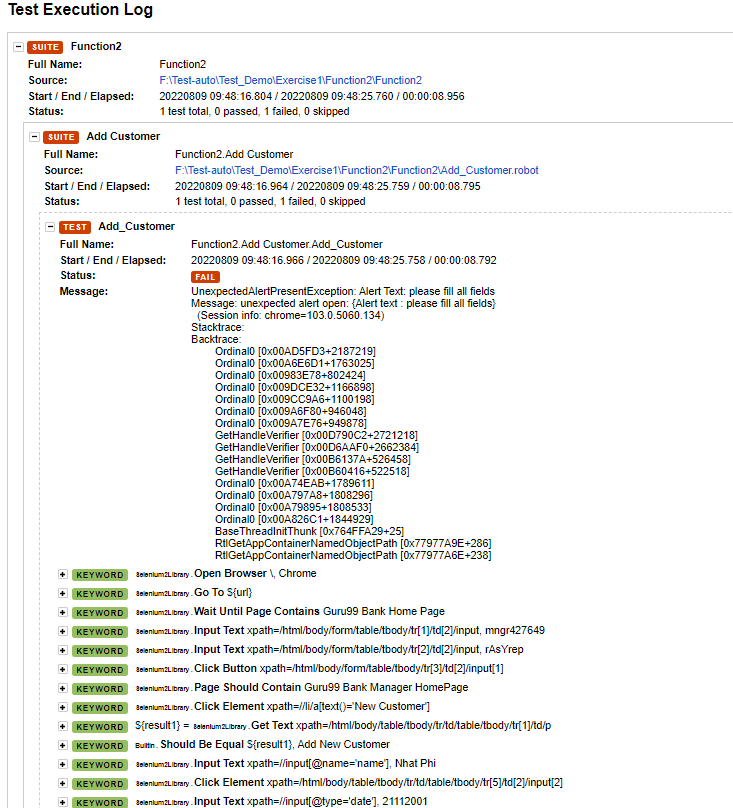
Hình 4.3.2‑1Giao diện sau khi kiểm thử

* **Test Report**

****

Hình 4.3.2‑2 Test Report chức năng thêm khách hàng

* **Test Log**

****

Hình 4.3.2‑3 Test log chức năng thêm khách hàng(1)

****

Hình 4.3.2‑4 Test log chức năng thêm khách hàng (2)

### KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. **Kết quả đạt được**

Sau thời gian 8 tuần thực tập tại doanh nghiệp TMA, bản thân em đã có những thay đổi và tiến bộ theo chiều hướng tích cực như sau:

* Nâng cao khả năng tự học, khả năng tìm hiểu kiến thức một cách chủ động..
* Cải thiện kĩ năng giao tiếp, nâng cao hiểu biết của bản thân về môi trường làm việc doanh nghiệp chuyên nghiệp.
* Kỹ năng tiếng anh được cải thiện, gia tăng khả năng đọc hiểu tài liệu bằng tiếng anh, cải thiện khả năng giao tiếp bằng tiếng anh.
* Có thêm nhiều kiến thức mới mẻ mà chưa được học, kiến thức liên quan về mảng Tester cũng như các công việc liên quan được nâng cao.
* Nắm rõ quy trình phát triển phần mềm và quản lý dự án.
* Nhờ các buổi Training của công ty TMA mà các kỹ năng mềm của bản thân được nâng cao như: Viết email chuyên nghiệp hơn, làm và trình bày slide chuyên nghiệp hơn, khả năng trình bày trước đám đông được cải thiện rõ, kỹ năng làm việc nhóm tốt hơn.
* 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.

1. **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, với những gì đã học được, 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.

Vì thời gian thực tập không cho phép do đó bản thân em chưa được tìm hiểu thêm nhiều về Tester. Kĩ năng, cũng như 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 kiểm thử tự động trên môi trường Robot Framework. Vì thế, thời gian tới em rất mong muốn được học hỏi và tìm hiểu thêm về Automation Testing trên nhiều công cụ khác. 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

<https://www.guru99.com/software-testing.html>

<https://www.guru99.com/scrum-vs-kanban.html>

<https://viblo.asia/p/tim-hieu-ve-sdlc-software-development-life-cycle-3Q75w209lWb>

<https://existek.com/blog/sdlc-models/>

<https://www.w3schools.com/python/python_syntax.asp>

<https://hr1tech.com/vi/news/quy-trinh-kiem-thu-phan-mem-software-testing-life-cycle-stlc-121.html>

<https://www.w3schools.com/xml/xpath_intro.asp>

<https://www.w3schools.com/html/html_attributes.asp>

<https://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html>

<https://robotframework.org/robotframework/latest/libraries/BuiltIn.html>

<http://robotframework.org/Selenium2Library/Selenium2Library.html>

# PHỤ LỤC