UNIVERSITY OF ECONOMICS AND LAW

**FACULTY OF INFORMATION SYSTEMS**

**GRADUATION INTERNSHIP**

**REPORT**

MAJOR IN MANAGEMENT INFORMATION SYSTEM

**BUILDING TEST CASE AND APPLICATIONS ROBOT FRAMEWORK IN AUTOMATION TESTING FOR TMS SOFTWARE**

**Company/Organization:**

**NashTech Viet Nam (Harvey Nash Group)**

Supervisor: **M.A Tran Thi Anh**

Student:

Student ID: K184060713

Full name: Nguyen Thi Thanh Hoa

Class: K18406

**Ho Chi Minh City, 4/2022**

**ACKNOWLEDGEMENTS**

After more than two months of internship at NashTech - Harvey Nash Vietnam, I have learned a lot of knowledge and skills to solve problems in work and life. This will be a very meaningful time for me to experience the position of a QC Engineer intern at this company. This job has helped me to form a sense of responsibility and initiative in solving all problems raised. In addition, it also motivates me to keep going in my career and pursue my goals.

First of all, I would like to thank the teachers in the Faculty of Systems for their dedication to imparting useful knowledge to me so that I can apply it today. I would like to thank Ms. Tran Thi Anh for her advice and experience and for taking the time to help me complete this report.

Thank you to Mr.Vinh and Ms.Hoa, who guided me and gave me the opportunity to participate in the company's projects so that I could complete the report in the best way. Thank you for your dedicated teaching during the internship period.

I also want to thank Ms. Yen and other members of the training team for their full support and willingness to help me complete this report.

Finally, I would like to wish your company, colleagues and teachers good health, success and happiness in work and life.

Thanks and best regards, Nguyen Thi Thanh Hoa.

**INTERNSHIP RESULT ASSESSMENT FORM**

ID student: K184060713

Full name: Nguyen Thi Thanh Hoa

Name of internship company: NashTech (Harvey Nash VietNam)

Company address: E-town, 3rd floor, 364 Cong Hoa Street, Ward 13, Tan Binh District

Company phone: (+84) 28 3810 6200

Full name of company representative: Vu Thuy Hoang Yen

Position: Training & Development Executive Phone: 0908362421   
Internship period: 11/01/2022 – 06/05/2022

*\* Please evaluate* ***by marking X*** *in the classification columns of the following table.*

*Notes:*

***Type A: 2 points****;* ***Type B: 1,5 points***

***Type C: 1 point****;* ***Type D: 0,5 point***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Content of assessment | Rating | | | |
| **A** | **B** | **C** | **D** |
| **1. Attitude and discipline** | | | | |
| 1. Implement company’s policies | X |  |  |  |
| 1. Comply with working hours | X |  |  |  |
| 1. Communication attitude with colleagues and associates |  | X |  |  |
| 1. Property protection consciousness | X |  |  |  |
| 1. Be active at work | X |  |  |  |
| **2. Advanced skills and qualifications** | | | | |
| 1. Meet job requirements |  | X |  |  |
| 1. The spirit of learning and improving professional qualifications | X |  |  |  |
| 1. Be innovative and dynamic at work | X |  |  |  |
| **3. Internship results** | | | | |
| 1. Have practical application products that benefit the company | X |  |  |  |
| 1. The level of completion of the internship |  | X |  |  |

**COMMENTARY FORM OF THE INTERNSHIP COMPANY**

- Have clear/well in communication with others

- Be proactive in studying new things from program

- Have high responsibility to complete lessons/exercises of courses as expectation

- Need to build more skills to be stronger in working environment in future

Ho Chi Minh, April 12st, 2022

**CONFIRMATION OF THE COMPANY**

*Sign, write full name and stamp*

**INTERNSHIP REPORT RESULT ASSESSMENT FORM**

**(For instructor)**

ID\_student: K184060713

Full name: Nguyen Thi Thanh Hoa

Instructors: M.A Tran Thi Anh

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Criteria** | **Specific criteria** | **Point** | **Note** |
| 1 | **Report Format (20%)** | Presentation (5%) |  |  |
| Report structure (10%) |  |  |
| Writing style (5%) |  |  |
| 2 | **Report content (45%)** | Analysis skill (5%) |  |  |
| Objective (10%) |  |  |
| Specialized knowledge results (30%) |  |  |
| 3 | **Student’s attitude (15%)** |  |  |  |
| 4 | **Evaluation of enterprise (20%)** |  |  |  |
| **TOTAL** | | |  |  |

Ho Chi Minh, April 15th, 2022

**Instructors**

*Sign, write full name*

**COMMENTARY FORM OF INSTRUCTORS**

**Instructors**

*Sign, write full name*

**WORKING SCHEDULE AND ASSIGNED TASKS**

Time: 10/01/2022 – 08/04/2022

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Time** | **Task** | **Workplace** |
| 1 | 10/01 – 14/01 | * Training about Testing * Estimation Testing | Work from Home |
| 2 | 17/01 – 21/01 | - Analysis of requirements  - Build Test Case for the following functions:  + Add and Edit User Roles  + Delete User Role | Work from Home |
| 3 | 14/02 – 18/02 | - Analysis of requirements  - Build Test Case for the following functions:  + Login and Logout  + Forget Password  + Create Employee  + Search Employee | Work from Home |
| 4 | 21/02 – 25/02 | - Analysis of requirements  - Build Test Case for the following functions:  + Manage Project's Information  - Run Test Cases and Raise Bugs for the following functions:  + Login and Logout  + Forget Password  + Search Employee | NashTech Office |
| 5 | 28/02 – 04/03 | - Run Test Cases and Raise Bugs for the following functions:  + Manage Project's Information  - Training on git and git lab  - Training on automation testing and get locators | NashTech Office |
| 6 | 07/03 – 11/03 | - Applying Selenium IDE tool to automation testing  - Training on Page Object Model | NashTech Office |
| 7 | 14/03 – 18/03 | - Training on Robot Framework  - Build test automation using Robot Framework | NashTech Office |
| 8 | 21/03 – 25/03 | - Write Test Case for API Testing  - Execute API Testing in Postman software | NashTech Office |
| 9 | 28/03 – 01/04 | - SQL Query for Testing  - Test Strategy and Testing process convention | NashTech Office |
| 10 | 04/04 – 08/04 | - Accessibility Testing  - Regression Test | NashTech Office |

**CONTENTS**

[CHAPTER 1: TOPIC OVERVIEW 1](#_Toc100699200)

[1.1. Reasons 1](#_Toc100699201)

[1.2. Objectives 2](#_Toc100699202)

[1.3. Objects and scopes 2](#_Toc100699203)

[1.4. Implementation plan 2](#_Toc100699204)

[1.5. Structure 3](#_Toc100699205)

[CHAPTER 2: INTRODUCTION TO THE COMPANY AND PROJECT 4](#_Toc100699206)

[2.1. Introduction to the company 4](#_Toc100699207)

[2.2. Introduction to the project 6](#_Toc100699208)

[CHAPTER 3: THEORETICAL BACKGROUND 8](#_Toc100699209)

[3.1. Automation Testing 8](#_Toc100699210)

[3.2. Robot Framework 8](#_Toc100699211)

[3.2.1. Introduction 8](#_Toc100699212)

[3.2.2. Robot Framework Features 8](#_Toc100699213)

[3.2.3. Robot Framework Advantages 10](#_Toc100699214)

[3.2.4. Robot Framework Limitations 10](#_Toc100699215)

[3.2.5. Conclusion 10](#_Toc100699216)

[CHAPTER 4: SURVEY CURRENT STATUS, DETERMINE REQUIREMENTS 11](#_Toc100699217)

[4.1. Survey current status 11](#_Toc100699218)

[4.2. Actors 11](#_Toc100699219)

[4.3. Conditions 12](#_Toc100699220)

[4.4. Flow of Events 12](#_Toc100699221)

[4.4.1. Basic Flow – Add a new Project 12](#_Toc100699222)

[4.4.2. Alternative Flow – View Project Information 12](#_Toc100699223)

[4.4.3. Alternative Flow – Update Project Information 13](#_Toc100699224)

[4.4.4. Alternative Flow – Delete Project 13](#_Toc100699225)

[4.5. Suggested New Application Screens 15](#_Toc100699226)

[4.5.1. Screen – Project screen 15](#_Toc100699227)

[4.6. Business Rules 19](#_Toc100699228)

[4.7. Message 19](#_Toc100699229)

[CHAPTER 5: BUILD TEST CASE 21](#_Toc100699230)

[5.1. Create Project 21](#_Toc100699231)

[5.2. Edit Project 31](#_Toc100699232)

[5.3. View Project 40](#_Toc100699233)

[5.4. Delete Project 42](#_Toc100699234)

[CHAPTER 6: APPLICATION ROBOT FRAMEWORK 47](#_Toc100699235)

[6.1. Install Robot FrameWork 47](#_Toc100699236)

[6.2. Components of RIDE IDE 48](#_Toc100699237)

[6.2.1. Setting 48](#_Toc100699238)

[6.2.2. Test Cases 49](#_Toc100699239)

[6.2.3. Keywords 50](#_Toc100699240)

[6.2.4. Variables 50](#_Toc100699241)

[6.3. Create test project 51](#_Toc100699242)

[6.3.1. Create test demo 51](#_Toc100699243)

[6.3.2. Test Result 58](#_Toc100699244)

[CHAPTER 7: CONCLUSION 59](#_Toc100699245)

[7.1. Results: 59](#_Toc100699246)

[7.2. Limitations 59](#_Toc100699247)

[7.3. Future works 59](#_Toc100699248)

[REFERENCES 61](#_Toc100699249)

LIST OF FIGURES

[Figure 2.1: Logo of Harvey Nash Group & members 4](#_Toc100699367)

[Figure 2.2: Location of Harvey Nash 5](#_Toc100699368)

[Figure 2.3: Some of Nash Tech’s clients 6](#_Toc100699369)

[Figure 4.1: Project Information Tab 15](#_Toc100699370)

[Figure 6.1: Download Python 47](#_Toc100699371)

[Figure 6.2: Screen of RIDE 48](#_Toc100699372)

[Figure 6.3: Setting of RIDE 49](#_Toc100699373)

[Figure 6.4: Test Case of RIDE 49](#_Toc100699374)

[Figure 6.5: Keywords of RIDE 50](#_Toc100699375)

[Figure 6.6: Variables of RIDE 51](#_Toc100699376)

[Figure 6.7: Steps to import library 51](#_Toc100699377)

[Figure 6.8: Steps to create variables 52](#_Toc100699378)

[Figure 6.9: Keywords - Go to Create Project Page 53](#_Toc100699379)

[Figure 6.10: Keyword – Enter Project Name 54](#_Toc100699380)

[Figure 6.11: Keyword – Select ProjectType 54](#_Toc100699381)

[Figure 6.12: Keyword - Click Create Project 55](#_Toc100699382)

[Figure 6.13: Keyword – Verify Project displays correctly - Part 1 56](#_Toc100699383)

[Figure 6.14: Keyword – Verify Project displays correctly - Part 2 56](#_Toc100699384)

[Figure 6.15: Test Case - Create Project with all fields successfully 56](#_Toc100699385)

[Figure 6.16: Button to Run Test 57](#_Toc100699386)

[Figure 6.17: Screen of Test Case successfully 58](#_Toc100699387)

LIST OF TABLES

[Table 1.1: Implementation plan 3](#_Toc100620717)

[Table 4.1: Actors of TMS softsware 12](#_Toc100620718)

[Table 4.2: Basic Flow - Add a new Project 12](#_Toc100620719)

[Table 4.3: Alternative Flow – View Project Information 13](#_Toc100620720)

[Table 4.4: Alternative Flow – Update Project Information 13](#_Toc100620721)

[Table 4.5: Alternative Flow – Delete Project 14](#_Toc100620722)

[Table 4.6: Requirements 19](#_Toc100620723)

[Table 4.7: Business Rules 19](#_Toc100620724)

[Table 4.8: Description of message 20](#_Toc100620725)

[Table 5.1: Test Cases of Create Project 31](#_Toc100620726)

[Table 5.2: Test Cases of Edit Project 40](#_Toc100620727)

[Table 5.3: Test Cases of View Project 42](#_Toc100620728)

[Table 5.4: Test Cases of Delete Project 46](#_Toc100620729)

[Table 6.1: Variables 53](#_Toc100620730)

LIST OF ACRONYMS

|  |  |
| --- | --- |
| QC | Quality Control |
| PM | Project Manager |
| SM | Senior Manager |
| DM | Delivery Manager |
| IT | Information Technology |
| BPS | Business Process Services |
| CMMI | Capability Maturity Model Integration |
| RIDE | Robot Integrated Development Environment |
| MSG | Message |
| UI | User Interface |

# **TOPIC OVERVIEW**

* 1. **Reasons**

With the rapid development of information technology in general and software technology in particular, software development is increasingly supported by many advanced tools, making software development less tiring and more effective. However, because of the complexity of the software and the limitations of time and cost, even though software quality assurance activities in general and testing, in particular, are increasingly rigorous and scientific, it is still not guaranteed. ensure that the software products being applied are error-free. Errors are always latent in every software product and cause incalculable damage.

Software testing is a continuous process, throughout every phase of software development to ensure that the software meets the design requirements and that those requirements meet the needs of the users. Software testing techniques have been studied, and software testing has become a mandatory process in software development projects worldwide. Software testing is very expensive, time-consuming, and challenging to detect activity. So, software testing requires the right strategy, a reasonable plan and tightly managed execution.

Test automation is the software testing trend. It has become the key enabler of many advanced development and deployment practices. As we can see, switching from manual testing to automation will reduce the waiting time. Automated tests are completed rapidly and can be run repeatedly. With test automation, you are more likely to have error-free releases. Thanks to minimized human intervention, automated testing is more accurate than testing applications manually. The thing is, a human tester can make mistakes during any step of evaluation. But the machine won’t. Generated test cases are more precise than human testers, which means that by eliminating human errors, reduce the risk of failure.

Those are the reasons that motivated me to do this topic “Building test case and applications robot framework in automation testing for TMS software”. After more than 3 years of studying at the school, because of the desire to have more practical experience, as well as to participate in learning in a professional environment. Therefore, I decided to choose Nash Tech - an ideal, modern, professional environment - as the place that will help me realize this plan.

* 1. **Objectives**

The topic explores the theoretical basis of testing in general, builds a Test Case system in particular, as well as how to deploy software automation testing tools to reduce testing personnel and ensure software quality better with manual testing. The main objective of the project is TMS software testing.

One of the critical objectives of this report is to improve software quality through software testing. High-Quality software means a lesser number of defects. In other words, the more efficient the testing process is, the fewer errors we will get in the end product. Which, in turn, will increase the overall quality of the test object. Excellent quality contributes to a significant increase in customer satisfaction as well as lower maintenance costs.

Another essential objective this report is to identify all defects in a product. The main motto of testing is to find maximum defects in a software product while validating whether the program is working as per the user requirements or not. Defects should be identified as early in the test cycle as possible.

* 1. **Objects and scopes**

**Objects:** This topic focuses on analyzing requirements to build Test Cases for the project. At the same time, apply Robot FrameWork to the automation testing process for TMS software.

**Scope:** Because of the limitation of the report, the report will focus the analysis on the Manage Project’s Information function of TMS software.

* 1. **Implementation plan**

|  |  |
| --- | --- |
| **Period** | **Work** |
| Period 1 | Analyze and define requirements |
| Period 2 | Build Test Case |
| Period 3 | Execute Test Case and Raise Bugs |
| Period 4 | Application Robot FrameWork |
| Period 5 | Test report |

Table 1.1: Implementation plan

* 1. **Structure**

The report consists of 6 chapters as follows:

CHAPTER 1: TOPIC OVERVIEW

In this chapter, the report presents the following content: Overview of the Reasons, Objectives, objects and research scope of the topic; implementation plan and project structure.

CHAPTER 2: INTRODUCTION TO THE COMPANY AND PROJECT

Introduction about NashTech (a part of Harvey Nash Group) and Information project of TMS software.

CHAPTER 3: THEORETICAL BACKGROUND

Introduction to Automation Testing and Robot FrameWork.

CHAPTER 4: SURVEY CURRENT STATUS, DETERMINE REQUIREMENTS

Survey current status,Analysis and details of Basic Flow and Alternative Flow, conditions for building Test Case.

CHAPTER 5: BULID TEST CASE

The detailed process of the test case in four functions: Create Project, Edit Project, View Project and Delete Project.

CHAPTER 6: APPLICATION ROBOT FRAMEWORK

Execute and Implement some of test case by Robot FrameWork.

CHAPTER 7: CONCLUSION

Outlining a number of points achieved and limitations of the topic when participating in the project.

# **INTRODUCTION TO THE COMPANY AND PROJECT**

## **2.1. Introduction to the company**

Harvey Nash is a global recruitment consultancy and IT outsourcing service provider. Established in 1988, Harvey Nash has supported many of the world's leading organizations to recruit, source and manage the highly skilled talent they need to succeed in an increasingly competitive and technology driven world.



Figure 2.1: Logo of Harvey Nash Group & members

Harvey Nash - Technology Recruitment: Leadership recruitment, Tech expert recruitment, Solutions.

Nash Tech - IT Solutions: Software Development, Technology Advisory, Business Process Management.

Alumni - Leadership Services: Executive Search & interim, Assessment services, Leadership & Board development.

With over 2,500 employees in 36 locations across the USA, Europe and Asia-Pacific the Group has the reach and resources of a global organization, whilst fostering a culture of innovation and agility that empowers its people to respond to constantly changing client needs.



Figure 2.2: Location of Harvey Nash

NashTech is a division of Harvey Nash Group and is UK-owned and regulated, the world's leading provider of technology solutions.

• Established in 1988 and development centers in Vietnam since 2000.

• Employs over 2,000 of Vietnam's best-trained specialists and works in over 45 countries globally.

• Location in Ho Chi Minh: 364 Cong Hoa street, Etown Tan Binh.

NashTech  provides a unique portfolio of services including:

• Software Development

• Business Process Services (BPS)

• IT Infrastructure & Maintenance

• Communications Technology R&D

NashTech is an expert in technology, delivering smart solutions that solve business challenges and create value. The award-winning teams apply deep expertise and passion to deliver complex IT projects globally.

Up to now, NashTech has become the number two technology company in Vietnam’s market (ranked by Vinasa - 2016). The business operates in 42 offices worldwide and it is the strongest business of Vietnam in Europe. NashTech is also one of the most sustainable companies in Vietnam with innovation at our heart and seeing significant business growth.

Over 20 years of development in Vietnam, NashTech is proud to have built and trained thousands of talented Vietnamese engineers and experts, contributing to putting Vietnam on the map of Information Technology and Outsourcing Solutions of the world. gender. During its operation, NashTech has achieved many typical achievements such as 13 consecutive years of winning the Sao Khue title, the top 4 best places to work in Vietnam in the field of Information Technology, the top 50 leading companies in the world. Leading in the field of Information Technology in Vietnam and becoming one of the few companies in the world to achieve the highest certificates of CMMi L5 V2.0 and certified as a Microsoft Gold Partner in 7 areas of main technology.

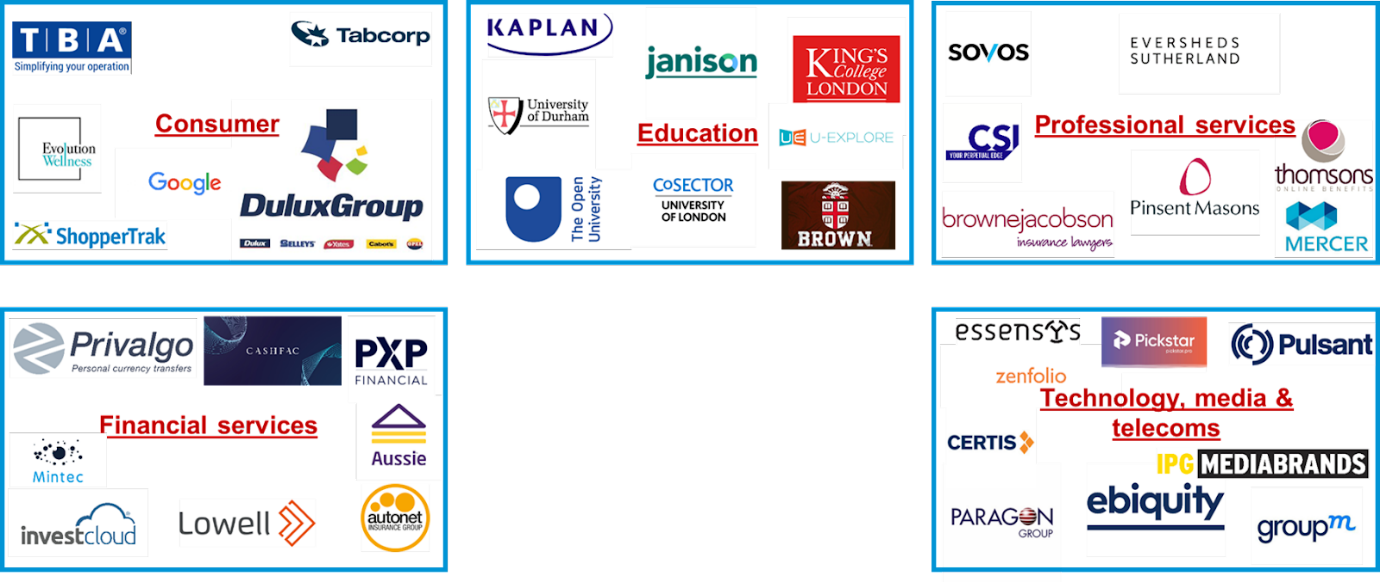


Figure 2.3: Some of Nash Tech’s clients

## **2.2. Introduction to the project**

* Job position: QC Engineer Intern
* Responsibility and authority: As assigned by the mentor.
* Assigned project: TMS is software to manage projects and employees used for project planning, resource allocation and scheduling. It enables project managers as well as entire teams to control resources, quality management and all documentation exchanged throughout a project. This software also serves as a platform for facilitating collaboration among project stakeholders.
* Tasks assigned and performed:
* Participate in review and analyze the project requirement to understand test objectives, provide input on testability of requirements and estimation for the testing activities.
* Collaborate with other QC Engineers to develop effective strategies and test plans.
* Develop manual test cases and prioritize testing activities.
* Execute all the manual test case and report defects, define severity and priority for each defect.
* Prepare the reports related to software testing carried out.
* Ensure that all tested related work is carried out as per the defined standards and procedures.

# **THEORETICAL BACKGROUND**

* 1. **Automation Testing**

Automation testing is the process of testing software and other tech products to ensure it meets strict requirements. Essentially, it’s a test to double-check that the equipment or software does exactly what it was designed to do. It tests for bugs, defects, and any other issues that can arise with product development.

Although some types of testing, such as regression or functional testing can be done manually, there are greater benefits of doing it automatically. Automation testing can be run at any time of the day. It uses scripted sequences to examine the software. It then reports on what’s been found, and this information can be compared with earlier test runs.

* 1. **Robot Framework**
     1. **Introduction**

Robot Framework is an open-source test automation framework for acceptance testing and acceptance test-driven development. It follows different test case styles − keyword-driven, behavior-driven and data-driven for writing test cases. Robot Framework provides support for external libraries, tools which are open source and can be used for automation. The most popular library used is Selenium Library used for web development & UI testing.

Test cases are written using keyword style in a tabular format. You can use any text editor or Robot Integrated Development Environment (RIDE) for writing test cases.

Robot framework works fine on all the Operating Systems available. The framework is built on Python and runs on Jython (JVM) and IronPython (.NET).

* + 1. **Robot Framework Features**

**Tabular format for test cases**

Robot framework comes with a simple tabular format where the test cases are written using keywords. It is easy for a new developer to understand and write test cases.

**Keywords**

Robot framework comes with built-in keywords available with robot framework, keywords available from the libraries like Selenium Library (open browser, close browser, maximize browser, etc.). We can also create user-defined keywords, which are a combination of other user-defined keywords or built-in or library keywords. We can also pass arguments to those keywords, which make the user-defined keywords like functions that can be reused.

**Variables**

Robot framework supports variables – scalar, list and dict. Variables in robot framework are easy to use and are of great help while writing complex test cases.

**Libraries**

Robot framework has support for a lot of external libraries like SeleniumLibrary, Database Library, FTP Library and http library. SeleniumLibrary is mostly used as it helps to interact with the browsers and helps with web application and UI testing. Robot framework also has its own built-in libraries for strings, date, numbers etc.

**Resources**

Robot framework also allows the import of robot files with keywords externally to be used with test cases. Resources are very easy to use and are of great help when we need to use some keywords already written for other test projects.

**Data driven test cases**

Robot framework supports keyword driven style test cases and data driven style. Data driven works with high-level keyword used as a template to the test suite and the test cases are used to share data with the high-level keyword defined in the template. It makes the work very easy for testing UI with different inputs.

**Test Case Tagging**

Robot framework allows to tag test-cases so that we can either run the tags test-cases or skip the tagged testcases. Tagging helps when we want to run only a group of test cases or skip them.

**Reports and Logs**

Robot framework provides all the details of test suite, test case execution in the form of report and logs. All the execution details of the test case are available in the log file. The details like whether the test case has failed or passed, time taken for execution, steps followed to run the test case are provided.

**RIDE**

This editor available with Robot framework helps in writing and running test cases. The editor is very easy to install and use. RIDE makes life easy for writing test cases by providing framework specific code completion, syntax highlighting, etc. Creation of project, test suite, test case, keywords, variables, importing library, executing, tagging the test case is easily done in the editor. Robot framework also provides plugins for eclipse, sublime, Textmate, Pycharm that has support for robot test cases.

* + 1. **Robot Framework Advantages**

Robot framework is open source, so anyone who wants to try out can easily do so.

* It is very easy to install and helps in creating and executing test cases. Any new comer can easily understand and does not need any high-level knowledge of testing to get started with robot framework.
* It supports keyword-driven, behavior-driven and data-driven style of writing test cases.
* It is a good support for external libraries. Most used is Selenium Library, which is easy to install and use in robot framework.
  + 1. **Robot Framework Limitations**

Robot lacks support for if-else, nested loops, which are required when the code gets complex.

* + 1. **Conclusion**

Robot Framework is an open-source test automation framework for acceptance testing and acceptance test-driven development. The test cases in Robot Framework are based on keywords written in tabular format, which makes it clear and readable, and conveys the right information about the intention of the test case. For example, to open browser, the keyword used is “Open Browser”.

# **SURVEY CURRENT STATUS, DETERMINE REQUIREMENTS**

* 1. **Survey current status**

Through TMS Software, the company easily manages information flows in the enterprise. From tracking down deliverables to managing resources and from time management to collaborating with team members, there is a lot to be considered when running and managing projects. Here is a list of the functional aspects of project management software.

* Login-Logout – when logging into different accounts, each account will have its own permissions. This makes it possible for project managers to view and edit only projects related to their role.
* Manage project – being able to assign and update the status of tasks, roles, case studies so that everyone in the team is able to follow the project. Besides, the Project Manager or project leader can create, edit, delete, search and view projects.
* Manage employee – Allows users to add, delete, edit, search and view employee information. This helps managers manage the number of employees and assign reasonable permissions for each different employee account.

In this report, focusing on analyzing the Manage Project function. From this function will provide Admin, and Project Manager (PM) the abilities to maintain the project information.

* 1. **Actors**

| **Actor** | **Description** | **Scope** | **Condition** |
| --- | --- | --- | --- |
| Admin | Has full control on the use case | Full control | Full control |
| PM | Create Projects  Update Projects  View Projects  Delete Projects | Projects they are PM | No Assignments AND No Case Studies Created |
| SM | Projects they are DM |

Table 4.1: Actors of TMS softsware

* 1. **Conditions**

Precondition: The actor must be signed on to the system.

Postcondition: The actor managed the project information.

* 1. **Flow of Events**
     1. **Basic Flow – Add a new Project**

| **Step** | **Event Description** |
| --- | --- |
| 1 | Click **Project** on the Menu bar  Select **Create Project** in the dropdown list  Jump to The Project Information Screen  Project Role, Project Member, and Case Studies are **NOT** accessible |
| 2 | The user enters project information and clicks Save button |
| 3 | The system validates inputted data:   * If any error happens, the system shows error message correspondingly. * If all validations were passed, then the Projects is created and following information is added: Go to Step 4 |
| 4 | The Project Role, Project Member, Case Studies are now accessible |

Table 4.2: Basic Flow - Add a new Project

* + 1. **Alternative Flow – View Project Information**

| **Step** | **Event Description** |
| --- | --- |
| 1. | Click on Project Name in **the search results** (Refer to UC16)  Or  If the user has opened a Project, the user clicks on “Project Information” tab. |
| 2. | The system retrieves data and shows the Project Information screen. |
|  | The flow ends |

Table 4.3: Alternative Flow – View Project Information

* + 1. **Alternative Flow – Update Project Information**

| **Step** | **Event Description** |
| --- | --- |
|  | On Project Information screen (read mode), the user clicks on “Edit” button |
| 2. | The system displays the corresponding screen. |
| 3. | The user updates the project information and clicks “Save” button |
| 4 | The system validates the data.   * If failed, the system shows error message correspondingly * Else, the system saves the updated information and the screen switches to read mode. |
|  | Flow ends |

Table 4.4: Alternative Flow – Update Project Information

* + 1. **Alternative Flow – Delete Project**

| **Step** | **Event Description** |
| --- | --- |
| 1. | On the Project Information, the user clicks on “Delete” button |
| 2. | If Case Study > 0 OR Assignment > 0 Checks:   |  |  | | --- | --- | | If at least 1 member’s Responsibility = **Approved**  OR  If at least 1 Case Study = **Approved**  Then  Shows MSG 2  Go to step 3.a | If at least 1 member’s Responsibility = **Waiting for approval**  OR  If at least 1 Case Study = **Waiting for approval**  Then  Shows MSG 3  Go to step 3.c |   If Case Study = 0 AND Assignment = 0 then Shows MSG 4, Go to step 3.b |
| 3.a | Click OK to close MSG. Do Nothing. The flow ends |
| 3.b | Click **No**:   * Do nothing * Close MSG. The flow ends.   Click **Yes**:   * Delete Project * Close the MSG * Return to search project screen * Show MSG5 on the top |
| 3.c | Click OK to close MSG. Do Nothing. The flow ends |
|  | The flow ends |

Table 4.5: Alternative Flow – Delete Project

* 1. **Suggested New Application Screens**
     1. **Screen – Project screen**

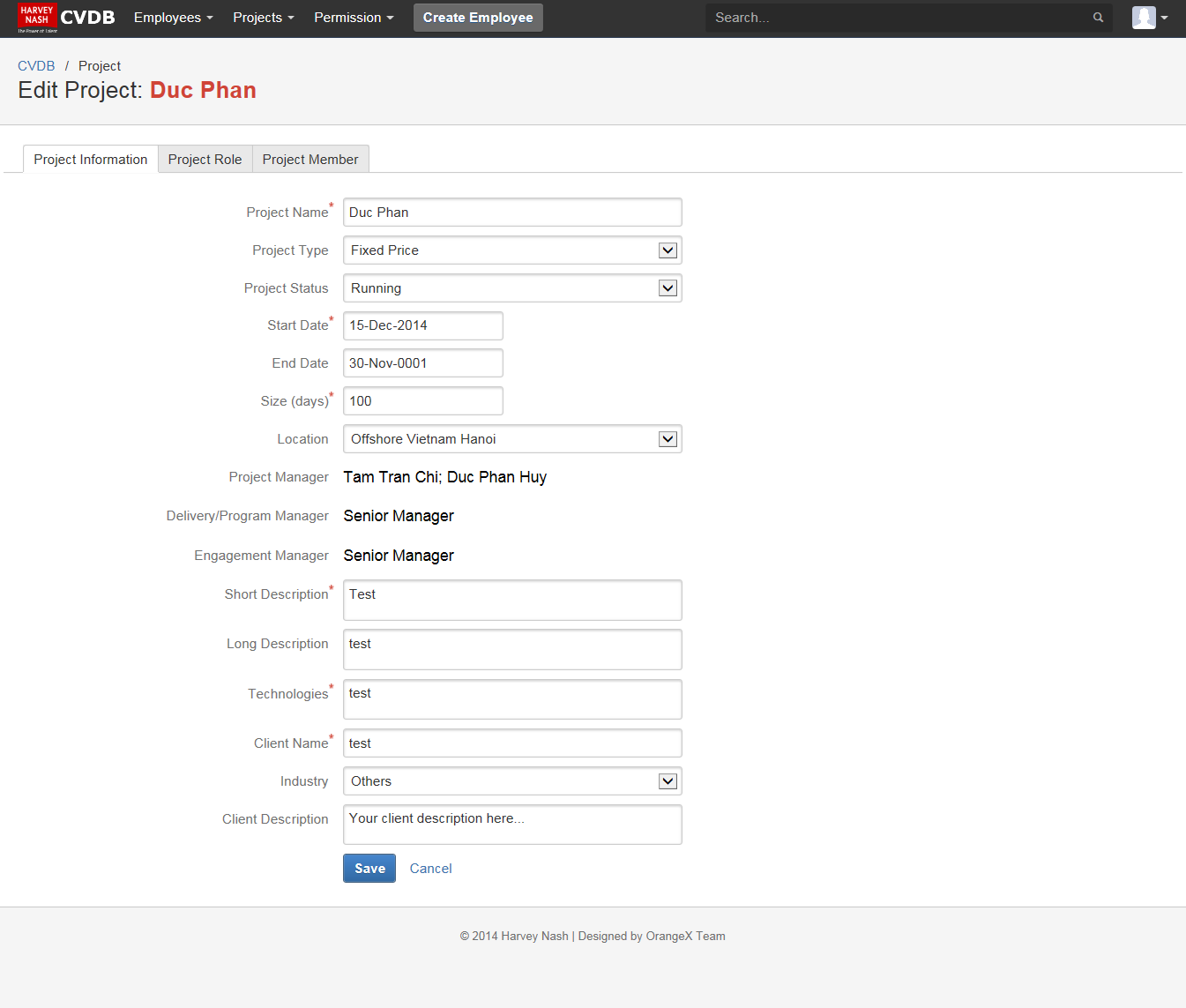


Figure 4.1: Project Information Tab

| **BUTTON** | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **Control** | **Description** | | **Required** | | **Validation** |
| Create / Save | Button | Save the latest information | |  | | Only shows in Edit Mode  Display Create if Project = New  Display Save if Project <> New |
| Cancel | Button | Discard changed information | |  | |  |
| Edit | Button | Change Read Mode to Edit Mode | |  | | Visible only in Read Mode  When delete project, navigate to search project page |
| Delete | Button | Delete Project | |  | |
| **PROJECT TABS** | | | | | | |
| **Project Information** | | | | | | |
| In **Read Mode**:  Enable to switch between Project Roles / Project Members / Project Case Studies tab  Unable to edit Fields value  In **Edit Mode**:  Unable to switch between Project Roles / Project Members / Project Case Studies tab  Enable to edit Fields value | | | | | | |
| **Project Members** | | | | | | |
| Refer to UC05: Manage Project’s members  The tab is accessible only when all of other tabs in read mode. | | | | | | |
| **Case Studies** | | | | | | |
| Refer to UC06: Manage Project’s Case Studies  The tab is accessible only when all of other tabs in read mode. | | | | | | |
| **Project Roles** | | | | | | |
| Refer to UC17 Manage Project’s Roles  The tab is accessible only when all of other tabs in read mode. | | | | | | |
| **INPUTS** | | | | | | |
| **Field Name** | **Control** | **Description** | **Required** | | **Validation** | |
| Project Name | Textbox | Name of the project | Y | | Min: 1 char  Max: 100 chars | |
| Project Type | Dropdown List | Type of the project | Y | | Default value: Please select project type  Valid Values:   * Fixed Price * ODC * Time & Material | |
| Project Status | Dropdown List | Status of the project | Y | | Default value: Please select project status  Valid Values:   * Cancelled * Closed * Pending * Running | |
| Start Date | Date picker | The date the project starts | Y | | Allow all date (past and future) | |
| End Date | Date Picker | The date the project ends |  | | Automated generated when project closed | |
| Size (days) | Number | The project size | Y | |  | |
| Location | Dropdown List | Location where the project is maintained |  | | Default value: Offshore Vietnam Ho Chi Minh city  Valid values: Refer to Reference file | |
| Project Manager | Dropdown List | Project’s PM | Y | | Values List: Project manager role users | |
| Delivery/ Program Manager | Dropdown List | Project’s DM or Program Manager | Y | | Values List: Senior manager users | |
| Engagement Manager | Dropdown List | Project’s EM | Y | | Values List: Senior manager users | |
| Short Description | Text Area | Short description of the project | Y | | Min: 1 char  Max: 1000 chars | |
| Long Description | Text Area | Full description of the project | Y | | Min: 1 char  Max: 4000 chars | |
| Technologies | Text Area | Technologies that are used in the project | Y | | Min: 1 char  Max: 4000 chars | |
| Client Name | Textbox | Name of the client | Y | | Min: 1 char  Max: 100 chars | |
| Client Industry/Sector | Dropdown List |  |  | | Default value: Select an Industry/Sector  Valid Values:   * Energy * Financial Services * Health Care * Manufacturing * Public Sector * Retail * Technology * Telecommunication * Others | |
| Client Description | Text Area | Description about client |  | | Min: 0 char  Max: 4000 chars | |

Table 4.6: Requirements

* 1. **Business Rules**

| **Business Rule ID** | **Description** |
| --- | --- |
| 1 | When click Save or Create, check all **Required** fields. If at least one field = **empty**, show MSG1 next to the field |

Table 4.7: Business Rules

* 1. **Message**

| **Message ID** | **Description** |
| --- | --- |
| MSG 1 | This is a required field. |
| MSG 2 | This project has members / case studies approved. You are not allowed to delete this project |
| MSG 3 | This project has members’ responsibilities / case studies that are waiting for approval. If you want to delete this project, these members or case studies must be rejected first. |
| MSG 4 | Are you sure you want to delete this project? |
| MSG 5 | The project was deleted |

Table 4.8: Description of message

# **BUILD TEST CASE**

## **5.1. Create Project**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Step** | **Expected Output** |
| 1 | Verify that default value and UI of Create a new project Screen | | |
|  | Pre-condition: The user must have logged into the system | 1. Click on Project on the Menu bar |  |
|  | 2. Select Create Project in the Project dropdown list |  |
|  | 3. Check UI of Default Create a new project Screen | The system shows the Create a new project Screen contains the follows: - Project Name (textbox) - Project Type (dropdown list) - Project Status (dropdown list) - Start Date (date picker) - End Date (date picker) - Size (days) (number) - Location (dropdown list) - Project Manager (dropdown list) - Delivery/ Program Manager (dropdown list) - Engagement Manager (dropdown list) - Short Description (text area) - Long Description (text area) - Technologies (text area) - Client Name (textbox) - Client Industry/Sector (dropdown list) - Client Description (text area) |
| 2 | Verify max-length of all fields in Create a new Project Screen | | |
|  | Pre-condition: The user must have logged into the system | 1. Click on Project on the Menu bar |  |
|  | 2. Select Create Project in the Project dropdown list |  |
|  | 3. Check max-length of all fields | The max-length of fields are displayed correctly:  - Project name: 100 chars  - Short Description: 1000 chars  - Long Description: 4000 chars  - Technologies: 4000 chars  - Client Name: 100 chars  - Client Description: 4000 chars |
| 3 | Verify data of Project Type dropdown list | | |
|  | Pre-condition: The user must have logged into the system | 1. Click on Project on the Menu bar |  |
|  | 2. Select Create Project in the Project dropdown list |  |
|  | 3. Check data of Project Type dropdown list | The value of "Project Type" dropdown list is displayed as below:  - Default value: Please select project type  - Valid values:  • Fixed Price  • ODC  • Time & Material |
| 4 | Verify data of Project Status dropdown list | | |
|  | Pre-condition: The user must have logged into the system | 1. Click on Project on the Menu bar |  |
|  | 2. Select Create Project in the Project dropdown list |  |
|  | 3. Check data of Project Status dropdown list | The value of "Project Status" dropdown list is displayed as below:  - Default value: Please select project status  - Valid Values:  • Cancelled  • Closed  • Pending  • Running |
| 5 | Verify data of Location dropdown list | | |
|  | Pre-condition: The user must have logged into the system | 1. Click on Project on the Menu bar |  |
|  | 2. Select Create Project in the Project dropdown list |  |
|  | 3. Check data of Location dropdown list | The value of "Location" dropdown list is displayed as below:  - Default value: Offshore Vietnam Ho Chi Minh city - Valid values: Refer to Reference file |
| 6 | Verify data of Project Manager dropdown list | | |
|  | Pre-condition: The user must have logged into the system | 1. Click on Project on the Menu bar |  |
|  | 2. Select Create Project in the Project dropdown list |  |
|  | 3. Check data of Project Manager dropdown list | The value list of "Project Manager" dropdown list is displayed: Project manager role users |
| 7 | Verify data of Project Manager dropdown list | | |
|  | Pre-condition: The user must have logged into the system | 1. Click on Project on the Menu bar |  |
|  | 2. Select Create Project in the Project dropdown list |  |
|  | 3. Check data of Project Manager dropdown list | The value list of "Project Manager" dropdown list is displayed: Project manager role users |
| 8 | Verify data of Delivery/ Program Manager dropdown list | | |
|  | Pre-condition: The user must have logged into the system | 1. Click on Project on the Menu bar |  |
|  | 2. Select Create Project in the Project dropdown list |  |
|  | 3. Check data of Delivery/ Program Manager dropdown list | The value list of "Project Manager" dropdown list is displayed: Senior manager users |
| 9 | Verify data of Engagement Manager dropdown list | | |
|  | Pre-condition: The user must have logged into the system | 1. Click on Project on the Menu bar |  |
|  | 2. Select Create Project in the Project dropdown list |  |
|  | 3. Check data of Engagement Manager dropdown list | The value list of "Engagement Manager" dropdown list is displayed as below: Senior manager users |
| 10 | Verify data of Client Industry/Sector dropdown list | | |
|  | Pre-condition: The user must have logged into the system | 1. Click on Project on the Menu bar |  |
|  | 2. Select Create Project in the Project dropdown list |  |
|  | 3. Check data of Client Industry/Sector dropdown list | The value of "Client Industry/Sector" dropdown list is displayed as below:  - Default value: Select an Industry/Sector - Valid Values: • Energy • Financial Services • Health Care • Manufacturing • Public Sector • Retail • Technology • Telecommunication • Others |
| 11 | Verify that data in all dropdown lists are not spilled out of the frame | | |
|  | Pre-condition: The user must have logged into the system | 1. Click on Project on the Menu bar |  |
|  | 2. Select Create Project in the Project dropdown list |  |
|  | 3. Check the all dropdown popover | The data in all dropdown lists are not spilled out of the frame |
| 12 | Verify that Project is created successfully when entering valid data into all fields as Admin | | |
|  | Pre-condition: The user must have logged into the system | 1. Click on Project on the Menu bar |  |
|  | 2. Select Create Project in the Project dropdown list |  |
|  | 3. Input valid data into all fields |  |
|  | 4. Click on "Save" button | - The Project is created and following information is added  - The system shows message: "Add project successfully" |
| 13 | Verify that Project is created successfully when entering valid data into all fields as PM | | |
|  | Pre-condition: The user must have logged into the system | 1. Click on Project on the Menu bar |  |
|  | 2. Select Create Project in the Project dropdown list |  |
|  | 3. Input valid data into all fields |  |
|  | 4. Click on "Save" button | - The Project is created and following information is added  - The system shows message: "Add project successfully" |
| 14 | Verify that Project is created successfully when entering valid data into all fields as SM | | |
|  | Pre-condition: The user must have logged into the system | 1. Click on Project on the Menu bar |  |
|  | 2. Select Create Project in the Project dropdown list |  |
|  | 3. Input valid data into all fields |  |
|  | 4. Click on "Save" button | - The Project is created and following information is added  - The system shows message: "Add project successfully" |
| 15 | Verify that Project is created successfully when entering valid data into all mandatory fields | | |
|  | Pre-condition: The user must have logged into the system | 1. Click on Project on the Menu bar |  |
|  | 2. Select Create Project in the Project dropdown list |  |
|  | 3. Input valid data into all mandatory fields |  |
|  | 4. Click on "Save" button | - The Project is created and following information is added  - The system shows message: "Add project successfully" |
| 16 | Verify that Project is created unsuccessfully when missing Project Name | | |
|  | Pre-condition: The user must have logged into the system | 1. Click on Project on the Menu bar |  |
|  | 2. Select Create Project in the Project dropdown list |  |
|  | 3. Input valid data into all fields except "Project Name" |  |
|  | 4. Click on "Save" button | The system shows error message: "This is a required field" |
| 17 | Verify that Project is created unsuccessfully when missing Project Type | | |
|  | Pre-condition: The user must have logged into the system | 1. Click on Project on the Menu bar |  |
|  | 2. Select Create Project in the Project dropdown list |  |
|  | 3. Input valid data into all fields except "Project Type" |  |
|  | 4. Click on "Save" button | The system shows error message: "This is a required field" |
| 18 | Verify that Project is created unsuccessfully when missing Project Status | | |
|  | Pre-condition: The user must have logged into the system | 1. Click on Project on the Menu bar |  |
|  | 2. Select Create Project in the Project dropdown list |  |
|  | 3. Input valid data into all fields except "Project Status" |  |
|  | 4. Click on "Save" button | The system shows error message: "This is a required field" |
| 19 | Verify that Project is created unsuccessfully when missing Start Date | | |
|  | Pre-condition: The user must have logged into the system | 1. Click on Project on the Menu bar |  |
|  | 2. Select Create Project in the Project dropdown list |  |
|  | 3. Input valid data into all fields except "Start Date" |  |
|  | 4. Click on "Save" button | The system shows error message: "This is a required field" |
| 20 | Verify that Project is created unsuccessfully when missing Size (days) | | |
|  | Pre-condition: The user must have logged into the system | 1. Click on Project on the Menu bar |  |
|  | 2. Select Create Project in the Project dropdown list |  |
|  | 3. Input valid data into all fields except "Size (days)" |  |
|  | 4. Click on "Save" button | The system shows error message: "This is a required field" |
| 21 | Verify that Project is created unsuccessfully when missing Project Manager | | |
|  | Pre-condition: The user must have logged into the system | 1. Click on Project on the Menu bar |  |
|  | 2. Select Create Project in the Project dropdown list |  |
|  | 3. Input valid data into all fields except "Project Manager" |  |
|  | 4. Click on "Save" button | The system shows error message: "This is a required field" |
| 22 | Verify that Project is created unsuccessfully when missing Delivery/ Program Manager | | |
|  | Pre-condition: The user must have logged into the system | 1. Click on Project on the Menu bar |  |
|  | 2. Select Create Project in the Project dropdown list |  |
|  | 3. Input valid data into all fields except "Delivery/ Program Manager" |  |
|  | 4. Click on "Save" button | The system shows error message: "This is a required field" |
| 23 | Verify that Project is created unsuccessfully when missing Engagement Manager | | |
|  | Pre-condition: The user must have logged into the system | 1. Click on Project on the Menu bar |  |
|  | 2. Select Create Project in the Project dropdown list |  |
|  | 3. Input valid data into all fields except "Engagement Manager" |  |
|  | 4. Click on "Save" button | The system shows error message: "This is a required field" |
| 24 | Verify that Project is created unsuccessfully when missing Short Description | | |
|  | Pre-condition: The user must have logged into the system | 1. Click on Project on the Menu bar |  |
|  | 2. Select Create Project in the Project dropdown list |  |
|  | 3. Input valid data into all fields except "Short Description" |  |
|  | 4. Click on "Save" button | The system shows error message: "This is a required field" |
| 25 | Verify that Project is created unsuccessfully when missing Long Description | | |
|  | Pre-condition: The user must have logged into the system | 1. Click on Project on the Menu bar |  |
|  | 2. Select Create Project in the Project dropdown list |  |
|  | 3. Input valid data into all fields except "Long Description" |  |
|  | 4. Click on "Save" button | The system shows error message: "This is a required field" |
| 26 | Verify that Project is created unsuccessfully when missing Technologies | | |
|  | Pre-condition: The user must have logged into the system | 1. Click on Project on the Menu bar |  |
|  | 2. Select Create Project in the Project dropdown list |  |
|  | 3. Input valid data into all fields except "Technologies" |  |
|  | 4. Click on "Save" button | The system shows error message: "This is a required field" |
| 27 | Verify that Project is created unsuccessfully when missing Client Name | | |
|  | Pre-condition: The user must have logged into the system | 1. Click on Project on the Menu bar |  |
|  | 2. Select Create Project in the Project dropdown list |  |
|  | 3. Input valid data into all fields except "Client Name" |  |
|  | 4. Click on "Save" button | The system shows error message: "This is a required field" |
| 28 | Verify that Project is created unsuccessfully when Project Name is duplicated | | |
|  | Pre-condition: The user must have logged into the system | 1. Click on Project on the Menu bar |  |
|  | 2. Select Create Project in the Project dropdown list |  |
|  | 3. Input existing "Project Name" |  |
|  | 4. Click on "Save" button | The system shows error message: "Project name has already existed" |
| 29 | Verify that End Date automatically generates when Project ends | | |
|  | Pre-condition: The user must have logged into the system | 1. Click on Project on the Menu bar |  |
|  | 2. Select Create Project in the Project dropdown list |  |
|  | 3. Select "Closed" in Project Status dropdown list | End Date automatically generates |
| 30 | Verify that clicking on "X" button | | |
|  | Pre-condition: The user must have logged into the system | 1. Click on Project on the Menu bar |  |
|  | 2. Select Create Project in the Project dropdown list |  |
|  | 3. Input valid data into all fields |  |
|  | 4. Click on "Cancel" button | - The system doesn't save project information - The system closes the pop-up and returns to the previous screen |
| 31 | Verify that clicking on "Cancel" button | | |
|  | Pre-condition: The user must have logged into the system | 1. Click on Project on the Menu bar |  |
|  | 2. Select Create Project in the Project dropdown list |  |
|  | 3. Input valid data into all fields |  |
|  | 4. Click on "Cancel" button | - The system doesn't save project information - The system closes the pop-up and returns to the previous screen |

Table 5.1: Test Cases of Create Project

## **5.2. Edit Project**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Step** | **Expected Output** |
| 1 | Verify max-length of all fields in Update Project Information Screen | | |
|  |  | 1. Go to Project Information Screen |  |
|  | 2. Click on "Edit" button |  |
|  | 3. Check max-length of all fields | The max-length of fields are displayed correctly:  - Project name: 100 chars  - Short Description: 1000 chars  - Long Description: 4000 chars  - Technologies: 4000 chars  - Client Name: 100 chars  - Long Description: 4000 chars  - Client Description: 4000 chars |
| 2 | Verify data of Project Status dropdown list | | |
|  |  | 1. Go to Project Information Screen |  |
|  | 2. Click on "Edit" button |  |
|  | 3. Check data of Project Status dropdown list | The value of "Project Status" dropdown list is displayed as below:  - Default value: Please select project status - Valid Values: • Cancelled • Closed • Pending • Running |
| 3 | Verify data of Project Type dropdown list | | |
|  |  | 1. Go to Project Information Screen |  |
|  | 2. Click on "Edit" button |  |
|  | 3. Check data of Project Type dropdown list | The value of "Project Type" dropdown list is displayed as below:  - Default value: Please select project type  - Valid values:  • Fixed Price  • ODC  • Time & Material |
| 4 | Verify data of Location dropdown list | | |
|  |  | 1. Go to Project Information Screen |  |
|  | 2. Click on "Edit" button |  |
|  | 3. Check data of Location dropdown list | The value of "Location" dropdown list is displayed as below:  - Default value: Offshore Vietnam Ho Chi Minh city - Valid values: Refer to Reference file |
| 5 | Verify data of Project Manager dropdown list | | |
|  |  | 1. Go to Project Information Screen |  |
|  | 2. Click on "Edit" button |  |
|  | 3. Check data of Project Manager dropdown list | The value list of "Project Manager" dropdown list is displayed: Project manager role users |
| 6 | Verify data of Project Manager dropdown list | | |
|  |  | 1. Go to Project Information Screen |  |
|  | 2. Click on "Edit" button |  |
|  | 3. Check data of Project Manager dropdown list | The value list of "Project Manager" dropdown list is displayed: Project manager role users |
| 7 | Verify data of Delivery/ Program Manager dropdown list | | |
|  |  | 1. Go to Project Information Screen |  |
|  | 2. Click on "Edit" button |  |
|  | 3. Check data of Delivery/ Program Manager dropdown list | The value list of "Project Manager" dropdown list is displayed: Senior manager users |
| 8 | Verify data of Engagement Manager dropdown list | | |
|  |  | 1. Go to Project Information Screen |  |
|  | 2. Click on "Edit" button |  |
|  | 3. Check data of Engagement Manager dropdown list | The value list of "Engagement Manager" dropdown list is displayed as below: Senior manager users |
| 9 | Verify data of Client Industry/Sector dropdown list | | |
|  |  | 1. Go to Project Information Screen |  |
|  | 2. Click on "Edit" button |  |
|  | 3. Check data of Client Industry/Sector dropdown list | The value of "Client Industry/Sector" dropdown list is displayed as below:  - Default value: Select an Industry/Sector - Valid Values: • Energy • Financial Services • Health Care • Manufacturing • Public Sector • Retail • Technology • Telecommunication • Others |
| 10 | Verify that data in all dropdown lists are not spilled out of the frame | | |
|  | Pre-condition: The user must have logged into the system | 1. Go to Project Information Screen |  |
|  | 2. Click on "Edit" button |  |
|  | 3. Check the all dropdown popover | The data in all dropdown lists are not spilled out of the frame |
| 11 | Verify that Project Information is edited successfully when changing data of all fields as Admin | | |
|  | Pre-condition: The user must have logged into the system by Admin account | 1. Go to Project Information Screen |  |
|  | 2. Click on "Edit" button |  |
|  | 3. Input valid data into all fields |  |
|  | 4. Click on "Save" button | - The system saves the updated information and the screen switches to read mode  - The system shows message: "Edit project successfully" |
| 12 | Verify that Project Information is edited successfully when changing data of all fields as PM | | |
|  | Pre-condition: The user must have logged into the system by PM account | 1. Go to Project Information Screen |  |
|  | 2. Click on "Edit" button |  |
|  | 3. Input valid data into all fields |  |
|  | 4. Click on "Save" button | - The system saves the updated information and the screen switches to read mode  - The system shows message: "Edit project successfully" |
| 13 | Verify that Project Information is edited successfully when changing data of all fields as SM | | |
|  | Pre-condition: The user must have logged into the system by SM account | 1. Go to Project Information Screen |  |
|  | 2. Click on "Edit" button |  |
|  | 3. Input valid data into all fields |  |
|  | 4. Click on "Save" button | - The system saves the updated information and the screen switches to read mode  - The system shows message: "Edit project successfully" |
| 14 | Verify that Project Information is edited successfully when changing data of all mandatory fields | | |
|  | Pre-condition: The user must have logged into the system | 1. Go to Project Information Screen |  |
|  | 2. Click on "Edit" button |  |
|  | 3. Input valid data into all mandatory fields |  |
|  | 4. Click on "Save" button | - The system saves the updated information and the screen switches to read mode  - The system shows message: "Edit project successfully" |
| 15 | Verify that Project is edited unsuccessfully when missing Project Name | | |
|  | Pre-condition: The user must have logged into the system | 1. Go to Project Information Screen |  |
|  | 2. Click on "Edit" button |  |
|  | 3. Input valid data into all fields except "Project Name" |  |
|  | 4. Click on "Save" button | The system shows error message: "This is a required field" |
| 16 | Verify that Project is edited unsuccessfully when missing Project Type | | |
|  | Pre-condition: The user must have logged into the system | 1. Go to Project Information Screen |  |
|  | 2. Click on "Edit" button |  |
|  | 3. Input valid data into all fields except "Project Type" |  |
|  | 4. Click on "Save" button | The system shows error message: "This is a required field" |
| 17 | Verify that Project is edited unsuccessfully when missing Project Status | | |
|  | Pre-condition: The user must have logged into the system | 1. Go to Project Information Screen |  |
|  | 2. Click on "Edit" button |  |
|  | 3. Input valid data into all fields except "Project Status" |  |
|  | 4. Click on "Save" button | The system shows error message: "This is a required field" |
| 18 | Verify that Project is edited unsuccessfully when missing Start Date | | |
|  | Pre-condition: The user must have logged into the system | 1. Go to Project Information Screen |  |
|  | 2. Click on "Edit" button |  |
|  | 3. Input valid data into all fields except "Start Date" |  |
|  | 4. Click on "Save" button | The system shows error message: "This is a required field" |
| 19 | Verify that Project is edited unsuccessfully when missing Size (days) | | |
|  | Pre-condition: The user must have logged into the system | 1. Go to Project Information Screen |  |
|  | 2. Click on "Edit" button |  |
|  | 3. Input valid data into all fields except "Size (days)" |  |
|  | 4. Click on "Save" button | The system shows error message: "This is a required field" |
| 20 | Verify that Project is edited unsuccessfully when missing Project Manager | | |
|  | Pre-condition: The user must have logged into the system | 1. Go to Project Information Screen |  |
|  | 2. Click on "Edit" button |  |
|  | 3. Input valid data into all fields except "Project Manager" |  |
|  | 4. Click on "Save" button | The system shows error message: "This is a required field" |
| 21 | Verify that Project is edited unsuccessfully when missing Delivery/ Program Manager | | |
|  | Pre-condition: The user must have logged into the system | 1. Go to Project Information Screen |  |
|  | 2. Click on "Edit" button |  |
|  | 3. Input valid data into all fields except "Delivery/ Program Manager" |  |
|  | 4. Click on "Save" button | The system shows error message: "This is a required field" |
| 22 | Verify that Project is edited unsuccessfully when missing Engagement Manager | | |
|  | Pre-condition: The user must have logged into the system | 1. Go to Project Information Screen |  |
|  | 2. Click on "Edit" button |  |
|  | 3. Input valid data into all fields except "Engagement Manager" |  |
|  | 4. Click on "Save" button | The system shows error message: "This is a required field" |
| 23 | Verify that Project is edited unsuccessfully when missing Short Description | | |
|  | Pre-condition: The user must have logged into the system | 1. Go to Project Information Screen |  |
|  | 2. Click on "Edit" button |  |
|  | 3. Input valid data into all fields except "Short Description" |  |
|  | 4. Click on "Save" button | The system shows error message: "This is a required field" |
| 24 | Verify that Project is edited unsuccessfully when missing Long Description | | |
|  | Pre-condition: The user must have logged into the system | 1. Go to Project Information Screen |  |
|  | 2. Click on "Edit" button |  |
|  | 3. Input valid data into all fields except "Long Description" |  |
|  | 4. Click on "Save" button | The system shows error message: "This is a required field" |
| 25 | Verify that Project is edited unsuccessfully when missing Technologies | | |
|  | Pre-condition: The user must have logged into the system | 1. Go to Project Information Screen |  |
|  | 2. Click on "Edit" button |  |
|  | 3. Input valid data into all fields except "Technologies" |  |
|  | 4. Click on "Save" button | The system shows error message: "This is a required field" |
| 26 | Verify that Project is edited unsuccessfully when missing Client Name | | |
|  | Pre-condition: The user must have logged into the system | 1. Go to Project Information Screen |  |
|  | 2. Click on "Edit" button |  |
|  | 3. Input valid data into all fields except "Client Name" |  |
|  | 4. Click on "Save" button | The system shows error message: "This is a required field" |
| 27 | Verify that Project is edited unsuccessfully when Project Name is duplicated | | |
|  | Pre-condition: The user must have logged into the system | 1. Go to Project Information Screen |  |
|  | 2. Click on "Edit" button |  |
|  | 3. Input existing "Project Name" |  |
|  | 4. Click on "Save" button | The system shows error message: "Project name has already existed" |
| 28 | Verify that the system is unable to switch between Project Roles / Project Members / Project Case Studies tab | | |
|  | Pre-condition: The user must have logged into the system | 1. Go to Project Information Screen |  |
|  | 2. Click on "Edit" button | The system is unable to switch between Project Roles / Project Members / Project Case Studies tab |
| 29 | Verify that clicking on "X" button | | |
|  | Pre-condition: The user must have logged into the system | 1. Go to Project Information Screen |  |
|  | 2. Click on "Edit" button |  |
|  | 3. Input valid data into all fields |  |
|  | 4. Click on "Cancel" button | - The system does not save the changing information and the screen switches to read mode  - The system closes the pop-up and returns to the previous screen |
| 30 | Verify that clicking on "Cancel" button | | |
|  | Pre-condition: The user must have logged into the system | 1. Go to Project Information Screen |  |
|  | 2. Click on "Edit" button |  |
|  | 3. Input valid data into all fields |  |
|  | 4. Click on "Cancel" button | - The system does not save the changing information and the screen switches to read mode  - The system closes the pop-up and returns to the previous screen |

Table 5.2: Test Cases of Edit Project

## **5.3. View Project**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Step** | **Expected Output** |
| 1 | Verify that UI of Project Information Screen | | |
|  | Pre-condition: The user must have logged into the system | 1. Go to Project Information Screen |  |
|  | 2. Check UI of Project Information Screen | The system shows the Project Information Screen contains the follows:  - Project Name  - Project Type  - Project Status  - Start Date  - End Date  - Size (days)  - Location  - Project Manager  - Delivery/ Program Manager  - Engagement Manager  - Short Description  - Long Description  - Technologies  - Client Name  - Client Industry/Sector  - Client Description |
| 2 | Verify that Project Information is displayed successfully when clicking on Project Name in the search results | | |
|  |  | 1. Click on Project Name in the search results | The system retrieves data and shows the Project Information screen |
| 3 | Verify that Project Information is displayed successfully when clicking on “Project Information” tab | | |
|  | Pre-condition: The user has opened a Project | 1. Click on “Project Information” tab | The system retrieves data and shows the Project Information screen |
| 4 | Verify that Project Information is displayed successfully as Admin | | |
|  | Pre-condition: The user must have logged into the system by Admin account | 1. Click on Project Name regarding to Admin role in the search results | - The system allows Admin to view all Projects  - The system shows Project Information |
| 5 | Verify that Project Information is displayed successfully as PM | | |
|  | Pre-condition: The user must have logged into the system by PM account | 1. Click on Project Name regarding to PM role in the search results | - The system allows PM to view Projects regarding to PM role  - The system shows Project Information |
| 6 | Verify that Project Information is displayed successfully as SM | | |
|  | Pre-condition: The user must have logged into the system by SM account | 1. Click on Project Name regarding to PM role in the search results | - The system allows PM to view Projects regarding to PM role  - The system shows Project Information |
| 7 | Verify that all fields are unable to edit when viewing Project Information | | |
|  |  | 1. Go to Project Information Screen |  |
|  | 2. Try to edit data all fields | The system prevents user to edit all fields |

Table 5.3: Test Cases of View Project

## **5.4. Delete Project**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Step** | **Expected Output** |
| 1 | Verify that the confirmation message appears when no Case Study and no Assignment | | |
|  | Pre-condition: The user must have logged into the system | 1. Go to Project Information Screen |  |
|  | 2. Click on "Delete" button | The system is displayed the confirmation message: "Are you sure you want to delete <Project Name>?" with 2 options: "Yes", "No" |
| 2 | Verify that Project is deleted successfully when selecting Project has no Case Study and no Assignment with clicking "Yes" in message | | |
|  | Pre-condition: The user must have logged into the system | 1. Go to Project Information Screen |  |
|  | 2. Click on "Delete" button |  |
|  | 3. Click on "Yes" in message | - Project is deleted successfully  - Message is closed  - The screen returns to search project screen  - The system is displayed the message on the top: "The project was deleted" |
| 3 | Verify that Project is deleted unsuccessfully when selecting Project has no Case Study and no Assignment with clicking "No" in message | | |
|  | Pre-condition: The user must have logged into the system | 1. Go to Project Information Screen |  |
|  | 2. Click on "Delete" button |  |
|  | 3. Click on "No" in message | - Project is deleted unsuccessfully  - Message is closed  - The screen returns to search project screen |
| 4 | Verify that Project is deleted unsuccessfully when selecting Project has Case Study = Approved and no member’s Responsibility | | |
|  | Pre-condition: The user must have logged into the system | 1. Go to Project Information Screen |  |
|  | 2. Click on "Delete" button |  |
|  | 3. Click on "Yes" in message | The system is displayed the message: "This project has members / case studies approved. You are not allowed to delete this project" |
|  | 4. Click on "OK" to close message | - Message is closed  - Project is deleted unsuccessfully |
| 5 | Verify that Project is deleted unsuccessfully when selecting Project has member’s Responsibility = Approved and no Case Study | | |
|  | Pre-condition: The user must have logged into the system | 1. Go to Project Information Screen |  |
|  | 2. Click on "Delete" button |  |
|  | 3. Click on "Yes" in message | The system is displayed the message: "This project has members / case studies approved. You are not allowed to delete this project" |
|  | 4. Click on "OK" to close message | - Message is closed  - Project is deleted unsuccessfully |
| 6 | Verify that Project is deleted unsuccessfully when selecting Project has Case Study = Waiting for approval and no member’s Responsibility | | |
|  | Pre-condition: The user must have logged into the system | 1. Go to Project Information Screen |  |
|  | 2. Click on "Delete" button |  |
|  | 3. Click on "Yes" in message | The system is displayed the message: "This project has members’ responsibilities / case studies that are waiting for approval. If you want to delete this project, these members or case studies must be rejected first." |
|  | 4. Click on "OK" to close message | - Message is closed  - Project is deleted unsuccessfully |
| 7 | Verify that Project is deleted unsuccessfully when selecting Project has member’s Responsibility = Waiting for approval and no Case Study | | |
|  | Pre-condition: The user must have logged into the system | 1. Go to Project Information Screen |  |
|  | 2. Click on "Delete" button |  |
|  | 3. Click on "Yes" in message | The system is displayed the message: "This project has members’ responsibilities / case studies that are waiting for approval. If you want to delete this project, these members or case studies must be rejected first." |
|  | 4. Click on "OK" to close message | - Message is closed  - Project is deleted unsuccessfully |
| 8 | Verify that Project is deleted unsuccessfully when selecting Project has Case Study = Approved and no Assignment | | |
|  | Pre-condition: The user must have logged into the system | 1. Go to Project Information Screen |  |
|  | 2. Click on "Delete" button |  |
|  | 3. Click on "Yes" in message | The system is displayed the message: "This project has members / case studies approved. You are not allowed to delete this project" |
|  | 4. Click on "OK" to close message | - Message is closed  - Project is deleted unsuccessfully |
| 9 | Verify that Project is deleted unsuccessfully when selecting Project has Case Study = Waiting for approval and no Assignment | | |
|  | Pre-condition: The user must have logged into the system | 1. Go to Project Information Screen |  |
|  | 2. Click on "Delete" button |  |
|  | 3. Click on "Yes" in message | The system is displayed the message: "This project has members’ responsibilities / case studies that are waiting for approval. If you want to delete this project, these members or case studies must be rejected first." |
|  | 4. Click on "OK" to close message | - Message is closed  - Project is deleted unsuccessfully |
| 10 | Verify that Project is deleted unsuccessfully when selecting Project has Case Study = Approved and member’s Responsibility = Waiting for approval | | |
|  | Pre-condition: The user must have logged into the system | 1. Go to Project Information Screen |  |
|  | 2. Click on "Delete" button |  |
|  | 3. Click on "Yes" in message | The system is displayed the message: "This project has members / case studies approved. You are not allowed to delete this project" |
|  | 4. Click on "OK" to close message | - Message is closed  - Project is deleted unsuccessfully |
| 11 | Verify that Project is deleted unsuccessfully when selecting Project has Case Study = Waiting for approval and member’s Responsibility = Approved | | |
|  | Pre-condition: The user must have logged into the system by SM account | 1. Go to Project Information Screen |  |
|  | 2. Click on "Delete" button |  |
|  | 3. Click on "Yes" in message | The system is displayed the message: "This project has members’ responsibilities / case studies that are waiting for approval. If you want to delete this project, these members or case studies must be rejected first." |
|  | 4. Click on "OK" to close message | - Message is closed  - Project is deleted unsuccessfully |

Table 5.4: Test Cases of Delete Project

# **APPLICATION ROBOT FRAMEWORK**

## **6.1. Install Robot FrameWork**

**Install Python**

Python 3 Installed: https://www.python.org/downloads/

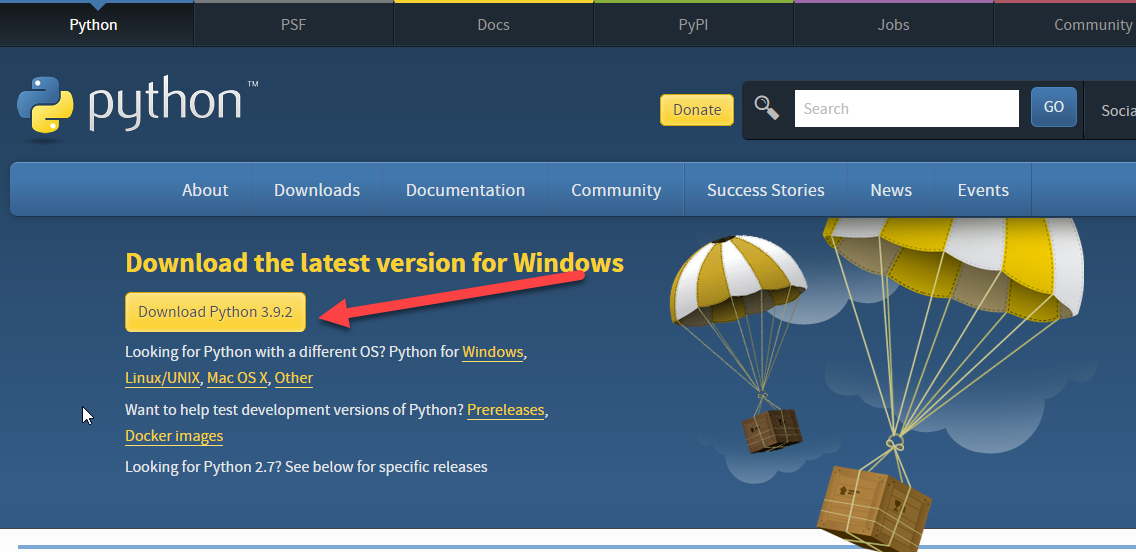


Figure 6.1: Download Python

**Install Robot Framework and Selenium**

Open cmd and enter with the following:

* Install Python and check version
  + python –version
* Install PIP and check version
  + python get-pip.py
  + pip –version
* Install Selenium Library:
  + pip install --upgrade robotframework-seleniumlibrary

**Install RIDE IDE**

* Required Python > 3.0
* Install RIDE
  + pip install psutil
  + pip install -U --pre robotframework-ride
* Start RIDE from command line
  + ride.py
* Set Environment variable geckodriver if got error message.

**WebDriverManager**

[WebdriverManager](https://github.com/omenia/webdrivermanager) which can find the latest version or when required, any version of appropriate webdrivers for you and then download and link/copy it into right location. Tool can run on all major operating systems and supports downloading of Chrome, Firefox, Opera & Edge webdrivers.

* pip install webdrivermanager
* webdrivermanager firefox chrome --linkpath <path>

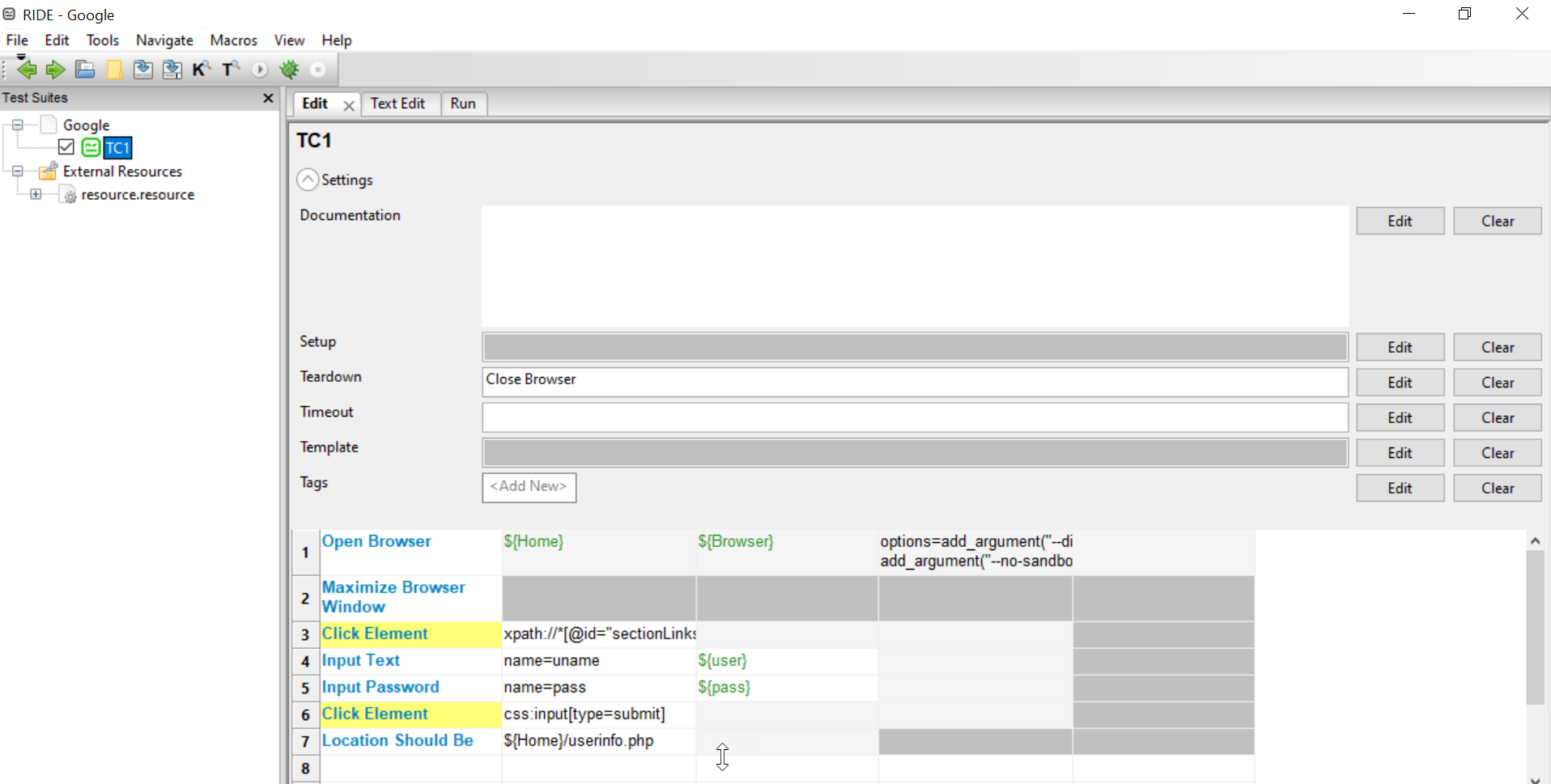


Figure 6.2: Screen of RIDE

## **6.2. Components of RIDE IDE**

### **6.2.1. Setting**

The Settings section will define the initial settings for the script, such as describing what it will do, and which libraries to use.

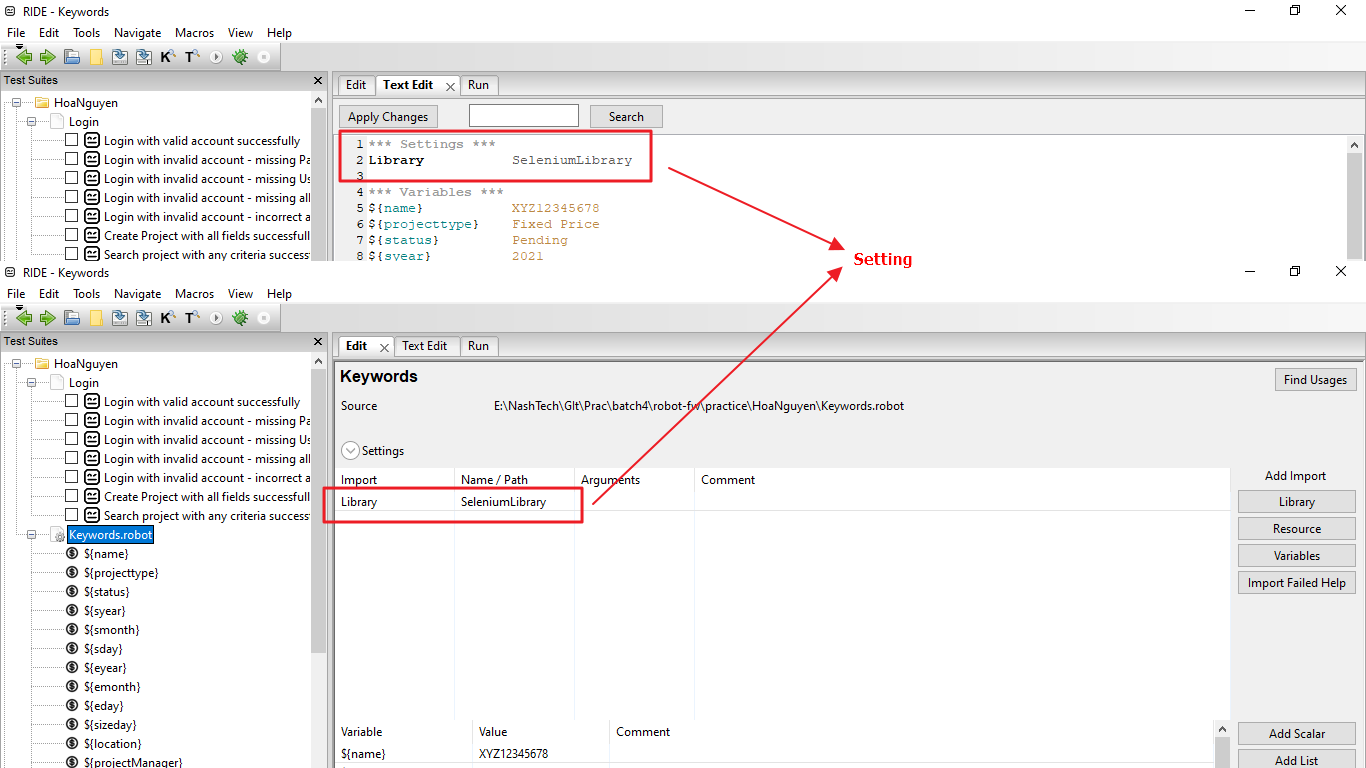


Figure 6.3: Setting of RIDE

### **6.2.2. Test Cases**

The Test Cases section is the main part that includes the test cases, in this part we just need to call the Keywords to make them run and check if the Output is correct with Expected.

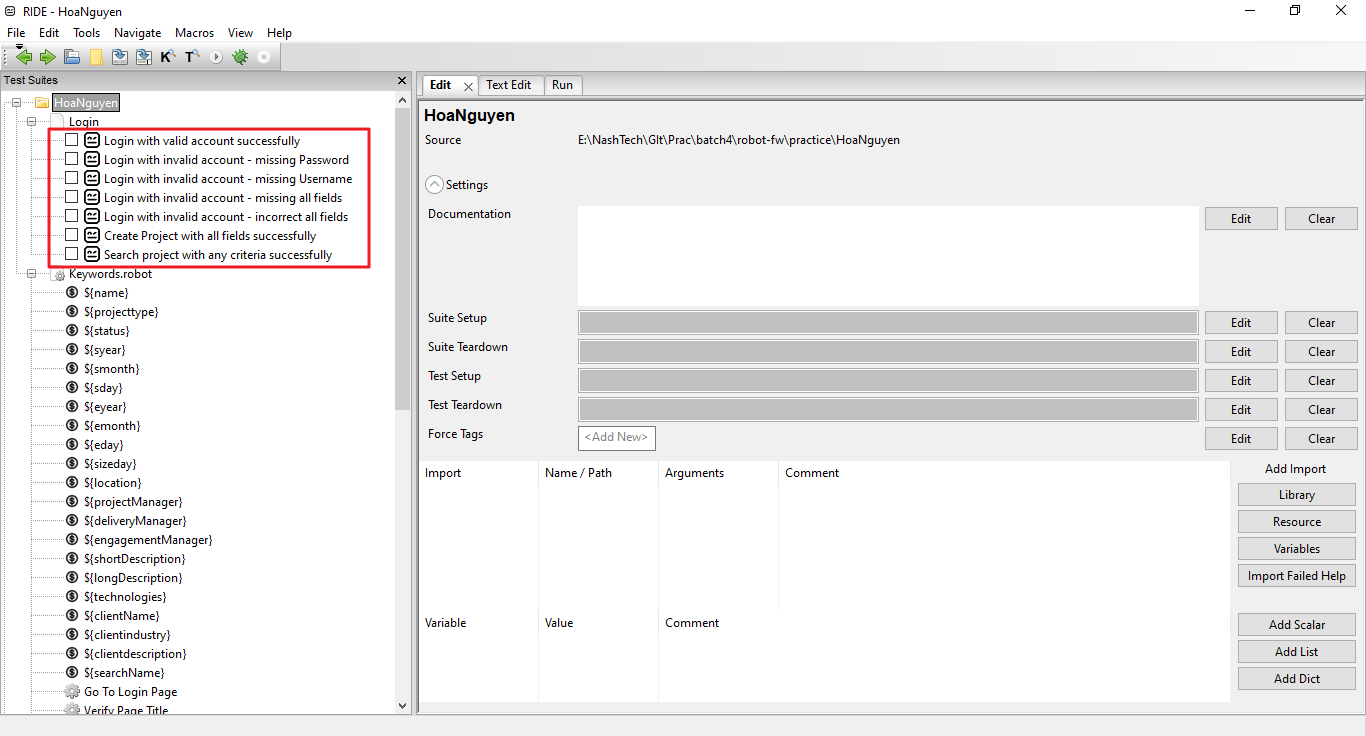


Figure 6.4: Test Case of RIDE

### **6.2.3. Keywords**

Keywords are the steps from preparation to checking the results, Robot is the world of keywords. In keywords we also call other keywords to perform the processing.

The Keywords themselves can receive arguments to use in similar cases, such as Input Text, Submit Form,… In addition, keywords are completely freely defined in any language. In any language, just the Test Cases section is called that way.

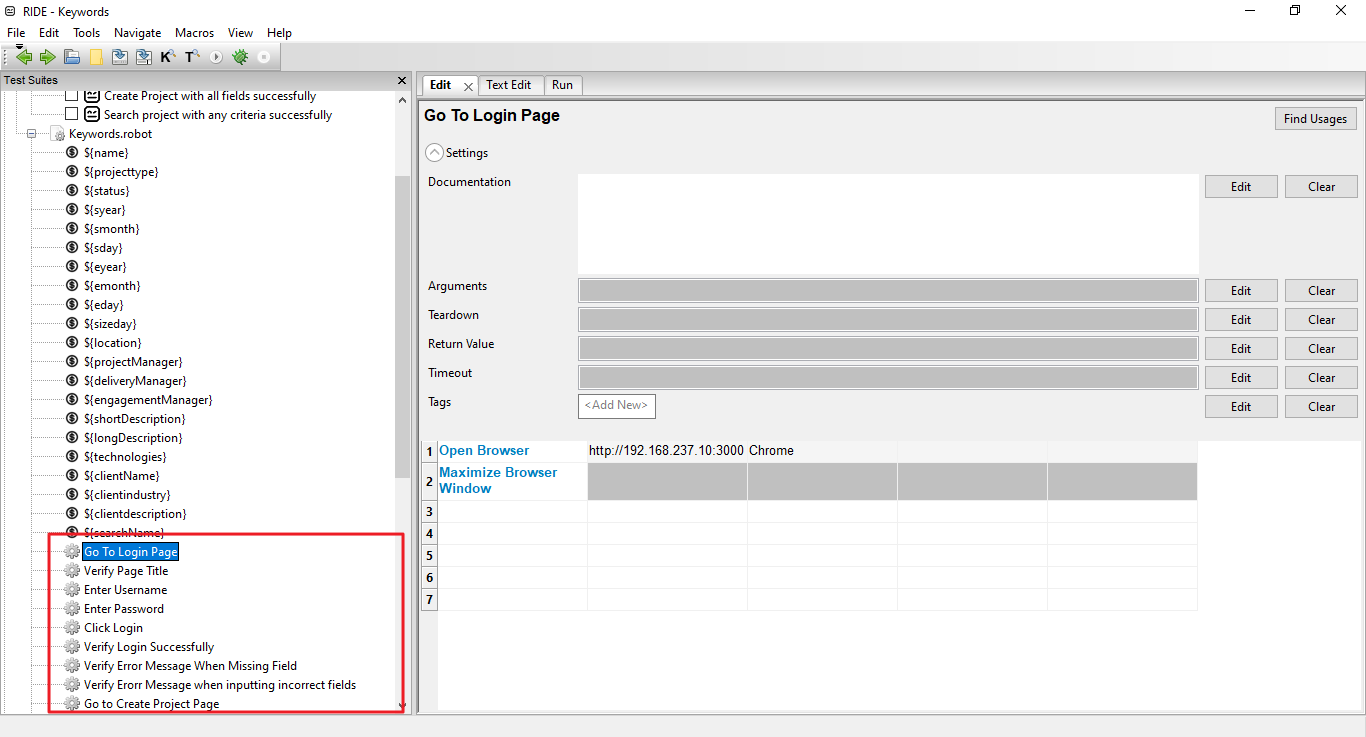


Figure 6.5: Keywords of RIDE

### **6.2.4.** **Variables**

Integrating variables makes it much easier to code and maintain tests. Most suitable for Data-driven Test.

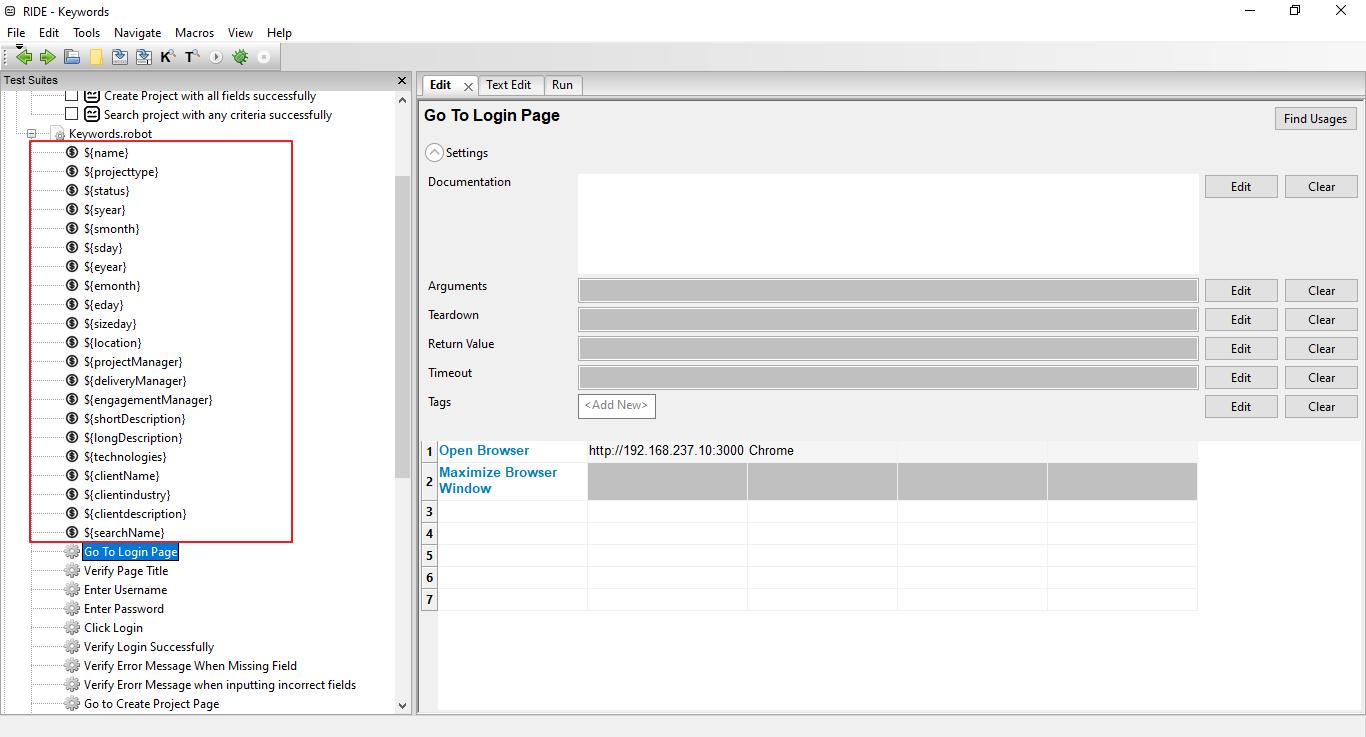


Figure 6.6: Variables of RIDE

## **6.3. Create test project**

### **6.3.1. Create test demo**

* Function: Create Project functional
* Test Case: Verify that Project is created successfully when entering valid data into all fields as Admin (*Test case ID: 12 - Table 6.1: Test Cases of Create Project*)

**Step 1**: Import Library: Click Library / Fill in Name of Library / Click on OK

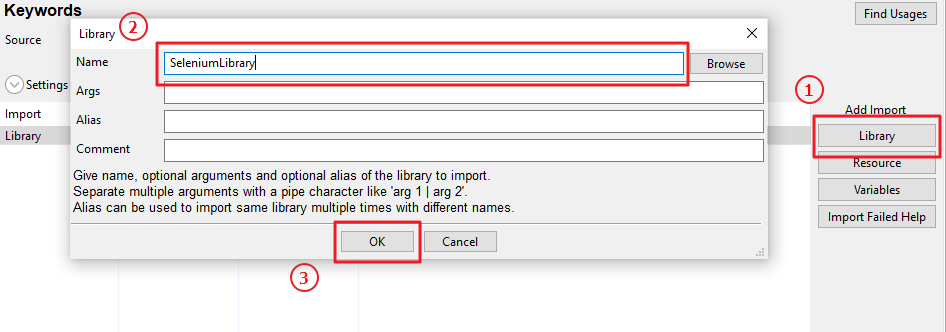


Figure 6.7: Steps to import library

**Step 2:** Create Variables: Click on Add Scalar / Fill in Name of Scalar Variable and value for it / Click on OK

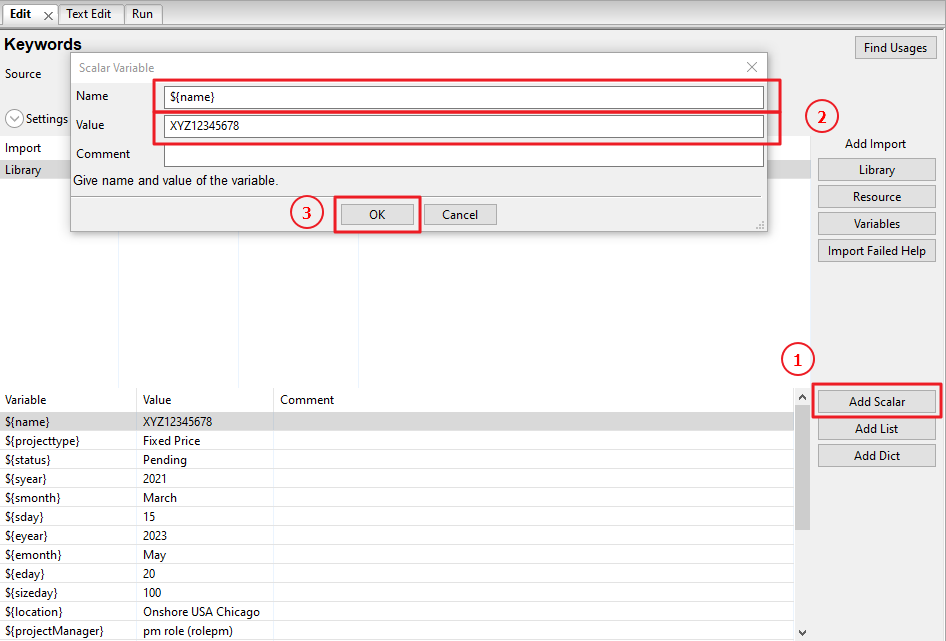


Figure 6.8: Steps to create variables

Variables is used in this functional:

|  |  |  |
| --- | --- | --- |
| **Variable** | **Value** | **Description** |
| ${name} | XYZ12345678 | Name of project |
| ${projecttype} | Fixed Price | Project Type |
| ${status} | Pending | Status of project |
| ${syear} | 2021 | Start year of project |
| ${smonth} | March | Start month of project |
| ${sday} | 15 | Start day of project |
| ${eyear} | 2023 | End year of project |
| ${emonth} | May | End month of project |
| ${eday} | 20 | End day of project |
| ${sizeday} | 100 | Size day |
| ${location} | Onshore USA Chicago | Location of project |
| ${projectManager} | pm role (rolepm) | Project Manager |
| ${deliveryManager} | Nguyen Sang (smsang) | Delivery Manager |
| ${engagementManager} | Hue Nguyen (huesm) | Engagement Manager |
| ${shortDescription} | ABC | Short Description |
| ${longDescription} | ABC | Long Description |
| ${technologies} | ABC | Technologies |
| ${clientName} | ABC | Client Name |
| ${clientindustry} | Health Care | Client Industry |
| ${clientdescription} | ABC | Client Description |
| ${searchName} | ABC | Search Name |

Table 6.2: Variables

**Step 3:** Create Keywords

* Keyword - Go to Create Project Page: is created to navigate to 'Create Project' Page after login. (Click on Project / Create Project)

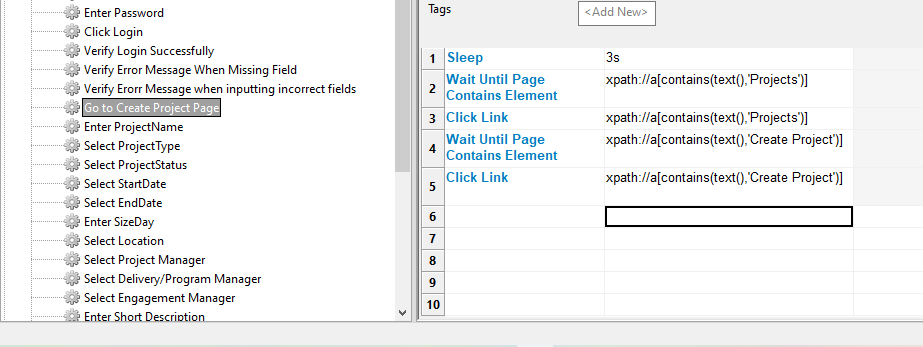


Figure 6.9: Keywords - Go to Create Project Page

* Sleep – 3s: Pauses the test executed for the given time.
* Wait Until Page Contains Element - xpath://a[contains(text(),'Projects')]: Waits until the element locator contains text ‘Project’ appears on the current page. In this keyword, using locator is Xpath.
* Click Link - xpath://a[contains(text(),'Projects')]: Clicks a link identified by locator. Locator is Xpath, where have text ‘Project’.
* Wait Until Page Contains Element - xpath://a[contains(text(),'Create Projects')]: Waits until the element locator contains text ‘Create Project’ appears on the current page. In this keyword, using locator is Xpath. Because it is dropdowlist, this step is performed again and is different from the step above.
* Click Link - xpath://a[contains(text(),'Create Projects')]: Clicks a link identified by locator. Locator is Xpath. The step is repeat and contains text ‘Create Project’.
* Keyword – Enter Project Name: is created to input value into ‘Project Name’ textbox.

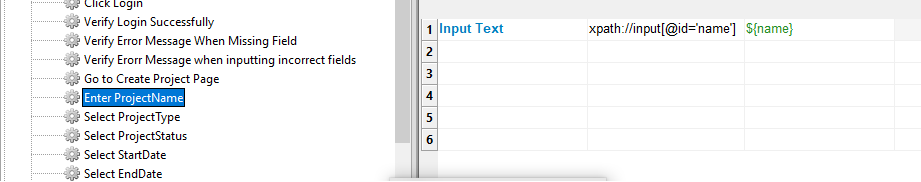


Figure 6.10: Keyword – Enter Project Name

* Input Text - xpath://input[@id='name']: Types the given text into the text field identified by locator. Besides, we use ‘${name}’ variable to facilitate data change.

The remaining keywords such as Enter Sizeday, Enter Short Description, Enter Long Description, Enter Technologies, Enter ClientName, Select Client Industry, Enter Client Description, Enter ProjectNameSearch will be performed as Enter Project Name. It it will be changed locator and position corresponding to keyword.

* Keyword – Select ProjectType: apply for fields which is Dropdown list. In this situation, dropdown list is ‘Project Type’.

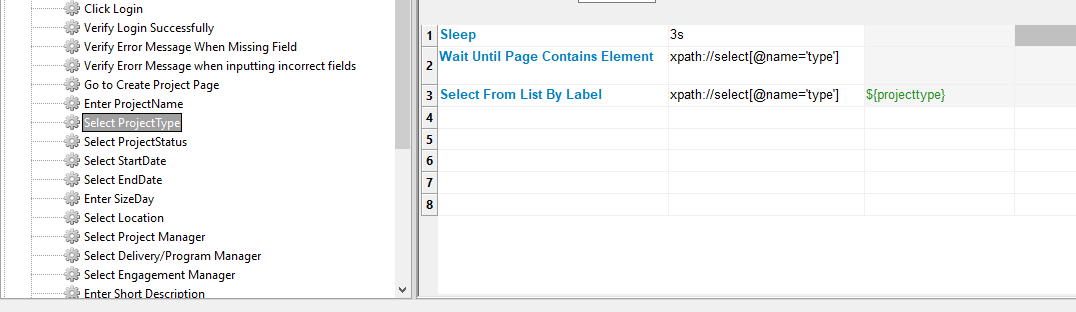


Figure 6.11: Keyword – Select ProjectType

* Sleep – 3s: Pauses the test executed for the given time. Because it is dropdown list, it will take a long time to load data. Therefore, it is added to help the page load the full data.
* Wait Until Page Contains Element - xpath://select[@name='type']: Waits until the element locator having ‘Type’ name and it appears on the current page. In this keyword, using locator is Xpath.
* Select From List By Label - xpath://select[@name='type']: Selects options from selection list locator by labels. This step identify value for ‘Project Type’dropdown list.

The remaining keywords such as Select From List By Label, Select StartDate, Select EndDate, Select Location, Select Project Manager, Select Delivery/Program Manager, Select Engagement Manager, Select Client Industry will be performed as Select ProjectType. It it will be changed locator and position corresponding to keyword.

* Keyword - Click Create Project: After inputting data into all fields, click the button to perform create a new Project.

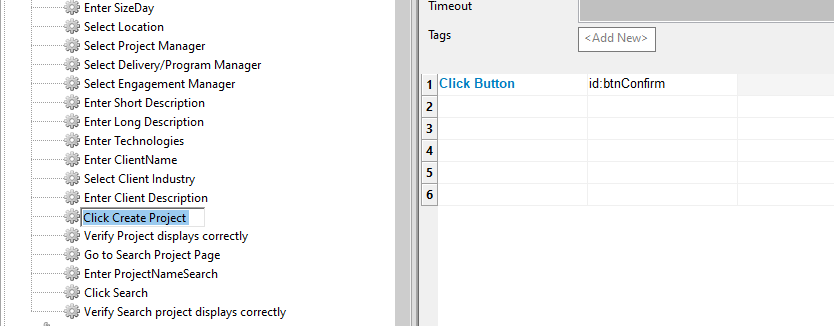


Figure 6.12: Keyword - Click Create Project

* Click Button – id: btnConfirm: Clicks the button identified by locator.
* Keyword – Verify Project displays correctly: after creating a new project successfully, this helps us check that the displayed data matches the entered data.

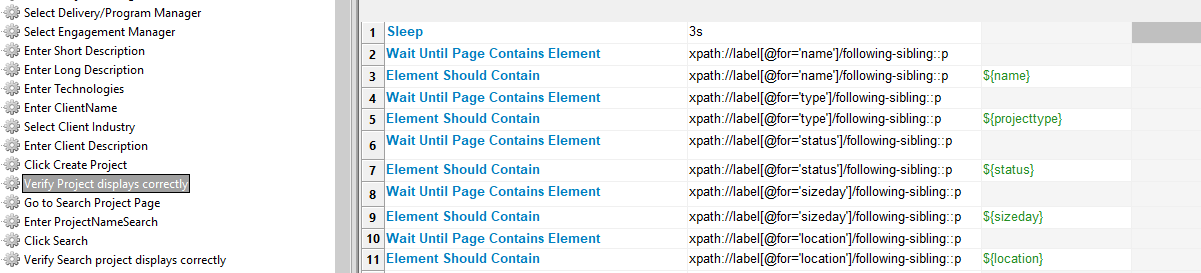


Figure 6.13: Keyword – Verify Project displays correctly - Part 1

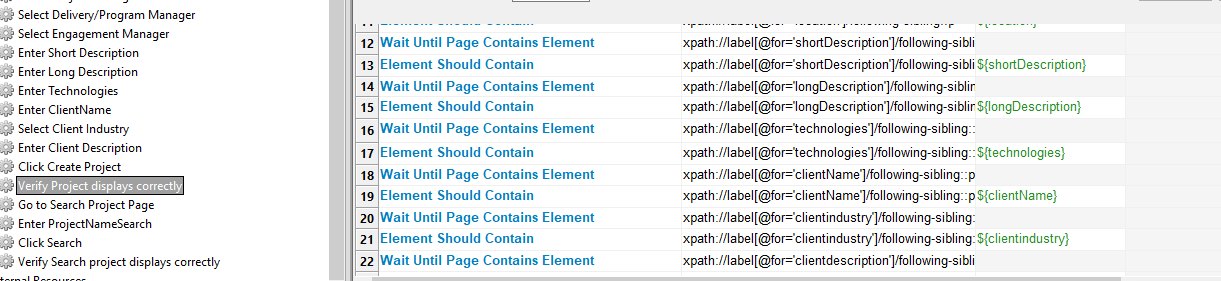


Figure 6.14: Keyword – Verify Project displays correctly - Part 2

**Step 4:** Create testcase - Create Project with all fields successfully

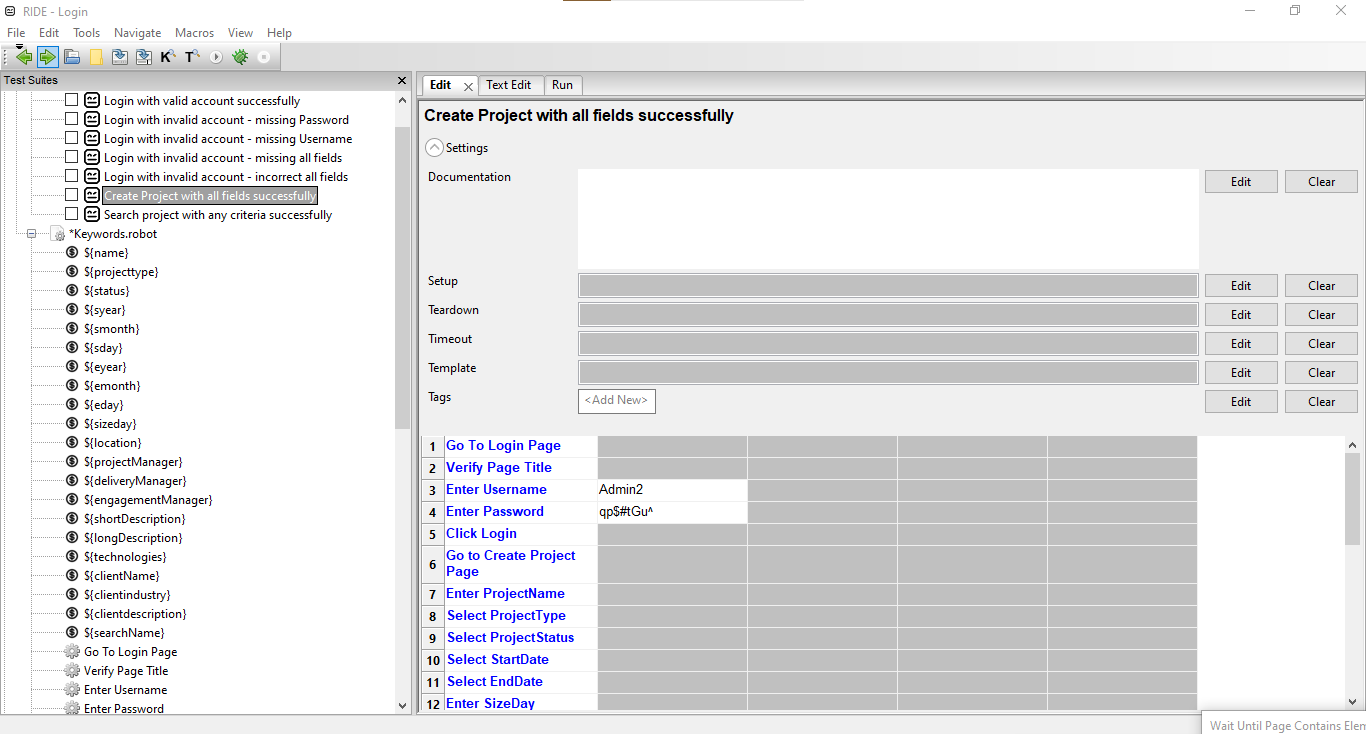


Figure 6.15: Test Case - Create Project with all fields successfully

By listing the keywords that have been added below to proceed to create a complete test case.

* Go To Login Page
* Verify Page Title
* Enter Username
* Enter Password
* Go to Create Project Page
* Enter ProjectName
* Select From List By Label
* Select StartDate
* Select EndDate
* Enter SizeDay
* Select Location
* Select Project Manager
* Select Delivery/Program Manager
* Select Engagement Manager
* Enter Short Description
* Enter Long Description
* Enter Technologies
* Enter ClientName
* Select Client Industry
* Enter Client Description
* Click Create Project
* Verify Search project displays correctly

**Step 5:** Run Test Case by clicking ‘Run Test with Debug’ or press F9.

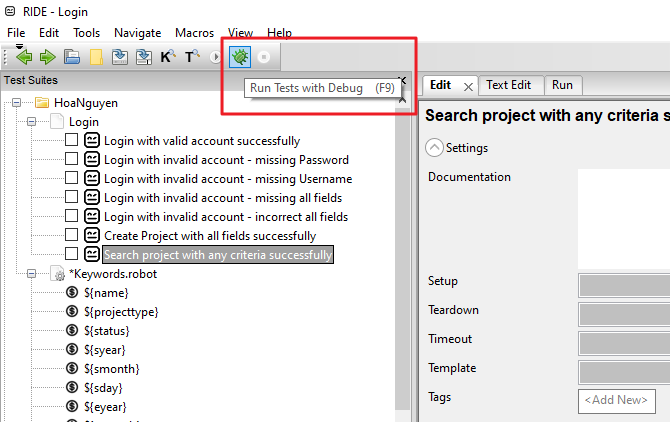


Figure 6.16: Button to Run Test

### **6.3.2. Test Result**

* Test Case: Verify that Project is created successfully when entering valid data into all fields as Admin (*Test case ID: 12 - Table 6.3: Test Cases of Create Project*)
* Result: Passed

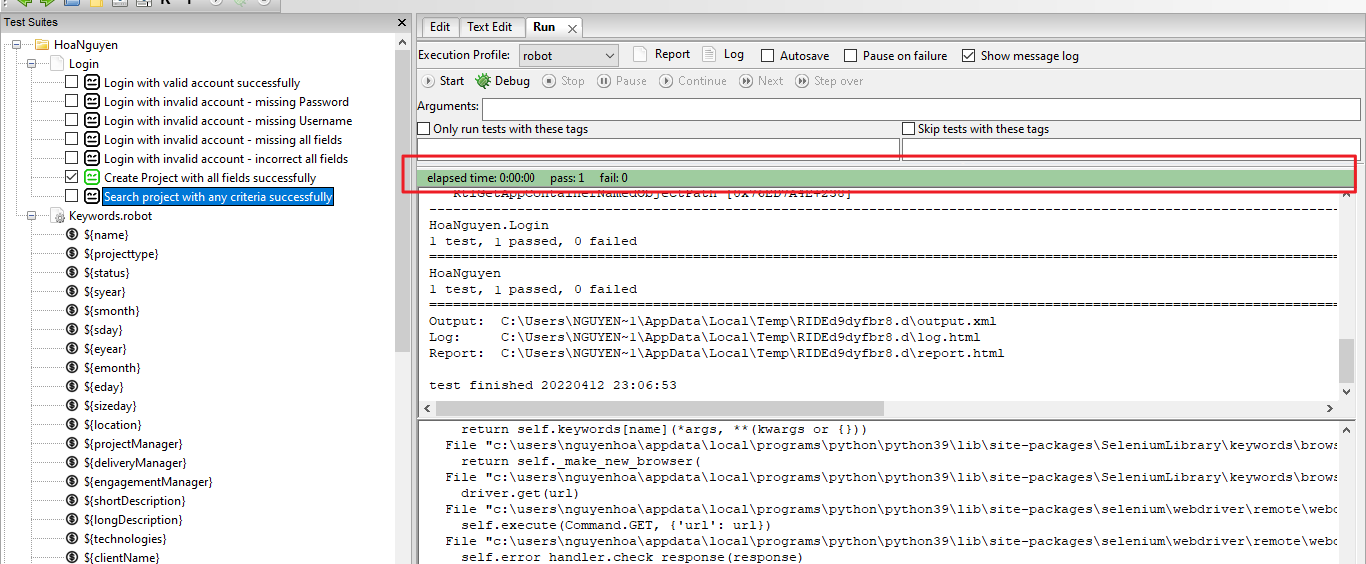


Figure 6.17: Screen of Test Case successfully

# **CONCLUSION**

## **7.1. Results:**

After researching and researching this topic for a while, have reached the following results:

- Learn about a very important and indispensable phase in the software development process, that is testing. Especially go deeper in testing.

- Learn how to build test cases, generate Bug reports.

- Successfully applied Robot Framework software to generate input data for automated functional testing of a software.

- Robot Framework is a free, open-source automation tool, is easy to install and keyword-driven.

- Automated testing brings a lot of benefits: Saves testing time, Automation does not need human intervention so can run automated tests without supervision, Automatically speeds up test execution, Automation increases test coverage, Manual testing can become boring and hence error prone, Improves accuracy, 70% faster than manual testing.

## **7.2. Limitations**

In first time, experimenting with building Test Case and automated testing with Robot Framework, do not have much in-depth knowledge about this field, there are still limitations in this project. after:

- Due to the limitation of report length, some test cases are not covered in this report.

- Since the application is newly built, it is inevitable that some Test cases will fail quite a lot.

- Due to lack of experience, it is inevitable that test cases will be missing some cases.

- Have not built a number of ad-hod Test cases, to dig deeper into finding bugs and propose some solutions to improve the software.

## **7.3. Future works**

In order for the project to be completed and more effective, orient the work to be done in the future as follows:

- Build more ad-hoc Testing

- Apply a variety of automated testing tools such as Selenium, Cucumber, QTP (HP UFT), WATIR,...

- In addition, apply more API testing to help the testing process happen quickly when some functions do not have UI yet.

# **REFERENCES**

[1] Nguyễn Văn Hà, Nguyễn Văn Vỵ, Giáo trình Kỹ nghệ phần mềm, Hà Nội: Nhà xuất bản giáo dục, 2009.

[2] C. Kaner, Exploratory Testing, Orlando, FL: Florida Institute of Technology, November 2006.

[3] Trương Anh Hoàng, Đặng Văn Hưng, Phạm Ngọc Hùng, Giáo trình kiểm thử phần mềm, Hà Nội: Đại học Quốc gia Hà Nội, Tháng 1 năm 2014.

[3] Robot Framework, <https://robotframework.org>, Retrieved March 15th, 2022.

[4] NashTech, About us, <https://nashtechglobal.com>, Retrieved February 10th, 2022.