**HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY AND EDUCATION**

**FACULTY FOR HIGH QUALITY TRAINING**

**

**GRADUATE THESIS**

**BUILD AN APPLICATION THAT SUPPORTS AUTOMATION TESTING**

**VO THI KIEU DIEM**

**Student ID: 17110109**

**BUI THI HONG NHUNG**

**Student ID**: **17110201**

**PHAM THI THANH HANG**

**Student ID: 17110132**

**Course: 2017**

**Major**: **INFORMATION TECHNOLOGY**

**Advisor**: **NGUYEN DUC KHOAN, M.Sc.**

Ho Chi Minh City, July 2021

**HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY AND EDUCATION**

**FACULTY FOR HIGH QUALITY TRAINING**

**

**GRADUATE THESIS**

**BUILD AN APPLICATION THAT SUPPORTS AUTOMATION TESTING**

**VO THI KIEU DIEM**

**Student ID: 17110109**

**BUI THI HONG NHUNG**

**Student ID**: **17110201**

**PHAM THI THANH HANG**

**Student ID: 17110132**

**Course: 2017**

**Major**: **INFORMATION TECHNOLOGY**

**Advisor**: **NGUYEN DUC KHOAN, M.Sc.**

Ho Chi Minh City, July 2021

|  |  |
| --- | --- |
|  | THE SOCIALIST REPUBLIC OF VIETNAM  **Independence – Freedom– Happiness**  --------  *Ho Chi Minh City, July 01, 2021* |

GRADUATION PROJECT ASSIGNMENT

Student name: VO THI KIEU DIEM Student ID: 17110109

Student name: BUI THI HONG NHUNG Student ID: 17110201

Student name: PHAM THI THANH HANG Student ID: 17110132

Major: Information Technology Class: 17110CLST

Advisor: NGUYEN DUC KHOAN M.Sc. Phone number:

Date of assignment: 01/01/2021 Date of submission: 01/07/2021

|  |
| --- |
| 1. Project title: Build an application that supports automation testing. 2. Initial materials provided by the advisor: Implementation using Java Selenium Framework, Blazor, ASP.NET CORE 3.x, SQL Server. 3. Content of the project: The main scope builds an application that supports automation testing, test cases are written on .CSV files and use Selenium to perform automated testing. 4. Final product: Design and build a web application full of basic automated testing and management functions. |

**CHIEF OF DEPARTMENT ADVISOR**

*(Sign with full name) (Sign with full name)*

|  |  |
| --- | --- |
|  | THE SOCIALIST REPUBLIC OF VIETNAM  **Independence – Freedom– Happiness**  --------  *Ho Chi Minh City, July 01, 2021* |

ADVISOR’S EVALUATION SHEET

Student name: VO THI KIEU DIEM Student ID: 17110109

Student name: BUI THI HONG NHUNG Student ID: 17110201

Student name: PHAM THI THANH HANG Student ID: 17110132

Major: Information Technology Class: 17110CLST

Project title: BUILD AN APPLICATION THAT SUPPORTS AUTOMATION TESTING.

Advisor: NGUYEN DUC KHOAN, M.Sc.

**EVALUATION**

1. Content of the project:

.......................................................................................................................................

.......................................................................................................................................

1. Strengths:

.......................................................................................................................................

.......................................................................................................................................

1. Weaknesses:

.......................................................................................................................................

.......................................................................................................................................

1. Approval for oral defense? (Approved or denied)

.......................................................................................................................................

1. Overall evaluation: (Excellent, Good, Fair, Poor)

.......................................................................................................................................

1. Mark:……………….(in words:................................................................................)

*Ho Chi Minh City, .............................*

**ADVISOR**

*(Sign with full name)*

|  |  |
| --- | --- |
|  | THE SOCIALIST REPUBLIC OF VIETNAM  **Independence – Freedom– Happiness**  --------  *Ho Chi Minh City, July 01, 2021* |

PRE-DEFENSE EVALUATION SHEET

Student name: VO THI KIEU DIEM Student ID: 17110109

Student name: BUI THI HONG NHUNG Student ID: 17110201

Student name: PHAM THI THANH HANG Student ID: 17110132

Major: Information Technology Class: 17110CLST

Project title: BUILD AN APPLICATION THAT SUPPORTS AUTOMATION TESTING.

Name of Reviewer: ......................................................................................................

**EVALUATION**

1. Content of the project:

.......................................................................................................................................

.......................................................................................................................................

1. Strengths:

.......................................................................................................................................

.......................................................................................................................................

1. Weaknesses:

.......................................................................................................................................

.......................................................................................................................................

1. Approval for oral defense? (Approved or denied)

.......................................................................................................................................

1. Overall evaluation: (Excellent, Good, Fair, Poor)

.......................................................................................................................................

1. Mark:……………….(in words: .............................................................................)

*Ho Chi Minh City, .............................*

**REVIEWER**

*(Sign with full name)*

ACKNOWLEDGEMENT

First of all, the implementation team would like to send their sincere thanks to the High-Quality Training Department - the Ho Chi Minh City University of Technical Education for creating all the most favorable conditions for the group to perform. The association is free to access, refer, and expand knowledge in the field of Information Technology in general and the Graduation Thesis in particular!

The sincerest thanks to the implementation team would like to sincerely send to **Mr**. **NGUYEN DUC KHOAN** - who used all the teacher's enthusiasm and knowledge, accompanied and directly taught, guided, and created all favorable conditions, help the group to develop their best as well as improve their knowledge throughout the learning process, especially in the process of preparing and implementing the Good Thesis Course. Thank you for your enthusiasm, which is a tremendous motivation to help the group perform, persevere throughout the process of implementing the topic and discover new interesting and useful knowledge related to the topic. namely, building an application that supports automation testing.

**Student groups perform,**

**Best regards**

FOREWORD

Today information technology has been developing to a high level, is applied to all sectors and fields of the country's economy, making an important contribution to making these industries develop faster and more efficiently. The design and construction of websites in this era of 4.0 technology to meet the increasing needs of users in many fields, in addition, the field of software testing also plays an important role in helping the product become better and not be ignored.

With Automation Test, Tester just needs to write a piece of code or use some tools like Selenium... to run all steps automatically including entering information, clicking, checking results, comparing actual results with hypothetical results. Therefore, it is necessary to build a website allowing user login to manage projects and test functional screens with data read from the input file.

# **TABLE OF CONTENTS**

[GRADUATION PROJECT ASSIGNMENT i](#_Toc77708544)

[ADVISOR’S EVALUATION SHEET ii](#_Toc77708545)

[PRE-DEFENSE EVALUATION SHEET iii](#_Toc77708546)

[ACKNOWLEDGEMENT iv](#_Toc77708547)

[FOREWORD v](#_Toc77708548)

[TABLE OF CONTENTS vi](#_Toc77708549)

[ABSTRACT x](#_Toc77708550)

[LIST OF TABLES xi](#_Toc77708551)

[LIST OF FIGURES xiii](#_Toc77708552)

[Chapter 1: INTRODUCTION 16](#_Toc77708553)

[1.1. Problem 16](#_Toc77708554)

[1.2. Current Situation 16](#_Toc77708555)

[1.3. Proposed Solution 16](#_Toc77708556)

[1.3.1. Purpose 16](#_Toc77708557)

[1.3.2. Scope 16](#_Toc77708558)

[Chapter 2: APPLICATION REVIEW AND TECHNOLOGIES 18](#_Toc77708559)

[2.1. Application Review 18](#_Toc77708560)

[2.1.1. Selenium IDE 18](#_Toc77708561)

[2.1.2. Test Collab 19](#_Toc77708562)

[2.1.3. Conclusion 20](#_Toc77708563)

[2.2. Technologies 21](#_Toc77708564)

[2.2.1. Selenium Testing 21](#_Toc77708565)

[2.2.2. ASP.NET Entity Framework 24](#_Toc77708566)

[2.2.3. Microsoft SQL Server 2017 25](#_Toc77708567)

[2.2.4. Blazor 25](#_Toc77708568)

[Chapter 3: REQUIREMENTS 27](#_Toc77708569)

[3.1. User Requirement 27](#_Toc77708570)

[3.2. System Requirement 28](#_Toc77708571)

[3.2.1. Actor 28](#_Toc77708572)

[3.2.2. Use Case 29](#_Toc77708573)

[3.2.2.1 Use Case Diagram 29](#_Toc77708574)

[3.2.2.2 Use Case Description 31](#_Toc77708575)

[3.3. Non-Functional Requirement 64](#_Toc77708576)

[3.4. Entity-Relationship Diagram 65](#_Toc77708577)

[Chapter 4: SYSTEM DESIGN & IMPLEMENTATION 66](#_Toc77708578)

[4.1. File Test Design 66](#_Toc77708579)

[4.1.1. File Input Design 66](#_Toc77708580)

[4.1.2. File Output Design 69](#_Toc77708581)

[4.1.3. File Log Design 70](#_Toc77708582)

[4.2. System Architecture 71](#_Toc77708583)

[4.3. System Component 72](#_Toc77708584)

[4.3.1. Web MVC Component Diagram 72](#_Toc77708585)

[4.4. Class Diagram. 72](#_Toc77708586)

[4.4.1. Class Description for Class BaseModel 73](#_Toc77708587)

[4.4.2. Class Description for Class Project 73](#_Toc77708588)

[4.4.3. Class Description for Class ProjectDetail 73](#_Toc77708589)

[4.4.4. Class Description for Class FunctionTesting 73](#_Toc77708590)

[4.4.5. Class Description for Class TestScreen 74](#_Toc77708591)

[4.4.6. Class Description for Class Report 74](#_Toc77708592)

[4.4.7. Class Description for Class TestSchedule 74](#_Toc77708593)

[4.5. Database Diagram. 75](#_Toc77708594)

[4.6. Sequence Diagram. 76](#_Toc77708595)

[4.6.1. Login Sequence Diagram 76](#_Toc77708596)

[4.6.2. Run Test Sequence Diagram 77](#_Toc77708597)

[4.6.3. Import File Sequence Diagram 77](#_Toc77708598)

[4.7. Activity Diagram. 78](#_Toc77708599)

[4.7.1. Daily Schedule Test Activity Diagram 78](#_Toc77708600)

[4.7.2. One Time Schedule Test Activity Diagram 78](#_Toc77708601)

[4.8. User Interface 78](#_Toc77708602)

[4.8.1. Screen Flow for General 78](#_Toc77708603)

[4.8.1.1 SCG001. Login 79](#_Toc77708604)

[4.8.1.2 SCG002. Register Account 81](#_Toc77708605)

[4.8.2. Screen Flow for Customer 82](#_Toc77708606)

[4.8.2.1 SCC001. Home page 83](#_Toc77708607)

[4.8.2.2 SCC002. Manage Projects 84](#_Toc77708608)

[4.8.2.3 SCC003. Manage Project Functions 86](#_Toc77708609)

[4.8.2.4 SCC004. Manage File Test Of Function 88](#_Toc77708610)

[4.8.2.5 SCC005. Run Test 91](#_Toc77708611)

[4.8.2.6 SCC006. Report 92](#_Toc77708612)

[4.9. System Implementation 93](#_Toc77708613)

[4.9.1. Development Environment 93](#_Toc77708614)

[4.9.1.1 Automation Test 93](#_Toc77708615)

[4.9.2. Source Code Management 95](#_Toc77708616)

[Chapter 5: SELENIUM IDE 96](#_Toc77708617)

[5.1. Project 96](#_Toc77708618)

[5.2. Test Case 96](#_Toc77708619)

[5.3. Guideline 97](#_Toc77708620)

[5.4. Advantage 101](#_Toc77708621)

[5.5. Disadvantage 101](#_Toc77708622)

[Chapter 6: SYSTEM TESTING 102](#_Toc77708623)

[6.1. Test Plan 102](#_Toc77708624)

[6.1.1. List Of Stages Of Testing 102](#_Toc77708625)

[6.1.2. List Of Test Types 102](#_Toc77708626)

[6.1.3. Constraints 102](#_Toc77708627)

[6.1.4. Plan Defect 102](#_Toc77708628)

[6.2. Test Design 102](#_Toc77708629)

[6.2.1. Items To Be Tested 102](#_Toc77708630)

[6.2.2. Features To Be Tested 102](#_Toc77708631)

[6.3. Test Case 103](#_Toc77708632)

[6.3.1. Test Case For Web Application 103](#_Toc77708633)

[6.3.1.1 Test Case Of Login (Lg) 103](#_Toc77708634)

[6.3.1.2 Test Case Of Project (Prt) 104](#_Toc77708635)

[6.3.1.3 Test Case Of Manage Project Function (Prf) 105](#_Toc77708636)

[6.3.1.4 Test Case Of Manage File Test (Mft) 106](#_Toc77708637)

[6.3.1.5 Test Case Of Run Test (Rt) 107](#_Toc77708638)

[6.4. Experiment Results 108](#_Toc77708639)

[6.5. System Deployment 110](#_Toc77708640)

[6.5.1. Publish And Deploy On Heroku Hosting 110](#_Toc77708641)

[Chapter 7: CONCLUSION 114](#_Toc77708642)

[7.1. Result 114](#_Toc77708643)

[7.2. Advantage 114](#_Toc77708644)

[7.3. Disadvantage 114](#_Toc77708645)

[7.4. Future Plan 115](#_Toc77708646)

[References 116](#_Toc77708647)

[Appendices 117](#_Toc77708648)

ABSTRACT

|  |  |
| --- | --- |
| **Term** | **Definition** |
| DB | Database |
| VPS | Virtual Private Server |
| JS | JavaScript |
| IDE | Integrated Development Environment |
| CSV | Comma Separated Values |
| Lg | Login |
| Prt | Project |
| Prf | Project Function |
| Mft | Manage File Test |
| Rt | Run test |

LIST OF TABLES

[Table 3.1 User requirement 27](#_Toc77708410)

[Table 3.2 Actor 28](#_Toc77708411)

[Table 3.3 Use Case 30](#_Toc77708412)

[Table 3.4 UC\_1.1 Register 31](#_Toc77708413)

[Table 3.5 UC\_1.2 Login 33](#_Toc77708414)

[Table 3.6 UC\_2.1 Add Project 35](#_Toc77708415)

[Table 3.7 UC\_2.2 Delete Project 37](#_Toc77708416)

[Table 3.8 UC\_2.3 Edit Project 39](#_Toc77708417)

[Table 3.9 UC\_2.4 View Project Detail 41](#_Toc77708418)

[Table 3.10 UC\_3.1 Add Function 42](#_Toc77708419)

[Table 3.11 UC\_3.2 Edit Function 44](#_Toc77708420)

[Table 3.12 UC\_3.3 Delete Function 46](#_Toc77708421)

[Table 3.13 UC\_3.4 View Function Detail 48](#_Toc77708422)

[Table 3.14 UC\_3.5 Schedule Test 49](#_Toc77708423)

[Table 3.15 UC\_3.6 Send mail 50](#_Toc77708424)

[Table 3.16 UC\_4.1 Import File 52](#_Toc77708425)

[Table 3.17 UC\_4.2 Download File 54](#_Toc77708426)

[Table 3.18 UC\_4.3 Delete File 55](#_Toc77708427)

[Table 3.19 UC\_4.4 View Test Screen 57](#_Toc77708428)

[Table 3.20 UC\_4.5 Run Test 59](#_Toc77708429)

[Table 3.21 UC\_4.6 View Log Steps 60](#_Toc77708430)

[Table 3.22 UC\_4.7 Download File Template 62](#_Toc77708431)

[Table 3.23 Non-Functional Requirement 64](#_Toc77708432)

[Table 4.1 Keyword of Action column 66](#_Toc77708433)

[Table 4.2 Keyword of Target column 67](#_Toc77708434)

[Table 4.3 Keyword of Value column 67](#_Toc77708435)

[Table 4.4 Description about architecture. 71](#_Toc77708436)

[Table 4.5 Class description for class BaseModel 73](#_Toc77708437)

[Table 4.6 Class description for class Project 73](#_Toc77708438)

[Table 4.7 Class description for class ProjectDetail 73](#_Toc77708439)

[Table 4.8 Class description for class FunctionTesting 73](#_Toc77708440)

[Table 4.9 Class description for class TestScreen 74](#_Toc77708441)

[Table 4.10 Class description for class Report 74](#_Toc77708442)

[Table 4.11 Class description for class TestSchedule 74](#_Toc77708443)

[Table 4.12 Database Diagram Description. 75](#_Toc77708444)

[Table 4.13 Screen flow for general. 78](#_Toc77708445)

[Table 4.14 Login description 79](#_Toc77708446)

[Table 4.15 Register Account description 81](#_Toc77708447)

[Table 4.16 Screen flow for Customer 82](#_Toc77708448)

[Table 4.17 Home page description 83](#_Toc77708449)

[Table 4.18 Manage Projects description 84](#_Toc77708450)

[Table 4.19 Add/Edit Project description 85](#_Toc77708451)

[Table 4.20 Manage Project Functions description 86](#_Toc77708452)

[Table 4.21 Add/Edit Functions description 87](#_Toc77708453)

[Table 4.22 Manage File Test of Function description 88](#_Toc77708454)

[Table 4.23 Upload File description 90](#_Toc77708455)

[Table 4.24 Run Test description 91](#_Toc77708456)

[Table 4.25 Report description 92](#_Toc77708457)

[Table 4.26 Development environment 93](#_Toc77708458)

[Table 6.1 Test case of Login (Lg) 103](#_Toc77708459)

[Table 6.2 Test case of Project (Prt) 104](#_Toc77708460)

[Table 6.3 Test case of Manage Project Function (Prf) 105](#_Toc77708461)

[Table 6.4 Test case of Manage File Test (Mft) 107](#_Toc77708462)

[Table 6.5 Test case of Run Test (Rt) 108](#_Toc77708463)

[Table 7.1 Job table of work 117](#_Toc77708464)

LIST OF FIGURES

[Figure 2.1 Selenium IDE 18](#_Toc77708465)

[Figure 2.2 Test Collab 20](#_Toc77708466)

[Figure 2.3 The popularity of Selenium 21](#_Toc77708467)

[Figure 2.4 Component Selenium 22](#_Toc77708468)

[Figure 2.5 WebDriver 23](#_Toc77708469)

[Figure 2.6 Browsers – Selenium 24](#_Toc77708470)

[Figure 3.1 Use Case Diagram 29](#_Toc77708471)

[Figure 3.2 UC\_1.1 Register 31](#_Toc77708472)

[Figure 3.3 UC\_1.2 Login 33](#_Toc77708473)

[Figure 3.4 UC\_2.1 Add Project 35](#_Toc77708474)

[Figure 3.5 UC\_2.2 Delete Project 37](#_Toc77708475)

[Figure 3.6 UC\_2.3 Edit Project 39](#_Toc77708476)

[Figure 3.7 UC\_2.4 View Project Detail 41](#_Toc77708477)

[Figure 3.8 UC\_3.1 Add Function 42](#_Toc77708478)

[Figure 3.9 UC\_3.2 Edit Function 44](#_Toc77708479)

[Figure 3.10 UC\_3.3 Delete Function 46](#_Toc77708480)

[Figure 3.11 UC\_3.4 View Function Detail 47](#_Toc77708481)

[Figure 3.12 UC\_3.5 Schedule Test 49](#_Toc77708482)

[Figure 3.13 UC\_3.6 Send mail 50](#_Toc77708483)

[Figure 3.14 UC\_4.1 Import File 52](#_Toc77708484)

[Figure 3.15 UC\_4.2 Download File 54](#_Toc77708485)

[Figure 3.16 UC\_4.3 Delete File 55](#_Toc77708486)

[Figure 3.17 UC\_4.4 View Test Screen 57](#_Toc77708487)

[Figure 3.18 UC\_4.5 Run Test 58](#_Toc77708488)

[Figure 3.19 UC\_4.6 View Log Steps 60](#_Toc77708489)

[Figure 3.20 UC\_4.7 Download File Template 62](#_Toc77708490)

[Figure 3.21 Entity-Relationship Diagram 65](#_Toc77708491)

[Figure 4.1 File Input Design 66](#_Toc77708492)

[Figure 4.2 File Output Design 69](#_Toc77708493)

[Figure 4.3 Testcase Fail. 69](#_Toc77708494)

[Figure 4.4 Testcase fail (example) 70](#_Toc77708495)

[Figure 4.5 File Log Design. 70](#_Toc77708496)

[Figure 4.6 Architecture diagram for the system. 71](#_Toc77708497)

[Figure 4.7 Web MVC Component diagram 72](#_Toc77708498)

[Figure 4.8 Database Diagram. 75](#_Toc77708499)

[Figure 4.9 Login sequence diagram. 76](#_Toc77708500)

[Figure 4.10 Run test sequence diagram 77](#_Toc77708501)

[Figure 4.11 Import file sequence diagram. 77](#_Toc77708502)

[Figure 4.12 Daily schedule test activity diagram 78](#_Toc77708503)

[Figure 4.13 One time schedule test activity diagram 78](#_Toc77708504)

[Figure 4.14 Login 79](#_Toc77708505)

[Figure 4.15 Forgot password. 81](#_Toc77708506)

[Figure 4.16 Screen flow for Customer 82](#_Toc77708507)

[Figure 4.17 SCC001. Home page 83](#_Toc77708508)

[Figure 4.18 SCC002. Manage Projects 84](#_Toc77708509)

[Figure 4.19 SCC002-1. Add/Edit Project 85](file:///C:\Users\kieud\Downloads\VoThiKieuDiem_BuiThiHongNhung_PhamThiThanhHang.docx#_Toc77708510)

[Figure 4.20 SCC003. Manage Project Functions 86](#_Toc77708511)

[Figure 4.21 SCC003-1. Add/Edit Functions. 87](#_Toc77708512)

[Figure 4.22 SCC004. Manage File Test of Function. 88](#_Toc77708513)

[Figure 4.23 SCC004-1. Upload File 90](#_Toc77708514)

[Figure 4.24 SCC005. Run Test 91](#_Toc77708515)

[Figure 4.25 Run Test - Log 91](#_Toc77708516)

[Figure 4.26 SCC006. Report 92](#_Toc77708517)

[Figure 5.1 URL 96](#_Toc77708518)

[Figure 5.2 Selenium screen 96](#_Toc77708519)

[Figure 5.3 Install Selenium IDE 97](#_Toc77708520)

[Figure 5.4 Turn on Selenium IDE. 97](#_Toc77708521)

[Figure 5.5 Create a new project. 98](#_Toc77708522)

[Figure 5.6 Create a new project(1). 98](#_Toc77708523)

[Figure 5.7 Playback BaseUrl 98](#_Toc77708524)

[Figure 5.8 Start recording 99](#_Toc77708525)

[Figure 5.9 Recording. 99](#_Toc77708526)

[Figure 5.10 Save Test Name 100](#_Toc77708527)

[Figure 5.11 Run current test 100](#_Toc77708528)

[Figure 5.12 Save test case. 101](#_Toc77708529)

[Figure 6.1 Features to be tested 103](#_Toc77708530)

[Figure 6.2 Testcase of Login. 108](#_Toc77708531)

[Figure 6.3 Test case of Project. 109](#_Toc77708532)

[Figure 6.4 Test case of Manage Project Function 109](#_Toc77708533)

[Figure 6.5 Test case of Manage File Test 109](#_Toc77708534)

[Figure 6.6 Test case of Run Test 110](#_Toc77708535)

[Figure 6.7 Heroku dashboard 110](#_Toc77708536)

[Figure 6.8 Create a new app 111](#_Toc77708537)

[Figure 6.9 Connect 111](#_Toc77708538)

[Figure 6.10 Add buildpack. 112](#_Toc77708539)

[Figure 6.11 Buildpack URL. 112](#_Toc77708540)

[Figure 6.12 Starts building 113](#_Toc77708541)

[Figure 6.13 View to run 113](#_Toc77708542)

[Figure 6.14 Team’s website 113](#_Toc77708543)

# INTRODUCTION

## Problem

Today, with the strong development of technology 4.0 as well as the development of many professions, especially the applications of the information technology industry. There are more and more products about software, so the software testing field is also playing an important role to help products become better.

In addition to checking Manual Tests by hand, nowadays, the Automation Test also contributes to the productivity of testing, reduces the boredom of testing by hand for a long time, and repeating many times. From there the team got the idea of developing a software testing support system to help the tester can easily perform the testing process.

## Current Situation

Currently, there are many websites that support testing but mainly focus on managing test suites, creating test cases, creating test plans, assigning test executions, but not much support for users in terms of automated testing.

So, when conducting testing, users often have to perform manual testing based on the created test cases. This is quite time-consuming, labor intensive, and inefficient.

## Proposed Solution

### Purpose

The content of the thesis has two main purposes:

* Build an application that supports automation testing with test cases are written on .CSV files.
* Using Selenium to write test cases and test automation.

### Scope

Build an application that supports automation testing aims to solve the test execution difficulties and easily handle when there is a change coming from the website to be tested.

Research and design the input data file of the test system: .CSV file.

+ Contains step-by-step details for testers to track the error checking process and contains test data.

+ For the case of unconstrained content: design random values ​​(string, number, date).

Manage test process: schedule automatic execution for each .csv file, results of each run, export report and display log.

Project management, Project functions and input files.

Build web API to connect between components in the test system.

Use Selenium IDE to execute automated test on a specific website and build test cases.

# APPLICATION REVIEW AND TECHNOLOGIES

## Application Review

### Selenium IDE

Selenium IDE [1] (Integrated Development Environment) is the simplest tool in the Selenium Suite. It is a Firefox add-on that creates tests very quickly through its record-and-playback functionality.

Selenium IDE supports autocomplete mode when creating tests.

* It helps the tester to enter commands more quickly.
* It restricts the user from entering invalid commands.

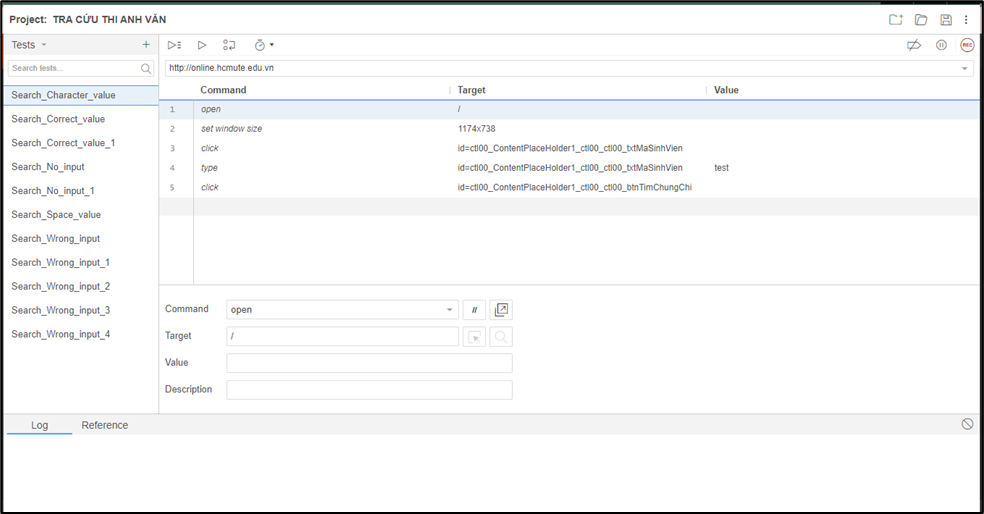


Figure 2.1 Selenium IDE

Functions:

* The menu bar allows the user to create, save, edit, and convert the recorded Selenium IDE test scripts.
* The toolbar allows the user to set the execution speed of 1 test case (Fast-Slow), pause and resume the test case, roll up commands, etc.
* Base URL: is the place where the homepage of the web.
* Test Case web application is stored: list listing all test cases in the test suite.
* There are two execution modes: run all tests in suite case and run the current test case.
* The editor allows the user to record or create test scripts. The editor has two views “table” and “source”.
* In the table view, each test step is comprised of a command, target, and a value.
* Source view displays the test case in the HTML format.
* Test case pane shows a comprehensive list of failed and passed test cases with the relevant color-coding.
* Log Pane displays the test execution heath in the form of message.
* Log messages can be saved in a file using “File Logging” plug-in.
* Reference pane shows the description of every selected command.

Advantages:

* Manage many test cases, many projects.
* Allows to archive the record tables for each test step.
* Clear demonstration of test steps (combination of command, target, and value).
* Can display test results in the message format.
* Can convert to other programming languages ​​such as HTML, Java, C#, Python, Ruby
* Can debug, set breakpoints, add comments to scripts for readability.
* Can edit test case content: command (open, click...), target, value and add a description.

Disadvantages:

* Test results have not been sent to email.
* Condition expressions/loops are not supported (additional installation is required to run the loop).
* The test report is not supported (additional settings are required to display the report).
* Do not allow reading data from file: text (.txt), excel (.xls), .csv… (need additional installation to read data from .csv/ .xml file).
* Difficult to handle complex cases/ workflow.

### Test Collab

Test management tool to help development teams carry out testing effectively and avoid failures. It integrates with all popular bug trackers and test automation tools.

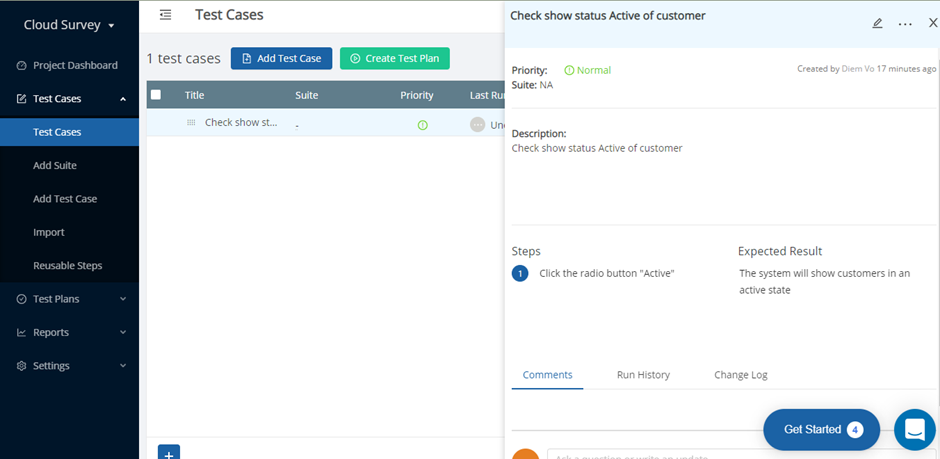


Figure 2.2 Test Collab

* Functions:
  + Create projects.
  + Manage project's requirements.
  + Set milestones.
  + Define test suites, create test cases, create test plans, assign test executions, get them running, and analyze the results.
* Advantages:
  + Supports agile methodology, requirements management, test plans and scheduling.
  + Supports two ways integration with issue tracking systems like JIRA and Redmine.
  + Can also be integrated with any test automation tool.
* Disadvantages:
  + Does not offer the option to create custom reports.

### Conclusion

Through the survey of related products, the group draws out the required functions:

Customer role:

* Register, log in to your account, edit personal information.
* Allows project management, project function management, test case file management.
* Import and export .CSV files.
* Automated test execution.
* View details of test case file and log when running on test screen.
* Schedule test execution.
* Export report results sent to email.

## Technologies

### Selenium Testing

Selenium is a set of open-source web automation tools that leverages the power of web browsers and helps in automating workflows of how users interact with the web application within the browser.

The primary purpose of Selenium, as highlighted by the Selenium Website, is – “Selenium Automates Browsers, what you do with this power is entirely up to you.” Even with the emerging tools and technologies, Selenium is still leading the board in the list of Web Automation tools and Automation Testing. The below survey conducted by Katalon depicts the popularity of Selenium:

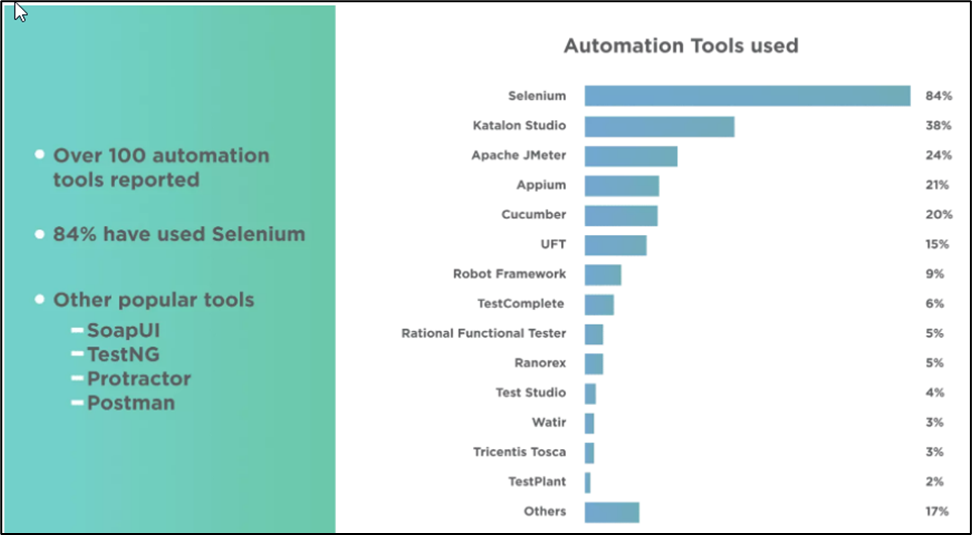


Figure 2.3 The popularity of Selenium

Component Selenium:

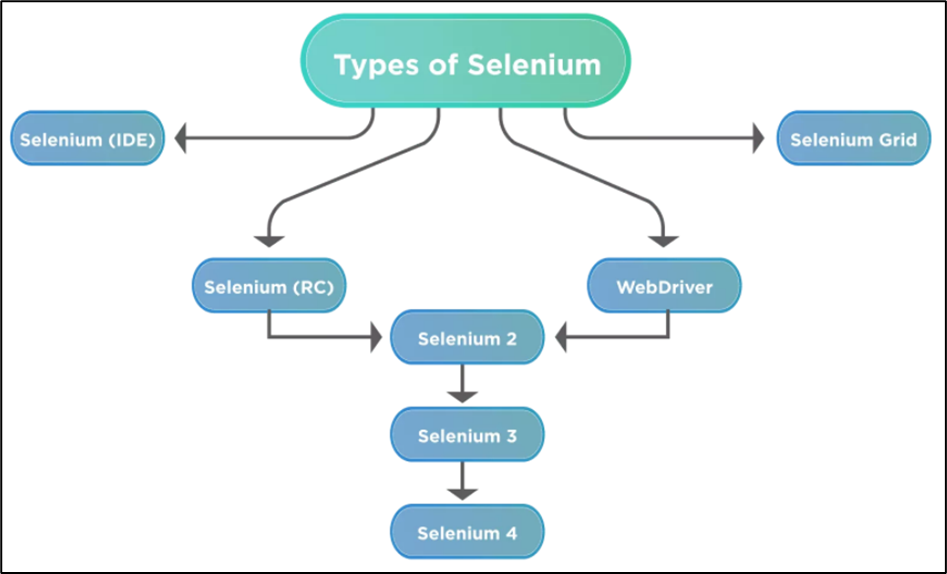


Figure 2.4 Component Selenium

In our project, our team focus on Selenium IDE and Selenium WebDriver.

* Selenium IDE is an extension available for both Firefox and Chrome, which has the record and replay functionality available. Selenium IDE also now has the capability of exporting the code in various predefined languages. It also gives the ability to use one test case inside another.
* Selenium WebDriver is the most used component of Selenium.
* WebDriver allows users to write custom code in their language of choice and interact with the browser of their choice, through browser-specific drivers.
* WebDriver works on the OS level and uses a Protocol called [JSON Wire Protocol](https://github.com/SeleniumHQ/selenium/wiki/JsonWireProtocol) to communicate with browsers.[2]

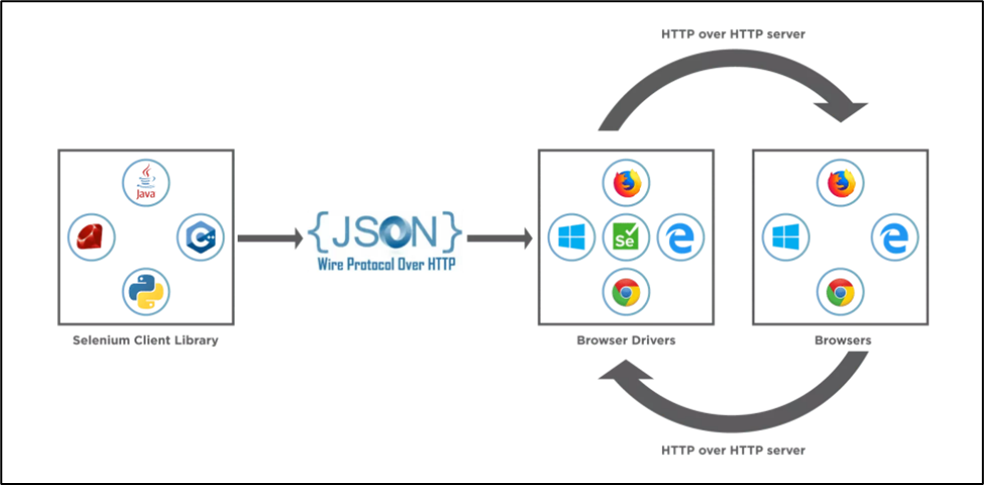


Figure 2.5 WebDriver

This image tells us about the **core selenium webdriver architecture and the major selenium components** which comprise WebDriver.

* **Selenium WebDriver Client Libraries / Language Bindings** – Software Testers want to select languages that they are comfortable with. Since WebDriver Architecture supports different languages, so there are bindings available for a range of languages like Java, C#, Python, Ruby, PHP, etc. Anyone who has a basic knowledge of working with any programming language can get specific language bindings and can start off. This is how Selenium Architecture provides flexibility to testers to do automation in their comfort zone.
* **Json Wire Protocol** – As per the Selenium Architecture above, the JSON Wire Protocol facilitates all the communication that is happening in Selenium between the browser and the code. This is the heart of Selenium. JSON Wire Protocol provides a medium for data transfer using a RESTful (Representational State Transfer) API which provides a transport mechanism and defines a RESTful web service using JSON over HTTP.
* **Browser Drivers** – Since there are various browsers that are supported by Selenium, each browser has its own implementation of the W3C standard that Selenium provides. As such browser-specific binaries are available that are specific to the browser and hides the implementation logic from the end-user. JSONWire protocol establishes a connection between the browser binaries and the client libraries.
* **Browsers** – Selenium will be only able to run tests on the browsers if they are locally installed, either on the local machine or on the server machines. So, browser installation is necessary.

Diagram

Description automatically generated

Figure 2.6 Browsers – Selenium

* When a user writes a WebDriver code in Selenium and executes it, the following actions happen in the background.
* An HTTP request generates, and it goes to the respective browser driver (Chrome, IE, Firefox). There is an individual request for each Selenium command.
* The browser driver receives the request through an HTTP server.
* The HTTP server decides which actions/instructions need to execute on the browser.
* The browser executes the instructions/steps as decided above.
* The HTTP server then receives the execution status and then sends back the status to an automation script, which then shows the result (as passed or an exception or error).[3]

### ASP.NET Entity Framework

ASP.NET extends the .NET platform with tools and libraries specifically for building web apps. These are some things that ASP.NET adds to the .NET platform:

* Base framework for processing web requests in C# or F#.
* Web-page templating syntax, known as Razor, for building dynamic web pages using C#.
* Libraries for common web patterns, such as Model View Controller (MVC).
* Authentication system that includes libraries, a database, and template pages for handling logins, including multi-factor authentication and external authentication with Google, Twitter, and more.
* Editor extensions to provide syntax highlighting, code completion, and other functionality specifically for developing web pages. [4]

With the Code First approach, Entity Framework creates database table objects based on the model the programmer creates to represent application data.

Pros: Very universal, full control over model code, super easy to add - remove - edit properties without thinking about DB.

Cons: Direct structural changes to the DB will be difficult to control with the columns that will be created on the DB, a bit difficult to combine with an existing DB.

### Microsoft SQL Server 2017

Microsoft SQL Server is a relational database management system developed by Microsoft. As a database server, it is a software product with the primary function of storing and retrieving data as requested by other software applications which may run either on the same computer or on another computer across a network (including the Internet). Microsoft markets at least a dozen different editions of Microsoft SQL Server, aimed at different audiences and for workloads ranging from small single-machine applications to large Internet-facing applications with many concurrent users.

SQL Server 2017 represents a major step towards making SQL Server a platform that gives you choices of development languages, data types, on-premises or cloud, and operating systems by bringing the power of SQL Server to Linux, Linux-based Docker containers, and Windows [5].

Advantage:

* Cross-Platform Compatibility.
* Reduction of Total Cost of Ownership.
* Better Statistical and Data Science Analysis Services.
* Better Performance.
* Best-in-Class Security.
* Cross-Platform Visual Studio.
* Linux Support.

### Blazor

Blazor lets you build interactive web UIs using C# instead of JavaScript. Blazor apps are composed of reusable web UI components implemented using C#, HTML, and CSS. Both client and server code are written in C#, allowing you to share code and libraries.

Blazor is a feature of ASP.NET, the popular web development framework that extends the .NET developer platform with tools and libraries for building web apps.

Blazor is part of the open-source .NET platform that has a strong community of contributors from more than 3,700 companies. [6]

Blazor web assembly runs on a web browser it some functions similar to JavaScript framework Angular JS or Reacts.

Features of Blazor Framework:

* Forms and validation feature.
* Dependency injection features.
* Routing features.
* Layout features.
* Advantages of Blazor:
* Make web application using C#.
* It provides an interactive look to our website.
* Blazor server allows the application to run on the server-side.
* In the Blazor web assembly, code can be share client-side and server-side.
* Disadvantages of Blazor:
* Following disadvantages of the Blazor web application
* It is not compatible with all browsers. It is compatible only with a modern browser.
* Small communities developed the Blazor web assembly in comparison to Angular or React.js.

# REQUIREMENTS

## User Requirement

Table . User requirement

|  |  |  |
| --- | --- | --- |
| **#** | **Title** | **User Story** |
|  | The excel file contains the test execution steps and maybe have the random valuation. | File .CSV contains step-by-step details for testers to track the error checking process and contains test data.  For the case of the unconstrained content: design random values (string, number, date). |
|  | Test automation cases can be handled. | Test on textbox, dropdown, alert box, inner element, open new tab, radio button, checkbox, upload file, delete file. |
|  | Web application allows the customer to manage the project and its functionality. | Allows customers to manage projects according to functions and each function includes a test case file. |
|  | Web application allows to manage the input file. | Each function will include an input file. Customers can download the sample template file to their personal computer (containing sample data and instruction on random values).  Customers can upload and download .CSV files to the website. |
|  | Web application for customer manage test process. | Allow customer schedule automatic execution for each .csv file, review results of each run, export report and display log.  Can be executed automatically by schedule or manually.  + Allow manual test execution with UI less browser. |

## System Requirement

### Actor

*Table 3.2 Actor*

|  |  |  |
| --- | --- | --- |
| **#** | **Name** | **Description** |
| 1 | Unauthenticated user | User can register and log in to the system |
| 2 | Customer | The user login to the system as a customer. who can manage projects, manage functions, manage input files and execute test automation |

### Use Case

#### Use Case Diagram

Diagram, schematic

Description automatically generated

Figure 3.1 Use Case Diagram

*Table 3.3 Use Case*

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | ***Main Function*** | ***Main Use Case*** | **ID Use Case** |
| 1 | Authentication | Register | UC\_1.1 |
| Login | UC\_1.2 |
| 2 | Manage Project | Add Project | UC\_2.1 |
| Delete Project | UC\_2.2 |
| Edit Project | UC\_2.3 |
| View Project Detail | UC\_2.4 |
| 3 | Manage Project Function | Add Function | UC\_3.1 |
| Edit Function | UC\_3.2 |
| Delete Function | UC\_3.3 |
| View Function Detail | UC\_3.4 |
| Schedule Test | UC\_3.5 |
| Send Report to Mail of Schedule Test | UC\_3.6 |
| 4 | Manage Test of Function | Import File | UC\_4.1 |
| Download File | UC\_4.2 |
| Delete File | UC\_4.3 |
| View Test Screen | UC\_4.4 |
| Run Test | UC\_4.5 |
| View Logs Step | UC\_4.6 |
| Download File Template | UC\_4.7 |

#### Use Case Description

##### UC\_1.1 Register

A picture containing application

Description automatically generated

Figure 3.2 UC\_1.1 Register

*Table 3.4 UC\_1.1 Register*

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE-UC\_1.1** | | | |
| **Use Case No.** | UC\_1.1 | **Use Case Version** | 1.0 |
| **Use Case Name** | Register | | |
| **Author** | Pham Thi Thanh Hang | | |
| **Priority** | High | | |
| **Actor:**  - Unauthenticated user  **Summary:**  - Allow a guest to register an account.  **Goal:**  - A guest can create an account with user role.  **Triggers:**  - A guest sends register command by click to register button at homepage.  **Preconditions:**  **-** A guest is required to have a new email address.  **Post Conditions:**  **- Success:** A new account is created.  **- Fail:** System shows error message.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | A guest enters email. |  | | **2** | A guest enters password. |  | | **3** | A guest confirm password. |  | | **4** | A guest clicks to “Register” button. | System creates a new account with the input information with user role.  System redirect to the login page. |   **Alternative Scenario: N/A**  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** |  | System shows message "Please check your internet connection" when the internet is lost. | | **2** | A guest uses the exist email before. | System shows message "Email is already used by another user. Please choose another email." | | **3** | A guest enters wrong format email. | System shows message “The format email is wrong, try it again!” | | **4** | A guest uses password’s length which is less than 6 characters or not contain uppercase character or not contain lower case character. | System shows message "Password’s length is at least 6 characters and contain at least 1 uppercase character and at least one lower case character". | | **5** | User confirm wrong password. | System shows message “Please confirm password again.” |   **Business Rules:**   * Password: password’s length is at least 6 characters and contain at least 1 uppercase charcter and at least one lower case character to ensure security. * System must ensure that no one can read user password. * The password must be encrypted in database. * Password field and confirm password field match together. * Email field must be right format. | | | |

##### UC\_1.2 Login

A white circle with a black background

Description automatically generated with low confidence

Figure 3.3 UC\_1.2 Login

*Table 3.5 UC\_1.2 Login*

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE-UC\_1.2** | | | |
| **Use Case No.** | UC\_1.2 | **Use Case Version** | 1.0 |
| **Use Case Name** | Login | | |
| **Author** | Pham Thi Thanh Hang | | |
| **Priority** | High | | |
| **Actor:**   * Unauthenticated user   **Summary:**   * Allow a guest log in to the system.   **Goal:**   * A guest can log in to the system with specific role.   **Triggers:**   * A guest sends login command by click to login button at homepage.   **Preconditions:**   * A guest completed register account successfully.   **Post Conditions:**  **- Success:** A guest is exactly authorized with specific role in the system.  **- Fail:** System shows error messages  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | A guest provides email. |  | | **2** | A guest provides password. |  | | **3** | A guest clicks “Login” button. | System redirect to the main page. A guest logins into system with their specific role. |   **Alternative Scenario: N/A**  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** | Guest enters wrong identity information. | System shows an error message “Wrong username or password. Please try again.” | | **2** | Guest enters nothing in email and password fields. | System shows “The email field is required” and “The password field is required”. | | **3** | Guest enters information of email and nothing of password. | System shows “The password field is required”. | | **4** | Guest enters nothing of email and information of password. | System shows “The email field is required” | | **5** |  | System shows message "Please check your internet connection" when the internet is lost. |   **Business Rules:**   * Username: must be not empty. * Password: must be not empty. | | | |

##### UC\_2.1 Add Project

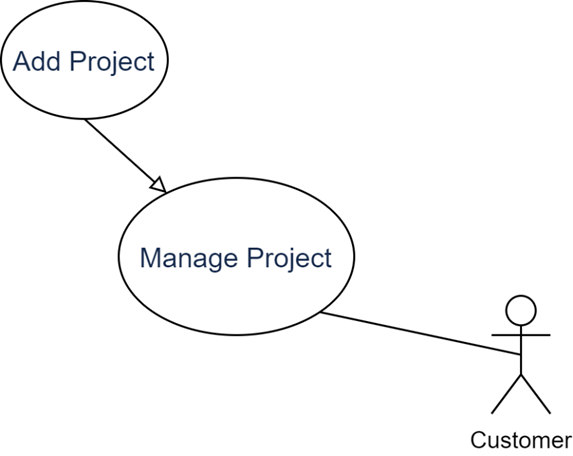


Figure 3.4 UC\_2.1 Add Project

*Table 3.6 UC\_2.1 Add Project*

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE-UC\_2.1** | | | |
| **Use Case No.** | UC\_2.1 | **Use Case Version** | 1.0 |
| **Use Case Name** | Add Project | | |
| **Author** | Pham Thi Thanh Hang | | |
| **Priority** | High | | |
| **Actor:**   * Customer   **Summary:**   * User creates project with URL respectively.   **Goal:**   * User creates project to manage every project easily.   **Triggers:**   * User clicks button “Add Project”.   **Preconditions:**   * User access “Project” page   **Post Conditions:**  **- Success:** A project is created successfully  **- Fail:** System shows error message.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | User clicks button “Add Project”. | System shows pop up “Add new project”. | | **2** | User enters name project at “Project Name”. |  | | **3** | User enters link project at “Project Name”. |  | | **4** | User clicks button “Submit” project at “Project Name”. | The pop up close. A new line is added to “Project” page. |   **Alternative Scenario: N/A**  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** |  | System shows message "Please check your internet connection" when the internet is lost. | | **2** | User clicks button “Cancel” of pop up. | The pop up close. No line is added to “Project” page. | | **3** | User enters same name with other name projects. | System shows message “The name project is existed. Try it again”. | | **4** | User enters a large character at textbox name and link. | System shows message “The project is invalid. Try it again”. | | **5** | User enters nothing at textbox name and link. | System shows message “The project is invalid. Try it again”. |   **Business Rules:**  **- C**onstraint textbox enter invalid character such as: special, large, space character. | | | |

##### UC\_2.2 Delete Project

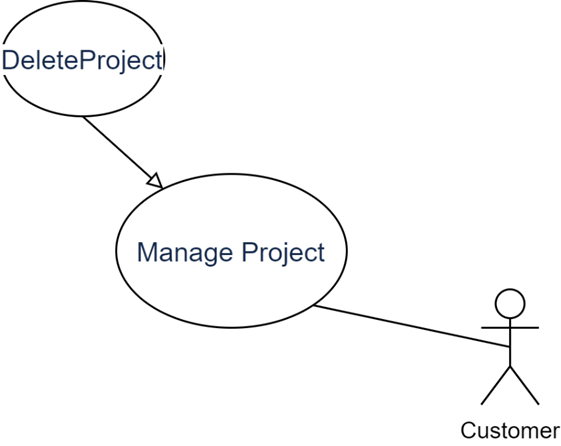


Figure 3.5 UC\_2.2 Delete Project

*Table 3.7 UC\_2.2 Delete Project*

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE-UC\_2.2** | | | |
| **Use Case No.** | UC\_2.2 | **Use Case Version** | 1.0 |
| **Use Case Name** | Delete Project | | |
| **Author** | Pham Thi Thanh Hang | | |
| **Priority** | High | | |
| **Actor:**  - Customer  **Summary:**  - User can delete unnecessary project.  **Goal:**  - User delete the unused project.  **Triggers:**  - User click button “Delete”.  **Preconditions:**  -User have project before.  **Post Conditions:**  **- Success:** The project is removed from list project.  **- Fail:** System shows error message.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | User clicks button “Delete” in project page. | System shows delete pop up “Do you want to delete project:…?”. | | **2** | User clicks button “Detele” in pop up. | System shows “Delete successfully”. The pop up close. The project is removed from list project. |   **Alternative Scenario: N/A**  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** | User clicks button “Cancel” in delete pop up. | The pop up close. The project is removed from list project. | |  |  | System shows message "Please check your internet connection" when the internet is lost. |   **Business Rules:**  **-** Delete action cannot remove completely, it is just hidden. | | | |

##### UC\_2.3 Edit Project

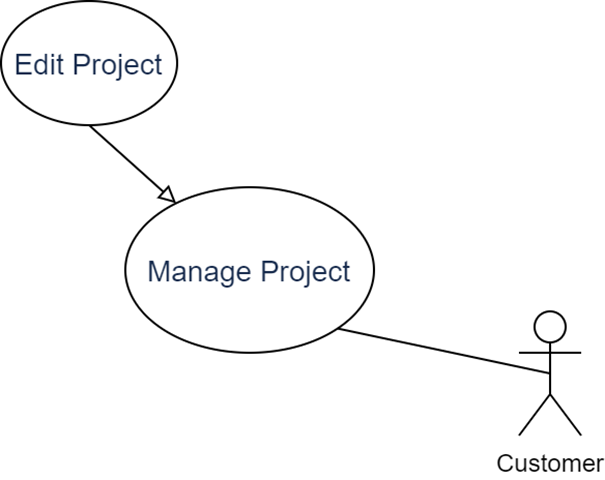


Figure 3.6 UC\_2.3 Edit Project

*Table 3.8 UC\_2.3 Edit Project*

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE-UC\_2.3** | | | |
| **Use Case No.** | UC\_2.3 | **Use Case Version** | 1.0 |
| **Use Case Name** | Edit Project | | |
| **Author** | Pham Thi Thanh Hang | | |
| **Priority** | High | | |
| **Actor:**  - Customer  **Summary:**  - User update information project.  **Goal:**  - User want to change information project.  **Triggers:**  - User click button “Edit”.  **Preconditions:**  **-** User have project before.  **Post Conditions:**  **- Success:** The project is updated in list project.  **- Fail:** System shows error message.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | User clicks button “Edit”. | System shows pop up “Edit project” | | **2** | User enters name project at “Project Name”. |  | | **3** | User enters link project at “Project Name”. |  | | **4** | User clicks button “Submit” project at “Project Name”. | The pop up close. A updated line is sort at the top of table to “Project” page. |   **Alternative Scenario: N/A**  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** |  | System shows message "Please check your internet connection" when the internet is lost. | | **2** | User clicks button “Cancel” of pop up. | The pop up close. No line is added to “Project” page. | | **3** | User enters same name with other name projects. | System shows message “The name project is existed. Try it again”. | | **4** | User enters a large character at textbox name and link. | System shows message “The project is invalid. Try it again”. | | **5** | User enters nothing at textbox name and link. | System shows message “The project is invalid. Try it again”. |   **Business Rules:**  - The updated information is sorted at the top of list project. | | | |

##### UC\_2.4 View Project Detail

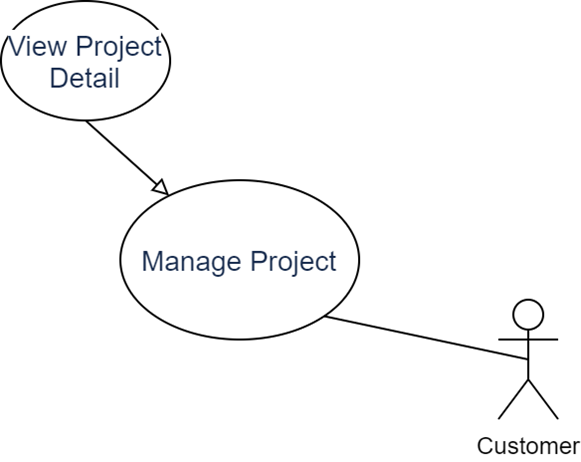


Figure 3.7 UC\_2.4 View Project Detail

*Table 3.9 UC\_2.4 View Project Detail*

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE-UC\_2.4** | | | |
| **Use Case No.** | UC\_2.4 | **Use Case Version** | 1.0 |
| **Use Case Name** | View Project Detail | | |
| **Author** | Pham Thi Thanh Hang | | |
| **Priority** | High | | |
| **Actor:**  - Customer  **Summary:**  - User view detail all function belong to this project.  **Goal:**  - User want to redirect “Function” page.  **Triggers:**  - User click button “Detail” page.  **Preconditions:**  **-** User have project before.  **Post Conditions:**  **- Success:** The project is redirected to “Function” page.  **- Fail:** System shows error message.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | User clicks button “Detail”. | System redirect to “Function” page. |   **Alternative Scenario: N/A**  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** |  | System shows message "Please check your internet connection" when the internet is lost. |   **Business Rules:N/A** | | | |

##### UC\_3.1 Add Function

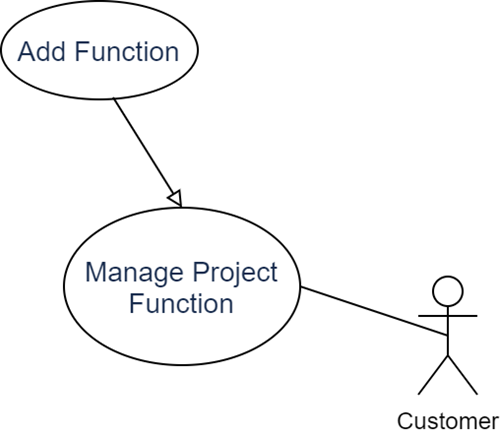


Figure 3.8 UC\_3.1 Add Function

*Table 3.10 UC\_3.1 Add Function*

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE-UC\_3.1** | | | |
| **Use Case No.** | UC\_3.1 | **Use Case Version** | 1.0 |
| **Use Case Name** | Add Function | | |
| **Author** | Bui Thi Hong Nhung | | |
| **Priority** | Medium | | |
| **Actor:**   * Customer   **Summary:**   * User creates function with name respectively.   **Goal:**   * User creates function to manage every function easily.   **Triggers:**   * User wants to create a function and click button “Add Function”.   **Preconditions:**   * User has project before. * User access “Manage Project Function” page.   **Post Conditions:**  **- Success:** A function is created successfully.  **- Fail:** System shows error message.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | User clicks button “Add Function”. | System shows pop up “Add new function”. | | **2** | User enters name function at “Function Name”. |  | | **3** | User clicks name project at “Submit”. | The pop up close. A new line is added to “Manage Project Function” page. |   **Alternative Scenario: N/A**  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** |  | System shows message "Please check your internet connection" when the internet is lost. | | **2** | User clicks button “Cancel” of pop up. | The pop up close. No line is added to “Manage Project Function” page. | | **3** | User enters same name with others name function. | System shows message “The name function is existed. Try it again”. | | **4** | User enters a large character at textbox name. | System shows message “The function is invalid. Try it again”. | | **5** | User enters nothing at textbox name. | System shows message “The function is invalid. Try it again”. |   **Business Rules:**  **-** Constraint textbox enter invalid character such as: special, large, space character. | | | |

##### UC\_3.2 Edit Function

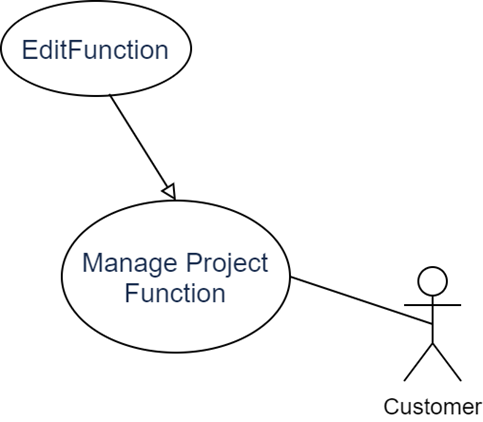


Figure 3.9 UC\_3.2 Edit Function

*Table 3.11 UC\_3.2 Edit Function*

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE-UC\_3.1** | | | |
| **Use Case No.** | UC\_3.1 | **Use Case Version** | 1.0 |
| **Use Case Name** | Edit Function | | |
| **Author** | Bui Thi Hong Nhung | | |
| **Priority** | Must have | | |
| **Actor:**   * Customer   **Summary:**   * User edit function name respectively.   **Goal:**   * User edit function to manage every function easily.   **Triggers:**   * User want to edit function and click “Edit” button.   **Preconditions:**   * User have project before. * User have function before.   **Post Conditions:**  **- Success:** The function is updated in list project functions.  **- Fail:** System shows error message.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** |  | System shows project functions list at Manage Project functions. | | **2** | User select any function click on “Edit” button. | System shows pop up “Edit function”. | | **3** | User enters name function at “Function Name”. |  | |  | User clicks a “Submit” button. | The pop up close. A line is update to “Manage Project Function” page. |   **Alternative Scenario: N/A**  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** |  | System shows message "Please check your internet connection" when the internet is lost. | | **2** | User clicks button “Cancel” of pop up. | The pop up close. No line is added to “Manage Project Function” page. | | **3** | User enters the same name with other name function. | System shows message “The name function is existed. Try it again”. | | **4** | User enters a large character at textbox name. | System shows message “The function is invalid. Try it again”. | | **5** | Users enter nothing at textbox name. | System shows message “The function is invalid. Try it again”. |   **Business Rules:**  **-** Constraint textbox enter invalid character such as: special, large, space character. | | | |

##### UC\_3.3 Delete Function

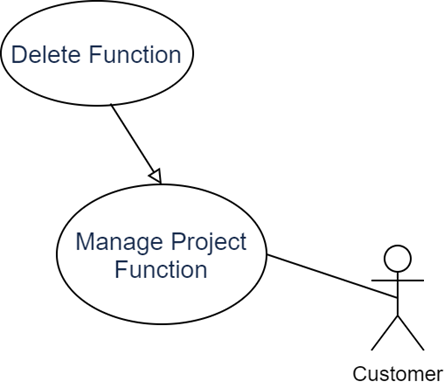


Figure 3.10 UC\_3.3 Delete Function

*Table 3.12 UC\_3.3 Delete Function*

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE-UC\_3.1** | | | |
| **Use Case No.** | UC\_3.1 | **Use Case Version** | 1.0 |
| **Use Case Name** | Delete Function | | |
| **Author** | Bui Thi Hong Nhung | | |
| **Priority** | Must have | | |
| **Actor:**   * Customer   **Summary:**   * User delete function respectively.   **Goal:**   * User delete function to manage every function easily.   **Triggers:**   * User want to delete function and click “Delete” button.   **Preconditions:**   * User have project before. * User have function before.   **Post Conditions:**  **- Success:** The function is removed from the list project functions.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** |  | System shows project functions list at Manage Project functions. | | **2** | User select any function click on “Delete” button. | System delete function is removed from the list project functions. |   **Alternative Scenario: N/A**  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** |  | System shows message "Please check your internet connection" when the internet is lost. |   **Business Rules:** | | | |

##### UC\_3.4 View Function Detail

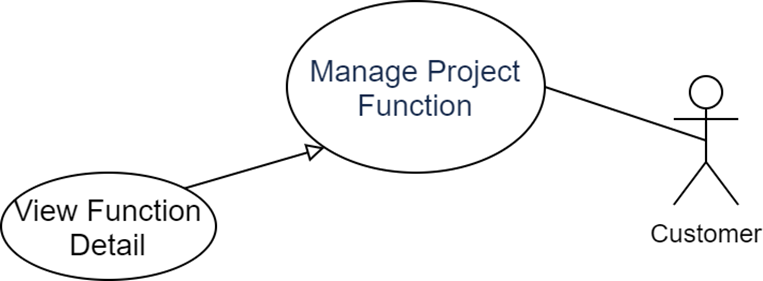


Figure 3.11 UC\_3.4 View Function Detail

*Table 3.13 UC\_3.4 View Function Detail*

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE-UC\_3.4** | | | |
| **Use Case No.** | UC\_3.4 | **Use Case Version** | 1.0 |
| **Use Case Name** | View Function Detail | | |
| **Author** | Pham Thi Thanh Hang | | |
| **Priority** | High | | |
| **Actor:**  - Customer  **Summary:**  - User view all testcase file belong to this function  **Goal:**  - User manage all testcase file to check this function  **Triggers:**  - User click “Detail” button in “Project Detail” page  **Preconditions:**  -User have project before  -User have function project before.  **Post Conditions:**  **- Success:** The function project is redirected to “Function Testing” page  **- Fail:** System shows error message  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | User clicks button “Detail” in “Manage Project functions” page. | System redirects to “Manage File Test of Function” page. |   **Alternative Scenario: N/A**  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** |  | System shows message "Please check your internet connection" when the internet is lost. |   **Business Rules: N/A** | | | |

##### UC\_3.5 Schedule Test

Graphical user interface, application

Description automatically generated

Figure 3.12 UC\_3.5 Schedule Test

*Table 3.14 UC\_3.5 Schedule Test*

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE-UC\_3.5 | | | |
| Use Case No. | UC\_3.5 | Use Case Version | 1.0 |
| Use Case Name | Schedule Test | | |
| Author | Pham Thi Thanh Hang | | |
| Priority | High | | |
| Actor:  - Customer  Summary:  - User set time to run automation background  Goal:  - User want to receive result which don’t need open browser  Triggers:  - User click turn switch “Schedule Test” on  Preconditions:  -User have project before  -User have function project before  -User have csv file before  - User access “Project Detail” page  Post Conditions:  - Success: The schedule test is saved and run on time set  - Fail: System shows error message  Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User click turn switch “Schedule Test” on | The switch is turn on. On time job background will run follow config |   Alternative Scenario: N/A  Exceptions: N/A  Business Rules:  -Help user do not take time run manual test again | | | |

##### UC\_3.6 Send mail

Graphical user interface, application

Description automatically generated

Figure 3.13 UC\_3.6 Send mail

*Table 3.15 UC\_3.6 Send mail*

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE-UC\_3.6** | | | |
| **Use Case No.** | UC\_3.6 | **Use Case Version** | 1.0 |
| **Use Case Name** | Send Mail | | |
| **Author** | Bui Thi Hong Nhung | | |
| **Priority** | Must Have | | |
| **Actor:**  - Customer  **Summary:**  - User receive report mail from system after done each schedule test  **Goal:**  - User controls the performance of the function  **Triggers:**  - User clicks button “Run”.  - User turn on “Schedule test” for function  **Preconditions:**  **-** User have project before.  - User have function project before.  - User have a CSV file before.  - User access “Test Screen” page.  **Post Conditions:**  **- Success:** User receive report mail notification report result  **- Fail:** User don’t receive report mail  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** |  | System shows project functions list at Manage Project functions. | | **2** | User select any function turn on “Schedule test” switch or click button “Run”. | System will auto schedule run test and send report to mail of user after done. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** |  | System shows message "Please check your internet connection" when the internet is lost. | | **2** | Select any project but don’t have function. | No data, system don’t schedule run test. |   **Business Rules:**  **-** Email of user is correct and active | | | |

##### UC\_4.1 Import File

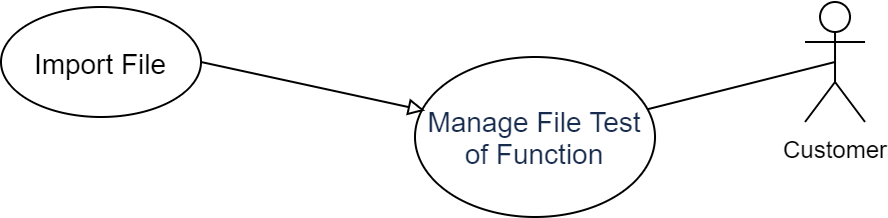


Figure 3.14 UC\_4.1 Import File

*Table 3.16 UC\_4.1 Import File*

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE-UC\_4.1** | | | |
| **Use Case No.** | UC\_4.1 | **Use Case Version** | 1.0 |
| **Use Case Name** | Import File | | |
| **Author** | Pham Thi Thanh Hang | | |
| **Priority** | High | | |
| **Actor:**  - Customer  **Summary:**  - User import file csv to start check website  **Goal:**  - User use input file to start schedule automation check website  **Triggers:**  - User click button “Import”  **Preconditions:**  -User have project before.  -User have function project before.  - User access “Function Testing” page.  **Post Conditions:**  **- Success:** The csv file is uploaded successfully.  **- Fail:** System shows error message.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | User clicks button “Import” in “Function Testing” page. | System show pop up “Upload File”. | | **2** | User clicks button “Choose File”. | The file explorer is show. | | **3** | User chose csv file size 5Mb. | The pop-up show information of this csv file such as: name, type, size, date modified. | | **4** | User clicks button “Upload this”. | After that, this button shows “Upload 100%”. | | **5** | User clicks button “Close”. | The pop up close. A new file is show in “Function Testing” page. |   **Alternative Scenario: N/A**  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** | User upload file non csv file. | System show “Please only upload csv file”. | | **2** | User clicks button “Close” in pop up “Upload File”. | The pop up close. No line is not added in “Function Testing” page | | **3** | User upload size file over 5Mb. | System show “Please upload file size around 5Mb”. | | **4** |  | System shows message "Please check your internet connection" when the internet is lost. |   **Business Rules:**  - Not allow upload file non csv file.  - Not allow upload size file over 5Mb. | | | |

##### UC\_4.2 Download File

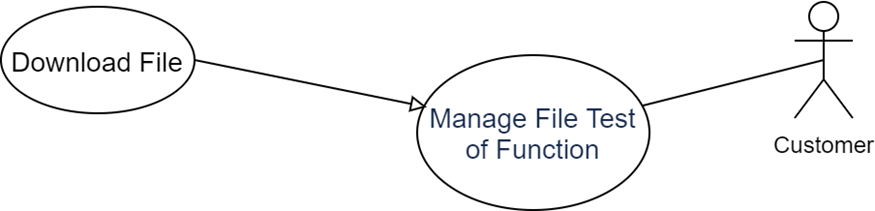


Figure 3.15 UC\_4.2 Download File

*Table 3.17 UC\_4.2 Download File*

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE-UC\_4.2** | | | |
| **Use Case No.** | UC\_4.2 | **Use Case Version** | 1.0 |
| **Use Case Name** | Download File | | |
| **Author** | Pham Thi Thanh Hang | | |
| **Priority** | High | | |
| **Actor:**  - Customer  **Summary:**  - User download csv file.  **Goal:**  - User can download csv to reuse for other project.  **Triggers:**  - User click button “Download” in “Function Testing” page.  **Preconditions:**  -User have project before.  -User have function project before.  -User have csv file before.  - User access “Function Testing” page.  **Post Conditions:**  **- Success:** The csv file is downloaded successfully.  **- Fail:** System shows error message.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | User select any file csv and click button “Download” in “Function Testing” page. | System show “Download file successfully”. |   **Alternative Scenario: N/A**  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** |  | System shows message "Please check your internet connection" when the internet is lost. |   **Business Rules:**  - File download have location Download folder | | | |

##### UC\_4.3 Delete File

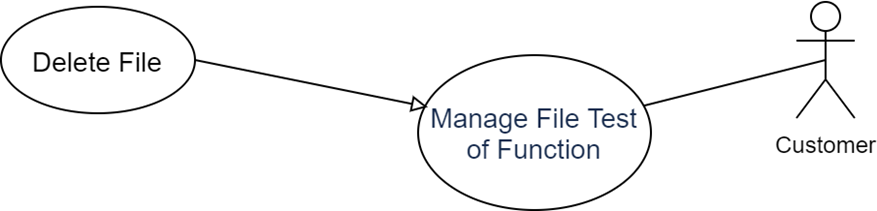


Figure 3.16 UC\_4.3 Delete File

*Table 3.18 UC\_4.3 Delete File*

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE-UC\_4.3** | | | |
| **Use Case No.** | UC\_4.3 | **Use Case Version** | 1.0 |
| **Use Case Name** | Delete File | | |
| **Author** | Pham Thi Thanh Hang | | |
| **Priority** | High | | |
| **Actor:**  - Customer  **Summary:**  - User delete csv file  **Goal:**  - User do not need use this csv  **Triggers:**  - User click button “Delete”  **Preconditions:**  -User have project before.  -User have function project before.  -Users have csv file before.  - User access “Function Testing” page.  **Post Conditions:**  **- Success:** The csv file is removed successfully.  **- Fail:** System shows error message.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | User clicks button “Delete” in “Function Testing” page. | System show pop up “Do you want to delete this file?”. | | **2** | User clicks button “Yes”. | System show “Delete successfully”. A line of this csv is deleted in “Function Testing” page |   **Alternative Scenario: N/A**  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** | User clicks button “No” pop up. | The pop up is closed. | | **2** |  | System shows message "Please check your internet connection" when the internet is lost. |   **Business Rules:**  - Should be ask again before to delete action | | | |

##### UC\_4.4 View Test Screen



Figure 3.17 UC\_4.4 View Test Screen

*Table 3.19 UC\_4.4 View Test Screen*

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE-UC\_4.4** | | | |
| **Use Case No.** | UC\_4.4 | **Use Case Version** | 1.0 |
| **Use Case Name** | View Test Screen | | |
| **Author** | Pham Thi Thanh Hang | | |
| **Priority** | High | | |
| **Actor:**  - Customer  **Summary:**  - User can view all information csv file and activity automation test.  **Goal:**  - User view activity automation test.  **Triggers:**  - User click button “Detail” in “Function Testing” page.  **Preconditions:**  -User have project before.  -User have function project before.  -User have csv file before.  - User access “Function Testing” page.  **Post Conditions:**  **- Success:** The system redirect to “Test Screen”.  **- Fail:** System shows error message.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | User clicks button “Detail” in “Function Testing” page. | System redirects to “Test Screen” page. All data of csv file is showed in “Test Screen” page. |   **Alternative Scenario: N/A**  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** |  | System shows message "Please check your internet connection" when the internet is lost. |   **Business Rules:**  -Show all information of csv file to user easy to follow steps | | | |

##### UC\_4.5 Run Test

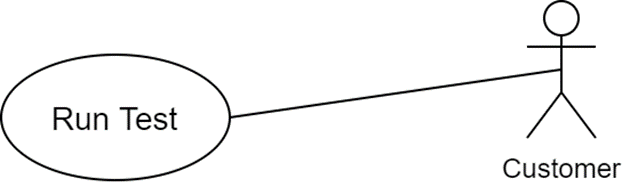


Figure 3.18 UC\_4.5 Run Test

*Table 3.20 UC\_4.5 Run Test*

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE-UC\_4.5** | | | |
| **Use Case No.** | UC\_4.5 | **Use Case Version** | 1.0 |
| **Use Case Name** | Run Test | | |
| **Author** | Pham Thi Thanh Hang | | |
| **Priority** | High | | |
| **Actor:**  - Customer  **Summary:**  - User run csv file to start automation test.  **Goal:**  - User run automation test to get result.  **Triggers:**  - User click button “Run”.  **Preconditions:**  **-**User have project before .  -User have function project before.  -User have csv file before.  - User access “Test Screen” page.  **Post Conditions:**  **- Success:** The system return report result.  **- Fail:** System shows error message.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | User clicks button “Run” in “Test Screen” page. | System open new browser to execute action in csv file.  Log file show all steps execute.  After execute completely, this browser close.  Log file show result test how many pass/fail.  Then report is sent to customer via mail. |   **Alternative Scenario: N/A**  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** | User upload file is not right format | System show “Please check format file. You can download template file above.” | | **2** |  | System shows message "Please check your internet connection" when the internet is lost. |   **Business Rules:**  - Check location element is invalid | | | |

##### UC\_4.6 View Log Steps

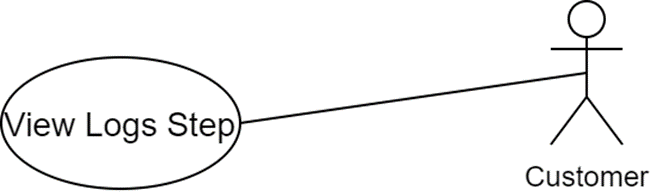


Figure 3.19 UC\_4.6 View Log Steps

*Table 3.21 UC\_4.6 View Log Steps*

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE-UC\_4.6** | | | |
| **Use Case No.** | UC\_4.6 | **Use Case Version** | 1.0 |
| **Use Case Name** | View Log Steps | | |
| **Author** | Pham Thi Thanh Hang | | |
| **Priority** | High | | |
| **Actor:**  - Customer  **Summary:**  - User can view every step execute automation test.  **Goal:**  - User can observe steps run right ways.  **Triggers:**  - User click button “Run”.  **Preconditions:**  **-**User have project before.  -Users have function project before.  -Users have csv file before.  - User access “Test Screen” page.  **Post Conditions:**  **- Success:** The system return steps log and total result.  **- Fail:** System shows error message.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | User clicks button “Run” in “Test Screen” page. | Log file show all steps execute.  After execute completely, this browser close.  Log file show result test how many pass/fail. |   **Alternative Scenario: N/A**  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** | User upload file is not right format | System show “Please check format file. You can download template file above.” | | **2** |  | System shows message "Please check your internet connection" when the internet is lost. |   **Business Rules:**  - File log help user know step where stuck having problem. | | | |

##### UC\_4.7 Download File Template.

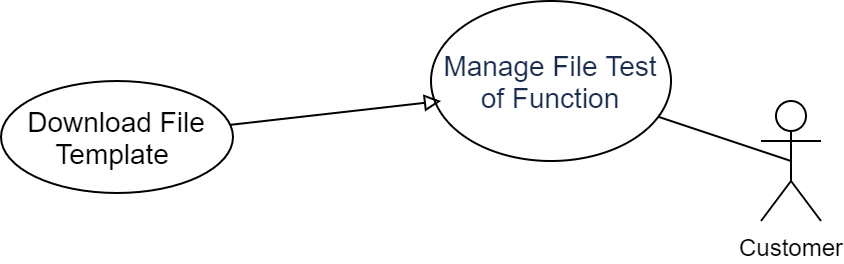


Figure 3.20 UC\_4.7 Download File Template

*Table 3.22 UC\_4.7 Download File Template*

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE-UC\_4.7** | | | |
| **Use Case No.** | UC\_4.7 | **Use Case Version** | 1.0 |
| **Use Case Name** | Download File template | | |
| **Author** | Pham Thi Thanh Hang | | |
| **Priority** | High | | |
| **Actor:**  - Customer  **Summary:**  -User download guideline csv file.  **Goal:**  - User know guideline to input right data in csv file.  **Triggers:**  - User click “here” link at the top of “Function Test” page.  **Preconditions:**  **-**User have project before.  - User access “Function Test” page.  **Post Conditions:**  **- Success:** The system return steps log and total result.  **- Fail:** System shows error message.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | At the top of “Function Test” page, the sentence “You can try our template test case file in here”, user click “here” | The template csv file is downloaded successfully. |   **Alternative Scenario: N/A**  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** |  | System shows message "Please check your internet connection" when the internet is lost. |   **Business Rules:**  - The template describe to use easily for user. | | | |

## Non-Functional Requirement

*Table 3.23 Non-Functional Requirement*

|  |  |  |
| --- | --- | --- |
| **#** | **Attribute** | **Description** |
|  | Usability | The website’s UI is fit for the Chrome browser on each device.  Easy to use. |
|  | Reliability | The information stored in the database is permanent. |
|  | Availability | Use the "extent report" library to generate a report file.  Use Selenium Java to support Headless browser and automation UI less browser testing |
|  | Maintainability | The system is divided into separate component for easy maintenance. |

## Entity-Relationship Diagram

Diagram, schematic

Description automatically generated

Figure 3.21 Entity-Relationship Diagram

# SYSTEM DESIGN & IMPLEMENTATION

## File Test Design

### File Input Design

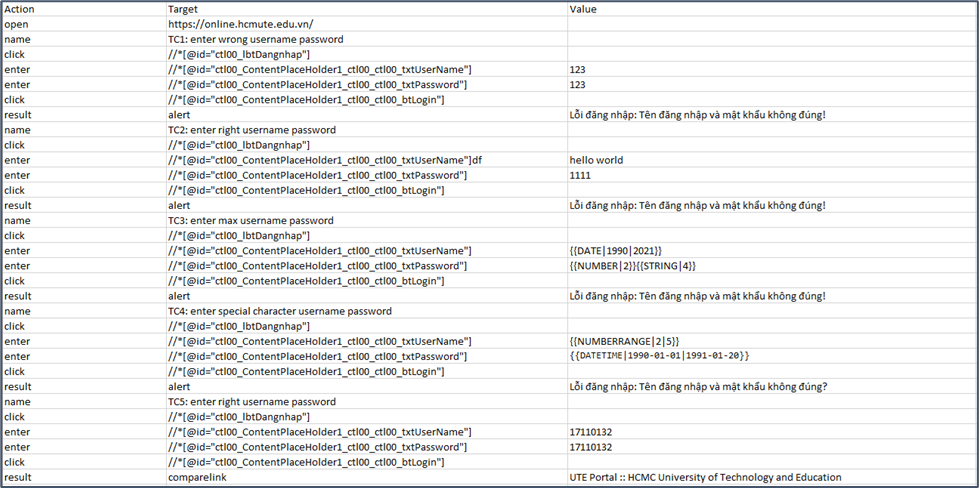


Figure 4.1 File Input Design

File Input: the column includes the contents:

* Action: actions on the screen include open, name, enter, click, result...
* Target: the locations of the element XPath include URL, test case name, Xpath...
* Value: the value passed on input include value, expected result...

The Action column includes the keywords:

*Table 4.1 Keyword of Action column*

|  |  |
| --- | --- |
| **Keywords** | **Description** |
| open | This keyword represents the action of open browser going to URL. |
| name | This keyword represents the name of the testcase. |
| enter | This keyword represents the action of clicking on the element. |
| dropdown | This keyword defines the action of entering information into element. |
| newtab | This keyword represents the action open a new tab. |
| result | This keyword is responsible for comparing actual vs expect results. |

The Target column includes the keywords:

*Table 4.2 Keyword of Target column*

|  |  |
| --- | --- |
| **Keywords** | **Description** |
| alert | This keyword is responsible for getting the value in the alert dialog box. |
| comparelink | This keyword is responsible for getting the title value of the website. |
| <url> | The user enters the URL of the website he wants to execute the test.  For example: https://online.hcmute.edu.vn/ |
| <xpath> | The user enters element of the Xpath.  For example: //\*[@id="ctl00\_lbtDangnhap"] |
| <expected result> | The user needs to enter the expect result. |
| <TC:name testcase> | The user enters the test case name according to structure TC<number>: name of the test case.  For example: TC1: Login with wrong password. |

The Value column includes the keywords:

<value>: the user enters the value to be tested, or random values in the following format:

*Table 4.3 Keyword of Value column*

|  |  |
| --- | --- |
| **Keywords** | **Description** |
| {{STRING}} | This keyword is used to generate a random string. |
| {{STRING|LEN}} | This keyword is used to generate a string of the corresponding length.  For example: {{STRING|5}} |
| {{NUMBER}} | This keyword is used to generate a string of numbers with random length. |
| {{NUMBER|LEN}} | This keyword is used to generate a string of numbers with the corresponding length.  For example: {{NUMBER|8}} |
| {{NUMBERRANGE|FROM|TO}} | This keyword is used to generate a random sequence of numbers between from and to respectively.  For example: {{NUMBERRANGE|2|5}} |
| {{DATE}} | This keyword is used to generate a random date 1-day value. |
| {{DATE|FROM|TO}} | This keyword is used to generate a random date in the interval between the respective dates.  For example: {{DATE|1990|2021}} |
| {{DATETIME|FROM|TO}} | This keyword is used to generate a string of the date and time between the corresponding date.  For example:  {{DATETIME|1990-01-01|1991-01-20}} |
| {{NUMBER|LEN}}{{STRING|LEN}} | This keyword is used to generate a random alphanumeric string with the corresponding length for each string.  For example:  {{NUMBER|2}}{{STRING|4}} |

### File Output Design

Design a report file to send to the email after the customer executes the test.

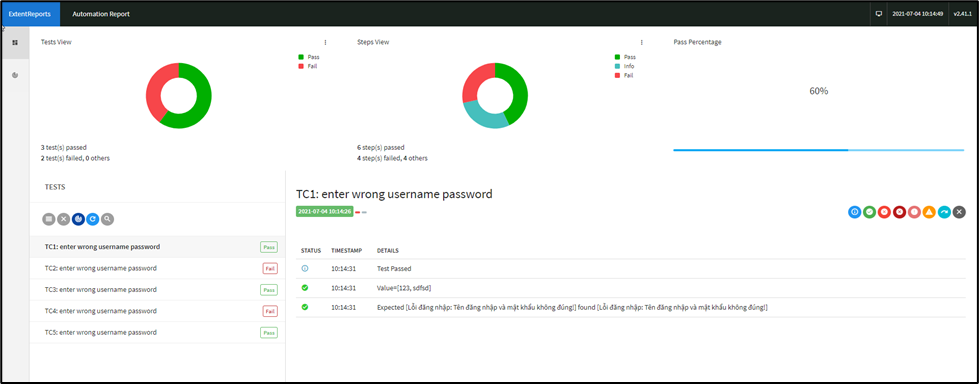


Figure 4.2 File Output Design

Test view: show the user the number of test cases passed and test cases failed.

Step view: show the user how many steps passed and how many steps failed.

Pass Percentage: show the user how many percent passed the result.

Testcase pass: show the user the test case pass information, the value passed, compare the actual results with the expected results.

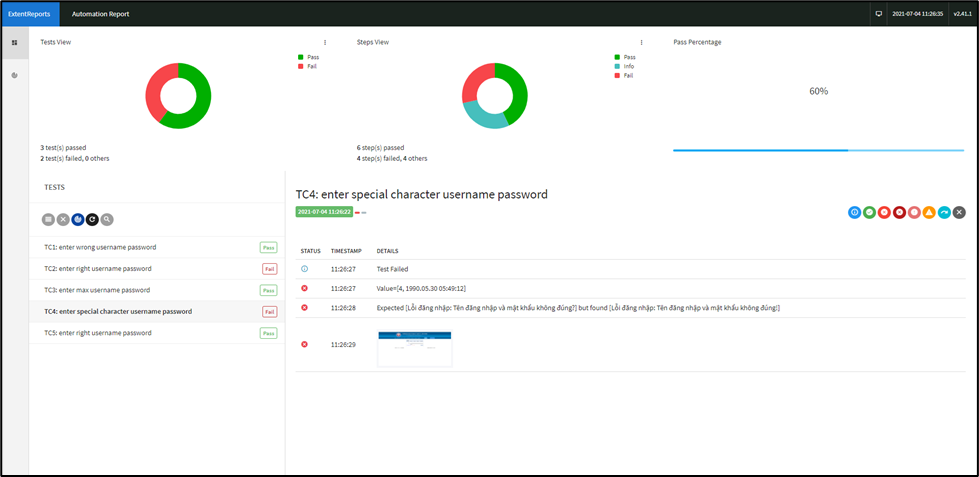


Figure 4.3 Testcase Fail.

Testcase failed: show the user the test case pass information, the value passed, compare the actual result with the expected result, the screenshot is wrong.

For example:

Testcase failed: wrong error message user entered incorrect Xpath.

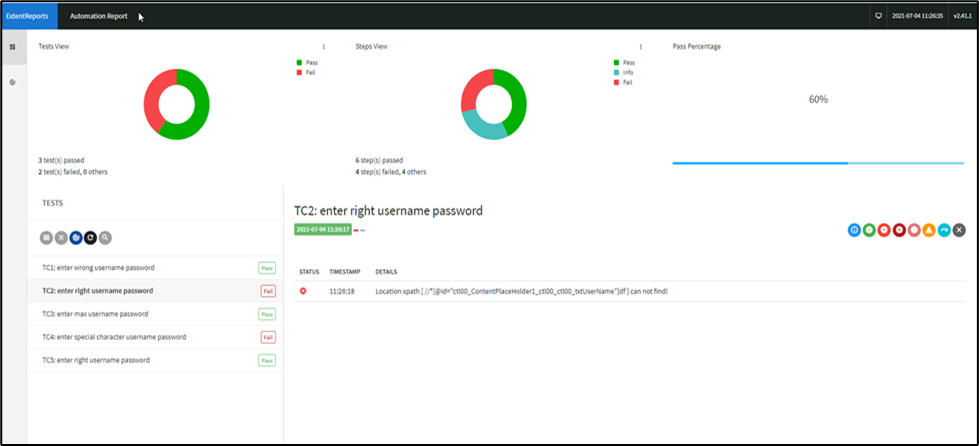


Figure 4.4 Testcase fail (example)

### File Log Design

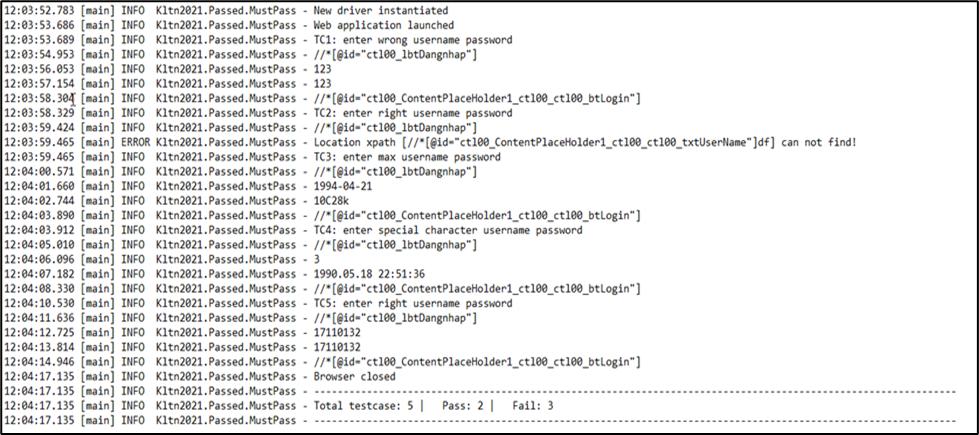


Figure 4.5 File Log Design.

The log file displayed to the user:

* Information about the execution time of the test case file.
* Actions such as initializing open a browser, test case name, values passed into the textbox, close browser.
* When the entire test case is executed, it will summarize the total number of test cases including the number of test cases passed and the number of failed test cases.

## System Architecture

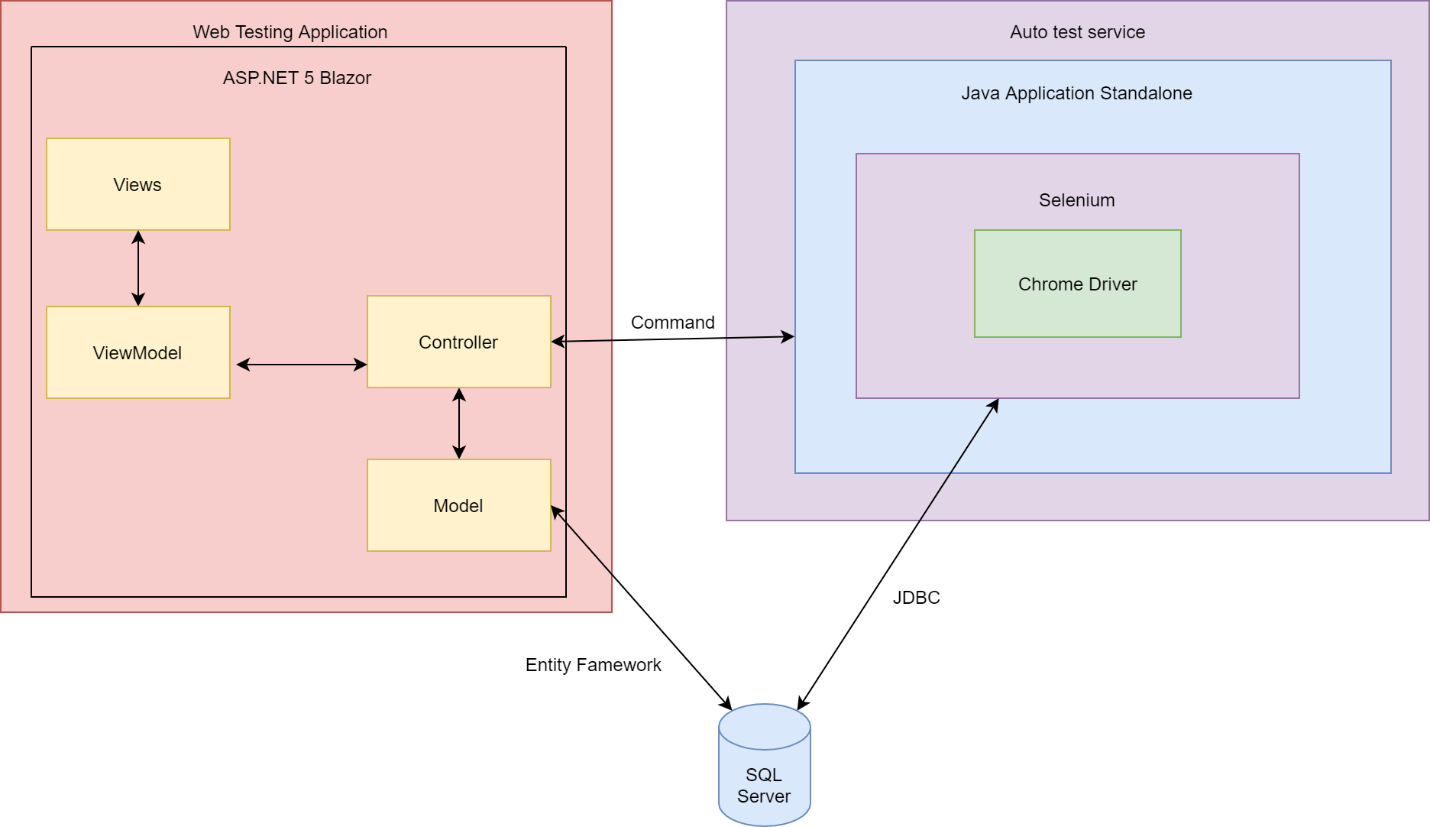


Figure 4.6 Architecture diagram for the system.

*Table 4.4 Description about architecture.*

|  |  |  |
| --- | --- | --- |
| # | **Component** | **Description** |
| 1 | Web Testing Application | It is responsible for processing interaction of the user through User web interface and call the test service to make it all functions work. |
| 2 | Auto test Service | It is responsible for execute these test case written on the user’s uploaded file and take the result to save and deliver for respective user. |

## System Component

### Web MVC Component Diagram

Chart, box and whisker chart

Description automatically generated

Figure 4.7 Web MVC Component diagram

## Class Diagram.

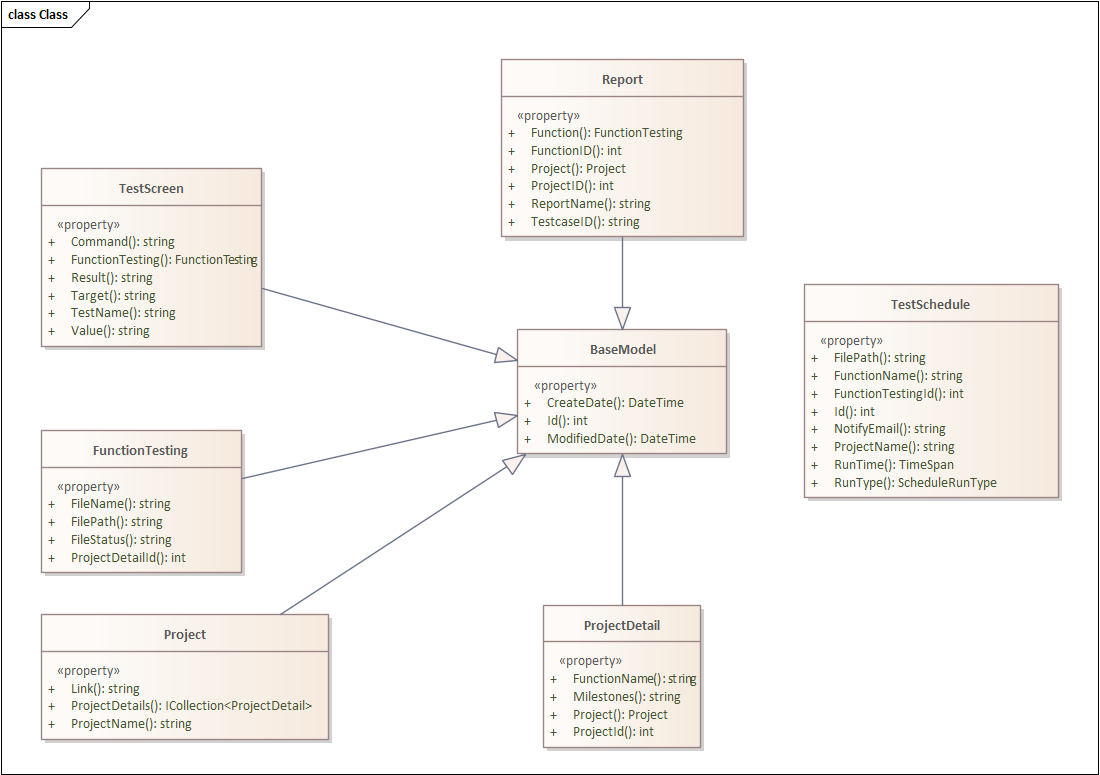


Figure 4.8 Class Diagram

### Class Description for Class BaseModel

*Table 4.5 Class description for class BaseModel*

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Access level** | **Description** |
| Id | int | public | Act as identity property of an object. |
| CreatedDate | datetime | public | The date of the record that was created in database. |
| ModifiedDate | datetime | public | The last date of the record was updated in database. |

### Class Description for Class Project

*Table 4.6 Class description for class Project*

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Access level** | **Description** |
| Link | String | public | Use for storing the link to the website’s project need to be automated test. |
| ProjectName | String | public | Use for storing name of the project. |

### Class Description for Class ProjectDetail

*Table 4.7 Class description for class ProjectDetail*

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Access level** | **Description** |
| ProjectId | Int | public | Id of its belonged Project. |
| FunctionName | String | public | Name of the testing function of the project’s website that. |
| Milestones | String | public | Defined milestones of the testing function. |

### Class Description for Class FunctionTesting

Table . Class description for class FunctionTesting

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Access level** | **Description** |
| ProjectDetailId | Int | public | Id of its belonged ProjectDetail. |
| Filename | String | public | The actual name of the user’s uploaded csv file that contains test case for the function. |
| FilePath | String | public | The physical path that the file saved in the system’s server. |
| FileStatus | String | public | The status of the file in server (found, not-found). |

### Class Description for Class TestScreen

Table . Class description for class TestScreen

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Access level** | **Description** |
| TestName | String | public | Name of the automation test session. |
| Target | String | public | Target of the automation test session. |
| Value | String | public | Value of the automation test session. |
| Command | String | public | The command used to guide automation test session to execute the test. |
| Result | String | public | The result of the test. |

### Class Description for Class Report

Table . Class description for class Report

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Access level** | **Description** |
| FunctionId | Int | public | Id of its belonged Function. |
| ProjectId | Int | public | Id of its belonged Project. |
| TestCaseId | Int | public | Id of its belonged TestCase. |
| ReportName | String | public | The name of testcase’s generated report after being executed. |

### Class Description for Class TestSchedule

This class is designed for the scan service for execute the scheduled test file.

Table . Class description for class TestSchedule

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Access level** | **Description** |
| Id | Int | public | Id of schedule. |
| FunctionName | String | public | Name of the being testing function. |
| FunctionTestingId | Int | public | Id of its belonged FunctionTesting. |
| ProjectName | String | public | Name of its belonged Project. |
| NotifyEmail | String | public | The email that will used to send result when the scheduled test was done. |
| RunTime | timespan | public | The time when the scheduled test will be executed. |
| RunType | Enum | public | The running type of the scheduled test (hourly, daily, monthly). |

## Database Diagram.

Diagram

Description automatically generated

Figure 4.8 Database Diagram.

*Table 4.12 Database Diagram Description.*

|  |  |
| --- | --- |
| **Table** | **Description** |
| Projects | Use to save project info. |
| FunctionTestings | Use to save testcase’s file info. |
| TestScreens | Use to save test case running and resulting info. |
| ProjectDetails | Use to save the project’s detail info. |
| Reports | Use to save the report of these testcases that had been done. |
| TestSchedules | Use to save the scheduler’s info for automation running. |

## Sequence Diagram.

### Login Sequence Diagram

Diagram, schematic

Description automatically generated

Figure 4.9 Login sequence diagram.

### Run Test Sequence Diagram

Diagram

Description automatically generated

Figure 4.10 Run test sequence diagram

### Import File Sequence Diagram

Diagram, schematic

Description automatically generated

Figure 4.11 Import file sequence diagram.

## Activity Diagram.

### Daily Schedule Test Activity Diagram

Diagram

Description automatically generated

Figure 4.12 Daily schedule test activity diagram

### One Time Schedule Test Activity Diagram

Diagram

Description automatically generated

Figure 4.13 One time schedule test activity diagram

## User Interface

### Screen Flow for General

*Table 4.13 Screen flow for general.*

|  |  |  |
| --- | --- | --- |
| No | Screen Name | Description |
| SCG001 | Login | The screen for login into the system. |
| SCG002 | Register Account | The screen for register account into the system. |

#### SCG001. Login

Figure 4.14 Login

*Table 4.14 Login description*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No. | Name | Format | Required | Reference | Note |
| 1 | Email | Input | True |  | The user can only input an email that the user created.  If user selected “remember me?” then system will display email.  If user don’t select “Remember me” then user will enter email. |
| 2 | Password | Input | True |  | Display the content as \*.  If user selected “remember me?” then system will display password.  If user don’t select “Remember me” then user will enter password. |
| 3 | Remember me | Checkbox |  |  | Save the login state of user that the user doesn’t need to perform login into the system. |
| 4 | Login | Button |  |  | When clicking, the system will check the data.  If invalid, then show error information.  If the data is valid, will redirect the user in the default page depend on role of user. |
| 5 | Register | Link page |  |  | Next to **SCG002.** Register Account. |
| 6 | Login | Link page |  |  | Reload **SCG001.** Login screen. |

#### SCG002. Register Account

Figure 4.15 Forgot password.

*Table 4.15 Register Account description*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No. | Name | Format | Required | Reference | Note |
| 1 | Email | Input | True |  | The user can only input an email that the user creates. |
| 2 | Password | Input | True |  | The user enter password with password include: 8 characters, number, word. Display the content as \*. |
| 3 | Confirm password | Input | True |  | The user confirm password. Display the content as \*. |
| 4 | Register | Button |  |  | When clicking, the system will check the data and save |

### Screen Flow for Customer

*Table 4.16 Screen flow for Customer*

|  |  |  |
| --- | --- | --- |
| No | Screen Name | Description |
| SCC001 | Home page | The screen display information product |
| SCC002 | Manage Projects | The screen shows a list of projects |
| SCC003 | Manage Project Functions | The screen shows a list of functions of a project |
| SCC004 | Manage File Test of Function | Displays list of input files of a function |
| SCC005 | Run Test | Screen displays list of test cases in the input file and displays log. |
| SCC006 | Report | Show result and report before done run test process |

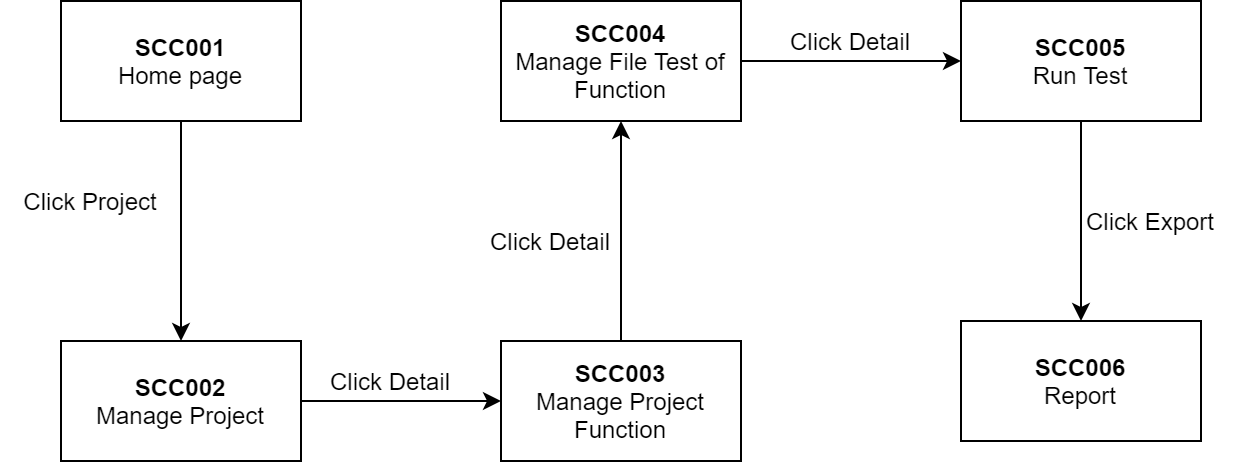


Figure 4.16 Screen flow for Customer

#### SCC001. Home page

Figure 4.17 SCC001. Home page

*Table 4.17 Home page description*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Name | Format | Reference | Note |
| 1 | Home page | Textbox |  | When click on “Homepage” will be reload Dashboard screen |
| 2 | Projects | Textbox |  | When click on “Projects” next to Management Project screen |
| 3 | Email user | Label |  | Display email user registered |
| 4 | Log out | Link page |  | Next to **SCG001** Login screen |

#### SCC002. Manage Projects

Figure 4.18 SCC002. Manage Projects

*Table 4.18 Manage Projects description*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Name | Format | Reference | Note |
| 1 | Add Project | Button |  | When click on “Add project” button will show **SCC002-1.** Add project pop up |
| 2 | Project list | Table |  | Display Project list user created |
| 3 | Detail | Button |  | Next to **SCC003.** Management Project Function screen |
| 4 | Edit | Button |  | When click on “Edit” button will show **SCC002-1.** Edit project pop up |
| 5 | Delete | Button |  | When click the “Delete” button, the project will be deleted out project list |

##### SCC002-1. Add/Edit Project

Figure 4.19 SCC002-1. Add/Edit Project

*Table 4.19 Add/Edit Project description*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Name | Format | Reference | Note |
| 1 | Project Name | Textbox |  | The user will enter project name.  If user edit Project, Project name textbox will display default project name filled. |
| 2 | Link | Textbox |  | The user will enter project name.  If the user edits Link of project, Link textbox will display the default Link filled. |
| 3 | Submit | Button |  | Save data and back to **SCC002.** Manage Projects screen.  If user at Edit Project pop up when enter Submit then data will update. |
| 4 | Cancel | Button |  | Data do not change and back to **SCC002.** Manage Projects screen. |

#### SCC003. Manage Project Functions

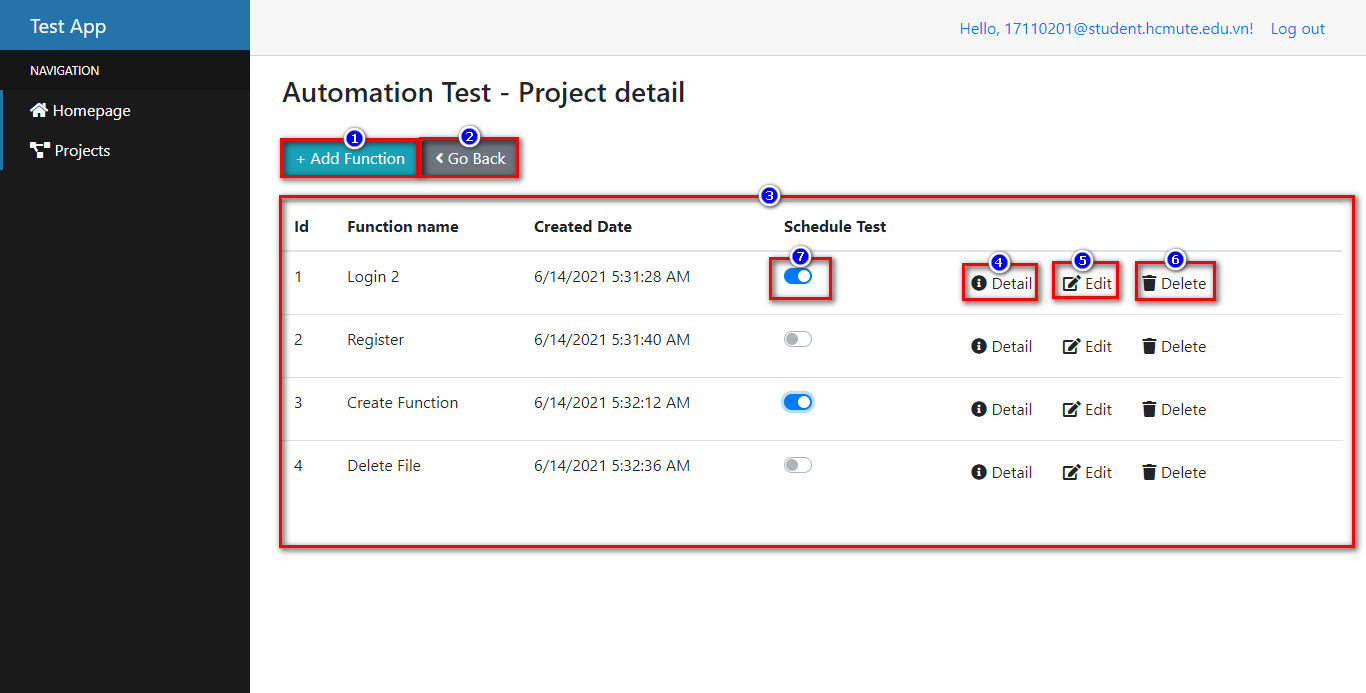


Figure 4.20 SCC003. Manage Project Functions

*Table 4.20 Manage Project Functions description*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Name | Format | Reference | Note |
| 1 | Add Function | Button |  | When click on “Add function” button will show **SCC003-1.** Add new function pop up. |
| 2 | Go back | Button |  | Back to **SCC002.** Manage Project screen. |
| 3 | Function list | Table |  | Display Function list user created of Project. |
| 4 | Detail | Button |  | Next to **SCC004**. Manage File Test of Function screen of function. |
| 5 | Edit | Button |  | When click on “Edit” button will show **SCC003-1.** Edit function pop up. |
| 6 | Delete | Button |  | When click the “Delete” button, the function will be deleted out function list. |
| 7 | Schedule | Switch |  | Display defaults turn off the Scheduler.  When user turn on Schedule of a function the then system will auto run test function schedule with each day and send report to mail of user  When user turn off Schedule, system won’t auto run test function. |

##### SCC003-1. Add/Edit Functions



Figure 4.21 SCC003-1. Add/Edit Functions.

*Table 4.21 Add/Edit Functions description*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Name | Format | Reference | Note |
|  | Function Name | Input |  | The user will enter function name.  If user at edit function pop up, Function name textbox will display default function name filled. |
| 2 | Submit | Button |  | Save data and back to **SCC003.** Manage Projects Function screen  If user at Edit Function pop up when click on button Submit then data will update. |
| 3 | Cancel | Button |  | Data do not change and back to **SCC003.** Manage Projects Function screen. |

#### SCC004. Manage File Test Of Function

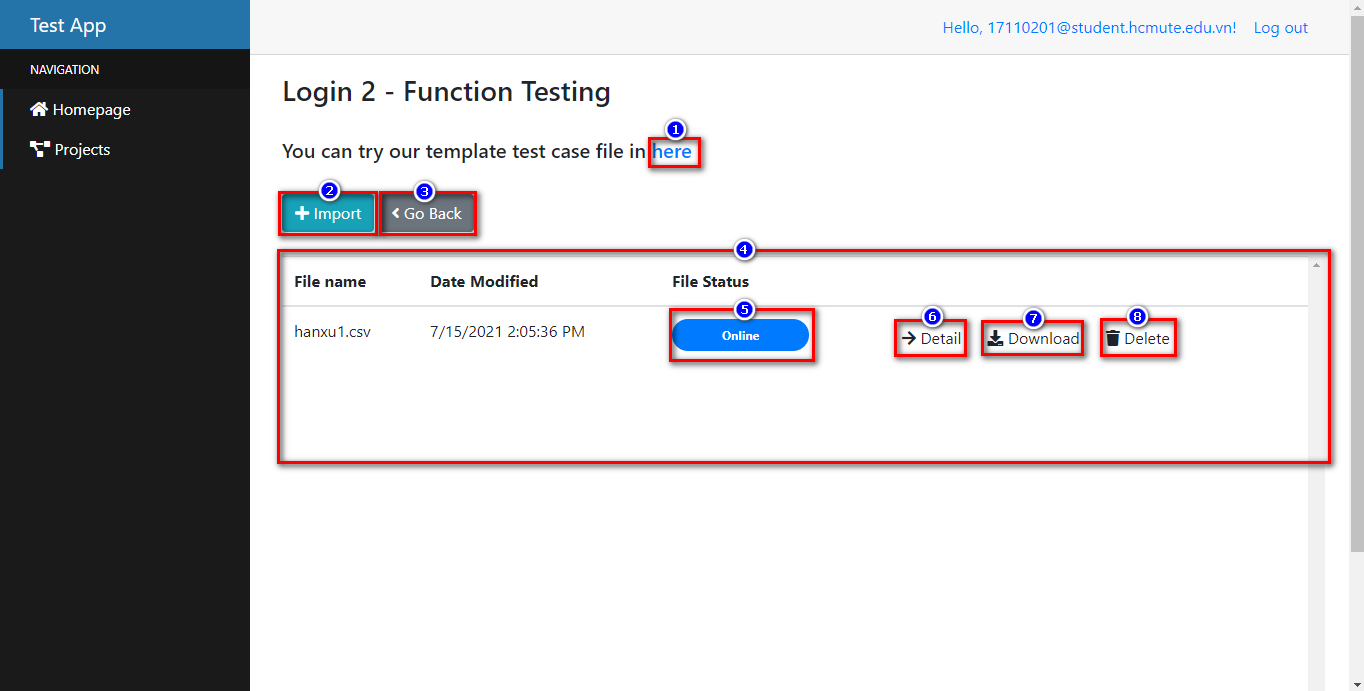


Figure 4.22 SCC004. Manage File Test of Function.

*Table 4.22 Manage File Test of Function description*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Name | Format | Reference | Note |
| 1 | Template | Link |  | Display default “here” word blue color.  When click on “here” will be download template file test import. |
| 2 | Import | Button |  | When click on “Import” will be **SCC004-1**. Upload file pop up. |
| 3 | Go back | Button |  | Back to **SCC003.** Management Project Function screen. |
| 4 | File List | Table |  | Display list imported files test. |
| 5 | Status | Button |  | Display status of files test.  If file wrong: Status is File not found.  If file success: Status is Online. |
| 6 | Detail | Button |  | Next to **SCC005.** Run Text Screen. |
| 7 | Download | Button |  | When click on “Download” will be download file test. |
| 8 | Delete | Button |  | Delete file selected. |

##### SCC004-1. Upload File

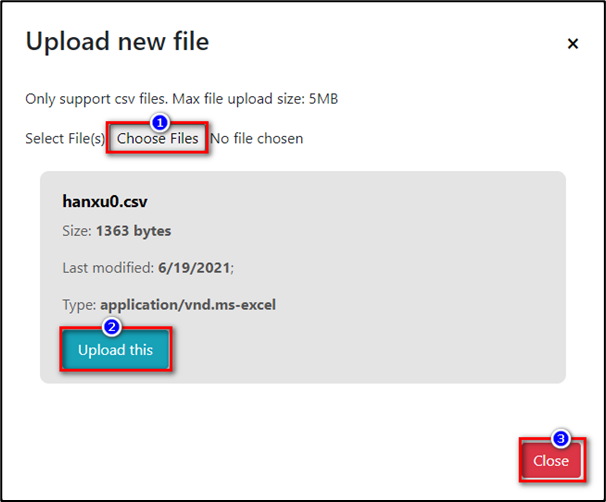


Figure 4.23 SCC004-1. Upload File

*Table 4.23 Upload File description*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Name | Format | Reference | Note |
| 1 | Choose Files | Button |  | When click on “Choose Files” will show Folder Desktop pop up.  If choose file wrong format, size >5mb then system will show information of file test in pop up.  If choose file success, system will show information of file and show “Upload this” button. |
| 2 | Upload this | Button |  | Display when user choose file correct format and size.  Hide when user choose file wrong format and size.  When click on “Upload this” button, file upload success, name “Upload this” will change “Upload 100%” and disable button. |
| 3 | Close | Button |  | Data upload and back to **SCC004.** Manage File Test of Function screen. |

#### SCC005. Run Test

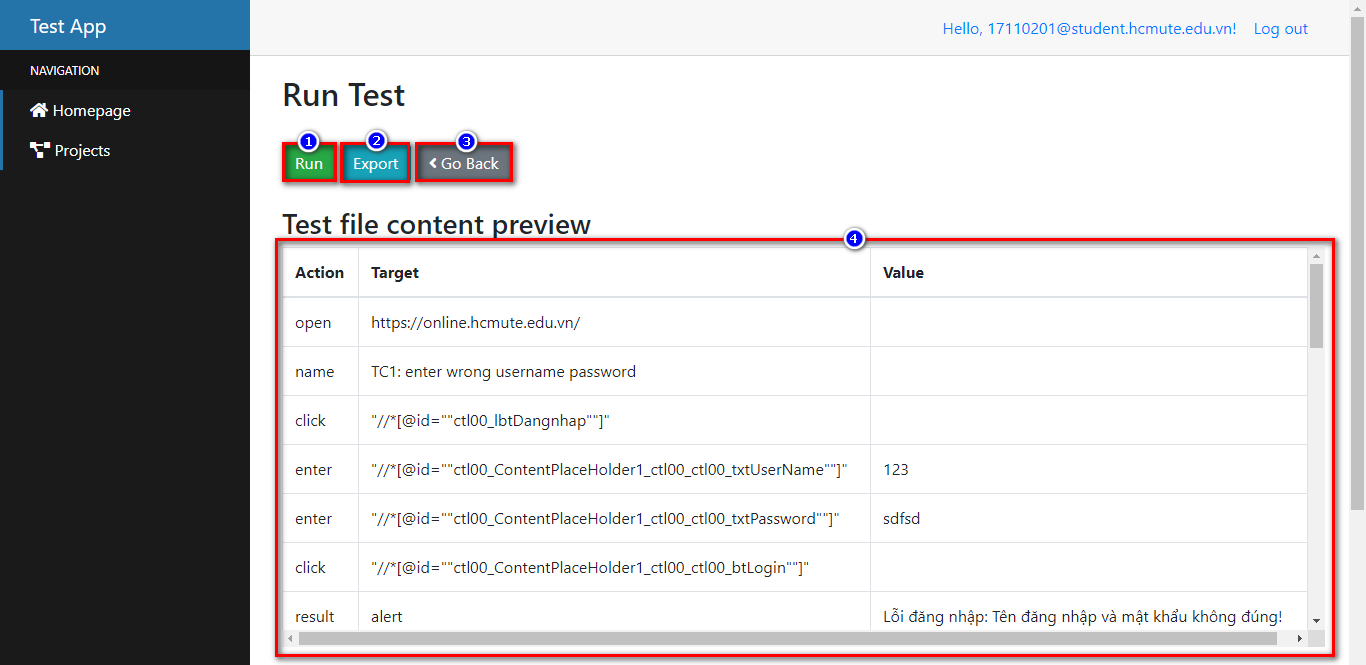


Figure 4.24 SCC005. Run Test

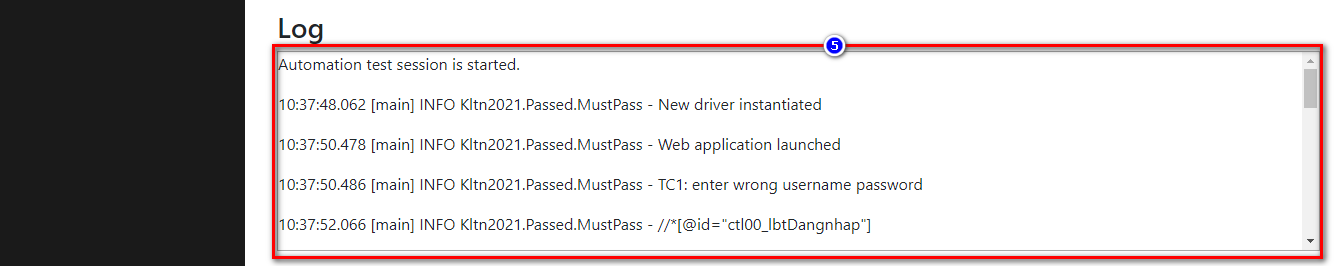


Figure 4.25 Run Test - Log

*Table 4.24 Run Test description*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Name | Format | Reference | Note |
| 1 | Run | Button |  | When click on “Run” will be start test function |
| 2 | Export | Button |  | Before done run test, click on “Export” system will send report to mail include information report |
| 3 | Go back | Button |  | Back to **SCC004.** Management File Test of Function Screen |
| 4 | Test file content preview | Table |  | Table data from file test import |
| 5 | Log |  |  | Show step run test process |

#### SCC006. Report

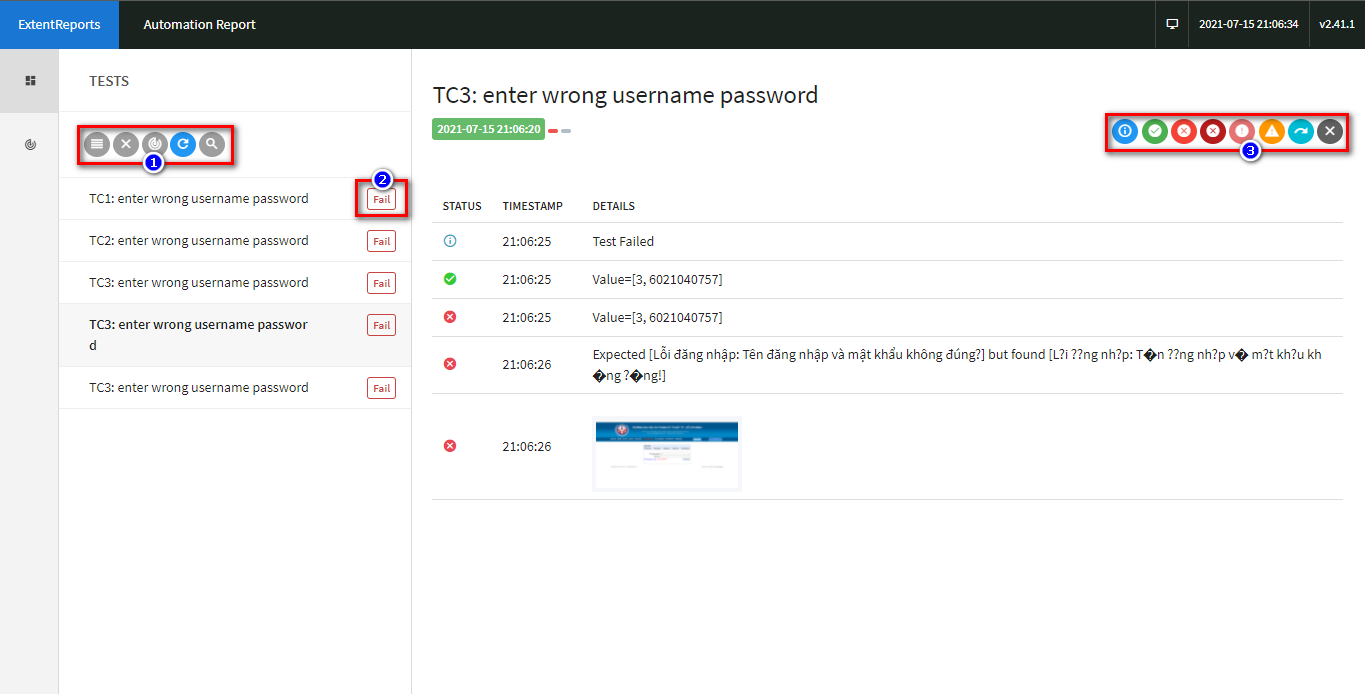


Figure 4.26 SCC006. Report

*Table 4.25 Report description*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Name | Format | Reference | Note |
| 1 | Filter toolkit Testcase | Button |  | Filter toolkit:  Search testcase name  Reload  Report |
| 2 | Status Testcase | Status button |  | Display status testcase |
| 3 | Filter toolkit Step | Button |  | Filter toolkit:  Filter success  Filter warning  Filter info  Clear… |

## System Implementation

### Development Environment

*Table 4.26 Development environment*

|  |  |
| --- | --- |
|  | **Version** |
| Chrome Diver | 91.0.4472 |
| Selenium | 4.0.0 |
| Eclipse | 2021-03 (4.19.0) |
| Visual Studio | 2019 |

#### Automation Test

**Maven** is an automation and management tool developed by Apache Software Foundation. It is written in Java Language to build projects written in C#, [Ruby](https://www.guru99.com/ruby-on-rails-tutorial.html), Scala, and other languages. It allows developers to create projects, dependency, and documentation using Project Object Model and plugins. Maven can also build any number of projects into desired output such as jar, war, metadata.

Maven helps the developer to create a java-based project more easily. Accessibility of new feature created or added in Maven can be easily added to a project in Maven configuration. It increases the performance of project and building process. The main feature of Maven is that it can download the project dependency libraries automatically. Below are the examples of some popular IDEs supporting development with Maven Framework:

* Eclipse
* IntelliJ IDEA
* JBuilder
* NetBeans
* MyEclipse

Set up Java:

* Download Java Development Kit (JDK)
* Install Java
* Set the Java Environment Path

Set up Eclipse:

* Download Eclipse

Download Selenium WebDriver

Configure Selenium WebDriver with Eclipse

* Install Maven project
* Create Maven project
* Add library into pom.xml: Selenium java.

+ Extent report

+ Log4j

+ Open CSV

+ Guava JDK

+ Java JDK

+ Commons-lang

+ Servlet-api

+ Guava

+ Webdriver manager

+ Slf4j-api

+ Groovy-all

+ Icu4j

+ Guice

+ Google-api-client-appengine

+ Gsonv

### Source Code Management

Any source code is managed by GitHub.

GitHub management tool is source tree and GitHub desktop.

Link GitHub backend and frontend: <https://github.com/DIEMVO/GRADUATION-PROJECT-2021>

Customer account: [17110109@student.hcmute.edu.vn/](mailto:17110109@student.hcmute.edu.vn/) Password: Helo@321

# SELENIUM IDE

## Project

Use Selenium IDE to perform automated testing on UTE Portal: HCMC University of Technology and Education website and build test cases.

URL: <https://online.hcmute.edu.vn/>

Graphical user interface, text, application, email

Description automatically generated

Figure 5.1 URL

## Test Case

Graphical user interface, application

Description automatically generated

Figure 5.2 Selenium screen

This feature allows view Testcase on website:

* Project Name (1): The name of my project.
* Testcases Name (2): Display name of test cases in project.
* Run All (3): Run the entire test suite.
* Run (4): Run by single test case.
* Test Execution Speed (5): Adjust speed/slow mode.
* URL (6): A valid base URL for my project.
* Reference Pane (7): Show detail a test case.
* Command (8): Selenium IDE commands.
* Target (9): Locators (ID, Name, CSS Selector, Xpath...).
* Value (10): Input values/compare with actual result.
* Result (11): Results after executing test script.
* Log (12): Display the test log during the run.

## Guideline

The team will guide through the steps of creating a basic test case on selenium IDE.

The website used: https://online.hcmute.edu.vn/.

User screen: the screen to look up English test scores of the HCMC University of Technology and Education, including selecting the form of the English test, entering the student code and press the "View Score" button.

Demo steps:

1. Install Selenium IDE extension on Google Chrome.

Graphical user interface, text, application

Description automatically generated

Figure 5.3 Install Selenium IDE

1. Turn on Selenium IDE on the extension bar.



Figure 5.4 Turn on Selenium IDE.

1. On the pop-up dialog box, click “Create a new project”.

Text

Description automatically generated with medium confidence

Figure 5.5 Create a new project.

1. On the pop-up dialog box:

* Enter “Project Name” (1).
* Click “OK” button (2).

Graphical user interface, application

Description automatically generated

Figure 5.6 Create a new project(1).

1. Put the link of the website to be tested in the Playback BaseUrl dataframe.

Graphical user interface, text, application, email

Description automatically generated

Figure 5.7 Playback BaseUrl

1. Click the “REC” icon to start recording.



Figure 5.8 Start recording

1. Navigate to the website on the Chrome browser.

* Click on the dropdown to choose the exam type (1).
* Enter student code (2).
* Click the button “View score” (3).
* Corresponding on Selenium IDE will record these actions.
* Click the “Stop” icon to stop recording.

Graphical user interface, text, email, website

Description automatically generated

Figure 5.9 Recording.

1. On the pop-up dialog box:

* Enter “TEST NAME” (1).
* Click the “OK” button (2).

Graphical user interface, application

Description automatically generated

Figure 5.10 Save Test Name

1. Then click the "Run current test" button (1) to run the test case creation process again and will get the following running results:

Graphical user interface, text, application

Description automatically generated

Figure 5.11 Run current test

* View test log after a run (2).

1. Save test case to the personal computer:

* Click the "Save Project" icon (1).
* Select the project storage path (2).
* Click the "Save" button (3).

Graphical user interface, application, Teams

Description automatically generated

Figure 5.12 Save test case.

## Advantage

* Easy to install, use save time.
* Good support for manual testers when no programming experience is required.
* Can group test cases into one test suites.
* Can edit steps via selenium command.
* Can export test cases to languages.

## Disadvantage

* Difficult to handle complex cases / workflow.
* The test report is not supported.

# SYSTEM TESTING

## Test Plan

### List Of Stages Of Testing

* Unit test.
* System test.
* Acceptance test.
* Integration test.

### List Of Test Types

* Function test.
* User interface test.
* Data and database integrity testing.
* Regression Testing.

### Constraints

* Operating system: Windows 10.
* Browser: Google Chrome.

### Plan Defect

During the testing process, when there is an error, the tester will notify the developer immediately.

## Test Design

### Items To Be Tested

* Automation Test.
* Web Application.
* Database.

### Features To Be Tested

Automation Test: UI testing, Headless Brower, data input, data output and file Log.

Web Application: Add, edit, delete, import file .CSV, export file .CSV, export report, send mail and save data.

Table, Excel

Description automatically generated

Figure 6.1 Features to be tested

Database: Update data.

## Test Case

Here are some typical test cases that have been tested:

### Test Case For Web Application

#### Test Case Of Login (Lg)

Prerequisites:

* The device must be connected to the internet.
* Direct to “Login” page.

*Table 6.1 Test case of Login (Lg)*

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case ID** | **Test Description** | **Test steps** | **Expected Result** |
| ***Lg\_01*** | Login the website | 1. On website, enter your Email/Password into the textbox. 2. Click “Log in” button. | Can login the website, the login page navigates to dashboard page. |
| ***Lg\_02*** | Login the website but using wrong Gmail. | 1. On website, enter your Gmail/Password into the textbox. 2. Click “Log in” button. | The system must be showed the message box: “*Invalid login attempt*.” |
| ***Lg\_03*** | Login the website but using wrong Password. | 1. On website, enter your Gmail/Password into the textbox. 2. Click “Log in” button. | The system must be showed the message box: “*Invalid login attempt*.” |
| ***Lg\_04*** | Check “Remember me] checkbox. | 1. On website, enter your Gmail/Password into the textbox. 2. Check into “Remember me” checkbox. 3. Click “Log in” button. | The system must remember the account the next time it is logged in. |
| ***Lg\_05*** | Logout the website | 1. On website, click “Logout”. | Can logout the website. |

#### Test Case Of Project (Prt)

Prerequisites:

The device must be connected to the internet.

Direct to “Project” page.

*Table 6.2 Test case of Project (Prt)*

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case ID** | **Test Description** | **Test steps** | **Expected Result** |
| ***Prt\_01*** | Add new the project | 1. Click the “Add Project” button. 2. Enter the content into the textbox. 3. Click the “Submit” button. | The new Project must be created and shows on the table on website. |
| ***Prt\_02*** | Edit the project | 1. Click the “Edit” icon on the row. 2. Edit the content into the textbox. 3. Click the “Submit” button. | The Project must be edited and shows on the table on website. |
| ***Prt\_03*** | Delete the project | 1. Click “Delete” icon on row. 2. The system popup displays. 3. Click the “Delete” button. | The Project must be deleted from data table. |
| ***Prt\_04*** | Check navigate page | 1. Click “Detail” icon on row. | The system can navigate and linking pages. |
| ***Prt\_05*** | Check go back | 1. Click the “Go Back” button. | The system can navigate and linking pages. |

#### Test Case Of Manage Project Function (Prf)

Prerequisites:

The device must be connected to the internet.

Direct to “Manage Project Function” page.

*Table 6.3 Test case of Manage Project Function (Prf)*

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case ID** | **Test Description** | **Test steps** | **Expected Result** |
| ***Prf\_01*** | Add new the function | 1. Click the “Add Function” button. 2. Enter the content into the textbox. 3. Click the “Submit” button. | The new Function must be created and shows on the table on website. |
| ***Prf\_02*** | Edit the function | 1. Click the “Edit” icon on the row. 2. Edit the content into the textbox. 3. Click the “Submit” button. | The Function must be edited and shows on the table on website. |
| ***Prf\_03*** | Delete the function | 1. Click “Delete” icon on row. 2. The system popup displays. 3. Click the “Delete” button. | The Function must be deleted from data table. |
| ***Prf\_04*** | Schedule Test | 1. Click “Schedule” icon | Every day the system will automatically execute the test according to the headless browser mechanism and send the result report to the email |
| ***Prf\_05*** | Check navigate page | 1. Click “Detail” icon on row. | The system can navigate and linking pages. |
| ***Prf\_06*** | Check go back | 1. Click the “Go Back” button. | The system can navigate and linking pages. |

#### Test Case Of Manage File Test (Mft)

Prerequisites:

The device must be connected to the internet.

Direct to “Manage File Test” page.

*Table 6.4 Test case of Manage File Test (Mft)*

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case ID** | **Test Description** | **Test steps** | **Expected Result** |
| ***Mft\_01*** | Import the new file .CSV | 1. Click the “Import” button. 2. Click the “Choose Files” button. 3. Choose file .CSV and click “Upload this”. 4. Click the “Close” button. | The new file must be upload and shows on the table on website. |
| ***Mft\_02*** | Download the template test case file | 1. Click the “here” label on website. | The template file .CSV must be downloaded to the personal computer. |
| ***Mft\_03*** | Download file .CSV | 1. Click the “Download” icon on the row. | The .CSV file must be downloaded to the personal computer. |
| ***Mft\_04*** | Delete the File | 1. Click “Delete” icon on row. | The file must be deleted from data table. |
| ***Mft\_05*** | Check navigate page | 1. Click “Detail” icon on row. | The system can navigate and linking pages. |
| ***Mft\_06*** | Check go back | 1. Click the “Go Back” button. | The system can navigate and linking pages. |

#### Test Case Of Run Test (Rt)

Prerequisites:

The device must be connected to the internet.

Direct to “Test Screen” page.

*Table 6.5 Test case of Run Test (Rt)*

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case ID** | **Test Description** | **Test steps** | **Expected Result** |
| ***Rt\_01*** | Preview Test file content |  | Details of each test case in the .CSV file will be shown on the website. |
| ***Rt\_02*** | Run manual | 1. Click the “Run” button on website. | Automated test execution. |
| ***Rt\_03*** | Show log | 1. Click the “Run” button on website. | When executing the test, it will also show the log. |
| ***Rt\_04*** | Send email | 1. Run Manual/Automation. | After successful test execution will automatically send report results to mail. |

## Experiment Results

Below is a statistical chart of the test coverage results for each test case in the screen.

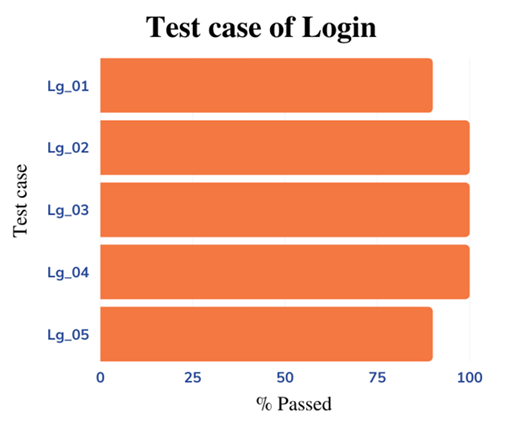


Figure 6.2 Testcase of Login.

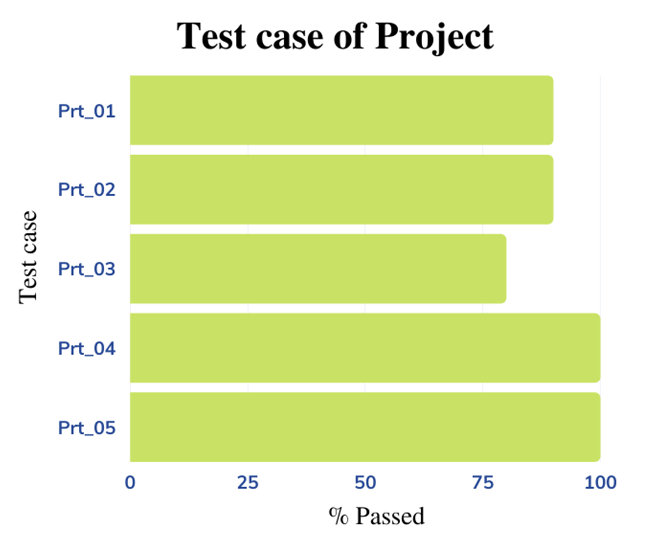


Figure 6.3 Test case of Project.

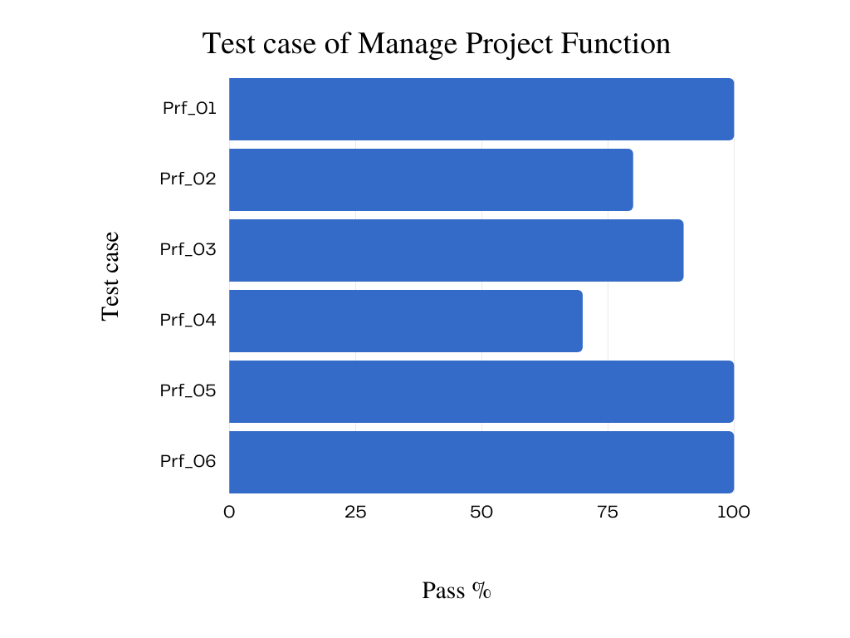


Figure 6.4 Test case of Manage Project Function

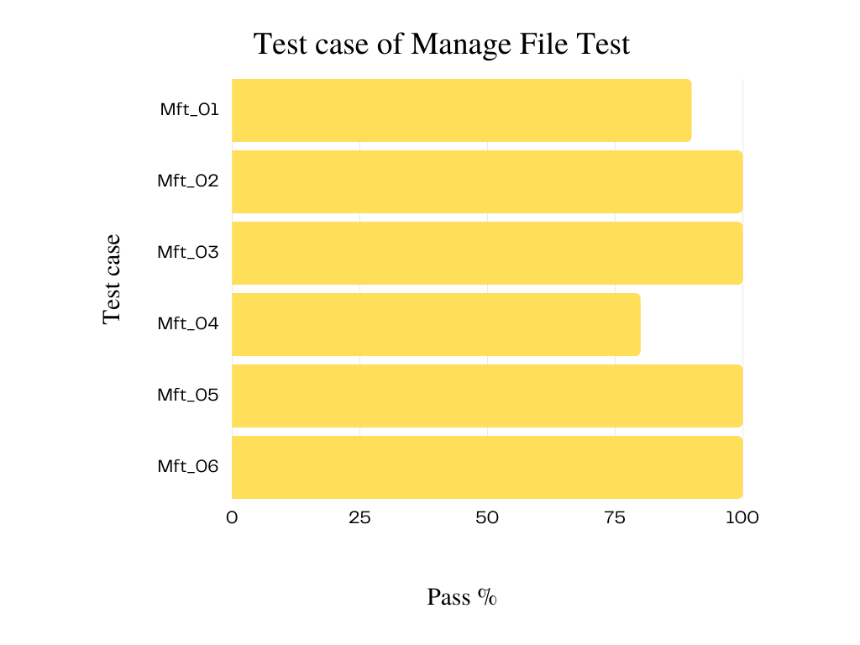


Figure 6.5 Test case of Manage File Test

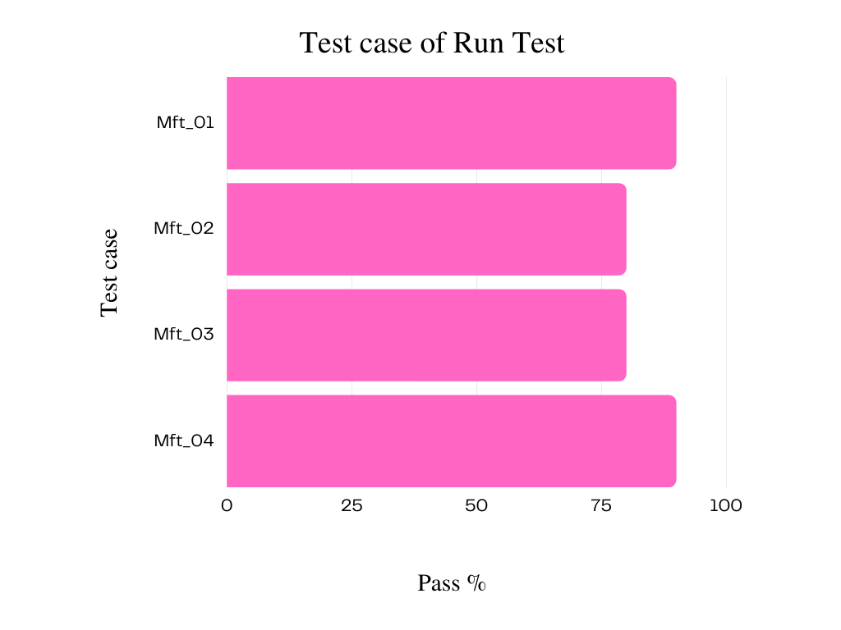


Figure 6.6 Test case of Run Test

## System Deployment

### Publish And Deploy On Heroku Hosting

1. Prerequisites:

Blazor app in your GitHub

Heroku account

2. Go to the Heroku dashboard and create a new app.

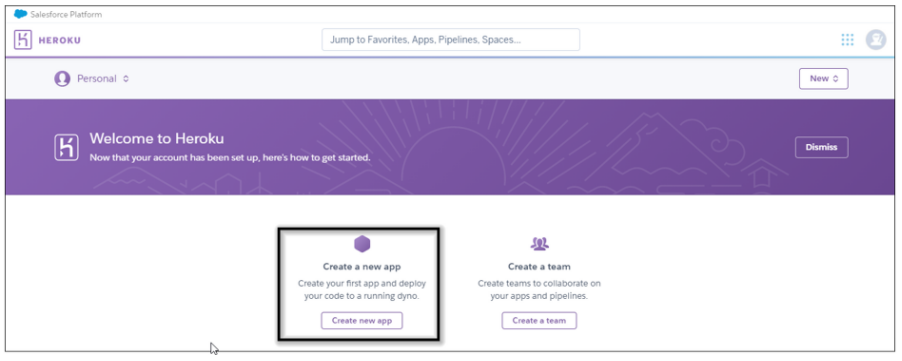


Figure 6.7 Heroku dashboard

3. Provide the app name and create an app.

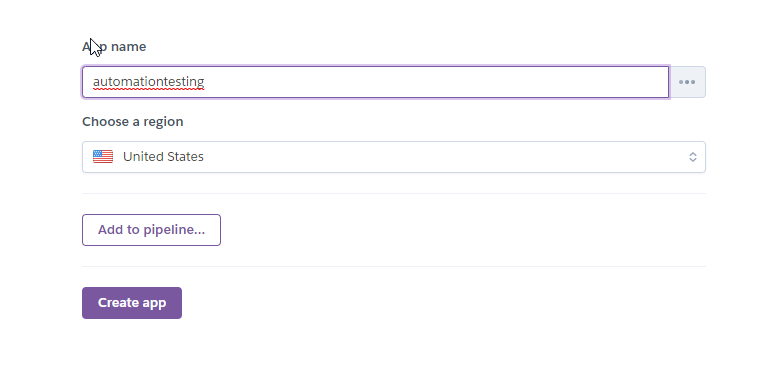


Figure 6.8 Create a new app

1. Now the new app is created. Next connect your GitHub Blazor Web app repository to Heroku for deployment.

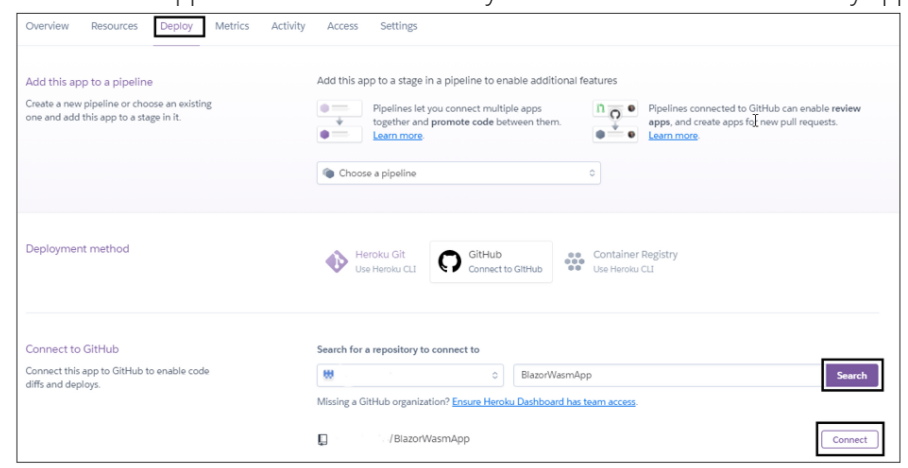


Figure 6.9 Connect

1. By default, Heroku does not support .NET apps for deployment. So, we add third-party build packs to enable the .NET app deployment support. Go to the settings tab and click Add buildpack.

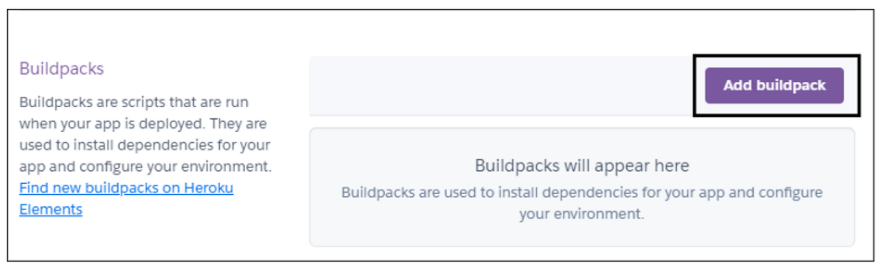


Figure 6.10 Add buildpack.

1. Now the dialog box will open. Copy the link below, paste it into the Buildpack URL section, and save your changes.

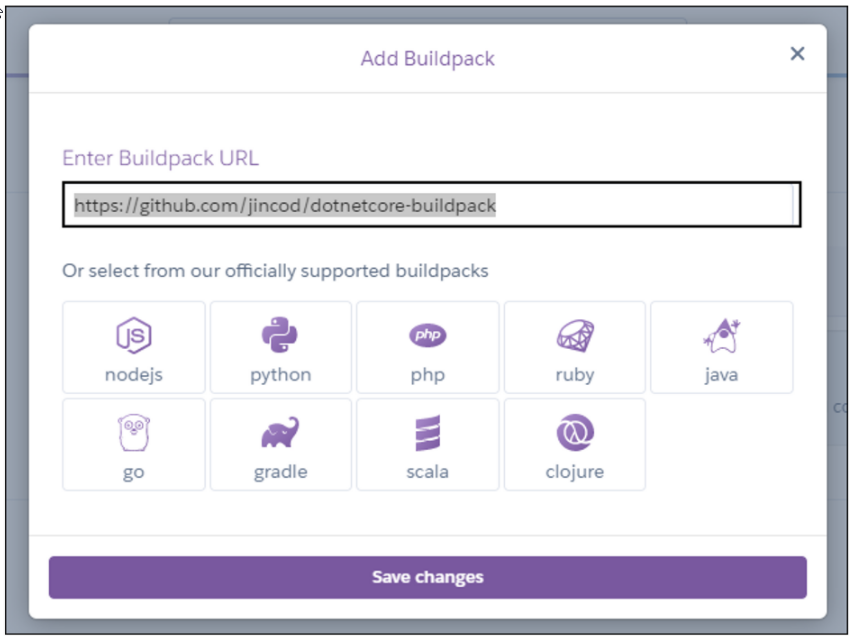


Figure 6.11 Buildpack URL.

1. The build pack has been added. Go to the Deploy tab and deploy the repository manually. The Blazor app starts building now.

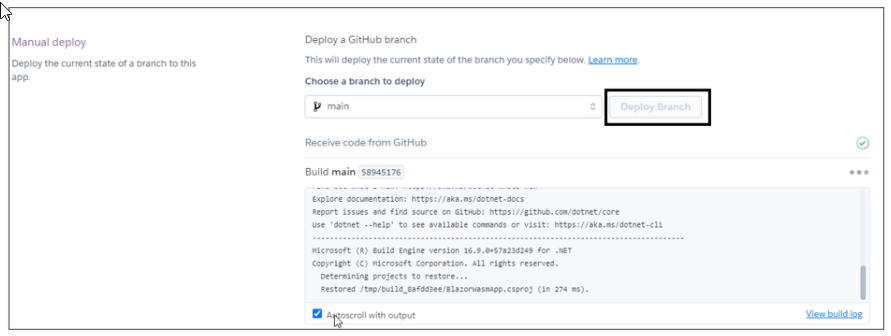


Figure 6.12 Starts building

1. Now the Blazor app is deployed. Click View to run the deployed application.

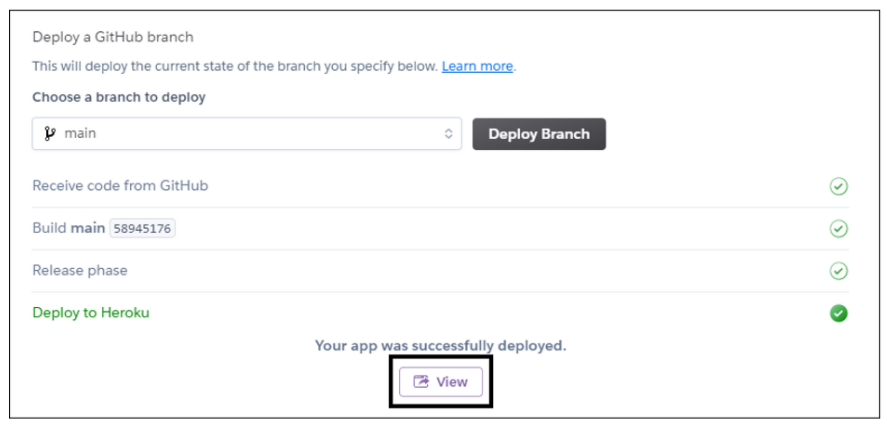


Figure 6.13 View to run

1. The deployed application opens in the default browser. See the following image for reference. This is team’s website: <http://automationtesing.herokuapp.com/> [7]

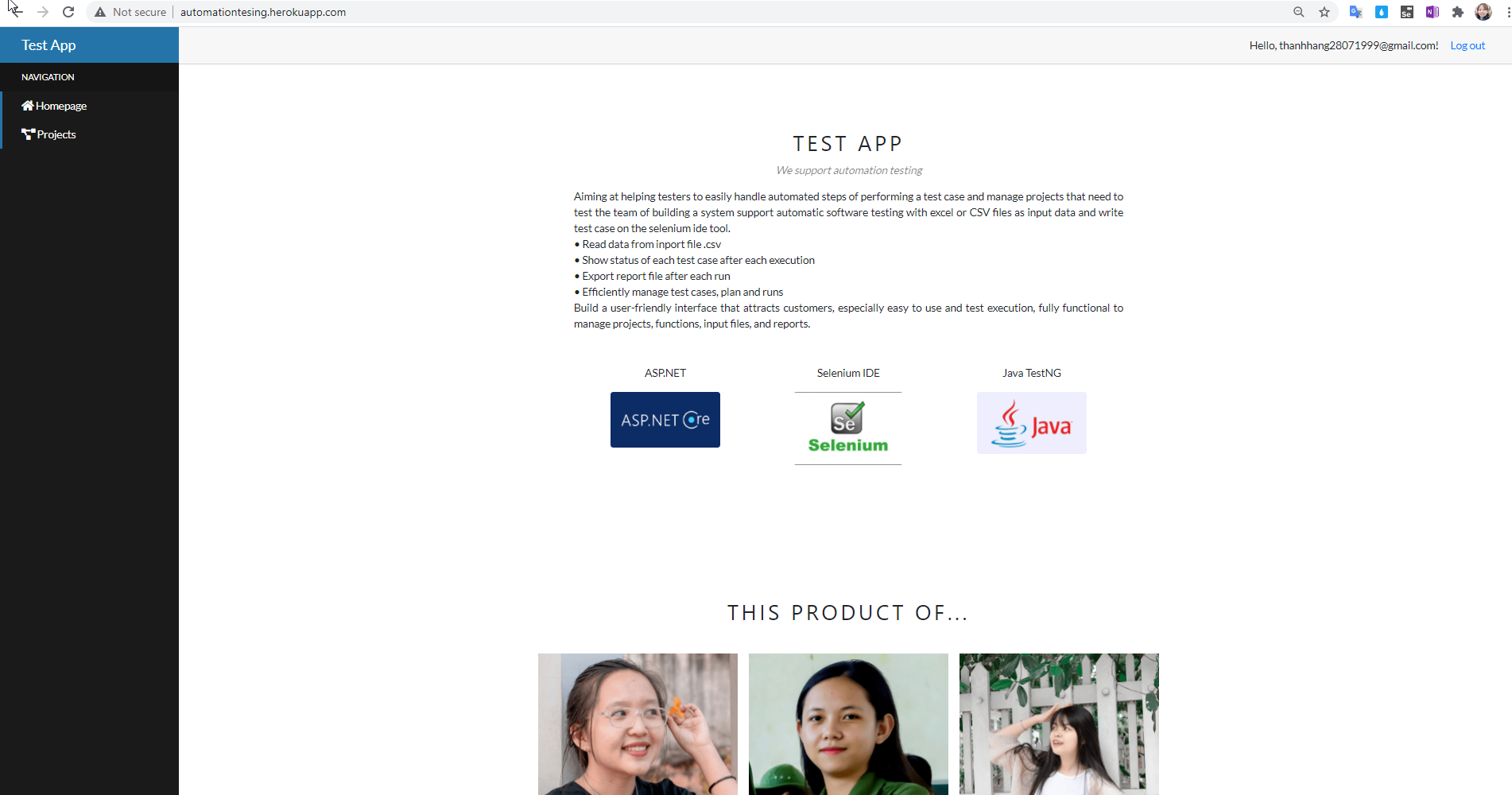


Figure 6.14 Team’s website

# CONCLUSION

## Result

Complete the process of learning the process of Selenium IDE and apply the knowledge learned to build a website that supports automation testing.

Handle the test process easily from management to reporting, besides, users can also configure and manage test run schedules.

Complete the web application part including the functionality:

* Management Project.
* Management Function.
* Management File.
* Management Run.
* Management Report File.

Test automation cases can be handled: test on textbox, dropdown, alert box, inner element, open new tab, radio button, checkbox, upload file, delete file.

## Advantage

Make testing tasks easy, save time and effort.

More keywords can help with user automation tasks.

More random test cases with many patterns.

Report file show information clearly, user can filter pass, fail result and show pass/fail percentage.

User can see fail test case by the image in the report result file.

Log file can inform users of every single step run automation of .CSV file and know step fails.

After the run test is done, the user receives the report result via email.

User can set a time to run the underground.

## Disadvantage

Only supporting one browser is Chrome.

Only supporting upload file type .csv.

There are some cases where the system has not resolved: tooltips, links, images, tables.

## Future Plan

Support test multi browser such as: Firefox, Microsoft Edge, …

Add more keywords to do diverse workflow.

Support more types of excel files.

Improve performance with optional speed such as: low, fast, medium speed.

Change the deployment environment to support UI less browser functionality more easily.

Schedule auto test for a complete project (weekly, monthly).

References

[1] “Selenium Testing”. [Online]. Available:

<https://www.toolsqa.com/selenium-webdriver/selenium-testing/>. [Accessed 04 2021].

[2] “Selenium WebDriver Architecture”. Available:

<https://www.toolsqa.com/selenium-webdriver/selenium-webdriver-architecture/> .

[Accessed 06 2021].

[3] “What is ASP.NET?” [Online]. Available:

<https://dotnet.microsoft.com/learn/aspnet/what-is-aspnet> [Accessed 04 2021].

[4] “What's new in SQL Server 2017”, 10/20/2017 [Online] Available:

<https://docs.microsoft.com/en-us/sql/sql-server/what-s-new-in-sql-server-2017?view=sql-server-ver15> [Accessed 05 2021].

[5] “Blazor” [Online] Available:

<https://dotnet.microsoft.com/apps/aspnet/web-apps/blazor> [Accessed 05 2021].

[6] “Selenium IDE Tutorial for Beginners” [Online] Available:

<https://www.guru99.com/introduction-selenuim-ide.html> [Accessed 06 2021].

[7] “How do I deploy a Blazor WebAssembly app to Heroku?” [Online] Available:

<https://www.syncfusion.com/faq/blazor/host-and-deploy/how-do-i-deploy-a-blazor-webassembly-app-to-heroku> [Accessed 07 2021].

Appendices

Table . Job table of work

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Job | | Vo Thi Kieu Diem | Bui Thi Hong Nhung | Pham Thi Thanh Hang |
| Related product survey | | X | X | X |
| Learn the business of automated test pages | | X | X | X |
| Research using java for automation testing and Asp.net core Blazor for website. | | X | X | X |
| Learn about Java TestNG Framework | | X | X | X |
| Design prototype | | X | X | X |
| System Design | Use Case Diagram | X | X | X |
| Architecture Diagram | X | X | X |
| Sequence Diagram | | X | X | X |
| ER Diagram | | X | X | X |
| Database Diagram | | X | X | X |
| Research and design the input data file of the test system | | X | X | X |
| Design Web Application | | X | X | X |
| Automation test | | X | X | X |
| Export report file after each run | |  |  | X |
| Display the test log during the run. | | X |  | X |
| Handling unconstrained data cases, allowing to run testcases with random data | | X | X | X |
| **File Word** | | X | X | X |
| **File Power Point** | | X | X | X |