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

**FACULTY FOR HIGH QUALITY TRAINING**

**

**GRADUATION PROJECT**

**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**

**Major: INFORMATION TECHNOLOGY**

**Advisor: NGUYEN DUC KHOAN, PhD**

Ho Chi Minh City, January 2020

|  |  |
| --- | --- |
|  | THE SOCIALIST REPUBLIC OF VIETNAM  **Independence – Freedom– Happiness**  --------  *Ho Chi Minh City, January 20, 2020* |

# 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: ThS. NGUYEN DUC KHOAN 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 TestNG Framework, Selenium, 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 in Excel, and uses Selenium IDE to perform automated testing on a specific website and build test cases. 4. Final product: Design and build a webapp full of basic automated testing and management functions. | |
| **CHAIR OF THE PROGRAM**  *(Sign with full name)* | **ADVISOR**  *(Sign with full name)* |

|  |  |
| --- | --- |
|  | THE SOCIALIST REPUBLIC OF VIETNAM  **Independence – Freedom– Happiness**  --------  *Ho Chi Minh City, January 20, 2020* |

# 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

Project title: Build an application that supports automation testing.

Advisor: NGUYEN DUC KHOAN

**EVALUATION**

1. Content of the project: ...................................................................................................................................................................................................................................................................................... ...........................................................................................................................................
2. Strength: ...........................................................................................................................................

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

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

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, month day, year*

**ADVISOR**

*(Sign with full name)*

|  |  |
| --- | --- |
|  | THE SOCIALIST REPUBLIC OF VIETNAM  **Independence – Freedom– Happiness**  --------  *Ho Chi Minh City, January 20, 2020* |

# 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: NGUYEN DUC KHOAN

**EVALUATION**

1. Content of the project:

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

1. Strength:

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

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, month day, year*

**REVIEWER**

*(Sign with full name)*

**DISCLAIMER**

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 most sincere 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 group implementation**

**ACKNOWLEDGEMENTS**

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](#_Toc74606320)

[GRADUATION PROJECT ASSIGNMENT ii](#_Toc74606321)

[PRE-DEFENSE EVALUATION SHEET iii](#_Toc74606322)

[DISCLAIMER iv](#_Toc74606323)

[ACKNOWLEDGEMENTS v](#_Toc74606324)

[TABLE OF CONTENTS vi](#_Toc74606325)

[ABSTRACT ix](#_Toc74606326)

[LIST OF FIGURES x](#_Toc74606327)

[LIST OF TABLES xi](#_Toc74606328)

[Chap 1: INTRODUCTION 1](#_Toc74606329)

[1.1. The reason for choosing the topic 1](#_Toc74606330)

[1.2. Goal 1](#_Toc74606331)

[1.3. Technology 1](#_Toc74606332)

[Chap 2: THEORETICAL BASIS 2](#_Toc74606333)

[2.1. Java TestNG Framework 2](#_Toc74606335)

[2.2. ASP.NET Entity Framework 2](#_Toc74606336)

[2.3. Microsoft SQL Server 2017 3](#_Toc74606337)

[2.4. Blazor 4](#_Toc74606338)

[Chap 3: LITERATURE REVIEW 5](#_Toc74606339)

[3.1. Related product survey 5](#_Toc74606341)

[3.1.1. Selenium IDE 5](#_Toc74606342)

[3.1.2. Test Collab 7](#_Toc74606343)

[3.1.3. Conclusion 8](#_Toc74606344)

[3.2. File Input Design 8](#_Toc74606345)

[3.3. File Output Design 9](#_Toc74606346)

[3.4. System Design 10](#_Toc74606347)

[3.4.1. Use Case Diagram 10](#_Toc74606348)

[3.4.2. Use case Description 11](#_Toc74606349)

[3.5. Sequence Diagram 28](#_Toc74606350)

[3.5.1. <Customer> Create Project 28](#_Toc74606351)

[3.5.2. < Customer > Login 29](#_Toc74606352)

[3.5.3. <Customer> Register 30](#_Toc74606353)

[3.5.4. <Customer> Run Test 31](#_Toc74606354)

[3.5.5. <Customer> Import File 32](#_Toc74606355)

[3.5.6. <Customer> Send Mail 33](#_Toc74606356)

[3.6. Architecture diagram 33](#_Toc74606357)

[3.7. Database Design 33](#_Toc74606358)

[3.7.1. ER Diagram 33](#_Toc74606359)

[3.7.2. Database Diagram 34](#_Toc74606360)

[3.8. User Interfaces Design 35](#_Toc74606361)

[3.8.1. Login screen 35](#_Toc74606362)

[3.8.2. Register 35](#_Toc74606363)

[3.8.3. Dashboard screen 35](#_Toc74606364)

[3.8.4. Management Project screen 36](#_Toc74606365)

[3.8.5. Management Function screen 36](#_Toc74606366)

[3.8.6. Management File screen 37](#_Toc74606367)

[3.8.7. Test screen 37](#_Toc74606368)

[3.8.8. Report screen 37](#_Toc74606369)

[Chap 4: SYSTEM IMPLEMENTATION 38](#_Toc74606370)

[4.1. Software development environment 38](#_Toc74606372)

[4.2. Source code management 38](#_Toc74606373)

[Chap 5: SELENIUM IDE 39](#_Toc74606374)

[5.1. Project 39](#_Toc74606377)

[5.2. Testcase 39](#_Toc74606378)

[5.3. Guideline 40](#_Toc74606379)

[5.4. Advantage 44](#_Toc74606380)

[5.5. Disadvantage 44](#_Toc74606381)

[Chap 6: CONCLUSION AND RECOMMENDATIONS 45](#_Toc74606382)

[6.1. Conclusion 45](#_Toc74606383)

[6.2. Advantage 45](#_Toc74606384)

[6.3. Disadvantage 45](#_Toc74606385)

[6.4. Future planning 45](#_Toc74606386)

[REFERENCES 46](#_Toc74606387)

[APPENDICES 47](#_Toc74606389)

[JOB TABLE OF WORK 47](#_Toc74606390)

# ABSTRACT

|  |  |
| --- | --- |
| **Term** | **Definition** |
| DB | Database |
|  |  |

**LIST OF FIGURES**

[Figure 1: Selenium IDE 5](#_Toc74608521)

[Figure 2: Test Collab 7](#_Toc74608522)

[Figure 3: File Input 9](#_Toc74608523)

[Figure 4: Report Template 9](#_Toc74608524)

[Figure 5: Use Case Diagram 10](#_Toc74608525)

[Figure 6: <Customer> Manage Test Screen 11](#_Toc74608526)

[Figure 7: <Customer> Run Test 11](#_Toc74608527)

[Figure 8: < Customer> Export Report 13](#_Toc74608528)

[Figure 9: <Customer> Send Mail 15](#_Toc74608529)

[Figure 10: <Customer> Manage Test File 17](#_Toc74608530)

[Figure 11: <Customer> View Template 17](#_Toc74608531)

[Figure 12: <Customer> Import Input File 19](#_Toc74608532)

[Figure 13: <Customer> Export File 21](#_Toc74608533)

[Figure 14: <Customer> Add Function 23](#_Toc74608534)

[Figure 15: <Customer> Add Project 25](#_Toc74608535)

[Figure 16: <Customer> Create Project 28](#_Toc74608536)

[Figure 17: <Customer > Login 29](#_Toc74608537)

[Figure 18: <Customer> Register 30](#_Toc74608538)

[Figure 19: <Customer> Run Test 31](#_Toc74608539)

[Figure 20: <Customer> Import File 32](#_Toc74608540)

[Figure 21: Architecture diagram 33](#_Toc74608541)

[Figure 22: ER Diagram 33](#_Toc74608542)

[Figure 23: Database Diagram 34](#_Toc74608543)

[Figure 24: Dashboard screen 35](https://d.docs.live.net/e7ccdad05aa968a0/Desktop/KLTN_DNH.docx#_Toc74608544)

[Figure 25: Management Project screen 36](https://d.docs.live.net/e7ccdad05aa968a0/Desktop/KLTN_DNH.docx#_Toc74608545)

[Figure 26: Management Function screen 36](https://d.docs.live.net/e7ccdad05aa968a0/Desktop/KLTN_DNH.docx#_Toc74608546)

[Figure 27: Management File screen 37](#_Toc74608547)

[Figure 28: UTE Portal 39](#_Toc74608548)

[Figure 29: Test Case Selenium 40](#_Toc74608549)

[Figure 30: Install Selenium IDE 41](#_Toc74608550)

[Figure 31: Turn on Selenium IDE 41](#_Toc74608551)

[Figure 32: Create a new project 41](#_Toc74608552)

[Figure 33: Create a new project (1) 42](#_Toc74608553)

[Figure 34: BaseUrl 42](#_Toc74608554)

[Figure 35: Start recording 42](#_Toc74608555)

[Figure 36: Recording 43](#_Toc74608556)

[Figure 37: Test Name 43](#_Toc74608557)

[Figure 38: Detail test case 44](#_Toc74608558)

[Figure 39: Save test case 44](#_Toc74608559)

# LIST OF TABLES

[Table 1: Use Case Description 10](#_Toc74608560)

[Table 2: <Customer> Run Test 13](#_Toc74608561)

[Table 3: <Customer> Export Report 15](#_Toc74608562)

[Table 4: <Customer> Send Mail 17](#_Toc74608563)

[Table 5: <Customer> View Template 19](#_Toc74608564)

[Table 6: <Customer> Import Input File 20](#_Toc74608565)

[Table 7: <Customer> Export File 22](#_Toc74608566)

[Table 8: <Customer> Add Function. 24](#_Toc74608567)

[Table 9: <Customer> Add Project 27](#_Toc74608568)

[Table 10: Database Description 34](#_Toc74608569)

[Table 11: User Interfaces Design 35](#_Toc74608570)

[Table 12: Job table of work 49](#_Toc74608571)

# Chap 1: INTRODUCTION

## The reason for choosing the topic

Currently, 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.

## Goal

Aiming at helping testers to easily handle automated steps of performing a test case and manage projects that need to test the team of building a system support automatic software testing with excel or CSV files as input data and write test case on the selenium ide tool.

* Learn Selenium web driver, execution mechanism.
* Learn Selenium IDE to write test case for any website.
* Learn to design an input file structure.

Build a user-friendly interface that attracts customers, especially easy to use and test execution, fully functional to manage projects, functions, input files, and reports.

## Technology

Implementation using Java TestNG Framework, Selenium, Blazor, ASP.NET CORE 3.x, SQL Server.

# Chap 2: THEORETICAL BASIS



## Java TestNG Framework

TestNG is an automation testing framework in which NG stands for "Next Generation". TestNG is inspired from JUnit which uses the annotation. TestNG overcomes the disadvantages of JUnit and is designed to make end-to-end testing easy. [1]

Using TestNG, you can generate a proper report, and you can easily come to know how many test cases are passed, failed, and skipped. You can execute the failed test cases separately.

There are three major advantages of TestNG over JUnit:

* Annotations are easier to understand.
* Test cases can be grouped more easily.
* Parallel testing is possible.

## 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. [2]

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 [3].

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. [4]

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.

# Chap 3: LITERATURE REVIEW



## Related product survey

### Selenium IDE

Selenium IDE (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 [5].

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.

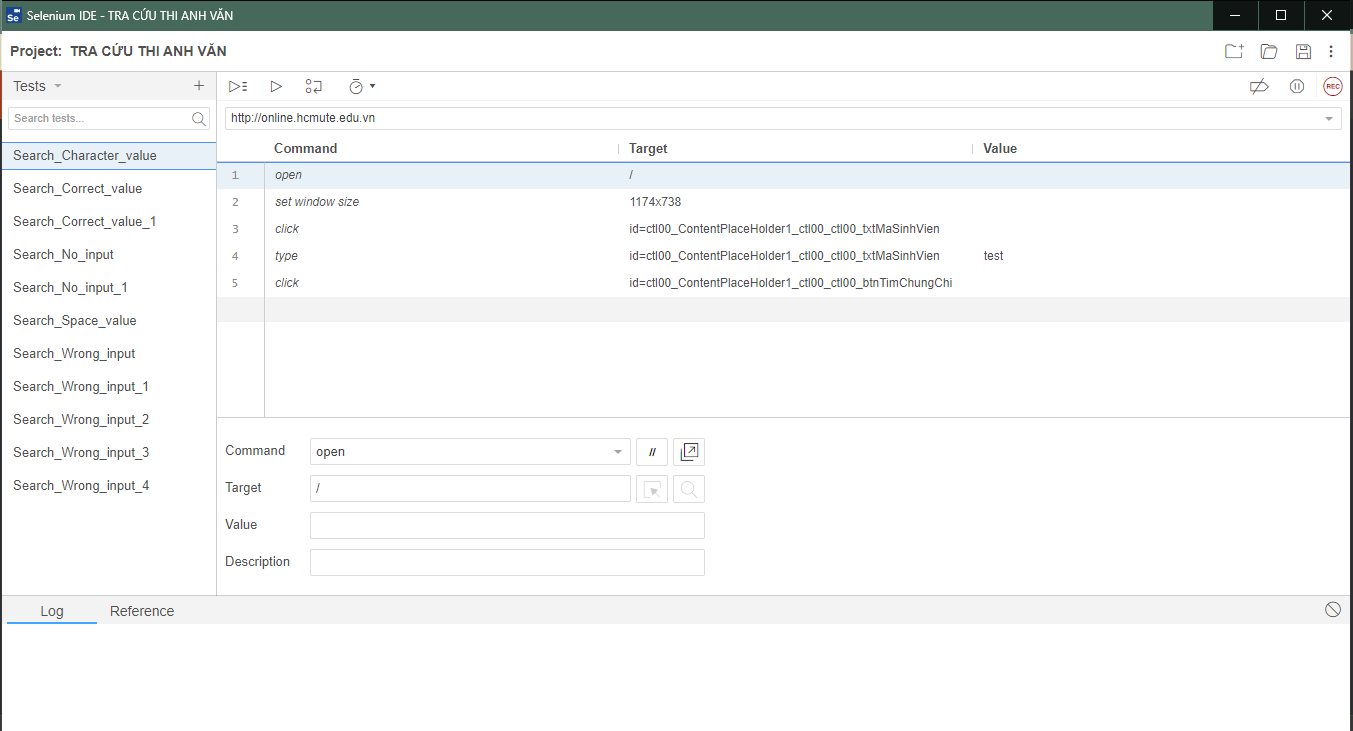


Figure : Selenium IDE

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.

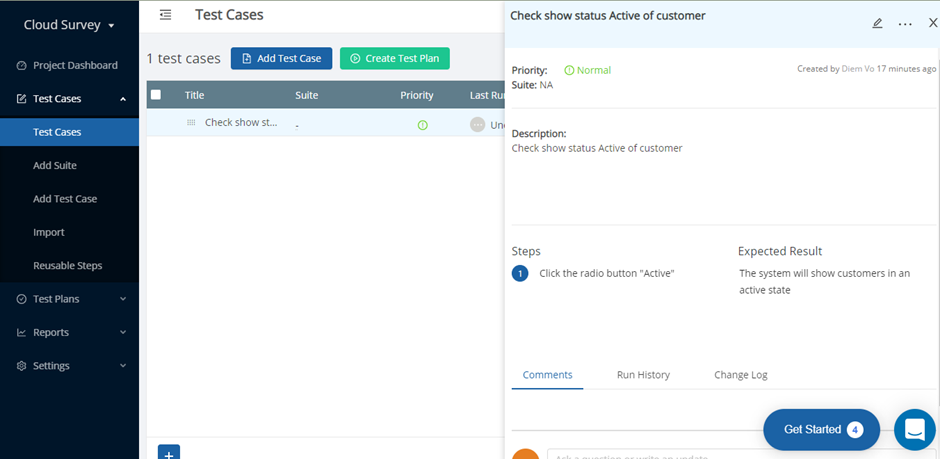


Figure : Test Collab

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 twoway 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:

* Manage Project, project Detail, manage Testcase, manage Report.
* Read data from import file .csv
* Displays the status of each testcase after each execution (number of testcase pass/fail).
* Display the test log during the run.
* Handling unconstrained data cases, allowing to run testcases with random data.
* Export report file after each run and customer can export the results sent to mail.
* Research and design the input data file of the test system, contains step-by-step details for users to track the error checking process and contains test data.

## File Input Design

File Input: the column includes the contents:

* Action: open, name, enter, click, result...
* Target: URL, test case name, Xpath ...
* Value: value, expected result...

Graphical user interface, text, application

Description automatically generated

Figure : File Input

## File Output Design

File report to send mail.

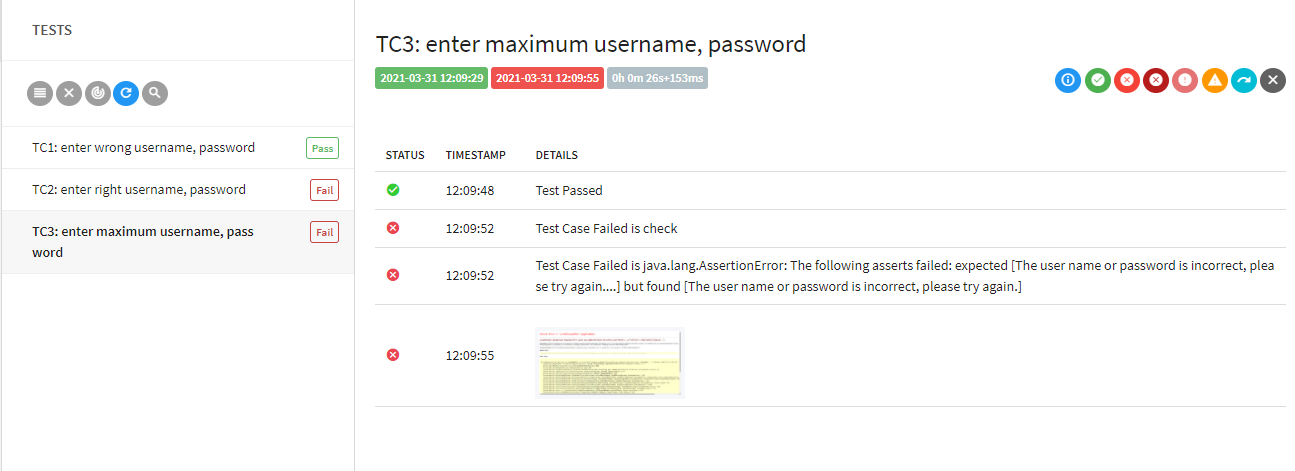


Figure : Report Template

## System Design

### Use Case Diagram

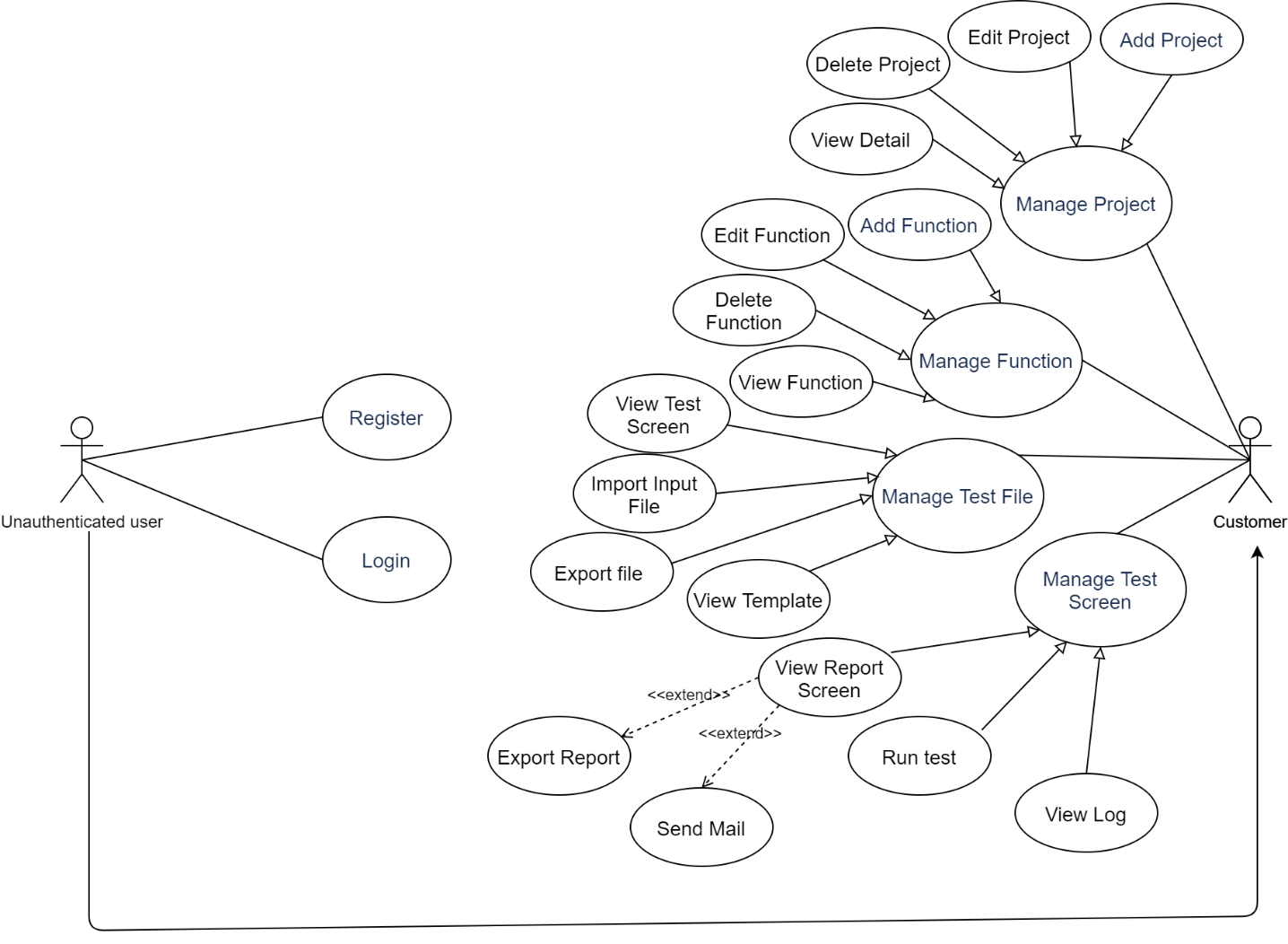


Figure : Use Case Diagram

|  |  |
| --- | --- |
| **Actor Name** | **Description** |
| Unauthenticated user | Who doesn't login to the system |
| Customer | Who logged in into the system using a customer account |

Table : Use Case Description

### Use case Description

#### <Customer> Manage Test Screen

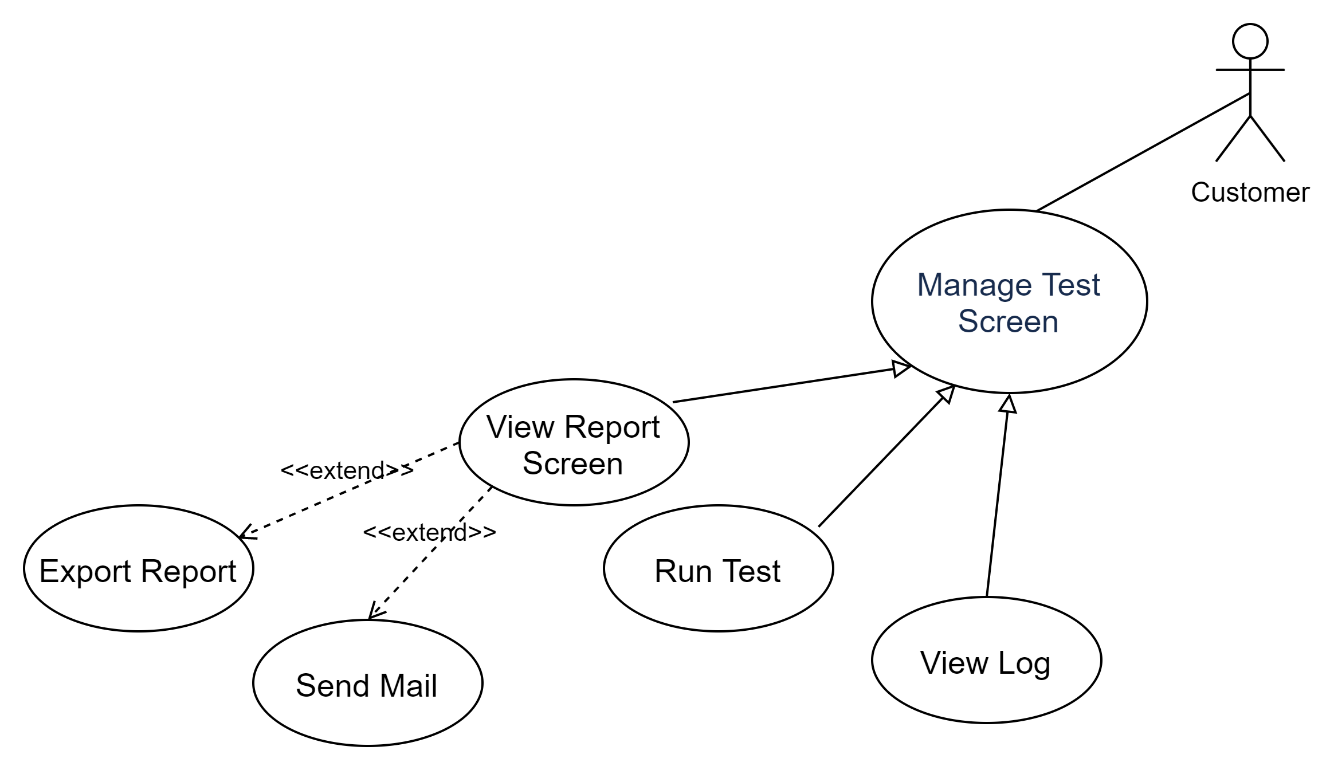


Figure : <Customer> Manage Test Screen

##### <Customer> Run Test

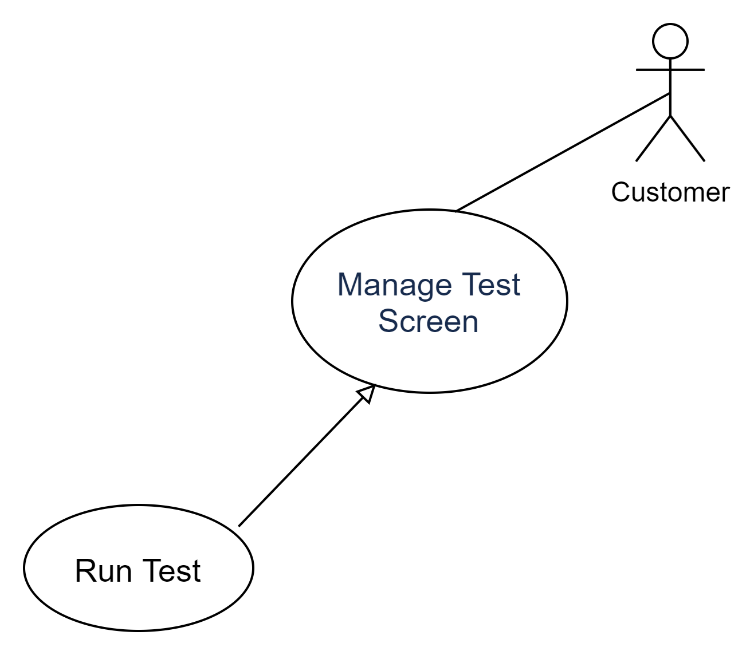


Figure : <Customer> Run Test

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE-AUTO\_UC\_01** | | | |
| **Use Case No.** | AUTO\_UC\_01 | **Use Case Version** | 1.0 |
| **Use Case Name** | Run Test | | |
| **Author** | Diem Vo | | |
| **Priority** | High | | |
| **Actor:**  - Customer  **Summary:**  - Allow to customer run input file with automation test.  **Goal:**  - Customers can execute automated testing.  **Triggers:**  - The customer selects the input file to be executed.  - The customer clicks on the run icon to send the execution request to the system.  **Preconditions:**  **-** Input the file in the correct format.  **Post Conditions:**  **- Success:** Run file success and show log run.  **- Fail:** System shows error messages.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | Customer selects input file. | The system will navigate to Test Screen. | | **2** | Customer clicks “Run” icon. | The system will execute the test automatically and show a log after a run. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** | Customer selects input file. | System shows error messages. |   **Exceptions: N/A**  **Relationships: N/A**  **Business Rules:**  **-** Input the file in the correct format. | | | |

Table : <Customer> Run Test

##### <Customer> Export Report

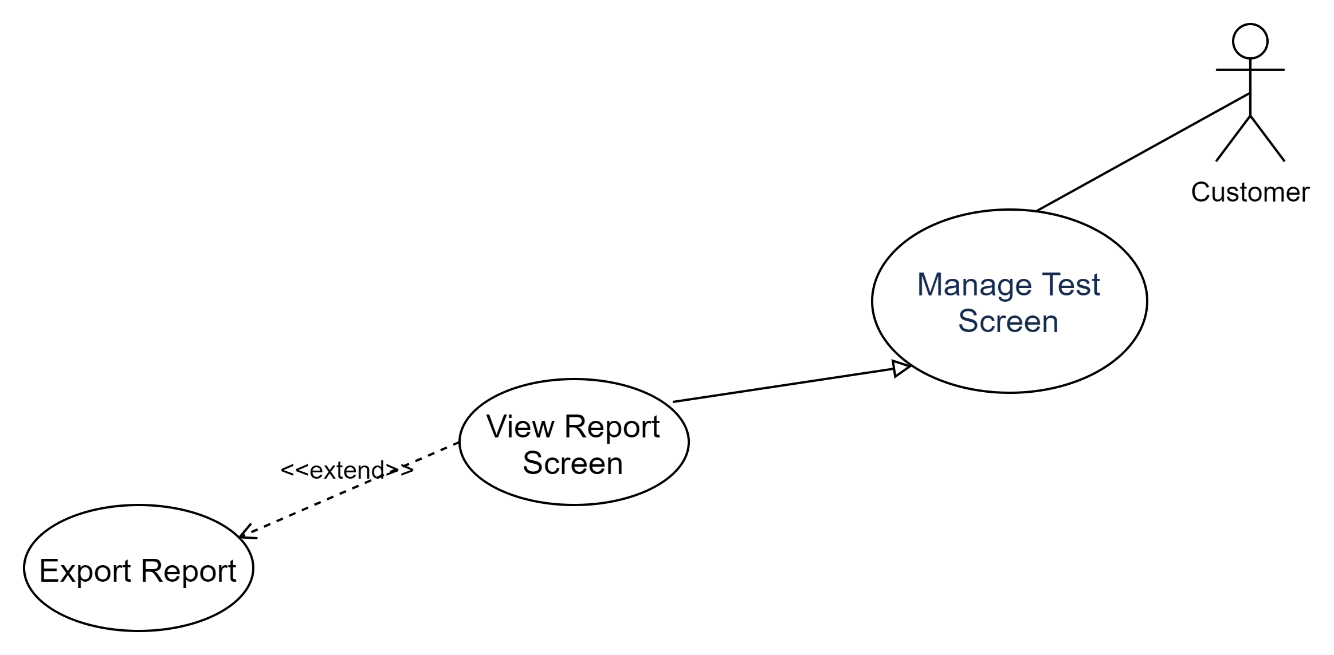


Figure : < Customer> Export Report

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE-AUTO\_UC\_02** | | | |
| **Use Case No.** | AUTO\_UC\_02 | **Use Case Version** | 1.0 |
| **Use Case Name** | Export Report | | |
| **Author** | Diem Vo | | |
| **Priority** | Normal | | |
| **Actor:**  - Customer  **Summary:**  - After executing a test file, export a report file and allow the customer to export it to a personal computer.  **Goal:**  - The customer to export a report file to a personal computer.  **Triggers:**  - The customer login with the customer account.  - The customer selects the input file to execute the test.  - The customer selects the export file button to download to the computer.  **Preconditions:**  **-** The input file has been executed.  **Post Conditions:**  **- Success:** Download file success.  **- Fail:** System shows error messages.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | Customer selects export report button. | System open popup for user to choose the path to save the file. | | **2** | The customer chooses the path. |  | | **3** | Click “OK”. | File report will be downloaded to the correct path. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** | Customer clicks “Cancel” | System close popup. |   **Exceptions: N/A**  **Relationships: N/A**  **Business Rules:**  **-** The input file has been executed. | | | |

Table : <Customer> Export Report

##### <Customer> Send Mail

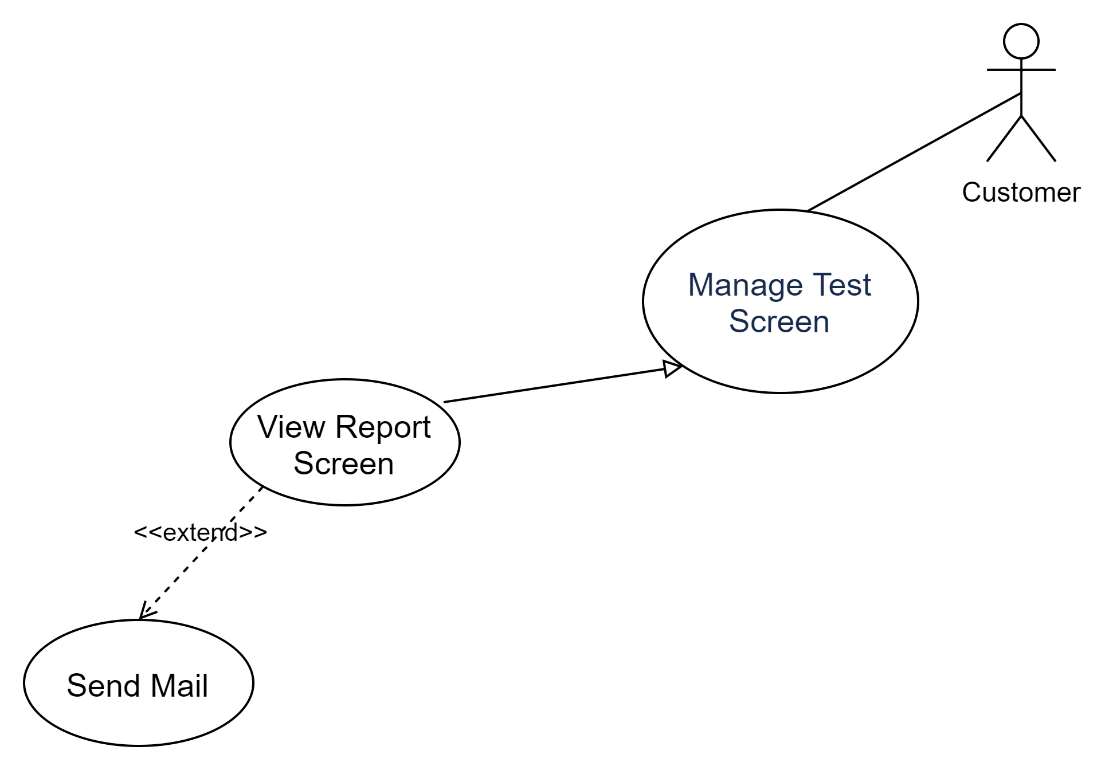


Figure : <Customer> Send Mail

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE-AUTO\_UC\_03** | | | |
| **Use Case No.** | AUTO\_UC\_03 | **Use Case Version** | 1.0 |
| **Use Case Name** | Send Mail | | |
| **Author** | Diem Vo | | |
| **Priority** | Normal | | |
| **Actor:**  - Customer  **Summary:**  - After executing a test file, export a report file and allow the customer to send it to mail.  **Goal:**  - The customer to send a report file to a personal mail.  **Triggers:**  - The customer login with the customer account.  - The customer selects the input file to execute the test.  - The customer selects the “Send Mail” button send a report file to a personal mail.  **Preconditions:**  **-** The input file has been executed.  **Post Conditions:**  **- Success:** Send a file to mail success.  **- Fail:** System shows error messages.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | Customer clicks the “Send Mail” button |  | | **2** |  | File report will be sent to mail. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** | The input file has not been executed. | The system will not send reports via email. |   **Exceptions: N/A**  **Relationships: N/A**  **Business Rules:**  **-** The input file has been executed.  - Gmail must be structured and precise. | | | |

Table : <Customer> Send Mail

#### <Customer> Manage Test File

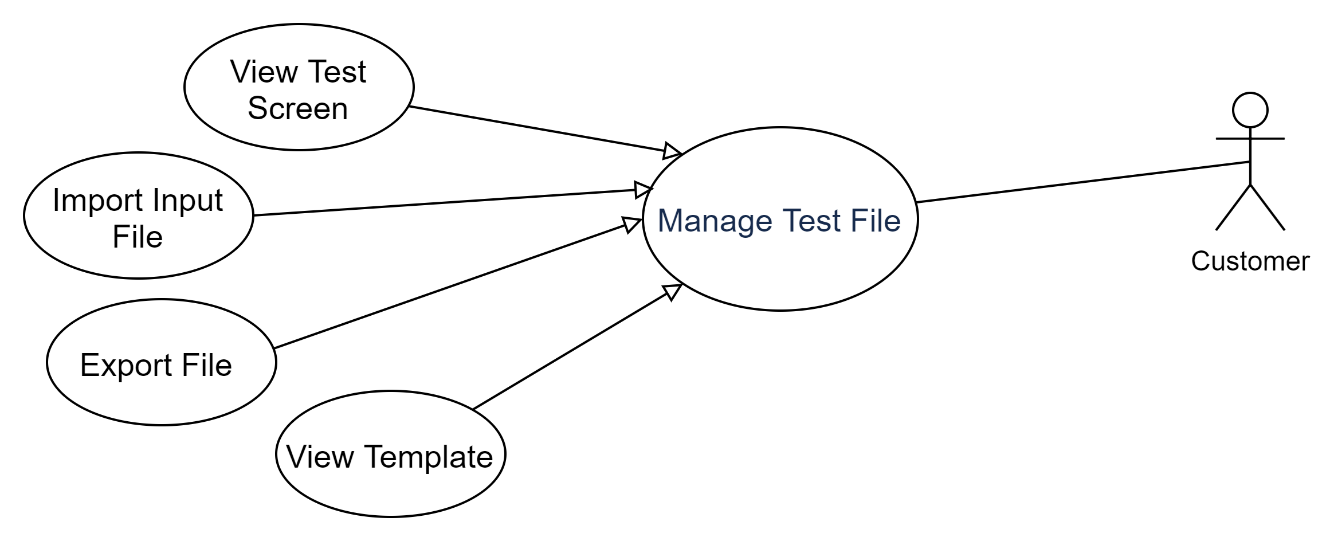


Figure : <Customer> Manage Test File

##### <Customer> View Template



Figure : <Customer> View Template

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE-AUTO\_UC\_04** | | | |
| **Use Case No.** | AUTO\_UC\_04 | **Use Case Version** | 1.0 |
| **Use Case Name** | View Template | | |
| **Author** | Diem Vo | | |
| **Priority** | High | | |
| **Actor:**  - Customer  **Summary:**  - Allows the customer to view the template of the input file.  **Goal:**  - The customer can view the template of the input file.  **Triggers:**  - The customer login with the customer account.  - The customer selects the “Template” label to view the template.  - The system will automatically download the sample file to the customer's computer.  **Preconditions:**  - The client login with the Customer role.  **Post Conditions:**  **- Success:** Download file success.  **- Fail:** System shows error messages.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | The customer clicks “Template” label. |  | | **2** |  | The system will automatically download the sample file. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** | The customer clicks “Template” label. |  | | **2** |  | The System shows error messages. |   **Exceptions: N/A**  **Relationships: N/A**  **Business Rules: N/A** | | | |

Table : <Customer> View Template

##### <Customer> Import Input File



Figure : <Customer> Import Input File

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE-AUTO\_UC\_05** | | | |
| **Use Case No.** | AUTO\_UC\_05 | **Use Case Version** | 1.0 |
| **Use Case Name** | Import Input File | | |
| **Author** | Diem Vo | | |
| **Priority** | High | | |
| **Actor:**  - Customer  **Summary:**  - Allow the customer to upload the input file to web.  **Goal:**  - The customer can upload the input file to executed test.  **Triggers:**  - The customer login with the customer account.  - The customer selects the files to be uploaded on the computer to the page and clicks the "import" button.  - The client sends the command to add files to the function of the system.  **Preconditions:**  - The customer has selected the file.  **Post Conditions:**  **- Success:** The customer successfully updated the input file and can proceed with the test execution.  **- Fail:** System shows error messages.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | The customer clicks “Import” button. | System open popup for user to choose the input file. | | **2** | The customer select file .CSV. |  | | 3 | Click “OK” button. | The input file will be check and upload to the web. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** | Click “Cancel” button. | The system does not upload files. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | **1** | Input file is not in the correct format. | System shows error messages. |   **Relationships: N/A**  **Business Rules:**  **-** The input file must be in the correct format. | | | |

Table : <Customer> Import Input File

##### <Customer> Export File



Figure : <Customer> Export File

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE-AUTO\_UC\_06** | | | |
| **Use Case No.** | AUTO\_UC\_06 | **Use Case Version** | 1.0 |
| **Use Case Name** | Export File | | |
| **Author** | Diem Vo | | |
| **Priority** | Normal | | |
| **Actor:**  - Customer  **Summary:**  - Allow the customer to download the input file to a personal computer.  **Goal:**  - The customer can download the input file to a personal computer.  **Triggers:**  - The customer login with the customer account.  - The customer selects the files to be download on the web to a personal computer and clicks the "Export" icon.  - The customer sends the command to save files to the system.  **Preconditions:**  - The customer has selected the file.  **Post Conditions:**  **- Success:** The customer successfully the input file to a personal computer.  **- Fail:** System shows error messages.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | The customer clicks “Import” icon. | System open popup for user to choose the path to save the file. | | **2** | The customer chooses the path. |  | | 3 | Click “OK” button. | File will be downloaded to the correct path. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** | Click “Cancel” button. | The system does not download files. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | **1** | The customer does not choose the path. | System shows error messages. |   **Relationships: N/A**  **Business Rules:**  **-** The file path must be correct. | | | |

Table : <Customer> Export File

#### <Customer> Add Function

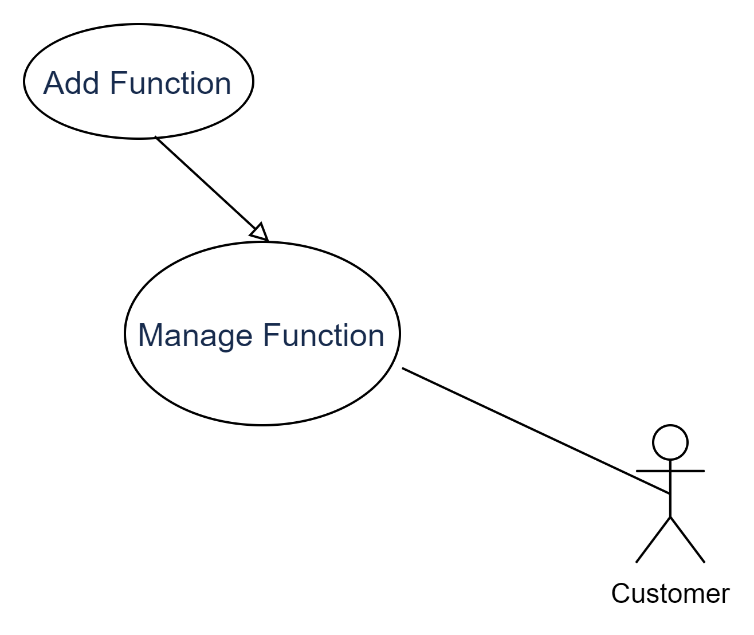


Figure : <Customer> Add Function

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE-AUTO\_UC\_07** | | | |
| **Use Case No.** | AUTO\_UC\_07 | **Use Case Version** | 1.0 |
| **Use Case Name** | Add Function | | |
| **Author** | Diem Vo | | |
| **Priority** | Normal | | |
| **Actor:**  - Customer  **Summary:**  - Allow customers to add projects to the system.  **Goal:**  - Customers can add projects to the system.  **Triggers:**  - The Customer clicks the "Add Function" button, enters information, and presses the "Submit" button.  - The Customer sends the command to add the function to the system.  **Preconditions:**  - The client login with the Customer role.  **Post Conditions:**  **- Success:** Add function success and can manage the function.  **- Fail:** System shows error messages.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | The Customer clicks the "Add Function" button. | System open popup for user to enter information:   * Function Name: Text | | **2** | The customer click “Submit” button. | The system reports successful information updates and closes the popup. |   **Alternative Scenario: N/A**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** | The customer click “Cancel” button. | The system will close popup. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | **1** | The customer entered the wrong information. | System shows error messages. |   **Relationships: N/A**  **Business Rules:**  - Function information cannot be left blank.  - The function has not been created before. | | | |

Table : <Customer> Add Function.

#### <Customer> Add Project

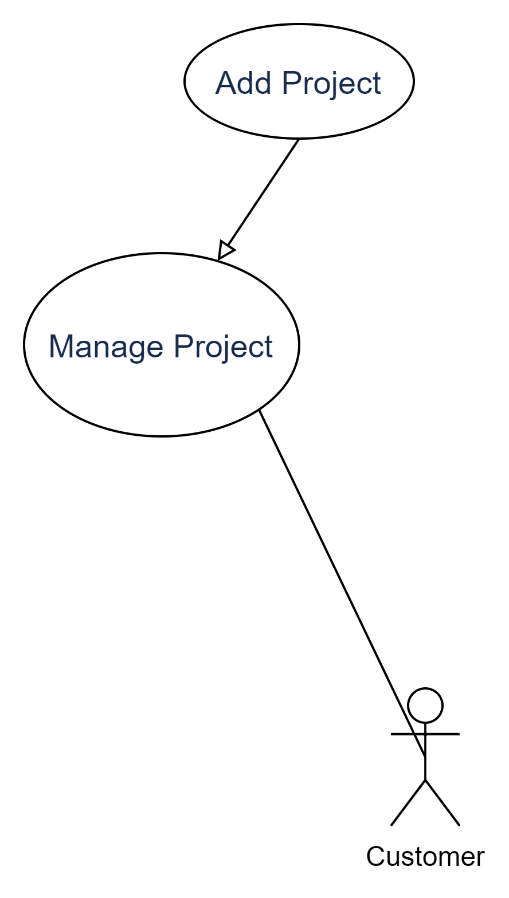


Figure : <Customer> Add Project

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE-AUTO\_UC\_08** | | | |
| **Use Case No.** | AUTO\_UC\_08 | **Use Case Version** | 1.0 |
| **Use Case Name** | Add Project | | |
| **Author** | Diem Vo | | |
| **Priority** | Normal | | |
| **Actor:**  - Customer  **Summary:**  - Allow customers to add projects to the system.  **Goal:**  - Customers can add projects to the system.  **Triggers:**  - The Customer clicks the "Add Project" button, enters information, and presses the "Submit" button.  - The Customer sends the command to add the project to the system.  **Preconditions:**  **-** The client login with the Customer role.  **Post Conditions:**  **- Success:** Add project success and can manage the project.  **- Fail:** System shows error messages.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | The Customer clicks the "Add Project" button. | System open popup for user to enter information:   * Project Name: Text * Link: Text | | **2** | The customer click “Submit” button. | The system reports successful information updates and closes the popup. |   **Alternative Scenario: N/A**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | **1** | The customer click “Cancel” button. | The system will close popup. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | **1** | The customer entered the wrong information. | System shows error messages. |   **Relationships: N/A**  **Business Rules:**  - Project information cannot be left blank.  - The weblink must be exactly.  - The project has not been created before. | | | |

Table : <Customer> Add Project

## Sequence Diagram

### <Customer> Create Project



Figure : <Customer> Create Project

### < Customer > Login



Figure : <Customer > Login

### <Customer> Register



Figure : <Customer> Register

### <Customer> Run Test



Figure : <Customer> Run Test

### <Customer> Import File



Figure : <Customer> Import File

### <Customer> Send Mail

## Architecture diagram

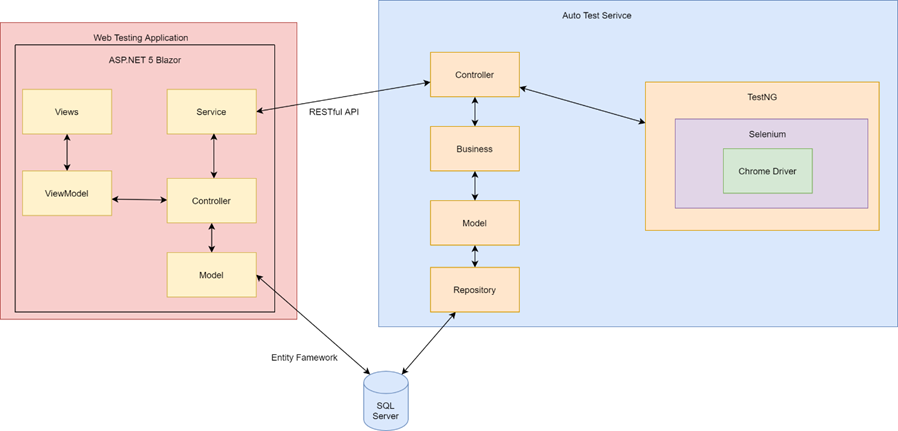


Figure 21: Architecture diagram

## Database Design

### ER Diagram

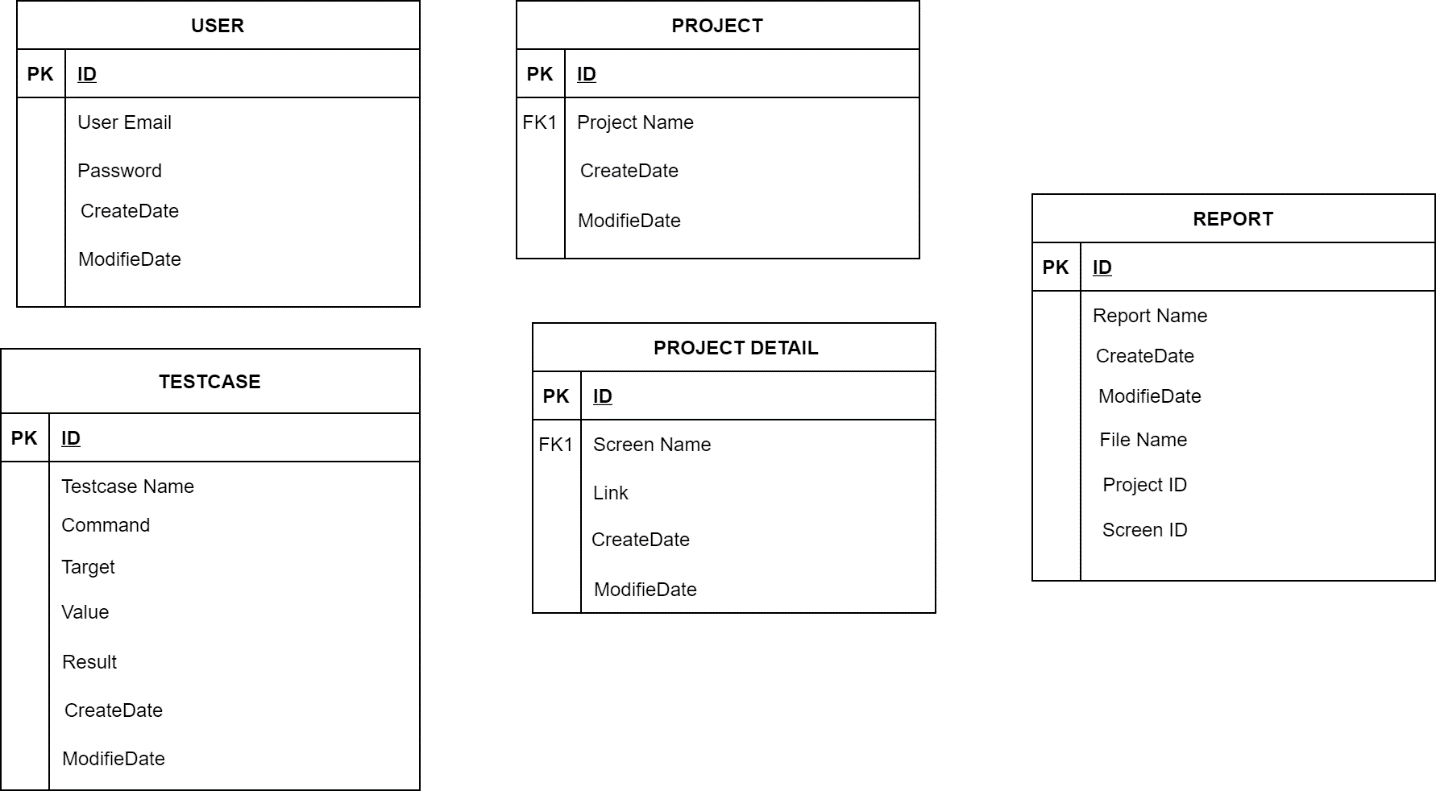


Figure 22: ER Diagram

### Database Diagram

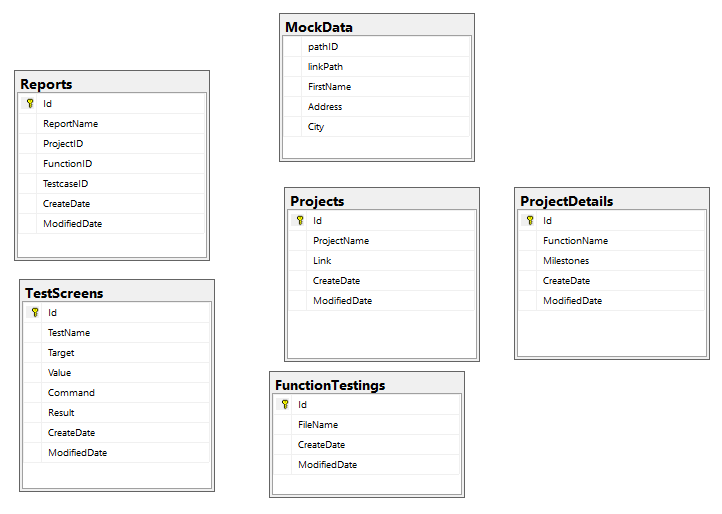


Figure 23: Database Diagram

|  |  |
| --- | --- |
| **Table** | **Description** |
| Projects | Is a place to store project information. |
| FunctionTestings | Is a place to store file input information. |
| TestScreens | Is a place to store result and detail of file input information. |
| ProjectDetails | Is a place to store function information. |
| Reports | Is a place to store report information. |
| MockData | Is a place to store list path link of files. |

Table 10: Database Description

## User Interfaces Design

|  |  |
| --- | --- |
| **Screen flow** | **Description** |
| Manage Projects | Projects list display |
| Manage Functions | The screen shows a list of functions on one screen |
| Manage Test Files | Displays list of input files of a function |
| Test Screen | Screen displays list of test cases in the input file and displays log. |

Table 11: User Interfaces Design

### Login screen

### Register

### Dashboard screen

Figure : Dashboard screen

### Management Project screen

Figure : Management Project screen

### Management Function screen

Figure 26: Management Function screen

### Management File screen

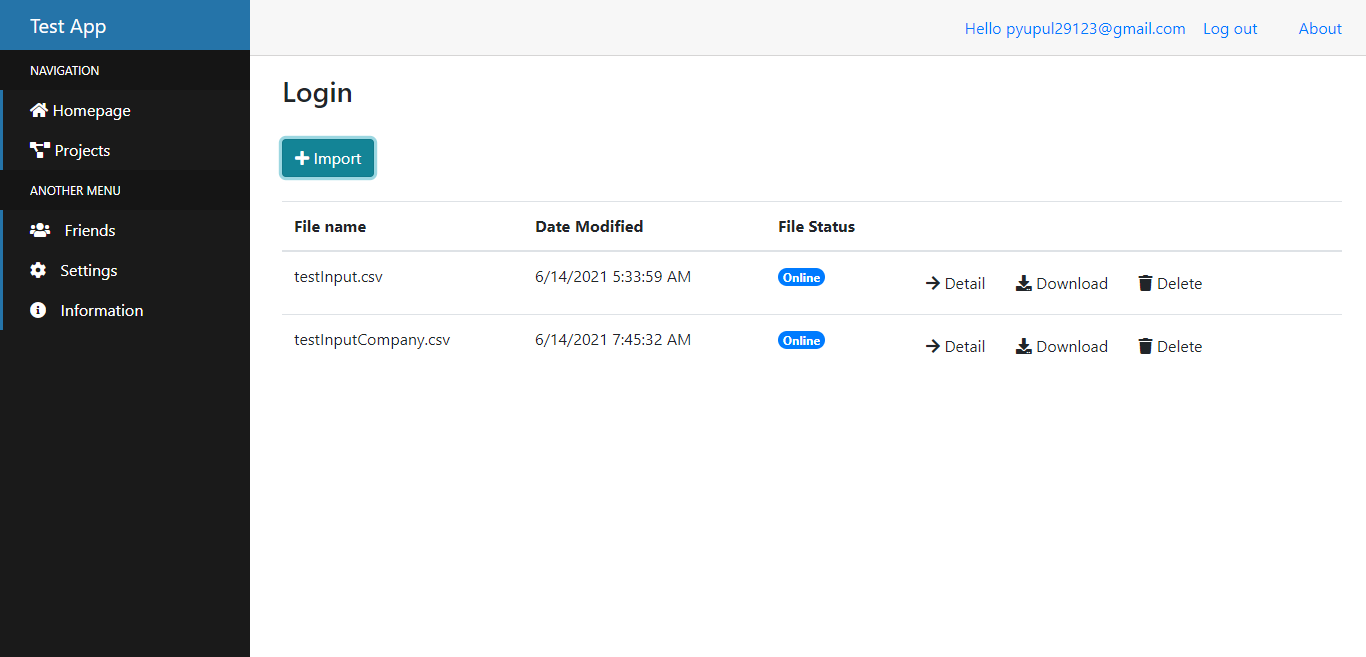


Figure 27: Management File screen

### Test screen

### Report screen

# Chap 4: SYSTEM IMPLEMENTATION



## Software development environment

Backend:

* Develop in Visual Code IDE, Eclipse IDE.

Frontend:

* Develop in Visual Code IDE

## Source code management

Any source code is managed by Github.

Link github:

Customer account: Test123/1234

# Chap 5: 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/

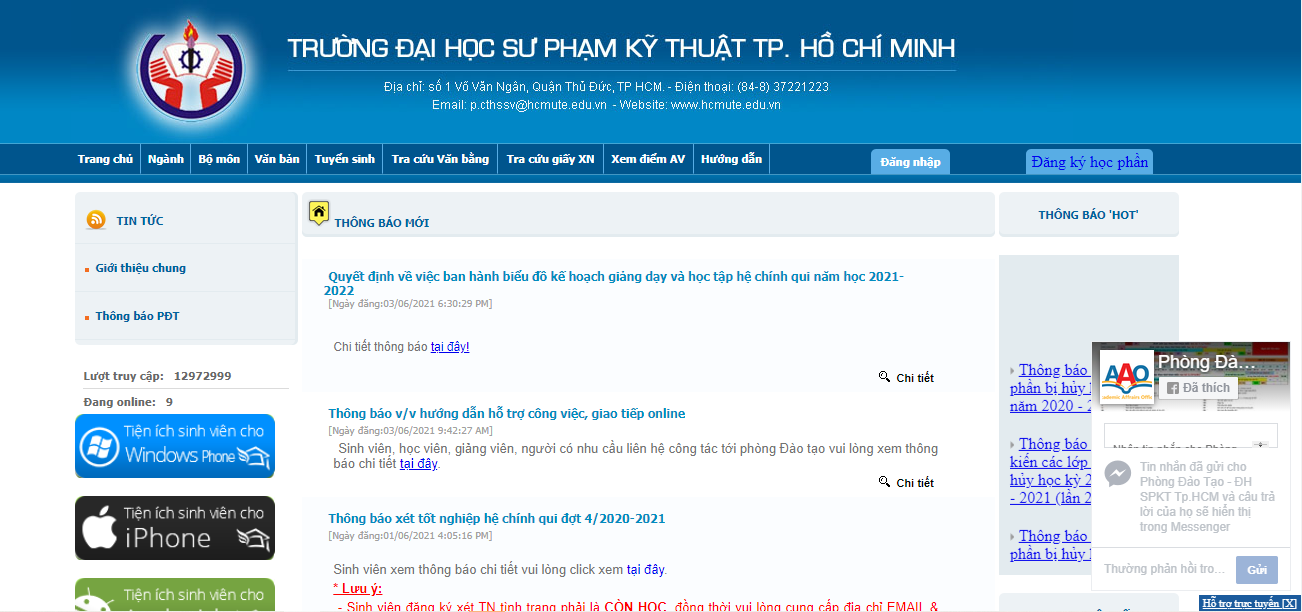


Figure 28: UTE Portal

## Testcase

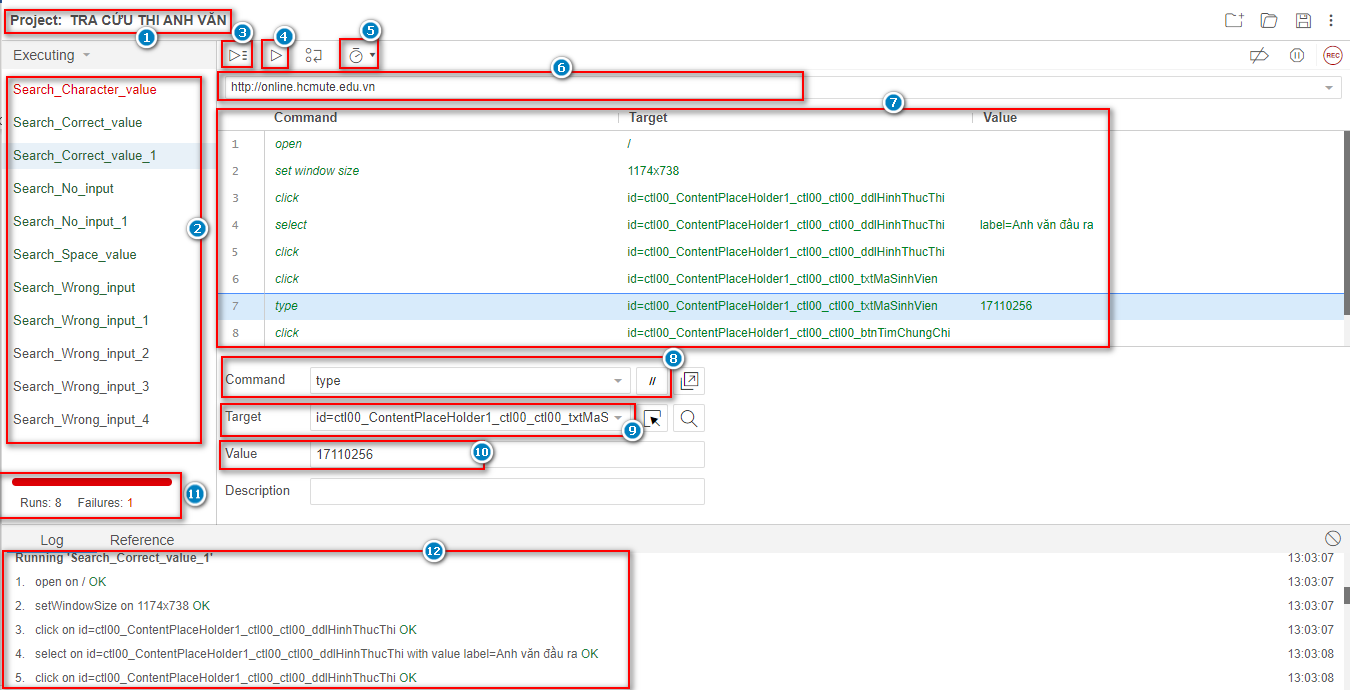


Figure 29: Test Case Selenium

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 Excution 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.

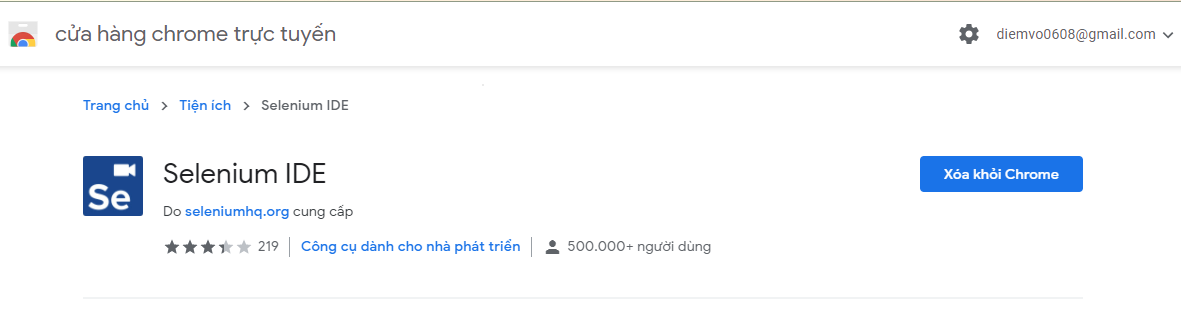


Figure 30: Install Selenium IDE

1. Turn on Selenium IDE on the extension bar.



Figure 31: Turn on Selenium IDE

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



Figure 32: Create a new project

1. On the pop-up dialog box:

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

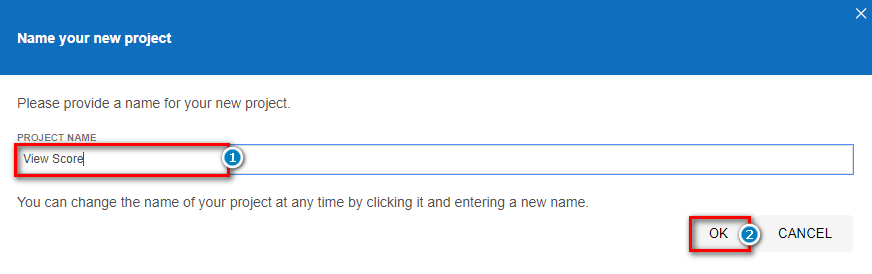


Figure 33: Create a new project (1)

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

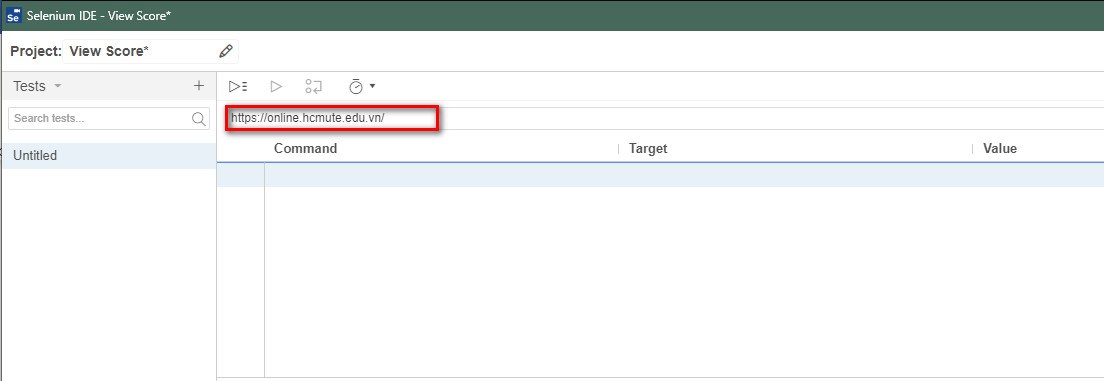


Figure 34: BaseUrl

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



Figure 35: 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.

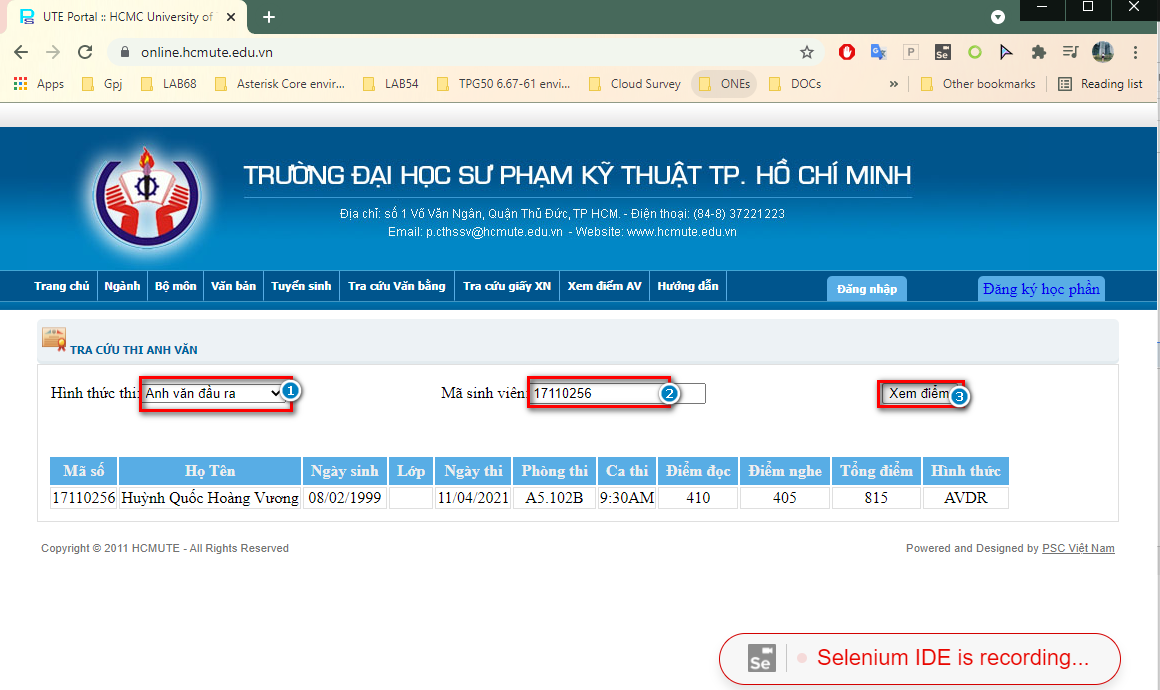


Figure 36: Recording

1. On the pop-up dialog box:

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

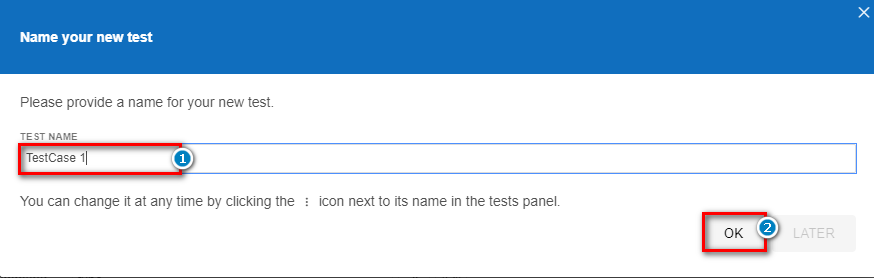


Figure 37: Test Name

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

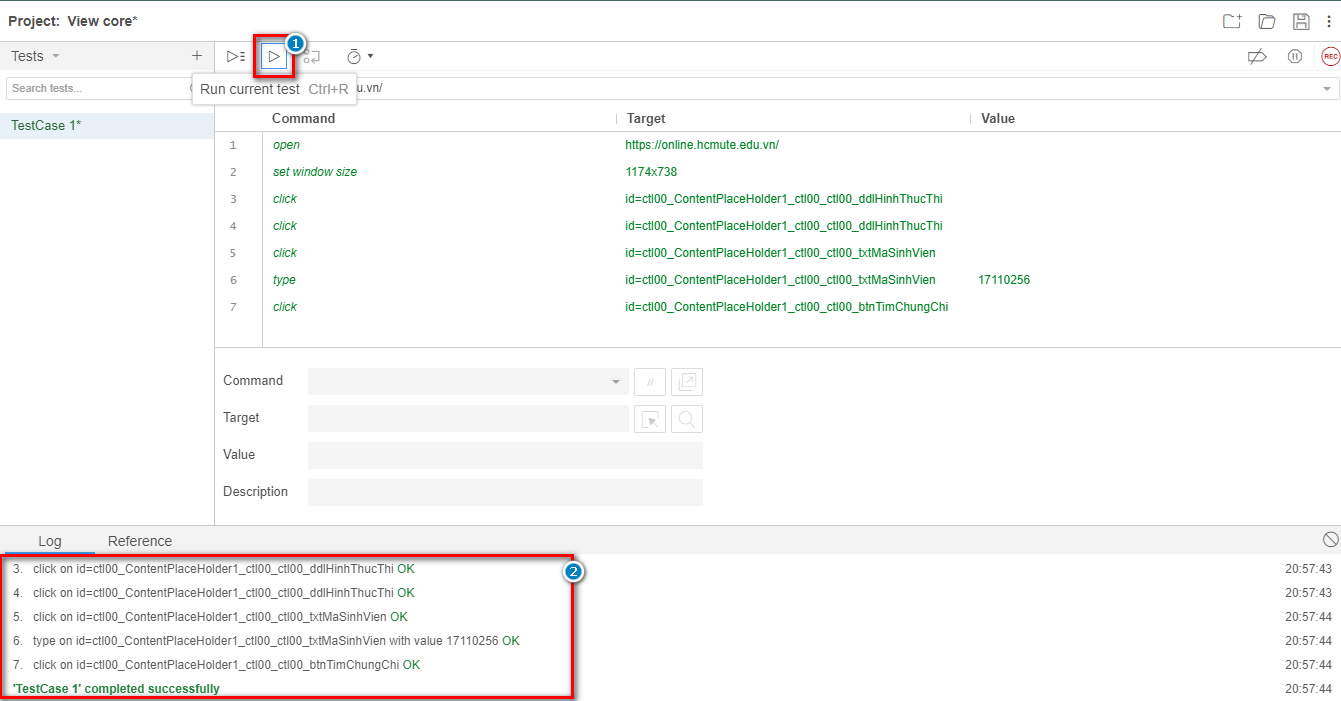


Figure 38: Detail test case

* 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).

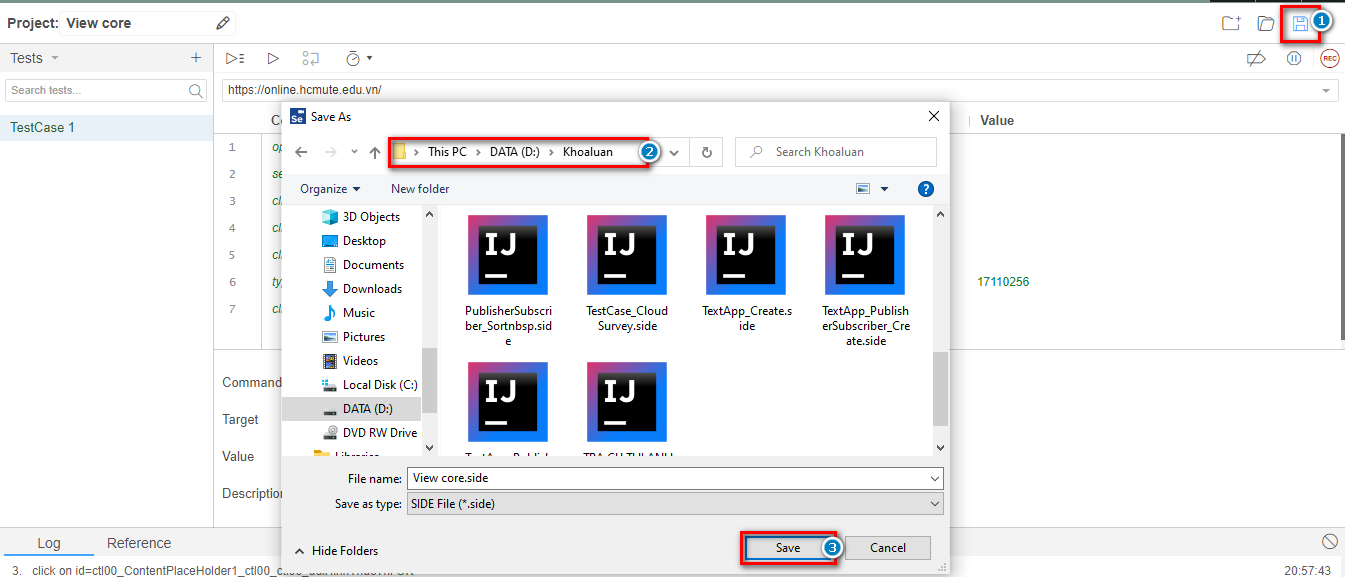


Figure 39: Save test case

## 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.

# Chap 6: CONCLUSION AND RECOMMENDATIONS

## Conclusion

## Advantage

## Disadvantage

## Future planning

# REFERENCES

[1] TestNG Tutorial: What is, Annotations & Framework in Selenium

https://www.guru99.com/all-about-testng-and-selenium.html

[2] What is ASP.NET?

https://dotnet.microsoft.com/learn/aspnet/what-is-aspnet

[3] What's new in SQL Server 2017, 10/20/2017

https://docs.microsoft.com/en-us/sql/sql-server/what-s-new-in-sql-server-2017?view=sql-server-ver15

[4] Blazor

https://dotnet.microsoft.com/apps/aspnet/web-apps/blazor

[5] Selenium IDE Tutorial for Beginners

https://www.guru99.com/introduction-selenuim-ide.html



# APPENDICES

# JOB TABLE OF WORK

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

Table 12: Job table of work