**Individual Project 03: Test Plans and Cases**

A Web Service to Manage User’s Password

Hong Van

CS 504 – Software Engineering

Instructor: Shane Frigon

City University of Seattle

August 15, 2021

**Introduction**

Testing has become an important part of the software development lifecycle. A text plan is the schematic for covering assessment of software functionality. It details each step that leads to the test outcome, underscoring projected risks, and the required resources for software application. Hence, creating a great software test plan remains essential. A test case is a set of actions executed to verify a particular feature of functionality of the software application. It includes test steps, test data, precondition, postcondition developed for specific test scenario to verify any requirement. In this project, the author will create a test case and plan for the system that designed for the previous assignment, which is a web service to manage user’s password. The author will also explain the process model that used for the development of the system.

**Test Plans**

*Project testing objectives and scopes are:*

* Ensure the AUT conforms to functional and non-functional requirements
* Make sure the AUT meets the quality specifications defined by the users
* Focusing on testing all the login functionalities of the web application

*Roles and Responsibilities:*

|  |  |
| --- | --- |
| **Member** | **Roles and Responsibilities** |
| Test Manager | Manage the whole project, create project direction, and define appreciate resources |
| Tester | Implement test cases and do test report |
| Test Administrator | Build up and ensure test environment for test execution |
| SQA Members | Responsible for making sure the testing process is meeting specified requirement |

*Test Methodology:*

In this project, there are 3 types of testing should be conducted:

* Unit testing: Testing a specific feature, which is Login
* System testing: Test the entire integrated system against its requirements
* Compatibility testing may be used to test the software on different hardware, OS, and environment.

*Test Environment:* the test environment for this project is similar to the figure 2 below*:*

Diagram

Description automatically generated

*Figure 2: Test Environment. Source:* [*https://www.guru99.com/what-everybody-ought-to-know-about-test-planing.html*](https://www.guru99.com/what-everybody-ought-to-know-about-test-planing.html)

*Test Deliverables:*

There are different test deliverables at each phase of the testing including before, during, and after the testing. The test deliverables are provided before testing phase are test plans document, test cases document, and test design specifications. The test deliverables during the testing include test scripts, test data, simulators, and test matrix. The test deliverables are provided after the testing are test reports, test procedure guidelines, and notes (Guru99, 2021).

**Test Cases**

**End-to-End test: User Experience**

To use the web service, the users will need to create an account and then, logging in to the web. The user’s login interface will look like the figure 1 below. In this part, the author is going to write three test cases for the scenario: Check Login Functionality. The purpose of these tests is to answer the following questions below:

* What happens when user login with a valid email address and password?
* What happens when an invalid email address and password are entered or either email address or password fields are left blank?
* Is the FORGOT PASSWORD? option working as expect?

Graphical user interface, application

Description automatically generated

*Figure 1: User Login Interface Example. Source:* [*https://lastpass.com/?ac=1*](https://lastpass.com/?ac=1)

To implement the test, testers would need data, which is email address and password. In addition, the testers will need to perform a specific set of actions on the application under test (AUT). These steps are open the login page, enter email address, input password, and click LOG IN or click FORGOT PASSWORD? to execute each of the test case. After that, the testers will check the result to see if it happens as expected and then, assign a pass or fail status for each test case.

Precondition for implementing these tests is having a browser installed and Internet connection to have access to the web page. The postcondition would be the number of user login and the time and date of login that is stored in the database.

The table below shows the detail information for each test case:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Results** | **Pass/Fail** |
| TC01 | Check user login with correct email address and password | 1.Open the web  2.Enter Email address  3.Enter password  4.Click Login | Email Address: [jane.doe@mail.com](mailto:jane.doe@mail.com)  Password: P@ssw0rd | User should login into an application successfully | As expected | Pass |
| TC02 | Check user login with incorrect email or password | 1.Open the web  2.Enter Email address  3.Enter password  4.Click LOG IN | Email Address: [jane.doe@mail.com](mailto:jane.doe@mail.com)  Password: P@ssword | User fails to login into an application | As expected | Pass |
| TC03 | Check FORGOT PASSWORD? option | 1.Open the web  2.Enter Email address  4.Click FORGOT PASSWORD? | Email Address: [jane.doe@mail.com](mailto:jane.doe@mail.com) | Send a link to the email address that user enter to reset their password. | As expected | Pass |

**Reference:**

Bose, S. (May 10, 2021). *How to create test scenarios with examples.*

https://www.browserstack.com/guide/how-to-create-test-scenarios

Copeland, D. (March 30, 2016). *Full Stack Testing: Balancing Unit and End-to-End Tests*.

https://www.infoq.com/articles/balancing-unit-and-end-to-end-tests/

Lastpass. (2021). *Log In*.

https://lastpass.com/?ac=1

Guru99. (2021). Test Plan: *What is, How to create (with Example).*

https://www.guru99.com/what-everybody-ought-to-know-about-test-planing.html