Test Plan

QA Course Project - Front End

21/05/2022

Ligia Samara Diaz Hirashi

Version: 1.1

Created: 21/05/2022

Last Updated: 21/06/2022

Status: DRAFT

Location: Monterrey, N.L. Mexico.

Revision and Sign-off Sheet

Document History-

Version	Date	Author	Description of Change
1	21/05/2022	Ligia Samara Diaz Hirashi	Draft
1.1	16/06/2022	Ligia Samara Diaz Hirashi	Added Budget

Approvers List-

Name	Role	Approver/Reviewer	Approval/Review Date

Reference Documents-

Version	Date	Document Name
1.0	21/05/2022	QA Course Project (Front End)
1.1	16/06/2022	QA Course Project (Front End)
1.1	17/06/2022	QA Course Project Gantt

1. Introduction

1.1 Purpose

This test plan describes the testing approach and overall framework that will drive the testing of the *UI Test Automation Playground* website. The document introduces:

- **Test Strategy:** Rules the test will be based on, including the givens of the project (e.g.: start / end dates, objectives, assumptions); description of the process to set up a valid test (e.g.: entry/exit criteria, creation of test cases, specific tasks to perform, scheduling, data strategy).
- **Execution Strategy:** Describes how the test will be performed and process to identify and report defects, and to fix and implement fixes.
- **Test Management:** Process to handle the logistics of the test and all the events that come up during execution (e.g.: Communications, escalation procedures, risk and mitigation, team roster)

1.2 Project Overview

UI Test Automation Playground is a powerful tool providing a platform for sharpening UI test automation skills. The functionality of this tool helps users practice, learn and test their automation skills by emulating different automation pitfalls used in modern web applications. The contents of the website can be accessed by anyone and users are encouraged to use it to learn test automation techniques.

1.3 Audience

- Project team members perform tasks specified in this document, and provide input and recommendations about this document.
- Project Manager plans for the testing activities in the overall project schedule, reviews the document, tracks the performance of the test according to the specified task, approved the document and is accountable for the results.
- Technical Team ensures that the test plan and deliverables are in line with the design, provides the environment for testing and follows the procedures related to the fixes of defects.

2. Test Strategy

2.1. Test Objectives

The objective of the test is to verify that the functionality of the *UI Testing Playground* website works according to the specifications.

The test will verify the following aspects of *UI Testing Playground*.

Website:

- **Dynamic ID:** Verify that dynamic IDs of elements are not being used while testing.
- **Class Attribute:** Check that class attribute based XPath is well formed.
- **Hidden Layers:** Verify that your test does not interact with elements invisible because of z-order.
- **Load Delay:** Ensure that a test is capable of waiting for a page to load.
- **AJAX Data:** Check that some elements appear on a page after loading data with AJAX request.
- **Client Side Delay:** Verify that some elements appear after client-side time consuming JavaScript calculations.
- **Click:** Corroborate that an event based click on an element may not always work.
- **Text Input:** Ensure that entering text into an edit field may not have effect.
- **Scrollbars:** Validate that an element is visible on screen while scrolling.
- **Dynamic Table:** Verify cell value in a dynamic table.
- **Verify Text:** Verify the contents of a text within a page.
- **Progress Bar:** Check that the progress is carried upon completion.
- **Visibility:** Ensure that an element is visible on screen.
- **Sample App:** Fill in and submit a form to authenticate login.
- **Mouse Over:** Verify that mouseover leads to changes in the DOM tree.
- **Non-Breaking Space:** Check the usability of non breaking spaces while finding selectors.
- **Overlapped Element:** Validate scrolling while entering text into a partially visible element.
- **Shadow DOM:** Check that shadow DOM components generate a guid.
- **UI:** Check that the interface of the website such as UI is working as expected.
- **Usability:** Verify the usability of the website.

2.2. Test Assumptions

General

- The system will be treated as a black box, if the information shows correctly online and in the reports, it will be assumed that it's working properly.
- Unit testing and acceptance testing will be considered for this estimation.
- The test team will perform unit and functionality testing simultaneously in the *UI Test Automation Playground* webpage.

2.3. Test Principles

- Testing will be focused on meeting the business objectives, cost efficiency, and quality.
- Testing processes will be well defined, yet flexible, with the ability to change as needed.
- Testing environment and data will emulate a production environment as much as possible.
- Testing will be divided into distinct phases, each with clearly defined objectives and goals.
- There will be entrance and exit criteria.

2.4. Data Approach

- During testing for the website, the tester will provide the data used for the testing activities.
- In functionality features testing, the UI Test Automation Playground will specify pre-loaded data which will be used for the testing activities.

2.5. Scope and Levels of Testing

- In the testing for the website, 2 types of testing should be conducted.
- **Unit testing:** Small pieces of verifiable software in the application.
- **Acceptance testing:** Verifies that application features work as per the software requirements.

2.5.2. Features to be tested

Module Name	Applicable Roles	Description
Dynamic ID	Tester	A tester makes sure that the dynamic IDs of elements are not being used while testing.

Module Name	Applicable Roles	Description
Class Attribute	Tester	A tester checks that class attribute based XPath is well formed.
Hidden Layers	Tester	A tester verifies that the test does not interact with elements invisible because of z-order.
Load Delay	Tester	A tester ensures that a test is capable of waiting for a page to load.
AJAX Data	Tester	A tester verifies that some elements appear on a page after loading data with AJAX request.
Client Side Delay	Tester	A tester checks that some elements appear after client-side time consuming JavaScript calculations.
Click	Tester	A tester validates that an event based click on an element may not always work.
Text Input	Tester	A tester validates that entering text into an edit field may not have effect.
Scrollbars	Tester	A tester checks how to scroll an element into view.
Dynamic Table	Tester	A tester verifies a cell value in a dynamic table.
Verify Text	Tester	A tester finds an element by displayed text.
Progress Bar	Tester	A tester follows the progress of a lengthy process until completion.
Visibility	Tester	A tester checks if an element is visible on screen.
Sample App	Tester	A tester checks a demo application with dynamically generated element attributes.
Mouse Over	Tester	A tester verifies that placing mouse over an element may change DOM and make the element unavailable.
Non-Breaking Space	Tester	A tester validates non-breaking spaces while testing.
Overlapped Element	Tester	A tester verifies that there's a visible element to enter text.
Shadow DOM	Tester	A tester looks inside a Shadow DOM component.

2.5.3. Features not to be tested

These features are not to be tested because they are not included in the software requirement specs.

- Website Security and Performance
- Hardware Interfaces

2.6. Test Type

Functional Test

Purpose: Functional testing will be performed to check the functionality of the application. Testing is carried out by feeding the input and validating the output from the application.

2.7. Test Risks and Mitigation Factors

Risk	Probability	Impact	Mitigation Plan
Schedule Testing schedule is tight. If the start of the testing is delayed, it cannot be extended beyond the deadline.	High	High	The testing team can control the preparation tasks to start as soon as possible.
Defects Defects are found at a late stage of the cycle, defects discovered late are most likely to be time consuming to resolve.	Medium	High	Defect management plan is in place to ensure prompt communication and fixing of issues.
Budget Wrong budget estimate and cost overruns.	Medium	High	Establish the scope before beginning to test.
Natural disasters	Low	Medium	Teams and responsibilities are spread between different geographic areas. In a catastrophic event in one of the areas, there will be team members available in the other areas to continue the testing activities.

2.8. Role Expectations

The following list defines in general terms the expectations related to the roles directly involved in the management, planning or execution of the test for the project.

	Roles	Name	Contact Information
1	Project Manager	Samara Diaz Hirashi	ligia.diaz@enroutesystems.com
2	Test Lead	Samara Diaz Hirashi	ligia.diaz@enroutesystems.com
3	Testing Team	Samara Diaz Hirashi	ligia.diaz@enroutesystems.com

2.8.1. Project Management

- Project Manager: Reviews the content of the test plan, test strategy and test estimates and signs off on it.

2.8.2. Test Planning (Test Lead)

- Develops a test plan and the guidelines to create test conditions, test cases, expected results and execution scripts.
- Provides guidelines on how to manage defects.
- Communicates to the test team any changes that need to be made.
- Acknowledges the completion of a section within a cycle.
- Gives the OK to start next level of testing.

2.8.3. Test Team

- Develops test conditions, test cases, expected results and execution scripts.
- Performs execution and validation.
- Identifies, documents and prioritizes defects according to the guidance provided by the test lead.

3. Test Logistics

The tester will start the test execution when all the following inputs are ready:

- Website is available for testing.
- Test Specification is created.
- Test Environment is built.

4. Test Criteria

4.1. Suspension Criteria

- If the team members report that there are 40% of test cases failed, testing will be suspended until the development team fixes the failed cases.

4.2. Entry Criteria

- The test environment, test tools and test data are available and ready to use.

4.3. Exit Criteria

- Specifies the criteria that denotes a successful completion of a test phase.
- Run rate is mandatory to be 100% unless a clear reason is given.
- Achieving the pass rate is mandatory.

Exit Criteria	Test Team	Notes
100% Test Scripts executed.	✅	
85% Pass Rate of Test Scripts.	✅	
No open Critical and High severity defects.	✅	
95% of Medium severity defects have been closed.	✅	
All remaining defects are either cancelled or documented as Change Requests for future release.	✅	
All expected and actual results are captures and documented with the test script.	✅	

4.4. Test Cycles

- There will be two cycles for functional testing. Each cycle will execute all the scripts.
- The objective of the first cycle is to identify any blocking, critical defects, and most of the high defects. It is expected to use some work-around in order to get to all the scripts.

- The objective of the second cycle is to identify remaining high and medium defects, remove the work-around from the first cycle, correct gaps in the scripts and obtain performance results.

4.5. Validation and Defect Management

- It is expected that the testers execute all the scripts in each of the cycles described above. However it is recognized that the testers could also do additional testing if they identify a possible gap in the scripts. If a gap is identified, the scripts and traceability matrix will be updated and then a defect logged against the script.
- It is the responsibility of the tester to open the defects, link them to the corresponding script, assign an initial severity and status, retest and close the defect.

Defects found during testing will be categorized according to the following categories:

Severity	Impact
1 (Critical)	• This bug is critical enough to crash the system, cause file corruption, or cause potential data loss. • It causes an abnormal return to the operating system (crash or a system failure message appears). • It causes the application to hang and requires re-booting the system.
2 (High)	• It causes a lack of vital program functionality with workaround.
3 (Medium)	• This Bug will degrade the quality of the System. However there is an intelligent workaround for achieving the desired functionality - for example through another screen. • This bug prevents other areas of the product from being tested. However other areas can be independently tested.
4 (Low)	• There is an insufficient or unclear error message, which has minimum impact on product use.
5 (Cosmetic)	• There is an insufficient or unclear error message that has no impact on product use.

4.6. Test Metrics

Test Metrics to measure the progress and level of success of the test will be developed and shared with the project manager for approval. Below are some of the metrics.

Report	Description
Test Preparation and Execution Status	To report on % complete, %pass or %fail. Defects severity wise Status - Open, closed or any other.
Execution Status	To report on pass, fail, total defects, highlights or critical defects.
Status Report	Project driven reporting (As requested by the PM)

5. Resource Planning

5.1. System Resources

No.	Resources	Description
1	Network	A stable internet line with the speed at at least 5 Mb/s.
2	Computer	At least 1 computer running macOs Catalina or higher.
3	Test tool	Visual Studio Code version 1.67.2 (Universal). WebdriverIO version 7 for browser automation, Mocha Framework as a plugin for WDIO and Typescript.

5.1.1. Test Tool

Visual Studio Code features a lighting fast source code editor, perfect for creating automation scripts. With support for hundreds of languages, VS Code helps you be instantly productive with syntax highlighting, bracket-matching, auto-indentation, box-selection, snippets, and more. WebdriverIO is a progressive automation framework built to automate modern web and mobile applications. It simplifies the interaction with your app and provides a set of plugins that help you create scalable, robust and stable test suites. Mocha is a simple, flexible and one of the widely adopted JS test framework. Mocha usually runs test serially which enables accurate reporting and asynchronous functions.

6. Test Environment

- *UI Test Automation Playground* servers will be hosted at the company's site. (http://uitestingplayground.com)

- A macOS environment with at least Google Chrome version 102.0.5005.115 and Visual Studio version 1.67.2 should be available to each tester.

7. Schedule and Estimation

7.1. Project tasks and estimation

Task	Members	Estimate Effort
Research and Discovery	Project Manager	4 hours
Test Planning	Test Lead	10 hours
Test Monitoring and Control	Test Lead	6 hours
Test Analysis	Test Lead and Tester	4 hours
Test Design	Tester	12 hours
Test Implementation	Test Lead	28 hours
Test Execution	Tester	9 hours
Test Completion	Test Lead and Tester	5 hours
Total		**78 hours**

7.2. Schedule to complete these tasks

Task Name	Start Date	End Date
Research and Discovery	29/05/2022	29/05/2022
Test Planning	29/05/2022	15/06/2022
Test Monitoring and Control	29/05/2022	26/06/2022
Test Analysis	16/06/2022	16/06/2022
Test Design	17/06/2022	19/06/2022
Test Implementation	16/06/2022	26/06/2022
Test Execution	24/06/2022	26/06/2022
Test Completion	26/06/2022	26/06/2022

7.3. Budget Estimation

Task Name	No. Of Team Members	Effort	Budget Estimation
Research and Discovery	1	4 hours	500 MXN
Test Planning	1	10 hours	1250 MXN
Test Monitoring and Control	1	6 hours	750 MXN
Test Analysis	2	4 hours	1000 MXN
Test Design	1	12 hours	1500 MXN
Test Implementation	1	28 hours	3500 MXN
Test Execution	1	9 hours	1125 MXN
Test Completion	2	5 hours	1250 MXN

Hourly pay per team member: 125 MXN.

8. Test Deliverables

Test deliverables are provided as below

8.1. Before testing phase

- Test plan document
- Test cases document
- Test specifications.

8.2. During the testing

- Test tool
- Simulators
- Test Data

8.3. After the testing

- Test result / reports
- Defect report

9. Approvals

The names and titles of people who must approve this plan.

Signature:	
Name:	
Role:	
Date:	