Test Plan

SauceDemo

11/28/2022
Nurhayat Köklü

Note: This is a test plan created for self-learning purposes

Version: 1.0
Created: 11/22/2022
Last Updated: 11/28/2022
Status: FINAL

Revision and Signoff Sheet

Document History

Version	Date	Author	Description of Change
1	11/22/2022	Nurhayat Köklü	Draft
2	11/25/2022	Nurhayat Köklü	Draft - Reviewed
3	11/28/2022	Nurhayat Köklü	Final

Reference Documents

Version	Date	Document Name
1.0	11/28/2022	SAUCEDEMO_VERSION_1.0 – TEST PLAN DOCUMENT
2.0	11/25/2022	SAUCEDEMO_VERSION_2.0 – TEST PLAN DOCUMENT
3.0	11/28/2022	SAUCEDEMO_VERSION_3.0 – TEST PLAN DOCUMENT

Table of Contents

1. INTRODUCTION

This test plan provides detailed information about the testing process of SAUCEDEMO– https://www.saucedemo.com/ website. The test plan consists of:

- Product Analysis section which provides general information about the product under test.
- Testing Strategy section which describes how the test will be performed and the processes to execute the test cases.
- Testing Objectives section which describes the overall goal and achievement of test execution.
- Test Criteria section which identifies the standard or rule on which the test procedure is based on.
- Resource Planning section which summarizes all types pf resources required to complete the project task.
- Test Environment section which describes the setup of software and hardware on which the test cases is going to be executed.
- Schedule section which includes an estimation abut the effort to complete the project.
- Test Deliverables section which lists all the documents, and other components that is going to be delivered during the software testing lifecycle.

2. PRODUCT ANALYSIS

SAUCEDEMO.COM is a sample website built by SauceLabs that allows testers to practice browser automation. It is designed as an e-commerce mock website where one should login in order to use the functionalities of the website.

When the users visit the website, the first thing they see is the login page with four valid usernames, a valid password and login button to login on the site and start using the functionalities that website provides.

After logging in, user sees products with their images, short descriptions, prices and a “ADD TO CART” button. There is also a filter on right top of the page, just below to cart to sort the products alphabetically and according to their price.

If user adds a product to the cart, they see how many items they have added to the cart directly on the cart on right corner. They can remove the item on products page as well. User can click to cart and see the items added to the cart. They can also remove the products from the cart on this list.

The cart page shows the quantity, description, name and the price of the product. User can select to CONTINUE SHOPPING option or CHECKOUT option on this page.

If user decides to CHECKOUT, they should fill a form. This form requires the name and the last name of the user and a postal code. After filling the form, a page appears with the quantity, description, name and the price of the product, payment information, shipping information and the payment total. User can CANCEL or FINISH shopping.

If they click FINISH button, a page is displayed mentioning that the order is received.

3. TESTING STRATEGY

Testing strategy of this project consists of the following steps:

- Defining the scope of testing
- Identify Test Type

3.1. Scope of Testing

In scope: The scope of this testing will be limited to front-end testing of the web application under test.

Out of scope: This test will not include any non-functional testing such as performance testing, load testing, security testing, stress testing etc. Furthermore, this test will not include any back-end testing and API testing. scope of this testing will be limited to Front-end testing of the web application under test. Also, the design tests and visual controls are out of the scope for this testing project.

3.2. Testing Type

The testing type of this project is Functional System Testing limited to Front-end testing in order to evaluate if the application works as expected.

3.3. Test Logic

This testing project is designed in such a way that the core test cases that will need to be part of the Regression automated test suite are created and their automation test scripts are developed so that they can be executed whenever required to reduce the effort in executing the test cases manually. During this test project, the tester(s) will adhere to best practices as much as possible when it comes to developing a framework to automate the testing of saucedemo.com especially so that you may want to re-use the framework to test another website or application.

4. TESTING OBJECTIVES

The main goal of this testing is to test SAUCEDEMO.COM web application comprehensively and automate the test cases 100% in order to avoid repetitive manual testing.

5. TESTING CRITERIA

5.1. Entry Criteria

The entry criteria refer to the desirable conditions in order to start test execution. In this testing project, the entry criteria is the ready test cases that includes the tests for the main functions of the web application.

5.2. Exit Criteria

The exit criteria for this testing project is the automate the prepared and manually tested test cases 100% percent.

6. RESOURCE PLANNING

The start date of the project is **Tuesday 22th November, 9:00 am (GMT+3)** and the deadline is **Monday 28th November, 5:00pm (GMT+1).** There will be one tester to automate 6 test cases and 15 test scenarios. There is no technical scarcity or impediment to meet the project deadline.

There is just one risk factor that jeopardize the delivery of the project safe and sound which is that the tester who is going to automate test cases has never used the automation tools and programming language that the project requires.

7. TEST ENVIRONMENT

- Visual Studio 2022
- C# code (using .NET core)
- UUnit framework
- Specflow BDD Framework

8. SCHEDULE

8.1. 11/22/2022

Manually test the https://www.saucedemo.com/ website, create test cases, check https://petstore.swagger.io/#/ website, create a test plan draft for Front-end testing and API testing project.

8.2. 11/23/2022

Install Visual Studio 2022 and study C#, create a NUnit project, install RestSharp from Nuget Packages and study API Testing with RestSharp.

8.3. 11/24/2022

Create automated REST API testing for https://petstore.swagger.io/#/ by using NUnit Framework and RestSharp.

8.4. 11/25/2022

Create a NUnit project, install Selenium, Selenium WebDriver, Specflow to create a framework for front-end testing of https://www.saucedemo.com/.

8.5. 11/26/2022

Continue automating test cases for front-end testing with Specflow BDD Framework.

8.6. 11/27/2022

Continue automating test cases for front-end testing with Specflow BDD Framework.

8.7. 11/28/2022

Finalize test plans, framework and automation test scripts and create a Github repository for the delivery and deliver the project.

9. TEST DELIVERABLES

1. A test plan pf the tests intended to be executed
2. Detailed test cases that are going to be automated
3. Automated test scripts in C# code (using .Net core) to regress the functionality of the website under test.
4. Use of Page Object Model to be able to re-use them accordingly where applicable.
5. A test technique where step definitions (with the use of Specflow) are to be re-used as much as possible for all page objects.
6. A formal test execution report to communicate findings and statuses in an effective way.