|  |
| --- |
| Photo displaying partial image of two pie charts on a canvas-textured page |
| **Test Automation Plan**  Webapp (https://www.saucedemo.com/) |
| |  |  |  | | --- | --- | --- | | Tibor Weigand | 5/25/24 | C# exercise | |

# Table of Contents

[Table of Contents 1](#_Toc167538881)

[**1. Introduction** 2](#_Toc167538882)

[1.1 Purpose and Objectives 2](#_Toc167538883)

[1.2 Scope 2](#_Toc167538884)

[**2. Requirements and Test Cases** 2](#_Toc167538885)

[**3. Test Automation Strategy** 2](#_Toc167538886)

[3.1 Tools and Technologies 2](#_Toc167538887)

[3.2 Framework Design 2](#_Toc167538888)

[**4. Test Case Design** 3](#_Toc167538889)

[4.1 Test Case: Verify Login Screen Display 3](#_Toc167538890)

[4.2 Test Case: Verify Username and Password Textboxes 3](#_Toc167538891)

[4.3 Test Case: Verify Login Button 3](#_Toc167538892)

[4.4 Test Case: Verify Error Message for Invalid Credentials 4](#_Toc167538893)

[4.5 Test Case: Verify Web Shop Test Page Loaded After Login 4](#_Toc167538894)

[**5. Test Execution Plan** 4](#_Toc167538895)

[5.1 Schedule 4](#_Toc167538896)

[5.2 Environment 4](#_Toc167538897)

[**6. Resources** 4](#_Toc167538898)

[**7. Risk Management** 5](#_Toc167538899)

[**8. Test Maintenance** 5](#_Toc167538900)

[**9. Approvals** 5](#_Toc167538901)

## **1. Introduction**

### 1.1 Purpose and Objectives

The purpose of this test automation plan is to outline the strategy, approach, resources, and schedule for the automation of test cases for the web application hosted at [SauceDemo](https://www.saucedemo.com/). The primary objective is to ensure that the login functionality and related requirements are thoroughly tested to verify compliance with the specified requirements.

### 1.2 Scope

This plan covers the automation of test cases related to the login screen and its functionalities, as well as the functionality of the web shop. Specifically, it addresses the requirements as listed below, ensuring that the application behaves as expected when users interact with the login elements.

## **2. Requirements and Test Cases**

|  |  |
| --- | --- |
| Requirement Mark | Requirement |
| 1 | Login screen (“Login”) is displayed when the application is started. |
| 1.1 | Login screen shall have "Username" and "Password" textboxes. They should be enabled and empty. |
| 1.2 | Login screen shall have a "Login" button. It should be enabled. |
| 1.3 | Login screen shall have an "Error message" displayed when invalid credentials are provided. |
| 1.4 | Web shop test page is loaded after login. |

## **3. Test Automation Strategy**

### 3.1 Tools and Technologies

* **WebDriver**: Selenium WebDriver
* **Language**: C#
* **Framework**: NUnit
* **CI Tool**: Jenkins (Optional)
* **Browser**: Chrome

### 3.2 Framework Design

* **Page Object Model (POM)**: Each web page will have a corresponding Page Object class containing web elements and actions.
* **Test Data Management**: Test data will be managed using CSV or JSON files.

## **4. Test Case Design**

### 4.1 Test Case: Verify Login Screen Display

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Requirement Mark | Requirement | Test Case Description | Steps | Expected Result |
| 1 | Login screen (“Login”) is displayed when the application is started. | Verify that the login screen is displayed when the user navigates to the application URL. | 1. Open the application URL. | The login screen should be displayed with the "Username" and "Password" fields and "Login" button. |

### 4.2 Test Case: Verify Username and Password Textboxes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Requirement Mark | Requirement | Test Case Description | Steps | Expected Result |
| 1.1 | Login screen shall have "Username" and "Password" textboxes. They should be enabled and empty. | Verify that "Username" and "Password" textboxes are present, enabled, and empty. | 1. Open the application URL  2. Check the "Username" textbox  3. Check the "Password" textbox. | "Username" and "Password" textboxes should be enabled and empty. |

### 4.3 Test Case: Verify Login Button

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Requirement Mark | Requirement | Test Case Description | Steps | Expected Result |
| 1.2 | Login screen shall have a "Login" button. It should be enabled. | Verify that the "Login" button is present and enabled on the login screen. | 1. Open the application URL  2. Check the "Login" button. | The "Login" button should be present and enabled. |

### 4.4 Test Case: Verify Error Message for Invalid Credentials

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Requirement Mark | Requirement | Test Case Description | Steps | Expected Result |
| 1.3 | Login screen shall have an "Error message" displayed when invalid credentials are provided. | Verify that an error message is displayed when the user attempts to log in with invalid data. | 1. Open the application URL  2. Enter invalid credentials  3. Click the "Login" button. | An error message should be displayed indicating invalid login credentials. |

### 4.5 Test Case: Verify Web Shop Test Page Loaded After Login

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Requirement Mark | Requirement | Test Case Description | Steps | Expected Result |
| 1.4 | Web shop test page is loaded after login. | Verify that the web shop test page is loaded after a successful login. | 1. Open the application URL  2. Enter valid login credentials  3. Click the "Login" button. | The web shop test page should be loaded, displaying the product inventory. |

## **5. Test Execution Plan**

### 5.1 Schedule

* **Test Development**: June 1 - June 5
* **Test Execution**: June 6 - June 7
* **Report Generation**: June 8

### 5.2 Environment

* **Browsers**: Chrome (latest version)
* **Test Environment**: Local machines or CI environment (e.g., Jenkins)

## **6. Resources**

* **Personnel**: 1 Test Automation Engineer
* **Hardware**: Laptops/desktops for development and execution
* **Software**: Selenium WebDriver, NUnit, Chrome browser, Jenkins (optional)

## **7. Risk Management**

* **Risk**: Changes in the web application UI
  + **Mitigation**: Regular updates to the Page Object classes and test cases.
* **Risk**: Network issues causing test failures
  + **Mitigation**: Ensure stable internet connection and retry mechanisms in test scripts.

## **8. Test Maintenance**

* **Regular Review**: Test scripts will be reviewed bi-weekly to ensure they are up to date with application changes.
* **Version Control**: Test scripts will be maintained in a version control system (e.g., Git).

## **9. Approvals**

|  |  |  |  |
| --- | --- | --- | --- |
| Role | Name | Signature | Date |
| Test Lead |  |  |  |
| Project Manager |  |  |  |
| QA Manager |  |  |  |

This test automation plan outlines a structured approach to automate the login functionalities of the SauceDemo web application, ensuring thorough testing and quality assurance.