Test Strategy

Project FORMY

Fernando Garzon

July, 2024

Summary

Introduction

This document outlines the test strategy for the web form application. The objective is to ensure the form functions correctly, captures and processes data accurately, and meets usability standards.

Objectives

The testing process aims to deliver a fully functional web form that fulfills the required standards and exceeds performance expectations. Our primary goals are to identify and resolve defects, ensure data validation, and confirm that the form submits data correctly.

Test Levels

Testing will be conducted at multiple levels:

- **Unit Testing**: Focused on individual form components to ensure they function correctly in isolation.
- **Integration Testing**: Ensures that the form fields work together as expected and the data submission process is seamless.

Test Types

The following test types will be conducted:

- **Functional Testing**: Validates that the form functions according to the specified requirements.
- **Regression Testing**: Ensures that recent changes have not adversely affected existing functionality.
- **Smoke Testing**: Conducted to verify that the basic functionalities of the form are working as expected after each build.
- **Usability Testing**: Ensures the form is user-friendly and accessible.

Test Techniques

We will utilize automated black-box testing techniques. This will allow us to focus on user perspective application behavior without considering the internal code structure.

Test Deliverables

The following deliverables will be produced during the testing process:

- **Test Plans**: Outline the scope, approach, resources, and schedule of testing activities.
- **Test Cases**: Detailed steps to verify specific functionalities of the form.
- **Test Automated Scripts**: Scripts developed for automated testing using the Robot Framework and Selenium library.
- **Test Reports**: Summarize the results of testing activities, including any identified defects and their status.

Test Environment

The testing environment will consist of:

- A computer with an updated operating system, connected to the internet with decent bandwidth.
- Browsers to be tested include Chrome, Firefox, and Edge to guarantee compatibility.
- Automation scripts will be written in Python using the Robot Framework and Selenium library.
- Pabot library will be utilized to run parallel tests.
- Jenkins servers will be used to control the test builds and generate the reports.

Test Schedule

A detailed test schedule will be developed, giving the timeline for each phase of the testing process. The schedule will include timeframes for each test type.

Test Scenarios

Specific test scenarios for the web form include:

1. **Form Field Validation**:
    - Verify that all mandatory fields are marked and validated correctly.
    - Ensure the form does not submit if required fields are empty.
    - Validate input types (e.g., text, radio buttons, checkboxes, dropdowns).
2. **Data Submission**:
    - Ensure the form data is submitted correctly.
    - Verify that the form displays a success message upon submission.
    - Confirm that the form can handle incorrect data inputs gracefully.
3. **Cross-Browser Testing**:
    - Ensure the form functions correctly across different browsers (Chrome, Firefox, Edge).

Risk and Mitigation

Potential risks and their mitigation strategies include:

- **Compatibility Issues**: Testing across multiple browsers (Chrome, Firefox, Edge) to ensure consistent behavior.
- **Data Validation Issues**: Implementing thorough validation checks to ensure data accuracy.
- **Automation Script Failures**: Continuous integration with Jenkins to detect and address failures early.

Conclusion

This test strategy aims to ensure that the web form application is robust, reliable, and ready for deployment. By following this strategy, we aim to identify and resolve defects early, maintain high quality, and meet user expectations.