**Coffee Shop**

**Software Test Plan**

**(STP) Document**

Version 1.0

25.02.2023

Shaden Ghazalin

|  |
| --- |
| **Contents** |

[1. Introduction 3](#_Toc244519333)

[2. Test Data 3](#_Toc244519334)

[3. Testing Scope 3](#_Toc244519335)

[4. Testing setup 4](#_Toc244519336)

[5. Testing Requirements 5](#_Toc244519341)

[6. Testing Strategy 5](#_Toc244519333)

[7. Testing Metholodogy 5](#_Toc244519333)

[8. Risk assessment 5](#_Toc244519333)

1. Introduction

This Software Test Plan (STP) document outlines the testing approach, testing schedule, and testing resources required for the coffee shop web application. The purpose of this document is to define the scope of testing, the test objectives, and the procedures and tools to be used during testing.

2. Test Data

Note: Due to my unfamiliarity with what I should write, I found on Google that I should give an example that can be modified depending on the requirements.

1. User Registration:

* Name: John Doe
* Email: [johndoe@example.com](mailto:johndoe@example.com)
* Password: Test123
* Confirm Password: Test123

1. Menu Navigation:

* Click on "Product" on Menutab
* Select "Latte".
* Select "Grande" size
* Add "Extra Shot" and "Whipped Cream" options

1. Place an Order:

* Select "Latte" from the menu
* Select "Grande" size
* Add "Extra Shot" and "Whipped Cream" options
* Click on "Add to Cart" button
* Click on "Checkout" button
* Select "Pickup" option
* Select "ASAP" as pickup time

1. Payment Processing:

* Enter valid credit card information
* Click on "Place Order" button

1. Order Status:

* Check the status of the order on the "Order History" page
* Verify that the status is "Preparing"
* Check the status of the order again after a few minutes
* Verify that the status has changed to "Ready for Pickup"

3. Testing Scope

1. Functionality Testing:

* Test all the features and functions of the coffee shop web application including user registration, menu navigation, placing orders, payment processing, and order status tracking.
* Verify that all the buttons, links, forms, and fields are working properly and as expected.
* Test the error handling and validation for all inputs.

1. User Interface Testing:

* Test the user interface of the coffee shop web application for consistency, usability, and accessibility.
* Verify that the layout, design, and colors are consistent throughout the application.Test the responsiveness and compatibility of the application on different devices, browsers, and screen sizes.
* Verify that the font size and contrast meet the accessibility standards.

1. Performance Testing:

* Test the performance of the coffee shop web application under different loads and conditions.
* Measure the response time, page load time, and server response time for different.
* scenarios and user actions.
* Test the application for scalability and capacity to handle many concurrent users and orders.

1. Security Testing:

* Test the security of the coffee shop web application against common vulnerabilities and threats such as SQL injection, cross-site scripting, and cross-site request forgery.
* Verify that all user inputs are properly validated and sanitized to prevent malicious attacks.
* Test the password strength and encryption for user accounts and credit card information.
* Test the authentication and authorization mechanisms for different user roles and permissions.

1. Compatibility Testing:

* Test the compatibility of the coffee shop web application with different operating systems, devices, browsers, and versions.
* Verify that the application works properly on different platforms such as Windows, Mac, iOS, and Android.Test the application on different browsers such as Chrome, Firefox, Safari, and Edge.
* Verify that the application supports different versions of HTML, CSS, and JavaScript.Note: The above testing scope is just an example and can be modified based on specific requirements and priorities for the coffee shop web application.

4. Testing Setup

1. Hardware: Computer or laptop, mobile device, and internet connection.
2. Software: Latest browsers, Node.js, NPM, Git, and operating system.
3. Test Environment: Clone the repository, install dependencies, set up database and server, run application in development mode.
4. Test Data: Create test user accounts, menu items, orders, and use mock data.
5. Test Execution and Reporting: Use a test management tool, execute test cases, report results, and share summary and metrics with stakeholders and development team.

5. Testing Requirements

1. Functional Testing: Ensure the coffee shop menu, cart, checkout, and user account features work correctly.
2. Usability Testing: Verify the user interface is easy to use, responsive, accessible, and visually appealing.
3. Performance Testing: Verify the web application can handle high traffic, has acceptable response and loading times, and can handle large amounts of data.
4. Security Testing: Test for vulnerabilities, ensure sensitive data is encrypted, and verify authentication and compliance with security protocols and regulations.
5. Compatibility Testing: Test on different browsers, devices, and screen resolutions, and ensure compatibility with assistive technologies.

6. Testing Strategy

The test strategy for the coffee shop web application is to use a combination of manual and automated testing to ensure that the application meets the functional and non-functional requirements. The testing approach will involve the following:

* Functional testing: Ensuring that the application meets all the functional requirements specified in the SRS document.
* Non-functional testing: Ensuring that the application meets all the non-functional requirements specified in the SRS document.
* Usability testing: Ensuring that the application is easy to use and meets the needs of the target audience.
* Performance testing: Ensuring that the application performs well under different load conditions.
* Security testing: Ensuring that the application is secure and meets industry standards for security.

7. Testing Methodology

1. Plan: Define the testing goals, scope, timelines, and resources.
2. Design: Identify and create test cases based on the requirements and testing types.
3. Execute: Run the test cases and record the results, including any defects found.
4. Report: Analyze the test results, prioritize and track defects, and create reports for stakeholders.
5. Retest: Verify defects are fixed, and retest to ensure the changes did not introduce new issues.

8. Risk Assessment

1. Data Breaches: The web application will handle sensitive customer data such as payment information, so the risk of a data breach must be assessed and mitigated.
2. Server Downtime: The web application must be available for customers to use at all times, so the risk of server downtime must be assessed and a backup and recovery plan put in place.
3. Compatibility Issues: The web application must work correctly on different browsers and devices, so the risk of compatibility issues must be assessed and testing plans put in place to mitigate any issues found.
4. User Experience: The web application must provide a positive user experience, so the risk of usability issues must be assessed and testing plans put in place to ensure a user-friendly interface.
5. Security: The web application must be secure, so the risk of vulnerabilities and attacks must be assessed, and security measures put in place to protect against them.