Test Plan for End-to-End Testing of Xaltsocn Portal Application

Version: 1.0

Author: Abishek Raja N

Date: 1/14/2025

Reviewed By: [Reviewer Name]

1. Introduction

1.1 Purpose

This test plan outlines the approach to validate the Xaltsocn Portal Web Application. The application allows users to interact with a blockchain system, including signing up, signing in, onboarding nodes to an existing blockchain, creating a new private blockchain, and signing out. This document provides the testing strategy for ensuring these functionalities work as expected.

1.2 Scope

This test plan focuses on the following user actions:

Sign Up: Creating a new user account.

Sign In: Logging in with an existing account.

Submit Request to Onboard Nodes to Existing Blockchain: Adding nodes and wallet details for an onboarding request.

Submit Request to Create New Private Blockchain: Adding nodes and wallet details for creating a new private blockchain.

Sign Out: Logging out from the application.

The test plan will also address functional, security, performance, Accessibility and usability testing to ensure the application meets its business and technical requirements.

2. Test Objectives

The primary objectives of this test plan are:

Verify Account Creation: Ensure that users can sign up with a valid email and password.

Verify Login: Ensure users can sign in with valid credentials.

Validate Node Onboarding to Existing Blockchain: Ensure that nodes and wallet details can be added correctly, and the user can proceed to the next steps.

Validate Private Blockchain Creation: Ensure users can successfully create new private blockchains by adding nodes and wallet details.

Ensure Proper Sign Out: Validate that users can sign out properly.

Test Edge Cases: Ensure that invalid inputs and incorrect formats are handled appropriately.

Security Testing: Test for common security vulnerabilities.

Performance Testing: Ensure the application performs well under normal and high loads.

Accessibility Testing: Ensure the application is accessible to users with disabilities, adhering WCAG 2.1 guidelines.

Cross-Browser Testing: Ensure the application works across multiple browsers (e.g., Chrome, Firefox, Safari).

3. Test Types

3.1 Functional Testing

Functional testing focuses on verifying that the application's features function according to the user requirements.

Objective: Ensure all user actions (sign up, sign in, node onboarding, private blockchain creation, sign out) work as intended.

3.2 Security Testing

Security testing ensures that the application is secure and can handle sensitive data properly.

Objective: Verify that no security vulnerabilities exist, such as SQL injection, cross-site scripting (XSS), or session hijacking.

3.3 Performance Testing

Performance testing evaluates how the application performs under expected load conditions.

Objective: Ensure the application can handle concurrent users and large amounts of data without performance degradation.

3.4 Usability Testing

Usability testing evaluates the user interface and overall user experience.

Objective: Ensure the application is intuitive, and the user flows are easy to navigate.

3.5 Compatibility Testing

Compatibility testing ensures that the application works on various devices, operating systems, and browsers.

Objective: Verify that the application functions across major browsers (Chrome, Firefox, Safari, Edge) and devices (desktop, mobile).

3.6 Accessibility Testing

Accessibility testing ensures that the application is usable by people with disabilities, adhering to WCAG (Web Content Accessibility Guidelines).

Objective: Ensure that users with visual, auditory, cognitive or other disabilities can navigate and interact with the application effectively.

4. Test Scenarios / Test Cases

4.1 Functional Test Cases

Test Case 1: Successful Sign Up

Description: Ensure the user can sign up with a valid email and password.

Pre-condition: User is on the sign-up page.

Test Steps:

Enter a valid email address (e.g., user@example.com).

Enter a valid password (e.g., ValidPassword123).

Click the "Sign Up" button.

Expected Result: User is successfully signed up and redirected to the login page.

Pass/Fail Criteria: Pass if the user is redirected to the login page; fail if the sign-up fails or no redirection occurs.

Test Case 2: Successful Sign In

Description: Ensure the user can log in with valid credentials.

Pre-condition: User has an existing account.

Test Steps:

Navigate to the Sign In page.

Enter a valid email and password.

Click the "Sign In" button.

Expected Result: User is redirected to the homepage or dashboard.

Pass/Fail Criteria: Pass if the user is successfully logged in; fail if login fails.

Test Case 3: Submit Request to Onboard Nodes to Existing Blockchain

Description: Ensure the user can add nodes and wallets and submit the onboarding request to an existing blockchain.

Pre-condition: User is logged in and on the "Onboard Nodes" page.

Test Steps:

Enter valid node details (e.g., NodeID-1, 192.168.1.1).

Click "ADD NODE" to add the node to the list.

Repeat the above steps for additional nodes.

Enter wallet address (e.g., 0x88fa61d2faA13aad8Fbd5B030372B4A159BbbDFb) and select a permission type.

Click "ADD WALLET" to add the wallet to the list.

Repeat the above steps for additional wallets.

Click "NEXT" to proceed.

Review details and click "SUBMIT".

Expected Result: User is able to successfully onboard nodes and wallets to the existing blockchain.

Pass/Fail Criteria: Pass if the request is successfully submitted; fail if there are any errors during node or wallet addition.

Test Case 4: Submit Request to Create New Private Blockchain

Description: Ensure the user can submit a request to create a new private blockchain.

Pre-condition: User is logged in and on the "Create Private Blockchain" page.

Test Steps:

Enter a valid network name.

Enter a valid wallet address (e.g., 0x88fa61d2faA13aad8Fbd5B030372B4A159BbbDFb).

Click "NEXT".

Add nodes by entering NodeID-{number} and a valid public IP (e.g., 192.168.1.1).

Click "ADD NODE" to add each node to the list.

Click "NEXT" after adding all nodes.

Review details and click "SUBMIT".

Expected Result: User can create a new private blockchain successfully.

Pass/Fail Criteria: Pass if the request is submitted successfully; fail if there are issues in adding nodes or submitting the request.

Test Case 5: Sign Out

Description: Ensure the user can sign out successfully.

Pre-condition: User is logged in.

Test Steps:

Click the "Sign Out" button.

Expected Result: User is logged out and redirected to the Sign In page.

Pass/Fail Criteria: Pass if the user is signed out successfully and redirected to the Sign In page; fail if the user remains logged in.

4.2 Security Test Cases

Test Case 6: SQL Injection in Sign In

Description: Ensure that SQL injection attempts are blocked in the Sign In form.

Pre-condition: User is on the Sign In page.

Test Steps:

Enter a malicious input (e.g. admin' OR 1=1 --) in the email field.

Enter any password.

Click "Sign In".

Expected Result: The system should reject the input and display an error message.

Pass/Fail Criteria: Pass if the input is rejected and the user is not logged in; fail if login succeeds.

4.3 Performance Test Cases

Test Case 7: Load Testing for Node Onboarding

Description: Ensure the system can handle multiple requests to onboard nodes.

Pre-condition: JMeter software.

Test Steps:

Simulate 1000 users concurrently adding nodes to the onboarding list.

Expected Result: The system should handle the load without significant delays or errors.

Pass/Fail Criteria: Pass if the application remains responsive; fail if response times exceed acceptable limits.

4.4 Accessibility Test Cases

Test Case 8: Screen Reader Compatibility

Description: Ensure the application is compatible with screen readers for visually impaired users.

Pre-condition: Screen reader software is enabled (JAWS / NVDA)

Test Steps:

1. Navigate to various pages of the application (Sign up, Sign in, Node Onboarding).
2. Use the screen reader to navigate and listen to the descriptions and labels.

Expected Result: All important elements (buttons, form fields, images) should be correctly announced by the screen reader.

Pass/Fail Criteria: Pass if the screen reader announces the elements correctly; fail if elements are not announced or incorrectly announced.

Test Case 9: Keyboard Navigation

Description: Ensure users can navigate the application using only the keyboard.

Pre-condition: Application is accessible via a keyboard.

Test Steps:

1. Navigate through the application only the keyboard (Tab, Enter Space and arrow keys).
2. Ensure that all interactive elements are accessible using the 'Tab' key.
3. Verify that focus indicators are visible and clearly guide the user through the interface.

Expected Result: All interactive elements should be accessible via keyboard, and focus indicators should be visible.

Pass/Fail Criteria: Pass if all the elements are accessible and keyboard navigation works seamlessly; fail if any elements are inaccessible or focus is not indicated.

Test Case 10: Color Contrast and Text Readability

Description: Ensure that the application follows proper color contrast guidelines for readability.

Pre-condition: Color Contrast Analyser software

Test Steps:

Review key areas of the user interface to check for color contrast.

Expected Result: The application should meet the WCAG contrast ratio requirements for readability.

Pass/Fail Criteria: Pass if all text meets contrast guidelines; fail if the contrast is insufficient for users with visual impairments.

4.5 Usability Test Cases

Test Case 11: Usability Testing

Description: Ensure the application is compatible with screen readers for visually impaired users.

Pre-condition:

1. The user is not logged in (for testing the sign-up and sign-in flow).
2. User has at least one node to add and is able to submit requests (for onboarding and blockchain creation).

Test Steps:

1. Evaluate if users can easily understand and complete key tasks within the application.
2. Check that form fields, buttons, and actions are clearly labelled and intuitive.
3. Observe how users interact with the interface and if they encounter any difficulty or confusion.

Expected Result:

1. The application provides a smooth, intuitive experience, with clear instructions and error messages when necessary.
2. Users should be able to complete the tasks without feeling lost or needing external assistance.

Pass/Fail Criteria: Pass if users can navigate the application effortlessly; fail if users encounter confusion, cannot complete a task or receive unclear error messages.

5. Test Execution Strategy

5.1 Manual Testing

Manual testing will be executed to verify Accessibility, usability and all user actions such as sign up, sign in, node onboarding, and private blockchain creation.

Exploratory testing will cover edge cases, UI flows, and validation of various input formats (node IDs, IP addresses, wallet addresses).

5.2 Automated Testing

Automated tests will be created for functional cases, particularly for repetitive tasks like node and wallet addition.

Tools like Selenium will be used for UI automation and Rest Assured for API Automation.

JMeter will be used for performance testing to simulate concurrent users.

6. Test Deliverables

Test case documentation, including manual and automated tests.

Test execution results and logs.

Defect reports for any identified issues.

Test summary report for stakeholders.

7. Risk and Mitigation Plan

Risk: Test environment downtime.

Mitigation: Coordinate with the DevOps team to ensure uptime for the test environment.

Risk: Slow response times during peak load.

Mitigation: Use performance optimization techniques, monitor system performance regularly.

8. Test Exit Criteria

All critical test cases have passed.

No critical or high-severity defects are outstanding.

Test results have been documented, and a summary report has been submitted.

9. Conclusion

This test plan ensures comprehensive testing of the Xaltsocn Portal Application, covering all critical functionalities such as user registration, login, node onboarding, private blockchain creation, and logout. The testing strategy includes both manual and automated approaches and ensures all business and security requirements are met.