**End-to-End Test Plan**

**Project :** Open Capital Network

**Documented by :** Sofia Sahoo

**Date :** 28/02/2025

1.Introduction

1.1 Purpose

The purpose of this test plan is to define the strategy, scope, approach, and resources required to validate the Open Capital Network (OCN) Portal. The plan covers end-to-end (E2E) testing, including functional, non-functional, and integration aspects.

1.2 Objectives

* Validate all critical end-to-end user journeys from login to transaction processing.
* Confirm role-based access, security, and compliance with industry standards.

2.Scope

2.1 In Scope-

* User authentication (login/logout)
* Navigation and UI verification
* Dashboard data validation
* Node Details Form Validation(Mandatory fields: Node ID and Public IP)
* Submitting requests for onboarding nodes to an existing blockchain.
* Submitting requests for creating a new private blockchain.
* Adding and submitting node details, wallet addresses, and related permissions
* Performance, security, and usability testing
* Cross-browser and cross-device testing

2.2 Out-of-Scope-

* Backend blockchain mechanics.
* Third-party service integrations.

3.Test Deliverables

* **Test Plan**: This document.
* **Test Cases**: Detailed test cases and scripts.
* **Test Reports**: Results from test executions.
* **Defect Logs**: List of identified defects, their severity, and resolutions.
* **Test Summary Report**: Final report summarizing the testing process, results, and suggestions for improvements.

4.Test Strategy

4.1 Test Levels

The testing will be performed at different levels to ensure the application works correctly across all components:

1. Unit Testing:

* **Scope**: Testing individual units (e.g., functions or methods) responsible for specific logic, such as adding nodes or wallets, user registration, and network creation
* **Objective**: Ensure that each function behaves as expected in isolation.

1. Integration Testing:

* **Scope**: Testing the interaction between different components, such as the integration of the frontend and backend, API calls, database interactions, and the ability to add nodes to the blockchain and wallets to the network.

**Objective**: Verify that the system’s components work together correctly (e.g., node and wallet details are stored in the database after being added via the UI).

1. System Testing:

* **Scope**: End-to-end testing of the entire application. This includes testing the workflows of creating an account, logging in, adding nodes and wallets, and submitting requests to onboard nodes or create a new blockchain.
* **Objective**: Ensure that the full application works as intended, from account creation to node addition and network creation.

1. User Acceptance Testing (UAT):

* **Scope**: Validation of the system’s functionality from the user’s perspective. In this case, it’s essential that the user can smoothly create an account, onboard nodes, and create a new blockchain.
* **Objective**: Validate the application against user requirements and ensure it meets business needs (e.g., ease of adding nodes, wallet management, etc.)

4.2 Testing Types

Different types of testing will be employed to verify the functional and non-functional aspects of the system.

1. Functional Testing:

* User Registration (Sign Up)
* User Login (Sign In)
* Node Onboarding (Add Node to Blockchain)
* Wallet Onboarding (Add Wallet to Blockchain)
* Create Private Blockchain
* Data Validation

1. Regression Testing:

* Ensure that any new changes (e.g., adding a feature to handle new types of wallets or nodes) do not break existing functionality.
* Verify that the previously tested flows still function correctly after modifications.

1. Performance Testing:

Load Testing: Simulate a large number of users performing sign-up, login, node addition, and blockchain creation at the same time to evaluate how the system handles multiple concurrent requests.

Check system performance under stress (e.g., slow response when adding many nodes or wallets).

Stress Testing:Test the system’s behavior under extreme conditions (e.g., large data input, hundreds of nodes and wallets added simultaneously).

1. Security Testing:

Authentication:

Test login and session management to ensure that user credentials are handled securely.

Verify that password hashing and encryption are implemented properly.

Authorization:

Verify that users can only access their own data and are restricted from unauthorized actions (e.g., accessing someone else’s blockchain).

1. Usability Testing:

Ensure that the user interface is intuitive for users, especially in the areas of:

Adding nodes and wallets.

Navigating through the onboarding and blockchain creation process

1. Compatibility Testing: Ensure that the application works across different browsers (Chrome, Firefox, Safari, Edge).

4.Test Approach

4.1 Test Case Development

* Test cases will be created based on the functional specifications, user stories, and requirements.
* Test cases will be organized by user actions such as "Sign Up", "Sign In", "Add Node", “Request To Onboard Nodes to Existing Blockchain’’ and "Create Blockchain".
* Request To Onboard Nodes to Existing Blockchain.

4.2 Test Data Preparation

* A set of valid and invalid email addresses, passwords, node IDs, public IPs, wallet addresses, and permissions will be prepared for testing.
* Dummy data for blockchain (e.g., node IDs, wallet permissions) will be used in the test cases

5. Test Schedule

| Deliverable | Date | Responsibility |
| --- | --- | --- |
| Test Plan Approval | Insert Date | Test Manager |
| Test Case Design Completion | Insert Date | QA team |
| PO review/Approval | Insert Date | Product Owner/Lead |
| Test Execution | Insert Date | QA team |
| Defect Reporting & Resolution | Insert Date | QA team/Dev Team |
| Test Summary Report | Insert Date | QA Lead |

6.Test Resources

6.1 Human Resources

* **Test Manager**: [Name]
* **QA Lead**: [Name]
* **Test Engineers**: [Names]
* **Developers**: [Names]

6.2 Tools

* **Test Management Tool**: [e.g., JIRA,ADO]
* **Automation Tools**: [e.g., Cypress]
* **API Testing Tools**: Postman
* **Bug Tracking Tool**: JIRA, ADO

7.Risks and Mitigation

| Risk | Mitigation |
| --- | --- |
| Incorrect login credentials | Implement input validation with clear error messages. |
| Incorrect data entry (node ID, wallet address) | Implement input validation with clear error messages. |
| Limited availability of test environment | Coordinate with IT/DevOps to ensure availability |
| Time constraints for test execution | Prioritize test cases based on critical functionality |
| Complex UI flows for multi-step actions | Perform thorough manual testing and walkthroughs all basic functionalities |

8. Test Environment

A new server is required for the web server, the application and the database.

9. Entry and Exist criteria

9.1 Entry Criteria

* Test environment is set up and accessible.
* Test cases are reviewed and approved.
* All necessary tools and access are available.

9.2 Exist Criteria

* All planned tests have been executed.
* Critical and major defects have been resolved.
* Test summary report has been reviewed and signed off.

10. Testcases:

10.1 Positive Scenarios

| **Testcase** | **Precondition** | **Steps** | **Expected Results** |
| --- | --- | --- | --- |
| | Verify successful user  sign-up with valid credentials. | | --- |  |  | | --- | | | User has not created an account yet. | | --- |  |  | | --- | | 1. Navigate to URL.  2. Click "Sign In" or “Get Started” button  3. Enter valid email, password and confirm password. | User successfully  signs up and is redirected to the Sign-In page |
| Verify successful login with  Valid credentials | User created an account | 1. Navigate to URL.  2. Click "Sign In" or “Get Started” button  3.Click on “already have an account” link  4. Enter valid email, password. | User successfully gets logged in and navigate to dashboard. |
| Verify user able to Submit Request to Onboard Nodes to Existing Blockchain and Adds Multiple Nodes   |  | | --- | | User successfully log in | 1.Click on Onboard ONC Node 2.Enter valid node ID (e.g., NodeID123).  3. Enter a valid public IP (e.g., 192.168.1.1).  4. Click on "ADD NODE".  5. Repeat the process for multiple nodes (e.g., Node124, Node125).  6. Click "NEXT" to proceed to the next step. | Nodes are successfully added to the list and the user proceeds to the next step. |
| Verify User Successfully Adds Multiple Wallets | 1.User successfully log in  2.Added nodes | 1.Check nodes 2.click on next  2. Enter wallet address (e.g., "0xabc123...").  2. Select permission type (e.g., "Admin").  3. Click "ADD WALLET".  4. Repeat for multiple wallets (e.g., "0xdef456...", "0xghi789...").  5. Click "NEXT" to proceed to the next step | wallets are successfully added to the list and the user proceeds to the next step. |
| Verify the user can successfully submit the request. | | User has added nodes and wallets. | | --- |  |  | | --- | | 1. Add all node and wallet details.  2. Click "SUBMIT". | User able to submit successfully |
| Verify User Successfully Creates a New Private Blockchain | | User is logged in | | --- |  |  | | --- | | 1.Click on Launch ONC network  2. Enter a valid network name (e.g., "MyPrivateNetwork").  3.Enter a valid wallet address (e.g., "0xabc123...").  4.click on next  5.Enter valid node details and click "ADD NODE".  6.Click "NEXT" after adding nodes.  7. Click "SUBMIT" to create the blockchain. | User successfully creates a new private blockchain. |

Negative Scenarios:

| Testcase | Precondition | Steps | Expected Results |
| --- | --- | --- | --- |
| Verify User Enters Invalid Email Format | Navigate to url | 1.click on sign in  2. Enter an invalid email ID (e.g., userexample.com).  3. Enter a valid password.  4. Click on "Sign Up". | System displays an error message about the invalid email format (e.g., "Please enter a valid email address"). |
| Verify User Enters a Weak Password | Navigate to url | 1.click on sign in  2. Enter an valid email ID (e.g., userexample.com).  3. Enter a weak password.  4. Click on "Sign Up" | System displays an error message indicating the password doesn't meet the required strength (e.g., "Password must contain at least 8 characters, including one uppercase letter, one number, and one special character"). |
| Verify User Sign-In (Log in with an Existing Account) | Navigate to URL | 1.click on sign in  2. Enter an valid email ID (e.g., userexample.com).  3. Enter a valid password.  4. Click on "Sign Up". | System displays an error message |
| Verify User sign in with Email and Password Blank | Navigate to URL | 1.click on sign in  2. Enter an blankemail ID (e.g., userexample.com).  3. Enter a blank password.  4. Click on "Sign Up". | System displays an error message |
| Verify User Enters Invalid Public IP Address and Request to Onboard Nodes to Existing Blockchain | | User is logged in and on the node onboarding page. | | --- |  |  | | --- | | 1.Click on Onboard ONC Node 2.Enter invalid node ID (e.g., NodeID123).  3. Enter a valid public IP (e.g., 192.168.1.1).  4. Click on "ADD NODE".  5. Repeat the process for multiple nodes (e.g., Node124, Node125).  6. Click "NEXT" to proceed to the next step. | System displays an error message about the invalid IP format (e.g., "Invalid IP address format"). |
| Verify User Adds a Duplicate Node | | User is logged in and on the node onboarding page. | | --- |  |  | | --- | | 1.Click on Onboard ONC Node 2.Enter duplicate node ID (e.g., NodeID123).  3. Enter a valid public IP (e.g., 192.168.1.1).  4. Click on "ADD NODE".  5. Repeat the process for multiple nodes (e.g., Node124, Node125).  6. Click "NEXT" to proceed to the next step | System displays an error message about the invalid IP format (e.g., "Invalid IP address format"). |
| Verify User Enters Invalid Wallet Address | | User is on the wallet onboarding page. | | --- |  |  | | --- | | 1.Check nodes 2.click on next  2. Enter invalid wallet address.  2. Select permission type (e.g., "Admin").  3. Click "ADD WALLET".  4. Repeat for multiple wallets (e.g., "0xdef456...", "0xghi789...").  5. Click "NEXT" to proceed to the next step | System displays an error message |
| Verify User Enters Invalid Network Name | User logged in. | 1.Click on Launch ONC network  2. Enter an invalid network name (e.g., "MyPrivateNetwork").  3.Enter a valid wallet address (e.g., "0xabc123...").  4.click on next  5.Enter valid node details and click "ADD NODE".  6.Click "NEXT" after adding nodes.  7. Click "SUBMIT" to create the blockchain. | System displays an error message |
| Verify User Enters Invalid Wallet Address | User logged in | 1. Enter a valid network name.  2. Enter an invalid wallet address (e.g., "xyz123").  3. Click "NEXT". | System displays an error message |