**Test Plan for CRUD Blockchain Application**

### **Overview**

This document outlines the end-to-end test plan for the CRUD Blockchain Application. The plan includes the testing scope, objectives, approach, test scenarios, and test cases for various functionalities such as sign-up, sign-in, node onboarding, private blockchain creation, and sign-out.

### **Scope**

The application provides users the ability to:

1. Create a new account (Sign Up).
2. Log in to an existing account (Sign In).
3. Submit requests to onboard nodes to an existing blockchain.
4. Submit requests to create new private blockchains.
5. Log out of the application (Sign Out).

Testing will cover both **functional** and **non-functional** aspects, including:

* Functional testing
* Integration testing
* Boundary testing
* Security testing
* Performance testing

### **Testing Types and Objectives**

#### **1. Functional Testing**

Objective: Verify that all functionalities of the application work as intended.

#### **2. Boundary Testing**

Objective: Test with valid and invalid inputs to ensure the application handles edge cases correctly.

#### **3. Integration Testing**

Objective: Test interactions between modules (e.g., sign-up module and database integration).

#### **4. Security Testing**

Objective: Ensure user data is protected and that the application is secure against vulnerabilities like SQL injection, XSS, etc.

#### **5. Performance Testing**

Objective: Evaluate the application’s responsiveness and stability under load.

### **Test Scenarios and Cases**

#### **1. Sign-Up**

| **Test Case ID** | **Test Scenario** | **Steps** | **Expected Result** |
| --- | --- | --- | --- |
| SU\_01 | Valid Sign-Up | Enter valid email and password, click Sign-Up | User account is created, redirected to the dashboard |
| SU\_02 | Invalid Email Format | Enter invalid email (e.g., "invalidemail"), valid password, click Sign-Up | Error message: "Invalid email address" |
| SU\_03 | Password Too Short | Enter valid email, password with fewer than 8 characters, click Sign-Up | Error message: "Password must be at least 8 characters" |
| SU\_04 | Email Already Registered | Enter an already-registered email, valid password, click Sign-Up | Error message: "Email already exists" |

#### **2. Sign-In**

| **Test Case ID** | **Test Scenario** | **Steps** | **Expected Result** |
| --- | --- | --- | --- |
| SI\_01 | Valid Sign-In | Enter valid email and password, click Sign-In | User is redirected to the dashboard |
| SI\_02 | Invalid Password | Enter valid email, incorrect password, click Sign-In | Error message: "Invalid credentials" |
| SI\_03 | Non-Registered Email | Enter non-registered email, valid password, click Sign-In | Error message: "User not found" |
| SI\_04 | Empty Email or Password | Leave email or password field blank, click Sign-In | Error message: "Fields cannot be empty" |

#### **3. Node Onboarding Request**

| **Test Case ID** | **Test Scenario** | **Steps** | **Expected Result** |
| --- | --- | --- | --- |
| NR\_01 | Add Valid Node | Enter valid Node ID and public IP, click Add Node | Node is added to the list |
| NR\_02 | Invalid Node ID Format | Enter invalid Node ID (e.g., "123"), valid public IP, click Add Node | Error message: "Invalid Node ID format" |
| NR\_03 | Invalid Public IP Format | Enter valid Node ID, invalid public IP (e.g., "256.256.256.256"), click Add Node | Error message: "Invalid IP address format" |
| NR\_04 | Add Multiple Nodes | Add multiple valid nodes, click Next | All nodes are displayed in the review step |

#### **4. Private Blockchain Creation**

| **Test Case ID** | **Test Scenario** | **Steps** | **Expected Result** |
| --- | --- | --- | --- |
| PC\_01 | Create Blockchain with Valid Data | Enter valid network name, wallet address, and nodes; click Submit | Blockchain is created, success message displayed |
| PC\_02 | Invalid Wallet Address Format | Enter invalid wallet address, valid network name and nodes, click Submit | Error message: "Invalid wallet address" |
| PC\_03 | Empty Fields | Leave any field empty, click Submit | Error message: "Fields cannot be empty" |

#### **5. Sign-Out**

| **Test Case ID** | **Test Scenario** | **Steps** | **Expected Result** |
| --- | --- | --- | --- |
| SO\_01 | Valid Sign-Out | Click Sign-Out button | User is redirected to the Sign-In page |

### **Test Data**

#### **1. Sign-Up and Sign-In Data**

| Email | Password |
| --- | --- |
| testuser@example.com | Test@1234 |
| invalidemail | Test@1234 |
| user@example.com | short |

#### **2. Node Data**

| Node ID | Public IP |
| --- | --- |
| NodeID-1 | 192.168.1.1 |
| NodeID-ABC | 256.256.256.256 |

#### **3. Wallet Data**

| Wallet Address |
| --- |
| 0x88fa61d2faA13aad8Fbd5B030372B4A159BbbDFb |
| InvalidWalletAddress  0x88fa61d2faA13aad8Fbd5B030372B4A159BbbDJk |

### **Non-Functional Testing**

#### **1. Security Testing**

* Ensure passwords are hashed in the database.
* Verify input fields are protected from SQL injection.
* Ensure that session tokens are securely stored and expire after logout.

#### **2. Performance Testing**

* Test with 100+ concurrent users signing in.
* Measure response times for node onboarding requests.

#### **3. Usability Testing**

* Verify intuitive navigation for all user actions.
* Ensure error messages are clear and actionable.

### **Execution Strategy**

#### **Tools:**

* Selenium WebDriver (Test Automation)
* TestNG (Test Management)
* Apache JMeter (Performance Testing)
* OWASP ZAP (Security Testing)

#### **Environment:**

* Browser: Chrome, Firefox
* OS: Windows 10, macOS
* Test Data: As outlined in the "Test Data" section

### **Reporting**

* Generate test execution reports using ExtentReports.
* Log issues in a bug tracking tool (e.g., JIRA).