**Test Plan: Blockchain Onboarding Application**

**1. Overview**

This document outlines an exhaustive test plan for the end-to-end testing of a Blockchain Onboarding Application. It covers manual and automated testing approaches, including various test types and detailed test cases for success and failure scenarios.

**2. Testing Types and Objectives**

**2.1 Functional Testing**

Ensures that the system meets the functional requirements as per the specification.

* **Sign Up:** Verify account creation using a valid email and password.
* **Sign In:** Validate authentication with correct and incorrect credentials.
* **Submit Request to Onboard Nodes:** Check node addition and wallet association.
* **Submit Request to Create Private Blockchain:** Validate new blockchain creation.
* **Sign Out:** Ensure session termination upon logout.

**2.2 UI Testing**

Validate that UI elements are correctly displayed and function as intended.

**2.3 Boundary Value Testing**

Check edge cases for input fields (e.g., minimum and maximum lengths for email/password).

**2.4 Security Testing**

Ensure proper encryption and protection of user credentials and blockchain transactions.

**2.5 Performance Testing**

Evaluate application speed, response times, and system stability under heavy load.

**2.6 API Testing**

Validate API endpoints for authentication, node onboarding, and blockchain creation.

**2.7 Negative Testing**

Verify system behavior when users enter invalid or malicious data.

**3. Test Cases**

**3.1 Sign Up**

**Success Scenarios:**

1. Create an account with a valid email and password.
2. Verify email confirmation after sign-up.
3. Ensure session is created upon successful registration.

**Failure Scenarios:**

1. Attempt to register with an already existing email.
2. Use an invalid email format.
3. Use a weak password (e.g., less than 8 characters).
4. Submit empty fields.

**3.2 Sign In**

**Success Scenarios:**

1. Sign in with correct credentials.
2. Remember me functionality works correctly.

**Failure Scenarios:**

1. Sign in with incorrect credentials.
2. Attempt to sign in with an unregistered email.
3. Login while account is locked or inactive.

**3.3 Submit Request to Onboard Nodes**

**Success Scenarios:**

1. Add a node with a valid Node ID and public IP.
2. Add multiple nodes and wallets successfully.

**Failure Scenarios:**

1. Add a node with an invalid Node ID format.
2. Add a node with an invalid IP address.

**3.4 Submit Request to Create Private Blockchain**

**Success Scenarios:**

1. Create a private blockchain with a valid network name and wallet address.

**Failure Scenarios:**

1. Enter an invalid wallet address format.
2. Submit request with an empty network name.

**3.5 Sign Out**

**Success Scenarios:**

1. Log out successfully and invalidate session.

**Failure Scenarios:**

1. Attempt to log out when not signed in.

## ****4. Automated Test Cases (Sign Up, Sign In, Sign Out)****

### ****4.1 Framework & Tools****

* **Selenium with Java** for UI automation
* **TestNG** for test execution
* **Maven** for dependency management

## ****5. Conclusion****

This test plan ensures the Blockchain Onboarding Application is fully tested across multiple scenarios, covering functional, security, UI, and performance aspects. Automated tests enhance efficiency for frequent testing cycles.