**Test Plan**

**Introduction**:- Test Plan is a document which is consist of all the future testing related activity. Usually Test manager or Test lead should prepare the test plan.

**Objective**:- What is the aim of writing the test plan. Here we are following the procedure and process in order to the good quality of software to give the customer. Objective of this project to onboard the user for the creation of node with the existing blockchain technologies.

**Scope**: In the Scope what are the feature we are going to test and what are the feature we are not going to test. Mostly I will test only critical and major feature due the limited availability of time after that I will test minor feature. So here I am going to test following modules:-

1. Sign Up
2. Sign In
3. Onboard node to Existing Blockchain
4. Create New Private Blockchain
5. Sign Out

**Testing Methodologies**:- Depending upon the application whatever type of testing I am going to test if it is a web application then i will start (1) Smoke Testing (2) Component testing (3) Integration testing (4) System testing(End-to-End) (5) Performance testing- Load testing, stress testing (6) Usability Testing (7) Exploratory Testing (8) Acceptance Testing (9) Regression Testing (10) Compatibility Testing Ex: Standalone application we don’t require performance testing for ex: MS Paint, MS Excel So here we require only these type of testing which I mentioned above.

1. **Smoke testing:-** In the Smoke testing I will check first weather the build is testable or not and it should be proper installed. While doing smoke testing I will consider positive scenario. I will be focused only blocker & critical issues.
2. **Component Testing:**- I will check all of the component is there like email id, password text box, sign in CTA, sign out CTA, Node ID and IP Address input box and all other tick box and radio button.
3. **Integration Testing:**- I will test the data flow among all the modules.

Basically from sign in to the home page module and also adding the nodes and wallet will be shown on the another tab.

(i)**Positive Integration testing**:- Testing the application with the valid data.

Ex:- for valid email id and password user should able to sign up and login.

(ii)**Negative Integration Testing**:- Testing the application with the invalid data.

Ex:- for invalid email id and password user should not able to login.

Here we will test the application thoroughly with the valid and invalid data.

1. **System Testing(End-to-End)**:- Navigate all the feature sign up, sign in, add nodes , add wallet and check whether end feature sign out is working or not.
2. **Performance Testing**:- In Performance testing I will check the stability and response time of web application by applying load and will check load testing by applying load less or equal to the desired number of users.
3. **Regression Testing**:- I Will test the new changes, bug fixes and impacted areas and also non-impacted area of the feature.
4. **Exploratory Testing**:- first I will explore the whole application and I will identify the all possible scenarios, based on the identification then document all the possible scenarios and execute it this testing will be done when we don’t have requirement. This testing will also help in identify hidden area.
5. **Acceptance Testing**:- This testing is used to check whether the software is capable of handling all the real time business scenarios. By using this testing I can find out blocker and critical defects in the software. By doing this testing we can check the delivered software is working as per the requirements.
6. **Usability Testing**:- Testing the user friendliness of the application is called as usability testing. By this testing we can check the navigational bar and location of the component.

**(10)Compatibility Testing**:- In compatibility testing I can check the application is able to work in different hardware and software platform so that user can run the application on different O.S. like android, IOS, windows.

**Approach**:- Basically approach is varying company to company. Some of the companies is expecting only scenarios or some of the companies is expecting only test cases. In the approach I will take care of all the high level and low scenario so that test coverage should be 100 percentage. In some cases there may be required flow chart to test the application.

Sign Up Sign In Add Nodes Add Wallet Sign Out

**Assumptions**:- While writing the Test plan certain assumptions will be made like tester team will get proper support from the Developer team. We will assume only positive thing like none of the engineer will leave the job till the completion of the project.

**Risk**:- We go for risk only when the assumptions fail. If the assumptions fail, risks are involved. Here we discuss about negative things.

**Backup plan/Mitigation plan:-** We should make the backup plan to made the overcome of the mistake. Risk & mitigation plan is not same for all companies. It might vary from one company to another company. Thus one engineer quits the job then secondary owner will be replaced to that position so that the backup will be there.

**Effort Estimation:-** This section covers how many test engineers are need to complete the task. I can find out the effort estimation of this project so If only one QA engineer is working on this project it may take 3 days to complete and if two engineers are working then it may take 2 days to complete the project. We can find out the budget & cost of the testing.

**Defect Tracking:-** In this section we mention how to communicate the defects found during testing to the development team and also how development team should respond to it. We should also mention the priority of the defect:-

1. High

2. Medium

3. Low

We will also mention the name of the tool. We will use defect tracking tool JIRA and Bugzilla to track the defect.

1. **Severity:-** There are different types of severity. It can be decided based on the impact of the defect on the business. There are 4 different types of severity.
2. **Blocker-** sign up, sign in, add nodes, add wallet, sign out, next CTA
3. **Critical-** node and wallet is adding with invalid input
4. **Major –** User is not able to add node more than 2 or 3
5. **Minor –** Back CTA and Pagination is not working
6. **Priority:-** It is importance given to fix the bug or priority decides how soon bug should be fixed. So depending on the priority of the defects we classify it as P1,P2,P3.
7. **High(P1):-** This will be included in the current feature that we are releasing right now. This is the blocker so this should be fixed earlier like sign up, sign in
8. **Medium(P2):-** This is not a blocker bug but it should fix after the blocker issue like add nodes and wallet.
9. **Low(P3):-** This is low level of bug so this should be fixed in the last when all the blocker and medium bug fixed like Back CTA and Pagination.

**Scheduling:-** This section contains when exactly each activity should start and end exact date should be mentioned like

System study - 00/00/0000 to 00/00/0000 PRD and design document and feature scope

Write test cases – date 00/00/0000 2 days for all the feature in this project

Execute test cases – date 00/00/0000 3 days for one QA engineer

Release date – 00/00/0000 one day + and –

**Test Environment:-** This is the stage where we decide how our test environment should be similar to production environment. Here we discuss about various software and hardware component.

Hardware:- processor like intel 2Ghz

Server :- which company server we are using for our web application.

Software :- Types of Operating system and database server we are using.

**Entry & Exit Criteria:-** These are setup condition that should be made. For the entry criteria following things should be made:-

Test cases should be written and reviewed.

First smoke testing should be completed then start with component testing.

For Exit Criteria – 0 blocker bugs (sign up and sign in, sign out should be happen). We need to pass the integration testing then only we can go for system testing.

0 critical bugs ( node should be added )

Minimum no. of major and minor bugs that can we target in next release.

**Test Stop Criteria:-** We stop testing if the quality of the product is very good or very bad. Quality of the product is very good means all the feature requested like sign up, sign in , add node, add wallet, sign out is working as expected.

When there are no blockers and critical bugs.

When the software is working in the environment similar to production environment.

When the quality of software is very bad means there are too many blocker and critical bugs and you feel that is not eligible for further testing.

It is crossing the budget and deadline of the project taking too much time to complete so we can release the important feature like signup sign in and sign out.

**Test Automation:-** This section covers how we go about our sign up, sign in , add node add wallet , sign out to automate the feature so which tool we are using for automation like selenium, QTP, Appium

Which feature we are automating first like sign up sign in and sign out

Which automation framework we will use like BDD , Method driven, Cucumber, Hybrid.

How many automation engineers will be needed according to the feature and assign each feature to automation engineer.

**Roles & Responsibility:-** Test Manager will allocate the each project to each test engineer

**Test Manager:-** Write and review the test plan

Interact with product manager, development team, and management team

**Test Lead**:- Write or review the test plan.

Allocate the work every test engineers and ensure that they are completing the work within the schedule time(sprint).

**Test Engineer:-** Test Engineer should write the test cases and also reviewed by product manager.

Execute the test cases.

Perform all type of testing smoke, component, integration, system, Regression, API testing,

Write the tracebility matrix.

Perform testing on the different browser google chrome, Mozilla, firefox.

All the positive and negative scenario should be covered by test engineer.

**Deliverables:-** These are the outcomes of the testing team which I am going to give the Manager then manager will give to the customer. This is a final product.

**Release Note:-** It is a document which will be prepared during the project time. This will be given to the customer along with software. It consist of multiple things:-

1. Procedure of Installation of software.
2. Version of software.
3. Feature we have added.
4. Feature we have removed.
5. List of pending/open bug.
6. List of bugs fixed in current release.
7. Build ID