TEST PLAN FOR

TEAM AMAZE

ChangeLog

Version	Change Date	By	Description
version number	Date of Change	Name of person who made changes	Description of the changes made
V1.0	02/24/2020	Abhi, Akshay, Madi, Zack	Initialized test plan v1

1 Introduction

This Test plan describes the scope, approach, resources and schedule of testing activities for BisonCoin. The test plan includes the following components:

- Unit tests to test all service and blockchain functions
- Integration tests to test the service endpoints
- End-to-end (E2E) tests to test the front end
- A GitHub workflow that runs all the unit and integration tests for each PR and merge
- A staging branch which acts as a testing environment. All the changes are merged here and tested manually before merging them to the master branch.
- A PR practice that requires implementation of functional tests for all newly created endpoints, and unit tests for all newly created functions before changes can be approved.

1.1 Scope

1.1.1 In-Scope

The following features and requirements fall in the scope of this test plan:

Features:

- Creating a transaction
- Managing a wallet
- Managing mining
- Managing accounts

Functional requirements:

- All the functions implemented throughout the backend
- All the endpoints offered by the backend services
- All the components/flows implemented in the frontend

Non-Functional requirements:

- The validity of the transaction signatures to verify that a transaction is initiated by the owner of the wallet sending the money and ensure secure transactions.
- The validity of the login functionality to ensure secure account access during the session.
- Correct functioning of the frontend flow using end-to-end tests.

1.1.2 Out-of-Scope

The following aspects will not be tested:

- Performance of the system
- Scalability of the system

1.2 Quality Objective

This test plan is designed to achieve the following objectives:

- To ensure that the tests detect all the bugs and/or pieces of broken code before they make it into the master branch of the project.
- To ensure that the system conforms to the functional and non-functional requirements of the project.
- To ensure that the system meets the quality expectations of our clients and the development team.

1.3 Roles and Responsibilities

Name	Net ID	GitHub username	Role
Abhi Sachdev	sachdev1	abhisachdev17	Developer in Test
Akshay Sharma	sharmaa2	akshaysharma21	Developer in Test
Madison Fines	finesm3	madisonfines	Developer in Test
Zack Holmberg	holmbezt	ZackHolmberg	Developer in Test

2 Test Methodology

2.1 Overview

This project includes the following test methodologies:

- Unit testing
- Integration testing
- System testing

Incremental testing is used in agile development methods which allows for every release of the project to be tested thoroughly. This ensures that any bugs in the system are fixed before the next release. This form of testing allows changes in the project at any time to comply with the requirements. This form of also minimizes risk of a bug appearing on the production environment.

However, one downside of this method is the added time pressure on all stakeholders including the client, developers, and testers that arises due to instant client interaction.

2.2 Test Levels

This project includes the following 4 levels for testing:

- Unit tests
- Integration tests
- System/end-to-end tests
- GitHub workflow to run all the aforementioned tests whenever a PR is created or merged into either staging or master branches.

2.3 Test Completeness

- The test coverage is over 95% and includes all of the endpoints and helper functions in blockchain/services
- The Integration tests cover all endpoints and communication between the services.
- The production data is not affected. Stubs/mocks are created for endpoints that are not required in the tests.
- All components are tested through E2E tests.
- All tests are executed and passed every time a PR is created and merged.
- All bugs are fixed before the iteration is released.
- The bug PRs are thoroughly reviewed, and every PR has 2 approvals.

3 Test Deliverables

- Test plan
- Bug Reports
- Test Strategy
- Test Scenarios

4 Resource & Environment Needs

4.1 Testing Tools

- GitHub workflow for running suite of tests
- Shell script for executing all end-to-end tests
- GitHub issues/project board to document and track bugs

4.2 Test Environment

- Cypress software for end-to-end tests
- Pytest for backend unit and integration tests
- GitHub workflow for executing test suite
- Hardware with python, pip to install dependencies for the tests, and npm installed to execute the frontend tests.

5 Terms/Acronyms

Make a mention of any terms or acronyms used in the project

TERM/ACRONYM	DEFINITION
PR	Pull Request
E2E	End-to-end
npm	Node package manager