



UTM
UNIVERSITI TEKNOLOGI MALAYSIA

Software Quality Assurance – 2022/2023/2

Test plan

Project Test Plan – Group 3

FreeWork Freelancing Marketplace

1 Overview

1-1 Introduction

This document aims to provide the test plan and testing strategy for FreeWork project, which involves all the main functionalities of the system.

1-2 Scope

This document covers all the main project functionalities which includes different types and variance for testing the project.

1-3 References

- [Individual Reflective report \(IRR\)](#)
- [Project Github Repository](#)
- [Hood, C., & Wiebel, R. Software and systems engineering—Software testing—Part 4: Test techniques. Optimieren von Requirements Management & Engineering mit dem HOOD Capability Model, 29119-4.](#)
- [Vujičić, D., Jagodić, D., & Randić, S. \(2018, March\). Blockchain technology, bitcoin, and Ethereum: A brief overview. In 2018 17th international symposium infoteh-jahorina \(infoteh\) \(pp. 1-6\). IEEE.](#)

1-4 Glossary

The following is a glossary list related to this document:

Blockchain: a distributed network, where it keeps track of records in a chained approach, it is transparent and achieves its security from cryptography, each block which contains a set of transactions has a hash which identify and ensure the integrity of the latest data that has been stored.

MongoDB: non-sql database, it stores data in a document-based model and adopt a flexible approach unlike the regular column and row approach.

Structure-based testing: a testing technique which focuses on inspecting the internal code to design and perform the test cases.

Decision table testing: similar to if-else statements, this technique relies on a conditions and verify that the system behaves as expected under different environments.

Cause-effect graphing: focuses on identifying the inputs and outputs, and creating a graph to map out the chain of effects that leads to the output.

State transition testing: this technique focuses on the part of the system that has different states, it focuses on testing the transition between the state happens as expected.

Requirements-based testing: this approach relies on the software requirements, to develop and design the test cases, this approach is adequate to ensure that the system is meeting the client expectations.

User story testing: this approach breakdown the system into user stories which are segregated features that aims to meet the user's expectations.

Random testing: similar to fuzz testing, this approach relies on creating random combination of actions to stress test the system to ensure that all invariant and tests are valid, it aims to reach to an unexpected state that exposes a defect.

2 Overview / Context of Testing

2-1 Projects, Test Types and Test Levels

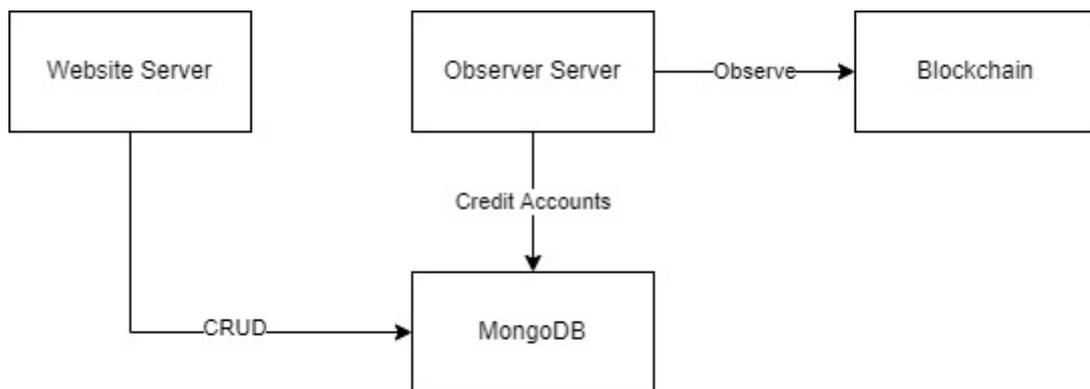
The FreeWork project consists of the following hardware components:

- Server (host website)
- Server (observe blockchain deposits)
- MongoDB Database
- Blockchain (Test Network - localhost)

The system consists of the following software modules:

- Smart Contract Token (Mock USDC)
- Account module
- Payment module
- Listings module
- Proposals module
- Order module
- Submission module

The hardware architecture is demonstrated in the following:



This test plan is applicable to the types and levels of testing listed in section 5-1.

2-2 Test Items

The test for this project includes testing of:

- Each module of the PC software, as listed in 2-1 above
- Each component of the PC software modules listed in 2-1 above
- The functionality of the complete software system

The versioning of each test unit will be accounted for and recorded as the tests are being executed to keep track of the progress.

2-3 Test Scope

The software modules are all listed in section 2-1. The Smart contract token module won't need any testing since in production environment we would be utilizing a third party stablecoin which we would be accounting for inflow and outflow, which can be deemed safe to utilize. For demonstration and testing purposes we will utilize a mock token. For non-functional factors such as performance, security and reliability will be outsourced to a third-party, since part of testing procedure is the system to be tested by multiple parties with special expertise to ensure an adequate software quality testing. All modules will be tested under the assumptions that the system components (OS, network...) are correctly working.

2-4 Test Basis

The requirements for the system are captured in the following documents:

- Individual Reflective report (IRR)

2-5 Assumptions and Constraints

AS-1: it is assumed that all team members will be conducting online meeting for convenience.

AS-2: it is assumed that all the data provided by the blockchain is secured and credible.

AS-3: we disregard the blockchain transaction 12 block confirmation time, we assume finality of transactions as soon as they get included in the mined block.

2-6 Stakeholders

Stakeholders are mainly two; Customers and Freelancers, where the former posts listings looking for applicants to get the job done, while the latter apply for it, in the hopes their proposal get accepted by the customer.

3 Testing Communication

Please refer to PP for a list of teams and individuals involved in test and project communication.

4 Risk Register

The following abbreviations are used in the risk tables:

- P = probability or likelihood of the risk
- C = Consequence, impact or effect if risk materializes
- E = Exposure = Probability x Consequence

The scales for both probability and consequence will be 1 to 6, where 6 is the highest.

4-1 Product Risks

Risk ID & Name	P	C	E	Mitigation Activities
1 Low quality work	2	6	12	Ensure that there are measures to limit the impact and encourage reputation-based model to reduce the exposure.
2 Payment disputes	3	6	18	A major risk, since the system is adopting an optimistic approach that the customer will approve the order on delivery and review. Thorough unit, system & acceptance testing
3 Risk of user data leakage	1	5	5	Design and code inspection, ensure industry standard security best practices. Thorough unit, system & acceptance testing
4 Fraudulent activities	2	6	12	Design and test the system for any possibilities for unexpected behavior or irregular activities within the platform, and thorough checking before publishing on the platform.
Risk ID & Name	P	C	E	Mitigation Activities

5 Technical difficulties	1	6	6	Perform regular survey, and provide contact support to ease the process for problem resolving
6 Regulatory challenges	2	5	10	Research and ensure that system is adhering to regulatory policies on the jurisdictions that the application operates on. Thorough unit, system & acceptance test
7 Risk that system crashes in middle of inspection	2	3	6	Planning for disaster/recovery testing to stress-test the system crashing during analysis

4-2 Project Risks

Risk ID & Name	P	C	E	Mitigation Activities
1 Risk that staff numbers in specific roles are insufficient	2	4	8	Ensure estimates consider staff capability requirements. Track test progress closely and report any predicted budget or schedule issues.
2 Risk that available staff lack sufficient knowledge and experience	2	3	6	Perform gap analysis of needs vs. staff availability. Prepare a training plan for individual participants. Include training time in the schedule. Find mentors, if possible. Ensure time is available for reviews by senior engineers.
3 Risk that some of the people who are available will not / cannot work together	5	2	10	Try to identify what the problem is. Arrange for arbitration if necessary / practicable. Where necessary, arrange for a Belbin analysis to increase understanding between the various types. Distribute activities so that there is less contact between the individuals in question.
4 Risk that there are too few licenses available for test execution tools	3	2	6	Prepare business case for additional licences. Distribute, plan and track test activities in detail, to reduce licence waiting times as far as possible. Report related problems as early as possible.
5 Risk that testing staff are unfamiliar with the test management tool	4	1	4	Prepare business case for funding tools training. Find someone from within the company who has experience of using the tool and share knowledge. Allow for additional time in initial test design & execution. Report related problems as early as possible.
Risk ID & Name	P	C	E	Mitigation Activities
6 Risk that vendors do not supply the expected material on time and to the expected quality	3	4	12	Ensure contractual requirements for quality and delivery are stated precisely, and if not, follow up with procurement. Provide templates to vendors wherever possible. Establish specific quality criteria for deliverables. Specify the consequences of breaches of quality and scheduled delivery. Follow progress and quality closely.

7 Risk that teams use more than 30 % of test execution time on incident reporting, causing schedule slippage	4	5	20	Review unit test coverage and adequacy. Request additional until testing if gaps are identified.
--	---	---	----	---

5 Test Strategy

5-1 Test Levels & Types

The test for the FreeWork product shall include the following test levels:

- T1: Unit Testing
- T2: Integration Testing (component level)
- T3: System Testing
- T4: System Integration Testing
- T5: User Acceptance Testing
- T6: Production Verification Testing

As stated in section 2-3, non-functional testing will be outsourced to a third-party supplier, with details to be provided in their Non-Functional Test Plan.

5-2 Test Deliverables

For each test level and type of testing, the following documentation will be produced:

- Test Plan
- Test Model Specification
- Test Cases, Test Procedures, Actual Results, Test Result, Test Log, Incident Reports and Test Status Reports in the organization's test management tool
- Test Summary Report

An overall Project Test Completion Report will also be produced at the end of the project, summarizing the results of testing across all levels of testing.

5-3 Test Design Techniques

Based on ISO/IEC/IEEE 29119-4:2021, we implement the following test techniques for each test level:

- T1: structure-based test design
- T2: decision table testing, cause-effect graphing and state transition testing.
- T3: Scenario testing
- T4: requirements-based testing
- T5: user story testing
- T6: random testing, and requirements-based testing

5-4 Entry and Exit Criteria

We have the following are the entry criteria on each test level:

- T1: Prepare all the units to be tested, and isolate them to perform the desired testing without any interference of one on another, unless there are dependencies.

T2: Gather the components to be tested, and ensure they are in a functioning state to perform tests.

T3: Setup the system to be tested, and ensure that all the processes are setup correctly to execute the set of system tests.

T4: Prepare the components, and ensure the setup of any dependencies to perform the tests to verify the components are working correctly as expected.

T5: Select the set of users that will perform the tests, and ensure that the system is ready to be tested with a ready and easy to use setup for regular user.

T6: Ensure that the system has all the components ready and tested from all previous testing levels with resolved defects if any have been found, prepare tests and target edge cases that target common attack vectors which can be costly to fix in production.

For all testing levels, the exit criteria will be that the expected outcome has been achieved on each level and tests have been concluded with a set of passed/failed tests, and if any defects have been found, it is reported and patched before going to the next testing level.

5-5 Test Completion Criteria

- All areas that carries a user data or financial risk, will undergo a full testing coverage, to reduce the risk exposure for the project, and aim to target unusual edge cases to achieve a thorough testing.
- While any area that is not part of the former will have sufficient testing to ensure a proper functioning of the system.

5-6 Degree of Independence

Unit testing will be conducted by developers. All other levels of testing will be carried out by independent testers.

5-7 Metrics to be Collected

The following metrics are to be collected during the testing procedure:

- The number of Expected and actual tests cases designed and executed, with their respective pass/fail ratio.
- Defect leakage, the percentage of defects that escaped from previous testing level.
- Defect age, representing the period of time that the defect went unnoticed.
- ROI on automated testing, representing the saved cost by running automated tests over manual ones.

5-8 Test Data Requirements

Sanitized data from the production environment will be provided in the system test environment.

5-9 Test Environment Requirements

The test environment will be setup to mimic production. Specific test tool requirements are:

— Github

— Jira

— VS Code

— Goerli Test Network

— Web3 Wallet

Additional environment details will be specified in a separate Test Environment Requirements document.

5-10 Retesting

It is recommended to run regular tests on the code as it is being developed, so that the system remains well functioning and there is less maintenance cost.

5-11 Regression Testing

The following are regression tests for each test level:

T1: Execute the same set of unit tests after each iteration or changes to ensure there are no defects or bugs are introduced.

T2: Outline the integration tests that cover the code that involves the changes, and re-execute the tests to ensure an expected outcome.

T3: Scope the system tests for the code changes that have been made and re-run the tests to ensure for them to pass safely.

T4: Identify the system integration tests that are impacted by the introduced changes, and battle-test for any unexpected behavior or failed tests.

T5: Prepare the suitable environment for the stakeholder to run the tests ensure that the system meets their needs and expectations.

T6: To ensure a safe production ready product, it is advised to run the related production verification tests which can prove whether or not the system is behaving as expected.

In the case any of the above tests fail, investigate the root cause, report and fix any defects that were found, while in the case nothing is found, a rigorous testing by writing test cases for special edge cases could help to make the system more robust.

5-12 Suspension and Resumption Criteria

Suspension should be proceeded with in the case, one of the CRUD operations fails or a login/registration issues are encountered.

5-13 Deviations from Organizational Test Practices

There are no expected deviations from the organizational test practices. Any deviations deemed necessary will be captured and communicated to relevant roles in a later stage.

6 Testing Activities and Estimates

The following are the testing activities to be conducted:

- Requirement analysis (1W)
- Test planning (4D)
- Test case design and development (10D)
- Test environment setup (1W)
- Test execution (5D)
- Test cycle closure (9D)

7 Staffing

7-1 Roles, Activities and Responsibilities

Further details about roles, activities and responsibilities will be provided in Test Plans of each level of testing.

We represent the responsibilities and roles on each testing level as a RACI matrix (R=Responsible, A=Accountable, C=Consulted, and I=Informed):

Phase\ Role	Project Manager	Test Manager	Developer	Test Analyst	Operation Staff	User Representative	Technical Support Staff	Quality Support Staff
T1	A	R	C	I	I	I	C	C
T2	C	R	C	I	I	I	C	C
T3	C	R	I	I	I	I	C	C
T4	C	R	I	I	I	I	C	C
T5	C	C	I	I	I	R	C	A
T6	R	C	I	I	I	I	C	A

7-2 Hiring Needs

Experienced and independent unit and integration testers will be needed, interviews will be proceeded with shortly.

7-3 Training Needs

Training in the company's test case management tool will be provided for any tester that do not have existing experience in the tool, via existing in-house training materials.

8 Schedule

The test schedule is provided in the following Gantt chart:

