ADMAS UNIVERSITY SCHOOL OF POST GRADUATE STUDIES

DEPARTMENT OF COMPUTER SCIENCE - SECTION 2

OBJECT ORIENTED SYSTEM DESIGN

CROSS-BORDER MONEY TRANSFER USING BLOCKCHAIN

Test Plan Document (TPD)

Project Members:

Name	ID No.
Seifu Birega	PGMGC/8062/20
Yordanos Woldu	PGMGC/8053/20
Mekonnen Ayalew	PGMGC/3118/19
Simeon Gebre Yohans	PGMGC/8067/20
Habtamu Birhan Godana	PGMGC/8023/20

Submitted to: Abdi Mulatu (Ass. Professor)

MARCH 13, 2021

Table of Contents

1. Introduction

1.1. Overview

Testing is critical in any system, but especially in a money transaction system such as CBMTS due to complex data transaction nature of cryptocurrency software/ application. In many complex software intensive systems, major problems can manifest at any point in its feature, and therefore comprehensive testing is crucial.

The CBMTS software has different features so the objective of the testing activities is to check function and features of a software elaborated for Android devices

1.2. Objective

The objective of prepare this test plan is to: -

- Identify existing project information and software component to be tested.
- Show the specific tasks/features of the software.
- Identify required resources and provide a test effort estimate.
- Identify the integration between actors and applications.
- Use the document as report, user guidance and SLA.

Notice:

The system implementation uses simple file storage and doesn't have any interface for the purpose of this project. This document is prepared as a standard for testing CBMTS, when user interface and full functionality of the system is implemented in the future.

The testing focuses on the following major functionalities – in a simple file storage system

- Block and Blockchain creation
- Validation of the Blocks
- Validation of Sufficient fund and transfer
- Transaction creation
- Creation of hashes

2. Relation between other documents

Test plan is a document detailing the objectives, resources, and process for specific test for a software or hardware product. The plan typically contains a detailed understanding of the eventual workflow. So, to define workflow of the CBMTS the requirement analysis design (RAD) and system design document (SSD) document of CBMTS should be refer to define the applicability, function and interface of the product.

- RAD consists functional and non-functional requirement of the systems, define actors, system interface and relationship between class.
- SSD guide database model, the relationship of the table and how the data transact on the system and how look like the main interface and architecture of the system.

3. Features to be tested or not to be tested.

3.1. Features to be tested

- Block and Blockchain creation
- Validation of the Blocks created
- Validation of Sufficient fund for transfer
- Transaction creation
- Creation of hashes

Notice: This test focuses only on the high level data flow diagram of the system presented in the design document.

The following types of tests are applied in the system except integration testing with other systems

- Unit testing – Gray Box approach
- Validation testing – Data validation and Block validation are tested for some functionalities
- System testing – The Security of the system is tested: By its nature Blockchain applications are secured because their data can't be affected or tampered after a Block is created.

3.2. Features not to be tested.

- Integration Testing
    - CBMTS does not integrate with other system.
- Interface design tests and all necessary data validations are not included in this project.

4. Testing Strategy or Approach

Testing strategy is an outline that describes the testing approach of the software development cycle. The purpose of a test strategy is providing a rational deduction from organizational, high level objective to actual test strategies to meet those objectives from quality assurance perspective.

4.1. Functional testing

The objective of functional testing is to make sure that the whole CBMTS software module works according to the requirement, and no significant errors appear in the application.

Functional testing is the most substantial part of CBMTS software testing. It involves checking of different aspect of the application. A software product must pass all the planned test. Only in this case its quality can be assured.

Notice: Is a proposed approach to the testing, not all functional tests on the figure are not included.

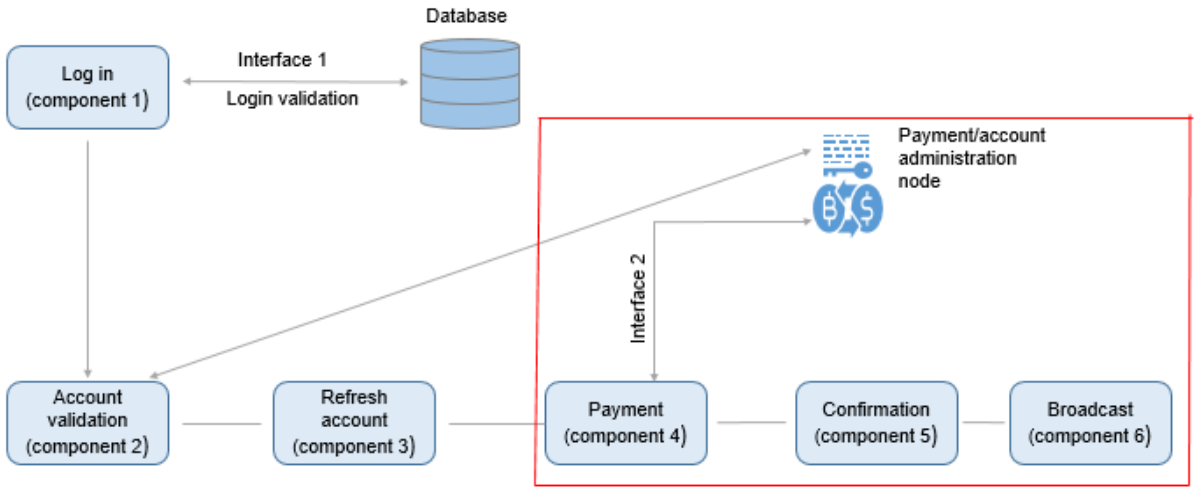

Fig 1: CBMTS functional testing diagram

4.2. Unit testing

Unit testing is a type of testing where individual units or components of a software are tested. The purpose is to validate that each unit of the CBMTS software code performs as expected. All components (Login, account validation, refresh transaction, payment, confirmation, and broadcast) of CBMTS software code will be test in order to prevent significant errors appear in the application.

5. Test case application

Test case is a set of actions executed to verify a particular feature or functionality of software application. Test case contains steps, test data, precondition, post condition for specific test scenario to verify any requirement. As stated on the test strategy CBMTS have different components the following test case need according to the requirement.

Notice: The test cases below are standard approaches and examples not part of the CBMTS. In the future after proper user interfaces are designed the test cases can be used.

5.1. Login test case

Test Scenario ID	Login-1	Test case ID	Login 1A
Test case description	Login – Positive test case	Test priority	High
Pre-requisite	NA	Post requisite	NA

Test execution steps:

s.no	Action	Input	Expect output	Actual output	Test device	Test result	Test comment
1	Launch app	Click app	CBMTS login interface	CBMTS login interface	Android device	Pass	[Yordi 14/03/2021 10:30:44AM]: launch successful

2	Enter user ID and password and hit login button	User ID: 1234 Password :*********	Home page	Home page	Android device	Pass	[Yordi 14/03/2021 10:32:01AM]: login successful

Test Scenario ID	Login-1	Test case ID	Login 1B
Test case description	Login – Negative test case	Test priority	High
Pre-requisite	NA	Post requisite	NA

Test execution steps:

s.no	Action	Input	Expect output	Actual output	Test device	Test result	Test comment
1	Launch app	Click app	CBMTS login interface	CBMTS login interface	Android device	Pass	[Yordi 14/03/2021 10:30:44AM]: launch successful
2	Enter user ID and password and hit login button	User ID: 1234 Password :*********	Home page	The user id or password you entered incorrect	Android device	Pass	[Yordi 14/03/2021 10:32:01AM]: invalid login attempt
3	Enter user ID and password and hit login button	User ID: 1234 Password :*********	Home page	The password you entered incorrect	Android device	Pass	[Yordi 14/03/2021 10:32:03AM]: invalid login attempt

5.2 Payment Transfer Test case

Test Scenario ID	Payment	Test case ID	Payment 1A
Test case description	Payment – positive test case	Test priority	High
Pre-requisite	NA	Post requisite	NA

Test execution steps:

s.no	Action	input	Expect output	Actual output	Test device	Test result	Test comment
1	Launch app	Click app	CBMTS login interface	CBMTS login interface	Android device	Pass	[Yordi 14/03/2021

							10:45:44AM]: launch successful
2	Enter user ID and password and hit login button	User ID: 1234 Password :*********	Home page	Home page	Android device	Pass	[Yordi 14/03/2021 10:45:46AM]: login successful
3	Select payment icon enter the receiver account number	enter the receiver account number and amount of money and click submit	Confirmation popup window	Confirmation pop up window	Android device	Pass	[Yordi 14/03/2021 10:45:48AM]: confirmation pop up displayed
4	Hit submit or cancel button	Click submit button	Payment is successfully done pop up and show the current amount	Payment is successfully done pop up and show the current amount	Android device	Pass	[Yordi 14/03/2021 10:45:49AM]: successful payment

Test Scenario ID	Payment	Test case ID	Payment 1B
Test case description	Payment – negative test case	Test priority	High
Pre-requisite	NA	Post requisite	NA

Test execution steps:

s.no	Action	Input	Expect output	Actual output	Test device	Test result	Test comment
1	Launch app	Click app	CBMTS login interface	CBMTS login interface	Android device	Pass	[Yordi 14/03/2021 10:45:44AM]: launch successful
2	Enter user ID and password and hit login button	User ID: 1234 Password :*********	Home page	Home page	Android device	Pass	[Yordi 14/03/2021 10:45:46AM]: login successful
3	Select payment icon enter	enter the receiver account	Confirmation popup window	The receiver account	Android device	Pass	[Yordi 14/03/2021

	the receiver account number	number and amount of money and click submit		number you entered incorrect please enter the correct account			10:45:48AM]: confirmation pop up displayed
4	Select payment icon enter the receiver account number	enter the receiver account number and amount of money and click submit	Confirmation popup window	You have no sufficient balance	Android device	Pass	[Yordi 14/03/2021 10:45:49AM]: successful payment

6. Glossary

CBMTS - Cross Border Money transfer System

RAD – Requirement Analysis Document

SDD – System Design document

TPD – Test Plan Document

SLA – Service Level Agreement

List of figures

No	Name	Description
1	Fig 4.1	Functional test Diagram
2	Fig 4.2	Unit Test Diagram

References

1. https://www.guru99.com/unit-testing-guide.html#:~:text=UNIT%20TESTING%20is%20a%20type,an%20application%20by%20the%20developers.
2. https://www.donnywals.com/getting-started-with-unit-testing-on-ios-part-2/
3. https://en.m.wikipedia.org/wiki/Test_strategy#:~:text=A%20test%20strategy%20is%20an,from%20a%20quality%20assurance%20perspective.
4. https://www.simform.com/functional-testing-types/
5. https://docs.python.org/3/library/unittest.html#:~:text=A%20test%20case%20is%20the%2C%2