U Mobile Cryptocurrency Prediction Software

Test Plan

Code Black

Name	Position	email	phone
Bernard Joshua	Team Leader/ Data Engineer	*103365867@student.swin.edu.au*	016-331 7910
Lionel Low	Backend Engineer	*103235180@student.swin.edu.au*	013-484 1402
Danial Imran	Frontend Engineer	*103701416@student.swin.edu.au*	012-644 5100
Ming Xuan	Frontend Engineer	*103701377@student.swin.edu.au*	012-360 9691

SWE40001-Software Engineering Project A, 4th Semester 1st May 2022

DOCUMENT CHANGE CONTROL

Version	Date	Authors	Summary of Changes
1	11/6/2022	Bernard Joshua	Introduction
1	12/6/2022	Danial Imran	Test Items
1	13/6/2022	Lionel Low	Strategy
1	14/6/2022	Ming Xuan	Pass Fail Criteria
2	15/6/2022	Bernard Joshua	Test Items/ Strategy / Pass Fail criteria
3	20/6/2022	Bernard Joshua	Final Proof-Reading and Editing

DOCUMENT SIGN OFF

Name	Position	Signature	Date
Bernard Joshua	*Team Leader/Data Engineer*	Bernard	**20th June 2022**
Lionel Low	*Backend Engineer*		**20th June 2022**
Danial Imran	*Frontend Engineer*	Danial Imran	**20th June 2022**
Ming Xuan	*Frontend Engineer*	MingXuan	**20th June 2022**

CLIENT SIGN OFF

Name	Position	Signature	Date
Chew Yew Choong	Head of Data Science & CVM		25/06/2022

Organisation
U Mobile Sdn Bhd

Contents

1. Introduction

1.1. Purpose of The Test Plan

The test plan is a document that outlines the scope, resources, technique, strategy, and criteria for all testing activities performed during and after system development. The strategy will explicitly indicate which features will be tested and which will not, as well as the stages at which testing will be performed.

1.2. Scope

As the system is developed, testing will take place at various stages of its life cycle. Testing ensures that bugs and defects are detected earlier on in the development process and allows the team to revise their approach to developing a specific part of the project. Because the system architecture and functionality have not achieved a low-level design yet, the test strategy is now confined to high-level test cases and assumptions. Because many parts of the system are still unknown, the test plan will serve as a high-level guide for testing for the time being and will be revised as the system evolves.

1.3. Constraints

The following are the identified constraints:

- The team is restrained to testing the software/system on the Azure Cloud platform.
- The team is restrained to test the software/system within the cost of $400 in Azure Credits.

1.4. Assumptions

The following are the identified assumptions:

- The testing will be done with the assumption that the user's have access to the internet.
- The testing will be done with the assumption that the user's are using the Chrome web browser or any other web browser that is compatible with Windows/MacOS/Linux operating systems.

1.5. Strategy

The software/system will be tested in four-phases :

- Unit Testing
- Integration Testing
- Systems Integration Testing
- Security Testing
- User Acceptance Testing

This will allow the project to be tested for maximum efficiency.

1.6. Reference Materials

The following document was prepared based on the information in the documents listed below. The documents listed below will provide additional information on the project.

- Project Plan
- Software Quality Assurance Plan
- Software Requirements Specification
- Test Plan
- Microsoft Azure Documentation

1.7. Objectives

To create a set of guidelines to assist the development and testing teams in carrying out the appropriate tests at the appropriate development phases in the Software Development Lifecycle.

1.8. Resources Required

The following are the resources that will be required for testing the software/system:

- Laptop
- Power Outlets for laptop charger
- GoCrypt prediction software with multiple user accounts.
- Good and stable internet access.

1.9. Environment Requirements

The following are the environmental requirements that will be required for testing the software/system:

- Azure Student Account
- Empty room that is optimally in a secluded area with no distractions of any kind.

2. Test Items

The product to be tested is the U-Mobile Crypto Prediction Software

2.1 Features to be Tested

- System must allow users to log in via their Google Account or Phone number.
- System must allow users to change/reset their passwords.
- System must display a graph to show the predicted performance of the crypto
- System must provide a numeric number on the accuracy of the prediction. (In percentage)
- System must allow the user to change the type of crypto to predict performance.
- System must allow users to save/mark crypto to follow for performance prediction.

- System must allow for users to receive notification about a specific cryptos performance if that crypto was marked
- System must allow administrator to notify the user about any changes to the system.

2.2 Functional Test Cases

Test Case(s)	**Steps**	**Expected Results**
Test case:	System Log In	
Check if users can login via their Google Account or Phone number.	**1.** User enters website **2.** User chooses Google Account or Phone number as log in option	System allows user to log in with google account or phone number and proceeds to dashboard.
	Password Reset	
Check if users can change/reset their passwords.	User clicks change/reset password button	**Change** – System prompts password change, insert old password, new password and confirm new password. **Reset** – System will send password confirmation to email or phone number
	Crypto Performance Graph	
Check if graph displayed to shows the predicted performance of the crypto.	User selects preferred crypto to see the predicted performance of the crypto.	User is able to see displayed graph that shows the predicted performance of the crypto when clicked.
	Prediction Accuracy Display	
Check if system provides a numeric number on the accuracy of the prediction. (In percentage)	User selects preferred crypto to see the predicted performance of the crypto.	User will be able to see accuracy of the prediction in percentage based on crypto chosen. Accuracy must be more than 40%

	Crypto Selectability	
Check if user can change the type of crypto to predict performance	**1.**User clicks on different crypto available **2.**Crypto clicked is chosen to predict performance	User is able to see different crypto chosen performance based on historical and real-time analysis
	Save/Mark Crypto	
Check if users can save/mark crypto to follow for performance prediction.	User clicks on preferred crypto to save/mark	System saves/marks crypto to allow user to follow performance prediction of preferred crypto.
	Notification of Performance	
Check if users receive notification about a specific cryptos performance if that crypto was marked.	User clicks on preferred crypto to save/mark	User will be notified of performance of saved/marked crypto
	Administrative Privilege	
Check if administrator can notify the user about any changes to the system.	Administrator is able to notify user about any changes made to the system.	User will receive notification of changes to the system by administrator

Table 3. Functional Test Cases

2.3 Non-functional Testing

Testing Goal: The purpose of the following test cases is to make sure that the application we develop has would not only be functional but be user friendly with a small learning curve allowing the average user to quickly adapt to the user interface.

Testing Procedure: The test would be performed on a set of participants, who will be asked to perform basic tasks using the applications. Upon completing the task, the user would be expected to be able to give feedback on the difficulty of the task and what their suggestion to improve the user interface would be.

Pass/Fail Criteria: The tasks given are very simple and would have constant pass criteria. Each task should be completed before 30 seconds and would have a difficulty goal of being less than 5(1 being very easy, and 10 being very hard)

Testing Constants: All of the tests will be carried out on the U-Mobile Crypto Prediction Software, using sample data provided to them.

Testing Assumptions: The tests will be carried out in a group of participants in an environment that is clear of any distractions and has a stable internet connection.

Test Case(s)	**Steps**	**Expected Results**
	Functional Suitability	
Functional completeness – Test if functions such as performance notification, marking the crypto for performance notification, prediction graphs and accuracy percentages	**1.** User will mark crypto performance notifications to test notification performance of system **2.** User will test prediction performance of graphs and accuracy percentages of cryptos	**1.** System will send notifications on time with no error message **2.** Prediction algorithm of system will display accurate information of graphs and accuracy percentage.
Functional Correctness – Check if prediction of the crypto is available on a weekly basis.	User will check on crypto predictions each week to see availability	System will display crypto predictions on a weekly basis.
	Usability	
Learnability – GUI interface - Check if the user is able to know what each part of the application should do when they use it for the first time.	User will go through the GUI application for the first time and should be able to understand what each part of the application does without assistance	User should be able to understand what each part of the application does without any problems
Operability – GUI interface - To check for operability, the user should be able to navigate within the application and operate each feature without problems to get their desired outcomes.	User will go through the GUI application and should be able to navigate and operate each feature	User should be able to navigate and operate each feature without any problems

Reliability		
Availability – Check if the user is able to use the application anytime when they need to. This is if the crypto markets are open	User will enter the application at different random times each day only when market is open to use crypto prediction software	System is available at all times for User to use except for when crypto market is not open
Recoverability - Check if the user is able to use the application and retrieve data in the event of a system failure.	User will enter the application during a system failure and will continue to use the application	User should be able to use the application and retrieve data in the event of a system failure.
Security		
Confidentiality – Check if login attempts made with non-existent accounts/ Attempt to access the application without an account result in a denied access.	User will attempt to log in with non-existent account to access application	System will not allow non-existent account to access the application
Integrity – Check if normal users cannot modify or access sensitive data.	Users will try to access and modify sensitive data.	System will not allow non-administrative users to access, edit, modify or change current sensitive data
Portability		
Adaptability - Testing out the applications use on Windows, Mac, Linux, Android, and iOS.	System will be tested on different operating systems such as Windows, Mac, Linux, Android, and iOS.	Application is able to run on all operating systems.
Maintenance		
Modifiability – Check if the functions can be modified without affecting the main application too much or at all	Administrator will modify functions while application is online	Application is able to run without problems while being modified by administrator

Testability - Testing out the new/modified functions and it passes the new or existing criteria’s set.	User will test the new/modified functions altered by administrators	Application functions runs smoothly and passes criteria’s set.
Performance		
Time behaviour - Monitoring time while calling the functions	Function time performance will be timed by user	Function time performance is within acceptable criteria range.
Resource utilisation - Monitoring of the resources using the PC’s inbuilt “Resource Monitor	Resource Monitor will be utilized to monitor resource utilisation such as RAM, CPU usage on test machine	Resource utilisation is within acceptable criteria range.

2.4 Features not to be Tested

Since all the features are being implemented for the first time and are all capable of being tested. There are currently no features that will not be tested

3. Strategy

As stated, the project will be tested following a five-level testing strategy

- **Unit Testing –** This stage of testing will be carried out during the initial coding stage when there are multiple modules of code being developed by different members of the team. This will be carried out by generating multiple unit tests to test pieces of code. Will help detect errors early on.
- **Integration Testing –** This stage of testing will be carried out when the system reaches a low-level design and all prior unit tests have been passed. Multiple modules of the system will be tested together using appropriate test cases. This will confirm if various modules function effectively altogether.
- **System Testing –** This stage will be carried out when the system reaches a functional level performing the core functions it was developed for. This will ensure if specified requirements are reached and product works. The system will undergo usability testing according to the non-functional test cases.
- **Security Testing –** This stage will be tested last as to see if the complete system is not vulnerable to any sort of malicious attacks by third-parties and is truly secured. This will ensure that the confidentiality and integrity of the system is ensured. This is because the system works with a lot of data and holds sensitive information that others may use with malicious intent.

- **Acceptance Testing –** The last stage of testing to make sure the client is happy with the product and all requirements have been achieved successfully.

3.1 Roles and Responsibilities

- **Unit Testing –** Must be carried out by individual developers on a continuous basis by creating unit tests throughout the development of the system.
- **Integration Testing –** May be carried out by testing managers (refer to Project Plan Document) by using appropriate test cases.
- **Security Testing –** May be carried out by testing managers (refer to Project Plan Document) by using appropriate scenarios.
- **System Testing –** Must be carried out by a separate testing team under supervision of the team leader. Users testing the system are preferred to be candidates who have experience with testing systems and interfaces. Recommended to get those users from a pool of developers from the clients firm.
- **Acceptance Testing –** Must be carried out by separate testing team as well the team leader under supervision of the client. Requires the users to be non - developers.

3.2 Test Deliverables

- **Test Plan** – Guidelines for testing purposes.
- **Defect/Enhancement logs** – Contains known bugs identified by the developer team as well as how to resolve them.
- **Test Reports** – Shows the results of the testing.
- **Usability Test Guide** – Test guide for usability evaluation.

3.3 Schedule

Table that displays schedule:

Features to test	**Estimated Testing**
System Log in and Password Reset	Semester 2
Crypto Prediction Graph	Semester 2
Prediction Accuracy	Semester 2
Crypto Selection and Save/Mark Crypto	Semester 2
Notification of Performance	Semester 2

Administrative Privileges	Semester 2

Table 5. Semester 2 Schedule

3.4 Risk and Contingency

Table displaying risks associated with testing:

Risks	**Contingency**
Frontend and Backend unable to communicate with each other	Carry out tests frequently as the system is developed instead of assigning a particular date.
Client may request for changes while performing testing	Communicate with client regularly so no last-minute changes are required.
Team member leaves	Have each module worked on and tested by 2 members.
Unable to complete full testing in time	Have testing done while developing instead of at the end.

Table 7. Risks associated with testing

3.5 Testing Tasks

- Testing scenarios shall be taken from the client and documented. The test scenarios will be required before creating the test cases.
- Test cases shall be created for individual features of the system.
- Required resources shall be accounted for before carrying out the tests.
- Bugs will be logged and reported along with how to resolve them.

4. Pass/Fail Criteria

4.1 Product Level

Testing Criteria	**Pass Criteria**	**Fail Criteria**

User login into personal account	User logged in and has access to their personal account.	User cannot log in successfully or not logged in to the correct account.
Server detects user login	Server detect user login successfully	Server did not detect user login at all
User greeted with the correct Home page	Home page greet user with correct information	Home page greet user with wrong information or did not greet at all
User access to Market page	User able to connect to Market page with stable connection	User not able to connect to Market page or wrong pages shown
Market page shows cryptocurrency	Market page shows correct cryptocurrency with correct information	Market page shows wrong information on cryptocurrency or not showing at all
User adds cryptocurrency to watchlist	User able to add cryptocurrency to watchlist successfully	User not able to add cryptocurrency to watchlist or wrong cryptocurrency is added
User removes cryptocurrency from watchlist	User able to removes cryptocurrency from watchlist successfully	User not able to removes cryptocurrency from watchlist successfully
Cryptocurrency graph shows prediction	Cryptocurrency graph shows correct prediction with a certain accuracy	Cryptocurrency does not show prediction or accuracy is lower than target
User logs out from personal account	User able to log out from personal account successfully	User is not able to log out

Table 8. Test Case Pass/Fail Criteria

4.2 Testing Stages

During each development stage test will be conducted and judged according to the following criteria

Testing Level	Pass criteria	Fail criteria

Unit Testing	● All unit tests have been passed successfully ● At least 80% of all code written has been covered during testing. ● All bugs and errors found have been logged and been accounted for.	● Not all unit tests have been passed ● Not enough code written has been included in testing.
Integration Testing	● 90% of all modules developed have been tested. ● Modules perform their assigned function successfully when tested together. ● All issues have been logged and corrected.	● Modules fail to carry out their function when put together. ● Too many critical issues found.
Security Testing	● 80% of security issues have been tested. ● Modules/System withstands all simulated security attacks. ● All issues have been logged and corrected.	● Modules/System fails withstand all simulated security attacks. ● Too many critical security issues found.
System Testing	● Entire System has been tested. ● 100% of all specified requirements have been successfully achieved. ● Minor Issues found have been logged and fixed. ● 100% of all system features functioning appropriately.	● Not all specified requirements have been achieved. ● Critical issues and defects found during the test. ● System features are not functioning accurately.
Acceptance Testing	● When client is satisfied with the product.	● Does not achieve the requirements specified. ● Critical issues found by client. ● Does not satisfy the client.

4.3 Suspension criteria and resumption requirements

4.3.1. Suspension Criteria

- The design of the system is found to be defective.
- Too many security issues found.
- The client has changed his mind about the specified requirements.

- The module tested is found to have critical issues which have to be fixed before proceeding further.

4.3.2. Resumption Requirements

- All critical issues found have been fixed.
- Critical security issues found and fixed.
- Modified or fixed modules have passed testing successfully.
- The new changes have gained the approval of the testing manager as well as the Team Leader.

4.4 Approvals

- Both the Team leader and the Testing manager must agree to the completion of a testing level before moving on to the next level.
- Any changes or addition to the system features would have to be approved by the client.