**TASK #1**

# Test Plan for Carbon Android App

By

John Kelvin Aondoakaa

1st Nov, 2020.

## Table of Contents

## 1.0 Introduction

The Test Plan is for Carbon Android App. It is designed to communicate the test approach to team members. It includes the objectives, scope, schedule, risks and approach. This document will clearly identify what the test deliverables will be and what is deemed in and out of scope.

### 1.1 Objectives

The purpose of this document is to describe in details the functional testing of a Carbon Android App. It will explain the purpose and features of the App, the functionalities and user requirements:

### 1.2 Team Members

| Resource Name | Role |
|---|---|
| Project Manager | Manage the entire project |
| Team Lead QA | Responsible for the entire testing process |
| Tester | Carry out the various test cases, report the results |

## 2.0 Scope

The Test Plan will include all 'must have' requirements. These and any other requirements that get included must all be tested. A tester must be able to:

1. Create a manual/Automation test with as many Test Cases/Steps as necessary
2. Save it
3. Retrieve it and have the ability to view it when running the test
4. Enter results and appropriate comments
5. View results
6. Report Bugs

As the team carryout QA test on the Android app, they will define the needs to include any other user stories where necessary.

Load testing will not be considered part of this project since the user base is known and not an issue.

### 2.1 Functional Requirements

The functional requirement is based on the **user stories**:

- Ability of user to download and install app
- Ability of user to successfully lunch app
- Ability of user to sign up/sign into the app.
- Ability of user to successfully recharge airtime
- Users' ability to add money (N1000) to their wallet
- Ability of user to view their records of wallet transaction.
- Ability of user's to filter transaction records based on date.
- Ability to logout/close app.

## 3.0 Assumptions / Risks for Test Execution

### 3.1 Assumptions

This section lists assumptions that are made specific to this project.

1. Use cases have been developed and approved by Lead QA for User Functionality testing
2. Test scripts are developed and approved
3. Major dependencies should be reported immediately after the testing kickoff meeting

### 3.2 Risks

The following risks have been identified and the appropriate action identified to mitigate their impact on the project.

| # | Risk | Impact | Trigger | Mitigation Plan |
|---|---|---|---|---|
| 1 | Changes to the functionality may negate the tests already written and test cases already written may be lost | High – to schedule and quality | Loss of all test cases | Export data prior to any upgrade, massage as necessary and re-import after upgrade. |
| 2 | Reported Bugs not fixed because developer works off site | Medium | Product did not get delivered on schedule | The QA Tester and Developer needs to sync so to fix reported bugs in due time |

## 4.0 Test Approach

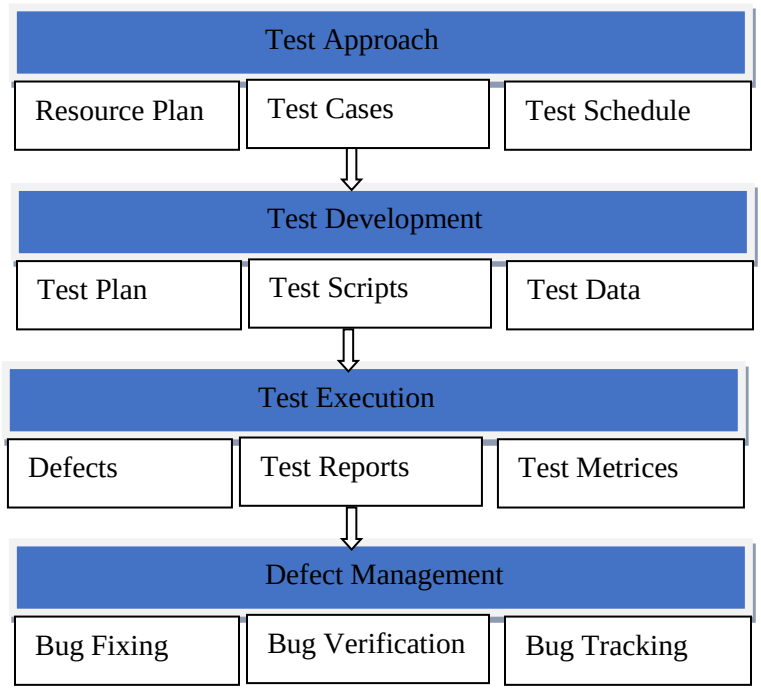


**Fig 1.0 Test Approach**

### 4.1 Functionality Test

The test approach is will be based on the user stories developed from the Business Requirement Document (BRD) and System Requirement Specifications (SRS).

| Project Name | Carbon Android App | | | |
|---|---|---|---|---|
| QA Officer | JOHN KELVIN AONDOAKAA | | | |
| Module Name | Functional Requirements | | | |
| Module Role | User | | | |
| Test Description | Ability of user to Register and login | | | |
| TEST #ID | TEST CASE | EXPECTED RESULT | ACTUAL RESULT | PASS/FAIL |
| 1. | | | | |
| 2. | | | | |

### 5.0 Test Environment

1. A functional Computer
2. Appium tools with TestNG or Cucumber framework are preferred. However, any open source tool with a framework, library and packages can be used.
3. Any Android device or Emulator that runs on Android 6 or up

### 5.1 Test Pass/Fail Criteria

Tests executed against the functional requirements will be used cases to determine pass or fail. If a test exhibits a product failure to meet the objectives of any of the requirements or the use cases, will fail and a defect/issue will be reported in the defect report.

### 5.2 Test Entry/ Exit Criteria

The Software Testing Life Cycle (STLC) specifies the entry criteria required during each testing phase. It also specifies the time interval or required amount of lead time to make the entry criteria item available to the process. The inputs can be divided into two categories inputs received from development and inputs produced from the test phases at the end of STLC.

## 6.0 Milestones / Deliverables

### 6.1 Test Schedule

The initial test schedule follows…….

| Task Name | Start | Finish | Effort | Comments |
|---|---|---|---|---|
| Test Planning | | | | |
| Review Requirements documents | | | Day1 | |
| Create initial test estimates | | | Day2 | |
| First deploy to QA test environment | | | Day2 | |
| Functional testing – Iteration 1 | | | Day3 | |
| Iteration 2 deploy to QA test environment | | | Day3 | |
| Functional testing – Iteration 2 | | | Day4 | |
| Resolution of final defects and final build testing | | | Day4 | |
| Deploy to Staging environment | | | | |
| Release to Production | | | | |

### 6.2 Deliverables

| Deliverable | For | Date / Milestone |
|---|---|---|
| Test Plan | Project Manager; QA Lead; Test Team | |
| Test Cases | QA Lead, Test Team | |
| Traceability Matrix | Project Manager; QA Lead | |
| Test Results | Project Manager | |
| Test Status report | QA Manager, QA Lead | |
| Metrics | All team members | |

### 6.3 Test Plan Approval

Signature: ---------- Date: ----------
Name: ---------- Date: ----------
Role: ----------Date: ----------

Signature: ---------- Date: ----------
Name: ---------- Date: ----------
Role: ----------Date: ----------