**TEST PLAN FOR CARBON MOBILE APPLICATION**

**Prepared By: Umeadi Ikem**

|  |  |  |
| --- | --- | --- |
| **Version** | **Author** | **Description of Change** |
| 1.0 | Umeadi Ikem |  |
|  |  |  |

1. **INTRODUCTION**

**1.1 Purpose**

This test plan describes the testing approach and overall process that will drive the testing of the Carbon Mobile Application. The document introduces:

* **Test Strategy:** The rules the test will be based on, including the givens of the project (e.g start / end dates, objectives, assumptions), description of the process to set up a valid test such as entry / exit criteria, creation of test cases, specific tasks to perform, scheduling, and data strategy.
* **Execution Strategy:** describes how the test will be performed and process to identify and report defects and to fix and implement fixes.
* **Test Management:**  The process to handle the logistics of the test and all the events that come up during execution such as communication, escalation procedures, risk and mitigation, team roster.
  1. **Project Overview**

The Carbon mobile app is a digital banking solution where users have access to features such as bill payments, money transfers, instant loans, high interest savings alongside other interesting features.

1. **TEST STRATEGY**
   1. **Test Objectives**

The objective of the test is to verify and ensure that the Carbon app works according to the business specifications and meets user and management needs.

The final product of the test is twofold:

1. Production-ready software.
2. A set of stable test scripts that can be reused for Automation and UAT test execution.
   1. **Test Principles**

* Testing will be focused on meeting the business objectives, cost efficiency and quality.
* There will be common, consistent procedures for all teams supporting testing activities.
* Testing processes will be well defined, yet flexible, with the ability to change as needed.
* Testing activities will build upon previous stages to avoid redundancy or duplication of effort.
* Testing will be a repeatable, quantifiable and measurable activity.
* Testing will be divided into distinct phases, each with clearly defined objectives and goals.
* There will be entrance and exit criteria.
  1. **Scope & Levels of Testing**

|  |  |  |
| --- | --- | --- |
| **Functional Testing** | **Non Functional Testing** | **Automated Testing** |
| Systems Testing | Performance Testing | UI Automation |
| Integration Testing | Load Testing | API Automation |
| Sanity Testing |  |  |
| Regression Testing |  |  |
|  |  |  |

1. **EXECUTION STRATEGY**

**3.1 Validation & Bug Management**

The bugs will be tracked through the Jira Project board. The product manager/product owner can gather information on a daily/weekly basis from the Jira board, and request additional details from QA if necessary. The development team would work on fixes.

It is the responsibility of the tester to open the bugs, link them to the corresponding script assign an initial severity and status, retest and close the bug. It is the responsibility of the development team to review the bugs raised, ask for details if necessary, fix the defect, communicate to QA when the fix is done, and implement the solution.

Bugs found during Testing will be categorized according to the bug-reporting tool, and the categories are:

|  |  |
| --- | --- |
| **Severity** | **Impact** |
| 1. (Critical) | The bug is critical enough to crash the system, because file corruption, or cause potential data loss.  It causes an abnormal return to the operating system (crash or a system failure message appears).  It causes the application to hang and requires rebooting the system. |
| 1. (High) | It causes a lack of vital program functionality with workaround. |
| 1. (Medium) | It causes a lack of vital program functionality with workaround. |
| 1. (Low) | There is an insufficient or unclear error message, which has minimum impact on product use. |
| 1. (Lower) | There is an insufficient or unclear error message that has no impact on product use. |

**3.2 Test Metrics**

Test metrics to measure the progress and level of success of the test will be developed and shared with the project/product manager. The below are some of the metrics

|  |  |  |
| --- | --- | --- |
| **Report** | **Description** | **Frequency** |
| Test preparation & Execution Status | To report on % complete, %Not Executed, %Passed, %Fail, %Blocked  Defects Status – open, closed, any other status | Weekly/Daily |
| Daily Execution Status | To report on Pass, Fail, Total Defects, highlight Blockers/Critical defects | Daily |
| Project Weekly Status Report | Project driven reporting (As requested by PM or Test Lead) | Weekly |

1. **TEST MANAGEMENT PROCESS**

**4.1 Test Management Tool**

All testing artifacts such as Test cases, test reports are updated on Sheets/Jira as the case may be.

* During the Test Design phase, all test cases are written directly on Google Sheets /Jira. Any change to the test case will be directly updated on either platform.
* Each Tester will directly access their respective assigned test cases and update the status of each executed step on the Jira Board/Sheet.
* During Bug Fix retesting, bugs are reassigned to the tester to validate the bug fix. The tester validates the bug fix and updates the status directly on JIra board/Sheets.
* Reports can be generated from the Jira Project board/Sheet to provide status of Test Execution.

**4.2 Test Execution Process**

* Once all test cases have been approved and the test environment is ready for testing, the tester would start an exploratory test of the application to ensure the application is stable for testing.
* Testers are to ensure necessary access to the testing environment, Jira Board/Sheets for updating test status and raising bugs. If any issues, it can be escalated to the Test Lead/Product Manager as an escalation.
* If there are any blockers during exploratory testing, it would be escalated to the respective development team for fixes.
* Each tester performs step by step execution and updates the execution status. The tester enters Pass or Fail Status for each of the steps directly in Jira board/Sheets.
* If there are any errors, bugs will be raised as per severity guidelines in bug management system detailing steps to simulate along with screenshots/videos if appropriate.
* Daily Test Execution status to be reported in Stand Ups.
* This process is repeated until all test cases are executed fully with Pass/Fail status.

**4.3 Role Expectations**

The following list defines in general terms the expectations related to the roles directly involved in the management, planning or execution of the test for the project.

**4.3.1 Project Management**

Project Manager sets Timelines, interfaces with clients, provides & reviews documentation, tracks progress.

**4.3.2 Test Planning (Test Lead)**

* Develop a test plan and the guidelines to create test conditions, test cases, expected results and execution scripts.
* Provide guidelines on how to manage bugs.
* Attend status meetings in person or online.
* Communicate to the test team any changes that need to be made to the test deliverables or application and when they will be completed.
* Provide reports and non-technical breakdown for management where needed.

**4.3.3 Test Team**

* Develop test conditions, test cases, expected results, and execution scripts.
* Perform execution and validation.
* Identify, document and prioritize bugs according to the guidance provided by the Test lead.
* Re-test after software modifications have been made according to the schedule.
* Prepare testing metrics and provide regular status.

**4.3.4 Development Team**

* Support the development and testing processes being used for the project.
* Certify correct components have been delivered to the test environment at the points specified in the testing schedule.
* Keep the project team and leadership informed of potential software delivery date slips based on the current schedule.
* Define processes/tools to facilitate the initial and ongoing migration of components.
* Conduct first line investigation into execution discrepancies and assist test executors in creation of accurate bugs.
* Implement fixes to bugs according to schedule.

1. **TEST ENVIRONMENT**

* Staging apk’s would be provided by dev team.

1. **SCHEDULES**

**Test Deliverables**

|  |  |
| --- | --- |
| **Test Life Cycle Stage** | **Deliverables** |
| Capture Business Requirements | Requirements Coverage |
| Plan Tests | Test Plan |
| Develop Test Cases/Scripts & Scenarios | Test Scripts/Cases |
| Prepare Test Data | Test Data |
| Test Execution | Status Report/Execution Report |
| Raise Bugs/Track & Monitor Bugs | Bug Report |
| System / Functional Testing Test Report | Test Execution Report |
| Performance Testing | Performance Test Analysis Report |
| Test Automation | Regression Test Suite |
| Test Closure | Final Test Report |

1. **TOOLS**

Bugs would officially be tracked on Jira. All bugs should be logged on the Jira Project Board.

**Other Tools**

* Cypress
* Performance & Load Test: Jmeter, k6 & Chrome Lighthouse
* API Test: Postman, Newman, Cypress
* Communication: Slack
* Continuous Integration: Jenkins, Dockers