TEST PLAN

Carbon Paylater Mobile Application

Prepared By:

Owojaye Olutoye

Document History

Version	Author	Revision Date	Approved by	Outline
1.0	*Olutoye Owojaye*	*30/03/2019*		*Initial Draft*

Approvers List - To track who has reviewed and signoff on the Test plan

Name	Role	Approver / Reviewer	Approval / Review Date

Reference Documents - Clearly mark the document used as an input to create the test plan

Version	Date	Document Name
1.0	4th March, 2020	Business and Functional Spec Document
1.0		Carbon GUI Prototype

TABLE OF CONTENTS

1.0 INTRODUCTION

The Carbon app lets you control your finances with a few clicks. You can get instant short-term loans for urgent needs, check your credit reports, invest money to earn high-interest rates, recharge airtime on your mobile phone, and make bill payments for services.

1.1 Purpose

This document describes the plan for testing the **Carbon Paylater Android Application** System. This Test Plan document supports the following objectives:

- Identify existing project information and the software that should be tested.
- List the recommended test requirements (high level).
- Recommend and describe the testing strategies to be employed.
- Identify the required resources and provide an estimate of the test efforts.
- List the deliverable elements of the test activities.

1.2 Scope

This Test Plan describes the integration and system tests that will be conducted on the Paylater Application following the integration of the subsystems and components identified in the process Flow. Testing of system functionality and features will be conducted based on documented requirements.

The interfaces between the following subsystems will be tested:

- Social Authentication Engines
- Payment routing Gateway or Engines
- Short Messaging Service

1.3 Referred Document

- Functional and Technical Specification Document
- Business Requirement Document
- Carbon GUI Prototype

1.4 Audience

- Project team members perform tasks specified in this document, and provide input and recommendations on this document.
- Project Manager Plans for the testing activities in the overall project schedule, reviews the document, tracks the performance of the test according to the task herein specified, approves the document and is accountable for the results.
- The stakeholders' representatives and participants may take part in the UAT test to ensure the business is aligned with the results of the test.
- Technical Team ensures that the test plan and deliverables are in line with the design, provides the environment for testing and follows the procedures related to the fixes of defects.
- Business analysts will provide their inputs on functional changes.

1.5 Test Risks and Mitigations

Risk	Prob.	Impact	Mitigation Plan
SCHEDULE Testing schedule is tight. If the start of the testing is delayed due to design tasks, the test cannot be extended beyond the UAT scheduled start date.	High	High	• The testing team can control the preparation tasks (in advance) and the early communication with involved parties. • Some buffer has been added to the schedule for contingencies, although not as much as best practices advise.
RESOURCES Not enough resources, resources on boarding too late (process takes around 15 days.	Medium	High	Holidays and vacation have been estimated and built into the schedule; deviations from the estimation could derive in delays in the testing.
DEFECTS Defects are found at a late stage of the cycle or at a late cycle; defects discovered late are most likely be due to unclear specifications and are time consuming to resolve.	Medium	High	Defect management plan is in place to ensure prompt communication and fixing of issues.
SCOPE Scope completely defined	Medium	Medium	Scope is well defined but the changes are in the functionality are not yet finalized or keep on changing.
Non-availability of Independent Test environment and accessibility	Medium	High	Due to non-availability of the environment, the schedule gets impacted and will lead to delayed start of Test execution.

2.0 TEST REQUIREMENTS

The listing below identifies those items (use cases, functional requirements and non-functional requirements) that have been identified as targets for testing. This list represents what will be tested.

2.1 Functional Testing (Integration)

Process Flow:

- The application should leverage on social authentication engines to handle fast user registration and logging in
- Payment Engine or gateway will be integrated to the application to handle transaction routing
- Short Message Service (SMS) will be integrated to the application to notify customers of transaction status

2.2 User Interface Testing

- Verify ease of navigation through a sample set of screens.
- Verify sample screens conform to GUI standards.
- The System shall be easy-to-use and shall be appropriate for the target market of the public (Importers) and individuals.

2.3 User Acceptance Test (UAT)

- UAT test execution will be performed by end-users and QA team will provide support creating a UAT script
- End-users will validate the application's robustness and usability
- End-users will validate the application's conformity to business requirements

2.4 Regression Testing

- Regression testing will cater for the integration of a new feature to the existing application
- Test automation will be employed to ease the stress of executing existing testcases or test suite

2.0 TEST STRATEGY

The Test Strategy presents the recommended approach to the testing of software applications. The previous section on Test Requirements described what will be tested; this describes how it will be tested.

The main considerations for the test strategy are the techniques to be used and the criterion for knowing when the testing is completed.

In addition to the considerations provided for each test below, testing should only be executed using known, controlled databases, in secured environments.

The final product of the test is twofold:

- A production-ready software;
- A set of stable test scripts that can be reused for Functional and UAT test execution.

3.1 Test Principles

- Testing will be focused on meeting 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 environment and data will emulate a production environment as much as possible.
- 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 an entrance and exit criteria.

1.2 Functional Testing

Testing of the application should focus on any target requirements that can be traced directly to use cases (or business functions) and business rules. The goals of these tests are to verify proper data acceptance, processing, retrieval and the appropriate implementation of the business rules. This type of testing is based upon black-box techniques that is; verifying the application (and its internal processes) by interacting with the application via the GUI and analyzing the output (results).

Test Objective:	Ensure proper application navigation, data entry, processing, and retrieval.
Technique:	• Execute each use case, use case flow, or function, using valid and invalid data, to verify the following: • The expected results occur when valid data is used. • The appropriate error/warning messages are displayed when invalid data is used. • Each business rule is properly applied.
Completion Criteria:	• All planned tests have been executed. • All identified defects have been addressed.
Special Considerations:	• Access to the CARBON application is required to run the identified System tests on staging

3.3 User Interface Testing

User Interface testing verifies a user's interaction with the software. The goal of UI Testing is to ensure that the User Interface provides the user with the appropriate access and navigation through the functions of the applications. In addition, UI Testing ensures that the objects within the UI function as expected and conform to corporate or industry standards.

Test Objective:	Verify the following: • Navigation through the application properly reflects business functions and requirements, including window to window, field to field, and use of access methods (tab keys, mouse movements, accelerator keys) • Window objects and characteristics, such as menus, size, position, state, and focus conform to standards.
Technique:	• Create/modify tests for each window to verify proper navigation and object states for each application window and object.
Completion Criteria:	Each window successfully verified to remain consistent with the benchmark version or within acceptable standard
Special Considerations:	• Not all properties for custom and third-party objects can be accessed.

3.4 User acceptance testing (UAT)

User acceptance testing (UAT) is formal testing with respect to user needs, requirements, and business processes conducted to determine whether or not a system satisfies the acceptance criteria and to enable the users, customers or other authorized entity to determine whether or not to accept the system

Test Objective:	Verify the following: • All business requirement and logic are achieved • Robustness and usability of CARBON
Technique:	• Execute each use case, use case flow, or function, using valid and invalid data, to verify the following: • The expected results occur when valid data is used. • The appropriate error/warning messages are displayed when invalid data is used. • Each business rule is properly applied.
Completion Criteria:	All business logic and process flow are successfully verified to remain consistent with benchmark version or within acceptable standard
Special Considerations:	Identified defects are documented as Change Requests for a future release

3.5 Deliverables

S/N	Deliverable Name	Author	Reviewer
1.	Test Plan	QA	Head of Dept/ Test Lead
2.	Functional Test Cases	QA	Business Analyst's Sign off
3.	Logging Defects	QA	Test Lead/ Programming Lead
4.	Daily/weekly status report	QA	Test Lead/ Business Analyst
5.	UAT Script	QA	Business Analyst's Sign off
6.	Test Closure report	QA	Head of Dept/ Project Manager

4.0 EXECUTION STRATEGY

This section defines all actions, resources, tools, and techniques that will be considered during the execution and implementation of the test strategy.

4.1 Entry and Exit Criteria

- The entry criteria refer to the desired conditions in order to start test execution; only the migration of the code and fixes need to be assessed at the end of each cycle.
- The exit criteria are the desirable conditions that need to be met in order to proceed with the implementation.
- Entry and exit criteria are flexible benchmarks. If they are not met, the test team will assess the risk, identify mitigation actions and provide a recommendation. All this is input to the project manager for a final "go-no go" decision.
- Entry criteria to start the execution phase of the test: the activities listed in the Test Planning section of the schedule are 100% completed.
- Entry criteria to start each cycle: the activities listed in the Test Execution section of the schedule are 100% completed at each cycle.

Exit Criteria	Test Team	Technical Team	Notes
100% Test Scripts executed			
95% pass rate of Test Scripts			
No open Critical and High severity defects			
95% of Medium severity defects have been closed			
All remaining defects are either canceled or documented as Change Requests for a future release			
All expected and actual results are captured and documented with the test script			
All test metrics collected based on reports from TFS			
All defects logged on TFS			
Test Closure Memo completed and signed off			
Test environment cleanup completed and a new back up of the environment			

4.2 Test Cycles

- There will be two cycles for functional testing. Each cycle will execute all the scripts.
- The objective of the first cycle is to identify any blocking, critical defects, and most of the high defects. It is expected to use some workaround in order to get to all the scripts.
- The objective of the second cycle is to identify remaining high and medium defects, remove the workaround from the first cycle, correct gaps in the scripts and obtain performance results.
- UAT test will consist of one cycle.

4.3 Test Management Tools

Microsoft Team Foundation Server is the tool used for Test Management. All testing artifacts such as Test cases, test results are updated in the Microsoft Team Foundation Server (TFS) tool and google sheet. Microsoft Team Foundation creates an enabling environment among project stakeholders to collaborate and deliver a stable application. The following are the steps involved in test management processes

- A project-specific folder structure will be created on TFS to manage the status of this IDEC project.
- Each resource in the Testing team will be provided with Read/Write access to add/modify Test cases in TFS.
- During the Test Design phase, all test cases are written directly into TFS and google sheet.
- Any change to the test case will be directly updated in the TFS.
- The Tester will directly access assigned test cases and update the status of each executed step in TFS directly.
- Any defect encountered will be raised in TFS linking to the particular Test case/test step.
- During Defect fix testing, defects are re-assigned back to the tester to verify the defect fix. The tester verifies the defect fix and updates the status directly in TFS.
- Various reports can be generated from TFS to provide the status of Test execution. For example, Status report of Test cases executed, Passed, Failed, No. of open defects, Severity wise defects, etc.

4.4 Resources

This section presents the recommended resources for testing IDEC, their main responsibilities, and their knowledge or skill set.

Human Resources		
Role	**Minimum Resources Recommended** (number of workers allocated full-time)	**Specific Responsibilities/Comments**
Test Designer	Quality Assurance Analyst /Manager	Identifies, prioritizes, and implements test cases Responsibilities: • Generate a test plan • Generate Test Suite • Evaluate the effectiveness of test effort
System Tester	Software Tester	Executes the tests Responsibilities: • Execute tests • Log results • Recover from errors • Document defects
Test System Administrator	Support Department	Ensures test environment and assets are managed and maintained. Responsibilities: • Administer test management system • Install / manage worker access to test systems
Business Analyst	Business Analyst and Project Manager	Analyze content of the Test Plan, Responsibilities: • Test Strategy meets system requirements. • Communicate new requirement • Facilitate defect communications between the testing team and the development team
Development team	Software Developers	Ensure defect fixes and cleanups Responsibilities: • Certify correct components that have been deployed to the test environment at the points specified in the testing schedule. • Implement fixes to defects according to schedule