**TEST PLAN FOR LAMBDA APP**

| Date | Version | Description | Author | Reviewer | Approver |
| --- | --- | --- | --- | --- | --- |
| 12/4/2024 | 0.1.0 | Test plan creation | Reuben Amissah |  |  |
|  |  |  |  |  |  |

1. **INTRODUCTION**

1.1. Purpose

1.2. Project Overview

1.3. Audience

2.  **QUALITY OBJECTIVES**

2.1. Test Objectives

2.1.1 Primary Objectives

2.1.2. Secondary Objectives

3. **SCOPE**

3.1. Features/Roles To Be Tested

3.2. Features No To Be Tested

4. **TEST APPROACH**

5. **TEST MANAGEMENT PROCESS**

5.1.Test Management Tool

5.2. Test Design Process

5.3. Test Execution Process

5.4. Test Risks and Mitigation Factors

6. **ENTRY AND EXIT CRITERIA**

6.1 Entry Criteria

6.2. Exit Criteria

6.3. Suspension Criteria

6.4. Resumption Criteria

7. **TEST STRATEGY**

7.1. Test Assumptions

7.2. Test Principles

7.3.. Data Approach

7.4 QA role in the test process

7.4.1 Understanding Requirements:

7.4.2 Preparing Test Cases:

7.4.3 Preparing Test Matrix:

7.4.4. Reviewing test cases and matrix:

7.4.5 Creating Test Data:

7.4.6 Executing Test Cases:

7.4.7 Retesting and Regression Testing:

7.4.8 Deployment/Delivery:

7.5. Testing types

7.5.1. Exploratory

7.5.2. Functional Test

TEST ACCEPTANCE CRITERIA

TEST DELIVERABLES

MILESTONE LIST

7.5.3. User Acceptance Test (UAT)

TEST DELIVERABLES

7.5.4. Performance Test

TEST DELIVERABLES

7.5.5. Security Test

TEST DELIVERABLES

7.5.6. Automation Test

TEST DELIVERABLES

7.6. Test Cycles

7.7. Validation and Defect Management

7.9. Defect Tracking & Reporting

8.0 **RESOURCES AND TEST ENVIRONMENT**

8.1. Test Tools

8.1. Test Environment

5.1.1. Support Level 1 (W browsers):

8,1.2. Support level 1 (devices, IOS and Android)

8.2. Configuration Management

9. **ROLES AND RESPONSIBILITIES**

9.1. Role Expectations

10. APPROVALS

# 1. **INTRODUCTION**

## 1.1. **Purpose**

The Test Plan has been created to facilitate communication among the team members. This

document describes approaches and methodologies that will apply to the testing of the Lamda Mobile App. It includes the objectives, test responsibilities, entry and exit criteria, scope, schedule major milestones, entry and exit criteria, and approach. This document has clearly identified what the test deliverables will be, and what is deemed in and out of scope.

## 1.2. **Project Overview**

Lambda App is a powerful web and mobile app to enables users to shop online for Phones, Gadgets, Cameras, etc., with seamless payment and successful delivery of products to customers, Customers will be able to use the frontend website to search the Lambda Products and place the order by making online payment of the order in advance.

## 1.3. **Audience**

1. Project team members perform tasks specified in this document, and provide input and recommendations on this document.
2. The 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.
3. The stakeholders’ representatives and participants may take part in the UAT test to ensure the business is aligned with the results of the test.
4. The Technical Team ensures that the test plan and deliverables are in line with the design, provides an environment for testing, and follows the procedures related to the fixes of defects.
5. Business analysts will provide their input on functional changes.

# **2. QUALITY OBJECTIVES**

## 2.1. **Test Objectives**

### **2.1.1 Primary Objectives**

A primary objective of the test is to ensure that the Lambda App system meets the full requirements, including quality requirements (functional and non-functional requirements) and fit metrics for each quality requirement, works according to business specifications, satisfies the use case scenarios, and maintain the quality of the product. At the end of the project development cycle, the end users/client should find that the project has met or exceeded all of their expectations as detailed in the requirements.

The final product of the test is in two folds:

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

### **2.1.2. Secondary Objectives**

The secondary objectives of testing will be to: identify and expose all issues and associated risks, communicate all known issues to the project team, and ensure that all issues are addressed in an appropriate manner before release. As an objective, this requires careful and methodical testing of the application to first ensure all areas of the system are scrutinized and, consequently, all issues (bugs) found are dealt with appropriately

# 

# 

# **3. SCOPE**

| Functional & Non-Functional Testing | Automated Testing | Security Testing |
| --- | --- | --- |
| User Acceptance Testing | API Automation | Vulnerability Scanning |
| Usability Testing | GUI Automation | Security Scanning |
| Integration Testing | Continuous Integration |  |
| Smoke/Sanity Testing |  |  |
| Regression Testing |  |  |
| End-to-End Testing |  |  |
| Compatibility Test |  |  |
| Performance(Load & Stress Test) |  |  |

## **3.1. Features**

1. Sign Up
2. Login
3. Search
4. Add to cart

## **3.2. Features No To Be Tested**

1. Not other than mentioned above in section 3.1

# **4. TEST APPROACH**

The approach, that used, is Analytical, therefore, in accordance to requirements-based

strategy, where an analysis of the requirements specification forms the basis for planning,

estimating and designing tests. Test cases will be created during exploratory testing. All

test types are determined in the Test Strategy.

The team also must use experience-based testing and error guessing to utilize testers' skills and

intuition, along with their experience with similar applications or technologies.

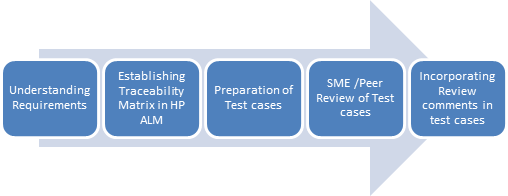
# **5. TEST MANAGEMENT PROCESS**

## **5.1.Test Management Tool**

All testing artifacts such as Test cases, and test results are updated in the Prospa Defect Management System tool (JIRA, Spira Team, Excel Sheet).

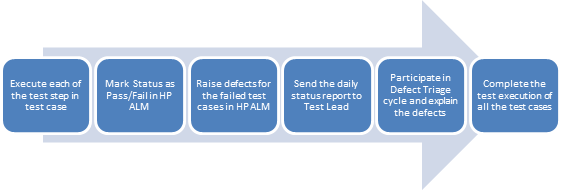
* The project-specific folder structure will be created in the Prospa defect management system to manage the status of this project.
* Each resource in the Testing team will be provided with Read/Write access to add/modify Test cases in the Prospa Defect Management System.
* During the Test Design phase,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

## **5.2. Test Design Process**



* The tester will understand each requirement and prepare a corresponding test case to ensure all requirements are covered.
* Each Test case will be mapped to Use cases to Requirements as part of the Traceability matrix.
* Each of the Test cases will undergo review by the PM and the review defects are captured and shared with the Test team. The testers will rework the review defects and finally obtain approval and sign-off.
* During the preparation phase, the tester will use the prototype, use cases, and functional specifications to write step-by-step test cases.
* Testers will maintain a clarification Tracker sheet and the same will be shared periodically with the Requirements team and accordingly, the test case will be updated. The clarifications may sometimes lead to Change Requests or not in scope or detailing implicit requirements.
* Sign-off for the test cases would be communicated through mail by PM.
* Any subsequent changes to the test case if any will be directly updated in Prospa Defect Management System.

## **5.3. Test Execution Process**



* Once all Test cases are approved and the test environment is ready for testing, the tester will start an exploratory test of the application to ensure the application is stable for testing.
* Testers to ensure necessary access to the testing environment, Prospa defect management system for updating test status and raising defects. If any issues will be escalated to the Test Lead and in turn to the Project Manager as escalation.
* Any show-stopper during exploratory testing will be escalated to the respective development SPOCs 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 the Prospa Project management system.
* If any failures, defects will be raised as per severity guidelines in the Prospa defect management system tool detailing steps to simulate along with screenshots if appropriate.
* Daily Test execution status as well as Defect status will be reported to all stakeholders.
* The testing team will participate in defect triage meetings in order to ensure all test cases are executed with either pass/fail category.
* If there are any defects that are not part of the steps but could be outside the test steps, such defects need to be captured in the Prospa defect management system and mapped against the test case level or at the specific step that issue was encountered after confirming with the Test Lead.
* This process is repeated until all test cases are executed fully with Pass/Fail status.
* During the subsequent cycle, any defects fixed applied will be tested and the results will be updated

As per Process, the final sign-off or project completion process will be followed

## **5.4. Test Risks and Mitigation Factors**

| Risk | Prob. | Impact | Mitigation Plan |
| --- | --- | --- | --- |
| **SCHEDULE**  The 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 (the process takes around 15 days. | Medium | High | Holidays and vacations have been estimated and built into the schedule; deviations from the estimation could result 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 due to unclear specifications and are time-consuming to resolve. | Medium | High | The defect management plan is in place to ensure prompt communication and fixing of issues. |
| **SCOPE**  Scope completely defined | Medium | Medium | The scope is well defined but the changes in the functionality are not yet finalized or keep on changing. |
| Natural disasters | Low | Medium | Teams and responsibilities have been spread to two different geographic areas. In a catastrophic event in one of the areas, there will be resources in the other areas needed to continue (although at a slower pace) the testing activities. |
| Non-availability of Independent Test environment and accessibility | Medium | High | Due to the non-availability of the environment, the schedule gets impacted and will lead to the delayed start of Test execution. |
| Delayed Testing Due to New Issues | Medium | High | During testing, there is a good chance that some “new” defects may be identified and may become an issue that will take time to resolve.  There are defects that can be raised during testing because of unclear document specifications. These defects can yield an issue that will need time to be resolved.  If these issues become showstoppers, it will greatly impact the overall project schedule.  If new defects are discovered, the defect management and issue management procedures are in place to immediately provide a resolution. |

# **6. ENTRY AND EXIT CRITERIA**

## **6.1 Entry Criteria**

The entry criteria refer to the desirable 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.

* All test hardware platforms must have been successfully installed, configured, and functioning properly
* All the necessary documentation, design, and requirements information should be available that will allow testers to operate the system and judge the correct behavior
* All the standard software tools including the testing tools must have been successfully installed and functioning properly
* Proper test data is available.
* The test environment such as lab, hardware, software, and system administration support should be ready.
* QA resources have completely understood the requirements
* QA resources have sound knowledge of functionality
* Reviewed test scenarios, test cases, and RTM

## **6.2. Exit Criteria**

* 100% Test Case executed
* 95% pass rate of test cases
* A certain level of requirements coverage has been achieved
* All defects logged in JIRA/Google sheets
* No open Critical and High-severity defects
* All expected and actual results are captured and documented with the test script
* All high-risk areas have been fully tested, with only minor residual risks left outstanding.
* Test Closure Memo completed and signed off
* Test environment cleanup completed and a new backup of the environment

## **6.3. Suspension Criteria**

* The build contains many serious defects which seriously limit testing progress.
* A significant change in requirements suggested by the client
* Software/Hardware problems
* Assigned resources are not available when needed by the test team.

## **6.4. Resumption criteria**

* Resumption will only occur when the problem(s) that caused the caused the suspension have been resolved

# 

# **7. TEST STRATEGY**

## **7.1. Test Assumptions**

**Key Assumptions**

Production like data required and be available in the system prior to the start of Functional Testing

In each testing phase, Cycle 2 will be initiated if the defect rate is high in Cycle 1.

**General**

* Exploratory Testing will be carried out once the build is ready for testing
* Performance testing would be carried out in a development environment.
* Automation test would be carried out by the QA team
* Security testing would be carried out by the QA team
* Defects would come along with a snapshot JPEG format or URL links.
* The Test Team will be provided with access to the Test environment
* The Test Team assumes all necessary inputs required during Test design and execution will be supported by Development/PM appropriately.
* Test case design activities will be performed by the QA team
* Test environment and preparation activities will be owned by the Dev Team
* The dev team will provide Bug-fix plans based on the retrospective meetings during each cycle plan. The same will be informed to the Test team prior to the start of Bug fix cycles
* The PM will review and sign off all Test cases prepared by the Test Team prior to the start of Test execution.
* The Bugs will be tracked through the Prospa defect management tool only. Any bug fixes planned will be shared with the Test Team prior to applying the fixes on the Test environment.
* The project Manager will review and sign off all test deliverables
* The project will provide test planning, test design, and test execution support
* The test team will manage the testing effort with close coordination with Project PM.
* The project team has the knowledge and experience necessary or has received adequate training in the system, the project, and the testing processes.
* There is no environment downtime during the test due to outages or bug fixes.

The system will be treated as a black box; if the information shows correctly online and in the reports, it will be assumed that the database is working properly.

## **7.2. 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.
* The testing environment and data will emulate a production environment as much as possible.
* Testing will be 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.

## **7.3.. Data Approach**

* In functional testing, the Lambda App System will or may contain pre-loaded test data which is used for testing activities.

## **7.4 QA role in the test process**

### **7.4.1 Understanding Requirements:**

* Requirement specifications will be sent by the client.
* Understanding of requirements will be done by QA

### **7.4.2 Preparing Test Cases:**

* QA will be preparing test cases and scripts based on the exploratory testing. This will cover all scenarios for requirements.
* Updating of test cases when required

### **7.4.3 Preparing Test Matrix**:

* QA will be preparing a test matrix that maps test cases to respective requirements. This will ensure the coverage for requirements.

### **7.4.4. Reviewing test cases and matrix:**

* Peer review will be conducted for test cases and test matrix by the QA Lead.
* Any comments or suggestions on test cases and test coverage will be provided by reviewer's respective Author of the Test Case and Test Matrix
* Suggestions or improvements will be re-worked by the author and will be sent for approval
* Re-worked improvements will be reviewed and approved by the reviewer

### **7.4.5 Creating Test Data:**

* Test data will be created by respective QA on the client's developments/test site based on scenarios and Test cases.

### **7.4.6 Executing Test Cases:**

* Test cases will be executed by respective QA on the client's development/test site based on designed scenarios, test cases, and Test data.
* Test results (Actual Result, Pass/Fail) will be updated in the test case document Defect Logging and Reporting:
* QA will be logging the defects/bugs in a Word document, found during execution of test cases. After this, QA will inform the respective developer about the defect/bugs.

### **7.4.7 Retesting and Regression Testing:**

* Retesting for fixed bugs will be done by the respective QA once it is resolved by the respective developer and bug/defect status will be updated accordingly. In certain cases, regression testing will be done if required.

### **7.4.8 Deployment/Delivery:**

* Once all bugs/defects reported after complete testing is fixed and no other bugs are found, the report will be deployed to the client’s test site by PM.
* Once round of testing is done by QA on the client’s test site if required Report will be delivered along with sample output by email to respective lead and Report group.
* QA will be submitting the filled hard copy of delivery slip to respective developer.
* Once lead gets the hard copy of delivery slip filled by QA and developer, he will send the report delivery email to client.

## **7.7. Validation and Defect Management**

* It is expected that the testers execute all the scripts in each of the cycles described above. However it is recognized that the testers could also do additional testing if they identify a possible gap in the scripts. If a gap is identified, the scripts and traceability matrix will be updated and then a defect logged against the scripts.
* The defects will be tracked through Prospa’s defect management tool (Git-lab). The technical team will gather information on a daily basis from Prospa Defect Management System, and request additional details from the Defect Coordinator. The technical team will work on fixes.
* It is the responsibility of the tester to open the defects, link them to the corresponding script, assign an initial severity and status, retest and close the defect; it is the responsibility of the Defect Manager to review the severity of the defects and facilitate with the technical team the fix and its implementation, communicate with testers when the test can continue or should be halt, request the tester to retest, and modify status as the defect progresses through the cycle; it is the responsibility of the technical team to reviewProspa Defect Management System on a daily basis, ask for details if necessary, fix the defect, communicate to the Defect Manager the fix is done, implement the solution per the Defect Manager request.

Defects found during the Testing will be categorized according to the bug-reporting tool “Prospa Defect Management System ” and the categories are:

| **Severity** | **Impact** |
| --- | --- |
| 1 (Critical) | This bug is critical enough to crash the system, cause file corruption, or cause potential data loss  It causes an abnormal return to the operating system (a crash or a system failure message appears).  It causes the application to hang and requires rebooting the system. |
| 2 (High) | It causes a lack of vital program functionality with a workaround. |
| 3 (Medium) | This Bug will degrade the quality of the System. However, there is an intelligent workaround for achieving the desired functionality - for example through another screen.  This bug prevents other areas of the product from being tested. However other areas can be independently tested. |
| 4 (Low) | There is an insufficient or unclear error message, which has a minimum impact on product use. |
| 5(Cosmetic) | There is an insufficient or unclear error message that has no impact on product use. |

**7.8. Test Metrics**

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

| **Report** | **Description** | **Frequency** |
| --- | --- | --- |
| Test Preparation & Execution Status | To report on % complete, %Not Executed, % Pass, % Fail, % Blocked  Defects severity-wise Status – Open, closed, any other Status | Weekly / Daily |
| Daily execution  status | To report on Pass, Fail, Total defects, highlight Showstopper/ Critical defects | Daily |
| Project Weekly Status Report | Project-driven reporting (As requested by PM) | Weekly – If the project team needs weekly updates apart from daily and there is a template available with the project team to use. |

# 

# 

# **8.0 RESOURCES AND TEST ENVIRONMENT**

## **8.1. Test Tools**

The following tools will be used to facilitate and execute the testing of Ebonylife Place Indigo V2 in collaboration with all parties involved.

| **S/N** | **Process** | **Tools** |
| --- | --- | --- |
| **1** | Performance Test | Jmeter  BlazeMeter  Load Runner |
| **2** | Automation & Continuous Integration | Cypress  Katalon  Jenkins  Docker |
| **4** | Security | Web application vulnerability scanners  Nexus  Burp Suite |
| **5** | Test case execution and tracking | Google sheet & Jira |
| **6** | Bug Tracking tools | Testrail & Google Sheet |
| **7** | Communication Tools | Slack & Teams |
| **8.** | Checklist Creating | Google sheet |

# 

Lambda App will be hosted on two environments i.e production and development environment

## **8.1. Test Environment**

#### **5.1.1. Support Level 1 (Web browsers):**

* **Windows 10:** Edge, Chrome (latest), Firefox (latest), Safari (latest)
* **Mac OS X:** Chrome (latest), Firefox (latest), Safari (latest)

#### **8,1.2. Support level 1 (devices, IOS and Android)**

* iPhone 11, 12, 13 etc.
* Samsung Galaxy A23

#### **8.2. Configuration Management**

**9. ROLES AND RESPONSIBILITIES**

## **9.1. 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.

| SN0. | Roles | Name | Responsibilities |
| --- | --- | --- | --- |
| 1. | Quality Assurance Engineer | Olusoga F. David | * Analysis and understanding of functional requirements and research of past defects * Define Scenarios: Develop User Stories * Test Plan Design * Test Design & Execution * Test cases development * Perform sanity tests on new releases in preparation for end2end system testing * Setup test management tool * Dependency Analysis based on processes * Ensure quality standards, test procedures and methodology are adhered to * Assist with developing and maintaining quality procedures and procedure documents. * Communicate quality standards and principles to project team member * Assist with identifying and evaluating continuous process improvement opportunities. * Test and analyse each component/phase of the systems development life cycle including the project requirements, design and programs. * Maintain test deliverables including test approach, scenarios, conditions & expected results, test cycle control sheet, test scripts. * Resolve, document and report problems that arise during testing. * Document and communicate results from testing. * Assist with conducting quality-control tests and analyses to ensure that the software meets or exceeds specified standards and end-user requirements. * Retest corrections to ensure problems are resolved. * Provide support to Business users * Coordinate with QA Lead for any issues or problems encountered during test preparation/execution/defect handling. |

# 

# 

# **10. APPROVERS**

|  | **Name** | **Role** | **Organization** | **Signature** | **Date** |
| --- | --- | --- | --- | --- | --- |
| **1** | Reuben Amissah | Quality Assurance Engineer |  |  |  |
| **2** |  | Project Manager |  |  |  |
| **3** |  | Development Lead |  |  |  |