Yalla Kora

“YallaKora”

Project Test Plan

Version 1.0

Author: Omar Abo Smrh

Index

Document Revision History

Date	Version	Description	Author	Reviewer	Approver
5-8-22	1.0	Test Plan was created	O. Smrh		
00-00-00	1.1	PO editions	O. Smrh		

Introduction

Instabug want YallaKora Product, which should pass the full cycle of testing. Given the specificity of the product it is very important to have the same quality.

The Test Plan has been created to facilitate communication within the team members. This document describes approaches and methodologies that will apply to the unit, integration and system testing of the "YallaKora" app. It includes the objectives, test responsibilities, entry and exit criteria, scope, schedule major milestones and approach. This document has clearly identified what the test deliverables will be, and what is deemed in and out of scope.

Scope

The document mainly targets the Functional testing and validating data in report output as per Requirements Specifications provided by Client or Product Owner.

Functions to be tested:

- Business Flow Logic
- GUI
- Calculations Logic
- Configurations
- Reports Logic
- Performance
- Security

Quality objectives

Primary Objectives

A primary objective of testing is to: assure that the app meets the full requirements, including quality requirements (functional and non-functional requirements) and fit metrics for each quality requirements and satisfies the use case scenarios and maintain the quality of the product. At the end of the project development cycle, the user should find that the project has met or exceeded all of their expectations as detailed in the requirements.

Any changes, additions, or deletions to the requirements document, Functional Specification, or Design Specification will be documented and tested at the highest level of quality allowed within the remaining time of the project and within the ability of the test team.

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 matter before release. As an objective, this requires careful and methodical testing of the application to first ensure all areas of the software are scrutinized and, consequently, all issues (bugs) found are dealt with appropriately.

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. All test types are determined in Test Strategy.

Team also must use Experienced-Based Testing, along with their experience with similar applications or technologies.

The project is using an agile approach (Scrum), with 2 weeks iterations. At the end of each sprint the requirements identified for that iteration will be delivered to the team and will be tested.

Roles and Responsibilities

Role	Staff Member	Responsibilities
Project Manager	X.Y	1. Acts as a primary contact for development and QC team. 2. Responsible for Project schedule and the overall success of the project.
QC	O. Smrh	1. Participation in the project plan creation/update process. 2. Planning and organization of test process for the release.

Role	Staff Member	Responsibilities
QC	O.Smrh	3. Report progress on work assignments to the PM. 4. Understand requirements. 5. Writing and executing Test cases. 6. Preparing RTM. 7. Reviewing Test Cases and RTM. 8. Defect reporting and tracking 9. Retesting and regression testing.

Entry and Exit Criteria

Entry Criteria

- All the necessary documentation, design, and requirements information should be available that will allow tester(s) to operate the software 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 hardware, software, and software administration support should be ready.
- QC resources have completely understood the requirements.
- QC resources have sound knowledge of functionality.
- Reviewed test scenarios, test cases and RTM.

Exit Criteria

- 100% of requirements coverage has been achieved (at least in the 1st release).
- No high priority or severe bugs are left outstanding.
- All high-risk areas have been fully tested.
- Cost – when the budget has been spent (planned man-hour).
- The schedule has been achieved.

Suspension criteria and resumption requirements

Suspension criteria

- The build contains many serious defects which seriously or limit testing progress.
- Significant change in requirements suggested by client.
- Software/Hardware problems.
- Assigned resources aren't available when needed by tester(s).

Resumption criteria

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

Test strategy

QC role in test process

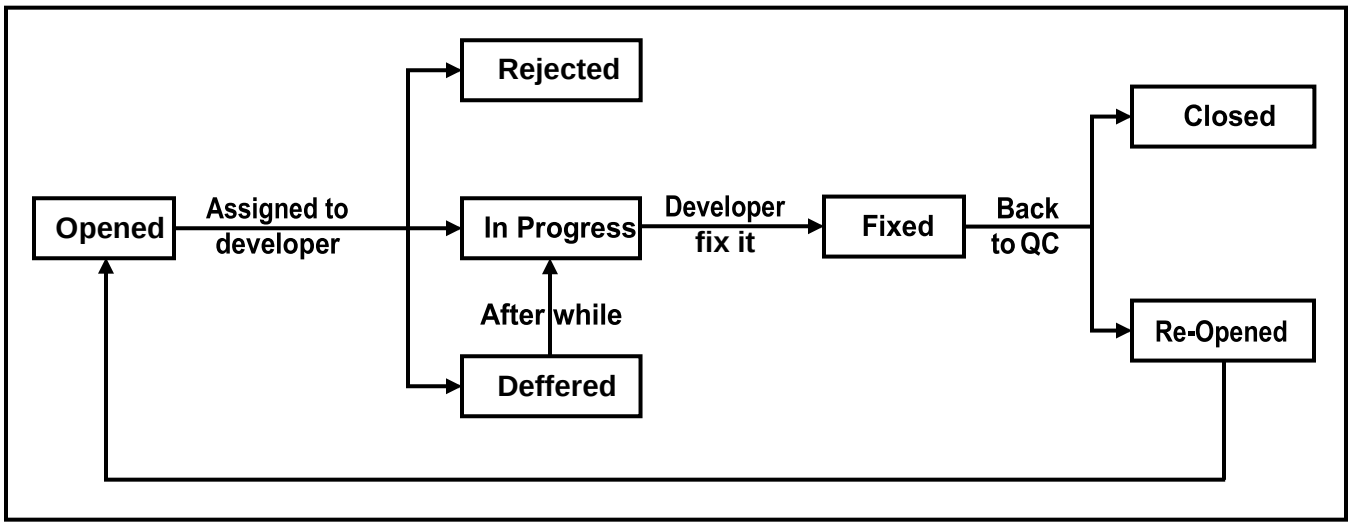
- Understanding requirements
  - Requirement specifications will be sent by client.
  - Understanding of requirements will be done by QC.
- Preparing Test Cases
  - QC will be preparing test cases based on the exploratory testing. This will cover all scenarios for requirements.
- Preparing Test Matrix
  - QC will be preparing test matrix which maps test cases to respective requirement. This will ensure the coverage for requirements.
- Reviewing test cases and matrix:
  - Peer review will be conducted for test cases and test matrix by QC Lead.
  - Any comments or suggestions on test cases and test coverage will be provided by reviewer respective author of Test Case and Test Matrix.
  - Suggestions or improvements will be re-worked by author and will be send for approval.
  - Re-worked improvements will be reviewed and approved by reviewer.

- Creating Test Data
    - Test data will be created by respective QC on Product Owner's site based on scenarios and Test Cases.
- Executing Test Cases
    - Test cases will be executed by respective QC on Product Owner's site based on designed scenarios, test cases and test data.
    - Test result (Actual Result, Pass/Fail) will be updated in test case document Defect Logging and Reporting
        - QC will be logging the defects/bugs in Jira/TFS or Excel (if TFS isn't existing), found during execution of test cases. After this, QC will inform respective developer about the defects/bugs.
- Retesting and Regression Testing
    - Retesting for fixed bugs will be done by respective QC once it is resolved by respective developer and bug/defect status will be updated accordingly. In certain cases, regression testing will be done if required.
- Development/Delivery
    - Once all bugs/defects reported after complete testing is fixed and no other bugs are found, report will be deployed to client's test site by PM.
    - Once round of testing will be done by QC on client's test site if required report will be delivered along with sample output by email to respective lead and report group.
    - QC 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'll send the report delivery email to client.

Bug life cycle

All the issues found while testing will be logged into Jira/ TFS or Excel (if TFS isn't existing).

Bug life cycle for this project is as follows



Testing types

Black box testing

This kind of testing focuses on the functional requirements of the software. It enables one to derive sets of input conditions that will fully exercise all functional requirements for a program.

GUI testing

GUI testing will include testing the UI part of report. It covers users report format, look and feel, error messages, spelling mistakes, GUI guideline violations.

Integration testing

Integration testing is systematic technique for constructing the program structure while conducting test to uncover errors associated with interacting. In report, integration testing includes the testing report from respective location(s).

Functional testing

Functional testing is carried out in order to find out unexpected behavior of the report. The characteristic of functional testing is to provide correctness, reliability, testability and accuracy of the report output/data.

Performance testing

Performance testing of a function is testing the software according to performance criteria like response time on a particular operation/function.

Security testing

Security testing of a function is testing the software according to security criteria such as SQL injection and all security threats.

System testing

System testing of software is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements.

User acceptance testing

The purpose behind user acceptance testing is to conform that system is developed according to the specified user requirements and is ready for operational use. Acceptance testing is carried out at two levels – Alpha and Beta testing. User acceptance testing (UAT) will be done by the client.

Alpha testing

The alpha test is conducted at the developer's site by Product Owner (PO).

Beta testing

The beta test is conducted at the client's site by him/her.

Bug severity and priority definition

Bug severity and priority fields are both very important for categorizing bugs and prioritizing if and when the bugs will be fixed. The bug severity and priority levels will be defined as outlined in the following tables below. Testing will assign a severity level to all bugs. The test lead will be responsible to see that a correct severity level is assigned to each bug.

The QC Lead, Development Lead and Project Manager will participate in bug review meetings to assign the priority of all currently active bugs. This meeting will be known as "Bug Triage Meetings". The QC Lead is responsible for setting up these meetings on a routine basis to address the current set of new and existing but unresolved bugs.

Severity List

The tester entering a bug into Jira/TFS or Excel (If TFS isn't existing) is also responsible for entering the bug Severity.

Severity ID	Severity	Severity Description
1	Critical	The module/product crashes or the bug causes non – recoverable conditions. System crashes, or database or file corruption, or potential data loss, program hangs requiring reboot are all examples of Sev.1 bug.
2	High	Major system component unusable due to failure or incorrect functionality. Sev.2 bugs cause serious problems such as a lack of functionality, or insufficient or unclear error messages that can have a major impact to the user, prevents other areas of the app from being tested, etc. Sev.2 bugs can have a work around, but the work around is inconvenient or difficult.
3	Normal	Incorrect functionality of component or process. There is a simple work around for the bug if it is Sev.3.
4	Low	Documentation errors or signed off severity 3 bugs.

Priority List

Priority	Priority Level	Priority Description
1	Must Fix	This bug must be fixed immediately; the product cannot ship with this bug.
2	Should Fix	These are important problems that should be fixed as soon as possible. It would be an embarrassment to the company if this bug shipped
3	Fix when have time	The problem should be fixed within the time available. If the bug doesn't delay shipping date, then fix it.
4	Low Priority	It isn't important (at this time) that these bugs be addressed. Fix these bugs after all other bugs have been fixed. Enhancements/ Good to have features is also priority.4.

Resource and environment needs

Testing Tools

Process	Tool
Test case creation	Jira/ Excel / TFS
Test case tracking	Jira / Excel / TFS
Test case execution	Manual/automation
Test case management	Jira / Excel / TFS
Defect management	Jira /TFS
Test reporting	PDF
Checklist creating	Microsoft Word

Configuration Management

- Documents CM: Jira/TFS.
- Code CM: Jira/TFS.

Test Environment

Support Level	Operating System (OS)	mobile
Support Level 1	Android	Samsung, OPPO, Huawei, Infinix, Honor
Support Level 2	APPLE IOS Microsoft OS	iPhone (Latest Version) Lumia (Latest Version)
Support Level 3	Anything else	Anything else

Test Schedule

Task Name	Start	Finish	Effort	Comments
Test Planning	5-8-2022	5-8-2022		First time to make a plan
Review Requirements Documents				
Create test basis				Not ready yet
Staff and train new test Resources				
First deploy to QC test Environment				Already installed
Functional testing – Sprint 1				
Sprint 2 deploy to QC test environment				
Functional testing – Sprint 2				
Sprint 3 deploy to QC test environment				
Functional testing – Sprint 3				
Sprint 4 deploy to QC test environment				
Functional testing – Sprint 4				
Sprint 5 deploy to QC test environment				
Functional testing – Sprint 5				
Sprint 6 deploy to QC test environment				
Functional testing – Sprint 6				
Sprint 7 deploy to QC test environment				
Functional testing – Sprint 7				
Sprint 8 deploy to QC test Environment				
Functional testing – Sprint 8				

Task Name	Start	Finish	Effort	Comments
Sprint 9 deploy to QC test environment				
Functional testing – Sprint 9				
System testing				
Regression testing				
UAT				
Resolution of final defects and final build testing				
Deploy to Staging environment				
Release to Production				

Approvals

	Project Manager	QC Leader
Name	X.Y	Omar Smrh
Signature		

Terms/Acronyms

Term/Acronym	Definition
GUI	Graphical User Interface
PM	Project Manager
PO	Product Owner
UAT	User Acceptance Testing
CM	Configuration Management
QC	Quality Control
RTM	Requirements Traceability Matrix