Table of Contents

Test Plan

Project “MASIK”

Document Revision History

Date	Version	Description	Author	Reviewer	Approver
05.05	0.1	Test plan was created	A.Troitska		

1. INTRODUCTION

The customer wants to receive a perfect mobile application that has passed a full cycle of manual testing. Testing a mobile application is not just about finding bugs or crashes. It is a holistic approach that ensures the overall quality, performance and usability of the application. The test plan was created to facilitate communication between team members. This document describes the approaches and methodologies that will be used for unit, integration and system testing of the MASIK mobile application. It includes goals, tester responsibilities, entry and exit criteria, scope, schedule milestones, and approach. This document clearly defines what the test results will be and what is considered to be included and what is not included in the scope.

2 SCOPE

The Scope section outlines the boundaries and focus of testing. Clearly defining the scope of testing helps set expectations, focus testing efforts, and ensure critical aspects of a mobile photo sharing app are thoroughly covered. The main focus is on ensuring the

convenience, functionality, performance and security of the photo sharing feature.

2.1 Functions to be tested.

- Upload and publish functionality
- Sharing photos with other users
- Commenting on photos
- Tagging users in photos
- Privacy and security
- Photo sharing performance in various network conditions to ensure smooth uploading and viewing of photos.
- User Experience (UX): Evaluate the overall user experience of photo sharing, including ease of use, intuitive design, and responsiveness.

2.2 Functions not to be tested.

- Advanced image editing capabilities
- Internal server testing or third-party integrations that are not part of the core photo sharing functionality.
- Testing the performance of the program when downloading and publishing photos in conditions of low connection quality.

3. QUALITY OBJECTIVES

3.1 Primary Objectives

The main purpose of testing is to: ensure that the system meets the complete requirements, including the quality requirements (functional and non-functional requirements) and the metrics for each quality requirement, and satisfies the usage scenarios and maintains the quality of the product. At the end of the project development cycle, the user should find that the project meets or exceeds 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 to the highest level of quality permitted within the time remaining to complete the project and within the capabilities of the testing team.

3.2 Secondary Objectives

The secondary objectives of testing are to: identify and expose all issues and associated risks, communicate all known issues to the project team, and ensure that all issues are resolved properly before release. This requires careful and methodical testing of the program to first ensure a thorough check of all areas of the system and, accordingly, the proper resolution of all problems (errors) found.

4. TEST APPROACH

The approach used is analytical and proactive, in accordance with a requirements-based strategy, where the analysis of the requirements specification is the basis for planning, evaluation and test development. Test cases will be created during exploratory testing. All types of tests are defined in the Testing Strategy.

The team should also use experiential testing and error guessing, using the skills and intuition of the testers and their experience with similar applications or technologies.

The project uses a flexible approach with weekly iterations. At the end of each week, the requirements defined for that iteration will be provided to the team and tested.

4.1 Test Automation

Automated testing will not be performed.

5. ROLES AND RESPONSIBILITIES

Role	Staff Member	Responsibilities
Project Manager		1. Acts as a primary contact for development and QA team. 2. Responsible for Project schedule and the overall success of the project.
QA Lead		1. Participation in the project plan creation/update process. 2.Planning and organization of test process for the release. 3.Coordinate with QA analysts/engineers on any issues/problems encountered during testing. 4.Report progress on work assignments to the PM
QA		1. Understand requirements 2. Writing and executing Test cases 3. Preparing RTM 4. Reviewing Test cases, RTM 5. Defect reporting and tracking 6. Retesting and regression testing 7. Bug Review meeting 8. Preparation of Test Data 9. Coordinate with QA Lead for any issues or problems encountered during test preparation/execution/defect handling.

6. ENTRY AND EXIT CRITERIA

6.1 Entry Criteria

All test hardware platforms must be successfully installed, configured, and functioning properly.

All necessary documentation, design documentation, and requirements information must be available to allow testers to operate the system and evaluate its correct behavior.

All standard software tools, including testing tools, must be successfully installed and functioning properly.

Correct test data available.

The test environment, such as hardware, software, and system administration support, must be ready.

QA resources fully understand the requirements, have deep knowledge of the functionality

Test scenarios, test cases and RTMs are revised

6.2 Exit Criteria

A certain level of requirements coverage has been achieved.

No high priority or severe bugs are left outstanding.

All high-risk areas have been fully tested, with only minor residual risks left outstanding.

Cost – when the budget has been spent.

The schedule has been achieved

7 SUSPENSION CRITERIA AND RESUMPTION REQUIREMENTS

7.1 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 are not available when needed by test team.

7.2 Resumption criteria

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

8. TEST STRATEGY

8.1 QA role in test process

The role of the quality assurance (QA) professional in the testing process is critical to ensuring the quality of the software being developed. Here are the main responsibilities of a QA specialist in the testing process:

- Test Planning and Strategy: The QA Specialist is responsible for planning and developing the overall test strategy, test plans and test cases for the project.

- Development of test cases: Creation of detailed test cases based on requirements and project documentation to verify software functionality.

- Implementation: Performing manual testing based on test cases to identify defects and ensure compliance of the software with quality standards.

- Defect Reporting: Documenting and reporting defects found during testing, including clear steps to reproduce the problem and detailed information on its impact.

- Regression Testing: Performing regression testing to ensure that new changes to the code have not introduced new defects and that existing features continue to work as they should.

- Collaboration: Work closely with developers, product managers, and other stakeholders to ensure a deep understanding of requirements and facilitate effective

communication throughout the testing process.

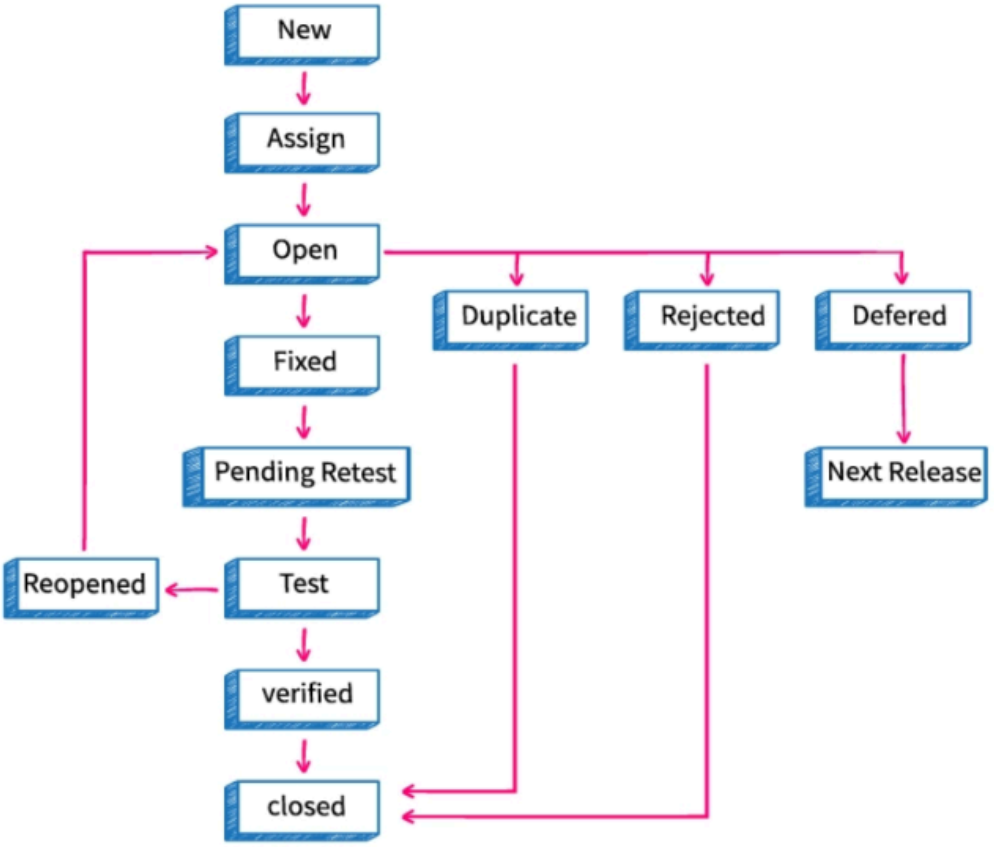
- Continuous Improvement: Proactively seeks opportunities to improve testing processes, tools and methods to increase the efficiency and effectiveness of the quality control function.

- Deployment/Delivery: After all bugs/defects reported after full testing are fixed and no other bugs are found, the report will be deployed to the customer's test site by the project manager.

In general, the QA professional plays a critical role in maintaining the quality standards of the software product and ensuring that it meets the needs and expectations of the end users.

8.2 Bug life cycle:

1. New: The bug was reported by a tester and is in the early stages.
2. Assigned: The bug has been assigned to the developer for further investigation and resolution.
3. In progress: The developer is actively working on fixing the bug.
4. Fixed: The bug has been fixed by the developer.
5. Verified: The tester checks that the defect is fixed and closes it if it is fixed.
6. Reopened: If the defect reappears or is not properly resolved, it is reopened.
7. Closed: The bug is closed after checking and confirming that it has been fixed correctly.



8.3 Testing types

Black box testing:

It is some time called behavioral testing or Partition testing. This kind of testing focuses on the functional requirements of the software. It enables one to derive sets of input conditions that that will fully exercise all functional requirements for a program. GUI Testing:

GUI testing will includes 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 are to provide correctness, reliability, testability and accuracy of the report output/data.

System Testing:

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

Performance Testing:

- Check the optimal time the page is loaded
- Check the operation of the system under load

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 at the Client.

Alpha testing:

The alpha test is conducted at the developer's site by client.

8.4 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 QA 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 QA 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

Severity ID	Level	Severity Description
1	Critical	The module/product crashes or the bug causes non recoverable conditions. System crashes, GP Faults, or database or file corruption, or potential data loss, program hangs requiring reboot are all examples of a 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	Medium	Incorrect functionality of component or process. There is a simple work around for the bug if it is Sev. 3.
4	Minor	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 does not delay shipping date, then fix it.
4	Low Priority	It is not important (at this time) that these bugs be addressed. Fix these bugs after all other bugs have been fixed. Enhancements/ Good to have features incorporated just are out of the current scope.

9. RESOURCE AND ENVIRONMENT NEEDS

9.1 Testing Tools

Process	Tool
Test case creation	Microsoft Excel
Test case tracking	Microsoft Excel
Test case execution	Manual
Test case management	Microsoft Excel
Defect management	Microsoft Word
Test reporting	PDF
Check list creating	Microsoft Excel
Project structure	Mind Map (.xmind)

9.2 Configuration Management

Documents CM: SVN

Code CM: Git

9.3 Test Environment

Operating System Android 4.2.2 and higher versions

Operating System IOS 4.0 and higher versions

10. TEST SCHEDULE

Task Name	Start	Finish	Effort	Comments
Test Planning	6.05	7.05		Support!
Review Requirements documents	7.05	8.05		
Create test basis	12.05	30.05		
Staff and train new test resources	-	-		
First deploy to QA test environment	2.06			
Functional testing – Iteration 1	3.06	6.06		
Iteration 2 deploy to QA test environment	6.06	8.06		
Functional testing – Iteration 2	8.06	16.06		
System testing				
Regression testing				
UAT				
Resolution of final defects and final build testing				
Deploy to Staging environment				
Performance testing				
Release to Production				

APPROVALS:

	Project Manager	QA Lead
Name		
Signature		

TERMS/ACRONYMS

The below terms are used as examples, please add/remove any terms relevant to the document.

TERM/ACRONYM	DEFINITION
PM	Project manager
QA	Quality Assurance
RTM	Requirements Traceability Matrix