Sample Test Plan

Table of Contents

Project “PetMedia”

Document Revision History

Date	Version	Description	Author	Reviewer	Approver
19.07.2025	0.1	Test plan was created	D. Pyrohova		

1. General information

The Startup team wants to create a unique cat images sharing app, which has passed the full cycle of manual testing. Given the specificity of the app it is very important to maintain consistent app quality.

The Test Plan has been created to facilitate communication within the team members. It is meant to be used as a manual during testing activities. It describes the procedure of the testing process. The test plan is intended for the Project Lead, Product Developers (back-end, Android and iOS), UI/UX Designer and QA Engineers.

This document describes approaches and methodologies that apply to the unit, integration and system testing of the app. It includes the objectives, test responsibilities, entry and exit criteria, scope, major milestones and approach.

The objective of the testing activities is to check functions and features of the software product developed for modern Android and iOS devices.

This test plan describes a hypothetical mobile application created for learning purposes.

Practical testing activities were performed using public demo web applications to validate individual features and testing techniques.

2. Scope

Testing of the mobile application is in the scope of this test plan. The following components and functions that will be tested:

- Create an account using email;
- Create an account using social media;
- Log in with email and password;

- Recover password;
- Upload images with cats;
- Upload videos with cats;
- Send messages;
- Add items to the favorites list;
- Find device contacts in the app;
- Start chats;
- Create group chats;
- Save and edit the user profile;
- Share messages to other social media platforms;
- Push notifications;
- Graphical User Interface (GUI).

3. Objectives

The main objective of testing is to ensure that the application meets all defined functional and non-functional requirements, and no significant errors appear in the application. Testing will confirm that all use case scenarios are executed correctly and meet user expectations. All identified issues and risks will be documented, communicated, and addressed appropriately before release. Any changes to requirements or design will be tested within available time and resources to maintain product quality. The goal is to deliver a stable, secure, and user-friendly application that aligns with the stakeholder expectations.

4. Approach

The approach that is used is analytical and follows a requirements-based strategy, where analysis of the requirements specification forms the basis for test planning, estimation, and test design. Test cases will also be created during exploratory testing.

The project follows an Agile methodology, with iterations every two weeks. At the end of each sprint, the requirements identified for that iteration will be delivered to the team and will be tested.

5. Roles

Product Lead	1 person	- Defines product vision and priorities; - Approves test plan and acceptance criteria; - Coordinates the team and makes key decisions.

Back-end Developer	1 person	- Develops server-side logic; - Supports APIs for mobile apps; - Collaborates with QA to test back-end components.
Android Developer	1 person	- Develops the Android app; - Implements unit tests; - Collaborates with QA to test mobile interface.
iOS Developer	1 person	- Develops the iOS app; - Implements unit tests; - Collaborates with QA to test mobile interface.
QA Engineers	2 people	- Create and execute test cases; - Perform exploratory and regression testing; - Report bugs and work closely with developers; - Coordinate with Product Lead on product quality; - One of the QA engineers is responsible for approving the test plan.
UX/UI Designer	1 person	- Designs prototypes; - Tests usability; - Collaborates with developers and QA to improve UI/UX.

6. Entry and exit criteria

6.1 Entry criteria

- All test platforms (Android, iOS, server) are installed, configured, and working stably;
- All the necessary documentation is available: requirements, design, technical specifications to understand the functionality of the application;
- All necessary testing tools are installed and working;
- Test data for various scenarios (photo/video upload, posting, comments, privacy settings, adding to the favorite list) are prepared;

- The test environment is ready: including test devices (Android/iOS smartphones), server infrastructure, and network access;
- 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

- A certain level of requirements coverage has been achieved;
- No critical or high-priority bugs left outstanding;
- All critical and high risk features are tested; only minor or low risk bugs remain;
- Testing is completed within the allocated budget and time;
- The release is ready for launch with confirmed quality.

7. QA role in test process

- Understanding of requirements will be done by QA;
- QA will be preparing test cases based on the exploratory testing. This will cover all
- scenarios for requirements;
- QA will be preparing a test matrix which maps test cases to respective requirements. This will ensure the coverage for requirements;
- Reviewing test cases and matrix;
- QA will be conducting peer review for test cases and test matrix;
- Test data will be created by respective QA in the test environment based on real usage scenarios;
- Test cases will be executed by respective QA in the test environment based on real usage scenarios;
- QA will log the bugs in Jira, found during execution of test cases. After this, QA will inform the respective developer about the bugs;
- Retesting for fixed bugs will be done by respective QA once it is resolved by the respective developer and bugs status will be updated accordingly. In certain cases, regression testing will be done if required;
- After all identified bugs are fixed and no new bugs are found, the PM will deploy the build to the test environment.

8. The main testing types that would be executed

- Functional Testing
- UI Testing
- Usability Testing
- Compatibility Testing (2 modern devices on Android and iOS)

- Regression testing
- Security testing

9. Bug Reports

Bug reports will be created in order to provide the development team and the project lead with exhaustive information about the discovered defects. They must be helpful in determining causes of the errors and correcting them.

The information that will be indicated in each bug report:

- Summary;
- Description;
  Priority (Highest / High / Medium / Low / Lowest);
- Saverity (Blocker / Critical / Major / Minor / Trivial);
- Status;
- Assignee;
- Reporter;
- Steps to Reproduce;
- Expected Result;
- Actual Result;
- Environment;
- Frequency of the defect occurrence;
- Attachments.

10. Bug life cycle

The bug lifecycle will include several key stages that ensure that defects are effectively identified, processed, and resolved:

- Unconfirmed - a new bug is reported. At this stage, the bug is not yet confirmed;
- New - the bug has been confirmed but not yet assigned to a developer;
- Assigned - the bug is being worked on by the developer;
- Resolved - after the work is completed. The bug is marked as resolved with one of the possible resolutions: fixed, duplicate, wontfix, worksforme, invalid, remind, later;
- Verified - QA confirms that the solution is working and the problem has been resolved;
- Reopen - QA has not confirmed the solution or the issue has re-emerged. The bug is reassigned to the developer and goes through the cycle again;
- Closed - the bug is finally resolved and closed.

11. Resource and environment needs

11.1 Testing Tools:

Test case creation	TestRail
Test case tracking	TestRail
Test case execution	TestRail
Test case management	TestRail
Defect management	Jira
Test reporting	TestRail
Check list creating	Google Sheets

11.2 Test Environment (devices):

iOS 16 and above;

Android 13 and above.

12. Approval

	Project Lead	Senior QA Engineer
Name		
Siganute		

13. Terms/Acronym definition

API	Application Program Interface
GUI	Graphical user interface
PM	Project manager
QA	Quality Assurance
RTM	Requirements Treceability Matrix

UI	User Interface
UX	User Experience