**An app for sharing cat photos**

**Test Strategy**

**Revision History**

| Date | Version | Author | Description |
| --- | --- | --- | --- |
| 30.04.2023 | 1.3.3.3 | Serhii Zombra | - |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

[**1. Scope 3**](#_gjdgxs)

[**2. Test Approach 4**](#_30j0zll)

[**3. Test Environment**](#_1fob9te) **7**

[**4. Testing Tools**](#_3znysh7) **9**

[**5. Release Control**](#_2et92p0) **10**

[**6. Risk Analysis**](#_tyjcwt) **11**

[**7. Review and Approvals**](#_3dy6vkm) **13**

# Scope

Reviewed by: Oksana Revvo (QA Team Lead)  
Approved by: Oksana Revvo (QA Team Lead)

# Test Approach

It defines

* Process of testing

The testing process will follow the software development lifecycle (SDLC) and will include the following phases: requirement analysis, test planning, test design, test execution, and test closure.

* Testing levels

The application will be tested at different levels: unit testing, integration testing, system testing, and acceptance testing.

* Roles and responsibilities of each team member

The testing team will consist of Test Manager, Test Lead, Test Engineers, and Automation Engineers (if applicable). The roles and responsibilities of each team member will be defined as follows:

Test Manager: responsible for overall testing activities and managing the testing team.

Test Lead: responsible for test planning, test design, test execution, and defect management.

Test Engineers: responsible for executing the test cases, reporting the defects, and retesting the defects.

Automation Engineers: responsible for developing and executing the automated test scripts (if applicable).

* Types of Testing ( Load testing, Security testing, Performance testing etc.)

The types of testing that will be performed are:

Functional Testing: to ensure that the application meets the functional requirements.

Regression Testing: to ensure that the new changes have not affected the existing functionalities.

Usability Testing: to ensure that the application is user-friendly and easy to use.

Performance Testing: to ensure that the application can handle a large number of users and requests.

Security Testing: to ensure that the application is secure and protected from unauthorized access.

Compatibility Testing: to ensure that the application is compatible with different devices, browsers, and operating systems.

Load Testing: to ensure that the application can handle the expected load.

* Testing approach & automation tool if applicable

The testing approach will be a combination of manual and automation testing. The automation tool that will be used is Selenium WebDriver with Java.

* Adding new defects, re-testing, Defect triage, Regression Testing and test sign off

Defect Management:

Defects will be reported using a defect tracking tool (JIRA). Defects will be triaged based on their severity and priority. The defects will be retested and closed once they are fixed.

Regression Testing:

Regression testing will be performed after each new build to ensure that the new changes have not affected the existing functionalities.

Test Sign-Off:

The testing team will sign off on the testing activities once all the test cases have been executed and all the defects have been closed.

Continuous Testing:

Continuous testing will be implemented to ensure that the application is always ready for release. This will involve running automated tests on a regular basis and integrating testing into the development process.

Testing Environment:

The testing environment will be a separate environment from the production environment. It will include a test database, test server, and test data.

Test Data Management:

Test data will be managed separately from the production data. Test data will be created, maintained, and deleted by the testing team.

Test Reporting:

Test results will be reported to the development team and the stakeholders. The test report will include the test execution status, test coverage, defect status, and recommendations for improvement.

Test Exit Criteria:

The testing team will define the exit criteria for each testing level. The exit criteria will include the number of defects, test coverage, and test execution status.

Test Training:

The testing team will receive training on the testing approach, test tools, and techniques. The team will also receive training on the application domain and the business requirements.

# Test Environment

To ensure thorough testing of the application, it is important to have different test environments set up. This section of the test strategy document outlines the various environments required for testing and the setup needed for each environment.

1. Development environment:

This environment is used by developers to code and test their changes. It requires the following setup:

Integrated development environment (IDE)

Code repository such as Git or SVN

Debugging tools

Application server

1. Test environment:

This environment is used by the testing team to conduct functional and non-functional testing of the application. It requires the following setup:

Application server

Database server

Test data

Test scripts

Test management tool

1. Staging environment:

This environment is used to simulate the production environment before releasing the application to production. It requires the following setup:

Application server

Database server

Test data

Test scripts

Load balancer

Monitoring tools

1. Production environment:

This environment is used by end-users to access the application. It requires the following setup:

Application server

Database server

Load balancer

Monitoring tools

Backup and Restore Strategy:

To ensure that the test data is protected and can be restored in case of any issues, it is important to have a backup and restore strategy in place. The strategy should include:

Regular backups of the test data

Off-site storage of backup data

Automated backup process

Restoration testing to ensure data can be recovered successfully

Having a well-defined test environment and backup and restore strategy helps in ensuring that the application is tested thoroughly and any issues can be resolved quickly.

# Testing Tools

Automation and Test Management tools are essential for efficient and effective testing. Some of the commonly used automation tools for testing mobile applications include Appium, Selenium, and Calabash. Test Management tools like Jira, HP ALM, and TestRail can be used for test case management, test execution, and reporting.

For the current project, we will need to evaluate the available testing tools, both open-source and commercial, to determine the best fit for our requirements. We will consider the following factors while choosing the tools:

1. Compatibility with our test environment
2. Ease of use and learning curve
3. Availability of features required for our testing needs
4. Cost and licensing requirements
5. Scalability and support for multiple users

Once the tools are selected, we will ensure that they are properly installed and configured for our testing environment. We will also establish guidelines for their usage, including version control and backup procedures.

We will also evaluate the feasibility of using any additional tools or utilities that can aid in testing, such as load testing tools, network sniffers, or emulators/simulators. If required, we will procure and integrate these tools into our testing framework.

Finally, we will define a clear process for the usage of testing tools and provide training to the testing team members to ensure they are proficient in using the selected tools.

# Release Control

This section outlines the release management plan for the application. It includes the following:

1. Release strategy: Define the strategy for releasing new features or changes to the application. This can include release cadence, release criteria, and release approval process.
2. Version control: Implement a version control system to manage the source code and other related artifacts.
3. Build management: Define the build process and tools used to create builds for each release.
4. Deployment strategy: Define the strategy for deploying the application to different environments, such as development, testing, staging, and production.
5. Rollback plan: Develop a plan for rolling back to a previous version in case of any issues during the release process.
6. Release notes: Document the release notes for each version, including the list of new features, enhancements, bug fixes, and known issues.
7. User acceptance testing: Define the process for user acceptance testing before the release, including the test cases, acceptance criteria, and testing environment.
8. Post-release testing: Define the process for post-release testing, including monitoring, logging, and error handling.
9. Backward compatibility: Ensure backward compatibility of the application with previous versions.
10. Release sign-off: Define the process for release sign-off, including the criteria and responsibilities of each team member involved.

By having a well-defined release control plan, the team can ensure that each release is tested thoroughly and deployed in a controlled manner, minimizing the risk of issues and ensuring a smooth user experience.

# Risk Analysis

Risk Management

Identify potential risks associated with the project and its impact on the testing process. Some of the risks that may be encountered during the testing of the photo-sharing app are:

1. Technical risks such as system failure, data loss, and poor application performance.
2. User experience risks such as poor interface design, navigation issues, and difficulty in using the application.
3. Security risks such as unauthorized access, data theft, and cyber attacks.
4. Compliance risks such as non-compliance with privacy and data protection regulations.

Mitigation Plan

To mitigate the risks, we will take the following actions:

1. Technical risks - We will perform rigorous testing at all stages of the development process, including functional, performance, and security testing. We will also ensure proper backup and disaster recovery mechanisms are in place.
2. User experience risks - We will conduct usability testing with a group of representative users to ensure that the app's interface design is user-friendly and easy to navigate. We will also incorporate feedback from user testing into the app's design and development.
3. Security risks - We will perform comprehensive security testing and penetration testing to identify and fix vulnerabilities in the app. We will also ensure that the app adheres to industry standards and best practices for data security and privacy.
4. Compliance risks - We will ensure that the app complies with all applicable laws and regulations, including GDPR, CCPA, and other data protection laws. We will also ensure that user data is collected and used in accordance with our privacy policy.

Contingency Plan

In case any of the identified risks materialize, we have a contingency plan in place to minimize the impact. The contingency plan includes:

1. Technical risks - We will have a disaster recovery plan in place to restore the app to its previous state in case of system failure or data loss.
2. User experience risks - We will immediately address any issues identified during usability testing or user feedback and take necessary steps to improve the user experience.
3. Security risks - We will have a response plan in place to handle any security breaches or attacks, including notifying affected users, fixing the vulnerability, and enhancing security measures.
4. Compliance risks - We will have a plan in place to address any non-compliance issues identified, including modifying the app to ensure compliance with data protection laws and regulations.

# Review and Approvals

Documentation and sign-off are important aspects of the testing strategy. In this section, we will outline the review and approval process for the testing strategy document.

The testing strategy document will be reviewed by the business team, project management, development team, and other stakeholders as necessary. Each reviewer will provide feedback and suggestions for improvement based on their area of expertise. The testing team will review the feedback and incorporate the changes into the document as appropriate.

Once the document has been updated with all feedback, it will be presented for sign-off by the business team, project management, and development team. The sign-off process will include a review of the summary of changes made to the document, along with the approved date, name, and comment.

This process ensures that all stakeholders have reviewed and agreed upon the testing strategy document, and any changes made have been documented and approved.