Test strategy template

"Crazy Cats" is a mobile application for iOS and Android devices. The startup has a team of three developers (back-end, iOS, and Android).

The app is created for those who love cats and want to share photos of their pets with other cat enthusiasts. Built using Lean Software Development methodology, this app allows users to upload photos of their cats, browse photos from other users, create communities, and comment on photos.

It not only entertains users but also creates new opportunities for social interaction and support. Additionally, viewing funny and beautiful cat photos can boost mood and even reduce stress during these challenging times.

Revision History

Date	Version	Author	Description
2024-05-23	1.09	Prorochenko J	

1. Scope

The document will be reviewed by the development team and approved by the project manager.

Testing activities will be conducted throughout the entire development process with specific steps and timelines.

2. Test Approach

Testing Process: Testing will adhere to the Lean Software Development methodology, emphasizing efficiency and optimization.

Testing Levels:

- Unit Testing
- Integration Testing
- System Testing
- Acceptance Testing

Roles and Responsibilities:

- The testing team will be responsible for executing all levels of testing.
- Each developer is accountable for the quality of their code and its testing.

Types of Testing:

- Functional Testing
- Usability Testing
- Compatibility Testing
- Performance Testing
- Security Testing

Automation Tools:

- For automated testing of the mobile application, tools such as Appium or XCTest (for iOS) and Espresso or UI Automator (for Android) will be used.

Defect Management:

- Defects will be tracked and managed in JIRA

3. Test Environment

Development Environment:

- Each developer will have their own local development environment with individual configurations suited to their needs.

Testing Environment:

- Dedicated servers will be allocated for conducting integration, system, and acceptance testing.

Production Environment:

- Live servers will host the final testing before the release of new software versions.

4. Testing Tools

Automation Tools:

- Depending on the platform, automation will utilize:
  - Appium for cross-platform mobile testing.
  - XCTest for iOS-specific testing.
  - Espresso for Android-specific testing.

Test Management Tools:

- JIRA will be used for test management, including tracking and managing test cases, execution, and defects.

5. Release Control

Each release will be managed through version control system Git.

All test scenarios must be successfully executed before releasing a new version of the software.

6. Risk Analysis

Risks: potential risks include data security issues, software compatibility across different platforms, and system instability.

Risk Mitigation: active testing on various devices and in different environments will help identify compatibility and stability issues. Security audits and regular updates to security measures will also be conducted to minimize risks. In case of critical issues, rapid response and recovery procedures are in place.

7. Review and Approvals

The document will be reviewed and approved by all project participants, including developers and managers.
All changes and comments will be documented for future reference.