“KittySnap”

Test Strategy

Revision History

Date	Version	Author
18.03.2023	0.1	Matsyhin M.

Table of Contents

1. Scope

The scope of testing will cover the following areas:

- User interface and usability testing
- Functional testing
- Performance testing
- Security testing
- Compatibility testing
- Regression testing
- Acceptance testing

The Test Strategy document will be reviewed by the project manager, development team, and testing team. They will ensure that the document covers all the necessary aspects of testing and meets the project's requirements.

The Test Strategy document will be approved by the project manager, who will ensure that it aligns with the project's goals and objectives.

The testing activities for "KittySnap" will include the following:

- User interface and usability testing: to ensure that the application is user-friendly and easy to navigate. This will be conducted throughout the development cycle.
- Functional testing: to verify that the application functions correctly as per the requirements. This will be conducted during the functional testing phase, which will take place after the development of each module.
- Performance testing: to evaluate the application's performance under different load conditions. This will be conducted during the performance testing phase, which will take place after the functional testing phase.
- Security testing: to identify vulnerabilities and ensure that the application is secure. This will be conducted during the security testing phase, which will take place after the performance testing phase.
- Compatibility testing: to verify that the application works correctly on different devices, browsers, and platforms. This will be conducted during the compatibility testing phase, which will take place after the security testing phase.
- Regression testing: to ensure that new changes or updates do not affect the existing functionality. This will be conducted throughout the development cycle and after each testing phase.
- Acceptance testing: to ensure that the application meets the client's requirements and expectations. This will be conducted during the acceptance testing phase, which will take place after the completion of all other testing phases.

The timelines for each testing phase will depend on the project's size, complexity, and deadlines. The project manager and testing team will work together to create a testing plan with specific timelines for each testing phase. The plan will be reviewed and approved by the project manager and development team before the testing phase begins.

2. Test Approach

The testing approach for "KittySnap" will be a combination of manual and automated testing. Manual testing will be used for UI testing, functional testing, and acceptance testing, while automated testing will be used for performance testing, security testing, and regression testing.

Process of testing:

- The testing process for "KittySnap" will follow a structured approach, which includes the following steps:
    - Test planning: where the testing plan and test cases are created.
    - Test design: where the test cases are designed based on the requirements and functional specifications.
    - Test execution: where the test cases are executed, and defects are logged.
    - Defect management: where the defects are reported, tracked, and resolved.
    - Test closure: where the test results are analyzed, and the test report is generated.

Testing levels:

- The testing levels for "KittySnap" will include the following:
    - Unit testing: to verify the functionality of individual components or modules.
    - Integration testing: to verify the functionality of different modules when integrated.
    - System testing: to verify the functionality of the entire system as per the requirements.
    - Acceptance testing: to verify that the application meets the client's requirements and expectations.

The roles and responsibilities of each team member in the testing process will be defined as follows:

- Project manager: responsible for overall project management, including testing.
- Test manager: responsible for planning, designing, and executing testing activities.
- Test lead: responsible for coordinating testing activities, assigning tasks, and tracking progress.
- Test engineer: responsible for designing and executing test cases and reporting defects.
- Developer: responsible for fixing defects identified during testing.
- Business analyst: responsible for defining requirements and functional specifications.

The types of testing for "KittySnap" will include the following:

- User interface and usability testing: to ensure that the application is user-friendly and easy to navigate.

- Functional testing: to verify that the application functions correctly as per the requirements.
- Performance testing: to evaluate the application's performance under different load conditions.
- Security testing: to identify vulnerabilities and ensure that the application is secure.
- Compatibility testing: to verify that the application works correctly on different devices, browsers, and platforms.
- Regression testing: to ensure that new changes or updates do not affect the existing functionality.
- Acceptance testing: to ensure that the application meets the client's requirements and expectations.

Testing approach & automation tool if applicable:

- The testing approach for "KittySnap" will be a combination of manual and automated testing. The automation tool used will be selected based on the project's requirements and budget.

The process for adding new defects, re-testing, defect triage, regression testing, and test sign-off will be defined as follows:

- Defects will be logged using a defect tracking tool.
- Defects will be prioritized based on their severity and impact on the application.
- Defects will be assigned to the relevant team member for fixing.
- Once fixed, the defects will be retested to ensure that they have been resolved.
- Regression testing will be performed after each testing phase to ensure that new changes or updates do not affect the existing functionality.
- Test sign-off will be given by the test manager once all testing activities are completed, and the application meets the client's requirements and expectations.

3. Test Environment

For "KittySnap", the following environments will be required:

- Development environment: for development and unit testing.
- Integration environment: for integration testing.
- Test environment: for system testing and acceptance testing.
- Production environment: for the live application.

Each environment will have different setup requirements, as follows:

- Development environment: this will require a development machine or virtual machine with the necessary software, tools, and libraries.
- Integration environment: this will require servers, databases, and other systems to be integrated with the application.
- Test environment: this will require servers, databases, and other systems to run the application and support testing activities.
- Production environment: this will require servers, databases, and other systems to support the live application.

To ensure the safety and integrity of test data, the following backup and restore strategy will be implemented:

- Regular backups of the test data will be taken to prevent data loss in the event of a system failure or other unexpected events.
- The backup data will be stored on a separate server or storage device to prevent data loss in the event of a disaster or hardware failure.
- The backup data will be encrypted and password-protected to ensure data security.
- The restore strategy will be defined in the event of a data loss or system failure. This will include restoring the backup data to the relevant environment and verifying the data integrity.

4. Testing Tools

The following testing tools will be used:

1. Test management tool:
- to manage test cases, test plans, and test execution
2. Automated testing tools:
- for performance testing, security testing, and regression testing
3. Bug tracking tool:
- to log and track defects

For "KittySnap", the following tools will be required:

- Test management tool: to manage test cases, test plans, test runs, and test results. Examples of test management tools include Jira, TestRail, and HP ALM.
- Automation testing tool: to automate the execution of test cases. Examples of automation testing tools include Selenium, Appium, and TestComplete.
- Performance testing tool: to simulate real-world user load and measure application performance. Examples of performance testing tools include Apache JMeter, LoadRunner, and Gatling.
- Security testing tool: to identify and fix security vulnerabilities. Examples of security testing tools include OWASP ZAP, Acunetix, and Burp Suite.

The number of tools required will depend on the testing requirements and the budget. The selection of tools will be based on the following criteria:

- The tool should meet the testing requirements and support the required testing types.
- The tool should be cost-effective and fit within the budget.
- The tool should be easy to use and support collaboration among team members.
- The tool should be scalable and support the required number of users.

Once the tools are selected, the number of users that can be supported will be determined, and the licensing and installation requirements will be planned accordingly. Open-source tools will be preferred wherever possible to keep the costs low, but commercial tools will be used if they offer significant advantages over the open-source options.

5. Release Control

The release management plan will define the process for managing the release of new versions of the application. The plan will include the following:

- A version history that tracks all changes made to the application and their respective versions.
- A schedule for release cycles, including dates for feature freeze, code freeze, and release.
- A process for testing each release, including a list of test cases to be executed.
- A process for identifying and managing defects found during testing.
- A process for deploying each release to production.

The version history will be maintained in a version control system like Git, which will track changes made to the codebase. Each change made to the codebase will be associated with a specific version number, which will be used to track the progress of testing for that version.

Before each release, the test team will execute a set of test cases to ensure that all modifications in that release are tested thoroughly. The results of the test execution will be recorded in a test management tool, and defects found during testing will be logged and prioritized based on severity.

Once testing is complete and all defects are resolved, the release will be deployed to production. The deployment process will be automated wherever possible to minimize errors and ensure consistency across environments. A rollback plan will also be in place in case any issues arise during deployment.

The release management plan will be reviewed and updated regularly to ensure that it remains effective and up-to-date with the changing needs of the application and the business.

6. Risk Analysis

The risk analysis will involve the identification and assessment of potential risks that could impact the testing process and the overall quality of the application. Risks can come in various forms, such as technical, process-related, or business-related. Examples of potential risks that could impact "KittySnap" application are:

1. Technical risks:
    - Compatibility issues with different devices, operating systems, and web browsers
    - Network connectivity issues, leading to slow response times or errors
    - Security vulnerabilities that could compromise user data
    - Poor performance under high load
2. Process-related risks:
    - Inadequate test coverage due to limited resources or time constraints
    - Insufficient documentation or unclear requirements, leading to misunderstandings and misinterpretations
    - Lack of communication or collaboration between teams, leading to misalignments and delays
3. Business-related risks:
    - Changes in market demand or user behavior that could impact the application's relevance and popularity
    - Legal or regulatory changes that could impact the application's compliance and liability
    - Competitors launching similar or superior products that could impact the application's market share and revenue potential

This prioritization will help the team focus their efforts on the most critical risks and allocate resources accordingly. Mitigation strategies will be developed for each risk, including contingency plans and mitigation actions. The risks and their mitigation strategies will be documented and reviewed regularly to ensure that they remain relevant and up-to-date with the evolving project needs.

7. Review and Approvals

The Test Strategy document will be reviewed by various stakeholders, including the business team, project management, development team, testing team, and any other relevant parties. The purpose of the review is to ensure that the document is accurate, complete, and aligned with the project objectives and requirements.

Each reviewer will provide feedback and comments on the document, which will be addressed by the testing team. The testing team will revise the document based on the feedback received and ensure that all comments are addressed satisfactorily. The revised document will be shared with the reviewers for final approval.

The final approval will be given by the project sponsor or any other designated authority, who will ensure that the document is aligned with the project goals and objectives. The approved document will be stored in a central repository, and any changes made to it will be documented and tracked.

A summary of all review changes will be provided at the beginning of the document, along with the approved date, name, and comment. This summary will help stakeholders understand the changes made to the document and provide clarity on the version history of the Test Strategy.

Overall, the review and approval process ensures that the Test Strategy document is accurate, complete, and aligned with the project objectives and requirements. It also ensures that all stakeholders have a clear understanding of the testing approach, scope, environment, tools, and release control for the "KittySnap" application.