**ASTANA IT UNIVERSITY**

**SOFTWARE QUALITY ASSURANCE**

**TEST PLANS FOR CHAT APP TESTING**

**Project Name:** Chat App Testing

**Test Plan ID:**  CAT

**Version:**  1.0

**Date Prepared:** 12.06.2023

**Last Updated By:** Beibaryssultan Aitbay

**Last Updated Date:** 14.06.2023

**Test Environment:** Android Studio, Appium, WebdriverIO

**Testing Approach:** Automated testing

**Test Coverage:** Account, Service, Security

**Prepared By:** Aruzhan Amangeldi, Beibaryssultan Aitbay

**Reviewed By:** Tleubayeva A.O.

**Link from Roadmap:** [link](https://alie164.atlassian.net/jira/software/c/projects/CAT/boards/5/roadmap?shared=&atlOrigin=eyJpIjoiNTlkODBkMzZhY2QxNDM4ODlkYzRkZTM3MDdmNzRjNmQiLCJwIjoiaiJ9)

**Link to the Github repository:** [link](https://github.com/whyne164/chatapp_sqa)

**Astana, 2023**

**Objective:**

The primary objective of this test plan for the Chat app is to validate its functionality, integrity, performance, and security. The specific objectives are as follows:

1. Functionality: Validate that the Chat app performs the intended functions, including sending and receiving text and multimedia messages, supporting one-on-one and group conversations, and enabling audio and video calls. Test different user interactions, such as navigation, input validation, and error handling.
2. Integrity and Accuracy: Ensure data integrity and accuracy within the Chat app. Validate that messages are delivered accurately and in the correct order. Verify that data processing operations, such as message retrieval and deletion, maintain the integrity of the chat history.
3. Security: Perform security testing to identify vulnerabilities and ensure the secure storage and transmission of user data. Test the authentication and authorization mechanisms of the Chat app. Verify that sensitive user information is properly protected and that secure communication protocols are used.
4. Compatibility: Test the Chat app on various devices, screen sizes, and resolutions to ensure compatibility across different Android platforms. Verify the app's compatibility with different versions of the Android operating system. Test the app's responsiveness and adaptability to different orientations and display sizes.
5. Error Handling and Recovery: Evaluate the Chat app's error handling and recovery mechanisms. Test scenarios where the app encounters unexpected errors or crashes. Verify that appropriate error messages are displayed to the user, and the app recovers gracefully from errors.

The test plan will utilize Android Studio, Appium, and WebdriverIO as testing tools in the specified test environment. Well-defined test cases with clear steps and expected results will be documented. Detailed test execution reports and defect reports will be generated. A specified test schedule will be established, indicating the start and end dates of the testing process. Mitigation strategies will be implemented to address potential risks and challenges in the testing process.

**Scope**

The testing scope encompasses different aspects of the CometChat Kitchen Sink Sample App. It involves verifying the functionality of one-on-one and group messaging, ensuring that messages are transmitted accurately and efficiently. Additionally, testing focuses on the app's calling features, including audio and video calls, to ensure seamless communication. Furthermore, testing encompasses the proper handling and display of various multimedia formats, such as images, videos, and documents. The scope also extends to compatibility testing across different devices, localization and internationalization aspects, adherence to app store guidelines, and robust error handling and recovery mechanisms.

**User registration/login:** This test plan focuses on validating the functionality and security of the user registration and login process, ensuring seamless access to the app for users.

**UI/UX design and device-specific testing**: This test plan aims to evaluate the user interface and experience design of the app, ensuring consistency across different devices and optimizing the app's performance on specific device models.

**Messaging in-app**: This test plan verifies the messaging functionality of the app, including sending and receiving text messages, multimedia messages, and group messaging, ensuring smooth communication between users.

**Making voice calls:** This test plan focuses on testing the app's capability to make voice calls, assessing call connectivity, call quality, and features like call initiation, termination, and in-call functionality.

**Error handling:** This test plan aims to identify and validate the app's behavior when encountering unexpected errors or exceptions, ensuring that appropriate error messages are displayed, and the app recovers gracefully from errors.

**Localization testing:** This test plan verifies the app's compatibility with different languages, character sets.

**Test Case**

| **Test Case ID** | **Description** | **Status** |
| --- | --- | --- |
| CAT-1 | User login with provided account | Pass |
| CAT-3 | User login using UID | Pass |
| CAT-10 | Checking card component | Pass |
| CAT-11 | Checking conversation component | Pass |
| CAT-19 | Checking messages component | Pass |
| CAT-20 | Checking Users component | Pass |
| CAT-21 | Checking Groups component | Pass |
| CAT-22 | Checking Calls component | Passed with restrictions |
| CAT-23 | Checking Shared component | Pass |
| CAT-24 | Checking Logout button | Pass |
| CAT-25 | No Internet error handling | Pass |
| CAT-26 | Using the Russian in registration and messaging | Pass |
| CAT-27 | Checking adaptive design | Pass |
| CAT-28 | The UX/UI design | Pass |

**Test Schedule**

The test execution for the database testing will be conducted according to the following schedule:

| **Test Activity** | **Start Date** | **End Date** |
| --- | --- | --- |
| Test Case Creation | 2023-06-13 | 2023-06-14 |
| Test Environment Setup | 2023-06-13 | 2023-05-13 |
| Test Execution | 2023-05-13 | 2023-05-14 |
| Test Reporting and Documentation | 2023-06-13 | 2023-06-14 |

**Test Deliverables**

1. Test Plan Document:

A comprehensive document outlining the objectives, scope, approach, and schedule of the testing process for the Chat App. It serves as a guide for the testing team to follow.

2. Test Case Documents:

Detailed documents containing test cases that cover various scenarios to validate the different functionalities of the Chat App. Each test case includes step-by-step instructions, expected results, and any required test data.

3. Test Execution Reports:

Reports summarizing the execution of test cases, including the test case identification, description, pass or fail status, and any issues encountered during testing. These reports provide an overview of the testing progress and outcomes.

4. Defect Logs:

A log documenting all identified defects or issues discovered during testing. Each defect includes a description, severity level, steps to reproduce, and the assigned person responsible for resolving the defect.

5. Final Test Report:

A comprehensive report that summarizes the entire testing effort for the Chat App. It includes an overview of the testing activities, test coverage, test execution results, identified issues, recommendations for improvement, and a concluding statement.

**Test Dependencies**

Test Environment Setup: Outline the setup process for the test environment, which includes installing and configuring Android Studio with Appium and WebdriverIO. Specify the required settings, access credentials, and network configurations.

Test Execution Reports: Generate comprehensive reports that detail the test execution process, including the steps performed, actual results, expected results, and any identified defects or issues. Include relevant screenshots or logs to provide supporting evidence.

Defect Reports: Compile a separate report documenting any identified defects or issues, including a unique defect ID, a clear description of the problem, its severity level, step-by-step instructions to reproduce the issue, the observed actual results, and the expected results. Collaborate closely with the development team to track and resolve these issues efficiently.

Test Plan Updates: Continuously update the test plan to reflect any changes or additions resulting from evolving requirements for the Chat app. Ensure the document is well-organized, easily accessible, and properly versioned for future reference and maintain clear communication with the team regarding updates.