**ASTANA IT UNIVERSITY**

**SOFTWARE QUALITY ASSURANCE**

**TEST REQUIREMENTS FOR CHAT APP TESTING**

**Project Name:** Chat App Testing

**Test Plan ID:**  CAT-001

**Version:**  1.0

**Date Prepared:** 12.06.2023

**Last Updated By:** Aruzhan Amangeldi

**Last Updated Date:** 14.06.2023

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

**Testing Approach:** Automated testing

**Test Coverage:** Accessibility, Service

**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**

**Content**

**1 Introduction —————————————————————————— 3**

**1.1 Purpose —————————————————————————— 3**

**1.2 Scope —————————————————————————— 3**

**2 Test Requirements —————————————————————————— 3**

**3. Assumptions and Dependencies ———————————————————— 4**

**4. Testing Approach —————————————————————————— 4**

**4.1 Test Levels ——————————————————————— 4**

**4.2 Test Types ——————————————————————— 4**

**4.3 Test Techniques ——————————————————————— 5**

**4.4 Test Coverage ——————————————————————— 5**

**5. Test Environment ——————————————————————— 5**

**5.1 Hardware Requirements ————————————————— 5**

**5.2 Software Requirements ————————————————— 5**

**5.3 Test Data —————————————————————————— 6**

**Test Deliverables —————————————————————————— 6**

**Test Schedule —————————————————————————— 6**

**Change History —————————————————————————— 6**

**References —————————————————————————— 6**

**1. INTRODUCTION**

The CometChat Kitchen Sink Sample App is a feature-rich messaging application developed using the CometChat UIKit framework. It offers a comprehensive set of functionalities, including one-on-one and group messaging, as well as audio and video calling capabilities. Users can exchange text and multimedia messages, such as images, videos, and documents, making communication dynamic and engaging. The app caters to the needs of individuals and groups seeking seamless and efficient communication.

**1.1 Purpose**

The purpose of testing the CometChat Kitchen Sink Sample App is to ensure its functionality, reliability, and usability. Testing aims to identify and rectify any defects or issues that may hinder the smooth operation of the application. By thoroughly testing the app's various features, such as messaging, calling, and multimedia sharing, the goal is to deliver a high-quality user experience, safeguard data security, and meet user expectations.

**1.2 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.

**2. TEST REQUIREMENTS**

The specific test specifications for testing the Chat App are described in this section. Each requirement is given a special identification number and is specified along with the test cases, acceptance standards, and status that go with it. The table below provides a table for test requirements:

| **Requirement ID** | **Requirement Description** | **Test Cases Executed** | **Test Case Status** |
| --- | --- | --- | --- |
| CAT-REQ-001 | Users should be able to registrate or login using valid data | CAT-1, CAT-3, CAT-24, CAT-26 | Passed |
| CAT-REQ-002 | App should has adaptive, UX/UI design | CAT-10, CAT-20, CAT-27, CAT-28 | Passed |
| CAT-REQ-003 | Chat app should support one-on-one messaging | CAT-11, CAT-19, CAT-23, CAT-26 | Passed |
| CAT-REQ-004 | Chat app should support group messaging | CAT-19, CAT-21, CAT-23, CAT-26 | Passed |
| CAT-REQ-005 | Users should be able to make calls | CAT-22 | Passed with restrictions |
| CAT-REQ-006 | Chat app should be able to handle errors | CAT-25 | Passed |
| CAT-REQ-007 | App should support localization | CAT-26 | Passed |

**3. ASSUMPTIONS AND DEPENDENCIES**

This section aims to establish a clear understanding of the necessary conditions and requirements for successfully executing the tests in the context of the Chat app.

Assumptions:

- It is assumed that the Chat app has been correctly developed and configured according to the recommended specifications and guidelines provided.

- The testing team has been granted the required access privileges and permissions, including the ability to perform necessary actions such as user registration, login, messaging, and making voice calls.

Dependencies:

- Availability of the test environment, including the necessary hardware and software resources such as Android Studio, Appium, and WebdriverIO.

- Completion of any prerequisite tasks or activities before initiating the testing process, such as setting up the development and testing environment, installing dependencies, and configuring the necessary tools.

- Timely availability of any dependent components or systems that are required for testing, such as a stable internet connection and access to other devices for testing features like making voice calls.

**4. TESTING APPROACH**

**4.1 Test Levels:**

1. Unit Testing:

Validate the correctness and functionality of individual components within the Chat app, such as message sending and receiving functionalities.

2. Integration Testing:

Test the integration and interaction between different components of the Chat app, such as the messaging module and user authentication system.

3. Acceptance Testing:

Ensure that the Chat app meets the user's expectations, performs as intended, and delivers the required functionalities, such as sending messages, making calls, and managing contacts.

**4.2 Test Types:**

1. Functional Testing:

Verify that the Chat app behaves correctly and performs its intended functions, such as sending and receiving messages, making voice calls, and managing chat groups.

2. Security Testing:

Identify vulnerabilities and weaknesses in the Chat app's security measures, ensuring secure storage and transmission of user data, and validating user access controls.

3. Data Integrity Testing:

Ensure that the Chat app maintains data integrity, enforces data validation rules, and handles data consistency across different devices and platforms.

**4.3 Test Techniques:**

1. White-box Testing:

Inspect and test the internal structure and logic of the Chat app, including the code-level testing of message handling, call initiation, and user authentication functionalities.

2. Error Guessing:

Apply testing based on the tester's intuition and past experience to identify potential errors or issues, such as unexpected app crashes, network interruptions, or message delivery failures.

**4.4 Test Coverage:**

1. Data Integrity:

Validate the integrity of user data, ensuring the correct storage and retrieval of messages, user profiles, and chat history.

2. Data Retrieval:

Test the accuracy and performance of data retrieval operations, such as fetching chat messages, searching for specific messages, and loading chat history.

3. Security:

Verify the Chat app's security measures, including user authentication, encryption of sensitive data, protection against unauthorized access, and prevention of data breaches.

**5. TEST ENVIRONMENT**

**5.1.Hardware Requirements**

Processor: Dual-core or higher

RAM: 4 GB or higher

Storage: 8GB or higher

Network Interface: Ethernet or higher for network connectivity

**5.2.Software Requirements**

Operating System: Windows 10

Development Environment: Android Studio

Test Automation Framework: Appium

Web Automation Framework: WebdriverIO

Messaging Framework: CometChat UIKit

**5.3.Test Data**

1. Types of Data:

Customer information, product data, payment records, user roles

2. Data Generation Approach:

Test data can be manually created using queries and specific data values.

3. Data Privacy and Security Considerations:

Ensure that any sensitive data transmitted or stored in the testing environment is encrypted to prevent unauthorized access.

4. Data Volume and Variety:

Consider generating test data to assess the scalability of the database system.

**TEST SCHEDULE**

| **Test Description** | **Start date** | **Finish date** | **Assigned** |
| --- | --- | --- | --- |
| Functionality Testing | 13.05.2023 | 13.06.2023 | Beibaryssultan Aitbay |
| Design Testing | 13.05.2023 | 13.06.2023 | Aruzhan Amangeldi |
| Security Testing | 13.05.2023 | 13.06.2023 | Beibaryssultan Aitbay |

**CHANGE HISTORY**

| **Name and Surname** | **Updated date** | **Changes** |
| --- | --- | --- |
| Aruzhan Amangeldi | 12.06.2023 | Approved |
| Beibaryssultan Aitbay | 13.06.2023 | Approved |
| Aruzhan Amangeldi | 01.06.2023 | Approved |

**REFERENCES**

Android Studio - [link](https://developer.android.com/studio?gclid=CjwKCAjwp6CkBhB_EiwAlQVyxW1pH1_vrlbtnjUmi79x-UKs2dOo_5OROcQ0YWGYiq_GkG-uXn4GQBoC5RUQAvD_BwE&gclsrc=aw.ds)

Jira Software - <https://support.atlassian.com/jira-software-cloud/resources/>