



SOFTWARE REQUIREMENTS SPECIFICATION

CHATZEN



AUGUST 11, 2023

RUBIX!

CZ!
CHATZEN
Version 1.0

Team: RUBIX!
Shreya Singh (LCS2022046)
Raghunandan Bansal (LCS2022058)
Rishi Raj Maheshwari (LCB2022005)

Table of Contents

1. Introduction

1.1 Purpose

1.2 Scope

1.3 Definitions, Acronyms, and Abbreviations

1.4 References

1.5 Overview

2. The Overall Description

2.1 Product Perspective

2.2 Product Functions

2.3 User Characteristics

2.4 Constraints

2.5 Assumptions and Dependencies

3. External interface Requirements

3.1 User Interfaces

3.2 Hardware Interfaces

3.3 Software Interfaces

3.4 Communications Interfaces

4. System Features

5. Other Non-Functional Requirements

5.1 Performance Requirements

5.1.1 Speed

5.1.2 Scalability

5.1.3 Reliability

5.1.4 Security

5.1.5 Compatibility

5.1.6 Resource Usage

5.2 Software System Attributes

5.2.1 Compatibility

5.2.2 Performance

5.2.3 Maintainability

5.2.4 Accessibility

5.2.5 Usability

6. Other Requirements

- **Appendix A: Project Document**
- **Appendix B: Test Cases**

1. Introduction

There are numerous products available that allow us to chat over internet. ChatZen is a social-networking tool that leverages on technology advancement thereby allowing its users communicate and share media very privately and securely. It offers a wonderful experience for keeping in touch with your loved ones. It can be used for messaging, placing voice messages, making voice and video calls, share updates and photos, share locations, enhance local socializing.

1.1 Purpose

The purpose of this Software Requirement Specification (SRS) document is to provide a comprehensive description of the requirements for the development of a Kotlin Chat App. This document serves as a comprehensive reference for the design, development, and testing of the chat app.

1.2 Scope

The ChatZen is intended to provide users with a platform to engage in real-time text-based communication. Users will be able to send and receive messages, create chat rooms, and manage their contacts through user Ids. The app will be available for mobile platforms only due to security reasons.

1.3 Definitions, Acronyms, and Abbreviations

APP	Application
API	Application Programming Interface
Chat App	The application being developed as per this document
CZ	ChatZen
GC	Group Chat
PC	Private Chat
SRS	Software Requirements Specification.
UI	User Interface
VC	Video Chat

1.4 References

The references for the above software are as follows:-

- <https://www.google.com/>
- <https://www.wikipedia.org/>
- <https://www.youtube.com/>
- <https://kotlinlang.org/>
- <https://www.geeksforgeeks.org/>
- <https://www.mongodb.com/>
- <https://firebase.google.com/>
- https://www.canva.com/en_in/
- <https://www.mysql.com/>
- <https://code.visualstudio.com/>

1.5 Overview

The Kotlin Chat app is a mobile application designed to facilitate real-time communication between users. The app will allow users to send text messages, images, videos, and voice recordings to their contacts. The app will also support group chats, allowing users to chat with multiple contacts at the same time through user ids. The app will be developed using the Kotlin framework, which is a popular open-source mobile app development platform developed by Google. The app will be available for both Android and iOS devices, and it will be designed to provide a seamless and intuitive user experience. The following sections of this document will provide a detailed description of the requirements for the development of the ChatZen App.

2. The Overall Description

2.1 Product Perspective

The app will function as an independent software application, interacting with users through its user interfaces. It may also utilize third-party services for features such as authentication and notifications. The system to be developed here is an Chat facility. It is a centralized system. It is Client-Server system with centralized database server.

There is a two-way communication between different clients and server. This chat application can be used for group discussion. It allows users to find other logged in users.

2.2 Product Functions

The main functions of the app are:

- User registration and authentication
- Creating and joining chat rooms
- Sending and receiving messages
- Adding and managing contacts
- Video Calling
- Notification system for new messages
- Profile customization

2.3 User Characteristics

Casual chatters: These users want to chat socially with friends and family. They prefer frequent messaging with multimedia.

Student chatters: These are typically young app users who enjoy socializing and sharing funny photos/videos. Custom emojis and stickers appeal to them.

Professional chatters: These users need secure 1-on-1 and group chats for work. They require some collaboration features.

International chatters: These users chat with contacts in different countries. They need free messaging over data/WiFi.

2.4 Constraints

- The application does not open on any browser due to security reasons.
- The system need to be permanently connected through internet.

2.5 Assumptions and Dependencies

- There should be internet connection.
- Users should know each other.
- The app depends on third-party authentication providers for user registration.

3. External interface Requirements

3.1 User Interfaces

The Chat App will have the following user interfaces:

- Registration and login screens
- Chat room creation and management interfaces
- Contact management interface
- Chat interface with message input and display
- User profile customization

3.2 Hardware Interfaces

- Device camera access for capturing and sharing photos/videos in chats
- Photo library access for sending existing images
- Microphone access for voice messages
- Push notifications require mobile device token registration

3.3 Software Interfaces

- REST API backend service for client syncing and user management
- MySQL database for chat data storage
- Amazon S3 for storing media attachments
- VS Code for writing the code
- Android Studio for integration
- Firebase Cloud Messaging for Android notifications

3.4 Communications Interfaces

The app uses HTTPS protocol for secure API access over internet.

Voice/video calling features will use WebRTC protocol.

4. System Features

➤ One on one Chats

Users can search and select a contact to start a private 1-1 chat.
Sent/received messages are saved per contact chat history.
Tap and hold on a message to see options like copy, forward, delete.
New incoming messages trigger a notification.

➤ Group Chats

Users can create and name a new group chat with multiple contacts.
Group chat history is retained. New messages notify all participants.
Group admins can add/remove participants.
Leave option lets a user exit a group chat.

➤ Media Sharing

Attach photos from camera or photo library.
Capture and attach new videos up to 1 minute long.
Send GIFs and stickers to illustrate reactions.
Record and send voice messages up to 30 seconds.
Share documents from cloud or local device storage.

➤ User Profile

Create a profile with name, profile photo and optional bio.
Set a custom status message visible to contacts.
Sync profile information across mobile and desktop apps.

➤ Settings

- Notifications: Mute chats, disable read receipts, etc.
- Themes: Choose colour, background image, etc.
- Support: FAQ, contact us, terms of use.
- Account: Change username, password, etc.
- Privacy: Block users, hide status, limit profile visibility.
- Storage: Manage chat media storage usage.
- Friends: Manage contacts from address book.

5. Other Non-Functional Requirements

5.1 Performance Requirements

5.1.1 Speed

The Chat app should respond quickly to user input and should provide a smooth and responsive user experience. The app should load quickly and should not have any significant delays when switching between screens or performing common actions like sending or receiving messages.

5.1.2 Scalability

The app should be designed to handle many users and a high volume of messages without significant performance degradation. The app should be able to handle simultaneous connections from multiple users and should be designed to scale up or down as needed to accommodate changes in user demand.

5.1.7 Reliability

The app should be designed to be highly reliable and should minimize the risk of data loss or corruption. The app should include features like automatic data backup and recovery to ensure that user data is always safe and accessible.

5.1.8 Security

The app should be designed to be highly secure and should protect user data from unauthorized access or tampering. The app should use industry-standard encryption and security protocols to protect user data in transit and at rest.

5.1.9 Compatibility

The app should be designed to be compatible with a wide range of devices, operating systems. The app should be tested on multiple devices and platforms to ensure that it works correctly and efficiently on all supported configurations.

5.1.7 Resource Usage

The app should be designed to use system resources (such as CPU and memory) efficiently and effectively. The app should minimize unnecessary resource usage to ensure that it does not slow down other apps or affect the overall performance of the user's device.

5.2 Software System Attributes

5.2.1 Compatibility

The ChatZen app must be designed to be compatible with a wide range of devices and platforms, including different screen sizes, resolutions, and operating systems. The app must be designed to work seamlessly across all supported platforms, including mobile devices.

5.2.2 Performance

The ChatZen app must be designed to perform efficiently and effectively, with fast load times and responsive user interactions. The app must be designed to minimize resource usage and avoid memory leaks or other performance issues that could affect the user experience.

5.2.3 Maintainability

The ChatZen app must be designed to be easy to maintain and update, with a modular and extensible architecture that allows for easy addition of new features or modifications to existing features. The app must be designed to minimize code complexity and maximize code reuse to reduce maintenance costs.

5.2.4 Accessibility

The ChatZen app must be designed to be accessible to all users, including those with disabilities or impairments. The app must be designed to meet accessibility guidelines and standards, such as those set by the Web Content Accessibility Guidelines (WCAG)

5.2.5 Usability

The ChatZen app must be designed to be easy to use and navigate, with clear and concise instructions and feedback for users. The app must be designed to minimize the risk of user errors, such as accidental message deletion or sending messages to the wrong recipient.

6. Other Requirements

- **Appendix A: Project Document**

The project documentation for an application for online commodity and delivery system

DETAILED PROJECT DOCUMENTATION

Candidate Name & Roll No. : Shreya Singh (LCS2022046)

Raghunandan Bansal (LCS2022058)

Rishi Raj Maheshwari (LCB2022005)

Course of Study: B.Tech

- **Appendix B: Test Cases**

- HIGHER LEVEL ITEMS TO BE TESTED**

- Chat application and supporting infrastructure
 - Application running on different client devices

- HIGHER LEVEL ITEMS NOT TO BE TESTED**

- SRS of chat application
 - User Manual of chat application
 - Already existing chat application
 - Manual processes related to the application
 - Any legacy system

- LOWER-LEVEL ITEMS TO BE TESTED**

- User Profile
 - Chatting
 - Add Friend
 - Remove Friend
 - Find Friend
 - Register
 - Login
 - Verification
 - Logout

- LOWER-LEVEL ITEMS NOT TO BE TESTED**

- **User**