



IJIRCCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 10, Issue 7, July 2022

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 8.165



9940 572 462



6381 907 438



ijircce@gmail.com



www.ijircce.com

Talkrrr! Real-time File Sharing and Chat Application

B Varaprasad, Harsh, Nishith Ganesh Umesh, Manohar R

Student, Dept. of Information Science and Engineering, SMVIT-Bengaluru, Karnataka India

Assistant Professor, Dept. Of Information Science and Engineering, SMVIT-Bengaluru, Karnataka India

ABSTRACT: Talkrrr! is an instant messaging and file-sharing platform. Talkrrr! allows users to communicate with each other through various means. Users can communicate with text messaging, media and files in private chats or as part of communities called Servers. Servers are a collection of persistent chat rooms and voice chat channels. The application is implemented using ReactJS to power the front-end and Firebase for authentication and database. This project focuses on bringing different communities like working professionals, students, artists, gamers and casual users together.

I. INTRODUCTION

Instant messaging (IM) is a technology that allows text to be transmitted in real time over the Internet or other computer network. Messages are typically transmitted between two or more parties when each user enters text and initiates a transmission to recipients who are all connected to a common network. It differs from email in that instant messaging conversations take place in real time (hence "instant"). Most modern instant messaging apps (sometimes called "social messengers", "messaging apps" or "chat apps") use push technology and also add additional features such as emoticons (or graphic smileys), file transfer, chatbots, voice over IP or video. chat skills.

Instant messaging is a set of communication technologies used for text communication between two (private messaging) or more participants (chat room) over the Internet or other types of networks (see also LAN messenger). IM - chat takes place in real time. Importantly, online chat and instant messaging differ from other technologies such as e-mail because of the perceived quasi-synchrony of communication by users. Some systems allow messages to be sent to users who are not then "logged in" (offline messages), removing some of the differences between instant messaging and email (often by sending the message to an associated email account).

IM enables effective and efficient communication and allows instant confirmation or reply. However, IM is not necessarily supported by transaction control in principle. In many cases, instant messaging has added features that make it even more popular. For example, users can see each other through webcams or talk directly for free over the Internet using a microphone and headphones or speakers. Many applications allow file transfer, although they are usually limited to the allowed file size. It is usually possible to save a text conversation for later use. Instant messages are often recorded in a local message history, similar to the persistent nature of email.

II. OBJECTIVE

This project focuses on bringing different communities like working professionals, students, artists, gamers and casual users together. To do this, our project has introduced the concept of servers to keep the communities organized.

Talkrrr! is about giving people the power to create space to find belonging in their lives. Its aim is to make it easier for users to talk regularly with the people they care about, to build genuine relationships with their friends and communities close to home or around the world.

Our project is aimed towards bringing different communities of people to enable them to interact with one another and bring together a united front to learn, create, share and simply socialize.

II. METHODOLOGY

A. ARCHITECTURE

While the system design of a chat application is unique in how it deals with idiosyncratic business needs, you can always break it down into two main components: the chat client and the chat server.

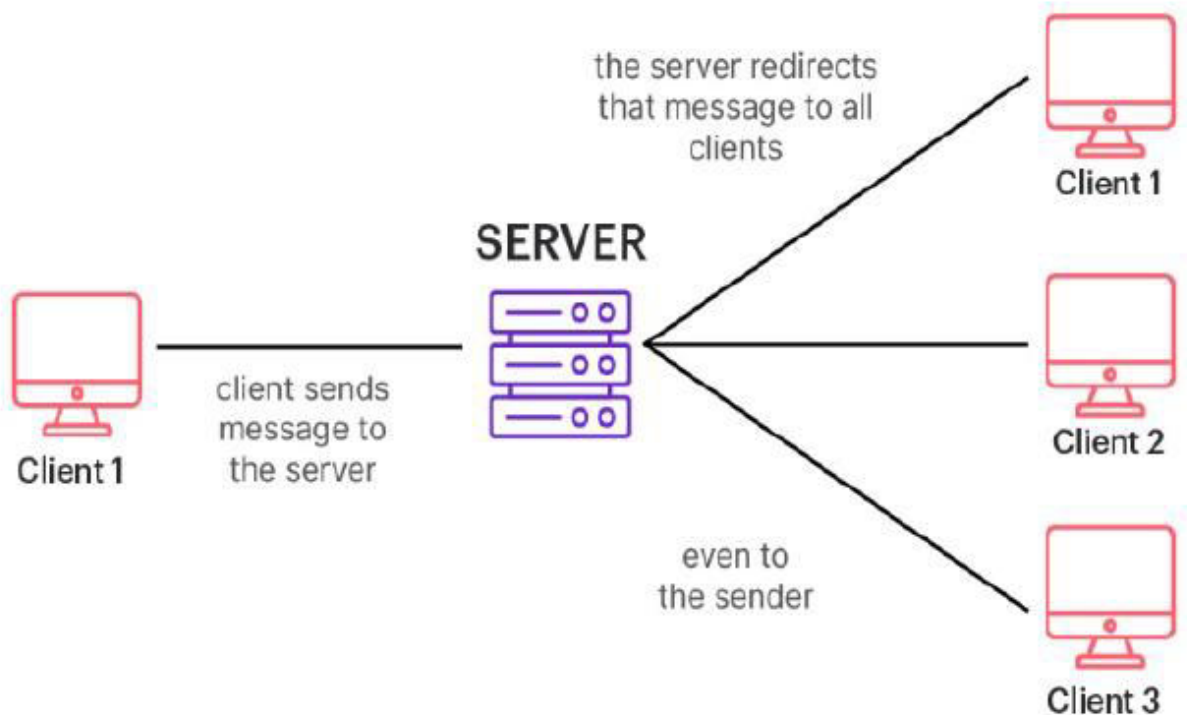


Figure 1: System Architecture

Chat Client

The chat client is what the user experiences. A chat application for desktop, web or smartphone, the chat client is responsible for interacting with the operating system (ie your computer, browser or smartphone). Interactions include sending push notifications, displaying data to the user, and saving messages and files. When you write a message and press send, the chat client transfers that message to the next major component: the chat server.

Chat Server

A chat server is just that, a server (or usually many servers) that hosts all the software, frameworks, and databases needed to run a chat application. This server or group of servers is responsible for securely receiving the message,

identifying the correct recipient, enqueueing the message, and then delivering the message to the chat client. Chat server resources can include a REST API, a WebSocket server, an AWS instance for media storage, etc.

Chat REST API

The Chat REST API is most often used to facilitate the functionality of a chat application outside of messaging. For example, authentication, profile settings, and notification settings can be managed through the REST API. All these features are built into the CometChat SDK and Chat API.

WebSocket Server

The WebSocket server and client library are useful components for a chat application. Typical client-server communication is done using HTTP and requires the client to request data from the server. The server itself cannot send data to the client without the client first asking.

In a chat application, this quickly leads to inefficiency, as the client would have to poll the server every second for new messages. A WebSocket is a persistent connection between a client and a server that provides a two-way communication path. This means that the server can send data to the client without first receiving the request. Real-time chat is a perfect use case for WebSockets.

Media storage

Chat applications require both data storage and media storage. Profiles, messages and media files must always be available. Your choice of database, whether it's a relational database or a NoSQL database (like MongoDB), largely depends on your use cases, what's important to your users, and what your development team knows. You can even use a combination of the two, using a reliable and robust relational database for general data like profile settings, and using a NoSQL database for messages. Because key-value stores allow for easier horizontal scaling and low data access latency, NoSQL databases like Cassandra are ideal for storing messages. While every chat application has a chat client and server, the size and functionality of these components will vary depending on the intended use of the application. Components can scale from an MVP chat app with basic features to a full-fledged, mature chat app like WhatsApp, which has over two billion active users.

III. IMPLEMENTATION

The implementation of any software is always preceded by important decisions regarding the choice of platform, language used, etc. These decisions are often influenced by several factors, such as the real environment in which the system works, the required speed, security concerns, other implementation-specific details, etc. the fact of implementing a technical specification or algorithm as a program, software component or other computer system through programmatic deployment. There may be many implementations for a given particular standard.

A. LIBRARIES USED

React

React is a popular JavaScript framework for building front-end applications such as user interfaces that allow users to interact with programs. Originally created by Facebook, it gained popularity by allowing developers to build fast applications using an intuitive programming paradigm that combines JavaScript with an HTML-like syntax known as JSX.



CometChat SDK

CometChat's in-app communication platform provides a scalable, secure and easy-to-use solution that supports meaningful user-to-user interaction for your growing business. Quickly build a reliable and full-featured chat into any mobile or web application. CometChat APIs are built specifically to help deliver products faster and be completely flexible.

Firebase

Firebase Authentication provides backend services, easy-to-use SDKs, and out-of-the-box UI libraries to authenticate users in your app. It supports authentication with passwords, phone numbers, popular federated identity providers like Google, Facebook, and Twitter, and more. Firebase Authentication integrates tightly with other Firebase services and uses industry standards such as OAuth 2.0 and OpenID Connect, so it can be easily integrated with your own backend.

IV. CONCLUSION AND FUTURE WORK

In conclusion, Talkrrr! provides a better and flexible system for chatting. It is developed with recent advanced technologies in a way to provide a reliable system. Main advantages of the system are instant messaging, real-world connectivity, group chat, etc. This application can find better need in the market for most of the organizations aim at having private applications for them. Additional features included in the system, based on the public needs which includes conference call, video chat, file share, etc.

Currently Talkrrr! exists only as a web application that can be accessed via any web browsers like Chrome, Edge, Safari etc. Future work would include making the app accessible on smartphones(Android or iOS) and also making it a stand-alone application for Windows, Linux and Mac based operating systems. Additionally, we will aim to provide better and more robust security.

REFERENCES

- [1] "Instant Messaging and Social Networks — The Advantages in Online Research Methodology" by Kazeem Oluwakemi Oseni, Kate Dingley, and Penny Hart, International Journal of Information and Education Technology, Vol. 8, No. 1, January 2018.
- [2] "Online Chatting Application", By Jhalak Mittal, Arushi Garg, Shivani Sharma, 6.565, Volume 10 Issue 04, April 2020.
- [3] "Survey Analysis on the usage and Impact of Whatsapp Messenger", By Naveen Kumar and Sudhansh Sharma, SOCIS, Indira Gandhi National Open University, Vol 8, Issue 3, July-September 2016.
- [4] "A Survey of WebSocket Development Techniques and Technologies", By T. Wirasingha, N.R. Dissanayake, Informatics Institute of Technology, No.57, Ramakrishna road, Colombo 06, Sri Lanka, September 2016.
- [5] "Purely Decentralized P2P File Sharing Systems and Usability", By Nicky Reid, June 3, 2015.
- [6] Cometchat SDK – "<https://www.cometchat.com/docs/react-chat-ui-kit/overview>"
- [7] Firebase – "<https://firebase.google.com/docs>"



INNO  SPACE
SJIF Scientific Journal Impact Factor

Impact Factor: 8.165

 **doi**[®]
cross  **ref**

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

 9940 572 462  6381 907 438  ijircce@gmail.com



www.ijircce.com

Scan to save the contact details