Chat Messenger - An Application for Conversation

Ritik Sharma
Department of Computer Science and Engineering
Chandigarh University
Mohali, India
ritiksharma98166@gmail.com

***Abstract*— This project (Chat messenger) is about building a chat messenger in which we are able to send our message to another person with the help of the internet. We can register and login into our account with the help of the e-mail services and user can login into their account with these credentials. This application enables us to communicate with any person in the world only if they have this application and a working internet connection. This application works for free and provide us quick responses. Due to its cost, this application is very useful for the business purpose as it provides us free communication among the people in the world.**

Keywords*— Artificial Intelligence, chatbot, machine learning, NLP, patterns, responses, heuristic, goal-oriented.

I. INTRODUCTION

In this world, communication is the most important thing. To have a communication we use various methods in order to have a well communication. Communication means exchange of messages or exchange of our thoughts with someone else. With the passage of time the communication methods are being developing. We learn many languages, gestures, or any other visual things by which we can have communication.

In this project we will develop an application which uses internet as a medium to for sharing the messages among the different individuals. This application will run on android and firebase. For the frontend we are using the Kotlin language which will provide us the layout for our application. We will run Kotlin language in android studio. Android studio basically runs all the code that will generate the android app. Android app can be build using java or Kotlin.

Chat application will offer us to have a real-time transmission of the messages between the people. This application enables us to have point to point communication which means the two ends are connected in a form that there is no one else can have access of that tunnel or path which is being established at the time of establishment of the end to end encryption. [1]

II. RELATED WORK

A. WhatsApp: -
WhatsApp is chat messenger-based application in which we are able to chat with another person. This application is currently owned by the Facebook organization.

B. Instagram: -
Instagram is a social media-based application in which we can post our messages to others, post pictures and videos and get liked by other persons.

C. Facebook: -
Facebook is the application which is similar to the Instagram and these both applications are being owned by the same organization. The working of Facebook and Instagram are same. We can post our messages, pictures and videos which can be seen by others and get our posts liked.

D. Messenger: -
Messenger is used to have a conversation by internet and is being used by the Facebook. In order to chat on Facebook, we need messenger to be installed on our devices.

E. Snapchat: -
Snapchat is a chat-based application which is designed to have a conversation between its user and we are able to send text, voice messages and video notes to the other person on the other side receiving the messages.

III. PROPOSED ARCHITECTURE

IV.

When a user send a particular message to another user, then this message will be save in the payload section of the packet and then this message will be forwarded to the Firebase Cloud Messaging of Firebase and then this message will be stored in the database of the Firebase containing all the essentials details like metadata, message

and other information like destination and source of the message and then finally this message will be delivered to the receiver end. A receiver will only be able to view the notification and message if and only if the user is logged in into the application with their own login credentials. These credentials will be asked at the time of registration and then this information will be stored in the Firebase Database for the future use.

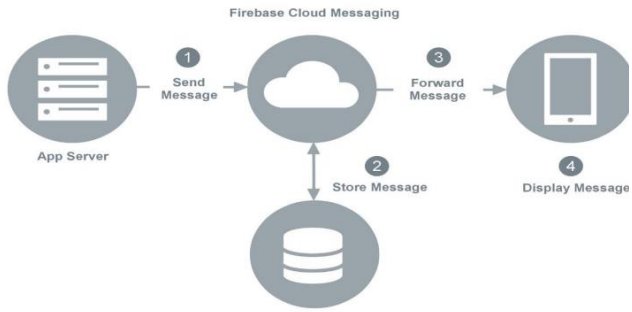


Fig. 1. Architecture Diagram for the messenger

The following facts need to be kept in mind during designing a Chatbot:

A. Selection of OS

Windows is used for this project because it is user friendly. It is also robust.

B. Selection of Software

Android Studio is used for programming using Kotlin language makes it easy to share code that contains live code, visualizations and explanatory text.

C. Creating an application

For creating a messenger, a program has to be written. Kotlin programming language is used for programming. The Kotlin is created in such a way to help the user, improve the communication and amuse the user.

D. Creating a Chat

The chat is created using a pattern that is known to the user and is easy to understand. Chat dialog box show up to create conversation. This dialog box is created using Kotlin.

E. Simple GUI

The design of a Chat-messenger is very simple. Simple GUI attracts the user to interact with the application. And which help user to work for on the application for much more time than a usual person can do.

V. METHODOLOGY

This project is built on the android platform and this application will able to run on the all android Operating System on mobile phones. We will build this project with the help of the android studio and a programming language i.e. Kotlin. For the UI and other function, we will use Kotlin language and the environment is android studio. The IDE i.e. android studio provides us the environment by which we can write our code for the designing purpose and rest of the database will be handles by the Firebase Database provided by the Google. Google provide us the cloud services by which we can perform these actions. We can manage our users, can grant and deny the access to one account. This Firebase also provides us the functionality of the authentication by which the data can be available to those who have access to it.

Steps for building chat messenger: -

• Designing the UI of the application.

• Creating the database for the application.

• Connecting the database.

• Creating the user's profile.

• Managing users accounts. •

Establishing the connection between user to user account.

• Testing the particular modules.

• System testing i.e. testing all the modules working accurately together.

VI. IMPLEMENTATION

When a user send a particular message to another user, then this message will be save in the payload section of the packet and then this message will be forwarded to the Firebase Cloud Messaging of Firebase and then this message will be stored in the database of the Firebase containing all the essentials details like metadata, message and other information like destination and source of the message and then finally this message will be delivered to the receiver end. A receiver will only be able to view the notification and message if and only if the user is logged in into the application with their own login credentials. These credentials will be asked at the time of registration and then this information will be stored in the Firebase Database for the future use.

This project is built on the android platform and this application will able to run on the all android Operating System on mobile phones. We will build this project with the help of the android studio and a programming language i.e. Kotlin. For the UI and other function, we will use Kotlin language and the

environment is android studio. The IDE i.e. android studio provides us the environment by which we can write our code for the designing purpose and rest of the database will be handles by the Firebase Database provided by the Google. Google provide us the cloud services by which we can perform these actions. We can manage our users, can grant and deny the access to one account. This Firebase also provides us the functionality of the authentication by which the data can be available to those who have access to it.

This chat messenger is able to create user, login with their credentials. In the current time now, we are able to send messages to all the people having the same platform and vice-versa. The Firebase database is now ready and is fully functional, it stores the unique Id's of the user and other information which tells us who is that user. The Firebase storage is also ready, in this module the storage will used to store the profile picture on the cloud and other essential information. Firebase handles all the incoming and outgoing calls for the message request and all the login or registration request.

Login and Registration Page: Registration window Login window These snapshots are of the Registration and of login window from an android mobile phone. This app is installed on mobile phone and currently working accurately. In the registration window, there are input fields, Profile Picture, username, email and password. On entering the valid data, the app will display a pop message that your account has been created. In the login window, there are two input fields, these fields will contain the data for login credentials of the user. User is asked to enter their Email and password and then click on the login button, then the user will be redirected to their account.

Chat log is the place where all the contacts are being displayed. Even if we text a person only single time, then we will be able to see that contact in the chat log. Here, a user had a chat with a person name tester99, so that's the reason why his name is in the chat log. If we don't want to remain login into our account then we can logout our account by tapping the sign-out icon on the top right corner, and now the application will bring us back to the registration window. In the bottom right corner of the chat log, we have a button which will enables us to display the all the contacts that are currently using this application and now we can communicate with them by selecting their name, and the application will take us to that chat window, where we can chat with the other person.

The **Firebase Database** will give us real time database in which we used to store the information of the user. User Id, username and the profile picture path of the Firebase Storage where their pictures area stored on the cloud. User Id is created with the function UUID which gives us a random user id in the form of string which is unique. All the user data can be managed from here. Addition or removal of user's account directly effects our database. If a user register himself then the details are automatically inserted here.

The **Firebase cloud storage** provide us space on the disk at their data centres. In this storage we are will store the information and other things related to the users' account. Now, we are storing the profile pictures of the user which they have set at the time of the registration. These things are kept in the folder by managing all the data in the hierarchal order. These images have a key value which is unique and can be used to represent a particular account of the user and we can get their path in which they are stored on the servers.

Managing User's access and Authentication: - After a successful registration of the user, he will be asked login into their account, so in-order to grant the access we have authentication system which show us how much user we have, to whom we want to remove or block access or we can just delete the account from here. All these things can be managed by the UUID provided by the system to the particular user.

I. Graphical User Interface (GUI) Programming

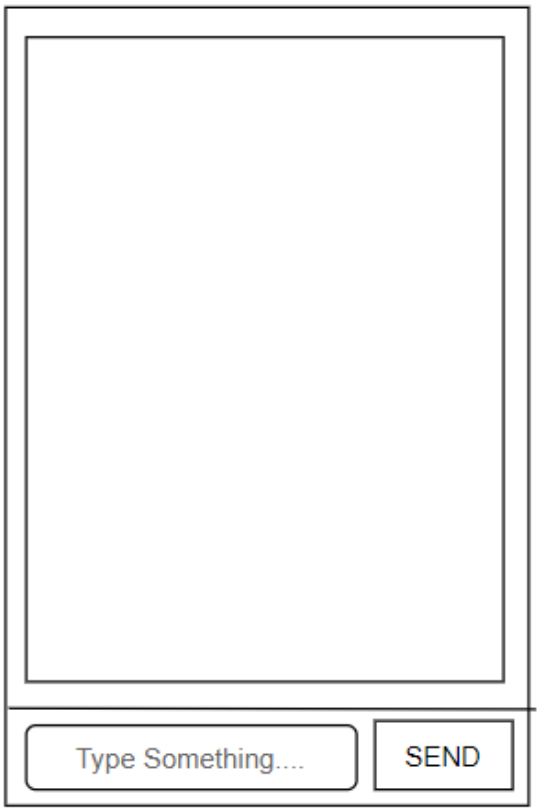


Fig. 5. Basic GUI/ Messaging Interface Design

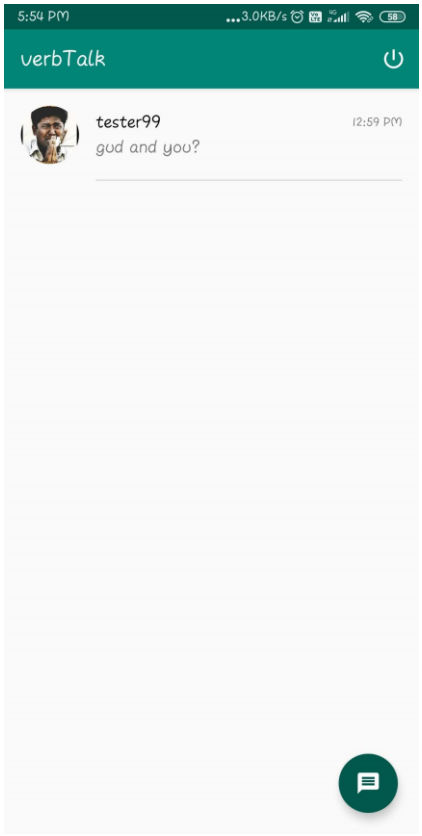
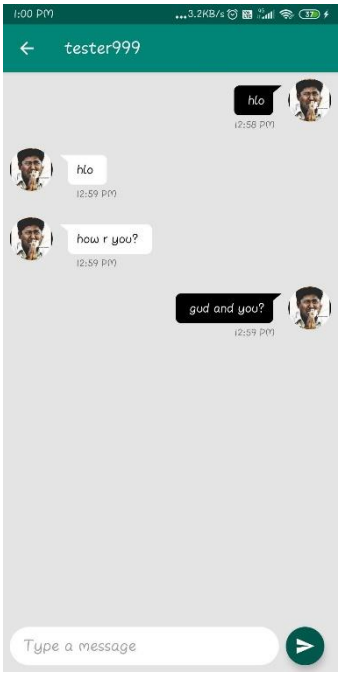


Fig. 6. An instance of conversation between users with the help of GUI.



VII. FUTURE SCOPE

The chat messenger is able to transmit the data only in the form of the text. In future, it will able to send the images, videos, documents and other type of data. Not just, text or other format of data it will able to send data. It will be able to transmit the voice recording and the audio files. Options to select between different privacy modes which will give some extend of privacy to the user. In-built application password authentication with support of pin and biometric. In-built font style options like style and size. Adding new themes will make our application more personalize or customize according to our need like dark, light themes are the most popular one. [6-7]

VIII. CONCLUSION

A chatbot is one of the simple ways to transport data from a computer without having to think for proper keywords to look up in a search or browse several web pages to collect information; users can easily type their query in natural language and retrieve information. In this paper, information about the design, implementation of the chatbot has been presented. From the survey above, it can be said that the development and improvement of chatbot design grow at an unpredictable rate due to variety of methods and approaches used to design a chatbot. Chatbot is a great tool for quick interaction with the user. They help us by providing entertainment, saving time and answering the questions that are hard to find. The Chatbot must be simple and conversational. Since there are many designs and approaches for creating a chatbot, it can be at odds with commercial considerations. Researchers need to interact and must agree on a common approach for designing a Chatbot. In this project, we looked into how Chatbots are developed and the applications of Chatbots in various fields. In addition comparison has been made with other Chatbots. General purpose Chatbot must be simple, user friendly, must be easily understood and the knowledge base must be compact. Although some of the commercial products have recently emerged, improvements must be made to find a common approach for designing a Chatbot.[6-7]

REFERENCES

- https://www.geeksforgeeks.org/project-idea-magical-hangouts-an-android-messaging-app/
- https://en.proft.me/2017/06/2/firebase-android-notifications-firebase-cloud-mess/
- https://en.wikipedia.org/wiki/Instant_messaging
- https://en.wikipedia.org/wiki/Facebook_Messenger

- https://firebase.google.com/docs/reference/android/com/google/firebase/messaging/FirebaseMessagingService
- https://www.simplifiedcoding.net/firebase-cloud-messaging-tutorial-android/