International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 06 | June - 2023 SJIF Rating: 8.176 ISSN: 2582-3930

Real-Time Communication Application Based on Android Using Google Firebase

Anurag Dobhal[1]
Department of Computer Applications
Graphic Era Hill university
Dehradun, India

Yogesh Negi[2]
Department of Computer Applications
Graphic Era Hill university
Dehradun, India

Abstract: In today's world, communication is extremely vital and keeping this communication real-time is essential as our lives have become more fast paced. Keeping this is mind, a communication application should be able to transfer files and messages instantly without or with minimal delay, depending on the broadcast medium. For such a system to be functional, there must be a database which will update in real-time so as to keep track of all the data being transferred. Google Firebase is a service which provides such a real-time database server, along with a host of other features and Firebase enables us to develop communication-based applications with relative ease. In this paper, we propose a system which will be capable of sending text-based messages and files such as images, audio, videos, texts over through the internet between two users on the network in real-time. We make use of the Android operating system and Google Firebase to handle the backend of the communication operation, highlighting the various features of both the operating system and the service. The use of chat applications has increased significantly in recent years due to the rise of mobile devices. In this research paper, we present the development and implementation of an android chat application using Java programming language. The chat application is designed to provide real-time communication between users, with features such as group chat, media sharing, and user authentication. The application was developed using Android Studio and Firebase for the backend. We discuss the design and architecture of the chat application, the features implemented, and the challenges faced during the development process. Finally, we evaluate the performance of the application and discuss its limitations and future directions.

Keywords: communication, real-time, android, firebase, messaging.

INTRODUCTION

Communication is a vital aspect of our daily lives in this world. So, since time immemorial, people have been communicating with each other via various mediums. As new lands were discovered, the distance between people increased, the communication mediums changed. From letters to telegrams, people used different means to communicate with each other but the drawback of most of them were the time they took to be delivered. Letters could take up to days and this was a major issue with communication back then. With the advent of cell phones, developers looked to implement a text-based messaging service which would allow instantaneous communication facilities. Hence, in 1984, the concept of SMS was developed in the Franco - German GSM cooperation by Friedhelm Hillebrand and Bernard Ghillebaert. However, the main limitation in the case of the new idea was the limited size available for writing the message, mainly 128 bytes. The first SMS was sent in 1992 after various improvements after the original inception of the idea. The first commercial SMS service was deployed in the following year by Aldiscon with Telia in Sweden. SMS was the main form of communication in the 2000s but their high costs were a disadvantage, although people widely

used them to communicate swiftly or to convey any urgent message. As smartphones began to rise in the late 2000s, a wide range of messaging applications based on the different operating systems became available and rose to popularity among the general consensus of people. Among these most notable were Whatsapp, WeChat, Viber, Snapchat and a few others. The proliferation of mobile devices has led to an increase in the use of chat applications for real-time communication. The ability to communicate with others regardless of their location and time has made chat applications a popular means of communication. Android, being the most widely used mobile operating system, has become a popular platform for the development of chat applications. In this research paper, we present the development and implementation of an android chat application using Java programming language.

In this paper, we look to create a similar application based on previous works and determine how useful Firebase is, in the case of development of messaging applications. Firebase was established by Andrew Lee and James Tamplin back in 2011 yet was launched formally in April 2012. Initially, the framework was designed to be used solely as a real-time database giving its APIs, enabling users to store and synchronize data and information across various users. However, Firebase was taken over by Google in 2014 and today, the service has various functionalities that offer development tools to various enthusiasts and entrepreneurs alike.

Firebase is a framework which is useful for building portable and web applications for businesses which require real-time database which implies when one user updates a record in the database, the update should be conveyed to every single user instantly. It gives a basic and unified platform to many applications along with a host of other Google features packed-in with the service. Firebase handles most of the server-side work when it comes to the development of applications. There are numerous elements that make Firebase such an essential tool in development from a developer's point of view. In this way, it helps maintain a state of harmony between the developer and the client by causing minimal delay of work. In case of the development of a communication or chat application, the most essential components or services offered by Firebase are: -

Real-time Database: Real-time Database is a cloud-hosted database. Data is stored as JSON and synchronized continuously to each associated client. When any cross-platform application is developed with iOS, Android, and JavaScript SDKs, the greater part of the user's demand is based on one Real-time database instance and this instance gets updated with each new data. This feature allows developers to skip the step of developing a database, and Firebase handles most of the backend for the applications. It gives an adaptable, expression-based rules language to define how data should be organized and when information can be perused from or composed to.

Authentication: Firebase Authentication gives backend services, simple-to-use SDKs, and instant UI libraries to confirm clients over an application. It supports authentication using passwords, email id or username, phone number, etc. Users can be allowed to sign in to a Firebase app either by using FirebaseUI as a complete drop-in authentication solution or by using the SDK to manually integrate one or a few sign-in techniques.

Storage: Firebase Storage was designed for application developers who need to store and serve user-generated content, for example photos or any other file. It gives secure document transfers and download for Firebase applications, regardless of network quality. Firebase Storage is upheld by Google Cloud Storage, a capable, basic and cost-effective object storage service.

Cloud Messaging: It is a cross-platform solution that allows developers to dependably convey messages at zero expense. Developers can send notification messages to drive user reengagement and maintenance.

RELATED WORK

Many developers are currently developing messaging applications with online solutions similar to Firebase, which provide real-time database integration facilities. Various open-source platforms such as Parse Server or Horizon offer similar services such as Firebase and offer developers to migrate from one vendor to another, but they also come with their own problems. Developers are also trying to develop methods to optimize file transfer through such applications and to integrate more technologically advanced features into their applications.

WORKING ALGORITHM

Our communication application or messaging system aims to provide a platform for two individual users separated by a certain geographical distance to communicate with each other, through the Internet with the help of various Firebase tools. The system is divided into various modules, each module dealing a separate task and working in sync with the other modules without any conflicts. The logic and working algorithm behind the major modules shall be explored in the following sections:

I. **Register Module***:* This module is launched when the user wishes to create a new account. This creation is handled on the server side by Firebase Authentication, based on the Sign-In Method selected. Upon successful registration, a new entry is created in the database and the user is registered into the authentication section of the Firebase console.

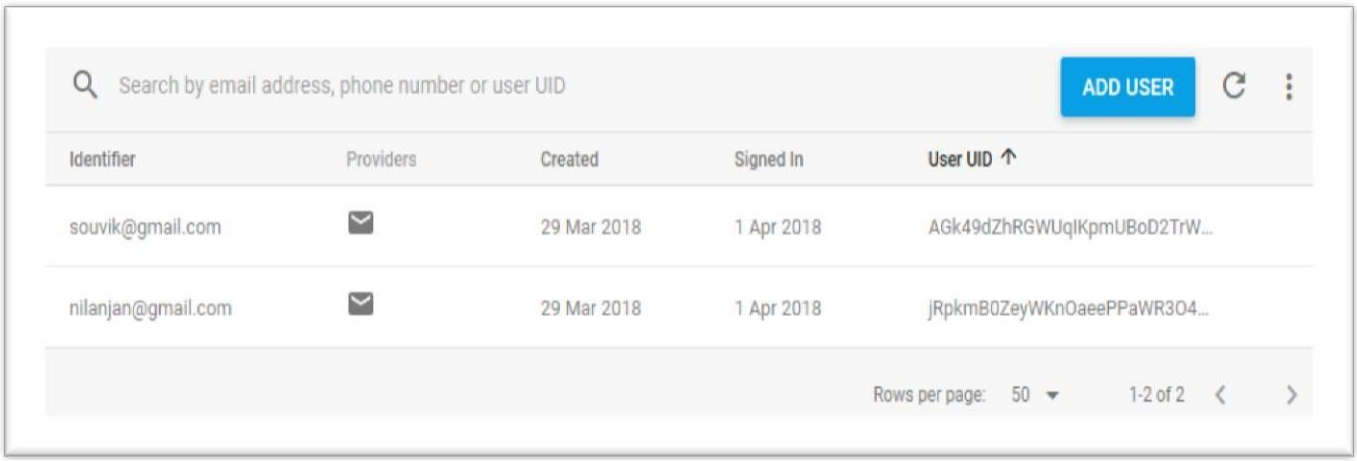


Fig. 1: Authentication Tab of Firebase Console

Each user registered on the network is assigned a unique UID used to identify and refer to them in our system.

II. **Login Module***:* This module is launched when the user wishes to login into the system. The user has to input the required credentials and the Firebase system checks whether these credentials are correct or not. The password is stored as a hash which can be decrypted using the following algorithm:

```
hash_config {
algorithm: SCRYPT,
base64_signer_key:*unique key*rounds: 8,
mem_cost: 14,
}
```

This assures the security of the system.

III. **Chat Module:** This is the main module of the system which will handle the messaging activity of the system. Thismodule

is initiated when the user wishes to chat with other users on the network. On launch of this module, a Firebase databasereference is instantiated along with the different UI elements

```
ChatActivity(){

//initialize database reference

//initialize UI else
}
```

On launch, we load any previous messages with a specified function which fetches previously sent messages. After initializing all elements, if the users send a message then a check is performed if the sent message is a text, image or any other file.

```
CheckMessageType(){

String type; switch(type) {

case “text”; //Initiate normal text upload;case “image”; //Initiate image upload;

case “document”; // Initiate file upload;

}
```

Upon determining the type of the message, a function is run to upload the message, image or file to the Firebase Storage and an entry is made into the Firebase Real-time Database, helping us to monitor the working of our application.

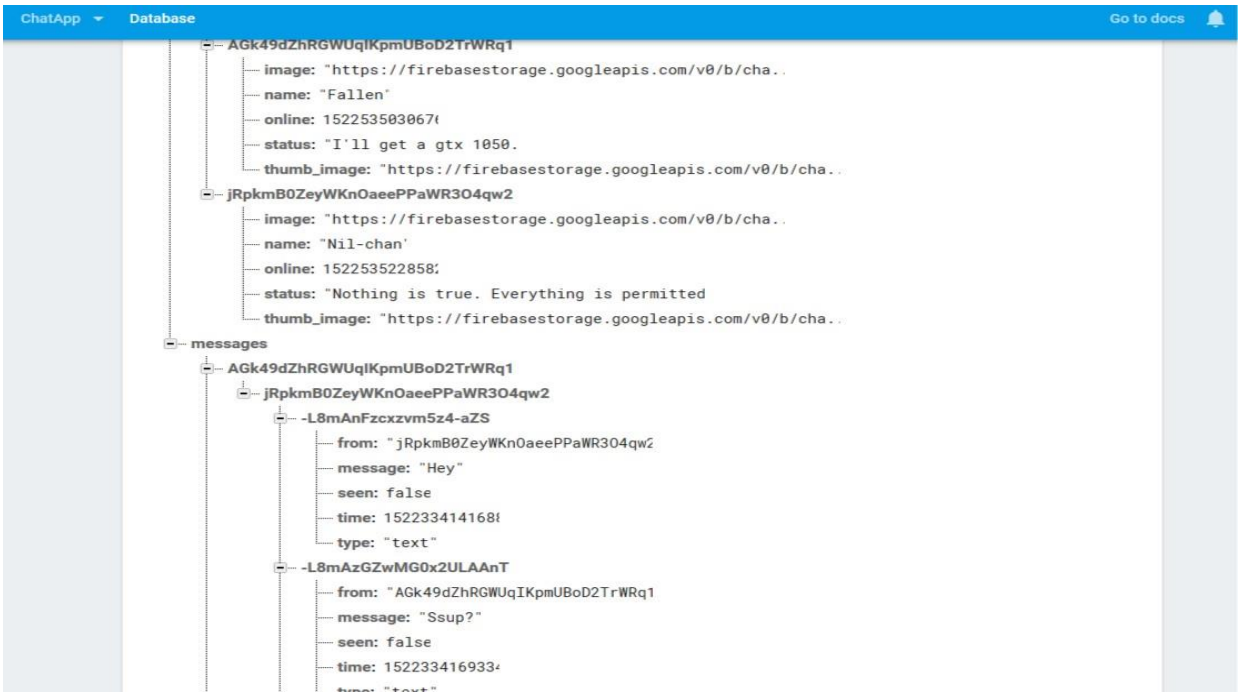


Fig. 2 The Firebase Database with user entries and message entries

This is the main algorithm on which the applications work and other add-on functions are used which affect the UI and other functionalities of the application to make it more appealing to the users.

METHODOLOGY

We developed an android chat application using Java programming language, Android Studio, and Firebase backend. The application provides real-time communication between users, with features such as group chat, media sharing, and user authentication. We used Firebase for the backend, which provides a real- time database for storing chat messages, user authentication for secure communication, and cloud messaging for notification of new messages. We used the Model-View-Controller (MVC) architecture pattern to design the chat application.

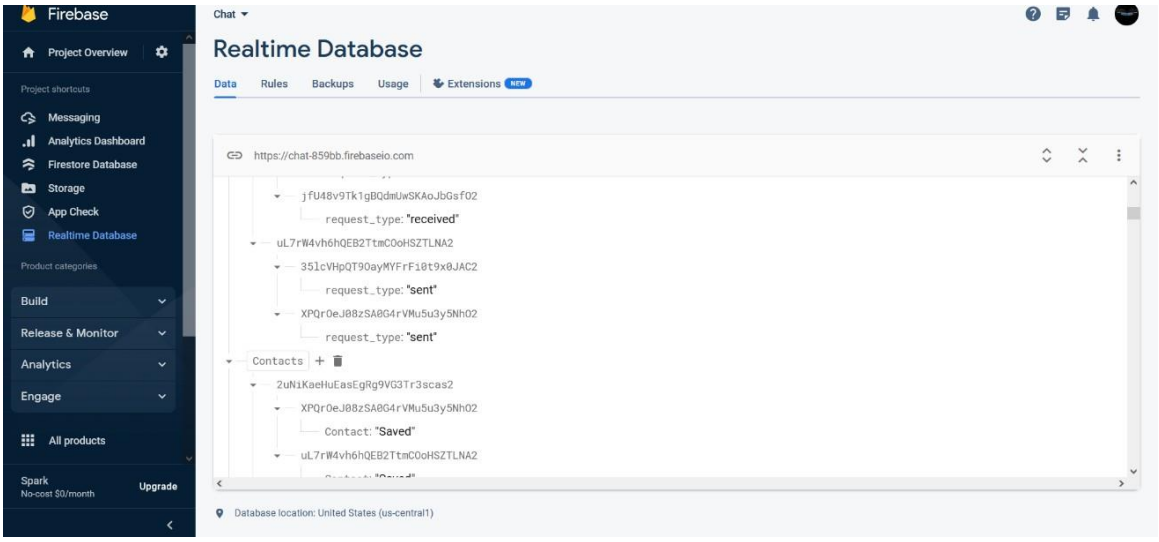
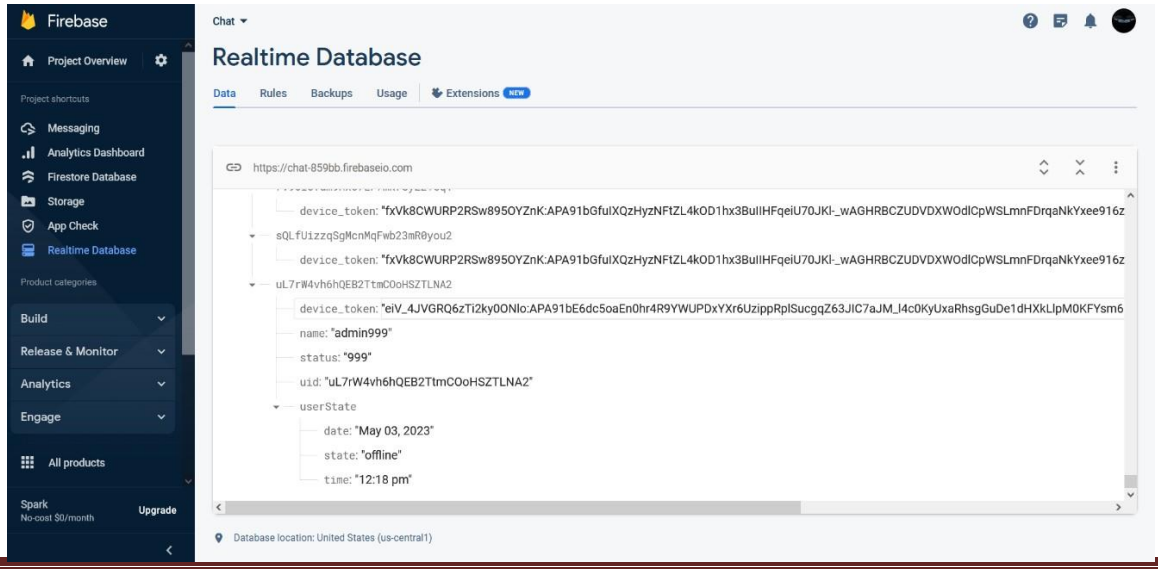
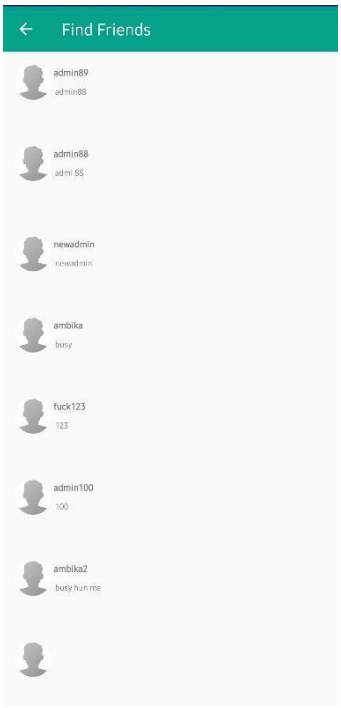


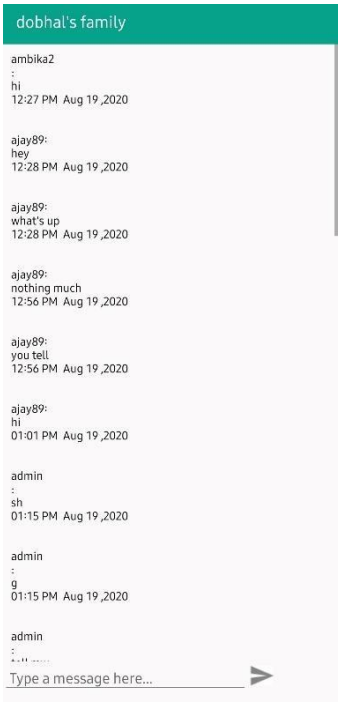
Fig. 3 The Firebase Database with categories



RESULT AND DISCUSSION

On completion of this system, we are left with a fully functional communication application capable of sending messages in real-time and images were also being transferred. The Firebase services are important in the sense that those tools made the development of this applications a lot more efficient and lot faster compared to building a traditional server-side database using a scripting language. The application is user-friendly and intuitive, so using it is not a difficult task. We developed an android chat application that provides real-time communication between users. The application was designed using the Model-View- Controller (MVC) architecture pattern and implemented using Java programming language. The application allows users to create groups and chat with other members in the group. Users can share media such as images, videos, and audio files. We used Firebase for the backend, which provided a real-time database for storing chat messages, user authentication for secure communication, and cloud messaging for notification of new messages. We evaluated the performance of the application and found that it was fast and responsive.





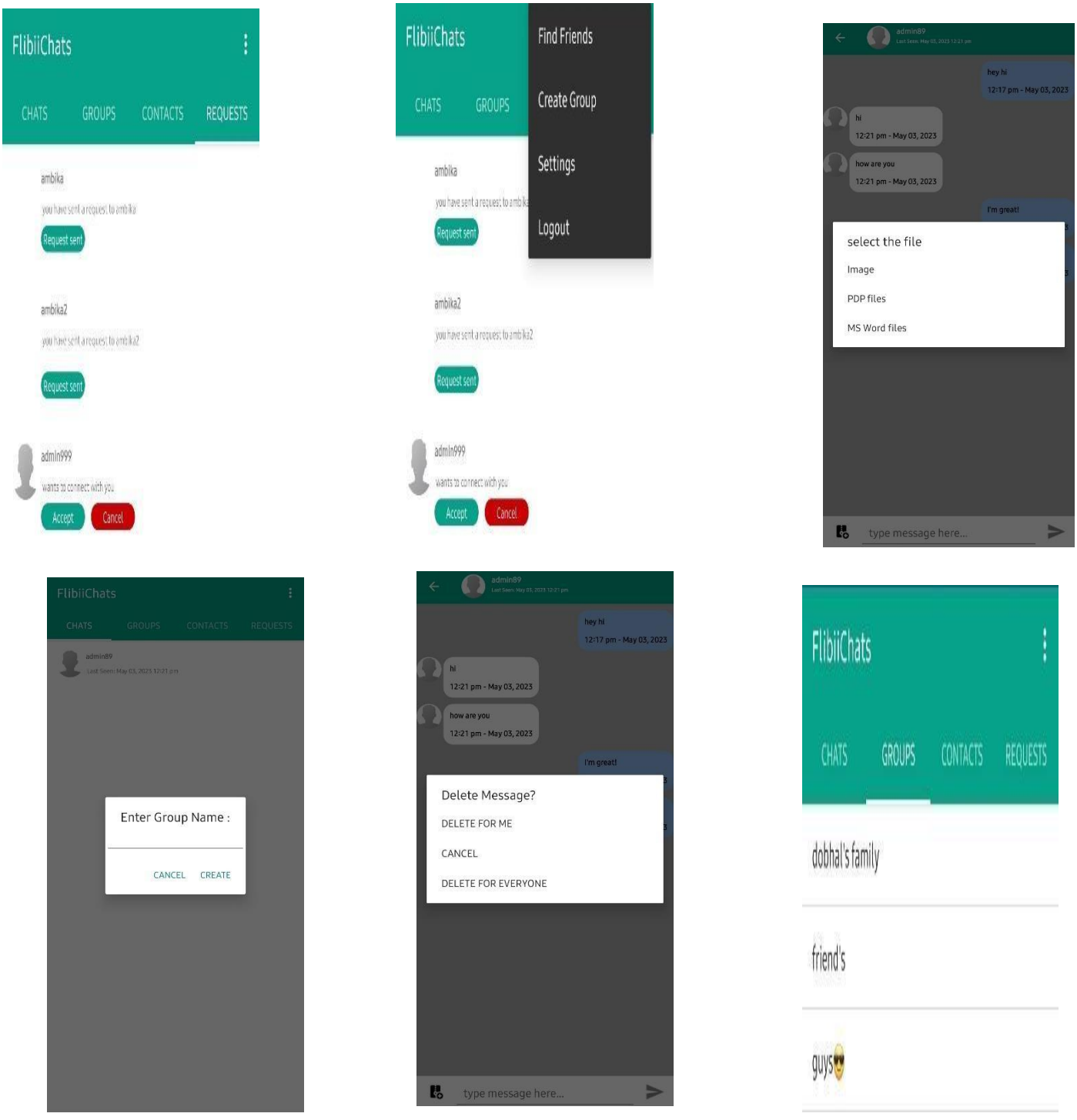


Fig. Interface of the Chat window

CONCLUSION AND FUTURE SCOPE

Firebase is a step forward in the right direction in the context of application development in the sense that it provides an all-round service for developers. Many companies will follow in its footsteps and hence the development of the applications will be much smoother and efficient. As for our application, it has numerous upgrade paths from implementing call functionality or any other enhanced future thanks to the flexible nature of the application and the application will continue to upgrade as long as the technology upgrades. So, the application is extremely future proof. In this research paper, we presented the development and implementation of an android chat application using Java programming language. The application was designed using the Model-View-Controller (MVC) architecture pattern and implemented using Android Studio and Firebase backend. The application provides real-time communication between users, with features such as group chat, media sharing, and user authentication. We evaluated the performance of the application and found that it was fast and responsive. The application has some limitations, such as the lack of end-to-end encryption, which can be addressed in future research.

REFRENCES

1) Firebase. Firebase - Google. https://firebase.google.com/
2) Google. Android Developers. https://developer.android.com/
3) Joshi, A. (2018). Building a Chat Application for Android Using Firebase.
4) Medium. https://medium.com/@aashishj