

Android Based Instant Messaging Tool Using Firebase as Backend

Ashita¹ | Vasudha Bahl¹ | Dr.Amita Goel¹ | Nidhi Sengar¹

¹Department of Information Technology, Maharaja Agrasen Institute Of Technology,

To Cite this Article

Ashita, Vasudha Bahl, Dr.Amita Goel and Nidhi Sengar, "Android Based Instant Messaging Tool Using Firebase as Backend", *International Journal for Modern Trends in Science and Technology*, 6(12): 198-201, 2020.

Article Info

Received on 10-November-2020, Revised on 30-November-2020, Accepted on 04-December-2020, Published on 09-December-2020.

ABSTRACT

In today's world, communication is very important and keeping this communication in real time is important as our lives become faster. Internet communication is becoming more and more important these days. Online communication allows users to communicate with other people in a fast and easy way. Keeping this in mind, the communication app should be able to transfer files and messages instantly without or with little delay, depending on the broadcast field. For such a system to work there must be a database that will update in real time to store all the data transferred. Google Firebase is a service that provides real-time data server, as well as many other features and Firebase enables us to develop applications that are easy to connect to.

In this paper, we propose a system that will be able to send text and text-based files such as photos, audio, video, text online between two users on the network in real time. We use the Android and Google Firebase operating system to manage contact back functionality, highlighting various features of the application and service.

KEYWORDS: *communication, real-time, android, firebase, messaging.*

INTRODUCTION

Communication is an important part of our daily lives in this world. So, from time immemorial, people have been communicating through various means. At first people communicated with one another using books or other sources, as these witches can take a long time to deliver content. Cell phones are another form of communication but going back to any limited or minimal message that needs to be passed on to another user when calling is not a good option. Developers then looked to use text-based communication that would allow for faster communication service. As smartphones began to rise in the late 2000s, more messaging-based messaging systems became available and began to gain popularity among the common people. Among the most notable were

Whatsapp, WeChat, Viber, Snapchat and a few others.

In this paper, we look at creating a similar application based on previous activities and determine how useful Firebase is, in the case of messaging apps development. Firebase is a useful framework for building portable and web-based business applications that require a real-time database which means that when one user updates a record to a database, the update must be transmitted to all individual users instantly. It provides a basic and integrated platform for many operating systems and many other Google features packed within the service. Firebase handles most of the work on the server side when it comes to app development. This feature allows developers to bypass the database creation step, and Firebase manages most of the applications' backend.

In this application we have used Google Firebase as a real-time database that provides real-time data connection [1]. Firebase database is reliable, stable and up-to-date with new market releases. Our communication system or messaging system aims to provide a platform for two users separated by a certain distance to communicate, online with the help of various Firebase tools. The Firebase Real-time[2] database is a cloud-based database that supports multiple Android, iOS and Web platforms.

Firebase is a platform and tool known for its speed and reliability in building real-time applications with a very simple platform, many advanced Google features and other advanced features such as crash reporting and thus allowing developers to build more sophisticated and efficient applications that offer a variety of services.

Most applications require user ID which will help make user data safer and more secure in the cloud. Firebase provides a backend, SDK and is ready to use libraries that help the developer provide authenticity without trying.

In this paper, we look at creating a similar application based on previous activities and determine how useful Firebase is, in the context of the development of messaging applications. In the event that a communication or chat system is developed, the most important features [3] or services offered by Firebase are

Real-time database

All data is stored in the JSON format with any changes to the data, displaying instantly by syncing across all platforms and devices. The Firebase real-time database is a sub-schema database where data is stored in the JSON format. Basically the entire database is a large JSON tree with multiple nodes.

NoSQL provides non-relational data storage and retrieval data. It is not limited to fixed data and can store data as column-centered, document-based or graph-based. Graphical sources or a broad column, instead of table-like SQL data details [4].



Fig 1 :Realtime database

[src:

<https://console.firebase.google.com/project/fir-demo-project/features/develop>]

Authentication

Authorized users are provided with customized experience and the user's identity must be identified in order to grant them access to information [5]. The authentication process verifies the user's identity. User information is retrieved by calling the client device. Returned user information contains uid (unique ID), which is completely different and unique to all providers, and does not change its value to any specific authorized user. For authorized users it contains user variations (auth.uid) in the form of an object and other user-related data.



Fig 2 : Firebase authentication

[src:

<https://console.firebase.google.com/project/fir-demo-project/features/develop>]

Remote Config

Remote Config is an FCM service that uses the behavior and appearance of our app that can be changed and updated without the need to get the app first. If you use Remote Config, the functionality and appearance of our app can be controlled using the built-in default app values. After that, the Firebase console can be used later to overwrite the in-app values of all app users or parts of our user.

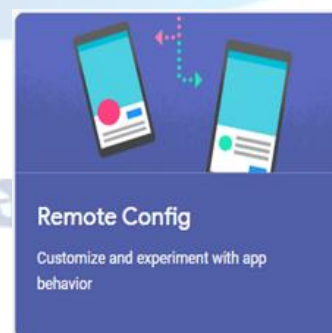


Fig 3: Remote config

[src:

<https://console.firebase.google.com/project/fir-demo-project/features/develop>]

Crash Reporting

The FCM crash reporting feature is used to identify errors in the application. When errors occur in the application they are grouped into similar properties in stack format and are sorted out as soon as possible.

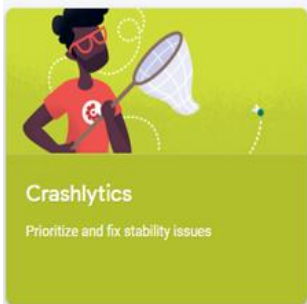


Fig 4: Crash Reporting

[src:

<https://console.firebase.google.com/project/fir-demo-project/features/develop>]

Cloud messaging

It includes versatile message targeting. Data sync is done in real-time and is available even when the application goes offline[6]

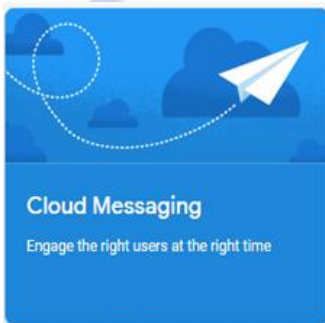


Fig 5: Cloud messaging

[src:

<https://console.firebase.google.com/project/fir-demo-project/features/develop>]

METHODOLOGY

An online messaging app that allows the user to communicate with other users in a fast and easy way. Both devices must have an active Internet connection for communication. As part of daily life, users of academic or technical life need to send and receive files.

Our communication system or messaging system aims to provide a platform for two users separated by a certain distance to communicate, online with the help of various Firebase tools. The system is divided into different modules, each module has a different function and works in sync with other modules without conflicts. Major modules are as follows:

Start Module

This is the first module to be launched when the program is launched. The system first checks

whether the user is online or not and then whether the user is connected or not. If both situations fail, this module is launched where the current user is introduced with the option to sign in to their account or register a new account.

Register Module

This module is introduced when a new user wishes to create a new account. This creation is managed on the server side by Firebase Authentication, based on the selected Login Process. When successfully registered, new entries are created in the database and the user is registered in the confirmation section of the Firebase console.

Identifier	Provider	Created	Signed in	User UID
+11911029466	Phone	Nov 23, 2020	Nov 23, 2020	Y1MfPgn4JR1snnC9kCagJTe63
+11950895606	Phone	Nov 19, 2020	Nov 19, 2020	Y1LaAZZDhVcBLwK0qgDh42
+118851411207	Phone	Nov 23, 2020	Nov 23, 2020	J6a71et1PRcbqHXEYFSPRIEF02

Fig 6: Authentication Tab of Firebase Console

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

Login Module

This module is introduced when a user wishes to sign in to the system. The user must enter the required verification and the Firebase system will check whether this evidence is correct or not. The password is stored as a hash that ensures system security.

Chat Module

This is the main program module that will handle the messaging function of the system. This module started when a user wishes to chat with other users on the network. When this module was introduced, the Firebase database reference is enhanced with various UI elements. At launch, we load any previous messages with a specific function that downloads previously sent messages. After starting everything, when users send a message then a check is made if the message sent is text, image or other file.



Fig 7: The Firebase Database with user entries and message entries

RESULTS

Upon completion of this program, we are left with a fully functional communication application that can send messages in real time. After the User number is verified with otp and recaptcha verification is done, he can create his profile with a picture and a name. Firebase services are important in the sense that those tools make the development of these applications more efficient and faster compared to building a traditional separate database using scripting language. The app is easy to use and usable, so using it is not a difficult task.



Fig 8 :Interface of chat window

DISCUSSION

The latter system will result in a real-time communication application that allows users to easily communicate. The app will have a login page where the user can register and sign in personally. The app's home page contains previous messages if available. A user can search for another user. Users can send and receive messages.

REFERENCES

- [1] <https://firebase.google.com/docs/android/setup>
- [2] Neil Smyth. Firebase Realtime Database, In:Firebase Essentials- Android Edition. [Online] Payload Media; 2017.p 163-171.
- [3] Neil Smyth. Getting Started with Firebase, In:Firebase Essentials- Android Edition. [Online] Payload Media; 2017.p 3-7.
- [4] <https://content.techgig.com/5-major-differences-between-sql-database-and-nosql-database/articleshow/79330492.cms>
- [5] Neil Smyth. Firebase User Authentication, In:Firebase Essentials- Android Edition. [Online] Payload Media; 2017.p 7-11.

- [6] Neil Smyth. Firebase Cloud Messaging, In:Firebase Essentials-Android Edition. [Online] Payload Media; 2017.p 225-235.