REVA UNIVERSITY
Bengaluru, India

SCHOOL OF COMPUTING AND INFORMATION TECHNOLOGY

A Project Report

on

CHAT APPLICATION USING A CLOUD SERVICE(FIREBASE)

Bachelor of Technology In

Computer Science & Technology

Submitted by

I.MOHAMMED JUNAID	*R14CS068*
MOHAMMED IBRAHIM	*R14CS109*
HARISH KUMAR M M	*R14CS063*
K NIROSH	*R14CS073*

Under the guidance of

PROF ASHARANI V

May 2017

Rukmini Knowledge park , Kattigenahalli, Yelahanka, Bengaluru-560064
www.reva.edu.in

*SCHOOL OF COMPUTING & **INFORMATION TECHNOLOGY***

CERTIFICATE

*Certified that the project work entitled **CHAT APPLICATION USING A CLOUD SERVICE (FIREBASE)** carried out under my guidance by **I MOHAMMED JUNAID**, **R14CS068**, **MOHAMMED IBRAHIM**, **R14CS109**, **HARISH KUMAR M M**, **R14CS063**, **K NIROSH**, **R14CS073**, bonafide students of REVA University during the academic year 2016-17, are submitting the mini project report under **Bachelor of Technology** degree in Computer Science & Technology during the academic year **2016-17**. The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.*

Signature with date

PROF Asharani v
Guide
School of Computing &
Information Technology

Signature with date

Dr. SunilKumarS.Manvi
Director
School of Computing &
Information Technology

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crowned our effort with success.

I express my sincere gratitude to **Dr. Sunilkumar S Manvi**, Principal, Reva University for providing facilities.

I wish to place on record my grateful thanks to **Dr. Mallikarjuna Shastry**, Head of the Department, School of Computing and Information Technology, Reva University, Bangalore for providing encouragement and guidance.

I would like to thank our Guide **Prof. Asharani V**, Professor, School of Computing and Information Technology, Reva University, Bangalore on her periodic inspection, time to time evaluation of the seminar and help to bring the seminar to the present form through proper guide.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF SCREENSHOTS

ABSTRACT

We have come across various chat applications to instantly communicate with people. We have used various types of chat application in web-based applications. All these chat applications support text messages to be sent between the users in the instant they press Enter key. We are implementing a similar chat application with a few key differences. The chat application we intend to do ensures privacy for chatting within individual organizations. This could be helpful to many of the professional institutions like schools, colleges and industrial units.

1. INTRODUCTION

We have come across various chat applications to instantly communicate with people. We have used various types of chat application in web-based applications. All these chat applications support text messages to be sent between the users in the instant they press Enter key. We are implementing a similar chat application with a few key differences. The chat application we intend to do ensures privacy for chatting within individual organizations. This could be helpful to many of the professional institutions like schools, colleges and industrial units.

Teleconferencing or chatting is a method of using technology to bring people and ideas “together”. Our project is an example of Chat Application using a Cloud Service(firebase). It is made up of two applications Chat application and Server application. To start chatting client should be connected to the Cloud Server(firebase). The messages sent by the send will be stored in a Realtime Database. From the database, the messages are transferred to all the clients.

1.1 PROBLEM DEFINITION

The Client Server model is a distributed model application structure the partitions tasks or workloads between the providers of a resource or service called Servers and service requesters called as Clients. Often Clients and Servers communicate over a Cloud Service. The Client-Server characteristic describes the relationship of co-operating programs in an application.

1.2 OBJECTIVE OF PROPOSED SYSTEM

- The main objective of this project is to build a chat application for chatting.
- To ensure privacy in Chatting.
- To make a private Chat Application for individual organizations.
- To implement a Cloud service in the chatting application.

2. LITERATURE SURVEY

The objective behind making this application was to bring the functionalities of a Network Service provider onto a mobile service. So while surveying as to on which platform or rather Operating System the project has to be implemented, we selected Android for the following reasons:

- Android is an Open Source Platform
- Supports Multi-functioning
- Provides rich tools to make interactive applications
- Downloading the Softwares required for making the application are absolutely free.

3. METHODOLOGY

3.1 PROTOCOLS USED BY FIREBASE

Currently FCM provides two connection server protocols: HTTP and XMPP. Your app server can use them separately or in tandem. XMPP messaging differs from HTTP messaging in the following ways:

- Upstream/Downstream messages

1. **HTTP:** Downstream only, cloud-to-device up to 4KB of data.
2. **XMPP:** Upstream and downstream (device-to-cloud, cloud-to-device), up to 4 KB of data.

- Messaging (synchronous or asynchronous)

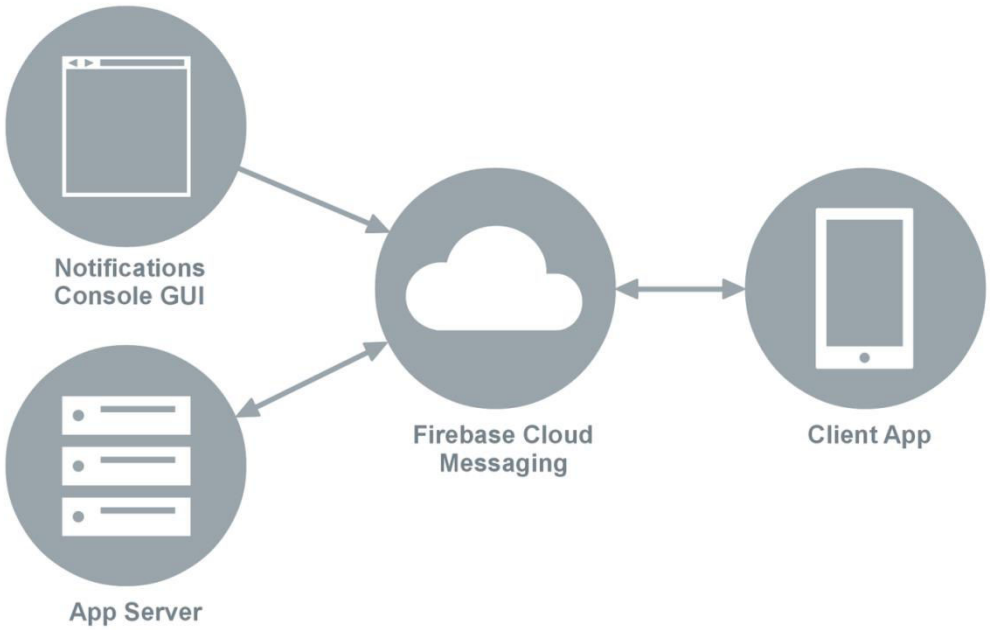
1. **HTTP:** Synchronous. App servers send messages as HTTP POST requests and wait for a response. This mechanism is synchronous and blocks the sender from sending another message until the response is received.
2. **XMPP:** Asynchronous. App servers send/receive messages to/from all their devices at full line speed over persistent XMPP connections. The XMPP connection server sends acknowledgment or failure notifications (in the form of special ACK and NACK JSON-encoded XMPP messages) asynchronously.

3.2 WORKING MODEL:

- In this project, we will be implementing an existing Cloud Service(FIREBASE by Google) which will become the Server for our Messaging Database.
- The server side of Firebase Cloud Messaging consists of 2 Components:

a) FCM connection servers provided by Google. These servers take messages from an app server and sends them to a client app running on a device. Google provide connection servers for HTTP and XMPP.

b) An app server that we implement in our environment. This app server sends data to a client via the chosen FCM connection server, using the appropriate XMPP or HTTP protocol.



3.1 Firebase working model

3.3 FIREBASE DATABASE:

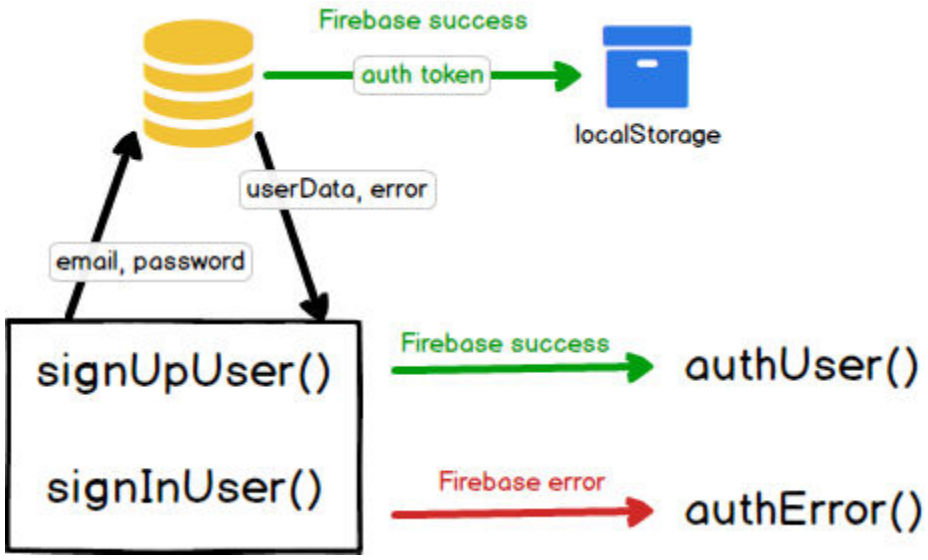
- When you build cross-platform apps with Android, all of our clients share one Real-time Database instance and automatically receive updates with the newest data
- Offline Firebase apps remain responsive even when offline because the Firebase Realtime Database SDK persists your data to disk. Once connectivity is reestablished, the client device receives any changes it missed, synchronizing it with the current server state.

3.3 Firebase Database

3.4 FIREBASE AUTHENTICATION:

- The app need to know the identity of a user. Knowing a user's identity allows our app to securely save user data in the cloud and provide the same personalized experience across all of the user's devices.
- The user can sign into the app using the following methods:

1. Email and password based authentication: Authenticate users with their email addresses and passwords.
2. Federated identity provider integration: Authenticate users by integrating with federated identity providers i.e.; Google Sign in.



3.4 Firebase Authentication

4.SYSTEM SPECIFICATIONS

4.1 HARDWARE REQUIREMENTS

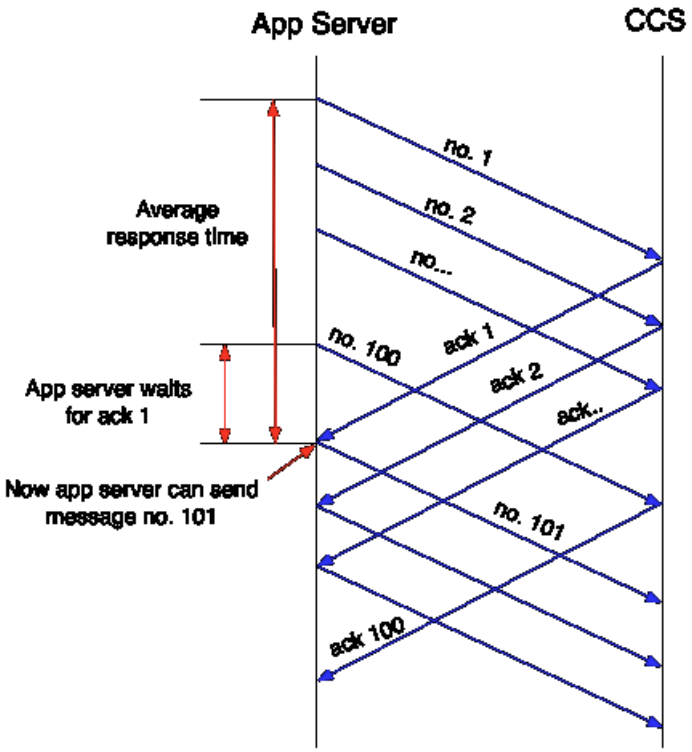
- In Hardware Requirements we require all those components which will provide us the platform for the development of the Project.
- Two or more machines where the application could be run and one server machine.
- The Minimum Hardware requirements for the development of this project are as follows:
- RAM: Minimum 2 GB
- Hard Disk: Minimum 5 GB
- Processor: CORE i3

4.2 SOFTWARE REQUIREMENTS

- Software's can be defined as programs which run on our computer. It acts as petrol in the vehicle. It provides a relationship between Human and a Computer. It is very Important to run software to function the computer. Various Software's are needed in this project for its development which are as follows:
- Operating System : Ubuntu 16.04
- Others: Android Studio

5. ANALYSIS OF DATA

When you build cross-platform apps with Android, all of our clients share one Real-time Database instance and automatically receive updates with the newest data. Offline Firebase apps remain responsive even when offline because the Firebase Realtime Database SDK persists your data to disk. Once connectivity is reestablished, the client device receives any changes it missed, synchronizing it with the current server state.



Every message sent to CCS receives either an ACK or a NACK response. Messages that haven't received one of these responses are considered pending. If the pending message count reaches 100, the app server should stop sending new messages and wait for CCS to acknowledge some of the existing pending messages as illustrated in figure. Conversely, to avoid overloading the app server, CCS stops sending if there are too many unacknowledged messages. Therefore, the app server should "ACK" upstream messages, received from the client application via CCS, as soon as possible to maintain a constant flow of incoming messages.

6. IMPLEMENTATION OF PROJECT

- The Firebase Real-time Database is a cloud-hosted database.
- When you build cross-platform apps with Android, all of our clients share one Real-time Database instance and automatically receive updates with the newest data
- Offline Firebase apps remain responsive even when offline because the Firebase Realtime Database SDK persists your data to disk. Once connectivity is reestablished, the client device receives any changes it missed, synchronizing it with the current server state.
- The Firebase Realtime Database can be accessed directly from a mobile device or web browser; there's no need for an application server. Security and data validation are available through the Firebase Realtime Database Security Rules, expression-based rules that are executed when data is read or written.
- The Chat application we intend to design would provide us with the following features:

    a. **Creating an Account**: The user gives his personal information, chooses a UserID and a password of his own choice. The database on the server side is updated after this.

    b. **Login and Logout:** These functions indicate the availability or non-availability of the user.

    c. **Handling Users:** It allows administrator to add or remove any user account in case of changes due to termination, resignation, violation of any rules or for some other reasons.

    d. **Instant Messaging:** This is a regular feature of any chat. This chat application would also provide this feature between multiple users.

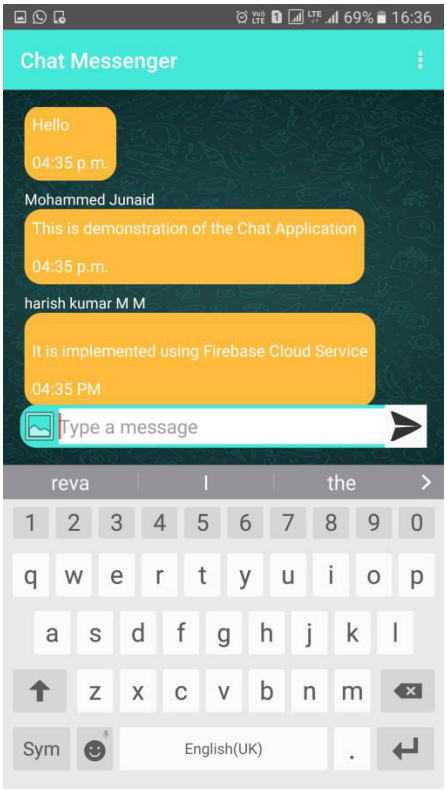
    e. **Transferring images:** The chat application would also have to transfer images drawn by the users between each other.

7. SCREENSHOTS

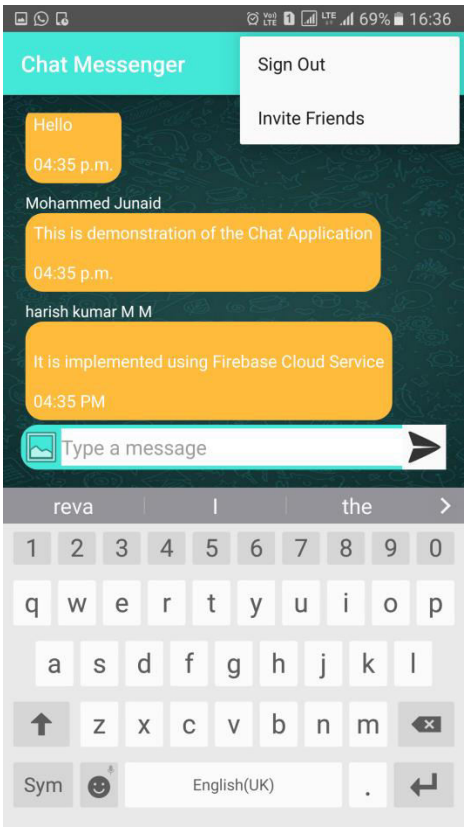
REVA UNIVERSITY

7.1 OPENING SCREEN

7.2 LOGIN SCREEN



7.3 CHAT SCREEN



7.4 SIGN-OUT SCREEN

8. CONCLUSION:

- This program enables two parties to Communicate with each other over a network which would serve many applications.
- As these days everything on the internet is going social as more and more people are becoming attached to sites like Facebook and twitter, the demand for social features is on rise. people want to be able to communicate with one and another, share content easily and feel as if they are part of content they are consuming.
- The chat application is so aimed that the people could have a better experience of chatting. It has the potential to attract more users to interact and connect.

9. FUTURE SCOPE

- This project can be updated to send audios, videos and documents in the future to be on par with other chatting applications.
- This application can also be updated with a Notice Board feature which will enable the organizations to circulate information safely and efficiently.

10. REFERENCES

- Google Firebase – https://firebase.google.com/docs/cloud -messaging/server
- https://firebase.google.com/docs/auth/
- https://firebase.google.com/docs/cloud-messaging/
- https://firebase.google.com/docs/database/