Developing Chat Application using Firebase

DeveshSharma[1], MadhavAgarawal[2], HradeshUpadhyay[3], G.Akilarasu[4]

[1-3]Department of Computer Science, Lovely Professional University, Punjab
[4]Department of Computer Science, Network and Security, Lovely Professional University, Punjab

---***---

Abstract - *Chat applications got one among the premier significant and mainstream applications on cell phones. It has the ability of trade instant messages, pictures and documents which it cost free for the clients to speak with one another. All messages should be ensured. The point of the paper is to propose chat application that gives End-to-End security that let securely trade private data with one another without agonizing over information and a continuous message sharing. Notwithstanding the assurance of capacity. A rundown of prerequisites to shape secure chat application is introduced during this paper and upheld these necessities, the machine was planned. The proposed chat application was contrasted and other mainstream applications upheld those prerequisites likewise on the grounds that it has been tried as an image for giving End-to-End security.*

Key Words: Secure chat application, Security, Android, Secure session,
Secure storage

1. INTRODUCTION

Remotely coordinating or Chatting is a strategy for utilizing Technology to unite individuals and thoughts notwithstanding of geological obstructions. The innovation has been accessible for quite a long time yet the acknowledgment it was very later. Our task is a chatting application that is utilizes firebase for data set administration which is gotten and simple to oversaw. Lately, chat applications have developed and rolled out a significant improvement in web-based media on account of their unmistakable highlights that draw in crowds. It gives ongoing informing and offers various administrations including, trade instant messages, pictures, records and so forth Besides, it upholds cross stages like Android and iOS. There are at present hundred huge number of clients cell phone are utilizing chat applications on month to month premise.

1.1 Types of Architecture

There are two kinds of architecture in those applications, customer worker and distributed organizations.

In a distributed organization, there is no focal worker and every client has his/her own information stockpiling. Unexpectedly, there are committed workers and customers in a customer worker organization and the information is put away on a focal worker our application is created on customer worker architecture.

1.2 Goal and Contribution

The goal of the paper is to develop a secure and fast chatting application that allows user to share text, media(pictures and videos) free of cost.

This paper is schematizing in distinctive sections as follows:

1.Introduction
2.available chatting applications
3.Proposed Architecture
4.Feedback Survey
5.Features
6.Algorithms
7.Conclusion
8.References

2. AVAILABLE CHATTING APPLICATIONS

In this segment, we momentarily present a considerable lot of famous chat applications in the portable market as indicated by security and protection concerns. Sadly, some chat applications are not public or open source makes it hard for assessed by the developer's local area, security specialists or analyst scholarly.

2.1 Whatsapp

WhatsApp is perhaps the most well known informing application, as of late empowered start to finish encryption for its 1 billion clients across all stages. WhatsApp utilizes part of a security convention created by Open Whisper System, so gives a security-confirmation code that can impart to a contact to guarantee that the discussion is encoded. It is hard to trust in WhatsApp application totally in light of the fact that the application isn't open source, making it hard to check the working cycle and match them with crafted by the encryption convention which was reported.

2.2 *Telegram*

Telegram is an open source texting administration empowers clients to send messages, photographs, recordings, stickers and documents . Telegram gives two methods of informing is standard chat and mystery chat. Ordinary chat is customer worker dependent on cloud-based informing, it doesn't give start to finish encryption, stores all messages on its workers and synchronizes with all client gadgets . More, nearby capacity isn't scrambled as a matter of course. Secret chat is customer gives start to finish encryption. As opposed to ordinary chat messages, messages that are sent in a mysterious chat must be gotten to on the gadget that has been started a mysterious chat and the gadget that has been acknowledged a mysterious chat they can't be gotten to on different gadgets. Messages sent inside secret chats can be erased whenever and can alternatively fall to pieces . Telegram utilizes its own cryptographic convention MT Proto, and has been scrutinized by a critical piece of the cryptographic local area about its security. The enlistment cycle of Telegram, Viber and WhatsApp rely upon SMS. SMS is shipped by means of Signaling System 7 (SS7) convention. The weakness lies in SS7 . Aggressors misused SS7 convention to login into casualty's record by capturing SMS messages . As a result of Telegram cloud-based, the aggressor abuses it and makes full control of the casualty account and can forestall him to go into his record. To make the record

2.3 Facebook

Facebook Messenger is a popular messaging service available for Android and iOS. It gives two methods of informing is ordinary chat and mystery discussions. Standard chat doesn't give start to finish encryption just secure correspondence by utilizing TLS, and it stores all messages on its workers. Secret discussions have a similar thought of Telegram secret chat

3. PROPOSED ARCHITECTURE

The proposed architecture is planned utilizing firebase . In customer side, when a client sets up the application, the client either chooses enrollment or sign in. At that point the qualifications or subtleties entered by client are shipped off firebase verification SDK these certification can be email, telephone no and so forth then firebase check the accreditation and accordingly tells if confirmation is fruitful or not.

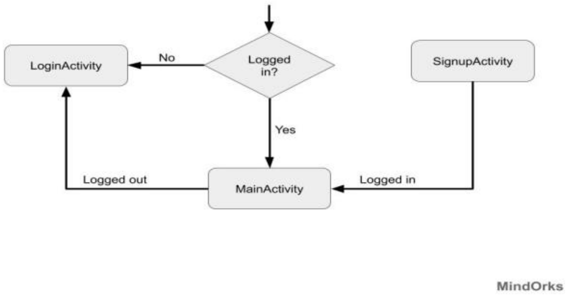


Fig -1: Login Actitvity

3.1 Registration

For signup client need to enter credentials . These credentials can be the client's telephone no and secret phrase, or an OAuth token from a unified character supplier. At that point, these credentials are passed to Firebase Authentication SDK. Backend administrations will at that point confirm those credentials and return a reaction to the customer.

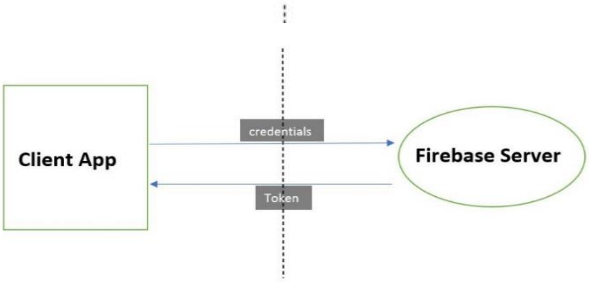


Fig -2: Registration Process

3.2 Login

After clients signup another client account is made and connected to the credentials for example the client name and secret phrase, telephone number or auth supplier data - the client signed in with, this new client is saved in firebase and can be utilized to distinguish client next time he login.

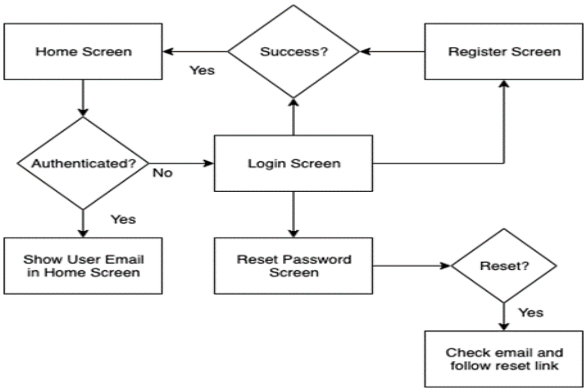


Fig -3: Login Process

3.3 Firebase Cloud Messaging

Firebase Cloud Messaging Platform (some time ago named as GCM) is a free portable notice administration by Google that empowers (outsider) application developers to send notices from GCM (Google Cloud Messaging) workers to their users.It will give you the privilege to illuminate your client continuously about the new email or other information accessible for sync. It chips away at the guideline of down streaming messages from FCM workers to client's application and upstream messages from client's applications to FCM workers. Firebase accompanies a great deal of new highlights alongside the GCM foundation.

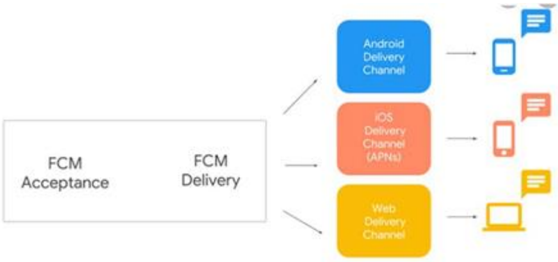


Fig -4: Message Process

3.4 Sending Friend Request

For sending a request to a companion with the understanding that the primary client knows telephone of the subsequent client because of the telephone number are a special for every client and the subsequent client ought to have effectively enlisted in the worker. The telephone number client one composed in search box is being kept an eye on the firebase if the telephone number is related with any client the public profile of the client is introduced before the client one when client one add client two then a pop warning is conveyed to client two that client one needs to add him he can acknowledge or get the greeting assuming client two decays the greeting of client one, client one won't send message to client two.

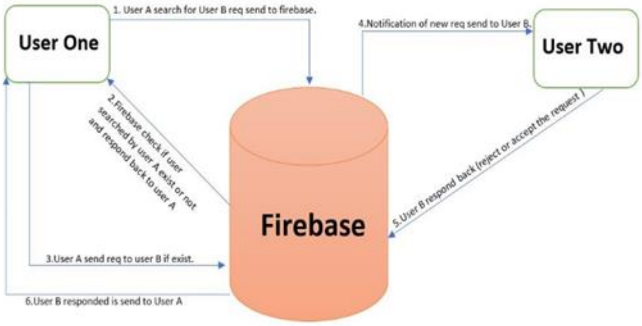

User One
1. User A search for User B req send to firebase.
4.Notification of new req send to User B.
User Two
2.Firebase check if user searched by user A exist or not and respond back to user A
Firebase
5.User B respond back (reject or accept the request)
3.User A send req to user B if exist.
6.User B responded is send to User A

Fig -5: Request Process

3.5 Exchanging messages

At the point when a message is composed, the application encodes the message utilizing XSalsa20 encryption calculation to scramble the message body and Poly1305 to figure a Message Authentication Code (MAC). Each message has its own separate key and nonce which brings better security for each single message in such finding one of the keys can't unscramble past messages. Subsequent to scrambling the message, it is scrambled again utilizing the beneficiary's meeting key then it is shipped off the firebase. After the message is gotten from FCM, the MAC of the encoded message is determined and contrasts it and the got MAC to confirm the trustworthiness of the message. In the event that the outcomes are not the equivalent, it is dismissed and doesn't show to the client else it is decoded by the sender meeting key. Then, the message body is confirmed in similar strides above. Presently the key and nonce to unscramble the message are known. The message is then unscrambled and put away in the nearby stockpiling and showed to the beneficiary.

In the event that the application is behind the scenes the message will be shown as a warning while if the beneficiary uses the application it will be shown in the chat window.

4. FEEDBACK SURVEY

Many chatting applications are emerging these days and are being used by people very effectively. In order to develop the application, the customer feedback about what is needed and what is existing in the current applications available. This survey was based on the features of the existing applications like WhatsApp, Telegram, Facebook Messenger. The following are some questions which were asked among a group of 50 people of different age group

4.1 Most liked features from the following:

- Unnamed messaging (receivers don’t know who the sender is)
- Disappearing messages (self-delete after viewing for a given/mentioned time)
- VoLTE (Voice over LTE)
- Group messaging
- Text message (SMS)

File sharing (e.g. sharing Word, Excel, PDF files)

- Emojis (Small digital icons)
- Stickers

4.2 Purpose of using.

- Texting
- File sharing
- Image sharing

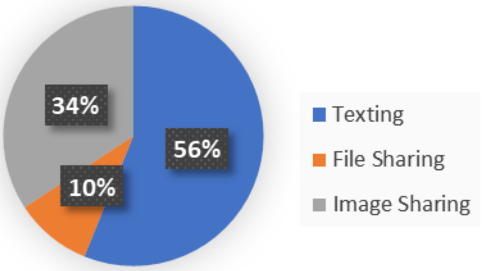


Chart -1: Purpose of Using App

4.3 Time Spent per day

- Less than 1 hour
- 1-3 hours
- 3-5 hours
- More than 5 hours

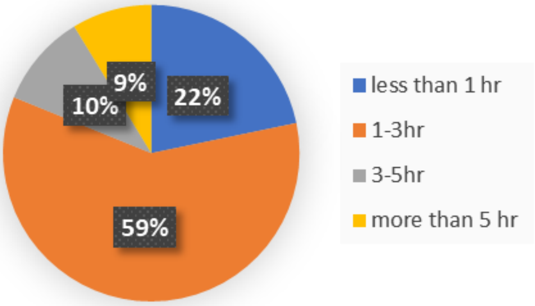


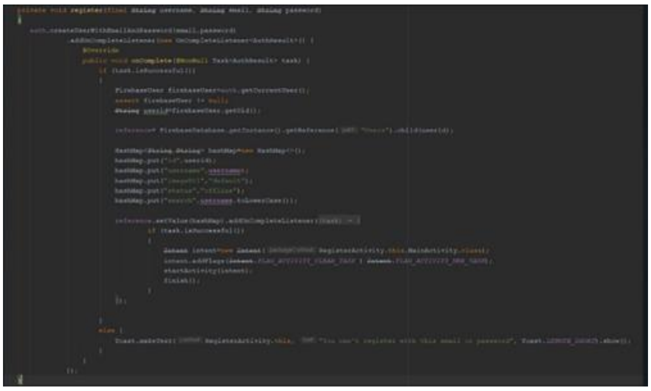
Chart -2: Time App Used

5. ALGORITHMS USED

There are certain algorithms which are been used to develop the application, which includes.

5.1 Sign In

After successful signup user credentials are stored in firebase and after that user can be able to sign up using those credentials.

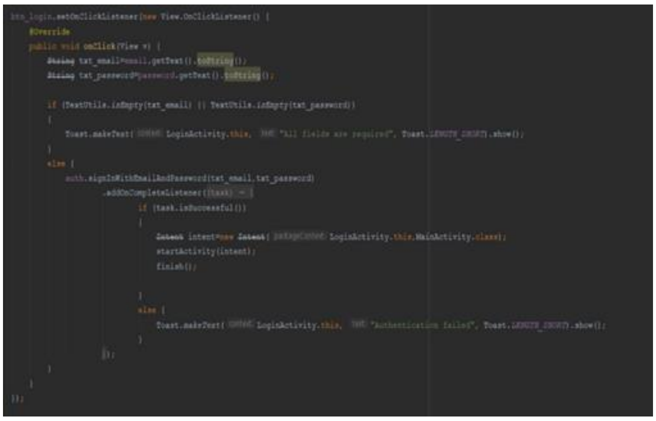


```
private void register(final String username, String email, String password)
{
    auth.createUserWithEmailAndPassword(email,password)
            .addOnCompleteListener(new OnCompleteListener<AuthResult>() {
                @Override
                public void onComplete(@NonNull Task<AuthResult> task) {
                    if (task.isSuccessful())
                    {
                        FirebaseUser firebaseUser=auth.getCurrentUser();
                        assert firebaseUser != null;
                        String userid=firebaseUser.getUid();

                        reference= FirebaseDatabase.getInstance().getReference( path: "Users").child(userid);

                        HashMap<String,String> hashMap=new HashMap<>();
                        hashMap.put("id",userid);
                        hashMap.put("username",username);
                        hashMap.put("imageURL","default");
                        hashMap.put("status","offline");
                        hashMap.put("search",username.toLowerCase());

                        reference.setValue(hashMap).addOnCompleteListener((task1) -> {
                            if (task1.isSuccessful())
                            {
                                Intent intent=new Intent( packageContext: RegisterActivity.this,MainActivity.class);
                                intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK | Intent.FLAG_ACTIVITY_NEW_TASK);
                                startActivity(intent);
                                finish();
                            }
                        });


                    }
                    else {
                        Toast.makeText( context: RegisterActivity.this, text: "You can't register with this email or password", Toast.LENGTH_SHORT).show();
                    }
                }
            });
}
```

Fig -6: Sign In Algorithm

5.2 Authentication

Most of the application requires the identity of the user which will help making the data of the user safer and more secured in a cloud. Firebase provides backend, SDK and ready to use libraries which help the developer to provide authentications effortlessly.



```
btn_login.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String txt_email=email.getText().toString();
        String txt_password=password.getText().toString();

        if (TextUtils.isEmpty(txt_email) || TextUtils.isEmpty(txt_password))
        {
            Toast.makeText( context: LoginActivity.this, text: "All fields are required", Toast.LENGTH_SHORT).show();
        }
        else {
            auth.signInWithEmailAndPassword(txt_email,txt_password)
                    .addOnCompleteListener((task) -> {
                        if (task.isSuccessful())
                        {
                            Intent intent=new Intent( packageContext: LoginActivity.this,MainActivity.class);
                            startActivity(intent);
                            finish();

                        }
                        else {
                            Toast.makeText( context: LoginActivity.this, text: "Authentication failed", Toast.LENGTH_SHORT).show();
                        }
                    });
        }
    }
});
```

Fig -7: Authentication Algorithm

6. CONCLUSIONS

The final application will allow user to connect and communicate in real time with ease. The application has a sign up by which user register them on the server after that user can search other user by their phone numbers and emails also and send them request and communicate with them.

But There is always some place for enhancements in any software application, however good and efficient the application may be right noe the application only providing messaging and image sharing features but in future we are planning for adding more features like:

1. Video calling
2. Voice calling

3. Messages delete after some time.

REFERENCES

[1]. Shao Guo-Hong, Application Development Research Based on Android Platform,2014 7th International Conference on Intelligent Computation Technology and Automation, 08 January 2015

[2]. S Karthick, Android security issues and solutions, 2017 International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), 13 July 2017

[3]. Pravin Auti, Sangam Mahale, Vikram Zanjad, Madhuri Dangat, n.d. An Android Based Global Chat Application,pp. 1-2.

[4]. S, A. K., n.d. Mastering Firebase for Android Development: Build real-time, scalable, and cloud-enabled Android apps with Firebase. s.l.: s.n

[5]. Abhinav Kathuria et al, Challenges in Android Application Development: A Case Study, Vol.4 Issue.5, May- 2015, pg. 294-299

[6]. Nikhil M. Dongre, Nikhil M. Dongre, Journal of Computer Engineering (IOSR-JCE), Volume 19, Issue 2, Ver. I (Mar.-Apr. 2017), PP 65-77

[7]. Javed Ahmad Shaheen et al, Android OS with its Architecture and Android Application with Dalvik Virtual Machine Review, International Journal of Multimedia and Ubiquitous Engineering Vol. 12, No. 7 (2017), pp. 19-30

[8]. Sajid Nabi Khan, Ikhlaq Ul Firdous, Review on Android App Security, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 7, Issue 4, April 2017