Social Networking Application using FaceNet and Firebase

A PROJECT REPORT

Submitted by

K.Mohanakrishnan [RA1711003010371]

Gautham Vijayaraj [RA1711003010381]

Under the Guidance of

Mr. A. M. J. Muthu Kumaran

(Associate Professor, Department of Computer Science and Engineering)

In partial fulfillment of the requirements for the degree of

BACHELOR OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR- 603 203

MAY 2021

ACKNOWLEDGEMENT

We express our humble gratitude to **C. Muthamizhchelvan**, Vice Chancellor (I/C), SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support.

We extend our sincere thanks to **Dr. Revathi Venkataraman**, Professor & Chairperson, School of Computing, SRM Institute of Science and Technology, for his invaluable support.

We wish to thank **Dr.B.Amutha**, Professor & Head, Department of Computer Science and Engineering, SRM Institute of Science and Technology, for her valuable suggestions and encouragement throughout the period of the project work.

We are extremely grateful to our Academic Advisor **Dr.P.Vishalakshi,** Assistant Professor, Department of Computer Science and Engineering, SRM Institute of Science and Technology, for their great support at all the stages of project work. We would like to convey our thanks to our Panel Head, **Dr.A.Murugan,** Assistant Professor, Department of Computer Science and Engineering, SRM Institute of Science and Technology, for his inputs during the project reviews.

We register our immeasurable thanks to our Faculty Advisor**, Dr.S.Saravanan,** Assistant Professor, Department of Computer Science and Engineering, SRM Institute of Science and Technology, for leading and helping us to complete our course.

Our inexpressible respect and thanks to my guide, **A.M.J.Muthu Kumaran,** Assistant Professor, Department of Computer Science and Engineering, SRM Institute of Science and Technology, for providing me an opportunity to pursue my project under his/her mentorship. He provided me the freedom and support to explore the research topics of my interest. Her passion for solving the real problems and making a difference in the world has always been inspiring.

We sincerely thank staff and students of the Computer Science and Engineering Department, SRM Institute of Science and Technology, for their help during my research. Finally, we would like to thank my parents, our family members and our friends for their unconditional love, constant support and encouragement.

K.Mohanakrishnan

Gautham Vijayaraj

SRM Institute of Science & Technology
Own Work* Declaration Form

This sheet must be filled in (each box ticked to show that the condition has been met). It must be signed and dated along with your student registration number and included with all assignments you submit – work will not be marked unless this is done.

To be completed by the student for all assessments

Degree/ Course	**:**	B.Tech, CSE
Student Name	**:**	K.Mohanakrishnan, Gautham Vijayaraj
Registration Number	**:**	RA1711003010371, RA1711003010381
Title of Work	**:**	Social Networking Application using FaceNet and Firebase

We hereby certify that this assessment compiles with the University's Rules and Regulations relating to Academic misconduct and plagiarism**, as listed in the University Website, Regulations, and the Education Committee guidelines.

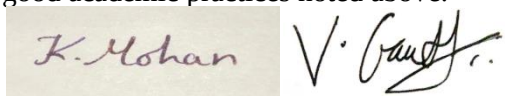
We confirm that all the work contained in this assessment is our own except where indicated, and that We have met the following conditions:

- Clearly references / listed all sources as appropriate
- Referenced and put in inverted commas all quoted text (from books, web, etc)
- Given the sources of all pictures, data etc. that are not my own
- Not made any use of the report(s) or essay(s) of any other student(s) either past or present
- Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources)
- Compiled with any other plagiarism criteria specified in the Course handbook / University website

I understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

DECLARATION:

I am aware of and understand the University's policy on Academic misconduct and plagiarism and I certify that this assessment is our own work, except where indicated by referring, and that I have followed the good academic practices noted above.

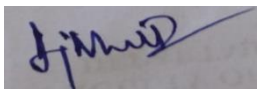


If you are working in a group, please write your registration numbers and sign with the date for every student in your group.

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY KATTANKULATHUR – 603 203

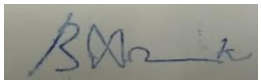
BONAFIDE CERTIFICATE

Certified that this B.Tech project report titled "**Social Networking Application using FaceNet and Firebase"** is the bonafide work of **Mr.K.Mohanakrishnan and Mr. Gautham Vijayaraj** who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion for this or any other candidate.



<<Signature of the Supervisor>>

Mr. A.M.J.Muthu Kumaran
SUPERVISOR
Assistant Professor
Department of Computer Science and Engineering



<<Signature>>

Dr. B. AMUTHA
HEAD OF THE DEPARTMENT
Department of Computer Science and Engineering

Signature of the External Examiner

Signature of the External Examiner

ABSTRACT

As we all know most of our social lives are controlled by social networking sites. The grouping of individuals and organizations together in order to share thoughts, activities, interests, etc refers to as social networking. There are countless web based social networking services that are in existence. But most of these sites are vulnerable to our personal information being exploited. The goal of this project is to explore and spearhead the use of face authentication in social media applications, which we believe will be a great addition to the already existing security features for social media platforms. Thus here we present our own social media platform(JC) that is authenticated using the FaceNet(Face Authentication) and made using the Google's Firebase platform.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF DIAGRAMS

LIST OF SYMBOLS AND ABBREVIATIONS

API	-	Application Programming Interface
REST	-	Representational State Transfer
CNN	-	Convolution Neural Network

INTRODUCTION

Nowadays face recognition has been gaining a lot of attention from researchers in the field of pattern recognition, computer vision groups, and biometrics. The following topics such as Anatomy of SOAP Security Performance from [2] and Social media security from [3] gives us reason to under the need for new security measures and also our own approach using the Face recognition techniques. There are many forensic applications, and security measures that have the need for technologies related to face recognition. It is very open to understanding the day to day importance of face detection systems. When taking into consideration of the various types of biometrics, face recognition and detection system can be considered as the most accurate one. In [4], Ashu Kumar et al[4] have presented a survey on the existing face detection techniques. With their help we have settled on the use of an API to make the face authentication process efficient and at the same time more accurate.[9] and [8] specifies the use of AI and API in with face authentication and also helps cement the process of using API in our project also. One of our main basis of belief that an application where face authentication can be a viable option can be brought to light with the help of [8]. The system proposed in [8] had been created so as to run combine computation on mobile devices and the advantages of cloud computing. The computations on mobile devices (on loading) will execute the augmented reality and module for face detection which is then used to Communicate with the users. With the help of Google App Engine (GAE) and Face.com services other computations run on cloud servers (offloading). The system found in [8] had been designed into three modules which are augmented reality, face recognition on cloud server using API, and face detection on Android mobile device as a result is system is gives inspiration for us to follow what we have done with ours. [8] Thus with the help of [4],[8] and [9]

we have come to the conclusion that face authentication using an assisted API.

LITERATURE SURVEY

SNO	TITLE	YEAR	AUTHOR	ADVANTAGE	DISADVANTAGE
1	Web Services Attacks and Security – A Systematic Literature Review	2016	Varsha R.Mouli K.P.Jevitha	The objective of this paper is to present a systematic review on the studies of web service security. There is no single solution for mitigating all the attacks on web services.	Web Services allow applications to communicate with each other independent of platform and/or language. They are prone to attacks in the form of Denial-Of-Service, SQL injection and spoofing, making implementation of web service security vital.
2	Performance of Web Services Security	2005	Hongbin Liu Shrideep Pallickara Geoffrey Fox	The review paper illustrates about the performances of different types of cryptographic algorithms that are commonly used for securing and processing the data found in SOAP documents	It talks about the advantages and disadvantages of AES, DES and other symmetric algorithms. However, this paper does seem to address the application for cryptography systems in social media.
3	Social media security and trust worthiness: Overview and new direction	2016	Zhiyong Zhang Brij B.Gupta	The survey investigates social media security. There is a wide range of information available in this paper but the key areas that we have focused on include the various kinds of attacks on social media. To be more specific the attacks on peer-to-peer messages	The required information is hard to find since this paper address the trustworthiness of the social media sites and only small part is allocated to the basic cryptographic algorithms used.
4	Face detection techniques: a review	2018	Ashu Kumar Amandeep Kaur Munish Kumar	The review paper gives an over view of the implementation, the need, advantages and disadvantages of the face detection. It also discusses about the various kind of implementation this can have in the field of computer science(cyber security related applications).	This review paper only provides the fundamental technique of face detection but we need to find a more advanced and efficient method.
5	A Modified Symmetric Key Cryptography Method for Secure Data Transmissions	2017	Dr. M. Ilayaraja Dr. K.Shankar Dr. G. Devika	In this paper a modified Caesar cipher symmetric key encryption method is developed to convert the actual message into secret information using mathematical principles.	The proposed technique produces the ciphertext includes more number of special characters. The result of the proposed method proves that it is very difficult to identify the original message from the encrypted message.
6	A Comprehensive Evaluation of Cryptographic Algorithms: DES, 3DES, AES, RSA and Blowfish	2015	Priyadarshini Patila Prashant Narayankarb Narayan D G c Meena S Md	In the paper, they have implemented and analyzed in detail cost and performance of popularly used cryptographic algorithms DES, 3DES, AES, RSA and blowfish to show an overall performance analysis, unlike only theoretical comparisons. This in return has helped us identify the best approach for choosing/creating a cryptographic algorithm.	The evaluation is only done for some of the cryptographic algorithm.

7	Real-time Communication Application Based on Android Using Google Firebase	2018	Nilanjan Chatterjee Souvik Chakraborty Aakash Decosta Dr. Asoke Nath	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.	The limitation is that firebase uses static hosting and API's for hosting. This limits the server side control and customization.
8	Face Recognition for Social Media With Mobile Cloud Computing	2013	Prasetyawidi Indrawan Slamet Budiyatno Nur Muhammad Ridho Riri Fitri Sari	The implementation of a real-time face recognition system and user interface on Android mobile device. also an augmented reality application was created in this project to provide information about the recognizable faces viewed by the user.	The paper is old, so it uses a equally old system of face recognition and the accuracy the systems in place is only 85%.
9	DeepFace: Closing the Gap to Human-Level Performance in Face Verification	2014	Yaniv Taigman Ming Yang Marc'Aurelio Ranzato Lior Wolf	The proposed face representation is learned from a large collection of photos from Facebook, referred to as the Social Face Classification (SFC) dataset. This dataset contains about 4.4 million labeled faces from 4,030 people each with 800 to 1200 faces.	Here the deepface system is only used as a reference to understand the new REST face recognition API's. So we will not be using deepface in our own application.
10	An Encrypted Messaging Application with Multi Fragmented Caesar Encryption Method between Mobile Devices	2017	Levent Gokrem Omer Faruk	By using the Multi Fragmented Caesar Encryption algorithm the password for logging in to the application the level of security that is much greater than a normal Caesar cipher algorithm.	However, the encrypted text can be analyzed with Kasiski analysis method.
11	Real-time Image classification using Tensorflow Lite and Flutter	2020	Marcos Carlomagno	TensorFlow Lite is designed to be lightweight, with a small binary size and fast initialization. It is also compatible with a variety of platforms, including Android and IOS. And to enhance the mobile experience, it is optimized for mobile devices with improved load times and hardware acceleration.	The main disadvantage is existence many other kinds of simple ant effective object recognition ML services. So this can only used for understanding purposes
12	FaceNet Architecture	2020	Milind Deore	In conventional CNN, convolution is done on an image with a given filter to construct a correlation statistics, layer-by-layer and then clustering these neurons that are highly correlated as an output. Important point to note is the correlation is local to the image patch and the highest correlation exist in the earlier layers of the network and hence large filter size and early pooling would reduce the important information hidden in the image patch.	Compared to pre-existing types of CNN systems this is very new and thus the use and scope of such a system is not yet clearly defined.
13	Face Recognition Authentication using Flutter and Tensorflow Lite	2020	Marcos Carlomagno	This paper highlights the use of an face recognition using the FaceNet and Firebase. This articles main goal is to build a face recognition app.	However, one the main things this article forgets is the evaluation of the security of this app.

INFERENCE FROM THE SURVEY

- Most the security measure taken for social media still depend on the only authentication using cryptographic algorithm. We can take note on stagnation of security measures for messaging in specific.
- We can also take note how vulnerable the state of the privacy on the social media space.
- The use and need for the face recognition in authentication system is highlighted.
- Also the working and types of face detection/systems are talked about and this helps us narrow down to the use of a REST API for our project.
- New type of methods for protection of social media data is not taken a concern and the papers for it are very old and outdated.
- We can see that using less complex methods we can achieve a different and relatively more secure system of using apps in general.
- The use of platforms like Firebase, flutter and TensorFlow allow for the easy creation of various apps and medium to test ideas. These ideas can then also be taken public using the platforms.

OBJECTIVE OF THE PROJECT

- Objectives 1:

  To create a facial detection system using firebases ML Kit.
- Objective 2:

  To create the facial recognition system using the MobileFaceNet model.
- Objective 3:

  To create a fully working social media application.

ARCHITECTURE DIAGRAM

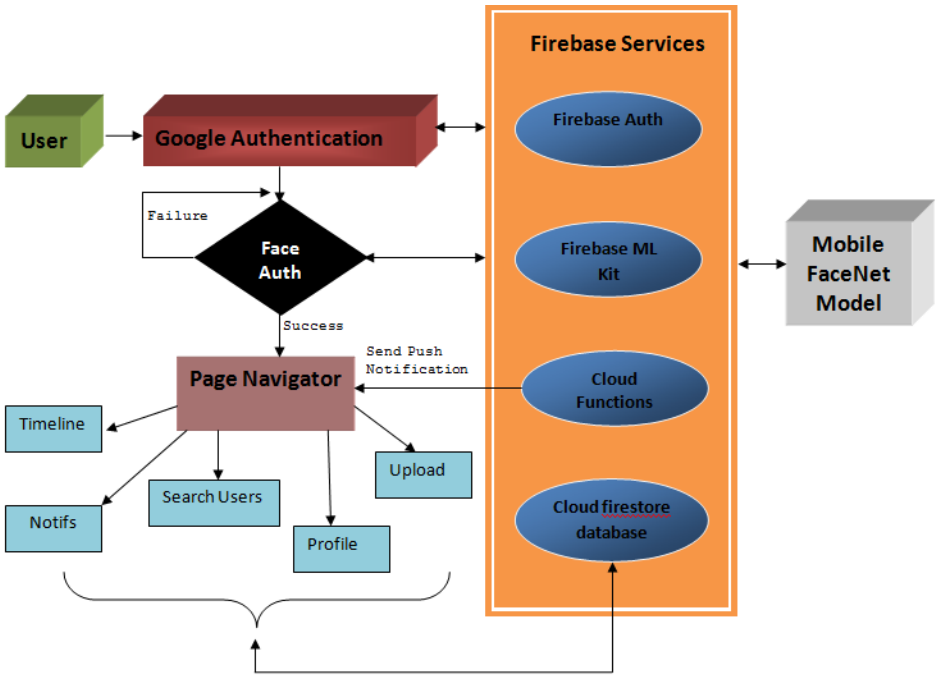


Fig No3.1 :Architecture Diagram of the application

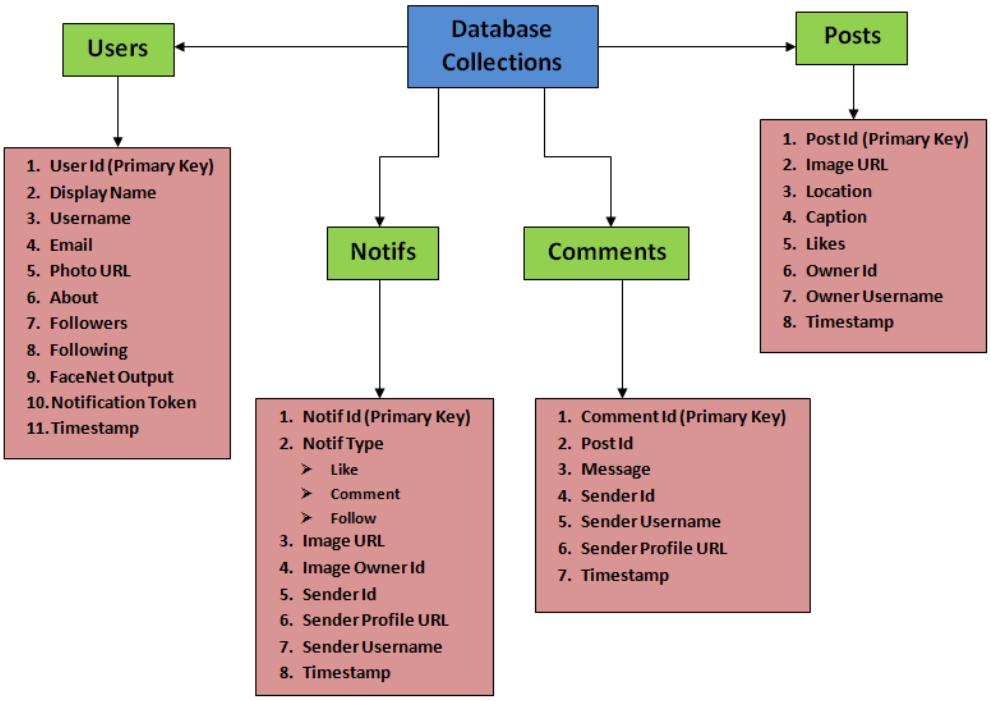


Fig No.3.2 :Block diagram of Cloud-Firestore Database

FACE AUTHENTICATION

To complete the primary objective we are using the face detection API that's available within the ML kit provided by Google. The biggest reason for our preference of this particular API is thanks to our whole application being run on the firebase and flutter platforms. The ML Kit that are often available in can be easily linked with firebase and used.

The features available for face detection in ML Kit are as follows:

Face tracking allows for face detection for even video sequences. Faces that appear during a video for any length of time are tracked from the frame to border. This suggests that a face that is detected in consecutive video frames are often identified as being an equivalent person.

A **landmark** can be the eyes, dimples, etc. or a point of interest on the face. Using the kit we can detect these and make use of them.

A **contour** is a collection of common points that may form a facial feature. These contours can be found and made of use with the help of the ML Kit.

The Facial Recognition method we are going to use is FaceNet. FaceNet is a unified facial recognition system that is based on learning a Euclidean embedding per image using Convolution Neural Networks that consists of 22 layers. The CNN that is used by FaceNet is trained so as to optimize the embedding that will then directly replace the intermediate bottleneck layer that is found in the previous deep learning approaches (Schroff, Kalenichenko & Philbin 2015). The FaceNet system can be recommended since the model has extensively been pre-trained using the VGGFace22 dataset that consists of about 3.3 million images[11]. From [11] it was analyzed that FaceNet showed the best accuracy results on a comparison with the other methods. *Why FaceNet?* From [11] we find that FaceNet can be used to achieve a precision that is close to the 100%. According to the figure we can find that the second best technique is scikit-learn implementation of Fisherface using KNN which

gives a precision of 70.44%. Thus the results show that FaceNet clearly outperforms all other techniques where the minimum difference is about 29 percent points. Precision is here calculated by dividing the TPs with all positive predictions, whether they are correct or not, which mathematically is: $TP/(TP+FP)$.

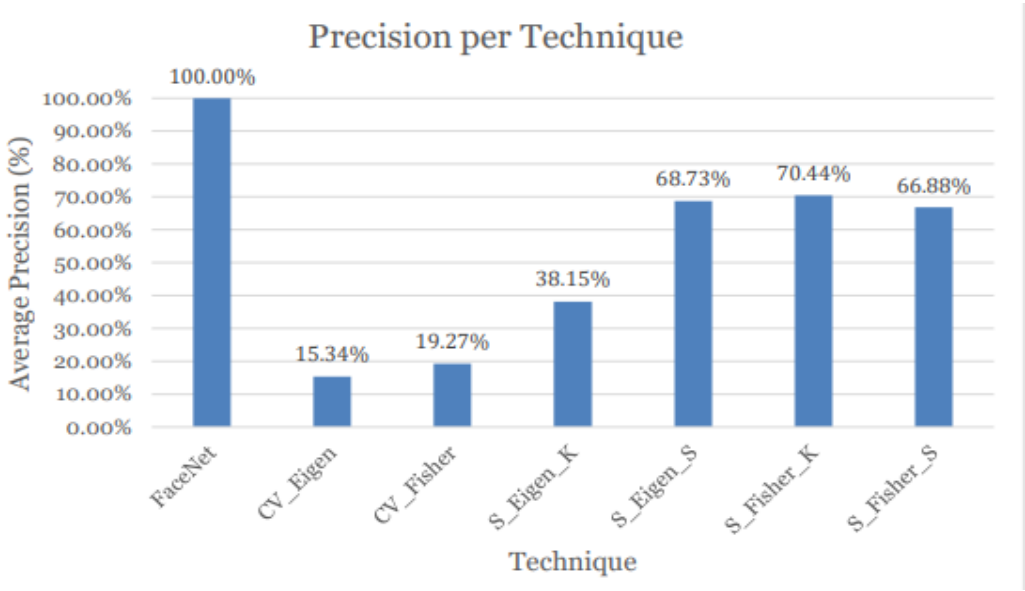


Fig No 3.3; Precision of FaceNet and other methods

MODULES

Our project contains two modules/phases and they are as follows:

- The face detection phase
- The triplet loss phase

6.1 The Face Detection Phase

TensorFlow provides a set of pre-trained models that are able to solve a plethora of machine learning related problems. All the models have been converted to be able work with TensorFlow and are ready to be used in applications. TensorFlow is designed to run various models efficiently on mobile devices and other embedded devices that have limited computing resources and memory.[12] Part of their efficiency is gained from using a special format to store the models. These models must be converted to the special format before TensorFlow can be able to implement them. The process of running data through a model to obtain recognitions is called inference. It requires a model, input data, and an interpreter.[12] Converting FaceNet.pb to FaceNe.tflite. The pre-trained FaceNet model is quantized, where the embedding size can be chosen as either 512 or 128. The software requirements for this process are just Python (at least version *3.4).*After that we need to install the tensor flow package using the pip install command. The FaceNet model can be converted from (.pb) to (.tflite) file by following the steps from [12]How is the .tflite file implemented in flutter? With Face Detection API found in the ML kit, you can detect faces that exist in an image, obtain the contours and identify key facial features of detected faces. It works very well in the preprocessing of images to detect zones to be cropped and then processed by the MobileFaceNet model. [14]The ML kits Face detection model detects existing human faces in the frames that the camera preview and the class Face contains the coordinates of the points that make up the frame around the face.

List<Face> faces = await _mlVisionService.getFacesFromImage(image);
(This command detects a list of images from the picture taken from the camera of our mobile.)

6.1 The Triplet Loss Phase

The detected face of the last frame is captured, cropped and then pre-processed to be processed by the MobileFaceNet model.
The MobileFaceNet model returns an output (array of numbers). If it's the sign up process, the application requests a name and a password, and later saves the three data (name, password and ML output). If it's the Sign In process, the application performs a search in the database comparing the Euclidean distance between the ML output of the image and the ML outputs stored for each user, the one that matches or is close enough (accuracy above the threshold) the application brings the data and requests a password.

$$d(\mathbf{p}, \mathbf{q}) = d(\mathbf{q}, \mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}.$$

Fig N 3.4:. Euclidian distance formula

IMPLEMENTATION AND WORKING

It works with two computer vision models working together, the Firebase ML vision model to perform the face detection and preprocessing in the image, and the MobileFaceNet model to process, classify and transform into a data structure ‘savable’ by a database (an array of numbers). The process summary is as follows:

Sign up

(1) The user clicks a photo.
(2) The ML models processes it and create an output in the form array of numbers that is stored in a database.
(3) Name and a password are then requested.

Sign in

(1) The user clicks a photo.
(2) The ML models processes it and create the required output.
(3) The output will be compared against the outputs already stored in the database (it compares by proximity the closest one it finds). As condition, the proximity has to be under the *threshold* (minimum distance), if overcomes it, it will process it as a non-existent user.
(4) If the user exists (face already processed) it requests the password for that user, validates and authenticates it.[13]

7.1. Frontend – (Flutter Framework)

Flutter is an open-source User Interface SDK made by Google. It is used for the development of IOS, Android, Windows, Mac, Linux, and the web applications from a single codebase. This is the platform we are going to use to design the front end of our application. The coding will be done in the dart language which was also developed by Google.

7.2. Backend – (Firebase and Node Js)

Firebase is also is a platform developed by Google for storing data and accessing using the Cloud Firestore Database. The primary authentication is

done through this platform for all the users to sign in. NodeJs is used for creating trigger functions in order to send push notifications and confirmation emails to all the users in real time.

7.3. Cloud Functions

Once the user signs in we want to program 2 functions. One function to send a welcome message to the registered email and another to send push notifications if that user receives any new notifications. Since we are creating a social media application the need is imminent and it is specified by the figure 3.2. Modules required for the functions:

- Firebase-functions
- Firebase-Admin
- Node-Mailer
- Cors

Now we need to go to this link: https://myaccount.google.com/lesssecureapp and enable Less Secure Apps to sign in from the respective Gmail account from which you want to send the automated message.

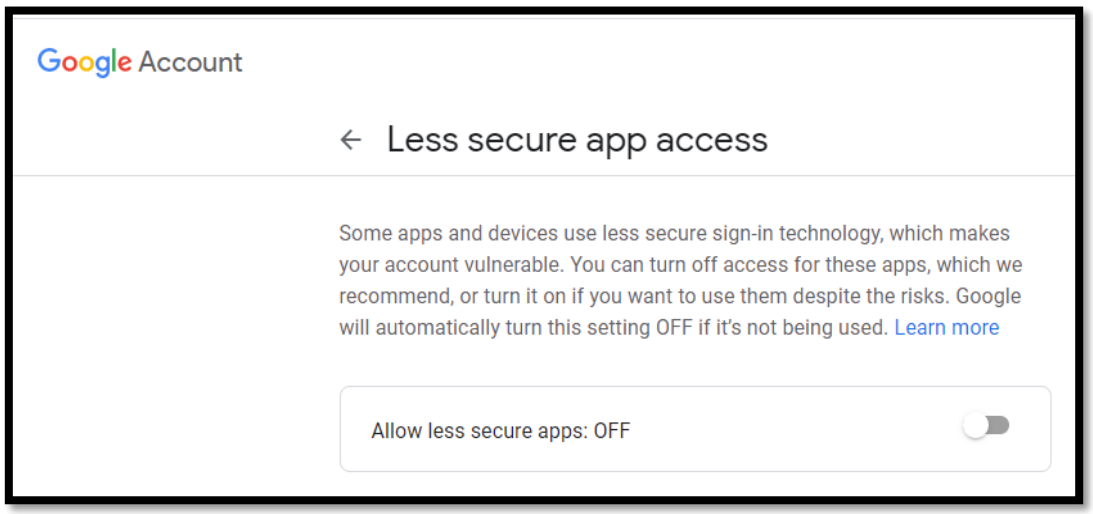


Fig N3.5:. Enabling of Less Secure Apps

```
let transporter = nodemailer.createTransport({
service: 'gmail',
auth: {
user: '<email address>',
pass: '<password>'  //your password
}
});
exports.sendMail = functions.https.onRequest((req, res) => {
cors(req, res, () => {
// getting destination email by query string
const dest = req.query.dest;
const mailOptions = {
from: "Gauthiii's Applications <gauthamcrazychicken1999@gmail.com>", //
to: dest,
subject: 'Welcome to the Joint-Club!', // email subject
html: `Dear User, Welcome to JC, <p>You have logged into the JC App`
};
// returning result
return transporter.sendMail(mailOptions, (erro, info) => {
if(erro){
return res.send(erro.toString());
}
return res.send('Email Sent');
});
});
});
```

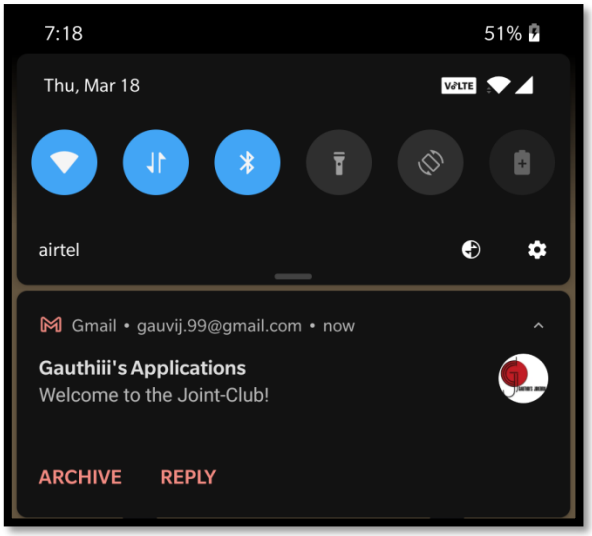


Fig N 3.6: Notifications sent after registration using the app

Nowin order to create a function for every notification the user gets this is what the code looks like :

```
exports.onCreateActivityFeedItem = functions.firestore
.document("/feed/{userId}/feedItems/{activityFeedItem}")
.onCreate(async (snapshot, context) => {
console.log("Activity Feed Item Created", snapshot.data());
    // 1) Get user connected to the feed
    const userId = context.params.userId;
    const userRef = admin.firestore().doc(`users/${userId}`);
    const doc = await userRef.get();
    // 2) Once we have user, check if they have a notification token; send
notification, if they have a token
    const androidNotificationToken = doc.data().androidNotificationToken;
    const createdActivityFeedItem = snapshot.data();
    if (androidNotificationToken) {
sendNotification(androidNotificationToken, createdActivityFeedItem);
```

```
    } else {
console.log("No token for user, cannot send notification");
    }
    function sendNotification(androidNotificationToken, activityFeedItem) {
      let body;
      // 3) switch body value based off of notification type
      switch (activityFeedItem.type) {
        case "comment":
          body = `${activityFeedItem.username} replied: ${
activityFeedItem.commentData
          }`;
          break;
        case "like":
          body = `${activityFeedItem.username} liked your post`;
          break;
        case "follow":
          body = `${activityFeedItem.username} started following you`;
          break;
        default:
          break;
      }
      // 4) Create message for push notification
      const message = {
        notification: { body },
        token: androidNotificationToken,
        data: { recipient: userId }
      };
      // 5) Send message with admin.messaging()
      admin
.messaging()
```

```
.send(message)
.then(response => {
        // Response is a message ID string
console.log("Successfully sent message", response);
      })
.catch(error => {
console.log("Error sending message", error);
      });
   }
 });
exports.onCreateAccount = functions.firestore
.document("/users/{userId}")
.onCreate(async (snapshot, context) => {
console.log("Account Created", snapshot.data());
   // 1) Get user connected to the feed
   const userId = context.params.userId;
   const userRef = admin.firestore().doc(`users/${userId}`);
   const doc = await userRef.get();
   // 2) Once we have user, check if they have a notification token; send
notification, if they have a token
   const androidNotificationToken = doc.data().androidNotificationToken;
   const createdAccountItem = snapshot.data();
   if (androidNotificationToken) {
sendNotification(androidNotificationToken, createdAccountItem);
   } else {
console.log("No token for user, cannot send notification");
   }
   function sendNotification(androidNotificationToken, activityFeedItem) {
     let body;
      body=`Welcome to JC ${activityFeedItem.displayName}`;
```

```
      // 4) Create message for push notification
      const message = {
        notification: { body },
        token: androidNotificationToken,
        data: { recipient: userId }
      };
      // 5) Send message with admin.messaging()
      admin
.messaging()
.send(message)
.then(response => {
         // Response is a message ID string
console.log("Successfully sent message", response);
        })
.catch(error => {
console.log("Error sending message", error);
        });
    }
  });
```

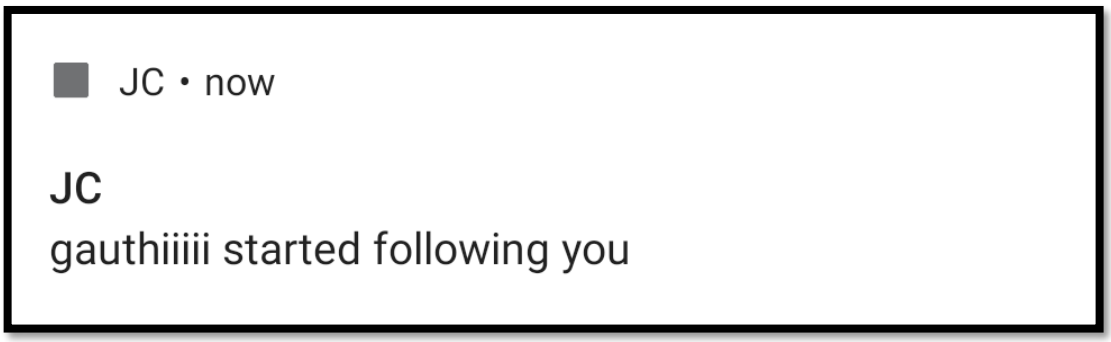


Fig No 3.7: Notifications while using the app

7.4. Messenger Feature

This Application wouldn't be a complete Social Networking App without the chatting feature so that's the next feature that is going to be focused on. The messages would be encrypted using the Triple DES algorithm and stored in the cloud firestore and few more additional collections in the database. The message which gets stored in the database looks like the figure 6,7 and the end output will be like the figure 8(both given below):

Fig No 3.8:Block diagram of Sending and Receiving and sending of messages

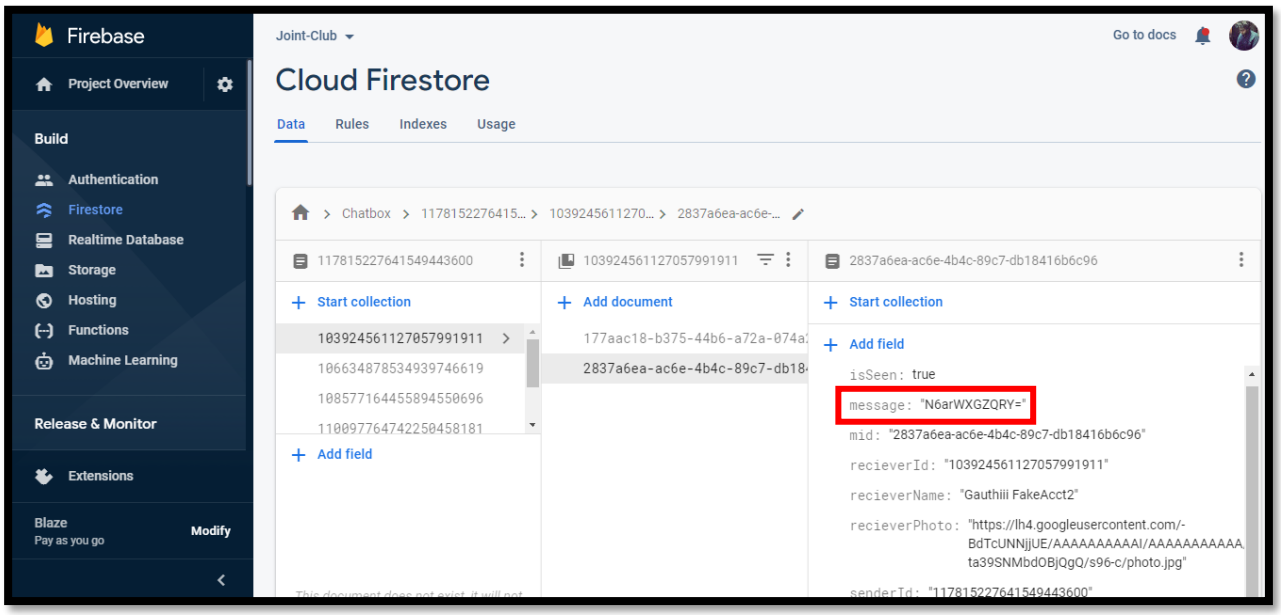


Fig No 3.9This is how the messages are stored in the Google Firestore

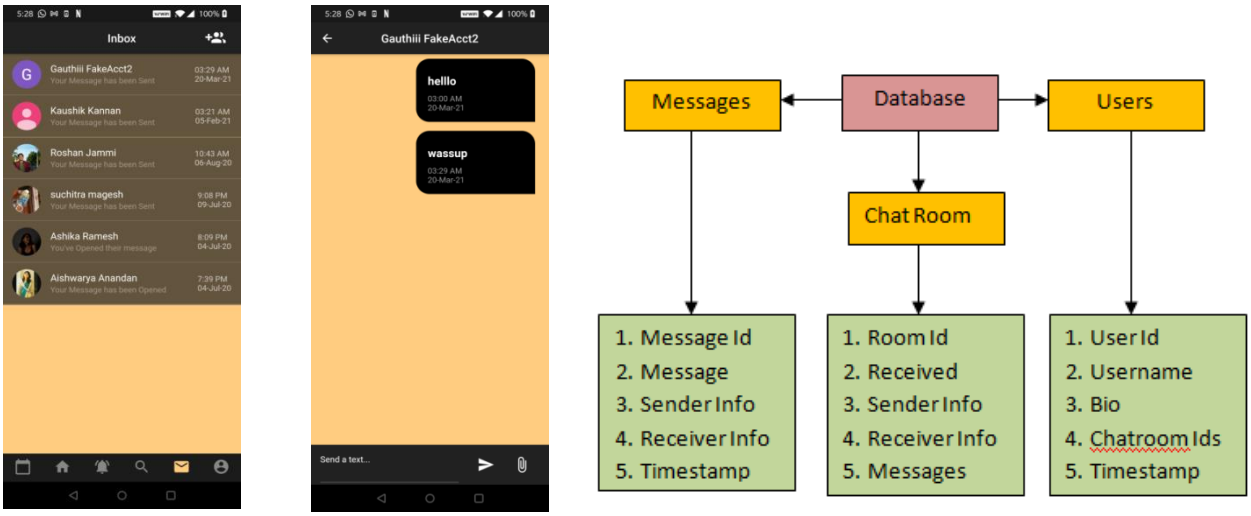


Fig No 3.10:Messaging in the app(left) and its underlying features(Block diagram to the right).

7.5. Software Requirements

The software requirements to create this application would be:

- Flutter
- Visual Studio Code (Any IDE)
- Android Studio
- NodeJs
- A registered Google Account
- A Firebase account to manage your Console
- Java SE Development kit (version 8 or above)
- A mobile device to test with the developer option enabled
- Git

7.6. Dependencies Used

A dependency is another package that your package needs in order to work. Dependencies can be installed using the (pubspec.yaml) file of every Flutter project folder. So the dependencies that we would need are:

- cupertino_icons
- firebase_messaging
- google_sign_in
- cloud_firestore
- firebase_storage
- image_picker
- firebase_auth
- path_provider
- uuid
- geolocator
- cached_network_image
- timeago
- tflite_flutter
- camera
- path
- firebase_ml_vision
- image
- flutter_local_notifications
- fluttertoast
- clippy_flutter

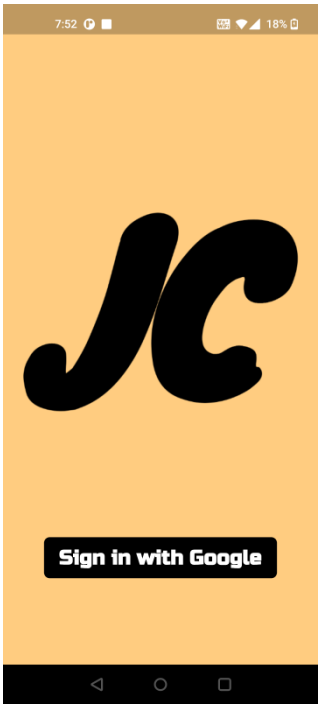
CONCLUSION

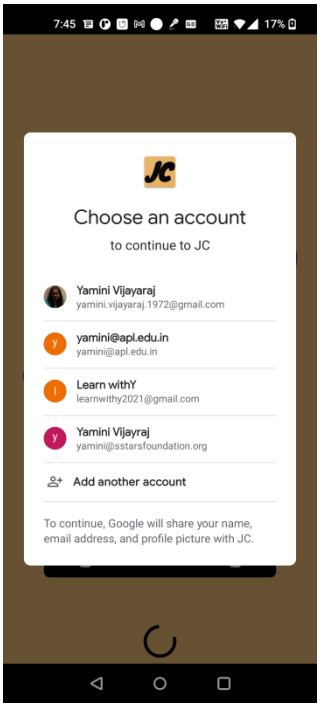
To sum up, JC App (Joint-Club – the name of the app) is different from the rest of the following apps because this app has Face-Recognition enabled authentication. Even if there are other social media apps which facial recognition, the JC uses FaceNet Model for process, classify and transform into a data structure ‘savable’ by a database (an array of numbers). The next additional feature that is implemented is regarding the security of the conversations. Data leaks are very common nowadays and the Chat-Screen is designed in such a way that the conversations can neither be saved by taking Screenshots nor Screen-Recorded by the users. As of now the messages are encrypted using the Triple DES Algorithms. However the application would be updated once new ideas are brainstormed to make this application more efficient and secure.

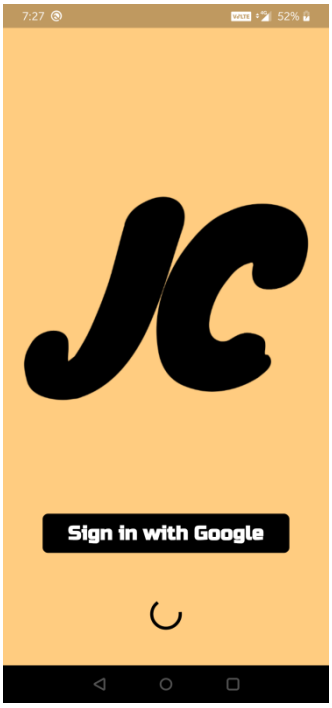
Fig No 3.11:The application can be tested from the apk file that can be downloaded from the QR Code.

APPENDIX

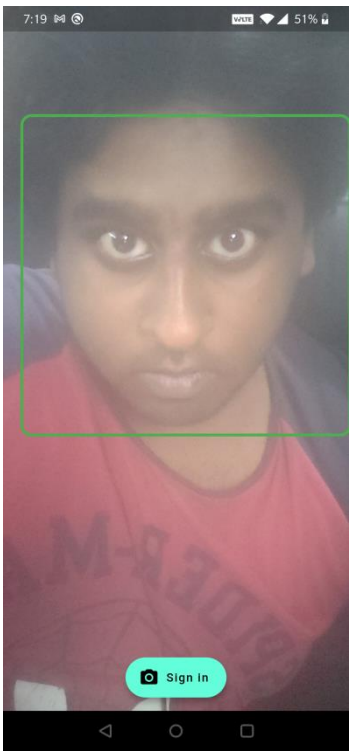
Login Screen

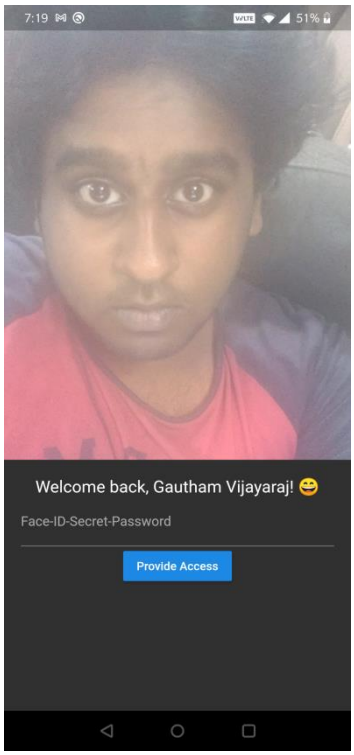


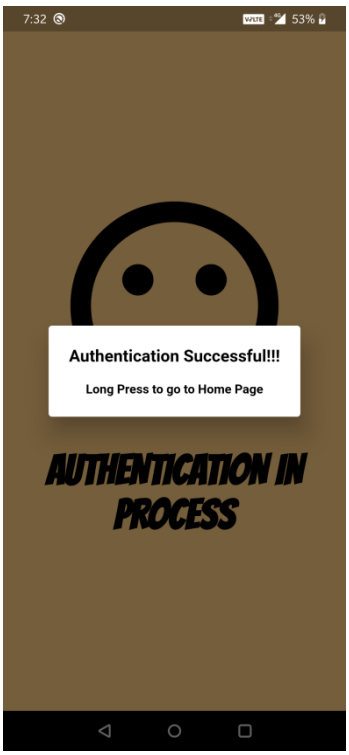




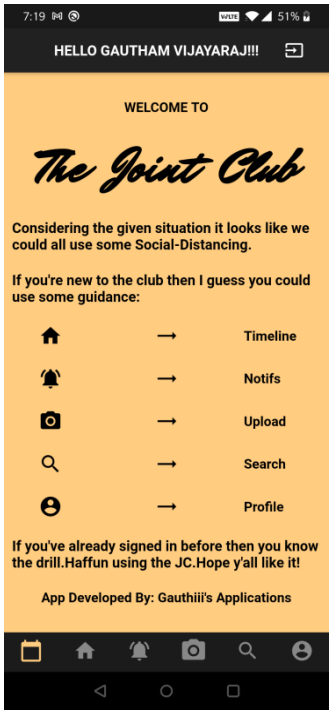
Face Authentication Sign In



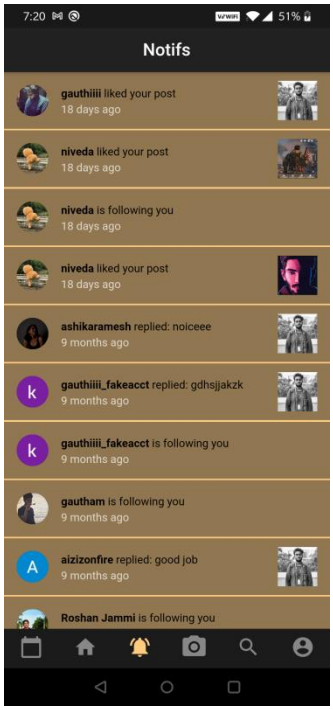




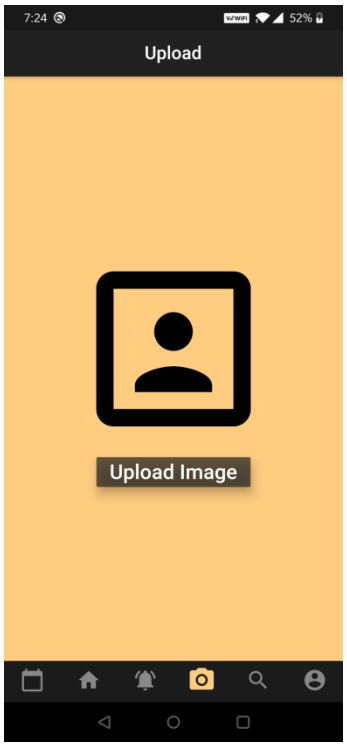
Home Screen

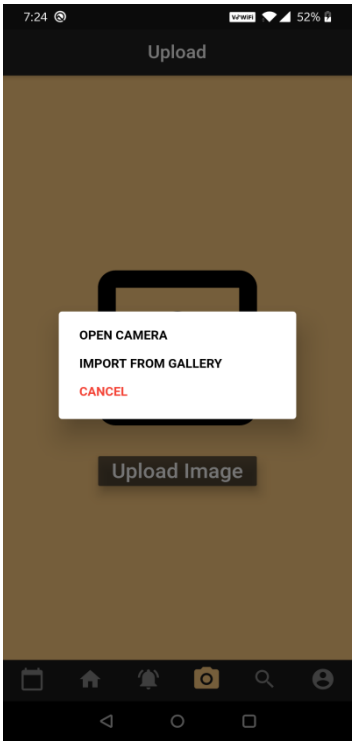


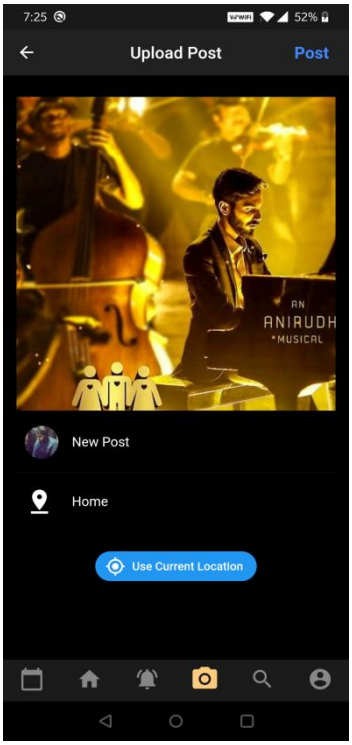
Notifications Screen

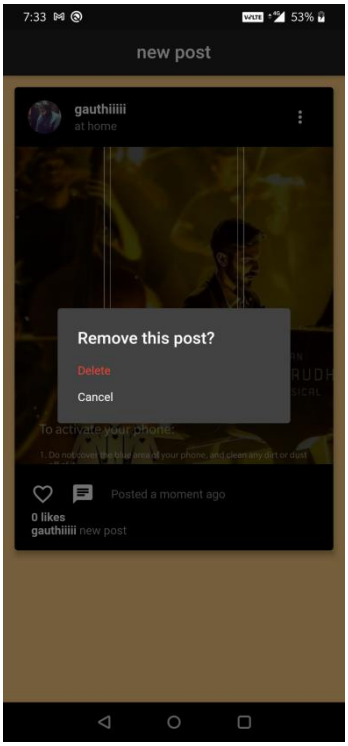


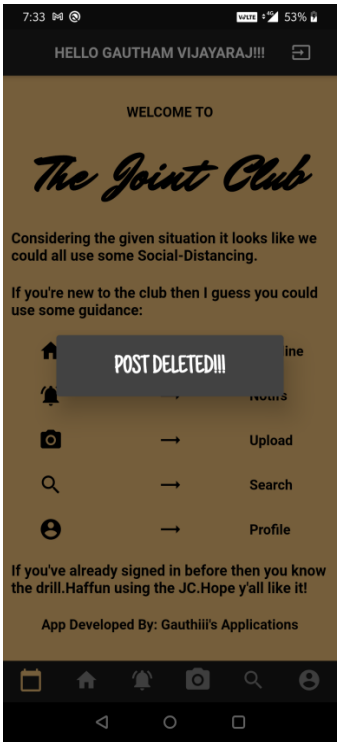
Upload Screen



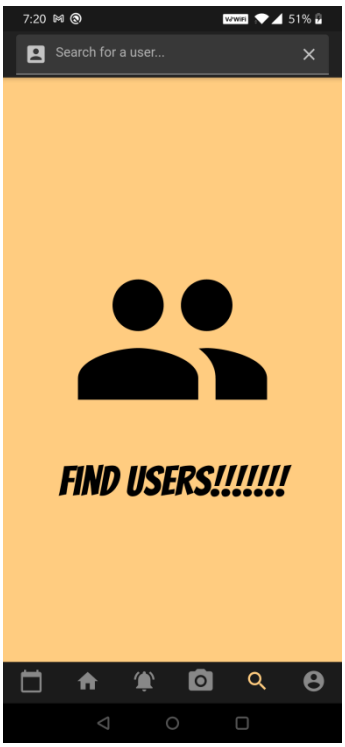


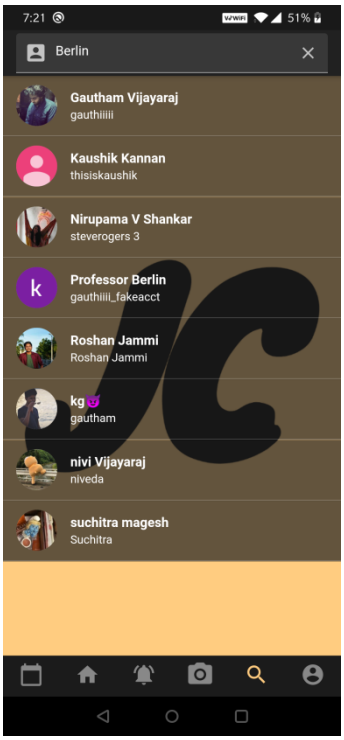




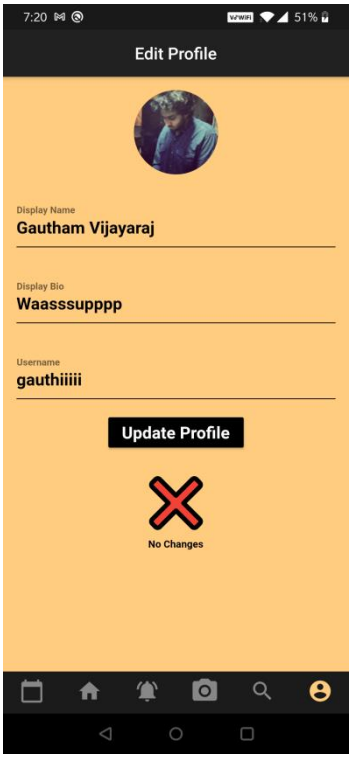


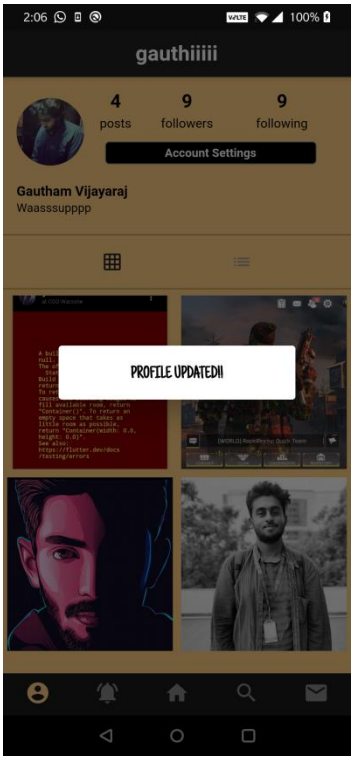
Search Page





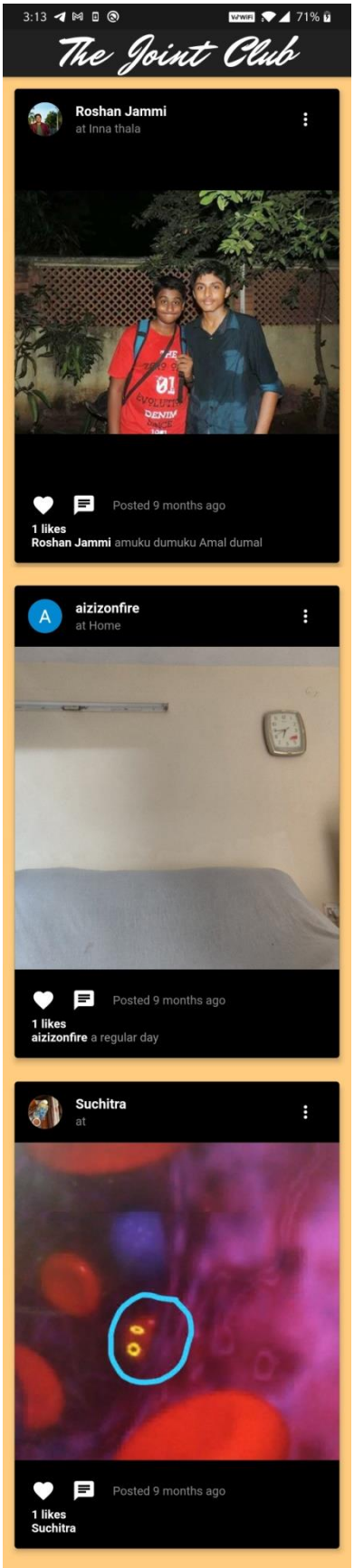
Edit Profile



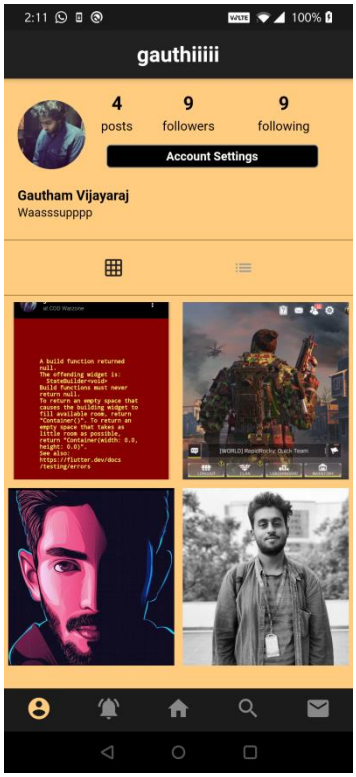


Profile (List View)

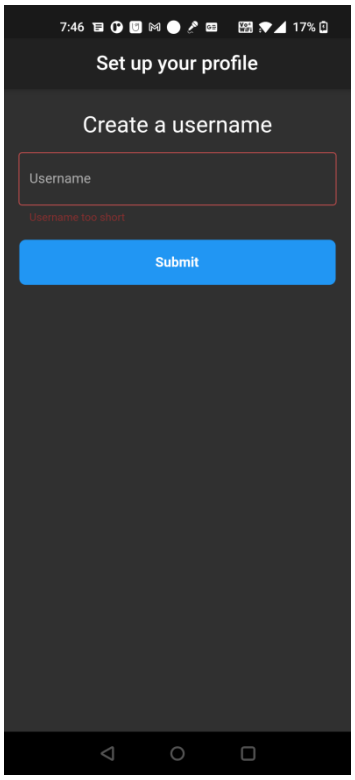
Timeline


3:13
71%
The Joint Club
Roshan Jammi
at Inna thala
Posted 9 months ago
1 likes
Roshan Jammi amuku dumuku Amal dumal
aizizonfire
at Home
Posted 9 months ago
1 likes
aizizonfire a regular day
Suchitra
at
Posted 9 months ago
1 likes
Suchitra

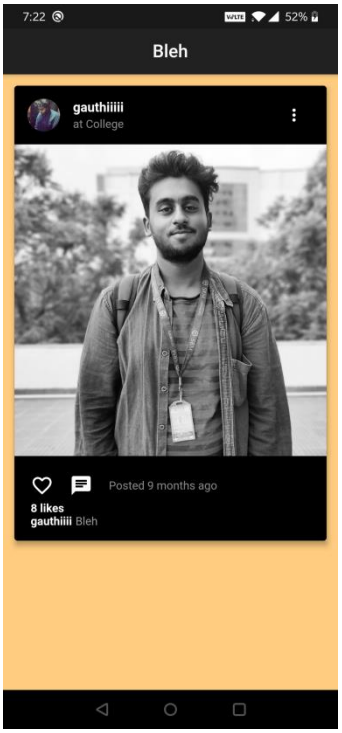
Profile (Grid View)

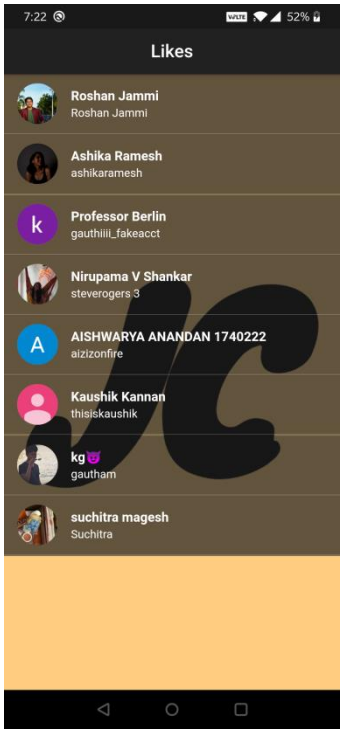


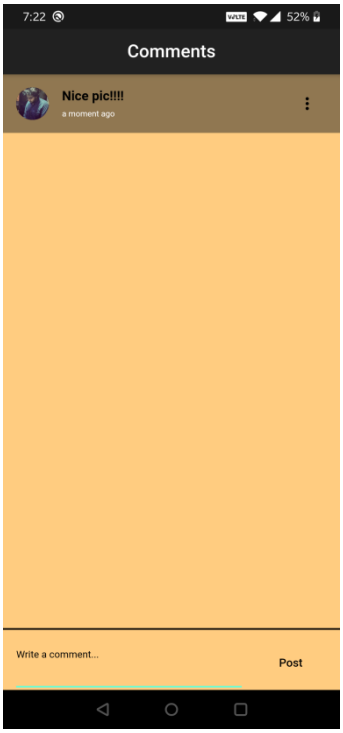
Username Setup

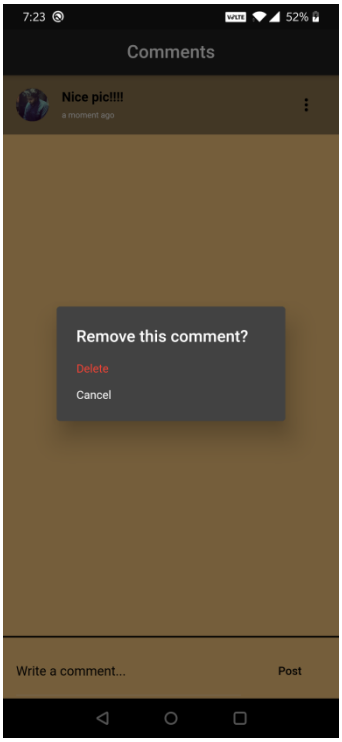


Post Screen


7:22
52%
Bleh
gauthiiiii
at College
Posted 9 months ago
8 likes
gauthiiiii Bleh


7:22
52%
Likes
Roshan Jammi
Roshan Jammi
Ashika Ramesh
ashikaramesh
Professor Berlin
gauthiiii_fakeacct
Nirupama V Shankar
steverogers 3
AISHWARYA ANANDAN 1740222
aizizonfire
Kaushik Kannan
thisiskaushik
kg
gautham
suchitra magesh
Suchitra


7:22
52%
Comments
Nice pic!!!!
a moment ago
Write a comment...
Post


7:23
52%
Comments
Nice pic!!!!
a moment ago
Remove this comment?
Delete
Cancel
Write a comment...
Post

REFERENCES

[1] Varsha R. Mouli, and K.P. Jevitha, "Web Services Attacks and Security- A Systematic Literature Review" in 6th International Conference On Advances In Computing &Communications(ICACC), 2016, pp.1-8.

[2] Hongbin Liu, ShrideepPallickara, and Geoffrey Fox, "Performance of Web Services Security", Proceedings of the 13th Mardi Gras conference, 2005, pp.1-8.

[3] Zhiyong Zhang, and Brij B.Gupta, "Social media security and trust worthiness: Overview and new direction", Future Generation Computer Systems(FGCS), 2016, pp.914-925

[4] Ashu Kumar, Amandeep Kaur, and Munish Kumar," Face detection techniques: a review", Artificial Intelligence Review, 2018, pp.1-23.

[5] Dr. M. Ilayaraja, Dr. K.Shankar, and Dr. G. Devika, "A Modified Symmetric Key Cryptography Method for Secure Data Transmission", International Journal of Pure and Applied Mathematics(IJPAM), Volume 116 , No. 10 , 2017,pp 301-308.

[6] Priyadarshini Patila, Prashant Narayankarb, Narayan D G c, and Meena S Md, "A Comprehensive Evaluation of Cryptographic Algorithms: DES, 3DES, AES, RSA and Blowfish", International Conference on Information Security &Privacy(ICISP), 2015, pp.617-624.

[7] Nilanjan Chatterjee, Souvik Chakraborty, Aakash Decosta, Dr. Asoke Nath, "Real-time Communication Application Based on Android Using Google Firebase", International Journal of Advance Research in Computer Science and Management Studies(IJARCSMS), 2018, pp. 74-79.

[8] PrasetyawidiIndrawan ,SlametBudiyatno , Nur Muhammad Ridho , and Riri Fitri Sari, "Face Recognition for Social Media With Mobile Cloud Computing", International Journal on Cloud Computing: Services and Architecture (IJCCSA), 2013, pp. 23-35.

[9] Yaniv Taigman, Ming Yang, Marc'AurelioRanzato, Lior Wolf, "DeepFace: Closing the Gap to Human-Level Performance in Face Verification", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 1701-1708.

[10] LeventGokrem , Omer Faruk Nasip, "An Encrypted Messaging Application with Multi Fragmented Caesar Encryption Method between Mobile Devices", Journal of New Results in Science (JNRS), 2017, pp. 1-10.
[11] Marcos Carlomagno **,"**Real-time Image classification using Tensorflow Lite and Flutter**"**, 2020
[12] Milind Deore , "FaceNet Architecture", 2020
[13] Marcos Carlomagno , "Face Recognition Authentication using Flutter and TensorflowLite , 2020

Format - I

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Deemed to be University u/s 3 of UGC Act, 1956)

Office of Controller of Examinations

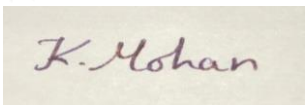
REPORT FOR PLAGIARISM CHECK ON THE DISSERTATION/PROJECT REPORTS FOR UG/PG PROGRAMMES

(**To be attached in the dissertation/ project report**)

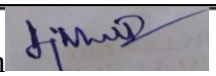
1	Name of the Candidate **(IN BLOCK LETTERS)**	K.MOHANAKRISHNAN
2	Address of the Candidate	No. 10, 3rd Cross Street, Kamarajapuram, Nandhivaram, Guduvancheri, Tamil Nadu, 603202 **Mobile Number :** 9894284530
3	Registration Number	RA1711003010371
4	Date of Birth	09/09/1999
5	Department	Computer Science and Engineering
6	Faculty	Dr.S.Saravanan
7	Title of the Dissertation/Project	Social Networking Application using FaceNet and Firebase
8	Whether the above project/dissertation is done by	Individual or group : Group (Strike whichever is not applicable) a) If the project/ dissertation is done in group, then how many students together completed the project : 2 b) Mention the Name & Register number of other candidates : Gautham Vijayaraj, RA1711003010381
9	Name and address of the Supervisor / Guide	A.M.J.Muthu kumaran, Department of Computer Science & Engineering, SRMIST, Potheri, Kattankulathur, Chengalpattu, Tamil Nadu, 603203. muthu.a@ktr.srmuniv.ac.in, 9080583664 **Mail ID : Mobile Number :**
10	Name and address of the Co-Supervisor / Co- Guide (if any)	**Mail ID : Mobile Number :**

11	Software Used	Flutter, Visual Studio Code, Android Studio, NodeJs, and Firebase		
12	Date of Verification	26/5/2021		
13	**Plagiarism Details: (to attach the final report from the software)**			
Chapter	**Title of the Chapter**	**Percentage of similarity index (including self citation)**	**Percentage of similarity index (Excluding self citation)**	**% of plagiarism after excluding Quotes, Bibliography, etc.,**
1	Introduction	0%	0%	0%
2	Literature Survey	0%	0%	0%
3	Objective of Project	0%	0%	0%
4	Architecture Diagram	0%	0%	0%
5	Face Authentication	3%	1%	2%
6	Modules	0%	0%	0%
7	Implementation and Working	3%	1%	2%
8	Conclusion	0%	0%	0%
9				
10				
Appendices				

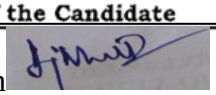
I / We declare that the above information have been verified and found true to the best of my / our knowledge.



Signature of the Candidate

AMJ Muthu Kumaran 

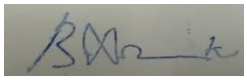
Name & Signature of the Staff (Who uses the plagiarism check software)

AMJ Muthu Kumaran 

Name & Signature of the Supervisor/Guide

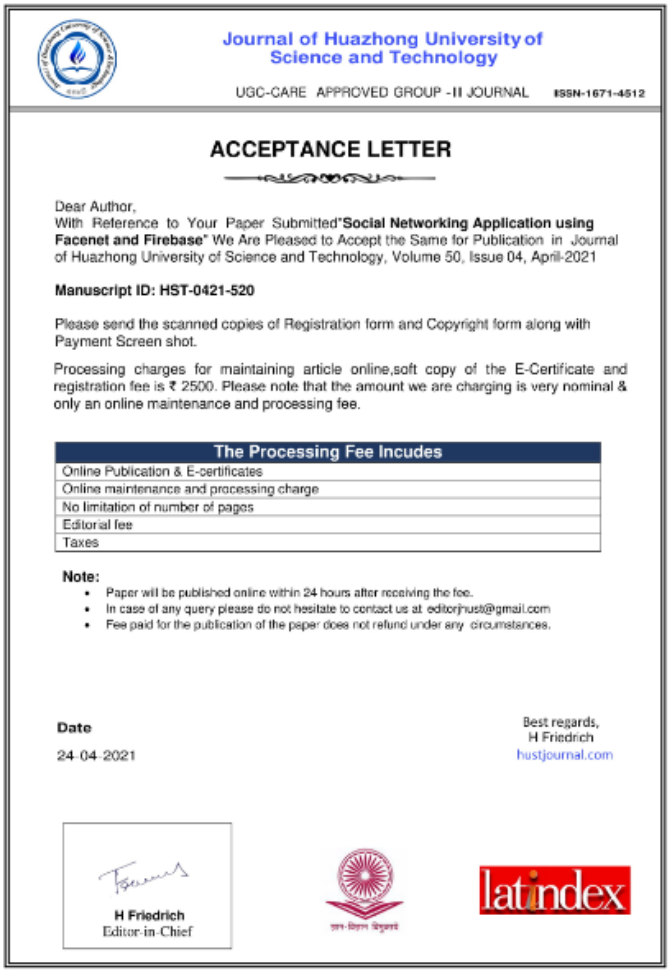
Name & Signature of the Co-Supervisor/Co-Guide

Dr B.Amudha



Name & Signature of the HOD

ACCEPTANCE LETTER



Journal of Huazhong University of Science and Technology

UGC-CARE APPROVED GROUP -II JOURNAL ISSN-1671-4512

ACCEPTANCE LETTER

Dear Author,

With Reference to Your Paper Submitted"**Social Networking Application using Facenet and Firebase**" We Are Pleased to Accept the Same for Publication in Journal of Huazhong University of Science and Technology, Volume 50, Issue 04, April-2021

Manuscript ID: HST-0421-520

Please send the scanned copies of Registration form and Copyright form along with Payment Screen shot.

Processing charges for maintaining article online,soft copy of the E-Certificate and registration fee is ₹ 2500. Please note that the amount we are charging is very nominal & only an online maintenance and processing fee.

The Processing Fee Incudes
Online Publication & E-certificates
Online maintenance and processing charge
No limitation of number of pages
Editorial fee
Taxes

Note:

- Paper will be published online within 24 hours after receiving the fee.
- In case of any query please do not hesitate to contact us at editorjhust@gmail.com
- Fee paid for the publication of the paper does not refund under any circumstances.

Date

24-04-2021

Best regards,
H Friedrich
hustjournal.com

H Friedrich
Editor-in-Chief

latindex