

# **IMAGE SHARING APPLICATION**

A report submitted for the CSE-200 Project

*By*

**Anupriya Sinha**

Bachelor of Technology, IV Semester

Roll No. 16010108



Department of Computer Science and Engineering  
**Indian Institute of Information Technology Manipur**  
April, 2018

# **Abstract**

In this project, Attempts have been made to develop an Image sharing application that allows user to share their images with other users. This Application is developed keeping photographers in mind. This Application is user friendly and helpful for those looking for image sharing platform.

# **Declaration**

I declare that this submission represents my idea in my own words and where others' idea or words have been included, I have adequately cited and referenced the original source. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/sources in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and can also evoke penal action from the sources which have thus not been properly cited or from proper permission has not been taken when needed.

(Anupriya Sinha)  
(16010108)

Date: April, 2018



Department of Computer Science & Engineering  
Indian Institute of Information Technology Manipur

---

Dr. Thoudam Doren Singh  
Assistant Professor

Email: doren@iiitmanipur.ac.in

## To Whom It May Concern

This is to certify that the report entitled "**IMAGE SHARING APPLICATION**" submitted to by "Anupriya Sinha", has been carried out under my supervision and that this work has not been submitted elsewhere for a degree, diploma or a course.

Signature of Supervisor

(Dr. Thoudam Doren Singh )

# Acknowledgement

I would like to extend my special thanks to my mentor "**Dr. Thoudam Doren Singh**", Assistant Professor at IIIT Manipur, Department of CSE, for providing such a golden opportunity to do the project "IMAGE SHARING APPLICATION" under his guidance. I would also like to thank my seniors and friends who helped me whenever I was stuck and share their ideas which ultimately helped me to make my Application better.

Anupriya Sinha

# Contents

<b>Abstract</b>	ii
<b>Declaration</b>	iii
<b>Certificate</b>	iv
<b>Acknowledgement</b>	v
<b>Table of contents</b>	vi
<b>List of figures</b>	ix
<b>List of abbreviations</b>	xi
<b>1 Introduction</b>	1
1.1 Outline of the report . . . . .	2
1.1.1 Gantt chart . . . . .	2
<b>2 Existing System Study</b>	3
<b>3 System Analysis, Design &amp; Implementation</b>	4
3.1 Introduction . . . . .	4
3.2 Design . . . . .	5
3.2.1 PROJECT LAYOUT . . . . .	5

3.2.2	Database Design . . . . .	7
3.2.3	ER Diagram . . . . .	9
3.3	Implementation . . . . .	10
3.3.1	Authentication . . . . .	10
3.3.2	User Interface . . . . .	10
3.3.3	Data Retrieval and Displaying . . . . .	10
<b>4</b>	<b>Conclusion</b>	<b>12</b>
4.1	Limitations . . . . .	12
4.2	Future direction . . . . .	13
<b>Appendix A</b>	<b>Screenshot and Description of the Implemented System</b>	<b>14</b>
A.1	Application Screenshots . . . . .	14
A.2	Database Implementation . . . . .	23
<b>Appendix B</b>	<b>User manual</b>	<b>27</b>
B.1	Introduction . . . . .	27
B.1.1	Logging In . . . . .	27
B.1.2	Uploading Post . . . . .	27
B.1.3	Settings . . . . .	27
B.1.4	Users List . . . . .	28
B.1.5	Signing Out . . . . .	28
B.2	Step to install . . . . .	28
B.3	Code Snippet . . . . .	29
B.3.1	Creating Reference to Root Node . . . . .	29
B.3.2	Setting Listener on reference . . . . .	29
B.3.3	Populating recyclerView using FirebaseRecyclerAdapter . . . . .	29

B.3.3.1 ViewHolder of RecyclerView . . . . .	31
--	----

# List of Figures

1.1	Gantt Chart . . . . .	2
3.1	Project Layout . . . . .	6
3.2	Project Layout . . . . .	6
3.3	JSON Tree for users . . . . .	7
3.4	JSON Tree for Posts . . . . .	7
3.5	JSON Tree for Comments . . . . .	8
3.6	JSON Tree for Likes . . . . .	8
3.7	JSON Tree for Following . . . . .	8
3.8	ER Diagram . . . . .	9
3.9	Multivalued Attribute . . . . .	9
A.1	All Authentication methods . . . . .	14
A.2	Gmail Authentication . . . . .	14
A.3	Add account if doesn't have one . . . . .	15
A.4	Signing Up with existing Email id . . . . .	15
A.5	Invalid message for empty field . . . . .	16
A.6	Signing Up with existing Email id . . . . .	16
A.7	Posts by Users . . . . .	17
A.8	Comments by Users . . . . .	17
A.9	Navigation Bar . . . . .	18

A.10 Image Selection . . . . .	18
A.11 Page for Uploading Post . . . . .	19
A.12 Set Profile Picture . . . . .	19
A.13 Settings Page . . . . .	20
A.14 Email Syntax verification . . . . .	20
A.15 Home Page of user . . . . .	21
A.16 Deleting Post . . . . .	21
A.17 List of Users . . . . .	22
A.18 search Users . . . . .	22
A.19 Root Nodes . . . . .	23
A.20 JSON Tree for Users . . . . .	23
A.21 JSON Tree for Posts . . . . .	24
A.22 JSON Tree for Post Comments . . . . .	25
A.23 JSON Tree following . . . . .	26
A.24 JSON Tree for Post Like . . . . .	26

# List of abbreviations

**A**

---

App. Application

**D**

---

DB Database

**S**

---

SDK Software Development Kit



# Chapter 1

## Introduction

Many image sharing App. are available with various features, but there very few App. available for photographers. The idea is to make an Image sharing application from photographer's point of view with following feature.

- SignUp/LogIn
- Upload Images (Single Image not album of images)
- Like
- Comment
- Follow
- Settings (User Name, Email, About, Date of birth)
- Searching user by "UserName"
- Delete Post

## 1.1 Outline of the report

This report is organised around four main parts.

**Chapter 2** Existing System Study

**Chapter 3** System Analysis, Design & Implementation

**Chapter 4** Conclusion

### 1.1.1 Gantt chart

Gantt chart shown below is the graphical depiction of project schedule that shows the start and finish dates of several elements involved in this project.

Gantt Chart	February	March	April
Information Gathering, Feasibility Study, Design	1-02-2018 to 26-02-2018		
Implementation		28-02-2018 to 10-04-2018	
Testing			11-04-2018 to 17-04-2018
Maintainence			17-04-2018 onwards

Figure 1.1: Gantt Chart

# Chapter 2

## Existing System Study

The idea of developing an Image Sharing App. came from existing App. “AGORA”, which include following features :

- SignUp/LogIn
- Upload Image (Single Image at a time)
- Like
- Follow
- Settings (User Name, Email, About, Date of birth)
- Searching user by “UserName”
- Searching images with tags

This App. is designed from photographers point of view, where they can share the images clicked by them with all the users. This App. has like feature but doesn't have comment feature for posts. In this Application, the comment feature for each post has been included.

# **Chapter 3**

## **System Analysis, Design & Implementation**

### **3.1 Introduction**

**For the development of Image sharing Application, following platforms are used :**

- Android studio : For App. designing
- Firebase : To store data, images and authenticate users
- Mobile phone (Android based with minimum SDK 15) : To test the App.

Android studio is open source software. Firebase is a Backend-as-a-Service (BaaS), mobile application development platform provided by Google. Firebase services used in this project are :

- Firebase Authentication
- Firebase Storage (Store Images)

- Firebase RealTime Database (NoSQL DB)

Firebase allows 10,000 users authentication free, 30GB storage free and 1GB database storage free, which is enough for testing purpose. Therefore, the firebase services used for the development of this App. are free.

## 3.2 Design

This design section discusses the database design, authentication, posts retrieval. Authentication is required in order to use this Application. Authentication methods included in this App. are :

- Login with Gmail
- Login with Facebook
- Sign up if don't have account using Email with the application:  
Check the existance of Email id and register the user with given Email and password.  
Just after registration, user will be directed to Main page that contains recent posts by the App. users.
- Login with registered Email and password

On successful user authentication, user will be directed to "main page" that contains posts uploaded by users. Displaying post include image retrieval from the firebase storage and posts related other information (Caption, Likes, comments, user name) from firebase realtime database. This "main page" contains four intents to four different activities that are "SETTINGS PAGE", "USERS HOME PAGE", "USERS LIST" and "UPLOAD POST". "SETTINGS PAGE" allows users to change/set their information like UserName, Date of Birth, Email, Profile Image and something about them. The "USERS LIST" contains the list of users registered with the App. and it also contain search feature to search user by user name.

### 3.2.1 PROJECT LAYOUT

Here is the layout of my Application :

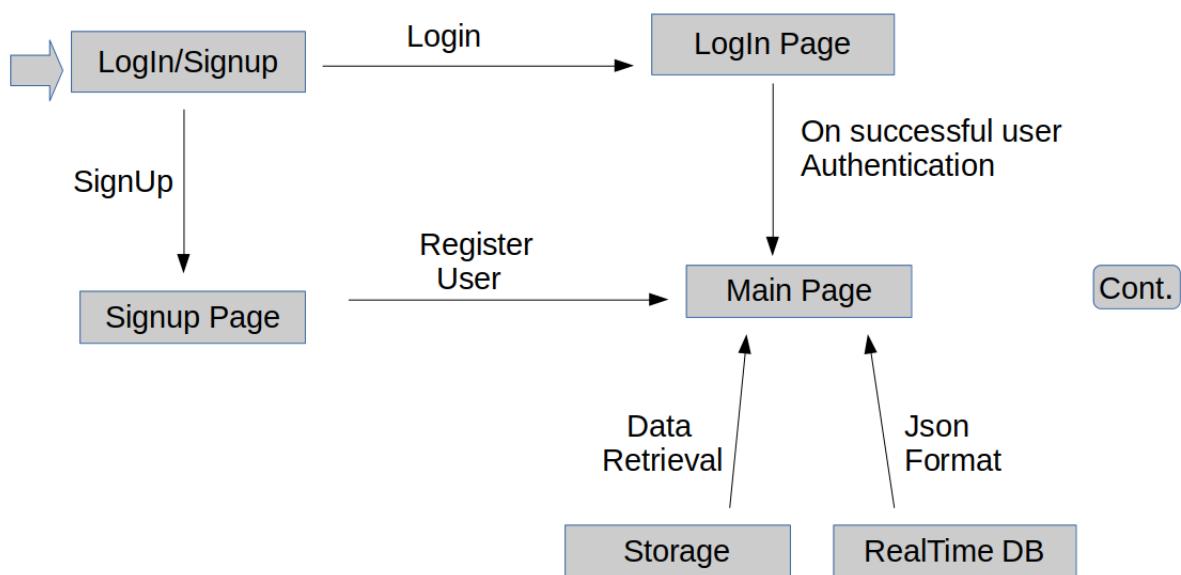


Figure 3.1: Project Layout

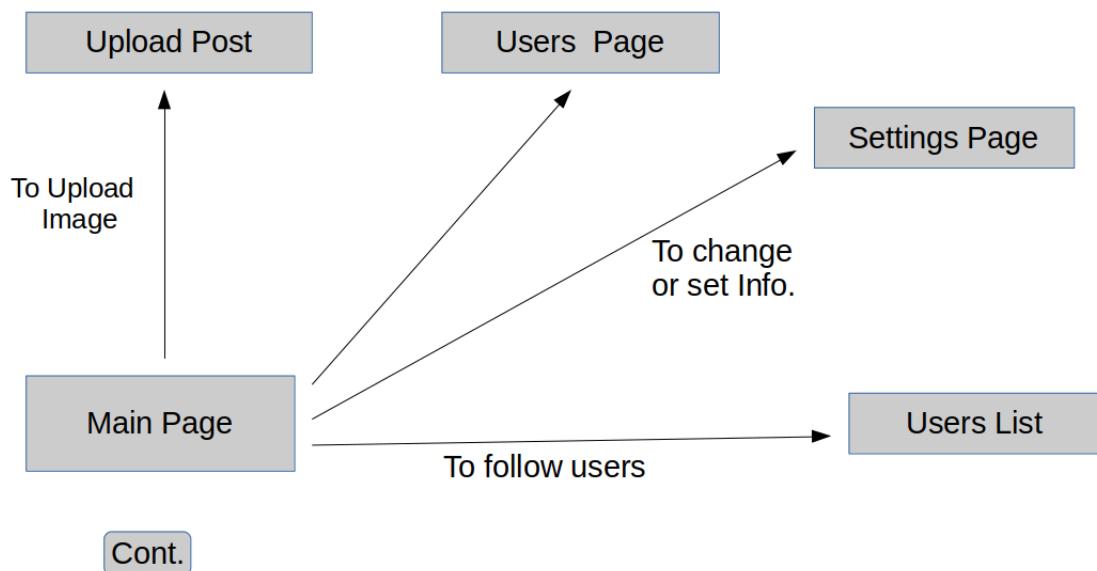


Figure 3.2: Project Layout

### 3.2.2 Database Design

NoSql database has been used in this project with 5 root nodes in all. Structure of each child nodes are shown below on figures :

Store user's profile photo url and other information :

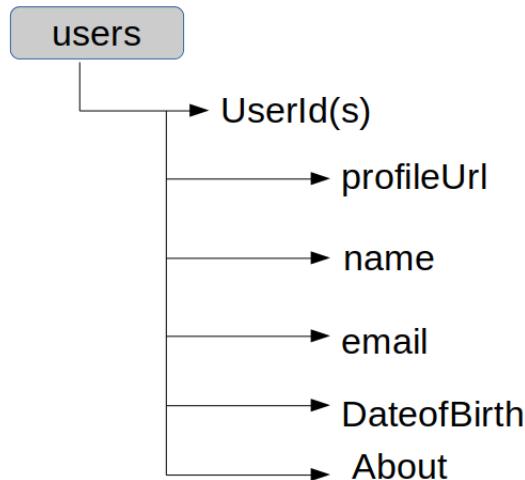


Figure 3.3: JSON Tree for users

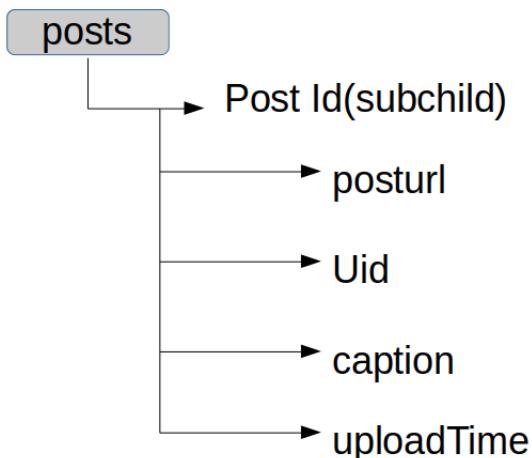


Figure 3.4: JSON Tree for Posts

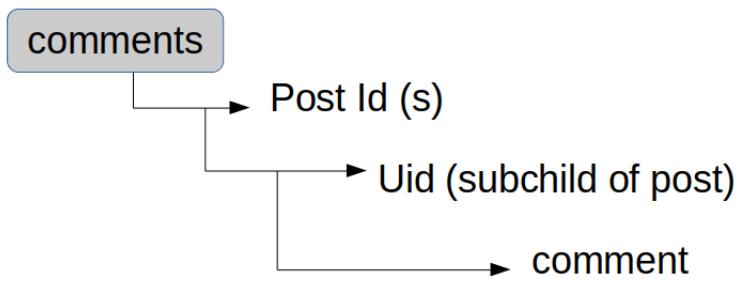


Figure 3.5: JSON Tree for Comments

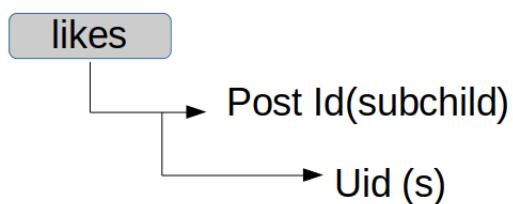


Figure 3.6: JSON Tree for Likes

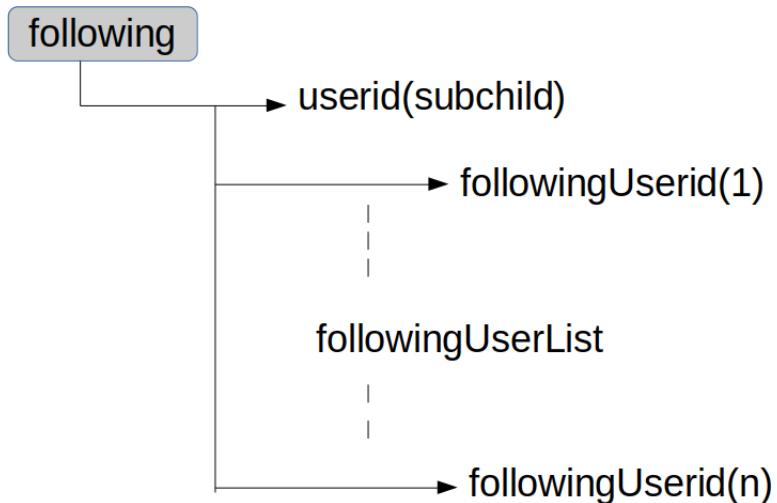


Figure 3.7: JSON Tree for Following

### 3.2.3 ER Diagram

Below is the ER diagram for users and posts with cardinality ratio 1:n as single user can upload any number of posts.

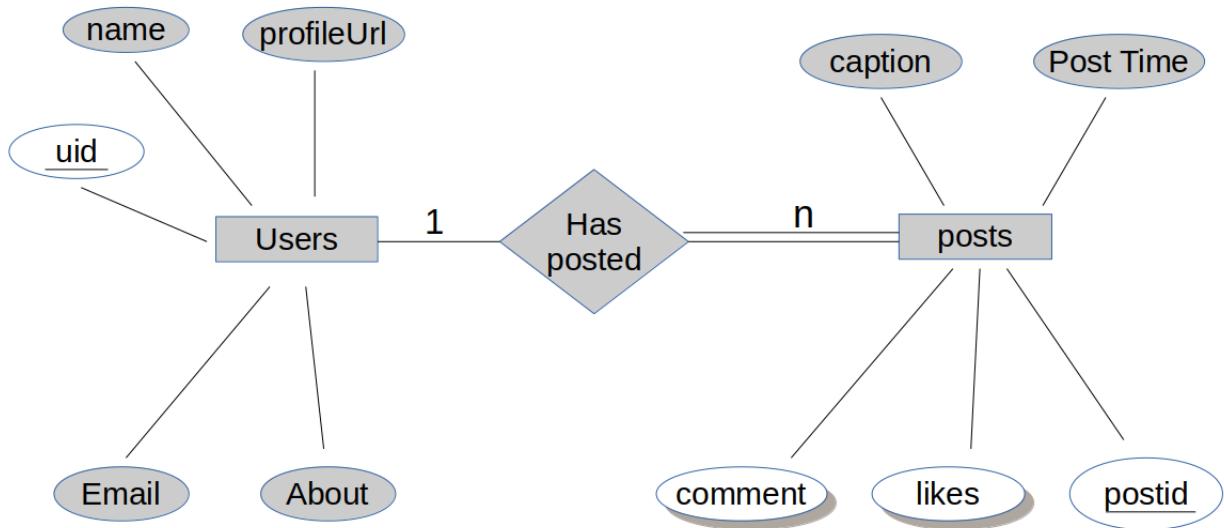


Figure 3.8: ER Diagram

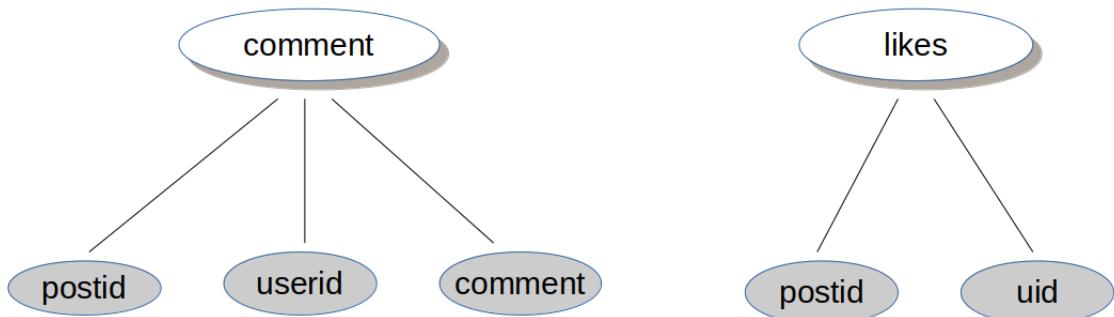


Figure 3.9: Multivalued Attribute

## **3.3 Implementation**

This section discusses the implementation of “PicStack” Application.

### **3.3.1 Authentication**

Authentication is the very first phase of implementation that prompts user to verify their identity. Three authentication methods (Gmail, Facebook, Creating own account) are implemented in this application. Gmail authentication and Creating/Logging In with this application are implemented by enabling permission in firebase and to achieve facebook authentication, I have registered my application with “facebook developer” (“developers.facebook.com”).

### **3.3.2 User Interface**

User Interface are designed using Android widget listed below :

Textview  
Plain Text  
E-mail  
Password  
Imageview  
Buttons  
Recyclerview  
Cardview.

Recyclerview is preferred over listview as it reuses cells while scrolling and its efficient.

### **3.3.3 Data Retrieval and Displaying**

On successful authentication, user will be directed to “Main Page” that contains all the posts uploaded by users. Data are retrieved from firebase realtime database in JSON format. Displaying post involves retrieval of image from firebase storage and other post related information from firebase realtime database in JSON format.

Steps involved in displaying retrieving data from Storage and DB are as follow :

- Create reference to the root node of database.
- Set listeners, “`addValueEventListener`” on the reference nodes.
- Recyclerview are populated with using “`FirebaseRecyclerAdapter`” and `populateViewHolder`.

The some of the codes used in this project are included in Appendix B and the screenshots of result of the project implementation are shown in Appedix A.

# **Chapter 4**

## **Conclusion**

An Image Sharing Application has been developed where users can register themselves and they can upload images with caption. Each post has comment and like features. One user can choose to follow other users. Users can also change/set their details(Username, Email, Date of birth, Profile picture).

### **4.1 Limitations**

This application have the following limitation :

- Even if user wants his/her posts to hide from someone, they can't do this
- For particular post, each user can have at most one comment section. If user has commented on a post and again comment something on the same post, then the old comment will be overwritten with the new comment.

## **4.2 Future direction**

There are many more features that can be included in this Image Sharing Application. Some of the features that can be added to this application are as follow :

- Image copyright feature
- Automatic video generation with few posts having maximum number of likes after every certain interval of time
- Multi-comment by single user on a particular post
- Hiding post from intended user

# Appendix A

## Screenshot and Description of the Implemented System

### A.1 Application Screenshots

Below are the screenshots of each activity of the developed "PicStack" Application :

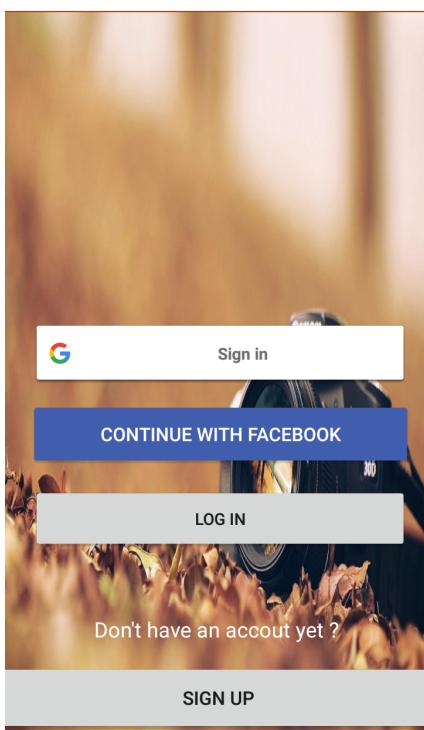


Figure A.1: All Authentication methods

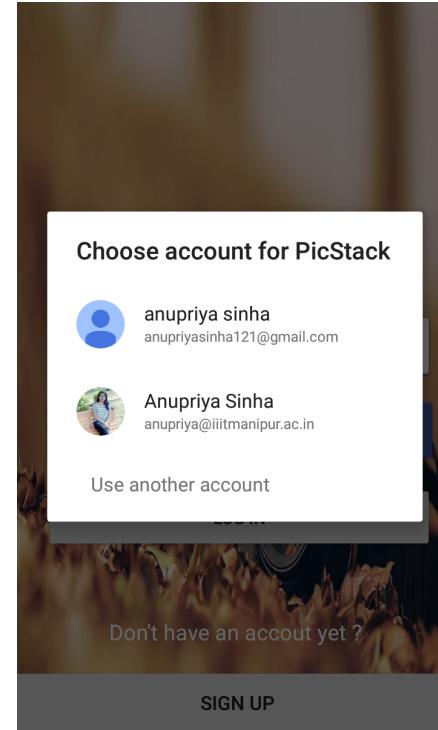


Figure A.2: Gmail Authentication

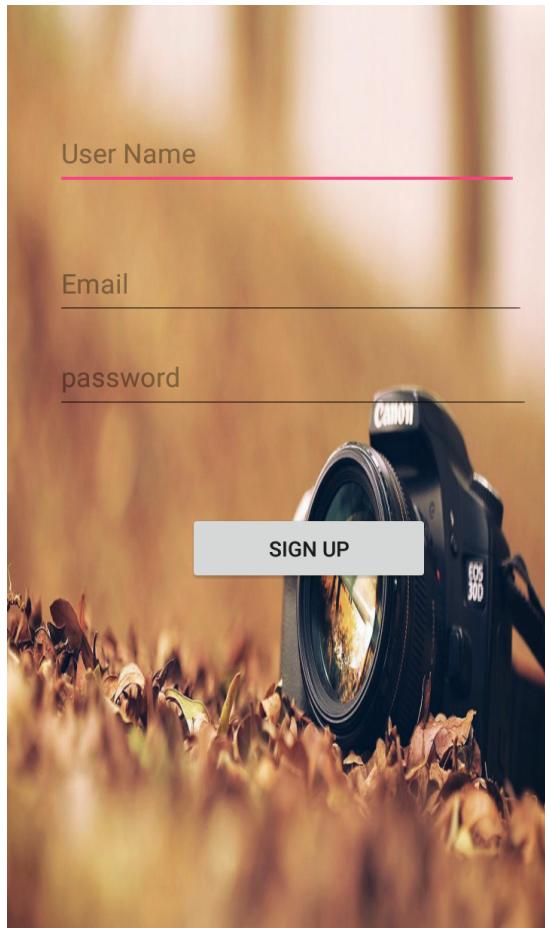


Figure A.3: Add account if doesn't have one

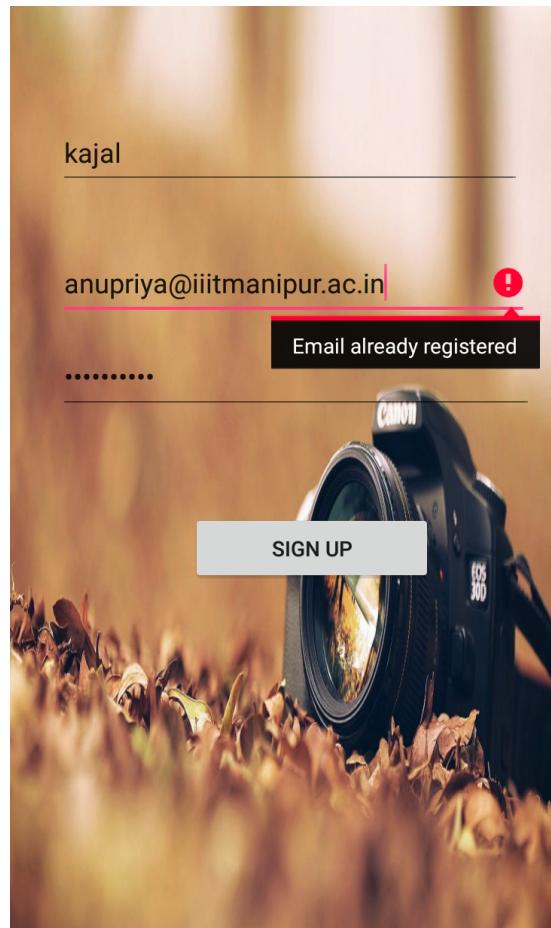


Figure A.4: Signing Up with existing Email id

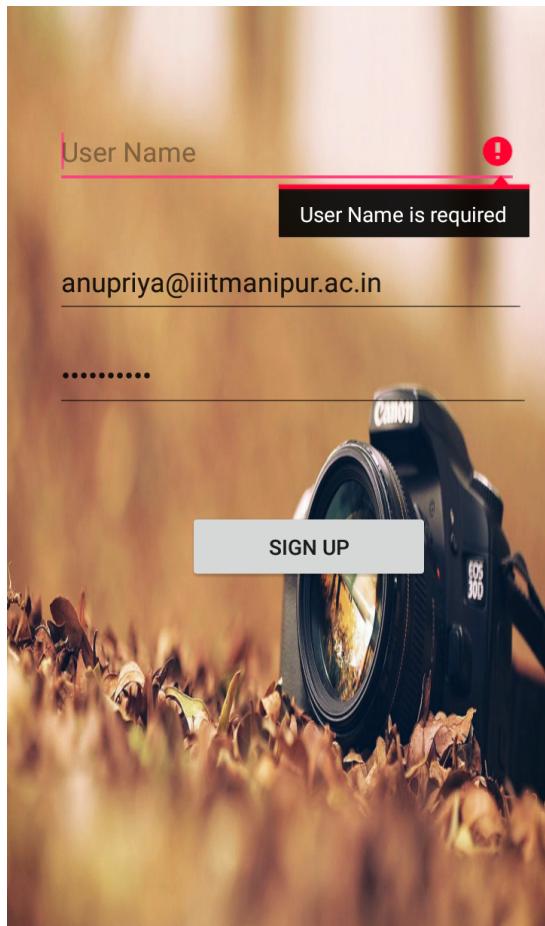


Figure A.5: Invalid message for empty field

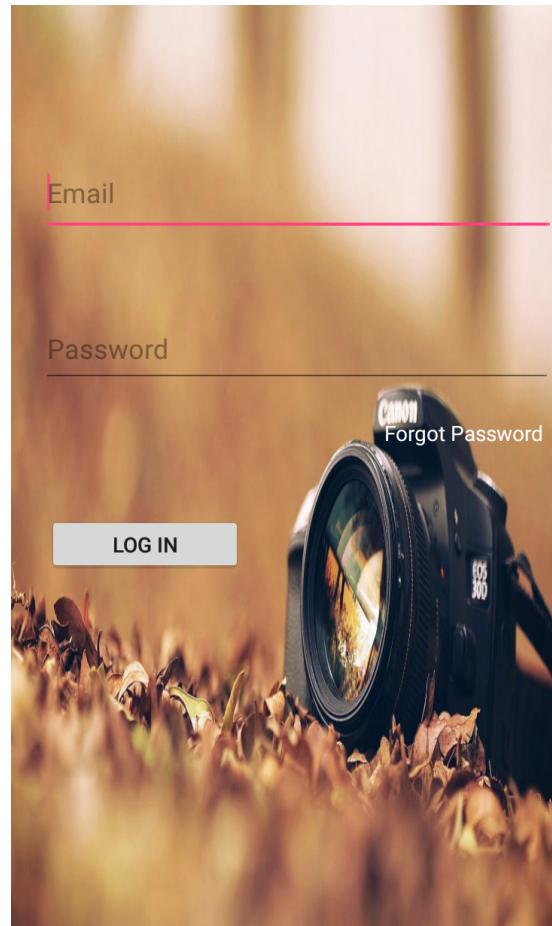


Figure A.6: Signing Up with existing Email id



Figure A.7: Posts by Users

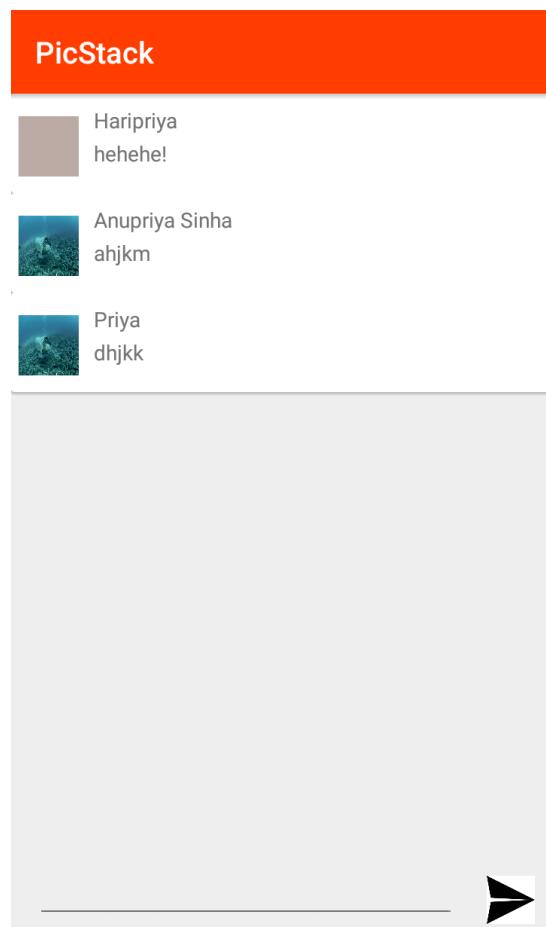


Figure A.8: Comments by Users

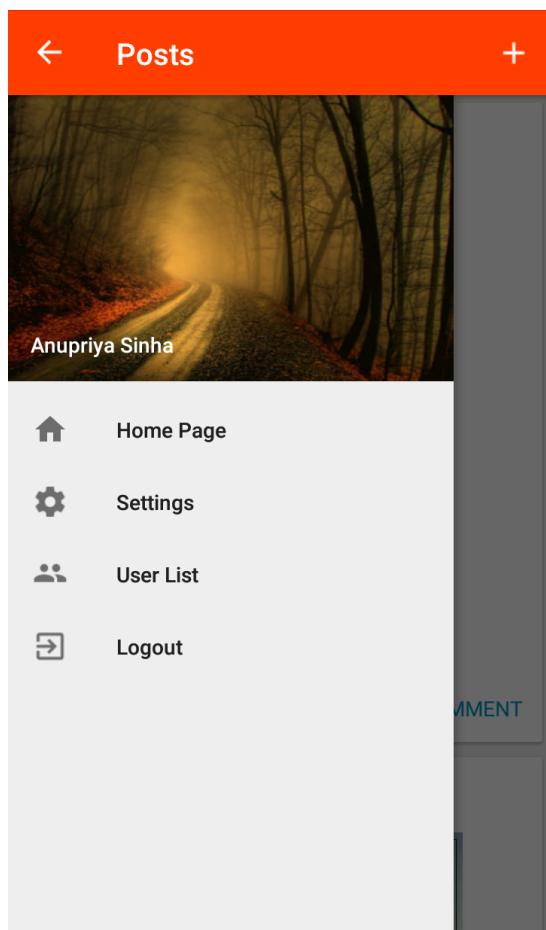


Figure A.9: Navigation Bar

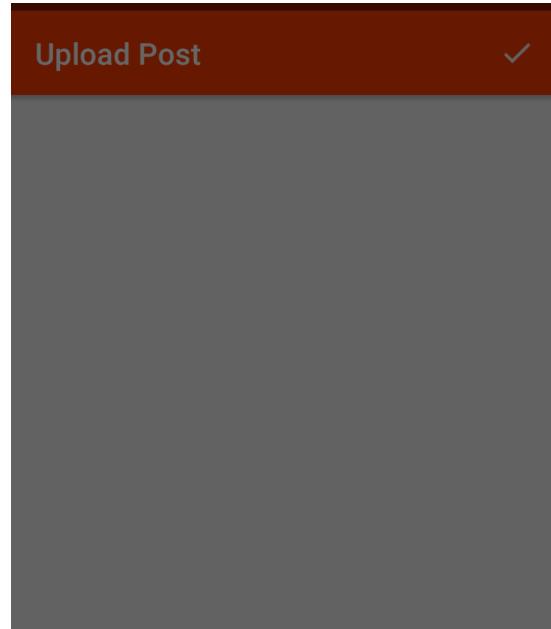


Figure A.10: Image Selection

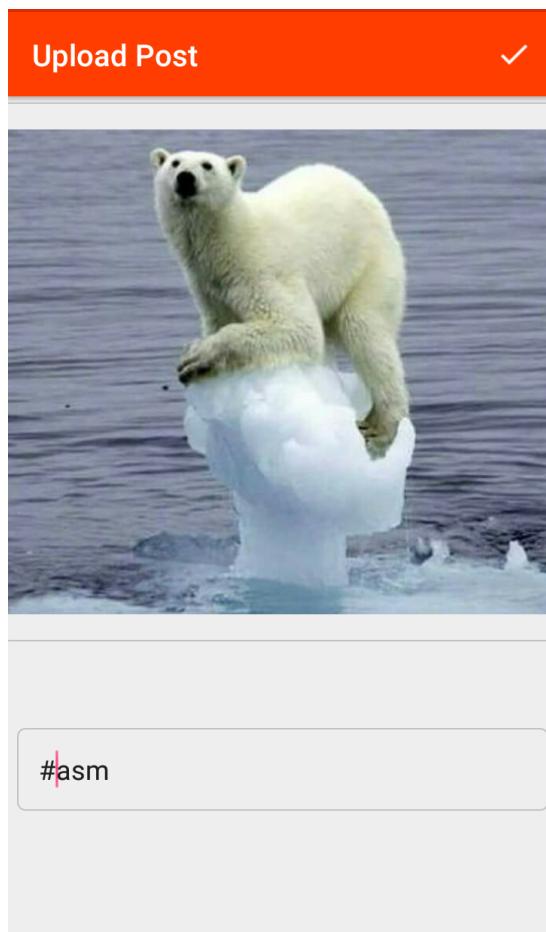


Figure A.11: Page for Uploading Post

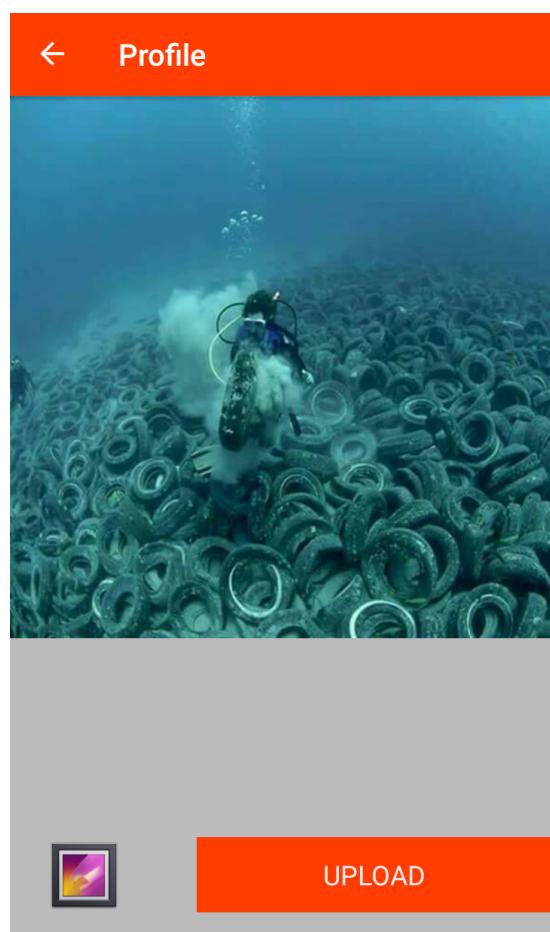


Figure A.12: Set Profile Picture

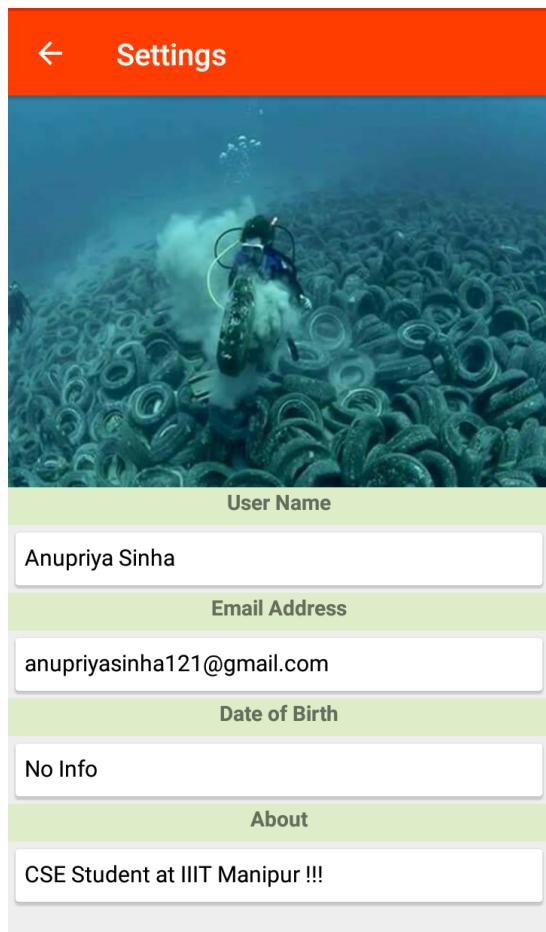


Figure A.13: Settings Page

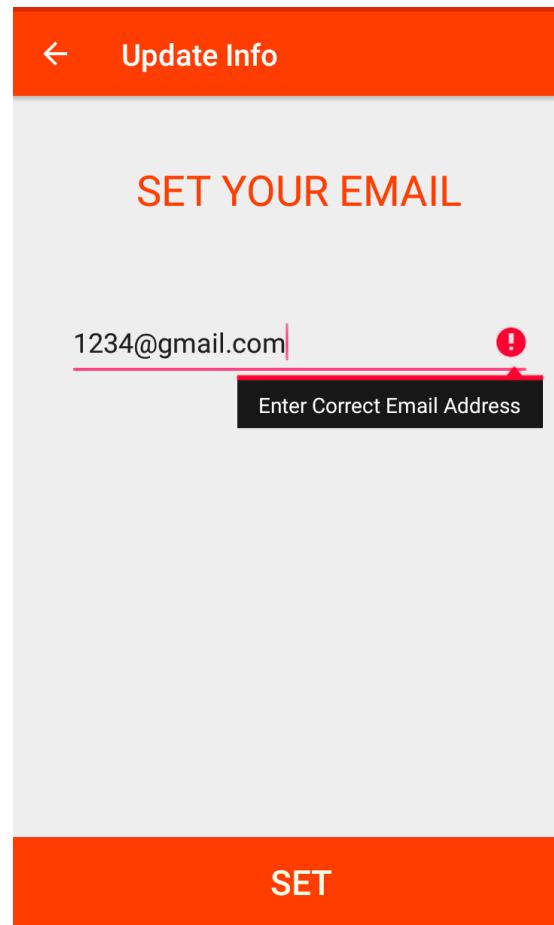


Figure A.14: Email Syntax verification



Anupriya Sinha

Apr 18, 2018 3:52:42 PM

Your mind will always BELIEVE everything you tell it. FEED it hope. FEED it truth. FEED it with love.

fgjk

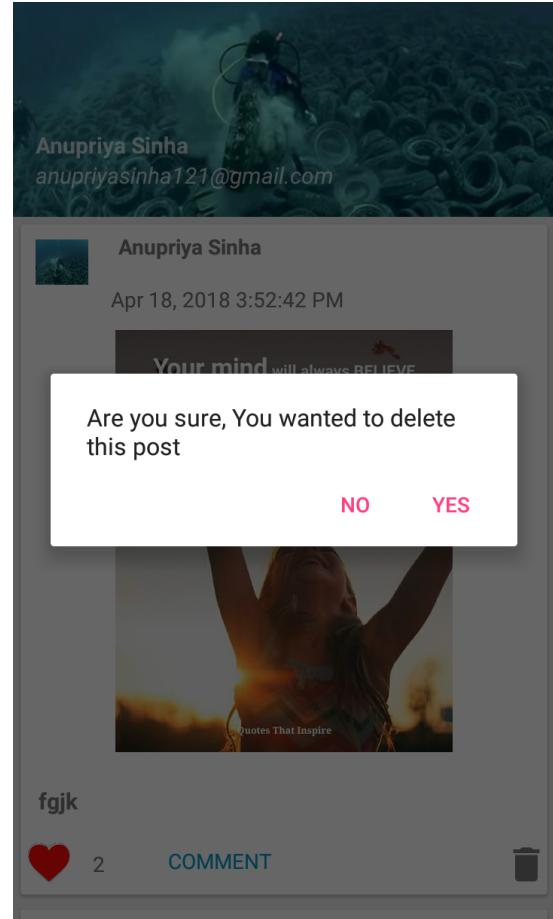


Figure A.15: Home Page of user

Figure A.16: Deleting Post

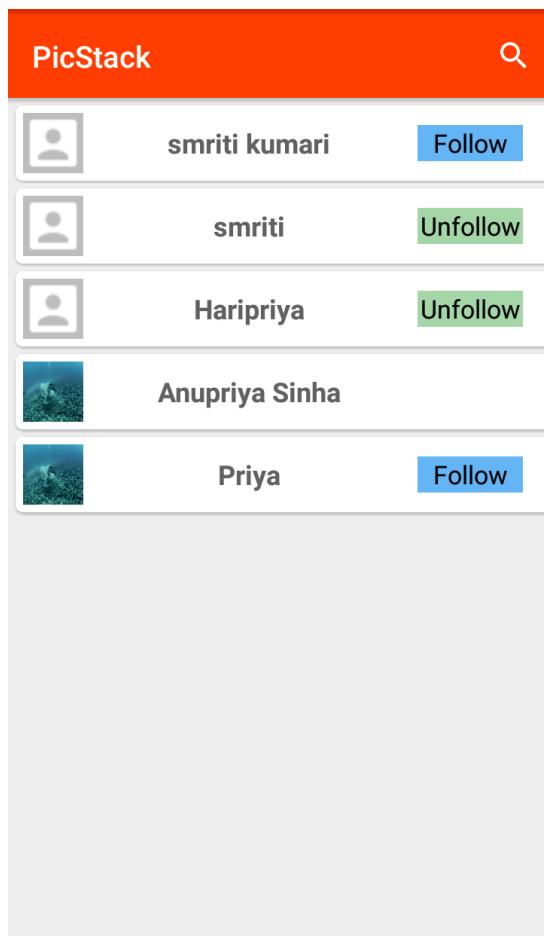


Figure A.17: List of Users

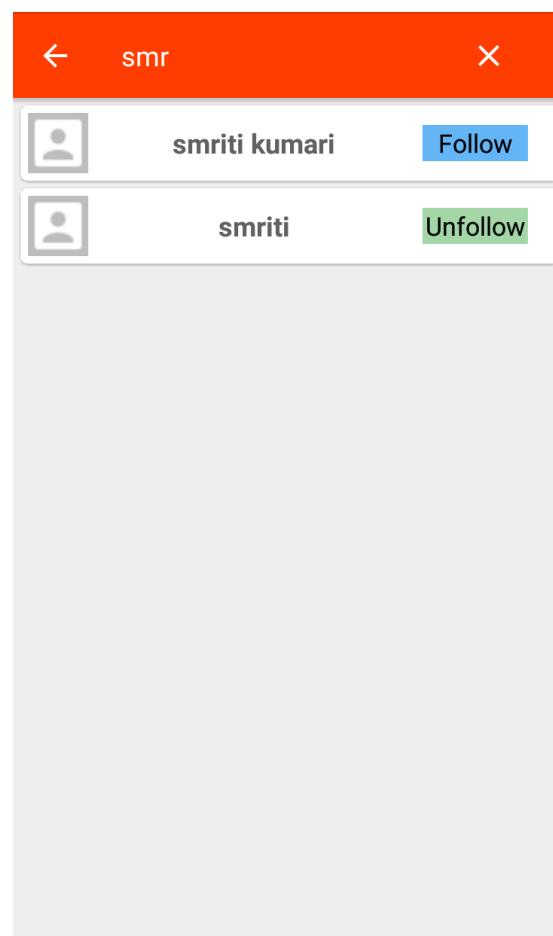


Figure A.18: search Users

## A.2 Database Implementation

Following figures show how the data are stored in Firebase realtime DB and the children and subchildren of each nodes :



Figure A.19: Root Nodes

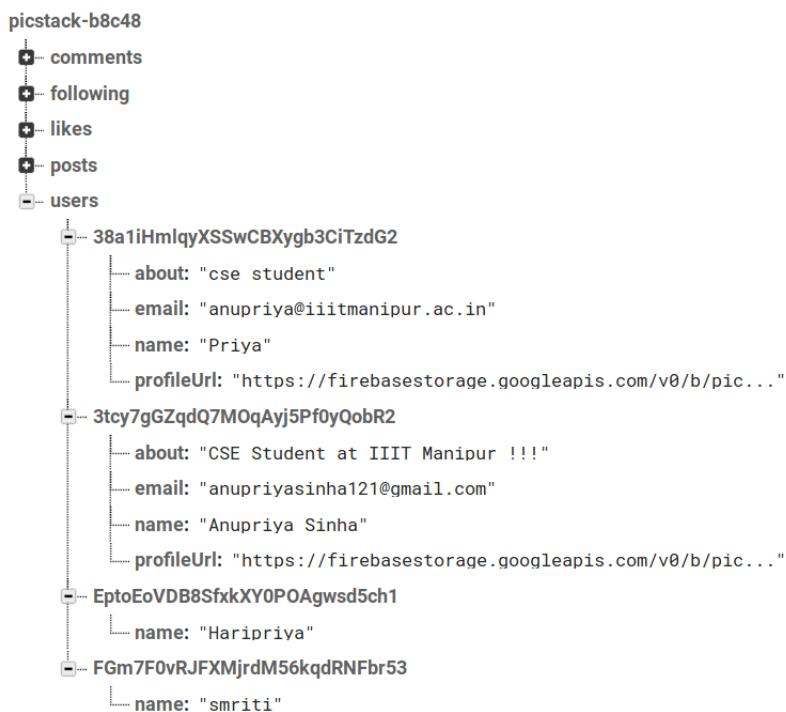


Figure A.20: JSON Tree for Users

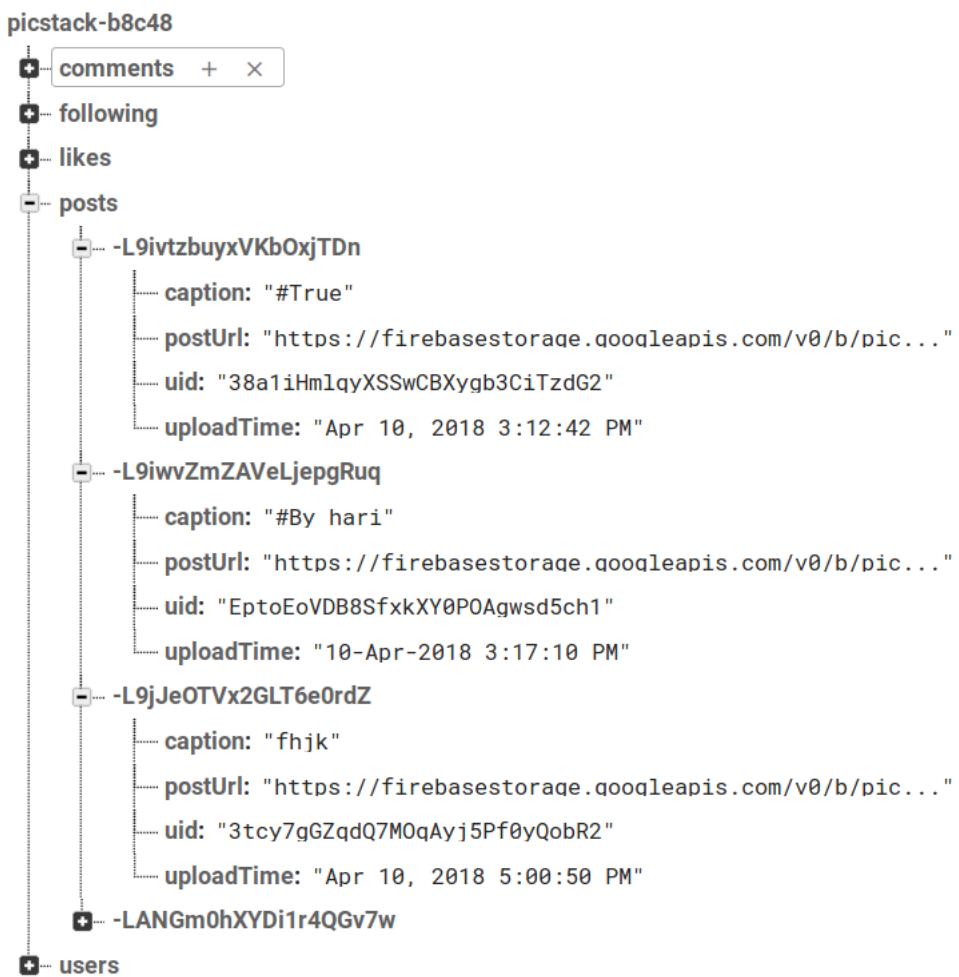


Figure A.21: JSON Tree for Posts



Figure A.22: JSON Tree for Post Comments



Figure A.23: JSON Tree following



Figure A.24: JSON Tree for Post Like

# **Appendix B**

## **User manual**

This chapter provides a general walkthrough of the system from initiation through exit to familiarize users with the important features of the Image Sharing Application, "Pic-Stack".

### **B.1 Introduction**

This Application is based on Android OS with SDK version 15 or above.

#### **B.1.1 Logging In**

There are three methods of logging In, LogIn with Gmail, Facebook and with registered Email Id and Password with the Application. On successful authenticaton user will be directed to the "Main Page" that contains recent posts by all users registered with Application and intents to other avtivities.

#### **B.1.2 Uploading Post**

To upload Image click "add" button on the right of "Main Page", then select image and then post.

#### **B.1.3 Settings**

To change/set "User Name", "Email Address", "Date of Birth", "something about them-self" and "Profile Picture", click on the "settings" option in navigation bar of "Main

Page". This will direct user to the settings Page where users can change/set their details.

#### **B.1.4 Users List**

Click on the "User List" option in navigation bar of "Main Page". This will direct user to the activity containing the list of all the users registered with the Application. This activity also contains the search feature, where users can search the user with "User Name" and choose to follow them.

#### **B.1.5 Signing Out**

Click on the "Logout" option in navigation bar of "Main Page" to Sign Out from the Logged In account.

### **B.2 Step to install**

Download the apk (Android based mobile with SDK 15 or more) and install the application.

## B.3 Code Snippet

### B.3.1 Creating Reference to Root Node

```
private DatabaseReference users = FirebaseDatabase.getInstance().getReference().child("/users");
```

### B.3.2 Setting Listener on reference

```
users.addValueEventListener(new ValueEventListener()
@Override
public void onDataChange(DataSnapshot dataSnapshot)
String name = dataSnapshot.child(pAuth.getCurrentUser().getUid()).child("name")
.getValue().toString();
ptvname.setText(name);

@Override
public void onCancelled(DatabaseError databaseError)

);
```

### B.3.3 Populating recyclerView using FirebaseRecyclerAdapter

```
FirebaseRecyclerAdapter<DisplayPost, DisplayPostHolder> firebaseRecyclerAdapter =
new FirebaseRecyclerAdapter<DisplayPost, DisplayPostHolder>(
    DisplayPost.class, R.layout.post_row, DisplayPostHolder.class, pdatabaseReference
) @Override protected void populateViewHolder(final DisplayPostHolder viewHolder,
DisplayPost model, int position)
final String postKey = getRef(position).getKey();
viewHolder.checkUserName(getApplicationContext(), model.getUid());
viewHolder.setUserName(getApplicationContext(), model.getUid());
viewHolder.setPostImage(getApplicationContext(), model.getPostUrl());
viewHolder.setCaption(model.getCaption());
viewHolder.setTime(model.getUploadTime());
viewHolder.setLike(postKey);
```

```

        viewHolder.plike.setOnClickListener(new View.OnClickListener()
@Override
public void onClick(View view)

        pprocessLike = true;

        pdatabaseReferenceLikes.addValueEventListener(new ValueEventListener()
@Override public void onDataChange(DataSnapshot dataSnapshot)

        if(pprocessLike)

            if(dataSnapshot.child(postKey).hasChild(pAuth.getCurrentUser().getUid())) pdatabaseReferenceLikes.child(postKey).child(pAuth.getCurrentUser().getUid()).removeValue();
viewHolder.plike.setImageResource(R.drawable.blank_love);
pprocessLike = false;
else pdatabaseReferenceLikes.child(postKey).child(pAuth.getCurrentUser().getUid()).setValue("lik

        @Override
public void onCancelled(DatabaseError databaseError)

);

);

        viewHolder.pcomment.setOnClickListener(new View.OnClickListener()
@Override
public void onClick(View view)

        Intent i = new Intent(MainPage.this, Comment.class);
i.putExtra("back", "main");
i.putExtra("id", postKey);
startActivity(i); )

        viewHolder.pname.setOnClickListener(new View.OnClickListener()
boolean in = true;

        @Override public void onClick(View view)

        pdatabaseReference.addValueEventListener(new ValueEventListener()
@Override
public void onDataChange(DataSnapshot dataSnapshot)

```

```

        id = dataSnapshot.child(postKey).child("uid").getValue().toString();
        userid(id);

    @Override public void onCancelled(DatabaseError databaseError)

    );

    );

    private void userid(final String id)

        getPdatabaseReferencefollowing.addValueEventListener (new ValueEventListener
        terner()
    @Override public void onDataChange(DataSnapshot dataSnapshot)

        if(dataSnapshot.child(pAuth.getCurrentUser().getUid()).hasChild(id))
        startIntent(id);

    @Override public void onCancelled(DatabaseError databaseError)

    );

    private void startIntent(String id)

        Intent i = new Intent(MainPage.this, UserHomePage.class);
        i.putExtra("user", "following");
        i.putExtra("id", id); startActivity(i); ;

        recyclerView.setAdapter(firebaseRecyclerAdapter);
    }
}

```

### B.3.3.1 ViewHolder of RecyclerView

```
public static class DisplayPostHolder extends RecyclerView.ViewHolder
```

```

        View mview; TextView pNoLikers, pcomment, ptime; TextView pname; ImageView plike; DatabaseReference pdatabaseLikes, pdatabaseusers, pdatabasePosts;
```

```

        public DisplayPostHolder(View itemView) super(itemView); mview = itemView;

        pNoLikers = (TextView)mview.findViewById(R.id.likers);
        plike = (ImageView)mview.findViewById(R.id.like);
        pcomment = (TextView) mview.findViewById(R.id.comment);
        pname = (TextView)mview.findViewById(R.id.uname);
        ptime = (TextView)mview.findViewById(R.id.time);
    }
}

```

```

public void checkUserName(final Context applicationContext, final String uid)

final ImageView pimage = (ImageView) mview.findViewById(R.id.profile);

pdatabaseusers = FirebaseDatabase.getInstance().getReference().child("/users");
pdatabaseusers.addValueEventListener(new ValueEventListener()
@Override
public void onDataChange(DataSnapshot dataSnapshot)

    if(dataSnapshot.child(uid).hasChild("profileUrl"))
String image = dataSnapshot.child(uid).child("profileUrl").getValue().toString();
Picasso.with(applicationContext).load(image).into(pimage);

@Override public void onCancelled(DatabaseError databaseError)
);

public void setUserName(Context applicationContext, final String id)

pdatabaseusers = FirebaseDatabase.getInstance().getReference().child("/users");

pdatabaseusers.addValueEventListener(new ValueEventListener()
@Override public void onDataChange(DataSnapshot dataSnapshot)

    String name = dataSnapshot.child(id).child("name").getValue().toString();
pname.setText(name);

@Override public void onCancelled(DatabaseError databaseError)
);

public void setPostImage(Context context, String image) ImageView pimage =
(ImageView)mview.findViewById(R.id.postimage);
Picasso.with(context).load(image).into(pimage);

public void setCaption(String caption)
TextView pname = (TextView)mview.findViewById(R.id.pcaption);

```

```
pname.setText(caption);
```

```
    public void setLike(final String key)
pdatabaseLikes = FirebaseDatabase.getInstance().getReference().child("/likes");
pdatabaseLikes.keepSynced(true);
```

```
        pdatabaseLikes.addValueEventListener(new ValueEventListener()
@Override public void onDataChange(DataSnapshot dataSnapshot)
if(dataSnapshot.child(key).hasChild(pAuth.getCurrentUser().getUid()))
plike.setImageResource(R.drawable.love);
else
plike.setImageResource(R.drawable.blank_love);
```

```
        long plikes = dataSnapshot.child(key).getChildrenCount();
pNoLikers.setText(Long.toString(plikes));
if (plikes == 0)
pNoLikers.setText(null);
```

```
@Override public void onCancelled(DatabaseError databaseError)
```

```
);
```

```
public void setTime(String time)
```

```
ptime.setText(time);
```