SIKANA – Teacher Assistant

JAIN UNIVERSITY

Declared as Deemed-to-be University u/s 3 of the UGC Act 1956

DEPARTMENT OF COMPUTER SCIENCE & IT

JAIN KNOWLEDGE CAMPUS
JAIN UNIVERSITY
BANGALORE- 560069.

MARCH- 2018

Annexure-I

SIKANA – Teacher Assistant

PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF

BACHELORS IN COMPUTER APPLICATION
(SPECIALIZATION MOBILE APPLICATION AND INFORMATION SECURITY)

IN

DEPARTMENT OF COMPUTER SCIENCE & IT

BY

Sirasit Thitirattanakorn
Kapil Bastola
Naina Mittal

JAIN KNOWLEDGE CAMPUS
JAIN UNIVERSITY
BANGALORE- 560069. INDIA

MARCH- 2018

Annexure - II

JGi JAIN UNIVERSITY
Declared as Deemed-to-be University u/s 3 of the UGC Act 1956

Forwarding Certificate

I hereby recommend that the project work prepared under my supervision by Sirasit Thitirattanakorn, Kapil Bastola and Naina Mittal entitled ***SIKANA*** be accepted in partial fulfillment of the requirements for the degree of BACHELORS IN COMPUTER APPLICATION.

Internal Supervisor
Department of Computer Science & IT

Annexure-III

JGi **JAIN UNIVERSITY**

Declared as Deemed-to-be University u/s 3 of the UGC Act 1956

Certificate of approval

The foregoing project work by Sirasit Thitirattanakorn (15BCA14018), Kapil Bastola (15BCA14018), and Naina Mittal (15BCA15033) is hereby approved as a creditable study carried out and presented in a manner satisfactory to warrant its acceptance as a pre-requisite to the degree for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein but approve only for the purpose for which it is submitted.

Committee
On ________________________________

Final Examination for

Evaluation of project work ________________________________

DECLARATION

I affirm that the project work titled SIKANA being submitted in partial fulfillment for the award of BCA is the original work carried out by me. It has not formed the part of any other project work submitted for award of any degree or diploma, either in this or any other University.

Sirasit Thitirattanakorn 15BCA14018

Kapil Bastola 15BCA14016

Naina Mittal 15BCA15033

ACKNOWLEDGEMENT

It is not the completion of project that is most important but more so, the interaction of roles played by various people in the satisfactory completion. We take this opportunity to express our deep gratitude and appreciation of all those who encouraged us to successfully complete the project.

With profound sense of gratitude and regards, We acknowledge with great pleasure the guidance and support extended by **Dr. Eshwaran Iyer**, Dean, Jain Knowledge Campus, Bangalore, **Dr. B.A Vasu**, Center Head, Jain Knowledge Campus Bangalore, **Dr. SJagannathan**, Head, Department of Computer Science & IT, Jain University & **Prof. Achutha V**, Department of Computer science & IT, Jain University, Bangalore for their interest & encouragement towards the completions of this project

We would like to express our deep sense of gratitude to our coordinator **Prof. Rashmi G** for her immense support and encouragement. Last but not the least our Guide **Mr. Abhishek Kumar Sarkar**, Assistant Professor, Department of BCA, Jain University, Bangalore for his/her guidance and valuable time spent with us towards successful completion of this project.

We would also thank our parents, friends & non-teaching staff for their support in completing this project.

Sirasit Thitirattanakorn 15BCA14018

Kapil Bastola 15BCA14016

Naina Mittal 15BCA15033

5. INTRODUCTION

To begin with, education is one of the most important aspects that determines success and growth rates of a country. Our project's focus is to make educational tasks easier for faculties who struggle with daily tasks such as making an appointment with colleagues, taking student attendance in classes, class scheduling, and so on.

In our current generation, mobile devices are what closely related to our lives more than what it was compared to the Nokia generation. Smartphone makes humans live easier and better through their advantages such as mobility, portability, and utility.

According to researches, Android OS is one of the fastest growing mobile operating system in the world especially in India. It would be advantageous for us to create an Android application that solve educational problems.

Our application is named as SIKANA, which is related to a Hindi word that means teaching. SIKANA has several features that solve problems that are mentioned above, this documentation contains information about features and functionality in the following chapters.

In conclusion, we hope this project makes faculties can do their educational tasks easier, faster, and better.

6. REVIEW OF LITERATURE

Lectures were traditionally optional for students in higher education with no registers or records of attendance. This flexible approach was tolerated if performance in end of year or final examinations was satisfactory. The “sign-up” system has caused resentment because students dislike being treated like school children when they are learning the attributes of a profession and some students can even resort to forgery. General attendance sheets are easily lost and are rarely inspected by faculty members while taking attendance takes a disproportionate amount of time.

- **An Online Attendance Record System (OARS)**
- H.C. Ting, T.O. Ting
- **Publisher:** IEEE

The usage for text-based communication in between two or more members of the online network is done by instant messaging which is a specified communication technology. Instant messaging permits effective and economical communication, allowing up-to-date receipt of confirmation or reply. Workmates in the company can deliver and acquire instant message quickly without one-on-one interaction and during the process instant chat session the work can be mutually shared. A virtual conference can be done without one-on-one interaction in a meeting room with the help of instant messaging. The interoffice communication is faster than the phone calls and more than one can communicate at the same duration with the help of instant messages. The usage of instant messenger is advantageous. The public can connect and have a debate in real time instead of depending on a conference call or an email message. By taking into consideration all your options, you can communicate instantly which are better than email. Instant messaging is now impending to cell phones but before email was the first application for the online network.

-**A review on instant messaging system based on cloud for smartphone**

- Swapnil Manekar, Prashant Borkar, Vedprakash Hirwani

- **Publisher:** IEEE

7. SCOPE

SIKANA’s customer target is aimed for teachers at all level, not only for higher study education. We hope that teachers can manage their mundane tasks such as taking attendance, sheets maintenance, and communication easier, faster, and better.

8. DESIGN METHODOLOGY OF PROPOSED WORK

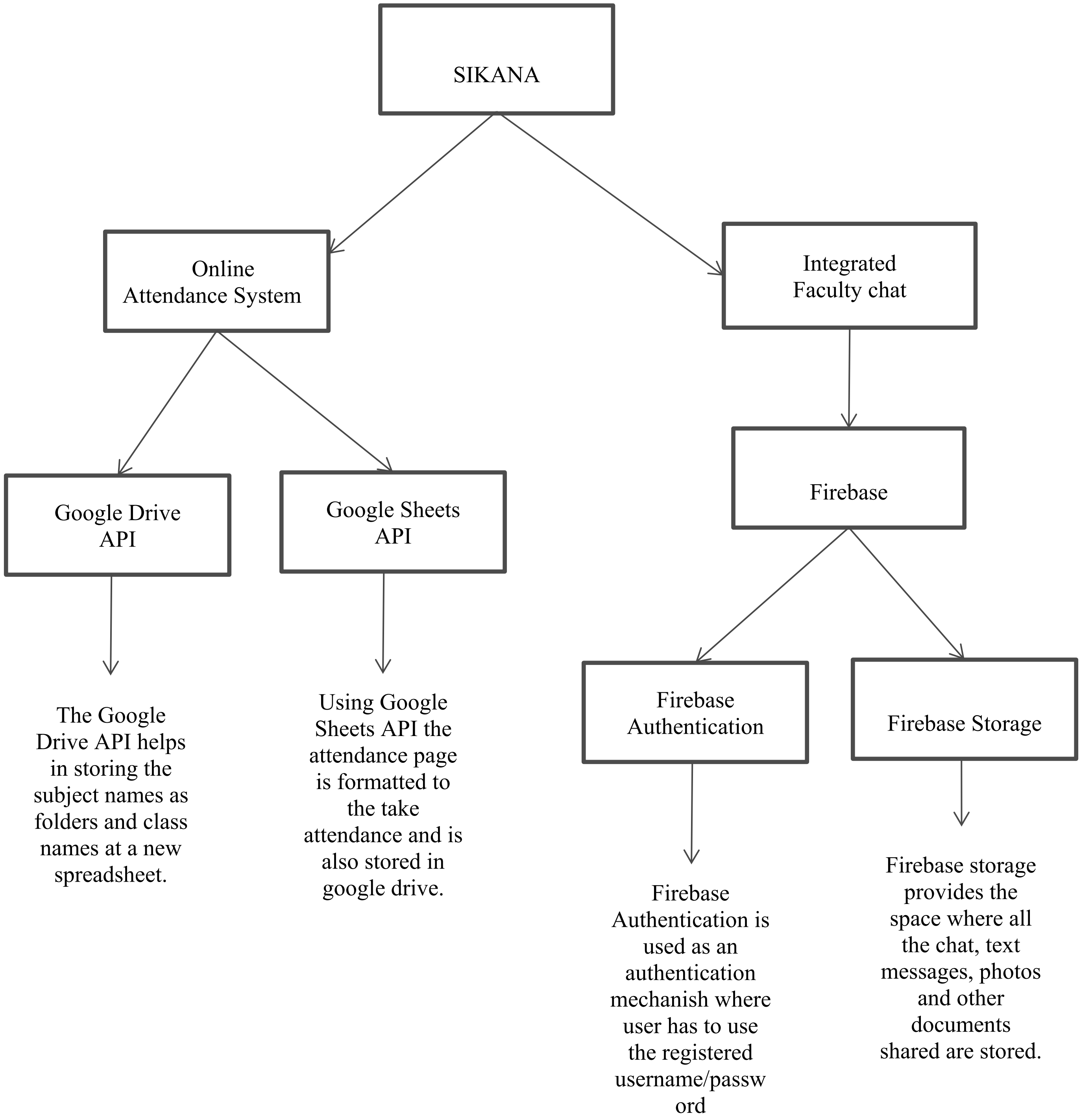

SIKANA
Online Attendance System
Integrated Faculty chat
Google Drive API
Google Sheets API
Firebase
Firebase Authentication
Firebase Storage
The Google Drive API helps in storing the subject names as folders and class names at a new spreadsheet.
Using Google Sheets API the attendance page is formatted to the take attendance and is also stored in google drive.
Firebase Authentication is used as an authentication mechanish where user has to use the registered username/password
Firebase storage provides the space where all the chat, text messages, photos and other documents shared are stored.

fig .1 : Proposed System

9. RESOURCE AND TOOLS

Android Studio

Android Studio is the official integrated development environment (IDE)[1] for Google's[2] Android operating system[3], built on JetBrains'[4] IntelliJ IDEA[5] software and designed specifically for Android development[6]. It is available for download on Windows, macOS and Linux based operating systems. It is a replacement for the Eclipse_ADT as primary IDE for native Android application development.

Firebase

Firebase is a mobile and web application development platform developed by Firebase, Inc. in 2011, then acquired by Google in 2014.

Firebase evolved from Envolve, a prior startup founded by James Tamplin and Andrew Lee in 2011. Envolve provided developers an API that enables the integration of online chat functionality into their websites. After releasing the chat service, Tamplin and Lee found that it was being used to pass application data that weren't chat messages. Developers were using Envolve to sync application data such as game state in real time across their users. Tamplin and Lee decided to separate the chat system and the real-time architecture that powered it

Google sheet

Google sheets is a spread sheet, all part of a free, web-based software office suite offered by Google within its Google Drive service. The app is available as web applications, mobile apps for Android and iOS, and desktop applications on Google's ChromeOS. The app is compatible with Microsoft Office file formats. The suite also includes Google Forms(survey software), Google Drawings (diagramming software), Google Sites (web building software), Google My Maps (map overlay editor), Google Apps Script (code editor for the G-Apps Script coding language) and Google Fusion Tables (database manager; experimental).

Google Drive

Google Drive is a file storage and synchronization service developed by Google. Launched on April 24, 2012, Google Drive allows users to store files on their servers, synchronize files across devices, and share files. In addition to a website, Google Drive offers apps with offline capabilities for Windows and macOS computers, and Android and iOS smartphones and tablets. Google Drive encompasses Google Docs, Sheets and Slides, an office suite that permits collaborative editing of documents, spreadsheets, presentations, drawings, forms, and more. Files created and edited through the office suite are saved in Google Drive.

10. IMPLEMENTATION AND RESULTS

10.1 Firebase Authentication

10.1.1 Firebase Set-up

Procedure:

1. Add Firebase to your Android project.
2. Add the dependency for Firebase Authentication to your app-level `build.gradle` file:

```
compile 'com.google.firebase:firebase-auth:11.8.0'
```

3. If you haven't yet connected your app to your Firebase project, do so from the Firebase console.
4. Enable Email/Password sign-in:
    a. In the Firebase console, open the **Auth** section.
    b. On the **Sign in method** tab, enable the **Email/password** sign-in method and click **Save**.

10.1.2 User registration in Firebase

To create a new user account with a password, complete the following steps in your app's sign-in activity:

1. In your sign-up activity's `onCreate` method, get the shared instance of the `FirebaseAuth` object:

```
private FirebaseAuth mAuth;
// ...
mAuth = FirebaseAuth.getInstance();
```

2. When initializing your Activity, check to see if the user is currently signed in:

```
@Override
public void onStart() {
    super.onStart();
    // Check if user is signed in (non-null) and update UI accordingly.
    FirebaseUser currentUser = mAuth.getCurrentUser();
    updateUI(currentUser);
}
```

3. When a new user signs up using your app's sign-up form, complete any new account validation steps that your app requires, such as verifying that the new account's password was correctly typed and meets your complexity requirements.
4. Create a new account by passing the new user's email address and password to `createUserWithEmailAndPassword`:

```
mAuth.createUserWithEmailAndPassword(email, password)
        .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {
                if (task.isSuccessful()) {
                    // Sign in success, update UI with the signed-in user's information
                    Log.d(TAG, "createUserWithEmail:success");
                    FirebaseUser user = mAuth.getCurrentUser();
                    updateUI(user);
                } else {
                    // If sign in fails, display a message to the user.
                    Log.w(TAG, "createUserWithEmail:failure", task.getException());
                    Toast.makeText(EmailPasswordActivity.this, "Authentication
failed.",
                            Toast.LENGTH_SHORT).show();
                    updateUI(null);
                }

                // ...
            }
        });
```

If the new account was created, the user is also signed in. In the callback, you can use the getCurrentUsermethod to get the user's account data.

10.1.3 User sign-in in Firebase

The steps for signing in a user with a password are similar to the steps for creating a new account. In your app's sign-in activity, do the following:

1. In your sign-in activity's onCreate method, get the shared instance of the FirebaseAuth object:

```
private FirebaseAuth mAuth;
// ...
mAuth = FirebaseAuth.getInstance();
```

2. When initializing your Activity, check to see if the user is currently signed in:

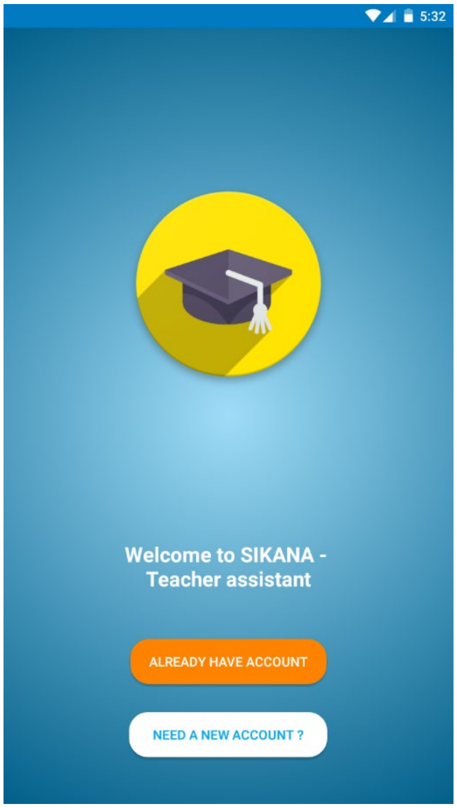
```
@Override
public void onStart() {
    super.onStart();
    // Check if user is signed in (non-null) and update UI accordingly.
    FirebaseUser currentUser = mAuth.getCurrentUser();
    updateUI(currentUser);
}
```

3. When a user signs in to your app, pass the user's email address and password to signInWithEmailAndPassword:

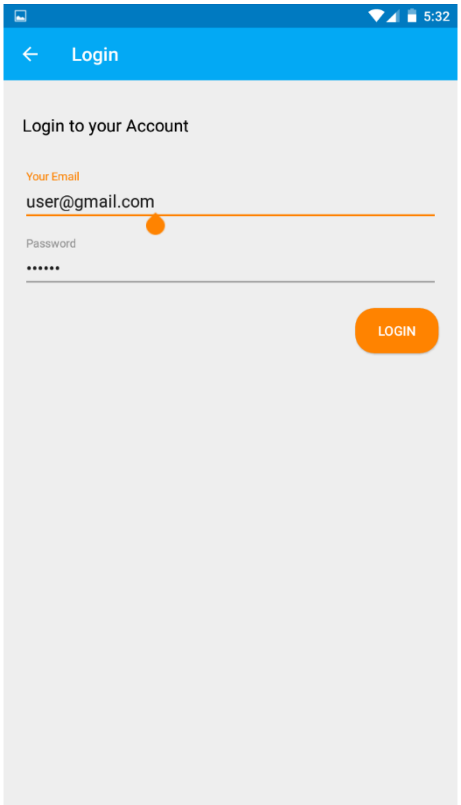
```
mAuth.signInWithEmailAndPassword(email, password)
        .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {
                if (task.isSuccessful()) {
                    // Sign in success, update UI with the signed-in user's information
                    Log.d(TAG, "signInWithEmail:success");
                    FirebaseUser user = mAuth.getCurrentUser();
                    updateUI(user);
                } else {
                    // If sign in fails, display a message to the user.
                    Log.w(TAG, "signInWithEmail:failure", task.getException());
                    Toast.makeText(EmailPasswordActivity.this, "Authentication failed.",
                            Toast.LENGTH_SHORT).show();
                    updateUI(null);
                }

                // ...
            }
        });
```

If sign-in succeeded, you can use the returned FirebaseUser to proceed.

10.1.4 Screenshots



Welcome Screen Activity

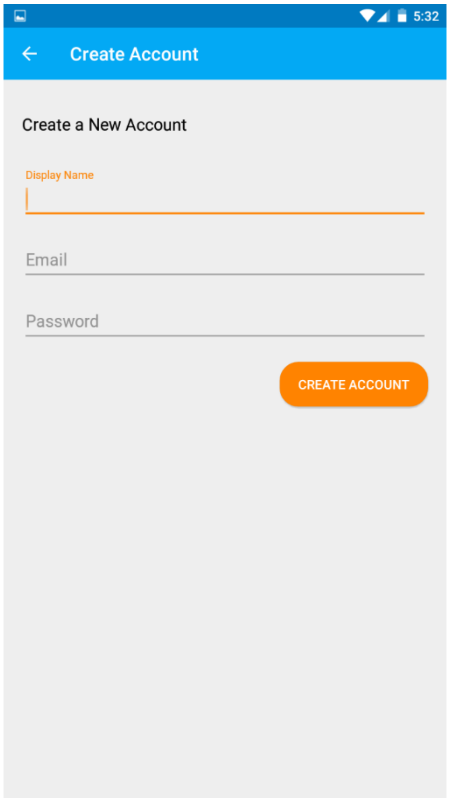


Login to your Account

Your Email
user@gmail.com

Password
••••••

Sign in Activity



5:32

Create Account

Create a New Account

Sign up Activity

10.2 Profile update

In this module, we use Firebase Database feature to store users' data such as name, status, and image url.

The procedure for profile data storing is as following:

1. Create a Firebase Database reference using getRegerence()

```
mDatabase = FirebaseDatabase.getInstance().getReference()
.child("Users").child(uid);
```

2. Create a HashMap object to store users' data

```
HashMap<String, String> userMap = new HashMap<>();
userMap.put("name", display_name);
userMap.put("status", "Hi there I'm using SIKANA.");
userMap.put("image", "default");
userMap.put("thumb_image", "default");
userMap.put("device_token", device_token);
```

3. Upload users’ image using Picasso

```
Picasso.with(SettingsActivity.this)       .load(image)
.networkPolicy(NetworkPolicy.OFFLINE)
.placeholder(R.drawable.default_avatar) .into(mDisplayImage);
```

10.2.1 Screenshots

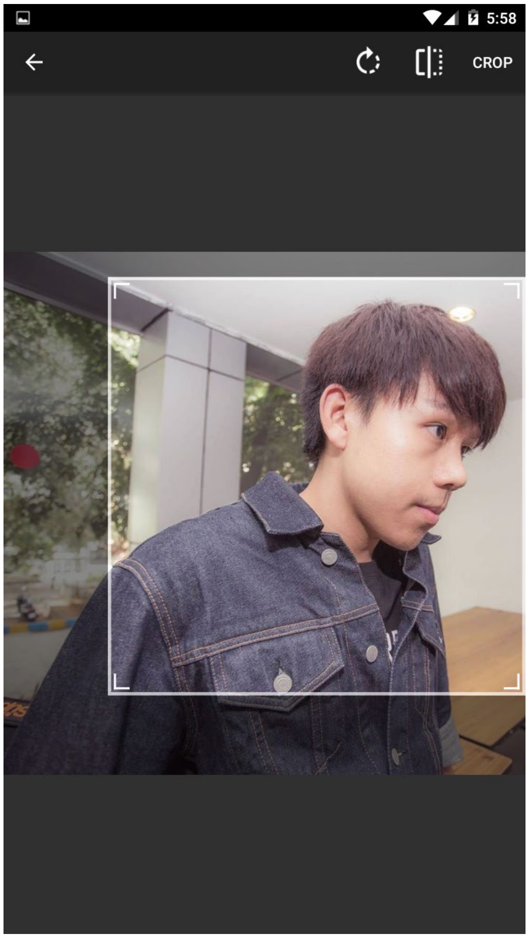
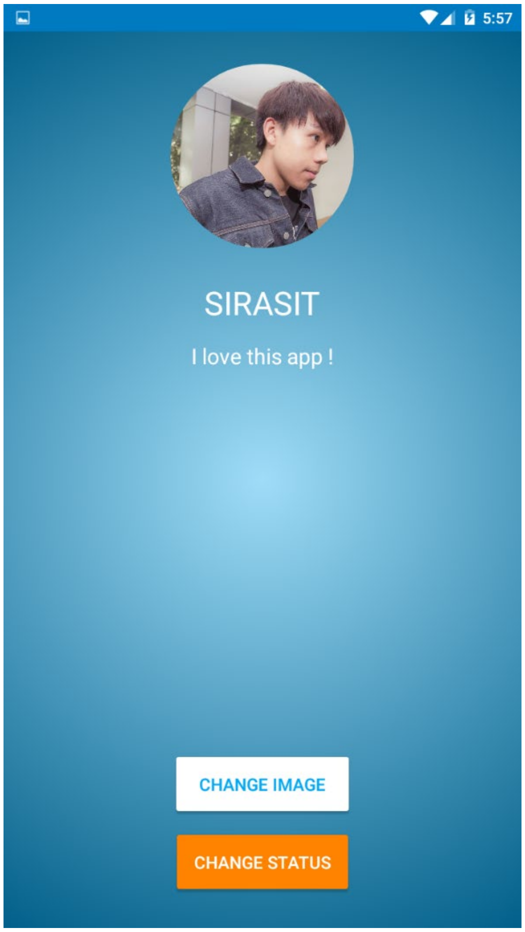


Image Croping

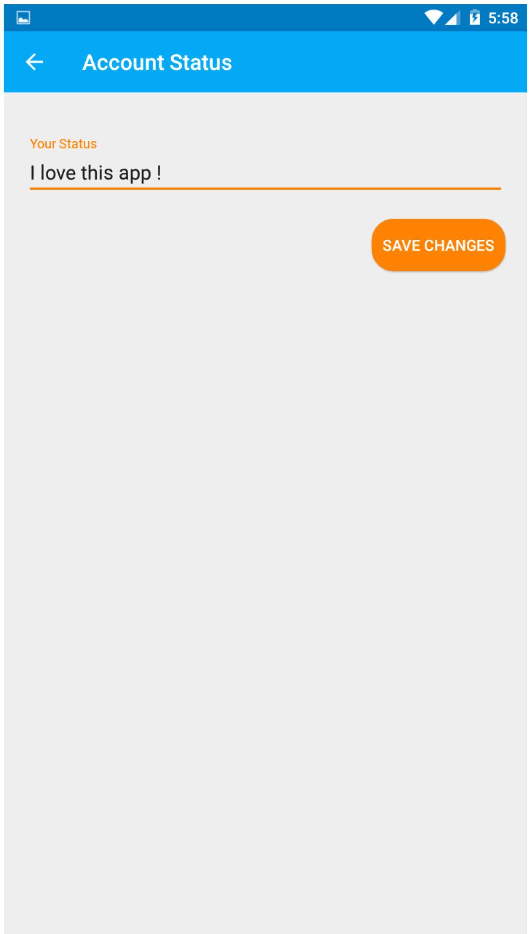
10.2.1 Screenshots



SIRASIT

I love this app !

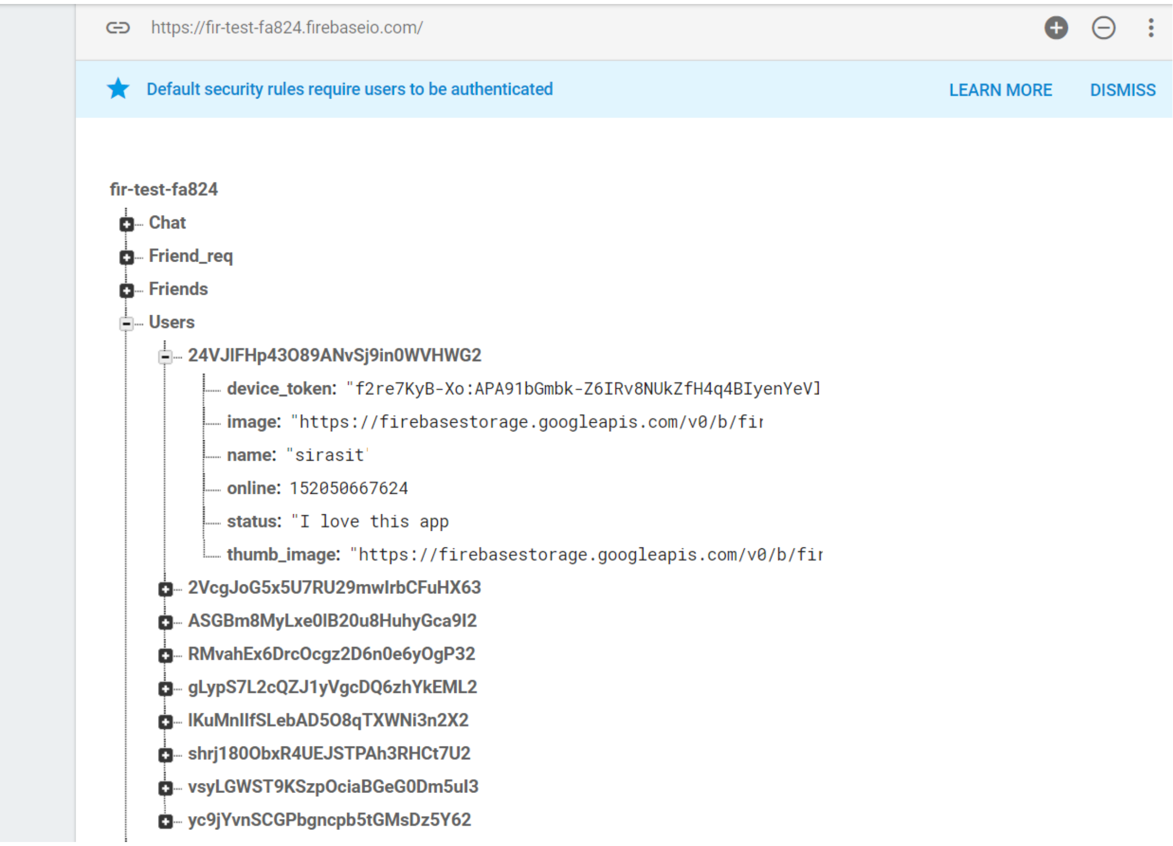
Profile Activity



Your Status

I love this app !

Status Update



Default security rules require users to be authenticated

```
fir-test-fa824
  Chat
  Friend_req
  Friends
  Users
    24VJlFHp43O89ANvSj9in0WVHWG2
      device_token: "f2re7KyB-Xo:APA91bGmbk-Z6IRv8NUkZfH4q4BIyenYeV]
      image: "https://firebasestorage.googleapis.com/v0/b/fi
      name: "sirasit
      online: 152050667624
      status: "I love this app
      thumb_image: "https://firebasestorage.googleapis.com/v0/b/fi
    2VcgJoG5x5U7RU29mwlrbCFuHX63
    ASGBm8MyLxe0lB20u8HuhyGca9l2
    RMvahEx6DrcOcgz2D6n0e6yOgP32
    gLypS7L2cQZJ1yVgcDQ6zhYkEML2
    lKuMnllfSLebAD5O8qTXWNi3n2X2
    shrj180ObxR4UEJSTPAh3RHCt7U2
    vsyLGWST9KSzpOciaBGeG0Dm5ul3
    yc9jYvnSCGPbgncpb5tGMsDz5Y62
```

Firebase Database Structure

10.3 Sending Request

In this module, we connect users using request sending-responding relationship, which means 2 peers can communicate only when they accept each other's friend requests. The steps are described below:

1. Create a new repository in the firebase database called as "**Friend_req**"

```
mFriendReqDatabase = FirebaseDatabase
.getInstance().getReference().child("Friend_req");
```

2. Send a friend request from Profile Activity. Once the request is sent, the text appears as pending on the sender's screen, and accept on the receiver screen.

```
mFriendReqDatabase.child(mCurrent_user.getUid()).addListenerForSin
gleValueEvent(new ValueEventListener() {
```

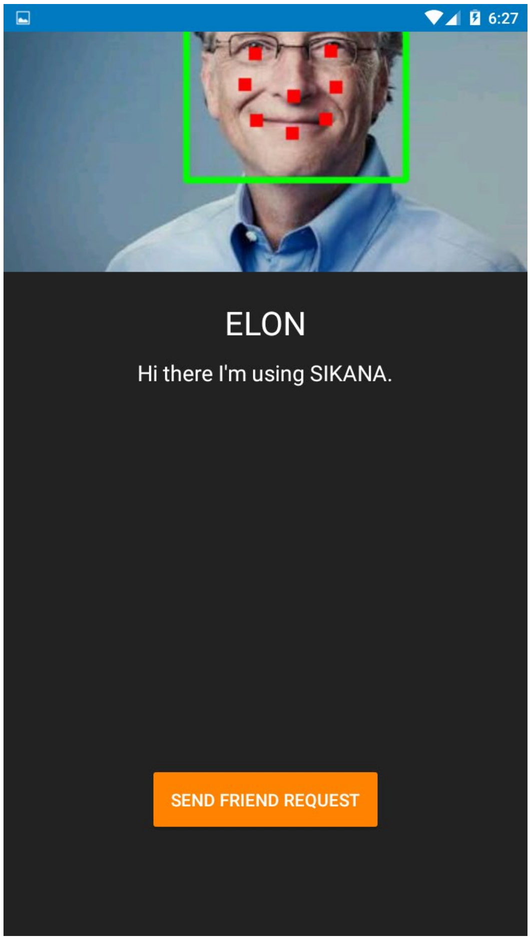
```
@Override
public void onDataChange(DataSnapshot dataSnapshot) {
if(dataSnapshot.hasChild(user_id)){
String req_type =
dataSnapshot.child(user_id).child("request_type").getValue().toString();

if(req_type.equals("received")){
mCurrent_state = "req_received";
mProfileSendReqBtn.setText("Accept Friend Request");
mDeclineBtn.setVisibility(View.VISIBLE);
mDeclineBtn.setEnabled(true);
} else if(req_type.equals("sent")) {
mCurrent_state = "req_sent";
mProfileSendReqBtn.setText("Cancel Friend Request");
mDeclineBtn.setVisibility(View.INVISIBLE);
mDeclineBtn.setEnabled(false);
}
```

3. Once the request is accepted, the relationship for both users turns from not_friend into friends

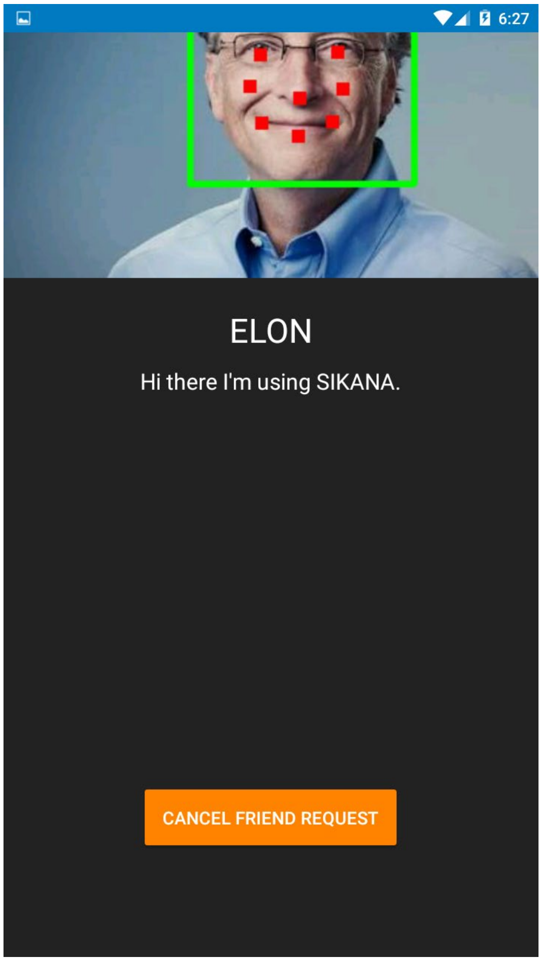
```
if(mCurrent_state.equals("req_received")){
final String currentDate = DateFormat.getDateTimeInstance().format(new
Date());

Map friendsMap = new HashMap();
friendsMap.put("Friends/" + mCurrent_user.getUid() + "/" + user_id +
"/date", currentDate);
friendsMap.put("Friends/" + user_id + "/"  + mCurrent_user.getUid() +
"/date", currentDate);
```

```
friendsMap.put("Friend_req/" + mCurrent_user.getUid() + "/" + user_id,
null);
friendsMap.put("Friend_req/" + user_id + "/" + mCurrent_user.getUid(),
null);


              mRootRef.updateChildren(friendsMap, new
DatabaseReference.CompletionListener() {
@Override public void onComplete(DatabaseError databaseError,
DatabaseReference databaseReference) {
if(databaseError == null){
mProfileSendReqBtn.setEnabled(true);
mCurrent_state = "friends";
                     mProfileSendReqBtn.setText("Unfriend this Person");

                      mDeclineBtn.setVisibility(View.INVISIBLE);
                      mDeclineBtn.setEnabled(false);
} else {
String error = databaseError.getMessage();
Toast.makeText(ProfileActivity.this, error,
Toast.LENGTH_SHORT).show(); }
```

10.3.1 Screenshots:



Hi there I'm using SIKANA.

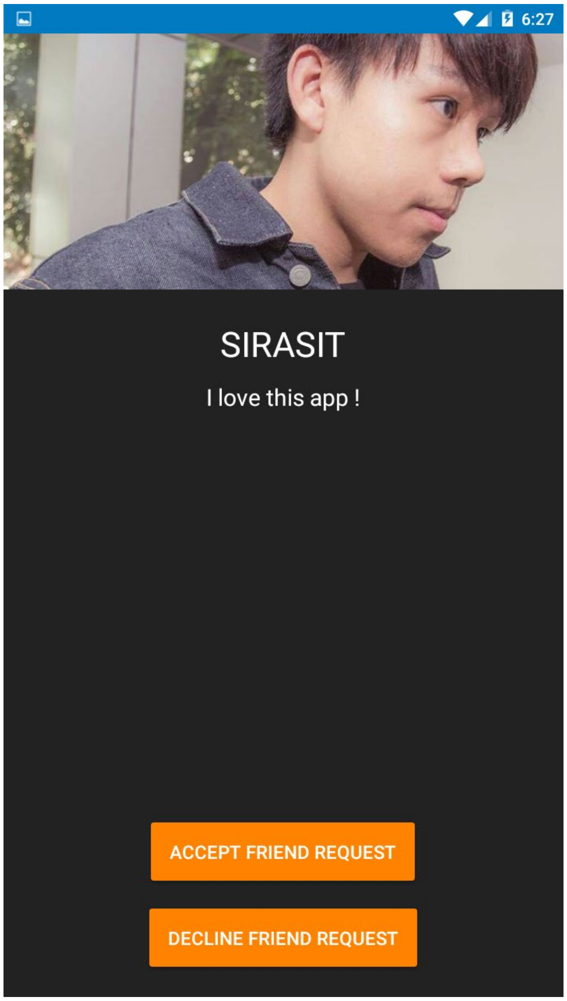


Hi there I'm using SIKANA.

Sending a request from SIRASIT to ELON

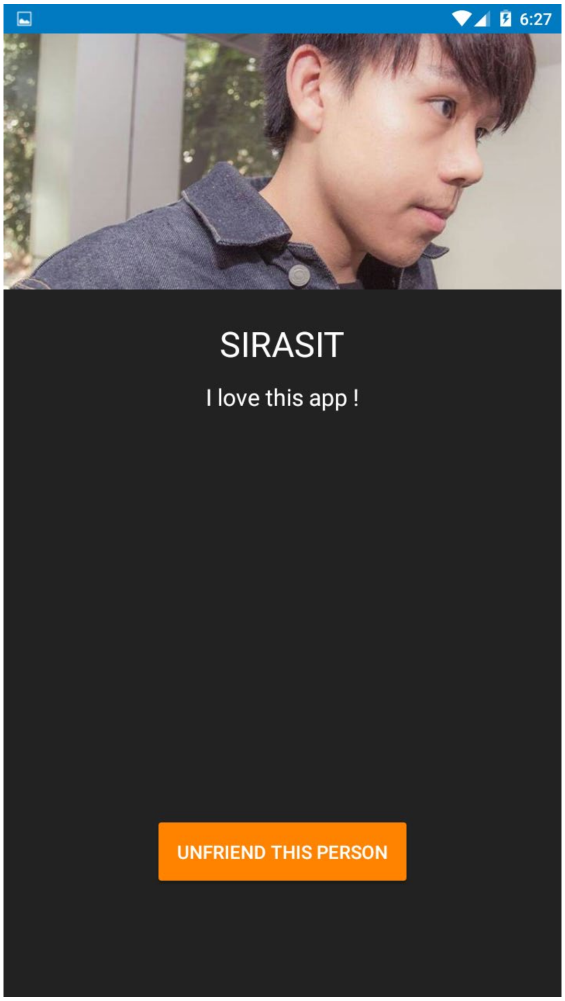
Request is sent from SIRASIT to ELON

10.3.1 Screenshots:


6:27

SIRASIT

I love this app !


6:27

SIRASIT

I love this app !

ELON accept SIRASIT's request

Both peers have become friends (and can be unfriended as well)

10.4 Messaging using Firebase Database

In this module, we use Firebase Database to store users’ chat messages.
The procedure is discussed below:

1. Create a new repository called messages

```
DatabaseReference user_message_push = mRootRef.child("messages")
.child(mCurrentUserId).child(mChatUser).push();
```

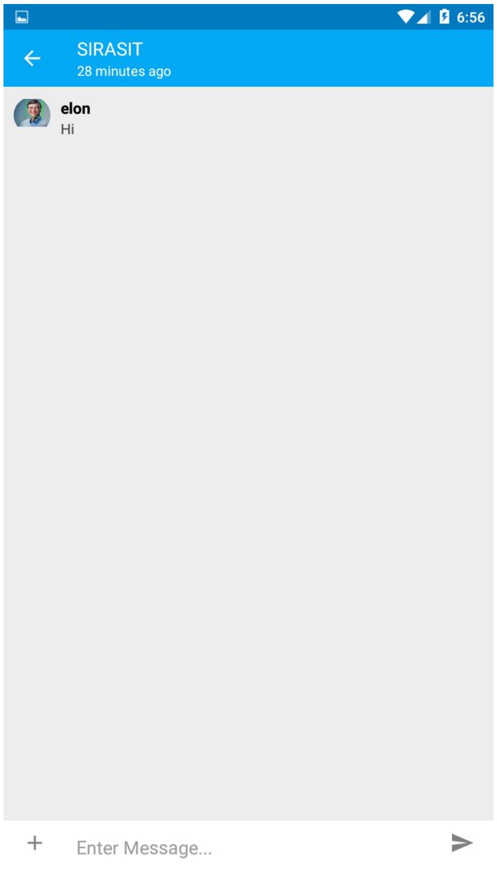
2. Create a HashMap to store information about a conversation such as messages, timestamp, sender, type, etc.

```
Map messageMap = new HashMap();
messageMap.put("message", message);
messageMap.put("seen", false);
messageMap.put("type", "text");
messageMap.put("time", ServerValue.TIMESTAMP);
messageMap.put("from", mCurrentUserId);
```

3. Use updateChildren() to update multiple values to a database repositoty(messages)

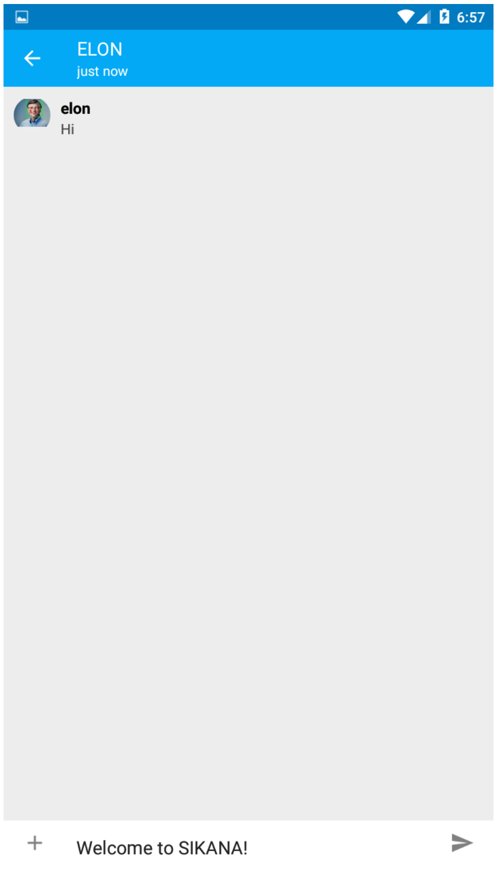
```
mRootRef.updateChildren(messageUserMap, new
DatabaseReference.CompletionListener() {
@Override
public void onComplete(DatabaseError databaseError, DatabaseReference
databaseReference) {
if(databaseError != null){
Log.d("CHAT_LOG", databaseError.getMessage().toString());
}}});
```

10.4.1 Screenshots:



elon
Hi

Message sent from ELON

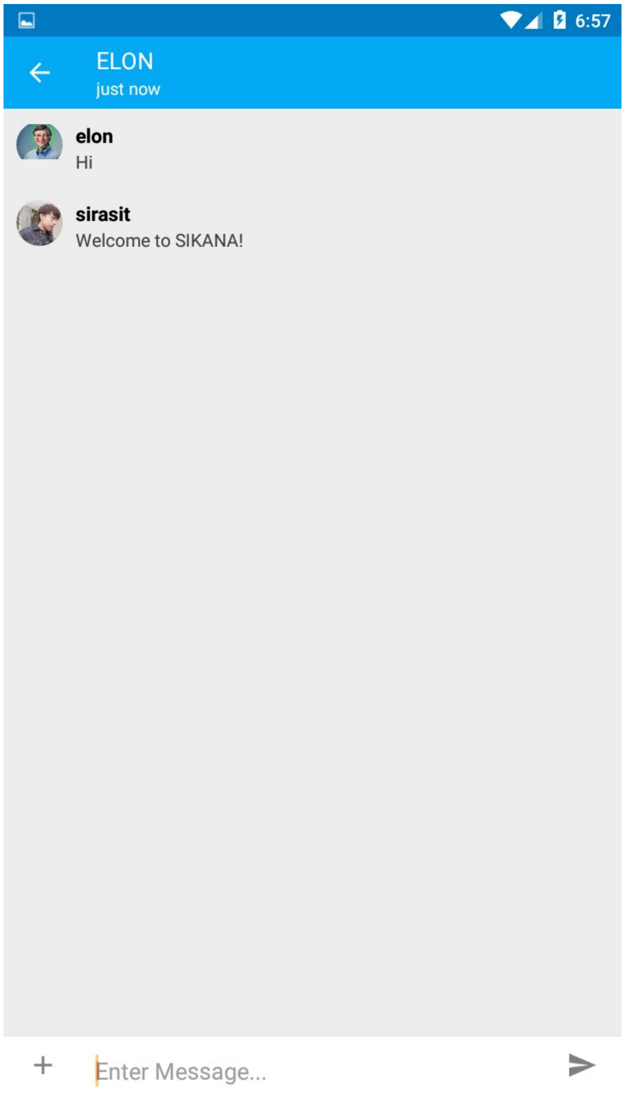


elon
Hi

Welcome to SIKANA!

SIRASIT is typing a message

10.4.1 Screenshots:

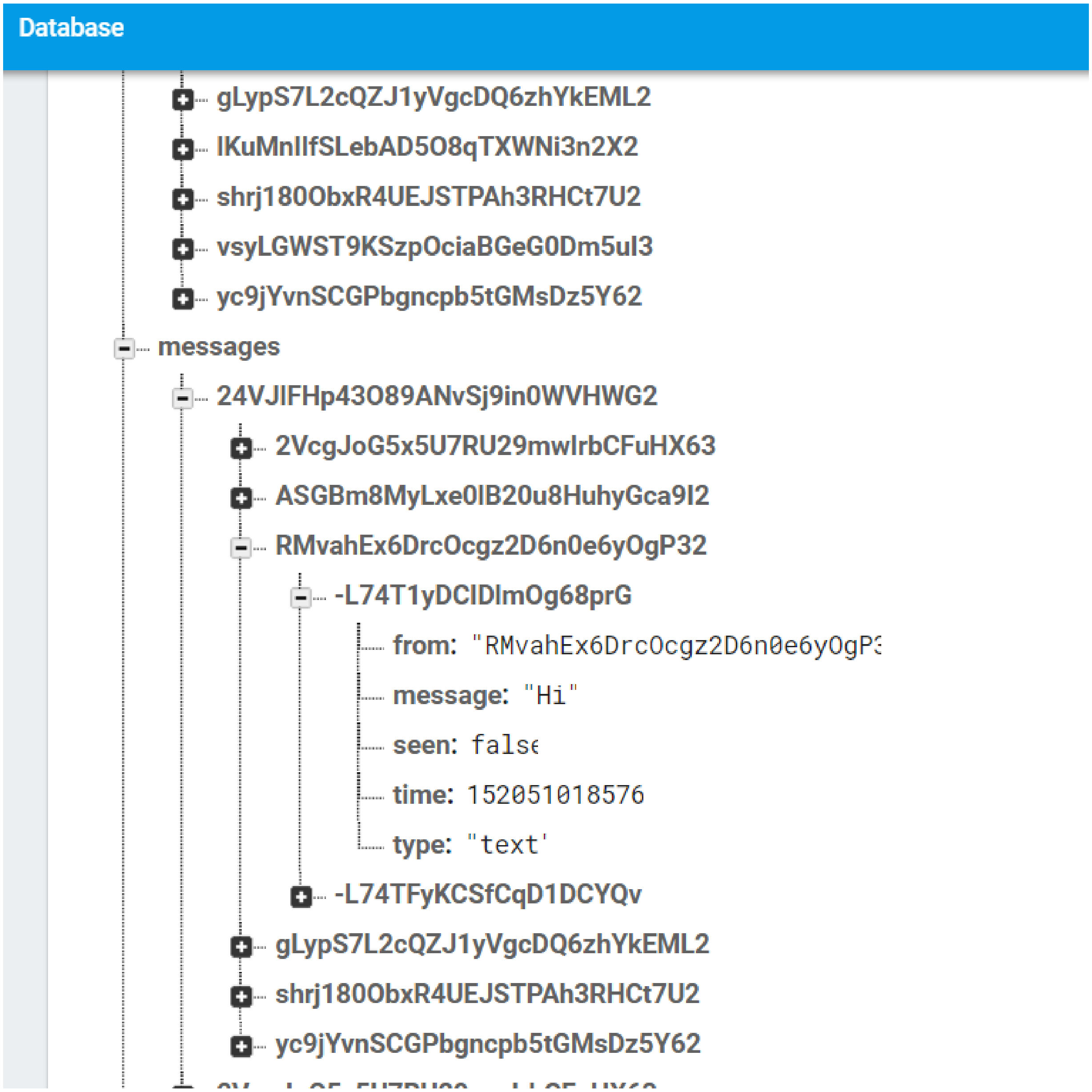


elon
Hi

sirasit
Welcome to SIKANA!

SIRASIT sent a message to ELON

10.4.1 Screenshots:



```
Database
        gLypS7L2cQZJ1yVgcDQ6zhYkEML2
        IKuMnIlfSLebAD5O8qTXWNi3n2X2
        shrj180ObxR4UEJSTPAh3RHCt7U2
        vsyLGWST9KSzpOciaBGeG0Dm5ul3
        yc9jYvnSCGPbgncpb5tGMsDz5Y62
    messages
        24VJlFHp43O89ANvSj9in0WVHWG2
            2VcgJoG5x5U7RU29mwIrbCFuHX63
            ASGBm8MyLxe0lB20u8HuhyGca9I2
            RMvahEx6DrcOcgz2D6n0e6yOgP32
                -L74T1yDCIDlmOg68prG
                    from: "RMvahEx6DrcOcgz2D6n0e6yOgP3
                    message: "Hi"
                    seen: false
                    time: 152051018576
                    type: "text'
                -L74TFyKCSfCqD1DCYQv
            gLypS7L2cQZJ1yVgcDQ6zhYkEML2
            shrj180ObxR4UEJSTPAh3RHCt7U2
            yc9jYvnSCGPbgncpb5tGMsDz5Y62
```

Message Structure in firebase

10.5 Online Attendance System

10.5.1: Using Google Drive To Store Attendance Sheets

In this module we will be storing all the attendance sheets in the google drive using Google Drive API. Using this module a faculty can create a new subject (i.e creating a folder with the subject name) and can store the attendance sheets of different classes according to the subject he/she teaches. The whole process can be divided as:-

a. Creating a new subject if doesn’t exists.

b. Showing all the subject created before.

c. Selecting a created subject to view contents.

d.Showing all the attendace sheets of a particular subject.

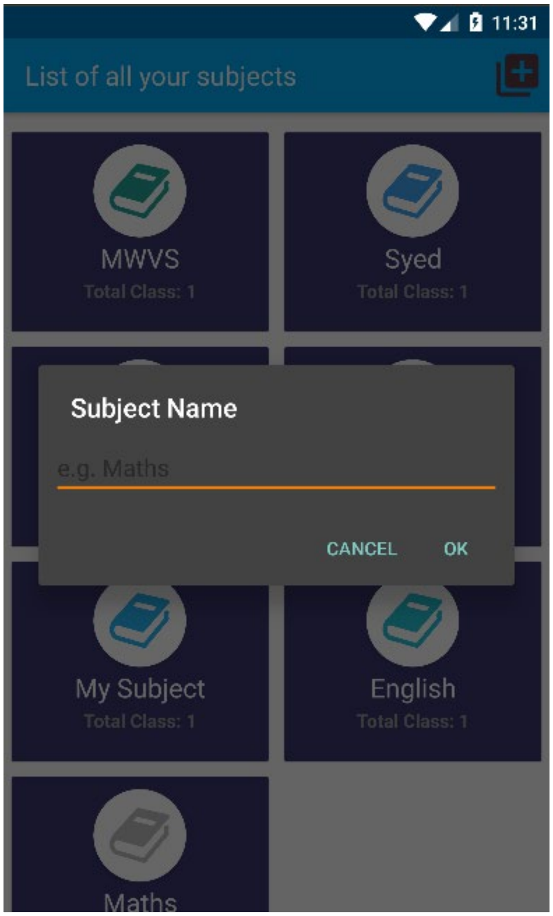
e. Creating a new attendance sheet.

10.5.1.1Creating a new subject if doesn’t exists.

In this step, the faculty member can create a new subject which will automatically sync with Google Drive API and create a folder with the named provided.

```
private void addNewFolder() throws IOException {
```

```
HttpTransport transport = AndroidHttp.newCompatibleTransport();
JsonFactory jsonFactory = JacksonFactory.getDefaultInstance();

//using Goolge Drive API object i.e. ‘Drive’
private Drive driveService = new Drive.Builder(transport, jsonFactory,
credential)
        .setApplicationName("SheetDemo")
        .build();


//initialize a new google drive folder object
            File fileMetadata = new File();
            fileMetadata.setName(getNewSubjectName());
            fileMetadata.setMimeType("application/vnd.google-apps.folder");


//create a new google folder (google drive api is used here)
            File file = driveService.files().create(fileMetadata)
                    .setFields("id, name, spaces, parents")
                    .execute();
        }
```

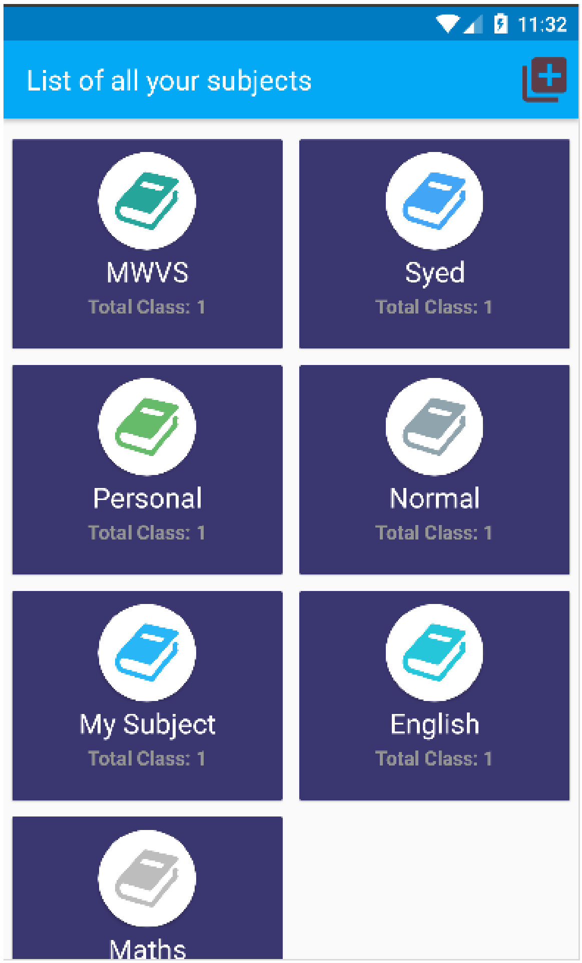


Screenshot: adding new subject

10.5.1.2 Showing all the subject created before.

Using this functionality the faculty can see all the previously created subjects (google drive folders) stored. Subject name and total number of attendance sheets inside it would be shown to the faculty giving more info about the subject. The subjects will be shown in grid view with using an adapter.

```
private void listDriveFolders() throws IOException {

HttpTransport transport = AndroidHttp.newCompatibleTransport();
JsonFactory jsonFactory = JacksonFactory.getDefaultInstance();

//using Goolge Drive API object i.e. ‘Drive’
private Drive driveService = new Drive.Builder(transport, jsonFactory, credential)
        .setApplicationName("SheetDemo")
        .build();

    String pageToken = null;
    do {


//here drive service is used to list all the folders of the user
//which are not in trash.
        FileList foldersList = driveService.files().list()
                .setSpaces("drive")
                .setQ("mimeType = 'application/vnd.google-apps.folder' and trashed = false")
                .setPageToken(pageToken)
                .setFields("nextPageToken, files(id, name, parents)")
                .execute();
        for (File folders:foldersList.getFiles()){
            byte classes = listDriveFiles(folders.getId());
            driveFolders.add(new Folders(folders.getName(),folders.getId(),classes));
        }


        pageToken = foldersList.getNextPageToken();
    } while (pageToken != null);
}
```



List of all your subjects

MWVS
Total Class: 1

Syed
Total Class: 1

Personal
Total Class: 1

Normal
Total Class: 1

My Subject
Total Class: 1

English
Total Class: 1

Maths

Screenshot : displaying all the subjects

10.5.1.3 Selecting a created subject to view classes (contents).

Now as we have already created a subject before; we would want to see what are the contents of the subject. That would be handled by the use of adapter. On clicking any of the subjects icon the google drive api request will be sent which will return all the contents of the folder (but here as we are working on attendance sheets, we have filtered the contents to be only attendance sheets).

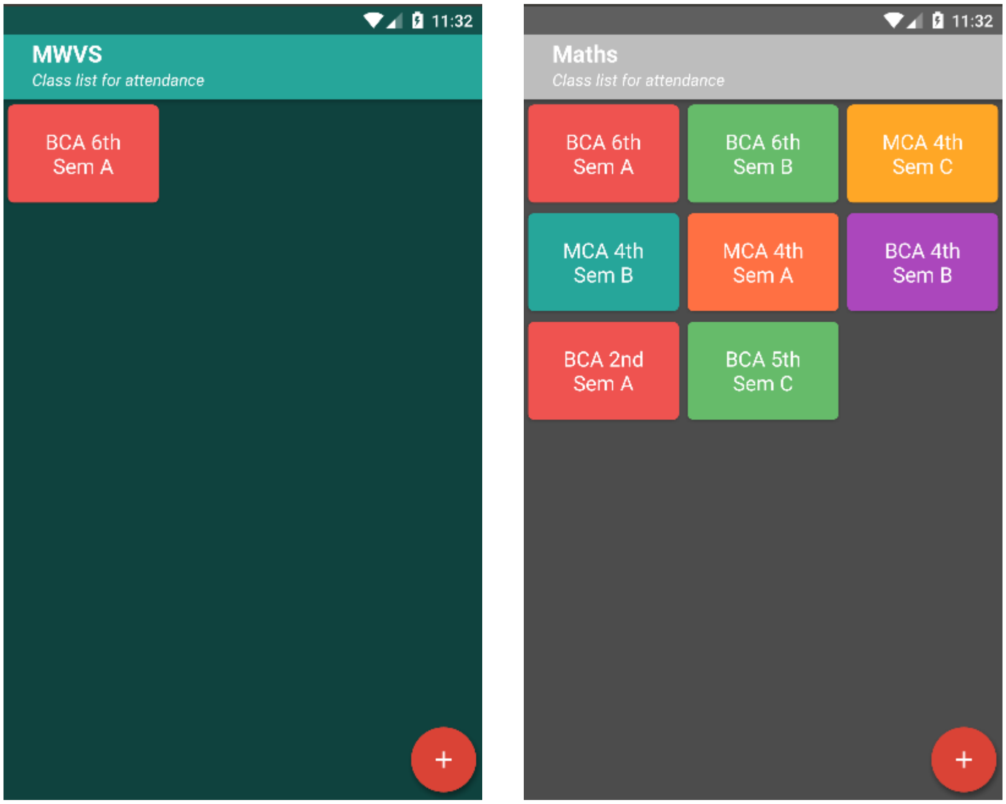
```
holder.myView.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent showSheets = new Intent(context, DisplaySheetsActivity.class);
        showSheets.putExtra("folderId",folderId);
        showSheets.putExtra("selectedAccount",selectGoogleAccount);
        showSheets.putExtra("subjectName",folders.getFolderName());
        showSheets.putExtra("actionBarColor",color);
        if (android.os.Build.VERSION.SDK_INT >=
android.os.Build.VERSION_CODES.LOLLIPOP) {
            ActivityOptions options =
ActivityOptions.makeSceneTransitionAnimation((Activity)context,holder.subjectName,"
totalClasses");
            context.startActivity(showSheets,options.toBundle());
        }else {
            context.startActivity(showSheets);
        }
    }
});
```

10.5.1.4 Showing all the classes (attendace sheets) of a particular subject.

As we have selected a particular subject in the previous step, now we will like to see the actual contents related to the subject. Here we will use the response returned by google drive api from the previous step.

```
private void listDriveFiles() throws IOException {

   HttpTransport transport = AndroidHttp.newCompatibleTransport();
   JsonFactory jsonFactory = JacksonFactory.getDefaultInstance();
   private Drive driveService = new Drive.Builder(transport, jsonFactory, credential)
                                    .setApplicationName("SheetDemo")
                                    .build();



   String pageToken = null;

    //filtering that we only need to show spreadsheets
   String query = "mimeType = 'application/vnd.google-apps.spreadsheet' and trashed =
    false and \'"+subjectId +"\' in parents";

   do {

      //api request if fired to get the contents of the folder
      FileList result = driveService.files().list()
             .setSpaces("drive")
             .setQ(query)
             .setPageToken(pageToken)
             .setFields("nextPageToken, files(id, name, parents)")
             .execute();
```

```
        for (File file : result.getFiles()) {
            //System.out.println("files:"+file.getParents());
            classList.add(new SheetsModel(file.getName(),file.getId()));
        }

        pageToken = result.getNextPageToken();
    } while (pageToken != null);
}
```

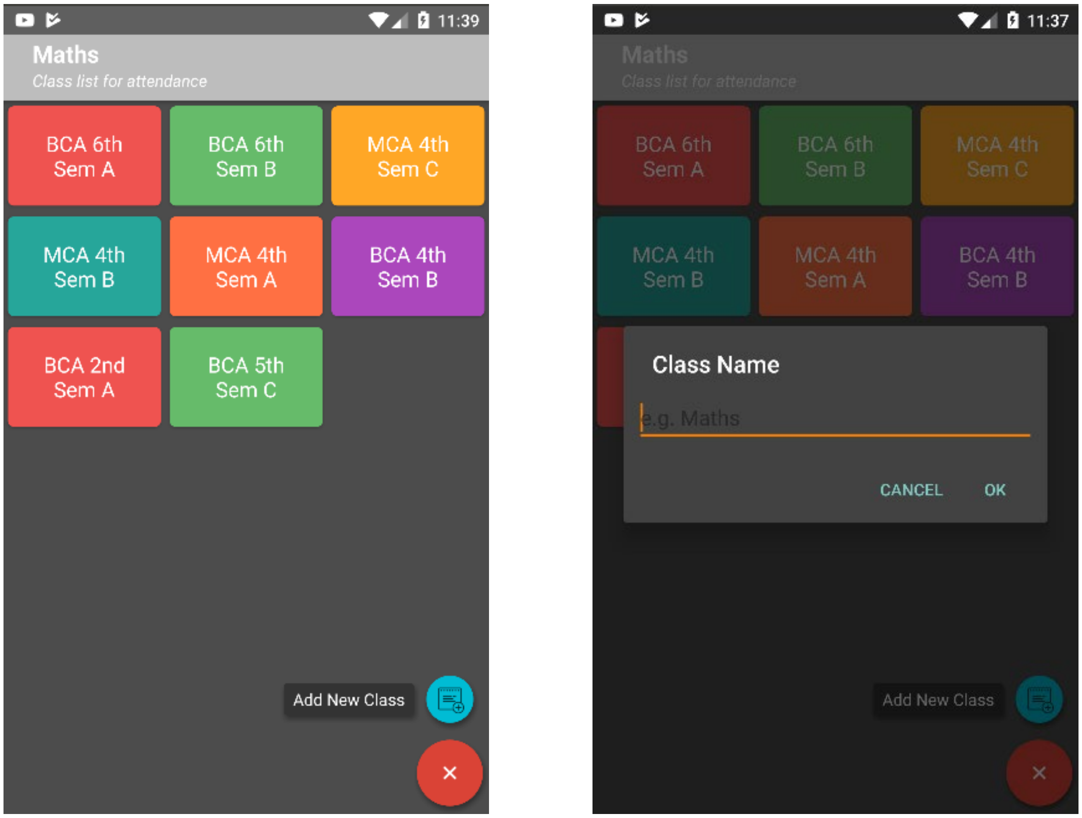


Screenshot: showing the list of all attendance sheets.

10.5.1.5 Creating a new attendance sheet.

Now that all the classes (attendance sheets) regarding the subject is shown to the user (the attendace sheet is shown in gridview with attendace sheet name e.g. BCA6thSem. **Note**: The spreadsheet view won’t be visible only an icon with attendence sheet name will be shown) , the user also can add a new attendace sheet of that subject. This very created attendace sheet will be stored in drive using drive api.

```
private void addNewSheet() throws IOException {



    File spreadSheetFile = new File();
    List<String> parents = new ArrayList<>();
    parents.add(subjectId);
    spreadSheetFile.setName(sheetname);//sheetname will be provided by the user
    spreadSheetFile.setMimeType("application/vnd.google-apps.spreadsheet");
    spreadSheetFile.setParents(parents);

    File spreadsheet = driveService.files().create(spreadSheetFile)
            .setFields("id, name, spaces, parents")
            .execute();
}
```



Screenshot : Adding new class/attendance sheet

10.5.2 Taking Attendance and Updating the attendace Sheet.

As we have already created the attendance sheet in the previous module, this module will perform the tasks like taking attendance, updating attendance sheet and formatting the attendance sheet. Thus this module can be broken down in three main tasks as:

a. Taking attendance

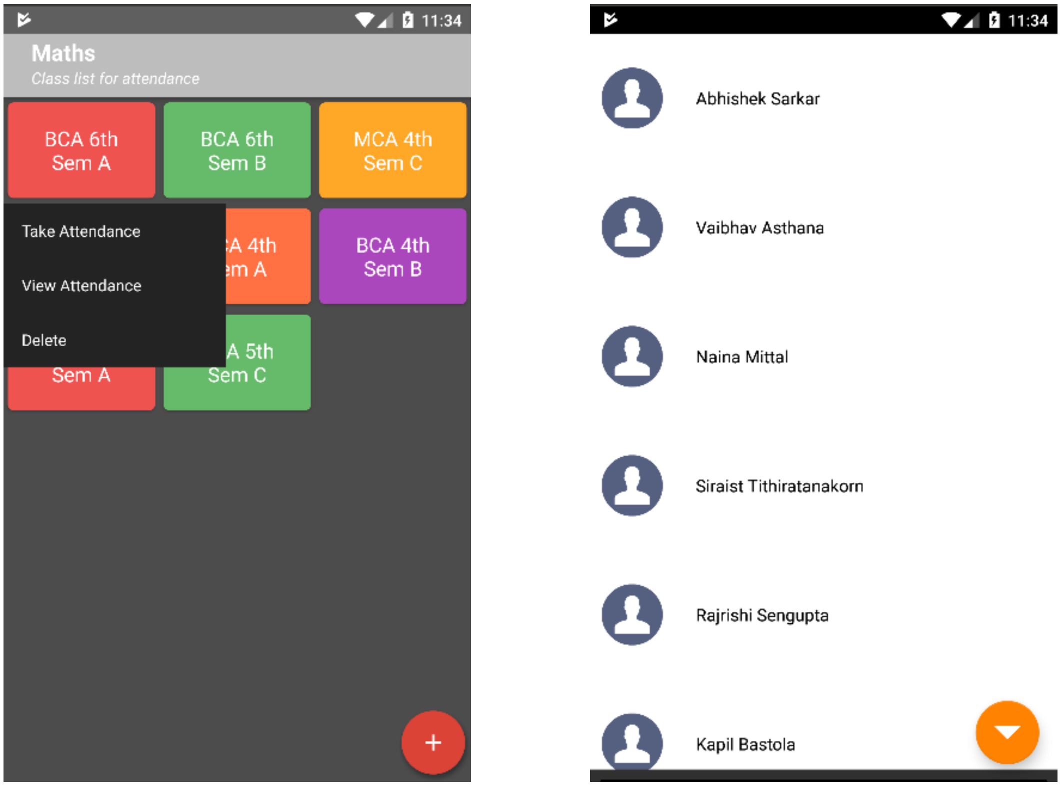
b. Updating Attendance Sheet

c. Formatting Attendance Sheet

10.5.2.1Taking attendance.

Whenever the faculty member wants to take attendance of any class, he can do it so by clicking on the attendance sheet icon which opens a new activity/window showing the names of students of that class. And by clicking on their name they will be marked as present, and if not clicked they will be marked as absent. All the students data is fetched from google sheets using Google Sheets API v4.

```
holder.myView.setOnClickListener(new View.OnClickListener() {
    boolean isClicked = false;
    @Override
    public void onClick(View v) {
        /**
         * When absent button is pressed the background color and text color is changes.
         * And also with that change we have to change the color of the absent button.
         * This is required when the user wants to change the button click. i.e.
         * have already clicked a button before but again want to change the decision
         * */

        if(!isClicked) {
            isClicked = true;
            attendanceTracker.put(studentName,(byte)(currentAttendance+1));
            booleanAttendance.put(studentName,(byte)1);

holder.student_name.setTextColor(v.getResources().getColor(R.color.colorWhite));

holder.myView.setBackgroundColor(v.getResources().getColor(R.color.colorGreen));
        } else{
            isClicked = false;
            attendanceTracker.put(studentName,currentAttendance);
            booleanAttendance.put(studentName,(byte)0);

holder.student_name.setTextColor(v.getResources().getColor(R.color.colorBlack));

holder.myView.setBackgroundColor(v.getResources().getColor(R.color.colorWhite));
        }
    }
});
```



Abhishek Sarkar

Vaibhav Asthana

Naina Mittal

Siraist Tithiratanakorn

Rajrishi Sengupta

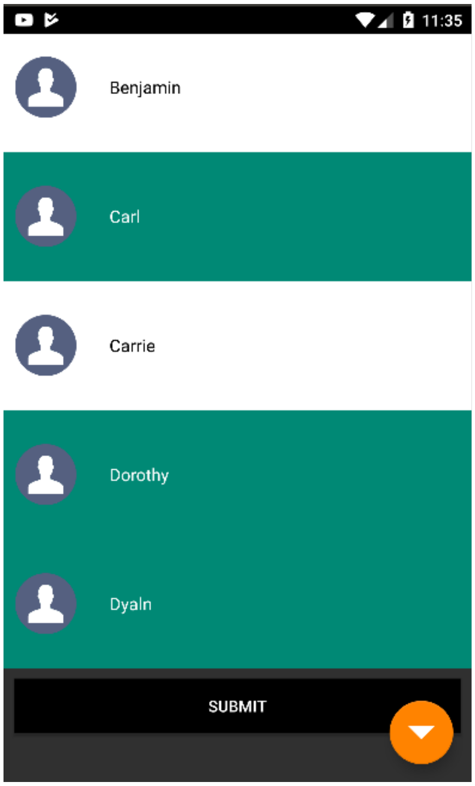
Kapil Bastola

Screenshot: showing list of students to take attendance

10.5.2.2 Updating the attendance sheet.

Upon submitting the attendance data the data will be updated in the same google sheets from where it was fetched. All the students data is updated to google sheets using Google Sheets API v4.

```
void updateAttendance(List<Object> attendanceResults,String column,byte
totalAttendance) {

        try {
           if(getgSheetId() != null && mSheetService!=null) {
              String writeRange = getSheetName()+"!"+column+"1:"+column;
              List<List<Object>> writeData = new ArrayList<>();

              Date today = new Date();
              @SuppressLint("SimpleDateFormat") SimpleDateFormat dateFormat = new
                                 SimpleDateFormat("dd/MM/yy");
              String date = dateFormat.format(today);
              attendanceResults.add(0,date);
              attendanceResults.add(1,String.valueOf(totalAttendance+1));
              attendanceResults.add(2,""); //the third row won't be having any values
              writeData.add(attendanceResults);
              ValueRange vr = new ValueRange()
                                   .setValues(writeData)
                                   .setMajorDimension("COLUMNS");
              try {

//                    AppendValuesResponse result =

                  //updating the data to the attendance sheet directly
//                      these values also can be returned to variable result declared above
                  mSheetService.spreadsheets().values()
                                      .append(gSheetId, writeRange, vr)
                                      .setValueInputOption("RAW")
                                      .execute();

              }catch (Exception e){

                  StringWriter sw = new StringWriter();
                  PrintWriter pw = new PrintWriter(sw);
                  e.printStackTrace(pw);
//                      String sStackTrace = sw.toString(); // stack trace as a string
                  //System.out.println(sStackTrace); //Debugging
              }

           }else{
              showToast("no id found");
           }
        } catch (Exception e) {
           // handle exception
        }
    }
```



Screenshot: attendance taken and ready to submit

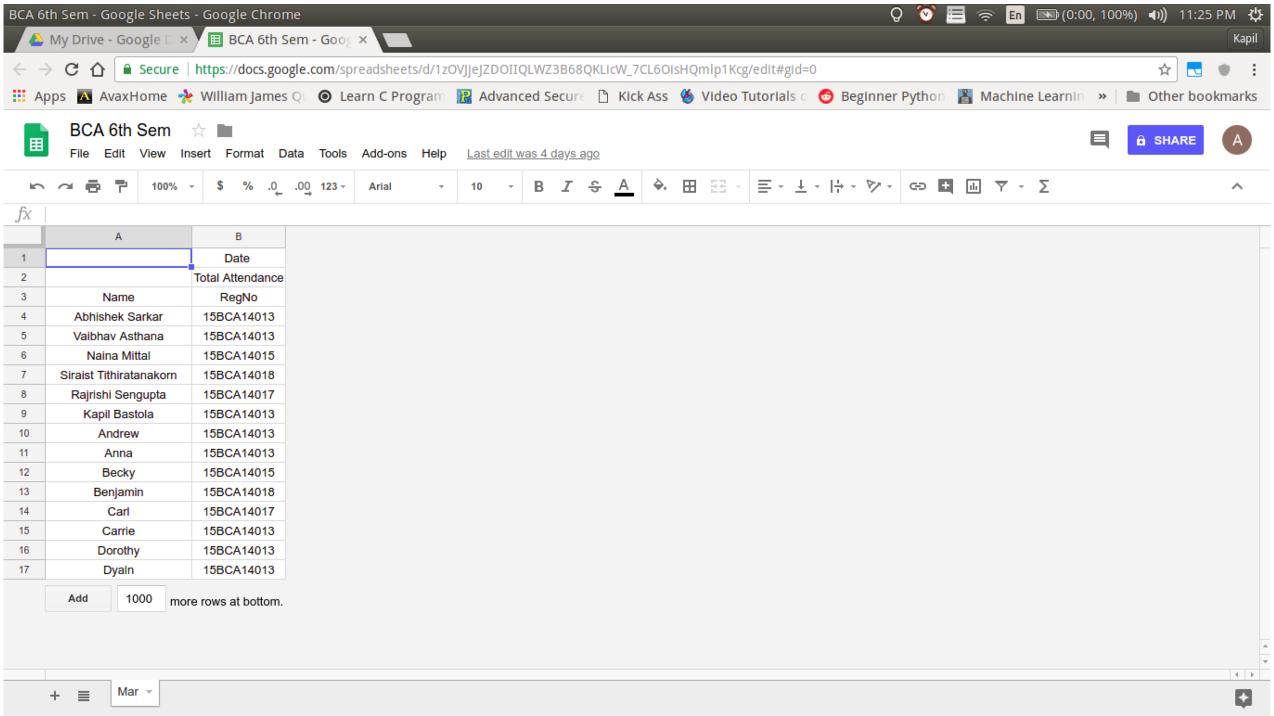
10.5.2.3 Formatting the attendance.

Now as the attendance is taken and updated in the google sheets this module will then format the attendance sheet. For all those students who are absent it will mark their attendance count as red.

```
void formatAttendance() throws IOException{
        byte currRow = 4;
        List<Byte> attendanceBoolean =  studentAttendanceAdapter
                                        .getCurrentAttendanceBoolean();
        //specifying where the filters should be applied

        Color redColor = new Color().setRed((float)0.8).setGreen((float)0.2);
        /*
         * The logic for cell formatting is that it checks the previous attendance
         * value of a student and formats the cell.
         * The syntax for the below formatting can be found at:
         *  URL : https://developers.google.com/sheets/api/samples/conditional-
formatting
         *
         * */

        AddConditionalFormatRuleRequest conditionalFormat;
        ConditionalFormatRule conditionalFormatRule;
        List<Request> requests = new ArrayList<>();
        List<GridRange> ranges;
        String condition;
        for(Byte attendance: attendanceBoolean){

          //algorithm for formatting the absent students data

           if (currentColumnNo > 2) {
               condition = "=EQ(" +nextColumn + String.valueOf(currRow )  + "," +
lastColumn + String.valueOf(currRow) + ")";
           }else{
               condition = "=EQ("+nextColumn + String.valueOf(currRow)+",0)";
               RepeatCellRequest cellRequest = new RepeatCellRequest()
                       .setRange(new GridRange()
                               .setStartColumnIndex((int)currentColumnNo)
                               .setEndRowIndex(0)
                               .setEndRowIndex(1))
                       .setCell(new CellData()
                               .setUserEnteredFormat(new CellFormat()
                                       .setTextFormat(new TextFormat()
                  .setForegroundColor(new Color()
                      .setRed((float)0.067)
                      .setGreen((float)0.423)
                      .setBlue((float)0.117))
                  .setItalic(true)
                  .setBold(true))
              .setHorizontalAlignment("CENTER")))
      .setFields("userEnteredFormat(textFormat,horizontalAlignment)");

  requests.add(new Request().setRepeatCell(cellRequest));
}

if(attendance == 0) {
  conditionalFormat = new AddConditionalFormatRuleRequest();
  conditionalFormatRule = new ConditionalFormatRule();
  ranges = new ArrayList<>();

  GridRange columnRange = new GridRange()
      .setStartRowIndex((int) currRow-1)
      .setEndRowIndex((int) currRow)
      .setStartColumnIndex((int) currentColumnNo)
      .setEndColumnIndex(currentColumnNo + 1);



  CellFormat cellFormat = new CellFormat()
      .setBackgroundColor(redColor);

  ranges.add(columnRange);



  List<ConditionValue> conditionValues = new ArrayList<>();
```

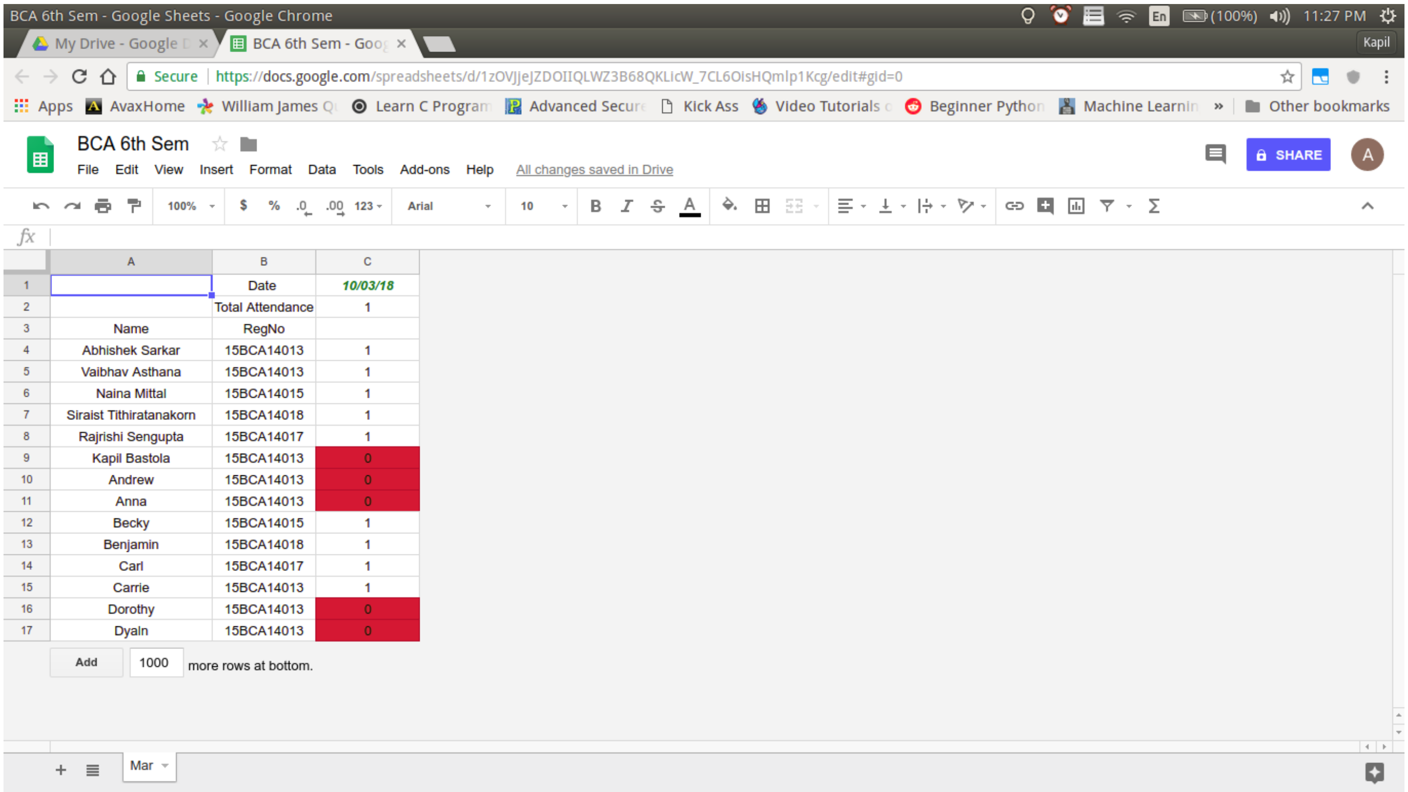
```
                    conditionValues.add(new
ConditionValue().setUserEnteredValue(condition));
                    BooleanRule booleanRule = new BooleanRule()
                            .setCondition(new BooleanCondition()
                                    .setType("CUSTOM_FORMULA")
                                    .setValues(conditionValues))
                            .setFormat(cellFormat);



                    conditionalFormatRule.setRanges(ranges);
                    conditionalFormatRule.setBooleanRule(booleanRule);



                    conditionalFormat.setRule(conditionalFormatRule);



                    requests.add(new
Request().setAddConditionalFormatRule(conditionalFormat));
                    Log.d("condition",condition);
                    Log.d("formatCell", nextColumn + String.valueOf(currRow+1) + ":" +
nextColumn + String.valueOf(currRow+1));
                }
                currRow++;
            }

            AutoResizeDimensionsRequest dimensionsRequest = new
AutoResizeDimensionsRequest()
                    .setDimensions(new DimensionRange()
                    .setDimension("COLUMNS")
                    .setStartIndex((int)currentColumnNo)
                    .setEndIndex(currentColumnNo+1));
```

```
      requests.add(new Request().setAutoResizeDimensions(dimensionsRequest));

      requests.add(new Request().setRepeatCell(new RepeatCellRequest()
          .setRange(new GridRange()
          .setStartColumnIndex((int)currentColumnNo)
          .setEndColumnIndex(currentColumnNo+1))
          .setCell(new CellData()
              .setUserEnteredFormat(new CellFormat()
              .setHorizontalAlignment("CENTER")))
          .setFields("userEnteredFormat(horizontalAlignment)")));

      BatchUpdateSpreadsheetRequest formatReq = new
BatchUpdateSpreadsheetRequest()
          .setRequests(requests);

//        request to format the attendance
//        the request returns a BatchUpdateSpreadsheetResponse object
//        which can be used to check for results
      mSheetService.spreadsheets()
          .batchUpdate(getgSheetId(),formatReq)
          .execute();

    }
```



	A	B
1		Date
2		Total Attendance
3	Name	RegNo
4	Abhishek Sarkar	15BCA14013
5	Vaibhav Asthana	15BCA14013
6	Naina Mittal	15BCA14015
7	Siraist Tithiratanakorn	15BCA14018
8	Rajrishi Sengupta	15BCA14017
9	Kapil Bastola	15BCA14013
10	Andrew	15BCA14013
11	Anna	15BCA14013
12	Becky	15BCA14015
13	Benjamin	15BCA14018
14	Carl	15BCA14017
15	Carrie	15BCA14013
16	Dorothy	15BCA14013
17	Dyaln	15BCA14013

Screenshot: The attendance sheet before taking the attendance.



	A	B	C
1		Date	*10/03/18*
2		Total Attendance	1
3	Name	RegNo	
4	Abhishek Sarkar	15BCA14013	1
5	Vaibhav Asthana	15BCA14013	1
6	Naina Mittal	15BCA14015	1
7	Siraist Tithiratanakorn	15BCA14018	1
8	Rajrishi Sengupta	15BCA14017	1
9	Kapil Bastola	15BCA14013	0
10	Andrew	15BCA14013	0
11	Anna	15BCA14013	0
12	Becky	15BCA14015	1
13	Benjamin	15BCA14018	1
14	Carl	15BCA14017	1
15	Carrie	15BCA14013	1
16	Dorothy	15BCA14013	0
17	Dyaln	15BCA14013	0

Add 1000 more rows at bottom.

Screenshot: The attendance sheet after taking attendance and formatting it.

11. APPENDIX

11.1 CODE SNIPPET

11.1.1 Messaging with Firebase

11.1.1 JAVA CODE SNIPPETS

11.1.1.A Package ‘Authentication’

1. a. RegisterActivity.java

```
package com.project.jain.sikana.Authentication;

import android.app.ProgressDialog;
import android.content.Intent;
import android.support.annotation.NonNull;
import android.support.design.widget.TextInputLayout;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.v7.widget.Toolbar;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.iid.FirebaseInstanceId;

import java.util.HashMap;

import com.project.jain.sikana.MainActivity;
import com.project.jain.sikana.R;

public class RegisterActivity extends AppCompatActivity {

    private TextInputLayout mDisplayName;
    private TextInputLayout mEmail;
    private TextInputLayout mPassword;
    private Button mCreateBtn;


    private Toolbar mToolbar;
    private DatabaseReference mDatabase;

    //ProgressDialog
    private ProgressDialog mRegProgress;

    //Firebase Auth
    private FirebaseAuth mAuth;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_register);

        //Toolbar Set
        mToolbar = (Toolbar) findViewById(R.id.register_toolbar);
        setSupportActionBar(mToolbar);
        getSupportActionBar().setTitle("Create Account");
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);


        mRegProgress = new ProgressDialog(this);



        // Firebase Auth

        mAuth = FirebaseAuth.getInstance();


        // Android Fields

        mDisplayName = (TextInputLayout) findViewById(R.id.register_display_name);
        mEmail = (TextInputLayout) findViewById(R.id.register_email);
        mPassword = (TextInputLayout) findViewById(R.id.reg_password);
        mCreateBtn = (Button) findViewById(R.id.reg_create_btn);


        mCreateBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                String display_name = mDisplayName.getEditText().getText().toString().trim();
                String email = mEmail.getEditText().getText().toString().trim();
                String password = mPassword.getEditText().getText().toString().trim();

                if(!TextUtils.isEmpty(display_name) || !TextUtils.isEmpty(email) ||
!TextUtils.isEmpty(password)){

                    mRegProgress.setTitle("Registering User");
                    mRegProgress.setMessage("Please wait while we create your account !");
                    mRegProgress.setCanceledOnTouchOutside(false);
                    mRegProgress.show();

                    register_user(display_name, email, password);

                }



            }
        });


    }

    private void register_user(final String display_name, String email, String password) {

        mAuth.createUserWithEmailAndPassword(email, password).addOnCompleteListener(new
OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {

                if(task.isSuccessful()){


                    FirebaseUser current_user = FirebaseAuth.getInstance().getCurrentUser();
                    String uid = current_user.getUid();

                    mDatabase =
FirebaseDatabase.getInstance().getReference().child("Users").child(uid);

                    String device_token = FirebaseInstanceId.getInstance().getToken();

                    HashMap<String, String> userMap = new HashMap<>();
                    userMap.put("name", display_name);
                    userMap.put("status", "Hi there I'm using SIKANA.");
                    userMap.put("image", "default");
                    userMap.put("thumb_image", "default");
                    userMap.put("device_token", device_token);

                    mDatabase.setValue(userMap).addOnCompleteListener(new
OnCompleteListener<Void>() {
                        @Override
                        public void onComplete(@NonNull Task<Void> task) {

                            if(task.isSuccessful()){

                                mRegProgress.dismiss();

                                Intent mainIntent = new Intent(RegisterActivity.this, MainActivity.class);
                                mainIntent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK |
Intent.FLAG_ACTIVITY_CLEAR_TASK);
```

```
                                startActivity(mainIntent);
                                finish();

                            }

                        }
                    });


                } else {

                    mRegProgress.hide();
                    Toast.makeText(RegisterActivity.this, "Cannot Sign in. Please check the form and
try again.", Toast.LENGTH_LONG).show();

                }

            }
        });

    }
}
```

1.b. LogInActivity.java

```
package com.project.jain.sikana.Authentication;

import android.app.ProgressDialog;
import android.content.Intent;
import android.support.annotation.NonNull;
import android.support.design.widget.TextInputLayout;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.v7.widget.Toolbar;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.iid.FirebaseInstanceId;

import com.project.jain.sikana.MainActivity;
import com.project.jain.sikana.R;
```

```java
public class LoginActivity extends AppCompatActivity {

    private Toolbar mToolbar;

    private TextInputLayout mLoginEmail;
    private TextInputLayout mLoginPassword;

    private Button mLogin_btn;

    private ProgressDialog mLoginProgress;

    private FirebaseAuth mAuth;

    private DatabaseReference mUserDatabase;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);

        mAuth = FirebaseAuth.getInstance();

        mToolbar = (Toolbar) findViewById(R.id.login_toolbar);
        setSupportActionBar(mToolbar);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        getSupportActionBar().setTitle("Login");


        mLoginProgress = new ProgressDialog(this);

        mUserDatabase = FirebaseDatabase.getInstance().getReference().child("Users");


        mLoginEmail = (TextInputLayout) findViewById(R.id.login_email);
        mLoginPassword = (TextInputLayout) findViewById(R.id.login_password);
        mLogin_btn = (Button) findViewById(R.id.login_btn);

        mLogin_btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                String email = mLoginEmail.getEditText().getText().toString().trim();
                String password = mLoginPassword.getEditText().getText().toString().trim();

                if(!TextUtils.isEmpty(email) || !TextUtils.isEmpty(password)){

                    mLoginProgress.setTitle("Logging In");
                    mLoginProgress.setMessage("Please wait while we check your credentials.");
                    mLoginProgress.setCanceledOnTouchOutside(false);
                    mLoginProgress.show();
                    loginUser(email, password);

                }

            }
        });


    }



    private void loginUser(String email, String password) {


        mAuth.signInWithEmailAndPassword(email, password).addOnCompleteListener(new
OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {

                if(task.isSuccessful()){

                    mLoginProgress.dismiss();

                    String current_user_id = mAuth.getCurrentUser().getUid();
                    String deviceToken = FirebaseInstanceId.getInstance().getToken();


mUserDatabase.child(current_user_id).child("device_token").setValue(deviceToken).addOnSu
ccessListener(new OnSuccessListener<Void>() {
                        @Override
                        public void onSuccess(Void aVoid) {

                            Intent mainIntent = new Intent(LoginActivity.this, MainActivity.class);
                            mainIntent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK |
Intent.FLAG_ACTIVITY_CLEAR_TASK);
                            startActivity(mainIntent);
                            finish();


                        }
                    });

                } else {

                    mLoginProgress.hide();

                    String task_result = task.getException().getMessage().toString();

                    Toast.makeText(LoginActivity.this, "Error : " + task_result,
Toast.LENGTH_LONG).show();

                }

            }
        });


    }
}
```

2. Package ‘Fragments’

2.a. ChatFragment.java

```
package com.project.jain.sikana.Fragments;


import android.content.Context;
import android.content.Intent;
import android.graphics.Typeface;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;

import com.firebase.ui.database.FirebaseRecyclerAdapter;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.ChildEventListener;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.Query;
import com.google.firebase.database.ValueEventListener;
import com.squareup.picasso.Picasso;
import com.project.jain.sikana.Conv;

import de.hdodenhof.circleimageview.CircleImageView;
import com.project.jain.sikana.ChatActivity;

import com.project.jain.sikana.R;
```

```
public class ChatsFragment extends Fragment {

    private RecyclerView mConvList;

    private DatabaseReference mConvDatabase;
    private DatabaseReference mMessageDatabase;
    private DatabaseReference mUsersDatabase;

    private FirebaseAuth mAuth;

    private String mCurrent_user_id;

    private View mMainView;


    public ChatsFragment() {
        // Required empty public constructor
    }


    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {

        mMainView = inflater.inflate(R.layout.fragment_chats, container, false);

        mConvList = (RecyclerView) mMainView.findViewById(R.id.conv_list);
        mAuth = FirebaseAuth.getInstance();

        mCurrent_user_id = mAuth.getCurrentUser().getUid();

        mConvDatabase =
FirebaseDatabase.getInstance().getReference().child("Chat").child(mCurrent_user_id);

        mConvDatabase.keepSynced(true);
        mUsersDatabase = FirebaseDatabase.getInstance().getReference().child("Users");
        mMessageDatabase =
FirebaseDatabase.getInstance().getReference().child("messages").child(mCurrent_user_id);
        mUsersDatabase.keepSynced(true);

        LinearLayoutManager linearLayoutManager = new LinearLayoutManager(getContext());
        linearLayoutManager.setReverseLayout(true);
        linearLayoutManager.setStackFromEnd(true);

        mConvList.setHasFixedSize(true);
        mConvList.setLayoutManager(linearLayoutManager);


        // Inflate the layout for this fragment
        return mMainView;
    }


    @Override
    public void onStart() {
        super.onStart();

        Query conversationQuery = mConvDatabase.orderByChild("timestamp");

        FirebaseRecyclerAdapter<Conv, ConvViewHolder> firebaseConvAdapter = new
FirebaseRecyclerAdapter<Conv, ConvViewHolder>(
                Conv.class,
                R.layout.users_single_layout,
                ConvViewHolder.class,
                conversationQuery
        ) {
            @Override
            protected void populateViewHolder(final ConvViewHolder convViewHolder, final
Conv conv, int i) {



                final String list_user_id = getRef(i).getKey();

                Query lastMessageQuery = mMessageDatabase.child(list_user_id).limitToLast(1);

                lastMessageQuery.addChildEventListener(new ChildEventListener() {
                    @Override
                    public void onChildAdded(DataSnapshot dataSnapshot, String s) {

                        String data = dataSnapshot.child("message").getValue().toString();
                        convViewHolder.setMessage(data, conv.isSeen());

                    }

                    @Override
                    public void onChildChanged(DataSnapshot dataSnapshot, String s) {

                    }

                    @Override
                    public void onChildRemoved(DataSnapshot dataSnapshot) {

                    }

                    @Override
                    public void onChildMoved(DataSnapshot dataSnapshot, String s) {

                    }

                    @Override
                    public void onCancelled(DatabaseError databaseError) {

                    }
                });


                mUsersDatabase.child(list_user_id).addValueEventListener(new
ValueEventListener() {
                    @Override
                    public void onDataChange(DataSnapshot dataSnapshot) {

                        final String userName =
dataSnapshot.child("name").getValue().toString().toUpperCase();
                        String userThumb = dataSnapshot.child("thumb_image").getValue().toString();

                        if(dataSnapshot.hasChild("online")) {

                            String userOnline = dataSnapshot.child("online").getValue().toString();
                            convViewHolder.setUserOnline(userOnline);

                        }

                        convViewHolder.setName(userName);
                        convViewHolder.setUserImage(userThumb, getContext());

                        convViewHolder.mView.setOnClickListener(new View.OnClickListener() {
                            @Override
                            public void onClick(View view) {


                                Intent chatIntent = new Intent(getContext(), ChatActivity.class);
                                chatIntent.putExtra("user_id", list_user_id);
                                chatIntent.putExtra("user_name", userName);
                                startActivity(chatIntent);

                            }
                        });


                    }

                    @Override
                    public void onCancelled(DatabaseError databaseError) {

                    }
                });

            }
        };

        mConvList.setAdapter(firebaseConvAdapter);
```

```
    }

    public static class ConvViewHolder extends RecyclerView.ViewHolder {

        View mView;

        public ConvViewHolder(View itemView) {
            super(itemView);

            mView = itemView;

        }

        public void setMessage(String message, boolean isSeen){

            TextView userStatusView = (TextView) mView.findViewById(R.id.user_single_status);
            userStatusView.setText(message);

            if(!isSeen){
                userStatusView.setTypeface(userStatusView.getTypeface(), Typeface.BOLD);
            } else {
                userStatusView.setTypeface(userStatusView.getTypeface(), Typeface.NORMAL);
            }

        }

        public void setName(String name){

            TextView userNameView = (TextView) mView.findViewById(R.id.user_single_name);
            userNameView.setText(name);

        }

        public void setUserImage(String thumb_image, Context ctx){

            CircleImageView userImageView = (CircleImageView)
mView.findViewById(R.id.user_single_image);

Picasso.with(ctx).load(thumb_image).placeholder(R.drawable.default_avatar).into(userImageVie
w);

        }

        public void setUserOnline(String online_status) {

            ImageView userOnlineView = (ImageView)
mView.findViewById(R.id.user_single_online_icon);

            if(online_status.equals("true")){
                userOnlineView.setVisibility(View.VISIBLE);

            } else {

                userOnlineView.setVisibility(View.INVISIBLE);

            }

        }


    }



}
```

2.b. FriendsFragment.java

```
package com.project.jain.sikana.Fragments;


import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.support.v7.app.AlertDialog;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;

import com.firebase.ui.database.FirebaseRecyclerAdapter;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import com.squareup.picasso.Picasso;
import com.project.jain.sikana.Friends;
import com.project.jain.sikana.ProfileActivity;
import de.hdodenhof.circleimageview.CircleImageView;
import com.project.jain.sikana.ChatActivity;

import com.project.jain.sikana.R;


/**
 * A simple {@link Fragment} subclass.
 */
public class FriendsFragment extends Fragment {

    private RecyclerView mFriendsList;

    private DatabaseReference mFriendsDatabase;
    private DatabaseReference mUsersDatabase;

    private FirebaseAuth mAuth;

    private String mCurrent_user_id;

    private View mMainView;


    public FriendsFragment() {
        // Required empty public constructor
    }


    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {

        mMainView = inflater.inflate(R.layout.fragment_friends, container, false);

        mFriendsList = (RecyclerView) mMainView.findViewById(R.id.friends_list);
        mAuth = FirebaseAuth.getInstance();

        mCurrent_user_id = mAuth.getCurrentUser().getUid();

        mFriendsDatabase =
FirebaseDatabase.getInstance().getReference().child("Friends").child(mCurrent_user_id);
        mFriendsDatabase.keepSynced(true);
        mUsersDatabase = FirebaseDatabase.getInstance().getReference().child("Users");
        mUsersDatabase.keepSynced(true);


        mFriendsList.setHasFixedSize(true);
        mFriendsList.setLayoutManager(new LinearLayoutManager(getContext()));

        // Inflate the layout for this fragment
        return mMainView;
    }


    @Override
    public void onStart() {
        super.onStart();

        FirebaseRecyclerAdapter<Friends, FriendsViewHolder> friendsRecyclerViewAdapter =
new FirebaseRecyclerAdapter<Friends, FriendsViewHolder>(

                Friends.class,
                R.layout.users_single_layout,
                FriendsViewHolder.class,
                mFriendsDatabase


        ) {
            @Override
            protected void populateViewHolder(final FriendsViewHolder friendsViewHolder,
Friends friends, int i) {

                friendsViewHolder.setDate(friends.getDate());

                final String list_user_id = getRef(i).getKey();

                mUsersDatabase.child(list_user_id).addValueEventListener(new
ValueEventListener() {
                    @Override
                    public void onDataChange(DataSnapshot dataSnapshot) {

                        final String userName =
dataSnapshot.child("name").getValue().toString().toUpperCase();
                        String userThumb = dataSnapshot.child("thumb_image").getValue().toString();

                        if(dataSnapshot.hasChild("online")) {

                            String userOnline = dataSnapshot.child("online").getValue().toString();
                            friendsViewHolder.setUserOnline(userOnline);

                        }

                        friendsViewHolder.setName(userName);
                        friendsViewHolder.setUserImage(userThumb, getContext());

                        friendsViewHolder.mView.setOnClickListener(new View.OnClickListener() {
                            @Override
                            public void onClick(View view) {

                                CharSequence options[] = new CharSequence[]{"Open Profile", "Send
message"};
```

```
                            final AlertDialog.Builder builder = new AlertDialog.Builder(getContext());

                            builder.setTitle("Select Options");
                            builder.setItems(options, new DialogInterface.OnClickListener() {
                                @Override
                                public void onClick(DialogInterface dialogInterface, int i) {

                                    //Click Event for each item.
                                    if(i == 0){

                                        Intent profileIntent = new Intent(getContext(), ProfileActivity.class);
                                        profileIntent.putExtra("user_id", list_user_id);
                                        startActivity(profileIntent);

                                    }

                                    if(i == 1){

                                        Intent chatIntent = new Intent(getContext(), ChatActivity.class);
                                        chatIntent.putExtra("user_id", list_user_id);
                                        chatIntent.putExtra("user_name", userName);
                                        startActivity(chatIntent);

                                    }

                                }
                            });

                            builder.show();

                        }
                    });


                }

                @Override
                public void onCancelled(DatabaseError databaseError) {

                }
            });

        }
    };

    mFriendsList.setAdapter(friendsRecyclerViewAdapter);


}
```

```
    public static class FriendsViewHolder extends RecyclerView.ViewHolder {

        View mView;

        public FriendsViewHolder(View itemView) {
            super(itemView);

            mView = itemView;

        }

        public void setDate(String date){

            TextView userStatusView = (TextView) mView.findViewById(R.id.user_single_status);
            userStatusView.setText(date);

        }

        public void setName(String name){

            TextView userNameView = (TextView) mView.findViewById(R.id.user_single_name);
            userNameView.setText(name);

        }

        public void setUserImage(String thumb_image, Context ctx){

            CircleImageView userImageView = (CircleImageView)
mView.findViewById(R.id.user_single_image);

Picasso.with(ctx).load(thumb_image).placeholder(R.drawable.default_avatar).into(userImageVie
w);

        }

        public void setUserOnline(String online_status) {

            ImageView userOnlineView = (ImageView)
mView.findViewById(R.id.user_single_online_icon);

            if(online_status.equals("true")){

                userOnlineView.setVisibility(View.VISIBLE);

            } else {

                userOnlineView.setVisibility(View.INVISIBLE);

            }

        }
    }
}
```

3.Package ‘SIKANA’

3.a. ChatActivity.java

```
package com.project.jain.sikana;

import android.content.Context;
import android.content.Intent;
import android.net.Uri;
import android.media.MediaPlayer;
import android.support.annotation.NonNull;
import android.support.v4.widget.SwipeRefreshLayout;
import android.support.v7.app.ActionBar;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.support.v7.widget.Toolbar;
import android.text.TextUtils;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.TextView;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.ChildEventListener;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.Query;
import com.google.firebase.database.ServerValue;
import com.google.firebase.database.ValueEventListener;
import com.google.firebase.storage.FirebaseStorage;
import com.google.firebase.storage.StorageReference;
import com.google.firebase.storage.UploadTask;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

import de.hdodenhof.circleimageview.CircleImageView;
import com.project.jain.sikana.R;

public class ChatActivity extends AppCompatActivity {

    private String mChatUser;
    private Toolbar mChatToolbar;

    private DatabaseReference mRootRef;

    private TextView mTitleView;
    private TextView mLastSeenView;
    private CircleImageView mProfileImage;
    private FirebaseAuth mAuth;
    private String mCurrentUserId;

    private ImageButton mChatAddBtn;
    private ImageButton mChatSendBtn;
    private EditText mChatMessageView;

    private RecyclerView mMessagesList;
    private SwipeRefreshLayout mRefreshLayout;

    private final List<Messages> messagesList = new ArrayList<>();
    private LinearLayoutManager mLinearLayout;
    private MessageAdapter mAdapter;

    private static final int TOTAL_ITEMS_TO_LOAD = 10;
    private int mCurrentPage = 1;

    private static final int GALLERY_PICK = 1;

    // Storage Firebase
    private StorageReference mImageStorage;

    private MediaPlayer mediaPlayer;


    //New Solution
    private int itemPos = 0;

    private String mLastKey = "";
    private String mPrevKey = "";


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_chat);

        mChatToolbar = (Toolbar) findViewById(R.id.chat_app_bar);
        setSupportActionBar(mChatToolbar);
        ActionBar actionBar = getSupportActionBar();

        actionBar.setDisplayHomeAsUpEnabled(true);
        actionBar.setDisplayShowCustomEnabled(true);

        mRootRef = FirebaseDatabase.getInstance().getReference();
        mAuth = FirebaseAuth.getInstance();
        mCurrentUserId = mAuth.getCurrentUser().getUid();

        mChatUser = getIntent().getStringExtra("user_id");
        String userName = getIntent().getStringExtra("user_name");

        LayoutInflater inflater = (LayoutInflater)
    this.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        View action_bar_view = inflater.inflate(R.layout.chat_custom_bar, null);

        actionBar.setCustomView(action_bar_view);

        mediaPlayer = MediaPlayer.create(this,R.raw.inquisitiveness);

        // ---- Custom Action bar Items ----

        mTitleView = (TextView) findViewById(R.id.custom_bar_title);
        mLastSeenView = (TextView) findViewById(R.id.custom_bar_seen);
      mProfileImage = (CircleImageView) findViewById(R.id.custom_bar_image);

        mChatAddBtn = (ImageButton) findViewById(R.id.chat_add_btn);
        mChatSendBtn = (ImageButton) findViewById(R.id.chat_send_btn);
        mChatMessageView = (EditText) findViewById(R.id.chat_message_view);

        mAdapter = new MessageAdapter(messagesList);

        mMessagesList = (RecyclerView) findViewById(R.id.messages_list);
        mRefreshLayout = (SwipeRefreshLayout) findViewById(R.id.message_swipe_layout);
        mLinearLayout = new LinearLayoutManager(this);

        mMessagesList.setHasFixedSize(true);
        mMessagesList.setLayoutManager(mLinearLayout);

        mMessagesList.setAdapter(mAdapter);

        //------- IMAGE STORAGE ---------
        mImageStorage = FirebaseStorage.getInstance().getReference();


    mRootRef.child("Chat").child(mCurrentUserId).child(mChatUser).child("seen").setValue(t
    rue);

        loadMessages();
        mTitleView.setText(userName);

        mRootRef.child("Users").child(mChatUser).addValueEventListener(new
ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {

                String online = dataSnapshot.child("online").getValue().toString();
                String image = dataSnapshot.child("image").getValue().toString();

                if(online.equals("true")) {

                    mLastSeenView.setText("Online");

                } else {

                    GetTimeAgo getTimeAgo = new GetTimeAgo();

                    long lastTime = Long.parseLong(online);

                    String lastSeenTime = getTimeAgo.getTimeAgo(lastTime, getApplicationContext());

                    mLastSeenView.setText(lastSeenTime);

                }

            }

            @Override
            public void onCancelled(DatabaseError databaseError) {

            }
        });


        mRootRef.child("Chat").child(mCurrentUserId).addValueEventListener(new
ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {

                if(!dataSnapshot.hasChild(mChatUser)){

                    Map chatAddMap = new HashMap();
                    chatAddMap.put("seen", false);
                    chatAddMap.put("timestamp", ServerValue.TIMESTAMP);

                    Map chatUserMap = new HashMap();
                    chatUserMap.put("Chat/" + mCurrentUserId + "/" + mChatUser, chatAddMap);
                    chatUserMap.put("Chat/" + mChatUser + "/" + mCurrentUserId, chatAddMap);
```

```
          mRootRef.updateChildren(chatUserMap, new
DatabaseReference.CompletionListener() {
            @Override
            public void onComplete(DatabaseError databaseError, DatabaseReference
databaseReference) {

              if(databaseError != null){

                Log.d("CHAT_LOG", databaseError.getMessage().toString());

              }

            }
          });

        }

      }

      @Override
      public void onCancelled(DatabaseError databaseError) {

      }
    });



    mChatSendBtn.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View view) {

        sendMessage();

      }
    });



    mChatAddBtn.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View view) {

        Intent galleryIntent = new Intent();
        galleryIntent.setType("image/*");
        galleryIntent.setAction(Intent.ACTION_GET_CONTENT);

        startActivityForResult(Intent.createChooser(galleryIntent, "SELECT IMAGE"),
GALLERY_PICK);

      }
    });
```

```
        mRefreshLayout.setOnRefreshListener(new SwipeRefreshLayout.OnRefreshListener() {
            @Override
            public void onRefresh() {

                mCurrentPage++;

                itemPos = 0;

                loadMoreMessages();


            }
        });


    }


    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);

        if(requestCode == GALLERY_PICK && resultCode == RESULT_OK){

            Uri imageUri = data.getData();

            final String current_user_ref = "messages/" + mCurrentUserId + "/" + mChatUser;
            final String chat_user_ref = "messages/" + mChatUser + "/" + mCurrentUserId;

            DatabaseReference user_message_push = mRootRef.child("messages")
                    .child(mCurrentUserId).child(mChatUser).push();

            final String push_id = user_message_push.getKey();


            StorageReference filepath = mImageStorage.child("message_images").child( push_id +
".jpg");

            filepath.putFile(imageUri).addOnCompleteListener(new
OnCompleteListener<UploadTask.TaskSnapshot>() {
                @Override
                public void onComplete(@NonNull Task<UploadTask.TaskSnapshot> task) {

                    if(task.isSuccessful()){

                        String download_url = task.getResult().getDownloadUrl().toString();
                        Map messageMap = new HashMap();
                        messageMap.put("message", download_url);
                        messageMap.put("seen", false);
                        messageMap.put("type", "image");
                        messageMap.put("time", ServerValue.TIMESTAMP);
                        messageMap.put("from", mCurrentUserId);

                        Map messageUserMap = new HashMap();
                        messageUserMap.put(current_user_ref + "/" + push_id, messageMap);
                        messageUserMap.put(chat_user_ref + "/" + push_id, messageMap);

                        mChatMessageView.setText("");

                        mRootRef.updateChildren(messageUserMap, new
DatabaseReference.CompletionListener() {
                            @Override
                            public void onComplete(DatabaseError databaseError, DatabaseReference
databaseReference) {

                                if(databaseError != null){

                                    Log.d("CHAT_LOG", databaseError.getMessage().toString());

                                }

                            }
                        });


                    }

                }
            });

        }

    }

    private void loadMoreMessages() {

        DatabaseReference messageRef =
mRootRef.child("messages").child(mCurrentUserId).child(mChatUser);

        Query messageQuery = messageRef.orderByKey().endAt(mLastKey).limitToLast(10);

        messageQuery.addChildEventListener(new ChildEventListener() {
            @Override
            public void onChildAdded(DataSnapshot dataSnapshot, String s) {


                Messages message = dataSnapshot.getValue(Messages.class);
                String messageKey = dataSnapshot.getKey();

                if(!mPrevKey.equals(messageKey)){

                    messagesList.add(itemPos++, message);

                } else {

                    mPrevKey = mLastKey;

                }


                if(itemPos == 1) {

                    mLastKey = messageKey;

                }


                Log.d("TOTALKEYS", "Last Key : " + mLastKey + " | Prev Key : " + mPrevKey
+ " | Message Key : " + messageKey);

                mAdapter.notifyDataSetChanged();

                mRefreshLayout.setRefreshing(false);

                mLinearLayout.scrollToPositionWithOffset(10, 0);

            }

            @Override
            public void onChildChanged(DataSnapshot dataSnapshot, String s) {

            }

            @Override
            public void onChildRemoved(DataSnapshot dataSnapshot) {

            }

            @Override
            public void onChildMoved(DataSnapshot dataSnapshot, String s) {

            }

            @Override
            public void onCancelled(DatabaseError databaseError) {

            }
        });
```

```
    }

    private void loadMessages() {

        DatabaseReference messageRef =
mRootRef.child("messages").child(mCurrentUserId).child(mChatUser);

        Query messageQuery = messageRef.limitToLast(mCurrentPage *
TOTAL_ITEMS_TO_LOAD);


        messageQuery.addChildEventListener(new ChildEventListener() {
            @Override
            public void onChildAdded(DataSnapshot dataSnapshot, String s) {

                Messages message = dataSnapshot.getValue(Messages.class);

                itemPos++;

                if(itemPos == 1){

                    String messageKey = dataSnapshot.getKey();

                    mLastKey = messageKey;
                    mPrevKey = messageKey;

                }

                messagesList.add(message);
                mAdapter.notifyDataSetChanged();

                mMessagesList.scrollToPosition(messagesList.size() - 1);

                mRefreshLayout.setRefreshing(false);

            }

            @Override
            public void onChildChanged(DataSnapshot dataSnapshot, String s) {

            }

            @Override
            public void onChildRemoved(DataSnapshot dataSnapshot) {

            }

            @Override
            public void onChildMoved(DataSnapshot dataSnapshot, String s) {
            }

            @Override
            public void onCancelled(DatabaseError databaseError) {

            }
        });

    }

    private void sendMessage() {


        String message = mChatMessageView.getText().toString();

        if(!TextUtils.isEmpty(message)){

            String current_user_ref = "messages/" + mCurrentUserId + "/" + mChatUser;
            String chat_user_ref = "messages/" + mChatUser + "/" + mCurrentUserId;

            DatabaseReference user_message_push = mRootRef.child("messages")
                    .child(mCurrentUserId).child(mChatUser).push();

            String push_id = user_message_push.getKey();

            Map messageMap = new HashMap();
            messageMap.put("message", message);
            messageMap.put("seen", false);
            messageMap.put("type", "text");
            messageMap.put("time", ServerValue.TIMESTAMP);
            messageMap.put("from", mCurrentUserId);

            Map messageUserMap = new HashMap();
            messageUserMap.put(current_user_ref + "/" + push_id, messageMap);
            messageUserMap.put(chat_user_ref + "/" + push_id, messageMap);

            mChatMessageView.setText("");
            mediaPlayer.start();



mRootRef.child("Chat").child(mCurrentUserId).child(mChatUser).child("seen").setValue(true
);

mRootRef.child("Chat").child(mCurrentUserId).child(mChatUser).child("timestamp").setValu
e(ServerValue.TIMESTAMP);


mRootRef.child("Chat").child(mChatUser).child(mCurrentUserId).child("seen").setValue(false
);
```

```
mRootRef.child("Chat").child(mChatUser).child(mCurrentUserId).child("timestamp").setValu
e(ServerValue.TIMESTAMP);

            mRootRef.updateChildren(messageUserMap, new
DatabaseReference.CompletionListener() {
                @Override
                public void onComplete(DatabaseError databaseError, DatabaseReference
databaseReference) {

                    if(databaseError != null){

                        Log.d("CHAT_LOG", databaseError.getMessage().toString());

                    }

                }
            });

        }

    }
}
```

3.b. Conv.java

```
package com.project.jain.sikana;


public class Conv {

    public boolean seen;
    public long timestamp;

    public Conv(){

    }

    public boolean isSeen() {
        return seen;
    }

    public void setSeen(boolean seen) {
        this.seen = seen;
    }

    public long getTimestamp() {
        return timestamp;
    }
    public void setTimestamp(long timestamp) {
        this.timestamp = timestamp;
    }

    public Conv(boolean seen, long timestamp) {
        this.seen = seen;
        this.timestamp = timestamp;
    }
}
```

3.c. EndlessRecyclerOnScrollListener .java

```
package com.project.jain.sikana;

import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;



public abstract class EndlessRecyclerOnScrollListener extends RecyclerView.OnScrollListener
{
    public static String TAG = EndlessRecyclerOnScrollListener.class.getSimpleName();

    private int previousTotal = 0; // The total number of items in the dataset after the last load
    private boolean loading = true; // True if we are still waiting for the last set of data to load.
    private int visibleThreshold = 5; // The minimum amount of items to have below your current
scroll position before loading more.
    int firstVisibleItem, visibleItemCount, totalItemCount;

    private int current_page = 1;

    private LinearLayoutManager mLinearLayoutManager;

    public EndlessRecyclerOnScrollListener(LinearLayoutManager linearLayoutManager) {
        this.mLinearLayoutManager = linearLayoutManager;
    }

    @Override
    public void onScrolled(RecyclerView recyclerView, int dx, int dy) {
        super.onScrolled(recyclerView, dx, dy);

        visibleItemCount = recyclerView.getChildCount();
        totalItemCount = mLinearLayoutManager.getItemCount();
        firstVisibleItem = mLinearLayoutManager.findFirstVisibleItemPosition();

        if (loading) {
            if (totalItemCount > previousTotal) {
                loading = false;
                previousTotal = totalItemCount;
            }
        }
        if (!loading && (totalItemCount - visibleItemCount)
                <= (firstVisibleItem + visibleThreshold)) {
            // End has been reached

            // Do something
            current_page++;

            onLoadMore(current_page);

            loading = true;
        }
    }

    public abstract void onLoadMore(int current_page);
}
```

3.d. FirebaseMessagingService.java

```
package com.project.jain.sikana;

import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Intent;
import android.support.v4.app.NotificationCompat;

import com.google.firebase.messaging.RemoteMessage;

import com.project.jain.sikana.R;


public class FirebaseMessagingService extends
com.google.firebase.messaging.FirebaseMessagingService {


    @Override
    public void onMessageReceived(RemoteMessage remoteMessage) {
        super.onMessageReceived(remoteMessage);

        String notification_title = remoteMessage.getNotification().getTitle();
        String notification_message = remoteMessage.getNotification().getBody();

        String click_action = remoteMessage.getNotification().getClickAction();

        String from_user_id = remoteMessage.getData().get("from_user_id");

        NotificationCompat.Builder mBuilder =
                new NotificationCompat.Builder(this)
```

```
            .setSmallIcon(R.mipmap.ic_launcher)
            .setContentTitle(notification_title)
            .setContentText(notification_message);


    Intent resultIntent = new Intent(click_action);
    resultIntent.putExtra("user_id", from_user_id);


    PendingIntent resultPendingIntent =
        PendingIntent.getActivity(
            this,
            0,
            resultIntent,
            PendingIntent.FLAG_UPDATE_CURRENT
        );

    mBuilder.setContentIntent(resultPendingIntent);




    int mNotificationId = (int) System.currentTimeMillis();

    NotificationManager mNotifyMgr =
        (NotificationManager) getSystemService(NOTIFICATION_SERVICE);

    mNotifyMgr.notify(mNotificationId, mBuilder.build());


  }
}
```

3.e. Friends.java

```
package com.project.jain.sikana;


public class Friends {

  public String date;
```

```
    public Friends(){

    }

    public Friends(String date) {
        this.date = date;
    }

    public String getDate() {
        return date;
    }

    public void setDate(String date) {
        this.date = date;
    }
}
```

3.f. GetTimeAgo.java

```
package com.project.jain.sikana;

import android.app.Application;
import android.content.Context;


public class GetTimeAgo extends Application{


    private static final int SECOND_MILLIS = 1000;
    private static final int MINUTE_MILLIS = 60 * SECOND_MILLIS;
    private static final int HOUR_MILLIS = 60 * MINUTE_MILLIS;
    private static final int DAY_MILLIS = 24 * HOUR_MILLIS;


    public static String getTimeAgo(long time, Context ctx) {
        if (time < 1000000000000L) {
            // if timestamp given in seconds, convert to millis
            time *= 1000;
        }

        long now = System.currentTimeMillis();
        if (time > now || time <= 0) {
            return null;
        }

        // TODO: localize
        final long diff = now - time;
        if (diff < MINUTE_MILLIS) {
            return "just now";
        } else if (diff < 2 * MINUTE_MILLIS) {
            return "a minute ago";
        } else if (diff < 50 * MINUTE_MILLIS) {
            return diff / MINUTE_MILLIS + " minutes ago";
        } else if (diff < 90 * MINUTE_MILLIS) {
            return "an hour ago";
        } else if (diff < 24 * HOUR_MILLIS) {
            return diff / HOUR_MILLIS + " hours ago";
        } else if (diff < 48 * HOUR_MILLIS) {
            return "yesterday";
        } else {
            return diff / DAY_MILLIS + " days ago";
        }
    }

}
```

3.g. SIKANA.java

```
package com.project.jain.sikana;

import android.app.Application;

import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ServerValue;
import com.google.firebase.database.ValueEventListener;
import com.squareup.picasso.OkHttpDownloader;
import com.squareup.picasso.Picasso;

public class SIKANA extends Application{

    private DatabaseReference mUserDatabase;
    private FirebaseAuth mAuth;

    @Override
    public void onCreate() {
        super.onCreate();

        FirebaseDatabase.getInstance().setPersistenceEnabled(true);

        /* Picasso */

        Picasso.Builder builder = new Picasso.Builder(this);
        builder.downloader(new OkHttpDownloader(this, Integer.MAX_VALUE));
        Picasso built = builder.build();
        built.setIndicatorsEnabled(true);
        built.setLoggingEnabled(true);
        Picasso.setSingletonInstance(built);

        mAuth = FirebaseAuth.getInstance();

        if(mAuth.getCurrentUser() != null) {

            mUserDatabase = FirebaseDatabase.getInstance()
                    .getReference().child("Users").child(mAuth.getCurrentUser().getUid());

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

                    if (dataSnapshot != null) {


mUserDatabase.child("online").onDisconnect().setValue(ServerValue.TIMESTAMP);

                    }

                }

                @Override
                public void onCancelled(DatabaseError databaseError) {

                }
            });

        }


    }


}
```

3.h. MainActivity.java

```
package com.project.jain.sikana;

import android.content.Intent;
import android.support.design.widget.TabLayout;
import android.support.v4.view.ViewPager;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.v7.widget.Toolbar;
import android.view.Menu;
import android.view.MenuItem;

import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ServerValue;

import com.project.jain.sikana.R;

public class MainActivity extends AppCompatActivity {

  private FirebaseAuth mAuth;
  private Toolbar mToolbar;

  private ViewPager mViewPager;
  private SectionsPagerAdapter mSectionsPagerAdapter;

  private DatabaseReference mUserRef;

  private TabLayout mTabLayout;

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    mAuth = FirebaseAuth.getInstance();

    mToolbar = (Toolbar) findViewById(R.id.main_page_toolbar);
    setSupportActionBar(mToolbar);
    getSupportActionBar().setTitle("SIKANA");

    if (mAuth.getCurrentUser() != null) {


      mUserRef =
FirebaseDatabase.getInstance().getReference().child("Users").child(mAuth.getCurrentUser().ge
tUid());

    }


    //Tabs
    mViewPager = (ViewPager) findViewById(R.id.main_tabPager);
    mSectionsPagerAdapter = new SectionsPagerAdapter(getSupportFragmentManager());

    mViewPager.setAdapter(mSectionsPagerAdapter);

    mTabLayout = (TabLayout) findViewById(R.id.main_tabs);
```

```
        mTabLayout.setupWithViewPager(mViewPager);




    }

    @Override
    public void onStart() {
        super.onStart();
        // Check if user is signed in (non-null) and update UI accordingly.
        FirebaseUser currentUser = mAuth.getCurrentUser();

        if(currentUser == null){

            sendToStart();

        } else {

            mUserRef.child("online").setValue("true");

        }

    }


    @Override
    protected void onStop() {
        super.onStop();

        FirebaseUser currentUser = mAuth.getCurrentUser();

        if(currentUser != null) {

            mUserRef.child("online").setValue(ServerValue.TIMESTAMP);

        }

    }

    private void sendToStart() {

        Intent startIntent = new Intent(MainActivity.this, StartActivity.class);
        startActivity(startIntent);
        finish();

    }


    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        super.onCreateOptionsMenu(menu);

        getMenuInflater().inflate(R.menu.main_menu, menu);


        return true;
    }


    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        super.onOptionsItemSelected(item);

        if(item.getItemId() == R.id.main_logout_btn){

            mUserRef.child("online").setValue(ServerValue.TIMESTAMP);

            FirebaseAuth.getInstance().signOut();
            sendToStart();

        }

        if(item.getItemId() == R.id.main_settings_btn){

            Intent settingsIntent = new Intent(MainActivity.this, SettingsActivity.class);
            startActivity(settingsIntent);

        }

        if(item.getItemId() == R.id.main_all_btn){

            Intent settingsIntent = new Intent(MainActivity.this, UsersActivity.class);
            startActivity(settingsIntent);

        }

        return true;
    }
}
```

3.i. MessageAdapter.java

```
package com.project.jain.sikana;

import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import com.squareup.picasso.Picasso;

import java.util.List;

import de.hdodenhof.circleimageview.CircleImageView;
import com.project.jain.sikana.R;

public class MessageAdapter extends
RecyclerView.Adapter<MessageAdapter.MessageViewHolder>{


    private List<Messages> mMessageList;
    private DatabaseReference mUserDatabase;

    public MessageAdapter(List<Messages> mMessageList) {

        this.mMessageList = mMessageList;

    }

    @Override
    public MessageViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {

        View v = LayoutInflater.from(parent.getContext())
                .inflate(R.layout.message_single_layout ,parent, false);

        return new MessageViewHolder(v);

    }

    public class MessageViewHolder extends RecyclerView.ViewHolder {

        public TextView messageText;
        public CircleImageView profileImage;
        public TextView displayName;
        public ImageView messageImage;

        public MessageViewHolder(View view) {
            super(view);

            messageText = (TextView) view.findViewById(R.id.message_text_layout);
            profileImage = (CircleImageView) view.findViewById(R.id.message_profile_layout);
            displayName = (TextView) view.findViewById(R.id.name_text_layout);
            messageImage = (ImageView) view.findViewById(R.id.message_image_layout);
```

```
      }
    }

    @Override
    public void onBindViewHolder(final MessageViewHolder viewHolder, int i) {

      Messages c = mMessageList.get(i);

      String from_user = c.getFrom();
      String message_type = c.getType();


      mUserDatabase =
  FirebaseDatabase.getInstance().getReference().child("Users").child(from_user);

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

          String name = dataSnapshot.child("name").getValue().toString().toUpperCase();
          String image = dataSnapshot.child("thumb_image").getValue().toString();

          viewHolder.displayName.setText(name);

          Picasso.with(viewHolder.profileImage.getContext()).load(image)
              .placeholder(R.drawable.default_avatar).into(viewHolder.profileImage);

        }

        @Override
        public void onCancelled(DatabaseError databaseError) {

        }
      });

      if(message_type.equals("text")) {

        viewHolder.messageText.setText(c.getMessage());
        viewHolder.messageImage.setVisibility(View.INVISIBLE);


      } else {

        viewHolder.messageText.setVisibility(View.INVISIBLE);
        Picasso.with(viewHolder.profileImage.getContext()).load(c.getMessage())
            .placeholder(R.drawable.default_avatar).into(viewHolder.messageImage);

      }

    }
    @Override
    public int getItemCount() {
        return mMessageList.size();
    }

}
```

3.j. Messages.java

```
package com.project.jain.sikana;

public class Messages {

    private String message, type;
    private long  time;
    private boolean seen;

    private String from;

    public Messages(String from) {
        this.from = from;
    }

    public String getFrom() {
        return from;
    }

    public void setFrom(String from) {
        this.from = from;
    }

    public Messages(String message, String type, long time, boolean seen) {
        this.message = message;
        this.type = type;
        this.time = time;
        this.seen = seen;
    }

    public String getMessage() {
        return message;
    }

    public void setMessage(String message) {
        this.message = message;
    }

    public String getType() {
        return type;
    }

    public void setType(String type) {
        this.type = type;
    }

    public long getTime() {
        return time;
    }

    public void setTime(long time) {
        this.time = time;
    }
    public boolean isSeen() {
        return seen;
    }

    public void setSeen(boolean seen) {
        this.seen = seen;
    }

    public Messages(){

    }

}
```

3.k. ProfileActivity.java

```
package com.project.jain.sikana;


public class Messages {

    private String message, type;
    private long  time;
    private boolean seen;

    private String from;

    public Messages(String from) {
        this.from = from;
    }

    public String getFrom() {
        return from;
    }

    public void setFrom(String from) {
        this.from = from;
    }

    public Messages(String message, String type, long time, boolean seen) {
        this.message = message;
        this.type = type;
        this.time = time;
        this.seen = seen;
    }

    public String getMessage() {
        return message;
    }

    public void setMessage(String message) {
        this.message = message;
    }
```

```java
    public String getType() {
        return type;
    }

    public void setType(String type) {
        this.type = type;
    }

    public long getTime() {
        return time;
    }

    public void setTime(long time) {
        this.time = time;
    }

    public boolean isSeen() {
        return seen;
    }

    public void setSeen(boolean seen) {
        this.seen = seen;
    }

    public Messages(){

    }

}
```

3.l. SecondPagerAdapter.java

```java
package com.project.jain.sikana;

import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentManager;
import android.support.v4.app.FragmentPagerAdapter;

import com.project.jain.sikana.Fragments.ChatsFragment;
import com.project.jain.sikana.Fragments.FriendsFragment;
```

```
class SectionsPagerAdapter extends FragmentPagerAdapter{


    public SectionsPagerAdapter(FragmentManager fm) {
        super(fm);
    }

    @Override
    public Fragment getItem(int position) {

        switch(position) {

            case 0:
                ChatsFragment chatsFragment = new ChatsFragment();
                return  chatsFragment;

            case 1:
                FriendsFragment friendsFragment = new FriendsFragment();
                return friendsFragment;

            default:
                return  null;
        }

    }

    @Override
    public int getCount() {
        return 2;
    }

    public CharSequence getPageTitle(int position){

        switch (position) {

            case 0:
                return "CHATS";

            case 1:
                return "FRIENDS";

            default:
                return null;
        }

    }

}
```

3.m. SettingActivity.java

```
package com.project.jain.sikana;

import android.app.ProgressDialog;
import android.content.Intent;
import android.graphics.Bitmap;
import android.net.Uri;
import android.support.annotation.NonNull;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import com.google.firebase.storage.FirebaseStorage;
import com.google.firebase.storage.StorageReference;
import com.google.firebase.storage.UploadTask;
import com.squareup.picasso.Callback;
import com.squareup.picasso.NetworkPolicy;
import com.squareup.picasso.Picasso;
import com.theartofdev.edmodo.cropper.CropImage;

import java.io.ByteArrayOutputStream;
import java.io.File;
import java.util.HashMap;
import java.util.Map;
import java.util.Random;

import de.hdodenhof.circleimageview.CircleImageView;
import id.zelory.compressor.Compressor;
import com.project.jain.sikana.R;

public class SettingsActivity extends AppCompatActivity {

    private DatabaseReference mUserDatabase;
    private FirebaseUser mCurrentUser;


    //Android Layout
    private CircleImageView mDisplayImage;
    private TextView mName;
    private TextView mStatus;

    private Button mStatusBtn;
    private Button mImageBtn;


    private static final int GALLERY_PICK = 1;

    // Storage Firebase
    private StorageReference mImageStorage;

    private ProgressDialog mProgressDialog;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_settings);

        mDisplayImage = (CircleImageView) findViewById(R.id.settings_image);
        mName = (TextView) findViewById(R.id.settings_name);
        mStatus = (TextView) findViewById(R.id.settings_status);

        mStatusBtn = (Button) findViewById(R.id.settings_status_btn);
        mImageBtn = (Button) findViewById(R.id.settings_image_btn);

        mImageStorage = FirebaseStorage.getInstance().getReference();

        mCurrentUser = FirebaseAuth.getInstance().getCurrentUser();

        String current_uid = mCurrentUser.getUid();



        mUserDatabase =
FirebaseDatabase.getInstance().getReference().child("Users").child(current_uid);
        mUserDatabase.keepSynced(true);

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

                String name = dataSnapshot.child("name").getValue().toString().toUpperCase();
                final String image = dataSnapshot.child("image").getValue().toString();
                String status = dataSnapshot.child("status").getValue().toString();
                String thumb_image = dataSnapshot.child("thumb_image").getValue().toString();

                mName.setText(name);
        mStatus.setText(status);

        if(!image.equals("default")) {


//Picasso.with(SettingsActivity.this).load(image).placeholder(R.drawable.default_avatar).into(m
DisplayImage);


Picasso.with(SettingsActivity.this).load(image).networkPolicy(NetworkPolicy.OFFLINE)
              .placeholder(R.drawable.default_avatar).into(mDisplayImage, new Callback()
{
            @Override
            public void onSuccess() {

            }

            @Override
            public void onError() {


Picasso.with(SettingsActivity.this).load(image).placeholder(R.drawable.default_avatar).into(mD
isplayImage);

            }
          });

        }


      }

      @Override
      public void onCancelled(DatabaseError databaseError) {

      }
    });


    mStatusBtn.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View view) {

        String status_value = mStatus.getText().toString();

        Intent status_intent = new Intent(SettingsActivity.this, StatusActivity.class);
        status_intent.putExtra("status_value", status_value);
        startActivity(status_intent);

      }
    });
```

```
        mImageBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {


                Intent galleryIntent = new Intent();
                galleryIntent.setType("image/*");
                galleryIntent.setAction(Intent.ACTION_GET_CONTENT);

                startActivityForResult(Intent.createChooser(galleryIntent, "SELECT IMAGE"),
GALLERY_PICK);


                /*
                CropImage.activity()
                        .setGuidelines(CropImageView.Guidelines.ON)
                        .start(SettingsActivity.this);
                        */

            }
        });


    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);

        if(requestCode == GALLERY_PICK && resultCode == RESULT_OK){

            Uri imageUri = data.getData();

            CropImage.activity(imageUri)
                    .setAspectRatio(1, 1)
                    .setMinCropWindowSize(500, 500)
                    .start(this);

            //Toast.makeText(SettingsActivity.this, imageUri, Toast.LENGTH_LONG).show();

        }


        if (requestCode == CropImage.CROP_IMAGE_ACTIVITY_REQUEST_CODE) {

            CropImage.ActivityResult result = CropImage.getActivityResult(data);

            if (resultCode == RESULT_OK) {
```

```
        mProgressDialog = new ProgressDialog(SettingsActivity.this);
        mProgressDialog.setTitle("Uploading Image...");
        mProgressDialog.setMessage("Please wait while we upload and process the
image.");
        mProgressDialog.setCanceledOnTouchOutside(false);
        mProgressDialog.show();


        Uri resultUri = result.getUri();

        File thumb_filePath = new File(resultUri.getPath());

        String current_user_id = mCurrentUser.getUid();


        Bitmap thumb_bitmap = new Compressor(this)
            .setMaxWidth(200)
            .setMaxHeight(200)
            .setQuality(75)
            .compressToBitmap(thumb_filePath);

        ByteArrayOutputStream baos = new ByteArrayOutputStream();
        thumb_bitmap.compress(Bitmap.CompressFormat.JPEG, 100, baos);
        final byte[] thumb_byte = baos.toByteArray();


        StorageReference filepath =
mImageStorage.child("profile_images").child(current_user_id + ".jpg");
        final StorageReference thumb_filepath =
mImageStorage.child("profile_images").child("thumbs").child(current_user_id + ".jpg");



        filepath.putFile(resultUri).addOnCompleteListener(new
OnCompleteListener<UploadTask.TaskSnapshot>() {
          @Override
          public void onComplete(@NonNull Task<UploadTask.TaskSnapshot> task) {

            if(task.isSuccessful()){

              final String download_url = task.getResult().getDownloadUrl().toString();

              UploadTask uploadTask = thumb_filepath.putBytes(thumb_byte);
              uploadTask.addOnCompleteListener(new
OnCompleteListener<UploadTask.TaskSnapshot>() {
                @Override
                public void onComplete(@NonNull Task<UploadTask.TaskSnapshot>
thumb_task) {

                  String thumb_downloadUrl =
thumb_task.getResult().getDownloadUrl().toString();

                                    if(thumb_task.isSuccessful()){

                                        Map update_hashMap = new HashMap();
                                        update_hashMap.put("image", download_url);
                                        update_hashMap.put("thumb_image", thumb_downloadUrl);


mUserDatabase.updateChildren(update_hashMap).addOnCompleteListener(new
OnCompleteListener<Void>() {
                                            @Override
                                            public void onComplete(@NonNull Task<Void> task) {

                                                if(task.isSuccessful()){

                                                    mProgressDialog.dismiss();
                                                    Toast.makeText(SettingsActivity.this, "Success Uploading.",
Toast.LENGTH_LONG).show();

                                                }

                                            }
                                        });


                                    } else {

                                        Toast.makeText(SettingsActivity.this, "Error in uploading
thumbnail.", Toast.LENGTH_LONG).show();
                                        mProgressDialog.dismiss();

                                    }


                                }
                            });



                        } else {

                            Toast.makeText(SettingsActivity.this, "Error in uploading.",
Toast.LENGTH_LONG).show();
                            mProgressDialog.dismiss();

                        }

                    }
                });
```

```
            } else if (resultCode ==
CropImage.CROP_IMAGE_ACTIVITY_RESULT_ERROR_CODE) {

                Exception error = result.getError();

            }
        }


    }


    public static String random() {
        Random generator = new Random();
        StringBuilder randomStringBuilder = new StringBuilder();
        int randomLength = generator.nextInt(20);
        char tempChar;
        for (int i = 0; i < randomLength; i++){
            tempChar = (char) (generator.nextInt(96) + 32);
            randomStringBuilder.append(tempChar);
        }
        return randomStringBuilder.toString();
    }


}
```

3.n. StartActivity.java

```
package com.project.jain.sikana;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

import com.project.jain.sikana.Authentication.LoginActivity;
import com.project.jain.sikana.Authentication.RegisterActivity;

import com.project.jain.sikana.R;

public class StartActivity extends AppCompatActivity {


    private Button mRegBtn;
    private Button mLoginBtn;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_start);

        mRegBtn = (Button) findViewById(R.id.start_reg_btn);
        mLoginBtn = (Button) findViewById(R.id.start_login_btn);

        mRegBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                Intent reg_intent = new Intent(StartActivity.this, RegisterActivity.class);
                startActivity(reg_intent);

            }
        });

        mLoginBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                Intent login_intent = new Intent(StartActivity.this, LoginActivity.class);
                startActivity(login_intent);

            }
        });  }
}
```

3.o. StatusActivity.java

```
package com.project.jain.sikana;

import android.app.ProgressDialog;
import android.support.annotation.NonNull;
import android.support.design.widget.TextInputLayout;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.v7.widget.Toolbar;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;

import com.project.jain.sikana.R;

public class StatusActivity extends AppCompatActivity {

    private Toolbar mToolbar;

    private TextInputLayout mStatus;
    private Button mSavebtn;


    //Firebase
    private DatabaseReference mStatusDatabase;
    private FirebaseUser mCurrentUser;


    //Progress
    private ProgressDialog mProgress;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_status);

        //Firebase
        mCurrentUser = FirebaseAuth.getInstance().getCurrentUser();
        String current_uid = mCurrentUser.getUid();

        mStatusDatabase =
FirebaseDatabase.getInstance().getReference().child("Users").child(current_uid);

        mToolbar = (Toolbar) findViewById(R.id.status_appBar);
        setSupportActionBar(mToolbar);
        getSupportActionBar().setTitle("Account Status");
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);


        String status_value = getIntent().getStringExtra("status_value");

        mStatus = (TextInputLayout) findViewById(R.id.status_input);
        mSavebtn = (Button) findViewById(R.id.status_save_btn);
        mStatus.getEditText().setText(status_value);

        mSavebtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                //Progress
                mProgress = new ProgressDialog(StatusActivity.this);
                mProgress.setTitle("Saving Changes");
                mProgress.setMessage("Please wait while we save the changes");
                mProgress.show();

                String status = mStatus.getEditText().getText().toString();

                mStatusDatabase.child("status").setValue(status).addOnCompleteListener(new
OnCompleteListener<Void>() {
                    @Override
                    public void onComplete(@NonNull Task<Void> task) {

                        if(task.isSuccessful()){

                            mProgress.dismiss();
                            Toast.makeText(StatusActivity.this, "Status updated.",
Toast.LENGTH_SHORT).show();

                        } else {

                            Toast.makeText(getApplicationContext(), "There was some error in saving
Changes.", Toast.LENGTH_LONG).show();

                        }

                    }
                });

            }
        });
    }
}
```

3.p. Users.java

```
package com.project.jain.sikana;


public class Users {

    public String name;
    public String image;
    public String status;
    public String thumb_image;
```

```java
    public Users(){
    }
    public Users(String name, String image, String status, String thumb_image) {
        this.name = name;
        this.image = image;
        this.status = status;
        this.thumb_image = thumb_image;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getImage() {
        return image;
    }

    public void setImage(String image) {
        this.image = image;
    }
    public String getStatus() {
        return status;
    }
    public void setStatus(String status) {
        this.status = status;
    }
    public String getThumb_image() {
        return thumb_image;
    }
    public void setThumb_image(String thumb_image) {
        this.thumb_image = thumb_image;
    }

}
```

3.q. UsersActivity.java

```java
package com.project.jain.sikana;

import android.content.Context;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.support.v7.widget.Toolbar;
import android.view.View;
import android.widget.TextView;

import com.firebase.ui.database.FirebaseRecyclerAdapter;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.squareup.picasso.Picasso;

import de.hdodenhof.circleimageview.CircleImageView;
import com.project.jain.sikana.R;

public class UsersActivity extends AppCompatActivity {

    private Toolbar mToolbar;

    private RecyclerView mUsersList;

    private DatabaseReference mUsersDatabase;

    private LinearLayoutManager mLayoutManager;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_users);

        mToolbar = (Toolbar) findViewById(R.id.users_appBar);
        setSupportActionBar(mToolbar);

        getSupportActionBar().setTitle("All Users");
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);

        mUsersDatabase = FirebaseDatabase.getInstance().getReference().child("Users");

        mLayoutManager = new LinearLayoutManager(this);

        mUsersList = (RecyclerView) findViewById(R.id.users_list);
        mUsersList.setHasFixedSize(true);
        mUsersList.setLayoutManager(mLayoutManager);


    }

    @Override
    protected void onStart() {
        super.onStart();

        FirebaseRecyclerAdapter<Users, UsersViewHolder> firebaseRecyclerAdapter = new
FirebaseRecyclerAdapter<Users, UsersViewHolder>(

                Users.class,
                R.layout.users_single_layout,
                UsersViewHolder.class,
                mUsersDatabase

        ) {
            @Override
            protected void populateViewHolder(UsersViewHolder usersViewHolder, Users users,
int position) {

                usersViewHolder.setDisplayName(users.getName());
                usersViewHolder.setUserStatus(users.getStatus());
                usersViewHolder.setUserImage(users.getThumb_image(), getApplicationContext());

                final String user_id = getRef(position).getKey();

                usersViewHolder.mView.setOnClickListener(new View.OnClickListener() {
                    @Override
                    public void onClick(View view) {

                        Intent profileIntent = new Intent(UsersActivity.this, ProfileActivity.class);
                        profileIntent.putExtra("user_id", user_id);
                        startActivity(profileIntent);

                    }
                });

            }
        };


        mUsersList.setAdapter(firebaseRecyclerAdapter);

    }


    public static class UsersViewHolder extends RecyclerView.ViewHolder {

        View mView;

        public UsersViewHolder(View itemView) {
            super(itemView);

            mView = itemView;

        }

        public void setDisplayName(String name){

            TextView userNameView = (TextView) mView.findViewById(R.id.user_single_name);
            userNameView.setText(name.toUpperCase());
        }

        public void setUserStatus(String status){

            TextView userStatusView = (TextView) mView.findViewById(R.id.user_single_status);
            userStatusView.setText(status);

        }

        public void setUserImage(String thumb_image, Context ctx){

            CircleImageView userImageView = (CircleImageView)
mView.findViewById(R.id.user_single_image);


Picasso.with(ctx).load(thumb_image).placeholder(R.drawable.default_avatar).into(userImageVie
w);

        }

    }

}
```

11.1.1.B UI CODE SNIPPETS (THE DESIGN PART)

1. activity_chat.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
```

```
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:background="#ededed"
  tools:context="com.project.jain.sikana.ChatActivity">

  <include
    android:id="@+id/chat_app_bar"
    layout="@layout/app_bar_layout" />

  <android.support.v4.widget.SwipeRefreshLayout
    android:id="@+id/message_swipe_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_above="@+id/linearLayout"
    android:layout_alignParentStart="true"
    android:layout_below="@+id/chat_app_bar">

    <android.support.v7.widget.RecyclerView
      android:id="@+id/messages_list"
      android:layout_width="match_parent"
      android:layout_height="match_parent"
      android:layout_above="@+id/linearLayout"
      android:layout_alignParentStart="true"

android:layout_below="@+id/chat_app_bar"></android.support.v7.widget.RecyclerView>

  </android.support.v4.widget.SwipeRefreshLayout>

  <LinearLayout
    android:id="@+id/linearLayout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_alignParentStart="true"
    android:background="@android:color/white"
    android:orientation="horizontal"
    android:weightSum="10">

    <ImageButton
      android:id="@+id/chat_add_btn"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_weight="1"
      android:alpha="0.5"
      android:background="@android:color/white"
      android:padding="10dp"
      app:srcCompat="@drawable/ic_add_black_24dp" />

    <EditText
      android:id="@+id/chat_message_view"
      android:layout_width="wrap_content"
```

```
            android:layout_height="wrap_content"
            android:layout_weight="8"
            android:background="@android:color/white"
            android:ems="10"
            android:hint="Enter Message..."
            android:inputType="textPersonName"
            android:paddingBottom="12dp"
            android:paddingLeft="10dp"
            android:paddingRight="10dp"
            android:paddingTop="14dp" />

        <ImageButton
            android:id="@+id/chat_send_btn"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:alpha="0.5"
            android:background="@android:color/white"
            android:padding="10dp"
            app:srcCompat="@drawable/ic_send_black_24dp" />
    </LinearLayout>


</RelativeLayout>
```

2. activity_login.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.project.jain.sikana.Authentication.LoginActivity">

    <include
        android:id="@+id/login_toolbar"
        layout="@layout/app_bar_layout"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginEnd="0dp"
        android:layout_marginStart="0dp"
        android:layout_marginTop="0dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
```

```
<Button
   android:id="@+id/login_btn"
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:layout_alignParentBottom="true"
   android:layout_alignParentStart="true"
   android:layout_marginEnd="20dp"
   android:background="@drawable/round_button_orange"
   android:paddingLeft="20dp"
   android:paddingRight="20dp"
   android:text="LOGIN"
   android:textColor="@android:color/white"
   app:layout_constraintEnd_toEndOf="parent"
   android:layout_marginTop="20dp"
   app:layout_constraintTop_toBottomOf="@+id/login_password" />

<TextView
   android:id="@+id/login_label"
   android:layout_width="0dp"
   android:layout_height="wrap_content"
   android:layout_marginStart="20dp"
   android:layout_marginTop="35dp"
   android:text="Login to your Account"
   android:textColor="@android:color/black"
   android:textSize="18sp"
   app:layout_constraintStart_toStartOf="parent"
   app:layout_constraintTop_toBottomOf="@+id/login_toolbar" />

<android.support.design.widget.TextInputLayout
   android:id="@+id/login_email"
   android:layout_width="0dp"
   android:layout_height="wrap_content"
   android:layout_marginEnd="20dp"
   android:layout_marginStart="20dp"
   android:layout_marginTop="35dp"
   app:layout_constraintEnd_toEndOf="parent"
   app:layout_constraintStart_toStartOf="parent"
   app:layout_constraintTop_toBottomOf="@+id/login_label">

   <EditText
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:hint="Your Email" />
</android.support.design.widget.TextInputLayout>

<android.support.design.widget.TextInputLayout
   android:id="@+id/login_password"
   android:layout_width="0dp"
   android:layout_height="wrap_content"
   android:layout_marginEnd="20dp"
   android:layout_marginStart="20dp"
```

```
        android:layout_marginTop="15dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/login_email">

        <EditText
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:inputType="textPassword"
            android:hint="Password" />
    </android.support.design.widget.TextInputLayout>

</android.support.constraint.ConstraintLayout>
```

3. activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.project.jain.sikana.MainActivity">

    <android.support.design.widget.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/appBarLayout">

        <include layout="@layout/app_bar_layout" android:id="@+id/main_page_toolbar"
/>

        <android.support.design.widget.TabLayout
            android:id="@+id/main_tabs"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"

android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar"></android.support.
design.widget.TabLayout>

    </android.support.design.widget.AppBarLayout>

    <android.support.v4.view.ViewPager
        android:id="@+id/main_tabPager"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/appBarLayout">

    </android.support.v4.view.ViewPager>
```

```
</RelativeLayout>
```

4. activity_profile.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/darkColor"
    tools:context="com.project.jain.sikana.ProfileActivity">

    <ImageView
        android:id="@+id/profile_image"
        android:layout_width="match_parent"
        android:layout_height="220dp"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:adjustViewBounds="true"
        android:scaleType="centerCrop"
        app:srcCompat="@drawable/default_avatar" />

    <TextView
        android:id="@+id/profile_displayName"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/profile_image"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="26dp"
        android:text="Display Name"
        android:textColor="@android:color/white"
        android:textSize="30sp" />

    <TextView
        android:id="@+id/profile_status"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/profile_displayName"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="12dp"
        android:text="Current User Status"
        android:textColor="@android:color/white"
        android:textSize="20sp" />

    <TextView
        android:id="@+id/profile_totalFriends"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_above="@+id/profile_send_req_btn"
        android:layout_centerHorizontal="true"
        android:layout_marginBottom="12dp"
        android:textColor="@android:color/white"
        android:textSize="16sp" />
    <!-- android:text="Total Friends" -->

    <Button
        android:id="@+id/profile_send_req_btn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_above="@+id/profile_decline_btn"
        android:layout_centerHorizontal="true"
        android:layout_marginBottom="12dp"
        android:padding="20dp"
        android:textSize="16sp"
        android:backgroundTint="@color/colorAccent"
        android:text="SEND FRIEND REQUEST"
        android:textColor="@android:color/white" />

    <Button
        android:id="@+id/profile_decline_btn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:layout_marginBottom="20dp"
        android:padding="20dp"
        android:textSize="16sp"
        android:backgroundTint="@color/colorAccent"
        android:text="DECLINE FRIEND REQUEST"
        android:textColor="@android:color/white" />

</RelativeLayout>
```

5. activity_register.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.project.jain.sikana.Authentication.RegisterActivity">
```

```
<include
    android:id="@+id/register_toolbar"
    layout="@layout/app_bar_layout"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginEnd="0dp"
    android:layout_marginStart="0dp"
    android:layout_marginTop="0dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<android.support.design.widget.TextInputLayout
    android:id="@+id/register_display_name"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginEnd="20dp"
    android:layout_marginStart="20dp"
    android:layout_marginTop="35dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/login_label">

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/display_name" />
</android.support.design.widget.TextInputLayout>

<android.support.design.widget.TextInputLayout
    android:id="@+id/register_email"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginEnd="20dp"
    android:layout_marginStart="20dp"
    android:layout_marginTop="10dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/register_display_name">

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/email"
        android:inputType="textEmailAddress" />
</android.support.design.widget.TextInputLayout>

<android.support.design.widget.TextInputLayout
    android:id="@+id/reg_password"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
```

```
        android:layout_marginEnd="20dp"
        android:layout_marginStart="20dp"
        android:layout_marginTop="10dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/register_email">

        <EditText
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="@string/password"
            android:inputType="textPassword" />
    </android.support.design.widget.TextInputLayout>

    <Button
        android:id="@+id/reg_create_btn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentStart="true"
        android:layout_marginEnd="20dp"
        android:layout_marginTop="20dp"
        android:backgroundTint="@color/colorAccent"
        android:paddingLeft="20dp"
        android:paddingRight="20dp"
        android:background="@drawable/round_button_orange"
        android:text="@string/create_account"
        android:textColor="@android:color/white"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/reg_password" />

    <TextView
        android:id="@+id/login_label"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="20dp"
        android:layout_marginTop="35dp"
        android:text="Create a New Account"
        android:textColor="@android:color/black"
        android:textSize="18sp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/register_toolbar"
        tools:layout_editor_absoluteX="26dp"
        tools:layout_editor_absoluteY="104dp" />


</android.support.constraint.ConstraintLayout>
```

6.activity_settings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:background="@drawable/gradient"
   tools:context="com.project.jain.sikana.SettingsActivity">

   <de.hdodenhof.circleimageview.CircleImageView
      android:id="@+id/settings_image"
      android:layout_width="170dp"
      android:layout_height="170dp"
      android:layout_alignParentTop="true"
      android:layout_centerHorizontal="true"
      android:layout_marginTop="30dp"
      android:src="@drawable/default_avatar" />

   <TextView
      android:id="@+id/settings_name"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_below="@+id/settings_image"
      android:layout_centerHorizontal="true"
      android:layout_marginTop="30dp"
      android:text="Display Name"
      android:textColor="@android:color/white"
      android:textSize="30sp" />

   <Button
      android:id="@+id/settings_status_btn"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_alignParentBottom="true"
      android:layout_centerHorizontal="true"
      android:layout_marginBottom="30dp"
      android:backgroundTint="@color/colorAccent"
      android:padding="20dp"
      android:textSize="16sp"
      android:text="CHANGE STATUS"
      android:textColor="@android:color/white" />

   <Button
      android:id="@+id/settings_image_btn"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_above="@+id/settings_status_btn"
      android:layout_centerHorizontal="true"
      android:layout_marginBottom="10dp"
```

```
        android:backgroundTint="@android:color/white"
        android:paddingLeft="25dp"
        android:paddingRight="25dp"
        android:paddingTop="20dp"
        android:paddingBottom="20dp"
        android:textSize="16sp"
        android:text="CHANGE IMAGE"
        android:textColor="@color/colorPrimary" />

    <TextView
        android:id="@+id/settings_status"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/settings_name"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="15dp"
        android:textSize="20sp"
        android:text="@string/default_status"
        android:textColor="@android:color/white" />

</RelativeLayout>
```

7.activity_start.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/gradient"
    tools:context="com.project.jain.sikana.StartActivity">


    <ImageView
        android:id="@+id/start_imageView_sikana"
        android:layout_width="300dp"
        android:layout_height="300dp"
        android:layout_above="@id/textView"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:src="@drawable/web_hi_res_512" />


    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_above="@+id/start_login_btn"
        android:layout_centerHorizontal="true"
        android:layout_marginBottom="50dp"
        android:text="Welcome to SIKANA - \n Teacher assistant "
        android:textAlignment="center"
        android:textColor="@android:color/white"
        android:textSize="22sp"
        android:textStyle="bold" />

    <Button
        android:id="@+id/start_login_btn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_above="@+id/start_reg_btn"
        android:layout_centerHorizontal="true"
        android:layout_marginBottom="10dp"
        android:background="@drawable/round_button_orange"
        android:backgroundTint="@color/colorAccent"
        android:paddingLeft="20dp"
        android:paddingRight="20dp"
        android:text="ALREADY HAVE ACCOUNT"
        android:textColor="@android:color/white" />

    <Button
        android:id="@+id/start_reg_btn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:layout_marginBottom="50dp"
        android:layout_marginTop="20dp"
        android:background="@drawable/round_button_orange"
        android:backgroundTint="@android:color/white"
        android:paddingLeft="25dp"
        android:paddingRight="25dp"
        android:text="@string/need_a_new_account"
        android:textColor="@color/colorPrimary" />


</RelativeLayout>
```

8. activity_status.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.project.jain.sikana.StatusActivity">
    <include
        android:id="@+id/status_appBar"
        layout="@layout/app_bar_layout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

    <android.support.design.widget.TextInputLayout
        android:id="@+id/status_input"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/status_appBar"
        android:layout_centerHorizontal="true"
        android:layout_marginLeft="20dp"
        android:layout_marginRight="20dp"
        android:layout_marginTop="40dp">

        <EditText
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Your Status" />
    </android.support.design.widget.TextInputLayout>

    <Button
        android:id="@+id/status_save_btn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignEnd="@+id/status_input"
        android:layout_below="@+id/status_input"
        android:layout_marginTop="20dp"
        android:paddingStart="10dp"
        android:paddingEnd="10dp"
        android:background="@drawable/round_button_orange"
        android:backgroundTint="@color/colorAccent"
        android:text="SAVE CHANGES"
        android:textColor="@android:color/white" />

</RelativeLayout>
```

9. activity_users.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.project.jain.sikana.UsersActivity">
    <include layout="@layout/app_bar_layout" android:id="@+id/users_appBar" />

    <android.support.v7.widget.RecyclerView
        android:id="@+id/users_list"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_alignParentStart="true"

android:layout_below="@+id/users_appBar"></android.support.v7.widget.RecyclerView>

</RelativeLayout>
```

10. app_bar_layout.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.v7.widget.Toolbar
xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/main_app_bar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@color/colorPrimary"
    android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar">

</android.support.v7.widget.Toolbar>
```

11. chat_custom_bar.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/colorPrimary">

 <de.hdodenhof.circleimageview.CircleImageView
        android:id="@+id/custom_bar_image"
        android:layout_width="36dp"
        android:layout_height="36dp"
        android:layout_alignParentEnd="true"
        android:layout_alignParentTop="true"
        android:layout_marginRight="10dp"
        />

    <TextView
        android:id="@+id/custom_bar_title"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:text="Display Name"
        android:textColor="@android:color/white"
        android:textSize="18sp" />
    <TextView
        android:id="@+id/custom_bar_seen"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentStart="true"
        android:layout_below="@+id/custom_bar_title"
        android:text="Last Seen"
        android:textColor="@android:color/white"
        android:textSize="13sp" />

</RelativeLayout>
```

12. fragment_chats.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.project.jain.sikana.Fragments.ChatsFragment">

    <!-- TODO: Update blank fragment layout -->

    <android.support.v7.widget.RecyclerView
        android:id="@+id/conv_list"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_alignParentStart="true"></android.support.v7.widget.RecyclerView>

</RelativeLayout>
```

13. fragment_friends.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.project.jain.sikana.Fragments.FriendsFragment">

    <android.support.v7.widget.RecyclerView
        android:id="@+id/friends_list"
        android:layout_width="match_parent"
        android:layout_height="match_parent">

    </android.support.v7.widget.RecyclerView>

</RelativeLayout>
```

14. message_single_layout.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:id="@+id/message_single_layout"
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:padding="20dp"
   android:weightSum="10">

   <de.hdodenhof.circleimageview.CircleImageView
      android:id="@+id/message_profile_layout"
      android:layout_width="50dp"
      android:layout_height="50dp"
      android:layout_weight="0"
      android:src="@drawable/default_avatar" />

   <TextView
      android:id="@+id/message_text_layout"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_below="@+id/name_text_layout"
      android:layout_marginLeft="10dp"
      android:layout_toEndOf="@+id/message_profile_layout"
      android:padding="0dp"
      android:text="Message Text"
      android:textColor="#444444"
      android:textSize="16sp" />

   <TextView
      android:id="@+id/name_text_layout"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_alignParentTop="true"
      android:layout_marginLeft="10dp"
      android:layout_toEndOf="@+id/message_profile_layout"
      android:text="Display Name"
      android:textColor="@android:color/black"
      android:textSize="16sp"
      android:textStyle="bold" />


   <ImageView
      android:id="@+id/message_image_layout"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_alignParentEnd="true"
      android:layout_alignStart="@+id/message_text_layout"
      android:layout_below="@+id/message_text_layout"
```

```
        android:layout_marginLeft="0dp"
        android:layout_toEndOf="@+id/message_profile_layout"
        android:padding="0dp"
        android:scaleType="centerCrop" />

</RelativeLayout>
```

15. users_single_layout.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="?android:selectableItemBackground">

    <de.hdodenhof.circleimageview.CircleImageView
        android:id="@+id/user_single_image"
        android:layout_width="64dp"
        android:layout_height="64dp"
        android:layout_marginBottom="15dp"
        android:layout_marginLeft="15dp"
        android:layout_marginTop="15dp"
        android:src="@drawable/default_avatar" />

    <TextView
        android:id="@+id/user_single_name"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignTop="@+id/user_single_image"
        android:layout_marginStart="20dp"
        android:layout_toEndOf="@+id/user_single_image"
        android:text="Display Name"
        android:textColor="@android:color/black"
        android:textSize="18sp" />

    <TextView
        android:id="@+id/user_single_status"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignStart="@+id/user_single_name"
        android:layout_below="@+id/user_single_name"
        android:layout_marginTop="10dp"
        android:text="User default Status" />

    <ImageView
        android:id="@+id/user_single_online_icon"
        android:layout_width="8dp"
        android:layout_height="wrap_content"
        android:layout_alignTop="@+id/user_single_name"
        android:layout_marginLeft="10dp"
        android:layout_toEndOf="@+id/user_single_name"
        android:visibility="invisible"
        app:srcCompat="@drawable/online_icon" />

</RelativeLayout>
```

11.1.2 Online Attendance Sheet

11.1.2.A. JAVA CODE SNIPPETS

11.1.2.1 Package ‘sheetdemo’.

1.a. GoogleService.java

```
package com.project.jain.sikana.sheetdemo;

import android.Manifest;
import android.app.Dialog;
import android.content.Context;
import android.content.Intent;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.v7.app.AppCompatActivity;
import android.widget.Toast;

import com.google.android.gms.common.ConnectionResult;
import com.google.android.gms.common.GoogleApiAvailability;
import com.google.api.client.googleapis.extensions.android.gms.auth.GoogleAccountCredential;
import com.google.api.client.util.ExponentialBackOff;
import com.google.api.services.sheets.v4.SheetsScopes;

import java.util.Arrays;
import java.util.List;

import pub.devrel.easypermissions.AfterPermissionGranted;
import pub.devrel.easypermissions.EasyPermissions;


public abstract class GoogleService extends AppCompatActivity implements
EasyPermissions.PermissionCallbacks{

   public static final String PREF_ACCOUNT_NAME = null;
   static final int REQUEST_ACCOUNT_PICKER = 1000;
   static final int REQUEST_AUTHORIZATION = 1001;
   static final int REQUEST_GOOGLE_PLAY_SERVICES = 1002;
   static final int REQUEST_PERMISSION_GET_ACCOUNTS = 1003;
   static final String[] SCOPES = {
SheetsScopes.DRIVE_FILE,SheetsScopes.SPREADSHEETS };
   public String sheetName = null;
   public String gSheetUrl = null;
   public String mEditTextInputString = null;
   public String gSheetId = null;
   GoogleAccountCredential mCredential;

   @Override
   protected void onCreate(Bundle savedInstanceState) {
       super.onCreate(savedInstanceState);
   }


   public void setmCredential(){
       this.mCredential = GoogleAccountCredential.usingOAuth2(
               this, Arrays.asList(SCOPES))
               .setBackOff(new ExponentialBackOff());
   }


   public GoogleAccountCredential getmCredential() {
       return mCredential;
   }





   /**
    * Check that Google Play services APK is installed and up to date.
    * @return true if Google Play Services is available and up to
    *     date on this device; false otherwise.
    */
   public boolean isGooglePlayServicesAvailable() {
       GoogleApiAvailability apiAvailability =
               GoogleApiAvailability.getInstance();
       final int connectionStatusCode =
               apiAvailability.isGooglePlayServicesAvailable(this);
       return connectionStatusCode == ConnectionResult.SUCCESS;
   }


   /**
    * Attempt to resolve a missing, out-of-date, invalid or disabled Google
    * Play Services installation via a user dialog, if possible.
    */
   public void acquireGooglePlayServices() {
       GoogleApiAvailability apiAvailability =
               GoogleApiAvailability.getInstance();
       final int connectionStatusCode =
               apiAvailability.isGooglePlayServicesAvailable(this);
       if (apiAvailability.isUserResolvableError(connectionStatusCode)) {
           showGooglePlayServicesAvailabilityErrorDialog(connectionStatusCode);
       }
   }

   /**
    * Display an error dialog showing that Google Play Services is missing
    * or out of date.
    * @param connectionStatusCode code describing the presence (or lack of)
    *     Google Play Services on this device.
    */
   void showGooglePlayServicesAvailabilityErrorDialog(
           final int connectionStatusCode) {
       GoogleApiAvailability apiAvailability = GoogleApiAvailability.getInstance();
       Dialog dialog = apiAvailability.getErrorDialog(
               this,
               connectionStatusCode,
               REQUEST_GOOGLE_PLAY_SERVICES);
       dialog.show();
   }


   /**
    * Checks whether the device currently has a network connection.
    * @return true if the device has a network connection, false otherwise.
    */
   public boolean isDeviceOnline() {
       ConnectivityManager connMgr =
               (ConnectivityManager) getSystemService(Context.CONNECTIVITY_SERVICE);
       NetworkInfo networkInfo = connMgr.getActiveNetworkInfo();
       return (networkInfo != null && networkInfo.isConnected());
   }

   /**
    * Attempts to set the account used with the API credentials. If an account
    * name was previously saved it will use that one; otherwise an account
    * picker dialog will be shown to the user. Note that the setting the
    * account to use with the credentials object requires the app to have the
    * GET_ACCOUNTS permission, which is requested here if it is not already
    * present. The AfterPermissionGranted annotation indicates that this
    * function will be rerun automatically whenever the GET_ACCOUNTS permission
    * is granted.
    */
   @AfterPermissionGranted(REQUEST_PERMISSION_GET_ACCOUNTS)
   public void chooseAccount() {
       if (EasyPermissions.hasPermissions(
               this, Manifest.permission.GET_ACCOUNTS)) {
           String accountName = getPreferences(Context.MODE_PRIVATE)
                   .getString(PREF_ACCOUNT_NAME, null);
           if (accountName != null) {
               mCredential.setSelectedAccountName(accountName);
           } else {
               // Start a dialog from which the user can choose an account
               startActivityForResult(
                       mCredential.newChooseAccountIntent(),
                       REQUEST_ACCOUNT_PICKER);
           }
       } else {
           // Request the GET_ACCOUNTS permission via a user dialog
           EasyPermissions.requestPermissions(
                   this,
                   "This app needs to access your Google account (via Contacts).",
                   REQUEST_PERMISSION_GET_ACCOUNTS,
                   Manifest.permission.GET_ACCOUNTS);
       }
   }


   /**
    * Respond to requests for permissions at runtime for API 23 and above.
    * @param requestCode The request code passed in
    *     requestPermissions(android.app.Activity, String, int, String[])
    * @param permissions The requested permissions. Never null.
    * @param grantResults The grant results for the corresponding permissions
    *     which is either PERMISSION_GRANTED or PERMISSION_DENIED. Never null.
    */
   @Override
   public void onRequestPermissionsResult(int requestCode,
                                          @NonNull String[] permissions,
                                          @NonNull int[] grantResults) {
       super.onRequestPermissionsResult(requestCode, permissions, grantResults);
       EasyPermissions.onRequestPermissionsResult(
               requestCode, permissions, grantResults, this);
   }



   /**
    * Callback for when a permission is granted using the EasyPermissions
    * library.
    * @param requestCode The request code associated with the requested
    *         permission
    * @param list The requested permission list. Never null.
    */
   @Override
   public void onPermissionsGranted(int requestCode, List<String> list) {
       // Do nothing.
       chooseAccount();
   }



   /**
    * Callback for when a permission is denied using the EasyPermissions
    * library.
    * @param requestCode The request code associated with the requested
    *        permission
    * @param list The requested permission list. Never null.
    */
   @Override
   public void onPermissionsDenied(int requestCode, List<String> list) {
       // Do nothing.
   }


   /**
    * Called when an activity launched here (specifically, AccountPicker
    * and authorization) exits, giving you the requestCode you started it with,
    * the resultCode it returned, and any additional data from it.
    * @param requestCode code indicating which activity result is incoming.
    * @param resultCode code indicating the result of the incoming
    *     activity result.
    * @param data Intent (containing result data) returned by incoming
    *     activity result.
    */
   @Override
   protected void onActivityResult(int requestCode, int resultCode, Intent data) {
       super.onActivityResult(requestCode,resultCode,data);
   }


   public String getgSheetId() {
       return gSheetId;
   }

   public void setgSheetId(String gSheetId) {
       this.gSheetId = gSheetId;
   }

   public String getgSheetUrl() {
       return gSheetUrl;
   }

   public void setgSheetUrl(String gSheetUrl) {
       this.gSheetUrl = gSheetUrl;
   }

   public String getSheetName() {
       return sheetName;
   }

   public void setSheetName(String sheetName) {
       this.sheetName = sheetName;
   }
```

```
    public boolean hasError(){

        if (! isGooglePlayServicesAvailable()) {
            acquireGooglePlayServices();
        } else if (mCredential.getSelectedAccountName() == null) {
            chooseAccount();
        } else if (! isDeviceOnline()) {
            return true;
        }
        return false;
    }


    public void showToast(String msg) {
        Toast.makeText(this, msg, Toast.LENGTH_SHORT).show();
    }

    public void setDialogInputValue(String inputValue){
        mEditTextInputString = inputValue;
    }

}
```

1.b. MainActivity.java

```
package com.project.jain.sikana.sheetdemo;

import android.content.Intent;
import android.os.Build;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.transition.Fade;
import android.view.View;
import android.widget.Button;


public class MainActivity extends AppCompatActivity {

    private Button take_attendance_Button,buttonOriginal;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        buttonOriginal = findViewById(R.id.newLayout);
        setupWindowAnimations();
        buttonOriginal.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent takeAttendanceIntent = new
Intent(MainActivity.this,DisplaySubjectsActivity.class);
                startActivity(takeAttendanceIntent);
            }
        });
    }

    private void setupWindowAnimations() {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
            Fade fade = new Fade();
            fade.setDuration(10000);
            getWindow().setExitTransition(fade);
        }
    }
}
```

1.c. DisplaySubjectsActivity.java

```
package com.project.jain.sikana.sheetdemo;

import android.accounts.AccountManager;
import android.annotation.SuppressLint;
import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Typeface;
import android.os.AsyncTask;
import android.os.Build;
import android.os.Bundle;
import android.support.v7.app.ActionBar;
import android.support.v7.widget.DefaultItemAnimator;
import android.support.v7.widget.GridLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.transition.Explode;
import android.transition.Transition;
import android.util.Log;
import android.view.Gravity;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.Window;
import android.widget.EditText;
import android.widget.LinearLayout;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;

import com.google.api.client.extensions.android.http.AndroidHttp;
import com.google.api.client.googleapis.extensions.android.gms.auth.GoogleAccountCredential;
import com.google.api.client.http.HttpTransport;
import com.google.api.client.json.JsonFactory;
import com.google.api.client.json.jackson2.JacksonFactory;
import com.google.api.services.drive.Drive;
import com.google.api.services.drive.model.File;
import com.google.api.services.drive.model.FileList;
import com.project.jain.sikana.models.Folders;
import com.project.jain.sikana.myAdapters.SubjectAdapter;

import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

public class DisplaySubjectsActivity extends GoogleService {

    private GoogleAccountCredential mCredential;
    private List<Folders> driveFolders;
    private SubjectAdapter subjectAdapter;
    private ProgressBar mProgressBar;
    private String newSubjectName = null;
    private boolean newSubject = false;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setupWindowAnimations();

        setContentView(R.layout.activity_display_subjects);
        setmCredential();
        mCredential = getmCredential();
        initActionbar();

        RecyclerView recyclerView = findViewById(R.id.folders_recyclerView);
        mProgressBar = findViewById(R.id.indeterminateBar);


        newSubject = false;
        makeDriveRequest();

        driveFolders = new ArrayList<>();
        subjectAdapter = new
SubjectAdapter(this,driveFolders,getmCredential().getSelectedAccountName());


        recyclerView.setHasFixedSize(true);
        final GridLayoutManager mLayoutManger = new GridLayoutManager(this,2);
        recyclerView.setLayoutManager(mLayoutManger);
        recyclerView.setItemAnimator(new DefaultItemAnimator());
        recyclerView.setMotionEventSplittingEnabled(false);
        recyclerView.setAdapter(subjectAdapter);


    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.add_subject_menu,menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {

        int id = item.getItemId();
        if(id == R.id.addnewSubjectMenu){
            newSubject = true;
            makeDriveRequest();
            return true;
        }

        return super.onOptionsItemSelected(item);
    }

    /*
    * Applying a custom view to the action bar layout
    * Linear Layout with a textview
    * and a imageview inside a linear layout
    * */

    private void initActionbar() {


        TextView actionBarSubTitle = new TextView(getApplicationContext());
        actionBarSubTitle.setText(R.string.subjectActivityTitle);
        actionBarSubTitle.setTypeface(null,Typeface.NORMAL);
        actionBarSubTitle.setGravity(Gravity.CENTER);
        actionBarSubTitle.setTextColor(getResources().getColor(R.color.colorWhite));
        actionBarSubTitle.setTextSize(20);




        LinearLayout linearLayout = new LinearLayout(getApplicationContext());
        linearLayout.addView(actionBarSubTitle);
        linearLayout.setLayoutParams(new
LinearLayout.LayoutParams(LinearLayout.LayoutParams.MATCH_PARENT,LinearLayout.Lay
outParams.MATCH_PARENT));
        linearLayout.setOrientation(LinearLayout.HORIZONTAL);
        linearLayout.setVerticalGravity(Gravity.CENTER_VERTICAL);
        if (getSupportActionBar() != null) {
            getSupportActionBar().setDisplayOptions(ActionBar.DISPLAY_SHOW_CUSTOM);
            getSupportActionBar().setCustomView(linearLayout);
        }
    }

    /*
    * Basic android activity to activity transition
    * */
    private void setupWindowAnimations() {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
            getWindow().requestFeature(Window.FEATURE_CONTENT_TRANSITIONS);
            Transition explode = new Explode();
            explode.setDuration(300);
            getWindow().setExitTransition(explode);
            getWindow().setEnterTransition(explode);
            getWindow().setReenterTransition(explode);

        }
    }


    /*
    * makeDriveRequest
    * checks for google authentication,network availability and
    * also prompts user for permissions as well as display
    * dialog to choose google account to be used
    * */

    private void makeDriveRequest(){
        if(hasError()){
            Toast.makeText(this,"Please Connect to Internet",Toast.LENGTH_SHORT).show();
        }
        else {
            // if no account is selected then displays a toast. Only done for debugging purpose
            // will be deleted later
            if (mCredential.getSelectedAccountName() == null){
                showToast("No account selected");
            }
            else if (! mCredential.getSelectedAccountName().equals(""))
//                after the user account is selected, checks whether the user
//                wants to add a new subject in the database;
                if(newSubject){
                    askSubjectNameFromDialog();
                }else {
                    new DisplaySubjectsActivity.RequestDriveTask(mCredential).execute();
                }
        }
    }



    @Override
    protected void onActivityResult(
            int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        switch(requestCode) {
            case REQUEST_GOOGLE_PLAY_SERVICES:
                if (resultCode != RESULT_OK) {
                    showToast(
                            "This app requires Google Play Services. Please install " +
                                    "Google Play Services on your device and relaunch this app.");
                } else {
                    makeDriveRequest();
                }
                break;
            case REQUEST_ACCOUNT_PICKER:
                if (resultCode == RESULT_OK && data != null &&
                        data.getExtras() != null) {
                    String accountName =
                            data.getStringExtra(AccountManager.KEY_ACCOUNT_NAME);
                    if (accountName != null) {
                        SharedPreferences settings =
                                getPreferences(Context.MODE_PRIVATE);
                        SharedPreferences.Editor editor = settings.edit();
                        editor.putString(PREF_ACCOUNT_NAME, accountName);
                        editor.apply();
                        mCredential.setSelectedAccountName(accountName);
                        makeDriveRequest();
                    }
                }
                break;
            case REQUEST_AUTHORIZATION:
                if (resultCode == RESULT_OK) {
                    makeDriveRequest();
                }
                break;
        }
    }

    public void askSubjectNameFromDialog(){


        final AlertDialog.Builder builder;
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
            builder = new AlertDialog.Builder(DisplaySubjectsActivity.this,
android.R.style.Theme_Material_Dialog_Alert);
        } else {
            builder = new AlertDialog.Builder(DisplaySubjectsActivity.this);
        }


        //add a edittext to the dialog box for user input
        final LayoutInflater inflater = getLayoutInflater();
        @SuppressLint("InflateParams") final View dialogView =
inflater.inflate(R.layout.custom_dialog, null);
        final EditText editText = dialogView.findViewById(R.id.editTextDialog);
        builder.setView(dialogView);

        //set the dialog title
        builder.setTitle(R.string.sud_dialog_title);
        builder.setCancelable(false);
        builder.setPositiveButton("OK", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int which) {
                String value = editText.getText().toString();
                if(! value.equals("")){

                    //close the fab menu
//                      if(addSubjectMenu.isOpened()){
//                          addSubjectMenu.close(true);
//                      }


                    setNewSubjectName(value);
                    new DisplaySubjectsActivity.RequestDriveTask(mCredential).execute();
                }
            }
        });
        builder.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int which) {
                setDialogInputValue("");
            }
        });

        final AlertDialog alert = builder.create();
        alert.setOnDismissListener(new DialogInterface.OnDismissListener() {
            @Override
            public void onDismiss( DialogInterface dialog) {
            }
        });
        alert.show();
    }

    public String getNewSubjectName() {
        return newSubjectName;
    }

    public void setNewSubjectName(String newSubjectName) {
        this.newSubjectName = newSubjectName;
    }

    private class RequestDriveTask extends AsyncTask<Void,Void,List<String>>{
        private Drive driveService = null;


        RequestDriveTask(GoogleAccountCredential credential){
            HttpTransport transport = AndroidHttp.newCompatibleTransport();
            JsonFactory jsonFactory = JacksonFactory.getDefaultInstance();

             driveService = new Drive.Builder(transport, jsonFactory, credential)
                    .setApplicationName("SheetDemo")
                    .build();

        }

        @Override
        protected List<String> doInBackground(Void... voids) {
            try {
                if (newSubject){
                    addNewFolder();
                }else {
                    listDriveFolders();
                }
            } catch (IOException e) {
                Log.d("BackGroundTask",e.getMessage());
                e.printStackTrace();
            }

            return null;
        }


        @Override
        protected void onPreExecute(){
            mProgressBar.setVisibility(View.VISIBLE);
        }


        @Override
        protected void onPostExecute(List<String> output){
            if(newSubject) {
                newSubject = false;
                setNewSubjectName(null);
            }
            subjectAdapter.notifyDataSetChanged();
            mProgressBar.setVisibility(View.GONE);
            Toast.makeText(DisplaySubjectsActivity.this,"Drive Request
Completed",Toast.LENGTH_SHORT).show();
        }
        private void listDriveFolders() throws IOException {
            String pageToken = null;
            do {

                FileList foldersList = driveService.files().list()
                        .setSpaces("drive")
                        .setQ("mimeType = 'application/vnd.google-apps.folder' and trashed = false")
                        .setPageToken(pageToken)
                        .setFields("nextPageToken, files(id, name, parents)")
                        .execute();
                for (File folders:foldersList.getFiles()){
                    byte classes = listDriveFiles(folders.getId());
                    driveFolders.add(new Folders(folders.getName(),folders.getId(),classes));
                }

                pageToken = foldersList.getNextPageToken();
            } while (pageToken != null);

        }

        private void addNewFolder() throws IOException {


            File fileMetadata = new File();
            fileMetadata.setName(getNewSubjectName());
            fileMetadata.setMimeType("application/vnd.google-apps.folder");


            File file = driveService.files().create(fileMetadata)
                    .setFields("id, name, spaces, parents")
                    .execute();

            File spreadSheetFile = new File();
            List<String> parents = new ArrayList<>();
            parents.add(file.getId());
            spreadSheetFile.setName("DEMO");
            spreadSheetFile.setMimeType("application/vnd.google-apps.spreadsheet");
            spreadSheetFile.setParents(parents);


//            request a new spreadsheet using drive api
//            this methods returns a File object
            driveService.files().create(spreadSheetFile)
                    .setFields("id, name, spaces, parents")
                    .execute();


            driveFolders.add(new Folders(file.getName(),file.getId(), (byte) 1));

            //System.out.println("spreadsheetName"+spreadsheet.getParents());
        }


        private byte listDriveFiles(String subjectId) throws IOException {

            byte classes = 0;
            String query = "mimeType = 'application/vnd.google-apps.spreadsheet' and trashed =
false and \'"+subjectId +"\' in parents";

                FileList result = driveService.files().list()
                        .setSpaces("drive")
                        .setQ(query)
                        .setFields("nextPageToken, files(id, name, parents)")
                        .execute();

                    classes+=result.getFiles().size();

            return classes;
        }
    }


}
```

1.d. DisplaySheetsActivity.java

```
package com.project.jain.sikana.sheetdemo;

import android.accounts.AccountManager;
import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.res.Resources;
import android.graphics.Color;
import android.graphics.Typeface;
import android.graphics.drawable.ColorDrawable;
import android.os.AsyncTask;
import android.os.Build;
import android.os.Bundle;
import android.support.v7.app.ActionBar;
import android.support.v7.widget.DefaultItemAnimator;
import android.support.v7.widget.GridLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.transition.Explode;
import android.transition.Fade;
import android.transition.Transition;
import android.transition.Visibility;
import android.util.Log;
import android.view.Gravity;
import android.view.LayoutInflater;
import android.view.View;
import android.view.Window;
import android.view.WindowManager;
import android.widget.EditText;
import android.widget.LinearLayout;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;

import com.github.clans.fab.FloatingActionButton;
import com.github.clans.fab.FloatingActionMenu;
import com.google.api.client.extensions.android.http.AndroidHttp;
import com.google.api.client.googleapis.extensions.android.gms.auth.GoogleAccountCredential;
import com.google.api.client.http.HttpTransport;
import com.google.api.client.json.JsonFactory;
import com.google.api.client.json.jackson2.JacksonFactory;
import com.google.api.services.drive.Drive;
import com.google.api.services.drive.model.File;
import com.google.api.services.drive.model.FileList;
import com.google.api.services.sheets.v4.Sheets;
import com.project.jain.sikana.models.SheetsModel;
import com.project.jain.sikana.myAdapters.ClassAdapter;

import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

import static com.project.jain.sikana.myUtils.MaterialUtils.modifyBrightness;

public class DisplaySheetsActivity extends GoogleService {


    private GoogleAccountCredential mCredential;
    private String subjectId; //folderId

    private List<SheetsModel> classList;
    private ClassAdapter classAdapter;
    private ProgressBar mProgressBar;
    private String newClassName = null; //newSheetName
    private boolean newClass = false;
    private FloatingActionMenu addNewClassMenu;
    private String subName;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setupWindowAnimations();
    setContentView(R.layout.activity_display_sheets);

    setmCredential();
    mCredential = getmCredential();

    if(getIntent().getExtras() != null) {
      subjectId = getIntent().getExtras().getString("folderId");
      subName = getIntent().getExtras().getString("subjectName");
      String selectedGoogleAccount = getIntent().getExtras().getString("selectedAccount");
      mCredential.setSelectedAccountName(selectedGoogleAccount);

      //passing the color to action bar
      int backgroundColor = (int)getIntent().getExtras().get("actionBarColor");
      initActionbar(backgroundColor);

getWindow().getDecorView().setBackgroundColor(modifyBrightness(backgroundColor,(float)0.
4));

      //change the status bar color according to the color value obtained from previous activity
      if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
        Window window = getWindow();

window.clearFlags(WindowManager.LayoutParams.FLAG_TRANSLUCENT_STATUS);

window.addFlags(WindowManager.LayoutParams.FLAG_DRAWS_SYSTEM_BAR_BACKGR
OUNDS);
        window.setStatusBarColor(modifyBrightness(backgroundColor,(float)0.5));
      }

    }


    initFab();

    classList = new ArrayList<>();
    classAdapter = new
ClassAdapter(this,classList,mCredential.getSelectedAccountName());
    RecyclerView recyclerView = findViewById(R.id.recyclerViewDS);
    mProgressBar = findViewById(R.id.indeterminateBarDS);


    final GridLayoutManager gridLayoutManager = new GridLayoutManager(this,3);
    recyclerView.setHasFixedSize(true);
    recyclerView.setItemAnimator(new DefaultItemAnimator());
    recyclerView.setLayoutManager(gridLayoutManager);
    recyclerView.setMotionEventSplittingEnabled(false);
    recyclerView.setAdapter(classAdapter);

    newClass = false;

    makeDriveRequest();
    }


    private void setupWindowAnimations() {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
            getWindow().requestFeature(Window.FEATURE_CONTENT_TRANSITIONS);
            Transition explode = new Explode();
            explode.setDuration(300);
            getWindow().setExitTransition(explode);
            getWindow().setEnterTransition(explode);
            getWindow().setReenterTransition(explode);

        }
    }

    private void initActionbar(int actionbarColor){



        //title for the action bar
        TextView actionBarTitle = new TextView(getApplicationContext());
        actionBarTitle.setText(subName);
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
            actionBarTitle.setTransitionName("subjectTextView");
        }
        actionBarTitle.setGravity(Gravity.START);
        actionBarTitle.setTextColor(getResources().getColor(R.color.colorWhite));
        actionBarTitle.setTypeface(null, Typeface.BOLD);
        actionBarTitle.setTextSize(20);
        actionBarTitle.setPadding(30,10,0,0);


        TextView actionBarSubTitle = new TextView(getApplicationContext());
        actionBarSubTitle.setText("Class list for attendance");
        actionBarSubTitle.setTypeface(null,Typeface.ITALIC);
        actionBarSubTitle.setPadding(30,0,0,0);
        actionBarSubTitle.setTextColor(getResources().getColor(R.color.colorWhite));
        actionBarSubTitle.setTextSize(14);

        LinearLayout linearLayout = new LinearLayout(getApplicationContext());
        linearLayout.addView(actionBarTitle);
        linearLayout.addView(actionBarSubTitle);
        linearLayout.setLayoutParams(new
LinearLayout.LayoutParams(LinearLayout.LayoutParams.MATCH_PARENT,LinearLayout.Lay
outParams.MATCH_PARENT));
        linearLayout.setOrientation(LinearLayout.VERTICAL);
        if (getSupportActionBar() != null) {
            getSupportActionBar().setDisplayOptions(ActionBar.DISPLAY_SHOW_CUSTOM);
            getSupportActionBar().setCustomView(linearLayout);
            getSupportActionBar().setBackgroundDrawable(new ColorDrawable(actionbarColor));
        }

    }

    private void initFab(){
        addNewClassMenu = findViewById(R.id.fabMenuDS);
        final FloatingActionButton addNewClass = findViewById(R.id.newSheetDS);

        addNewClassMenu.setOnMenuToggleListener(new
FloatingActionMenu.OnMenuToggleListener() {
            @Override
            public void onMenuToggle(boolean opened) {

            }
        });

        addNewClassMenu.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if(addNewClassMenu.isOpened()){
                    addNewClassMenu.close(true);
                }
            }
        });

        addNewClass.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                newClass = true;
                makeDriveRequest();
            }
        });
    }


    private void makeDriveRequest(){

            if (mCredential.getSelectedAccountName() == null){
                showToast("No account selected");
            }
            else {
                if(newClass){
                    setClassNamefromDialog("Class Name");
                }else {
                    new DisplaySheetsActivity.RequestDriveTask(mCredential).execute();
                }

            }
    }
    @Override
    protected void onActivityResult(
            int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        switch(requestCode) {
            case REQUEST_GOOGLE_PLAY_SERVICES:
                if (resultCode != RESULT_OK) {
                    showToast(
                            "This app requires Google Play Services. Please install " +
                                    "Google Play Services on your device and relaunch this app.");
                } else {
                    makeDriveRequest();
                }
                break;
            case REQUEST_ACCOUNT_PICKER:
                if (resultCode == RESULT_OK && data != null &&
                        data.getExtras() != null) {
                    String accountName =
                            data.getStringExtra(AccountManager.KEY_ACCOUNT_NAME);
                    if (accountName != null) {
                        SharedPreferences settings =
                                getPreferences(Context.MODE_PRIVATE);
                        SharedPreferences.Editor editor = settings.edit();
                        editor.putString(PREF_ACCOUNT_NAME, accountName);
                        editor.apply();
                        mCredential.setSelectedAccountName(accountName);
                        makeDriveRequest();
                    }
                }
                break;
            case REQUEST_AUTHORIZATION:
                if (resultCode == RESULT_OK) {
                    makeDriveRequest();
                }
                break;
        }
    }

    public void setClassNamefromDialog(String title){

        final AlertDialog.Builder builder;
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
            builder = new AlertDialog.Builder(DisplaySheetsActivity.this,
android.R.style.Theme_Material_Dialog_Alert);
        } else {
            builder = new AlertDialog.Builder(DisplaySheetsActivity.this);
        }


        //adding a textview to the dialog box
        final LayoutInflater inflater = getLayoutInflater();
        final View dialogView = inflater.inflate(R.layout.custom_dialog, null);
        final EditText editText = dialogView.findViewById(R.id.editTextDialog);
        builder.setView(dialogView);

        //settings the dialog title
        builder.setTitle(title);
        builder.setCancelable(false);
        builder.setPositiveButton("OK", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int which) {
                String value = editText.getText().toString();
                if(! value.equals("")){

                    //closing the fab menu
                    if(addNewClassMenu.isOpened()){
                        addNewClassMenu.close(true);;
                    }


                    newClassName = value;
                    new DisplaySheetsActivity.RequestDriveTask(mCredential).execute();
                }
            }
        });
        builder.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int which) {
                setDialogInputValue("");
            }
        });

        final AlertDialog alert = builder.create();
        alert.setOnDismissListener(new DialogInterface.OnDismissListener() {
            @Override
            public void onDismiss( DialogInterface dialog) {
            }
        });
        alert.show();
    }

    private class RequestDriveTask extends AsyncTask<Void,Void,List<String>> {
        private Drive driveService = null;
        private Sheets sheetService = null;
        private byte totalClasses = -1;
        private String folderNamefromDialog;

        RequestDriveTask(GoogleAccountCredential credential){
            HttpTransport transport = AndroidHttp.newCompatibleTransport();
            JsonFactory jsonFactory = JacksonFactory.getDefaultInstance();

            driveService = new Drive.Builder(transport, jsonFactory, credential)
                    .setApplicationName("SheetDemo")
                    .build();
```

```
      sheetService = new com.google.api.services.sheets.v4.Sheets.Builder(
          transport, jsonFactory, credential)
          .setApplicationName("SheetDemo")
          .build();

    }

    @Override
    protected List<String> doInBackground(Void... voids) {
      try {
        if (newClass){
          addNewClass();
        }else {
          listDriveFiles();
        }
      } catch (IOException e) {
        Log.d("BackGroundTask",e.getMessage());
        e.printStackTrace();
      }

      return null;
    }


    @Override
    protected void onPreExecute(){
      mProgressBar.setVisibility(View.VISIBLE);
    }


    @Override
    protected void onPostExecute(List<String> output){
      if(newClass) {
        newClass = false;
        newClassName = null;
      }
      classAdapter.notifyDataSetChanged();
      mProgressBar.setVisibility(View.GONE);
      Toast.makeText(DisplaySheetsActivity.this,"Drive Request
Completed",Toast.LENGTH_SHORT).show();
    }

    private void listDriveFiles() throws IOException {
      String pageToken = null;
      String query = "mimeType = 'application/vnd.google-apps.spreadsheet' and trashed =
false and \'"+subjectId +"\' in parents";
      do {
        FileList result = driveService.files().list()
            .setSpaces("drive")
            // Available Query parameters here:
```

```
                        //https://developers.google.com/drive/v3/web/search-parameters
                        .setQ(query)
                        .setPageToken(pageToken)
                        .setFields("nextPageToken, files(id, name, parents)")
                        .execute();



                for (File file : result.getFiles()) {
                    //System.out.println("files:"+file.getParents());
                    classList.add(new SheetsModel(file.getName(),file.getId()));
                }

                pageToken = result.getNextPageToken();
            } while (pageToken != null);
        }

        private void addNewClass() throws IOException {


            File spreadSheetFile = new File();
            List<String> parents = new ArrayList<>();
            parents.add(subjectId);
            spreadSheetFile.setName(newClassName);
            spreadSheetFile.setMimeType("application/vnd.google-apps.spreadsheet");
            spreadSheetFile.setParents(parents);

            File spreadsheet = driveService.files().create(spreadSheetFile)
                    .setFields("id, name, spaces, parents")
                    .execute();


            classList.add(new SheetsModel(spreadsheet.getName(),spreadsheet.getId()));

            //System.out.println("spreadsheetName"+spreadsheet.getParents());

        }
    }
}
```

1.e Attendance.java

```
package com.project.jain.sikana.sheetdemo;

import android.annotation.SuppressLint;
import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.DialogInterface;
import android.graphics.Point;
import android.os.AsyncTask;
import android.os.Build;
import android.os.Bundle;
import android.os.Handler;
import android.support.design.widget.FloatingActionButton;
import android.support.v7.widget.DefaultItemAnimator;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.util.Log;
import android.view.Display;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

import com.google.api.client.extensions.android.http.AndroidHttp;
import com.google.api.client.googleapis.extensions.android.gms.auth.GoogleAccountCredential;
import
com.google.api.client.googleapis.extensions.android.gms.auth.GooglePlayServicesAvailabilityIO
Exception;
import
com.google.api.client.googleapis.extensions.android.gms.auth.UserRecoverableAuthIOException
;
import com.google.api.client.googleapis.json.GoogleJsonResponseException;
import com.google.api.client.http.HttpTransport;
import com.google.api.client.json.JsonFactory;
import com.google.api.client.json.jackson2.JacksonFactory;
import com.google.api.services.drive.Drive;
import com.google.api.services.sheets.v4.Sheets;
import com.google.api.services.sheets.v4.model.AddConditionalFormatRuleRequest;
import com.google.api.services.sheets.v4.model.AddSheetRequest;
import com.google.api.services.sheets.v4.model.AutoResizeDimensionsRequest;
import com.google.api.services.sheets.v4.model.BatchUpdateSpreadsheetRequest;
import com.google.api.services.sheets.v4.model.BooleanCondition;
import com.google.api.services.sheets.v4.model.BooleanRule;
import com.google.api.services.sheets.v4.model.CellData;
import com.google.api.services.sheets.v4.model.CellFormat;
import com.google.api.services.sheets.v4.model.Color;
import com.google.api.services.sheets.v4.model.ConditionValue;
import com.google.api.services.sheets.v4.model.ConditionalFormatRule;
import com.google.api.services.sheets.v4.model.DimensionRange;
```

```java
import com.google.api.services.sheets.v4.model.GridProperties;
import com.google.api.services.sheets.v4.model.GridRange;
import com.google.api.services.sheets.v4.model.RepeatCellRequest;
import com.google.api.services.sheets.v4.model.Request;
import com.google.api.services.sheets.v4.model.SheetProperties;
import com.google.api.services.sheets.v4.model.TextFormat;
import com.google.api.services.sheets.v4.model.ValueRange;
import com.project.jain.sikana.models.Student;
import com.project.jain.sikana.myAdapters.StudentAttendanceAdapter;

import java.io.IOException;
import java.io.PrintWriter;
import java.io.StringWriter;
import java.security.GeneralSecurityException;
import java.text.DateFormatSymbols;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.LinkedHashMap;
import java.util.List;
import java.util.Map;

import static com.project.jain.sikana.myUtils.SpreadSheetUtils.ColumnNumberToLetter;


public class Attendance extends GoogleService {

    private StudentAttendanceAdapter studentAttendanceAdapter;
    private RecyclerView recyclerView;
    private List<Student> studentList;
    private ProgressDialog mProgress;
    private Map<String,Byte> studentsAttendance = null;
    private byte totalAttendance = -1;
    private boolean submitAttendance = false;
    private String nextColumn = null,lastColumn = null;
    private byte currentColumnNo = -1;
    private int screenHeight;
    private String sheetName;



    private Button submitBtn;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_attendance);


//        @params setSheetName check whether the sheetName should be
//        for previous month
```

```
        setSheetName(false);

        //setting google credentials
        setmCredential();
        mCredential = getmCredential();

        Display display = getWindowManager().getDefaultDisplay();
        Point size = new Point();
        display.getSize(size);
        screenHeight = size.y;

        if(getIntent().getExtras() != null) {
            String accountName = getIntent().getExtras().getString("selectedAccount");
            mCredential.setSelectedAccountName(accountName);
        }

        if (getIntent().getExtras().getString("selectedSheet") != null) {
            setgSheetId(getIntent().getExtras().getString("selectedSheet"));
            startTakingAttendance();
        } else {
            Toast.makeText(this, "No students Found", Toast.LENGTH_SHORT).show();
        }


    }

    private void startTakingAttendance(){
        mProgress = new ProgressDialog(this);
        mProgress.setMessage("Calling Google Sheets API ...");


        recyclerView = findViewById(R.id.recycler_view);
        submitBtn = findViewById(R.id.submit_Attendance);
        final FloatingActionButton auto_scroll_fab = findViewById(R.id.auto_scroll);
        auto_scroll_fab.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                recyclerView.smoothScrollBy(0, screenHeight);

            }
        });


        //initialize student list and student adapter
        studentList = new ArrayList<>();
        studentAttendanceAdapter = new StudentAttendanceAdapter(this,studentList);

        recyclerView.setHasFixedSize(true);
        final LinearLayoutManager mLayoutManager = new LinearLayoutManager(this);
        recyclerView.setLayoutManager(mLayoutManager);
        recyclerView.setItemAnimator(new DefaultItemAnimator());
        recyclerView.setMotionEventSplittingEnabled(false);
        recyclerView.setAdapter(studentAttendanceAdapter);

        recyclerView.addOnScrollListener(new RecyclerView.OnScrollListener() {
            @Override
            public void onScrolled(RecyclerView recyclerView, int dx, int dy) {
                super.onScrolled(recyclerView, dx, dy);

Log.d("findLastCom",String.valueOf(mLayoutManager.findLastCompletelyVisibleItemPositio
n()));
                if( mLayoutManager.findLastCompletelyVisibleItemPosition() >=
studentAttendanceAdapter.getItemCount()){
                    if(auto_scroll_fab.isShown()) auto_scroll_fab.hide();
                } else {
                    if(!auto_scroll_fab.isShown()) auto_scroll_fab.show();
                }
            }
        });




        //clear the adapter if is called twice
        studentAttendanceAdapter.clear();
        submitBtn.setVisibility(View.INVISIBLE);
        submitBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                submitAttendance = true;
                showConfirmationDialog();
            }
        });
        submitAttendance = false;
        new MakeRequestTask(mCredential).execute();

    }

    private void setSheetName(boolean previousMonth){
        Date today = new Date();
        @SuppressLint("SimpleDateFormat") SimpleDateFormat dateFormat = new
SimpleDateFormat("MM");
        String date = dateFormat.format(today);

        if (previousMonth) {
            this.sheetName = new DateFormatSymbols().getShortMonths()[Integer.parseInt(date) -
2];
        }else{
            this.sheetName = new DateFormatSymbols().getShortMonths()[Integer.parseInt(date) -
1];
        }
    }
    public String getSheetName(){
        return sheetName;
    }


    public void prepareStudentList(){
        if (studentsAttendance.size() >0) {
            for (Map.Entry<String,Byte> entry: studentsAttendance.entrySet()) {
                Student student = new Student(entry.getKey(),entry.getValue());
                studentList.add(student);
            }
            studentAttendanceAdapter.notifyDataSetChanged();
//            scrollViewHeight = scrollView.getChildAt(0).getHeight();
            submitBtn.setVisibility(View.VISIBLE);

        }else{
            Toast.makeText(this,"studentsName is Null",Toast.LENGTH_SHORT).show();
        }
    }

    @Override
    public void onBackPressed() {

        final AlertDialog.Builder builder;
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
            builder = new AlertDialog.Builder(this, android.R.style.Theme_Material_Dialog_Alert);
        } else {
            builder = new AlertDialog.Builder(this);
        }
        builder.setTitle("Cancel Taking Attendance?");
        builder.setCancelable(false);
        builder.setMessage("Your Attendance won't be udpated");
        builder.setPositiveButton(R.string.yes, new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog, int which) {
                        Attendance.super.onBackPressed();
                    }
                });
        builder.setNegativeButton(R.string.no, new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog, int which) {
                        //do nothing is if the user clicks cancel button
                    }
                });

        final AlertDialog alert = builder.create();
        alert.show();
    }



    /**
```

```
 * Overriding all the parent methods and calling the parent class handle all
 * life cycle processes
 * */
@Override
public void onRestart(){
    super.onRestart();
}

@Override
public void onDestroy(){
    mProgress.hide();
    mProgress.dismiss();
    super.onDestroy();
}

@Override
protected void onStart() {
    super.onStart();
}

@Override
protected void onResume() {
    super.onResume();
}

@Override
protected void onStop() {
    mProgress.hide();
    mProgress.dismiss();
    super.onStop();
}

@Override
protected void onPause() {
    super.onPause();
}

private void showConfirmationDialog(){
    final AlertDialog.Builder builder;
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
        builder = new AlertDialog.Builder(this, android.R.style.Theme_Material_Dialog_Alert);
    } else {
        builder = new AlertDialog.Builder(this);
    }
    builder.setTitle("Confirm Attendance Update?");
    builder.setCancelable(true);
    builder.setPositiveButton(R.string.submit, new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int which) {
            new MakeRequestTask(mCredential).execute();
        }
    });
```

```
        builder.setNegativeButton(R.string.review_once, new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int which) {
                //do nothing is if the user clicks cancel button
            }
        });
        final AlertDialog alert = builder.create();
        alert.show();
    }

    private class MakeRequestTask extends AsyncTask<Void, Void, List<String>> {
        private Exception mLastError = null;
        private Sheets mSheetService = null;
        private Drive mDriveService = null;

        MakeRequestTask(GoogleAccountCredential credential) {
            HttpTransport transport = AndroidHttp.newCompatibleTransport();
            JsonFactory jsonFactory = JacksonFactory.getDefaultInstance();


            mSheetService = new com.google.api.services.sheets.v4.Sheets.Builder(
                    transport, jsonFactory, credential)
                    .setApplicationName("SheetDemo")
                    .build();
            mDriveService = new Drive.Builder(transport, jsonFactory, credential)
                    .setApplicationName("SheetDemo")
                    .build();
        }

        /**
         * Background task to call Google Sheets API.
         * @param params no parameters needed for this task.
         */
        @Override
        protected List<String> doInBackground(Void... params) {
            try {

                if (!submitAttendance){
                    return getDataFromApi();
                }else{
                    if (studentAttendanceAdapter.getStudentsAttendance().size() > 0) {

updateAttendance(studentAttendanceAdapter.getStudentsAttendance(),nextColumn,totalAtte
ndance);
                    } else {
                        Handler handler = new Handler(Attendance.this.getMainLooper());
                        handler.post(new Runnable() {
                            public void run() {
                                Toast.makeText(Attendance.this, "Attendence List is empty",
Toast.LENGTH_LONG).show();
                            }
                        });
            }
            return null;
        }

    } catch (final Exception e) {
        mLastError = e;
        cancel(true);
        StringWriter sw = new StringWriter();
        PrintWriter pw = new PrintWriter(sw);
        e.printStackTrace(pw);
        String sStackTrace = sw.toString(); // stack trace as a string
        System.out.println(sStackTrace);
        return null;
    }
}

/**
 * Fetch a list of names,last attendance value and total attendance value from spreadsheet
 * @return List of student details
 * @throws IOException check whether data is fetched from api
 */
private List<String> getDataFromApi() throws IOException, GeneralSecurityException {
    List<String> results = new ArrayList<>();
    /*
     *   If no Google Spreadsheet is found then ask user to create a new google sheet;
     *
     */



    if (getgSheetId() == null) {
        results.add("NoSheetId");
        return results;
    }

    if(mSheetService == null || mDriveService == null){
        results.add("NoService");
        return results;
    }

    currentColumnNo = getLastColumnNumber();

    nextColumn = ColumnNumberToLetter(currentColumnNo+1);
    results = getValuesFromLastColumn(currentColumnNo);
    return results;
}



@Override
protected void onPreExecute() {
```

```
            mProgress.show();
        }

        @Override
        protected void onPostExecute(List<String> output) {
            if(!submitAttendance) {
                prepareStudentList();
                mProgress.hide();
                mProgress.dismiss();
            }else{
                mProgress.hide();
                mProgress.dismiss();
                Toast.makeText(Attendance.this,"Successfully Updated
Attendance",Toast.LENGTH_SHORT).show();
                finish();
            }

        }

        @Override
        protected void onCancelled() {
            mProgress.hide();
            if (mLastError != null) {
                if (mLastError instanceof GooglePlayServicesAvailabilityIOException) {
                    showGooglePlayServicesAvailabilityErrorDialog(
                            ((GooglePlayServicesAvailabilityIOException) mLastError)
                                    .getConnectionStatusCode());
                } else if (mLastError instanceof UserRecoverableAuthIOException) {
                    startActivityForResult(
                            ((UserRecoverableAuthIOException) mLastError).getIntent(),
                            REQUEST_AUTHORIZATION);
                } else {
                    showToast("ERROR encountered");
                }
            } else {
                Toast.makeText(Attendance.this,"Request
cancelled.",Toast.LENGTH_SHORT).show();
            }
        }


        byte getLastColumnNumber() throws IOException {


            String range = getSheetName()+"!A4:Z4";
            byte lastColumn = -1;
            ValueRange response;

            System.out.println("range "+range);

            try {
```

```
            response = this.mSheetService.spreadsheets().values()
                    .get(getgSheetId(), range)
                    .execute();
        }catch (GoogleJsonResponseException e) {
            createNewSheet();

//              retry again with new sheet created
            response = this.mSheetService.spreadsheets().values()
                    .get(getgSheetId(), range)
                    .execute();
        }

        List<List<Object>> values = response.getValues();


        if (values != null) {
            for (List row : values) {
                lastColumn = (byte)row.size();
            }
        } else{
            showToast("NO VALUES RETURNED");
        }

        return lastColumn;

    }


    List <String> getValuesFromLastColumn(byte lastColumnNumber) throws IOException {
        studentsAttendance = new LinkedHashMap<>();
        lastColumn = ColumnNumberToLetter(lastColumnNumber);
        String lastColumnRange = getSheetName()+"!A2:"+lastColumn;
        List <String> output = new ArrayList<>();


        ValueRange response = this.mSheetService.spreadsheets().values()
                    .get(getgSheetId(), lastColumnRange)
                    .execute();
        List<List<Object>> values = response.getValues();

        List firstrow = null;
        if (values != null) {
            firstrow = values.get(0);

            //removing the first row value as it contains total attendance count
            //not the information of student
            values.remove(0);

            values.remove(0); //also removing the second row as it has table headers
            for (List row : values) {
                if (lastColumnNumber < 3)
```

```
                        studentsAttendance.put((String)row.get(0),(byte)0);
                    else{

studentsAttendance.put((String)row.get(0),(byte)Integer.parseInt((String)row.get(lastColumnN
umber-1)));
                    }
                }
            }

            if (lastColumnNumber < 3){
                totalAttendance = (byte)0;
            }else{
                if(firstrow != null)
                    totalAttendance = (byte)Integer.parseInt((String)firstrow.get(lastColumnNumber-
1));
            }
            studentAttendanceAdapter.setTotalAttendance((totalAttendance));
            return output;
        }



        void updateAttendance(List<Object> attendanceResults,String column,byte
totalAttendance) {

            try {
                if(getgSheetId() != null && mSheetService!=null) {
                    String writeRange = getSheetName()+"!"+column+"1:"+column;
                    List<List<Object>> writeData = new ArrayList<>();

                    Date today = new Date();
                    @SuppressLint("SimpleDateFormat") SimpleDateFormat dateFormat = new
SimpleDateFormat("dd/MM/yy");
                    String date = dateFormat.format(today);
                    attendanceResults.add(0,date);
                    attendanceResults.add(1,String.valueOf(totalAttendance+1));
                    attendanceResults.add(2,""); //the third row won't be having any values
                    writeData.add(attendanceResults);
                    ValueRange vr = new
ValueRange().setValues(writeData).setMajorDimension("COLUMNS");
                    try {

//                          AppendValuesResponse result =

                        //updating the data to the attendance sheet directly
//                          these values also can be returned to variable result declared above
                        mSheetService.spreadsheets().values()
                        .append(gSheetId, writeRange, vr)
                        .setValueInputOption("RAW")
                        .execute();
                        formatAttendance();
                    }catch (Exception e){
                        StringWriter sw = new StringWriter();
                        PrintWriter pw = new PrintWriter(sw);
                        e.printStackTrace(pw);
//                        String sStackTrace = sw.toString(); // stack trace as a string
                        //System.out.println(sStackTrace); //Debugging
                    }

                }else{
                    showToast("no id found");
                }
            } catch (Exception e) {
                // handle exception
            }
        }


        void formatAttendance() throws IOException{
            byte currRow = 4;
            List<Byte> attendanceBoolean =
studentAttendanceAdapter.getCurrentAttendanceBoolean();
            //specifying where the filters should be applied

            Color redColor = new Color().setRed((float)0.8).setGreen((float)0.2);
            /*
             * The logic for cell formatting is that it checks the previous attendance
             * value of a student and formats the cell.
             * The syntax for the below formatting can be found at:
             *  URL : https://developers.google.com/sheets/api/samples/conditional-formatting
             *
             * */

            AddConditionalFormatRuleRequest conditionalFormat;
            ConditionalFormatRule conditionalFormatRule;
            List<Request> requests = new ArrayList<>();
            List<GridRange> ranges;
            String condition;
            for(Byte attendance: attendanceBoolean){
                if (currentColumnNo > 2) {
                    condition = "=EQ(" +nextColumn + String.valueOf(currRow )  + "," +
lastColumn + String.valueOf(currRow) + ")";
                }else{
                    condition = "=EQ("+nextColumn + String.valueOf(currRow)+",0)";
                    RepeatCellRequest cellRequest = new RepeatCellRequest()
                            .setRange(new GridRange()
                                    .setStartColumnIndex((int)currentColumnNo)
                                    .setEndRowIndex(0)
                                    .setEndRowIndex(1))
                            .setCell(new CellData()
                                    .setUserEnteredFormat(new CellFormat()
                                            .setTextFormat(new TextFormat()
                          .setForegroundColor(new Color()
                              .setRed((float)0.067)
                              .setGreen((float)0.423)
                              .setBlue((float)0.117))
                          .setItalic(true)
                          .setBold(true))
                      .setHorizontalAlignment("CENTER")))
              .setFields("userEnteredFormat(textFormat,horizontalAlignment)");

          requests.add(new Request().setRepeatCell(cellRequest));
        }

        if(attendance == 0) {
          conditionalFormat = new AddConditionalFormatRuleRequest();
          conditionalFormatRule = new ConditionalFormatRule();
          ranges = new ArrayList<>();

          GridRange columnRange = new GridRange()
              .setStartRowIndex((int) currRow-1)
              .setEndRowIndex((int) currRow)
              .setStartColumnIndex((int) currentColumnNo)
              .setEndColumnIndex(currentColumnNo + 1);


          CellFormat cellFormat = new CellFormat()
              .setBackgroundColor(redColor);

          ranges.add(columnRange);


          List<ConditionValue> conditionValues = new ArrayList<>();
          conditionValues.add(new ConditionValue().setUserEnteredValue(condition));
          BooleanRule booleanRule = new BooleanRule()
              .setCondition(new BooleanCondition()
                  .setType("CUSTOM_FORMULA")
                  .setValues(conditionValues))
              .setFormat(cellFormat);


          conditionalFormatRule.setRanges(ranges);
          conditionalFormatRule.setBooleanRule(booleanRule);


          conditionalFormat.setRule(conditionalFormatRule);


          requests.add(new Request().setAddConditionalFormatRule(conditionalFormat));
          Log.d("condition",condition);
          Log.d("formatCell", nextColumn + String.valueOf(currRow+1) + ":" +
nextColumn + String.valueOf(currRow+1));
        }
        currRow++;
      }

      AutoResizeDimensionsRequest dimensionsRequest = new
AutoResizeDimensionsRequest()
          .setDimensions(new DimensionRange()
          .setDimension("COLUMNS")
          .setStartIndex((int)currentColumnNo)
          .setEndIndex(currentColumnNo+1));

      requests.add(new Request().setAutoResizeDimensions(dimensionsRequest));

      requests.add(new Request().setRepeatCell(new RepeatCellRequest()
          .setRange(new GridRange()
          .setStartColumnIndex((int)currentColumnNo)
          .setEndColumnIndex(currentColumnNo+1))
          .setCell(new CellData()
              .setUserEnteredFormat(new CellFormat()
              .setHorizontalAlignment("CENTER")))
          .setFields("userEnteredFormat(horizontalAlignment)")));

      BatchUpdateSpreadsheetRequest formatReq = new BatchUpdateSpreadsheetRequest()
          .setRequests(requests);

//       request to format the attendance
//       the request returns a BatchUpdateSpreadsheetResponse object
//       which can be used to check for results
      mSheetService.spreadsheets()
          .batchUpdate(getgSheetId(),formatReq)
          .execute();

    }


    void createNewSheet() throws IOException{

      BatchUpdateSpreadsheetRequest addNewSheetRequest = new
BatchUpdateSpreadsheetRequest();
      AddSheetRequest sheetRequest = new AddSheetRequest();
      SheetProperties properties = new SheetProperties();

      setSheetName(true);
      String range = getSheetName()+"!A:B";

      ValueRange response = this.mSheetService.spreadsheets().values()
          .get(getgSheetId(), range)
          .execute();
      List<List<Object>> values = response.getValues();
//       if (values != null){
//         for(List row : values) {
//
//              }
//          }


//          setSheetName(false);
//          range = getSheetName()+"!A:B";
//          properties.setTitle(getSheetName());
//          GridProperties mGridProperties = new GridProperties();
//          mGridProperties.setRowCount(2);
//          mGridProperties.setColumnCount(2);
//          properties.setGridProperties(mGridProperties);
//          sheetRequest.setProperties(properties);
//          List<Request> mRequestList = new ArrayList<>();
//          mRequestList.add(new Request().setAddSheet(sheetRequest));
//          addNewSheetRequest.setRequests(mRequestList);
//          mSheetService.spreadsheets()
//                  .batchUpdate(getgSheetId(),addNewSheetRequest)
//                  .execute();
//          //add the data in the new sheet

        }


    }



}
```

11.1.2.2Package ‘models’.

2.a Folders.java

```
package com.project.jain.sikana.models;


public class Folders {

    private String folderName;
    private String folderId;
    private byte totalClasses;

    public Folders(){

    }

    public Folders(String folderName,String folderId,byte totalClasses){
        this.folderName = folderName;
        this.folderId = folderId;
        this.totalClasses = totalClasses;
    }


    public String getFolderName() {
        return folderName;
    }

    public void setFolderName(String folderName) {
        this.folderName = folderName;
    }

    public String getFolderId() {
        return folderId;
    }

    public void setFolderId(String folderId) {
        this.folderId = folderId;
    }

    public byte getTotalClasses() {
        return totalClasses;
    }

    public void setTotalClasses(byte totalClasses) {
        this.totalClasses = totalClasses;
    }
}
```

2.b SheetsModel.java

```
package com.project.jain.sikana.models;


public class SheetsModel {


    private String sheetName;
    private String sheetId;
    private String totalStudents;


    public SheetsModel(String sheetName,String sheetId){
        this.sheetName = sheetName;
        this.sheetId = sheetId;
    }

    public String getSheetName() {
        return sheetName;
    }

    public void setSheetName(String sheetName) {
        this.sheetName = sheetName;
    }

    public String getSheetId() {
        return sheetId;
    }

    public void setSheetId(String sheetId) {
        this.sheetId = sheetId;
    }

    public String getTotalStudents() {
        return totalStudents;
    }

    public void setTotalStudents(String totalStudents) {
        this.totalStudents = totalStudents;
    }
}
```

2.c. Student.java

```
package com.project.jain.sikana.models;


public class Student {
    private String studentName;
    private byte attendance;


    public Student(){

    }

    public Student(String studentName,byte attendance) {
        this.studentName = studentName;
        this.attendance = attendance;
    }

    public String getStudentName() {
        return studentName;
    }

    public void setStudentName(String studentName) {
        this.studentName = studentName;
    }
    public byte getAttendance() {
        return attendance;
    }

    public void setAttendance(byte attendance) {
        this.attendance = attendance;
    }
}
```

11.1.2.3 Package ‘myAdapters’.

3.a. ClassAdapter.java

```
package com.project.jain.sikana.myAdapters;

import android.content.Context;
import android.content.Intent;
import android.support.v7.widget.CardView;
import android.support.v7.widget.PopupMenu;
import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageButton;
import android.widget.TextView;
import android.widget.Toast;

import com.project.jain.sikana.models.SheetsModel;
import com.project.jain.sikana.sheetdemo.Attendance;
import com.project.jain.sikana.sheetdemo.R;

import java.util.List;

import static com.project.jain.sikana.myUtils.MaterialUtils.getLightRandomColors;


public class ClassAdapter extends RecyclerView.Adapter<ClassAdapter.MyViewHolder> {

    private Context context;
    private List<SheetsModel> mSheetsList;
    private String selectedGoogleAccount;


    public ClassAdapter(Context context, List<SheetsModel> mSheetsList, String
selectedGoogleAccount){
        this.context = context;
        this.mSheetsList = mSheetsList;
        this.selectedGoogleAccount = selectedGoogleAccount;
    }

    @Override
    public MyViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        View itemView = LayoutInflater.from(parent.getContext())
                .inflate(R.layout.sheet_cardview,parent,false);
        return new MyViewHolder(itemView);
    }

    @Override
    public void onBindViewHolder(final ClassAdapter.MyViewHolder holder, int position) {
        holder.position = position;
        SheetsModel sheets = mSheetsList.get(holder.position);
        String sheetName = sheets.getSheetName();
        final String sheetId = sheets.getSheetId();
        holder.sheetView.setCardBackgroundColor(getLightRandomColors(holder.position));

        holder.sheetNameTxtView.setText(sheetName);

        holder.sheetView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                showPopUpMenu(holder.sheetView,holder.position,sheetId);
            }
        });
    }

    @Override
    public int getItemCount() {
        return mSheetsList.size();
    }

    public void clear() {
        int size = this.mSheetsList.size();
        this.mSheetsList.clear();
        notifyItemRangeRemoved(0, size);
    }

    /**
     * Very Common bug i.e. values in the recyclerview changes as you scroll down
     * Thus overriding the method getItemViewType will remember the position of your views
     * and restore their values.
     */
    @Override
    public int getItemViewType(int position){
        return position;
    }
```

```
private void showToast(String message){
    if(message!=null){
        Toast.makeText(context,message,Toast.LENGTH_SHORT).show();
    }
}

private void showPopUpMenu(View view,final int position,final String sheetId){
    PopupMenu popupMenu = new PopupMenu(view.getContext(),view);
    final MenuInflater menuInflater = popupMenu.getMenuInflater();
    menuInflater.inflate(R.menu.sheet_more_menu,popupMenu.getMenu());
    popupMenu.setOnMenuItemClickListener(new PopupMenu.OnMenuItemClickListener() {
        @Override
        public boolean onMenuItemClick(MenuItem menuItem) {
            switch (menuItem.getItemId()){
                case R.id.sheetAttendanceMenu:
                    showToast("take Attendance");
                    startAttendance(sheetId);
                    return true;

                case R.id.sheetDeleteMenu:
                    mSheetsList.remove(position);
                    notifyDataSetChanged();
                    showToast("sheet deleted");
                    return true;

                case R.id.sheetViewMenu:
                    showToast("view attendance");
                    return true;
            }
            return false;
        }
    });
    popupMenu.show();
}

private void startAttendance(String sheetId){
    if(  sheetId != null && selectedGoogleAccount != null){

        Intent takeAttendanceIntent = new Intent(context,Attendance.class);
        takeAttendanceIntent.putExtra("selectedSheet", sheetId);
        takeAttendanceIntent.putExtra("selectedAccount", selectedGoogleAccount);
        context.startActivity(takeAttendanceIntent);
    }else{
        showToast("No Attendance Chosen");
        showToast("sheetId: "+sheetId);
        showToast("Name:" + selectedGoogleAccount);
    }
}

class MyViewHolder extends RecyclerView.ViewHolder{
        TextView sheetNameTxtView;
        int position;
        CardView sheetView;


        MyViewHolder(View view) {
            super(view);
            sheetNameTxtView = view.findViewById(R.id.sheetName);
            sheetView = view.findViewById(R.id.classCardViewLayout);

        }
    }
}
```

3.b. SubjectAdapter.java

```
package com.project.jain.sikana.myAdapters;

import android.app.Activity;
import android.app.ActivityOptions;
import android.content.Context;
import android.content.Intent;
import android.content.res.ColorStateList;
import android.os.Build;
import android.support.annotation.RequiresApi;
import android.support.v7.widget.CardView;
import android.support.v7.widget.PopupMenu;
import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import com.project.jain.sikana.models.Folders;
import com.project.jain.sikana.sheetdemo.DisplaySheetsActivity;
import com.project.jain.sikana.sheetdemo.R;

import java.util.List;

import static com.project.jain.sikana.myUtils.MaterialUtils.getRandomColor;

public class SubjectAdapter extends RecyclerView.Adapter<SubjectAdapter.MyViewHolder> {

    private Context context;
    private List<Folders> folderList;
    private String selectGoogleAccount;

    public SubjectAdapter(Context context, List<Folders> folderList, String
selectGoogleAccount){
        this.context = context;
        this.folderList = folderList;
        this.selectGoogleAccount = selectGoogleAccount;
    }

    @Override
    public MyViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        View itemView = LayoutInflater.from(parent.getContext())
                .inflate(R.layout.subject_cardview,parent,false);
//        Log.d("onCreateViewHolder","Called");
        return new MyViewHolder(itemView);
    }

    @RequiresApi(api = Build.VERSION_CODES.LOLLIPOP)
    @Override
    public void onBindViewHolder(final MyViewHolder holder, final int position) {
        holder.position = position;
        final Folders folders = folderList.get(holder.position);
        final String folderId = folders.getFolderId();
        String totalClasses = "Total Class: "+String.valueOf(folders.getTotalClasses());
        holder.subjectName.setText(folders.getFolderName());
        holder.totalClasses.setText(totalClasses);
        final int color = getRandomColor(holder.position);

holder.circular_image_view.setImageTintList(ColorStateList.valueOf(getRandomColor(holder.p
osition)));



//        handling the popup menu for the subject add button
        holder.myView.setOnLongClickListener(new View.OnLongClickListener() {
            @Override
            public boolean onLongClick(View view) {
                showPopupMenu(holder.myView,position);
                return true;
            }
        });

        holder.myView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent showSheets = new Intent(context, DisplaySheetsActivity.class);
                showSheets.putExtra("folderId",folderId);
                showSheets.putExtra("selectedAccount",selectGoogleAccount);
                showSheets.putExtra("subjectName",folders.getFolderName());
                showSheets.putExtra("actionBarColor",color);
                if (android.os.Build.VERSION.SDK_INT >=
android.os.Build.VERSION_CODES.LOLLIPOP) {
                    ActivityOptions options =
ActivityOptions.makeSceneTransitionAnimation((Activity)context,holder.subjectName,"totalCl
asses");
                    context.startActivity(showSheets,options.toBundle());
                }else {
                    context.startActivity(showSheets);
                }
            }
        });

    }

    @Override
    public int getItemCount() {
        return folderList.size();
    }

    public void clear() {
        int size = this.folderList.size();
        this.folderList.clear();
        notifyItemRangeRemoved(0, size);
    }

    /**
     * Very Common bug i.e. values in the recyclerview changes as you scroll down
     * Thus overriding the method getItemViewType will remember the position of your views
     * and restore their values.
     */
    @Override
    public int getItemViewType(int position) {
        return position;
    }

    private void showPopupMenu(final View view, final int position){
        final PopupMenu popup = new PopupMenu(view.getContext(),view);
        MenuInflater inflater = popup.getMenuInflater();
        inflater.inflate(R.menu.subject_more_menu,popup.getMenu());
        popup.setOnMenuItemClickListener(new PopupMenu.OnMenuItemClickListener() {
            @Override
            public boolean onMenuItemClick(MenuItem menuItem) {
                switch (menuItem.getItemId()){
                    case R.id.deleteMenu:
                        folderList.remove(position);
                        notifyDataSetChanged();
                        Toast.makeText(context,"Item Deleted",Toast.LENGTH_SHORT).show();
                        return true;
                    case R.id.addnewSheetMenu:
                        Toast.makeText(context,"New Sheet Added",Toast.LENGTH_SHORT).show();
                        return  true;
                }
                return false;
            }
        });
        popup.show();
    }

    class MyViewHolder extends RecyclerView.ViewHolder{

        TextView subjectName;
        int position;
        CardView myView;
        ImageView circular_image_view;
        TextView totalClasses;

        MyViewHolder(View view) {
            super(view);
            subjectName = view.findViewById(R.id.subjectNameTxtView);
            myView = view.findViewById(R.id.folderCardViewLayout);
            totalClasses = view.findViewById(R.id.classesTotal);
            circular_image_view = view.findViewById(R.id.imageView1);
        }
    }
}
```

3.c. StudentAdapter.java

```
package com.project.jain.sikana.myAdapters;

import android.content.Context;
import android.graphics.Point;
import android.support.v7.widget.CardView;
import android.support.v7.widget.RecyclerView;
import android.util.Log;
import android.view.Display;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.view.WindowManager;
import android.widget.FrameLayout;
import android.widget.TextView;

import com.project.jain.sikana.models.Student;
import com.project.jain.sikana.sheetdemo.R;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.LinkedHashMap;
import java.util.List;
import java.util.Map;
```

```
public class StudentAttendanceAdapter extends
RecyclerView.Adapter<StudentAttendanceAdapter.MyViewHolder> {


    private Context context;
    private List<Student> studentList;
    private Map<String,Byte> attendanceTracker = null;
    private Map<String,Byte> booleanAttendance = null;
    private Byte totalAttendance = -1;
    private int NUMBERS_OF_ITEM_TO_DISPLAY = 6;
    public class MyViewHolder extends RecyclerView.ViewHolder{

        public TextView student_name;
        public int position;
        public CardView myView;

        public MyViewHolder(View view) {
            super(view);
            student_name = view.findViewById(R.id.student_name);
            myView = view.findViewById(R.id.student_details_cardview);
        }
    }

    public StudentAttendanceAdapter(Context context, List<Student> Student) {
        this.context = context;
        this.studentList = Student;
        this.attendanceTracker = new LinkedHashMap<>();
        this.booleanAttendance = new LinkedHashMap<>();
    }

    @Override
    public MyViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        View itemView = LayoutInflater.from(parent.getContext())
                .inflate(R.layout.student_detail_view,parent,false);
//        Log.d("onCreateViewHolder","Called");
        itemView.getLayoutParams().height = (int) ( getScreenHeight()/
NUMBERS_OF_ITEM_TO_DISPLAY);
        return new MyViewHolder(itemView);
    }

    @Override
    public void onBindViewHolder(final MyViewHolder holder, int position) {
//        Log.d("onBindViewHolder","Called");
        holder.position = position;
        final Student student = studentList.get(holder.position);
        final String studentName = student.getStudentName();
        final byte currentAttendance = student.getAttendance();
        holder.student_name.setText(studentName);

        if(booleanAttendance.get(studentName) == null) {
            attendanceTracker.put(studentName, currentAttendance);
            booleanAttendance.put(studentName, (byte) 0);
        }
        holder.myView.setOnClickListener(new View.OnClickListener() {
            boolean isClicked = false;
            @Override
            public void onClick(View v) {
                /**
                 * When absent button is pressed the background color and text color is changes.
                 * And also with that change we have to change the color of the absent button.
                 * This is required when the user wants to change the button click. i.e.
                 * have already clicked a button before but again want to change the decision
                 * */

                if(!isClicked) {
                    isClicked = true;
                    attendanceTracker.put(studentName,(byte)(currentAttendance+1));
                    booleanAttendance.put(studentName,(byte)1);
                    holder.student_name.setTextColor(v.getResources().getColor(R.color.colorWhite));

holder.myView.setBackgroundColor(v.getResources().getColor(R.color.colorGreen));
                } else{
                    isClicked = false;
                    attendanceTracker.put(studentName,currentAttendance);
                    booleanAttendance.put(studentName,(byte)0);
                    holder.student_name.setTextColor(v.getResources().getColor(R.color.colorBlack));

holder.myView.setBackgroundColor(v.getResources().getColor(R.color.colorWhite));
                }
            }
        });
    }

    @Override
    public int getItemCount() {
        return studentList.size();
    }


    public void clear() {
        int size = this.studentList.size();
        this.studentList.clear();
        notifyItemRangeRemoved(0, size);
    }
    /**
     * Very Common bug i.e. values in the recyclerview changes as you scroll down
     * Thus overriding the method getItemViewType will remember the position of your views
     * and restore their values.
     */
    @Override
    public int getItemViewType(int position) {
        return position;
    }


    public List<Object> getStudentsAttendance(){
        List<Object> attendanceCollector = new ArrayList<>();
        if(attendanceTracker.size() > 0){
            for(Map.Entry<String,Byte> eachEntry: attendanceTracker.entrySet()){
                attendanceCollector.add(eachEntry.getValue());
            }

        }
        return attendanceCollector;
    }

    public List<Byte> getCurrentAttendanceBoolean(){
        List<Byte> attendanceCollector = new ArrayList<>();
        if(booleanAttendance.size() > 0){
            for(Map.Entry<String,Byte> eachEntry: booleanAttendance.entrySet()){
                attendanceCollector.add(eachEntry.getValue());
                Log.d("StudentName: ",eachEntry.getKey()+" : "+eachEntry.getValue());
            }
        }
        return attendanceCollector;
    }

    public void setTotalAttendance(byte totalAttendance){
        this.totalAttendance = totalAttendance;
    }

    public int getScreenHeight() {

        WindowManager wm = (WindowManager)
context.getSystemService(Context.WINDOW_SERVICE);
        Display display = wm.getDefaultDisplay();
        Point size = new Point();
        display.getSize(size);

        return size.y;
    }

}
```

11.1.2.4 Package ‘myUtils’.

4.a. ArrayUtils.java

```
package com.project.jain.sikana.myUtils;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;


public class ArrayUtils {


    public static String listToString(List<String> list){
        StringBuilder buf = new StringBuilder();
        for (String each: list) buf.append(each).append(",");
        String s = buf.toString();
        return s;
    }

    public static List<String> stringToList(String string){
        List<String> list = new ArrayList<String>();
        list = Arrays.asList(string.split(","));
        return list;
    }
}
```

4.b. MaterialUtils.java

```
package com.project.jain.sikana.myUtils;

import android.graphics.Color;

import java.util.ArrayList;
import java.util.List;


public class MaterialUtils {

    public static int getRandomColor(int position){
        List<String> colorsList = new ArrayList<>();
        colorsList.add("#26A69A");//teal
        colorsList.add("#42A5F5");//blue
        colorsList.add("#66BB6A");//green
        colorsList.add("#90A4AE");//blue grey
        colorsList.add("#29B6F6");//lightblue
        colorsList.add("#26C6DA");//cyan
        colorsList.add("#BDBDBD");//grey
        String randomColor = colorsList.get(position%colorsList.size());
        return Color.parseColor(randomColor);
    }


    public static int getLightRandomColors(int position){
        List<String>colorsList = new ArrayList<>();
        colorsList.add("#EF5350"); //red
        colorsList.add("#66BB6A"); //green
        colorsList.add("#FFA726");//orange
        colorsList.add("#26A69A"); //teal
        colorsList.add("#FF7043"); //deep orange
        colorsList.add("#AB47BC");//Purple
        String randomLightColor = colorsList.get(position%colorsList.size());
        return Color.parseColor(randomLightColor);
    }

    public static int modifyBrightness(int color, float factor) {
        float[] hsv = new float[3];
        Color.colorToHSV(color, hsv);
        hsv[2] *= factor;
        return Color.HSVToColor(hsv);
    }

}
```

4.c. SpreadSheetUtils.java

```
package com.project.jain.sikana.myUtils;


public class SpreadSheetUtils {

    /**
     * ColumnNumberToLetter Converts the column number to letter
     * this method will to get the range of last column in letter
     * such that we can post the value in the last column
     * */

    public static String ColumnNumberToLetter(int inputColumnNumber) {
        String outputColumnName = "";
        int Base = 26;
        String chars = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";

        int TempNumber = inputColumnNumber;
        while (TempNumber > 0) {
            int position = TempNumber % Base;
            outputColumnName = (position == 0 ? 'Z' : chars.charAt(position > 0 ? position - 1 : 0))
+ outputColumnName;
            TempNumber = (TempNumber - 1) / Base;
        }
        return outputColumnName;
    }

    public static int ColumnLetterToNumber(String inputColumnName) {
        int outputColumnNumber = 0;

        if (inputColumnName == null || inputColumnName.length() == 0) {
            throw new IllegalArgumentException("Input is not valid!");
        }

        int i = inputColumnName.length() - 1;
        int t = 0;
        while (i >= 0) {
            char curr = inputColumnName.charAt(i);
            outputColumnNumber = outputColumnNumber + (int) Math.pow(26, t) * (curr - 'A' + 1);
            t++;
            i--;
        }
        return outputColumnNumber;
    }
}
```

11.1.2.B. UI CODE SNIPPETS (The Design Part)

11.1.2.5Layout Folder Contents.

5.a. activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.project.jain.sikana.sheetdemo.MainActivity">


    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent">

        <Button
            android:id="@+id/newLayout"
            style="@style/Base.Widget.AppCompat.Button.Borderless"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:text="Click Me"
            android:textAllCaps="false"
            android:textSize="20sp"
            />


    </LinearLayout>


</android.support.constraint.ConstraintLayout>
```

5.b. sheet_cardview.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.v7.widget.CardView
xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   android:id="@+id/classCardViewLayout"
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:elevation="3dp"
   app:cardCornerRadius="5dp"
   app:cardUseCompatPadding="true">

   <LinearLayout
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:orientation="horizontal"
      android:weightSum="5">

      <TextView
         android:id="@+id/sheetName"
         android:layout_width="match_parent"
         android:layout_height="wrap_content"
         android:layout_gravity="center"
         android:layout_marginEnd="@dimen/margin_padding_small"
         android:layout_marginStart="@dimen/margin_padding_small"
         android:gravity="center"
         android:lines="3"
         android:padding="@dimen/margin_padding_small"
         android:textColor="@color/colorWhite"
         android:textSize="@dimen/textsize_medium"
         android:textStyle="normal" />
   </LinearLayout>


</android.support.v7.widget.CardView>
```

5.c. subject_cardview.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.v7.widget.CardView
xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   android:id="@+id/folderCardViewLayout"
   android:layout_width="match_parent"
   android:layout_height="150dp"
   android:layout_margin="6dp"
   android:fitsSystemWindows="true"
```

```
    app:cardBackgroundColor="@color/background"
    app:cardCornerRadius="1dp"
    app:cardElevation="1dp"
    >
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:padding="@dimen/margin_padding_small"
        android:weightSum="3"
        >
        <FrameLayout
            android:layout_width = "70dp"
            android:layout_height = "70dp"
            android:layout_gravity="center">
            <include layout="@layout/circular_image_view"/>
        </FrameLayout>

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical"
            android:weightSum="2">

            <TextView
                android:id="@+id/subjectNameTxtView"
                android:layout_width="match_parent"
                android:layout_height="0dp"
                android:layout_gravity="center"
                android:layout_weight="1"
                android:gravity="center"
                android:lines="2"
                android:textColor="@color/colorWhite"
                android:textSize="20sp"
                android:textStyle="normal"
                android:transitionName="subjectTextView" />
            <TextView
                android:id="@+id/classesTotal"
                android:layout_width="match_parent"
                android:layout_height="0dp"
                android:layout_weight="0.5"
                android:gravity="center"
                android:lines="1"
                android:textColor="@color/classNumberGrey"
                android:textSize="@dimen/textsize_small"
                android:textStyle="normal|bold"/>

        </LinearLayout>
    </LinearLayout>

</android.support.v7.widget.CardView>
```

5.d. student_detail_view.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<android.support.v7.widget.CardView
xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="match_parent"
  android:layout_height="wrap_content"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  app:cardBackgroundColor="@color/colorWhite"
  android:id="@+id/student_details_cardview"
  app:cardCornerRadius="0dp"
  android:elevation="2dp">

      <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:padding="@dimen/margin_padding_small"
        android:gravity="center"
        android:weightSum="10">

        <ImageView
          android:layout_width="wrap_content"
          android:layout_height="match_parent"
          android:src="@drawable/user"
          android:adjustViewBounds="true"
          android:padding="@dimen/margin_padding_small"
          android:maxWidth="80dp"
          android:maxHeight="80dp"
          android:layout_weight="1"
          android:scaleType="fitCenter"
          />

        <TextView
          android:layout_width="match_parent"
          android:layout_height="match_parent"
          android:layout_weight="7"
          android:layout_marginStart="@dimen/text_size_header"
          android:gravity = "center_vertical|start"
          android:layout_gravity="start"
          android:textColor="@color/colorBlack"
          android:textStyle="normal"
          android:textSize="@dimen/textsize_medium_small"
          android:id="@+id/student_name"/>

      </LinearLayout>

</android.support.v7.widget.CardView>
```

5.e. custom_dialog.xml

```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="10dp"
    android:orientation="vertical">

        <EditText
            android:id="@+id/editTextDialog"
            android:layout_width="match_parent"
            android:layout_gravity="center"
            android:hint="@string/add_sub_hint"
            android:layout_height="wrap_content"
            android:textColor="@color/colorBlack"/>

</LinearLayout>
```

5.f. circular_image_view.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.v7.widget.CardView
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="70dp"
    android:layout_height="70dp"
    android:layout_centerHorizontal="true"
    android:innerRadius="0dp"
    android:shape="ring"
    android:thicknessRatio="1.9"
    app:cardBackgroundColor="@color/colorWhite"
    app:cardCornerRadius="35dp">

    <ImageView
        android:id="@+id/imageView1"
        android:layout_width="45dp"
        android:layout_height="45dp"
        android:layout_gravity="center"
        android:contentDescription="@string/subjectImgViewDesctription"
        android:src="@drawable/subject"
        />
</android.support.v7.widget.CardView>
```

5.g. activity_display_subjects.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:layout_marginTop="@dimen/margin_padding_small"
  tools:context="com.project.jain.sikana.sheetdemo.DisplaySubjectsActivity">

  <ProgressBar
    android:id="@+id/indeterminateBar"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:foregroundGravity="center_horizontal"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    />

  <android.support.v7.widget.RecyclerView
    android:id="@+id/folders_recyclerView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:scrollbars="none"/>


</android.support.constraint.ConstraintLayout>
```

5.h. activity_display_sheets.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  tools:context="com.project.jain.sikana.sheetdemo.DisplaySheetsActivity">

  <ProgressBar
    android:id="@+id/indeterminateBarDS"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:foregroundGravity="center_horizontal"
    app:layout_constraintBottom_toBottomOf="parent"
```

```
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        />

    <android.support.v7.widget.RecyclerView
        android:id="@+id/recyclerViewDS"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:scrollbars="none"/>

    <com.github.clans.fab.FloatingActionMenu
        android:id="@+id/fabMenuDS"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentEnd="true"
        android:background="@color/transparent"
        app:fab_colorRipple="#99FFFFFF"
        app:fab_shadowColor="#66000000"
        app:fab_showShadow="true"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:menu_fab_size="normal"
        app:menu_labels_colorNormal="#333333"
        app:menu_labels_colorPressed="#444444"
        app:menu_labels_colorRipple="#66FFFFFF"
        app:menu_labels_ellipsize="end"
        app:menu_labels_maxLines="-1"
        app:menu_labels_position="left"
        app:menu_labels_showShadow="true"
        app:menu_labels_singleLine="true"
        app:menu_openDirection="up">

        <com.github.clans.fab.FloatingActionButton
            android:id="@+id/newSheetDS"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:src="@drawable/ic_action_create_sheet"
            app:fab_colorNormal="@color/sheetd_primary"
            app:fab_colorPressed="@color/sheetd_primary_dark"
            app:fab_label="@string/addNewClass"
            app:fab_size="mini" />
    </com.github.clans.fab.FloatingActionMenu>

</android.support.constraint.ConstraintLayout>
```

5.i. activity_attendance.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context="com.project.jain.sikana.sheetdemo.Attendance">

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="parent">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            android:weightSum="10">

        <view
            android:id="@+id/recycler_view"
            class="android.support.v7.widget.RecyclerView"
            android:layout_width="match_parent"
            android:layout_height="0dp"
            android:layout_weight="9.5"
            android:clipToPadding="false"/>

            <Button
                android:id="@+id/submit_Attendance"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_gravity="center"
                android:layout_margin="@dimen/margin_padding_small"
                android:background="@color/colorBlack"
                android:text="@string/submit_btn"
                android:textColor="@color/colorWhite"/>

        </LinearLayout>
    </ScrollView>

    <android.support.design.widget.FloatingActionButton
        android:id="@+id/auto_scroll"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="@dimen/fab_margin"
        android:src="@drawable/down_icon"
        android:tint="@android:color/white"
        app:backgroundTint="@color/colorAccent"
        app:borderWidth="1dp"
        app:elevation="8dp"
        app:fabSize="normal"
        app:layout_constraintBottom_toTopOf="parent"
        app:layout_constraintEnd_toEndOf="parent" />

</android.support.constraint.ConstraintLayout>
```

11.1.2.6‘menu’ folder contents.

6.a add_subject_menu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">

    <item
        android:id="@+id/addnewSubjectMenu"
        android:icon="@drawable/ic_add_subject"
        android:orderInCategory="100"
        android:title="@string/new_subject"
        app:showAsAction="always"
        />
</menu>
```

6.b. sheet_more_menu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    >

    <item
        android:id="@+id/sheetAttendanceMenu"
        android:orderInCategory="100"
        android:title="@string/sheetAttendanceMenu" />

    <item
        android:id="@+id/sheetViewMenu"
        android:orderInCategory="101"
        android:title="@string/sheetViewMenu" />

    <item
        android:id="@+id/sheetDeleteMenu"
        android:orderInCategory="102"
        android:title="@string/delete"/>
</menu>
```

6.c. subject_more_menu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
  >

  <item
    android:id="@+id/addnewSheetMenu"
    android:orderInCategory="1"
    android:title="@string/addSheetMenu" />


  <item
    android:id="@+id/deleteMenu"
    android:orderInCategory="2"
    android:title="@string/delete"/>
</menu>
```

11.1.2.7 'value' folder contents

7.a. Colors.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <color name="colorPrimary">#03a9f4</color>
  <color name="colorPrimaryDark">#007ac1</color>
  <color name="colorAccent">#FF8300</color>
  <color name="colorBlack">#000000</color>
  <color name="colorWhite">#ffffff</color>
  <color name="colorGreen">#008975</color>
  <color name="classNumberGrey">#FF949494</color>

  <color name="sheetd_primary">#00BCD4</color>
  <color name="sheetd_primary_dark">#0097A7</color>

  <color name="transparent">#00000000</color>
  <color name="background">#3A366F</color>
  <color name="popupBackground">#232323</color>
</resources>
```

7.b. Dimens.xml

```
<resources>
    <!-- SP adjusts according the users setting, DP is static -->
    <dimen name="textsize_medium">18sp</dimen>
    <dimen name="textsize_medium_small">15sp</dimen>
    <dimen name="textsize_small">14sp</dimen>


    <!-- Default screen margins, per the Android Design guidelines. -->
    <dimen name="card_height">304dp</dimen>
    <dimen name="md_keylines">16dp</dimen>
    <dimen name="tile_padding">8dp</dimen>
    <dimen name="tile_height">160dp</dimen>
    <dimen name="article_keylines">72dp</dimen>
    <dimen name="avator_size">40dp</dimen>
    <dimen name="card_title_height">40dp</dimen>
    <dimen name="article_titles">20sp</dimen>
    <dimen name="cards_button_width">48dp</dimen>
    <dimen name="cards_button_height">48dp</dimen>
    <dimen name="article_subheading">16sp</dimen>
    <dimen name="card_image_height">200dp</dimen>
    <dimen name="list_body">14sp</dimen>
    <dimen name="navheader_height">222dp</dimen>


    <!-- Typography -->
    <dimen name="material_typography_button">15sp</dimen>
    <dimen name="material_typography_caption">13sp</dimen>
    <dimen name="material_typography_body1">15sp</dimen>
    <dimen name="material_typography_body2">15sp</dimen>
    <dimen name="material_typography_subheading">17sp</dimen>
    <dimen name="material_typography_title">21sp</dimen>
    <dimen name="material_typography_headline">24sp</dimen>
    <dimen name="material_typography_display1">34sp</dimen>
    <dimen name="material_typography_display2">45sp</dimen>
    <dimen name="material_typography_display3">56sp</dimen>
    <dimen name="material_typography_display4">112sp</dimen>



    <!-- Text Size -->
    <dimen name="text_size_header">20sp</dimen>



    <!-- Margin & padding -->
    <dimen name="activity_horizontal_margin">16dp</dimen>
    <dimen name="activity_vertical_margin">16dp</dimen>
    <dimen name="nav_header_vertical_spacing">16dp</dimen>
    <dimen name="text_margin">16dp</dimen>
    <dimen name="nav_header_height">160dp</dimen>
    <dimen name="app_bar_height">180dp</dimen>
    <dimen name="appbar_padding_top">8dp</dimen>
    <dimen name="fab_margin">16dp</dimen>
    <dimen name="margin_padding_tiny">4dp</dimen>
    <dimen name="margin_padding_xtiny">6dp</dimen>
    <dimen name="margin_padding_small">9dp</dimen>
    <dimen name="margin_padding_xsmall">12dp</dimen>
    <dimen name="margin_padding_normal">15dp</dimen>
    <dimen name="margin_padding_xnormal">20dp</dimen>
    <dimen name="margin_padding_xxnormal">25dp</dimen>
    <dimen name="margin_padding_large">30dp</dimen>
    <dimen name="margin_padding_xlarge">40dp</dimen>
    <dimen name="margin_padding_xxlarge">50dp</dimen>
    <dimen name="margin_large">70dp</dimen>
    <dimen name="spacing">9dp</dimen>
    <dimen name="dividerHeight">8dp</dimen>
    <dimen name="icon_width_height">30dp</dimen>

</resources>
```

7.c. Strings.xml

```
<resources>
    <string name="yes">It\'s okay</string>
    <string name="no">Take Attendance</string>
    <string name="addNewClass">Add New Class</string>
    <string name="submit">Proceed</string>
    <string name="review_once">Review Once</string>
    <string name="submit_btn">Submit</string>
    <string name="new_subject">New subject</string>
    <string name="add_sub_hint">e.g. Maths</string>
    <string name="subjectImgViewDesctription">This is a subject Icon</string>
    <string name="delete">Delete</string>
    <string name="addSheetMenu">Add New Class</string>
    <string name="sheetAttendanceMenu">Take Attendance</string>
    <string name="sheetViewMenu">View Attendance</string>
    <string name="subjectActivityTitle">List of all your subjects</string>
    <string name="sud_dialog_title">Subject Name</string>
</resources>
```

7.d styles.xml

```
<resources>

    <!-- Base application theme. -->
    <style name="AppTheme" parent="Theme.AppCompat.Light.NoActionBar">
        <!-- Customize your theme here. -->
        <item name="colorPrimary">@color/colorPrimary</item>
        <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
        <item name="colorAccent">@color/colorAccent</item>

    </style>

    <style name="SubjectTheme" parent="Theme.AppCompat.DayNight.DarkActionBar">
        <!-- Customize your theme here. -->
        <item name="colorPrimary">@color/colorPrimary</item>
        <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
        <item name="colorAccent">@color/colorAccent</item>
        <item name="popupMenuStyle">@style/PopupMenu</item>
        <item name="android:itemTextAppearance">@style/itemTextStyle.AppTheme</item>

    </style>

    <style name="PopupMenu" parent="ThemeOverlay.AppCompat.Dark">
        <item name="android:popupBackground">@color/popupBackground</item>
    </style>

    <style name="itemTextStyle.AppTheme"
parent="@android:style/TextAppearance.Widget.IconMenu.Item">
        <item name="android:textColor">@drawable/color_item_popup</item>
    </style>

</resources>
```

11.1.2.8 ‘drawable’ folder contents

8.a. color_item_popup.xml

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:color="@color/colorPrimaryDark" android:state_pressed="true"/>
    <item android:color="@color/background" android:state_focused="true"/>
    <item android:color="@color/colorWhite"/>
</selector>
```

11.1.2.9 ‘anim’ folder contents

9.a. abc_slide_in_bottom.xml

```
<?xml version="1.0" encoding="utf-8"?>
<translate xmlns:android="http://schemas.android.com/apk/res/android"
    android:duration="1000"
    android:fromYDelta="100%"
    android:toYDelta="0%" />
```

9.b. abc_slide_out_bottom.xml

```
<?xml version="1.0" encoding="utf-8"?>
<translate xmlns:android="http://schemas.android.com/apk/res/android"
    android:interpolator="@android:anim/accelerate_interpolator"
    android:fromYDelta="0" android:toYDelta="50%p"
    android:duration="@android:integer/config_mediumAnimTime"/>
```

12. CONCLUSION

The project entitled **“SIKANA”** was completed on time with total satisfaction after testing with possible sample data. The performance was found to be error free and efficient. This is a user friendly application which is very easy to access and understand. Anyone with knowledge of smart phone will find it very easy to use this software and perform the required operations.

In this project, first an attempt has been made to make the digitization concept come true where in which the traditional use of taking attendance is being introduced by a new one.

To fulfill the needs, a detailed study had been conducted to find various requirements for the application. The particular application has been designed in an attractive manner, to make it more user friendly.

This application was developed to help teachers where they have a platform to chat with each other easily and securely and also the online attendance system is a new advancement which makes their task easier. Finally the application was tested with real data and everything worked successfully.

13. FUTURE SCOPE

As the app is companion to a faculty, in future there are certain functionalities that will be added. In the chat section the functionality of video chat will be integrated. The capability to share location will also be added. And automatic reply mechanism is to be added where in which if the faculty's status is busy then automatic reply will be sent to the sender. This reply mechanism will use insight extraction method and reply accordingly.

The attendance taking methond i.e. clicking buttons will be changed in near future and attedance will be taken using face recognition. The teacher just has to take a picture of the class and this picture will be sent to face recognition ai which in turn will list all the names of the students thus making taking attendance a lot easier. As currently there is no option to edit the attendance sheet from mobile phone that functionality will be added later. Also the functionality of local storage will be developed along with automatic online sync i.e. when there is no internet connection the teachers can take attendance offline where data will be stored locally and as soon as there is internet connection the data will be sync automatically in google drive.

14. BIBLIOGRAPHY

14.1 WEBSITES

[1] https://en.wikipedia.org/wiki/Integrated_development_environment

[2] https://en.wikipedia.org/wiki/Google

[3] https://en.wikipedia.org/wiki/Operating_system

[4] https://en.wikipedia.org/wiki/JetBrains

[5] https://en.wikipedia.org/wiki/IntelliJ_IDEA

[6] https://en.wikipedia.org/wiki/Android_software_development