**“MINGO”**

A Friend Affinity Finder

A PROJECT REPORT

BY

**“TRICKY TRACKERS”**

*KISHORE.S*

*KAYALVIZHI.S*

*S.CHRISTIKA*

*J.JANAKIRAM*

TABLE OF CONTENTS

* ABSTRACT
* INTRODUCTION
* FRIEND AFFINITY
* SPECIFICATIONS
* LANGUAGE & TOOLS
* SOURCE CODE
* FLOW CHART
* SURVEY REPORT
* CONCLUSION

**ABSTRACT:**

The most important aspect of our project is to set S.M.A.R.T ( Specific , Measurable, Achievable, Realistic and Timely) goals. Our aim of the “MINGO APP” is to create friend affinity between people in which it is not necessary to know the identity of a particular people.

**INTRODUCTION:**

In this current generation, people started to chase an easy way to connect with others. Even an ideal person could share their thoughts throughout the word.

***“IF YOU FEEL SAD AND BAD ………***

***USE CONNECT TO CONNECT”***

**FRIEND AFFINITY:**

Social media are interactive web internet based applications. Social media plays a major role in friend affinity. The active and animated engagement from all strata of society which has led to explosion in information sharing.

**“I NEED A FRIEND ………ANY CONNECTORS ”**

**SPECIFICATIONS:**

Reduces phone storage and prevent from heating of the mobiles. It is used to connect with stranger in order to know more about them. This app is used to chat, upload status , sharing thoughts , describe yourself in the bio and stay connected with your loved ones.

Moreover the speciality of our **mingo** app is to connect the people who passed by you. It provides a tremendous storage capacity with the help of fire base storage. It also concentrated on education by making the user to use **“eChat”**, which helps the to create their own platform to communicate with their students, which breaks the bridge of fear between the students on teacher. They can also clarify their queries by using our app.

And added that we focus on one of the basic need of the program i.e.,”FOOD!!!!!”. You can make possible to deliver of your food on your door step using our **“YUMMY icon”**.

There is another interesting part our app is **“CHAT ROBOT”** which is called as **“cBot”**. It allow you to communicate with it. With the help of our latest AI technology cBot response to your feelings, thoughts, ideas in a useful manner.

The entertainment part of our app is attained by clicking an **“eTainment”** icon for booking “MOVIE tickets”.

* PRIVACY: The messages which the user sends will be an “End to End encrypted” messages.
* SECURITY: This app is built with the “THREE BACKDOOR FIREWALL”

**LANGUAGE AND TOOLS:**

* Java version9
* Android App Studio version 3.4
* fire base

**SOURCE CODE:**

**MAIN ACTIVITY:**

package com.example.codingcafe.socialnetwork;

import android.content.Context;

import android.content.Intent;

import android.os.RecoverySystem;

import android.support.annotation.NonNull;

import android.support.design.widget.NavigationView;

import android.support.v4.widget.DrawerLayout;

import android.support.v7.app.ActionBarDrawerToggle;

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.MenuItem;

import android.view.View;

import android.widget.ImageButton;

import android.widget.ImageView;

import android.widget.TextView;

import android.widget.Toast;

import com.firebase.ui.database.FirebaseRecyclerAdapter;

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.squareup.picasso.Picasso;

import de.hdodenhof.circleimageview.CircleImageView;

public class MainActivity extends AppCompatActivity

{

private NavigationView navigationView;

private DrawerLayout drawerLayout;

private ActionBarDrawerToggle actionBarDrawerToggle;

private RecyclerView postList;

private Toolbar mToolbar;

private CircleImageView NavProfileImage;

private TextView NavProfileUserName;

private ImageButton AddNewPostButton;

private FirebaseAuth mAuth;

private DatabaseReference UsersRef, PostsRef;

String currentUserID;

@Override

protected void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

mAuth = FirebaseAuth.getInstance();

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

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

PostsRef = FirebaseDatabase.getInstance().getReference().child("Posts");

mToolbar = (Toolbar) findViewById(R.id.main\_page\_toolbar);

setSupportActionBar(mToolbar);

getSupportActionBar().setTitle("Home");

AddNewPostButton = (ImageButton) findViewById(R.id.add\_new\_post\_button);

drawerLayout = (DrawerLayout) findViewById(R.id.drawable\_layout);

actionBarDrawerToggle = new ActionBarDrawerToggle(MainActivity.this, drawerLayout, R.string.drawer\_open, R.string.drawer\_close);

drawerLayout.addDrawerListener(actionBarDrawerToggle);

actionBarDrawerToggle.syncState();

getSupportActionBar().setDisplayHomeAsUpEnabled(true);

navigationView = (NavigationView) findViewById(R.id.navigation\_view);

postList = (RecyclerView) findViewById(R.id.all\_users\_post\_list);

postList.setHasFixedSize(true);

LinearLayoutManager linearLayoutManager = new LinearLayoutManager(this);

linearLayoutManager.setReverseLayout(true);

linearLayoutManager.setStackFromEnd(true);

postList.setLayoutManager(linearLayoutManager);

View navView = navigationView.inflateHeaderView(R.layout.navigation\_header);

NavProfileImage = (CircleImageView) navView.findViewById(R.id.nav\_profile\_image);

NavProfileUserName = (TextView) navView.findViewById(R.id.nav\_user\_full\_name);

UsersRef.child(currentUserID).addValueEventListener(new ValueEventListener() {

@Override

public void onDataChange(DataSnapshot dataSnapshot)

{

if(dataSnapshot.exists())

{

if(dataSnapshot.hasChild("fullname"))

{

String fullname = dataSnapshot.child("fullname").getValue().toString();

NavProfileUserName.setText(fullname);

}

if(dataSnapshot.hasChild("profileimage"))

{

String image = dataSnapshot.child("profileimage").getValue().toString(); Picasso.with(MainActivity.this).load(image).placeholder(R.drawable.profile).into(NavProfileImage);

}

else

{

Toast.makeText(MainActivity.this, "Profile name do not exists...", Toast.LENGTH\_SHORT).show();

}

}

}

@Override

public void onCancelled(DatabaseError databaseError) {

}

});

navigationView.setNavigationItemSelectedListener(new NavigationView.OnNavigationItemSelectedListener() {

@Override

public boolean onNavigationItemSelected(@NonNull MenuItem item)

{

UserMenuSelector(item);

return false;

}

});

AddNewPostButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v)

{

SendUserToPostActivity();

}

});

DisplayAllUsersPosts();

}

private void DisplayAllUsersPosts()

{

FirebaseRecyclerAdapter<Posts, PostsViewHolder> firebaseRecyclerAdapter =

new FirebaseRecyclerAdapter<Posts, PostsViewHolder>

(

Posts.class,

R.layout.all\_posts\_layout,

PostsViewHolder.class,

PostsRef

)

{

@Override

protected void populateViewHolder(PostsViewHolder viewHolder, Posts model, int position)

{

viewHolder.setFullname(model.getFullname());

viewHolder.setTime(model.getTime());

viewHolder.setDate(model.getDate());

viewHolder.setDescription(model.getDescription());

viewHolder.setProfileimage(getApplicationContext(), model.getProfileimage());

viewHolder.setPostimage(getApplicationContext(), model.getPostimage());

}

};

postList.setAdapter(firebaseRecyclerAdapter);

}

public static class PostsViewHolder extends RecyclerView.ViewHolder

{

View mView;

public PostsViewHolder(View itemView)

{

super(itemView);

mView = itemView;

}

public void setFullname(String fullname)

{

TextView username = (TextView) mView.findViewById(R.id.post\_user\_name);

username.setText(fullname);

}

public void setProfileimage(Context ctx, String profileimage)

{

CircleImageView image = (CircleImageView) mView.findViewById(R.id.post\_profile\_image);

Picasso.with(ctx).load(profileimage).into(image);

}

public void setTime(String time)

{

TextView PostTime = (TextView) mView.findViewById(R.id.post\_time);

PostTime.setText(" " + time);

}

public void setDate(String date)

{

TextView PostDate = (TextView) mView.findViewById(R.id.post\_date);

PostDate.setText(" " + date);

}

public void setDescription(String description)

{

TextView PostDescription = (TextView) mView.findViewById(R.id.post\_description);

PostDescription.setText(description);

}

public void setPostimage(Context ctx1, String postimage)

{

ImageView PostImage = (ImageView) mView.findViewById(R.id.post\_image);

Picasso.with(ctx1).load(postimage).into(PostImage);

}

}

private void SendUserToPostActivity()

{

Intent addNewPostIntent = new Intent(MainActivity.this, PostActivity.class);

startActivity(addNewPostIntent);

}

@Override

protected void onStart()

{

super.onStart();

FirebaseUser currentUser = mAuth.getCurrentUser();

if(currentUser == null)

{

SendUserToLoginActivity();

}

else

{

CheckUserExistence();

}

}

private void CheckUserExistence()

{

final String current\_user\_id = mAuth.getCurrentUser().getUid();

UsersRef.addValueEventListener(new ValueEventListener() {

@Override

public void onDataChange(DataSnapshot dataSnapshot)

{

if(!dataSnapshot.hasChild(current\_user\_id))

{

SendUserToSetupActivity();

}

}

@Override

public void onCancelled(DatabaseError databaseError) {

}

});

}

private void SendUserToSetupActivity()

{

Intent setupIntent = new Intent(MainActivity.this, SetupActivity.class);

setupIntent.addFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK | Intent.FLAG\_ACTIVITY\_CLEAR\_TASK);

startActivity(setupIntent);

finish();

}

private void SendUserToLoginActivity()

{

Intent loginIntent = new Intent(MainActivity.this, LoginActivity.class);

loginIntent.addFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK | Intent.FLAG\_ACTIVITY\_CLEAR\_TASK);

startActivity(loginIntent);

finish();

}

@Override

public boolean onOptionsItemSelected(MenuItem item)

{

if(actionBarDrawerToggle.onOptionsItemSelected(item))

{

return true;

}

return super.onOptionsItemSelected(item);

}

private void UserMenuSelector(MenuItem item)

{

switch (item.getItemId())

{

case R.id.nav\_post:

SendUserToPostActivity();

break;

case R.id.nav\_profile:

Toast.makeText(this, "Profile", Toast.LENGTH\_SHORT).show();

break;

case R.id.nav\_home:

Toast.makeText(this, "Home", Toast.LENGTH\_SHORT).show();

break;

case R.id.nav\_friends:

Toast.makeText(this, "Friend List", Toast.LENGTH\_SHORT).show();

break;

case R.id.nav\_find\_friends:

Toast.makeText(this, "Find Friends", Toast.LENGTH\_SHORT).show();

break;

case R.id.nav\_messages:

Toast.makeText(this, "Messages", Toast.LENGTH\_SHORT).show();

break;

case R.id.nav\_settings:

Toast.makeText(this, "Settings", Toast.LENGTH\_SHORT).show();

break;

case R.id.nav\_Logout:

mAuth.signOut();

SendUserToLoginActivity();

break;

}

**BUILD GRADLE(MODULE APP):**

apply plugin: 'com.android.application'

android {

compileSdkVersion 27

defaultConfig {

applicationId "com.example.codingcafe.socialnetwork"

minSdkVersion 19

targetSdkVersion 27

versionCode 1

versionName "1.0"

testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"

multiDexEnabled true

}

buildTypes {

release {

minifyEnabled false

proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'

}

}

}

dependencies {

implementation fileTree(dir: 'libs', include: ['\*.jar'])

implementation 'com.android.support:appcompat-v7:27.1.0'

implementation 'com.android.support:design:27.1.0'

implementation 'com.android.support.constraint:constraint-layout:1.0.2'

implementation 'com.google.firebase:firebase-auth:11.0.2'

implementation 'com.google.android.gms:play-services-auth:11.0.2'

implementation 'com.google.firebase:firebase-database:11.0.2'

implementation 'com.google.firebase:firebase-storage:11.0.2'

implementation 'de.hdodenhof:circleimageview:2.2.0'

implementation 'com.theartofdev.edmodo:android-image-cropper:2.6.+'

implementation 'com.squareup.picasso:picasso:2.5.2'

implementation 'com.firebaseui:firebase-ui-database:2.1.1'

testImplementation 'junit:junit:4.12'

androidTestImplementation 'com.android.support.test:runner:1.0.1'

androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.1'

}

apply plugin: 'com.google.gms.google-services'

**POST ACTIVITY:**

package com.example.codingcafe.socialnetwork;

import android.app.ProgressDialog;

import android.content.Intent;

import android.net.Uri;

import android.support.annotation.NonNull;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.support.v7.widget.Toolbar;

import android.text.TextUtils;

import android.view.MenuItem;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.ImageButton;

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.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 java.text.SimpleDateFormat;

import java.util.Calendar;

import java.util.HashMap;

public class PostActivity extends AppCompatActivity

{

private Toolbar mToolbar;

private ProgressDialog loadingBar;

private ImageButton SelectPostImage;

private Button UpdatePostButton;

private EditText PostDescription;

private static final int Gallery\_Pick = 1;

private Uri ImageUri;

private String Description;

private StorageReference PostsImagesRefrence;

private DatabaseReference UsersRef, PostsRef;

private FirebaseAuth mAuth;

private String saveCurrentDate, saveCurrentTime, postRandomName, downloadUrl, current\_user\_id;

@Override

protected void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_post);

mAuth = FirebaseAuth.getInstance();

current\_user\_id = mAuth.getCurrentUser().getUid();

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

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

PostsRef = FirebaseDatabase.getInstance().getReference().child("Posts");

SelectPostImage = (ImageButton) findViewById(R.id.select\_post\_image);

UpdatePostButton = (Button) findViewById(R.id.update\_post\_button);

PostDescription =(EditText) findViewById(R.id.post\_description);

loadingBar = new ProgressDialog(this);

mToolbar = (Toolbar) findViewById(R.id.update\_post\_page\_toolbar);

setSupportActionBar(mToolbar);

getSupportActionBar().setDisplayHomeAsUpEnabled(true);

getSupportActionBar().setDisplayShowHomeEnabled(true);

getSupportActionBar().setTitle("Update Post");

SelectPostImage.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v)

- {

OpenGallery();

}

});

UpdatePostButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v)

{

ValidatePostInfo();

}

});

}

private void ValidatePostInfo()

{

Description = PostDescription.getText().toString();

if(ImageUri == null)

{

Toast.makeText(this, "Please select post image...", Toast.LENGTH\_SHORT).show();

}

else if(TextUtils.isEmpty(Description))

{

Toast.makeText(this, "Please say something about your image...", Toast.LENGTH\_SHORT).show();

}

else

{

loadingBar.setTitle("Add New Post");

loadingBar.setMessage("Please wait, while we are updating your new post...");

loadingBar.show();

loadingBar.setCanceledOnTouchOutside(true);

StoringImageToFirebaseStorage();

}

}

private void StoringImageToFirebaseStorage()

{

Calendar calFordDate = Calendar.getInstance();

SimpleDateFormat currentDate = new SimpleDateFormat("dd-MMMM-yyyy");

saveCurrentDate = currentDate.format(calFordDate.getTime());

Calendar calFordTime = Calendar.getInstance();

SimpleDateFormat currentTime = new SimpleDateFormat("HH:mm");

saveCurrentTime = currentTime.format(calFordDate.getTime());

postRandomName = saveCurrentDate + saveCurrentTime;

StorageReference filePath = PostsImagesRefrence.child("Post Images").child(ImageUri.getLastPathSegment() + postRandomName + ".jpg");

filePath.putFile(ImageUri).addOnCompleteListener(new OnCompleteListener<UploadTask.TaskSnapshot>() {

@Override

public void onComplete(@NonNull Task<UploadTask.TaskSnapshot> task)

{

if(task.isSuccessful())

{

downloadUrl = task.getResult().getDownloadUrl().toString();

Toast.makeText(PostActivity.this, "image uploaded successfully to Storage...", Toast.LENGTH\_SHORT).show();

SavingPostInformationToDatabase();

}

else

{

String message = task.getException().getMessage();

Toast.makeText(PostActivity.this, "Error occured: " + message, Toast.LENGTH\_SHORT).show();

}

}

});

}

private void SavingPostInformationToDatabase()

{

UsersRef.child(current\_user\_id).addValueEventListener(new ValueEventListener() {

@Override

public void onDataChange(DataSnapshot dataSnapshot)

{

if(dataSnapshot.exists())

{

String userFullName = dataSnapshot.child("fullname").getValue().toString();

String userProfileImage = dataSnapshot.child("profileimage").getValue().toString();

HashMap postsMap = new HashMap();

postsMap.put("uid", current\_user\_id);

postsMap.put("date", saveCurrentDate);

postsMap.put("time", saveCurrentTime);

postsMap.put("description", Description);

postsMap.put("postimage", downloadUrl);

postsMap.put("profileimage", userProfileImage);

postsMap.put("fullname", userFullName);

PostsRef.child(current\_user\_id + postRandomName).updateChildren(postsMap)

.addOnCompleteListener(new OnCompleteListener() {

@Override

public void onComplete(@NonNull Task task)

{

if(task.isSuccessful())

{

SendUserToMainActivity();

Toast.makeText(PostActivity.this, "New Post is updated successfully.", Toast.LENGTH\_SHORT).show();

loadingBar.dismiss();

}

else

{

Toast.makeText(PostActivity.this, "Error Occured while updating your post.", Toast.LENGTH\_SHORT).show();

loadingBar.dismiss();

}

}

});

}

}

@Override

public void onCancelled(DatabaseError databaseError) {

}

});

}

private void OpenGallery()

{

Intent galleryIntent = new Intent();

galleryIntent.setAction(Intent.ACTION\_GET\_CONTENT);

galleryIntent.setType("image/\*");

startActivityForResult(galleryIntent, Gallery\_Pick);

}

@Override

protected void onActivityResult(int requestCode, int resultCode, Intent data)

{

super.onActivityResult(requestCode, resultCode, data);

if(requestCode==Gallery\_Pick && resultCode==RESULT\_OK && data!=null)

{

ImageUri = data.getData();

SelectPostImage.setImageURI(ImageUri);

}

}

@Override

public boolean onOptionsItemSelected(MenuItem item)

{

int id = item.getItemId();

if(id == android.R.id.home)

{

SendUserToMainActivity();

}

return super.onOptionsItemSelected(item);

}

private void SendUserToMainActivity()

{

Intent mainIntent = new Intent(PostActivity.this, MainActivity.class);

startActivity(mainIntent);

}

}

**POSTS:**

package com.example.codingcafe.socialnetwork;

/\*\*

\* Created by Coding Cafe on 3/4/2018.

\*/

public class Posts

{

public String uid, time, date, postimage, description, profileimage, fullname;

public Posts()

{

}

public Posts(String uid, String time, String date, String postimage, String description, String profileimage, String fullname) {

this.uid = uid;

this.time = time;

this.date = date;

this.postimage = postimage;

this.description = description;

this.profileimage = profileimage;

this.fullname = fullname;

}

public String getUid() {

return uid;

}

public void setUid(String uid) {

this.uid = uid;

}

public String getTime() {

return time;

}

public void setTime(String time) {

this.time = time;

}

public String getDate() {

return date;

}

public void setDate(String date) {

this.date = date;

}

public String getPostimage() {

return postimage;

}

public void setPostimage(String postimage) {

this.postimage = postimage;

}

public String getDescription() {

return description;

}

public void setDescription(String description) {

this.description = description;

}

public String getProfileimage() {

return profileimage;

}

public void setProfileimage(String profileimage) {

this.profileimage = profileimage;

}

public String getFullname() {

return fullname;

}

public void setFullname(String fullname) {

this.fullname = fullname;

}

}

**SETUP:**

package com.example.codingcafe.socialnetwork;

import android.app.ProgressDialog;

import android.content.Intent;

import android.net.Uri;

import android.os.storage.StorageManager;

import android.support.annotation.NonNull;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.text.TextUtils;

import android.transition.CircularPropagation;

import android.view.View;

import android.widget.Button;

import android.widget.CursorAdapter;

import android.widget.EditText;

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.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.Picasso;

import com.theartofdev.edmodo.cropper.CropImage;

import com.theartofdev.edmodo.cropper.CropImageView;

import java.util.HashMap;

import de.hdodenhof.circleimageview.CircleImageView;

public class SetupActivity extends AppCompatActivity

{

private EditText UserName, FullName, CountryName;

private Button SaveInformationbuttion;

private CircleImageView ProfileImage;

private ProgressDialog loadingBar;

private FirebaseAuth mAuth;

private DatabaseReference UsersRef;

private StorageReference UserProfileImageRef;

String currentUserID;

final static int Gallery\_Pick = 1;

@Override

protected void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_setup);

mAuth = FirebaseAuth.getInstance();

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

UsersRef = FirebaseDatabase.getInstance().getReference().child("Users").child(currentUserID);

UserProfileImageRef = FirebaseStorage.getInstance().getReference().child("Profile Images");

UserName = (EditText) findViewById(R.id.setup\_username);

FullName = (EditText) findViewById(R.id.setup\_full\_name);

CountryName = (EditText) findViewById(R.id.setup\_country\_name);

SaveInformationbuttion = (Button) findViewById(R.id.setup\_information\_button);

ProfileImage = (CircleImageView) findViewById(R.id.setup\_profile\_image);

loadingBar = new ProgressDialog(this);

SaveInformationbuttion.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view)

{

SaveAccountSetupInformation();

}

});

ProfileImage.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view)

{

Intent galleryIntent = new Intent();

galleryIntent.setAction(Intent.ACTION\_GET\_CONTENT);

galleryIntent.setType("image/\*");

startActivityForResult(galleryIntent, Gallery\_Pick);

}

});

UsersRef.addValueEventListener(new ValueEventListener() {

@Override

public void onDataChange(DataSnapshot dataSnapshot)

{

if(dataSnapshot.exists())

{

if (dataSnapshot.hasChild("profileimage"))

{

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

Picasso.with(SetupActivity.this).load(image).placeholder(R.drawable.profile).into(ProfileImage);

}

else

{

Toast.makeText(SetupActivity.this, "Please select profile image first.", Toast.LENGTH\_SHORT).show();

}

}

}

@Override

public void onCancelled(DatabaseError databaseError) {

}

});

}

@Override

protected void onActivityResult(int requestCode, int resultCode, Intent data)

{

super.onActivityResult(requestCode, resultCode, data);

if(requestCode==Gallery\_Pick && resultCode==RESULT\_OK && data!=null)

{

Uri ImageUri = data.getData();

CropImage.activity()

.setGuidelines(CropImageView.Guidelines.ON)

.setAspectRatio(1, 1)

.start(this);

}

if(requestCode==CropImage.CROP\_IMAGE\_ACTIVITY\_REQUEST\_CODE)

{

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

if(resultCode == RESULT\_OK)

{

loadingBar.setTitle("Profile Image");

loadingBar.setMessage("Please wait, while we updating your profile image...");

loadingBar.show();

loadingBar.setCanceledOnTouchOutside(true);

Uri resultUri = result.getUri();

StorageReference filePath = UserProfileImageRef.child(currentUserID + ".jpg");

filePath.putFile(resultUri).addOnCompleteListener(new OnCompleteListener<UploadTask.TaskSnapshot>() {

@Override

public void onComplete(@NonNull final Task<UploadTask.TaskSnapshot> task)

{

if(task.isSuccessful())

{

Toast.makeText(SetupActivity.this, "Profile Image stored successfully to Firebase storage...", Toast.LENGTH\_SHORT).show();

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

UsersRef.child("profileimage").setValue(downloadUrl)

.addOnCompleteListener(new OnCompleteListener<Void>() {

@Override

public void onComplete(@NonNull Task<Void> task)

{

if(task.isSuccessful())

{

Intent selfIntent = new Intent(SetupActivity.this, SetupActivity.class);

startActivity(selfIntent);

Toast.makeText(SetupActivity.this, "Profile Image stored to Firebase Database Successfully...", Toast.LENGTH\_SHORT).show();

loadingBar.dismiss();

}

else

{

String message = task.getException().getMessage();

Toast.makeText(SetupActivity.this, "Error Occured: " + message, Toast.LENGTH\_SHORT).show();

loadingBar.dismiss();

}

}

});

}

}

});

}

else

{

Toast.makeText(this, "Error Occured: Image can not be cropped. Try Again.", Toast.LENGTH\_SHORT).show();

loadingBar.dismiss();

}

}

}

private void SaveAccountSetupInformation()

{

String username = UserName.getText().toString();

String fullname = FullName.getText().toString();

String country = CountryName.getText().toString();

if(TextUtils.isEmpty(username))

{

Toast.makeText(this, "Please write your username...", Toast.LENGTH\_SHORT).show();

}

if(TextUtils.isEmpty(fullname))

{

Toast.makeText(this, "Please write your full name...", Toast.LENGTH\_SHORT).show();

}

if(TextUtils.isEmpty(country))

{

Toast.makeText(this, "Please write your country...", Toast.LENGTH\_SHORT).show();

}

else

{

loadingBar.setTitle("Saving Information");

loadingBar.setMessage("Please wait, while we are creating your new Account...");

loadingBar.show();

loadingBar.setCanceledOnTouchOutside(true);

HashMap userMap = new HashMap();

userMap.put("username", username);

userMap.put("fullname", fullname);

userMap.put("country", country);

userMap.put("status", "Hey there, i am using Poster Social Network, developed by Coding Cafe.");

userMap.put("gender", "none");

userMap.put("dob", "none");

userMap.put("relationshipstatus", "none");

UsersRef.updateChildren(userMap).addOnCompleteListener(new OnCompleteListener() {

@Override

public void onComplete(@NonNull Task task)

{

if(task.isSuccessful())

{

SendUserToMainActivity();

Toast.makeText(SetupActivity.this, "your Account is created Successfully.", Toast.LENGTH\_LONG).show();

loadingBar.dismiss();

}

else

{

String message = task.getException().getMessage();

Toast.makeText(SetupActivity.this, "Error Occured: " + message, Toast.LENGTH\_SHORT).show();

loadingBar.dismiss();

}

}

});

}

}

private void SendUserToMainActivity()

{

Intent mainIntent = new Intent(SetupActivity.this, MainActivity.class);

mainIntent.addFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK | Intent.FLAG\_ACTIVITY\_CLEAR\_TASK);

startActivity(mainIntent);

finish();

}

}

**LOGIN:**

package com.example.codingcafe.socialnetwork;

import android.app.ProgressDialog;

import android.content.Intent;

import android.support.annotation.NonNull;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.text.TextUtils;

import android.util.Log;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.ImageView;

import android.widget.TextView;

import android.widget.Toast;

import com.google.android.gms.auth.api.Auth;

import com.google.android.gms.auth.api.signin.GoogleSignInAccount;

import com.google.android.gms.auth.api.signin.GoogleSignInOptions;

import com.google.android.gms.auth.api.signin.GoogleSignInResult;

import com.google.android.gms.common.ConnectionResult;

import com.google.android.gms.common.api.ApiException;

import com.google.android.gms.common.api.GoogleApiClient;

import com.google.android.gms.tasks.OnCompleteListener;

import com.google.android.gms.tasks.Task;

import com.google.firebase.auth.AuthCredential;

import com.google.firebase.auth.AuthResult;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.auth.FirebaseUser;

import com.google.firebase.auth.GoogleAuthProvider;

public class LoginActivity extends AppCompatActivity

{

private Button LoginButton;

private ImageView googleSignInButton;

private EditText UserEmail, UserPassword;

private TextView NeedNewAccountLink;

private ProgressDialog loadingBar;

private FirebaseAuth mAuth;

private static final int RC\_SIGN\_IN = 1;

private GoogleApiClient mGoogleSignInClient;

private static final String TAG = "LoginActivity";

@Override

protected void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_login);

mAuth = FirebaseAuth.getInstance();

NeedNewAccountLink = (TextView) findViewById(R.id.register\_account\_link);

UserEmail = (EditText) findViewById(R.id.login\_email);

UserPassword = (EditText) findViewById(R.id.login\_password);

LoginButton = (Button) findViewById(R.id.login\_button);

googleSignInButton = (ImageView) findViewById(R.id.google\_signin\_button);

loadingBar = new ProgressDialog(this);

NeedNewAccountLink.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view)

{

SendUserToRegisterActivity();

}

});

LoginButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view)

{

AllowingUserToLogin();

}

});

// Configure Google Sign In

GoogleSignInOptions gso = new GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT\_SIGN\_IN)

.requestIdToken(getString(R.string.default\_web\_client\_id))

.requestEmail()

.build();

mGoogleSignInClient = new GoogleApiClient.Builder(this)

.enableAutoManage(this, new GoogleApiClient.OnConnectionFailedListener() {

@Override

public void onConnectionFailed(@NonNull ConnectionResult connectionResult)

{

Toast.makeText(LoginActivity.this, "Connection to Google Sign in failed...", Toast.LENGTH\_SHORT).show();

}

})

.addApi(Auth.GOOGLE\_SIGN\_IN\_API, gso)

.build();

googleSignInButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v)

{

signIn();

}

});

}

private void signIn()

{

Intent signInIntent = Auth.GoogleSignInApi.getSignInIntent(mGoogleSignInClient);

startActivityForResult(signInIntent, RC\_SIGN\_IN);

}

@Override

public void onActivityResult(int requestCode, int resultCode, Intent data)

{

super.onActivityResult(requestCode, resultCode, data);

if (requestCode == RC\_SIGN\_IN)

{

loadingBar.setTitle("google Sign In");

loadingBar.setMessage("Please wait, while we are allowing you to login using your Google Account...");

loadingBar.setCanceledOnTouchOutside(true);

loadingBar.show();

GoogleSignInResult result = Auth.GoogleSignInApi.getSignInResultFromIntent(data);

if (result.isSuccess())

{

GoogleSignInAccount account = result.getSignInAccount();

firebaseAuthWithGoogle(account);

Toast.makeText(this, "Please wait, while we are getting your auth result...", Toast.LENGTH\_SHORT).show();

}

else

{

Toast.makeText(this, "Can't get Auth result.", Toast.LENGTH\_SHORT).show();

loadingBar.dismiss();

}

}

}

private void firebaseAuthWithGoogle(GoogleSignInAccount acct) {

Log.d(TAG, "firebaseAuthWithGoogle:" + acct.getId());

AuthCredential credential = GoogleAuthProvider.getCredential(acct.getIdToken(), null);

mAuth.signInWithCredential(credential)

.addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {

@Override

public void onComplete(@NonNull Task<AuthResult> task) {

if (task.isSuccessful())

{

Log.d(TAG, "signInWithCredential:success");

SendUserToMainActivity();

loadingBar.dismiss();

}

else

{

Log.w(TAG, "signInWithCredential:failure", task.getException());

String message = task.getException().toString();

SendUserToLoginActivity();

Toast.makeText(LoginActivity.this, "Not Authenticated : " + message, Toast.LENGTH\_SHORT).show();

loadingBar.dismiss();

}

}

});

}

@Override

protected void onStart()

{

super.onStart();

FirebaseUser currentUser = mAuth.getCurrentUser();

if(currentUser != null)

{

SendUserToMainActivity();

}

}

private void AllowingUserToLogin()

{

String email = UserEmail.getText().toString();

String password = UserPassword.getText().toString();

if(TextUtils.isEmpty(email))

{

Toast.makeText(this, "Please write your email...", Toast.LENGTH\_SHORT).show();

}

else if(TextUtils.isEmpty(password))

{

Toast.makeText(this, "Please write your password...", Toast.LENGTH\_SHORT).show();

}

else

{

loadingBar.setTitle("Login");

loadingBar.setMessage("Please wait, while we are allowing you to login into your Account...");

loadingBar.setCanceledOnTouchOutside(true);

loadingBar.show();

mAuth.signInWithEmailAndPassword(email, password)

.addOnCompleteListener(new OnCompleteListener<AuthResult>() {

@Override

public void onComplete(@NonNull Task<AuthResult> task)

{

if(task.isSuccessful())

{

SendUserToMainActivity();

Toast.makeText(LoginActivity.this, "you are Logged In successfully.", Toast.LENGTH\_SHORT).show();

loadingBar.dismiss();

}

else

{

String message = task.getException().getMessage();

Toast.makeText(LoginActivity.this, "Error occured: " + message, Toast.LENGTH\_SHORT).show();

loadingBar.dismiss();

}

}

});

}

}

private void SendUserToMainActivity()

{

Intent mainIntent = new Intent(LoginActivity.this, MainActivity.class);

mainIntent.addFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK | Intent.FLAG\_ACTIVITY\_CLEAR\_TASK);

startActivity(mainIntent);

finish();

}

private void SendUserToLoginActivity()

{

Intent mainIntent = new Intent(LoginActivity.this, LoginActivity.class);

mainIntent.addFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK | Intent.FLAG\_ACTIVITY\_CLEAR\_TASK);

startActivity(mainIntent);

finish();

}

private void SendUserToRegisterActivity()

{

Intent registerIntent = new Intent(LoginActivity.this, RegisterActivity.class);

startActivity(registerIntent);

}

}

**REGISTER:**

package com.example.codingcafe.socialnetwork;

import android.app.ProgressDialog;

import android.content.Intent;

import android.support.annotation.NonNull;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.text.TextUtils;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

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;

public class RegisterActivity extends AppCompatActivity

{

private EditText UserEmail, UserPassword, UserConfirmPassword;

private Button CreateAccountButton;

private ProgressDialog loadingBar;

private FirebaseAuth mAuth;

@Override

protected void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_register);

mAuth = FirebaseAuth.getInstance();

UserEmail = (EditText) findViewById(R.id.register\_email);

UserPassword = (EditText) findViewById(R.id.register\_password);

UserConfirmPassword = (EditText) findViewById(R.id.register\_confirm\_password);

CreateAccountButton = (Button) findViewById(R.id.register\_create\_account);

loadingBar = new ProgressDialog(this);

CreateAccountButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view)

{

CreateNewAccount();

}

});

}

@Override

protected void onStart()

{

super.onStart();

FirebaseUser currentUser = mAuth.getCurrentUser();

if(currentUser != null)

{

SendUserToMainActivity();

}

}

private void SendUserToMainActivity()

{

Intent mainIntent = new Intent(RegisterActivity.this, MainActivity.class);

mainIntent.addFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK | Intent.FLAG\_ACTIVITY\_CLEAR\_TASK);

startActivity(mainIntent);

finish();

}

private void CreateNewAccount()

{

String email = UserEmail.getText().toString();

String password = UserPassword.getText().toString();

String confirmPassword = UserConfirmPassword.getText().toString();

if(TextUtils.isEmpty(email))

{

Toast.makeText(this, "Please write your email...", Toast.LENGTH\_SHORT).show();

}

else if(TextUtils.isEmpty(password))

{

Toast.makeText(this, "Please write your password...", Toast.LENGTH\_SHORT).show();

}

else if(TextUtils.isEmpty(confirmPassword))

{

Toast.makeText(this, "Please confirm your password...", Toast.LENGTH\_SHORT).show();

}

else if(!password.equals(confirmPassword))

{

Toast.makeText(this, "your password do not match with your confirm password...", Toast.LENGTH\_SHORT).show();

}

else

{

loadingBar.setTitle("Creating New Account");

loadingBar.setMessage("Please wait, while we are creating your new Account...");

loadingBar.show();

loadingBar.setCanceledOnTouchOutside(true);

mAuth.createUserWithEmailAndPassword(email, password)

.addOnCompleteListener(new OnCompleteListener<AuthResult>() {

@Override

public void onComplete(@NonNull Task<AuthResult> task)

{

if(task.isSuccessful())

{

SendUserToSetupActivity();

Toast.makeText(RegisterActivity.this, "you are authenticated successfully...", Toast.LENGTH\_SHORT).show();

loadingBar.dismiss();

}

else

{

String message = task.getException().getMessage();

Toast.makeText(RegisterActivity.this, "Error Occured: " + message, Toast.LENGTH\_SHORT).show();

loadingBar.dismiss();

}

}

});

}

}

private void SendUserToSetupActivity()

{

Intent setupIntent = new Intent(RegisterActivity.this, SetupActivity.class);

setupIntent.addFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK | Intent.FLAG\_ACTIVITY\_CLEAR\_TASK);

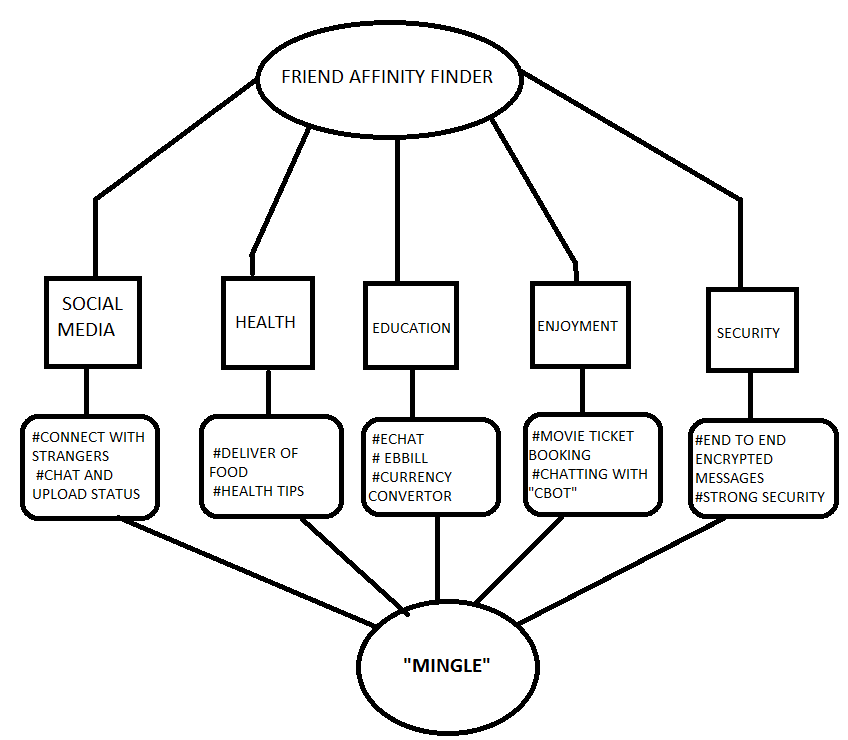
startActivity(setupIntent);

finish();

}

}

**FLOW CHART:**

****

**SURVEY REPORT:**

**CONCLUSION:**

This app fulfils all our needs and worries. This is not an end of our application. We used to provide a monthly updates with new features. Thus we conclude that our app going to make a revolution in 2020.