*package* com.csee5590.lab3;  
  
*import* android.content.Intent;  
*import* android.os.Bundle;  
*import* android.support.annotation.NonNull;  
*import* android.support.v7.app.AppCompatActivity;  
*import* android.support.v7.widget.Toolbar;  
*import* android.util.Log;  
*import* android.util.Patterns;  
*import* android.view.MotionEvent;  
*import* android.view.View;  
*import* android.view.*Menu*;  
*import* android.view.*MenuItem*;  
*import* android.view.animation.Animation;  
*import* android.view.animation.AnimationUtils;  
*import* android.widget.Button;  
*import* android.widget.EditText;  
*import* android.widget.TextView;  
  
*import* com.google.android.gms.tasks.*OnCompleteListener*;  
*import* com.google.android.gms.tasks.*OnFailureListener*;  
*import* com.google.android.gms.tasks.*OnSuccessListener*;  
*import* com.google.android.gms.tasks.Task;  
*import* com.google.firebase.auth.FirebaseAuth;  
*import* com.google.firebase.firestore.DocumentReference;  
*import* com.google.firebase.firestore.FirebaseFirestore;  
*import* com.google.firebase.firestore.QueryDocumentSnapshot;  
*import* com.google.firebase.firestore.QuerySnapshot;  
  
*import* java.util.HashMap;  
*import* java.util.*Map*;  
  
*//AppCompatActivity:  
public class* EditDetailsActivity *extends* AppCompatActivity {  
 *private static final* String TAG = "EditDetailsActivity";  
 TextView errorText;  
 *private* FirebaseAuth mAuth;  
 *private* EditText mEmail, mName, mUniversity, mAge, mPhone, mMajor;  
 *private* Button mApply;  
 String user\_id;  
 *private* FirebaseFirestore db;  
 @Override  
 *protected void* onCreate(Bundle savedInstanceState) {  
 *super*.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_edit\_details);  
 Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);  
 setSupportActionBar(toolbar);  
*// RunAnimation();* mAuth = FirebaseAuth.*getInstance*();  
 db = FirebaseFirestore.*getInstance*();  
 mApply = (Button)findViewById(R.id.btn\_apply);  
 mEmail = (EditText)findViewById(R.id.txt\_uname);  
 mName = (EditText) findViewById(R.id.txt\_name);  
 mPhone = (EditText) findViewById(R.id.txt\_phone);  
 mUniversity = (EditText) findViewById(R.id.txt\_university);  
 mAge = (EditText) findViewById(R.id.txt\_age);  
 mMajor = (EditText) findViewById(R.id.txt\_major);  
 errorText = (TextView)findViewById(R.id.lbl\_Error);  
  
 */\*\*  
 \* This function will make sure that the error message disappears  
 \* before user retypes the username/password  
 \*/* mEmail.setOnTouchListener(*new* View.OnTouchListener() {  
 @Override  
 *public boolean* onTouch(View view, MotionEvent event) {  
 *switch*(event.getAction()) {  
 *case* MotionEvent.ACTION\_DOWN:  
 mEmail.requestFocus();  
 errorText.setVisibility(View.GONE);  
 *// PRESSED  
 return true*; *// if you want to handle the touch event  
 case* MotionEvent.ACTION\_UP:  
 *// RELEASED  
 return true*; *// if you want to handle the touch event* }  
 *return false*;  
 }  
 });  
 mApply.setOnClickListener(*new* View.OnClickListener() {  
 @Override  
 *public void* onClick(View v) {  
 applyChanges(v);  
 }  
 });  
 user\_id = mAuth.getCurrentUser().getUid();  
 getUserInfo(user\_id);  
 }  
  
 @Override  
 *public boolean* onCreateOptionsMenu(*Menu* menu) {  
 *// Inflate the menu; this adds items to the action bar if it is present.* getMenuInflater().inflate(R.menu.menu\_login, menu);  
 *return true*;  
 }  
  
 @Override  
 *public boolean* onOptionsItemSelected(*MenuItem* item) {  
 *// Handle action bar item clicks here. The action bar will  
 // automatically handle clicks on the Home/Up button, so long  
 // as you specify a parent activity in AndroidManifest.xml.  
 int* id = item.getItemId();  
  
 *//noinspection SimplifiableIfStatement  
 if* (id == R.id.action\_settings) {  
 *return true*;  
 }  
  
 *return super*.onOptionsItemSelected(item);  
 }  
 *//This is where the identity check is. The main variable is the validation flag. If it is false, then the user  
 //will not go anywhere beyond the login page. Else if the user enters the right username and password, he will be directed  
 //to the next page. Username and password are determined by the 2 variables "userName" and "password"  
 public void* applyChanges(View v)  
 {  
 *final* String email = mEmail.getText().toString();  
 *final* String age = mAge.getText().toString();  
 *final* String phone = mPhone.getText().toString();  
 *final* String university = mUniversity.getText().toString();  
 *final* String name = mName.getText().toString();  
 *final* String major = mMajor.getText().toString();  
 *if*(email.isEmpty()){  
 mEmail.setError("Email Address cannot be blank");  
 }  
 *else if*(!Patterns.EMAIL\_ADDRESS.matcher(email).matches()){  
 mEmail.setError("Enter a Valid Email");  
 mEmail.requestFocus();  
 }  
 *else if*(name.isEmpty()){  
 mName.setError("Name cannot be blank");  
 mName.requestFocus();  
 }  
 *else if*(age.isEmpty()){  
 mAge.setError("Age cannot be blank");  
 mAge.requestFocus();  
 }  
 *else if*(university.isEmpty()){  
 mUniversity.setError("University cannot be blank");  
 mUniversity.requestFocus();  
 }  
 *else if*(phone.isEmpty()){  
 mPhone.setError("Phone cannot be blank");  
 mPhone.requestFocus();  
 }  
 *else if*(major.isEmpty()){  
 mMajor.setError("Major cannot be blank");  
 mMajor.requestFocus();  
 }  
 *else* {  
 *///Using FireStore* DocumentReference docRef = db.collection("users").document(user\_id);  
 *// Add document data with string user\_id using a hashmap  
 Map*<String, Object> data = *new* HashMap<>();  
 data.put("email", email);  
 data.put("name", name);  
 data.put("phone", phone);  
 data.put("age", age);  
 data.put("university", university);  
 data.put("major", major);  
 *//asynchronously write data* docRef.update(data)  
 .addOnSuccessListener(*new* OnSuccessListener<Void>() {  
 @Override  
 *public void* onSuccess(Void aVoid) {  
 Log.*d*(TAG, "Personal Details Updated Successfully!");  
 Intent intent = *new* Intent(EditDetailsActivity.*this*, MainActivity.*class*);  
 startActivity(intent);  
 finish();  
 }  
 })  
 .addOnFailureListener(*new* OnFailureListener() {  
 @Override  
 *public void* onFailure(@NonNull Exception e) {  
 Log.*w*(TAG, "Error Updating Details! Please Try Again Later!", e);  
 }  
 });  
  
 }  
 }  
 *public void* getUserInfo(*final* String ID) {  
 db.collection("users")  
 .get()  
 .addOnCompleteListener(*new* OnCompleteListener<QuerySnapshot>() {  
 @Override  
 *public void* onComplete(@NonNull Task<QuerySnapshot> task) {  
 *if* (task.isSuccessful()) {  
 *for* (QueryDocumentSnapshot document : task.getResult()) {  
 *if* (document.getId().equals(ID)) {  
  
 *Map*<String, Object> user;  
 user = document.getData();  
 displayUserInfo(user);  
  
 }  
 *//Log.d(TAG, document.getId() + " => " + document.getData());* }  
 } *else* {  
 *//Log.w(TAG, "Error getting documents.", task.getException());* }  
 }  
 });  
 }  
 *public void* displayUserInfo (*Map*<String, Object> usermap){  
*// ArrayList<String> items = new ArrayList<String>();  
// items.add("Name: " + usermap.get("name"));  
// items.add("Email: " + usermap.get("email"));  
// items.add("Age: " + usermap.get("age"));  
// items.add("Phone Number: " + usermap.get("phone"));  
// items.add("Institution/ Organization: " + usermap.get("university"));  
// items.add("Major: " + usermap.get("major"));  
// ArrayAdapter<String> itemsAdapter =  
// new ArrayAdapter<String>(this, android.R.layout.simple\_list\_item\_1, items);  
// listView.setAdapter(itemsAdapter);* String name = ((usermap == *null*) || (usermap.get("name") == *null*) ? "N/A" : usermap.get("name").toString());  
 String age = ((usermap == *null*) || (usermap.get("age") == *null*) ? "N/A" : usermap.get("age").toString());  
 String email = ((usermap == *null*) || (usermap.get("email") == *null*) ? "N/A" : usermap.get("email").toString());  
 String phone = ((usermap == *null*) || (usermap.get("phone") == *null*) ? "N/A" : usermap.get("phone").toString());  
 String university = ((usermap == *null*) || (usermap.get("university") == *null*) ? "N/A" : usermap.get("university").toString());  
 String major = ((usermap == *null*) || (usermap.get("major") == *null*) ? "N/A" : usermap.get("major").toString());  
 mEmail.setText(email);  
 mName.setText(name);  
 mAge.setText(age);  
 mUniversity.setText(university);  
 mPhone.setText(phone);  
 mMajor.setText(major);  
 }  
  
 *public void* goBack() {  
 *return*;  
 }  
 */\*\*  
 \* this function animates the Login title by continuously scaling it back and forth  
 \*/  
// private void RunAnimation()  
// {  
// Animation a = AnimationUtils.loadAnimation(this, R.anim.fade\_in);  
// a.reset();  
// TextView tv = (TextView) findViewById(R.id.lbl\_Header);  
// tv.clearAnimation();  
// tv.startAnimation(a);  
// }* @Override  
 *protected void* onStart() {  
 *super*.onStart();  
 }  
  
 @Override  
 *protected void* onStop() {  
 *super*.onStop();  
 }  
}

*package* com.csee5590.lab3;  
  
*import* android.content.Intent;  
*import* android.support.annotation.NonNull;  
*import* android.support.v7.app.AppCompatActivity;  
*import* android.os.Bundle;  
*import* android.util.Log;  
*import* android.widget.ArrayAdapter;  
*import* android.widget.ListView;  
*import* android.widget.Toast;  
  
*import* java.text.DateFormat;  
*import* java.text.ParseException;  
*import* java.text.SimpleDateFormat;  
*import* java.util.ArrayList;  
*import* java.util.Calendar;  
  
  
*import* com.android.volley.Request;  
*import* com.android.volley.RequestQueue;  
*import* com.android.volley.Response;  
*import* com.android.volley.VolleyError;  
*import* com.android.volley.toolbox.JsonObjectRequest;  
*import* com.android.volley.toolbox.Volley;  
*import* com.google.android.gms.tasks.*OnFailureListener*;  
*import* com.google.android.gms.tasks.*OnSuccessListener*;  
*import* com.google.firebase.auth.FirebaseAuth;  
*import* com.google.firebase.firestore.FirebaseFirestore;  
*import* com.google.firebase.firestore.SetOptions;  
  
*import* org.json.JSONArray;  
*import* org.json.JSONException;  
*import* org.json.JSONObject;  
  
*import* java.util.Date;  
*import* java.util.HashMap;  
*import* java.util.*List*;  
*import* java.util.*Map*;  
*import* java.util.TimeZone;  
  
*public class* GetInfoActivity *extends* AppCompatActivity {  
 *private* FirebaseAuth mAuth;  
 *private* FirebaseFirestore db;  
 String country\_code;  
 ArrayList<String> neighbors = *new* ArrayList<String>();  
 *private* ArrayList<String> items;  
 *private* ListView listView;  
 ArrayAdapter<String> itemsAdapter;  
 @Override  
 *protected void* onCreate(Bundle savedInstanceState) {  
 *super*.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_get\_info);  
 mAuth = FirebaseAuth.*getInstance*();  
 db = FirebaseFirestore.*getInstance*();  
 Intent intent = getIntent();  
 country\_code = intent.getStringExtra("country\_code");  
 String country\_name = intent.getStringExtra("country\_name");  
 Log.*d*("Country Code 2", country\_code);  
 Log.*d*("Country Name 2", country\_name);  
 String url = "https://restcountries.eu/rest/v2/alpha/" + country\_code;  
 RequestQueue queue = Volley.*newRequestQueue*(*this*);  
 JsonObjectRequest jsonObjectRequest = *new* JsonObjectRequest  
 (Request.*Method*.GET, url, *null*, *new* Response.Listener<JSONObject>() {  
  
 @Override  
 *public void* onResponse(JSONObject response) {  
 Log.*d*("Response","Response: " + response.toString());  
 *try* {  
 savetoDB(response);  
 displayInfo(response);  
 } *catch* (JSONException e) {  
 e.printStackTrace();  
 }  
*// catch (ParseException e) {  
// e.printStackTrace();  
// }* }  
 }, *new* Response.ErrorListener() {  
  
 @Override  
 *public void* onErrorResponse(VolleyError error) {  
 *// TODO: Handle error* Log.*d*("Error","Response ERROR!");  
 }  
 });  
 queue.add(jsonObjectRequest);  
 listView = (ListView) findViewById(R.id.listView);  
 items = *new* ArrayList<String>();  
 ArrayAdapter<String> itemsAdapter =  
 *new* ArrayAdapter<String>(*this*, android.R.layout.simple\_list\_item\_1, items);  
 listView.setAdapter(itemsAdapter);  
 }  
 *private void* savetoDB(JSONObject obj) *throws* JSONException{  
 String user\_id = mAuth.getCurrentUser().getUid();  
 Intent intent = getIntent();  
 *Map*<String, Object> country = *new* HashMap<>();  
 country.put("name", obj.get("name"));  
 country.put("capital", obj.get("capital"));  
 JSONArray array = (JSONArray) obj.get("currencies");  
 *List*<String> data = *new* ArrayList<>();  
 *for* (*int* i = 0; i < array.length(); i++) {  
 JSONObject row = array.getJSONObject(i);  
 String currencyname = row.getString("name");  
 data.add(currencyname);  
  
 }  
 country.put("currencies", data);  
 country.put("checkin\_time", currentDate());  
 country.put("latitude", intent.getDoubleExtra("lat", 0));  
 country.put("longitude", intent.getDoubleExtra("long", 0));  
 country.put("population", obj.get("population"));  
  
 array = (JSONArray) obj.get("borders");  
 *for* (*int* i = 0; i < array.length(); i++) {  
 String row = array.getString(i);  
 String neighbor\_code = row;  
 getCountryName(neighbor\_code);  
 }  
 country.put("neighbors", neighbors);  
 db.collection("users").document(user\_id).  
 collection("countries").document(obj.get("alpha2Code").toString())  
 .set(country)  
 .addOnSuccessListener(*new* OnSuccessListener<Void>() {  
 @Override  
 *public void* onSuccess(Void aVoid) {  
 Log.d("GetInfoActivity", "DocumentSnapshot successfully written!");  
 Toast.makeText(GetInfoActivity.*this*, "Information successfully saved to database!",  
 Toast.LENGTH\_LONG).show();  
 }  
 })  
 .addOnFailureListener(*new* OnFailureListener() {  
 @Override  
 *public void* onFailure(@NonNull Exception e) {  
 Log.w("GetInfoActivity", "Error writing document", e);  
 }  
 });  
  
 }  
 *private void* displayInfo(JSONObject obj) *throws* JSONException {  
 Intent intent = getIntent();  
  
 Map<String, Object> country = *new* HashMap<>();  
 country.put("name", obj.get("name"));  
 country.put("capital", obj.get("capital"));  
 JSONArray array = (JSONArray) obj.get("currencies");  
 List<String> data = *new* ArrayList<>();  
 *for* (*int* i = 0; i < array.length(); i++) {  
 JSONObject row = array.getJSONObject(i);  
 String currencyname = row.getString("name");  
 data.add(currencyname);  
 }  
  
 country.put("currencies", data);  
 country.put("checkin\_time", currentDate());  
 country.put("latitude", intent.getDoubleExtra("lat", 0));  
 country.put("longitude", intent.getDoubleExtra("long", 0));  
 country.put("population", obj.get("population"));  
 items = *new* ArrayList<String>();  
 String name = ((country == *null*) || (country.get("name") == *null*) ? "N/A" : country.get("name").toString());  
 String checkin\_time = ((country == *null*) || (country.get("checkin\_time") == *null*) ? "N/A" : country.get("checkin\_time").toString());  
 String latitude = ((country == *null*) || (country.get("latitude") == *null*) ? "N/A" : country.get("latitude").toString());  
 String longitude = ((country == *null*) || (country.get("longitude") == *null*) ? "N/A" : country.get("longitude").toString());  
 String population = ((country == *null*) || (country.get("population") == *null*) ? "N/A" : country.get("population").toString());  
 String currencies = ((country == *null*) || (country.get("currencies") == *null*) ? "N/A" : country.get("currencies").toString());  
 items.add("Your Country's Name: " + name);  
 items.add("Checked-in Time: " + checkin\_time);  
 items.add("Checked-in Coordinates: (" + latitude + ", " + longitude + ")");  
 items.add("Population: " + population + " people");  
 items.add("Currencies: " + currencies);  
 itemsAdapter =  
 *new* ArrayAdapter<String>(*this*, android.R.layout.simple\_list\_item\_1, items);  
 listView.setAdapter(itemsAdapter);  
  
 }  
  
 *private* String currentDate(){  
 String DATE\_FORMAT = "dd/MM/yyyy hh:mm:ss a";  
 Calendar calendar = Calendar.getInstance(TimeZone.getTimeZone("GMT-5"));  
 System.out.println("TimeZone of calendar : " + calendar.getTimeZone().getID());  
 SimpleDateFormat formatter = *new* SimpleDateFormat(DATE\_FORMAT);  
 formatter.setTimeZone(TimeZone.getTimeZone("GMT-5"));  
 System.out.println("Central Daylight Time : " + formatter.format(calendar.getTime()));  
 *return* formatter.format(calendar.getTime());  
 }  
 *private void* getCountryName(String code) {  
 String url = "https://restcountries.eu/rest/v2/alpha/" + code;  
 RequestQueue queue = Volley.newRequestQueue(*this*);  
 JsonObjectRequest jsonObjectRequest = *new* JsonObjectRequest  
 (Request.Method.GET, url, *null*, *new* Response.Listener<JSONObject>() {  
  
 @Override  
 *public void* onResponse(JSONObject response) {  
 Log.d("Response","Response: " + response.toString());  
 *try* {  
 Log.d("Neighbor's Name", response.get("name").toString() );  
 neighbors.add(response.get("name").toString());  
 saveNeighborToDB();  
 displayNeighbor();  
 } *catch* (JSONException e) {  
 e.printStackTrace();  
 }  
 }  
 }, *new* Response.ErrorListener() {  
  
 @Override  
 *public void* onErrorResponse(VolleyError error) {  
 *// TODO: Handle error* Log.d("Error","Response ERROR!");  
 }  
 });  
 queue.add(jsonObjectRequest);  
 }  
 *private void* saveNeighborToDB(){  
 String user\_id = mAuth.getCurrentUser().getUid();  
 Map<String, Object> neighbor\_map = *new* HashMap<>();  
 neighbor\_map.put("neighbors", neighbors);  
 db.collection("users").document(user\_id).  
 collection("countries").document(country\_code)  
 .set(neighbor\_map, SetOptions.merge())  
 .addOnSuccessListener(*new* OnSuccessListener<Void>() {  
 @Override  
 *public void* onSuccess(Void aVoid) {  
 Log.d("GetInfoActivity", "DocumentSnapshot successfully updated with new neighbor!");  
 Toast.makeText(GetInfoActivity.*this*, "Neighbor successfully saved to database!",  
 Toast.LENGTH\_LONG).show();  
 }  
 })  
 .addOnFailureListener(*new* OnFailureListener() {  
 @Override  
 *public void* onFailure(@NonNull Exception e) {  
 Log.w("GetInfoActivity", "Error writing document", e);  
 }  
 });  
  
 }  
 *private void* displayNeighbor(){  
 Log.d("current neighbors ", neighbors.toString());  
 *if* (items.get(items.size()-1).contains("Neighbors: ")){  
 items.remove(items.remove(items.size()-1));  
 items.add("Neighbors: " + neighbors.toString());  
 }  
 *else*{  
 items.add("Neighbors: " + neighbors.toString());  
 }  
 itemsAdapter =  
 *new* ArrayAdapter<String>(*this*, android.R.layout.simple\_list\_item\_1, items);  
 listView.setAdapter(itemsAdapter);  
  
 }  
}

*package* com.csee5590.lab3;  
  
*import* android.content.Intent;  
*import* android.support.annotation.NonNull;  
*import* android.support.design.widget.Snackbar;  
*import* android.support.v7.app.AppCompatActivity;  
*import* android.os.Bundle;  
*import* android.util.Log;  
  
*import* com.google.android.gms.auth.api.signin.GoogleSignIn;  
*import* com.google.android.gms.auth.api.signin.GoogleSignInAccount;  
*import* com.google.android.gms.auth.api.signin.GoogleSignInClient;  
*import* com.google.android.gms.auth.api.signin.GoogleSignInOptions;  
*import* com.google.android.gms.common.api.ApiException;  
*import* com.google.android.gms.tasks.*OnCompleteListener*;  
*import* com.google.android.gms.tasks.*OnFailureListener*;  
*import* com.google.android.gms.tasks.*OnSuccessListener*;  
*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;  
*import* com.google.firebase.firestore.DocumentReference;  
*import* com.google.firebase.firestore.FirebaseFirestore;  
  
*import* java.util.HashMap;  
*import* java.util.*Map*;  
  
*public class* GoogleSignInActivity *extends* AppCompatActivity {  
 *private static final* String TAG = "GoogleSignInActivity";  
 *private* GoogleSignInClient mGoogleSignInClient;  
 *private static final int* RC\_SIGN\_IN = 9001;  
 *private* FirebaseAuth mAuth;  
 *private* FirebaseFirestore db;  
 DocumentReference docRef;  
 *Map*<String, Object> data;  
 @Override  
 *protected void* onCreate(Bundle savedInstanceState) {  
 *super*.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_google\_sign\_in);  
 mAuth = FirebaseAuth.*getInstance*();  
 db = FirebaseFirestore.*getInstance*();  
 *///Google Sign In Implementation  
 // Configure Google Sign In* GoogleSignInOptions gso = *new* GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT\_SIGN\_IN)  
 .requestIdToken(getString(R.string.default\_web\_client\_id))  
 .requestEmail()  
 .build();  
 mGoogleSignInClient = GoogleSignIn.*getClient*(*this*, gso);  
 mAuth = FirebaseAuth.*getInstance*();  
 signIn();  
 }  
 *private void* signIn() {  
 Intent signInIntent = mGoogleSignInClient.getSignInIntent();  
 startActivityForResult(signInIntent, RC\_SIGN\_IN);  
 }  
 @Override  
 *public void* onActivityResult(*int* requestCode, *int* resultCode, Intent data) {  
 *super*.onActivityResult(requestCode, resultCode, data);  
  
 *// Result returned from launching the Intent from GoogleSignInApi.getSignInIntent(...);  
 if* (requestCode == RC\_SIGN\_IN) {  
 Task<GoogleSignInAccount> task = GoogleSignIn.*getSignedInAccountFromIntent*(data);  
 *try* {  
 *// Google Sign In was successful, authenticate with Firebase* GoogleSignInAccount account = task.getResult(ApiException.*class*);  
 firebaseAuthWithGoogle(account);  
 } *catch* (ApiException e) {  
 *// Google Sign In failed, update UI appropriately* Log.*w*(TAG, "Google sign in failed", e);  
 *// ...* }  
 }  
 }  
 *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()) {  
 *// Sign in success, update UI with the signed-in user's information* Log.*d*(TAG, "signInWithCredential:success");  
 FirebaseUser user = mAuth.getCurrentUser();  
*// Pass user object to this method* updateDB(user);  
 } *else* {  
 *// If sign in fails, display a message to the user.* Log.*w*(TAG, "signInWithCredential:failure", task.getException());  
 Snackbar.*make*(findViewById(R.id.gsignin\_layout), "Authentication Failed.", Snackbar.LENGTH\_SHORT).show();  
 updateDB(*null*);  
 }  
  
 *// ...* }  
 });  
 }  
 *private void* updateDB(FirebaseUser user) {  
*// hideProgressDialog();  
 if* (user != *null*){  
 String user\_email = user.getEmail();  
 String user\_id = user.getUid();  
 String user\_name = user.getDisplayName();  
 *///Using FireStore to update document with id = user\_id which is passed as a parameter* docRef = db.collection("users").document(user\_id);  
 *// Add document data with string user\_id using a hashmap* data = *new* HashMap<>();  
 data.put("email", user\_email);  
 data.put("name", user\_name);  
 *//asynchronously write data* docRef.update(data)  
 .addOnSuccessListener(*new* OnSuccessListener<Void>() {  
 @Override  
 *public void* onSuccess(Void aVoid) {  
 Log.*d*(TAG, "Personal Details Updated Successfully!");  
 Intent intent = *new* Intent(GoogleSignInActivity.*this*, MainActivity.*class*);  
 startActivity(intent);  
 finish();  
 }  
 })  
 .addOnFailureListener(*new* OnFailureListener() {  
 @Override  
 *public void* onFailure(@NonNull Exception e) {  
 Log.*w*(TAG, "Error Updating Details! Please Try Again Later!", e);  
 docRef.set(data);  
 Intent intent = *new* Intent(GoogleSignInActivity.*this*, MainActivity.*class*);  
 startActivity(intent);  
 }  
 });  
 }  
  
 Log.*d*(TAG, "in updateDB()");  
 }  
 @Override  
 *public void* onStart() {  
 *super*.onStart();  
 *// Check if user is signed in (non-null) and update UI accordingly.* FirebaseUser currentUser = mAuth.getCurrentUser();  
 updateDB(currentUser);  
 }  
}

*package* com.csee5590.lab3;  
  
*import* android.content.Intent;  
*import* android.os.Bundle;  
*import* android.support.annotation.NonNull;  
*import* android.support.v7.app.AppCompatActivity;  
*import* android.support.v7.widget.Toolbar;  
*import* android.util.Patterns;  
*import* android.view.MotionEvent;  
*import* android.view.View;  
*import* android.view.*Menu*;  
*import* android.view.*MenuItem*;  
*import* android.view.animation.Animation;  
*import* android.view.animation.AnimationUtils;  
*import* android.widget.Button;  
*import* android.widget.EditText;  
*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.*AuthResult*;  
*import* com.google.firebase.auth.FirebaseAuth;  
*import* com.google.firebase.auth.FirebaseUser;  
  
*//AppCompatActivity:  
public class* LoginActivity *extends* AppCompatActivity {  
 TextView errorText;  
 *private* FirebaseAuth mAuth;  
 *private* EditText mEmail, mPassword;  
 *private* Button mLogin, mBack;  
 *private* FirebaseAuth.*AuthStateListener* firebaseAuthListener;  
 @Override  
 *protected void* onCreate(Bundle savedInstanceState) {  
 *super*.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_login);  
 Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);  
 setSupportActionBar(toolbar);  
*// RunAnimation();* mAuth = FirebaseAuth.*getInstance*();  
 firebaseAuthListener = *new* FirebaseAuth.AuthStateListener() {  
 @Override  
 *public void* onAuthStateChanged(@NonNull FirebaseAuth firebaseAuth) {  
 FirebaseUser user = FirebaseAuth.*getInstance*().getCurrentUser(); *//get the information of the current user  
 if* (user != *null*){  
 *//if not null, move to another activity, to be created later.  
 //Remember the current context!* Intent intent = *new* Intent(LoginActivity.*this*, MainActivity.*class*);  
 intent.putExtra("user\_id", user.getUid());  
 intent.putExtra("user\_email", user.getEmail());  
 startActivity(intent);  
 finish();  
 }  
 }  
 };  
 mLogin = (Button)findViewById(R.id.btn\_login);  
 mEmail = (EditText)findViewById(R.id.txt\_uname);  
 mPassword = (EditText) findViewById(R.id.txt\_Pwd);  
 errorText = (TextView)findViewById(R.id.lbl\_Error);  
 mBack = (Button) findViewById(R.id.btnLoginBack);  
  
 mBack.setOnClickListener(*new* View.OnClickListener() {  
 @Override  
 *public void* onClick(View v) {  
 Intent intent = *new* Intent(LoginActivity.*this*, WelcomeActivity.*class*);  
 startActivity(intent);  
 finish();  
 }  
 });  
   
 */\*\*  
 \* This function will make sure that the error message disappears  
 \* before user retypes the username/password  
 \*/* mEmail.setOnTouchListener(*new* View.OnTouchListener() {  
 @Override  
 *public boolean* onTouch(View view, MotionEvent event) {  
 *switch*(event.getAction()) {  
 *case* MotionEvent.ACTION\_DOWN:  
 mEmail.requestFocus();  
 errorText.setVisibility(View.GONE);  
 *// PRESSED  
 return true*; *// if you want to handle the touch event  
 case* MotionEvent.ACTION\_UP:  
 *// RELEASED  
 return true*; *// if you want to handle the touch event* }  
 *return false*;  
 }  
 });  
 mPassword.setOnTouchListener(*new* View.OnTouchListener() {  
 @Override  
 *public boolean* onTouch(View view, MotionEvent event) {  
 *switch*(event.getAction()) {  
 *case* MotionEvent.ACTION\_DOWN:  
 mPassword.requestFocus();  
 errorText.setVisibility(View.GONE);  
 *// PRESSED  
 return true*; *// if you want to handle the touch event  
 case* MotionEvent.ACTION\_UP:  
 *// RELEASED  
 return true*; *// if you want to handle the touch event* }  
 *return false*;  
 }  
 });  
 mLogin.setOnClickListener(*new* View.OnClickListener() {  
 @Override  
 *public void* onClick(View v) {  
 checkCredentials(v);  
 }  
 });  
 }  
  
 @Override  
 *public boolean* onCreateOptionsMenu(*Menu* menu) {  
 *// Inflate the menu; this adds items to the action bar if it is present.* getMenuInflater().inflate(R.menu.menu\_login, menu);  
 *return true*;  
 }  
  
 @Override  
 *public boolean* onOptionsItemSelected(*MenuItem* item) {  
 *// Handle action bar item clicks here. The action bar will  
 // automatically handle clicks on the Home/Up button, so long  
 // as you specify a parent activity in AndroidManifest.xml.  
 int* id = item.getItemId();  
  
 *//noinspection SimplifiableIfStatement  
 if* (id == R.id.action\_settings) {  
 *return true*;  
 }  
  
 *return super*.onOptionsItemSelected(item);  
 }  
 *//This is where the identity check is. The main variable is the validation flag. If it is false, then the user  
 //will not go anywhere beyond the login page. Else if the user enters the right username and password, he will be directed  
 //to the next page. Username and password are determined by the 2 variables "userName" and "password"  
 public void* checkCredentials(View v)  
 {  
 *final* String email = mEmail.getText().toString();  
 *final* String password = mPassword.getText().toString();  
 *if*(email.isEmpty() ){  
 mEmail.setError("Email Address cannot be blank");  
 }  
 *else if*(!Patterns.EMAIL\_ADDRESS.matcher(email).matches()){  
 mEmail.setError("Enter a Valid Email");  
 }  
 *else if*(password.isEmpty()){  
 mPassword.setError("Password cannot be blank");  
 }  
 *else* {  
 mAuth.signInWithEmailAndPassword(email, password).addOnCompleteListener(LoginActivity.*this*, *new* OnCompleteListener<*AuthResult*>() {  
 @Override  
 *public void* onComplete(@NonNull Task<*AuthResult*> task) {  
 *if* (!task.isSuccessful()) {  
 Toast.*makeText*(LoginActivity.*this*, "Authentication Failed!", Toast.LENGTH\_SHORT).show();  
 }  
 }  
 });  
 }  
  
 }  
  
 */\*\*  
 \* this function animates the Login title by continuously scaling it back and forth  
 \*/  
// private void RunAnimation()  
// {  
// Animation a = AnimationUtils.loadAnimation(this, R.anim.fade\_in);  
// a.reset();  
// TextView tv = (TextView) findViewById(R.id.lbl\_Header);  
// tv.clearAnimation();  
// tv.startAnimation(a);  
// }* @Override  
 *protected void* onStart() {  
 *super*.onStart();  
 mAuth.addAuthStateListener(firebaseAuthListener);  
 }  
  
 @Override  
 *protected void* onStop() {  
 *super*.onStop();  
 mAuth.removeAuthStateListener(firebaseAuthListener);  
 }  
}

*package* com.csee5590.lab3;  
  
*import* android.app.Activity;  
*import* android.content.Intent;  
*import* android.graphics.Bitmap;  
*import* android.graphics.Color;  
*import* android.net.Uri;  
*import* android.provider.MediaStore;  
*import* android.support.annotation.NonNull;  
*import* android.support.annotation.Nullable;  
*import* android.support.v7.app.AppCompatActivity;  
*import* android.os.Bundle;  
*import* android.util.Log;  
*import* android.view.MotionEvent;  
*import* android.view.View;  
*import* android.widget.ArrayAdapter;  
*import* android.widget.Button;  
*import* android.widget.ImageView;  
*import* android.widget.ListView;  
*import* android.widget.Toast;  
  
*import* com.google.android.gms.tasks.*OnCompleteListener*;  
*import* com.google.android.gms.tasks.*OnFailureListener*;  
*import* com.google.android.gms.tasks.*OnSuccessListener*;  
*import* com.google.android.gms.tasks.Task;  
*import* com.google.firebase.auth.FirebaseAuth;  
*import* com.google.firebase.firestore.DocumentReference;  
*import* com.google.firebase.firestore.FirebaseFirestore;  
*import* com.google.firebase.firestore.QueryDocumentSnapshot;  
*import* com.google.firebase.firestore.QuerySnapshot;  
*import* com.google.firebase.storage.FirebaseStorage;  
*import* com.google.firebase.storage.StorageReference;  
*import* com.google.firebase.storage.UploadTask;  
  
*import* java.io.ByteArrayOutputStream;  
*import* java.io.IOException;  
*import* java.util.ArrayList;  
*import* java.util.HashMap;  
*import* java.util.*Map*;  
  
*public class* MainActivity *extends* AppCompatActivity {  
 *private static final* String TAG = "MainActivity";  
 *private* FirebaseFirestore db;  
 *private* String user\_id;  
 *private* ArrayList<String> items;  
 *private* ListView listView;  
 *private* ImageView mProfileImage;  
 *private* String mProfileURL;  
 *private* Uri resultURI;  
 *//Firebase Authentication Instances  
 private* FirebaseAuth mAuth;  
 @Override  
 *protected void* onCreate(Bundle savedInstanceState) {  
 *super*.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_main);  
 ImageView imageView = (ImageView) findViewById(R.id.imageView);  
 Button logOutBtn = (Button) findViewById(R.id.button);  
 Button editBtn = (Button) findViewById(R.id.btn\_edit);  
 Button checkinBtn = (Button) findViewById(R.id.btn\_checkin);  
 Button photosBtn = (Button) findViewById(R.id.btn\_photos);  
 db = FirebaseFirestore.*getInstance*();  
 mAuth = FirebaseAuth.*getInstance*();  
*// Animation upon clicking/releasing click* imageView.setOnTouchListener(*new* View.OnTouchListener() {  
 @Override  
 *public boolean* onTouch(View view, MotionEvent event) {  
 *switch*(event.getAction()) {  
 *case* MotionEvent.ACTION\_DOWN:  
 *// PRESSED* view.setScaleY((*float*) 1.2);  
 view.setScaleX((*float*) 1.2);  
 *return true*; *// if you want to handle the touch event  
 case* MotionEvent.ACTION\_UP:  
 *// RELEASED* view.setScaleY((*float*) 1);  
 view.setScaleX((*float*) 1);  
 *return true*; *// if you want to handle the touch event* }  
 *return false*;  
 }  
 });  
 logOutBtn.setOnTouchListener(*new* View.OnTouchListener() {  
 *private long* start = 0;  
 *private long* end = 0;  
 @Override  
 *public boolean* onTouch(View view, MotionEvent event) {  
 *switch*(event.getAction()) {  
 *case* MotionEvent.ACTION\_DOWN:  
 *// PRESSED  
 //Start counting the pressing duration using start and end* start = System.*currentTimeMillis*();  
 view.setScaleY((*float*) 1.2);  
 view.setScaleX((*float*) 1.2);  
 view.setBackgroundColor(Color.BLUE);  
 *return true*; *// if you want to handle the touch event  
 case* MotionEvent.ACTION\_UP:  
 *// RELEASED* end = System.*currentTimeMillis*();  
 view.setScaleY((*float*) 1);  
 view.setScaleX((*float*) 1);  
 view.setBackgroundColor(Color.RED);  
 *//If the button is held/pressed for more than 3 seconds, the screen will not log out to prevent loss of information.  
 //Holding a button for more than three seconds can mean that the user is hesitating leaving the site.  
 if* (end - start <=3000){  
 LogOut(view);  
 }  
 *return true*; *// if you want to handle the touch event* }  
 *return false*;  
 }  
 });  
*// Click listener for edit button* editBtn.setOnClickListener(*new* View.OnClickListener() {  
 @Override  
 *public void* onClick(View v) {  
 Intent intent = *new* Intent(MainActivity.*this*, EditDetailsActivity.*class*);  
 startActivity(intent);  
 }  
 });  
 checkinBtn.setOnClickListener(*new* View.OnClickListener() {  
 @Override  
 *public void* onClick(View v) {  
 Intent intent = *new* Intent(MainActivity.*this*, LocationActivity.*class*);  
 startActivity(intent);  
 }  
 });  
 photosBtn.setOnClickListener(*new* View.OnClickListener() {  
 @Override  
 *public void* onClick(*final* View v) {  
 Intent intent = *new* Intent(MainActivity.*this*, PhotosActivity.*class*);  
 startActivity(intent);  
 }  
 });  
  
 *///Retrieve User Info* user\_id = mAuth.getCurrentUser().getUid();  
 getUserInfo(user\_id);  
 listView = (ListView) findViewById(R.id.listView);  
 items = *new* ArrayList<String>();  
 ArrayAdapter<String> itemsAdapter =  
 *new* ArrayAdapter<String>(*this*, android.R.layout.simple\_list\_item\_1, items);  
 listView.setAdapter(itemsAdapter);  
  
 *//Profile Image* mProfileImage = (ImageView) findViewById(R.id.profileImg);  
 *//Click a picture* mProfileImage.setOnClickListener(*new* View.OnClickListener() {  
 @Override  
 *public void* onClick(View v) {  
 Intent intent = *new* Intent(Intent.ACTION\_PICK);  
 intent.setType("image/\*"); *//restricts selection* startActivityForResult(intent, 1);  
 }  
 });  
 }  
*// Log out function  
 public void* LogOut(View v){  
 FirebaseAuth.*getInstance*().signOut();  
 Intent intent = *new* Intent(MainActivity.*this*, WelcomeActivity.*class*);  
 startActivity(intent);  
 finish();  
 Toast.*makeText*(MainActivity.*this*, "logout", Toast.LENGTH\_SHORT);  
 }  
*// Retrieve user information  
 public void* getUserInfo(*final* String ID) {  
 db.collection("users")  
 .get()  
 .addOnCompleteListener(*new* OnCompleteListener<QuerySnapshot>() {  
 @Override  
 *public void* onComplete(@NonNull Task<QuerySnapshot> task) {  
 *if* (task.isSuccessful()) {  
 *for* (QueryDocumentSnapshot document : task.getResult()) {  
 *if* (document.getId().equals(ID)) {  
  
 *Map*<String, Object> user;  
 user = document.getData();  
 displayUserInfo(user);  
  
 }  
 *//Log.d(TAG, document.getId() + " => " + document.getData());* }  
 } *else* {  
 *//Log.w(TAG, "Error getting documents.", task.getException());* }  
 }  
 });  
 }  
*// This function takes in a map and display it as a listview  
 public void* displayUserInfo (*Map*<String, Object> usermap){  
 ArrayList<String> items = *new* ArrayList<String>();  
 String name = ((usermap == *null*) || (usermap.get("name") == *null*) ? "N/A" : usermap.get("name").toString());  
 String age = ((usermap == *null*) || (usermap.get("age") == *null*) ? "N/A" : usermap.get("age").toString());  
 String email = ((usermap == *null*) || (usermap.get("email") == *null*) ? "N/A" : usermap.get("email").toString());  
 String phone = ((usermap == *null*) || (usermap.get("phone") == *null*) ? "N/A" : usermap.get("phone").toString());  
 String university = ((usermap == *null*) || (usermap.get("university") == *null*) ? "N/A" : usermap.get("university").toString());  
 String major = ((usermap == *null*) || (usermap.get("major") == *null*) ? "N/A" : usermap.get("major").toString());  
  
 items.add("Name: " + name);  
 items.add("Email: " + email);  
 items.add("Age: " + age);  
 items.add("Phone Number: " + phone);  
 items.add("Institution/ Organization: " + university);  
 items.add("Major: " + major);  
 ArrayAdapter<String> itemsAdapter =  
 *new* ArrayAdapter<String>(*this*, android.R.layout.simple\_list\_item\_1, items);  
 listView.setAdapter(itemsAdapter);  
 }  
*// The section below is for a profile picture, not tested thoroughly yet.  
// Profile Picture (Not tested)* @Override  
 *protected void* onActivityResult(*int* requestCode, *int* resultCode, @Nullable Intent data) {  
 *super*.onActivityResult(requestCode, resultCode, data);  
 *//See if it returns the same code from the startactivityforresult above  
 if* (requestCode == 1 && resultCode == Activity.RESULT\_OK){  
 *final* Uri imageURI = data.getData();  
 resultURI = imageURI;  
 mProfileImage.setImageURI(resultURI);  
 applyChanges();  
 }  
 }  
 *//Apply changes for profile images on Firebase Storage  
 //Note: need to enable storage first for permission issues.  
 private void* applyChanges(){  
  
 *if* (resultURI != *null*){  
 *final* StorageReference filePath = FirebaseStorage.getInstance().getReference().child("profile\_images").child(user\_id);  
 Bitmap bitmap = *null*;  
 *//Add uri to bitmap  
 try* {  
 bitmap = MediaStore.Images.Media.getBitmap(getApplication().getContentResolver(), resultURI);  
 } *catch* (IOException e) {  
 e.printStackTrace();  
 }  
 ByteArrayOutputStream baos = *new* ByteArrayOutputStream(); *//compress images* bitmap.compress(Bitmap.CompressFormat.JPEG, 20, baos);  
 *byte*[] data = baos.toByteArray();  
 UploadTask uploadTask = filePath.putBytes(data);  
 *//Upload task will save the image to Firebase Database* uploadTask.addOnSuccessListener(*new* OnSuccessListener<UploadTask.TaskSnapshot>() {  
 @Override  
 *public void* onSuccess(UploadTask.TaskSnapshot taskSnapshot) {  
 filePath.getDownloadUrl().addOnSuccessListener(*new* OnSuccessListener<Uri>() {  
 @Override  
 *public void* onSuccess(Uri uri) {  
 Map newImage = *new* HashMap();  
 newImage.put("profileImageURL", uri.toString());  
 DocumentReference docRef = db.collection("users").document(user\_id);  
 docRef.update("profileImageURL", uri.toString())  
 .addOnSuccessListener(*new* OnSuccessListener<Void>() {  
 @Override  
 *public void* onSuccess(Void aVoid) {  
 Log.d(TAG, "DocumentSnapshot successfully updated!");  
 }  
 });  
 *return*;  
 }  
 }).addOnFailureListener(*new* OnFailureListener() {  
 @Override  
 *public void* onFailure(@NonNull Exception exception) {  
 finish();  
 *return*;  
 }  
 });  
 }  
 });  
 }  
 *else* {  
 finish();  
 }  
 }  
 @Override  
 *protected void* onStart() {  
 *super*.onStart();  
 }  
  
 @Override  
 *protected void* onStop() {  
 *super*.onStop();  
 }  
}

*package* com.csee5590.lab3;  
  
*import* android.app.ProgressDialog;  
*import* android.content.Intent;  
*import* android.content.pm.PackageManager;  
*import* android.graphics.Bitmap;  
*import* android.graphics.BitmapFactory;  
*import* android.net.Uri;  
*import* android.os.Bundle;  
*import* android.provider.MediaStore;  
*import* android.support.annotation.NonNull;  
*import* android.support.v4.app.ActivityCompat;  
*import* android.support.v7.app.AppCompatActivity;  
*import* android.support.v7.widget.Toolbar;  
*import* android.util.Log;  
*import* android.view.*Menu*;  
*import* android.view.*MenuItem*;  
*import* android.support.v7.widget.GridLayout;  
*import* android.widget.ImageView;  
*import* android.widget.Toast;  
  
*import* com.google.android.gms.tasks.*Continuation*;  
*import* com.google.android.gms.tasks.*OnCompleteListener*;  
*import* com.google.android.gms.tasks.*OnFailureListener*;  
*import* com.google.android.gms.tasks.*OnSuccessListener*;  
*import* com.google.android.gms.tasks.Task;  
*import* com.google.firebase.firestore.FirebaseFirestore;  
*import* com.google.firebase.firestore.QueryDocumentSnapshot;  
*import* com.google.firebase.firestore.QuerySnapshot;  
*import* com.google.firebase.storage.FirebaseStorage;  
*import* com.google.firebase.storage.*OnProgressListener*;  
*import* com.google.firebase.storage.StorageReference;  
*import* com.google.firebase.storage.UploadTask;  
  
*import* java.io.ByteArrayOutputStream;  
*import* java.util.HashMap;  
*import* java.util.*Map*;  
*import* java.util.UUID;  
  
*public class* PhotosActivity *extends* AppCompatActivity {  
  
 *private static final int* REQUEST\_IMAGE\_CAPTURE = 1001;  
 *private static final int* CAMERA\_REQUEST\_CODE = 1002;  
 FirebaseStorage mFirebaseStorage;  
 StorageReference mStorageReference;  
 *private* FirebaseFirestore mFirebaseFirestore;  
 GridLayout mGridLayout;  
  
 @Override  
 *protected void* onCreate(Bundle savedInstanceState) {  
 *super*.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_photos);  
 mGridLayout = findViewById(R.id.grid\_layout);  
 Toolbar toolbar = findViewById(R.id.toolbar);  
 setSupportActionBar(toolbar);  
  
 mFirebaseStorage = FirebaseStorage.*getInstance*();  
 mStorageReference = mFirebaseStorage.getReference();  
 mFirebaseFirestore = FirebaseFirestore.*getInstance*();  
  
 mFirebaseFirestore.collection("images").get().addOnCompleteListener(*new* OnCompleteListener<QuerySnapshot>() {  
 @Override  
 *public void* onComplete(@NonNull Task<QuerySnapshot> task) {  
 *if* (task.isSuccessful()) {  
 *for* (QueryDocumentSnapshot document : task.getResult()) {  
 Log.*d*("PHOTO", document.getString("url"));  
 downloadImage(document.getString("url"));  
 }  
 }  
 }  
 });  
 }  
  
 *private void* downloadImage(*final* String url) {  
 StorageReference reference = mFirebaseStorage.getReferenceFromUrl(url);  
 *final long* ONE\_MEGABYTE = 1024 \* 1024;  
 reference.getBytes(ONE\_MEGABYTE).addOnSuccessListener(*new* OnSuccessListener<*byte*[]>() {  
 @Override  
 *public void* onSuccess(*byte*[] data) {  
 ImageView imageView = *new* ImageView(PhotosActivity.*this*);  
 Bitmap bitmap = BitmapFactory.*decodeByteArray*(data, 0, data.length);  
 imageView.setImageBitmap(bitmap);  
 mGridLayout.addView(imageView);  
 *// Data for "images/island.jpg" is returns, use this as needed* }  
 });  
 }  
  
 @Override  
 *public boolean* onCreateOptionsMenu(*Menu* menu) {  
 *// Inflate the menu; this adds items to the action bar if it is present.* getMenuInflater().inflate(R.menu.menu\_photos, menu);  
 *return true*;  
 }  
  
 @Override  
 *public boolean* onOptionsItemSelected(*MenuItem* item) {  
 *if* (item.getItemId() == R.id.action\_add\_photo) {  
 *// User chose the "Add Photo" item, show the camera* showCamera();  
 *return true*;  
 }  
  
 *// If we got here, the user's action was not recognized.  
 // Invoke the superclass to handle it.  
 return super*.onOptionsItemSelected(item);  
 }  
  
 *private void* showCamera() {  
 *if* (ActivityCompat.*checkSelfPermission*(*this*, android.Manifest.permission.CAMERA) != PackageManager.PERMISSION\_GRANTED) {  
 ActivityCompat.*requestPermissions*(*this*, *new* String[]{android.Manifest.permission.CAMERA}, CAMERA\_REQUEST\_CODE);  
 } *else* {  
 Intent takePictureIntent = *new* Intent(MediaStore.ACTION\_IMAGE\_CAPTURE);  
 *if* (takePictureIntent.resolveActivity(getPackageManager()) != *null*) {  
 startActivityForResult(takePictureIntent, REQUEST\_IMAGE\_CAPTURE);  
 }  
 }  
 }  
  
 @Override  
 *public void* onActivityResult(*int* requestCode, *int* resultCode, Intent data) {  
 *if* (requestCode == REQUEST\_IMAGE\_CAPTURE && resultCode == RESULT\_OK) {  
 *final* ProgressDialog progressDialog = *new* ProgressDialog(*this*);  
 progressDialog.setTitle("Uploading...");  
 progressDialog.show();  
 Bitmap bitmap = (Bitmap) data.getExtras().get("data");  
 ByteArrayOutputStream baos = *new* ByteArrayOutputStream();  
 bitmap.compress(Bitmap.CompressFormat.JPEG, 100, baos);  
 *byte*[] bitmapData = baos.toByteArray();  
  
 *final* String uuid = UUID.randomUUID().toString();  
 *final* StorageReference ref = mStorageReference.child("images/" + uuid);  
 ref.putBytes(bitmapData).addOnSuccessListener(*new* OnSuccessListener<UploadTask.TaskSnapshot>() {  
 @Override  
 *public void* onSuccess(UploadTask.TaskSnapshot taskSnapshot) {  
 progressDialog.dismiss();  
 Toast.makeText(PhotosActivity.*this*, "Uploaded", Toast.LENGTH\_SHORT).show();  
 }  
 }).addOnFailureListener(*new* OnFailureListener() {  
 @Override  
 *public void* onFailure(@NonNull Exception e) {  
 progressDialog.dismiss();  
 Toast.makeText(PhotosActivity.*this*, "Failed " + e.getMessage(), Toast.LENGTH\_SHORT).show();  
 }  
 }).addOnProgressListener(*new* OnProgressListener<UploadTask.TaskSnapshot>() {  
 @Override  
 *public void* onProgress(UploadTask.TaskSnapshot taskSnapshot) {  
 *double* progress = (100.0 \* taskSnapshot.getBytesTransferred() / taskSnapshot  
 .getTotalByteCount());  
 progressDialog.setMessage("Uploaded " + (*int*) progress + "%");  
 }  
 }).continueWithTask(*new* Continuation<UploadTask.TaskSnapshot, Task<Uri>>() {  
 @Override  
 *public* Task<Uri> then(@NonNull Task<UploadTask.TaskSnapshot> task) *throws* Exception {  
 *if* (!task.isSuccessful()) {  
 *throw* task.getException();  
 }  
  
 *// Continue with the task to get the download URL  
 return* ref.getDownloadUrl();  
 }  
 }).addOnCompleteListener(*new* OnCompleteListener<Uri>() {  
 @Override  
 *public void* onComplete(@NonNull Task<Uri> task) {  
 *if* (task.isSuccessful()) {  
 Uri downloadUri = task.getResult();  
 Map<String, String> doc = *new* HashMap<>();  
 String url = downloadUri.toString();  
 doc.put("url", url);  
 mFirebaseFirestore.collection("images").add(doc);  
 downloadImage(url);  
 } *else* {  
 *// Handle failures  
 // ...* }  
 }  
 });  
 }  
 }  
  
 @Override  
 *public void* onRequestPermissionsResult(*int* requestCode, @NonNull String[] permissions, @NonNull *int*[] grantResults) {  
 *if* (requestCode == CAMERA\_REQUEST\_CODE  
 && grantResults.length > 0  
 && grantResults[0] == PackageManager.PERMISSION\_GRANTED) {  
 showCamera();  
 }  
 }  
}

*package* com.csee5590.lab3;  
  
*import* android.content.Intent;  
*import* android.os.Bundle;  
*import* android.support.annotation.NonNull;  
*import* android.support.v7.app.AppCompatActivity;  
*import* android.support.v7.widget.Toolbar;  
*import* android.util.Patterns;  
*import* android.view.MotionEvent;  
*import* android.view.View;  
*import* android.view.*Menu*;  
*import* android.view.*MenuItem*;  
*import* android.view.animation.Animation;  
*import* android.view.animation.AnimationUtils;  
*import* android.widget.Button;  
*import* android.widget.EditText;  
*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.*AuthResult*;  
*import* com.google.firebase.auth.FirebaseAuth;  
*import* com.google.firebase.auth.FirebaseUser;  
*import* com.google.firebase.firestore.DocumentReference;  
*import* com.google.firebase.firestore.FirebaseFirestore;  
  
*import* java.util.HashMap;  
*import* java.util.*Map*;  
  
*//AppCompatActivity:  
public class* SignUpActivity *extends* AppCompatActivity {  
 TextView errorText;  
 *private* FirebaseAuth mAuth;  
 *private* EditText mEmail, mPassword, mName, mUniversity, mAge, mPhone, mMajor;  
 *private* Button mSignUp, mCancel;  
 *private* FirebaseAuth.*AuthStateListener* firebaseAuthListener;  
 String user\_id;  
 *private* FirebaseFirestore db;  
 @Override  
 *protected void* onCreate(Bundle savedInstanceState) {  
 *super*.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_sign\_up);  
 Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);  
 setSupportActionBar(toolbar);  
 mAuth = FirebaseAuth.*getInstance*();  
 db = FirebaseFirestore.*getInstance*();  
 firebaseAuthListener = *new* FirebaseAuth.AuthStateListener() {  
 @Override  
 *public void* onAuthStateChanged(@NonNull FirebaseAuth firebaseAuth) {  
 FirebaseUser user = FirebaseAuth.*getInstance*().getCurrentUser(); *//get the information of the current user  
 if* (user != *null*){  
 *//if not null, move to another activity, to be created later.  
 //Remember the current context!* Intent intent = *new* Intent(SignUpActivity.*this*, MainActivity.*class*);  
 intent.putExtra("user\_id", user.getUid());  
 intent.putExtra("user\_email", user.getEmail());  
 startActivity(intent);  
 finish();  
 }  
 }  
 };  
 mSignUp = (Button)findViewById(R.id.btn\_signup);  
 mEmail = (EditText)findViewById(R.id.txt\_uname);  
 mPassword = (EditText) findViewById(R.id.txt\_Pwd);  
 mName = (EditText) findViewById(R.id.txt\_name);  
 mPhone = (EditText) findViewById(R.id.txt\_phone);  
 mUniversity = (EditText) findViewById(R.id.txt\_university);  
 mAge = (EditText) findViewById(R.id.txt\_age);  
 mMajor = (EditText) findViewById(R.id.txt\_major);  
 errorText = (TextView)findViewById(R.id.lbl\_Error);  
 mCancel = (Button)findViewById(R.id.btnCancel);  
  
 mCancel.setOnClickListener(*new* View.OnClickListener() {  
 @Override  
 *public void* onClick(View v) {  
 Intent intent = *new* Intent(SignUpActivity.*this*, WelcomeActivity.*class*);  
 startActivity(intent);  
 finish();  
 }  
 });  
  
 */\*\*  
 \* This function will make sure that the error message disappears  
 \* before user retypes the username/password  
 \*/* mEmail.setOnTouchListener(*new* View.OnTouchListener() {  
 @Override  
 *public boolean* onTouch(View view, MotionEvent event) {  
 *switch*(event.getAction()) {  
 *case* MotionEvent.ACTION\_DOWN:  
 mEmail.requestFocus();  
 errorText.setVisibility(View.GONE);  
 *// PRESSED  
 return true*; *// if you want to handle the touch event  
 case* MotionEvent.ACTION\_UP:  
 *// RELEASED  
 return true*; *// if you want to handle the touch event* }  
 *return false*;  
 }  
 });  
 mPassword.setOnTouchListener(*new* View.OnTouchListener() {  
 @Override  
 *public boolean* onTouch(View view, MotionEvent event) {  
 *switch*(event.getAction()) {  
 *case* MotionEvent.ACTION\_DOWN:  
 mPassword.requestFocus();  
 errorText.setVisibility(View.GONE);  
 *// PRESSED  
 return true*; *// if you want to handle the touch event  
 case* MotionEvent.ACTION\_UP:  
 *// RELEASED  
 return true*; *// if you want to handle the touch event* }  
 *return false*;  
 }  
 });  
 mSignUp.setOnClickListener(*new* View.OnClickListener() {  
 @Override  
 *public void* onClick(View v) {  
 checkCredentials(v);  
 }  
 });  
 }  
  
 @Override  
 *public boolean* onCreateOptionsMenu(Menu menu) {  
 *// Inflate the menu; this adds items to the action bar if it is present.* getMenuInflater().inflate(R.menu.menu\_login, menu);  
 *return true*;  
 }  
  
 @Override  
 *public boolean* onOptionsItemSelected(MenuItem item) {  
 *// Handle action bar item clicks here. The action bar will  
 // automatically handle clicks on the Home/Up button, so long  
 // as you specify a parent activity in AndroidManifest.xml.  
 int* id = item.getItemId();  
  
 *//noinspection SimplifiableIfStatement  
 if* (id == R.id.action\_settings) {  
 *return true*;  
 }  
  
 *return super*.onOptionsItemSelected(item);  
 }  
 *//This is where the identity check is. The main variable is the validation flag. If it is false, then the user  
 //will not go anywhere beyond the login page. Else if the user enters the right username and password, he will be directed  
 //to the next page. Username and password are determined by the 2 variables "userName" and "password"  
 public void* checkCredentials(View v)  
 {  
 *final* String email = mEmail.getText().toString();  
 *final* String password = mPassword.getText().toString();  
 *final* String age = mAge.getText().toString();  
 *final* String phone = mPhone.getText().toString();  
 *final* String university = mUniversity.getText().toString();  
 *final* String name = mName.getText().toString();  
 *final* String major = mMajor.getText().toString();  
 *if*(email.isEmpty()){  
 mEmail.setError("Email Address cannot be blank");  
 }  
 *else if*(!Patterns.EMAIL\_ADDRESS.matcher(email).matches()){  
 mEmail.setError("Enter a Valid Email");  
 mEmail.requestFocus();  
 }  
 *else if*(password.isEmpty()){  
 mPassword.setError("Password cannot be blank");  
 mPassword.requestFocus();  
 }  
 *else if*(password.length()<6 || password.length()>12){  
 mPassword.setError("Password should be of minimum 6 characters and maximum 12 characters");  
 mPassword.requestFocus();  
 }  
 *else if*(name.isEmpty()){  
 mName.setError("Name cannot be blank");  
 mName.requestFocus();  
 }  
 *else if*(age.isEmpty()){  
 mAge.setError("Age cannot be blank");  
 mAge.requestFocus();  
 }  
 *else if*(university.isEmpty()){  
 mUniversity.setError("University cannot be blank");  
 mUniversity.requestFocus();  
 }  
 *else if*(phone.isEmpty()){  
 mPhone.setError("Phone cannot be blank");  
 mPhone.requestFocus();  
 }  
 *else if*(major.isEmpty()){  
 mMajor.setError("Major cannot be blank");  
 mMajor.requestFocus();  
 }  
 *else* {  
 mAuth.createUserWithEmailAndPassword(email, password).addOnCompleteListener(SignUpActivity.*this*, *new* OnCompleteListener<AuthResult>() {  
 @Override  
 *public void* onComplete(@NonNull Task<AuthResult> task) {  
 *//If user has already signed up, therefore the createUserWithEmailandPassword would fail.  
 if* (!task.isSuccessful()) {  
 Toast.makeText(SignUpActivity.*this*, "Registration Failed!", Toast.LENGTH\_SHORT).show();  
 }  
 *//If the user email cannot be found in the database, reference the database and add variables to it.  
 else* {  
 user\_id = mAuth.getCurrentUser().getUid(); *//id assigned to Technician at moment of sign-up  
  
 ///Using FireStore* DocumentReference docRef = db.collection("users").document(user\_id);  
 *// Add document data with string user\_id using a hashmap* Map<String, Object> data = *new* HashMap<>();  
 data.put("email", email);  
 data.put("name", name);  
 data.put("phone", phone);  
 data.put("age", age);  
 data.put("university", university);  
 data.put("major", major);  
 *//asynchronously write data* docRef.set(data);  
 }  
 }  
 });  
 }  
 }  
  
 @Override  
 *protected void* onStart() {  
 *super*.onStart();  
 mAuth.addAuthStateListener(firebaseAuthListener);  
 }  
  
 @Override  
 *protected void* onStop() {  
 *super*.onStop();  
 mAuth.removeAuthStateListener(firebaseAuthListener);  
 }  
}

*package* com.csee5590.lab3;  
  
*public class* User {  
 String name;  
 String email;  
 String age;  
 String phone;  
 String university;  
 String major;  
  
 *public* User(String Email, String Name, String Age, String Phone, String Uni, String Major) {  
 name = Name;  
 email = Email;  
 phone = Phone;  
 age = Age;  
 university = Uni;  
 major = Major;  
 }  
 *public* String getEmail(){ *return* email;}  
  
 *public* String getName() {  
 *return* name;  
 }  
  
 *public* String getPhone() {  
 *return* phone;  
 }  
  
 *public* String getAge() {  
 *return* age;  
 }  
  
 *public* String getUniversity (){*return* university;}  
  
 *public* String getMajor () {*return* major;}  
}

*package* com.csee5590.lab3;  
  
*import* android.content.Intent;  
*import* android.support.v7.app.AppCompatActivity;  
*import* android.os.Bundle;  
*import* android.view.View;  
*import* android.widget.Button;  
  
*import* java.text.ParseException;  
*import* java.text.SimpleDateFormat;  
*import* java.util.Calendar;  
*import* java.util.Date;  
*import* java.util.TimeZone;  
  
*public class* WelcomeActivity *extends* AppCompatActivity {  
 *private* Button mSignUp, mSignIn, mGSignIn;  
 @Override  
 *protected void* onCreate(Bundle savedInstanceState) {  
 *super*.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_welcome);  
*// Basic Sign-in/Sign-up Button Layout* mSignUp = (Button) findViewById(R.id.signup\_btn);  
 mSignIn = (Button) findViewById(R.id.signin\_btn);  
 mGSignIn = (Button) findViewById(R.id.signin\_gg\_btn);  
 mSignUp.setOnClickListener(*new* View.OnClickListener() {  
 @Override  
 *public void* onClick(View v) {  
 Intent intent = *new* Intent(WelcomeActivity.*this*, SignUpActivity.*class*);  
 startActivity(intent);  
 finish();  
 }  
 });  
 mSignIn.setOnClickListener(*new* View.OnClickListener() {  
 @Override  
 *public void* onClick(View v) {  
 Intent intent = *new* Intent(WelcomeActivity.*this*, LoginActivity.*class*);  
 startActivity(intent);  
 finish();  
 }  
 });  
 mGSignIn.setOnClickListener(*new* View.OnClickListener() {  
 @Override  
 *public void* onClick(View v) {  
 Intent intent = *new* Intent(WelcomeActivity.*this*, GoogleSignInActivity.*class*);  
 startActivity(intent);  
 finish();  
 }  
 });  
 }  
  
}