שם הפרויקט - Get Sitter.

חוזר מנהל מה"ט 51-4-11 – נספח מס '1 (הצעה לפרויקט גמר)

תאריך:\_\_\_\_\_\_\_3.6.19\_\_\_\_

לכבוד

יחידת הפרויקטים מה"ט

**הצעה לפרויקט גמר**

\* יש להדפיס את כל הנתונים הנדרשים

**א.** **פרטי הסטודנטים**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| תאריך סיום  הלימודים | טלפון נייד | כתובת | ת.ז .9 ספרות | שם הסטודנט |
| ספטמבר 2019 | 052-6656514 | הקליר 7 , נתניה | 20583615 | רועי סעדון |
| ספטמבר 2019 | 054-3255182 | רבי אלימלך 7 , כפר יונה | **205500051** | אורם עבו |

שם המכללה המרכז הטכנולוגי אריאל סמל המכללה: 72211

מסלול ההכשרה: הנדסאים

מגמת לימוד: הנדסת תוכנה וסייבר

מקום ביצוע הפרויקט:אריאל

**ב.** **פרטי המנחה האישי**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| מקום עבודה/תפקיד | תואר | טלפון נייד | כתובת | שם המנחה \* |
| רמ"ח הנדסאי תוכנה אריאל |  | 0527291300 |  | אפי פרופוס |

\* עבור מנחה אישי חדש יש לצרף קורות חיים, ניסיון מקצועי ותעודות השכלה לאישור מה"ט.

\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ חתימת הסטודנט חתימת המנחה האישי חתימת הגורם המקצועי מטעם מה"ט

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ חתימת הסטודנט חתימת המנחה האישי

**ג.** **הערות ראש המגמה במכללה**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ד.** **אישור ראש המגמה**

שם: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ חתימה: \_\_\_\_\_\_\_\_\_\_\_\_ תאריך:\_\_\_\_\_\_\_\_\_\_\_\_\_

**ה.** **הערות הגורם המקצועי מטעם מה"ט**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ו.** **אישור הגורם המקצועי מטעם מה"ט**

שם: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ חתימה: \_\_\_\_\_\_\_\_\_\_\_\_ תאריך:\_ \_\_\_\_\_\_\_\_\_\_\_\_

**2) רקע -** האפליקציה באה להנגיש מגוון של שירותים , שידועים כמסורבלים למציאה כמו בייביסיטר או דוגווקינג.

בעידן בו הטכנולוגיה שולטת, הבנו שלא ייתכן מצב שבו יש קושי כה גדול למצוא בייביסיטר, בזמן שמנגד ישנם המון אנשים טובים שישמחו לעזור אך לא יפגשו לעולם.

בהתחלה זה נשמע קצת הזוי ? איך נכניס מישהו לטפל בילדים מבלי שאנחנו מכירים אותו ?

פתרון פשוט, בעזרת מערכת אימותים שתחייב כל משתמש שרוצה לעסוק בבייביסיטינג, להעלות תמונת סלפי עם תאריך עדכני ומספר המשתמש שלו, קישור לפייסבוק שלו , וצילום תעודת זהות ורק אז יוכל להיות מאושר לתת שירות בפלטפורמה.

מטרתינו היא שכל אדם יוכל להזמין מעכשיו לעכשיו שירותים אלו , מצד אחד במהירות , ומצד שני גם בידיעה שהאדם הנמצא אצלו בבית מאומת , ולחסוך הרבה כאב ראש להרבה אנשים, ומנגד גם להשלים הכנסה לאלו שמחפשים זאת.

**בהתחלה נתמקד אך ורק בבייביסיטינג כדי להיות ממוקדים יותר, ועם הדרישה שתעלה נרחיב גם את מיספר הפיצ'רים ונכניס דוגווקינג, מורה פרטי, שירותי ניקיון ואוכל ביתי.**

**3) מצב קיים בשוק -** ישנן מספר אפליקציות שמנסות לספר מטרות דומות, אך הן פחות ממוקדות ויורות לכל כיוון. האפליקציות איטיות ונתקעות, עם המון הורדות מה שמראה על שוק פוטנציאל גדול, ואם מתעמקים בתגובות של האפליקציות האלו מגלים שרוב הדרוגים הם נמוכים מאוד ושאנשים לא מרוצים מהאפליקציות כלל, אנו נשפר וניישם אצלנו את הטעויות של האחרים, ונשאף ליצור אפליקציה מוגמרת ברמה גבוהה ובעלת נוחות משתמש מירבית.

בלי יותר מידי שירותים מיותרים שמסרבלים את כל העסק, רק האלו **שמאוד** קשה למצוא, וחבל.

**4) מה הפרויקט בא לחדש ולשפר ? -** הפרויקט בא לתת מענה לצורך קיים , שעדיין לא קיבל מענה הולם.

לשפר את חווית המשתמש מהאפליקציות המסובכות והאיטיות,

חווית המשתמש ב**כל** האפליקציות הדומות מסורבלת, וכדי להגיע למצב של שימוש צריך לעבור הרבה מסכים שונים, להכניס המון פרטים , הרבה מהם בכלל לא רלוונטים, ובסוף האפליקציה קורסת.

לא בבית ספרנו .

אנחנו מתבדלים מהאפליקציות הדומות בצ'אט בין המשתמשים, אימותים של נותני שירות, ראיון ראשוני בפלטפורמה עם נותן השירות, ומערכת ביקורות שתספק מידע ישיר על איכויות נותן השירות.

**5) דרישות מערכת -**

טלפון חכם

**מערכת הפעלה:** אנדרואיד, API 23 ומעלה .

**דרישות משתמש מהמערכת -** הרשאות מיקום, הרשאות לשמירת קבצים, הרשאות לתמונות, הרשאת קבלת התראות מהשרת, נדרש חיבור לרשת אינטרנט .

**6) בעיות הצפויות במהלך הפיתוח -**

* בעיות תקשורת ברשת האינטרנט.
* בעיות של העברות מידע בין מכשירים בפורמטים שונים ומגוונים

**7) פתרון טופולוגי נבחר -**

* שרת : Node JS מאוחסן באמזון - שפה נוחה לכתיבת צד השרת מתממשקת בצורה נוחה עם מאגרי המידע.
* DB : לMongoDB - מאפשר עבודה נוחה יותר עם קבצי JSON ושאילתות נוחות יותר של קישור בין טבלאות.
* צד לקוח - אפליקציה לאנדרואיד כתובה בשפת ג'אווה.



**8 .שימוש במבני נתונים וארגון קבצים:**

קובץ entities: קובץ המכיל את כל האישויות שיש במסד הנתונים, שבעזרתו אפשר לוודא שהנתונים אשר נכנסים למסד הם אכן מסוג האישיות הקיימות במסד.

region\_sch

אזור שירות של נותני שירות

role\_sch

קלאס המכיל בתוכו נתוני מגורים של משתמשים

adress\_sch

קלאס משתמש רגיל

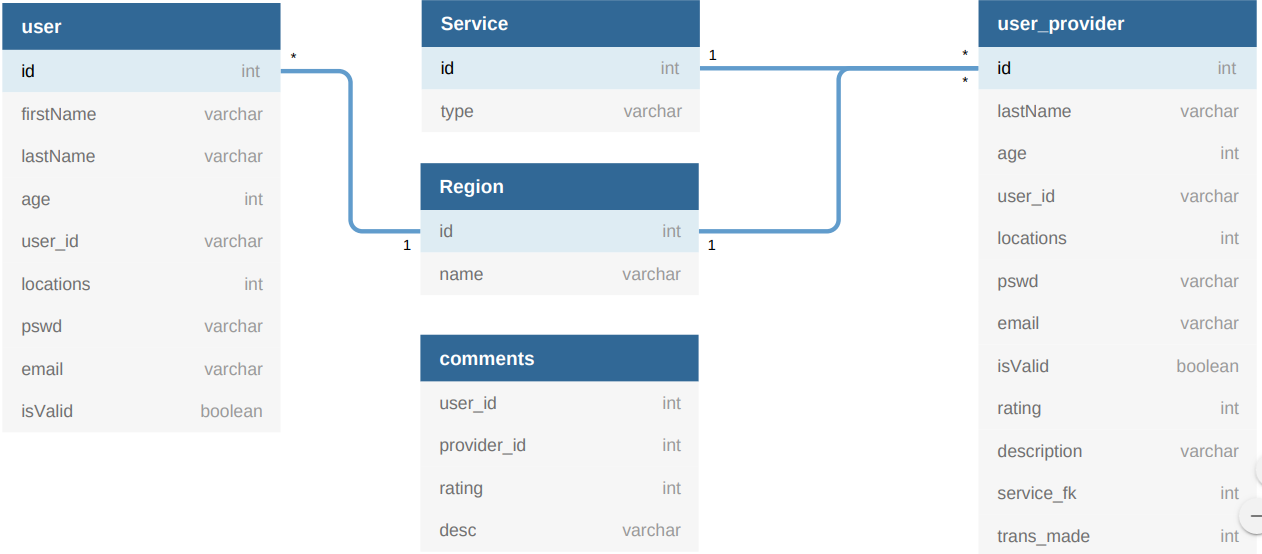
user\_sch

קלאס של נותני שירות המכיל בתוכו גם את סוגי השירות שהמשתמש נותן

user\_prov\_sch

קלאס של תגובה ששומר בתוכו את הID של המשתמשים

comment\_sch



**פונקציונליות השרת(BL):**

פונקציונליות השרת מתחלקת לפי הטבלאות בדאטה בייס:

User,user\_provider,comment,region,service

כאשר region וservice שמור אדמיניסטרציה ולא ניתן לעריכה מצד המשתמש.

user,user\_provider,comment

מכילים בתוכם פונקציות crud.

רשימת קלאסיים:

מנג'רים:user,user\_provider,comment,region,role

מנהלים את כל החיבורים של הקלאסים לדטה בייס

validateUser()

פונקציה שמשתילה בתוך המשתמש איזשהו האש לאחר אימות באימייל

IsUser()

בעת בקשה מהצד לקוח מוודא שהמשתמש קיים

OnLogin()

כאשר משתמש מתחבר מהפלאפון הקליינט שולח טוקן שמאומת דרך הפונקציה הזאת ומחזיר טוקן חדש

**ניהול לוגים:**

השרת ישמור לוגים בנושאים: קריסה של הצד שרת\לקוח, מי שלח בקשות ואיזה בקשות, מי התחבר,

עיסקאות מוצלחות בין משתמשים.

**11. תיאור/התייחסות לנושאי אבטחת מידע -** התחברות לשרת רק באמצעות מפתח ציבורי, מפתח מיוחד לכול משתמש אשר הוא צריך להעביר בכול בקשה על מנת לפנות לשרת, שמירת פרטי כרטיס אשרי באופן מוצפן.

**12. משאבים הנדרשים לפרויקט:**

מספר שעות: 500 שעות

ציוד נדרש: מחשב עם אנדרואיד סטודיו

תוכונות נדשרות: אנדרואיד סטודיו VSCODE

ידע חדש: שרת אמזון

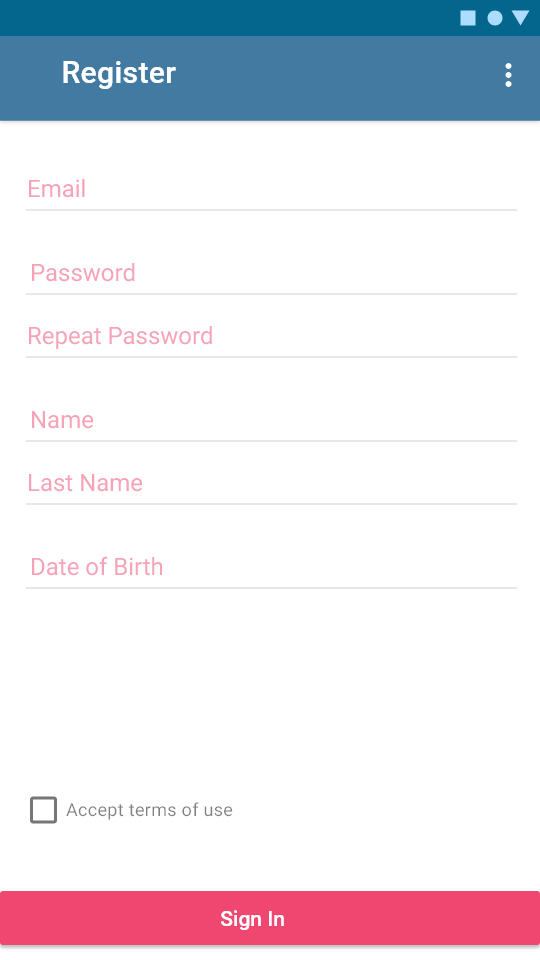
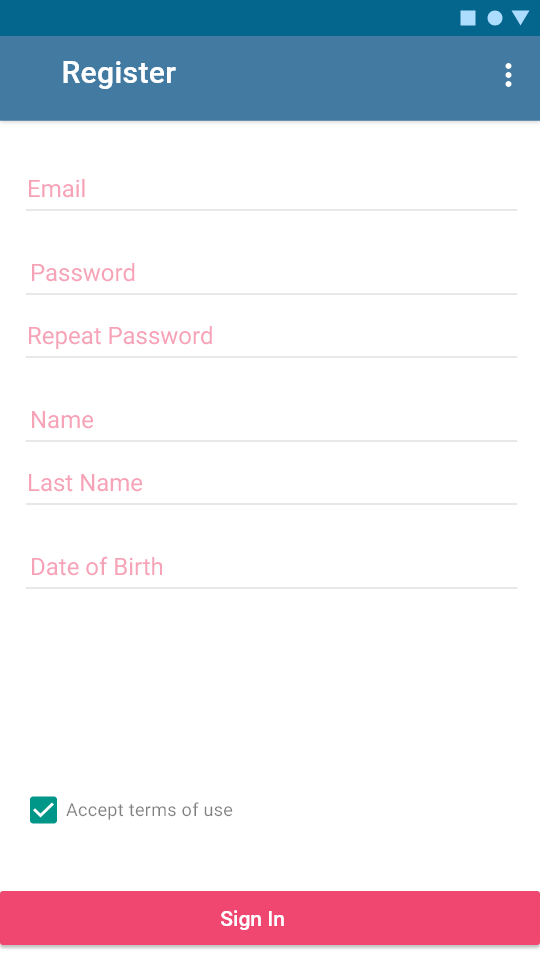
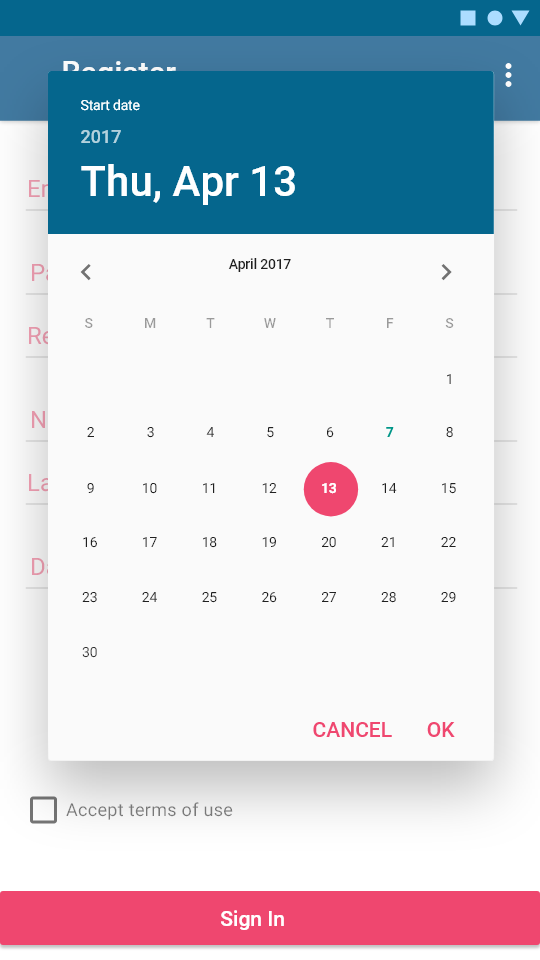
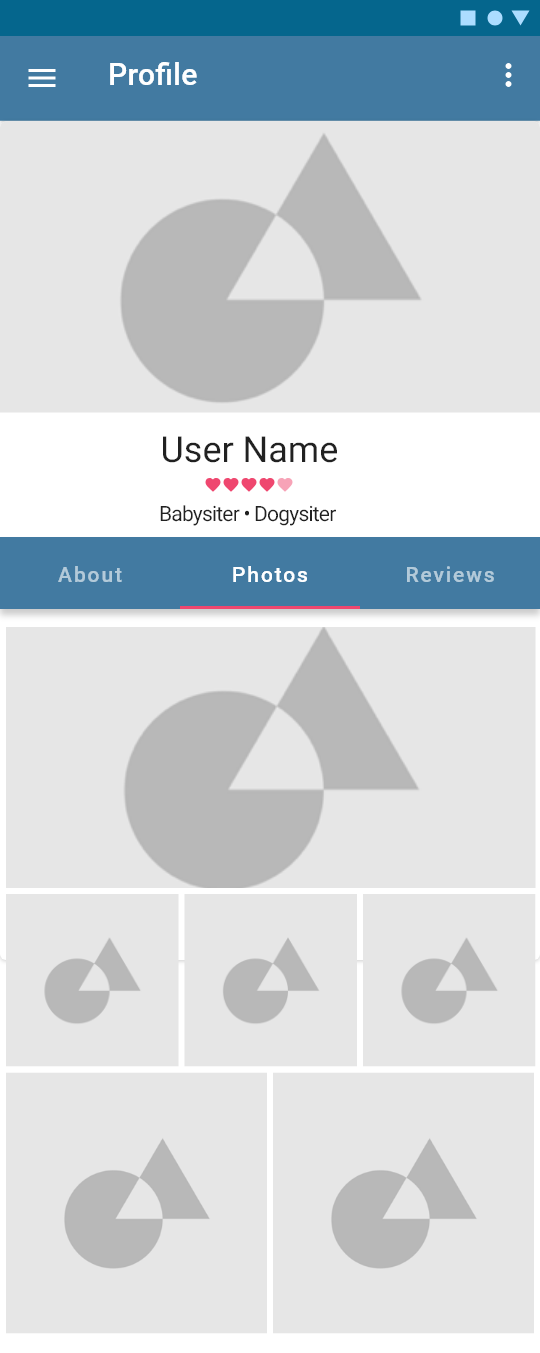
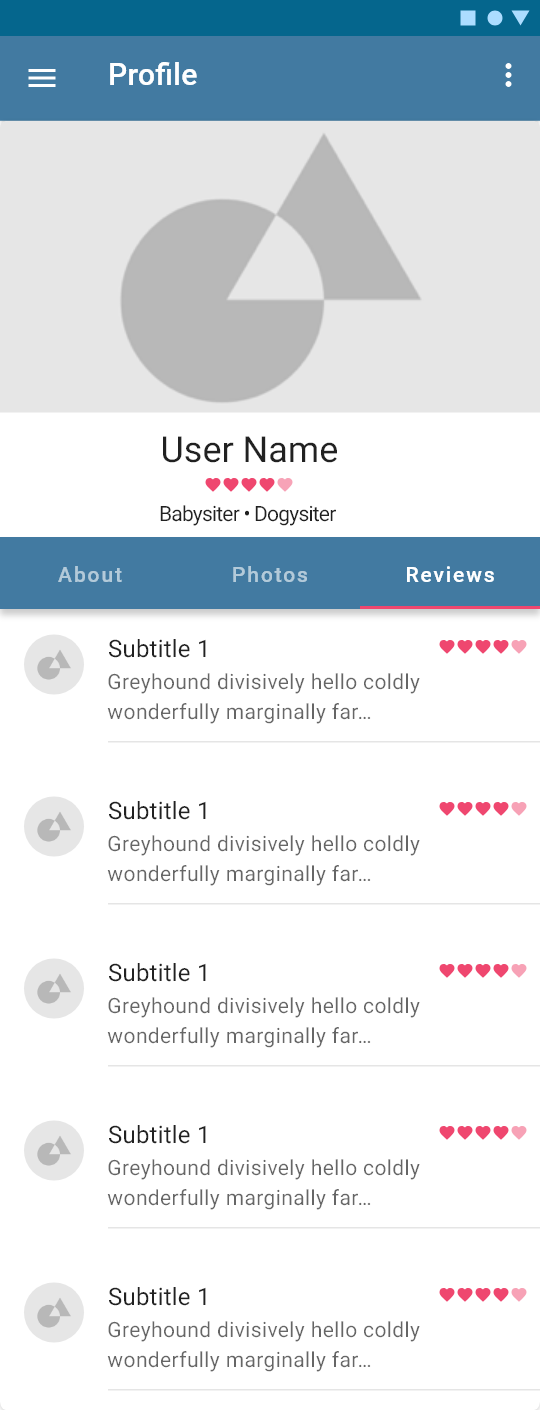
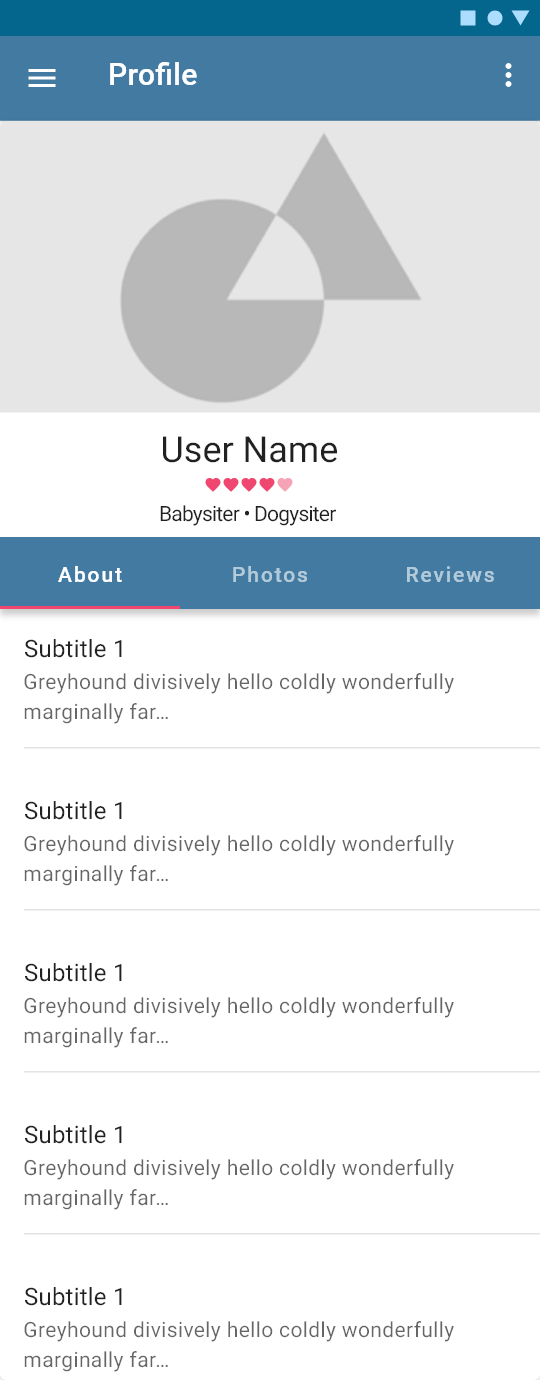
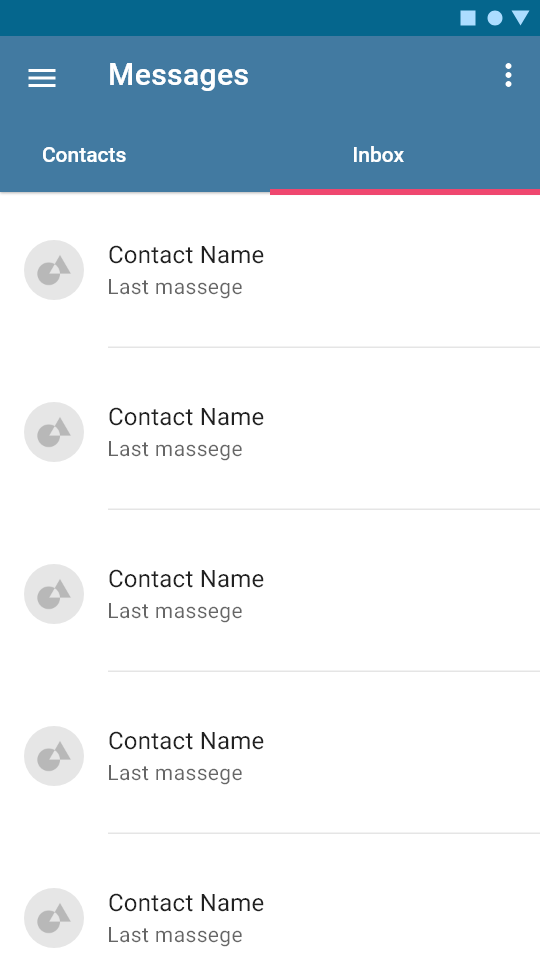
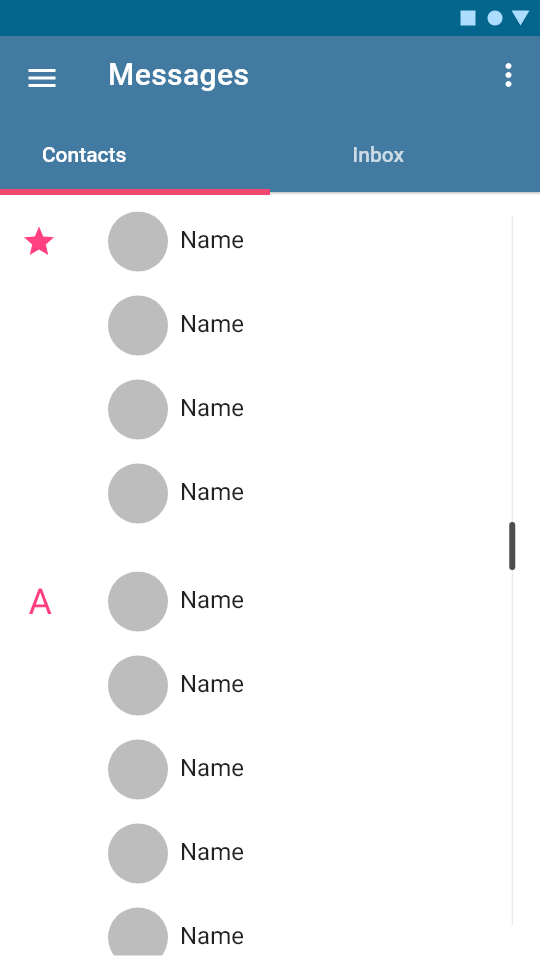
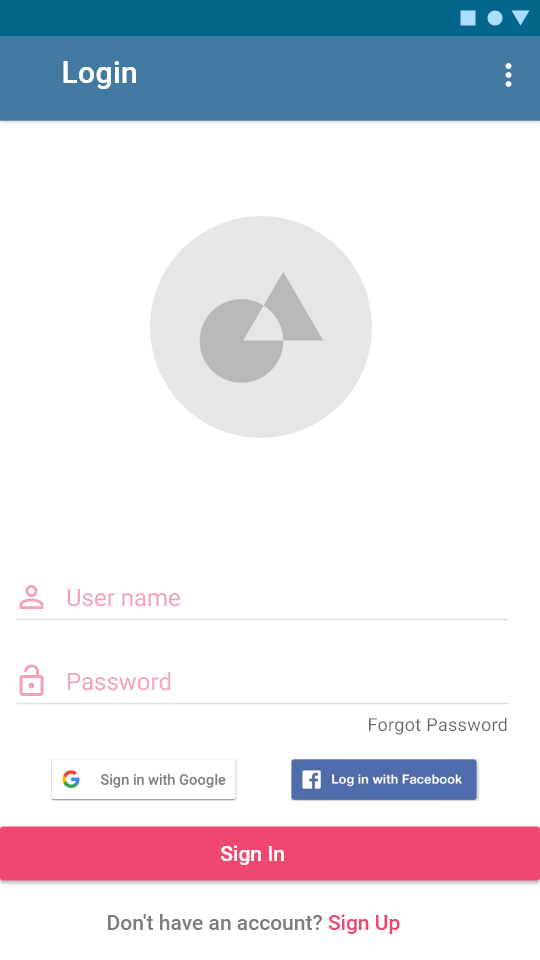
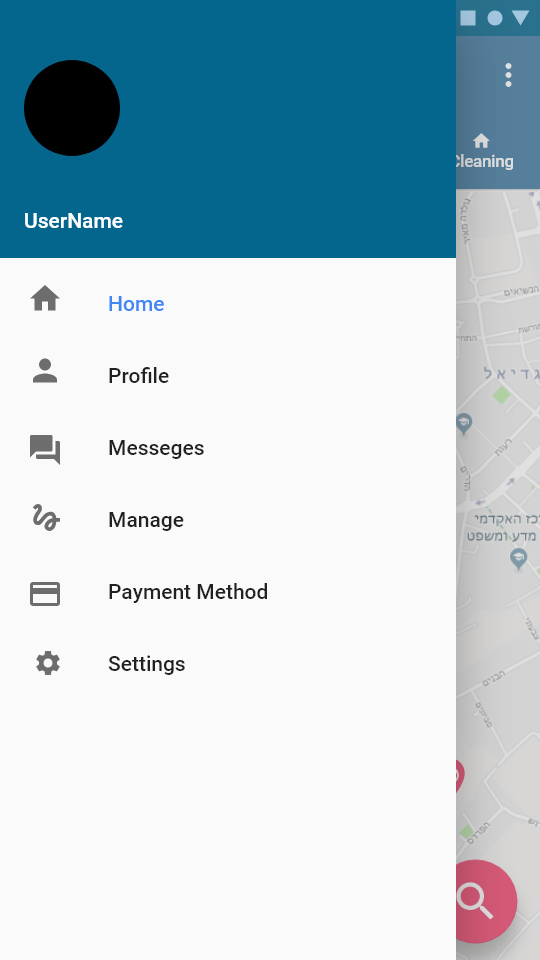
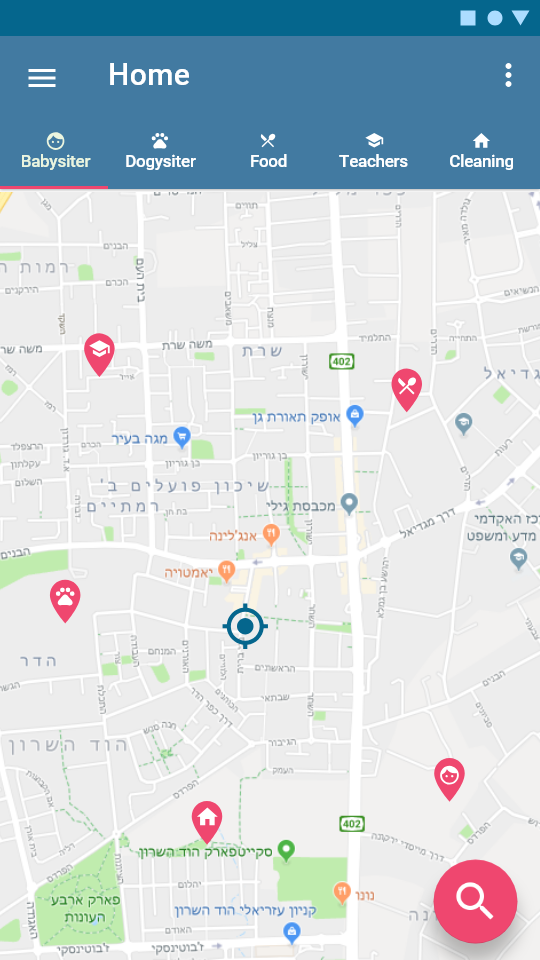
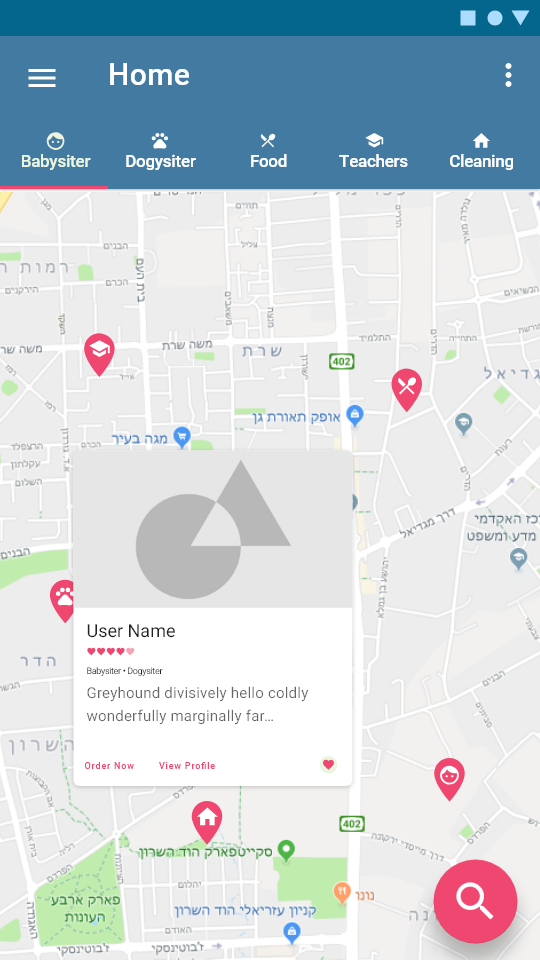
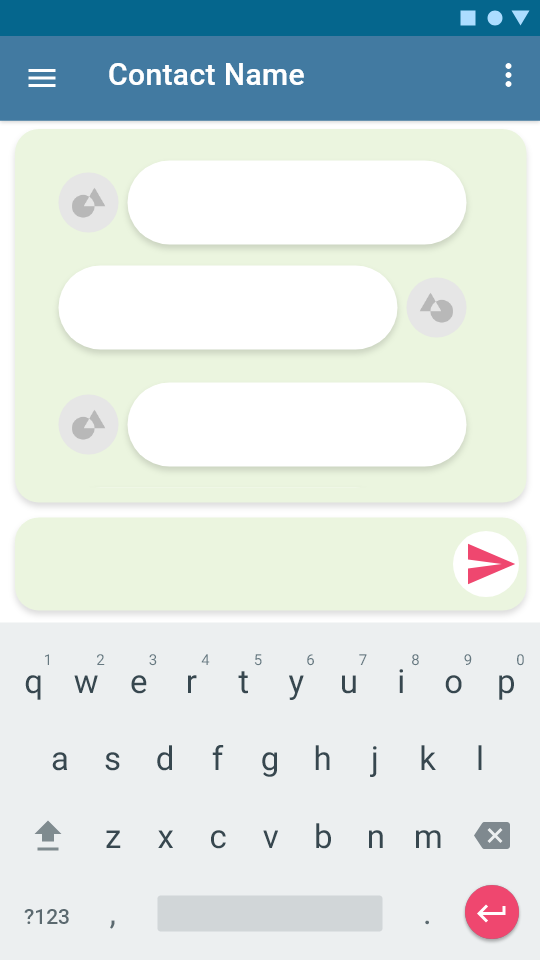
מקורות ידע: UDEMY

**13.תכנית עבודה ושלבים למימוש הפרויקט**

הקמת הDB, עיצוב וניתוח של דפי האפלקציה, בניית אםלקציה הראשונית המכילה מעבר בין הדפים,

כתיבת פונקציות בצד השרת לקבלת ושימרת מידע חדש, מעבר על הדפים וסיום כול דף עד סופו.

**15. Version control -** שימוש בGIT עם גיט האב.

****

**תרשים גאנט :**

חודש מאי - עיצוב כל הדפים הדרושים .

חודש יוני - קישור בין כל הדפים והתחלת יישום פונקציונאליות.

חודש יולי - קישור עם הDB.

חודש אוגוסט - בדיקות מערכת ראשוניות ותחילת הכנת ספר פרוייקט

חודש ספטמבר - שליחת האפליקצייה למכרים לשיפור וייעול המערכת, ולקבלת רעיונות נוספים לשיפור.

חודש אוקטובר - יישום שידרוגיי המערכת ברמה המקסימלית, סיום כתיבת ספר פרוייקט והגשת הפרוייקט.

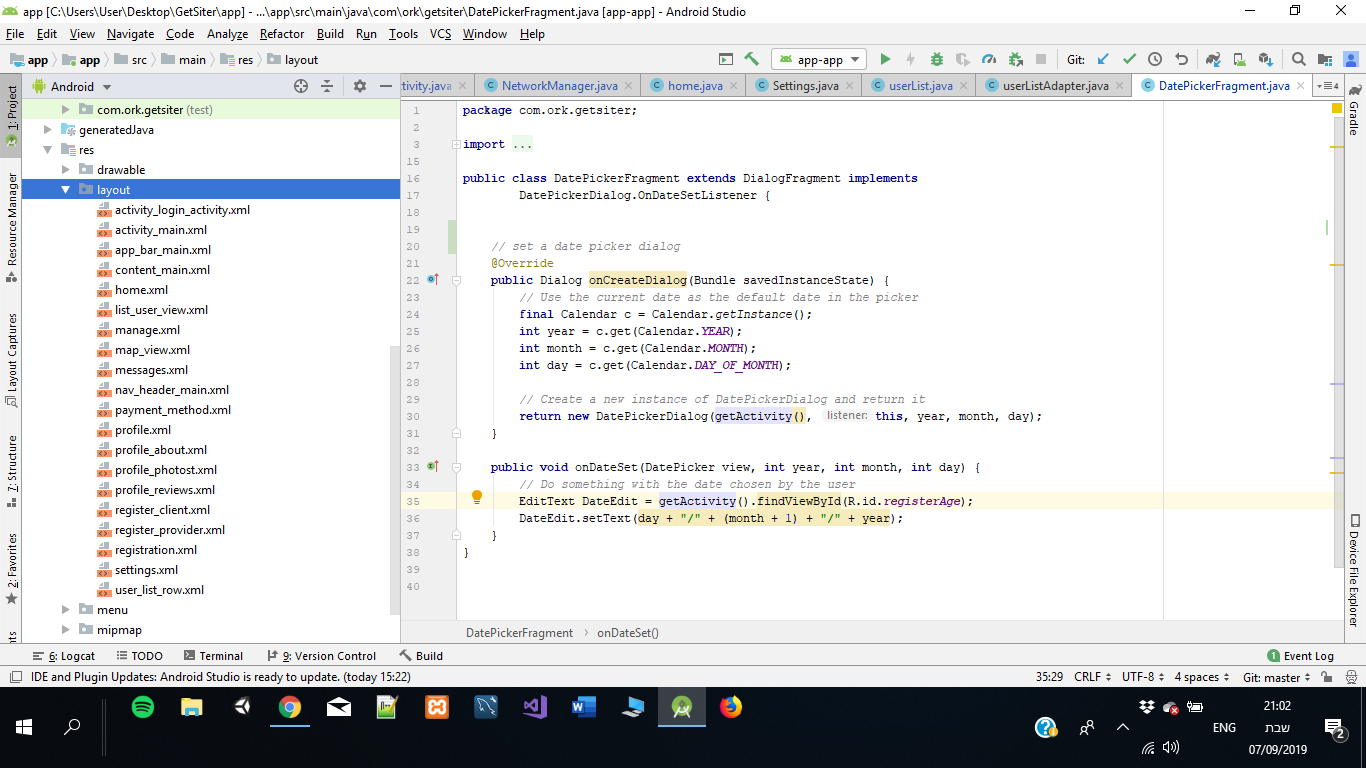
**חתימה:**

**אורם עבו: רועי סעדון:**

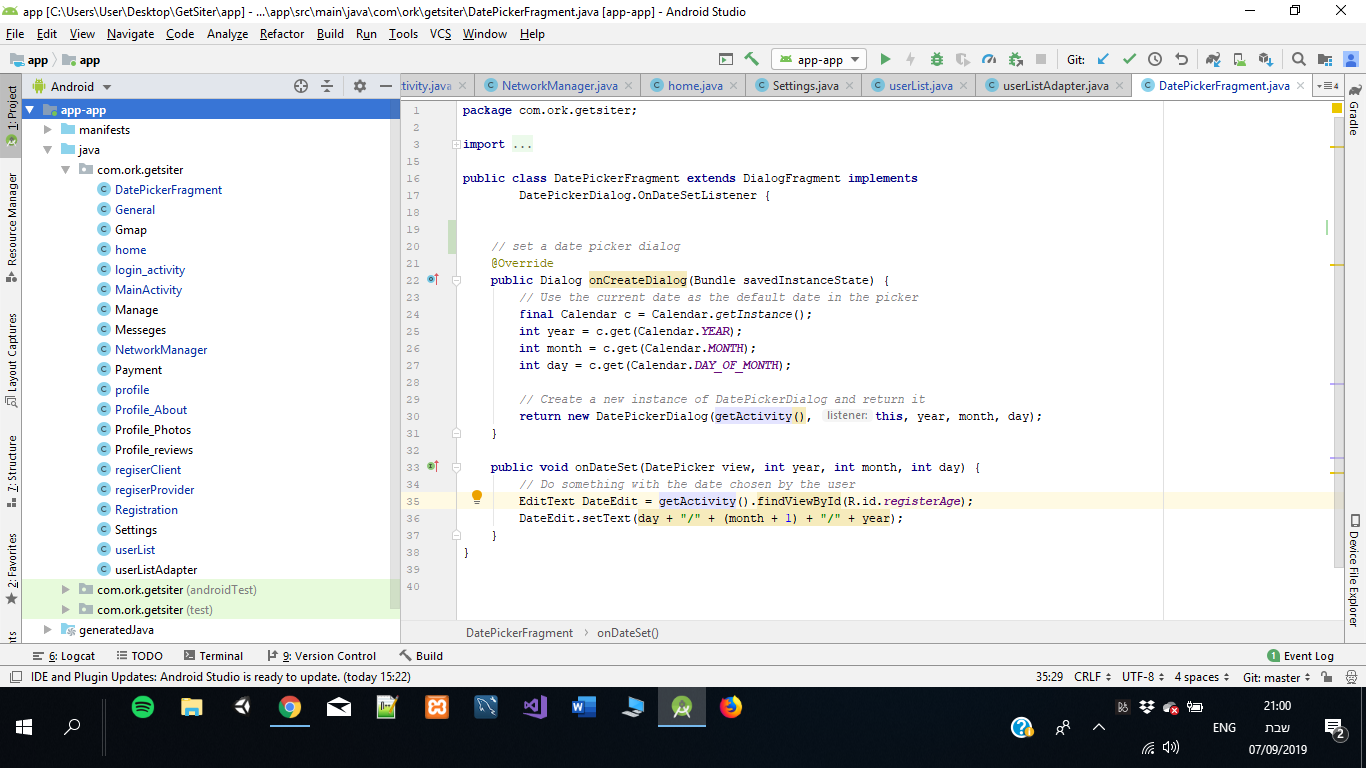


****

**Layouts :**

****

**Class list :**

****

**General.java -**

**general functions to be used all over the project.**

**package** com.ork.getsiter;

**import** android.content.Intent;

**import** android.database.Cursor;

**import** android.graphics.Bitmap;

**import** android.graphics.BitmapFactory;

**import** android.net.Uri;

**import** android.os.Build;

**import** android.provider.MediaStore;

**import** android.support.annotation.RequiresApi;

**import** android.text.TextUtils;

**import** android.util.Base64;

**import** android.util.Log;

**import** android.util.Patterns;

**import** java.io.ByteArrayOutputStream;

**import** java.io.File;

**import** java.io.FileInputStream;

**import** java.security.MessageDigest;

**import** java.security.NoSuchAlgorithmException;

**import** java.text.ParseException;

**import** java.text.SimpleDateFormat;

**import** java.time.LocalDate;

**import** java.time.Period;

**import** java.util.Calendar;

**import** java.util.Date;

**import** java.util.regex.Matcher;

**import** java.util.regex.Pattern;

**public class** General {

*// general functions to use all over the project*

*// get user age*

**public** String getAge(String date) **throws** ParseException {

*//using Calendar Object*

SimpleDateFormat sdf = **new** SimpleDateFormat(**"dd/MM/YYYY"**);

Date d = sdf.parse(date);

Calendar c = Calendar.*getInstance*();

c.setTime(d);

**int** year = c.get(Calendar.***YEAR***);

**int** month = c.get(Calendar.***MONTH***) + 1;

**int** day = c.get(Calendar.***DATE***);

LocalDate l1 = LocalDate.*of*(year, month, day);

LocalDate now1 = LocalDate.*now*();

Period diff1 = Period.*between*(l1, now1);

**return** Integer.*toString*( diff1.getYears());

}

*//decode user image from base64*

**public** Bitmap base64Decode(String img){

**byte**[] decodedString = Base64.*decode*(img, Base64.***DEFAULT***);

Bitmap decodedByte = BitmapFactory.*decodeByteArray*(decodedString, 0, decodedString.**length**);

**return** decodedByte;

}

*//encode user image to base 64*

**public** String base64Encode(Bitmap bitmap) {

ByteArrayOutputStream baos = **new** ByteArrayOutputStream();

bitmap.compress(Bitmap.CompressFormat.***JPEG***, 50, baos);

**byte**[] b = baos.toByteArray();

String temp = Base64.*encodeToString*(b, Base64.***DEFAULT***);

**return** temp;

}

*//password validation*

**public boolean** isValidPassword(**final** String password) {

Pattern pattern;

**final** String PASSWORD\_PATTERN = **"^(?=.\*[0-9])(?=.\*[A-Z])(?=.\*[@#$%^&+=!])(?=\\S+$).{4,}$"**;

pattern = Pattern.*compile*(PASSWORD\_PATTERN);

Matcher matcher = pattern.matcher(password);

**return** matcher.matches();

}

*//email validation*

**public boolean** isValidEmail(CharSequence target) {

**return** (!TextUtils.*isEmpty*(target) && Patterns.***EMAIL\_ADDRESS***.matcher(target).matches());

}

*//transfering string to sha512*

**public** String sha512( String value ) **throws** NoSuchAlgorithmException {

MessageDigest md = MessageDigest.*getInstance*(**"SHA-512"**);

**byte**[] digest = md.digest(value.getBytes());

StringBuilder sb = **new** StringBuilder();

**for** (**int** i = 0; i < digest.**length**; i++) {

sb.append(Integer.*toString*((digest[i] & 0xff) + 0x100, 16).substring(1));

}

**return** sb.toString();

}

*//connection with the server*

**public** String CallServer(String[] params) {

String aResultM = **""**;

**try**

{

NetworkManager task = **new** NetworkManager();

task.execute(params);

aResultM = task.get(); *//Add this*

}

**catch**(Exception ex)

{

ex.printStackTrace();

}

**return** aResultM;*//Need to get back the result*

}

}

**MainActivity.java**

package com.ork.getsiter;

import android.content.Intent;

import android.os.Bundle;

import android.support.design.widget.NavigationView;

import android.support.v4.app.Fragment;

import android.support.v4.app.FragmentManager;

import android.support.v4.view.GravityCompat;

import android.support.v4.widget.DrawerLayout;

import android.support.v7.app.ActionBarDrawerToggle;

import android.support.v7.app.AppCompatActivity;

import android.support.v7.widget.Toolbar;

import android.view.Menu;

import android.view.View;

import android.view.MenuItem;

import com.google.android.gms.maps.OnMapReadyCallback;

import com.google.android.gms.maps.SupportMapFragment;

public class MainActivity extends AppCompatActivity

implements NavigationView.OnNavigationItemSelectedListener {

public FragmentManager fragmentManager;

static int *cnt* = 0;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.*activity\_main*);

fragmentManager = getSupportFragmentManager();

Toolbar toolbar = (Toolbar) findViewById(R.id.*toolbar*);

setSupportActionBar(toolbar);

*//comiting come fragment*

getSupportFragmentManager().beginTransaction().replace(R.id.*cont*, new home()).commit();

DrawerLayout drawer = (DrawerLayout) findViewById(R.id.*drawer\_layout*);

ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(

this, drawer, toolbar, R.string.*navigation\_drawer\_open*, R.string.*navigation\_drawer\_close*);

drawer.addDrawerListener(toggle); *// add drawer listener*

toggle.syncState();

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

navigationView.setNavigationItemSelectedListener(this);

}

@Override

public void onBackPressed() {

DrawerLayout drawer = (DrawerLayout) findViewById(R.id.*drawer\_layout*);

if (drawer.isDrawerOpen(GravityCompat.*START*)) {

drawer.closeDrawer(GravityCompat.*START*);

} else {

super.onBackPressed();

}

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

*// Inflate the menu; this adds items to the action bar if it is present.*

getMenuInflater().inflate(R.menu.*main*, 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);

}

@SuppressWarnings("StatementWithEmptyBody")

@Override

public boolean onNavigationItemSelected(MenuItem item) {

*// Handle navigation view item clicks here.*

int id = item.getItemId();

switch (id){

case R.id.*nav\_profile*:

getSupportFragmentManager().beginTransaction().replace(R.id.*cont*, new profile()).commit();

break;

case R.id.*nav\_home*:

getSupportFragmentManager().beginTransaction().replace(R.id.*cont*, new home()).commit();

break;

case R.id.*nav\_msg*:

getSupportFragmentManager().beginTransaction().replace(R.id.*cont*, new Messeges()).commit();

break;

case R.id.*nav\_manage*:

getSupportFragmentManager().beginTransaction().replace(R.id.*cont*, new Manage()).commit();

break;

case R.id.*settings*:

getSupportFragmentManager().beginTransaction().replace(R.id.*cont*, new Settings()).commit();

break;

case R.id.*payment*:

getSupportFragmentManager().beginTransaction().replace(R.id.*cont*, new Payment()).commit();

break;

}

DrawerLayout drawer = (DrawerLayout) findViewById(R.id.*drawer\_layout*);

drawer.closeDrawer(GravityCompat.*START*);

return true;

}

}

**Login\_Activity.Java :**

**Login screen functionality:**

**package com.ork.getsiter;**

**import android.content.Context;**

**import android.content.Intent;**

**import android.content.SharedPreferences;**

**import android.support.v7.app.AppCompatActivity;**

**import android.os.Bundle;**

**import android.util.Log;**

**import android.view.View;**

**import android.widget.CheckBox;**

**import android.widget.TextView;**

**import android.widget.Toast;**

**import com.google.android.gms.maps.GoogleMap;**

**import org.json.JSONException;**

**import org.json.JSONObject;**

**import java.security.MessageDigest;**

**import java.security.NoSuchAlgorithmException;**

**public class login\_activity extends AppCompatActivity {**

**static public String *userId*; *// save user id staticly for easy use later on***

**static public Boolean *isBabysiter*;*//checking user type***

**public General general = new General();**

**SharedPreferences sharedPreferences; *//local storage variable***

***//sign in function***

**public void signIn(View view) throws JSONException, NoSuchAlgorithmException {**

**TextView userName = findViewById(R.id.*loginUserName*);**

**TextView password = findViewById(R.id.*loginPass*);**

***// username length validation***

**if(userName.getText().length() == 0){**

**Toast.*makeText*(this, "Enter user name", Toast.*LENGTH\_LONG*).show();**

**return;**

**}**

***// password length validation***

**if(password.getText().length() == 0){**

**Toast.*makeText*(this, "Enter Password", Toast.*LENGTH\_LONG*).show();**

**return;**

**}**

**String userNameTxt = userName.getText().toString(); *//get the username***

**String passwordTxt = general.sha512(password.getText().toString());*//get password***

**CheckBox checkBox = findViewById(R.id.*isBabysiter*);**

**String url;**

***isBabysiter* = checkBox.isChecked();**

**if( *isBabysiter* ){ *// checking type of user***

**url = "provider/login";**

**}else{**

**url = "user/login";**

**}**

**String[] data = {url, "username="+userNameTxt+"&password="+ passwordTxt };*//data for the request***

**Log.*i*("data",data.toString());**

**Log.*i*("url",url);**

**JSONObject res = new JSONObject (general.CallServer(data)); *// server request with the data***

**if(res.has("err")){ *//check for errors***

**Toast.*makeText*(this, res.get("err").toString(), Toast.*LENGTH\_LONG*).show();**

**return;**

**}else{ *// if no errors saving needed information into the local storage for quick connections later on.***

**sharedPreferences.edit().putString("username",userNameTxt).apply();**

**sharedPreferences.edit().putString("isBabysiter",*isBabysiter*.toString()).apply();**

**sharedPreferences.edit().putString("token", (String) res.getJSONObject("data").get("token")).apply();**

**sharedPreferences.edit().putString("userid", (String) res.getJSONObject("data").get("id")).apply();**

**Intent intent = new Intent(getApplicationContext(),MainActivity. class);**

**startActivity(intent);**

**Log.*i*("username",sharedPreferences.getString("username",""));**

**}**

**}**

***//clicking the btn will change the activity***

**public void signUpBtnFunc(View view) {**

**Intent intent = new Intent(this, Registration.class);**

**startActivity(intent);**

**}**

**@Override**

**protected void onCreate(Bundle savedInstanceState) {**

**super.onCreate(savedInstanceState);**

***//getting information from the local storage***

**sharedPreferences = this.getSharedPreferences("com.getsiter.SharedPreferences", Context.*MODE\_PRIVATE*);**

***// sharedPreferences.edit().clear().commit();***

**Log.*i*("username",sharedPreferences.getString("username",""));**

**Log.*i*("token",sharedPreferences.getString("token",""));**

**Log.*i*("userid",sharedPreferences.getString("userid",""));**

***isBabysiter* = Boolean.*parseBoolean*( sharedPreferences.getString("isBabysiter",""));**

***//sharedPreferences.edit().clear().commit();***

**if( sharedPreferences.getString("username","").length() > 0){**

**if( sharedPreferences.getString("token","").length() > 0){**

**if( sharedPreferences.getString("userid","").length() > 0) {**

**Log.*i*("haskey", "key");**

***// if everything is valid, changing to mainActivity***

**Intent intent = new Intent(getApplicationContext(), MainActivity.class);**

**startActivity(intent);**

**}**

**}**

**}**

**setContentView(R.layout.*activity\_login\_activity*);**

**}**

**}**

**NetworkManager.Java**

**package com.ork.getsiter;**

**import android.app.ProgressDialog;**

**import android.os.AsyncTask;**

**import android.util.Log;**

**import android.widget.TextView;**

**import org.json.JSONObject;**

**import java.io.BufferedReader;**

**import java.io.IOException;**

**import java.io.InputStream;**

**import java.io.InputStreamReader;**

**import java.net.HttpURLConnection;**

**import java.net.MalformedURLException;**

**import java.net.URL;**

**public class NetworkManager extends AsyncTask<String, Void, String>{**

***//private String url = "http://10.0.2.2:4000/";***

**private String url = "http://ec2-18-218-31-200.us-east-2.compute.amazonaws.com:3000/";**

**@Override**

**protected String doInBackground(String... params) {**

**HttpURLConnection connection = null;**

**BufferedReader reader = null;**

**try { *// try to open connection***

**URL url = new URL(this.url+params[0]+"?"+params[1]);**

**Log.*i*("url", url.toString());**

**connection = (HttpURLConnection) url.openConnection();**

**connection.connect();**

**InputStream stream = connection.getInputStream();**

**reader = new BufferedReader(new InputStreamReader(stream));**

**StringBuffer buffer = new StringBuffer();**

**String line = "";**

**while ((line = reader.readLine()) != null) {**

**buffer.append(line+"\n");**

**Log.*d*("Response: ", "> " + line); *//here u ll get whole response...... :-)***

**}**

**return buffer.toString();**

**} catch (MalformedURLException e) {**

**e.printStackTrace();**

**} catch (IOException e) {**

**e.printStackTrace();**

**} finally {**

**if (connection != null) {**

**connection.disconnect();**

**}**

**try {**

**if (reader != null) {**

**reader.close();**

**}**

**} catch (IOException e) {**

**e.printStackTrace();**

**}**

**}**

**return null;**

**}**

**@Override**

**protected void onPostExecute(String result){**

***//how i will pass this result where i called this task?***

**}**

**}**

**Registration.java -**

**package** com.ork.getsiter;

**import** android.content.Context;

**import** android.content.Intent;

**import** android.content.SharedPreferences;

**import** android.os.Bundle;

**import** android.support.design.widget.TabLayout;

**import** android.support.v4.app.FragmentActivity;

**import** android.util.Log;

**import** android.view.View;

**import** android.widget.Checkable;

**import** android.widget.EditText;

**import** android.widget.Toast;

**import** org.json.JSONException;

**import** org.json.JSONObject;

**import** java.security.NoSuchAlgorithmException;

**import static** com.ork.getsiter.login\_activity.*userId*;

**public class** Registration **extends** FragmentActivity {

**static** EditText *DateEdit*;

login\_activity **login**;

TabLayout **tabs**;

**boolean isBybsiter** = **false**;

Checkable **terms**; *//terms of service field*

SharedPreferences **sharedPreferences**;

**public** General **general** = **new** General(); *// to use general's class functions*

regiserClient **client**; *// userType*

regiserProvider **provider**;*// userType*

**static public** String *username* = **""**;

*//sign up function*

**public void** signUp(View view) **throws** JSONException, NoSuchAlgorithmException {

String res;

String[] data = **new** String[2];

**if**(**terms**.isChecked()) { *// if terms of service is checked*

**if** (!**isBybsiter**) { *// checking user type*

res = **client**.getClientData(getApplicationContext());

data[0] = **"user/register"**;

data[1] = res;

}

**else**{

res = **provider**.getProviderData(getApplicationContext());

data[0] = **"provider/register"**;

data[1] = res;

}

Log.*i*(**"dataRegister"**, data[0] + data[1] + res);

JSONObject results = **new** JSONObject(**general**.CallServer(data)); *// calling the server with the right info inside data variable*

**if** (results.has(**"err"**)) { *//looking for errors*

JSONObject err = (JSONObject) results.get(**"err"**);

Toast.*makeText*(**this**, err.get(**"errmsg"**).toString(), Toast.***LENGTH\_LONG***).show();

**return**;

}

*// if there are no errors found take the information from data variable.*

Toast.*makeText*(**this**, **"Register success"**, Toast.***LENGTH\_LONG***).show();

*userId* = (String) results.get(**"data"**);

**sharedPreferences**.edit().putString(**"username"**, *username*).apply();

**sharedPreferences**.edit().putString(**"isBabysiter"**, Boolean.*toString*(**isBybsiter**)).apply();

**sharedPreferences**.edit().putString(**"userid"**, *userId*).apply();

} **else** {

Toast.*makeText*(**this**, **"Please read Terms of service and check the box"**, Toast.***LENGTH\_LONG***).show();

**return**;

}

Log.*i*(**"res"**,res);

Intent intent = **new** Intent(getApplicationContext(),login\_activity.**class**);

startActivity(intent);

}

**protected void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

*// set up the registration layout*

setContentView(R.layout.***registration***);

*//set up the registration wrapper fragment.*

getSupportFragmentManager().beginTransaction().replace(R.id.***registerWrap***, **new** regiserClient()).commit();

**tabs** = findViewById(R.id.***registerTabs***);

**client** = **new** regiserClient();

**provider** = **new** regiserProvider();

*//waiting for events on the tab panel*

**tabs**.addOnTabSelectedListener(**new** TabLayout.BaseOnTabSelectedListener() {

@Override

**public void** onTabSelected(TabLayout.Tab tab) {

**if**( tab.getPosition() == 0){

**isBybsiter** = **false**;

getSupportFragmentManager().beginTransaction().replace(R.id.***registerWrap***, **client**).commit(); *// client fragment*

}**else**{

**isBybsiter** = **true**;

getSupportFragmentManager().beginTransaction().replace(R.id.***registerWrap***, **provider** ).commit(); *//provider fragment*

}

}

@Override

**public void** onTabUnselected(TabLayout.Tab tab) {

}

@Override

**public void** onTabReselected(TabLayout.Tab tab) {

}

});

**sharedPreferences** = **this**.getSharedPreferences(**"com.getsiter.SharedPreferences"**, Context.***MODE\_PRIVATE***);

**login** = **new** login\_activity();

**terms** = findViewById(R.id.***termsOfServiceCheckBox***);

}

}

**Register provider.Java**

**package** com.ork.getsiter;

**import** android.content.Context;

**import** android.content.Intent;

**import** android.database.Cursor;

**import** android.graphics.Bitmap;

**import** android.graphics.BitmapFactory;

**import** android.graphics.drawable.BitmapDrawable;

**import** android.net.Uri;

**import** android.os.Bundle;

**import** android.provider.MediaStore;

**import** android.support.annotation.NonNull;

**import** android.support.annotation.Nullable;

**import** android.support.v4.app.DialogFragment;

**import** android.support.v4.app.Fragment;

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

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

**import** android.util.Log;

**import** android.view.LayoutInflater;

**import** android.view.View;

**import** android.view.ViewGroup;

**import** android.widget.EditText;

**import** android.widget.ImageButton;

**import** android.widget.LinearLayout;

**import** android.widget.Toast;

**import** org.json.JSONArray;

**import** org.json.JSONException;

**import** java.io.FileNotFoundException;

**import** java.io.InputStream;

**import** java.security.NoSuchAlgorithmException;

**import static** com.ork.getsiter.Registration.*DateEdit*;

**import static** com.ork.getsiter.Registration.*username*;

**import static** com.ork.getsiter.home.*usersData*;

*// providers registration*

**public class** regiserProvider **extends** Fragment {

View **curView**;

ImageButton **imgBtn**;

General **general** = **new** General();

String **imageBase64**;

**static** LinearLayout *layout*;

**public static final int *PICK\_IMAGE*** = 1;

*//add image to your user*

**public void** addImage(){

Intent intent = **new** Intent();

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

intent.setAction(Intent.***ACTION\_GET\_CONTENT***);

startActivityForResult(Intent.*createChooser*(intent, **"Select Picture"**), ***PICK\_IMAGE***);

}

@Override

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

{

**if** (requestCode == ***PICK\_IMAGE***) {

**try** {

**final** Uri imageUri = data.getData();

String[] filePathColumn = {MediaStore.Images.Media.***DATA***};

Cursor cursor = **curView**.getContext().getContentResolver().query(imageUri,

filePathColumn, **null**, **null**, **null**);

cursor.moveToFirst();

**int** columnIndex = cursor.getColumnIndex(filePathColumn[0]);

cursor.getString(columnIndex);

cursor.close();

*//getting the image stream*

**final** InputStream imageStream = **curView**.getContext().getContentResolver().openInputStream(imageUri);

Bitmap selectedImage = BitmapFactory.*decodeStream*(imageStream);

*//transfering the image to base64*

**imageBase64** = **general**.base64Encode(Bitmap.*createScaledBitmap*(selectedImage, 250, 250, **false**) );

**imgBtn**.setBackground(**new** BitmapDrawable(getResources(), **general**.base64Decode(**imageBase64**)));

} **catch** (FileNotFoundException e) {

e.printStackTrace();

}

}

}

*// getting the data of the provider*

**public** String getProviderData(Context context) **throws** NoSuchAlgorithmException{

String res = **""**;

**for** (**int** i = 0; i < *layout*.getChildCount(); i++) { *//running on each field and using the info as needed.*

View v = *layout*.getChildAt(i);

**if** (v **instanceof** EditText) { *// if its a text box take wanted action by the user*

String name = context.getResources().getResourceEntryName(v.getId()).replace(**"register"**,**""**).toLowerCase();

String value = ((EditText) v).getText().toString();

**switch** (name){

**case "email"**:

**if**(**general**.isValidEmail(value)){ *//email validation*

*username* = value;

res += name+**"="**+value+**"&"**;

}

*// else{*

*// Toast.makeText(this, "Email is not valid", Toast.LENGTH\_LONG).show();*

*// return "err";*

*// }*

**break**;

**case "password"**:

EditText pass2 = *layout*.findViewById(R.id.***registerPass2***);

*//if(general.isValidPassword( value ) ){*

*//if(value.equals(pass2.getText().toString())){*

res += name+**"="**+**general**.sha512( value )+**"&"**;

*// }else{*

*// Toast.makeText(this, "Password not match", Toast.LENGTH\_LONG).show();*

*// return "err";*

*// }*

*// }*

*// else{*

*// Toast.makeText(this, "Password is not valid", Toast.LENGTH\_LONG).show();*

*// return "err";*

*// }*

**break**;

**default**:

**if**(name.toLowerCase().equals(**"pass2"**)){

**break**;

}

**if**(value.length() > 0){

res += name+**"="**+value+**"&"**;

}

**else** {

Toast.*makeText*(context, **"Please fill all the fields"**, Toast.***LENGTH\_LONG***).show();

}

**break**;

}

}

}

*// if(imageBase64.length() > 0){*

res += **"image="**+**imageBase64**;

*// }*

*// else{*

*// Toast.makeText(context, "Please upload your image", Toast.LENGTH\_LONG).show();*

*// return "err";*

*// }*

**return** res;

}

@Override

**public** View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {

**curView**= inflater.inflate(R.layout.***register\_provider***,container,**false**);

MainActivity mainActivity = **new** MainActivity();

*layout* = **curView**.findViewById(R.id.***registerLayout***);

*DateEdit* = **curView**.findViewById(R.id.***registerAge***);

**imgBtn** = **curView**.findViewById(R.id.***imageButton***);

**imgBtn**.setOnClickListener(**new** View.OnClickListener() {

@Override

**public void** onClick(View v) {

addImage();

}

});

*DateEdit*.setOnClickListener(**new** View.OnClickListener() {

@Override

**public void** onClick(View v) {

showTruitonDatePickerDialog(v);

}

});

**return curView**;

}

**public void** showTruitonDatePickerDialog(View v) {

DialogFragment newFragment = **new** DatePickerFragment();

newFragment.show(getChildFragmentManager(), **"datePicker"**);

}

}

**registerClient.java**

**package** com.ork.getsiter;

**import** android.content.Context;

**import** android.os.Bundle;

**import** android.support.annotation.NonNull;

**import** android.support.annotation.Nullable;

**import** android.support.v4.app.DialogFragment;

**import** android.support.v4.app.Fragment;

**import** android.util.Log;

**import** android.view.LayoutInflater;

**import** android.view.View;

**import** android.view.ViewGroup;

**import** android.widget.DatePicker;

**import** android.widget.EditText;

**import** android.widget.LinearLayout;

**import** android.widget.Toast;

**import** java.security.NoSuchAlgorithmException;

**import static** com.ork.getsiter.Registration.*username*;

**public class** regiserClient **extends** Fragment {

View **curView**;

**static** EditText *DateEdit*;

**static** LinearLayout *layout*;

**public** General **general** = **new** General();

*// get client data*

**public** String getClientData (Context context) **throws** NoSuchAlgorithmException {

String res = **""**;

**for** (**int** i = 0; i < *layout*.getChildCount(); i++) {

View v = *layout*.getChildAt(i);

**if** (v **instanceof** EditText) {

String name = context.getResources().getResourceEntryName(v.getId()).replace(**"register"**,**""**).toLowerCase();

String value = ((EditText) v).getText().toString();

**switch** (name){

**case "email"**:

*//if(general.isValidEmail(value)){*

*username* = value;

res += name+**"="**+value+**"&"**; *// set the email and username in the string*

*// }*

*// else{*

*// Toast.makeText(this, "Email is not valid", Toast.LENGTH\_LONG).show();*

*// return;*

*// }*

**break**;

**case "password"**:

EditText pass2 = *layout*.findViewById(R.id.***registerPass2***);

*//if(general.isValidPassword( value ) ){*

*//if(value.equals(pass2.getText().toString())){*

res += name+**"="**+**general**.sha512( value )+**"&"**; *// set the pass to the string after sha512*

*// }else{*

*// Toast.makeText(this, "Password not match", Toast.LENGTH\_LONG).show();*

*// return "err";*

*// }*

*// }*

*// else{*

*// Toast.makeText(this, "Password is not valid", Toast.LENGTH\_LONG).show();*

*// return "err";*

*// }*

**break**;

**default**:

**if**(name.toLowerCase().equals(**"pass2"**)){

**break**;

}

**if**(value.length() > 0){

res += name+**"="**+value+**"&"**;

}

**else** {

Toast.*makeText*(context, **"Please fill all the fields"**, Toast.***LENGTH\_LONG***).show();

}

**break**;

}

}

}

**return** res;

}

@Override

**public** View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {

**curView**= inflater.inflate(R.layout.***register\_client***,container,**false**);

MainActivity mainActivity = **new** MainActivity();

*DateEdit* = **curView**.findViewById(R.id.***registerAge***);

*layout* = **curView**.findViewById(R.id.***registerClientLayout***);

*DateEdit*.setOnClickListener(**new** View.OnClickListener() {

@Override

**public void** onClick(View v) {

showTruitonDatePickerDialog(v);

}

});

**return curView**;

}

**public void** showTruitonDatePickerDialog(View v) {

DialogFragment newFragment = **new** DatePickerFragment();

newFragment.show(getChildFragmentManager(), **"datePicker"**);

}

}

**Home.java**

**package** com.ork.getsiter;

**import** android.content.Context;

**import** android.content.SharedPreferences;

**import** android.os.Bundle;

**import** android.support.annotation.NonNull;

**import** android.support.annotation.Nullable;

**import** android.support.design.widget.TabLayout;

**import** android.support.v4.app.Fragment;

**import** android.util.Log;

**import** android.view.LayoutInflater;

**import** android.view.View;

**import** android.view.ViewGroup;

**import** android.widget.TableLayout;

**import** com.google.android.gms.maps.GoogleMap;

**import** com.google.gson.JsonObject;

**import** org.json.JSONException;

**import** org.json.JSONObject;

**public class** home **extends** Fragment {

View **curView**;

TabLayout **tabs**;

SharedPreferences **sharedPreferences**;

**public** General **general**;

**static public** JSONObject *usersData*;

@Nullable

@Override

**public** View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {

**curView**= inflater.inflate(R.layout.***home***,container,**false**);

MainActivity mainActivity = **new** MainActivity();

**general** = **new** General(); *// get general class for local usage*

*//get local storage information*

**sharedPreferences** = getActivity().getSharedPreferences(**"com.getsiter.SharedPreferences"**, Context.***MODE\_PRIVATE***);

String[] data = {**"user/getAll"**,**"token="**+**sharedPreferences**.getString(**"token"**, **""**)};

**try** { *// try to make a server request*

*usersData* = **new** JSONObject( **general**.CallServer(data) );

Log.*i*(**"user data"**,*usersData*.toString() );

} **catch** (JSONException e) {

e.printStackTrace();

}

**tabs** = **curView**.findViewById(R.id.***homeTabs***);

*// event lsitenet for tab panel*

**tabs**.addOnTabSelectedListener(**new** TabLayout.BaseOnTabSelectedListener() {

@Override

**public void** onTabSelected(TabLayout.Tab tab) {

**if**(tab.getPosition() == 0){

*// set on home fragment googleMap view*

getChildFragmentManager().beginTransaction().replace(R.id.***homeFragment***, **new** Gmap()).commit(); *// set home fragment*

}**else**{

*// set on home fragment the user List*

getChildFragmentManager().beginTransaction().replace(R.id.***homeFragment***, **new** userList()).commit();

}

}

@Override

**public void** onTabUnselected(TabLayout.Tab tab) {

}

@Override

**public void** onTabReselected(TabLayout.Tab tab) {

}

});

**return curView**;

}

}

**Gmap.java**

**Google map view**

**package** com.ork.getsiter;

**import** android.os.Bundle;

**import** android.support.annotation.NonNull;

**import** android.support.annotation.Nullable;

**import** android.support.design.widget.TabLayout;

**import** android.support.v4.app.Fragment;

**import** android.view.LayoutInflater;

**import** android.view.View;

**import** android.view.ViewGroup;

**import** com.google.android.gms.maps.GoogleMap;

**import** com.google.android.gms.maps.OnMapReadyCallback;

**public class** Gmap **extends** Fragment **implements** OnMapReadyCallback {

**public** GoogleMap **gmap**;

View **curView**;

@Override

**public void** onMapReady(GoogleMap googleMap) {

**gmap** = googleMap;

}

@Override

**public** View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {

**curView**= inflater.inflate(R.layout.***map\_view***,container,**false**);

MainActivity mainActivity = **new** MainActivity();

**return curView**;

}

}

**Profile.java**

**package** com.ork.getsiter;

**import** android.content.Context;

**import** android.content.SharedPreferences;

**import** android.graphics.Bitmap;

**import** android.graphics.drawable.BitmapDrawable;

**import** android.os.Bundle;

**import** android.support.annotation.NonNull;

**import** android.support.annotation.Nullable;

**import** android.support.design.widget.TabLayout;

**import** android.support.v4.app.Fragment;

**import** android.util.Log;

**import** android.view.LayoutInflater;

**import** android.view.View;

**import** android.view.ViewGroup;

**import** android.widget.ImageView;

**import** android.widget.RatingBar;

**import** android.widget.TextView;

**import** android.widget.Toast;

**import** org.json.JSONException;

**import** org.json.JSONObject;

**import static** com.ork.getsiter.login\_activity.*isBabysiter*;

**public class** profile **extends** Fragment {

View **curView**;

General **general** = **new** General();

**static** JSONObject *profileData*; *// saving staticly profileData for easy usage*

**void** initTabs(){

TabLayout tabLayout = **curView**.findViewById(R.id.***profileTabs***);

getFragmentManager().beginTransaction().replace(R.id.***tabs\_cont***, **new** Profile\_About()).commit();

tabLayout.addOnTabSelectedListener(**new** TabLayout.OnTabSelectedListener() { *//event click listener for tabpanel*

@Override

**public void** onTabSelected(TabLayout.Tab tab) {

**switch**(tab.getPosition()){

**case** 0:

*// commiting about fragment*

getFragmentManager().beginTransaction().replace(R.id.***tabs\_cont***, **new** Profile\_About()).commit();

**break**;

**case** 1:

*// commiting photos fragment*

getFragmentManager().beginTransaction().replace(R.id.***tabs\_cont***, **new** Profile\_Photos()).commit();

**break**;

**case** 2:

*//commiting reviews fragment*

getFragmentManager().beginTransaction().replace(R.id.***tabs\_cont***, **new** Profile\_reviews()).commit();

**break**;

**case** 3:

**break**;

}

}

@Override

**public void** onTabUnselected(TabLayout.Tab tab) {

}

@Override

**public void** onTabReselected(TabLayout.Tab tab) {

}

});

}

@Nullable

@Override

**public** View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {

**curView** = inflater.inflate(R.layout.***profile***,container,**false**);

TextView userName = **curView**.findViewById(R.id.***userName***);

TextView userTitle = **curView**.findViewById(R.id.***userTitle***);

ImageView userImage = **curView**.findViewById(R.id.***profileImage***);

RatingBar stars = **curView**.findViewById(R.id.***userRating***);

initTabs();

*//get info from local storage*

SharedPreferences sharedPreferences = getActivity().getSharedPreferences(**"com.getsiter.SharedPreferences"**, Context.***MODE\_PRIVATE***);

String path = *isBabysiter*? **"provider"** : **"user"**; *//setting user type*

String[] params = {path+**"/getbyid"**,**"token="**+sharedPreferences.getString(**"token"**,**""**)+**"&id="**+sharedPreferences.getString(**"userid"**,**""**)+ **"&username="** +sharedPreferences.getString(**"username"**,**""**) };

**try** {

*//trying to call the server and transfer the data*

*profileData* = **new** JSONObject( **general**.CallServer(params));

**if**(*profileData*.has(**"err"**)) {

*//looking for errors*

String err = (String) *profileData*.get(**"err"**);

Toast.*makeText*(**curView**.getContext(),err, Toast.***LENGTH\_LONG***).show();

}**else**{

*//if no errors found seting the info for the user*

*profileData* = *profileData*.getJSONObject(**"data"**);

Log.*i*(**"profileData"**, *profileData*.toString());

userName.setText(*profileData*.getString(**"firstname"**) + **" "** +*profileData*.getString(**"lastname"**));

userTitle.setText(*profileData*.getString(**"role"**));

Log.*i*(**"img"**, *profileData*.getString(**"img"**));

Bitmap bm = **general**.base64Decode(*profileData*.getString(**"img"**));

Log.*i*(**"bm"**, bm.toString());

userImage.setImageBitmap(bm );

}

} **catch** (Exception e) {

e.printStackTrace();

}

**return curView**;

}

}

**Profile\_About.Java**

**package** com.ork.getsiter;

**import** android.os.Bundle;

**import** android.support.annotation.NonNull;

**import** android.support.annotation.Nullable;

**import** android.support.v4.app.Fragment;

**import** android.view.LayoutInflater;

**import** android.view.View;

**import** android.view.ViewGroup;

**import** android.widget.TextView;

**import static** com.ork.getsiter.profile.*profileData*;

**public class** Profile\_About **extends** Fragment {

*//about info at user profile*

View **curView**;

General **general** = **new** General();

@Nullable

@Override

**public** View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {

**curView** = inflater.inflate(R.layout.***profile\_about***,container,**false**);

TextView genralInfo = **curView**.findViewById(R.id.***userDeteils***);

TextView about = **curView**.findViewById(R.id.***aboutUser***);

**try** { *// trying to set user info for about pagge*

genralInfo.setText(**"age: "** + **general**.getAge(*profileData*.getString(**"age"**) + **"\nCity: "** + *profileData*.getJSONObject(**"adress"**).getString(**"city"**)));

} **catch** (Exception e) {

e.printStackTrace();

}

**return curView**;

}

}

**Profile\_Photos.java**

**package** com.ork.getsiter;

**import** android.os.Bundle;

**import** android.support.annotation.NonNull;

**import** android.support.annotation.Nullable;

**import** android.support.v4.app.Fragment;

**import** android.view.LayoutInflater;

**import** android.view.View;

**import** android.view.ViewGroup;

**public class** Profile\_Photos **extends** Fragment {

@Nullable

@Override

**public** View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {

**return** inflater.inflate(R.layout.***profile\_photost***,container,**false**);

}

}

**Profile\_Reviews.java**

**package** com.ork.getsiter;

**import** android.os.Bundle;

**import** android.support.annotation.NonNull;

**import** android.support.annotation.Nullable;

**import** android.support.v4.app.Fragment;

**import** android.view.LayoutInflater;

**import** android.view.View;

**import** android.view.ViewGroup;

**public class** Profile\_reviews **extends** Fragment {

@Nullable

@Override

**public** View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {

**return** inflater.inflate(R.layout.***profile\_reviews***,container,**false**);

}

}

**UserList.java**

**package** com.ork.getsiter;

**import** android.os.Bundle;

**import** android.support.annotation.NonNull;

**import** android.support.annotation.Nullable;

**import** android.support.v4.app.Fragment;

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

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

**import** android.view.LayoutInflater;

**import** android.view.View;

**import** android.view.ViewGroup;

**import** org.json.JSONArray;

**import** org.json.JSONException;

**import static** com.ork.getsiter.home.*usersData*;

**public class** userList **extends** Fragment {

*//user list view for all of the service providers*

View **curView**;

@Override

**public** View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {

**curView**= inflater.inflate(R.layout.***list\_user\_view***,container,**false**);

MainActivity mainActivity = **new** MainActivity();

JSONArray listData;

userListAdapter adapter;

*// set up the RecyclerView*

**try** {

listData = *usersData*.getJSONArray(**"data"**); *//get json of user data*

RecyclerView recyclerView = **curView**.findViewById(R.id.***userList***);

recyclerView.setLayoutManager(**new** LinearLayoutManager(**curView**.getContext()));

adapter = **new** userListAdapter(**curView**.getContext(), listData); *//set an list adapter*

recyclerView.setAdapter(adapter);

} **catch** (JSONException e) {

e.printStackTrace();

}

**return curView**;

}

}

**UserAdapter.java**

**package** com.ork.getsiter;

**import** android.annotation.SuppressLint;

**import** android.content.Context;

**import** android.support.annotation.NonNull;

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

**import** android.view.LayoutInflater;

**import** android.view.View;

**import** android.view.ViewGroup;

**import** android.widget.ImageView;

**import** android.widget.RatingBar;

**import** android.widget.TextView;

**import** org.json.JSONArray;

**import** org.json.JSONException;

**import** org.json.JSONObject;

**import** java.util.List;

**public class** userListAdapter **extends** RecyclerView.Adapter<userListAdapter.ViewHolder> {

**private** JSONArray **mData**;

**private** LayoutInflater **mInflater**;

**private** ItemClickListener **mClickListener**;

*// data is passed into the constructor*

userListAdapter(Context context, JSONArray data) {

**this**.**mInflater** = LayoutInflater.*from*(context);

**this**.**mData** = data;

}

*// inflates the row layout from xml when needed*

@Override

**public** ViewHolder onCreateViewHolder(ViewGroup parent, **int** viewType) {

View view = **mInflater**.inflate(R.layout.***user\_list\_row***, parent, **false**);

**return new** ViewHolder(view);

}

*// binds the data to the TextView in each row*

@SuppressLint(**"SetTextI18n"**)

@Override

**public void** onBindViewHolder(ViewHolder holder, **int** position) {

JSONObject cur;

**try** {

cur = **mData**.getJSONObject(position);

holder.**username**.setText(cur.getString(**"firstname"**) + **" "** + cur.getString(**"lastname"**));

*//holder.userImage.setImageBitmap();*

*// holder.usreRate.setNumStars(cur.get("rating"));*

holder.**usreRate**.setNumStars(3);

holder.**userAge**.setText(cur.getString(**"age"**));

holder.**userLocation**.setText(**"tel aviv achi"**);

} **catch** (JSONException e) {

e.printStackTrace();

}

}

*// total number of rows*

@Override

**public int** getItemCount() {

**return mData**.length();

}

*// stores and recycles views as they are scrolled off screen*

**public class** ViewHolder **extends** RecyclerView.ViewHolder **implements** View.OnClickListener {

TextView **username**;

ImageView **userImage**;

RatingBar **usreRate**;

TextView **userAge**;

TextView **userLocation**;

ViewHolder(View itemView) {

**super**(itemView);

**username** = itemView.findViewById(R.id.***userName***);

**userImage** = itemView.findViewById(R.id.***userImage***);

**usreRate** = itemView.findViewById(R.id.***userRating***);

**userAge** = itemView.findViewById(R.id.***userAge***);

**userLocation** = itemView.findViewById(R.id.***userLocation***);

itemView.setOnClickListener(**this**);

}

@Override

**public void** onClick(View view) {

**if** (**mClickListener** != **null**) **mClickListener**.onItemClick(view, getAdapterPosition());

}

}

*// convenience method for getting data at click position*

JSONObject getItem(**int** id) **throws** JSONException {

**return mData**.getJSONObject(id);

}

*// allows clicks events to be caught*

**void** setClickListener(ItemClickListener itemClickListener) {

**this**.**mClickListener** = itemClickListener;

}

*// parent activity will implement this method to respond to click events*

**public interface** ItemClickListener {

**void** onItemClick(View view, **int** position);

}

}

**DatePickerFragment**

**package** com.ork.getsiter;

**import** android.app.DatePickerDialog;

**import** android.app.Dialog;

**import** android.os.Bundle;

**import** android.support.v4.app.DialogFragment;

**import** android.widget.DatePicker;

**import** android.widget.EditText;

**import** java.util.Calendar;

**import static** android.view.WindowManager.LayoutParams.***SOFT\_INPUT\_STATE\_ALWAYS\_HIDDEN***;

**import static** android.view.WindowManager.LayoutParams.***SOFT\_INPUT\_STATE\_HIDDEN***;

**import static** android.view.WindowManager.LayoutParams.***SOFT\_INPUT\_STATE\_VISIBLE***;

**public class** DatePickerFragment **extends** DialogFragment **implements**

DatePickerDialog.OnDateSetListener {

*// set a date picker dialog*

@Override

**public** Dialog onCreateDialog(Bundle savedInstanceState) {

*// Use the current date as the default date in the picker*

**final** Calendar c = Calendar.*getInstance*();

**int** year = c.get(Calendar.***YEAR***);

**int** month = c.get(Calendar.***MONTH***);

**int** day = c.get(Calendar.***DAY\_OF\_MONTH***);

*// Create a new instance of DatePickerDialog and return it*

**return new** DatePickerDialog(getActivity(), **this**, year, month, day);

}

**public void** onDateSet(DatePicker view, **int** year, **int** month, **int** day) {

*// Do something with the date chosen by the user*

EditText DateEdit = getActivity().findViewById(R.id.***registerAge***);

DateEdit.setText(day + **"/"** + (month + 1) + **"/"** + year);

}

}