

2019

פיקס Fix



עבודות
מזדמנות
על הדרך

ספר פרויקט –
אפליקציית FIX

מגיש: ישי בראון

תוכן

| | |
|---------|--|
| 3..... | רקע כללי |
| 3..... | מטרת הפרויקט: |
| 3..... | הצורך במערכת: |
| 3..... | תיאור קצר ונתונים טכניים |
| 3..... | בעיות אפשריות |
| 3..... | פתרונות אפשריים |
| 4..... | מבנה נתונים |
| 4..... | אמצעי אחסון |
| 5..... | תרשים זרימת מידע |
| 6..... | גבולות המערכת |
| 6..... | מספר שעות המוקדש לפרויקט: 500 שעות |
| 6..... | הידע המוקנה ע"י הפרויקט |
| 6..... | שלבים בפיתוח הפרויקט |
| 6..... | ספרות וביבליוגרפיה: |
| 7..... | עקרונות התכנון והביצוע |
| 7..... | הפעלת התוכנה |
| 8..... | ממשק אימות firebase |
| 14..... | ממשק Firebase Realtime Database |
| 14..... | ממשק נותן שירות: שגיאה! הסימניה אינה מוגדרת. |
| 16..... | המחלקות של הפרויקט |

תאריך: 3.9.19

לכבוד יחידת הפרויקטים מה"ט

הצעה לפרויקט גמר

א. פרטי הסטודנטים

| שם הסטודנט | ת.ז. 9. ספרות | כתובת | טלפון נייד | תאריך סיום הלימודים |
|------------|---------------|---------------|------------|---------------------|
| ישי בראון | 308396761 | קיבוץ סופה 45 | 0548029847 | 13/06/19 |

שם המכללה: אריאל

סמל המכללה: 72211

מסלול ההכשרה: **הנדסאים**

מגמת לימוד: תוכנה

מקום ביצוע הפרויקט: אריאל

ב. פרטי המנחה האישי

| שם המנחה * | כתובת | טלפון נייד | תואר | מקום עבודה/תפקיד |
|------------|-------|------------|------|------------------|
| אפי פרופוס | - | - | - | - |

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

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

רקע כללי

מטרת הפרויקט:

שוק לעבודות קטנות ומזדמנות. משתמשים יכולים לפרסם עבודות לא דחופות ובעלי מקצוע יכולים לראות איזה עבודות יש בסביבה.

המודל המוצע הוא אפליקציה סלולרית לאיתור שירותי נותן שירות בתוך האזור של המשתמש, כדי לסייע בייעול תהליך זה ומתן גישה מובנית לקביעת מיקום ואיכות השירות שיסופקו על ידי נותני שירות. האפליקציה פותחה במערכת הפעלה אנדרואיד בגלל הפופולריות שלה בקרב משתמשים ניידים רבים.

הצורך במערכת:

קיים צורך גובר בשירותי ה'שיפוצניק', המיוחס לכמה גורמים תורמים הכוללים, כמו: ביקוש מקומי, השפעה בשוק, בעלי בתים שניים, אחזקת נכסים מסחריים, וחייהם של אנשים הופכים להיות עמוסים יותר ויותר קדחתניים.

יותר אנשים מחפשים כעת עזרה בעבודות מזדמנות בביתם, כמו החלפת אביזרי תאורה או העמדת מדפים. הפתרונות הקיימים, המנסים לפתור בעיה זו, מבוטלים ומציעים אנשי קשר המפוזרים ברחבי האינטרנט ואין שום שיטה מובנית לקביעת מיקום ואיכות השירות הניתנים על ידי בעלי מלאכה.

תיאור קצר ונתונים טכניים

המערכת מורכבת מאפליקציה סלולרית למערכת אנדרואיד. בסיס הנתונים של המערכת ממומש ע"י Firebase. בסיס הנתונים משמש לאחסון נתונים הן מהיישום הנייד של הלקוח והן ממרכז השליטה באינטרנט. השרת משמש לעיבוד בקשות בין הלקוחות למסד הנתונים.

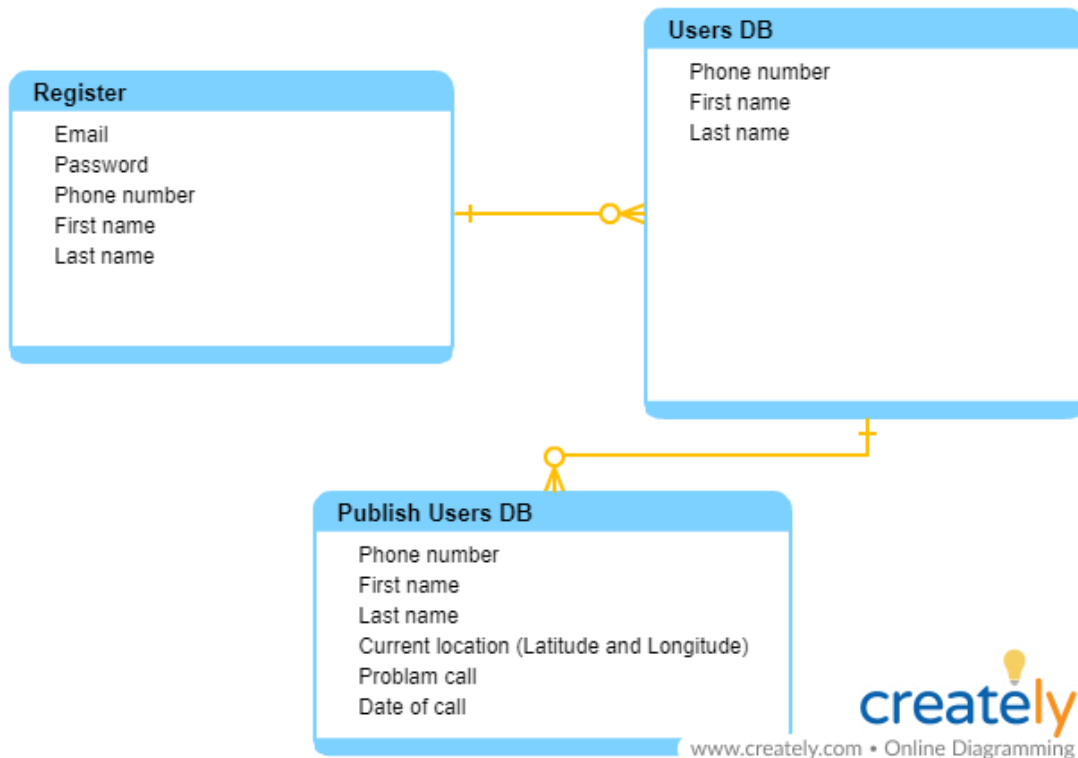
בעיות אפשריות

- בעיות בתקשורת נתונים ובחיבור לאינטרנט
- עדכון נתונים על מיקום בזמן אמת
- אימות משתמשים
- עומס נתונים על השרת
- באגים לדוגמה: תקלת אינטרנט, שירותי מיקום

פתרונות אפשריים

- גיבוי הנתונים שעל השרת בצורה מסונכרנת
- הגדרת בסיס נתונים ב-firebase
- מערכת אימות דרך המייל, באפליקציה
- בקרת עומס מובנת בשירות firebase ומענה בהתאם לבעיה.
- מענה בקוד באמצעות try catch

מבנה נתונים

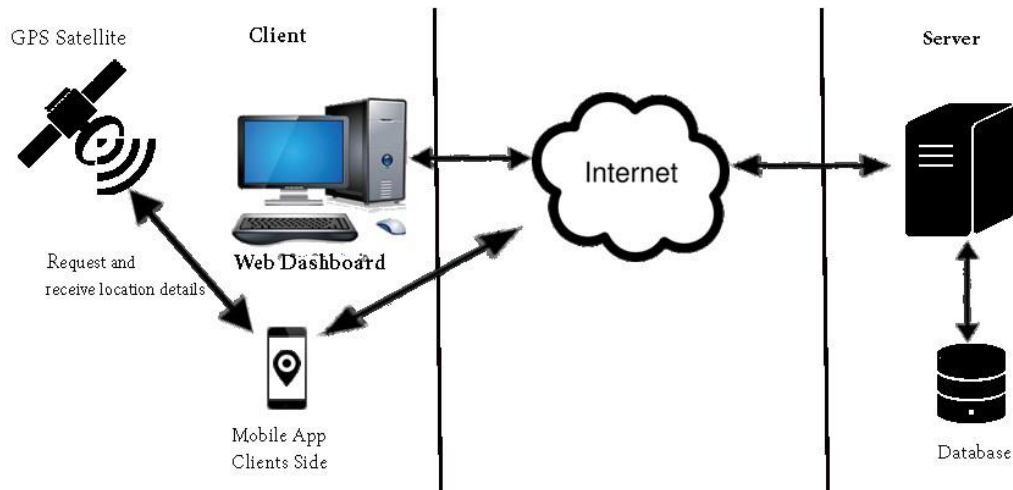


אמצעי אחסון

הפרויקט יוגש מותקן על פלאפון מסוג אנדרואיד. כמו כן יעלה מודל בהמשך לחנות Google Play. הקודים והמחלקות יוגשו בנפרד ע"ג מחשב והתקן חיצוני.

תרשים זרימת מידע

התרשים מציג את הארכיטקטורה של המערכת ואת האינטראקציות בין מרכיביה השונים. המודל המוצע משתמש ב-GPS בכדי לספק מידע מבוסס מיקום ללקוחות. המערכת המוצעת הכוללת חזית שהיא אפליקציית אנדרואיד, יישום שרת אחורי ומסד נתונים. הלקוח מבקש פרטי מיקום מ-GPS. לאחר מכן משתמשים בפרטי מיקום אלה כדי לקבוע את מיקום הלקוח כדי שיוכל לספק מידע על סמך מיקומו.



גבולות המערכת

המערכת אינה עוסקת בכל מה שנוגע בתשלום של הלקוח: לא מבצעת קליטת נתוני אשראי ושמירתם ולא מנהלת דו"חות כספיים, לא ממומש פרוטוקול מוגן להעברת נתונים אלה. המערכת אינה עוסקת בניהול נתוני המשתמשים, אלא בתיאום בין הלקוחות על בסיס המפה. האפליקציה לא תעבוד ללא קישור לרשת והרשאת גישה למיקום.

מספר שעות המוקדש לפרויקט: 500 שעות

הידע המוקנה ע"י הפרויקט

ידע בנושא כתיבה בשפות ג'אווה בשילוב עם ממשק אנדרואיד סטודיו
ידע בבניית מסד נתונים
ידע וניסיון בחקירה, בנייה וניתוח של המערכת

שלבים בפיתוח הפרויקט

שלב א' – לימוד החומר התיאורטי והרחבתו.
שלב ב' – חלוקת הפיתוח לשלבים אטומיים.
שלב ג' – תכנון של כל גורם עצמאי.
שלב ד' – ביצוע אינטגרציה בין שני חלקי המערכת.
שלב ה' – הרצת מערכת ותיקון תקלות.
שלב ו' – כתיבת ספר פרויקט.

ספרות וביבליוגרפיה:

www.stackoverflow.com

www.androidhive.info

www.androidcentral.com

www.developer.android.com

עקרונות התכנון והביצוע

הפעלת התוכנה

א. הרשמה

המשתמש נרשם ע"י כתובת מייל וסיסמא, ופרטים אישיים (טלפון, ושם מלא)

תהליך ביצוע קריאת שירות למשתמש

המשתמש נכנס למסך הראשי ובוחר באופציה - כניסה ללקוח.

האלגוריתם המשמש ליישום פרסום עבודה כל שהיא על ידי השגת תחילה של קואורדינטות המשתמשים, כלומר קו אורך ורוחב ממסד הנתונים ושליפת השירות הרצוי מבקשת החיפוש.

המשתמש לוחץ על כפתור "בצע קריאת שירות"

המשתמש מזין בחלון שקפץ את פירוט הבעיה

המשתמש מקבל הודעת Toast שבוצע בהצלחה, וחוזר למסך הראשי.

במסך הראשי יש מסוף שכותב אם יש קריאת שירות פעילה, תאריך ומיקום (ע"י מתודת Geocoder)

ע"י כניסה להגדרות, המשתמש יכול לשנות את פרטי המודעה וגם פרטים אישיים. מתעדכן אוטומטית גם במפה.

תהליך ביצוע חיפוש קריאות שירות לנותן שירות (בעל מקצוע)

המשתמש נכנס למסך הראשי ובוחר באופציה - כניסה לבעל מקצוע.

האפליקציה מזהה את המיקום של המשתמש.

איפה שהמשתמשים סימנו פרסום קריאת שירות, מסומן בדגלים ירוקים.

כל לחיצה על דגל ירוק נותן את הפרטים הבאים:

שם מלא

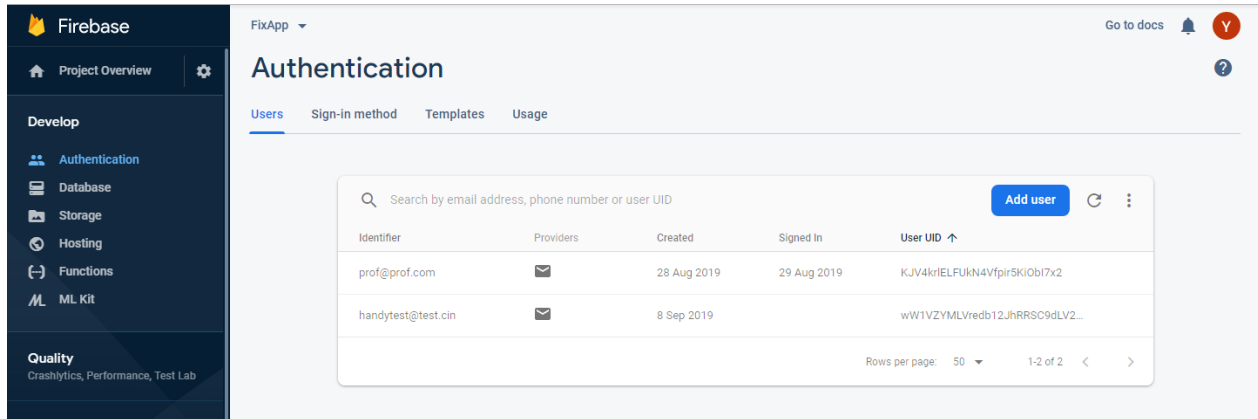
תאריך

תיאור הבעיה

ושלושה כפתורים- קישור להודעה מיידית באפליקציית Whatsup, ניווט בWaze, והתקשרות למשתמש.

ממשק אימות firebase

משמש למנהל המערכת לניהול הנתונים בענן. יש לו זכויות ניהול על המערכת ויכול לגשת לכל התכונות של המערכת. הוא יכול לערוך את פרטיהם של נותני השירות, להוסיף אותם למערכת ולהסיר אותם אם הם כבר לא נחוצים במערכת.



ממשק Firebase real time

המשתמשים עוברים תהליך של אימות ומקבלים ID מזהה משתמש. כאשר נכנסים לממשק לקוח משדרים firebase CustomerPublish נתוני מיקום אורך ורוחב ותאריך קריאת שירות.

במקרה של ביטול קריאת שירות, כל הפרטים מCustomerPublish נמחקים.



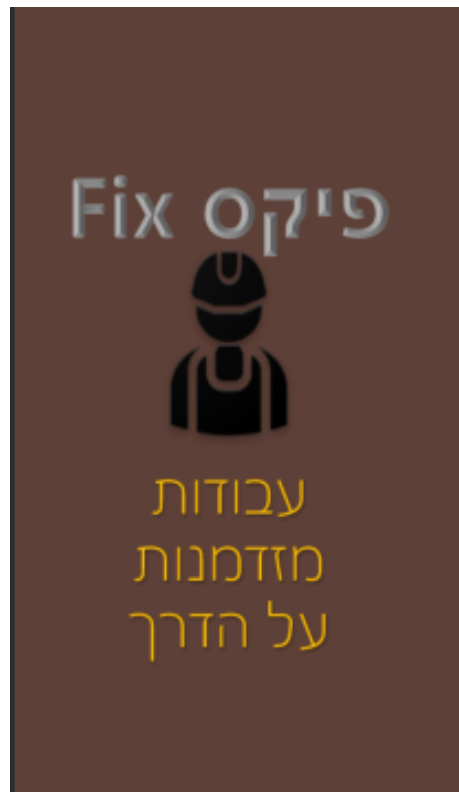
Rules:

"read": "auth != null"

"write": "auth != null"

רק משתמשים מאומתים יכולים לקרוא ולכתוב
מלבסיס הנתונים

כניסה לאפליקציה (splash):



מסך Login



מסך הרשמה



אימייל

סיסמה

שם פרטי

שם משפחה

מספר טלפון נייד

הירשם

שכחת סיסמה?

כבר רשום. תחבר אותי!!

Reset Password מסך



שכחת סיסמה?

אנו זקוקים לכתובת הדואר האלקטרוני הרשום שלך כדי לשלוח לך הוראות לאיפוס סיסמה.

אימייל

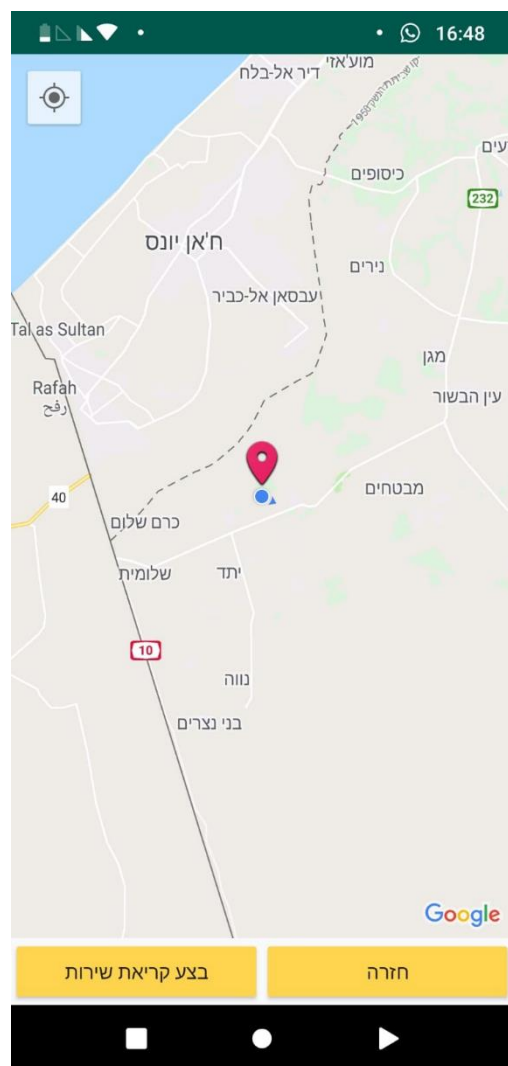
איפוס סיסמה

חזרה

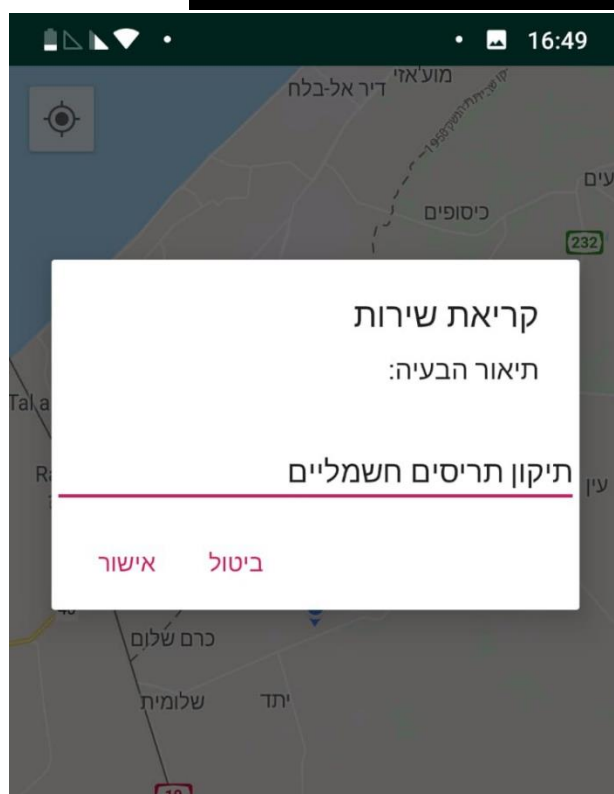
מסך ראשי



כניסה ללקוח



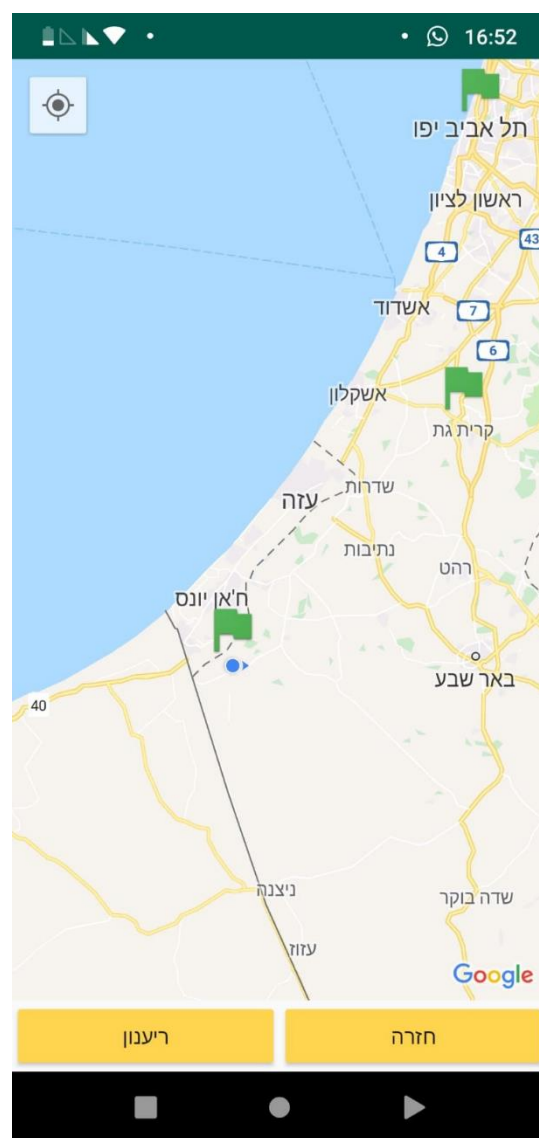
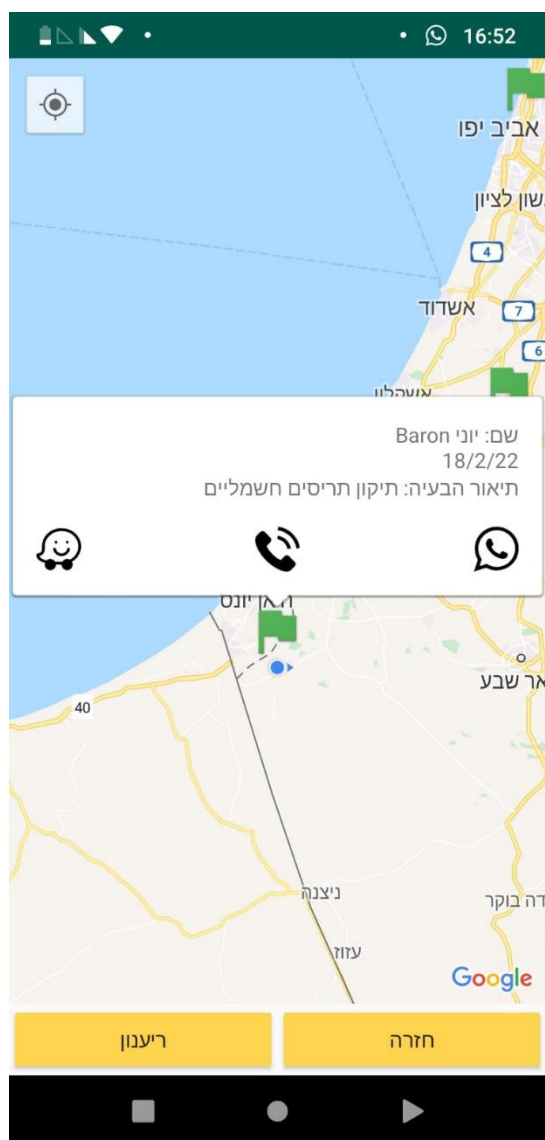
בצע קריאת שירות



אישור וחזרה לתפריט הראשי



כניסה לבעל מקצוע:



כניסה להגדרות:

במקרה של ביטול קריאת שירות.

שם:

ירין

משפחה:

זילברמן

טלפון נייד:

0548029847

קריאת שירות אחרונה:

תיקון תריסים חשמליים

עדכון

ביטול קריאת שירות

חזרה

שם:

ירין

משפחה:

זילברמן

טלפון נייד:

0548029847

עדכון

חזרה

המחלקות של הפרויקט

MainActivity

```
package com.example.fixxapp;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;

import android.Manifest;
import android.content.Intent;

import android.os.Bundle;

import com.google.firebase.auth.FirebaseAuth;

public class MainActivity extends AppCompatActivity {
    // private Button mDriver, mCustomer, mDemo, mGoogle,mFacebook,mlogin;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // setTheme(R.style.AppTheme);
        super.onCreate(savedInstanceState);
        try
        {
            this.getSupportActionBar().hide();
        }
        catch (NullPointerException e){}

        ActivityCompat.requestPermissions(this,new
String[]{Manifest.permission.ACCESS_FINE_LOCATION}, 1);
        ActivityCompat.requestPermissions(this,new
String[]{Manifest.permission.CALL_PHONE}, 1);

        FirebaseAuth mAuth = FirebaseAuth.getInstance();
        if (mAuth.getCurrentUser() != null) {
            Intent intent = new Intent(MainActivity.this,
ChooseScreenActivity.class);
            intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
            startActivity(intent);
        }
        else {
            Intent intent = new Intent(MainActivity.this,
LoginActivity.class);
            intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
            startActivity(intent);
        }

    }
}
```

LoginActivity

```
package com.example.fixxapp;
import android.content.Intent;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ProgressBar;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;

public class LoginActivity extends AppCompatActivity {

    private EditText mEmail, mPassword;
    private Button mLogin, btnReset, btnSignup;
    private ProgressBar progressBar;

    private FirebaseAuth mAuth;
    private FirebaseAuth.AuthStateListener firebaseAuthListener;
    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        try
        {
            this.getSupportActionBar().hide();
        }
        catch (NullPointerException e){}

        setContentView(R.layout.activity_login);
        progressBar = (ProgressBar) findViewById(R.id.progressBar);
        btnSignup = (Button) findViewById(R.id.btn_signup);

        mEmail = (EditText) findViewById(R.id.email);
        mPassword = (EditText) findViewById(R.id.password);

        mLogin = (Button) findViewById(R.id.btn_login);
        btnReset = (Button) findViewById(R.id.btn_reset_password);

        mAuth = FirebaseAuth.getInstance();

        firebaseAuthListener = new FirebaseAuth.AuthStateListener() {
            @Override
            public void onAuthStateChanged(@NonNull FirebaseAuth
firebaseAuth) {
                FirebaseUser user =
FirebaseAuth.getInstance().getCurrentUser();
                if(user!=null){
                    Intent intent = new Intent(LoginActivity.this,
ChooseScreenActivity.class);
                    startActivity(intent);
                    finish();
                    return;
                }
            }
        };
    }
};
```

```

        btnSignup.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new Intent(LoginActivity.this,
SignupActivity.class));
            }
        });

        btnReset.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new Intent(LoginActivity.this,
ResetPasswordActivity.class));
            }
        });

        mLogin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String email = mEmail.getText().toString();
                final String password = mPassword.getText().toString();

                if (TextUtils.isEmpty(email)) {
                    Toast.makeText(getApplicationContext(), "דוא'ל כתובת הזן",
                    Toast.LENGTH_SHORT).show();
                    return;
                }
                if
(!android.util.Patterns.EMAIL_ADDRESS.matcher(email).matches())
                {
                    Toast.makeText(getApplicationContext(), "תקנית דוא'ל כתובת הזן",
                    Toast.LENGTH_SHORT).show();
                    return;
                }

                if (TextUtils.isEmpty(password)) {
                    Toast.makeText(getApplicationContext(), "הסיסמה את הזן!",
                    Toast.LENGTH_SHORT).show();
                    return;
                }
                if (password.length() < 6) {

                    Toast.makeText(getApplicationContext(), "מדי קצרה הסיסמה,
מינימום הזן 6 תווים", Toast.LENGTH_SHORT).show();
                    return;
                }

                progressBar.setVisibility(View.VISIBLE);

                //authenticate user
                mAuth.signInWithEmailAndPassword(email, password)
                    .addOnCompleteListener(LoginActivity.this, new
                    OnCompleteListener<AuthResult>() {
                        @Override
                        public void onComplete(@NonNull Task<AuthResult> task) {
                            // If sign in fails, display a message to
                            the user. If sign in succeeds
                            // the auth state listener will be notified
                            and logic to handle the
                            // signed in user can be handled in the
                            listener.
                            progressBar.setVisibility(View.GONE);
                            if (!task.isSuccessful()) {
                                // there was an error
                                if (password.length() < 6) {
                                    mPassword.setError(getString(R.string.minimum_password));
                                }
                            }
                        }
                    });
            }
        });
    }
}

```

```

        } else {
            Toast.makeText(LoginActivity.this,
getString(R.string.auth_failed), Toast.LENGTH_LONG).show();
        }
    } else {
        Intent intent = new
Intent(LoginActivity.this, ChooseScreenActivity.class);
        startActivity(intent);
        finish();
    }
}
});
}
});
}

@Override
public void onBackPressed() {
    moveTaskToBack(true);
}

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

@Override
protected void onStop() {
    super.onStop();
    mAuth.removeAuthStateListener(firebaseAuthListener);
}
}
}

```

SignupActivity

```
package com.example.fixxapp;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.content.Intent;

import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ProgressBar;
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.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;

import java.util.HashMap;

public class SignupActivity extends AppCompatActivity {

    private EditText inputEmail, inputPassword, mFirstName, mLastname, mPhone ;
    private Button btnSignIn, btnSignUp, btnResetPassword;
    private ProgressBar progressBar;
    private FirebaseAuth auth;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        try
        {
            this.getSupportActionBar().hide();
        }
        catch (NullPointerException e){}
        setContentView(R.layout.activity_signup);

        //Get Firebase auth instance
        auth = FirebaseAuth.getInstance();

        btnSignIn = (Button) findViewById(R.id.sign_in_button);
        btnSignUp = (Button) findViewById(R.id.sign_up_button);

        inputEmail = (EditText) findViewById(R.id.email);
        inputPassword = (EditText) findViewById(R.id.password);
        mFirstName = (EditText) findViewById(R.id.firstname);
        mLastname = (EditText) findViewById(R.id.lastname);

        mPhone = (EditText) findViewById(R.id.fphone);

        progressBar = (ProgressBar) findViewById(R.id.progressBar);
        btnResetPassword = (Button) findViewById(R.id.btn_reset_password);

        btnResetPassword.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new Intent(SignupActivity.this,
ResetPasswordActivity.class));
            }
        })
    }
}
```

```

});

btnSignIn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        finish();
    }
});

btnSignUp.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        String email = inputEmail.getText().toString().trim();
        String password = inputPassword.getText().toString().trim();
        final String firstrname = mFirstName.getText().toString();
        final String lastname = mLastname.getText().toString();

        final String phone = mPhone.getText().toString();

        if (TextUtils.isEmpty(email)) {
            Toast.makeText(getApplicationContext(), "כתובת הזן",
                Toast.LENGTH_SHORT).show();
            return;
        }
        if (!android.util.Patterns.EMAIL_ADDRESS.matcher(email).matches()) {
            Toast.makeText(getApplicationContext(), "כתובת הזן",
                Toast.LENGTH_SHORT).show();
            return;
        }

        if (TextUtils.isEmpty(password)) {
            Toast.makeText(getApplicationContext(), "את הזן",
                Toast.LENGTH_SHORT).show();
            return;
        }
        if (TextUtils.isEmpty(phone)) {
            Toast.makeText(getApplicationContext(), "מספר הכנס",
                Toast.LENGTH_SHORT).show();
            return;
        }
        if (TextUtils.isEmpty(firstrname)) {
            Toast.makeText(getApplicationContext(), "פרטי שם הזן",
                Toast.LENGTH_SHORT).show();
            return;
        }
        if (TextUtils.isEmpty(lastname)) {
            Toast.makeText(getApplicationContext(), "משפחה שם הזן",
                Toast.LENGTH_SHORT).show();
            return;
        }

        if (password.length() < 6) {
            Toast.makeText(getApplicationContext(), "קצרה הסיסמה",
                Toast.LENGTH_SHORT).show();
            return;
        }
        if (phone.length() != 10) {
            Toast.makeText(getApplicationContext(), "טלפון מספר הכנס",
                Toast.LENGTH_SHORT).show();
            return;
        }

        progressBar.setVisibility(View.VISIBLE);
        //create user
        auth.createUserWithEmailAndPassword(email, password)

```

```

        .addOnCompleteListener(SignupActivity.this, new
OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult>
task) {
                Toast.makeText(SignupActivity.this, "בהצלחה נרשמת!" ,
Toast.LENGTH_SHORT).show();
                progressBar.setVisibility(View.GONE);
                // If sign in fails, display a message to
the user. If sign in succeeds
                // the auth state listener will be notified
and logic to handle the
                // signed in user can be handled in the
listener.
                if (!task.isSuccessful()) {
                    Toast.makeText(SignupActivity.this, "נכשל אימות. " +
task.getException(),
                        Toast.LENGTH_SHORT).show();
                } else {
                    String user_id =
auth.getCurrentUser().getUid();

                    HashMap hashMap = new HashMap();
                    hashMap.put("firstname", firsrtname);
                    hashMap.put("lastname", lastname);
                    hashMap.put("phone", phone);

                    DatabaseReference databaseReference =
FirebaseDatabase.getInstance().getReference().child("newUsers").child(user_i
d);

                    databaseReference.updateChildren(hashMap);
                    startActivity(new
Intent(SignupActivity.this, ChooseScreenActivity.class));
                    finish();
                }
            }
        });
    }

    @Override
    protected void onResume() {
        super.onResume();
        progressBar.setVisibility(View.GONE);
    }
}

```

ResetPasswordActivity

```
package com.example.fixxapp;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ProgressBar;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;

public class ResetPasswordActivity extends AppCompatActivity {

    private EditText inputEmail;
    private Button btnReset, btnBack;

    private FirebaseAuth auth;
    private ProgressBar progressBar;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        try
        {
            this.getSupportActionBar().hide();
        }
        catch (NullPointerException e){}
        setContentView(R.layout.activity_reset_password);

        inputEmail = (EditText) findViewById(R.id.email);
        btnReset = (Button) findViewById(R.id.btn_reset_password);
        btnBack = (Button) findViewById(R.id.btn_back);
        progressBar = (ProgressBar) findViewById(R.id.progressBar);

        auth = FirebaseAuth.getInstance();

        btnBack.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                finish();
            }
        });

        btnReset.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                String email = inputEmail.getText().toString().trim();

                if (TextUtils.isEmpty(email)) {
                    Toast.makeText(getApplicationContext(), "הרשום הדוא'ל את הזן", Toast.LENGTH_SHORT).show();
                    return;
                }
                if (!android.util.Patterns.EMAIL_ADDRESS.matcher(email).matches()) {
                    Toast.makeText(getApplicationContext(), "כתובת הזן", Toast.LENGTH_SHORT).show();
                }
            }
        });
    }
}
```



```

        return;
    }

    progressBar.setVisibility(View.VISIBLE);
    auth.sendPasswordResetEmail(email)
        .addOnCompleteListener(new
    OnCompleteListener<Void>() {
        @Override
        public void onComplete(@NonNull Task<Void> task)
    {
        if (task.isSuccessful()) {

            Toast.makeText(ResetPasswordActivity.this, "הסיסמה לאיפוס הוראות לך שלחנו"
            "שלך", Toast.LENGTH_SHORT).show();
        } else {

            Toast.makeText(ResetPasswordActivity.this, "נכשלה לאיפוס דוא'ל שליחת",
            Toast.LENGTH_SHORT).show();
        }

        progressBar.setVisibility(View.GONE);
    }
    });
}
}
}
}
}

```

ChooseScreenActivity

ChooseScreenActivity

```
package com.example.fixxapp;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;

import android.app.Dialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.location.Address;
import android.location.Geocoder;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;

import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import java.io.IOException;
import java.util.Calendar;
import java.util.List;
import java.util.Locale;

public class ChooseScreenActivity extends AppCompatActivity {
    private Button cust, prof, mLogout, mSetting;
    private TextView mHello;
    private TextView mDet, mKet, mRet;
    DatabaseReference myRefr;
    private ProgressBar spinner;

    private int backButtonCount;
    private FirebaseAuth mAuth;
    private FirebaseAuth.AuthStateListener firebaseAuthListener;

    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        try
        {
            this.getSupportActionBar().hide();
        }
        catch (NullPointerException e){}

        if(!CheckNetwork.isInternetAvailable(ChooseScreenActivity.this))
        //returns true if internet available
        {
            final AlertDialog.Builder builder1 = new
AlertDialog.Builder(ChooseScreenActivity.this);
            builder1.setMessage("לאינטרנט חיבור דורשת האפליקציה");
            builder1.setCancelable(true);
            builder1.setPositiveButton(
                "יציאה",
                new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog, int id)

```

```

{
    dialog.cancel();
    Intent intent = new Intent(Intent.ACTION_MAIN);
    intent.addCategory(Intent.CATEGORY_HOME);

intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP); /**Change Here**
    startActivity(intent);
    finish();
    System.exit(0);

    }
});

AlertDialog alert11 = builder1.create();
alert11.setCanceledOnTouchOutside(false);
alert11.setCancelable(false);

    alert11.show();
}
 mAuth = FirebaseAuth.getInstance();
  FirebaseAuthListener = new FirebaseAuth.AuthStateListener() {
      @Override
      public void onAuthStateChanged(@NonNull FirebaseAuth
firebaseAuth) {
          FirebaseUser user =
FirebaseAuth.getInstance().getCurrentUser();
          if (user == null) {
              Intent intent = new Intent(ChooseScreenActivity.this,
LoginActivity.class);
              startActivity(intent);
              finish();
              return;
          }
      }
  };
  //Toast.makeText(this, greeting, Toast.LENGTH_SHORT).show();
  setContentView(R.layout.activity_choose_screen);
  cust = (Button) findViewById(R.id.customer);
  prof = (Button) findViewById(R.id.proff);
  mSetting = (Button) findViewById(R.id.setting);
  mLogout = (Button) findViewById(R.id.logout);
  mHello = (TextView) findViewById(R.id.hello);
  mDet = (TextView) findViewById(R.id.details);
  mKet = (TextView) findViewById(R.id.ket);
  mRet = (TextView) findViewById(R.id.ret);

  spinner = (ProgressBar) findViewById(R.id.progressBar1);

  final String userId =
FirebaseAuth.getInstance().getCurrentUser().getUid();
  FirebaseDatabase database = FirebaseDatabase.getInstance();
  myRefr = database.getReference();
  spinner.setVisibility(View.VISIBLE);
  mHello.setText("[עוץ מ'נתן]...");

  Locale lHebrew = new Locale("he");

  final Geocoder mGeocoder = new Geocoder(ChooseScreenActivity.this,
lHebrew);
  myRefr.addValueEventListener(new ValueEventListener() {

      @Override
      public void onDataChange(DataSnapshot dataSnapshot) {
          String fname =
dataSnapshot.child("newUsers").child(userId).child("firstname").getValue().t
oString();

          String servicecall;

```

```

        String mdate = "";
        String prob = "";
        String city = "";
        String address="";
        Double longi,lati;

        if
        (dataSnapshot.child("customerPublish").child(userId).hasChildren()) {
            prob =
            dataSnapshot.child("customerPublish").child(userId).child("Problem").getValue().toString();

            longi =
            Double.parseDouble(dataSnapshot.child("customerPublish").child(userId).child("longi").getValue().toString());
            lati =
            Double.parseDouble(dataSnapshot.child("customerPublish").child(userId).child("lati").getValue().toString());

            List<Address> addresses = null;
            try {
                addresses = mGeocoder.getFromLocation(lati, longi, 1);
            } catch (IOException e) {
                e.printStackTrace();
            }

            if(addresses != null && addresses.size() > 0 ){
                Address maddress = addresses.get(0);
                if (maddress.getLocality() !=null){
                    city = "מיקום: "+maddress.getLocality();
                    if (maddress.getThoroughfare() !=null)
                        address = "
רמזוב:"+maddress.getThoroughfare();
                }
                mdate = "\n" +
                dataSnapshot.child("customerPublish").child(userId).child("date").getValue().toString();

                servicecall = "פעילה שירות קריאת לך יש";
            } else
                servicecall = "פעילה שירות קריאת לך אין";

            Calendar c = Calendar.getInstance();
            int hours = c.get(Calendar.HOUR_OF_DAY);
            String greeting = null;
            if (hours >= 0 && hours <= 12) {
                greeting = "טוב בוקר ";
            } else if (hours >= 12 && hours <= 16) {
                greeting = "טובים צהריים ";
            } else if (hours >= 16 && hours <= 21) {
                greeting = "טוב ערב ";
            } else if (hours >= 21 && hours <= 24) {
                greeting = "טוב לילה ";
            }
            mHello.setText(greeting + fname + ",");
            mDet.setText(servicecall);
            mKet.setText(prob);
            mRet.setText(city+address+mdate);
            spinner.setVisibility(View.GONE);
        }

        @Override
        public void onCancelled(DatabaseError databaseError) {
            Toast.makeText(getApplicationContext(),
            databaseError.getDetails(), Toast.LENGTH_SHORT).show();
        }
    });
    mSetting.setOnClickListener(new View.OnClickListener() {
        @Override

```

```

        public void onClick(View v) {
            Intent intent = new Intent(ChooseScreenActivity.this,
CustomerSettingActivity.class);
            startActivity(intent);
            finish();
            return;
        }
    });

    mLogout.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            FirebaseAuth.getInstance().signOut();
            Intent intent = new Intent(ChooseScreenActivity.this,
LoginActivity.class);
            startActivity(intent);
            finish();
            return;
        }
    });

    cust.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent(ChooseScreenActivity.this,
ClientActivity.class);
            startActivity(intent);
            finish();
            return;
        }
    });

    prof.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent(ChooseScreenActivity.this,
ProffDemo.class);
            startActivity(intent);
            finish();
            return;
        }
    });
}

@Override
public void onBackPressed() {
    if(backButtonCount >= 1)
    {
        Intent intent = new Intent(Intent.ACTION_MAIN);
        intent.addCategory(Intent.CATEGORY_HOME);
        intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        startActivity(intent);
    }
    else
    {
        Toast.makeText(this, "את לסגור כדי החזרה לחצן על שוב לחץ", Toast.LENGTH_SHORT).show();
        backButtonCount++;
    }
}

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

}

```

```

package com.example.fixxapp;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.app.Activity;
import android.content.Intent;
import android.graphics.Bitmap;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.util.HashMap;
import java.util.Map;

public class CustomerSettingActivity extends AppCompatActivity {

    private EditText mPhoneField, mFirstName, mLastName, mServiceCall;

    private Button mBack, mConfirm, mCancel;
    private TextView mServiceText;
    private ImageView mProfileImage;

    private FirebaseAuth mAuth;
    private DatabaseReference mCustomerDatabase, mCustomerDatabaseP;

    private String userID;
    private String mName, mLast, mPhone, mService;

    private String mProfileImageUrl;

    private Uri resultUri;
    private boolean isPublish = false;

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

        try
        {
            this.getSupportActionBar().hide();
        }
        catch (NullPointerException e){}

        setContentView(R.layout.activity_customer_setting);

        mFirstName = (EditText) findViewById(R.id.firstname);
        mLastName = (EditText) findViewById(R.id.lastname);
        mPhoneField = (EditText) findViewById(R.id.phone);
        mServiceCall = (EditText) findViewById(R.id.servicecall);

```

```

        mServiceText = (TextView) findViewById(R.id.servicetext);

        mBack = (Button) findViewById(R.id.back);
        mConfirm = (Button) findViewById(R.id.confirm);
        mCancel = (Button) findViewById(R.id.cancel);

        mAuth = FirebaseAuth.getInstance();
        userID = mAuth.getCurrentUser().getUid();
        mCustomerDatabase =
        FirebaseDatabase.getInstance().getReference().child("newUsers").child(userID
        );

        mCustomerDatabaseP =
        FirebaseDatabase.getInstance().getReference().child("customerPublish").child
        (userID);

        getUserInfo();
    /*
        mProfileImage.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(Intent.ACTION_PICK);
                intent.setType("image/*");
                startActivityForResult(intent, 1);
            }
        });
    */
    mConfirm.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if (mPhoneField.length() != 10 ) {
                Toast.makeText(getApplicationContext(), "סלפון חספר הכנס",
                "טלפון 10 בעל תקני נייד", Toast.LENGTH_SHORT).show();
                return;
            }
            saveUserInformation();
            Toast.makeText(getApplicationContext(), "עודכנו הפרטים",
            "בהצלחה", Toast.LENGTH_SHORT).show();
            Intent intent = new Intent(CustomerSettingActivity.this,
            ChooseScreenActivity.class);
            startActivity(intent);
            finish();
        }
    });

    mBack.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent(CustomerSettingActivity.this,
            ChooseScreenActivity.class);
            startActivity(intent);
            finish();
        }
    });

    mCancel.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            DatabaseReference data =
            FirebaseDatabase.getInstance().getReference().child("customerPublish").child
            (userID);

            Intent i = new Intent(CustomerSettingActivity.this,
            CustomerSettingActivity.class); //your class

```

```

        startActivity(i);
        data.removeValue();
        Toast.makeText(getApplicationContext(), "הוסרה שירות קריאת",
בהצלחה", Toast.LENGTH_SHORT).show();

        finish();
    }
});

}

private void getUserInfo() {
    mCustomerDatabase.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(DataSnapshot dataSnapshot) {
            if(dataSnapshot.exists() &&
dataSnapshot.getChildrenCount()>0){
                Map<String, Object> map = (Map<String, Object>)
dataSnapshot.getValue();
                if(map.get("firstname")!=null){
                    mName = map.get("firstname").toString();
                    mFirstName.setText(mName);
                }
                if(map.get("lastname")!=null){
                    mLast = map.get("lastname").toString();
                    mLastName.setText(mLast);
                }
                if(map.get("phone")!=null){
                    mPhone = map.get("phone").toString();
                    mPhoneField.setText(mPhone);
                }
            }
        }

        @Override
        public void onCancelled(DatabaseError databaseError) {
        }
    });
    mCustomerDatabaseP.addValueEventListener(new ValueEventListener() {
        public void onDataChange(DataSnapshot dataSnapshot) {
            isPublish = dataSnapshot.hasChildren();
            if(isPublish)
            {
                Map<String, Object> map = (Map<String, Object>)
dataSnapshot.getValue();
                mServiceCall.setVisibility(View.VISIBLE);
                mServiceText.setVisibility(View.VISIBLE);
                mCancel.setVisibility(View.VISIBLE);
                if(map.get("Problam")!=null) {
                    mService = map.get("Problam").toString();
                    mServiceCall.setText(mService);
                }
            }
        }

        public void onCancelled(DatabaseError databaseError) {
        }
    });
}

private void saveUserInformation(){
    mName = mFirstName.getText().toString();
    mLast = mLastName.getText().toString();
    mPhone = mPhoneField.getText().toString();
}

```



```

        mService=mServiceCall.getText().toString();

        Map userInfo = new HashMap();
        userInfo.put("firstname", mName);
        userInfo.put("lastname", mLast);
        userInfo.put("phone", mPhone);
        mCustomerDatabase.updateChildren(userInfo);
        if (isPublish)
        {
            userInfo.put("Problem", mService);
            mCustomerDatabaseP.updateChildren(userInfo);

        }

    }

    @Override
    public void onBackPressed() {
        // Here you want to show the user a dialog box
        Intent intent = new Intent(CustomerSettingActivity.this,
ChooseScreenActivity.class);
        startActivity(intent);
        finish();
    }
}

```

```

package com.example.fixxapp;
import androidx.appcompat.app.AlertDialog;

import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import androidx.fragment.app.FragmentActivity;

import android.content.DialogInterface;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.location.Location;
import android.os.Build;
import android.os.Bundle;
import android.os.Looper;

import android.view.View;
import android.widget.Button;
import android.widget.EditText;

import android.widget.Toast;

import com.google.android.gms.location.FusedLocationProviderClient;
import com.google.android.gms.location.LocationCallback;
import com.google.android.gms.location.LocationRequest;
import com.google.android.gms.location.LocationResult;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.BitmapDescriptorFactory;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;
import java.util.HashMap;
import java.util.Map;

public class ClientActivity extends FragmentActivity implements
OnMapReadyCallback {
    private GoogleMap mMap;

    private Button mButton,mCancel,mBack;
    private FusedLocationProviderClient mFusedLocationClient;
    private SupportMapFragment mapFragment;

    Location mLastLocation;

    LocationRequest mLocationRequest;

    @Override
    protected void onCreate(Bundle savedInstanceState) {

        checkIfPublish();

```

```

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_client);

        mFusedLocationClient =
        LocationServices.getFusedLocationProviderClient(this);

        mapFragment = (SupportMapFragment)
        getSupportFragmentManager().findFragmentById(R.id.map);
        mapFragment.getMapAsync(this);
        // Get the application context
        mButton = (Button) findViewById(R.id.btn1);
        mCancel = (Button) findViewById(R.id.btn2);
        mBack = (Button) findViewById(R.id.back);

        mBack.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(ClientActivity.this,
                ChooseScreenActivity.class);
                startActivity(intent);
                finish();
            }
        });

        FirebaseDatabase mDatabase = FirebaseDatabase.getInstance();
        final String userId =
        FirebaseAuth.getInstance().getCurrentUser().getUid();

        final Map userInfo = new HashMap();

        FirebaseDatabase database = FirebaseDatabase.getInstance();
        DatabaseReference myRef =
        database.getReference().child("newUsers").child(userId);
        myRef.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {
                String phone =
                dataSnapshot.child("phone").getValue().toString();
                String fname =
                dataSnapshot.child("firstname").getValue().toString();
                String lname =
                dataSnapshot.child("lastname").getValue().toString();
                userInfo.put("phone", phone);
                userInfo.put("firstname", fname);
                userInfo.put("lastname", lname);
            }

            @Override
            public void onCancelled(DatabaseError databaseError) {
                Toast.makeText(getApplicationContext(),
                databaseError.getDetails(), Toast.LENGTH_SHORT).show();
            }
        });

        mButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                AlertDialog.Builder alert = new
                AlertDialog.Builder(ClientActivity.this);

                alert.setTitle("שירות קריאת");
            }
        });

```

```

        alert.setMessage("הבעיה תיאור:");

        final EditText input = new EditText(ClientActivity.this);
        alert.setView(input);
        alert.setPositiveButton("אישור", new
DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int
whichButton) {

                String mInput = input.getText().toString();
                String userId =
FirebaseAuth.getInstance().getCurrentUser().getUid();
                // String date = new Date().toString();

                String pattern = "תאריך:dd/MM/yyyy שעה: HH:mm";
                DateFormat df = new SimpleDateFormat(pattern);
                Date today = Calendar.getInstance().getTime();
                String todayAsString = df.format(today);

                DatabaseReference data =
FirebaseDatabase.getInstance().getReference("customerPublish").child(userId)
;

                LatLng myLocation = new
LatLng(mLastLocation.getLatitude(), mLastLocation.getLongitude());
                userInfo.put("lati", myLocation.latitude);
                userInfo.put("longi", myLocation.longitude);
                userInfo.put("Problem", mInput);
                userInfo.put("date", todayAsString);

                data.updateChildren(userInfo);
                // Do something with value!
                Toast.makeText(getApplicationContext(), "בוטלה קריאת",
Toast.LENGTH_SHORT).show();
                Intent intent = new Intent(ClientActivity.this,
ChooseScreenActivity.class);
                startActivity(intent);
                finish();

            }
        });
        alert.setNegativeButton("ביטול", new
DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int
whichButton) {

                // Canceled.
            }
        });
        alert.show();

    }

    });

    mCancel.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            DatabaseReference data =
FirebaseDatabase.getInstance().getReference().child("customerPublish").child
(userId);
            data.removeValue();
            Toast.makeText(getApplicationContext(), "בוטלה שירות קריאת",
Toast.LENGTH_SHORT).show();
            Intent intent = new Intent(ClientActivity.this,
ChooseScreenActivity.class);
            startActivity(intent);

```

```

        finish();
    }
    });
}

private void checkIfPublish() {
    FirebaseDatabase database = FirebaseDatabase.getInstance();
    String userId =
        FirebaseAuth.getInstance().getCurrentUser().getUid();

    DatabaseReference myRefi =
        database.getReference().child("customerPublish").child(userId);
    myRefi.addValueEventListener(new ValueEventListener() {

        @Override
        public void onDataChange(DataSnapshot dataSnapshot) {
            if (dataSnapshot.hasChildren()) {
                mButton.setVisibility(View.GONE);
                mCancel.setVisibility(View.VISIBLE);

                String pb =
                    dataSnapshot.child("Problem").getValue().toString();
                String dt =
                    dataSnapshot.child("date").getValue().toString();

                final AlertDialog.Builder builder1 = new
                    AlertDialog.Builder(ClientActivity.this);
                builder1.setTitle("אחרונה שירות קריאת:");

                builder1.setMessage(pb + "\n" + dt);
                builder1.setCancelable(true);
                builder1.setPositiveButton(
                    "אישור",
                    new DialogInterface.OnClickListener() {
                        public void onClick(DialogInterface dialog,
int id) {

                            dialog.cancel();
                        }
                    });

                AlertDialog alert11 = builder1.create();
                alert11.setCanceledOnTouchOutside(false);
                alert11.show();

            }
            else
            {
                mButton.setVisibility(View.VISIBLE);
                mCancel.setVisibility(View.GONE);
            }
        }

        @Override
        public void onCancelled(DatabaseError databaseError) {
            Toast.makeText(getApplicationContext(),
                databaseError.getDetails(), Toast.LENGTH_SHORT).show();
        }
    }
}

```

```

    });
}

@Override
public void onBackPressed() {
    // Here you want to show the user a dialog box
    Intent intent = new Intent(ClientActivity.this,
ChooseScreenActivity.class);
    startActivity(intent);
    finish();
}
@Override

public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;

    mLocationRequest = new LocationRequest();
    mLocationRequest.setInterval(1000);
    mLocationRequest.setFastestInterval(1000);

mLocationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);

    if (android.os.Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
        if (ContextCompat.checkSelfPermission(this,
android.Manifest.permission.ACCESS_FINE_LOCATION) ==
PackageManager.PERMISSION_GRANTED) {

            } else {
                checkLocationPermission();
            }
        }

        mFusedLocationClient.requestLocationUpdates(mLocationRequest,
mLocationCallback, Looper.myLooper());
        mMap.setMyLocationEnabled(true);
    }

    LocationCallback mLocationCallback = new LocationCallback() {
        @Override
        public void onLocationResult(LocationResult locationResult) {
            for (Location location : locationResult.getLocations()) {
                if (getApplicationContext() != null) {
                    mLastLocation = location;

                    LatLng latLng = new LatLng(location.getLatitude(),
location.getLongitude());
                    mMap.moveCamera(CameraUpdateFactory.newLatLng(latLng));
                    mMap.animateCamera(CameraUpdateFactory.zoomTo(11));
                    mMap.addMarker(new
MarkerOptions().position(latLng).title("המיקום
שלך").icon(BitmapDescriptorFactory.fromResource(R.mipmap.ic_location)));

                    // if (!getDriversAroundStarted)
                    //     getDriversAround();
                }
            }
        }
    };

    private void checkLocationPermission() {
        if (ContextCompat.checkSelfPermission(this,
android.Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
            if (ActivityCompat.shouldShowRequestPermissionRationale(this,
android.Manifest.permission.ACCESS_FINE_LOCATION)) {

```

```

        new android.app.AlertDialog.Builder(this)
            .setTitle("give permission")
            .setMessage("give permission message")
            .setPositiveButton("OK", new
DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface
dialogInterface, int i) {

                    ActivityCompat.requestPermissions(ClientActivity.this, new
String[]{android.Manifest.permission.ACCESS_FINE_LOCATION}, 1);
                }
            })
            .create()
            .show();
    }
    else{
        ActivityCompat.requestPermissions(ClientActivity.this, new
String[]{android.Manifest.permission.ACCESS_FINE_LOCATION}, 1);
    }
}
}
}

```

```

package com.example.fixxapp;

import android.content.DialogInterface;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.os.Build;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import androidx.fragment.app.FragmentActivity;

import com.appolica.interactiveinfowindow.InfoWindow;
import com.appolica.interactiveinfowindow.InfoWindowManager;
import com.appolica.interactiveinfowindow.fragment.MapInfoWindowFragment;
import com.example.fixxapp.fragments.FormFragment;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.model.BitmapDescriptorFactory;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.Marker;
import com.google.android.gms.maps.model.MarkerOptions;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

public class ProffDemo
    extends FragmentActivity
    implements InfoWindowManager.WindowShowListener,
        GoogleMap.OnMarkerClickListener {
    private Button button,mBack;
    private InfoWindow formWindow;
    private InfoWindowManager infoWindowManager;
    private FirebaseAuth.AuthStateListener firebaseAuthListener;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_map_fragment);
        button = (Button)findViewById(R.id.btn1) ;
        mBack = (Button) findViewById(R.id.back);

        firebaseAuthListener = new FirebaseAuth.AuthStateListener() {
            @Override
            public void onAuthStateChanged(@NonNull FirebaseAuth
firebaseAuth) {
                FirebaseUser user =
FirebaseAuth.getInstance().getCurrentUser();
                if (user == null) {
                    Intent intent = new Intent(ProffDemo.this,
LoginActivity.class);
                    startActivity(intent);
                    finish();
                    return;
                }
            }
        };
        mBack.setOnClickListener(new View.OnClickListener() {

```



```

        @Override
        public void onClick(View v) {
            Intent intent = new Intent(ProffDemo.this,
ChooseScreenActivity.class);
            startActivity(intent);
            finish();
        }
    });

    button.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent i = new Intent(ProffDemo.this, ProffDemo.class);
//your class
            startActivity(i);
            finish();
        }
    });

    final MapInfoWindowFragment mapInfoWindowFragment =
        (MapInfoWindowFragment)
getSupportFragmentManager().findFragmentById(R.id.infoWindowMap);

    infoWindowManager = mapInfoWindowFragment.infoWindowManager();
    infoWindowManager.setHideOnFling(true);

    mapInfoWindowFragment.getMapAsync(new OnMapReadyCallback() {
        @Override
        public void onMapReady(final GoogleMap googleMap) {
            if (android.os.Build.VERSION.SDK_INT >=
Build.VERSION_CODES.M) {
                if (ContextCompat.checkSelfPermission(ProffDemo.this,
android.Manifest.permission.ACCESS_FINE_LOCATION) ==
PackageManager.PERMISSION_GRANTED) {

                } else {
                    checkLocationPermission();
                }
            }

            GpsTracker gps = new GpsTracker(ProffDemo.this);
            double lati = gps.getLatitude(); // returns latitude
            double longi = gps.getLongitude();
            LatLng latLng = new LatLng(lati, longi);
            googleMap.moveCamera(CameraUpdateFactory.newLatLng(latLng));
            googleMap.animateCamera(CameraUpdateFactory.zoomTo(9));
            googleMap.setMyLocationEnabled(true);

            // Marker Myloc= googleMap.addMarker(new
MarkerOptions().position(latLng).title("מיקום הלקוח").icon(BitmapDescriptorFactory.fromResource(R.mipmap.ic_location)).visib
le(false));
            // Marker markerMyLoc = googleMap.addMarker(new
MarkerOptions().position(latLng).title("מיקום הלקוח").icon(BitmapDescriptorFactory.fromResource(R.mipmap.ic_location));

            FirebaseDatabase database = FirebaseDatabase.getInstance();
            DatabaseReference myRef =
database.getReference().child("customerPublish");
            myRef.addValueEventListener(new ValueEventListener() {

                @Override
                public void onDataChange(DataSnapshot dataSnapshot) {

                    for (DataSnapshot item_snapshot :

```

```

dataSnapshot.getChildren()) {

    Double longi =
Double.parseDouble(item_snapshot.child("longi").getValue().toString());
    Double lati =
Double.parseDouble(item_snapshot.child("lati").getValue().toString());
    String Problem =
item_snapshot.child("Problem").getValue().toString();
    String phone =
item_snapshot.child("phone").getValue().toString();
    String fname =
item_snapshot.child("firstname").getValue().toString();
    String lname =
item_snapshot.child("lastname").getValue().toString();
    String date =
item_snapshot.child("date").getValue().toString();

    LatLng pos = new LatLng(lati, longi);
    googleMap.addMarker(new
MarkerOptions().position(pos).title("\n"+"\n טלפון: "+phone).snippet("שם:
"+fname+" "+lname+"\n"+date+"\n"+"\n תיאור: "+Problem).icon(BitmapDescriptorFactory.fromResource(R.mipmap.ic_flag)));

    }
}
@Override
public void onCancelled(DatabaseError databaseError) {

Toast.makeText(getApplicationContext(),databaseError.getDetails(),Toast.LENGTH
TH_SHORT).show();

}

});

googleMap.setOnMarkerClickListener(ProffDemo.this);

}

});

infoWindowManager.setWindowShowListener(ProffDemo.this);

}

@Override
public void onBackPressed() {
    Intent intent = new Intent(ProffDemo.this,
ChooseScreenActivity.class);
    startActivity(intent);
    finish();
}

@Override
public boolean onMarkerClick(Marker marker) {
    final int offsetX = (int)
getResources().getDimension(R.dimen.marker_offset_x);
    final int offsetY = (int)
getResources().getDimension(R.dimen.marker_offset_y);
    InfoWindow.MarkerSpecification markerSpec =
new InfoWindow.MarkerSpecification(offsetX, offsetY);
    String title = marker.getTitle();
    FormFragment.setMarker(marker);
    formWindow = new InfoWindow(marker, markerSpec, new FormFragment());
}

```

```

        InfoWindow infoWindow = null;

        infoWindow = formWindow;

        if (infoWindow != null) {
            infoWindowManager.toggle(infoWindow, true);
        }

        return true;
    }

    private void checkLocationPermission() {
        if (ContextCompat.checkSelfPermission(this,
            android.Manifest.permission.ACCESS_FINE_LOCATION) !=
            PackageManager.PERMISSION_GRANTED) {
            if (ActivityCompat.shouldShowRequestPermissionRationale(this,
                android.Manifest.permission.ACCESS_FINE_LOCATION)) {
                new android.app.AlertDialog.Builder(this)
                    .setTitle("give permission")
                    .setMessage("give permission message")
                    .setPositiveButton("OK", new
                        DialogInterface.OnClickListener() {
                            @Override
                            public void onClick(DialogInterface
                                dialogInterface, int i) {

                                    ActivityCompat.requestPermissions(ProffDemo.this, new
                                        String[]{android.Manifest.permission.ACCESS_FINE_LOCATION}, 1);
                                    }
                                })
                    .create()
                    .show();
            }
            else{
                ActivityCompat.requestPermissions(ProffDemo.this, new
                    String[]{android.Manifest.permission.ACCESS_FINE_LOCATION}, 1);
            }
        }
        @Override
        public void onWindowShowStarted(@NonNull InfoWindow infoWindow) {
            // Log.d("debug", "onWindowShowStarted: " + infoWindow);
        }

        @Override
        public void onWindowShown(@NonNull InfoWindow infoWindow) {
            // Log.d("debug", "onWindowShown: " + infoWindow);
        }

        @Override
        public void onWindowHideStarted(@NonNull InfoWindow infoWindow) {
            // Log.d("debug", "onWindowHideStarted: " + infoWindow);
        }

        @Override
        public void onWindowHidden(@NonNull InfoWindow infoWindow) {
            // Log.d("debug", "onWindowHidden: " + infoWindow);
        }
    }
}

```

```

package com.example.fixxapp;

import android.os.Bundle;

import androidx.annotation.NonNull;
import androidx.fragment.app.FragmentActivity;

import com.appolica.interactiveinfowindow.InfoWindow;
import com.appolica.interactiveinfowindow.InfoWindowManager;
import com.appolica.interactiveinfowindow.fragment.MapInfoWindowFragment;
import com.example.fixxapp.fragments.FormFragment;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.Marker;
import com.google.android.gms.maps.model.MarkerOptions;

public class MapFragmentActivity
    extends FragmentActivity
    implements InfoWindowManager.WindowShowListener,
        GoogleMap.OnMarkerClickListener {

    private static final String FORM_VIEW = "FORM_VIEW_MARKER";
    private InfoWindow formWindow;
    private InfoWindowManager infoWindowManager;

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

        final MapInfoWindowFragment mapInfoWindowFragment =
            (MapInfoWindowFragment)
                getSupportFragmentManager().findFragmentById(R.id.infoWindowMap);

        infoWindowManager = mapInfoWindowFragment.infoWindowManager();
        infoWindowManager.setHideOnFling(true);

        mapInfoWindowFragment.getMapAsync(new OnMapReadyCallback() {
            @Override
            public void onMapReady(GoogleMap googleMap) {

                // final Marker marker1 = googleMap.addMarker(new
                MarkerOptions().position(new LatLng(5, 5)).snippet(RECYCLER_VIEW));
                final Marker marker = googleMap.addMarker(new
                MarkerOptions().position(new LatLng(1, 1)).snippet(FORM_VIEW));

                final int offsetX = (int)
                    getResources().getDimension(R.dimen.marker_offset_x);
                final int offsetY = (int)
                    getResources().getDimension(R.dimen.marker_offset_y);

                final InfoWindow.MarkerSpecification markerSpec =
                    new InfoWindow.MarkerSpecification(offsetX,
                    offsetY);

                // FirebaseDatabase database =
                FirebaseDatabase.getInstance();
                // DatabaseReference myRef =
                database.getReference().child("customerPublish");
            }
        });
    }
}

```

```

        formWindow = new InfoWindow(marker, markerSpec, new
FormFragment());

googleMap.setOnMarkerClickListener(MapFragmentActivity.this);
    }
    });

    infoWindowManager.setWindowShowListener(MapFragmentActivity.this);

}

@Override
public boolean onMarkerClick(Marker marker) {
    InfoWindow infoWindow = null;
    switch (marker.getSnippet()) {
        case FORM_VIEW:
            infoWindow = formWindow;
            break;
    }

    if (infoWindow != null) {
        infoWindowManager.toggle(infoWindow, true);
    }

    return true;
}

@Override
public void onWindowShowStarted(@NonNull InfoWindow infoWindow) {
    // Log.d("debug", "onWindowShowStarted: " + infoWindow);
}

@Override
public void onWindowShown(@NonNull InfoWindow infoWindow) {
    // Log.d("debug", "onWindowShown: " + infoWindow);
}

@Override
public void onWindowHideStarted(@NonNull InfoWindow infoWindow) {
    // Log.d("debug", "onWindowHideStarted: " + infoWindow);
}

@Override
public void onWindowHidden(@NonNull InfoWindow infoWindow) {
    // Log.d("debug", "onWindowHidden: " + infoWindow);
}
}

```

```

package com.example.fixxapp.fragments;

import android.content.ActivityNotFoundException;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;

import com.example.fixxapp.R;
import com.google.android.gms.maps.model.Marker;

public class FormFragment extends Fragment {
    private TextView textView;
    private static Marker a;
    private static String phone;

    public static void setMarker(Marker marker) {
        a=marker;
    }

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

        final View view =
inflater.inflate(R.layout.info_window_form_fragment, container, false);
        TextView tv = view.findViewById(R.id.editTextFormInfoWindow); //id
defined in camera.xml

        tv.setText(a.getSnippet()+"\n");

        return view;
    }

    @Override
    public void onViewCreated(View view, @Nullable Bundle
savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);

        final View.OnClickListener onClickListener1 = new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String number = ("tel:" + a.getTitle());
                Intent mIntent = new Intent(Intent.ACTION_DIAL);
                mIntent.setData(Uri.parse(number));
                startActivity(mIntent);
            }
        };

        final View.OnClickListener onClickListener2 = new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                double lati = a.getPosition().latitude;
                double longi = a.getPosition().longitude;
                try {

```

```

String url = "waze://?ll=" + lati + "," + longi + "&navigate=yes\n";
Intent i = new Intent(Intent.ACTION_VIEW);
i.setData(Uri.parse(url));
startActivity(i);
    }

    catch ( ActivityNotFoundException ex )
    {
        // If Waze is not installed, open it in Google Play:
        Intent intent = new Intent( Intent.ACTION_VIEW,
Uri.parse( "market://details?id=com.waze" ) );
        startActivity(intent);
    }
}

};
final View.OnClickListener onClickListener3 = new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String wa = a.getTitle().substring(1);

        String url = "https://wa.me/" + "972" + wa;
        Intent i = new Intent(Intent.ACTION_VIEW);
        i.setData(Uri.parse(url));
        startActivity(i);

    }
};

view.findViewById(R.id.button2).setOnClickListener(onClickListener1);
view.findViewById(R.id.button1).setOnClickListener(onClickListener2);
view.findViewById(R.id.button3).setOnClickListener(onClickListener3);

    }
}

```

```

package com.example.fixxapp;

import android.Manifest;
import android.app.Service;
import android.content.Context;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.os.IBinder;
import android.util.Log;

import androidx.core.content.ContextCompat;

public class GpsTracker extends Service implements LocationListener {

    private final Context mContext;
    Location location;
    double latitude;
    double longitude;

    private static final long MIN_DISTANCE_CHANGE_FOR_UPDATES = 10;
    private static final long MIN_TIME_BW_UPDATES = 1000 * 60 * 1;
    protected LocationManager locationManager;

    public GpsTracker(Context context) {
        this.mContext = context;
        getLocation();
    }

    public Location getLocation() {
        try {
            locationManager = (LocationManager)
mContext.getSystemService(LOCATION_SERVICE);

            boolean isGPSEnabled =
locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER);
            boolean isNetworkEnabled =
locationManager.isProviderEnabled(LocationManager.NETWORK_PROVIDER);

            if (!isGPSEnabled && !isNetworkEnabled) {

                } else {

                    int hasFineLocationPermission =
ContextCompat.checkSelfPermission(mContext,
Manifest.permission.ACCESS_FINE_LOCATION);
                    int hasCoarseLocationPermission =
ContextCompat.checkSelfPermission(mContext,
Manifest.permission.ACCESS_COARSE_LOCATION);

                    if (hasFineLocationPermission ==
PackageManager.PERMISSION_GRANTED &&
hasCoarseLocationPermission ==
PackageManager.PERMISSION_GRANTED) {

                        ;
                    } else
                        return null;

                    if (isNetworkEnabled) {

```



```

locationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER,
MIN_TIME_BW_UPDATES, MIN_DISTANCE_CHANGE_FOR_UPDATES, this);

        if (locationManager != null)
        {
            location =
locationManager.getLastKnownLocation(LocationManager.NETWORK_PROVIDER);
            if (location != null)
            {
                latitude = location.getLatitude();
                longitude = location.getLongitude();
            }
        }

        if (isGPSEnabled)
        {
            if (location == null)
            {

locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER,
MIN_TIME_BW_UPDATES, MIN_DISTANCE_CHANGE_FOR_UPDATES, this);
                if (locationManager != null)
                {
                    location =
locationManager.getLastKnownLocation(LocationManager.GPS_PROVIDER);
                    if (location != null)
                    {
                        latitude = location.getLatitude();
                        longitude = location.getLongitude();
                    }
                }
            }
        }
    }
    catch (Exception e)
    {
        Log.d("@@@", ""+e.toString());
    }

    return location;
}

public double getLatitude()
{
    if(location != null)
    {
        latitude = location.getLatitude();
    }

    return latitude;
}

public double getLongitude()
{
    if(location != null)
    {
        longitude = location.getLongitude();
    }

    return longitude;
}

```

```

@Override
public void onLocationChanged(Location location)
{
}

@Override
public void onProviderDisabled(String provider)
{
}

@Override
public void onProviderEnabled(String provider)
{
}

@Override
public void onStatusChanged(String provider, int status, Bundle extras)
{
}

@Override
public IBinder onBind(Intent arg0)
{
    return null;
}

public void stopUsingGPS()
{
    if(locationManager != null)
    {
        locationManager.removeUpdates(GpsTracker.this);
    }
}
}

```

```

package com.example.fixxapp;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.widget.TextView;

import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.model.Marker;

public class CustomMarkerInfoWindowView implements
GoogleMap.InfoWindowAdapter {
    private Context context;
    public CustomMarkerInfoWindowView(Context context) {
        this.context = context.getApplicationContext();
    }

    @Override
    public View getInfoWindow(Marker arg0) {

        LayoutInflater inflater = (LayoutInflater)
context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        View v = inflater.inflate(R.layout.marker_info_window, null);

        // LatLng latLng = arg0.getPosition();

        String name = arg0.getTitle();
        String snip = arg0.getSnippet();

        TextView Mname = (TextView) v.findViewById(R.id.name);
        TextView Mphone = (TextView) v.findViewById(R.id.phone);
        //TextView MProblemDetail = (TextView)
v.findViewById(R.id.ProblamDetail);

        Mname.setText(name);
        Mphone.setText(snip);

        // MProblemDetail.setText("Problem Detail:"+ProblemDetail);

        return v;

    }

    @Override
    public View getInfoContents(Marker arg0)
    { return null;
    }
}

```

```

package com.example.fixxapp;

import android.content.Context;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.util.Log;

public class CheckNetwork {

    private static final String TAG = CheckNetwork.class.getSimpleName();

    public static boolean isInternetAvailable(Context context)
    {
        NetworkInfo info = (NetworkInfo) ((ConnectivityManager)
context.getSystemService(Context.CONNECTIVITY_SERVICE)).getActiveNetworkInfo
();

        if (info == null)
        {
            Log.d(TAG, "no internet connection");
            return false;
        }
        else
        {
            if(info.isConnected())
            {
                Log.d(TAG, "internet connection available...");
                return true;
            }
            else
            {
                Log.d(TAG, "internet connection");
                return true;
            }
        }
    }
}

```