**נושא העבודה:** שעון מעורר הפועל על מיקום

**סמל בגרות:** עבודת גמר תכנון ותכנות מערכות 883599

**שם המגיש:** מיכאל מספין

**כיתה: יב' 5**

**תאריך הגשה:** יום ראשון ז' באדר ב' התשע"ט2019.03.14

**שם המנחה:** לביא בונימוביץ'

**בית ספר:** מכללת אורט גבעת רם ע"ש יוסף חרמץ

**תוכן העניינים**

מבוא 3

מדריך משתמש 3

מטרת המערכת 4

תפעול המערכת שגיאה! הסימניה אינה מוגדרת.

תרשים זרימה בין המסכים 4

מסך פתיחה 5

מסך משחק 5

הרשאות 5

דרישות מיוחדות ומגבלות 5

גירסת Android מינימלית 5

מכשירים עליהם נבדקה המערכת 5

מדריך למתכנת 6

אלגוריתמים מרכזיים 6

אתגרים מרכזיים 6

עיצוב 6

תרשים מחלקות 6

Activities 6

MainActivity 6

GameActivity 6

Services 7

SoundService 7

VibratorService 7

Threads 7

AnimationThread 7

Content Providers 7

Broadcast Receivers 7

מחלקות המודל 7

מחלקה Ball 8

resources 8

Layouts 8

Menus 8

Drawables 8

קבצים 8

UserDetails.txt 8

סכמת Database 9

פרוטוקול תקשורת 9

נספחים 10

תיעוד JavaDoc שגיאה! הסימניה אינה מוגדרת.

כללי משחק (אופציונלי) 11

# Mapss :*שם הפרויקט*

# מה זה יהיה?

# אפליקציית ניווט ונוחות למשתמש יחיד.

# האפליקצייה תופעל בשפה האנגלית

# הפרויקט הוא מעיין מפה דיגידלית מגוגל מפות שבה המיקום שלך מופיע ומקום היעד שלך כאשר יש אינטרציה בין השניים בהתאם לרדיוס האפליקצייה תעורר תגובה (מוסיקה).

# הפרויקט מתבסס על גוגל מפות .

# למי זה מיועד?

# האפליקציה מיועדת לכול מי שמשתמש בתחבורה ציבורית, במיוחד לאנשים שנרדמים בה

# איך זה יעבוד?

# • האפליקציה תתאפשר לשימוש ליחיד

# • האפליקציה תכיל מספר שלבים

# • האפליקציה תכלול מספר מסכים: מסך טעינה, מסך פתיחה, מסך חיפוש,

# מסך תוצאה.

# •תוכנות לפיתוח המשחק: Studio Android

**מדריך למשתמש**

## מטרת המערכת

# מטרת המערכת: אפליקציית התראה

# • בהתחלה המשתמש נדרש לחפש ו לבחור מיקום בעולם אליו הוא רוצה להגיע .

# לאחר מכן, בהתאם למיקום הנבחר. יוצג למשתמש מסך בוא יהיה ניתן לבחור רדיוס (מרחק מסוים מהמיקום הנבחר\*),ולערוך החלטות לגבייו.

# אחר כך, המשתמש עובר למסך שבו מוצגים המיקום שלו הנוכחי והמיקום הנבחר על המפה.

# תוצג תיבה שתאמר את המרחק במטרים שנשאר למשתמש מהנק.

# בסופו של דבר כאשר מיקומו הנוכחי של המשתמש יהיה בתוך הרדיוס של המיקום הנבחר. הטלפון ירטוט ו יצלצל.

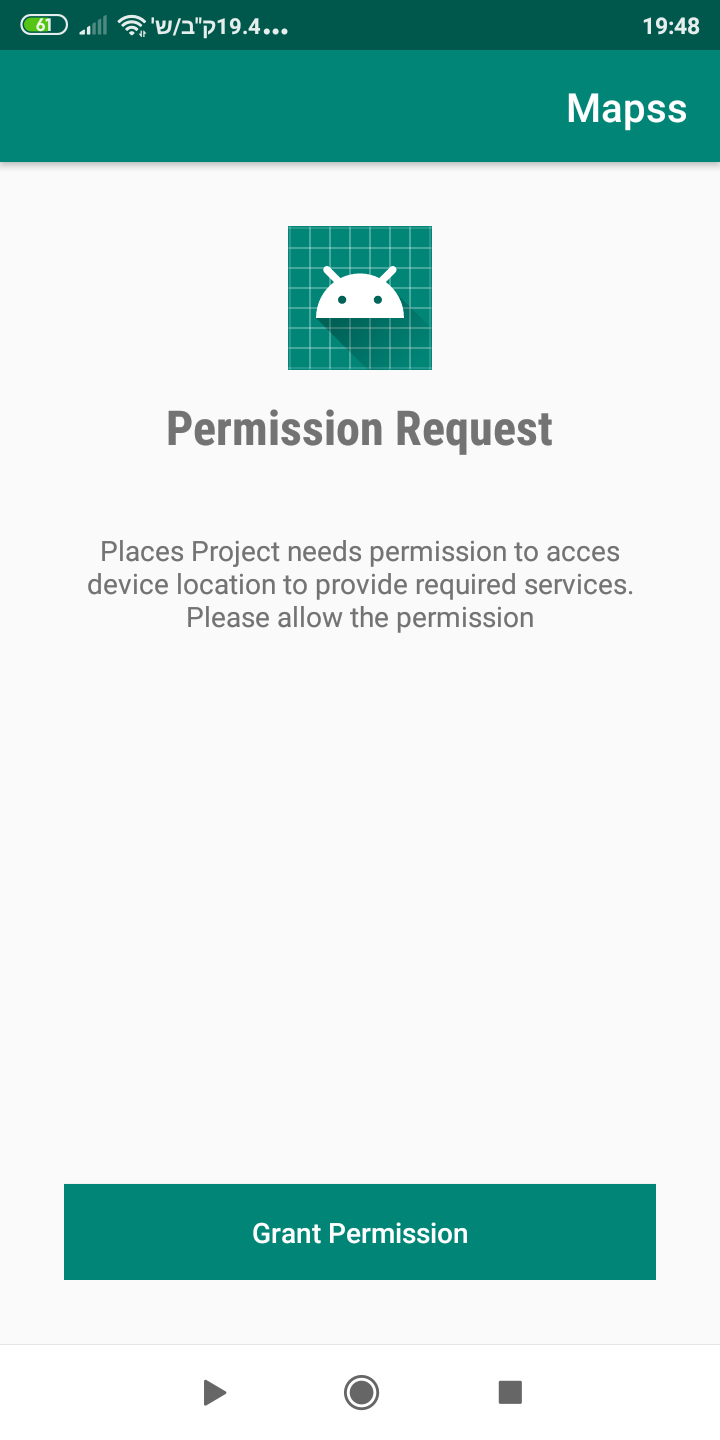
### תרשים זרימה בין המסכים





**צילומי המסכים**

### *מסך פתיחה*



מסך זה הוא המסך הראשון שהמשתמש רואה מסך זה הינו המסך הפתיחה ו כאן המשתמש מאשר את הרשאות של האפליקציה. בעזרת לחיצה על כפתור Grant Permisson/

### *מסך חיפוש*



מסך זה הוא המסך השני שהמשתמש רואה.

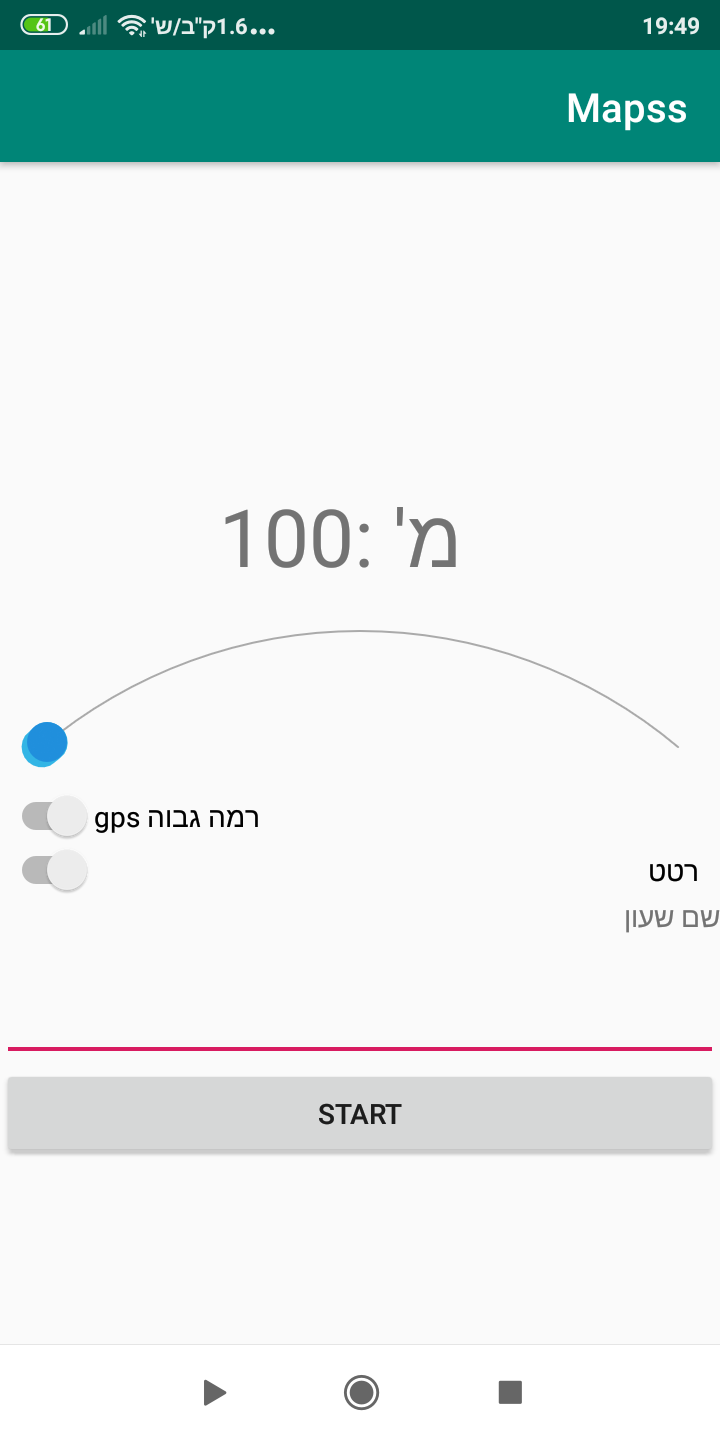
מסך זה הינו מסך חיפוש מיקום, וציון המיקום העדכני שלך .

במסך רואים את המיקום העדכני שלך , וכאשר המשתמש יחפש מיקום בSearch viev יופיע לו סמן מיקום נוסף.

במסך כפתור הSEARCH/ כפתור זה מופיע לאחר הקלדת מיקום .

ותפקידו הוא להעביר את המשתמש לדף הערכים ולהראות את המרחק בין שתי המיקומים.

### *מסך ערכים*



מסך זה הוא המסך השליש שהמשתמש רואה.

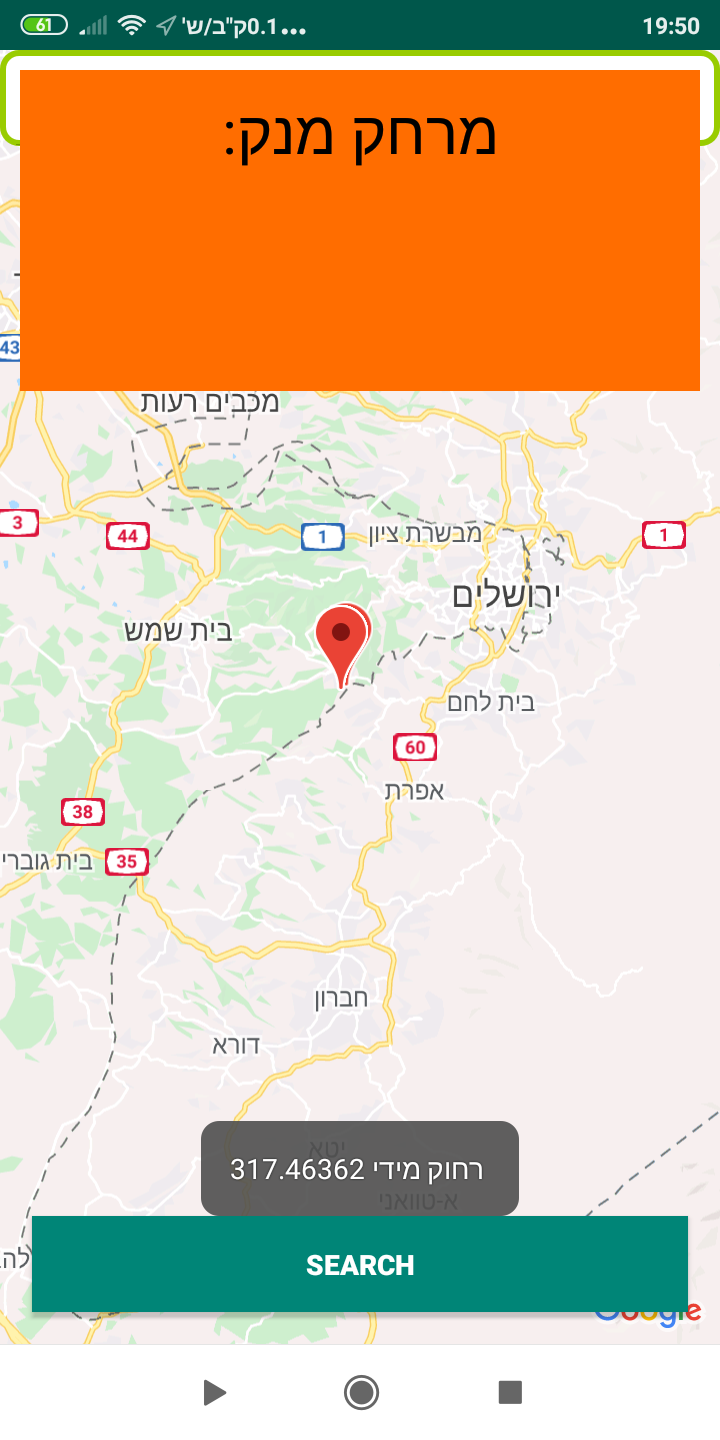
מסך זה הינו מסך ערכים מהמשתמש, המשתמש בחור רדיוס לנקודה הרצויה ועוד.. .

במסך רואים את הערכים שהמשתמש יכול להוסיף לאפליקציה , וכאשר המשתמש ישנה בSeek bar התוצאה במטרים תישמר ותעבור למסך הבאה.

במסך כפתור START/ כפתור זה מופיע מתחת לערכים .

ותפקידו הוא להעביר את המשתמש מדף הערכים לדף הפעולה ולהראות את המרחק בין שתי המיקומים.

### *מסך פעולה*



מסך זה הוא המסך הרביעי שהמשתמש רואה.

מסך זה הינו מסך שמטרטו להציג את שתי הנקודות מיקומך הנוכחי ונקודת הסיום , ולחשב את המרחק ביניהם על פי הרדיוס

במסך רואים את הערכים שהמשתמש שהוסיף , וכאשר המשתמש ישנה יכנס לטווח הרדיוס של נקןדת הסיום .

הטלופון ירטוט ויצלצל.

במסך רואים גם את המרחק החסר במטרים מהנקודה.

הרשאות

VIBRATE.permission.android – נדרשת כדי לאפשר שימוש ברטט במכשיר.

**ACCESS\_COARSE\_LOCATION.permission**– נדרשת כדי לאפשר שימוש במיקום במכשיר ברמת דיוק גבוהה.

**ACCESS\_FINE\_LOCATION.permission**– נדרשת כדי לאפשר שימוש במיקום במכשיר.

**android.permission.INTERNET**–

## דרישות מיוחדות ומגבלות

רישום לגוגל וקבלת מפתח המאפשר לממש את הapi של map .

הוספת implementation 'com.google.android.gms:play-services-maps:17.0.0'

implementation 'com.google.android.gms:play-services-location:17.0.0'

מגבלות בשימוש במערכת מערכת שמחייבת תקשורת WIFI, מערכת שאינה יכולה לרוץ על אמולטור אלא רק על מכשיר פיזי בגלל הצורך במיקום.

**גירסת Android מינימלית**

API 21 גירסת Android 5.0

**מכשירים עליהם נבדקה המערכת**

Android 9 המריץ LG Nexus 5

Android 8 המריץ Galaxy A7 2018

Android 8 המריץ Galaxy A7 2017

Android 8.1 המריץ Galaxy

*LG Nexus 5 המריץ Android 9*

# מדריך למתכנת

אלגוריתמים מרכזיים

• אלגוריתם מציאת המיקום העדכני

אלגוריתם זה מטפל בעדכון המיקום שלה משתמש. המיקום העדכני נקבע על ידי בקשת שנויי מיקום. והאזנה על המיקום המשתנה מתבצעת בדיקה האם יש מיקום חדש אשר משתנה ונבדק כל כמה שניות עד 5 שניות וכתוצאה מכך האלגוריתם דואג לעדכן את המיקום על מנת ליצור נק עדכנית. האלגוריתם ממומש במחלקהMapsActivity

• אלגוריתם חיפוש מיקום

אלגוריתם זה מטפל בחיפוש המיקום היעד של המשתמש. המיקום הסופי נקבע על ידי האזנה לשינוי הטקסט שמתבצעים הטקסט הסופי מומר ל כתובת ורצה חיפוש של הכתובת לפי השם.כאשר הכתובת נמצאת נלקחים מאיתה נקות הציון ומיקום חדש אשר יופיע במפה עם המיקום הנוכחי של המשתמש נוצר. . האלגוריתם ממומש במחלקהMapsActivity

## אתגרים מרכזיים

## • אלגוריתם חיפוש מיקום

תחילה לא ידעתי כיצד לקבל את ערך חיפוש המקום. לאחר מכן כאשר הבנתי שבמחלקה כתובות של גוגל יש לכל כתובת String משלו הבנתי שכל שעליי לעשות הוא להתאים את השניים ולארוך פעולה חיפוש והתאמה.

### עיצוב

### תרשים מחלקות

<יש ליצור תרשים מחלקות בעזרת plugin בשם simpleUMLCE. התקנה מפורטת בקישור זה:

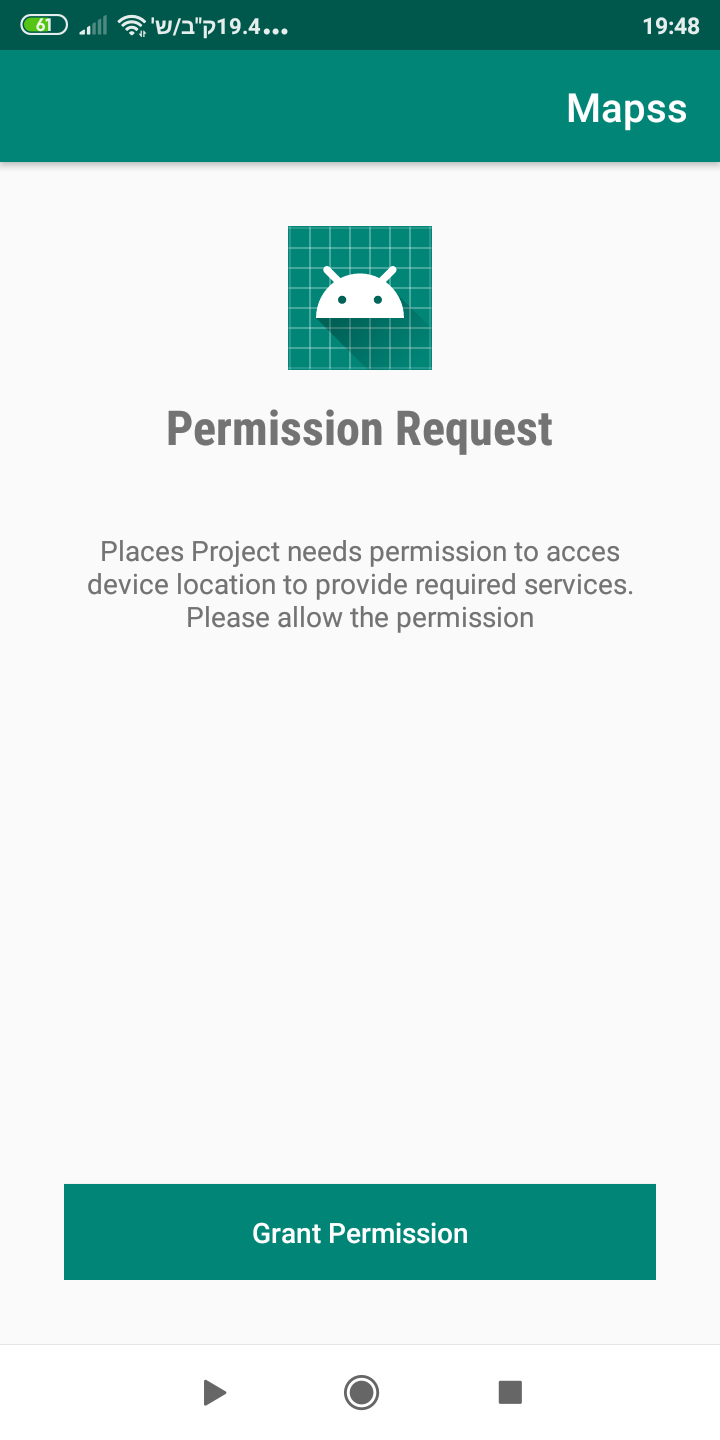
<http://stackoverflow.com/questions/17123384/how-to-generate-class-diagram-uml-on-android-studio/36823007#36823007>

אם התרשים מסובך אפשר ורצוי לפרק לכמה תרשימים (אפשר לשים כמה תרשימים שרוצים בתוך אותו פרויקט) ולשים בדפים שונים. לחילופין אם יש מחלקות שאינן מקושרות למחלקות אחרות, אפשר לשים אותן באזור נפרד באותו שרטוט ואז לצלם את המסך ע"י Alt-PrtScn, לחתוך את השוליים (Crop) ולשים בדף נפרד.

**זהו אחד החלקים הכי חשובים בספר הפרויקט שלכם. דאגו שהוא יהיה בנוי ומעומד כמו שצריך.>**

### רשימת Activities

Permission Activity/



מסך טעינת האפליקציה.

המעבר לactivity הבא נעשה אוטומטי כאשר ההרשאות מאושרים .

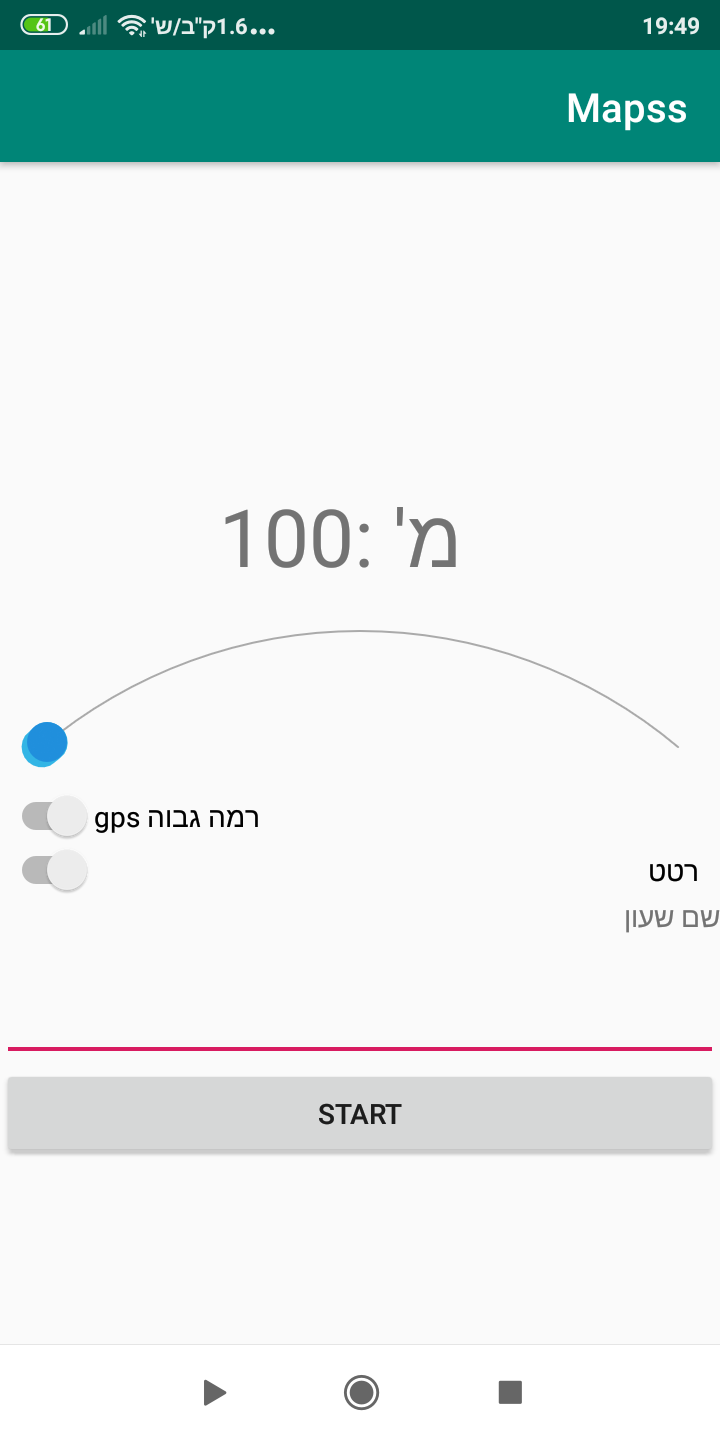
***activity\_maps***

**

מסך מפה

המעבר לactivity הבא נעשה כאשר המשתמש לוחץ הכפתור.

***activity\_main***



מסך בחירת נתונים.

המעבר לactivity הבא נעשה כאשר המשתמש לוחץ על כפתור START .

#### 

### Services

#### *VibratorService*

*שירות של המערכת, המפעיל את הרטט במכשיר. נעשה בו שימוש ב-GameActivity כדי לסמן פסילה.*

### Threads

### *AnimationThread*

*מנהל את האנימציה במשחק. מופעל ע"י ה-CustomView. מתקשר עם ה- CustomView ע"י SurfaceHolder holder*.

### 

### Broadcast Receivers

Receivers Broadcast Alarm זה מופעל כאשר המיקום העדכני הגיע ליעד.ישמע צלצול ורטט.

### מחלקות המודל

#### *מחלקה Mapss*

*מממשת מפה ובה קוראים כל הפעולות*

*מחלקה Main*

*מחלקה זו מייצגת את הנתונים. המחלקה שומרת את המיקום בו רוצה במשתמש לקבוע טווח ועוד,*

#### 

**משאבים (resources)**

#### Layouts

#### MainActivityל מתקשר Activity\_main1.xml

#### MapsActivityל מתקשר Activity\_maps.xml

#### PermissionActivityל מתקשר Activity\_Permisson.xml

#### MapsActivityל מתקשר Fregment\_First.xml

#### 

**סיכום אישי**

<בפרק הזה יופיע סיכום אישי של מבצע הפרויקט. נקודות ההתייחסות האפשריות:

הצדדים החזקים והחלק המיוחד בפרויקט:

* רעיון יצרתי ופותר בעיה לקהל יעד נרחב
* מבוסס על גוגל מפות
* בא בשילוב טוב עם אפליקציות מסוג מוביט וכדומה

הצדדים החלשים והחלק המסובך בפרויקט:

* עידכון המיקום יוצר מרקר חדש כל פעם במקום לעדכן את הקודם מה שיוצר חוויה לא נוחה לעין.
* עידכון מיקום
* מימוש מפות

היה ניתן לשנות בפרויקט:

* דרך החיפוש
* דרך עידכון המיקום
* תצוגת נק הסיום

היה ניתן להוסיף לפרויקט:

* תפריט
* היסטרית חיפוש
* משתמש
* שעונים שמורים

במהלך ביצוע הפרויקט למדתי לא לוותר ותמיד להמשיך.אף פעם לא להרים ידיים למדתי כיצד להתנהל ולעבוד עם פרויקטים גדולים,וכמה שעבודה מסודרת היא חשובה כאשר העובדה שלך גדולה. לסיכום למדתי מהפרוייקט המון על תכנות אנדרואיד, חשיבות העיצוב והתכנון, למדתי על

מימוש ממשקים. פרגמנטים קונפגירציה רשתות, והכול בעצמי .

# נספחים

**תדפיסי קוד של כל המחלקות**

import android.Manifest;

import android.content.DialogInterface;

import android.content.Intent;

import android.content.pm.PackageManager;

import android.net.Uri;

import android.os.Bundle;

import android.provider.Settings;

import android.view.View;

import android.widget.Button;

import android.widget.Toast;

import androidx.appcompat.app.AlertDialog;

import androidx.appcompat.app.AppCompatActivity;

import androidx.core.content.ContextCompat;

import com.google.android.gms.common.util.Clock;

import com.karumi.dexter.Dexter;

import com.karumi.dexter.PermissionToken;

import com.karumi.dexter.listener.PermissionDeniedResponse;

import com.karumi.dexter.listener.PermissionGrantedResponse;

import com.karumi.dexter.listener.PermissionRequest;

import com.karumi.dexter.listener.single.PermissionListener;

public class PermissionsActivity extends AppCompatActivity

{

private Button btnGrant;

@Override

protected void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_permissions);

if(ContextCompat.checkSelfPermission(PermissionsActivity.this, Manifest.permission.ACCESS\_FINE\_LOCATION) == PackageManager.PERMISSION\_GRANTED){

startActivity(new Intent(PermissionsActivity.this, MapsActivity.class));

finish();

return;

}

btnGrant = findViewById(R.id.btn\_grant);

btnGrant.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Dexter.withActivity(PermissionsActivity.this)

.withPermission(Manifest.permission.ACCESS\_FINE\_LOCATION)

.withListener(new PermissionListener() {

@Override

public void onPermissionGranted(PermissionGrantedResponse response) {

startActivity(new Intent(PermissionsActivity.this, MapsActivity.class));

finish();

}

@Override

public void onPermissionDenied(PermissionDeniedResponse response) {

if(response.isPermanentlyDenied()){

AlertDialog.Builder builder = new AlertDialog.Builder(PermissionsActivity.this);

builder.setTitle("Permission Denied")

.setMessage("Permission to access device location is permanently denied. you need to go to setting to allow the permission.")

.setNegativeButton("Cancel", null)

.setPositiveButton("OK", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

Intent intent = new Intent();

intent.setAction(Settings.ACTION\_APPLICATION\_DETAILS\_SETTINGS);

intent.setData(Uri.fromParts("package", getPackageName(), null));

}

})

.show();

} else {

Toast.makeText(PermissionsActivity.this, "Permission Denied", Toast.LENGTH\_SHORT).show();

}

}

@Override

public void onPermissionRationaleShouldBeShown(PermissionRequest permission, PermissionToken token) {

token.continuePermissionRequest();

}

})

.check();

}

});

}

}

**import** androidx.annotation.NonNull;

**import** androidx.annotation.Nullable;

**import** androidx.appcompat.app.AlertDialog;

**import** androidx.core.app.ActivityCompat;

**import** androidx.fragment.app.Fragment;

**import** androidx.fragment.app.FragmentActivity;

**import** androidx.fragment.app.FragmentManager;

**import** androidx.fragment.app.FragmentTransaction;

**import** androidx.transition.FragmentTransitionSupport;

**import** android.Manifest;

**import** android.app.AlarmManager;

**import** android.app.PendingIntent;

**import** android.content.Context;

**import** android.content.DialogInterface;

**import** android.content.Intent;

**import** android.content.pm.PackageManager;

**import** android.location.Address;

**import** android.location.Geocoder;

**import** android.location.Location;

**import** android.location.LocationManager;

**import** android.os.Bundle;

**import** android.provider.Settings;

**import** android.view.View;

**import** android.widget.Button;

**import** android.widget.SearchView;

**import** android.widget.Toast;

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

**import** com.google.android.gms.common.api.GoogleApi;

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

**import** com.google.android.gms.location.FusedLocationProviderClient;

**import** com.google.android.gms.location.LocationListener;

**import** com.google.android.gms.location.LocationRequest;

**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.LatLng;

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

**import** com.google.android.gms.maps.model.MarkerOptions;

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

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

**import** com.karumi.dexter.Dexter;

**import** com.karumi.dexter.PermissionToken;

**import** com.karumi.dexter.listener.PermissionDeniedResponse;

**import** com.karumi.dexter.listener.PermissionGrantedResponse;

**import** com.karumi.dexter.listener.single.PermissionListener;

**import** java.io.IOException;

**import** java.util.List;

**public class** MapsActivity **extends** FragmentActivity **implements** OnMapReadyCallback,

GoogleApiClient.ConnectionCallbacks,

GoogleApiClient.OnConnectionFailedListener, LocationListener, View.OnClickListener {

**private** GoogleMap **map**;

**private** GoogleApiClient **mGoogleApiClient**;

**private** Location **mLocation**;

**private** LocationManager **mLocationManager**;

**private** LocationRequest **mLocationRequest**;

**private** com.google.android.gms.location.LocationListener **listener**;

**private long UPDATE\_INTERVAL** = 2000;

**private long FASTEST\_INTERVAL** = 5000;

**private** LocationManager **locationManager**;

**private** LatLng **latLng**;

**private boolean isPermission** = **false**;

SearchView **searchView**;

**float r**;

FragmentManager **fragmentManager**;

FragmentTransaction **fragmentTransaction**;

FirstFragment **firstFragment**;

**private** LatLng **searchLoca**;

**boolean aBoolean**=**false**;

Button **btnfind**;

Location **locationInsert** =**new** Location(**""**);

Location **locationSt** =**null**;

Location **locationEn** =**null**;

FusedLocationProviderClient **mFusedLocationClient**;

FragmentTransaction **fragmentTransitionSupport**;

@Override

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

**super**.onCreate(savedInstanceState);

**firstFragment**=**new** FirstFragment();

**fragmentManager**=getSupportFragmentManager();

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

SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager().findFragmentById(R.id.***map***);

**searchView** = findViewById(R.id.***SvS***);

**btnfind** = findViewById(R.id.***btn\_find***);

**btnfind**.setOnClickListener(**this**);

**searchView**.setOnQueryTextListener(**new** SearchView.OnQueryTextListener()

{

@Override

**public boolean** onQueryTextSubmit(String s) {

String location =**searchView**.getQuery().toString();

List<Address> addressList = **null**;

**if** (location !=**null** || !location.equals(**""**))

{

Geocoder geocoder = **new** Geocoder(MapsActivity.**this**);

**try** {

addressList = geocoder.getFromLocationName(location,1);

} **catch** (IOException e) {

e.printStackTrace();

}

Address address =addressList.get(0);

**searchLoca**=**new** LatLng(address.getLatitude(),address.getLongitude());

**locationInsert**.setLatitude(address.getLatitude());

**locationInsert**.setLongitude(address.getLongitude());

**locationEn**=**locationInsert**;

**map**.addMarker(**new** MarkerOptions().position(**searchLoca**).title(location));

**map**.moveCamera(CameraUpdateFactory.*newLatLngZoom*(**searchLoca**,10));

**isPermission** = **true**;

**if** (**isPermission**)

{

**btnfind**.setClickable(**true**);

**btnfind**.setText(**"Search"**);

}

}

**return false**;

}

@Override

**public boolean** onQueryTextChange(String s) {

**return false**;

}

});

mapFragment.getMapAsync(**this**);

**mFusedLocationClient** = LocationServices.*getFusedLocationProviderClient*(**this**);

*// Obtain the SupportMapFragment and get notified when the map is ready to be used.*

**mGoogleApiClient** = **new** GoogleApiClient.Builder(**this**)

.addConnectionCallbacks(**this**)

.addOnConnectionFailedListener(**this**)

.addApi(LocationServices.***API***)

.build();

**mLocationManager** = (LocationManager) **this**.getSystemService(Context.***LOCATION\_SERVICE***);

checkLocation();

}

**import** androidx.appcompat.app.AppCompatActivity;

**import** androidx.fragment.app.Fragment;

**import** androidx.transition.FragmentTransitionSupport;

**import** android.app.FragmentManager;

**import** android.app.FragmentTransaction;

**import** android.content.Context;

**import** android.content.Intent;

**import** android.os.Bundle;

**import** android.util.Log;

**import** android.view.View;

**import** android.widget.Button;

**import** android.widget.SeekBar;

**import** android.widget.TextView;

**import** com.google.android.gms.dynamic.SupportFragmentWrapper;

**import** com.marcinmoskala.arcseekbar.ArcSeekBar;

**import** com.marcinmoskala.arcseekbar.ProgressListener;

**public class** MainActivity **extends** AppCompatActivity **implements** ProgressListener, View.OnClickListener {

SeekBar **seekBarn**;

ArcSeekBar **arcSeekBar**;

TextView **num**;

**float radius** = 100;

Button **button**;

Communicator **c**;

@Override

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

**super**.onCreate(savedInstanceState);

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

**button** = findViewById(R.id.***bottomS***);

**num**=findViewById(R.id.***Num***);

**button**.setOnClickListener(**this**);

**arcSeekBar**=findViewById(R.id.***SeekBar***);

**arcSeekBar**.setOnProgressChangedListener(**this**);

}

@Override

**public void** invoke(**int** i)

{

**radius**=i;

**num**.setText(**" מ' :"**+i+**""**);

}

@Override

**public void** onClick(View view)

{

**if** (**button**==view)

{

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

intent.putExtra(**"key1"**,**radius**);

setResult(***RESULT\_OK***, intent);

}

finish();

}

}

**import** android.content.BroadcastReceiver;

**import** android.content.Context;

**import** android.content.Intent;

**import** android.media.Ringtone;

**import** android.media.RingtoneManager;

**import** android.os.Vibrator;

**import** android.widget.Toast;

**public class** Alarm **extends** BroadcastReceiver

{

**private** Ringtone **ringtone**;

@Override

**public void** onReceive(Context context, Intent intent)

{

**ringtone** = RingtoneManager.*getRingtone*(context, RingtoneManager.*getDefaultUri*(RingtoneManager.***TYPE\_RINGTONE***));

Toast.*makeText*(context,**" רחוק מידי"**, Toast.***LENGTH\_SHORT***).show();

Vibrator v = (Vibrator)context.getSystemService(context.***VIBRATOR\_SERVICE***);

**ringtone**.play();

v.vibrate(3000);

}

}