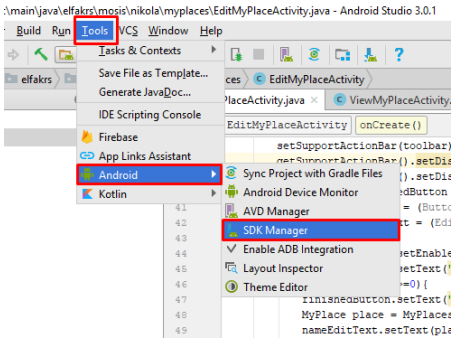
MOSIS - Laboratorijska vežba 4

Na poslednjoj laboratorijskoj vežbi je omogućen unos omiljenih lokacija i njihovo pregledavanje. Najznačajniji prikaz omiljenih lokacija je onaj koji vezan za kontekst njihove stvarne lokacije na mapi. Za prikaz objekata na mapi, *Google* nudi programerima svoje rešenje za rad sa mapama u vidu posebnog *map* fragmenta koji je moguće dodati u sopstveni *Activity*. Ovim je značajno olakšan rad sa mapama i nije više potrebno biti GIS ekspert da bi se prikaz mape ugradio u aplikacije. Ugrađena podrška za servise za lociranje takođe omogućava lakšu navigaciju mape, očitavanje trenutne lokacije i prikaz željene lokacije na mapi korišćenjem markera. U ovoj vežbi će akcenat biti na radu sa mapama čime aplikacija dobija svoju originalnu svrhu. Kao i prethodne laboratorijske vežbe, i ova se oslanja na kod sa prethodnih laboratorijskih vežbi.

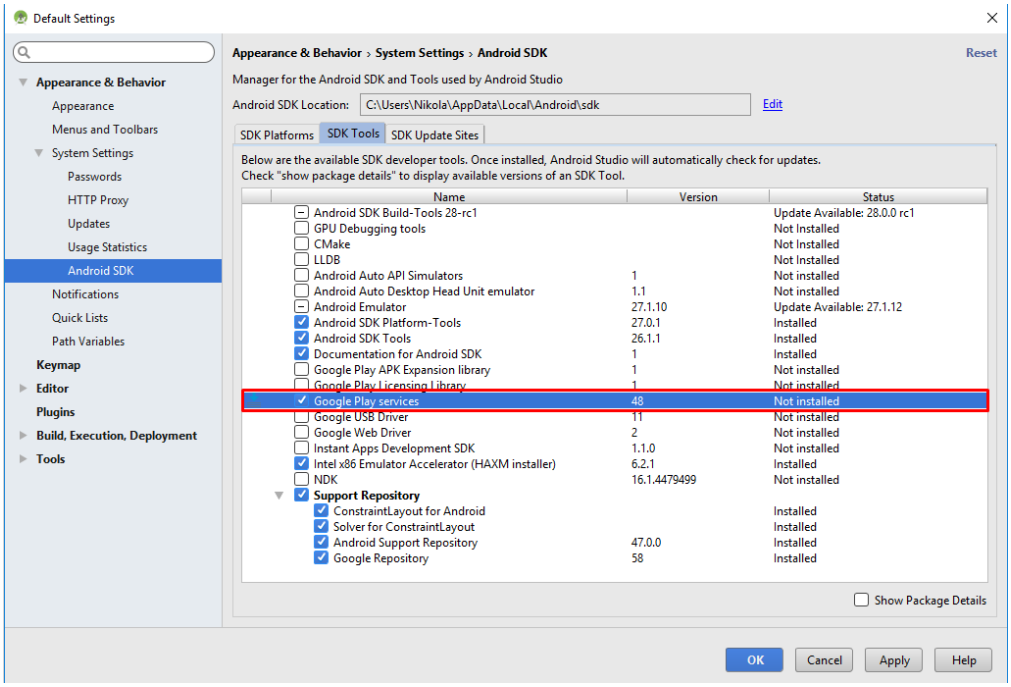
Ciljevi ove vežbe su:

1. Preuzimanje *Google Play Services* biblioteke i njeno uključivanje u projekat
2. Registrovanje debug ključa za rad sa mapama.
3. Priprema AndroidManifest.xml-a.
4. Omogućavanje rada sa Map fragmentom.
5. Kreiranje *MyPlacesMapActivity.java* klase i prikaz mape sa trenutnom lokacijom korisnika.
6. Dodavanje kontrola za unos i prikaz koordinata omiljene lokacije u *EditMyPlaceActivity.java* i podrška za izbor koordinata omiljene lokacije sa mape.
7. Prilagođavanje *MyPlacesMapActivity.java* za različita stanja rada (prikaz lokacija na mapi, centriranje prikaza omiljene lokacije na mapi, izbor koordinata omiljene lokacije na mapi).
8. Dodavanje markera omiljenih lokacija na mapi.
9. Prikaz određene omiljene lokacije centrirano na mapi.

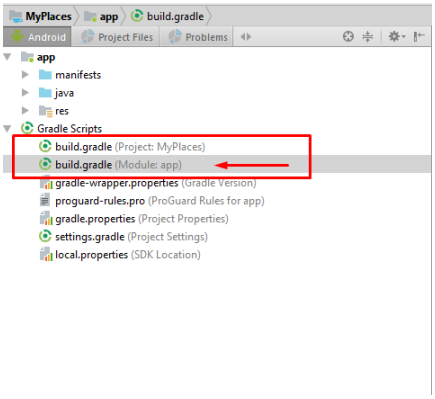
1. Preuzimanje Google Play Services biblioteke i njeno uključivanje u projekat.
   a. Ukoliko *google_play_services* biblioteka nije instalirana na računaru, instalirati je korišćenjem *Android SDK Manager*-a.
   b. Pokrenuti *Android SDK Manager* iz *Android Studio*-a klikom na *Tools->Android-> SDK Manager*.



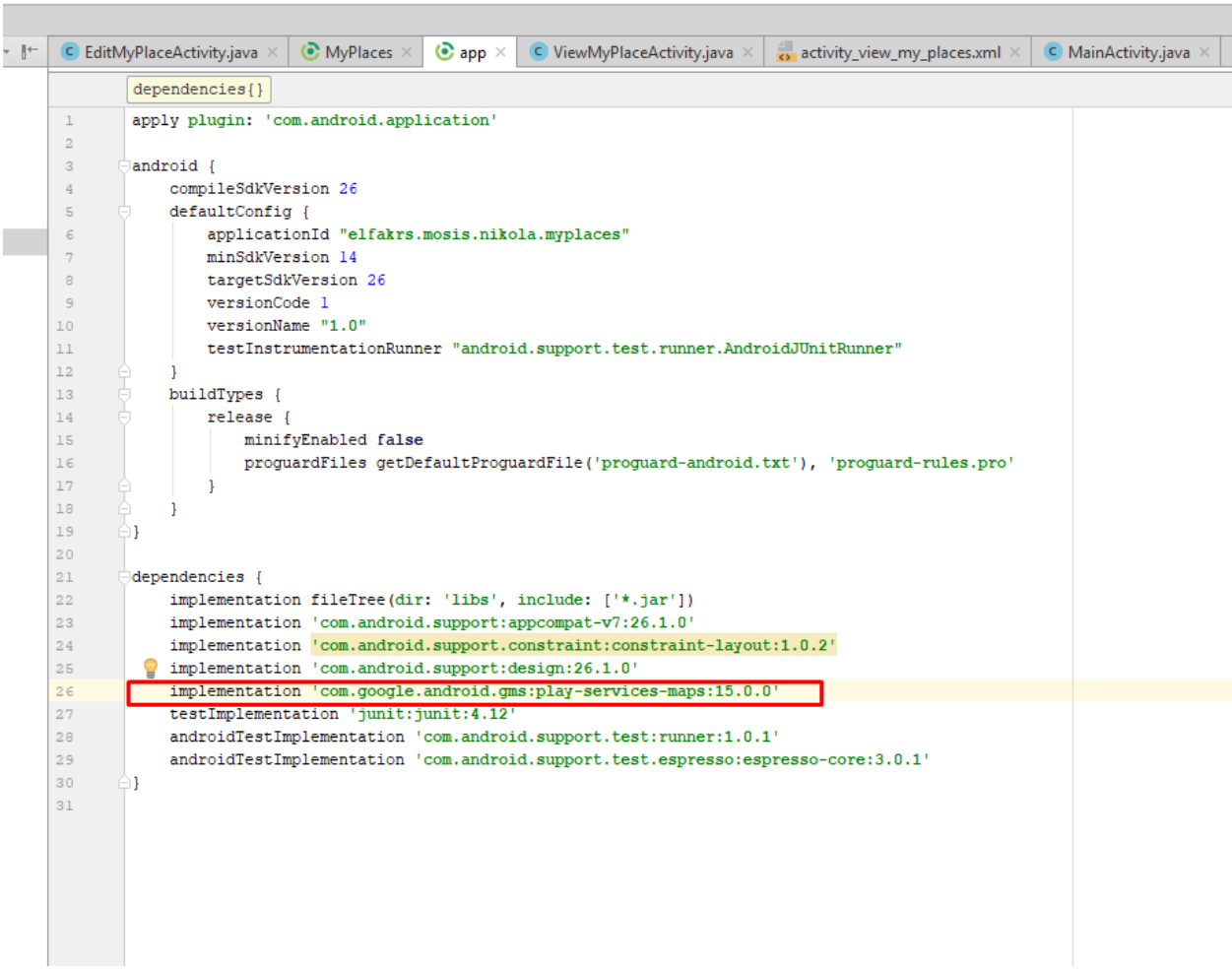
c. Na dnu liste u odeljku *Extras* izabrati opciju *Extras > Google Play services* a zatim kliknuti na *Apply* dugme.



d. Nakon što je *Google Play services* biblioteka preuzeta, potrebno ju je uključiti u *build.gradle* fajl projekta. Po pravilu, android projekat ima dva *build.gradle* fajla, jedan je na nivou čitavog projekta, dok drugi odgovara aplikaciji.

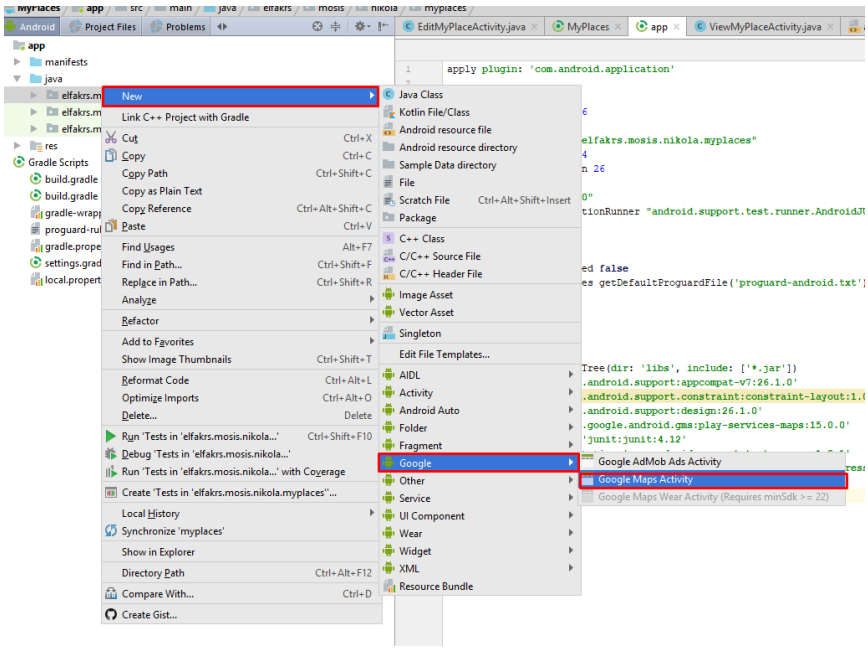


e. Otvoriti *build.gradle* fajl koji odgovara modulu aplikacije (Module:app) i dodati: *implementation 'com.google.android.gms:play-services-map:15.0.0'.*

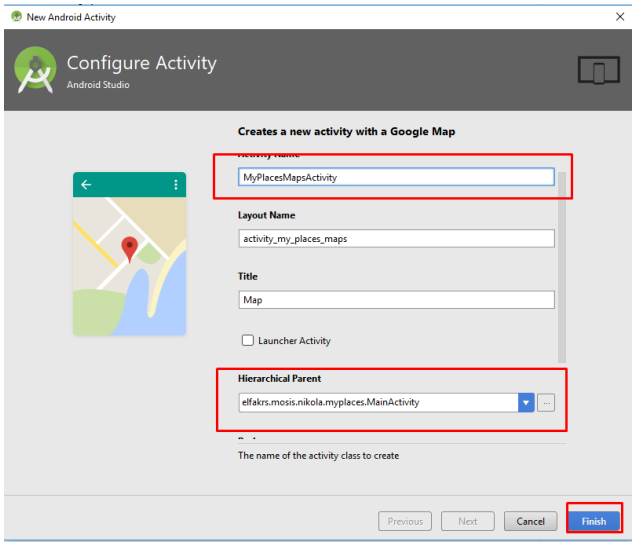


Detaljno uputstvo je moguće naći na adresi:
https://developers.google.com/android/guides/setup

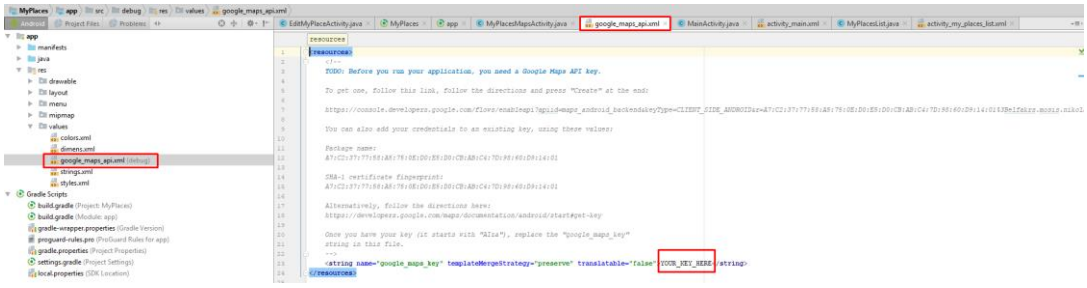
2. Dodavanje Google Maps Activity-a u MyPlaces projekat.
    a. Potrebno je pokrenuti dodavanje novog Activity-a u projekat.



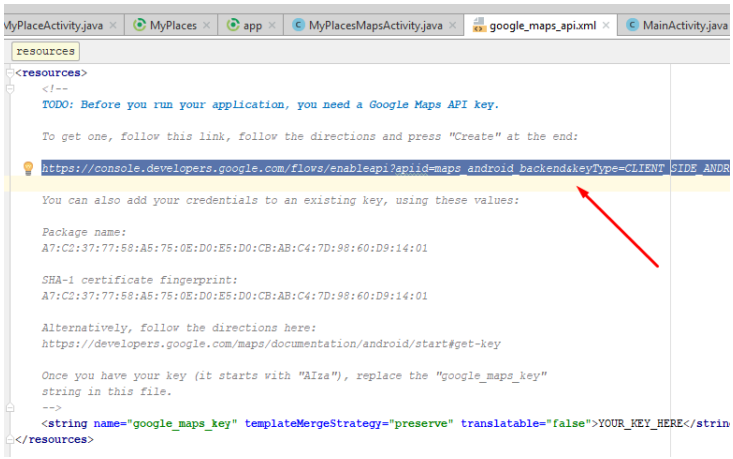
b. Podesiti ime *Activity*-a kao i hijerarhijski roditeljski *Activity*.



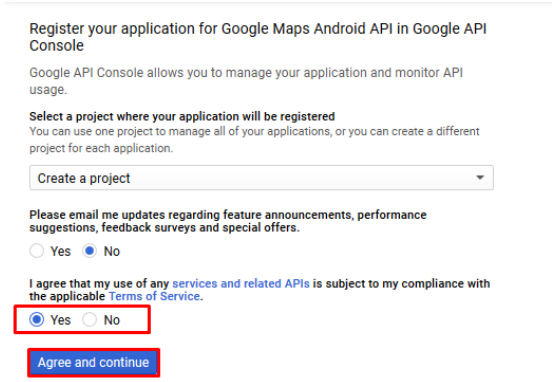
c. Nakon kreiranja *Map Activity*-a korisniku se automatski otvara google_maps_api.xml fajl sa uputstvom za registrovanje API ključa.



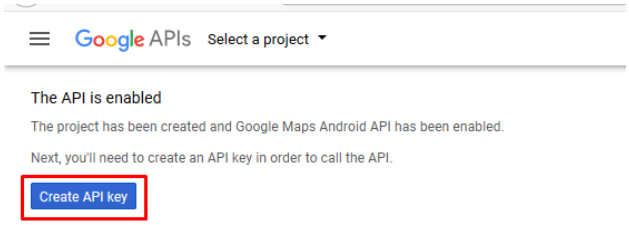
d. Najlakši način za registraciju je otvaranjem linka koji se nalazi u google_maps_api.xml fajl-u. Potrebno je da korisnik bude registrovan svojim google nalogom u izabranom *browser*-u.



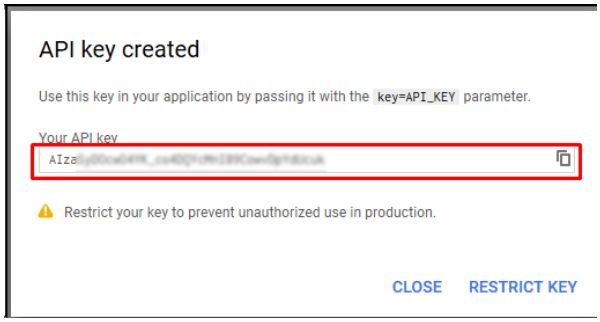
e. Prihvatiti uslove korišćenja.



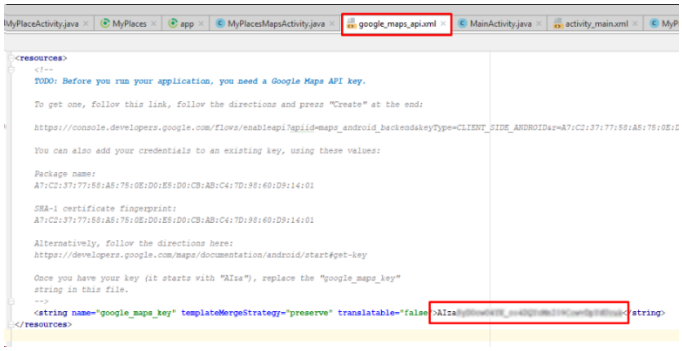
f. Nakon toga je potrebno kreirati *API KEY*.



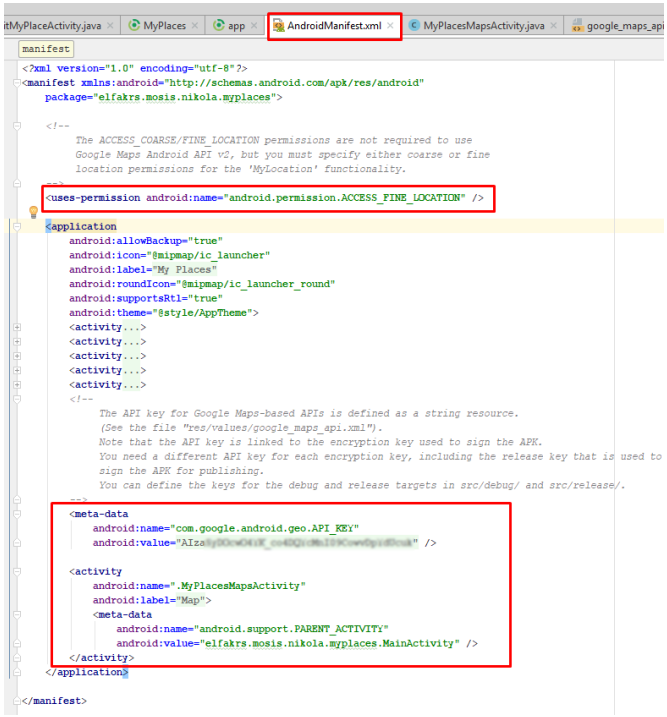
g. Ključ će biti prikazan korisniku u posebnom dijalogu.



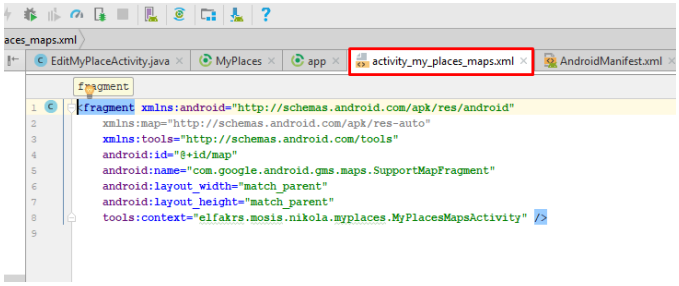
h. Dobijeni ključ iskopirati u google_maps_api.xml fajl.

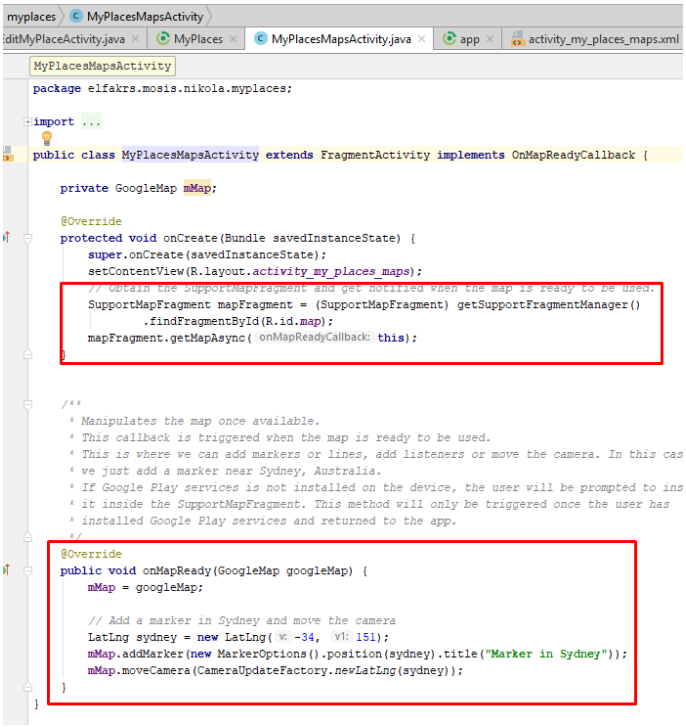


i. Izgled AndroidManifest fajla je promenjen dodavanjem *GoogleMap Activity*-a. Moguće je uočiti da je dodat zahtev za pristup finoj lokaciji kao i API ključ.



j. Potrebno je pregledati kreirani kod u *activity_my_places_maps.xml* i *MyPlacesMapsActivity*.





```
package elfakrs.mosis.nikola.myplaces;

import ...

public class MyPlacesMapsActivity extends FragmentActivity implements OnMapReadyCallback {

    private GoogleMap mMap;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_my_places_maps);
        // Obtain the SupportMapFragment and get notified when the map is ready to be used.
        SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()
                .findFragmentById(R.id.map);
        mapFragment.getMapAsync( onMapReadyCallback: this);
    }


    /**
     * Manipulates the map once available.
     * This callback is triggered when the map is ready to be used.
     * This is where we can add markers or lines, add listeners or move the camera. In this cas
     * we just add a marker near Sydney, Australia.
     * If Google Play services is not installed on the device, the user will be prompted to ins
     * it inside the SupportMapFragment. This method will only be triggered once the user has
     * installed Google Play services and returned to the app.
     */
    @Override
    public void onMapReady(GoogleMap googleMap) {
        mMap = googleMap;

        // Add a marker in Sydney and move the camera
        LatLng sydney = new LatLng( v: -34,  v1: 151);
        mMap.addMarker(new MarkerOptions().position(sydney).title("Marker in Sydney"));
        mMap.moveCamera(CameraUpdateFactory.newLatLng(sydney));
    }
}
```

k. U cilju testiranja dodatog Google Maps Activity-a, potrebno je dodati odgovarajući poziv u *MainActivity*-i.

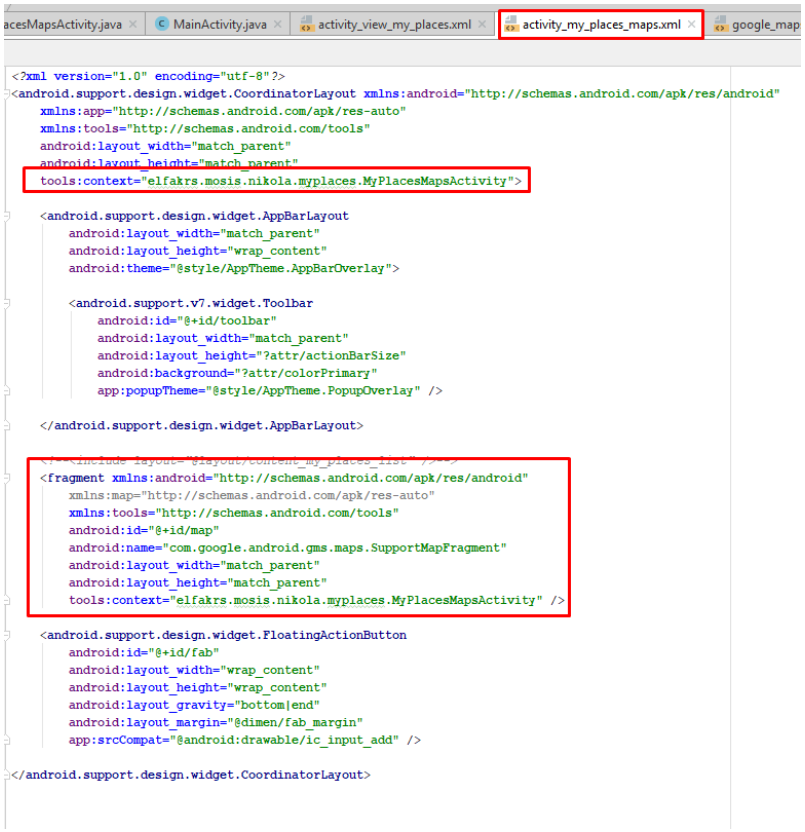
```
        setSupportActionBar(toolbar);

        FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
        fab.setOnClickListener((view) → {
                Intent i = new Intent( packageContext: MainActivity.this, EditMyPlaceActivity.class);
                startActivityForResult(i, NEW_PLACE);
        });
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }

    static int NEW_PLACE = 1;
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();

        //noinspection SimplifiableIfStatement
        if (id == R.id.show_map_item) {
            Intent i = new Intent( packageContext: this, MyPlacesMapsActivity.class);
            startActivity(i);
        } else if (id == R.id.new_place_item){
            Intent i = new Intent( packageContext: this, EditMyPlaceActivity.class);
            startActivityForResult(i, NEW_PLACE);
        } else if (id == R.id.my_places_list_item){
            Intent i = new Intent( packageContext: this, MyPlacesList.class);
```

l. Potrebno je kreirati emulator sa odgovarajućom verzijom Android-a tako da podržava Google API. Ukoliko je potrebno izvršiti ažuriranje Google Play Services, neophodno je korišćenje Google naloga.

m. Dobijena mapa zauzima celokupni ekran uređaja. U cilju dodavanja Toolbar-a, potrebno je dodati odgovarajuće stavke u layout fajlove. Najlaše je to uraditi kopiranjem iz drugih postojećih klasa koje su već dostupne. Iskopirati celokupan sadržaj iz *MainActivity*-a sadržaj layout fajla activity_my_places_list.xml. Potrebno je zameniti označene delove.

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="elfakrs.mosis.nikola.myplaces.MainActivity">

    <android.support.design.widget.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:theme="@style/AppTheme.AppBarOverlay">

        <android.support.v7.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            android:background="?attr/colorPrimary"
            app:popupTheme="@style/AppTheme.PopupOverlay" />

    </android.support.design.widget.AppBarLayout>

    <include layout="@layout/content_main" />

    <android.support.design.widget.FloatingActionButton
        android:id="@+id/fab"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom|end"
        android:layout_margin="16dp"
        app:srcCompat="@android:drawable/ic_input_add" />

</android.support.design.widget.CoordinatorLayout>
```

n. Umesto taga koji uključuje konktent fajl, iskopirato postojeći fragment. Izgled fajla activity_my_places_maps.xml treba da bude kao što je prikazano na slici.



o. Kreirati novi fajl menija pod nazivom menu_my_places_maps.xml i podesiti da izgleda kao na datoj slici:

```xml
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    tools:context="elfakrs.mosis.nikola.myplaces.MainActivity">
    <item
        android:id="@+id/new_place_item"
        android:orderInCategory="100"
        android:title="@string/new_place"
        app:showAsAction="ifRoom"
        android:icon="@drawable/ic_add_white_24dp"/>
    <item
        android:id="@+id/about_item"
        android:orderInCategory="100"
        android:title="@string/about"
        app:showAsAction="ifRoom"
        android:icon="@drawable/ic_info_outline_white_24dp"/>
</menu>
```

p. Dodati kod u *MyPlacesMapsActivity* klasu.

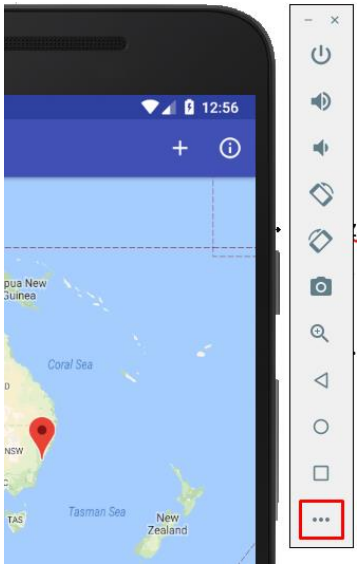
```java
import ...

public class MyPlacesMapsActivity extends AppCompatActivity implements OnMapReadyCallback {

    private GoogleMap mMap;
    static int NEW_PLACE=1;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_my_places_maps);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        getSupportActionBar().setDisplayShowHomeEnabled(true);

        FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
        fab.setOnClickListener((view) -> {
                Intent i = new Intent( packageContext: MyPlacesMapsActivity.this, EditMyPlaceActivity.class);
                startActivityForResult(i, NEW_PLACE);
        });
        // Obtain the SupportMapFragment and get notified when the map is ready to be used.
        SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()
                .findFragmentById(R.id.map);
        mapFragment.getMapAsync( onMapReadyCallback: this);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.menu_my_places_maps, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();

        //noinspection SimplifiableIfStatement
        if (id == R.id.new_place_item){
            Intent i = new Intent( packageContext: this, EditMyPlaceActivity.class);
            startActivityForResult(i,  requestCode: 1);
        } else if (id == R.id.about_item) {
            Intent i = new Intent( packageContext: this, About.class);
            startActivity(i);
        } else if (id == android.R.id.home) {
            finish();
        }

        return super.onOptionsItemSelected(item);
    }
```

q. Potrebno je dodati i u *AndroidManifest.xml* odgovarajuću temu za *Activity* kako bi *Toolbar* bio dostupan.

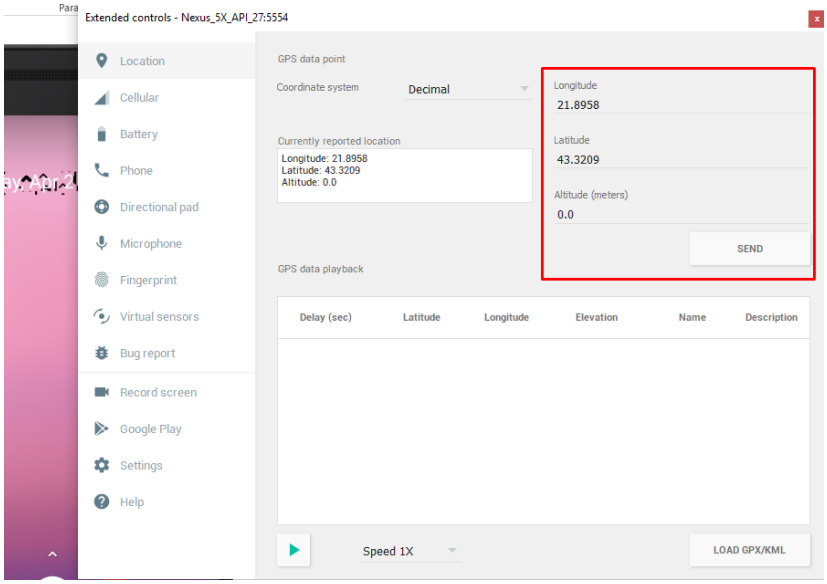
```
        You can define the keys for the debug and release targets in src/debug/ and
    -->
    <meta-data
        android:name="com.google.android.geo.API_KEY"
        android:value="@string/google_maps_key" />

    <activity
        android:name=".MyPlacesMapsActivity"
        android:label="@string/title_activity_my_places_maps"
        android:theme="@style/AppTheme.NoActionBar" >
        <meta-data
            android:name="android.support.PARENT_ACTIVITY"
            android:value="elfakrs.mosis.nikola.myplaces.MainActivity" />
    </activity>
</application>

</manifest>
```

r. Nakon pokretanja mape na ekranu se prikazuje mapa sa *Toolbar*-om.

3. Prikaz korisnikove lokacije na mapi.
    a. Implementirana mapa prikazuje samo fiksnu lokaciju koja je prosleđena u implementiranoj metodi.
    b. U cilju dobijanja lokacije u toku testiranja aplikacije, potrebno je da koristiti nekakav izvor trenutne lokacije koji ne može biti GPS, jer se prilikom testiranja obično korisnik nalazi unutar objekta. U tom slučaju je, ukoliko se koristi emulator (neophodno je kreirati emulator koji kao target koristi GoogleAPI), potrebno lokaciju postaviti u Extended Controls aplikaciji. Aplikacija je potrebno pokrenuti iz menija emulatora.

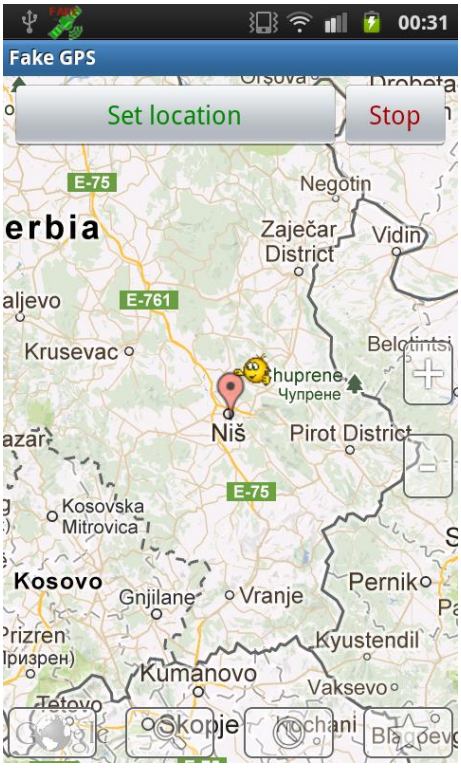


c. Nakon što je aplikacija pokrenuta, moguće je uneti željenu lokaciju. Potreno je u aplikaciji postaviti lokaciju upisujući longitudu i latitudu u odgovarajuća polja i kliknuti na *Send*. Ove koordinate će automatski biti dostupne u emulatoru.



d. Ukoliko je za testiranje potrebno koristiti uređaj, lokaciju je moguće dobiti ili od nekog drugog provajdera (poput WiFi ili GSM) ili koristiti aplikaciju koja emulira lokacijski provajder (preporuka *FakeGPS* aplikacija).

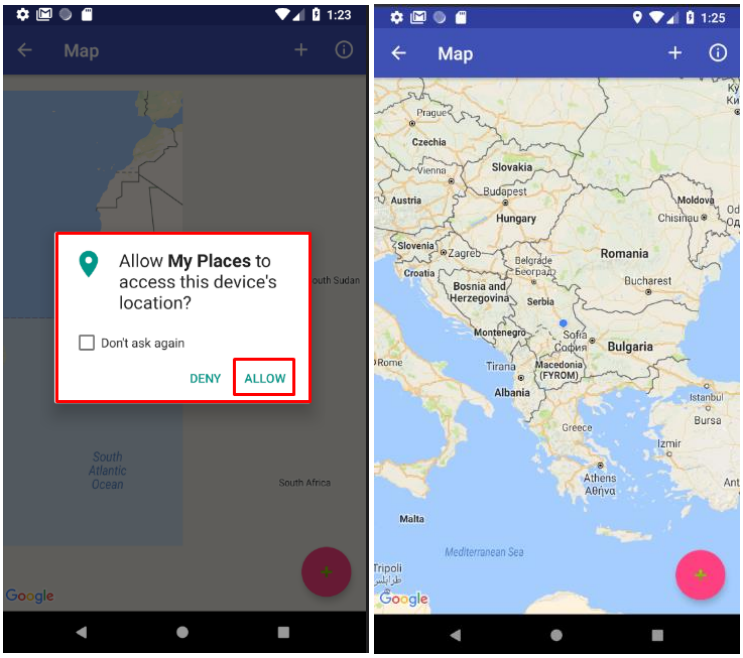
e. Korišćenjem *FakeGPS* aplikacije lokaciju je moguće postaviti izborom lokacije na mapi.



f. Prepraviti kod aplikacije tako da postavlja na mapi trenutnu korisnikovu lokaciju. Da bi bilo moguće postaviti tu lokaciju, neophodno je od korisnika zahtevati dopuštanje privilegije pribavljanja lokacije. *Permission* je već postavljen u AndroidManifest fajl, potrebno je da korisnik da saglasnost. Kod koji ovo omogućava je:

```
     * installed Google Play services and returned to the app.
     */
    static final int PERMISSION_ACCESS_FINE_LOCATION = 1;
    @Override
    public void onMapReady(GoogleMap googleMap) {
        mMap = googleMap;
        if (ActivityCompat.checkSelfPermission( context: this, Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED
                && ActivityCompat.checkSelfPermission( context: this, Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions( activity: this, new String[]{Manifest.permission.ACCESS_FINE_LOCATION}, PERMISSION_ACCESS_FINE_LOCATION);
        } else {
            mMap.setMyLocationEnabled(true);// Permission has already been granted
        }
    }

    @SuppressLint("MissingPermission")
    @Override
    public void onRequestPermissionsResult(int requestCode, String permissions[], int[] grantResults) {
        switch (requestCode) {
            case PERMISSION_ACCESS_FINE_LOCATION: {
                // If request is cancelled, the result arrays are empty.
                if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                    mMap.setMyLocationEnabled(true);// Permission has already been granted
                }
                return;
            }
        }
    }
}
```

g. Isprobati aplikaciju. Nakon pokretanja aplikacije, potrebno je odobriti aplikaciji korišćenje pristupa finoj lokaciji.



4. Proširenje klase *MyPlace.java*
    a. Dodati u klasu string polja longitude i latitude.
    b. Kreirati odgovarajuće *get* i *set* metode za dodata polja.

```
public class MyPlace {
    String name;
    String description;
    String longitude;
    String latitude;

    public String getLongitude() {
        return longitude;
    }

    public String getLatitude() {
        return latitude;
    }

    public void setLongitude(String longitude) {
        this.longitude = longitude;
    }

    public void setLatitude(String latitude) {
        this.latitude = latitude;
    }

    public MyPlace(String nme, String desc)
    {
        this.name = nme;
        this.description = desc;
    }
```

    c. Dodati u klasu *MyPlace.java long ID* polje koje će držati odgovarajući *ID* lokacije. Dodati i odgovarajuće *get* i *set* metode.

```
int ID;
public int getID() {
    return ID;
}

public void setID(int ID) {
    this.ID = ID;
}
```

    d. Potrebno je izmeniti izgled *EditMyPlaceActivity*-a tako da je moguće uneti longitudu i latitudu. Dodati i dugme koje će biti korišćeno za očitavanje koordinata sa mape.
        i. Dodati odgovarajuće string resurse

```
<string name="editmyplace_name_label">Name:</string>
<string name="editmyplace_desc_label">Description:</string>
<string name="editmyplace_cancel_label">Cancel</string>
<string name="editmyplace_lat_label">Latitude:</string>
<string name="editmyplace_lon_label">Longitude:</string>
<string name="editmyplace_location_label">Get Location</string>
<string name="title_activity_view_my_places">ViewMyPlacesActivity</string>
<string name="viewmyplace_finished_button">OK</string>
```

ii. Izmeniti *layout*

```
<TableLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:stretchColumns="*">
    <TableRow
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical">
            <TextView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="@string/editmyplace_lat_label"/>
            <EditText
                android:id="@+id/editmyplace_lat_edit"
                android:layout_width="match_parent"
                android:layout_height="wrap_content" />
            <TextView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="@string/editmyplace_lon_label"/>
            <EditText
                android:id="@+id/editmyplace_lon_edit"
                android:layout_width="match_parent"
                android:layout_height="wrap_content" />
        </LinearLayout>
        <Button
            android:id="@+id/editmyplace_location_button"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:text="@string/editmyplace_location_label"/>
    </TableRow>
    <TableRow
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <Button
            android:id="@+id/editmyplace_finished_button"
            android:layout_weight="1"/>
        <Button
            android:id="@+id/editmyplace_cancel_button"
            android:layout_weight="1"
            android:text="Cancel"/>
    </TableRow>
</TableLayout>
</LinearLayout>
```

e. Potrebno je povezati *EditMyPlaceActivity* tako da se vrši ažuriranje podataka o lokaciji.

```
@Override
public void onClick(View view) {
    switch(view.getId()) {
        case R.id.editmyplace_finished_button: {
            EditText etName = (EditText)findViewById(R.id.editmyplace_name_edit);
            String nme = etName.getText().toString();
            EditText etDesc = (EditText)findViewById(R.id.editmyplace_desc_edit);
            String desc = etDesc.getText().toString();
            EditText latEdit = (EditText)findViewById(R.id.editmyplace_lat_edit);
            String lat = latEdit.getText().toString();
            EditText lonEdit = (EditText)findViewById(R.id.editmyplace_lon_edit);
            String lon = lonEdit.getText().toString();
            if(!editMode)
            {
                MyPlace place = new MyPlace(nme, desc);
                place.setLatitude(lat);
                place.setLongitude(lon);
                MyPlacesData.getInstance().addNewPlace(place);
            }
            else
            {
                MyPlace place = MyPlacesData.getInstance().getPlace(position);
                place.setName(nme);
                place.setDesc(desc);
                place.setLatitude(lat);
                place.setLongitude(lon);
            }
            setResult(Activity.RESULT_OK);
            finish();
            break;
        }

        case R.id.editmyplace_cancel_button: {
            setResult(Activity.RESULT_CANCELED);
            finish();
            break;
        }
    }
}
```

f. Dodati mogućnost brisanja omiljene lokacije u kontekstni meni *MyPlacesList.java* klase. Potrebno je dodati novu stavku menija u metodi *onCreateContextMenu()*. Obraditi klik na dodatu stavku tako da se poziva metod *deletePlace()* iz klase *MyPlacesData.java*.

```
    myPlacesList.setOnCreateContextMenuListener((contextMenu, view, contextMenuInfo) -> {
            AdapterView.AdapterContextMenuInfo info = (AdapterView.AdapterContextMenuInfo) contextMenuInfo;
            MyPlace place = MyPlacesData.getInstance().getPlace(info.position);
            contextMenu.setHeaderTitle(place.getName());
            contextMenu.add( i: 0, i1: 1, i2: 1, charSequence: "View place");
            contextMenu.add( i: 0, i1: 2, i2: 2, charSequence: "Edit place");
            contextMenu.add( i: 0, i1: 3, i2: 3, charSequence: "Delete place");
    });
}

@Override
public boolean onContextItemSelected(MenuItem item){
    AdapterView.AdapterContextMenuInfo info = (AdapterView.AdapterContextMenuInfo)item.getMenuInfo();
    Bundle positionBundle = new Bundle();
    positionBundle.putInt("position", info.position);
    Intent i = null;
    if(item.getItemId() == 1) {
        i = new Intent( packageContext: this, ViewMyPlaceActivity.class);
        i.putExtras(positionBundle);
        startActivity(i);
    }
    else if(item.getItemId() == 2) {
        i = new Intent( packageContext: this, EditMyPlaceActivity.class);
        i.putExtras(positionBundle);
        startActivityForResult(i, requestCode: 1);
    }
    else if(item.getItemId() == 3){
        MyPlacesData.getInstance().deletePlace(info.position);
        setList();
    }
    return super.onContextItemSelected(item);
}

private void setList(){
    ListView myPlacesList = (ListView)findViewById(R.id.my_places_list);
    myPlacesList.setAdapter(new ArrayAdapter<MyPlace>( context: this, android.R.layout.simple_list_item_1, MyPlacesData.getInstance().getMyPlaces()));
}
```

5. Za domaći: dodati prikaz koordinata u *ViewMyPlaceActivity.java*.

6. Dodavanje podrške za izbor koordinata omiljene lokacije sa mape u *EditMyPlaceActivity.java*.
   a. Dodati kod koji dugmetu *Get Location* daje funkcionalnost izbora odgovarajuće koordinate na mapi.

```
        }
        finishedButton.setOnClickListener(this);
        finishedButton.setEnabled(false);
        cancelButton.setOnClickListener(this);
        nameEditText.addTextChangedListener(new TextWatcher() {
            @Override
            public void beforeTextChanged(CharSequence charSequence, int i, int i1, int i2) {

            }

            @Override
            public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) {

            }

            @Override
            public void afterTextChanged(Editable editable) {
                finishedButton.setEnabled(editable.length()>0);
            }
        });
        Button locationButton = (Button)findViewById(R.id.editmyplace_location_button);
        locationButton.setOnClickListener(this);
    }

    @Override
```

   b. Potrebno je dodati u metodu *onClick()*obradu klika na dugme *Get Location*.

```
            finish();
            break;
        }

        case R.id.editmyplace_cancel_button: {
            setResult(Activity.RESULT_CANCELED);
            finish();
            break;
        }
        case R.id.editmyplace_location_button:
        {
            Intent i = new Intent( packageContext: this, MyPlacesMapsActivity.class);
            startActivityForResult(i, requestCode: 1);
        }
    }
}

@Override
```

   c. Kreirati metodu *onActivityResult()* i u njoj imeplementirati preuzimanje koordinata lokacije koje će biti postavljene nakon završetka poziva *MyPlacesMapActivity*-a.

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data)
{
    super.onActivityResult(requestCode, resultCode, data);
    try
    {
        if (resultCode == Activity.RESULT_OK)
        {
            String lon = data.getExtras().getString( key: "lon");
            EditText lonText = (EditText)findViewById(R.id.editmyplace_lon_edit);
            lonText.setText(lon);
            String lat = data.getExtras().getString( key: "lat");
            EditText latText = (EditText)findViewById(R.id.editmyplace_lat_edit);
            latText.setText(lat);
        }
    }
    catch (Exception e) {
        // TODO: handle exception
    }
}
```

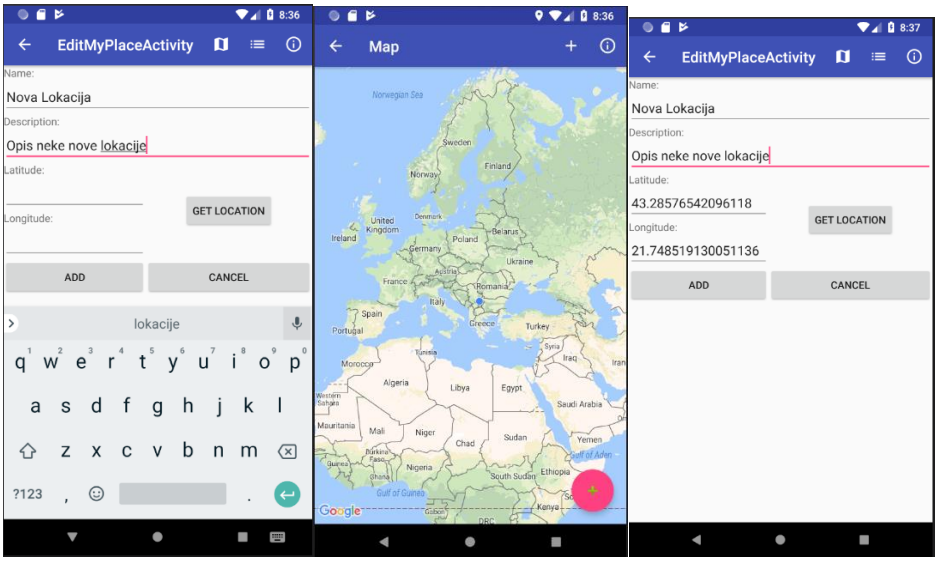
d. U *MyPlacesMapActivity.java* klasu je potrebno dodati mogućnost klika na mapu i izbora lokacije.

```
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;
    if (ActivityCompat.checkSelfPermission( context: this, Manifest.permission.ACCESS_FINE_LOCATION) != PackageMa
            && ActivityCompat.checkSelfPermission( context: this, Manifest.permission.ACCESS_COARSE_LOCATION) !=
        ActivityCompat.requestPermissions( activity: this, new String[]{Manifest.permission.ACCESS_FINE_LOCATION},
    } else {
        mMap.setMyLocationEnabled(true);// Permission has already been granted
        setOnMapClickListener();
    }
}
```

```
private void setOnMapClickListener(){
    if(mMap!=null){
        mMap.setOnMapClickListener(new GoogleMap.OnMapClickListener() {
            @Override
            public void onMapClick(LatLng latLng) {
                String lon=Double.toString(latLng.longitude);
                String lat=Double.toString(latLng.latitude);
                Intent locationIntent = new Intent();
                locationIntent.putExtra( name: "lon", lon);
                locationIntent.putExtra( name: "lat", lat);
                setResult(Activity.RESULT_OK, locationIntent);
                finish();
            }
        });
    }
}
```

```
@SuppressLint("MissingPermission")
@Override
public void onRequestPermissionsResult(int requestCode, String permissions[], int[] grantResults) {
    switch (requestCode) {
        case PERMISSION_ACCESS_FINE_LOCATION: {
            // If request is cancelled, the result arrays are empty.
            if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                mMap.setMyLocationEnabled(true);// Permission has already been granted
                setOnMapClickListener();
            }
            return;
        }
    }
}
```

e. Ovim je omogućneo preuzimanje lokacije na koju je kliknuto na mapi. Potrebno je probati aplikaciju.



7. Prilagođavanje *MyPlacesMapActivity*-a za različita stanja rada (prikaz lokacija na mapi, centriranje prikaza omiljene lokacije na mapi, izbor koordinata omiljene lokacije na mapi).
    a. Trenutno je moguće izabrati koordinate sa mape ali nije moguće pomerati mapu (*pan*) i uvećavati/umanjivati mapu (*zoom*). Potrebno je implementirati u *MyPlacesMapActivity.java* klasi različita stanja, prosleđena putem *Intent*-a tako da se rad sa mapom prilagodi odgovarajućoj nameni u aplikaciji. Prvo je potrebno napraviti stanja i promenljivu koja će ih čuvati. Dobra je praksa napraviti konstante kako ne bi došlo do greške prilikom poziva.

```
public class MyPlacesMapsActivity extends AppCompatActivity implements OnMapReadyCallback {

    private GoogleMap mMap;
    static int NEW_PLACE = 1;
    public static final int SHOW_MAP = 0;
    public static final int CENTER_PLACE_ON_MAP = 1;
    public static final int SELECT_COORDINATES = 1;

    private int state = 0;
    private boolean selCoorsEnabled = false;
    private LatLng placeLoc;
```

    b. Potrebno je prepraviti *onCreate* metodu tako da očitava stanje *Activity*-a prosleđeno preko *Intent*-a i postavlja odgovarajuće promenljive. U zavisnosti od postavljenog stanja, potrebno je i pozvati odgovarajuće metode. Ukoliko je stanje SHOW_MAP omogućiti prikaz trenutne korisnikove lokacije. Za stanje SELECT_COORDINATES omogućiti *Click* na mapu a za stanje CENTER_PLACE_ON_MAP postaviti kameru zumirati mapu na koordinate omiljenog mesta prosleđene preko *Intent*-a.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    try{
        Intent mapIntent = getIntent();
        Bundle mapBundle = mapIntent.getExtras();
        if(mapBundle != null){
            state = mapBundle.getInt( key: "state");
            if(state == CENTER_PLACE_ON_MAP){
                String placeLat = mapBundle.getString( key: "lat");
                String placeLon = mapBundle.getString( key: "lon");
                placeLoc = new LatLng(Double.parseDouble(placeLat), Double.parseDouble(placeLon));
            }
        }
    } catch(Exception e) {
        Log.d( tag: "Error",  msg: "Error reading state");
    }
    setContentView(R.layout.activity_my_places_maps);
    Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);
    getSupportActionBar().setDisplayHomeAsUpEnabled(true);
    getSupportActionBar().setDisplayShowHomeEnabled(true);


    FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
    if(state != SELECT_COORDINATES){
        fab.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
                Intent i = new Intent( packageContext: MyPlacesMapsActivity.this, EditMyPlaceActivity.class);
                startActivityForResult(i, NEW_PLACE);
            }
        });
    } else {
        ViewGroup layout = (ViewGroup) fab.getParent();
        if(null!=layout) //for safety only  as you are doing onClick
            layout.removeView(fab);
    }
    // Obtain the SupportMapFragment and get notified when the map is ready to be used.
    SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()
            .findFragmentById(R.id.map);
    mapFragment.getMapAsync( onMapReadyCallback: this);
}
```

```
@Override
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;
    if (ActivityCompat.checkSelfPermission( context: this, Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED
            && ActivityCompat.checkSelfPermission( context: this, Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
        ActivityCompat.requestPermissions( activity: this, new String[]{Manifest.permission.ACCESS_FINE_LOCATION}, PERMISSION_ACCESS_FINE_LOCATION);
    } else {
        if(state == SHOW_MAP)
            mMap.setMyLocationEnabled(true);// Permission has already been granted
        else if(state == CENTER_PLACE_ON_MAP)
            setOnMapClickListener();
        else
            mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(placeLoc,  v: 15));
    }
}

private void setOnMapClickListener(){
    if(mMap!=null){
        mMap.setOnMapClickListener(new GoogleMap.OnMapClickListener() {
            @Override
            public void onMapClick(LatLng latLng) {
                String lon=Double.toString(latLng.longitude);
                String lat=Double.toString(latLng.latitude);
                Intent locationIntent = new Intent();
                locationIntent.putExtra( name: "lon", lon);
                locationIntent.putExtra( name: "lat", lat);
                setResult(Activity.RESULT_OK, locationIntent);
                finish();
            }
        });
    }
}

@SuppressLint("MissingPermission")
@Override
public void onRequestPermissionsResult(int requestCode, String permissions[], int[] grantResults) {
    switch (requestCode) {
        case PERMISSION_ACCESS_FINE_LOCATION: {
            // If request is cancelled, the result arrays are empty.
            if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                if(state == SHOW_MAP)
                    mMap.setMyLocationEnabled(true);// Permission has already been granted
                else if(state == CENTER_PLACE_ON_MAP)
                    setOnMapClickListener();
                else
                    mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(placeLoc,  v: 15));
            }
            return;
        }
    }
}
```

c. U slučaju korišćenja mape kada se vrši izbor lokacije, korisniku će inicijalno biti omogućeno da radi sa njom. Nakon što u options meniju izabere odgovarajuće dugme

(*Select Coordinates*), sledeći klik (*tap*) na mapu će izvršiti izbor koordinata. Izmeniti metode *onCreateOptionsMenu()* i *onOptionsItemSelected()* tako da je moguće aktivirati izbor koordinate.

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    if(state == SELECT_COORDINATES && !selCoorsEnabled){
        menu.add( i: 0, i1: 1, i2: 1,  charSequence: "Select Coordinates");
        menu.add( i: 0, i1: 2, i2: 2,  charSequence: "Cancel");
        return super.onCreateOptionsMenu(menu);
    } else {
        getMenuInflater().inflate(R.menu.menu_my_places_maps, menu);
        return true;
    }
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();

    //noinspection SimplifiableIfStatement
    if(state == SELECT_COORDINATES && !selCoorsEnabled){
        if(id == 1){
            selCoorsEnabled = true;
            Toast.makeText( context: this,  text: "Select coordinates", Toast.LENGTH_SHORT).show();
        } else if(id == 2){
            setResult(Activity.RESULT_CANCELED);
            finish();
        }
    } else {
        if (id == R.id.new_place_item) {
            Intent i = new Intent( packageContext: this, EditMyPlaceActivity.class);
            startActivityForResult(i,  requestCode: 1);
        } else if (id == R.id.about_item) {
            Intent i = new Intent( packageContext: this, About.class);
            startActivity(i);
        } else if (id == android.R.id.home) {
            finish();
        }
    }
    return super.onOptionsItemSelected(item);
}
```

d. Potrebno je izvršiti izmene i u metodi *onMapClick()* tako da se preuzima lokacija samo onda kada smo spremni da izaberemo odgovarajuće koordinate.

```
private void setOnMapClickListener(){
    if(mMap!=null){
        mMap.setOnMapClickListener(new GoogleMap.OnMapClickListener() {
            @Override
            public void onMapClick(LatLng latLng) {
                if(state == SELECT_COORDINATES && selCoorsEnabled){
                    String lon=Double.toString(latLng.longitude);
                    String lat=Double.toString(latLng.latitude);
                    Intent locationIntent = new Intent();
                    locationIntent.putExtra( name: "lon", lon);
                    locationIntent.putExtra( name: "lat", lat);
                    setResult(Activity.RESULT_OK, locationIntent);
                    finish();
                }
            }
        });
    }
}
```

e. Potrebno je dodati odgovarajuće stanja u *Intent*-e poziva koji su ranije kreirani za pozive *MyPlacesMapActivity*-a

```
                place.setDesc(desc);
                place.setLatitude(lat);
                place.setLongitude(lon);
            }
            setResult(Activity.RESULT_OK);
            finish();
            break;
        }

        case R.id.editmyplace_cancel_button: {
            setResult(Activity.RESULT_CANCELED);
            finish();
            break;
        }
        case R.id.editmyplace_location_button: {
            Intent i = new Intent( packageContext: this, MyPlacesMapsActivity.class);
            i.putExtra( name: "state", MyPlacesMapsActivity.SELECT_COORDINATES);
            startActivityForResult(i, requestCode: 1);
        }
    }
}
```

```
static int NEW_PLACE = 1;
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();

    if (id == R.id.show_map_item) {
        Intent i = new Intent(this, MyPlacesMapActivity.class);
        i.putExtra("state", MyPlacesMapActivity.SHOW_MAP);
        startActivity(i);
    } else if(id == R.id.new_place_item){
        Intent i = new Intent(this, EditMyPlaceActivity.class);
        startActivityForResult(i, NEW_PLACE);
    } else if(id == R.id.my_places_list_item){
        Intent i = new Intent(this, MyPlacesList.class);
        startActivity(i);
    } else if(id == R.id.about_item){
        Intent i = new Intent(this, About.class);
        startActivity(i);
    }

    return super.onOptionsItemSelected(item);
}
```

f. Probati aplikaciju.

g. Ukoliko je mapa pozvana klikom na dugme *Get Location*, u gornjem desnom uglu su dodate stavke Action Bar-a.

Domaći: dodati odgovarajuću ikonu i onemogućiti ponovni klik na Set Coordinates item.

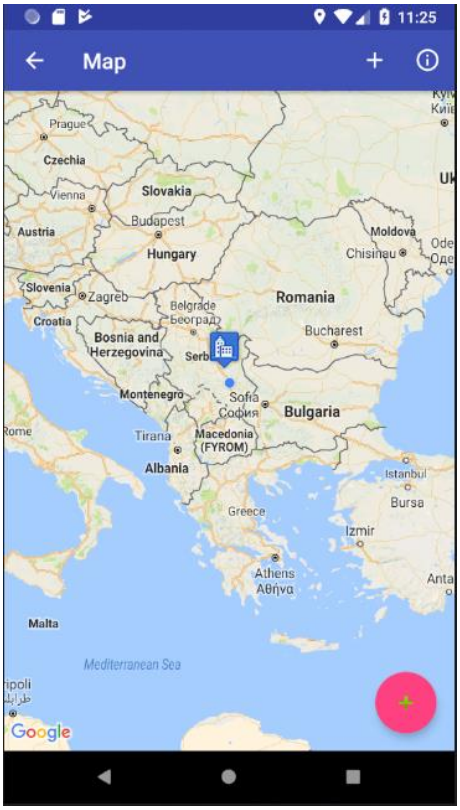
8. Prikaz markera omiljenih lokacija na mapi.

a. Pošto je moguće pribaviti koordinate omiljenih lokacija, moguće ih je lako prikazati na mapi korišćenjem markera.
b. Prvo je potrebno dodati ikonu koja će biti prikazivana kao marker. Smestiti odgovarajuće ikone u drawable foldere, na način na koji je to bilo rađeno na prethodnim vežbama (dostupne na sajtu predmeta).
c. U *onCreate()* metodi *MyPlacesMapActivity* klase dodati kod koji omogućuje dodavanje markera za svako omiljeno mesto koje se nalazi u bazi podataka i mapira objekte dodatih markera sa njihovim pozicijama u listi u *HashMap* objekat tako da konačno metod izgleda kao što je prikazano na slici.

```
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;
    if (ActivityCompat.checkSelfPermission( context: this, Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED
            && ActivityCompat.checkSelfPermission( context: this, Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
        ActivityCompat.requestPermissions( activity: this, new String[]{Manifest.permission.ACCESS_FINE_LOCATION}, PERMISSION_ACCESS_FINE_LOCATION);
    } else {
        if(state == SHOW_MAP)
            mMap.setMyLocationEnabled(true);// Permission has already been granted
        else if(state == SELECT_COORDINATES)
            setOnMapClickListener();
        else
            mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(placeLoc,  v: 15));
        addMyPlaceMarkers();
    }
}

private HashMap<Marker, Integer> markerPlaceIdMap;

private void addMyPlaceMarkers(){
    ArrayList<MyPlace> places = MyPlacesData.getInstance().getMyPlaces();
    markerPlaceIdMap = new HashMap<Marker, Integer>((int)((double)places.size()*1.2));
    for(int i = 0;i<places.size();i++){
        MyPlace place = places.get(i);
        String lat = place.getLatitude();
        String lon = place.getLongitude();
        LatLng loc = new LatLng(Double.parseDouble(lat), Double.parseDouble(lon));
        MarkerOptions markerOptions = new MarkerOptions();
        markerOptions.position(loc);
        markerOptions.icon(BitmapDescriptorFactory.fromResource(R.drawable.myplace));
        markerOptions.title(place.getName());
        Marker marker = mMap.addMarker(markerOptions);
        markerPlaceIdMap.put(marker, i);
    }

    mMap.setOnMarkerClickListener(new GoogleMap.OnMarkerClickListener() {
        @Override
        public boolean onMarkerClick(Marker marker) {
            Intent intent = new Intent( packageContext: MyPlacesMapsActivity.this, ViewMyPlaceActivity.class );
            int i = markerPlaceIdMap.get(marker);
            intent.putExtra( name: "position", i);
            startActivity(intent);
            return true;
        }
    });
}
```

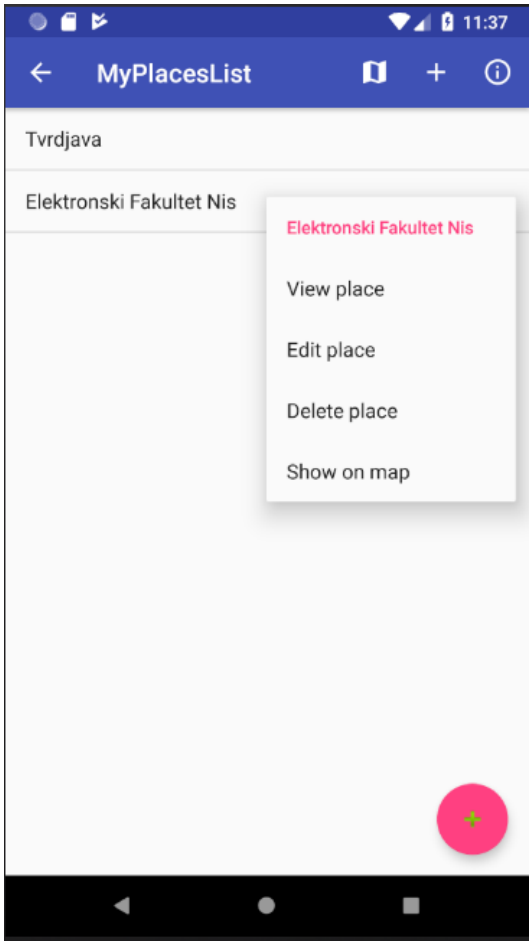
```
@SuppressLint("MissingPermission")
@Override
public void onRequestPermissionsResult(int requestCode, String permissions[], int[] grantResults) {
    switch (requestCode) {
        case PERMISSION_ACCESS_FINE_LOCATION: {
            // If request is cancelled, the result arrays are empty.
            if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                if(state == SHOW_MAP)
                    mMap.setMyLocationEnabled(true);// Permission has already been granted
                else if(state == SELECT_COORDINATES)
                    setOnMapClickListener();
                else
                    mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(placeLoc,  v: 15));
                addMyPlaceMarkers();
            }
            return;
        }
    }
}
```

d. Probati aplikaciju



9. Dodati mogućnost prikaza omiljene lokacije na mapi dodavanjem opcije u kontekstni meni *MyPlacesList.java* klase.
    a. Potrebno je dodati novu stavku menija u metodi *onCreateContextMenu()*
    b. Obraditi klik na dodatu stavku tako da se kreira intent za *MyPlaceMapActivity* i prosleđuje odgovarajuće stanje (*CENTER_PLACE_ON_MAP*).

```
    });
    myPlacesList.setOnCreateContextMenuListener((contextMenu, view, contextMenuInfo) → {
            AdapterView.AdapterContextMenuInfo info = (AdapterView.AdapterContextMenuInfo) contextMenuInfo;
            MyPlace place = MyPlacesData.getInstance().getPlace(info.position);
            contextMenu.setHeaderTitle(place.getName());
            contextMenu.add( i: 0, i1: 1, i2: 1, charSequence: "View place");
            contextMenu.add( i: 0, i1: 2, i2: 2, charSequence: "Edit place");
            contextMenu.add( i: 0, i1: 3, i2: 3, charSequence: "Delete place");
            contextMenu.add( i: 0, i1: 4, i2: 4, charSequence: "Show on map");
    });
}

@Override
public boolean onContextItemSelected(MenuItem item){
    AdapterView.AdapterContextMenuInfo info = (AdapterView.AdapterContextMenuInfo)item.getMenuInfo();
    Bundle positionBundle = new Bundle();
    positionBundle.putInt("position", info.position);
    Intent i = null;
    if(item.getItemId() == 1) {
        i = new Intent( packageContext: this, ViewMyPlaceActivity.class);
        i.putExtras(positionBundle);
        startActivity(i);
    }
    else if(item.getItemId() == 2) {
        i = new Intent( packageContext: this, EditMyPlaceActivity.class);
        i.putExtras(positionBundle);
        startActivityForResult(i,  requestCode: 1);
    }
    else if(item.getItemId() == 3){
        MyPlacesData.getInstance().deletePlace(info.position);
        setList();
    }
    else if(item.getItemId() == 4){
        i = new Intent( packageContext: this, MyPlacesMapsActivity.class);
        i.putExtra( name: "state", MyPlacesMapsActivity.CENTER_PLACE_ON_MAP);
        MyPlace place = MyPlacesData.getInstance().getPlace(info.position);
        i.putExtra( name: "lat", place.getLatitude());
        i.putExtra( name: "lon", place.getLongitude());
        startActivityForResult(i, requestCode: 2);
    }
    return super.onContextItemSelected(item);
}
```



10. Domaći: Dodati prikaz koordinata u *ViewMyPlaceActivity.java*