

MOSIS - Laboratorijska vežba 4

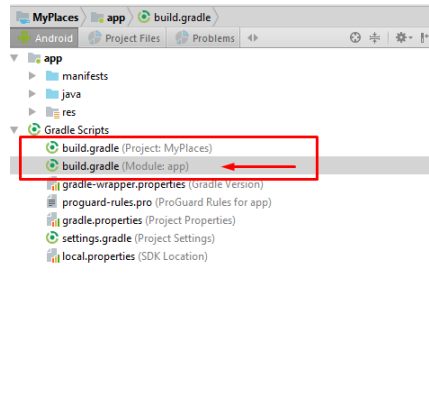
Na poslednjoj laboratorijskoj vežbi je omogućen unos omiljenih lokacija i njihovo pregledavanje. Najznačajniji prikaz omiljenih lokacija je onaj koji vezan za kontekst njihove stvarne lokacije na mapi. Za prikaz objekata na mapi, *Google* nudi programerima svoje rešenje za rad sa mapama u vidu posebnog *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. U poslednje vreme, Google je promenio poslovnu politiku vezanu za korišćenje mapa u svojim aplikacijama i trenutno zahteva unos podataka o kreditnoj kartici kako bi bilo moguće koristiti API key za Google Maps. Zbog toga je neophodno potražiti alternativne načine prikaza mapa u Android projektima.

Jedna od mogućnosti je korišćenje Open Street Maps mapa. Open Street Maps su besplatne mape koje kreiraju volonteri. Njihovo korišćenje je slobodno, ali usled izostanka komercijalnih uslova kojima je podržan hosting mapa, potrebno je koristiti različite izvore dobijenih od strane Open Street Maps. U ovoj vežbi će akcenat biti na radu sa OSMDroid mapama, kao besplatnim modulom koji omogućuje korišćenje mapa u projektu, 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. Uključivanje osmdroid biblioteke u Android projekat.
 2. Priprema AndroidManifest.xml-a i dodavanje preferenci.
 3. Uključivanje osmdroid MapView komponente.
 4. Kreiranje *MyPlacesMapActivity.java* klase i prikaz mape sa trenutnom lokacijom korisnika.
 5. Dodavanje kontrola za unos i prikaz koordinata omiljene lokacije u *EditMyPlaceActivity.java* i podrška za izbor koordinata omiljene lokacije sa mape.
 6. Prilagođavanje *MyPlacesMapActivity.java* za različita stanja rada (prikaz lokacija na mapi, centriranje prikaza omiljene lokacije na mapi, izbor koordinata omiljene lokacije na mapi).
 7. Dodavanje markera omiljenih lokacija na mapi.
 8. Prikaz određene omiljene lokacije centrirano na mapi.
-
1. Uključivanje osmdroid biblioteke u projekat.
 - a. Implementacija biblioteke osmdroid se nalazi na adresi:
<https://github.com/osmdroid/osmdroid> dok je tutorijal za korišćenje dat na adresi
<https://github.com/osmdroid/osmdroid/wiki/How-to-use-the-osmdroid-library>.
 - b. Biblioteku je potrebno 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.

Android

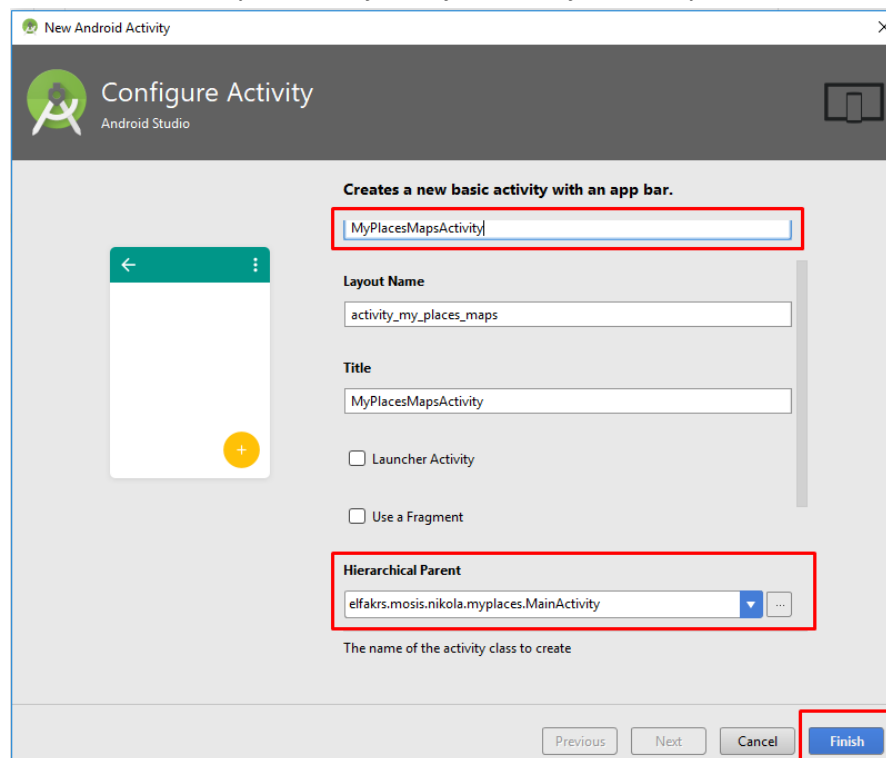


- c. Otvoriti *build.gradle* fajl koji odgovara modulu aplikacije (Module:app) i dodati: *implementation 'org.osmdroid:osmdroid-android:6.1.0'*.

```
dependencies {  
    implementation fileTree(dir: 'libs', include: ['*.jar'])  
    implementation 'com.android.support:appcompat-v7:26.1.0'  
    implementation 'com.android.support.constraint:constraint-layout:1.0.2'  
    implementation 'com.android.support:design:26.1.0'  
    implementation 'org.osmdroid:osmdroid-android:6.1.0'  
    testImplementation 'junit:junit:4.12'  
    androidTestImplementation 'com.android.support.test:runner:1.0.1'  
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.1'  
}
```

2. Kreiranje MyPlacesMapsActivity-a.

- a. Potrebno je pokrenuti dodavanje novog Activity-a (*Basic Activity*) u projekat.
b. Podesiti ime *Activity*-a kao i hijerarhijski roditeljski *Activity*.



- c. Dodati potrebne dozvole za pristup aplikacije resursima telefona. Tipično je potrebno dodati dozvolu za pristup finoj lokaciji sa senzora za lokaciju kako bi mogla biti očitana trenutna lokacija, pristup internetu i očitavanju statusa mreže, što omogućuje preuzimanje mape preko online izvora i za pristup fajl sistemu kako bi mapa mogla biti učitana sa fajl sistema i kako bi delovi mape mogli biti keširani.
- d. Potrebno je pravilno postaviti layout na sledećin način:

```
<?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.MyPlacesMapsActivity">

    <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_my_places_maps" />

    <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>
```

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    app:layout_behavior="android.support.design.widget.AppBarLayout$ScrollingViewBehavior"
    tools:context="elfakrs.mosis.nikola.myplaces.MyPlacesMapsActivity"
    tools:showIn="@layout/activity_my_places_maps">

    <org.osmdroid.views.MapView android:id="@+id/map"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tilesource="Mapnik"/>

</LinearLayout>
```

- e. Dodati odgovarajući kod u MyPlacesMapsActivity.java koji inicijalizuje mapu.

```

public class MyPlacesMapsActivity extends AppCompatActivity {
    MapView map = null;
    @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);

        FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
        fab.setOnClickListener((view) -> {
            Snackbar.make(view, text: "Replace with your own action", Snackbar.LENGTH_LONG)
                .setAction(text: "Action", listener: null).show();
        });
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);

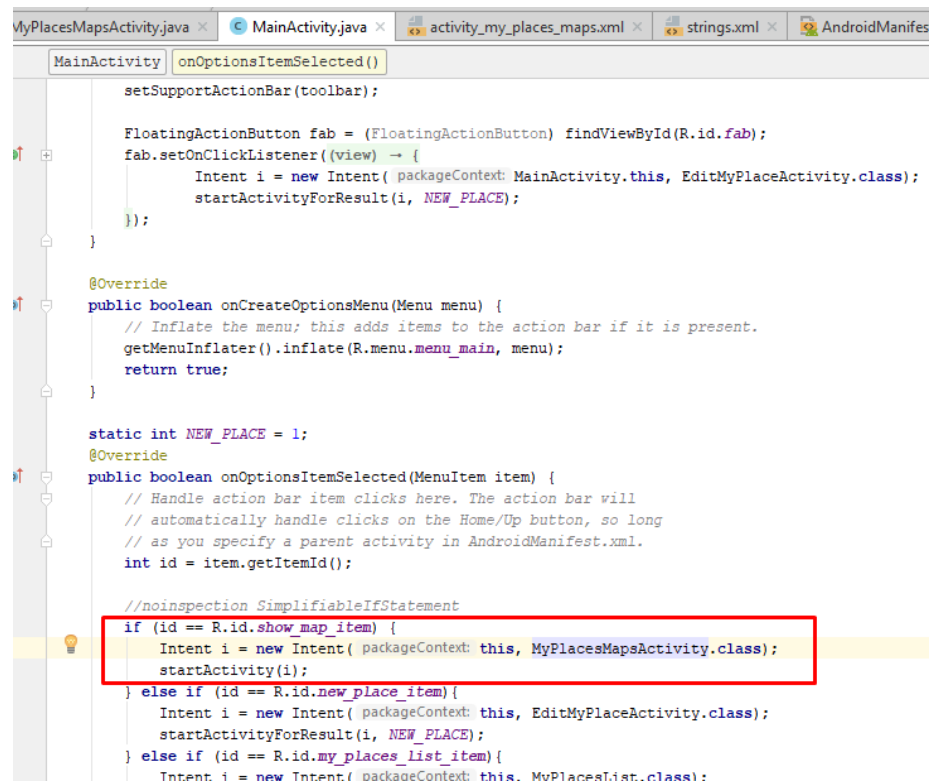
        Context ctx = getApplicationContext();
        Configuration.getInstance().load(ctx, PreferenceManager.getDefaultSharedPreferences(ctx));
        map = (MapView) findViewById(R.id.map);
    }

    public void onResume() {
        super.onResume();
        map.onResume();
    }

    public void onPause() {
        super.onPause();
        map.onPause();
    }
}

```

- f. U cilju testiranja dodatog Map Activity-a, potrebno je dodati odgovarajući poziv u *MainActivity-i*.



```

MainActivity.java x MainActivity.java x activity_my_places_maps.xml x strings.xml x AndroidManifest
MainActivity onOptionsItemSelected()

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);
});

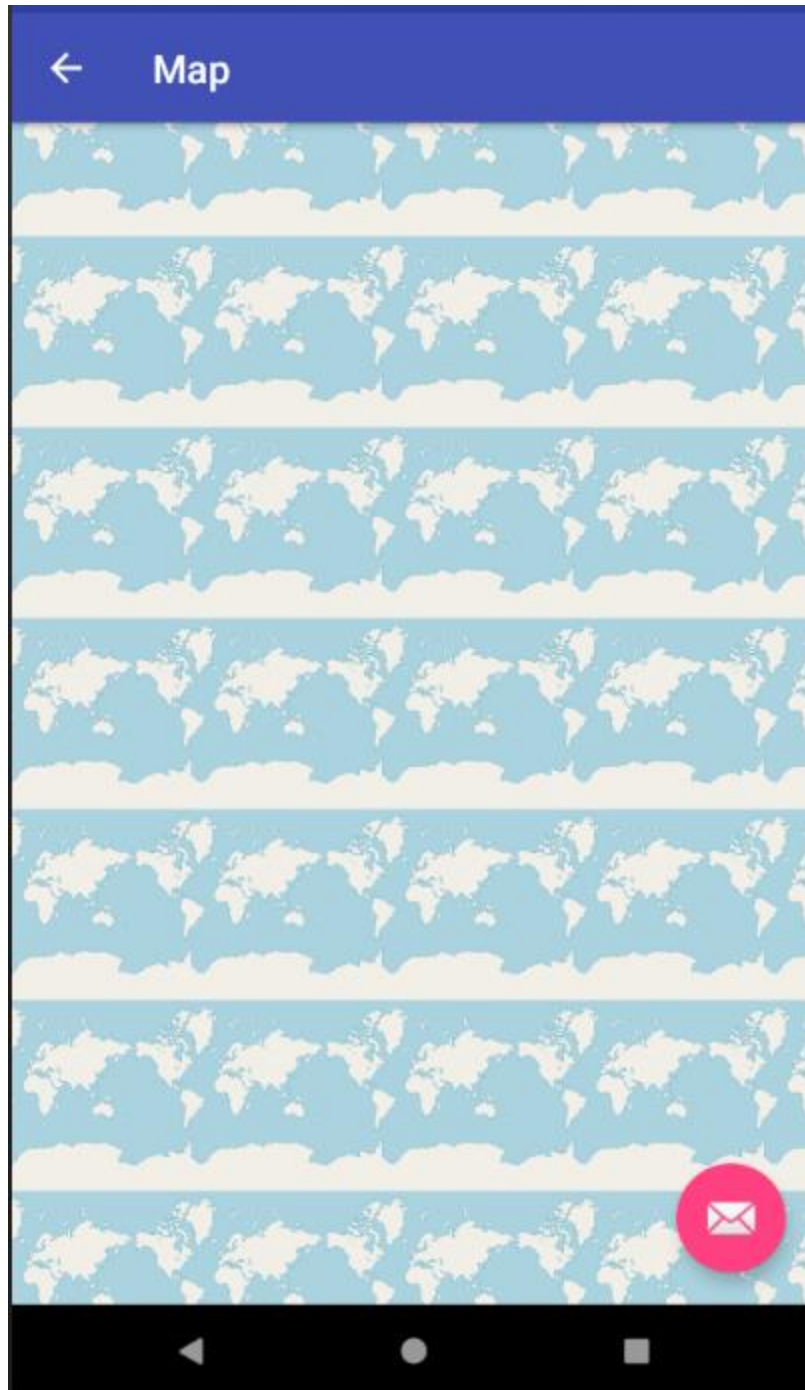
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.menu_main, menu);
    return true;
}

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);
    }
}

```

- g. Pokrenuti aplikaciju i otvoriti mapu



- h. Postaviti da *Action* dugme ukazuje na akciju dodavanja.

```
<include layout="@layout/content_my_places_maps" />

<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>
```

- i. Dodati kod koji poziva `EditMyPlaceActivity` za dodavanje lokacija.

```

public class MyPlacesMapsActivity extends AppCompatActivity {
    MapView map = null;
    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);

        FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
        fab.setOnClickListener((view) -> {
            Intent i = new Intent( packageContext: MyPlacesMapsActivity.this, EditMyPlaceActivity.class);
            startActivityForResult(i, NEW_PLACE);
        });
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);

        Context ctx = getApplicationContext();
        Configuration.getInstance().load(ctx, PreferenceManager.getDefaultSharedPreferences(ctx));
        map = (MapView) findViewById(R.id.map);
    }
}

```

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

```

<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>

```

- k. Dodati kod koji kontroliše rad menija u *MyPlacesMapsActivity* klasu.

```

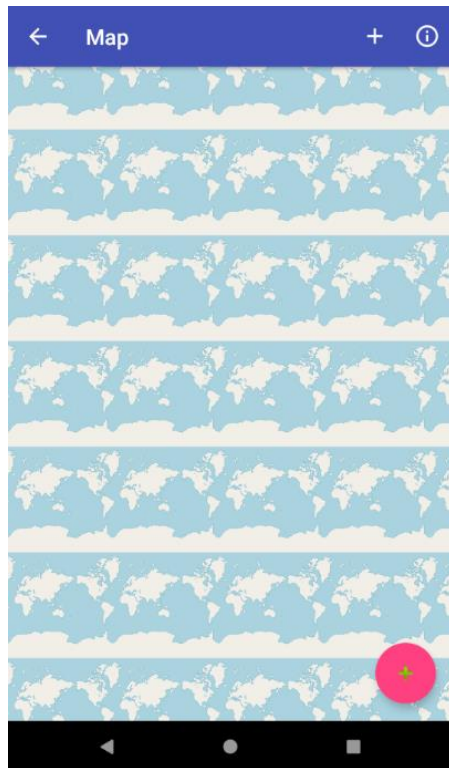
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.menu_my_places_maps, menu);
    return true;
}

@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);
}

```

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



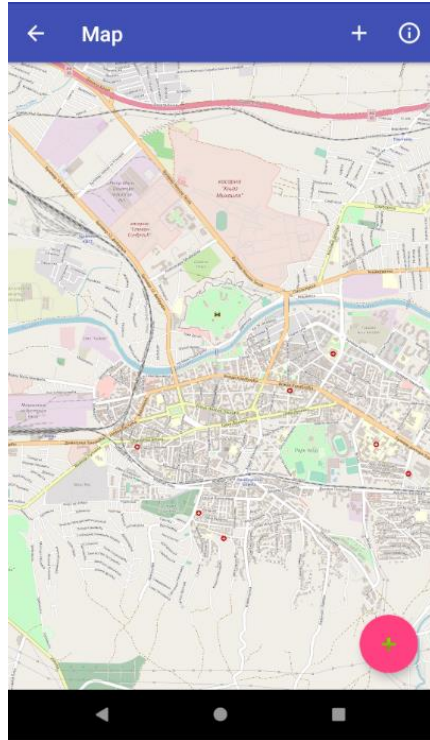
- m. Dodati kontrolu mape I postaviti je tako da ukazuje na grad Niš (koordinate latituda: 43.3209, longituda: 21.8958)

```
public class MyPlacesMapsActivity extends AppCompatActivity {
    MapView map = null;
    IMapController mapController = null;
    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);

        FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
        fab.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent i = new Intent(MyPlacesMapsActivity.this, EditMyPlaceActivity.class);
                startActivity(i);
            }
        });
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);

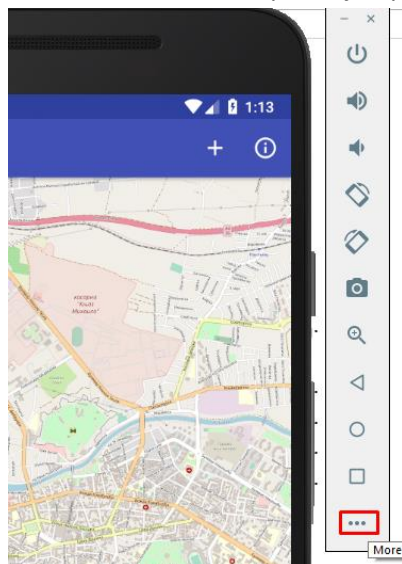
        Context ctx = getApplicationContext();
        Configuration.getInstance().load(ctx, PreferenceManager.getDefaultSharedPreferences(ctx));
        map = (MapView) findViewById(R.id.map);
        map.setMultiTouchControls(true);
        mapController = map.getController();
        if (mapController != null) {
            mapController.setZoom(15.0);
            GeoPoint startPoint = new GeoPoint(43.3209, 21.8958);
            mapController.setCenter(startPoint);
        }
    }
}
```

- n. Pokrenuti mapu ponovo!

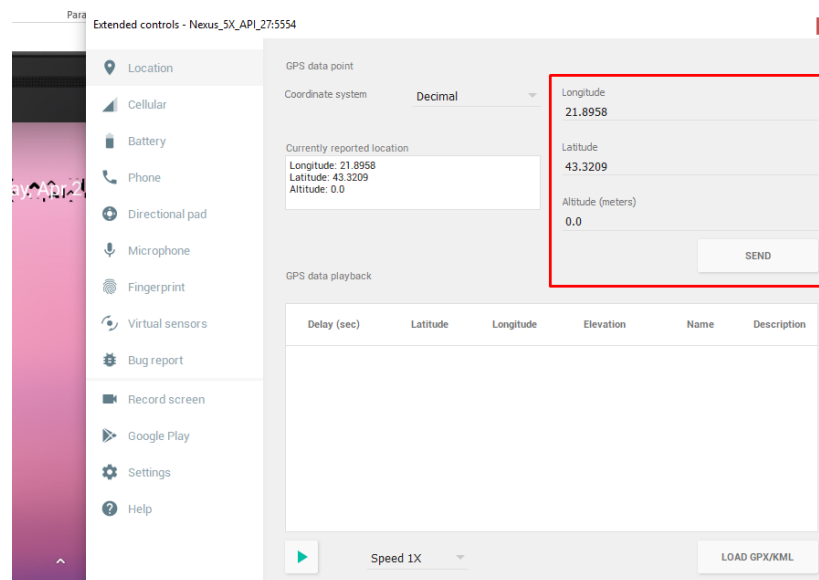


3. Prikaz lokacije korisnika na mapi.

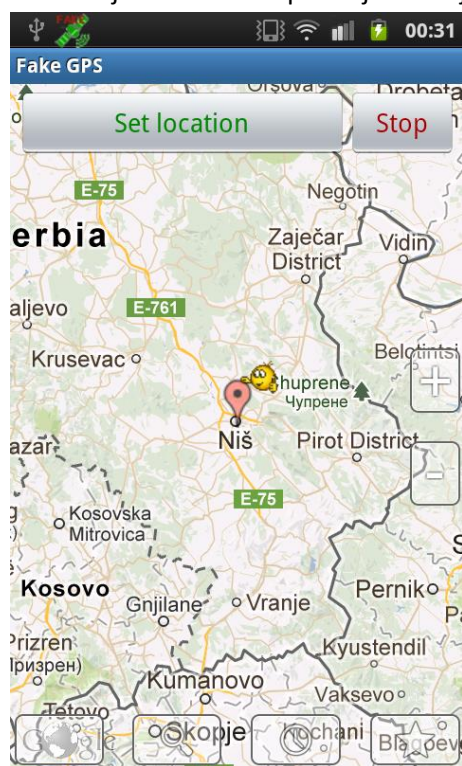
- a. Implementirana mapa prikazuje samo fiksnu lokaciju koja je prosleđena u implementiranoj metodi.
- b. U cilju dobijanja lokacije u toku testiranja aplikacije, potrebno je koristiti nekakav izvor trenutne lokacije koji ne može biti GPS, jer se prilikom testiranja obično korisnik nalazi unutar objekta. U tom slučaju je, ukoliko se koristi emulator, potrebno lokaciju postaviti u Extended Controls aplikaciji. Aplikaciju je potrebno pokrenuti iz menija emulatora.



- c. Nakon što je aplikacija pokrenuta, moguće je uneti željenu lokaciju. Potrebno 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 dodat u *AndroidManifest* fajl, ali je neophodno i da korisnik potvrdi saglasnost. Kod koji ovo omogućava je:

Android

```
static final int PERMISSION_ACCESS_FINE_LOCATION = 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);

    FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
    fab.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent i = new Intent(MyPlacesMapsActivity.this, EditMyPlaceActivity.class);
            startActivity(i);
        }
    });
    getSupportActionBar().setDisplayHomeAsUpEnabled(true);

    Context ctx = getApplicationContext();
    Configuration.getInstance().load(ctx, PreferenceManager.getDefaultSharedPreferences(ctx));
    map = (MapView) findViewById(R.id.map);
    map.setMultiTouchControls(true);

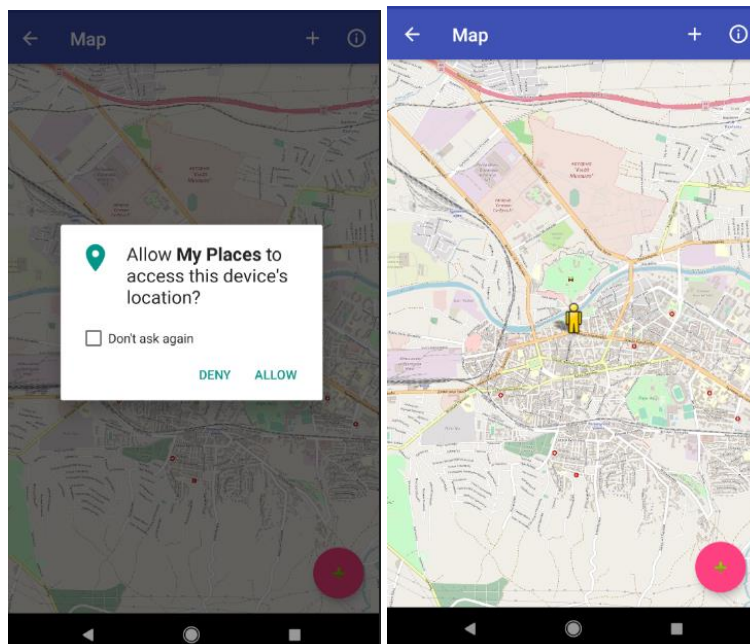
    if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED
        && ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
        ActivityCompat.requestPermissions(this, new String[]{Manifest.permission.ACCESS_FINE_LOCATION, PERMISSION_ACCESS_FINE_LOCATION}, 1);
    } else {
        setMyLocationOverlay();
    }

    mapController = map.getController();
    if (mapController != null) {
        mapController.setZoom(15.0);
        GeoPoint startPoint = new GeoPoint(43.3209, 21.8958);
        mapController.setCenter(startPoint);
    }
}

private void setMyLocationOverlay() {
    myLocationOverlay = new MyLocationNewOverlay(new GpsMyLocationProvider(this), map);
    myLocationOverlay.enableMyLocation();
    map.getOverlays().add(this.myLocationOverlay);
}

@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) {
                setMyLocationOverlay();
            }
            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>
<string name="title_activity_my_places_map">Map</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 dodatnu 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(0, 1, 1, CharSequence: "View place");
    contextMenu.add(0, 2, 2, CharSequence: "Edit place");
    contextMenu.add(0, 3, 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 il, int i2) {

        }

        @Override
        public void onTextChanged(CharSequence charSequence, int i, int il, 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.

```

private void setOnMapClickOverlay(){
    MapEventsReceiver mReceive = new MapEventsReceiver() {
        @Override
        public boolean singleTapConfirmedHelper(GeoPoint p) {
            String lon=Double.toString(p.getLongitude());
            String lat=Double.toString(p.getLatitude());
            Intent locationIntent = new Intent();
            locationIntent.putExtra( name: "lon", lon);
            locationIntent.putExtra( name: "lat", lat);
            setResult(Activity.RESULT_OK, locationIntent);
            finish();
            //ostavljeno zbog kompajliranja, nece se izvrstiti
            return false;
        }

        @Override
        public boolean longPressHelper(GeoPoint p) {
            return false;
        }
    };

    MapEventsOverlay OverlayEvents = new MapEventsOverlay(mReceive);
    map.getOverlays().add(OverlayEvents);
}

```

- e. Dodati odgovarajuće pozive prethodno definisane funkcije:

```

map.setMultiTouchControls(true);
if (ActivityCompat.checkSelfPermission( context: this, Manifest.perm
    && ActivityCompat.checkSelfPermission( context: this, Manife
        ActivityCompat.requestPermissions( activity: this, new String[] {M
    }
} else {
    setMyLocationOverlay();
    setOnMapClickOverlay();
}

```

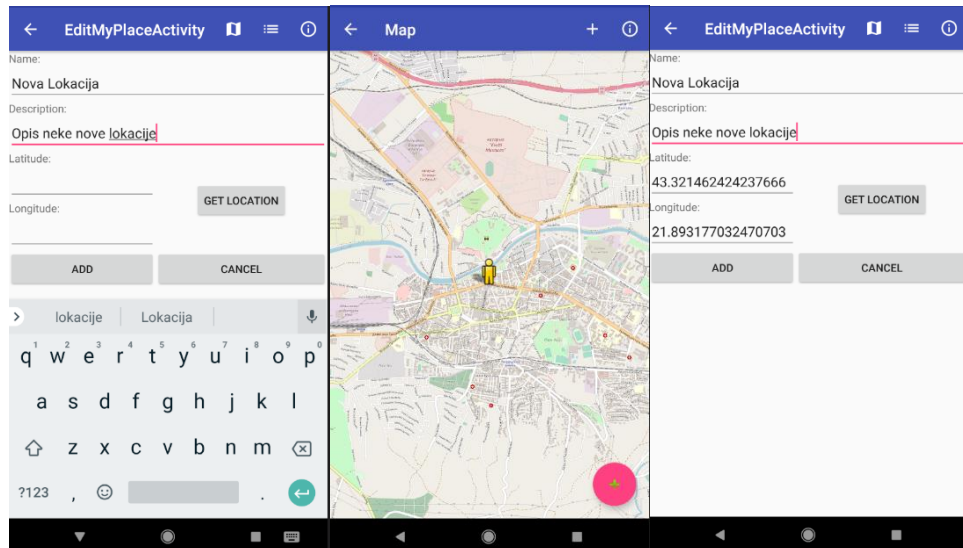


```

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) {
                setMyLocationOverlay();
                setOnMapClickOverlay();
            }
            return;
        }
    }
}

```

- f. Ovim je omogućeno 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).
- 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 {
    MapView map = null;
    IMapController mapController = null;
    MyLocationNewOverlay myLocationOverlay;
    static int NEW_PLACE=1;
    static final int PERMISSION_ACCESS_FINE_LOCATION = 1;

    public static final int SHOW_MAP = 0;
    public static final int CENTER_PLACE_ON_MAP = 1;
    public static final int SELECT_COORDINATES = 2;

    private int state = 0;
    private boolean selCoorsEnabled = false;
    private GeoPoint placeLoc;
}

```

- 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 prosledene preko *Intent*-a. Ukoliko stanje nije `SELECT_COORDINATES`, ukloniti akcionu dugme iz prikaza mape.

```
@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 GeoPoint(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);

    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( packageName: 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);
    }
}
```

c. Dodati i izmeniti sledeće funkcije:

```
private void setCenterPlaceOnMap() {
    mapController = map.getController();
    if (mapController != null) {
        mapController.setZoom(15.0);
        mapController.animateTo(placeLoc);
    }
}

private void setupMap() {
    switch (state) {
        case SHOW_MAP:
            setMyLocationOverlay();
            break;
        case SELECT_COORDINATES:
            mapController = map.getController();
            if (mapController != null) {
                mapController.setZoom(15.0);
                mapController.setCenter(new GeoPoint( aLatitude: 43.3209, aLongitude: 21.8958));
            }
            setOnMapClickOverlay();
            break;
        case CENTER_PLACE_ON_MAP:
        default:
            setCenterPlaceOnMap();
            break;
    }
}

private void setMyLocationOverlay() {
    myLocationOverlay = new MyLocationNewOverlay(new GpsMyLocationProvider( context: this, map);
    myLocationOverlay.enableMyLocation();
    map.getOverlays().add(this.myLocationOverlay);
    mapController = map.getController();
    if (mapController != null) {
        mapController.setZoom(15.0);
        myLocationOverlay.enableFollowLocation();
    }
}
```

d. Dodati odgovarajuće pozive funkciji *setupMap()*.

```

Context ctx = getApplicationContext();
Configuration.getInstance().load(ctx, PreferenceManager.getDefaultSharedPreferences(ctx));
map = (MapView) findViewById(R.id.map);
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{
    setupMap();
}

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) {
                setupMap();
            }
            return;
        }
    }
}

```

- e. 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(0, 1, 1, 1, charSequence: "Select Coordinates");
        menu.add(0, 2, 2, 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);
}

```

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

```
private void setOnMapClickOverlay() {
    MapEventsReceiver mReceive = new MapEventsReceiver() {
        @Override
        public boolean singleTapConfirmedHelper(GeoPoint p) {
            if(state == SELECT_COORDINATES && selCoorsEnabled) {
                String lon = Double.toString(p.getLongitude());
                String lat = Double.toString(p.getLatitude());
                Intent locationIntent = new Intent();
                locationIntent.putExtra(name: "lon", lon);
                locationIntent.putExtra(name: "lat", lat);
                setResult(Activity.RESULT_OK, locationIntent);
                finish();
            }
            return false;
        }

        @Override
        public boolean longPressHelper(GeoPoint p) {
            return false;
        }
    };
};
```

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

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

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

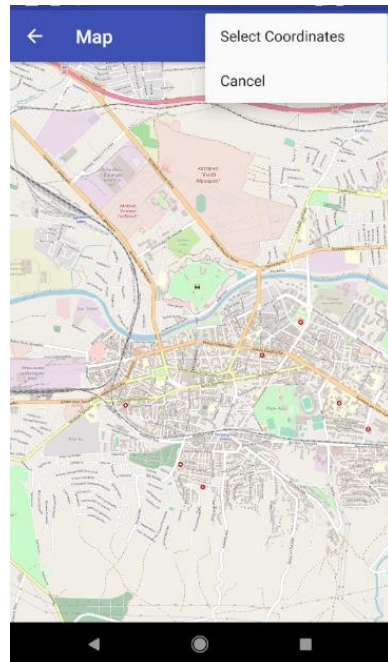
case R.id.editmyplace_location_button: {
    Intent i = new Intent(getApplicationContext(), MyPlacesMapActivity.class);
    i.putExtra(name: "state", MyPlacesMapActivity.SELECT_COORDINATES);
    startActivityForResult(i, requestCode: 1);
}
}
```

```
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);
}
```

- b. Probati aplikaciju.
- c. 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* objekata, tako da konačno metod izgleda kao što je prikazano na slici.

```
private void showMyPlaces() {
    if (myPlacesOverlay != null)
        this.map.getOverlays().remove(myPlacesOverlay);
    final ArrayList<OverlayItem> items = new ArrayList<>();
    for (int i = 0; i < MyPlacesData.getInstance().getMyPlaces().size(); i++) {
        MyPlace myPlace = MyPlacesData.getInstance().getMyPlaces().get(i);
        OverlayItem item = new OverlayItem(myPlace.name, myPlace.description, new GeoPoint(Double.parseDouble(myPlace.latitude), Double.parseDouble(myPlace.longitude)));
        item.setMarker(this.getResources().getDrawable(R.drawable.myPlace));
        items.add(item);
    }
    myPlacesOverlay = new ItemizedIconOverlay<>(items,
        new ItemizedIconOverlay.OnItemGestureListener<OverlayItem>() {
            @Override
            public boolean onItemSingleTapUp(final int index, final OverlayItem item) {
                Intent intent = new Intent(packageContext: MyPlacesMapActivity.this, ViewMyPlaceActivity.class);
                intent.putExtra(name: "position", index);
                startActivity(intent);
                return true;
            }

            @Override
            public boolean onItemLongPress(final int index, final OverlayItem item) {
                Intent intent = new Intent(packageContext: MyPlacesMapActivity.this, EditMyPlaceActivity.class);
                intent.putExtra(name: "position", index);
                startActivityForResult(intent, EDIT_PLACE);
                return true;
            }
        }, getApplicationContext());
    this.map.getOverlays().add(myPlacesOverlay);
}
```

```
@SuppressWarnings("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, 15));
                addMyPlaceMarkers();
            }
            return;
        }
    }
}
```

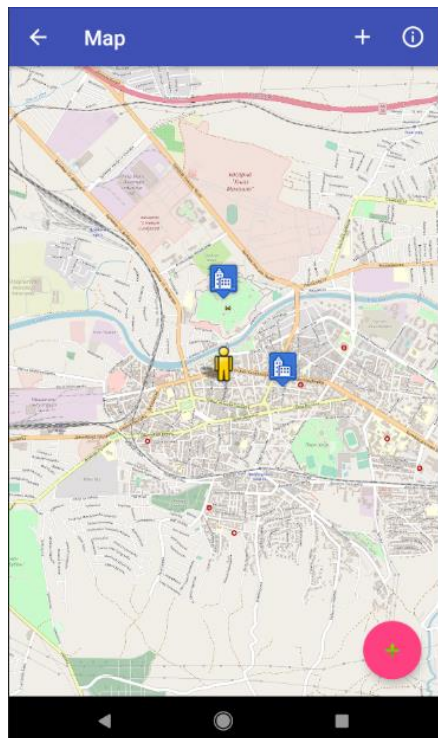
- d. Dodati poziv funkcije *showMyPlaces()* na kraj funkcije *setupMap()*

```
private void setupMap() {
    switch (state) {
        case SHOW_MAP:
            setMyLocationOverlay();
            break;
        case SELECT_COORDINATES:
            mapController = map.getController();
            if (mapController != null) {
                mapController.setZoom(15.0);
                mapController.setCenter(new GeoPoint(43.3209, 21.8958));
            }
            setOnMapClickListener();
            break;
        case CENTER_PLACE_ON_MAP:
        default:
            setCenterPlaceOnMap();
            break;
    }
    showMyPlaces();
}
```

- e. Dodati implementaciju *onActivityResult()* funkcije u *MyPlacesMapsActivity.java* kako bi bio prikazan objekat koji je dodat u listu.

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (resultCode == Activity.RESULT_OK) {
        setList();
    }
}
```

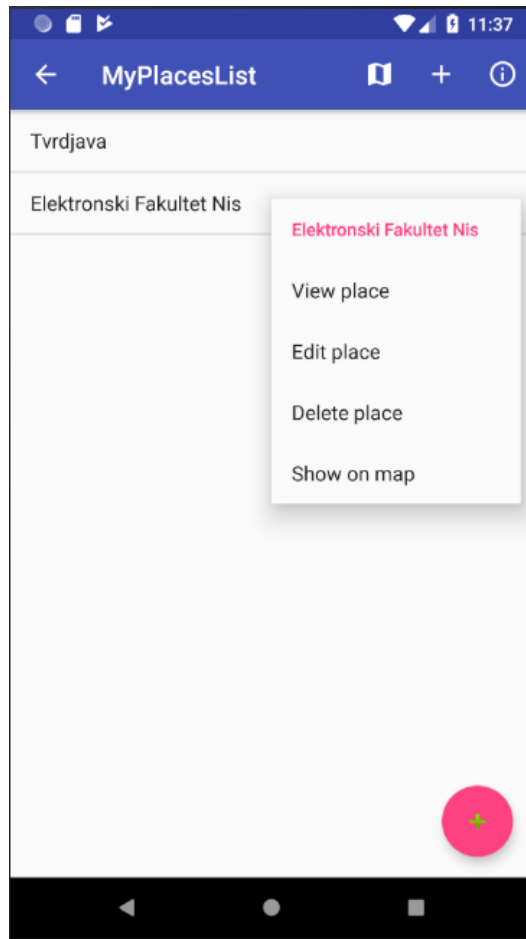
- f. Probati aplikaciju



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 dodatnu 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(0, 1, 1, CharSequence: "View place");
    contextMenu.add(0, 2, 2, CharSequence: "Edit place");
    contextMenu.add(0, 3, 3, CharSequence: "Delete place");
    contextMenu.add(0, 4, 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*

11. Domaći: Optimizovati kod koji vrši ažuriranje prikaza sloja mape koji sadrži objekte.