

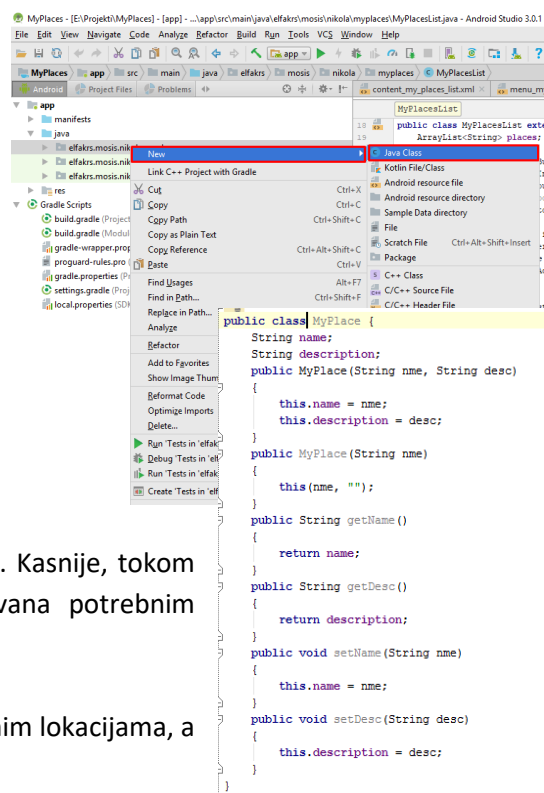
MOSIS - Laboratorijska vežba 3

Na prošloj vežbi je kreiran osnovni kostur aplikacije *MyPlaces*. Aplikaciji je potrebno dodati osnovnu logiku tako da obezbedi korisniku dodavanje omiljenih lokacija, njihovo pregledanje i editovanje. Logiku aplikacije je potrebno implementirati kreiranjem različitih klasa koje služe za smeštanje podataka kao i stranica koje omogućavaju rad sa omiljenim lokacijama. Kao što je na prethodnoj laboratorijskoj vežbi napomenuto ovaj projekat će kroz laboratorijske vežbe biti nadograđivan i neophodno je da studenti redovno rade na njemu i čuvaju prethodne verzije.

Ciljevi ove vežbe su:

1. Kreiranje klase koja opisuje omiljenu lokaciju (*MyPlace.java*)
2. Kreiranje klase u kojoj će biti smešteni podaci o omiljenim lokacijama, a kojoj će se pristupati iz ostalih *Activity*-a.
3. Povezivanje *MyPlacesList Activity*-a sa modelom podataka.
4. Definisanje *Activity*-a za unos, editovanje i pregledanje omiljene lokacije.
5. Kreiranje *EditMyPlaceActivity*-a koji će u početku biti korišćen za dodavanje *Activity*-a a kasnije i za njihovo editovanje.
6. Kreiranje *ViewMyPlaceActivity*-a.
7. Omogućavanje editovanja postojeće lokacije korišćenjem *EditMyPlaceActivity*-ja i konteksnog menija.

1. Kreiranje klase koja opisuje omiljenu lokaciju (*MyPlace.java*)
 - a. Potrebno je pokrenuti *Android Studio*.
 - b. Kreirati u projektu *MyPlaces* novu klasu *MyPlace.java* (desni klik na paket u aplikaciji->*New*->*Java Class*). Nova klasa treba da bude obična *Java* klasa.
 - c. Za početak klasa treba da sadrži privatna polja *name* i *description* tipa *String*, odgovarajuće konstruktore i javne *get* i *set* metode za polja. Kasnije, tokom razvoja aplikacije, ova klasa će biti proširivana potrebnim poljima i metodama.
2. Kreiranje klase u kojoj će biti smešteni podaci o omiljenim lokacijama, a kojoj će se pristupati iz ostalih *Activity*-a.



- a. Na sličan način kao u prethodnom koraku, potrebno je kreirati klasu *MyPlacesData.java* koja će držati instance klase *MyPlace.java* za svaku omiljenu lokaciju koju budemo kreirali.
 - b. Klasa treba da sadrži privatno polje (objekat) generičke liste *ArrayList<MyPlace>* u koju ćemo smestati kreirane instance *MyPlace* klase.
 - c. Pošto ćemo ovoj klasi pristupati iz različitih klasa projekta a želimo da nam se ključni podaci nalaze na jednom mestu, dobro bi bilo ovu klasu implementirati prema *Singleton* paternu. Vodićemo računa da implementirana klasa bude *Thread Safe Singleton*
 - d. Implementirati i metode za pristup celoj listi, pristup elementu liste, dodavanje novog elementa u listu i brisanje elementa iz liste.
3. Povezivanje *MyPlacesList Activity*-a sa modelom podataka.
- a. Na prošloj laboratorijskoj vežbi je dodat *Activity* sa *ListView* *View* kontrolom za prikaz omiljenih mesta tako da koristi podatke iz statičke liste.
 - b. U cilju povezivanja liste sa postojećim modelom podataka koji se zasniva na generičkoj *ArrayList*-i objekata potrebno je koristiti generički *ArrayAdapter*. Kako bi *ArrayAdapter* mogao da prikaže ime *MyPlace* instance, potrebno je u *MyPlace* klasi pri izvršiti *override* metode *toString()*. Dovoljno je da *toString()* metoda vrati vrednost *name* polja klase.
 - c. U *MyPlacesList.java* klasi je potrebno obrisati lokalnu listu, njenu inicijalizaciju i zameniti argumente prilikom poziva metoda *setListAdapter* tako da mu se, kao argument, prosleđuje lista klase *MyPlacesData.java*.

```
public class MyPlacesData {
    private ArrayList<MyPlace> myPlaces;
    private MyPlacesData() {
        myPlaces = new ArrayList<MyPlace>();
    }

    private static class SingletonHolder {
        public static final MyPlacesData instance = new MyPlacesData();
    }

    public static MyPlacesData getInstance() {
        return SingletonHolder.instance;
    }

    public ArrayList<MyPlace> getMyPlaces() {
        return myPlaces;
    }

    public void addNewPlace(MyPlace place) {
        myPlaces.add(place);
    }

    public MyPlace getPlace(int index) {
        return myPlaces.get(index);
    }

    public void deletePlace(int index) {
        myPlaces.remove(index);
    }
}
```

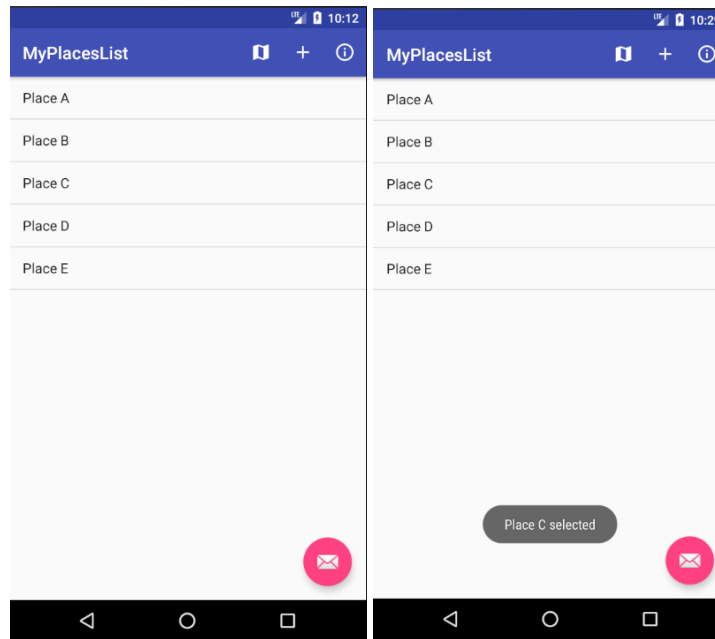
```
@Override
public String toString() {
    return this.name;
}
```

```
public class MyPlacesList extends ActionBarActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_my_places_list);
        ListView myPlacesList = (ListView) findViewById(R.id.my_places_list);
        myPlacesList.setAdapter(new ArrayAdapter<MyPlace>(this, android.R.layout.simple_list_item_1, MyPlacesData.getInstance().getMyPlaces()));
    }
}
```

Nakon ovoga, lista će biti automatski povezana sa podacima koji se nalaze u *ArrayList*-i *MyPlacesData.java* klase.

- d. U konstruktor klase *MyPlacesData* potrebno je dodati nekoliko instanci *MyPlace* klase, kako bi bilo moguće testirati trenutnu implementaciju.
- e. Pokrenuti aplikaciju.

```
public class MyPlacesData {
    private ArrayList<MyPlace> myPlaces;
    private MyPlacesData() {
        myPlaces = new ArrayList<MyPlace>();
        myPlaces.add(new MyPlace("Place A"));
        myPlaces.add(new MyPlace("Place B"));
        myPlaces.add(new MyPlace("Place C"));
        myPlaces.add(new MyPlace("Place D"));
        myPlaces.add(new MyPlace("Place E"));
    }
}
```



- f. Potrebno je dodati i metod koji će nakon klika na element liste prikazati *Toast* poruku u kojoj piše ime omiljenog mesta. Za ovo je potrebno postaviti *OnItemClickListener* na instancu *ListView*-a.

```
public class MyPlacesList extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_my_places_list);
        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) {
                Snackbar.make(view, "Replace with your own action", Snackbar.LENGTH_LONG)
                    .setAction("Action", null).show();
            }
        });

        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()));

        myPlacesList.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {
                MyPlace place = (MyPlace) adapterView.getAdapter().getItem(i);
                Toast.makeText(getApplicationContext(), "place: " + place.getName() + " selected", Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

- g. Da bi Activity mogao da ima podršku za navigaciju, i kako bi klikom na strelicu u zaglavlju Activity-a bilo moguće vraćanje na pozivajući Activity, potrebno je dodati u *onCreate* metod *MyPlacesList* Activity-a sledeći kod.

```
public class MyPlacesList extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_my_places_list);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);

        FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
        fab.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Snackbar.make(view, "Replace with your own action", Snackbar.LENGTH_LONG)
                    .setAction("Action", null).show();
            }
        });

        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()));
    }

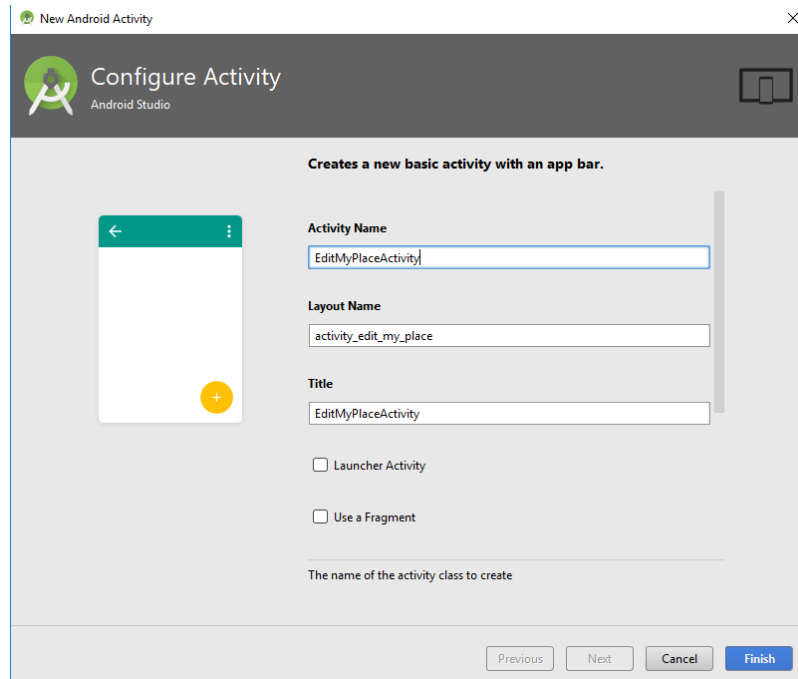
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();

        //noinspection SimplifiableIfStatement
        if (id == R.id.show_map_item) {
            Toast.makeText(context, "Show Map!", Toast.LENGTH_SHORT).show();
        } else if (id == R.id.new_place_item) {
            Toast.makeText(context, "New Place!", Toast.LENGTH_SHORT).show();
        } else if (id == R.id.about_item) {
            Intent i = new Intent(context, About.class);
            startActivity(i);
        } else if (id == android.R.id.home) {
            finish();
        }

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

4. Kreiranje *EditMyPlaceActivity*-a:

- a. Dodati novi *Activity* pod imenom *EditMyPlaceActivity* u projekat.



- b. *Android Studio* podrazumevano dodaje *ConstraintLayout* kao roditeljski *layout* u *Activity*-u u XML fajlu *content_edit_my_okace.xml*. Potrebno je zameniti *ConstraintLayout* u *LinearLayout*. Dodati odgovarajuće *TextView* i *EditText* kontrole.



e. Prva verzija ovog *Activity*-a je namenjena dodavanju nove lokacije. U kodu je potrebno postaviti click listener-e za oba dugmeta, postaviti tekst "Add" *finish* dugmetu. Radi vežbe, umesto *inline* dodavanja *OnClickListener*-a dugmetu (kao kod *About Activity*-a), potrebno

je implementirati metode *OnClickListener* interfejsa u klasi *EditMyPlaceActivity*.

```
public class EditMyPlaceActivity extends AppCompatActivity implements View.OnClickListener{

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_edit_my_place);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        Button finishedButton = (Button) findViewById(R.id.editmyplace_finished_button);
        finishedButton.setOnClickListener(this);
        Button cancelButton = (Button) findViewById(R.id.editmyplace_cancel_button);
        cancelButton.setOnClickListener(this);
    }

    @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();
                MyPlace place = new MyPlace(nme, desc);
                MyPlacesData.getInstance().addNewPlace(place);
                setResult(Activity.RESULT_OK);
                finish();
                break;
            }

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

- f. Moguće je uočiti da se nakon klika na bilo koje dugme *Activity* završava. Poželjno je u takvim situacijama vratiti rezultat jer takva informacija može biti značajna iz konteksta pozivajućeg *Activity*-a. U tom slučaju u pozivajućem *Activity*-u (*MainActivity.java* i *MyPlacesList.java*) neophodno je koristiti metod *startActivityForResult()*. Obradu vraćenog rezultata je potrebno izvršiti u metodi *onActivityResult()* u pozivajućem *Activity*-u. U klasu *MainActivity.java* je potrebno dodati sledeći kod (na sličan način implementirati i u *MyPlacesList.java* klasi):

```
static int NEW_PLACE = 1;

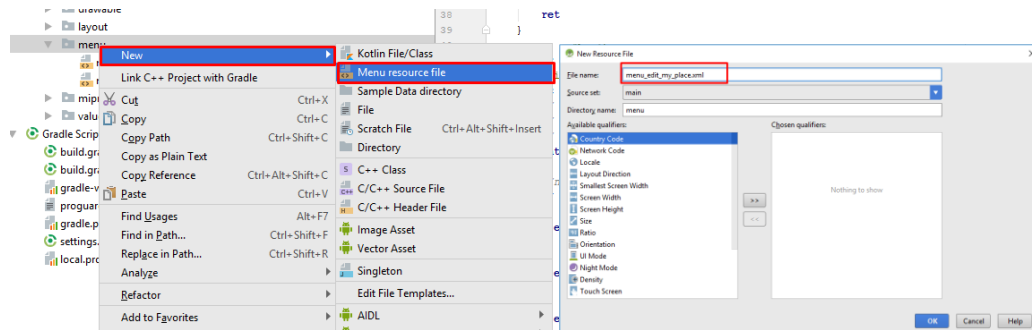
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();

    //noinspection SimplifiableIfStatement
    if (id == R.id.show_map_item) {
        Toast.makeText(context, this, "Show Map!", Toast.LENGTH_SHORT).show();
    } else if (id == R.id.new_place_item) {
        Intent i = new Intent(context, EditMyPlaceActivity.class);
        startActivityForResult(i, NEW_PLACE);
    } else if (id == R.id.my_places_list_item) {
        Intent i = new Intent(context, MyPlacesList.class);
        startActivity(i);
    } else if (id == R.id.about_item) {
        Intent i = new Intent(context, About.class);
        startActivity(i);
    }

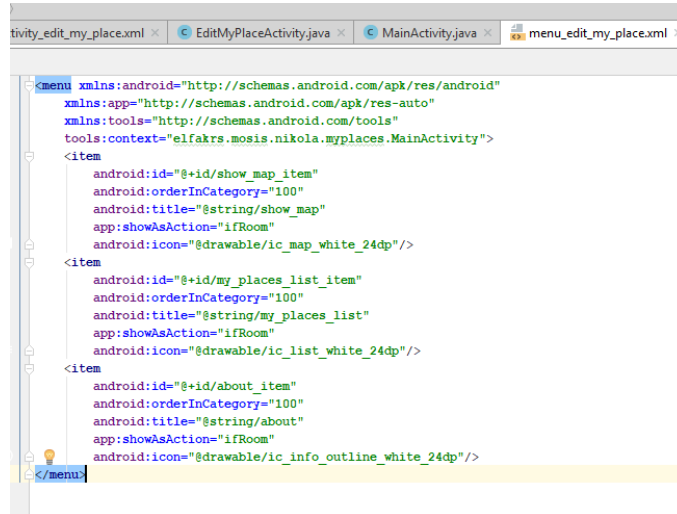
    return super.onOptionsItemSelected(item);
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if(resultCode == Activity.RESULT_OK) {
        Toast.makeText(context, this, "New Place added", Toast.LENGTH_SHORT).show();
    }
}
```

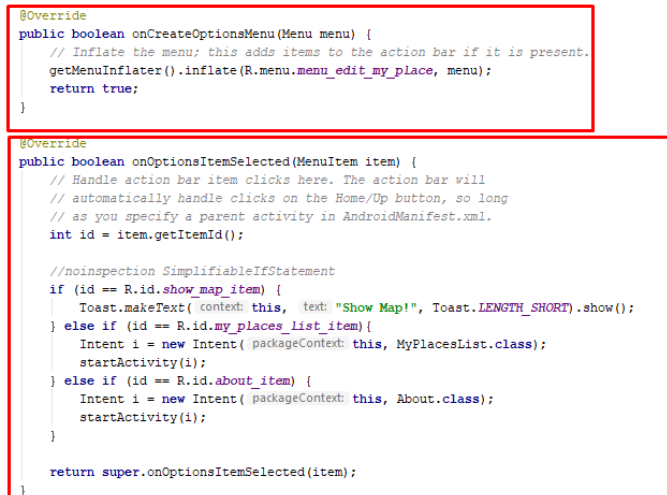
- g. Dodati XML fajl za meni *EditMyPlace.java Activity*-akoje će biti vidljive na *Toolbar*-u:



h. Potrebno je iskopirati sadržaj menu_main.xml bez new_place_item-a.



i. I dodati odgovarajuće pozive koji se dešavaju na klik na item-a Action Bar-a.



j. Dodati podršku za back dugme.

```

public class EditMyPlaceActivity extends AppCompatActivity implements View.OnClickListener{

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_edit_my_place);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        Button finishedButton = (Button) findViewById(R.id.editmyplace_finished_button);
        finishedButton.setOnClickListener(this);
        Button cancelButton = (Button) findViewById(R.id.editmyplace_cancel_button);
        cancelButton.setOnClickListener(this);
    }

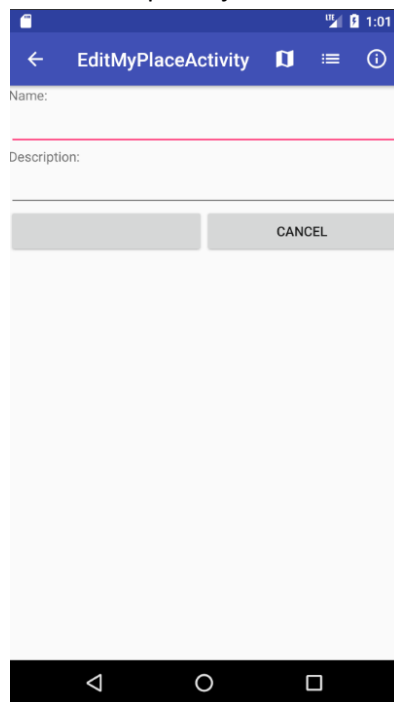
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();

        //noinspection SimplifiableIfStatement
        if (id == R.id.show_map_item) {
            Toast.makeText( context this, text: "Show Map!", Toast.LENGTH_SHORT).show();
        } else if (id == R.id.my_places_list_item) {
            Intent i = new Intent( packageName: this, MyPlacesList.class);
            startActivity(i);
        } else if (id == R.id.about_item) {
            Intent i = new Intent( packageName: this, About.class);
            startActivity(i);
        } else if (id == android.R.id.home) {
            finish();
        }

        return super.onOptionsItemSelected(item);
    }
}

```

k. Pokrenuti aplikaciju.



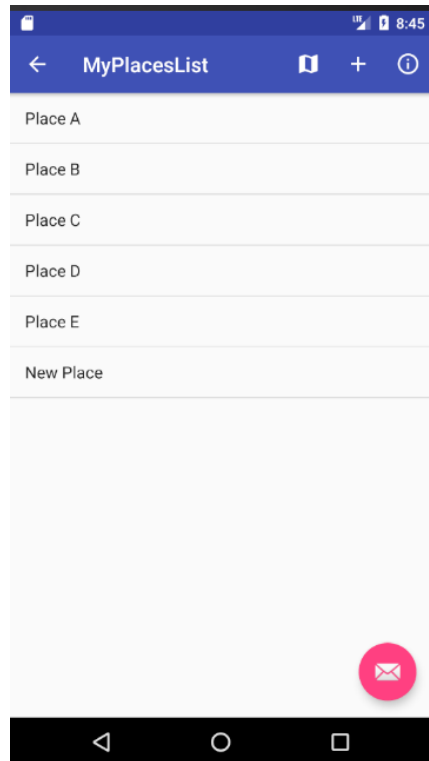
l. Ukoliko je Activity *EditMyPlaceActivity* pozvan iz Action Bar-a Activity-a *MyPlacesList*, potrebno je da lista bude osvežena tako da dodati element liste bude prikazan. Dovoljno je ponovo pozvati *setListAdapter* metod.


```

    return super.onOptionsItemSelected(item);
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data)
{
    super.onActivityResult(requestCode, resultCode, data);
    if (resultCode == Activity.RESULT_OK)
    {
        ListView myPlacesList = (ListView) findViewById(R.id.my_places_list);
        myPlacesList.setAdapter(new ArrayAdapter<MyPlace>(this, android.R.layout.simple_list_item_1, MyPlacesData.getInstance().getMyPlaces()));
    }
}
}

```



- m. Trenutno je moguće dodati novu lokaciju čak i ako su oba polja za unos prazna. Da bi ovo bilo onemogućeno, potrebno je zabraniti korisniku klik na dugme za dodavanje ukoliko nikakav tekst nije unet u polje za ime lokacije.
- Potrebno je postaviti u *OnCreate* metodi *Activity*-a dugme *finished* u neaktivno stanje.
 - Potrebno je implementirati *TextWatcher* i pratiti promene teksta u *EditText* polju imena. Ukoliko postoji neki tekst, postaviti da dugme bude aktivno.

```

public class EditMyPlaceActivity extends AppCompatActivity implements View.OnClickListener{

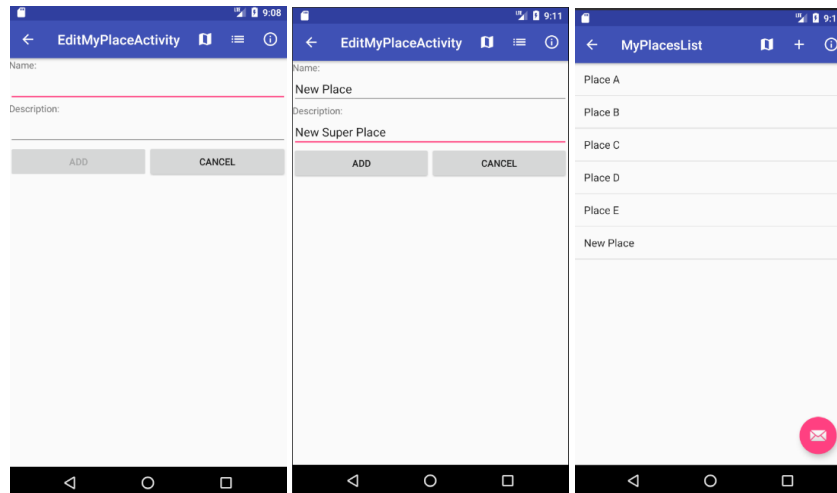
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_edit_my_place);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        getSupportActionBar().setDisplayShowHomeEnabled(true);
        final Button finishedButton = (Button) findViewById(R.id.editmyplace_finished_button);
        finishedButton.setOnClickListener(this);
        finishedButton.setEnabled(false);
        finishedButton.setText("Add");
        Button cancelButton = (Button) findViewById(R.id.editmyplace_cancel_button);
        cancelButton.setOnClickListener(this);
        EditText nameEditText = (EditText) findViewById(R.id.editmyplace_name_edit);
        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);
            }
        });
    }
}

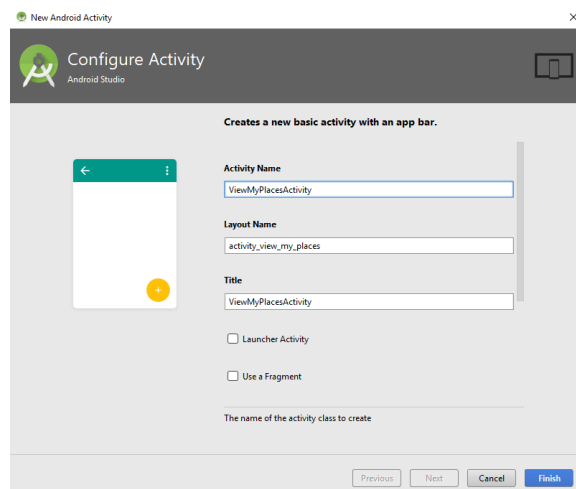
```

n. Probajte aplikaciju. Dodajte novu lokaciju i probajte listu.



5. Kreiranje *ViewMyPlaceActivity*-a.

a. Implementirati *Activity* na osnovu dosadašnjih primera. Nakon kreiranja ukloniti *Action button*.



```

<string name="title_activity_my_places_list">MyPlacesList</string>
<string name="title_activity_edit_my_place">EditMyPlaceActivity</string>
<string name="editmyplace_name_label">Name:</string>
<string name="editmyplace_desc_label">Description:</string>
<string name="editmyplace_cancel_label">Cancel</string>
<string name="title_activity_view_my_places">ViewMyPlacesActivity</string>
<string name="viewmyplace_finished_button">OK</string>

```

./resources>

```

<?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$ScrollingView..."
    tools:context="elfakrs.mosis.nikola.myplaces.ViewMyPlacesActivity"
    tools:showIn="@layout/activity_view_my_places"
    android:orientation="vertical">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/editmyplace_name_label"/>
    <TextView
        android:id="@+id/viewmyplace_name_text"
        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_desc_label"/>
    <TextView
        android:id="@+id/viewmyplace_desc_text"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />
    <Button
        android:id="@+id/viewmyplace_finished_button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/viewmyplace_finished_button"/>
</LinearLayout>

```

- Klik na neku od stavki iz liste treba da prikaže izabranu lokaciju u kreiranom *Activity*-ju. Za to je potrebno proslediti podatak o poziciji kliknutog elementa kroz *Bundle* objekat koji dodeljujemo *Intent*-u prilikom poziva.
- Taj podatak u metodi *onCreate* *ViewMyPlaceActivity*-ja preuzimamo i koristimo za preuzimanje odgovarajuće lokacije iz klase sa podacima.
- Nakon promena, *MyPlaceListActivity* izgleda ovako:

```

public class MyPlacesList extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_my_places_list);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);

        FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
        fab.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Snackbar.make(view, "Replace with your own action", Snackbar.LENGTH_LONG)
                    .setAction("Action", null).show();
            }
        });
        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()));
        myPlacesList.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {
                Bundle positionBundle = new Bundle();
                positionBundle.putInt("position", i);
                Intent intent = new Intent( PackageContext: MyPlacesList.this, ViewMyPlacesActivity.class);
                intent.putExtra(positionBundle);
                startActivity(intent);
            }
        });
    }
}

```

- ViewMyPlaceActivity* izgleda kao na slici:

```

public class ViewMyPlacesActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_view_my_places);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        getSupportActionBar().setDisplayShowHomeEnabled(true);
        int position = -1;
        try{
            Intent listIntent = getIntent();
            Bundle positionBundle = listIntent.getExtras();
            position = positionBundle.getInt( key: "position");
        } catch (Exception e) {
            Toast.makeText( context: this, e.getMessage(), Toast.LENGTH_SHORT).show();
            finish();
        }
        if(position >= 0){
            MyPlace place = MyPlacesData.getInstance().getPlace(position);
            TextView twName = (TextView) findViewById(R.id.viewmyplace_name_text);
            twName.setText(place.getName());
            TextView twDesc = (TextView) findViewById(R.id.viewmyplace_desc_text);
            twDesc.setText(place.getDesc());
        }
        final Button finishedButton = (Button) findViewById(R.id.viewmyplace_finished_button);
        finishedButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                finish();
            }
        });
    }
}

```

f. Potrebno je podesiti *Toolbar*.

```

    @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_view_my_place, 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.show_map_item) {
            Toast.makeText( context: this, text: "Show Map!", Toast.LENGTH_SHORT).show();
        } else if (id == R.id.my_places_list_item) {
            Intent i = new Intent( packageContext: this, MyPlacesList.class);
            startActivity(i);
        } 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);
    }
}

```

g. Probati aplikaciju.



6. Omogućavanje editovanja postojeće lokacije korišćenjem *EditMyPlaceActivity*-ja i konteksnog menija.

- Potrebno je registrovati kontekсни meni za listu postavljanjem *onCreateContextMenuListener* objekta koji je kreiran *inline* (ili je implementiran metod *onCreateContextMenu* interfejsa *onCreateContextMenuListener*-a).
- Nakon toga, potrebno je dodati naslov konteksnom meniju i dodati dve stavke, "View place" i "Edit place".

```

startActivity(intent);
}
}

myPlacesList.setOnItemClickListener(new View.OnCreateContextMenuListener() {

    @Override
    public void onCreateContextMenu(ContextMenu contextMenu, View view, ContextMenu.ContextMenuInfo contextMenuInfo) {
        AdapterView.AdapterContextMenuInfo info = (AdapterView.AdapterContextMenuInfo) contextMenuInfo;
        MyPlace place = MyPlacesData.getInstance().getPlace(info.position);
        contextMenu.setHeaderTitle(place.getName());
        contextMenu.add(0, 0, 1, 1, CharSequence: "View place");
        contextMenu.add(0, 0, 2, 2, CharSequence: "Edit place");
    }
});

```

- Potrebno je obratiti pažnju na način preuzimanja pozicije kliknutog elementa liste.
- Obraditi klik kreiranjem odgovarajućih *Intent*-a.

```

@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(this, ViewMyPlaceActivity.class);
        i.putExtras(positionBundle);
        startActivity(i);
    }
    else if (item.getItemId() == 2)
    {
        i = new Intent(this, EditMyPlaceActivity.class);
        i.putExtras(positionBundle);
        startActivityForResult(i, 1);
    }
    return super.onContextItemSelected(item);
}

```

- Potrebno je u *EditMyPlaceActivity* podržati prosleđivanje parametra i u skladu sa tim omogućiti editovanje i snimanje. To je moguće izvršiti proverom da li je u *Bundle* objektu

pozivajućeg *Intent*-a prosleđena pozicija ili nije. Nakon promena izgled *EditMyPlaceActivity*.

```
public class EditMyPlaceActivity extends AppCompatActivity implements View.OnClickListener{
    boolean editMode = true;
    int position = -1;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_edit_my_place);

        try {
            Intent listIntent = getIntent();
            Bundle positionBundle = listIntent.getExtras();
            if(positionBundle != null)
                position = positionBundle.getInt( Key<> "position");
            else
                editMode = false;
        } catch (Exception e) {
            editMode = false;
        }

        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        getSupportActionBar().setDisplayShowHomeEnabled(true);
        final Button finishedButton = (Button) findViewById(R.id.editmyplace_finished_button);
        Button cancelButton = (Button) findViewById(R.id.editmyplace_cancel_button);
        EditText nameEditText = (EditText) findViewById(R.id.editmyplace_name_edit);
        if(!editMode){
            finishedButton.setEnabled(false);
            finishedButton.setText("Add");
        } else if (position >= 0){
            finishedButton.setText("Save");
            MyPlace place = MyPlacesData.getInstance().getPlace(position);
            nameEditText.setText(place.getName());
            EditText descEditText = (EditText) findViewById(R.id.editmyplace_desc_edit);
            descEditText.setText(place.getDesc());
        }
        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 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();
                if(!editMode){
                    MyPlace place = new MyPlace(nme, desc);
                    MyPlacesData.getInstance().addNewPlace(place);
                } else {
                    MyPlace place = MyPlacesData.getInstance().getPlace(position);
                    place.setName(nme);
                    place.setDesc(desc);
                }
                setResult(Activity.RESULT_OK);
                finish();
                break;

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

7. Prepraviti Action dugme u *MyPlacesList* i *MainActivity* tako da ima funkciju dodavanja novog mesta.

```

public class MyPlacesList extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_my_places_list);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        getSupportActionBar().setDisplayShowHomeEnabled(true);

        FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
        fab.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent i = new Intent( packageContext MyPlacesList.this, EditMyPlaceActivity.class);
                startActivityForResult(i, NEW_PLACE);
            }
        });

        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().getMyPla
        myPlacesList.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {
                Bundle positionBundle = new Bundle();
                positionBundle.putInt("position", i);
                Intent intent = new Intent( packageContext MyPlacesList.this, ViewMyPlaceActivity.class);
            }
        });
    }
}

```

`<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>`

8. Obrisati statičko dodavanje lokacija iz konstruktora klase *MyPlacesData* i probati aplikaciju.

