

MODIS - 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 ćemo u početku koristiti 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. 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, klasu ćemo proširivati 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, 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 smestiti 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 smo dodali *Activity* sa *ListView* *View*-em za prikaz omiljenih mesta ali smo je testirali sa podacima iz statičke liste.
 - b. U cilju povezivanja liste sa postojećim modelom podataka koji se zasniva na generičkoj *ArrayList*-i listi 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 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);
    }
}
```

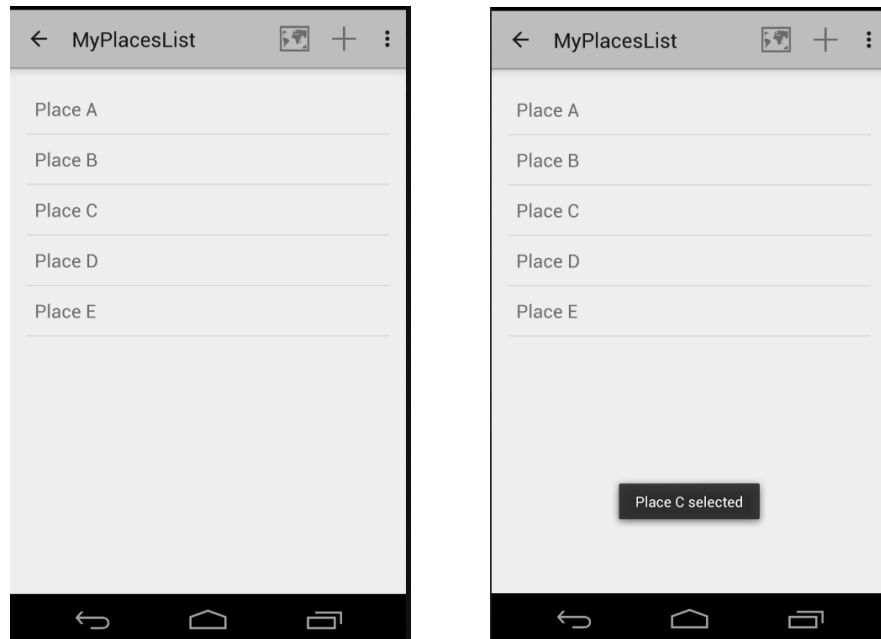
```
@Override
public String toString() {
    return this.name;
}
```

```
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()));
    }
}
```

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

```
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"));
    }
}
```

- d. Ukoliko u konstruktoru klase *MyPlacesData* dodamo nekoliko instanci *MyPlace* klase, možemo da proverimo da li lista prikazuje validne podatke.
- e. Pokrenite aplikaciju.

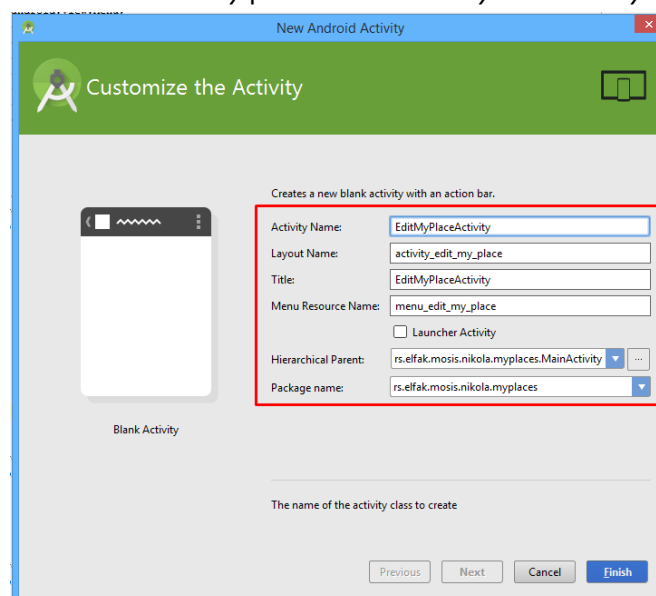


- f. Dodajmo i metod koji će nakon klika na element liste prikazati *Toast* poruku u kojoj piše ime omiljenog mesta. Za ovo je potrebno postaviti *OnItemClickListener* na instancu *ListView*-a.

```
public class MyPlacesList extends 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()));
        myPlacesList.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
                MyPlace place = (MyPlace) parent.getAdapter().getItem(position);
                Toast.makeText(getApplicationContext(), place.getName() + " selected", Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

4. Kreiranje *EditMyPlaceActivity*-a:

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



- b. *Android Studio* podrazumevano dodaje *RelativeLayout* kao roditeljski *layout* u *Activity*-u. U cilju vežbanja drugačijih *layout*-a, zameniti *RelativeLayout* u *LinearLayout*. Dodati odgovarajuće *TextView* i *EditText* *View*-eve Kako bi smo vežbali različite *layout*-e.

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="fill_parent"
    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    tools:context="rs.elfak.mosis.nikola.myapplication.EditMyPlaceActivity"
    android:orientation="vertical">
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/editmyplace_name_label"
        />
    <EditText
        android:id="@+id/editmyplace_name_edit"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        />
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/editmyplace_desc_label"
        />
    <EditText
        android:id="@+id/editmyplace_desc_edit"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        />
    <TableRow
        android:layout_width="fill_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="@string/editmyplace_cancel_label" />
    </TableRow>
</LinearLayout>
```

- c. Dodati odgovarajuće string resurse.

```
<resources>
    <string name="app_name">My Places</string>
    <string name="show_map">Show Map</string>
    <string name="new_place">New place</string>
    <string name="my_places_list">My Places</string>
    <string name="about">About</string>
    <string name="about_text">
        <i>My places</i> is the application that will help you remember charact
        It will be able to show your places on map and enable you to share them
        and you will earn <i>points</i> from your teaching assistant.
    </string>
    <string name="ok">OK</string>
    <string name="title_activity_my_places_list">MyPlacesList</string>
    <string name="action_settings">Settings</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>
</resources>
```

- d. Prva verzija ovog *Activity*-a će biti namenjena dodavanju nove lokacije. U kodu postaviti click listener-e za oba dugmeta, postaviti tekst "Add" finish dugmetu. Radi vežbe, umesto inline dodavanja *OnClickListener*-a dugmetu (kako je urađeno u *About Activity*-

u) implementirati metode OnClickListener interfejsa u klasi EditMyPlaceActivity.

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

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_edit_my_place);
        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 v) {
        switch(v.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;
        }
    }
}
```

- e. Možete uočiti da se nakon klika na bilo koje dugme Activity završava. Dobro je vratiti rezultat u takvim situacijama jer nam kasnije takva informacija može biti značajna. U tom slučaju u potivajućem Activity-ju (MainActivity.java i MyPlacesList.java) moramo da koristimo metod *startActivityForResult()*. Ukoliko želimo da obrađujemo vraćeni rezultat potrebno je *override*-ovati i metod *onActivityResult()* u pozivajućem Activity-ju. U klasi *MainActivity.java* (na sličan način implementirati i u *MyPlacesList.java* klasi):

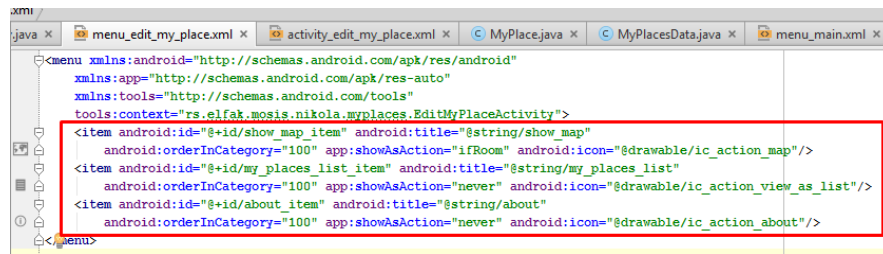
```
static int NEW_PLACE = 1;
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();

    if (id == R.id.show_map_item) {
        Toast.makeText(this, "Show Map!", Toast.LENGTH_SHORT).show();
    } 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);
}

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

- f. Dodati stavke menija klase *EditMyPlace.java* koje će biti vidljive na *Action Bar*-u:



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

```

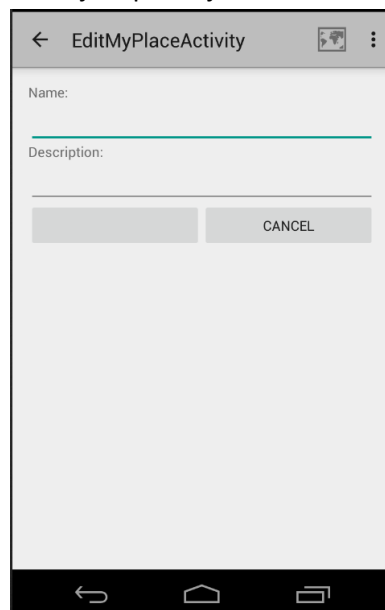
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();

    if (id == R.id.show_map_item) {
        Toast.makeText(this, "Show Map!", Toast.LENGTH_SHORT).show();
    } 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);
}

```

- h. Probajte aplikaciju.



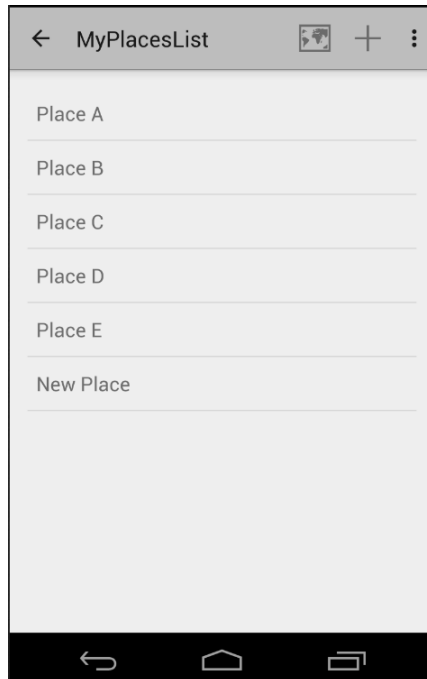
- i. Ukoliko je *Activity EditMyPlaceActivity* pozvan iz *Action Bar*-a *Activity*-a *MyPlacesList*, potrebno je da lista bude osvežena tako da dodati element liste bude prikazan. Dovoljno je ponovo pozvati *setListAdapter* metod.

```

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()));
    }
}
}

```



- j. Trenutno je moguće dodati novu lokaciju čak i ako su oba polja za unos prazna. Da bi ovo rešili moramo da onemogućimo korisniku klik na dugme za dodavanje ukoliko nikakav tekst nije dodan u polje za ime lokacije.
- Postaviti u *onCreate* metodi *Activity*-a da dugme *finished* bude neaktivno
 - Implementirati *TextWatcher* i pratiti promene teksta u *EditText* polju za ime. Ukoliko postoji neki tekst postaviti da dugme bude aktivno.

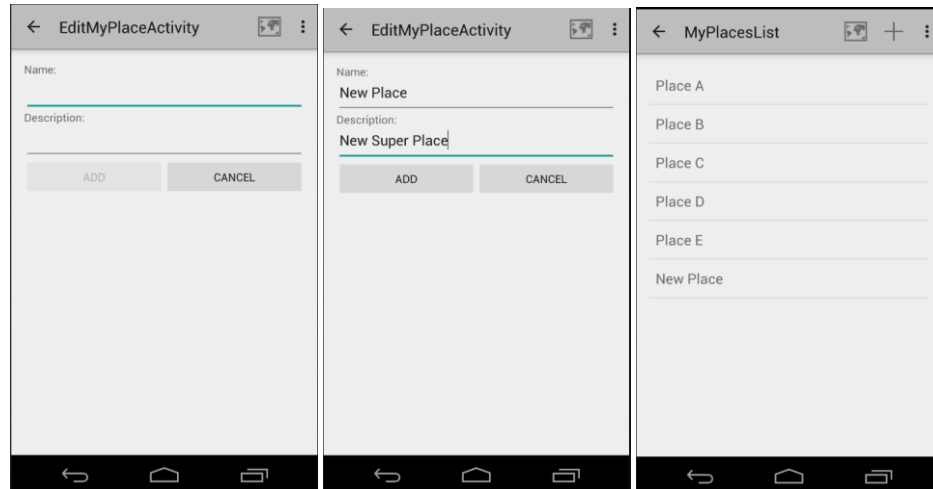
```
public class EditMyPlaceActivity extends ActionBarActivity implements View.OnClickListener{

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_edit_my_place);
        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 s, int start, int count, int after) {
            }

            @Override
            public void onTextChanged(CharSequence s, int start, int before, int count) {
            }

            @Override
            public void afterTextChanged(Editable s) {
                finishedButton.setEnabled(s.length() > 0);
            }
        });
    }
}
```

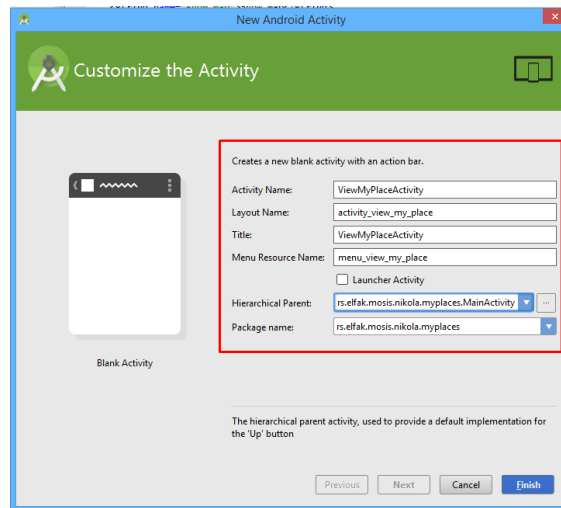
- k. Probajte aplikaciju. Dodajte novu lokaciju i probajte listu.



5. Kreiranje *ViewMyPlaceActivity*-a.

- a. Sa postojećim znanjem, implementacija *Activity*-a je jednostavna.

```
<string name="editmyplace_name_label">Name:</string>
<string name="editmyplace_desc_label">Description:</string>
<string name="editmyplace_cancel_label">Cancel</string>
<string name="viewmyplace_finised_label">OK</string>
resources>
```




```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
    android:layout_height="match_parent" android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    android:paddingBottom="16dp"
    tools:context="rs.elifak.mosis.nikola.myplaces.ViewMyPlaceActivity"
    android:orientation="vertical">
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/editmyplace_name_label"
    />
    <TextView
        android:id="@+id/viewmyplace_name_text"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
    />
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/editmyplace_desc_label"
    />
    <TextView
        android:id="@+id/viewmyplace_desc_text"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
    />
    <Button
        android:id="@+id/viewmyplace_finished_button"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/viewmyplace_finished_label"
    />
</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.
- Sada *MyPlacesListActivity* izgleda ovako:

```

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

```

- Obratite pažnju na poziv *Intent*-a.
- ViewMyPlaceActivity* izgleda ovako:

```

public class ViewMyPlaceActivity extends ActionBarActivity {

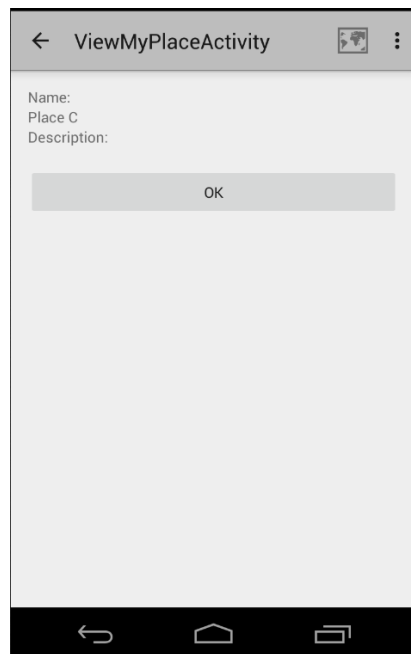
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_view_my_place);

        int position = -1;
        try
        {
            Intent listIntent = getIntent();
            Bundle positionBundle = listIntent.getExtras();
            position = positionBundle.getInt("position");
        }
        catch (Exception e) {
            Toast.makeText(this, e.getMessage(), Toast.LENGTH_LONG).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 v) {
                finish();
            }
        });
    }
}

```

g. Podesite *Action Bar* item-e

h. Probajte aplikaciju.



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

- a. Registrovati kontekсни meni za listu postavljanjem *OnCreateContextMenuListener* objekta koji je kreiran *inline* (ili je implementiran metod *onCreateContextMenu* intefejsa *OnCreateContextMenuListener*-a).
- b. Dati naslov konteksnom meniju i dodati dve stavke, "View place" i "Edit place".

```

myPlacesList.setAdapter(new ArrayAdapter<MyPlace>(this, android.R.layout.simple_list_item_1, MyPlacesData.get
myPlacesList.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
        Bundle positionBundle = new Bundle();
        positionBundle.putInt("position", position);
        Intent i = new Intent(MyPlacesList.this, ViewMyPlaceActivity.class);
        i.putExtra(positionBundle);
        startActivity(i);
    }
});
myPlacesList.setOnCreateContextMenuListener(new View.OnCreateContextMenuListener() {
    @Override
    public void onCreateContextMenu(ContextMenu menu, View v, ContextMenu.ContextMenuInfo menuInfo) {
        AdapterView.AdapterContextMenuInfo info = (AdapterView.AdapterContextMenuInfo)menuInfo;
        MyPlace place = MyPlacesData.getInstance().getPlace(info.position);
        menu.setHeaderTitle(place.getName());
        menu.add(0, 1, 1, "View place");
        menu.add(0, 2, 2, "Edit place");
    }
});

```

- c. Obratiti pažnju kako se preuzima pozicija kliknutog elementa liste.
- d. 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.putExtra(positionBundle);
            startActivity(i);
        }
        else if (item.getItemId() == 2)
        {
            i = new Intent(this, EditMyPlaceActivity.class);
            i.putExtra(positionBundle);
            startActivityForResult(i, 1);
        }
        return super.onContextItemSelected(item);
    }

```

- e. Ostalo je još da u klasi *EditMyPlaceActivity* podržimo prosleđivanje parametra *i* u skladu sa tim omogućimo editovanje i snimanje. To možemo da omogućimo proverom da li je u *Bundle* objektu pozivajućeg *Intent*-a prosleđena pozicija ili nije. Sledi novi izgled klase *EditMyPlaceActivity*.

```

public class EditMyPlaceActivity extends ActionBarActivity 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("position");
            else
                editMode = false;
        } catch (Exception e) {
            editMode = false;
        }
        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);
        cancelButton.setOnClickListener(this);
        nameEditText.addTextChangedListener(new TextWatcher() {
            @Override
            public void beforeTextChanged(CharSequence s, int start, int count, int after) {
            }
            @Override
            public void onTextChanged(CharSequence s, int start, int before, int count) {}
            @Override
            public void afterTextChanged(Editable s) {
                finishedButton.setEnabled(s.length() > 0);
            }
        });
    }
}

```

```

@Override
public void onClick(View v) {
    switch(v.getId())
    {
        case R.id.editmyplace_finished_button:
            EditText nameEditText = (EditText)findViewById(R.id.editmyplace_name_edit);
            String nme = nameEditText.getText().toString();
            EditText descEditText = (EditText)findViewById(R.id.editmyplace_desc_edit);
            String desc = descEditText.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;
    }
}

@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_edit_my_place, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();
    if (id == R.id.show_map_item) {
        Toast.makeText(this, "Show Map!", Toast.LENGTH_SHORT).show();
    } 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);
}
}

```

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

