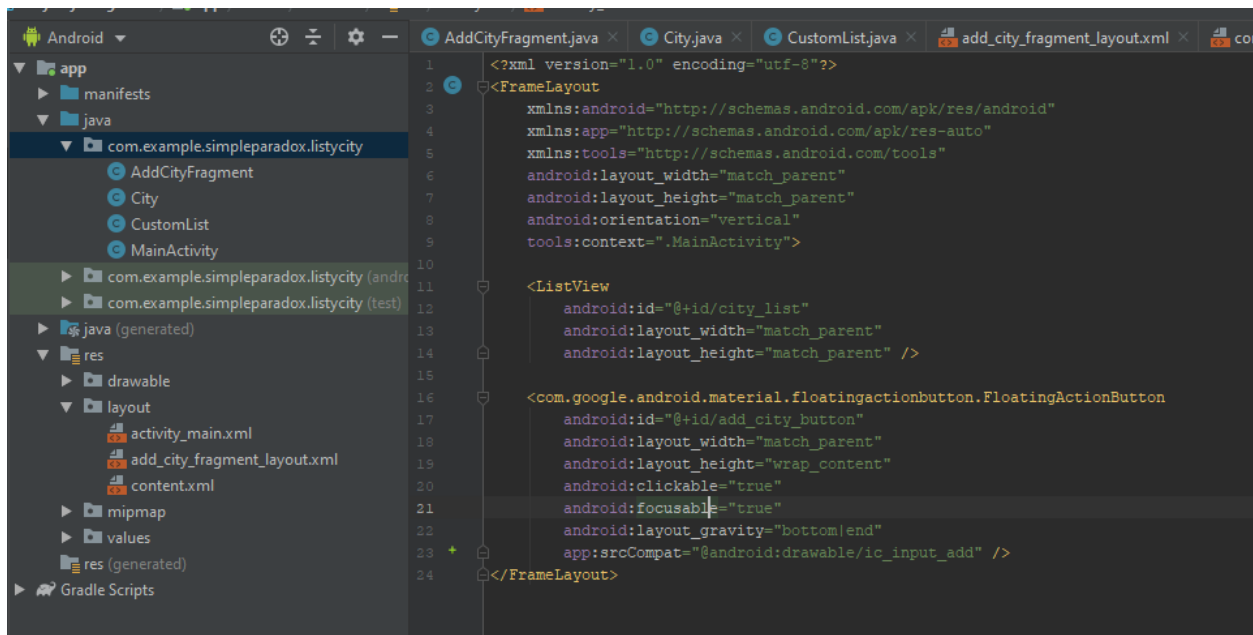


## Lab 3 Instructions – Fragment

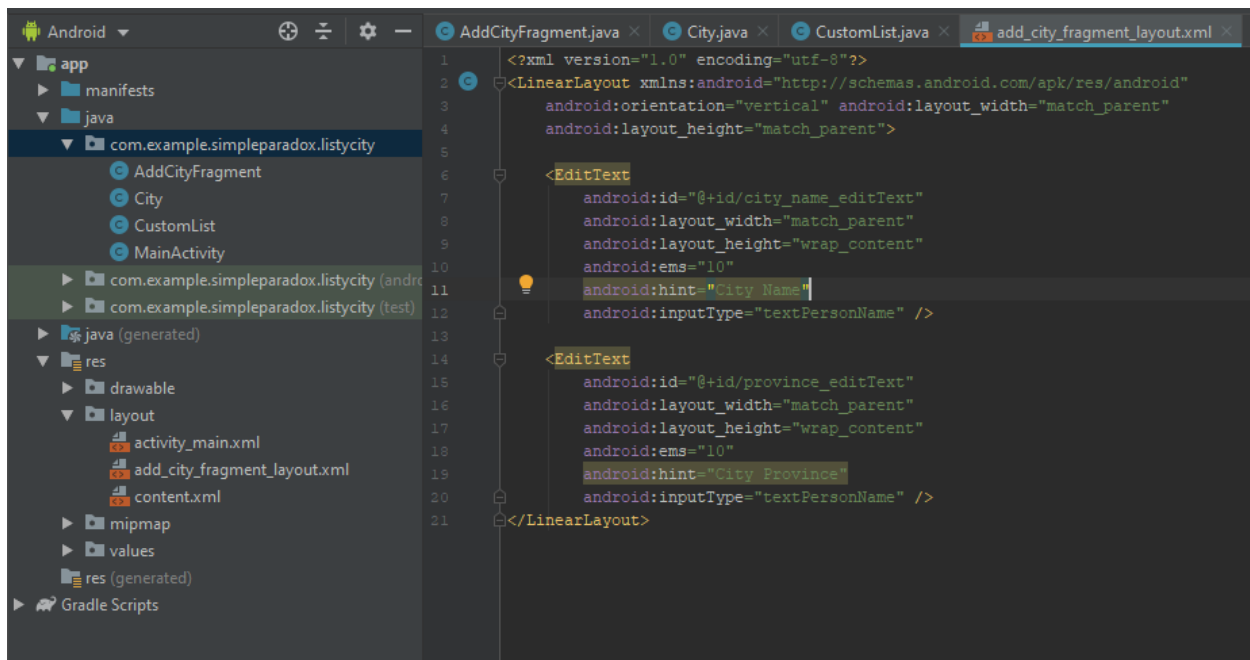
Take the ListyCity project and apply the changes described in “Lab 3 Instructions – CustomList” before proceeding with this demo.

Quick Summary: this demo shows one possible method of adding a fragment to your MainActivity which will be used to receive user input for the addition of new cities.

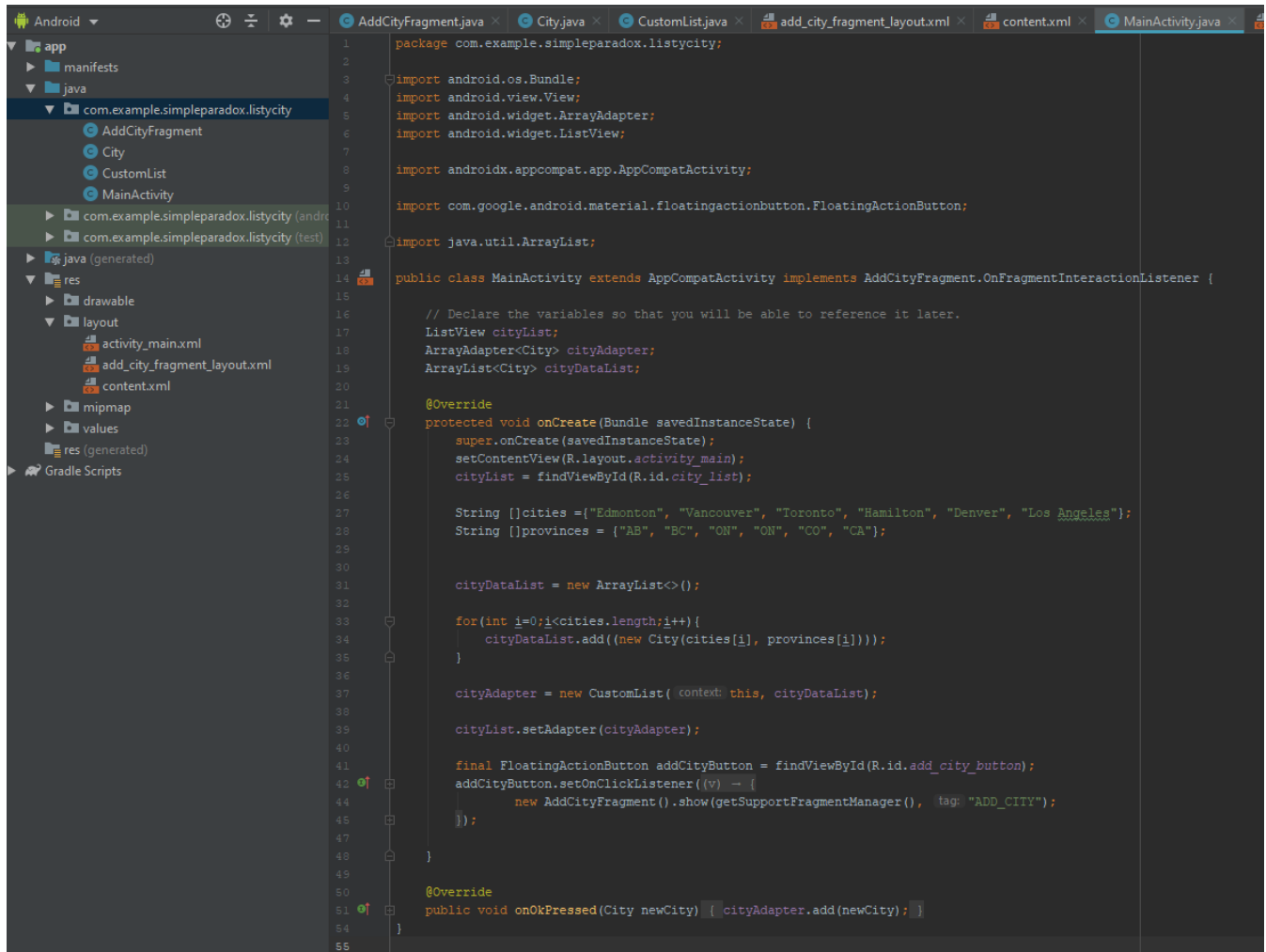
1. Open the Project in Android Studio and then navigate to the **activity\_main.xml** file. Edit the xml so that it has a **FloatingActionButton** as depicted in the screenshot below. The **srcCompat** attribute specifies the appearance of the button thumbnail. Change the **activity\_main.xml** to use a **FrameLayout** instead of a **LinearLayout**. Also note the additional **layout\_gravity** attributes of the **FloatingActionButton** (to position it in the bottom right corner).



2. Create a layout file for the fragment – we'll call it add\_city\_fragment\_layout.xml then add two editText fields (1 for the city name and 1 for the province).



3. Now we will create an **onClickListener()** for the **FloatingActionButton** as well as override the **onOkPressed()** method in the **MainActivity** class. The **addCityButton** listener will call the **FragmentManager** to display the **ADD CITY** Fragment while the **onOkPressed()** method will take a **City** object as an argument and then add it to the **listview** by calling **cityAdapter.add(newCity)**



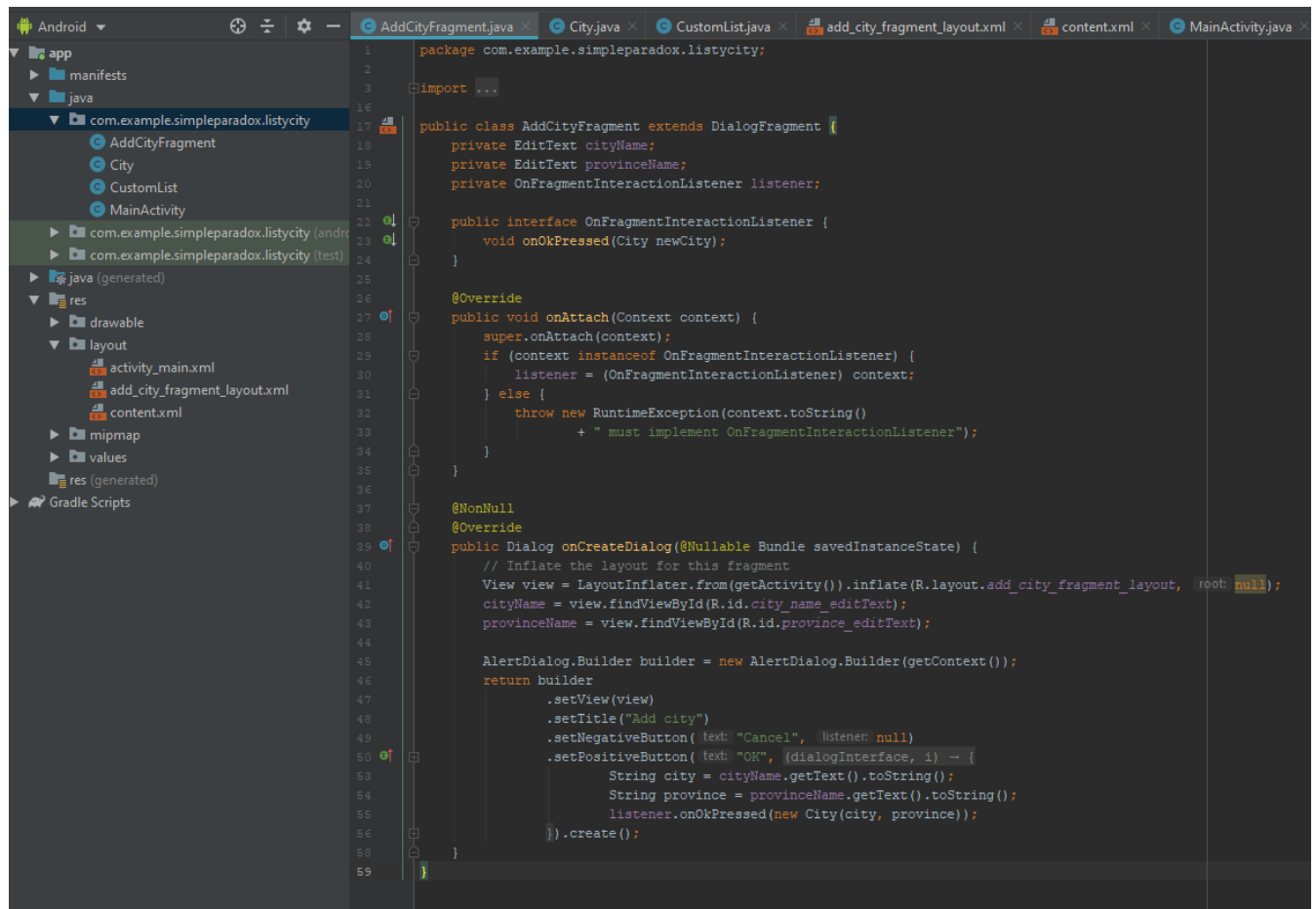
```
1 package com.example.simpleparadox.listcity;
2
3 import android.os.Bundle;
4 import android.view.View;
5 import android.widget.ArrayAdapter;
6 import android.widget.ListView;
7
8 import androidx.appcompat.app.AppCompatActivity;
9
10 import com.google.android.material.floatingactionbutton.FloatingActionButton;
11
12 import java.util.ArrayList;
13
14 public class MainActivity extends AppCompatActivity implements AddCityFragment.OnFragmentInteractionListener {
15
16     // Declare the variables so that you will be able to reference it later.
17     ListView cityList;
18     ArrayAdapter<City> cityAdapter;
19     ArrayList<City> cityDataList;
20
21     @Override
22     protected void onCreate(Bundle savedInstanceState) {
23         super.onCreate(savedInstanceState);
24         setContentView(R.layout.activity_main);
25         cityList = findViewById(R.id.city_list);
26
27         String []cities = {"Edmonton", "Vancouver", "Toronto", "Hamilton", "Denver", "Los Angeles"};
28         String []provinces = {"AB", "BC", "ON", "ON", "CO", "CA"};
29
30
31         cityDataList = new ArrayList<>();
32
33         for(int i=0;i<cities.length;i++){
34             cityDataList.add(new City(cities[i], provinces[i]));
35         }
36
37         cityAdapter = new CustomList( context this, cityDataList);
38
39         cityList.setAdapter(cityAdapter);
40
41         final FloatingActionButton addCityButton = findViewById(R.id.add_city_button);
42         addCityButton.setOnClickListener(w -> {
43             new AddCityFragment().show(getSupportFragmentManager(), tag "ADD_CITY");
44         });
45
46     }
47
48     @Override
49     public void onOkPressed(City newCity) { cityAdapter.add(newCity); }
50
51 }
```

4. Finally, we will create a new fragment object called **AddCityFragment** by extending the **DialogFragment** class. In the class, we will override the **onCreateDialog()** method where we will initialize our AddCityFragment's **editText** fields. Additionally, create the **OnFragmentInteractionListener** interface that will call **onOkPressed** (in **MainActivity**) and pass a new **City** Object as a parameter.

In the **onCreateDialog()** method we will do the following: After initializing the View and the EditText fields we will create a new **AlertDialog** object using its **builder** method.

Within the **builder** method we will set the view, set a Title and finally set the OK and Cancel buttons. Note that for the OK button, we will implement the **listener interface** allowing us to pass a new **City** object (created using the entered city name and province name) to the **onOkPressed()** method in **MainActivity**.

Note the boilerplate code for the **onAttach()** method.



```
1 package com.example.simpleparadox.listcity;
2
3 import ...
4
5
6
7
8
9
10
11
12
13
14
15
16
17 public class AddCityFragment extends DialogFragment {
18     private EditText cityName;
19     private EditText provinceName;
20     private OnFragmentInteractionListener listener;
21
22     public interface OnFragmentInteractionListener {
23         void onOkPressed(City newCity);
24     }
25
26     @Override
27     public void onAttach(Context context) {
28         super.onAttach(context);
29         if (context instanceof OnFragmentInteractionListener) {
30             listener = (OnFragmentInteractionListener) context;
31         } else {
32             throw new RuntimeException(context.toString()
33                 + " must implement OnFragmentInteractionListener");
34         }
35     }
36
37     @NonNull
38     @Override
39     public Dialog onCreateDialog(@Nullable Bundle savedInstanceState) {
40         // Inflate the layout for this fragment
41         View view = LayoutInflater.from(getActivity()).inflate(R.layout.add_city_fragment_layout, root: null);
42         cityName = view.findViewById(R.id.city_name_editText);
43         provinceName = view.findViewById(R.id.province_editText);
44
45         AlertDialog.Builder builder = new AlertDialog.Builder(getContext());
46         return builder
47             .setView(view)
48             .setTitle("Add city")
49             .setNegativeButton("Cancel", listener: null)
50             .setPositiveButton("OK", (dialogInterface, i) -> {
51                 String city = cityName.getText().toString();
52                 String province = provinceName.getText().toString();
53                 listener.onOkPressed(new City(city, province));
54             }).create();
55     }
56 }
57
58
59 }
```