

A thick blue vertical bar runs down the left side of the page. A blue arrow points to the right, overlapping the bar, with the date '6/28/2018' written inside it in white text.

6/28/2018

Android Project report

Supervised /Evaluated by: Zhangquan
Zhao

- 
- Several thin, curved lines in blue and grey originate from the bottom left and sweep upwards and to the right, creating a dynamic, abstract design element.
1. Günther Tresor Tchogmakwe Yango
9140214004

Table of Contents

Introduction.....	2
User Interface View	3
Flow Chart	5
UML Diagram.....	6
App Structure	7
Xml and main code Explanation and functionality	7

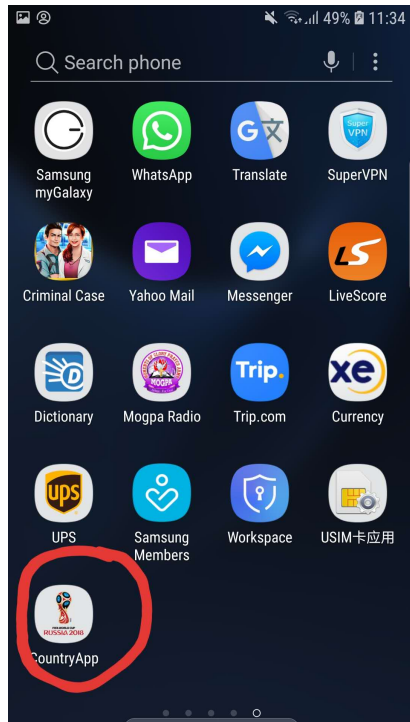
Introduction

The FIFA world Cup is a very popular event and this app gives you some information about countries and their brief History in relation to football, the FIFA world cup and how many appearances at the world cup.

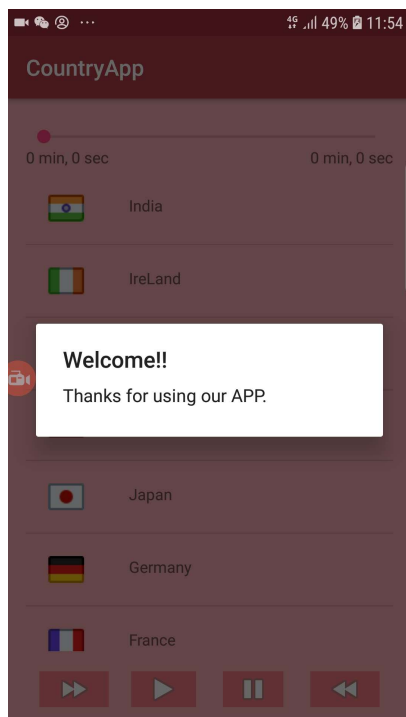
1. Designing and implement simple layouts for all activities based on the provided resources (Country Flag, Country Name, Country Information)
2. Implementation of different classes.
3. Handle audio resources (play, pause, forward and backward options available)
4. Using an array list to store Values/data.
5. Welcome pop-up window thanking the app users each time they access the MainActivity.

User Interface View

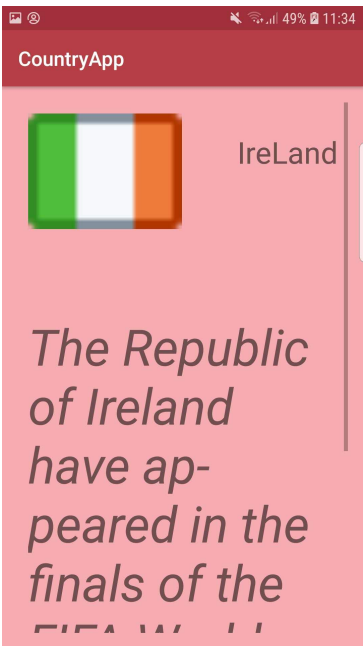
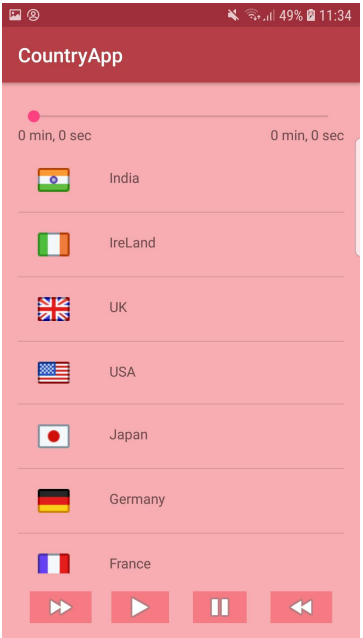
I. App icon(screenshot)



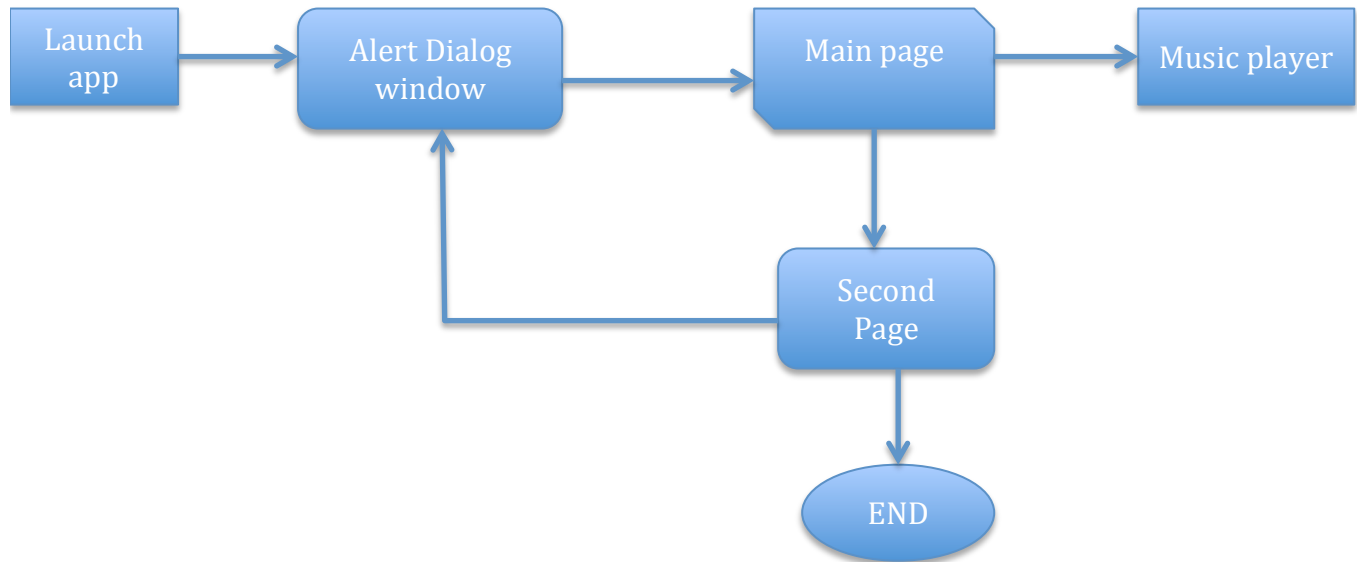
II. App UI (screenshot)



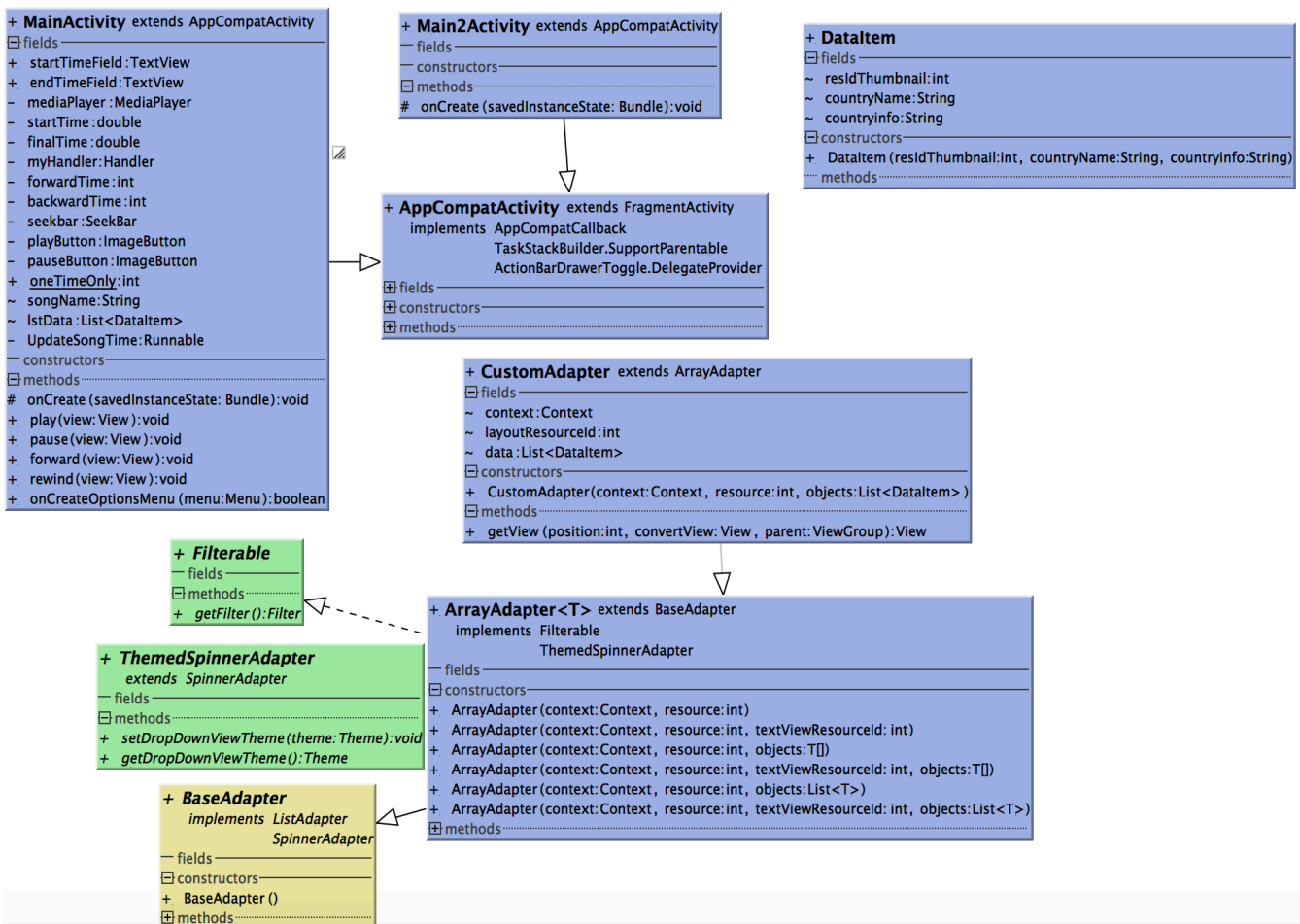
III. App UI for item clicked (screenshot)



Flow Chart



UML Diagram



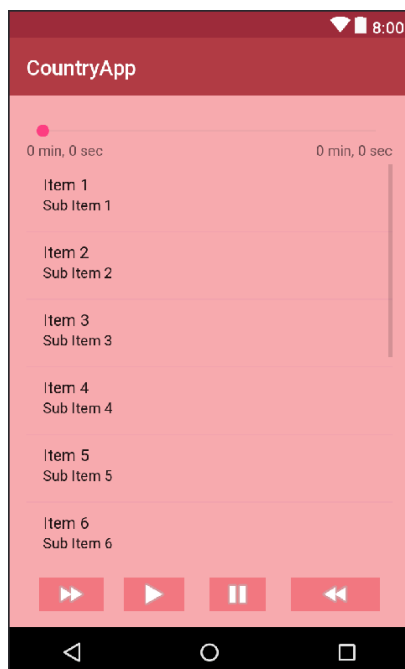
App Structure

Xml and main code Explanation and functionality

The app consist of 3 .xml files and 4 .java files.

The xml files consist of : **(screenshots included)**

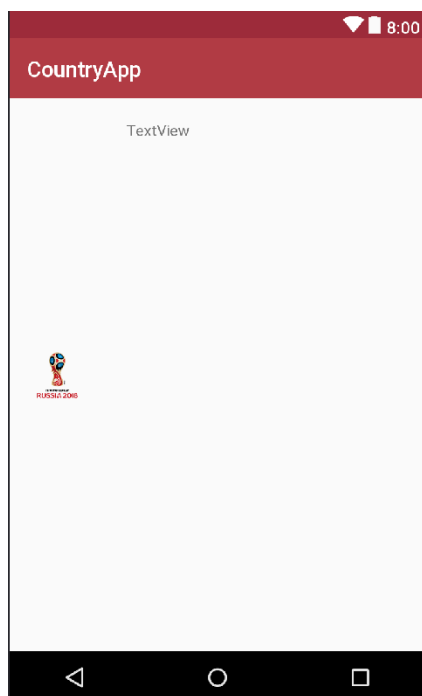
1. Activity_main.xml: It is made of the ImageButtons at the bottom to control the music, the SeekBar at the top to view the music progression, TextView to view the time progress in minutes and seconds.



2. Activity_main2.xml: Contains the scrollView for texts, the ImageView the display the respective country flag (Indian flag is set as default image), 2 TextView displaying the country Name and country information from top to bottom respectively.



3. Itemrow.xml: Here we set the structure of the customListView. ImageView(flag) on the left and TextView(country name) following the image towards the right.



The java files consist of:

1. MainActivity: Here you will find the ArrayList, Adapter, AlertDialog, and MediaPlayer.

```
lstData = new ArrayList<>();
listView.setAdapter (adapter);
AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity.this);
mediaPlayer = MediaPlayer.create(this, R.raw.fifasong2018);
```

The above syntaxes control the different actions.

2. Main2Activity: Here you will find the following codes which get the Values from the ArrayList which will be displayed in the activity_main2.xml

```
textView.setText(getIntent().getStringExtra("Country"));
imageView.setImageResource(getIntent().getIntExtra("Flag", R.drawable.india));
countryinfoView.setText(getIntent().getStringExtra("countryinfo"));
```

3. DataItem.java: This class holds the data for the customlistView. That is, the imageView (Flag; int resIdThumbnail), the TextViews(String countryName,String countryinfo) respectively.

```
public class DataItem {
    int resIdThumbnail;
    String countryName;
    String countryinfo;

    public DataItem(int resIdThumbnail, String countryName, String countryinfo) {
        this.resIdThumbnail = resIdThumbnail;
        this.countryName = countryName;
        this.countryinfo = countryinfo;
    }
}
```

NB: "String countryinfo" is being hold in the dataitem but only use by main2activity ,

```
countryinfoView.setText(getIntent().getStringExtra("countryinfo"));
```

get the data and displays it in activity_2main.

4. CustomAdapter.java: A customAdapter is needed to create a customLayout.

```
public class CustomAdapter extends ArrayAdapter<DataItem> {
```

It will basically find the corresponding resource id in the ArrayList and get its corresponding strings. In other words; getting the flag and the country name.

The Dataholder, hold references for all the views on the ListView.

```
static class DataHolder
```

```
{
    ImageView ivFlag;
    TextView tvCountryName;
    //TextView tvCountryinfo;
}
```

The getView method will be called for every role/dataitem in the ListView.

```
public View getView(int position, View convertView, ViewGroup parent) {
```

In this case its been called 15 times.