Code Day Assignment

Oliver Njeru-663565

School of Sciences and Technology, United States International University-Africa

APT3060A: Mobile Programming

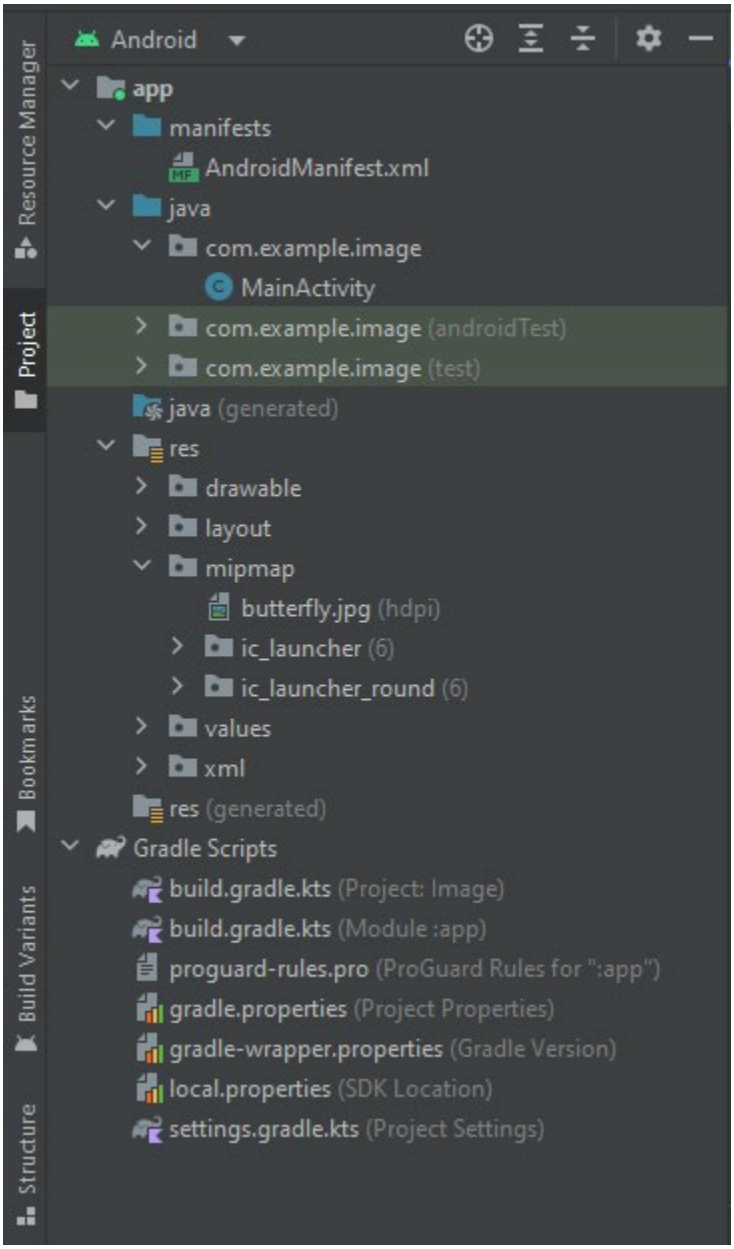
Dr. Lawrence Nderu

November 05, 2023

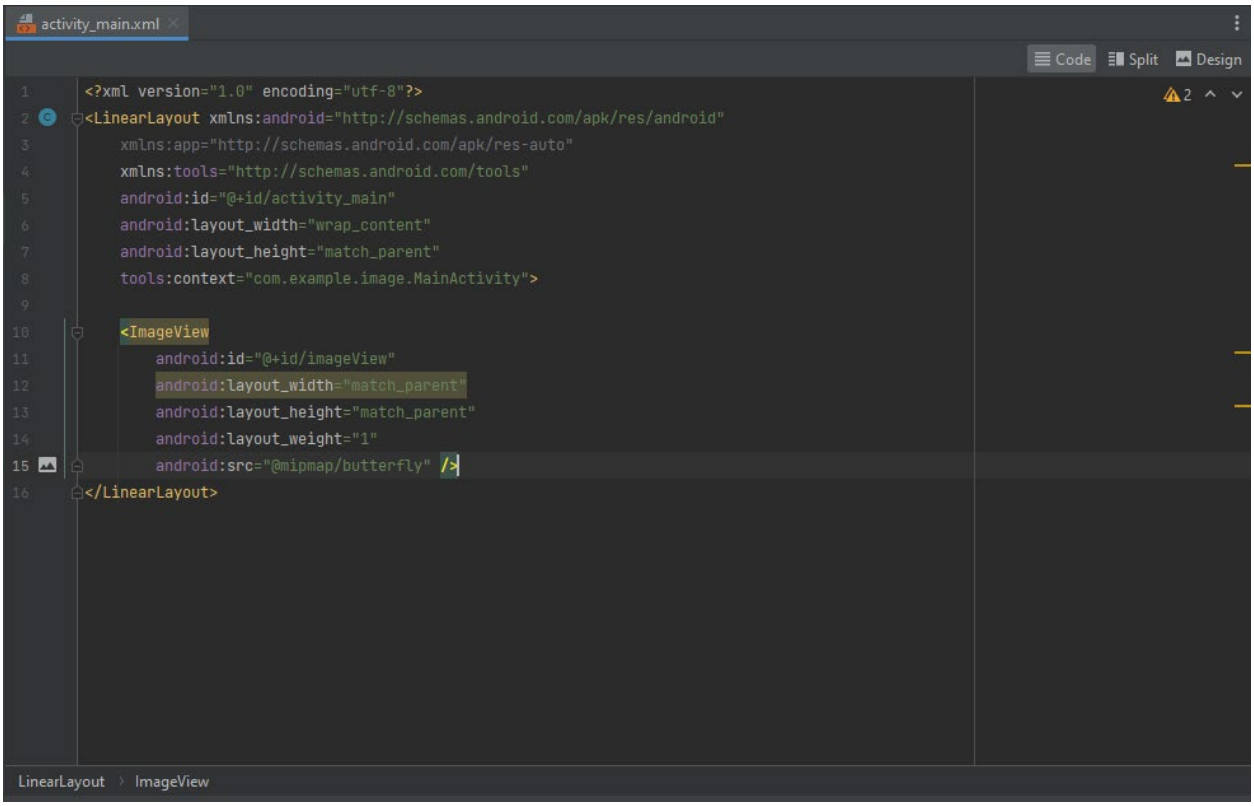
Code Day Assignment.

Image Project

In this project, our goal is to create an application that displays an image of a butterfly. The first thing I did was to add the butterfly image in the mipmap folder as shown below.



The next thing I did was navigate to the activity_main.xml file and add a LinearLayout tag as well as an ImageView tag that is going to locate the image and display it when the application is run. The code added is shown below.



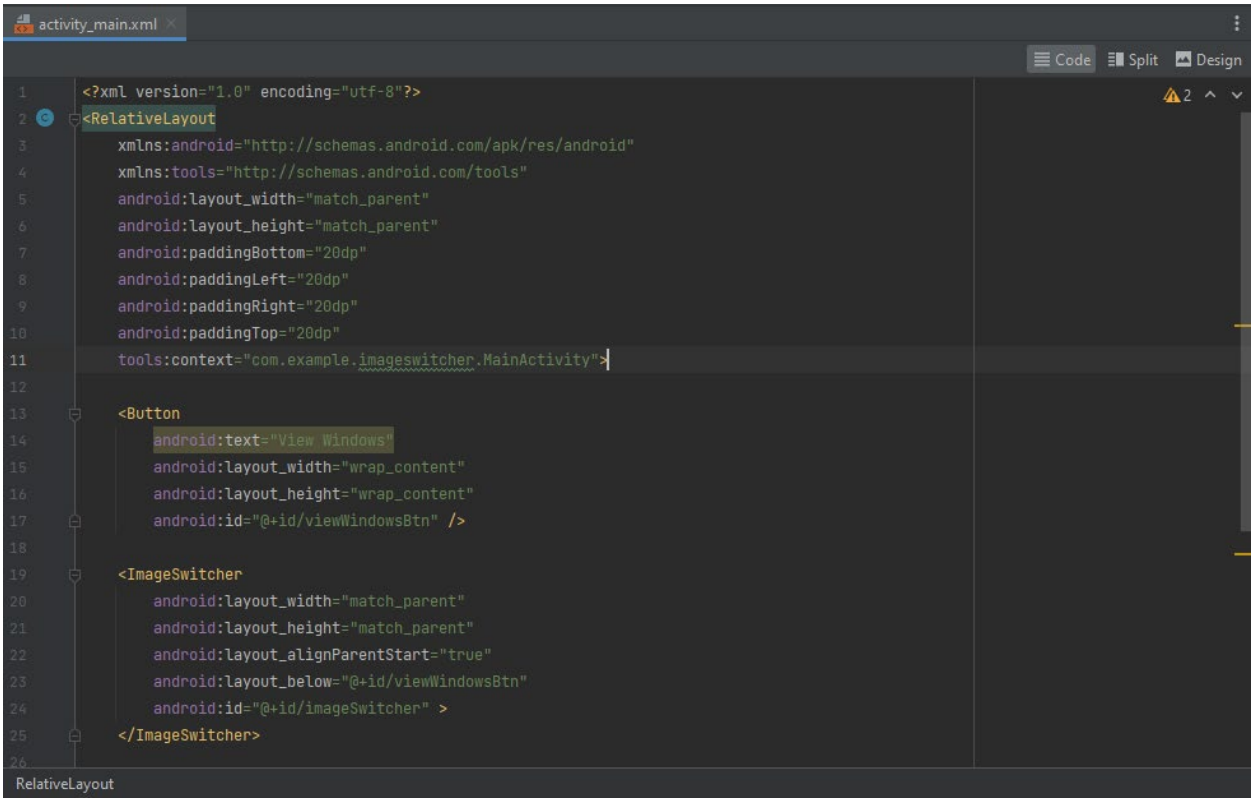
The next step is to run the application. What the application displays is as shown below.

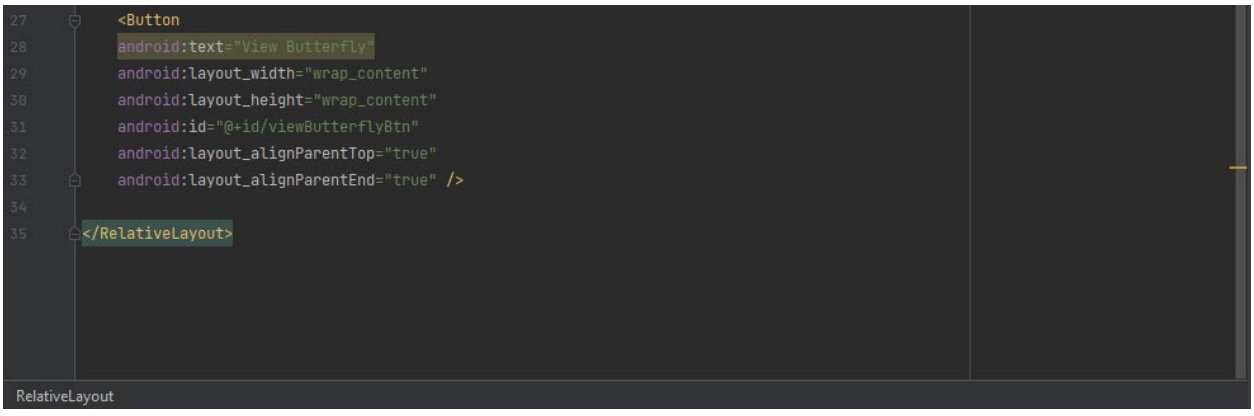
This marks the completion of this project’s goal.

ImageSwitcher

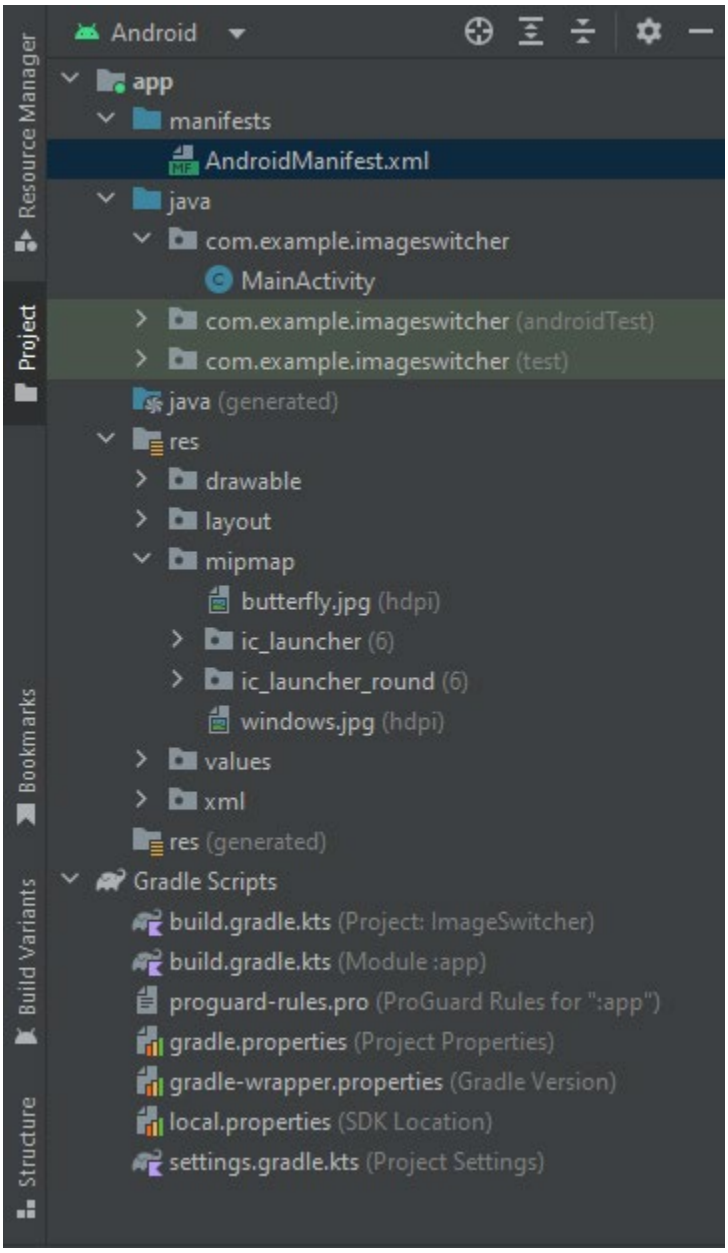
This project builds off on top of the previous project. It displays more than one image with a transition in between them.

In the activity_main.xml file, I added a RelativeLayout tag, two Button tags and an ImageSwitcher tag as shown below.

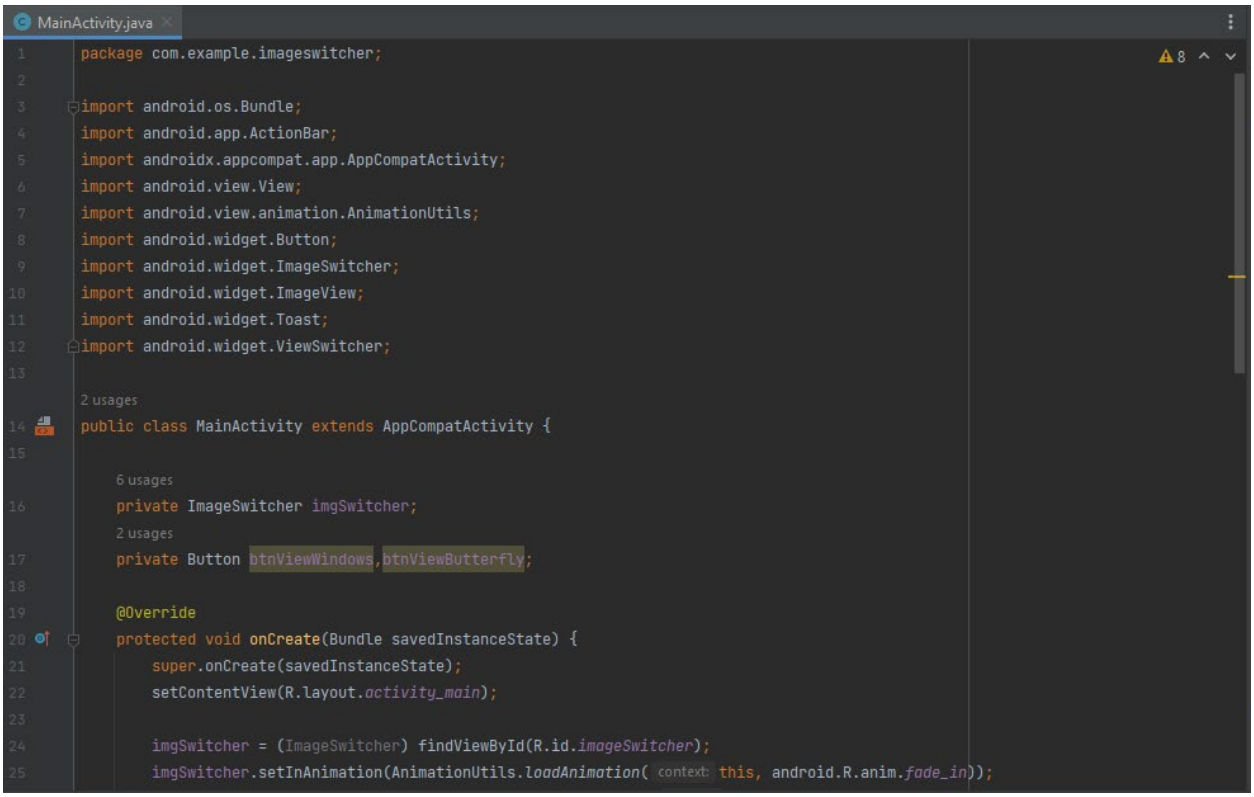




I then added two images in the mipmap folder that I am going to use in this project. The images are shown below.



In MainActivity.java file, I added the following code. Three private variables with an ImageSwitcher type and Button type. Initialize imgSwitcher variable by finding ImageSwitcher widget using its id and casting the result of findViewById to the ImageSwitcher type. I then set the animation for ImageSwitcher transitions using a method to load fade-in and fade-out animations from the android system resources. I initialized the two buttons using their ids. I added a method that makes the image view. I added onClickListener methods for each of the two buttons to set the value for the toast message and view switch the correct image accordingly using the imgSwitcher variable.



MainActivity.java

```
26          imgSwitcher.setOutAnimation(AnimationUtils.loadAnimation( context: this, android.R.anim.fade_out));
27          btnViewWindows = (Button) findViewById(R.id.viewWindowsBtn);
28          btnViewButterfly = (Button) findViewById(R.id.viewButterflyBtn);
29
30          imgSwitcher.setFactory(new ViewSwitcher.ViewFactory() {
                1 usage
31              @Override
32              public View makeView() {
33                  ImageView myView = new ImageView(getApplicationContext());
34                  myView.setScaleType(ImageView.ScaleType.FIT_CENTER);
35                  myView.setLayoutParams(new ImageSwitcher.LayoutParams(ActionBar.LayoutParams.WRAP_CONTENT,
36                          ActionBar.LayoutParams.WRAP_CONTENT));
37                  return myView;
38              }
39          });
40
41          btnViewWindows.setOnClickListener(new View.OnClickListener() {
42              @Override
43              public void onClick(View v) {
44              Toast.makeText(getApplicationContext()
45                      , text: "View Windows",Toast.LENGTH_LONG).show();
46              imgSwitcher.setImageResource(R.mipmap.windows);
47              }
48          });
49          btnViewButterfly.setOnClickListener(new View.OnClickListener() {
50              @Override
51              public void onClick(View v) {
52                  Toast.makeText(getApplicationContext(),  text: "View Butterfly" ,Toast.LENGTH_LONG).show();
```

```
53                      imgSwitcher.setImageResource(R.mipmap.butterfly);
54              }
55          });
56      }
57  }
58
```

When the application is run, the following is seen.

On run:

On click of View Windows Button:

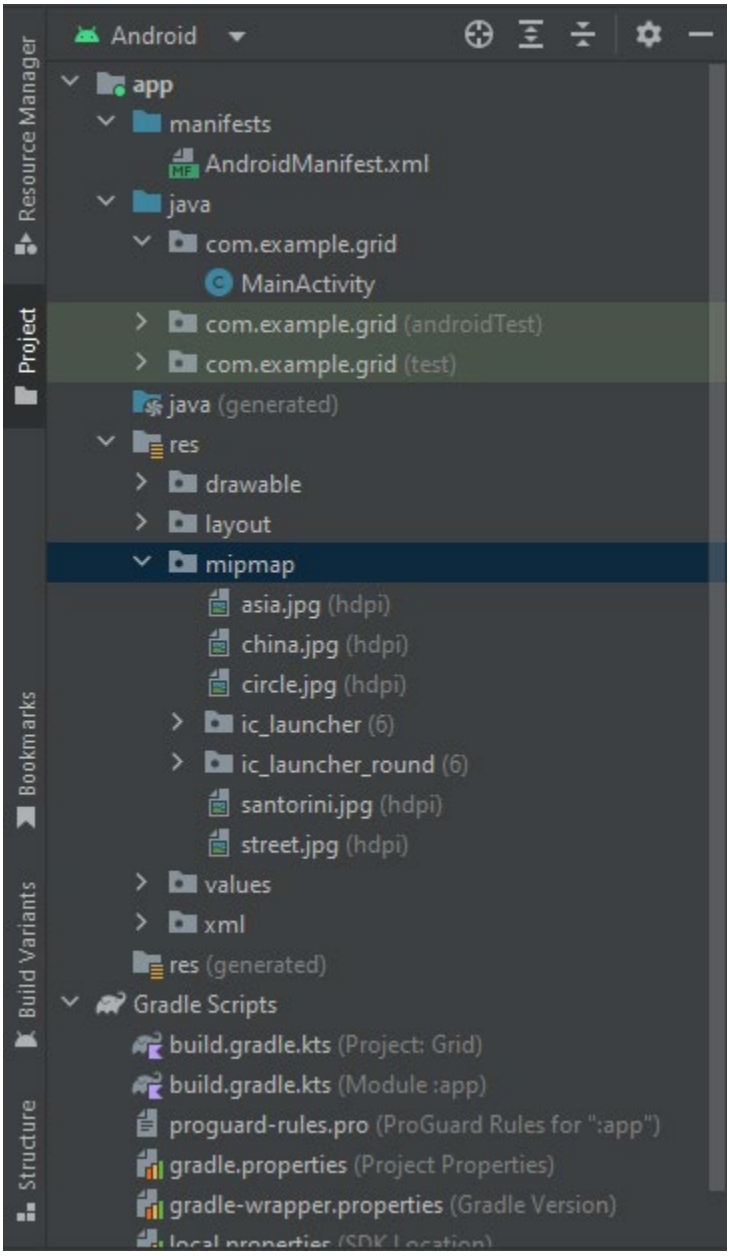
On click of View Butterfly button:

This marks the completion of the objective of this project.

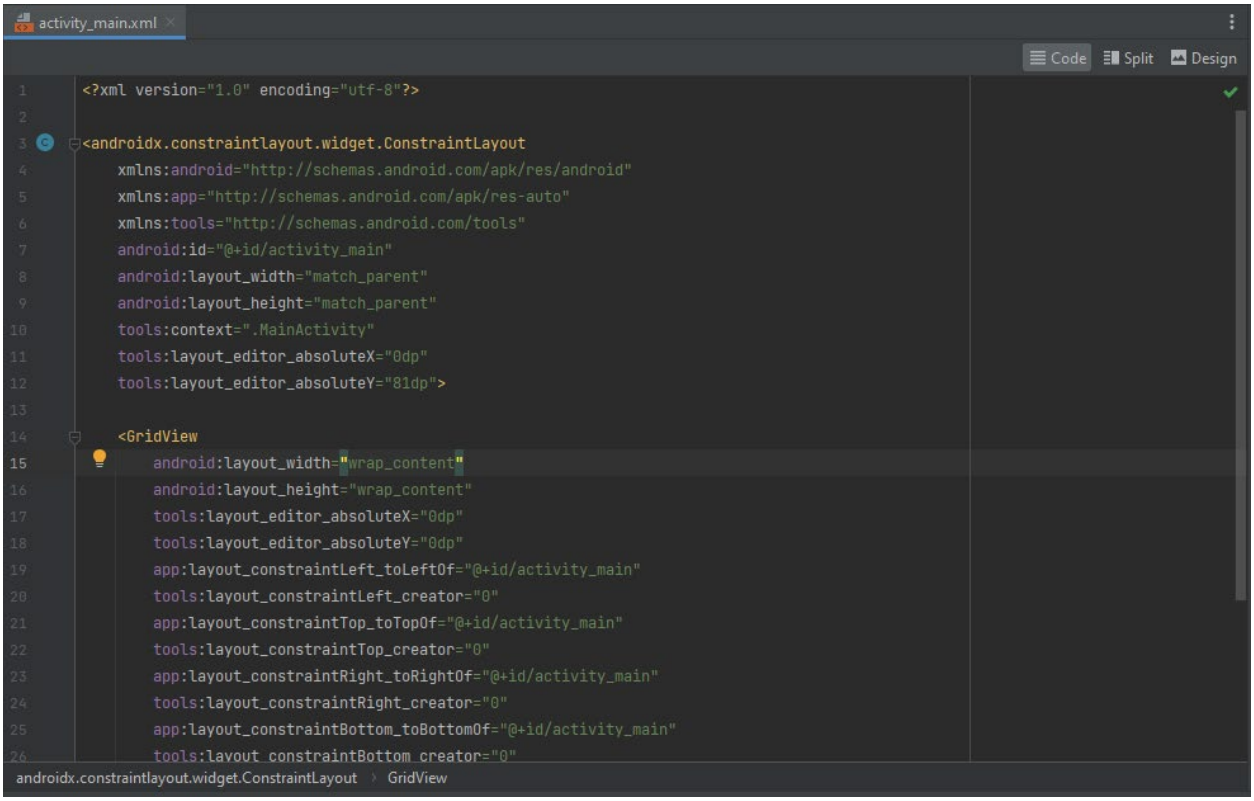
Grid

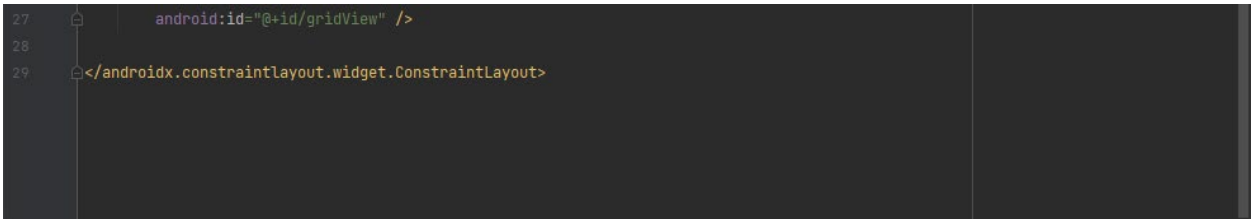
The objective of this project is to display images in a grid view and show a toast message of the image selected and its position.

In the mipmap folder, I added five images, asia, china, circle, santorini and street that are going to be used in this project as shown below.

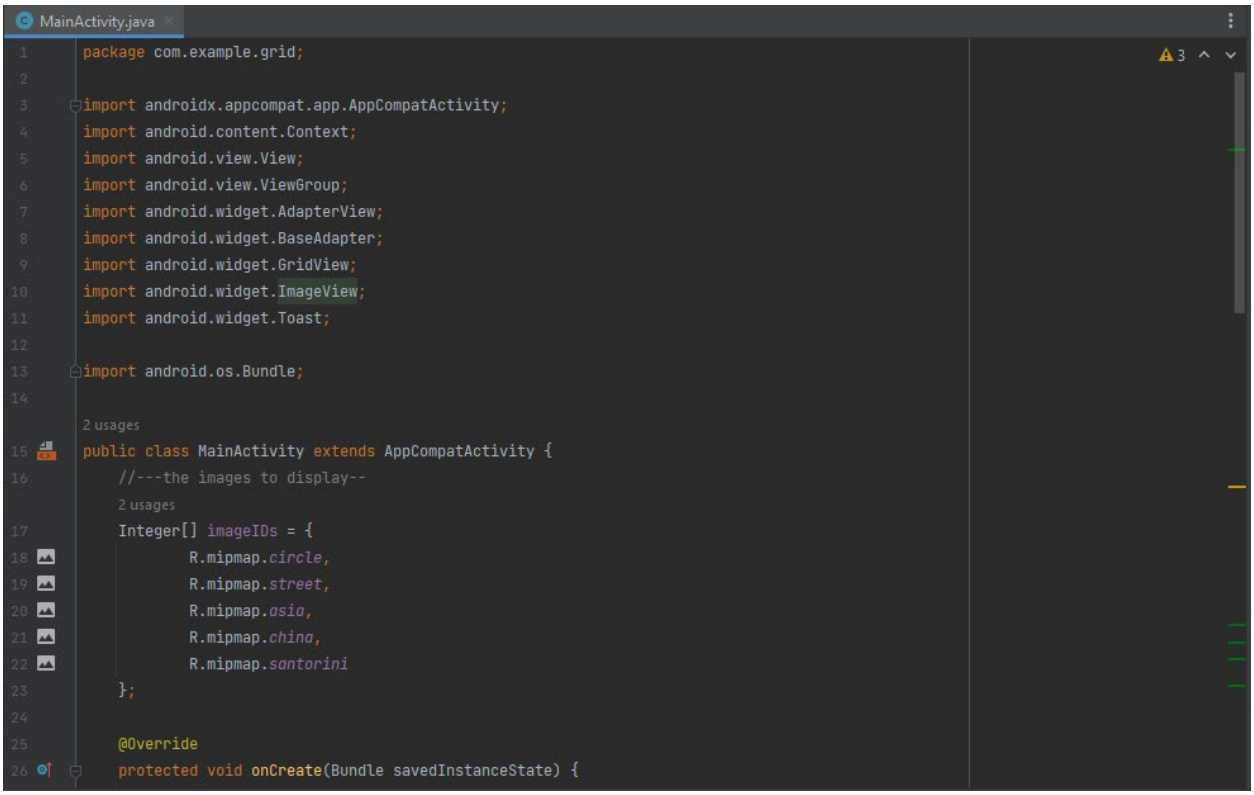


In the activity_main.xml file, I added a constraint layout and a GridView tag as shown below.





In the MainActivity.java file, I created an array that is going to store the images located and referenced in the mipmap folder. In the onCreate method, I created a gridView variable of type GridView and initialized it to be the gridView widget in the activity_main.xml file by finding its id. I then set an onClickListener to track the position of the image and set the toast message. I created an if statement to check on the convertView value and given it is null, to set the imageView layout, scale type and padding. Else, assign the variable imageView to be convertView casted to type ImageView. Then I set the imageView to the image depending on the position and returned the imageView. The code is as shown below.



MainActivity.java

```
27          super.onCreate(savedInstanceState);
28          setContentView(R.layout.activity_main);
29
30          GridView gridView = (GridView) findViewById(R.id.gridView);
31          gridView.setAdapter(new ImageAdapter( c: this));
32          gridView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
33              public void onItemClick(AdapterView parent,
34                                      View v,
35                                      int position,
36                                      long id) {
37                  Toast.makeText(getBaseContext(),
38                          text: "pic" + (position + 1) + " selected",
39                          Toast.LENGTH_SHORT).show();
40              }
41          });
42      }
43
        1 usage
44      public class ImageAdapter extends BaseAdapter {
            2 usages
45          private Context context;
46
            1 usage
47          public ImageAdapter(Context c) { context = c; }
50
51          //---returns the number of images--
52          public int getCount() { return imageIDs.length; }
55
```

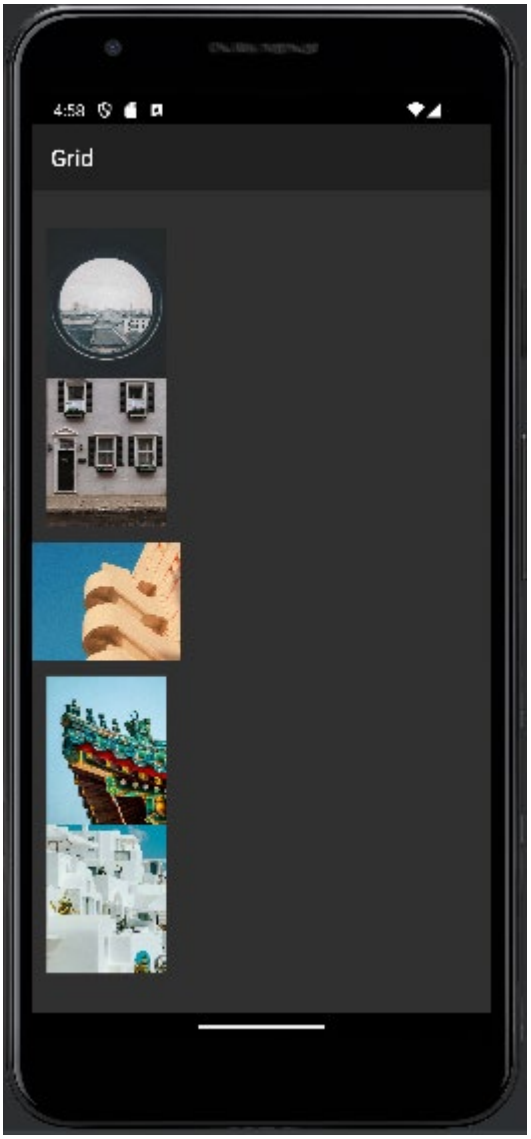
```
56          //---returns the item--
57          public Object getItem(int position) { return position; }
60
61          //---returns the ID of an item--
62          public long getItemId(int position) { return position; }
65
66          //---returns an ImageView view--
67          public View getView(int position, View convertView, ViewGroup parent) {
68              ImageView imageView;
69              if (convertView == null) {
70                  imageView = new ImageView(context);
71                  imageView.setLayoutParams(new GridView.LayoutParams( w: 350,  h: 350));
72                  imageView.setScaleType(ImageView.ScaleType.CENTER_CROP);
73                  imageView.setPadding( left: 35,  top: 35,  right: 35,  bottom: 35);
74              } else {
75                  imageView = (ImageView) convertView;
76              }
77              imageView.setImageResource(imageIDs[position]);
78              return imageView;
79          }
80      }
81  }
```

com.example.grid.MainActivity

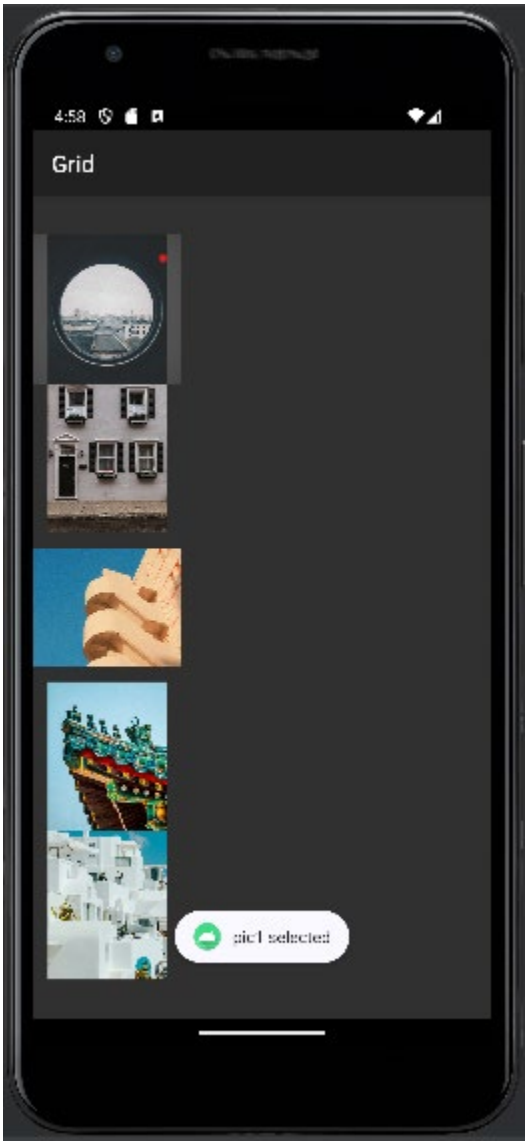
Integer[] imageIDs
= {...

Grid.app.main

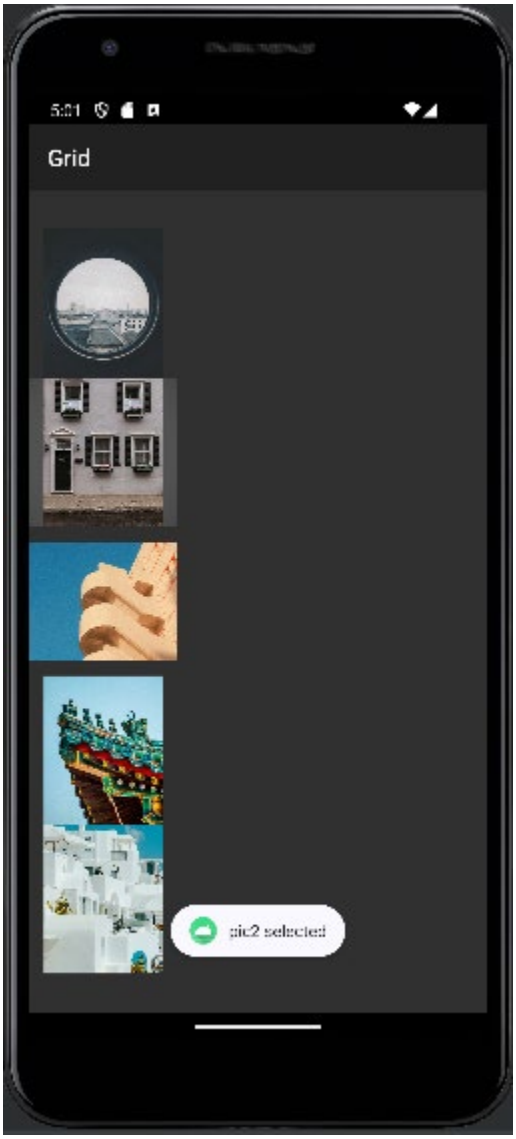
When the application is run, the following screen is displayed:



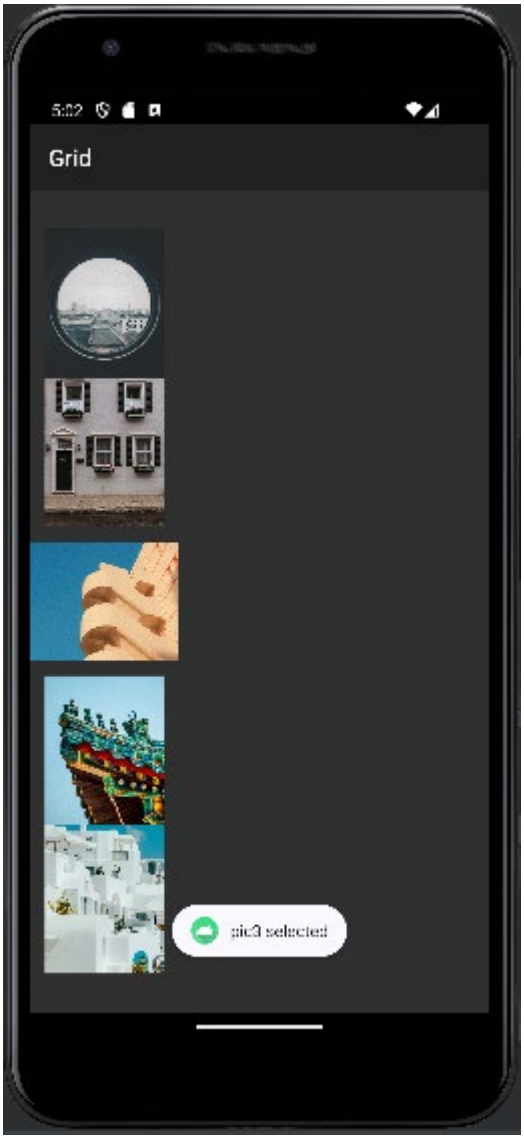
When I tap on the first image, the toast message is as show:



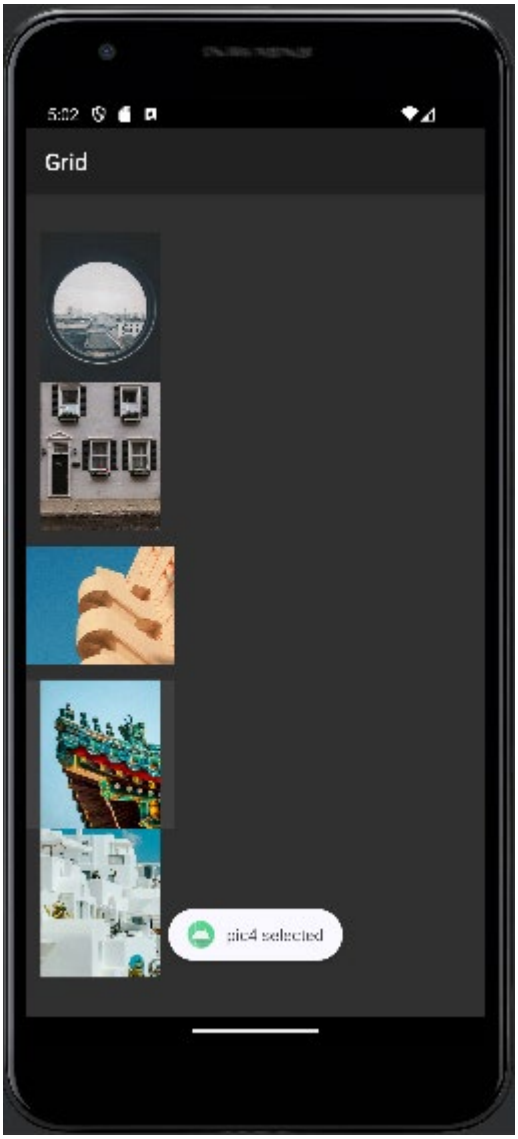
When I tap on the second image, the toast message is as shown:



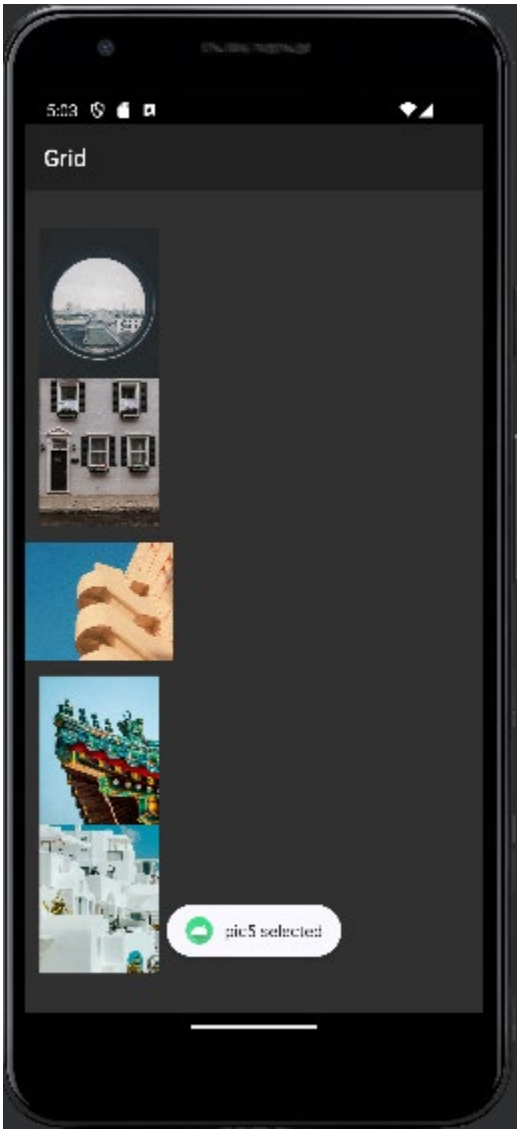
When I tap on the third image, the toast message is as shown:



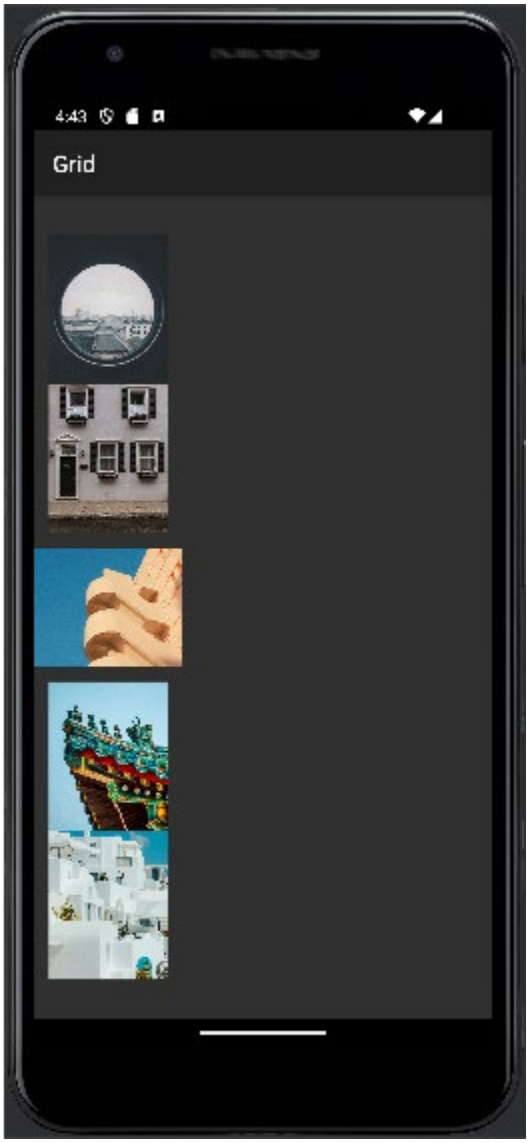
When I tap on the fourth image, the toast message is as shown:



When I tap on the fifth image, the toast message is as shown:



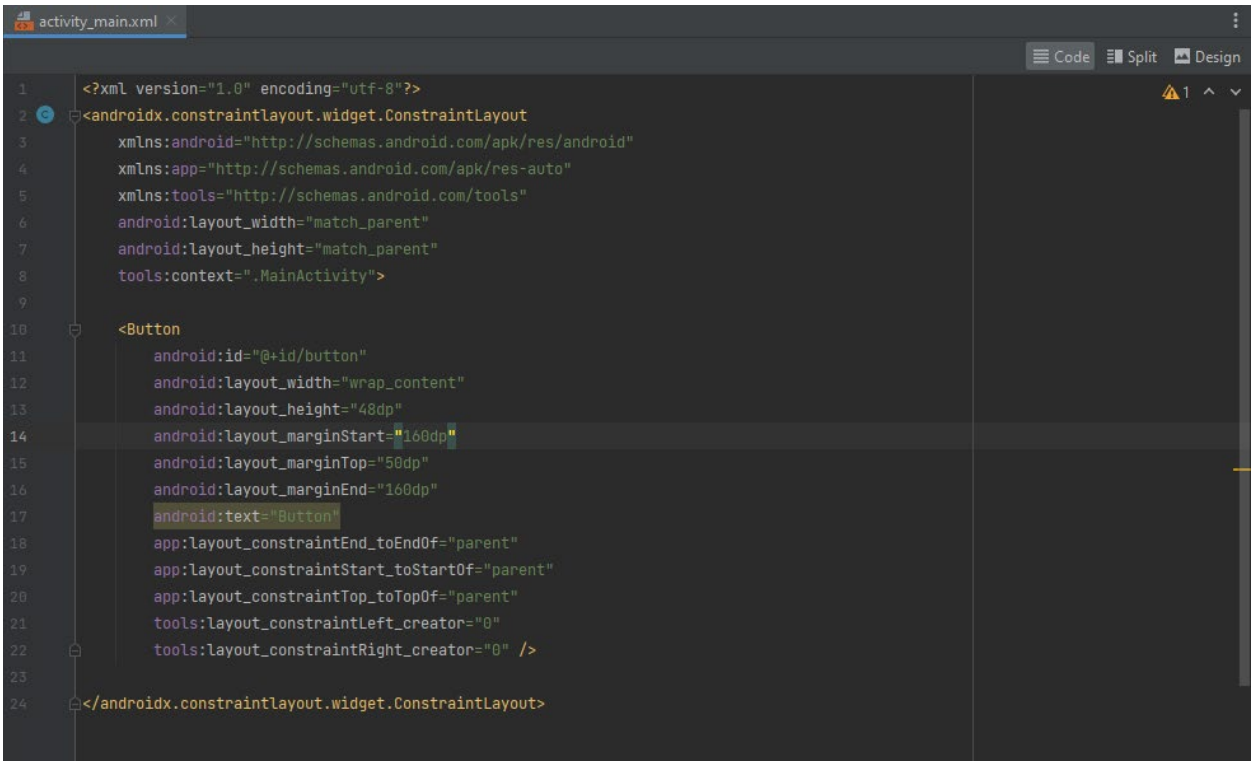
This shows that we have achieved the project's objective.



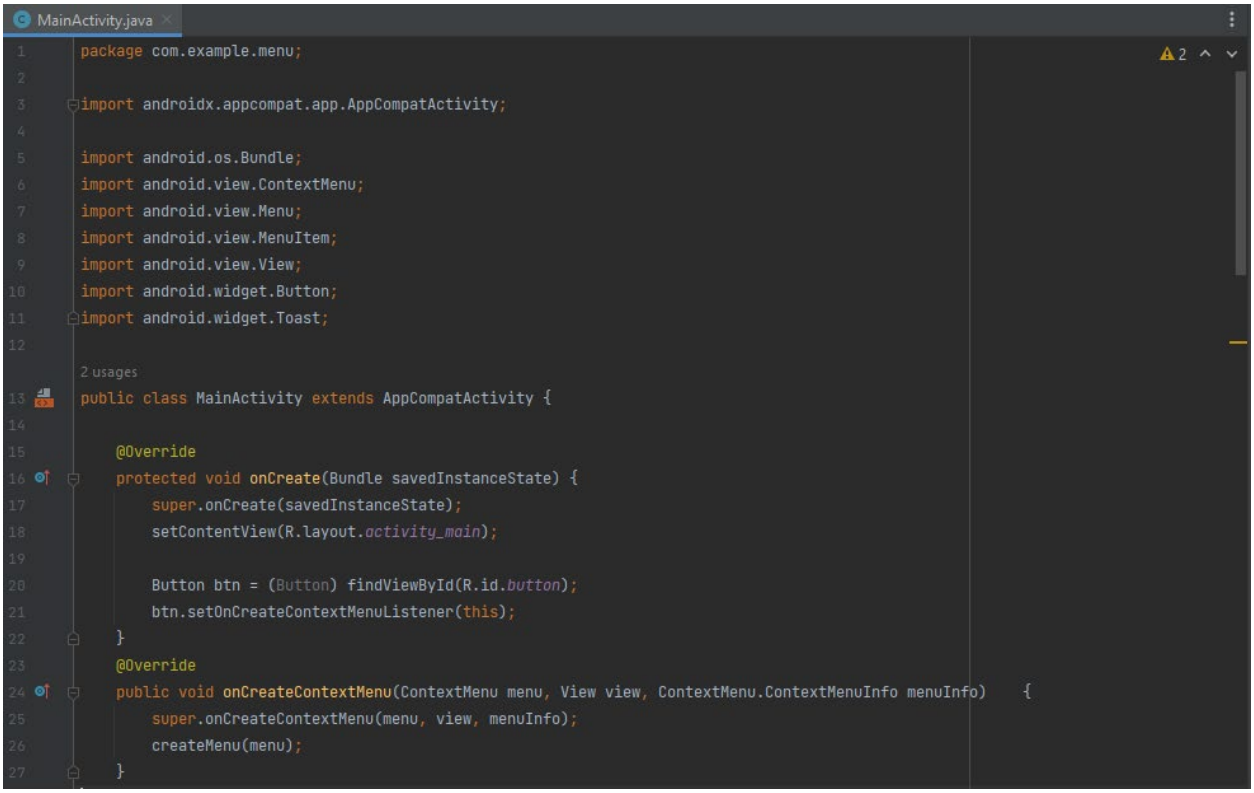
Menu

The objective of this project is to create a menu list and have a button display a menu list when a user presses and holds the button. The menu options, when clicked, should display a toast message indicating that the user clicked on an option with its appropriate number.

In activity_main.xml file, I added a constraint layout and a button widget with id button as show below.



In MainActivity.java file, below is the code. I initialized a button variable to be the id of the widget button. I created a void method that creates a contextMenu. I created a Boolean method that creates an options menu and return a true value. I created a Boolean method that returns the menu choice of the item selected. I created a void method that creates a menu with 7 items having the first for with alphabet shortcuts a-d. I created a Boolean method with a switch statement that checks on what was clicked and sets the toast message to display that the user clicked on an option with its number.



MainActivity.java

```java
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    super.onCreateOptionsMenu(menu);
    createMenu(menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) { return MenuChoice(item); }

2 usages
private void createMenu(Menu menu) {
    MenuItem mnu1 = menu.add( groupId: 0,  itemId: 0,  order: 0,  title: "Item 1"); {
        mnu1.setAlphabeticShortcut('a');
    }
    MenuItem mnu2 = menu.add( groupId: 0,  itemId: 1,  order: 1,  title: "Item 2"); {
        mnu2.setAlphabeticShortcut('b');
    }
    MenuItem mnu3 = menu.add( groupId: 0,  itemId: 2,  order: 2,  title: "Item 3"); {
        mnu3.setAlphabeticShortcut('c');
    }
    MenuItem mnu4 = menu.add( groupId: 0,  itemId: 3,  order: 3,  title: "Item 4"); {
        mnu4.setAlphabeticShortcut('d');
    }
    menu.add( groupId: 0,  itemId: 4,  order: 4,  title: "Item 5");
    menu.add( groupId: 0,  itemId: 5,  order: 5,  title: "Item 6");
    menu.add( groupId: 0,  itemId: 6,  order: 6,  title: "Item 7");
```

MainActivity.java

```java
}
1 usage
private boolean MenuChoice(MenuItem item) {
    switch (item.getItemId()) {
        case 0:
            Toast.makeText( context: this,  text: "You clicked on Item 1",
                Toast.LENGTH_LONG).show();
            return true;
        case 1:
            Toast.makeText( context: this,  text: "You clicked on Item 2",
                Toast.LENGTH_LONG).show();
            return true;
        case 2:
            Toast.makeText( context: this,  text: "You clicked on Item 3",
                 Toast.LENGTH_LONG).show();
            return true;
        case 3:
            Toast.makeText( context: this,  text: "You clicked on Item 4",
                 Toast.LENGTH_LONG).show();
            return true;
        case 4:
            Toast.makeText( context: this,  text: "You clicked on Item 5",
                  Toast.LENGTH_LONG).show();
            return true;
        case 5:
            Toast.makeText( context: this,  text: "You clicked on Item 6",
                   Toast.LENGTH_LONG).show();
            return true;
```

```
            case 6:
                Toast.makeText( context: this, text: "You clicked on Item 7",
                        Toast.LENGTH_LONG).show();
                return true;
    }
    return false;
    }
}

```

When the app is run, the following is the screen:

When the three dots at the top right of the screen are tapped, the following is displayed:

When I tap on Item1, the following toast message is displayed:

When I tap on Item2, the following toast message is displayed:

When I tap on Item3, the following toast message is displayed:

When I tap on Item4, the following toast message is displayed:

When I tap on Item5, the following toast message is displayed:

When I tap on Item6, the following toast message is displayed:

When I tap on Item7, the following toast message is displayed:

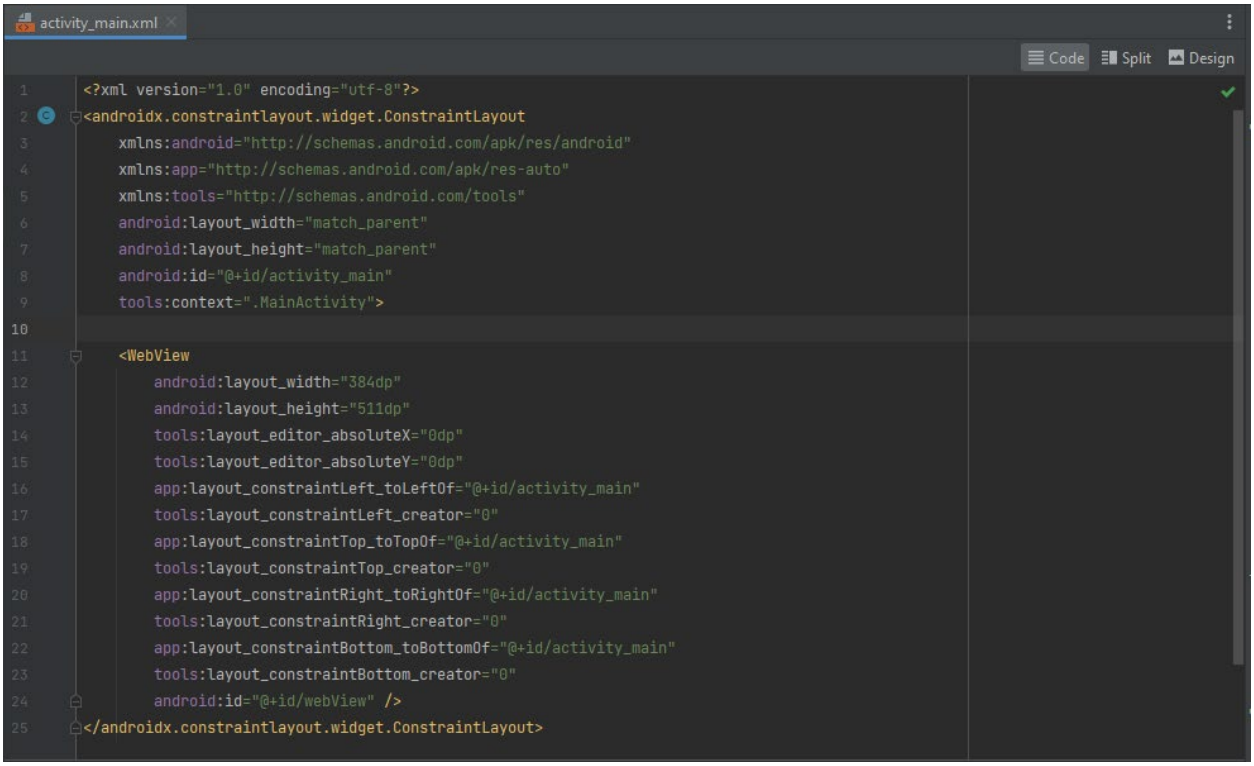
When I tap and hold the button at the center of the screen, I see the following:

This shows that the objective of the project has been achieved.

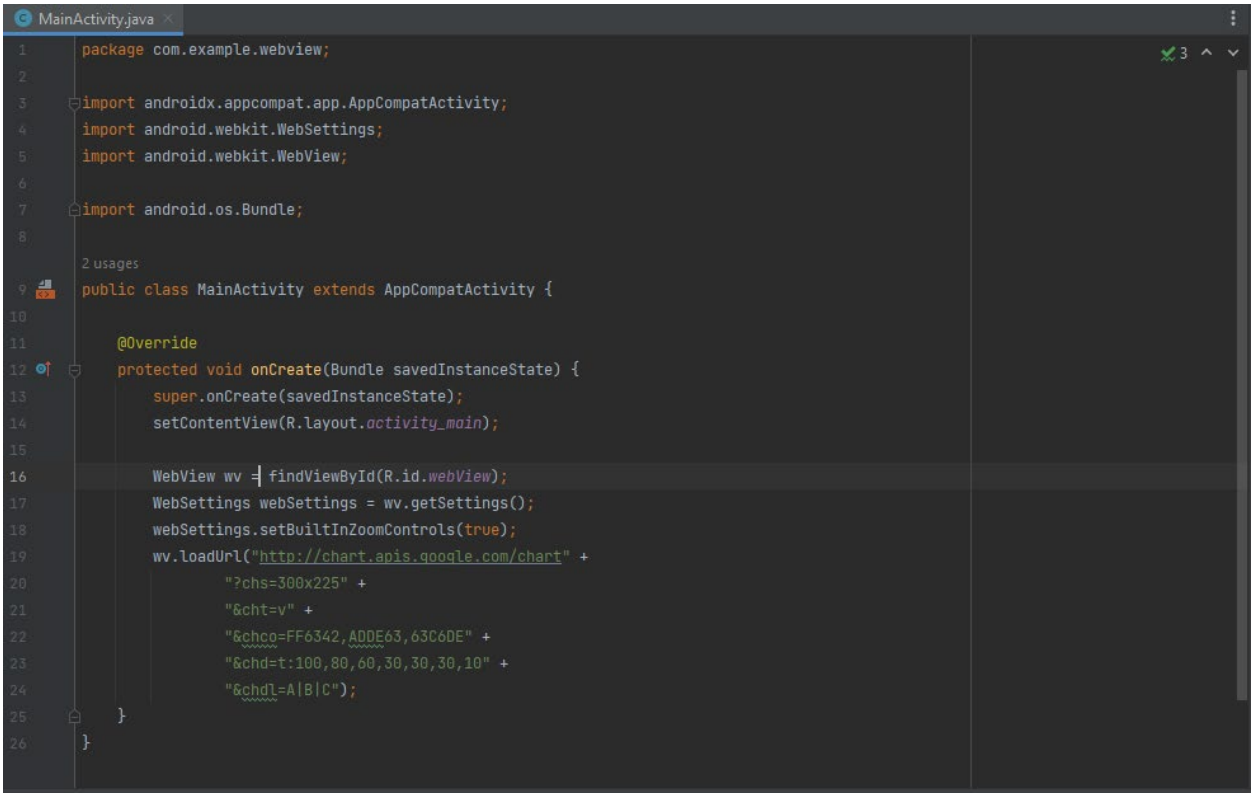
WebView

The objective of this project is to create a webview that shows the content of a specified url.

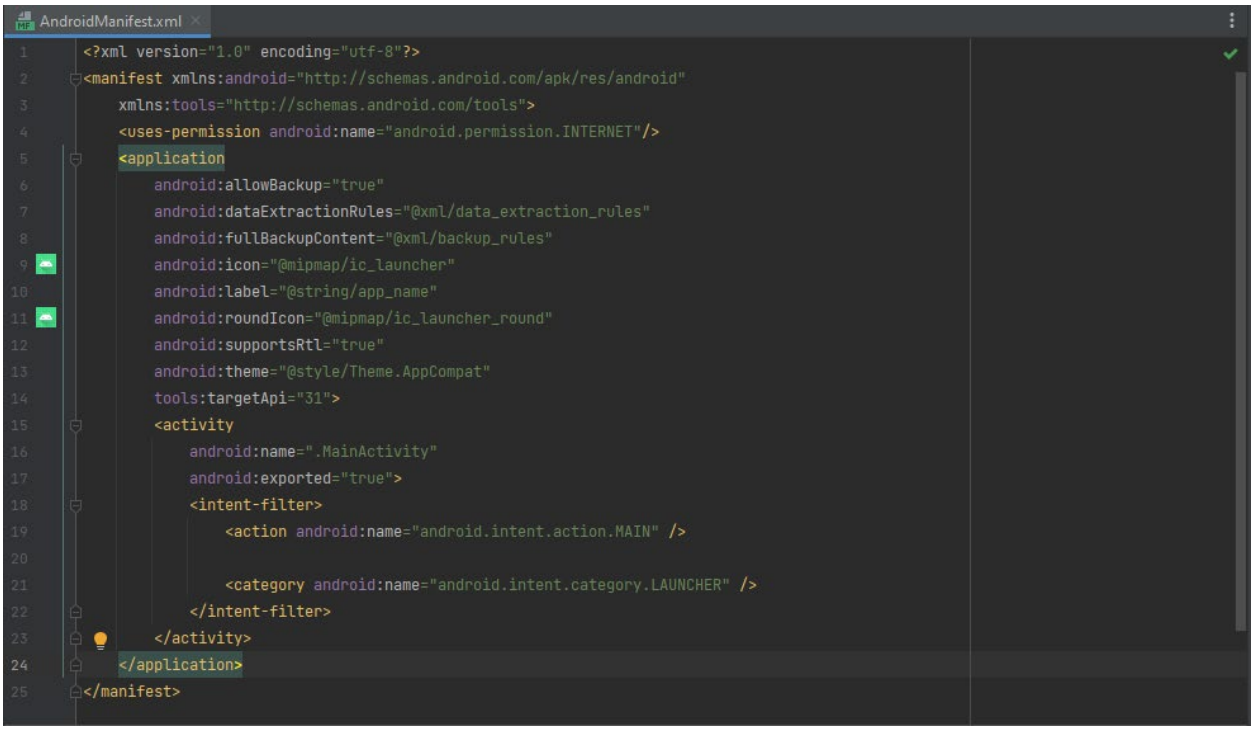
In the activity_main.xml file, I created a constraint layout and added a WebView widget with an id of webView as shown below.



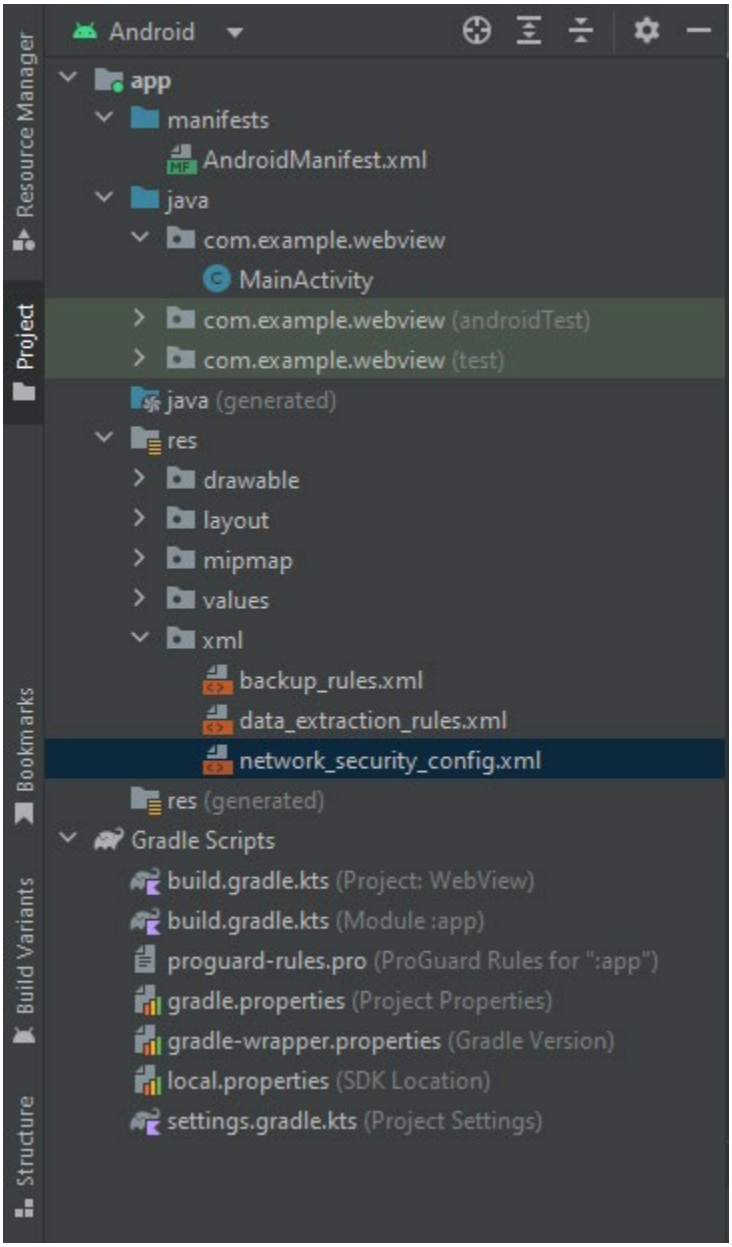
In the MainActivity.java file, I created a variable wv of type WebView and set it to be the webView widget I created earlier in activity_main.xml file. I created variable websettings of type WebSettings and set it to be the getSettings method. I set the websettings built in zoom controls to be of value true. I specified the webview url as shown below.



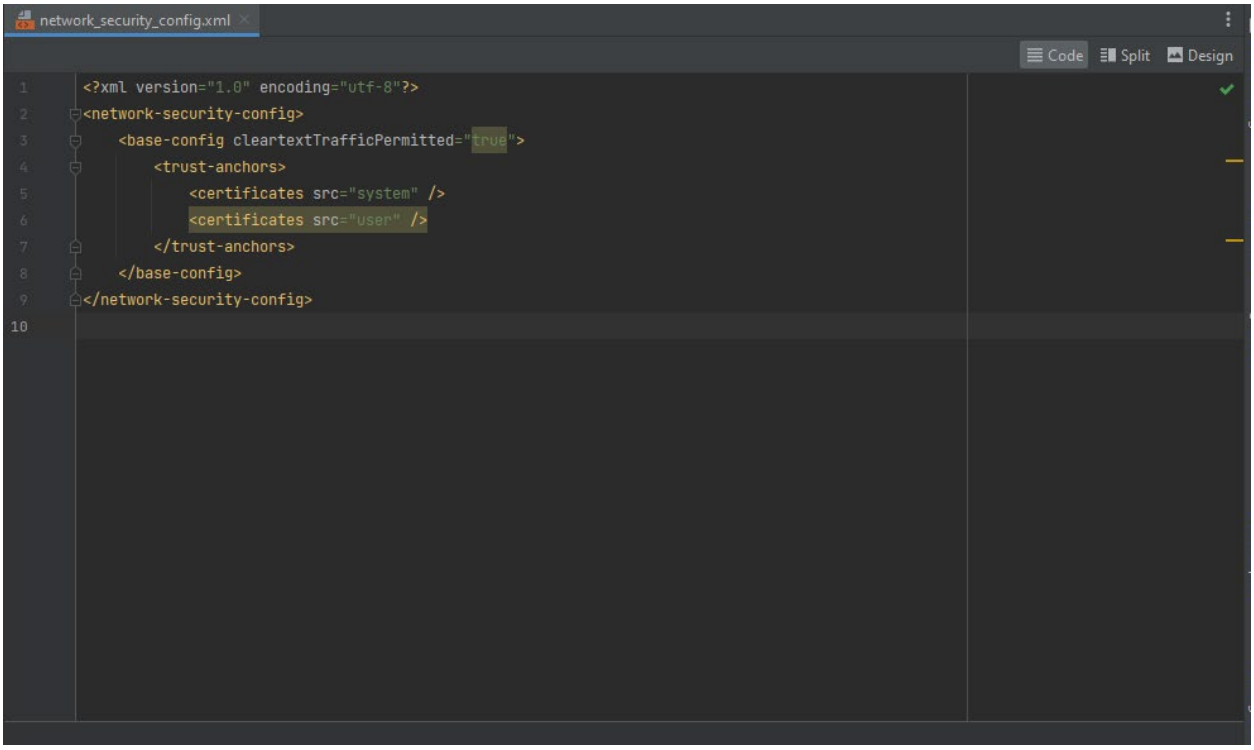
I gave permission for the project to use the internet in the AndroidManifest.xml file as shown below on line 4.



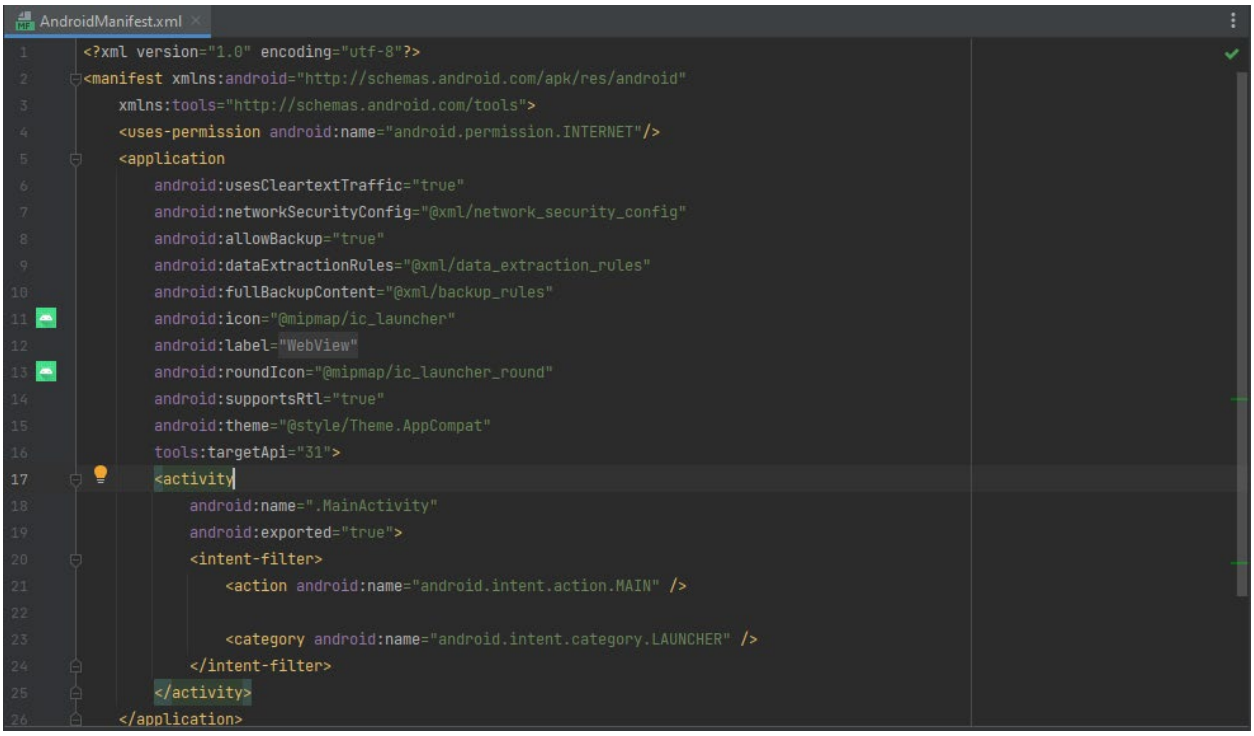
I created a network_security_config.xml file in the xml folder as shown below.



I configured it to allow access of the application to the internet as shown below.



Back in AndroidManifest.xml file, I set the usesClearTextTraffic value to be true and set the networkSecurityConfig value to look for the file I created previously in line 6 and 7 respectively. This is shown below.



This will allow the application to correctly run.

When I run the application, this is what I see.

When I drag the screen in the webView section, I see the zoom built in controls as show below:

When I tap the zoom in icon twice, it zooms in twice as show below:

When I tap the zoom out button twice, it zooms out as shown below:

This marks the completion of the project's objectives.