

Session 1 - Exercises

V1.0

JUAN RONDON

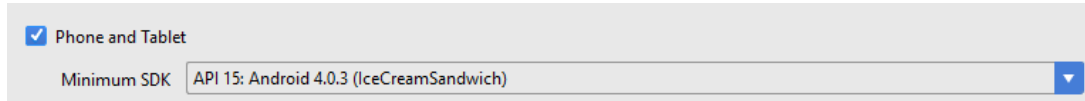


Table of Contents

Part 1	2
Layout Practice	2
Exercise 1 (Linear layout)	2
Exercise 2 (Relative layout)	3
Part 2	3
Android Note taking application	3

Part 1

Create a project for each of the following layouts in Android Studio. In the form factor selection make sure to select:



Layout Practice

Exercise 1 (Linear layout)

1. Make an Android application that contains eight **TextView** widgets and using the **Linear Layout** produce the layout from Figure 1.

Hint: In order to create the required layout, define the general layout of your application as a **Linear Layout** and then nest another **two Linear Layouts within the outer one** i.e.

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical">

    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:layout_weight="1"
        android:orientation="vertical">

        <LinearLayout>

        <LinearLayout
            android:layout_width="fill_parent"
            android:layout_height="fill_parent"
            android:layout_weight="1"
            android:orientation="horizontal">
        </LinearLayout>
    </LinearLayout>
</LinearLayout>
```

Note the **weight** and **orientation** property for each of the layouts.



Figure 1

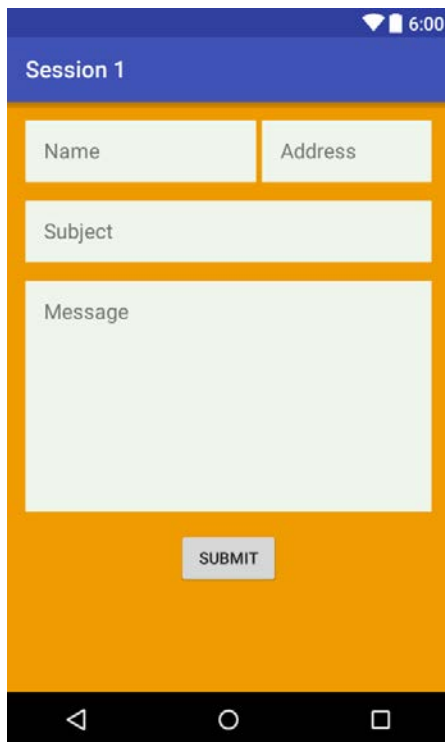
android:background

Red: #ff0202
 Green: #2bfc01
 Blue: #0328fe
 Yellow: #fee502

2. Test your app on the Android Emulator.
3. If you have an Android phone or tablet, test your app on it.
4. Update your app so that the **TextView text** is taken from **strings.xml resource file**.

Exercise 2 (Relative layout)

1. Make an Android application that contains four **EditText** widgets and one **Button**. There is a padding of 16dp in all sides for all elements including the layout.
2. Using a **RelativeLayout** produce the following output:



Colours:

Activity background: `#ef9b00`

EditText background: `#f8eff8f3`

Note: Use `android:hint` in order to add the text in the EditText widgets

3. Test your app on the emulator.
4. If you have an Android phone or tablet, test your app on it.
5. Update your app so that all text is taken from strings.xml.

Part 2

Android Note taking application

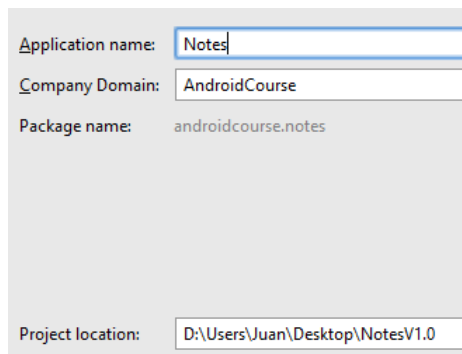
Through the course we will be working on creating a note taking application.

Each session we will be adding new functionality according to the topics covered during the session.

Today we will start with the basic layout required for all the required activities:

- MainActivity (Notes List)
- AddNoteActivity
- EditNoteActivity

1. Open Android Studio and create a new Android Application
2. From the configure window type the following:



Application name: Notes

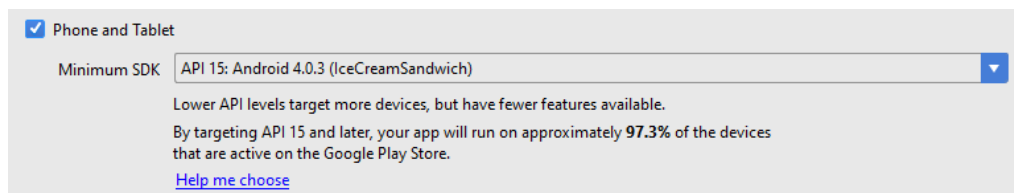
Company Domain: AndroidCourse

Package name: androidcourse.notes

Project location: D:\Users\Juan\Desktop\NotesV1.0

Note: Use any project location but save the project with the name of **NotesV1.0**

3. In the form factors window select:

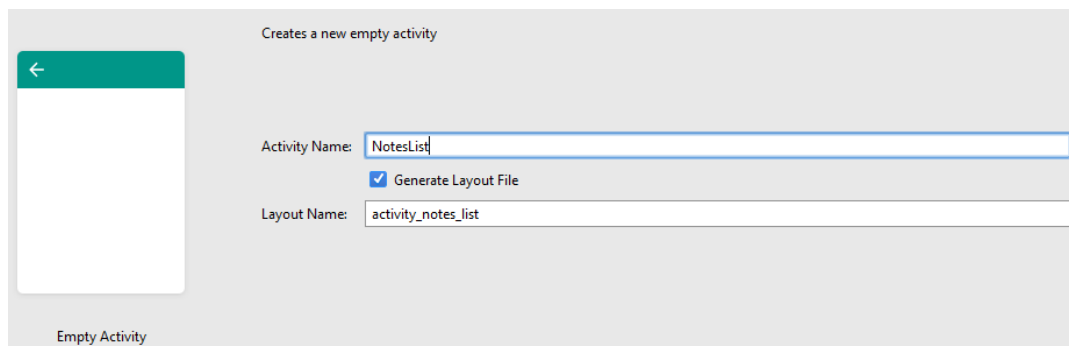


☒ Phone and Tablet

Minimum SDK: API 15: Android 4.0.3 (IceCreamSandwich)

Lower API levels target more devices, but have fewer features available.
By targeting API 15 and later, your app will run on approximately 97.3% of the devices that are active on the Google Play Store.
[Help me choose](#)

4. In the next window make sure to select an **Empty Activity**.
5. Apply the following settings/values for the first activity:



Creates a new empty activity

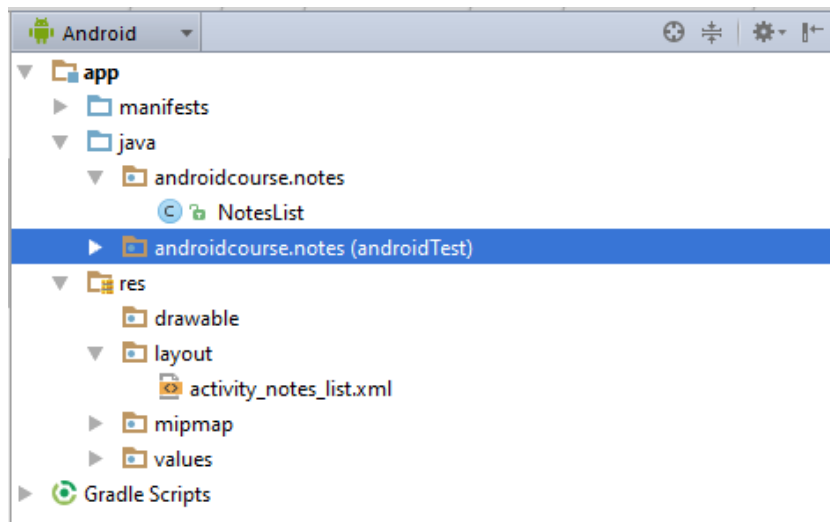
Activity Name: NotesList

☒ Generate Layout File

Layout Name: activity_notes_list

Empty Activity

6. After the project is created you should be able to see a project structure similar to:



7. Click on **activity_notes_list.xml** file and open the text view.
8. Replace the contents of the file with the following code:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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:background="#FFEB3B"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="androidcourse.notes.NotesList">

    <View
        android:id="@+id/top_rule"
        android:layout_width="fill_parent"
        android:layout_height="2dip"
        android:layout_marginTop="5dp"
        android:background="#696969" />

    <ListView
        android:id="@+id/listView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_above="@+id/bottom_rule"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_below="@+id/top_rule"
        android:layout_marginBottom="5dp"
        android:layout_marginTop="5dp"
        android:divider="@null"
        android:dividerHeight="4sp" />
```

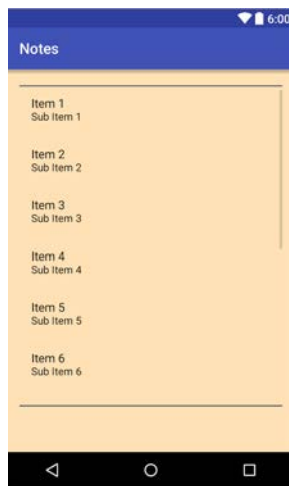
```

<View
    android:id="@+id/bottom_rule"
    android:layout_width="fill_parent"
    android:layout_height="2dip"
    android:layout_above="@+id/addNoteImg"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:layout_marginBottom="5dip"
    android:background="#696969" />

<ImageView
    android:id="@+id/addNoteImg"
    android:layout_width="40dp"
    android:layout_height="40dp"
    android:layout_alignParentBottom="true"
    android:layout_centerHorizontal="true"
    android:paddingBottom="5dp"
    android:src="@drawable/new_note_img" />
</RelativeLayout>

```

9. Preview the layout in the design view. (If you did everything correctly the following layout should be displayed):



Note: The `ImageView` widget won't be displayed because Android Studio can't find the image specified under `android:src="@drawable/new_note_img"`

```

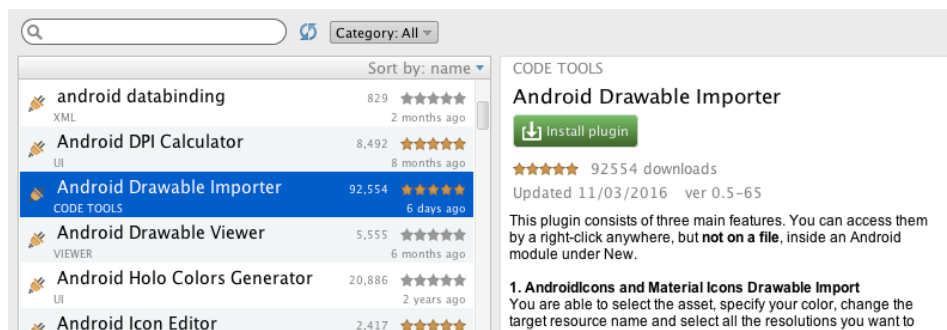
<ImageView
    android:id="@+id/addNoteImg"
    android:layout_width="40dp"
    android:layout_height="40dp"
    android:layout_alignParentBottom="true"
    android:layout_centerHorizontal="true"
    android:paddingBottom="5dp"
    android:src="@drawable/new_note_img" />

```

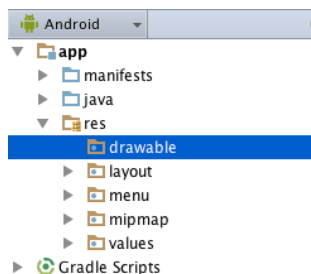
10. Now you will be adding the missing image used to create a new note.
11. In order to properly add images to our project, we want to install a plugin that will automatically create multiple versions of the images for all different devices resolutions:

- *ldpi* (low) ~120dpi
- *mdpi* (medium) ~160dpi
- *hdpi* (high) ~240dpi
- *xhdpi* (extra-high) ~320dpi
- *xxhdpi* (extra-extra-high) ~480dpi
- *xxxhdpi* (extra-extra-extra-high) ~640dpi

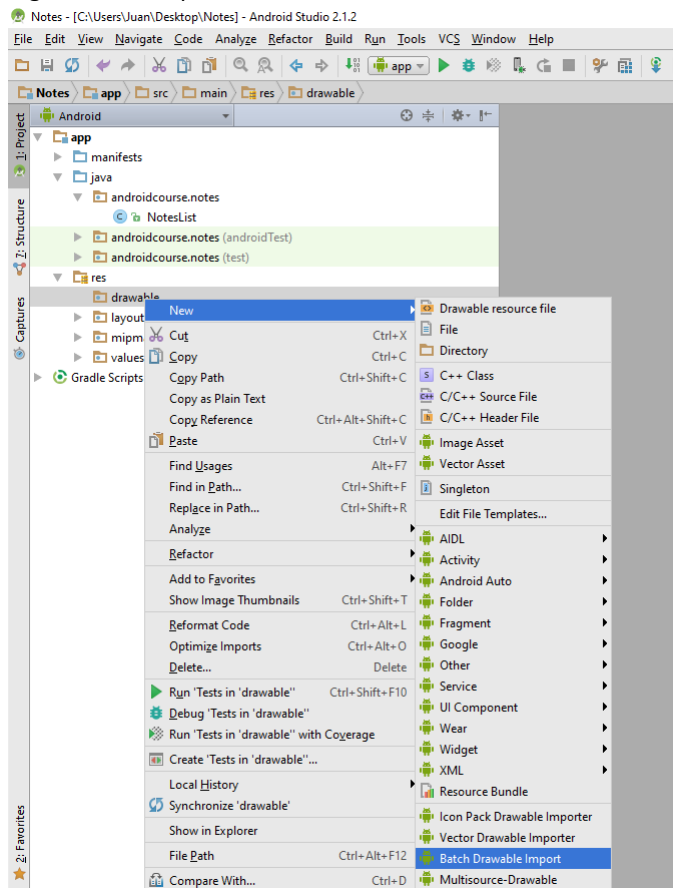
12. Open Android Studio preferences and then select plugins. A new window will open, here make sure to click on Browse repositories. From the list of plugins select and install (Android Drawable Importer)




13. Once the plugin is installed you will need to **restart** Android Studio.
14. Make sure the project structure view is set to **android** view and locate **drawable** resources folder.



15. Right click on top of **drawable** folder and select **new** then **batch drawable import** option.



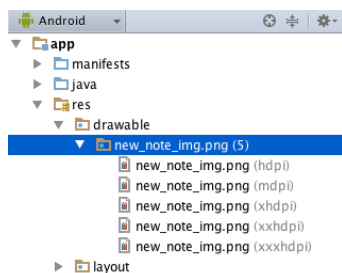
16. In the next window, click on  symbol and browse for the folder that contains the images for the project and select **new_note_img.png**. Make sure to apply the following values for your image.

Source-File	/Volumes/JUAN/2016/Android/Android/Session 1/Exercises/Required Images/new_note_img.png	
Source-Resolution	XHDPi	
Source-Width	256	
Source-Height	256	
Target-Resolution	<input type="checkbox"/> LDPI (96 px x 96 px) <input checked="" type="checkbox"/> XXHDPi (384 px x 384 px)	
	<input checked="" type="checkbox"/> MDPI (128 px x 128 px) <input checked="" type="checkbox"/> XXXHDPi (512 px x 512 px)	
	<input checked="" type="checkbox"/> HDPI (192 px x 192 px) <input type="checkbox"/> TVDPI (170 px x 170 px)	
	<input checked="" type="checkbox"/> XHDPi (256 px x 256 px)	
Target-Name	new_note_img	
Target-Root	/Volumes/JUAN/2016/Android/Android/Session 1/Temp/NotesV1.0/app/src/main/res	
Algorithm	Scalr	
Method	Automatic	
Format	PNG	

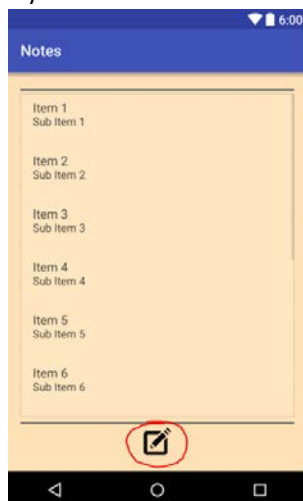
Note: do not modify any of the values after Target-Name even if they look different from the ones in this image.

Once everything is set up click on OK twice in order to close the wizard.

17. Now your image should be part of the project. Return to the Android view and you will see that your image is listed under **drawable** folder. The plugin automatically created multiple versions of the same image.



18. Now **activity_notes_list.xml** file should be able to reference the image and it will be displayed in your layout.



19. Now you will be adding the other two activities. In Android Studio select file → new → Activity → **empty activity** and make sure to set the following values for it.

Creates a new empty activity

Activity Name:

☒ Generate Layout File

Layout Name:

☐ Launcher Activity

Package name:

20. Repeat step 15 again but this time set the following values:

Creates a new empty activity

Activity Name:

☒ Generate Layout File

Layout Name:

☐ Launcher Activity

Package name:

21. Open **activity_add_note.xml** file in text view and replace the contents with the following code:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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:background="#FFEB3B"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="androidcourse.notes.AddNote">

    <View
        android:id="@+id/top_rule"
        android:layout_width="fill_parent"
        android:layout_height="2dip"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:layout_marginTop="55dp"
        android:background="#696969" />

    <EditText
        android:id="@+id/title_add"
        android:layout_width="250dp"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:layout_marginBottom="10dp"
        android:layout_toLeftOf="@+id/pwdCheckBox"
        android:layout_toStartOf="@+id/pwdCheckBox"
        android:hint="Note Title"
        android:singleLine="true"
        android:textStyle="bold" />
```

```

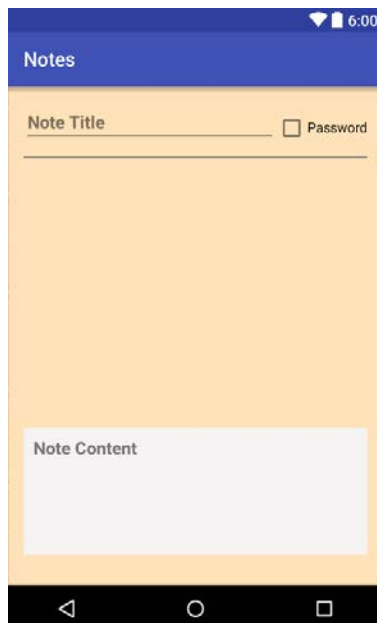
<EditText
    android:id="@+id/content_add"
    android:layout_width="wrap_content"
    android:layout_height="130dp"
    android:layout_alignEnd="@+id/top_rule"
    android:layout_alignParentBottom="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_alignRight="@+id/top_rule"
    android:layout_marginBottom="15dp"
    android:background="#f5f2f2"
    android:gravity="top"
    android:hint="Note Content"
    android:padding="10dp"
    android:textStyle="bold" />

<CheckBox
    android:id="@+id/pwdCheckBox"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignBottom="@+id/title_add"
    android:layout_alignEnd="@+id/top_rule"
    android:layout_alignRight="@+id/top_rule"
    android:checked="false"
    android:text="Password" />

</RelativeLayout>

```

22. Preview the layout in the design view. (If you did everything correctly the following layout should be displayed):



23. Repeat from step 21 but this time you will be adding the code for **activity_edit_note.xml**.

(Required layout code in next page)

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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:background="#FFEB3B"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="androidcourse.notes.EditNote">

    <View
        android:id="@+id/top_rule"
        android:layout_width="fill_parent"
        android:layout_height="2dip"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:layout_marginTop="55dp"
        android:background="#696969" />

    <ImageView
        android:id="@+id/deleteNoteImg"
        android:layout_width="30dp"
        android:layout_height="30dp"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_marginLeft="110dp"
        android:paddingBottom="5dp" />

    <EditText
        android:id="@+id/title_edit"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:imeActionLabel="Done"
        android:inputType="textCapSentences"
        android:singleLine="true"
        android:textStyle="bold"
        android:typeface="normal" />

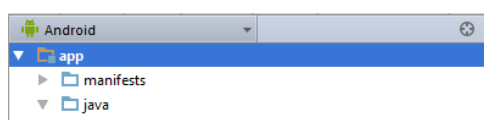
    <EditText
        android:id="@+id/note_info_edit"
        android:layout_width="wrap_content"
        android:layout_height="130dp"
        android:layout_alignEnd="@+id/top_rule"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_alignRight="@+id/top_rule"
        android:layout_marginBottom="15dp"
        android:background="#f6f3f3"
        android:gravity="top"
        android:padding="10dp" />
</RelativeLayout>

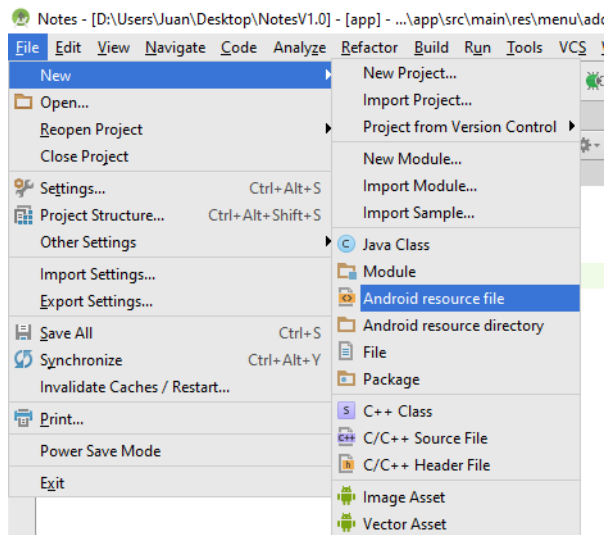
```

24. Preview the layout in the design view.

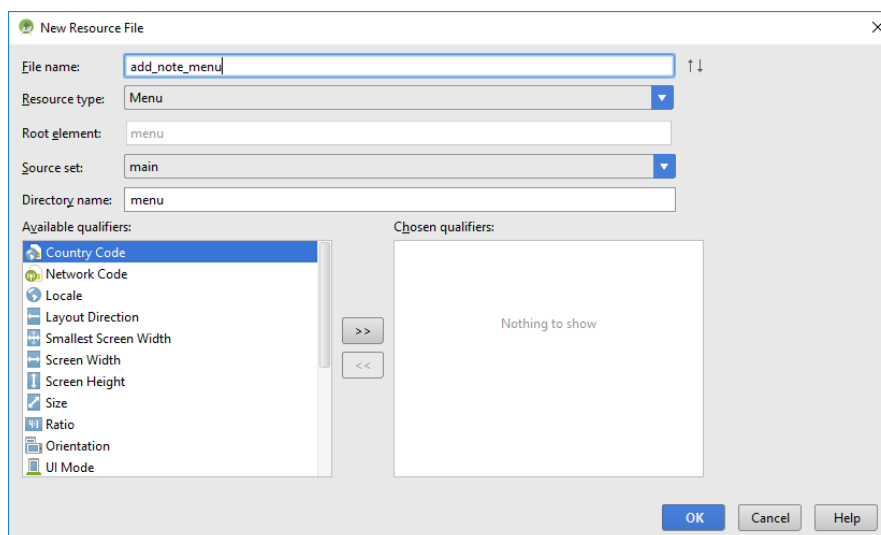
25. Now we need to add some action buttons for our Activities (**activity_edit_note.xml** and **activity_add_note.xml**). in order to do that you will be creating a **menu** for those two activities.

Note: Make sure that you highlight App before proceeding to the next step.



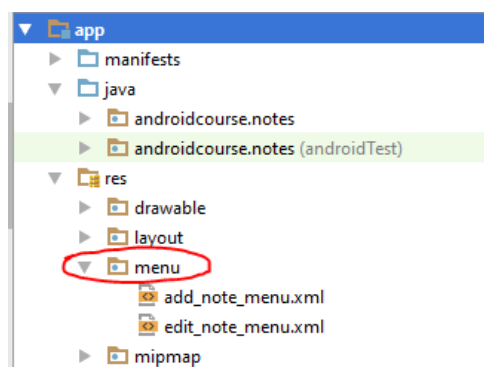


26. Make sure to input the following values:



27. Repeat the same process for adding an **edit_note_menu.xml** layout.

28. After the above steps a new folder named menu will be created in your project structure:



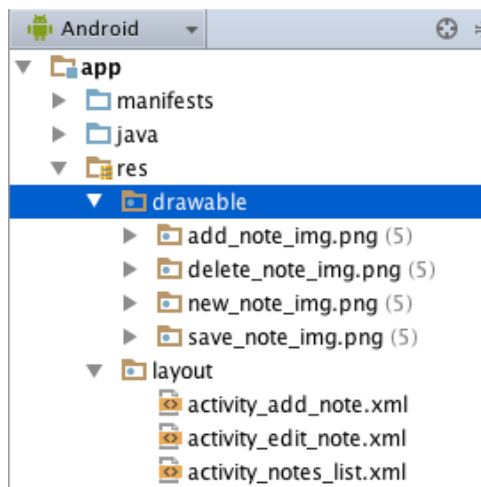
29. Open **add_note_menu.xml** and replace the contents with the following code:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:app="http://schemas.android.com/apk/res-auto">
    <item
        android:id="@+id/SaveNote"
        android:icon="@drawable/add_note_img"
        android:title="Save"
        app:showAsAction="ifRoom" />
</menu>
```

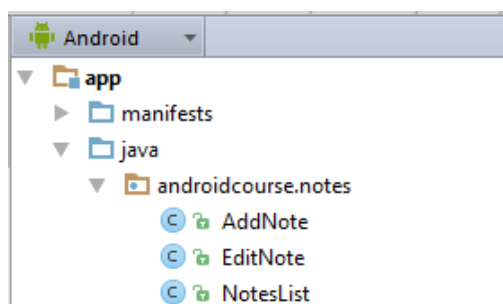
30. Repeat the same process for **edit_note_menu.xml**.

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:app="http://schemas.android.com/apk/res-auto">
    <item
        android:id="@+id/SaveNote"
        android:icon="@drawable/save_note_img"
        android:title="Save"
        app:showAsAction="ifRoom" />
    <item
        android:id="@+id/DeleteNote"
        android:icon="@drawable/delete_note_img"
        android:title="Delete"
        app:showAsAction="ifRoom" />
</menu>
```

31. Add **save_note_img**, **add_note_img** and **delete_note_img** to the **drawable** resources using the same procedure from steps 14-16.



32. Next we want to enable the menu for **AddNote** and **EditNote** activities. In the project explorer open the java folder.



33. Open the java file named **AddNote** and add the following code under **onCreate** method.
(resolve any missing dependencies *with alt + enter*)

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.add_note_menu, menu);
    return true;
}
```

34. The finished class should look like this: (Next page)

```
package androidcourse.notes;

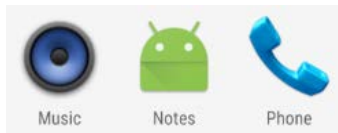
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuInflater;

public class AddNote extends AppCompatActivity {

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

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.add_note_menu, menu);
        return true;
    }
}
```

35. Repeat the same procedure for **EditNote** java file. Make sure to inflate **edit_note_menu** this time.
36. Open the application in Android Studio emulator (or your own device).
You will notice a new installed application with the default android icon on the emulator/device.



Touch/click on the app to open it.

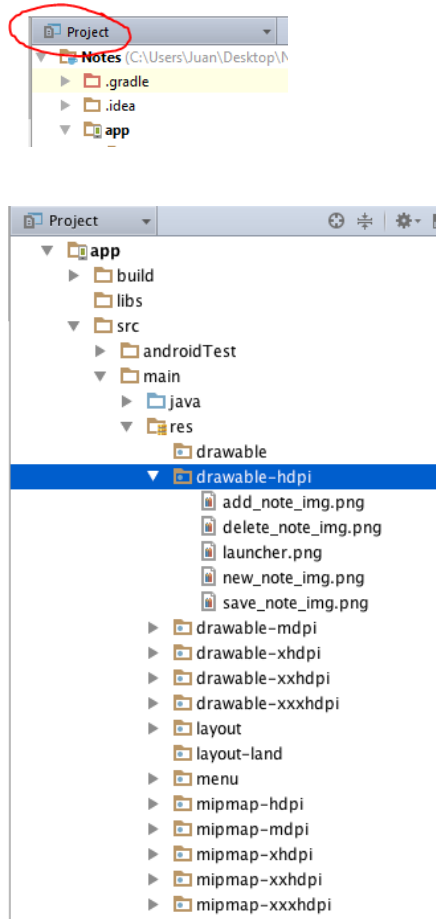


37. The final step is to change the default icon for the application. By default, android uses the android icon for your application but we like to use something more appropriate for our notes application.

Android Application icon should be added in a special resource folder called **mipmap**.

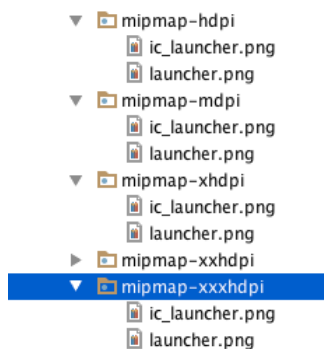
Unfortunately, the plugin used does not save the images in the **mipmap** folder, it always saves images to drawable folder. We will use it to generate all the different resolutions and then we have to manually move each of the images to the appropriate **mipmap** folder.

Note: In order to get the following view, change the default View from **Android** into **project**



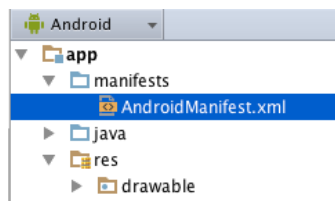
Important: Drag and drop **launcher.png** from **drawable-hdpi** into **mipmap-hdpi** and repeat for all the other folders.

38. After all is moved, you should have a structure similar to the following image:



39. Now we need to tell Android Studio the new location for the Application icon.

Open **AndroidManifest.xml** file located under **manifest** folder:

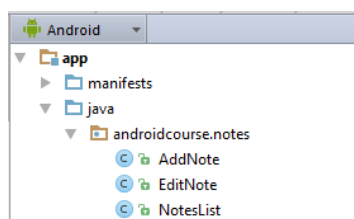


40. Locate the line that specifies the application icon and change it to:

android:icon="@mipmap/launcher"

41. Open the application in Android Studio emulator (or your own device). Make sure to test the three layouts that we created.

To do so locate the java files in your project structure:

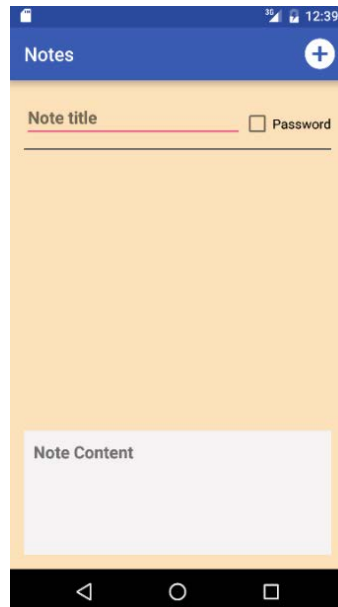


Right click on **AddNote** and select **Run AddNote** then repeat in order to test the other two activities.

Notes List



Add Note



Edit Note

