**Formative Assignment 1**

**Task 1**: Identify and discuss how Android framework supports fluid layout, which will gracefully adapt the user interface to the screen resolution automatically.

With the use of Android frameworks, fluid layouts means that no matter how does the screen resolution change, the layout and elements contained in a particular application will scale appropriately and ensures that functionality is not affected. Some examples would include layouts such as linear and frame layout which allows the layout and elements to scale accordingly to the screen resolutions. In addition to that, layouts can be coded specifically for specific screen sizes and also orientations to further ensure that the elements in the application are displayed in the appropriate manner. Different resolution of images in the application also can be supplied to ensure the elements in the application is not affected by the screen resolution.

Task 2: Create simple app to achieve a given interface using different layouts (linear, relative)

Some of the challenges faced during this task is deciding on how the layout should be organized, and how to nest layouts to make them look like the desired result. The only difficult part was understanding how the layouts work individually, and how to combine them together, and what properties needs to be used.

****

Code:

<?xml version="1.0" encoding="utf-8"?>  
<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"  
 tools:context="com.example.lesgo.assignment\_task2.MainActivity"  
 android:background="@drawable/bbbbbbbbbbbbbb"  
 android:orientation="vertical">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 >  
  
  
 <RelativeLayout  
 android:layout\_width="0dip"  
 android:layout\_weight="1"  
 android:layout\_height="wrap\_content">  
 <ImageView  
 android:paddingTop="20px"  
 android:layout\_width="match\_parent"  
 android:layout\_height="800px"  
 android:src="@drawable/mike"  
 android:id="@+id/Mike"  
 />  
  
 <TextView  
 android:paddingTop="30px"  
 android:paddingBottom="30px"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="@string/mike"  
 android:layout\_below="@id/Mike"  
 android:textAlignment="center"  
 android:textSize="14pt"  
 android:textStyle="bold"/>  
 </RelativeLayout>  
 </LinearLayout>  
  
  
  
  
 //before this  
 <RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:background="#B3ffffff"  
 android:paddingBottom="40dp">  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="@string/programming"  
 android:textSize="17dp"  
 android:paddingTop="5dp"  
 android:id="@+id/prog"  
 />  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/proficiency"  
 android:textSize="17dp"  
 android:id="@+id/pro"  
 android:layout\_below="@id/prog"  
 />  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/android"  
 android:layout\_alignRight="@id/prog"  
 android:paddingTop="5dp"  
 android:paddingRight="220dp"  
 android:textSize="13dp"  
 android:id="@+id/android"  
 android:textColor="#ff0000"/>  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/java"  
 android:layout\_alignRight="@id/android"  
 android:paddingTop="5dp"  
 android:paddingRight="180dp"  
 android:textSize="13dp"  
 android:textColor="#ff0000"/>  
  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/cs"  
 android:id="@+id/cs"  
 android:layout\_below="@id/android"  
 android:layout\_alignRight="@id/prog"  
 android:paddingTop="2dp"  
 android:paddingRight="220dp"  
 android:textSize="13dp"  
 android:textColor="#ff0000"/>  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="@string/personality"  
 android:textSize="18dp"  
 android:paddingTop="10dp"  
 android:id="@+id/personality"  
 android:layout\_below="@id/pro"  
 />  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/mike1"  
 android:id="@+id/perso"  
 android:layout\_below="@id/cs"  
 android:layout\_alignRight="@id/personality"  
 android:paddingLeft="140dp"  
 android:paddingTop="18dp"  
 android:textSize="13dp"  
 android:paddingRight="14dp"  
 android:textColor="#000000" />  
  
 </RelativeLayout>  
  
  
</LinearLayout>

String XML:

<resources>  
 <string name="app\_name">Assignment1\_task2</string>  
 <string name="mike">Mike Wazowski</string>  
 <string name="programming">Programming </string>  
 <string name="proficiency">Proficiency</string>  
 <string name="android">Android</string>  
  
 <string name="java">Java</string>  
  
 <string name="cs">C Sharp</string>  
  
 <string name="personality">Personality</string>  
  
 <string name="mike1">Mike Wozowski is a proud and confident monster. Mike has a romantic side, shown by dating Cella Mae, presumably the one Mike  
 wants to tie the knot with, and refers to as his <b>schmoopsie-poo</b> although he has a self-centered side</string>  
  
</resources>

Task 3: Briefly explain the separation of concerns in software engineering and application in task 2.

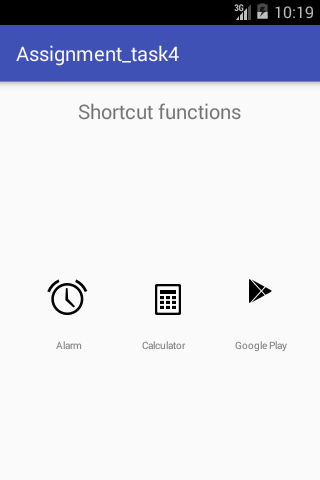
Separation of concerns in software engineering is the decomposition into parts which has minimal to no overlapping of functionality. The dependency between two parts should be minimal as this makes debugging and bug-fixing easier. In addition to that, it promotes extensibility in a software as new functionality can be added without having to worry about other aspects which are not related to functionality (such as the interface). In other words, separation of concerns is keeping codes and the interface, and changing the codes should not require changing the interface and vice versa.

In android studio, by the use of frameworks and libraries, presentation (XML files) and codes (java files) are separated entirely with very minimal dependency. Strings are added into strings.xml instead of using them under the *android:text* property*.*

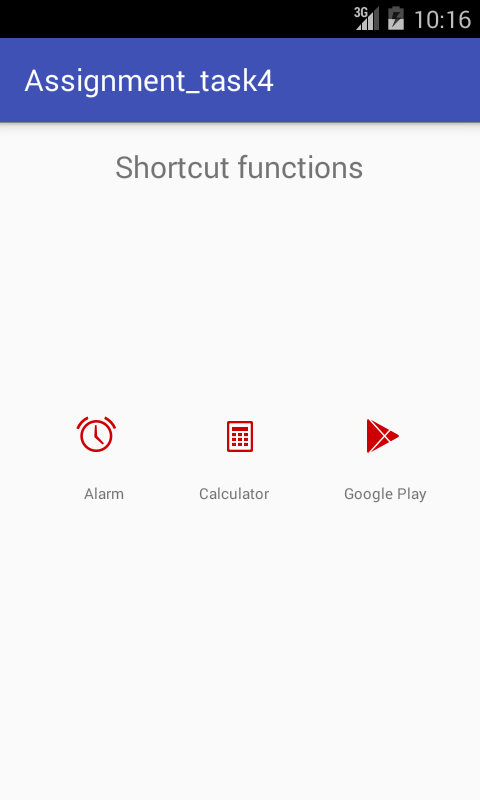
Task 4: Make app that uses different density images on different resolutions

Using a lower DPI image on a high DPI screen will cause an image to not look like how it was meant to be. Stretching a low DPI image will cause it to be pixelized and affect the entire outlook of the application

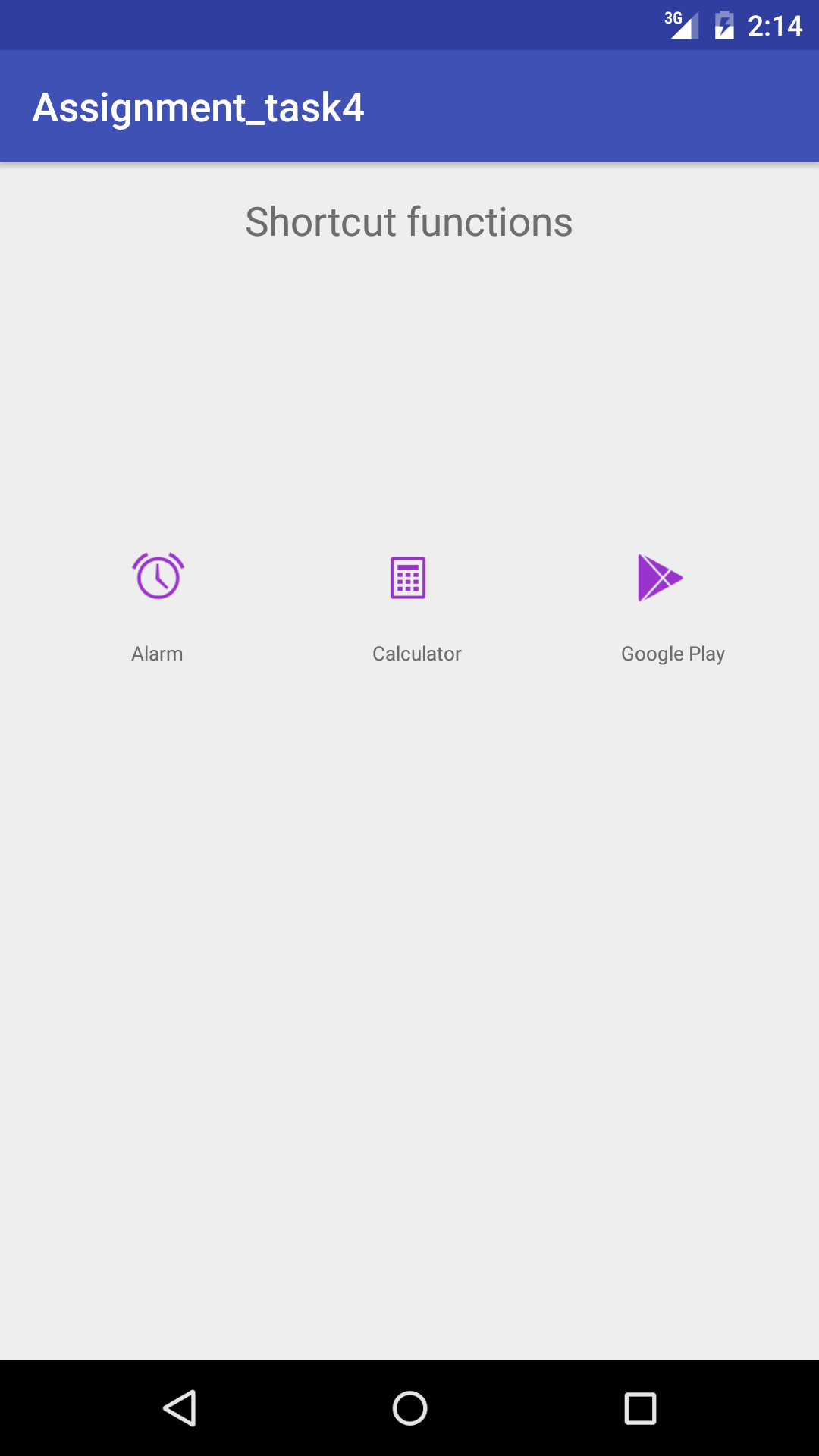
MDPI:



HDPI:



XHDPI:



XHDPI with LDPI:

