Google Project Challenge

By Rajeev Jhaj

Task:

For your selected assignment you must provide us with all the code and a file with instructions on how to build and install your application. Preferred option: upload your code to GitHub and provide us with a link to the source code.

You may use any available resources, tutorials, and assistance on these projects. What resources you used, how much you learned, and what you found challenging is of as much interest to us as how well you complete the task.
A few tips: While there are plenty of online resources where you can find code samples to get started, we'd like to see a significant amount of code written by you. You should understand any code you reuse. Including 3rd-party libraries is fine, and even encouraged rather than rewriting something very complex or very basic…as long as it's done smartly.

Option 1 - Android

Build a simple Android app that displays pictures of cats.

You can include the images inside your app or download the images while the app is running.

- Display at least 5 images.
- User must be able to view each image individually.
- User must be able to navigate between the different images.
- User must have an option to exit the app.

Some resources:
https://developer.android.com/training/basics/firstapp/index.html
https://developer.android.com/training/building-graphics.html
https://placekitten.com/

Android:

Introduction:
I have chosen to do the first option. This is mainly because it sounded interesting and beneficial to me to learn the skill of creating an android app. I also had a small amount of experience with this from a project I attempted a year ago in the Android Studio.

Thinking about prior knowledge of Android apps, I remember a simple app I attempted creating over 3 years ago in Android studio which created Birthday Cards, I hoped I would remember some of this project whilst creating this app. I also recalled hearing about the

Kotlin language introduced in Google IO 2017. Maybe this could be of some use while creating this app.

https://www.theverge.com/2017/5/17/15654988/google-jet-brains-kotlin-programming-language-android-development-io-2017

I also remembered hearing about Flutter during Google IO 2018, I remember this would help my app looking more fluid, this might also be of use while I create my app.

https://9to5google.com/2018/12/04/watch-flutter-live-2018/

After looking at the 4 criteria of the brief I decided that the best implementation of this app would be to have 5 buttons when the app is opened. The first 5 buttons display each picture of a cat. The final button will allow the user to exit the app. This design may change throughout my development as I learn more how Android apps are built and what looks better graphically.

Development:

I started with the first resource provided by the brief.
https://developer.android.com/training/basics/firstapp/index.html

This was a guide on how to build my first app. I first learnt about activities and their entry points. I learnt the “main” activity is opened when I click the app icon. This meant that when I click the icon button, I want the “main” activity open and this should contain my 6 buttons.

I then followed the instructions to download and install the Android Studio 3.3.2 for Windows. This took a while. While I waited, I went to the resource for kitten images and downloaded 5 images of very cute kittens.
https://placekitten.com/

Once installed and I created a project with the app name, “Cute Kittens”, I created an activity that had a BottomNavigationView instead of an empty activity as I have used this view on previous apps and thought it would be useful to have this view to navigate the images. I also chose to code with the Kotlin instead of Java. This meant I could no longer follow the instructions from the first resource. I found this instead:
https://codelabs.developers.google.com/codelabs/build-your-first-android-app-kotlin/index.html

In the first few steps of this tutorial I understood the basics of the Android Studio. It was similar to Pycharm as it was both created by JetBrains. When I ran my app, I connected my OnePlus 6T device to my PC, enabled ADB debugging and this allowed the app to appear on my phone. On Android Studio I could view the blueprint design of the app and the code behind it. For the navigation bar I edited the name dashboard to photos, and notifications to developer. I also changed a navigation icon and chose one that was already from a library.

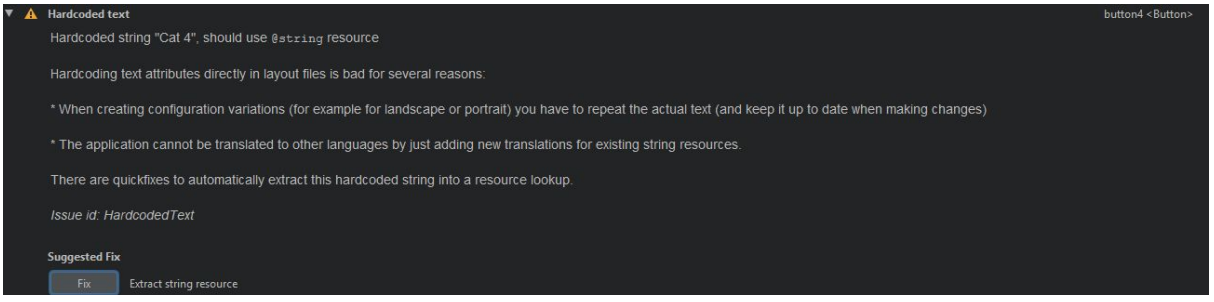
I next added a Welcome TextView message. Then inserted 6 buttons underneath.

activity_main.xml | activity_main1.xml | activity_main2.xml | activity_main3.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.a
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/container"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <android.support.design.widget.BottomNavigationView
        android:id="@+id/navigation"
        android:layout_width="411dp"
        android:layout_height="wrap_content"
        android:background="?android:attr/windowBackground"
        app:menu="@menu/navigation"
        tools:layout_editor_absoluteX="0dp"
        tools:layout_editor_absoluteY="686dp" />
    //Welcome Message
    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Welcome!"
        android:textAppearance="@style/TextAppearance.AppCompat.Display2"
        tools:layout_editor_absoluteX="106dp"
        tools:layout_editor_absoluteY="29dp" />
    //Image Button for First Cat
    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Cat 1"
        tools:layout_editor_absoluteX="161dp"
        tools:layout_editor_absoluteY="140dp" />
```

I then learnt about how the res folder contained all the assets I needed. I opened up the drawable res folder in my explorer and inserted 5 photos of cats.
I created 5 new activities and in each one I had a large image of each cat. I ran into many errors and warnings. For example:

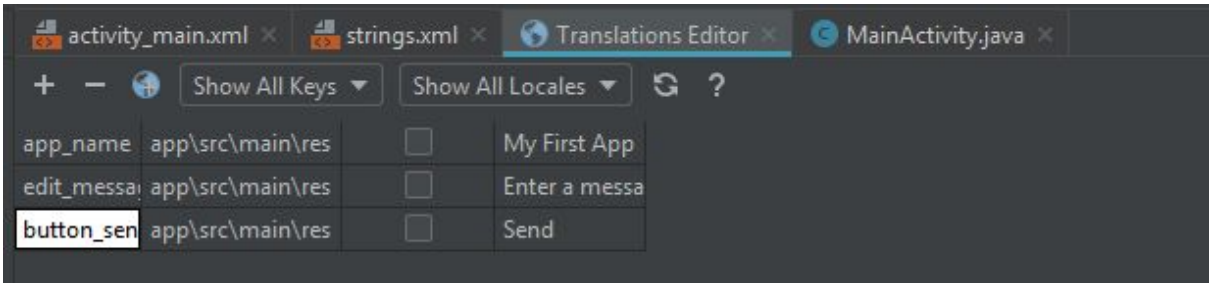


I had to fix this by changing the resource name to @string/cat_4. I also had missing constraint errors which I researched and found this was the height and width from the edges of the window. I had to add this to fix it:

```
android:layout_marginStart="8dp"
android:layout_marginLeft="8dp"
android:layout_marginEnd="8dp"
android:layout_marginRight="8dp"
```

Before I moved on I tried running the app again on my phone but there was lots of errors that came up, to rectify this I attempted to fix each one individually. For example, the images weren't named correctly and weren't in the right format. Also, I had edited the navigation bar incorrectly and I didn't know how to fix this so I tried again by opening a new project.

This time I would follow the instructions exactly and not use Kotlin. This app was called "MyFirstApp". In this project I felt a lot more comfortable as it was explaining what I was doing whilst I was doing it. For example:



This was the string resources file where I told it my UI strings which allowed me to manage all UI strings in a single place.

The instructions made me realise the importance of designing the layout, the constraints had to be perfect so that it could work on every device. Once I created a textbox and button I ran it and on my phone, a textbox and button appeared.

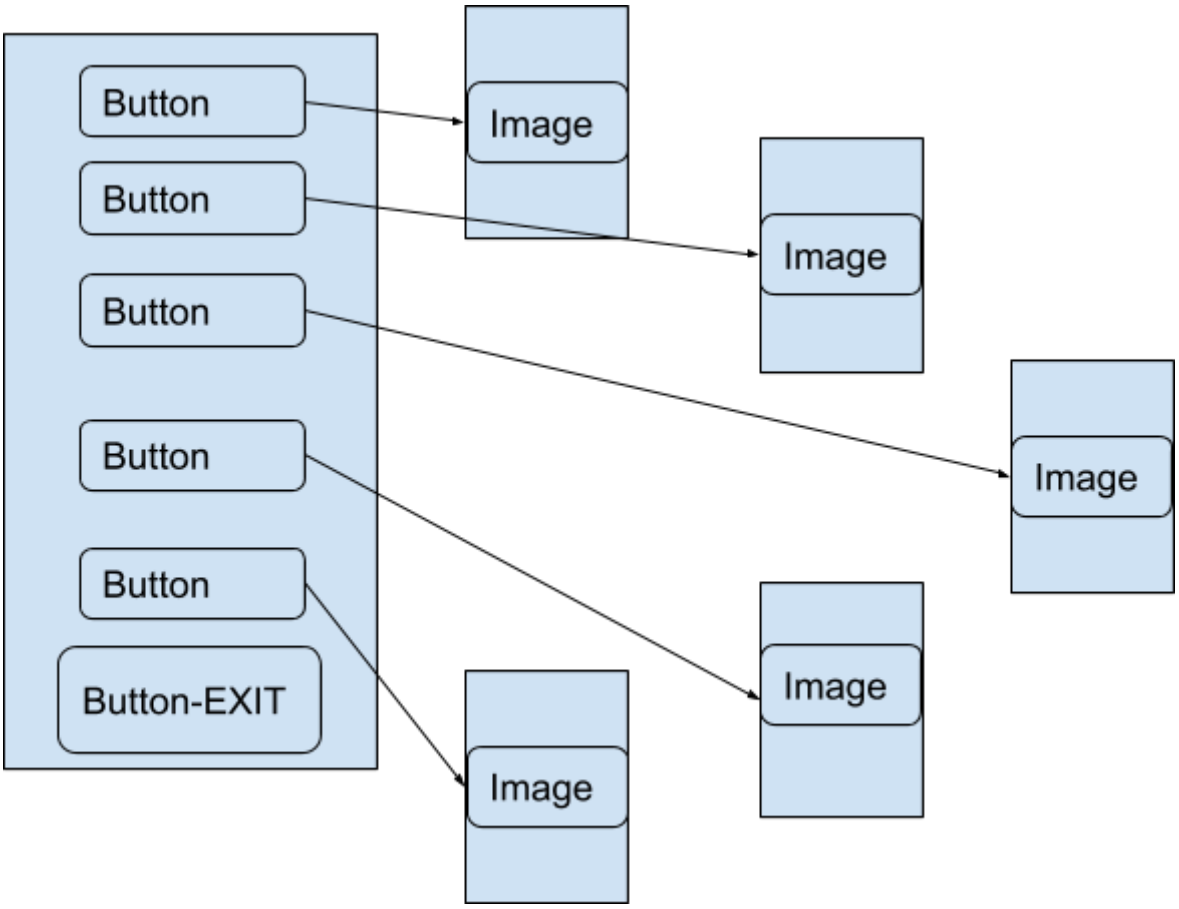
On the next lesson I was taught how to start another activity when the button was pressed. In the MainActivity file, I added:

```
public void sendMessage(View view) {
```

This was called when the user tapped the Send button which did something in response to button. I next had to an Intent. This is what binded to activities together. After I used the provided Java code, it allowed the text from the textbox appear in the TextView off the DisplayMessageActivity. I ran the app and it worked.

Now I finished the tutorial I understood how all the elements of the app worked but still had no understanding of how images appeared
I thought again how my design would be whilst Android Studio reopened after crashing multiple times. For my app to be successful it needed to meet the 4 criteria of the brief. To

do this I needed 6 buttons, 5 to display each image, another to exit.



I now knew how to create the first main activity and one button.I removed the textbox from the app and attempted to remove the code to send the text through, the button should only display the next window. I got this working for 1 button, when I clicked it, it displayed the activity with some text I had written on. I did this again for another button and activity, this produced another activity to display, I then managed to get all 5 buttons displaying 5 separate activities. My next step was to add an image, I tried this for the first button. This involved going to the design section of the first activity and dragging an image. I then downloaded 5 new png images of very cute cats and moved them to my res/drawable folder for use. This meant the photo showed up in Android Studio, I then set a constraint for 64 pixels below the text. I repeated this for the 4 other images and tested it out. It worked!

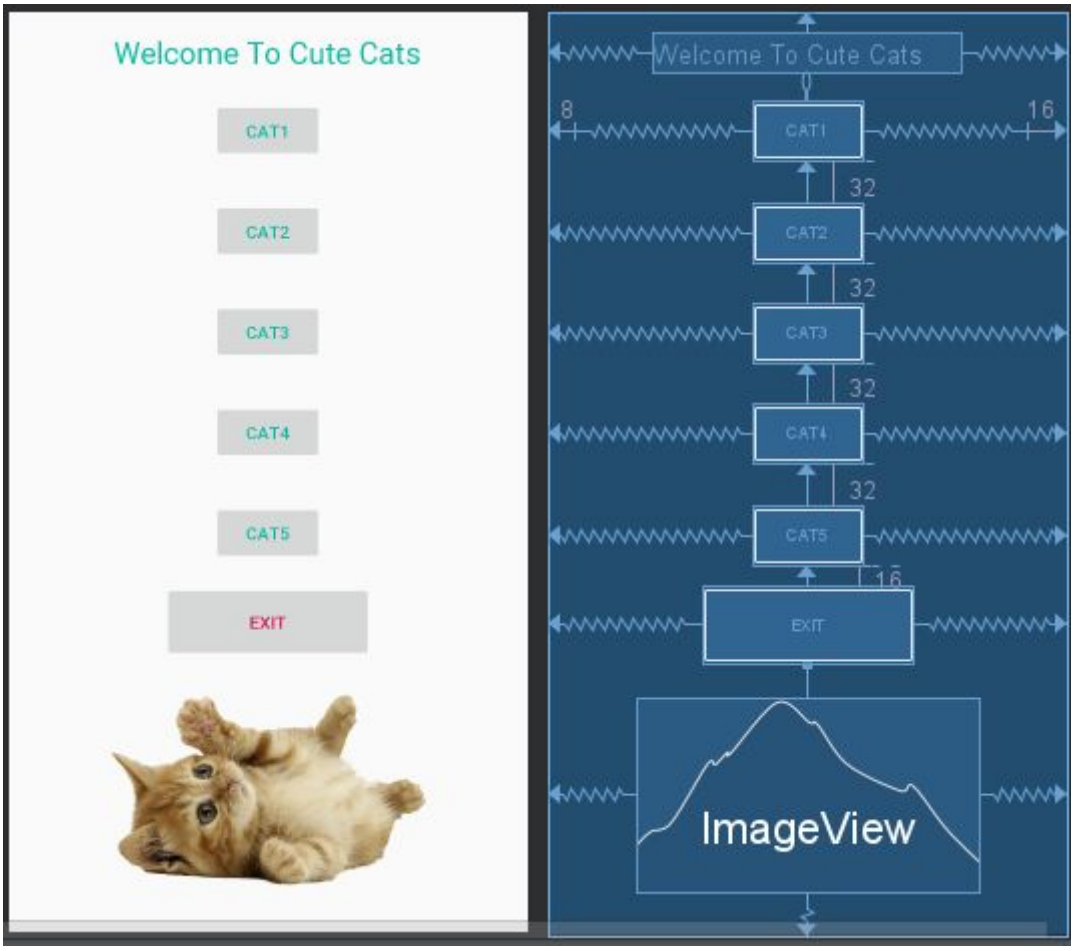
By now I had completed 3/4 of my criteria. I still had to create a button to exit the app.

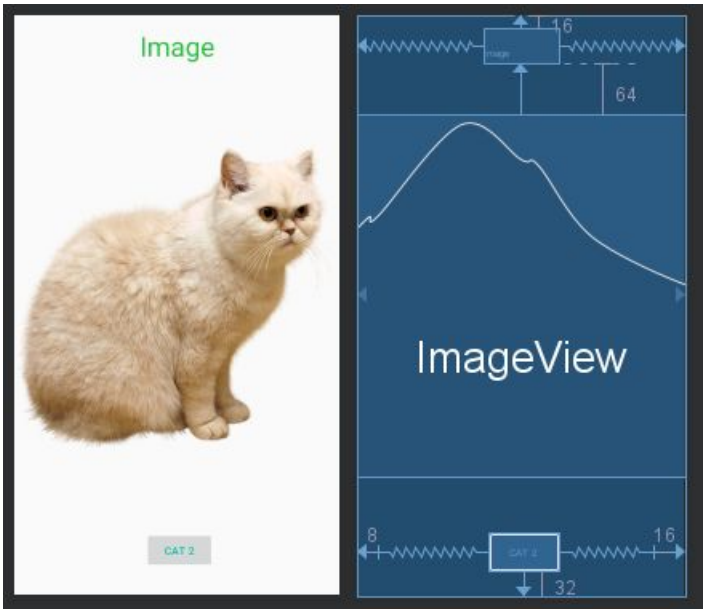
https://stackoverflow.com/questions/6014028/closing-application-with-exit-button

```
finish(); System.exit(0);
```

These lines told me how to exit the application. All that was left to do was create the button, let the onClick cause the exitApp method to run inside of the MainActivity and this would exit the program. I have just tested this and it fully works. The process was tedious and I did struggle at many parts but overcame this with perseverance and trial and error. I then created the app with help from:
https://www.youtube.com/watch?v=0ehWKcnDEz0

I tried out the app for user testing. This made me next added on smaller features like buttons on each activity to navigate through each page and colours to make it look a little nicer.





Build:

To create an APK of the app you click the build tab at the top of the Android Studio, click Generate Signed Bundle or APK.
Click APK.
Set a place to store the app.
Set a username (key alias) and key password.
Click Release.
Click V1 and V2 signature versions.
Click Finish.

This creates an APK available to install on an Android device.