Mobile Dev Week 3 Lab

Daniel Coady (102084174)

02/09/2019

Food Parcels

Task 1

```
// called when we wish to view the pizza
fun viewPizza(view: View) {
    val intent = Intent( packageContext: this, ImageDisplay::class.java).apply { this: Intent
        putExtra( name: "description", value: "pizza")
        putExtra( name: "id", R.drawable.pizza)
    }
    startActivity(intent)
}
```

Figure 1: The code that allows the two activities to connect with each other

To allow for each activity to communicate with each other for consistency across the app, we can use intents. With intents we are able to declare what activity we wish to be able to move onto as well as the data that should go along with it. In this case we are able to send a description and the id of the photo that has been touched so that we can show a larger version of it along with it's short description.

```
<TableLayout
        android:id="@+id/tblImages"
        android:layout_width="104dp"
        android:layout_height="408dp"
        android:layout_marginTop="15dp"
        android:layout_marginBottom="15dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">

    <TableRow
            android:layout_width="326dp"
            android:layout_height="match_parent">

        <ImageView
                android:id="@+id/imgPizza"
                android:layout_width="100dp"
                android:layout_height="100dp"
                android:onClick="viewPizza"
                app:srcCompat="@drawable/pizza" />
    </TableRow>

    <TableRow
            android:layout_width="match_parent"
            android:layout_height="match_parent">

        <ImageView
                android:id="@+id/imgNachos"
                android:layout_width="100dp"
                android:layout_height="100dp"
                android:onClick="viewNachos"
                app:srcCompat="@drawable/nachos" />
    </TableRow>

    <TableRow
            android:layout_width="match_parent"
            android:layout_height="match_parent">

        <ImageView
                android:id="@+id/imgTakis"
                android:layout_width="100dp"
```

```
<Button
        android:id="@+id/btnBack"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="17dp"
        android:onClick="back"
        android:text="Go Back"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/lblDescription" />

<TextView
        android:id="@+id/lblDescription"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="14dp"
        android:layout_marginBottom="7dp"
        android:text="TextView"
        app:layout_constraintBottom_toTopOf="@+id/btnBack"
        app:layout_constraintEnd_toEndOf="@+id/btnBack"
        app:layout_constraintTop_toBottomOf="@+id/imgLorge" />

<ImageView
        android:id="@+id/imgLorge"
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:layout_marginStart="55dp"
        android:layout_marginTop="40dp"
        android:layout_marginEnd="55dp"
        android:layout_marginBottom="172dp"
        app:layout_constraintBottom_toTopOf="@+id/lblDescription"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@mipmap/ic_launcher" />
```

Figure 2: Part of the XML used for the activities

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_image_display)

    // get the content from the intent that started this activity
    val message = intent.getStringExtra( name: "description")
    val id = intent.getIntExtra( name: "id",  defaultValue: 0)
    lblDescription.text = message
    imgLorge.setImageDrawable(resources.getDrawable(id))
}
```

Figure 3: Code used to set the views in the image view activity

Here we are now collecting the information stored in the intent from the previous activity to display the information requested.

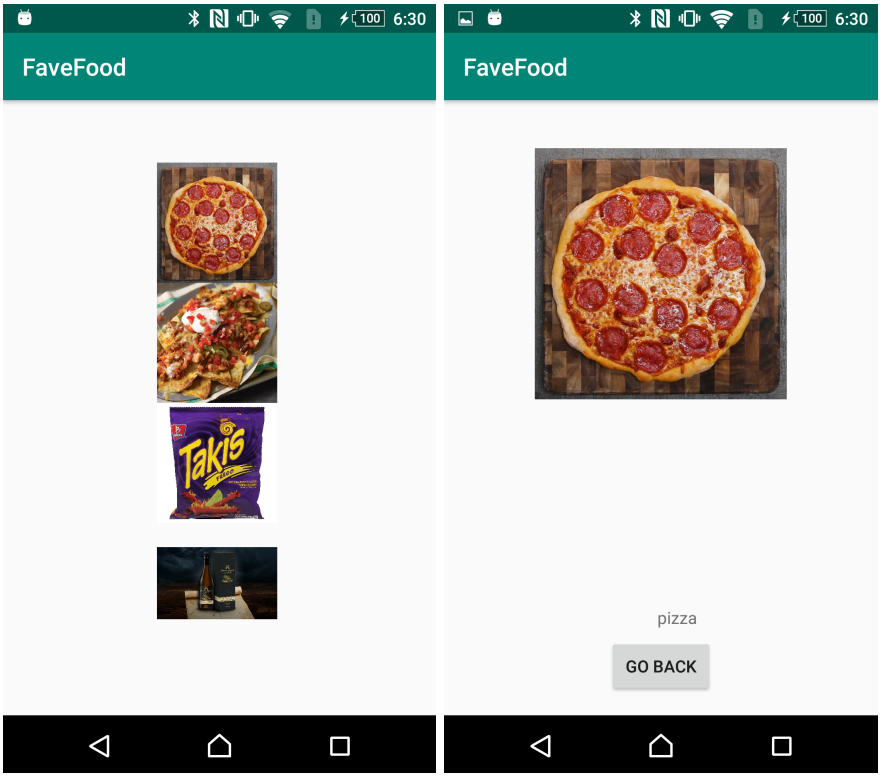

Figure 4: The application in action

Task 2

For this task we have expanded upon the idea of the first task by creating sets of metadata for each of the images that we are then able to edit once we inspect a photo.

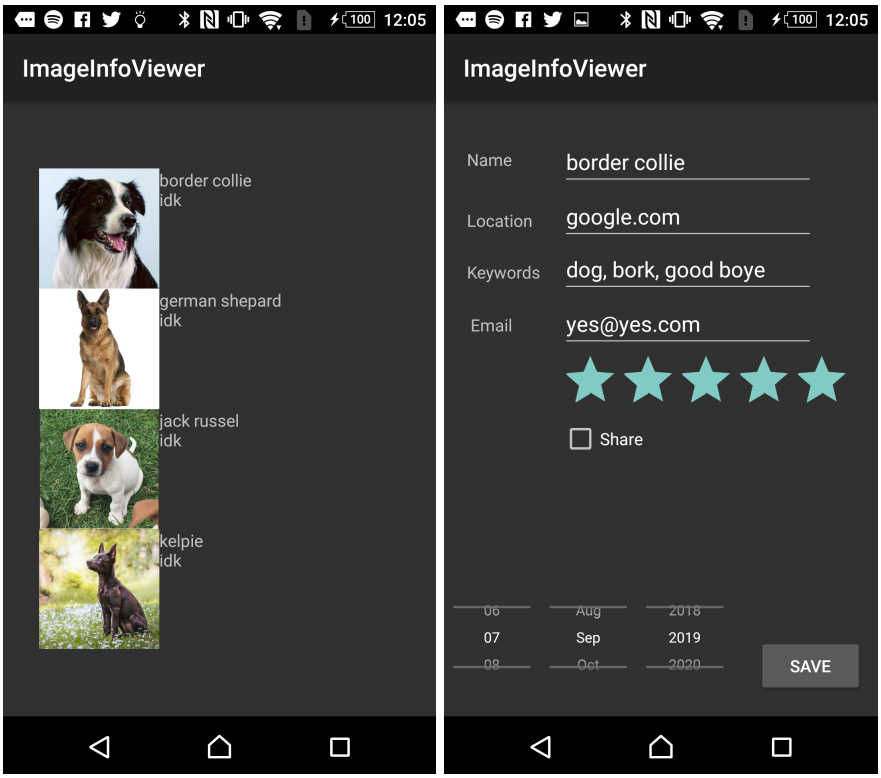

Figure 5: The two activities used for the task

To achieve the dark theme that is present in borht activites, I had to edit the AndroidManifest.xml file which is where many global settings for the application are stored. In this case we just need to edit the android:theme line te equal @style/Theme.AppCompat and then everything else would be handled for us.

```
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="ImageInfoViewer"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/Theme.AppCompat">
```

Figure 6: Part of the file AndroidManifest.xml

The other key part of this task was the use of the parcelable interface to allow for sending of data classes between two activities. The parcelable interface is used to allow classes that contain primitive types to be sent across activities using intents. Through the magic of polymorphism, we are able to implement this interface into our classes to enable them to be accepted as arguments when being added as an extra to an intent. From there, you are able to then access them as normal within the other activity.

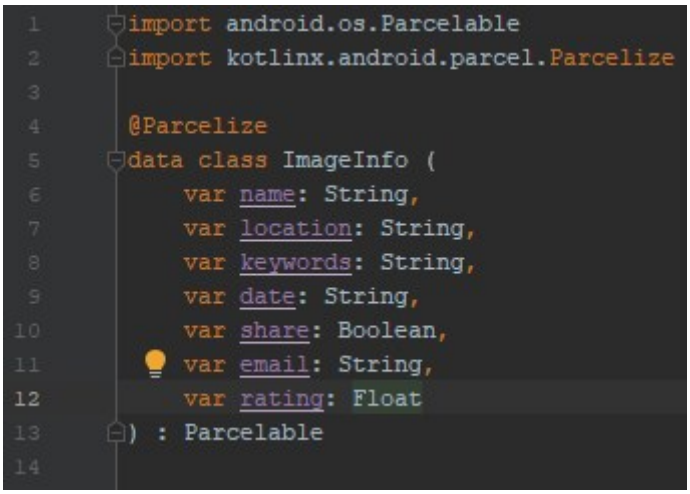

Figure 7: The data class used in this task, which implements the parcelable interface