Queens University Belfast

Android Studio

CSC3054 / CSC7054

Sharing an Image and Link via Email

Book 2

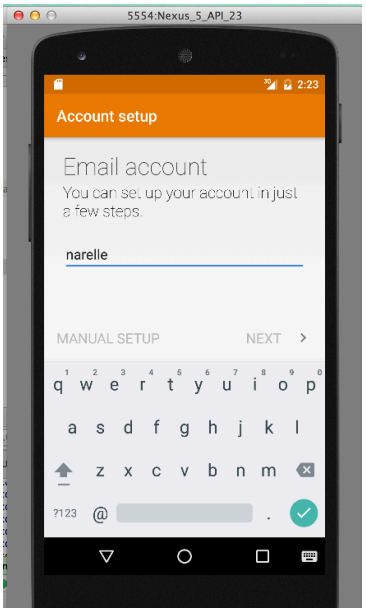
Queens University Belfast

Exercise – Sharing links and images via email

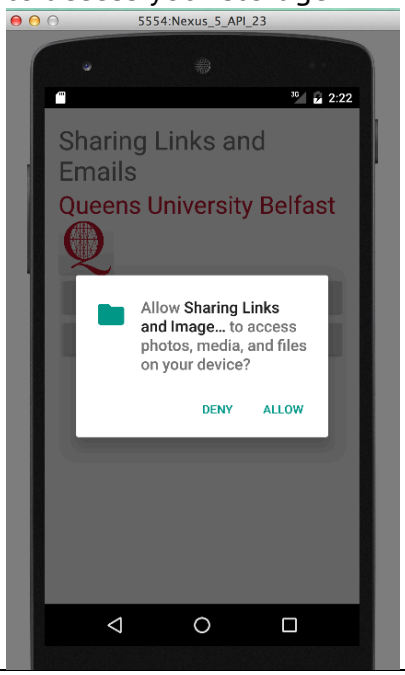
This tutorial will show you how to share a link or image via email using intents and permissions.

Final Product

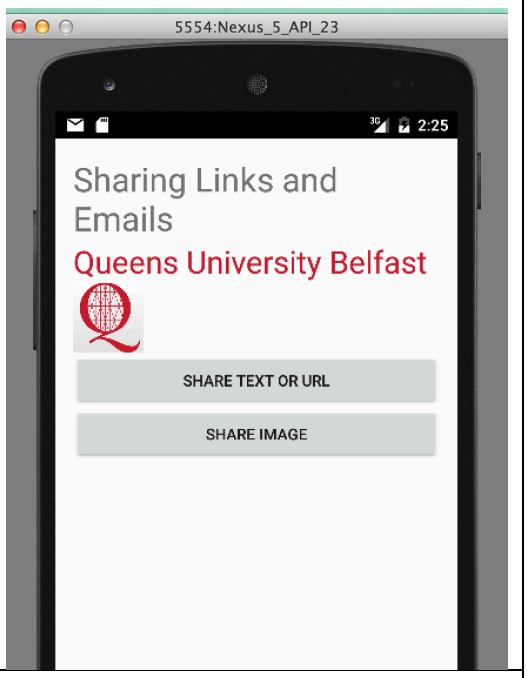
First set up an e-mail account on the emulator



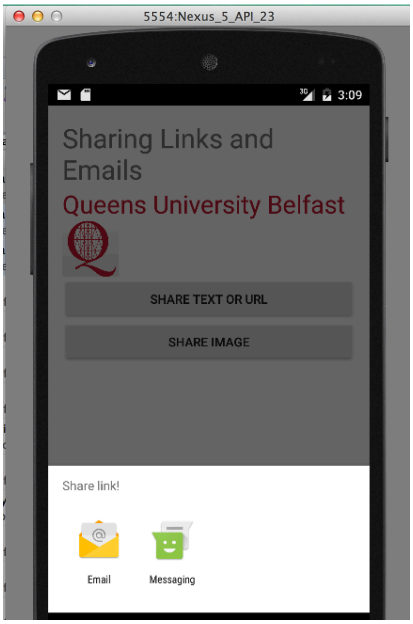
Next, give the app permission to access your storage



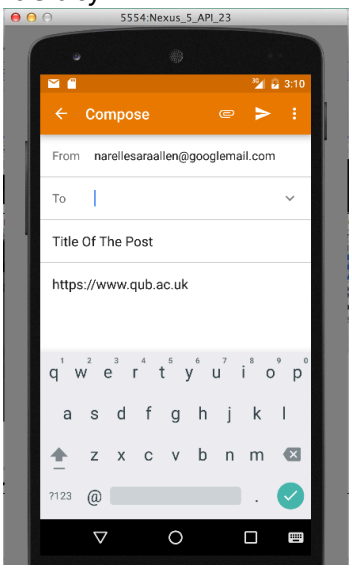
Open the app



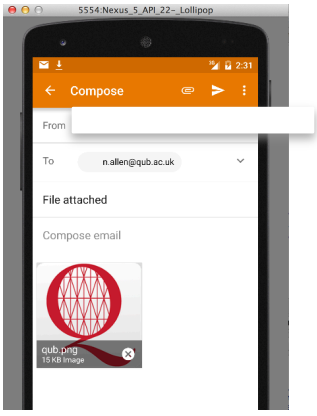
Test the first button



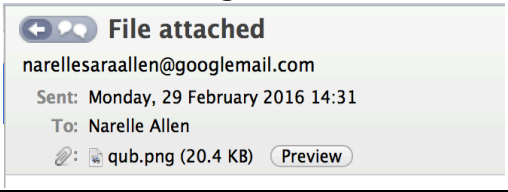
Link pastes into e-mail facility



Test the Second Button



Successful image received



Before You Begin

Open `Android Studio` and create a new project called "`SharingLinkImage`". Refer to the '`Creating your first project`' tutorial to help you create a project. Once created your project should look like figure 1. Switch from `Design` view to Text `view` .

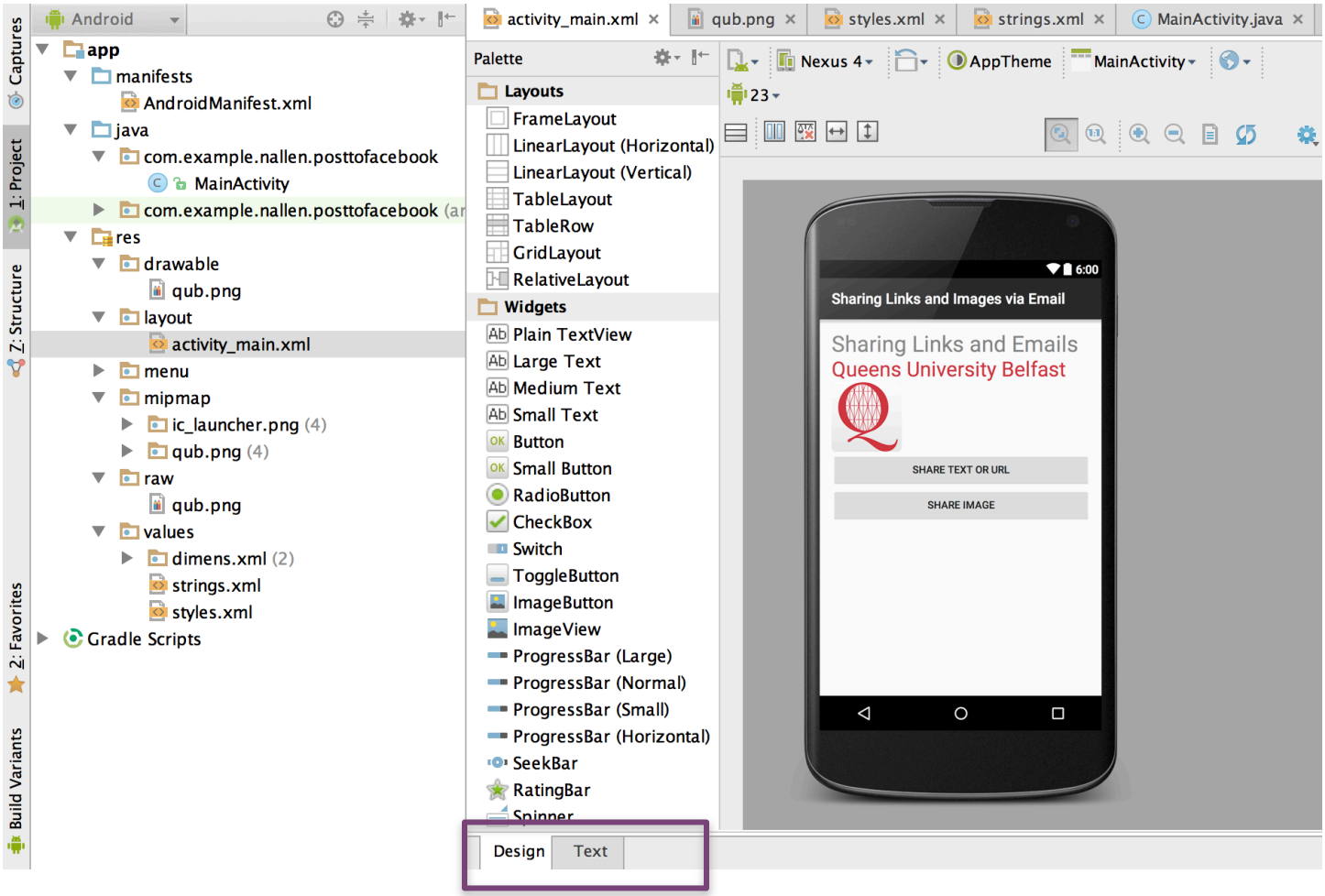


FIGURE 1 - OPEN PROJECT

Step 1 Create the layout

Using figure 1 as a guide, please create the layout shown for `activity_main.xml`. If you have any difficulty, the full code is listed below:

```
<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"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    tools:context=".MainActivity"
    android:orientation="vertical">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/textView"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:textSize="30dp"
        android:text="Sharing Links and Emails " />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Queens University Belfast"
        android:id="@+id/textView2"
        android:layout_below="@+id/textView"
        android:layout_centerHorizontal="true"
        android:textSize="28dp"
        android:textColor="#c42032" />

    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/imageView"
        android:layout_below="@+id/textView2"
        android:layout_centerHorizontal="true"
        android:src="@mipmap/qub"/>


    <Button
        android:id="@+id/buttonShareTextUrl"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:text="Share Text or URL" />

    <Button
        android:id="@+id/buttonShareImage"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
```

```
        android:layout_below="@+id/buttonShareTextUrl"
        android:text="Share Image" />


</LinearLayout>
```

Step 2 Add images

For this app, you need to add the qub.png into several folders as shown in figure 2. Please note that the QUB image MUST be in `.PNG` format.

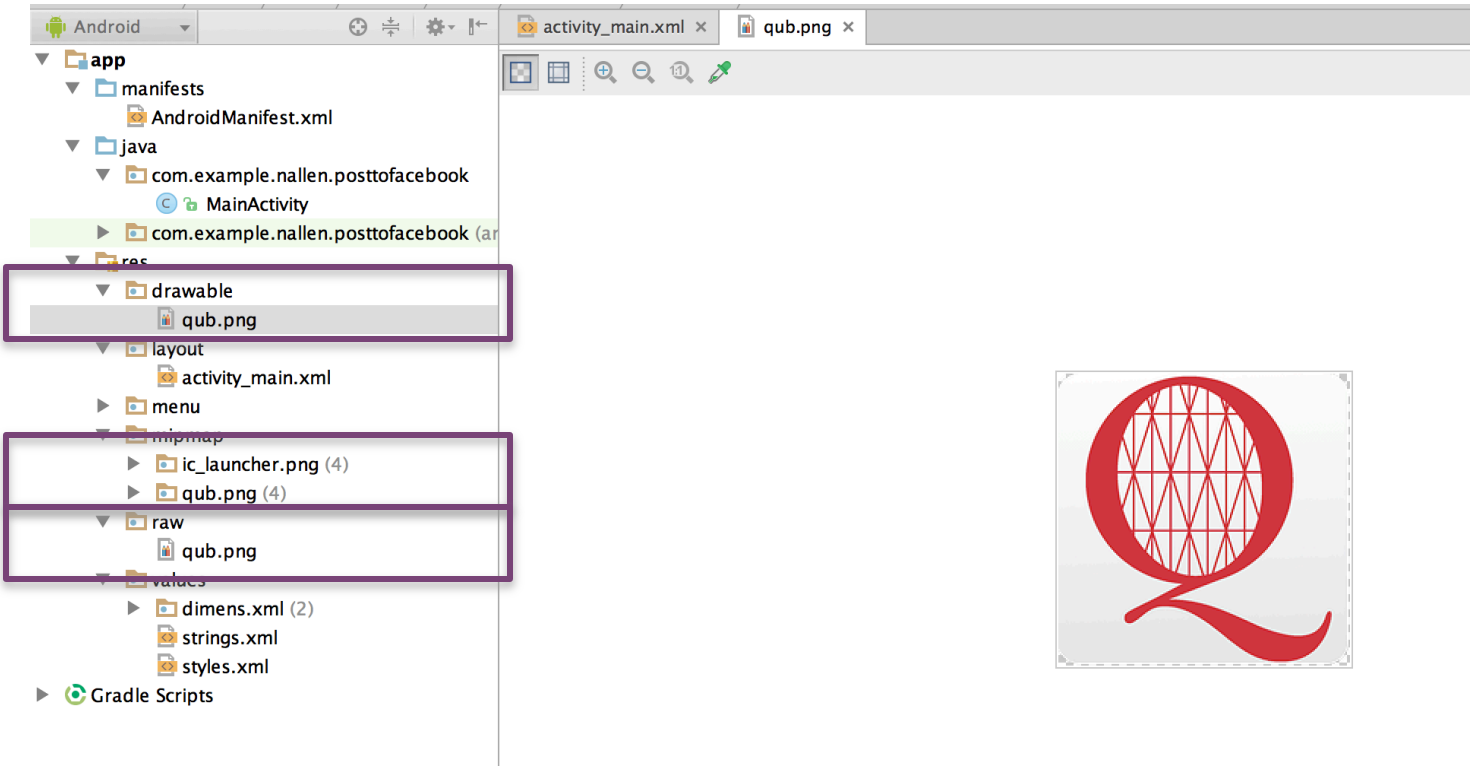


FIGURE 2 ADD IMAGES TO THE CORRECT FOLDER

Step 3 Add Functionality to `MainActivity.java`

For this app you need to request the read/write permissions even if they are already in your manifest. Just to clarify, in order for this to work, the activity must handle the activity permissions request response.

```
import android.Manifest;
import android.content.pm.PackageManager;
import android.os.Bundle;
import android.support.v4.app.ActivityCompat;
import android.util.Log;
import android.content.Intent;
import android.net.Uri;
import android.view.View;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.File;
import java.io.OutputStream;
import android.os.Environment;
import android.app.Activity;

public class MainActivity extends Activity {

    private static final String TAG = "My Activity";
    InputStream in = null;
    OutputStream out = null;

    // Storage Permissions
    private static final int REQUEST_EXTERNAL_STORAGE = 1;
    private static String[] PERMISSIONS_STORAGE = {
            Manifest.permission.READ_EXTERNAL_STORAGE,
            Manifest.permission.WRITE_EXTERNAL_STORAGE
    };
    /**
     * Checks if the app has permission to write to device storage
     * If the app does not has permission then the user will be prompted to grant
      permissions
     *
     * @param activity
     */
    public static void verifyStoragePermissions(Activity activity) {
        // Check if we have write permission
        int permission = ActivityCompat.checkSelfPermission(activity,
                           Manifest.permission.WRITE_EXTERNAL_STORAGE);

        if (permission != PackageManager.PERMISSION_GRANTED) {
            // We don't have permission so prompt the user
            ActivityCompat.requestPermissions(
                    activity,
                    PERMISSIONS_STORAGE,
                    REQUEST_EXTERNAL_STORAGE
            );
        }
    }
```

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    verifyStoragePermissions(this);
    setContentView(R.layout.activity_main);

    // listeners of our two buttons
    View.OnClickListener handler = new View.OnClickListener() {
        public void onClick(View v) {
            switch (v.getId()) {

                case R.id.buttonShareTextUrl:
                    shareTextUrl();
                    break;

                case R.id.buttonShareImage:
                    shareImage();
                    break;
            }
        }
    };

    // our buttons
    findViewById(R.id.buttonShareTextUrl).setOnClickListener(handler);
    findViewById(R.id.buttonShareImage).setOnClickListener(handler);
}

private void shareImage()
{
    try {
        //create an input stream and get the image from the drawable folder
        in = getResources().openRawResource(+ R.drawable.qub);
        // create an output stream copying the file onto the emulated storage
        out = new FileOutputStream(new
                File(Environment.getExternalStorageDirectory(), "qub.png"));
        copyFile(in, out);
        in.close();
        in = null;
        out.flush();
        out.close();
        out = null;
    } catch (Exception e) {
        Log.e("tag", e.getMessage());
        e.printStackTrace();
    }
    //create an intent and add extra information to it
    Intent emailIntent = new Intent(Intent.ACTION_SEND);
    emailIntent.setType("text/html");
    emailIntent.putExtra(Intent.EXTRA_SUBJECT, "File attached");
    Uri uri = Uri.fromFile(new File(Environment.getExternalStorageDirectory(),
            "qub.png"));
    emailIntent.putExtra(Intent.EXTRA_STREAM, uri);
    startActivity(Intent.createChooser(emailIntent, "Send mail..."));

}
```

```
    // Method to share either text or URL.
    private void shareTextUrl() {
        Intent share = new Intent(android.content.Intent.ACTION_SEND);
        share.setType("text/plain");
        share.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK);
        share.putExtra(Intent.EXTRA_SUBJECT, "Title Of The Post");
        share.putExtra(Intent.EXTRA_TEXT, "https://www.qub.ac.uk");
        startActivity(Intent.createChooser(share, "Share link!"));
    }

    // Method to copy file
    private void copyFile(InputStream in, OutputStream out) throws IOException {

            byte[] buffer = new byte[1024];
            int read;
            while ((read = in.read(buffer)) != -1) {
                out.write(buffer, 0, read);
            }
    }
}
```

Step 4 Update the `AndroidMainfest.xml`

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.example.nallen.posttofacebook" >
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:theme="@style/AppTheme" >
    <activity
        android:name=".MainActivity"
        android:label="@string/app_name" >

        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>

        <intent-filter>
            <action android:name="android.intent.action.SEND" />
            <category android:name="android.intent.category.DEFAULT" />
            <data android:mimeType="image/*" />
        </intent-filter>
    </activity>
</application>
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
</manifest>
```