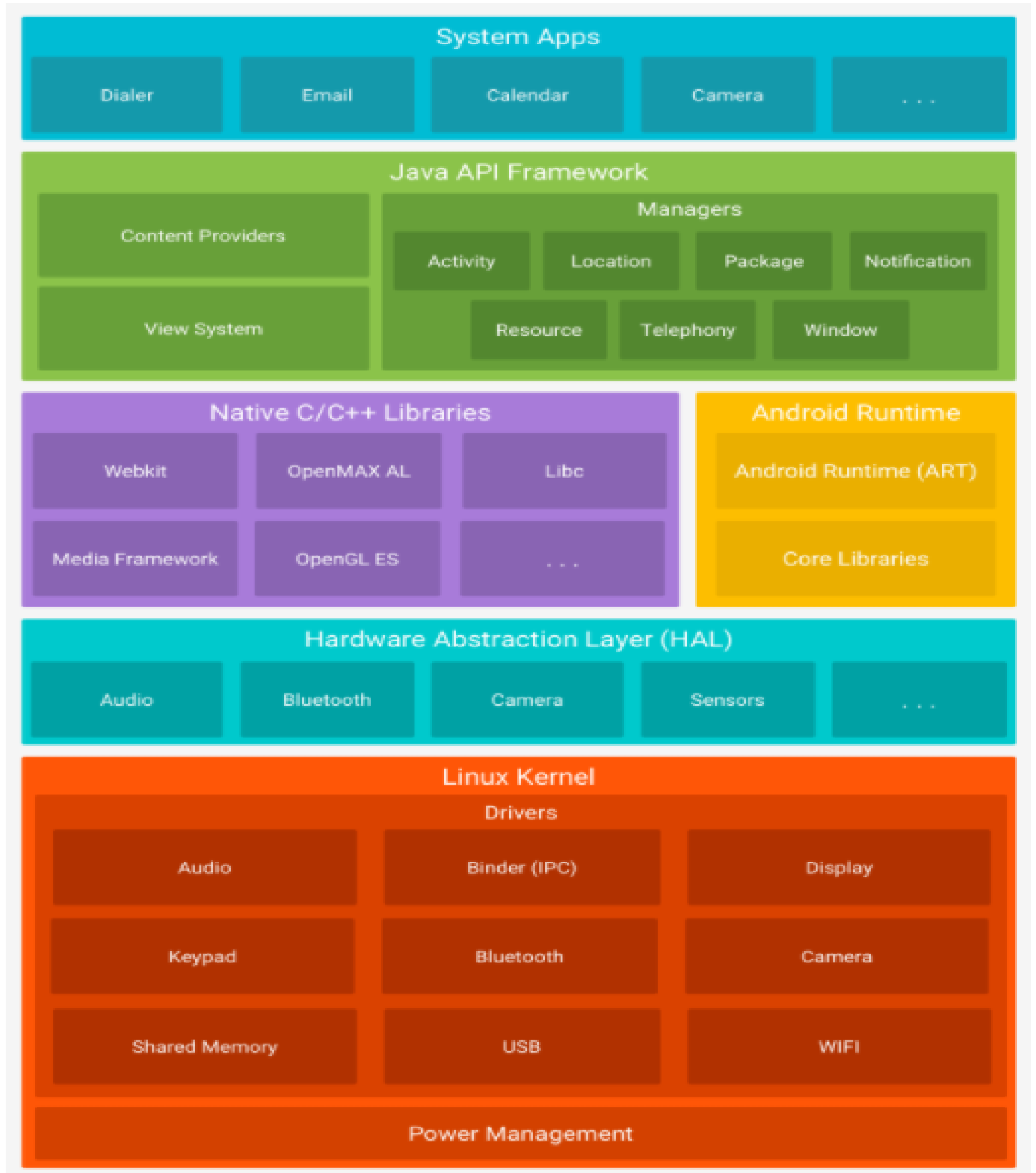
Android Introduction

- Open source and Linux based software stack
- Largest installed OS
- Vendors can customize and add extensions
- Advantages
    - user-friendly
    - huge community support
    - provides a greater extent of customization
    - large number of companies build Android-compatible smartphones
- Unified approach to app development

Android Architecture



List of Android versions

Android 1.0
September 23, 2008

1.5 - Cupcake
April 27, 2009

1.6 - Donut
September 15, 2009

2.0/2.1 - Éclair
October 26, 2009

2.2 - Froyo
May 20, 2010

2.3 - Gingerbread
December 6, 2010

3.0 - Honeycomb
February 22, 2011

4.0 - Ice Cream Sandwich
October 18, 2011

4.1/4.3 - Jelly Bean
July 9, 2012

4.4 - KitKat
October 31, 2013

5.0 - Lollipop
November 12, 2014

6.0 - Marshmallow
October 5, 2015

7.0 - Nougat
August 22, 2016

8.0 - Oreo
August 21, 2017

9.0 - Pie
August 6, 2018

Android 10
September 3, 2019

Android 11
September 8, 2020

Android 12
October 17, 2021

Features of Android

- Storage—SQLite, a lightweight relational database, for data storage
- Connectivity—GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth (includes A2DP and AVRCP), Wi-Fi, LTE and WiMAX
- Messaging—Both SMS and MMS
- Media support H.263, H.264 (in 3GP or MP4 container), MPEG-4 SP, AMR, AMR-WB (in 3GP container), AAC, HE-AAC (in MP4 or 3GP container), MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF and BMP

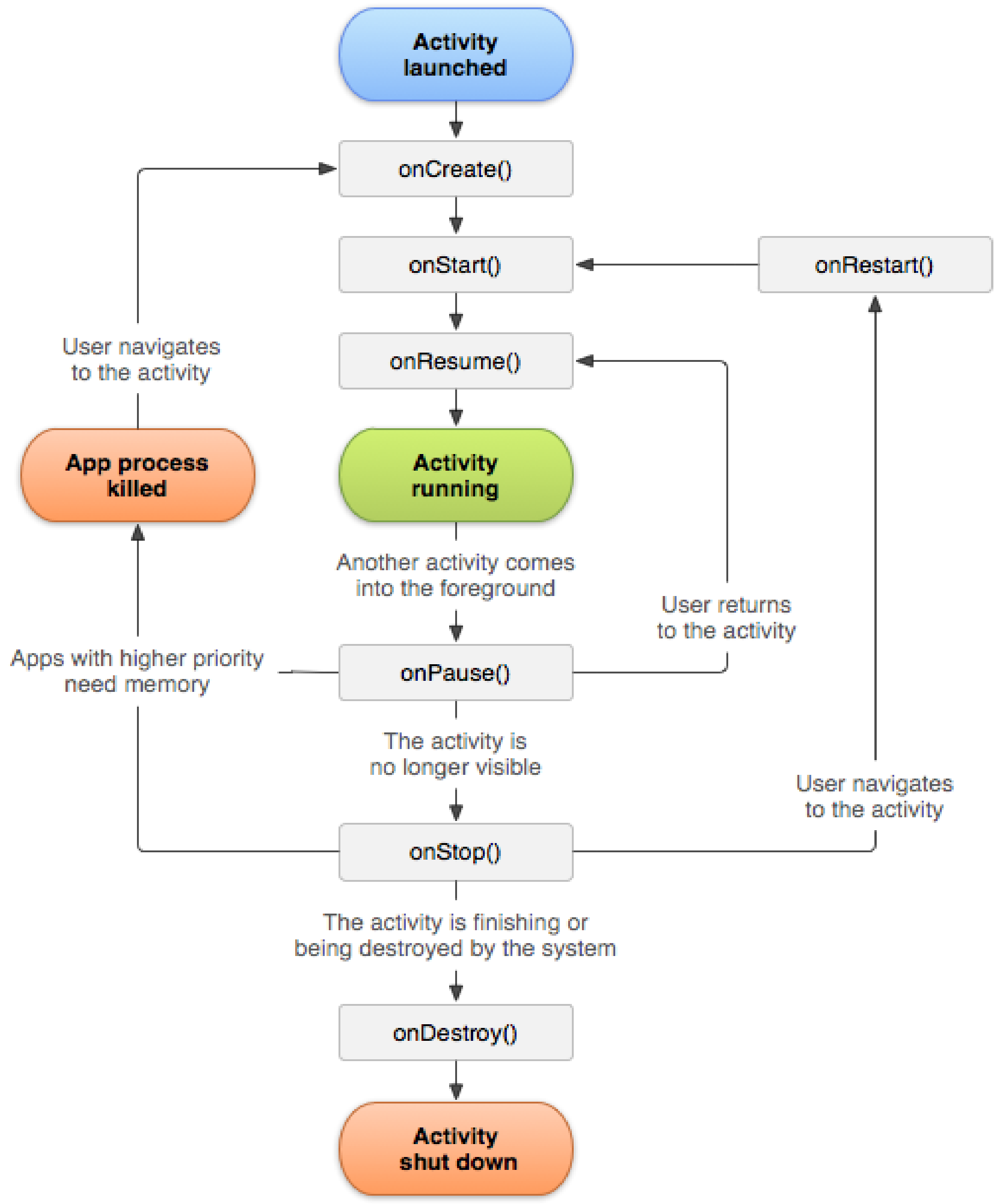
Features of Android

- Hardware support—Accelerometer sensor, camera, digital compass, proximity sensor and GPS
- Multi-touch—Multi-touch screens
- Multi-tasking—Multi-tasking applications
- Tethering—Sharing of Internet connections as a wired/wireless hotspot

Android’s web browser is based on the open source WebKit and Chrome’s V8 JavaScript engine

Activity and Activity Lifecycle

- Serves as the entry point for an app's interaction with the user
- Good implementation of the lifecycle callbacks can help ensure that your app avoids:
  - ➢ Crashing if the user receives a phone call or switches to another app while using your app
  - ➢ Consuming valuable system resources when the user is not actively using it
  - ➢ Losing the user's progress if they leave your app and return to it later
  - ➢ Crashing or losing the user's progress when the screen rotates between landscape and portrait orientation
- Activity class provides a core set of six callbacks
  - ➢ onCreate(), onStart(), onResume(), onPause(), onStop() and onDestroy()


Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

Calls in activity life cycle

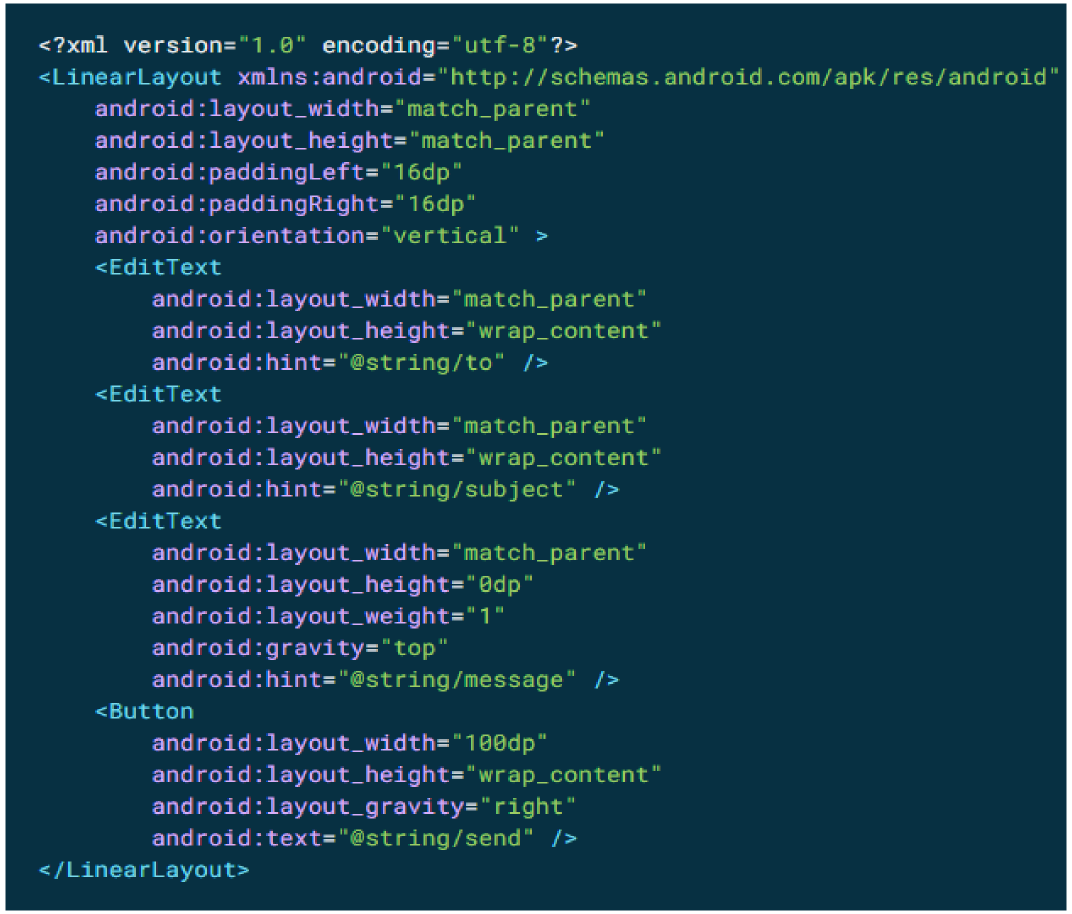
- **onCreate() –** Called when the activity is first created
- **onStart() –** Called just after it’s creation or by restart method after onStop() and here activity start becoming visible to user
- **onResume() –** Called when Activity is visible to user and user can interact with it
- **onPause() –** Called when Activity content is not visible because user resume previous activity
- **onStop() –** Called when activity is not visible to user because some other activity takes place of it
- **onRestart() –** Called when user comes on screen or resume the activity which was stopped
- **onDestroy –** Called when Activity is not in background

Views and ViewGroups

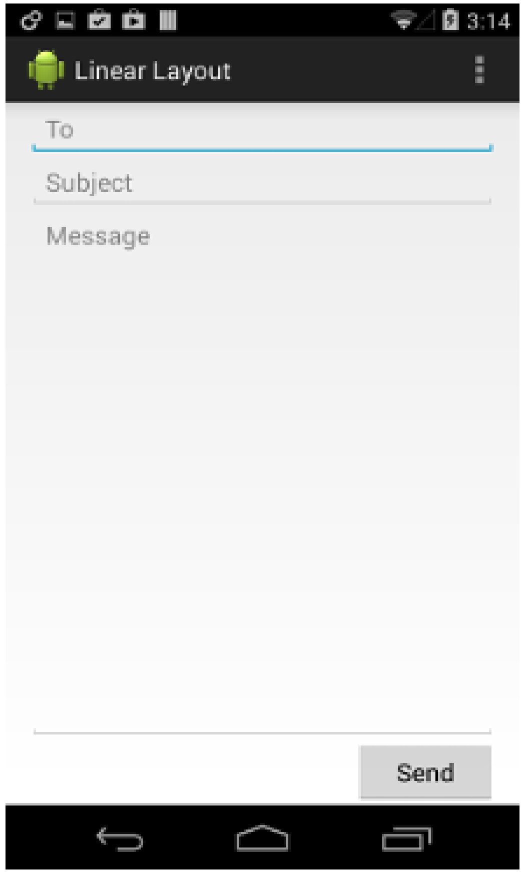
- An activity contains views and ViewGroups
- A view is a widget that has an appearance on screen. Examples of views are buttons, labels, and text boxes
- A view derives from the base class android.view.View
- One or more views can be grouped into a ViewGroup
- A ViewGroup (special type of view) provides the layout in which you can order the appearance and sequence of views
- A ViewGroup derives from the base class android.view.ViewGroup

Linear Layout

- View group that aligns all children in a single direction
- Two types of linear layout orientation
  1. Vertical
  2. Horizontal

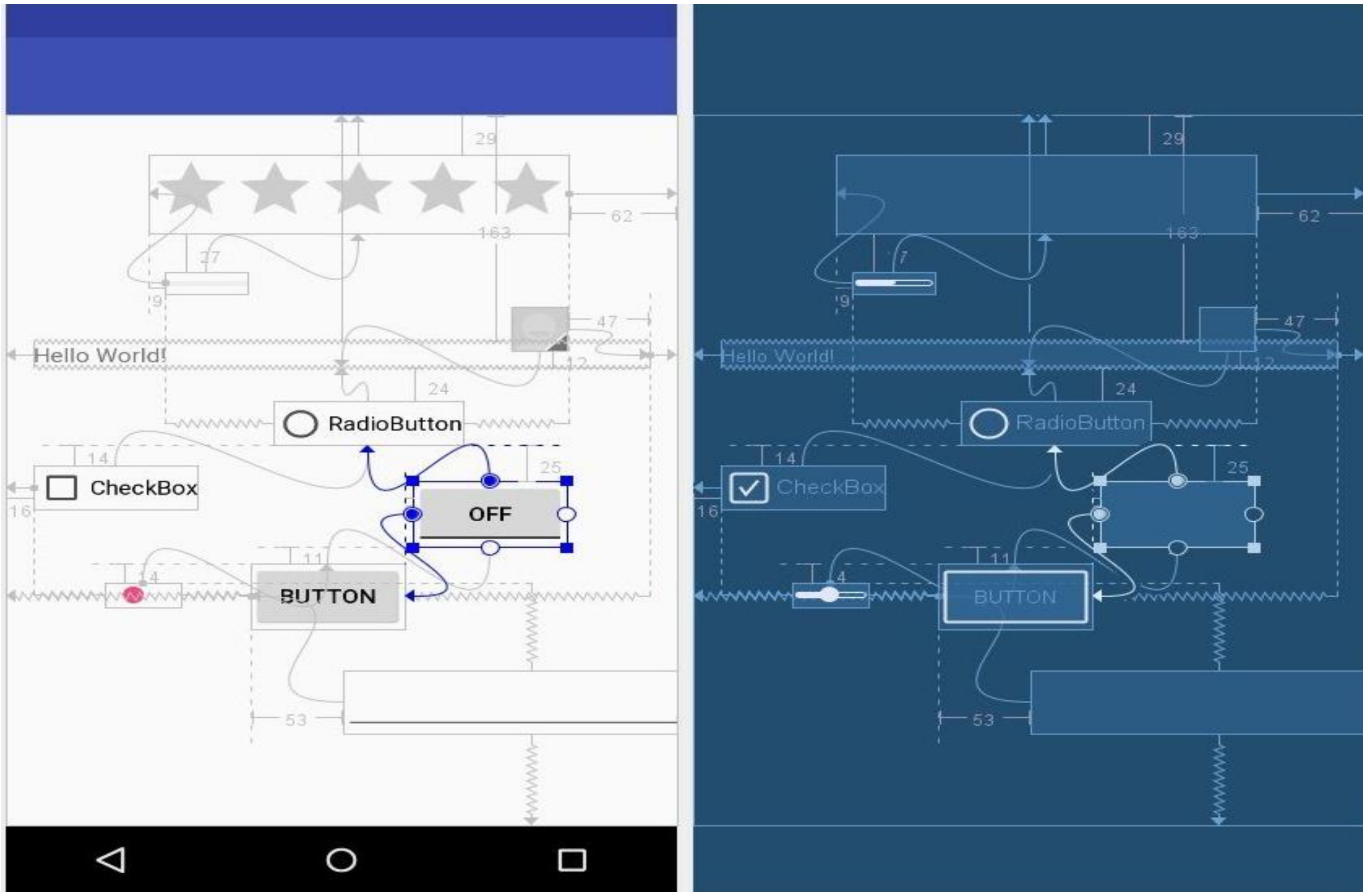


```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:orientation="vertical" >
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/to" />
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/subject" />
    <EditText
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:gravity="top"
        android:hint="@string/message" />
    <Button
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:layout_gravity="right"
        android:text="@string/send" />
</LinearLayout>
```



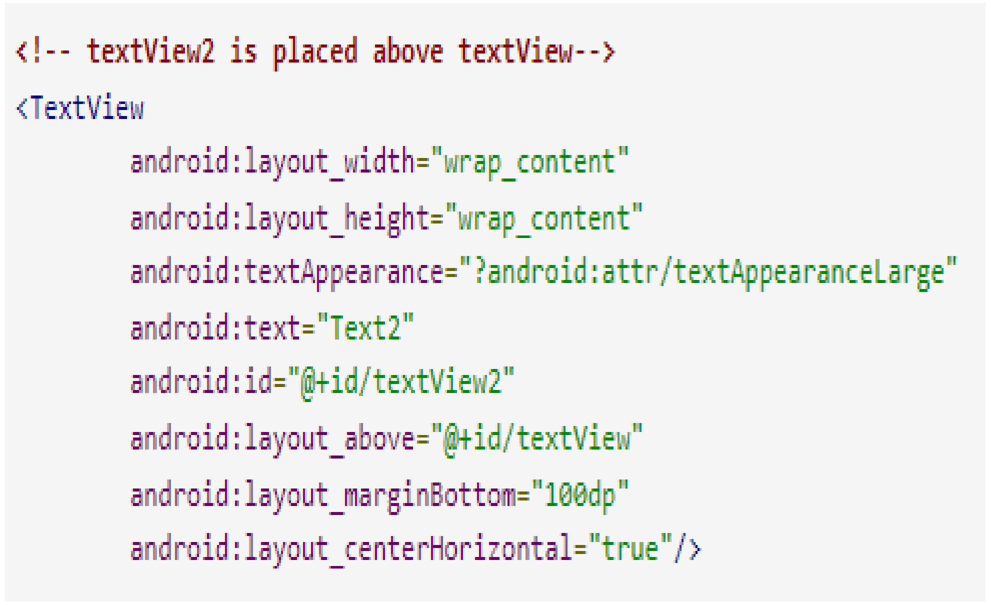
Constraint Layout

- ViewGroup which allows you to position and size widgets in a flexible way



Relative Layout

- Very flexible layout used in android for custom layout designing
- Flexibility to position our component/view based on the relative or sibling component's position



```
<!-- textView2 is placed above textView-->
<TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Text2"
        android:id="@+id/textView2"
        android:layout_above="@+id/textView"
        android:layout_marginBottom="100dp"
        android:layout_centerHorizontal="true"/>
```

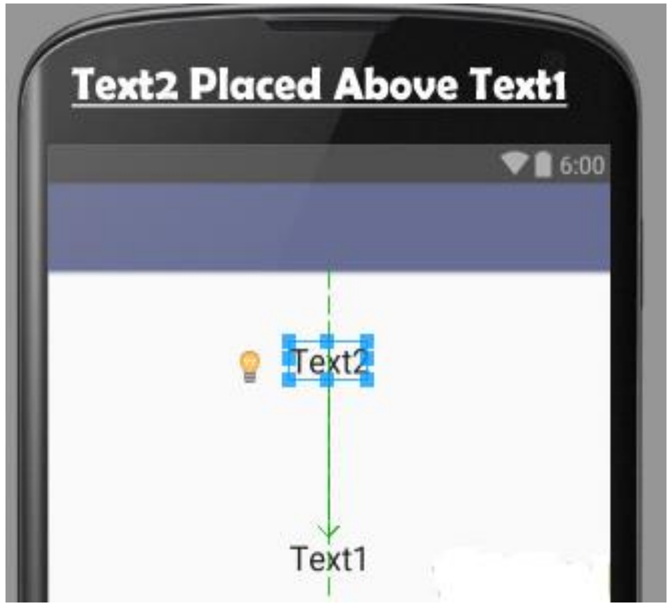


Table Layout

- Arrange the group of views into rows and columns

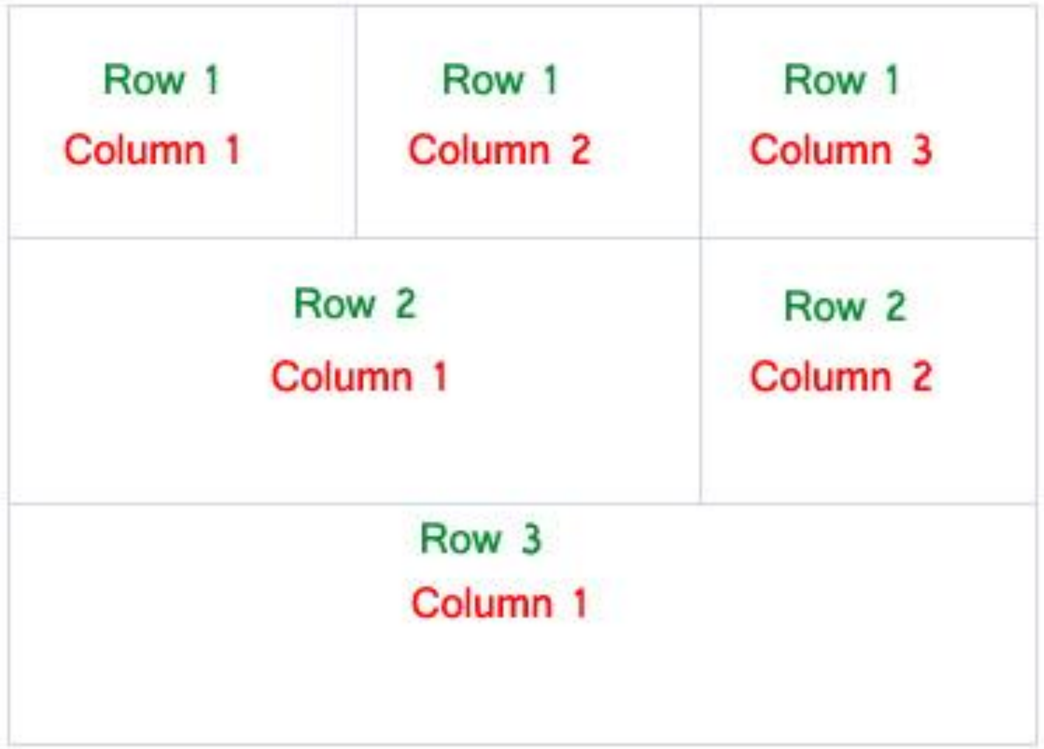
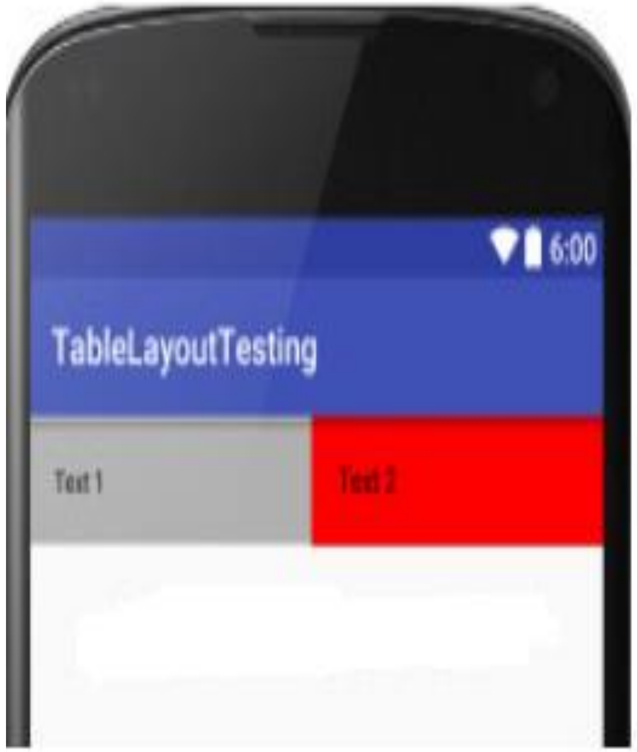


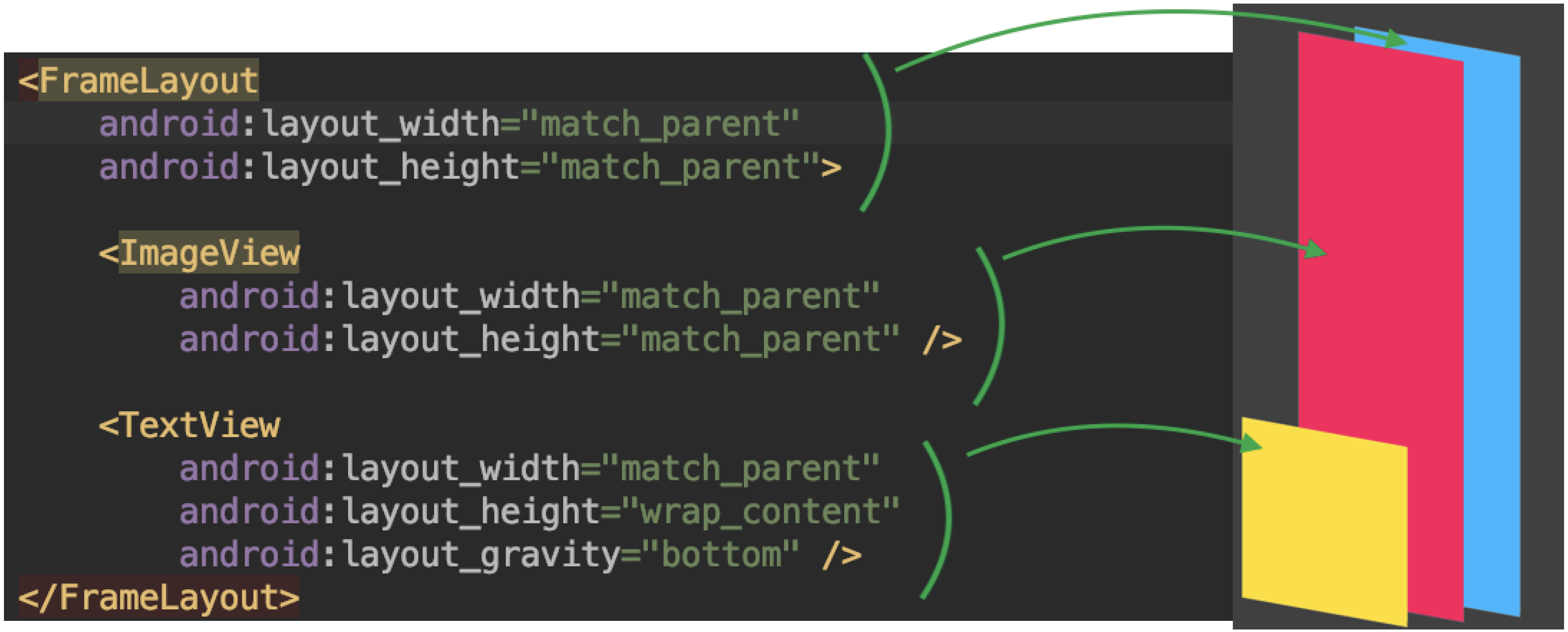
Table Layout

```xml
<?xml version="1.0" encoding="utf-8"?>

    <TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
        android:id="@+id/simpleTableLayout"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:stretchColumns="1"> <!-- stretch the second column of the layout-->

        <!-- first row of the table layout-->
        <TableRow

            android:id="@+id/firstRow"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content">

            <!-- first element of the row-->
            <TextView

                android:id="@+id/simpleTextView"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:background="#b0b0b0"
                android:padding="18dip"
                android:text="Text 1"
                android:textColor="#000"
                android:textSize="12dp" />

            <TextView

                android:id="@+id/simpleTextView"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:background="#FF0000"
                android:padding="18dip"
                android:text="Text 2"
                android:textColor="#000"
                android:textSize="14dp" />

        </TableRow>
    </TableLayout>
```



Frame Layout

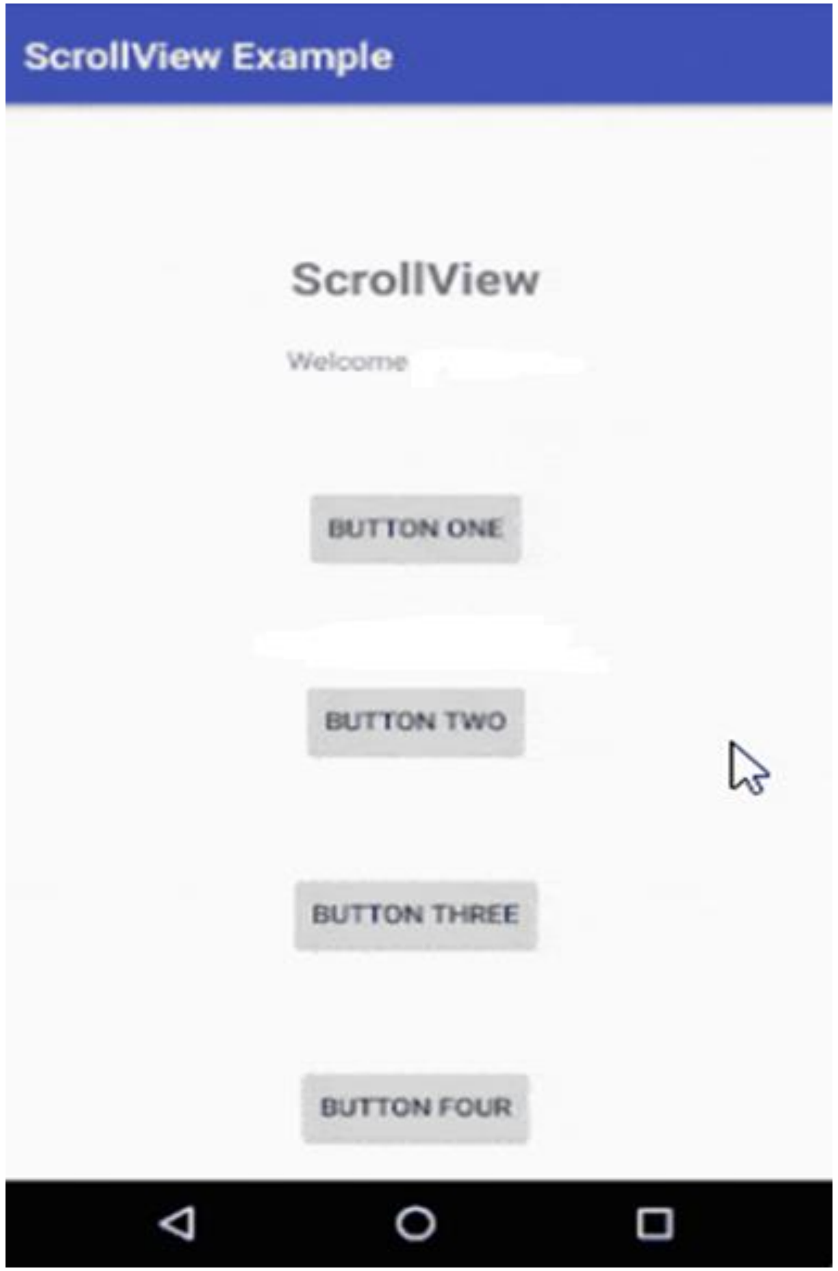
- Designed to block out an area on the screen to display a single item

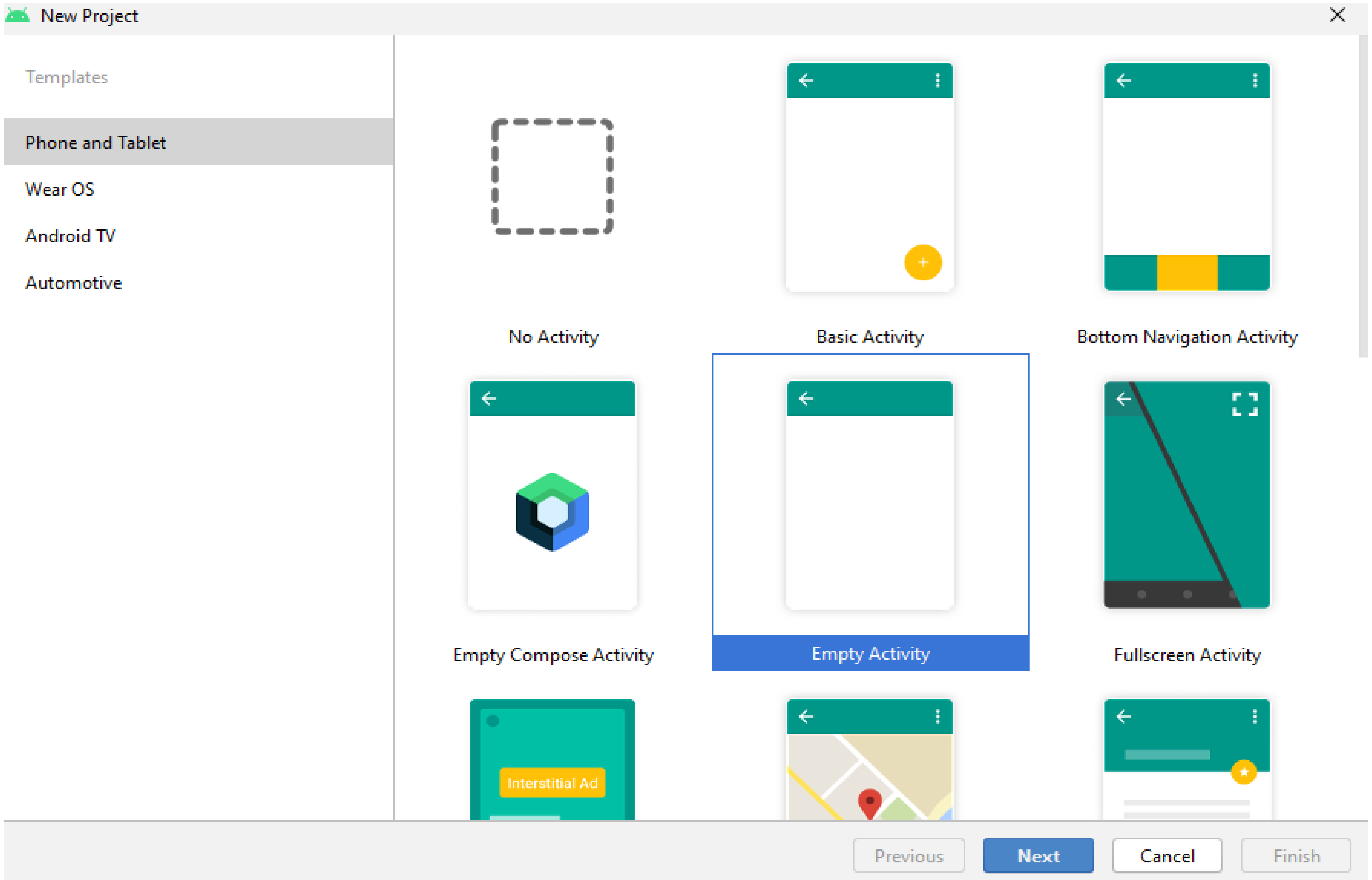


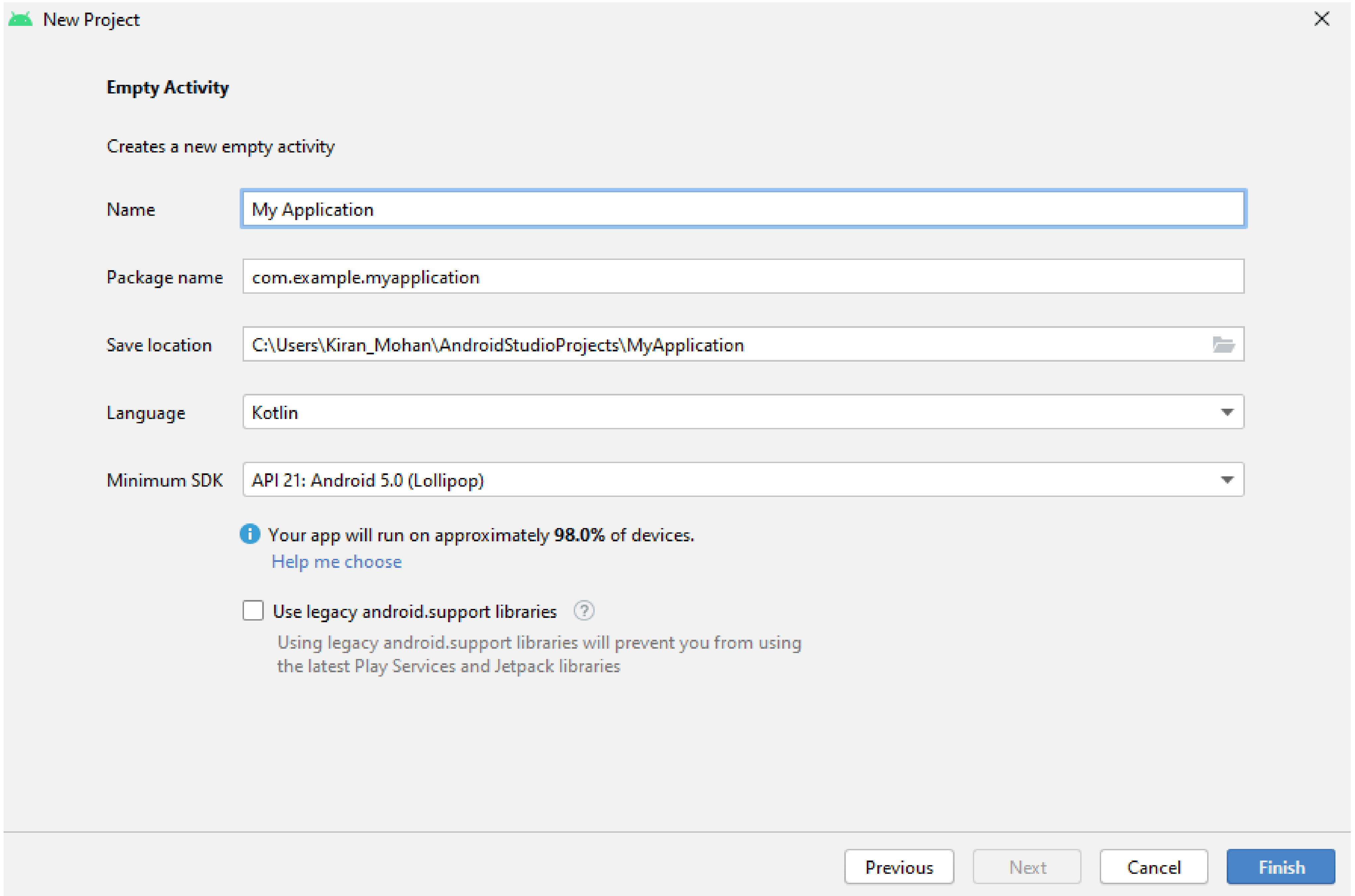
```
<FrameLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <ImageView
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom" />
</FrameLayout>
```

Scroll view

```
<ScrollView xmlns:android="http://schemas.android.
com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:fillViewport="false">
<LinearLayout xmlns:android="http://schemas.androi
d.com/apk/res/android"
    android:orientation="vertical" android:layout_
width="match_parent"
    android:layout_height="match_parent">
    <TextView android:id="@+id/loginscrn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="80dp"
        android:text="ScrollView"
        android:textSize="25dp"
        android:textStyle="bold"
        android:layout_gravity="center"/>
    <TextView android:id="@+id/fstTxt"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:text="Welcome "
        android:layout_gravity="center"/>
    <Button android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="60dp"
        android:text="Button One" />
    <Button android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="60dp"
        android:text="Button Two" />
    <Button android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="60dp"
        android:text="Button Three" />
    <Button android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="60dp"
        android:text="Button Four" />
    <Button android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="60dp"
        android:text="Button Five" />
    <Button android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="60dp"
        android:text="Button Six" />
    <Button android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="60dp"
        android:text="Button Seven" />
    <Button android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="60dp"
        android:text="Button Eight" />
    <Button android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="60dp"
        android:text="Button Nine" />
</LinearLayout>
</ScrollView>
```


ScrollView Example
ScrollView
Welcome
BUTTON ONE
BUTTON TWO
BUTTON THREE
BUTTON FOUR


New Project
Templates
Phone and Tablet
Wear OS
Android TV
Automotive
No Activity
Basic Activity
Bottom Navigation Activity
Empty Compose Activity
Empty Activity
Fullscreen Activity
Interstitial Ad
Previous
Next
Cancel
Finish



New Project

Empty Activity

Creates a new empty activity

Name: My Application

Package name: com.example.myapplication

Save location: C:\Users\Kiran_Mohan\AndroidStudioProjects\MyApplication

Language: Kotlin

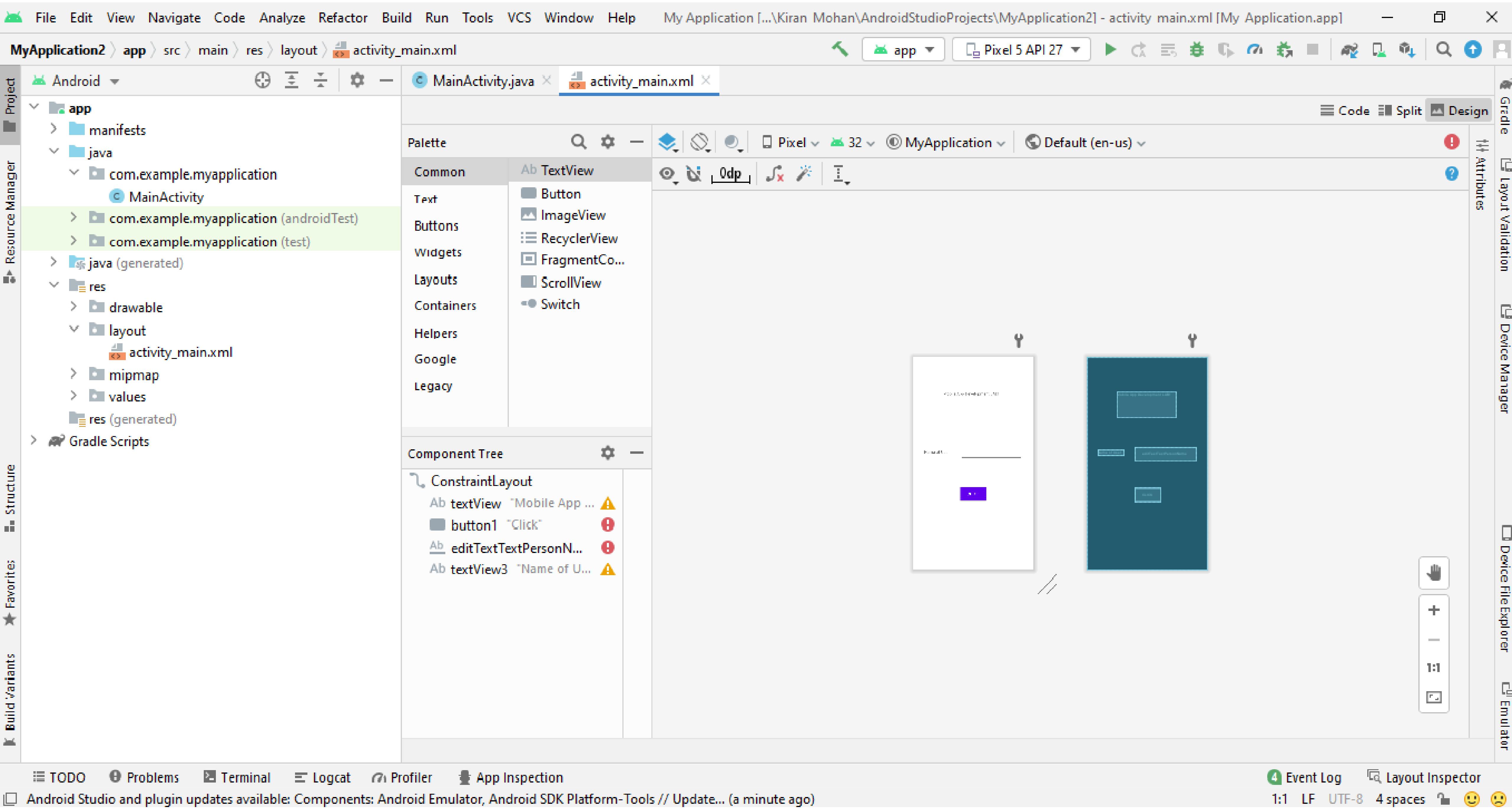
Minimum SDK: API 21: Android 5.0 (Lollipop)

Your app will run on approximately **98.0%** of devices.

Help me choose

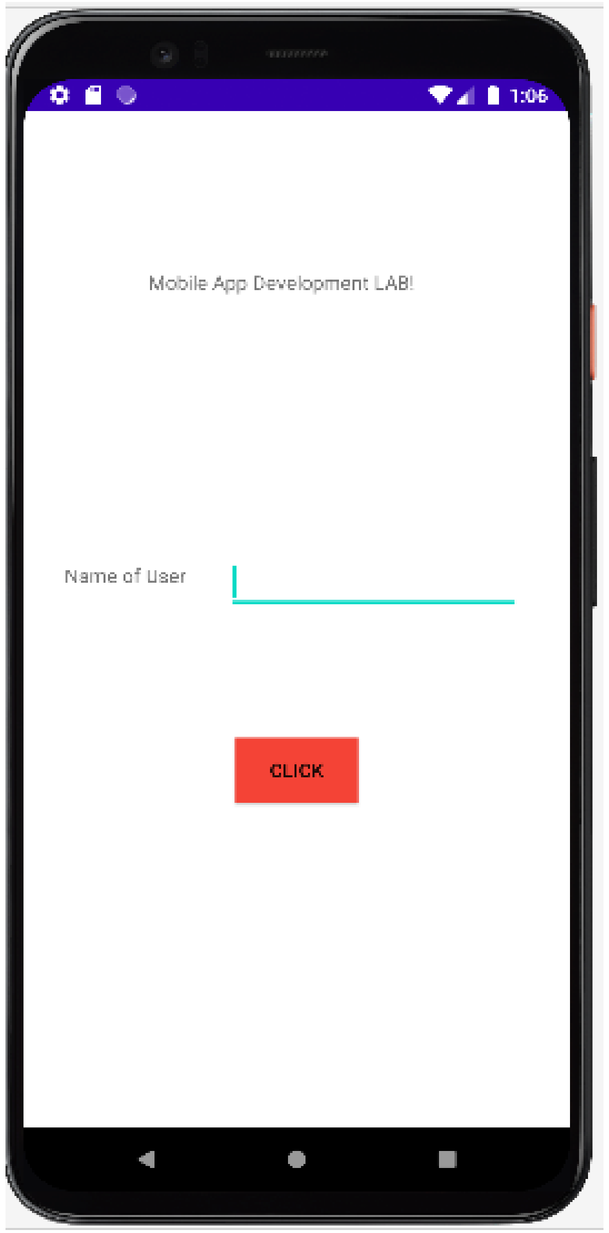
☐ Use legacy android.support libraries

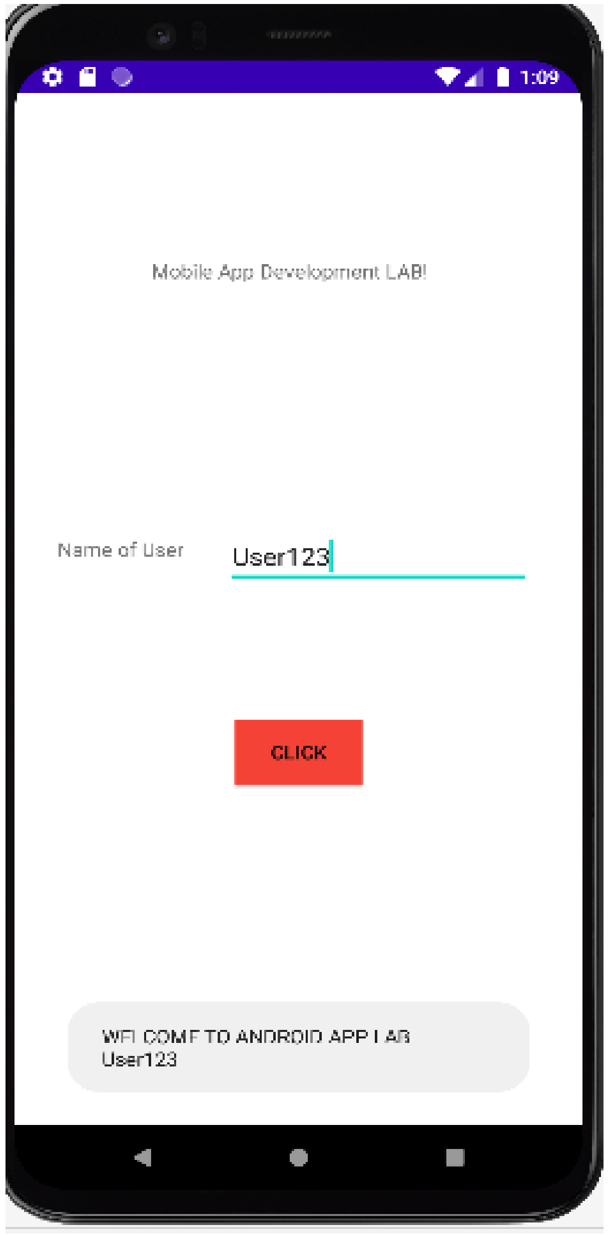
Using legacy android.support libraries will prevent you from using the latest Play Services and Jetpack libraries



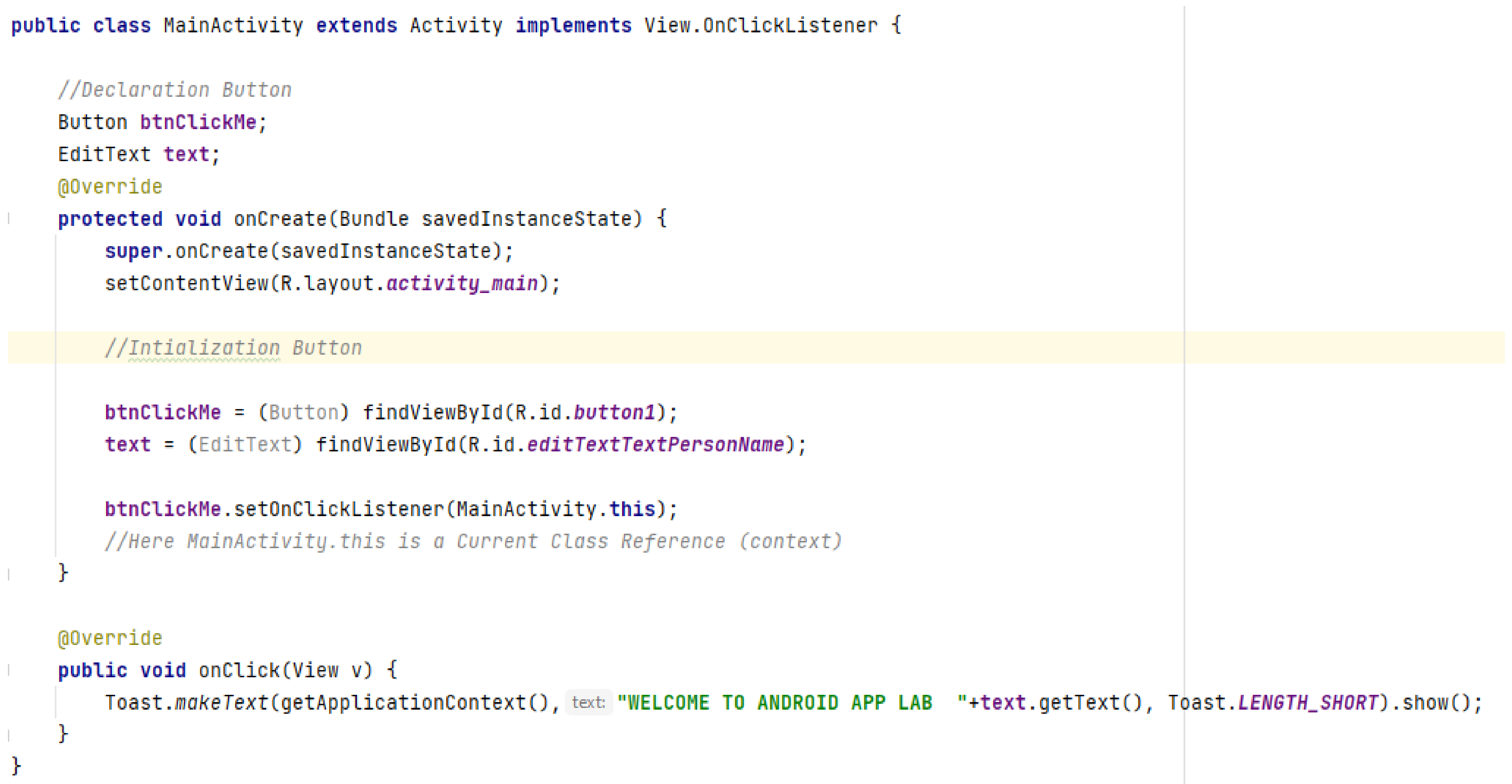
Android Studio and plugin updates available: Components: Android Emulator, Android SDK Platform-Tools // Update... (a minute ago)

1:1 LF UTF-8 4 spaces

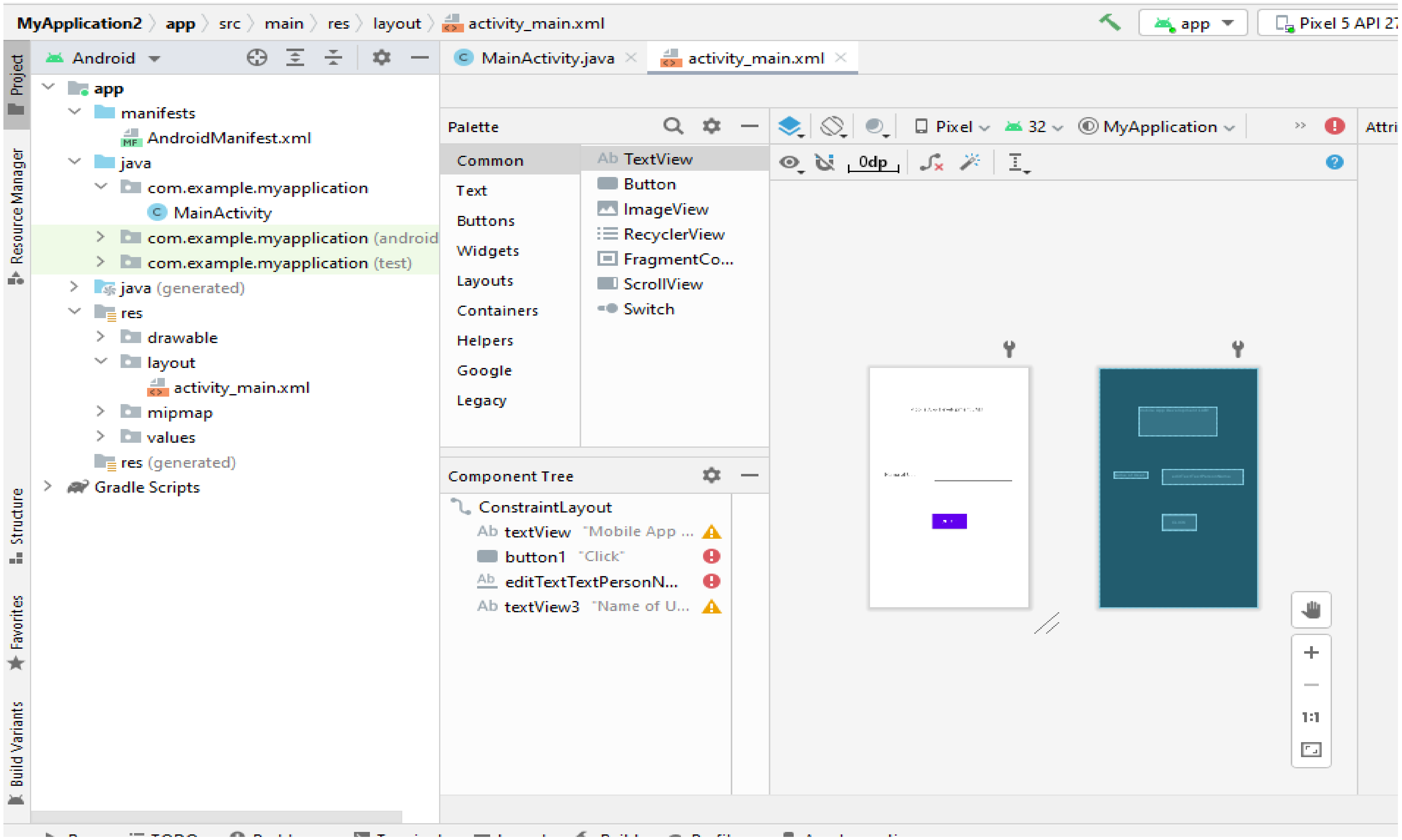




WELCOME TO ANDROID APP LAB
User123



```java
public class MainActivity extends Activity implements View.OnClickListener {

    //Declaration Button
    Button btnClickMe;
    EditText text;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        //Intialization Button

        btnClickMe = (Button) findViewById(R.id.button1);
        text = (EditText) findViewById(R.id.editTextTextPersonName);

        btnClickMe.setOnClickListener(MainActivity.this);
        //Here MainActivity.this is a Current Class Reference (context)
    }

    @Override
    public void onClick(View v) {
        Toast.makeText(getApplicationContext(), text: "WELCOME TO ANDROID APP LAB  "+text.getText(), Toast.LENGTH_SHORT).show();
    }
}
```


MyApplication2 〉 app 〉 src 〉 main 〉 res 〉 layout 〉 activity_main.xml
app
Pixel 5 API 27
Project
Resource Manager
Structure
Favorites
Build Variants
Android
app
manifests
AndroidManifest.xml
java
com.example.myapplication
MainActivity
com.example.myapplication (androidTest)
com.example.myapplication (test)
java (generated)
res
drawable
layout
activity_main.xml
mipmap
values
res (generated)
Gradle Scripts
MainActivity.java
activity_main.xml
Palette
Common
Text
Buttons
Widgets
Layouts
Containers
Helpers
Google
Legacy
TextView
Button
ImageView
RecyclerView
FragmentCo...
ScrollView
Switch
Pixel
32
MyApplication
Attri
0dp
Component Tree
ConstraintLayout
textView "Mobile App ...
button1 "Click"
editTextTextPersonN...
textView3 "Name of U...
1:1

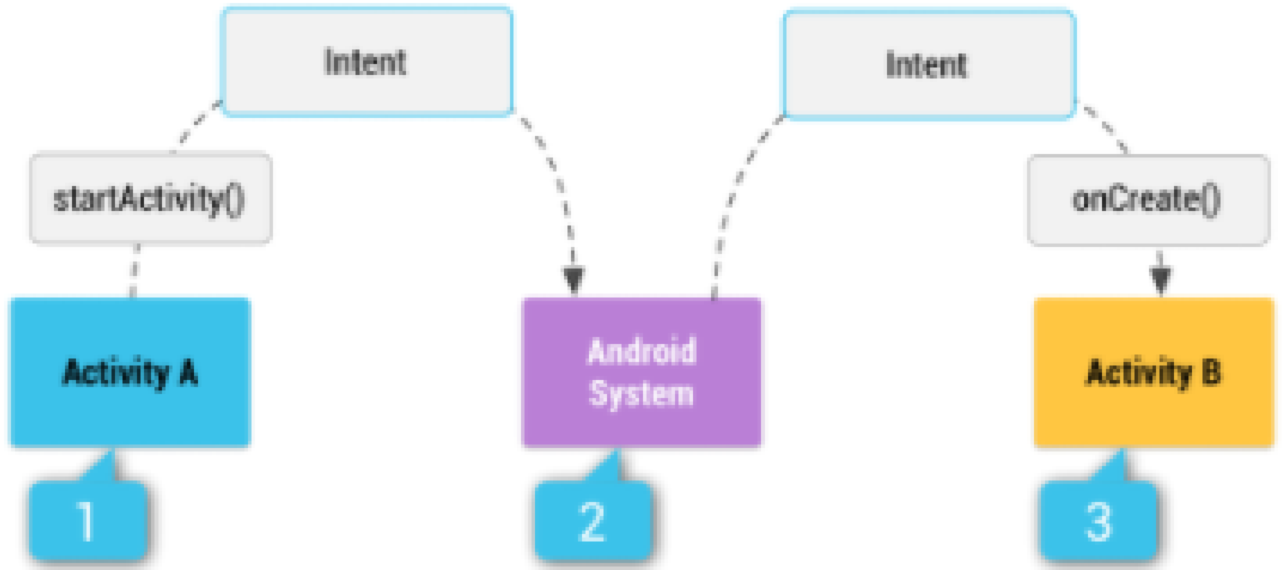
Specifying Permissions in Android App

```
<manifest ... >
    <uses-permission android:name="android.permission.SEND_SMS"/>
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.WAKE_LOCK" />
    <uses-permission android:name="android.permission.CAMERA" />
    <uses-feature android:name="android.hardware.camera" />
    ...
</manifest>
```

Intents and Intent filters

- Messaging object you can use to request an action from another app component
- **Used for**
  - Starting an activity
  - Starting a service
  - Delivering a broadcast
- **Intent types**
  - Explicit intents
  - Implicit intents

Implicit Intents

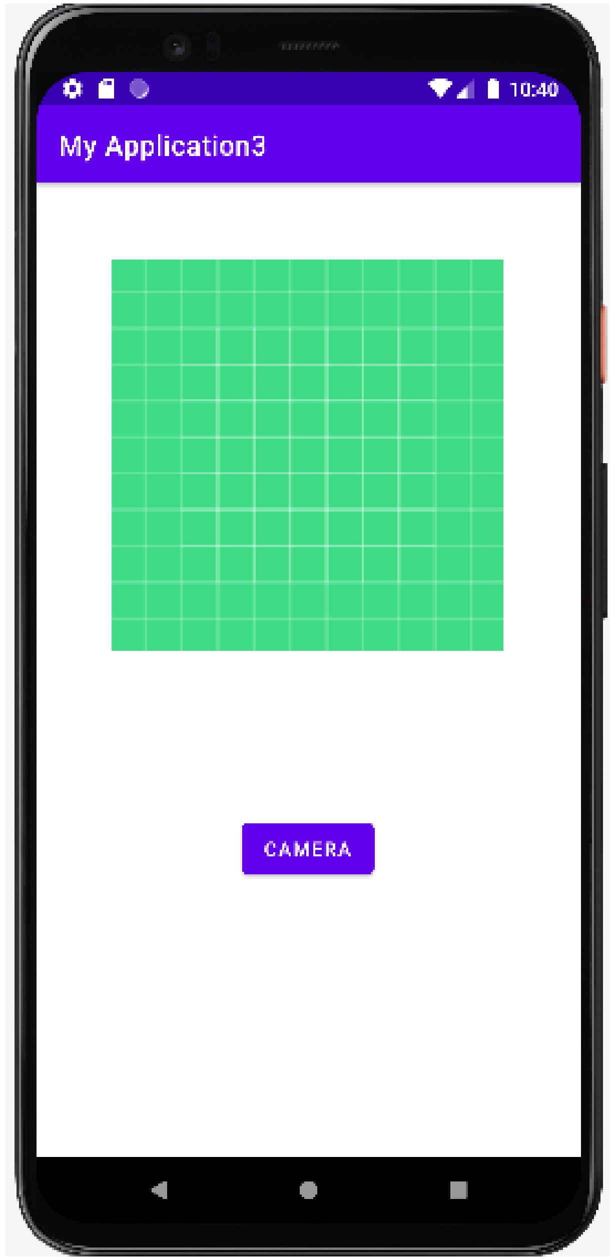


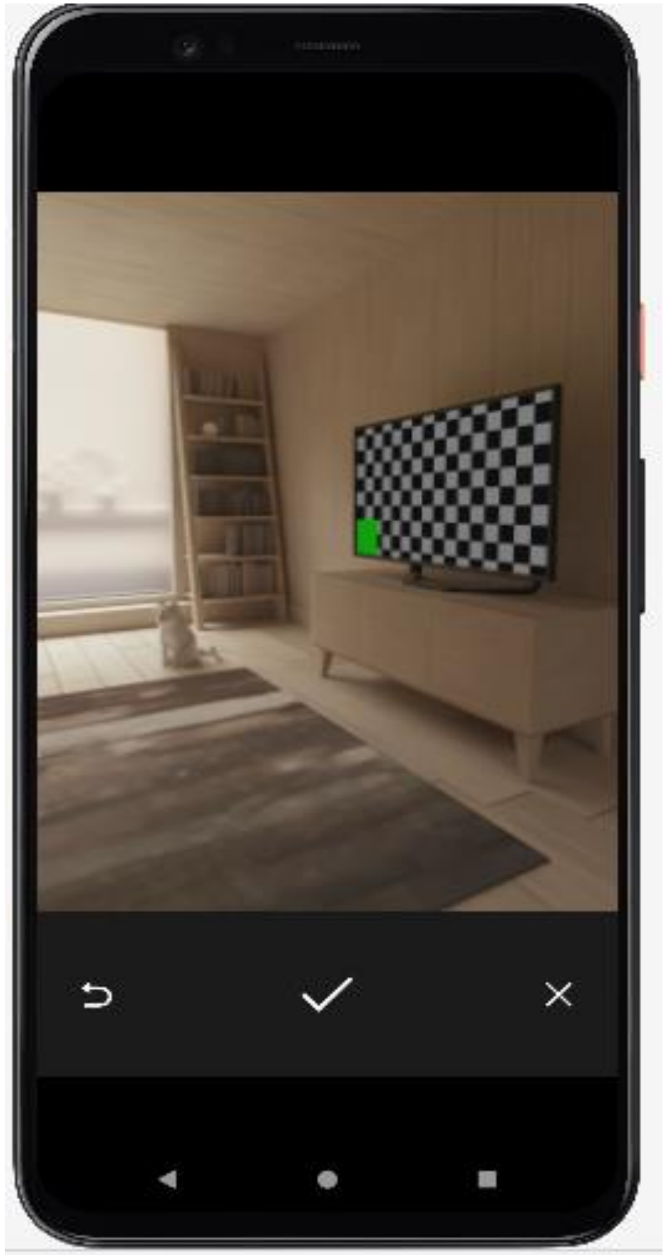
MainActivity.java (First activity)

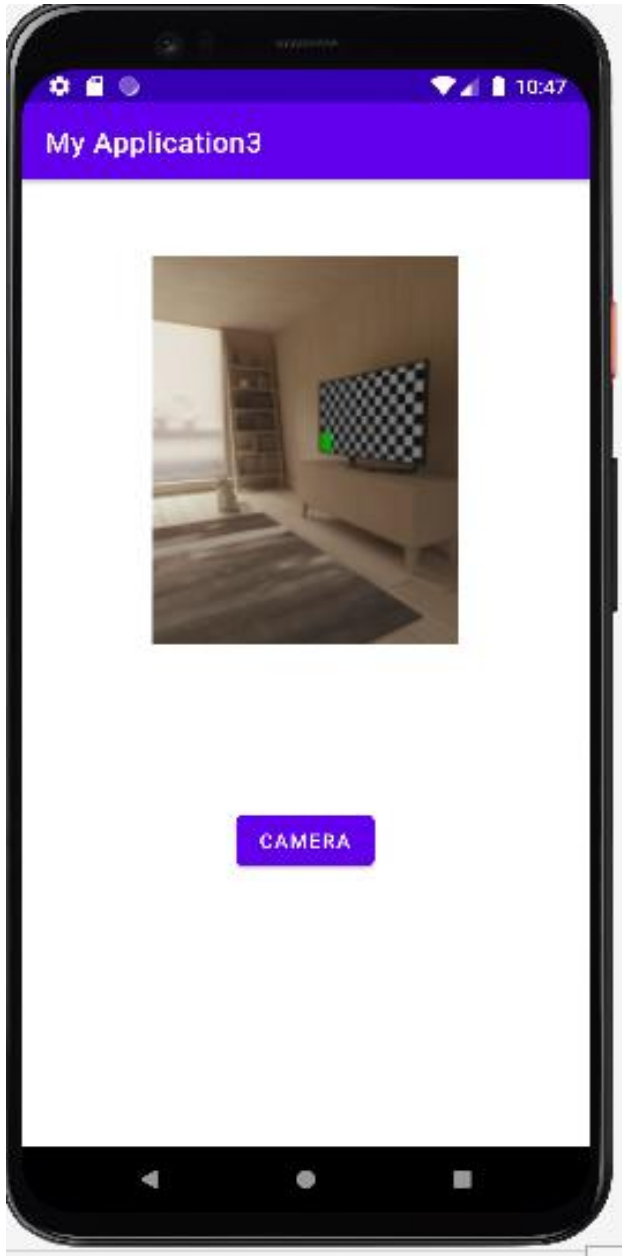
```java
import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
public class MainActivity extends AppCompatActivity {
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      Button send = findViewById(R.id.send);
      send.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
            Intent send = new Intent(MainActivity.this, SecondActivity.class);
            startActivity(send);
         }
      });
   }
}
```

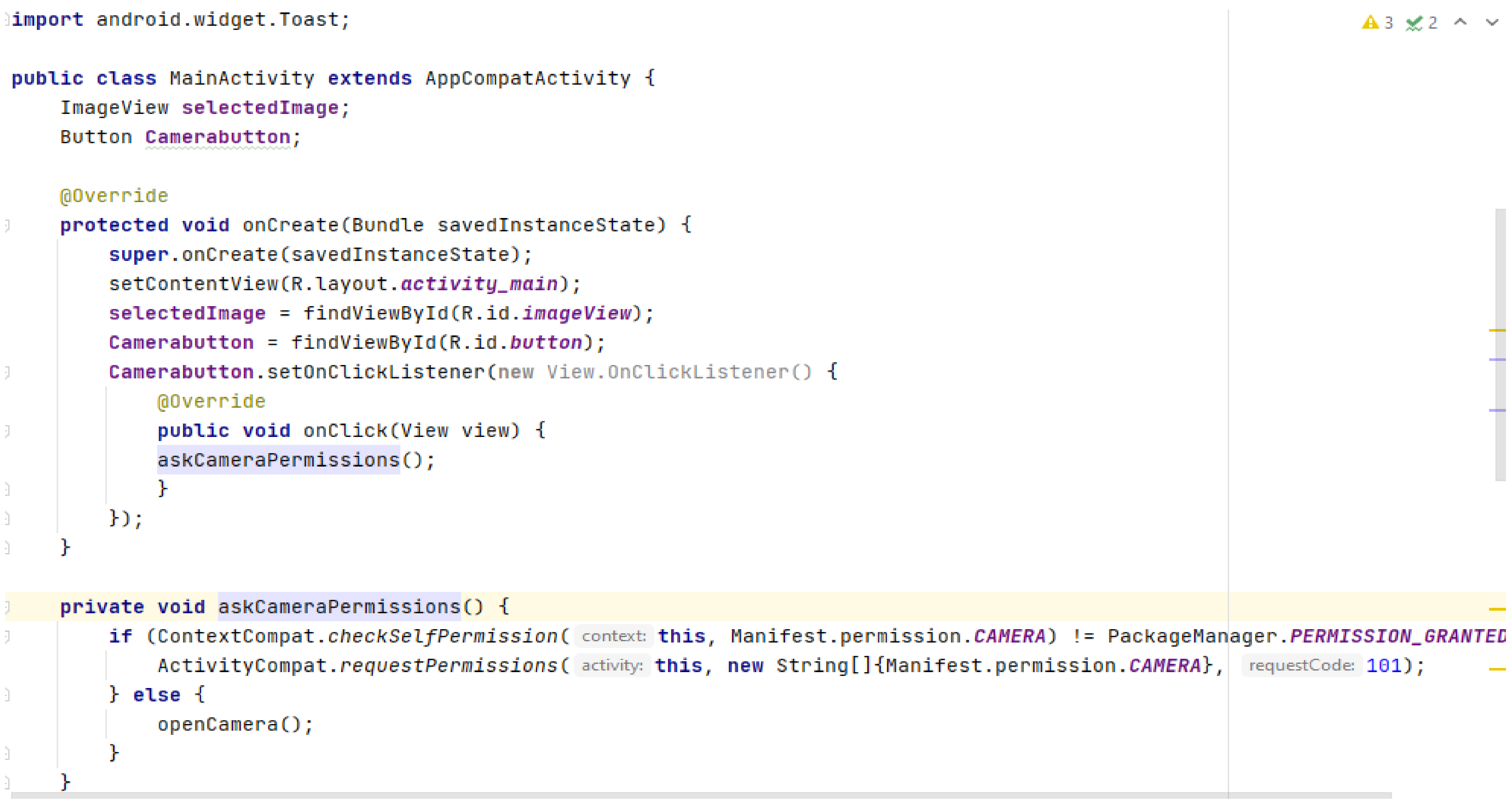
AndroidManifest.xml.

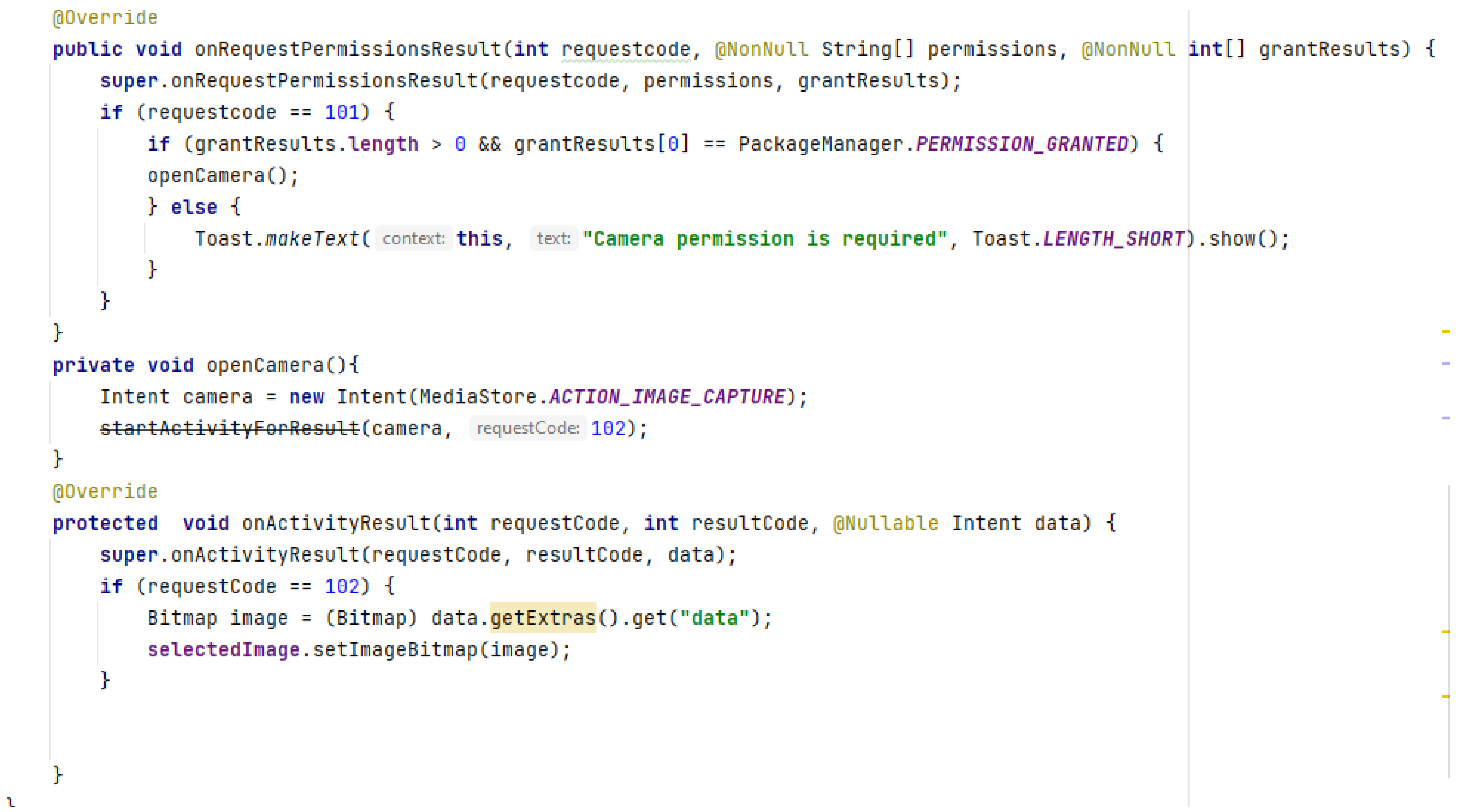
```
<?xml version = "1.0" encoding = "utf-8"?>
<manifest xmlns:android = "http://schemas.android.com/apk/res/android"
   package = "com.example.andy.myapplication">
   <application
      android:allowBackup = "true"
      android:icon = "@mipmap/ic_launcher"
      android:label = "@string/app_name"
      android:roundIcon = "@mipmap/ic_launcher_round"
      android:supportsRtl = "true"
      android:theme = "@style/AppTheme">
      <activity android:name = ".MainActivity">
         <intent-filter>
            <action android:name = "android.intent.action.MAIN" />
            <category android:name = "android.intent.category.LAUNCHER" />
         </intent-filter>
      </activity>
      <activity android:name = ".SecondActivity"></activity>
   </application>
</manifest>
```


10:40
My Application3
CAMERA



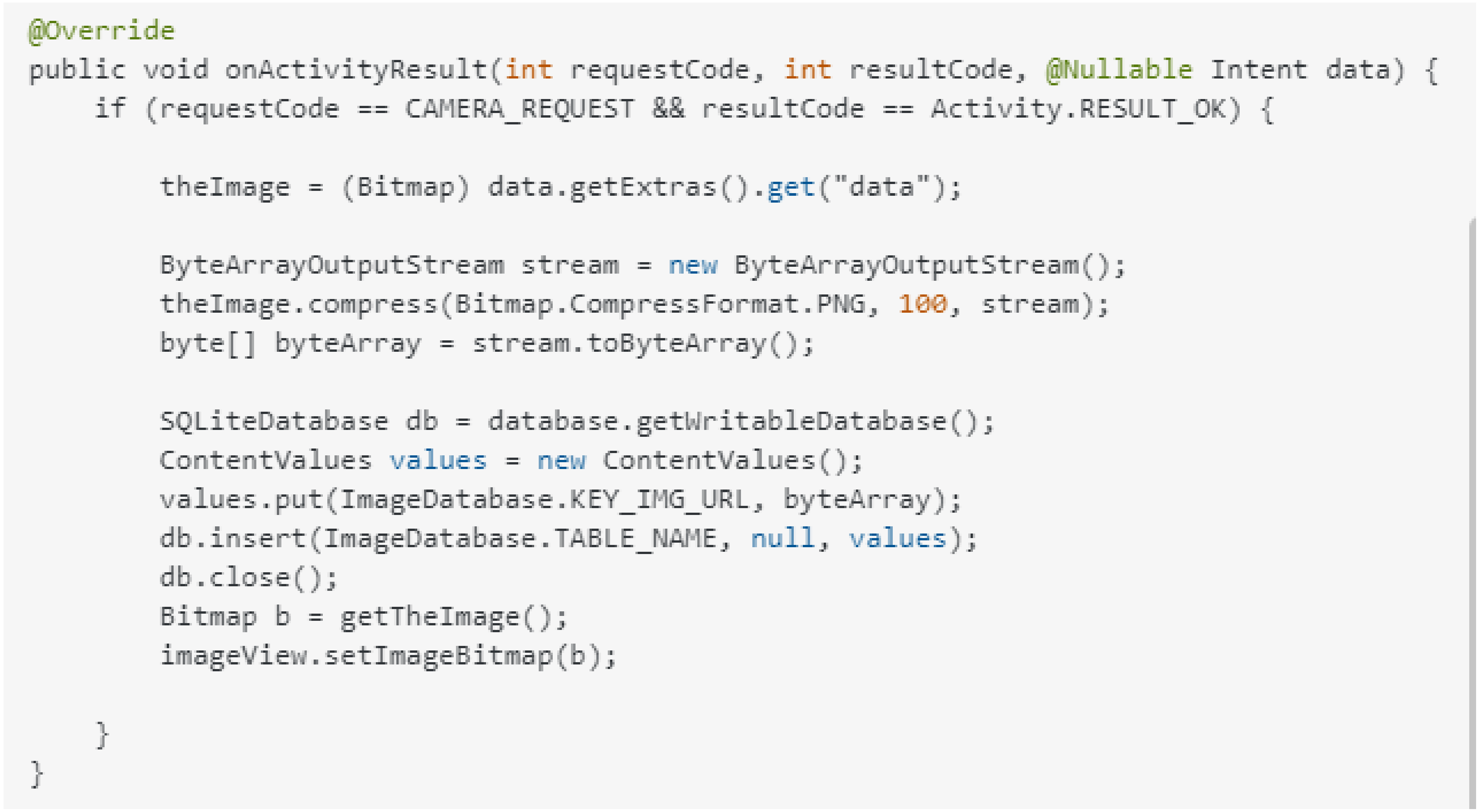

10:47
My Application3
CAMERA


```
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    ImageView selectedImage;
    Button Camerabutton;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        selectedImage = findViewById(R.id.imageView);
        Camerabutton = findViewById(R.id.button);
        Camerabutton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
            askCameraPermissions();
            }
        });
    }

    private void askCameraPermissions() {
        if (ContextCompat.checkSelfPermission( context: this, Manifest.permission.CAMERA) != PackageManager.PERMISSION_GRANTED
            ActivityCompat.requestPermissions( activity: this, new String[]{Manifest.permission.CAMERA},  requestCode: 101);
        } else {
            openCamera();
        }
    }
```



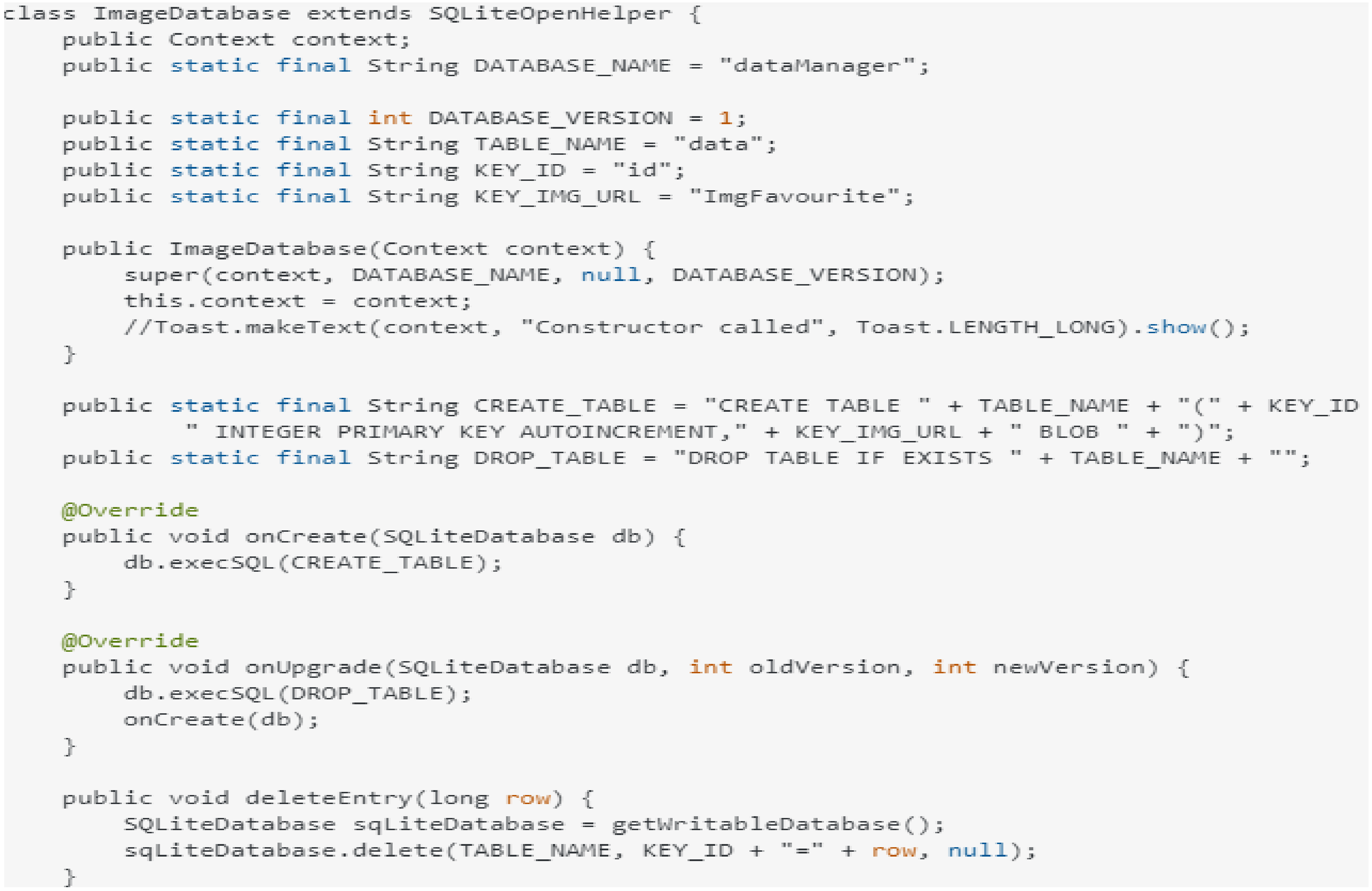
```
@Override
public void onRequestPermissionsResult(int requestcode, @NonNull String[] permissions, @NonNull int[] grantResults) {
    super.onRequestPermissionsResult(requestcode, permissions, grantResults);
    if (requestcode == 101) {
        if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
        openCamera();
        } else {
            Toast.makeText( context: this,  text: "Camera permission is required", Toast.LENGTH_SHORT).show();
        }
    }
}
private void openCamera(){
    Intent camera = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
    ~~startActivityForResult~~(camera,  requestCode: 102);
}
@Override
protected  void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == 102) {
        Bitmap image = (Bitmap) data.getExtras().get("data");
        selectedImage.setImageBitmap(image);
    }


}
}
```

database = new ImageDatabase(this);

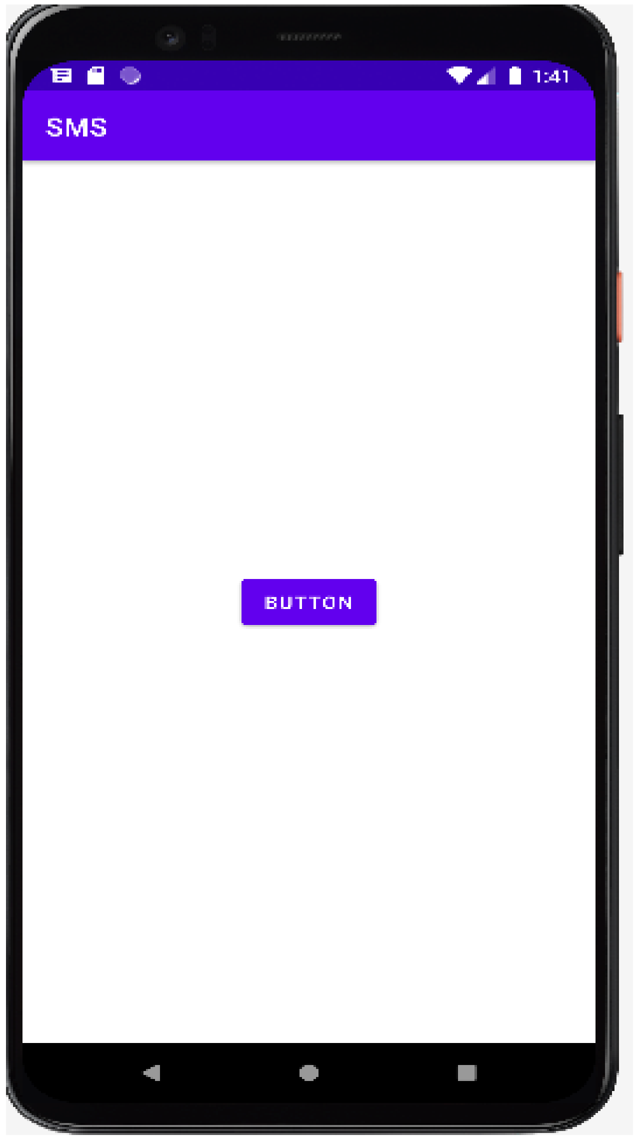


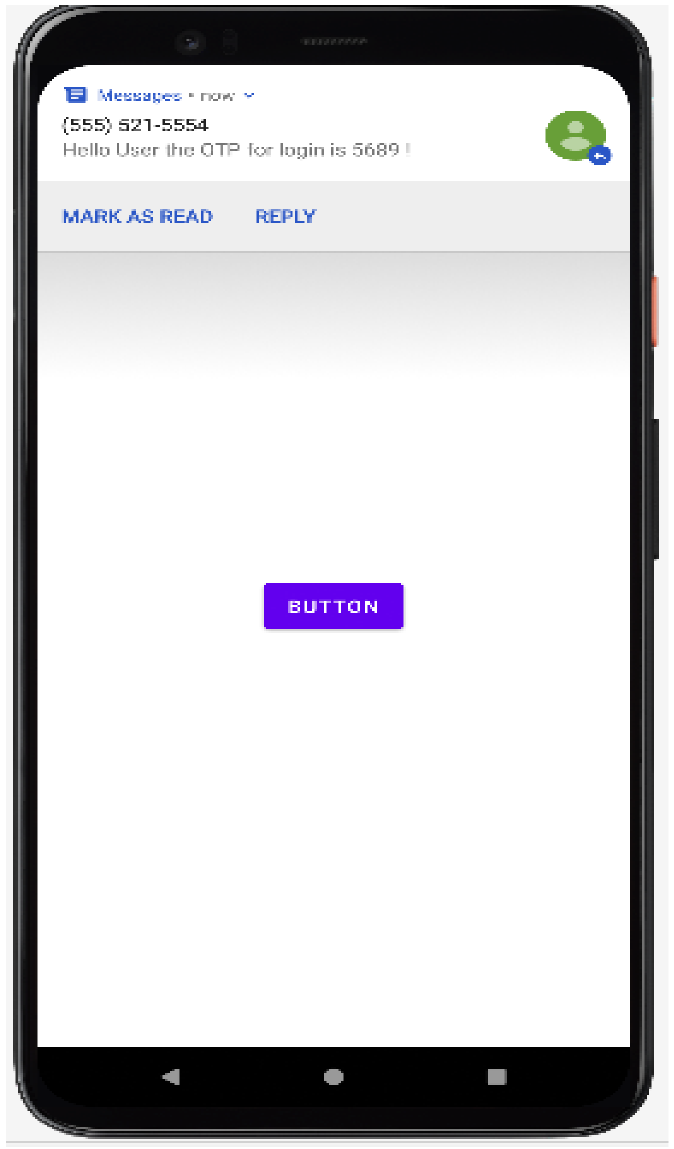
```
@Override
public void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
    if (requestCode == CAMERA_REQUEST && resultCode == Activity.RESULT_OK) {

        theImage = (Bitmap) data.getExtras().get("data");

        ByteArrayOutputStream stream = new ByteArrayOutputStream();
        theImage.compress(Bitmap.CompressFormat.PNG, 100, stream);
        byte[] byteArray = stream.toByteArray();

        SQLiteDatabase db = database.getWritableDatabase();
        ContentValues values = new ContentValues();
        values.put(ImageDatabase.KEY_IMG_URL, byteArray);
        db.insert(ImageDatabase.TABLE_NAME, null, values);
        db.close();
        Bitmap b = getTheImage();
        imageView.setImageBitmap(b);

    }
}
```

DATABASE CLASS



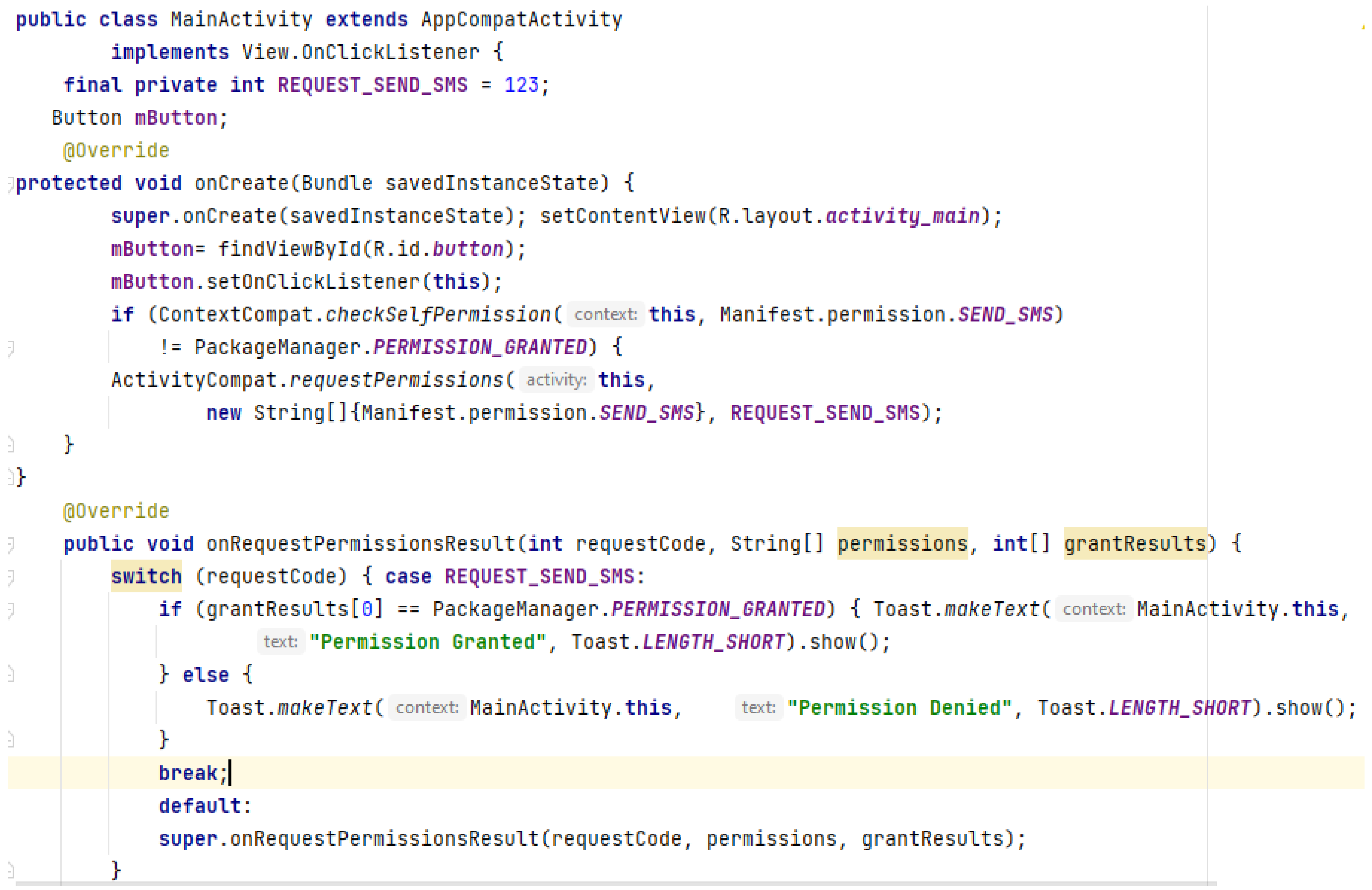
```java
class ImageDatabase extends SQLiteOpenHelper {
    public Context context;
    public static final String DATABASE_NAME = "dataManager";

    public static final int DATABASE_VERSION = 1;
    public static final String TABLE_NAME = "data";
    public static final String KEY_ID = "id";
    public static final String KEY_IMG_URL = "ImgFavourite";

    public ImageDatabase(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
        this.context = context;
        //Toast.makeText(context, "Constructor called", Toast.LENGTH_LONG).show();
    }

    public static final String CREATE_TABLE = "CREATE TABLE " + TABLE_NAME + "(" + KEY_ID
            " INTEGER PRIMARY KEY AUTOINCREMENT," + KEY_IMG_URL + " BLOB " + ")";
    public static final String DROP_TABLE = "DROP TABLE IF EXISTS " + TABLE_NAME + "";

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL(CREATE_TABLE);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL(DROP_TABLE);
        onCreate(db);
    }

    public void deleteEntry(long row) {
        SQLiteDatabase sqLiteDatabase = getWritableDatabase();
        sqLiteDatabase.delete(TABLE_NAME, KEY_ID + "=" + row, null);
    }
```



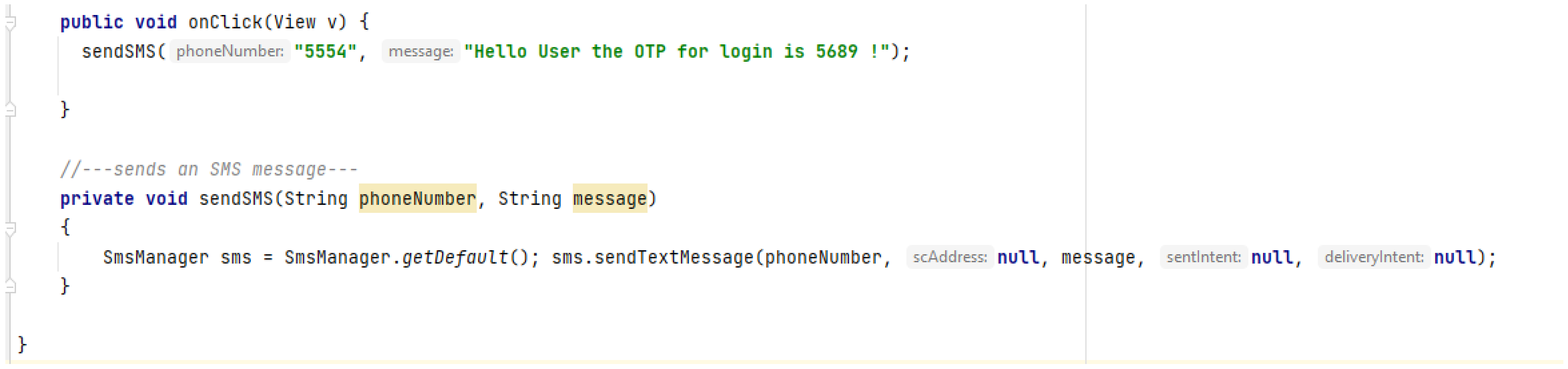

Messages • now

(555) 521-5554

Hello User the OTP for login is 5689 !



```
public class MainActivity extends AppCompatActivity
        implements View.OnClickListener {
    final private int REQUEST_SEND_SMS = 123;
   Button mButton;
    @Override
protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState); setContentView(R.layout.activity_main);
        mButton= findViewById(R.id.button);
        mButton.setOnClickListener(this);
        if (ContextCompat.checkSelfPermission( context: this, Manifest.permission.SEND_SMS)
            != PackageManager.PERMISSION_GRANTED) {
        ActivityCompat.requestPermissions( activity: this,
                new String[]{Manifest.permission.SEND_SMS}, REQUEST_SEND_SMS);
    }
}
    @Override
    public void onRequestPermissionsResult(int requestCode, String[] permissions, int[] grantResults) {
        switch (requestCode) { case REQUEST_SEND_SMS:
            if (grantResults[0] == PackageManager.PERMISSION_GRANTED) { Toast.makeText( context: MainActivity.this,
                    text: "Permission Granted", Toast.LENGTH_SHORT).show();
            } else {
                Toast.makeText( context: MainActivity.this,     text: "Permission Denied", Toast.LENGTH_SHORT).show();
            }
            break;
            default:
            super.onRequestPermissionsResult(requestCode, permissions, grantResults);
        }
```



```
    public void onClick(View v) {
      sendSMS( phoneNumber: "5554",  message: "Hello User the OTP for login is 5689 !");

    }

    //---sends an SMS message---
    private void sendSMS(String phoneNumber, String message)
    {
        SmsManager sms = SmsManager.getDefault(); sms.sendTextMessage(phoneNumber,  scAddress: null, message,  sentIntent: null,  deliveryIntent: null);
    }

}
```

References

- https://developer.android.com/
- https://www.geeksforgeeks.org/android-tutorial/
- https://www.tutorialspoint.com/android/index.htm