

# Android Development

Sanjit Singh Chouhan  
Sai Bhavani Kumar  
Paritosh Srivastav

# Android App Fundamentals

1. Activity
2. Fragments
3. Intents
4. Services
5. Android Manifest
6. Layouts
7. Components

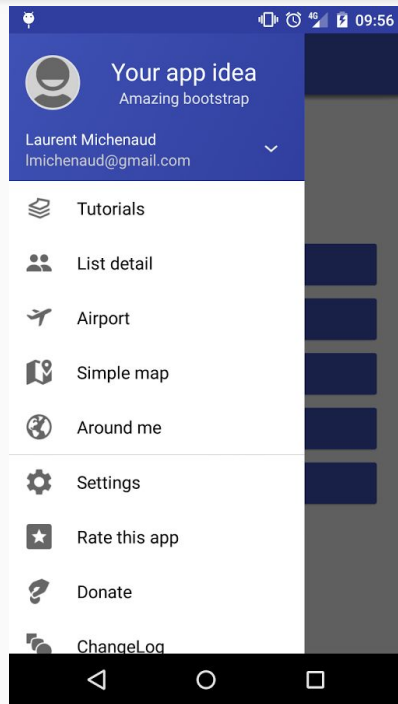
# Activity

- A window/interface which a user can interact with
- Every app has at least one activity



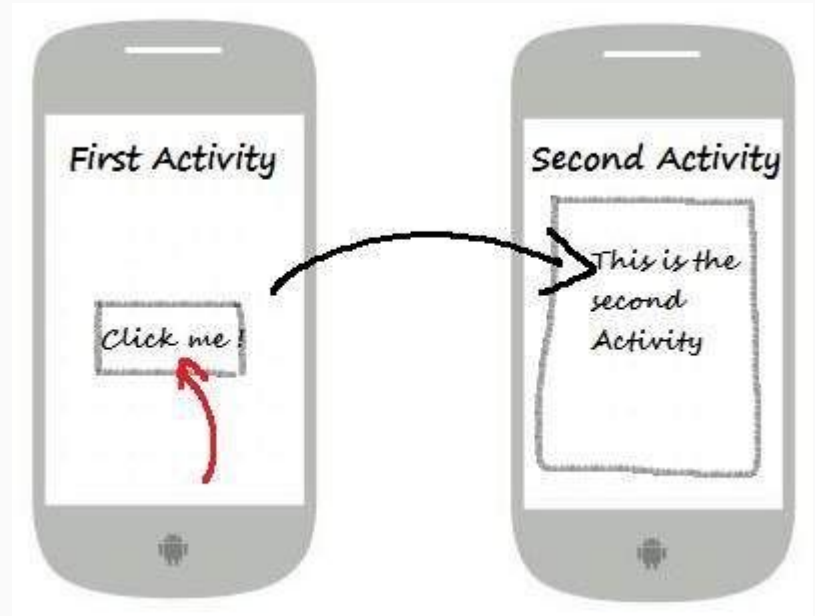
# Fragments

A piece of User Interface  
that is meant to be reused



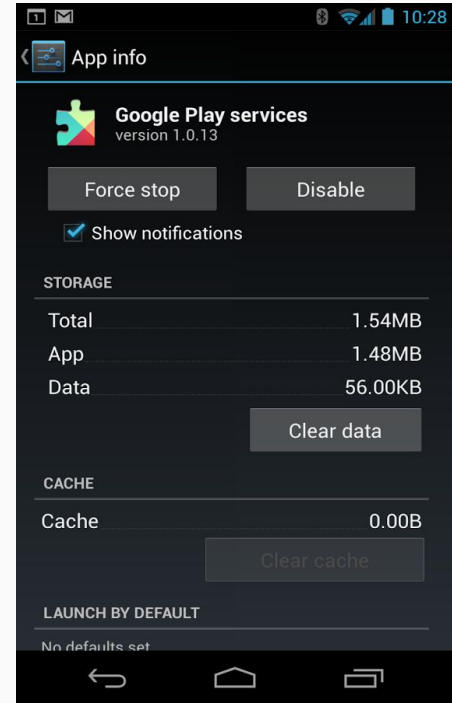
# Intents

Abstract Description of operation to  
be performed

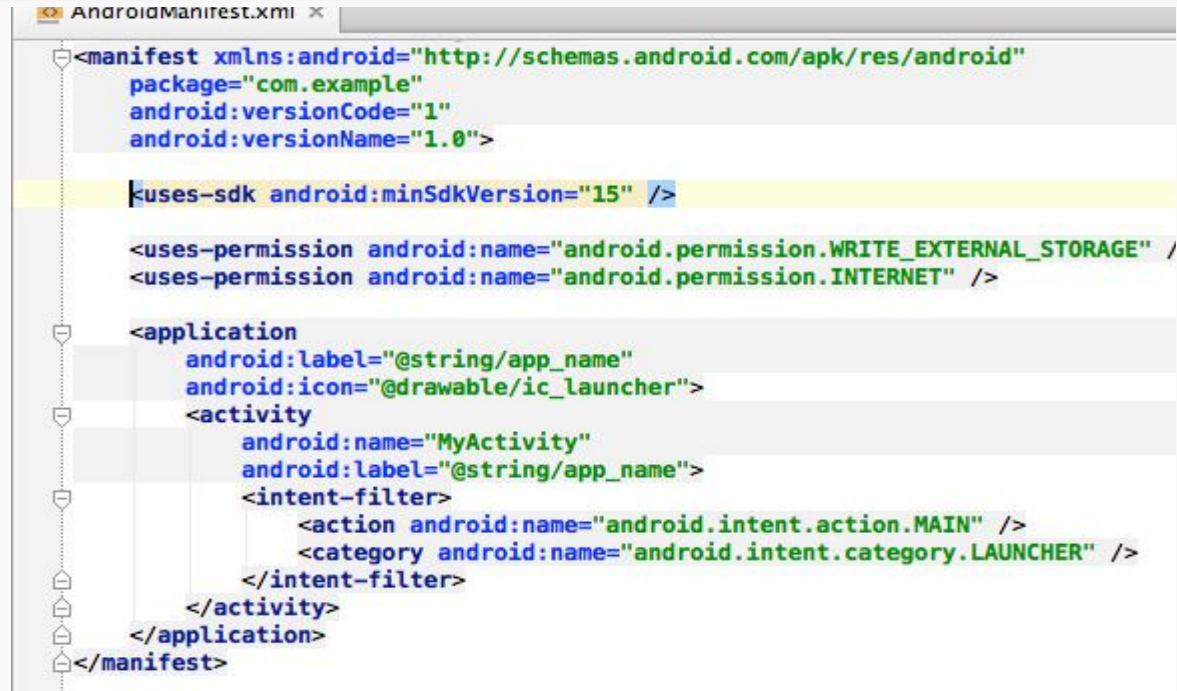


# Services

Services are long running operations  
in background



# Android Manifest

A screenshot of an IDE window showing the content of an AndroidManifest.xml file. The window title is 'AndroidManifest.xml'. The XML code is displayed with syntax highlighting. A tree view on the left shows the hierarchical structure of the manifest, with expandable icons for the root, application, activity, and intent-filter elements. The code defines the package, version, permissions, and a single activity with a specific intent filter.

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example"
    android:versionCode="1"
    android:versionName="1.0">

    <uses-sdk android:minSdkVersion="15" />

    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.INTERNET" />

    <application
        android:label="@string/app_name"
        android:icon="@drawable/ic_launcher">
        <activity
            android:name="MyActivity"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

# Layouts

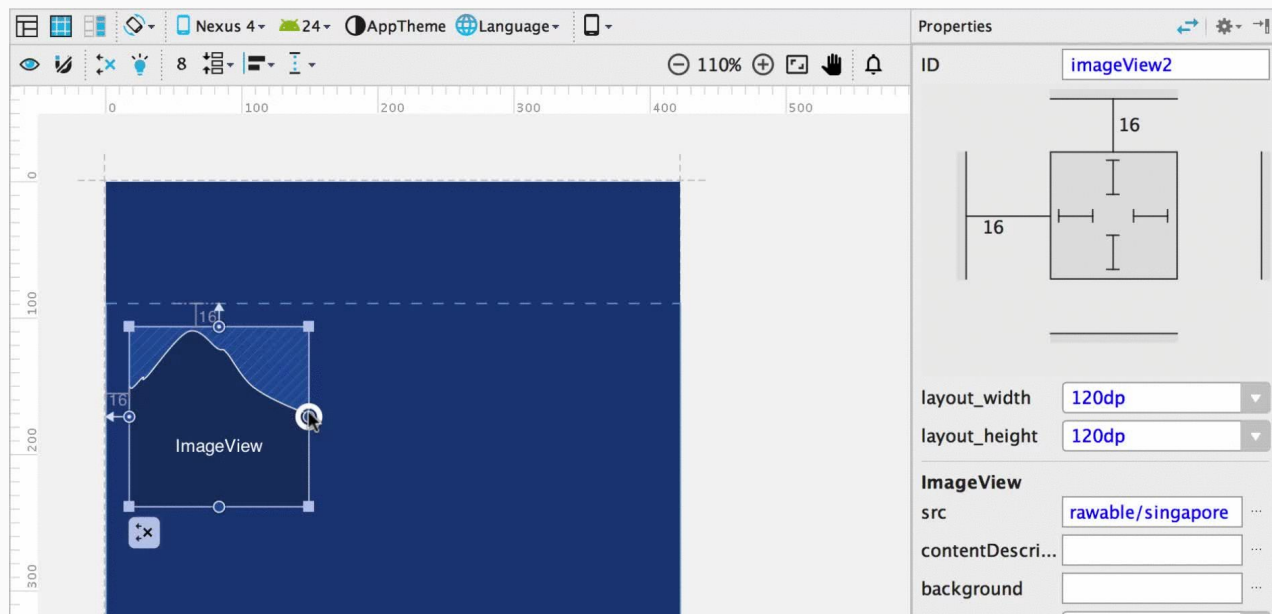
Layouts define visual structure for a user interface(activity)

1. Constraint Layout
2. RelativeLayout
3. LinearLayout
4. GridLayout



# Constraint Layout

Constraint Layout  
allows to create  
large and complex  
layouts with a flat  
view hierarchy



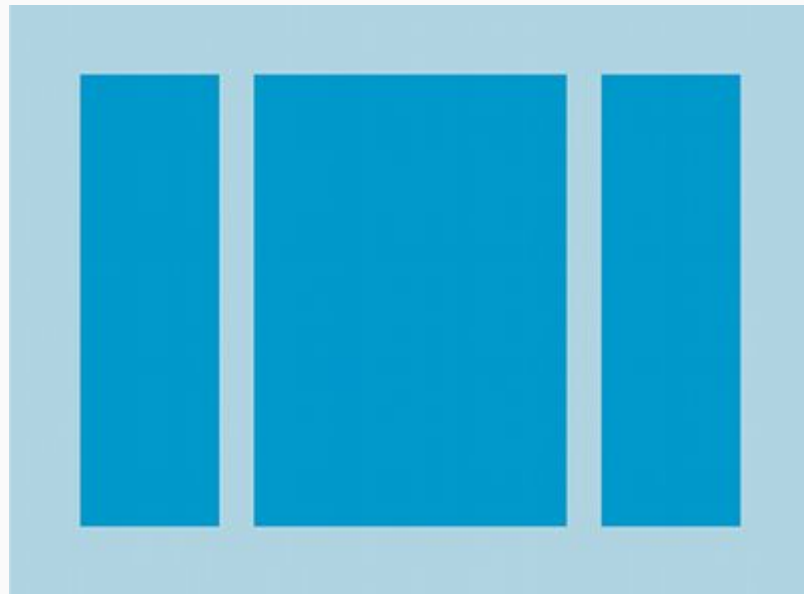
# RelativeLayout

RelativeLayout is a viewgroup that displays child views in relative positions



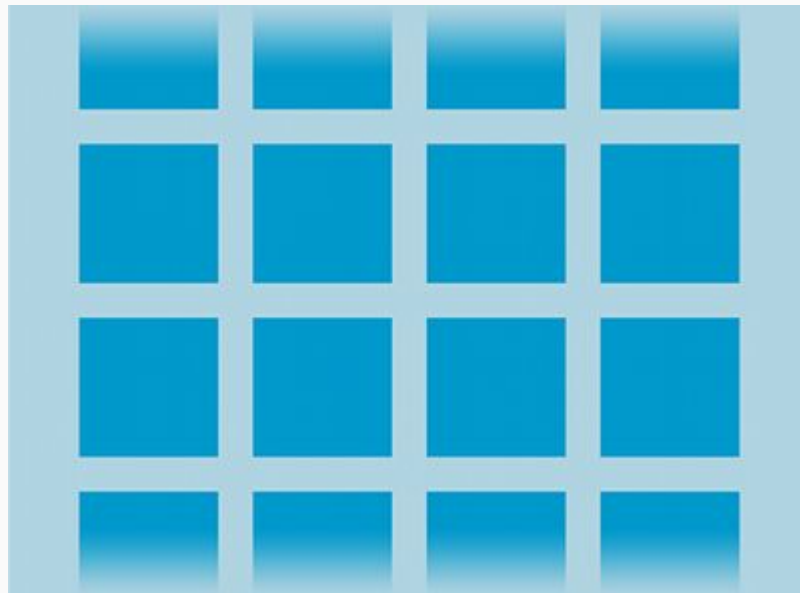
# LinearLayout

LinearLayout is a viewgroup that aligns all children in a single direction, vertically or horizontally



# GridLayout

GridLayout is a viewgroup that aligns all children in a grid



# Components(Views)

1. Buttons
2. EditText
3. TextView
4. CheckBox
5. ToggleButton
6. SeekBar
7. Spinner (DropDown)
8. ProgressBar

