Mobile Software Design and Development

2021/2022

T.A. : Morten Asmus Breum Nørgård

monoe18@student.sdu.dk

Frederik Verner Helth

Teacher: Mahyar T. Moghaddam

Android - Part 1

Experience

- Java
- Kotlin
- Android

Outline

- Materials
- Android
- Android Studio
- Kotlin
- Application structure
- Android Components
- Intent
- Views
- XML Layout files
- *Services
- *Content providers
- Exercise
- Assignment

Materials

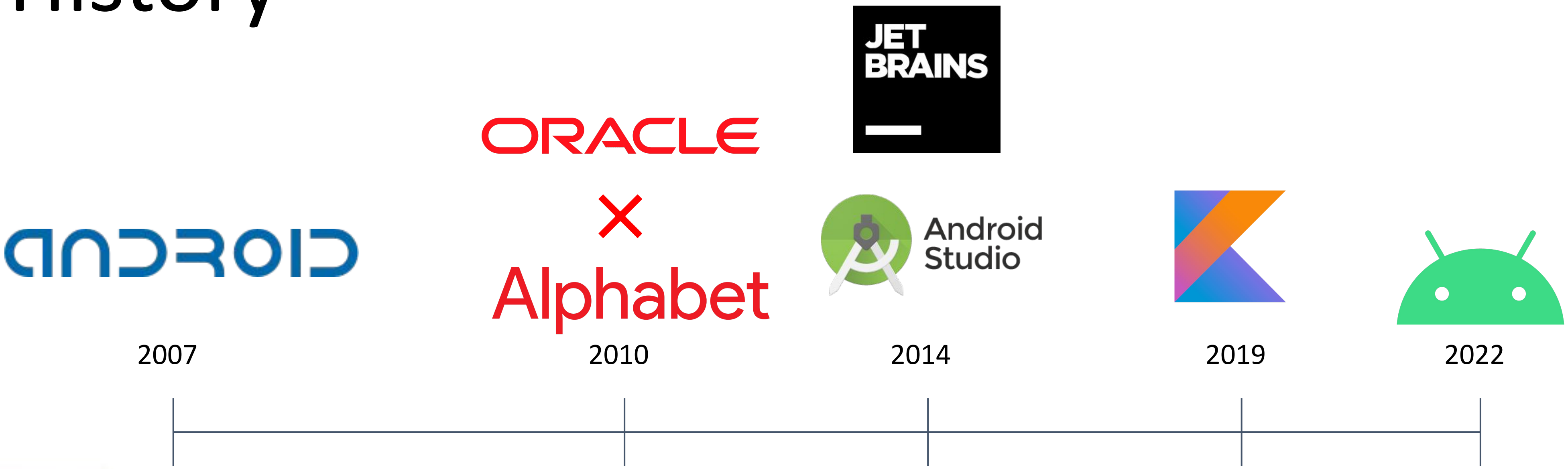
- https://developer.android.com/

- Youtube
- Udemy
- Udacity

Learn to:

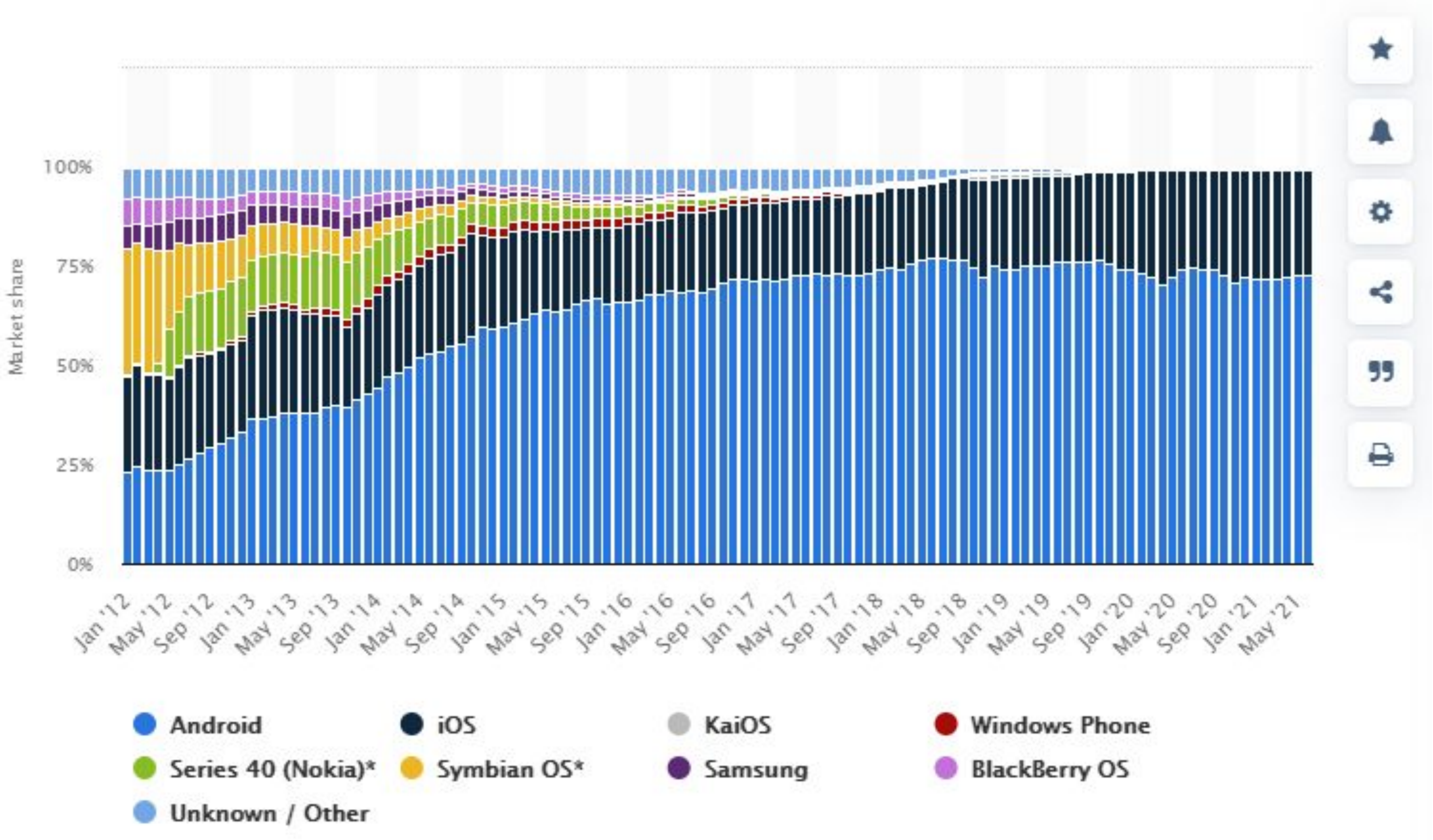
- Create amazing apps for the latest Android smartphones and tablets
- Download and install the SDK and start working with the Android Studio tools
- Adapt your apps for use on Android watches and TVs
- Publish your apps to the Google lay™ Store

History

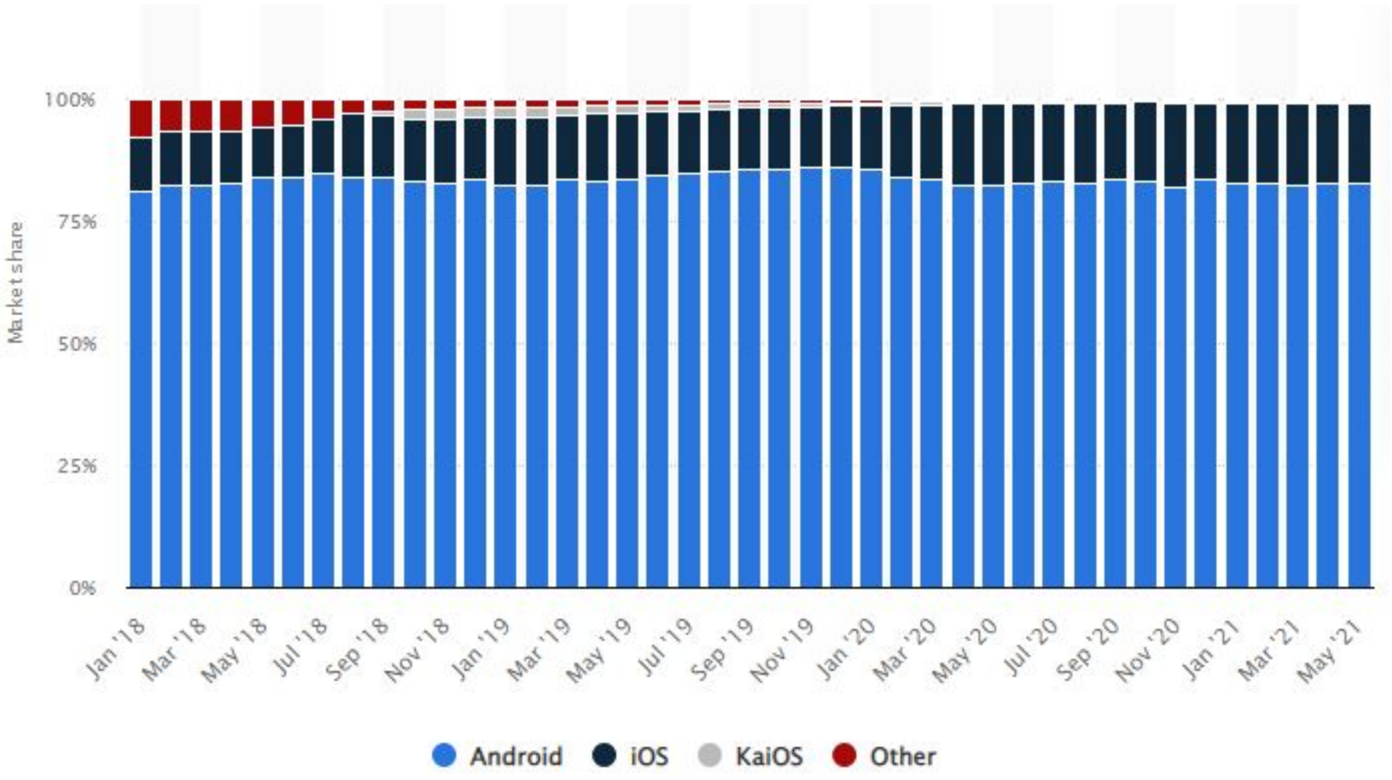

ANDROID
2007
ORACLE
×
Alphabet
2010
JET BRAINS
Android Studio
2014
2019
2022

T-Mobile
MENU

History


100%
75%
50%
25%
0%
Market share
Jan '12
May '12
Sep '12
Jan '13
May '13
Sep '13
Jan '14
May '14
Sep '14
Jan '15
May '15
Sep '15
Jan '16
May '16
Sep '16
Jan '17
May '17
Sep '17
Jan '18
May '18
Sep '18
Jan '19
May '19
Sep '19
Jan '20
May '20
Sep '20
Jan '21
May '21
Android
iOS
KaiOS
Windows Phone
Series 40 (Nokia)*
Symbian OS*
Samsung
BlackBerry OS
Unknown / Other

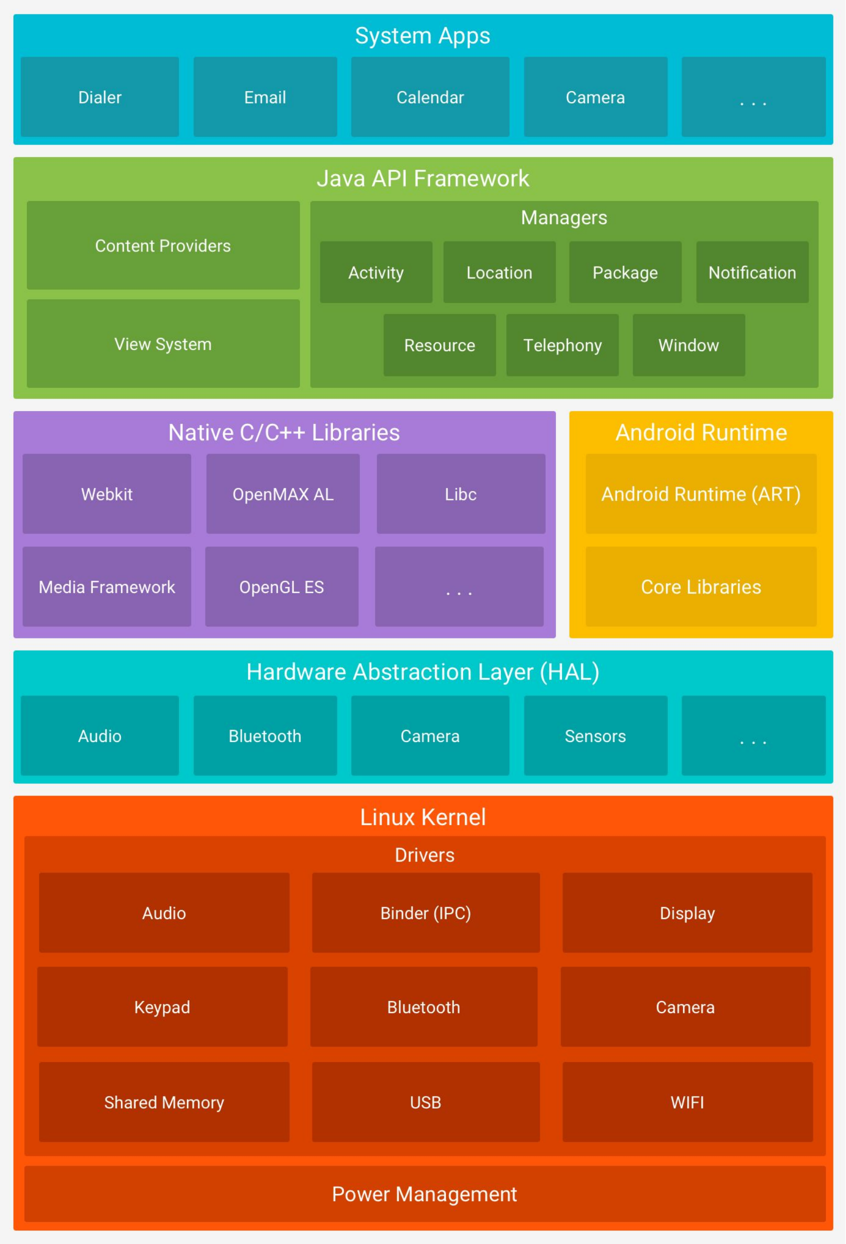
History


100%
75%
50%
25%
0%
Market share
Jan '18
Mar '18
May '18
Jul '18
Sep '18
Nov '18
Jan '19
Mar '19
May '19
Jul '19
Sep '19
Nov '19
Jan '20
Mar '20
May '20
Jul '20
Sep '20
Nov '20
Jan '21
Mar '21
May '21
● Android
● iOS
● KaiOS
● Other

Android System

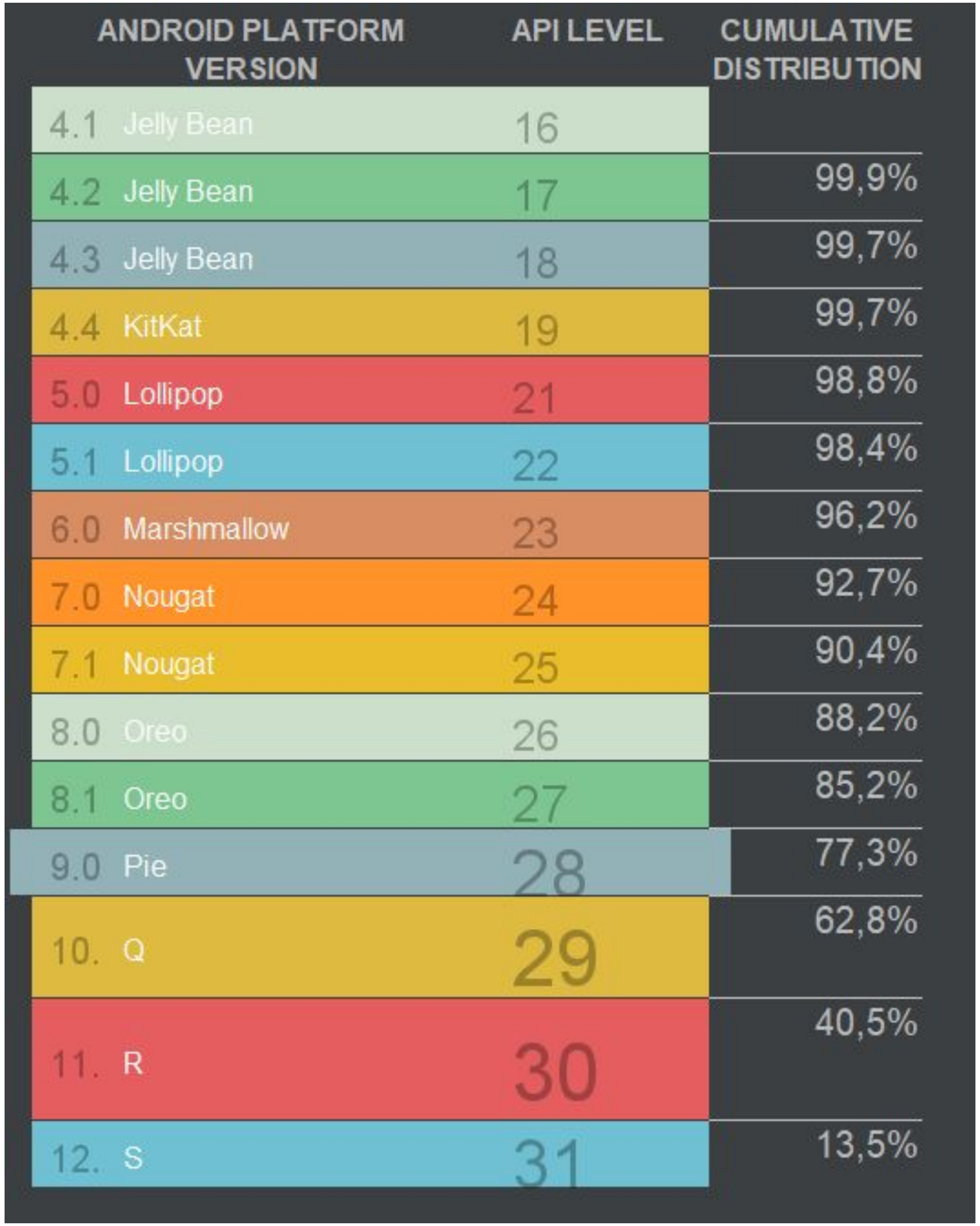
- Linux
- C/C++
- Java

- Process separation
  - Multi user linux environment
  - Unique VM
- Principle of least privilege



Version Distribution

- Forward compatibility



ANDROID PLATFORM VERSION		API LEVEL	CUMULATIVE DISTRIBUTION
4.1	Jelly Bean	16	
4.2	Jelly Bean	17	99,9%
4.3	Jelly Bean	18	99,7%
4.4	KitKat	19	99,7%
5.0	Lollipop	21	98,8%
5.1	Lollipop	22	98,4%
6.0	Marshmallow	23	96,2%
7.0	Nougat	24	92,7%
7.1	Nougat	25	90,4%
8.0	Oreo	26	88,2%
8.1	Oreo	27	85,2%
9.0	Pie	28	77,3%
10.	Q	29	62,8%
11.	R	30	40,5%
12.	S	31	13,5%

Android Studio

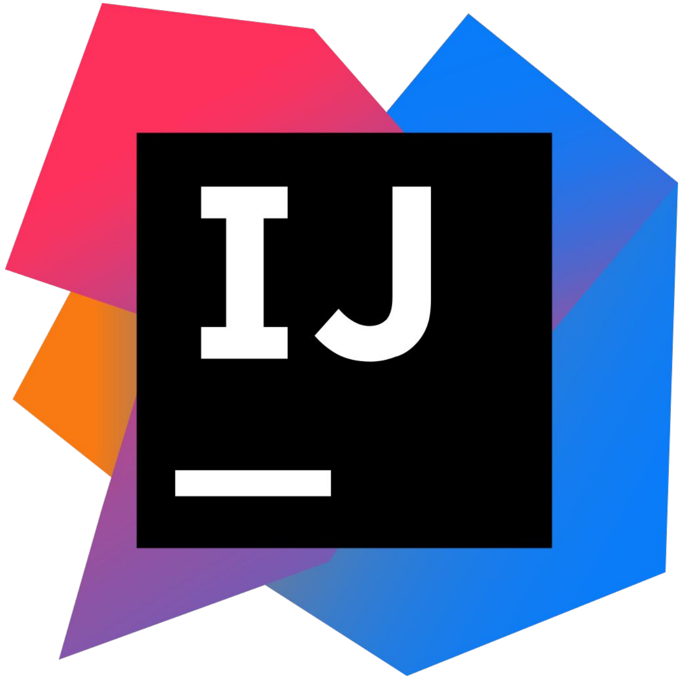
Dedicated Android IDE

AVD Manager

Layout editor

SDK Manager

Etc.



Kotlin

- Java 2.0
- Compatibility

- Null safe
- Type inference

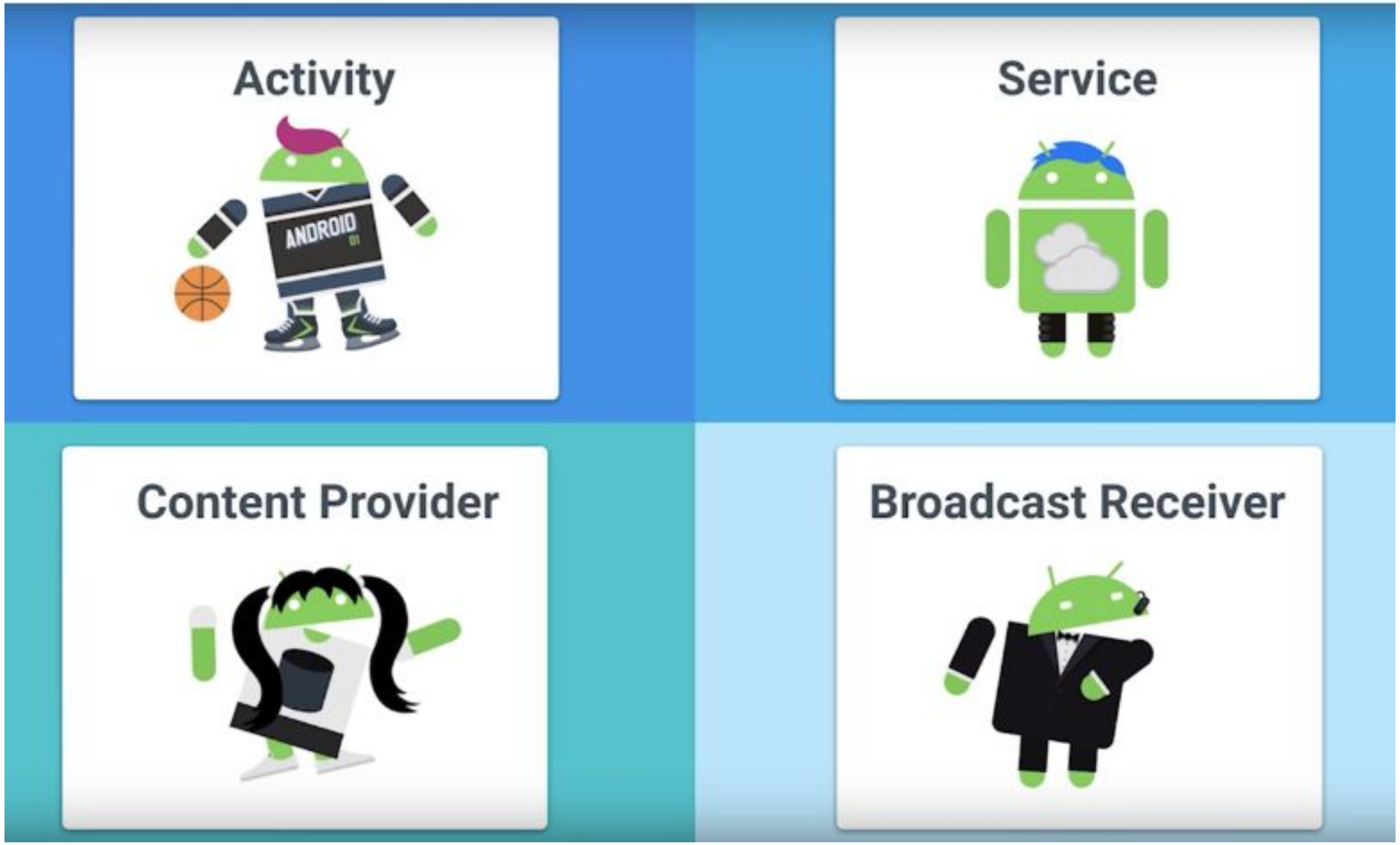
Syntax example

Getting Started

Follow this guide:

https://developer.android.com/training/basics/firstapp

Components

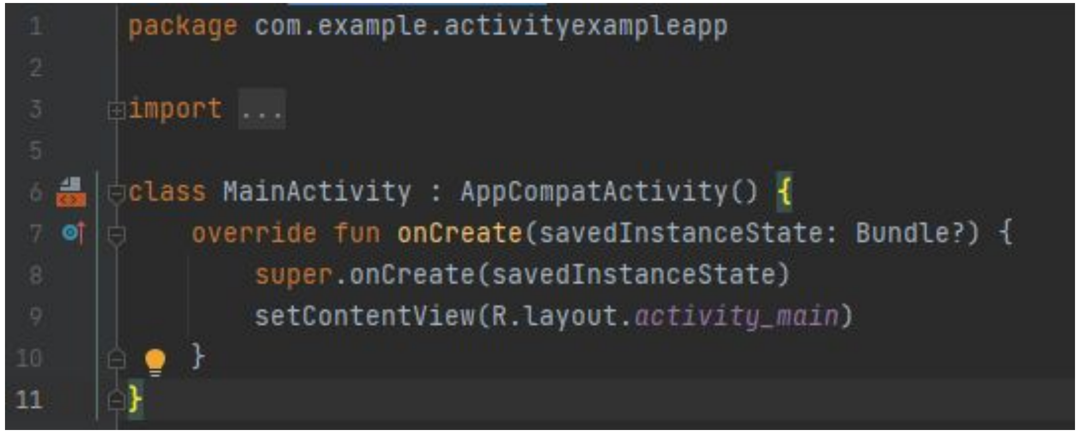

Activity
ANDROID
01
Service
Content Provider
Broadcast Receiver

Activity

A single frame

Inflates UI

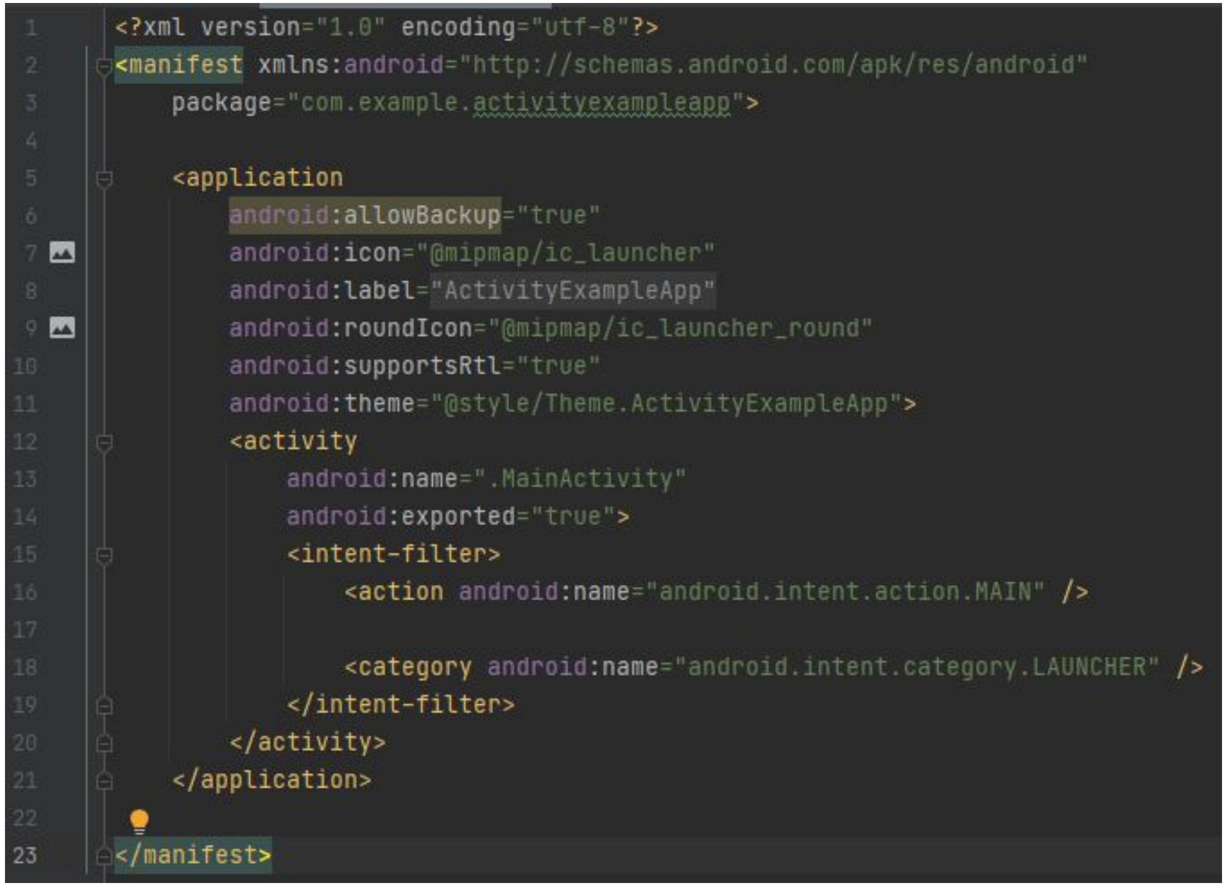
One Main activity



```kotlin
package com.example.activityexampleapp

import ...

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }
}
```



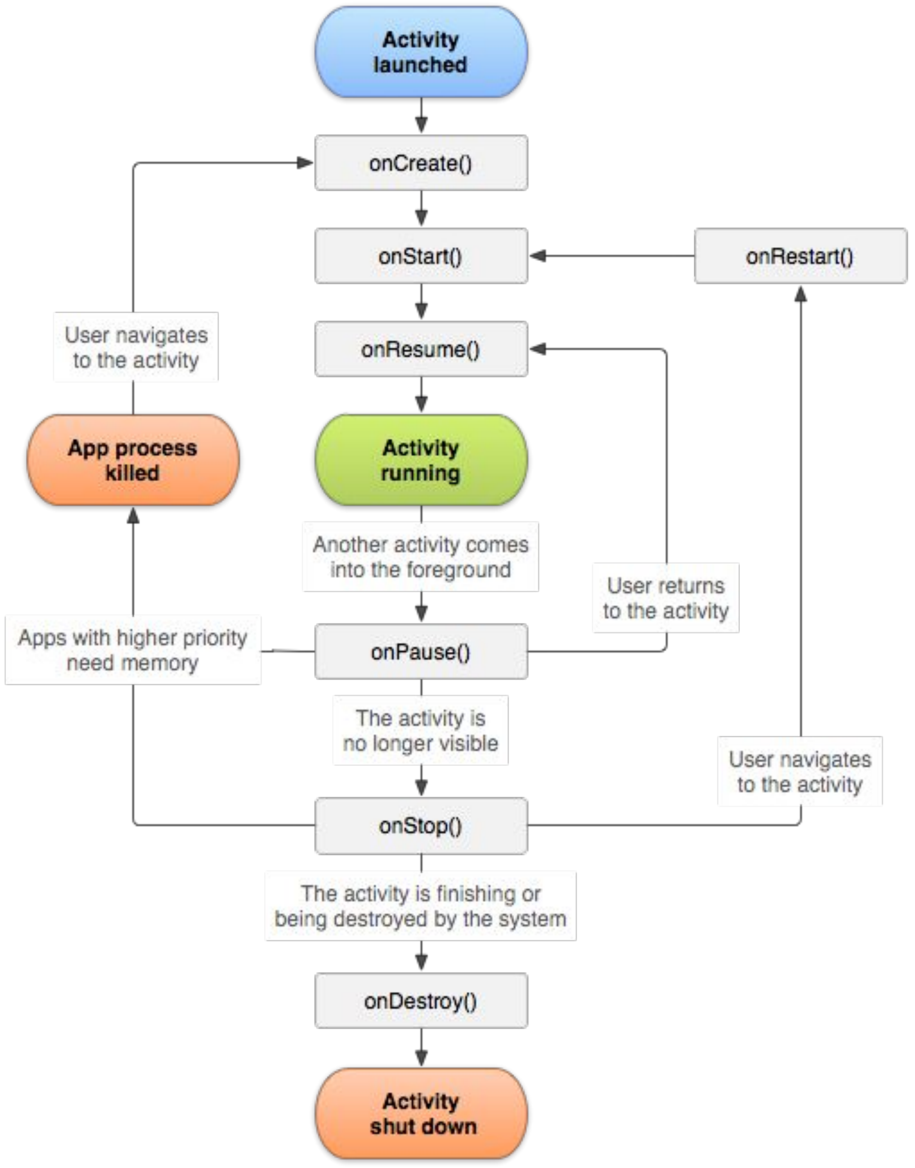
```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.activityexampleapp">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="ActivityExampleApp"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.ActivityExampleApp">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

Activity Lifecycle

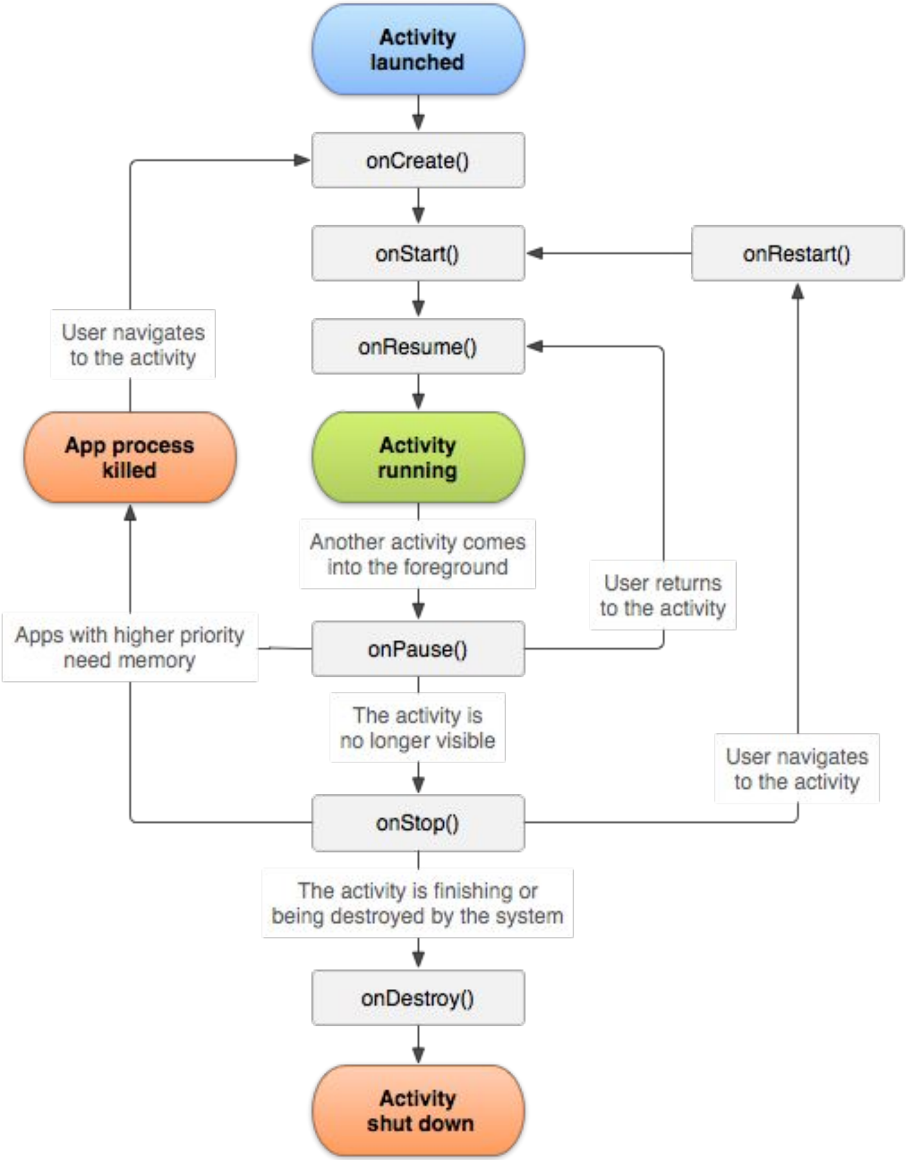

Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to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

Activity Lifecycle

OnCreate()

- Initialization

OnStart()


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

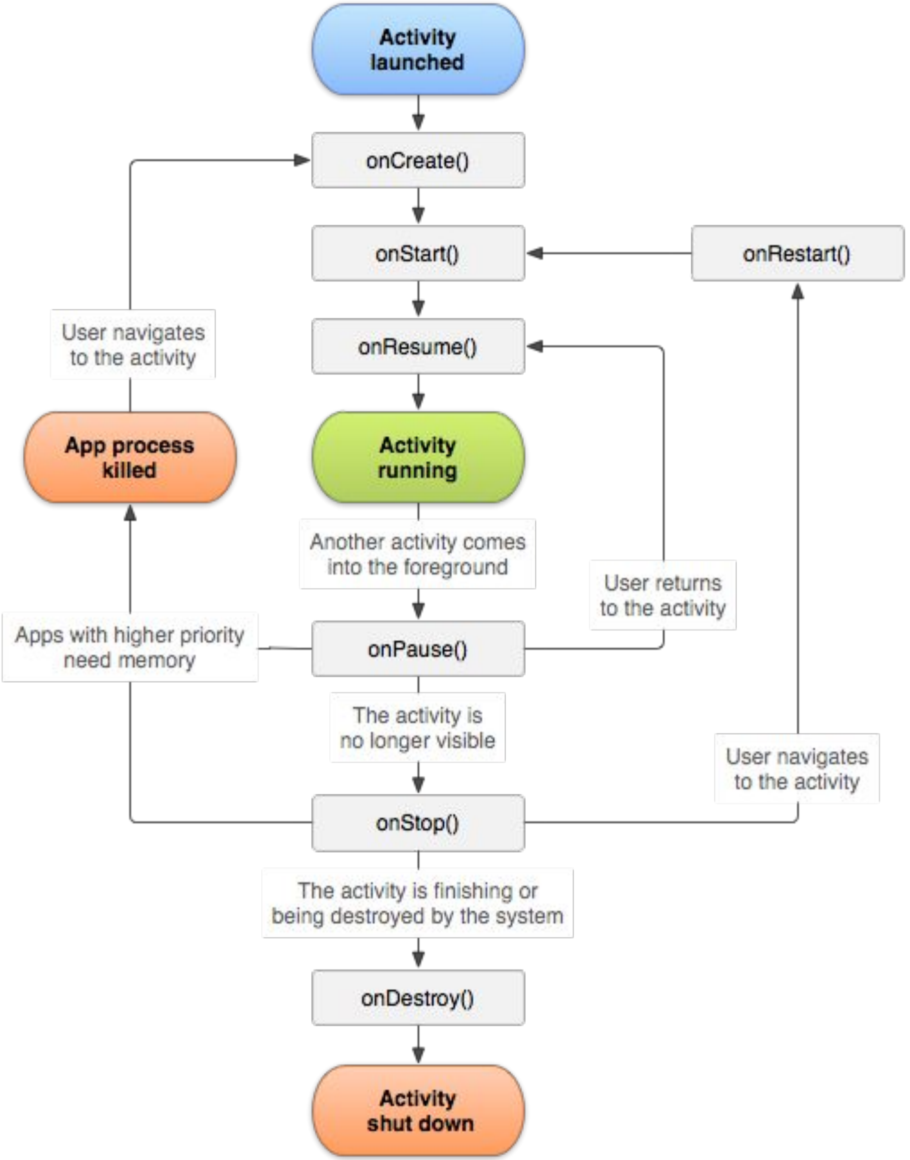
Activity Lifecycle

OnResume()

- Interaction available

OnPause()

- Interaction not available


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

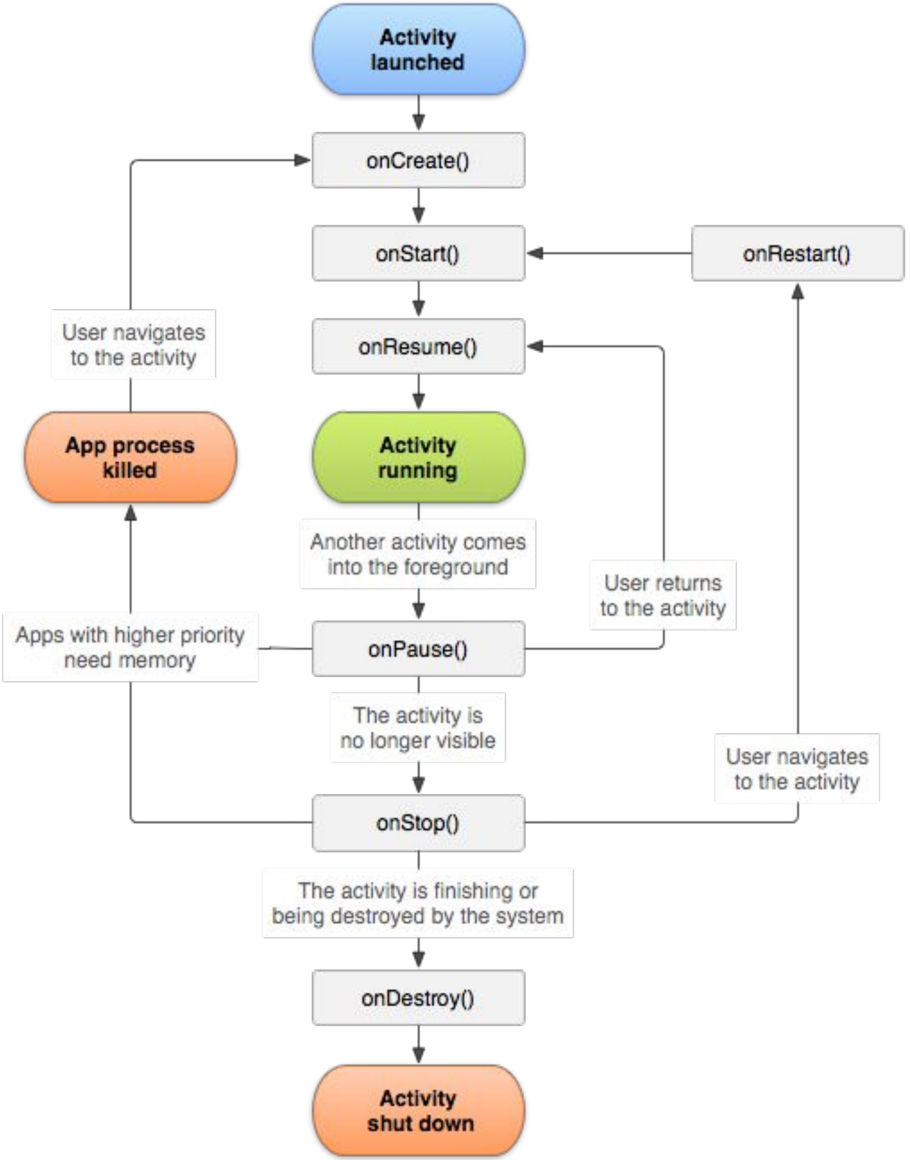
Activity Lifecycle

OnStop()

- Activity no longer visible

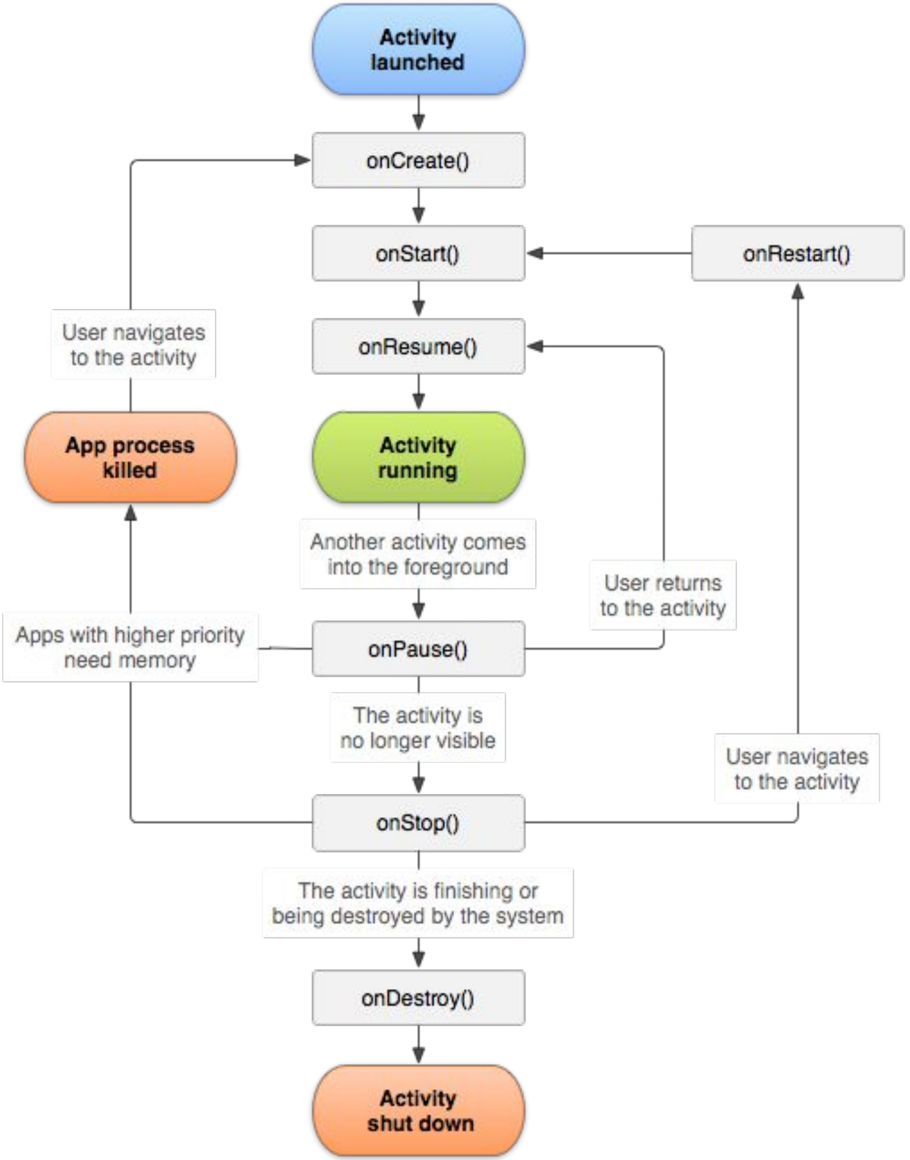
OnDestroy

- Activity has been finished or reconfigured.
  - Screen rotation
  - Language change


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

Activity Lifecycle

Example


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

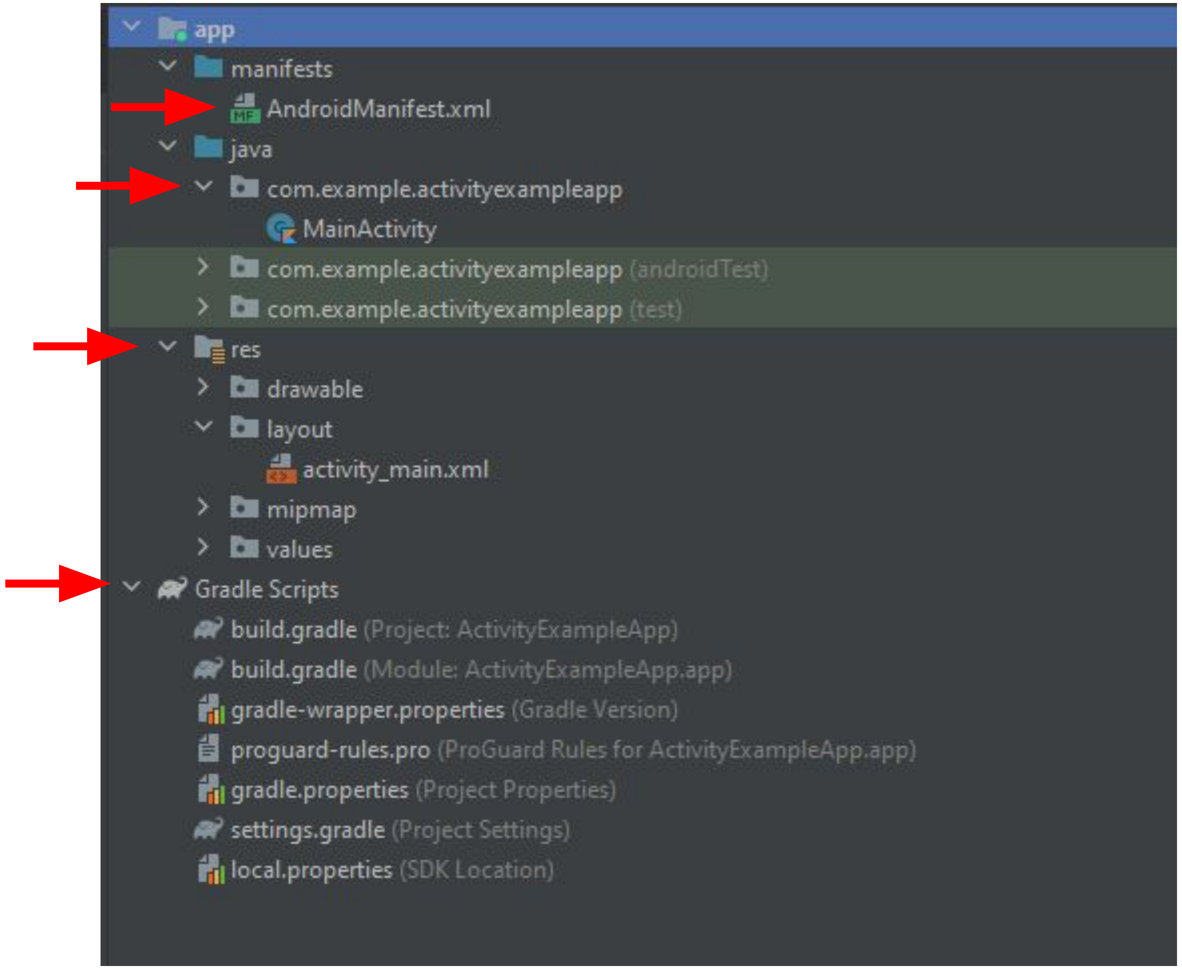
Application Structure

Code resources

Non-code resources

Gradle

Manifest



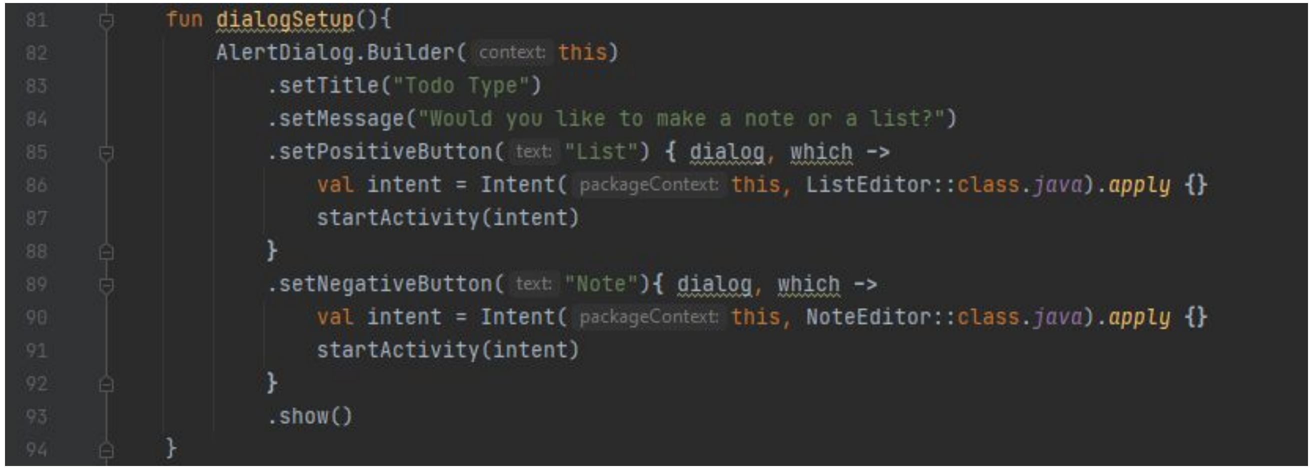
```
app
    manifests
        AndroidManifest.xml
    java
        com.example.activityexampleapp
            MainActivity
        com.example.activityexampleapp (androidTest)
        com.example.activityexampleapp (test)
    res
        drawable
        layout
            activity_main.xml
        mipmap
        values
Gradle Scripts
    build.gradle (Project: ActivityExampleApp)
    build.gradle (Module: ActivityExampleApp.app)
    gradle-wrapper.properties (Gradle Version)
    proguard-rules.pro (ProGuard Rules for ActivityExampleApp.app)
    gradle.properties (Project Properties)
    settings.gradle (Project Settings)
    local.properties (SDK Location)
```

Intents

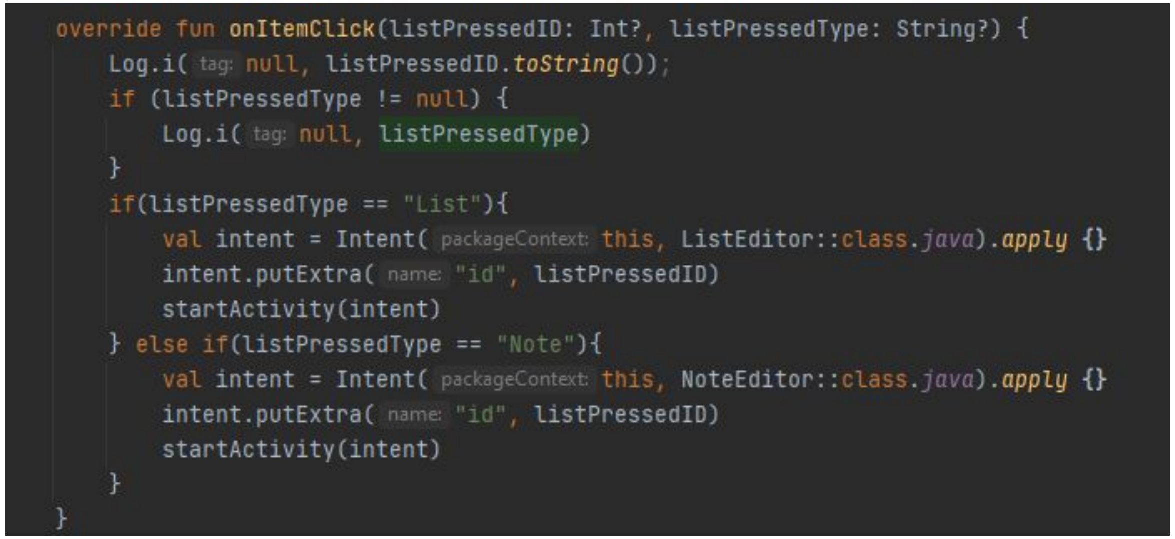
Messengers

Glue for the building blocks

Example



```
fun dialogSetup(){
    AlertDialog.Builder( context: this)
        .setTitle("Todo Type")
        .setMessage("Would you like to make a note or a list?")
        .setPositiveButton( text: "List") { dialog, which ->
            val intent = Intent( packageContext: this, ListEditor::class.java).apply {}
            startActivity(intent)
        }
        .setNegativeButton( text: "Note"){ dialog, which ->
            val intent = Intent( packageContext: this, NoteEditor::class.java).apply {}
            startActivity(intent)
        }
        .show()
}
```

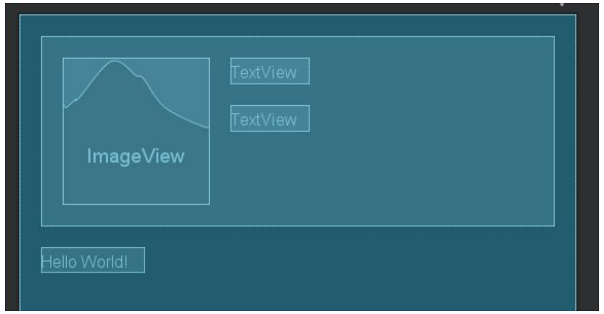


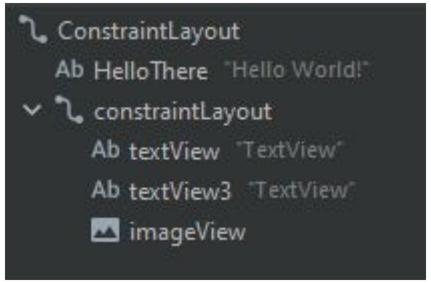
```
override fun onItemClick(listPressedID: Int?, listPressedType: String?) {
    Log.i( tag: null, listPressedID.toString());
    if (listPressedType != null) {
        Log.i( tag: null, listPressedType)
    }
    if(listPressedType == "List"){
        val intent = Intent( packageContext: this, ListEditor::class.java).apply {}
        intent.putExtra( name: "id", listPressedID)
        startActivity(intent)
    } else if(listPressedType == "Note"){
        val intent = Intent( packageContext: this, NoteEditor::class.java).apply {}
        intent.putExtra( name: "id", listPressedID)
        startActivity(intent)
    }
}
```

Views

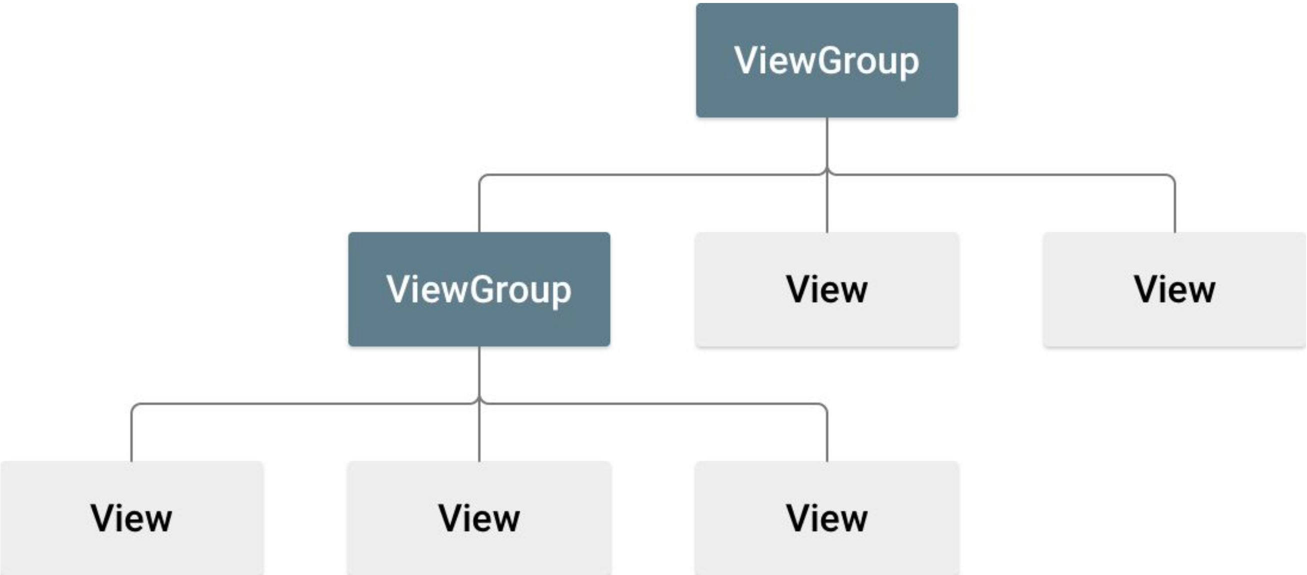
Trees

- ViewGroups/Layouts
- Views
- Custom Views





```
ConstraintLayout
  Ab HelloThere "Hello World!"
  ⌄ constraintLayout
      Ab textView "TextView"
      Ab textView3 "TextView"
      imageView
```



XML

Markup language

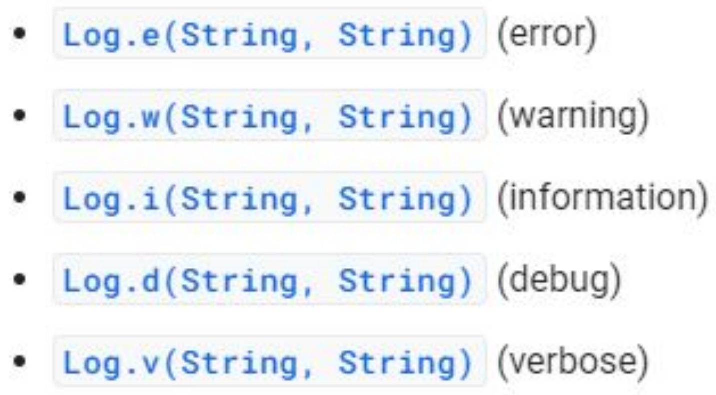
Android studio layout editor

Example

Debugging

Logcat

- Improved System.out.println()



- `Log.e(String, String)` (error)
- `Log.w(String, String)` (warning)
- `Log.i(String, String)` (information)
- `Log.d(String, String)` (debug)
- `Log.v(String, String)` (verbose)

Practice

- Finish the getting started guide

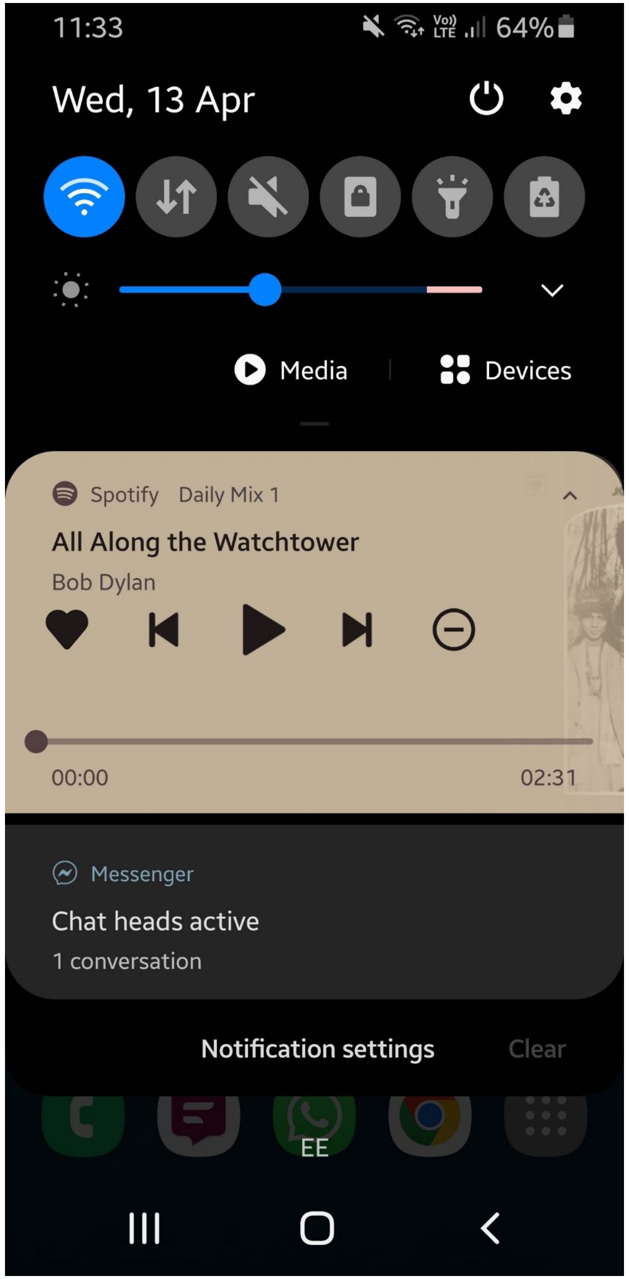
If you have time:

- Hook into the activity lifecycle methods and print the results through Logcat
- Try to recreate the “movie details” screen from your last assignment in the Layout Editor

Services

Foreground and background

Started and bound



Spotify Daily Mix 1

All Along the Watchtower

Bob Dylan

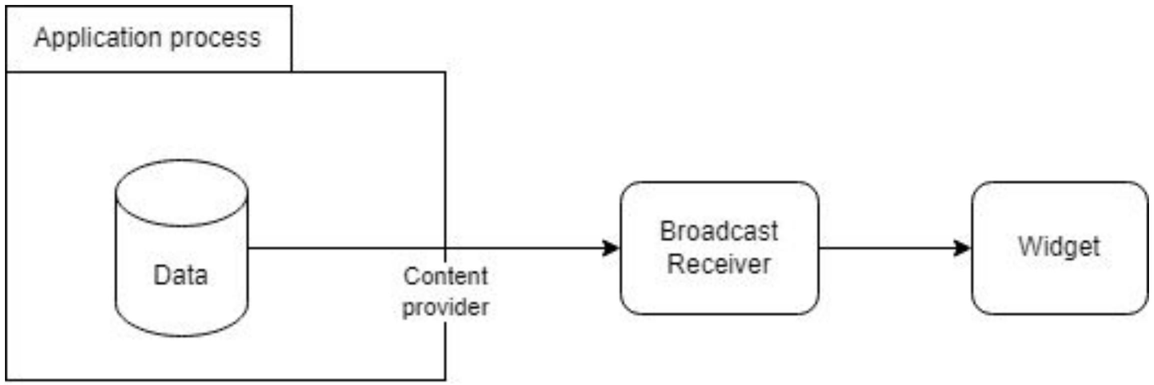
00:00 02:31

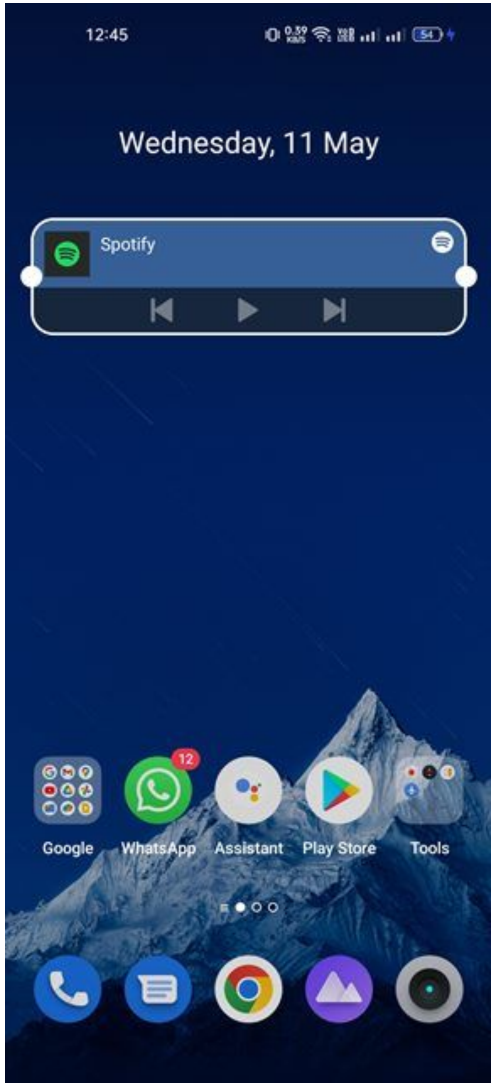
Messenger

Chat heads active

1 conversation

Content Providers and Broadcast Receiver



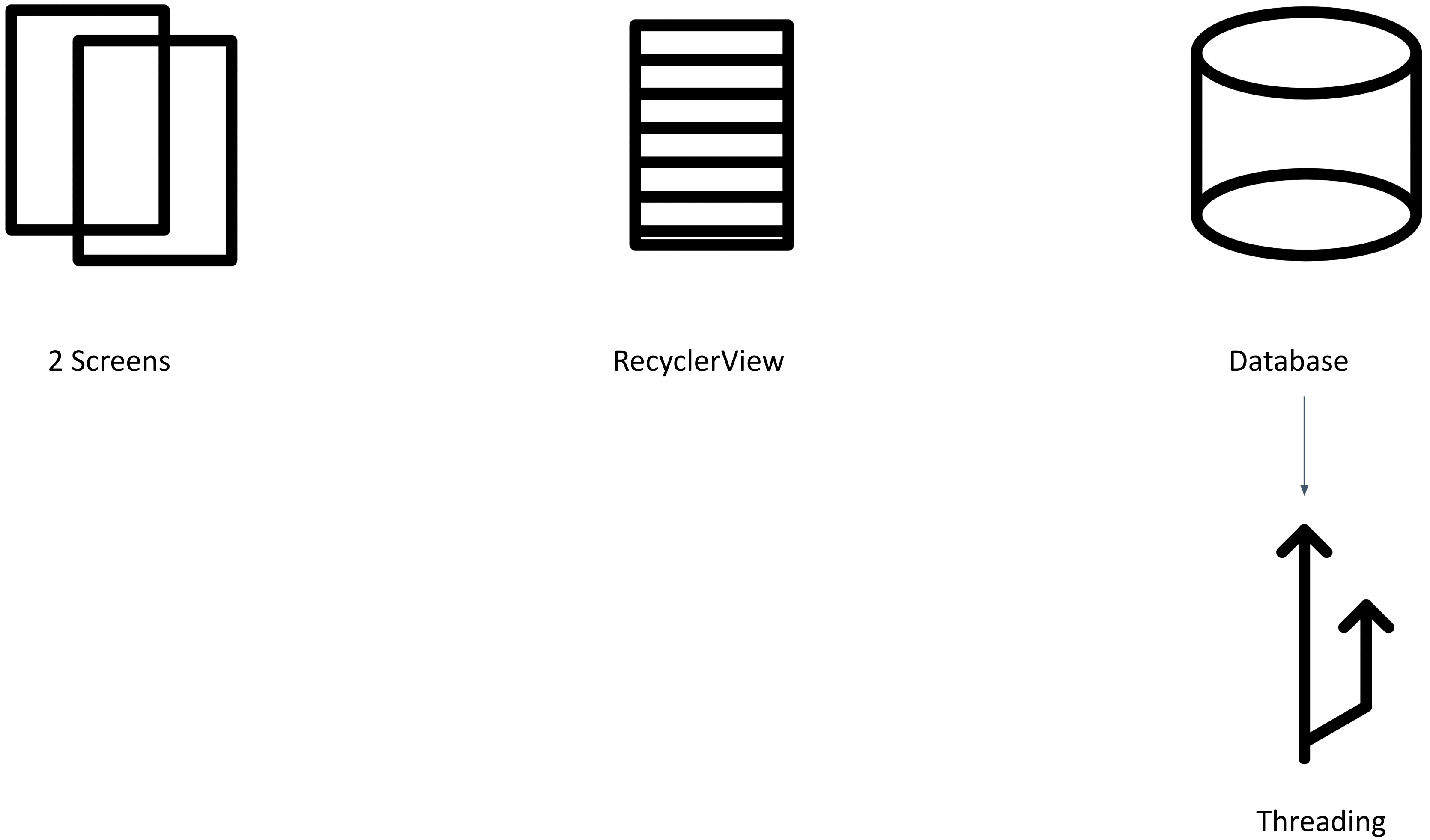


Assignment

Movie Database

Individual

Assignment



2 Screens

RecyclerView

Database

Threading

Assignment

Screens *(lecture 1 and 3)*

- One screen with movies in the database
- One screen with movie info
- Can be done in any way you desire (fragments count as screens)

Recyclerview *(lecture 2)*

- Recyclerview implemented into a relevant context

Database *(lecture 2)*

- Any local or online database (can be an external database)
- Correct use of threading

Any further technology you wish to implement *(lecture 2 and 3)*

- Course topics that you feel provide value to the solution

Grading and deliverable

Report

- Maximum one normal page (3000 characters with spaces)
  - Figures and code snippets does not count towards your character count
- Focus on arguing what the chosen technologies provide towards the project.
- Doesn’t have to cover everything you have done
  - I can check your application to see if you have fulfilled the requirements if you would rather spend the report talking about a different technology.

Application

- Either GitHub link or zip file
- Must be able to build and run in Android Studio
- Used as a basis to check on correct implementation of technologies

Further Reading

Further reading:

- https://developer.android.com/guide/components/fundamentals
- https://developer.android.com/guide/components/intents-filters

Github with examples:

https://github.com/monoe18/Mobile-Examples