Android Application Architecture

EE5415

Mobile Apps Design and Development

Message: Course Project Grouping

- This is just a friendly reminder for sending your group project members list to the instructor's email address at ee5415@gmail.com
- We need this information to arrange the proposal presentation on week 5.
- On or before 11.00pm on 5th Jan 2020 (Friday).

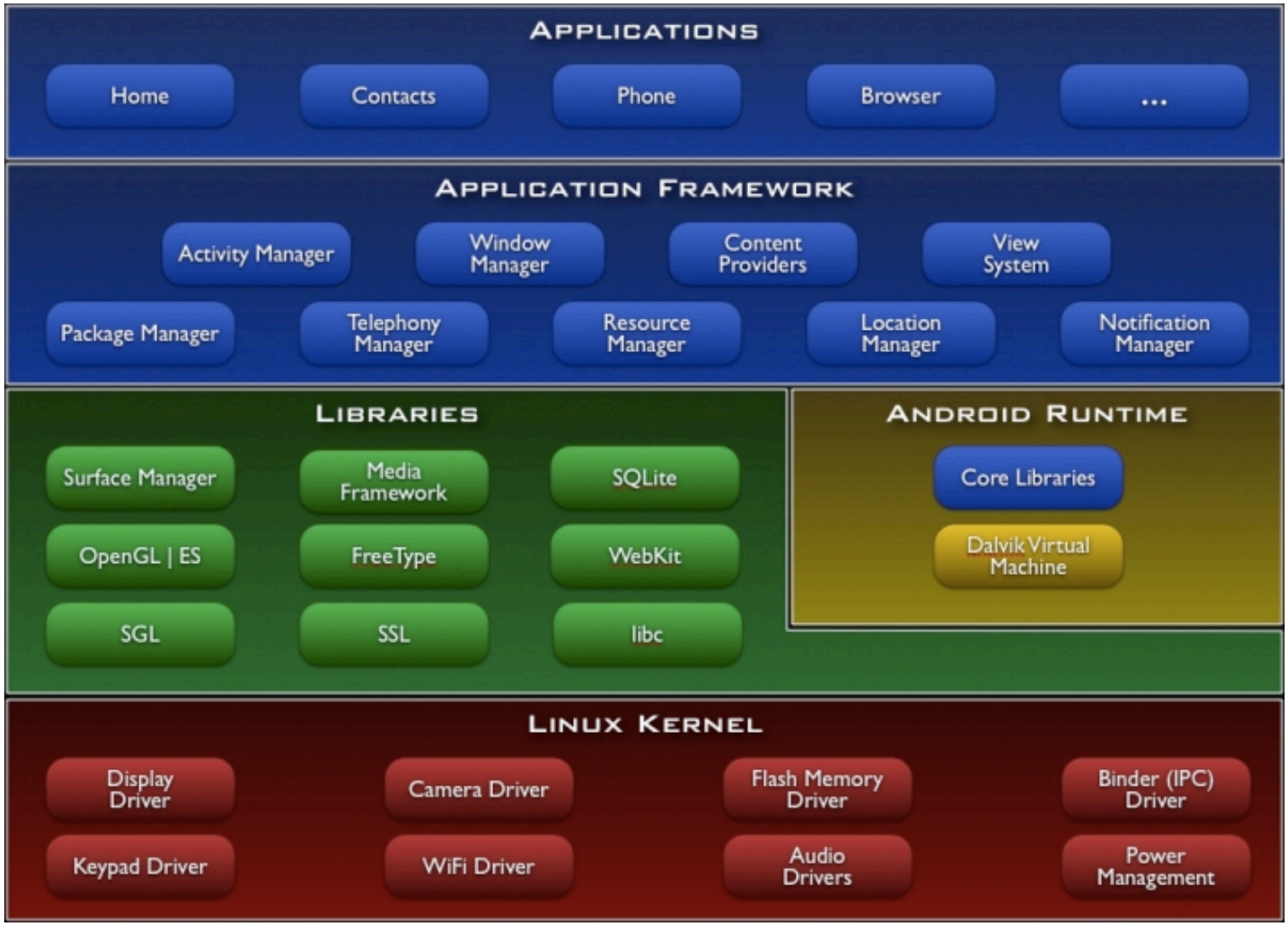
Message: Lab01 Homework

- This is just a friendly reminder for submitting your Android App Evaluation Homework of Lab01.
- Students are required to submit their app evaluation report in MS-Words format as an attachment to EE5415 course gmail account at ee5415@gmail.com **on or before 11.00pm of 5th Feb 2019.**
- The subject of your email should satisfy the following format:
  - EE5415 Lab01 Homework - Student Name (Student Number)
  - Example: EE5415 Lab01 Homework - Chan Chi Man (51234567)

Review of Last Lecture

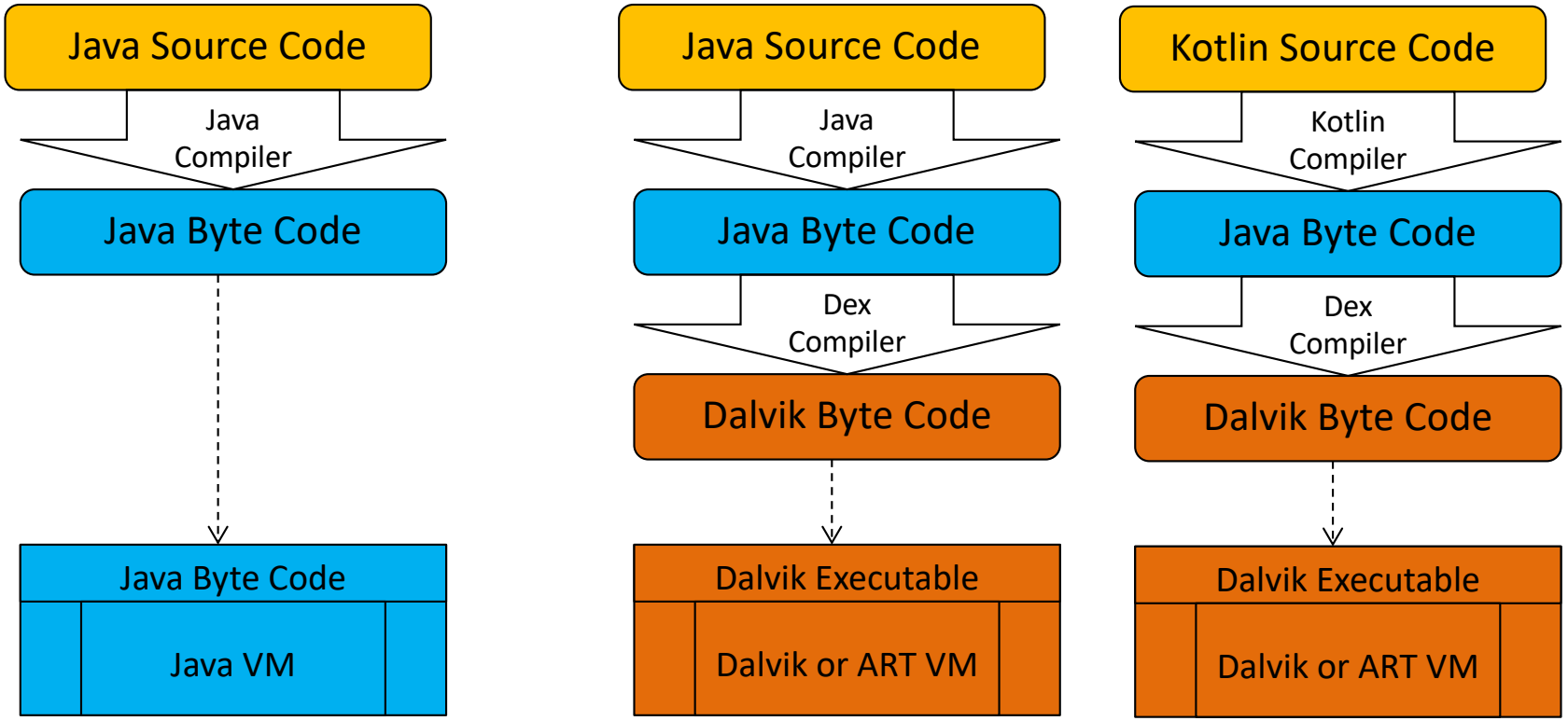
- **Android Software Stack** consists of
  - Operating System (Linux Kernel)
  - Middleware (Native Libs, Java Libs, Runtime (ART or DVM), Application Frameworks)
  - Key Applications (Home, Contacts, Phone, etc)
- **Android Studio**
  - It is the official IDE for Android app development
  - Setup Android Studio in your PC or laptop

Android Software Stack



Java and Kotlin

- Android app can be developed by Java or Kotlin and they are JVM (Java Virtual Machine) Languages.
- Their source code are compiler to Dalvik Byte Code (DEX) that execute on Dalvik or ART VM.



Contents

- **Introduction of Kotlin Programming Language**
- **Android App Architecture**
  - Understanding the Android Components for building Android Apps :
    - Activity
    - Service
    - Broadcast Receiver
    - Content Provider
    - Intent
- **Lab02: To build a simple BMI Calculator app**

Introduction to Kotlin Programming Language

Kotlin Highlights

- Kotlin is now Official Android Language
  - It is developed by JetBrains
    - JetBrains built Intellij IDEA and Android Studio is built on top of Intellij IDEA
- Kotlin is Open Source project primarily under Apache 2
  - Which again promotes developer ecosystem to be united and contribute more freely

Is it meant to Replace Java?

- **Not 100% but it is expected most of the Android apps will be developed by Kotlin**
- Kotlin is just more flexible Java
- It is a more powerful language
- The Android apps can be built on Kotlin along with using Java and C++ alongside

Why Kotlin?

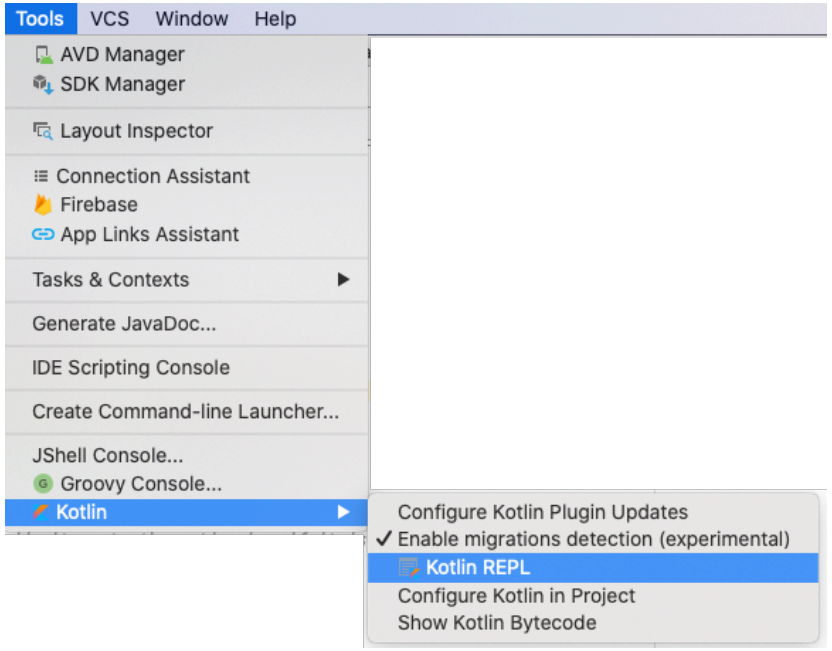
- Kotlin is just more powerful Java
  - Expressive
  - Concise
- Safety Features
  - Immutable
  - Nullability: No more null pointer exceptions
    - Makes your apps healthier and safer
    - Ensures more performance

Some other features of Kotlin

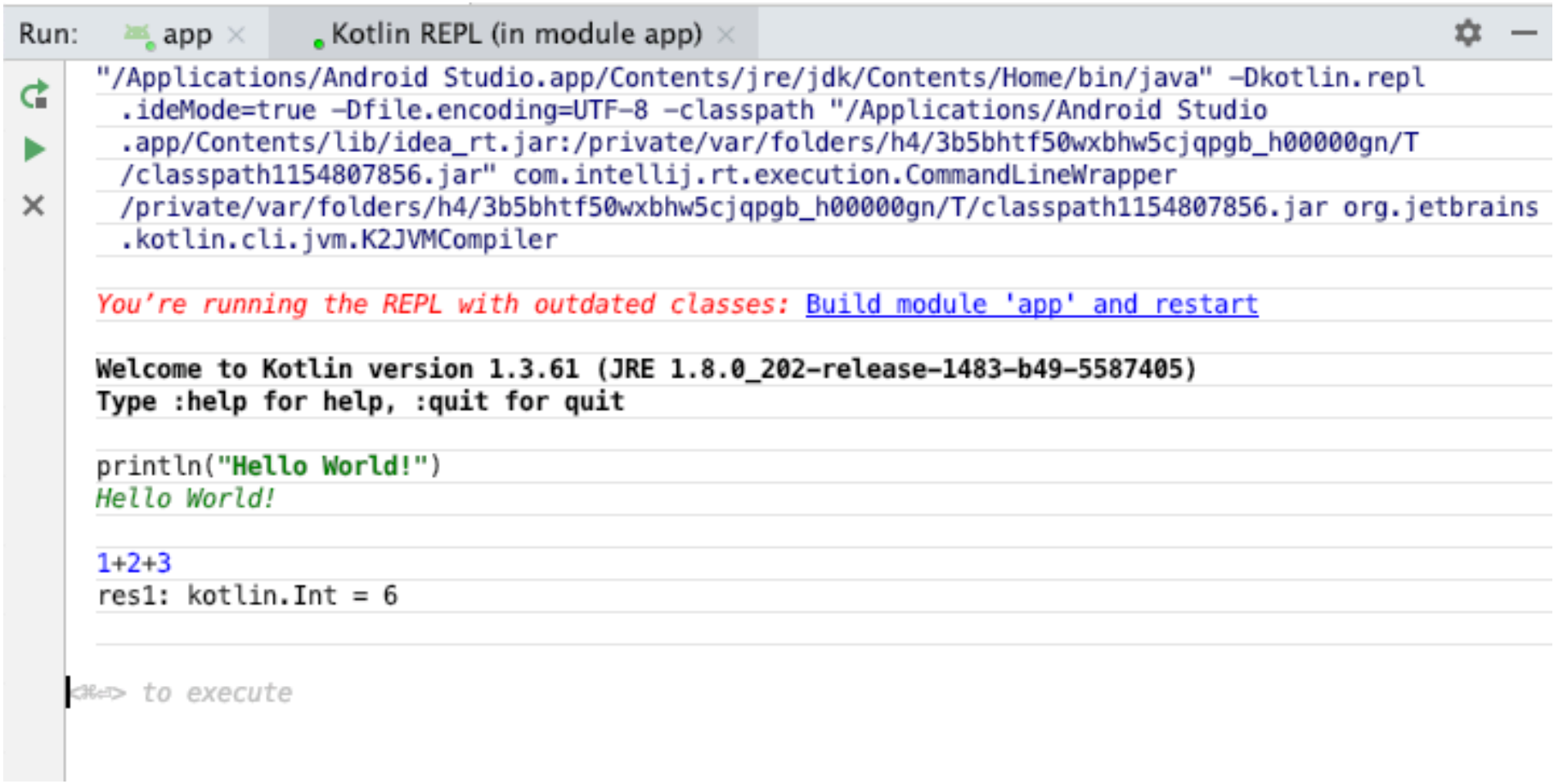
- It's a JVM Language
- Its syntax if more expressive and concise
- It supports
  - High-order functions
  - Lambda expressions
  - String Templates
  - and so on…
- Kotlin and Java are Interoperable
  - They both can be used together in the same project
  - You can call into Java language from Kotlin and you can call into Kotlin from the Java language

Kotlin for Android Developers

- Learn Kotlin in Android Studio using Kotlin REPL (Read Evaluate Print Loop)
- Variables and Data Types
- Null Safety
- if Expressions
- when Expressions
- Functions
- Class



Learn Kotlin in Android Studio using Kotlin REPL



Kotlin Variables and Data Types

- **Immutable Variable**

```
val name:String = “Peter”
val name = “Peter”
```

- **Mutable Variable**

```
var age = 24
var age:Int = 24
var height:Double = 1.756
var job:String = ”Developer”
var isDeveloper: Boolean = true
```

https://www.youtube.com/watch?v=mUJzEGSL_fI

Kotlin Null Safety

- Null cannot be a value of a non null type data

```
val name: String = null
```

This is not allowed in Kotlin.

- Null Safety

```
val name: String? = null
str!!.length
    error
str?.length
    null
val name: String? = "Peter"
str?.length
    5
val strlength = name?.length ?:0
```

https://www.youtube.com/watch?v=80C7e9SHsK0

Kotlin if Expression (1)

- In Kotlin, if is an expression, i.e. it returns a value. Therefore there is no ternary operator (condition ? then : else)

```
// Traditional usage
var max = a
if (a < b) max = b

// With else
var max: Int
if (a > b) {
    max = a
} else {
    max = b
}

// As expression
val max = if (a > b) a else b
```

https://kotlinlang.org/docs/reference/control-flow.html

Kotlin if Expression (2)

- if branches can be blocks, and the last expression is the value of a block:

```
val max = if (a > b) {
    print("Choose a")
    a
} else {
    print("Choose b")
    b
}
```

Kotlin when Expression (1)

- when replaces the switch operator of C-like languages. In the simplest form it looks like this

```
when (x) {
    1 -> print("x == 1")
    2 -> print("x == 2")
    else -> { // Note the block
        print("x is neither 1 nor 2")
    }
}
```

https://kotlinlang.org/docs/reference/control-flow.html

Kotlin when Expression (2)

- If many cases should be handled in the same way, the branch conditions may be combined with a comma:

```
when (x) {
    0, 1 -> print("x == 0 or x == 1")
    else -> print("otherwise")
}
```

- We can use arbitrary expressions (not only constants) as branch conditions

```
when (x) {
    parseInt(s) -> print("s encodes x")
    else -> print("s does not encode x")
}
```

Kotlin function

- Functions are declared with the fun keyword.
- For the parameters, you must declare not only their names, but also their types, and you must declare the type of the value the function is intending to return.

```
fun happyBirthday(name: String, age: Int): String {
    return "Happy ${age}th birthday, $name!"
   }
//Calling
val greeting = happyBirthday("Anne", 32)
```

Kotlin Class

- Like Java, Kotlin also allows to create several objects of a class and you are free to include its class members and functions.

```
class myClass {
   // property (data member)
   private var name: String = "Tutorials.point"

   // member function
   fun printMe() {
      print("You are at the best Learning website Named-"+name)
   }
}
fun main(args: Array<String>) {
   val obj = myClass() // create obj object of myClass class
   obj.printMe()
}
```

https://kotlinlang.org/docs/reference/classes.html

Android App Architecture

Android Apps

- Written in Java and Kotlin
  - It's possible to write native code such as C

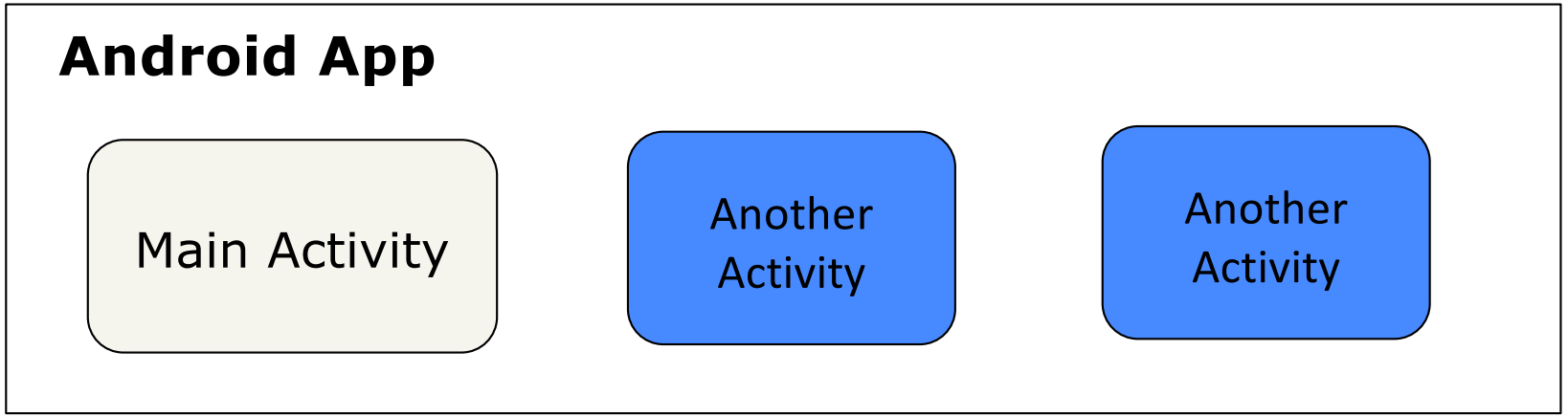
- Good separation from other apps:
  - Each app runs in its own process
  - Each process has its own separate virtual machine (VM)
  - During the installation, each app is assigned a unique Linux user ID
    - by default files of that app are only visible to that app (can be explicitly exported)

Android Components

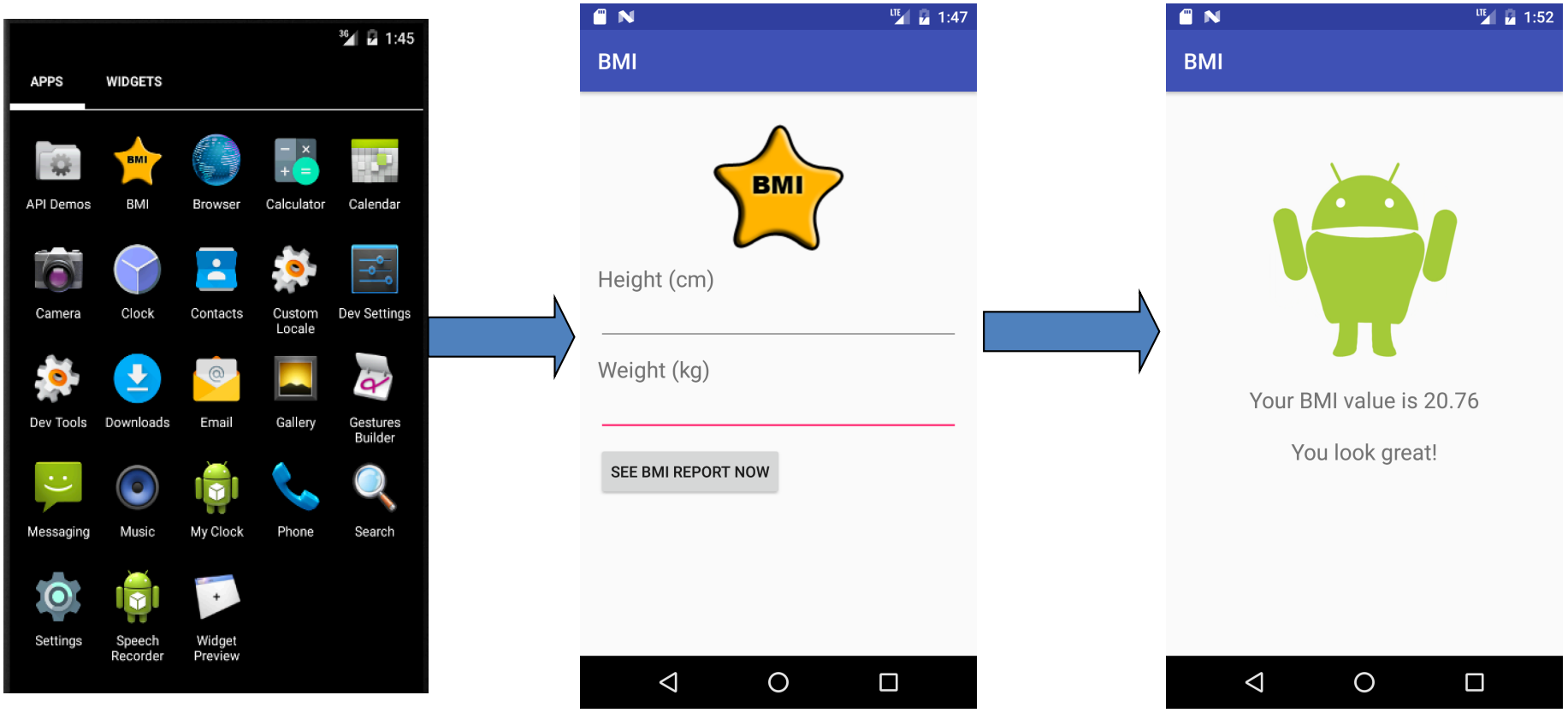
- **Activities**
  - An *activity* represents a single screen with UI elements to interact with users.
- **Services**
  - A long running background process without any user interaction, e.g. Alarm Service, Media Player, etc.
- **Broadcast Receivers**
  - Broadcast receivers are registered for system announcements.
- **Content Providers**
  - Content provider is used to share the data between multiple apps.

Activities

- An Activity represents a screen or a window for user interaction
  - It is basic component of most Android apps
  - Most apps have several activities that start each other as needed
  - Each activity is implemented as a subclass of the base Activity class



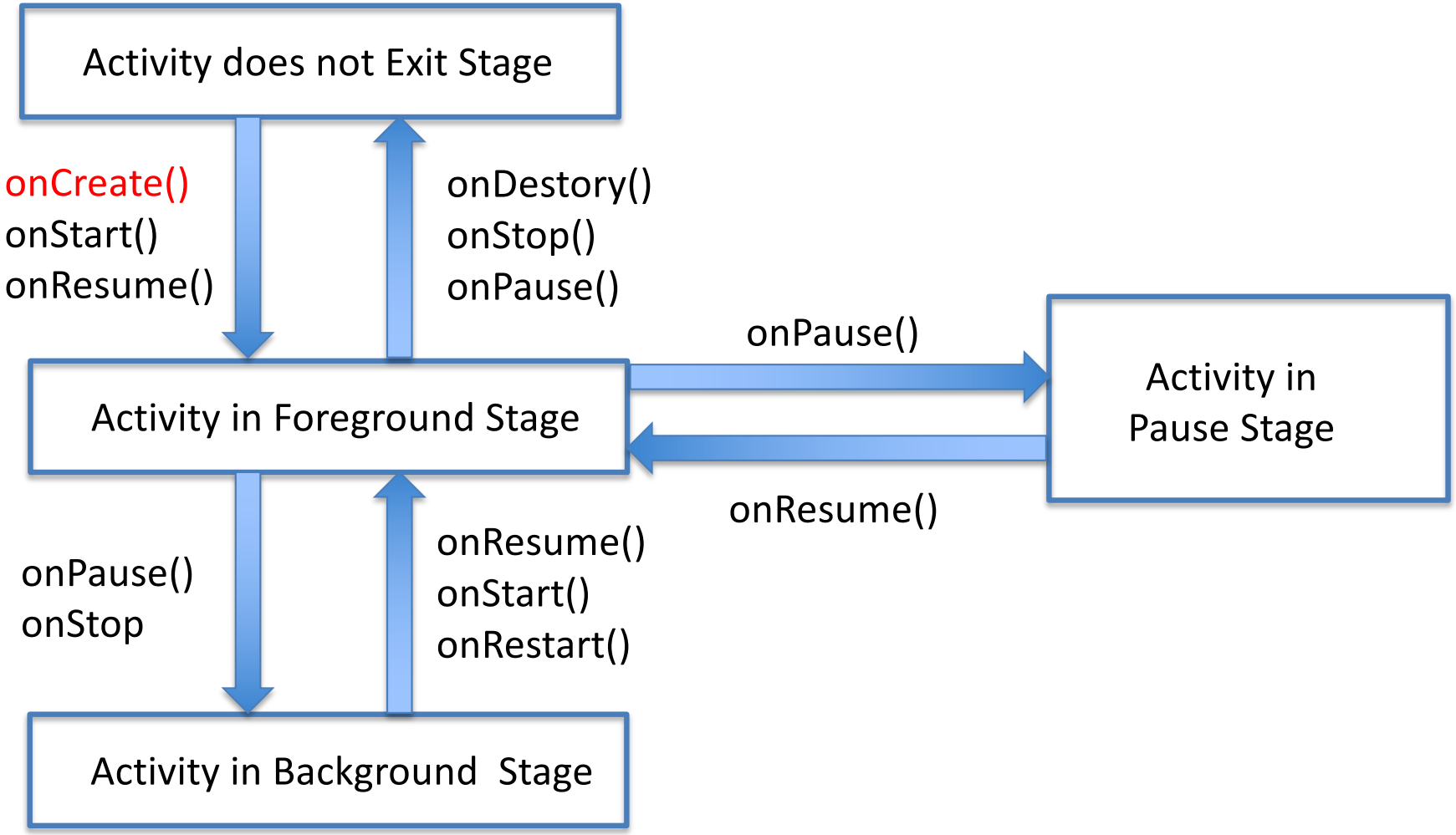
Activities start each other



Activity Life Cycle

- Activity is having 4 states
  - Does not Exist
  - Foreground
  - Pause
  - Background
- Following are the 7 major methods in Activity class which will participate in Activity lifecycle
  - onCreate(), onStat(), onResume(), onRestart(), onPasue(), onStop(), and onDestory()

Activity Lifecycle



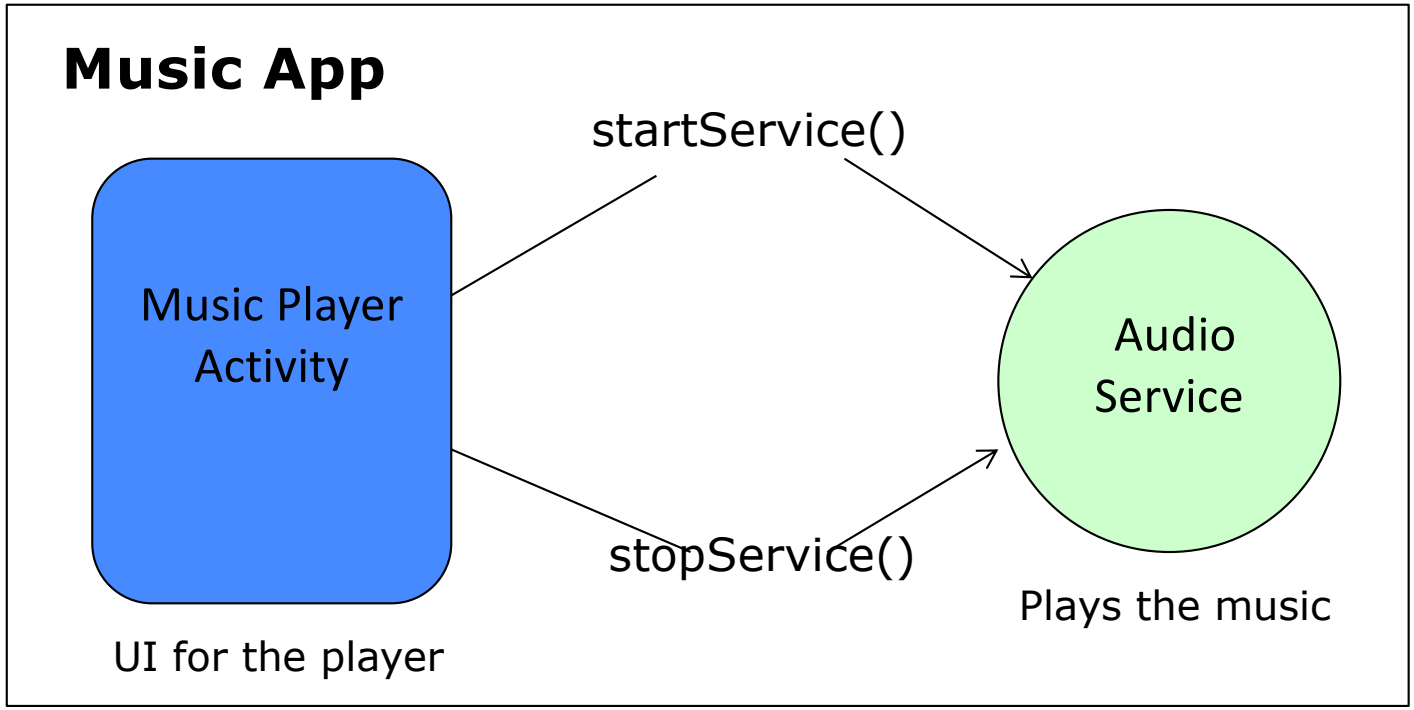
Similar to main() method in C/C++/Java in Android Activity onCreate() method be invoked first, so override onCreate() method.

Activity Life-Cycle Examples

- **Start for the first time**
  - OnCreate() => OnStart() => OnResume()
- **Open Another Activity and then press the Back button**
  - OnPause() => OnStop() => OnRestart() => OnResume()
- **Rotate the Screen**
  - OnPause() => OnStop() => OnDestroy() => OnCreate() => OnStart() => OnResume()
- **Press the Home button**
  - OnPause() => OnStop()

Services

- Services are processes that runs in the background
  - They can be started and stopped. Services doesn't have UI.
- To perform long-running operations
- To support interaction with remote processes

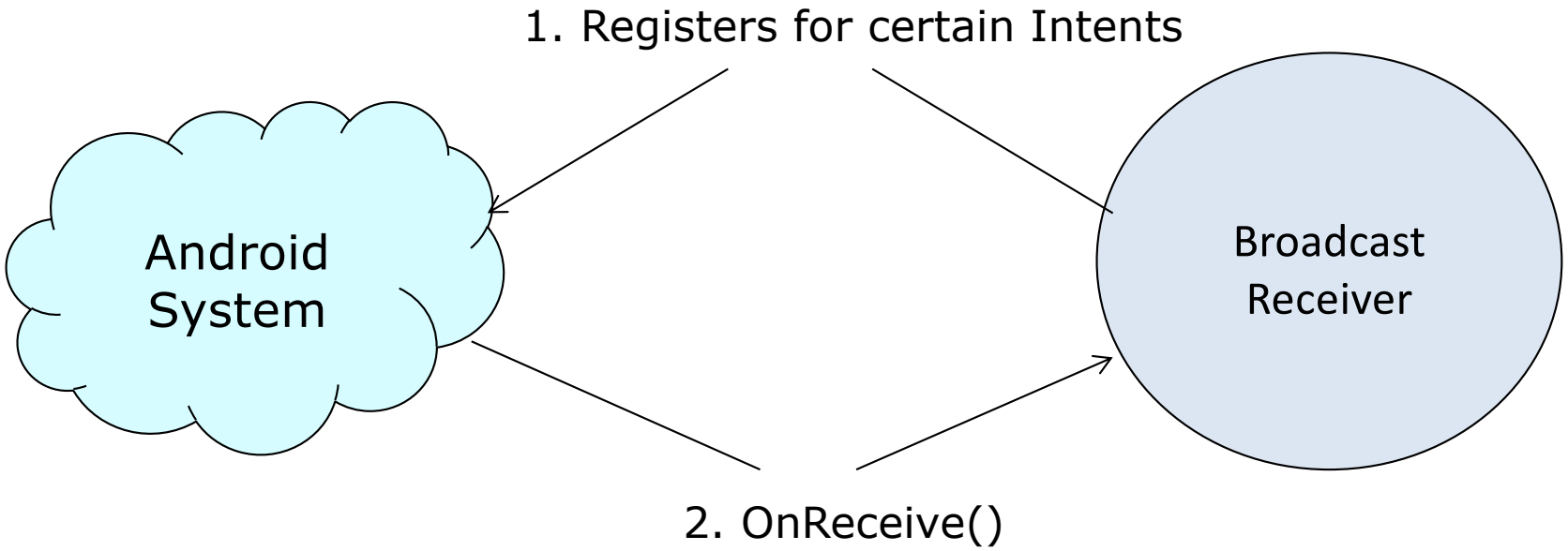


Service Life Cycle

- Service also has a lifecycle, but it's much simpler than activity's
- An activity typically starts and stops a service to do some works for it in the background, such as play music.

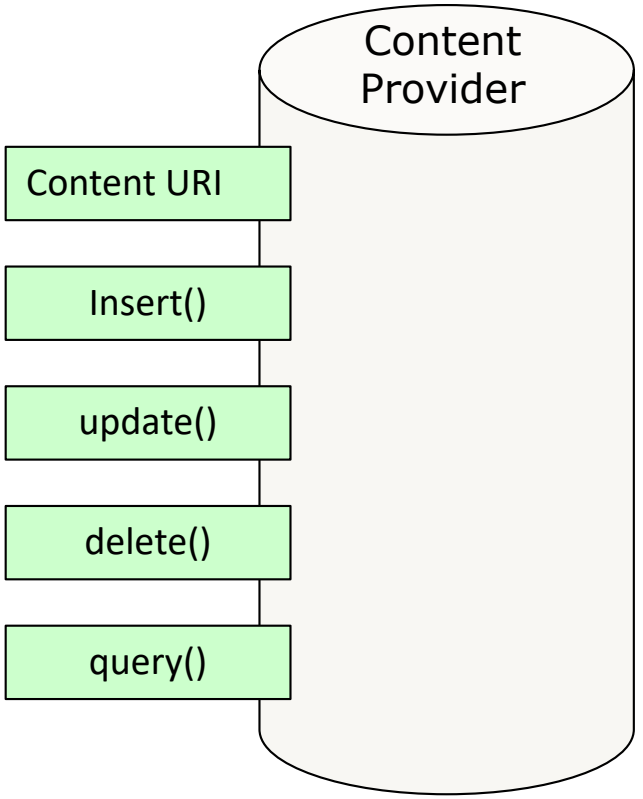
Broadcast Receiver

- Broadcast Receivers are registered for system announcements
  - Component that listens for and responds to events
  - The subscriber in publish/subscribe pattern
  - Events represented by the Intent class and then broadcast
  - Broadcast Receiver receives and responds to broadcast event
  - Great for listening system events such as SMS messages.
- E.g.: Headset plugin, charger connected/disconnected, etc

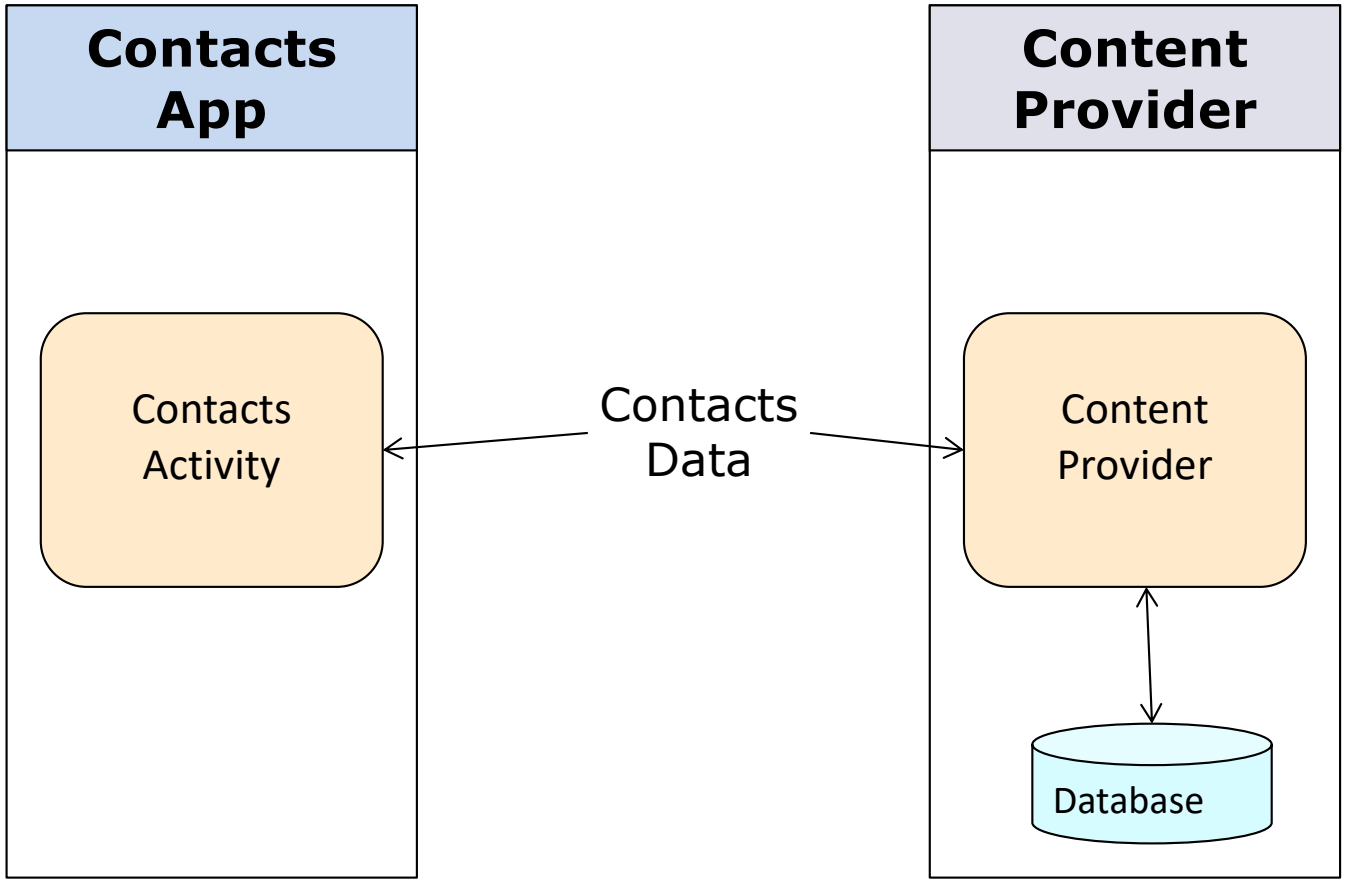


Content Providers

- Content Provider is used to share the data between multiple apps.
  - In Android one of the security feature is we cannot access the other apps data into our app directly, but if the app is providing content provider then we can access the data into our app.
- Examples of built-in Content Providers are
  - Contacts, Callog, Media, Setting, Calendar, etc

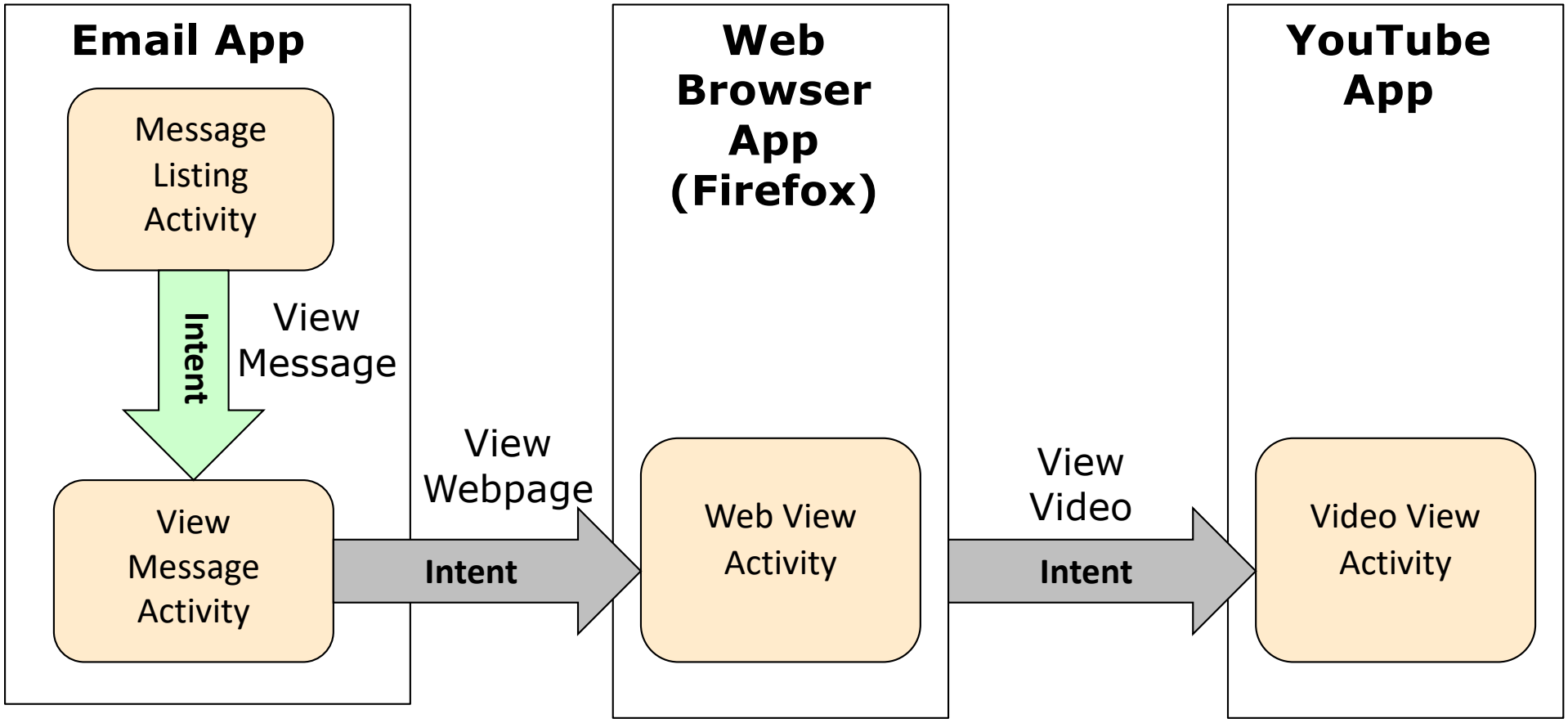


Content Provider Example



Intents

- An intent is an Intent object with a message content.
- Intents represent events or actions.
- Intents can be implicit or explicit



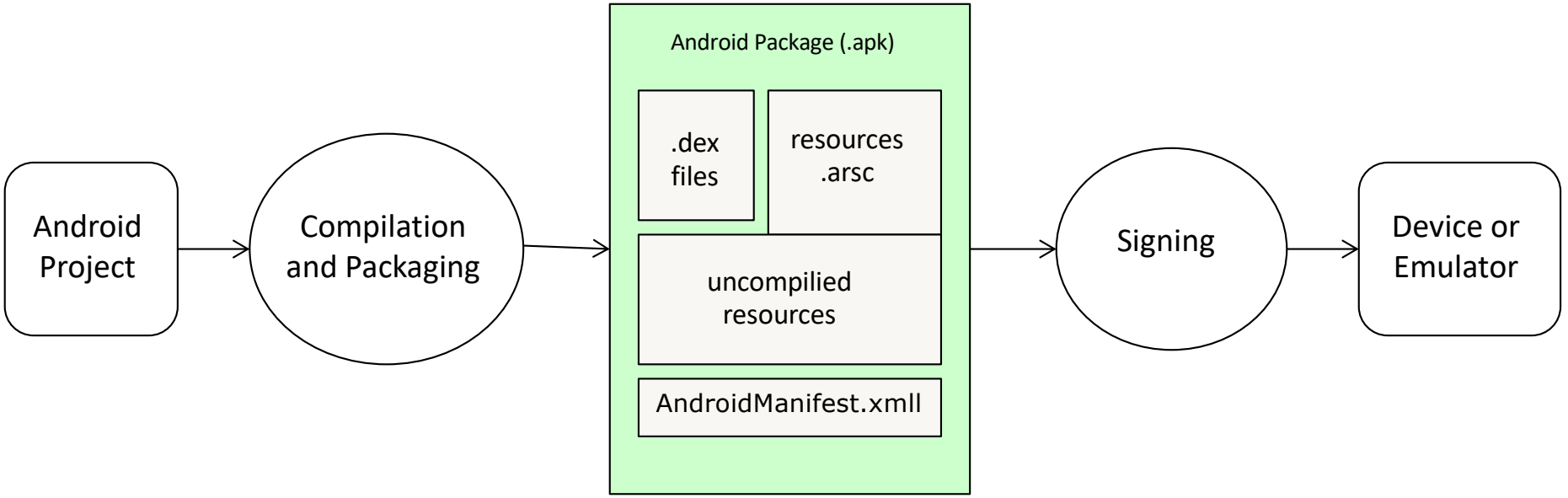
Advanced Topics

- Material Design
- Web Services [SOAP & REST]
- XML, JSON, GSON
- Firebase
- Google Maps
- Google Cloud Messing
- Graphics Programming

App Development Process

- **Prototype**
  - Though you have an idea to develop an app. It requires conceptualization and prototyping.
  - Design the functions and corresponding user interfaces
- **Create an Android Project**
  - You can start creating the android project by feeding the app name, ownership, app ID and version compatibilities
- **Debugging and Testing**
- **Analyze and Performance Evaluation**
  - You need to keep an eye on all the resources used in the app. Size matters the most and the resources should be optimized. We also need to evaluate the performance of the functionalities of the app.
- **Release**
  - The app has to be released with a certificate duly signed. The app has to be published on the Google Play Store for user download.

Building an Android App



See: http://developer.android.com/guide/developing/building

Android Project Structure

- **app**
  - **manifest**
    - AndroidManifest.xml (App Configuration)
  - **java**
    - pkg name >> *.java
  - **res**
    - drawable >> *.png (graphic images)
    - layout >> *.xml (UI XML)
    - mipmap >> app icon
    - values >> string.xml, styles.xml, dimensions.xml, etc
    - menu >> menu.xml
    - raw >> *.mp3, *.mp4
  - **build>>generate>>source>pkg name>r>debug>R.java** (Providing references for Java programs to access the resources in res)
  - **build>>outputs>>apk>>*.apk** (apk is termed as Android Application Package it is the installation file in Android platform, which include .dex, res, assets, lib, AndriodManifest.xml)
- **gradle** (A build tool for Android application)
  - local.properties (locations of SDK, NDK, etc)
  - gradle.properties (buildToolsversion, minSdkVersion, targetSdkVersion, versioncode, versionName, dependenices)

Creating an Android App

1. Define Resources
2. Implement Application
3. Classes
4. Package Application
5. Install & Run Application

1. Defining Resources

- Resources are non-source code entities
- Many different resource types, such as
  - Layout, Strings, Images, Menus, & Animations
- Allows apps to be customized for different devices and users

- See: http://developer.android.com/guide/topics/resources

Strings

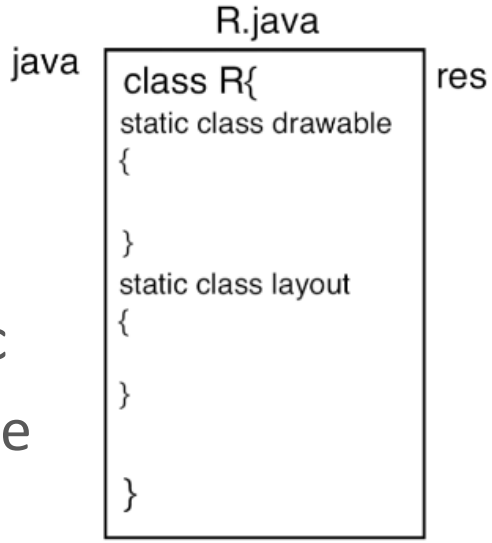
- **Types: String, String Array, Plurals (Quantity Strings)**
- Typically stored in folder of **`res/values/`**
  - `colors.xml`
  - `dimens.xml`
  - `strings.xml`
  - `styles.xml`
- Example: specified in `res/values/strings.xml` :
  - `<string name="hello">Hello World!</string>`
- Accessed by other resources as:
  - `@string/hello`
- Accessed in Java as:
  - `R.string.hello`

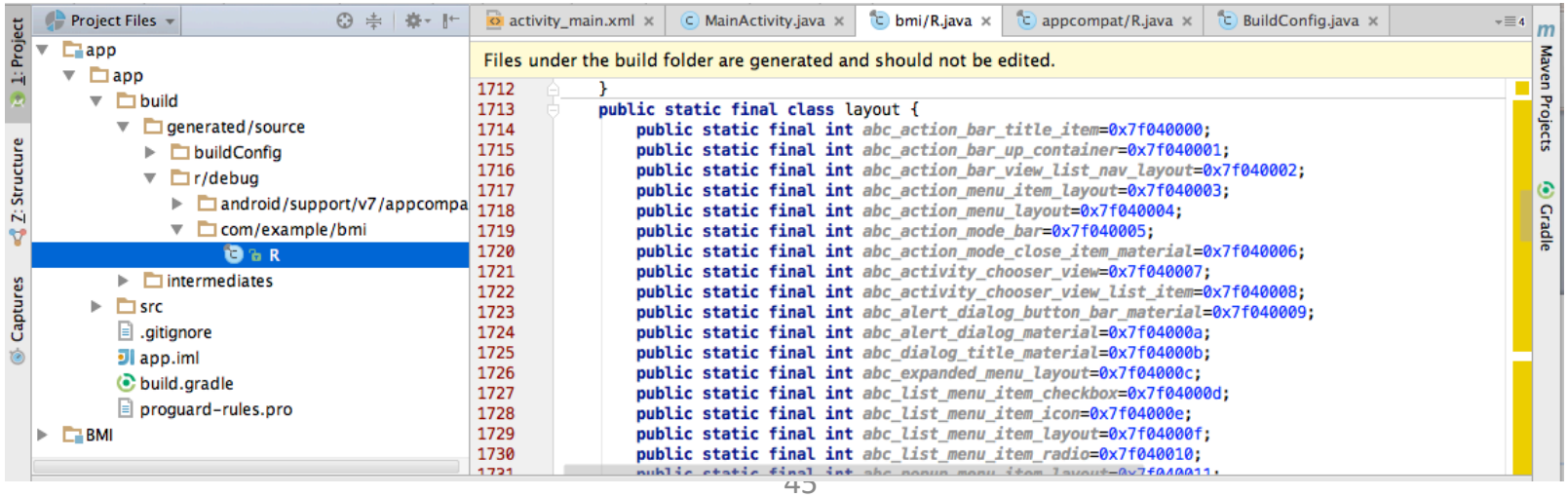
User Interface Layout

- UI Layout Specified in XML files
  - Some tools allow visual layout
- XML files typically stored in
  - `res/layout/*.xml`
- Accessed in Java as:
  - `R.layout.layout_name`
- Accessed by other resources as:
  - `@layout/layout_name`
- **Using multiple layout files**
  - Can specify different layout files based on your device's orientation, screen size, etc.

R.java

- R is termed as resource.
- R.java is an abstraction between the java file and different resources which are available in res folder
- For Every resource in res folder it will create a static integer field in R.java, with the help of this integrate field we can access the resource into Activity
- Project/app/build/generated/source/r/debug/com.android.'projectname'/R.java



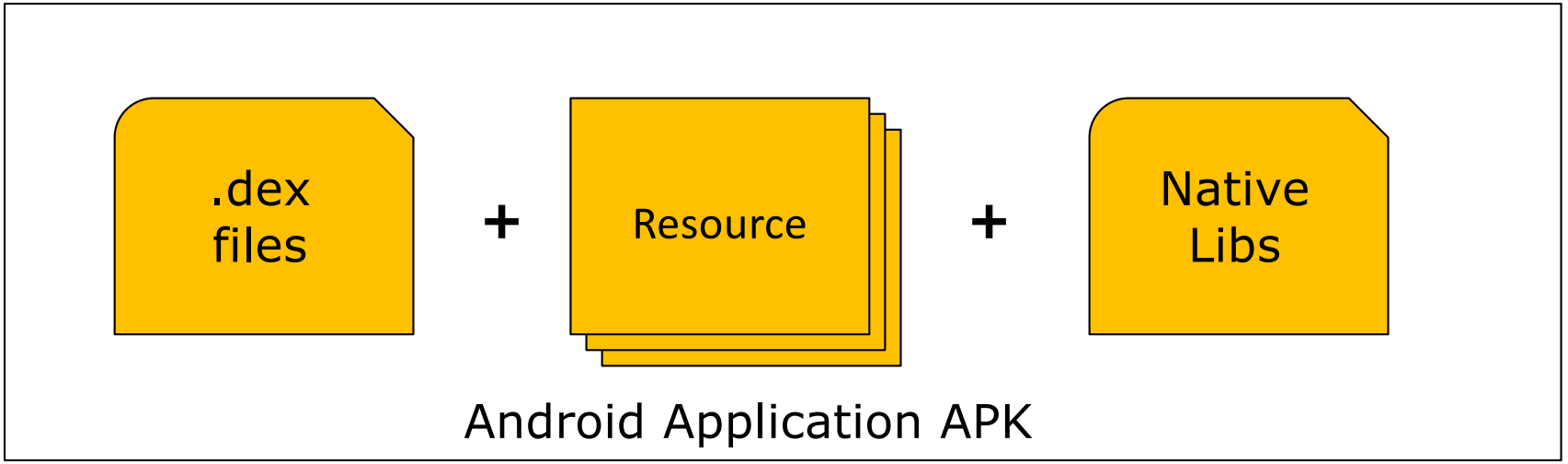


2. Implement Classes

- Usually involves at least one Activity
- Activity initialization code usually in **`onCreate()`**
- Typical `onCreate()` workflow
  - Restore saved state
  - Set content view
  - Initialize UI elements
  - Link UI elements to code actions

3. Package Application

- System packages application components & resources into a .apk file
- Developers specify required application
- Information in a file called AndroidManifest.xml



AndroidManifest.xml

Information includes:

- Application Name
- Application icon
- Declare components of your app
- Declare required features for your app
- State permissions required by your app
- State platform versions app is compatible with

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

  <uses-feature . . . />
  <uses-permission . . . />
  <uses-sdk . . . />

  <application . . . >
     <activity . . . >
        . . .
     </activity>
     <service . . . >
        . . .
     </service>
     <provider . . . >
        . . .
     </provider>
     . . .
   </application>

</manifest>
```

Intent Filters

- Declare Intents handled by the current application (in the AndroidManifest):

```
<?xml version="1.0" encoding="utf-8"?>
 <manifest . . . >
     <application . . . >
         <activity android:name="com.example.project.FreneticActivity"
                   android:icon="@drawable/small_pic.png"
                   android:label="@string/freneticLabel"
                   . . .  >
             <intent-filter . . . >
                 <action android:name="android.intent.action.MAIN" />
                 <category android:name="android.intent.category.LAUNCHER" />
             </intent-filter>
             <intent-filter . . . >
                 <action android:name="com.example.project.BOUNCE" />
                 <data android:mimeType="image/jpeg" />
                 <category android:name="android.intent.category.DEFAULT" />
             </intent-filter>
         </activity>
         . . .
     </application>
 </manifest>
```

Shows in the Launcher and is the main activity to start

Handles JPEG images in some way

4. Install & Run

- From Android Studio run in the emulator or device
- From command line
  - Enable USB debugging on the device
    - Setting > Applications > Development > USB Debugging
  - @adb install <path_to_apk>

Lab02:
BMI Calculator App

Objectives of Lab02

- To walk through a complete app design and development process
- To build a BMI calculator app
- Run your developed app (.apk) in the real device

App Design Process

1. To define the vision of the app
2. To define the functions, features and user interfaces to achieve the goals of your app
3. Implement the app by the Android Studio
4. Test the app on virtual device and then real devices
5. User evaluation and may be go back to one of the previous step.
6. Release to the market

Vision of the BMI App

- Provide user-friendly BMI calculation to let users know their health status
  - Clarity is the DNA of mastery.
  - We have to have very clear vision.
  - The rest of the app development are all based on this vision.

BMI Calculator

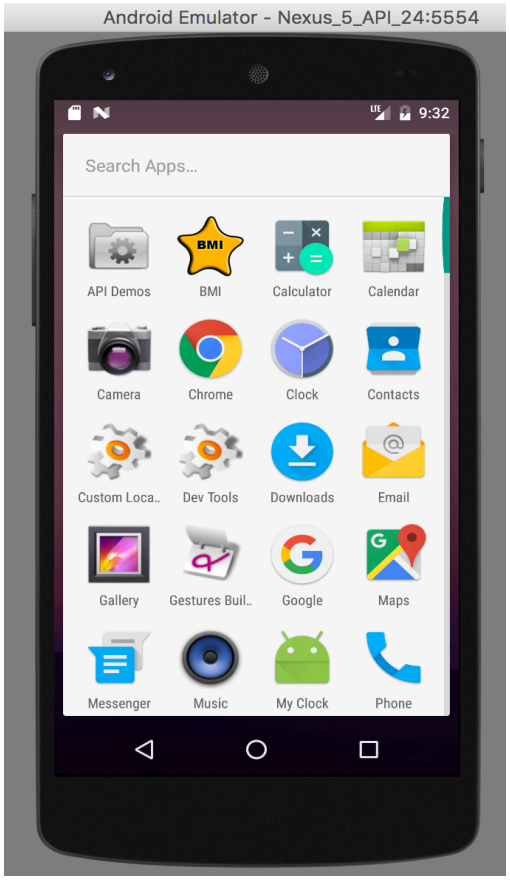
- The body mass index (BMI) is a statistical measure of body weight based on a person's weight and height.
- It is used to estimate a healthy body weight based on a person's height.

$$BMI = \frac{\text{mass}(kg)}{(\text{height}(m))^2}$$

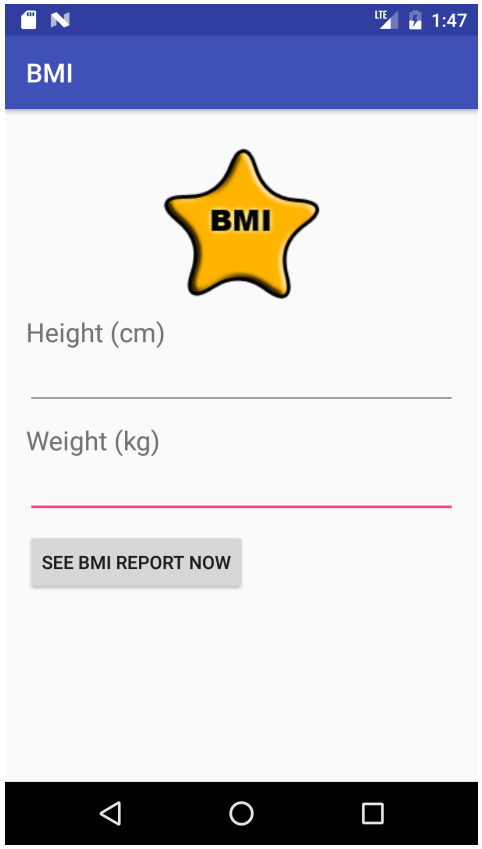
- Fat : BMI > 25
- Fit : 20 < BMI < 25,
- Thin : BMI < 20

User Interface Design

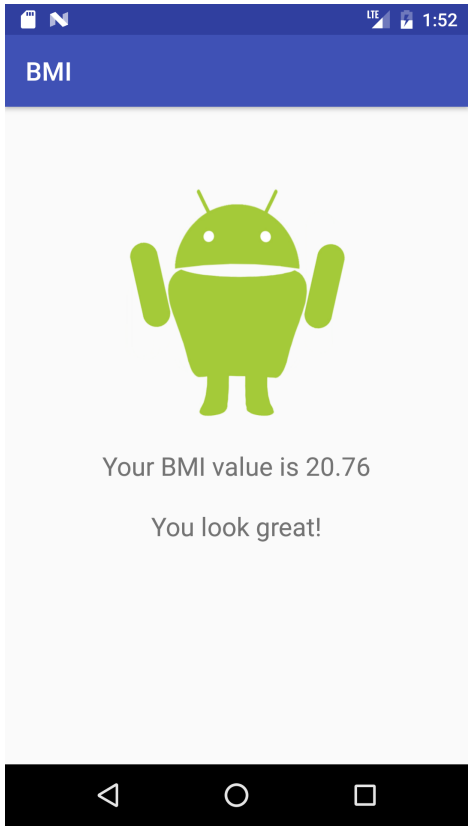
Icon of the BMI App



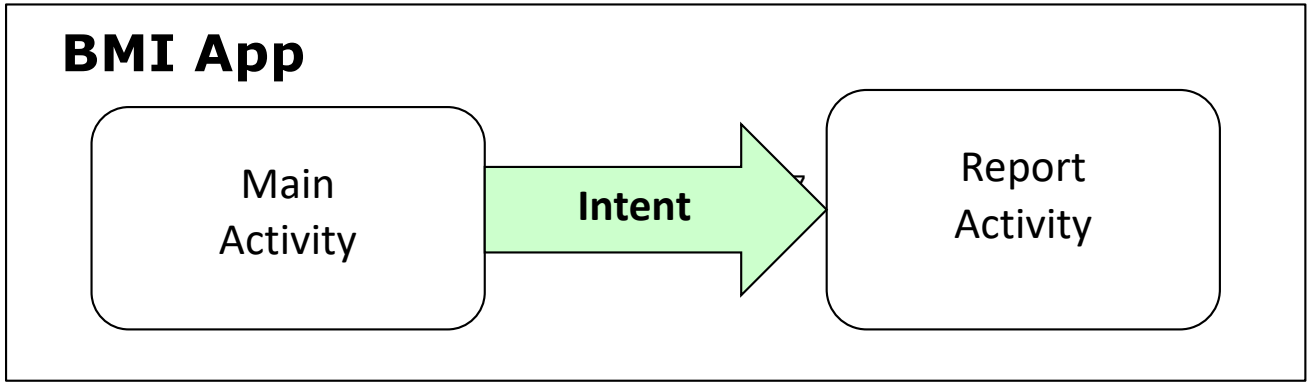
Front Page of the BMI App

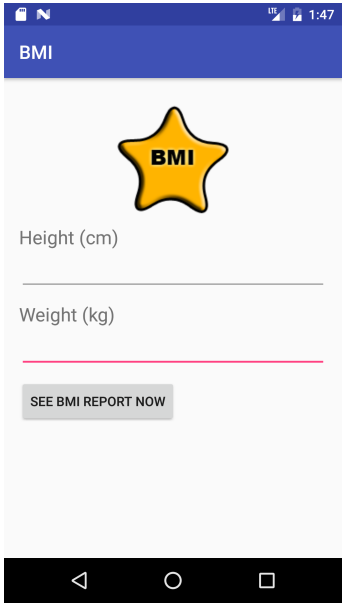


Report Page of the BMI App

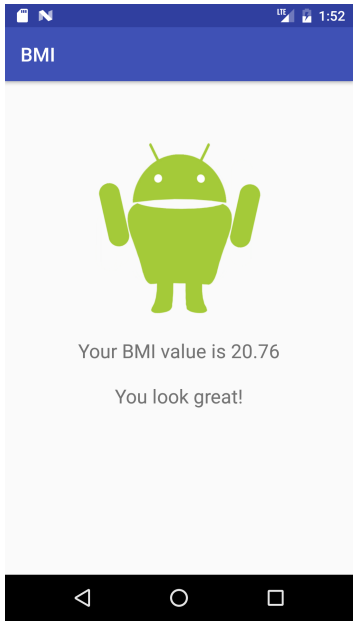


App Structure: Two Activities





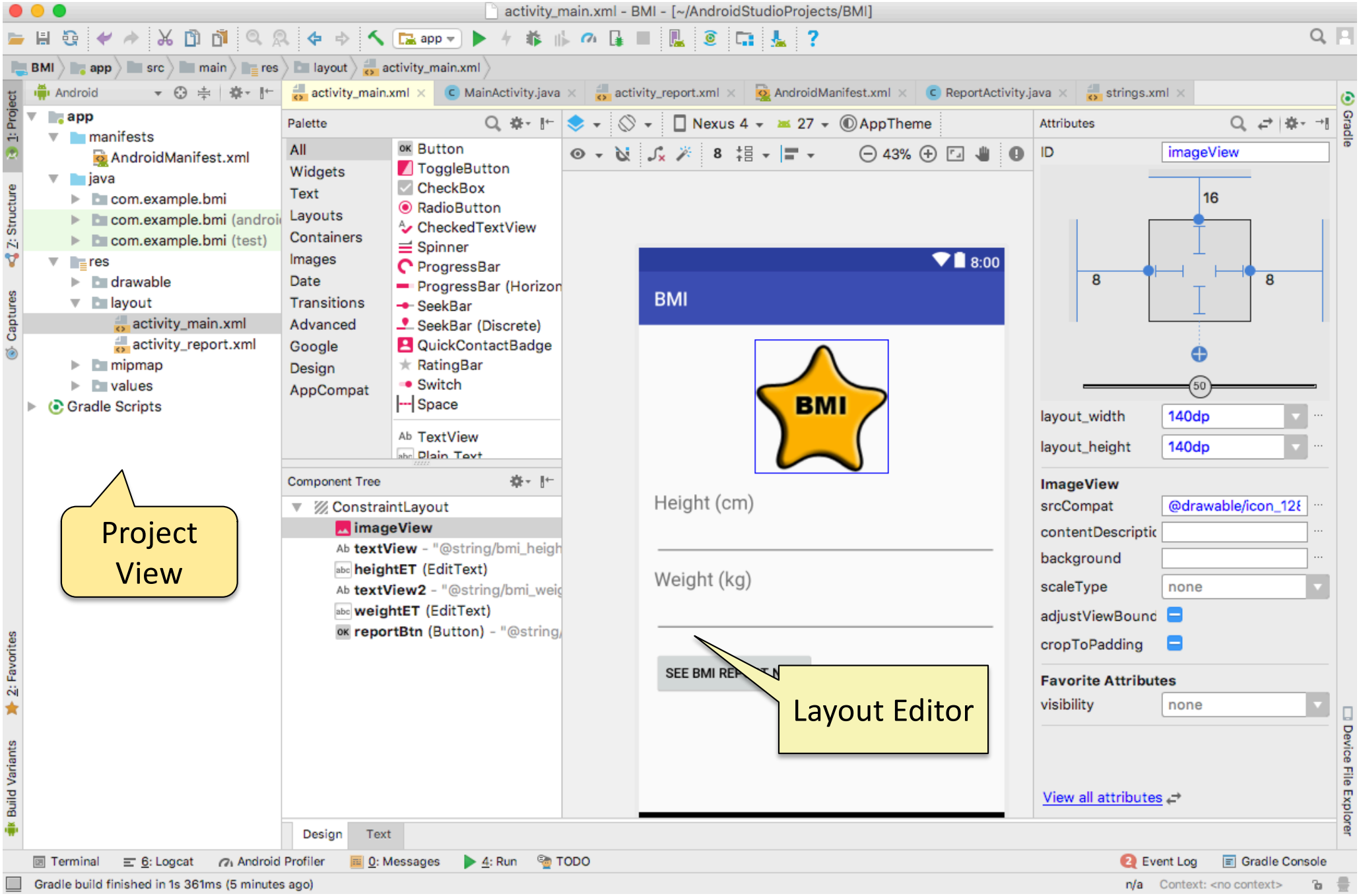
MainActivity.java
activity_main.xml



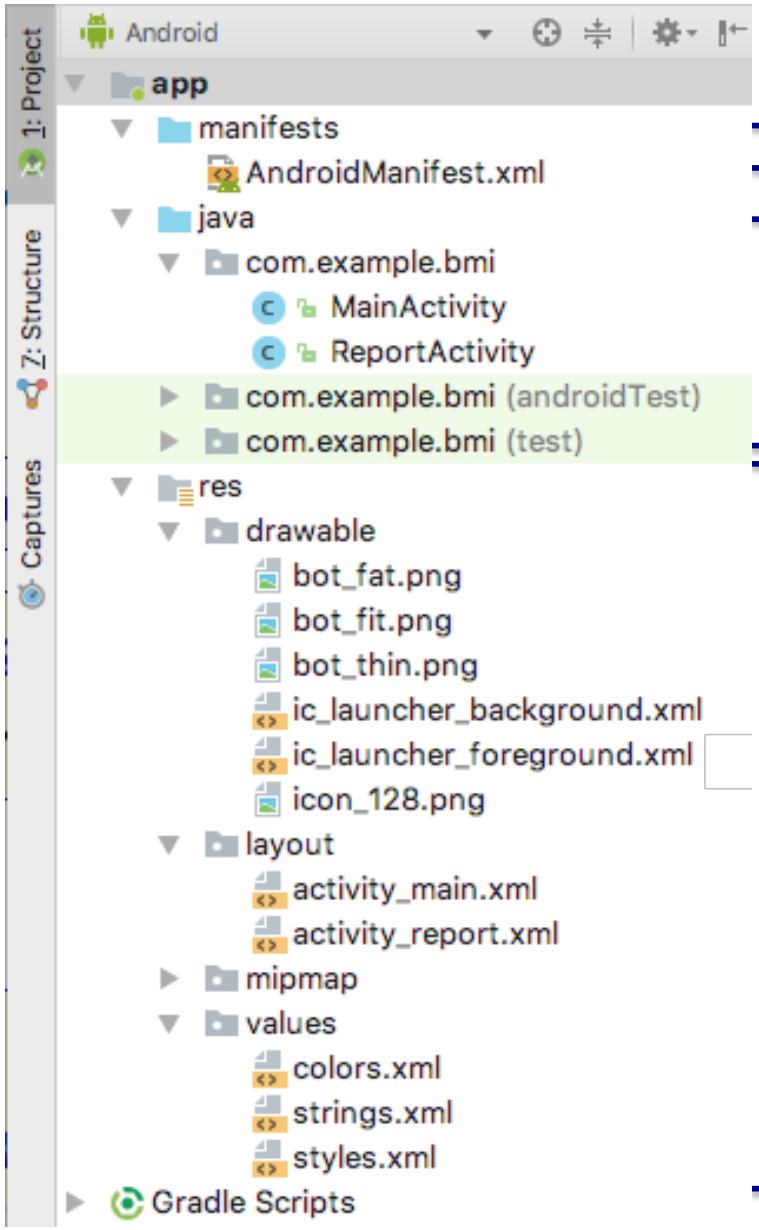
ReportActivity.java
activity_report.xml

Create a new project named BMI

- Choose your project : **Empty Activity**
- Application Name : **BMI**
- Package name : **com.example.bmi**
- Project location : use the default setting
- Language : **Java** or **Kotline**
- Minimum API level: **API 19: Android 4.4 (KitKat)**
- This configuration will create an Android Activity with following files:
  - Activity Name : **MainActivity**
  - Layout Name : **activity_main**


activity_main.xml - BMI - [~/AndroidStudioProjects/BMI]
Project View
Layout Editor
Palette
Component Tree
Attributes
BMI
Height (cm)
Weight (kg)
SEE BMI REPORT
8:00
ID
imageView
layout_width
140dp
layout_height
140dp
ImageView
srcCompat
@drawable/icon_128
contentDescriptic
background
scaleType
none
adjustViewBounc
cropToPadding
Favorite Attributes
visibility
none
View all attributes
Design
Text
Gradle build finished in 1s 361ms (5 minutes ago)

Project Scope => Android View



Edit program permissions and overall settings

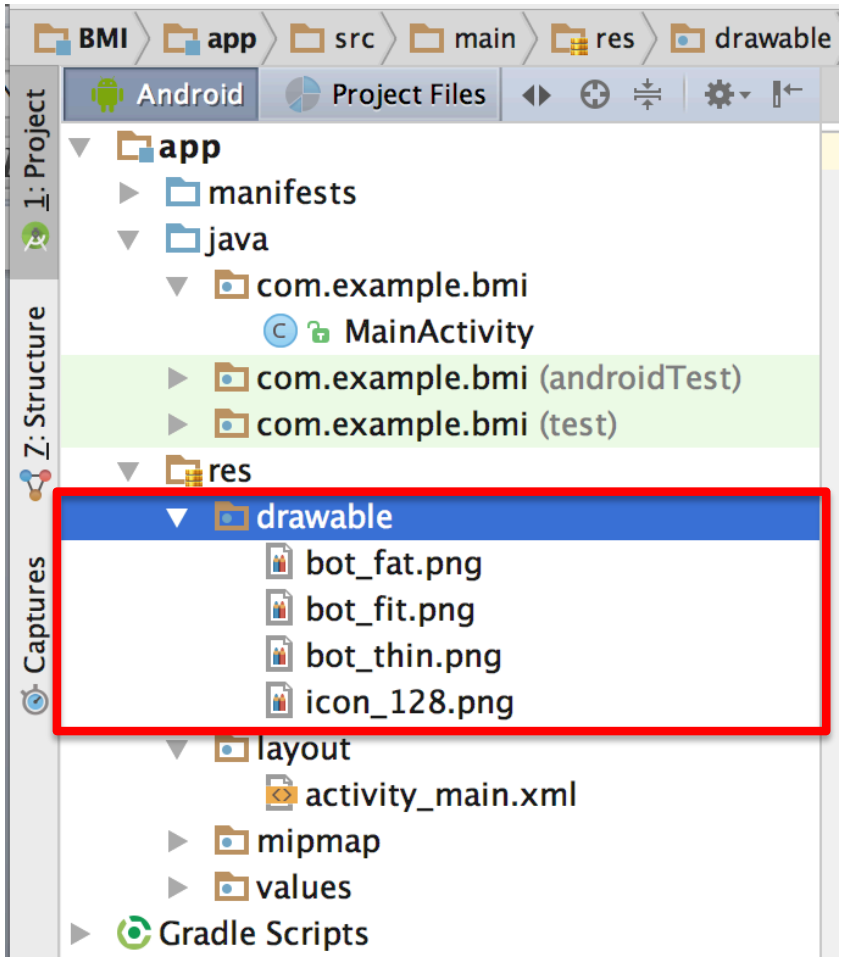
Java or Kotlin source files
Program new functions here

Resource files
Edit layout, define constants,
Insert external image files

Image/Icon Files in res/drawable Folder

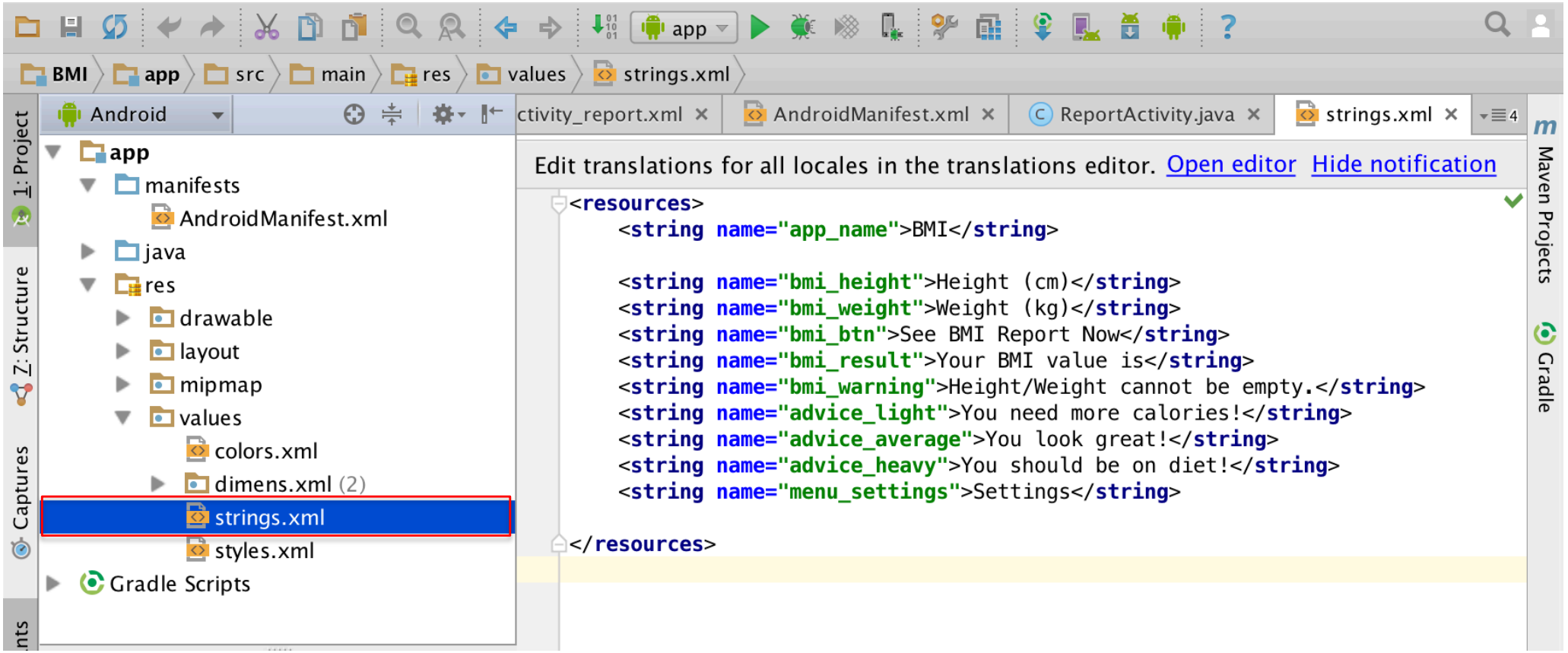
1. Download the sample image files from the course website with URL address

http://www.ee.cityu.edu.hk/~lmpo/ee5415/pdf/images.zip

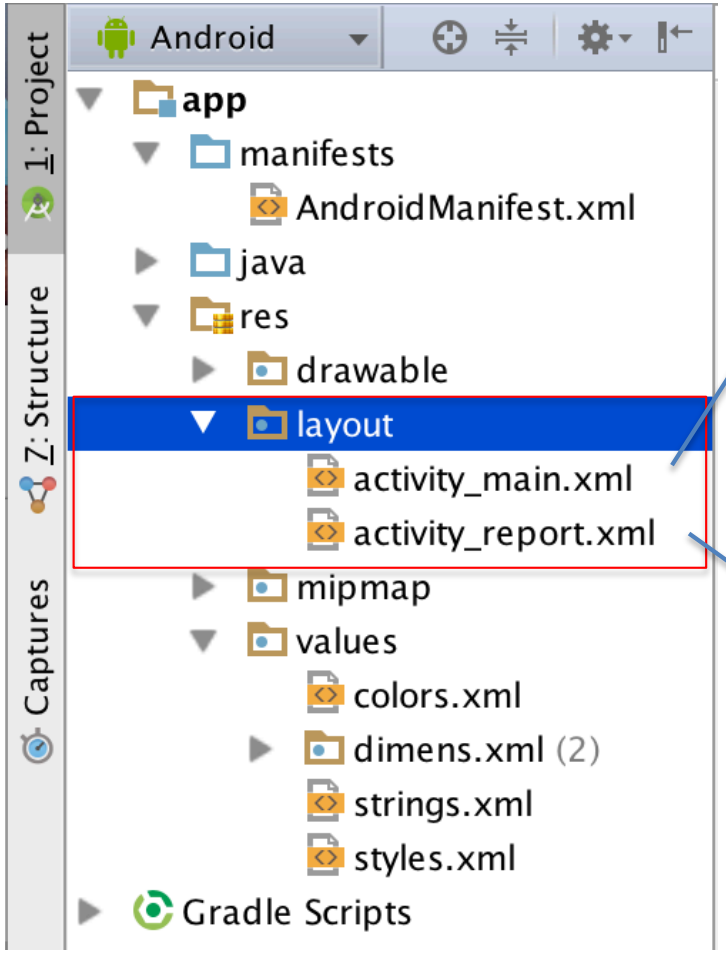


- **../app/src/main/res/drawable/**
- bot_fat.png 
- bot_fit.png 
- bot_thin.png 
- icon_128.png 

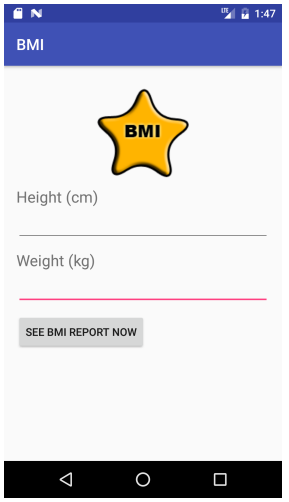
Edit the **strings.xml** file in **values** folder



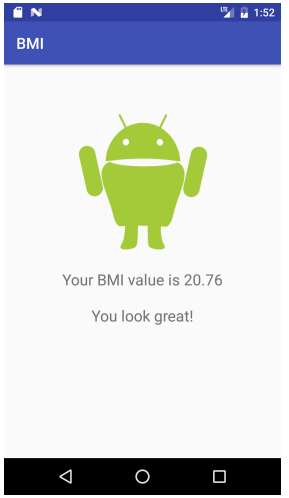
Edit and Create layout XML files in res/layout folder



- ../BMI/app/src/main/res/layout/
- activity_main.xml

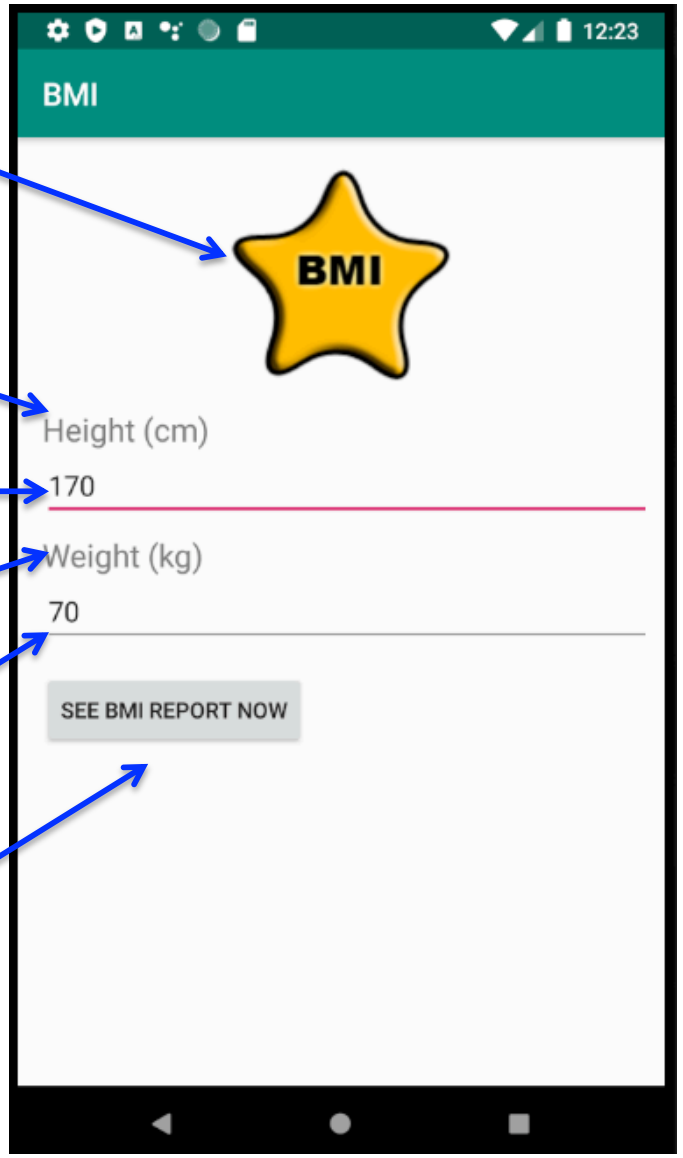


- activity_report.xml



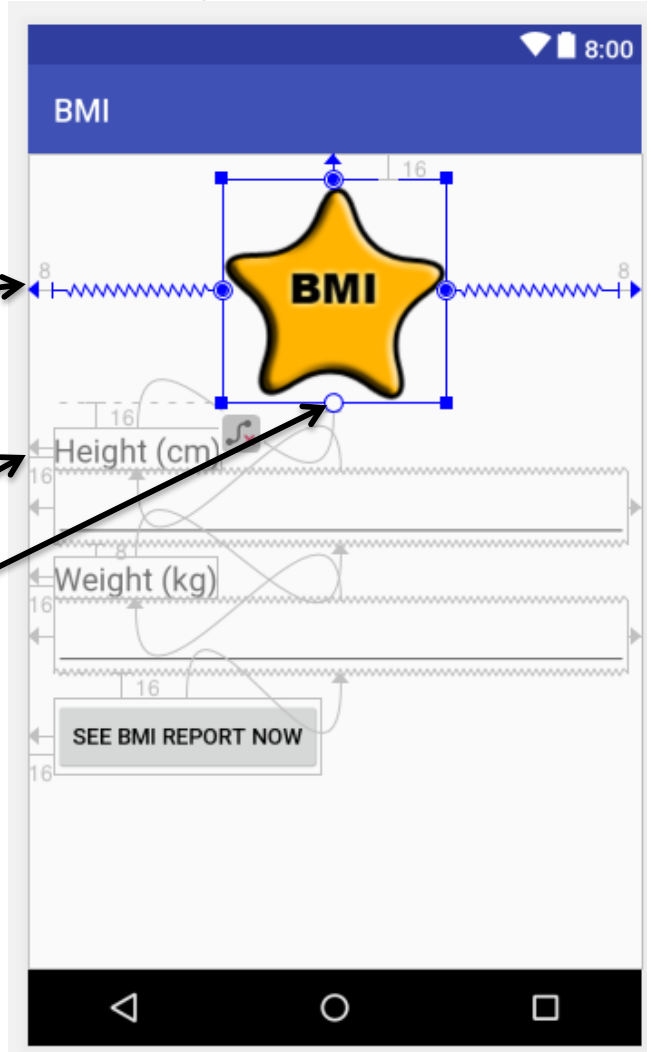
activity_main.xml (Constraint Layout)

```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto" xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="140dp"
        android:layout_height="140dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="16dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/icon_128" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="16dp"
        android:text="Height (cm)"
        android:textSize="20sp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/imageView" />

    <EditText
        android:id="@+id/heightET"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="16dp"
        android:ems="10"
        android:inputType="number"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="8dp"
        android:text="Weight (kg)"
        android:textSize="20sp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/heightET" />

    <EditText
        android:id="@+id/weightET"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="16dp"
        android:ems="10"
        android:inputType="number"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView2" />

    <Button
        android:id="@+id/reportBtn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="16dp"
        android:text="See BMI Report Now"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/weightET" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

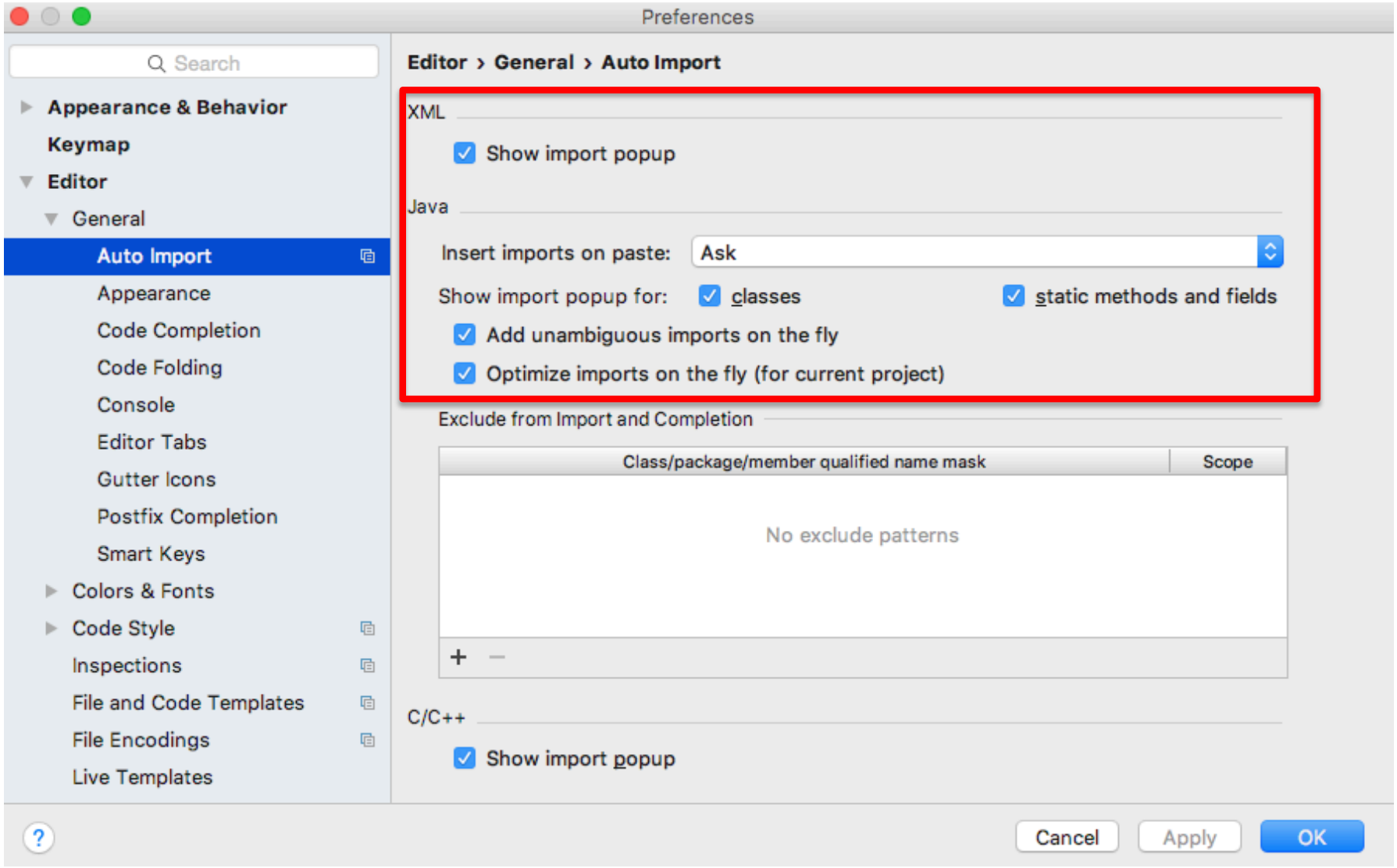


activity_main.xml (Constraint Layout)

```xml
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.bmi.MainActivity">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="140dp"
        android:layout_height="140dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="16dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/icon_128" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="16dp"
        android:text="@string/bmi_height"
        android:textSize="20sp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/imageView" />

    <EditText
        android:id="@+id/heightET"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="16dp"
        android:ems="10"
        android:inputType="number"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView" />
```

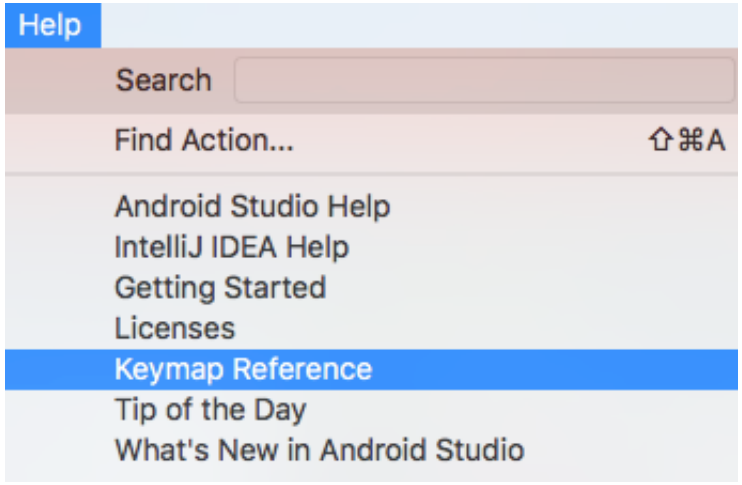


Auto Import

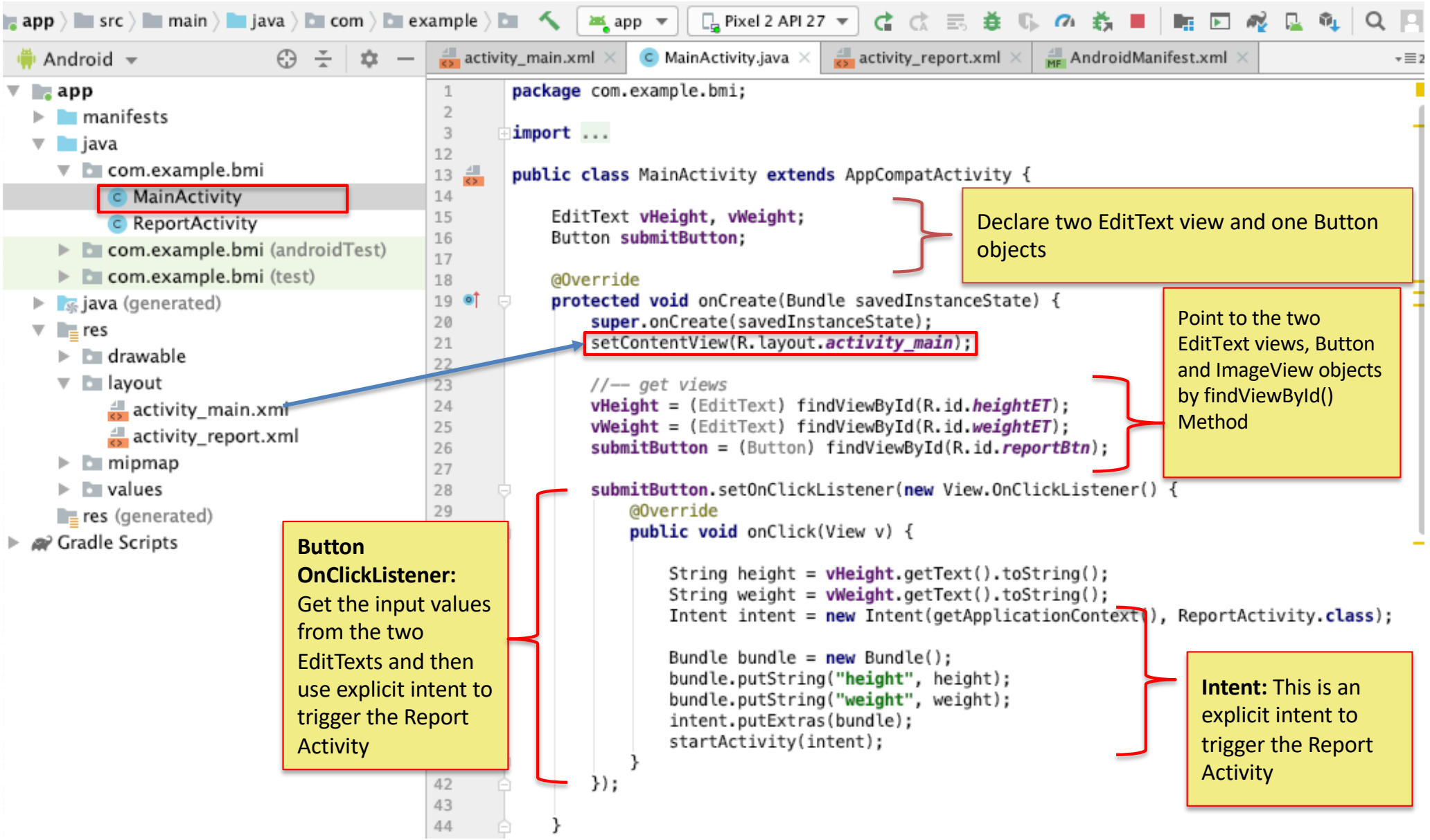

Preferences
Search
Appearance & Behavior
Keymap
Editor
General
Auto Import
Appearance
Code Completion
Code Folding
Console
Editor Tabs
Gutter Icons
Postfix Completion
Smart Keys
Colors & Fonts
Code Style
Inspections
File and Code Templates
File Encodings
Live Templates
Editor › General › Auto Import
XML
Show import popup
Java
Insert imports on paste:
Ask
Show import popup for:
classes
static methods and fields
Add unambiguous imports on the fly
Optimize imports on the fly (for current project)
Exclude from Import and Completion
Class/package/member qualified name mask
Scope
No exclude patterns
C/C++
Show import popup
Cancel
Apply
OK

Android Studio Preferences

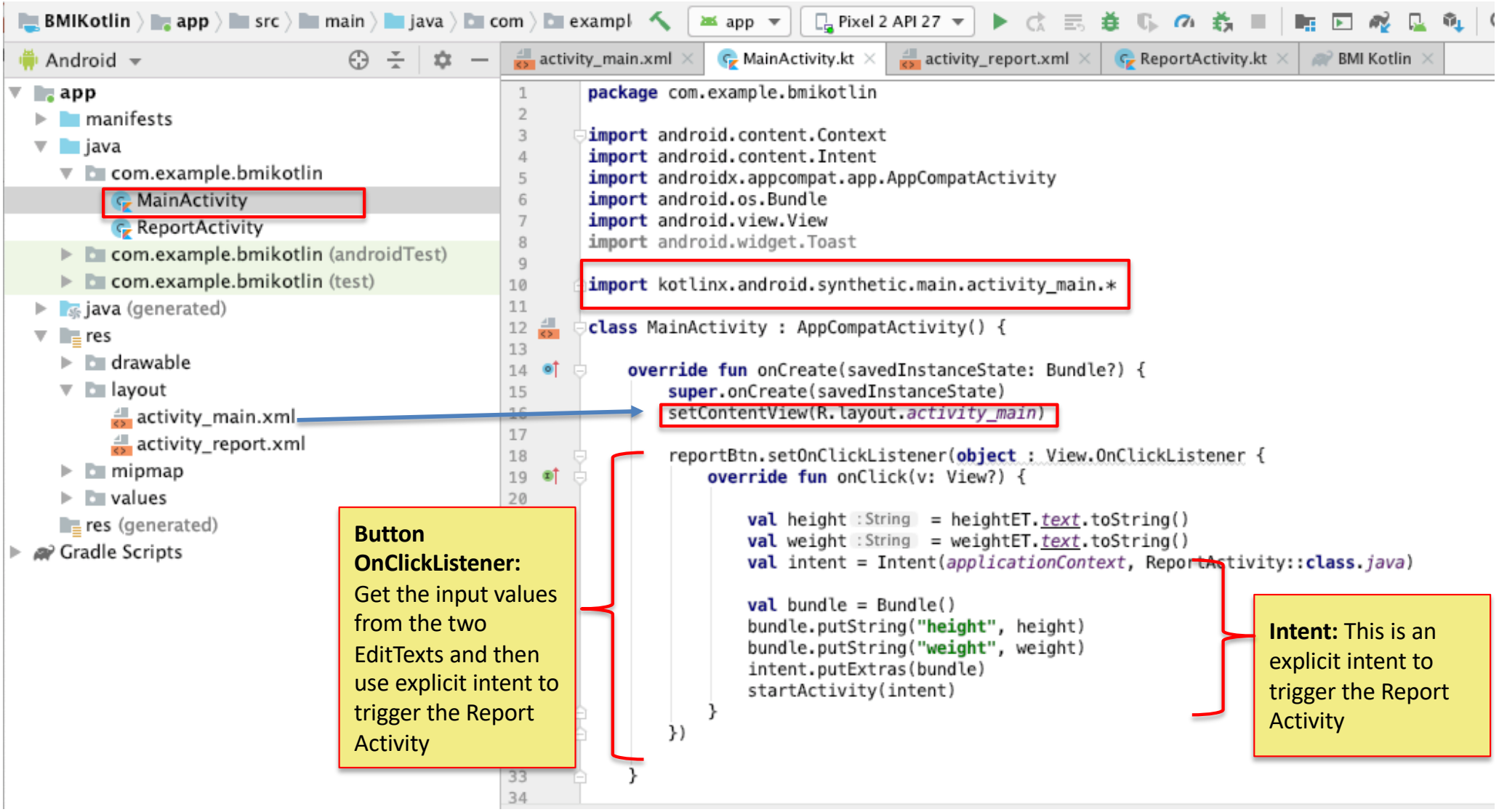
- Preference -> Editor
  - Appearance -> Show line numbers
  - Appearance -> Show method separators
  - Auto Import -> Optimize imports on the fly
  - Auto Import -> Add unambiguous imports on the fly
- Help -> Default Keymap Reference
  - Keyboard Short cut reference
- Find Action Function:
  - Command+Shift+a (Mac)
  - Ctrl+Shift+a (Win)



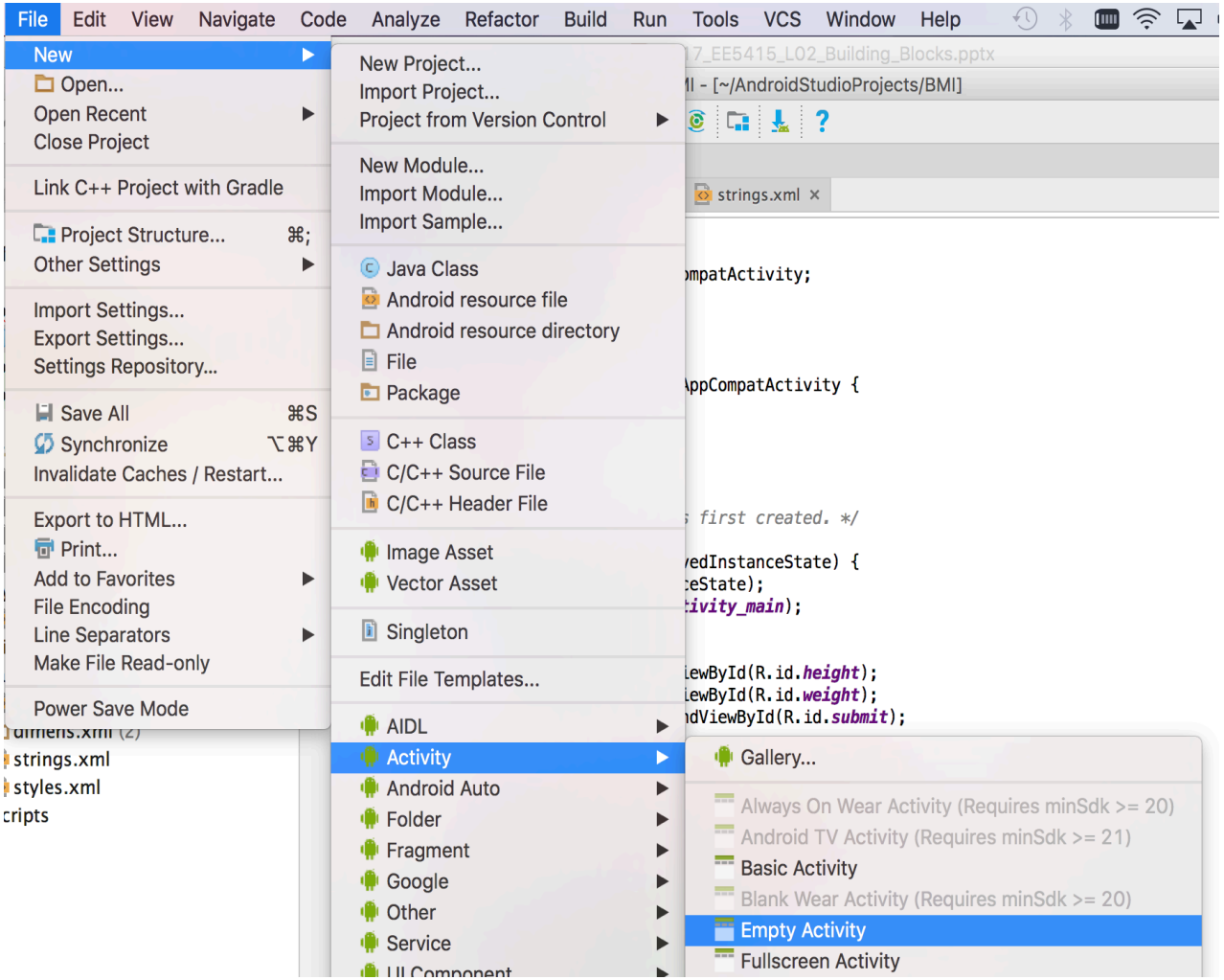
Edit the Java source file: MainActivity.java



Edit the Kotlin source file: MainActivity.kl

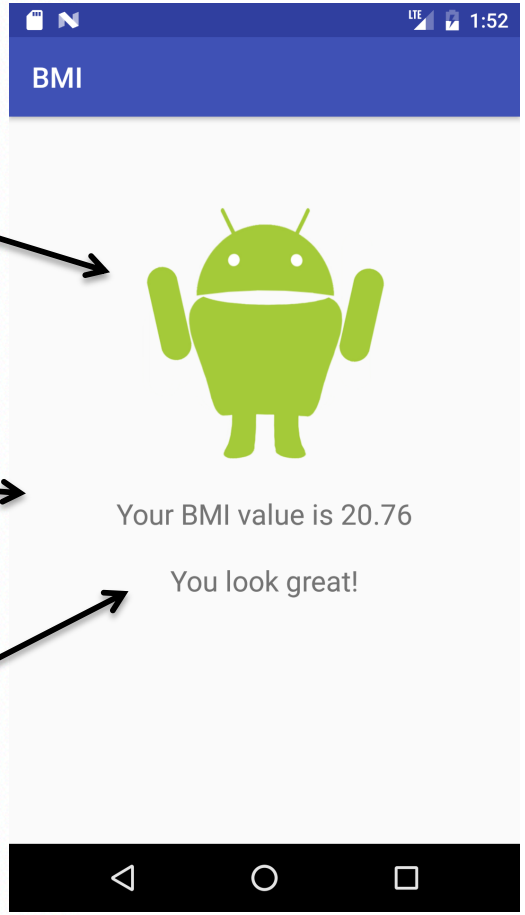


Create an Empty Activity

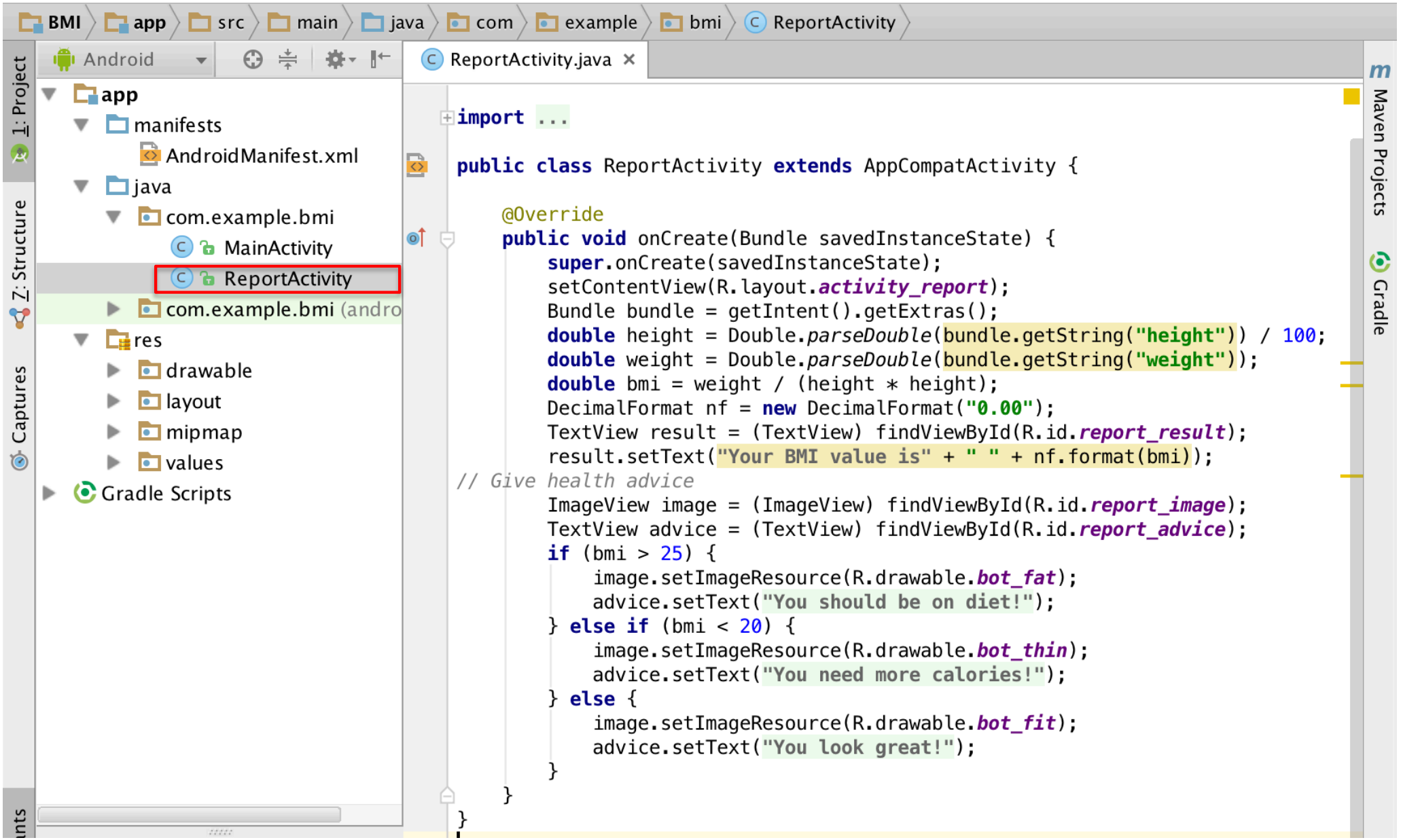


activity_report.xml (Linear Layout)

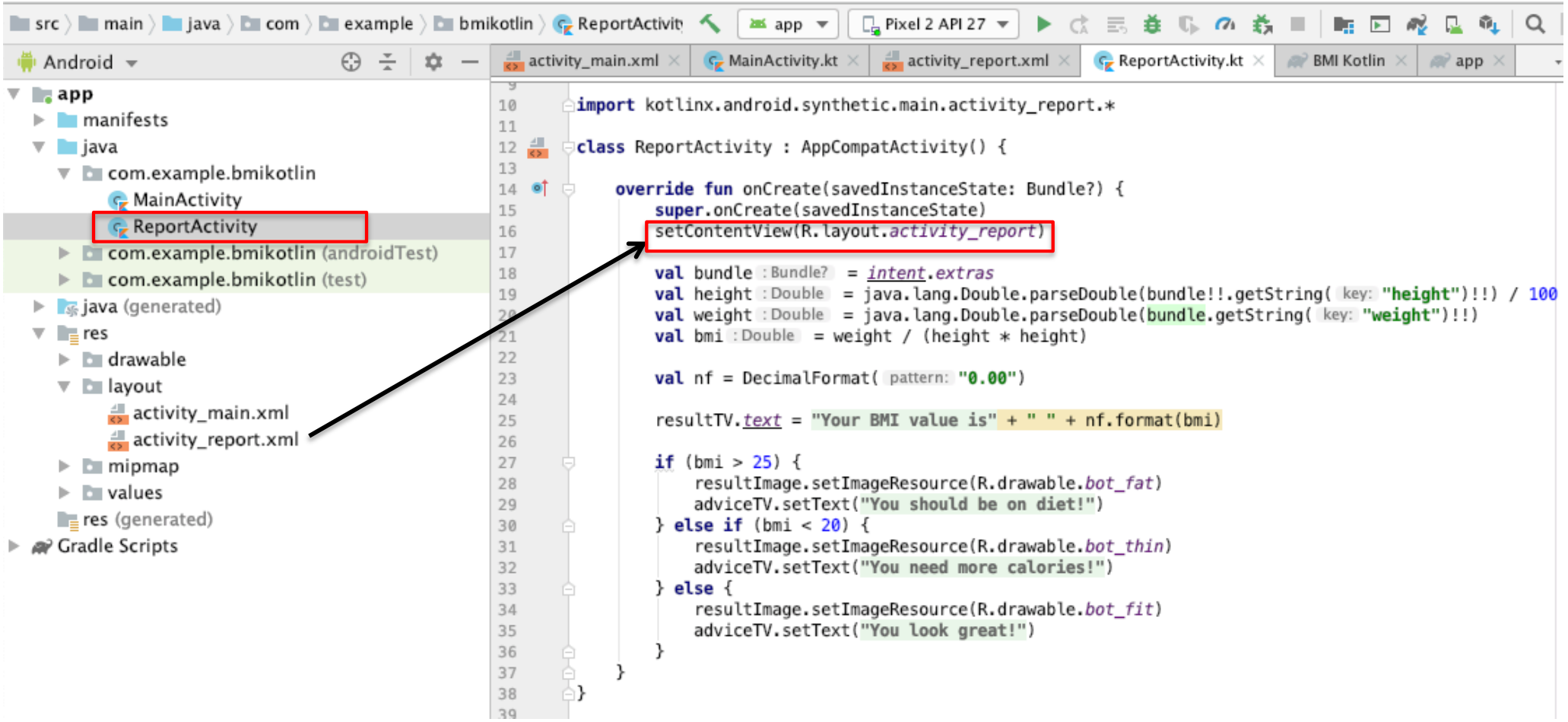
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:padding="@dimen/activity_vertical_margin">

    <ImageView android:layout_width="240dp"
        android:layout_height="240dp"
        android:layout_gravity="center_horizontal"
        android:paddingTop="40dp"
        android:paddingBottom="10dp"
        android:src="@drawable/bot_fit"
        android:id="@+id/report_image"/>

    <TextView android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:paddingTop="10dp"
        android:paddingBottom="10dp"
        android:textSize="20sp"
        android:text="@string/bmi_result"
        android:id="@+id/report_result"/>

    <TextView android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:paddingTop="10dp"
        android:paddingBottom="10dp"
        android:textSize="20sp"
        android:text="@string/advice_average"
        android:id="@+id/report_advice"/>
</LinearLayout>
```



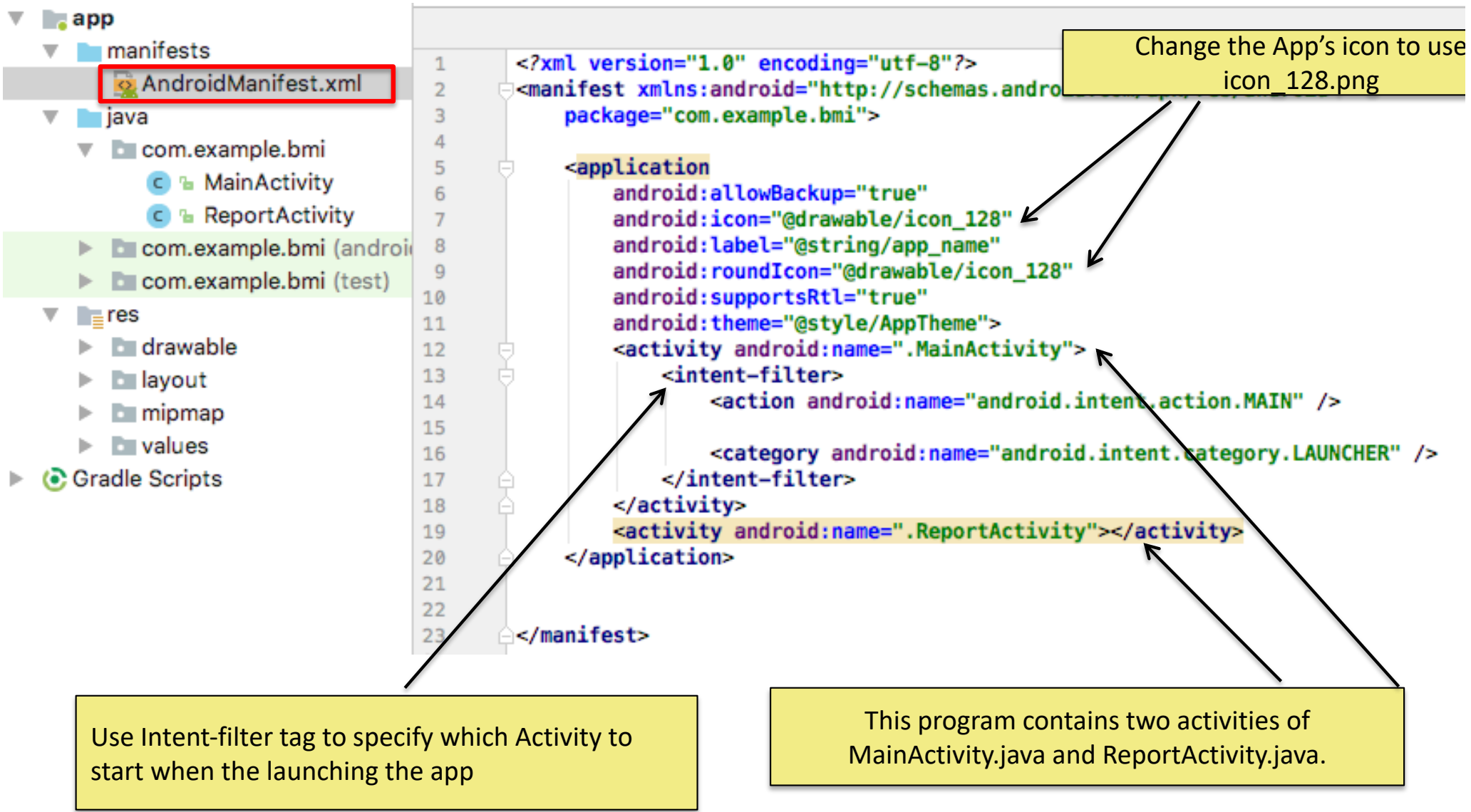
Create ActivityReport.java Class file



Create ActivityReport.kl Class file

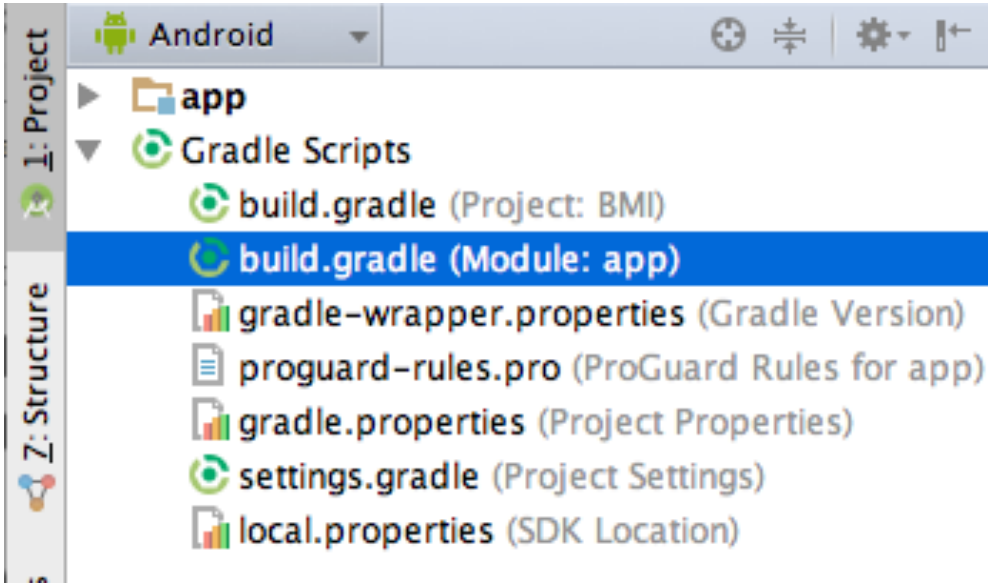


Edit AndroidManifest.xml

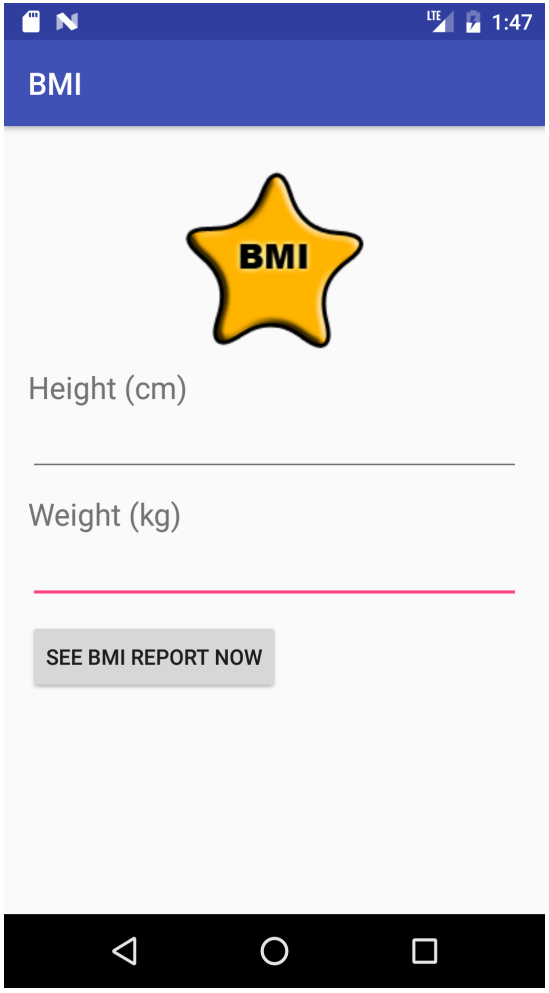


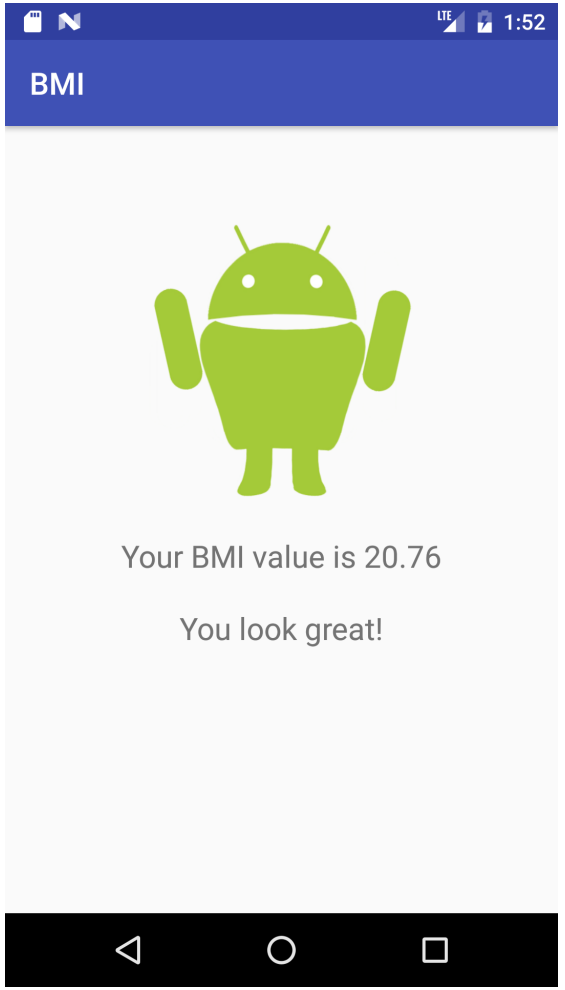
Managing Gradle Build Scripts

- local.proerties
  - Specifies the Android SDK location
- build.gradle (Module.app)
  - Configure a lot settings of your project
    - complileSdkVersion
    - BuildToolsversion
    - minSdkVersion
    - targetSdkVersion
    - versioncode
    - versionName
    - dependenices



Run your BMI App

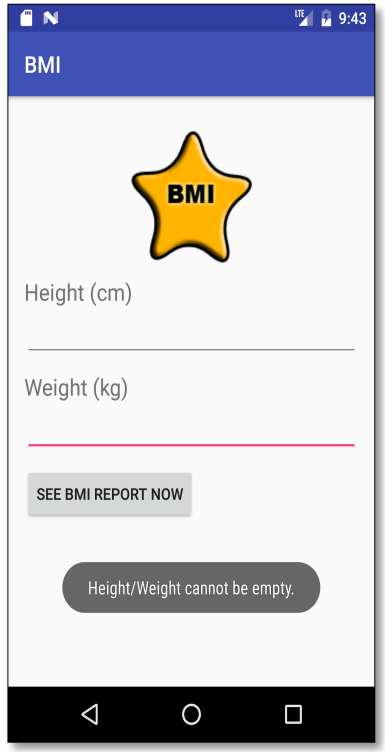




Data Validation (Java)

- You now have a simple BMI calculator that can measure your body weight. However a user may be able to submit empty values for calculation and it will cause an exception (force close) in your program.

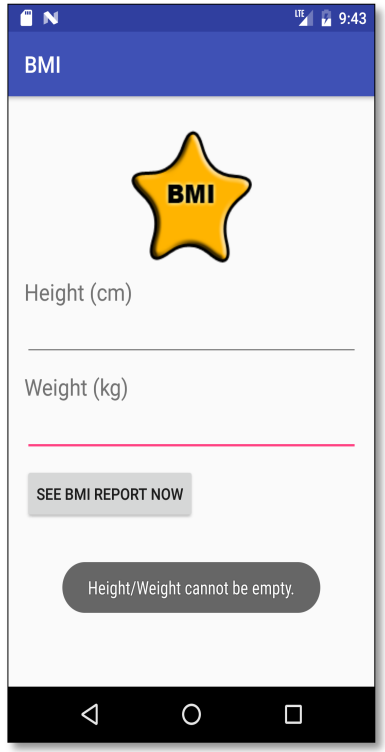
```
submitButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        String height = vHeight.getText().toString();
        String weight = vWeight.getText().toString();

        if (height.equals("") || weight.equals("")) {
            Toast.makeText(MainActivity.this, R.string.bmi_warning, Toast.LENGTH_LONG).show();
        } else {
            Intent intent = new Intent(getApplicationContext(), ReportActivity.class);

            Bundle bundle = new Bundle();
            bundle.putString("height", height);
            bundle.putString("weight", weight);
            intent.putExtras(bundle);
            startActivity(intent);
        }
    }
});
```



Data Validation (Kotlin)

- You now have a simple BMI calculator that can measure your body weight. However a user may be able to submit empty values for calculation and it will cause an exception (force close) in your program.

```
reportBtn.setOnClickListener(object : View.OnClickListener {
    override fun onClick(v: View?) {

        val height :String  = heightET.text.toString()
        val weight :String  = weightET.text.toString()

        if (height == "" || weight == "" ||height == "0" || weight=="0") {
            Toast.makeText( context: this@MainActivity, R.string.bmi_warning, Toast.LENGTH_LONG).show()
        }else {
            val intent = Intent(applicationContext, ReportActivity::class.java)
            val bundle = Bundle()
            bundle.putString("height", height)
            bundle.putString("weight", weight)
            intent.putExtras(bundle)
            startActivity(intent)
            savePreferences(height, weight)
        }
    }
})
```



User Preference

- Being a user, you may not want to enter your height and weight every time, especially your height and weight won't change very dramatically.
- You can actually save user's data in a persistent storage so that you can get it back when the application is started next time.

1. Open **MainActivity.java** and insert the following two methods to the class: **(Java)**

```
public void savePreferences(String h, String w) {
    SharedPreferences pref = getSharedPreferences("BMI", MODE_PRIVATE);
    pref.edit().putString("height", h).apply();
    pref.edit().putString("weight", w).apply();
}

public void loadPreferences() {
    SharedPreferences pref = getSharedPreferences("BMI", MODE_PRIVATE);
    vHeight.setText(pref.getString("height", "0"));
    vWeight.setText(pref.getString("weight", "0"));
}
```

2. Insert this code at the end of the **submitButton.setOnClickListener()** method such that it will save the inputs when the button is clicked:

```
savePreferences(height, weight);
```

3. Add the `onStart()` method with in the `MainActivity` class as shown below that calling the `loadPreferences()` method with the app start again every time.

```
@Override
protected void onStart() {
    super.onStart();
    loadPreferences(); }
```

1. Open **MainActivity.kl** and insert the following two methods to the class: **(Kotlin)**

```
fun savePreferences(h: String, w: String) {
    val pref = getSharedPreferences("BMI", Context.MODE_PRIVATE)
    pref.edit().putString("height", h).apply()
    pref.edit().putString("weight", w).apply()
}

fun loadPreferences() {
    val pref = getSharedPreferences("BMI", Context.MODE_PRIVATE)
    heightET.setText(pref.getString("height", "0"))
    weightET.setText(pref.getString("weight", "0"))
}
```

2. Insert this code at the end of the **reportBtn.setOnClickListener()** method such that it will save the inputs when the button is clicked:

```
savePreferences(height, weight)
```

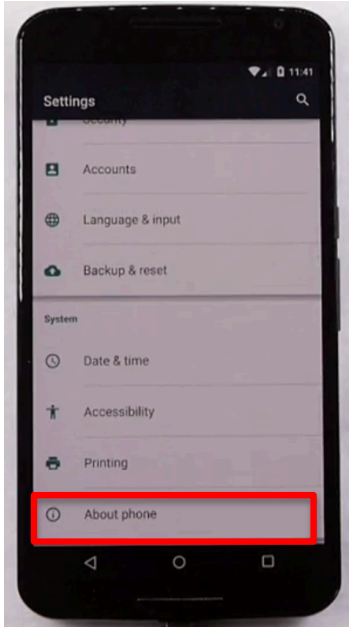
3. Add the `onStart()` method with in the `MainActivity` cleass as shown below that calling the `loadPreferences()` method with the app start again every time.

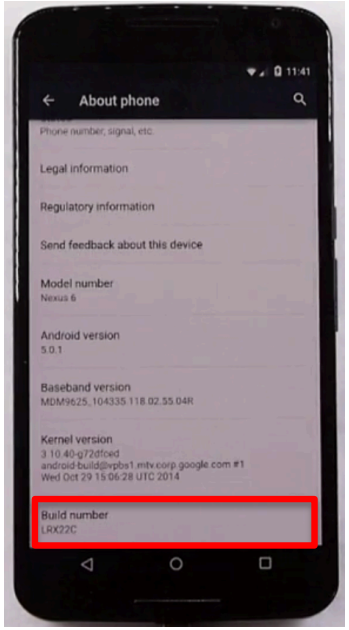
```
override fun onStart() {
    super.onStart()
    loadPreferences()
}
```

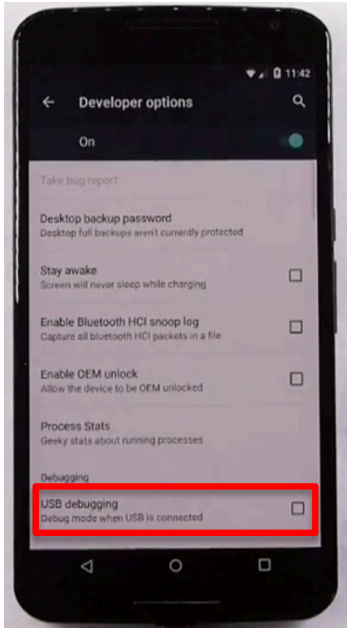
Test your app on real Android device

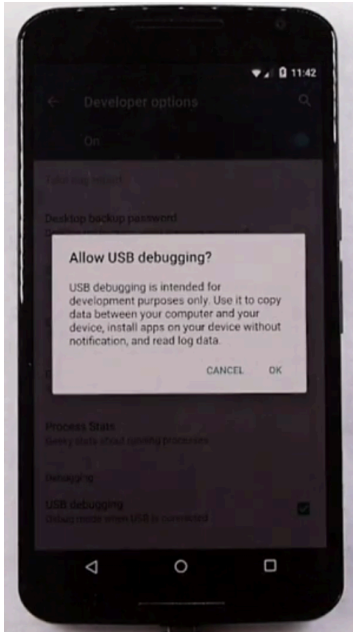
- We need to enable the USB Debugging and then connect your Android device to the PC.
- Settings >> Developer Options >> USB Debugging

About Phone >> Build Number >> Tap on the Screen

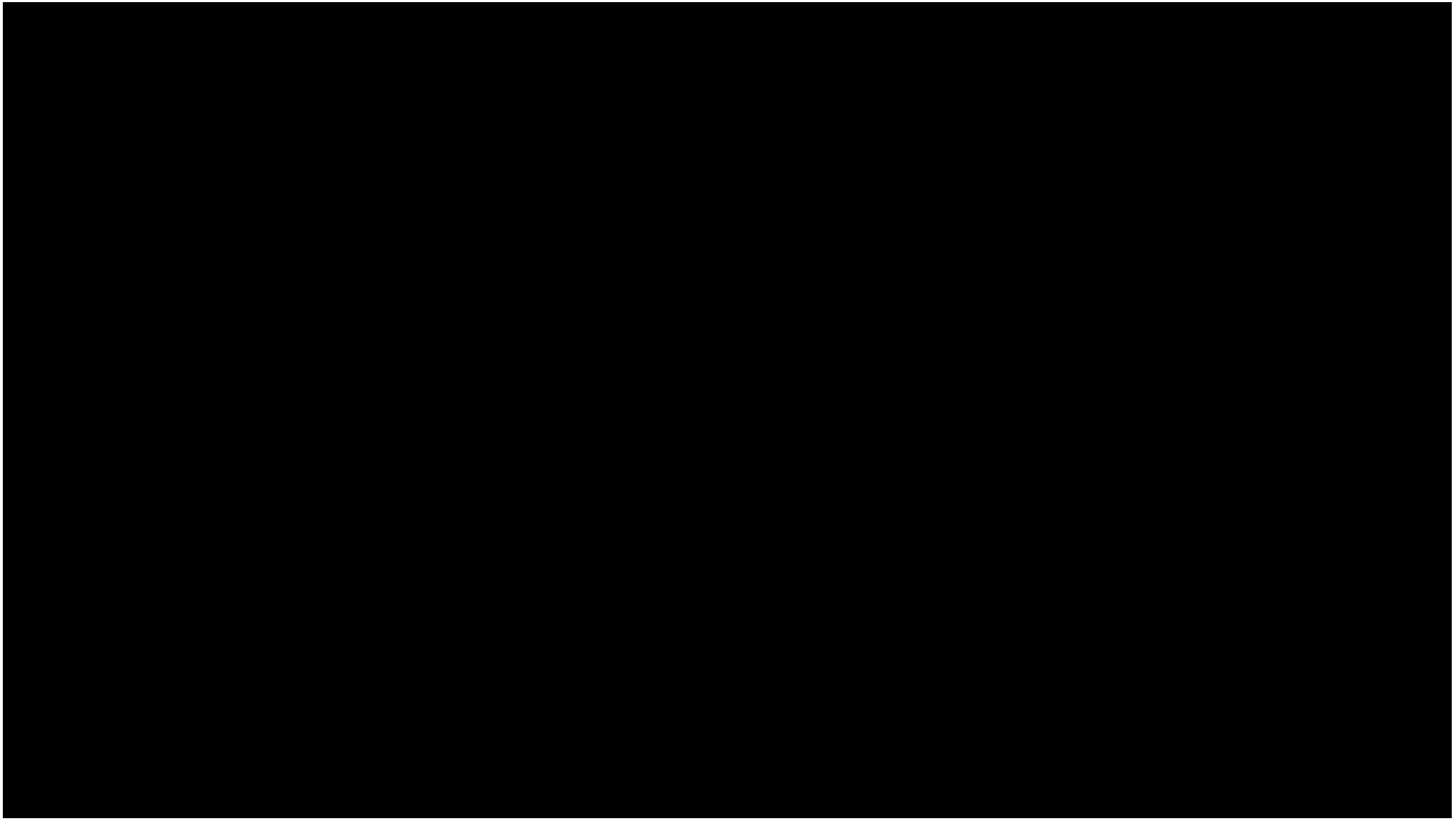








Enable Developer Mode on Android 5 Devices



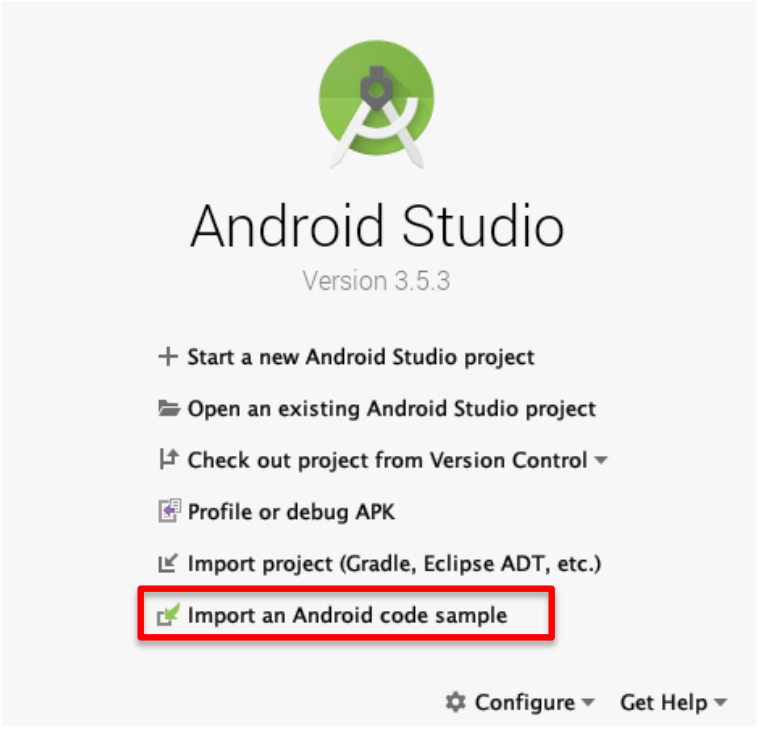
Enable Developer Mode on Samsung Galaxy S8

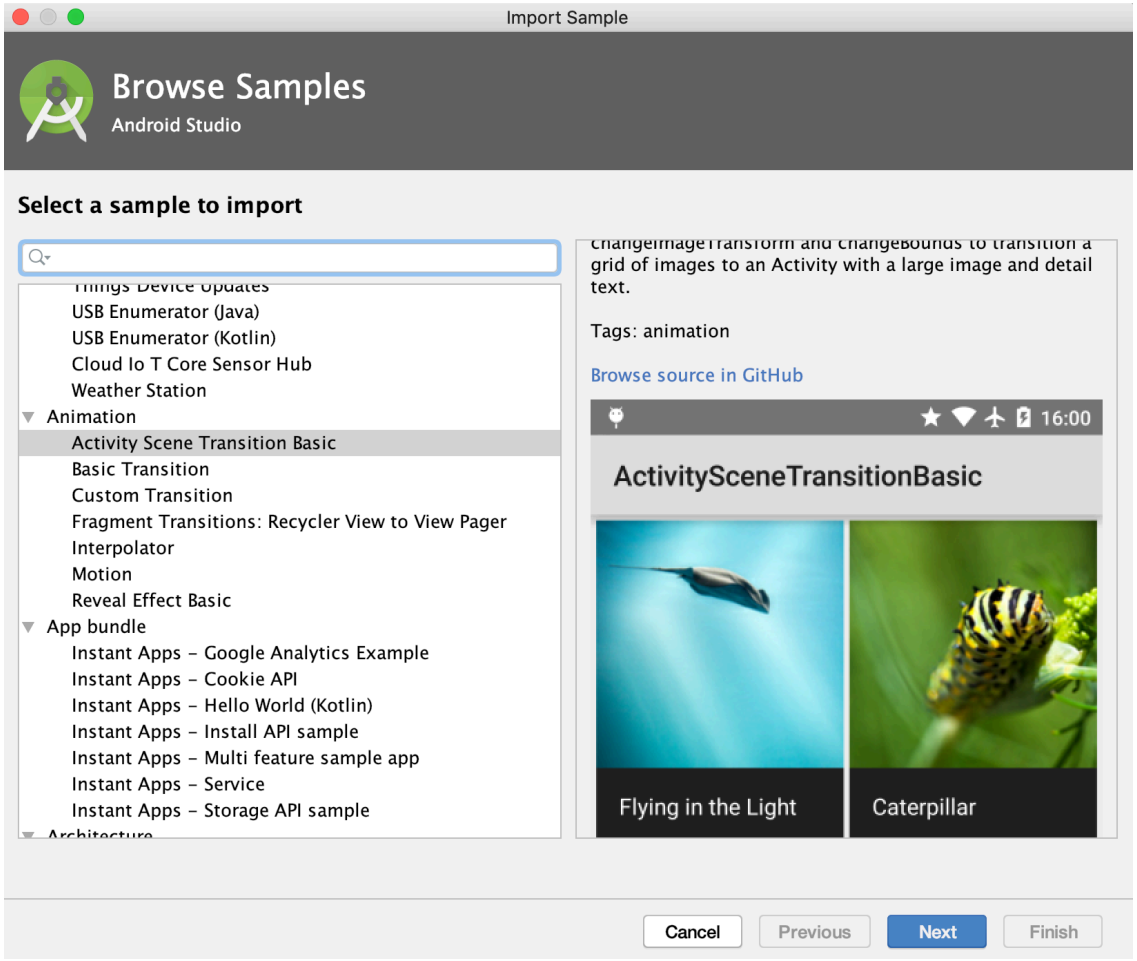
https://www.youtube.com/watch?v=AwYqeVqnyZw

Recommended Android Devices

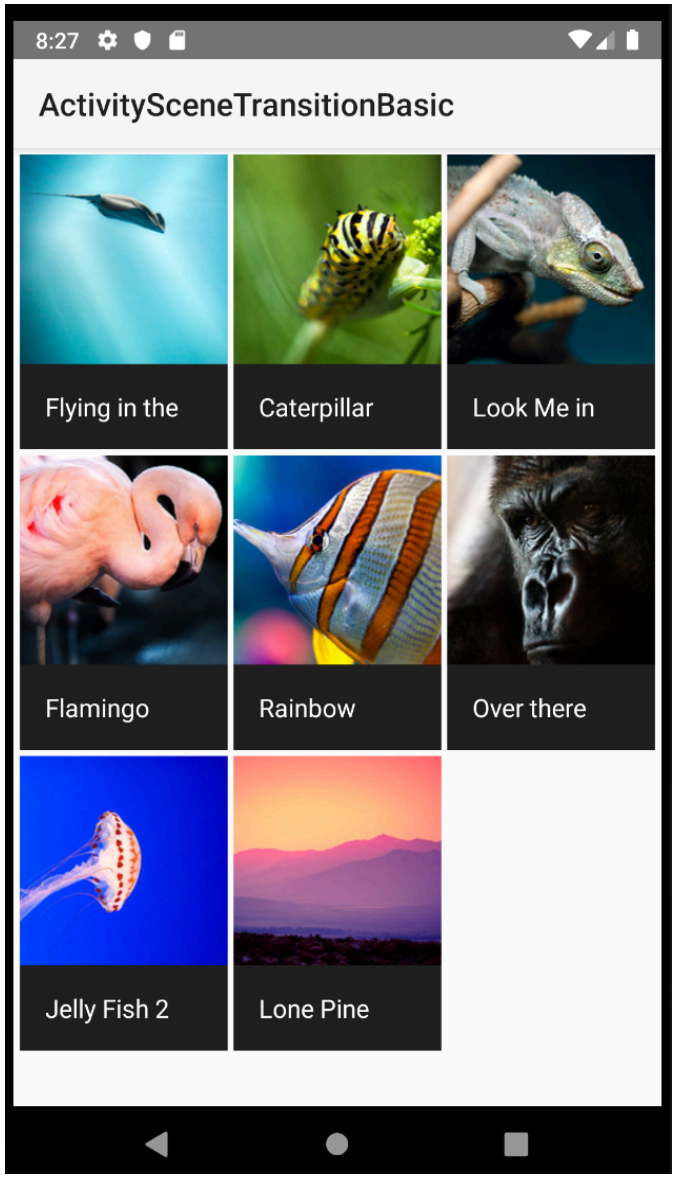
- It is recommended to have at least one Android device for testing
  - Nexus devices offer “pure” Android experience
  - Samsung devices have largest share of device market
  - Phones and tablets offer different screen sizes/densities

Import Sample Projects to AS





Run Activity Scene Translation Sample Project



Githut.com/googlesamples