01 Foundation

Create New Project

Create Android Project

Application name

Placemark

Company domain

wit.org

Project location

/Users/edeleastar/repos/modules/mobile/mad-2-2018/prj/placemark-projects/placemark-origin ...

Package name

org.wit.placemark Edit

- [ ] **Include C++ support**
- [x] **Include Kotlin support**

Cancel

Create New Project

Target Android Devices

Select the form factors and minimum SDK

Some devices require additional SDKs. Low API levels target more devices, but offer fewer API features.

- [x] **Phone and Tablet**

  API 23: Android 6.0 (Marshmallow)

  By targeting **API 23 and later**, your app will run on approximately **39.3%** of devices. Help me choose

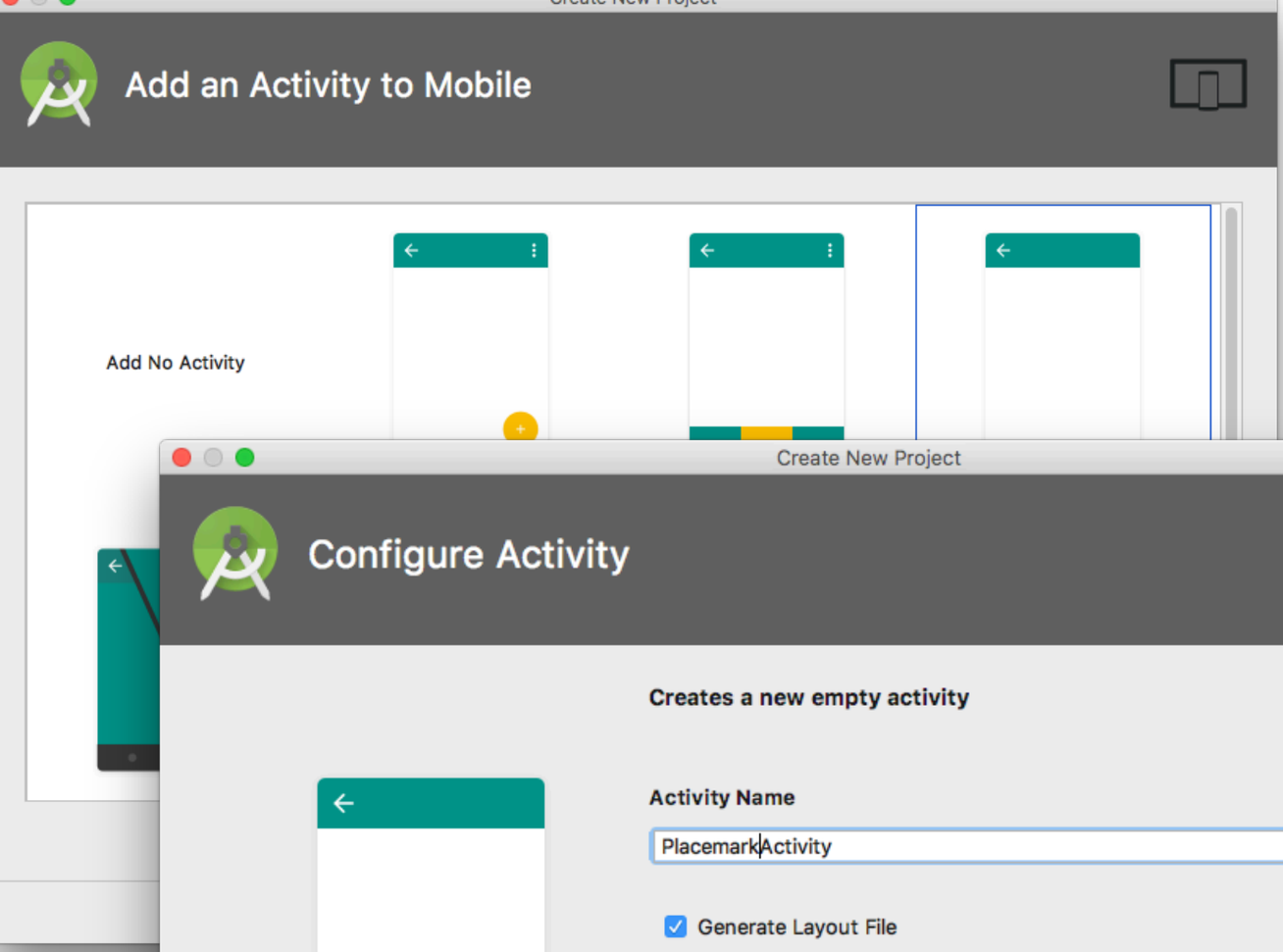
  - [ ] Include Android Instant App support
- [ ] **Wear**

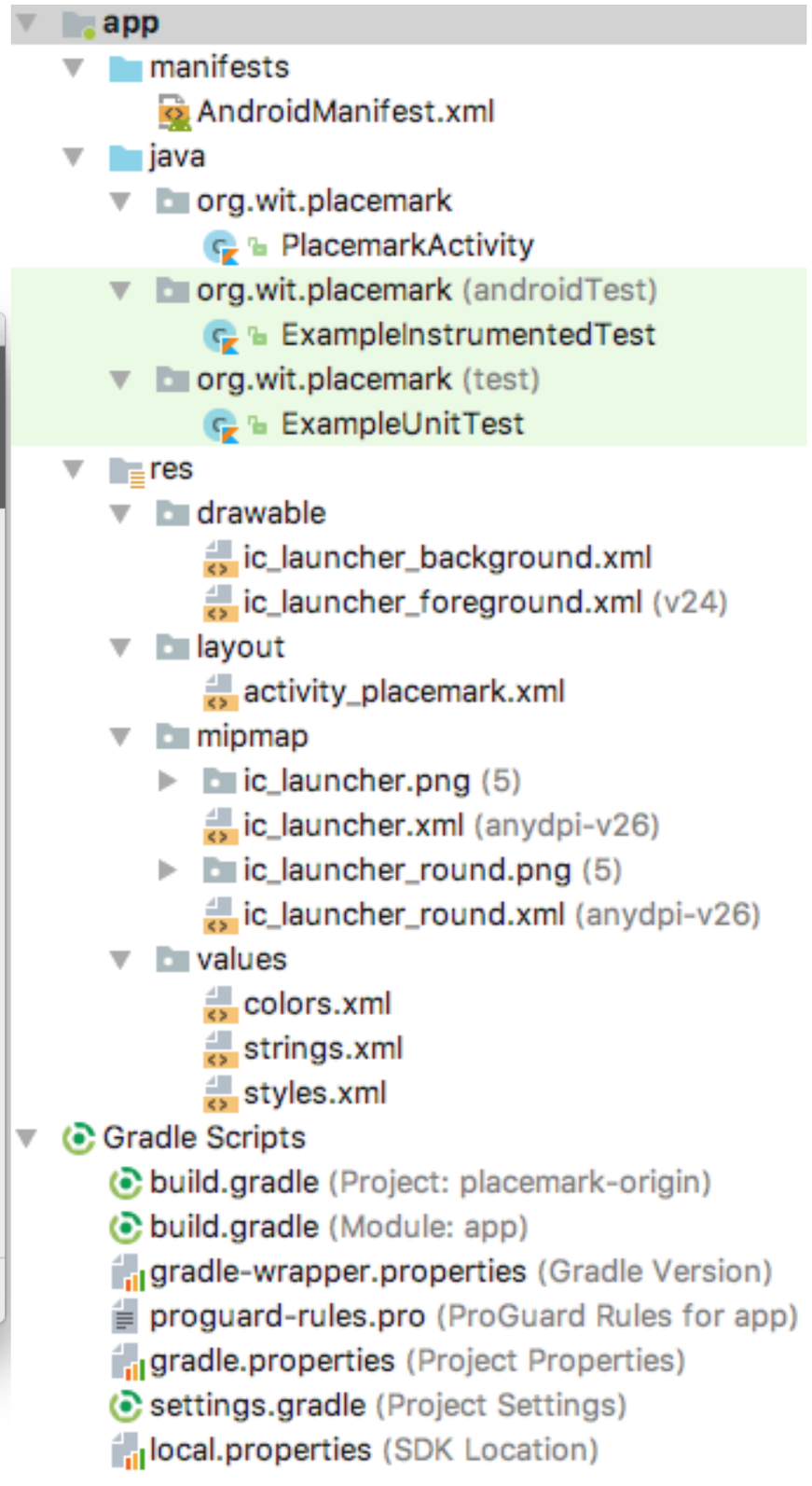
  API 21: Android 5.0 (Lollipop)
- [ ] **TV**

  API 21: Android 5.0 (Lollipop)
- [ ] **Android Auto**
- [ ] **Android Things**

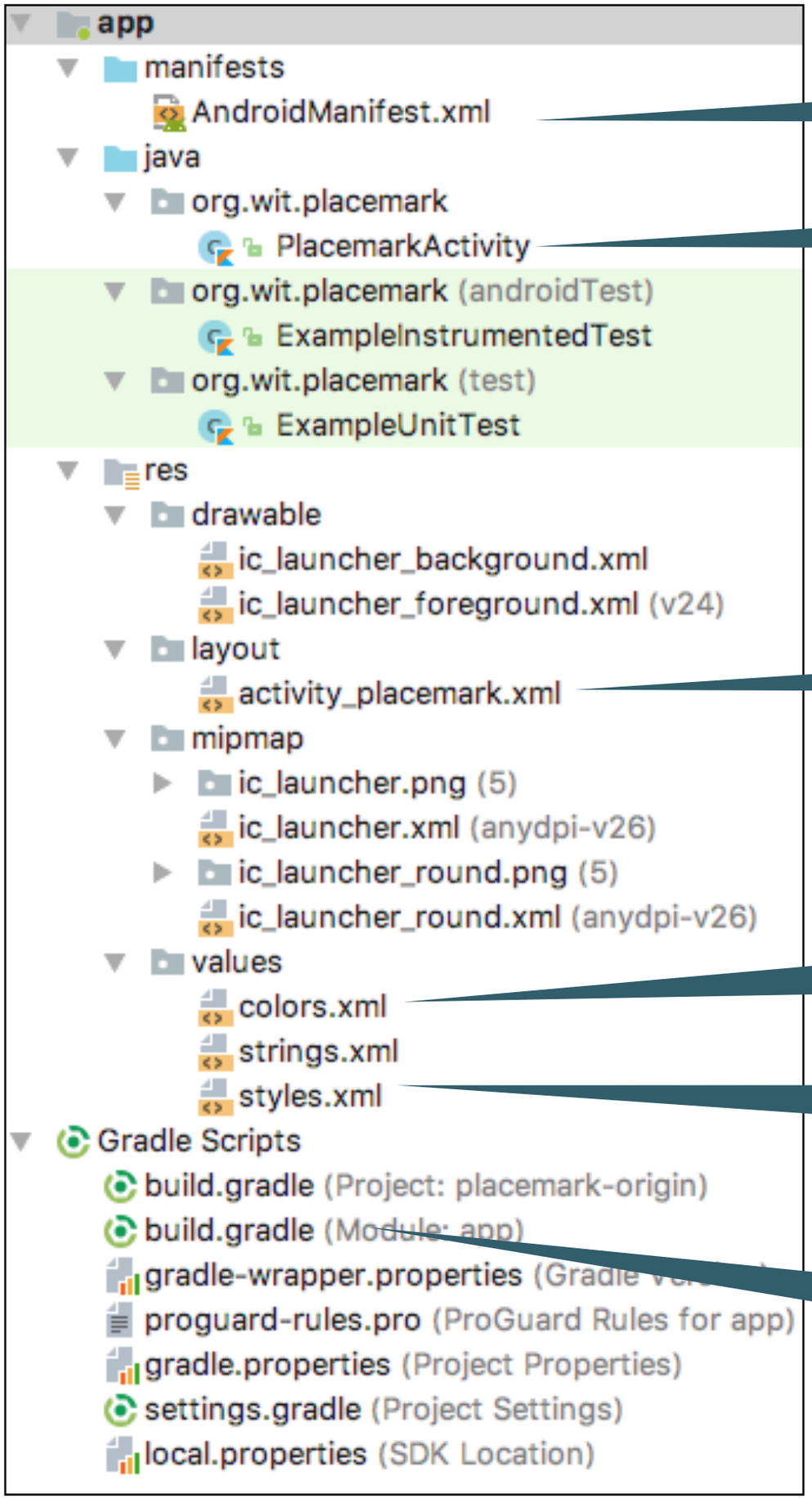
  API 24: Android 7.0 (Nougat)

Cancel Previous Next Finish






```
*.iml
.gradle
/local.properties
/.idea
.DS_Store
/build
/captures
.externalNativeBuild
```



AndroidManifest.xml

PlacemarkActivity.kt

activity_placemark.xml

colors.xml

styles.xml

build.gradle

PlacemarkActivity.kt

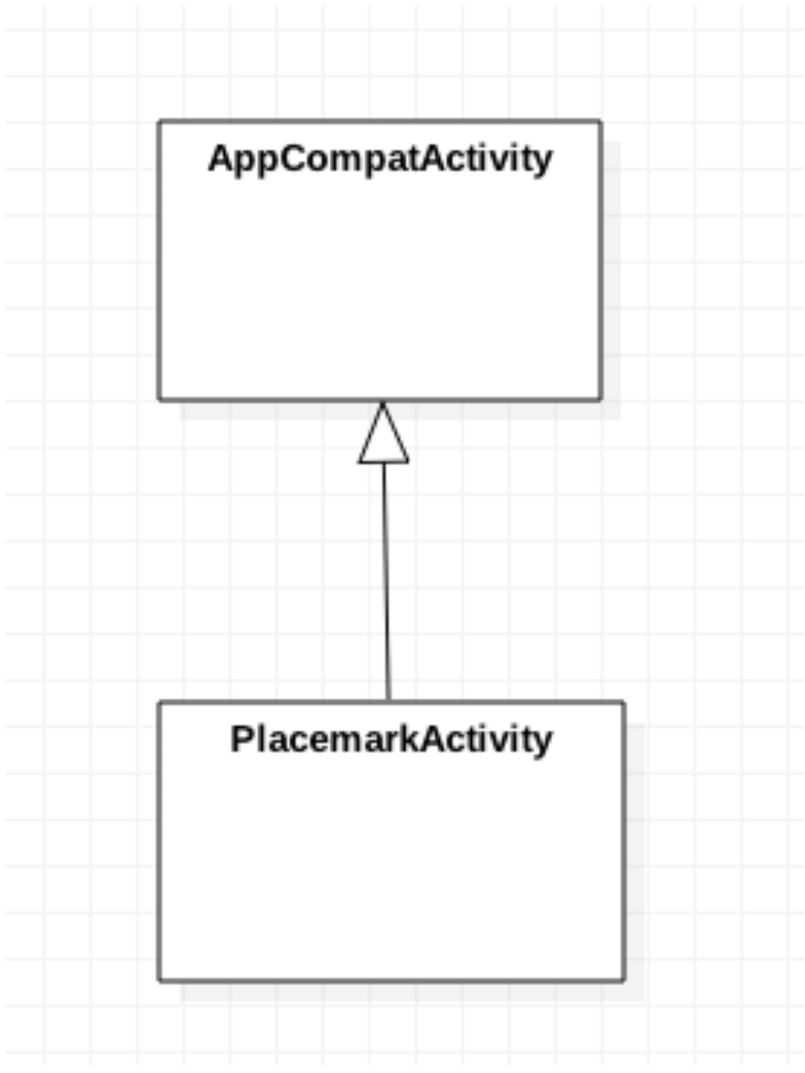
```
package org.wit.placemark

import android.support.v7.app.AppCompatActivity
import android.os.Bundle

class PlacemarkActivity : AppCompatActivity() {

  override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_placemark)
  }
}
```

Class Diagram

```
package org.wit.placemark

import android.support.v7.app.AppCompatActivity
import android.os.Bundle

class PlacemarkActivity : AppCompatActivity() {

  override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_placemark)
  }
}
```

Class Diagram: showing base class

```
package org.wit.placemark

import android.support.v7.app.AppCompatActivity
import android.os.Bundle

class PlacemarkActivity : AppCompatActivity() {

  override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_placemark)
  }
}
```



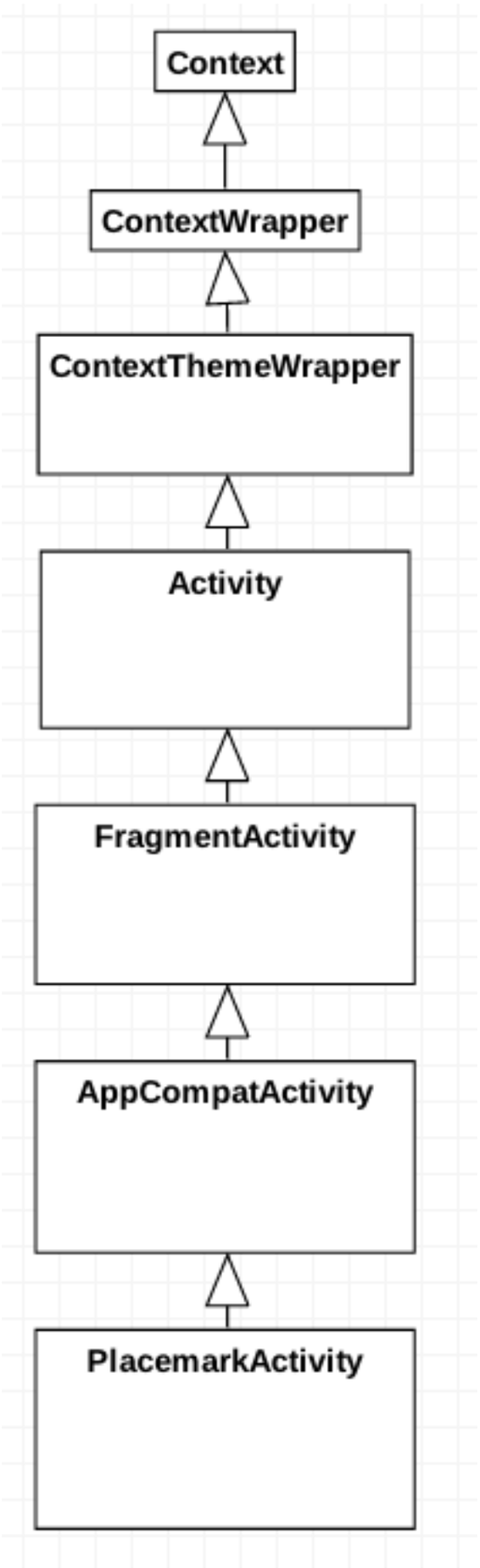
Class Diagram: showing all base class

```
package org.wit.placemark

import android.support.v7.app.AppCompatActivity
import android.os.Bundle

class PlacemarkActivity : AppCompatActivity() {

  override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_placemark)
  }
}
```

AppCompatActivity

```
public class AppCompatActivity
extends FragmentActivity implements AppCompatCallback,
TaskStackBuilder.SupportParentable, ActionBarDrawerToggle.DelegateProvider
```

java.lang.Object
- ↳ android.content.Context
  - ↳ android.content.ContextWrapper
    - ↳ android.view.ContextThemeWrapper
      - ↳ android.app.Activity
        - ↳ android.support.v4.app.FragmentActivity
          - ↳ android.support.v7.app.AppCompatActivity



AndroidManifest.xml

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

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

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

</manifest>
```

build.gradle

```
apply plugin: 'com.android.application'

apply plugin: 'kotlin-android'

apply plugin: 'kotlin-android-extensions'

android {
  compileSdkVersion 26
  defaultConfig {
    applicationId "org.wit.placemark"
    minSdkVersion 23
    targetSdkVersion 26
    versionCode 1
    versionName "1.0"
    testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
  }
  buildTypes {
    release {
      minifyEnabled false
      proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
    }
  }
}

dependencies {
  implementation fileTree(dir: 'libs', include: ['*.jar'])
  implementation "org.jetbrains.kotlin:kotlin-stdlib-jre7:$kotlin_version"
  implementation 'com.android.support:appcompat-v7:26.1.0'
  implementation 'com.android.support.constraint:constraint-layout:1.0.2'
  testImplementation 'junit:junit:4.12'
  androidTestImplementation 'com.android.support.test:runner:1.0.1'
  androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.1'
}
```

activity_placemark.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="org.wit.placemark.PlacemarkActivity">

  <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Hello World!"
      app:layout_constraintBottom_toBottomOf="parent"
      app:layout_constraintLeft_toLeftOf="parent"
      app:layout_constraintRight_toRightOf="parent"
      app:layout_constraintTop_toTopOf="parent"/>

</android.support.constraint.ConstraintLayout>
```

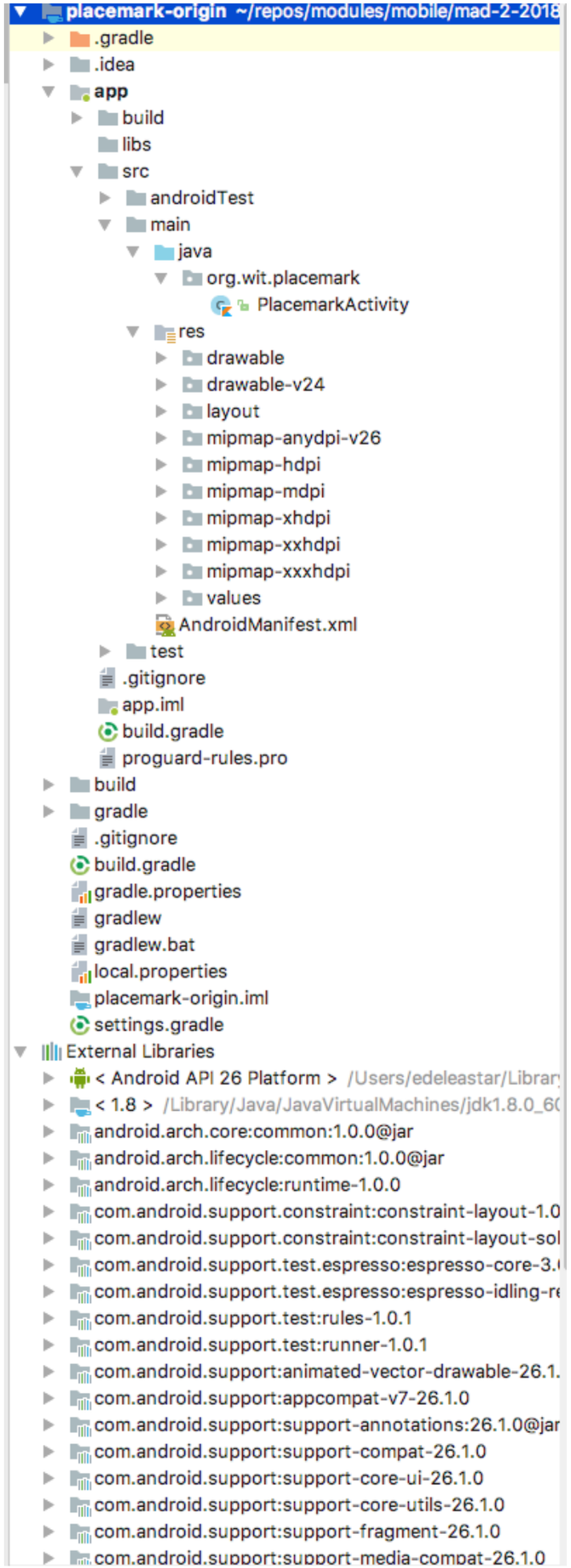
colors.xml

```xml
<resources>
  <string name="app_name">Placemark</string>
</resources>
```

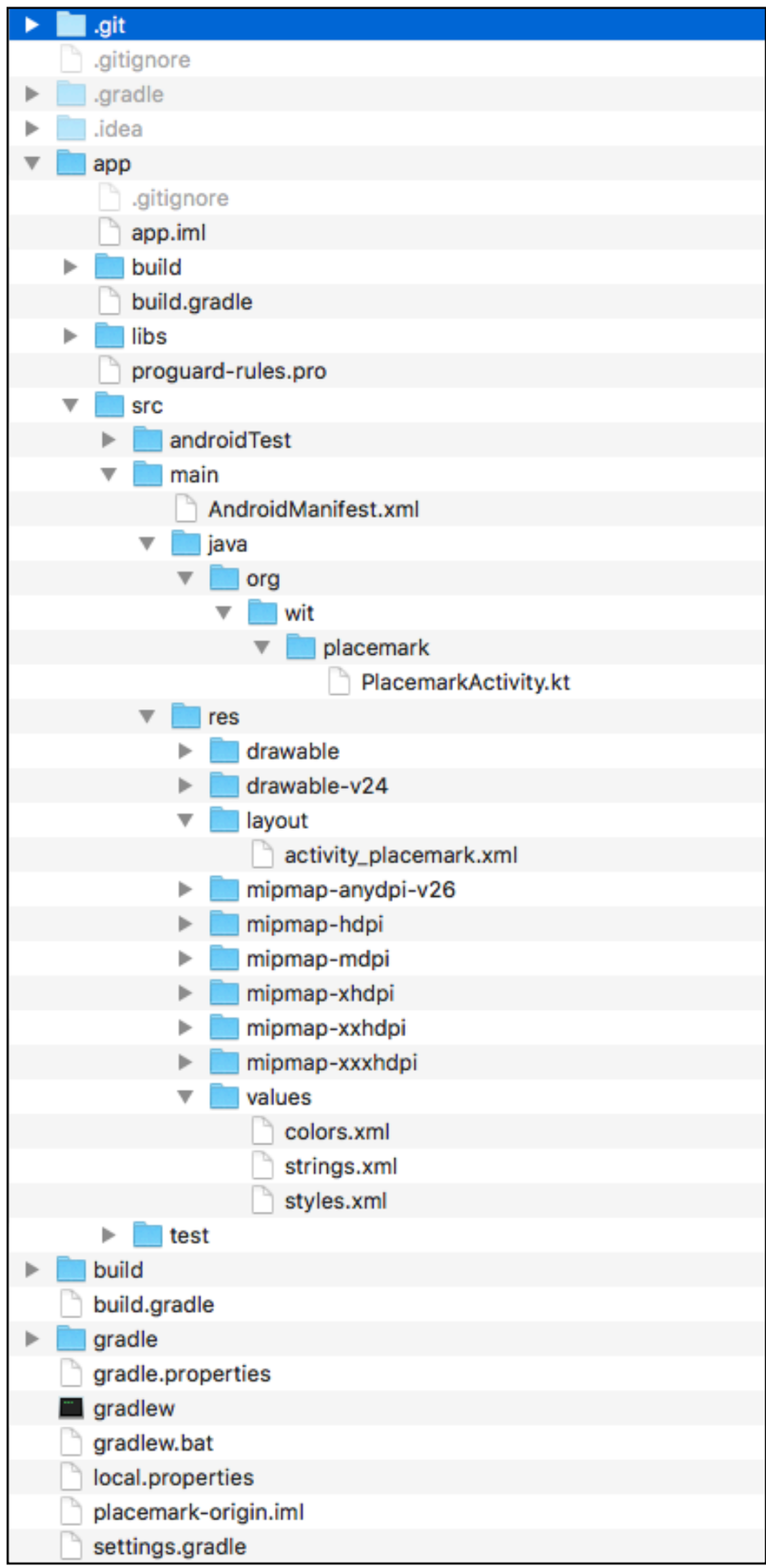
styles.xml

```xml
<resources>

  <!-- Base application theme. -->
  <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
    <!-- Customize your theme here. -->
    <item name="colorPrimary">@color/colorPrimary</item>
    <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
    <item name="colorAccent">@color/colorAccent</item>
  </style>

</resources>
```

Logical view



Physical View



Include Design Library

Develop › Training › Best Practices for User Interface › Creating Apps with Material Design

Using the Design Support Library

Previous lessons in this class have covered a variety of material design components that are available as part of the Android framework. The Design Support library provides APIs to support additional important material design components and patterns to your applications beyond those covered by the Android framework, to all devices running Android 2.1 or later. This lesson introduces two of the components from the Design Support library using examples that you can incorporate into your applications.

‹ Previous Next ›

This lesson teaches you to

- Add the Dependency
- Create a Floating Action Button
- Create a Navigation Drawer

You should also read

- Material design specification
- Material design on Android

Add the Dependency

The examples in this lesson rely on the Design Support Library, which you can make use of in your projects by adding the following Gradle dependency to your application's module:

```
compile 'com.android.support:design:27.0.2'
```

build.gradle

```
...
  implementation 'com.android.support:design:26.1.0'
...
```

styles.xml

```
  <style name="AppTheme" parent="Theme.AppCompat.Light.NoActionBar">
```

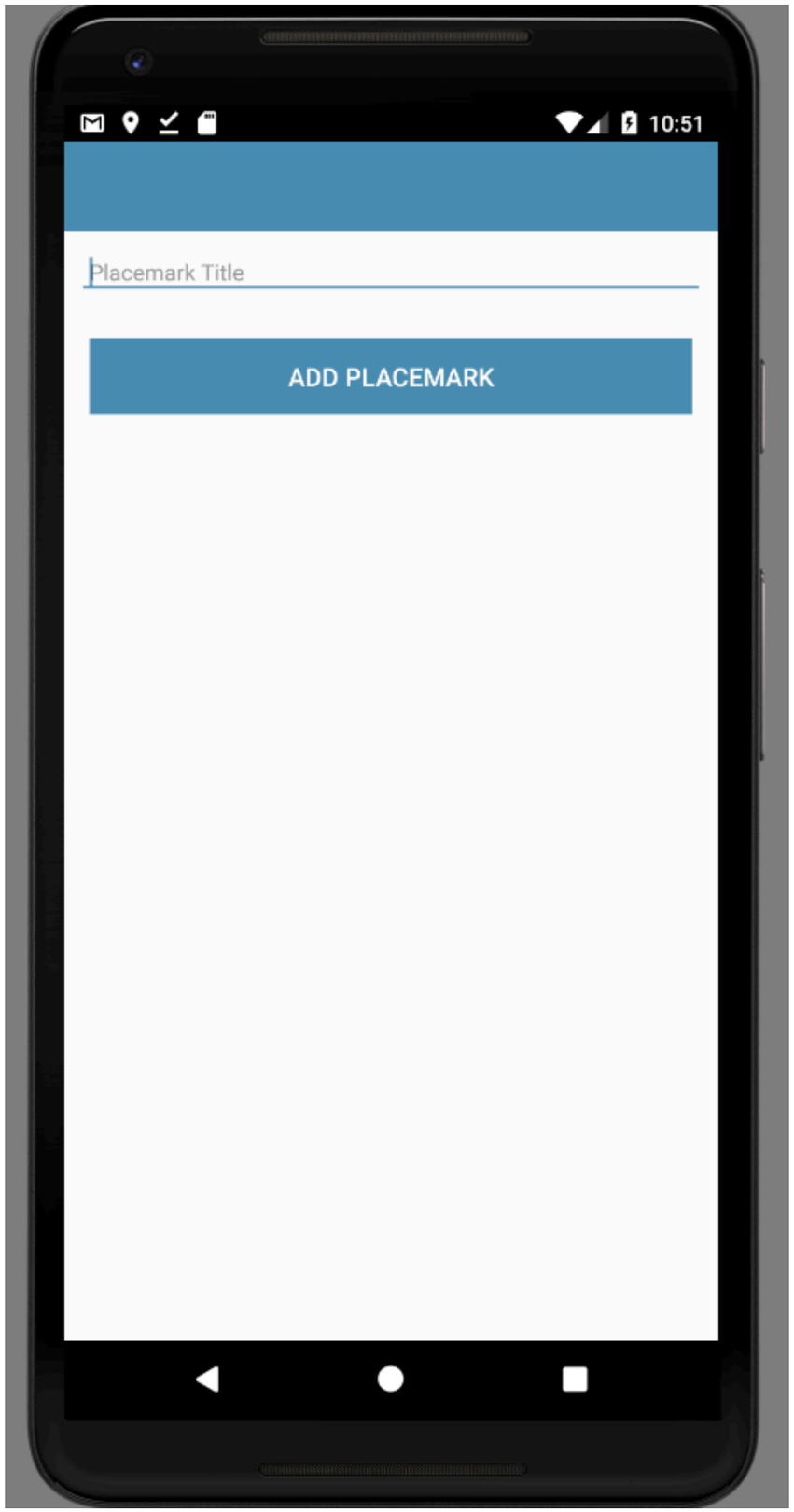
colours.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <color name="colorPrimary">#FFFFFF</color>
  <color name="colorPrimaryDark">#000000</color>
  <color name="colorAccent">#4c90af</color>
</resources>
```

As generated by Android Studio


```
<?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="org.wit.placemark.PlacemarkActivity">

  <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Hello World!"
      app:layout_constraintBottom_toBottomOf="parent"
      app:layout_constraintLeft_toLeftOf="parent"
      app:layout_constraintRight_toRightOf="parent"
      app:layout_constraintTop_toTopOf="parent"/>

</android.support.constraint.ConstraintLayout>
```

Revised Layout




```
<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="org.wit.placemark.PlacemarkActivity">

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

  <android.support.design.widget.AppBarLayout
      android:id="@+id/appBarLayout"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:background="@color/colorAccent"
      android:fitsSystemWindows="true"
      app:elevation="0dip"
      app:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar">

    <android.support.v7.widget.Toolbar
        android:id="@+id/toolbarAdd"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:titleTextColor="@color/colorPrimary"/>
  </android.support.design.widget.AppBarLayout>

  <ScrollView
      android:layout_width="match_parent"
      android:layout_height="match_parent"
      android:layout_below="@id/appBarLayout"
      android:fillViewport="true">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">

      <android.support.design.widget.TextInputEditText
          android:id="@+id/placemarkTitle"
          android:layout_width="match_parent"
          android:layout_height="wrap_content"
          android:layout_margin="8dp"
          android:hint="@string/hint_placemarkTitle"
          android:inputType="text"
          android:maxLength="25"
          android:maxLines="1"
          android:padding="8dp"
          android:textColor="@color/colorPrimaryDark"
          android:textSize="14sp"/>

      <Button
          android:id="@+id/btnAdd"
          android:layout_width="match_parent"
          android:layout_height="wrap_content"
          android:layout_margin="16dp"
          android:background="@color/colorAccent"
          android:paddingBottom="8dp"
          android:paddingTop="8dp"
          android:stateListAnimator="@null"
          android:text="@string/button_addPlacemark"
          android:textColor="@color/colorPrimary"
          android:textSize="16sp"/>
    </LinearLayout>
  </ScrollView>

</RelativeLayout>
</android.support.constraint.ConstraintLayout>
```

gradle.build

```
...
  implementation 'org.jetbrains.anko:anko:0.10.3'
  implementation 'org.jetbrains.anko:anko-commons:0.10.3'
...
```

PlacemarkActivity

```
..
class PlacemarkActivity : AppCompatActivity(), AnkoLogger {
...
```

```
...
    info("Placemark Activity started..")
...
```

```
12-21 11:07:20.590 21263-21263/org.wit.placemark I/P  cemarkActiv
```

Introduce Logging Support

Anko Commons – Logging

Narbonne edited this page on 26 May 2017 · 4 revisions

Contents

Using `AnkoLogger` in your project

`AnkoLogger` is inside the `anko-commons` artifact. Add it as a dependency to your `buil`

```
dependencies {
    compile "org.jetbrains.anko:anko-commons:$anko_version"
}
```

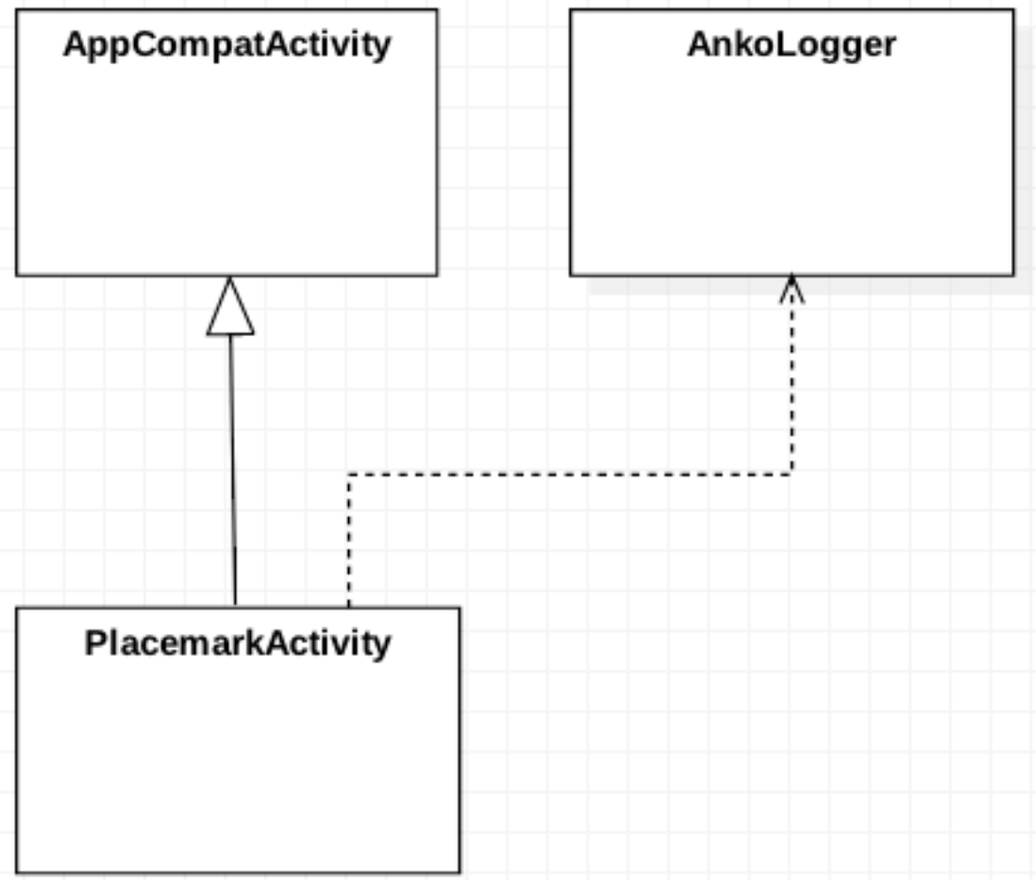
Trait-like style

Android SDK provides `android.util.Log` class with some logging methods. Usage i straightforward though the methods require you to pass a `tag` argument. You can this with using `AnkoLogger` trait-like interface:

```
class SomeActivity : Activity(), AnkoLogger {
    private fun someMethod() {
        info("London is the capital of Great Britain")
        debug(5) // .toString() method will be executed
        warn(null) // "null" will be printed
    }
}
```

Updated Class Diagram

```
package org.wit.placemark

import android.support.v7.app.AppCompatActivity
import android.os.Bundle
import kotlinx.android.synthetic.main.activity_placemark.*
import org.jetbrains.anko.AnkoLogger
import org.jetbrains.anko.info

class PlacemarkActivity : AppCompatActivity(), AnkoLogger {

  override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_placemark)

    info("Placemark Activity started..")

    btnAdd.setOnClickListener() {
      info("add Button Pressed")
    }
  }
}
```



Event Handler

PlacemarkActivity

```
btnAdd.setOnClickListener() {
  info("add Button Pressed")
}
```

Kotlin Extensions automatically binds button widget (in xml) to a class attribute (btnAdd) of the same name

activity_placemark.xml

```
<Button
    android:id="@+id/btnAdd"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="16dp"
    android:background="@color/colorAccent"
    android:paddingBottom="8dp"
    android:paddingTop="8dp"
    android:stateListAnimator="@null"
    android:text="@string/button_addPlacemark"
    android:textColor="@color/colorPrimary"
    android:textSize="16sp"/>
```

```
package org.wit.placemark

import android.support.v7.app.AppCompatActivity
import android.os.Bundle
import kotlinx.android.synthetic.main.activity_placemark.*
import org.jetbrains.anko.AnkoLogger
import org.jetbrains.anko.info

class PlacemarkActivity : AppCompatActivity(), AnkoLogger {

  override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_placemark)

    info("Placemark Activity started..")

    btnAdd.setOnClickListener() {
      info("add Button Pressed")
    }
  }
}
```

Event Handler

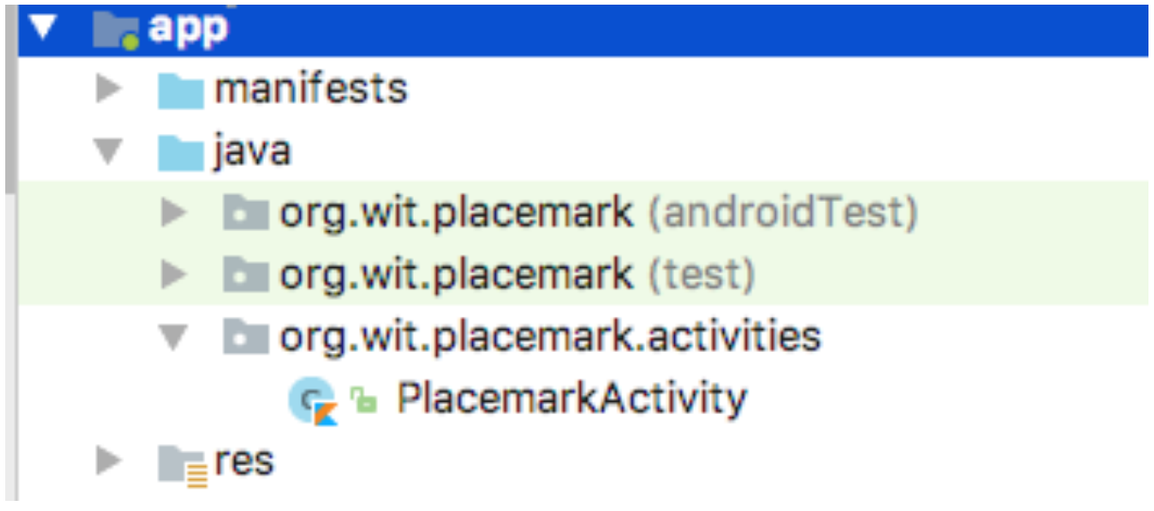
PlacemarkActivity

```
btnAdd.setOnClickListener() {
  val placemarkTitle = placemarkTitle.text.toString()
  if (placemarkTitle.isNotEmpty()) {
    info("add Button Pressed: $placemarkTitle")
  }
  else {
    toast ("Please Enter a title")
  }
}
```

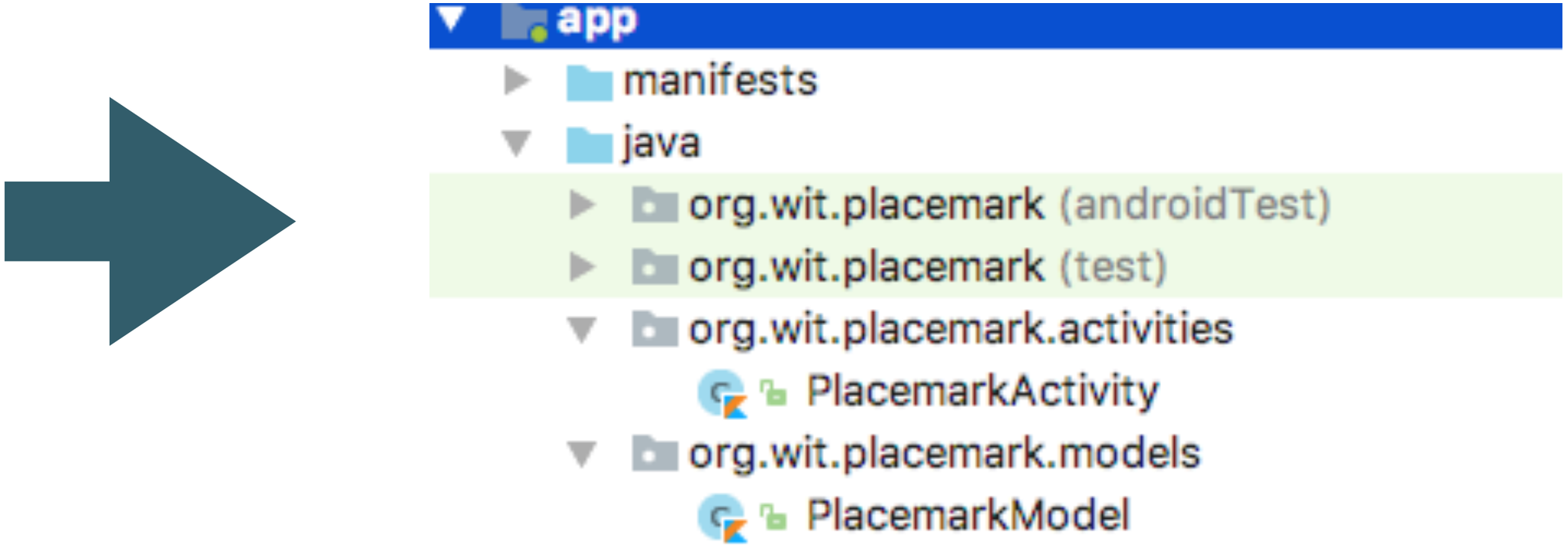
```
<android.support.design.widget.TextInputEditText
    android:id="@+id/placemarkTitle"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="8dp"
    android:hint="@string/hint_placemarkTitle"
    android:inputType="text"
    android:maxLength="25"
    android:maxLines="1"
    android:padding="8dp"
    android:textColor="@color/colorPrimaryDark"
    android:textSize="14sp"/>
```

activity_placemark.xml

Kotlin Extensions automatically binds TextInputEditText widget : placemarkTitle to a class of the same name

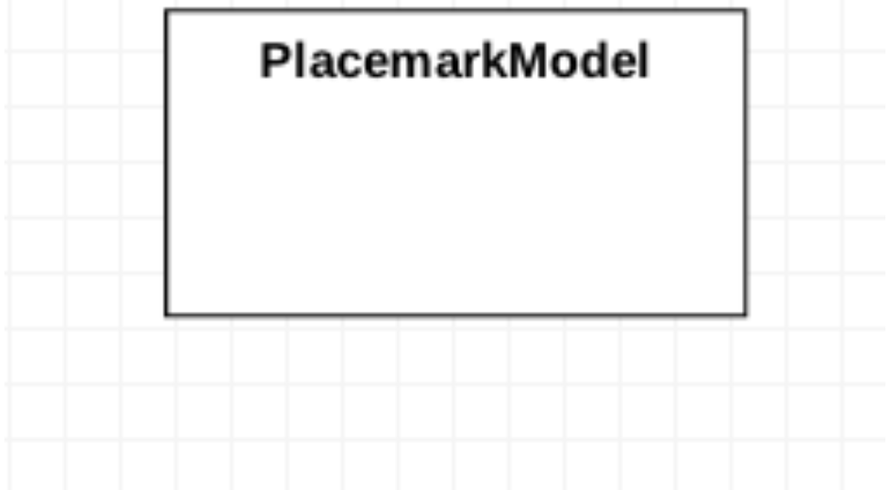


PlacemarkModel



PlacemarkModel

```
package org.wit.placemark.models

data class PlacemarkModel(var title: String = "")
```

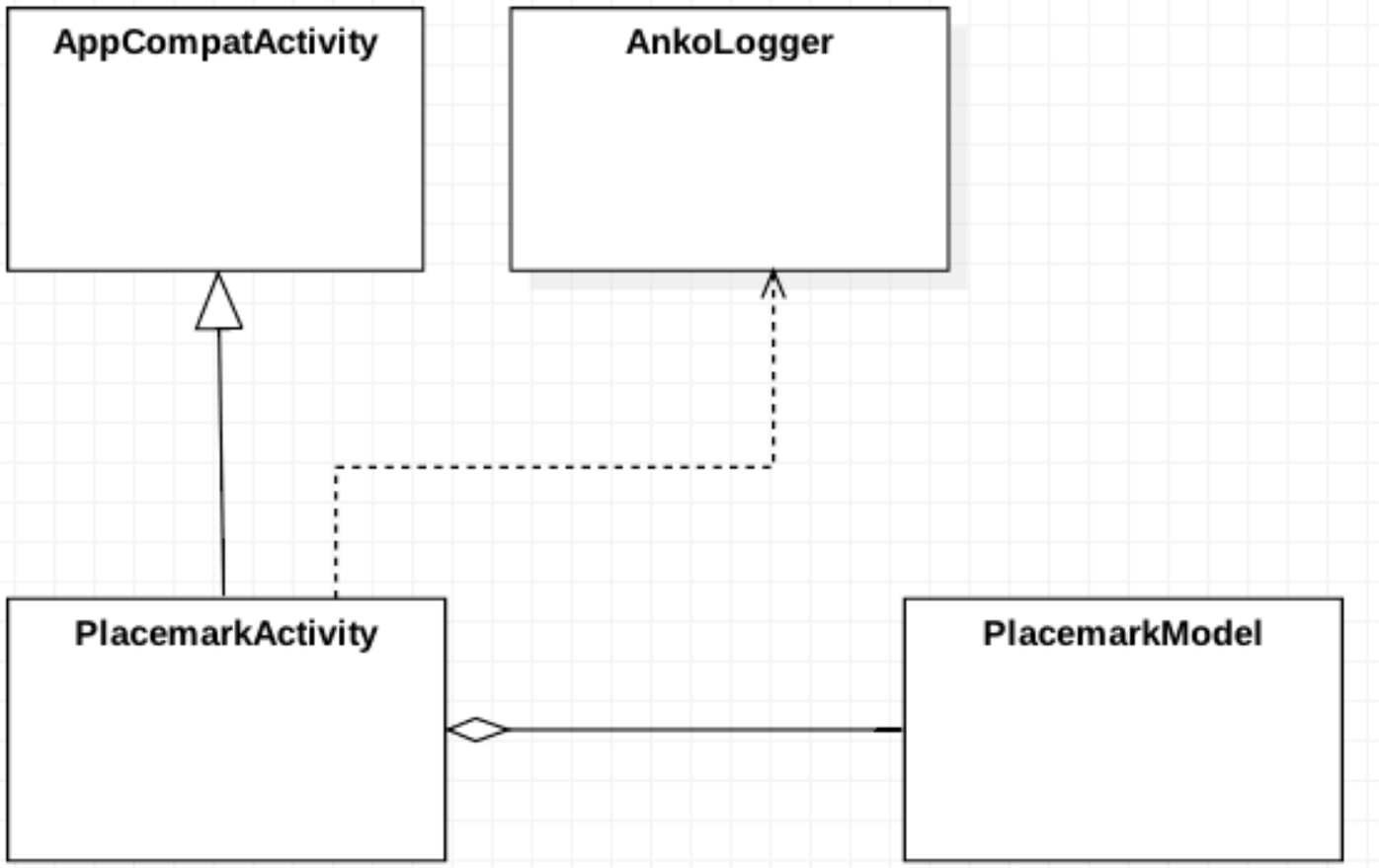


```
package org.wit.placemark.activities

import android.support.v7.app.AppCompatActivity
import android.os.Bundle
import kotlinx.android.synthetic.main.activity_placemark.*
import org.jetbrains.anko.AnkoLogger
import org.jetbrains.anko.info
import org.jetbrains.anko.toast
import org.wit.placemark.R
import org.wit.placemark.models.PlacemarkModel

class PlacemarkActivity : AppCompatActivity(), AnkoLogger {

  var placemark = PlacemarkModel()

  override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_placemark)

    btnAdd.setOnClickListener() {
      placemark.title = placemarkTitle.text.toString()
      if (placemark.title.isNotEmpty()) {
        info("add Button Pressed: $placemarkTitle")
      }
      else {
        toast ("Please Enter a title")
      }
    }
  }
}
```

Placemark V1 Class Diagram

```
class PlacemarkActivity : AppCompatActivity(), AnkoLogger {

  var placemark = PlacemarkModel()

  override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_placemark)

    btnAdd.setOnClickListener() {
      placemark.title = placemarkTitle.text.toString()
      if (placemark.title.isNotEmpty()) {
        info("add Button Pressed: $placemarkTitle")
      }
      else {
        toast ("Please Enter a title")
      }
    }
  }
}
```

```
package org.wit.placemark.models

data class PlacemarkModel(var title: String = "")
```



```
var placemark = PlacemarkModel()
```

```
btnAdd.setOnClickListener() {
  placemark.title = placemarkTitle.text.toString()
  if (placemark.title.isNotEmpty()) {
    info("add Button Pressed: $placemarkTitle")
  }
  else {
    toast ("Please Enter a title")
  }
}
```