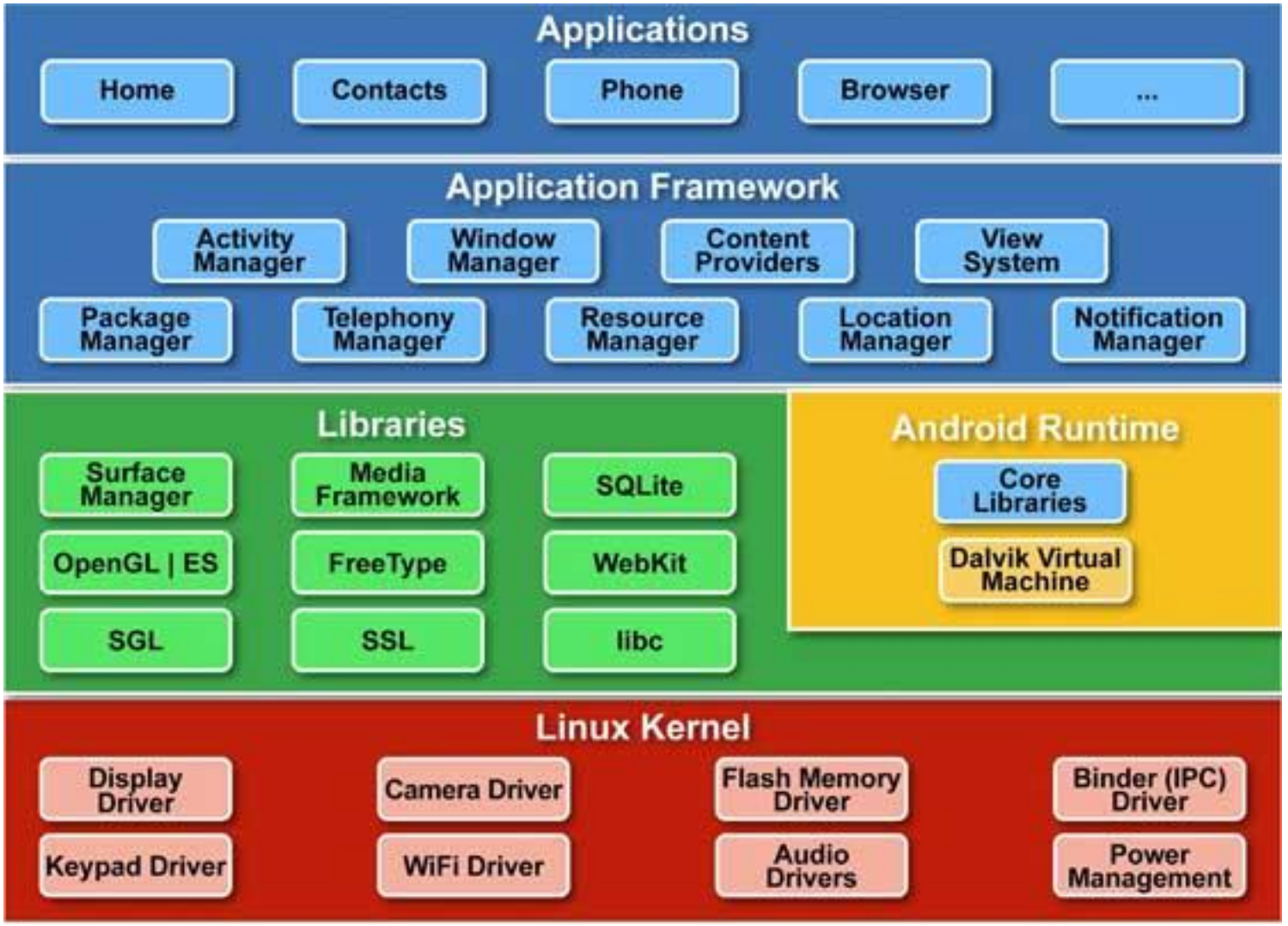
# Introduction To Android

Android Architecture, Android App Structure

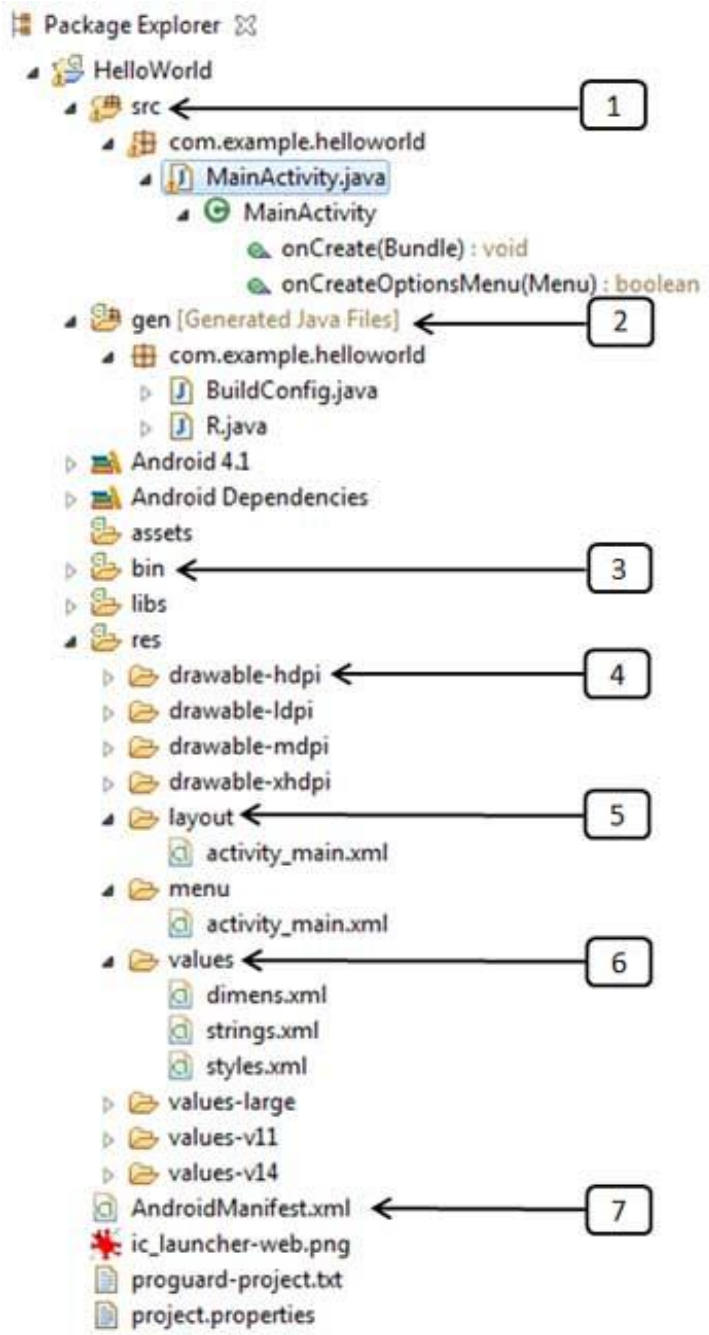
# Android Architecture



# Application Components

| Components | Description |
| --- | --- |
| Activities | They dictate the UI and handle the user interaction to the smartphone screen |
| Services | They handle background processing associated with an application. |
| Broadcast Receivers | They handle communication between Android OS and applications. |
| Content Providers | They handle data and database management issues. |

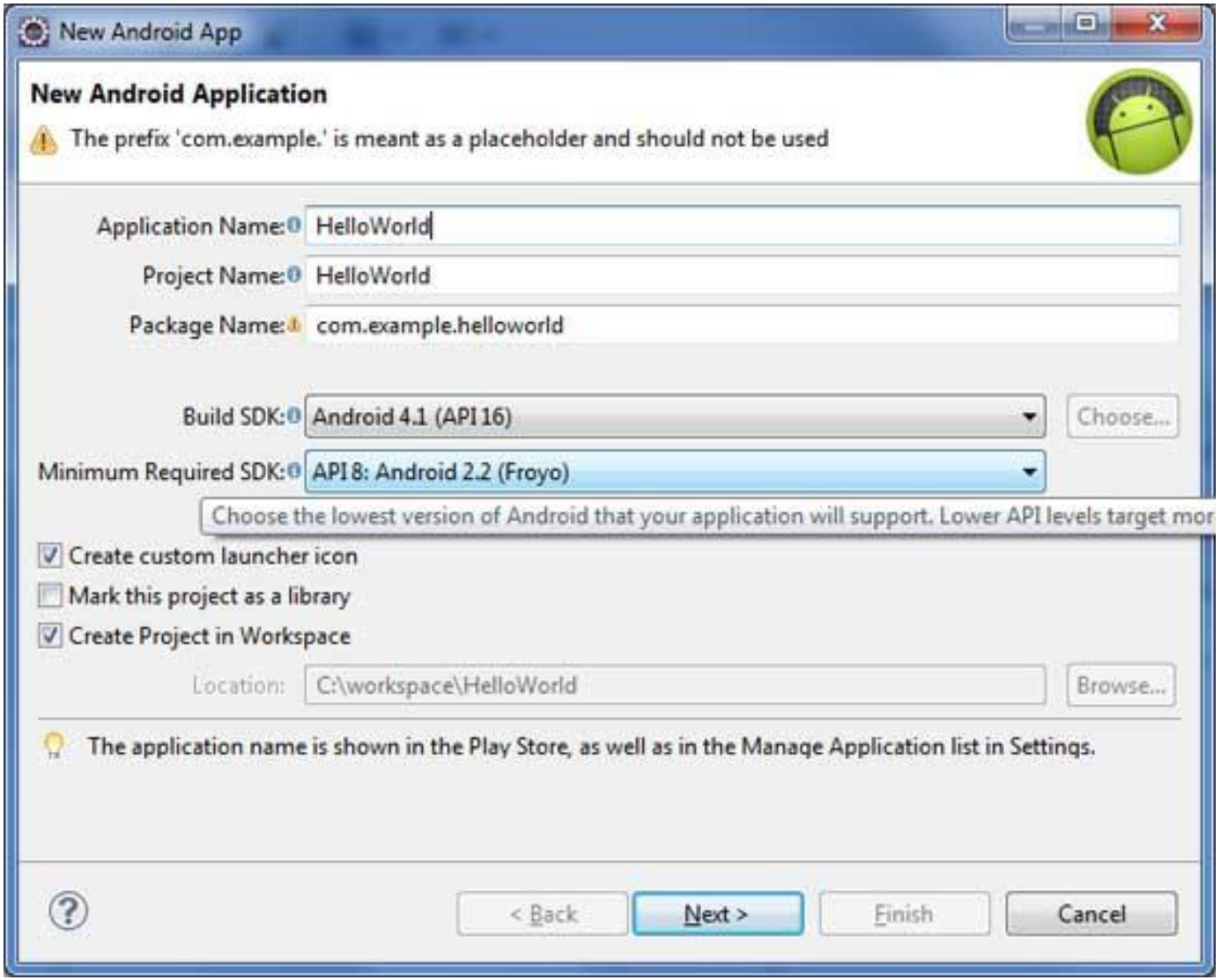
# Application Structure



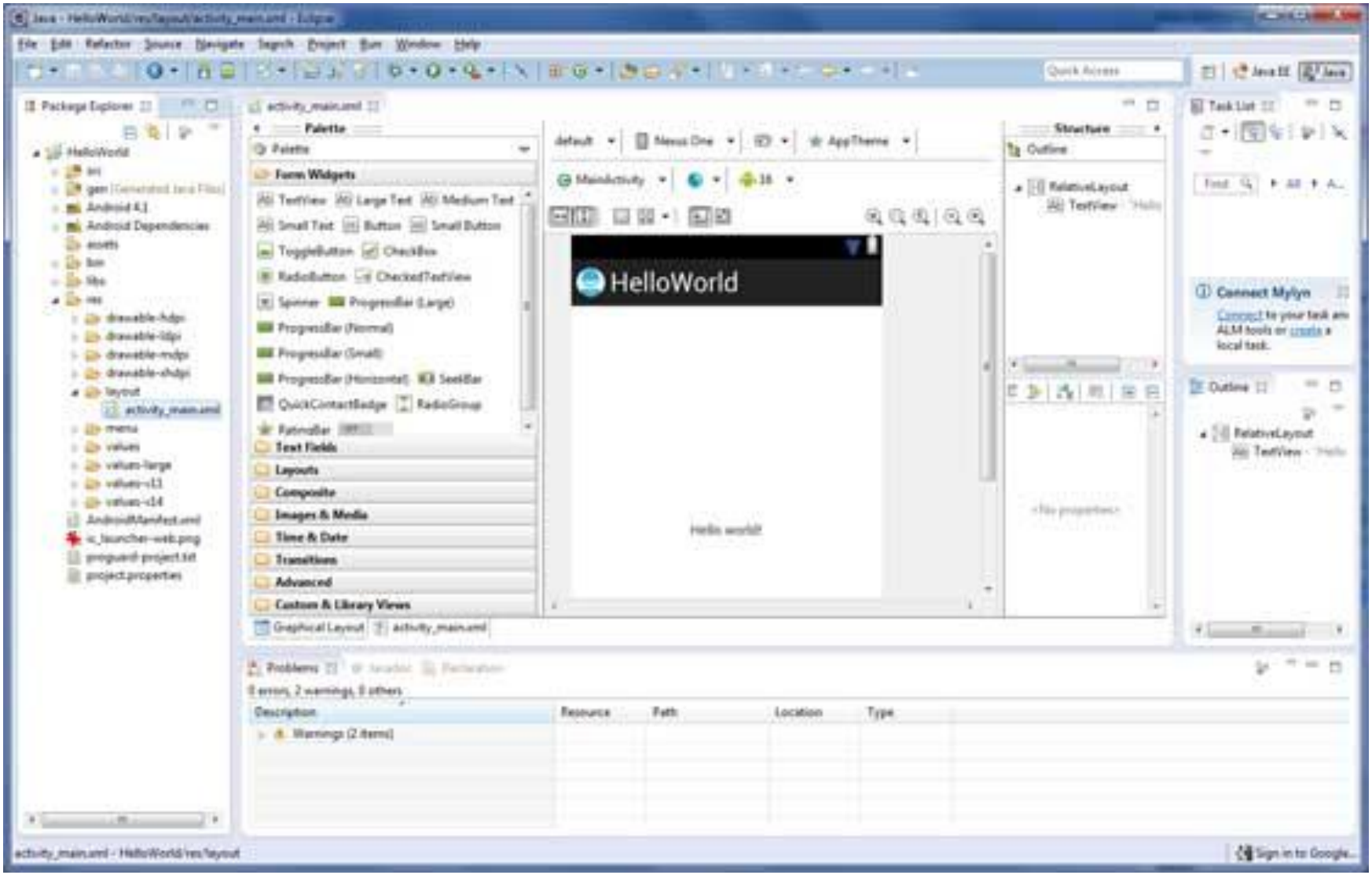
# Application Folders/Files

| | |
|---|---|
| ١ | src<br>This contains the .java source files for your project. By default, it includes an MainActivity.java source file having an activity class that runs when your app is launched using the app icon. |
| ٢ | gen<br>This contains the .R file, a compiler-generated file that references all the resources found in your project. You should not modify this file. |
| ٣ | bin<br>This folder contains the Android package files .apk built by the ADT during the build process and everything else needed to run an Android application. |
| ٤ | res/drawable-hdpi<br>This is a directory for drawable objects that are designed for high-density screens. |
| ٥ | res/layout<br>This is a directory for files that define your app's user interface. |
| ٦ | res/values<br>This is a directory for other various XML files that contain a collection of resources, such as strings and colors definitions. |
| ٧ | AndroidManifest.xml<br>This is the manifest file which describes the fundamental characteristics of the app and defines each of its components. |

# Creating New Application in Eclipse



# Creating New Application in Eclipse



# The Main Activity File

```
package com.example.helloworld;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.view.MenuItem;
import android.support.v4.app.NavUtils;

public class MainActivity extends Activity {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.activity_main, menu);
        return true;
    }
}
```

# The Manifest File

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
   package="com.example.helloworld"
   android:versionCode="1"
   android:versionName="1.0" >
   <uses-sdk
      android:minSdkVersion="8"
      android:targetSdkVersion="15" />
   <application
       android:icon="@drawable/ic_launcher"
       android:label="@string/app_name"
       android:theme="@style/AppTheme" >
       <activity
           android:name=".MainActivity"
           android:label="@string/title_activity_main" >
           <intent-filter>
               <action android:name="android.intent.action.MAIN" />
               <category android:name="android.intent.category.LAUNCHER"/>
           </intent-filter>
       </activity>
   </application>
</manifest>
```

# The String Resource File

```
<resources>
    <string name="app_name">HelloWorld</string>
    <string name="hello_world">Hello world!</string>
    <string name="menu_settings">Settings</string>
    <string name="title_activity_main">MainActivity</string>
</resources>
```

# The Layout File

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:paddingLeft="@dimen/activity_horizontal_margin"
   android:paddingRight="@dimen/activity_horizontal_margin"
   android:paddingTop="@dimen/activity_vertical_margin"
   android:paddingBottom="@dimen/activity_vertical_margin"
   tools:context=".MainActivity">

   <TextView android:text="@string/hello_world"
      android:id="@+id/txt"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content" />
</RelativeLayout>
```