



ICT PROJECT DEVELOPMENT AND IMPLEMENTATION

SESSION #1

ANDROID FUNDAMENTALS

OUTLINE



- ✧ What is Android?
 - Android History
- ✧ Android Fundamentals
 - Android Architecture
 - Android applications
- ✧ Android Studio
- ✧ Hello World App
- ✧ Mockup

WHAT IS ANDROID?



- ✧ According to Google:
 - ✧ Android, the world's most popular mobile platform
 - ✧ Global partnerships and large installed base
 - ✧ Rapid innovation
 - ✧ Powerful development framework
 - ✧ Open marketplace for distributing your apps
- ✧ Android is an operating system for mobile devices such as smartphones and tablets.
 - ✧ Low-consumption, connectivity and user-friendly

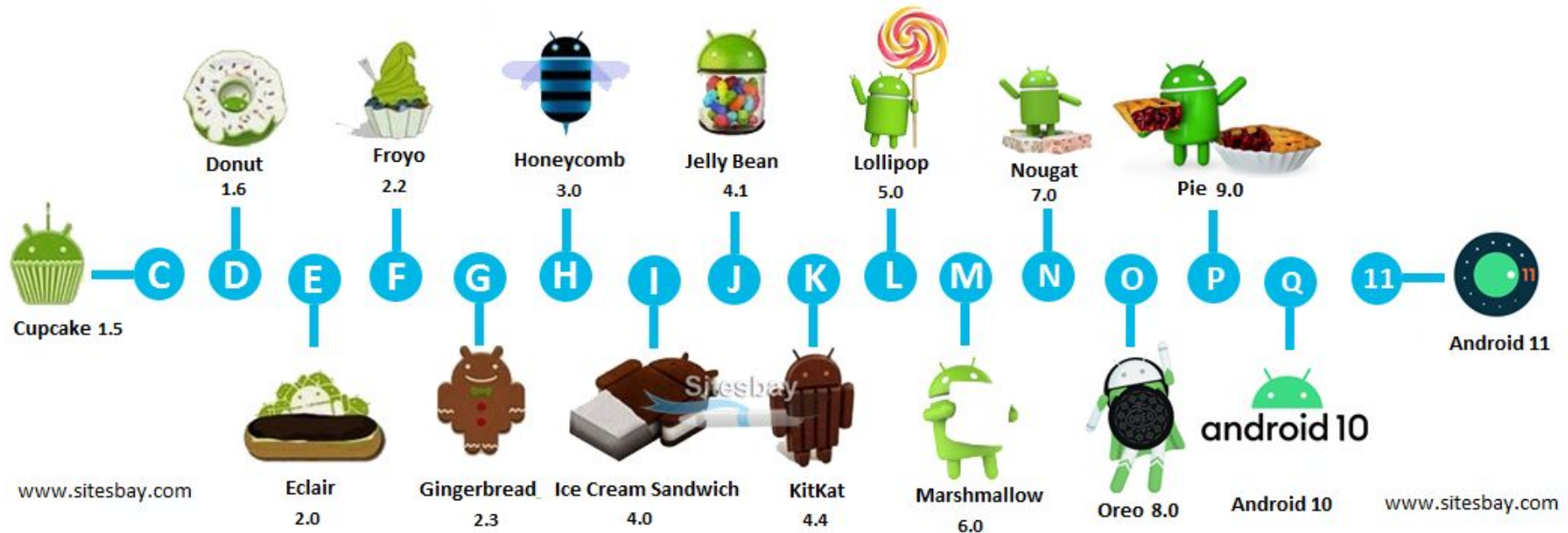
ANDROID HISTORY



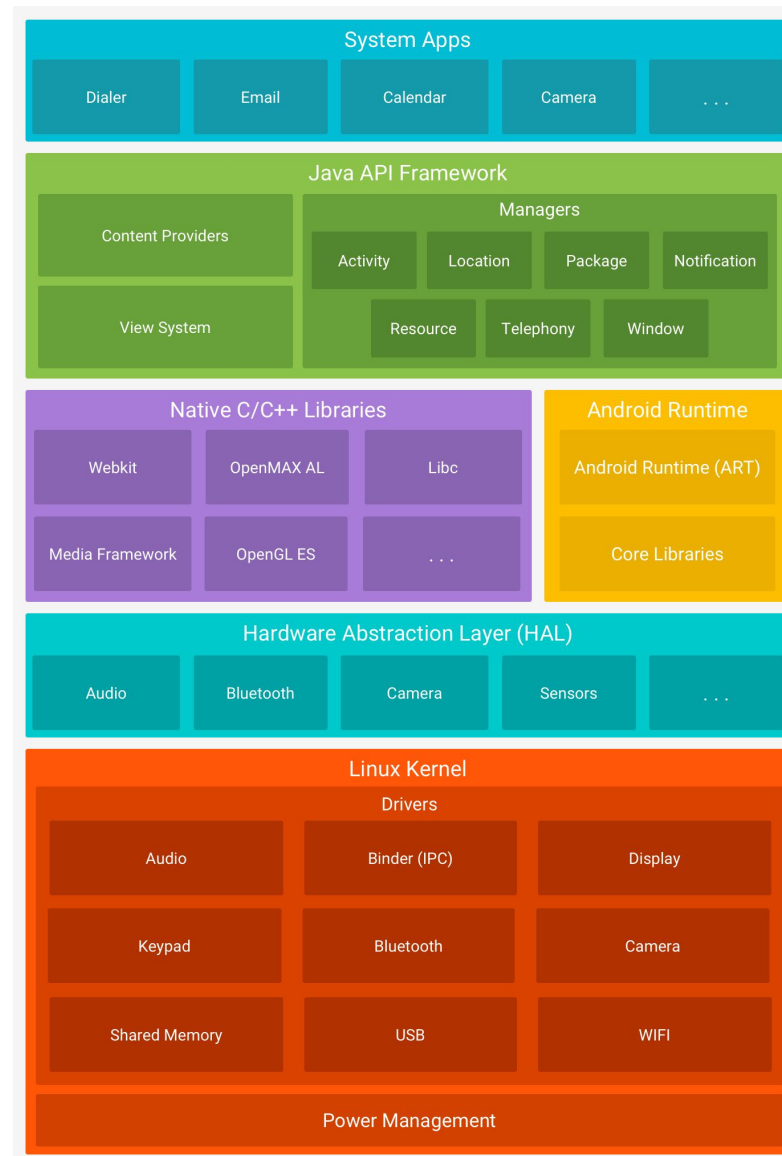
- ✧ Android 1.0 (2008) Alpha
- ✧ Android 1.1 (2008) Beta
- ✧ Android 1.5 (2009) Cupcake
- ✧ Android 1.6 (2009) Donut
- ✧ Android 2.0 (2009) Eclair
- ✧ Android 2.2 (2010) Froyo
- ✧ Android 2.3 (2010) Ginger Bread
- ✧ Android 3.0 (2011) Honeycomb
- ✧ Android 4.0 (2011) Ice cream Sandwich
- ✧ Android 4.1 (2012) Jelly Bean
- ✧ Android 4.4 (2013) Kitkat
- ✧ Android 5.0 (2014) Lollipop
- ✧ Android 6.0 (2015) Marshmallow
- ✧ Android N (2016) Nougat

<https://www.android.com/history/>

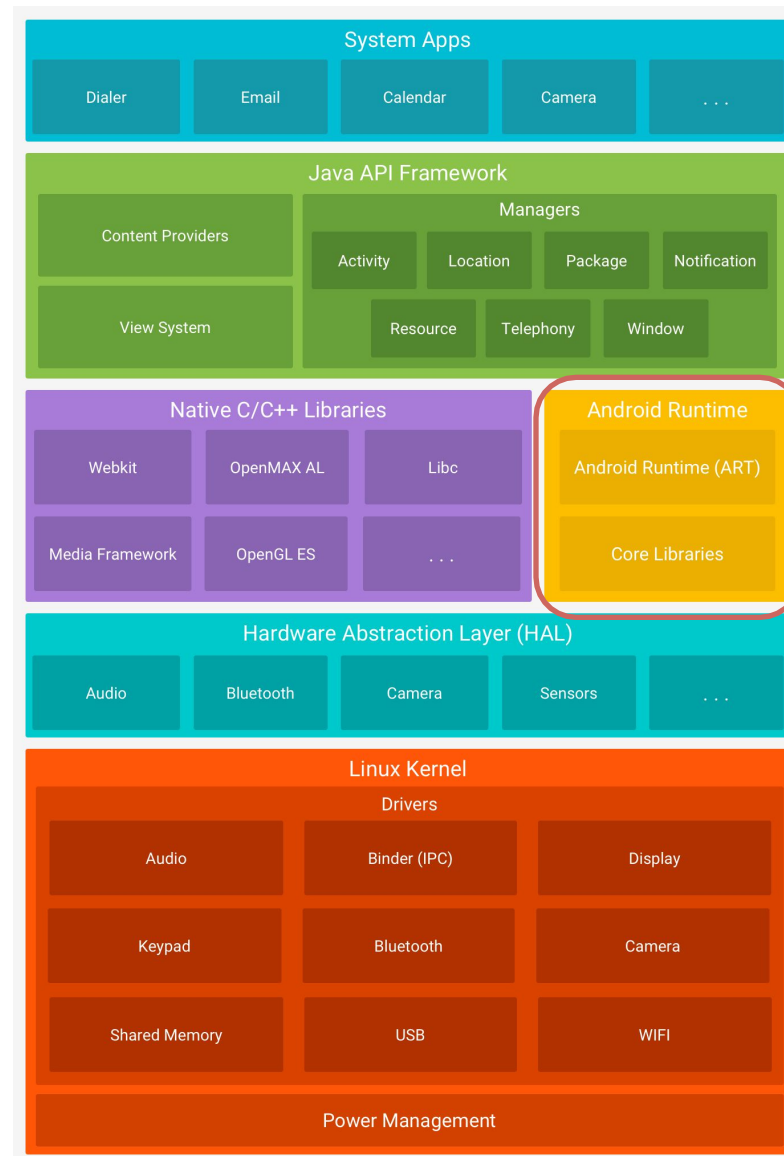
ANDROID HISTORY



ANDROID ARCHITECHTURE



ANDROID ARCHITECTURE



ANDROID RUNTIME



- ✧ For devices running **Android version 5.0 (API level 21) or higher**, each app runs in its own process and with its own instance of the Android Runtime (ART).
- ✧ ART is written to run multiple virtual machines on low-memory devices by executing **DEX files, a bytecode format designed specially for Android** that's optimized for minimal memory footprint.
- ✧ **Prior to Android version 5.0 (API level 21), Dalvik was the Android runtime.** If your app runs well on ART, then it should work on Dalvik as well, but the reverse may not be true.
- ✧ Android also includes a set of core runtime libraries that **provide most of the functionality of the Java programming language**, including some Java 8 language features, that the Java API framework uses.

APP FRAMEWORK



FEATURES

| Feature | Role |
|----------------------|--|
| View System | Used to build an application, including lists, grids, text boxes, buttons and embedded web browser |
| Content Provider | Enabling applications to access data from other applications or to share their own data |
| Resource Manager | Providing access to non-code resources (localized strings, Graphics and layout files) |
| Notification Manager | Enabling all applications to display customer alerts in the status bar |
| Activity Manager | Managing the lifecycle of applications and providing a common navigation backstack |

ANDROID APPLICATIONS



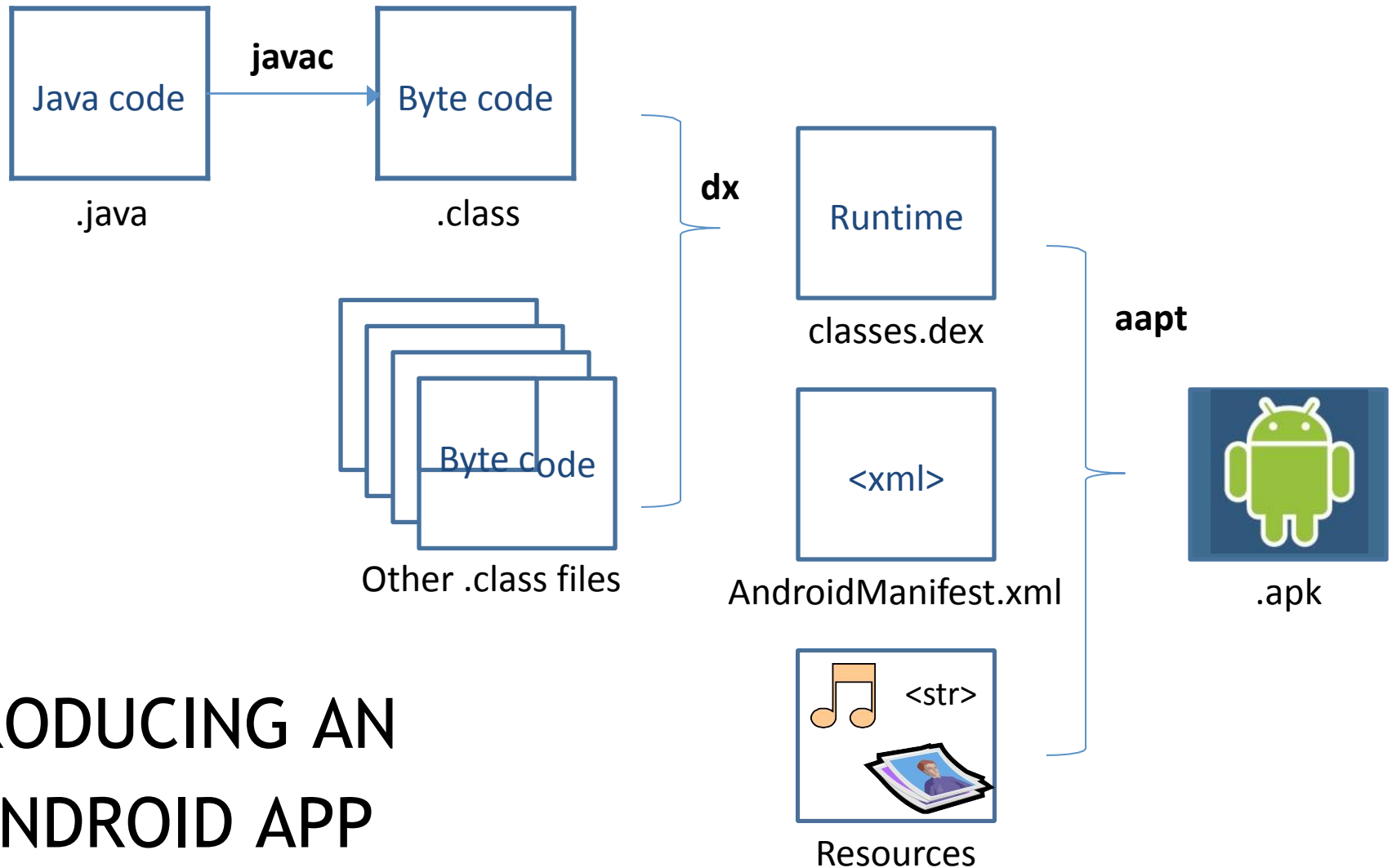
```
package com.google.android.helloactivity;

import android.app.Activity;
import android.os.Bundle;

public class HelloActivity extends Activity {
    public HelloActivity() {
    }
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.hello_activity);
    }
}
```

Android applications
are written in Java +
Android SDK Libraries

ANDROID APPLICATIONS



PRODUCING AN
ANDROID APP

ANDROID APPLICATIONS



- ✧ Applications are **boxed**. By default, each app is run in its own Linux process
 - Process started when app's code needs to be executed
 - Threads can be started to **handle time-consuming operations**
- ✧ By default, each app is assigned unique Linux ID
 - Permissions are set so app's files **are only visible** to that app
- ✧ Processes **may be killed** to reclaim resources

ANDROID APPLICATIONS



- ✧ Android designed to enable **reuse of components** in other applications
- ✧ Each application can **publish its capabilities** which other apps can use
- ✧ Android applications **don't have a single entry point** (main function, for example)
- ✧ They have essential components that the system can instantiate and run as needed.
- ✧ **Tools:** JDK + Android SDK + Android Studio + Device

APPLICATION STRUCTURE



Views such as lists, grids, text boxes, buttons, and even an embeddable web browser

Content Providers that enable applications to access data from other applications or to share their own data

An **Activity Manager** that manages the life cycle of applications and provides a common navigation back stack

A **Notification Manager** that enables all apps to display custom alerts in the status bar

A **Resource Manager**, providing access to non-code resources such as localized strings, graphics, and layout files

APPLICATION STRUCTURE



Broadcast receivers can trigger intents that start an application

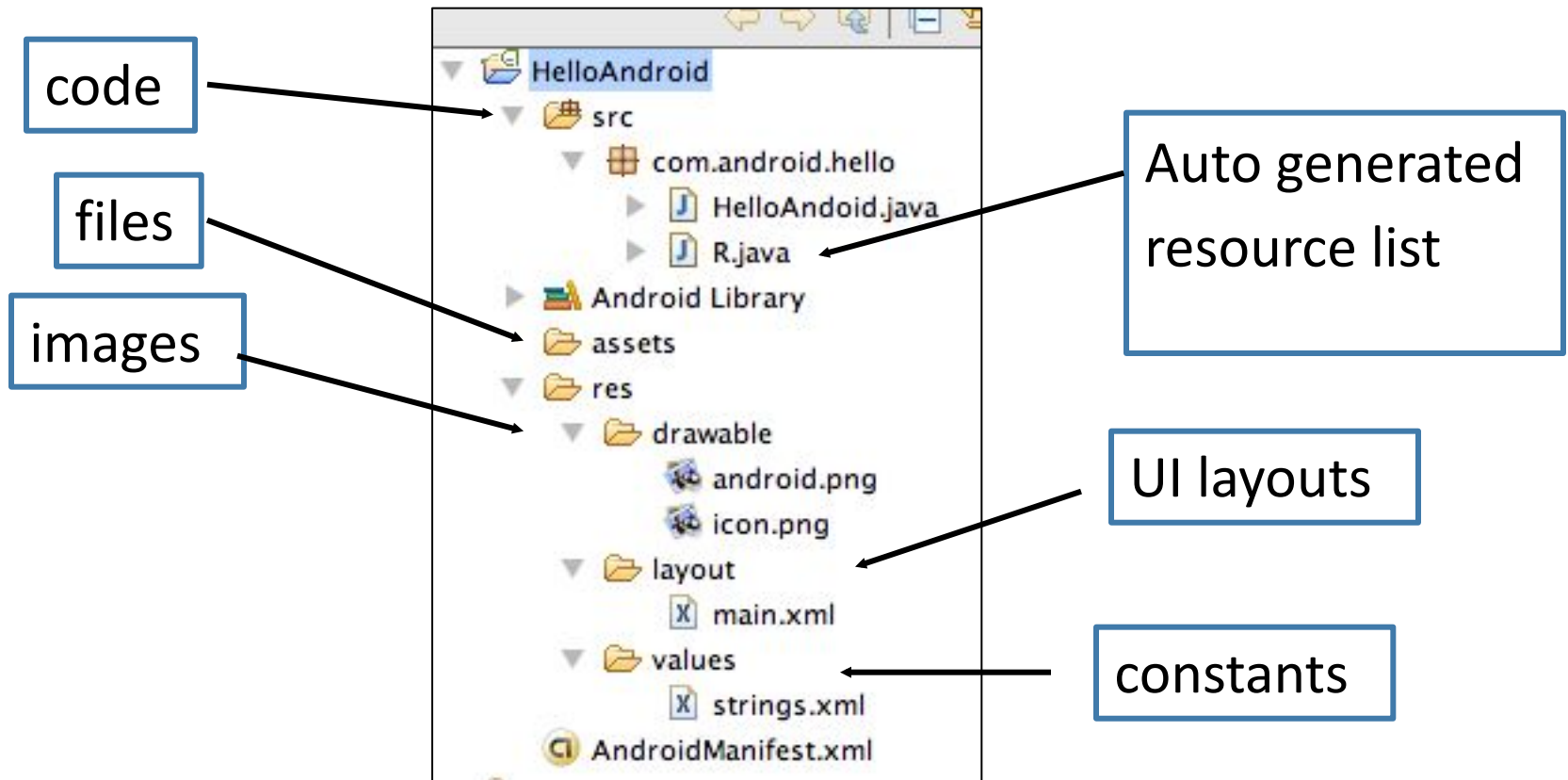
Data storage provide data for your apps, and can be shared between apps - database, file, and shared preferences used by group of applications

Activity is the presentation layer of your app: there will be one per screen, and the Views provide the UI to the activity

Intents specify what specific action should be performed

Services run in the background and have no UI for the user - they will update data, and trigger events

APP FILES STRUCTURE



ANDROID STUDIO



- ✧ Android Studio is the official IDE for Android application development, based on IntelliJ IDEA.
- ✧ Main Features:
 - Flexible Gradle-based build system
 - Build variants and multiple apk file generation
 - Code templates to help build common app features
 - Rich layout editor with support for drag and drop theme editing
 - lint tools to catch performance, usability, version compatibility, and other problems
 - ProGuard and app-signing capabilities
 - Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine

ANDROID STUDIO



INSTALLATION

- ✧ Download Android Studio from:
<https://developer.android.com/sdk/index.html>
- ✧ Set up Android Studio on Linux¹:
 - Unpack the downloaded ZIP file into an appropriate location.
 - To launch Android Studio, navigate to the android-studio/ bin/ directory in a terminal and execute studio.sh.
 - Add android-studio/bin/ to your PATH environmental variable so that you can start Android Studio from any directory.
 - if the SDK is not already installed, follow the setup wizard to install the SDK and any necessary SDK tools.

¹You may also need to install the `ia32-libs`, `lib32z1`, `lib32ncurses5-dev`, and `lib32stdc++6` packages. These packages are required to support 32-bit apps on a 64-bit machine.

ANDROID STUDIO

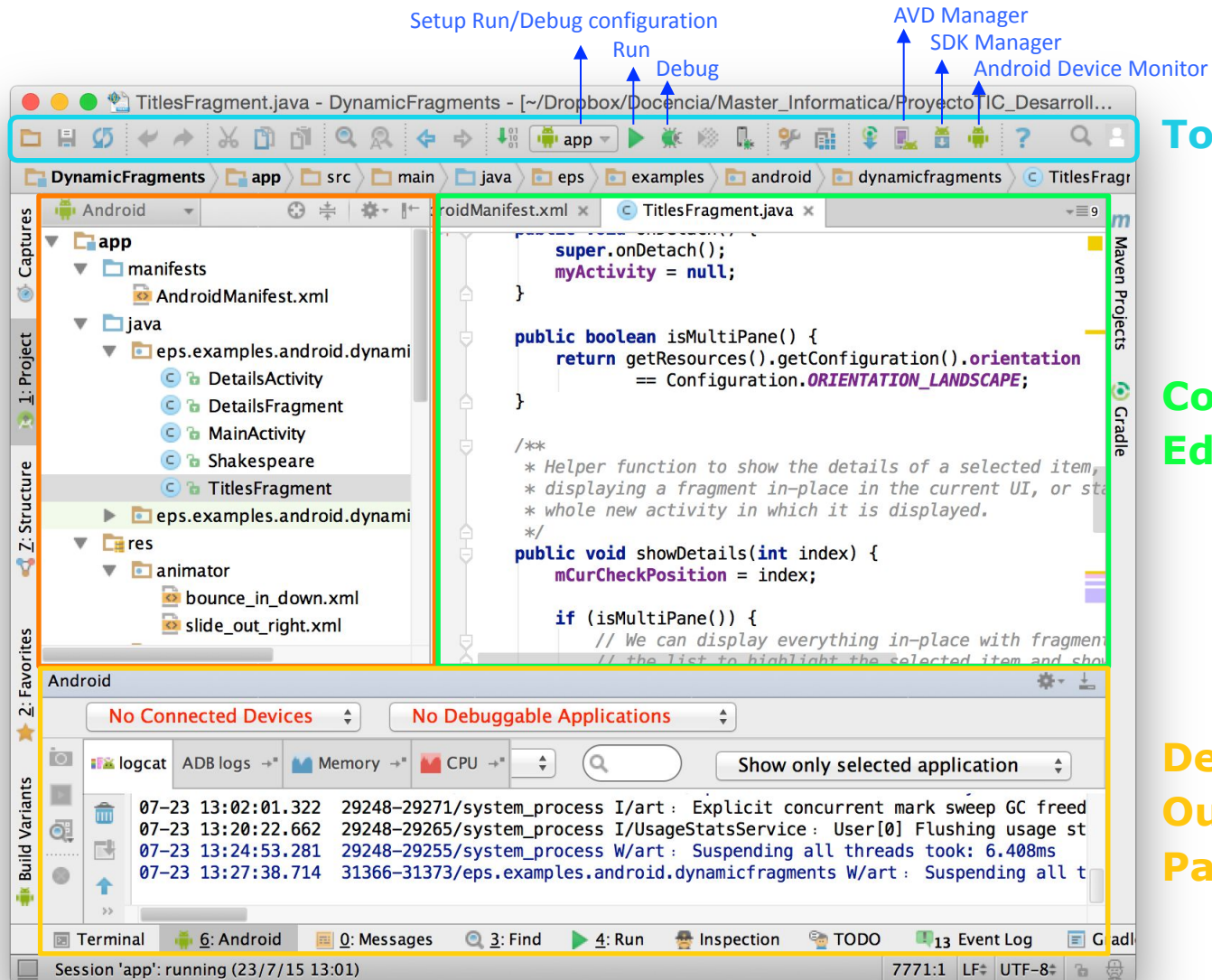


SDK PACKAGES

- The Android SDK does not include everything you need to start developing (different Android versions).
- The SDK separates tools, platforms, and other components into packages can be downloaded using the Android SDK Manager.
 1. Android SDK
 2. SDK Platforms
 3. SDK Tools
 4. SDK Update sites
 5. Check ☐ Show Package Details

INTERFACE

ANDROID STUDIO



Tools Bar

Code Editor

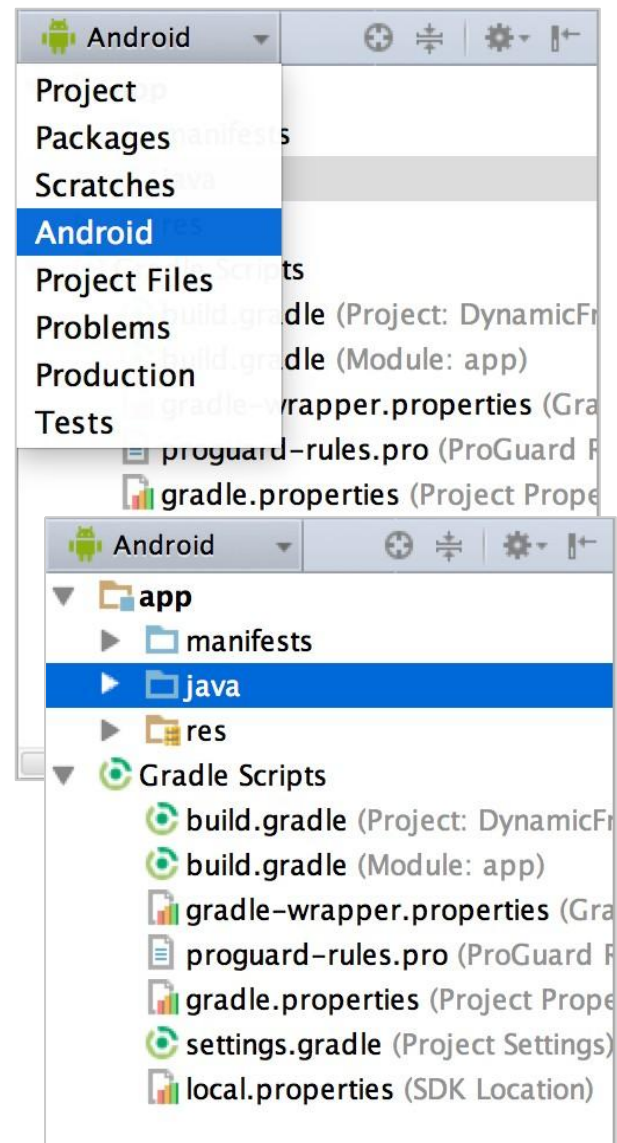
Debug & Output Panel

ANDROID STUDIO



ANDROID PROJECT VIEW

- ✧ This view shows a flattened version of your project's structure that provides quick access to the key source files of Android projects and work with the Gradle--based build system:
- Shows the most important source directories at the top level of the module hierarchy.
 - Groups the build files for all modules in a folder.
 - Shows resource files from all Gradle source sets
 - Groups resource files for different locales, orientations, and screen types in a single group per resource type.



THE CODE EDITOR



✧ Main Features:

- **Code Completion:** Code completion helps us to write code quickly by suggestion lists and automatically completing the code.
- **Code Generation:** Allow to automatically generate blocks of code in a class such as constructors, getters, and setters methods, some statements (**if**, **while**, **for**,...) or templates.

✧ Useful shortcuts:

- Alt + enter: import classes
- Ctrl + O: select methods to override/implement
- Ctrl + I: Implement methods
- Ctrl + P: Show parameters for selected method

THE CODE EDITOR



✧ Useful shortcuts:

- Ctrl + /: Comments each line of the selected code. To use block comments press Ctrl + Shift + /.
- Ctrl + Alt + I: Indents the selected code.
- Ctrl + Alt + O: Optimizes the imports, removing the unused ones and reordering the rest of them.
- Shift + Ctrl + Arrows: Moves the selected code to another line.
- Ctrl + F: Finds a string in the active tab of the editor.
- Ctrl + R: Replaces a string in the active tab of the editor.
- Ctrl + D: Duplicate current line or
- Ctrl + Y: Removes the entire line without leaving any blank line.

<https://developer.android.com/studio/intro/keyboard--shortcuts.html>

ANDROID EMULATOR - AVD



- ✧ Emulator is essential to testing app but is not a substitute for a real device
- ✧ Emulators are called **Android Virtual Devices (AVDs)**
- ✧ Android SDK and AVD Manager allows you to create AVDs that target any Android API level
- ✧ AVD have configurable resolutions, RAM, SD cards, skins, and other hardware



ANDROID EMULATOR - AVD



LIMITATIONS

The Android Emulator supports most features of a device, but doesn't include virtual hardware for:

- WiFi
- Bluetooth
- NFC
- SD card insert/eject
- Device-attached headphones
- USB

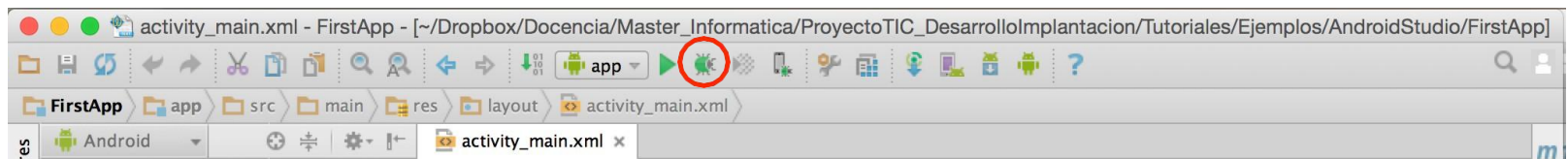
<http://developer.android.com/guide/developing/tools/emulator.html>

DEBUGGING



Android Studio enables you to debug apps running on the emulator or on an Android device. With Android Studio, you can:

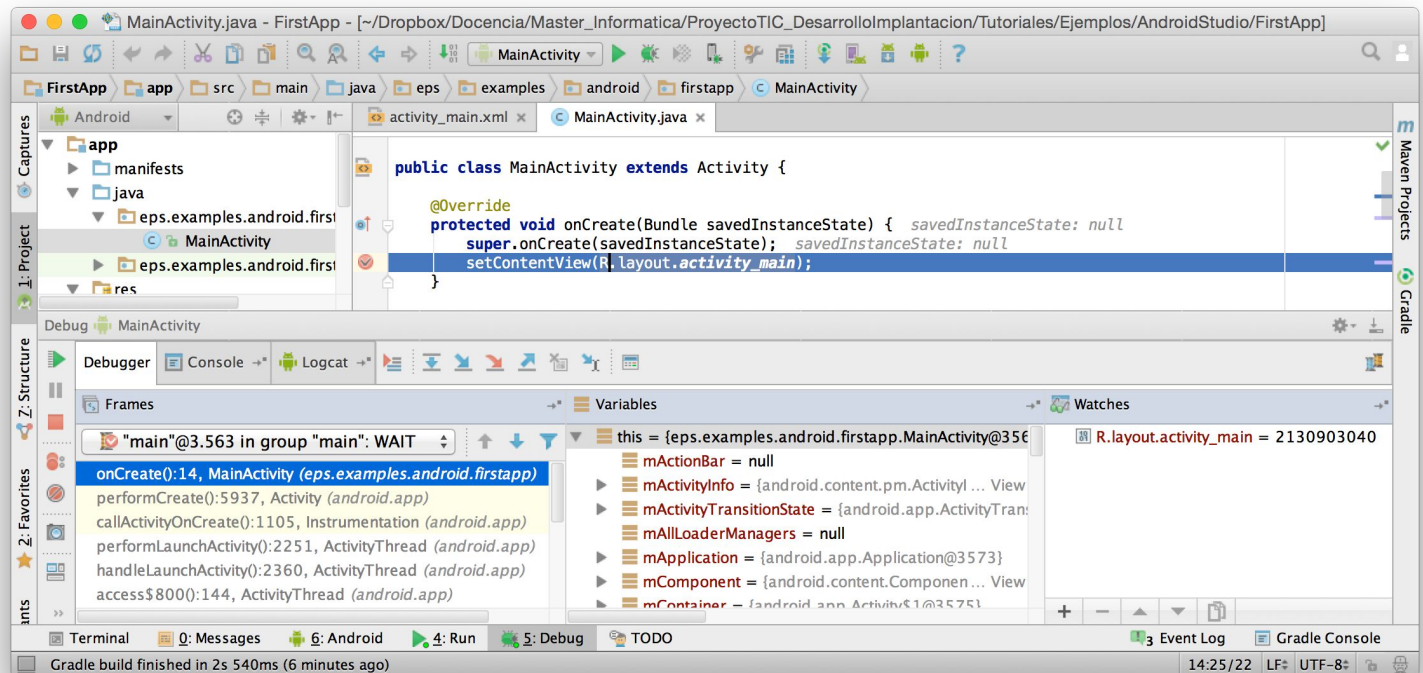
- Select a device to debug your app on.
- View the system log.
- Set breakpoints in your code.
- Examine variables and evaluate expressions at run time.
- Run the debugging tools from the Android SDK.
- Capture screenshots and videos of your app.



DEBUGGING



- ✧ Android Studio opens the Debug tool when you debug your app.
- ✧ It shows threads and variables in the Debugger tab, the device status in the Console tab, and the system log in the Logcat tab.



ANDROID DEVICE MONITOR



- ✧ A tool that lets developers manage processes on an emulator or device and assists in debugging.
- ✧ Functions:
 - Killing processes
 - Selecting a specific process to debug
 - Generating trace data
 - Viewing heap and thread information
 - Taking screenshots of the emulator or device
 - Manage file system
 - Change some parameters

HELLO ANDROID

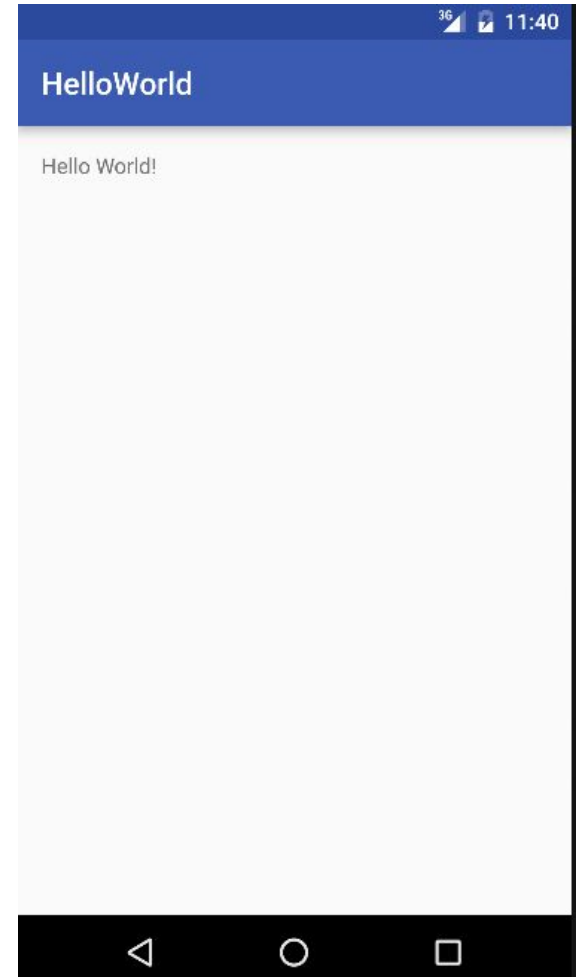


GOALS

- ✧ Create a very simple application
- ✧ Run it on the emulator
- ✧ Examine its structure

✧ Building Your First App:

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



HELLO ANDROID



NEW AVD

- ✧ An AVD defines the system image and device settings used by the emulator.
- ✧ Before you can launch the emulator, you must create an Android Virtual Device (AVD):
 - Choose a device definition, e.g. Nexus 5X.
 - Select a system image, e.g. KitKat you want to run on the emulator.
 - Type the name of the AVD.

HELLO ANDROID



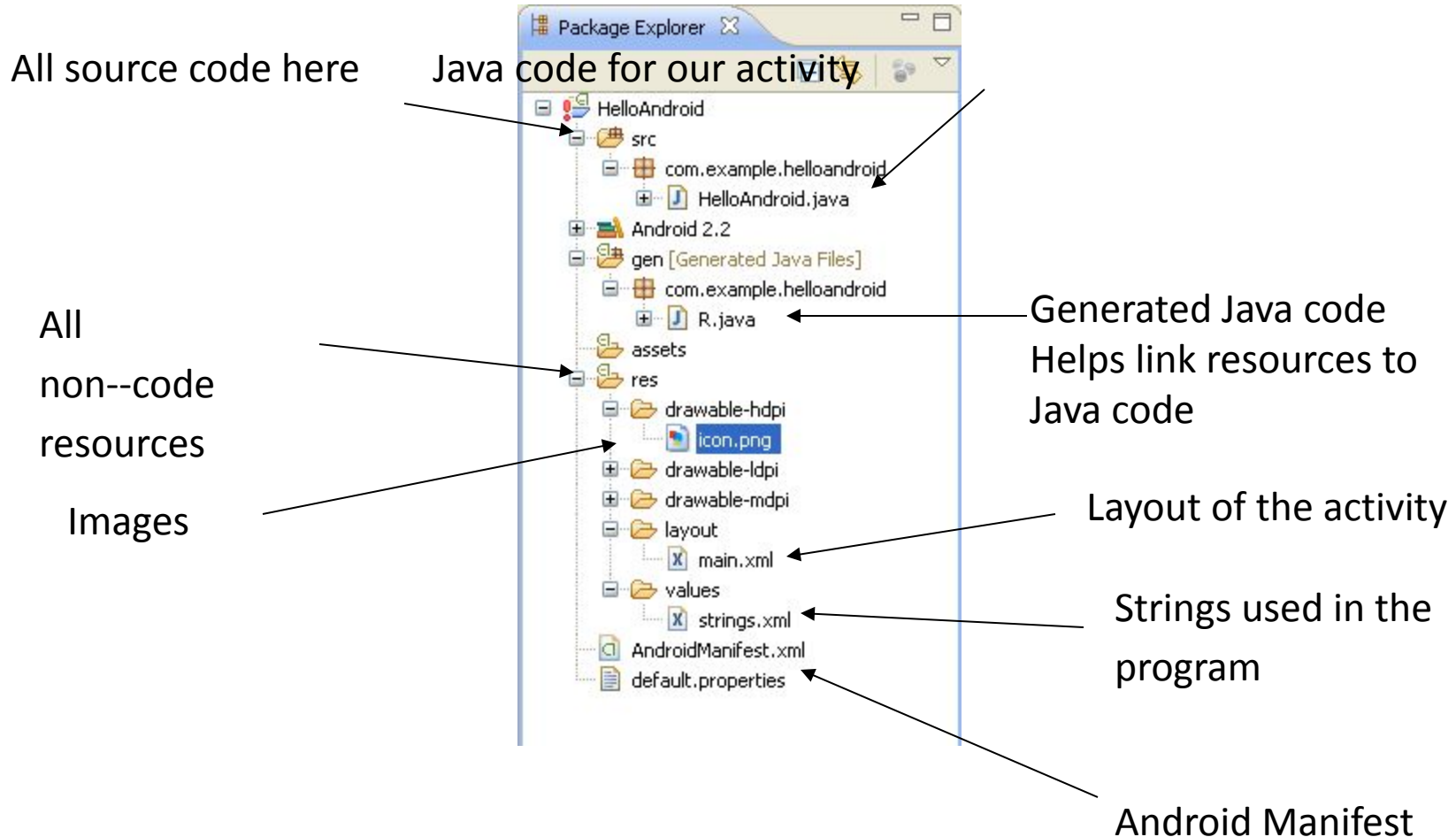
PROJECT CREATION

| Feature | Description |
|------------------|--|
| Application Name | The name of the directory or folder you want to contain the project |
| Company Domain | Is used by Android Studio to generate a package name |
| Package Name | The package namespace (following the same rules as for packages in the Java language) that all source code will reside under |
| Target Device | The Minimum Required SDK is the earliest version of Android that your app supports, which is indicated by the API level. |
| Activity Name | The name for the class stub that will be generated by the plug-in |
| Layout | Defines the layout of the activity |

HELLO ANDROID



PACKAGE CONTENT



HELLO ANDROID



MAIN FEATURES

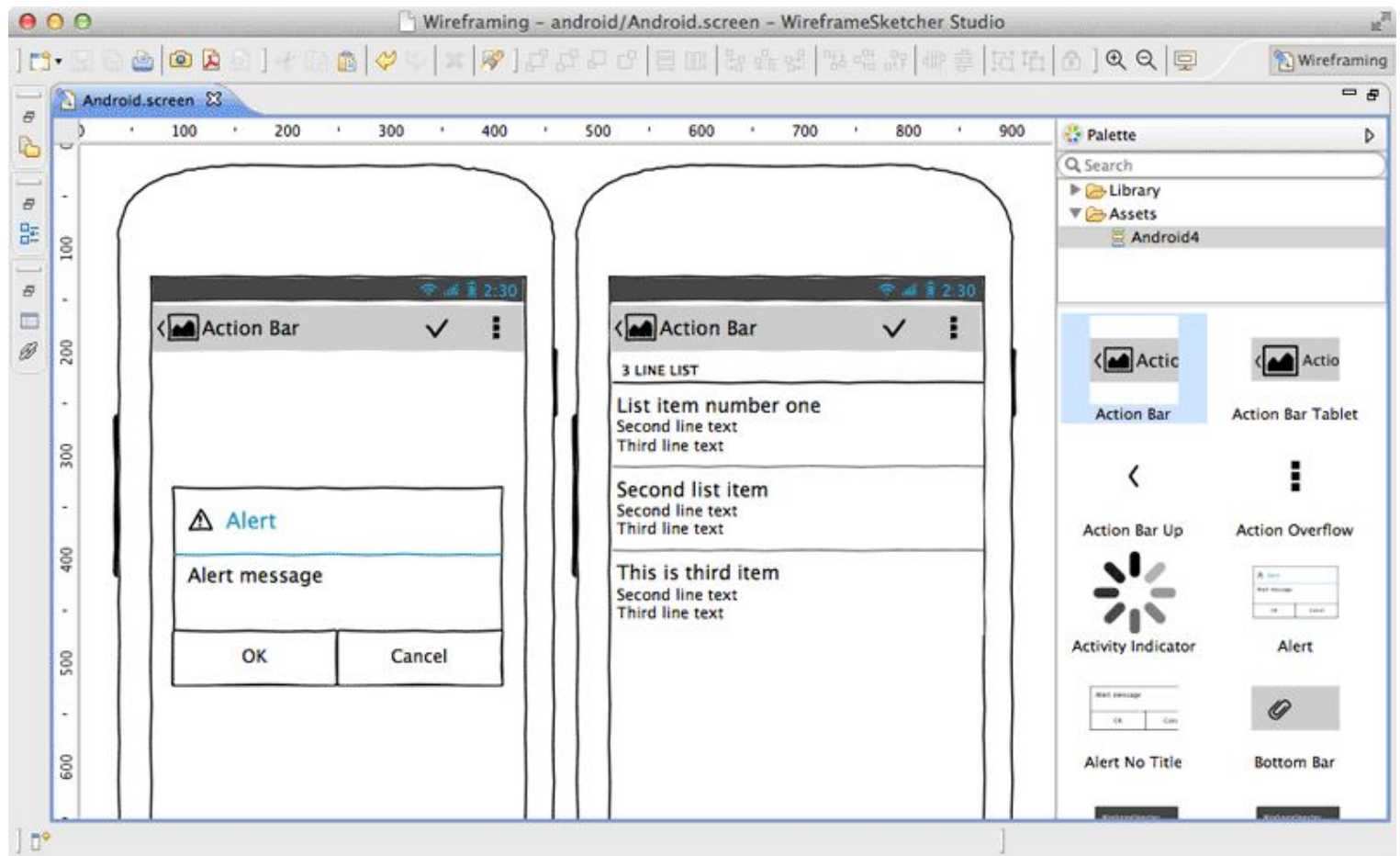
- ✧ Activity
 - ✧ setContentView
- ✧ XML Layout (res/layout)
 - ✧ RelativeLayout
 - ✧ TextView
- ✧ Strings file (res/values)
- ✧ Android Manifest
- ✧ R.java *

*
project/app/build/generated/source/r/debug/package_name/R.java

MOCKUP & PROTOTYPE



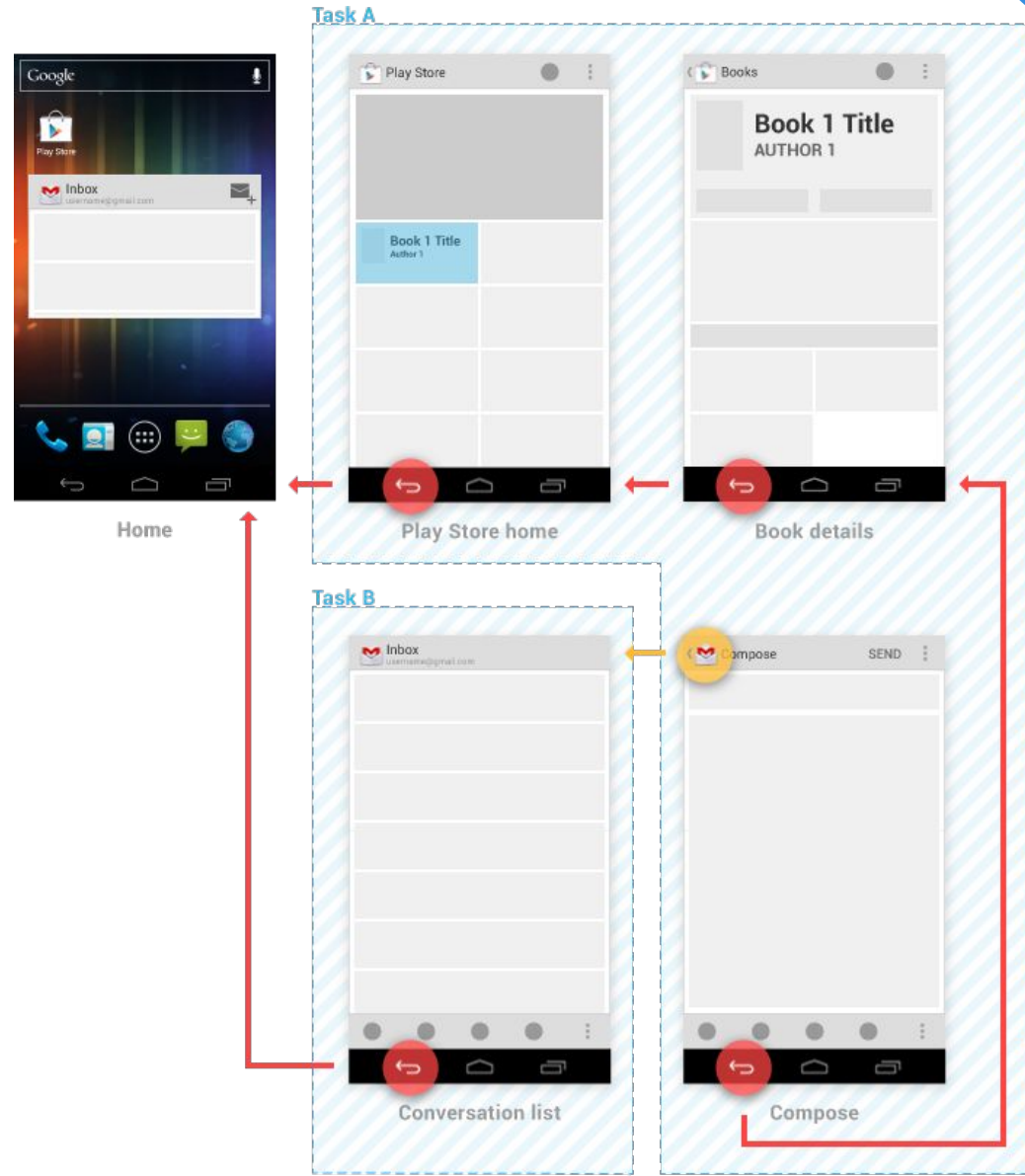
MAIN SCREENS DESIGN



MOCKUP & PROTOTYPE



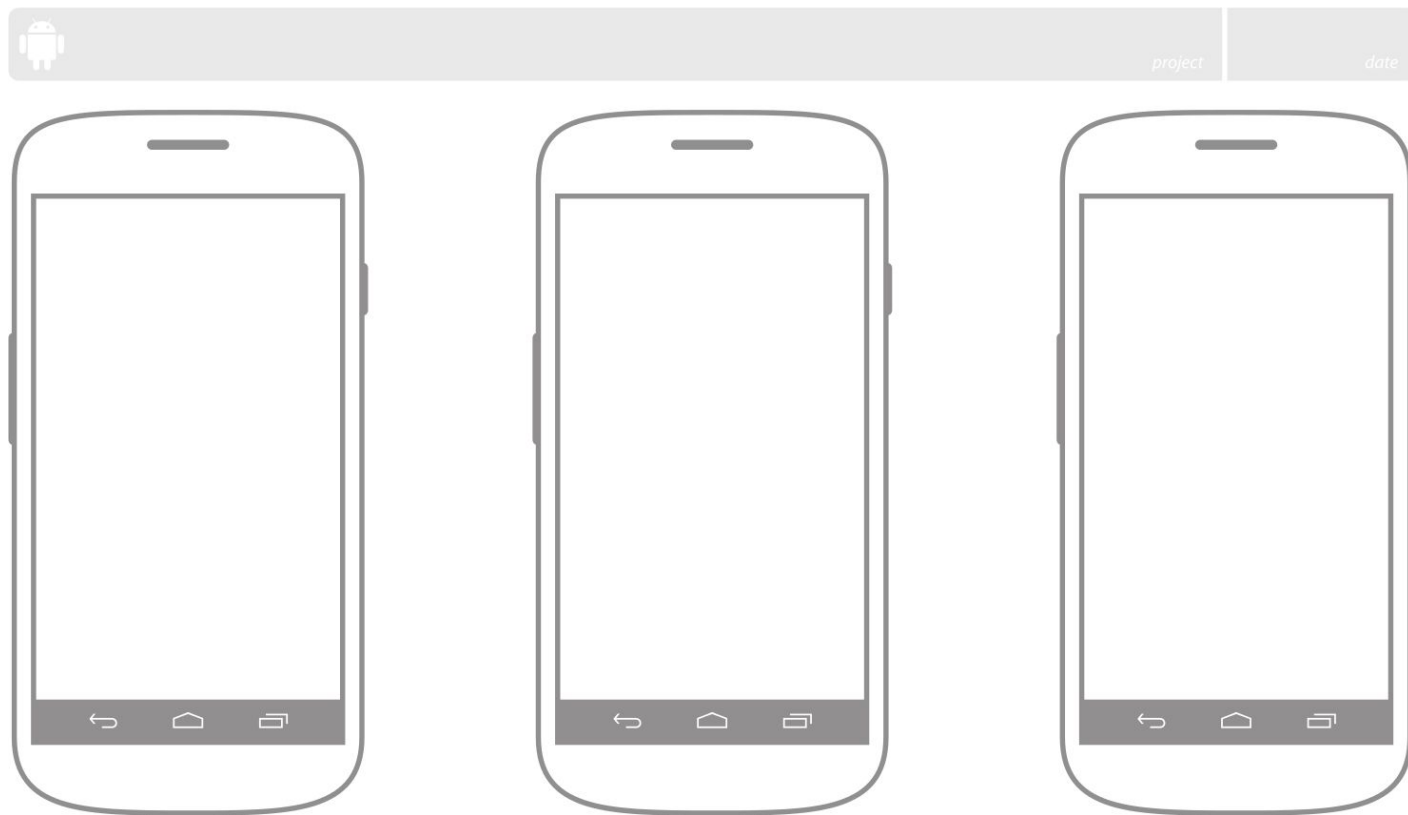
NAVIGATION (entry points)



MOCKUP & PROTOTYPE



HANDMADE



<https://www.smashingmagazine.com/2012/09/free-download-ux-sketching-wireframing-templates-mobile/>

MOCKUP & PROTOTYPE



TOOLS

- ✧ Interface sketch.
<http://www.interfacesketch.com/>
- ✧ Cacao. <https://cacao.com/lang/es/>
- ✧ Marvel. <https://marvelapp.com/>
- ✧ Invision. <https://www.invisionapp.com/>
- ✧ Balsamiq mockups.
<https://balsamiq.com/products/mockups/>

ANDROID DEVELOPMENT



<https://developer.android.com/>