SIT1402 - Mobile Application Development

Unit- I - Introduction and UI interface

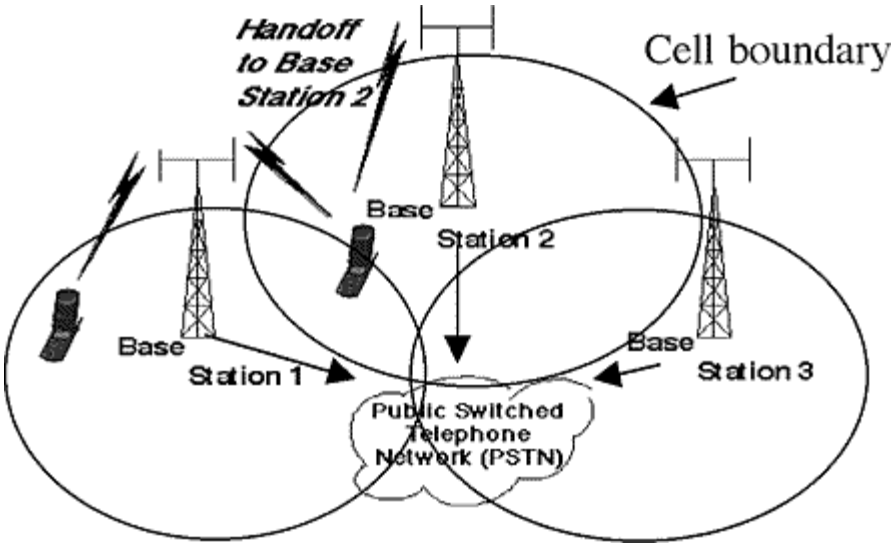
- 1. Introduction to mobile technologies
- 2. Mobile operating systems
- 3. Mobile devices – pros and cons
- 4. Introduction to Android, Versions,Features
- 5. Android architecture
- 6. UI Layouts
- 7. UI Controls / Widgets
- 8. Event handling
- 9. Required Tools- Eclipse, ADT, AVD
- 10. Application structure
- 11. Android manifest file
- 12. Android design philosophy
- 13. Creating andriod applications

1. Mobile Networks / Technologies

- GSM
- GPRS
- EDGE
- 1G, 2G, 3G, 4G, 5G
- IEEE 802.11
- Infrared
- Bluetooth

Cellular Network

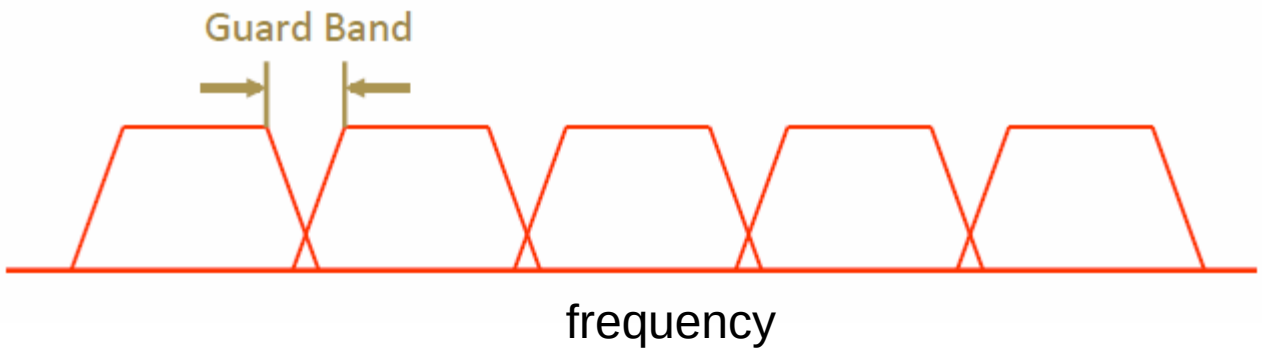
- Base stations transmit to and receive from mobiles at the assigned spectrum
  - Multiple base stations use the same spectrum (spectral reuse)
- The service area of each base station is called a cell
- Each mobile terminal is typically served by the 'closest' base stations
  - Handoff when terminals move



Cellular Network Generations

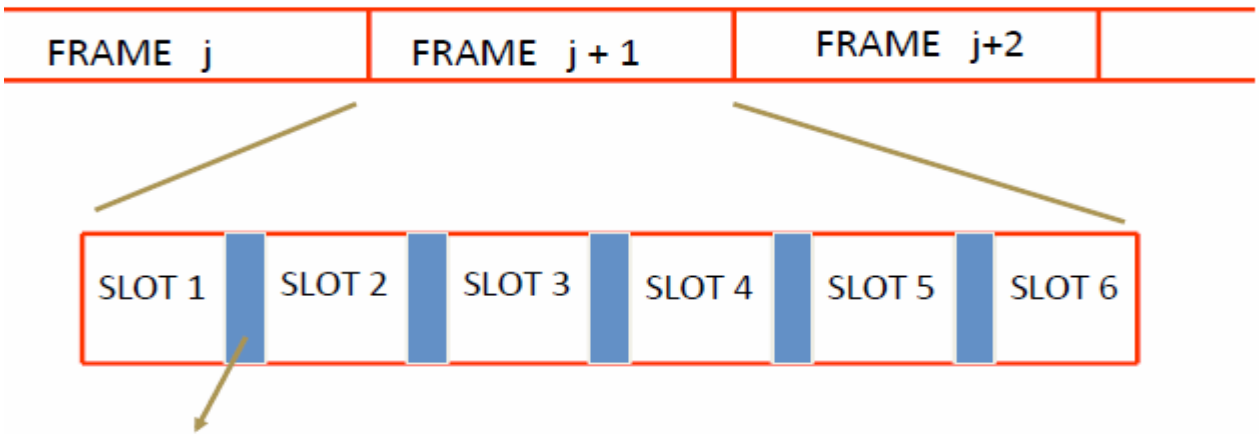
- It is useful to think of cellular Network/telephony in terms of generations:
  - 0G: Briefcase-size mobile radio telephones
  - 1G: *Analog* cellular telephony
  - 2G: *Digital* cellular telephony
  - 3G: *High-speed* digital cellular telephony (including *video telephony*)
  - 4G: IP-based “anytime, anywhere” voice, data, and multimedia telephony at *faster* data rates than 3G (to be deployed in 2012–2015)

Frequency Division Multiple Access



- Each mobile is assigned a separate frequency channel for the duration of the call
- Sufficient guard band is required to prevent adjacent channel interference
- Usually, mobile terminals will have one downlink frequency band and one uplink frequency band
- Different cellular network protocols use different frequencies
- Frequency is a precious and scare resource. We are running out of it.

Time Division Multiple Access



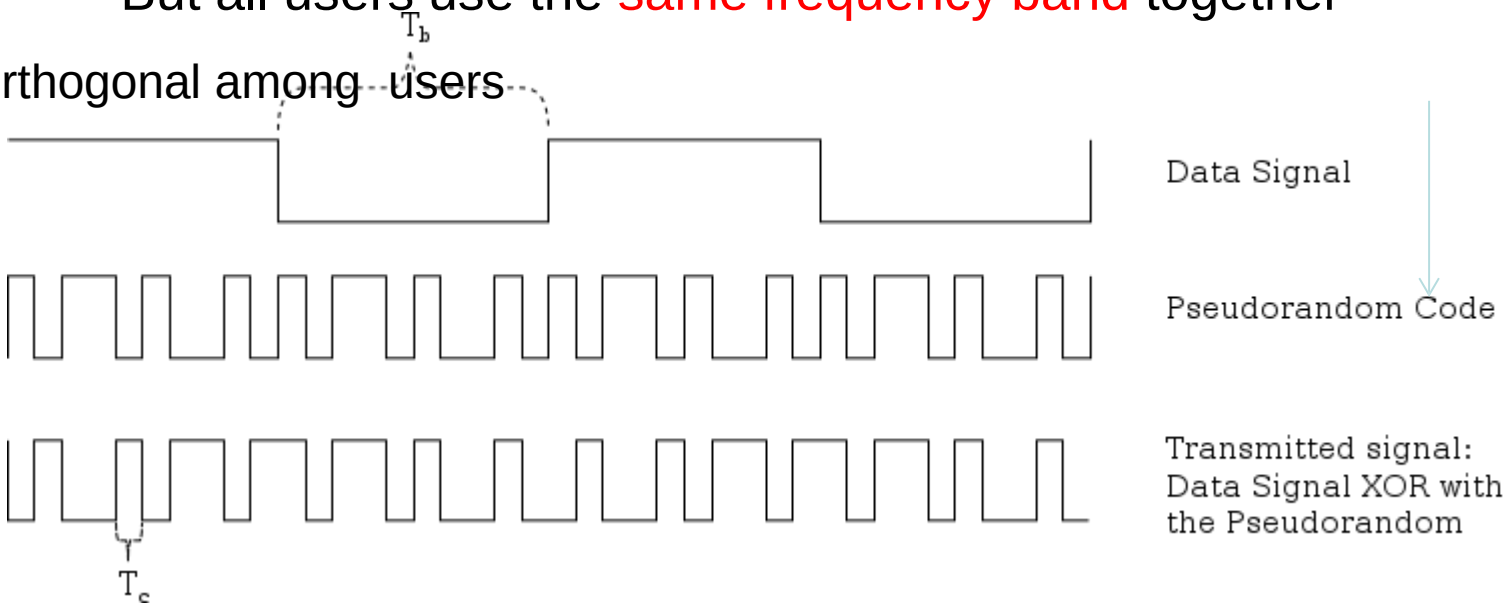
Guard time – signal transmitted by mobile terminals at different locations do no arrive at the base station at the same time

- Time is divided into slots and only one mobile terminal transmits during each slot
  - Like during the lecture, only one can talk, but others may take the floor in turn
- Each user is given a specific slot. No competition in cellular network
  - Unlike Carrier Sensing Multiple Access (CSMA) in WiFi

Code Division Multiple Access

- Use of orthogonal codes to separate different transmissions
- Each symbol of bit is transmitted as a larger number of bits using the user specific code – Spreading
  - Bandwidth occupied by the signal is much larger than the information transmission rate
  - But all users use the same frequency band together

Orthogonal among users



1 GENERATION

- First generation cellular networks
- Radio signals = analog
- Technologies - AMPS (Advanced Mobile Phone System)
- First Blackberry (850)

1G TECHNOLOGY

- ***1G refers to the first generation of wireless telephone technology, mobile telecommunications which was first introduced in 1980s and completed in early 1990s.***
- ***It's Speed was upto 2.4kbps.***
- ***It allows the voice calls in 1 country.***
- ***1G network use Analog Signal.***
- ***AMPS was first launched in USA in 1G mobile systems.***

DRAWBACKS OF 1G

- ***Poor Voice Quality***
- ***Poor Battery Life***
- ***Large Phone Size***
- ***No Security***
- ***Limited Capacity***
- ***Poor Handoff Reliability***

1G Wireless System

2G (GSM and GPRS Networks)

- 2G carriers continued to improve transmission quality and coverage paging, faxes, text messages and voicemail.
- 2.5G uses GPRS(General Packet Radio Services), which delivers packet-switched capabilities to existing GSM networks.

2G TECHNOLOGY

- 2G technology refers to the 2nd generation which is based on GSM.
- It was launched in Finland in the year 1991.
- 2G network use digital signals.
- It's data speed was upto 64kbps.

Features Includes:

- ✓ It enables services such as text messages, picture messages and MMS (multi media message).
- ✓ It provides better quality and capacity .

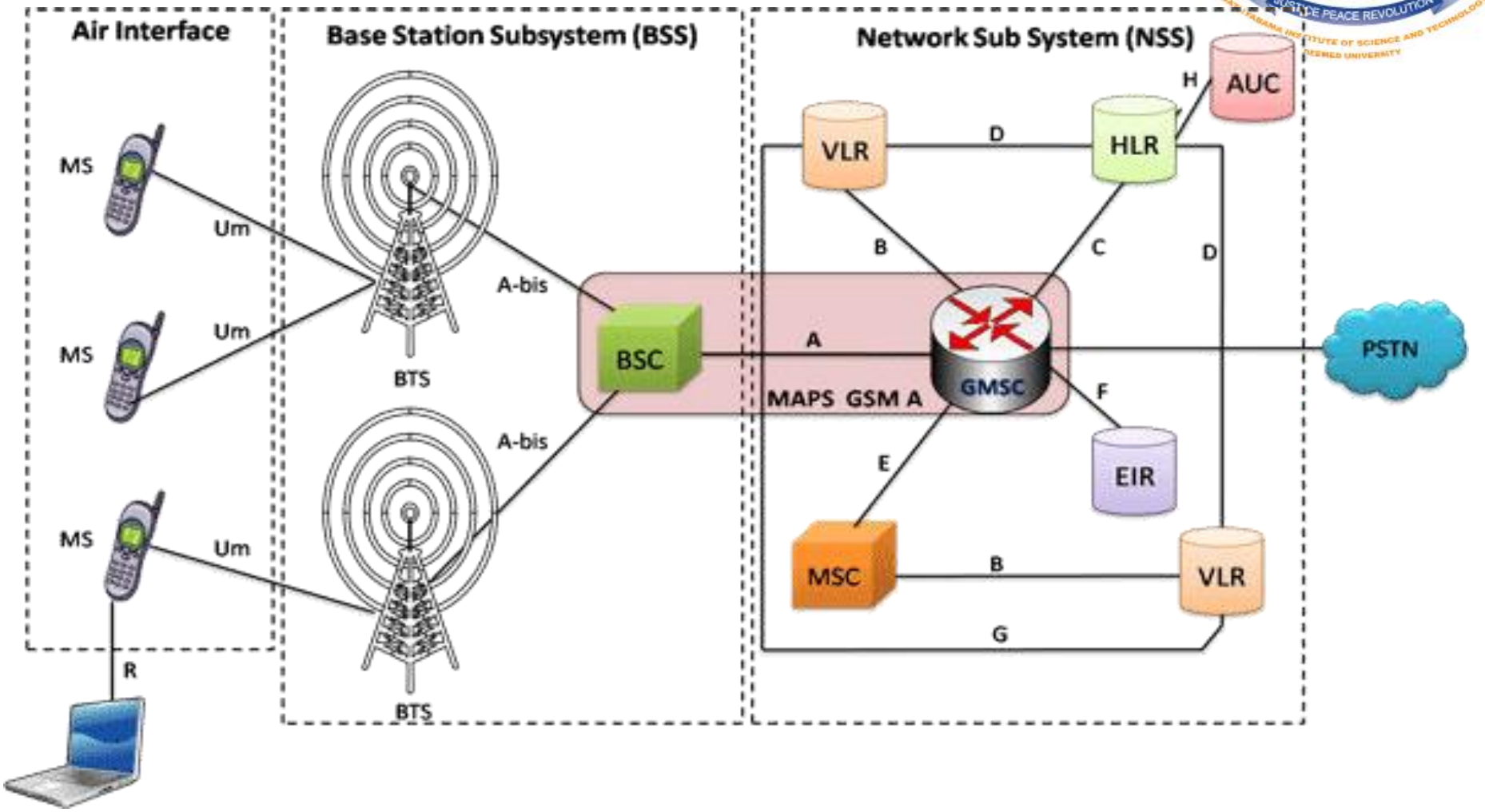
DRAWBACKS OF 2G

- *2G requires strong digital signals to help mobile phones work. If there is no network coverage in any specific area , digital signals would weak.*

- *These systems are unable to handle complex data such as Videos.*

2G Wireless System

GSM Architecture



FEATURES OF 3G TECHNOLOGY

- ✓ *Providing Faster Communication*
- ✓ *Send/Receive Large Email Messages*
- ✓ *High Speed Web / More Security*

Video Conferencing / 3D Gaming

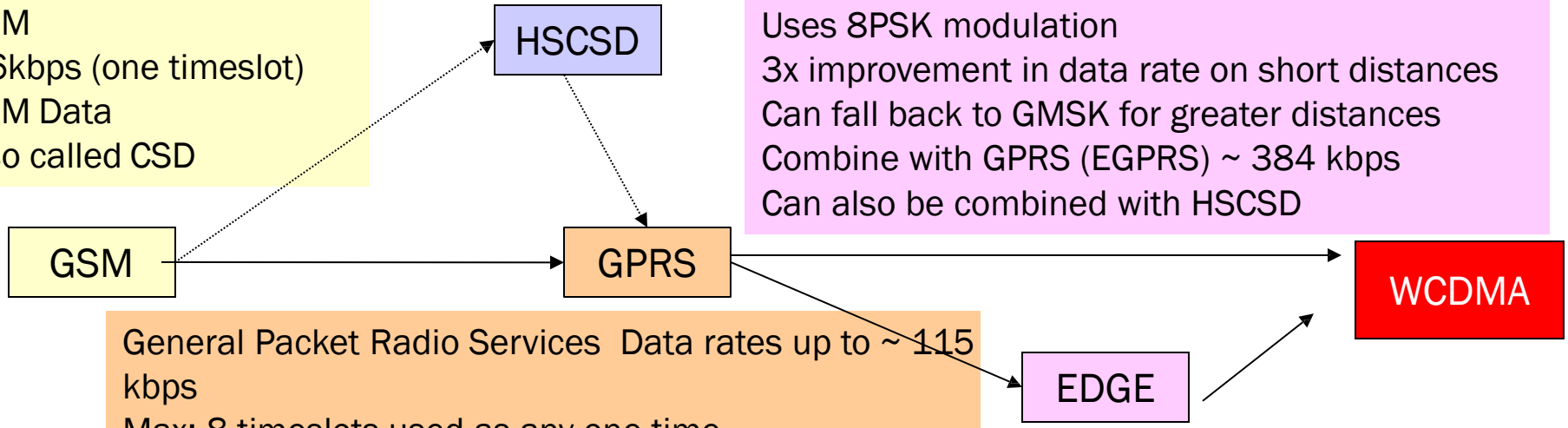
- ✓ *TV Streaming/ Mobile TV/ Phone Calls*
- ✓ *Large Capacities and Broadband Capabilities*
- ✓ *11 sec – 1.5 min. time to download a 3 min Mp3 song.*

GSM Evolution to 3G

High Speed Circuit Switched Data
Dedicate up to 4 timeslots for data connection ~ 50 kbps
Good for real-time applications c.w. GPRS
Inefficient -> ties up resources, even when nothing sent
Not as popular as GPRS (many skipping HSCSD)

GSM
9.6kbps (one timeslot)
GSM Data
Also called CSD

Enhanced Data Rates for Global Evolution
Uses 8PSK modulation
3x improvement in data rate on short distances
Can fall back to GMSK for greater distances
Combine with GPRS (EGPRS) ~ 384 kbps
Can also be combined with HSCSD



General Packet Radio Services Data rates up to ~ 115 kbps
Max: 8 timeslots used as any one time
Packet switched; resources not tied up all the time
Contention based. Efficient, but variable delays GSM / GPRS core network re-used by WCDMA (3G)

GSM Evolution to 3G (con...)

- W-CDMA (Wide Band Code Division Multiple Access) technology.
- It also used for services like Wireless Application Protocol (WAP) access, Multimedia Messaging Service (MMS) or Short Message Service (SMS)
- Internet communication through the excess to World Wide Web and E-mail.

DRAWBACKS OF 3G TECHNOLOGY

- *Expensive fees for 3G Licenses Services*
- *It was challenge to build the infrastructure for 3G*
- *High Bandwidth Requirement*
- *Expensive 3G Phones.*
- *Large Cell Phones*

4G TECHNOLOGY (Anytime ,Anywhere)

- *4G technology refer to or short name of fourth Generation which was started from late 2000s.*
- *Capable of providing 100Mbps – 1Gbps speed.*
- *One of the basic term used to describe 4G is MAGIC.*

MAGIC:

- *Mobile Multimedia*
- *Anytime Anywhere*
- *Global Mobility Support*
- *Integrated Wireless Solution*
- *Customized Personal Services*

Also known as Mobile Broadband Everywhere.

4G (Anytime, Anywhere)

- ***The next generations of wireless technology that promises higher data rates and expanded multimedia services.***
- ***Capable to provide speed 100Mbps-1Gbps.***
- ***High QOS and High Security***
- ***Provide any kind of service at any time as per user requirements, anywhere.***

<u>Features Include:</u>

- ***More Security***
- ***High Speed***
- ***High Capacity***
- ***Low Cost Per-bit etc.***

5G TECHNOLOGY

- ***5G technology refer to short name of fifth Generation which was started from late 2010s.***

- ***Complete wireless communication with almost no limitations.***

- ***It is highly supportable to WWWW (Wireless World Wide Web).***

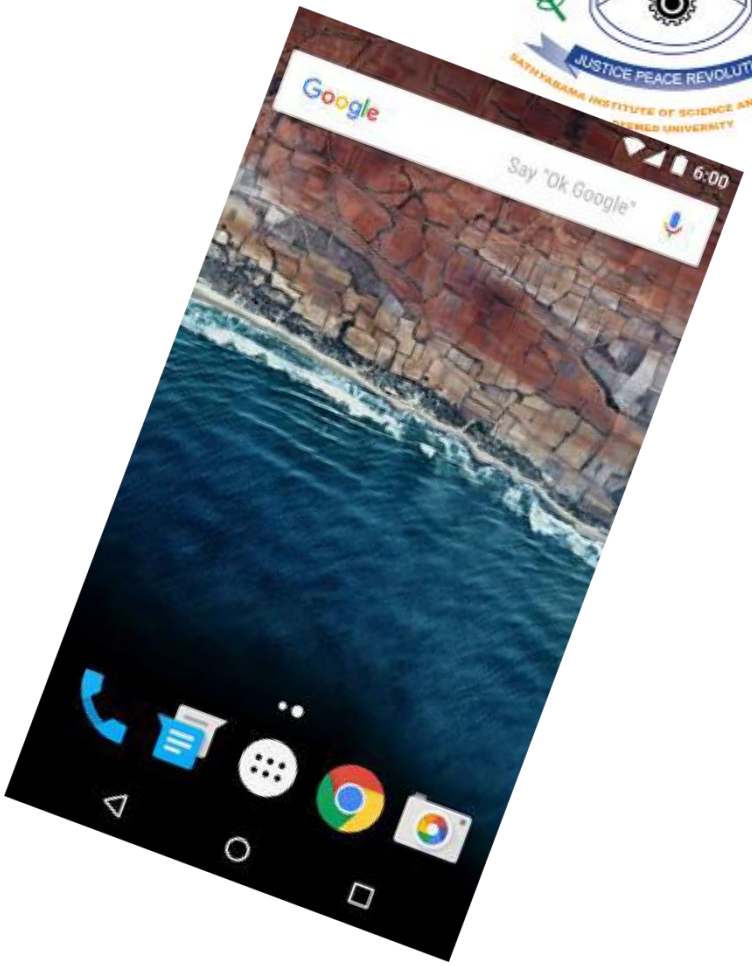
2. Mobile Operating Systems

What is Mobile OS?

- A Mobile OS is a very basic and essential software to operate a Mobile System.
- A Mobile OS is a software platform which is designed specially for mobile to handle the devices like Smart phone, Tablet, PDA with lot of features and facilities.

Google™

android



Android

-Android is a mobile operating system (OS) currently developed by Google, based on the Linux kernel and designed primarily for touchscreen mobile devices such as smartphones and tablets.

-It was developed by Google, Open Handset Alliance, Android Open Source Project, Android Inc.

-Source model, open source

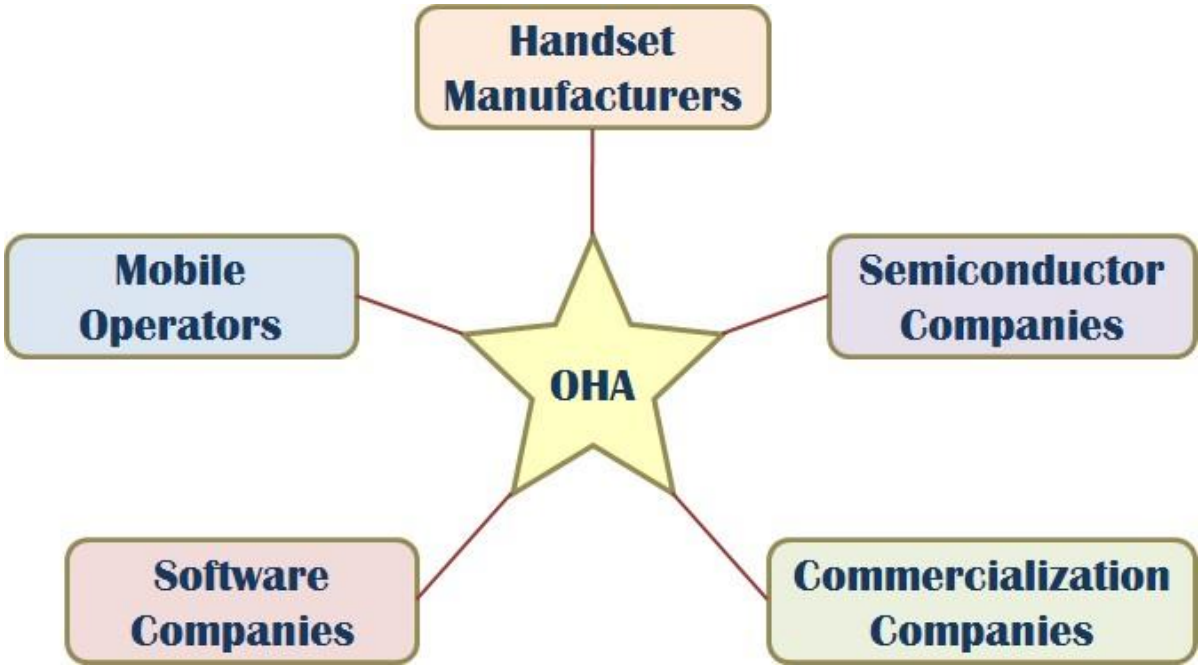
-Written in C (core), C++, and Java (UI)

-OS family, Unix

android

OHA (Open Handset Alliance)

- A business alliance consisting of 47+ companies to develop open standards for mobile devices



Apple iOS


O2-UK
20:51
100%
Messages
Wednesday
21
Calendar
Photos
Camera
Weather
Clock
Maps
Videos
Wallet
Notes
Reminders
Stocks
iTunes Store
App Store
iBooks
News
Health
Settings
Phone
Safari
Mail
Music

iOS

-iOS (originally iPhone OS) is a mobile operating system created and developed by Apple Inc. and distributed exclusively for Apple hardware. It is the operating system that presently powers many of the company's mobile devices, including the iPhone, iPad, and iPod touch.

-It was developed by Apple Inc. June 29, 2007

-Source model, closed source

-Written in, C,C++, Objective-C, and Swift

-OS family, Unix

iOS


Me
Paula Araujo
3:31
10
Email
Messaging
1
Diego Montes
Missed call at 3:24 PM
Phone
Internet Explorer
5
Office
Camera
Instagram BETA
4
Rita
Calendar
Sat 30

Windows

Windows Mobile

-Windows Mobile is a mobile operating system developed by Microsoft for smart phones and Pocket PC's

-It was first launched in October 2010 with Windows Phone 7

-Currently maintained with Micosoft Corporation

-Written in C, C++

-OS Family, Microsoft Windows

Windows Phone


Thu, May 21
12:21 PM
3G
Network
3
All
Messages
SMS
Contacts
27
Calendar
Browser
Maps
Clock
Camera

Blackberry

Blackberry

- BlackBerry OS is a proprietary mobile operating system developed by BlackBerry Ltd for its BlackBerry line of smartphone handheld devices.
-It was developed by BlackBerry Ltd on January 19, 1999
-Source model is closed source
-Written in, C++ and Java
-OS family, Mobile Operating Systems

Symbian

Symbian

-Symbian is a mobile operating system (OS) and computing platform designed for smart phones
-Symbian was originally developed as a closed-source OS for PDAs in 1998 by Symbian Ltd.
-Currently maintained by Accenture on behalf of

Nokia (historically Symbian Ltd. and Symbian Foundation)
-Written in C++
-OS Family RTOS

Bada


01:12
01:12
16°C
Johannesburg
Partly Cloudy with Th...
Thu.
12/01
Social Hub
ChatON
Google
Samsung Apps
Keypad
Contacts
Messages
Menu

BADA

-Bada is an operating system for mobile devices such as smartphones and tablet computers.

-It was developed by Samsung Electronics on April 2010.

-Source model is, Mixed: proprietary and open source

-Written in C++

-OS Family, POSIX (Portable Operating System Interface for Unix)

Why Mobile App Development?

- Mobile platform is the platform of the future world
- Job market is hot
  - Market for mobile software surges from $4.1 billion in 2009 to $17.5 billion by 2012
  - In 2010, www.dice.com survey: 72% of recruiters looking for iPhone app developers, 60% for Android
  - Dice.com: mobile app developers made $85,000 in 2010 and salaries expected to rise
  - According to 2016, 79% of the organizations surveyed plan to increase spending on mobile development by 36%
- Students (and faculty!) are naturally interested!

Types of Mobile Appications

- What are they?
  - Any application that runs on a mobile device
- Types
  - Web Apps
  - Native Apps
  - Hybrid Apps

Types of Mobile...(con...)

- Native Apps
  - It is live on the device and are accessed through icons on the device home screen.
  - They are installed through an application store (such as Google Play or Apple's App Store).
  - They are developed specifically for one platform, and can take full advantage of all the device features — they can use the camera, the GPS, the accelerometer, the compass, the list of contacts, and so on.

Types of Mobile...(con...)

- Web Apps
  - They are not real applications; they are really websites that, in many ways, *look and feel* like native applications, but are

not *implemented* as such.

  - They are run by a browser and typically written in HTML5
  - Web apps became really popular when HTML5 came around and people realized that they can obtain native-like functionality in the browser.

Types of Mobile...(con...)

- Hybrid apps
  - Hybrid apps are part native apps, part web apps.
  - Like native apps, they live in an app store and can take advantage of the many device features available.
  - Like web apps, they rely on HTML being rendered in a browser, with the caveat that the browser is embedded within the app.

3. Mobile Devices: Advantages (as compared to fixed devices)

- Always with the user
- Typically have Internet access
- Typically GPS enabled
- Typically have accelerometer & compass
- Mostly have cameras & microphones
- Many apps are free or low-cost and etc...

Mobile Devices: Limitations

- Limited memory
- Limited processing power
- Different technologies and standards
- Limited or awkward input: soft keyboard, phone keypad, touch screen, or stylus
- Small screens
- Limited and slow network access
- Slow hardware
- Limited battery life
- Limited web browser functionality
- Often inconsistent platforms across devices and etc...

Android

Mobile Application Development

Prerequisite

- Good knowledge of JAVA language
- Familiarity with Eclipse IDE

* All the above is not mandatory

4. Introduction to Android

- Open software platform for mobile development
- A complete stack – OS, Middleware, Applications
- An Open Handset Alliance (OHA) project
- Powered by Linux operating system
- Fast application development in Java
- Open source under the Apache 2 license

What is Android?

- Android is a software stack for mobile devices that includes an operating system, middleware and key applications.

Developed by

- Andy Rubin (co-founder of Danger),
- Rich Miner (co-founder of Wildfire Communications, Inc.),
- Nick Sears (once VP at T-Mobile)
- Chris White (headed design and interface development at WebTV).

History of Android

- 1) Initially, Andy Rubin founded Android Incorporation in Palo Alto, California, United States in October, 2003.
- 2) In 17th August 2005, Google acquired Android Incorporation. Since then, it is in the subsidiary of Google Incorporation.
- 3) The key employees of Android Incorporation are Andy Rubin, Rich Miner, Chris White and Nick Sears.

History of Android (con...)

- 4) Originally intended for camera but shifted to smart phones later because of low market for camera only.
- 5) Android is the nick name of Andy Rubin given by coworkers because of his love to robots.
- 6) In 2007, Google announces the development of Android OS.
- 7) In 2008, HTC launched the first android mobile.

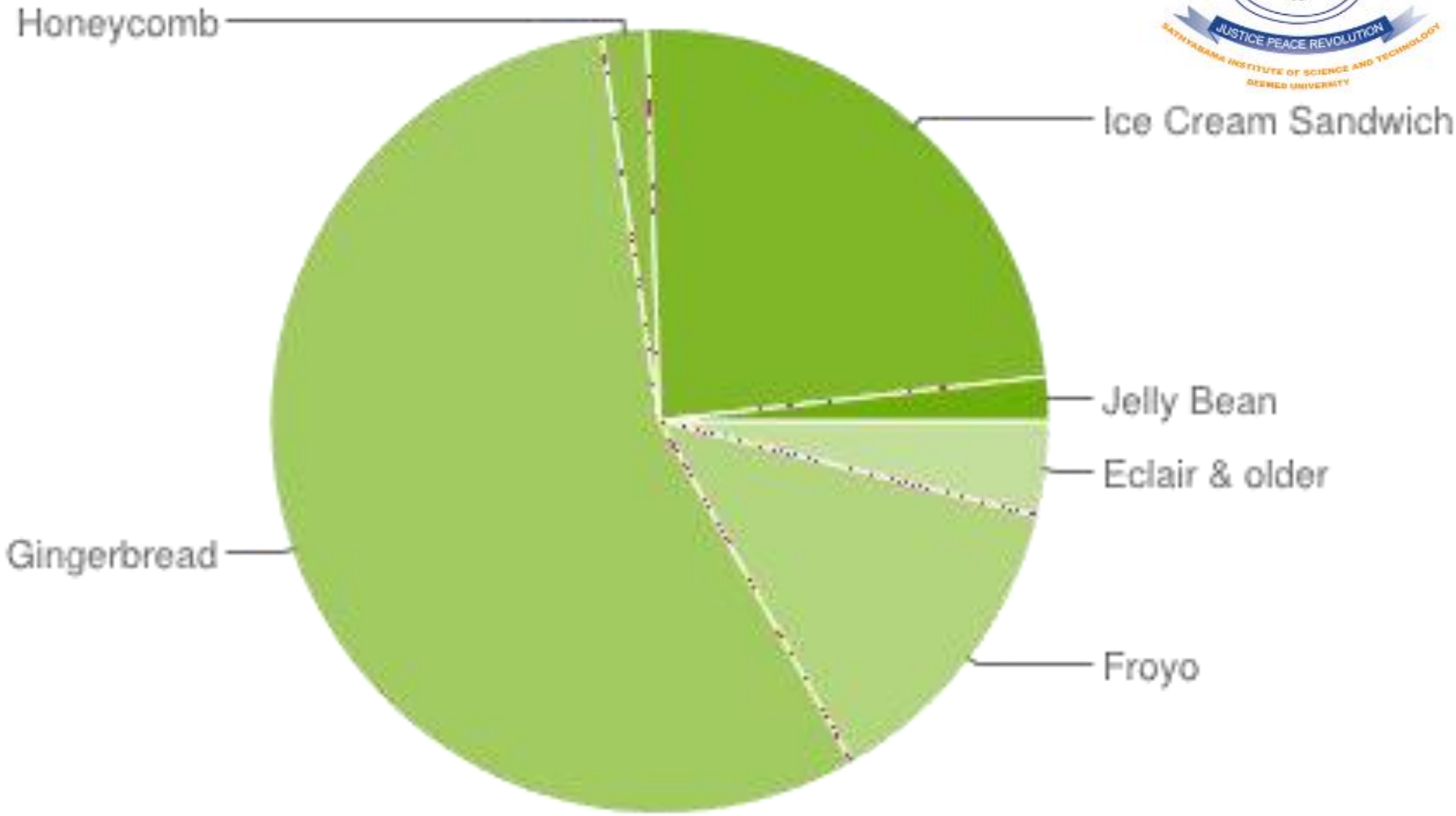
History of Android (con…)

- The code names of android ranges from A to N currently, such as
  - 1.0 Astro (some times says no code name)
  - 1.1 Bender (Some times say “Petit four”)
  - 1.5 Cupcake
  - 1.6 Donut
  - 2.x Eclair
  - 2.2 Froyo
  - 2.3.x Gingerbread
  - 3.x.x Honeycomb
  - 4.0.x Ice Cream Sandwitch
  - 4.1.x, 4.2.x and 4.3.x Jelly Bean
  - 4.4.x KitKat and
  - 5.x.x Lollipop
  - 6.0 MarshMallow
  - N (“A Few Weeks”)

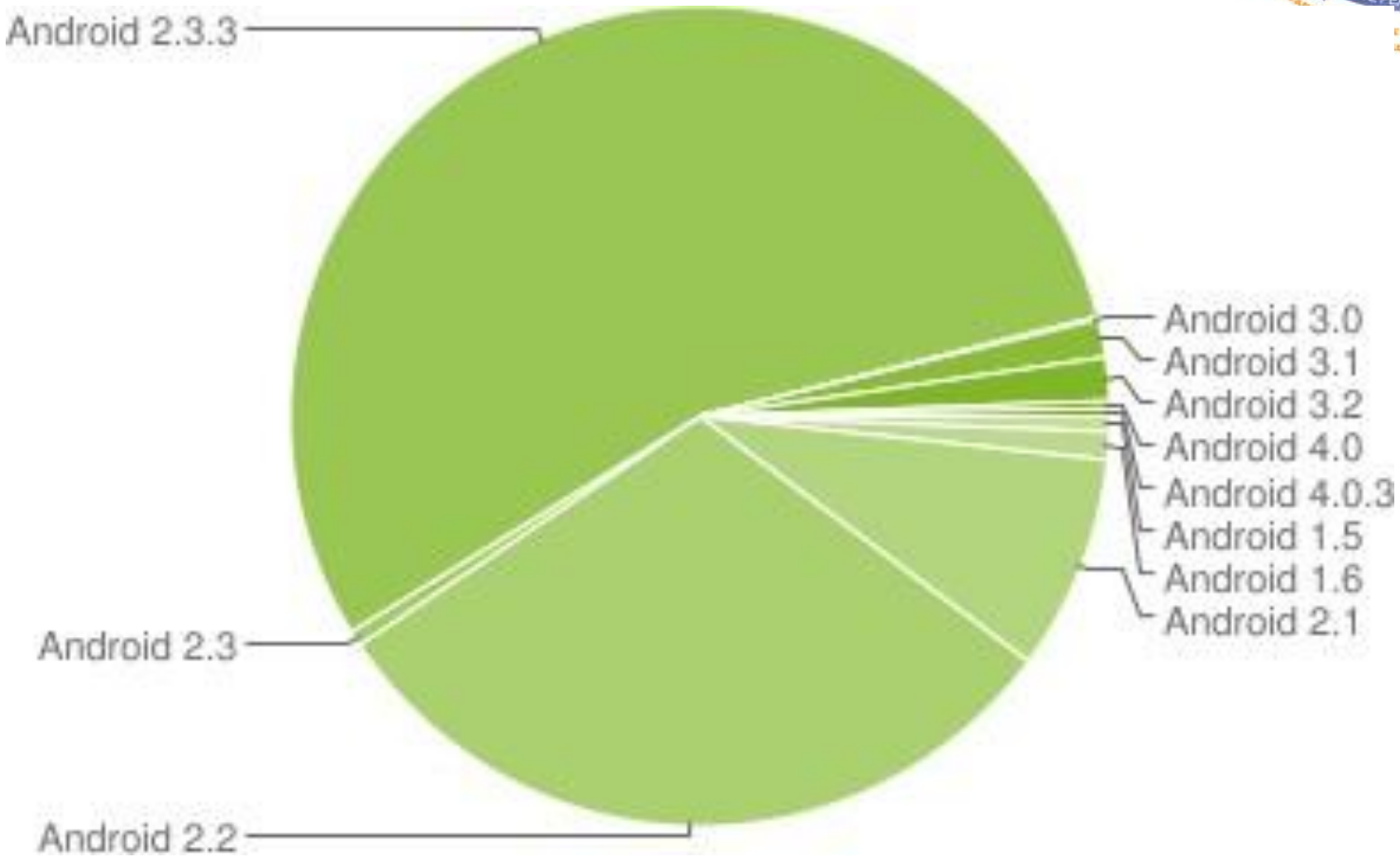
History of Android(con...)

- Let's understand the android history in a sequence.

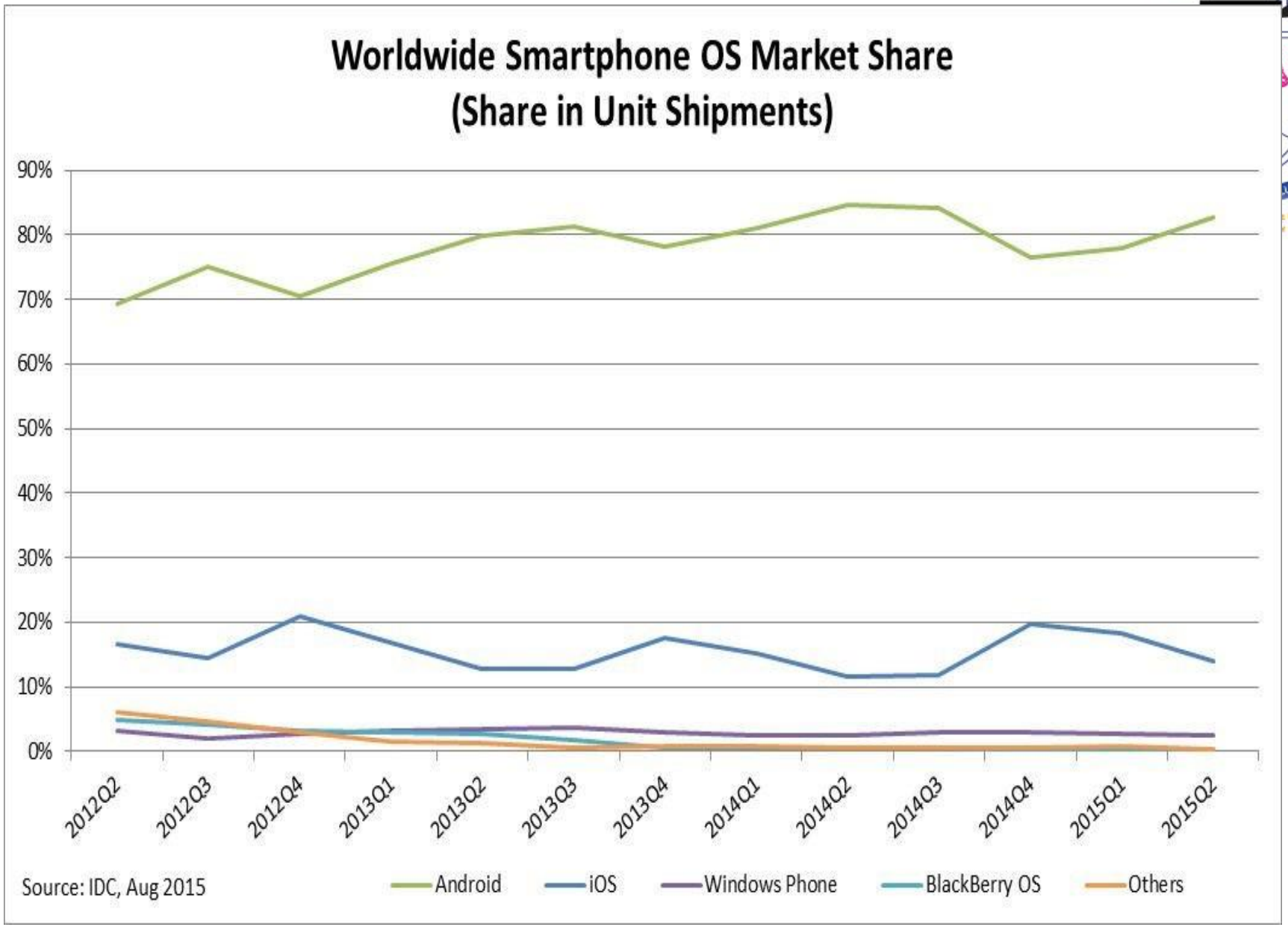
Android Survey



Distribution of Devices



Data collected during a 14-day period ending on January 3, 2012


Worldwide Smartphone OS Market Share
(Share in Unit Shipments)
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
2012Q2
2012Q3
2012Q4
2013Q1
2013Q2
2013Q3
2013Q4
2014Q1
2014Q2
2014Q3
2014Q4
2015Q1
2015Q2
Source: IDC, Aug 2015
Android
iOS
Windows Phone
BlackBerry OS
Others

World wide Mobile OS market share

Period	Android	iOS	Windows Phone	BlackBerry OS	Others
2015Q2	82.8%	13.9%	2.6%	0.3%	0.4%
2014Q2	84.8%	11.6%	2.5%	0.5%	0.7%
2013Q2	79.8%	12.9%	3.4%	2.8%	1.2%
2012Q2	69.3%	16.6%	3.1%	4.9%	6.1%

Source: IDC, Aug 2015

Android to lead the smartphone market with 49% market share by 2012 » Mobile OS Sales 2010-2015 – Gartner Survey

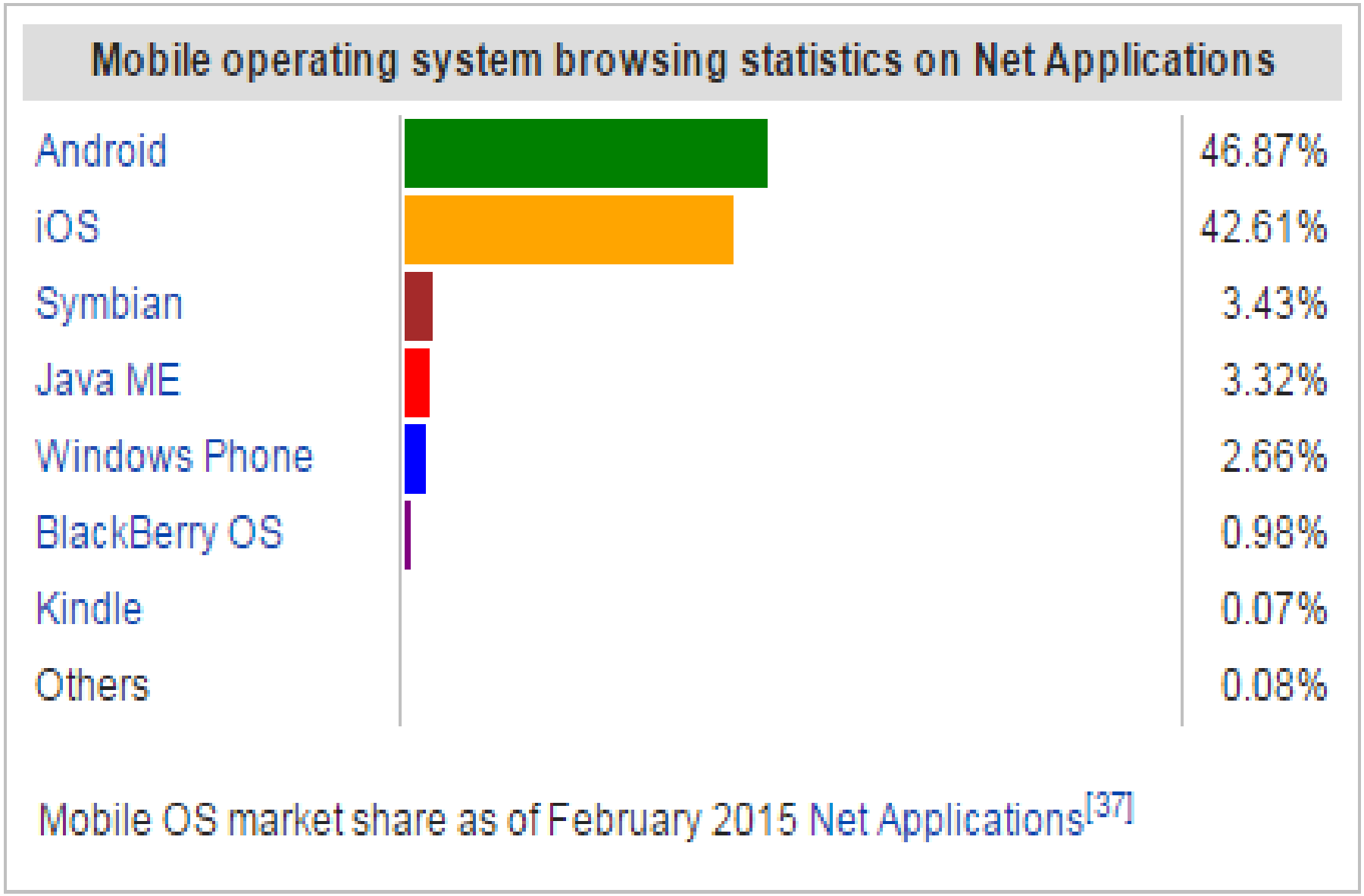
Previous Image

Gartner

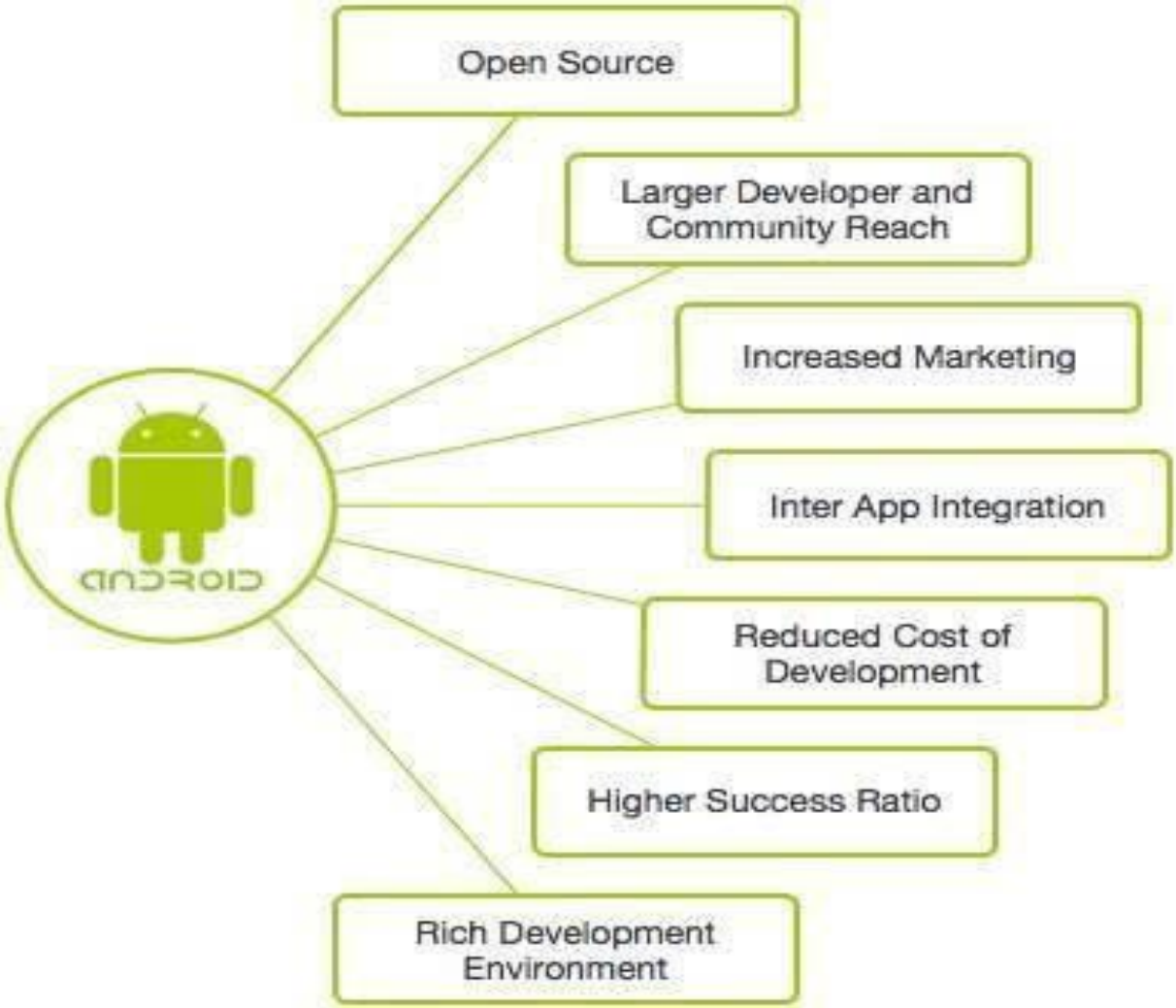
Worldwide Mobile Communications Device Open OS Sales to End Users by OS (Thousands of Units)

OS	2010	2011	2012	2015
Symbian	111,577	89,930	32,666	661
Market Share (%)	37.6	19.2	5.2	0.1
Android	67,225	179,873	310,088	539,318
Market Share (%)	22.7	38.5	49.2	48.8
Research In Motion	47,452	62,600	79,335	122,864
Market Share (%)	16.0	13.4	12.6	11.1
iOS	46,598	90,560	118,848	189,924
Market Share (%)	15.7	19.4	18.9	17.2
Microsoft	12,378	26,346	68,156	215,998
Market Share (%)	4.2	5.6	10.8	19.5
Other Operating Systems	11,417.4	18,392.3	21,383.7	36,133.9
Market Share (%)	3.8	3.9	3.4	3.3
Total Market	**296,647**	**467,701**	**630,476**	**1,104,898**

Source: Gartner (April 2011)


Mobile operating system browsing statistics on Net Applications
Android
46.87%
iOS
42.61%
Symbian
3.43%
Java ME
3.32%
Windows Phone
2.66%
BlackBerry OS
0.98%
Kindle
0.07%
Others
0.08%
Mobile OS market share as of February 2015 Net Applications[37]

Why Android?


Open Source
Larger Developer and Community Reach
Increased Marketing
Inter App Integration
Reduced Cost of Development
Higher Success Ratio
Rich Development Environment
android

Why Android? (con...)

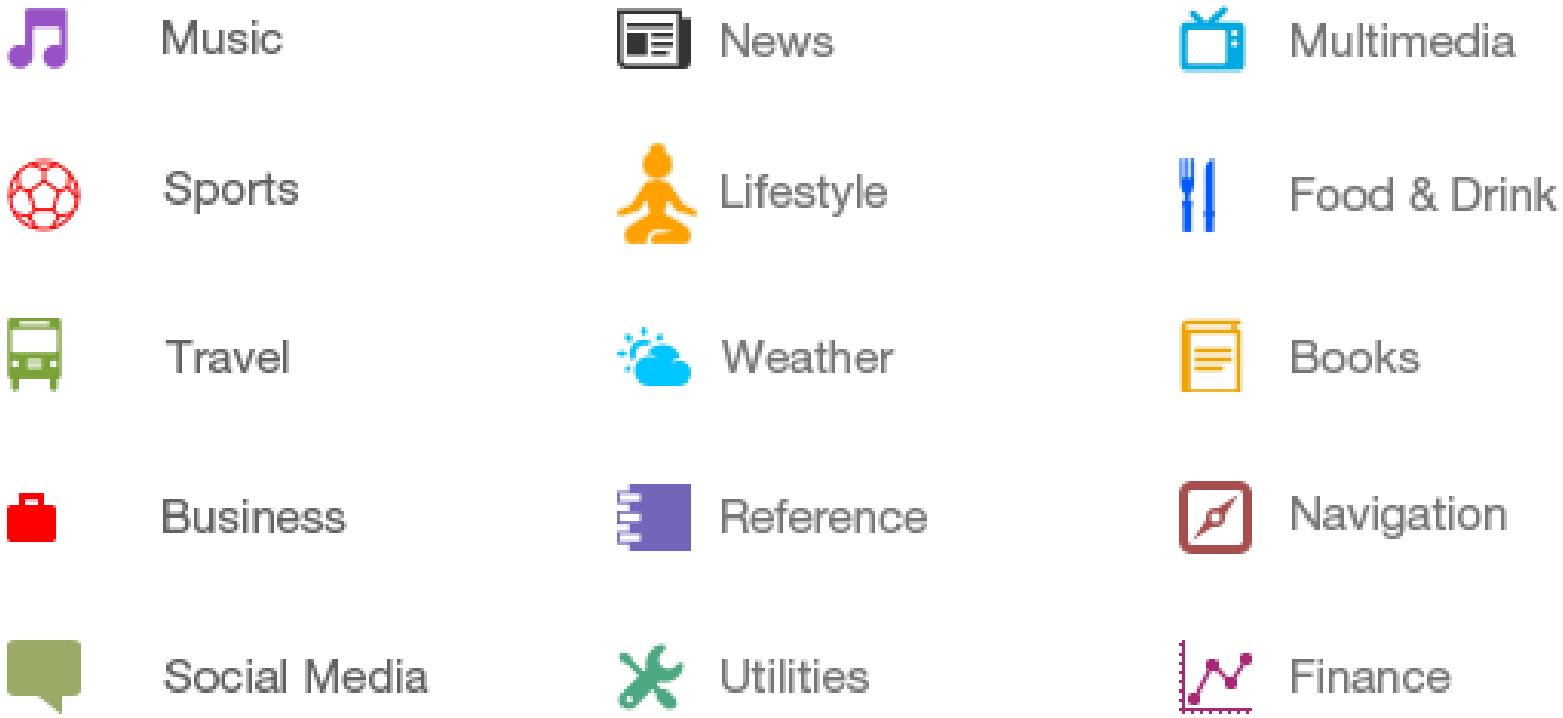
- A lot of students have them
  - 2010 survey by University of Colorado : 22% of college students have Android phone (26% Blackberry, 40% iPhone)
  - Gartner survey: Android used on 22.7% of smart phones sold world-wide in 2010 (37.6% Symbian, 15.7% iOS)
- Students already know Java and Eclipse
  - Low learning curve
  - CS students can use App Inventor for Android

Android Applications

- Android applications are usually developed in the Java language using the Android Software Development Kit
- Once developed, Android applications can be packaged easily and sold out either through a store such as **Google Play**, **SlideME**, **Opera Mobile Store**, **Mobango**, **F-droid** and the **Amazon Appstore**.
- Android powers hundreds of millions of mobile devices in more than 190 countries around the world. It's the largest installed base of any mobile platform and growing fast.
- Every day more than 1 million new Android devices are activated worldwide.

Categories of Android applications

- There are many android applications in the market



Features of Android

- Android is a powerful operating system competing with Apple 4GS and supports great features.

Features	Description
Beautiful UI	Android OS basic screen provides a beautiful and intuitive user interface.
Connectivity	GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, NFC and WiMAX.
Storage	SQLite, a lightweight relational database, is used for data storage purposes.

Features of Android (con..)

Features	Description
Media support	H.263, H.264, MPEG-4 SP, AMR, AMR-WB, AAC, HE-AAC, AAC 5.1, MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF, and BMP
Messaging	SMS and MMS
Web browser	Based on the open-source WebKit layout engine, coupled with Chrome's V8 JavaScript engine supporting HTML5 and CSS3.
Multi-touch	Android has native support for multi-touch which was initially made available in handsets such as the HTC Hero.
Multi-tasking	User can jump from one task to another and same time various application can run simultaneously.

Features of Android (con..)

Features	Description
Resizable widgets	Widgets are resizable, so users can expand them to show more content or shrink them to save space
Multi-Language	Supports single direction and bi-directional text.
GCM	Google Cloud Messaging (GCM) is a service that lets developers send short message data to their users on Android devices.
Wi-Fi Direct	A technology that lets apps discover and pair directly, over a high-bandwidth peer-to-peer connection.
Android Beam	A popular NFC-based technology that lets users instantly share, just by touching two NFC-enabled phones together.

Android Features (con...)

- **Application framework** enabling reuse and replacement of components
- **Dalvik virtual machine** optimized for mobile devices
- **Integrated browser** based on the open source webkit engine
- **Optimized graphics** powered by a custom 2D graphics library; 3D graphics based on the OpenGL ES 1.0 specification (hardware acceleration optional)
- **SQLite** for structured data storage
- **Media support** for common audio, video, and still image formats (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)

What does it have that other's don't?

- Google Map Applications
- Background Services and Applications
- Shared Data and Inter-process communication
- All Applications are created Equal
- P2P Inter-device Application Messaging
- MVC2 Architecture

MVC2

- The goal of the MVC design pattern is to separate the application object (model) from the way it is represented to the user (view) from the way in which the user controls it (controller).

Manufacturer and carrier support

- Almost all carriers have Android

- **HTC**
- **LG**
- **Sony-Ericsson**
- **Geeksphone**
- **Dell**
- **Motorola**
- **Acer**
- **Samsung**
- **Archos**
- **Lenovo**
- **Huawei**

Xperia X10a
Confirmed
Sony-Ericsson

GW880
Confirmed
LG

Nexus One
Available
HTC

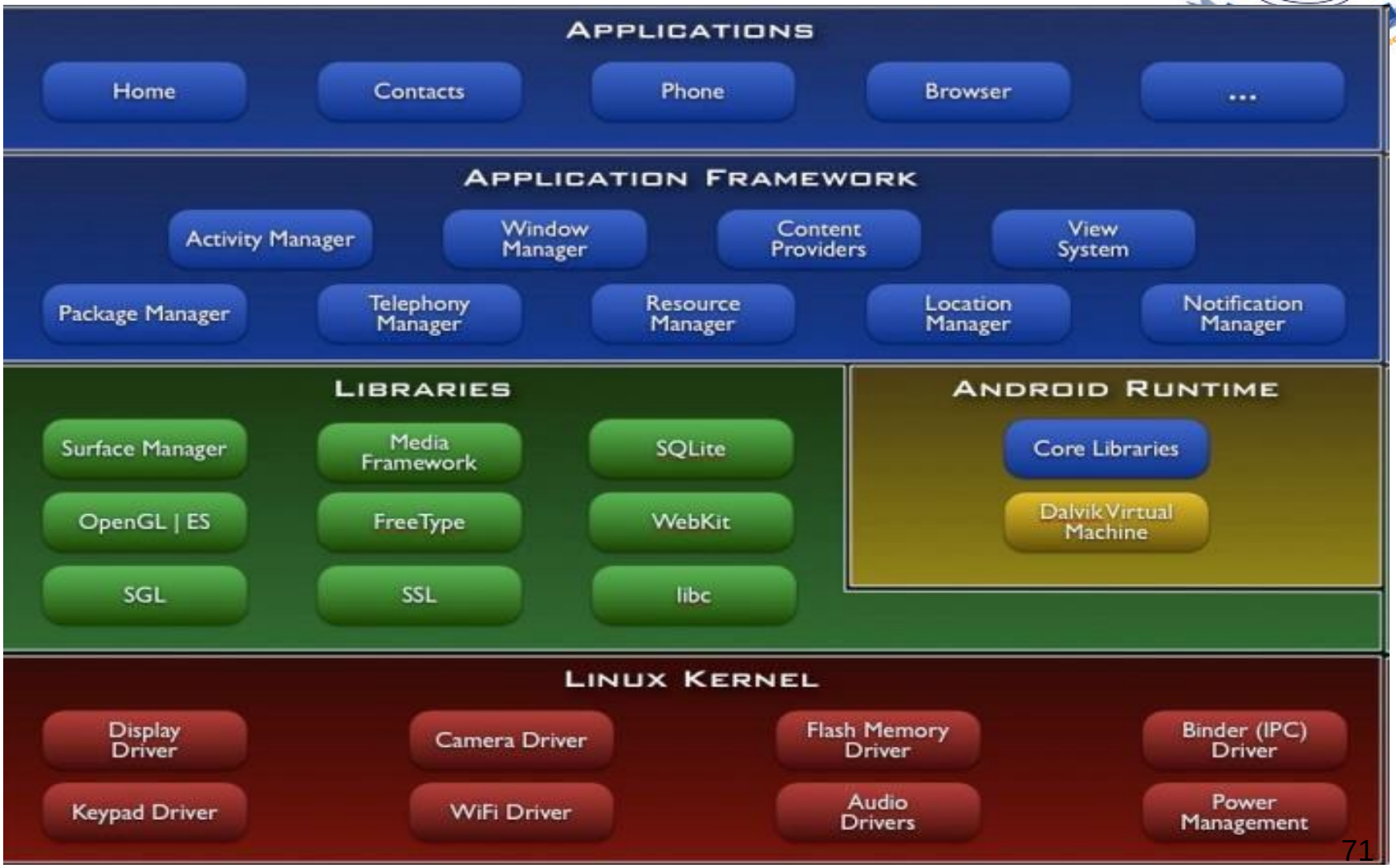
One
Confirmed
Geeksphone

Streak
Rumored
Dell

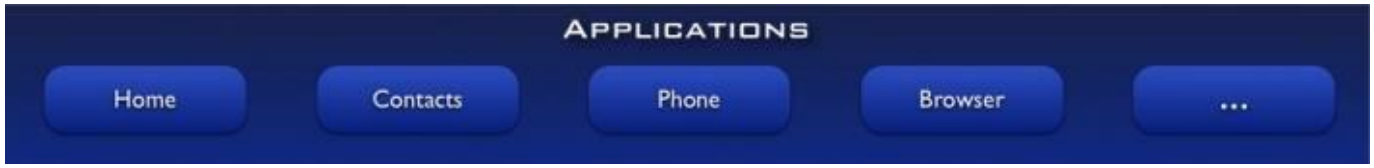
Archos Phone
Confirmed
Archos

Pulse
Available
Huawei

5. Architecture



Android S/W Stack - Applications



- Android provides a set of core applications:
  - ✓ Email Client
  - ✓ SMS Program
  - ✓ Calendar
  - ✓ Maps
  - ✓ Browser
  - ✓ Contacts
  - ✓ Etc

- All applications are written using the Java language.

Android S/W Stack – Application Framework

- Enabling and simplifying the reuse of components
  - ✓ Developers have full access to the same framework APIs used by the core applications.
  - ✓ Users are allowed to replace components.

Android S/W Stack – App Framework (Cont)

- 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 back-stack

Android S/W Stack - Libraries

- Including a set of C/C++ libraries used by components of the Android system
- Interface through Java
- Surface manager – Handling UI Windows
- 2D and 3D graphics
- Media codes, SQLite, Browser engine

Android S/W Stack - Runtime

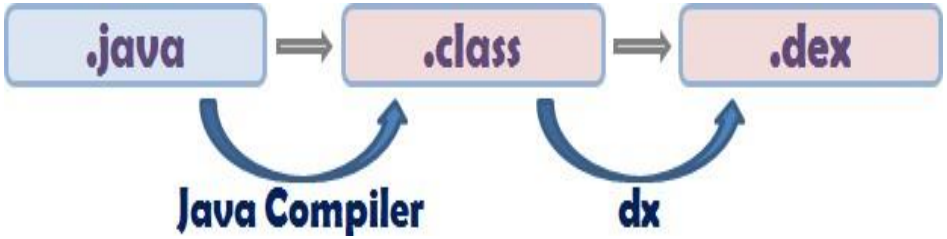
- Core Libraries
  - ✓ Providing most of the functionality available in the core libraries of the Java language
  - ✓ APIs
    - ➢ Data Structures
    - ➢ Utilities
    - ➢ File Access
    - ➢ Network Access
    - ➢ Graphics

Android S/W Stack – Runtime (Cont)

- Dalvik Virtual Machine
  - ✓ Providing environment on which every Android application runs
    - ➢ Each Android application runs in its own process, with its own instance of the Dalvik VM.
    - ➢ Dalvik has been written such that a device can run multiple VMs efficiently.
  - ✓ Register-based virtual machine

Android S/W Stack – Runtime (Cont)

- Dalvik Virtual Machine (Cont)
  - ✓Executing the Dalvik Executable (.dex) format
    - ➢.dex format is optimized for minimal memory footprint.
    - ➢Compilation



  - ✓Relying on the Linux Kernel for:
    - ➢Threading
    - ➢Low-level memory management

Android S/W Stack – Linux Kernel

- Relying on Linux Kernel 2.6 for core system services
  - ✓ Memory and Process Management
  - ✓ Network Stack
  - ✓ Driver Model
  - ✓ Security
- Providing an abstraction layer between the H/W and the rest of the S/W stack

Android development setup

Follow the instructions ...

Download the software from the URL:
http://developer.android.com/sdk/index.html

Install the following Softwares:
- Android SDK
- Eclipse IDE (3.4 or newer)
- Android Development Tools (ADT) Eclipse plug-in

Bring with you (optional):
- Android OS enabled Mobile device
- USB cable so you can test your app on your phone

Application Fundamentals

- Apps are written in Java
- Bundled by Android Asset Packaging Tool
- Every App runs its own Linux process
- Each process has it's own Java Virtual Machine
- Each App is assigned a unique Linux user ID
- Apps can share the same user ID to see each other's files

Applications

- Lifestyle applications for senior citizens
- Environmental applications that give data about pollution levels.
- Emergency services (Hospitals, Police station etc.,)
- Bus services
- Games
- E-governance
- Google map

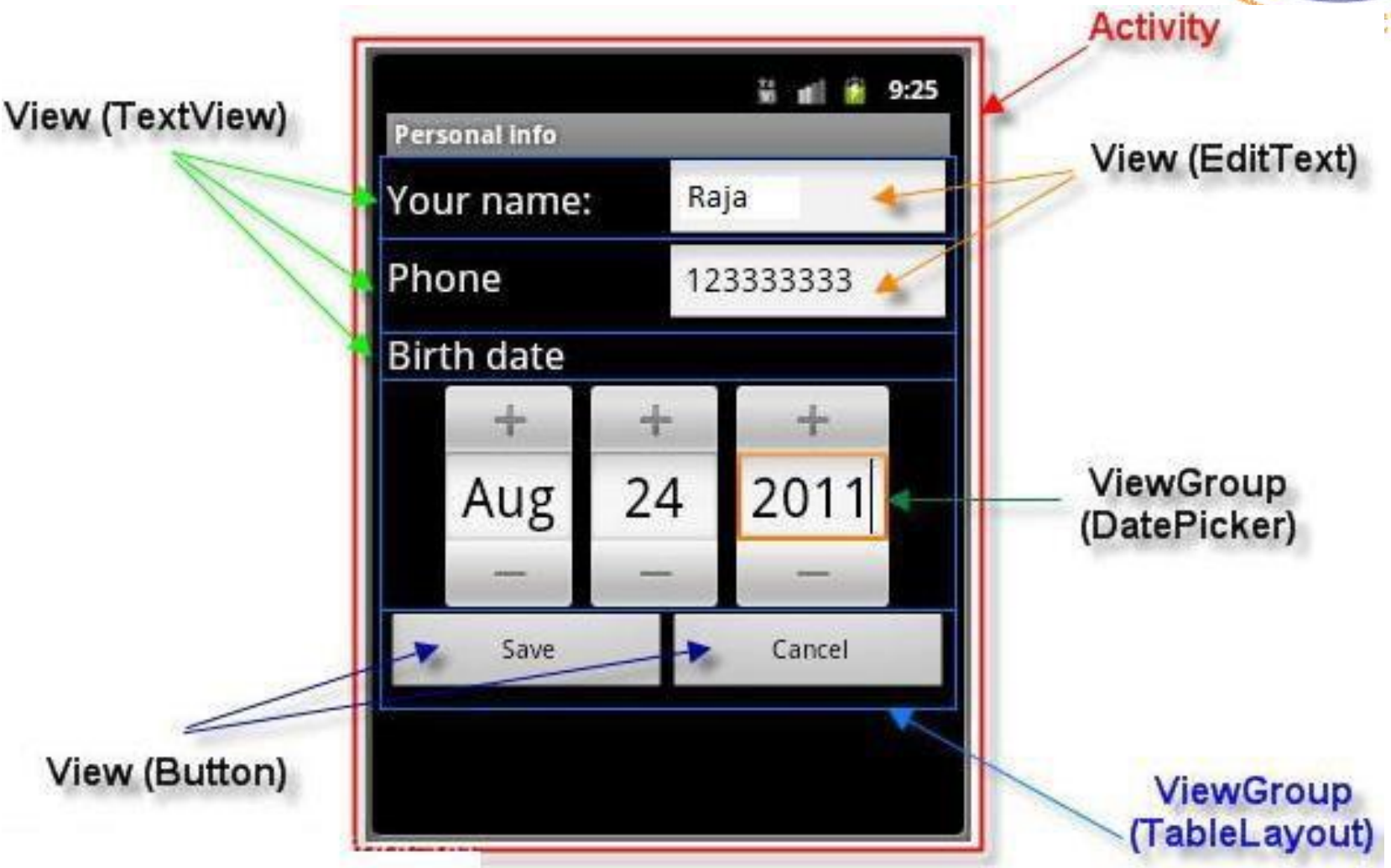
6. UI Layouts

- The basic building block for user interface is a View object which is created from the View class
- It occupies a rectangular area on the screen and is responsible for drawing and event handling.
- View is the base class for widgets, which are used to create interactive UI components like buttons, text fields, etc.

UI Layouts (con…)

- The ViewGroup is a subclass of View and provides invisible container that hold other Views or other ViewGroups and define their layout properties.
- At third level we have different layouts which are subclasses of ViewGroup class
- A typical layout defines the visual structure for an Android user interface.

UI Layouts (con…)



UI Layouts (con…)

- To declare the layout using simple XML file main_layout.xml which is located in the res/layout folder of your project.
- A layout may contain any type of widgets such as buttons, labels, textboxes, and so on.

A simple XML file having LinearLayout

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="fill_parent"  android:layout_height="fill_parent"
android:orientation="vertical" >
<TextView android:id="@+id/text"
android:layout_width="wrap_content" android:layout_height="wrap_content"
android:text="This is a TextView" />
<Button android:id="@+id/button"
android:layout_width="wrap_content" android:layout_height="wrap_content"
android:text="This is a Button" />
<!-- More GUI components go here -->
</LinearLayout>
```

..LinearLayout (con…)

- Once the layout has created, it can loaded by the help of application code
- Sample Code

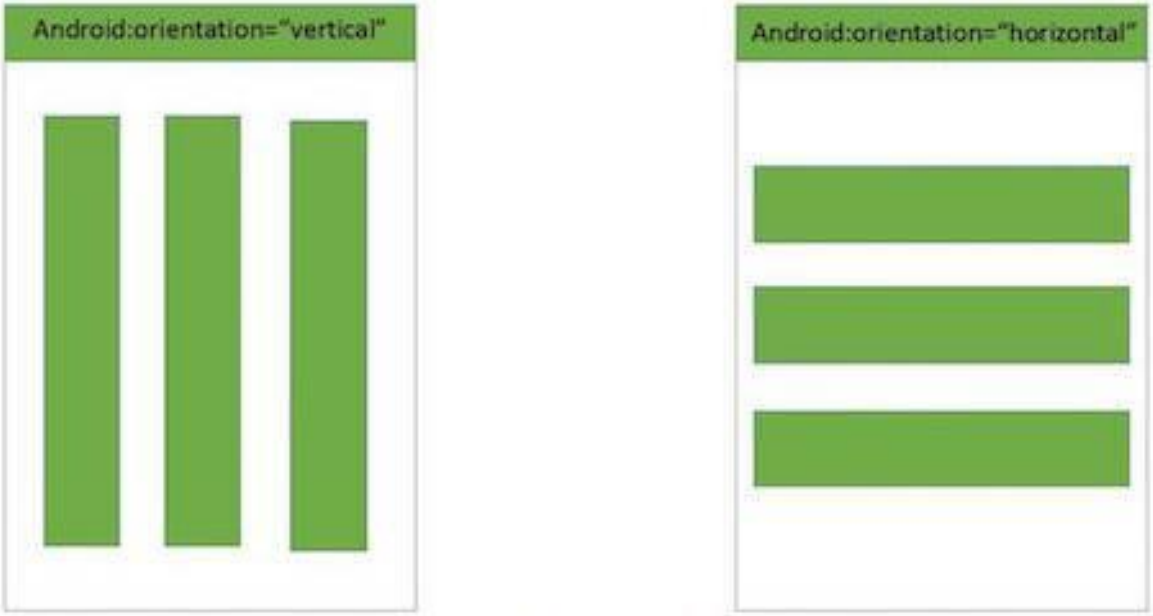
```
public void onCreate(Bundle
   savedInstanceState)
{
super.onCreate(savedInstanceState);
   setContentView(R.layout.activity_main);
}
```

Layout Types

- Linear Layout
- Relative Layout
- Table Layout
- Absolute Layout
- Frame Layout
- List View
- Grid View

Linear Layout

- Linear Layout is a view group that aligns all children in either vertically or horizontally.



LINEAR LAYOUT

Attributes

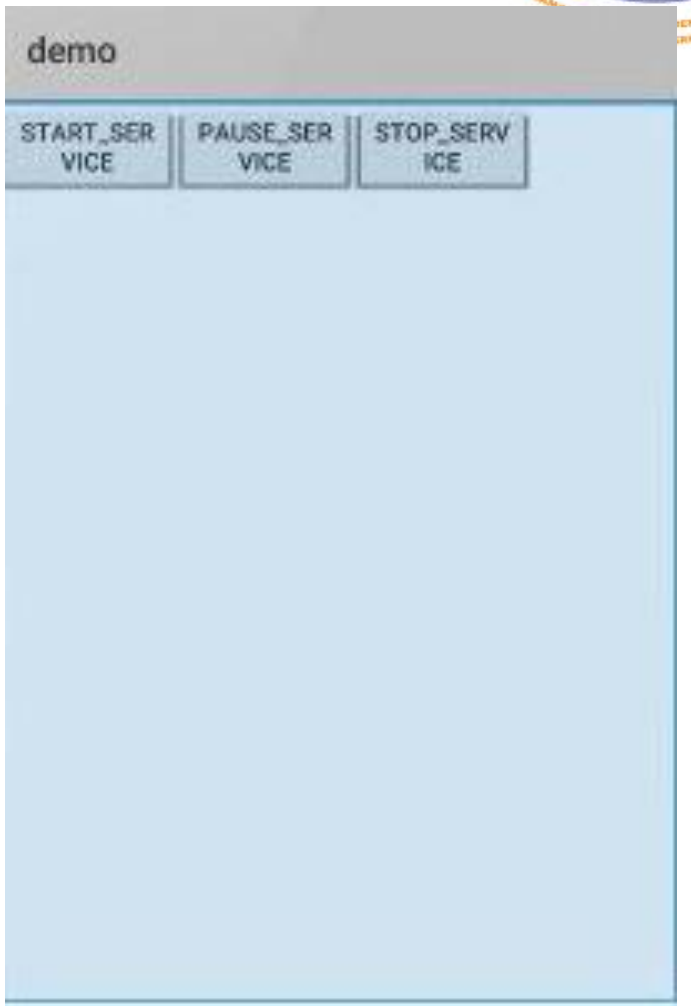
Attribute	Description
android:id	This is the ID which uniquely identifies the layout.
android:gravity	This specifies how an object should position its content, on both the X and Y axes. Possible values are top, bottom, left, right, center, center_vertical, center_horizontal etc.
android:orientation	This specifies the direction of arrangement and you will use "horizontal" for a row, "vertical" for a column. The default is horizontal.

Example

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/
res/android"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:orientation="vertical" >
<!-- More GUI components go here -->
</LinearLayout>
```

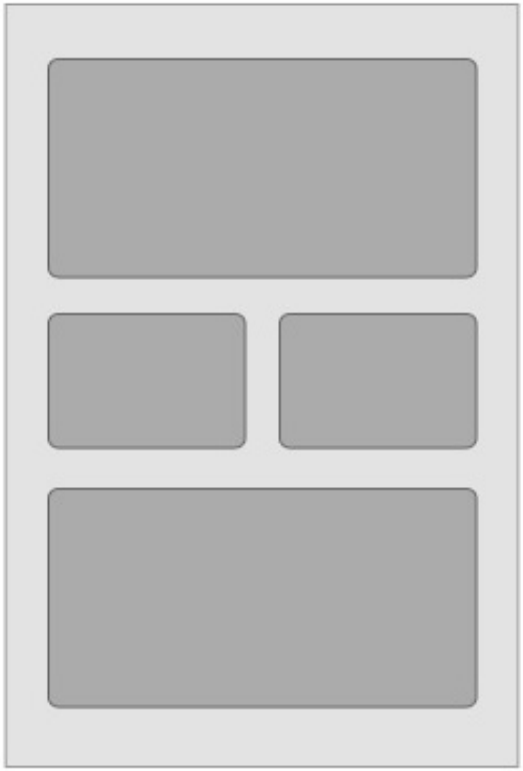
Output


demo
START_SERVICE
PAUSE_SERVICE
STOP_SERVICE


demo
START_SER VICE
PAUSE_SER VICE
STOP_SERV ICE

Relative Layout

- Relative Layout enables you to specify how child views are positioned relative to each other.
- The position of each view can be specified as relative to sibling elements or relative to the parent.



RELATIVE LAYOUT

Attributes

Attribute	Description
android:id	This is the ID which uniquely identifies the layout.
android:gravity	This specifies how an object should position its content, on both the X and Y axes. Possible values are top, bottom, left, right, center, center_vertical, center_horizontal etc.

By default, all child views are drawn at the top-left of the layout, so you must define the position of each view using the various layout properties.

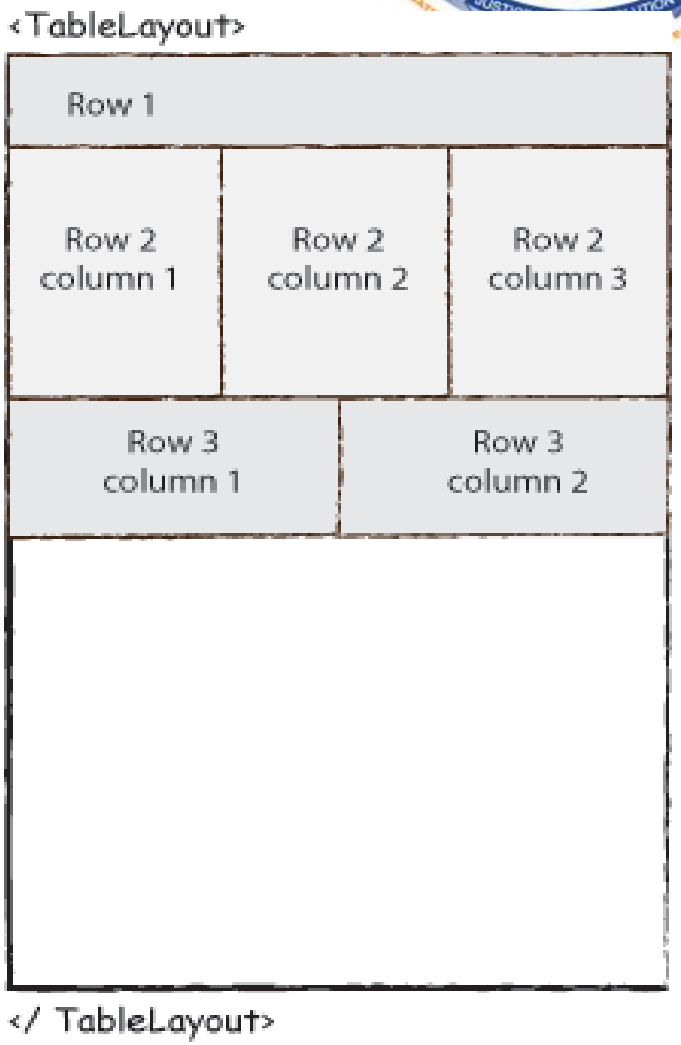
Example

```
<RelativeLayout
   xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="fill_parent"
   android:layout_height="fill_parent"
   android:paddingLeft="16dp"
   android:paddingRight="16dp" >
<!-- More GUI components go here -->
</RelativeLayout>
```

Output

Table Layout

- TableLayout going to be arranged groups of views into rows and columns.
- Use the <TableRow> element to build a row in the table.
- Each row has zero or more cells; each cell can hold one View object
- It don't display border lines for their rows, columns, or cells.



Attributes

Attribute	Description
android:id	This is the ID which uniquely identifies the layout.
android:collapseColumns	This specifies the zero-based index of the columns to collapse.
android:shrinkColumns	The zero-based index of the columns to shrink.
android:stretchColumns	The zero-based index of the columns to stretch.

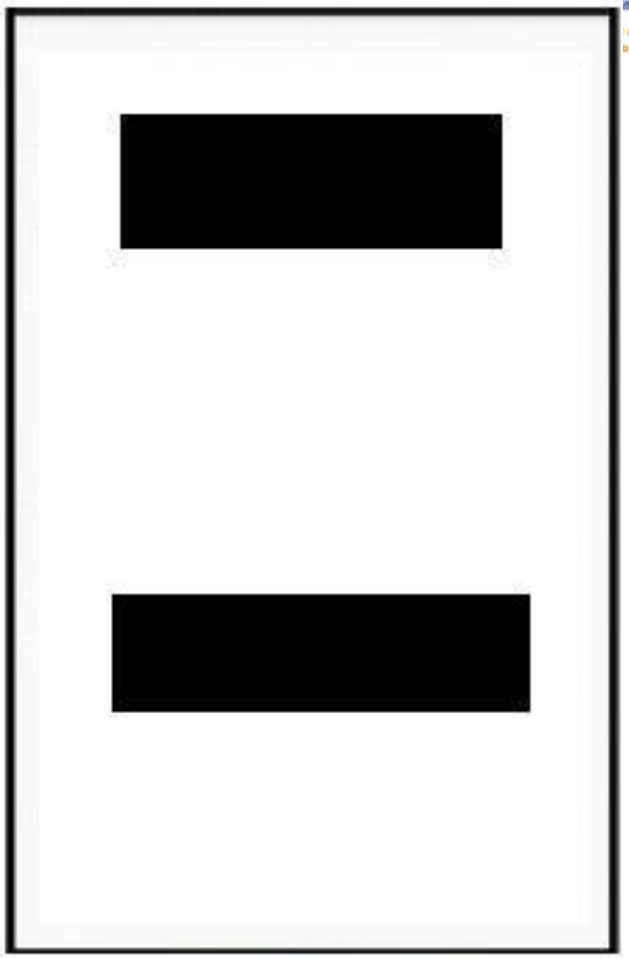
Example

```
<TableLayout
    xmlns:android="http://schemas.android.com/apk/res/and
    roid" android:layout_width="fill_parent"
    android:layout_height="fill_parent">
<TableRow
android:layout_width="fill_parent"
    android:layout_height="fill_parent">
<!-- More GUI components go here -->
</TableRow>
<!-- More Table rows go here -->
</TableLayout>
```

Output

Absolute Layout

- Absolute Layout lets you specify exact locations (x/y coordinates) of its children.
- Absolute layouts are less flexible and harder to maintain than other types of layouts without absolute positioning.



AbsoluteLayout

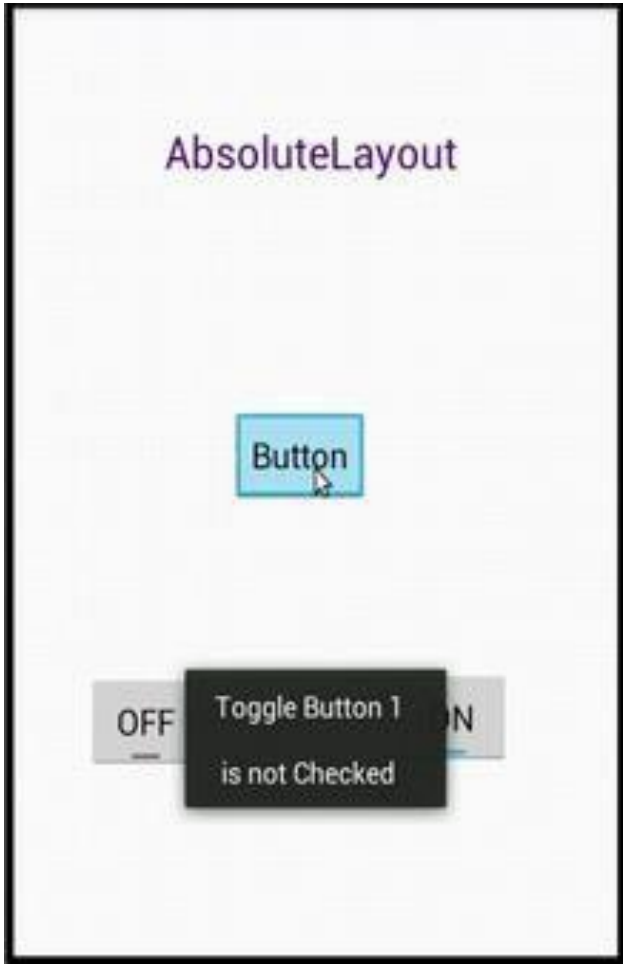
Attributes

Attribute	Description
android:id	This is the ID which uniquely identifies the layout.
android:layout_x	This specifies the x-coordinate of the view.
android:layout_y	This specifies the y-coordinate of the view.

Example

```
<AbsoluteLayout
    xmlns:android="http://schemas.android.com/apk/res/andr
    oid"
android:layout_width="fill_parent"
    android:layout_height="fill_parent">
<Button android:layout_width="100dp"
    android:layout_height="wrap_content"
    android:text="OK"
android:layout_x="50px"  android:layout_y="361px" />
<!-- More GUI components go here -->
</AbsoluteLayout>
```

Output



Frame Layout

- Frame Layout is designed to block out an area on the screen to display a single item.
- Generally, Frame Layout should be used to hold a single child view, because it can be difficult to organize child views in a way that's scalable to different screen sizes without the children overlapping each other.

Attributes

Attribute	Description
android:id	This is the ID which uniquely identifies the layout.
android:foreground	This defines the drawable to draw over the content and possible values may be a color value.
android:foregroundGravity	Defines the gravity to apply to the foreground drawable. The gravity defaults to fill. Possible values are top, bottom, left, right, center, center_vertical, center_horizontal etc.
android:measureAllChildren	Determines whether to measure all children or just those in the VISIBLE or INVISIBLE state when measuring. Defaults to false.

Example

```
<FrameLayout
   xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="fill_parent"
   android:layout_height="fill_parent">
<!-- More GUI components go here -->
</FrameLayout>
```

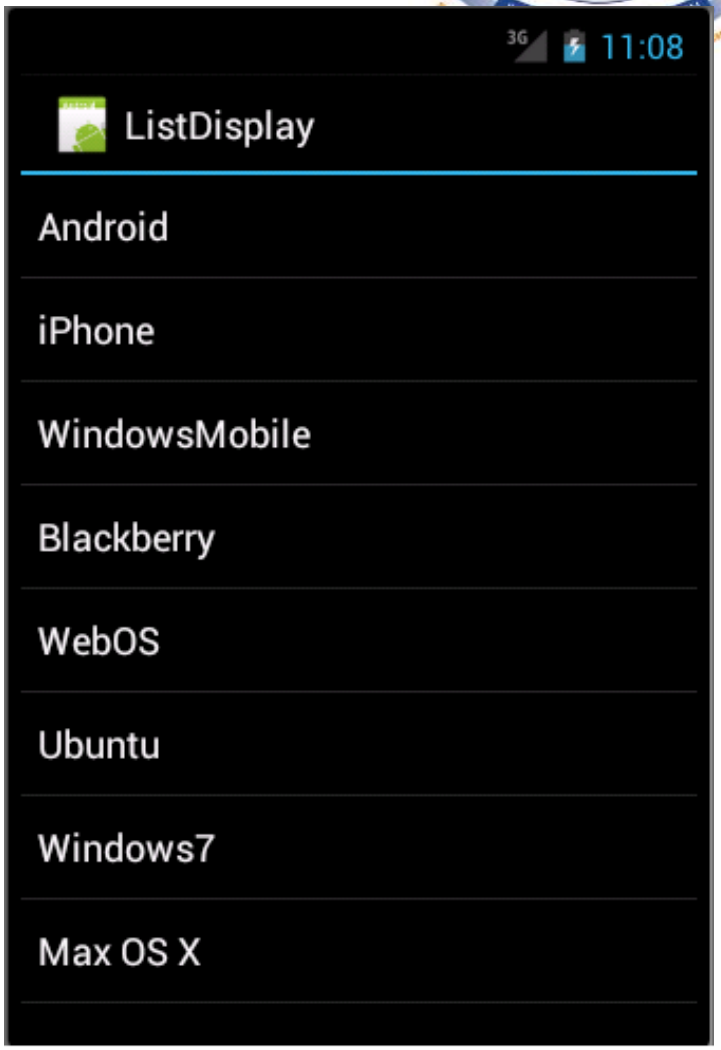
Output

FRAME LAYOUT

List View

- List View is a view which groups several items and display them in vertical scrollable list.
- The list items are automatically inserted to the list using

an Adapter that pulls content from a source such as an array or database.



LIST VIEW

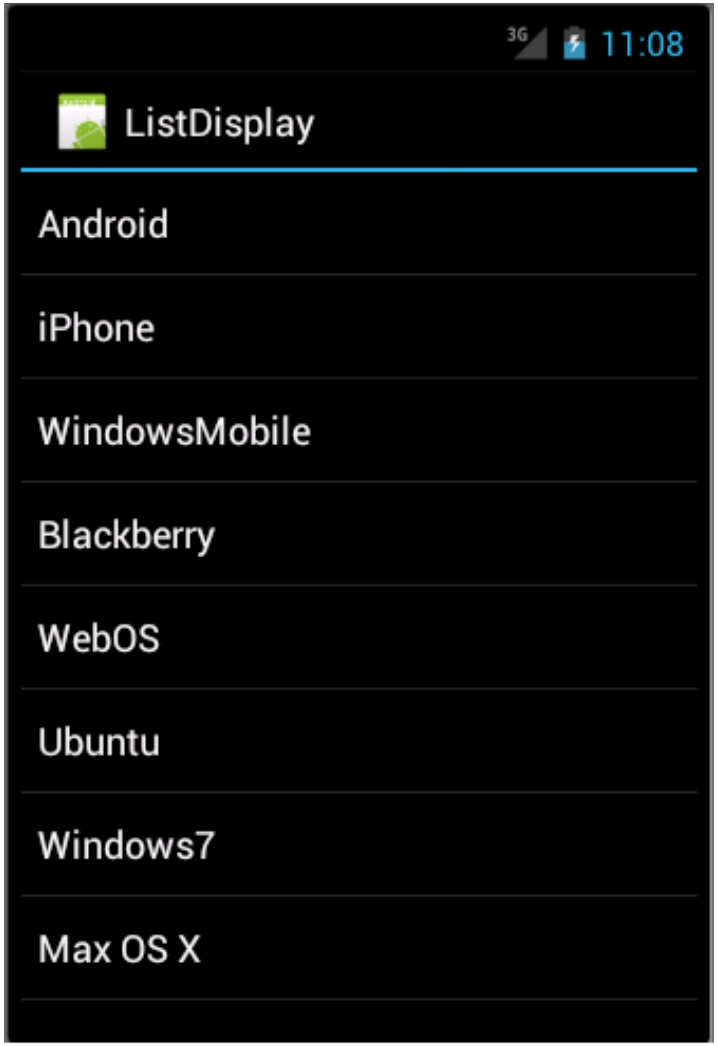
Attributes

Attribute	Description
android:id	This is the ID which uniquely identifies the layout.
android:divider	This is drawable or color to draw between list items.
android:dividerHeight	This specifies height of the divider. This could be in px, dp, sp, in, or mm.
android:entries	Specifies the reference to an array resource that will populate the ListView.
android:footerDividersEnabled	When set to false, the ListView will not draw the divider before each footer view. The default value is true.
android:headerDividersEnabled	When set to false, the ListView will not draw the divider after each header view. The default value is true.

Example

```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/and
    roid" xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
tools:context=".ListActivity" >
<ListView android:id="@+id/mobile_list"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >
</ListView>
</LinearLayout>
```

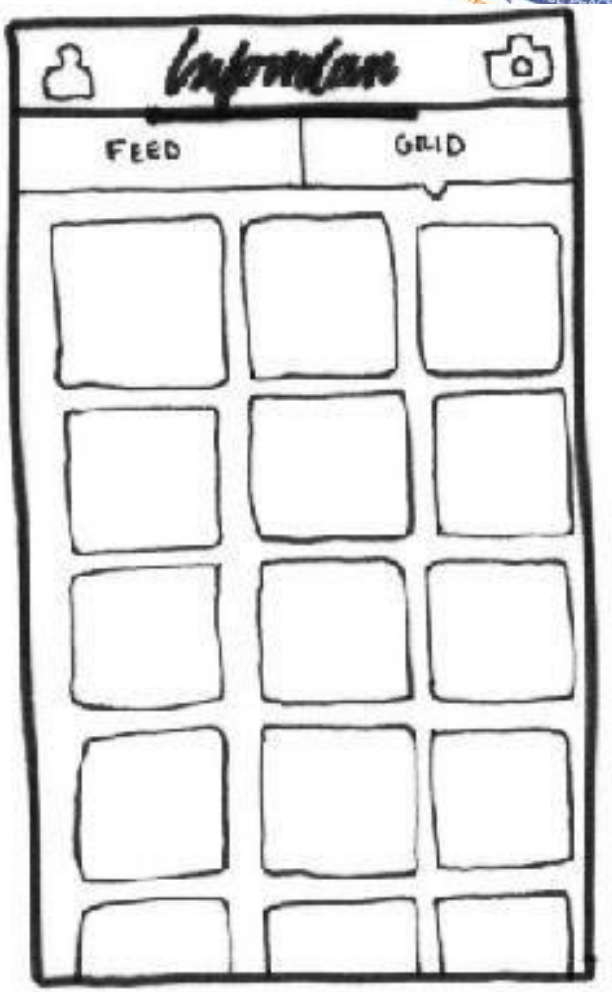
Output



LIST VIEW

Grid View

- Grid View shows items in two-dimensional scrolling grid (rows & columns)
- The grid items are not necessarily predetermined but they automatically inserted to the layout using a ListAdapter



GRID VIEW

Grid View (con…)

- An adapter actually bridges between UI components and the data source that fill data into UI Component.
- Adapter can be used to supply the data to like spinner, list view, grid view etc.

Attributes

Attribute	Description
android:id	This is the ID which uniquely identifies the layout.
android:columnWidth	This specifies the fixed width for each column. This could be in px, dp, sp, in, or mm.
android:gravity	Specifies the gravity within each cell. Possible values are top, bottom, left, right, center, center_vertical, center_horizontal etc.
android:horizontalSpacing	Defines the default horizontal spacing between columns. This could be in px, dp, sp, in, or mm.
android:numColumns	Defines how many columns to show.
android:verticalSpacing	Defines the default vertical spacing between rows. This could be in px, dp, sp, in, or mm.

Example

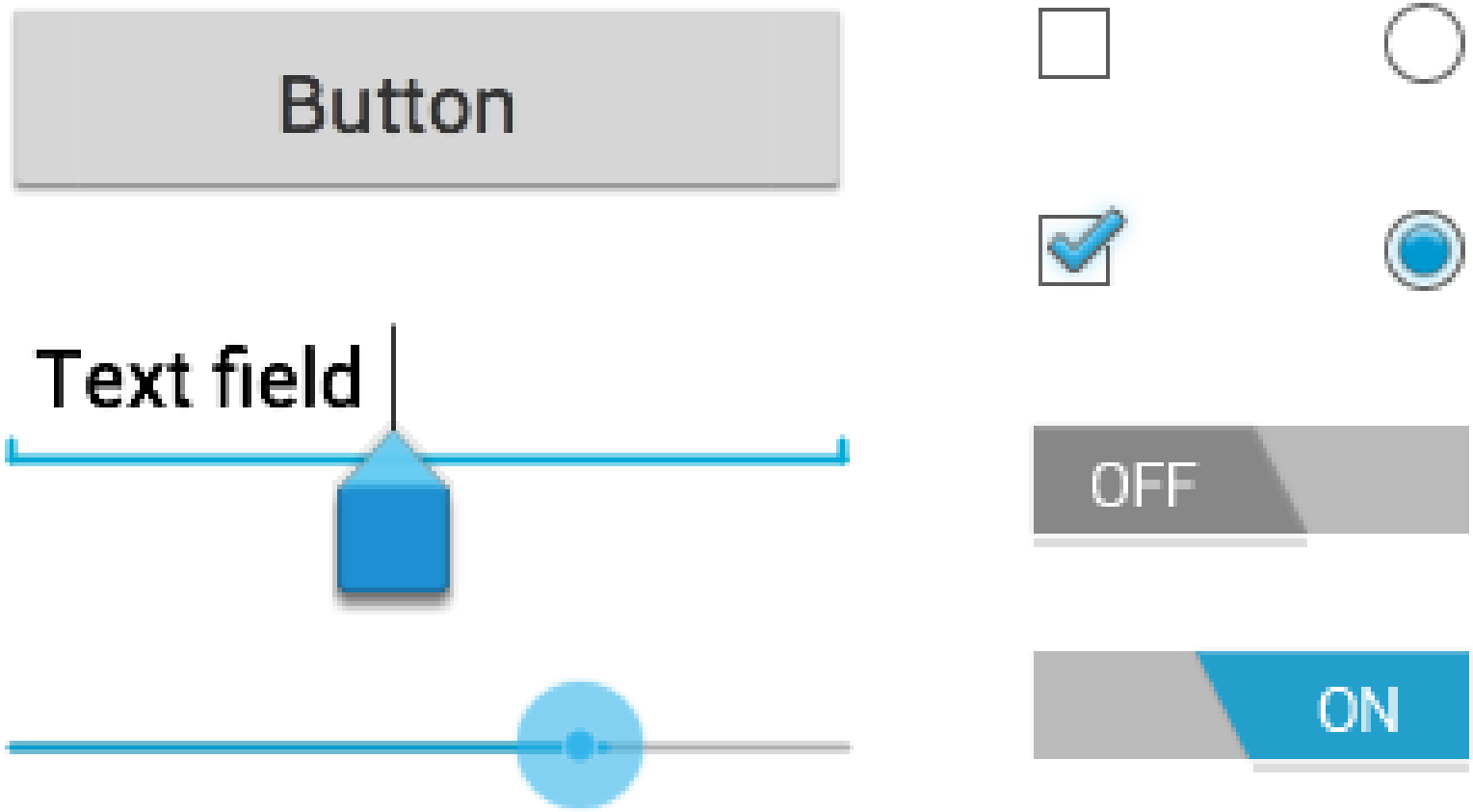
```
<?xml version="1.0" encoding="utf-8"?>
<GridView
xmlns:android="http://schemas.android.com/apk/res/android"
android:id="@+id/gridview"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:columnWidth="90dp"
android:numColumns="auto_fit"
android:verticalSpacing="10dp"
android:horizontalSpacing="10dp"
android:stretchMode="columnWidth"
android:gravity="center" />
```

Output

7. UI Controls / Widgets

- Input controls are the interactive components in your app's user interface.
- Android provides a wide variety of controls you can use in your UI, such as buttons, text fields, seek bars, check box, zoom buttons, toggle buttons, and many more

UI Controls (con…)



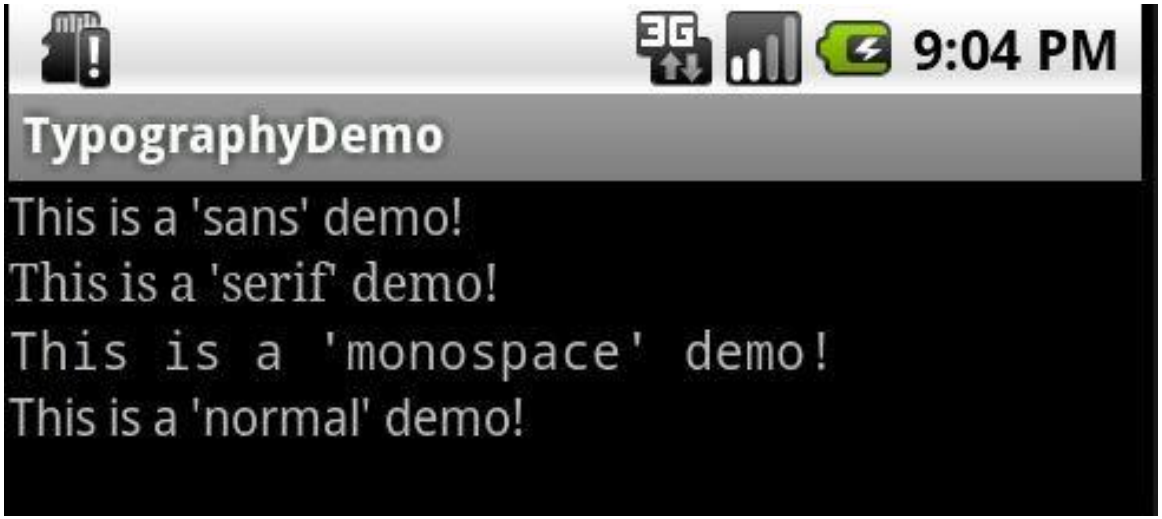
UI ELEMENTS

UI Controls (con…)

- TextView
- EditText
- Button
- ImageButton
- ToggleButton
- AutoCompleteTextVie
  w
- CheckBox
- RadioButton
- RadioGroup
- ProgressBar
- Spinner
- TimePicker
- DatePicker

TextView Control

- A TextView displays text to the user and optionally allows them to edit it.
- A TextView is a complete text editor, however the basic class is configured to not allow editing.



Attributes

Attribute	Description
android:id	This is the ID which uniquely identifies the control.
android:fontFamily	Font family (named by string) for the text.
android:inputType	The type of data being placed in a text field. Phone, Date, Time, Number, Password etc.
android:text	Text to display.
android:textAllCaps	Present the text in ALL CAPS. Possible value either "true" or "false".
android:textColor	Text color. May be a color value.
android:textSize	Size of the text. Recommended dimension type for text is "sp" for scaled-pixels.

Example

In XML:

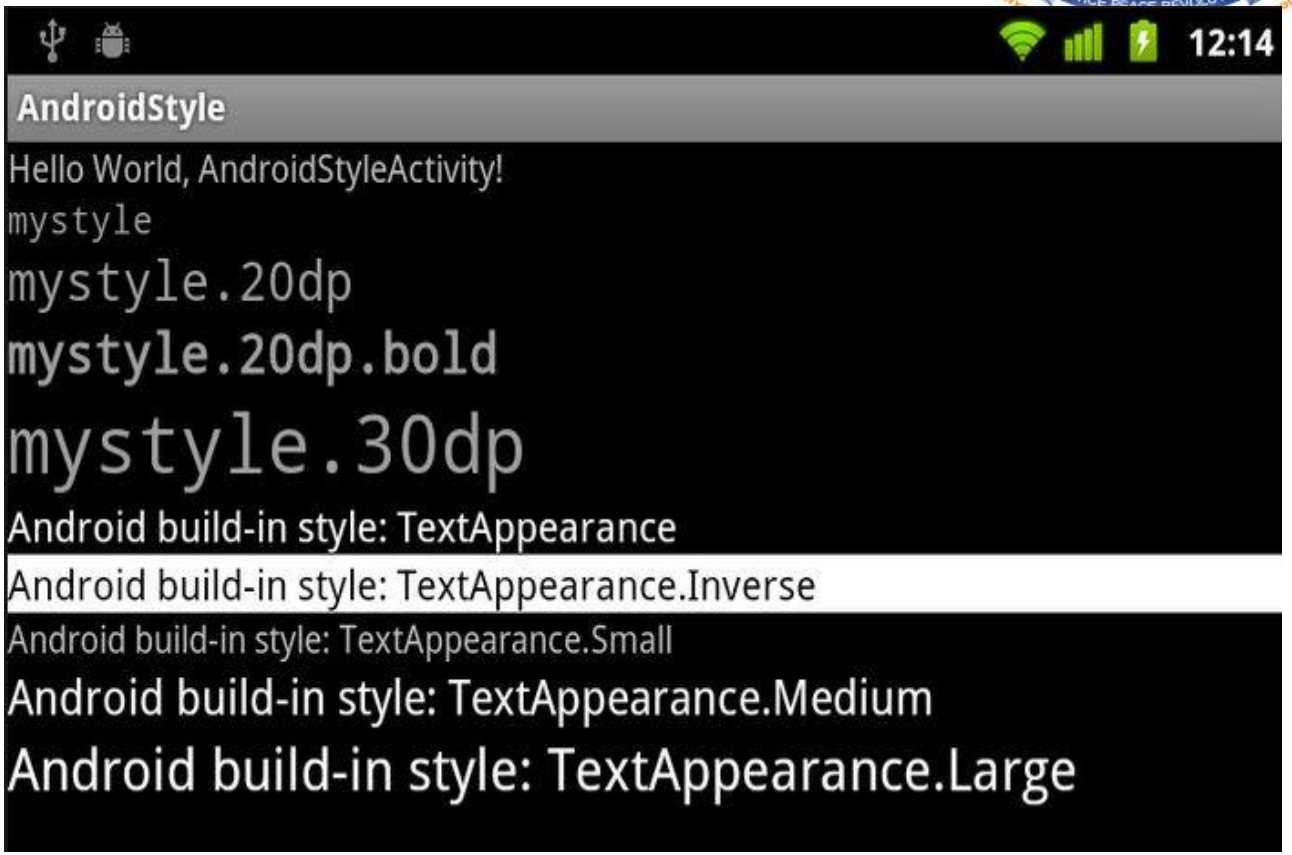
```
<TextView
android:id="@+id/text_id" android:layout_width="300dp"
android:layout_height="200dp"
android:capitalize="characters" android:text="hello_world"
android:textColor="@android:color/holo_blue_dark"
android:textColorHighlight="@android:color/primary_text_d
ark" android:layout_centerVertical="true"
android:layout_alignParentEnd="true"
android:textSize="50dp"/>
```

In JAVA:

```
TextView txtView = (TextView) findViewById(R.id.text_id);
```

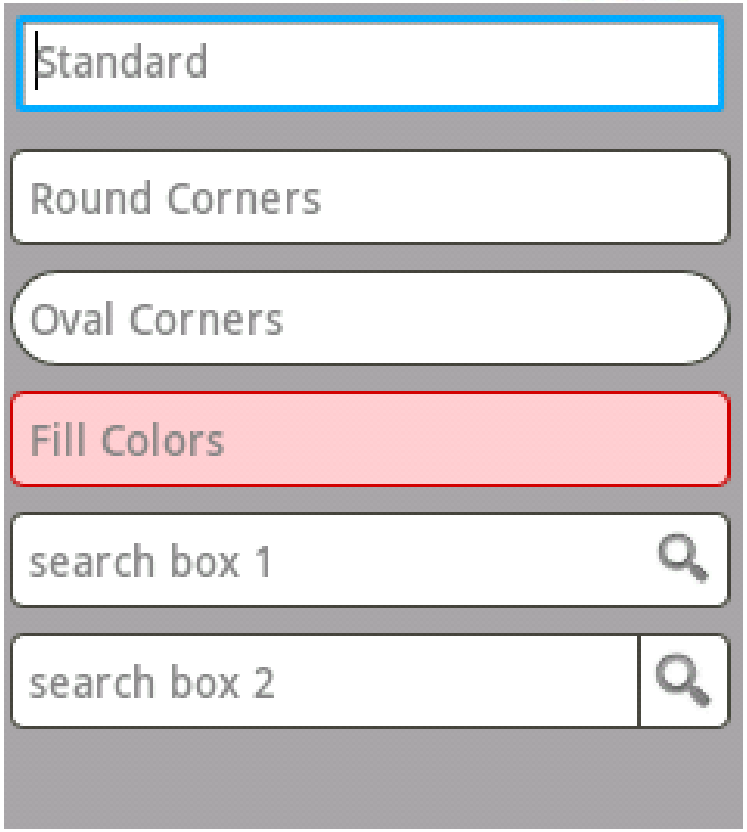
Output


demo
hello_world


12:14
AndroidStyle
Hello World, AndroidStyleActivity!
mystyle
mystyle.20dp
mystyle.20dp.bold
mystyle.30dp
Android build-in style: TextAppearance
Android build-in style: TextAppearance.Inverse
Android build-in style: TextAppearance.Small
Android build-in style: TextAppearance.Medium
Android build-in style: TextAppearance.Large

EditText Control

- A EditText is an overlay over TextView that configures itself to be editable.
- It is the predefined subclass of TextView that includes rich editing capabilities.



STYLES OF EDIT TEXT

Attributes

Attribute	Description
android:autoText	If set, specifies that this TextView has a textual input method and automatically corrects some common spelling errors.
android:drawableBottom	This is the drawable to be drawn below the text.
android:drawableRight	This is the drawable to be drawn to the right of the text.
android:editable	If set, specifies that this TextView has an input method.
android:text	This is the Text to display.

Example

In XML:

```
<EditText
android:id="@+id/edittext"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/button"
android:layout_below="@+id/textView1"
android:layout_marginTop="61dp" android:ems="10"
android:text="@string/enter_text"
android:inputType="text" />
```

In JAVA:

```
EditText        eText = (EditText)
findViewById(R.id.edittext);
```

Output

Button Control

- A Button is a Push-button which can be pressed, or clicked, by the user to perform an action.

Attributes

Attribute	Description
android:autoText	If set, specifies that this TextView has a textual input method and automatically corrects some common spelling errors.
android:drawableBottom	This is the drawable to be drawn below the text.
android:drawableRight	This is the drawable to be drawn to the right of the text.
android:editable	If set, specifies that this TextView has an input method.
android:text	This is the Text to display.

Example

In XML:

```
<Button android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Button" android:id="@+id/button"
android:layout_alignTop="@+id/editText"
android:layout_alignLeft="@+id/textView1"
android:layout_alignStart="@+id/textView1"
android:layout_alignRight="@+id/editText"
android:layout_alignEnd="@+id/editText" />
```

Example (con…)

In JAVA:

```
Button b1=(Button)findViewById(R.id.button);
b1.setOnClickListener(new View.OnClickListener()
{
@Override
public void onClick(View v)
{
Toast.makeText(MainActivity.this,"YOUR
MESSAGE",Toast.LENGTH_LONG).show();
}
});
```

Output

ImageButton Control

- A ImageButton is a AbsoluteLayout which enables you to specify the exact location of its children.
- This shows a button with an image (instead of text) that can be pressed or clicked by the user.

Attributes

Attribute	Description
android:adjustViewBounds	Set this to true if you want the ImageView to adjust its bounds to preserve the aspect ratio of its drawable.
android:baseline	This is the offset of the baseline within this view.
android:baselineAlignBottom	If true, the image view will be baseline aligned with based on its bottom edge.
android:cropToPadding	If true, the image will be cropped to fit within its padding.
android:src	This sets a drawable as the content of this ImageView.

Example

In XML:

```
<ImageButton
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/imageButton"
android:layout_centerVertical="true"
android:layout_centerHorizontal="true"
android:src="@drawable/abc"/>
```

Example (con…)

In JAVA:

```
ImageButton imgButton =(ImageButton)
findViewById(R.id.imageButton);
imgButton.setOnClickListener(new
View.OnClickListener()
{
@Override public void onClick(View v)
{

Toast.makeText(getApplicationContext(),“Test
Image Button",Toast.LENGTH_LONG).show();
}
});
```

Output

ToggleButton Control

- A ToggleButton displays checked/unchecked states as a button.
- It is basically an on/off button with a light indicator.

Attributes

Attribute	Description
android:disabledAlpha	This is the alpha to apply to the indicator when disabled.
android:textOff	This is the text for the button when it is not checked.
android:textOn	This is the text for the button when it is checked.

Example

In XML:

```
<ToggleButton
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="On"
android:id="@+id/toggleButton1“
android:checked="true" />
<ToggleButton
android:layout_width="wrap_content"
android:layout_height="wrap_content"
    android:text="Off"
    android:id="@+id/toggleButton2"
    android:checked="true“ />
```

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content
    "
    android:id="@+id/button2"
    android:text="ClickMe“ />
```

Example (con…)

In JAVA:

```
ToggleButton tg1,tg2;
Button b1;  tg1=(ToggleButton)findViewById(R.id.toggleButton1);
tg2=(ToggleButton)findViewById(R.id.toggleButton2);
b1=(Button)findViewById(R.id.button2);  b1.setOnClickListener(new
View.OnClickListener() {  @Override public void onClick(View v) {
StringBuffer result = new StringBuffer();
result.append("You have clicked first ON Button").append(tg1.getText());
result.append("\You have clicked Second ON Button
").append(tg2.getText());
Toast.makeText(MainActivity.this,result.toString(),Toast.LENGTH_SHORT)
.show(); } });
```

Output

AutoCompleteTextView Control

- A AutoCompleteTextView is a view that is similar to EditText, except that it shows a list of completion suggestions automatically while the user is typing.
- The list of suggestions is displayed in drop down menu.
- The user can choose an item from there to replace the content of edit box with.

Attributes

Attribute	Description
android:completionHintView	This defines the hint view displayed in the drop down menu.
android:completionThreshold	This defines the number of characters that the user must type before completion suggestions are displayed in a drop down menu.
android:dropDownAnchor	This is the View to anchor the auto-complete dropdown to.
android:dropDownHeight	This specifies the basic height of the dropdown.
android:dropDownSelector	This is the selector in a drop down list.
android:dropDownWidth	This specifies the basic width of the dropdown.
android:popupBackground	This sets the background.

Example

In XML:

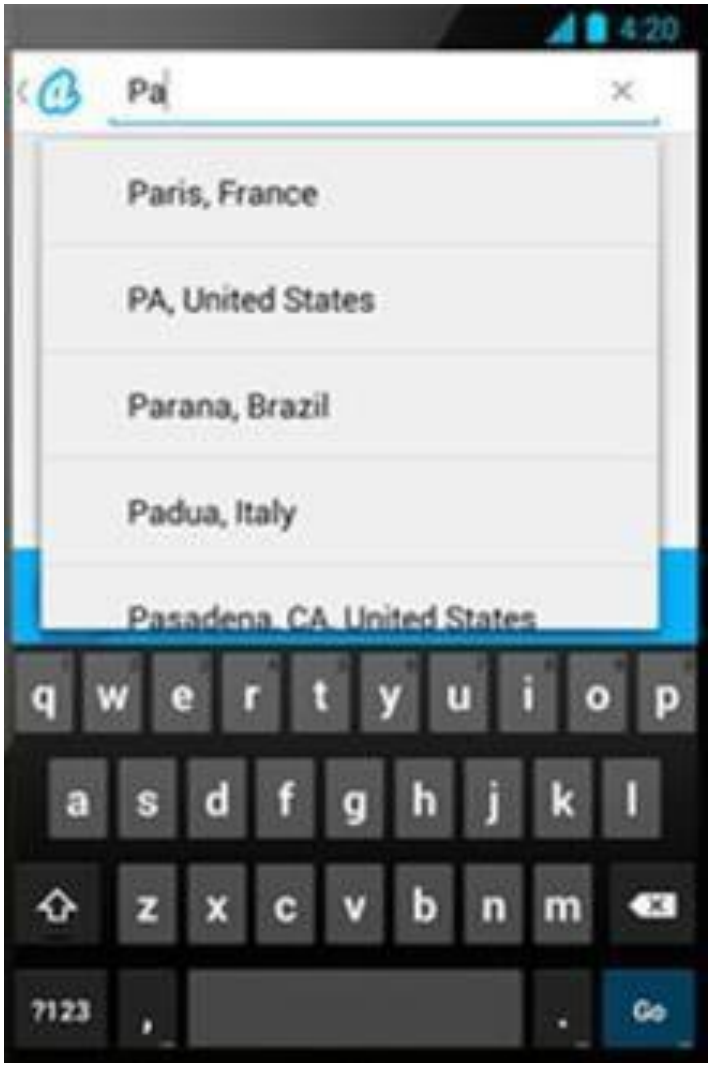
```
<AutoCompleteTextView
android:id="@+id/autoCompleteTextView1  "
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/textView2"
android:layout_below="@+id/textView2"
android:layout_marginTop="54dp"
android:ems="10" />
```

Example (con…)

In JAVA:

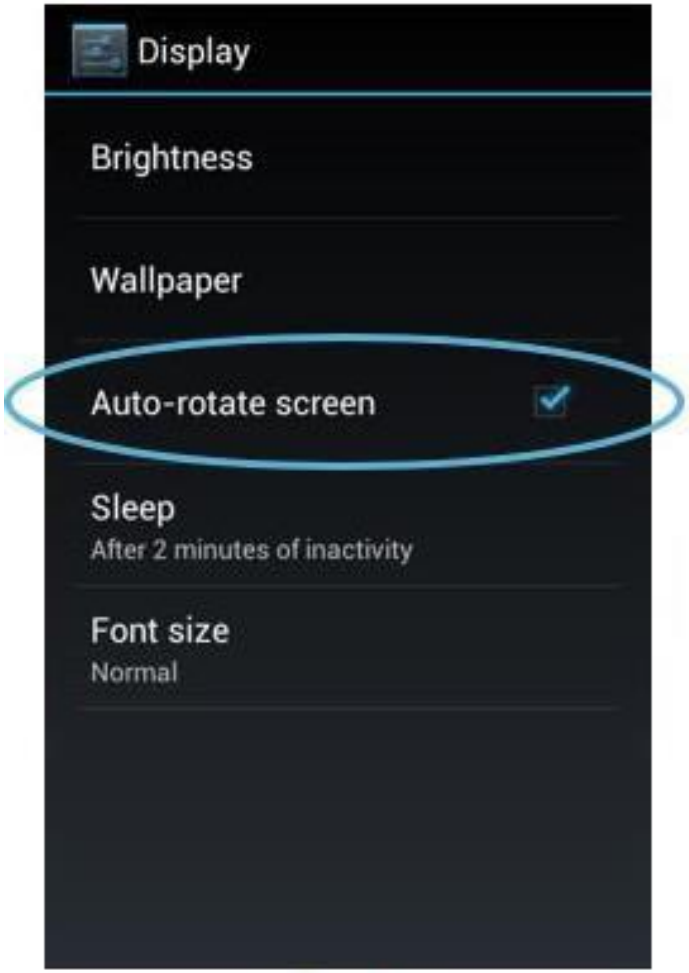
```
AutoCompleteTextView autocompletetextview; String[]
arr = { "Paries,France", "PA,United
States","Parana,Brazil", "Padua,Italy",
"Pasadena,CA,United States"};
autocomplete = (AutoCompleteTextView)
findViewById(R.id.autoCompleteTextView1);
ArrayAdapter<String> adapter = new
ArrayAdapter<String>
(this,android.R.layout.select_dialog_item, arr);
autocomplete.setThreshold(2);
autocomplete.setAdapter(adapter);
```

Output


4:20
Pa
Paris, France
PA, United States
Parana, Brazil
Padua, Italy
Pasadena, CA, United States
q w e r t y u i o p
a s d f g h j k l
z x c v b n m
?123
Go

CheckBox Control

- A CheckBox is an on/off switch that can be toggled by the user.
- To use check-boxes when presenting users with a group of selectable options that are not mutually exclusive.



CHECKBOX

Attributes

Attribute	Description
android:autoText	If set, specifies that this TextView has a textual input method and automatically corrects some common spelling errors.
android:drawableBottom	This is the drawable to be drawn below the text.
android:drawableRight	This is the drawable to be drawn to the right of the text.
android:editable	If set, specifies that this TextView has an input method.
android:text	This is the Text to display.

Example

In XML:

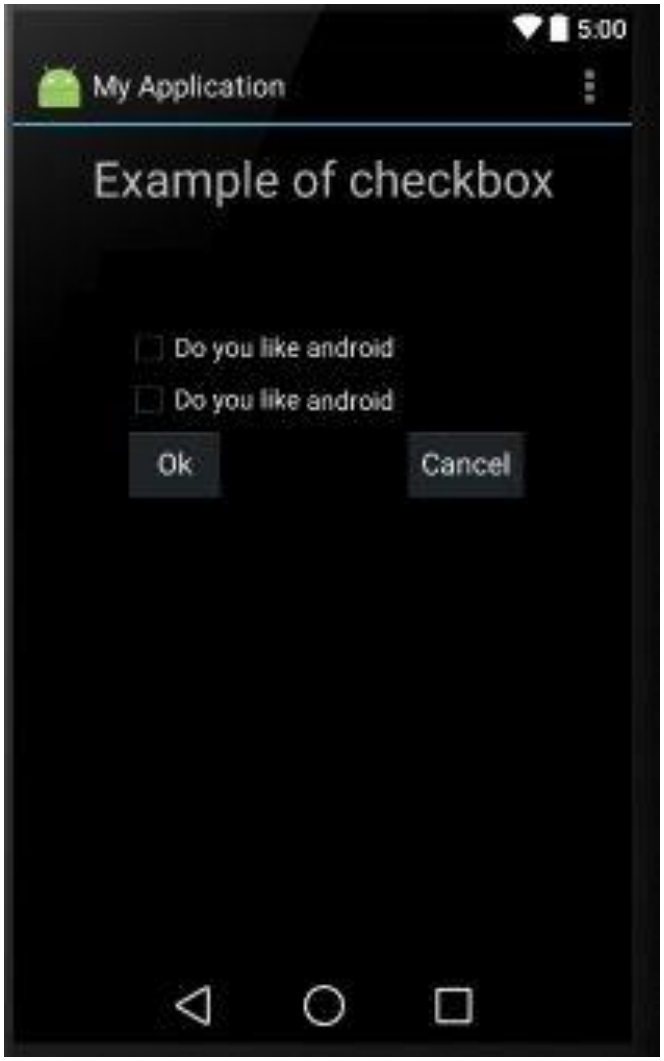
```
<CheckBox android:id="@+id/checkBox1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Do you like android“
android:checked="false“ />
<CheckBox android:id="@+id/checkBox2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Do you like android "
android:checked="false“ />
```

Example (con…)

In JAVA:

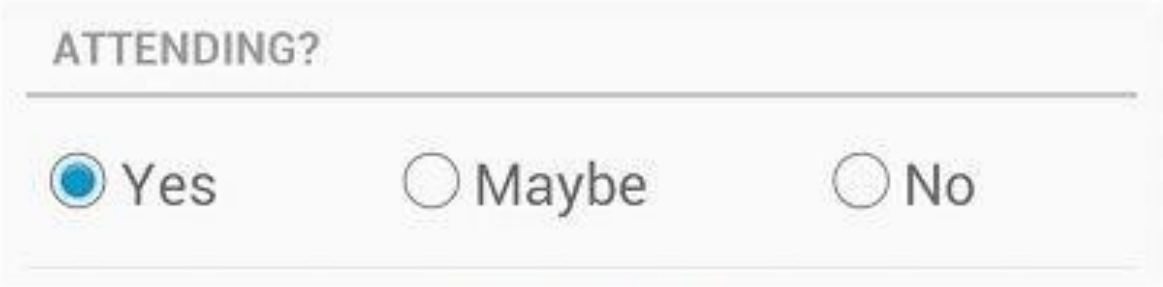
```
CheckBox ch1,ch2; Button b1,b2;
ch1=(CheckBox)findViewById(R.id.checkBox1);
ch2=(CheckBox)findViewById(R.id.checkBox2);
b1=(Button)findViewById(R.id.button);
b1.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {  StringBuffer result = new
StringBuffer();
result.append("Thanks : ").append(ch1.isChecked());
result.append("\nThanks: ").append(ch2.isChecked());
Toast.makeText(MainActivity.this, result.toString(),
Toast.LENGTH_LONG).show(); }
});
```

Output


5:00
My Application
Example of checkbox
Do you like android
Do you like android
Ok
Cancel

RadioButton Control

- A RadioButton has two states: either checked or unchecked.
- This allows the user to select one option from a set.



RADIO BUTTON

Example

In XML:

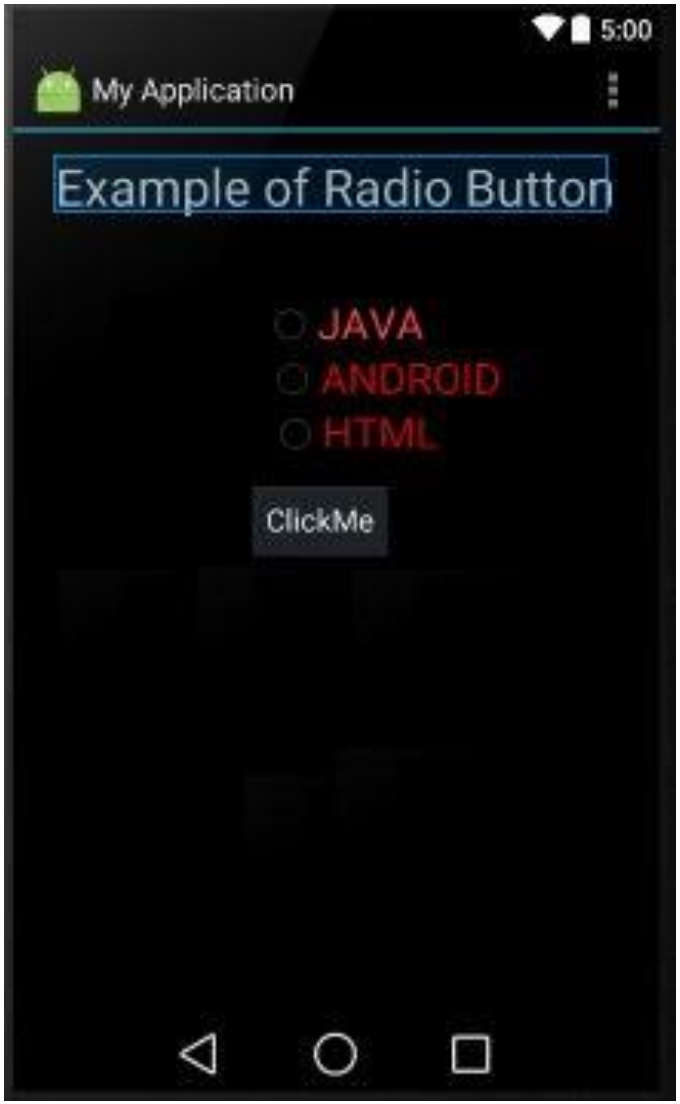
```
<RadioGroup
   <RadioButton
        android:text="JAVA"
        android:id="@+id/radioButton1"
        android:checked="false“ />
        <RadioButton
        android:text="ANDROID"
        android:id="@+id/radioButton2“
        android:checked="false“ />
        <RadioButton
        android:text="HTML"
        android:id="@+id/radioButton
3“
        android:checked="false“ />
</RadioGroup>
```

Example (con…)

In JAVA:

```
RadioButton rb1;     RadioGroup rg1;     Button b1;
addListenerRadioButton();
private void addListenerRadioButton() {
rg1 = (RadioGroup) findViewById(R.id.radioGroup);
b1 = (Button) findViewById(R.id.button1);
b1.setOnClickListener(new View.OnClickListener() {
@Override public void onClick(View v) {
int selected=rg1.getCheckedRadioButtonId();
rb1=(RadioButton)findViewById(selected);
Toast.makeText(MainActivity.this,rb1.getText(),Toast.LE
NGTH_LONG).show(); } }); }
```

Output


5:00
My Application
Example of Radio Button
JAVA
ANDROID
HTML
ClickMe

RadioGroup Control

- A RadioGroup class is used for set of radio buttons.
- If we check one radio button that belongs to a radio group, it automatically unchecks any previously checked radio button within the same group.

(Refer RadioButton)

Progress Bar Control

- Progress bars are used to show progress of a task.
- A class called ProgressDialog that allows you to create progress bar.
- Syntax:

```
ProgressDialog progress = new
    ProgressDialog(this);
```

- For example, when you are uploading or downloading something from the internet, it is better to show the progress of download/upload to the user.

ProgressDialog class methods

Methods	Description
getMax()	This method returns the maximum value of the progress.
incrementProgressBy(int diff)	This method increments the progress bar by the difference of value passed as a parameter.
setIndeterminate(boolean indeterminate)	This method sets the progress indicator as determinate or indeterminate.
setMax(int max)	This method sets the maximum value of the progress dialog.
setProgress(int value)	This method is used to update the progress dialog with some specific value.
show(Context context, CharSequence title, CharSequence message)	This is a static method, used to display progress dialog.

Example

In XML:

```
<Button
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="DOWNLOAD"
android:onClick="download"
android:id="@+id/button1"
android:layout_marginLeft="125dp"
android:layout_marginStart="125dp"
android:layout_centerVertical="true" />
```

Example (con…)

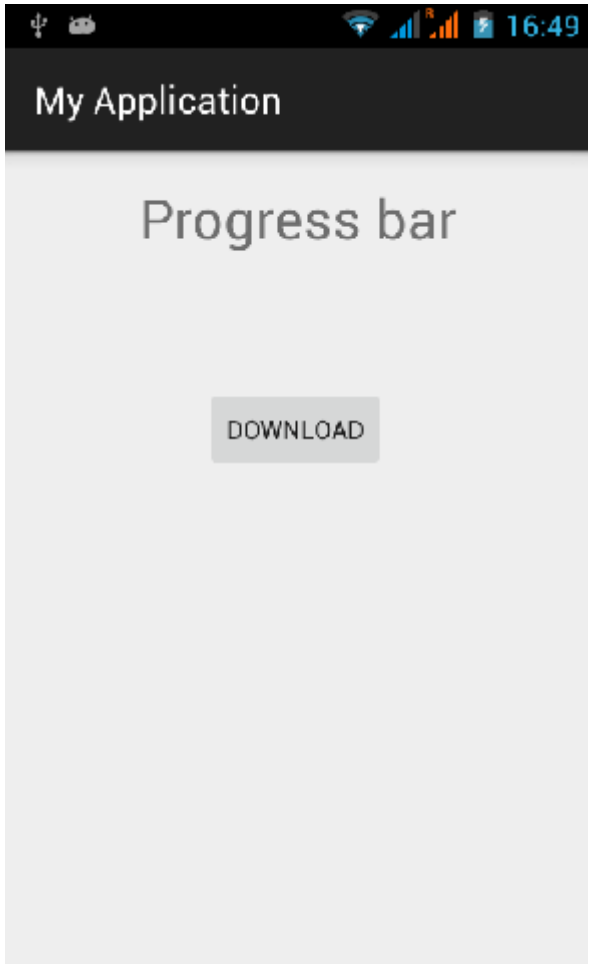
In JAVA:

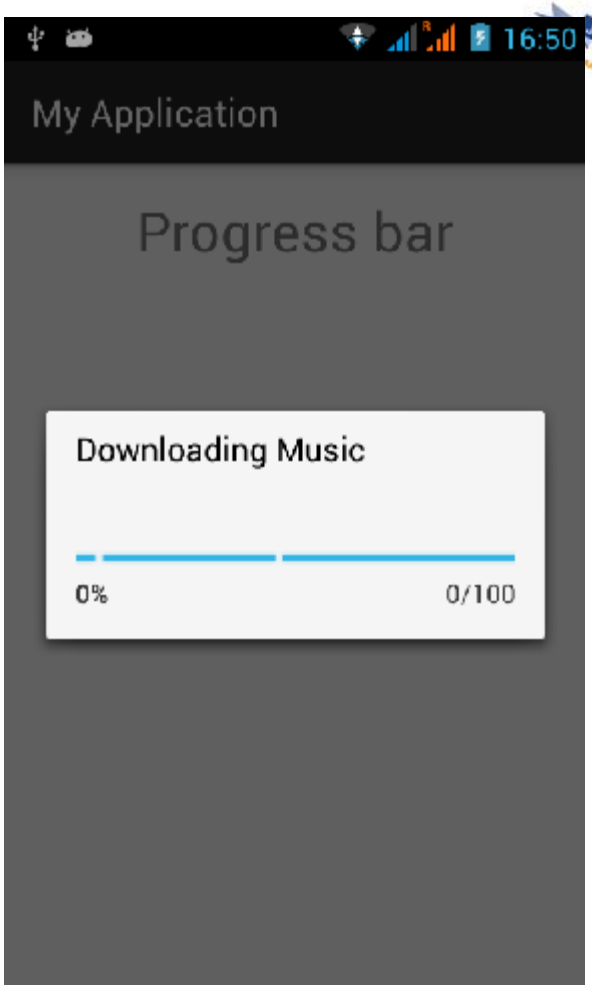
```
Button b1;
private ProgressDialog progress;
b1 = (Button) findViewById(R.id.button1);  public void
download(View view){
progress=new ProgressDialog(this);
progress.setMessage("Downloading Music");
progress.setProgressStyle(ProgressDialog.STYLE_HORI
ZONTAL); progress.setIndeterminate(true);
progress.setProgress(0);
progress.show();
```

Example (con…)

```
final int totalProgressTime = 100;  final Thread t = new
    Thread() {  @Override public void run() {
int jumpTime = 0;
while(jumpTime < totalProgressTime) {  try {
sleep(200);
jumpTime += 5;  progress.setProgress(jumpTime); }
catch (InterruptedException e) { }
} } };
t.start(); }
```

Output


16:49
My Application
Progress bar
DOWNLOAD


16:50
My Application
Progress bar
Downloading Music
0%
0/100

Spinner Control

- Spinner allows you to select an item from a drop down menu.

Example

In XML:

```
<Spinner android:id="@+id/spinner"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:prompt="@string/spinner_title"/>
```

In JAVA:

```
Spinner spinner = (Spinner) findViewById(R.id.spinner);
spinner.setOnItemSelectedListener(this);
List<String> categories = new ArrayList<String>();
categories.add(“Automobile");
categories.add(“Business Services");
```

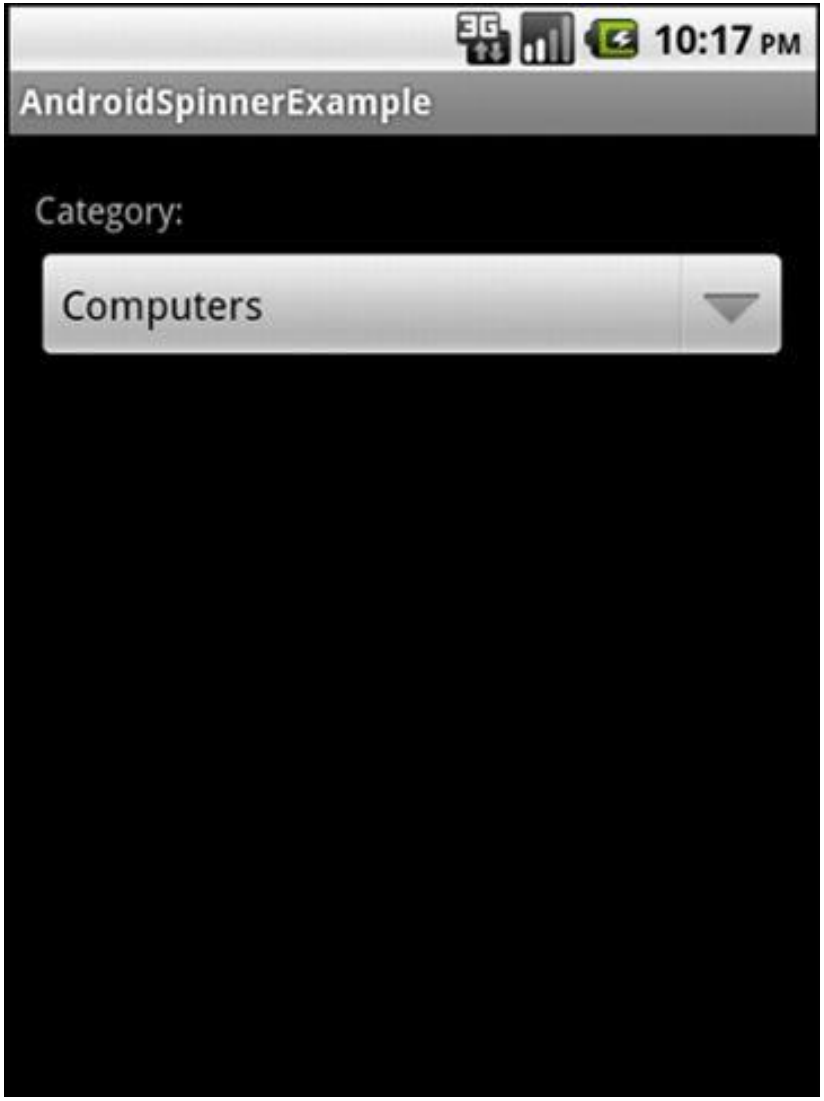
Example (con…)

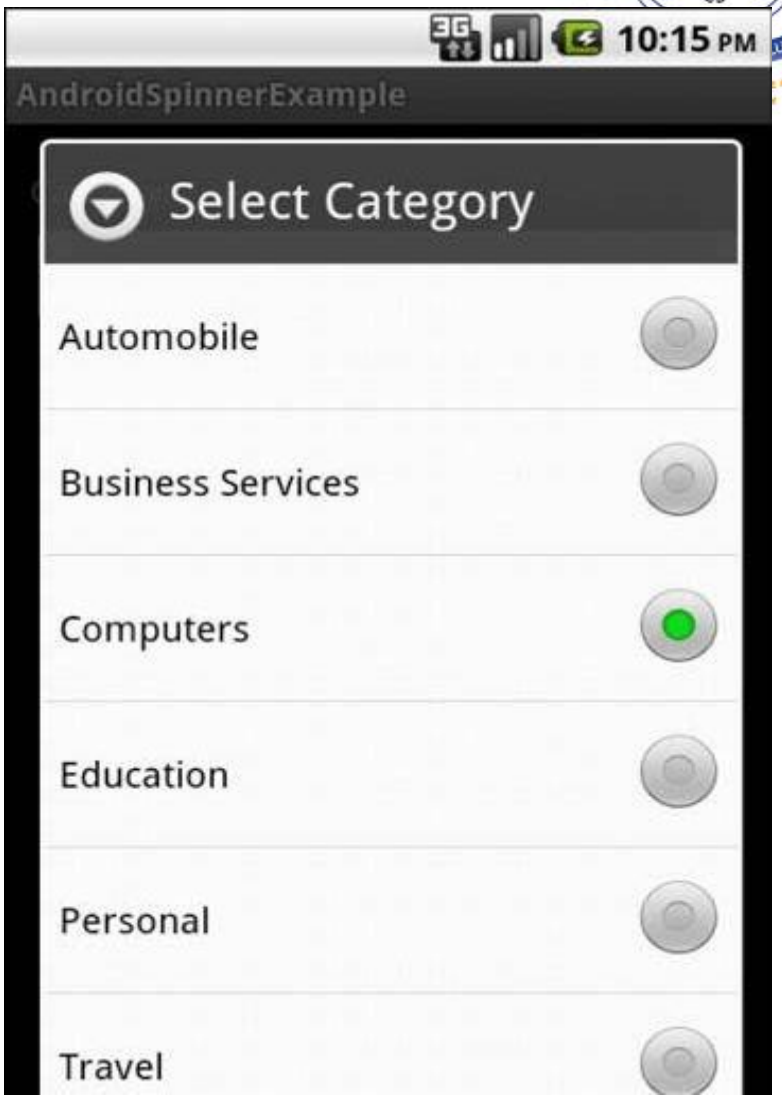
```
categories.add("Computers");
categories.add("Education");
categories.add("Personal");  categories.add("Travel");
ArrayAdapter<String> dataAdapter = new
ArrayAdapter<String>(this,
android.R.layout.simple_spinner_item, categories);
dataAdapter.setDropDownViewResource(android.R.layo
ut.
simple_spinner_dropdown_item);
spinner.setAdapter(dataAdapter);
```

Example (con…)

```
public void onItemSelected(AdapterView<?>
   parent, View view, int position, long id) {
String item =
   parent.getItemAtPosition(position).toString();
Toast.makeText(parent.getContext(), "Selected: "
+ item, Toast.LENGTH_LONG).show();
}
```

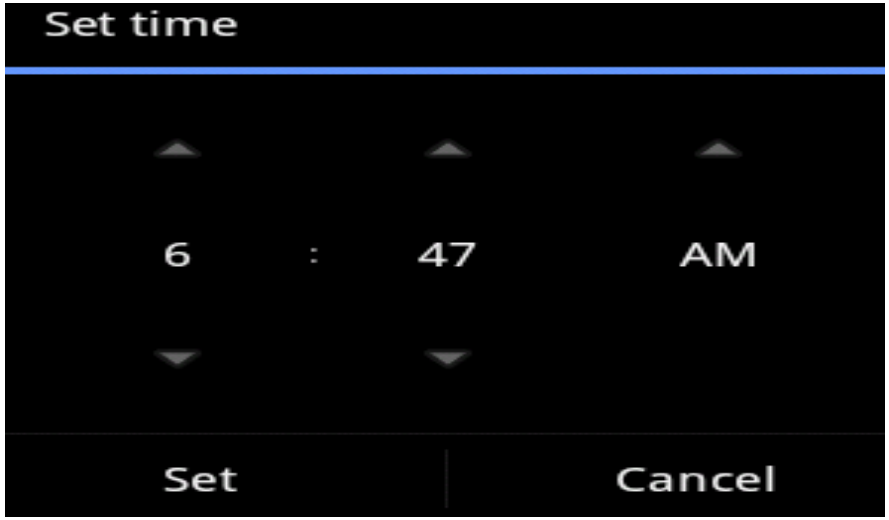
Output


10:17 PM
AndroidSpinnerExample
Category:
Computers


10:15 PM
AndroidSpinnerExample
Select Category
Automobile
Business Services
Computers
Education
Personal
Travel

TimePicker Control

- Time Picker allows you to select the time of day in either 24 hour or AM/PM mode.
- The time consists of hours, minutes and clock format.
- Android provides this functionality through TimePicker class.



Methods

Methods	Description
is24HourView()	This method returns true if this is in 24 hour view else false
isEnabled()	This method returns the enabled status for this view
setCurrentHour(Integer currentHour)	This method sets the current hour
setCurrentMinute(Integer currentMinute)	This method sets the current minute
setEnabled(boolean enabled)	This method set the enabled state of this view
setIs24HourView(Boolean is24HourView)	This method set whether in 24 hour or AM/PM mode
setOnTimeChangedListener(TimePicker.OnTimeChangedListener onTimeChangedListener)	This method Set the callback that indicates the time has been adjusted by the user

Example

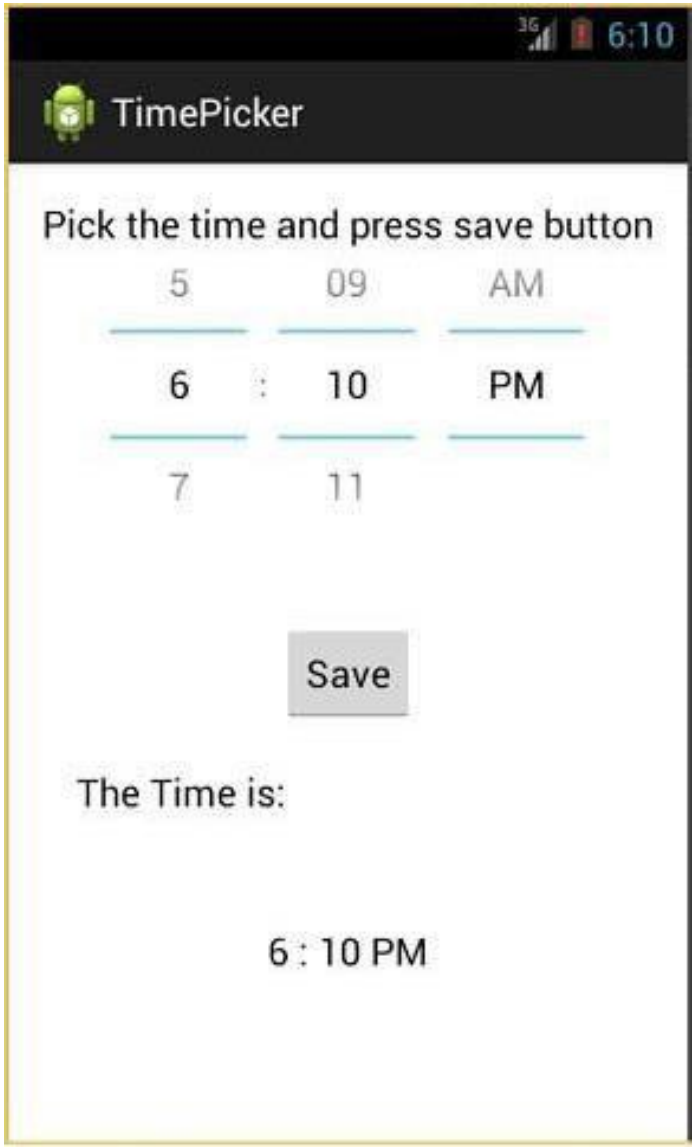
In XML:

```
<TimePicker android:id="@+id/timePicker1"
android:layout_width="wrap_content"
android:layout_height="wrap_content" />
```

In JAVA:

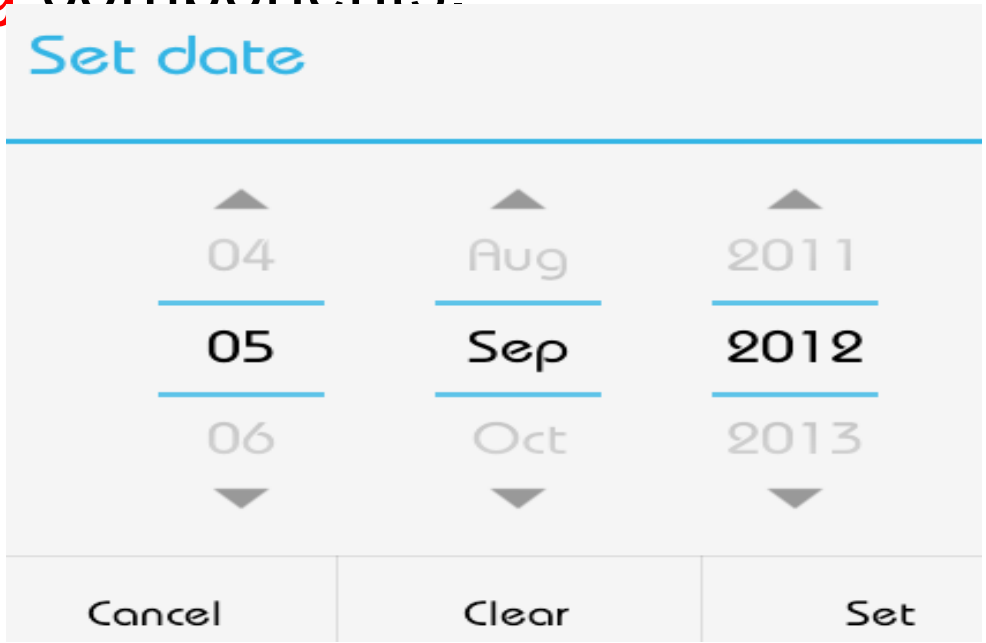
```
TimePicker timePicker1;

timePicker1 = (TimePicker)findViewById(R.id.timePicker1);

int hour = timePicker1.getCurrentHour();

int min = timePicker1.getCurrentMinute();
```

Output


6:10
TimePicker
Pick the time and press save button
5
09
AM
6
:
10
PM
7
11
Save
The Time is:
6 : 10 PM

DatePicker Control

- Date Picker allows you to select the date consisting of day, month and year in your custom user interface.
- Android provides DatePicker and DatePickerDialog components.



Methods

Methods	Description
getDayOfMonth()	This method gets the selected day of month
getMonth()	This method gets the selected month
getYear()	This method gets the selected year
setMaxDate(long maxDate)	This method sets the maximal date supported by this DatePicker in milliseconds
setMinDate(long minDate)	This method sets the minimal date supported by this NumberPicker in milliseconds
setSpinnersShown(boolean shown)	This method sets whether the spinners are shown
updateDate(int year, int month, int dayOfMonth)	This method updates the current date
getCalendarView()	This method returns calendar view
getFirstDayOfWeek()	This Method returns first day of the week

Example

In JAVA:

```
DatePicker datePicker;  Calendar calendar;
int year, month, day;
calendar = Calendar.getInstance();  year =
calendar.get(Calendar.YEAR);
month = calendar.get(Calendar.MONTH);
day = calendar.get(Calendar.DAY_OF_MONTH);
```

8. Event Handling

- Events are a useful way to collect data about a user's interaction with interactive components of Applications.
- Like button presses or screen touch etc.
- The Android framework maintains an event queue as first-in, first-out (FIFO) basis.
- Capture these events in program and take appropriate action as per requirements.

Event Handling (con…)

- Event Management
  - Event Listeners
    - An event listener is an interface in the View class that contains a single callback method.
    - These methods will be called by the Android framework when the View to which the listener has been registered is triggered by user interaction with the item in the UI.

Event Handling (con…)

- Event Handlers
  - When an event happens and we have registered in the event listener for the event, the event listener calls the Event Handlers, which is the method that actually handles the event.
- Event Listeners Registration
  - Event Registration is the process by which an Event Handler gets registered with an Event Listener so that the handler is called when the Event Listener fires the event

Event Listeners & Event Handlers

Event Handler	Event Listener	Description
onClick()	OnClickListener()	This is called when the user either clicks or touches or focuses upon any widget like button, text, image etc.
onLongClick()	OnLongClickListener()	This is called when the user either clicks or touches or focuses upon any widget like button, text, image etc. for one or more seconds.
onFocusChange()	OnFocusChangeListener()	This is called when the widget looses its focus.

Event Listeners & Event Handlers (con…)

onKey()	**OnFocusChangeListener ()**	This is called when the user is focused on the item and presses or releases a hardware key on the device.
onTouch()	**OnTouchListener()**	This is called when the user presses the key, releases the key, or any movement gesture on the screen.
onMenuItemClick()	**OnMenuItemClickListener()**	This is called when the user selects a menu item.
onCreateContextMenu ()	**onCreateContextMenuItemListener()**	This is called when the context menu is being built(as the result of a sustained "long click”).

Event Listeners Registration

- Event Registration is the process by which an Event Handler gets registered with an Event Listener so that the handler is called when the Event Listener fires the event.
- Top 3 ways are,
  - Using an Anonymous Inner Class
  - Activity class implements the Listener interface.
  - Using Layout file activity_main.xml to specify event handler directly.

Example

•Using an Anonymous Inner Class Button b1;

```
b1=(Button)findViewById(R.id.button);

b1.setOnClickListener(new View.OnClickListener()
{ @Override

public void onClick(View v) {

TextView txtView = (TextView)

findViewById(R.id.textView);

txtView.setTextSize(25); } });
```

Example (con…)

- Activity class implements the Listener interface

```
BtnListener listener = new BtnListener();
((Button) findViewById(R.id.btnNum0Id)).setOnClickListener(listener);
private class BtnListener implements OnClickListener { // On-click event handler for all the buttons @Override public void onClick(View view) {
//ToDo the code here….
} }
```

Example (con…)

- Using Layout file activity_main.xml to specify event handler directly
- In XML

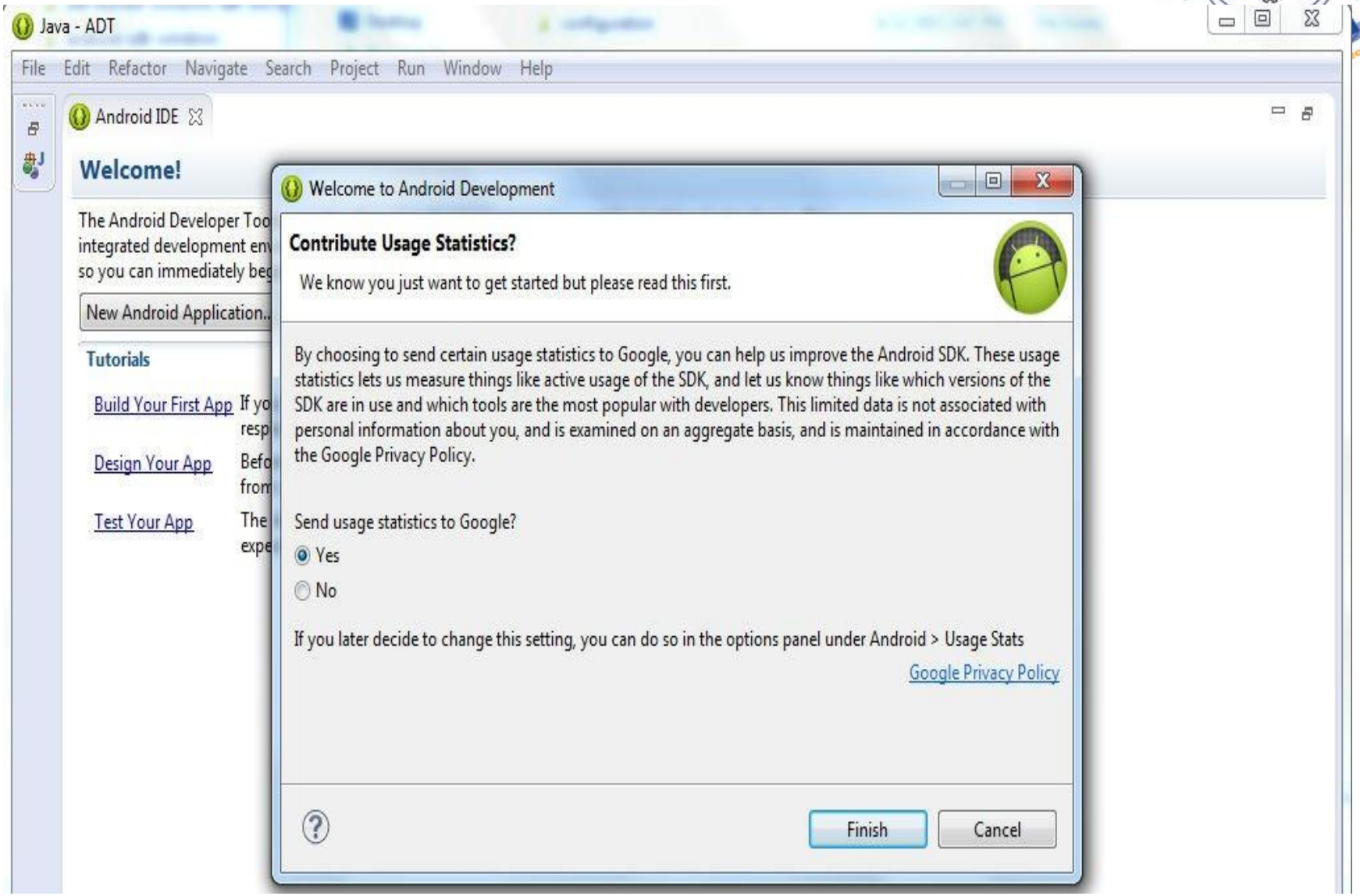
```
<Button
android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:text="Small font"
android:id="@+id/button“
   android:onClick=“Font_Change”/>
```

Example (con…)

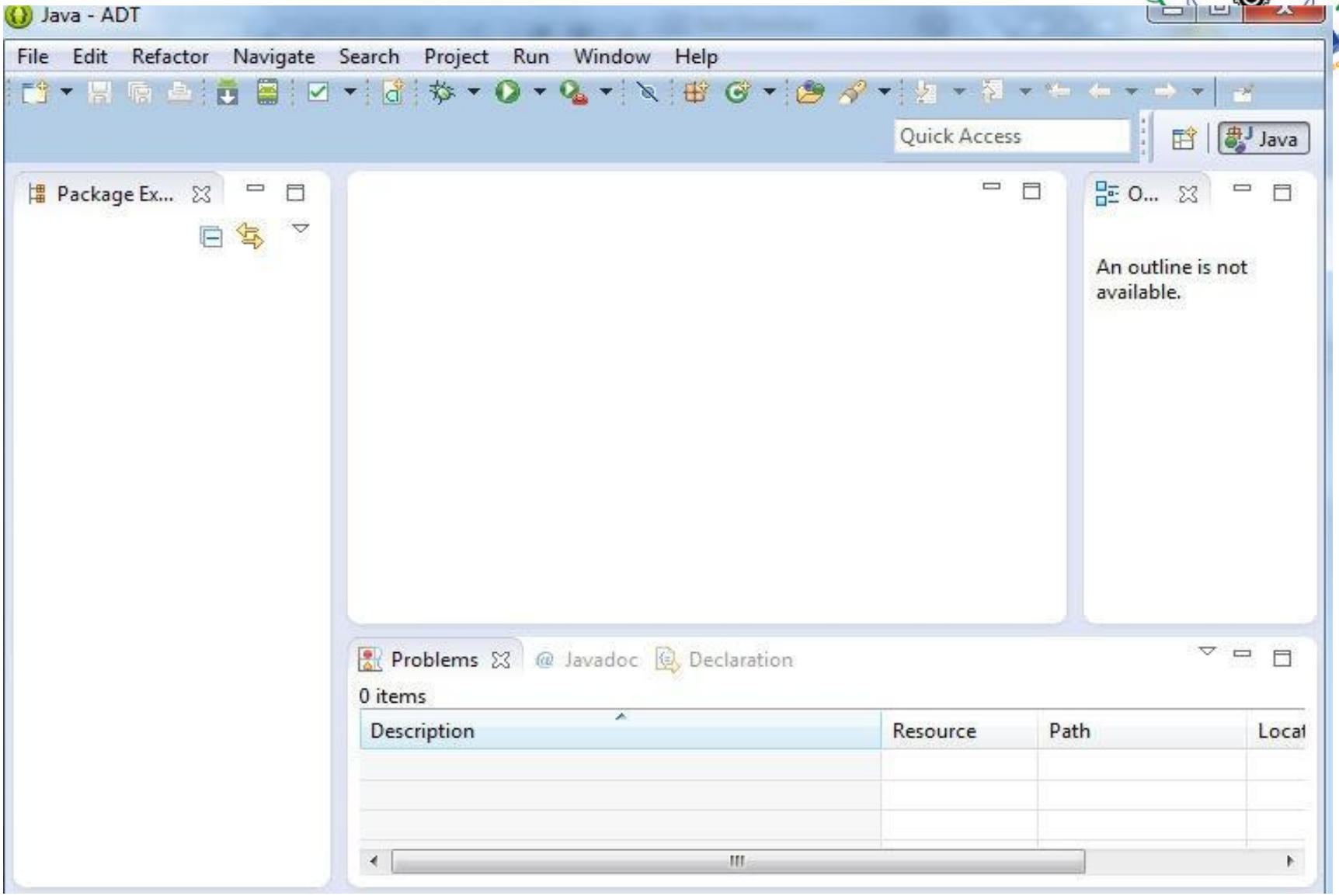
- In JAVA

```
public void Font_Change(View v) {
   TextView txtView = (TextView)
findViewById(R.id.textView);
txtView.setTextSize(25);
}
```

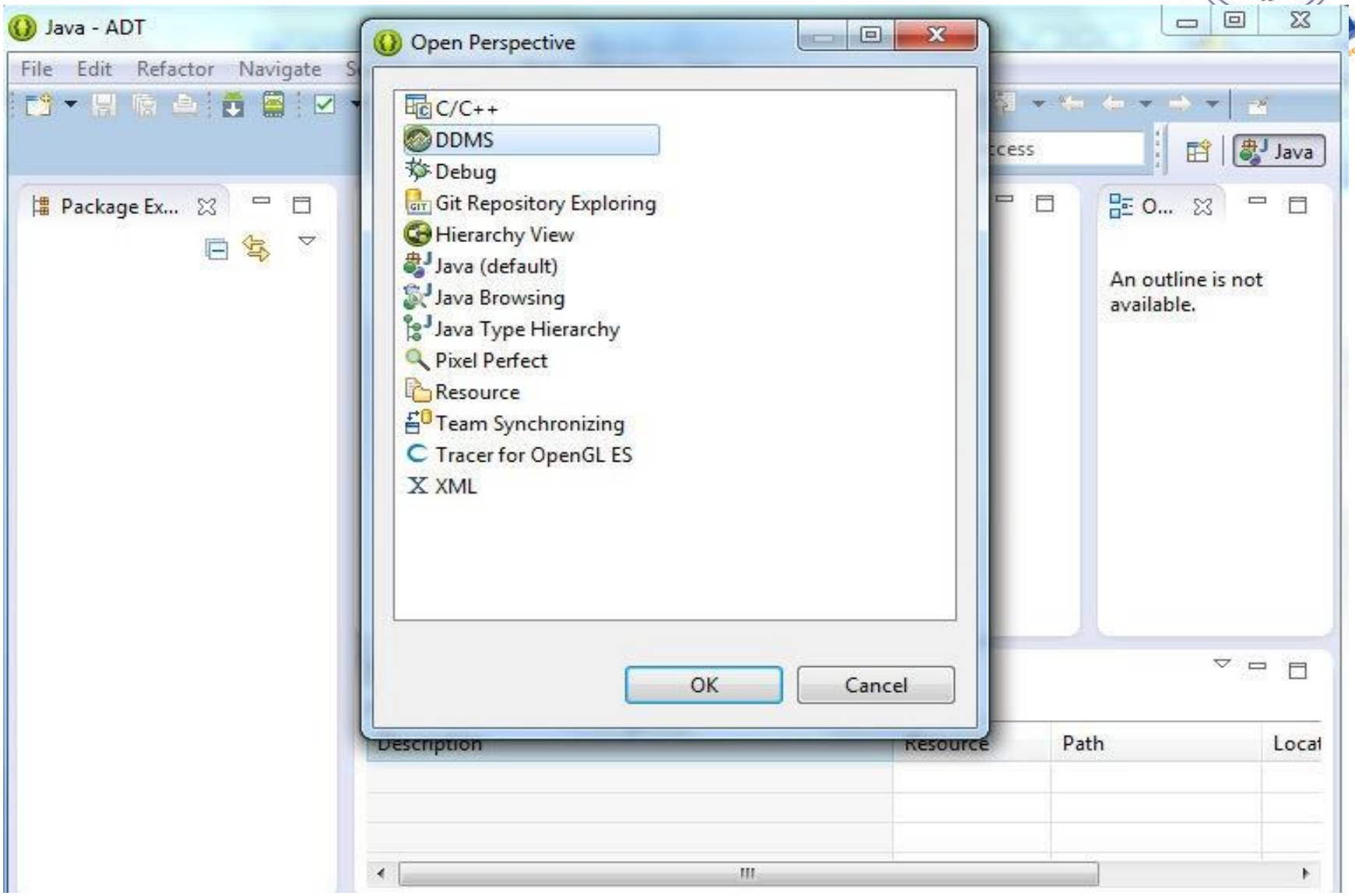
9. Tools - Eclipse IDE



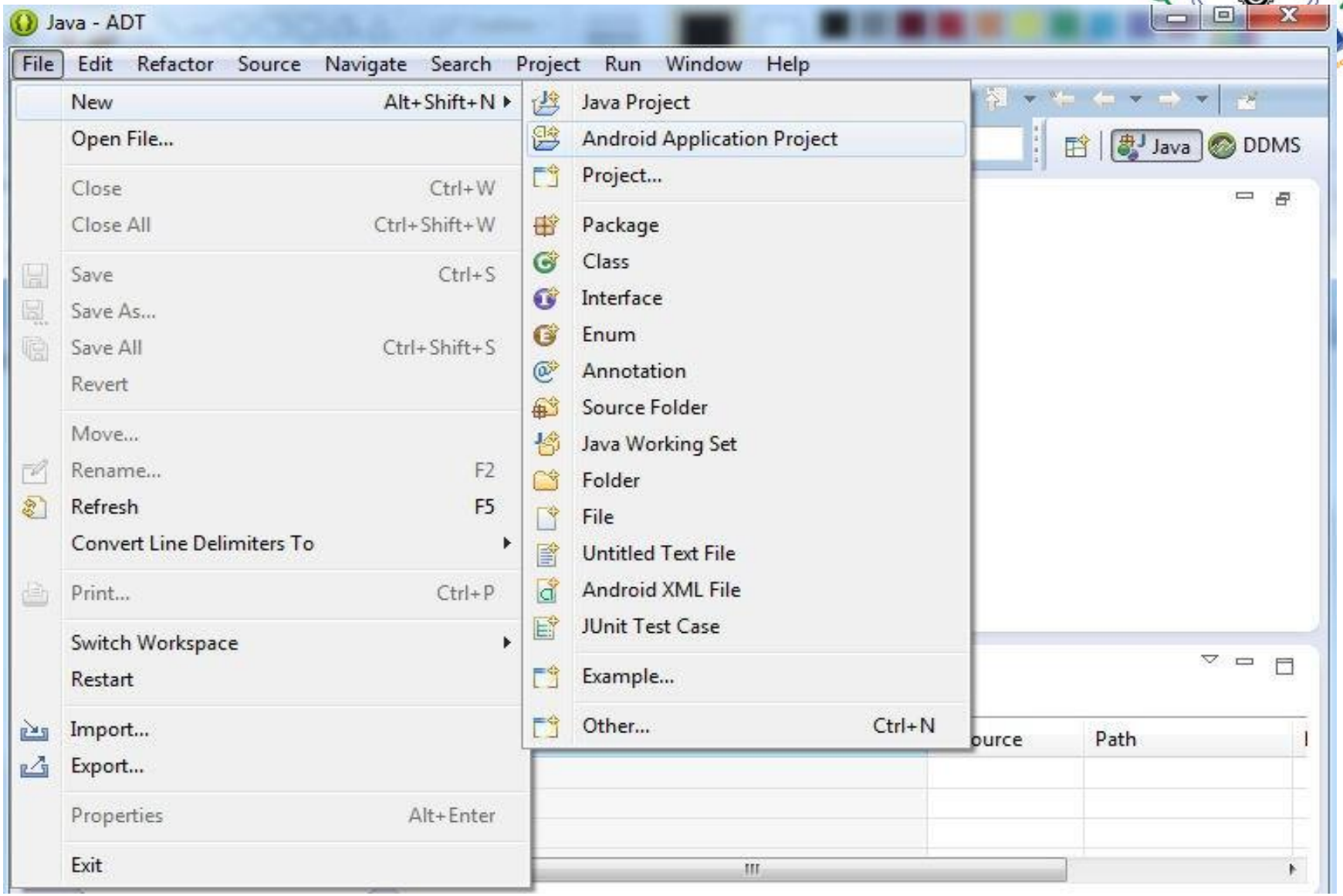
Eclipse IDE (con…)



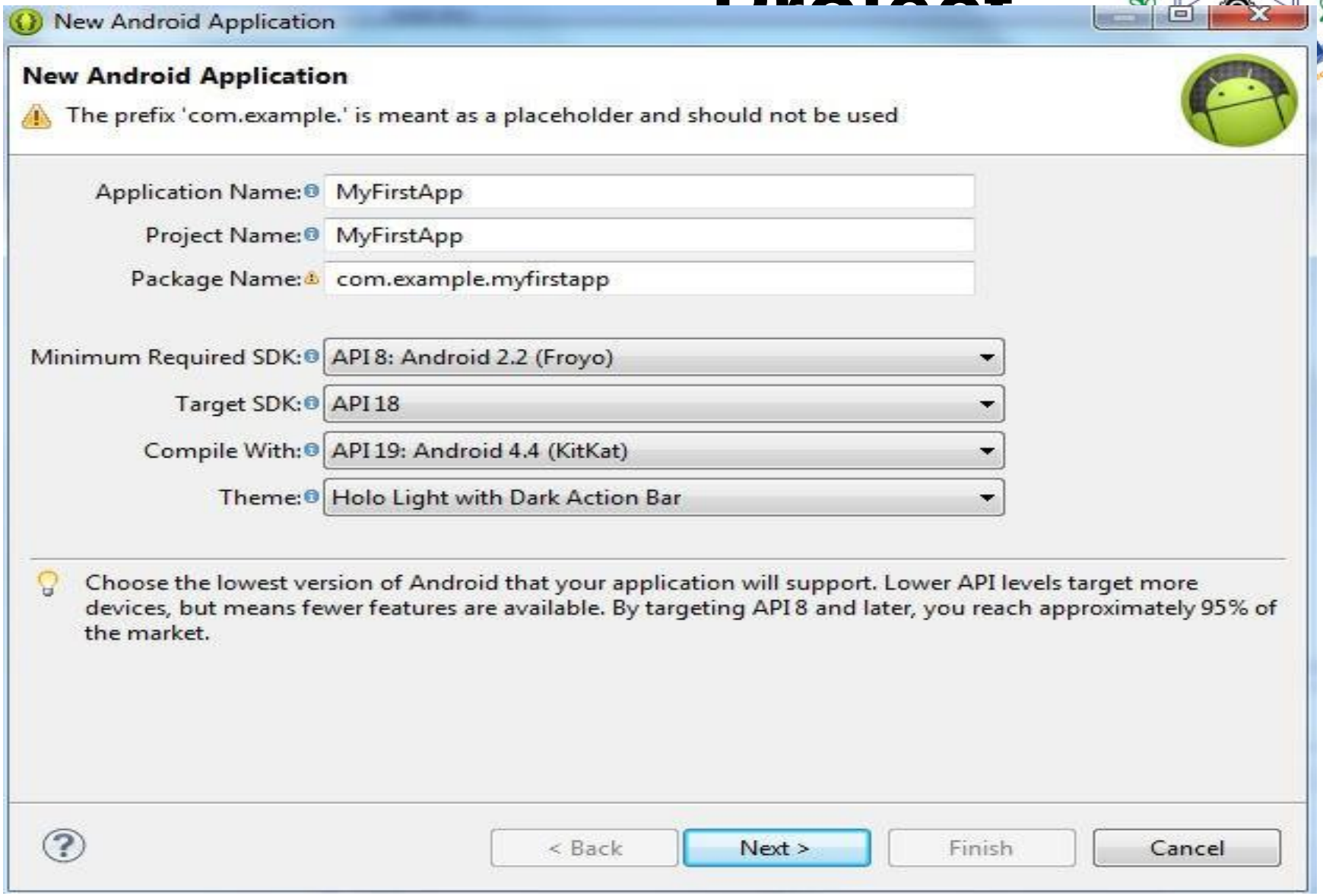
DDMS Configuration



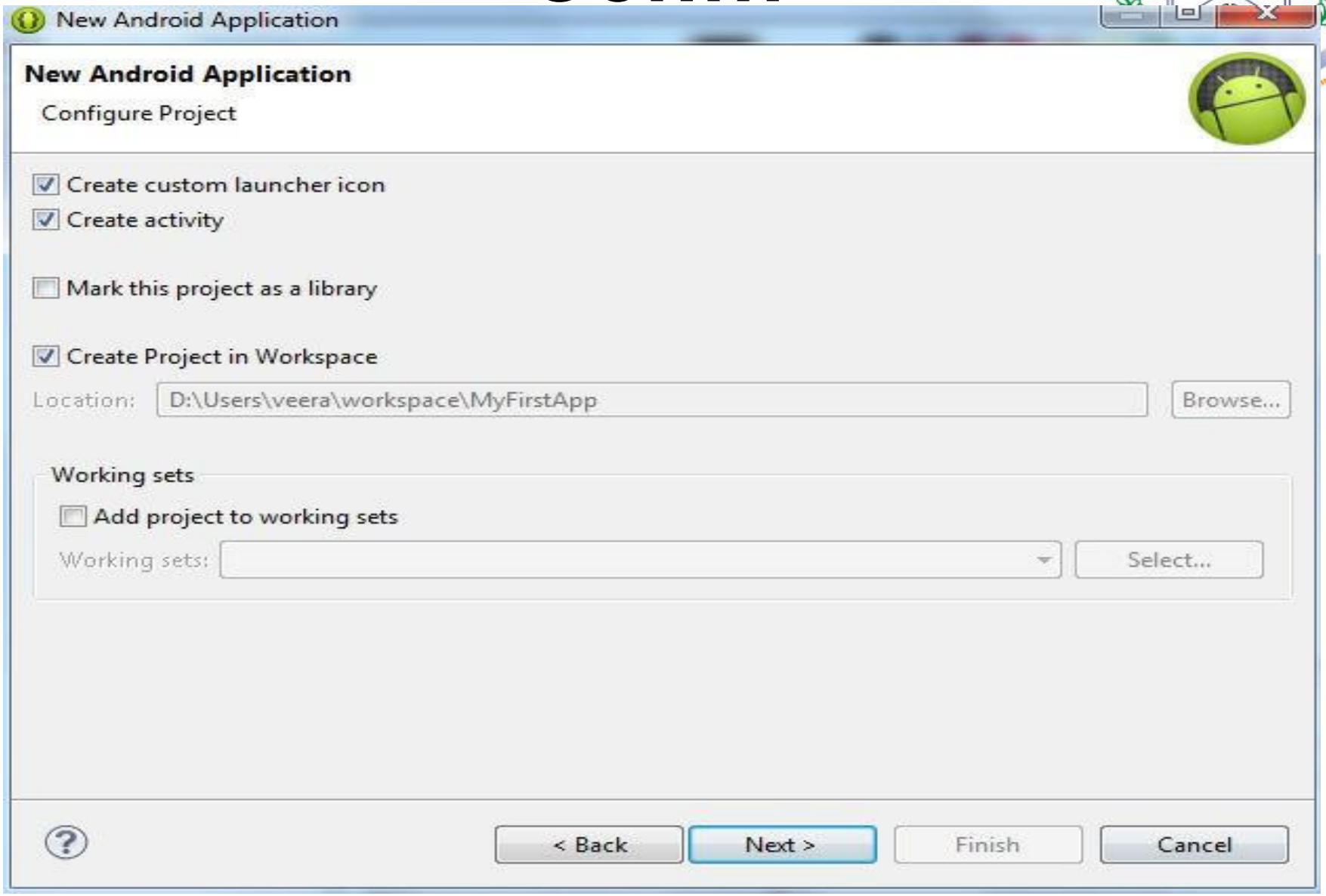
New Android Project Creation



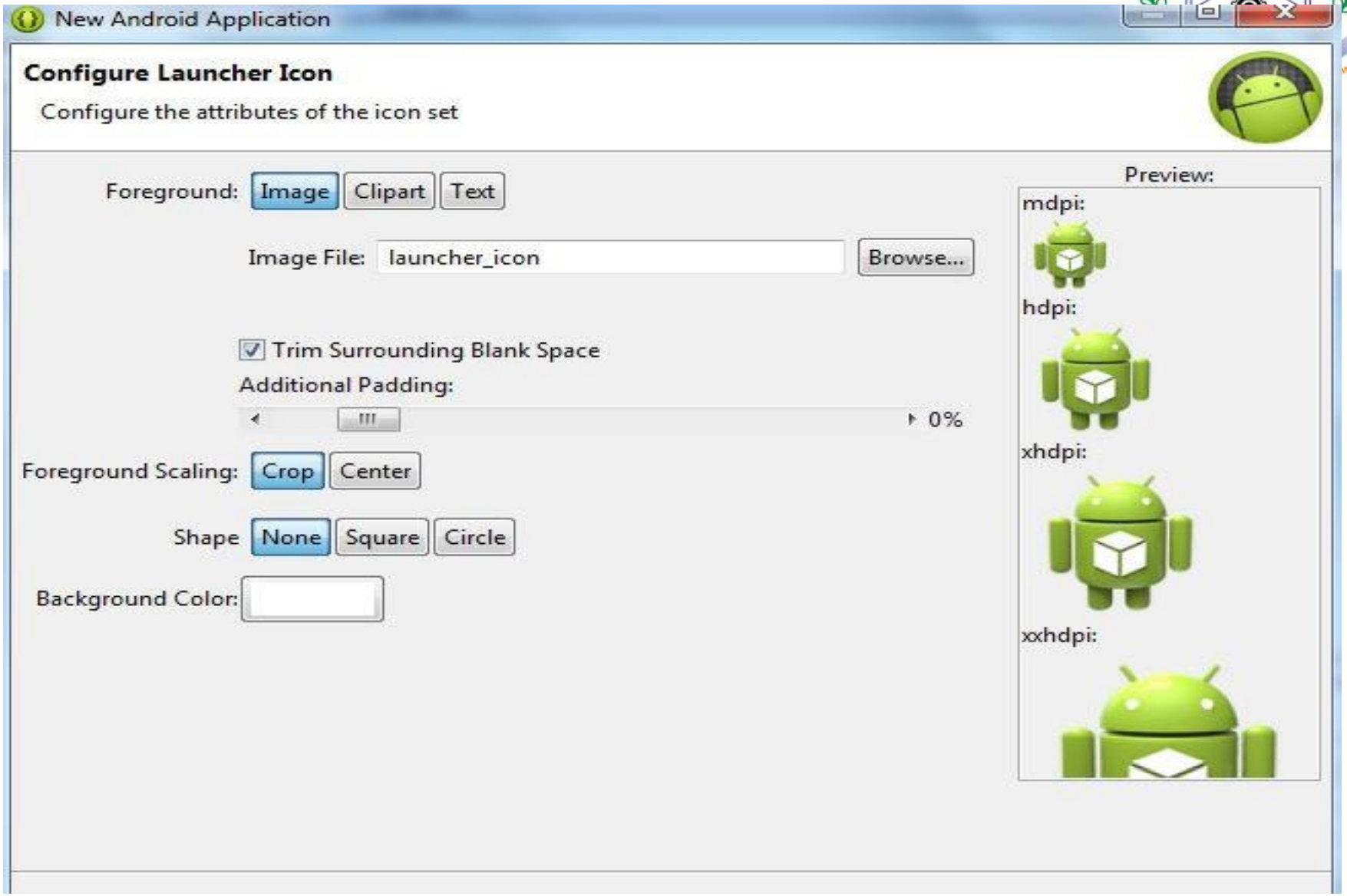
Giving Name Application / Project



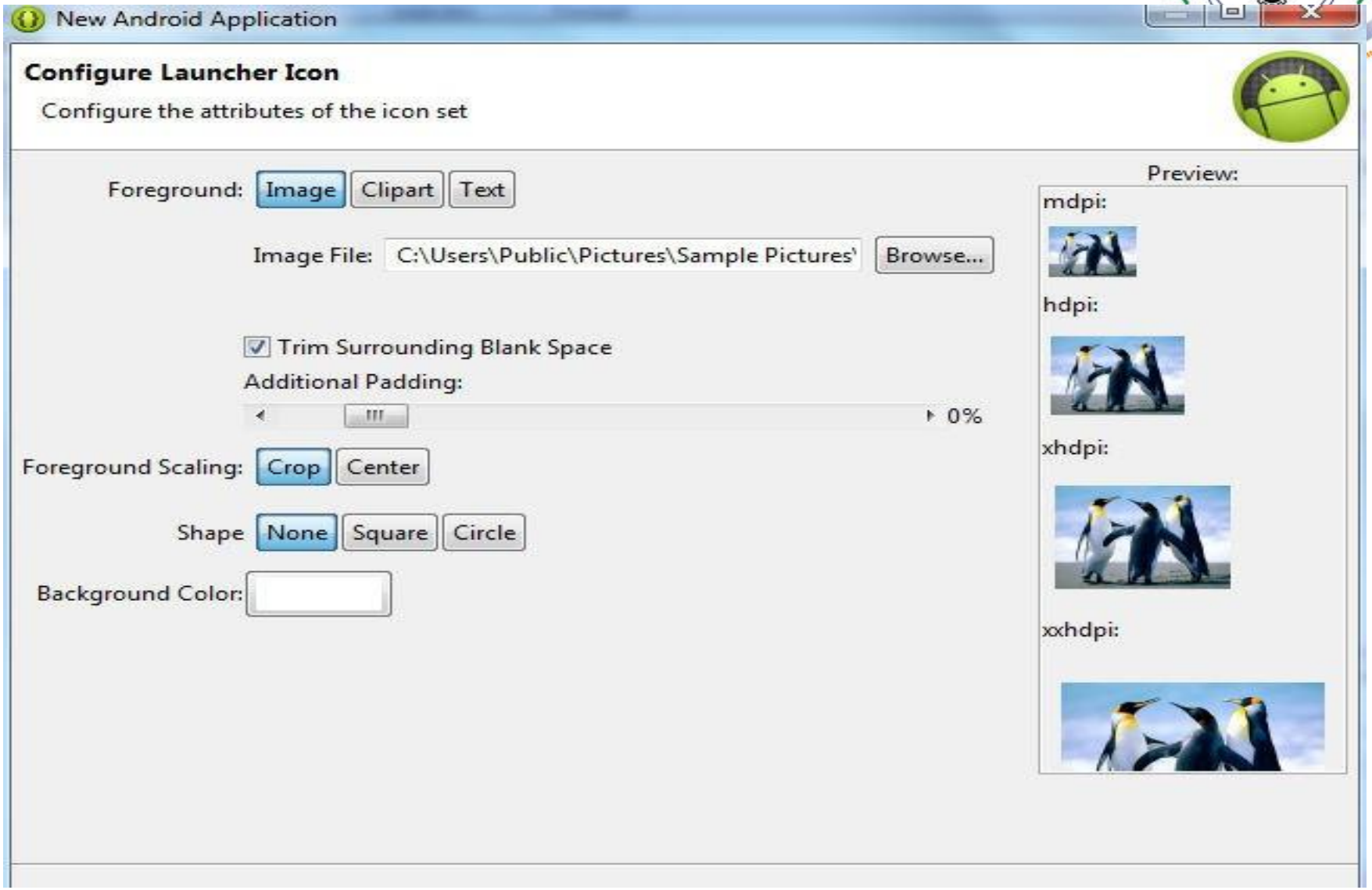
Con…


New Android Application
New Android Application
Configure Project
Create custom launcher icon
Create activity
Mark this project as a library
Create Project in Workspace
Location:
D:\Users\veera\workspace\MyFirstApp
Browse...
Working sets
Add project to working sets
Working sets:
Select...
< Back
Next >
Finish
Cancel

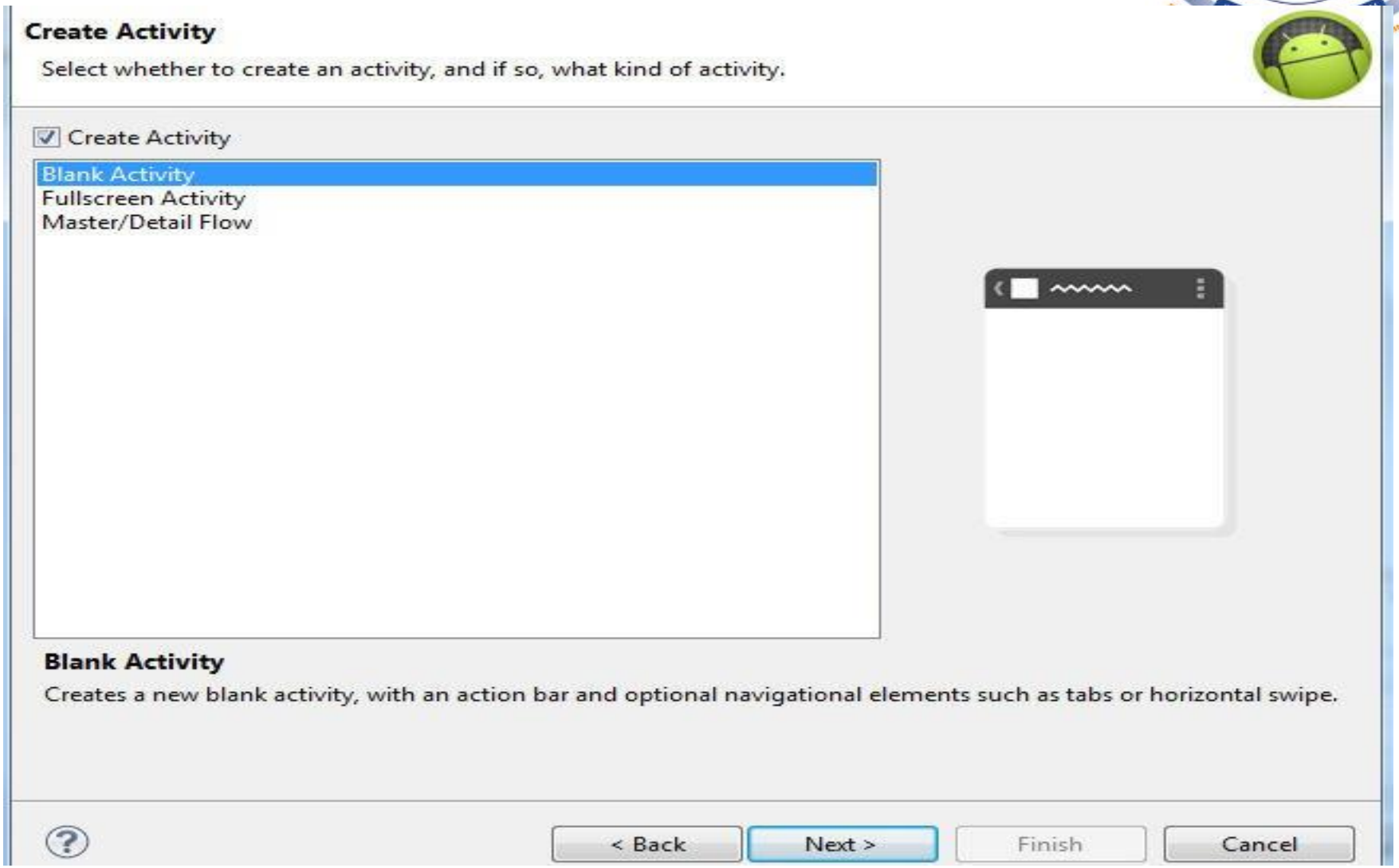
Icon Customization



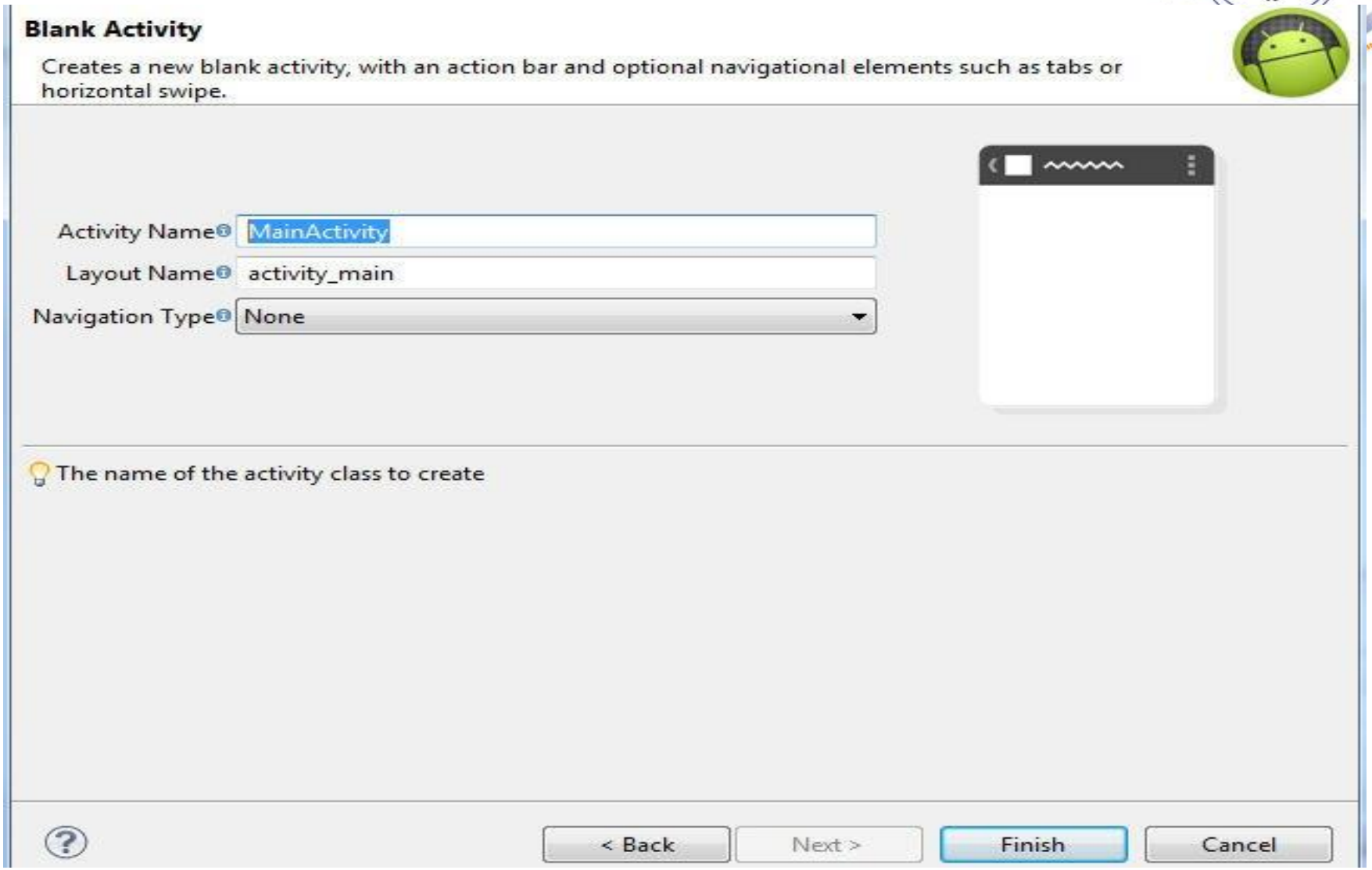
Customized Icon



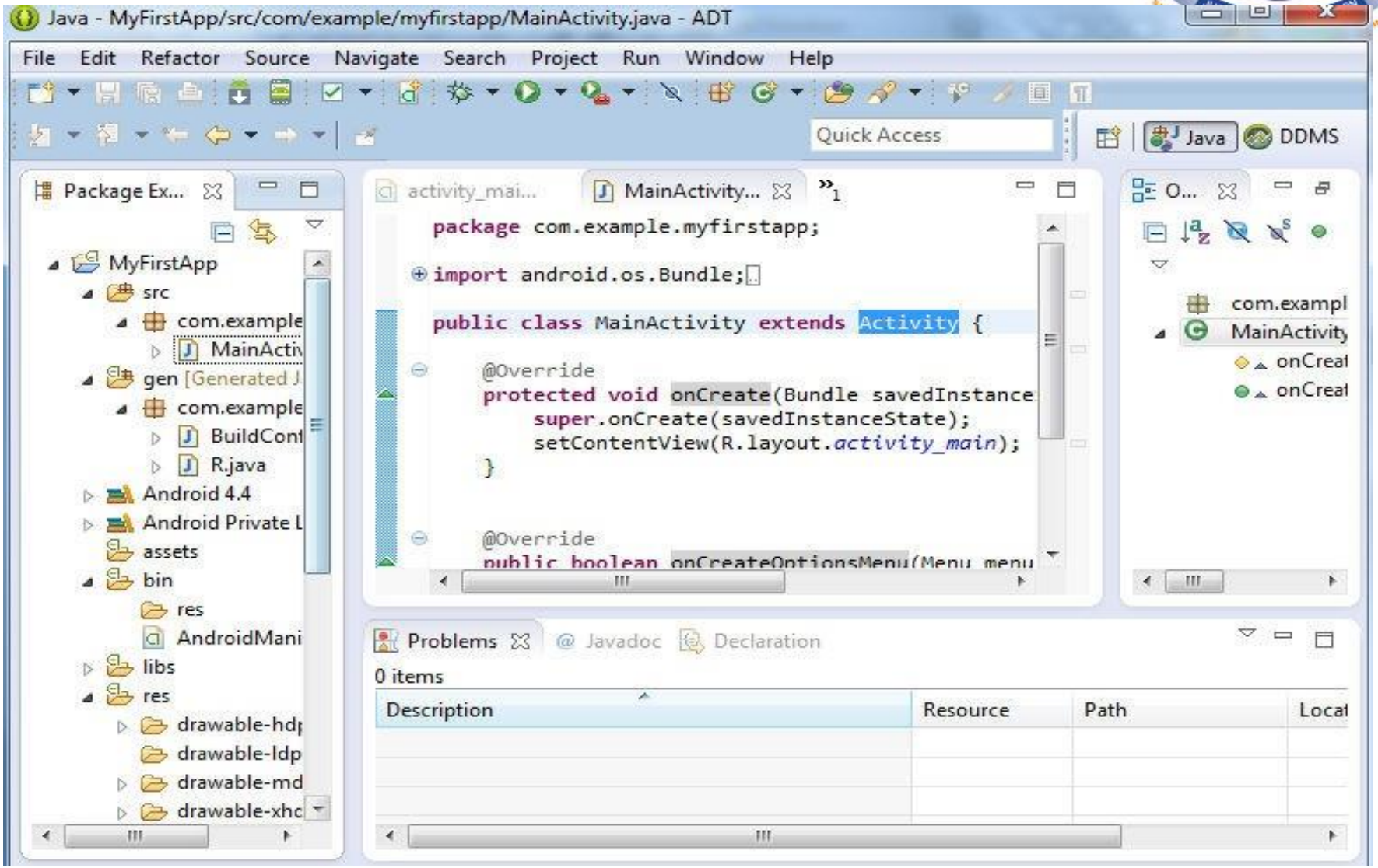
Activity type selection



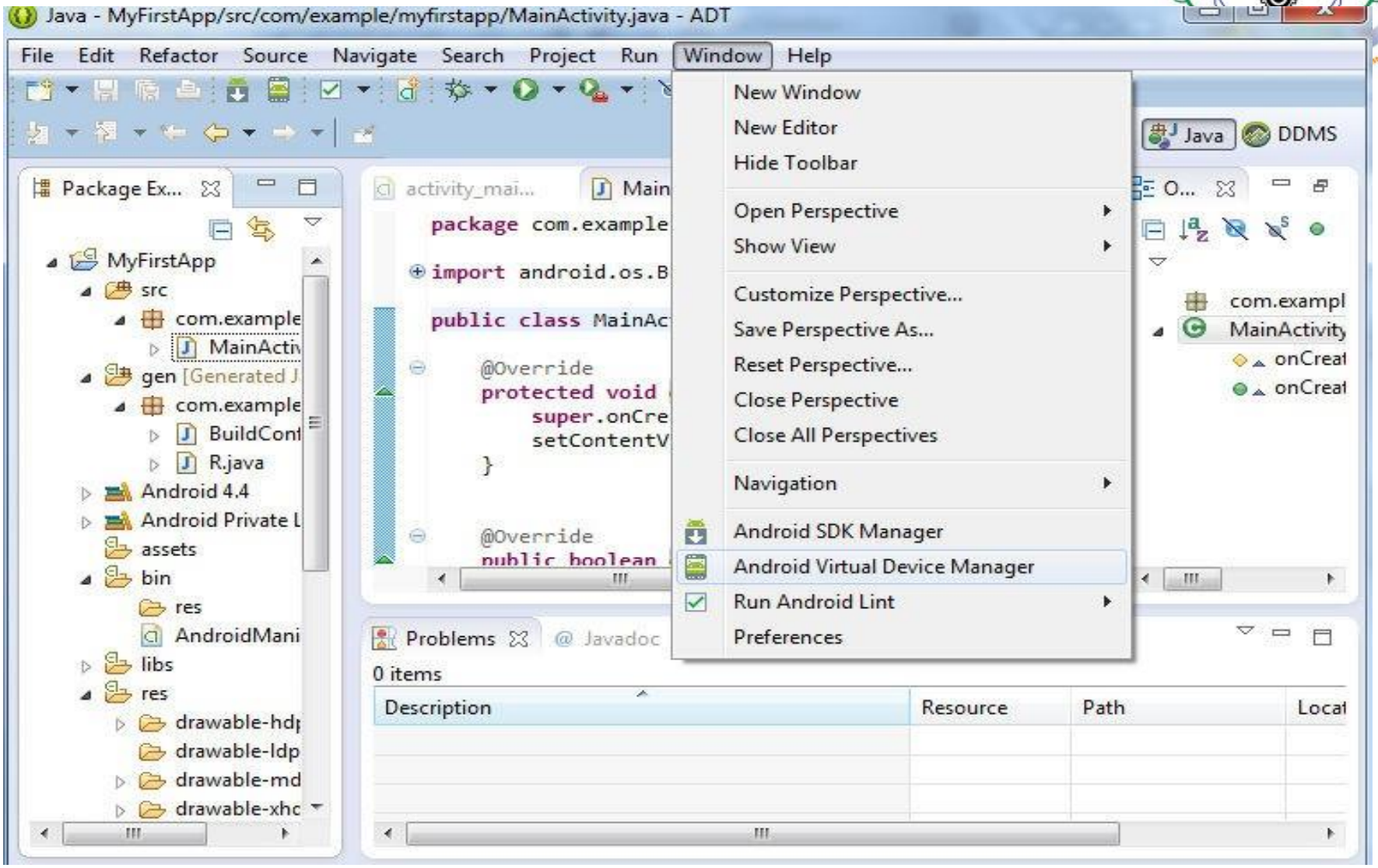
Customize the activity name



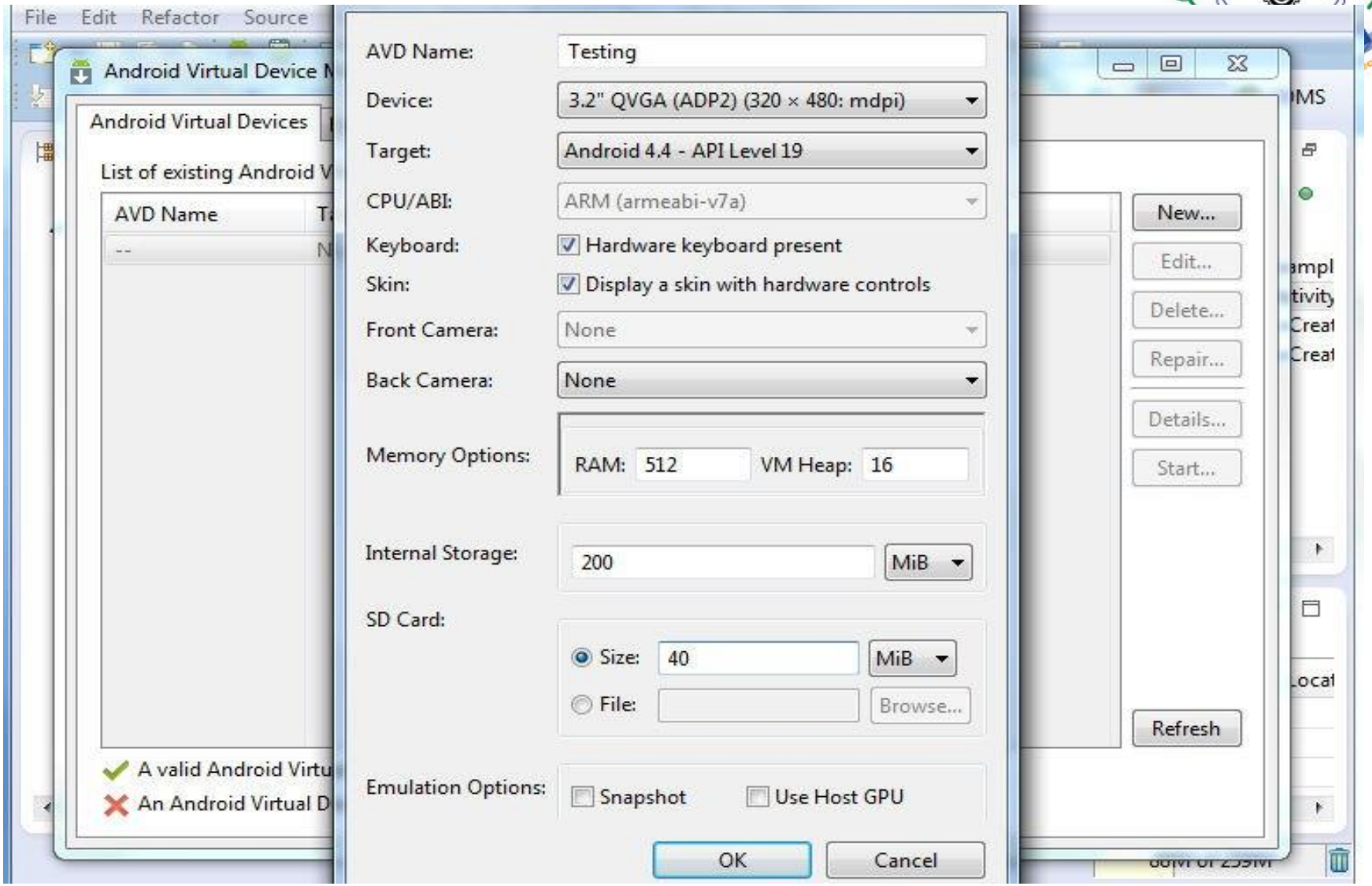
Default code appear in Eclipse IDE



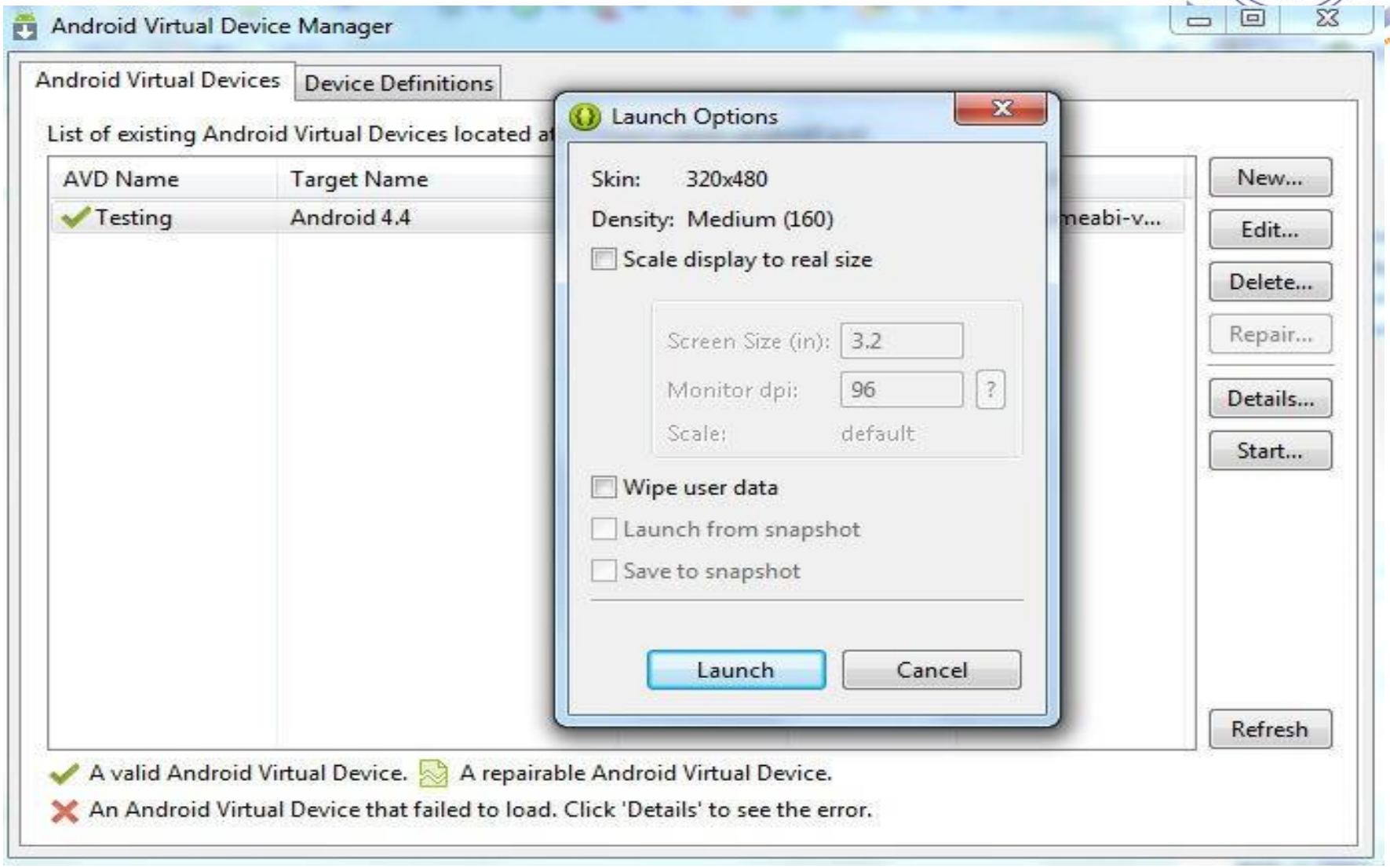
Creating AVD Manager



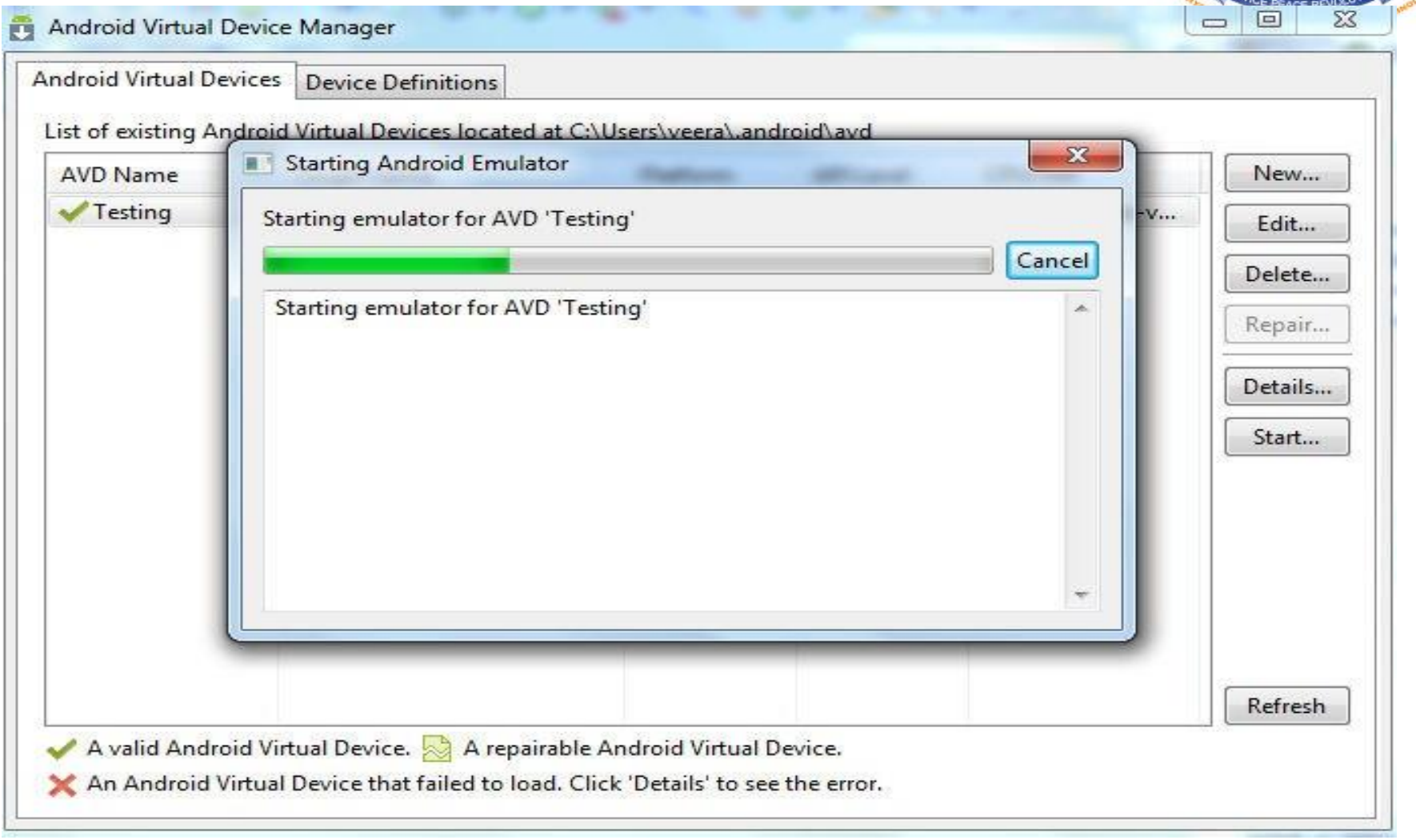
AVD Configuration



Launching the AVD

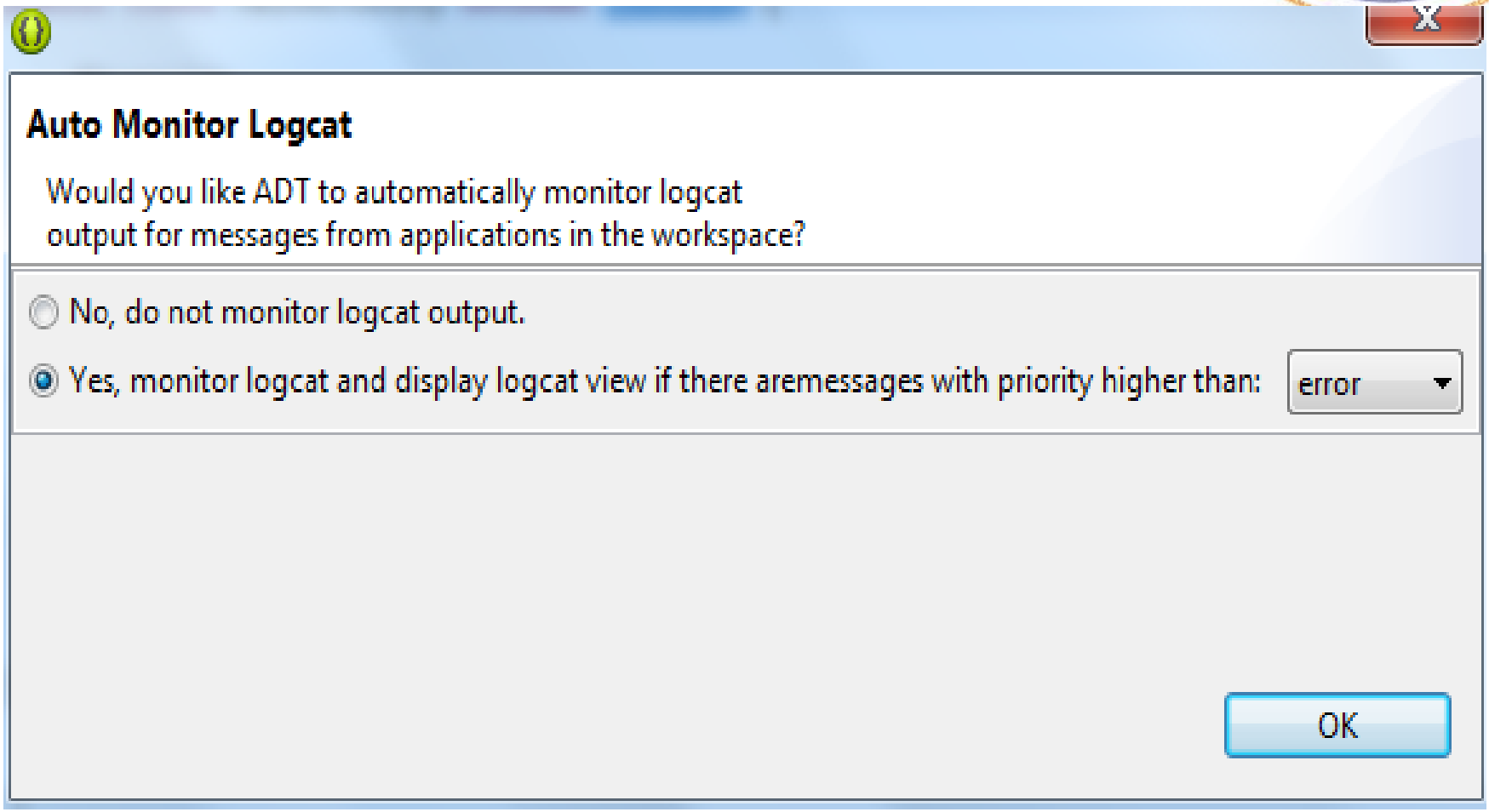


Launching the AVD (con...)

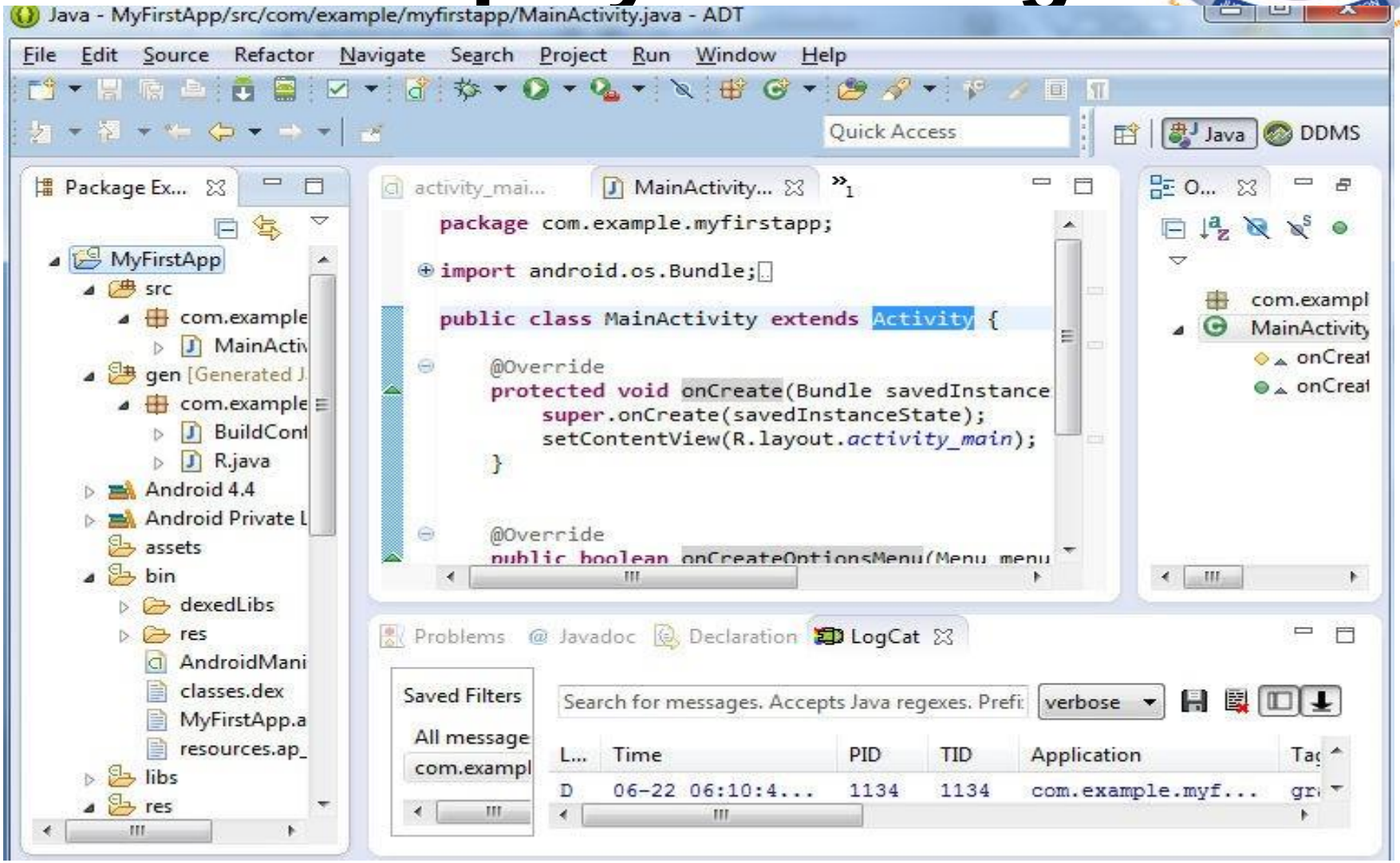


AVD – Emulator

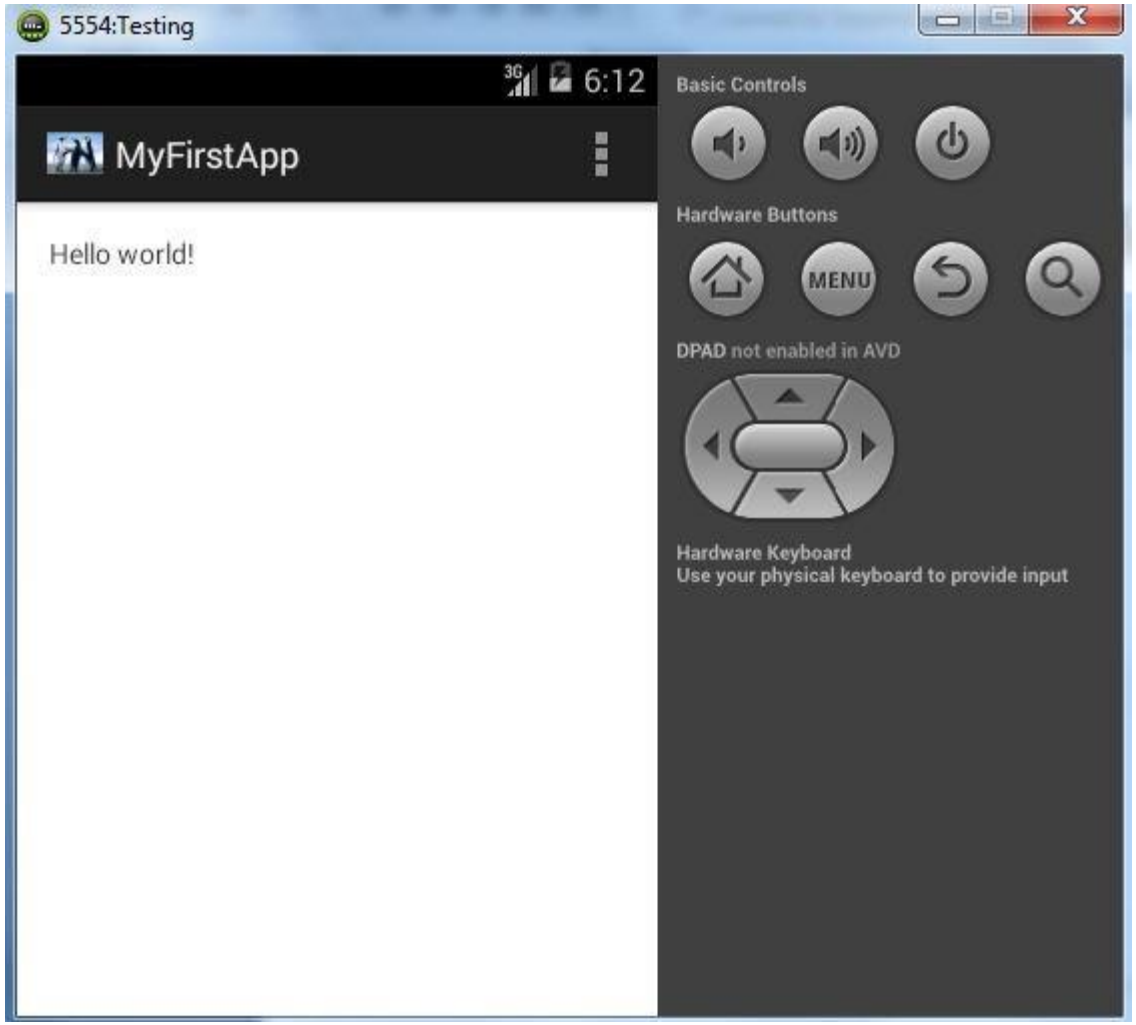
Configure the Logcat



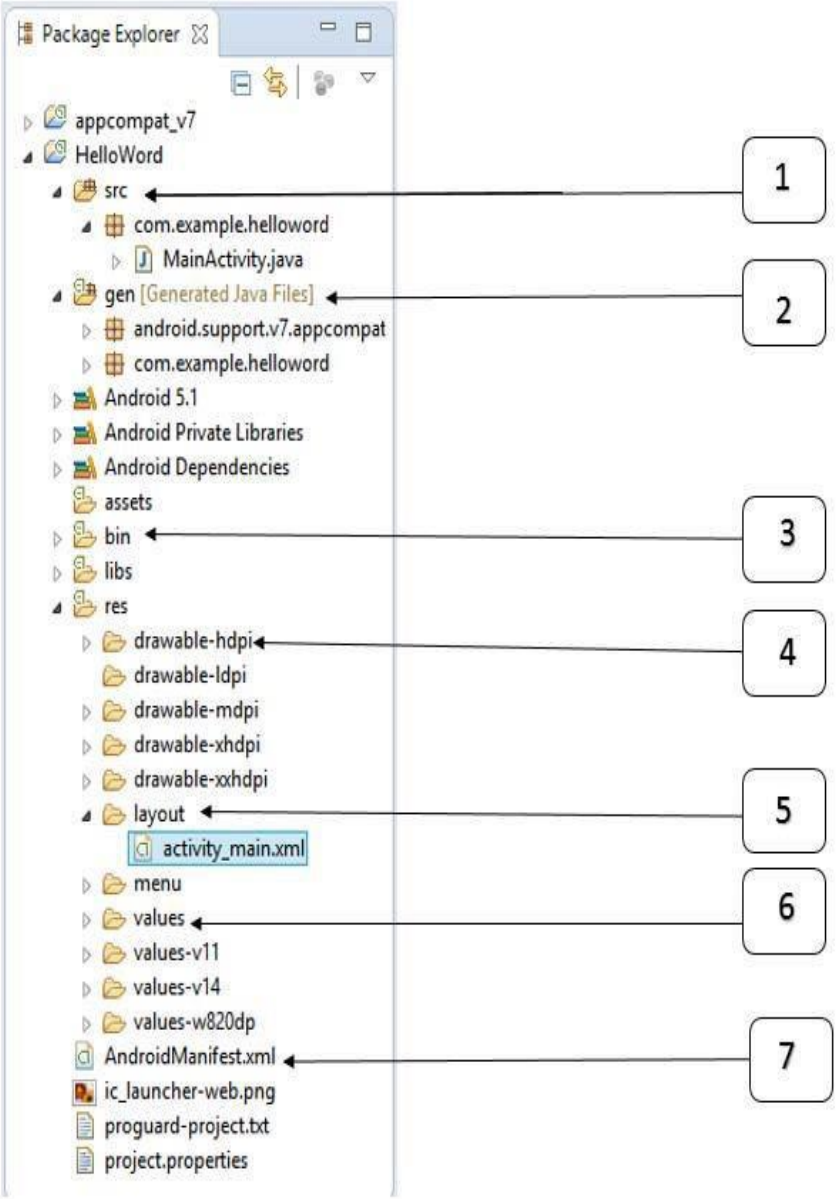
Application running status displayed in Logcat



Output


5554:Testing
6:12
MyFirstApp
Hello world!
Basic Controls
Hardware Buttons
MENU
DPAD not enabled in AVD
Hardware Keyboard
Use your physical keyboard to provide input

10. Application Structure



Application Structure (con…)

1. src
2. gen
3. bin
4. res/drawable-hdpi
5. res/layout
6. res/values
7. AndroidManifest.xml

Application Structure (con…)

1. src
   - 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.
2. gen
   - This contains the .R file, a compiler-generated file that references all the resources found in your project.
   - User should not modify this file.

Application Structure (con…)

- bin
  - 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.
- res/drawable-hdpi
  - This is a directory for drawable objects that are designed for high-density screens.
- res/layout
  - This is a directory for files that define your app's user interface.

Application Structure (con…)

- res/values
  - This is a directory for other various XML files that contain a collection of resources, such as strings and colours definitions.
- AndroidManifest.xml
  - This is the manifest file which describes the fundamental characteristics of the app and defines each of its components.

11. AndroidManifest

- The component you develop as a part of your application, you must declare all its components in a *manifest.xml* which resides at the root of the application project directory.
- This file works as an interface between Android OS and your application, so if you do not declare your component in this file, then it will not be considered by the OS.

AndroidManifest (con…)

- Default manifest file will look like as following file

```
<manifest
   xmlns:android="http://schemas.android.com/apk/res/and
   roid" package="com.example.helloworld"
android:versionCode="1"
android:versionName="1.0" >
<uses-sdk
android:minSdkVersion="8"
android:targetSdkVersion="22" />
```

AndroidManifest (con…)

```
<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>
```

AndroidManifest (con…)

- <application>...</application> tags enclosed the components related to the application.
- Attribute *android:icon* will point to the application icon available under*res/drawable-hdpi*.
- *The @string/app_name* refers to the *app_name* string defined in the strings.xml file, which is "HelloWorld"
- The <activity> tag is used to specify an activity and *android:name* attribute specifies the fully qualified class name of the *Activity* subclass.

AndroidManifest (con…)

- The *android:label* attributes specifies a string to use as the label for the activity / application.
- The action for the intent filter is named *android.intent.action.MAIN* to indicate that this activity serves as the entry point for the application.
- The category for the intent-filter is named *android.intent.category.LAUNCHER* to indicate that the application can be launched from the device's launcher icon.

AndroidManifest (con…)

- Following is the list of tags which you will use in your manifest file to specify different Android application components.
  - <activity> elements for activities
  - <service> elements for services
  - <receiver> elements for broadcast receivers
  - <provider> elements for content providers

Practices

- To know about the history, features and various versions of Android
- Draw the Android architecture
- To study various tools used in Android development
- To study about Eclipse IDE
- To develop first Android App “Hello World”
- To implement the various Android layouts
- To implement the various Android UI controls
- To study the importance of Android application structure and Android manifest file