Android Development

* Android is an open source & linux based operating system for mobile.
* It was developed by Open Handset Alliance led by google and other companies
* Android 1.0 was first version realed in 2008.
* 4.1 Jelly Bean is a incremental update, aim of improving UI.

* Features of android

→ Beautiful UI
→ Connectivity → Bluetooth, Wifi, LTE
→ Storage → SQLite
→ Media support → H.263, H.264, MP3, MP4, JPG etc.
→ Messaging
→ web browser
→ Multi touch
→ Multi tasking
→ Resizable widgets

* History

Android version	Name	API Level.
1.5	cupcake	3
1.6	Donut	4
2.0/2.1	Eclair	5, 6, 7
2.2.x	Froyo	8
2.3.x	Gingerbread	9, 10
3.x	Honeycomb	11, 12, 13
4.0.x	Ice cream sandwich	14, 15
4.4.x	KitKat	19, 20
5.0	Lolipop	21, 22
4.1.x	Jelly bean	16, 17, 18

* API Level :- is a integer value that uniquely identifies the framework API revision offered by a version of android platform

* Android Enviornment setup

i) Java JDK 5 or latest
ii) Android SDK
iii) JRE (Java Runtime enviornment) 6
iv) Android Studio
v) Eclipse IDE for Java developer (optional)
vi) Android Development tool (ADT) (optional)

* Importance of mobile strategies (not imp +)

- Having mobile strategy in place means including security as well as seamless integration of existing application.
- There is no mobile strategy you can use to develop & implement a plan.
- unique as per requirment of company
- you can take steps to identify and develop mobile strategy.

* Third party framework

- created by developers or companies that aren't google.
- eg :- app to schedule workout with friend.
- This app may request ~~main~~ access to your google calender and contacts to suggest times and friends for you to meet up with.

MAD framework.

↙ native App | ↓ web app | ↘ hybrid app

* Native framework :- for single platform, language such as java or kotlin for android and swift or objective-C for iOS

* Hybrid mobile apps :- allows you to develop an application once and then deploy it to different platforms including iOS, Android, or windows.
- saves times for developers by eliminating the need to work on diff project for diff platform.
- A hybrid app is created using single coding language and work on many platforms
- hybrid app dev is about striking the right blend of native and web development technology to bring your app to life.
- In traditional hybrid app to core application is written in HTML, CSS & JS then encapsulated into a container called web view

* MAD: Factor to consider
- Research
- Identify target audience
- Right platform selection
- set plan of action.
- know your budget.
- Think out of box.
- smooth & efficient
- UX

* Mobality : means in which are user are engaging with their devices

* MAD Platform :- software that allows a bussiness to rapidly build, test and perhaps deploy mobile apps for smartphones or tablets.

eg → Palm, Block Berry, iPhone, Android and Windows mobiles

TYPES of OS

* Symbian :- → Dropped mobile OS and platform developed for smartphones.
→ It is closed-source OS for personal digital assistant.
→ OS component microkernel & UI, written in C++

* Feature of Symbian

contains browser, messaging, multimedia, communication, protocol, mobile telephony, data synchronization, security, multitasking, relent web, flexible UI.

→ Advantage

- open platform to enable independent technology and software vendor to develop 3rd party app.
- impressive battery life
- low hardware requirement

→ Disadvantage

- Dependent on nokia
- provide late response as compare to other OS.
- touch was not smooth

Android

Linux based open source mobile OS. especially developed for touch screen.

Android OS is written in Java.

→ Advantage:

- Android app can be build by anyone as opensource.
- multitasking.
- Play store
- notification facility.
- widget.

→ Disadvantage:

- Most app require active internet connection.
- wastage of memory due to background processing
- many apps contain virus.
- not safe as iOS.

iOS

Initially designed and developed by Apple Inc. largest mobile OS after Android designed for apple smartphones & tablet. more secure iOS.

Features:- touch ID for app, icloud drive, Siri, Safari

→ Advantages:- optimized, fast & smooth, less heat, secure, gaming, Jailbreak for customization, face recognition.

Disadvantage:- depend of iOS device, not open source, expensive, app are expensive, single SIM, massive apps.

Android Architecture

Linux kernel :-

At bottom layer there is Linux. provides level of abstraction b/w the devia hardware and it contains all essential hardware drivers like camera, keypad, display etc.

Libraries

On the top of kernel there is set of library including open-source web browser engine Web kit, well known library libc, SQLite database, library to play record audio & video etc.

eg:-

android.app :- access of application model

android.content :- facilitates content access, publishing & msg b/w app

android.database :- access data published by content provider.

android.opengl :- 3D graphics rendering API.

android.os :- provide standard OS services such as msg etc.

Android Run time

It is the third section of architecture and is present on the second layer from the bottom. This section provides a key component called Dalvik Virtual machine which is kind of JVM especially designed & optimized for android.

The DVM make use of linux core features such as memory management & multithreading which is intrinsic in Java. The DVM enables every android app to run its own process, with its instance of the Dalvik virtual machine.

Application libraries.

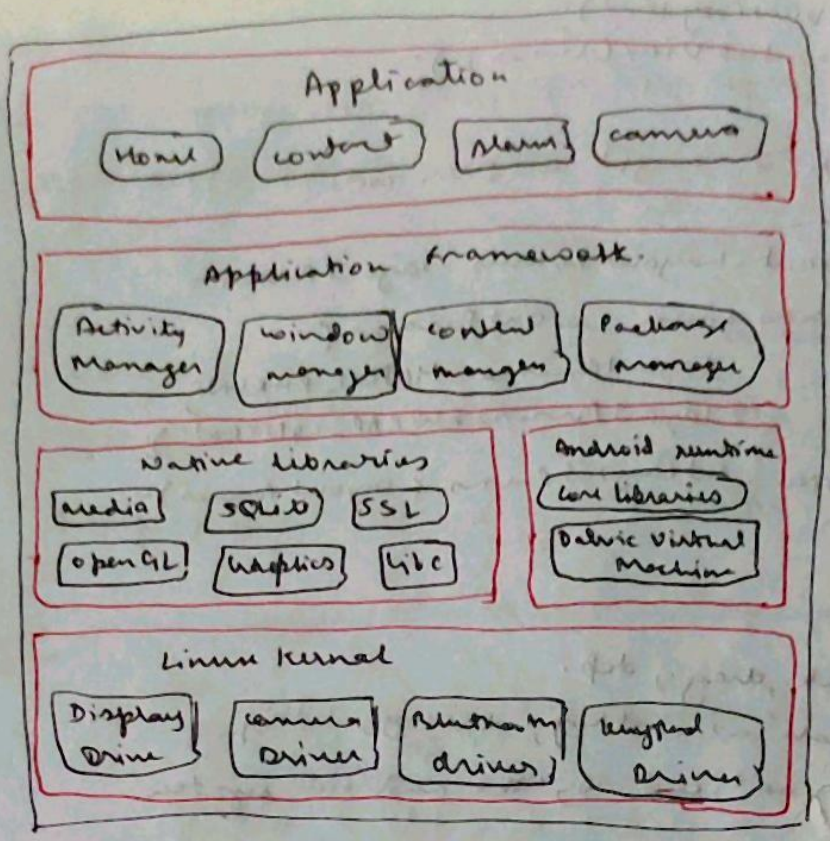
→ Activity manager :- controls all aspects of app lifecycle & activity stack.

→ content provider :- allow app to publish or share data with other apps.

→ Resource manager :- Provide access to resources such as strings, colors settings & UI layouts.

→ Notification manager :- allows app to display alerts & notifications to user.

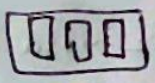
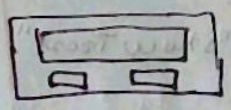
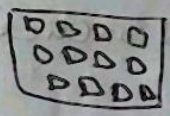
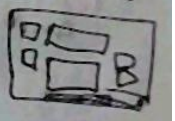
→ View system :- An extensive set of view used to create UI.



Android Architecture

~~View~~ Layouts → specific type of view groups
→ subclass of ViewGroup
→ contains child view
→ can be in row, col, grid, table.

eg → LinearLayout, RelativeLayout, GridLayout, TableLayout

eg in XML

```
<LinearLayout
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <EditText
    ... />
    <Button
    ... />
</LinearLayout>
```

In Java Activity

```
LinearLayout ll = new LinearLayout(this);
ll.setOrientation(LinearLayout.VERTICAL);
TextView tv = new TextView(this);
```

```
l.addView(msgText);
setContentView(linearL);
```

* setting width & height in Java code.

```
LinearLayout.LayoutParams layoutParams =
        new LinearLayout.LayoutParams(
            LayoutParams.MATCH_PARENT,
            LayoutParams.WRAP_CONTENT);
msgMyView.setLayoutParams(layoutParams);
```

→ Events in app

- In UI : click, drag., etc.
- Device : such as walking, driving, tilting.
- events are noticed by the android system.

→ Event handler → method that do something in response to a particular event.

→ Handling click eg XML & JAVA

XML

```
android:onClick="showToast"
```

Java

```
public void showToast(View v) {
    String msg = "Click Triggered";
    Toast t = new Toast.makeText(
        this, msg, duration);
    toast.show();
}
```

Resources :- resources are assets required to build projects these are located in res folder.

eg R.layout.activity_main → inside res → inside layout.
setContentView(R.layout.activity_main)

* measurement :-

(dp) → device independent pixel (for views)

(sp) → scale independent pixel (for text)

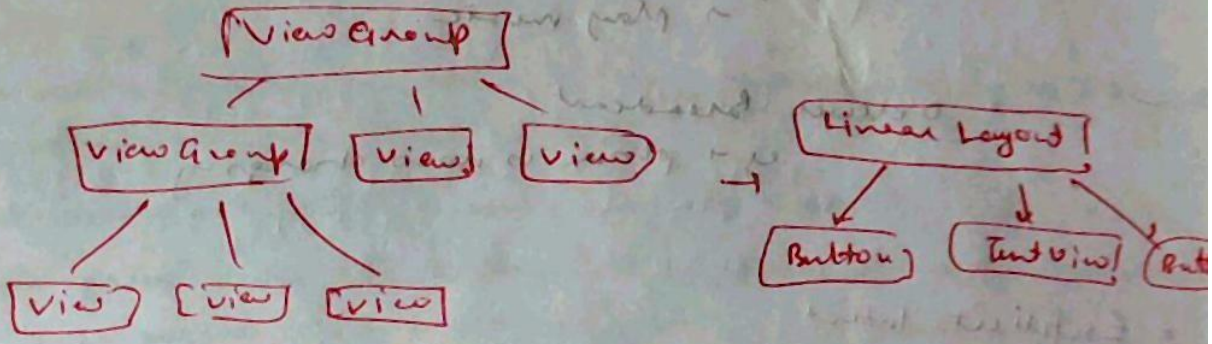
* Views :- every UI element that you see in an app is a view

eg → display text, edit text, button, menu etc.

has properties might, width, color, etc position etc.

* ~~Context~~ context - interface to global information about an app env.

eg code :- Context cnt = getApplicationContext();

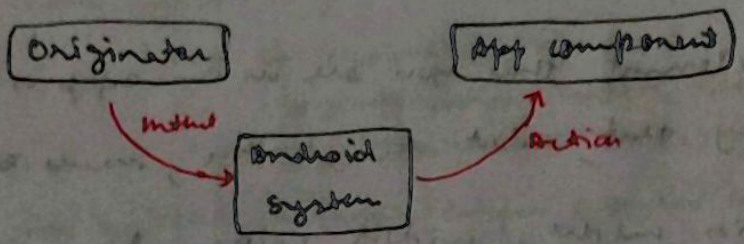


Activities

- is an app component
- represent one window, one hierarchy of views.
- Typically fill the screen but can appear as floating window.
- Java class, typically one activity in one file.
- handle user interaction.
- can start other activities
- has life cycle - is created, started, runs, is paused, resumed, stopped, & destroyed
- Activities are loosely tied to gether to make an app.
- main Activity → user first sees.
- Activities can be organized in parent-child relationships in the android manifest to aid navigation.

Intents

description of operation to be performed. It is an object used to request an action from another app component via the Android system



What Intent can do

- start Activity
  - eg → click to post picture
  - → click to navigate to new activity
- start Service
  - eg → downloading file in activity
  - → play music.
- Deliver Broadcast
  - eg → phone is now charging

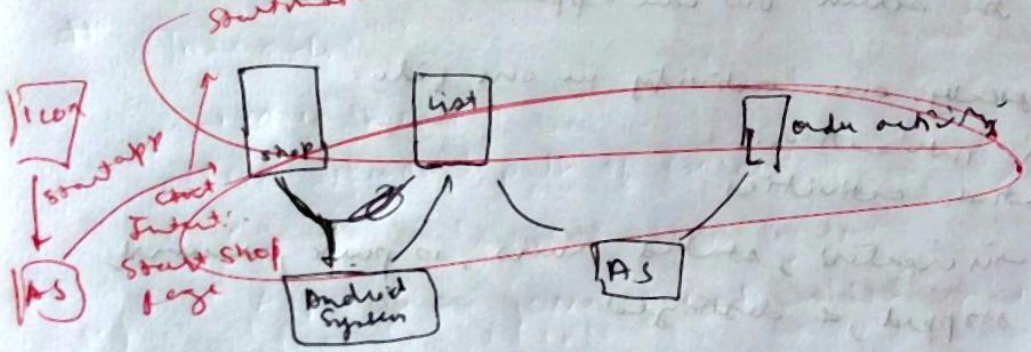
- **Explicit Intent**

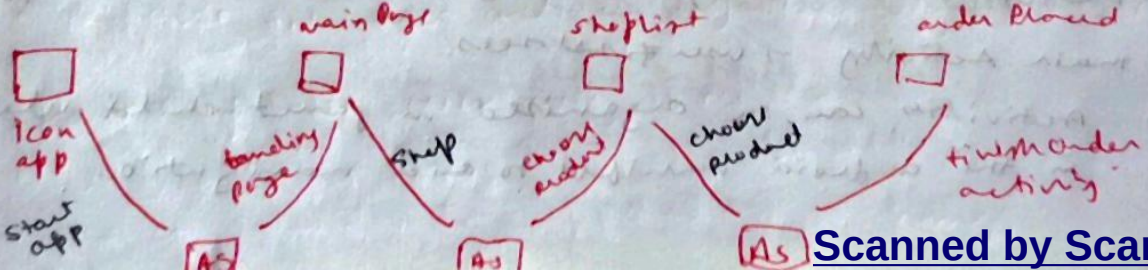
  start a specific activity

- **Implicit Intent**

  Ask system to find activity that can handle the req.

  eg → share app opens on click.





Sending and recieving the data

→ Data → one piece of information whose data can be locatio
located can be represented by an URI

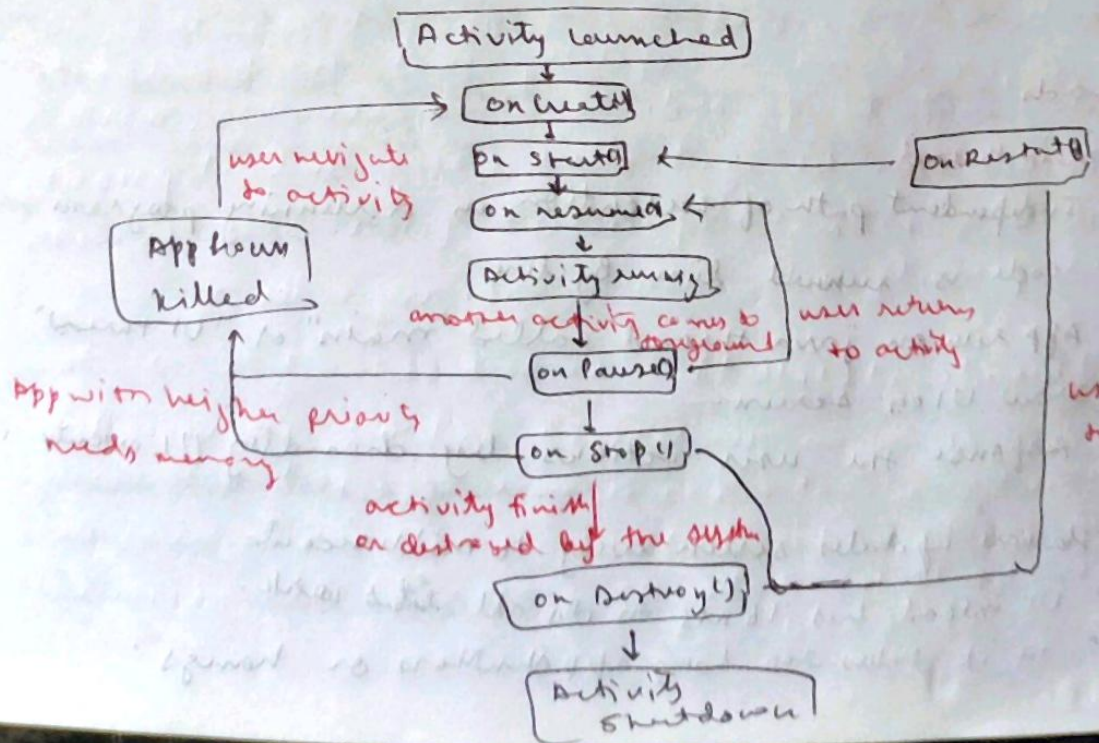
→ extras → one or more piece of information as a collection o
key-value pairs in a Bundle.

In first Activity
- create the intent object
- Put data or extra in the intent
- start the new activity with startActivity()

In second Activity
- Get the intent object the activity was started with.
- Retrieve the data or extras from the intent object.

Activity stack
- When new activity is started the previous activity is stopped and pushed on the back stack.
- Last-in-first out:- when current activity ends, or the user press back ◁ button, it is popped from the stack & the previous activity resumes.

Activity lifecycle :- The set of states of an activity during its lifetime from when it is created to when it is destroyed.



Activity states

- Created (not visible yet)
- Started (visible)
- Resume (visible)
- Paused (partially invisible)
- Stopped (hidden)
- Destroyed (gone from memory).

Call backs

- onCreate() → static initialization
- On start() → activity is becoming visible
- On Restart() → called if activity of stopped
- On Resume() → starts to interact with user
- on Pause() → about to resume previous activity.
- On Stop() → no longer visible to user, but still exists and all state info or persisted
- On Destroy → final call before destroyed from memory activity.

what does config change?

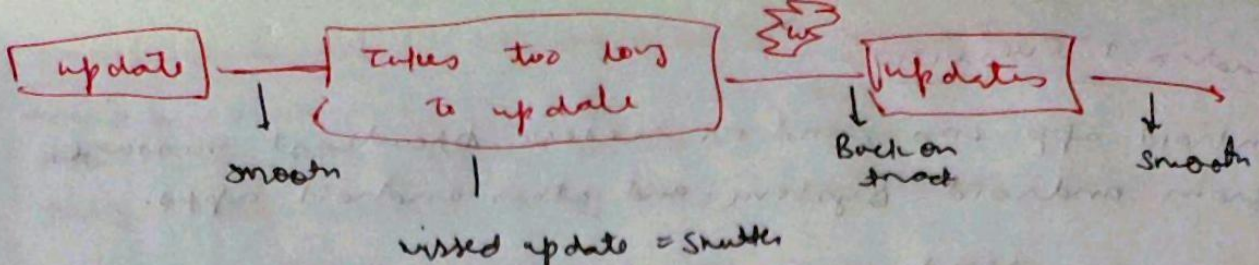
Configuration change invalidate the current layout or other resources in your activity with the user:
→ rotate the device
→ change language
→ enter multi window mode.

Threads

→ Main thread

- Independent path of execution in a running program
- code is runned line by line.
- App run on java thread called "main" or "UI thread"
- Draw UI on screen.
- Response the user actions by handling UI events.

imp. Human updates screen every 16 milliseconds
- UI thread has 16 ms to do all its work.
- If it takes too long app stutters or hangs.



→ Thread of execution for an app

- app is launced, the system creates a thread of execution for the app called "main".
- important because it is incharge of dispatching events to the appropriate UI widgits.
- The system does not create a seperate thread for each instance of component.
- All components that runs in same process are instantiated in the UI thread, and system calls to each component are dispatch from that thread.
- method that response to system callbacks such as onKeyD -n() always run in UI Thread of the process.

→ Application Not Responding (ANR)

If anything is happening in UI thread, performing long operations such as network access or database queries will block the whole UI component.

When Thread is blocked no event can dispatch, including drawables.

From the user perspective app appears to hang. Even worse if UI thread are blocked for more than few seconds (5 sec) the user is presented with "ANR" screen.

User might quit app and uninstall app if unhappy.

So conclusion

Do not block your UI

Do not access the Android UI toolkit from outside the UI thread.

Broadcast Overview :-

- Android app can send or recieve broadcast messages from android system and other android apps.
- These are send when event of interest occurs.
- eg - charging msg, download started, new msg etc.
- Apps can register to recieve specific broadcasts.
- When broadcast is send, the system automatically routes a route to that app.

System Broadcast Broadcast :-

- The system automatically sends broadcast when various evnt occurs. eg aeroplane mode ON/off etc.
- These are send to all apps that are subscribed to recieve the event.
- These are wrapped in intent object whose action string describes the event occured.
- Intent can also include extra information.

Android provides 3 ways to send broadcast msg :-

i) sendOrderBroadcast(Intent, string) :- It send broadcast to one reciever at a time. As each reciever execute in turns, it can propogate result to next reciever, or it can completely abort the broadcast so it won't be passed to other recievers. The order of reciever can be controlled using android:priority attribute of the matching intent-filter. Recievers with same priority will run in arbitrary order.

ii) sendBroadCast(Intent) :- sends broadcast to all the recievers in an undefined order. This is called Normal Broadcast. This is more efficient but recievers can not read results from the other recievers or abort the data or propogate data recieved from the broadcast.

iii) LocalBroadcastManager & sendBroadcast :- send broadcast to all receiver that are in the same app as the sender. If you dont need broadcast across the apps, use local broadcast easy implementation & efficient not to worry about security

Short Message service SMS :-

- standalized communication protocol that is used for the exchange of text msg b/w mobile devices.
- It is one of the most widely used data app
- One can send SMS using phone, computer etc.
- max length of text can be 160 Alpha-numeric characters.
- Provided by all telecom companies like Vodafone, Airtel etc.

→ Advantage :-

- reliable & tested platform for sending important msg.
- discrete form of sending info as compare to calls.
- no internet connection is required.
- conversation is automatically stored until deleted by user.
- pre provided in all mobile.

→ Disadvantages.

- Limited length of 160 - Alpha-numeric characters.
- One has to pay the cost of sending a SMS as it is not free of cost.
- Unsending the ~~msg~~ SMS is not possible.