Module - I

Introduction To Android (An)

- * Android is a s/w package & linux based os for mob devices such as tablet comp & Smart Phns.
- * Developed by Google & later the OHA. (open Handset Alliance).
- * Java lang is mainly used to write the android code evn though othr langs can be used.
- * open source (source code is visible to public & can be edit)
- * Anyone can customize the (An) systm.
- * There are a lot of mob appli~ that can be chosen by the consumer.
- * It provides many interesting features like weathr details, opening screen, etc.
- * categories of (An) appli~ :

- · Entertainmnt → Netflix, Amazon prime
- · commun → whtsapp, telegram
- · productivity → google drive, proof hub
- · personalization → wallpapers, Lauches
- · Music & Audio → MX player, Youtube music
- · social media & video → FB, instragram.

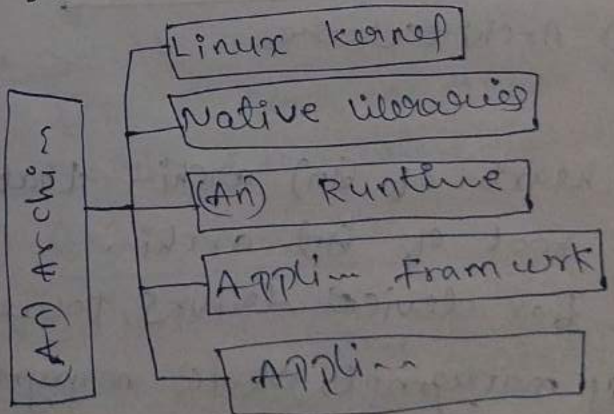
⇒ Android History :

- * Initially Andy Rubin founded (An) incorporaⁿ in 2003.
- * In 2005, google acquired (An) incorporaⁿ.
- * The key employees of (An) incorporaⁿ are Andy Rubin, Rich miner, Chris white & Nick Sears.
- * In 2007, google announces the developmt of android OS.
- * In 2008, HTC launched the 1st (An) mob.
- * (An) is the nick name of Andy Rubin gun by coworkrs bcz of his love to robots

version	codname	Features
• 1.5	cupcake	• soft keyboard • Advanced media recording (img, video) • widget, live folders
• 1.6	Donut	• Advanced search capability. • Text to speech
• 2.1	Eclair	• Improved google maps
• 2.2	Froyo	• USB tethering • wifi hotspot
2.3	Gingerbread	installation of apps frm SD card.
4.0	icecream sandwitch	HD screen, unlock screen in diffrnt ways
4.4	kitkat	More sensors, GPS spport.
10	—	sprt for foldable phn.
11	—	new permission control.

⇒ (An) Architecture | (An) s/w stack :

It is categorized into 5 parts —



x

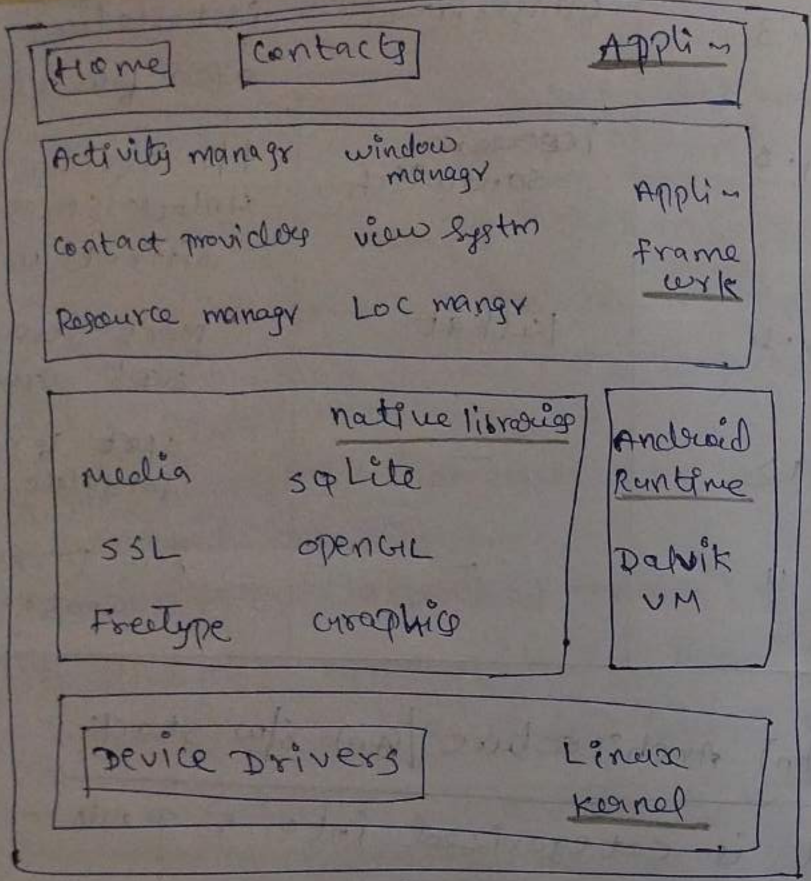


fig : (An) Architecture

1) Linux kernel :

It is the heart of (An) archi~ that exists at the root of (An) archi~

Responsible for devices drivrs, power managmnt, mry managmnt, device managmnt & resource access.

✓

2) Native libraries :

on the top of linux kernel, there are native libraries like webkit (responsible for browsr sprt), SQLite (for DB), FreeType (for font sprt), Media (for playing & recording audio & video)

3) (An) Runtime :

- * Here, there are core libraries & DVM (Dalvik virtual Machine) which is responsible to run (An) appli~
- * DVM is like JVM bt it is optimized for mob devices
- * It consumes less mry & provides fast performnce.
- * Dalvik VM takes the generated Java cls files & combines them into 1/more Dalvik Executable (.dex) files.

4) (An) Framewrk :

- * Includes (An) API's like UI, telephony, resources, loc, contnt providers, & pkge managrs.
- * provides a lot of clses & interfaces for (An) appli~ developmnt.

5) Appli~ :

* on the top of (An) framwork, there are appli.
* All appli~ like home, contact, settings, games, browsers are using (An) framewrk that uses (An) runtime & libraries.
* (An) runtime & native libraries are ~~use~~ using Linux kernel.

=> (An) Java packages :

* android.app → implemnts appli~ model for (An).
* android.app.admin → provides device administra^n^ features at the system level
* android.accounts → provides clses to manage accounts like google, Fb.
* andriod.database → contain clses to explore data returned through a contnt provider.
* android.animation → to apply animation for diff obj, animation clses & methods are included.
* android.contnt → manages access to central ~~respira~~repository of data.
* android.text → provides txt processing clses.

→ (An) Studio =

It is the official integrated develpmnt envrmnt (IDE) for (An) app developmnt.
∴ Based on intellij IDAE, a Java IDE for s/w.

=> Components of (An) / fundamntal componts :

There are some necessary building blocks that an (An) appli~ consist of.

1) Anctivity :
- → represnts a single screen with a user interface & generally contains 1/more views
- → implemnted as a subcls of cls activity
- → UI of our appli~ is built around 1/more extensions of activity cls.
- → public cls MainActivity extends Activity
  { - - - } → general formula.
- → user signup → eg.

2) service :
- → background process that can run for a long time.
- → 2 types → local & remote
  local service is accessed from within the appli~ whereas remote service is

accessed remotely frm othr appln running on same device
- eg → music
- Public class ServiceName extends Service{..}

3) view :
- UI elemnt like btn, label, txtfield, etc
- Anything that you see is a view.
- ~~public class ViewName extends View{..}~~

4) Intent :
- used to invoke components
- used to start the service, launch an activity, display a web pg.
- powerful intr-appln msg-passing framework.

5) Contnt providers :
- used to manage & persist the appln data also typically interacts with SQL DB.
- should be a sub-cls of cls contnt providers
  public cls c-p-Name extends c-p-{-
  public void onCreate() {}}.

6) Fragments :
- are like parts of activity
- An activity can display 1/more fragmnts on the screen at same time.

7) AndroidManifest.xml :
- contains info about activities, contnt providers, permissions, etc
- It is like web.xml file in Java EE

=> (An) virtual device (AVD) :

* It is an emulator configuration that allows developrs to test the appln by simulating the real device capabilities.
* we can configure the AVD by specifying the hardware & s/w options.
* AVD managr enables an easy way of creating & managing the AVD with its graphical intrface.
* Steps to create an AVD frm AVD managr graphical intrface —
- Go to window → AVD managr & select virtual devices.
- click on New to create a virtual device
- give it some name & select target (An) platform from the ~~top~~ dropdown list
- click ~~on~~ 'Create AVD' & we are done.

(ഒരു mob ഫോണിന്റെ [illegible] [illegible] [illegible] [illegible] കാണാൻ [illegible]).

→ Str of (An) Appli~ :

(An) appli~ in Eclipse (in any development tool hue a pre-defined Str with code & resource organized into a ~~n~~ion of folders.

Some components –

1) src – Stands for source code, contains Java source files.

2) gen – Stands for Generated Java library, This library is for (An) internal use only

3) assets → used to store raw asset files

4) Android 4.2.2 – (An) framework library is stored here.

5) libs – contains private libraries.

6) res – stands for resource files.

7) AndroidManifest.xml – (An) definition file.

8) project.properties.

9) MainLayout.xml , etc.

10) proguard-project.txt.

⇒ Appli~ life cycle :

* It is controlled by 7 methods of android.app.Activity cls. (An) activity is the subcls of Context Themewrapper cls.

obj
↑
context
↑
Context wrapper
↑
Context ~~wrapper~~ theme wrapper
↑
Activity.

* 7 lifecycle methods –

1) onCreate – called whn activity (A) is first created.

2) onStart – called whn (A) is becoming visible to the user.

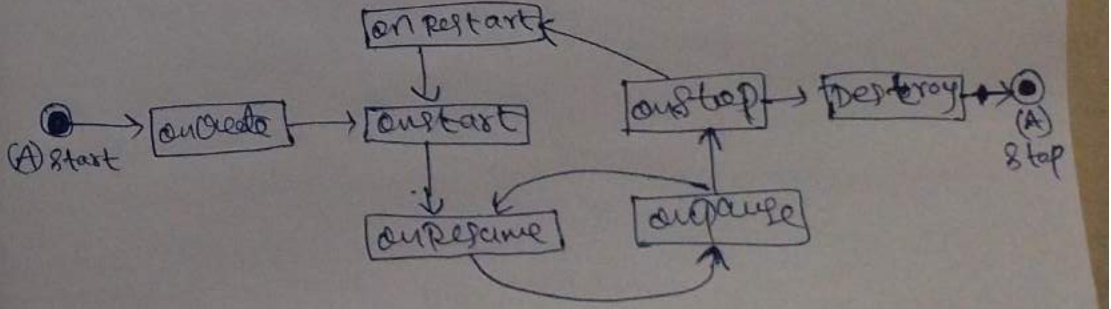
3) on pause – called whn (A) is not visible to the user.

4) On Resume – Called whn (A) will start interacting with the user

5) onStop – called whn (A) is no longr visible to the user.

6) onRestart – called ~~after~~ yr (A) is stopped, prior ([illegible]) to start

7) onDestroy – called bfre the (A) is destroyed.


onRestart
(A) Start
onCreate
onStart
onResume
onPause
onStop
Destroy
(A) Stop