

GOVERNMENT GIRLS POLYTECHNIC BILASPUR



Session 2024-25

MOBILE APPLICATION DEVELOPMENT (MAD)

BY: - VIKAS SINGH RAJPUT

(PTL)

* Development of Android Application *

(1)

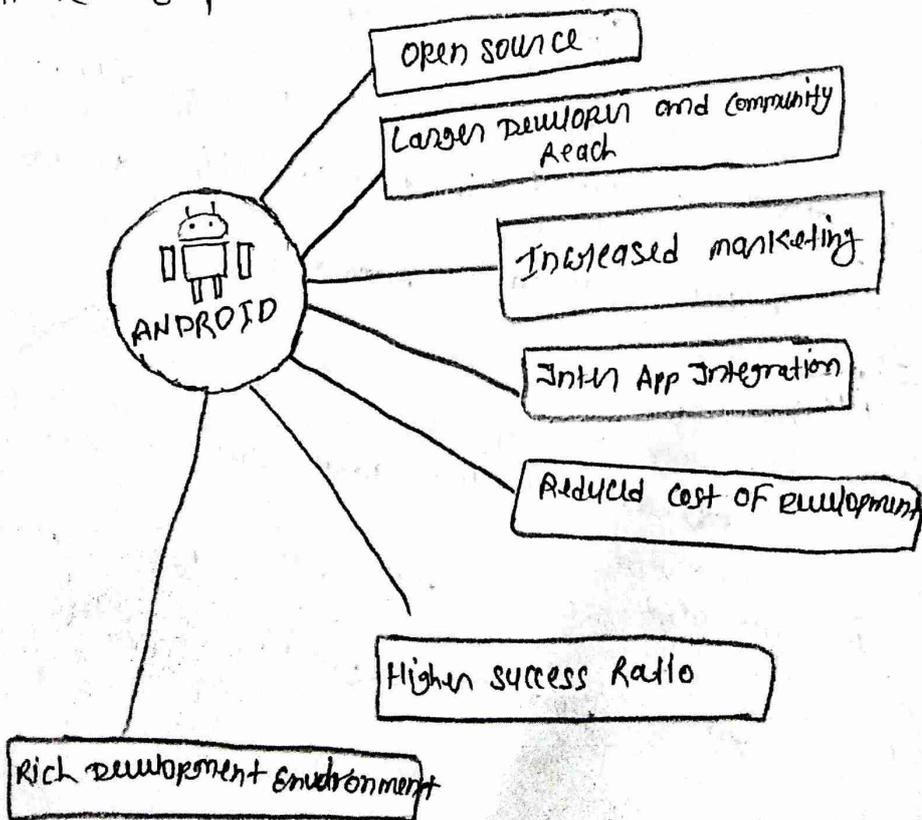
SATYAM SINGH

7275791567

** What is Android? *

Android एक OS & programming platform है जो Google द्वारा mobile phone & अन्य mobile phone उपकरणों में, टैबलेट, Computer & smartphone के लिए विकसित किया गया है।

Android एक open source linux base operation system है Android में एक software development kit (SDK) शामिल है।



* History of Android :-

Android Oct, 2003 में शुरू होता है। Android inc. की थापना कैलिफोर्निया के पालो अल्तो में हुई थी।

* इसके चार संस्थापक rich minner, nick siyars, krish white, & Andi Rubin थे। जून 2005 में Android का Google में विलय होने पर Google में शामिल हो गए थे।

* Android का पहला सार्वजनिक Android Beta version 1.0 की अन्तर्गत; 5 Nov, 2007 को प्रकाशित किया गया था।

* Android का नाम Sweet के नामों पर रखा जाता था।

(i) Android 1.0 (API 1), (ii) Android 1.5 (cupcake) (iii) Android 2.0, Eclair

(iv) Android 2.3, Gingerbread, (v) Android 3.0, Honeycomb (vi) Android 4.0, Ice Cream

Sandwich (vii) Android 4.4, KitKat, (viii) Android 5.0 Lollipop (ix) Android-6.0

marshmallow, (x) Android 7.0 Nougat (xi) Android 8.0 Oreo,

* Android 9.0 pie

Android 9.0 को Aug, 2018 में पेश किया गया था इसके various

aspect में अबरजस्त सुधार लाया और कृत्रिम बुद्धिमत्ता

की शक्ति का आभाचारण उपयोग किया।

*: Android की विशेषताएँ :-

- ** Beautiful UI :- Android OS basic screen from 16:9 और यह interface प्रस्तुत करता है।
- ** Connectivity :- सभी GSM/EDGE, 3G, CDMA, Bluetooth, Wi-Fi, LTE और Wi-max सभी की support करता है।
- ** Storage :- SD/HC, एक एक relational database और उपयोग data storage अलग की भी बात करता है।
- ** multitasking :- यहाँ एक कार्य में दूसरे पर जा सकता है।
- ** multi-touch :- multi touch के लिए Android में support करता है।
- ** multi-language :- single direction & bi-direction की support करता है।
- ** wifi-direct :- एक ऐसी technic है high bandwidth वाली peer to peer connection search कार्य आसानी से।
- ** Android Beam :- एक लोकप्रिय NFC आधारित तकनीक है जो उपयोगकर्ता को छूटते ही NFC सक्षम phone को एक साथ प्रेषित करने में सक्षम करती है।

* Android के लाभ :-

1. Android APPS तक आसान पहुँच।
2. यह Google की सभी services का support करती हैं।
3. OS आपको New update की सूचना देने में सक्षम हैं।
4. यह multi tasking का support करती हैं।
5. यह अनुकूलित करने के लिए स्वतंत्र हैं।
6. एक संगीधित RAM स्थापित कर सकती हैं।
7. यह 2D or 3D graphics का support करता हैं।

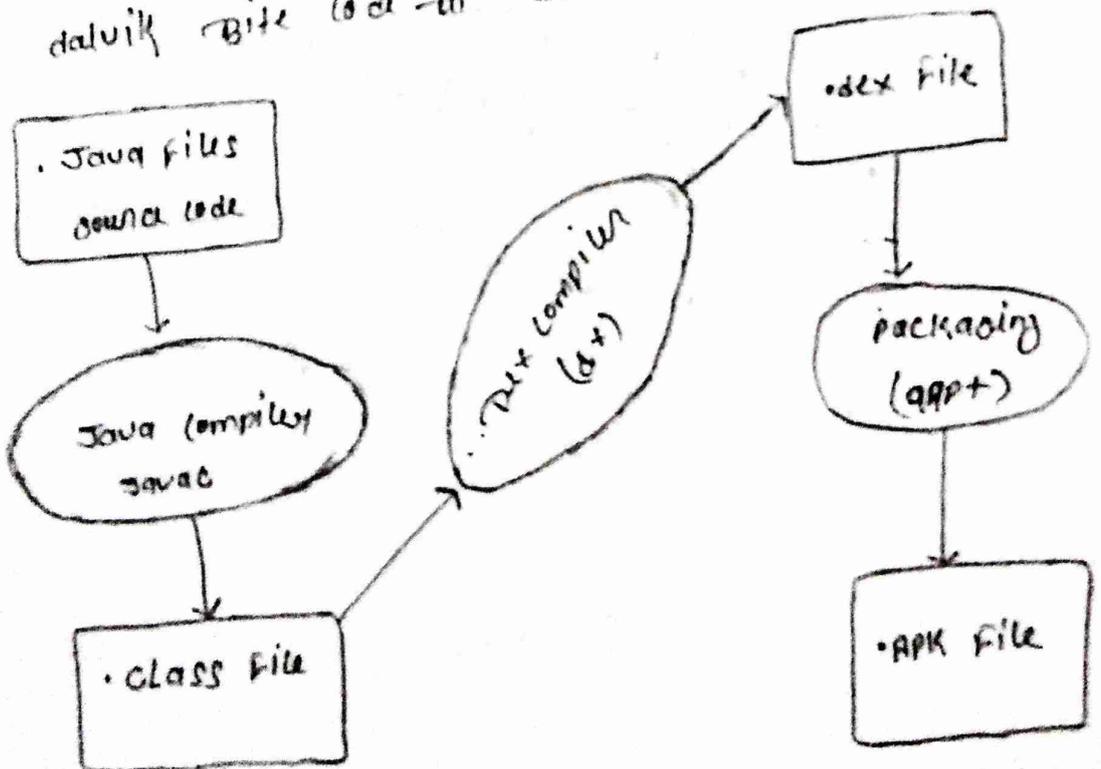
* Dalvik virtual machine (DVM) :-

Dalvik एक ऐसा मंच है जो एक जे.आर.एम. मशीन को एंड्रॉयड डिवाइस में चलाने के लिए डिज़ाइन किया गया है।

Dalvik virtual machine mobile डिवाइस के लिए अनुकूलित एक Android virtual machine है।

ये memory, battery life, और performance के लिए virtual machine optimizes करता है।

Java compiler, Java source code, Java byte code (.class) में परिवर्तित करता है जो Dex compiler को (.class) file को Dalvik Byte code - "dex" में change कर देती है।



* Advantage:-

1. Dvm केवल Android OS का support करती है।
2. Dvm executable में APK है।
3. execution fast है।
4. Dvm के लिए design किया गया था ताकि कई device virtual machine के multiple instances को effectively में run कर सकें।

* Disadvantage:-

1. Dvm केवल OS का support करती है।
2. Dvm के लिए बहुत-से tools Available हैं।
3. DEX के कारण APP installation में अधिक समय लगता है।
4. अधिक internal storage की आवश्यकता है।

* पृथ * में राखनी *

* APK file Extension :-

APK (Android package kit) एक Android application package है

APK Android के लिए उपयोग किए जाने वाले Application file प्रकार है।

* APK file Android Studio में compiled की जाती है, जो Android software के निर्माण के लिए official integrated Development environment (IDE) है।

* • APK file Android, Google operating system Android के लिए बनाई जाती एक APP है।

APK file manually install कर सकते हैं।

• APK file में assets, resources, certificates, manifest files, application program और data शामिल होते हैं।

APK files एक compressed .zip प्रकार में होती जाती हैं और इसे किसी भी ZIP decompression tool का उपयोग कर सकते हैं।

* Android basic building blocks component :-

Component^{एक} Android के आवश्यक building blocks है।
एक Android Component piece of code है।
जिसमें एक अच्छी तरह से परिभाषित life cycle है।
e.g. → activity, BroadcastReceiver, Service etc.

* Activities :-

यह केवल class है जो केवल एक screen को represents करता है। वे UI को करते हैं और smartphone screen पर user की interaction को संचालित है यह AWT में एक frame की तरह है।

```
public class MainActivity extends Activity  
{  
    ...  
}
```

* Activity Life Cycle :-

Android system on create() call back method पर call के साथ शुरू होने वाली activity से अपना program शुरू करता है और call back method का एक sequence होता है जो एक activity को काट देता है।

Service is started by start service

on create()

on start()

service is running

The service is stopped (no call back)

on destroy()

service is shut down

Un Bounded service

Service is created by bind service

on create()

on bind()

Client interacts with the service

on rebind()

on unbind()

on destroy

Service is shut down

* Broadcast Receivers:

Broadcast Receivers कोवल अन्य

Application या system से Broadcast message को
 प्रदान करता है। व Android, OS और Application
 के बीच संचार संभालता है।

एक Broadcast Receiver को BroadcastReceiver class
 के subclass के रूप में implements किया जाता है।
 और प्रत्येक msg Broadcaster के रूप में एक Intent
 object होता है।

```
public class MyReceiver extends BroadcastReceiver
```

```
{
    public void onReceive (Context, Intent) {}
}
```

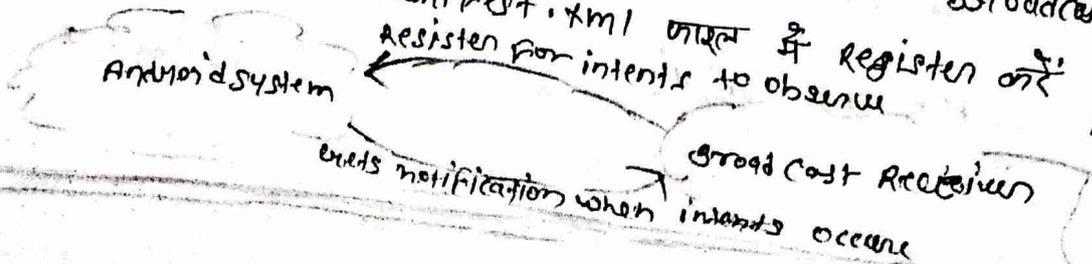
दो महत्वपूर्ण Step द्वारा Broadcast बनाया गया है।

1. Creating the Broadcast Receiver:

BroadcastReceiver को Broadcast
 class के subclass रूप में Intent object implement किया जाता
 है और onReceive method को override किया जाता है।
 यहां प्रत्येक msg को Intent object parameter के रूप में प्राप्त किया जाता।

2. Registering Broadcast Receiver:

एक Application specific
 Broadcast intents तभी listen कर सकता है जब Broadcast
 receiver को AndroidManifest.xml फाइल में Register करें।
 Register for intents to observe



Content provider:-

Content provider dat encapsulation करे Application को content प्रदान करते हैं। content provider का उपयोग multiple Application को मध्य data को share करने के लिए किया जाता है।

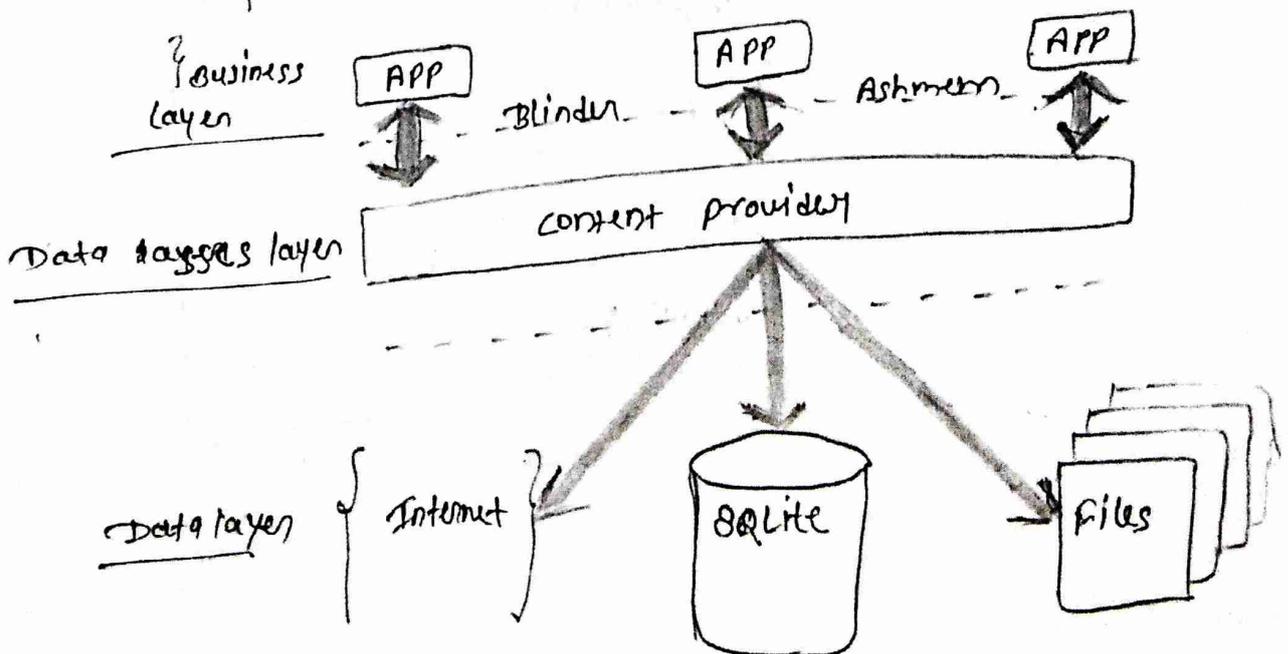
Content providers को content provider class के subclass के रूप में implement किया जाता है।

```
public class my content provider extends content provider
```

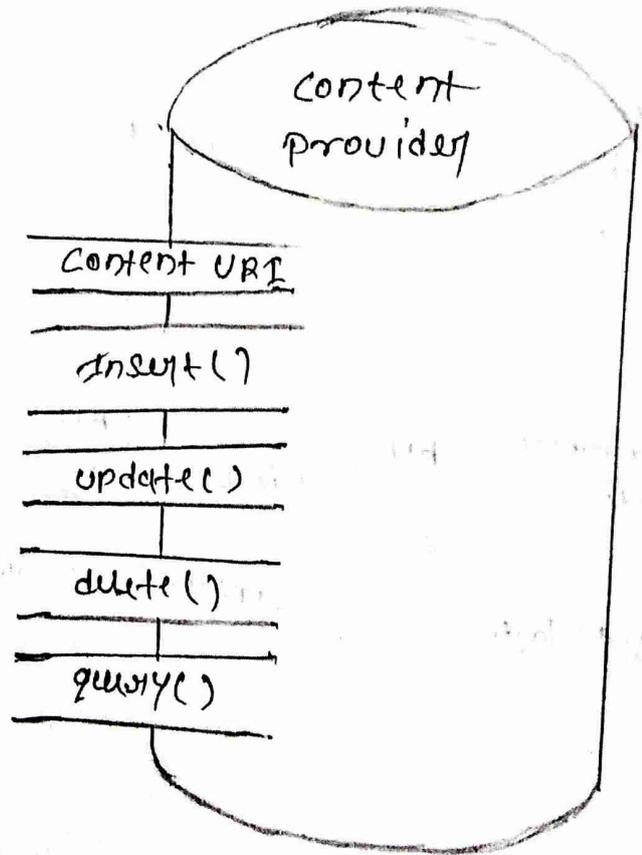
```
{
    public void onCreate ()
```

```
{
```

```
{
```



- एक Content provider एक data base की तरह व्यवहार करता है जहाँ आप सभी query कर सकते हैं। सभी content को edit कर सकते हैं, साथ ही insert(), update(), delete(), and query() method का उपयोग करके सभी को Add or delete कर सकते हैं।
 खाता पर यह data SQLite में store होता है।

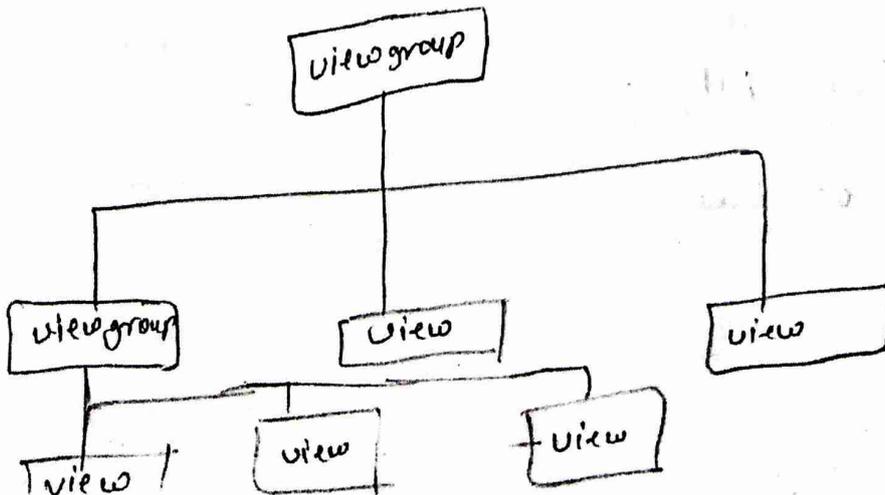


* UI Component :-

Android App के लिए user interface को layout और views के hierarchy के रूप में बनाया गया है। Android विभिन्न प्रकार के pre-built UI components जैसे layout objects और UI controls प्रदान करता है जो आपको अपने ~~आपने~~ app के लिए graphical user interface में मदद करता है।

* View :-

एक view एक ऐसी object है जो screen पर कुछ draw करता है जिसमें user interact कर सकता है। View, widgets के लिए basic class है जिसका उपयोग interactive UI components जैसे button, text field आदि बनाने के लिए किया जाता है। ViewGroup, view का एक subclass है और ~~invisible~~ invisible container प्रदान करता है। जो other view या other group को hold करता है।



Layout दो तरीके से declare किया जा सकता है।

* Declare UI elements in xml:- Android एक मूल XML vocabulary प्रदान करता है जो view classes और subclasses में मिला जाता है, जो कि widgets & layouts.

* Instantiate layout elements at runtime:- आप view और ViewGroup objects को प्रोग्रामिंग में बना सकते हैं।

* Notification :-

Notification एक msg है जिसकी Android आपके APP'S UI को outside display करता है ताकि user को reminder, communication या आपके APP के बारे में other information timely प्रदान कर सके।

1. status bar notification drawer
2. Heads up notifications
3. lock screen
4. App icon badge.
5. wear OS device

• Component for communication:

1. Intent:

एक intent एक messaging object है जिसका उपयोग किसी भी Another app component में Action perform करने के लिए किया जाता है।

Intent मुख्यतः तीन तरीकों में communication components के तीन fundamental use में काम करता है।

1. Starting an activity
2. Starting a service
3. Delivering a broadcast.

2. Intent Filters:

एक Intent filter एक app की manifest में एक expression है जो कि कि प्रकार के intent को specify करता है जो कि component प्राप्त करना चाहते हैं।

Intent filters (<intent-filter>) जो कि app की manifest

में हैं परिभाषित करते हैं, उन्हें संबंधित app components

के लिए निर्दिष्ट करते हैं। जो कि कि element को `<action>`, `<category>`, `<data>`

* Setting up development Android environment, Android

किसी भी Android operating system पर (जहाँ) Android development

कर सकते हैं :-

- 1. Microsoft Windows
- 2. Mac OS
- 3. Linux

ये apps को विकसित करने के लिए कुछ वातावरण की आवश्यकता है। इसलिए हमें Android environment सेट करने के लिए

कुछ प्रमुख tools, SDK और setup file download करना पड़ेगा।

* TOOLS

- 1. Java Development Kit
- 2. Android SDK

* Java Development Kit :-

Android SDK, Java Development Kit पर काम करता है। इसलिए पहले हमें system में SDK install करना होगा। तब आगे की process होगी।

* Android SDK / Android Studio / Eclipse :-

इसके बाद Android SDK आता है जहाँ Android के Release लिए नए संस्करणों के प्रत्येक API Level के लिए libraries, debuggers, एक emulator document sample codes और tutorials शामिल हैं।

* Setting up development Android environment

किसी भी Android operating system पर अपना Android development

कर सकते हैं :-

- 1. Microsoft Windows
- 2. Mac OS
- 3. Linux

इस apps की विभिन्न करने के लिए कुछ वातावरण की आवश्यकता है। इसलिए हमें Android environment set करने के लिए

कुछ प्रमुख tools, SDK और setup file download करना पड़ेगा।

Tools

- 1. Java Development Kit
- 2. Android SDK

* Java Development Kit :-

Android SDK, Java Development Kit पर काम करता है। इसलिए पहले हमें system में SDK install करना होगा। तब आगे की process होगी।

* Android SDK / Android Studio / Eclipse :-

इसके बाद Android SDK आता है। हमें Android के release लिए गए संस्करणों के प्रत्येक API level के लिए libraries, debuggers, एक emulator document sample codes और tutorials शामिल हैं।

* Android Manifest.xml file:-

Android के प्रत्येक project में एक manifest फाइल शामिल होती है जो कि AndroidManifest.xml है। जो अपने project hierarchy की root directory में होता है।

* AndroidManifest.xml file में package की जानकारी शामिल है जिसमें application के component जैसे Activity, service, BroadcastReceiver, सामग्री आदि शामिल है।

* इसके काम :-

i. Application को protect करने के लिए जिम्मेदार होता है।

ii. Android App को Declare करता है।

iii. यह Instrument classes को सूचीबद्ध करता है।

</manifest >

* Intent-filter :-

Intent filter, Activity का sub-element है जो उस Activity के प्रकार के intent को describes करता है जिसे वह Activity, Service या BroadcastReceiver response कर सकता है।

* </manifest > 'manifest' is the root element of the androidManifest.xml file.

* uses-features :- इसका उपयोग यह specify करने के लिए किया जाता है कि किस hardware आपकी application की आवश्यकता है।

* Application :- यह manifest का sub element है। इसमें namespace declaration शामिल है।

(i) android:icon

(ii) android:label

(iii) android:theme

* Activity :- Activity Application का subelement है और यह Activity को AndroidManifest.xml फाइल में परिभाषित किया जाना चाहिए।

* Gradle :-

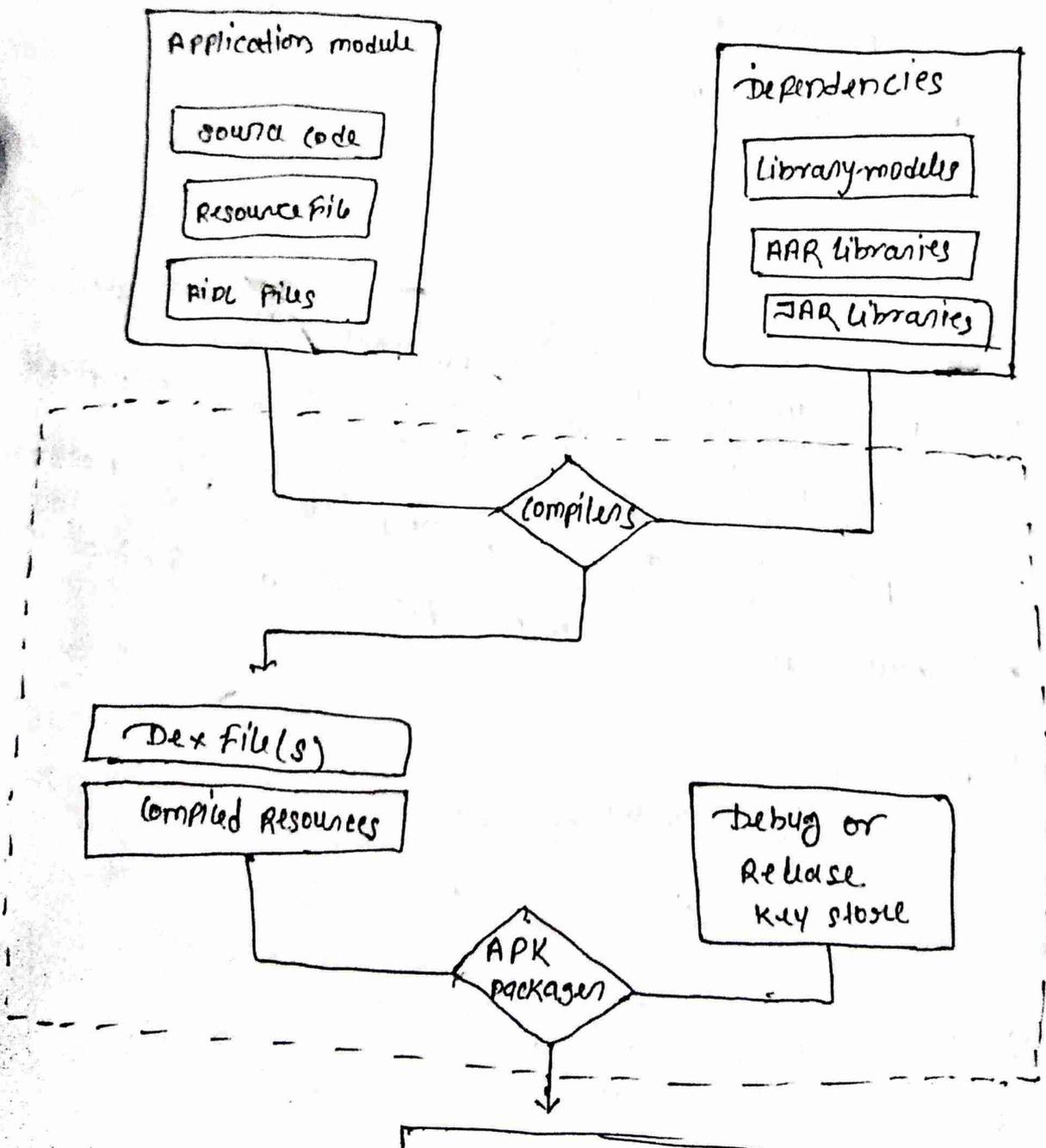
यह multi-language साफ्टवेयर development के लिए एक Build Automation tool है। यह testing deployment और publishing के लिए, compilation और packaging के कार्यों में development process को control करता है।

• Gradle, flexible custom building configurations परिभाषित करने की अनुमति देती है।

• Gradle, source और xml files को compress करके APK फाइल में बदल देता है।

- * In Android studio project, it uses gradle & ...
- * For application & ...
- * Project uses build file & ...

* the build process:-



* Gradle files का detail description :-

- (i) setting.gradle :- इसका उपयोग APP में उपयोग किए जाने वाले सभी module को specify करने के लिए किया जाता है।
- (ii) Build.gradle (Project level) :- यह Project level पर build configurations को परिभाषित करता है।
- (iii) Build.gradle (Application level) :- Application level build.gradle file Android project के सबसे नीचे module में स्थित है।

* Command line tool के माध्यम से Gradle system शुरू करना :-

- (a). `./gradlew build` - (build project)
- (b). `./gradlew clean build` - (build project complete scratch)
- (c). `./gradlew clean build` - (run the test)
- (d). `./gradlew wrapper` - (to see all the available tasks)

* ANDROID PERMISSIONS :-

ANDROID system permission का उपयोग करता है जो किसी भी application में क्या कर सकते हैं और क्या नहीं कर सकते, सब manage करता है।

* uses-Permission :-

इसके माध्यम से अज्ञात करने हैं एक system permission को specifies करता है जो user को app को सही ढंग से संचालित करने के लिए देना होगा।

Syntax:-

```
<uses-permission android:name="string"/>
```

Attribute:-

android ; name

यह permission का name है, जिसमें application द्वारा परिभाषित permission हो सकती है।

* uses-sdk :- यह app level के माध्यम से Android platform में एक या अधिक संस्करणों के साथ एक application's compatibility व्यक्त करने देता है।

* अम माँ सरावती *

* इस element का उपयोग API level को specify करने के लिए किया जाता है।

Syntax :-

```
<uses-sdk android:minSdkVersion = "integer" .  
android:targetSdkVersion = "integer"  
android:maxSdkVersion = "integer" />
```

Attribute :-

- Android:minSdkVersion Application की चलने के लिए आवश्यक minimum API level को specify करने वाला पूर्णांक
- एक integer का अधिकतम API level शामिल करता है जिसे यह Application की चलने के लिए design किया गया है।

* Resources & R. Java :-

(i) APP Resources :-

Resources आतिरिक्त file और static content से है जिसे को ~~आपका~~ code उपयोग करता है।

जैसे कि `string`, `bitmap`, `layout`, `definitions`, `user interface`
`string`, `animation` instruction जैसे बहुत कुछ ।

* इन resources को होता project की `res/` directory के
विशिष्ट तहत विभिन्न sub-directories में अलग-2 बना कर
रखा जाता है।

* Organize resource in Android Studio:



हमें project की `res/` directory के एक विशेष
sub-directory का रखा प्रभाव चाहिए ।

Directory: `color/`, `anim/`, `animation/`, `drawable/`,
`mipmap/`, `layout/`, `menu/`, `raw/`, `values/`.

* Alternative resources:

Alternative resource का उपयोग, हर app की specific
device configuration का निर्धारण करने के लिए
उपयोग करते हैं।

* Accessing Resources - Application Development के दौरान

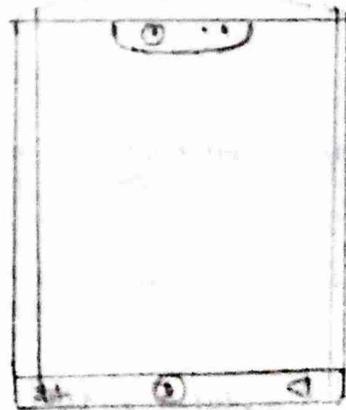
defined resources की अपनी कोड में, या `layout`
`xml` file में access करना होगा ।

* ANDROID EMULATOR :-

Android emulator computer पर Android device को simulate करता है ताकि प्रत्येक physical device के आवश्यकता के बिना प्रयोग उपकरणों और ANDROID APP LEVEL पर अपने Application का परीक्षण कर सकें।

• Emulator Android device की सभी क्षमता को प्रदान करता है।

Emulator विभिन्न Android phone, tablet, Android TV के लिए पूर्व निर्धारित configuration के साथ आता है।



* Android virtual device :-

Android emulator का प्रत्येक Instance Android version और

AVD का उपयोग करता है।

Android virtual device, Android version और simulated device hardware की विशेषताओं को specify करता है।

जब हम emulate launch करते हैं तो यह AVD directory में user data और sd card data load करता है।

* INSTALL the Emulator :-

Android Studio install करते समय emulator की install किया जाता है। Android emulator को install करने के लिए SDK Manager के SDK tool tab में Android emulator component का चयन करें।

* Run the Android Emulator directly in the Android Studio :-

Android Studio emulator चलाने के लिए इन steps का पालन करें।

i) File > setting > tools > Emulator > Preferences > tools > Emulator on macos पर click करें, फिर tool window में launch करें और OK पर click करें।

ii) यदि Emulator window automatically दिखाई नहीं देती है, तो इसे View > Tool window > Emulator पर click करके खोलें।

iii) AVD Manager का उपयोग करके या अपने APP को चलाने समय इसे launch करके अपने virtual Device को प्रारंभ करें।

* Emulator shortcut/command :-

- i) Home → Shifts to main screen.
- ii) F2 → Toggles context sensitive menu.
- iii) F3 → Bring out call log.
- iv) F4 → End call
- v) F5 → Search
- vi) F6 → Toggle track ball mode
- vii) F7 → Power button

* Emulator settings & ~~Extended control~~ :-

- i) General :-
 - Emulator window theme
 - Send keyboard shortcuts
 - Screenshot save location
 - Use default ADB location

when to send crash reports.

Show window frame around twice.

(ii) Advanced :

Open the ES menu :- Graphics acceleration प्रकाश चयन करें।

Open the ES App level :- Emulator के उपयोग के लिए स्मैक पर माथ(0) का उपयोग करें।

* Logcat usage :-

Android Studio में logcat window system संदर्भों को प्रदर्शित करती है।

1. View your app logs :- Application के लिए log message प्रदर्शित करने के लिए।

(i) किसी device पर अपना ऐप बनाएं और चलाएं।

(ii) view > Tool window > logcat पर click करें।

logcat toolbar विभिन्न बटन प्रदान करता है।

(i) Clear logcat

(ii) use soft wraps

(ii) Scroll to the end

(iii) Print

(iii) up the stack trace

(iv) Restart

(iv) Down the stack trace

(v) Screen capture

(vi) Screen record

* जम में सरहवरी *

Introduction to DDMS :- Dalvik debug monitor service.

DDMS Device पर कई सेवाएं प्रदान करता है जैसे कि message formation, call spoofing, capturing screenshot, exploring internal threads & file system etc. शामिल हो सकते हैं।

(i) working :- Android Studio में Tool > Android > Android device monitor

जब DDMS शुरू होता है तो यह ADB को खोजता है। जब कोई device connect होता है तो एक VM monitoring service ADB और DDMS के मध्य बनाई जाती है; जो device पर एक VM शुरू होने या समाप्त होने पर DDMS को सूचित करता है।

(ii) SMS करना :-

(iii) call करना :-

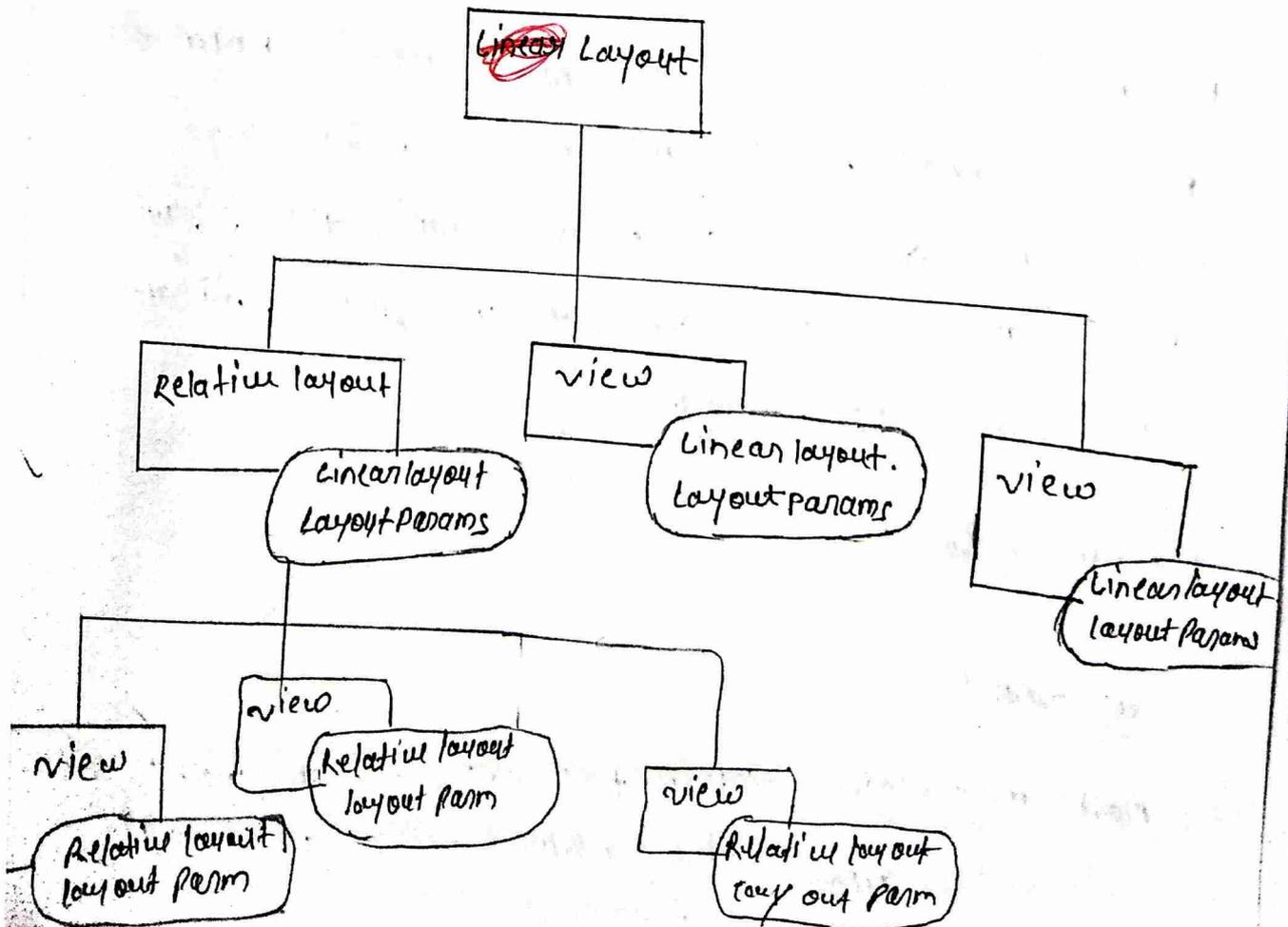
* Platform tools :- Platform tool जब हम नया SDK install करते हैं तब यह update होते हैं। Platform tool का update करने Platform की साथ पिछड़ा है।

* Layout Resources:-

एक layout resource एक activity के UI के लिए architecture या UI के एक component को परिभाषित करता है।

File location: res/layout/ filename.xml.

File name का उपयोग resource ID को रूप में लिया जाता है।



अंश 1 में परिवर्तनी *

तीसरे स्तर पर हमारे पास डाटा-2 layout है जो view group class के subclass है।

एक layout में किसी भी प्रकार के विजेट जैसे - button, text box, table etc बनाए जा सकते हैं।

एक बार layout बन जाने के बाद, हम अपने जातिविधि code में layout संभाषन को अपने activity, on.create() में call back कर सकते हैं।

* Android layout Types:-

(i) Linear layout (ii) Relative layout (iii) Table layout

(iv) Absolute layout (v) Frame layout (vi) List view

(vii) Grid view

* Layout Attribute :-

प्रत्येक layout में Attributes का एक set होता है जो उस layout के visual properties को परिभाषित करता है।

- (i) Android:TD :- यह वह ID है जो view को पहचानती है।
- (ii) Android:layout_width
- (iii) Android:layout_height
- (iv) Android:layout_margin :- यह चारों तरफ extra space में विस्तार
- (v) Android:layout_gravity :- यह specify करता है कि child view को positioned किया जाए है।
- (vi) Android:layout_weight :- यह specify करता है कि layout में कितना अतिरिक्त स्थान view को दिया जाना चाहिए।
- (vii) android:layout_x :- coordinate को specify करता है।
- (viii) Android:padding :- layout के लिए अतिरिक्त स्थान को specify करता है।

* Android Fundamentals and user Interface design *

* Activities and Activity Lifecycle:-

यह केवल एक class है जो single screen display करता है। वे UI को निर्देशित करते हैं और smart phone screen पर user की interaction को संभालते हैं।

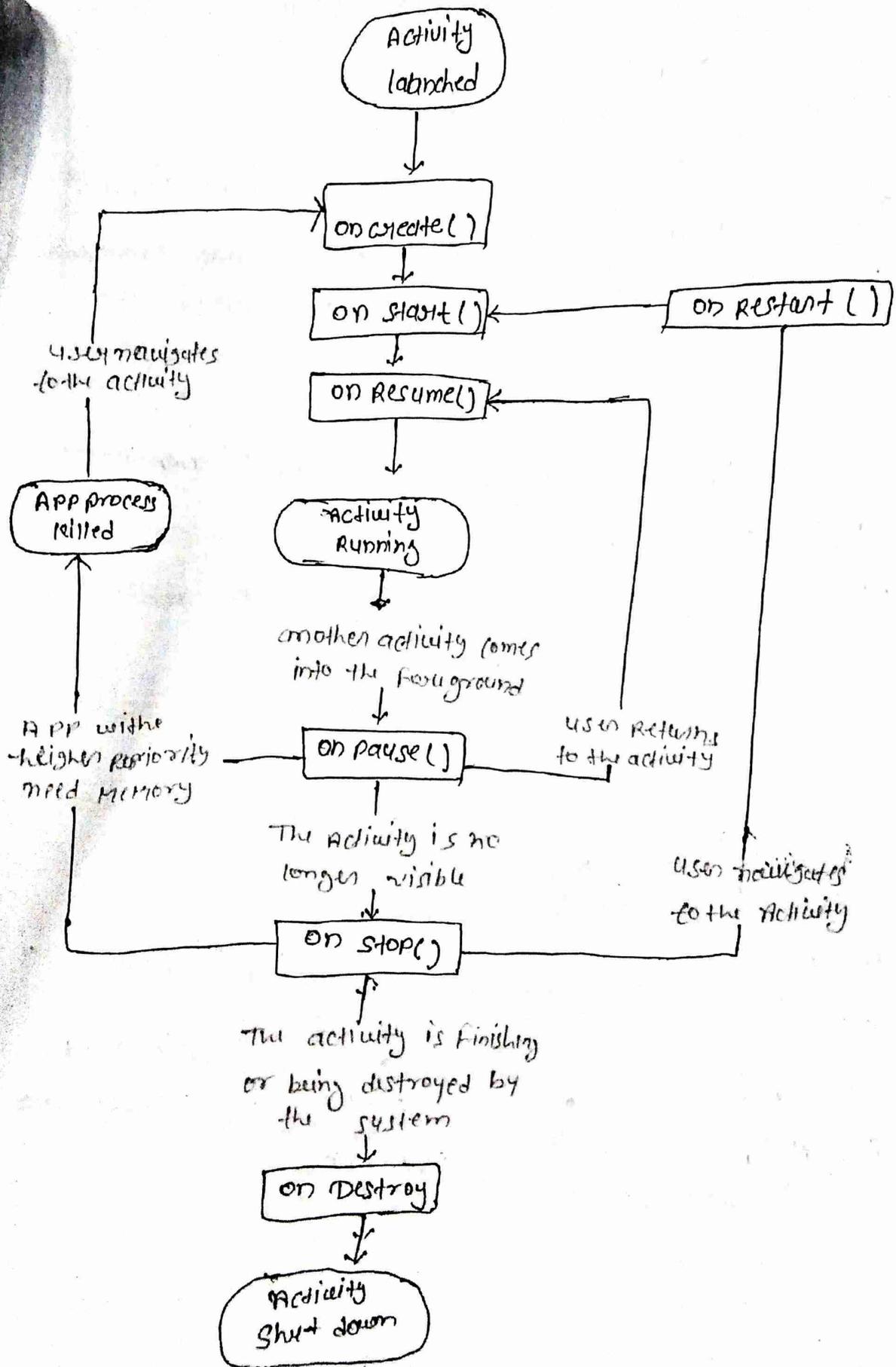
यह AWT में frame की तरह है एक activity की activity class के subclass के रूप में implemented किया जाता है।

syntax:- public class MainActivity extends Activity
{
.
.
}

* Activity Life cycle:-

Android system onCreate() callback method पर call के साथ शुरू होने वाली activity में अपना program शुरू करता है।

callback method का एक sequence है जो activity को शुरू करता है।



* पृथग गौ गुरवती *

* permissions on android :-

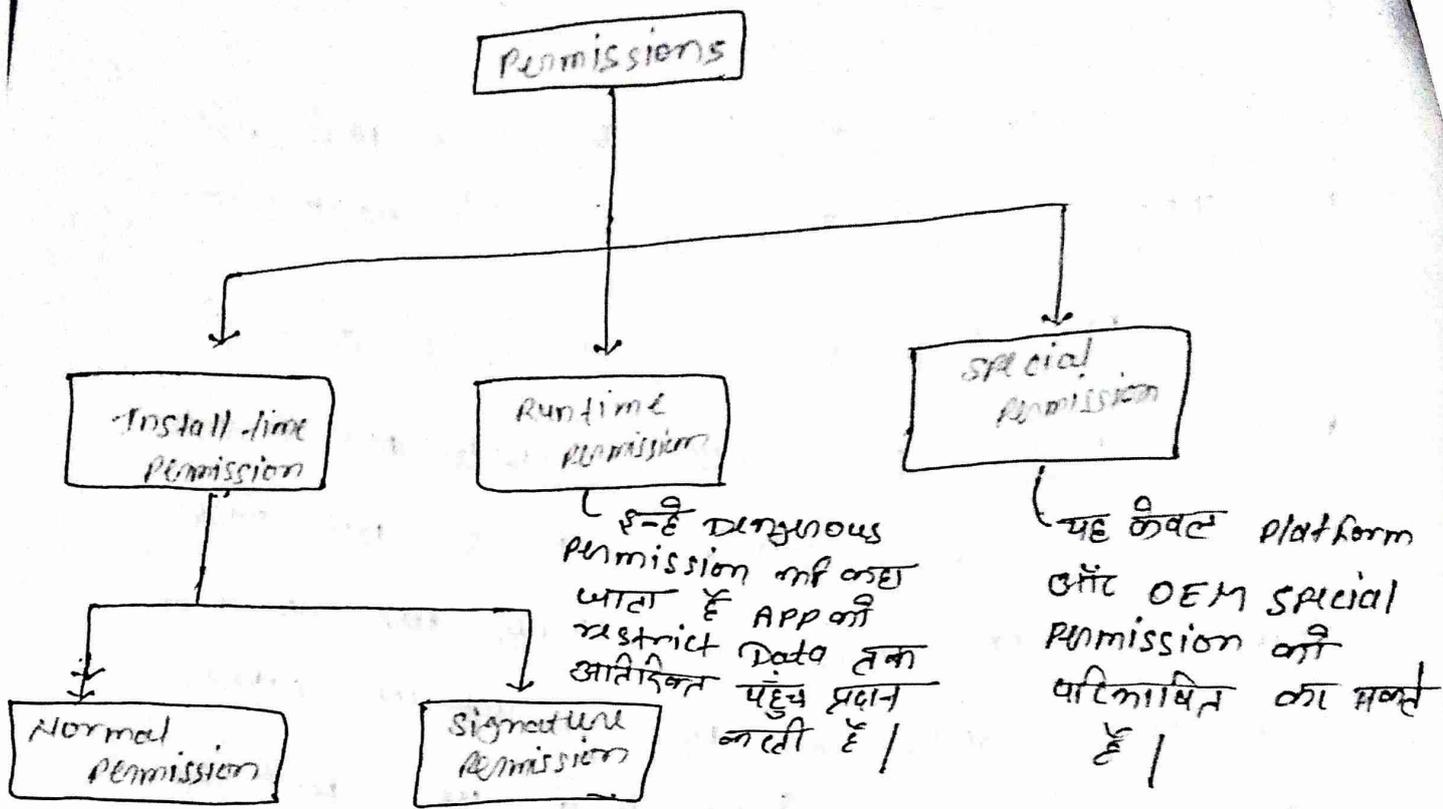
permission शब्द का ~~अर्थ~~ आधिकारिक अर्थ है किसी व्यक्ति की किसी special कार्य को करने की permission देना।

किसी भी computer की file system को सीधे access करना एक महत्वपूर्ण security risk हो सकता है इसलिए यह नियंत्रित करने के लिए पर्याप्त सुरक्षा है कि कोई भी malicious app user के data पर चले इसके लिए permission provide की जाती है। जिसका (उपयोग) करके user किसी भी app को रोक सकता है।

App permission access की सुरक्षा करके user की गोपनीयता का समर्थन करने में मदद करती है।

1. Restricted data जैसे कि system state & a user's constant information.

2. Restricted action जैसे कि connecting to a paired device और recording audio.



यह उन को data और कार्य की permission देती है जो आपके APP को send हो सके हैं beyond हैं।

* Basic UI Components:-

Android UI Control Android के 3 Component हैं जो UI को अधिक Interactive तरीके से Design करने के लिए उपयोग किए जाते हैं। Android हमें कई प्रकार के UI Component प्रदान करता है :-

* नाम गी सरवती *

* III Text view

II Edit view

I Button

II Radio Button

I Image view for Image

I Check bar

II Progress Bar.

* Text view:-

Text view एक UI component है जो text को user को अपनी Display screen पर दिखाता है। इसे हम दो तरीके से बना सकते हैं।

I XML File.

II Activity File.

* Edit text:-

यह एक user interface control है जो user को कुछ text enter करने की अनुमति देता है।

* Button:- यह एक UI है जो click करने पर कोई action perform करता है।

* Radio button:-

इसमें केवल दो possible states हैं जो या तो ~~check~~ check या uncheck किए जाते हैं।

* Image button:-

यह एक Button के समान है, लेकिन यह एक कार्य करने के लिए button या एक image प्रदर्शित करता है। हमें हमें image source देने की आवश्यकता है ताकि system उसे load कर सके।

* check box:-

यह एक UI control है जिसमें two states हैं। यह या तो check या uncheck किए जाते हैं।

* Progress Bar:-

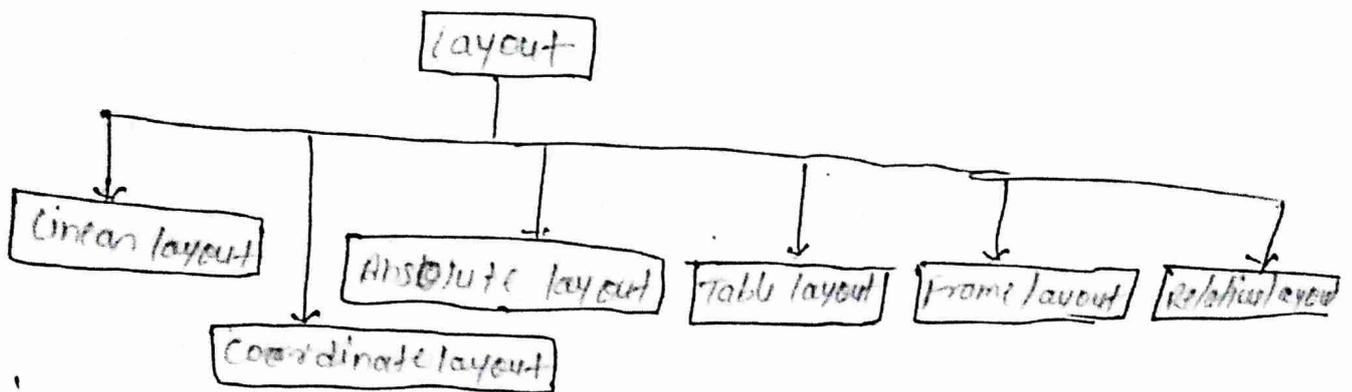
Android में ~~एक~~ progress bar होती है जो कि some action की progress को दिखाती है एक progress bar में दो mode हो सकते हैं।

1. Determinate Mode

2. Indeterminate Mode

Layouts:-

Layout का उपयोग किसी app या activity के लिए user interface को परिभाषित करने के लिए किया जाता है।
और यह UI को रखने पर users को दिखाई देगा।



* Linear layout:-

यह एक single row or single column में view को arrange कर सकता है।

* Absolute layout:-

यह layout को अपने children के exact location को specify करने में enable करता है।

* Table layout:-

table layout सप्ते 3 rows और column में विभाजित है।

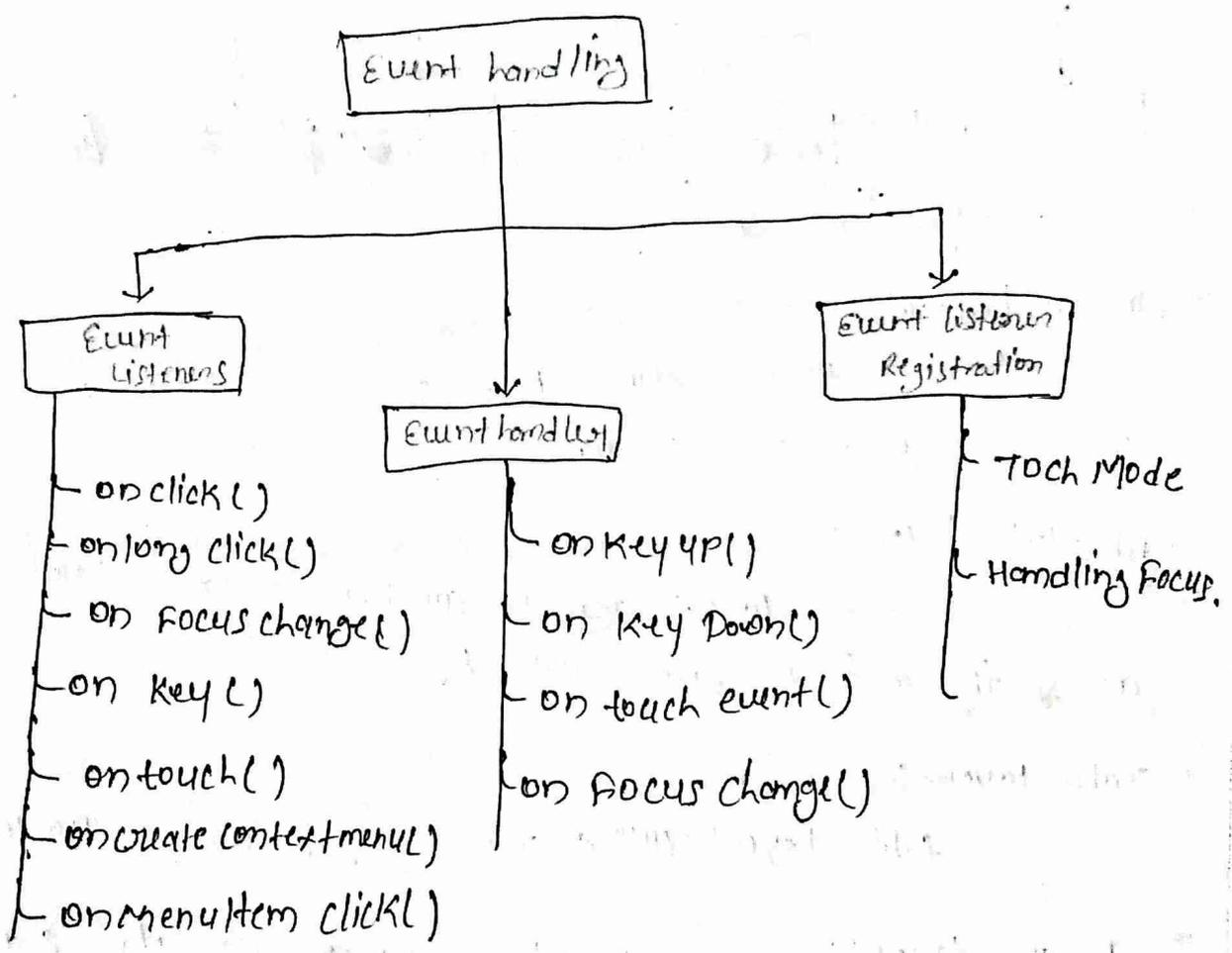
* Frame layout:- Frame layout screen पर placeholder है जिस पर single view के लिए उपयोग कर सकते है।

* Relative layout:-

Relative layout हमें यह specify करने में enable करता है child views एक दूसरे के सापेक्ष में position करे है।

* Event handling In Android:

Event handling, event और handling के साथ आता है
 यहाँ event और action के अलावा कुछ भी है।
 यहाँ handling का मतलब Manage करना है।



1. Event Listeners android के view class में एक interface है यह एक single call back method है।
2. event listeners, event handler को call करते हैं। जिनके द्वारा कुछ action perform किया जाता है।
3. Event listener registration वह process है जिसमें event listeners & event handler को साथ जोड़े register होते हैं।

* Coordinate layout :-

यह layout की parent child view के साथ coordinate dependencies करने की क्षमता है। यह नया layout behaviour से परिभाषित नियमों के एक set पर आधारित है। Coordinator layout, Frame layout की FEATURES को विस्तार करता है। ये दो primary use cases हैं :-

- (i) एक top level applications decor के रूप में।
- (ii) एक या अधिक child view के साथ एक structured interaction के लिए containers के रूप में।

* Unit of Measurement :-

UI पर किसी element की size specify कर रहे हैं।

units :-

- (i) dp/dip :- Density-independent pixel → एक abstract unit जो screen की physical density पर आधारित है। एक dp एक 160 dpi स्क्रीन पर पिक्सेल के बराबर है।
- (ii) sp :- scale independent pixel :- यह एक dp के समान है और font size specify करने के लिए recommended है।
- (iii) pt & points :- screen की physical size के आधार पर एक inch को 1/72
- (iv) px :- pixels :- screen पर actual pixels के corresponds.

* Intents:-

एक intent एक msging obj. है जिहका उपयोग
एक कोणी another APP में Action perform करने के लिए
कर सकते हैं।

* Importance:-

Intents कई ~~व्यक्ति~~ लाइकी से components के
मध्य communication की सुविधा प्रदान करते हैं, तीन
fundamental उपयोग के मामले निम्न हैं:-

(i) Starting in Activity :-

एक Activity ~~screen~~ में एक APP
में एक screen को represent करती है।

(ii) Starting a service :-

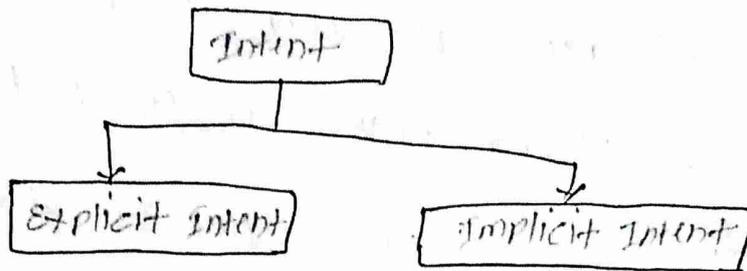
एक service एक component है जो
user interface के बिना background में operation
perform करती है। यह Job scheduler के साथ एक
service start कर सकते हैं।

(iii) Delivering a broadcast :-

Broad cast एक msg है जिसे
कौड़ी भी APP receive कर सकता है।

* भाग गौ सरावती *

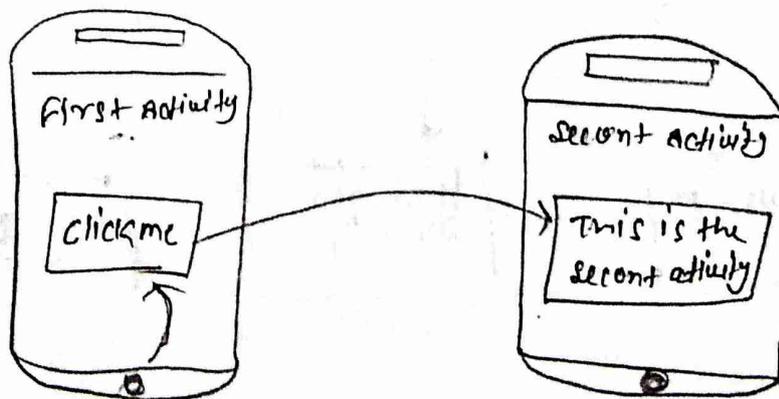
* Intent Type :-



* Explicit Intent :-

Explicit Intent में specify करते हैं कि किस-सा target APP'S package name या एक fully-qualified components class name का प्रयोग करके, intent को satisfy करेगा।

Explicit intent Application की internal work में पूरा हुआ है।

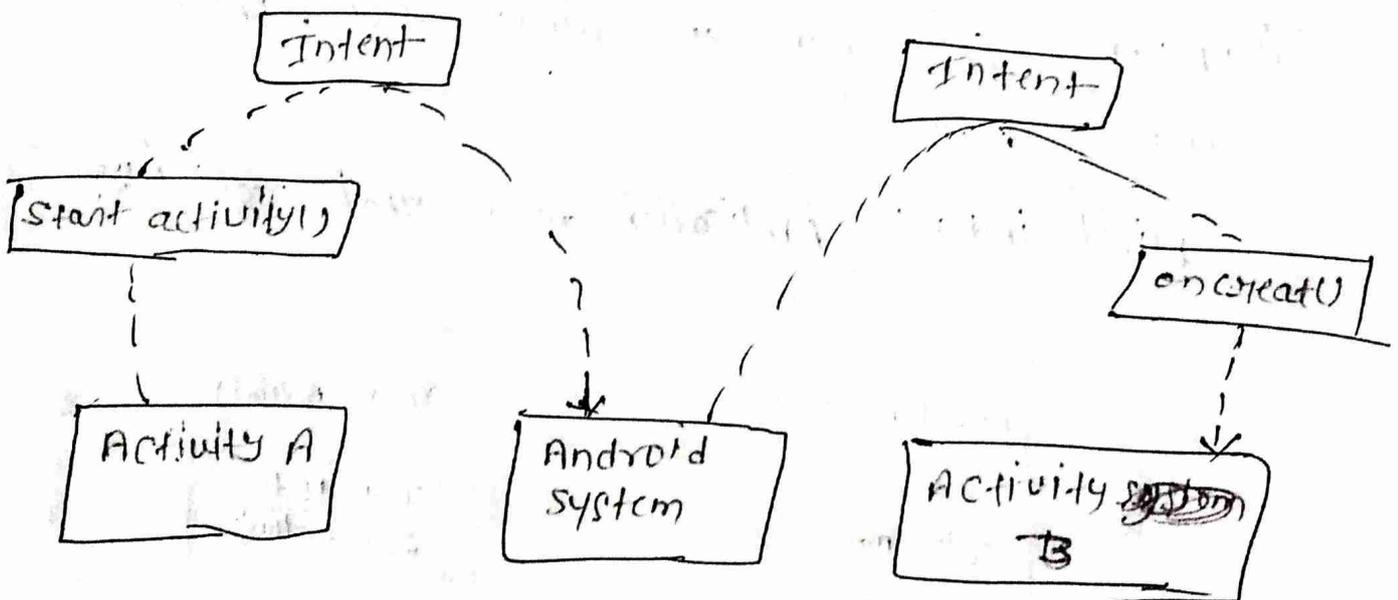


ये intents अपने name से target component को जागृत करे हैं।

* Implicit intents:

ये intent किसी specific activity या intent को प्रारंभ करने के लिए other component को specify नहीं करते हैं। इसके बजाय, action perform करने के लिए एक general action को declare करते हैं।

जब intent object किसी specific activity component को explicitly रूप में name देता है तो system उस component को शुरू करता है।

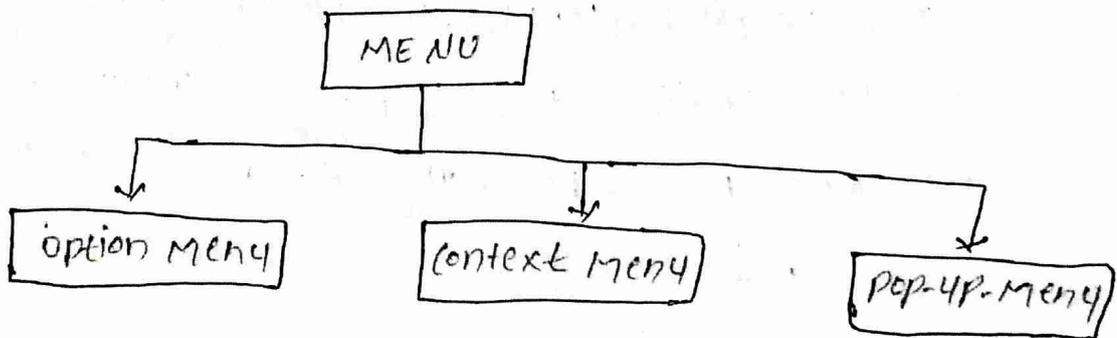


* Introduction to Menu:-

Android में menu user interface component का हिस्सा है जिसका उपयोग Application के साथ-साथ कुछ सामान्य functionality को संभालने के लिए किया जाता है।

Android Application Activities में user actions और अन्य options को Represent करने के लिए menu APIs का उपयोग कर सकते हैं।

* Android Menu के प्रकार:-



* option menu:-

option menu एक जातिविधि के लिए menu item का प्राथमिक संग्रह है। option menu एक जातिविधि के लिए विकल्पों का एक संग्रह है। इसके उन विकल्पों का set है जो क्रिया करने के लिए उपयोगी है।

* POP-UP-MENU :- POP-UP-MENU एक MENU है
जो pop-up window में Items की
सूची प्रदर्शित करता है।

POP-UP-MENU default रूप से view के नीचे दिखाई
देता है, यदि कोई स्थान नहीं है तो उसके ऊपर
दिखाई देता है।

* Contextual MENU :-

एक contextual MENU एक floating MENU है,
यह तभी दिखाई देता है जब user किसी element
को long press तक दबाते हैं या उस पर right
click करते हैं।

यह आमतौर पर selected ~~MENU~~ element को
प्रभावित करता है।

* भाग गों गुरुवती *

* Shared Preferences:-

Android किसी Application के Data को store करने के कई तरीके प्रदान करता है। इनमें से एक shared preference है।

Shared Preferences दो तरह के key, value pair के रूप में Data को store और retrieve करने की अनुमति देता है।

* Preference File:-

एक preference file वास्तव में device की internal memory ~~data को store~~ में save कि गई एक XML file है।

एक Application के पास कुछ data होता है जो किसी directory में save होता है, इसलिए अब मैं getSharedPreferences() function को call किया जाता है जो यह file को validates करता है कि मौजूद है या नहीं अगर नहीं है तो एक नया XML, passed के name से बनाया जाता है।

Should Preferences should preferences =

or should preferences (MY PREFERENCES, CONTROL MODE,
PRIVATE);

Should preference.Editor ^{class} का उपयोग करके Should Preference
में कुछ सार्वकालिक होते हैं।

Should preference instance return file का इंगित करता है
जिसमें values of preferences शामिल हैं।

Advanced OS components

* Time and date on Android :-

orcle database के आधार पर, simple data format एक स्थान-संबंधित तरीके से तिथियों को formatting और parsing करने के लिए एक class है।

Syntax:-

```
import java.text.SimpleDateFormat ;  
import java.util.Date ;
```

* List view :-

Android में List view एक view है जो कई items को समूह बनाता है और इसे vertical scroll करने योग्य सूची में प्रदर्शित करता है। सूची item स्वचालित रूप से Adapter का उपयोग करने सूची में डाला जाता है।

Android में List view, Data को List के रूप में बदलने का काम

List view में List, ~~out~~ List view का उपयोग

करके बनाया जाता है। यह Android के सबसे महत्वपूर्ण

सूची में से एक है।

* विशेषताएं :-

Android में List view की अनुकूलित करने के लिए
एक साथ कुछ विशेषताएं जोड़ सकते हैं।

- (i) ID :- यह विशेष रूप से List की पहचान करता है।
- (ii) Divider :- यह विभिन्न सूची वस्तुओं के मध्य रंग या आकारों
को परिभाषित करता है।
- (iii) Divider height :- यह विभिन्न सूची में मध्य divider height
को परिभाषित करता है।
- (iv) Item list selector :- यह संयोजित Listview के लिए चयनकर्ता उत्पन्न करता है।
- (v) प्रविष्टियाँ :- यह सावणी संसाधन के समर्थन को निर्दिष्ट करता है।
- (vi) पृष्ठभूमि :- यह Listview की पृष्ठभूमि को उत्पन्न करता है।

* Android Grid view :-

यह Layout हमारे पास बहुत महत्वपूर्ण में से एक है।
यह हमें Grid के रूप में व्यवस्थित दिखाने में मदद
करता है। एक ग्रिड मूल रूप से 2-D दृश्य में वस्तुओं
की सूची दिखाता है। यदि Item की सूची screen
पर fit नहीं हो रही है तो इसके लिए scroll
option भी दिया जा सकता है।

Grid, Android Grid view की तरह से बनाया जाता
है जो containers में मौजूद है।

* विशेषताएं :-

Aruid layout को अनुकूलित करने के लिए हम निम्नलिखित विशेषताओं का उपयोग करते हैं।

- (i) ID :- Aruid के पहचान करने के लिए उपयोग प्रोबली जाती हैं।
- (ii) Accuracy :- यह विशेषता Aruid view के संरेखण को संत करता है।
- (iii) Columnwidth :- इसका उपयोग column की एक निश्चित आकार संत करने के लिए किया जाता है।
- (iv) stretch mode :- इसका उपयोग किसी भी वाली पंक्त को भरने और फैलाने के लिए किया जाता है।
- (v) स्थिति अनुद :- यह विशेषता त्तरों के मध्य blank को परिभाषित करता है।
- (vi) new column :- यह विशेषता उन rows या photos की संख्या को परिभाषित करते हैं जिन्हें नए column में व्यवस्थित किया जाए।
- (vii) ~~the~~ virtual difference :- यह विशेषता पंक्तियों के मध्य दूरी को परिभाषित करता है।

* Card view:-

यह एक नया widget है जिसका उपयोग एक विशेष डिजाई के साथ गोल (rounded) layout प्रदान करके किसी भी प्रकार के content को प्रदर्शित करने के लिए किया जा सकता है।

Cardview वह दृश्य है जो एक इतरे के अंदर दृश्य प्रदर्शित करता है। इसका उपयोग यह है कि यह UI Design को समझ में रखने और देखने में मदद करता है। यह flexbox layout को विस्तार करता है और इसे सभी platform पर प्रदर्शित किया जा सकता है।

* Recycler view:-

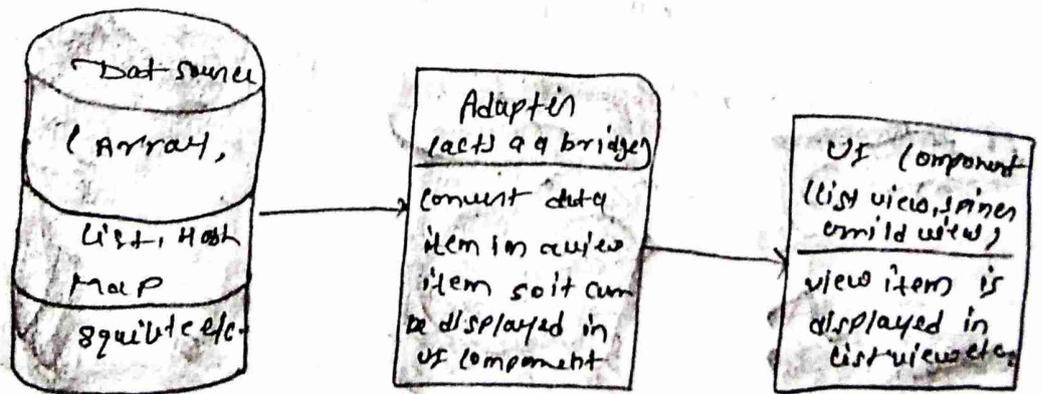
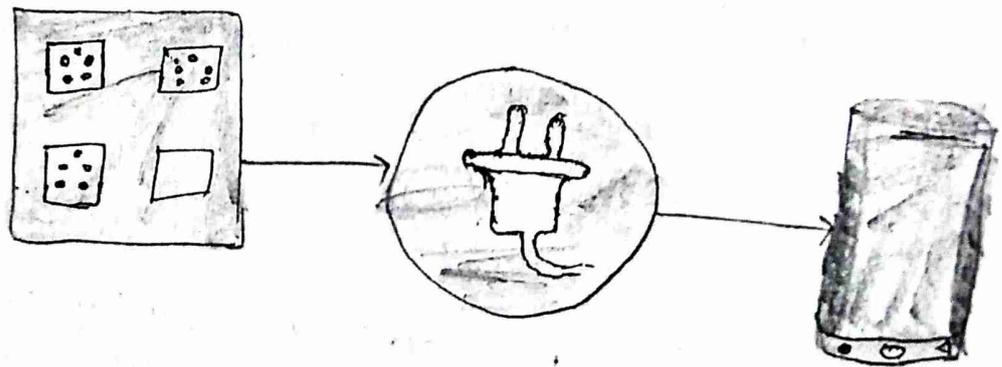
यह एक view group है, जो एक item के रूप में XML सिंगल layout के साथ किसी भी दृश्य का संभव सिंगल के लिए बनाया गया है। Listview & GridView की दृष्टि में काफी लुभाए जाते हैं।
इसे ~~सिगल~~ अनुकूलित किया जा सकता है।

* जग मां सरस्वती *

- * **working:-** एक बुनियादी recycle view को लागू करने के लिए तीन-उप-बाणों का निर्माण करने की आवश्यकता होती है।
- * **Card layout:-** यह एक extend layout है जिसे recycle view द्वारा बनाई गयी छवि के लिए एक item के रूप में माना जाएगा।
- * **view holder:-** यह एक Java class है जो card layout दृश्यों को एक संदर्भ को संग्रहीत करता है।
- * **Data class:-** Data class एक custom Java class है जो recycle view के हर item के लिए जानकारी रखने के लिए एक संरचना के रूप में कार्य करती है।

* Adapter:-

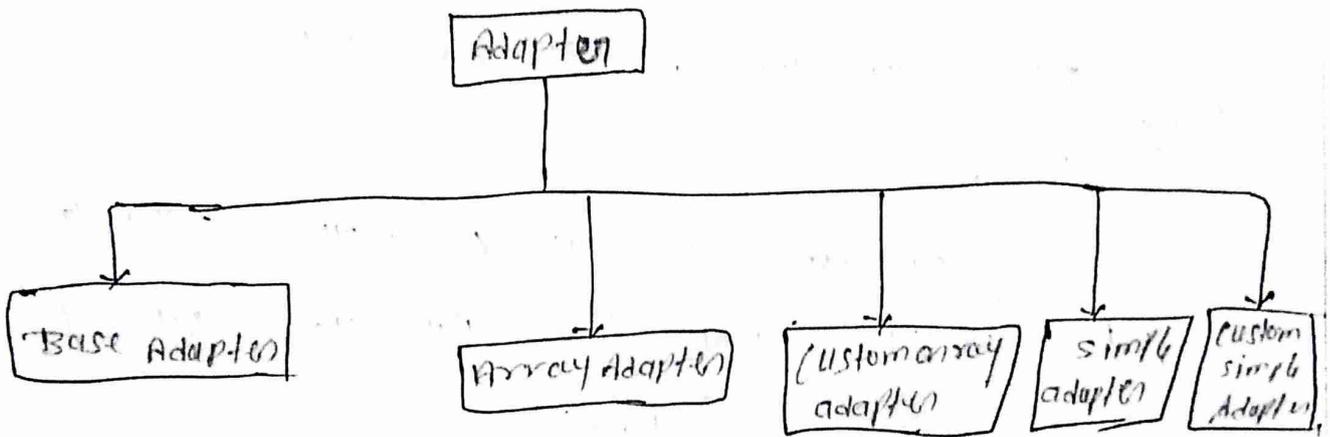
Android में, Adapter UI बटक और Data स्रोत के मध्य एक पुल है जो हमें UI बटक में Data भरने में मदद करता है यह Data को रखता फिर उसे Adapter view में send कर देता है, फिर view adapter में data ले सकता है।



किसी भी list or array में data करने के लिए हमें Adapter की आवश्यकता होती है।

* Android में Adapters:-

एक डेटा को से प्राप्त करने के लिए उपयोग किए जाने वाले Adapter में कुछ आमतौर पर उपयोग किए जाने वाले Adapters:



* Base Adapter:-

Base Adapter एक Adapter के सामान्य कार्यान्वयन का एक सामान्य आधार class है जिसका उपयोग ListView, GridView, Spinner आदि में किया जा सकता है।

* Array Adapter:-

जब भी हमारे पास एकल वस्तुओं की सूची होती है जो किसी सारणी द्वारा समर्थित होती है तो हम Array Adapter का उपयोग कर सकते हैं जैसे - phone books, etc. Array Adapter, Base Adapter का कार्यान्वयन है।

* TOAST :-

Android toast का उपयोग का समय के लिए जानकारी प्रदर्शित करने के लिए किया जा सकता है एक toast संदेश होता है जो जल्दी प्रदर्शित किया जाता है और कुछ समय बाद गायब हो जाता है।

Android.widget.Toast CLASS java.lang.Object class का subclass है।

• हम custom toast भी बना सकते हैं।

Toast class :-

Toast class का उपयोग एक विशेष अंतराल के लिए आधी घंटा दिखाने के लिए किया जाता है।

toast class के constant :-

public static final int LENGTH_LONG - लम्बे समय के लिए दृश्य प्रदर्शित

public static final int LENGTH_SHORT - कम समय के लिए दृश्य प्रदर्शित

Method of toast class

public static Toast makeText (Context context, CharSequence text, int duration) toast के पार और अधिकृत बनाता है।

public void show () toast प्रदर्शित करता है।

public void setMargin (int margin) margin set के लिए होता है।

* POPUP MENU:-

Android POP-UP MENU का मतलब है कि जब कोई user
Menu प्रदर्शित करता है तो वह 'अंतर्दृष्ट' अर्थात्
लगेर चारों ओर फैलता है जो कि आप POP-UP
को बाहर click करते हैं तो वह गायब हो जाता है।

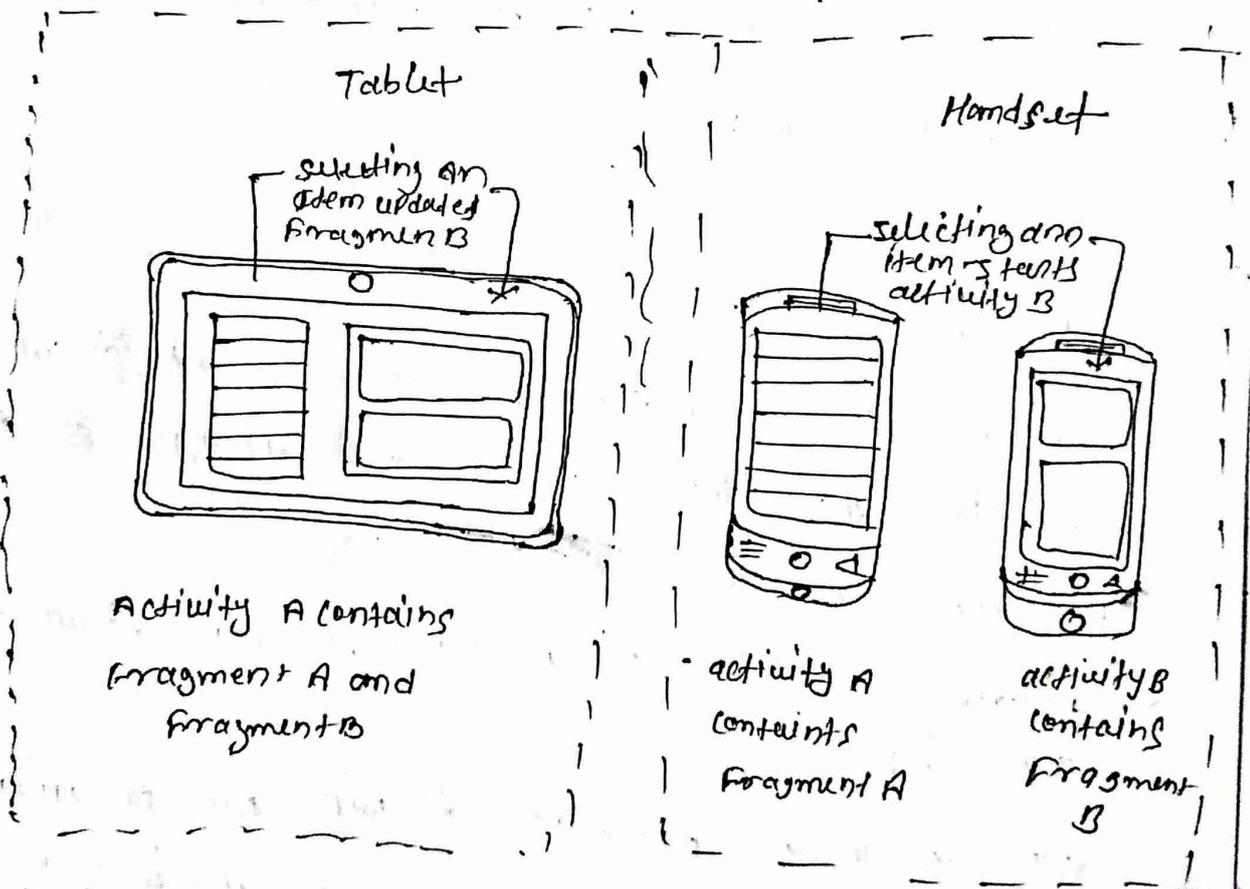
Android.widget.popupmenu.java.lang.object class का
प्रत्यक्ष subclass है।

* Fragment :-

Fragment एक गतिविधि का एक Fragment है जो अधिक
modular & गतिविधि Design करने का सहायक होता है।
Fragment को दो भागों में मटेल्डर्षण बिन्दु

- गतिविधि चलने के दौरान किसी भी Fragment को जोड़ या हटाना
सकता है।
- एक बटु-फलक पर बनाने के लिए एक ही गतिविधि
में कई Fragment को जोड़ सकता है।
- एक Fragment कई गतिविधियों में तैयार किए जा
सकते हैं।

- जब गतिविधि रुकी हुई हो तब तक Fragment को रोक दिया जाएगा ।
- Fragment class का विस्तार उनके Fragment करते हैं
- Android की honeycomb & Generation में Android APP में Fragment जोड़े गए थे ।



Application एक tablet के अकार्ड के device पर चले या गतिविधि A में ही Fragment को Embed कर सकते हैं

* Fragment के प्रकार :-

Fragment तीन भागों में divide किया जाता है /

(i) Single Frame Fragment :-

Single frame के fragment मोबाइल and handheld held devices के लिए use कर रहे हैं /

(ii) List Fragment :-

विशेष list scene वाले fragment को List Fragment कहते हैं /

(iii) Fragment transaction :-

Fragment को जोड़ने के साथ उपयोग करना /

* Android Material Components (MDC)

① Material Design :-

यह वेब और दूसरे Digital उत्पादों के निर्माण के लिए एक सहायक है /

Android के लिए material, एक बरक, application में पिरता बनने के लिए बरकों की library के साथ

design site engineering को एकत्र करता है।
वीडो-2 material design विकसित होती है वीडो-2
नए बटन को फ्रॉन्ट में @ update भी हो रहा है।

* text file:-

-एक लोग उपयोग करने का नाम और password add करने के लिए login page पर दो text field लगाएंगे।
और मॉडल text field बचक का उपयोग करेंगे।

* Add buttons:-

-एक login page पर दो बटन, Next & Cancel लगाएंगे और
इन्हें मॉडल बचक का उपयोग करेंगे।

* Navigation:-

-एक अपने next button को शुरू करने पर
हल करने के लिए login fragment Java में कुछ Java
code लगाएंगे।

* All done:- Android library के लिए सामग्री बचक ने
login page बनाने में मदद की है और सामग्री design &
विकसित के अंतर्गत है और सभी उपकरण में लगाता
-विद्यमान है।

Introduction
of
multithreading

* 1/11 of quality *

program, variable
memory

* Threads in Android *

(6)

* Threads:

Each thread has its own local variables & memory. Each thread is a lightweight process & has its own individual process. In Android, each thread has its own memory & is independent of the main thread. Each thread has its own stack & heap memory. Each thread has its own set of registers & program counter.

Android has a threading component & is designed to be efficient & easy to use.

* Threads that are attached to an activity/fragment

* Threads that are not attached to any activity/fragment.

* Types of threads:

Android development is the process of creating threads that are used to perform tasks.

* I Main thread

II UI Thread

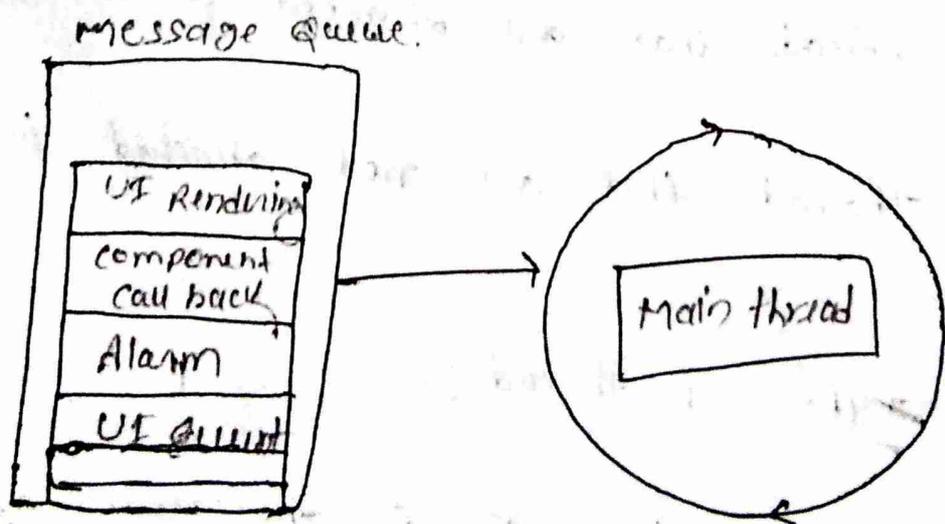
III Worker thread

IV Any thread

V Binder thread.

* Main thread:-

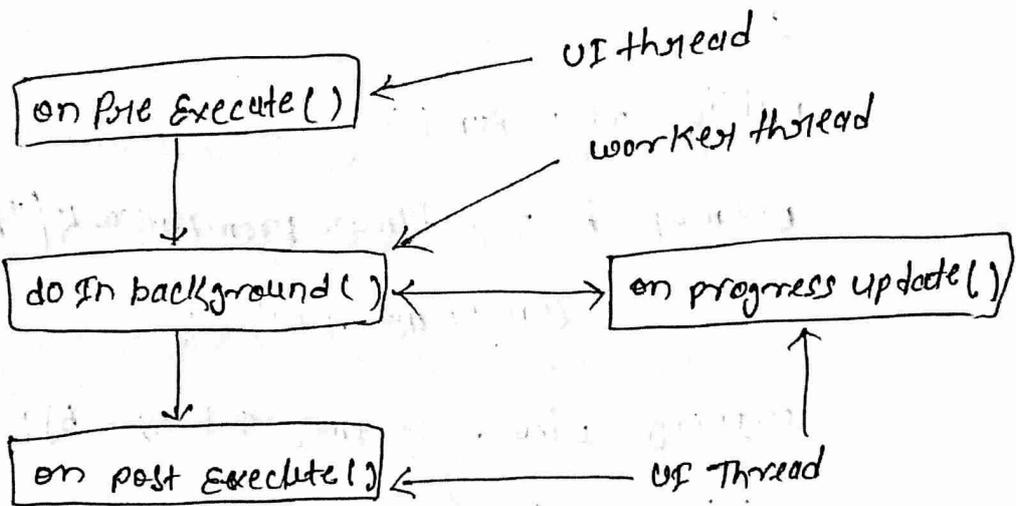
जब कोई Application Android में launch किया जाता है, तो यह execution का पहला thread होता है जिसे main thread के रूप में जाना जाता है।



* भाग में प्रकृति *

* UI Thread:

जब एक UI thread Android APP start होता है तो यह Main thread बनता है जिसे लेकर UI thread कहते हैं।



UI thread हमारे Application में Execution our main thread है।

* worker threads :-

worker thread background thread है यह UI thread
है और UI thread के अलावा अलग में
कार्य करे है

worker thread का निर्माण और कार्य करने का उदाहरण

```
public void onClick (view v) { newThread (new Runnable() {
```

```
public void run() {
```

```
Bitmap b = loadImageFromNetwork ("http://example.  
com/image.png");
```

```
ImageView view. setImageBitmap (b); }
```

```
}).start(); }
```

UI thread में image view पर worker thread में manipulate
कर रहा है)

Result :- वह Method के माध्यम से specified

Runnable एक्समपल UI thread पर run होगी यदि
वर्तमान thread UI में है तो एक्समपल लूट करे execute होगी.

अन्यथा Action event queue में भेजी जाएगी |

* Any thread :-

Annotation method को Any thread में call किया जा सकता है यदि Annotation method element एक class है जो class में सभी methods को किसी भी thread में call किया जा सकता है ।

① Target ([Annotation Target.FUNCTION ;

Annotation Target.PROPERTY_GETTER,

Annotation Target.PROPERTY_SETTER,

Annotation Target.CONSTRUCTOR,

Annotation Target.CLASS, Annotation Target.FILE,

Annotation Target.VALUE_PARAMETER]) .

CLASS Any Thread .

इस method का main उद्देश्य यह इंगित करना है कि आप जानते हैं कि method को Any Thread से call किया जा सकता है ।

Example :-

```
<code>
```

```
@Any thread
```

```
public void deliverResult (D data) { --- }
```

```
</code>
```

* Running on UI Thread :-

Android Application में एक main thread होता है जिस UI Thread भी कहा जाता है।

Application components भी SAME PROCESS में चलते हैं और default रूप में होता होता है, सभी SAME UI thread का उपयोग करते हैं।

Android, UI thread method का चलने की guaranteed वाली कोई method प्रदान करता है जो method में दो नियम लिखित हैं।

* The view.post method

boolean view.post (new Runnable () {

public void run () {

// UI code goes here

}

});

* 2. The Activity . Run on UI Thread.

Activity . Run ~~on~~ on UI Thread (Runnable action)

Run on UI Thread (new Runnable {

```
{  
    public void run ()  
    {  
        // UI code goes here  
    }  
};
```

दोनों method Runnable parameter में सपोर्ट करता है

* विभिन्न thread classes.

* 1. ASYNC TASK :- helps get work on/off the UI thread

* 2. Handler thread :- thread for callbacks

* 3. Thread pool Executor :- Running lots of parallel work

* 4. Intent service :- helps get intents off the UI thread.

* Handlers & Runnable:

* Handlers:

दो thread के बीच communication करने के लिए एक handler का प्रयोग किया जाता है।

Handler दो एक thread के message queue में डालने process message और Runnable object को अपडेट की अनुमति देता है। प्रत्येक handlers instance एक thread और उन method thread की message queue में डाला होता है।

Handler एक component है जिसे एक thread में जोड़ा जा सकता है और जो simple messages या ~~queue~~ Runnable task के माध्यम से उन thread पर Action के लिए बनाया जाता है।

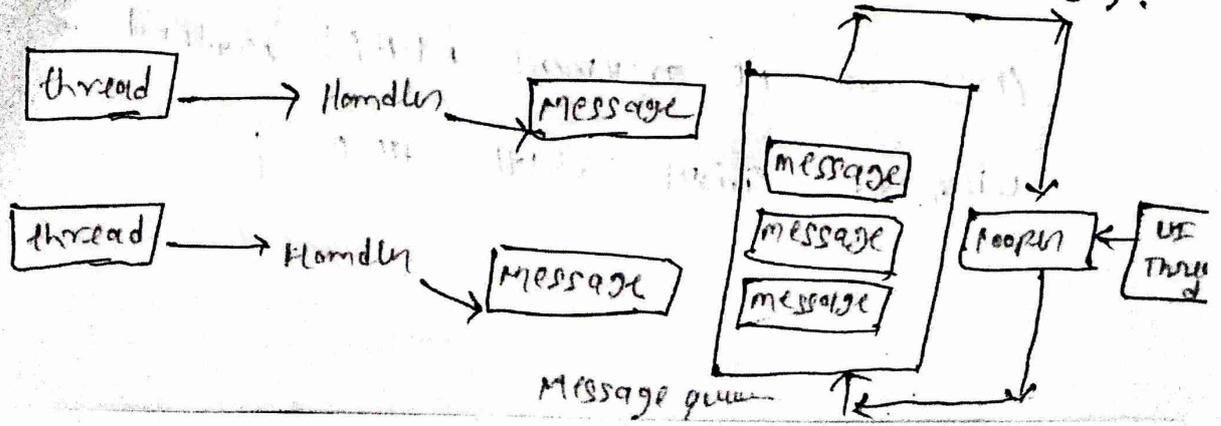
उपगमों का उपयोग

एक handler के दो मुख्य उपयोग :-

- 1. भाषण में किसी बिन्दु पर execute किए जाने वाले message और runnables को schedule करने के लिए।
 - 2. अपने से अलग thread पर perform किए जाने वाले action को enqueue करना।
- Scheduling message को किस function के द्वारा किया जाता है।

POST :- post (Runnable), post At Time (Runnable, long)
post delayed (Runnable, long).

Send Message :- send Empty message (int), send message (message), send message At time (message, long) and send message Delayed (message, long).



* Runnable :-

Java में interface Runnable मूल रूप से Abstraction for an executable command है।

यह एक interface को कहते हैं और इसे java.lang package में परिभाषित किया गया है और इसे abstract method के रूप में run() कहा जाता है।

```
package java.lang;
```

```
public interface Runnable
```

```
{
```

```
public abstract void run();
```

```
}
```

Runnable interface को किसी भी class द्वारा implement किया जाना चाहिए जिसके instances को thread द्वारा execute किया जाना चाहिए।

class को no arguments run() method के द्वारा परिभाषित किया जाना चाहिए।

* Async Task :-

ASYNC TASK Android द्वारा प्रदान किया गया एक Abstract class है जो हमें background में heavy tasks करने के लिए ज़रूरत देता है और UI threads को light में रखता है और हम ~~समस्या~~ ~~सकार~~ Application अधिक responsive बनाता है।

* Android Async Task class में उपयोग किया जाने वाले Basic methods :-

1. doInBackground() :- इस method में कोड शामिल है जिसे background में execute करने की आवश्यकता है।

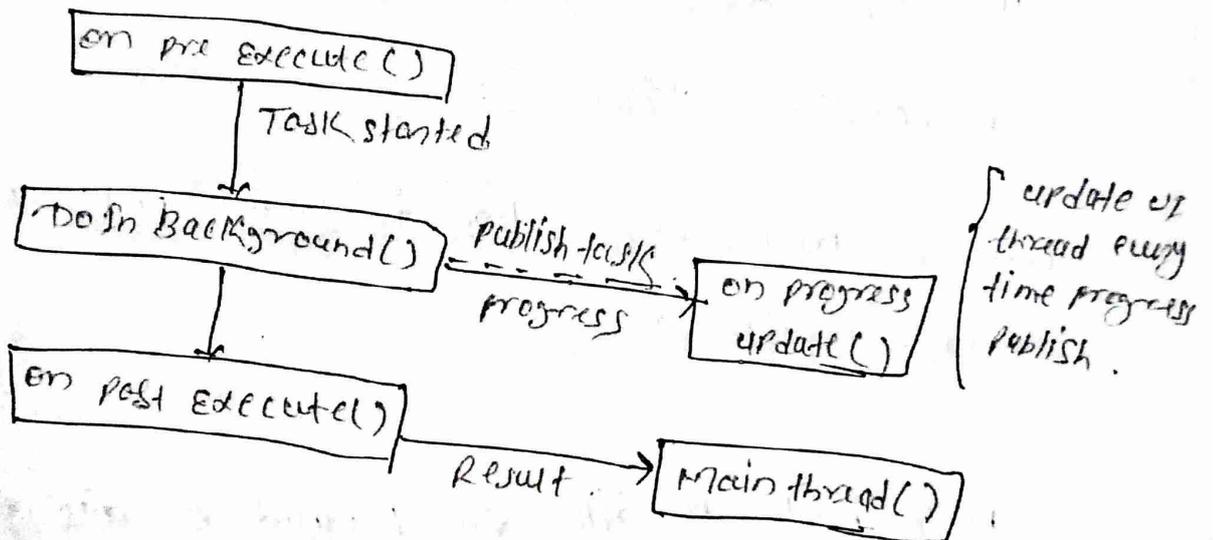
2. onPreExecute() :- इस method में वह कोड होता है जो background start होने से पहले process होता है।

3. onPostExecute()

13. on post execute():

इस method और doInBackground के बीच छेक के बाद call किया जाता है।

14. on progress update: यह method doInbackground method से progress update प्राप्त करती है।



Android AsyncTask से प्रयोग करने वाले तीन generic Topic:-

(i) Programs

(ii) Progress

(iii) Results

* calling web services and consuming JSON Data from web services :-

(a) Android web services :-

web service सभी प्रकार के client Application के लिए इस प्रकार के App server पर कार्य करने के लिए standard प्रदान करती है।

(b) Android web services Component :-

web server Architecture में आप में पर निम्न तीन Roles होते हैं।

(i) Publisher :-

Publisher web service बनाने और client के लिए उपलब्ध करने के लिए जिम्मेदार है।

(ii) Subscriber :-

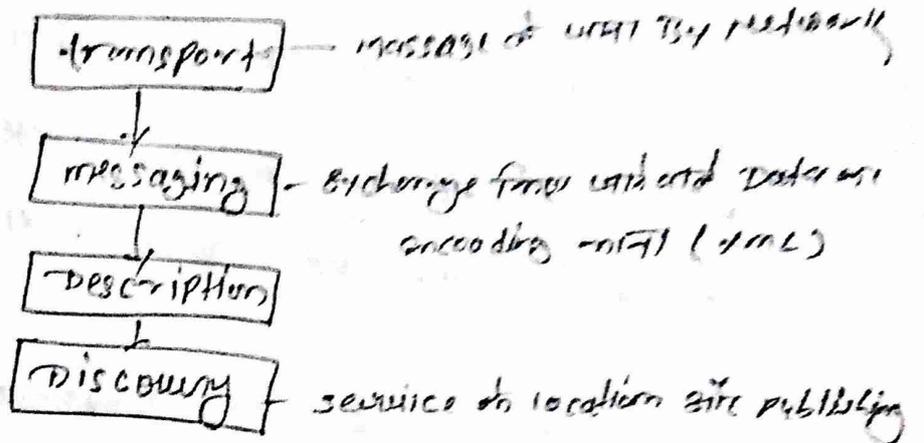
Subscriber कुछ नई service Requester है।

(iii) Broker :-

Broker वह Application है जो UDDI तक पहुंच प्रदान करता है।

* Web Service Architecture:

एक single web service provider को एक logical layers के around design बनाया जाता है।



* Characteristics of web services in Android:-

(i) web service xml आधारित है। वे हमें अपना data representational layer और हमें transport layer पर उपयोग करते हैं।

(ii) web service loosely coupled है।

(iii) web services में synchronous या asynchronous हो सकते हैं।

* भाग गी परवती *

110

web services, remote procedure call का समर्थन करती है।

115 web services में document exchange का समर्थन है।

* Types of web service IN Android:-

111 XML-RPC :- IN XML-RPC, RPC Remote procedural call के लिए है यह एक XML आधारित protocol है।

112 UPPER एका अर्ध सार्वभौमिक वर्णनात्मक, प्लैज और स्कीमिंग। यह XML आधारित standard है।

113 SOAP :- यह एक soap simple object access protocol के लिए है यह XML आधारित web service protocol है।

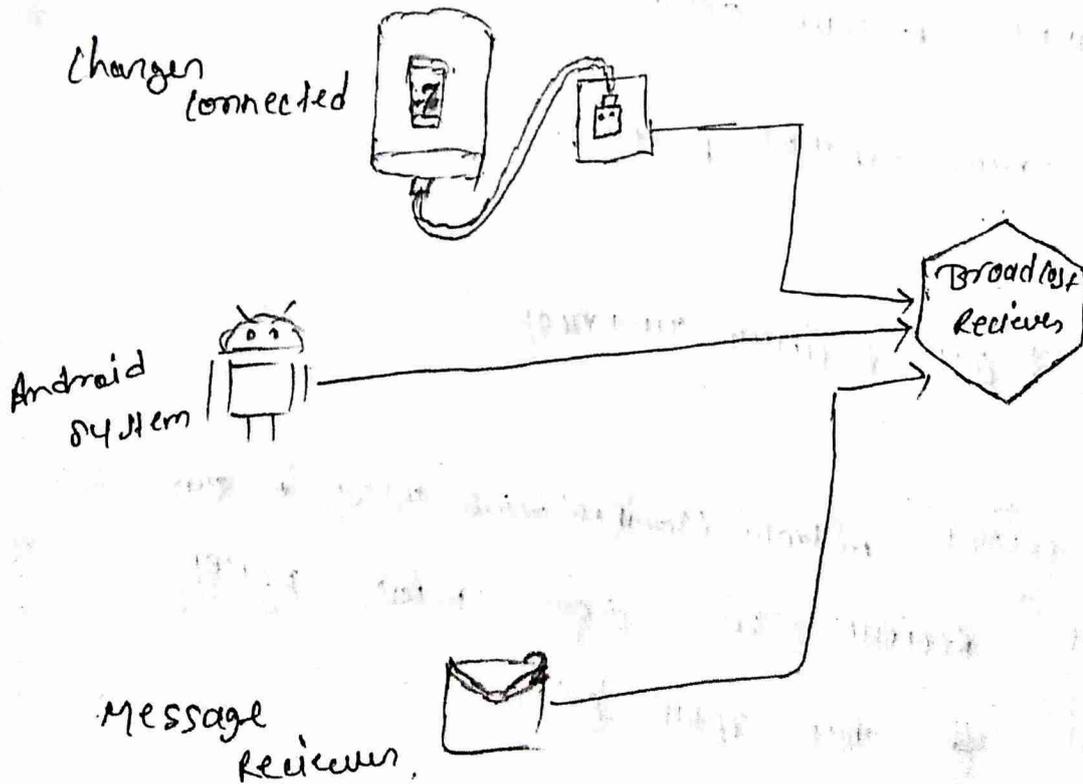
114 REST :- Representational state transfer है यह device और internet के मध्य संवाद और connectivity प्रदान करता है।

* Advantage of web services:-

- (i) विभिन्न प्रकार के Applications के मध्य interoperability को enable करती है
- (ii) web service को standard technology पर deploy किया जाता है
- (iii) वे हैं internet पर मौजूद कोड के कार्यों को उजागर करने की अनुमति देते हैं
- (iv) web services Application और organization को अलग से हैं और अलग से हैं संचार प्रदान करती है

* Broad Cast Receivers:-

Android broad cast Receiver एक android सॉफ्टवेयर है जिसे
उपयोगकर्ता को system या अन्य अनुप्रयोगों पर
समाहित करने के लिए किया जाता है।
समाख्य receivers, गतिविधियों के विषय में, उपयोगकर्ता
Interface है। इसका उपयोग Asynchronous Inter.
process संचार के लिए किया जाता है।



Broad cast receivers उन अनुप्रयोगों में या system में ही प्रसारण संदेशों का जवाब देते हैं इन संदेशों को कुछ समय रोकें या रखा जा सकता है।

System Broadcast किए गए संदेशों के लिए Broadcast Receiver works बनाने के लिए दो महत्वपूर्ण कदमों का पालन कर रहे हैं।

1. प्रसारण receiver बनाना।

2. पंजीकरण प्रसारण receiver.

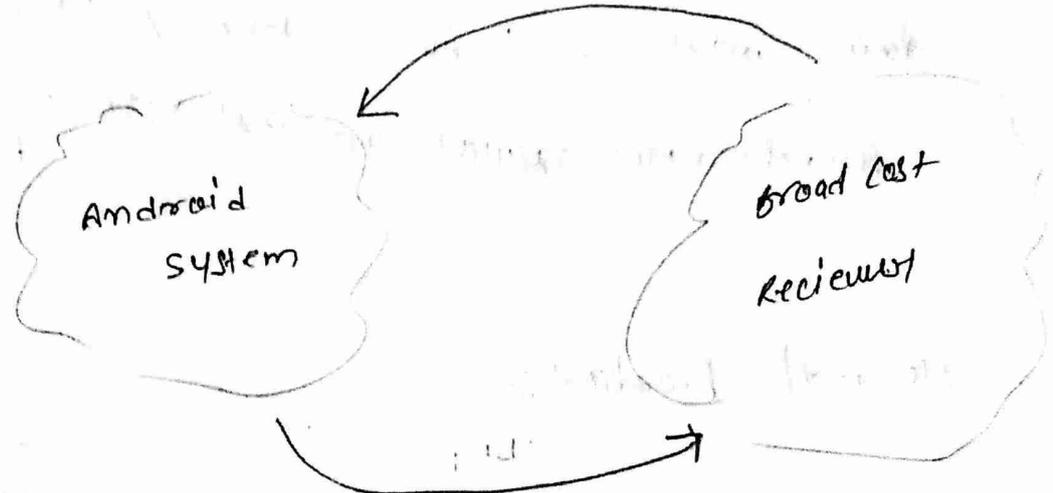
* Broad cast receiver पंजीकरण

एक आवेदन Android Manifest.xml फाइल में एक प्रसारण receiver दर्ज करके विशेष प्रसारण संदेशों के लिए जुड़ा है।

* एप में जरूरी *

ए system भारत event ACTION_BOOT_COMPLETED के लिए एप receiver registration करने जा रहे है।
ए Android system का BOOT प्रक्रिया पूरी करने के बाद system का विकास दिया जाता है।

Registers for intents to observe



Alerts notification when intents occur

* Types of Broadcast



1. Ordered Broadcast

Ordered broadcast समकालिक प्रसारण होता है।
ऑटो उचित क्रम में किए जाते हैं यह
आदेश Android प्राथमिकता विशेषता द्वारा व्य-
क्त किया जाता है। ऑटो आदेशित प्रसारण को
Asynchronous प्रसारण भी कहा जाता है।

2. Normal broadcast

Normal broadcast असमकालिक ऑटो अव्यवस्थित
होता है ये केवल एक समय में, कभी-2
एक अनिश्चित मा सफाई करता है नहीं कारण
है कि वे कुशल हैं लेकिन परिवारों के
एक उपयोग को करती हैं।

* Notifications IN ANDROID:

Notification एक प्रकार का संदेश है, जो तावनी
या किसी Application की स्थिति है जो
Android के UI तत्वों में दिखाई दे रहा
है या उपलब्ध है।

* सामान्य तौर पर किसी Developer द्वारा साफ़ प्रकार की
Android Notification देवी जा सकती है।

i) Status Bar Notification

ii) Notification drawer Notification

iii) Heads-up Notification

iv) Lock screen Notification

* Android Notification Properties:

set small icon () : Notification प्रियन सेट करता है

iii. set content Title(): Notification शीर्षक निर्धारित करने के लिए

iv. set content text(): text set करने के लिए

v. set auto cancel(): Notification खुद को गायब संपत्ति निर्धारित

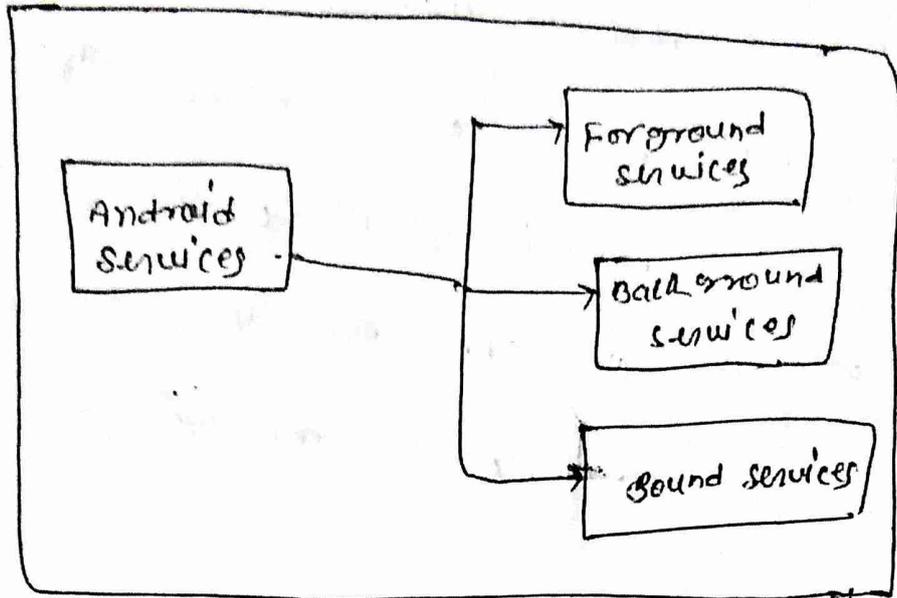
vi. set priority(): Notification प्राथमिकता निर्धारण

* Services In Android:-

Android में services एक विशेष ब्रह्म हैं जो लक्ष्य समय तक में चल रहे operation कार्यों को करने के लिए एक थ्रेड में चलने के लिए एक आवेदन की सुविधा प्रदान करता है।

एक सेवा मुख्य रूप से यह माना जाता है कि आवेदन एक थ्रेड के लिए सक्रिय रहे ताकि उपयोग-कर्ता एक ही समय में कई अनुप्रयोगों को संचालित कर सकें।

* Types of Android services :-



(i)

Foreground services :-

उपयोगकर्ता की स्क्रीन पर रहे कार्यों के बारे में सूचित करने वाली सेवाओं को foreground services कहा जाता है।

(ii)

Background services :-

ये services users द्वारा चल रहे Background कार्यों के बारे में सूचित नहीं किए जा सकते हैं और users भी उन तक नहीं पहुंच सकते हैं।

(iii)

Bound services:-

इस प्रकार Android सेवा आवेदन जैसे घटकों को हमें साथ में शुरू बंद करने की सुविधा देती है। Bound सेवाएँ तब तक अपना कार्य करती हैं जब तक कि कुछ कोई Application Component इसके लिए बंद नहीं है।

*

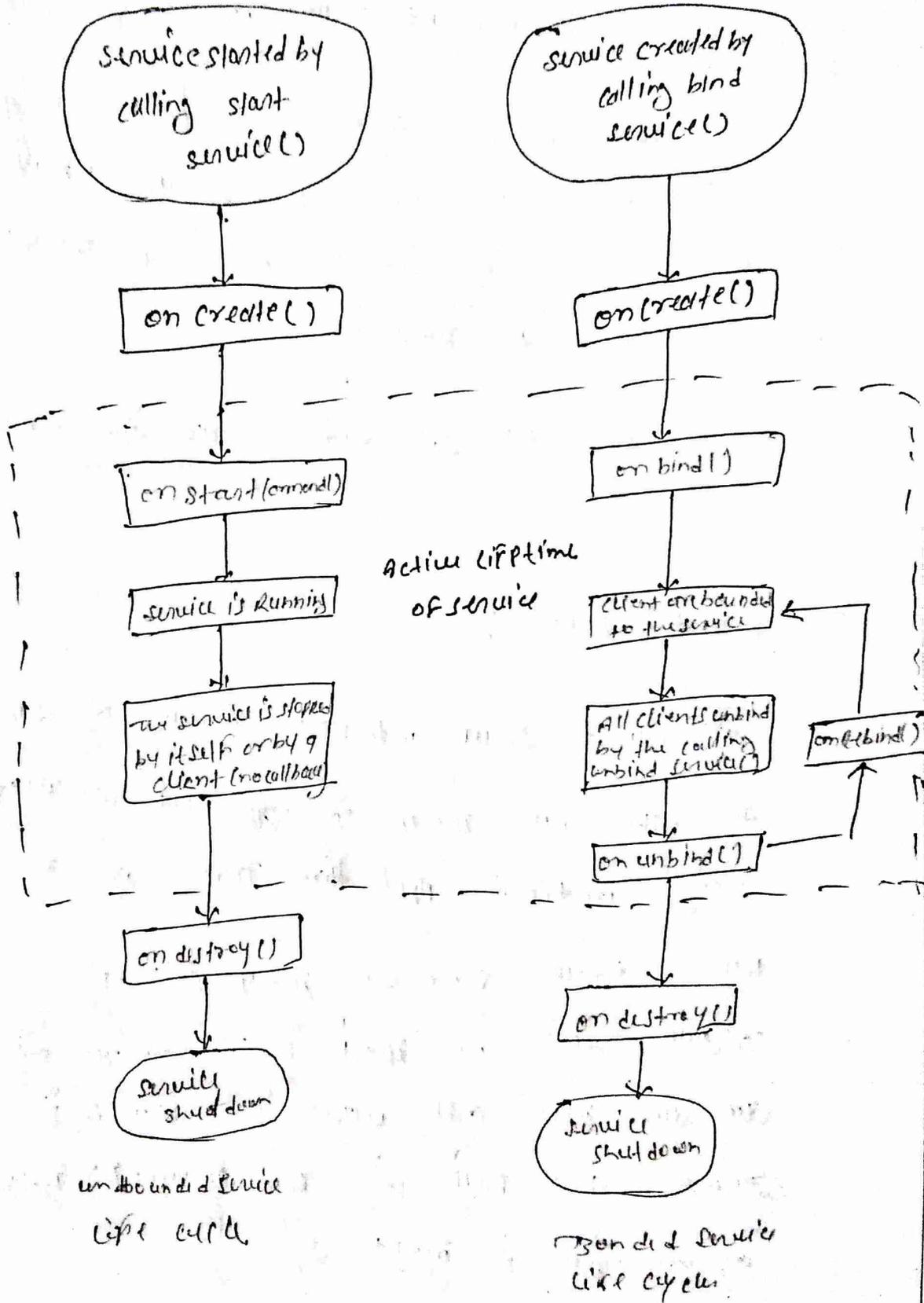
Life cycle of android services:-

Android में अपने सेवाओं के पास जीवन चक्र इस तरह के लिए दो राहें हैं।

(i) Started

(ii) Bounded

* Life cycle of service



* started service (unbounded service):-

इस विधि को call करने, एक सेवा शुरू होगी जब एक बंद आउट service () विधि को एक बार शुरू होने के बाद सेवा प्रकृति में लगातार चल सकती है।

• stop service () call करने से service रोक सकते हैं।

* Bounded service :-

इस client screen interface से सर्वर के रूप में काम करता है इस तरह का पालन करने Android Application बंद सेवा के लिए अनुरोध send कर सकता है और परिणाम प्राप्त कर सकता है। एक सेवा को तब तक बाध्य कर जाता है जब कोई आवेदन बंद स्वयं को bind service () विधि को बंद सेवा में बाध्यता है।

ii) Shared preferences:-

Shared preferences class एक framework प्रदान करता है जो हमारे primitive data types के persistent key-value pairs को save और retrieve करने की अनुमति देता है।

Application के लिए shared preferences object सफ़र करने के लिए दो method हैं:-

iii) get shared preference()::- इसका उपयोग होता है, जब हमें name से पहचानी जाने वाली multiple preferences file की आवश्यकता है, जिसे हम parameters के माध्यम से specify करते हैं।

iv) get preference()::- यदि हमें अपनी activity के लिए केवल एक preference file की आवश्यकता है तो इसका उपयोग करें।

* 2. Internal Storage:-

एक internal storage पर सीधा कार्य सम्भव नहीं है। Default रूप से सम्भव कि सभी file Application के लिए private हैं और अन्य Application के लिए उन्हें access नहीं कर सकते हैं, नहीं user। जब user Application install करता है तो ये files एक पाती है।

Internal storage में private file create और write करने के लिए :-

- (i) file के नाम और operating mode के साथ open file output () function को call करें।
- (ii) file को write करने के लिए write () function को call करें।
- (iii) close करने के लिए close function को call करें।

* Internal storage में file read करने के लिए :-

जो भी file read करनी है उसका नाम open file input () pass करत हुए call करें।

Other use full methods:-

(i) Get dir ()

Internal storage नाम के गीतर directory बताता है।

(ii) delete file ()

Internal storage पर saved कि सभी file delete करता।

(iii) File list ()

वर्तमान में application के द्वारा saved कि सभी file की Array of file देता है।

*3.

External storage:-

ए Android-compatibility compatible device एक shared external storage का समर्थन करता है जिसे उपयोग file को save करने के लिए करते हैं।

External storage में saved कि सभी file world-Readable हैं और user क्या संशोधित कि जा सकती हैं।

* App में गुपती *

* Saving file on external storage:-

file को save करना है जिसे application unInstall होने पर भी नहीं हटाया जाना चाहिए तो इसे external storage पर public directories में से एक में save करें।
तो प्रयोग में file को save कर सकते हैं।

ii) Public file:-

ऐसी file जो अन्य APPS & उपयोगकर्ता के लिए स्वतंत्र रूप से available होनी चाहिए।

iii) Private file:-

files जो APPS में सम्बन्धित हैं और जब users application unInstall करते हैं तो files को हटा दिया जाता है।

* Request external storage permission:-

public external storage में लिखने के लिए हमें अपनी manifest file में WRITE EXTERNAL storage अनुमति का अनुरोध करना होगा।

<manifest - - - >

<uses permission

android:name="android.permission.WRITE_EXTERNAL_STORAGE"

</manifest >

* SQLite Introduction:-

SQLite एक open source database है जो data को एक device पर एक text file में store करता है। Android built in SQLite database implementation के साथ आता है।

* SQLite के Features:-

- (i) Transaction ACID का पालन करता है।
- (ii) Configuration process easy होता है।
- (iii) अभी-2 यह direct file system (IO) पर होता है।
- (iv) यह टैबल, ~~कॉलम~~, आकार के Database और gigabyte के तार और blobs का समर्थन करता है।
- (v) लगभग यह सभी OS Android, BSD, iOS, Linux, Mac, Solaris, vxworks और windows और SQLite को support करता है।
- (vi) इस database एक single-platform disk file में storage किया जाता है।

* नाम गी यरावरी *

* Dis Advantage of SQLite:-

(i) इसका उपयोग वहाँ किया जाता है जहाँ माध्यम से कम traffic requests होता है।

(ii) Data base का size restricted होता है।

SQL Lite तीन प्रकार के Data का support करता है :-

- Text type
- Integer type
- Real type

* CRUD operation IN SQLite Database:-

CRUD का basic operation के लिए एक Abbreviation है जो हम किसी भी database में perform करते हैं।

* CRUD AND Other Database queries:-

(a) The database structure

employee		
ID	INT	PK
NAME	varchar(250)	
Department	varchar(250)	
Joining Date	Date Time	
SALARY	Double	

(b) Creating the table:-

```
CREATE TABLE EMPLOYEES (  
    id INTEGER NOT NULL CONSTRAINT employees_pk  
    primary key AUTOINCREMENT,  
    name varchar(200) NOT NULL;  
    department varchar(200) NOT NULL,  
    joiningdate datetime NOT NULL,  
    salary double NOT NULL.  
);
```

(c) Insert New Record:-

```
INSERT INTO EMPLOYEES  
    (Name, department, joiningdate, salary)  
VALUES  
    ('Mayank', 'Computer', '2021-204-01 11:05:17',  
    '50000');
```

(d) Reading ALL existing Record.

```
select * from employees;
```

+ 44 11 1111 +

(E.) Reading specific record:-

```
select * FROM employees WHERE id = 1;
```

(F.) Updating A Record:-

```
UPDATE employees  
SET  
NAME = 'Naveet'  
department = 'computer'  
salary = '45000'  
WHERE id = 1;
```

(G.) Deleting a Record:-

```
DELETE FROM employees WHERE id = 1;
```

* CURSOR:-

CURSOR वे होते हैं जिनसे Android में database में किए गए query का परिणाम set होता है। CURSOR class में एक API होता है जो किसी Application की column बचने की अनुमति देता है।

* Reading CURSOR Data:-

जब एक बार एक cursor database query लौटाया जाता है तो APP को परिणाम set का अनुसंधान करने और cursor में column data पढ़ने की आवश्यकता होती है।

जब कोई cursor query() method में लौटाया जाता है तो उसकी विधि return की पहली row से पहले की ओर इशारा करता है।

Cursor मे मरामती

CURSOR अपनी internal position मे Manipulate करने के लिए निम्न method प्रदान करती है।

(i) Boolean cursor.move (in offset)

(ii) Boolean cursor.move to first ()

(iii) Boolean cursor.move to last ()

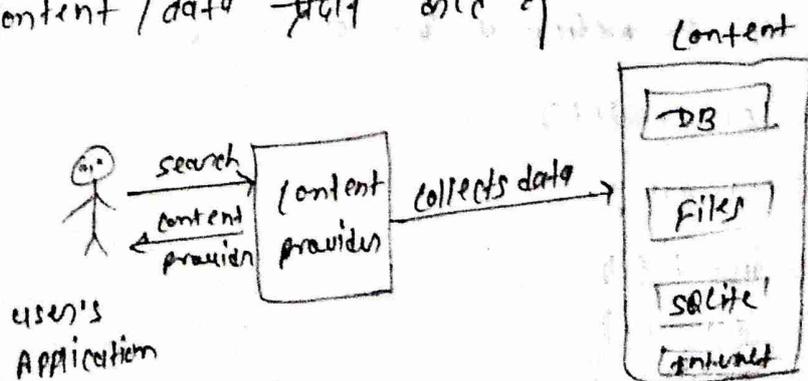
(iv) Boolean cursor.move to next ()

(v) Boolean cursor.move to position (int position)

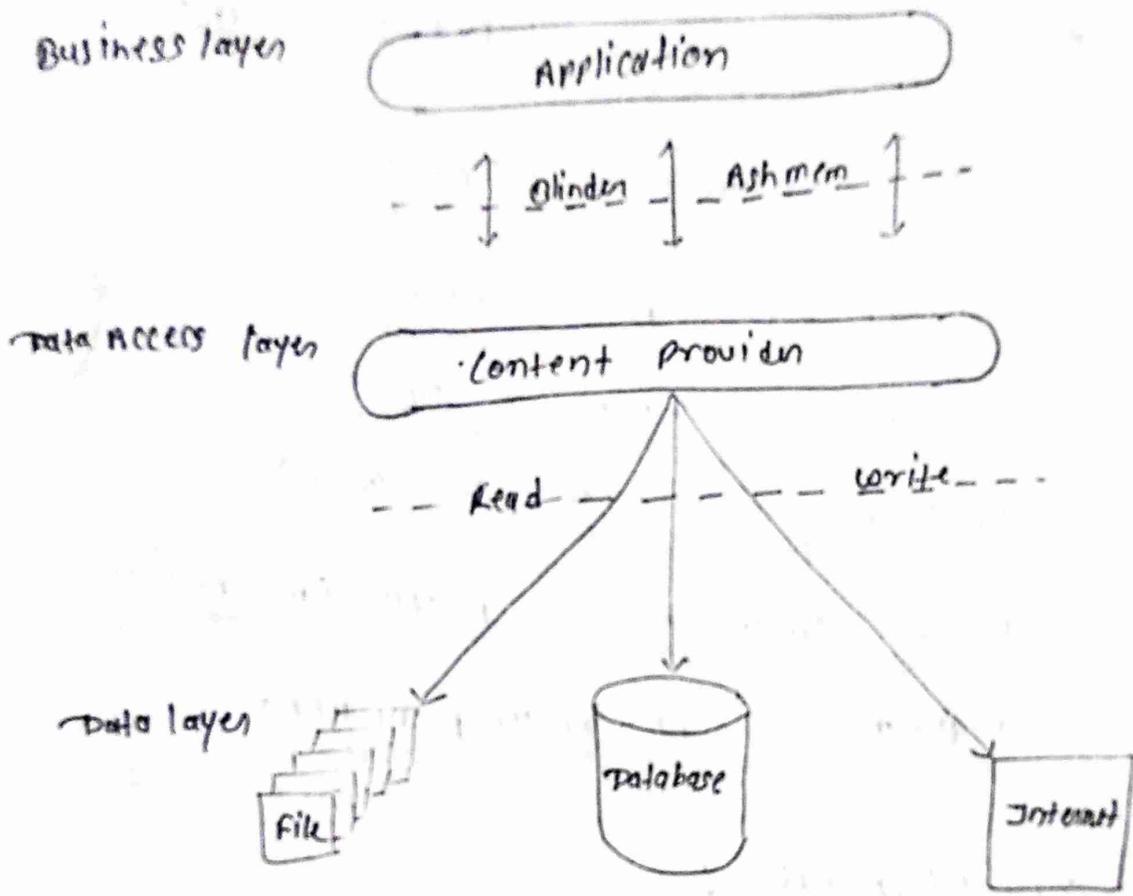
(vi) ~~Boolean~~ cursor.move to previous ()

* Content Provider :-

Content providers Android system या अन्य Android Application में Android Application को content / data प्रदान करते हैं।



* Content provider working:



* Methods of content provider in Android :-

येहाँ उन methods की सूची दी जाती है

- (i) on create ()
- (ii) on query ()
- (iii) insert ()
- (iv) delete ()
- (v) update ()
- (vi) getType ()