MAD

* Android Application components :-

(1) Activity
(2) ~~Intent~~
(3) Service
(4) Contract provider
(5) Broad cast Receiver

* Activity :- They dictate the UI and handle the user interaction to the smart phone screen. front-end (xml) and back-end (Java) is ~~[illegible]~~ Activity.

⟹ Public class mainActivity extends Activity {}

* Services :- They handle background processing associated with an application.
(Perform long-running operation)

⟹ Public class myservice extends Service {}

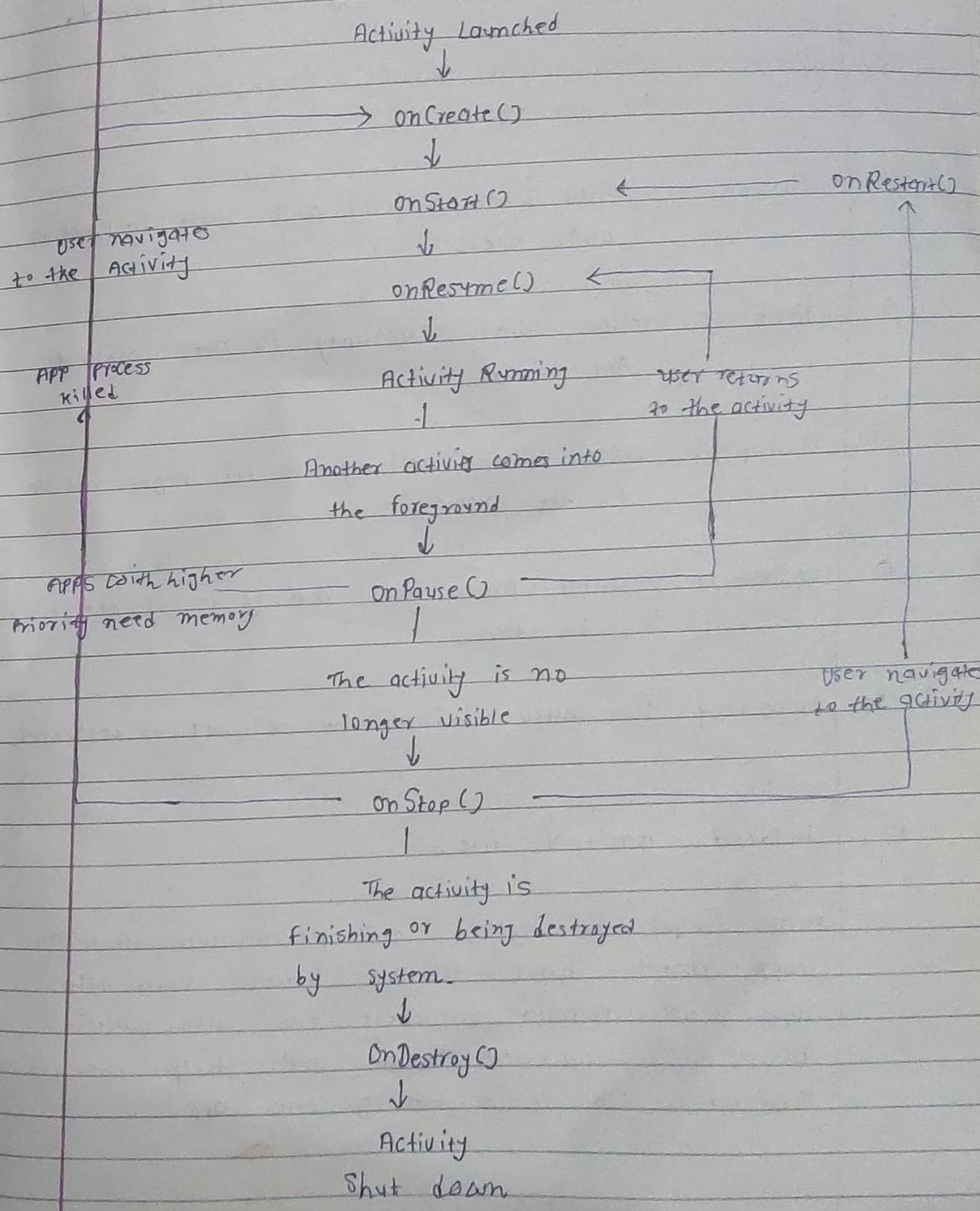
* Broadcast Receivers :- They handle communication between Android OS and application

⟹ Pubic class MyReceiver extends BroadcastReceiver {
Public void onReceive (context, intent) {}
}

* content providers :- They handle data and database management issues.

⟹ Public class MyContentProvider extends ContentProvider {
Public void onCreate() {}
}

* Activity Lifecycle :-



⇒ main components of lifecycle :-

(1) onCreate :- called when activity is first created.

(2) onStart :- called when activity is becoming visible to the user.

(3) onResume :- called when activity will start interacting with user.

(4) onPause :- called when activity is not visible to the user.

(5) onStop :- called when activity is no longer visible to the user.

(6) onRestart :- called after your activity is stopped, prior to start.

(7) onDestroy :- called before the activity is destroyed.

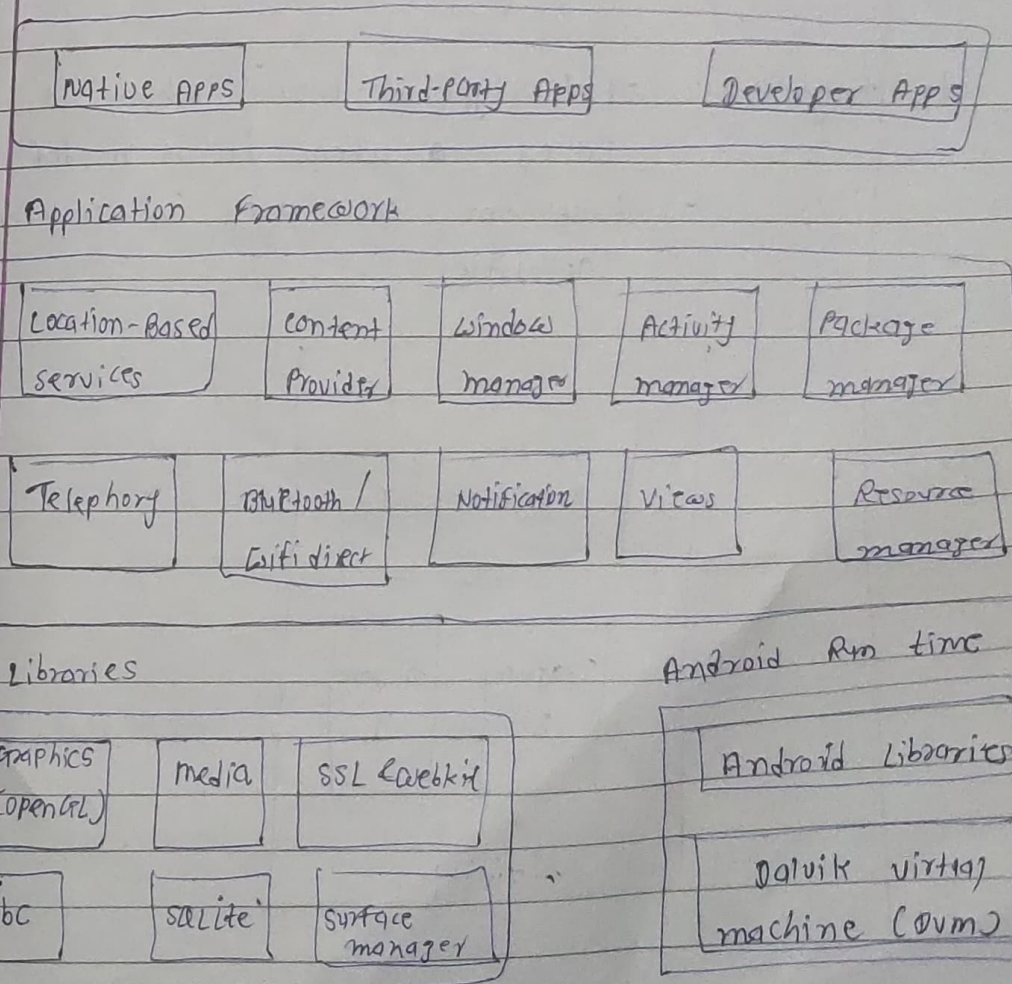
* Android manifest file :-

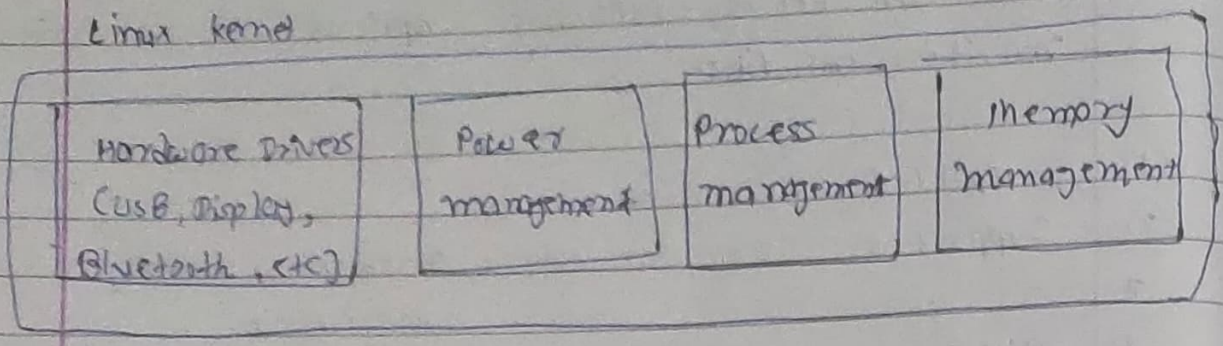
→ AME helps to declare the permissions that an app must have to access data from other apps. The android manifest file also specifies the app's package name that helps the Android SDK while building the app.

→ You can define following options in android manifest.xml file:-

(1) Supported screen size
(2) Supported SDK versions = minimum, target and maximum
(3) Ability to send push notification
(4) various permissions for the application

* Android Architecture:-

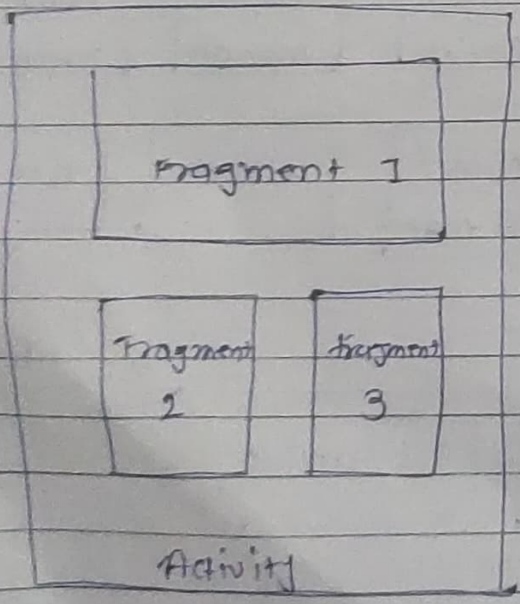




* Intent :-

two type = Implicit
explicit

* Fragment :- It is type of block which is called when it is needed.
- Fragment can use in several activities



* Need :

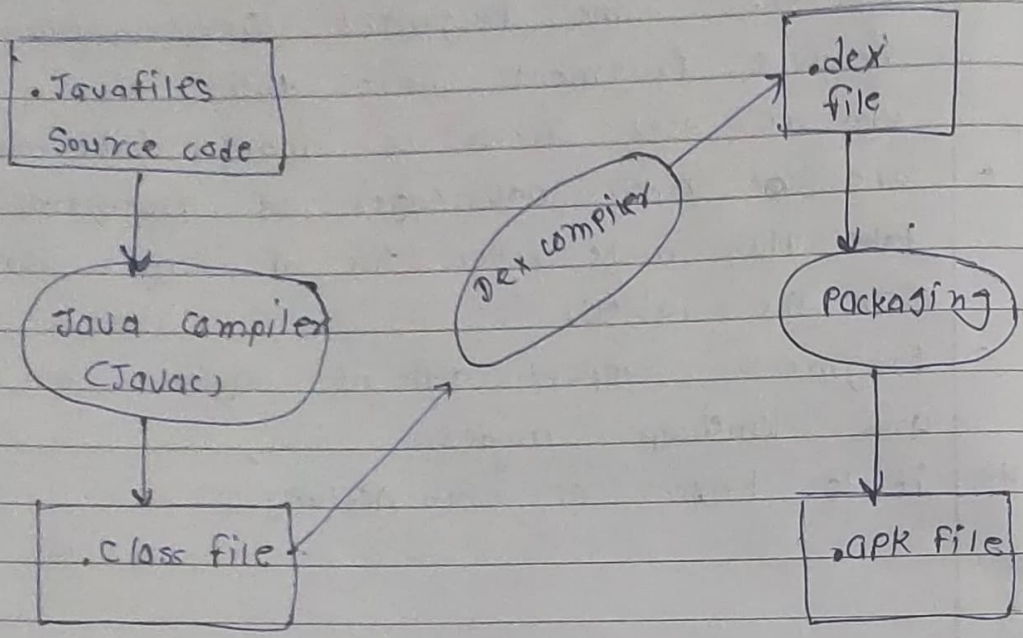
- We can combine several fragment in single Activity.
- Reuse the same fragment across several activities, because fragments have their own layout which can be used across the many different activities.
- one of many advantages of fragments, is that they make better use of larger screen space on the tablets.
- Fragments support different layouts on portrait and landscape modes.

* it is known as "sub-activity."

* DVM (Dalvik virtual machine) :-

→ As we know the modern JVM is high performance and provides excellent memory management.
- But it needs to be optimized for low-powered handheld devices as well.
- DVM is an android virtual machine for memory, battery life and performance.
- Dalvik is a name of town in iceland.
- The Dex compiler converts the class file into the .dex file that run on the DVM.
- Multiple class files are converted into one dex file.
- The Javac tool compiles the java source file into the class file.
- The dx tool takes all the class file of your

- The android Assets packaging Tools handles the packaging process.



* Android Menu :-

⇒ 3 type of menu = 1) Option menu
2) context menu
3) Popup menu

1) option menu :- Android option menus are primary menus of android.
- They can be used for setting, search and delete item etc.

2) context menu :- Android context menu appears when user press long click on the element.

- it is also known as floating menu.

3) Popup menu :- this menu displays the menu below the anchor text if space is available otherwise above the anchor text.

- It disappears if you click outside the popup menu.

* View and view Group :-

⇒ In android apps, the two central classes are Android View class and View Group class.

View →

View is a basic block of UI in android. A view is a small rectangular box that responds to user inputs. Ex Button, Edit Text, checkbox etc.

→ View is a UI feature that we intract with when we use an app, such as a button, editing text and images ....

→ The View class is a base class for all the GUI components in android.

→ View refer to the android.view.View class, which is the base class of all UI classes.

* ViewGroup :-

- View group is an invisible container of other views and View Groups.
  Ex Linear Layout is a View Group that can contain other views in it.
- View Group is a special kind of view that is extended from View as its base class.
- The ViewGroup class is a subclass of view class. And also it will be act as a base class for all layouts and layouts parameter.

Ex Linear Layout is the ViewGroup that contains UI controls like Button, text view.

- ViewGroup Refer to the android.View.ViewGroup class.

* Layouts :- A layout defines that the visual structure for a user interface, such as the UI for an activity or app widget.

⇒ Here, 8 types of layout :-

1) Linear Layout
2) Relative Layout
3) Table Layout
4) Absolute Layout
5) Frame Layout
6) List view
7) Grid view
8) Web view

1) Linear Layout :- This layout is a view group that aligns all children in a single direction, vertically or horizontally.

2) Relative Layout :- It is a view group that displays child views in relative positions.

3) Table Layout :- It is a view that groups view into rows and columns.

4) Absolute :- It is enables you to specify the exact location of its children.

5) Frame :- It is a placeholder on screen that you can use to display a single view.

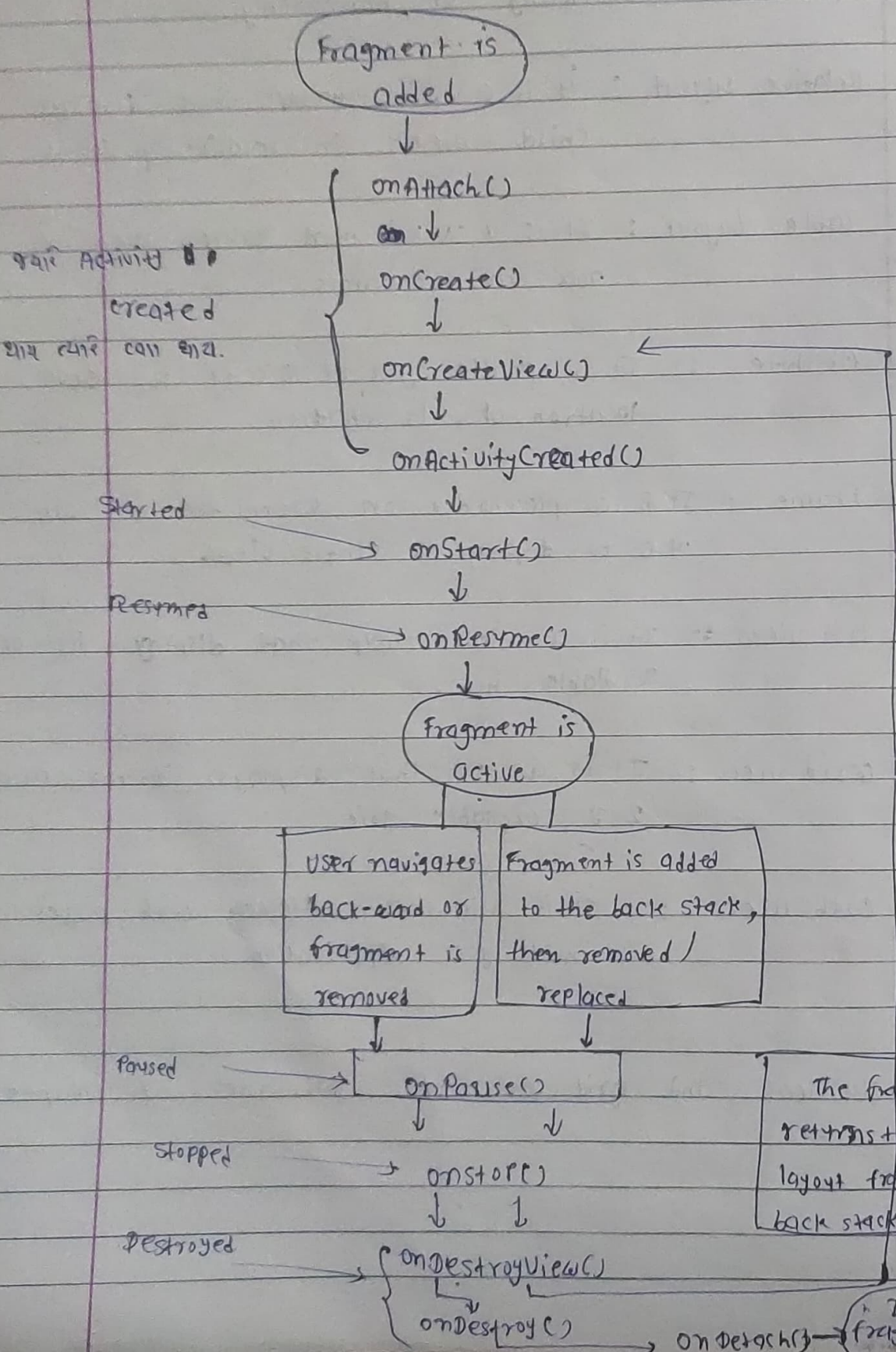
6) List view :- It is a view Group that display list of scrollable item.

7) Grid view :- It is a VG that displays items in a 2-D, scrollable grid.

8) Web view :- It is view that display web pages inside your application.

* List view and grid view are subclasses of Adapter view.

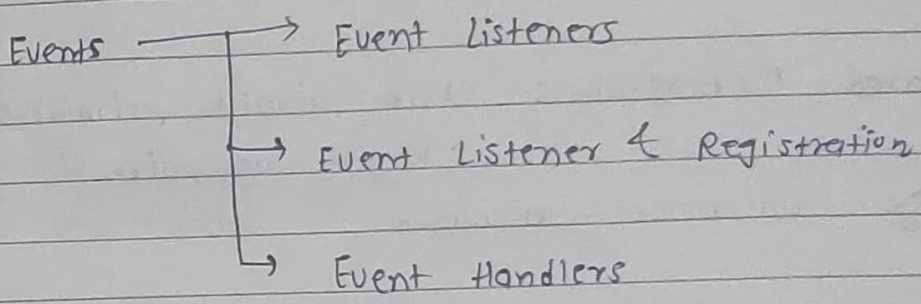
* Fragment Lifecycle :



यहाँ Activity created थाय त्यारे call थाय.

* 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.



* Event Listener & Handler :-

(1) onclick → onClickListener :- This is called when the user either click or touches or focuses upon any widget like button, text, image etc

* Event Listeners registration :-

→ 3 ways :- 1) Using an Anonymous inner class
2) Activity class implements the Listener interface.
3) Using Layout file activity_main.xml to specify event handler directly.

ch-4

* Storage options in android :-

1) Shared preferences
2) Internal Storage
3) External storage
4) SQLite Database
5) Network Connection

1) Shared Preferences :- Store private primitive data in key-value pairs.
   Ex Username - password

2) Internal storage :- Store private data on the device memory.
   Ex phone storage

3) External storage :- Store public data on the shared external storage.
   - Sometimes when internal storage was full then use external storage.
   Ex SD card.

4) SQLite Database :- Store structured data in private database. data which support SQL queries.
   Ex Phone contact list (structured form), true caller

5) Network Storage :- Store data on the web with your own network server.

* Working with content provider :-

- content provider are a very important component that serves the purpose of a relational database to store the data of application.
- The role of content provider in the android system is like a central repository in which data of applications are stored.
- And facilitates other applications to securely access and modifies that data based on user requirements.

⟹ Users can manage to store the application data like images, audio, videos and personal contact information by storing them in SQLite Database in files or even on a network.

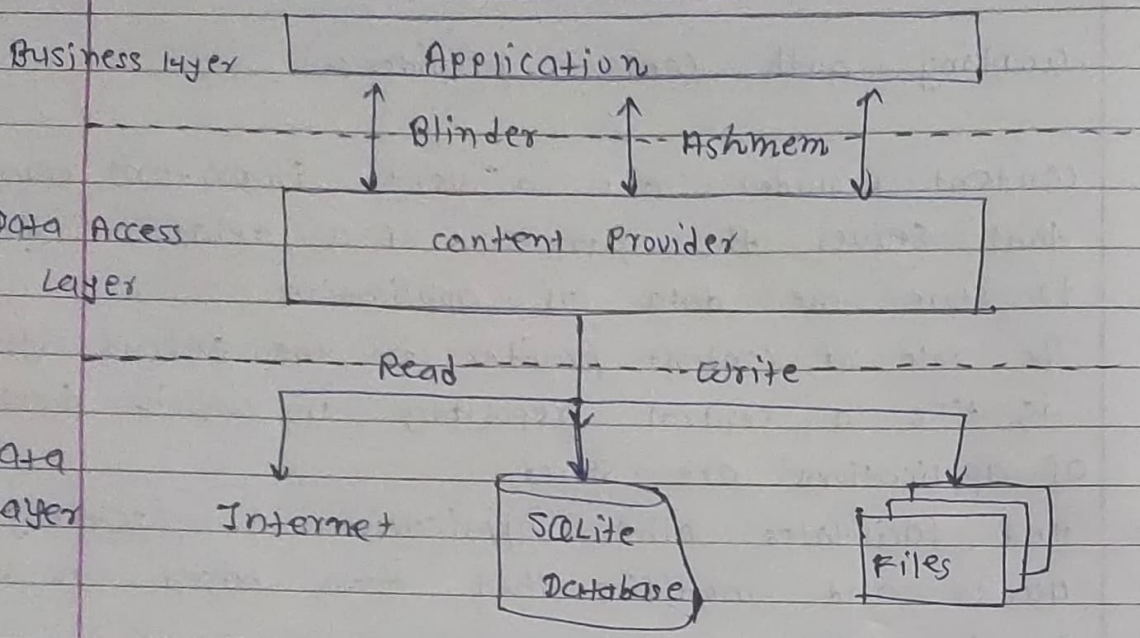
- In order to share data, content providers have certain permissions that are used to grant or restrict the rights to other applications to interfere with the data.

* content URI (Uniform Resource identifier) :-

- It is a key concept of content providers.
- To access the data from a content provider, URI is used as a query string.

→ Content URI Structure :

content URI : content : // authority / optional path / optional ID



⟹ Different parts of content URI :

1) content :// :- mandatory part of the URI as it represents that the given URI is a content URI.

2) authority :- Signifies the name of the content provider like contacts, browser, etc.
- This part must be "unique" for every content provider.

3) Optional Path :- Specifies the type of data provided by the content provider.
- It is essential as this part helps content providers to support different types of data that are not related to each other like audio and video files.

4) OptionalID :- It is a numeric value that is used when there is a need to access a particular record.

* If an ID is mentioned in a URI then it is an "id-based URI" otherwise a "directory-based URI".

* Operations in content provider :-

→ Four fundamental operations are possible in content provider :- Create
Read
Update
Delete.

1) Create :- Operation to create data in a content provider.
2) Read :- Used to fetch data in a content provider.
3) Update :- To modify existing data.
4) Delete :- To remove existing data from the storage.

* UI components of android applications like activity and fragments use an object "CursorLoader" to send query request to "ContentResolver".

- The