

Assignment - 1

e.1 Write a note on difference platform, application of the Android ?

- Android is a popular operating system for mobile devices & it continues to be a leading platform for app development.
- According to research conducted by MarketsandMarkets the market size for mobile app development will be 197.2 billion USD by 2021.

» Native Apps -

- Native apps are built for particular operating systems which are mostly android and iOS.
- There are more OS for mobile applications like BlackBerry & Windows.
- This is available for download on google play store & for IOS Apple App store.
- Native apps are generally built to make the most of all the features & tools of phones such as contacts, cameras, sensors etc.

Advantages -

- Native apps are designed for the particular OS and it gives the best user experience.
- Native apps are built with separate gestures it gives a good experience to users & it is very useful for all user.

Disadvantages

- Native apps are costly in comparison to others because they want separate maintenance.
- Require a separate codebase to add new features.

2) Web apps

- Web applications are built only the run on browsers.
- They are mainly the integrations of HTML, CSS & JS.
- It runs on chrome, firebox & other browsers.
- The responsiveness & functionality of the web apps could easily be confused with native app. both the native app & web apps have almost same features & responsive nature.

Advantages -

- Easy to build.
- Web apps are used less storage than other apps.
- Web apps are preinstalled on all device.
- Web app's are easily accessible in any type of application.

Disadvantages -

- Slower compared to native apps.
- There might be some user interface issues.
- In hybrid apps have limitations in using all the Hu & OS features.

Q.2 Explain history of android os.

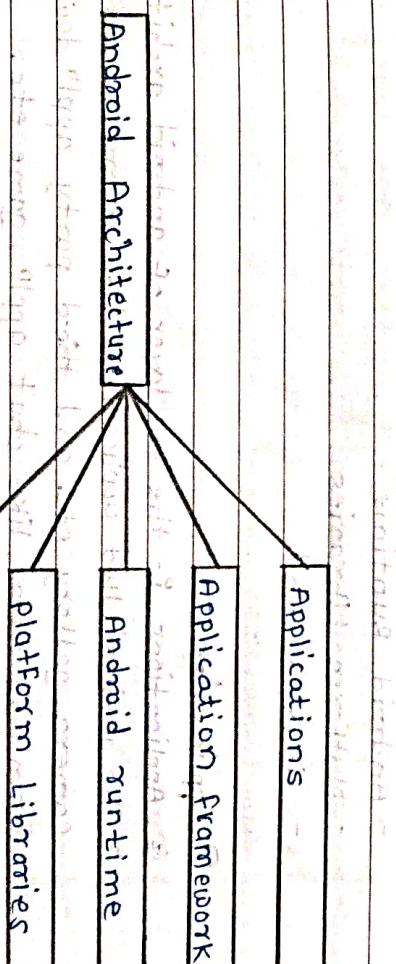
- 3) Hybrid apps -
 - Hybrid app's are also called cross platform app's
 - Hybrid app's are run on multiple platforms like Android & IOS.
 - These are made from the integration of web & native applications.
 - For hybrid applications dependent we use flutter, Dart, React Native etc.

Advantages -

- User can use it on more than one platform.
- It is integrated with browser.
- maintained by many versions.
- Shareable code makes it cheaper than a native app.

- Android 1.0 (2008)
 - First Android device : T-mobile G1.
- Initial features : touchscreen, web browsing, email, & third-party apps.
- Android 1.5 - This version came up in late April 2009 & was the first to have Google desert-themed naming scheme & be incorporated in the Samsung galaxy phone series.
- It was introduced with a lot of functionalities that we take for granted today.
- Android Innovations & Features
 - Google Play store (2009)
 - Multi-touch support (2010)
 - NFC Support (2010)
 - Google assistant (2016)
 - Artificial intelligence (AI) integration (2017)
 - Augmented reality (AR) support (2017)
 - 5G support (2020)
 - State & milestones
 - 1 billion + Android activations (2014)
 - 2 billion monthly active devices (2017)
 - 70% global smartphone market share (2020)
 - Features :
 - * Multi-touch support ; supports gestures & multi-touch interactions.
 - * Notification System : Displays notifications & alerts
 - * Google play store : official app store for Android app.
 - * Security : includes features like encryption, secure boot, & app sandboxing

Q3 Explain android architecture with diagram.



- Advantages :
 - Open-source : Allows customization & community driven development.
 - Accessibility : Supports a wide range of devices & HW configurations.
 - Customization : users can personalize their home screens, icon packs, & launchers.

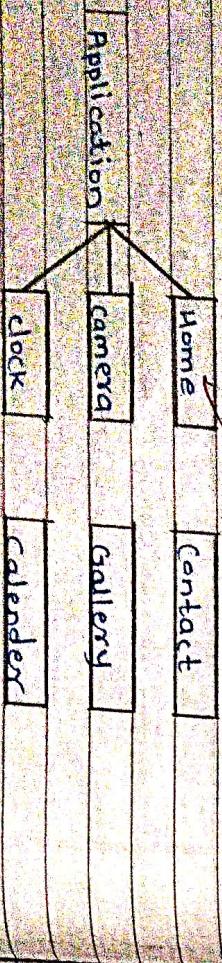
- In android architecture means abstract representation or static representation.
- our android operating system is made up of which components it is known as android architecture.
- In android architecture contain four rows & five different columns.

• Components of Android Architecture.

- Application ~~over installed apps~~
- Application Framework ~~is managed by android API~~
- Android Runtime ~~OS API Core Libraries~~
- Platform Libraries
- Linux Kernel ~~wifi driver~~

① Application -

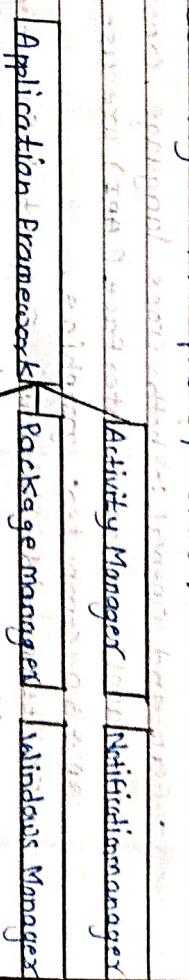
- ② Applications is the top layer of android architecture.
- ③ The preinstalled applications like Home, Contacts, camera, gallery etc. and third party appln downloaded from play store like chat appln, game etc.
- ④ Will be installed on this layer only.
- ⑤ It runs within the android run time with the help of the classes & services provided by the appln framework.



② Application Framework -

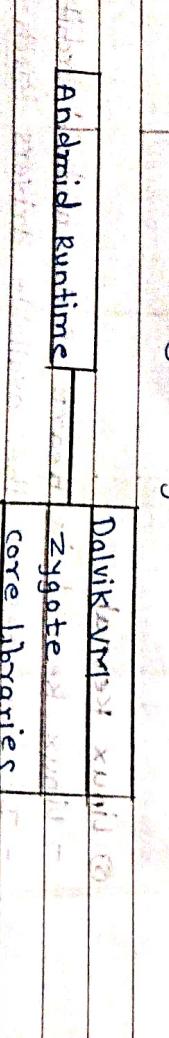
- ① Appln Framework provides several important classes which are used to create an Android appln.
- ② It provides generic abstraction for H/w access.
- ③ It provides the services with the help of which we can create particular class.

- ④ Which are helpful for the development of our application according to the prerequisite.



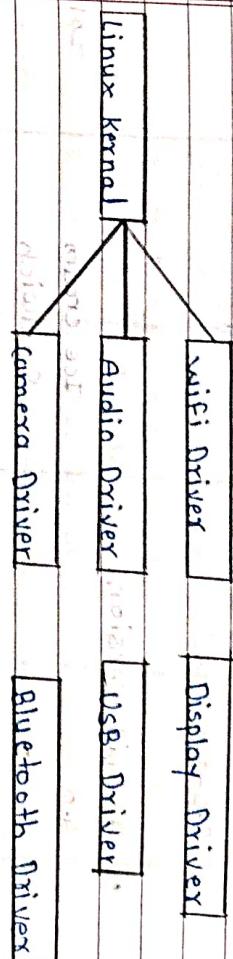
③ Application Runtime -

- ② Application runtime environment is one of the most important part of Android.
- ② It contains Components like core libraries and the Dalvik Virtual Machine (DVM)
- ③ It provides the base for the application framework and powers our appln with the help of core libraries like JVM, DVM.
- ④ It depends on layer Linux kernel for threading & low level memory management.



④ Platform Libraries

- The platform libraries includes various C/C++ core libraries & Java based libraries such as media, graphics, surface manager etc.
- Media :- Library provides support to play & record an audio and video formates.
- Surface manager :- responsible for managing access to the display subsystem.
- SGL and OpenGL :- both cross language, cross platform app's interface (API) are used for 2D & 3D computer graphics.
- SQLite provide database support & free type provides front support.
- Web-Kit :- This open source web browser engine provides all the functionality to display web content and to simplify page loading.
- SSL (Secure Sockets Layer) is security technology to establish an encrypted link between a web server & a web browser.



Q4 Write note on versions of Android OS.

Android Version	Android Version Names	Release Year
1.0 - 1.1	NO codename	2008
1.5	cupcake	2009
1.6	Donut	2009
Android versions	Android version	
2.0 - 2.1	Eclair	2009
Android version	Froyo	2010

⑤ Linux Kernel

- Linux kernel is heart of the android architecture
- It manages all the available drivers such as display, audio drivers, memory drivers, etc.

- Which are required during the runtime
- It responsible for management of memory, power devices etc.

• Android version	Gingerbread	2010	• Android version	Android 10	2019
2.3			10		
• Android versions	Honeycomb	2011	• Android version	Android 11	2020
3.0-3.2			11		
• Android Version			• Android version	Android 12	2021
4.0	Ice cream Sandwich	2011	12		
• Android versions	Jelly Bean	2012	• Android version	Android 13	2022
4.1-4.3			13		
• Android version	Kitkat	2013	• Android version	Android 14	2023
4.4			14		
• Android versions	Lollipop	2014	• Android version	Android 15	2024
5.0-5.1			15		
• Android versions			17	cupcake -	
Marshmallow	2015		- In 2009, Android released its second major release.		
6.0			Android version 1.5 cupcake.		
• Android versions	Nougat	2016	- The tradition of naming Android versions after confectionaries started.		
7.0-7.1			- Android introduced the first on-screen keyboard as people moved to touchscreen.		
• Android versions	Oreo	2017	- They also introduced a framework for third-party app widgets, which was a significant step.		
8.0-8.1			- cupcake also introduced the platform first step ever video recording option.		
• Android version	Pie	2018			
9					

2)

Donut -

- Android dropped its next major update, Android 1.6 Donut, come with built-in support for CDMA networks.
- It introduced multiple new features like voice & text entry search, book mark history.
- It allowed the users to select multiple photos for deletion at a time.

3)

Eclair -

- Eclair was released just six weeks after Donut.
- It introduced SMS, MMS, voice-guide turn-by-turn navigation & real-time traffic information.

4)

Froyo -

- Released four months after Eclair's introduction.
- This version of android focused on back-end performance, speed & memory optimization.

5)

KITKAT

- This Android version introduced "OK Google" support, offline music support, smart caller ID, better app compatibility and many other built-in features.

6)

Gingerbread -

- Android started portraying distinctive visual design.
- Gingerbread supported an extra-large screen, a simplified interface, enhanced copy/paste functions.

7)

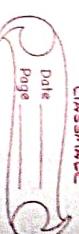
Honeycomb -

- Android released the Honeycomb version in 2011 for the first Android-based tablet.
- With subsequent updates 3.1 & 3.2, the Honeycomb remained a tablet-exclusive entity.

8)

Ice-cream sandwich -

- Ice-cream Sandwich marked the entry of Android into the modern design language.
- It also brought a framework or design standardization known as 'Holo' across all Androids app ecosystem.



9)

Lollipop

- It also introduced at a glance access for all the notifications from the lock screen itself.
- With the version Lollipop, Android reinvented itself.

10)

Marshmallow

- Android version was slightly updated when compare to the Lollipop version.
- With marshmallow, Android started the trend of releasing a major update per year.

(2) Nugat

- Android is popular for releasing Google Assistant
- This Android version offered improvements but all of them were significant.

(3) Oreo

- Android version included many elements aligned with Google's goal of aligning Android and Chrome OS and transforming the chrome book user experience.

(4) Pie

- also introduced many security & privacy enhancements & intelligent systems to manage power & screen brightness

(5) Android Version 10

- With this version, Android dropped the naming tradition of each major release along with an item of confectionery.
- this version introduced even more visual interface upgrades for android gestures & a swipe-driven navigation approach

(6) Android Version 11

- It is Android 11OS is the eleventh big release of Android.

- It is 11th version of Android mobile OS, which was released on 2 September 2020.

- The alphabetic naming system of Android based on deserts, was stopped since Android 10.

(7) Android 12 (Snoozie Cone)

- the most significant progression in these versions was the interface.

- It had an updated design standard called material you, implying that you as the user could customize the device appearance with dynamically generated themes adaptable to the wallpaper colors.

(8) Android 13 (Tiramisu)

- Android 13 took privacy & security to the next level.
- Every app needed to request permission from user before they could send notification.

(9) Android 14 (Upside Down Cake)

- by adding grammatical Inference API, so they could set the apps according to their preferred grammatical gender.

(10) Android 15 (Vanilla Ice cream)

- still released the first developer preview in February 2024, Google expects to share the final update in add dynamic performance, health connect & improved camera controls.

Q.5

Explain activity life cycle with suitable example.

Q.5

To navigate transitions b/w stages of the activity lifecycle, the activity class provides a core set of six callback:

onCreate(), onStart(), onResume(), onPause(),

onStop(), & onDestroy().

D) onCreate()

- You must implement this callback, which fires when the system first creates the activity.

- on activity creation, the activity enters the created state.

- In create method perform basic appn startup logic that happens once for the entire life of activity.

E) onStart()

- When the activity enters the started state, the system invokes onStart().

- This call makes the activity visible to the user as the app prepares for the activity to enter the foreground & become interactive.

- ex: this method is where the code that maintains the UI is initialized.

- When the activity moves to the started state any lifecycle-aware component tied to the activity lifecycle receives the ON-start event.

3) onResume()

- When the Activity enters the Resumed state, it comes to the foreground, & the system invokes the onResume() callback.

- This is the state in which the app interacts with the user.

- When the activity moves to the Resumed state, any lifecycle-aware component tied to the activity lifecycle receives the ON-RESUME event.

4) onPause()

- The system calls this method as the first indicates that the user is leaving your Activity, through

it does not always mean the activity is being destroyed.

- It indicates that the activity is no longer in the foreground, but it is still visible if the user is in multiwindow mode.

5) onStop()

- When your Activity is no longer visible to the user, it enters the stopped state & the system invokes the onStop() callback.

- This can occurs when a newly launched activity covers the entire screen.

- The system also calls onStop() when the activity finishes running & is about to be terminated.

⑥ OnDestroy()

- OnDestroy() is called before the activity is destroyed
- The system invokes this callback for one of two

Reasons:

- 1) The activity is finishing, due to the user completing dismissing the activity or due to finish() being called on the activity.
- 2) The system is temporarily destroying the activity due to configuration change such as device rotation or entering multi-window mode.

Q.6 Write note on back stack activity with example.

⑦ The backstack in android refers to the stack based navigation system used to manage the history of activity the user has visited within an app.

The key characteristics of the back stack in Android are:-

⑧ LIFO data structure:-

- The back stack operates like a stack, where the last activity opened is the first one to be removed when the user navigates back.

⑨ Automatic management :-

- The android system automatically manage the back stack for you as the user navigates through your app's activities.

⑩ Maintaining state:-

The back stack preserves the state of each activity allowing the user to seamlessly return to a previous activity & pick up where they left off.

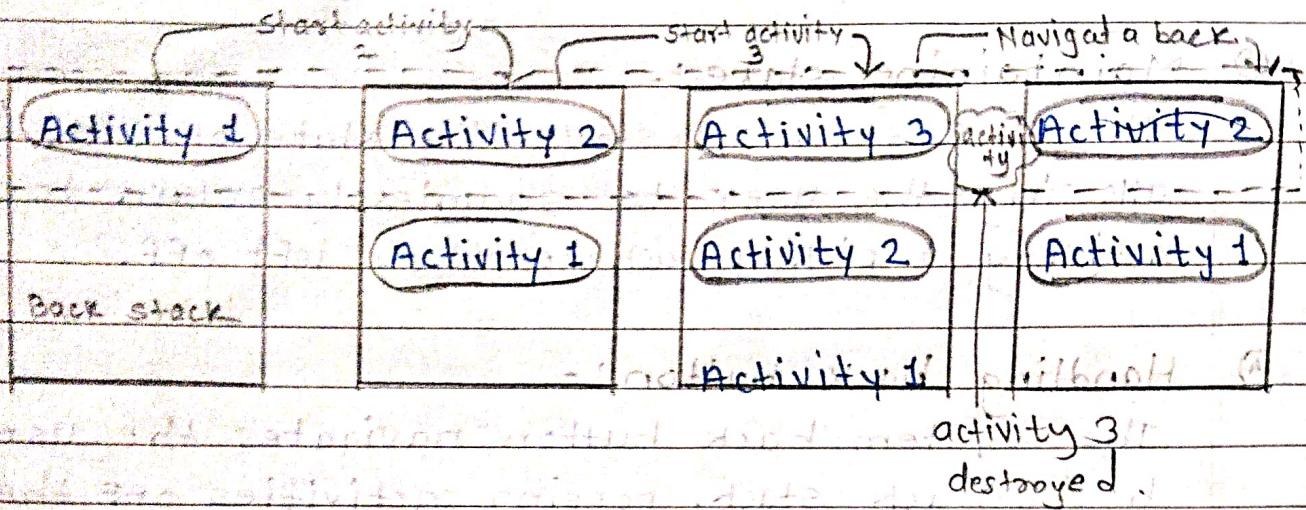
⑪ Handling back button:-

The system back button navigates the user through the back stack, popping activities off the stack until the user reaches the root activity or exits the app.

⑫ Custom back stack behaviour :-

Developers can customize the back stack behavior such as clearing the stack, adding activities to the top of the stack or removing specific activities from the stack.

- * - We start with activity 1 at app launch. The our app starts Activity 2 which brings it to the top of the stack.
- Then we start activity 3 such which replaces Activity 2 at the top.
- Then we press back or Activity 3 calls finish(). Activity 3 is destroyed & the next activity in the stack, Activity 2, is brought back to the top.



- This back stack activity behaviour can be very useful when building natural navigation through your app.
- Create natural navigation using the activity back stack is very important.
- When launching a new activity on android, the previously visible activity is moved onto what's known as the activity back stack.

ex. - an email app might have one activity to show a list of new messages.

~~Review~~