MAD

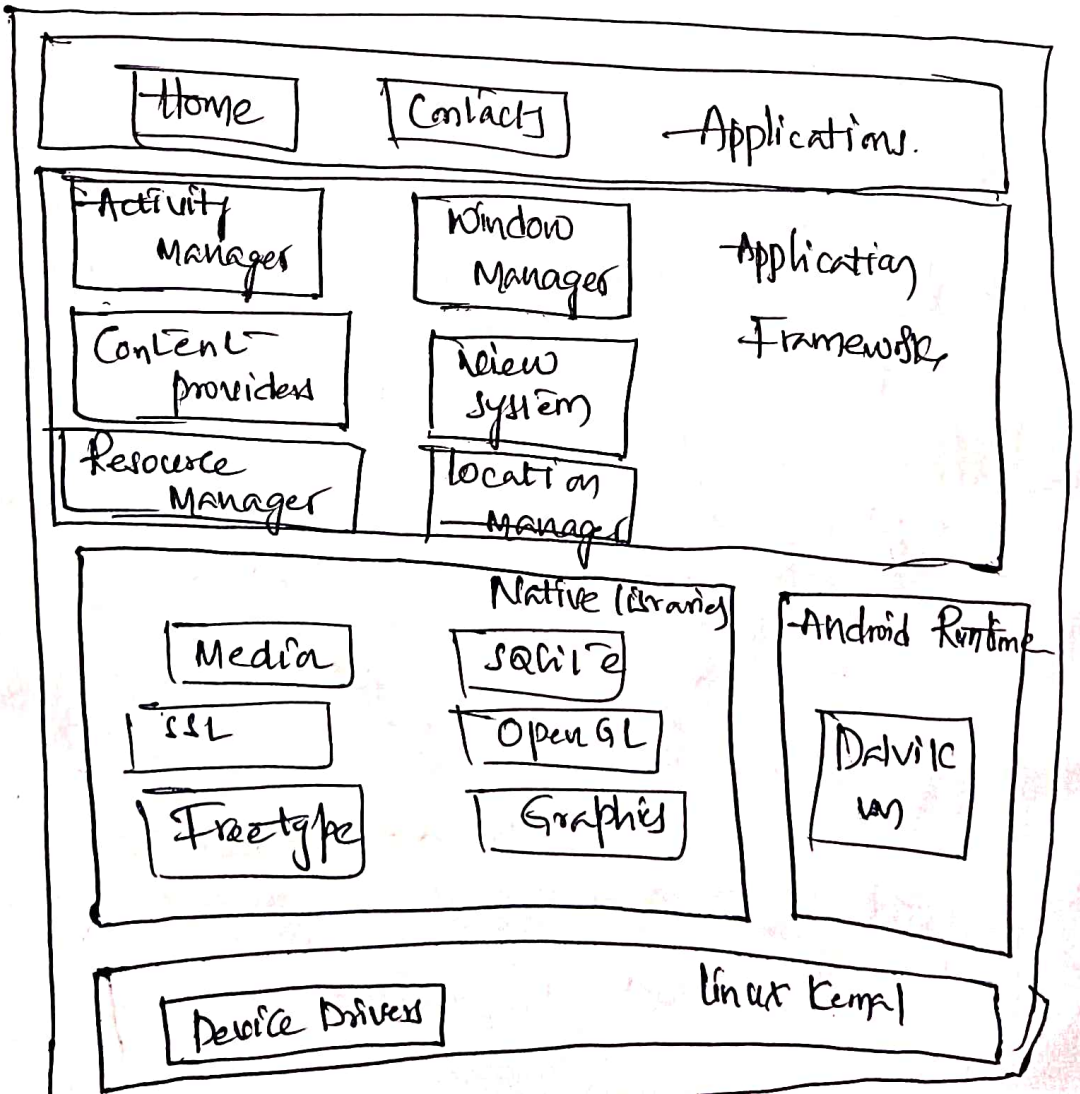
SK Gandhani

1. Explain the basic architecture of Android application stack.

A) Android architecture / Android software stack is categorized into five parts

1. Linux Kernel
2. Native libraries (middleware).
3. Android runtime
4. Application framework.
5. Applications.

Let's see the Android architecture first



1) Linux Kernal :-

⇒ It is heart of android architecture that exists at the root of android architecture

⇒ linux Kernal is responsible for
- device drivers,
- power management,
- Memory management,
- device management
- Resource access

2) Native Libraries :-

⇒ on the top of linux Kernal, their are Native libraries such as webkit, opengl, Freetype, sqlite, Media, C runy library etc.,

⇒ The webkit is responsible for Browser support

⇒ sqlite is for database

⇒ FreeType for font support

⇒ Media for playing and recording audio and video formats.

3) Android Runtime :-

In android runtime, there are core Libraries and Dalvik Virtual Machine which is responsible to run android application

⇒ DVM is like JVM but it is optimized for Mobile devices.

⇒ It consumes less memory and provides fast performance.

Android Framework :-

⇒ Android Framework includes Android API's such as UI (user interface), telephony, resources, locations, Content providers (data) and package managers.

⇒ It provides a lot of classes and interfaces for android application development.

Applications :-

⇒ All applications such as home, contact, settings, games, browsers are using android framework that uses android runtime and libraries.

⇒ Android runtime and native libraries are using linux kernal.

2. Enumerate the basic steps to create a simple Android project with a example?

A) Steps :- open Android Studio.

2. In the welcome to Android studio dialog, click start a new Android studio project.

3. Select Basic Activity (not the default). Click next.

4. Give your application a name such as My First App.

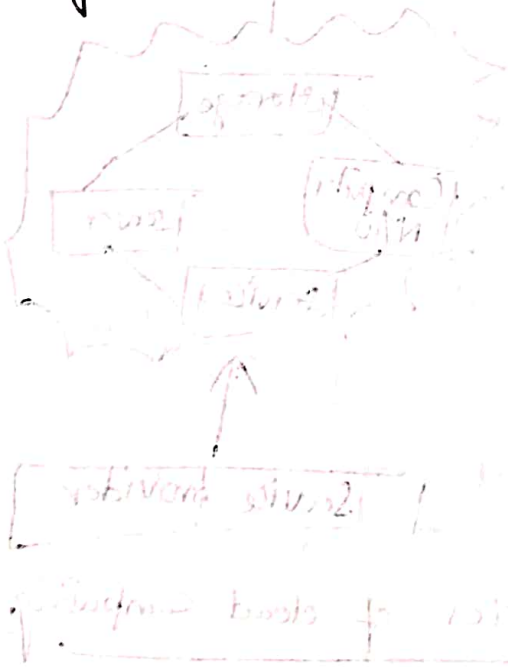
5. Make sure the language is set to Java.

6. leave the defaults for other fields.

7. click Finish

After these steps, Android studio:

- creates a folder for your Android studio project called My first App.



Mid-1

MAD

K. Sandhani

2) What are the essential components in creating Android virtual device?

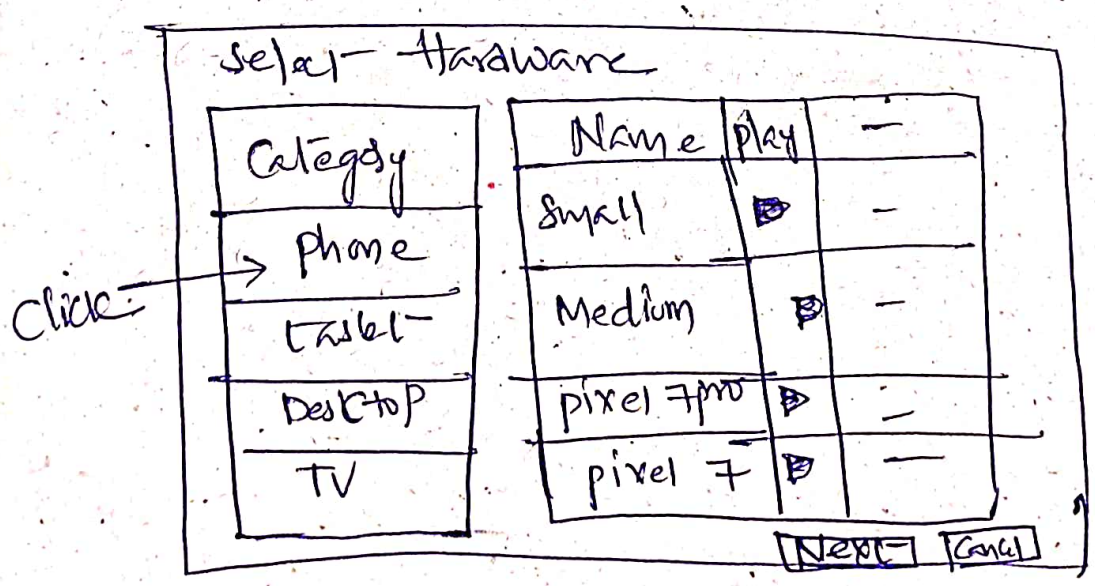
Sol:- Step1:- open Android studio

Step2:- Click on Device Manager in right-side

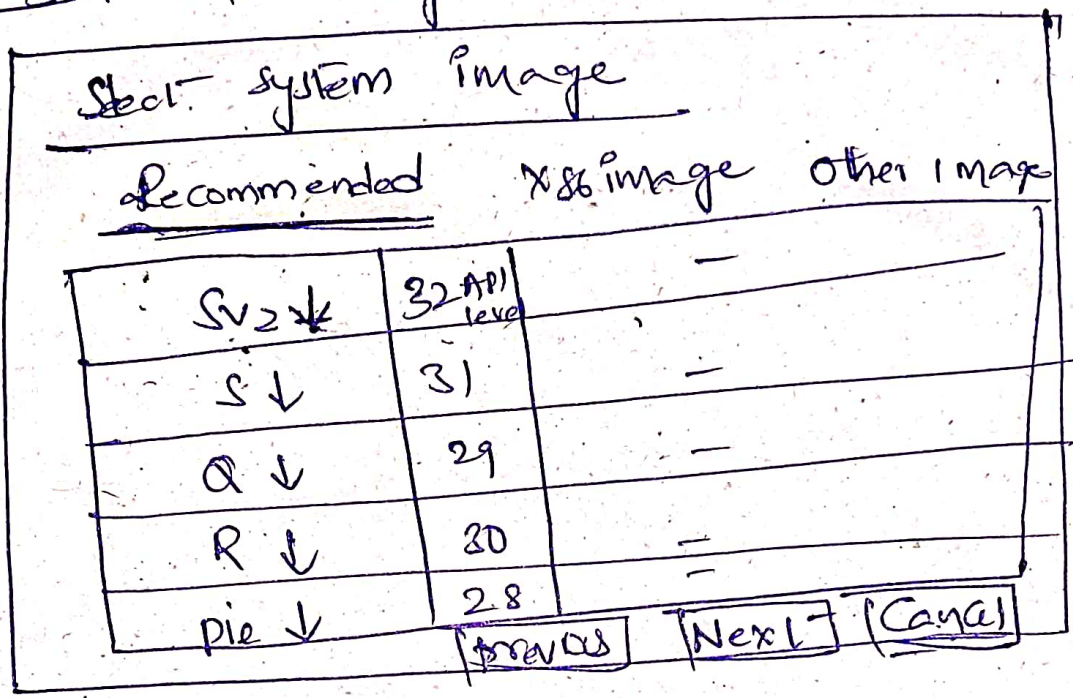
Step3:- There are 3 icons displayed

1. Group - None (✓)
   - ~~form factor~~
   - status
   - type
2. create virtual device (+)
3. ~~Fyjf~~ pair using wifi

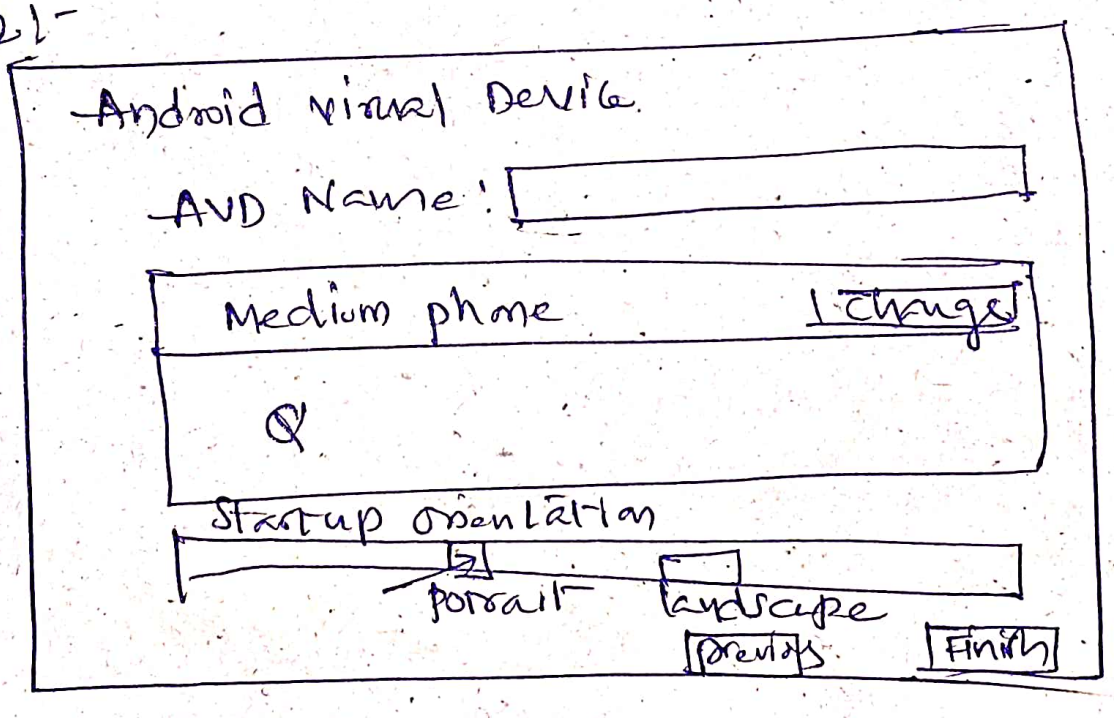
Step4:- Click on (+)



Step 5 :- Select system image



Step 6 :-



4) What is Android debug Bridge? Explain in brief?

Sol:- Android debug Bridge (ADB) is a Client-Server program that is part of Android SDK

⇒ It is used to communicate with device

⇒ The Client-Server program includes three steps

1. Client :- Who sends Commands
2. a daemon : (adbd) :- which runs Commands on a device
3. A server :- which Manages Communication between Client and daemon

⇒ ADB is included in Android SDK platform Tool package.

⇒ Download the package with SDK Manager, install it at

android-sdk/platform-tools/.

Working :-

⇒ When you start adb client, they client first checks whether there is any adb server process is running.

⇒ if process is not in running state, they client start the server process

⇒ When the server starts, it binds with local TCP port 5037 and listen for commands from adb client.

⇒ The server they setup connections to all running devices.

⇒ It locates emulators by scanning odd-numbered ports in the range 5555 to 5585, which is the range used by the first 16 emulators.

⇒ Each emulator uses a pair of sequential ports — an even-numbered port for console connections and odd for adb connections.

5. How Can we define action code through Java. Explain with example program?

Sol :- In Java, an action code typically refers to the code that is executed in response to an event.

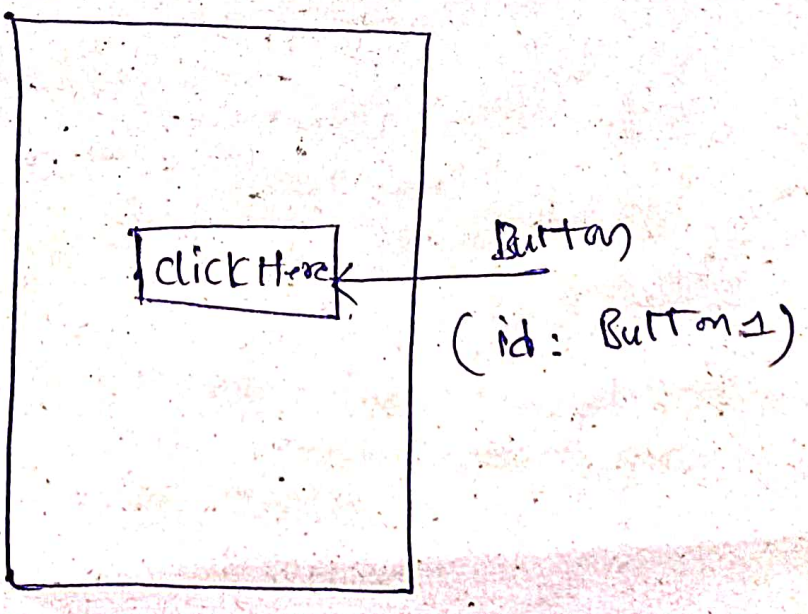
Eg :- clicking a button,
Menu selection

⟹ In android studio development, action code is commonly associated with event handling

Eg :- code :- (Button click action)

Activity - main.xml :-

Design :-



Ex:- 5554 → Emulator 1 (Console)

5555 → adb (Emulator 1)

5556 → Emulator 2 (Console)

5557 → adb (Emulator 2)

adb – wireless debugging :-

Step 1:- open android studio

Step → 2:- click on Device Manager
↳ click on pair using wifi
↳ open settings in phone
↳ click on wireless debugging
↳ allow wifi same in laptop
↳ ~~Ray~~ successfully paired

Step-3:- Ray we get output on device.

2nd unit — Sk. Sandhani

1. Explain the usage of Android Manifest file and activities?

Sol:- Manifest.xml :-

⇒ This file is a crucial configuration file for an Android application.

⇒ It provides essential information about the app to the Android system, such as

- App package name,
- permissions it requires,
- activities it contains
- services it uses.

→ Here are some key elements of the Android Manifest.xml file

- <manifest> : The root element contains all other elements
- <uses-permission> : permissions like internet, Camera, location

Java Code :-

```
package com.example.buttonaction;

import android.x.*;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

public class MainActivity extends
        AppCompatActivity {

    @Override
    protected void onCreate(Bundle
            savedInstanceState)
    {
        super.onCreate(savedInstanceSta
        setContentView(R.layout.activity_m
        Button myButton = findViewById
                (R.id.Button1)
        myButton.setOnClickListener(new
                View.OnClickListener(){
            @Override
            public void onClick(View v){
        Toast.makeText(MainActivity.this,
                "Button clicked").show()
    });}}
```

- <application> : It contains metadata like activities, services, etc.,
- <activity> : Describes an activity with in application like name, icon, launch mode etc.,
- <service> : It perform background tasks without a user interface

Activities :-

⇒ Each activity is implemented as a subclass of 'Activity' class.

- Launching Activities :- Activities can be launched using intents.
- Activity lifecycle :-
  - on create()
  - on start()
  - on Resume()
  - on pause()
  - on stop()
  - on Destroy() etc.,

layout and UI :-

- Buttons,
- Text-fields,
- Images etc;

Inter-Activity Communication :-

Activities can communicate with each other using intents.

Activity stack :- new activities arrive it pushes into stack. if activity is finish it popped from the stack

2) How can a user-interface be created efficiently in an android application explain?

Sol:- Step1 :- plan your UI Design

Step2 :- Use xml layout

Step3 :- Choose correct layouts

Step4 :- optimize the performance (UI must be lightweight)

Step-5 :- Reuse layout components

Step-6 :- Use RecyclerView for Dynamic lists.

Step-7 :- Optimize Images

Step-8 :- Handle screen sizes and orientations

Step-9 :- Test across Devices

Step-10 :- follow Accessibility Guidelines

3) Discuss edit options & checkbox options in creating an android application

Sol :- Edit options :- (Edit Text)

- purpose :- Edit Text is used to capture user input, such as text, numbers, passwords etc,
- Attributes :-
  - input type (text, number, number)
  - Hint text (placeholder),
  - text size
  - text color

Usage :- EditText is Commonly used for forms, search fields, chat interfaces, note-taking apps, and anywhere user input is required.

Events :-

TextChanges ('TextWatcher')

focus changes ('on focus change Listener')

Keyboard action ('on Editor Action Listener').

Validation :-

input Type = "numeric" for numeric input

Checkbox options :-

Purpose :- select one (or) more options from the list of choices.

Attributes :-

- text label
- checked state
- text color
- Background color

• Usage :- Checkboxes are Commonly used in forms, settings screens etc,

• Events :-

- click event ('on click listener')

• selection modes :-

- Single selection mode
- Multi selection mode

• Data Handling :-

— Maintain the state of each checkbox in a data structure

Eg :- (list, array).