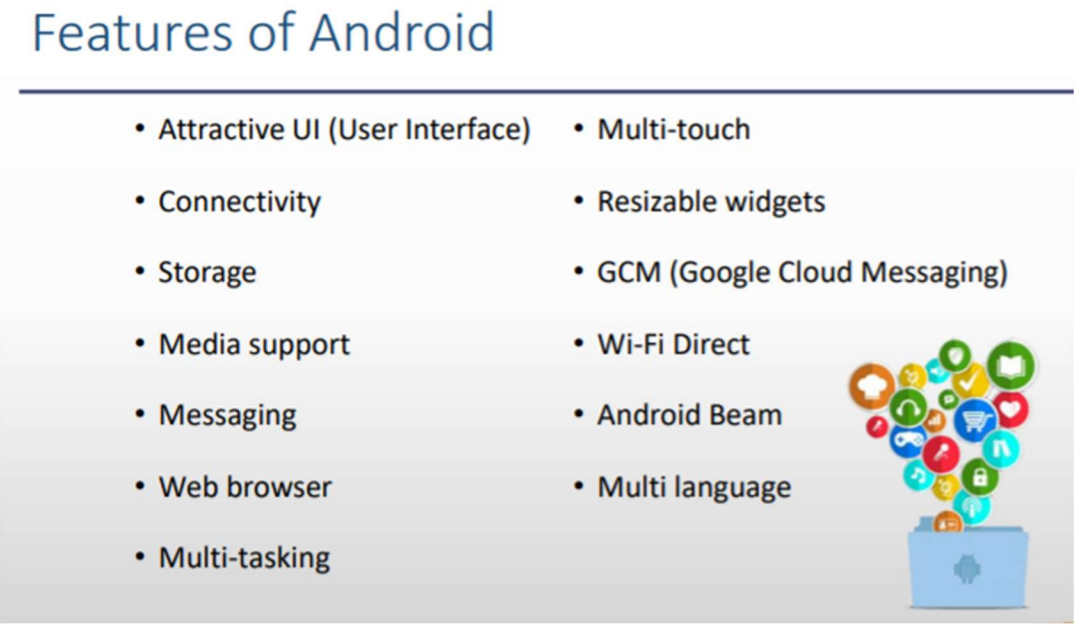
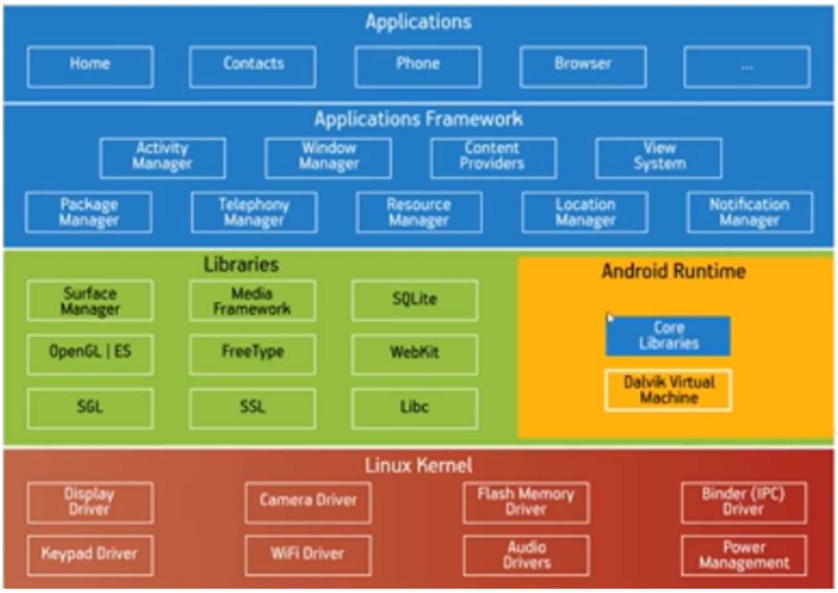
Lecture 4 - Introduction to Android Operating System

- Android platform consists of several major components in order to provide some facilities



Android platform architecture



- First one is linux kernel then libraries after application frameworks then applications finally android runtime

Linux kernel

- Kernel is the one who can operate anything
- In summary kernel is one who can take the internal activities of our application with device hardware components
- Drivers are the models which can engage with some hardware components
- Therefore, all hardware components are driven by those drivers
- Those drivers can be divided into three types
    - Device drivers
        - Example: camera
        - Manage devices
    - Memory management
        - Manage entire memory
        - Allocating and deallocating memory for file systems, processes and applications
    - Process management
        - All the processes of the device are management by this
        - Wifi drivers, audio driver, Bluetooth
- Reasons to use linux for this
    - Portability
        - Easily modifiable according to the requirements
    - Security
        - Linux is well secure
    - Features
        - Network features, security features, process features are there

Libraries

- In order to do the actions by linux kernel there should be some instructions/roots. Those roots/instructions are given by libraries
    - Webkit: fast web entering engine
    - Media framework: support for media files like images, audio, video
    - SQLite: library for database query
    - OpenGL: 3D graphic libraries
    - SGL: 2D graphic libraries
    - Libc: C library
    - Surface Manager: Access to display sub system

Application Framework

- This manage the applications in single way
- All the managers work together to get our work done
    - Activity manager
        - Activity in an android system is an interface what user can see at a given time
        - For one application there can be several activities (several number of user interfaces)
        - Those activities are managed by this
    - Content provider

        - It can access show data among other applications
        - We can directly connect with other applications
        - This is like a platform that connect with applications to share data

    - Resource manager
        - It provides access to embedded files like string.xml, style.xml (resource files)
    - Notification manager
        - Manage notifications and alerts
    - View system
        - Manage Display subsystems
    - Telephone manager
        - Manage xml system or phone system

Application

- Icons that we can see in mobile phone is application layer
- All the 3rd party apps that installed by the user can be seen in mobile phone layer

Android Runtime

- This enable us to use develop android application using java programming language
- This manage the battery life and processing power of the system because of the android runtime layer
    - Core libraries
        - Manage battery life and processing power
    - Dalvik Virtual Machine
        - Store application using less memory
        - At the end of compilations usual applications compile into ".jar" files
        - In android applications this compile two times
        - First time it converts to java bogie code and after that it again compile with Dalvik virtual machine after that it makes ". des" files
        - Those files can produce memory to store the application

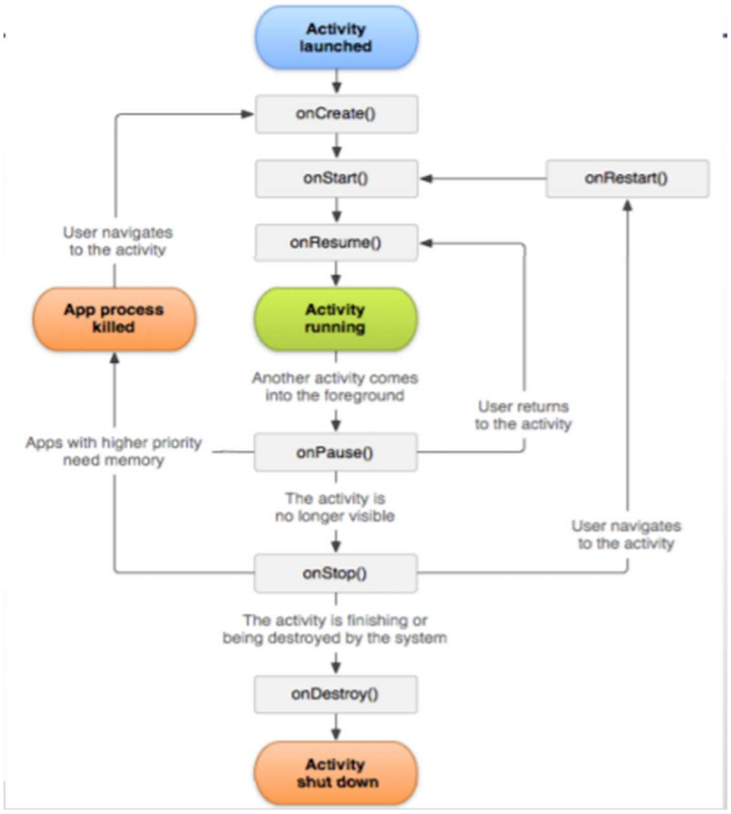
Android Application project structure

- Manifest: app/manifests/manifest.xml
    - This contain essential information about our project
    - Version, java package, what are the activities we going to use
    - In order to make an activity as launching activity inside activity tag copy paste the "<intent-filter>" tag
- MainActivity: app/java/com.example.xxxxxxx/Mainactivity
- app/java/com.example.xxxxxxx/instrumentaltesting
    - this is used for test cases for instrumental test cases
- app/java/com.example.xxxxxxx/exampleTest
    - unit test is done
- app/res/drawable/

    - this can be used to add pictures, drawable xml files

- Layout xml file: res/layout/activity_main.xml
    - All the layouts
    - Particular java activity files are in java folder
- res/mipnap/
    - for images
- res/values/
    - Color, string, style xml

Activity life cycle



- In one application there can be several activities
- We always can navigate from different to different interfaces
- Here this describe in which phase we are going to implement which part

How to create new activity

- Right click on java
- Then new >> Activity >> Empty Activity
- When activity java file created it also create new xml layout file also
-

How to change the text value when button pressed?

- Create a button in main activity and name an id in it
- Create a text field in main activity and give an id to id
- In order to use the UI elements, we have to use the corresponding java file
- *now we are going to use these text field and button*
- *create button type object and text type object*

```
"Button btn;"
"TextView text;"
```

- Button, TextView are classes that already implemented methods
- Now we have to inform the activity java file that there are UI elements in xml file

```
"btn = findViewById (R.id.button);"
"text = findViewById(R.id.textID);"
```

  - Here the "button" means the id we gave in xml file for button
  - "findViewById ()" means method that call for get the id
  - R means the resource file that contain ids of UI elements

- Now call the "onResume ()" method
  - Inside that we have to call the method "`btn. setOnClickListener" through the id`
  - Add the following code to this

```
@Override
protected void onResume() {
    super.onResume();
    btn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            text.setText("I am salitha");
        }
    });
}
```

- If we want navigate to 2nd activity, we created
  - We have to create an intent type object

```
protected void onResume() {
    super.onResume();
    btn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            //text.setText("I am salitha");
            Intent intent  = new Intent( packageContext: MainActivity.this ,
                    activity2.class);
            startActivity(intent);
        }
    });
}
```

- Here the first object we pass in Intent method is the java activity we are in and 2nd is the activity we want to go