

## Functional & Non Functional Req

Functional Req for MA describes what specifically needs to be implemented in a particular system or product and what actions users have to take to interact with the software. They determine what the system does.

Non Functional Req for MA show ~~what~~ what properties & features a particular solution has, how the system will work & why.

### Functional Requirements Include

#### ① Business Requirements

These define the high level goals set by the customer's company that is ordering the soft dev and stipulate what the system is to solve in terms of their business.

Ex → An appl allowing customers to browse through company's product catalog & to purchase products

#### ② User Requirements

These describe the system goals/objectives the users may reach when using the created system. Simply put this is what a user can do : sign up, view certain content & other functions

### ③ Functional Requirements.

These define a list of actions the system has to perform. In addition they have to specify how the system responds to various input data, how it behaves in particular situations.

### Functional Req

### • Non Functional Requirements.

\* These primarily include various quality attributes determining system quality features, most often as examples below.

product

① Availability: Requirements for app continuously running, for ex 24/7, min idle time

② Reliability: App behavior in case of alarm status. For ex automatic restart.

③ Scalability: Ways to expand the system & avoid affected performance.

④ Performance: How many simultaneous users or transactions the system is to service & its response time.

⑤ Security: App operation, fire of safety req related to access control, private data processing, external attack risk reduction.

Usability: Ease of use & user friendly interface that allow users to interact with the product.

Extensibility: Requirements for app extensibility in case there is a need to add new functional requirements.

Funct Req

Non Funct Req

What the system  
should do

How the system  
should do

Product Features

Product Properties

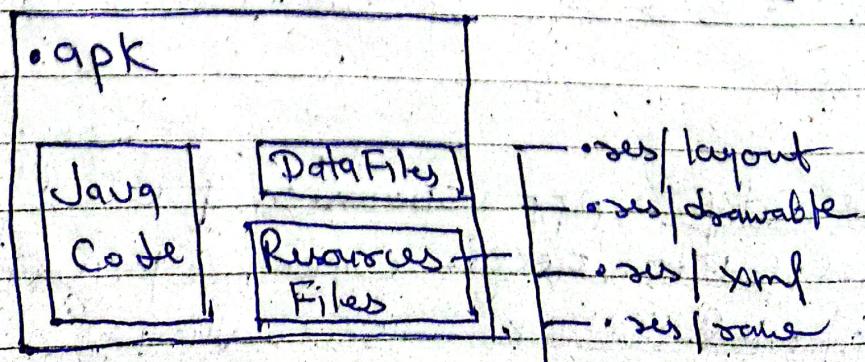
User Requirements

User Expectation

## Application Models for Mobile APP Frameworks

### Android App Package

- Android Apps are written in Java
- An Android Application is bundled by aapt tool into an Android Package (.apk)



## Application Components

Android Apps do not have a single entry point  
(e.g. no main() function).

They have essential components that the system can run as needed

### Components

- ① Activity : UI component typically corresponding to one screen
- ② Service : Background process without UI.
- ③ Broadcast Receiver : Components that respond to broadcast intents.
- ④ Content Provider : Component that enables app to share data.

### Components - Activity

- An activity is usually a single screen
  - implemented as a single class extending activity
  - display user interface controls (views)
  - Reacts on user input / events
- An application typically consists of several screens
  - each screen is implemented by one activity
  - moving to the next screen means starting a new activity.
  - an activity may return a result to the previous activity

## Components - Service

A service does not have a visual user interface but rather runs in the background for an indefinite period of time.

Ex → music player, network download.

Each service extends the Service Base Class

It is possible to bind to a running service and start the service if it is not already running.

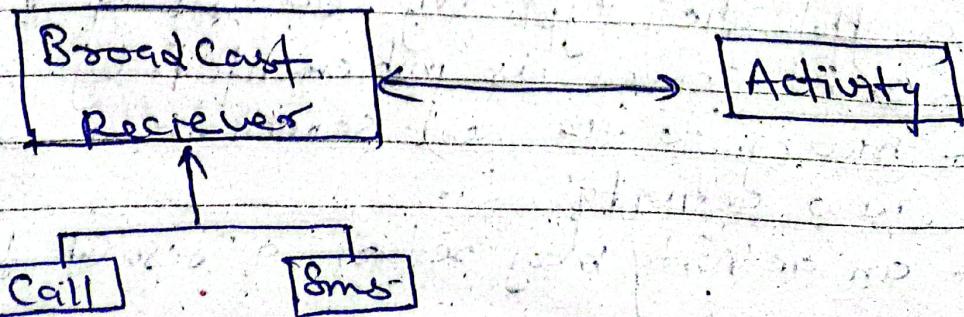
Adding a service with Android is quite similar than for an Activity

<service android:name = "app.LocalService"/>

## Component - Broadcast Receiver

A broadcast receiver is a component that receives & reacts to the broadcast announcements (Intents).

e.g. Announcements that the time zone has changed that the battery is low etc.



- Application can also initiate broadcasts.
- All receivers extend the Broadcast Receiver base class

### Components - Content Providers

A content provider makes a specific set of application's data available to other apps.

The data can be stored in the file system, in SQLite or in any other manner that makes sense.

Using a content provider is the only way to share data between android applications.

It extends the Content Provider Base Class and implements a standard set of methods to allow access to a data store

- Querying
- Delete, Update & insert

### Intents

- Intents are simple message objects each of which consists of
  - Action to be performed (MAIN/VIEW/EDIT/PICK/DELETE/DIAL)
  - Data to operate on

## Intent Filters

A component's intent filters in the manifest file inform Android

Manifest

Application

Activity

<intent-filters>

Action android:name="android.intent.action.MAIN"

</intent-filters>

### An Example

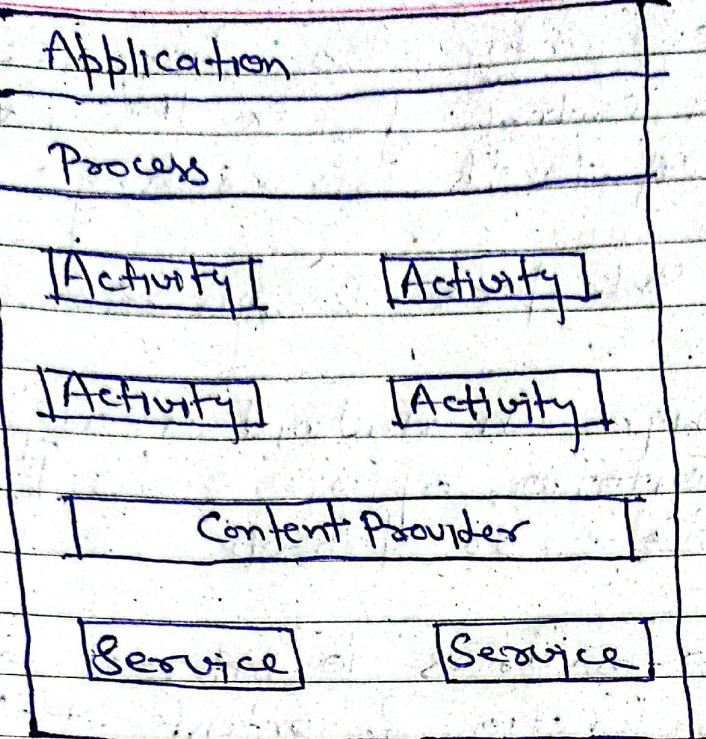
A component can have any no. of intent filters, each one declaring a different set of capabilities.

The first filter in the exq indicates that the activity is the entry point for app.

The second filter declares an action that activity can perform on a particular type of data.

## Android Component Model

- An Android App is packaged in .apk file.
  - A - apk file is a collection of components



### Apk File

- Components share a Linux process by default  
one process per .apk file.
- .apk files are isolated & communicate with each other via Intents or AIDL
- Every component has a managed lifecycle

### Data View Model

This is the model used by Swing library of Java.

The view is the interface through which the user interacts with the software.

Data are stored separately & can be displayed in different views.

The view may also change according to the context for ex the text changes according to the user's lang.

## User Interface Design in Android

UI Design in Android is a graphical representation of views displayed on smartphone or tablet

It allows users to interact with the features, function & contents of the app

To design UI there is no need to know any programming lang. Although it is nice to have web developing skills or programming skills

### What is View

- Views are used to customize UI design.
- A view is considered a building block for a proper User Interface created from the View Class

Ex → Rectangular Box

Buttons

Checkboxes

## Layout as a subclass of View Group

Layout defines the visual structures of UI

Layout can be declared through a program or through a simple XML layout file located in the resource layout folder of project you work on.

### Types of Layouts.

Android provides a number of layouts that provide different views.

#### ① Linear Layout

It is used to arrange views in a single row or column.

##### < Linear Layout

    android : layout\_width = "match\_parent"

    android : layout\_height = "match\_parent"

    android : orientation = "vertical" >

##### < Button

        android : id = "@+id/button"

        android : layout\_width = "wrap\_content"

        android : text = "Hello" />

##### </ Linear Layout>

## ② Frame Layout

It is a placeholder on the screen used to display a single screen.

## ③ Relative Layout

It enables to specify how child views are positioned to each other.

## ④ Table Layout

It groups the views into rows & columns.

## ⑤ Constraint Layout

It provides you with adaptable & flexible ways to create views for your applications.

## Unit of measurement

Dp - Density Independent Pixel.

Sp - Scale Independent Pixel

PX - Pixel

## Managing Application Data

Internal Storage

SD Cards

Databases

Shared Preferences

## Mobile Operating System

It is an operating system that helps to run other application software on mobile devices.

It is some kind of software as the famous computer operating systems like Unix, Windows and Mac OS.

## Popular Platforms of the Mobile OS

- ① Android OS - most popular OS, based on Linux kernel, Google
- ② Badg - Samsung mobile OS
- ③ Blackberry OS - developed by RIM (Research in Motion)
- ④ iPhone OS | iOS
- ⑤ Symbian OS Nokia
- ⑥ Windows Mobile OS Microsoft
- ⑦ Harmony OS Huawei
- ⑧ Palm OS Palm Ltd, Garnet OS
- ⑨ Web OS (Palm/HP) palm, based on Linux kernel