ANDROID MATERIAL

Android Architecture:

1. MVC (Model View Controller)

MVC or Model View Controller breaks the model into three main components Model that stores the application data , View UI layer that holds the component visible on the screen and Controller that establishes the relationship between Model and the View.

2. MVP (Model View Presenter)

To avoid complexities like maintainability, readability, scalability, and refactoring of applications we use MVP model. The basic working of this model relies of the points mentioned below:

- Communication between the View-Presenter and Presenter-Model happens with the help of Interface (also called Contract).
- There is One to One relationship between Presenter and View , One Presenter Class only manages One View at a time.
- Model and View doesn’t have any knowledge about each other.

3. MVVM (Model View ViewModel)

MVVM or Model View ViewModel as the name suggest like MVC model it contains three components too Model , View and ViewModel. Features of MVVM model are mentioned below:

- ViewModel does not hold any kind of reference to the View.
- Many to-1 relationships exist between View and ViewModel.
- No triggering methods to update the View.

And we can achieve this using 2 methods:

- Using DataBinding library of Google.
- Using Tools like RxJava for Data Binding.

Core Components

- **Activities**: Entry points for user interactions, representing a single screen with a UI.

- **Services**: Background operations without a user interface, such as playing music or fetching data.
- **Broadcast Receivers**: Components that respond to system-wide broadcast announcements.
- **Content Providers**: Manage and provide access to structured data sets, such as contacts or media.

Layouts

- **LinearLayout**: Arranges children in a single row or column.
- **RelativeLayout**: Positions children relative to each other or the parent.
- **ConstraintLayout**: Allows flexible positioning and sizing of children with constraints.
- **FrameLayout**: Designed to block out an area on the screen to display a single item.

UI Elements

- **Views**: Basic building blocks for user interface components (e.g., TextView, Button).
- **ViewGroups**: Containers that hold multiple views or other view groups (e.g., LinearLayout, ConstraintLayout).

Common Widgets

- **Toolbar**: A customizable action bar for providing navigation and interaction options.
- **TabLayout**: Provides horizontal layout to display tabs.
- **ScrollView**: Enables scrolling for content that's larger than the screen.
- **CardView**: A container with a rounded corner and shadow, used to display content.
- **GridView**: Displays items in a two-dimensional, scrollable grid.
- **ListView**: Displays a vertically scrollable list of items.
- **RecyclerView**: An advanced, flexible version of ListView, designed for large data sets.

Resources

- **Strings**: Text resources defined in res/values/strings.xml.
- **Colors**: Color resources defined in res/values/colors.xml.
- **Dimensions**: Size and spacing values defined in res/values/dimens.xml.
- **Drawables**: Graphics that can be drawn to the screen (e.g., images, shapes) defined in res/drawable.