Short Summary on Android App Architecture

Additional materials for 1D project

Introduction

- There are 3 common architecture patterns in Android: MVC, MVP, and MVVM
- There is no fix formula on how to apply the patterns, consider it as a design philosophy instead

MVC
Model-View-Controller

- **Model**: Handling business logic and communication with database
- **View**: UI interface layer, interaction to the users
- **Controller**: Connecting View and Model. Contains UI logic, takes user input, and updates Model.
- In Android, the **view and controller** roles are overlapping in **Activity** files
- You can easily separate the **Model** from the others

MVP
Model-View-Presenter

- **Model**: Handling business logic and communication with database
- **View**: UI interface layer, interaction to the users
- **Presenter**: Connecting View and Model. Contains UI logic, takes user input, and updates Model.
- The difference with MVC is that this approach can clearly **separate the Model and View**.
- There is a **one-on-one** relationship between View and Presenter. Activity files are exclusively the View component.

MVVM
Model-View-Viewmodel

- **Model**: Handling business logic and communication with database
- **View**: UI interface layer, interaction to the users
- **ViewModel**: Exposing data from Model to View. Contains UI logic, takes user input, and updates Model.
- The difference with MVP is that this approach has a **many-to-one** relationship between View and ViewModel.
- View has a reference to ViewModel and ViewModel **produce** the data and **update** views through **Observer** pattern.

Reading

- https://medium.com/upday-devs/android-architecture-patterns-part-1-model-view-controller-3baecef5f2b6
- https://upday.github.io/blog/model-view-presenter/
- https://upday.github.io/blog/model-view-viewmodel/