Architecture and concepts

1.Manage app state

Use bloc pattern (with flutter\_bloc package) to manage app state.

Bloc is similar to MVVM on Android, mainly based on streams and events to form a reactive architecture.

flutter\_bloc providers Flutter Widgets that make it easy to implement the BLoC (Business Logic Component) design pattern.

<https://pub.dev/packages/flutter_bloc>

2.Networking

Networking client : use dio package

<https://pub.dev/packages/dio>

Networking annotation/generator : use retrofit package (this relies on Dio client)

<https://pub.dev/packages/retrofit>

3.Dependency Injection

We use package get\_it to implement Dependency Injection for project

<https://pub.dev/packages/get_it>

4.Use Clean Architecture Approach

Diagram

Description automatically generated

a. Why:

- To adhere SOLID principles since we are using OOP for this project.

- Ensuring UI layers don't know what is going on at data layer at all.

- Might want to separate each layers into different packages.

b. The components:

Presentation - Domain - Data.

- Presentation layer consist of

+Widgets

+ BLoC

\* Bloc only manages UI state based on business logic

- Domain layer (Business logic layer)

+ Repositories (interfaces aka idea how the logic would behave)

+ Entities (which what UI needs)

+ Usecases (user stories)

\* Typically one function, but can be more if they are functionality related.

\* Remember, one class for one responsibility.

- Data layer

+Data Sources

\* remotes (API)

+Models

\* request

\* response

+ Repositories (Implementation from Domain layer)

c. Reference:

<https://github.com/ResoCoder/flutter-tdd-clean-architecture-course>

<https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html>

5/Locaization

We use intl to handle multi language for app

<https://pub.dev/packages/intl>

<https://pub.dev/packages/intl_utils>

6/Json parsing / serialization

This project is implementing [json\_serializable](https://pub.dev/packages/json\_serializable). It use build\_runner to generate files

7/Manage app’s assets

Image is handled by [flutter\_gen](https://pub.dev/packages/flutter\_gen) for auto-complete and not have to deal with typing mistakes.

8/Git Hooks

<https://github.com/evilmartians/lefthook>

We use it to :

Check the commit message for spelling errors.

Enforce project coding standards