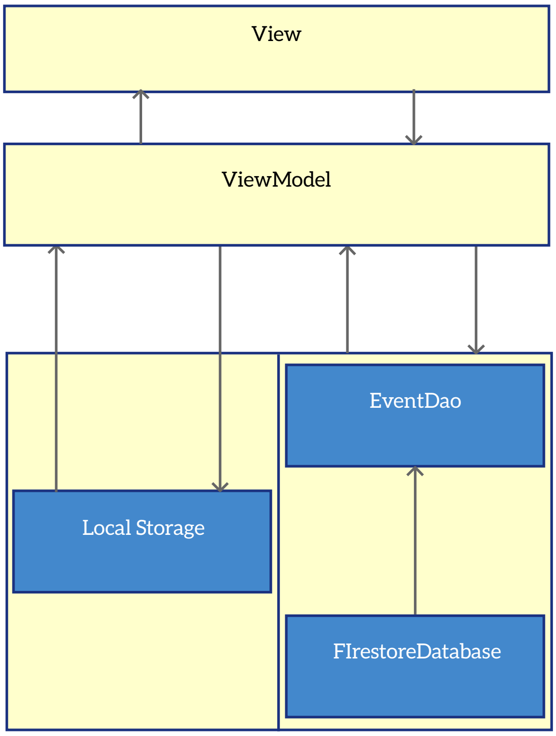
GenericX Architecture

Overview:

The app is divided into three distinct layers. The repository layer(the bottom-most one), the view model layer(the middle one) and the view layer(the top-most one).



The repository is the source of all data. The view model filters and transforms the data as needed by its view. It also is responsible for relaying back data changes to the repository. The view is responsible for displaying the data and handling UI events.

The repository pushes data to the viewmodel via RxJava2 Flowables. The viewmodel tells repository of any datachanges via normal method calls. The viewmodel exposes the data to the view through LiveData. The view tells the view model about the desired UI events via method calls.

Repository Layer:

It consists of two main sections: the first one is concerned with handling user preferences(like event filters) and the second one is concerned with supplying events to the view model.

The first section consists of the LocalStorage class. It is a reactive wrapper around SharedPreferences. It stores and retrieves user preferences. Since it is reactive, it automatically pushes the changes to its subscribers.

The second section consists of two classes interacting with each other: EventDao and FirestoreDatabase. The FirestoreDatabase retrieves the data from Google's Firestore service, exposes it through a reactive interface to the Repository. Whenever new data is pushed to it, the Repository:

- inserts the data into SQLite using the EventDao if it doesn't already exist.
- updates the data in SQLite through the EventDao if it already has been inserted before.
-

EventDao exposes the data in SQLite as Flowables. This allows the following flow of data: Whenever we change data in the Firestore service, the FirestoreDatabase receives it, pushes it to the database, which results in all the subscribers receiving the new data.