**Design**

I have used the following color palette, the base color of the app inspired by Internshala’s primary color.

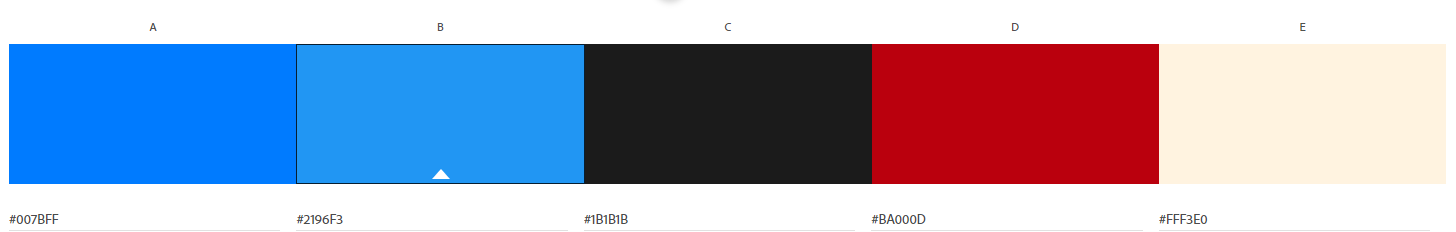


Fig 1: Color palette

**Firebase**

I have used the Firebase Google Authentication service to sign in the user using his Google account. The User’s display name is stored in the shared preferences.

**Jetpack Suite**

I have used the following jetpack libraries to incorporate best practices and write consistent and clean code.

**Navigation Component**

The Navigation component helped me ensure a consistent and predictable user experience by adhering to already established rules regarding user navigation in an application. I used 3 fragments whose navigation flows I defined in the navigation\_graph.xml.

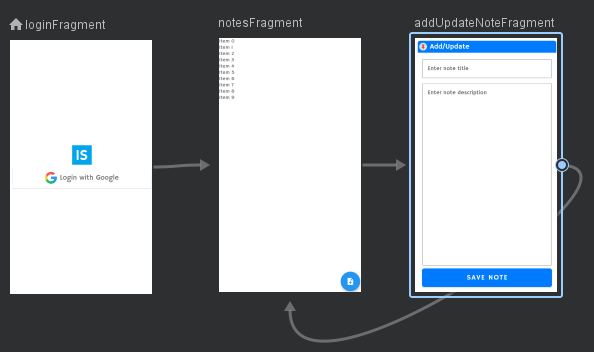


Fig 2: navigation\_graph.xml

**SafeArgs**

Instead of passing data between fragments via Bundles, which is the general mechanism used in Android, I have used a better alternative called SafeArgs. It is a gradle plugin that allows you to enter information into the navigation graph about the arguments that we want to pass to the destination fragment.

**View Binding**

I have used Jetpack’s View Binding feature to more easily write code that interacts with views.I replaced the traditional findViewById with generated binding objects to simplify code and remove bugs

**Project Overview**

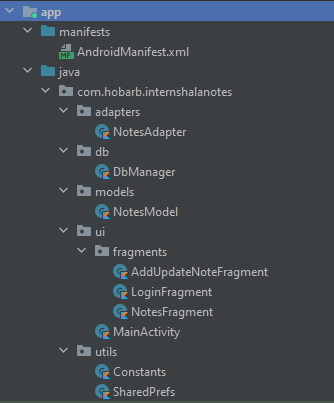
****

Fig 3: My project directory

I have separated my project directory into various packages for separation of concerns and to make the code more readable and clean. I have not used any design patterns because I felt it would be an overkill, since there was no online database and, other than the Google Authentication service, no api calls.

The **adapters** package contains the **NotesAdapter**, which contains the ViewHolder to wrapper around item\_note.xml which is the layout for a single note item.

The **db** package contains the **DbManager** class which has the logic for creating and managing the sqlite database. It is responsible for creating the notes table, and inserting/querying/deleting notes in it.

The **models** package contains the **NotesModel** which is a data class that describes the model for a single note.

The **ui** package contains fragments package which in turn holds my 3 fragments, and the **MainActivity**.

The **utils** package has a **Constants** class where I have defined constant variables used throughout the project. The **SharedPrefs** class is has two functions to read and write shared preferences value.

**UI**

**MainActivity -** Only one activity is used throughout the app. The MainActivity uses a FragmentContainer to display any one of the three fragments present in the app, at a time.

**LoginFragment** - This is the default fragment. It allows the user to sign in with his Google account. On successful authentication, the displayName of the google account is stored on the shared preferences of the app and displayed on the NotesFragment.

**NotesFragment** - This fragment has a TextView that displays the displayName of the Google account used to sign in. This is fetched from the shared preferences. The RecyclerView displays all the notes for the user from the sqlite database. The notes can be updated by clicking on them. The notes can be deleted using the delete button. The RecyclerView updates in real time on notes deletion. The FloatingActionButton can be used to create a new note.

**AddUpdateFragment** - This fragment is used to either create a new note or update an existing one, based on a unique alertCode passed from the previous fragment. On clicking the FloatingActionButton to create a new note, an alertCode = 1001 is passed to the AddUpdateFragment, and the Save Note button performs an insert operation in the sqlite database. On clicking a single note in the previous fragment, an alertCode = 1002 is passed to the AddUpdateFragment, the EditTexts are prefilled with the existing note’s details and the Save Note Button performs an update operation on the sqlite database.

**Duration**

I started the project on 15th June, 2020, and completed it on 18th June, 2020. Total time taken for the project is ~ 11 hours