

Fairsplit - Architecture

Technology used:

- Language: Kotlin
- UI Toolkit: Jetpack Compose (declarative UI)
- Navigation: Jetpack Navigation Compose
- Backend/Database: Firebase

Model-View-ViewModel (MVVM) pattern:

MainActivity is the main entrance point and sets the content view to the composable FairSplitApp which is responsible for the navigation between all other composables and equips them with their needed viewmodels.

Data Source:

- **Firebase:** access point for Firebase.
- **SessionManager:** handles local user session persistence and retrieval.

View:

- **FairSplitApp:** main entry point for Composable UI and navigation setup
 - **AppNavigation:** defines all navigation routes: NavController, NavGraph
- **LoginScreen**
- **RegistrationScreen**
- **MainScreen:** contains bottomBar to navigate between:
 - **GroupScreen:** user related group view. Actions:
 - Add a new group
 - Navigate to a group
 - **ActivityScreen:** shows all spendings related to the user
 - **GroupDetailScreen:** Actions:
 - Add new Spending
 - Settle Debt
 - Tab-Bar:
 - Edit Group
 - EditMembers
 - Delete Group (only as the creator)
 - Leave Group (only as a member)
 - Show Group invitation link
 - **ProfileScreen:** personal user data view
- **AddGroupScreen:** create a new Group:
 - Choose name, picture, description
 - Add members
- **AddSpendingScreen:** create a new Spending:
 - Choose the creditor
 - De-/Select debtors
 - Distribute debt evenly or individually
- **FriendsScreen:** view of user friends. Actions:
 - Add new friends / delete friends from list

- **LoadingScreen**
- **SettleUpScreen:**

ViewModel:

collection of classes that handle logic and communication between Firebase / UI

- **LoginViewModel**
- **RegistrationViewModel**
- **GroupViewModel**
- **SpendingViewModel**
- **UserViewModel**

Dialogs: dialog windows for different tasks, including simple choices or selection of options

- **AddFriendDialog:** dialog window for adding friends
- **AddUserDialog:** adding users into a group
- **EditGroupDialog:** editing users in a group
- **JoinGroupDialog:** accept/decline invitation
- **ShowGroupInvitationLinkDialog**

Miscellaneous:

- **UserCard:** clickable summarizing of a user

Features:

- We implemented DeepLinking through a domain we got from firebase hosting: You can share a link to a group which will open the app at the GroupDetailsScreen of this group. You will be prompted to accept or decline the invitation if you are not included in the member list. You will also have to login first if you aren't already.
- To make the payout process as efficient as possible, each user in a group is assigned a balance. The user with the highest negative balance starts by paying the user with the highest positive balance, and so on. This method minimizes the number of transactions needed for an optimal payout.

Data stored on firebase:

- Users
- Spendings
- Groups

Important aspects of Spendings and Groups: Groups know of each person and spending associated with them through the respective Id's, Spendings know each person and each Group they are associated with through Id's, this is needed to avoid nested datatypes, to maintain a simpler database structure.

Hint: User login credentials are stored inside the README on Github. Please read this file before testing.

