Candidate case (Flutter)

Start by creating a new flutter app, this app (across this document we will call it the container app) will split its screen so it can fit 2 apps inside it (a MaterialApp that wraps a Row which contains 2 MaterialApps)

The first app

This app will have a button that every time it is pressed it accesses the bored api, fetches an activity and displays it on the screen in a nice way.

and there should be a button on the app bar that opens a filter dialog, here are the filters:

- type
- Participants: 1 - 5
- min - max price: 1 - 200 USD, and an option to convert to TRY with 20x exchange rate (1 USD = 20 TRY)

each of these filters should be a radio option, so the user can only specify one filter.

each activity element should have a favorite button that will perform these actions:

1. add the activity to the favorites list which will be listed on a favorites screen which can be accessed through the bottom navigation bar.
2. add the activity to a shared stream on the container app, which will be listened to through the second app.

The second app

This app will have a stream listener on the favorites stream mentioned in the first app, the activity element ui should be the same with the first app but without the favorite option.

notes:

- each app should maintain a different state
- they can share some code (ActivityModel class, Activity List Item Widget, etc.)
- you can use any packages you want
- strict git usage is mandatory, there should be clean commits. the code should be published to a private github repo and shared with @catchpadadmin
- regular comments usage across the code is mandatory (don’t overdo it, use comments wherever needed)
- there should not be any plain string passed anywhere in the app. all the strings should be defined in a global map, so the app can be localized easily.
- the ui does not have to be extraordinary, just presentable and easy to use.
- adding documentation is advantageous