CS 402 Spring 2023

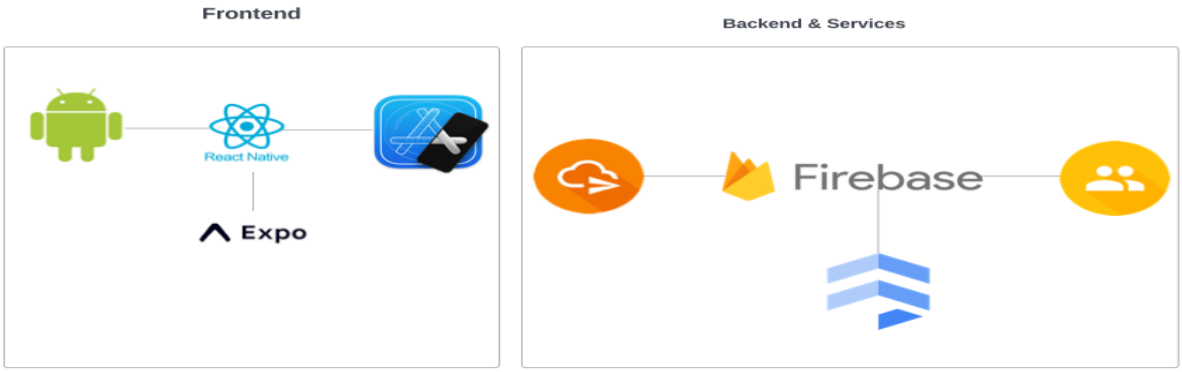
Robbie Gill
Tavi Kohn
Sina Khajehpour

Architecture
(Chat app)

Overview

For our project, various packages, services, and frameworks have been implemented. The main functionality of the project is about having connections between different users and creating a group chat for two or more users. The project can categorize into 3 main groups based on frameworks: 1. Frontend or user interface 2. Backend 3. Additional services

The front end has been implemented in the expo framework using Expo cli extension which uses the React native language. The backend of the project is supported by Firebase. Firebase uses the info of frontend and store data and users in the Firestore database. For services, technically we use Firebase as both backend and service. The implementation and functionalities of this project were satisfied by the actions of Firebase so we decided to keep our project as simple as possible to have a simple and fast application.



Static Architecture

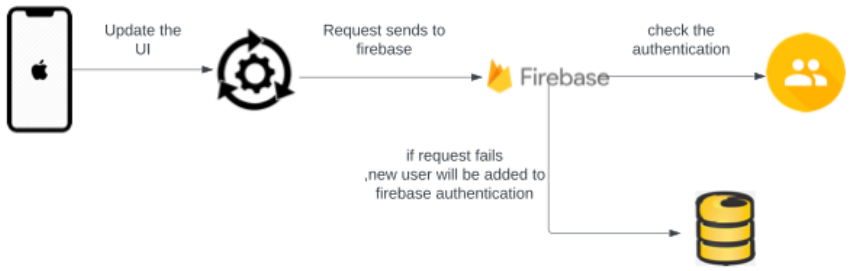
Frontend, backend, and services interact with each other in the following ways:

1.
The user requests to log in by requesting authentication from the Firebase authentication and getting permission from the Firebase authentication by calling the Firebase auth. In case the authentication is denied, another request will be sent to another page to create the user in the database.

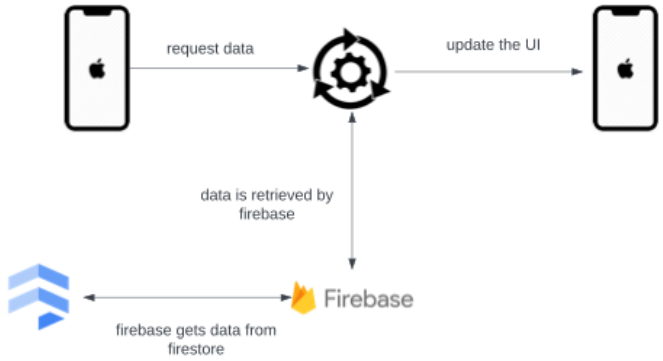
2.
After the authentication step, the user will navigate to another page (home page using the navigator) and a new request will be placed in this stage. The request will call the database API of the Firebase and retrieve all the lists of the Firebase database.

3.
Next, the user can make changes by adding new criteria or topics to the current list. This stage will be handled in two ways. First, add the current item to the current list using useState which counts as the local state. Second, it will be added to the Firebase database

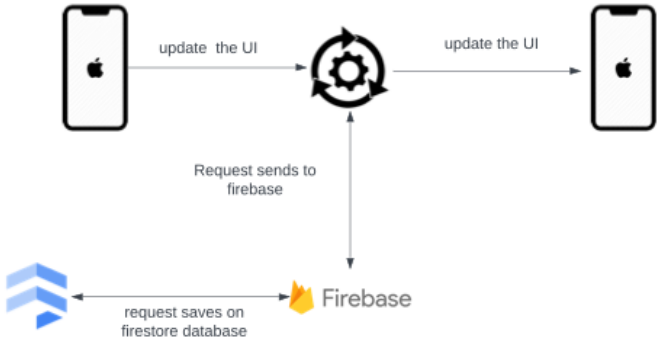
4.
the user requests a new set of data by choosing a topic or room. In this stage, conversations will be implemented from the Firebase database, and it will be shown as sender and receiver.



Next page after authentication approved

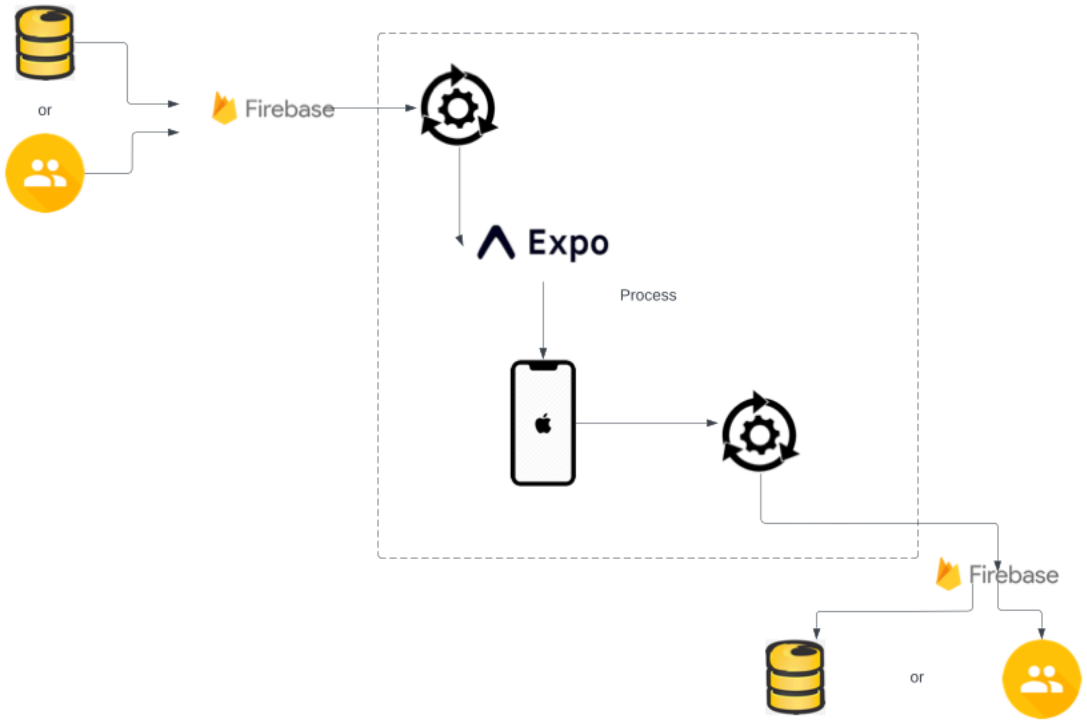


On creating a topic or Next page after choosing a topic or room



Mobile App Overview

The mobile app starts at the top level App.tsx file. In this file, authentications will be retrieved by the Firebase auth, and the authentication, username, and user id are stored. Two navigators use the stack model to store different screens. One of the stacks is used for storing login and signup screens and the other one uses the home and chat view screens. Inside each screen, we use different components and elements for the UI to have consistent performance. Finally, there are a bunch of different functions in each screen that connect each screen to the backend and help each screen to get/retrieve or update data from the database or UI.



Back End Overview

Most of the functionality of the backend will be supported by two main functions of Firebase: Firestore database and Firestore authentication. Authentication will give permission to a specific user to log in to the main room and it keeps track of the current user by calling the firestore/auth.if the authentication fails, the user can create a backend profile by creating an

account. The account would be requested on frontend and after that, it would pass to the authentication API and create a profile for a new user. The Firestore database will provide the information of all rooms and messages in each room by creating collections on the database and it will handle those in this way. I have to mention as well that these two services provide a lot of features that make the implementation consistent and easier. It covers all overlapping and keeps everything updated in case the user wants to make a change on UI which results in backend changes.

Below are two examples of Firebase services and how they work. The request can be read either by the Firestore database or Firebase authentication. Authentication read will parse the data and give set permission for the user. Database read will store the messages in the selected collection.