

The purpose of this assignment is to implement a simple (but functional) messaging app “simple chat” in Android, using Firebase’s Cloud Firestore as the backend. The app will have a chat screen and a contacts list (two separate activities).

You may want to complete the Friendly eats tutorial (intro to cloud firestore) before starting assignment:

This is also lab 8:

<https://firebaseopensource.com/projects/firebase/friendlyeats-android/>

Specs (20% each):

- 1) **Create Firebase account** (free) and login to console. In the database, create a new collection called “users” and add 2 documents, each with 1 field (name), name them “Jack” and “Jill”.
- 2) The **main screen** of your app will be a recycler view displaying a list of contacts. Immediately after the user logs in, this screen will appear, get a handle to the Firestore database using :

```
FirebaseFirestore db = FirebaseFirestore.getInstance();
```


and then fetch Jack and Jill from the database and use them to populate the RecyclerView. You can implement a spinner to indicate to the user data is loading, while db fetches the user info.
- 3) **Sign in** – implement email, Facebook, and gmail sign in using FirebaseUI and Firebase Authentication. Create a rule in the Firebase database under “rules” tab that only allows people who have signed in to read/write from the database (see *lecture 13, slide 11*). Once the user has signed in (before entering the app), add them to the Cloud Firestore database that you created in (1) as a user.
- 4) **Chat screen** - once user has signed in and RecyclerView has been populated with users (Jack + Jill + user who just signed in), should be able to tap a user to open a separate activity, where you can write/receive messages.
- 5) **Self-chat**: Select yourself from the recycler view user screen, opening a chat screen with yourself. This action should create a new document in Cloud Firestore database. Subscribe to the document using `addSnapshotListener` (*Lecture 13, Slide 24*). When you write messages in the chat screen, write it to a subcollection in the document on Cloud Firestore database. The chat window should then be updated using the callback from `addSnapshotListener`, so every time you write a message to yourself, it automatically duplicates itself in the chat window (because you will send the message to the online database, but you are also subscribed to the database, so you will receive it back immediately).

Bonus (+5% each)

- 1) When you initiate the keypad in chat screen, update a separate document on cloud firestore, that you are also subscribed to, and when the callback from this subscription triggers, show a “User x is typing” or “...” indicator in the chat screen (Basically, indicate in some way, through the database, that you are in the process of typing a message).
- 2) Request the `SYSTEM_ALERT_WINDOW` permission in manifest, and, with this permission granted, create a small window overlaying the current app (similar to facebook messenger’s chat head) whenever a callback from chat subscription is triggered (in 5). This can pop up in the current app (simple chat) whenever you receive back the message from Self-chat.

Resources:

Cloud Firestore is a flexible, scalable database for mobile, web, and server development.

<https://firebase.google.com/docs/firestore>

`SYSTEM_ALERT_WINDOW`:

https://developer.android.com/reference/android/Manifest.permission.html#SYSTEM_ALERT_WINDOW