# ReGroup

A group oriented Dating App

Alex Meier

## 1. Problem and Solution

The problem that ReGroup is trying to solve stems from the traditional dating app format. Apps like Tinder, Bumble, and their older siblings like eHarmony work by algorithmically pairing users together by interests, looks, and other factors (In the example of Tinder, users are paired by "swiping" on profiles they are interested in, and "matching" when they swipe on someone who also swipes on their own profile). Once two users are paired together by some means, they can chat with each other through text as a way to initially get to know each other before either moving forward with that person or continuing from step one to match with someone new. The problem with this is that it is not addressing a major aspect of meeting people in the real world, which is that people tend not to meet each other in a vacuum. In other words, most people are less likely to meet someone they like by talking to complete strangers with no context. Instead, they meet people who are friends of friends, or at work, or some other group setting. Being able to see how someone behaves and interacts in a group of peers is every bit as important as gaging their character through one-on-one conversation. This is the advantage that ReGroup hopes to provide to users over other similar apps.

In the current app market, there are some solutions to this problem among the popular dating apps. One way to solve this is to allow users to group up with their friends, and have their group "match" with other groups of friends. This allows group interaction between people that are already familiar with one another, which is already closer to an in person scenario, however there is a problem that this solution does not address. There is still somewhat of a stigma against online dating in general, so users may be uncomfortable with the idea of bringing their close friends along for the ride. The solution presented by ReGroup is different, however. The idea behind ReGroup is to match users with groups of other users rather than individuals. Instead of bringing a group of friends, ReGroup creates a group messaging chatroom for up to 4 people. These 4 people can chat and get to know each other. Users can save each other's profiles, and should they wish to chat one on one at a later time, they can do so with their saved friends. If they do not feel a connection to the group, they can leave the group and the system will generate a new group with new people. Profiles consist of a series of profile pictures, a username, and a short description.

## 2. Goals

The main goal of this project was to create a basic prototype (or part of one at least) of the app to encapsulate the basic features that the full version would have. The following are the main feature goals originally planned for the project:

1. The app must have some form of authentication in order to store user data remotely.

2. Users will be able to create a group and chat in real time with up to 3 other users. The chat functionality makes use of a remote server to send and receive messages to other users.
3. Users will be able to create a profile for themselves made up of a picture and a short description.  Profile data will be stored remotely on a cloud server such that one user is able to download and view the profile of another user.
4. Stretch Goal: Users will be able to save the profiles of other users and start an individual chat with saved users.

Of these goals, I have implemented the first 3, but due to time constraints and goal number two in particular taking much longer to get working than expected, I had to leave 4 as a stretch goal that was not completed.  While my app may seem a bit simple on the surface, quite a lot of work went into creating a back-end infrastructure stable enough to support 4 users sending messages to each other.


## 3. Limitations

My app has the following limitations, and features that did not make it to a usable state.
1. Due to the nature of the back-end implementation of the group message functionality, there are a variety of unintended behaviors such as duplicate messages, and users Occasionally appearing as being listed twice in the group profiles activity.
2. The "Saved Profiles" button in the main menu is nonfunctional, as this feature was planned but not implemented.  If users "saved" the profile of another user, this Is where they would be displayed.


## 3. Build and Install Instructions

### 3.1 Client Application

Building the main application is simple. Follow these steps to build:
1. Clone the git repository with the following command:
   git clone https://github.com/alexmeier2828/4630f2020.git
2. Open Android Studio, and click file->open and navigate to the project folder which is /4630f2020/ReGroup
3. Wait for the Gradle sync to finish.  The first time you open the project folder, this will take a few minutes to complete.  When the Gradle sync is complete, the green "Run" button will become clickable.
4. Press the Run button to build and run in the emulator.  If you do not have an android virtual device set up already, you will prompted to do so now. Once the AVD is up and running, the app should start.

Note: This app has been tested with android API version 30.  It may or may not run on a phone with an earlier version of android than this.


### 3.2 Backend Deployment

The backend for ReGroup is already deployed and running on Google Firebase which is linked to my google account, so it **does not need to be built locally**, nor can it be deployed

without my Google login credentials.  My API implementation resides in a single JavaScript file "/4630f2020/ReGroup/server/functions/index.js".  If there are any questions about my backend implementation, I can be contacted at my student email, [alex_meier@student.uml.edu](mailto:alex_meier@student.uml.edu).

## 4. Running Instructions

It should be noted that because this app is an online application, at least two devices, with two separate logins are required to properly demonstrate it working.  For that reason I have included a video file in the next section to help demonstrate what my app can do. The following steps will take the user through all of the features of the app:

1. To start, open the ReGroup app, and press the "Sign in with Google button". If this is the first time opening the app, you will be prompted to either select a google account to use for authentication, or if no account has been registered with this device, you will be prompted to log in through google.  Note:  for peace of mind, ReGroup's google authentication goes through the Google Play API, so Passwords are not handled by this application.
2. Once the sign in is complete, the apps main menu will open.  From here, you can press Profile to edit your user profile.
3. From the profile screen, you can press the "3 Dots" tool bar icon in the top right of the screen to bring up a menu.  In this menu you can tap "Edit Profile" to bring up a dialog that allows you to type out a text description that will be displayed with your profile.  Press "Done" on the dialog box to submit the changes.  To upload a profile picture, press the "Edit Profile Picture" menu option from the same menu as before.
4. From here, you will be prompted to choose between selecting a photo from the gallery or take a new picture with the camera. After selecting or taking a new photo, the new selection will appear in your user profile. This profile will be visible to other users.
5. To return to the main menu, press the android "back" button.
6. From the main menu, press the "New Group" button to go to start a new group chat.  Because of a limitation of the system, two users must have the app open and request a grouped in order to be grouped together.
7. To send a message (sending a message will still display a message bubble even if the group is empty), tap to edit the "text message…" EditText view, and use the keyboard to type out a message.  When ready, press the android "send" button attached to the keyboard on the lower right that looks like a right arrow.  When you press send, your message should appear in the message board above the "text message…" EditText view.
8. If there is more than one user in the group, pressing the "person" menu icon at the top right of the screen will display the profiles of other users who are part of the group.
9. Pressing the back button from the Group Chat screen will return to the main menu after prompting the user if they wish to leave the group.

## 5. Demonstrational Video

Because my app requires multiple logged in users on two or more devices to properly demonstrate, I have prepared a short demonstrational video to show some of my apps capabilities.

Video Link: https://youtu.be/G8oAqSLIvDE


## 6. Sources

The following Libraries were used in my project:

1. Firebase – A google service that provides authentication, database, and storage capabilities.  Documentation can be found here: https://firebase.google.com/docs/android/setup
2. Firestream – A lightweight messaging framework for android apps using Firebase.  This library was used to implement sending messages between two devices through a remote backend. Github: https://github.com/chat-sdk/firestream-android

The following Example code from documentation and other sources was used in my project as reference:

1. Reading and writing to Firebase Realtime Database from cloud functions - https://firebase.google.com/docs/database/web/read-and-write
2. Adding and retrieving documents from Firestore https://firebase.google.com/docs/firestore/manage-data/add-data#java
1. Loading an image from gallery - http://androidbitmaps.blogspot.com/2015/04/loading-images-in-android-part-iii-pick.html
2. Downloading an image directly into an image view from cloud storage - https://stackoverflow.com/questions/48762263/using-firebase-storage-image-with-glide/48762436