

Assignment 4

By Invitation Only: An Exclusive Chat App

In this assignment, you will create a group chat application that takes advantage of many features available through the Firebase platform. The application will incorporate common functionality of group chat applications to allow users to send messages to each other.

Overall Application Behavior

All users will have to be authenticated with Google authentication for the application. After signing in, users should be presented with a list of groups that they belong to. Navigating to one of these groups should take them to the group chat for that application. Users should be allowed to post a new chat message to a group.

Chat groups are **invite-only** and the only way to join one will be from an invitation from an existing member. Users of the application will be allowed to create a new group, hence becoming the first member, and then be allowed to issue invitations.

Authentication and Users

All users will have to be authenticated using Google authentication for this application. Once authenticated, the user should be added to a `user` database that tracks the users and the groups they participate in.

Using Firebase Cloud functions, you should send an email to a user when they authenticate for the first time welcoming them to the application.

Chat Groups

Chat groups are created by users who can invite other users to participate. A user should only see the groups that they have created or have been invited to. This should be enforced by the user interface in the client app (ie. It doesn't have to be enforced through permissions).

When a user creates a new chat group, they should automatically be added as the first participant. This can be done from the client app or using a Cloud Function. Chat groups should have a name and a unique identifier so they are unambiguous.

In the chat group list, you should show the number of users who are currently logged in to the group (ie. user presence).

Invites

A user who is in a chat group can invite other users to participate. This can be simply enforced by putting the invite button on the chat group scene or through any other mechanism you choose.

Using Firebase Dynamics Links, create a link that will bring the invited party into the application and automatically add them as a participants to the group. Users should be able to send the invitation using Firebase Invites or through standard UIKit.

For example, Sally is in a group named “Coffee Lovers” and she thinks Linus would like to be part of the group. Sally hits the “Invite” button the group message screen and sends the invite to Linus. Linus receives the invite as a SMS message and taps on the link. The application opens and he is prompted to authenticate with Google. After successful authentication, he is added a member of the “Coffee Lovers” group and taken directly to the messages for that group.

For this assignment, you can assume that the user has the application on their mobile device. For a true dynamic link install, you would need to have the app available in the App Store.

Group Chats

Groups chats consist of a single threaded messages that are sent by any members of the group. A chat message can consist of text or a image. The user who sent the message should be identifiable in some way. There is an excellent third-party view controller for chat applications (<https://github.com/jessesquires/JSQMessagesViewController>) that provides an interface comparable to native messages application.

Cloud Functions

As stated above, implement Firebase Cloud functions, to send an email to a user when they authenticate for the first time welcoming them to the application. In addition, you are required to implement another Cloud Function of your own design. This function should perform some processing on either the text or image posted as part of the chat.

Some examples might be to add an emoji if a keyword is detected (eg. :smile:, :angry:), color the text based on sentiment analysis, annotate a users image using to add metadata to user's posted image.

Notifications

Implement application notifications so that you can send a notification to all users of the app. Notifications only need to be sent through the Firebase console. Your application should be able to handle the notifications appropriately.

Database

Use Firebase Firestore to store the chat data.

Tutorials and Third-Party Code

This assignment aims to allow you to provide a fully featured application in a real-world scenario (ie. the Internet and tutorials exist). You can use any tutorials as references or third-party code for this assignment. Please include the required attribution. The majority of the work you will do will be interfacing with Firebase for your specific implementation.

Please look at the following tutorial for a chat application implementation that uses JSQMessagesViewController: <https://www.raywenderlich.com/140836/firebase-tutorial-real-time-chat-2>. While this application does provide much of the functionality we are going to implement, you will see that it does not use best practices in structuring the datastore (according to Firebase). You should take care to structure your database that will allow it to scale as discussed in class.

Requirements

The focus of this assignment is the backend design and development. You may consider this document to be guidelines to follow. You may choose an alternate strategy for any part as of the assignment as long as your application performs the required functionality.

There are no requirements for the user interface and design of the iOS application. You can design it such that it performs all the required task...although, it wouldn't take too much to make it look really nice.

The assignment is Due November 2, 2017 at 5:29 PM.

A bonus point will be awarded to applications that provide the "typing" functionality.