

whatsUp Documentation

Joo Yeong Moon (Sandy)

Ashley Hu

Overview

“What’s up? Where are you? What are you doing?” Have you ever received these questions from multiple people and found it tiring to repeat yourself? If you have, then you can easily solve this issue with **whatsUp!** A map-based social service app, whatsUp allows you to know where and what your friends are up to without the need to repeat yourself.

You can show your current location with an easy push of a button that places a marker on a map. Write a message for all your friends to see to save your hands from constantly retyping and upload a picture to show your current mood or surroundings!

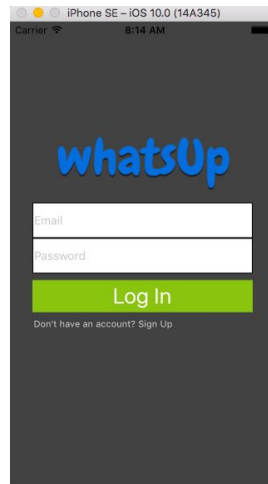
Essentially, the app is for users who are tired of retyping their whereabouts to their friends and for friends who are tired of asking where a particular friend is. The marker on the map can show details on what the user is doing at the particular location, and the time the user placed down the marker.

How to run

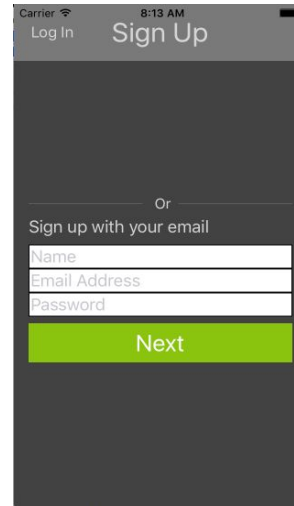
Be sure to run the whatsUp.xcworkspace file

1. Press the Run button in XCode to run the app, the app should start up in the login screen.
2. At the login screen, music will be playing in the background
3. **Log In/Sign Up**
 - a. In the login screen, you can enter a registered email and password if you have a registered account with us. If you do not have an account, press the “Don’t have an account? Sign up” button in the login screen. Take this chance to examine the custom segue animation. After the animation is completed, the Sign Up view will appear. Enter the necessary information and press “Next.” A success message will appear saying “Success. You made a new account!” After you get this message, press the “Log In” button on the top left of the screen. You will be able to log in with your newly created account.
 - b. For testing and this project’s purposes, we changed the read and write rules in firebase to true, meaning anyone can write to the database. This is only in place so we can access the accounts created without the need for actual authentication. Officially, we will change the rules back to the original default where it requires a user to authenticate their account.

Log In Screen



Sign Up Screen



4. Map

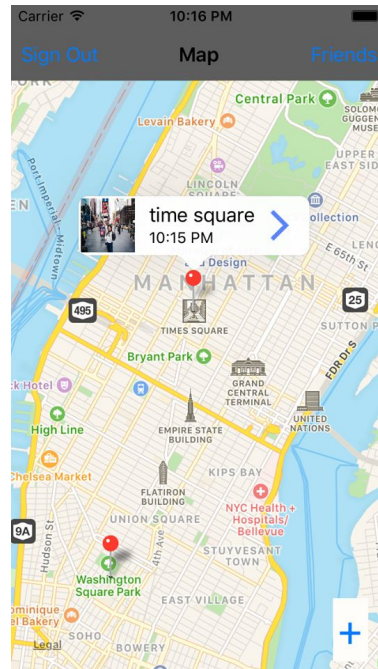
- a. After successfully logging in, you will see a map that utilizes the MapKit framework or API made by Apple. From here, you will not see any pins because you are not friends with anyone yet.
- b. **How to pin your location**
 - i. You can press the '+' button on the bottom right of the screen to add your current location. A Check In View screen will appear. Here, the user can add text to their location and upload a photo (photos are working, however because we have not worked with Firebase storage, the photos aren't saved. As an alternative, a default picture will be used). The location will grab the user's current GPS location and the current time. The location will then be stored it into Firebase under the key "checkin."
 - ii. User can click on the marker that was pinned. The marker will show the details on the pin.
- c. In the top right of the screen, you can sign out of your account

Sample View : Shows friend's location and user's location

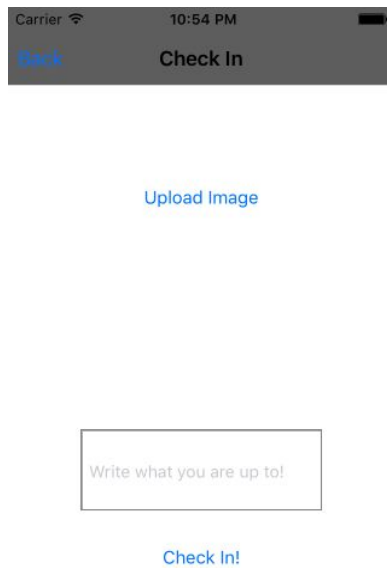
User pins location down



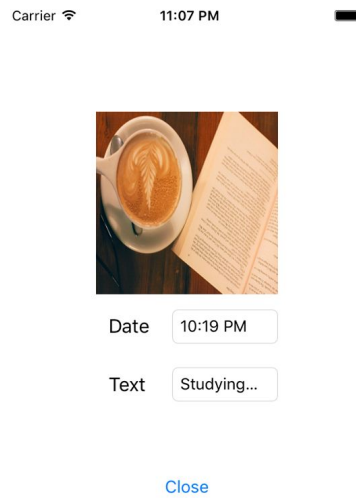
User can see friend's location



Check In View



When user clicks on the marker

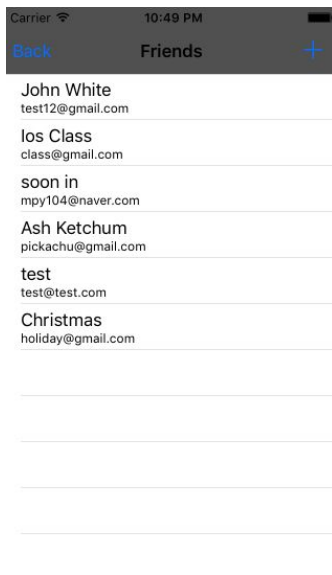


5. Friend List

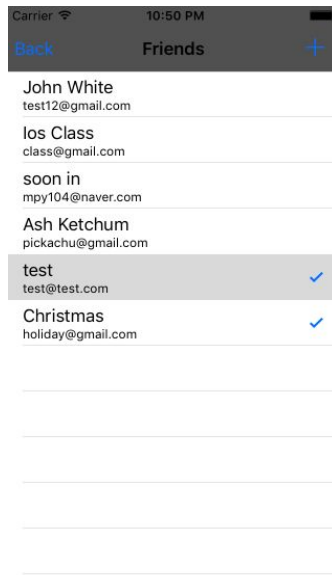
- Your friends can easily be checked by pressing the friend button on the top right in the navigation bar. If you are using a new account, you will see your own name appear because “you are your own friend.”
- To delete friends, one just simply needs to press a person’s name in the friends list and a check mark should appear. You can delete more than one friend at a time. The check mark signifies that the user is deleted off the friends list in the database. Press the “Back” button in the navigation bar on the top left to enter the map screen and press the “Friends” button again. You will now see your friend or friends deleted off your list.

Sample Friend List

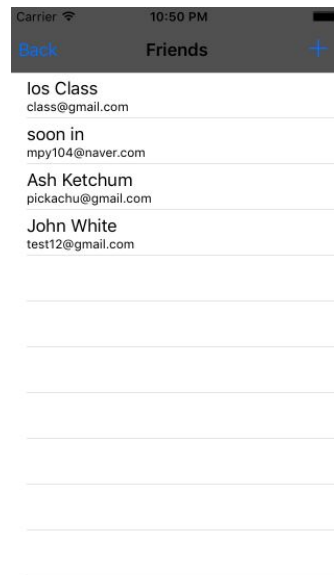
Friend List



Deleting Friends



Friend List after deleting



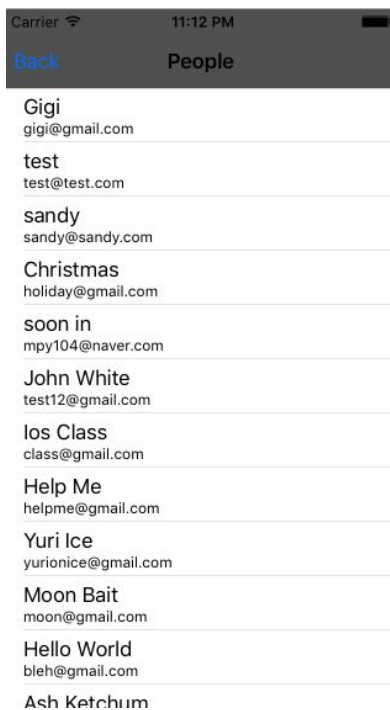
6. People List

- To add friends, press the ‘+’ sign in the top right of the navigation bar in the Friends List. From there you will see all the users that registered to firebase. Click on the user's’ name and the friend will be added to your friends list in the database, which is signalled by a check mark. You can add more than one friend

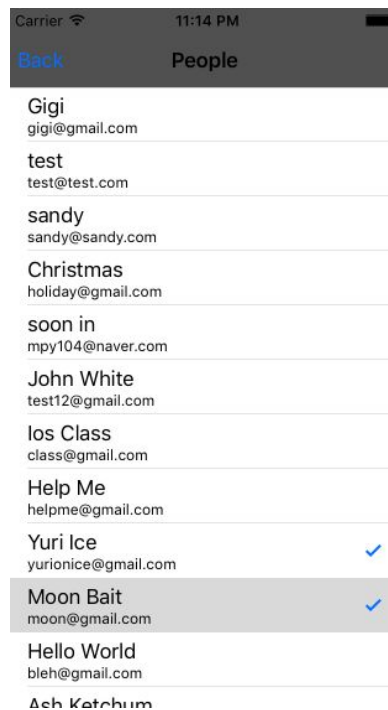
at a time. To check, navigate back to the map view and press “Friends” at the top right of the screen. The new friend or friends you added should now appear.

- b. We implemented a very rudimentary friending system for the purposes of this project. As stated in class, firebase is not a relational database nor does it have a friending system. The entire friending system was thought entirely by us which stores a user's unique id under the currently logged in user. The unique id is then searched through and all information regarding the user is pulled out
- c. If user presses the “Back” button, they will be brought back to the Friends List.

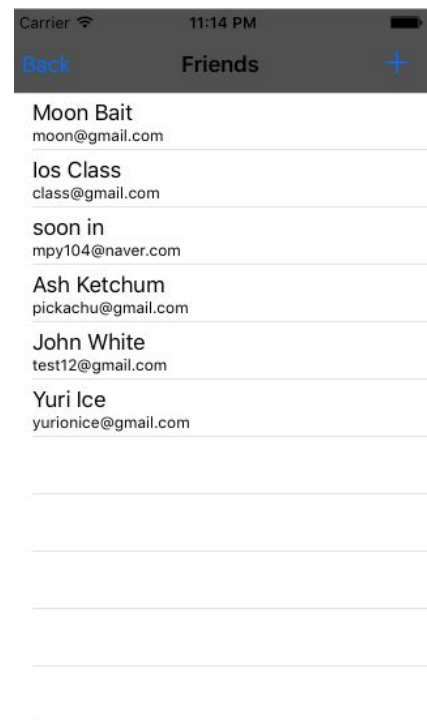
List of Registered Users



Adding users to friends



Friends list is updated



Database structure From Firebase

The string of numbers and letters are the user's unique id.



Features

- LogIn/Sign Up - Create or log in with account using Firebase Authentication
- Map
 - Allows user to check in and place marker down on map
 - User can write text and upload photo
 - User can click on placed down marker to see information
- Friend List
 - Shows the User's friends
- People List
 - Shows the list of registered users in Firebase Database

Other features that are included

- There is music playing during the LogIn Screen
- There is a custom segue animation from LogIn Screen to SignUp screen

API Used

We are using the MapKit Framework or API made by Apple. The API is used in the main map view after user logs in.

Technical Challenges

Technical challenges include learning how to use Firebase, storing data, and implementing the MapKit API from Apple. Firebase is not a relational database, so implementing a friending system was particularly challenging. Thinking about how to store data and retrieving information from Firebase, as well as ensuring the map shows the necessary data when user creates a pin, thinking about how to reload the map and showing their friend's pin on the map were things we had to consider and spent a lot of time thinking about. GPS was also challenging to keep track of because the simulator does not have an actual GPS to store location. Therefore we tested it using custom locations. The resources needed are Firebase for backend support, XCode, and MapKit API.

Future Implementations

Additional features we are considering is allowing the user to sync their Facebook friend list, add stickers, work with Firebase Storage, a better Friending system and have an even more visually appealing map. The visual part comes to play if a specific spot gets too crowded and has more than five markers, it should merge into one "super marker" on that spot. The user can then touch this "super marker" which will display a view that shows each of the details for the five or more markers. The Friending system will definitely have pending requests to ensure that the users are actual friends. Firebase Storage will enable users to upload their photos