

SIMPLE CHAT

The goal of this project is to demonstrate your ability to dig into new code and make use of Firebase. You should have been given access to an Xcode project which this exercise will be based on. The application here is a simple chat app. Feel free to delete and add from the project. Make sure you properly connect everything up after adding pieces. Whenever you write code, comment your lines like so: //HERE

Exercise: In this project you will be creating a compose message page. This app is based on the iPhone’s text message app. You will be using table views and search controllers. We recommend testing the app upon receiving it, as it is already working.

First steps – configure your own Firebase account:

* Create a Firebase project. Drag your new GoogleService-Info.plist into the Xode project to replace the current one.
* Create a new Realtime Database in Firebase.
* Create a storage folder titled: message\_images in Firebase.
* Remember Firebase’s security rules.

Project:

* Create a new view controller and set up the UI so that it mimics the iPhone’s Compose Message page.

**Hint**: There should be a search bar and table view with 2 sections: Contacts, Firebase Users.

* Ask for access to the user’s Contacts. Load the table view with the user’s Contacts. If no access is given, keep the table view empty.

**Hint**: Use the plist to request Contacts access.

* Manipulate the table view so that only contacts with an account show up.

**Hint**: You can check user accounts through Firebase. FirebaseDatabase.swift. You might consider checking how the app currently connects users, ie: createNewChatVC(). **Note**: It is unsafe to pull an entire User object from Firebase. You only need to check if the main user id exists, in this case phone number.

* Add a search bar. Filter both the Contacts and Firebase Users sections of the tableview. Add a label to a table view cell saying “N/A” if there are no matches.

**Hint**: This should filter both table view sections simultaneously. **Note**: You only need to pull the user\_name construct after you check that the user exists.

* Start a conversation with a user when that user is selected from the table view.

**Hint**: You might consider looking at the code we used to implement this originally. ChatSelectionViewController.

Useful functions:

Contact phone numbers are stored in various ways. Here is a useful function to ensure all numbers show up correctly. We recommend you use it when pulling trying to open a chat with a Contact’s number:

**func** formatNum(number: String) -> String {

**let** formNumber = String(number.filter { !" -()".contains($0) })

**var** newNumber: String?

**if** formNumber.hasPrefix("+"){

            newNumber = formNumber

        } **else** {

            newNumber = "+1\(formNumber)"

        }

**return** newNumber!

    }

Submission:

Send us a copy of your project over email ([hiring@valetfix.com](mailto:hiring@valetfix.com)) with a reference to your AngelList name.

We ask that you do NOT upload this project publically to Github. We had a candidate in the past try to reuse another person’s project that they found on Github. We use this project for multiple assessments and would like to keep it private. We do monitor Github.

If you have any questions, feel free to ask.

Best of luck!