The vast majority of my contribution to the product is HouseHub's group messaging function. Prior to this project, I've had no experience with the two major components that I utilized, namely databases and Swift, so it took me a while to find a nice rhythm when testing. The group message relies on Firebase in order to store and call messages and their associated attributes, so there are many potential points of failure between clicking "Run" and seeing a result. What became the best way to test the chat is to combine print statement monitoring and running two application simulators concurrently using different user accounts, which allowed me to observe the app's real-time nature.

I also contributed to the app's UserManager functionality, which ensures that all user attributes are set and found in JoinGroup, CreateGroup, LogIn, and Register functionalities. This too relied upon calls to Firebase, so I again faced many of the difficulties I encountered while testing the group message. I relied on cleverly placed print statements in order to test the values that certain variables held. I say "cleverly" because, in our application, there are many pieces that run concurrently; at times, my print test would fail yet the value would be correct if tested at a different point in the app's initialization.

If I were to start again, I would absolutely choose to employ testing stubs in place of using actual database calls while testing, which left more room for error and was expensive to make.