**Test Summary**

***Test Case #1*** In order to test the real-time on-screen and backend conversation logging, this test was devised to check the server response after a GET request for all the stored messages associated with the current user. **PASS**✅

***Test Case #2*** The team tested for user input to be sent to the database using the POST request. The required input is in JSON format with userID and text sent as a parameter in the body. If successful the expected return output from the server should be a 200 OK status, and initially our team could not find a way to successfully get the test to work because we were passing the parameters incorrectly. However once we found out how to send the parameter correctly the server sent a 200 OK status back to us and it. **PASS**✅

***Test Case #3*** Verified errors when submitting empty JSON objects onto the database. Responds with a 500 error to the client informing the object was not sent or NULL object was POSTed. **PASS**✅

***Test Case #4*** This test resulted in a pass if the mock server would properly link to the Firebase configuration. This would be implemented with a super-test, in order to validate that any further tests would be validated under a server. Failing this test would result in all test cases not being linked and otherwise failing. **PASS**✅

***Test Case #5*** verified the functionality of our keyword algorithm for our bot called findHighestScore(). The idea was to keep score of the various categories of keywords that could show up in a user statement. This list would then be put through the findHighestScore() function, which would identify the highest score and its index within the list and return the two values in a list of their own. If correct, the bot would then be able to accurately assess the user’s intent and respond accordingly. The test input is a list of sample scores, and the expected output is a list of appropriate values based on said input. **PASS**✅