Omni

Test Report Document Team 6

Table of Contents

Introduction

Purpose of this Document

The intended audience for this document is the CMSC 447 section 2 teaching team. The purpose of this document is to provide a description of the testing process used in creating Omni, as well as the test results and the team's own assessment of the testing process.

References

Omni User Interface Design Document

Havens, Micah, et al. "User Interface Design Document." 9 Nov. 2022

Omni System Design Document

Havens, Micah, et al. "System Design Document." 29 Oct. 2022, https://github.com/Skid-Team-6/Omnidocs/blob/main/Omni%20System%20Design%20Document.pdf

Omni System Requirements Specification Document

Havens, Micah, et al. "System Requirements Specification." 20 Oct. 2022, https://github.com/Skid-Team-6/Omnidocs/blob/main/Omni%20System%20Requirements%20Specification.pdf

Omni Project Proposal

Havens, Micah, et al. "Project Proposal." 12 Oct. 2022, https://github.com/Skid-Team-6/Omnidocs/blob/main/Omni%20Proposal.pdf

TR Template

Umrawal, Abhishek K., et al "Testing Report (TR) Instructions & Template"

Software Engineering 9

Sommerville, Ian, et al "Software Engineering 9th Edition", March 2010, https://ifs.host.cs.st-andrews.ac.uk/Books/SE9/index.html

Testing Process

Description

Testing for this project consisted of a number of trial and triage runs of our Omnichat demonstration webpage. This follows a standard quality assurance testing methodology in which testers actually use the end product and supply their own input, because this in and of itself provides close to real-world testing data, especially in terms of systems integration. In order to test the Omni server capabilities, we used a combination of unit, requirements, and integration testing between peer servers in Omni. Outside of these testing cases within Omni we also performed a number of tests on our user interface as described in order to make sure Omnichat worked as intended. If there was a malfunction of the Omni server or the Omnichat web server, it would quickly be indicated on the user interface (usually by a dropped connection), and the error would be logged in the console so that the bug could be assessed and fixed.

Testing Sessions

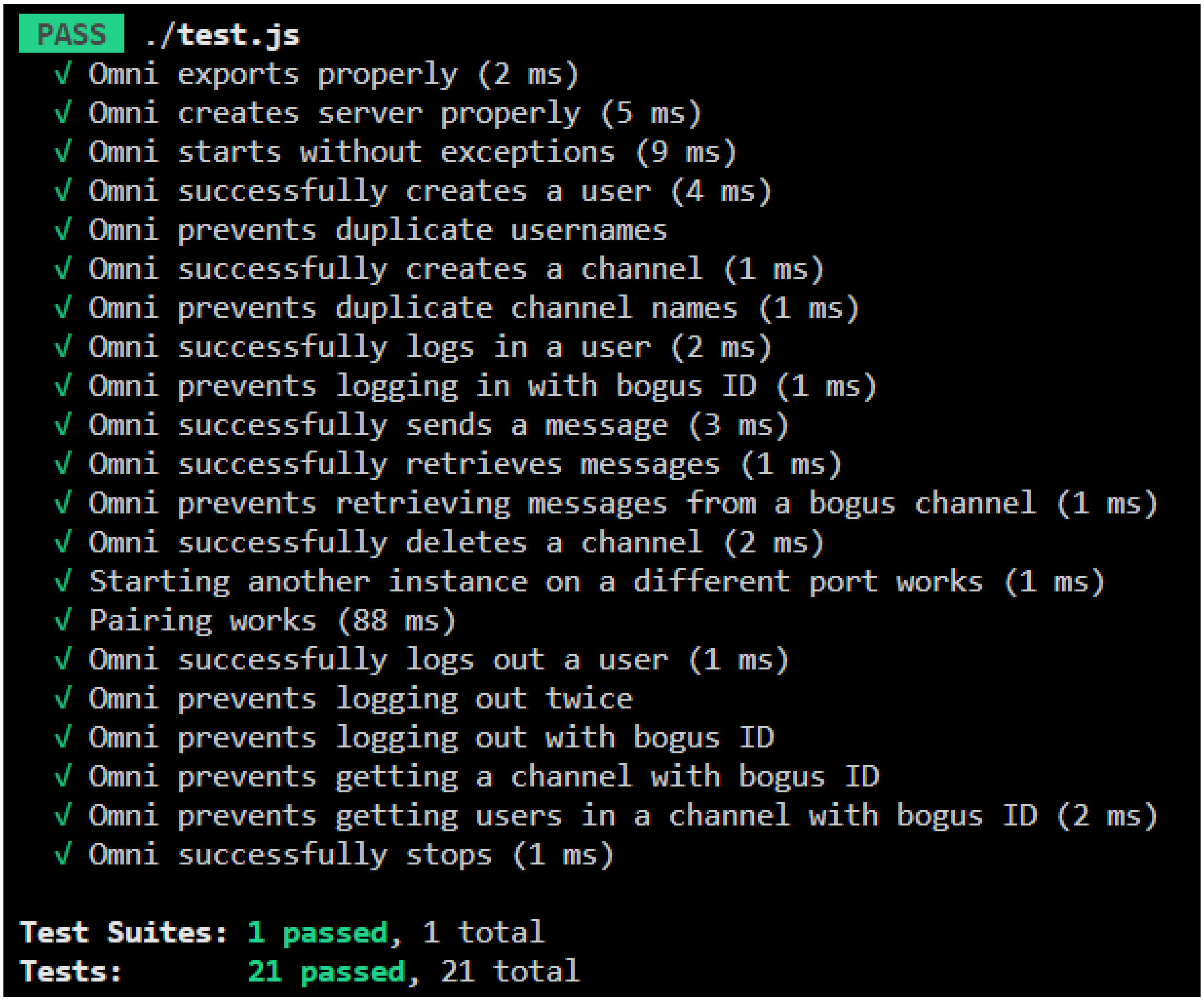
In order to test the user interface, also known as the Omnichat demonstration webpage, we ran numerous tests to uncover the limits of the Omni server and make sure as many aspects of the service as feasible worked as expected. To begin, we launched the page and made sure there were no initial errors in launching Omni. Next we logged in with a new user, old user, and admin user to ensure that each would not result in an error and that each type of user could enter Omni and use its services without any issues. Next we tested the abilities to create a new channel, create a hidden channel, and create an admin channel that could be accessed by one or more users and for them to chat freely. We then tested the pairing capabilities of the Omni chat. We attempted to pair a different server to the original and made sure we were able to see, access, and chat in the peer servers. So far, all of the tests have run smoothly with no issues. These sessions were held over Discord. As many members as possible participated in the calls in which we reviewed and discussed the tests.

Unit tests were written using the Jest framework. Because the API surface of Omni is very large, it was infeasible for the team to reach 100% code coverage in the tests. Instead, the team aimed for a more realistic goal of implementing as many of the SRSD tests as possible for Omni itself, and allowing manual testing to cover each of the requirements that we were not able to write as a test. In the end, we wrote 21 tests for Omni which together verify all of the fundamental functionalities of Omni as well as its ability to handle erroneous input to its various APIs.

Impressions of the Process

The code inspection process was overall very fluid. While this was not the part of the project we focused on the most, our group was able to do this alongside programming. We made sure to look out for our own mistakes as well as the mistakes of other members. Overall, our group was able to use our methodology to uncover most glaring mistakes or flaws, and because of this our code ran almost perfectly for our demonstration and in our manual usage of Omnichat.

Test Results



```
PASS  ./test.js
  √ Omni exports properly (2 ms)
  √ Omni creates server properly (5 ms)
  √ Omni starts without exceptions (9 ms)
  √ Omni successfully creates a user (4 ms)
  √ Omni prevents duplicate usernames
  √ Omni successfully creates a channel (1 ms)
  √ Omni prevents duplicate channel names (1 ms)
  √ Omni successfully logs in a user (2 ms)
  √ Omni prevents logging in with bogus ID (1 ms)
  √ Omni successfully sends a message (3 ms)
  √ Omni successfully retrieves messages (1 ms)
  √ Omni prevents retrieving messages from a bogus channel (1 ms)
  √ Omni successfully deletes a channel (2 ms)
  √ Starting another instance on a different port works (1 ms)
  √ Pairing works (88 ms)
  √ Omni successfully logs out a user (1 ms)
  √ Omni prevents logging out twice
  √ Omni prevents logging out with bogus ID
  √ Omni prevents getting a channel with bogus ID
  √ Omni prevents getting users in a channel with bogus ID (2 ms)
  √ Omni successfully stops (1 ms)

Test Suites: 1 passed, 1 total
Tests:       21 passed, 21 total
```

Appendix A - Customer and Contractor Agreement

The customer and Team including: Micah Havens, Scott Devere, C.J. Commodore, Adnaan Dasoo, and Josh Martin are agreeing to the implementation of Omni in accordance with the information listed above. The team and the customer are agreeing that everything listed above is acceptable and sufficient for the task that the customer needs. If future changes need to be made to this document all members of the team will meet with the customer to explain what needs to be changed and why, and upon agreement the changes will follow.

Dated Signatures:

- Micah Havens, 12/05/22, X____MH____
- Scott Devere, 12/05/22, X_____SD_______
- C.J. Commodore, 12/05/22, X_____CC_______
- Adnaan Dasoo, 12/05/22, X_____AD_______
- Josh Martin, 12/05/22, X______JM______

Customer Area:

Customer Comments:
Date:__________ Signature:X___________

Appendix B - Team Review Sign-off

All members, including Micah Havens, Scott Devere, C.J. Commodore, Adnaan Dasoo, and Josh Martin, have reviewed the system requirements specification document for our software, named “Omni”. Each team member has reviewed this document for accuracy and completeness in all parts, including text, diagrams, bullets, charts, and tables.

Dated Signatures:

- Micah Havens, 12/05/22, X____MH____
- Scott Devere, 12/05/22, X______SD______
- C.J. Commodore, 12/05/22, X____CC_________
- Adnaan Dasoo, 12/05/22, X_____AD______
- Josh Martin, 12/05/22, X______JM______

Appendix C - Document Contributions

- Micah Havens
  - Worked on: Description, Testing Sessions, Test Results
  - Percentage estimate: 20%
- Scott Devere
  - Worked on: Appendix, Introduction, Testing Sessions
  - Percentage estimate: 20%
- C.J. Commodore
  - Worked on: Impressions, Test Results
  - Percentage estimate: 20%
- Adnaan Dasoo
  - Worked on: Description, Testing Sessions, Impressions of the Process
  - Percentage estimate: 20%
- Josh Martin
  - Worked on: Purpose, Impressions
  - Percentage estimate: 20%