Omni

Code Inspection 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 detail the coding style used for Omni. It details the code inspection process used in making Omni.

References

Omni Style Guide

Havens, Micah, et al. “Style Guide” 9 Nov. 2022
https://github.com/Skid-Team-6/Omni/wiki/Style-Guide

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

CIR Template

Umrawal, Abhishek K., et al “Code Inspection Report (CIR) 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

Coding and Commenting Conventions

Code Style

When working on Omni, follow these style guidelines. Failure to follow these will slow the process of merging changes.

Whitespace

Use **tabs** instead of spaces for indentation and `\n` instead of CRLF line endings.

Variables and Names

Use **let** instead of var. Use **const** where appropriate. Variables and function names must use **snake_case**.

Correct:

```
let number_could_change = 20;
const number_stays_the_same = 10;
```

Incorrect:

```
var numberCouldChange = 20;
var numberStaysTheSame = 10;
```

Use **descriptive names**. It is better for a name to be long than too short. Unnecessary shortening **harms code readability**. Never use names like `x` or `num`, these do not tell the reader anything about what the variable represents. The only exception to this is in canonical uses such as `i` in for loops.

Strings

Use **double quotes** instead of single quotes. Use **string interpolation** with backticks where appropriate.

Correct:

```
let my_string = "what's up";
let new_string = `Hey everyone, ${my_string}!`;
```

Incorrect:

```
let my_string = 'what\'s up';
let new_string = 'Hey everyone, ' + my_string + '!';
```

Brackets

Brackets should start on the same line as the name of whatever is being defined.

Correct:

```
function my_function(some_parameter) {
//...
}
```

Incorrect:

```
function my_function(some_parameter)
{
//...
}
```

Code Inspection Process

Description

Any time code was committed to the Git repository we inspected it to make sure it aligned with the outlined coding style. We would review the code together utilizing static inspection through analysis of the code and symbolic execution.

Impressions of the Process

We believe our code inspection process caught many bugs and design anti-patterns that otherwise would have remained in the codebase, since having multiple eyes on one unit of code means that one person is more likely to spot a problem that the rest of the team does not see. Likely the best units of code are the message send and receive event handling units because they are factored nicely and are written expressively. The worst are probably the peer-to-peer communication handlers, since they contain duplicated and unorganized code. Duplicating code can cause bugs to duplicate as well, since any bugs present in the original code will be carried with it when it is copied and pasted. The message identification code is also essentially a large switch-case block, which is unwieldy.

Inspection Meetings

The meetings were held over Discord where the team reviewed commits to the Git repository that were made between meetings. Each team member took turns explaining their commits, and the rest of the members would voice any comments, questions, or concerns regarding the logic, semantics, and style of the code, among other qualities. After the meeting, revisions could then be made in separate future commits.

Modules Inspected

The Omni server, Omnichat web server, and Omnichat front-end are the individual modules which were each inspected for flaws and defects. Listed below are the defects we discovered during our final code review and inspection session.

Defects

Omni Server

#	Description
1	Omni peer-to-peer communication is not secure. It uses insecure WebSockets to connect, it would be better to use secure WebSockets (WSS).
2	No message authentication. A malicious Omni server is able to impersonate another Omni server.
3	IP address parsing is inconsistent. The pairing process might break if valid but unusual IP address formats are used. Anything that is not a perfectly formatted IPv4 address may cause errors.
4	The Omni server API is ad hoc, making it difficult for chat server administrators to understand and use. This negatively impacts user friendliness and ease of use.
5	Many APIs on the Omni server do not do enough validation. Despite testing, it is still possible to submit bad data to some API functions and get unexpected results.
6	We did not decide on whether admin-only channels should be visible to admins of peer Omni servers if they are not also marked as private. Because of this, admin-only channels are never visible to peers. We are not sure if this is the final behavior we would like this feature to have.
7	There is a DoS/spam vulnerability in which a malicious Omni peer could send many channel and message creation requests to another peer, causing the channel list to fill for all connected users and/or causing a mass spam of messages in one or more channels. This could be fixed with rate limiting and having a hard limit on the number of channels a peer can have.
8	Message backlogs are not shared between peers. If a user requests a backlog of messages from a remote (peer) server, this request should propagate up to the Omni server, which should then request these messages from the peer. The functionality for this exists within the code, but there is no logic that uses it yet, and it is as of yet untested.
9	Peers can take actions on behalf of remote users. A malicious peer server may be able to cause a user on another peer to appear online or offline incorrectly. It may also be able to delete user IDs on a remote server. This can likely be fixed with just a few lines of validation.

10	Code architecture. The Omni server should be refactored so that the main JavaScript file does not contain as much code as it does. It is difficult to maintain due to its size.

Omnichat Web Server

#	Description
1	It is not possible to view the backlog of messages in a channel more than 50 messages previous to when the user first viewed the channel. Omni provides the API to have this functionality, but we did not successfully implement the logic for it in Omnichat in time.
2	The web server does not use HTTPS. We decided this was a defect we were willing to accept, given that obtaining and using a certificate was likely beyond the scope of a class project. However, a good messaging platform should always communicate over HTTPS rather than HTTP.
3	Pair requests can be lost. The web server should cache notifications of pair requests that are received from Omni so that if there are no administrators online, the pop-up can be sent to the next administrator who logs in. Currently, if no administrators are online, the pair request will be essentially ignored.

Omnichat Front End

#	Description
1	The message pane may not reliably scroll all the way down to display the latest messages when a new message is received.
2	Inconsistent formatting - occasionally the channel list will appear wider or thinner, it is not always the same width. We believe this is caused by the user ID field at the bottom of the channel list.
3	Overflow is untested in some panels. We are unsure what happens if the channel list or user list is particularly long. It is likely that it breaks the layout in some way.

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: Coding and Commenting Conventions, Impressions of Code Inspection Process, Inspection Meetings, Defects
    - Percentage estimate: 25%
- Scott Devere
    - Worked on: Introduction, Appendix, Coding and Commenting Conventions, Code Inspection Process
    - Percentage estimate: 25%
- C.J. Commodore
    - Worked on: Modules Inspected, Defects
    - Percentage estimate: 15%
- Adnaan Dasoo
    - Worked on: Inspection Meetings, Impressions
    - Percentage estimate: 15%
- Josh Martin
    - Worked on: Modules Inspected, Defects
    - Percentage estimate: 20%