

## RTM

<i>Team Member</i>	<i>Responsibilities</i>
R-E Miller	<ul style="list-style-type: none"><li>- Team leader.</li><li>- Handled initial client/server connection code.</li><li>- Wrote majority of the Client code and assisted in Server code development for bug fixes and adding additional features.</li><li>- Educated teammates on setting up and managing a GitHub repository.</li><li>- Researched user interface technologies like Textualize and Curses (though not implemented in the final product).</li></ul>
Elliot Swan	<ul style="list-style-type: none"><li>- Developed all of the Shared Libraries code.</li><li>- Addressed bugs in both Client and Server code.</li><li>- Managed JSON related functionalities.</li><li>- Assisted Matthew in transitioning from C to Python and in learning Python basics.</li><li>- Conducted research on Curses and various methods for handling concurrent client connections (select, poll, thread, etc).</li></ul>
Matthew Hill	<ul style="list-style-type: none"><li>- Authored the majority of the Server code.</li><li>- Managed the GitHub repository to prevent merge conflicts.</li><li>- Created a significant portion of the logistical documentation.</li><li>- Developed the logging system for the server.</li></ul>

## Assumptions

Network Environment:

- Assumes that there is a stable network connection for uninterrupted client-server communication within the chat server application.

Unique Nicknames:

- Assumes that each client/user will choose a unique nickname for identification.

Graceful Shutdown:

- Assumes that the server can be terminated gracefully, using a keyboard interrupt (“CTRL-C”), which will signal all connected clients to shutdown.

JSON-based Communication:

- Assumes that all messages sent and received are in JSON format for consistent and structured data exchange using Python.

Command-Line Arguments:

- Assumes that users are familiar with running Python scripts and providing command-line arguments for hostname and port number.

#### Single Chat Room:

- Assumes all clients communicate in a single chat room, without private messaging or multiple chat rooms.

#### Port Number Range:

- Assumes that the server port number used is within the range of 10000 to 65535.

#### Error Handling:

- Assumes that basic error handling, such as network errors, or unexpected client disconnections, are handled and output to the user.
- Assumes that the user is able to understand basic error handling messages.

#### Message Display Integrity:

- Assumes clients must wait to receive a message before sending one to avoid message overlap or display issues.

#### Makefile Usage:

- Assumes that the user is familiar with Makefiles for compiling and executing.
- Assumes that the user will run “make clean” before compiling the application.

#### Open Network Port:

- Assumes the specified server port is open, unused, and not blocked by network security measures.