Protocol

# Types of Messages

## /say [message]

Example Command

* /say Hello

Requires a message be given after the command. This type of message can be sent by both the user and the server. This is used to allow the user to send messages to all other users connected. When passed to the server it will add the sender’s username and then pass it onto all other users.

## /msg [recipient] [message]

Example Command

* /msg Jim Hi There

This type of message represents a whisper from one user to another. It requires an input of the recipient’s username and then the message to be sent to them. This type of message can be sent by both the server and the user. When the server receives it, it will add the sender’s username to the message and passes it onto the corresponding user (if the user is connected).

## /list

Example Command

* /list

This type of message requires no input. It is sent by the client to server to receive the list of currently connected users. The response from the server is a space separated list of connected user’s usernames. The initial message is always sent from the client and the server solely uses it as a response.

## /info [newUsername] or /info [msg]

Example Commands

* /info Bob
* /info Jim is now Bob

This message is used by both the server and the client. By the client it is used with a new username to update the username, with the required input being the new name. By the server it is used to sent information about other users connected, such as users joining/leaving or users updating their name, with the input being the informational message.

## /error [message]

This message is used solely by the server. It is used as a response to other messages from the client if there is an error with it. The error message is sent to the user to inform them of the issue.

# Design Choices

I chose to use “/” type commands because they would be the most intuitive not only to implement but also to the user. Additionally, having a human readable protocol makes it easier to identify issues with messages when debugging. I chose to implement an /error command as it allows the client to differentiate from normal messages sent by the server and display them to the user differently.