Protocol Document

Introduction to Software Engineering

Server Team #1*

Monday, February 27th, 2017

In essence, our server will act as an IRC-esque chatroom. Because we are starting with the Minimal Viable Product (MVP), it will not particularly have all the features of an IRC server; also, because we have to provide a JSON-based API, that alters the design drastically. This document will provide a sufficient outline on the authentication process, the basic objects (user, channel, message), and other miscellanies. Required fields are marked by a red asterisk (*), and possibly required items are marked with a blue asterisk (*).

1 Authentication

At the time of the authentication (i.e. the login process) a few item will have to be provided.

1.1 Input

User* The user logging in. Refer to Section 2 for details and input. **String(s)/Boolean.**

Channel A list of channels that the user wishes to join. At the time of this writing, channels are statically joined, meaning only at the time of authentication can you join channels. If no channels provided, it is advised for the client to play elevator music. Refer to Section 3 for details and input. **Array.**

1.2 Output

Session Key Will be used in future development. **Array.**

Username Will be used to verify if submitted username was allowed. **String.**

---

*Ian Howell, Hunter Mathews, Illya Starikov, William Thurman, Zachary Wileman

Channel List Success Will return an a array of objects to specify the success of joining the channels. **Array of Objects** [{Name, Success Status, [Users, ···] }, ···].

2 User

The user object will help identify the particular user and session. Because the respective client teams are anarchists, we have the option to specify anonymity. If the anonymity option is selected, a randomized ID will be returned.

Name The name of the user. **String.**

Password* The password of the user. Can have any characters. **String.**

Anonymous* An option to allow anonymity when communicating. A randomized username will be returned, and IP will be added to government watch list. **Boolean.**

Username* A unique identifier to represent the user. If submitted, and not unique to channel, a randomized one will be returned. Can only consist of a mixture of numbers $[1 \cdots 9]$, letters $([a \cdots z]|[A \cdots Z])$ or underscores (_). **String.**

3 Channel

The channels is the actual location of the chat room. All messages posted by the user(s) will appear here. A channel can act as a direct message mechanism as well, by simply having two users.

Name The name of the channel. **String.**

Password If there a password, specify it. If there is no password, any submissions (including SQL injections) will be accepted.

4 Message

A message is a simple object with what you'd expect a message to have. Simple message object will be sent and received (with all the contents below).

Timestamp The time *the client* sent this message. A JavaScript format time object should be sent. **String.**

Sender* The *username* of the sender. **String.**

Message Actual message contents. No format characters will be accepted (i.e. \t, \n, or \r). **String.**

Channel What channel the message was posted on. **String.**