

CSCE 438: HW2

Colin Banigan & Katherine Drake

FB Chat Client Program

When the client program is run, the client is put into “command mode,” after which the user can issue one of the following commands to the server:

- LIST - Prompts the server for a string containing the formatted list of all of the users and the chat rooms that they are connected to.
- JOIN - Prompts the server to add the client to a specified chat room noted as an argument. A success string or an error string is then shown clientside.
- LEAVE - Prompts the server to remove the client from a specified chat room noted as an argument. A success string or an error string is then shown clientside.
- CHAT - Switches the client from “command mode” to “chat mode”. The 20 most recent client messages that the client is a part of are then displayed.

Chat Mode - Using both a writer and a reader thread, the client listens to the server for messages from users that it is JOINed to. If a message is sent by the client, the writer thread sends the data to the server. Messages from the server are interpreted through the reader thread and then displayed in the format of “[date/time]<username> message”.

FB Chat Room Server Program

When the server program is run the server connects to the host and port, and attempts to load the backend database, which contains the persistent data for all of the chat rooms, if this database exists. Client programs can issue one of the following commands to the server:

- LIST - The server returns a list of all the usernames in the chat room database to the client who issued the LIST request.
- JOIN - The server first checks if the specified client to join exists. If it does exist, the server then checks to see if the specified client to is not already added to the client. If both of these conditions pass, the command adds the user to the specified client and returns a success string. Otherwise, an error string is returned.
- LEAVE - The server first checks if the specified client to leave exists. If it does exist, the server then checks to see if the specified client to leave has already been added to the client. If both of these conditions pass, the command removes the user to the specified client and returns a success string. Otherwise, an error string is returned.
- CHAT - Retrieves the relevant chats that the client is subscribed to from the textfile “chathistory.txt”. Using a compare dates function, the relevant chats are then sorted by

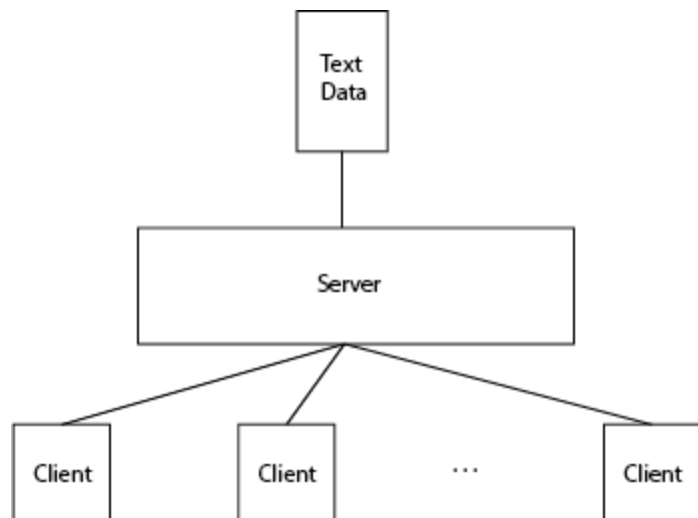
most recent post to least recent post. The 20 most recent posts are then passed back to the client before they begin chatting.

Chat Mode - Reading the messages from the client, the server then checks which users have the user that sent the message followed. If the user was following the user who sent the message, the message is then forwarded to that client to be interpreted client side.

Interaction Between Chat Client and Server

When a client first connects to the server the client's username is passed as a message. If this username is not in the server's database, a new chat room is created for that user. The server sends a message back to the client to inform them if their username was found or if a new chat room was created for that username. After creating the initial login information, the client enters "command mode" where it can enter LIST, JOIN, LEAVE, and CHAT commands respective. If a CHAT command is entered, the client then

switches into "chat mode" and the 20 most recent relevant messages to the Client are then promptly displayed. Once in "chat mode", the client is unable to switch back into "command mode" except through client termination. In "chat mode", the client is able to post messages similar to a Facebook wall which can be seen by people who have JOINed the respective user. In the same manner, the client is able to see messages posted by users that the client has JOINed itself with.



How to Run

- To compile
 - Run "make"
- To run the server
 - For default port: run "./fbsd"
 - For specified port: run "./fbsd <port>"
- To run the client
 - Run "./fbc <hostname> <port> <username>"