Libraries used in client

1. stdio.h
2. stdlib.h
3. winsock2.h
4. String.h
5. cJSON.h

Functions of client

int openSocket (void);

This function creates a socket for client and verifies it. If it was done successfully, function will assign port and IP to our socket and finally our client will be connected to server socket and this function returns client socket as output to use that in our functions.

void accountMenu (void);

The first menu of our application with just two options, register and log in. Each of them calls appropriate function. Unvalid number calls accountMenu again.

void reg (void);

Asks for username and password of user and prints server's response to request. Successful registration calls login function, otherwise reg calls itself again.

void login (void);

Function designed to login user. AuthToken is received from server in this function and is parameter of other functions. Successful log in calls mainMenu and unsuccessful login calls login again.

void mainMenu (char * authToken);

Another menu, this time with three possible options, creating a channel, joining a channel or logging out of account. Unvalid number calls mainMenu again.

void createChannel (char * authToken);

It wants name of channel and sends a request to channel. Successful creating channel calls chat menu but unsuccessful one calls itself.

void joinChannel (char * authToken);

It takes a name from user and joins existing channels. If the channel is not available it will return joinChannel again. If the user joins the channel it calls chat Menu.

void logout (char * authToken);

It sends log out request and logs out from account successfully.

void chatMenu (char * authToken);

The last menu which has the most options. Send message, refresh, viewing channel members and leaving channel are available in both phases but searching in messages and members of channel are exclusive to phase 3.

void message (char * authToken);

This function is designed to send a message to channel. The user sends content of message.

void refresh (char * authToken);

It sends a request to server and in answer, prints all of user’s unseen messages of channel.

void members (char * authToken);

This function is created to send a string to server and gets a string in JSON type from server and shows members of channel to user.

void leave (char * authToken);

This function leaves the channel and goes back to chat Menu.

void req (char *request, int size);

This function is designed to call openSocket and make a client socket, sends and receives buffer server and closes the socket. This function is called in many functions in client side.

void searchMember (char * authToken);

It gets a name and searches in names of members of the channel to find this name. The input and name of member must be exactly same.

void searchMessage (char * authToken);

This function is designed to find the exact word as the input from the user in messages of the channel. It prints the messages with senders and if there is no occurrence of this word, it will print “Not found”.

Libraries used in server

1. stdlib.h
2. stdio.h
3. stdbool.h
4. string.h
5. winsock2.h
6. time.h
7. ctype.h

Function in server

void openServer();

This function is designed to create and verify socket, assign port and IP, bind created socket to given port and ip and accept client socket.

void connecttoClinet(int client_socket);

This function receives and sends buffer to client, which is the main bridge between them. Also, this function calls other functions to perform and is called by openServer.

void regs();

This function checks if username is available or not. If it was, it makes a new file under “/Resources/Users” and writes password in it. If it wasn’t, it sends an error to client.

void logins();

At first, this function checks if username is existing or not. If yes, it checks password with the input and anwers with an appropriate response.

void logouts();

This function removes user from online users' array.

void makechannel();

This function checks if the name of channel is available or not. If it was, it makes a new file under "/Resources/Channels" and writes creation message in it. If it wasn't, it sends an error to client.

Format of creation message: "USER created the channel"

void joinch();

At first, this function checks if channel is available or not. If yes, it returns successful message and adds joining message to channel's messages, otherwise it sends an error to client.

Format of joining message: "USER joined the channel"

void refres();

First, this function finds out authToken is for which online user and then puts unseen messages into an array of JSON strings and sends it to buffer.

void leaveChannel();

It leaves the channel and sends the successful message to buffer and also reads channel messages and sends leaving message to channel.

Format of leaving message: "USER left the channel"

void sendmes();

It gets a message and an authentication token, verifies token and reads channel data and after that sends message to the channel.

void chMembers();

It verifies token first and searches in online members to find members of this channel and puts into an array of CJSON strings.

void searchMembers ();

It finds whose token is in the buffer and by that, identifies channel and after that compares the name in the buffer with the name of members of the channel.

void searchMessages ();

After verifying token, this function finds channel and searches for the exact occurrence of this word in content of messages of the channel and puts appropriate messages in the buffer.