# Multi\_User\_Messaging

## Server Component:

### Function Prototypes and Explanations:

1. **void \*handle\_client\_messages(void \*arg)**
   * **Inputs:** void \*arg (client socket)
   * **Outputs:** None
   * **Explanation:** This function runs in a separate thread for each connected client. It handles incoming messages from the client, interprets user input, and performs corresponding actions.
2. **void getCurrentDateAndTime(Date \*date)**
   * **Inputs:** Date \*date (a structure for holding date and time information)
   * **Outputs:** None
   * **Explanation:** Retrieves the current date and time and stores it in the provided Date structure.
3. **void listContacts(char \*userID, int client\_socket)**
   * **Inputs:** char \*userID, int client\_socket
   * **Outputs:** Sends a list of contacts to the client
   * **Explanation:** Reads the user’s contacts from a CSV file and sends them to the client.
4. **void listMessagesFromUser(char \*userID, Message message, int client\_socket)**
   * **Inputs:** char \*userID, Message message, int client\_socket
   * **Outputs:** Sends messages from a specified user to the client
   * **Explanation:** Reads messages from a CSV file, filters them based on the specified user, and sends the result to the client.
5. **void deleteMessage(char \*userID, Message message, int client\_socket)**
   * **Inputs:** char \*userID, Message message, int client\_socket
   * **Outputs:** Sends a confirmation or error message to the client
   * **Explanation:** Deletes a specified message from the user’s message history and sends a confirmation or error message to the client.
6. **void addUser(char \*userID, Message message, int client\_socket)**
   * **Inputs:** char \*userID, Message message, int client\_socket
   * **Outputs:** Sends a confirmation message to the client
   * **Explanation:** Adds a user to the contact list and sends a confirmation message to the client.
7. **void deleteUser(char \*userID, Message message, int client\_socket)**
   * **Inputs:** char \*userID, Message message, int client\_socket
   * **Outputs:** Sends a confirmation or error message to the client
   * **Explanation:** Deletes a specified user from the contact list and sends a confirmation or error message to the client.
8. **void sendMessage(char \*userID, Message message, int client\_socket)**
   * **Inputs:** char \*userID, Message message, int client\_socket
   * **Outputs:** Sends a confirmation message to the client
   * **Explanation:** Sends a message from the user to another user, updating the recipient’s message history.
9. **void checkMessages(char \*userID, int client\_socket)**
   * **Inputs:** char \*userID, int client\_socket
   * **Outputs:** Sends a list of messages to the client
   * **Explanation:** Reads the user’s messages from a CSV file and sends them to the client.
10. **void sortTheCSVFileAccordingToDate(char \*messagesCSVPath)**
    * **Inputs:** char \*messagesCSVPath
    * **Outputs:** None
    * **Explanation:** Sorts the messages in a CSV file based on their dates.
11. **int compareDates(const Date \*date1, const Date \*date2)**
    * **Inputs:** const Date \*date1, const Date \*date2
    * **Outputs:** Returns an integer (comparison result)
    * **Explanation:** Compares two date structures and returns the result.
12. **void createCSVIfNotExists(char \*userID)**
    * **Inputs:** char \*userID
    * **Outputs:** None
    * **Explanation:** Creates contacts and messages CSV files if they don’t exist for the given user.

## Client Component:

### Function Prototypes and Explanations:

1. **void displayMenu()**
   * **Inputs:** None
   * **Outputs:** None
   * **Explanation:** Displays the menu options for the user.
2. **void sendMessageToServer(int client\_socket, char \*userID)**
   * **Inputs:** int client\_socket, char \*userID
   * **Outputs:** Sends a message to the server
   * **Explanation:** Takes user input for sending a message, constructs a Message structure, and sends it to the server.
3. **void receiveMessagesFromServer(int client\_socket, char \*userID)**
   * **Inputs:** int client\_socket, char \*userID
   * **Outputs:** Displays messages received from the server
   * **Explanation:** Receives and displays messages from the server.
4. **void addUserToContacts(int client\_socket, char \*userID)**
   * **Inputs:** int client\_socket, char \*userID
   * **Outputs:** Adds a user to the contacts list
   * **Explanation:** Takes user input for adding a contact, constructs a Message structure, and sends it to the server.
5. **void deleteUserFromContacts(int client\_socket, char \*userID)**
   * **Inputs:** int client\_socket, char \*userID
   * **Outputs:** Deletes a user from the contacts list
   * **Explanation:** Takes user input for deleting a contact, constructs a Message structure, and sends it to the server.
6. **void displayContacts(int client\_socket, char \*userID)**
   * **Inputs:** int client\_socket, char \*userID
   * **Outputs:** Displays the list of contacts
   * **Explanation:** Requests and displays the list of contacts from the server.
7. **void displayMessagesFromUser(int client\_socket, char \*userID)**
   * **Inputs:** int client\_socket, char \*userID
   * **Outputs:** Displays messages from a specified user
   * **Explanation:** Takes user input for a specified user, constructs a Message structure, sends it to the server, and displays the response.
8. **void deleteMessageFromUser(int client\_socket, char \*userID)**
   * **Inputs:** int client\_socket, char \*userID
   * **Outputs:** Deletes a message from a specified user
   * **Explanation:** Takes user input for a specified user and message, constructs a Message structure, sends it to the server, and displays the response.
9. **void clearScreen()**
   * **Inputs:** None
   * **Outputs:** None
   * **Explanation:** Clears the console screen.

### Screenshots:

Screenshot 1 Screenshot 2 Screenshot 3 Screenshot 4 Screenshot 5 Screenshot 6 Screenshot 7 Screenshot 8 Screenshot 9 Screenshot 10 Screenshot 11 Screenshot 12 Screenshot 13 Screenshot 14