Distributed System Exercise 2 Report

Deliverable for Task 2.4

Protocol Design

This task is basically based on Remote Procedure call (RPC) in week 6

In the function onMessage, checking if there's socket.client or not. Socket.client is set by very first onConnect as None. So if user join the program, it check socket.client is none or not, if not then checking name is in list of client or not. If not, it just put name in the list of client. If there is, it send message to try user type again.

Next if socket.client is set, the order of command going through is Quit, SENT_ALL, WHISPER, CONNECT_ALL and message. First one is Quit because it needs to check User wants to Quit first, and the order of SENT_ALL,WHISPHER,CONNECT_ALL doesn't matter and the last part is sending message about checking command and parameter.

List of Commands : Quit, SENT_ALL, WHISPER, CONNECT_ALL, MESSAGE(message)

1. Quit

   The purpose of Quit command is making user to quit the program whenever the user wants.

   The command is just "Quit" so can just type Quit.

   IF command equal to Quit

   Send message Quiting program

   Return False

2. SENT_ALL

   The purpose of SENT_ALL command is sending a message to everyone that's connected to the server. How the command work. How the command work is about

   SENT_ALL message. For example, SENT_ALL hello world

   Pseudo code

   IF command equal to "SENT_ALL":

   FOR user in list of clients:

List of client [user]. send(message)

3. WHISPER

The purpose of WHISPER command is about sending a message to a specific user. The way the command work is WHISPER USERNAME MESSAGE. For example, WHISPER firstuser hello.

Pseudo code

IF command equal to WHISPER:

Separation = []

FOR word in parameter split:

Separation.append(word)

Specific user = separation[0]

If specific user in list of clients:

Message = join separation[1:]

SEND message to specific user

ELSE

SEND message "Wrong person try again"

4. CONNECT_ALL

The purpose of CONNECT_ALL command is for sending a list of registered/connected user to user. The command is just CONNECT_ALL. Then it will directly show the list of current users.

Pseudo code

IF command equal to CONNECT_ALL

Message = " "

FOR user in list of clients:

Message += user

SEND message to socket

5. MESSAGE (message)

The way the message command work is just sending message to itself and showing command and parameter in server. It basically sending message to server and command wise user can type either message or MESSAGE/. How the command work is Command and message. For example, message(or MESSAGE) hello world.

Pseudo code

IF lower case Command not equal to message:

    Send " Invalid command"

ELSE:

    IF parameter equal None:

        Invalid parameter

    ELSE:

        Print command ,parameter

        SEND message to itself

Furthermore, For the deadlock when the server is closed, client is closed so I deal with sending message to user to press enter to make an OS error about bad input so user can notice the server is closed and can get out from the program and can escape deadlock. Put send message in onDisconnect function.

Task 2.7 Critique

1. Is the protocol you created for this exercise stateless? Explain why or why not, or explain why this concept may not be applicable in the context of this application.

   Those protocol I created for this exercise is quite stateless because for the command wise. It works fine however, In program wise, it doesn't work well. The reason why I think is that client can't recognise the server even if the server is terminated or closed by accident so I need to send message to client to recongnise it But it's not just proper closing just giving an error to escape the program. so it can't be applicable in the context of real application.

2. Describe in detail how the current implementation could be extended/improved to allow subgroups of users to be created and messages to be sent only to members of a subgroup.

   For this problem, we can make separate subsets for subgroup chatting so if user only want to send a message to subgroup then user only can send a message to user who in the subset. For example, making an another dictionaries for subgroup so only put subgroup's socket in that dictionaries so It makes user can only send the message to subgroup.