2024

CSC3002F

ASSIGNMENT 1

AHMMAH003
PLXNIG001
RMKYAS002

NETWORKS

DESIGN AND FUNCTIONALITY

SERVER DESIGN

The server for this application was designed for the purposes of storing, updating and providing user information. For these purposes, an additional class (User.py) was created, which stores a user's username, password, status, IP address and the port number of its UDP chat socket. The server maintains a list of User objects in order to serve its purposes. The list is used to validate logins and add new users. It is used to get and set the status of a user, which allows clients to see and change their statuses. The server also uses the list to generate a String list of users and their statuses, which can be sent to clients for viewing. All messages sent between clients and the server are over a TCP connection. A separate thread is created for each client connection to the server, meaning that a new TCP socket is created for every connection. The server also uses its list to send peer information (status, UDP socket) to prospective chatters- allowing for initialization of UDP connections between clients.

CLIENT IMPLEMENTATIONS

CLIENT 1 : YASH [RMKYAS002]

The client application has been implemented to make the user experience as easy as possible. It allows users to log in to the server, validating username and password by communicating with the server. Once logged in, the user will be presented with a menu of options- chat, list clients, set status, exit. This list will be presented on loop- meaning that when the user finishes running through an option, the list will be presented again. The user's username and status (Available, Offline or Busy) will also be displayed each time the menu is printed. When chat is selected, the user will be able to enter a peer's username, and will be connected to the peer via UDP if they are available for chatting. If they choose to list clients, a tabbed and headed list will be printed. If they choose to set their status, their status will be changed on the server. If a user has set their status to 'Away', they will not be shown in the user list.

CLIENT 2: MAHIR [AHMMAH003]

The client side application allows the user to communicate with the server and receive information from server by means of set protocols. The program prompts user for login and password, which either creates a new user profile or logs into existing profile determined by the server. The username validation is done client side where it restricts the username to not be certain keywords and only alphanumerical, for the sake of minimalism and reducing errors. .

After successful login, the main menu is displayed which consists of options such as chat, show clients list, set status and logout. When chat is selected, the user displayed with any recent chat requests the user received. The use then proceeds with chat by entering peer username. If requested user is available, user will be assigned to a chat session where the user can send to and receive messages from peer. The user can also request and be shown the list of all users along with their statuses in a formatted manner. The program also also allows user to set their status.

CLIENT 3: NIGEL [PLXNIG001]

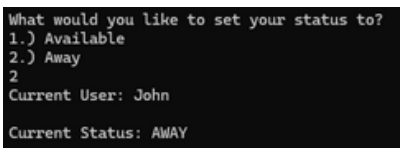
The client application was developed with the goal of simplifying the user interface. Through communication with the server, it enables users to log in with their password and username and will check whether their a returning or new user. The user will get a menu with the choices to chat, list clients, set status, and quit after logging in. This list will be shown in a loop, so after the user has completed using one choice, it will be shown until the user logs out. Every time the menu is printed, the user's username and current status (Available, Offline, or Busy) will also be shown, this status may be changed using the set status option. When the chat option is chosen, the user may input the username of a peer and, if the peer is available for chatting, they will establish a UDP connection, if the user wants to leave the chat they may simply type "EXITCHAT" .This client uses a substitution encryption as an extra security feature. The list clients option is used to show a list users which do not have their status set to offline.

SCREENSHOTS

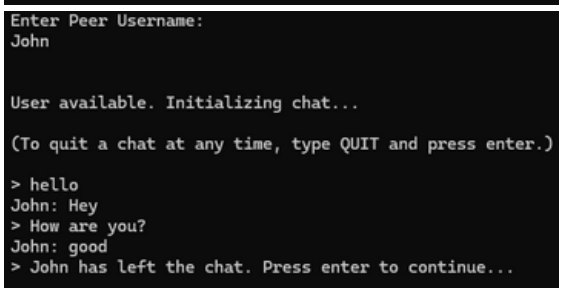
LOGIN

LIST

SET STATUS



CHAT



PROTOCOL SPECIFICATION

LOGIN PROTOCOL:

Client message:

LOGIN	cr	lf	**USERNAME**	sp	**username**	cr	lf
PASSWORD	sp	**password**	cr	lf	**IP NUMBER**	sp	**ip number**
cr	lf	cr	lf				

Server response:

SUCCESSFUL	sp	cr	lf	**REASON**	sp	cr	lf	cr	lf

OR

UNSUCCESSFUL	sp	cr	lf	**REASON**	sp	cr	lf	cr	lf

* Examples of REASON:

- Existing login
- New login
- Incorrect password
- User already logged in

GET STATUS PROTOCOL:

Client message:

GETSTATUS	sp	cr	lf	**USERNAME**	sp	**username**	cr	lf
cr	lf							

Server response:

STATUS	sp	cr	lf	**userstatus**	cr	lf	cr	lf

SET STATUS PROTOCOL:

Client message:

SETSTATUS	sp	cr	lf	**USERNAME**	sp	**username**	cr	lf
newstatus	cr	lf	cr	lf				

LIST CLIENTS PROTOCOL:

Client message:

LIST	sp	cr	lf

Server response:

LIST	sp	cr	lf	**username[0]**	cr	**status[0]**
cr	**IP[0]**	cr	lf	**username[1]**	cr	**status[1]**
cr	**IP[1]**	cr	lf	**...**		

CHAT PROTOCOL

Client message:

CHAT	sp	cr	lf	**peer username**	cr	lf	cr	lf

Server response:

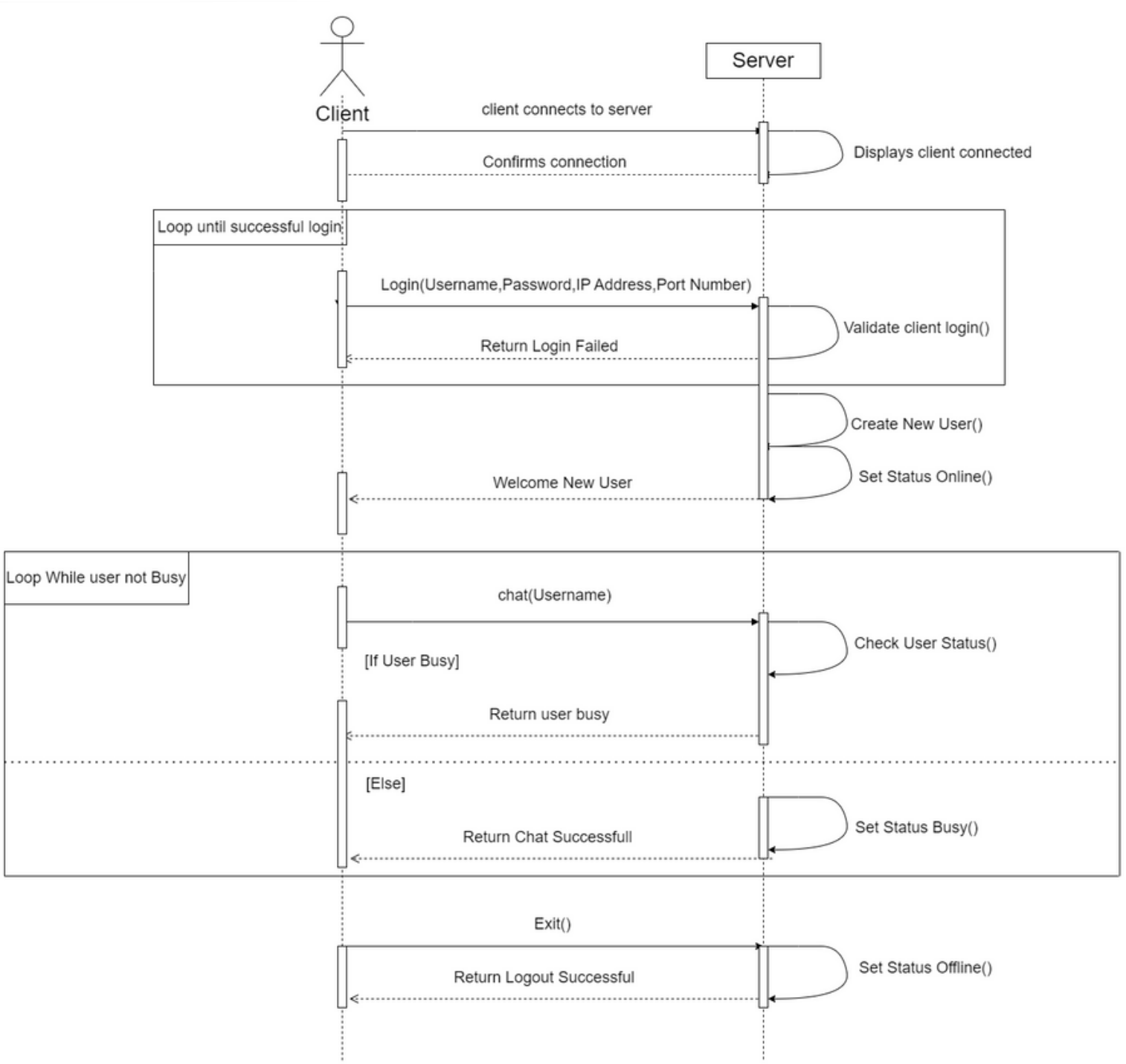
AVAILABLE	cr	lf	**peer IP**	cr	lf	**Peer UDP socket**	cr	lf

OR

BUSY/OFFLINE	cr	lf	cr	lf

SEQUENCE DIAGRAMS

SERVER-CLIENT:



CLIENT-CLIENT: