PSUEDOCODE

SERVER

CONNECT

```
Create stream socket
Set sock option to reuse address
Bind address to socket
Go to **Listen on Socket**
If listen_sd is set
    Index into client array
    Go to Listen on Socket
```

LISTEN ON SOCKET

```
Forever loop
    Call select
    If accepted client connection (listen_sd is set)
        Go to **Handle Connection**
    Else (sockfd is set)
        Go to **Handle Data**
```

HANDLE CONNECTION

```
Update list of clients and usernames
Send username list to all clients
Save client's descriptor to client array
Increase max index for client array
Go to **Handle Data**
```

HANDLE DATA

```
Check all client sockets
Read socket for data
While there is data to read
  Log message and sender
```

```
    If received username
        Add username to list
        Send list to all clients
    Else if received disconnect message
        Echo disconnect message back to client
        Remove from client list
        Send list to all clients
    Else regular message
        Echo message to all clients except sender
    If no more bytes read
        Close socket connection
        Clear descriptors and reset client array to empty
Go to Listen on Socket
```

CLIENT

INITIALIZE GUI

```
Set up GUI using QT Framework
Go to Get User Input for Username Port and IP State
```

GET USER INPUT FOR USERNAME PORT AND IP

```
Wait for input from user
Grab text from username port and IP text fields
Save these values into temporary strings
If user press connect button
        Go to Create Socket state
```

CREATE SOCKET

```
Create a stream socket with AF_INET, SOCK_STREAM, 0 parameters
        Check for any errors on socket call
Go to Bind Address state
```

BIND ADDRESS

```
Bind address to the socket
Allocate memory for server struct
Initialize server struct with AF_INET, port specified by user, and to accept connections from any client
Call bind(ListenSocket)
    Check for any errors on bind call
Go to Connect to Server state
```

CONNECT TO SERVER

```
Call connect()
If error
    Print message
Go to Send Username to Server
If disconnect button is pressed
    Close socket and update user list
```

SEND USERNAME TO SERVER

```
Create and join a pthread
Send username to server using send call
Go to Create Data Receiving Thread
```

CREATE DATA RECEIVING THREAD

```
Create a new thread
If error
    Print error message
Go to Wait for User Input
```

WAIT FOR USER INPUT

```
Forever loop
    If received <Enter> or Send Button
        Update chat window with local message
        Go to Transmit Data through Socket state
    If export button pressed
        Go to File Writing Process state
```

TRANSMIT DATA THROUGH SOCKET

```
Get text from user input
Add to buffer
```

```
Write buffer to socket
Go to Wait for User Input state
```

WAIT FOR INCOMING SERVER DATA

```
Call select()
Forever loop
    If received data is message
        Store data in buffer
        Go to Update Chat Window state
    If received data is username
        Store data in buffer
        Go to Update User Window state
```

UPDATE CHAT WINDOW

```
Get text in buffer
Output text to window
Go to Wait for Incoming Server Data state
```

UPDATE CHAT WINDOW

```
Get text in buffer
Add list item to user list widget
Go to Wait for Incoming Server Data state
```

WRITE TO FILE

```
Open File

Get text in buffer
Write text to opened file
Go to Wait for Incoming Server Data state
```