ENCRYPTED CHAT ROOM

COSC 4436 Fall 2022

Atul Kumar: 199636140

Emma Ude: 199697810

Jane Rebecca Kim: 219650890

Computer Science

Algoma University

**Encrypted Chat Room**

*COSC 4436 Term Project*

**I. Introduction**

Our group decided to create a multi-user encrypted chat room. This allows multiple clients to connect to a server and send and receive messages from one another. The messages are also encrypted as they are sent to and from the server for security purposes.

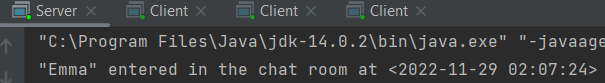
**II. Features of Application**

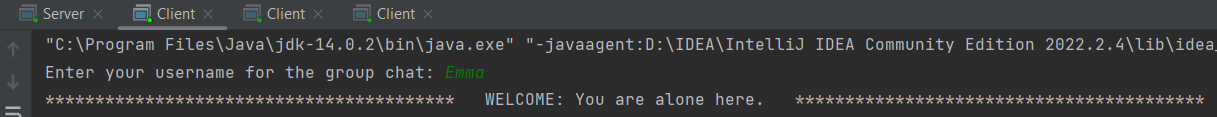
*Connecting Server and Client*

The primary feature of this application is that the various users connect to the server and are able to communicate with each other.

Process for connecting the first client:

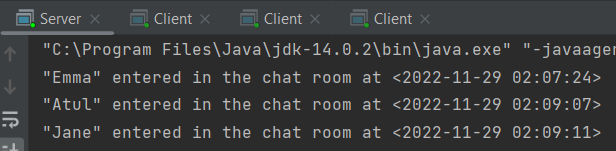
1. The server program is first run to ensure it is switched on.
2. Client one runs their program and is prompted to enter their name.
3. Client one is then connected to the server with the name they have chosen
4. Server accepts connection
   1. Server assigns the first client the name that they have chosen.
   2. Server prints the message “X has entered the chat at <date & time>”.
   3. Server then sends the message “WELCOME: You are alone here” to the chatroom

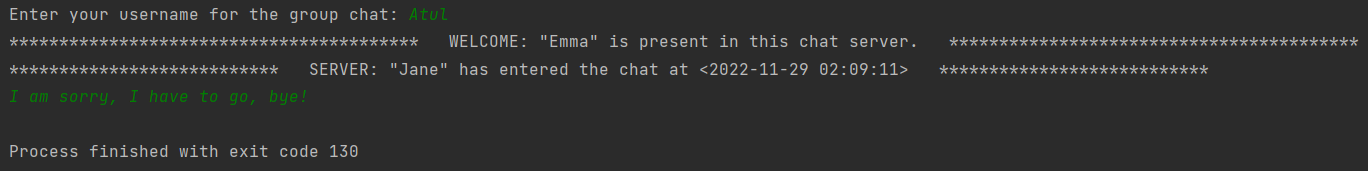




Process for connecting for other clients:

1. Other clients launch the program and are prompted to enter their name.
2. After entering their name, they are connected to the server.
3. Server accepts connection
   1. Server assigns the client the name that they have chosen.
   2. Server prints the message “X has entered the chat at <date & time>”.
   3. Server then sends the message “X has entered the chat” to the chatroom
   4. The Server also informs the new clients of the previous clients that joined the chat room before them.

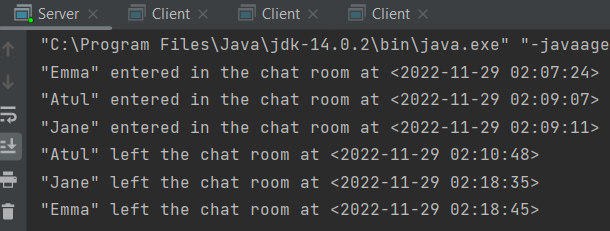




*Closing Connection*

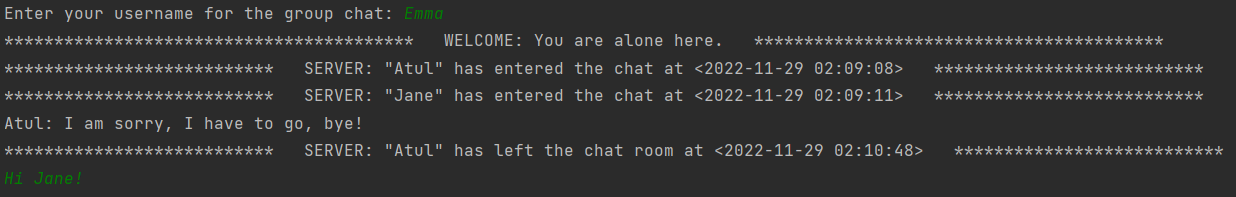
A client can exit the chat room by terminating the program. Once they terminate the program, the server sends a message “X has left the chat at <date & time>” to the chatroom. This way all other clients are aware that a client has left. It also records what time the client left the chat.





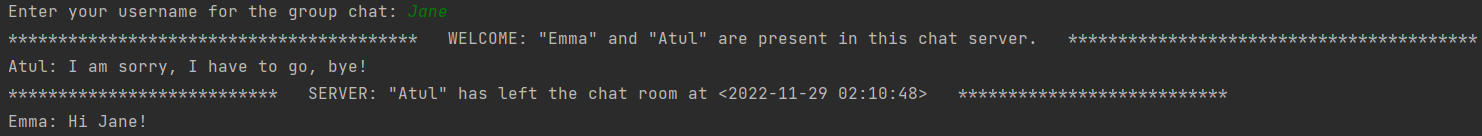
*Sending Messages*

To send a message the client has to type the message in the console and hit *Enter*. The message is then sent through the server to all other clients.



*Receiving a Message*

When a client sends a message, it is received by the other clients in the format “X: message”.



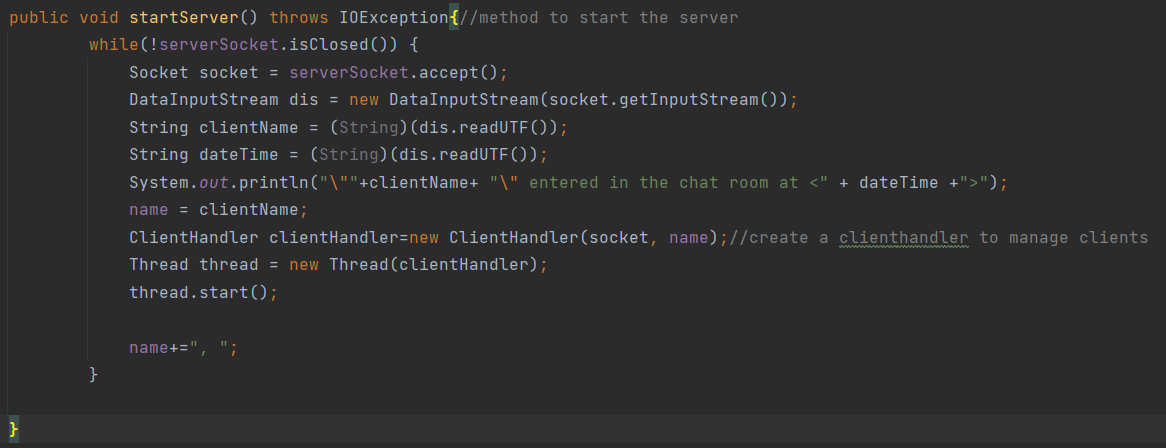
**III. Tools & Techniques**

There are three programs, the server, client and client handler. The server and client programs are written in java using the java.net package to establish a TCP (Transport Control Protocol) connection. The client handler program is used to manage how many clients are currently connected to the server. Communication between the client and server sockets is done with bytes using BufferedReader and BufferedWriter. The strings are encrypted before they are sent and decrypted when they are received. The clients all connect to socket 1234.

*Server*

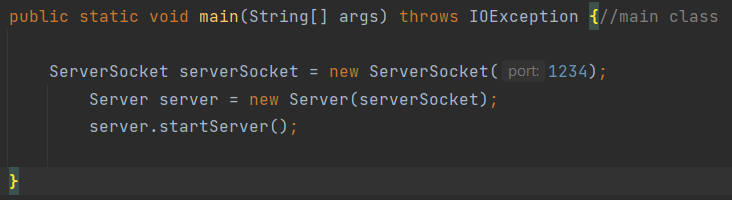
The program creates a socket the java.net ServerSocket and Socket classes. This program is in charge of starting and closing the socket.

The “startServer()” method creates a new socket with the “accept()” method for the ServerSocket. It then creates a ClientHandler and uses a Thread to connect the Server and ClientHandler.



The “closeServer()” method is used to close the ServerSocket.

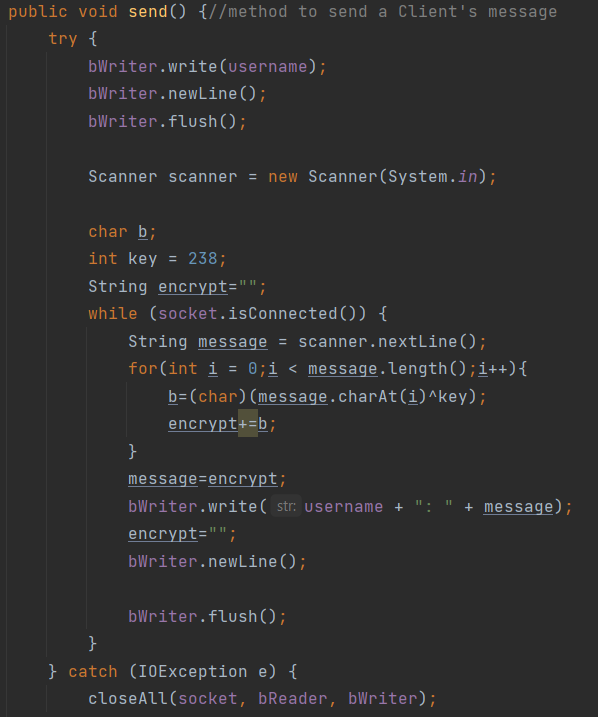
The main method initiates a serverSocket to socket 1234 when the Server program is run. It then sets the ServerSocket to the Server object created and starts the socket using the “startServer()” method.



*Client*

This program connects a client to a user with a particular username.

The “send()” method reads the user’s input from the console. The string is then encrypted before it is sent using the BufferedWriter.

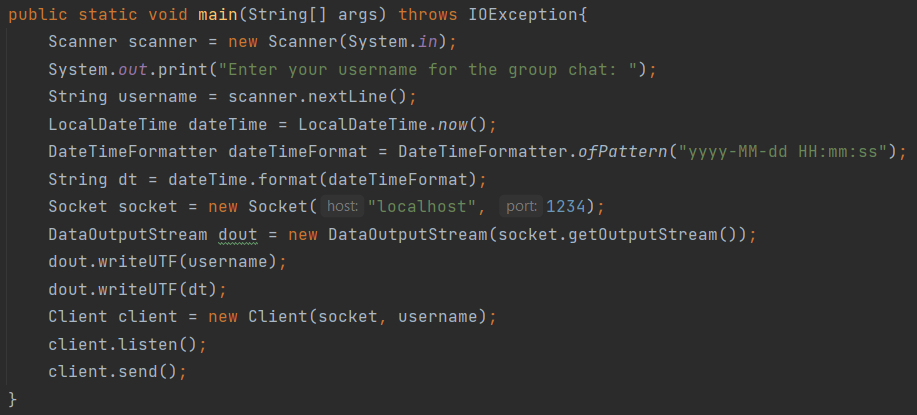


The “listen()” method uses the BufferedReader to read the encrypted message that is sent by the other clients, it is then decrypted before being displayed in the chatroom.





The main method then creates a Client and assigns it the socket 1234.

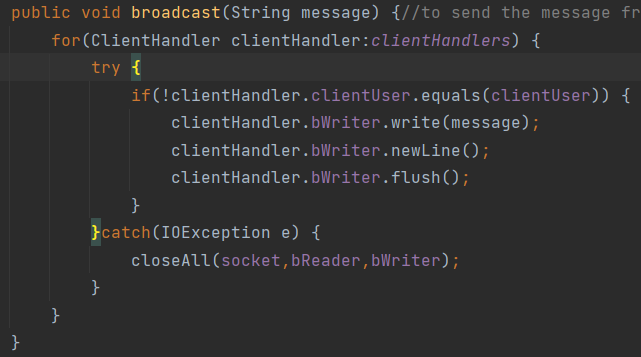


*Client Handler*

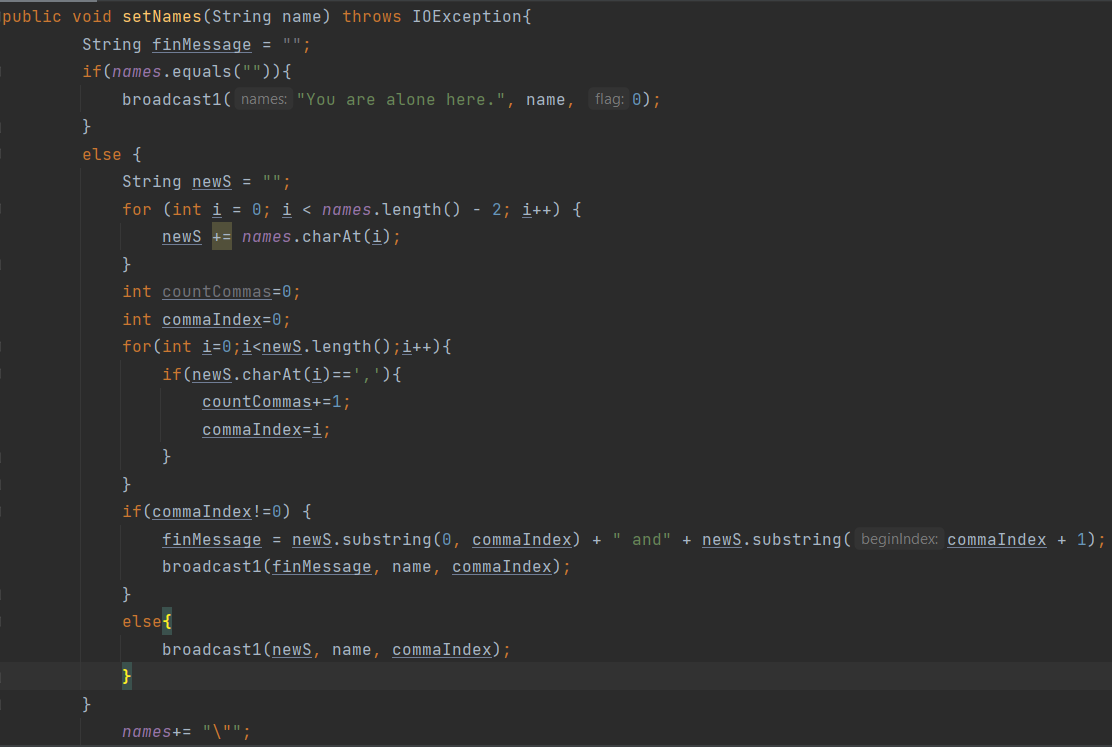
The program manages how many Client objects are connected to the Server and informs other users when a user joins or leaves.

The “broadcast()” method sends “SERVER” messages to all the clients at once. Usually these messages are about whether a new client has joined the chatroom. Or if a client has left the chatroom.

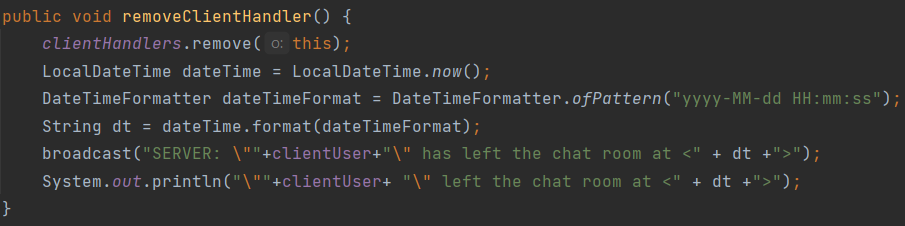




The “setNames()” is used to set the names for the clients.



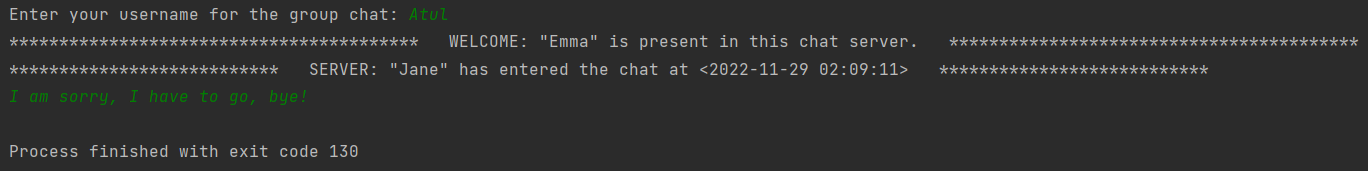
The “removeClientHandler()” removes a client from the list and tells the other users that the user has left the chatroom.

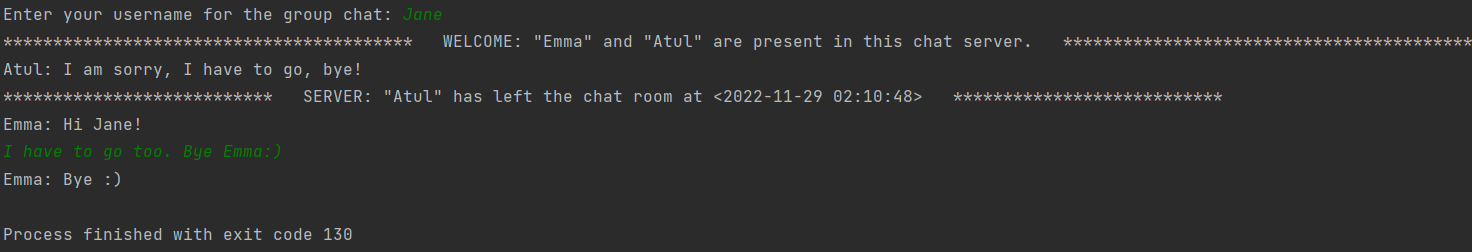


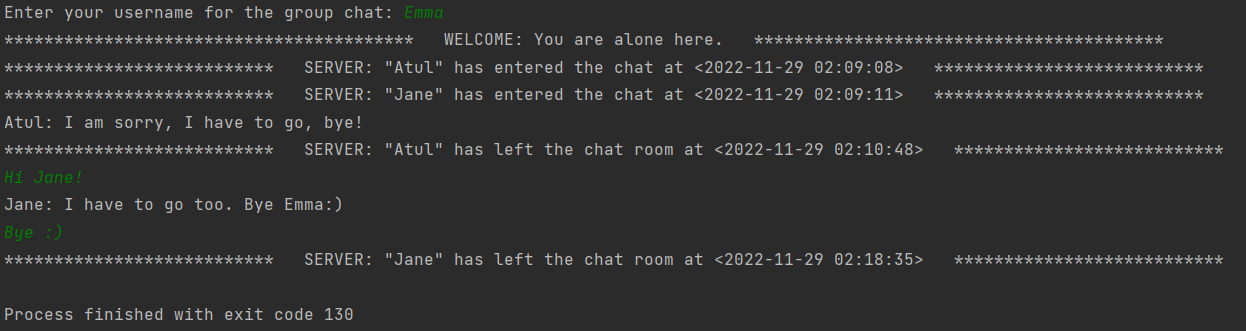
**IV. Implementation Details**

To use the chatroom, The Server is launched. Then as many Client programs that are needed can be launched.

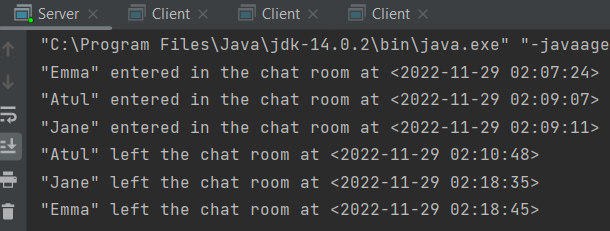
Next, each client has to enter their username. They are then connected to the Server and can begin chatting, after the first Client connects, all other client’s connections are broadcasted in the chatroom.







**Server snippet (at the end of the session):**



**V. Conclusion**

Encrypted Chat Room is a basic network application. The tools such as the java.net package cover the essential framework for many powerful network applications.

Future development of this Encrypted Chat Room may include a graphical interface, a chat database that records the usernames and all the messages sent (in their encrypted form) to keep a log of old messages.