

**Individual Portfolio Assignment 1**

Rateb Alzerkli

S351925

DATA2410

## Table of Contents

Introduction.....	3
Bot explanation .....	3
Client explanation .....	3
Server explanation .....	4
Code readability .....	5
Code reliability and inline comments .....	6
Conclusion .....	7

## Introduction

The project is simple chat server with functions that takes a string as input from clients and returns other string as respond to the input. The chat server has four bad chat bots on it, and they have responsibility to respond and extract input from each client. The chat bots are functions that takes a string as parameter extracted by client input and return a statement responding to the action. This project consist of two TCP programs, client and server which connect to each other and make a simple chat

## Bot explanation

Bots are functions that takes a string in and return a string. Once a client writes one of the suggestions provided by the server, bots respond and extract the word then return a statement that include the word. Each bot responds to the client differently and all responses comes at the same time and sends to all connected clients

## Client explanation

Client is a TCP program that takes 3 parameters IP, port and username. Client enters his username and connect to the server. Once the client enters the username and connect to the server, a round of suggestions sent by the server appears on the terminal. The client writes one of the suggestions and immediately get a respond by chat bots reacting to the clients input. We start

with creating a socket that's takes port as parameter and then try to connect to the server on the same port. Create two methods for sending and receiving messages, method send message scans the client input and send the message to the terminal and other clients without the need of spawning a thread. While listen for message method needs a separate thread to receive messages without blocking, it gets the message sent by the client and then print it to the console. Merk that each client needs a thread handling client to receive messages and then send it to the client

### Server explanation

Server is a TCP program that takes a single parameter Port. Server accepts all connections and makes sure that clients are connected before sending a round of suggestions. Once a client is connected a message appears on the server console so that server maintain a list of connected clients and their usernames. Server notify all connected clients when a new client connects or when one of the clients ends the communication. When a connected clients quite or end conversation, server send them a message and then remove them from the thread handlers. The server takes input from command line and then send all responses by bots to all clients but checks that's they are still connected first.

## Code readability

Code takes client input as a message; extract the suggested action from the line, send bots responses to the client, send an error message to client if input does not contain a specific word mentioned in the string, send a goodbye message with the client username if client wants to quite

```

37
38  @Override
39  public void run() {
40      while (socket.isConnected()){
41          try {
42              String message = in.readLine();           //read what th client send
43              sendMessage(message);                     //send the message to other
44              String [] matches = new String[] {"eat", "fight", "sleep", "sing", "play", "cry", "complain", "yell", "work"};
45              if (message.contains("bye")){             //if client message include word bye
46                  sendMessage( "ok bye " + username);  //remove client og send melding
47                  removeclient();
48                  //i could make a string with bad words, so that if client misbehave then remove him
49                  //inside the if setning add some bade words and then call function removeclient!
50              }
51              else if (!(message.contains("eat") || message.contains("fight") || message.contains("sleep") || message.contains("sing")
52                  || message.contains("play") || message.contains("cry") || message.contains("complain") || message.contains("yell")
53                  || message.contains("work"))))
54              {
55                  sendMessage( "i dont understand, please choose one of the suggestion above ");
56              }
57          }
58          for (String s : matches) {
59              //make a string that includes the words you want your bot to interact
60              if (message.contains(s)) {                //if the message from client includes one of this words
61                  sendMessage( "Alice: " + Alice(s) + "Bob: " + Bob(s) + // bot answers
62                      "Dora: " + Dora(s) + "Chunk: " + Chunk(s));
63              }
64          }
65      }
66  }

```

## Code reliability and inline comments

```

126
127
128
129 //define my 4 bots and let them extract the client word and respond to it
130 @ public static String Alice(String alice) throws IOException { //a Bot function that takes a string from client input
131     return "i think " + alice + "ing sounds great \n"; //return a string with the extracted word from client line
132 }
133
134 @ public static String Bob(String bob) throws IOException { //a Bot function that takes a string from client input
135     String [] bobactions = new String[] {"playing", "singing", "hugging", "working"}; //make a string with different actions
136     String bobaction = bobactions [ (int) (Math.random()*bobactions.length)]; //make a method to provide random actions
137     return bob + "ing sounds OK!, but i was thinking maybe we could do some " + bobaction + "\n" ; //return a string with the extracted word from client line
138     //and suggest something else instead
139 }
140
141 @ public static String Dora(String dora) throws IOException { //a Bot function that takes a string from client input
142     return "No i dont like " + dora + "ing ,can we do something else? \n"; //return a string with the extracted word from client line
143 }
144
145 @ public static String Chunk(String chunk) throws IOException { //a Bot function that takes a string from client input
146     return " yes time for " + chunk + "ing " ; //return a string with the extracted word from client line
147 }
148 }
149

```

```

    String [] matches = new String[] {"eat", "fight", "sleep", "sing", "play", "cry", "complain", "yell", "work"}; //string with words to extract
    for (String s : matches) { //make a string that includes the words you want your bot to extract
        if (message.contains(s)) { //if the message from client includes one of this words
            sendmessage( messagesend: "Alice: " + Alice(s) + "Bob: " + Bob(s) + // bot answers
                "Dora: " + Dora(s) + "Chunk: " + Chunk(s));
        }
    }
}

```

## Conclusion

A small project called chat bot with two TCP programs that connect to each other on a defined port. Many clients can connect to the same server in parallel and get round of suggestion each. The client should write down one of the suggestions provided by the server and the server respond to this suggestion through couple of bad bots which are formatted to respond to a action extracted by client input. Before a client is connected, server asks for his/her name and then accept connection before sending any information. The client can send multiple inputs and get respond or they can simply write bye to exit and end connection.

## References

- 1) W3school (2022). Java Tutorial <https://www.w3schools.com/java/default.asp>
- 2) YouTube David Dobervich (2019). Java Socket Programming  
link: [https://www.youtube.com/watch?v=BWjGQIlkgT4&t=1s&ab\\_channel=DavidDobervich](https://www.youtube.com/watch?v=BWjGQIlkgT4&t=1s&ab_channel=DavidDobervich)
- 3) GitHub Marcus Bugge (2021). portfolio exam1  
link: <https://github.com/marcusbugge/DATA2410-1-21V-Datanettverk-og-skytjenester---PortofolioExam1/tree/main/portofolio/src>
- 4) Java socket (2022). Class socket  
link: <https://docs.oracle.com/javase/6/docs/api/java/net/Socket.html#setSoTimeout%28int%29>