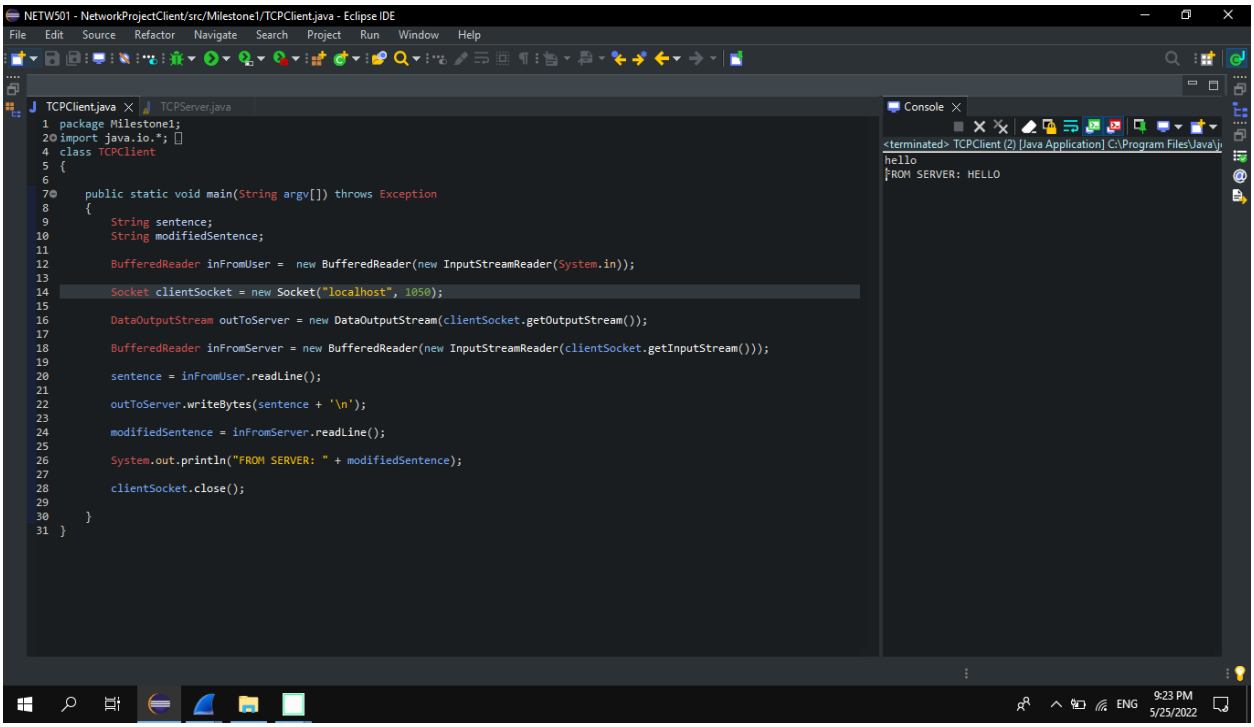
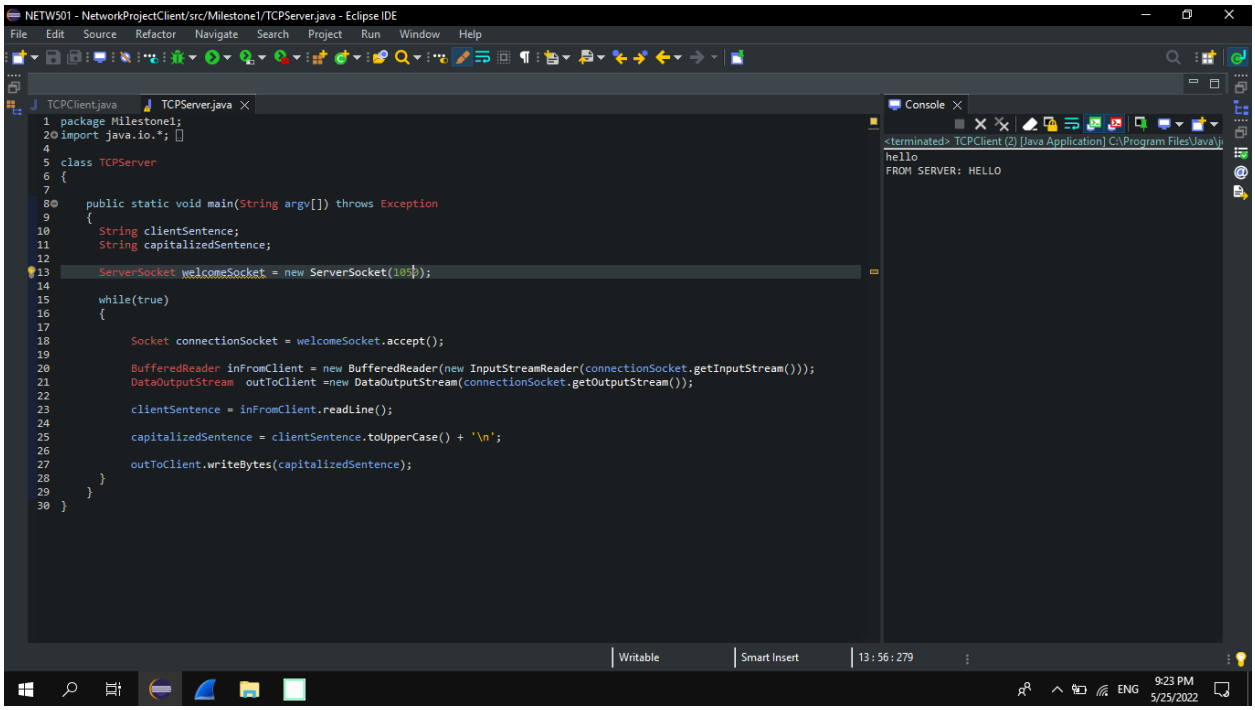
Project report

In this project we aimed to introduce a bi- direction chatting application using the “java socket programming” with imported libraries

MILESTONE1

1. We created a java project and implemented two packages in it
2. The first package concerning our first milestone which goes by the following steps
3. We created two Class (TCPClient , TCPServer) and pasted the code found In the slide but replacing the word “hostname” with “local host” as we are using the same PC.
4. We used a distinctive port number for each run and this is the code for the the client class and the Server respectively





MILESTONE2

1. In this milestone we tend to establish the real chatting application by editing our code as follows:
2. Firstly, the client can only establish the connection by typing the word “CONNECT” in the console of the client, otherwise no connection will be established so I’m using a loop to fulfill this case.
3. Secondly, once a connection is established the client send the first message to the server in one line and then presses enter , so the statement will apparat immediately in the console of the server
4. The Client and the server will keep sending the message alternatively each message per turn for only one party of the server and client.
5. The Chat will keep going till one of the party enters the word ”stop”, after that the socket will be closed from both sider and the connection will be terminated i.e. no one of the party will send any thing any more or an exception will be thrown.

The code for the client and the server as shown down here respectively

NETW501 - NetworkProjectClient/src/Milestone2/TCPClient.java - Eclipse IDE

TCPClient.java

```java
package Milestone2;
import java.io.*;
class TCPClient
{
    public static void main(String argv[]) throws Exception
    {
        while (true) // to handle the condition of starting the connection only when typing the word "CONNECT"
        {
            Scanner inFromUser1 = new Scanner(System.in);
            String firstCheck = inFromUser1.nextLine();
            if (firstCheck.equals("CONNECT"))
            {
                System.out.println("succesfully connected" + '\n') ;
                break;
            }
            else
                System.out.println("couldn't connect , please try again" + '\n');
        }

        while (true)// The real application chat code
        {
            //The first part when the client sends a message to the server
            String toServerSentence;
            String fromServerSentence;

            BufferedReader inFromUser = new BufferedReader(new InputStreamReader(System.in));
            Socket clientSocket = new Socket("192.168.1.51", 5550);
            DataOutputStream outToServer = new DataOutputStream(clientSocket.getOutputStream());
            BufferedReader inFromServer = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));
            //The first part responsible for sending messages from client to server only and displaying it on the console
            toServerSentence = inFromUser.readLine();
            //System.out.println("CLIENT: " + toServerSentence);
            if (toServerSentence.equals("stop")) //A condition to terminate the connection by the client
            {
                clientSocket.close();
                System.out.println("connection terminated, you can't send anymore");
                break;
            }
```

Console

```
<terminated> TCPClient (1) [Java Application] C:\Program Files\Java\j
CONNECT
succesfully connected

hello my friend , hope you are doing well these days
SERVER: yeah,Iam fine , how about you buddy?
doing well , what are you doing this evening
SERVER: nothing,iam free...wanna hang out?
yeah sure, do you want to go the cinema to watch a m
SERVER: if course, see you at 7
fine , i will be at yours then :D
SERVER: fine see you soon
bye
SERVER: bye
stop
connection terminated, you can't send anymore
```

eclipse-workspace - CommNetworkServer/src/TCPServer.java - Eclipse IDE

Package Explorer: CommNetworkServer, JRE System Library [JavaSE-16], src, (default package), TCPServer.java, Marvel-M2

Game.java | Champion.java | Cover.java | Hero.java | Villain.java | AntiHero.java | Quiz2V2.java

```java
import java.io.*;
import java.net.*;

class TCPServer
{

    public static void main(String argv[]) throws Exception
    {



        ServerSocket welcomeSocket = new ServerSocket(5550);
        while (true)
        {
            String fromclientSentence;
            String toClientSentence;


            BufferedReader inFromUser = new BufferedReader(new InputStreamReader(System.in));
            Socket connectionSocket = welcomeSocket.accept();
            BufferedReader inFromClient = new BufferedReader(new InputStreamReader(connectionSocket.getInputStream()));
            DataOutputStream  outToClient =new DataOutputStream(connectionSocket.getOutputStream());

            //The first part responsible for receiving messages from client to server and displaying it on the console
            fromclientSentence = inFromClient.readLine();
            System.out.println("CLIENT :" + fromclientSentence);

            //The second part responsible for sending messages to client and displaying it on the console
            toClientSentence = inFromUser.readLine();
            if (toClientSentence.equals("stop"))// A condition to terminate the connection by the server
            {
                welcomeSocket.close();
                System.out.println("connection terminated by the server , you can't send anymore");
            }
            outToClient.writeBytes(toClientSentence+ '\n');
```

Problems | Javadoc | Declaration | Console

```
TCPServer [Java Application] C:\Program Files\Java\jdk-16\bin\javaw.exe
CLIENT :hello my friend , hope you are doing well these days
yeah,Iam fine , how about you buddy?
CLIENT :doing well , what are you doing this evening
nothing,iam free...wanna hang out?
CLIENT :yeah sure, do you want to go the cinema to watch a movie
if course, see you at 7
CLIENT :fine , i will be at yours then :D
fine see you soon
CLIENT :bye
bye
CLIENT :null
```

MILESTONE3

- In this milestone we use the wireshark sniffing application to trace the packet we sent.
  - We chose wifi then set ip.addr = our ip address
  - We trace any packet we find that is using the TCP layering