Green University of Bangladesh

Department of Computer Science and Engineering

# Project Report

**Course ID : ** CSE_308

**Course Title:** Data Communication Lab

**Project Name:** "***SNS msg***" chatting application.

**Date of Submission : ** 06.01.2021

| Submitted by | Submitted to |
|---|---|
| **SHOVON SIKDER** :183002045<br>**NAZMUL HASAN** : 183002159<br>Section: 183-DB<br>Dept. of CSE<br>Green University of Bangladesh | **MD. ATIKUZZAMAN**<br>Lecturer<br>Dept. of CSE<br>Green University of Bangladesh |

| Signature & Remarks |
|---|
| |

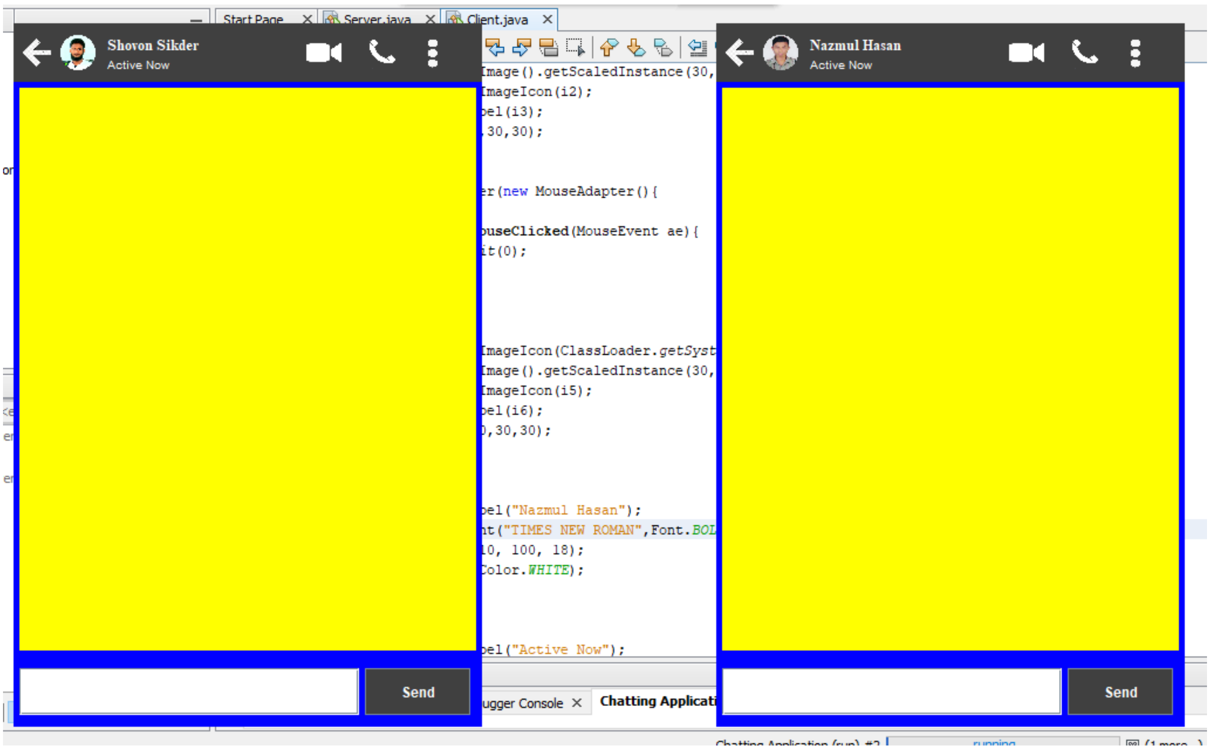
# A project on
# “*SNS msg* chatting Application”

-----------------------------------------------------Developed by Shovon Sikder & Nazmul Hasan.

# Introduction

SNS msg is a simple chatting application where you can send text message easily. The application can sent message in half-duplex mode. It's extremely easy to use, you need just one click to send. In this digital era everyone wants to communicate quite easily and quickly. Therefore, SNS msg application will provide these facilities.

We have developed the application by using java and socket programming. In the upcoming chapter we will look at the design methods and procedures.



```
Image().getScaledInstance(30,
ImageIcon(i2);
pel(i3);
,30,30);

er(new MouseAdapter(){

ouseClicked(MouseEvent ae){
it(0);



ImageIcon(ClassLoader.getSyst
Image().getScaledInstance(30,
ImageIcon(i5);
pel(i6);
),30,30);


pel("Nazmul Hasan");
nt("TIMES NEW ROMAN",Font.BOL
10, 100, 18);
Color.WHITE);


pel("Active Now");
```

Fig 1: Compete view of the Application

# Overview

The application connects a client with a server to exchange data between them. By using socket programming the server provide a port address and hold for connecting the client. On the other hand, client connects to the port of the server with an IP address. Now they are ready to communicate with each other.

The text box helps to take input from user and send button sends the inputted data. When user tap on send button, the message written to the port by using writeUTF(). Another function readUTF() used to read the message from the port. A text area is used to display all the message. The source messages are displayed in the right part of the text area and received messages in the left. Both the server and client application uses the same technique.

# Design Methods

The whole application has designed by java Graphical User Interface (GUI). The connection has designed by socket programming. Let's explore the design in details.

The design contains a JPanel, a JTextArea, a JTextField and a JButton. JPanel shows profile picture, name, active status and additional option. Below to the JPanel there is a JTextArea that's display all messages. Next to the JTextArea, a JTextField has placed which helps user to write text message. The JButton sends the message to a destination.

The design of the network side is done by socket programming. The server provides a port address to connect the client.

ServerSocket class is used to generate port address and Socket class is to connect, read and write a message. DataInputStream ,DataOutputStream class read, write data from a port.

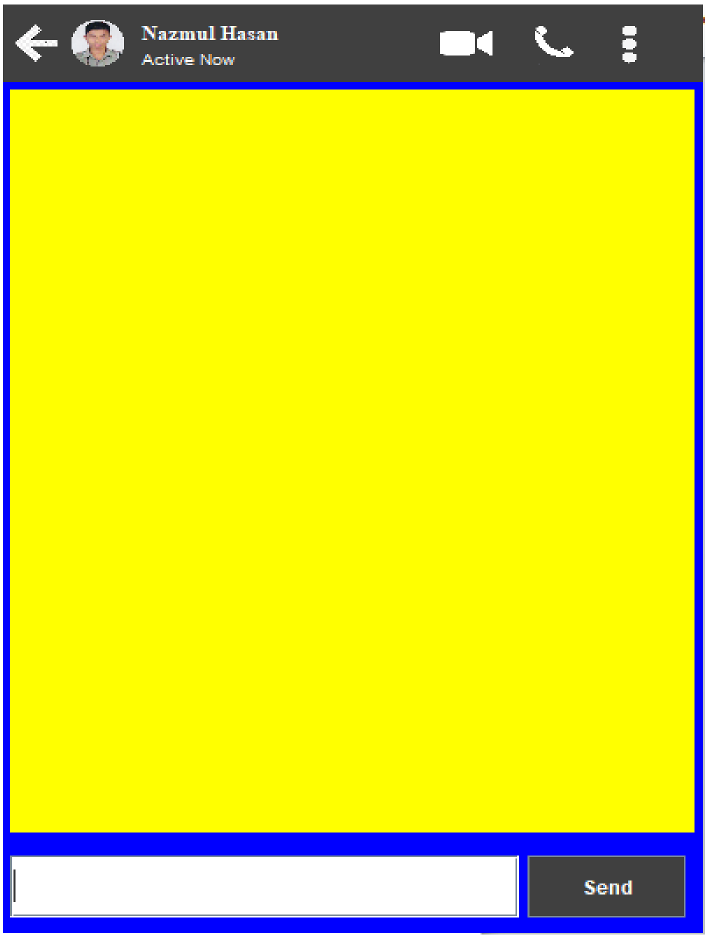


Fig 2: Design View

# Procedures

Firstly, the server and client site is designed by java Graphical User Interface (GUI). In this section, we create a JFrame using java swing then we use a JPanel to create the header. Inside header section, we set a back button in the upper left corner. If user click this button a mouseListener event will activate which exit the whole system.

Next to the exit button, we set an image of the user. This process is done by using image icon. Then there appears the user name and active status. Which is shown by using JPanel. There are three icons for audio, video and options respectively.

The textArea is built by using JTextArea class. The bounds of the textArea is (5, 55, 390, 480). The last part of design is done by using JTextField and JButton.

Secondly, the data transmission process is done by using socket programming. We need to create ServerSocket object and Socket object for Server and Socket object for Client.

DataInputStream receive the upcoming data and DataOutputStream end the data by UTF.

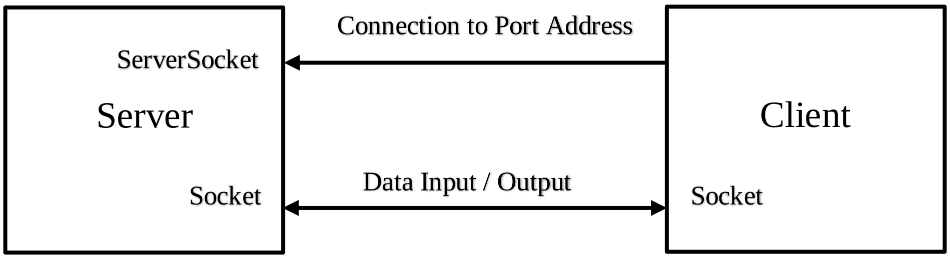
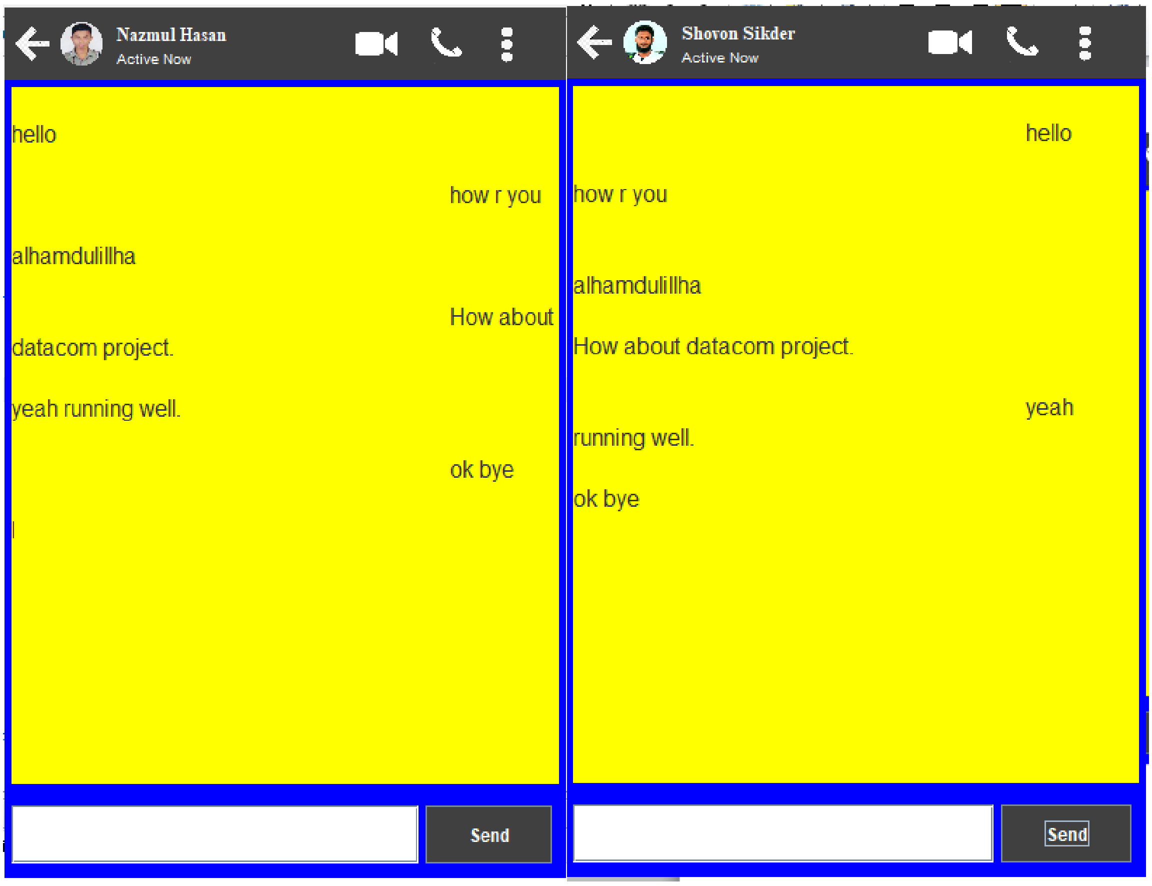


Fig 3: Working Procedure

# Result Analysis

The application compiled and executed successfully in IDE. First we have to turn on the server and then the client. Server and client transmitted message properly. All message display in TextArea which may not well decorated. Therefore sometimes message decoration create an issue if the message length is too long. The demo project shows only two side one server and one client transmission. Everything worked properly as expected except some tiny GUI design issues. We're working on it to solve those issues. After all, the result was satisfying.



hello

how r you

alhamdulillha

How about

datacom project.

yeah running well.

ok bye

hello

how r you

alhamdulillha

How about datacom project.

yeah

running well.

ok bye

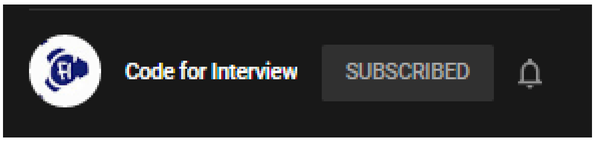
Fig 4: Result of the project

# Conclusion

This is a simple chatting application. It has some draw backs but in the end it works properly. In near future, we will work on it to develop a better version for the users. Upcoming design will be very user friendly and well decorated. We’re working to develop group messaging feature where multiple user can send message simultaneously. To keep users trust, we will consider a highly encryption technique for keep the data safe from unauthorized users.

# Reference

Thanks to



for such a creative project.

https://www.youtube.com/watch?v=0JgfnMdNt5Q&list=PL5BFcXE899zyMaBM7o3MPwd2d5qv1l4AR&index=1

END