SECURE CHAT ROOM SERVER

MINOR PROJECT REPORT

By

Sanyog Dani [Reg No.: RA2211031010087]
Arush Sirotiya [Reg No.: RA2211031010092]

Under the guidance of

Dr. M. Manickam

In partial fulfilment for the Course

of

21CSC203P – ADVANCED PROGRAMMING PRACTICE

in Department of Networking and Communications

FACULTY OF ENGINEERING AND TECHNOLOGY

SCHOOL OF COMPUTING

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR

NOVEMBER 2023

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this minor project report for the course **21CSC203P ADVANCED PROGRAMMING PRACTICE** entitled in "**Secure Chat Room server**" is the bonafide work of **Sanyog Dani (RA2211031010087)** and **Arush Sirotiya (RA2211031010092)** who carried out the work under my supervision.

Dr. M. Manickam
Assistant Professor
Department of Networking and Communications, School of Computing
SRM institute of science and technology
Kattankulathur campus Chennai

Dr. Annapurani Panaiyappan .K
Head of Department
Department of Networking and Communications, School of Computing
SRM institute of science and technology
Kattankulathur campus Chennai

Signature of the Internal Examiner **Signature of the External Examiner**

ABSTRACT

Our project is dedicated to breaking down geographical barriers and fostering communication through the use of technology. It comprises two vital components: a client application that operates within a user's web browser and a server application hosted on a network server. To initiate chat sessions, users establish connections with the server, enabling both private and group conversations. The driving force behind our research is the recent surge in smartphone-based instant messaging applications, which, despite their popularity, often grapple with security vulnerabilities and privacy issues. Our primary research objective is to identify the requisite security features for an instant messaging application and devise a comprehensive system that enhances data integrity, confidentiality, and privacy. At the heart of our approach lies the concept of End-to-End Encryption (E2EE), which fortifies the protection of user data. To achieve our goals, we scrutinize current security features in leading mobile messaging apps, distilling a set of security requirements, which, in turn, inform the architectural design of our secure messaging application. The ultimate validation comes through the implementation of a demo that is subjected to rigorous evaluation, ensuring it fulfills the defined security prerequisites while offering a robust level of privacy and data security.

ACKNOWLEDGEMENT

We express our heartfelt thanks to our honorable **Vice Chancellor Dr. C. MUTHAMIZHCHELVAN**, for being the beacon in all our endeavors.

We would like to express my warmth of gratitude to our **Registrar Dr. S. Ponnusamy,** for his encouragement.

We express our profound gratitude to our **Dean (College of Engineering and Technology) Dr. T. V.Gopal,** for bringing out novelty in all executions.

We would like to express my heartfelt thanks to Chairperson, School of Computing **Dr. Revathi Venkataraman,** for imparting confidence to complete my course project

We are highly thankful to our my Course project Faculty **Dr. M. Manickam (Assistant Professor , Department of Networking and Communications)** for his/her assistance, timely suggestion and guidance throughout the duration of this course project.

We extend my gratitude to our **HOD Dr. Annapurani Panaiyappan .K** (**Networking and Communication**) and my Departmental colleagues for their Support.

Finally, we thank our parents and friends near and dear ones who directly and indirectly contributed to the successful completion of our project.

TABLE OF CONTENTS

CHAPTER NO	CONTENTS	PAGE NO
1	**INTRODUCTION**	**5**
	1.1 Motivation	**5**
	1.2 Objective	**6**
	1.3 Problem Statement	**6**
	1.4 Challenges	**7**
2	**LITERATURE SURVEY**	**8**
3	**REQUIREMENT ANALYSIS**	**9**
4	**ARCHITECTURE & DESIGN**	**10**
5	**IMPLEMENTATION**	**11**
6	**EXPERIMENT RESULTS & ANALYSIS**	**17**
7	**CONCLUSION**	**18**
8	**REFERENCES**	**19**

1. INTRODUCTION

1.1 Motivation

Messaging apps serve as powerful tools to bridge geographical divides and facilitate instant communication, effectively shrinking the world. Whether it's staying connected with loved ones living far away or collaborating seamlessly with colleagues in different time zones, these applications play a vital role in fostering connectivity. Furthermore, chat apps provide an empowering platform for individuals to express their ideas, share experiences, and engage in dynamic discussions. These apps have evolved to create vibrant communities where people come together to discuss their interests and provide mutual support. In the ever-evolving landscape of communication technology, chat apps continually innovate, introducing new features and enhancing functionality. This constant evolution means there's always room for inventive and pioneering chat applications to enter the market, redefining how people interact. Beyond connecting people and enabling self-expression, chat apps also present promising business opportunities. The global chat app market is on track to reach a substantial $1.6 trillion by 2026, making the development of a successful chat app a potentially lucrative business endeavor.

1.2 OBJECTIVE

The software offers an intuitive Graphical User Interface (GUI), making it accessible to users with minimal system operation knowledge. It is designed for cross-platform compatibility, functioning seamlessly on various operating systems, thus ensuring its usability on any system, regardless of the underlying OS. Remarkably, the messenger supports an unlimited number of concurrent users without compromising the server's performance. The primary objective of this project is to develop an instant messaging solution that facilitates seamless communication between users. Notably, the project places a strong emphasis on user-friendliness, ensuring that even those with limited technical experience can easily use the application.

1.3 Problem Statement

This endeavor aims to construct a chat application encompassing a server and user interfaces, facilitating interactive conversations among its users. The central objective is the development of a real-time messaging solution, designed to foster seamless communication between users, bridging gaps and enhancing collaboration. A primary focus lies in ensuring the project's accessibility, making it user-friendly to the extent that even those with minimal technical experience can navigate and employ it effortlessly. Importantly, this undertaking holds the potential to serve as a valuable asset within organizational settings, offering a means for employees to connect via Local Area Network (LAN), thus promoting efficient communication and teamwork.

1.4 Challenges

The issue of scalability arises as the user base of a chat application expands. Scaling the system to handle the growing demand can be a complex task, potentially resulting in performance issues such as sluggish load times and message delivery problems.Security stands as a paramount concern in the realm of chat applications, necessitating robust measures to safeguard user data from unauthorized access. This challenge becomes more pronounced as chat applications evolve, introducing advanced features like file sharing and video calls.Achieving real-time communication in chat applications is another noteworthy challenge, particularly when dealing with varying network conditions and spanning considerable distances. Maintaining seamless and instantaneous interactions can be intricate.The user experience is a critical aspect of chat applications, demanding an interface that is both user-friendly and easy to navigate. This becomes increasingly challenging as chat applications incorporate more and more features, potentially complicating the user interface.In a competitive landscape with numerous established chat applications, breaking through and gaining recognition presents a formidable challenge for new entrants to the market.

2. LITERATURE SURVEY

1. A considerable research effort has been devoted to Chatting Application in the last few years. Many new researches have been proposed in recent times. This paper proposed a mechanism for creating professional chat application that will not permit the user to send inappropriate or improper messages to the participants by using natural language processing. The message typed by user is evaluated to find any inappropriate terms in it that may include foul/negative words using NLP. If the context of the message is negative, then the user not permitted to send the message .

2. In this paper, a model for developing a secure chatting application was proposed. In proposed model End to End Encryption was achieved by involving ECDH key exchange algorithm to provide the key pair between the two parties to generate the secure shared key that will be used as a key for the encryption. The algorithm used for encrypting text messages was AES standard which is slower than other block cipher. The algorithm used for encrypting voice and image messages was the RC4 which is suitable for the mobile device when encrypting vast amounts of data .

3. This paper introduced the working and implementation of encrypted chat application. It describes how the messages are encrypted on device before moving to server using various cryptographic algorithms. Computer vision techniques can be used for encrypting images. The performance of the application over hundreds of users is not tested yet . Performance testing and image encrytion are in the list of future tasks .

3. REQUIREMENTS

The hardware and software requirements for making a chat applications are :-

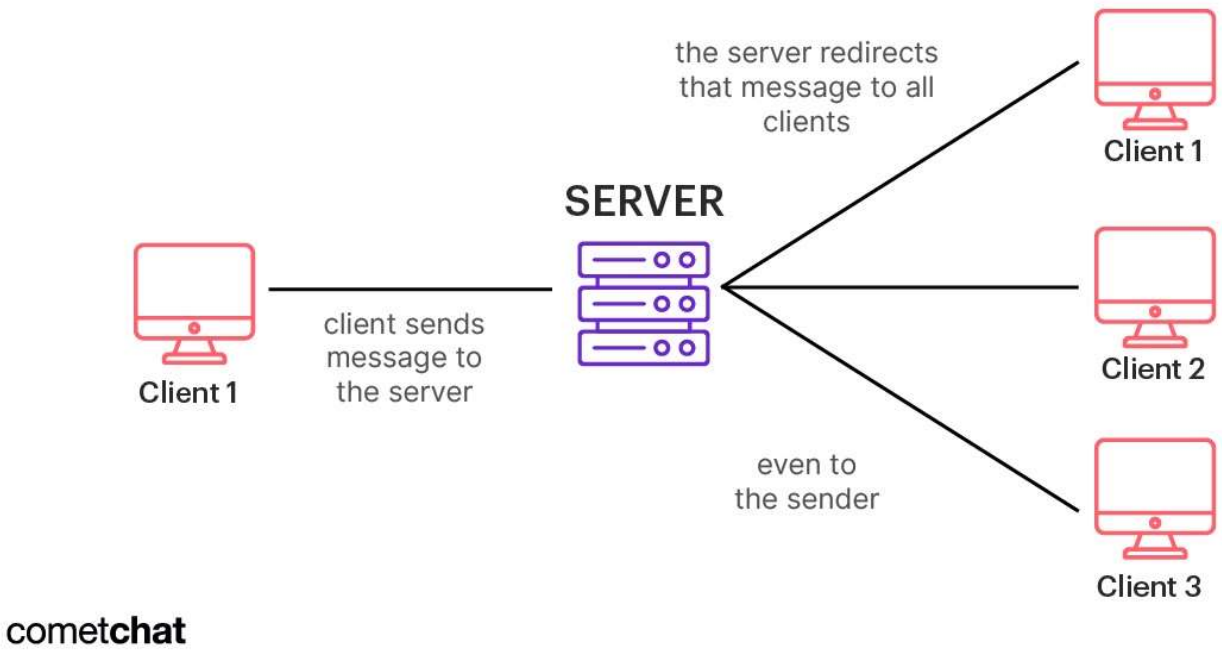
Hardware

To support your chat application, a server with a minimum of 256 megabytes of RAM and 300 megabytes of disk storage is essential. Additionally, for managing chat messages and user data, you'll require a dedicated database server equipped with ample disk space. Ensuring a robust network connection is vital to enable users to seamlessly connect to your chat application. A high-speed network connection is indispensable for maintaining smooth and uninterrupted communication among users.

Software

We will require a programming language, options include Python, Java, JavaScript, or Go. This language will serve as the foundation for your chat application's core functionality.If you're developing a web-based chat application, you may consider using a web framework for enhanced efficiency. Choices include Django, Flask, or Express.js, which can streamline web application development and user interactions. To manage data effectively, you'll need a database system. Options include PostgreSQL, MySQL, or MongoDB, each offering unique features for storing and retrieving chat messages and user information.In terms of real-time communication, a messaging protocol is a key component. WebSocket, XMPP, or MQTT can facilitate instant message delivery and user interactions within your chat application. On the server side, a robust operating system is pivotal. Opt for a server-grade operating system, like Linux or macOS, to provide a stable environment for hosting your chat application and its associated services.

4. ARCHITECTURE AND DESIGN



5. IMPLEMENTATION

6.1 Server File

```
package chatui;

import java.io.*;
import java.util.*;
import java.text.*;
import java.net.*;
import javax.swing.*;
import java.awt.*;
import javax.swing.border.*;
import java.awt.event.*;

public class Server implements ActionListener {

    JTextField text;
    JPanel a1;
    static Box vertical = Box.createVerticalBox();
    static JFrame f = new JFrame();
    static DataOutputStream dout;

    public void actionPerformed(ActionEvent ae) {
        try {
            String out = text.getText();
            JPanel p2 = formatLabel(out);
            a1.setLayout(new BorderLayout());
            JPanel right = new JPanel(new BorderLayout());
            right.add(p2, BorderLayout.LINE_END);
            vertical.add(right);
            vertical.add(Box.createVerticalStrut(height:15));
            a1.add(vertical, BorderLayout.PAGE_START);
            dout.writeUTF(out);
            text.setText(t:"");
            f.repaint();
            f.invalidate();
            f.validate();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public static JPanel formatLabel(String out) {
        JPanel panel = new JPanel();
        panel.setLayout(new BoxLayout(panel, BoxLayout.Y_AXIS));
        JLabel output = new JLabel("<html><p style=\"width: 150px\">" + out + "</p></html>");
        output.setFont(new Font(name:"Tahoma", Font.PLAIN, size:16));
        output.setBackground(new Color(r:0, g:156, b:255));
        output.setOpaque(isOpaque:true);
        output.setBorder(new EmptyBorder(top:15, left:15, bottom:15, right:50));
        panel.add(output);
        Calendar cal = Calendar.getInstance();
        SimpleDateFormat sdf = new SimpleDateFormat(pattern:"HH:mm");
        JLabel time = new JLabel();
```

```
52          time.setText(sdf.format(cal.getTime()));
53          panel.add(time);
54          return panel;
55      }
        Run | Debug
56      public static void main(String[] args) {
57          new Server();
58          try {
59              ServerSocket skt = new ServerSocket(port:6001);
60              while (true) {
61                  Socket s = skt.accept();
62                  DataInputStream din = new DataInputStream(s.getInputStream());
63                  dout = new DataOutputStream(s.getOutputStream());
64                  while (true) {
65                      String msg = din.readUTF();
66                      JPanel panel = formatLabel(msg);
67                      JPanel left = new JPanel(new BorderLayout());
68                      left.add(panel, BorderLayout.LINE_START);
69                      vertical.add(left);
70                      f.validate();
71                  }
72              }
73          } catch (Exception e) {
74              e.printStackTrace();
75          }
76      }
77
78      Server() {
79          f.setLayout(manager:null);
80          JPanel p1 = new JPanel();
81          p1.setBackground(new Color(r:255, g:165, b:0));
82          p1.setBounds(x:0, y:0, width:450, height:70);
83          p1.setLayout(mgr:null);
84          f.add(p1);
85
86          ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource(name:"images/3.png"));
87          Image i2 = i1.getImage().getScaledInstance(width:25, height:25, Image.SCALE_DEFAULT);
88          ImageIcon i3 = new ImageIcon(i2);
89          JLabel back = new JLabel(i3);
90          back.setBounds(x:5, y:20, width:25, height:25);
91          p1.add(back);
92
93          back.addMouseListener(new MouseAdapter() {
94              public void mouseClicked(MouseEvent ae) {
95                  System.exit(status:0);
96              }
97          });
98          ImageIcon i4 = new ImageIcon(ClassLoader.getSystemResource(name:""));
99          Image i5 = i4.getImage().getScaledInstance(width:50, height:50, Image.SCALE_DEFAULT);
100         ImageIcon i6 = new ImageIcon(i5);
101         JLabel profile = new JLabel(i6);
```

```
        profile.setBounds(x:40, y:10, width:50, height:50);
        p1.add(profile);
        ImageIcon i7 = new ImageIcon(ClassLoader.getSystemResource(name:"images/video.png"));
        Image i8 = i7.getImage().getScaledInstance(width:30, height:30, Image.SCALE_DEFAULT);
        ImageIcon i9 = new ImageIcon(i8);
        JLabel video = new JLabel(i9);
        video.setBounds(x:300, y:20, width:30, height:30);
        p1.add(video);
        ImageIcon i10 = new ImageIcon(ClassLoader.getSystemResource(name:"images/phone.png"));
        Image i11 = i10.getImage().getScaledInstance(width:35, height:30, Image.SCALE_DEFAULT);
        ImageIcon i12 = new ImageIcon(i11);
        JLabel phone = new JLabel(i12);
        phone.setBounds(x:360, y:20, width:35, height:30);
        p1.add(phone);
        ImageIcon i13 = new ImageIcon(ClassLoader.getSystemResource(name:"images/3icon.png"));
        Image i14 = i13.getImage().getScaledInstance(width:10, height:25, Image.SCALE_DEFAULT);
        ImageIcon i15 = new ImageIcon(i14);
        JLabel morevert = new JLabel(i15);
        morevert.setBounds(x:420, y:20, width:10, height:25);
        p1.add(morevert);
        JLabel name = new JLabel(text:"SANYOG");
        name.setBounds(x:110, y:15, width:100, height:18);
        name.setForeground(Color.WHITE);
        name.setFont(new Font(name:"SAN_SERIF", Font.BOLD, size:18));
        p1.add(name);
        JLabel status = new JLabel(text:"Active");
        status.setBounds(x:110, y:35, width:100, height:18);
        status.setForeground(Color.WHITE);
        status.setFont(new Font(name:"SAN_SERIF", Font.BOLD, size:14));
        p1.add(status);
        a1 = new JPanel();
        a1.setBounds(x:5, y:75, width:440, height:500);
        f.add(a1);
        text = new JTextField();
        text.setBounds(x:5, y:575, width:310, height:40);
        text.setFont(new Font(name:"SAN_SERIF", Font.PLAIN, size:16));
        f.add(text);
        JButton send = new JButton(text:"Send");
        send.setBounds(x:320, y:575, width:123, height:40);
        send.setBackground(new Color(r:255, g:165, b:0));
        send.setForeground(Color.WHITE);
        send.addActionListener(this);
        send.setFont(new Font(name:"SAN_SERIF", Font.PLAIN, size:16));
        f.add(send);
        f.setSize(width:450, height:700);
        f.setLocation(x:200, y:50);
        f.setUndecorated(undecorated:true);
        f.getContentPane().setBackground(Color.WHITE);
        f.setVisible(b:true);
    }
}
```

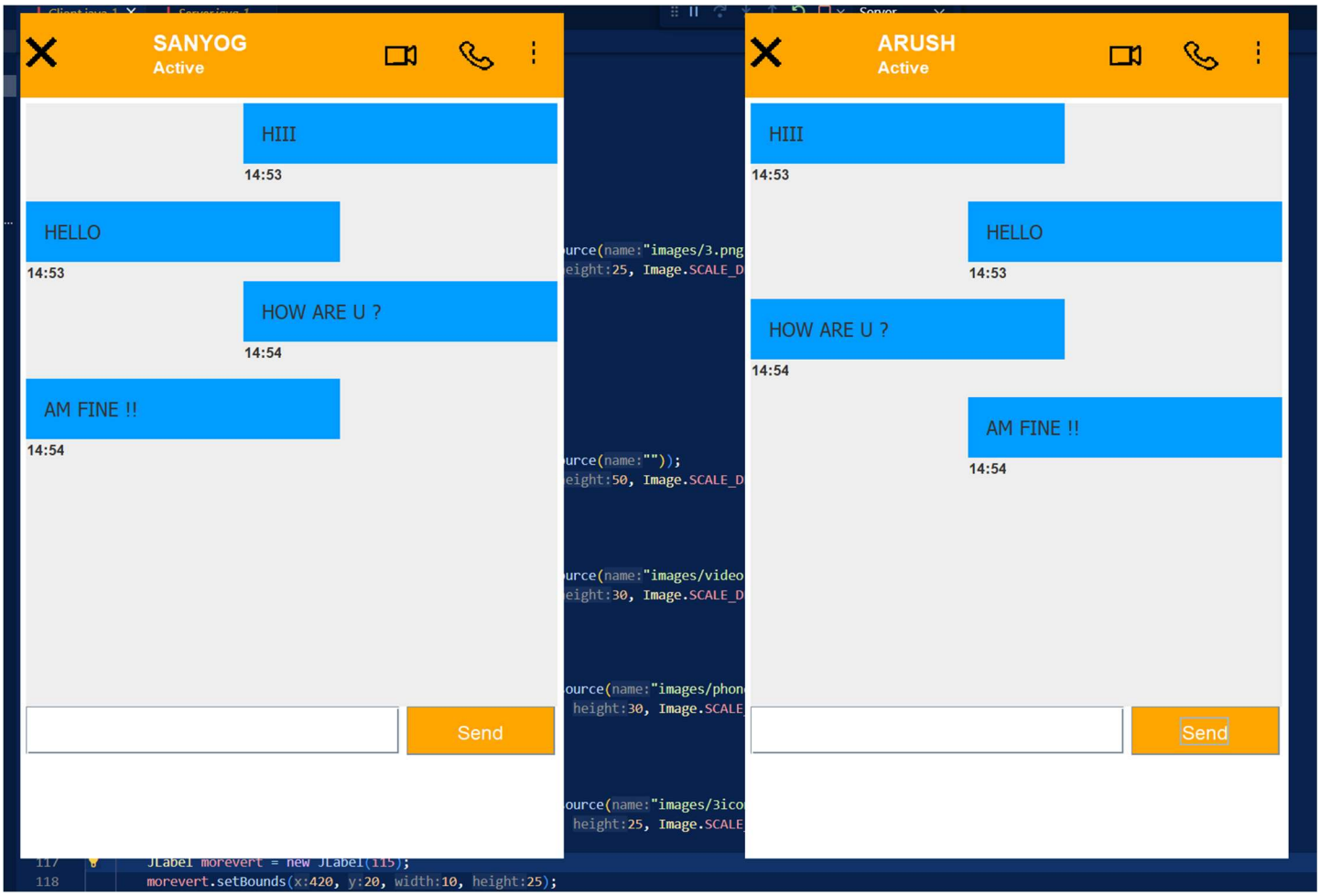
6.2 CLIENT FILE

```
package chatui;

import java.io.*;
import java.util.*;
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
import java.text.*;
import java.net.*;
import javax.swing.border.*;

public class Client implements ActionListener {

    JTextField text;
    static JPanel a1;
    static Box vertical = Box.createVerticalBox();
    static JFrame f = new JFrame();
    static DataOutputStream dout;

    public static JPanel formatLabel(String out) {
        JPanel panel = new JPanel();
        panel.setLayout(new BoxLayout(panel, BoxLayout.Y_AXIS));

        JLabel output = new JLabel("<html><p style=\"width: 150px\">" + out + "</p></html>");
        output.setFont(new Font(name:"Tahoma", Font.PLAIN, size:16));
        output.setBackground(new Color(r:0,g:156,b:255));
        output.setOpaque(isOpaque:true);
        output.setBorder(new EmptyBorder(top:15, left:15, bottom:15, right:50));
        panel.add(output);
        Calendar cal = Calendar.getInstance();
        SimpleDateFormat sdf = new SimpleDateFormat(pattern:"HH:mm");
        JLabel time = new JLabel();
        time.setText(sdf.format(cal.getTime()));
        panel.add(time);
        return panel;
    }

    public void actionPerformed(ActionEvent ae) {
        try {
            String out = text.getText();
            JPanel p2 = formatLabel(out);
            a1.setLayout(new BorderLayout());
            JPanel right = new JPanel(new BorderLayout());
            right.add(p2, BorderLayout.LINE_END);
            vertical.add(right);
            vertical.add(Box.createVerticalStrut(height:15));
            a1.add(vertical, BorderLayout.PAGE_START);
            dout.writeUTF(out);
            text.setText(t:"");
            f.repaint();
            f.invalidate();
```

```
52              f.validate();
53          } catch (Exception e) {
54              e.printStackTrace();
55          }
56      }
        Run | Debug
57      public static void main(String[] args) {
58          new Client();
59          try {
60              Socket s = new Socket(host:"127.0.0.1", port:6001);
61              DataInputStream din = new DataInputStream(s.getInputStream());
62              dout = new DataOutputStream(s.getOutputStream());
63              while (true) {
64                  a1.setLayout(new BorderLayout());
65                  String msg = din.readUTF();
66                  JPanel panel = formatLabel(msg);
67                  JPanel left = new JPanel(new BorderLayout());
68                  left.add(panel, BorderLayout.LINE_START);
69                  vertical.add(left);
70                  vertical.add(Box.createVerticalStrut(height:15));
71                  a1.add(vertical, BorderLayout.PAGE_START);
72                  f.validate();
73              }
74          } catch (Exception e) {
75              e.printStackTrace();
76          }
77      }
78      Client() {
79          f.setLayout(manager:null);
80          JPanel p1 = new JPanel();
81          p1.setBackground(new Color(r:255, g:165, b:0));
82          p1.setBounds(x:0, y:0, width:450, height:70);
83          p1.setLayout(mgr:null);
84          f.add(p1);
85          ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource(name:"images/3.png"));
86          Image i2 = i1.getImage().getScaledInstance(width:25, height:25, Image.SCALE_DEFAULT);
87          ImageIcon i3 = new ImageIcon(i2);
88          JLabel back = new JLabel(i3);
89          back.setBounds(x:5, y:20, width:25, height:25);
90          p1.add(back);
91          back.addMouseListener(new MouseAdapter() {
92              public void mouseClicked(MouseEvent ae) {
93                  System.exit(status:0);
94              }
95          });
96          ImageIcon i4 = new ImageIcon(ClassLoader.getSystemResource(name:""));
97          Image i5 = i4.getImage().getScaledInstance(width:50, height:50, Image.SCALE_DEFAULT);
98          ImageIcon i6 = new ImageIcon(i5);
99          JLabel profile = new JLabel(i6);
100         profile.setBounds(x:40, y:10, width:50, height:50);
101         p1.add(profile);
```

```
        profile.setBounds(x:40, y:10, width:50, height:50);
        p1.add(profile);
        ImageIcon i7 = new ImageIcon(ClassLoader.getSystemResource(name:"images/video.png"));
        Image i8 = i7.getImage().getScaledInstance(width:30, height:30, Image.SCALE_DEFAULT);
        ImageIcon i9 = new ImageIcon(i8);
        JLabel video = new JLabel(i9);
        video.setBounds(x:300, y:20, width:30, height:30);
        p1.add(video);
        ImageIcon i10 = new ImageIcon(ClassLoader.getSystemResource(name:"images/phone.png"));
        Image i11 = i10.getImage().getScaledInstance(width:35, height:30, Image.SCALE_DEFAULT);
        ImageIcon i12 = new ImageIcon(i11);
        JLabel phone = new JLabel(i12);
        phone.setBounds(x:360, y:20, width:35, height:30);
        p1.add(phone);
        ImageIcon i13 = new ImageIcon(ClassLoader.getSystemResource(name:"images/3icon.png"));
        Image i14 = i13.getImage().getScaledInstance(width:10, height:25, Image.SCALE_DEFAULT);
        ImageIcon i15 = new ImageIcon(i14);
        JLabel morevert = new JLabel(i15);
        morevert.setBounds(x:420, y:20, width:10, height:25);
        p1.add(morevert);
        JLabel name = new JLabel(text:"UNKNOWN");
        name.setBounds(x:110, y:15, width:100, height:18);
        name.setForeground(Color.WHITE);
        name.setFont(new Font(name:"SAN_SERIF", Font.BOLD, size:18));
        p1.add(name);
        JLabel status = new JLabel(text:"Active");
        status.setBounds(x:110, y:35, width:100, height:18);
        status.setForeground(Color.WHITE);
        status.setFont(new Font(name:"SAN_SERIF", Font.BOLD, size:14));
        p1.add(status);
        a1 = new JPanel();
        a1.setBounds(x:5, y:75, width:440, height:500);
        f.add(a1);
        text = new JTextField();
        text.setBounds(x:5, y:575, width:310, height:40);
        text.setFont(new Font(name:"SAN_SERIF", Font.PLAIN, size:16));
        f.add(text);
        JButton send = new JButton(text:"Send");
        send.setBounds(x:320, y:575, width:123, height:40);
        send.setBackground(new Color(r:255, g:165, b:0));
        send.setForeground(Color.WHITE);
        send.addActionListener(this);
        send.setFont(new Font(name:"SAN_SERIF", Font.PLAIN, size:16));
        f.add(send);
        f.setSize(width:450, height:700);
        f.setLocation(x:800, y:50);
        f.setUndecorated(undecorated:true);
        f.getContentPane().setBackground(Color.WHITE);
        f.setVisible(b:true);
    }
}
```

6. RESULTS


SANYOG
Active
HIII
14:53
HELLO
14:53
HOW ARE U ?
14:54
AM FINE !!
14:54
Send
ARUSH
Active
HIII
14:53
HELLO
14:53
HOW ARE U ?
14:54
AM FINE !!
14:54
Send
JLabel morevert = new JLabel(115);
morevert.setBounds(x:420, y:20, width:10, height:25);

7. CONCLUSION

Our exploration delved into the inner workings of server-client interactions in Java. We gained insights into crafting graphical user interfaces (GUI) through Java Swing, enhancing our proficiency in the language. This project served as a valuable educational experience, deepening our understanding of Java, Java Swing, as well as the dynamics of server-client communication in a comprehensible manner. Ultimately, our efforts culminated in the development of a fully operational and secure chat room server, offering users the capability to engage in private and confidential conversations.

8. REFERENCES

[1] “Most popular global mobile messenger apps 2014 | Statistic,” Statista. [Online]. Available: http://www.statista.com/statistics/258749/most-popular-global-mobilemessenger-apps/

[2] “Secure Messaging Scorecard,” Electronic Frontier Foundation. [Online]. Available: https://www.eff.org/secure-messaging-scorecard .

[3] “Hacking attack on accounts connected with popular messenger app - Australia Network News (Australian Broadcasting Corporation),” Hacking attack on accounts connected with popular messenger app. [Online]. Available: http://www.abc.net.au/news/2014-06-19/an-asia-cyber-crime/5537194 .

[4] “WeChat Developers,” WeChat API Documentation. [Online]. Available: http://dev.wechat.com/wechatapi/documentation .

[5] “• Number of apps available in leading app stores 2014 | Statistic.” [Online]. Available: http://www.statista.com/statistics/276623/number-of-apps-available-inleading-app-stores/ .

[6] “• Number of mobile messaging users worldwide 2014 | Statistic.” [Online]. Available: http://www.statista.com/statistics/369260/mobile-messenger-users/ .

[7] “Wickr | Top Secret Messenger,” Wickr | Top Secret Messenger. [Online]. Available: https://wickr.com/

[8] “unhcfreg | Viber Security Vulnerabilities: Do not use Viber until these issues are resolved.” [Online]. Available: http://www.unhcfreg.com/#!Viber-SecurityVulnerabilities-Do-not-use-Viber-until-these-issues-are-resolved/c5rt/BB4208CF7F0A-4DE1-92A4-529425549683 .

[9] S. Schrittwieser, P. Kieseberg, L. ̈Manuel, P. Fruhwirt, M. Mulazzani, M. Huber, and E. Weippl, “Guess Who’s Texting You? Evaluating the Security of Smartphone Messaging Applications.” [Online]. Available: https://www.sba- 36 | References research.org/wp content/uploads/publications/ndss2012_final.pdf

[10] C. Cattiaux, “iMessage Privacy,” 17-Oct-2013. [Online]. Available: http://blog.quarkslab.com/imessage-privacy.html .

[11] A. Iqbal, A. Marrington, and I. Baggili, “Forensic artifacts of the ChatON Instant Messaging application,” in 2013 Eighth International Workshop on Systematic Approaches to Digital Forensic Engineering (SADFE), 2013, pp. 1–6.

[12] “unhcfreg,” UNH Cyber Forensics Group Reveals Smartphone App Issues Affecting 968 Million. [Online]. Available: http://www.unhcfreg.com/. [Accessed: 03-Dec-2014]. [19] “SQLCipher has 100M+ Mobile Users (Thanks to WeChat!) | The Guardian Project,” SQLCipher has 100M+ Mobile Users (Thanks to WeChat!). [Online]. Available: https://guardianproject.info/2013/12/10/sqlcipher-has-300-millionmobile-users-thanks-to-wechat/ .