

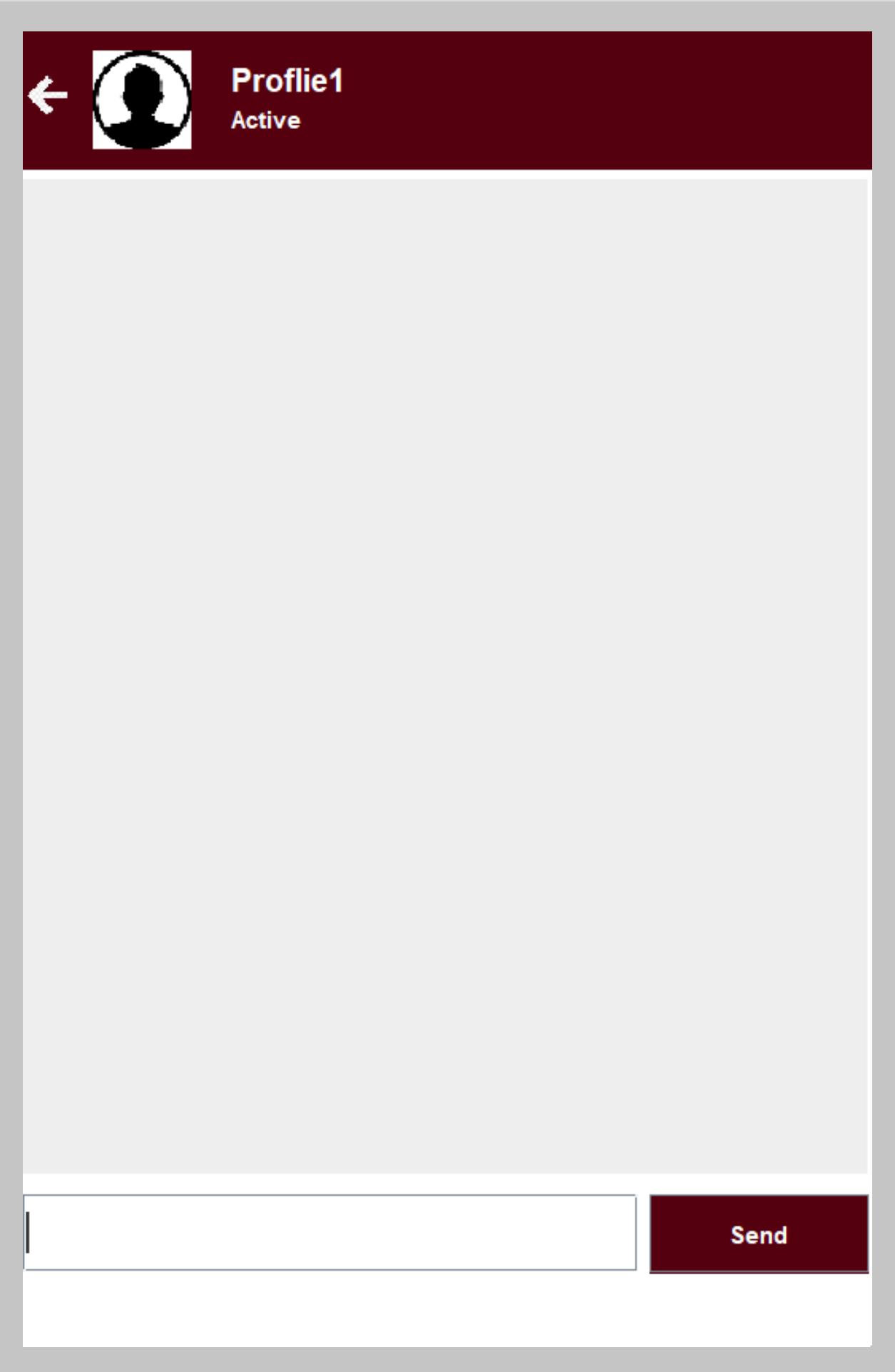
JAVA PROJECT

CHAT APP

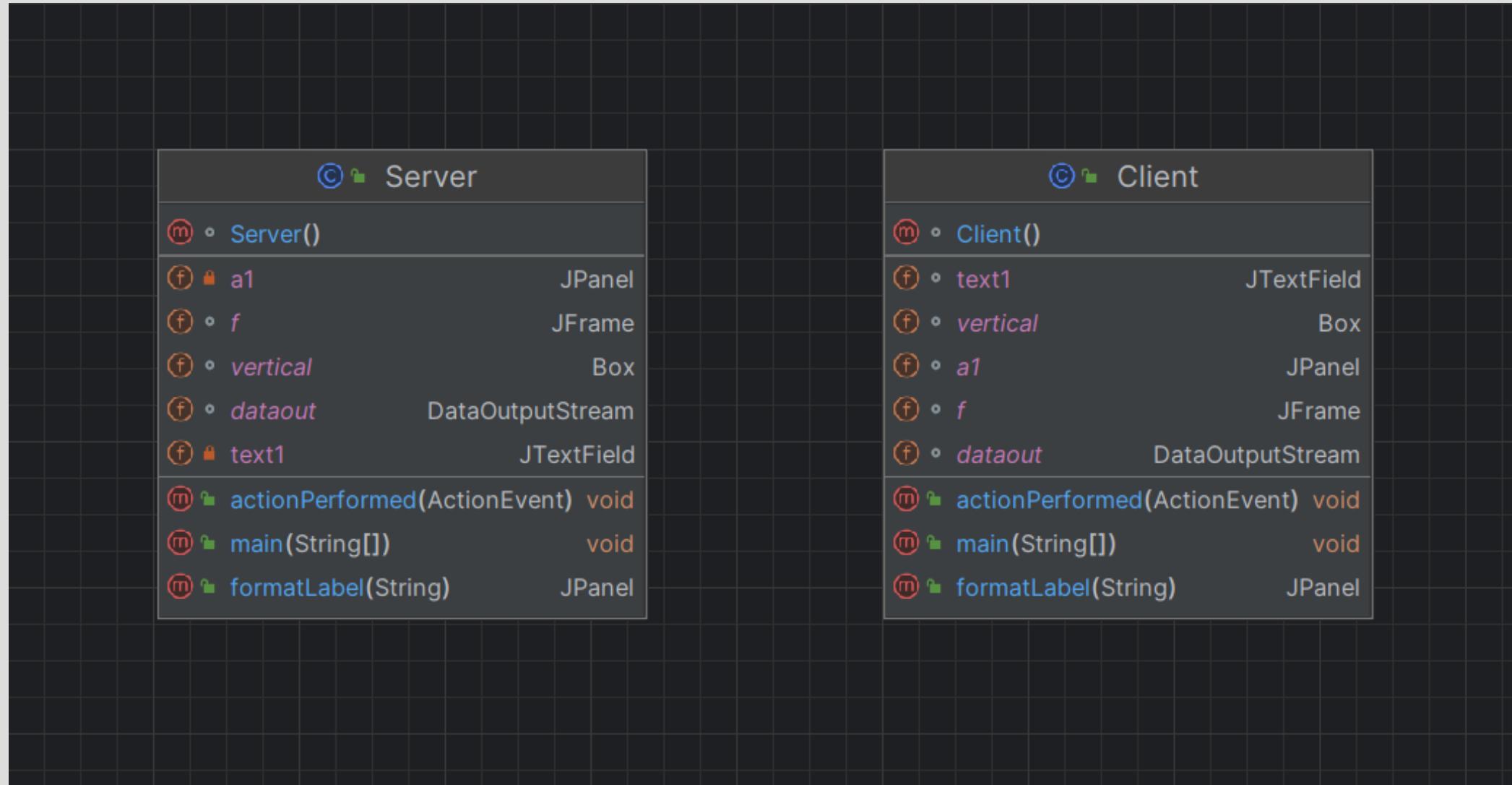
นาย พิสิฐพงศ์ จันทร์สุขตรี 6630300629
วิชา *Programing fundamentals II*

CHAT APP

Component ของ Chat APP จะประกอบด้วย *TextField*, *ImageIcon*, *Label*, *Panel* และ *Button*. *TextField* จะรับ *input* จากแป้นพิมพ์และเมื่อกดบุ่ม "Send" จะทำการนำข้อความจาก *TextField* ไปแสดงทางด้านขวาของ *Panel* และข้อความจะขึ้นทางด้านซ้าย *Panel* ของผู้รับ



UML DIAGRAM



-Class **Server** เป็นคลาส GUI ที่รับ input จากแป้นพิมพ์/คีย์บอร์ด และแสดงข้อความทางด้านขวา และในตัวคลาส **Server** มีการใช้ **Socket** และกำหนด **Port** ไว้เพื่อรอให้ **Client** มาเชื่อมต่อ

-Class **Client** เป็นคลาส GUI เช่นเดียวกับ **Server** เพื่อรับข้อความจาก **Server** และสามารถส่งข้อความไปหา **Server** ได้โดยใช้ **Port** ที่ **Server** กำหนดไว้

การทำงานของ Server ส่วน GUI

คลาส Server ส่วนนี้จะเป็นการ Import คลาสต่างๆที่จะใช้งาน และมี Constructor ที่ไม่ได้รับ parameter ใดๆ ภายใน constructor จะเป็นการจัด Component ต่างๆ เช่น JFrame , JPanel , JButton , JTextField เป็นต้น และยังมีการใช้ MouseListener เพื่อที่จะคลิกและปิดตัว GUI และ มีการใช้ KeyListener โดยเมื่อกดบุ้ม Enter บนแป้นพิมพ์สามารถส่งข้อความขึ้นไปยัง Panel.

```
1 import javax.swing.*;
2 import javax.swing.border.EmptyBorder;
3 import java.awt.*;
4 import java.awt.event.*;
5 import java.io.DataInputStream;
6 import java.net.ServerSocket;
7 import java.net.Socket;
8 import java.util.*;
9 import java.text.*;
10 import java.io.*;
```

```
1 public class Server implements ActionListener {
2     private JTextField text1;
3     private JPanel a1;
4     static Box vertical = Box.createVerticalBox();
5     static JFrame f = new JFrame("Chat GUI");
6     static DataOutputStream dataout;
7     Server() {
8
9         f.setLayout(null);
10
11         JPanel p1 = new JPanel();
12         p1.setBackground(new Color(84, 0, 15));
13         p1.setBounds(0, 0, 450, 70);
14         p1.setLayout(null);
15         f.add(p1);
16
17         ImageIcon im1 = new ImageIcon(ClassLoader.getSystemResource("image\\3.png"));
18         Image im2 = im1.getImage().getScaledInstance(25, 25, Image.SCALE_DEFAULT);
19         ImageIcon im3 = new ImageIcon(im2);
20         JLabel back = new JLabel(im3);
21         back.setBounds(5, 20, 25, 25);
22         p1.add(back);
23
24         back.addMouseListener(new MouseAdapter() {
25             @Override
26             public void mouseClicked(MouseEvent e) {
27                 System.exit(0);
28             }
29         });
30
31         ImageIcon im4 = new ImageIcon(ClassLoader.getSystemResource("image\\images.png"));
32         Image im5 = im4.getImage().getScaledInstance(50, 50, Image.SCALE_DEFAULT);
33         ImageIcon im6 = new ImageIcon(im5);
34         JLabel profile = new JLabel(im6);
35         profile.setBounds(40, 10, 50, 50);
36         p1.add(profile);
37
38         JLabel name = new JLabel("Profile1");
39         name.setBounds(110, 15, 150, 18);
40         name.setForeground(Color.WHITE);
41         name.setFont(new Font("TH SarabunPSK", Font.BOLD, 16));
42         p1.add(name);
43
44         JLabel status = new JLabel("Active");
45         status.setBounds(110, 35, 150, 18);
46         status.setForeground(Color.WHITE);
47         status.setFont(new Font("TH SarabunPSK", Font.BOLD, 12));
48         p1.add(status);
49
50         JButton send = new JButton("Send");
51         send.setBounds(320, 585, 111, 40);
52         send.setBackground(new Color(84, 0, 15));
53         send.setForeground(Color.WHITE);
54         send.addActionListener(this);
55         f.add(send);
56
57         text1 = new JTextField();
58         text1.setBounds(5, 585, 310, 40);
59         text1.setFont(new Font("Angsana New", Font.BOLD, 18));
56         f.add(text1);
57
58         text1.addKeyListener(new KeyAdapter() {
59             public void keyPressed(KeyEvent e) {
60                 if (e.getKeyCode() == KeyEvent.VK_ENTER) {
61                     actionPerformed(new ActionEvent(this, ActionEvent.ACTION_PERFORMED, null));
62                 }
63             }
64         });
65
66         a1 = new JPanel();
67         a1.setBounds(5, 75, 425, 500);
68
69         f.add(a1);
70         f.setSize(450, 700);
71         f.setLocation(200, 50);
72         f.getContentPane().setBackground(Color.WHITE);
73         f.setVisible(true);
74         f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
75
76     }
77
78 }
```

การทำงานของ Server ส่วน ActionEvent

ใน Code ล้วนนี้เป็นการทำงานของ *ActionEvent* โดยกำหนดเงื่อนไขว่าตัวข้อความจะไม่ใช้ข้อความเปล่า เมื่อตรงกับเงื่อนไขแล้ว จะทำการส่งข้อความไปที่ *Panel* ทางด้านขวา และส่งข้อความที่รับมาราจากแป้นพิมพ์ไปหา *Client* จากที่กดส่งข้อความแล้วจะทำให้ *TextField* ลบข้อความออกไป และทำการรีเฟรช *GUI*.

```
1  @Override
2      public void actionPerformed(ActionEvent e) {
3          try {
4              String out = text1.getText().trim();
5
6              if (!out.isEmpty()) {
7
8                  JPanel p2 = formatLabel(out);
9
10                 a1.setLayout(new BorderLayout());
11
12                 JPanel right = new JPanel(new BorderLayout());
13                 right.add(p2, BorderLayout.LINE_END);
14                 vertical.add(right);
15                 vertical.add(Box.createVerticalStrut(15));
16
17                 a1.add(vertical, BorderLayout.PAGE_START);
18
19                 dataout.writeUTF(out);
20                 text1.setText("");
21
22                 f.repaint();
23                 f.invalidate();
24                 f.validate();
25
26             }
27         }catch(IOException ex){
28             ex.printStackTrace();
29         }
30     }
```

การทำงานของ

Server ส่วนกล่องข้อความ และ Main

- ในส่วนของ `FormatLabel` จะเป็นเมธอดสำหรับสร้าง `Panel` ที่มีกล่องข้อมูลความและเวลาที่ส่งความนั้นๆ โดยในกล่องข้อความจะมีการใช้ `HTML` ช่วยในการกำหนดความกว้างของข้อความหากเกินจะทำการขึ้นบรรทัดใหม่.

- ในส่วนของ `Main` จะเป็นจดที่เริ่มต้นของ `Server` โดยจะทำการสร้าง `ServerSocket` และกำหนด Port โดยในที่นี้ได้กำหนด Port 6001 เพื่อรองรับการเชื่อมต่อจาก `Client` เมื่อเชื่อมแล้วตัวโปรแกรมจะสร้าง `Socket` เพื่อติดต่อกับ `Client`

```
1 public static JPanel formatLabel(String out) {
2     JPanel panel = new JPanel();
3     panel.setLayout(new BoxLayout(panel, BoxLayout.Y_AXIS));
4
5     JLabel output = new JLabel("<html><p style=" width: 150px\">" + out + "</p></html>");
6     output.setBackground(new Color(84, 0, 15));
7     output.setOpaque(true);
8     output.setFont(new Font("Angsana New", Font.BOLD, 18));
9     output.setBorder(new EmptyBorder(10, 10, 10, 10));
10    output.setForeground(Color.WHITE);
11
12    panel.add(output);
13
14    Calendar cal = Calendar.getInstance();
15    SimpleDateFormat sdf = new SimpleDateFormat("HH:mm");
16
17    JLabel time = new JLabel();
18    time.setText(sdf.format(cal.getTime()));
19
20    panel.add(time);
21    return panel;
22 }
23
24 public static void main(String[] args) {
25
26    new Server();
27
28    try {
29        ServerSocket skt = new ServerSocket(6001);
30        while(true){
31            Socket s = skt.accept();
32            DataInputStream data = new DataInputStream(s.getInputStream());
33            dataout = new DataOutputStream(s.getOutputStream());
34
35            while (true){
36                String msg = data.readUTF();
37                JPanel panel = formatLabel(msg);
38
39                JPanel left = new JPanel(new BorderLayout());
40                left.add(panel, BorderLayout.LINE_START);
41                vertical.add(left);
42                f.validate();
43            }
44        }
45    } catch (Exception e) {
46        e.printStackTrace();
47    }
48 }
49 }
```

การทำงานของ Client

ในส่วนของคลาส Client จะແນ້ມືອນກັນຕົວຂອງ Server ແຕ່ຈະມີສິ່ງທີ່ແຕກຕ່າງດີ່ອຕົວຂອງ Socket ໂດຍ Socket ຂອງ Client ຈະເປັນກາຮ່ອມຕ່ອງກັນ Server ໂດຍໃຫ້ທີ່ອູ່ IP ແລະ Port ໃນທີ່ເປັນຕ່ອງກັນ Server ໂດຍໃຫ້ IP 127.0.0.1 ແລະ Port 6001 ທີ່ຕົວຂອງ Server ກຳເຊັນແລ້ວຈາກນີ້ ຕົວ Client ຈະຮອັບຂ້ອມລຸ່ມທີ່ສົ່ງມາຈາກ Server ແລະ ຈະນຳຂ້ອດຄວາມທີ່ໄດ້ຮັນຈາກ Server ໄປໄວ້ກາງດ້ານຫ້າຍຂອງ Panel .

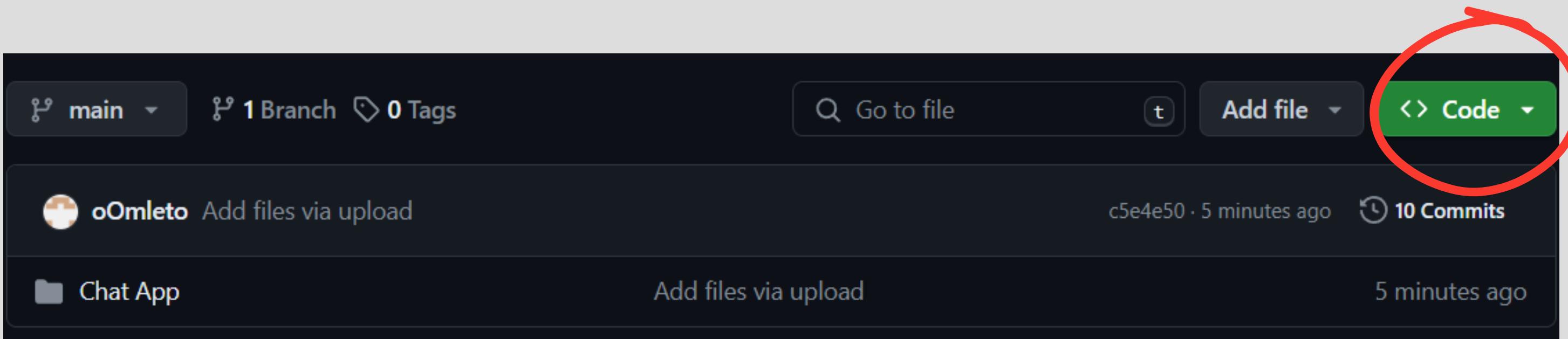
```
1  public static void main(String[] args) {  
2  
3      new Client();  
4  
5      try{  
6          Socket s = new Socket("127.0.0.1",6001);  
7          DataInputStream data = new DataInputStream(s.getInputStream());  
8          dataout = new DataOutputStream(s.getOutputStream());  
9          while (true){  
10              a1.setLayout(new BorderLayout());  
11              String msg = data.readUTF();  
12              JPanel panel = formatLabel(msg);  
13  
14              JPanel left = new JPanel(new BorderLayout());  
15              left.add(panel,BorderLayout.LINE_START);  
16              vertical.add(left);  
17  
18              vertical.add(Box.createVerticalStrut(15));  
19              a1.add(vertical,BorderLayout.PAGE_START);  
20              f.validate();  
21          }  
22      }catch (Exception e){  
23          e.printStackTrace();  
24      }  
25  }
```

ข้อจำกัดของ CHAT APP

- ในส่วนของโปรแกรม CHAT APP ยังมีข้อจำกัด ดื้อ หากต้องการที่จะใช้งาน จำเป็นที่จะต้อง Compile ตัว Server ก่อนแล้วตามด้วย Client .
- หลังจากที่ Compile ทึ้ง2แล้ว จะต้องให้ทาง Server ส่งข้อมูลมาให้ทาง Client ก่อนจึงจะสามารถใช้งานตัวโปรแกรมได้ตามปกติ

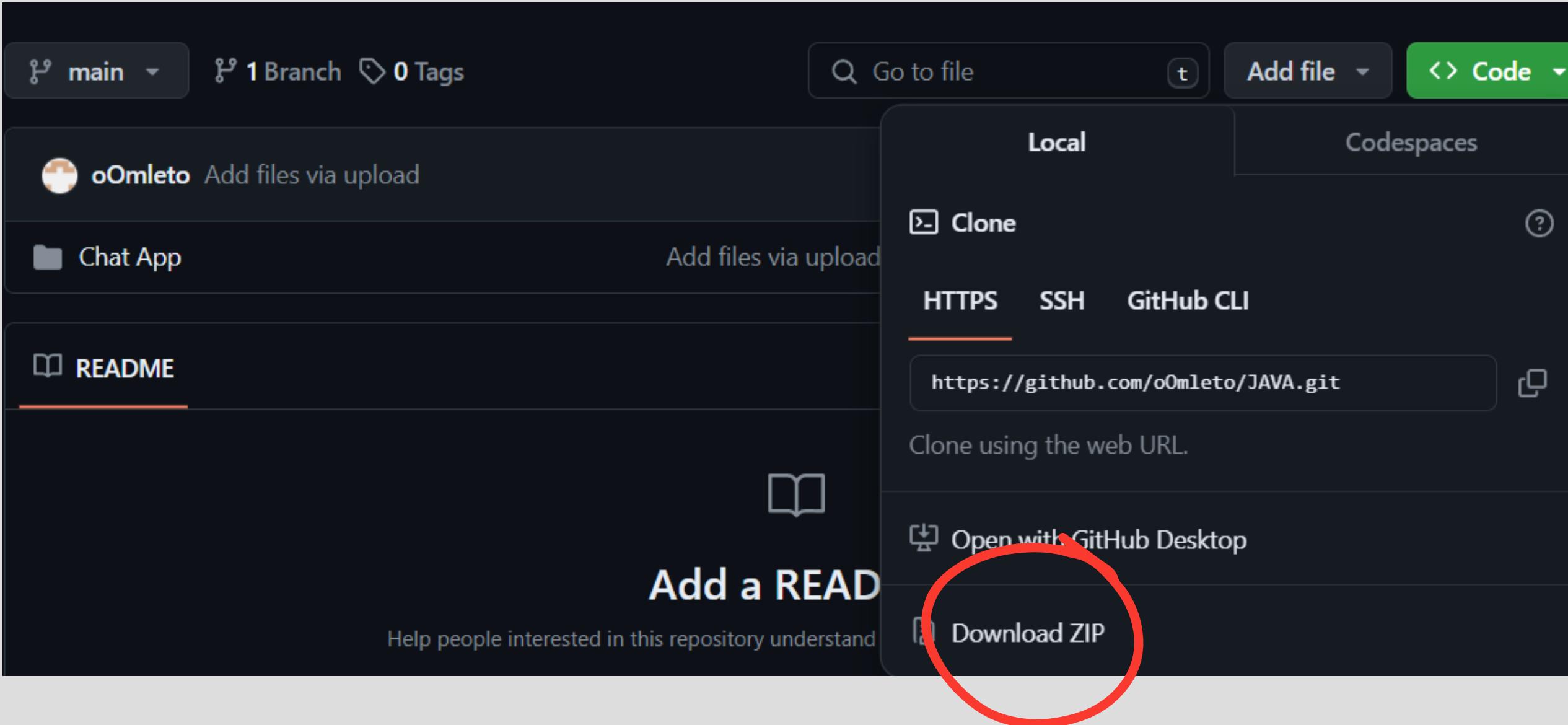
LIMITED

วิธีการติดตั้งและการใช้งาน



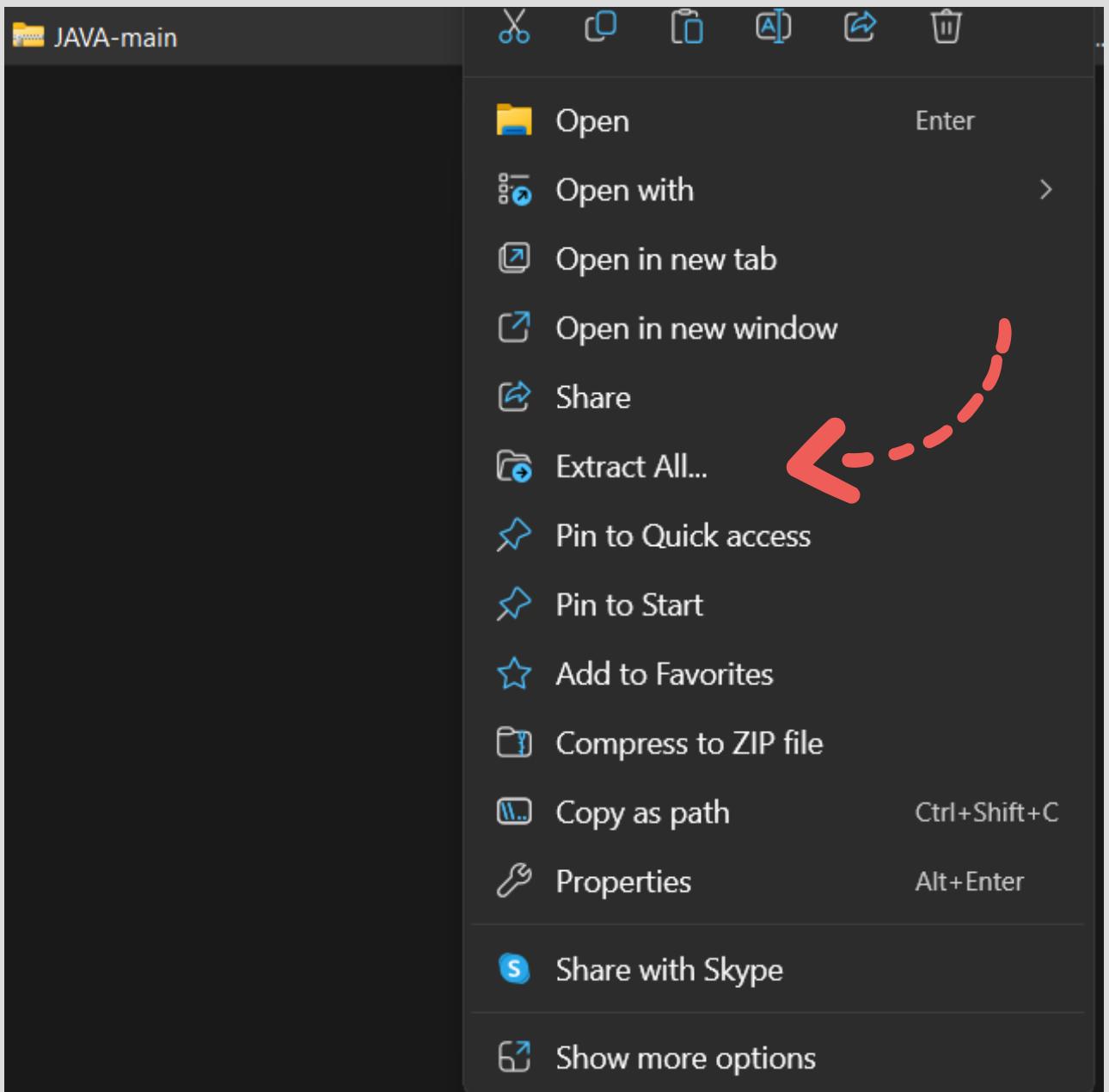
เข้าไปที่ <https://github.com/oOmleto/JAVA> และกดปุ่ม *Code* สีเขียว

วิธีการติดตั้งและการใช้งาน



กด *Download* เป็นไฟล์ zip

วิธีการติดตั้งและการใช้งาน



เมื่อ Download เสร็จ ให้ทำการแตกไฟล์

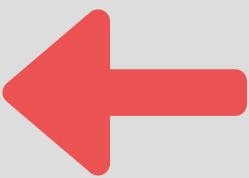
วิธีการติดตั้งและการใช้งาน

Name	Date modified	Type	Size
out	21/3/2567 21:56	File folder	
src	21/3/2567 21:56	File folder	
Chat App.iml	21/3/2567 21:56	IML File	1 KB

คลิกเข้าใน Folder “src”

วิธีการติดตั้งและการใช้งาน

📁 Client_jar	21/3/2567 21:56	File folder
📁 image	21/3/2567 21:56	File folder
📁 Server_jar	21/3/2567 21:56	File folder
📝 Client	21/3/2567 21:56	Java Source File
📝 Server	21/3/2567 21:56	Java Source File



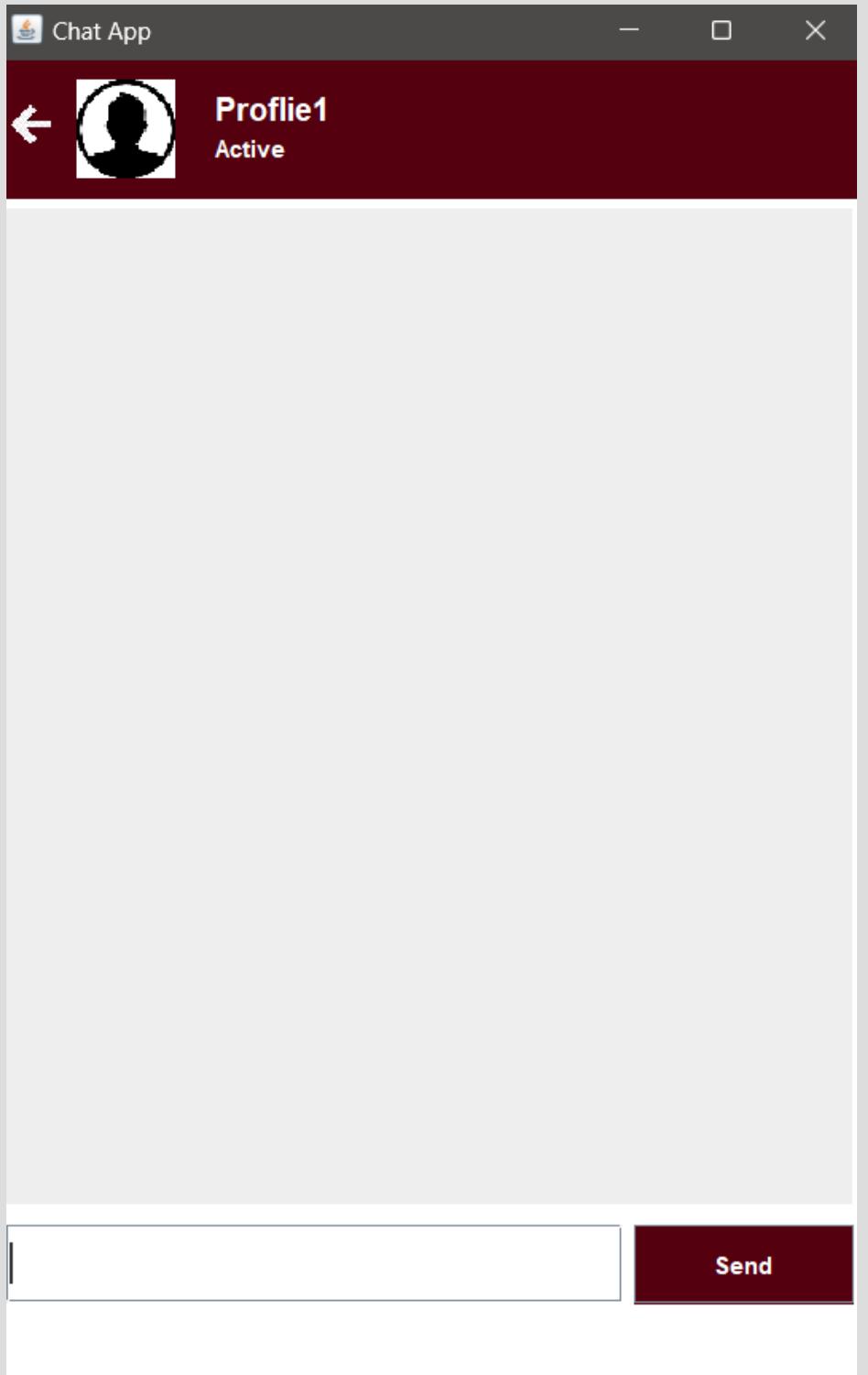
เข้าไปที่ *Server_jar*

💻 Server	21/3/2567 21:56	JAR File	39 KB
----------	-----------------	----------	-------

เปิด *Server* ได้เลย

วิธีการติดตั้งและการใช้งาน

จะได้เป็นตัว GUI ชื่อ Profile1 ขึ้นมาแต่เพียงเท่านั้น



วิธีการติดตั้งและการใช้งาน

📁 Client_jar	21/3/2567 21:56	File folder
📁 image	21/3/2567 21:56	File folder
📁 Server_jar	21/3/2567 21:56	File folder
📄 Client	21/3/2567 21:56	Java Source File 6 KB
📄 Server	21/3/2567 21:56	Java Source File 5 KB

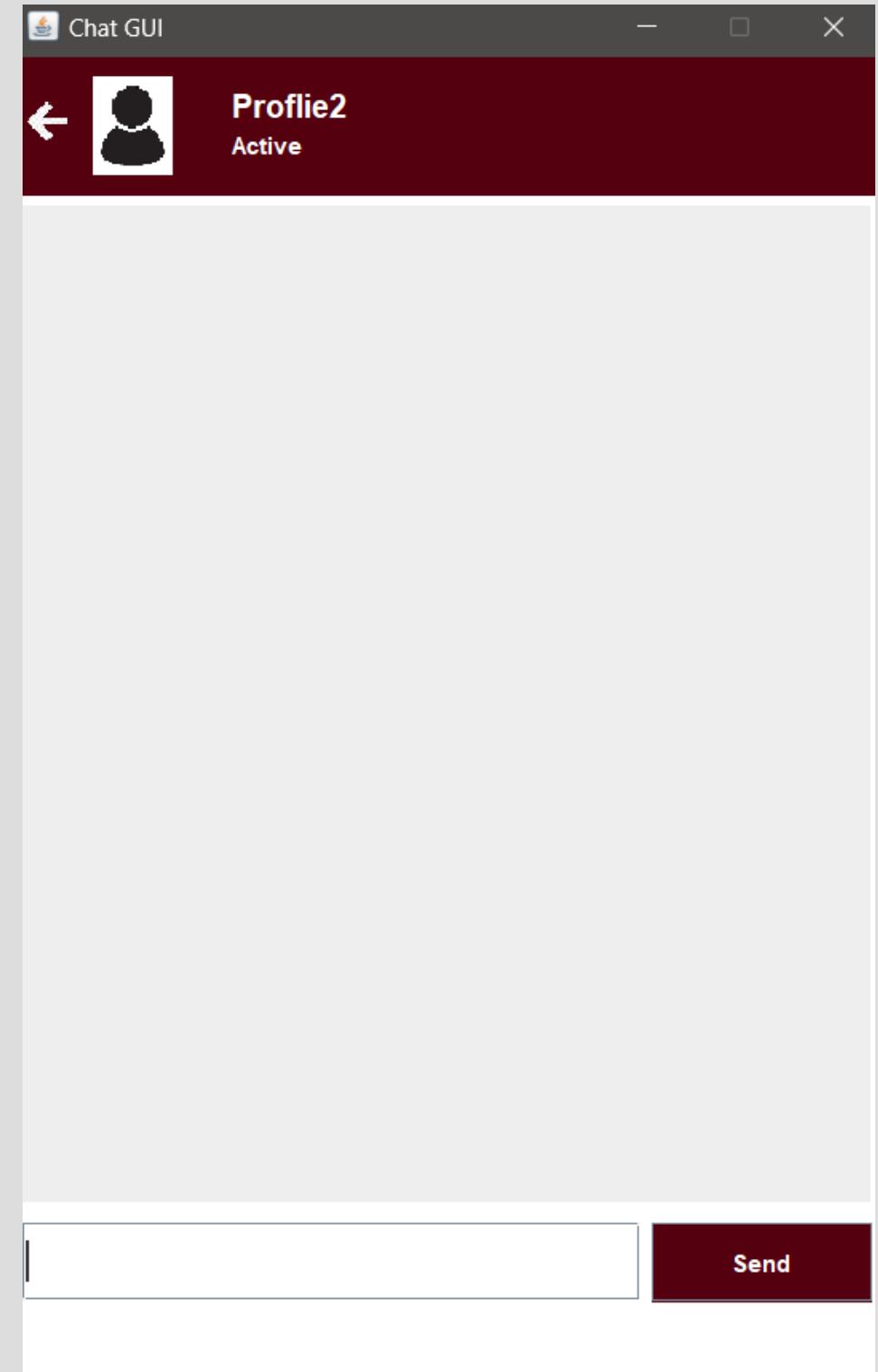
← คลิกจากนั้นให้ย้อนกลับมาเปิด Client

💻 Client	21/3/2567 21:56	JAR File	39 KB
----------	-----------------	----------	-------

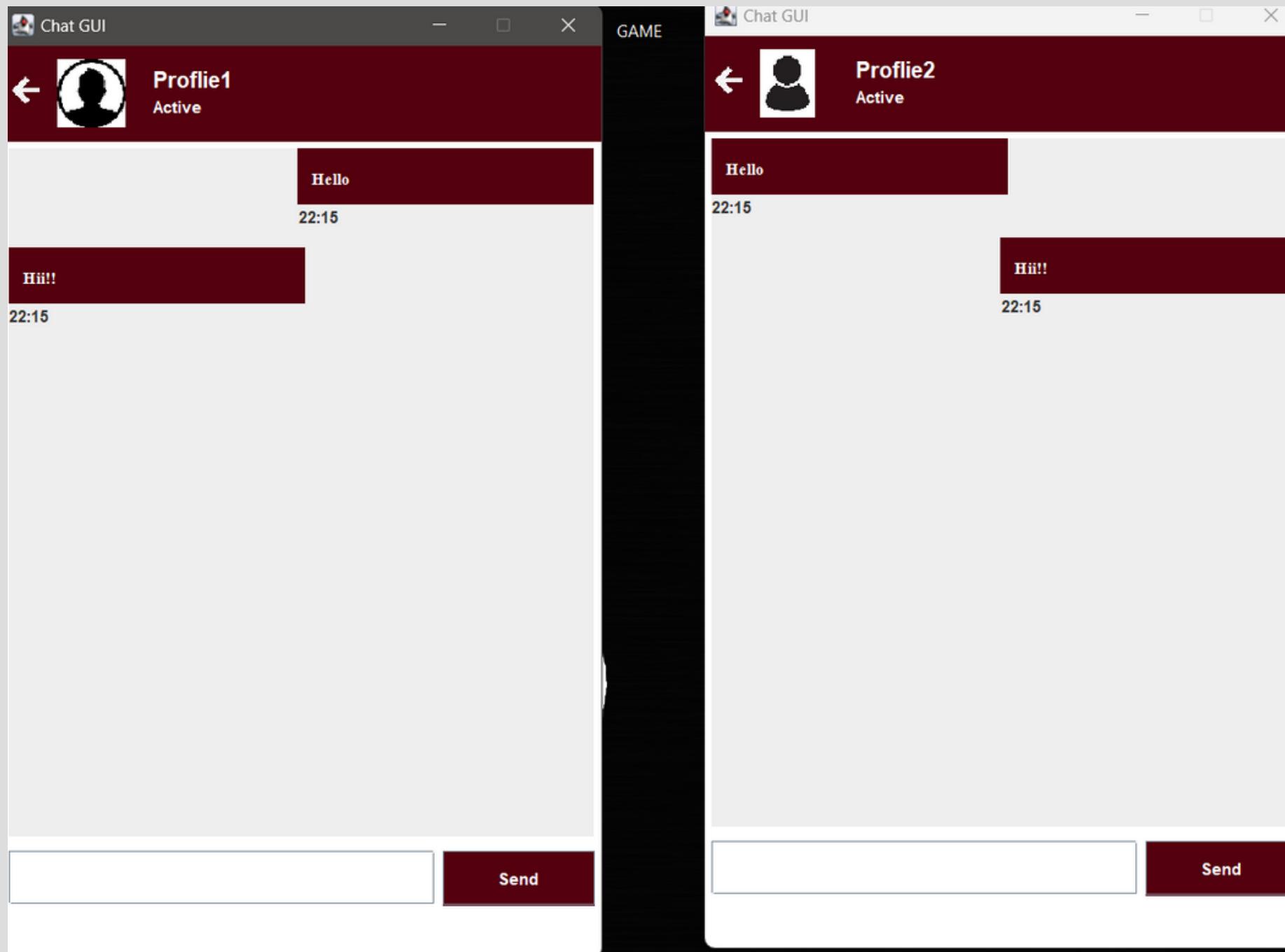
เปิด Client เลย

วิธีการติดตั้งและการใช้งาน

จะได้เป็นตัว GUI ชื่อ Profile2 ขึ้นมา



วิธีการติดต่อและการใช้งาน



คลิกจากนั้นให้ส่งข้อความจาก Profile1 ไปหา Profile2

**THANK
YOU VERY
MUCH!**