المعهد الوطني للعلوم التطبيقية والتكنولوجيا

Institut national des sciences appliquées et de technologies

Compte rendu

TP1

Fondements des systèmes répartis
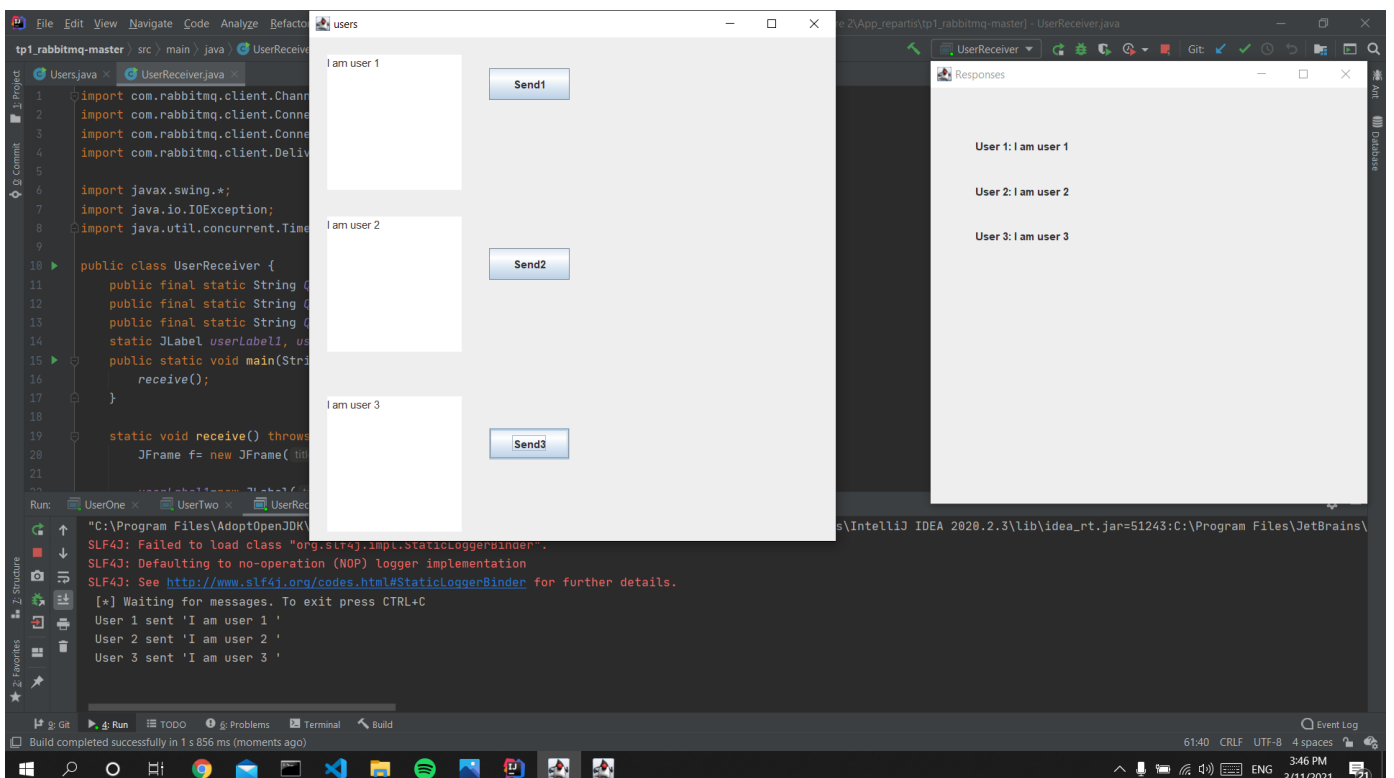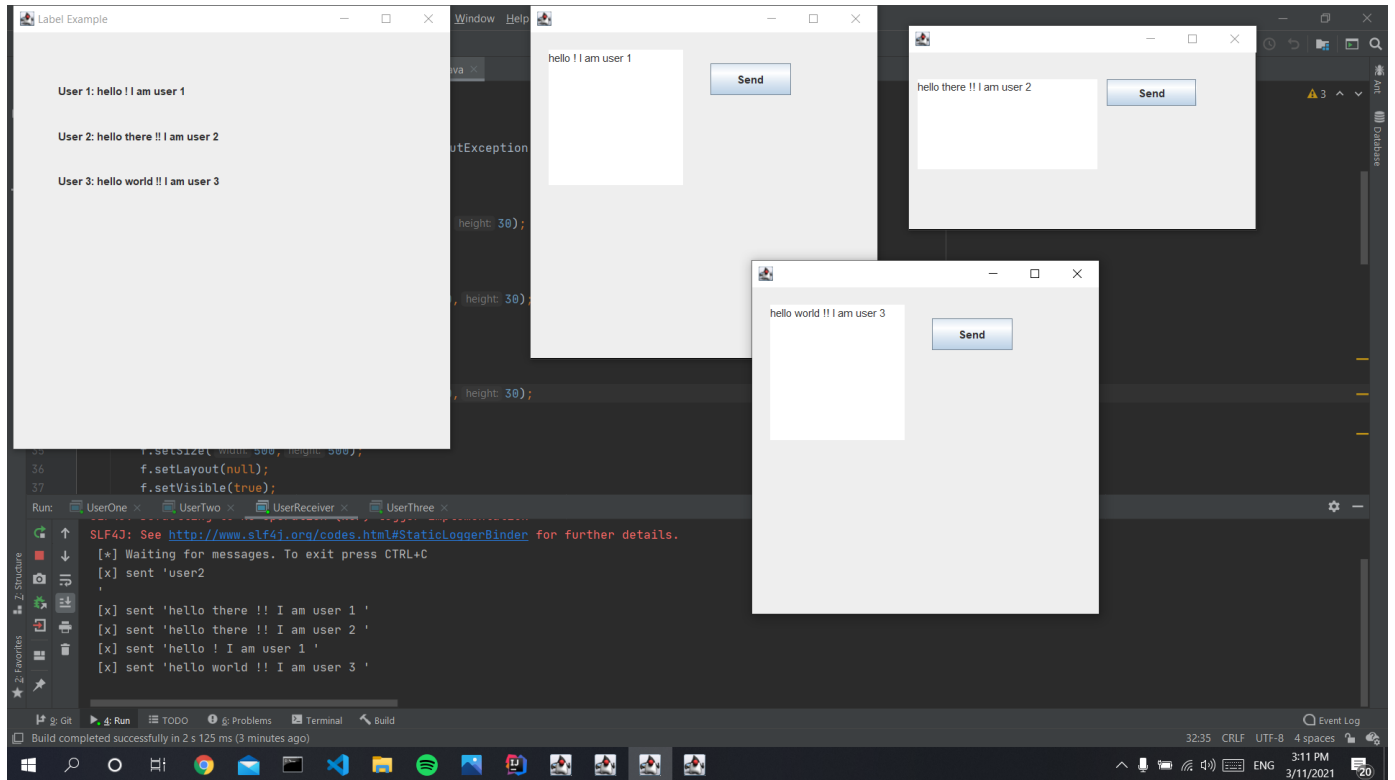
Réalisé par :
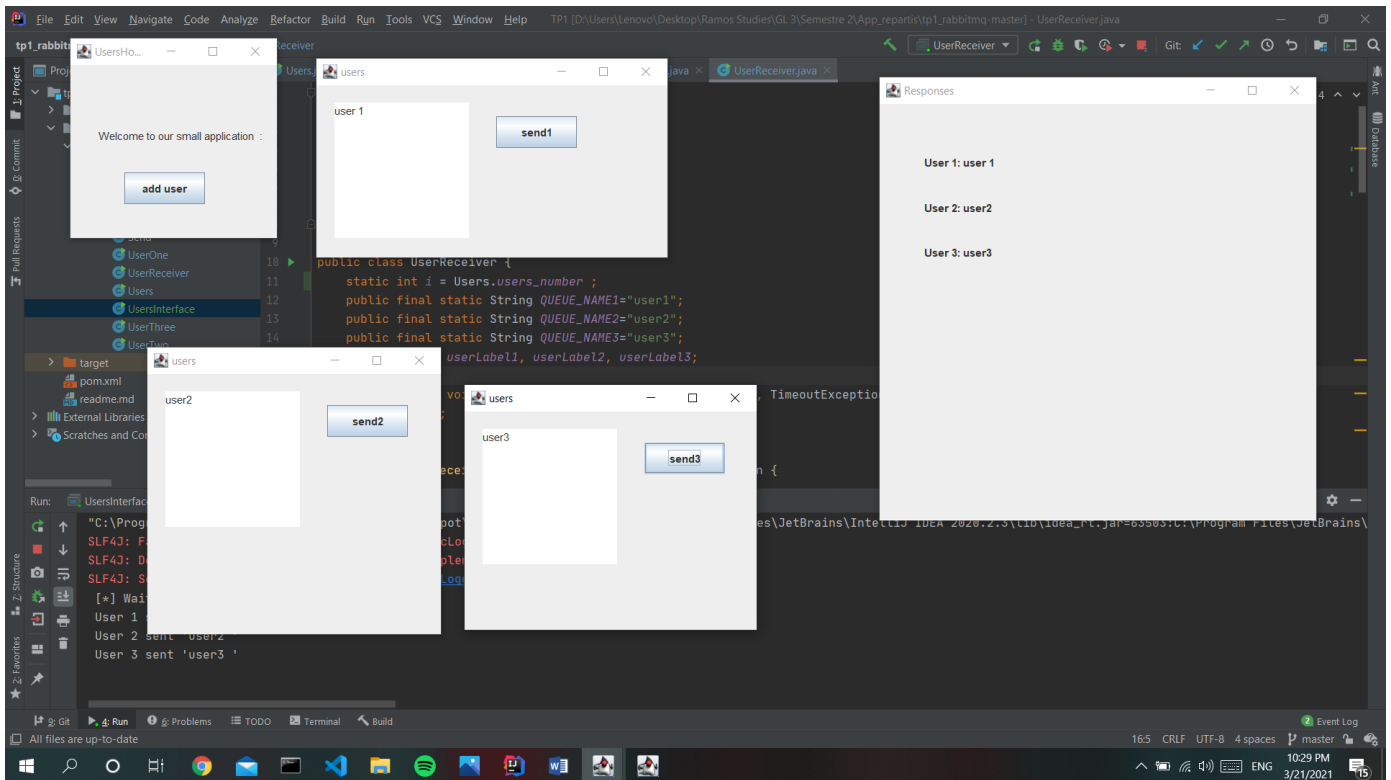
Ben Aribia Ramez

Hamdi Mahdi

GL 3/2
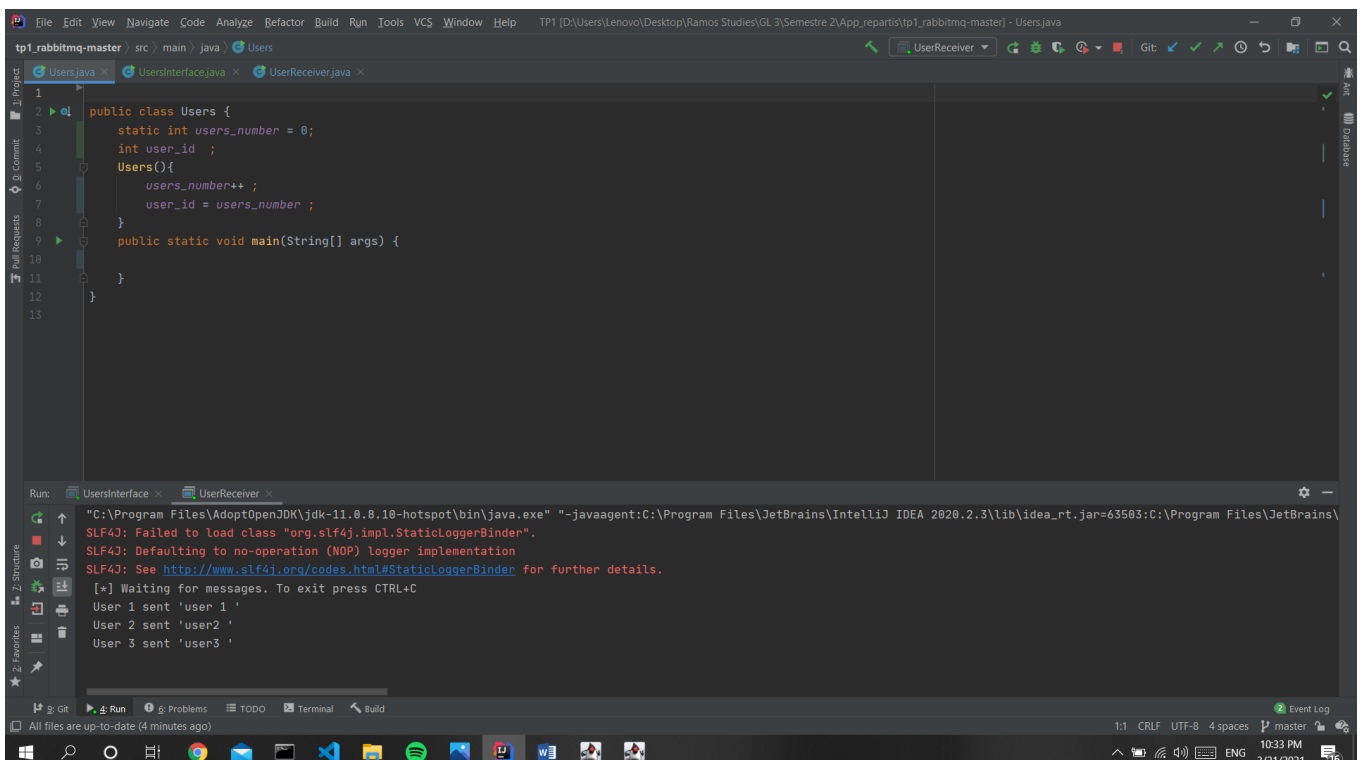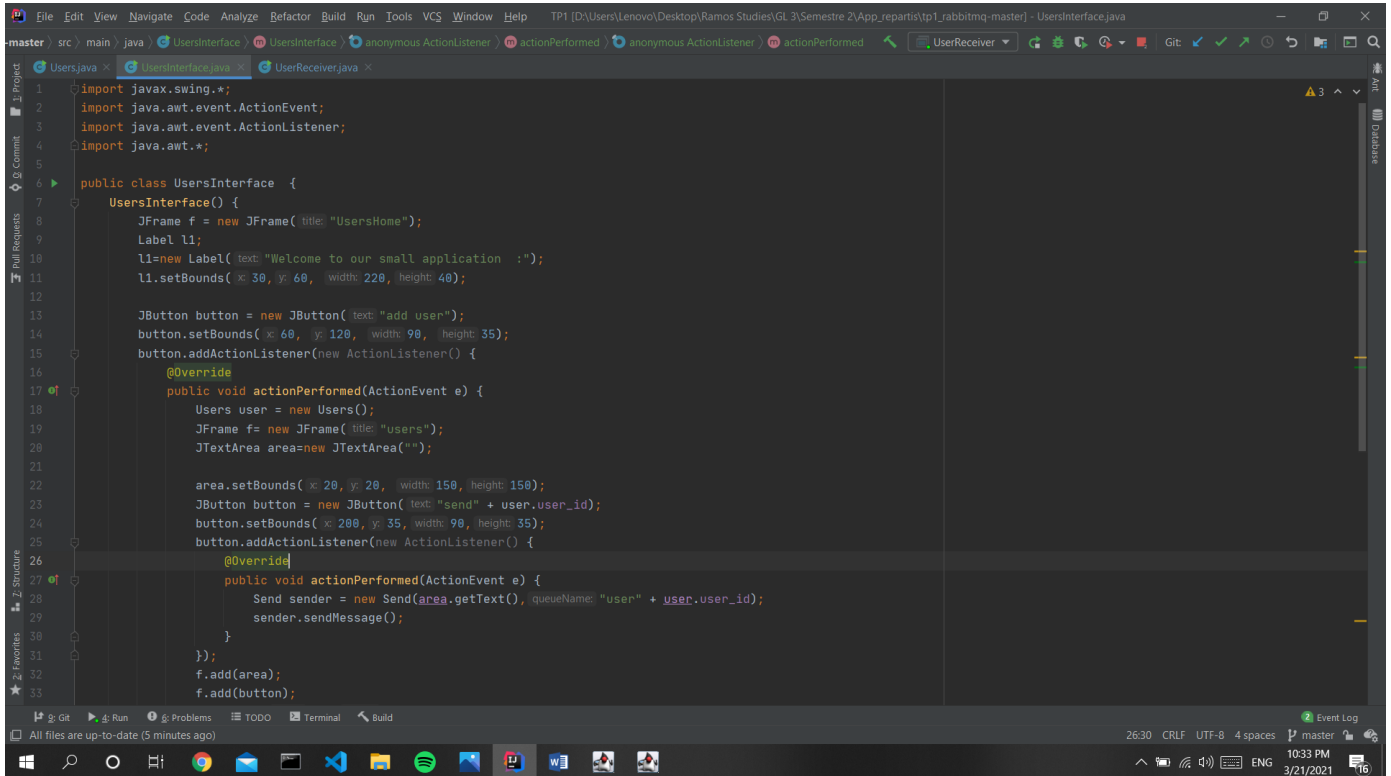
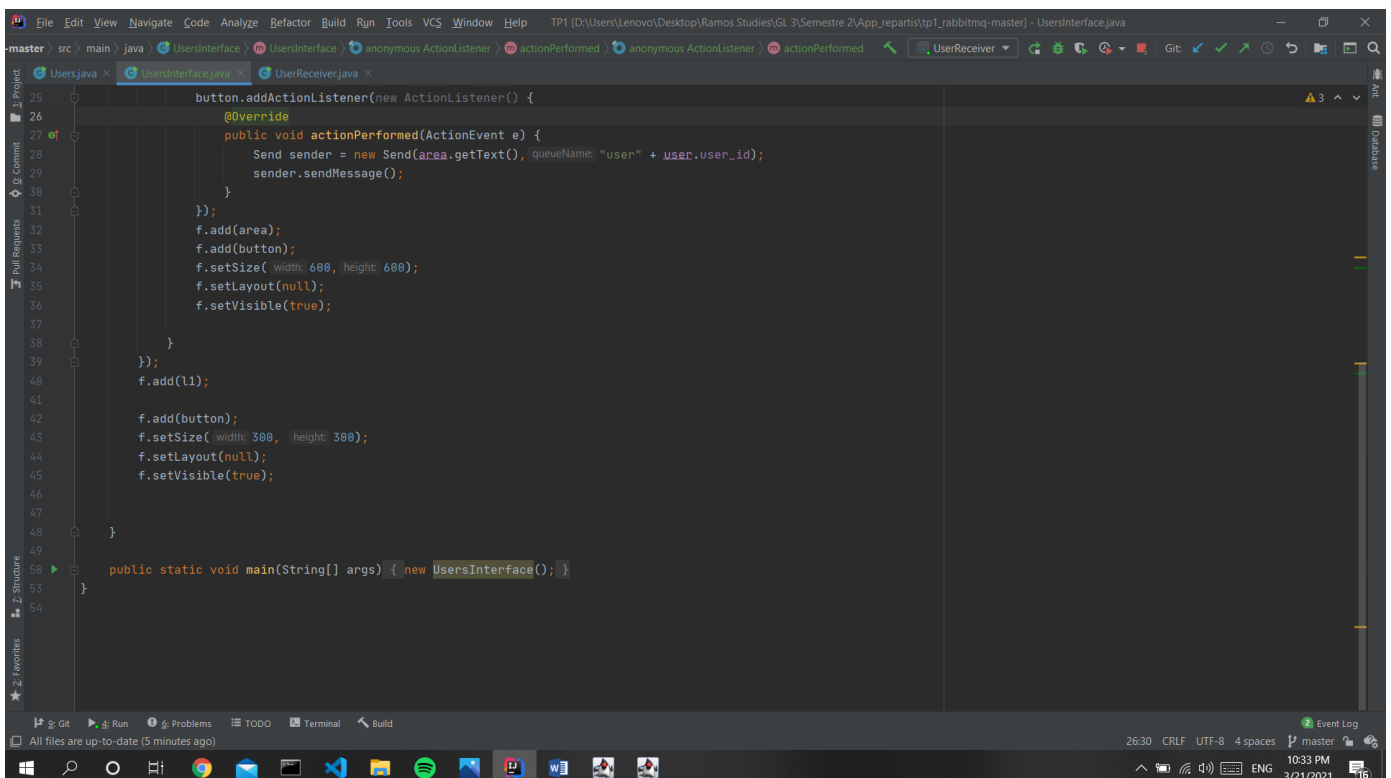## Année universitaire :

2020/2021

1)

3)

- ## User.java :

- ## UsersInterface.java :

```java
import javax.swing.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.*;

public class UsersInterface  {
    UsersInterface() {
        JFrame f = new JFrame( title: "UsersHome");
        Label l1;
        l1=new Label( text: "Welcome to our small application  :");
        l1.setBounds( x: 30, y: 60,  width: 220, height: 40);

        JButton button = new JButton( text: "add user");
        button.setBounds( x: 60, y: 120,  width: 90,  height: 35);
        button.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                Users user = new Users();
                JFrame f= new JFrame( title: "users");
                JTextArea area=new JTextArea("");

                area.setBounds( x: 20, y: 20,  width: 150, height: 150);
                JButton button = new JButton( text: "send" + user.user_id);
                button.setBounds( x: 200, y: 35,  width: 90, height: 35);
                button.addActionListener(new ActionListener() {
                    @Override
                    public void actionPerformed(ActionEvent e) {
                        Send sender = new Send(area.getText(), queueName: "user" + user.user_id);
                        sender.sendMessage();
                    }
                });
                f.add(area);
                f.add(button);
                f.setSize( width: 600, height: 600);
                f.setLayout(null);
                f.setVisible(true);


            }
        });
        f.add(l1);

        f.add(button);
        f.setSize( width: 300, height: 300);
        f.setLayout(null);
        f.setVisible(true);


    }


    public static void main(String[] args) { new UsersInterface(); }
}
```
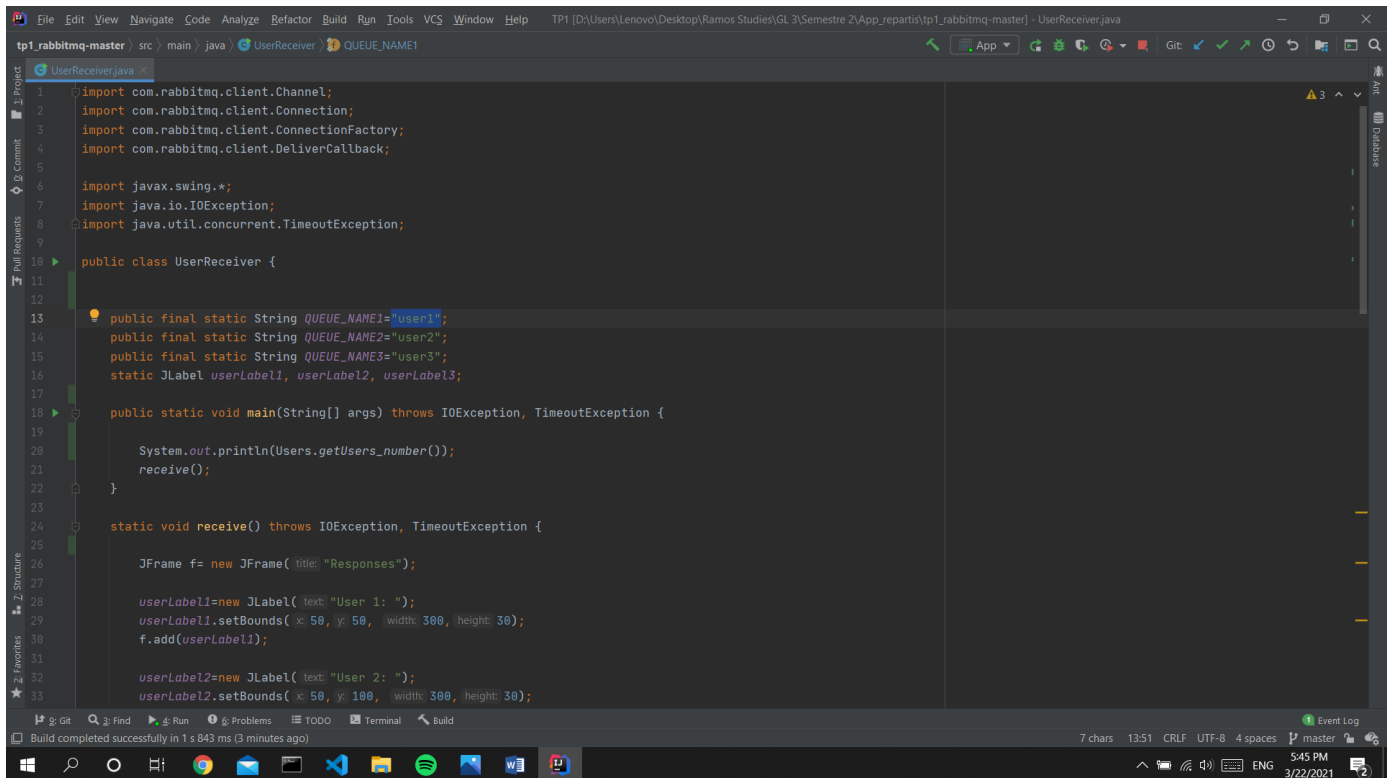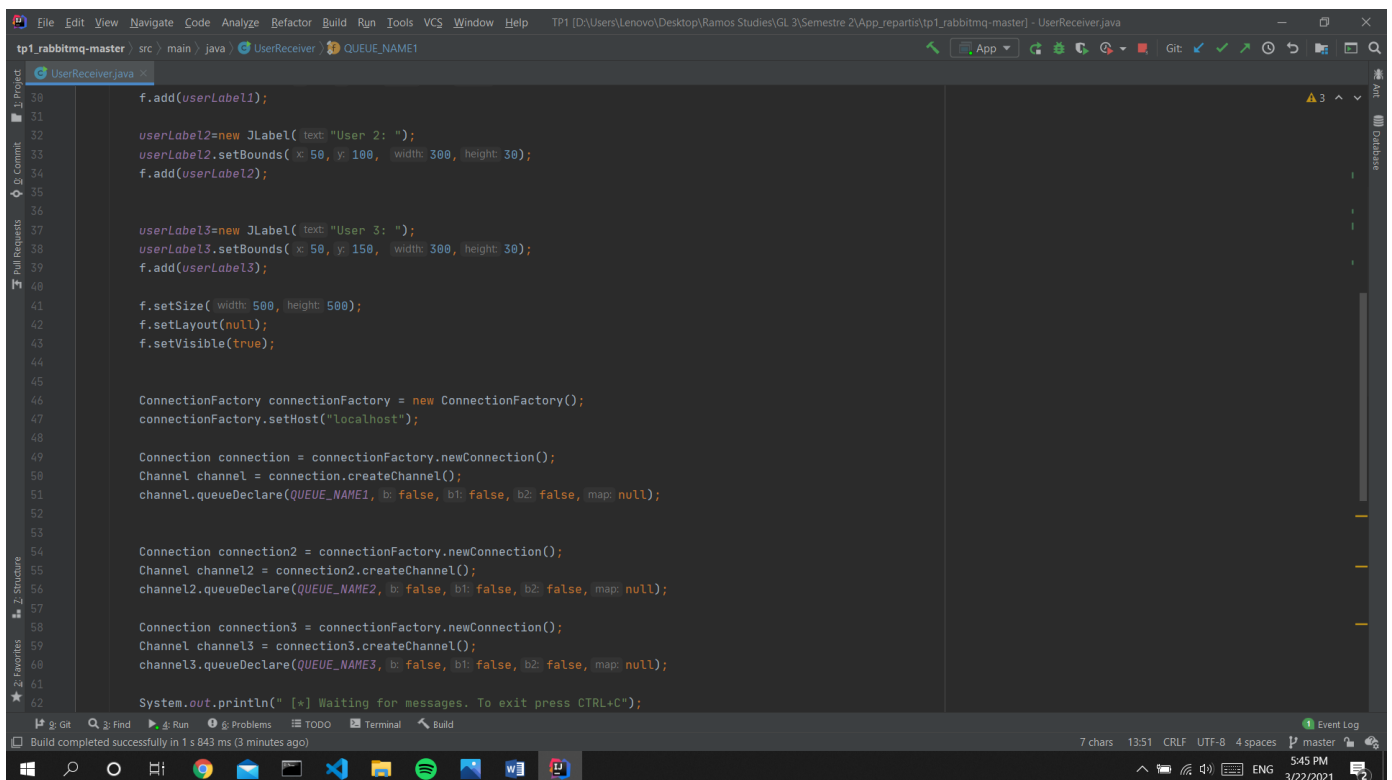
- **UseReceiver.java :**


```java
import com.rabbitmq.client.Channel;
import com.rabbitmq.client.Connection;
import com.rabbitmq.client.ConnectionFactory;
import com.rabbitmq.client.DeliverCallback;

import javax.swing.*;
import java.io.IOException;
import java.util.concurrent.TimeoutException;

public class UserReceiver {

    public final static String QUEUE_NAME1="user1";
    public final static String QUEUE_NAME2="user2";
    public final static String QUEUE_NAME3="user3";
    static JLabel userLabel1, userLabel2, userLabel3;

    public static void main(String[] args) throws IOException, TimeoutException {

        System.out.println(Users.getUsers_number());
        receive();
    }

    static void receive() throws IOException, TimeoutException {

        JFrame f= new JFrame( title: "Responses");

        userLabel1=new JLabel( text: "User 1: ");
        userLabel1.setBounds( x: 50, y: 50,  width: 300, height: 30);
        f.add(userLabel1);

        userLabel2=new JLabel( text: "User 2: ");
        userLabel2.setBounds( x: 50, y: 100,  width: 300, height: 30);
```



```java
        f.add(userLabel1);

        userLabel2=new JLabel( text: "User 2: ");
        userLabel2.setBounds( x: 50, y: 100,  width: 300, height: 30);
        f.add(userLabel2);

        userLabel3=new JLabel( text: "User 3: ");
        userLabel3.setBounds( x: 50, y: 150,  width: 300, height: 30);
        f.add(userLabel3);

        f.setSize( width: 500, height: 500);
        f.setLayout(null);
        f.setVisible(true);

        ConnectionFactory connectionFactory = new ConnectionFactory();
        connectionFactory.setHost("localhost");

        Connection connection = connectionFactory.newConnection();
        Channel channel = connection.createChannel();
        channel.queueDeclare(QUEUE_NAME1, b: false, b1: false, b2: false, map: null);

        Connection connection2 = connectionFactory.newConnection();
        Channel channel2 = connection2.createChannel();
        channel2.queueDeclare(QUEUE_NAME2, b: false, b1: false, b2: false, map: null);

        Connection connection3 = connectionFactory.newConnection();
        Channel channel3 = connection3.createChannel();
        channel3.queueDeclare(QUEUE_NAME3, b: false, b1: false, b2: false, map: null);

        System.out.println(" [*] Waiting for messages. To exit press CTRL+C");
```

6) conception des files :