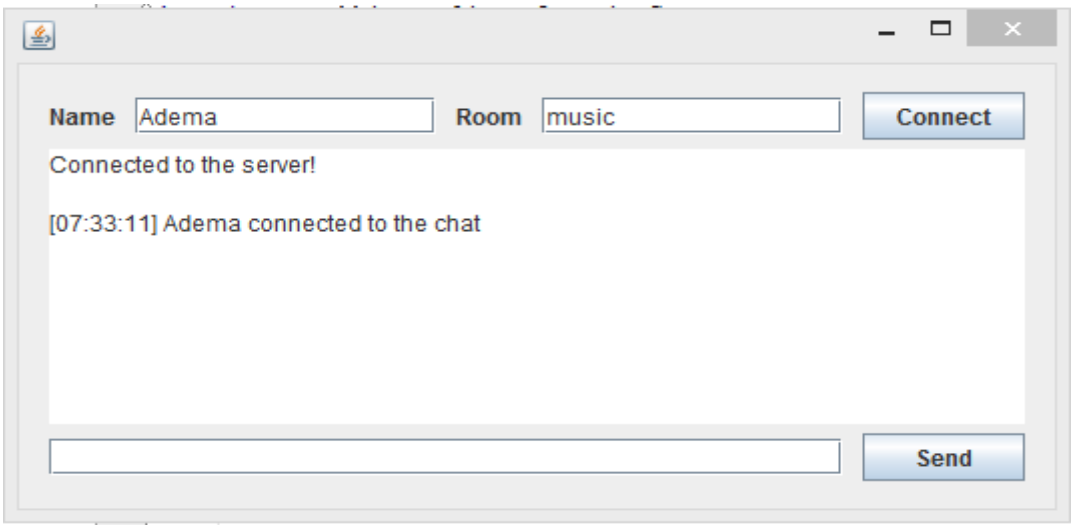
For this project I chose the server-based chat application, supporting multiple users and rooms

As in laboratory №5 I worked with RabbitMQ program. So I tried to download RabbitMQ program, but there were many problems. Therefore I used RabbitMQ online service, which accepts and forwards messages and downloaded the library package. I used all needed information from the web site http://www.rabbitmq.com/tutorials/tutorial-six-java.html, then I just combined needed tutorials especially RPC, work queues and Publish tutorials.

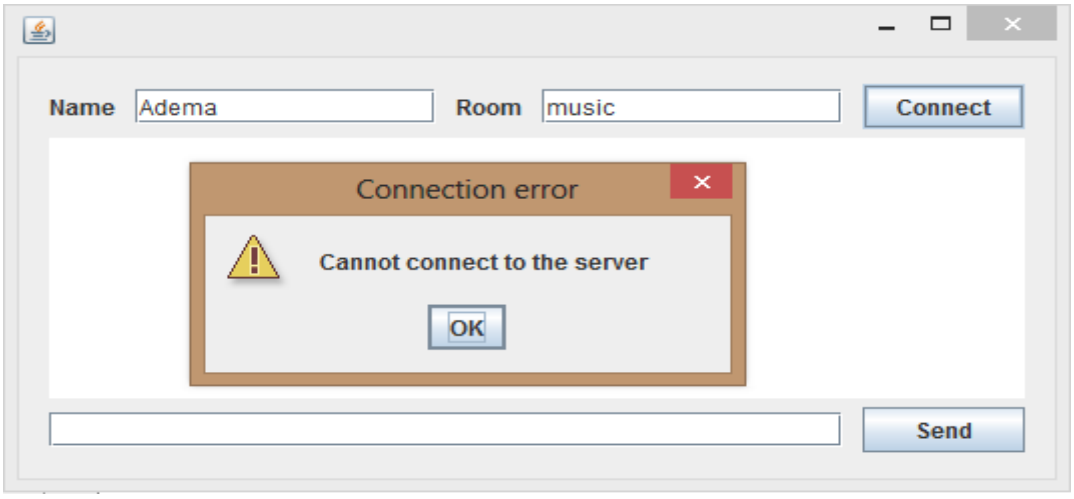
I used JFrame to make the GUI:



As you can see above, it consists of the listed text fields:

- Name
- Room
- Your message

Also it contains button CONNECT that will connect you to the server. If you do not have to access to net, you will get an error, like this:




```
catch (Exception e) {
```

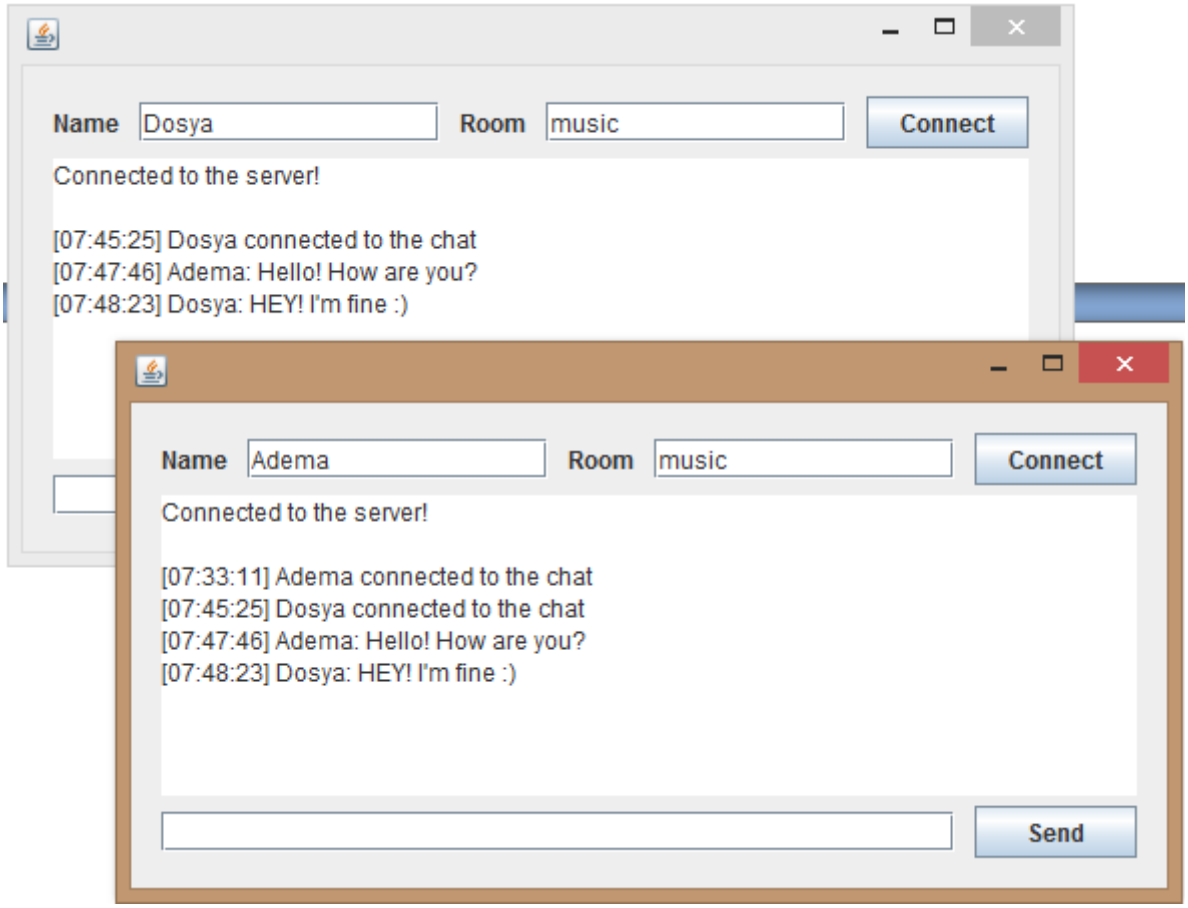
JOptionPane.showMessageDialog(null, "Cannot send a message to the server", "Connection error", JOptionPane.WARNING_MESSAGE);

When you send the message, there will be displayed the time when the person was connected to the room named ‘music’, the information about the message, its time and by whom it was sent.

String date = new SimpleDateFormat("[hh:mm:ss]").format(new Date());

messageToSend = date + " " + name + ": " + textFieldMessage.getText();

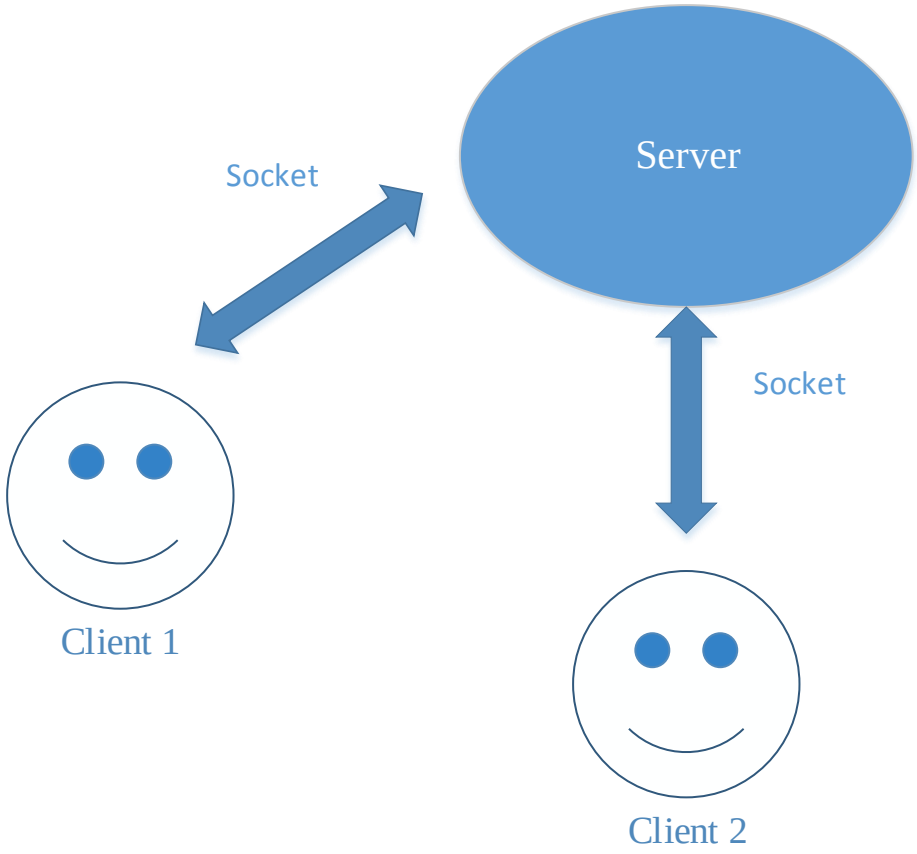
These messages can be presented to people that are in this room.



textFieldMessage.setText(""); // Clear the field to text the message

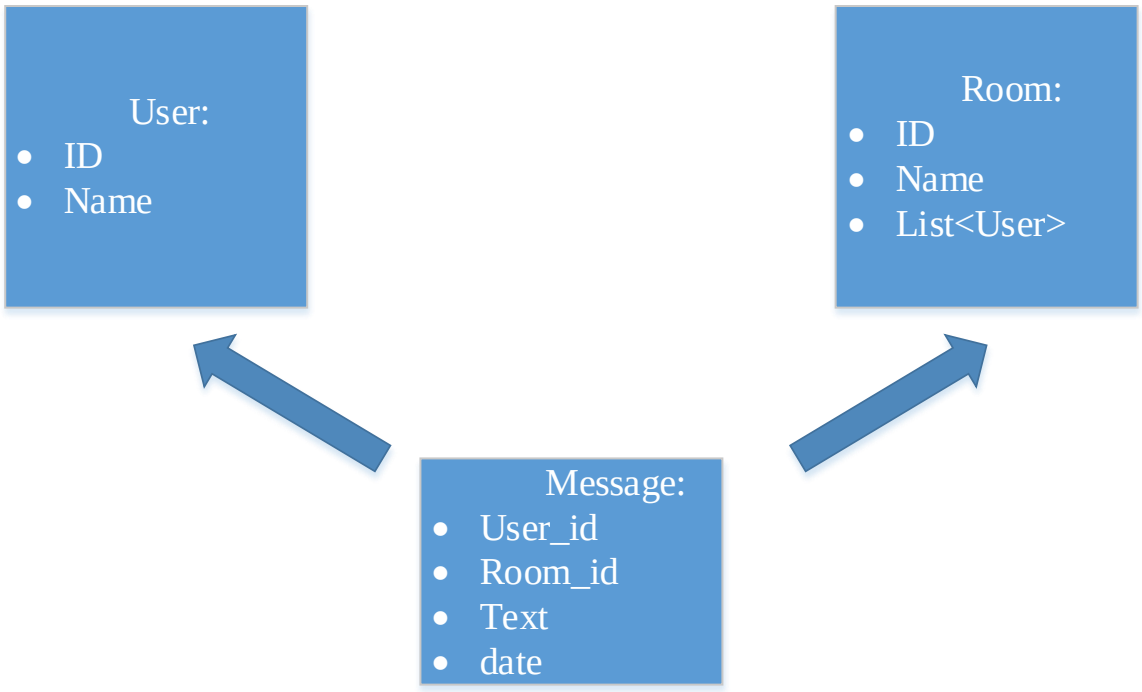
Send the message to the server:

```
ConnectionFactory factory = new ConnectionFactory();

factory.setUri(Constants.uri);

Connection connection = factory.newConnection();

Channel channel = connection.createChannel();

channel.exchangeDeclare(room, "fanout");

channel.basicPublish(room, "", null, messageToSend.getBytes());
```



Connection to server:

```
ConnectionFactory factory = new ConnectionFactory();
factory.setUri(Constants.uri);
Connection connection = factory.newConnection();
Channel channel = connection.createChannel();
channel.exchangeDeclare(room, "fanout");
String queueName = channel.queueDeclare().getQueue();
channel.queueBind(queueName, room, "");
textArea.append("Connected to the server!\n\n");

QueueingConsumer consumer = new QueueingConsumer(channel);
channel.basicConsume(queueName, true, consumer);
```



As you can see, people that are in exact room can send and receive messages, communicating with each other. Due to working through the free online server of RabbitMQ, I can open only 3 clients.