**NPR Mid-Term Project Report**

*Two-side chat application using TCP java socket*

|  |  |
| --- | --- |
| Vũ Ngọc Hà Minh | 1801040149 |
| Trần Thị Ngọc Ánh | 1801040017 |

1. **Project description:**

In this project, we develop the two applications which using TCP/IP socket method to complete following objectives:

* Create the project with Apache NetBeans IDE
* Create server with the favorite port
* Allow client access to server address.
* Connect and allow chatting with the server
* Support the User Interface: ClientChat and ServerChat

Diagram

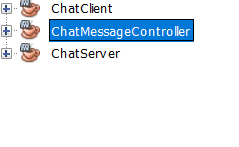
Description automatically generated

*Figure 1: The diagram of how client and server interact through the message controller*

1. **Implement**

Our project has 2 main GUI: ServerGui & ClientGui

We used JFrame Form in Apache NetBeans to create the form of chat application. Some components that we used: JTextPane to create a board to display the message content, Text Field allows the user to enter data such as entering a Serverhost, or entering a message, Button to listen for click button events; label...



*Figure 2: Our project code structure*

In the ChatMessageController, we build a contructure for both the server and the client. This class manages the connection information, and allows receiving and sending messages through socket. We used PrintWriter to send data and BufferedReader to receive and read the data through Socket. This class also includes two functions: send(String message) to handle sending and received() displaying all chat content.

In ChatServer and ChatClient contain JFrame Form java classes: ChatServerForm.java and ChatClientForm.java, respectively. In each class will contain the “Design” part to create the chat form and the “Source” contain code to handle the events performed in the chat app such as btnSendActionPerformed (), btnConnectActionPerformed (), txtMessageActionPerformed (),...In Dependencies of both ChatClient and ChatServer, we added ChatMessageController so that client and server side use the same contructure.

* 1. **ServerGUI**

ServerGUI is used to set up server and Port:

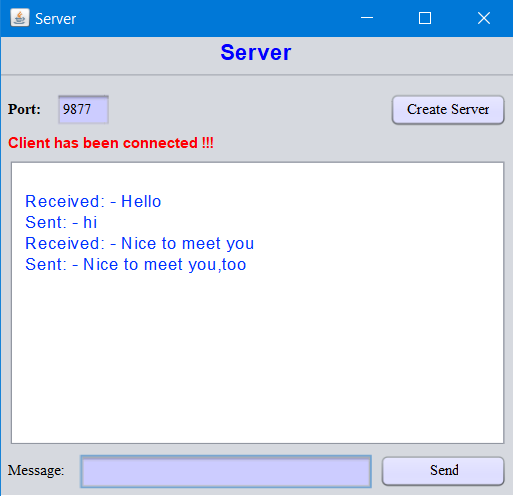
* In ServerGUI, it has a button “Create Server” which to start server. We can provide a customized port for the server (the default: 9877)
* The status with red color is informing the server status

Graphical user interface, application

Description automatically generated Graphical user interface, application, Word

Description automatically generated

*Figure 2: The Server with GUI Figure 3: Server has been started & waited for client*



*Figure 4: Server chat GUI*

* 1. **ClientGUI**

ClientGUI display the box chat for the user to chat with the server freely.

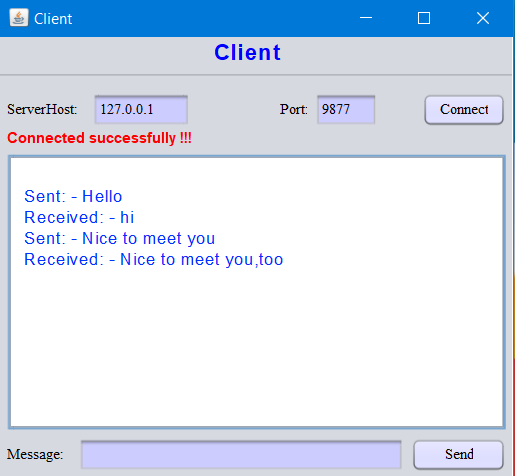
* Send message privately to the server.
* Users provide the server host and port to connect to the server (default: localhost & port: 9877).
* Status in red color show the status of client

**Graphical user interface, text, application, Word

Description automatically generated** Graphical user interface, text, application

Description automatically generated

*Figure 5: Client GUI Figure 6: Client when successfully connect to server*



*Figure 7: Client chat GUI*

When the Server is not created, The Client cannot connect, if user try to press the connect button, the GUI will show a message for warning that Server is not starting

Graphical user interface, application

Description automatically generated

*Figure 8: Client GUI inform Server is not started yet*

1. **Conclusion**

The application has been successfully built allowing two people to send and receive messages. This application will be developed further in the future, with the desire to create more functions such as creating chat groups, allowing to send icons and attachments....

After all, it can be seen that Socket and Thread in Java are crucial and well-supported over the year. As a result, this is the information that students and professionals who wish to work in the programming sector in general should study and teach. Overall, even if this project is only for annotation, it demonstrates the use of Java Socket and Thread in developing that.