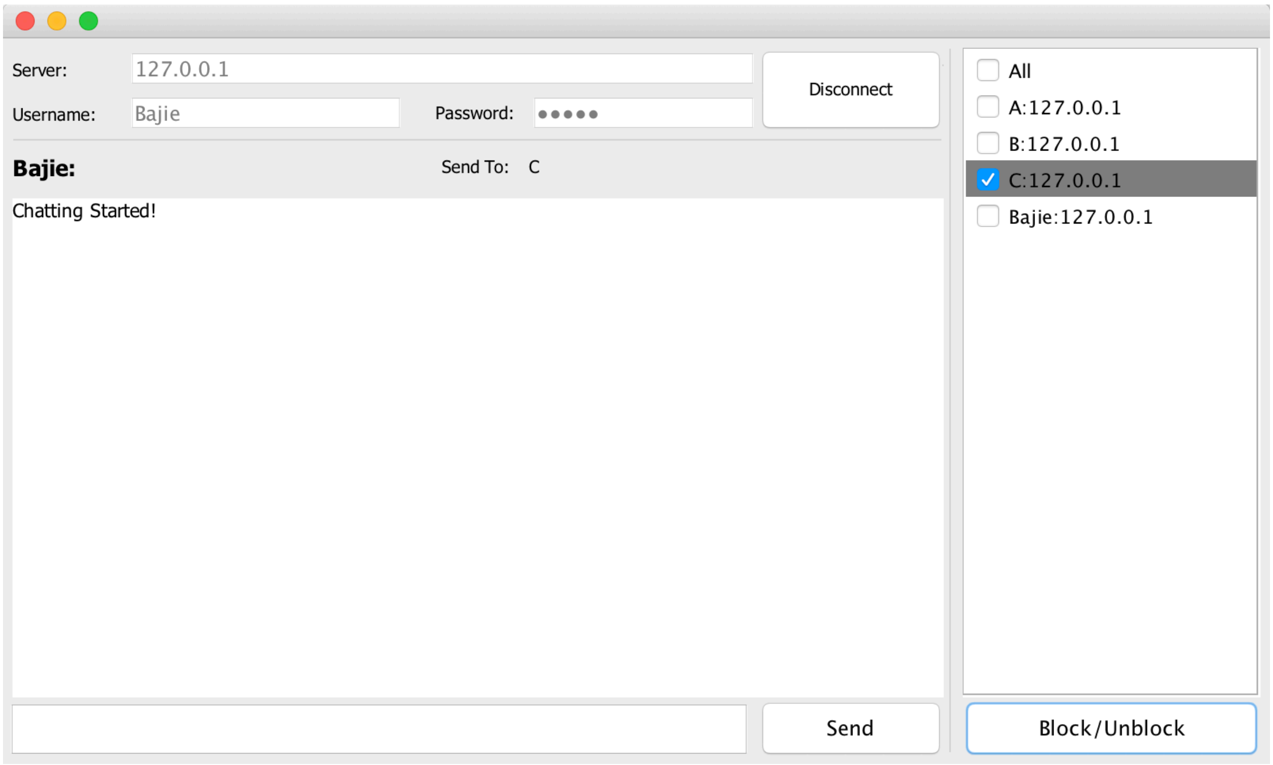
Description

These Chat Client and Chat Server met the Basic Requirements and five Extensions.



Server: 127.0.0.1

Username: Bajie

Password: •••••

Bajie: Send To: C

Chatting Started!

□ All

□ A:127.0.0.1

□ B:127.0.0.1

☑ C:127.0.0.1

□ Bajie:127.0.0.1

1. USAGE

- Unzip then load the project folder into Eclipse.
- Run “ChatServer.java” to start Server, please make sure that the 5000 port is not being used.
- Run some copy of “ChatGui.java” so you can send message to each other.
- Input the correct Server IP into Client, by default the server and client running on the same machine, the IP address already set up for you.

- Test Account:

Usename	Password
A	A
B	B
C	C
Bajie	B1234

- Enter username and password then “Connect”.
- Select the person or All to send message.
- Disconnect when you want to leave, you can also close the window directory. The Server will update the other that you already leaved.

2. BASIC REQUIREMENTS

2.1 Client:

- send a basic text message to the server
- receive text from the server

2.2 Server:

- receive text from a client
- broadcast a text message received from a client to all the other clients connected to the server.

3. EXTENSIONS

3.1. Username/password logon:

Design:

The login and connect area locate at the top left.

When account successfully login the login area Test Field will be set to disable, at the same time, the Connect Button text is set to be “Disconnect” waiting for user to disconnect.

3.2. Display IP address/nickname of other chat clients

All user’s IP address are attached behind their Username in the JList.

3.3. Private messaging between users

Design:

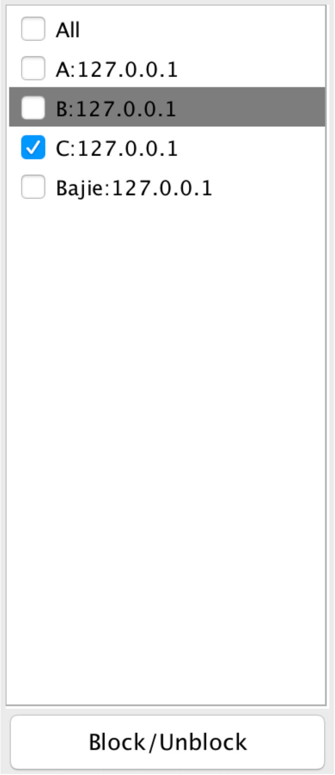
By select a contact from the User List, the message will only be delivered to that person.

Implementation:

When sending a private message, the first msg being sent to server is of type: CHAT_MESSAGE. The second msg is the sender name, the third msg is message itself.

The corresponding ClientHandler Thread maintain a total Client List will forward it to the right person.

3.4. Block/ignore user



□ All
□ A:127.0.0.1
□ B:127.0.0.1
☑ C:127.0.0.1
□ Bajie:127.0.0.1

Design:

When chose a user and click “Block/Unblock” button, this user will be block in this client.

Right after the click, the background color of Blocked-User cell is set to Color Gray.

If user A is block by user B, A will not received any notice about this blocking, this is for protecting B’s privacy.

Implementation:

Each Client maintain two BlockUser ArrayList variables, one is the index of block-user in JList which is used to highlight cell, another ArrayList restore the block-user name for checking the incoming msg.

3.5. Enhanced GUI JList

Design:

A checkbox is added to JList Cell to help distinguish select user and blocked user. The Checkbox always represent the current chatting friend.

One Client can block as many users as it want.

Implementation:

A CheckboxListCellRenderer class is use to render JList cells, this object is help render different cell background for the blocked users.

4. DOCUMENT DESCRIPTION

- ChatServer.java: The entry of the Server Application, run this app to start server
- ChatGui.java: The entry point of Chat Client, run some of these app to start chatting.
- ClientHandler.java: One ClientHandler thread serve one client.
- readByLine.java: Read registered user data from UserModel.txt (Restore Username and password)
- ServerConstants.java: Restore the constant variables which are used to defined the message type.
- run.sh is a bash script to recompile and run the server app.