Project 3

Released: March 28th

Wireframes and Class Diagrams Due: Thursday April 11th on Gradescope

Final Submission Due: Thursday, April 18th at 5pm on Blackboard

You may work individually or in pairs for this project

1 Introduction

For this project you will be extending on the simple GUI Client / Server messaging app to include some additional features. The starter code provided involves a client and a server GUI which then spawns networking threads to handle communications over the network as appropriate. In the starter code, only Strings are passed over the network and all strings are passed to all Clients. You will change this to pass a Serializable class called a Message which will allow Clients to create groups, send messages to all members of a group and send messages to individual other clients.

1.1 Getting Started

Download the two starter code projects from blackboard. One will be your server code and one will be your client code. As part of each of these, there should be three java files.

a) `GUIClient.java / GUIServer.java` The GUI code for either your client or server. This is also the main file that should be executed with a run to `mvn clean compile exec:java`. These files will spawn all other code.

b) `Client.java / Server.java` The code that will handle network communication for either a client or a server. Code is spawned by the GUI code and a thread should be created for each network communication that could occur. Test this starter code and verify you understand how each piece of code will work

c) `Message.java` This is the serializable class stub that you will need to implement. This object should contain all information that is sent over the `Socket`. You should **only** send Message objects over the network. The starter code sends Strings over the network and this will need to be modified. This class is also how the server will know which clients to which each message should be sent.

2 Deliverables

Your submission will occur in two phases. First, wireframes and class diagrams will be due April 11th to gradescope and final code will be deliverable to blackboard on April 18th.

2.1 Wireframes and Class Diagrams

Similar to Project 2, you will submit wireframes for your Client application to gradescope by April 11th. The February 27th lecture discusses wireframes and good design principles. You should sketch out what each of the graphical components for your Client app are for each scene. Simple boxes are sufficient for GUI components in the app. You may draw your wireframes digitally or by hand. You should have one wireframe sketch for each scene and you should indicate how a user transitions from one scene to another. Each scene should also have a one sentence description. Additionally, you should include UML class diagrams for both the client and server code and their threads and include this in your submission to gradescope. Consult the slides for 14.diagrams for an idea of what should be included **The wireframes and class diagrams are due to gradescope on April 11th at 5pm. Submit these as a group to gradescope**.

2.2 Code deliverables

Your submission should be a zip file containing the two maven projects, Project3Server.zip and Project3Client.zip. Make the following changes. **The Group EC assignments are worth 20% of your Project3 grade**

- In the starter code, all objects passed over the network are Strings. You should modify both Client and Server communication through the socket so it accepts Messages. As you implement other parts of the project, your Message will need to have more attributes and functionality.
- The ServerGUI code should print a digestible version of each message to the ListView as a log. The biggest changes will be to the `Platform.runLater(...)` commands.
- The Server code should be modified to allow communication of Messages through the input and output streams rather than Strings. It also needs to be extended to keep track of which groups currently exist and what the username of each client is.
- The ClientGUI code should be modified to allow the following operations:

  - Allow the user to pick a username when joining the server. The client should receive an error message if the name is already taken and be prompted to provide a new name. Once the client's username is chosen, it should not be changable.
  - View all users that are currently connected to the server
  - Send a message either to all users or to an individual user.
  - View all messages sent to the user.
  - **Optional:** View all groups that currently exist on the server
  - **Optional:** Create a new group from a list of users.
  - **Optional:** Send a message to a group

- The Client.java code will need to be modified to allow these operations.
- The Message code will need to include all information that may be sent over the network. The message needs to be able to: create a new user ID, create a group, send a message to all clients, a group of clients or an individual. You may add additional support as necessary. Both the client and server folders must have a matching version of the Message.java code to function.

3 Submitting your work

Before submitting, you should run a `mvn clean` build so that there are no unnecessary `.class` files sent with your submission. Create a zip file of both the `Project3Client` and `Project3Server` directory and submit the zipped file to blackboard. **If you are working in pairs, please submit the form to indicate who your partner is.**