Project: Register/Log on/Log off

Points:

Design: 20 points
Implementation: 20 points
Meeting: 10 points

due date:

- Wednesday, October 5 – Class and Sequence Diagrams
- Friday, October 14 - Implementation meeting

You will need to implement a separate project for this part of the project that will be your server. Then you will connect your GUI and your server using sockets. You will only be implementing these pieces of the project at this time.

- **registration**
- **logging on**
- **logging off**

You will have 2 separate projects in Netbeans – one for the server and one for the client.

Design requirements (for Wednesday the 5th).

You have already designed your GUI for this part of the project. Your design is for the server side of this project. Remember that – you are only designing the server.

Class Diagram

You are going to expand and shrink the initial class diagram that you will be submitting based on my comments on your initial class diagram. You will focus only on the part of the class diagram related to registering a new user, logging on a user, and logging off a user. You should produce a new class diagram that has only those parts (that is the shrinking). To this class diagram you need to add the methods that will be included in the classes. **Try not to over complicate this piece. It should not be hard or that extensive**. Please note these requirements for your class diagram:

- You must include any of Java's collection classes that you use in your project (LinkedList, ArrayList, HashMap, etc.) in the class diagram. You do NOT have to include any other Java classes such as Scanner or Random.
- For the classes that you will be writing, you must include your best estimation of what instance variables you will have. You should also include any methods you have listed in the sequence diagrams.
- For Java's collection classes you only need to include the class name. You do not need to include any instance variables or methods in the diagram.

Sequence Diagrams

Your second task in designing the server is to develop sequence diagrams for each of these three use cases listed above (Registration/Log on/Log off). Your sequence diagram is only for the server side of the system. The starting point of the diagrams will not be an actionPerformed method on the GUI. It will be something on the server that is communicating with the GUI – this will likely either be the main method of the server or a method that gets called from the main method.

Make sure your sequence diagrams and class diagram are consistent. If you indicated that a method will be used in your sequence diagrams, but it doesn't appear in your class diagram then you will lose points.

Your documents must be electronic!

Submit a printed copy of the class diagram and the sequence diagrams.

Implementation

The implementation which is due on October 14 should connect your GUI to the server. Your code should handle exceptions such as trying to log on with a bad password. At your meeting with me I will try "exceptional" cases to see if I can crash either your client or your server. I am not going to tell you which cases I will try because I want you to think through what cases are possible. However, if you ask me about a specific case I will tell you if your code needs to handle that case.

At this time you should be able to run both your client and your server on one computer and just use "localhost" for the ip address.