Project: Notifications and Private Messaging

Administrative Access

Points:

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

due dates:

- Monday, November 21 – Class and Sequence Diagrams, Protocol and GUI picture(s)
- Monday, December 5 - Implementation meeting

You will need to implement these two pieces of the project at this stage:

- **At least one server notification or online message delivery**
- **Private Messaging**

Private Messaging

A private message should go directly to the other user and not go through the server first. In order to send a private message **a user needs to know if a user is online and will also need to know the IP address of the machine they are logged in from**. You need to provide a way to check with the server to determine if a user is online or not. Determining if a user is online and retrieving their IP address can be done in at least 2 ways (probably more):

- Connect to the server when the user wants to send a private message and see if they are online and get their IP address from the server then
- Have the server notify the client and send them a notification with the IP address whenever someone logs in

Note that you will need get and save a user's IP address when they login and save it on the server! You can get this address from the socket.

When you have the IP address you will then connect to the listener (a thread) on the other client and send the private message.

You may decide if there are rules for who can send private messages to each other. Can a user send a private message to any other user? Only to people they are following or people who are following them?

You can implement this like email or like a chat system that is up to you.

Notifications

You must have your server notify clients who are online in at least one way. You can implement more of these if you wish. This will involve the server noticing that a certain event has happened and then opening a socket connection to a client and sending them a message. Some possible notifications include:

- Notify a user when someone they are following has posted a public message. You could just send a notification and then it is up to the user to retrieve that message or you could also send the message and update their feed automatically.
- Notify a user when someone they are following has logged in (or logs out). Or when any user logs in or out. You may want to also send their ip address to the client.
- Notify a user when someone new follows them or someone unfollows them

It is required that you save notifications on the server when a user is logged out and then send those notifications when the user logs in. The penalty for not implementing this, however, will be minimal.

You need to decide how to present the notifications to the user. You could use a JOptionPane showMessage method (simple), you could do what many applications do and have an icon in the corner of your application indicate how many notifications there are (harder), or you could do something else.

Client Thread

Your client code will need to implement a thread that will need to listen for private messages from other users and also whatever notifications you will have the server send. The code for the run method of your thread will look like the code for the main method of your server.

Where should this thread be started? What parts of the GUI does it need to have access to?

DESIGN REQUIREMENTS (for Monday, November 22).

Class Diagram

Draw a class diagram of your CLIENT. Include in the diagram any JFrame, JDialog, and any class that you wrote or will write for this phase of the project (including your class that will implement runnable and be your thread). You do NOT need to include any GUI components that are your frames and dialogs.

Indicate which class will start the thread to start listening.

Briefly explain how you will set it up so that the thread gets the private message to the appropriate place to display the private message and how it will present a notification.

Briefly explain how you will shut the thread down when a user logs out.

Update the class diagram of the SERVER to show how notifications will be saved if the user is not logged on when an event occurs.

Sequence Diagrams

For one of the notifications that you will have the server send, give an updated sequence diagram for that action that now includes the server finding the users it needs to notify and sending the notification.

If you will be implementing multiple notifications you may submit multiple sequence diagrams but only one is required.

Protocols

Give the protocol that **two clients** will follow when a private message is sent.

GUI Picture

Draw a picture (or implement) of your GUI that shows how private messaging will be accomplished and how notifications will appear on the client.

Administrative Access

It is required at this point that you implement administrative access, however, you will only lose 1 point on the implementation if you do nothing. You will lose ½ a point if you implement any part of this, but not everything.

For administrative access you should have a separate account. That account could also be a user and do user things if you want, but it could be a separate account that does only does administrative tasks.

You can try to build this into your current GUI, but it may be easier to simply create a separate project for the administrative GUI. Also, you don't have to provide the ability to create and register an administrative account. You can have one hardcoded into the system.

These are the tasks that an administrator needs to be able to do:

- Shut down the server
- Change the password for a user
- Delete a public message sent by a specific user with a specific hashtag. This would delete all messages sent by the user that have the hashtag. Messages sent by other users with the same hashtag should not be deleted.