Chat System Project

Background

This week, you are starting your team-project on the chat system. We are going to develop the chat system in a sequence of iterations - each iteration will last 2 weeks. By the end of the quarter we will have a functional (and usable) chat system with some of the features you find in well-known systems such as IRC, Yahoo chat, or Google chat. We will not pursue compatibility with any of these.

You will follow the HFSD agile process in developing the chat system. You will deliver not only the working software, but also documentation such as user stories, tasks, and design documents.

Dates

Your project development will have 3 iterations and 2 milestones.

Saturday 10/26	Project officially starts	Start working on user stories
End of Week 5	First meeting with your TA	Prioritize requirements and decide on Milestone 1 stories
End of Week 6	End of iteration 1	Present Milestone 1 to the TA Prioritize requirements for iteration 2. Decide on user stories for Milestone 2
End of Week 8	End of iteration 2	Send your code/documentation to TA. Adjust expectations as needed for Milestone 2
End of Week 10	End of iteration 3	Present Milestone 2 to the TA

You will be graded twice, when you present Milestone 1 and Milestone 2.

Requirements

To get started we will give you a set of rough requirements. Your job as a teams, is to refine them and obtain user stories and tasks with estimates. Each group will interact with one TA that will act as the customer. By the end of next week you will meet with your customer and prioritize user stories deciding what to develop for Milestone 1. Your job is to come up with a set of user stories and tasks. As you know, to do so you are supposed to interact with customer. To this end, please sent piazza messages to ask clarifications and remove assumptions.

Communication with the TAs. Please send messages to TAs as teams (title of the message "TEAM X Request") and discuss the message with the rest of your team first. Try to write clear requests that can help you in creating your user stories. We will put a limit on the number of requests per team of 5 requests per week. This is to simulate the real world, in with the customer is busy and you need to use his time carefully.

High level view

Typical chat systems consist of a server and a client part. Note that we distinguish between the user (somebody who wants to chat), and the client (the piece of software that allows the user to interact with others).

The server is used to manage all the clients and users who want to communicate with each other. In particular, the server maintains data structures in which it records users that sign on and off, and will route messages so that they reach all intended recipients.

The client part allows a user to communicate with other users - via the server. In particular, the client will allow the user to issue commands to the server, including sign on/sign off and sending messages. To that end, the client offers a user interface (textual/command line, or graphical) to the user by which the user can issue commands to both client and server. There will be commands specific to the client (connecting to a particular server, terminating the client, etc.) and commands specific to the server (sign on/off, send message, etc.).

Requirements list

1. Client connects to a specified server
2. User signs on to the server the client is connected to
3. User signs off from the server the client is connected to
4. User sends a message to all currently signed-on users (Broadcast)
5. Client quits. The Client application is terminated.
6. Sending text to a specific subset of signed on users (instead of a broadcast message.)
7. Listing all users signed on to your server.
8. User joins chat room and can send text messages to the chat room, which causes all other users joined to the chat room to receive the message.
9. User leaves chat room, which removes user from list of users receiving messages for the chat room.
10. User lists all chat rooms that exist.
11. User joined to a chat room lists all users also joined to the chat room.
12. Provide both a textual interface and a graphical interface for your client.

NOTE: As you can see this list is not a proper requirement list or a set of user stories. You will need to make assumptions on what is wanted by your customer. Then you will need to contact the customer and remove the assumptions.

Technical constraint

Your customer not only hands out a set of requirements. He also has a few technical constraints that you will need to follow in your development.

1. Use Spring to structure your server (you can use it also for the client but this is optional).
2. Use ActiveMQ for communication between clients and server.

A good example to get started is the code of Lab 4. It is a simple client server system that sends text strings from the client and print them on the server. While its functionalities are limited and do not implement all requirements needed, for example communication is unidirectional, it is a starting point.