

## **CS 375 Networks and Database Systems**

### **Assignment 3**

#### **Chat Room Web Application and Applet**

Implement a new version of the chat-room program that runs on the web. The server will be implemented as a Java servlet. The client program will be implemented as a Java applet. The two programs will use HTTP to communicate with each other.

**Server:** The server will support four commands, implemented as HTTP queries.

**Connect:** The connect command takes one parameter: a handle string representing the name of the user who wishes to chat. The server generates a unique numerical ID string for the client and stores the client's handle in a hash map indexed by the client's ID. The server also stores the string "<Handle> has entered the chat room." in an ordered container of all messages. The server responds by sending the client's ID string to the client.

**Send:** The send command takes two parameters: the client's ID number string; and the message string. The server stores the message in the message container, and returns to the client a string acknowledging receipt of the message.

**Receive:** The receive command takes one parameter: the client's ID number string. The server returns a string composed of all the messages in the message container that the client has not yet seen, including his/her own messages sent to the server.

**Disconnect:** The disconnect command takes one parameter: the client's ID number string. The server adds the message: "<Handle> has left the chat room" to the message container. The server also returns the string: "Disconnect request confirmed." to the client.

**Client:** The client sends commands to the server and gets back responses from the server. The commands are encoded as HTTP queries, as described above. The client has a user interface that acquires the user's handle (name) and allows the user to Connect, Send, Receive and Disconnect to/from the server. The client should display the responses of Connect, Send and Disconnect requests in a text area (labeled "Response:") in the client's user interface. The client also has a thread that runs continuously until the user Disconnects from the server. The thread sends a Receive request every N seconds and appends the new messages to a text area (labeled "Chat:") in the client's user interface.