

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 a combination of HTTP and TCP communicate with each other.

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

Connect: The connect command takes two parameters: a handle (representing the name of the user who wishes to chat); and a port (indicating the number of a port on the client host to which the server should make a TCP socket connection). The server should connect to the client host; store the resulting socket so that it can be used to send chat messages to the client; and broadcast a message to all clients announcing that the new user has entered the chat room.

Send: The send command takes one parameter: a message string. The server broadcasts the message to each connected client using the TCP socket connected to that client.

Disconnect: The disconnect command takes no parameters. It closes the socket connected to the client and removes the socket from any data structures in which it is stored. The server also broadcasts a message to any remaining clients informing them that the user has left the chat room.

Client: The client sends commands to the server. The commands are encoded as HTTP queries, as described above. The client receives chat messages from the server via a TCP socket. The client establishes this TCP socket connection in three steps. First the client constructs a `ServerSocket` associated with a port number. Then the client sends a connect command (via HTTP) to the server, passing the user's handle and the port number. Finally, the client accepts a connection from the server on the `ServerSocket` object. (Note the reversal of the usual roles: The client (applet) uses a `ServerSocket` while the server (servlet) uses a `Socket`.) The client stores the resulting socket so that it can be used to receive chat messages from the server. The client includes a thread that waits for messages arriving in the stored socket and displays them to the user when they arrive.

I recommend that you look some example programs that I have placed in the `375-projects-released.zip` file on our class wiki: `HttpEchoClient` and `HttpEchoServer` illustrate how a client (implemented as an applet) can communicate with a server (implemented as a servlet) using HTTP queries. `ServletPushesToAppletAlt` and `ServletPushesToAppletWebAppAlt` illustrate use of HTTP in combination with TCP to let a servlet send data to an applet continuously, without the applet needing to make an explicit requests. As you implement your programs, you should carefully consider all the ways that communication between the client and server can be terminated, and take appropriate steps to close sockets. If you wish, you may use any of the code in the chat client and server Java application programs that I have posted on our web site as solutions to the last assignment. You may also use any of the code in the example programs mentioned above. Your programs should use the port numbers you were given in previous assignments.