Distributed Systems-Write up

Lab-1

A chat room system

*submitted by,*

*Harshini Chandrasekar*

*1001586563*

*Overview*

Develop a program to implement a chat room system application that consists of a server and up to four client processes. Every client will connect to the server over a socket connection. A separate thread is created for each connected client. The program is written in Java language that is supported by Eclipse Integrated Development Environment (IDE).

The program consists of four classes

1. Server - it implements Runnable interface which executes the

Server thread and handles message from the client(s).

1. ServerThread - Manages each Client connection.
2. ClientApplet - Contains chat interface in which messages are sent to and received from the server
3. ClientThread- Listens to and handles the message received from the server

***INSTALLATION***

Eclipse Java EE IDE for Web Developers.

Version: Oxygen Release (4.7.0)

Build id: 20170620-1800

Install from: <http://www.eclipse.org/downloads>

**Note: Also install the latest JDK and JRE**

***INSTRUCTIONS TO COMPILE***

**Server**

1. Run the Server class as a java application and the server runs on

*Server : Localhost*

*Port : 8080*

1. The Server is bound to the port 8080 and it is started.
2. After the server starts, the server asks (Yes/No) for retrieval of the Chat Log file.

Chat Log file : A text file that maintains the log for all the chats in the chat room system application. The Log file is in the Project Folder as **Chat\_Log.txt** and this file can be retrieved every time when the server restarts.

1. Choice “YES” will retrieve the data from the Chat\_Log Text file and displays on the server. Then the server continuously listens for Clients to connect.
2. Choice “No” will not retrieve the data from the Chat\_Log Text and the server will start to continuously listen for Clients to connect.
3. The Server waits for client to connect.

**Client**

1. Run the ClientApplet as the java Applet and a chat Interface is displayed.

The chat interface has the following buttons

1. Connect Button : On-click of this button, the client is connected to the server.
2. TextField : When the connection is established, the client is asked to register a name which is entered in the TextField. If the entered name is valid (valid format displayed in chat window), the name is registered at the server.

Subsequently, the same Textfield is used by the client for chatting with other clients.

1. Send Button : The send button sends all the data in the TextField to the server which broadcasts the messages to all the connected clients.
2. Log Off Button : The log off button removes the connected client from the chat room. The client details are removed from the server and the applet window is closed to exit from the chat room.

**Assumptions:**

1. The Server and client can run only in Server “localhost” and port number “8080” unless it is changed in the program.
2. Timer is displayed in the following format(s). Ex: [(0:0), (0:14), (3:12)]

**Implementations:**

1. The messages that are sent across the Client-Server uses HTTP formats and commands. Messages sent to the server is sent by POST method(Server shows HTTP message format) and messages polled from the server uses the GET Method.
2. The chat room system can connect up to four clients.
3. Regular expression is used to invalidate any bad name when a client registers at the server.

**References:**

**1.** [**http://pirate.shu.edu/~wachsmut/Teaching/CSAS2214/Virtual/Lectures/chat-client-server.html**](http://pirate.shu.edu/~wachsmut/Teaching/CSAS2214/Virtual/Lectures/chat-client-server.html)

**2.** [**http://javarevisited.blogspot.com/2015/06/how-to-create-http-server-in-java-serversocket-example.html**](http://javarevisited.blogspot.com/2015/06/how-to-create-http-server-in-java-serversocket-example.html)

**3. Secure Program with static analysis by Brian Chess & Jacob West**- **(Page – 319 & 320)**

**4.** [**http://www.baeldung.com/java-write-to-file**](http://www.baeldung.com/java-write-to-file)

**5.** [**https://regex101.com/**](https://regex101.com/)

**6.** [**https://stackoverflow.com/questions/10820033/make-a-simple-timer-in-java/14323134**](https://stackoverflow.com/questions/10820033/make-a-simple-timer-in-java/14323134)