**LAB 1 Report**

**Instructions To compile and run the Program:**

1. Open Command Prompt and compile the Client.java, Server.java,Request.java and Controller.java.
2. Run the Server on this command prompt, Server window appears.
3. Open another command Prompt and Run Client.
4. Enter the username of client, Client window appears.
5. Continue step 3 and 4 for remaining 2 clients.
6. Click “Start Server” button on Server Window.
7. Click “Connect <client name>” button on Client side
8. In order to stop the connection Click on “Exit” button on server side.

**Known Bugs and Limitations:**

1. Once the Server starts it cannot be stopped.
2. After client exits still one message is printed on client side
3. Duplicate Clients cannot be handled.

**Files in Project:**

**Client.java**

This file is used to get username of Client and generate client GUI.

* On running this file there will be a popup asking for username of the client that is to be registered at the server.
* On entering a valid username the Client GUI window is displayed.
* If no username entered or “cancel” button is clicked error is displayed at command prompt and execution completes.
* The GUI consists of text area and 2 buttons
* When connect button is clicked Client tries to find the Server and if Server is not found it displays message in Text Area.
* If connected to server the client generates random number and sends it to server in HTTP format.
* Server Processes the request and returns the message which is printed on Text Area.

**Server.java**

This file is used to establish connection and the Controller is called for managing the requests from clients.

* On running this file the Server GUI window is displayed.
* This GUI has two Textboxes and one Button
* On Clicking Start button the server starts and looks for Clients.
* When Client connection is established, it forwards the request to Controller.

**Controller.java**

This file need not run just it handles the clients who request the server.

* Controller constructor initializes with parameters from Server and the thread is executed.
* In controller the message from client is received in HTTP format.
* Message is displayed in Text Area.
* After receiving message string operations are performed and time is retrieved in Integer.
* Server waits for that time and message is sent to Client in HTTP format.

**Request.java**

This file is used to manage Synchronization between requests of Clients.

* It sends the message to client side from server’s side.
* It makes the message in HTTP format.
* Adds the waiting time and processing time.

**Cites Referred:**

<Author-Rishabh Mahrsee><Title-Introducing Threads in Socket Programming in Java>

https://www.geeksforgeeks.org/introducing-threads-socket-programming-java/

https://stackoverflow.com/questions/18358469/how-can-i-use-the-input-from-a-joptionpane-showinputdialog-to-be-used-as-a-messa

.