***MAIN SERVER***

**ryt.java**

import java.io.\*;

import java.net.\*;

import java.lang.String;

import global.\*;

class ClientWorker implements Runnable {

private ServerSocket serv;

private Socket client;

private static int num = 0 ;

private int id;

ClientWorker(ServerSocket serv) {

this.serv = serv;

num++;

this.id = num;

}

public void run(){

String line;

BufferedReader in = null;

PrintWriter out = null;

String clientSentence ,modifiedSentence ;

String capitalizedSentence;

try

{

while(true) {

//System.out.println(" In " +g.i);

client = serv.accept();

//System.out.println(" accepted " +g.i);

BufferedReader inFromClient = new BufferedReader(new InputStreamReader(client.getInputStream()));

//System.out.println("buferread " +g.i);

DataOutputStream outToClient = new DataOutputStream(client.getOutputStream());

//System.out.println(" dataout " +g.i);

clientSentence = inFromClient.readLine();

// System.out.println("Client sent: " + clientSentence + " id is " + id);

// num = Integer.parseInt(clientSentence);

if (! (InetAddress.getByName(g.ip[g.i])).isReachable(4440) )

{

while(!(InetAddress.getByName(g.ip[g.i])).isReachable(4440) )

{

g.flag[g.i] = false ; g.i++; g.i = g.i%3;

}

} else g.flag[g.i] = true;

//capitalizedSentence = clientSentence.toUpperCase() + '\n';

Socket clientSocket = new Socket(g.ip[g.i], 4444);

DataOutputStream outToServer = new DataOutputStream(clientSocket.getOutputStream());

// Create (buffered) input stream attached to socket

BufferedReader inFromServer = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));

// Write line to server

outToServer.writeBytes(clientSentence + '\n');

modifiedSentence = inFromServer.readLine();

System.out.println("FROM SERVER: " + modifiedSentence);

outToClient.writeBytes(modifiedSentence);

client.close(); // System.out.println(" I " +g.i);

if (! (InetAddress.getByName(g.ip[g.i])).isReachable(4440)){ g.flag[g.i]= false; break;}

// System.out.println ("jerk");

}

}

catch (IOException e) {e.printStackTrace(); }

}

}

public class ryt {

public static void main(String[] args) throws Exception {

//try {

ServerSocket welcomeSocket = new ServerSocket(4444);

/\*} catch (IOException e) {

System.out.println("No I/O");

System.exit(1);

}\*/ //boolean k = true;

// global g;

//ServerSocket echoServer = new ServerSocket(6789);

//accept requests

// declare an adjacency matrix for g.ip addr and g.flag for if\_not\_down

long start = System.currentTimeMillis();

long end = start + 1\*1000; // 6 seconds \* 1000 ms/secd

while(true)

{

if(System.currentTimeMillis() == end)

{

// System.out.println("out I " +g.i);

g.i++;

g.i = g.i%3;

//System.out.println(" I " +g.i);

ClientWorker w; // = new ClientWorker(welcomeSocket);

try{

// if\_not\_down(g.i) getByName(g.ip[g.i])

InetAddress adr = InetAddress.getByName("0.0.0.0");

// if(g.flag[g.i])

{

adr = InetAddress.getByName(g.ip[g.i]); // for those g.ip addr which are not down

System.out.println("Host" + g.ip[g.i] +" is reachable: " +adr.isReachable(4440));

// System.out.println(" I " +g.i);

}

if (!adr.isReachable(4440))

{

g.flag[g.i] = false;

}

else

{ g.flag[g.i] = true;// System.out.println("hell");

w = new ClientWorker(welcomeSocket);

Thread t = new Thread(w);// System.out.println("hell1");

t.start(); //System.out.println("hell2");

}

} catch (IOException e) {

e.printStackTrace();

}

//forward requests if reachable and set unreachable to false

end = System.currentTimeMillis() + 1\*1000;

}

if(System.currentTimeMillis()> end) break;

}

}

}

**package global**

**g.java**

package global;

public class g {

public static String[] ip = {"192.168.18.80","192.168.18.139","192.168.18.244"};

public static boolean [] flag = {true, true ,true}; // 1 -> available 2-> busy 3-> unreachable

public static int i = -1;

}

***TERMINAL SERVER***

**TCPServer.java**

import java.io.\*;

import java.net.\*;

import java.sql.\*;

class TCPServer {

public static void main(String argv[]) throws Exception

{

String clientSentence;

String capitalizedSentence;

int num=0,sum=0;

//for DB connection

Connection conn = null;

String url = "jdbc:mysql://192.168.18.138:3306/";

String dbName = "project";

String driver = "com.mysql.jdbc.Driver";

String userName = "root";

String password = "vnitnagpur";

ServerSocket welcomeSocket = new ServerSocket(4444);

System.out.println("Server Ready for Connection");

while(true) {

Socket connectionSocket = welcomeSocket.accept();

System.out.println("Client Made Connection");

BufferedReader inFromClient = new BufferedReader(new InputStreamReader(connectionSocket.getInputStream()));

DataOutputStream outToClient = new DataOutputStream(connectionSocket.getOutputStream());

clientSentence = inFromClient.readLine();

System.out.println("client got "+clientSentence);

String delims = " ";

String[] tokens = clientSentence.split(delims);

num=Integer.parseInt(tokens[1]);

try

{

System.out.println("---------------------");

Class.forName(driver).newInstance();

conn = DriverManager.getConnection(url+dbName,userName,password);

System.out.println("Connected to the database");

String sql1 = "SELECT name FROM user ";

PreparedStatement psna = conn.prepareStatement(sql1);

ResultSet rs = psna.executeQuery();

boolean flag=false;

while (rs.next()) {

String sru = rs.getString(1);

if(sru.equals(tokens[0]))

flag=true;

}

if (flag)

{

System.out.println("into update function");int y=0;

String sum\_v = "SELECT age From user WHERE name=?";

PreparedStatement psval = conn.prepareStatement(sum\_v);

psval.setString(1,tokens[0]);

ResultSet valu= psval.executeQuery();

while (valu.next())

{y= valu.getInt(1);}

sum=y+num;

String sql = "UPDATE user SET age=? WHERE name=?";

PreparedStatement ps = conn.prepareStatement(sql);

ps.setInt(1,sum);

ps.setString(2,tokens[0]);

ps.executeUpdate();

System.out.println("updated");

}

else

{

System.out.println("inserting func");

sum=num;

String insertTableSQL = "INSERT INTO user (name, age) VALUES (?,?)";

PreparedStatement insps= conn.prepareStatement(insertTableSQL);

insps.setString(1, tokens[0]);

insps.setInt(2,sum);

insps.executeUpdate();

System.out.println("inserted");

}

conn.close();

}

catch (Exception e)

{

e.printStackTrace();

}

capitalizedSentence = new Integer(sum).toString();

System.out.println(tokens[0]);

// Write output line to socket

outToClient.writeBytes(capitalizedSentence);

connectionSocket.close();

} // end while; loop back to accept a new client connection

} // end main

} // end class

***CLIENT***

**TCPClient.java**

import java.io.\*;

import java.net.\*;

class TCPClient {

public static void main(String argv[]) throws Exception

{

String sentence = "sanjay", sent;

//Socket clientSocket = new Socket("192.168.18.63", 4444);

while(true)

{

BufferedReader reader;

reader = new BufferedReader(new InputStreamReader(System.in));

System.out.print("Enter value ");

String ipadd="192.168.18.83";

sentence = reader.readLine();

Socket clientSocket = new Socket("192.168.18.63", 4444);

String modifiedSentence;

// Create (buffered) input stream using standard input

BufferedReader inFromUser = new BufferedReader(new InputStreamReader(System.in));

System.out.println("Client ready for input");

// Create client socket with connection to server at port 6789

// Create output stream attached to socket

DataOutputStream outToServer = new DataOutputStream(clientSocket.getOutputStream());

// Create (buffered) input stream attached to socket

BufferedReader inFromServer = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));

sent = sentence;

sentence=ipadd+" "+ sentence;

System.out.println(sentence);

// Write line to server

outToServer.writeBytes(sentence + '\n');

//System.out.println("sent");

// Read line from server

modifiedSentence = inFromServer.readLine();

if (sent.equals ("EXIT") )

{ System.out.println("Exiting"); break;}

System.out.println("FROM SERVER: " + modifiedSentence);

/\*try

{

Process p=Runtime.getRuntime().exec("exit");

p.waitFor();

BufferedReader reader1=new BufferedReader(new InputStreamReader(p.getInputStream()));

String line=reader1.readLine();

while(line!=null)

{

System.out.println(line);

line=reader1.readLine();

}

}

catch(IOException e1) {}

catch(InterruptedException e2) {}

\*/

//break;

clientSocket.close();

}

//clientSocket.close();

} // end main

} // end class