CT3535 Assignment B2

**Source Code**

ServerSocket.java:

**import** java.io.\*;

**import** java.net.\*;

**import** java.util.\*;

**public** **class** SocketServer{

**public** **static** ArrayList<String> *TokenList* = **new** ArrayList<String>();

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

//Creates a new server with port 4444

ServerSocket TokenServer = **new** ServerSocket(4444);

**while**(**true**) {

//Creates a socket based on the server

Socket socket = TokenServer.accept();

//For input and output between the server and the client

PrintWriter out = **new** PrintWriter(socket.getOutputStream(), **true**);

BufferedReader in = **new** BufferedReader(**new** InputStreamReader (socket.getInputStream()));

//Takes the input from the client command prompt

String str;

**while**((str = in.readLine()) != **null**) {

//Splits the input into the command & the token to be input

String[] li = str.split(" ");

//If block if client query is SUBMIT

**if**(li[0].equals("SUBMIT")) {

//If clause for when list is full

**if**(*TokenList*.size()==10) {

out.println("Error, list is full.");

out.flush();

}

//Else if clause for when the input is already in the list

**else** **if**(*TokenList*.contains(li[1])){

out.println("OK.");

out.flush();

}

//Adds value otherwise

**else** {

*TokenList*.add(li[1]);

out.println("OK");

out.flush();

}

}

//Else-if block for if client query is REMOVE

**else** **if**(li[0].equals("REMOVE")) {

//Checks if the TokenList contains the value, then removes it

**if**(*TokenList*.contains(li[1])) {

*TokenList*.remove(li[1]);

out.println("OK");

out.flush();

}

//Otherwise, tells the user there's an error

**else** {

out.println("Error, value not found in list.");

out.flush();

}

}

//Otherwise close the Server

**else** **if**(li[0].equals("QUIT")) {

in.close();

TokenServer.close();

}

//prints out invalid statment otherweise

**else** {

out.println("Invalid.");

out.flush();

}

}

}

}

}

**Screenshot:**

