

Developing A Chat Server by Prabhat Chandra Computer Science Department IFTM University (Moradabad)

Mr. ABHISHEK KUMAR MISHRA

Assistant Professor, Department Of Computer Science and Engg., IFTM University Moradabad, UP, India

Abstract:- *Teleconferencing or chatting, is a method of using technology to bring people and ideas "together" despite of the geographical barriers. The technology has been available for years but the acceptance it was quit recent. our project is an example of a chat server. It is made up of 2 applications the client applications, which runs on the user's Pc and server application, which runs on any Pc on the network. To start chatting client should get connected to server where they can practice two kinds of chatting, public one (message is broadcasted to all connected users) and private one (between any 2 users only) and during the last one security measures were taken.*

Keywords:- Java, Chat, Console Window, AWT

I. INTRODUCTION

Chat server is an online system developed for the group of members to communicate with each other over the internet. This system solves almost all the limitations of the conventional system. Here in this, a java client server combination is used to chat with remote users. When a client wants to chat with a user on a remote host, he sends a request to the server with a identification name like chat-id, the server responds to the request by identifying the client-id which is already registered in the server domain and when matched this request is granted and the client can begin to chat with the remote users present on the internet or local network. The power of internet is such that it integrates together LANs located across diverse software and hardware forms into a single large communication network that spans the globe. The client needs to have a client software such as Netscape Navigator or Internet Explorer to retrieve information as well as chat on the www.WWW is referred to as Netsurfing. And it supports chatting, text, video and audio.

II. OBJECTIVE

This particular paper is a solution developed to communicate between the users worldwide through internet. "Chat Server" automates all the aspects related to communication in a highly secure environment. This paper has been developed to receive instant message and to provide total user satisfaction.

The entire process has been automated using JAVA technology and SQL SERVER to smoothen the flow of information in a highly secure environment across the network. The solution has been deployed, tested and validated thoroughly. While designing the system, care has been taken in efficiency, maintenance and reusability of the software for the present and future changes in the system.

The main objective of this paper is to bring all member of any particular field for solving problem related to their job. not only this paper will help to bring make new friends for social networking and keeping all of them in contact with the details that they provides on the profile while sign up or at the time of registration on the site of the chat server.

III. TECHNICAL DETAILS

(i) Proposed Systems: The proposed "Chat Server" is a chatting application that provides a good user interface with all the facilities of a chat application. First phase of that application user have to do a small registration to create a id and password. Users have their user-id. user can communicate with any other user, who is online. User gets a list of offline users. User can select any user and send a message.

Message can be formatted in terms of font style and size. He can also send emoticons. This application also manages the list of all online users and also provides the facility of browser window to surf simultaneous. In that apps there are two module used namely given below

1. Client module
2. Server module

Client Module: It provides a frame to login to the chat server. After providing the userid and password the user gets access to his main frame. He will get a list of friend and also get a list of online user. He will get a list of online users, area to enter the text, options to format the message. He can also send some image with the

message. This application provides option for changing his personal profile. The user can add any other user to his friend list.

Server Module: The server module is responsible for validating the user information. It allows the authorized user to get facilities provided by this application. It also displays the online users list. It is connected to the database server to perform the function like modifying the user profile, creating a new user account, etc.

- > **Administrator level**
- > **staff level**
- > **Student level**
- > **Parent level**

Administrator level: From the name itself we know that it is administrator's part. Only the administrator is authorized to log in to it. If any changes is needed in the system, he enter this level and will make enough changes .He is the only authorized person to alter the details in database and other important areas of the system. The updating of the details and other details are edited by him. If a new user is registered administrator has the privilege to approve or deny his /her request.

Staff level: This is for the staffs working at the various departments or branches. Staff will log in using the user id and password. Staff will enter the details(subject , department etc..) while registering feedback and chatting facility are available to teachers.

Student level: Student can login in to the site from anywhere .He can register online, if registration is completed, he is provided with login name and password .Using this he can login and know the status of the other chatters .He can chat with his friends(online and offline) and a also send feedback to his teachers.

Parent level: Student can login in to the site from anywhere .He can register online, if registration is completed, he is provided with login name and password .Using this he can login and know the status of the other chatters .He can chat with his sons studying in that college (online and offline) and a also make a conversation with teachers about the students.

User interface priorities:

- A. Professional look and feel
- B. Use of AJAX atleast with all registration forms
- C. Browser testing and support for IE, NN, Mozilla, and Firefox

IV. ADVANTAGES OF PROPOSED SYSTEM

The application gives economic as well as the technical advantages. Some of the main advantages are:

As the solutions is integrated and developed on the JAVA technology, it provides multi functionality services. The solution is built purely on the Java Technology using the latest version of Java Swing. So the solution is platform independent and architecture independent and supports different RDBMS packages as Java has in built support for the drivers of different databases. The JAVA based solution is defined as the solution for the enterprise wide application where each and every individual system are integrated on the JAVA platform for the smooth communication among different technologies and solutions.

This particular project is a solution developed to communicate between the users across the world through Internet. The concept of sending letters and telegraphs has been reduced due to the new era of Internet Mailing. One such facility is being provided by the Chat Server. A message or an information can be sent via many medias, such as telephone, telegrams, fax, etc to the recipient. Each such information requires a high level of security. To maintain such security and smooth completion of any communication it requires more time and human effort in manual systems.

4. Hardware Requirement Specification

Server side:

<u>Description</u>	<u>Minimum</u>	<u>Recommended</u>
System type	IBM compatible PC with 1GHz	IBM compatible PC with 2GHz
RAM	256 MB	512 MB
Cache	256 KB	512 KB
Storage	40 GB	120 GB (Convenient for Backup)

Disk Drive	Floppy or CD-RW (Back up purpose)	Floppy or CD-RW or DAT (Back up purpose)
Display	15" VGA	17" SVGA (LCD)
User Interface	Key Board and mouse	Compatible keyboard and mouse
Output media	DMP Printer	Laser Printer (For reports)

Client Side:

Description	Minimum	Recommended
System type	IBM compatible PC with 1GHz	IBM compatible PC with 1GHz
RAM	128 MB	256 MB
Cache	256 KB	512 KB
Storage	20 GB	40 GB (Convenient for Backup)
Disk Drive	Floppy or CD-RW (Back up purpose)	Floppy or CD-RW (Back up purpose)
Display (LCD)	14" VGA	15" SVGA
User Interface	Key Board	Compatible keyboard and mouse
Output media	DMP Printer	Laser Printer (For reports)

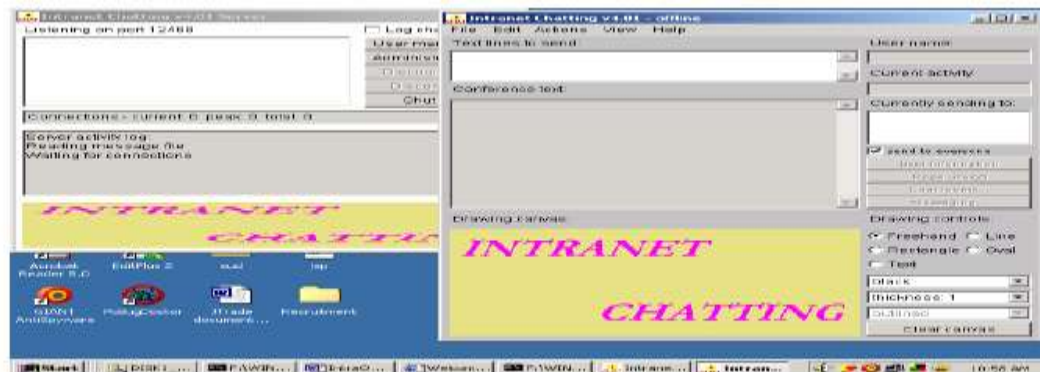
V. ALGORITHM USED

(i) **Control Strings:** A set of "Control Strings" was created, such that whenever the client or the server receives one of these control strings, it should do a certain task. To distinguish these strings from normal strings (chatting sentences) they are given the following format: "ControlCommand".

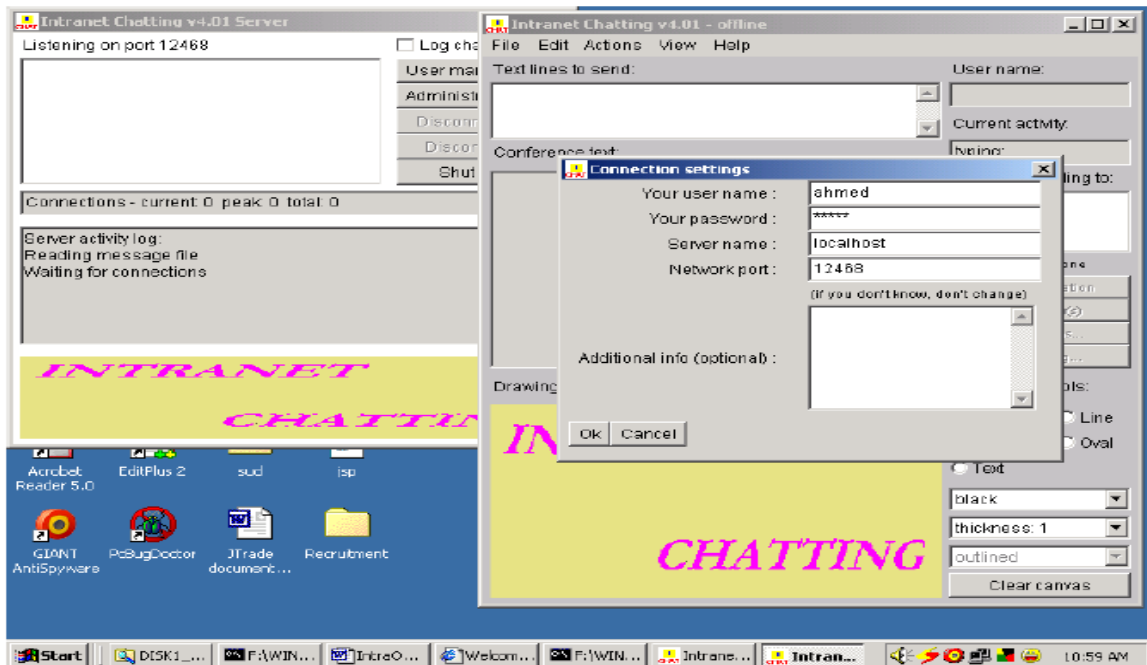
For example if the client sends to server the string "*Out valid", the server upon receiving this message, the message parsing operation starts. It looks at the first character if it is a "*" immediately it recognizes that this is a control string. It continues parsing until it reads the control command which is in this case "Out". It matches this control to the group of commands it has. Where it finds out that this client (valid) is trying to disconnect himself, so the function "User_Out" is called where the disconnection operation is completed. Else if the string does not begin with a star, then it is a normal string and therefore it should be broadcasted to other connected users to be displayed in their public window. In fact there are many controls in both server and client where each has its own job and some of these controls are common between both server and client.

(ii) **Client:** The client form is divided into four sections. A text box of the username, a text box of the host where the server application is running, another text box of the public sent data, and a list of the connected users. To get connected to server you just enter a username, the host name you want to connect to, and finally hit the connect button figure (ii) shows the client form.

During this connection operation, the client will send a control "*In username" indicating to the server that there is a new chatter. The server will parse this string and call the "User_In" function which adds this username to his list and retransmit another identical control string to all connected users containing all the online usernames in order to update the "connected users list" in the client's form. The same thing happens when the user wants to disconnect himself where instead of sending "*In username", "*Out username" is sent and "User Out" function is called which disconnects the user and do all the updates.

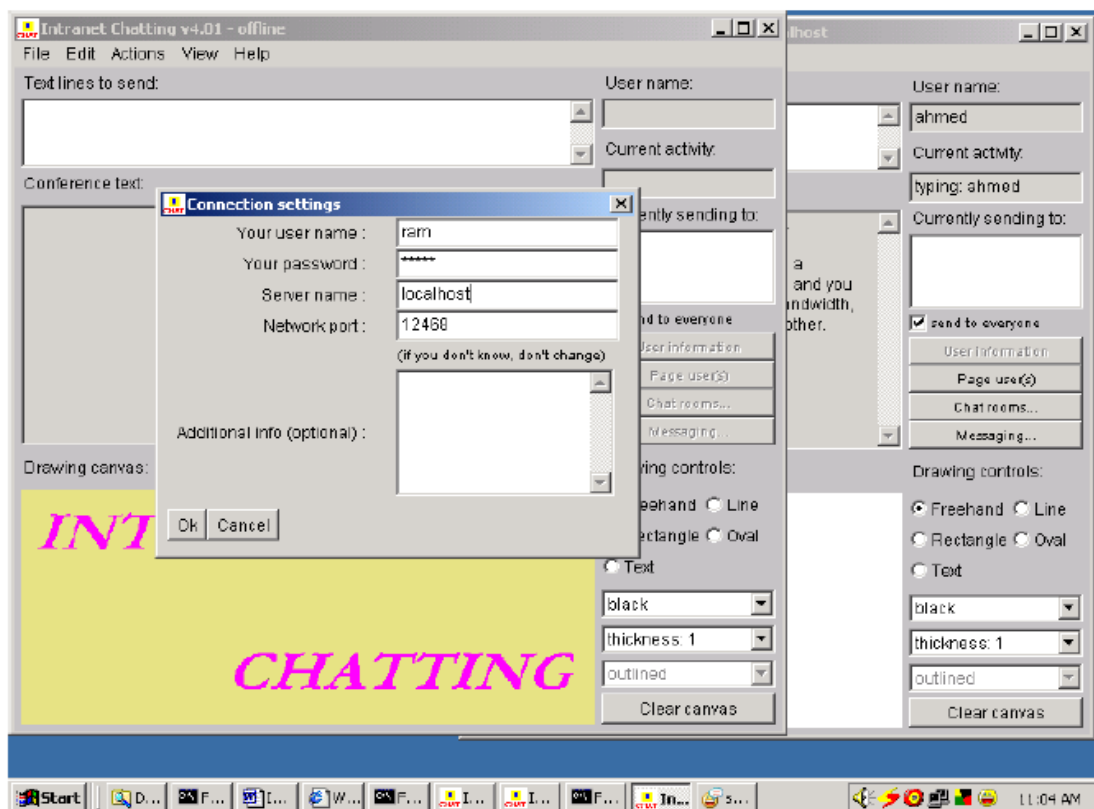


Client Started in another window

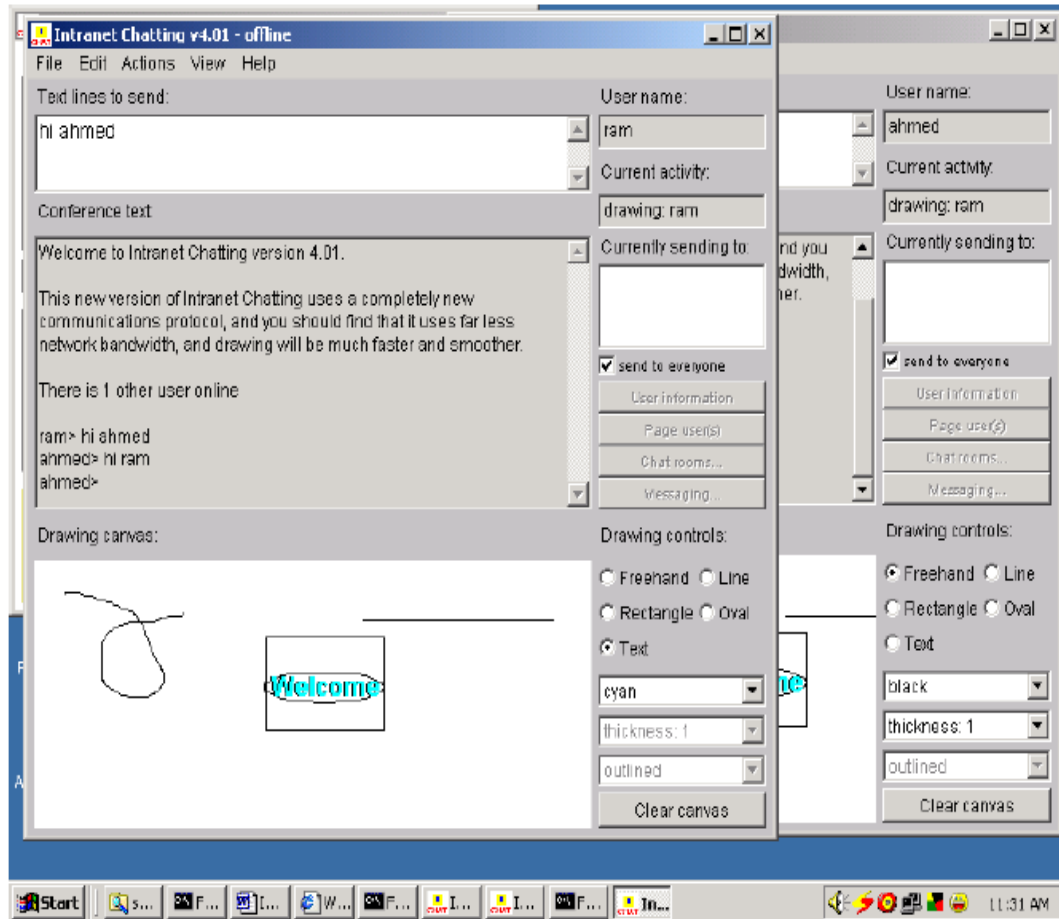


Client Login to Server

(ii) **Server:** Till now it is clear that the server application has dual job, one to receive data and another to transmit data. In fact it plays the role of a “traffic officer ” (putting everything in order the right order). Since all the chatting operations are passed through the server, then the main job of server is responding to the control strings sent by clients as we have seen in the “in” and “out” requests. Figures (ii) shows the server form. User online



(iii) **Private chatting:** When ever a client wants to initiate a private session, in the list of connected users, he double clicks the username he wants to chat with. Two phenomenon will take place. First, open a local-private session window. Second, to tell the server that he wants to chat privately in order to open a remote private window. When these conditions are satisfied, private chatting can be practiced by just writing in the send text box and then hitting the enter button in your keyboard. Figure (iii) shows the private session form.



VI. CONCLUSION

As a matter of fact this paper took me one year to be completed and this year is divided into two sections, six months of collecting information and six months of writing Visual Basic code. Having a chat server as a senior project obliged you and indirectly to go into the tiny details in networking and no one will teach you these details. Lots of experience was gained and another beautiful face of networking was discovered.

ACKNOWLEDGEMENT

We sincerely acknowledge all the cited authors for giving us base papers with strong fundamental concepts, we would like to thank faculty members of Department of Computer Science and Engineering, IFTM UNIVERSITY MORADABAD for reviewing and providing invaluable minute details on organization of the paper.

REFERENCES

- [1]. J2EE by GARRY CORNELL.
- [2]. J2EE by Allamaraju & Buest (Apress Publications).
- [3]. JSP & Servlets by Marty Hall (SUN Microsystems)
- [4]. SYSTEM ANALYSIS AND DESIGN by ELIAS M AWAD.
- [5]. SOFTWARE ENGINEERING by IAN SOMMEVILLE.
- [6]. GNIX INFO SOFT MATERIALS
- [7]. HEAD FIRST BY KATHY SIERA