



“Enhanced Chat Application”

By Avinash Bamane, Parikshit Bhoyar, Ashish Dugar & Lineesh Antony

Pune University

Abstract - We have come across various chat applications to instantly communicate with people. We have used various types of chat application in web-based applications. All these chat applications support text messages to be sent between the users in the instant they press Enter key. But if some problem has to be explained using diagrams, that is where most of these chats are helpless. The chat application we intend to do solves this problem as we are going to integrate a paint like editor to the chat window, so that diagrams could also be drawn and sent through chat. This could be helpful to many of the professional institutions like schools, colleges and industrial units. So we intend to design this application for LAN of these organizations. The people could use many features of this chat application to communicate and brainstorm within a LAN. For this application we need design it on a Client-Server architecture.

Though the primary focus of our application is on the feature of providing communication through diagrams and figures, we would also provide different features in our chat app like using predictive texting, providing themes, voice to smiley, etc. thus making an enhanced chat application.

GJCST-E Classification: H.4.3



Strictly as per the compliance and regulations of:



“Enhanced Chat Application”

Avinash Bamane^α, Parikshit Bhojar^σ, Ashish Dugar^ρ & Lineesh Antony^ω

Abstract - We have come across various chat applications to instantly communicate with people. We have used various types of chat application in web-based applications. All these chat applications support text messages to be sent between the users in the instant they press Enter key. But if some problem has to be explained using diagrams, that is where most of these chats are helpless. The chat application we intend to do solves this problem as we are going to integrate a paint like editor to the chat window, so that diagrams could also be drawn and sent through chat. This could be helpful to many of the professional institutions like schools, colleges and industrial units. So we intend to design this application for LAN of these organizations. The people could use many features of this chat application to communicate and brainstorm within a LAN. For this application we need design it on a Client-Server architecture.

Though the primary focus of our application is on the feature of providing communication through diagrams and figures, we would also provide different features in our chat app like using predictive texting, providing themes, voice to smiley, etc. thus making an enhanced chat application.

I. INTRODUCTION

We have used various types of chat application in web-based applications. All these chat applications support text messages to be sent between the users in the instant they press Enter key. Now suppose two friends are discussing Geometry problems which actually contain figures drawings and diagrams. When these two are actually far away from each other and still want discuss these problems simple texting is not the ideal solution for this. It would be so much better for them if they could communicate using actual diagrams and figures. Even telephoning won't help.

One solution we have come up with is to create a chat application using which we not only communicate using text messages but also with figures and diagrams. This would be helpful to a lot of professionals all over the world. It would also be popular for ordinary users. Our chat application is designed for professional organizations like schools, colleges, industrial units and organizations.

Though the primary focus of our application is on the feature of providing communication through diagrams and figures, we would also provide different features in our chat app like using predictive texting, exchange of files, providing themes, voice to smiley, etc.

Author α σ ρ ω : BE Computer, Pimpri Chinchwad College of Engineering.

II. SYSTEM FEATURES

The Chat application we intend to design would provide us with the following features:

- **Creating an Account:** The user gives his personal information, chooses a UserID and a password of his own choice. The database on the server side is updated after this.
- **Login and Logout:** These functions indicate the availability or non-availability of the user.
- **Handling Users:** It allows administrator to add or remove any user account in case of changes due to termination, resignation, violation of any rules or for some other reasons.
- **Instant Messaging:** This is a regular feature of any chat. This chat application would also provide this feature between multiple users.
- **Drawing:** The application allows users to use the paint window within the chat to draw basic figures like lines, rectangles, circles, etc or free hand.
- **Painting and Coloring:** The application allows users to use colors to draw figures and fill the figures. Different colors have been provided.
- **Predictive Text:** In this feature we would give user the option of using different words by predicting what the user is going to type from initial alphabets. The predictive texting is self learning. If the user has used typed “computer” word once. Then on typing of the letters “com”, the word “computer” would be displayed.
- **Transferring images:** The chat application would also have to transfer images drawn by the users between each other.

III. SYSTEM ARCHITECTURE

Following diagram shows the architecture for the Enhanced Chat Application. In this diagram it shows how Application Work.

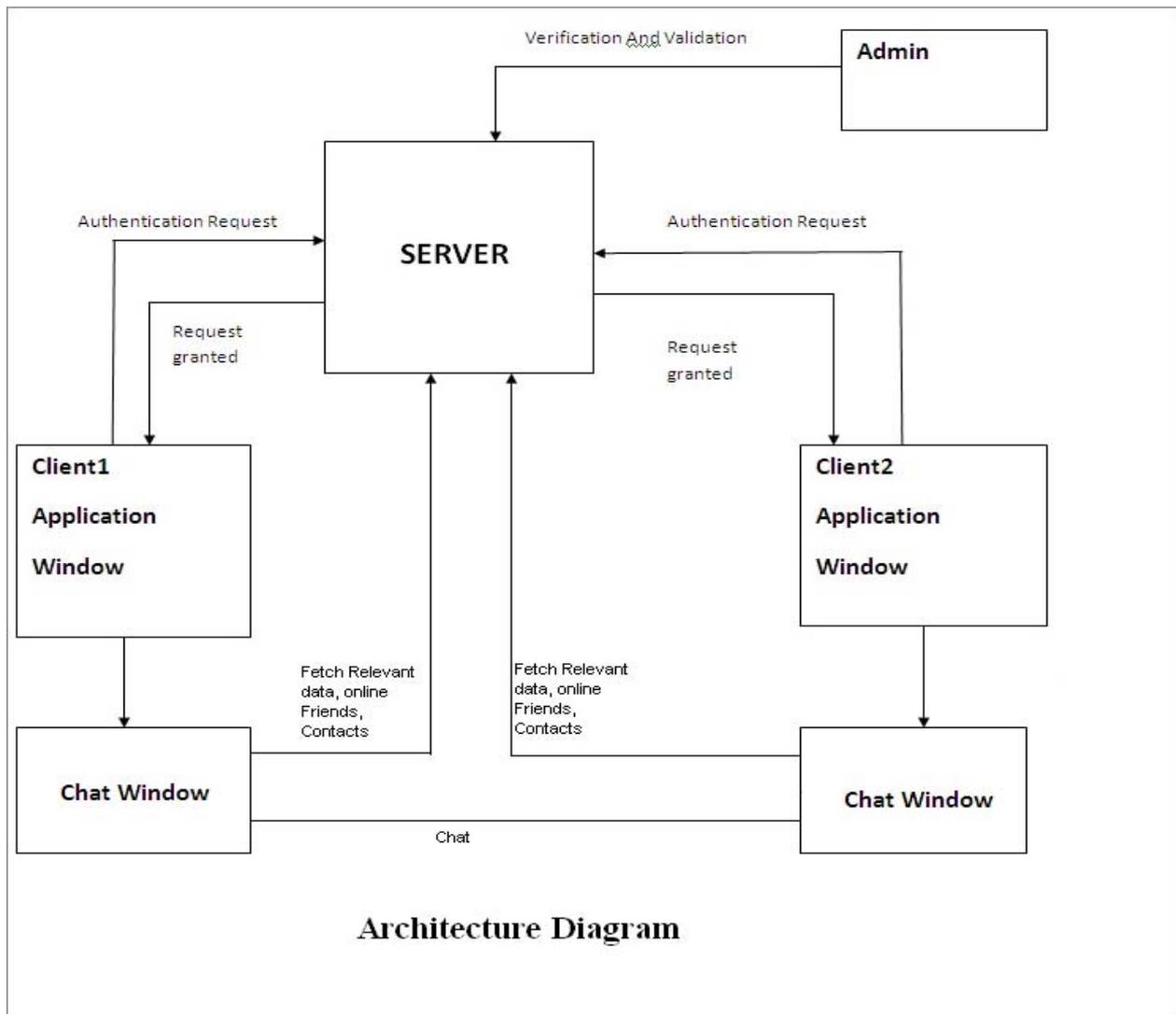


Fig.1 : System Architecture Diagram

The chat application would be accomplished on a Server-Client Architecture within a lan. The client-server model of computing is a distributed application that partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients. The Server Side would be a continuously running service listening to the different Clients asking its services. The chat application would be installed on every communicating client. A Database of users would be maintained by the Server. When a client PC logs in to the application, the Server authenticates the user of the client PC. Once the user is authenticated the IP address of the client is registered to the Server and it sends the list of online user friends and other relevant data to the Client. When the user wishes to chat to some other user, his IP address along with a Port address would be sent to the other user and vice versa. Thus a connection would be established and the two client PCs would be able to chat together.

IV. SYSTEM FEATURES

The system that we have built encompasses the following features. The working and explanation of the features have been given.

a) Chat Server

The System that we have designed is based on client-server architecture. So creating a server is necessary. The functions of a server are to maintain user information, maintaining the record of users logged in to the system and providing the IP addresses to the clients who want to chat.

In the Server side a server-socket is created which accepts connection from client those who have logged in. Once any user logs in and makes connection with the server, it is assigned a thread. Input stream and output stream objects are created in the run method. If the user has requested for registration then a new entry is added to the Database. If the request is for sign in

then the user is verified and then his flag is set as an online user.

b) Painting

Once the user logs in, he can use the application to draw and send images. This has been implemented by using a painting class. The following functionalities have been provided to the user. He can draw any of the following figures like point, line, circle, rectangle and rounded rectangle. Free hand drawing is also possible. Filled circle, filled rectangle and filled rounded rectangle are the other drawing figures available. User can use any of the ten colors for drawing. An eraser has been provided for erasing the drawn figures. These are the functionalities provided to the user. The working of the painting functionality can be explained. The painting class has implemented the

action listener and mouse listeners. The action such as selecting the drawing tool and color of drawing is recorded. On the mouse click event the initial coordinates of the location of mouse is saved. The initial coordinates and the new coordinates on mouse move are used to draw the figures. Whenever the mouse is moved the paint function is called and the figures are drawn. The socket used for sending to the other client has already been created and is used by the paint function to send the coordinates to the other client. The receive class receives the coordinates sent by the other class which is then used by the receive panel to draw the images and figures. The server socket created for that particular connection is used to receive the coordinates, type of the figure and the color. In this way drawing has been done in the application.

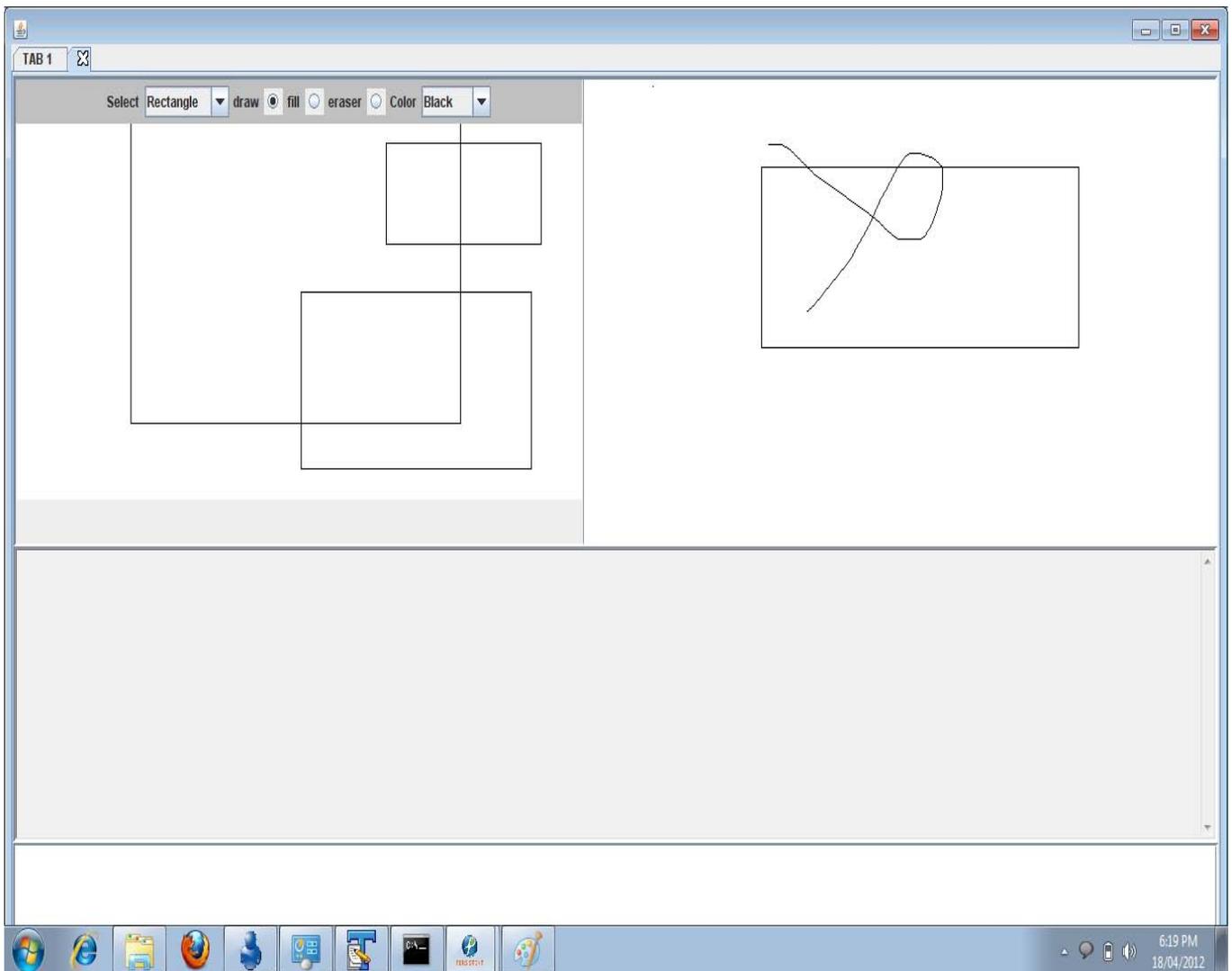


Fig. 2 : Painting done in the application

c) Message Texting

The user can type and send text messages to the other user instantly. This has been accomplished by socket connection. The SplitWindow class gets the

sending socket used for connection. The text messages are sent through the socket. The key listener has been implemented on the bottom-most window where a user types the message. On the press of the Enter key, the

text message is sent to the other user and displayed over the text area of both the user. The receive function on the other side differentiates the text messages from those of the drawing and text messages and sends it to the text-area of the receiving user. In this way the texting has been implemented in this application.

d) *Predictive Texting*

Predictive testing is another enhancement that has been made in this application. Basically predictive texting means to predict what a user is going to type from the set of letters (substring) that user has already typed. The predictive texting system that has been implemented for this application is a self-learning from the typing of the users using. It does not have a prior database of words but instead adds the database of words as the user types. The words are stored for the particular session of conversation after which the words are not available again. So that the words do not clash with some other user who might later sit on the machine and logins to his account.

The Split Window class is where the windows for texting and painting have been implemented. In the class the auto-text complete class is called on the

bottom most text field. So on key press events, the functioning of predictive texting begins. If the total length of the word is 3, the hash-chain class is called. A hash table is created for size of 10 words. The substring of 3 letters is passed as on of the parameters to the find function of the hash table. It computes the hashing value for the given substring which can be used to search all the words beginning with the substring or else can be used to insert the word to the specified location of the hash value. The hash function used for this purpose can be explained as follows. The hash value is calculated by summing up the ascii value of the letters of the substring, then a modular division is performed on the sum by the array size of the hash table. Thus a maximum of 10 words can be stored for a particular substring.

A frequency count of the words used has been maintained which provides the recurrence of words in typing. More the frequency of the words higher will be their priority. The lower priority words would be deleted from the hash table when it fills up. In this way the predictive texting has been implemented.

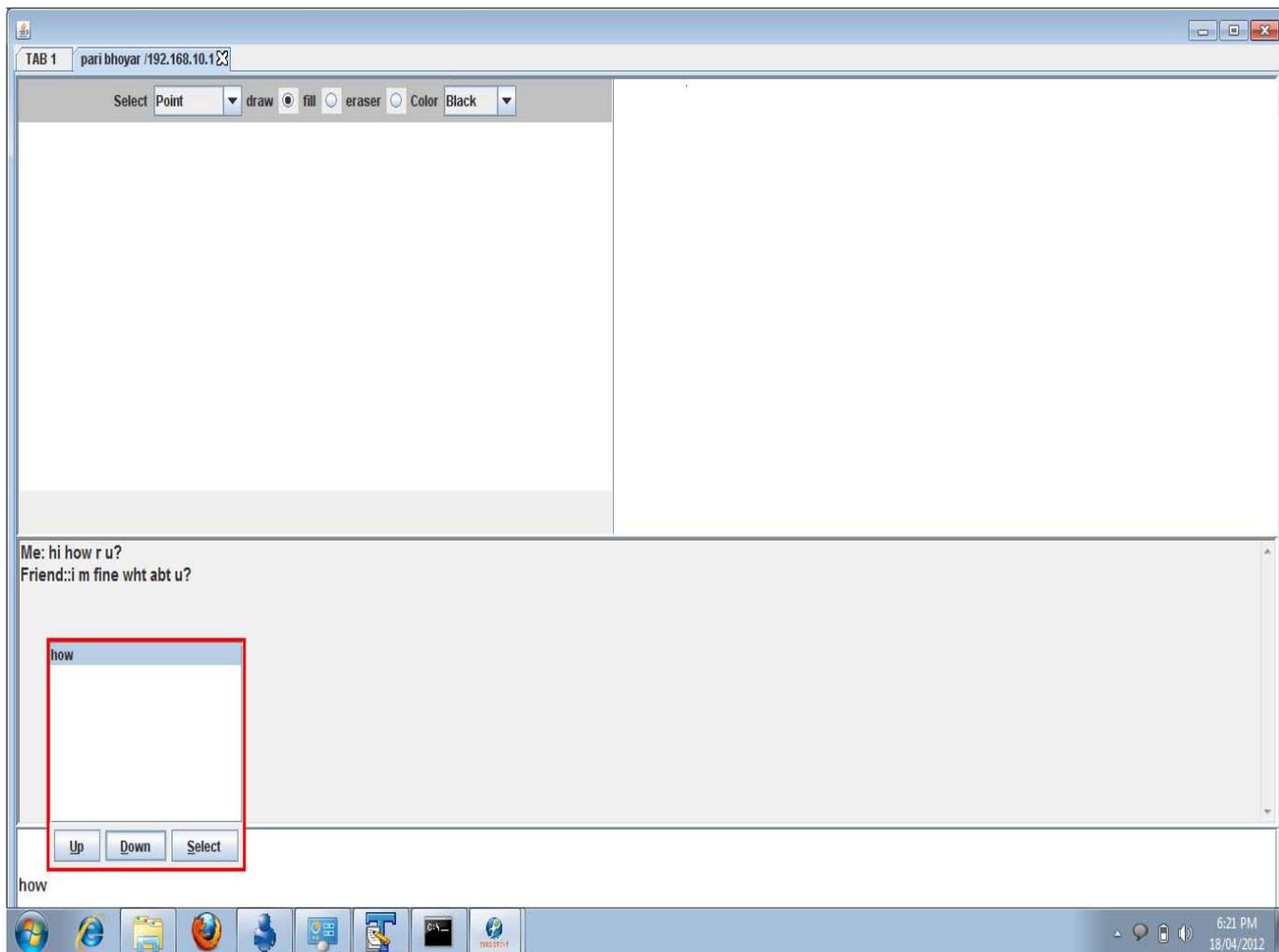


Fig. 2: Predictive texting done in the application

V. TOOLS USED AND OTHER REQUIREMENTS

a) Software tools used

- JDK for executing the application
- Oracle for storing the database of users
- Editplus, JCreator and Netbeans for coding the program

b) Hardware requirements

- A Local Area Network – can use switches, hubs and LAN cables.
- Two or more machines where the application could be run and one server machine.

VI. CONCLUSION

Chatting is a very common used application among the users. General users use the instant messaging services to communicate with other individual users. In our project we have provided with many enhanced features for a chat application. The features like painting along with chat would be a fun to use and interactive feature for a general user. For professional users it would be very useful for communicating important flowcharts, diagrammatic representation of some problem, making important symbols, etc. It opens up a wide variety of uses for individuals. The predictive texting feature would help a user to chat easily. Various figures of various formats could be opened and sent to other user. It would also give freedom of using any tool for drawing.

The chat application is so aimed that the people could have a better experience of chatting. It has the potential to attract more and more users to interact and connect.

BIBLIOGRAPHY

1. Lam, Edmund Y. and Joseph W. Goodman. "Discrete Cosine Transform domain restoration of defocussed images." *Applied Optics*. 37, 6213-6218 (1998).
2. Ken Kabeen and Peter Gent. "Image Compression and Discrete Cosine Transform" College of Redwoods.
3. Predictive Text Entry using Syntax and Semantics - Sebastian Ganslandt, Jakob Jörwall, Pierre Nugues
4. <http://www.java2s.com/Code/Java/JSTL/SimpleChatApplication.htm>
5. http://www.codeproject.com/KB/java/java_applet_chat_with_gui.aspx
6. <http://oreilly.com/catalog/javmessenger/chapter/ch02.html>



This page is intentionally left blank