



Vidyavardhini's College of Engineering & Technology

Vasai Road (W)

Department of Information Technology
Academic year (2021-22)

Mini Project Report on “CHAT APPLICATION”

Group Members: -

- + Sanskruti Kokare – 45
- + Harsh Sawant – 46
- + Pratyush Lokhande – 47
- + Vatsal shah – 48

Project Guided Under: -

- + Vaishali Shirsath mam

Abstract

Teleconferencing or chatting refers to any kind of communication that offers a real-time transmission of messages from sender to the receiver. Chatting is a method of using technology to bring people and ideas together despite the geographical barriers. The technology to provide the chatting facility has been available for years, but the acceptance is quite recent. Analysis of chatting provides an overview of the technologies used, available features, functions, system of the architecture of the application, the structure of database of an Instant Messaging application: IChat (IC). The objective of IC application is to facilitate text messaging, group chatting option, data transfer without size restriction which is commonly seen in most of the messaging applications.

Even we have been trying for the voice command scope for our chat application, as nowadays its common to have the chat application with the text features and many more features but having voice command which will help the user to save time and would be easier for many to handle things on their voice command only.

Introduction

CHATTING APPLICATION IS DESKTOP BASED APPLICATION:

The client server chat application is based on java swing and user socket packages. It's simple and easy and requires only core java knowledge. This application/program is good example of using java.io java.net packages to create a chat application.

Chatting is the method of using technology to bring people and ideas “together” despite of the geographical barriers.

It is made up of 2 applications the client application, which runs on the users Pc and server application, which runs on anyone Pc on the network. To start chatting client should get connected to server.

External Interface Requirements

USER INTERFACE:

The user interface required to be developed for the system should be user friendly and attractive. There are two sets of Java APIs for graphic programming:

AWT (Abstract Windowing Toolkit) and swing.

- ☐ AWT API was introduced in JDK 1.0. Most of the AWT components have become obsolete and should be replaced by newer Swing components.
- ☐ Swing API, a much more comprehensive set graphics libraries that enhances the AWT, was introduced as part of Java Foundation Classes (JFC) after and the

release of JDK 1.1. JFC consists of Swing, Java2D, Accessibility, Support APIs. JFC was an add-on to JDK 1.1 but has been integrated into core Java since 1.2.

SOFTWARE INTEFACE: -

Programming Language Java and socket programming.

Problem Statement: -

Communication is one of the most important part of daily life. There are many different means of communication. One such means of communication. One such means is chatting. There will be hardly any person who does not use a chat application nowadays.

Chat server is network-based application. Even though its small application, but eligible to meet client – side requirements. It will be made using graphical components in the swing toolkit in java.

Related literature

We have referred the follow sites or similar project which are existing. We have referred for the learning things related to project like for understanding packages, syntax, different classes, etc. for learning swing and implementing it in project source code, to solve errors. For connectivity and DBMS, we took help for our classmates – we learn things form them.

We have referred the following:

- Java documentation
- YouTube
- Wikipedia
- www.javatpoint.com
- stackoverflow.com
- itsourcecode.com
- www.freeprojectz.com
- code-projects.org
- Broadcasting Chatserver Project | Core java project-javatpoint
- projectsgeek.com
- freefrontend.com
- code.tutsplus.com

Proposed Method

This project can be mainly divided into two modules:

1. Server
2. Client

This project is mainly depended on client/server model. The client requests the server and server responses by granting the client request. The proposed system should provide the both of the above features with the followed one:

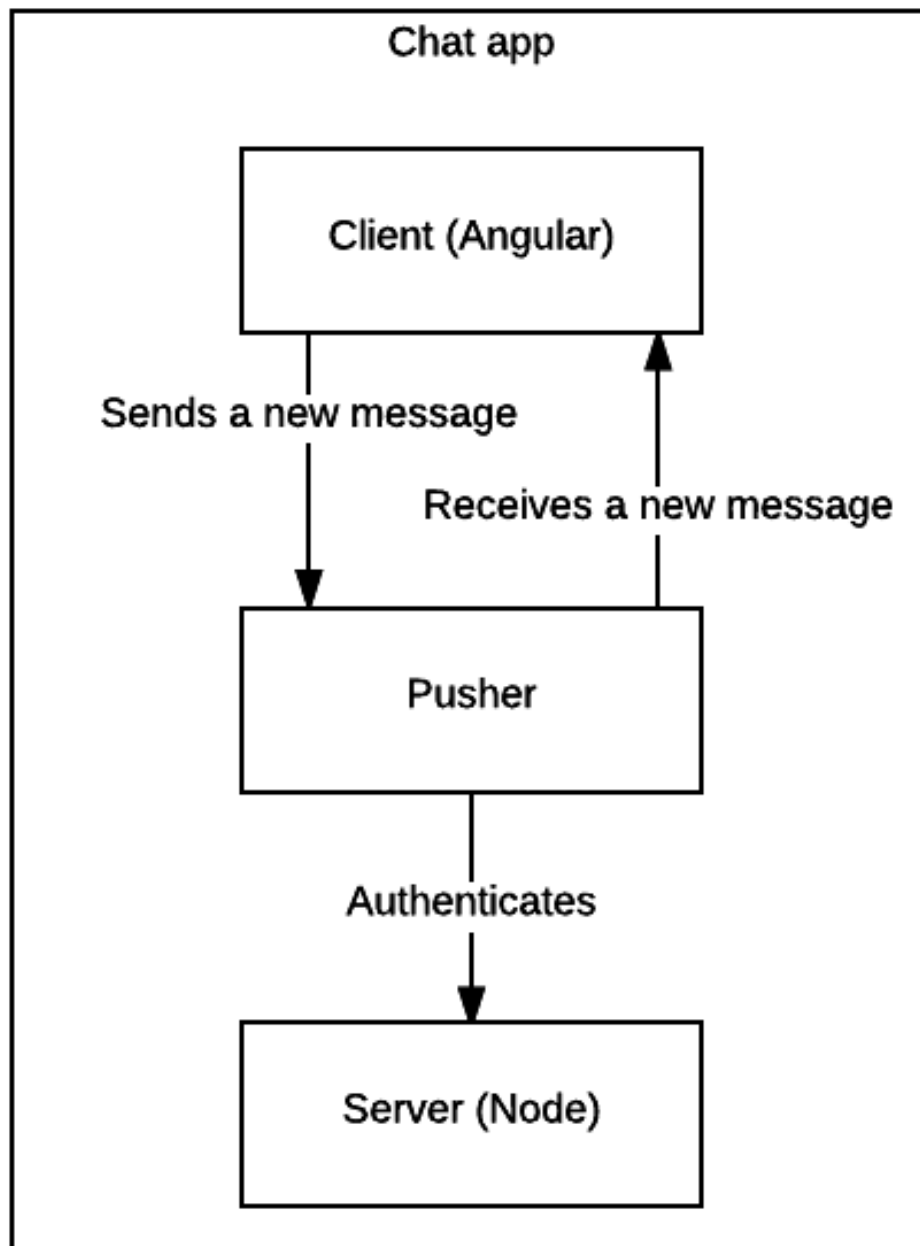
Flow of Program: -

The aim of this project is how we can implement a simple chat application between a server and client. The application is desktop based and is implemented using swing and AWT. The project is developed in java SE language executed on single alone java across a network using loop back address concept.

Application consists of two programs:

- Server
- Client

Flowchart for chat application: -

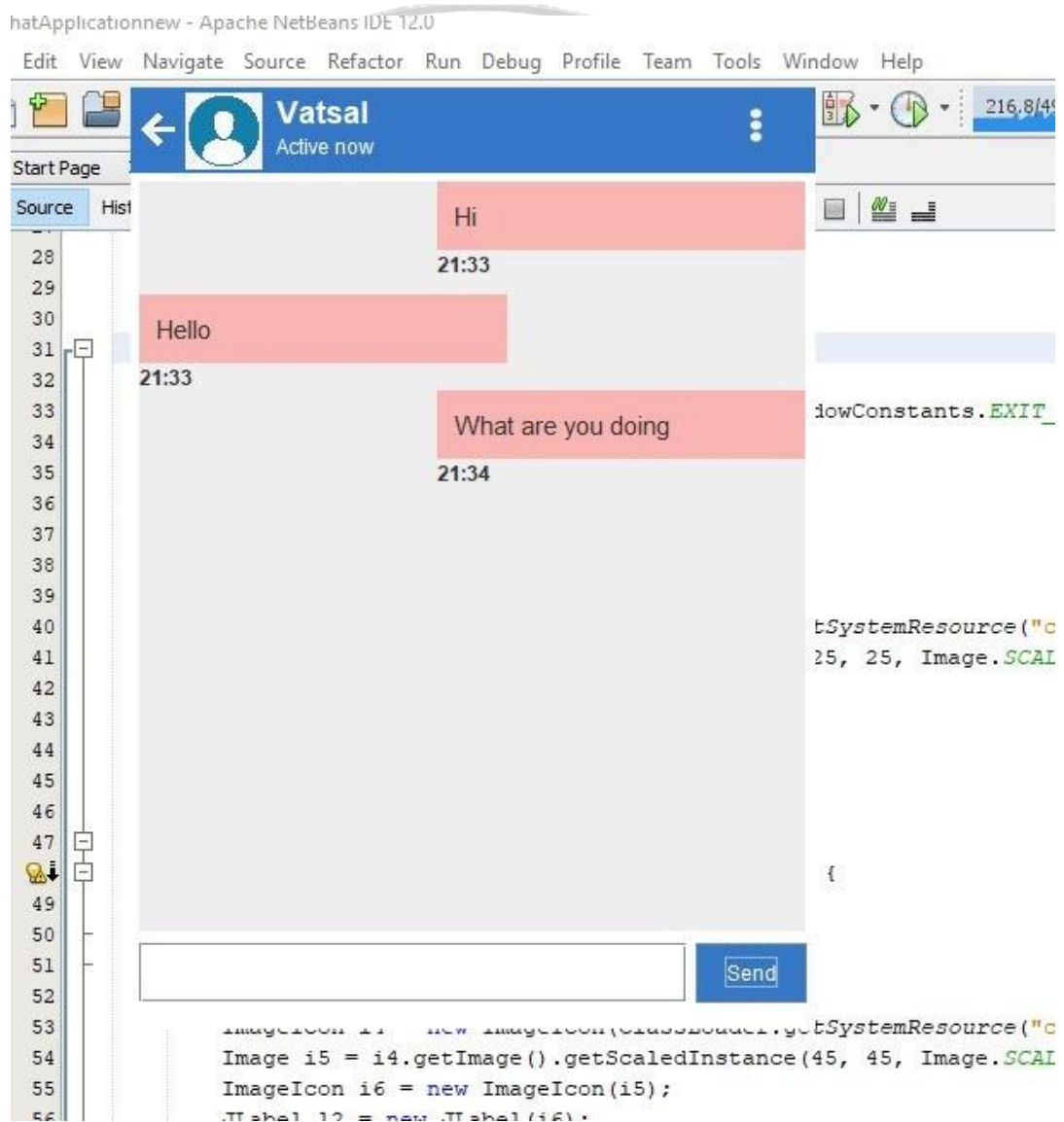


Overview of chat application: -

○ Server: -

The server module of the application waits for the client to connect to it. The duty of the server is to let clients exchange the messages.

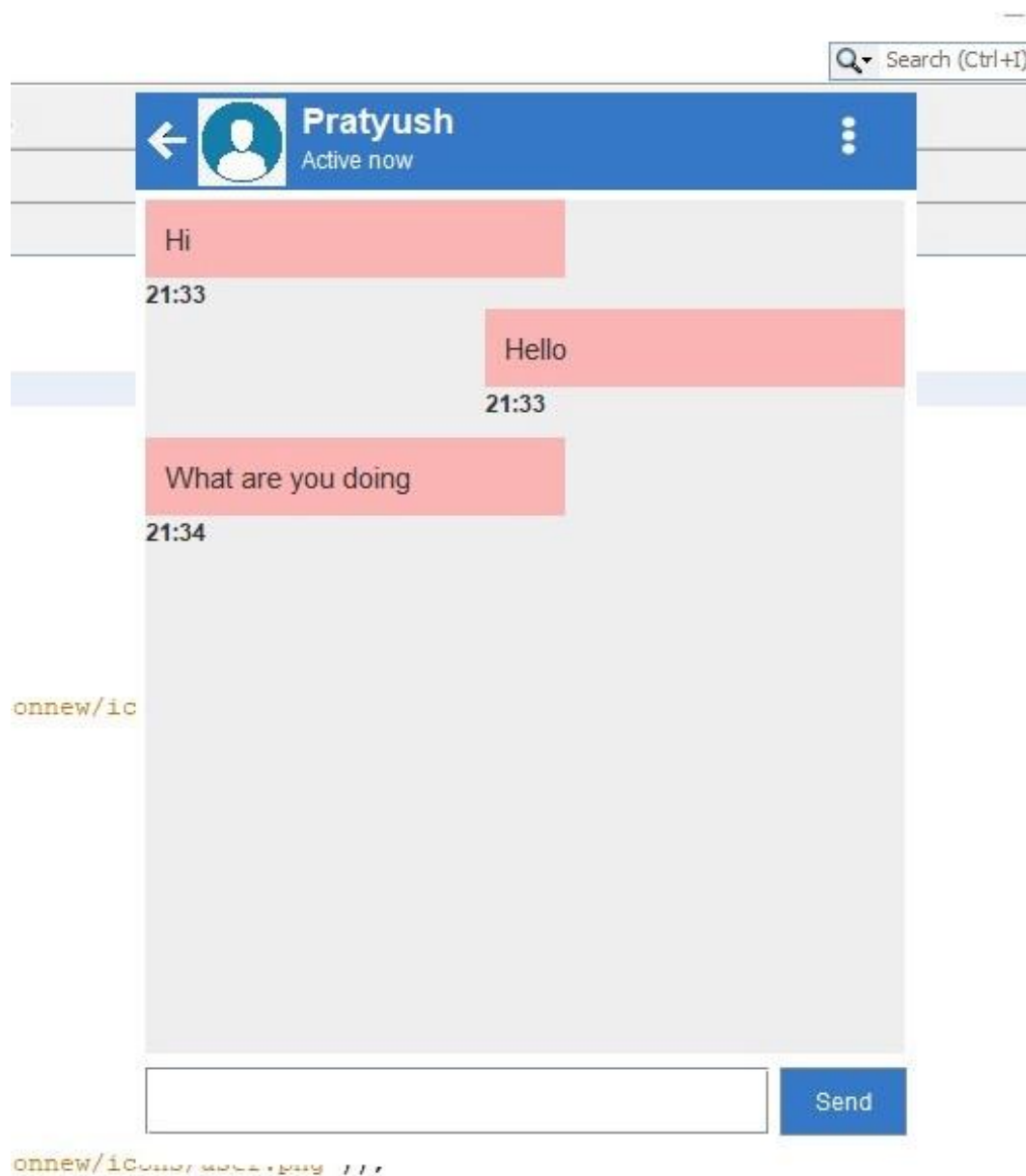
▪ The Server Screen:



- ***Client: -***

The client module is the one that utilizes requests to the server. Utilizer utilizes the client as the means to connect to the server. The connection can communicate to the connected server.

- ***The Client Screen:***



It can mutually communicate with the server and client and hence the user will get the output as the client server screen.

As shown in figure 1.

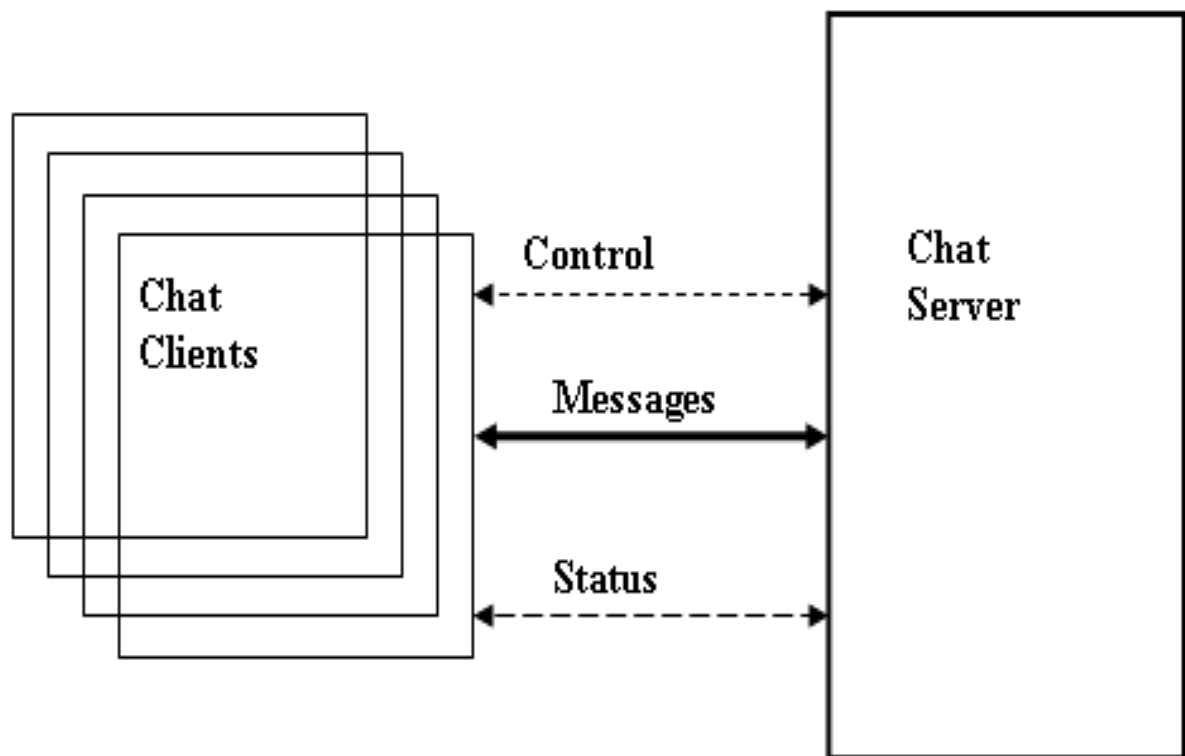
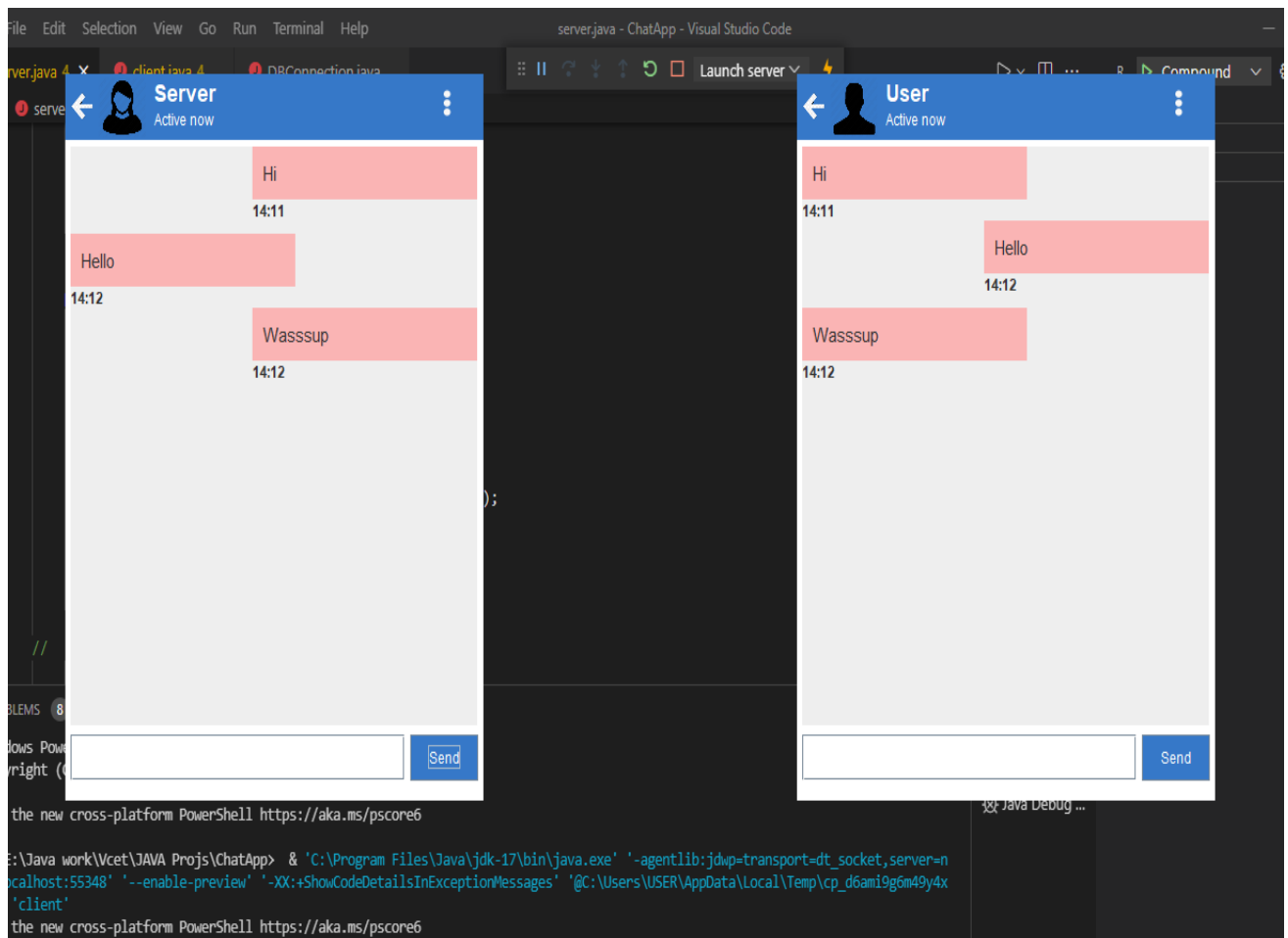


Figure 1: Chat Application Overview



▪ The Client Server Screen OR User Interface:



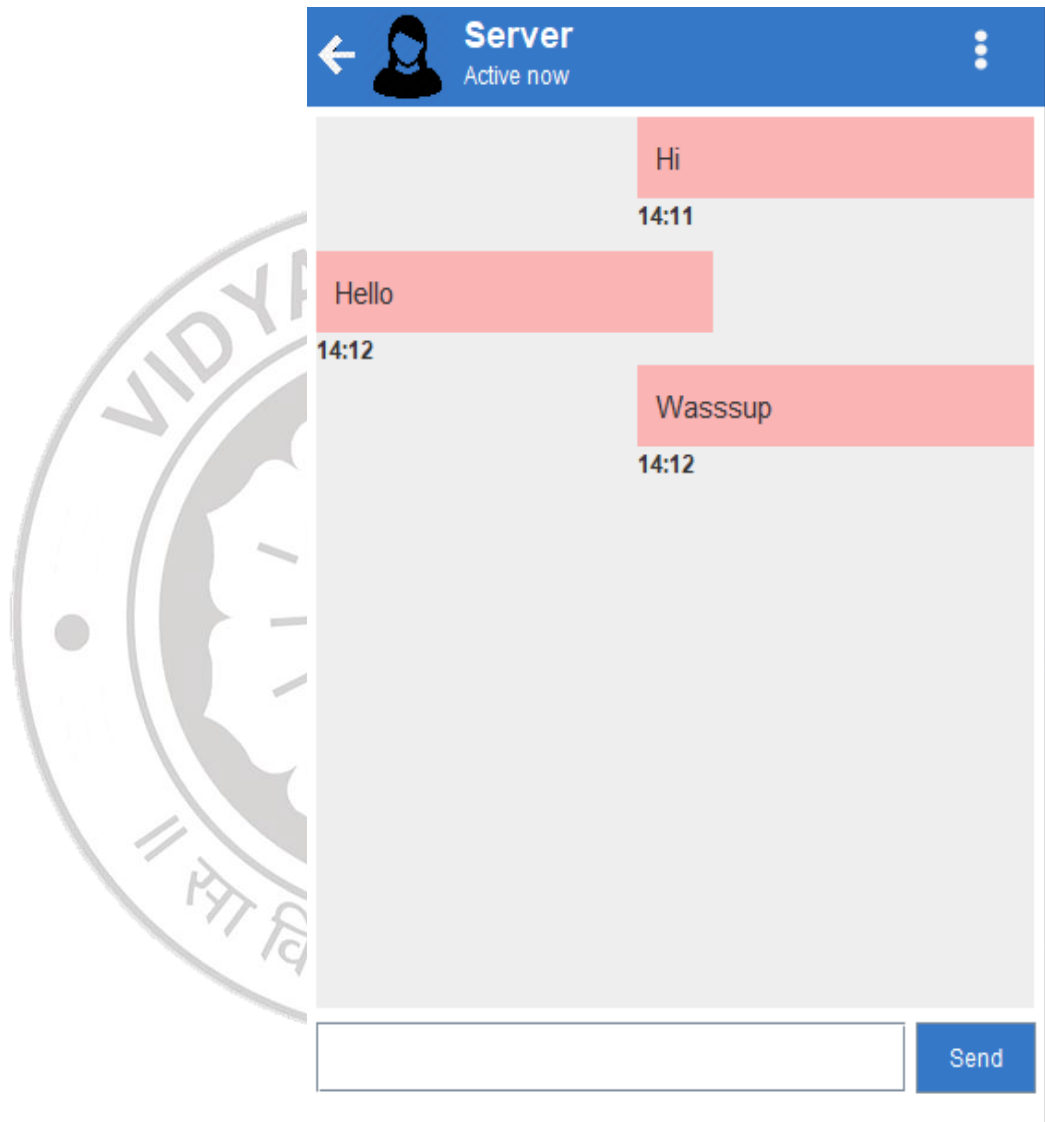
Database of project:-

✚ One on one chat:

We have use MySQL database as back end for our project. We have done JDBC – (Java Database Connection). It acts as the backup for the chat server all the messages are storage in the sql chat table. With the help of database user can find out the different chats or messages done at various time and it act as the chat history for the user.

Here, is the representation of the database for one-on-one chat server stated below:

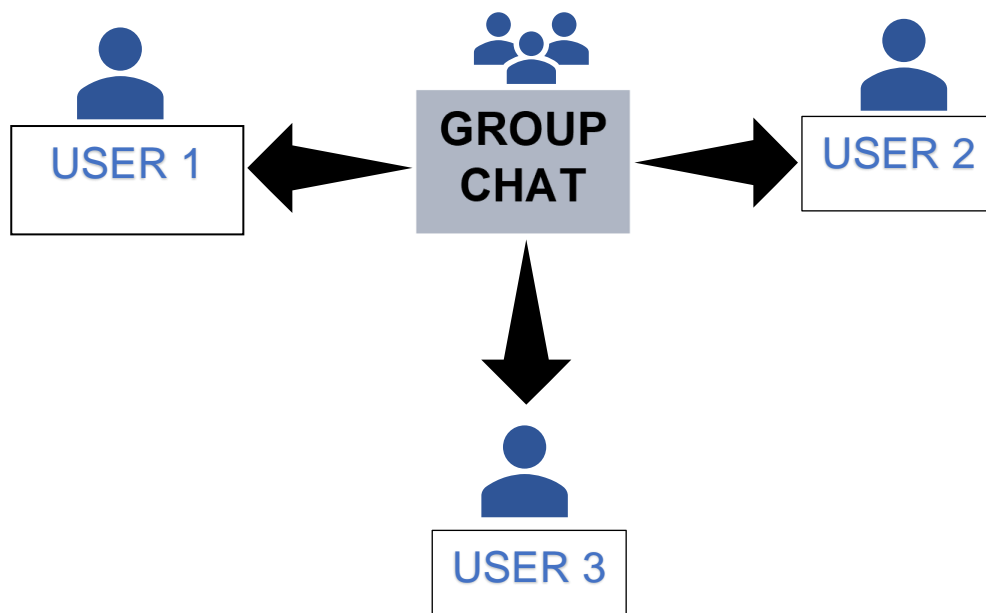
Database :



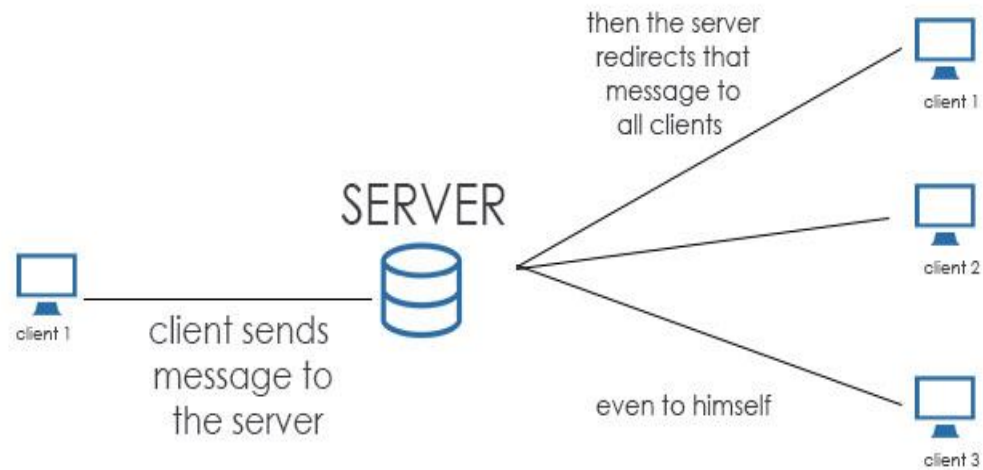
Result Grid		Filter Rows:		Export:
	date	time	name_user	chat
	05/12/2021	15:36	Server	Hi
	05/12/2021	15:37	User	Hello
	05/12/2021	15:37	Server	Wasssup

Flowchart for group chat –

All the users connect with server on same network and are able to pass messages. Further these messages are been store in MySQL database.



Scenario of group chat application:



Introducing threads in socket programming:

A simple date time server was created which handles multiple user requests at same time using threading. It explains the basic concepts of threading in network programming. The same concepts used in our chatting application for group chatting.

- Group Server class:

The main server implementation is easy and similar to the server client class used in one-on-one chat application.

The following points are to remember:

- 1) The server runs an infinite loop to keep accepting requests.
- 2) When a request comes, it assigns a new thread to handle the communication part.
- 3) The server also stores the client's name into vector, to keep a track to connected devices.

The vector stores the thread object corresponding to the current request. The helper class uses this vector to find the name of reception to which message is to delivered. As this vector holds all streams, user class can use it successfully deliver messages to specific clients.

4) Invoke the start () method.

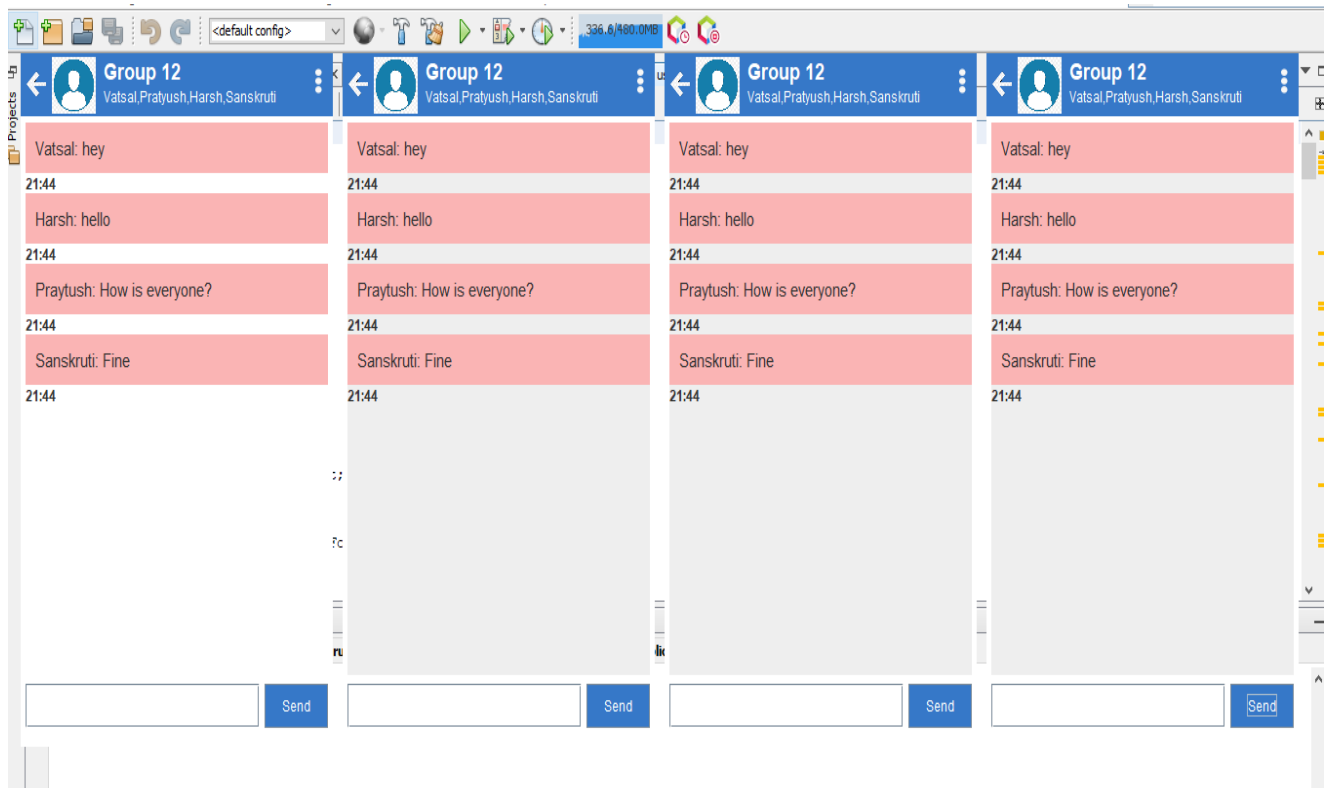
- User class:

We create helper class for handling various request. This time, along with socket and streams, we introduce a name variable. This will hold the name of the client that is connected to the server.



1. Whenever the user receiver any string, it breaks it into the message and recipient part. It uses StringTokenizer for this purpose with # as the delimiter.
2. If then searches for the name of recipients in the connected clients list, stored as vector in the server. If it finds the recipient's name in the clients list, it forwards the message on its output stream with the name of the sender prefixed to the message.

Here, is the representation of the database for group chat server stated below:

Preview of group chat Screen or user interface:



Database for group chat:

Result Grid  Filter Rows: <input type="text"/> Export:  W			
	date	chat_time	chat
▶	08/12/2021	21:44	Vatsal: hey
	08/12/2021	21:44	Harsh: hello
	08/12/2021	21:44	Praytush: How is everyone?
	08/12/2021	21:44	Sanskruti: Fine



Conclusion and Future Scope

Future Plan:

There is always a scope for improvement and adding on new this according to the demand of the public. But the most important thing should be flexible to accept further modification. In future the software maybe extended to include features such as:

➤ *Chat History:*

This will enable the user to restore the chat.

➤ *File Transfer:*

This will enable the user to send file of different formats to others via the chat application.

➤ *Voice Chat:*

This will enhance the application to a higher level where communication will be possible via video calling as in telephone.

➤ *Video Chat:*

This will further enhance the features of calling into video communication.

➤ *Voice command:*

This will enable to operate chat application on user's voice command.

Conclusion:

We have developed network application in java using swing and other different packages of java. This software is portable, efficient and easily maintainable for large number of clients and also stores the chat so that chats can be searched when required. Our developed web-based chatting software is unique in its features and more importantly easily customizable.



