

PROJECT Synopsis
ON
CHATING APPLICATION

SUBMITTED TO



J.C Bose University of Science and Technology, Faridabad

IN THE PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF THE DEGREE OF

BACHELOR OF TECHNOLOGY

IN

Computer Science and Engineering



Submitted To

Mr. Jyoti Kashyap
(Assistant Professor)

Submitted By

Raju Kumar
(22011004507)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
ARAVALI COLLEGE OF ENGINEERING AND MANAGEMENT, FARIDABAD - 121002
(JUNE-2025)

DECLARATION

I hereby declare that the project work entitled “Chatting Application” submitted to JC Bose University of Science and Technology Faridabad, Haryana (India), is a record of an original work done by us under the guidance of Mrs. Joyti (Assistant Professor) in Computer Science and Engineering, ARAVALI COLLEGE OF ENGINEERING AND MANAGEMENT, FARIDABAD, and this project work is submitted in the partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in COMPUTER SCIENCE AND ENGINEERING.

Dated

Student Name

Raju Kumar

ACKNOWLEDGEMENT

This project would not have taken shape without the guidance provided by **Ms. Joyti** Trainer who helped in the modules of our project and resolved all the technical as well as other problems related to the project and, for always providing us with a helping hand whenever we faced any bottlenecks, in spite of being quite busy with their hectic schedules. Above all we wish to express our heartfelt gratitude to **Ms. Sakshi Kumar**, H.O.D, CSE DEPARTMENT whose support has greatly boosted our self-confidence and will go a long way in helping us to reach further milestones and greater heights.

HOD

Ms Sakshi Kumar

Mentor

Ms.Jyoti Kashyap

Contents:

Abstract
Introduction
Main Objective
Requirement
Conclusion

INTRODUCTION

Chatting Application is a desktop based application.

This client server chat application is based on java swing and used socket package. its simple and easy and require only core java knowledge. I have taken this program from internet and modified a little bit to make it simpler and more elegant.

This application/program is a good example of using [java.io](#), [java.net](#) package to create a chat application. A beginner of java language, who is familiar with this packages can able, be beneficiate.

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 multiple client chat server.

It is made up of 2 applications the client application, 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. We will focus on TCP and UDP socket connections which are a fundamental part of socket programming.

Keywords: sockets, client-server, Java network programming-socket functions, Multicasting etc.

MAIN OBJECTIVE

The aim of this project is to express how we can implement a simple chat application between a server and a client? The application is a desktop based application and is implemented using Swing and awt. The project is developed in Java SE language executed on a single stand-alone java across a network using loop back address concept.

Application consists of two programs:

Server

Client

Server

The server module of the application waits for the client to connect to it. Then if connection is granted a client interacts communicates and connects to the server, it can mutually communicate with the server. The duty of the server is to let clients exchange the messages.

Client

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

REQUIREMENTS

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 graphics 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 of graphics libraries that enhances the AWT, was introduced as part of Java Foundation Classes (JFC) after the release of JDK 1.1. JFC consists of Swing, Java2D, Accessibility, Internationalization, and Pluggable Look-and-Feel Support APIs. JFC was an add-on to JDK 1.1 but has been integrated into core Java since JDK 1.2.

Software Interfaces

Programming Language Java

AND SOCKET PROGRAMMING

Operational Concepts and Scenarios

Operation of the application based on the inputs given by the user:

When the run button is clicked then the chat form is initialized with a connection

- between the host and the client machine.

Note: server must be started at first before a client start.

- Contains a rich textbox which send messages from one user to another
- Contains a textbox for messages to be written that is sent across the network.
- Contains a Send button
- When the sent button is clicked, in the background, the text in the textbox is encoded
- and sent as a packet over the network to the client machine. Here this message is decoded and is shown in the rich textbox.

Project Scope

This project can be mainly divided into two modules:

1. Server
2. Client

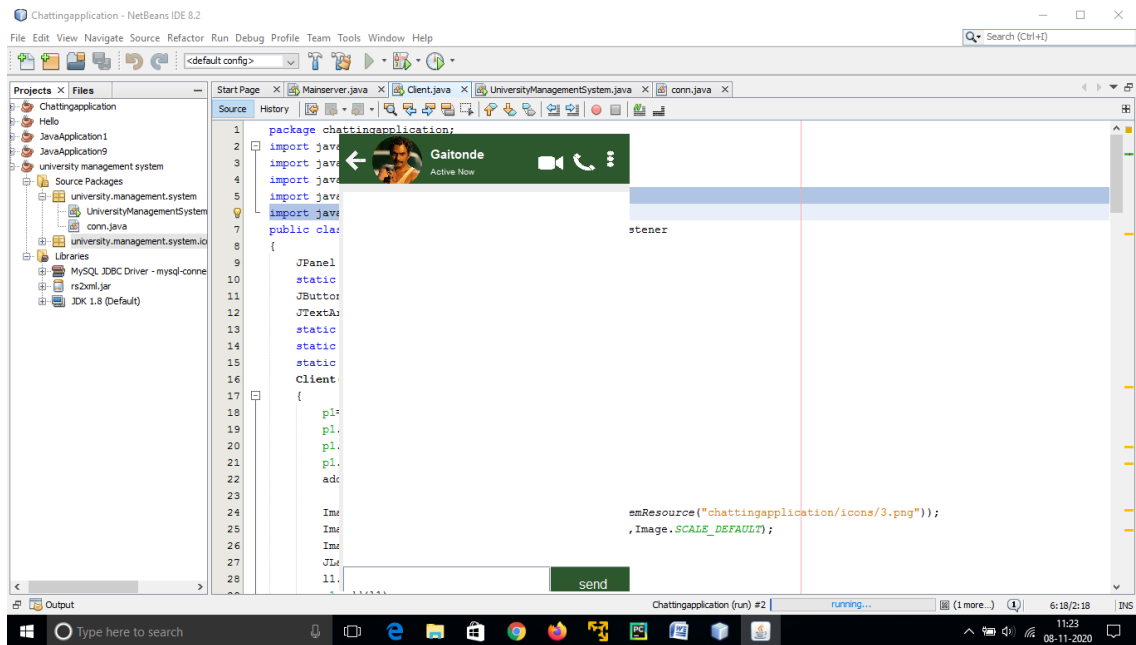
This project is mainly depended on client/server model. The client requests the server and server responses by granting the clients request. The proposed system should provide both of the above features along with the followed ones:

Server

A server is a computer program that provides services to other computer programs (and their users) in the same or other computers. The computer that a server program runs in is also frequently referred to as a server. That machine may be a dedicated server or used for other purposes as well. Example Server, Database, Dedicated, Fileserver, Proxy Server, Web Server. The server is always waiting for client's requests. The client come and go down but the server remains the same.

A server application normally listens to a specific port waiting for connection requests from a client. When a connection request arrives, the client and the server establish a dedicated connection over which they can communicate. During the connection process, the client is assigned a local port number, and binds a socket to it. The client talks to the server by writing to the socket and gets information from the server by reading from it. Similarly, the server gets a new local port number (it needs a new port number so that it can continue to listen for connection requests on the original port). The server also binds a socket to its local port and communicates with the client by reading from and writing to it. The client and the server must agree on a protocol that is, they must agree on the language of the information transferred back and forth through the socket. Normally, a server runs on a specific computer and has a socket that is bound to a specific port number. The server just waits, listening to the socket for a client to make a connection request.

1. THE SERVER SCREEN



Client

On the client site the client knows the hostname of the machine on which the server is running and the port number on which the server is listening.

To make a connection request, the client tries to rendezvous with the server on the server's machine and port. The client also needs to identify itself to the server so it binds to a local port number that it will use during this connection. This is usually assigned by the system.

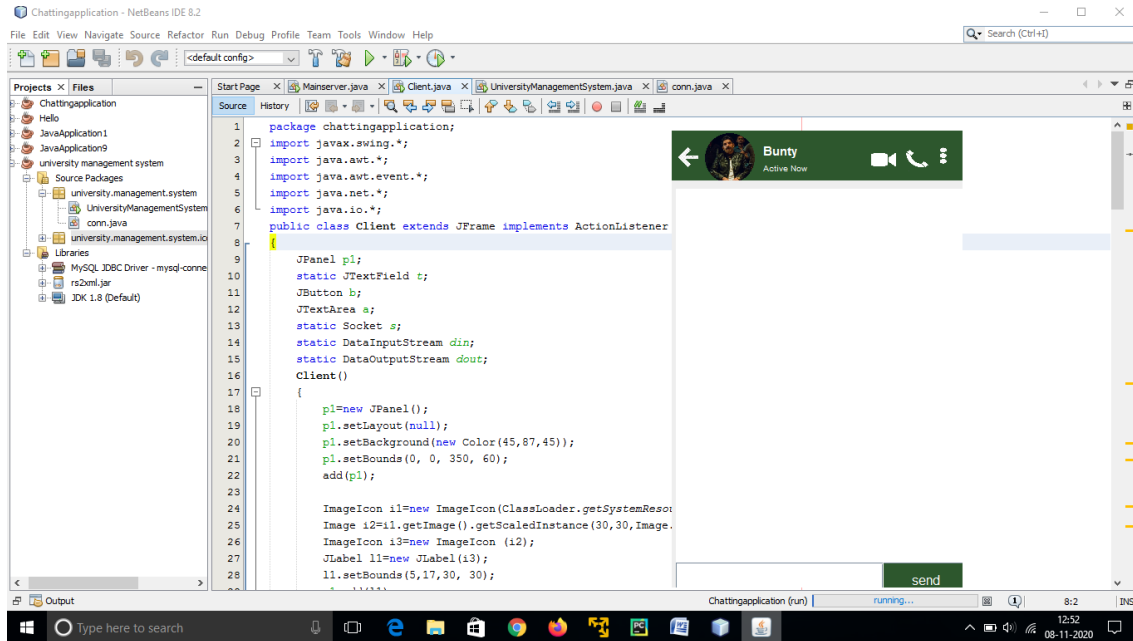
The model used for this project is the single server –

Single client models.

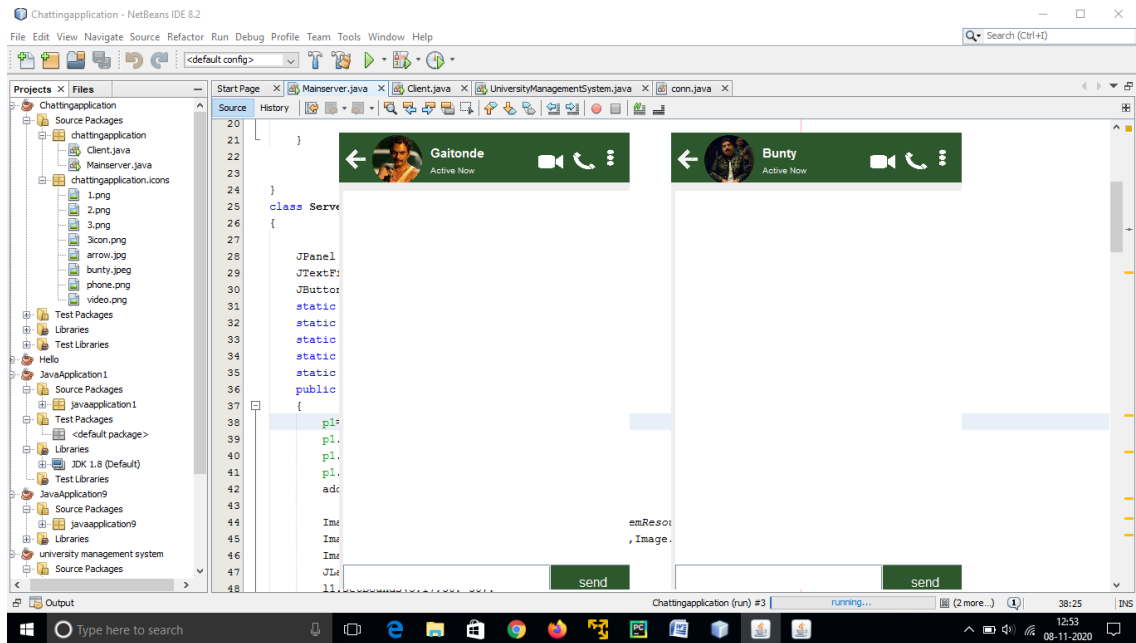
The following specifications must be implemented:

1. The server and client are two separate programs.

2. THE CLIENT SCREEN



3. THE CLIENT SERVER SCREEN



Future work

There is always a room for improvements in any software package, however good and efficient it may be done. But the most important thing should be flexible to accept further modification. Right now we are just dealing with text communication. In future this software may be extended to include features such as:

Files transfer: this will enable the user to send files 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 voice calling as in telephone.

Video chat: this will further enhance the feature of calling into video communication.

Conclusion

I Developed network applications in Java by using sockets, threads, and Web services. These software is portable, efficient, and easily maintainable for large number of clients. Our developed web-based chatting software is unique in its features and more importantly easily customizable. The java.net package provides a powerful and flexible set of classes for implementing network applications. Typically, programs running on client machines make requests to programs on a server Machine. These involve networking services provided by the transport layer. The most widely used transport protocols on the Internet are TCP (Transmission control Protocol) and UDP (User Datagram Protocol).

TCP is a connection-oriented protocol providing a reliable flow of data between two computers. On the other hand, UDP is a simpler message-based connectionless protocol which sends packets of data known as datagrams from one computer to another with no guarantees of arrival.