

---

## CHATTING APPLICATION

**Mr. Avinash Anil Pawar<sup>1</sup>, Mr. Saurabh Harinandan Mishra<sup>2</sup>,**

**Mr. Samarth Milind Gangawane<sup>3</sup>, Mr. Sabir Wali Hassan Khan<sup>4</sup>, Mr. V. B. Ohol<sup>5</sup>**

<sup>1,2,3,4</sup>Student, Information Technology, Sandip Foundation's Sandip Politechnic, Nashik, Maharashtra, India

<sup>5</sup>Professor, Information Technology, Sandip Foundation's Sandip Polytechnic, Nashik, Maharashtra, India

DOI: <https://www.doi.org/10.58257/IJPREMS31194>

---

### ABSTRACT

The Java-based intranet chatting application is a software system designed to enable users within an organization to communicate with each other using an internal network. This application provides a reliable and secure platform for communication between team members, departments, and project groups. The application is designed to be used in a client-server architecture with a central server that manages the communication between users. The application is built on Java technology, which makes it platform-independent and easily accessible to all users regardless of the operating system on their computer. It is designed to be user-friendly and intuitive, with a simple and modern user interface that enables users to easily navigate the system and quickly start chatting with their colleagues. The application includes various features such as group chat, file sharing, paging, and private messaging, which make it a comprehensive communication tool. The group chat feature allows users to create and join different chat groups, where they can discuss topics relevant to their work. The paging feature enables users to send important messages to their colleagues that require immediate attention, while the private messaging feature enables users to send messages to specific individuals. The application also includes an administrative control feature, which allows the administrator to manage users, chat groups, and user access permissions. The administrator can create and delete chat groups, add and remove users from groups, and control the access permissions of users. In terms of hardware and software requirements, the application requires a computer or laptop with Java Runtime Environment installed, an internet connection.

**Keywords** – Chatrooms, Platform Independent, internet

---

## 1. INTRODUCTION

In today's digital age, communication is an integral part of our lives. With the advent of the internet, people can now connect and communicate with anyone, anywhere in the world, with just a few clicks. The power of the internet is such that it has integrated together Local Area Networks (LANs) located across diverse software and hardware forms into a single large communication network that spans the globe. With the internet, people can share information, exchange ideas, and collaborate on projects from remote locations. As communication technology continues to advance, new tools and platforms are being developed to enhance and simplify the way people communicate online. One such tool is the "INTRANET CHATTING" project, which is specifically designed for chatting with remote clients or users on the internet or local networks. The project is developed using a Java client/server combination that enables users to chat with remote users. The Java programming language is known for its versatility, and it can run on any computer on the internet. This means that users can use the INTRANET CHATTING project to communicate with anyone, regardless of their location or device. The project's client/server model and whiteboard drawing utility are unique features that enhance the user experience and make it ideal for remote collaborations. The INTRANET CHATTING project is a must-have tool for anyone who values seamless and efficient communication in the digital age.

## 2. METHODOLOGY

The methodology used in the development of the intranet chatting application involved a series of steps, including requirements gathering, design, implementation, testing, and deployment. The requirements gathering process involved identifying the functional and non-functional requirements of the system through interactions with stakeholders and analysis of existing systems. Based on the requirements, a design was created using UML diagrams, including a use case diagram, behavior diagram, and component diagram, to guide the implementation process. The implementation involved using Java programming language and Eclipse IDE to develop the desktop application, which was tested extensively to ensure it met the specified requirements. The testing process included unit testing, integration testing, and system testing. Once the application was deemed satisfactory, it was deployed to the target environment, which involved setting up a server and client system. Finally, the application was documented, including the creation of user manuals and other documentation, and references were generated for any external resources used in the development process. In terms of features, this chatting application provides users with the ability to create and

participate in group chats, send and receive files, and exchange private messages. The admin controls allow for the management of user accounts, user permissions, and the monitoring of chat conversations. Additionally, the application allows users to save and view chat histories. Overall, the features of the chatting application were designed to promote efficient and effective communication among its users.

### 3. MODELING AND ANALYSIS

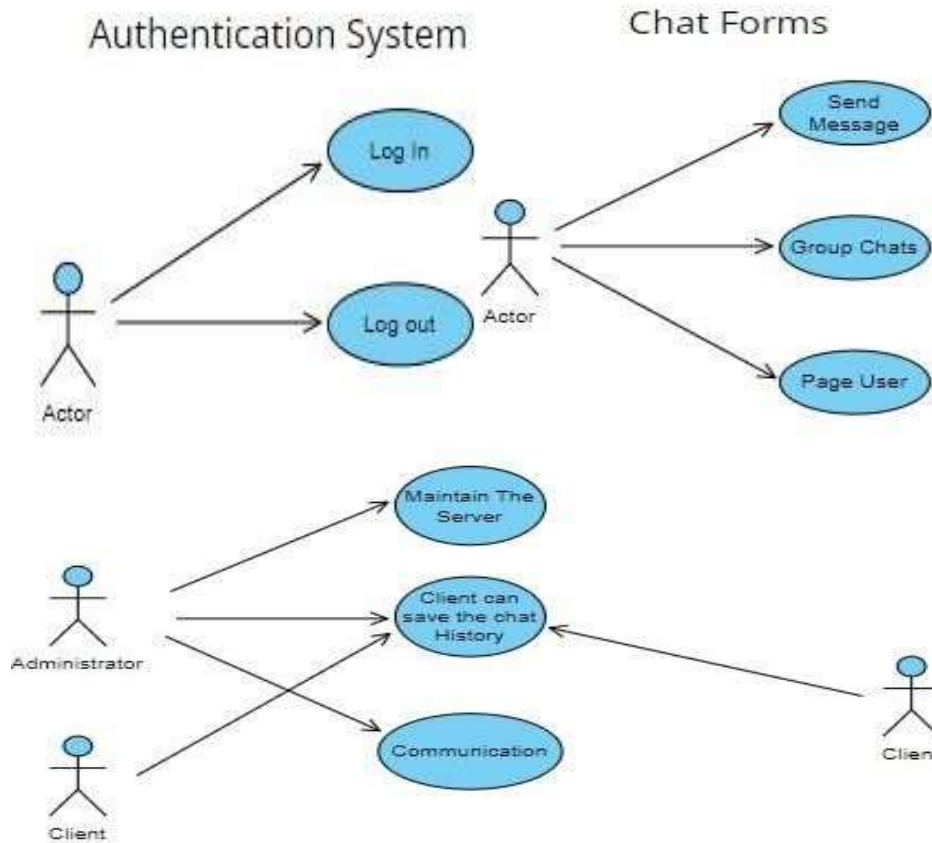


Figure 1: Use Case Diagram of Chat App

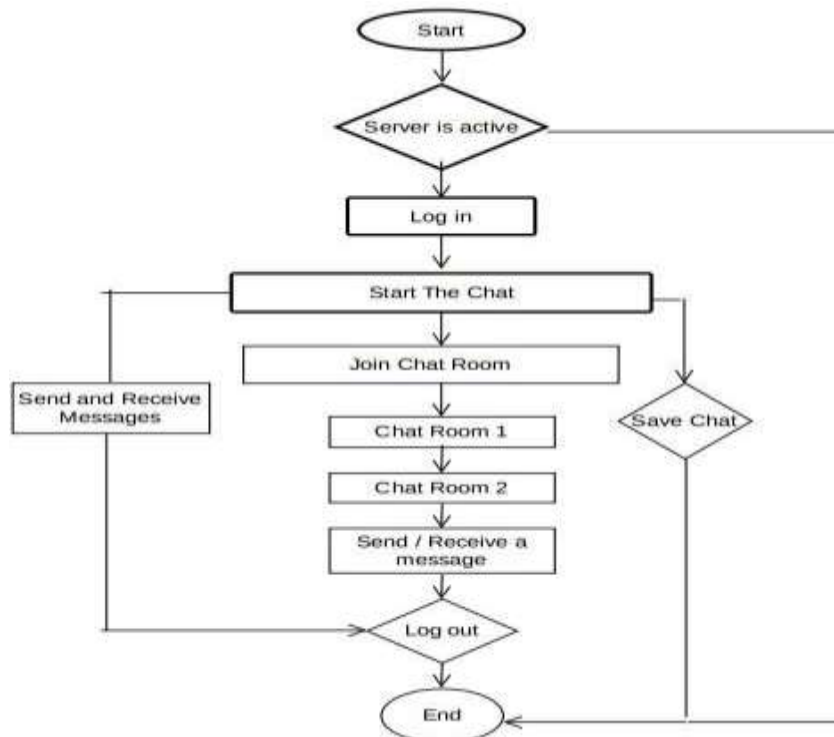


Fig 2

#### 4. RESULTS AND DISCUSSION

The INTRANET CHATTING application is developed used awt (Abstract Window Toolkit). The **java.awt** package is much useful for creating user interfaces and for painting graphics and images. A user interface object such as a button or a scrollbar is called, in AWT terminology, a component. The Component class is the root of all AWT components. Some components fire events when a user interacts with the components. A container is a component that can obtain components and other containers. A container can also have a layout manager that controls the visual placement of components in the container.

##### Server Started



Figure 3

The **java.awt** package implements different interfaces like **LayoutManager**, which defines the interface for classes that know how to layout Containers.

##### Client Login

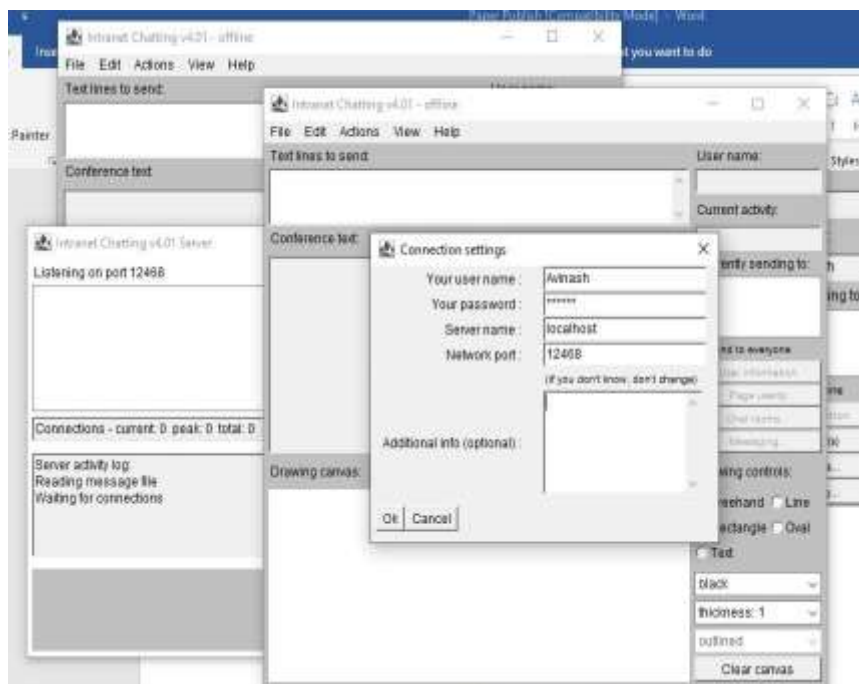


Fig 4

The user has to provide to which server it has to connect. In this, the server name to be provided is local host. The

user has to provide the username and the password for proceeding with the server for chatting purpose. The Client will be able to send message to anybody in the room with clients unique chat name created in the server for chatting purpose. By login in the server by entering Server Name and port with username and password.

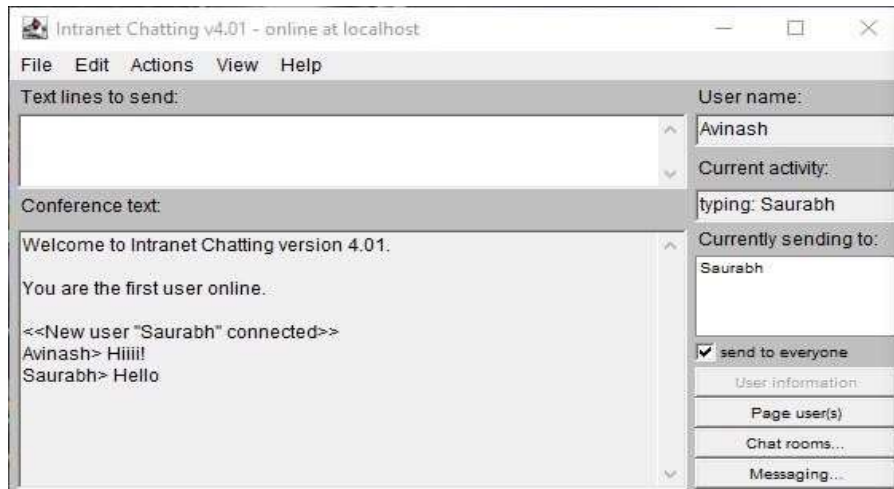


Fig 5

Clients communicating with each others

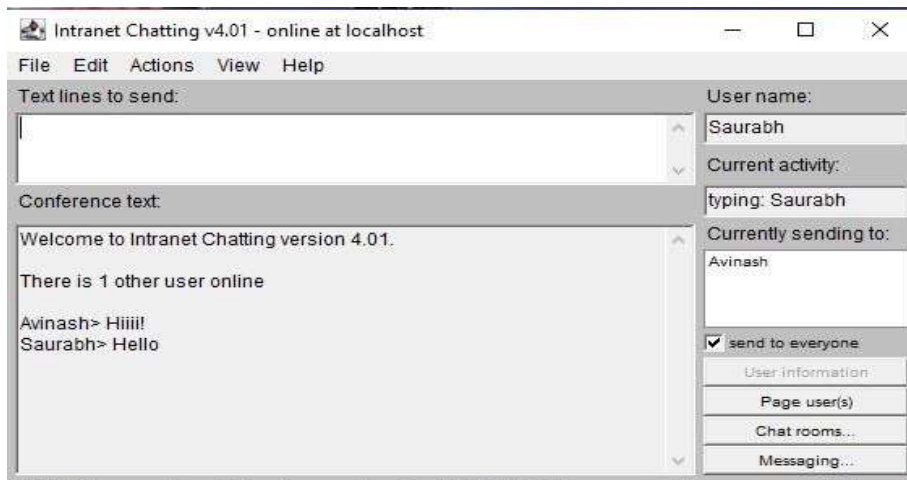


Fig 6

In all the network applications, we find two sort program where the first i.e., server sends the information and the second i.e., client receives the information.

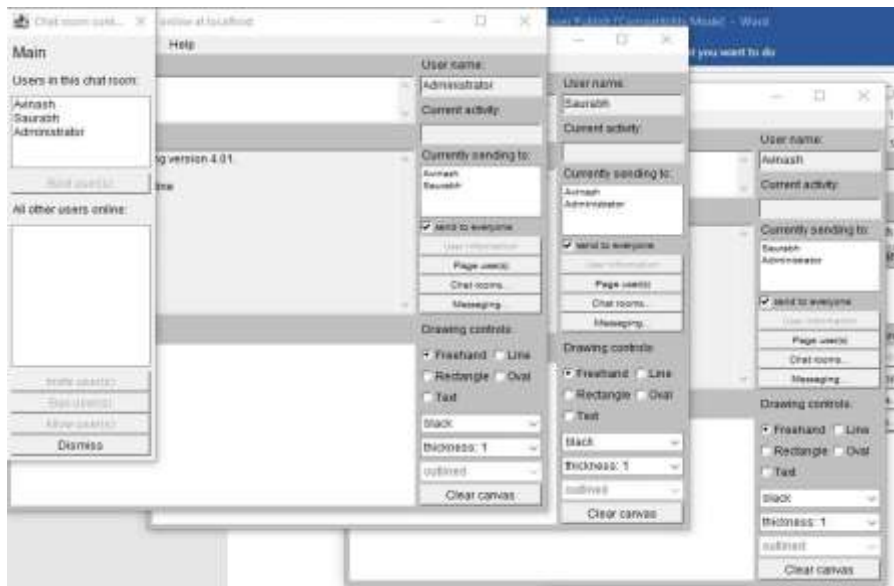


Fig 7

## Chat Modules

**Startup Tasks:** As the administrator, you are responsible for performing the initial startup tasks when beginning the chat server. This includes setting up the necessary configurations and ensuring that the server is running smoothly.

**User Management:** You have the ability to create new users and grant or revoke their access to the server. You can also manage their permissions and roles within the system.

**Session Management:** As the administrator, you can close the chat session, which will disconnect all users from the server. This may be necessary in situations where the server needs to be restarted or for security reasons.

**Authoritative Client:** As an administrative user, you can participate in chat conversations as a regular user using your administrator console. Shutdown: You have the ability to shut down the chat server when necessary, such as

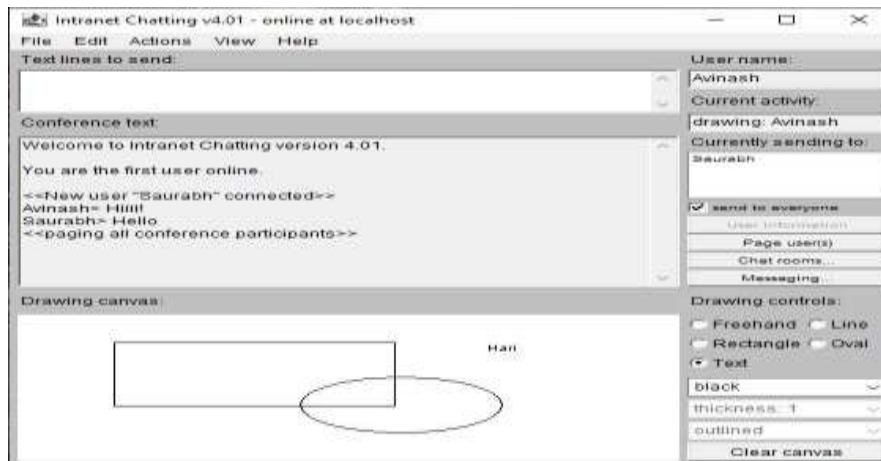


Fig 8

**Paint** interface defines how color patterns can be generated for Graphics2D operations. A class implementing the Paint interface is added to the Graphics2D context in order to define the color pattern used by the draw and fill methods. It gives a white board drawing office utilizing which the client can draw free hand, circles, rectangles, lines, texts, etc. Should be provided with the drawing tools like free hand, rectangle, oval, line and also sending text message over the room.

## 5. CONCLUSION

Even though this application has been developed with the users own Protocols, this can be used in an Intranet based organization.

- This system was developed so that people can exchange information as well as converse with each other.
- Through this system people can access chat rooms globally.
- The system is interactive and friendly.
- Entire system is fully automatic to the clients and satisfies the clients request
- Especially the system is more useful to the technical people when the need for sending pictures, images it is solved

through WHITE BOARD UTILITY OF “INTRANET CHATTING”.

## 6. REFERENCES

- [1] <https://docs.oracle.com/javase/8/docs/>
- [2] <http://en.wikipedia.org>
- [3] JavatPoint: <https://www.javatpoint.com>
- [4] Advance java Text Book
- [5] <https://stackoverflow.com/>
- [6] <https://www.javacodegeeks.com/>
- [7] <https://docs.oracle.com/en/java/javase/index.html>
- [8] <https://www.udemy.com/topic/java/>
- [9] <https://www.w3schools.com/java/default.asp>