Research paper on Group chatting Application

Research Proposal · May 2022

CITATIONS
0

READS
734

2 authors, including:

Ashutosh Kumar
Galgotias University
5 PUBLICATIONS **0** CITATIONS

SEE PROFILE

Research paper on Group chatting Application

Ashutosh Kumar
Galgotias University
(Under the Uttar Pradesh Private Universities Act No. 12 of 2019)
Greater Noida, India
Email: ashutosh.20scse1010626@galgotiasuniversity.edu.in

Atul Singh
Galgotias University
(Under the Uttar Pradesh Private Universites Act No. 12 of 2019)
Greater Noida,India
Email: atul.20scse1010218@galgotiasuniversity.edu.in

Under the guidance of

Mr. Shamesh Tabrej
Assistant Professor
(Galgotias University)

Greater Noida, India
Email: shamesh.tabrej@galgotiasuniversity.edu.in

ABSTRACT The latest development of the Internet has brought the world into our hands. Everything happens through Internet from passing information to purchasing something. Internet made the world as small circle. This project is also based on Internet. This paper shows the importance of chat application in day today life and its impact in technological world. This project is to develop a chat system based on Java multi threading and network concept. The application allows people to transfer messages both in private and public way .It also enables the feature of sharing resources like files, images, videos,etc. This online system is developed to interact or chat with one another on the Internet. It is much more reliable and secure than other traditional systems available. Java,multi threading and client-server concept were used to develop the web based chat application. This application is developed with proper architecture for future enhancement.

1. INTRODUCTION

Today Developers around the world are making efforts to enhance user experience of using application as well as to enhance the developer's workflow of designing applications to deliver projects and roll out change requests under strict time l ine. Stacks can be used to build web applications in the shortest span of time. The stacks used in web development are basically the response of software engineers to current demands. They have essentially adopted pre-existing frameworks (including Java Script) to make their lives easier. While there are many, MEAN and MERN are just two of the popular stacks that have evolved out of Java Script. Both stacks are made up of open source components and offer an end-to-end framework for building comprehensive web apps that enable browsers to connect with databases. The common theme between the two is Java Script and this is also the key benefit of using either stack. One can basically avoid any syntax errors or any confusion by just coding in one programming language, Java Script. Another advantage of building web projects with MERN is the fact that one can benefit from its enhanced flexibility. In order to understand MERN stack, we need to understand the four components that make up the MERN stack namely

–

Mongo DB, Express.js, React and Node. js.

Methodology

The whole idea of this proposed application is to avoid a centralized system (registration, login and buddy list) as that found in Skype (Baset and Schulzrinne, 2006; Azab et al., 2012). Using Skype, during registration, user profile will be stored in a centralized database and one can use the credential to login at anytime and anywhere as preferred. This certainly provides certain level of robustness although the question arises as to how secure the centralized database can be to prevent from attacks. Recently, a study on decentralized system was proposed but only for the purposed of improving the buddy list (Kundu, 2012). The idea was about developing a robust index system using distributed hash table for decentralized chat application. An indexing system is responsible for storing IP address and port of all users once they joined the chat. Users initialize their own buddy list by contacting the centralized indexing system once they logged in. When a user A wants to communicate to user B, B will act as a server and authenticate client A. As authentication is one-way, this opens up an opportunity for attackers to masquerade as user B. To cater some of these problems, in our proposed application, we come out with the following principle ideas:

Relation to External Environment

This tool helps in two major aspects -

• Resolving the names of all the system connected in a network and enlisting them.

• Used for communication between multiple systems enlisted in the resolved list.

System Architecture

The chat application works in two forms.

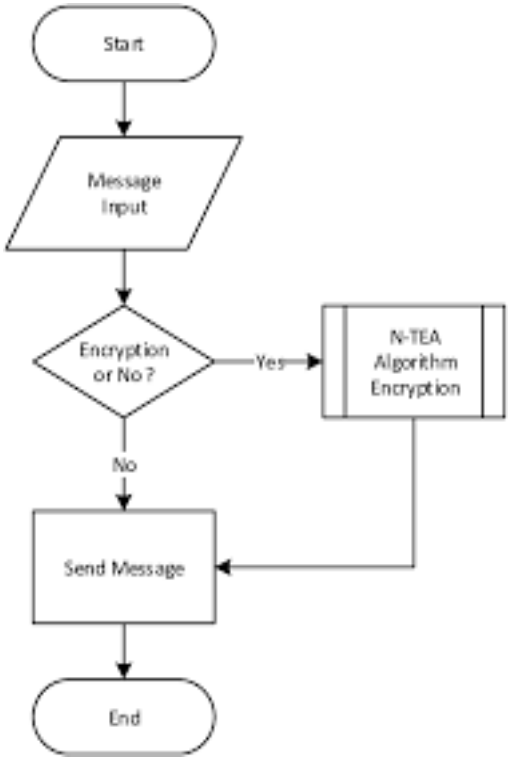
• **List form:**

In this form, all the names of the systems connected to a network are enlisted. These names can later be used for communication with the help of mouse event, or in simple language: a click or a double click.

•**Chat form**

This form is called only when an element is selected from the List form. In this form, a connection is created between the host system and the selected system with the help of a socket.

FLOW CHART



Operational Concepts and Scenarios

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

List Form:

•When initialized, returns a list containing the names of all the system connected in a network.

•Contains two buttons: Refresh and Connect.

• When Refresh button is clicked refreshes the list of names.

• When the Connect button is clicked or a name is double clicked, the chat form is initialized with a connection between the host and the client machine, Note: If no name is selected, and connect button is clicked an error box is displayed.

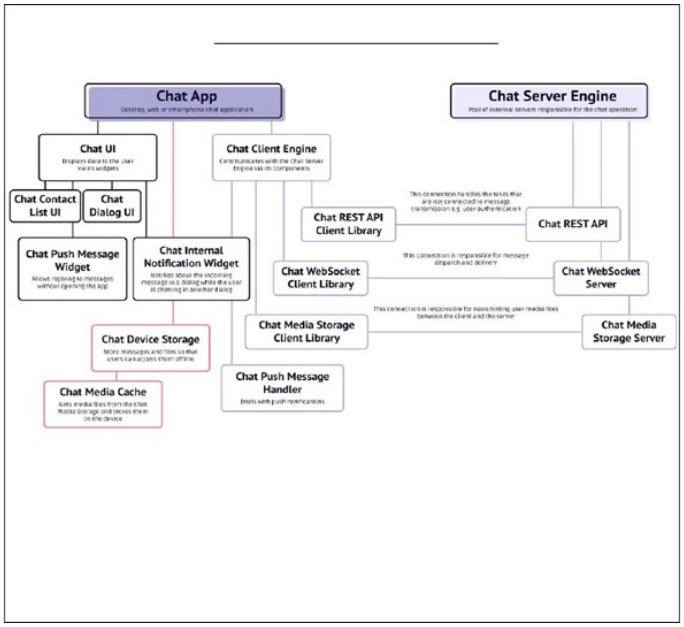
Architecture

SERVER

A server may be a computer dedicated to running a server application. Organizations have dedicated computer for server application which has to be maintained periodically and has to be monitored continuously for traffic loads would never let them go down which affects the company's revenue. Most organizations have a separate monitoring system to keep an eye over their server so that they can find their server downtime before its clients. These server computers accept clients over network connections that are requested. The server responds back by sending responses being requested. There are many different server applications that vary based on their dedicated work. Some are involved for accepting requests and performing all dedicated works like business application servers while others are just to bypass the request like a proxy server. These server computers must have a faster Central processing unit, faster and more plentiful RAM, and bigger hard disc drive. More obvious distinctions include redundancy in power supplies, network connections, and RAID also as Modular design.

CLIENT

A client is a software application code or a system that requests another application that is running on dedicated machine called Server. These clients need not be connected to the server through wired communication. Wireless communication takes place in this process. Client with a network connection can send a request to the server.



Chat application or Client Side

•**Chat application** is the other major part of the chat architecture, the one that users directly interact with. It's split into two separate root components:

•**Chat Client Engine** handles all the communication with the Chat Server Engine via its internal components: Chat REST API Client Library and Chat Web Socket Client Library.

•**Chat UI** displays data to users:
Chat Contact List UI, Chat Dialog UI

DESCRIPTION

1. A static Server socket is created in beginning which is then bind with host and port .
2. After server instantiation Socket in particular host, it begins to listen in the particular port. Then the server is made to accept the request from the client through the particular port.
3. After starting the server, it can accept the requests from clients.
4. The socket is instantiated in client side to connect to the server.
5. A new Server Thread using socket is created to accept all the requests from multiple clients.
6. After accepting the request both read and write operation occurs simultaneously, clients who request the server can communicate with each other and share resources.
7. After finishing the communication the socket is closed both in the client and server side.

Dependencies

The third-party package or modules installed using
Npm are specified in this segment. The package. json file is the heart of Node.js system. It is the manifest file of any Node.js project and contains the metadata of the project. The package. json file is the essential part to understand, learn and work with the Node.js. It is the first step to learn about development in Node.js.

Component:

Components are the building blocks of any React app and a typical React app will have many of these. Simply put, a component is a JavaScript class or function that optionally accepts inputs i.e. properties(props) and returns a React element that describes how a section of the UI (User Interface) should appear.

Chat Server Engine

This is a core of the chat architecture that handles message delivery and dispatch. In our version of chat architecture, it includes the following components

•Chat REST API

handles the tasks that are not connected directly to message dispatch and delivery, such as user authentication, changing of user settings, friends invitation, downloading sticker packs, etc. The Chat App (the chat client part) communicates with the Chat REST API via the
Chat REST API
Client Library

• **Chat WebSocket** Server is responsible for transmitting messages between users. The Chat App communicates with the Chat WebSocket Server via the Chat WebSocket Client Library. This connection is open two ways; that means users don't have to make requests to the server if there are any messages for them, they just get them right away

System Implementation and maintenance-

Implementation

Implementation is a vital step in ensuring the success of new system even a well designed system can fail if it is not a properly implemented. Implementation activities are needed to transform a newly developed information system into an operational system for end users.

Acquiring Hardware Software And Services:

These resources acquired from many sources in the computer industry. Some sources are as follows
a-hardware- IBM, HP, Apple computer etc.
b-software- Microsoft, Oracle etc..

Testing-

Testing of a developed system is an important implementation activity. System testing and debugging computer programs and testing information processing procedures.

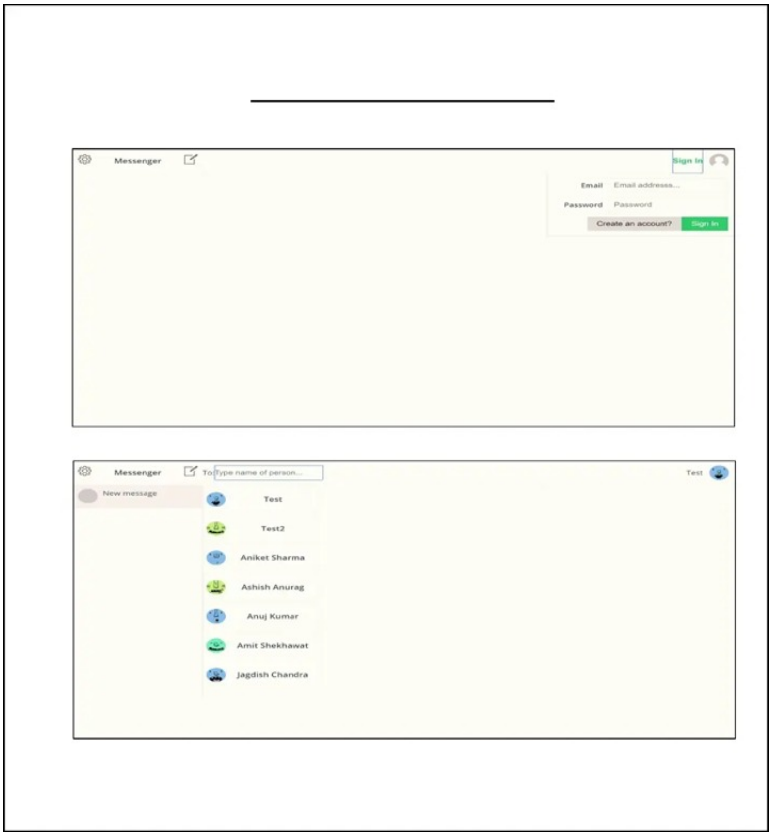
Training Method-

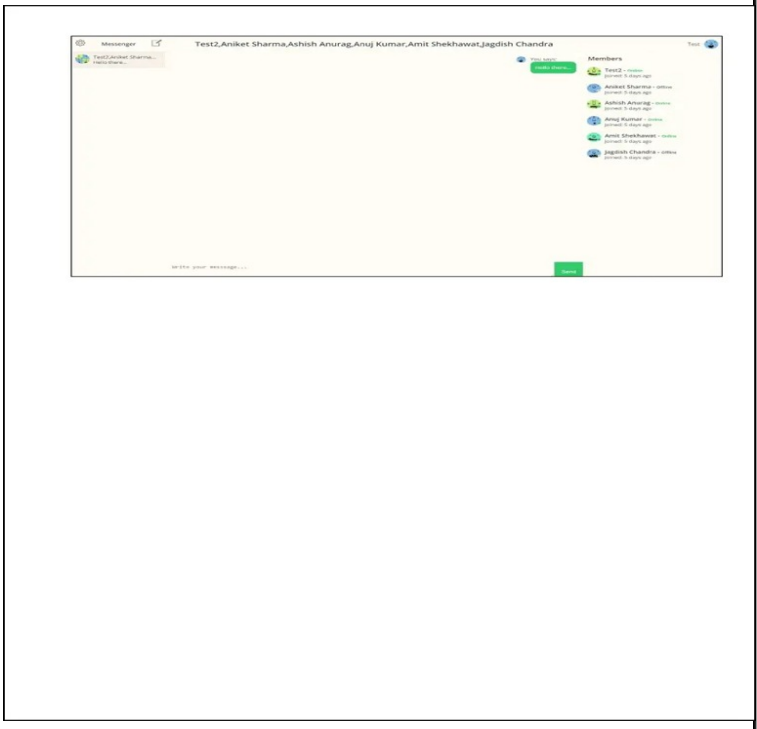
A-Vendor and in service training-Vendor offers extensive educational programs as part of their services the courses by experiences trainers and sales personnel cover all aspects of using the equipment participant actually use the system in the presence of trainer. It questions arise, they can quickly be answered.

Maintenance-

Once a system is fully implemented & being operated by end user & the maintenance function begins System maintenance is the monitoring to evaluating &modifying of operational information system to make desirable or necessary improvements for example the implementation of a new system usually results in the phenomenon known as the leaning curve person who operates &use the system will make mistakes simply because the are not familiar with it. Though such errors usually diminished experience is gained with a new system ,they do point out areas where system may be improved. Maintenance is also necessary for the failures and problems that arise during the operation of a system the maintenance activity includes a post implementation of a system review process to ensure that newly implemented system meet the system development objective established for them. Errors in the development of a system must be corrected by the maintenance process this includes a periodic review or audit of a system to ensure on operating system &meeting its objective.

SCREENSHORT





Conclusion-

There is always a room for improvements in any apps. Right now, we are just dealing with text communication. There are several chat apps which serve similar purpose as this project, but these apps were rather difficult to use and provide confusing interfaces. A positive first impression is essential in human relationship as well as in human computer interaction. This project hopes to develop a chat service Web app with high quality user interface. In future we may be extended to include features such as:

• File Transfer

. Voice Message

Video Message

. Audio Call

Video Call

• Group Call

REFERENCES-

1. https://www.ijeat.org/wp-content/uploads/papers/v9i5/E9578069520.pdf
2. http://indusedu.org/pdfs/IJREISS/IJREISS_3661_55346.pdf
3. https://thescipub.com/pdf/jcssp.2015.723.729.pdf
4. https://www.ijrte.org/wp-content/uploads/papers/v7i5s2/ES2063017519.pdf
5. https://core.ac.uk/download/pdf/187726106.pdf

.