**System Design Document**

**Introduction**

**Purpose**

This document provides a detailed design for the Chat Application, including system architecture, components, and the setup process for running the prototype.

**Scope**

The Chat Application is a real-time messaging platform developed using the MERN stack (MongoDB, Express.js, React.js, Node.js) and Socket.IO. The application supports dynamic chat rooms, user authentication, notifications, and administrative controls.

**Audience**

This document is intended for developers, system architects, and administrators who will be involved in setting up, maintaining, and extending the Chat Application.

**System Architecture**

High-Level Architecture

The system is structured into the following component:

**Frontend (Client-side)**

* + React.js: For building a dynamic and responsive user interface.

And allows for component-based architecture and efficient UI updates.

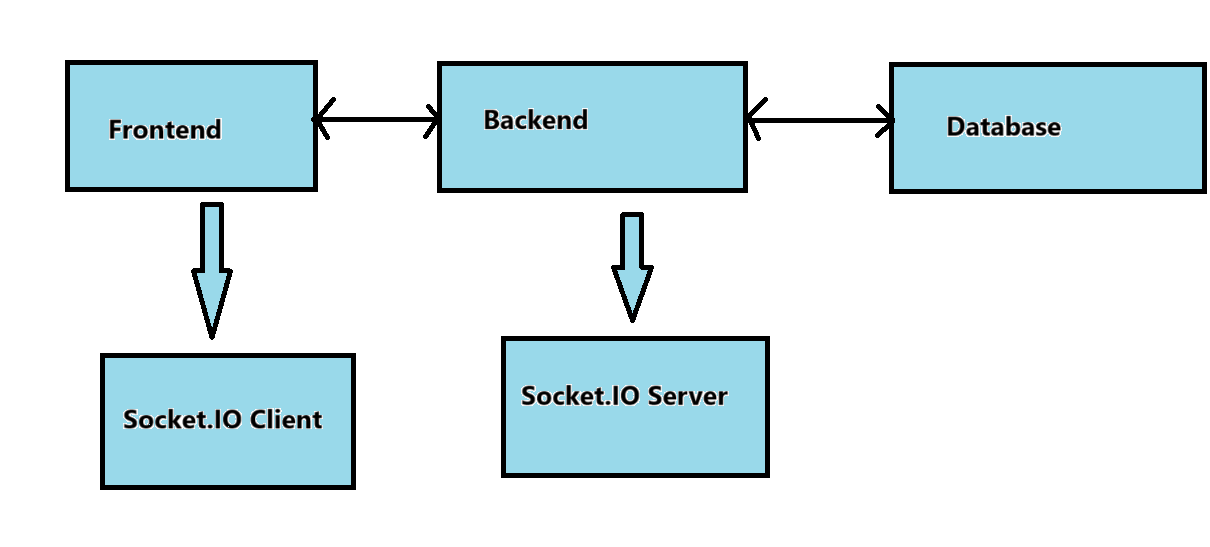
* + Socket.IO Client: For real-time communication with the server.

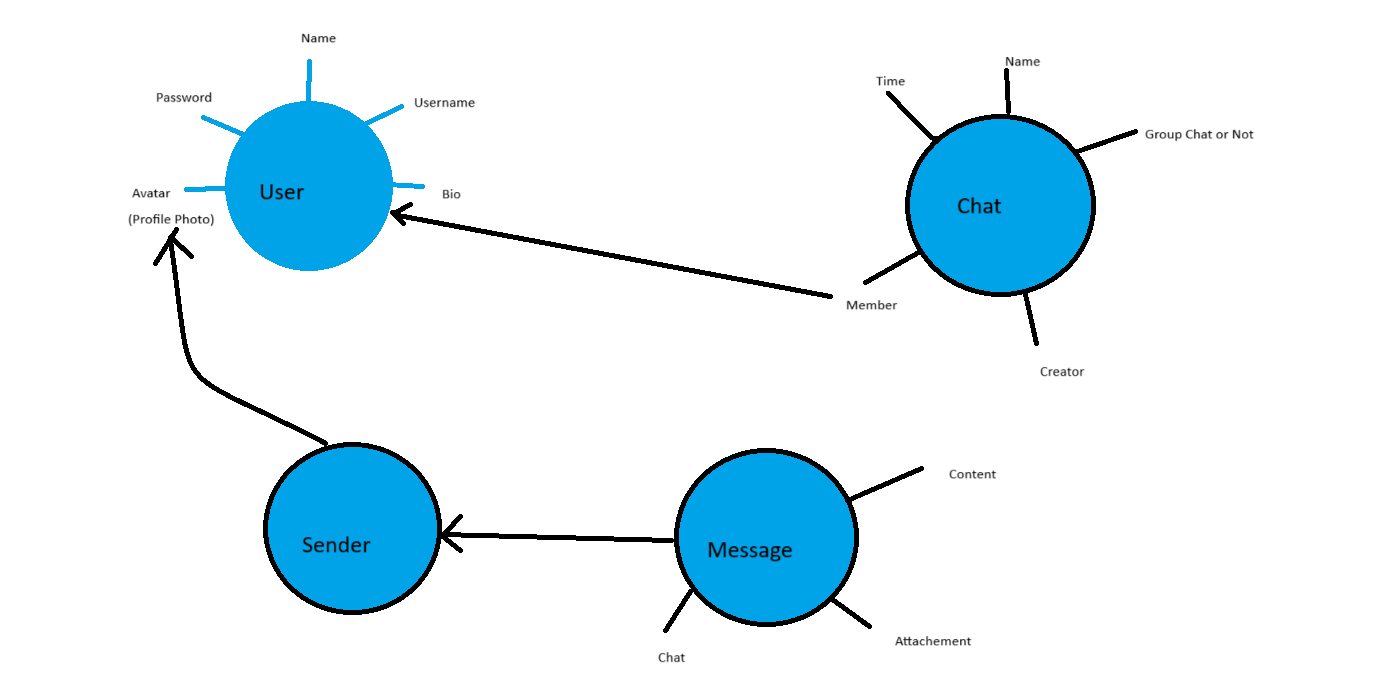
**Backend (Server-side)**

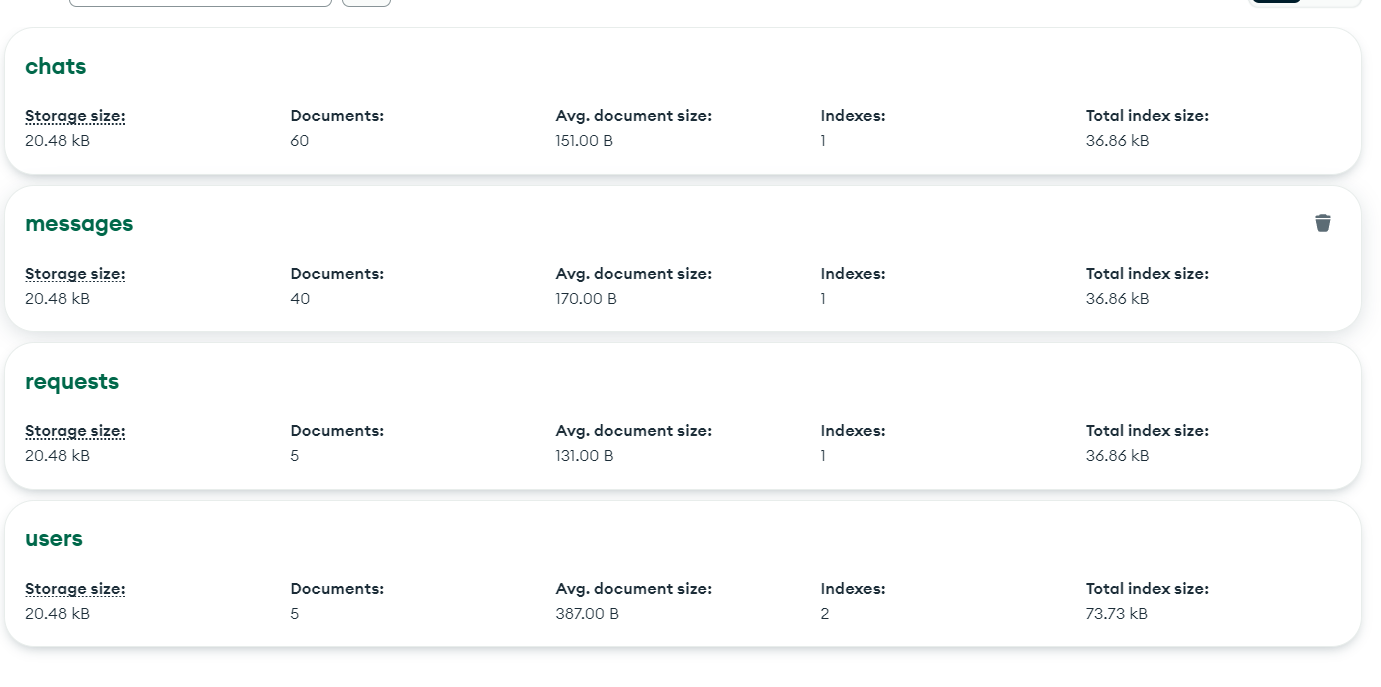
* + Node.js: For executing server-side JavaScript.
  + Express.js Handles RESTful API requests, routing, and middleware..

**Socket.IO Server:** For managing real-time, bidirectional communication between clients and the server.

**Components Diagram**

****

**Data base management**

****

**Dependencies and Libraries**

1. **For Backend**

Nodemon

Express

Dotenv

Express-validator

Multer

Soket.io

Bcrypt

Cloudinary

Cookies-parser

Cors

**2.For Frontent**

React-router-dom

React-host-toast

React-helmet-async

React-chartjs-2

Reduxjs/toolkit

[@reduxjs/toolkit](https://redux-toolkit.js.org/)

[@mui/x-data-grid](https://mui.com/x/react-data-grid/getting-started/)

[@mui/material](https://mui.com/)

[@mui/icons-material](https://mui.com/material-ui/material-icons/)

[@emotion/react](https://www.npmjs.com/package/@emotion/react)

**Future Enhancements**

**Enhanced Security**: Implement additional security measures, such as rate limiting and advanced encryption.

**Scalability:** Consider deploying the application using cloud services and load balancers for better scalability.

**User Experience Improvements**: Add more features, such as Video calling and voice calling and emoji support, to enrich the user experience.