Team Chat Application

Group Name: Group 7

Group Members:

1. Shivanandha reddy vasudevula (11709232)
2. Srichandan Kota (11711406)
3. Kantumutchu Dinesh(11638076)
4. Gana Deekshith (11614888)
5. Sandeep Chowdary Ari(11700315)
6. Bhanu Prasad Krishna Murthy(11654250)
7. Swapna Sonti (11653211)
8. Venkata sai shankar koppula(11692147)

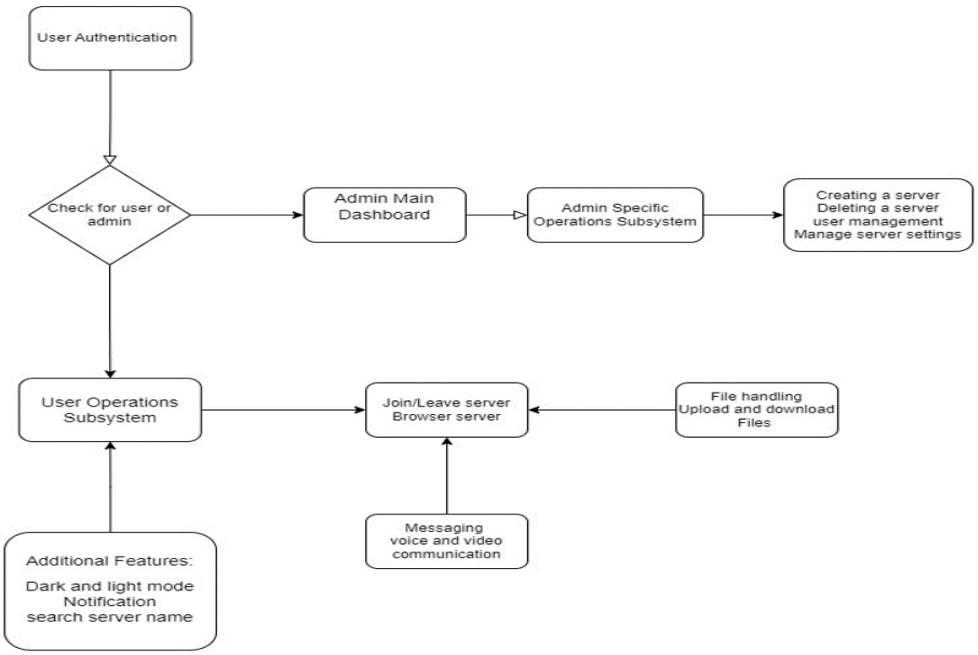
Table of Contents

Team Project
Deliverable 2 – Requirements Specifications
CSCE 5430 (Fall 2023)

1. System Overview

1.1. Diagram and Structure:

The system is structured to guide users from authentication through to engaging with a multitude of features within the application. It starts with User Authentication, where both regular users and admins identify themselves. Upon authentication, users are directed to the Main Dashboard, which acts as a central hub for accessing different functionalities based on their roles (admin or user). Admins have privileges for Server and Channel Operations, as well as User Management, while regular users participate in Messaging, File Handling, and Voice/Video Communications within servers and channels they have access to. The structure supports additional features like Search, Notifications, and theme toggling (Dark/Light Mode), enhancing user experience and interaction within the platform. The following is an overview of the system architecture:



The architecture consists of a number of linked components that together provide reliable and effective functioning.

1.2. Overview of System Components (Sub systems):

1.2.1. User Authentication:

Manages the login and signup processes.

Differentiates between regular users and admins, granting appropriate access levels.

1.2.2. Main Dashboard:

Serves as the landing page post-authentication, directing users to different areas of the application based on their roles.

1.2.3. Admin Specific Operations:

Create/Delete Server: Allows admins to initiate or remove communities.

Create/Delete Channel: Enables channel setup within servers for specific types of communication (text, voice, video).

Manage Server Settings: Admins can adjust server configurations and settings.

User Management: Admins can kick, ban users, or assign roles within the server for governance.

1.2.4. User Operations:

Join/Leave Server: Users can browse and select servers to join or leave.

Browse Servers and Channels: Users navigate through the available servers and their channels.

1.2.5. Messaging in Text Channels:

Supports sending, receiving, editing, and deleting messages.

Includes functionalities like mentions, emojis, and notifications for an enriched chatting experience.

1.2.6. File Handling in Channels:

Allows for the upload and download of files.

Supports file previewing, especially for images and PDFs, within the chat interface.

1.2.7. Voice and Video Communications:

Join/Leave Voice Channel: Users can participate in voice chat rooms.

Start/Join Video Call: Facilitates video conferencing within channels.

Screen Sharing: Users can share their screens during voice or video calls.

1.2.8. Additional Features:

Search Functionality: Enables searching for users, messages, or content within the platform.

Notifications: Alerts users about new messages, mentions, or other significant activities.

Dark/Light Mode: Allows users to toggle between themes for personal preference.

1.3. Relationships Between Subsystems:

1. User Authentication Subsystem:

Features:

User Login and Registration

Session Management

Role-based Access Control (Admin vs. User)

Relations:

To Main Dashboard: Grants access based on authentication status and user role.

To Admin/User Operations: Determines the level of access and permissions available to the user within the application.

2. Main Dashboard Subsystem:

Features:

Navigation Interface to Access Different Parts of the Application

Display of Available Servers and Channels

User Profile and Settings Access

Relations:

From User Authentication: Receives authenticated users and directs them based on their roles.

To Admin Specific Operations: Provides admins with pathways to manage servers, channels, and users.

To User Operations: Serves as the jumping-off point for users to join servers, engage in channels, and access additional features.

3. Admin Specific Operations Subsystem:

Features:

Server Creation and Deletion

Channel (Text, Voice, Video) Management

User Management (Kick, Ban, Role Assignment)

Server Settings Configuration

Relations:

From Main Dashboard: Admins access these operations via the dashboard.

To Database Subsystem: Directly interacts to create, update, or delete records related to servers, channels, and users.

To Messaging and File Handling: Influences the structure and availability of channels for messaging and file sharing.

4. User Operations Subsystem:

Features:

Joining and Leaving Servers

Navigating through Channels

Engaging in Community Activities

Relations:

From Main Dashboard: Users navigate to different servers and channels through the dashboard.

To Messaging in Text Channels: Engages in text messaging within selected channels.

To Voice and Video Communications: Participates in voice and video channels for real-time communication.

5. Messaging in Text Channels Subsystem:

Features:

Real-time Text Messaging

Message Editing and Deletion

Supports Mentions, Reactions, and Emoji

Relations:

To User Operations: Forms the core interaction within text channels.

To File Handling in Channels: Complements text messages with the ability to share and receive files.

To Database Subsystem: Stores and retrieves message data.

6. File Handling in Channels Subsystem:

Features:

File Upload and Download

Image and PDF Preview

File Management within Messages

Relations:

To Messaging in Text Channels: Integrates file sharing within messaging.

To Database and File Storage: Manages file metadata in the database and stores files in designated storage.

7. Voice and Video Communications Subsystem:

Features:

Real-time Voice and Video Calls

Screen Sharing

Voice Channel Participation

Relations:

To User Operations: Users join voice/video channels through user operations.

To External Services (e.g., LiveKit): Utilizes third-party services for managing real-time communication streams.

8. Additional Features Subsystem:

Features:

Search Functionality across Messages, Files, and Users

Notifications for Messages, Invitations, etc.

Theme Toggling (Dark/Light Mode)

Relations:

To All Subsystems: Cross-cuts the entire application to enhance user experience, providing functionalities that interact with messaging, file handling, user management, and more.

9. Database Subsystem (Implicitly Included in Relations):

Features:

Stores Users, Servers, Channels, Messages, and File Metadata

Manages Data Integrity and Relations

Relations:

To Nearly All Subsystems: Acts as the backend foundation, storing and serving data to the application, critical for the functionality of user management, messaging, and media sharing.

2. Functional Requirements

2.1. User Authentication and Authorization

Description: A foundational layer ensuring secure access to the system. It differentiates between regular users and administrators, providing appropriate access and permissions.

Interactions:

Interacts with the Database to verify user credentials and session management.

Enables Role-Based Access Control (RBAC), affecting visibility and actions within the Server Management and Channel Management subsystems.

2.2. Server Management (Admin Specific)

Description: Enables administrators to create and manage virtual spaces for community interaction. Includes setting up server-wide rules and permissions.

Interactions:

Directly influences the User Operations subsystem by determining the servers available for users to join.

Works in tandem with Channel Management to structure the communication framework within each server.

2.3. Channel Management (Admin Specific)

Description: Allows for the creation and customization of channels within servers, catering to different modes of communication (text, voice, video).

Interactions:

Forms the backbone of the Messaging in Text Channels and Voice and Video Communications subsystems, providing the platforms for these interactions.

Interacts with Server Management to nest within created servers, offering structured communication paths.

2.4. User Operations

Description: Facilitates user engagement with the application, allowing navigation through servers and channels, and customization of user profiles.

Interactions:

Serves as the entry point post-User Authentication, directing users to available Servers and Channels.

Enhances user experience by integrating with Additional Features like search and notifications.

2.5. Messaging in Text Channels

Description: Core communication feature allowing real-time text-based conversations within channels. Includes advanced messaging functionalities like edits, deletions, mentions, and reactions.

Interactions:

Relies on Channel Management for the operational context.

Engages with File Handling to complement text messages with multimedia sharing.

Utilizes Database for storing and retrieving messages.

2.6. File Handling in Channels

Description: Supports sharing and management of files within text channels, enhancing communication with visual and document-based data.

Interactions:

Embedded within Messaging in Text Channels, providing a seamless experience for sharing and viewing files.

Depends on external File Storage solutions and the Database for managing file metadata.

2.7. Voice and Video Communications

Description: Expands communication capabilities through real-time voice and video channels, including features like screen sharing.

Interactions:

Built on top of the Channel Management framework, specifically utilizing voice and video channels.

May integrate with external services (e.g., LiveKit) for managing the complexities of live audio/video streaming.

2.8. Additional Features

Description: Encompasses supplementary functionalities like search, notifications, and theme toggling, enriching the overall user experience.

Interactions:

Cross-functional, impacting various subsystems by enhancing User Operations with Search and Notifications.

Theme toggling interacts with the frontend presentation layer across the entire application.

2.9. Data Management and Integrity

Description: Ensures the reliability and consistency of user-generated and system data across the application.

Interactions:

Central to all subsystems that create, update, delete, or query data, including Messaging, Server Management, and User Profiles.

Backed by the Database subsystem for executing data operations with integrity constraints.

3. Non-Functional Requirements

1. Performance Efficiency

Description: The system must ensure swift response times for user interactions, real-time messaging, and media sharing.

Interactions:

Affects Messaging in Text Channels, Voice and Video Communications, ensuring minimal latency.

Influences the backend and frontend architecture to optimize load times and data processing.

2. Scalability

Description: The application must be capable of handling an increasing number of users, servers, channels, and simultaneous connections without degradation in performance.

Interactions:

Impacts Server Management and Channel Management, requiring dynamic resource allocation.

Guides the infrastructure setup, including database and server configurations to support scaling.

3. Security

Description: Protect user data and interactions through secure authentication, data encryption, and compliance with privacy standards.

Interactions:

Integral to User Authentication, ensuring secure login and data protection.

Affects all data transactions, especially in File Handling in Channels and Voice and Video Communications, to encrypt messages and files.

4. Reliability

Description: The system should be highly available and recover gracefully from failures, with minimal downtime.

Interactions:

Relates to database operations and server functionality, ensuring Data Management and Integrity through backups and redundancy.

Affects the overall system architecture, requiring robust error handling and failover strategies.

5. Usability

Description: The application should provide an intuitive and user-friendly interface, allowing for easy navigation and access to features.

Interactions:

Influences the design and layout of the Main Dashboard and User Operations, focusing on user experience (UX) principles.

Guides the development of Additional Features like search functionality and notifications to be user-centric.

6. Maintainability

Description: The codebase and system architecture should be designed for ease of maintenance, updates, and future enhancements.

Interactions:

Impacts how Server Management, Channel Management, and other subsystems are developed, emphasizing modular design and clear documentation.

Affects development practices, including version control, testing, and deployment strategies.

7. Compatibility

Description: The system must work across different devices, browsers, and operating systems, providing a consistent user experience.

Interactions:

Affects frontend development in User Authentication, Messaging, and Voice and Video Communications, ensuring responsive design.

Guides testing strategies to cover a range of platforms and devices.

8. Accessibility

Description: The application should be accessible to users with disabilities, complying with WCAG and other accessibility standards.

Interactions:

Influences UI/UX design across User Operations, Messaging, and Additional Features, incorporating accessibility features like keyboard navigation and screen reader support.

9. Internationalization and Localization

Description: The system should support multiple languages and regional settings, allowing users from different geographic locations to use the application in their preferred language.

Interactions:

Affects UI elements across the system, requiring the ability to dynamically change text, formats, and other locale-specific settings.

10. Environmental and Operational Sustainability

Description: The project should consider environmental impacts, promoting energy-efficient operations and minimizing resource consumption.

Interactions:

Influences infrastructure choices and operational practices, favoring green hosting providers and optimizing resource usage across the system's architecture.

4. Interfaces

User Interfaces

Web Application Interface:

Description: Serves as the main platform for interaction, enabling users to engage in real-time messaging, server and channel navigation, and access advanced settings. Designed with usability and accessibility in mind, it supports features like creating servers, real-time messaging, editing, deleting messages, and uploading attachments.

Interactions: Directly interacts with the backend for authentication, message retrieval and sending, server management, and theme customization (dark and light mode).

The user interface is designed to ensure ease of use, accessibility, and a positive user experience. It includes:

React: Facilitates the creation of a dynamic and interactive UI, supporting features like real-time chat updates, live notifications, and smooth navigation between chat rooms or channels.

Tailwind CSS & ChatUI: These tools work together to provide a responsive and aesthetically pleasing design, tailored for chat functionalities such as message bubbles, user avatars, and theme customization (light/dark mode through Next Themes).

Video and Audio Call UI: Leveraging WebRTC (integrated within the application framework), the UI supports video and audio calls, offering users the ability to initiate private or group calls directly within the chat interface.

Hardware Interfaces

Considering laptops as the primary hardware:

Webcam and Microphone: Essential for video and audio call features, allowing users to engage in face-to-face conversations and group meetings within the application. The application is optimized for compatibility with standard laptop webcams and microphones.

Speakers/Headphones: For audio output during calls and notifications, ensuring clear audio transmission, the application is designed to automatically work with the default audio output devices of a laptop.

Device Compatibility and Peripheral Access:

Description: The application is designed to be compatible across a wide range of devices with different operating systems, screen sizes, and input methods. It also requires access to device peripherals like cameras and microphones for video and audio communication features.

Interactions: The interface ensures adaptability to hardware capabilities, such as utilizing **camera and microphone for voice and video calls** or live streaming within the application, ensuring users can fully utilize the communication features regardless of their device.

Software Interfaces

Database Interface (MySQL through Prisma):

Description: Manages all data storage and retrieval operations, including user data, chat messages, server and channel information. Utilizes Prisma as an ORM to simplify database interactions, ensuring efficient and secure data handling.

Interactions: Central to the functioning of user profiles, messaging systems, and server/channel setups, enabling dynamic content updates and storage.

Backend API (Next.js/Node.js with socket.io):

Description: Facilitates communication between the client-side application and the server, handling API requests for user actions, real-time messaging, and dynamic content updates. Implements socket.io for real-time bi-directional communication.

Interactions: Supports user authentication, message exchange, server and channel management, and live updates, ensuring a seamless real-time experience.

The software interfaces ensure robust back-end functionality, database interaction, and application logic:

Next.js: Manages server-side operations, SEO optimization, and integrates with WebRTC for handling video and audio calls, ensuring efficient data loading and real-time updates.

Prisma & MySQL/PlanetScale: These components are responsible for data persistence, managing user data, chat messages, call logs, and ensuring scalability and reliability for the application's database needs.

React Query: Enhances data synchronization between the client and server, managing the state of messages, calls, and user presence effectively, reducing the need for manual data management and increasing application performance.

Socket.io Client: Enables real-time, bi-directional communication between web clients and servers, used for real-time messaging and notifications.

Communication Interfaces

Critical for real-time interactions, these interfaces facilitate immediate **communication across the application:**

Socket.io: Powers real-time messaging, notifications, and live updates for user presence. It is also pivotal in signaling for WebRTC, enabling real-time video and audio calls within the application.

WebRTC (Web Real-Time Communication): Allows direct peer-to-peer communication for video and audio calls, supported by modern browsers and integrated into the application for seamless call functionality.

Real-Time Communication Protocol (socket.io):

Description: Enables instant messaging, notifications, and real-time updates across the application. It is pivotal for the chat functionality, allowing for real-time interactions between users within servers and channels.

Interactions: Integral to messaging, alerting users to new messages, and supporting live interactions like typing indicators and read receipts.

Authentication Service API (Clerk):

Description: Provides a secure and flexible authentication system, managing user sessions, and safeguarding user data. Essential for user registration, login, and session management.

Interactions: Directly integrated into the user sign-up and login processes, enhancing security and user management within the application.

Video Communication Service API (LiveKit):
Description: Facilitates direct peer-to-peer video and audio communications, enabling features such as video calls, audio channels, and screen sharing, thereby augmenting the application's communication capabilities.

Interactions: Seamlessly integrates with the chat application, allowing users to start and join video/audio calls directly within chat channels or private conversations.

Google Meet for Team Collaboration:
Description: Google Meet will serve as the primary communication platform for team meetings, discussions, and collaboration. It offers services for screen sharing, video conferencing, and real-time texting.

Rationale: Google Meet provides a smooth virtual collaboration environment that enables team members to interact, share accomplishments, and successfully address issues in order to improve cooperation and productivity.

Trello for Task Management and Tracking:
Description: The task creation and management, project tracking, priority setting, and workflow visualization using boards and cards will all be done using the web-based project management platform Trello.

Rationale: Trello's user-friendly visual project management interface offers a straightforward method to assign tasks, monitor progress, and set priorities. This improves the project's overall organization and effectiveness.

5. Development Phase Plan

Phase 1: Initiation and Planning (Jan 23 - Feb 5, 2024) - Done

Objectives & Requirements:

Project Scope Definition: Establish clear boundaries for what the project will deliver, including real-time messaging, file sharing, and advanced user management. This sets the direction and goals.

Team Formation: Assemble a multidisciplinary team with roles in development, UX/UI design, QA, and project management to cover all aspects of the project lifecycle.

Tools Setup: Selection of GitHub for collaborative code management and version control, alongside Jira for tracking tasks, bugs, and milestones, ensures efficient project tracking and collaboration.

Risk Management Framework: Develop a strategy to identify, assess, and mitigate potential risks throughout the project lifecycle, ensuring preparedness and resilience.

Criticality:

Initiation and planning are critical for aligning team efforts, setting realistic expectations, and laying the groundwork for successful project execution.

Phase 2: System Design and Tech Selection (Feb 6 - Feb 26, 2024)

Objectives & Requirements:

Prioritization of Functionalities (1 to 3, 1 being top priority and rest follows order):

1.Microservices Architecture: Designing the system around microservices enhances scalability and maintainability, allowing individual features to evolve independently.

2.Database & Tech Stack Selection: Opting for MySQL, Next.js, React, Socket.io, and Prisma ensures a robust, scalable, and developer-friendly environment. This choice supports real-time interactions and efficient data handling.

3.Security Planning: Implementing JWT for authentication and enforcing HTTPS protocols for data transmission safeguards the application against common security threats.

Criticality:

The prioritization of functionalities in this section is based on their criticality to the technical foundation of the project. Microservices architecture, database & tech stack selection, and security planning are crucial elements that significantly impact future development efficiency, system performance, and overall security posture. Neglecting these aspects may lead to scalability issues, performance bottlenecks, security vulnerabilities, increased development costs, and difficulties in maintaining and evolving the system over time. Therefore, ensuring the proper implementation of these functionalities is essential for the long-term viability and success of the project.

Phase 3: Core Functionality Development (Feb 27 - Mar 18, 2024)

Objectives & Requirements:

Prioritization of Functionalities (1 to 3, 1 being top priority and rest follows order):

1.Real-time Messaging & File Sharing: Implement foundational chat functionalities that support text, media, and file messages in one-on-one and group settings. This includes backend services for message delivery and frontend components for displaying messages.

2.User Management: Develop systems for user registration, profile customization, friend lists, and group memberships. This entails database schemas for user data and server-side logic for managing user interactions.

3.UI Customization: Introduce UI themes, such as dark and light modes, allowing users to personalize their experience. This requires frontend logic to switch themes and store user preferences.

Criticality:

Real-time messaging & file sharing, user management, and UI customization are important functionalities for the application, but they differ in criticality. Real-time messaging & file sharing are top priorities as they directly impact the core functionality and user engagement of the application. User management follows closely behind as it ensures the security, integrity, and personalization of user interactions. UI customization, while enhancing user experience, is considered less critical but still significant for improving user satisfaction and retention. Overall, prioritizing these functionalities appropriately is essential for delivering a successful and competitive application.

Phase 4: Enhancements and Integrations (Mar 19 - Apr 8, 2024)

Objectives & Requirements:

Prioritization of Functionalities (1 to 3, 1 being top priority and rest follows order):

1.Video and Audio Calls: Leverage WebRTC for implementing direct peer-to-peer communication channels, offering users enhanced interaction options beyond text.

2.Server & Channel Management: Tools for users, especially admins, to create and manage communication spaces, including permissions and settings for channels within servers.

3.Collaboration Tools: Incorporate shared calendars and task management features to support team coordination and project management within the chat application.

Criticality:

Video and audio calls are identified as the top priority due to their critical role in enhancing communication capabilities and user engagement. Server and channel management tools follow closely behind, ensuring efficient organization and control over communication spaces within the application. Collaboration tools, while valuable for enhancing teamwork and project management, are considered less critical but still significant for expanding the application's utility. Prioritizing these functionalities appropriately is essential for delivering a competitive and comprehensive communication platform.

Phase 5: Testing, Final Review, and Handover (Apr 9 - Apr 29, 2024)

Objectives & Requirements:

Prioritization of Functionalities (1 to 3, 1 being top priority and rest follows order):

1.Comprehensive Testing: Execute a series of unit, integration, and end-to-end tests to ensure each component functions as expected and the system as a whole operates seamlessly.

2.Security Audits: Conduct security assessments to identify vulnerabilities, applying fixes and enhancements to ensure robust protection of user data and interactions.

3.User Feedback Collection: Engage real users in testing to gather insights and feedback for refining the application, enhancing usability, and ensuring the product meets market needs.

Criticality:

Comprehensive testing is identified as the top priority due to its critical role in ensuring the reliability, stability, and functionality of the application. Security audits follow closely behind, as they are essential for protecting user data and interactions from potential threats and vulnerabilities. User feedback collection, while valuable for enhancing usability and meeting market needs, is considered less critical but still significant for iterative improvements and user satisfaction. Prioritizing these functionalities appropriately is essential for delivering a secure, reliable, and user-centric application.

Each phase builds upon the previous, methodically addressing key aspects of development to ensure the project's success. The phased approach facilitates focused development, thorough validation, and iterative refinement, culminating in a robust, user-centric team chat application.

6. Member Contribution

The team collaborated, made use of their technological expertise, and remained unwavering in their commitment to achieving their goal of offering a seamless and state-of-the-art online chat application experience. With a shared goal, our large team of dedicated professionals tackled the challenging domains of system design and technology selection during Phase 2 of our project. From September to October, this time formed the firm foundation for our online chat platform. As part of this endeavor, each team member thoughtfully added their unique skills and viewpoints to different project components. The table below gives a detailed breakdown of these individual contributions, emphasizing the specific responsibilities and tasks that each person completed.

Team Member	Contributions	Overall Contribution	Note
Srichandan Kota	Managed project planning and timeline. Led the drafting of the System Overview and Development Phase Plan sections of the report. **Project Deliverable Section:** System Overview	12.5%	shown leadership in a variety of project-related areas, guaranteeing efficient project management and completion.
Swapna Sonti	Contributed to front-end development and wrote the User Operations, Messaging in Text Channels, and File Handling in Channels sections. **Project Deliverable Section:**Functional Requirements	12.5%	committed to enhancing front-end speed and user experience by carefully integrating React capabilities.
Sandeep Chowdary Ari	Implemented real-time backend, authored Server Management and Channel Management sections focusing on admin-specific functionalities. **Project Deliverable Section:** Non-Functional Requirements	12.5%	guaranteed flawless real-time communication functionality, which is essential for the main functions of the program.
Venkata Sai Shankar Koppula	Led UI/UX design and contributed to the Usability and Accessibility sections of the report, reflecting on design choices and user experience. **Project Deliverable Section:** Interfaces	12.5%	significantly improved the application's responsiveness of the user interface and visual attractiveness.
Shivanandha Reddy Vasudevula	Managed database architecture, drafted the Data Management and Integrity section, detailing database design and data relationship management.	12.5%	enhanced the overall efficiency of the program by providing crucial database

	Project Deliverable Section:Development Phase Plan		optimization knowledge.
Gana Deekshith	Led secure authentication flow integration, wrote the User Authentication and Authorization section, explaining security measures and user verification processes. **Project Deliverable Section**: Member Contribution	12.5%	enhanced application security by carefully integrating authentication methods.
Kantumutchu Dinesh	Enhanced UI components, contributed to the Non-Functional Requirements section, focusing on Performance Efficiency, Scalability, and Maintainability. **Project Deliverable Section**: Conclusion	12.5%	greatly aided in accessibility and UI flexibility, accommodating a wide range of user preferences.
Bhanu Prasad Krishna Murthy	Spearheaded documentation efforts, compiled the Table of Contents, and was responsible for the Conclusion, ensuring coherence and completeness of the report. **Project Deliverable Section** : Use Case Diagram	12.5%	increased productivity of the team by implementing best practices for database query optimization and automating documentation procedures.

7. Conclusion

We have effectively established the foundation for our sophisticated team chat application as Phase 2 of our project development draws to an end, making significant strides in system design and technology choices. In each of their fields, our committed team has not only met but surpassed expectations; together, they have created a solid and scalable infrastructure that will undoubtedly transform teamwork. We have tracked progress with efficiency and transparency during this phase by using a Trello Kanban board for task management. Frequent Google Meet meetings have promoted teamwork and expedited decision-making; concurrent Git contributions have maintained our codebase current and structured. Our team's experience working in tandem with these technologies has helped us advance and laid a strong basis for the next stages of growth.