iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: V Month of publication: May 2025

DOI: https://doi.org/10.22214/ijraset.2025.71869

Review Paper on: VartalApp

Tarun Soni[1], Shweta Singh[2], Jitendra Mishra[3], Ali Raza[4] , Sourabh Pandey[5]
[1, 2, 3, 4]Student, [5]Asst. Professor, Goel Institute of Technology and Management, Lucknow

Abstract: In today's fast-paced digital environment, real-time communication tools have become essential for effective collaboration, especially among distributed teams. This research project presents VartalApp, a web-based real-time messaging application designed to enable seamless team communication through virtual chat rooms. The system allows users to create and join rooms using a unique room ID and username, promoting focused and organized discussions. Built with Spring Boot, WebSocket, and MongoDB on the backend and React on the frontend, VartalApp ensures smooth and responsive interaction with a modern, user-friendly interface. The application leverages WebSocket technology to enable bi-directional, low-latency communication between clients and servers, ensuring real-time message delivery. This project explores the system's design, development process, architectural choices, and implementation details, with a focus on scalability, usability, and data integrity. Additionally, the study addresses security measures such as session management and message confidentiality to ensure a safe communication environment. The outcomes of this project demonstrate the feasibility and effectiveness of using open-source tools to develop a lightweight, real-time messaging platform suitable for academic, organizational, or personal use.

I. INTRODUCTION

Effective real-time communication is crucial in today's digital world, especially for teams working remotely. Many existing messaging platforms are either too complex or lack customization for focused group interactions. To address this, VartalApp was developed as a simple, web-based messaging application that allows users to create or join chat rooms using a username and unique room ID.

Built using Spring Boot, WebSocket, and MongoDB on the backend, and React on the frontend, VartalApp ensures smooth, real-time communication with a modern user interface. The project focuses on creating a scalable, responsive, and secure platform tailored for team collaboration. This report outlines the development process, system design, and key features of the application, highlighting the practical challenges and solutions encountered.

II. PROBLEM STATEMENT

In the current digital age, effective real-time communication is essential for smooth collaboration among team members, especially in remote or hybrid environments. While many existing messaging platforms offer robust features, they often come with unnecessary complexities, lack customization, or are not optimized for focused group interactions. Users may find it difficult to quickly create private chat spaces or engage in lightweight, distraction-free communication.

There is a need for a simple, secure, and efficient real-time messaging solution that allows users to easily create or join dedicated chat rooms without complex setup or registration processes. The challenge lies in building such a system with real-time capabilities, intuitive user experience, and data integrity using modern, scalable technologies.

III. OBJECTIVES

The main objective of the VartalApp project is to develop a web-based real-time messaging application that enables seamless communication between users through dedicated chat rooms. The specific goals include:

1) To design a simple and intuitive user interface for easy navigation and interaction.
2) To implement real-time, bi-directional messaging using WebSocket technology.
3) To allow users to create or join chat rooms using a username and unique room ID without the need for complex registration.
4) To ensure scalability and performance using Spring Boot and MongoDB.
5) To maintain data consistency, security, and privacy during communication.
6) To provide a responsive frontend using React for a smooth user experience across devices.

IV. LITERATURE REVIEW

In recent years, the demand for real-time communication platforms has increased significantly due to the rise of remote work, online learning, and virtual collaboration. Popular messaging applications like Slack, Microsoft Teams, and Discord have set benchmarks in providing rich features such as group chats, file sharing, and notifications. However, these platforms often include complex functionality and are not always suitable for lightweight or custom communication needs.

Several research studies and open-source projects have explored the implementation of real-time communication using WebSocket, a protocol that enables full-duplex communication between client and server. Compared to traditional HTTP polling, WebSocket provides faster and more efficient message exchange, which is essential for real-time chat systems.

In the backend, frameworks like Spring Boot are commonly used due to their simplicity, modularity, and support for WebSocket integration. MongoDB is also widely adopted in similar applications for storing flexible, unstructured chat data in real time.

On the frontend, React has become a preferred choice for building dynamic and responsive user interfaces. Its component-based architecture and state management tools like hooks and context API are highly effective for real-time applications.

While existing literature and platforms demonstrate powerful features, there is limited focus on lightweight, customizable chat apps built specifically for small teams or academic projects. VartalApp aims to bridge this gap by integrating proven technologies into a simple yet effective real-time messaging solution

V. SYSTEM ARCHITECTURE

The architecture of VartalApp follows a client-server model designed to support real-time, bi-directional communication between users. The system is divided into three main layers: Frontend, Backend, and Database.

1) *Frontend (Client Side) :*

- Built using React.js for a dynamic and responsive user interface.
- Handles user input (name and room ID), displays real-time messages, and maintains WebSocket connections.
- Communicates with the backend through WebSocket and REST APIs when necessary.

2) *Backend (Server Side)*

- Developed using Spring Boot, which manages application logic, WebSocket sessions, and message routing.
- WebSocket protocol is used for persistent, real-time communication.
- Validates room IDs and user entries, manages connected sessions, and broadcasts messages to appropriate rooms.

3) *Database Layer*

- Uses MongoDB, a NoSQL database, to store:
- Chat messages (with timestamps, sender, and room info)
- Room metadata (e.g., room ID, participants)
- Chosen for its flexibility and ease of handling unstructured data like chat logs.

4) *Communication Flow:*

- When a user enters their name and room ID, the frontend establishes a WebSocket connection with the backend.
- Messages sent by a user are routed to the backend via WebSocket.
- The backend broadcasts the message to all users connected in the same room.
- Messages are also stored in the database for future reference or logging.

VI. TECHNOLOGIES AND TOOLS USED

Category	Tool/Technology	Purpose
Frontend	React Native / Flutter	Cross-platform mobile UI
Backend	Node.js / Django	REST API and server-side
Database	Firebase / MongoDB	Storingmessages
Real-time Messaging	WebSocket / Socket.io	Instant messagetransmission

Authentication	Firebase Auth / OAuth	User login and sessionmanagement
Deployment	Docker, Kubernetes	Scalable deployment
Hosting	AWS / Google Cloud	Hosting development
Testing	Jest / Postman	Unit and API testing

VII.KEY FEATURES

VartalApp is designed to offer a simple, efficient, and real-time communication experience. The following are the key features of the application:

1) Real-Time Messaging

- Instant bi-directional message exchange using WebSocket technology.
- Messages are delivered and displayed without page refresh or delay.

2) Room-Based Communication

- Users can create or join chat rooms using a unique Room ID.
- Each room provides a separate space for focused discussions.

3) User-Friendly Interface

- Clean and responsive UI built with React.js.
- Easy navigation and smooth interaction across devices.

4) No Registration Required

- Users simply enter a name and Room ID to join a chat, eliminating complex signup processes.

5) Backend Efficiency

- Built with Spring Boot for fast and scalable server-side performance.
- Efficient handling of WebSocket connections and message routing.

6) Data Persistence

- Chat messages and room info are stored in MongoDB.
- Enables message logging or future data retrieval if needed.

7) Scalability and Modularity

- The system architecture supports easy addition of new features like file sharing, authentication, or notifications.

8) Cross-Platform Compatibility

- Works smoothly on desktop and mobile browsers without installing any application

VIII. TESTING AND EVALUATION

1) Unit Testing: Frontend and backend modules. Integration testing: client–server interaction.
2) Load Testing: App performance under high usage.
3) SecurityTesting:Vulnerabilityscanningand penetrationtesting.

IX. LIMITATIONS

1) Highservercostsforlarge-scaleimplementation.
2) Devicebatteryconsumptionforcontinuousbackgroundsynchronization Dependency on Internet connectivity.

X. CONCLUSION

VartalApp successfully demonstrates the development of a lightweight, real-time messaging application tailored for simple and effective team communication. By leveraging technologies like Spring Boot, WebSocket, MongoDB, and React, the application provides seamless bi-directional messaging with minimal setup, allowing users to create and join chat rooms easily.

The project highlights the importance of real-time communication in modern digital collaboration and shows that it is possible to build scalable and responsive chat systems using open-source tools. Although the current version focuses on core messaging features, future enhancements such as user authentication, message encryption, and multimedia sharing can further improve the platform’s functionality and security.

Overall, VartalApp meets its objectives by delivering an intuitive, responsive, and secure messaging environment suitable for academic, organizational, or casual use.

REFERENCES

[1] Fette, I., & Melnikov, A. (2011). The WebSocket Protocol. IETF RFC 6455. https://datatracker.ietf.org/doc/html/rfc6455
[2] Pivotal Software, Inc. (2020). Spring Boot Reference Guide.https://docs.spring.io/spring-boot/docs/current/reference/html/
[3] MongoDB Inc. (2023). MongoDB Manual. https://docs.mongodb.com/manual/
[4] React – A JavaScript library for building user interfaces. (2023). https://reactjs.org/
[5] L. Richardson, & S. Ruby. (2007). RESTful Web Services. O'Reilly Media.
[6] D. J. Cook & S. K. Das. (2005). Smart environments: Technology, protocols and applications. John Wiley & Sons.