ISSN: 2456-9348
Impact Factor: 8.232

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

REAL-TIME CHAT MESSENGER WITH STATE PERSISTENCE

ELANGOVAN.E
III BCA (GENERAL)
Department of Computer Application (UG)
VISTAS-Pllavaram
Chennai,India
elangoesakki20@gmail.com

Dr.JEBATHANGAM
Professor
Departmentof Computer Application
VISTAS-Pallavaram
Chennai, India
jthangam.scs@vistas.ac.in

ABSTRACT

The project "Real-Time Chat Messenger with State Persistence" presents a mobile-based communication system designed to enable instant, reliable, and continuous messaging across devices. The application is developed using Flutter for cross-platform user interface design and Firebase as the backend platform, ensuring seamless real-time message delivery and persistent data storage.

The system Integrates Firebase Authentication for secure user access and Cloud Firestore for structured message storage, enabling efficient data management. A key feature of the proposed system is state persistence, which maintains user sessions and preserves chat history even after application closure or device restart. This overcomes common limitations of traditional messaging systems such as data loss, delayed communication, and lack of scalability.

The application Is modular in design, incorporating authentication, user management, real-time chat handling, and database integration. It supports features such as real-time synchronization, message status tracking, and user presence monitoring. Experimental results indicate improved reliability, security, and performance.

This system demonstrates the effective use of modern mobile technologies to develop scalable and user-friendly communication platforms, with potential future enhancements including multimedia sharing, encryption, group messaging, and AI-based features.

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

Keywords

Real-Time Chat, State Persistence, Flutter, Firebase, WebSocket Communication, Cloud Firestore, Mobile Application

I.INTRODUCTION

In recent years, communication technologies have undergone rapid transformation, with real-time messaging applications becoming an essential part of daily life. From personal conversations to professional collaboration, instant messaging systems play a crucial role in enabling seamless interaction between users across the globe. The increasing demand for fast, reliable, and efficient communication has led to the development of advanced chat applications that support real-time data exchange and multi-device synchronization.

Traditional messaging systems primarily rely on request-response communication models, which often result in delays and inefficient data handling. These systems may also suffer from issues such as message loss, lack of synchronization, and inability to maintain session continuity. As the number of users and the volume of data increase, these limitations become more significant, affecting the overall user experience. Therefore, there is a need for a robust communication system that ensures real-time interaction while maintaining data integrity and persistence.

The concept of real-time communication has been significantly enhanced with the introduction of modern technologies such as WebSockets and cloud-based backend services. WebSockets enable continuous, bidirectional communication between the client and server, allowing messages to be transmitted instantly without repeated requests. This technology reduces latency and improves the efficiency of messaging systems. In addition, cloud platforms such as Firebase provide scalable infrastructure for data storage, authentication, and real-time synchronization, making them ideal for developing modern chat applications.

One of the critical challenges in messaging systems is maintaining state persistence. State persistence refers to the ability of an application to retain user data, session information, and chat history even after the application is closed or the device is restarted. Many traditional chat applications fail to provide reliable persistence, leading to loss of important data and disruption in communication. This issue becomes particularly important in scenarios where continuous communication and data tracking are required, such as customer support systems, online education platforms, and enterprise collaboration tools.

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

The project titled "Real-Time Chat Messenger with State Persistence" aims to address these challenges by developing a mobile-based messaging application that ensures both real-time communication and persistent data storage. The application is built using Flutter, a cross-platform framework that allows developers to create high-performance mobile applications with a single codebase. Firebase is used as the backend service, providing features such as authentication, cloud storage, and real-time database capabilities.

The proposed system Integrates Firebase Authentication to ensure secure user login and identity management. Cloud Firestore is used for storing messages and user data in a structured and scalable manner. The system is designed to handle multiple users simultaneously while maintaining low latency and high reliability. A key feature of this application is the implementation of state persistence, which ensures that chat history and user sessions are preserved across different devices and application states.

In addition to basic messaging functionality, the system includes features such as real-time message synchronization, message status tracking (sent, delivered, and read), and user presence detection. These features enhance the overall user experience and provide a more interactive communication platform. The modular architecture of the system allows for easy integration of additional features such as multimedia sharing, group chats, and end-to-end encryption.

The objective of this research Is to design, develop, and evaluate a real-time chat messaging system that overcomes the limitations of traditional communication platforms. The study focuses on improving system performance, ensuring data reliability, and enhancing user experience through the use of modern technologies. Furthermore, the research aims to demonstrate how cloud-based solutions can be effectively utilized to build scalable and efficient mobile applications.

This paper is structured as follows: Section II presents a review of related work in the field of real-time messaging and state persistence. Section III describes the methodology and system design. Section IV discusses the results and performance evaluation. Finally, Section V concludes the paper and outlines future research directions.

II. LITERATURE REVIEW

The rapid growth of internet technologies and mobile computing has significantly influenced the development of real-time communication systems. Messaging applications have evolved from simple text-based systems to complex platforms supporting multimedia sharing, real-time synchronization, and cloud integration. Researchers have extensively studied various aspects of real-time chat systems, including communication protocols, data storage techniques, scalability, and user experience.

One of the foundational technologies in real-time communication is the use of WebSocket protocols. Unlike traditional HTTP-based communication, which follows a request-response model, WebSockets enable persistent, bidirectional communication between the client and server. Studies have shown that

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

WebSockets reduce latency and improve the efficiency of message delivery, making them suitable for chat applications and live data streaming systems. Several research works highlight that WebSocket-based systems outperform polling-based approaches in terms of speed and resource utilization.

Another important area of research is the use of cloud-based backend services for messaging applications. Platforms such as Firebase have gained popularity due to their scalability, real-time database capabilities, and ease of integration with mobile applications. Researchers have demonstrated that cloud-based systems provide better data availability and fault tolerance compared to traditional server-based architectures. Cloud Firestore, in particular, has been widely used for storing structured data and enabling real-time updates across multiple clients.

State persistence is a critical factor in modern messaging systems, as it ensures that user data and chat history are maintained across sessions. Several studies have explored different approaches to achieving state persistence, including local storage, server-side databases, and hybrid models. Local storage techniques, such as SQLite databases on mobile devices, allow applications to store data offline; however, they may face synchronization issues when multiple devices are involved. On the other hand, cloud-based storage solutions provide centralized data management, ensuring consistency and accessibility from different devices.

Machine learning and data analytics have also been introduced in messaging systems to enhance functionality. Some research focuses on integrating intelligent features such as chatbots, spam detection, and sentiment analysis. These features improve user interaction and provide additional value in applications such as customer support and social networking. However, the integration of such technologies requires efficient data handling and processing capabilities.

Security and privacy are major concerns in messaging applications. Researchers have proposed various encryption techniques to protect user data from unauthorized access. End-to-end encryption ensures that messages are encrypted on the sender's device and decrypted only on the receiver's device, preventing interception during transmission. Authentication mechanisms, such as token-based authentication and multi-factor authentication, are also widely used to enhance system security.

Scalability is another critical aspect of real-time messaging systems. As the number of users increases, the system must handle large volumes of data and concurrent connections without performance degradation. Distributed system architectures and load balancing techniques have been proposed to address this challenge. Studies indicate that microservices-based architectures improve scalability and maintainability by dividing the system into smaller, independent components.

User experience plays a significant role in the success of messaging applications. Research shows that features such as real-time notifications, message status indicators, and user presence tracking significantly enhance usability. Mobile application frameworks like Flutter have been widely adopted for developing cross-platform applications with consistent performance and user interface design.

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

Despite the advancements in real-time communication technologies, several challenges remain. Many existing systems focus on either real-time communication or data persistence, but fail to effectively integrate both. This results in issues such as data inconsistency, synchronization delays, and poor user experience. Additionally, the dependency on stable internet connectivity limits the performance of real-time systems in regions with poor network infrastructure.

Another limitation observed in existing research is the lack of modular system design. Many applications are built as monolithic systems, making it difficult to scale or add new features. A modular architecture allows developers to integrate additional functionalities such as multimedia sharing, group messaging, and advanced security features without affecting the core system.

The proposed research addresses these gaps by developing a real-time chat messenger that combines efficient communication protocols with robust state persistence mechanisms. By leveraging modern technologies such as Flutter and Firebase, the system ensures real-time synchronization, secure data storage, and seamless user experience. The integration of state persistence with real-time messaging provides a comprehensive solution to the limitations of traditional chat applications.

In summary, the literature indicates that while significant progress has been made in the field of real-time messaging systems, there is still a need for integrated solutions that address communication efficiency, data persistence, scalability, and security simultaneously. This research aims to contribute to this area by proposing a system that combines these features into a unified and efficient platform.

III. METHODOLOGY / PROPOSED METHOD

The proposed system, "Real-Time Chat Messenger with State Persistence," is designed to provide a reliable and efficient communication platform by integrating real-time messaging capabilities with persistent data storage. The methodology focuses on developing a scalable and modular architecture that ensures seamless communication between users while maintaining data consistency across different sessions and devices. The system follows a client-server model, where the client-side application is developed using Flutter, and backend services are handled through Firebase. This combination allows for cross-platform compatibility, high performance, and real-time data synchronization.

The process begins with user registration and authentication, which is handled using Firebase Authentication. This module ensures secure access to the application by allowing users to register and log in using email and password credentials. Each user is assigned a unique identifier, which is used to manage user data and communication sessions. User details such as name, email, and online status are stored in the Cloud Firestore database. The system continuously updates user presence information, enabling real-time tracking of whether a user is active or offline.

The core functionality of the syst"m Is real-time messaging, which is achieved using Firebase's real-time synchronization capabilities. When a user sends a message, it is immediately stored in the Cloud

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

Firestore database and simultaneously delivered to the recipient without delay. This eliminates the need for repeated server requests and ensures low latency communication. Each message is associated with metadata, including sender ID, receiver ID, timestamp, and delivery status such as sent, delivered, or read. This information enhances the overall user experience by providing transparency in communication.

A key feature of the proposed system Is state persistence, which ensures that chat data and user sessions are maintained even after the application is closed or the device is restarted. This is achieved through cloud-based storage, where all messages are stored in a structured format in Firestore. In addition, local caching mechanisms are implemented to improve performance and enable limited offline access. When the application is reopened, it retrieves previously stored data and synchronizes it with the current state, ensuring continuity in communication without data loss.

The database design plays a crucial role In the efficiency of the system. Cloud Firestore is used as the primary database due to its scalability and real-time capabilities. Data is organized into collections and documents, where user data and chat messages are stored separately for better management. Messages are stored in a hierarchical structure, allowing quick retrieval of conversation history. Indexing techniques are used to optimize database queries, ensuring that the system performs efficiently even with a large number of users and messages. Additionally, Firebase Cloud Storage can be used to store multimedia files such as images and documents, enhancing the functionality of the application.

Security is an essential aspect of the system, and several measures are implemented to protect user data. Firebase Authentication ensures that only authorized users can access the application. Firestore security rules are configured to restrict unauthorized read and write operations. Furthermore, basic encryption techniques are used to protect data during transmission. These measures ensure that user information and communication remain secure and private.

To improve system performance, the application uses real-time listeners that automatically update the user interface when new messages are received. This reduces the need for manual refresh and enhances responsiveness. Firebase's scalable infrastructure allows the system to handle multiple users simultaneously without performance degradation. The system is designed to efficiently manage network usage and minimize latency, providing a smooth user experience.

Overall, the workflow of the system involves user authentication, selection of contacts, message exchange, and continuous synchronization of chat data. Messages are stored and retrieved efficiently, ensuring that users can access their conversation history at any time. The integration of real-time communication with state persistence provides a robust solution to the limitations of traditional messaging systems. The use of modern technologies such as Flutter and Firebase ensures that the application is scalable, efficient, and user-friendly, making it suitable for various real-world applications.

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

IV. RESULTS AND DISCUSSION

The implementation of the "Real-Time Chat Messenger with State Persistence" demonstrates significant improvements in communication efficiency, reliability, and user experience when compared to traditional messaging systems. The system was tested under various conditions to evaluate its performance in terms of real-time message delivery, data persistence, scalability, and security. The results indicate that the integration of Flutter and Firebase technologies provides a robust platform capable of handling modern communication requirements.

One of the key outcomes observed during testing is the efficiency of real-time message delivery. Messages are transmitted instantly between users with minimal latency due to the use of Firebase's real-time data synchronization. Unlike traditional HTTP-based systems that rely on repeated requests, the implemented system uses continuous data streaming, ensuring that messages are delivered without delay. This significantly enhances user experience, especially in scenarios requiring immediate communication. The system was tested with multiple users communicating simultaneously, and it maintained consistent performance without noticeable delays.

Another important result is the effectiveness of state persistence in maintaining chat continuity. All messages and user session data are stored in Cloud Firestore, ensuring that no data is lost even when the application is closed or the device is restarted. When users reopen the application, the chat history is automatically retrieved and synchronized with the current state. This feature addresses one of the major limitations of traditional messaging systems, where data loss or incomplete synchronization can occur. The implementation of local caching further improves performance by allowing quick access to recently viewed messages, even in low network conditions.

The system also demonstrates strong performance In handling multiple users and large volumes of data. Firebase's scalable infrastructure enables the application to manage concurrent connections efficiently without affecting performance. During testing, the system was able to support multiple chat sessions simultaneously, maintaining stable performance and consistent message delivery. This highlights the suitability of the proposed system for large-scale applications such as enterprise communication platforms and social networking services.

Message tracking and user presence features were also evaluated as part of the system's functionality. The application successfully tracks message status, including sent, delivered, and read indicators, providing users with real-time feedback on their communication. Additionally, the user presence feature accurately displays whether a user is online or offline, enhancing interactivity. These features contribute to improved usability and make the application more engaging for users.

Security and data protection were analyzed to ensure the safety of user information. Firebase Authentication effectively restricts access to authorized users, while Firestore security rules prevent unauthorized data operations. The system ensures that only authenticated users can access and modify their data, reducing the risk of data breaches. Although basic encryption techniques are implemented,

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

the system can be further enhanced by integrating end-to-end encryption for improved security in future versions.

Despite the positive results, some limitations were identified during the evaluation process. The system's performance is dependent on internet connectivity, and in areas with poor network conditions, there may be slight delays in message synchronization. Additionally, the reliance on cloud services may introduce latency if the server is geographically distant from the user. However, these issues can be mitigated by optimizing network usage and using content delivery networks.

Another limitation is the initial setup complexity associated with integrating multiple Firebase services. Developers need to configure authentication, database rules, and cloud storage, which may require technical expertise. However, once configured, the system provides a highly efficient and scalable solution for real-time communication.

Overall, the results demonstrate that the proposed system effectively combines real-time messaging with state persistence to deliver a reliable and user-friendly communication platform. The use of modern technologies ensures efficient performance, scalability, and secure data handling. The discussion highlights that the system not only meets the basic requirements of a messaging application but also provides advanced features that enhance user experience and system reliability.

The findings of this study suggest that Integrating real-time communication with persistent storage is essential for developing modern messaging applications. The proposed system provides a strong foundation for future enhancements, including multimedia support, group messaging, artificial intelligence-based features, and advanced security mechanisms. These improvements can further extend the applicability of the system in various domains such as education, healthcare, customer support, and enterprise communication.

V. CONCLUSION AND FUTURE WORK

The "Real-Time Chat Messenger with State Persistence" project presents an effective solution for modern communication needs by combining real-time messaging capabilities with reliable data storage mechanisms. The system successfully addresses the limitations of traditional chat applications, such as delayed message delivery, data inconsistency, and lack of session continuity. By utilizing advanced technologies like Flutter for cross-platform development and Firebase for backend services, the proposed system ensures efficient performance, scalability, and seamless user experience.

The Implementation of real-time synchronization enables instant message exchange between users with minimal latency, significantly improving communication efficiency. At the same time, the integration of state persistence ensures that chat history and user session data are securely stored and can be retrieved at any time, even after application closure or device restart. This feature plays a crucial role in

ISSN: 2456-9348
Impact Factor: 8.232

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

maintaining data integrity and enhancing reliability, making the system suitable for both personal and professional use cases.

Furthermore, the system incorporates essential functionalities such as secure user authentication, message status tracking, and user presence detection, which contribute to an interactive and user-friendly environment. The modular architecture of the application allows for easy scalability and future expansion without affecting the existing system structure. Performance evaluation indicates that the system can handle multiple users and concurrent messaging efficiently, demonstrating its suitability for real-world applications.

Despite its advantages, the system has certain limitations, including dependency on internet connectivity and reliance on cloud-based infrastructure. In areas with poor network availability, real-time synchronization may be affected. Additionally, while basic security measures are implemented, there is scope for enhancing data protection through advanced encryption techniques.

Future work can focus on extending the functionality of the system by incorporating features such as multimedia messaging, group communication, and end-to-end encryption for improved security. The integration of artificial intelligence-based features, such as chatbots and smart reply systems, can further enhance user interaction. Expanding the application to support web platforms and enterprise-level deployment will increase its usability across various domains. Overall, the proposed system provides a strong foundation for developing advanced, secure, and scalable real-time communication applications in the future.

REFERENCES

[1] J. F. Kurose and K. W. Ross, Computer Networking: A Top-Down Approach, 7th ed. Pearson, 2017.

[2] I. Sommerville, Software Engineering, 10th ed. Pearson, 2015.

[3] M. Kleppmann, Designing Data-Intensive Applications. O'Reilly Media, 2017.

[4] P. Saint-Andre, XMPP: The Definitive Guide. O'Reilly Media, 2009.

[5] Google, "Flutter Documentation," [Online]. Available: https://flutter.dev/docs□

[6] Google, "Firebase Documentation," [Online]. Available: https://firebase.google.com/docs□

[7] R. Fielding, "Architectural Styles and the Design of Network-Based Software Architectures," Doctoral dissertation, University of California, Irvine, 2000.

[8] E. Rescorla, "The Transport Layer Security (TLS) Protocol," IETF RFC 5246, 2008.

[9] H. Garcia-Molina, J. D. Ullman, and J. Widom, Database Systems: The Complete Book, 2nd ed. Pearson, 2008.

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

[10] A. Banks and E. Porcello, Learning React: Functional Web Development with React and Redux. O'Reilly Media, 2017.

[11] S. Tilkov and S. Vinoski, "Node.js: Using JavaScript to Build High-Performance Network Programs," IEEE Internet Computing, vol. 14, no. 6, pp. 80–83, 2010.

[12] N. Leavitt, "Will NoSQL Databases Live Up to Their Promise?" Computer, vol. 43, no. 2, pp. 12–14, 2010.

[13] A. Lakshman and P. Malik, "Cassandra: A Decentralized Structured Storage System," ACM SIGOPS Operating Systems Review, vol. 44, no. 2, pp. 35–40, 2010.

[14] G. DeCandia et al., "Dynamo: Amazon's Highly Available Key-Value Store," ACM SIGOPS Operating Systems Review, vol. 41, no. 6, pp. 205–220, 2007.

[15] M. Fowler and J. Lewis, "Microservices: A Definition of This New Architectural Term," 2014.

[16] D. Comer, Internetworking with TCP/IP, 6th ed. Pearson, 2013.

[17] T. Berners-Lee, R. Fielding, and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax," IETF RFC 3986, 2005.

[18] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, 1994.