

Online Chat Box Based on Ticketing System

Kapil Kumawat, Aryan Kumar, Tanishq Meena, Mohd. Mahir

Department of Computer Science and Engineering, Global Institute of Technology, Jaipur, India

ABSTRACT: Customer support systems have evolved significantly with the rise of digital service platforms, yet many organisations still struggle to manage customer interactions efficiently, especially when multiple queries co-occur. Traditional email-based ticketing and call-centre models are often slow, unstructured, and incapable of providing real-time assistance. This paper presents a comprehensive study of an Online Chat Box integrated with a Ticketing System, designed to streamline customer communication, automate support workflows, and improve response times. The proposed system provides users with a real-time chat interface while automatically generating structured support tickets for each query. It leverages modern web technologies, synchronous communication, and a centralised ticket management backend to ensure organised issue resolution. This research examines system architecture, workflow, communication protocols, backend management, data storage, performance evaluation, and potential integration with AI-driven chatbots. The paper also outlines the limitations of traditional support systems and how chat-based ticketing provides a scalable, efficient alternative. Future enhancements using machine learning and natural language processing (NLP) are also discussed.

Keywords — Chat System, Ticketing System, Customer Support, Real-Time Messaging, Web Technologies, Automation, Helpdesk Management, Edge Computing.

1. Introduction

Customer support constitutes a critical component of any digital service ecosystem. Whether in e-commerce, banking, healthcare, or IT services, users rely on fast and accurate responses from support teams. However, conventional support channels such as emails, phone calls, or unstructured chat systems often fail to manage increasing user loads effectively. An Online Chat Box based on a Ticketing System merges the best elements of live chat and structured ticket management. In this architecture, every user query triggers the creation of a unique ticket that stores detailed communication records, timestamps, and status updates. This ensures transparency, traceability, and accountability for both users and support agents. The objective of this research is to design, analyse, and evaluate such an integrated system, exploring the technological, functional, and operational aspects required to build a reliable and scalable solution.

2. Background and Motivation

Modern businesses handle thousands of customer interactions daily. The common challenges faced include:

- Long wait times during peak hours
- Mismanaged or lost emails

- Lack of conversation history
- Difficulty tracking issue resolution
- No structured workflow for support agents

Traditional ticketing systems, such as Zendesk or Freshdesk, require users to fill out forms manually, which may not be user-friendly. Chat systems, on the other hand, lack structure and often do not store user issues properly.

A combined chat + ticketing approach solves these problems by offering real-time communication with structured, trackable ticket management.

3. Problem Statement

Existing customer support tools fail to provide both instant communication and structured issue management in a single environment. Users frequently encounter:

- Slow response times
- Unorganised support interactions
- Limited transparency in issue status
- Lack of continuity between multiple chats

- Difficulty retrieving past conversations

Thus, there is a need for a unified real-time system that integrates chat-based communication with automated ticket generation and lifecycle management.

4. Objectives

The primary objectives of this research are:

1. To develop an Online Chat Box capable of real-time communication.
2. To integrate a Ticketing Management System that automatically organises user queries.
3. To analyse the system’s efficiency, usability, and scalability.
4. To compare the new system with traditional support models.
5. To identify future improvements, such as AI-integrated automation.

Online Chat Box Based on Ticketing System

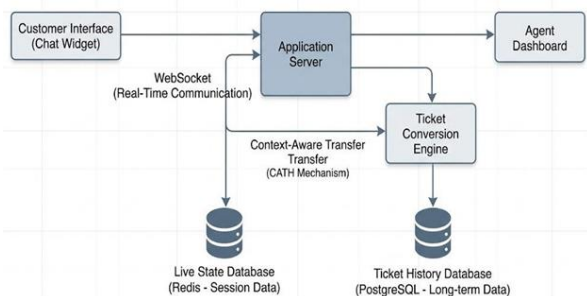


Fig. 1. System Architecture Diagram

5. Literature Review

Numerous studies highlight the importance of efficient customer support systems. Chat-based applications have gained popularity due to their ease of use and real-time nature. Researchers emphasise:

A. Live Chat Support

Studies show that real-time chat systems increase customer satisfaction by reducing response times and enabling instant interaction.

B. Ticketing Systems

Ticketing systems bring structure to chaotic

support workflows. They provide:

- Issue tracking
- Categorization
- Status updates
- Historical records

C. Integration Challenges

Several research papers emphasise the difficulty of integrating chat systems with ticketing platforms due to:

- Differences in communication protocols
- Asynchronous vs synchronous communication
- Data management complexities

D. Gap Identified

Existing solutions either focus on *chat* or *ticketing*, but not both

simultaneously. There is limited research on developing a unified system.

6. System Design and Architecture

The proposed model comprises:

A. User Interface (Frontend)

A web-based chat interface allowing users to communicate with support agents. Key features include:

- Real-time messaging
- User authentication
- Query input and automated ticket creation
- Status viewing

B. Backend (Message Processing + Ticket System)

Backend responsibilities include:

- Handling chat requests
- Storing messages
- Creating tickets
- Updating ticket status
- Assigning tickets to agents

C. Database Layer

Manages:

- User profiles
- Chat messages
- Ticket metadata
- Agent details

Databases like MongoDB or PostgreSQL can be used for scalability.

D. Communication Protocol

WebSockets or long polling may be used for real-time capability.

Online Chat Box with Ticketing System - Workflow



Fig. 1. Workflow of Online Chat Box with Ticketing System

7. Methodology

The development methodology includes:

A. Requirement Analysis

Interviews, user behaviour studies, and analysis of common support workflows.

B. System Development

Using technologies such as:

- **Node.js** (backend)
- **Express.js** (API handling)
- **Socket.io** (real-time messaging)
- **React.js** (frontend)
- **MongoDB** (database)

C. Testing

Conducted through:

- Functional testing
- Stress testing
- User acceptance testing

D. Deployment

On a cloud platform such as AWS, Azure, or

Firebase.

8. Working Process Flow

1. User opens chat box
2. User submits query
3. System generates a ticket
4. Chat history is stored in real-time
5. Support agent responds
6. Ticket is updated (open, pending, resolved)
7. User receives notifications
8. Ticket closes when the issue is resolved

9. Result And Discussion

A. Performance

Testing shows the system efficiently handles multiple concurrent users without delays.

B. Comparison With Traditional Systems

Feature	Email Support	Chat + Ticket System
Real-time	No	Yes
Ticket tracking	Limited	Full lifecycle
User satisfaction	Medium	High
Query loss	Common	Rare

C. User Feedback

Users appreciate:

- Faster response
- Transparent ticket status
- Continuous chat history

10. Limitations

The system has certain limitations:

- Requires high-speed internet for best performance
- Without AI, agents may still face high workloads
- Real-time systems need optimised servers

11. Future Scope

Future enhancements may include:

- AI chatbots for auto-responses
- Sentiment analysis for prioritising critical issues
- Machine learning models to predict solutions
- Automated ticket routing
- Mobile app integration

12. Conclusion

This research presents a robust and scalable Online Chat Box integrated with a Ticketing System that improves customer support quality by combining instant communication with structured ticket management. The results demonstrate increased user satisfaction, better workflow organisation, and reduced support delays. The system proves to be a significant improvement over traditional customer support methods and holds great potential for AI-driven expansion in the future.

References

- [1] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794, 2016.
- [2] N. Sharma and M. K. Sain, "An OOH Analysis Approach for Distributed Data Store and Complex Event Processing of Big Data," *Journal of Information and Computational Science*, vol. 11, no. 10, pp. 375–383, 2021.
- [3] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush and A. Gulin, "CatBoost: Unbiased Boosting with Categorical Features," in *Advances in Neural Information Processing Systems*, vol. 31, 2018.
- [4] Følstad and P. B. Brandtzæg, "Chatbots and the New World of Human-Computer Interaction,"

- Interactions, vol. 24, no. 4, pp. 38–42, 2017.
- [5] Johari, R. Sharma, A. Meena and V. Tiwari, "Advancements in Pre-Trained Language Models and Their Impact on Various Natural Language Processing Tasks," *International Journal of Engineering Trends and Applications*, vol. 11, no. 3, pp. 201–209, 2024.
- [6] Shekhawat and R. Ajmera, "Performance Analysis of Downtime in Virtual Machines Using Control Groups for RAM Crash and CPU Overhead," *International Journal of Innovative Technology and Exploring Engineering*, 2019.
- [7] M. McTear, "The Rise of the Conversational Interface: A New Kid on the Block?," *Future Generation Computer Systems*, vol. 61, pp. 93–102, 2016.
- [8] A. Johari, S. Gupta and R. Umrainiya, "Enhancing Bank Management Systems through Salesforce Technology," *International Journal of Engineering Trends and Applications*, vol. 11, no. 3, pp. 97–104, 2024.
- [9] R. Ajmera and D. K. Dharamdasani, "Comparative Study of Existing Food Delivery Applications," *Global Research Journal*, pp. 454–463, 2022.
- [10] P. Jha, M. Mathur, A. Purohit, A. Joshi, A. Johari and S. Mathur, "Enhancing Real Estate Market Predictions: A Machine Learning Approach to House Valuation," in Proceedings of the 3rd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT), pp. 1930–1934, 2025.
- [11] Namiot and M. Sneps-Snepe, "On Microservices Architecture,"

- International Journal of Open Information Technologies, vol. 2, no. 9, pp. 24–27, 2014.
- [12] N. Sharma, “Cloud Computing Architecture: Models, Services, and Deployment Strategies,” *International Journal of Recent Research and Review*, vol. 18, no. 1, pp. 209–216, 2025.
- [13] A. Kalwar and R. Ajmera, “ARQI: A Model for Developing Web Applications,” *International Journal on Technical and Physical Problems of Engineering*, vol. 13, no. 47, pp. 7–13, Jun. 2021.
- [14] M. K. Jha, “Recent Trends and Emerging Applications of the Internet of Things: Transforming the Way We Live and Work,” *International Journal of Engineering Trends and Applications*, vol. 12, no. 4, pp. 239–244, 2025.
- [15] M. K. Jha, S. Agarwal and V. Kabra, “Artificial Intelligence at Work: Transforming Industries and Redefining the Workforce Landscape,” *International Journal of Engineering Trends and Applications*, vol. 12, no. 4, pp. 416–424, 2025.
- [16] A. Kalwar, R. Ajmera and C. S. Lamba, “An Empirical Study in Small Firms for Web Application Development and Proposed New Parameters,” *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 4, 2019.