

Dynamic Bed Availability Tracker for Healthcare using Queuing Model

Dr. Savitha Hiremath¹, J Ravindra Ujwal², Sindhu Vijay³, Shreya Vinod Kumar⁴
Computer Science and Engineering Dayananda Sagar University Bengaluru, India

Abstract- This paper presents the design and implementation of a real-time hospital bed allocation system aimed at improving emergency care efficiency in urban healthcare networks. The system leverages queuing theory, geolocation services, and real-time data management to dynamically match patients with nearby hospitals based on bed availability and proximity. Developed using Python, Streamlit, MongoDB, Geopy, and Folium, the platform features two main interfaces: a patient-facing module for booking and a hospital admin dashboard for managing bed occupancy. A key innovation is the integration of a first-aid chatbot using Google's Gemini API, which offers AI-generated medical advice based on user symptoms, ensuring patients receive preliminary care even before reaching a hospital. The system supports concurrency, secure authentication, and live map visualization, and was evaluated for response time and booking accuracy. Results demonstrate that the platform significantly reduces treatment delays and enhances healthcare coordination during high-demand situations, aligning with smart city and digital health objectives.

I. INTRODUCTION

In modern urban healthcare systems, the increasing frequency of emergency cases, combined with constrained infrastructure and unpredictable patient inflow, has led to significant stress on hospital resources. One of the most critical challenges in emergency care delivery is the inability to track and allocate hospital beds in real time, resulting in treatment delays, overcrowding, and suboptimal utilization of medical infrastructure. The COVID-19 pandemic further exposed these inefficiencies, highlighting the urgent need for a smart, responsive, and scalable hospital bed management system.

Existing systems often rely on manual processes or isolated databases, which lack coordination between hospitals and do not incorporate dynamic data inputs such as patient location or symptom urgency. As a result, patients frequently encounter long wait times, are redirected between facilities, or receive delayed care during critical situations. Addressing this gap,

our project proposes a Dynamic Bed Availability Tracker, a web-based system that enables real-time tracking and intelligent allocation of hospital beds across multiple healthcare institutions.

The system utilizes queuing theory, geolocation services, and cloud-based data management to facilitate automated booking based on proximity and availability. Additionally, it integrates Google's Gemini API to provide AI-generated first-aid advice via a chatbot, offering preliminary support to patients before reaching a hospital. Developed using Python, Streamlit, MongoDB, Geopy, and Folium, the platform enhances coordination, reduces waiting time, and aligns with smart healthcare initiatives in urban environments.

II. LITERATURE SURVEY

[1] Dr. Biraj Chandra Paul and collaborators applied queuing theory to outpatient department optimization in AIIMS, New Delhi. By modeling arrival rates, service time distributions, and staffing patterns, the authors created a robust framework that improved patient throughput and appointment scheduling. Their analysis demonstrated that strategic queuing significantly enhances operational efficiency, even without expanding hospital infrastructure. Their findings reinforce the notion that mathematical optimization, when coupled with real-time system feedback, can address systemic inefficiencies in healthcare service delivery.

[2]

Amir Elalouf and Guy Wachte provided a comprehensive review of practical queuing approaches employed in emergency departments. Their study highlighted the effectiveness of simulation tools such as discrete event simulation (DES) and Markov-based models in managing critical patient surges. By incorporating real-time data, their models adapt dynamically to shifting patient loads and service capacities, enabling prioritization based on severity and urgency. The

study emphasized that integrating such models into operational planning allows hospitals to anticipate overcrowding, better allocate staff, and ultimately improve patient outcomes. Their work is particularly relevant for urban healthcare systems dealing with episodic crises and fluctuating resource demands.

[3]

In a technology-driven approach, Sumit Soman et al. proposed a Mobile-Augmented Smart Queue Management System for hospitals that utilizes smartphone interfaces, real-time tracking, and dynamic prioritization algorithms. The system allowed patients to remotely monitor their queue position, receive alerts, and even postpone or reschedule appointments. Experimental implementation showed a significant reduction in physical crowding, improved patient experience, and better time management for hospital staff. The use of mobile technologies in tandem with queuing logic reflects a growing trend towards hybrid systems that combine mathematical optimization with user-centric digital platforms.

[4]

Further emphasizing the adaptability of queuing models, F. O. Okoth and colleagues developed a framework to manage hospital congestion in resource-limited settings. Their model applied both single and multi-server queuing systems in a Kenyan hospital environment to simulate patient admissions and service times. The results demonstrated that strategic application of these models can lead to better distribution of hospital staff, reduced wait times, and improved response during peak periods. Their work highlights how such models can be tailored to specific institutional contexts, making them highly valuable for localized healthcare planning and emergency preparedness.

[5]

Yogendra Kumar and Dr. Renuka Sahu investigated the application of queuing theory to minimize patient waiting time in hospital systems. Their work focused on both single-server and multi-server models to simulate outpatient department (OPD) workflows. The results indicated that multi-server queuing systems substantially reduce bottlenecks and improve service rates in high-demand environments. Their findings support the notion that mathematical modeling offers a structured approach to managing uncertainty in patient inflow, leading to more efficient resource utilization, enhanced staff coordination, and improved patient satisfaction. This research underlines the relevance of queuing theory

as a foundational tool for optimizing healthcare delivery.

III. OBJECTIVE AND SCOPE

Conventional hospital bed management systems often require manual updates, phone-based coordination, or siloed databases, which create delays in emergency response and inefficiencies in hospital workflows. During high-pressure scenarios—such as pandemics, disasters, or peak admission periods—these limitations lead to critical lapses in patient care. The lack of a centralized system that can dynamically reflect real-time bed availability across multiple institutions has become a pressing concern in urban healthcare delivery.

The objective of this project is to address these gaps by developing a dynamic bed availability tracking system that integrates queuing theory with real-time location and data services. The platform enables patients to view, select, and book emergency hospital beds based on proximity, urgency, and bed availability, while hospital staff can update and manage resources in real time. By incorporating queuing logic, the system ensures fair allocation of beds and minimizes emergency wait times, especially when demand surges unpredictably.

The solution leverages open-source technologies including Python, MongoDB, Streamlit, Geopy, and Folium to build a modular, interactive, and scalable interface. A key feature is the integration of a Gemini-powered, AI chatbot, which provides users with first-aid support and basic triage before hospital arrival. The system consists of two interfaces: a patient-facing frontend for bed search and booking, and a hospital-admin dashboard for managing bed occupancy and patient discharge workflows.

Through the integration of location intelligence, real-time data handling, and chatbot-assisted triage, our project offers a scalable and cost-effective solution for emergency bed allocation. It empowers healthcare providers to evaluate patient needs remotely, improves early response during crises, and maximizes the efficiency of hospital resource usage across urban medical networks.

IV. METHODOLOGY

A. System Overview and Design Approach

The Dynamic Bed Availability Tracker was designed as a modular web-based application intended to streamline emergency hospital admissions by integrating patient-facing and administrator-facing tools. The system leverages a cloud-hosted NoSQL database, geolocation processing, and real-time web interfaces to create a responsive environment for bed search, booking, and hospital-side updates. Development followed a layered methodology, encompassing user interface creation, backend logic implementation, data management, and external API integration.

The system supports two primary stakeholders: patients and hospital administrators. Patients interact with a booking interface that recommends nearby hospitals based on availability and proximity, while administrators update real-time bed status, admit or discharge patients, and monitor incoming bookings. Additionally, a chatbot module aids patients with first-aid guidance and symptom-based triage.

B. Technology Stack and Implementation

The platform was implemented using Python as the core backend language, with Streamlit used to build dynamic, user-friendly web interfaces. MongoDB served as the central database for managing hospital data, patient records, and booking logs. For mapping and spatial analytics, the system employed Folium to render interactive maps and Geopy to compute distances using geodesic functions.

Hospital data including location coordinates, bed availability, and occupancy statistics were stored in structured MongoDB collections. On the frontend, patient input was processed through forms and sent to the backend for hospital matching. Distance calculations and bed availability checks were used to recommend optimal hospitals, which were then visualized via Folium markers and polyline paths. Unique booking IDs were generated for each reservation to ensure traceability.

C. Real-Time Bed Allocation Logic

A central component of the system's intelligence lies in the dynamic bed allocation algorithm, which is underpinned by queuing theory. Upon receiving patient location and symptoms, the backend retrieves hospital data filtered by radius and sorted based on distance and availability. If a hospital within the defined proximity has available beds, a booking is committed using atomic MongoDB transactions to prevent overbooking.

The queuing logic simulates service order based on real-time inflow and prioritizes patients by proximity, ensuring equitable resource usage. Bed counters are adjusted immediately upon confirmation, and admin panels reflect live updates to maintain transparency across facilities.

D. First-Aid Chatbot Integration

The platform incorporates a symptom-based virtual assistant, built using a hybrid approach combining Google's Gemini API and a rule-based local fallback engine. The chatbot processes user prompts via keyword recognition and, where applicable, calls the Gemini API for intelligent responses. In case of failure or timeout, the system retrieves predefined responses from a locally defined dictionary.

The chatbot assists users by interpreting symptoms, suggesting first-aid measures, and recommending when to seek emergency care. It operates without requiring login credentials and is embedded into the patient interface, making it accessible during critical early decision-making moments.

E. Data Handling, Security, and Validation

All user and hospital data are stored in structured MongoDB collections, accessed and manipulated using PyMongo. The application performs field-level validation to prevent incomplete or malformed data submissions. Admin authentication is secured through SHA-256 hashing, with optional integration of bcrypt for additional security layers.

Database operations such as bookings, discharges, and hospital status updates are encapsulated in try-except blocks to ensure graceful error handling. Real-time feedback is provided to users in cases of missing data, invalid coordinates, or unavailable beds, improving both usability and system robustness.

V. SYSTEM ARCHITECTURE AND DIAGRAMS

A. Class Structure Diagram

Figure 1 illustrates the class-level design of the system, showcasing the primary entities: Patient, Hospital, and Booking. Each class is defined with its respective attributes and functions. This diagram highlights how the system models its data internally, ensuring clear abstraction and efficient data handling during operations such as bed booking, discharge, and hospital updates.

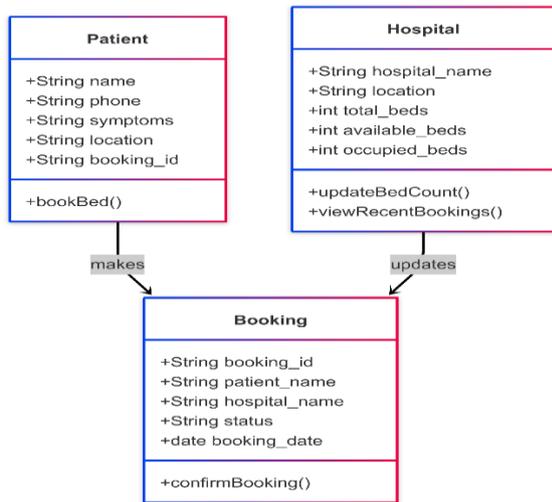


Fig.1 Class Structure Diagram – Patient, Hospital, Booking

B. Booking Interaction Flow

Figure 2 represents the interaction flow among the patient, system backend, hospital database, and the hospital interface. It outlines the steps involved from the moment a patient submits their details to when a hospital is allocated and booking confirmation is returned. This interaction ensures a seamless exchange of information while maintaining real-time synchronization of bed availability.

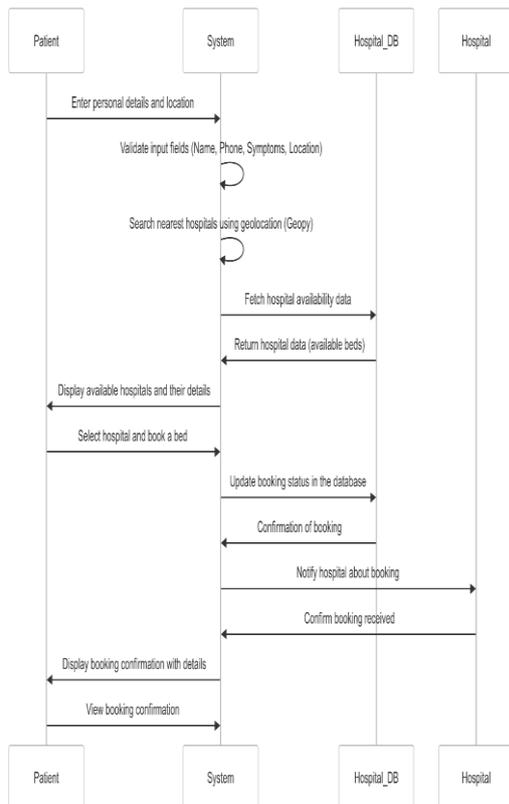


Fig.2 Booking Interaction Flow

C. Geolocation Mapping Interface

Figure 3 provides a visual snapshot of the real-time map interface rendered using Folium. It marks the patient’s current location alongside the nearest hospital with available beds. This visual aid supports user understanding and helps validate that the recommended hospital is both available and geographically optimal.

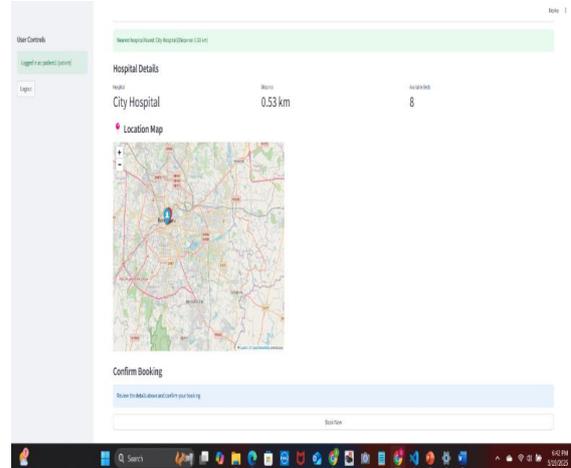


Fig. 3 Folium Map Interface Display

D. Chatbot User Interface

Figure 4 showcases the chatbot interface embedded within the patient portal. The chatbot assists users by providing first-aid suggestions and symptom-based triage using either Gemini API or rule-based logic. It is accessible without login and designed for real-time interaction. The interface is minimal, intuitive, and returns contextual medical advice instantly based on user input.

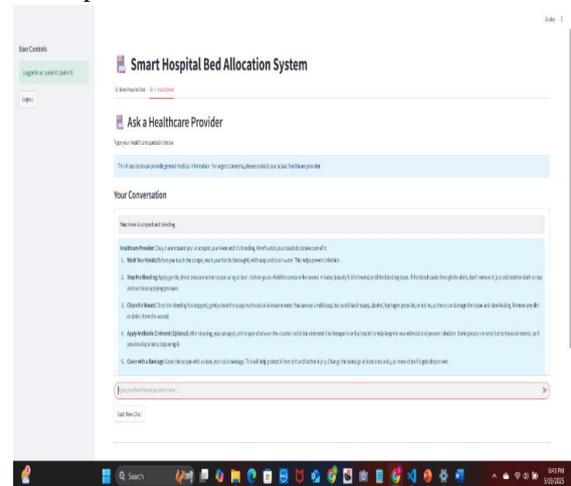


Fig. 4 Chatbot Interface Display

VI. CONCLUSION

The development of the Hospital Bed Allocation System, which integrates geolocation intelligence,

real-time data management, and AI-driven user support, offers a significant advancement in managing healthcare resources efficiently, especially during emergencies. By utilizing technologies like Streamlit for an intuitive user interface, MongoDB for scalable backend storage, Geopy and Folium for distance calculations, and Gemini API for AI-based symptom triage, the system facilitates smart bed booking and improves patient care by minimizing wait times. It supports two key user roles—patients and hospital administrators—offering functionalities such as patient registration, intelligent bed allocation, real-time bed management dashboards, and fallback mechanisms to ensure data integrity and system availability. The platform's design emphasizes scalability, modularity, and maintainability, making it adaptable to various healthcare environments, from hospitals to home-care settings. This system not only optimizes hospital operations but also enhances resource utilization, ensuring timely interventions and improving patient outcomes. By laying the foundation for seamless integration into broader health-tech ecosystems, the system demonstrates the potential to transform hospital management, driving efficiency and improving the quality of care in both routine and emergency healthcare scenarios.

REFERENCES

- [1] B. C. Paul, N. Kumar, A. Kumar, I. B. Singh, and S. K. Neogy, "Queuing-model to optimize patient waiting time in OPD of a super-speciality public hospital of India," *J. of Health Tech.*, vol. 5, no. 2, pp. 1–8, 2021.
- [2] A. Elalouf and G. Wachte, "Queuing problems in emergency departments: A review of practical approaches and research," *Oper. Res. Forum*, vol. 2, no. 3, pp. 1–20, 2021.
- [3] S. Soman, S. Rai, P. Ranjan, and A. S. Cheena, "Mobile-augmented smart queue management system for hospitals," in *Proc. IEEE 33rd Int. Symp. Comput.-Based Med. Syst. (CBMS)*, Rochester, MN, USA, 2020, pp. 289–294.
- [4] F. O. Okoth, A. W. Okoth, K. L. Nyongesa, and J. L. Sirengo, "Queuing model for hospital congestion with application," *Int. J. Math. Sci. Optim.: Theory Appl.*, vol. 9, no. 1, pp. 1–12, 2023.
- [5] Y. Kumar and R. Sahu, "Using queuing theory to minimize waiting time in the hospital system," *Int. J. Creative Res. Thoughts*, vol. 11, no. 2, pp. 1–8, 2023.
- [6] G. Lohar, "Medical Chatbot with Gemini 2.0, Flask and Vector Embedding," *Analytics Vidhya*, Jan. 2025. [Online]. Available: <https://www.analyticsvidhya.com/blog/2025/01/medical-chatbot-with-gemini-2-0-flask-and-vector-embedding/>
- [7] S. Geetha, S. Narayanamoorthy, T. Manirathinam, and D. Kang, "Smart Hospital Locator for Emergency Services," *Academia.edu*, 2023.[Online].Available:https://www.academia.edu/93536311/Hospital_Locator_and_Bed_Availability_Detector_for_Emergency_Cases
- [8] R. Rathore, "A review on study of application of queuing models in hospital sector," *Int. J. Glob. Acad. Sci. Res.*, vol. 9, no. 2, pp. 1–10, 2022.
- [9] P.-H. Liao, W. Chu, and C.-S. Ho, "An analysis of waiting time for emergency treatment and optimal allocation of nursing manpower," *Healthcare*, vol. 10, no. 1, pp. 1–15, 2022.
- [10] A. K. Elalouf and G. Wachte, "Queuing theory applications for hospital emergency services," *Oper. Res. Forum*, vol. 3, no. 1, pp. 1-12, 2023.
- [11] C. Costa, J. J. P. C. Rodrigues, et al., "Wearable Health Monitoring Systems: An Overview of Design and Real-Time Data Analysis," *Future Generation Computer Systems*, vol. 106, pp. 263–283, 2020.[Online].Available:<https://doi.org/10.1016/j.future.2020.01.029>