Real Time Queue Management System

Ahire Ayush Pravin[1], Pawar Aakash Pralhad[2], Nagare Pallavi Bansilal[3], Karad Shrirang Karad[4], Dr. R. M. Gawande[5]

[5]Guide
[1,2,3,4,5]Artificial Intelligence and Data Science
Matoshri College of Engineering and Research Centre Eklahre, Nashik

Abstract:
The increasing number of visitors in government offices often leads to long waiting times, overcrowding, and inefficient service management. Traditional queue systems lack real-time updates and proper communication, causing inconvenience for both citizens and staff. This paper presents a Real-Time Queue Management System designed to improve service efficiency using an Android-based application and a Python-Django backend. The system enables users to book queue tokens digitally, track their position, and receive real-time notifications about their turn. A Linear Regression algorithm is used to predict waiting times based on current queue data, enhancing accuracy and user convenience. Additionally, a chatbot module is integrated to assist users with common queries, and a dedicated section provides information about government schemes and eligibility criteria. The system ensures better organization, reduces physical crowding, and improves overall user experience. Experimental outcomes indicate improved efficiency, reduced waiting time, and enhanced service transparency, making the system suitable for modern government offices and public service environments.

Key Words: Queue Management System, Real-Time Prediction, Linear Regression, Android Application, Django Framework, Notification System, Chatbot, Government Services, Waiting Time Prediction, Smart Governance.

INTRODUCTION

Government offices handle a large number of citizens daily, often resulting in long queues, overcrowding, and inefficient service management. In traditional systems, people are required to stand in physical lines without knowing their waiting time or queue status. This leads to confusion, time wastage, and a poor user experience. The lack of real-time updates and proper communication further increases frustration among users and reduces overall service efficiency.

With the advancement of mobile technology and smart systems, there is a need for a digital solution that can manage queues more effectively. A real-time queue management system can help users book their position remotely, track queue progress, and receive timely notifications. Such a system not only reduces physical crowding but also improves transparency and organization in government services.

This paper proposes a Real-Time Queue Management System using an Android application integrated with a Python-Django backend. The system allows users to register, join queues, and monitor their status in real time. A Linear Regression algorithm is used to predict waiting time based on current queue data and service patterns, providing accurate estimates to users. Additionally, a chatbot module is included to assist users with common queries, and a government schemes section provides relevant information and eligibility details.

The proposed system aims to improve service efficiency, reduce waiting time, and enhance user satisfaction by introducing automation, real-time updates, and intelligent prediction techniques in queue management.

LITERATURE SURVEY

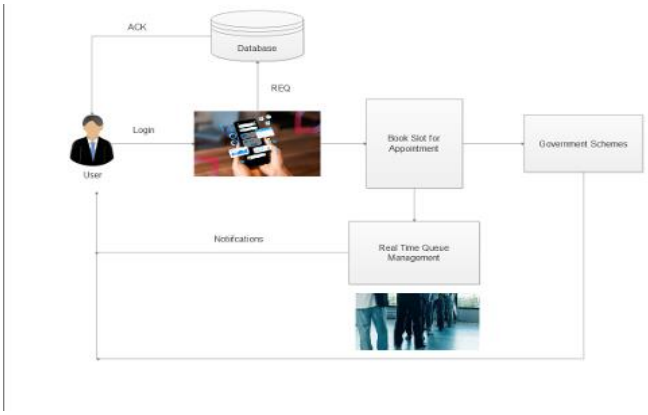
The concept of queue management systems has been widely studied to improve service efficiency and reduce waiting time in public and private sectors. Various research works have explored the use of queuing models, machine learning techniques, and smart systems to optimize queue handling and enhance user experience.

Abusair (2021) proposed a queue management approach for non-critical services using priority-based queuing and Support Vector Machine (SVM) modeling. The study focused on reducing delay and improving system performance; however, it lacked real-time user interaction and notification features.
Kallimani (2021) analyzed different queue disciplines such as FIFO and priority-based scheduling in wireless sensor networks using Markovian queuing theory. The research improved event handling and reduced waiting time, but it was mainly focused on IoT systems rather than user-centric applications.
Liang (2024) introduced an M/M/m queueing model combined with machine learning techniques like Random Forest to optimize system performance in edge computing environments. The study demonstrated improvements in throughput and delay reduction, but it required complex infrastructure and was not directly applicable to public service systems.
Bani Ahmad (2024) proposed an IoT and blockchain-based framework for smart city applications using optimal queue models. The system enhanced security and transparency; however, it increased computational complexity and lacked practical implementation for everyday queue management.
Overall, existing systems focus on theoretical models or specific domains and often lack real-time updates, user-friendly interfaces, and predictive capabilities. The proposed system addresses these limitations by integrating real-time queue tracking, waiting time prediction using Linear Regression, notification alerts, and chatbot assistance in a single user-friendly platform designed specifically for government office environments.

METHODOLOGY

The proposed Real-Time Queue Management System follows a structured approach to improve queue handling and reduce waiting time using mobile and web technologies. The system is designed with an Android application for users and a Python-Django backend for processing and management.
Initially, users register and log in through the Android application. After authentication, users can select a service and join the queue digitally without being physically present. Once a user enters the queue, the system generates a token and stores the relevant details in the backend database.
The system continuously collects real-time data such as the number of people in the queue, average service time per user, and current queue progress. This data is processed using a Linear Regression algorithm to predict the estimated waiting time for each user. The prediction is dynamically updated as the queue progresses, ensuring accuracy.
The backend system, developed using Django, manages queue operations, user data, and communication between modules. Admin or staff members can monitor the queue, call the next user, and update the system through a dashboard interface. As each user is served, the system automatically updates the queue and recalculates waiting times.
To enhance user experience, notifications are sent to users when their turn is approaching. A chatbot module is also integrated to provide instant assistance by answering common queries and guiding users. Additionally, a government schemes module is included to display information about available schemes and their eligibility criteria.
This methodology ensures efficient queue management, real-time updates, accurate waiting time prediction, and improved communication between users and service providers.

BLOCK DIAGRAM



OBJECTIVE

1. To develop a digital queue management system for efficient handling of visitors in government offices.
2. To provide real-time updates and notifications about queue status and waiting time.
3. To predict waiting time accurately using a Linear Regression algorithm.
4. To assist users through a chatbot for quick guidance and query resolution.
5. To display government schemes along with eligibility criteria for easy access to information.

PROBLEM DEFINATIONS

In government offices, queue management is mostly manual, leading to long waiting times, overcrowding, and lack of real-time information. Users are unaware of their queue status, causing confusion and inefficiency. Hence, a smart system is required to manage queues digitally, provide real-time updates, and reduce waiting time.

FUCTIONAL REQUIREMENTS

1. The system shall allow users to register and log in securely.
2. The system shall enable users to join queues and generate digital tokens.
3. The system shall display real-time queue status and estimated waiting time.
4. The system shall send notifications when the user's turn is near.
5. The system shall allow admin to manage queues and update status.

NON FUCTIONAL REQUIREMENTS

1. Performance: The system should provide fast and real-time updates.
2. Reliability: The system should work without crashes or delays.
3. Security: User data must be securely stored and transmitted.
4. Usability: The interface should be simple and user-friendly.
5. Scalability: The system should handle multiple users and queues efficiently.

IMPLEMENTAION

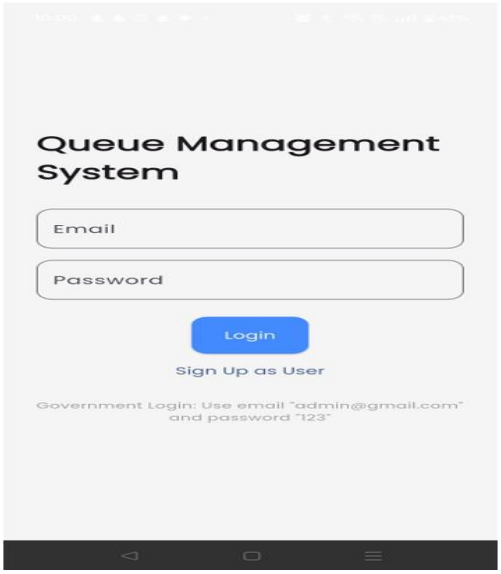


Fig: Home Page

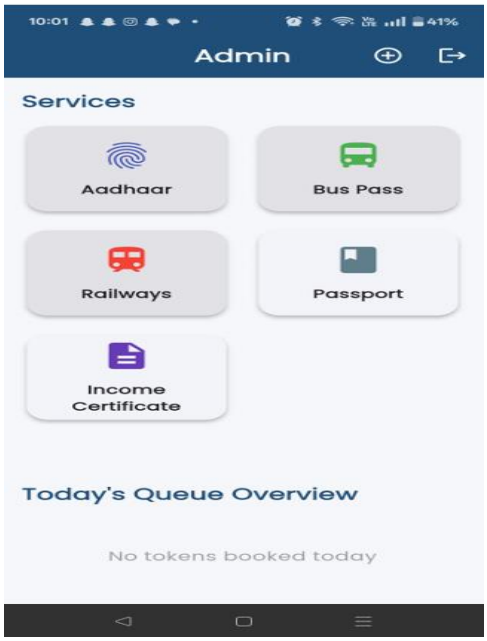


Fig: Dashboard

CONCLUSION

The proposed Real-Time Queue Management System provides an efficient and user-friendly solution to manage queues in government offices. By enabling digital token generation, real-time updates, and waiting time prediction using Linear Regression, the system reduces crowding and saves time. The integration of notifications and chatbot support further improves user experience. Overall, the system enhances service efficiency, transparency, and convenience, making it a practical solution for modern public service environments.

REFERENCES:

[1] L. L. Guo, R. Tang, J. Y. Wang, and S. Zheng, "Machine Learning-Based Prediction of Waiting Times for Medical Tasks," *IEEE Journal of Biomedical and Health Informatics*, 2025.
[2] T. K. Taton, B. Saha, and M. J. Islam, "A Comprehensive Approach to Queue Waiting Time Prediction Using Ensemble Learning," *IEEE Access*, 2025.
[3] N. Okafor, B. Lusch, and V. Vishwanath, "Queue Wait Time Prediction in High Performance Computing Systems," *IEEE Transactions on Parallel and Distributed Systems*, 2025.
[4] A. Stoltidis et al., "Active Queue Management in 5G and Beyond Cellular Networks Using Machine Learning," *IEEE Communications Surveys & Tutorials*, 2025.
[5] D. Efrosinin, "Machine Learning Applications in Queueing Theory: A Survey," *IEEE Access*, 2025.
[6] H. Attou, A. Gueroui, and A. Bounceur, "Cloud-Based Intrusion Detection Using Machine Learning," in *Proc. IEEE Smart Computing Conference*, 2023.
[7] N. J. Mohan, S. R. Dubey, and S. Chaudhuri, "Machine Learning-Based Intrusion Detection System for Cloud Security," *IEEE Transactions on Cloud Computing*, 2023.
[8] A. R. Al-Ghuwairi, Y. Sharrab, and D. Al-Fraihat, "Intrusion Detection in Cloud Computing Using Time Series Analysis," *IEEE Access*, 2023.
[9] S. K. Singh et al., "Achieving Cloud Security Solutions Based on Machine Learning," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 2021.
[10] R. Kallimani, "Analysis of Queue Discipline for Wireless Sensor Nodes Using Queueing Theory," in *Proc. IEEE International Conference on Mobile Networks*, 2021.