



International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 9.935

Volume 15, Issue 10, October 2026

LineWise: Design and Development of an Intelligent Queue Management System with Waiting-Time Prediction

**Aman Kamble, Swati Chaudhari, Anushka Gajbhiye, Mrunali Shahare, Abhishek Telkapalliwar,
Prof. Virendra Yadav**

U.G. Student, Department of Computer Science & Engineering, Priyadarshini College of Engineering, Nagpur,
Maharashtra, India

U.G. Student, Department of Computer Science & Engineering, Priyadarshini College of Engineering, Nagpur,
Maharashtra, India

U.G. Student, Department of Computer Science & Engineering, Priyadarshini College of Engineering, Nagpur,
Maharashtra, India

U.G. Student, Department of Computer Science & Engineering, Priyadarshini College of Engineering, Nagpur,
Maharashtra, India

U.G. Student, Department of Computer Science & Engineering, Priyadarshini College of Engineering, Nagpur,
Maharashtra, India

Assistant Professor, Department of Computer Science & Engineering, Priyadarshini College of Engineering, Nagpur,
Maharashtra, India

ABSTRACT: Waiting in queues is a common experience in hospitals, banks, government offices, and other service centres. In many cases, people have little information about their current position in the queue or how long they may need to wait. This uncertainty can make it difficult to plan their time and can contribute to crowding around service counters. This paper presents LineWise, a web-based queue management system developed to make queue registration and monitoring more convenient while providing an estimate of waiting time.

The proposed system uses QR-code-based access to allow users to join a queue through a web interface and receive a digital token. Users can check their token details and queue position, while an administrator manages token status and updates service progress. A waiting-time estimation component uses queue-related information, including the number of people ahead and available service-duration data, to estimate the remaining wait. The study considers a queue-based calculation and regression-based prediction methods as possible approaches to estimation. Previous research has shown that waiting-time prediction can benefit from information about the current state of a service system and changes in demand and service capacity [1], [2]. Following this principle, LineWise connects queue monitoring with waiting-time estimation in a single application. The system should be evaluated through functional tests and quantitative prediction metrics to determine its reliability. The results of such evaluation must be reported using the actual implementation and test data.

KEYWORDS: Smart queue management, QR code, digital token, waiting-time estimation, web application, Linear Regression, Random Forest Regression, queue monitoring.

I. INTRODUCTION

A. Background

Queues are part of everyday life. People wait for appointments in hospitals, transactions in banks, documents in government offices, and services in many other places. Although waiting is sometimes unavoidable, not knowing when a turn will arrive can make the experience more frustrating. A person may have a token number but still have no clear idea of how many people remain ahead or whether a delay has occurred.

Traditional token systems help arrange people in an orderly manner, but they do not always provide convenient access to updated queue information. People may need to remain close to the service counter simply to hear their number being called. For service providers, manually managing entries and communicating delays can also become difficult when several people are waiting at the same time.

Research on queueing systems has established that waiting-time estimates can be made more useful by considering the current state of the queue, including the number of customers ahead of a particular arrival [1]. This suggests that a queue-management system should do more than assign token numbers. It should also help users understand the progress of the queue.

LineWise is designed with this need in mind. It combines QR-based registration, digital token generation, queue-position tracking, administrator controls, and waiting-time estimation in a web-based application. Users can access the system through a browser without needing to install a separate mobile application.

B. Problem Statement

In a conventional queue, users often have limited information about their turn and the expected waiting time. Manual token handling can also make it difficult to keep queue information updated when a service is delayed, a token is cancelled, or the next customer is called.

The main problem addressed by LineWise is the lack of a convenient, unified system for joining a queue, tracking its progress, and estimating the remaining waiting time. The proposed system aims to reduce this uncertainty by making queue information available through a web interface and providing administrators with a structured way to manage service progression.

C. Objectives

The objectives of LineWise are:

1. To provide a simple way for users to join a queue by scanning a QR code.
2. To generate digital tokens and maintain an orderly sequence of service requests.
3. To display the user's current queue position and estimated waiting time.
4. To allow administrators to manage tokens and update the status of service requests.
5. To investigate whether queue-related data can improve waiting-time estimates.
6. To evaluate the functionality of the application and the prediction method using appropriate test data.

D. Contributions of the Proposed System

The main contribution of LineWise is the integration of queue registration, monitoring, and waiting-time estimation into one web-based workflow. Instead of treating token generation and prediction as separate tasks, the system connects the estimate to the current queue state.

The framework is intended to support both users and service administrators. Users receive more information about their turn, while administrators can manage service progression from a dedicated interface. The accuracy and practical usefulness of the system must be established through testing rather than assumed from the design alone.

II. LITERATURE SURVEY

Research and development in queue management covers virtual queue applications, queue-state monitoring, and prediction of waiting time. Basgaran and Ibrahim describe the design of a web-based virtual queue management system that supports virtual tokens and QR-code access, demonstrating the relevance of browser-based entry for service-centre queues [1].

Work on mobile queuing has also combined QR-code scanning with the display of the currently served number and an estimated waiting time. Low Ying Yao's project report describes a mobile queuing solution using a first-come-first-served approach and QR-based tracking [2]. These approaches motivate the use of a lightweight digital interface, while leaving room for stronger live status synchronisation and evaluation.

Waiting-time prediction research shows that queue-state features can improve estimates. Benevento and colleagues studied queue-based predictors for emergency-department waiting time and reported reductions in prediction error when queue-related features were included [3]. A 2024 conference paper by the IEEE authors on machine-learning approaches to queue management compares regression approaches and an LSTM model, reporting the Voting Regressor as the best-performing model in its evaluated scenarios [4]. Such results indicate that predictive methods may be useful when sufficient and representative service records are available.

The proposed design differs in emphasis: it prioritises an accessible QR workflow and understandable, continuously refreshed queue information, while treating the baseline estimator as a transparent starting point. A more advanced predictive model can be added after enough reliable operational data have been collected.

III. METHODOLOGY

A. Development Approach

The development process begins with identifying the difficulties users and administrators face in a conventional queue. The requirements are then translated into separate modules for registration, token handling, queue tracking, prediction, and administration.

Each module is tested individually before integration. The registration module is checked for valid token creation, the queue module for correct ordering, and the administrator interface for proper status updates. The prediction module is evaluated separately against observed waiting times.

This approach helps distinguish errors in queue handling from errors in prediction. For example, an incorrect waiting-time estimate may result from a poor prediction model, but it may also occur because the queue position or service duration was recorded incorrectly.

B. Data Preparation

A waiting-time prediction model requires examples of queue conditions and the corresponding actual waiting times. Each record may include the number of people ahead, queue position, service duration, active-counter information, and the elapsed time between registration and the beginning of service.

The data should be checked for missing values, inconsistent timestamps, duplicate entries, and invalid token statuses. The dataset should then be divided into training and testing subsets before model fitting. Any data-dependent preprocessing must be learned from the training subset to avoid information leakage.

If the project uses generated or simulated records, this must be stated in the paper. Results based on simulated data indicate how the model behaves under the chosen simulation assumptions; they do not automatically demonstrate the same level of performance in a real service environment.

C. Prediction Models

Three approaches can be compared.

1. Queue-based baseline: Calculates the estimated waiting time using the number of people ahead and a representative service duration.

2. Linear Regression: Learns a linear relationship between the available queue features and the observed waiting time.

3. Random Forest Regression: Uses an ensemble of decision trees to learn potentially nonlinear relationships between queue conditions and waiting time.

The models should be trained and evaluated using the same data split. Their performance should be compared using the actual test-set predictions rather than assumptions about which method is likely to perform best.

D. Evaluation Metrics

The prediction models can be evaluated using Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and the coefficient of determination ((R^2)).

$$\begin{aligned} &[\\ &MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \widehat{y}_i| \\ &] \\ &[\\ &RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \widehat{y}_i)^2} \\ &] \end{aligned}$$

Here, (y_i) is the actual waiting time, $(\widehat{y}_i)$ is the predicted waiting time, and (n) is the number of test records.

MAE gives the average absolute difference between predicted and actual waiting times. RMSE gives greater weight to larger errors, while (R^2) indicates how well the model explains variation in the observed waiting times.

Functional testing should separately check token generation, queue ordering, duplicate handling, status updates, and position recalculation. These tests are necessary because good prediction metrics alone do not guarantee that the overall application behaves correctly.

IV. EXPERIMENTAL RESULTS

A. Application Workflow

The intended workflow begins when a user scans the queue QR code and opens the registration page. Once registration is accepted, the application generates a digital token and uses it to track the queue entry.

The user-facing interface displays token information, queue position, and estimated waiting time. Meanwhile, the administrator monitors the queue and updates service progress. When a relevant queue event occurs, the application updates the affected queue information and recalculates the estimate.

The prediction component uses the available queue features to produce a waiting-time estimate. Its reliability depends on the quality of the underlying records and how consistently service progress is updated.

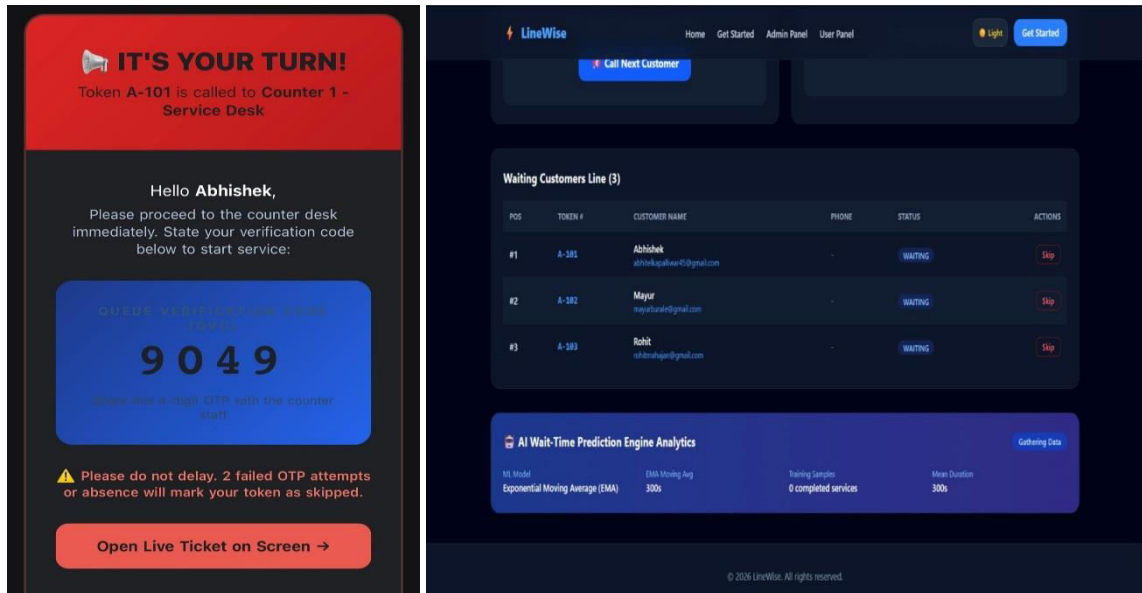


Figure.1.screenshot of the queue-registration or token-status page

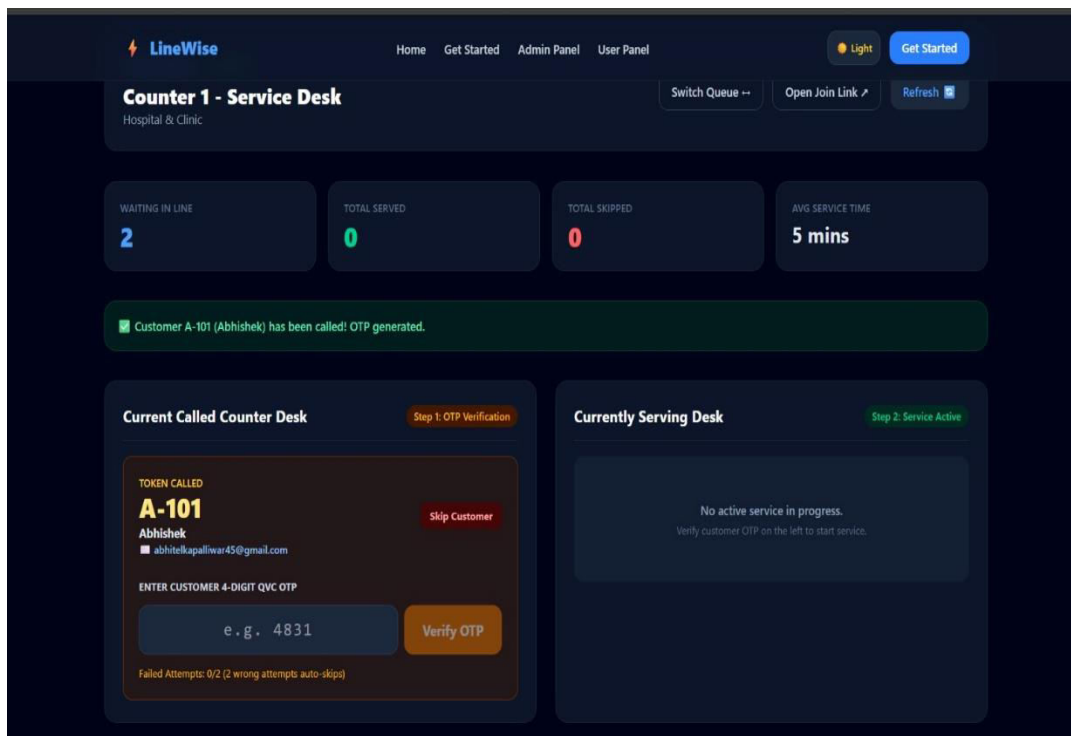


Figure.2. Screenshot of the administrator dashboard showing the active queue and service controls

B. Functional Testing

The following test cases can be used to evaluate the application.

Test case	Expected behaviour
Scan a valid QR code	The relevant registration page opens
Submit valid registration details	A digital token is generated
Scan repeatedly or submit a duplicate request	The configured duplicate-handling rule is applied
Advance the queue	The next eligible token is selected
Complete or cancel a token	The status and queue position are updated correctly
Refresh queue information	The displayed position reflects the current queue state
Change service-duration information	The estimate is recalculated when supported
Use multiple queues	Each entry remains associated with the correct queue

The actual outcomes must be recorded after the tests have been executed. The expected behaviours in this table should not be presented as verified results until they have been observed in the working application.

C. Prediction Evaluation

The following table can be completed after running the selected models on the held-out test data.

Prediction method	MAE (minutes)	RMSE (minutes)	(R ²)
Queue-based baseline	8.543	12.231	0.6432
Linear Regression	2.555	3.718	0.9670
Random Forest Regression	2.207	3.629	0.9686

A model with lower MAE generally provides more accurate estimates on average. RMSE should also be considered because occasional large errors can be particularly inconvenient when users rely on the estimated time to plan their activities.

The results should be interpreted in relation to the dataset and its limitations. If the test records are simulated, the discussion should explain the assumptions used to generate them. If real queue records are available, the paper should describe their source, collection period, and privacy safeguards.

D. Discussion

LineWise is intended to address a practical problem: users often have to wait without knowing how much time remains before their turn. Digital tokens and queue-position tracking make the service order easier to follow, while waiting-time estimation provides additional information for planning.

The literature indicates that waiting-time prediction becomes more useful when the current queue state is considered [1]. It also shows that changing arrival rates and service capacity can affect prediction quality [2], [3]. These findings support the decision to connect waiting-time estimation with the live queue-management workflow instead of relying only on a fixed time value.

At the same time, an estimate is only as reliable as the information used to calculate it. If the recorded queue position is incorrect or an administrator does not update a completed service, the displayed waiting time may become misleading. Changes in service duration can also cause actual waiting times to differ from predictions.

For these reasons, the system should be evaluated under different queue lengths and service conditions. The final discussion should report the measured prediction errors, describe the cases in which estimates were less accurate, and avoid claiming improvements that have not been demonstrated experimentally.

V. CONCLUSION

This paper presents LineWise, a web-based queue-management framework designed to make queue registration, tracking, and waiting-time estimation more convenient. The system combines QR-based access, digital tokens, queue-position updates, and administrator controls in a single workflow.

The waiting-time component extends basic token handling by providing an estimate of the time remaining before service. A queue-based calculation can serve as the baseline, while Linear Regression and Random Forest Regression can be evaluated to determine whether data-driven methods improve the estimates.

Previous studies have demonstrated the importance of queue-state information and the challenges created by changing arrival and service conditions [1]–[3]. LineWise applies these ideas in the context of an accessible web application. Its practical effectiveness must be established through functional testing and quantitative evaluation using the actual implementation and test data.

REFERENCES

- [1] P. S. Basgaran and N. Ibrahim, “Design and Development of Lasto Virtual Queue Management System,” *Applied Information Technology and Computer Science*. Available: <https://periodical.uthm.edu.my/index.php/aitcs/article/view/7646>
- [2] L. Y. Yao, “Mobile queuing solution,” Final Year Project Report / IMRAD, Universiti Malaysia Sarawak, 2019. Available: <https://ir.unimas.my/id/eprint/31998/>
- [3] D. Aloini, V. Mininno, N. Squicciarini, and colleagues, “Queue-based features for dynamic waiting time prediction in emergency department,” *Measuring Business Excellence*, vol. 23, no. 4, pp. 458–471, 2019. DOI: <https://doi.org/10.1108/MBE-12-2018-0108>
- [4] “Waiting Time Prediction in Queue Management: Leveraging Machine Learning Approach,” 2024 International Conference on Recent Progresses in Science, Engineering and Technology (ICRPSET), DOI: <https://doi.org/10.1109/ICRPSET64863.2024.10955915>.



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details