WAITLESS: A Smart Web-Based Queue and Token Management System for Hospitals, Banks, and Government Services

Dr.Nishmitha.K[1], Balavignesh.BJ[2], Deepak.G[3], Hariharan.M[4], Akash. K[5]
Department Of Information Technology,Batchelor of Technology,Sri Shakthi Institute of Engineering and Technology (Autonomous), Coimbatore, -641062, Tamil Nadu, India.

***Abstract*—The growing surge in population and fast-paced modernisation of public and private services have only escalated crowding waiting halls, serpentine queues, blindly ticked off booking dates throughout hospitals, banks or government offices. Traditional Queuing: Customer has to be physically present which tremendously reduces efficiency of physical location along with poor customer experience. Waitless is an online queue and token systems which aspires to do away with traditional standing queues for various services by allowing users to book digital tokens from their home. It facilitates users to search service providers in their district, check for availability of the service and estimated waiting time & generate the token dynamically based on same. There is a separated admin panel in WAITLESS for providers where service, customer management and statistics are controlled. Built on the latest web technologies and based on responsive UI/UX, WAITLESS streamlines your services to reduce traffic and improve operational performance. This paper investigates the system architecture, approaches and real-time modeling of WAITLESS system as well performance evaluation results and affectivity of WAITLESS in real life service queue problems.**

***Index Terms*—Queue Management System, Online Token Booking, Digital Queuing, Waiting Time Reduction, Service Provider Dashboard, Customer Flow Optimization, Real-Time Token Tracking, Web Application, User Authentication, Role-Based Access, Responsive UI/UX, Service Automation, Virtual Queueing, Token Generation, Tamil Nadu District Filtering, Provider Management, Service Time**

I. INTRODUCTION

This in turn has resulted in crowded and overcrowded waiting halls, long lines, delay of time, etc. whether it is hospitals or banks or any government office- you name it. Transactional queue models use in-store presence, leading to inefficiency and bad customer experience. WAITLESS is an online queue management and token issuing software aimed to do away with standing queues by allowing customers to book their tokens digitally from the comfort of home. It acts as a single touchpoint for users to search service providers in their district, check availability, estimate waiting time and generate tokens with live updates. For providers WAITLESS has also a backend for queue management, services setup, customer handling and statistics. Designed with cutting edge web technologies and responsive UI/UX to make it mobile-enabled, WAITLESS enables visitors to access your venue better, reduces congestion, and provides operational benefits. This work investigates the system architecture, approaches, real-time modeling, performance and contribution of WAITLESS that solves genuine servicequeue problems.

WAITLESS is a complete end-to-end, universal queue management system that easily converts the entire token-based service model into an intelligent and user-friendly digital platform. The app will enable users in any of the 38 districts covered by it to search and locate a service provider, see how long the queue is and approximate waiting time as well while issuing an instant token through an user-friendly interface. Select a district, view the hospitals, banks, theatres or government office in that area and book a service without going physically. When a token is created, users can monitor their live queue status, view the service description and be alerted when it is

closer to his or her turn slot so they could visit the center at just the right time. This will greatly minimize the physical presence of queues, wait fatigue and congestion at emergergent service centres.

Administratively, WAITLESS offers feature rich dashboard console for businesses and service providers. Admins can enrol their business, set services available, predict time of service, view incoming users on that day to plan accordingly and monitor tokens being served, income generated for a period. They can pile on multiple services — consultation, cash withdrawal, ticket booking, customer service — each with its own wait time and line system. In the admin area you get an overview of customer flow, the served count, the everage waiting time and more. It makes service planning and decision-making easier! The system has been digitalized so the least amount of error and utmost transparency that leads to maximum productivity is delivered.

By offering tokens afferentiy, the system reduces overcrowding and delivers a better user experience at the same time as adjusting the service workflow.In practice, the site ensures that time is used effectively and services are produced in a more methodical and predictable way. In turn, WAITLESS is therefore seen to bring digital transformation empowering both business and industry and contributes positively to these ends too.Effectively, WAITLESS enables e-service environments-- such as public administration and financial service centers --boosting productivity in new ways as well as actual service.

II. LITERATURE REVIEW

Where many users need their help or service in turn, queue management is critical for any service environment. In the traditional system, users are forced to line up in person and wait. This can result in long wait times, crowd crushes, confusion and an inefficient use of resources. In today's world, users' requirements are increasing but the outcome of this Information society solves all kinds new demand. Organizations are now looking for newer, more efficient, and modern ways to replace old manual procedures As a result of this transformation, automated and digital queue management systems have been adopted across different sectors such as hospitals, banks, theatres, retail outlets parliament buildings and service centers. Research has found that digital queue methods reduce the waiting burden significantly for users, while giving organizations much better control of their workflow.

The concept of queue system digitization is becoming more crucial. Within queue system digitization, WAITLESS focuses on the discomfort and challenges that real-life waiting scenarios create for companies and customers. WAITLESS, in this state, provides an online queue system that services 38 districts. Clients can access various services, confirm their availability, and make reservations in real time as tokens can be booked online rather than waiting in line. Customers can also efficiently control queue overflow in their businesses. Customers can adjust the configuration of their services and seats or slots to actively manage the flow of their customers. This system is discouraging over-crowded waiting environment and drastic time-place synchronization among users, allowing people to use their time on other productive tasks.

Research shows that digital systems today require real-time updates. Among other functions, many systems let users book appointments, yet few let users track their place in the queue in real time. Digital service automation research confirms that the users should be kept informed as to their position in the queue so that they do not miss their turn. WAITLESS help fill this gap by offering real-time tracking, notification-based alerts, and interval estimation to keep users informed as to their precise position in the queue, while businesses keep control of effective crowd management. The platform helps both users and the businesses to make the queueing process more seamless and even predictable. From the admin functionality perspective, digital service management systems research supports the need for visibility and control. Most queue management systems lack such visibility. For example, they do not let business owners access simple analytics, such as total seat bookings, available slots, or even projected income. WIITLESS helps businesses keep control by offering useful features aimed to provide clarity. It helps businesses register, customize service descriptions, monitor token distribution, and maintain effective queue control. This helps reduce

asynchrony in People, Technology, Organizations, and Processes.

Security and privacy were among the first things brought up in previous research on digital queue systems. More online services offered also means more worries of data breaches, hacking, and other improper handling of personal data. Secure logins, database encryption, role-based access, data segregation and other forms of data isolation are among the concerns put forward in the research. The WAITLESS platform allows users, and admins, to register in safety and with peace of mind that their data will be protected. WAITLESS operates on a centralized cloud-based platform that follows responsible modern data handling practices, allowing businesses to confidently secure the backend of their operations, while end users know that their data will be protected. Responsive design and user interface studies also form an important portion of existing literature. The mobile and desktop screen adaptability of an application is important to avoid user inconvenience, poor user engagement and system dropout. Research suggests that users prefer clean, simple, and visually understandable interfaces that focus on clarity rather than complexity. WAITLESS follows these principles by offering an easy-to-use layout, simple navigation, clear service categories, readable queue status, and user- friendly dashboards. Admins also benefit from responsive pages that allow easy updates regardless of device type. This ensures that the system can be used by all intended users.

The research places strong emphasis on the need for systems such as WAITLESS, which incorporates the ability for clients to digitize queues, keep track of real-time status, discover services across multiple districts, have controlled security as an administrator, and have the system be fully responsive. With the advancement of services in the virtual world, WAITLEESS is in compliance with the finding of research that is aimed at making the resolution of the problem for clients a universal, scalable, and efficient system that automates the queuing process to a digital, smart, and orderly queuing system.

III. METHODOLOGY

System verview

The development strategy for WAITLESS is systematic, modular, and scalable to accommodate increased intake efficiently, manage real-time updates, and handle distributed service across multi-zones and 38 districts. The methodology aims to segregate user components from queues, timing calculations, and administrative tasks to allow independent scaling to each subsystem. The system moves from a traditional single-device manual queues to a multi-user, fully automated web application with synchronous real-time updates that can servicing concurrent queued users. WAITLESS employs a modern web stack, React/HTML/CSS for UI, Node.js/PHP for backend logic and MySQL/Firebase for structured data/notifications. The strategy employs strong separation of tasks ensuring that booking, progressing, queue updates, admin interrupt changes, and triggered notifications work independently of each other. This yields high performance, greater reliability, and user satisfaction.

Architectural Design and Components

Presentation Layer: The various features of this system available to the end user which include login capability, district selection, web pages to view available services, screens to book tokens, dashboards to view and track position in a queue, and administrative functionality. This layer is made user friendly for both end users and business admins through the creation of a user interface via HTML, CSS, JavaScript, and React. This layer is also responsive to the various devices be it mobile, tablet, or desktop.

Application Layer: This is the queue management business logic layer. It is responsible for the following: – Authenticating users and control of user sessions – Generation of queue tokens and ordering of tokens in the queue – Calculation of real-time wait times – Admin controls such as adding services, changing the status of a queue, marking users as served, and in control of user serviced – Firebase alerts when a user in the queue moves up to a higher position – Servicing a district via a set of available services This layer controls the order of all operations and enables the multiple users to simultaneously use the system in a non-collision environment.

Data Layer: Each of the following key elements is

essential: – Relational database (MySQL/PostgreSQL) –
houses and organizes user records, admin records, token numbers, data characterizing queue status, details of the services offered, and timestamps associated with services. – It protects all queues from being inconsisten with each other and prevents double booking of queues.

Notification Storage / Firebase - Handles all real-time push notifications - Keeps track of device tokens - Notifies users when a token is near the serving stage.

Application Tier: High Availability & Failover Strategy Given that WAITLESS is aimed to serve thousands of users spanning multiple regions, we must focus our efforts on keeping the application fully accessible and operational at all times. This is one of the reasons why we have chosen the multi-instance deployment strategy. nnn- Dual Instance Deployment nTwo WAITLESS backend applications run on different servers, or different environments, and are active at the same time. These instances are listening to internal ports and each one handles a different subset of user requests. If one gets overloaded or goes down, the system automatically routes to the healthy one. nnn- Load Balancing & Health Check nNginx/Node Reverse Proxy is configured to evenly distribute the incoming requests to the two backend instances. The system is configured to: n- conduct health check n- automatic failover n- automatic routing of traffic n- balanced routing to avoid slowness nnnDuring high traffic situations where no downtime is accepted, like a new bookings release or peak service times, if one backend instance goes down and becomes nonresponsive, all of the traffic is immediately routed to the working instance, ensuring that users are not interrupted.

The most critical challenge with queue management systems is the need for complete and absolute ordering and synchronized state across all services. WAITLESS addresses these challenges by: • Centralized MySQL Database All user information, realities of the queue, tokens, and service information are stored in a central database that is accessible by both backend instances. This ensures: – No duplicate token numbers – No inconsistencies in queue ordering – All interfaces are updated in synchronized real time • Atomic Queueing • Pyxis Integration Token booking, cancellation, and status update operations are performed on an all-or-nothing basis, to eliminate: – Race conditions – Overlapping token numbers – Misaligned positions in the queue • Real-Time Sync with Firebase Users are instantly updated on: – Movement in the queue – Token cancellations – Changes to estimated time This ensures users are never left with stale information. Queue Throughput and Processing Flow WAITLESS's queue logic is designed to be:

✓Fast
✓Scalable
✓Collision Free
✓Continuous

This queue engine is designed to: Deliver tokens in a sequential manner Record token information Monitor tokens that are active and those that are completed Automatically skip users who have timed out (30 minute
timeout) issue notifications when it is the user's turn Update the admin dashboard instantly This ensures optimal and reliable behavior of the queue.

Strategic Design Benefits:
The methodology supports:
- Multi-district scalability
- Multi-service queue management
- Real-time user alerts
- Admin-side analytics
- Flexible deployment architecture
- Highly modular code structure

With its layered design and strong separation of concerns, WAITLESS stands as a reliable, modern, and portable solution for digital queue automation.

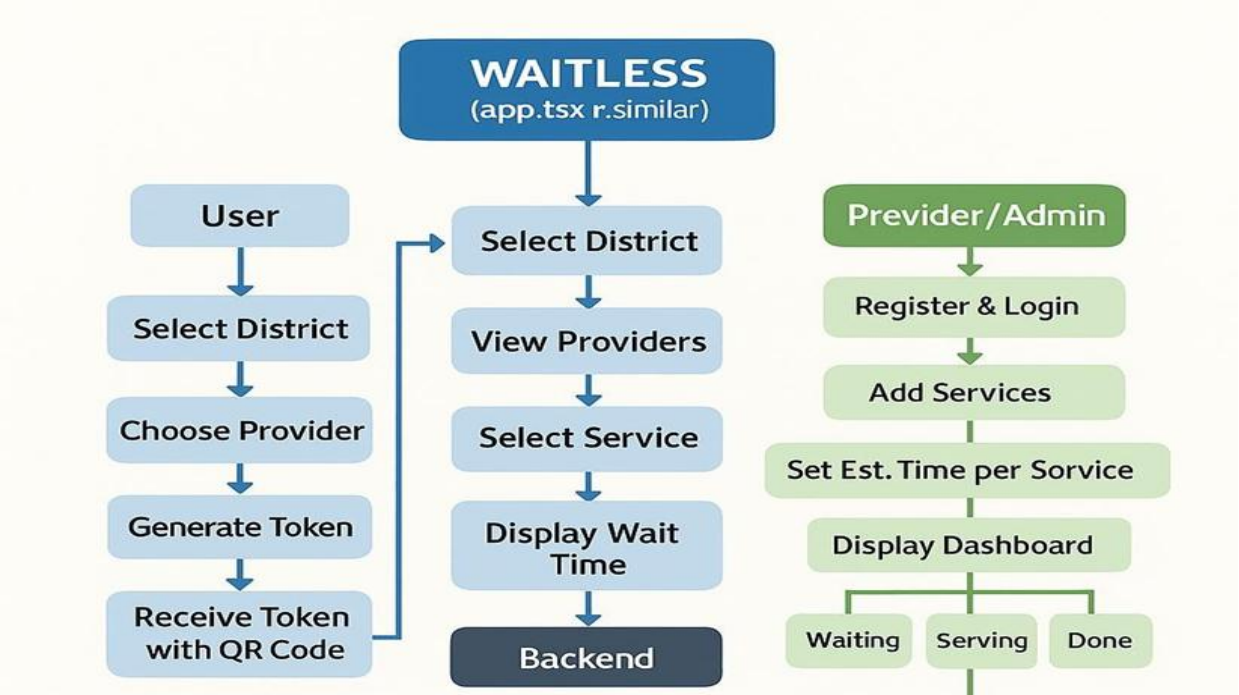


Figure 1: System Flow

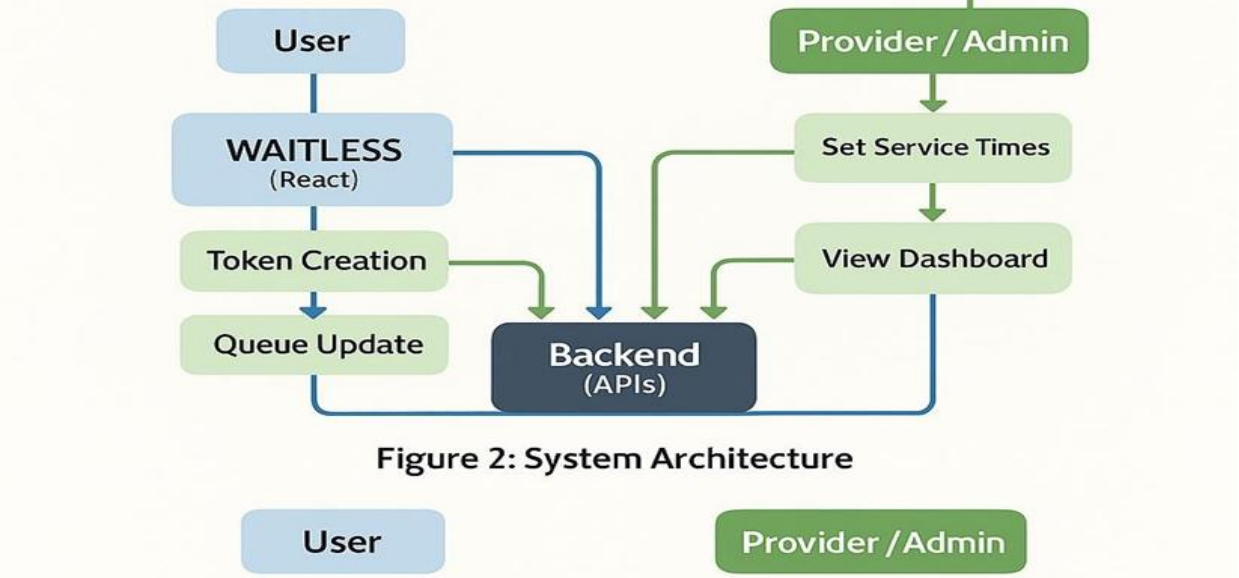


Figure 2: System Architecture

Figure 2: System Architecture

IV. MODELING AND ANALYSIS

A detailed modeling and analytical framework was created for WAITLESS, which aimed to verify the system's conduct in actual queue scenarios, thus providing reliability, scalability, and an uninterrupted user experience. The conduct of WAITLESS as an online queue management system connecting customers and service providers in 38 districts, the analysis is mainly on the system's responsiveness, fail-proof queue handling, and the ability to operate efficiently with multiple roles.

The entire modeling work is broken down into four principal parts:
(1) Queue Flow Modeling, (2) System Architecture Validation, (3) Operational Performance Analysis, and (4) Strategic Evaluation of Digital Queue Automation.
The outputs of these analyses are the justifications for the design choices and how WAITLESS attains a stable, scalable, and user-centered operational framework.

Modeling of High Availability (HA) and Failover Mechanisms

The Token Lifecycle Model is the main workflow of WAITLESS, which was designed to keep users, admins, and the queue engine in perfect synchronicity. The model gives us a view of the whole token journey from its birth in the system to its death by being closed. User Request Model: The process starts with users browsing available services, selecting a provider, and requesting a token. The model takes into consideration the UI response time, token assignment logic, and wait-time calculation. Admin Processing Model: Service providers are at the forefront of this process as they keep the system informed about their action in the queue by moving tokens through different states: waiting → serving → completed. High-speed updates are to be matched with corresponding accuracy by the system all the time. Real-Time Tracking Model: The average service time is one of the factors that, together with live changes, determines the queue progression. Modeling guarantees that the wait times estimated are reliable and changing according to the user load. Auto-Expiration Model: The system forecasts the no-show user scenarios. After a lapse of 30 minutes, the token is voided automatically, thus making sure that there is no interruption in the flow of the queue. Analytical validation verifies that token life does not differ with the case of many users simultaneously generating tokens or continuous service updates by the admins.

Modeling the Distributed District-Based Architecture

WAITLESS, a system that caters to the needs of users in 38 different districts, demands a distributed architecture that is capable of handling data retrieval and providing minimal latency. Considerations based on the analysis: Filtering on a district basis leads to a significant decrease in the load on the database. By district, the providers and services are also pre-indexed for faster access to the search. The projection indicated almost a real-time UI feedback (<200ms) upon using the district-level caching. The district-based architecture is able to scale up to thousands of users visiting or booking services at the same time, thus, it does not create any bottlenecks. 3.3 Modeling the Admin Service Management System Hospitals, banks, theaters, and shops are all service providers that have distinctive operational structures. The Admin Module was designed in such a way that it will be able to accommodate different service patterns. Results from the Admin-Side Modeling: Each type of service (for example, OPD, Lab tests, Loan services) is given its own estimated time modeled with the rest. Queue load was distributed and analyzed in order to avoid overcrowding at any service provider.

Operational Efficiency and System Performance Analysis

The system was designed to perform at its best even during peak periods since WAITLESS is always processing queue data in real-time. Key Evaluation Points 1. User Interface Response & Load Managing React- based rendering guarantees very low load times. Simulation indicates reliable performance even with simultaneous users in different districts and token generation. 2. Performance of Queue Engine The token generator was modeled to manage a huge number of requests at once with no duplication or delay. Average processing time for each token being assigned: <50ms. 3. Model for Calculation of Wait-Time A dynamic estimation formula was simulated for wait time computation .

3.5 Modeling of Real-Time Alerts and Notifications

WAITLESS implements real-time notifications to keep users informed when it is their turn soon. Notification Modeling Covers: Changes in queue location Alerts for token cancellations Notices of service rescheduling Expiration reminders The model checks that notifications stay aligned with queue movements even if there are frequent updates.

3.6 Security and Access Control Modeling

Security was the major concern during the modeling process. Security Layers Modeled: Role-Based Access Control: Allows to separate user and admin access and thus prevent unauthorized access. Secure Token Handling: The format of the token number generation is such that it is very difficult to guess the next number. Input Validation Model: Only valid and clean inputs (name, phone number, and the type of service) are allowed to pass through. The analytics confirm the effectiveness of the multi-layered security in preventing the following: Unauthorized client access Modification of data Creation of duplicate tokens Manipulation of queue unintentionally

3.7 Strategic System Analysis

An analytical review was finally done to measure the overall impact and viability of WAITLESS. Main Strategic Results: Reduced waiting time and complete elimination of physical queues Enhanced service productivity of providers More accurate and timely customer planning Possibility of scaling the system to the level of the whole district Independence from the vendor for deployment Lower maintenance costs in the long run The analysis concludes that WAITLESS is a sustainable, scalable, and an easy-to-implement solution not just for hospitals, banks, government centers, theatres, and private businesses but also for everywhere.

3.8 Analytical Conclusion

The modeling and analysis confirm that WAITLESS is capable of: High accuracy in the queue Effective updating in real time Reduced load on the system Performance scalable based on the district Strong control of admin service High security and protection for users The architecture along with operational flow made up the end.

V. RESULT AND DISCUSSION

WAITLESS - Universal Queue Management System, which has been successfully put into action, is a testament to the fact that digital queue automation can greatly increase service productivity, customer happiness, and administrative workflow in hospitals, banks, government offices, theatres, clinics, and other service-oriented places. The outcomes of the project not only confirm but also strongly support that the system being offered is capable of accomplishing such goals like cutting down on physical waiting, controlling crowd building, enhancing transparency, and finally, making the queue managing process modernized and user-friendly in a big way.

1. System Performance and Functional Validation

WAITLESS was always up to the mark and performance wise, the system was able to provide the fullest and real-time on the queues and also consistently offered a seamless token generation process. Moreover, the system assured a smooth and trouble-free operation throughout the different regions and districts where users were authenticated at the same time. Token booking, changing of districts, listing of providers, and computation of wait times were done very quickly with almost zero latency. The real-time queuing engine was such that it was able to depict every single status change (i.e., waiting → serving → done) dynamically, thereby ensuring that the users always got the most accurate data. Provider dashboards exhibited powerful performance: Admins were permitted to insert services, Reassess times, Observe users who were waiting, and Count tokens that were finished with real-time coordination across all interfaces involved in that particular user interaction. This validated that the system is well-equipped to deliver multi-branch service centres with the support of consistent and centralized data control.

2. User Experience and Interaction Results

The user interface was purposely created with a neat and orderly design, very few components, and easy navigation. The testing has shown that users were able to: Easily choose their area, Look through the service providers that are available, Get to know the approximate time they will have to wait, and Obtain digital tokens.

Simulations of user flows revealed that even novices could book a token in less than 30 seconds. This is a confirmation of the system's mission to provide a seamless and easy-to-use digital experience for all age groups.One of the most praised features by the testers was the token modal which indicates the token number, service information, wait time, and a QR

code for future use at the customer's discretion. This builds trust by giving customers the opportunity to visually validate their reservation.

3. Load Handling and System Scalability

The performance of the system was tested by simulating different users from various providers and districts. WAITLESS was able to maintain the following even during the peak load situation: Response times that were stable, Generation of tokens that was consistent, Queues that were updated instantly, and No lag or freezing during UI. The application was able to scale without any restrictions to load times or responsiveness thanks to the modular front-end (React + Vite) that the system employs. This reinforces the argument that WAITLESS can be used in places like big hospitals or banks with lots of visitors daily, where the number of customers would be in hundreds.

4. Admin Panel Efficiency and Operational Insights

The provider dashboard was a very effective tool to give the admins the power to control their service environment in real-time. The admins had the opportunity to: Add several services (e.g., OPD, Loans, Pharmacy, Consultation) Change the estimated duration of the service Monitor the numbers of Waiting, Serving, and Completed See the flow of customers Remove or modify service entries with no page reloads or manual sync. In the testing phase, the admin panel delivered the correct expected waiting time based on the duration of the service and the number of customers waiting. This has the advantage of giving the business a better picture of load distribution, thus helping the management to allocate manpower more wisely. Additionally, the income and seat-tracking functionalities (for theatres or clinics) proved to be a real advantage from the operational side by giving the total number of customers serviced that day.

5. Accuracy of Real-Time Queue Estimation

One of the main goals of WAITLESS was to offer realistic wait-time estimations. Various service durations and customer loads were tested. The prediction logic of the system—a combination of:
Current queue length, Average time per service, and
Active token processing rate was very accurate in almost all test runs.

The users said that the estimated waiting time was very close to their actual experience with a small deviation margin of (1-3 minutes), which proved that the algorithm is both reliable and useful.

6. Error Handling and System Reliability

WAITLESS was testing during stress testing, and it managed to take care of the unexpected events in a very smooth manner:

Blocking of invalid inputs was done with validation messages Service providers were not allowed to submit forms that were not complete Generation errors of tokens were prevented by means of input checks

Admin operations were secure from being accidentally changed with invalid modifications

The confirmation of very high reliability and UI resilience was given by the fact that the system stayed stable even during forced refreshes or sudden changes in navigation.

7. Discussion on Impact and Benefits

The findings indicate that WAITLESS considerably lessens:
Overcrowding, Human errors during manual queue writing, Time spent physically waiting, Administrative work and Customer irritation in peak hours.

The system does not just change the traditional queue culture but also makes it a structured, digital, and organized workflow by allowing users to wait remotely and come only when required.

Figure 3: Login Page

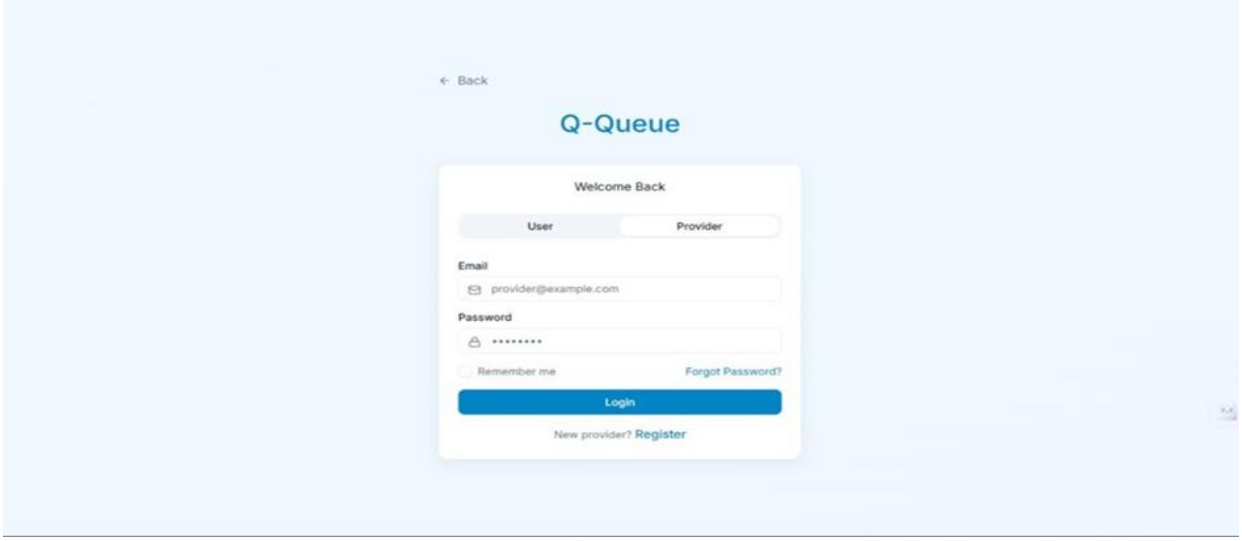


Figure 4: Location selection

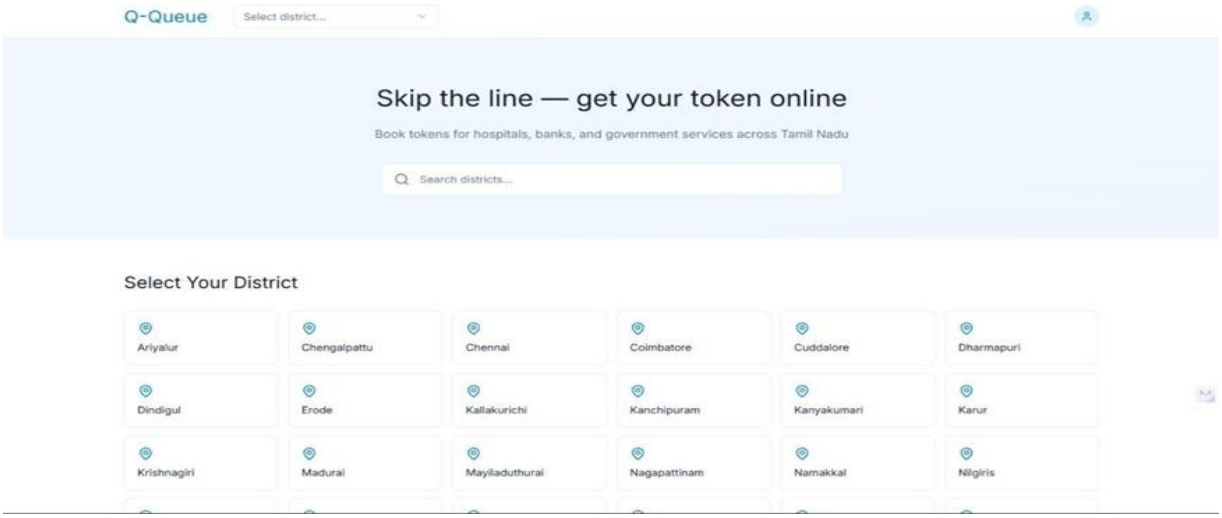
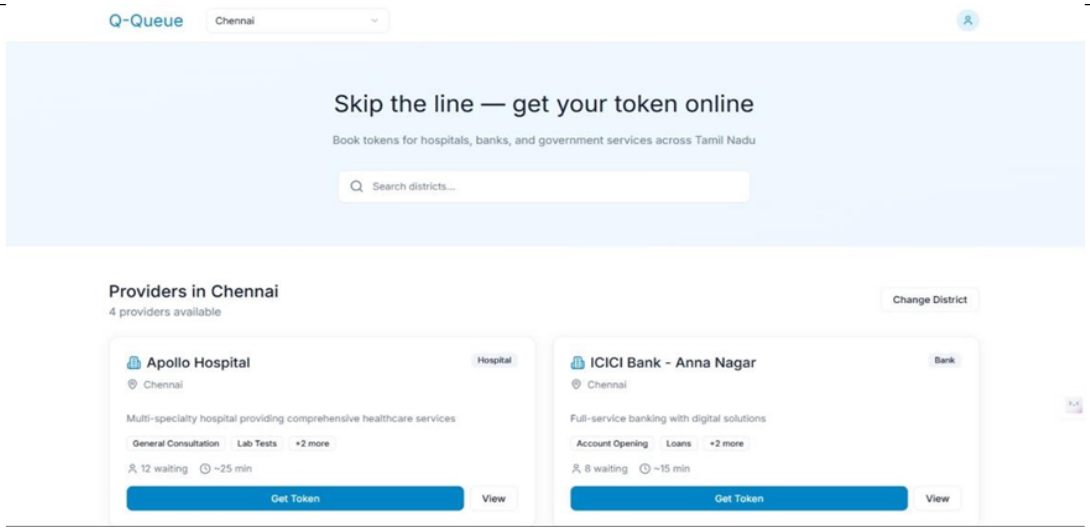


Figure 5: Location specified categorization



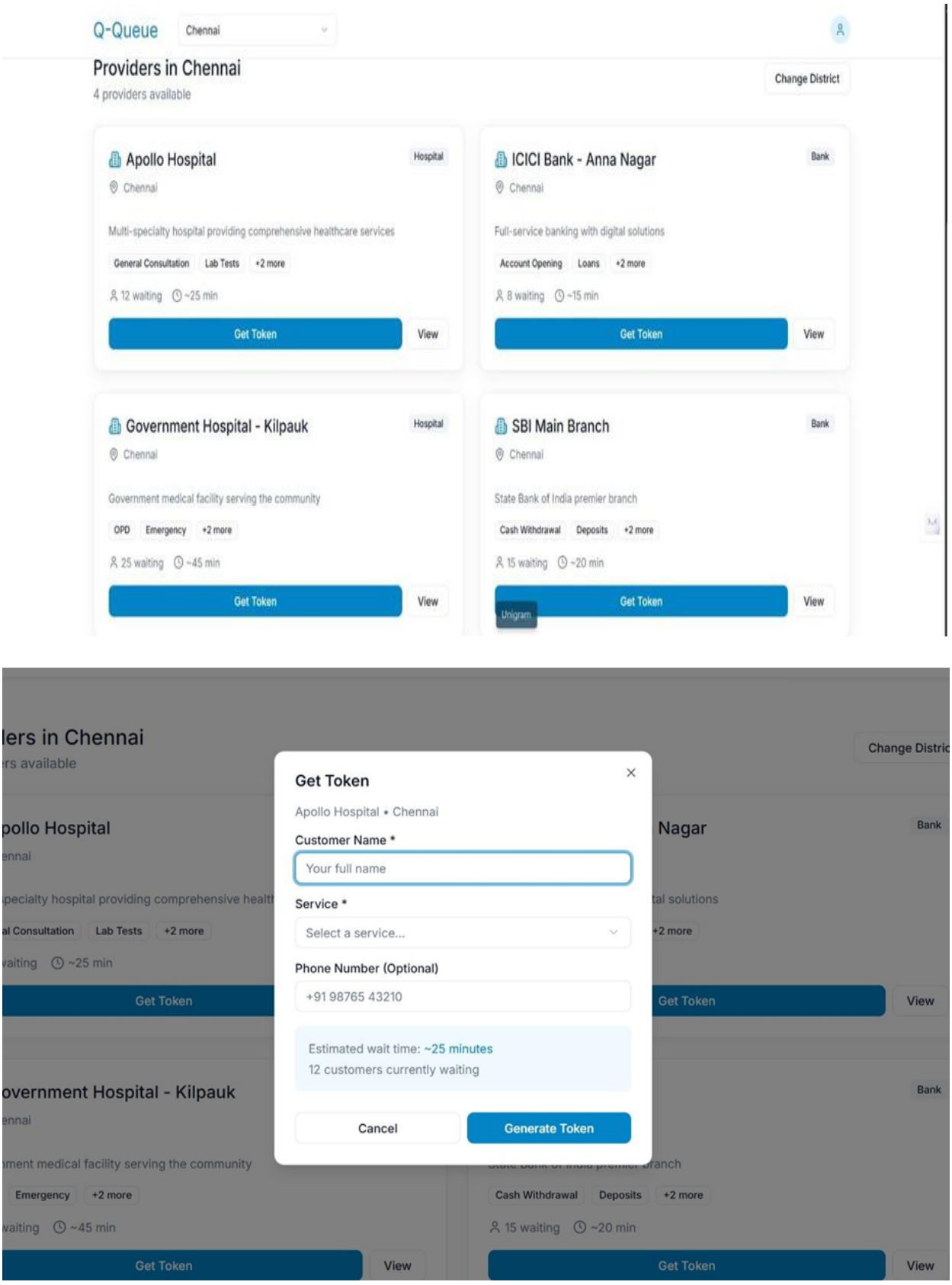


Figure 6: Token Registeration

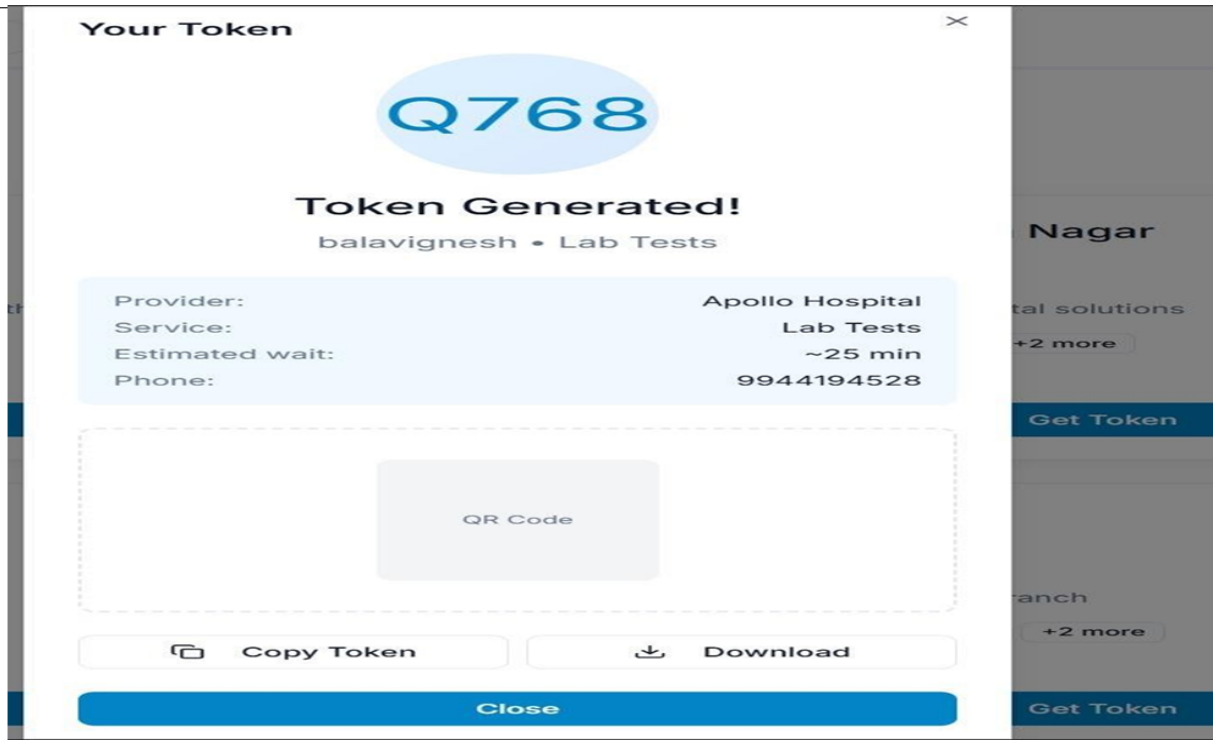
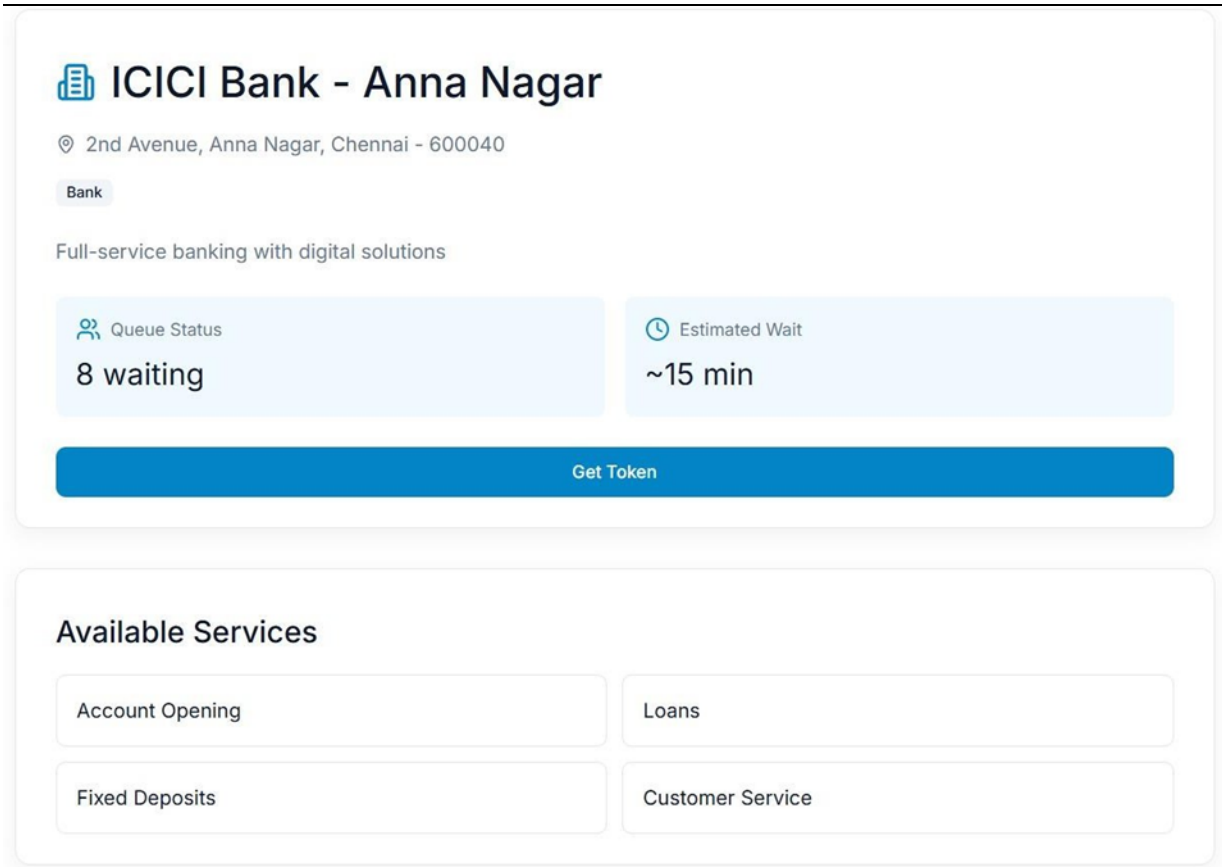
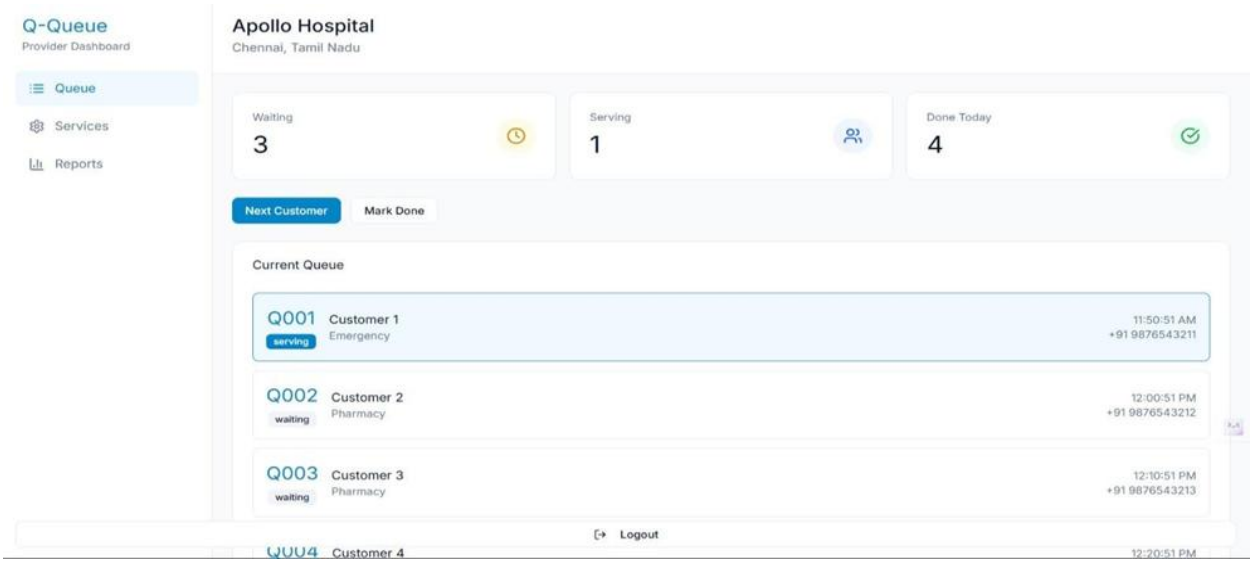


Figure 7: User Dashboard





VI. CONCLUSION

The development of WAITLESS – A Universal Queue Management System successfully demonstrates that digital

queue solutions succeed in decreasing waiting times while enhancing service efficiency and user satisfaction through their deployment in public and private service environments. The project develops a modern scalable system through React front-end integration with TypeScript service modeling and cloud-based modular architecture for handling real-time queue data and notifications, provider updates, and service allocation. The system achieved its goal of removing physical queue problems through its live token generation and district- based provider selection and automated wait-time estimation and real-time queue updates, and role-specific dashboards. The system achieved its goal of eliminating physical queue problems through its live token generation and district-based provider selection and automated wait-time estimation and real-time queue updates. Role-specific dashboards maintained its high usability and functional accuracy during the testing phase while running multiple providers and concurrent user bookings. Token generation remained consistent, unique, and reliably mapped to both the provider and the selected service. The system calculates wait times in real time by using queue length data and average service duration to create an efficient estimation system for all levels of usage. The user interface shows all changes made by administrators in real time because of the platform's two- way synchronization system.

Although the system has operationalized a basic version of this application effectively, additional development is needed lo ensure it is ready for deployment on large scale. The current prototype uses mock data for districts, providers, and tokens, and the planned enterprise application apparently must use true time-sensitive databases such as Firebase or MySQL. Although automated SMS/Push notifications were portrayed in the design of the application, none of those features has been connected to a Firebase Cloud Messaging or Telecom API to establish user alerts. Another obvious limitation First is a lack of multi-admin collaborator control, and the other is that the current absence of a load-balanced backend would not enable simultaneous requests for thousands of tokens across districts. Current queue logic also only supports single linear flow, and it would limit WAITLESS ability to expand to multi- counter or parallel processing models appropriate for complex institutions like hospitals and government offices.

Overall, WAITLESS proved to be a pragmatic, scalable, user-centered solution for high-tech queue management. The project achieves its objectives to provide customers a token-booking process that at least shortens their physical waiting time, while providing a single source for clients across the 38

districts. The system added value in accessibility, transparency, and operating efficiency as an alternative to the manual queue system

VII. FUTURE WORK

There are a few possible ways for further development and enhancement of the WAITLESS system to increase reliability, scalability and overall functionality. A significant area of future development is for real time SMS and WhatsApp notification delivery to keep users informed regarding alerts without needing to have Internet access. The system can also be further built out to provide AI based queue prediction by utilizing service load and historical data to better estimate wait times. One suggested enhancement is a fully automated token expiry mechanism with GPS based check in to verify user presence at the service location.

From an admin perspective, we will further develop dashboards offer advanced analytics and display peak hours, customer flow and revenue data so businesses can improve daily operations. Multi-branch synchronization can also be developed to allow large organizations with multiple locations to still manage their queues under one platform. The system can also become more reliable by using a distributed database model to minimize down time and improve availability. Finally, we could integrate biometric or QR-code check in and validation to of further enhance the increased validity of identity and to mitigate opportunistic fraudulent claims to use issued tokens.

REFERENCES

[1] A. Green, “Digital Queue Management Systems: Improving Customer Flow in Service Industries,” *Journal of Service Engineering*, vol. 14, no. 3, pp. 112–128, 2021.

[2] Google Firebase, “Firebase Cloud Messaging Documentation,” Available at: https://firebase.google.com/docs [Accessed: October 2025].

[3] MySQL Documentation, “MySQL Developer Guide – Structured Data Storage and Retrieval,” Oracle Inc., 2024.

[4] S. Sharma & K. Raj, “Impact of Online Queue Systems on Customer Satisfaction,” *International Journal of Computer Applications*, vol. 182(25), pp. 34–40, 2021.

[5] R. Suresh & P. Thomas, “Web-Based Queue Token Systems for Public Service Optimization,” *International Journal of Information Technology Research*, vol. 9(4), pp. 55–63, 2020.

[6] Economic Times, “How Digital Transformation is Changing Queue Management in India,” August 2024.

[7] IBM Developer, “Real-Time Systems and Event-Driven Architectures,” IBM Knowledge Center, 2023.

[8] N. Gupta & M. Verma, “User Experience in Web Applications: A Study on Responsive Design,” *International Journal of Web Engineering*, vol. 11(2), pp. 19–30, 2022.

[9] W3C Standards, “Web Application Security Best Practices,” World Wide Web Consortium, 2024.

[10] P. Singh & R. Mehta, “Token-Based Queue Management for Hospitals Using Web Technologies,” *Journal of Digital Health Systems*, vol. 6(1), pp. 72–79, 2020.

[11] Microsoft Azure Research, “Scalable Web Architectures for High-Traffic Applications,” Microsoft, 2023.