Undergraduate Conference on Intelligent Computing and Systems (UCICS 2025)
26-27 February, 2025; Varendra University, Rajshahi, Bangladesh

Electronic and Micro-Web based Approaches in Queue Management System with Prediction on Waiting Time

Amena Akter
Electrical and Electronic Engineering
University of Rajshahi
amena.akhi5@gmail.com

Bipin Saha
Electrical and Electronic Engineering
University of Rajshahi
bipinsaha.bd@gmail.com

Tapodhir Karmakar Taton
Electrical and Electronic Engineering
University of Rajshahi
tapodhirtaton@gmail.com

Khalid Mahmud
Electrical and Electronic Engineering
University of Rajshahi
khalidmahmud.km10@gmail.com

Md. Inzamul Haque
Electrical and Electronic Engineering
Varendra University
injamhaque1212@gmail.com

Arifa Ferdousi
Computer Science and Engineering
Varendra University
arifa@vu.edu.bd

Md. Johirul Islam*
Physics
Rajshahi University of Engineering & Technology
johirul@phy.ruet.ac.bd

Shaikh Khaled Mostaque*
Electrical and Electronic Engineering
University of Rajshahi
misha@ru.ac.bd

***Abstract*—This paper presents two distinct approaches to the development of a Queue Management System (QMS). The first approach outlines the design of an electronic QMS using an ATmega32 microcontroller as the central control unit, capable of serving up to 99 customers per counter and effectively displaying the average waiting time. The second approach introduces an advanced QMS that uses a Web environment, eliminating the need for complex electronic circuitry. This web-based system operates within a local area network, allowing customers to request queue tokens, check predicted waiting times, view the current service status, and receive call notifications via a web browser. Both approaches implement a Diffused Queuing Enhanced First Come, First Serve (DQE-FCFS) scheduling mechanism, optimizing the customer service process. The comparative analysis highlights the advantages and trade-offs between the electronic and web-based QMS implementations.**

***Index Terms*—Queue Management System (QMS), Average Waiting Time, ATmega 32, Customer Service Optimization, Diffused Queuing Enhanced First Come First Serve (DQE-FCFS)**

I. Introduction

A Queue Management System (QMS) provides real-time updates on waiting status, estimated waiting time, and assigned service counter, streamlining operations in service-based industries. In the wake of the COVID-19 pandemic, QMS has become crucial for maintaining social distancing and minimizing mass gatherings, especially in crowded areas like banks, retail shops, and telecommunication offices, offering an effective solution for managing customer congestion while ensuring health and safety compliance.

Over the years, various approaches have been proposed to address customer queueing and congestion. These strategies range from the Shortest Processed First (SPF) and First Come First Serve (FCFS) to the Single Queue (SQ), Multiple Queue (MQ), Diffused Queue (DQ), and Head of Queue (HQ) models [1]. Among these, FCFS has emerged as one of the fairest methods, ensuring equal treatment for all customers. Previous FCFS implementations have utilized electronic queue control systems powered by PIC microcontrollers, capable of managing 100 customers at four service counters [2]. These systems often employ token-based displays, with some designs incorporating centralized displays for all counters and others offering dedicated displays for each counter [3].

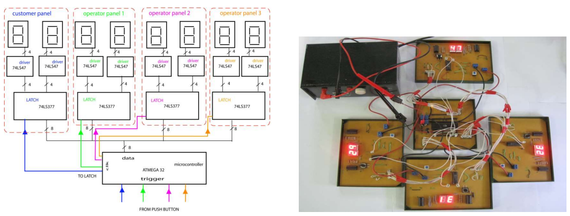
Advanced QMS implementations have explored novel features, such as dual PIC microcontrollers in master-slave configurations. One microcontroller handles transmission, while the other manages reception, offering the advantage of bidirectional displays for both operators and customers [3]. Additionally, speech-based notification systems interfaced with MATLAB provide audible queue announcements, improving customer awareness, but increasing system complexity, power consumption, and cost [4].

Other studies have used GSM transceivers for communication, enabling interaction between customers and server systems through SMS [5] [6]. While this approach ensures wider accessibility, it requires precise synchronization between the server and multiple users. Sensor-based techniques have also been proposed, such as the work by Usman et al., which employs IR sensors to monitor customer flow. These sensors achieve accuracy rates of 80% for counting incoming customers and 90% for those served, offering a practical, albeit limited, solution for real-time queue monitoring [7].

Conventional physical QMS are constrained by several challenges, such as the mandatory physical presence of users, inefficient space and time utilization, high operational expenses, and limited scalability [8]. In this context, We present two innovative queue management approaches: an electronic QMS using ATmega32 microcontrollers and a software-based QMS built with Python and Flask. Both systems employ the Diffused Queue Enhanced First Come First Serve (DQE-FCFS) methodology. The electronic system ensures robust hardware control, while the software system offers features like live tracking, streamlined registration, and service time prediction based on the last three customer service windows. In both systems, customers register, and the server dynamically allocates requests to available counters for efficient processing.

II. Electronic Approach

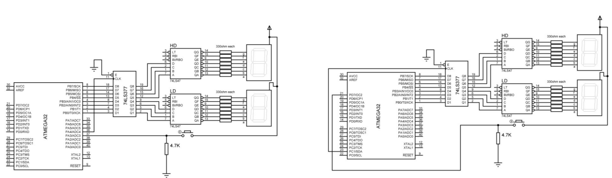
The system design, as shown in Figure 1(a), consists of a customer panel, three operator panels, and a timer panel that displays the average estimated service time for each customer. The Electronic Queueing System (EQS) is built using an ATmega32 CMOS 8-bit microcontroller based on an AVR-enhanced RISC architecture. 74LS47 ICs are used as BCD-to-seven-segment pattern converters, enabling clear numerical displays.



(a) Block diagram (b) Physical View

Fig. 1: Electronic QMS with ATmega32, 8 bit latch and seven segment display

In the design (Figure 2), when a customer presses a switch, the customer panel counter increments, and the corresponding token number is shown on the seven-segment display. Customers are required to remember their token numbers and wait for their turn, which is announced on the operator panels. An integer variable in the program tracks the customers in the queue.



(a) Customer Panel (b) Operator Panel

Fig. 2: Microcontroller Circuit Implementation

Each operator panel has push buttons to call token numbers from the queue to the seven-segment displays. Three operator panels function in parallel, calling token numbers as needed. Once an operator finishes serving a customer, they press a button to call the next token number. A clock pulse is sent to a latch, updating the value on the shared 8-bit data bus. The latches retain their previous values until the corresponding push button is pressed.

When the last token in the queue is served, the operator display does not increment further until a new token is issued by the customer panel. The system handles up to 99 tokens, with the two-digit limitation of the seven-segment displays. Once the maximum is reached, the system halts until manually reset via a designated switch. Binary-to-BCD conversion is used on the data bus, as the seven-segment driver ICs accept BCD input.

Both 8-bit and 16-bit timers (TIMER 0 and TIMER 1) are used to measure service session durations. The system calculates the average service time based on the three operator counters, which maintain averages of three consecutive service times. This reduces queue delays by distributing the workload and estimates the waiting time by multiplying the average service time with the number of remaining customers. The predicted waiting time is displayed in seconds, providing real-time updates. This design enhances queue management, minimizes delays, and improves customer satisfaction. Figure 1(b) shows the final system.

III. Web-based QMS Implementation

The micro web-based Queue Management System (QMS) consists of three primary components: the main computational unit (server), service counters, and customer-centric features such as tracking and waiting time prediction. Figure 3 illustrates the system architecture.

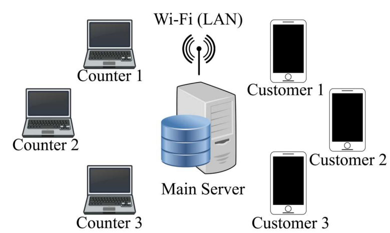


Fig. 3: Proposed model of QMS with micro web approach

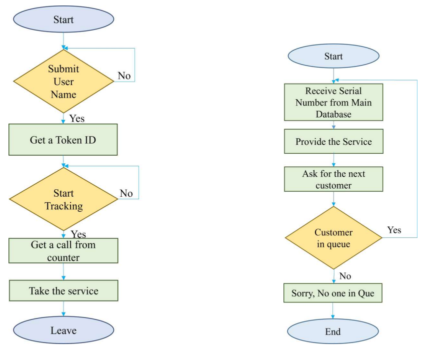
A. System Design and Customer Interaction

The micro web server, built using Python and Flask, provides a lightweight yet powerful framework for both local area network (LAN) and internet-based deployments. Customers connect to the system via the same LAN, such as Wi-Fi, to access the service. Figure 4(a) depicts the customer panel, where patrons can input their names to receive a unique serial number. Upon submitting their details, customers are issued a token ID, enabling them to enter the queue.

Once registered, the tracking feature becomes active, allowing customers to monitor their estimated waiting time. Unlike traditional queues, this system operates on the *Diffused Queue Enhanced First Come, First Serve (DQE-FCFS)* model, which organizes customers in a virtual queue, streamlining the process without the need for a physical line.

B. Counter Operations

On the service counter side, the system employs a classic FCFS algorithm. As illustrated in Figure 4(b), the main server sequentially fetches customer serial numbers from the database and assigns them to available counters upon request. A counter calls the serial number to provide service, and upon completing the task, it requests the next assignment.



(a) Customer Section (b) Counter Section

Fig. 4: Flowchart

During this process, two timestamps, T_1 (start time) and T_2 (end time), are recorded. The difference between these times represents the service duration, which is sent back to the main server for tracking and prediction. If the queue is empty, a notification displays: *"Sorry, No one in Queue."*

C. Main Server Functionality

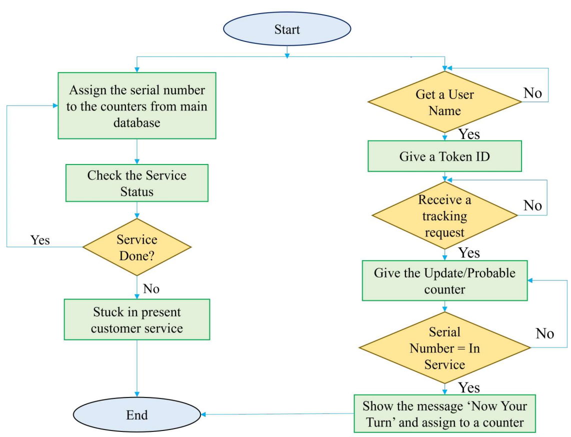


Fig. 5: Flowchart for Server Operation

Figure 5 highlights the critical role of the micro web server as the system's central hub. The server assigns serial numbers to counters, waits for service completion acknowledgments, and then allocates the next serial. Additionally, it manages the tracking feature, ensuring customers receive real-time updates on their service status.

When a counter begins serving a customer, the server records the service start time, T_0. For every subsequent customer, the service times T_n, T_{n-1}, and T_{n-2} are updated using a sliding window approach. The system calculates the average service time of the last three customers as:

$$\text{Average Time} = \frac{T_n + T_{n-1} + T_{n-2}}{3}$$

If the customer currently being served has serial number Y, and the customer in question has serial number X, the probable waiting time is estimated using the sliding window average as:

$$\text{Probable Waiting Time} = (X - Y) \times \text{Average Time}$$

D. Efficiency and Enhanced User Experience

The system's ability to predict service times based on the average duration of the last three transactions provides customers with accurate waiting time estimates. By combining real-time tracking and optimized service allocation, the micro web QMS minimizes delays and enhances the overall user experience, ensuring smooth and efficient queue management.

IV. Result & Discussion

A. Electronic Queue Management

The proposed system uses a customer panel for token generation and operator panels for token management, as shown in Figure 2. When a customer presses a switch on the customer panel, the token number increments and is displayed on a seven-segment display. Customers keep their token numbers and wait for their turn, which is displayed on the operator panels. An integer variable tracks the queue status. Each operator panel includes push buttons to call token numbers sequentially from the queue. Three operator panels operate in parallel, updating token numbers on the seven-segment displays through a shared 8-bit data bus. A latch mechanism ensures data integrity by retaining previous values until a button press triggers an update.

The system halts further operations when the token count reaches the 99-token limit due to the two-digit constraint of the seven-segment display, requiring a manual reset. Additionally, binary-to-BCD conversion is implemented to ensure compatibility with the seven-segment driver ICs. However, the system lacks real-time data updates and feedback to customers regarding waiting times. Furthermore, while the system was initially designed without a printer, operational scalability necessitates a printer for token printing. To address these limitations, a new approach using a software-based queue management system was implemented.

B. Software Queue Management

1) Token Registration and Tracking: Customers join the queue by connecting their devices to the local network and accessing the web application. They enter their details on a submission page, which are stored in the database. After submission, they are redirected to a confirmation page with the tracking feature activated, as shown in Figure 6.

For example, if Customer A is assigned serial number 55, the tracking system updates their queue position and provides real-time progress updates, as illustrated in Figure 7(a).

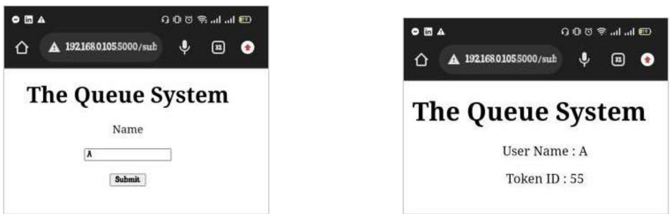
2) Counter Initialization: When service counters are initialized, the system automatically assigns the next available serial number to the corresponding counters. For instance, as shown in Figure 7(b), Counter 1 is currently serving the 11th customer while being ready to process subsequent requests.

TABLE I: Technical Comparison with existing QMS

Feature	Electronic QMS	Web-Based QMS	Existing Systems
Processing Unit	ATmega32 microcontroller	Server-based architecture (Flask & Python)	PIC microcontrollers, GSM-based, or cloud-based
Scalability	Limited (up to 99 customers)	Highly scalable with database management	Varies (cloud-based are scalable, others are not)
User Interface	Seven-segment display	Web-based UI with real-time tracking	Mobile apps, SMS, or local kiosks
Queue Handling	Manual token system with limited automation	Automated queue allocation and live tracking	Combination of manual and automated systems
Notification System	Average of last three service times	Sliding window-based real-time prediction	Basic estimates or fixed queue-length calculations
Waiting Time Prediction	Visual display only	Web notifications and updates	SMS, email, or app-based alerts
Hardware Requirements	Microcontroller, 7-segment display, switches	Server, database, and web interface	Varies based on implementation (some need IoT devices)
System Flexibility	Fixed hardware, difficult to upgrade	Easy software updates and feature additions	Cloud-based systems offer flexibility, hardware-based are rigid

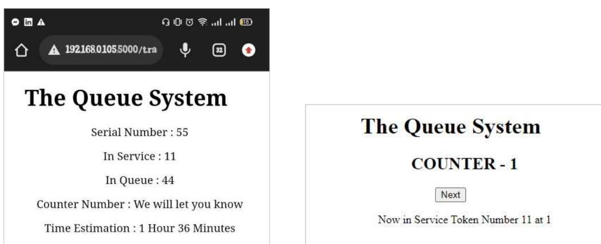
TABLE II: Economic Comparison with existing QMS

Aspect	Electronic QMS	Web-Based QMS	Existing Systems
Initial Cost	Low (ATmega 32, simple circuits)	Moderate (server setup, database)	High (IoT devices, cloud services)
Maintenance Cost	High (hardware wear & tear, repairs)	Low (software updates, minimal hardware)	Varies (cloud-based has subscription costs, hardware-based has repair costs)
Operational Cost	Minimal power consumption, but manual reset required	Higher initial but cost-effective in long run	Cloud-based have recurring fees, hardware-based need replacements



(a) (b)

Fig. 6: (a) Name submission and (b) confirmation



(a) (b)

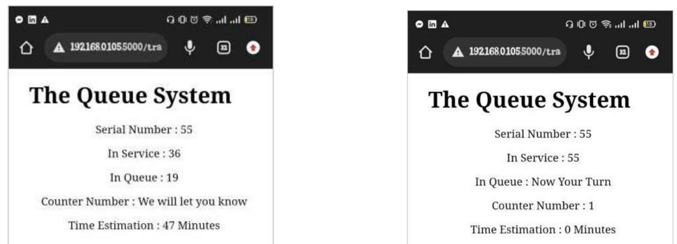
Fig. 7: (a) racking feature enabling and (b) Counter 1 at in service

This automated allocation ensures an efficient start to the queue management process.

3) Service Management: As counters complete service requests, they notify the system to assign the next customer, triggering real-time updates in the tracking system, as shown in Figure 8(a). Once the queue clears, the system displays a notification with the assigned counter for Customer A, enhancing clarity and user experience, as depicted in Figure 8(b). The system ensures smooth queue management, reducing wait times and improving customer satisfaction. Tables I and II provide a comparison of the proposed and existing systems from technical and economic perspectives.isting systems from technical and economic perspectives.

CONCLUSION

Most existing Queue Management Systems (QMS) are electronic-based, with our first approach using the cost-effective ATmega32 microcontroller and seven-segment displays, supporting up to 99 customers. While this system is affordable, it faces limitations such as the need for printed tickets and slower processing. To overcome these, we propose a micro-web-based QMS, offering live tracking and average waiting time prediction. This software-based solution enhances scalability, flexibility, and real-time updates, providing a more efficient and user-friendly alternative to traditional systems.



(a) (b)

Fig. 8: Tracking feature showing live updates over time

REFERENCES

[1] S. Zobair Ahmed et al., "Automated Queue Management System," *Ijaems Journal: A Survey on Patient Queue Management System*, vol. 16, 2016. [Online]. Available: http://creativecommons.J.ClerkMaxwell, ATreatiseonElectricityandMagnetism,3rded.,vol.2.Oxford:Clarendon, 1892,pp.6873.

[2] I. I. S. Branch, Indian Institute of Technology, Kharagpur, "Proceedings of the 2011 IEEE Students' Technology Symposium," 14-16 January, 2011, Indian Institute of Technology, Kharagpur, India. IEEE, 2011.

[3] B. Roy and A. Venugopal, "A Novel Intelligent System for Efficient Queue Management," 2013. [Online]. Available: www.ijareeie.com

[4] M. Mamun, A. H. M. Z. Karim, and M. I. Mahbub, "Low Power Microcontroller Based Intelligent Token Number Speaker and Display System," *IACSIT International Journal of Engineering and Technology*, vol. 3, no. 2, pp. 199–202, April 2011.

[5] A. A. Tahat, "Implementation of an SMS-based telemedicine system for patient electrocardiogram monitoring," in *Proceedings of the 4th IASTED International Conference on Telehealth and Assistive Technologies (AT'08)*, 2008, pp. 223–228.

[6] R. 1 Arun and P. P. Priyesh, "Smart Queue Management System Using GSM Technology," 2013. [Online]. Available: http://www.ripublication.com/aeee.html

[7] Gimba, U., Okoronkwo, C., Yusuf, M., Musa, A. & Ali, M. Queue monitoring system for bank. *Dutse Journal Of Pure And Applied Sciences (DUJOPAS)*. **6**, 269-276 (2020)

[8] Mallari, M., Guintu, J., Magalong, Y. & Yap, D. CLIQUE: A Web-Based Queue Management System with Real-Time Queue Tracking and Notification of Units for Angeles University Foundation Office of the University Registrar.