Smart Queue and Token Management System

MR S.P. Sandhya Rani[1], M. Kousheel[2], M. Srinivasa Chary[3], K. Deepak[4], S. Manoj Reddy[5]
[1,2,3,4,5]*Member, Department of Computer Science and Engineering (Cyber Security), Sphoorthy Engineering College, India*

***Abstract*—Queue management is essential in any service-based organizations such as hospitals, banks, government institutions, retail service sites, etc. However, the traditional system of queues results in longer waiting hours, congestion, and poor service management. To overcome such inefficiencies, this paper proposes a Smart Queue and Token Management System which aims to automate the queue management process using web applications. The system generates tokens containing digital number values for customers and manages the sequence of services. Customers can access the system at an administration page and get their tokens while the administrator/admin can login and control or access the incoming queued tokens. The serving token number is displayed on the digital display on service site allowing customer to track his number accordingly. This system has been implemented using web development tools such as HTML, CSS, JavaScript, and Python languages with Flask framework. The token management system implements faster service delivery, cuts down waiting hours, makes service more convenient, and improved customer satisfaction level.**

***Index Terms*—Digital queue, Service automation, Smart queue system, Token management, Web application.**

I. INTRODUCTION

Queue management is one of the concerns in service-oriented states such as hospitals, banks, government centers and retail service centers where large masses of people request for services at the same time. In usual queueing systems customers have to stand in the queue and wait until it is my turn at the counter. Sometimes this causes overcrowdings in the service centers and wastes waiting and service time. They also cause confusions. Manual task of queue handling is complicated.

With the popularity of digitals, automated queue management systems have become an efficient solution. In a Smart Queue and Token Management System, a number of features are used by the organization in order to handle the queue digitally by issuing tokens and keeping the service order automatically. The system generates tokens and saves data of the service order in a file. For the customers, a token printer is used in order to carry out this system using a server and a database.

The designed system is an organized way of handling customer queues efficiently. The system has an Admin End which can track the queue and call the next token and a Display Screen which can show the token in process. It reduces waiting time in the queue, maintains the service organized and increases the efficiency of the service centers.

II. SYSTEM ARCHITECTURE AND FLOW CHART

The Smart Queue & Token Management System is efficiently managed by the client–server-based design. It involves various components like User Interface, Token Generator, Queue Management Server, Database & Service Counters. It is implemented so that the user generates the tokens and keeps track of their queues from the web portal (UI). Later the backend server manages the queue operations & all the requests are handled using Python by considering Flask as the framework of implementation. The data of the individual tokens as well as the queue information is stored in the database (MySQL or SQLite). The overall system design facilitates linking the users, administrator, and the service counters together; and the working of the system starts when the users request for generating the token on the web portal and are assigned a queue position with a token number which is unique. The administrator calls their token sequentially and the user gets notified when their service time is near.

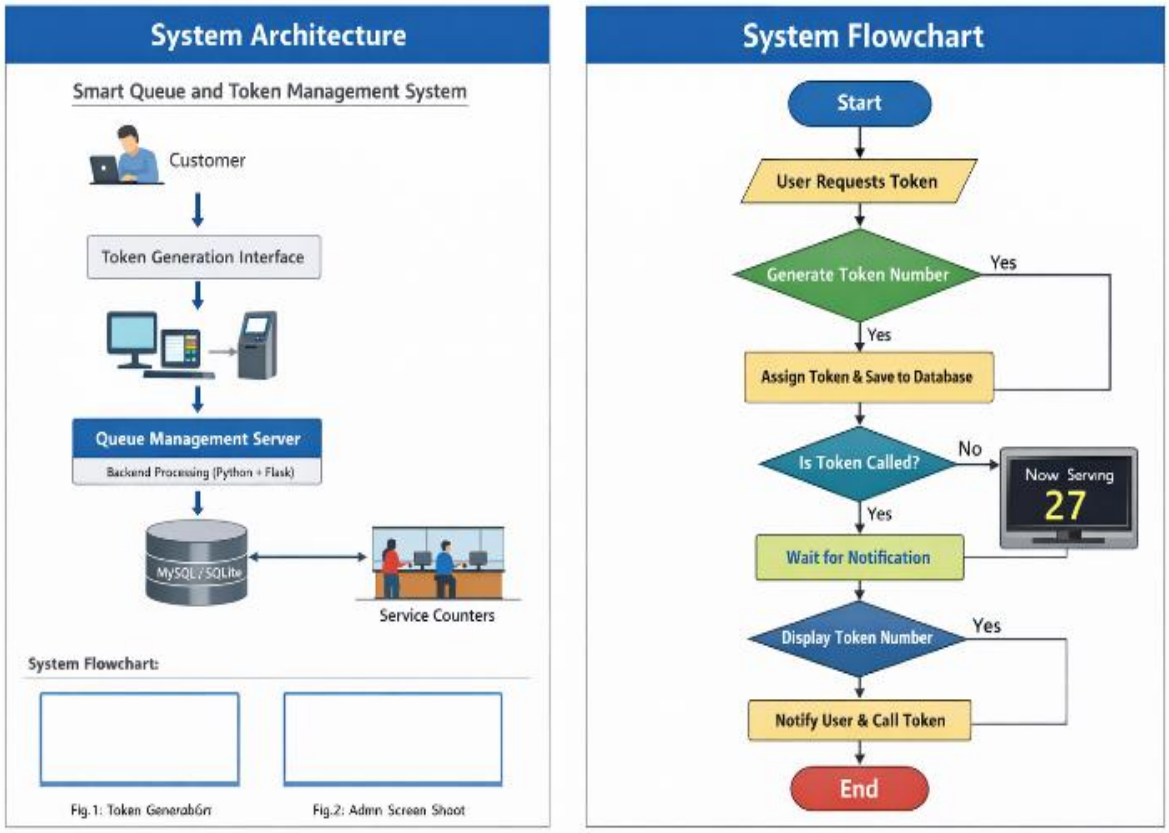

Fig. 1: System Architecture and Flowchart of Smart Queue and Token Management System

III. ALGORITHM FOR SMART QUEUE MANAGEMENT

Smart Queue and Token Management System utilize an algorithm that ensures efficient processing of tokens and queues. It generates unique tokens for each user and manages the order of service in an automated manner. Notifications are also sent to inform users about their queue position.

The algorithm is as follows:
Algorithm: Smart Queue Token Processing
Step 1: Start
Step 2: User opens the Smart Queue web application
Step 3: User logs into the system
Step 4: User selects service slot and requests token
Step 5: System generates a unique token number
Step 6: Store token in database
Step 7: Assign queue position
Step 8: Display queue status
Step 9: Admin calls next token
Step 10: Assign token to counter
Step 11: Send notification
Step 12: Display on screen
Step 13: Mark completed
Step 14: Update queue
Step 15: End

IV. SYSTEM DESIGN AND IMPLEMENTATION

A. System Home Page

The home page of the system acts as the central web page allowing access to the core features of the system. Users upon reaching the page can access either the user portal or administrator portal. It is the front end to the queue system.

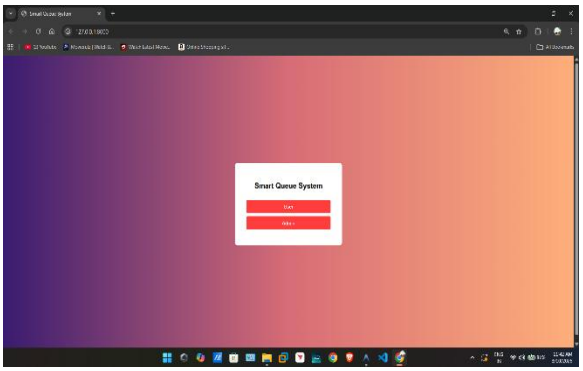

Fig. 2: Home Page of Smart Queue System

B. User Authentication

This module provides the safe authentication to the owner of the system. The owner can login with his or her own id and the owner can generate tokens successfully through this login process. Functionality like authentication of the users is performed through this module.

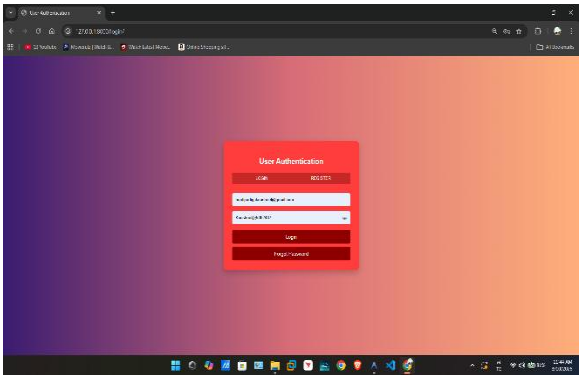

Fig. 3: User Login Interface

C. User Dashboard and Token Generation

Once logged in successfully, the user will be taken to the dashboard page where he can generate token for the service he wants. On the dashboard page, he can pick the available slots and generate token for it or cancel the token or re schedule his appointment if needed.

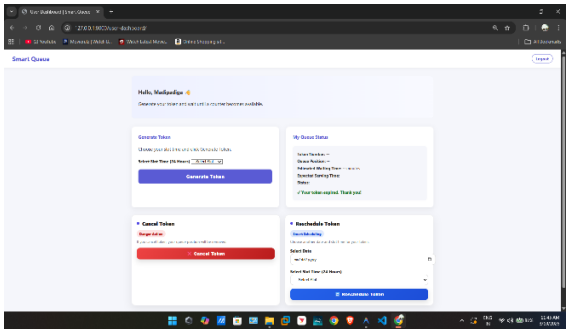

Fig. 4: User Dashboard for Token Generation

D. Queue Status Monitoring

Once logged in successfully, the user will be taken to the dashboard page where he can generate token for the service he wants. On the dashboard page, he can pick the available slots and generate token for it or cancel the token or re schedule his appointment if needed.

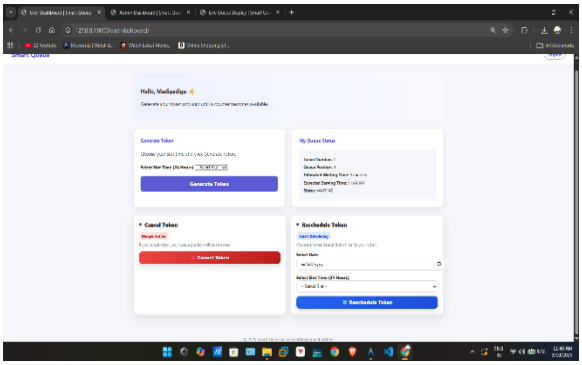

Fig. 5: User Queue Status and Token Details

E. Admin Authentication

Behind a special entry point, the admin steps into the system. Access happens solely through protected credentials, keeping others out. From there, oversight of queues begins - only those approved get this far. Each step ties back to one trusted doorway built for verification

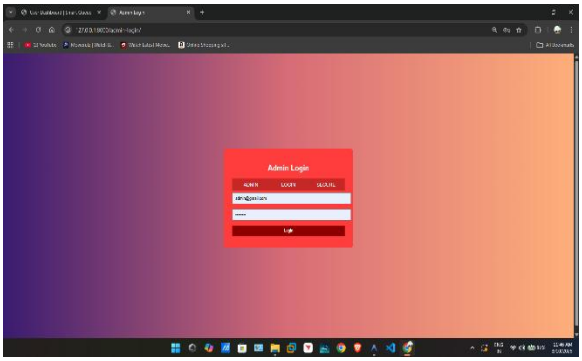

Fig. 6: Admin Authentication Interface

F. Admin Dashboard

From here, everything about the queue system comes together in one place. Watching current tokens happens right on screen, with updates appearing instantly. Next up, calling a new number works with just one click. Handling service points is part of the routine, adjusted as needed through simple tools. Insights into how well queues move show up clearly without clutter. Assigning people to specific desks fits naturally into daily tasks. Smooth movement across

lines stays possible because controls adapt quickly. Efficiency grows when each step links directly to real-time choices.

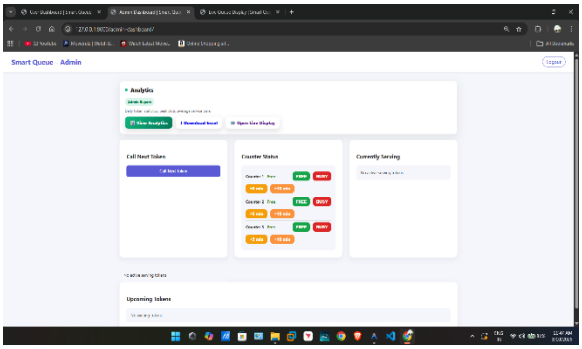

Fig. 7: Admin Dashboard

G. Token Serving Management

When someone gets a ticket, it instantly assigned to an available station. Following that step, employees may retrieve it again, hold it briefly, or let it go - based entirely on how events unfold. Then comes the next moment, shaped by choices made right then.

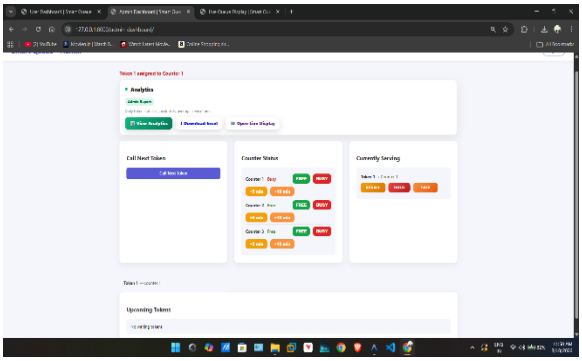

Fig. 8: Token Serving Status in Admin Panel

H. Live Queue Display System

A display mounted on the wall ticks forward every few seconds, revealing the current ticket number and the ones following. Without needing to ask, individuals in the room look up briefly to check their position. Waiting becomes less uncertain when the sequence shows itself so clearly.

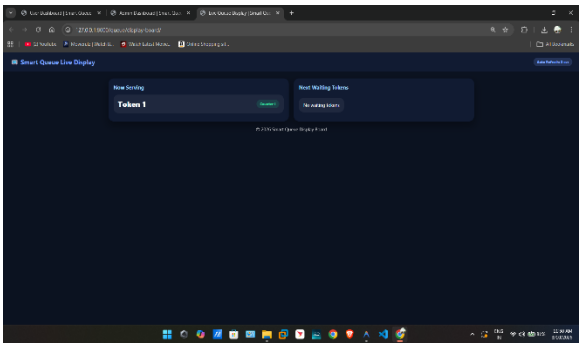

Fig. 9: Live Queue Display System

V. NOTIFICATION SYSTEM

The instant someone signs up, messages go out through email or text. Following registration, they find out where they stand in queue. Updates on progress come automatically - no request needed. Help underway brings a notification to reflect the start. Completion triggers one more message, confirming resolution.

A. Email Notification for Token Generation

Right after the token forms, a message arrives in the user's email. Inside it sits both queue details and the given number. Once creation finishes, emails send automatically no human step needed.

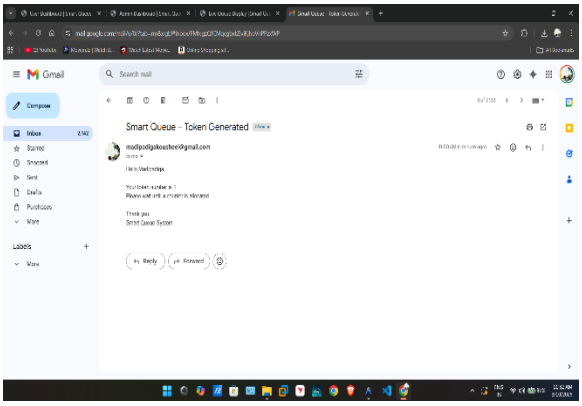

Fig. 10: Email Notification for Token Generation

B. Email Notification for Counter Allocation

When the user's token is about to be served, the system sends an email notifying the user about the assigned service counter.

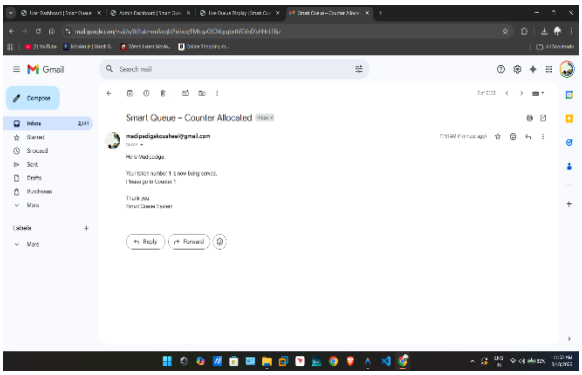

Fig. 11: Email Notification for Counter Allocation

C. Email Notification After Service Completion

Email Notification After Service Completion After the service wraps up, the system dispatches a concluding confirmation email. This message confirms the user's visit was completed.

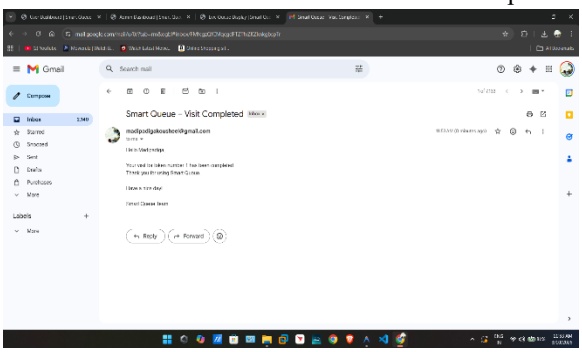

Fig. 12: Email Notification After Service Completion

D. SMS Notification Alerts

Beyond email, the system also dispatches SMS alerts. These keep users informed about significant queue developments, including impending services, counter assignments, and when their service is finished.

Fig. 13:SMS Notifications for Queue Updates

VI. RESULTS AND DISCUSSION

Now live: a fresh approach to managing lines, tested with real users joining as things happen. Instead of lining up early, people receive digital tokens online - alerts land directly in email or via message. Running the system, teams track progress through a dashboard made for fast choices and easier desk management. At each service spot, screens show clear status updates, removing uncertainty about position. Early runs confirm shorter waits, smoother support flow, cutting effort for those coming in.

VII. PERFORMANCE ANALYSIS

Testing looked at how fast the system responded, managed queues, handled notifications. Under load, many users created tokens at once inside a simulation. Performance checks focused on timing, flow, alert success rates. A virtual setup mimicked real-world usage patterns closely. Speed, accuracy, reliability shaped the evaluation framework used.

Every test confirmed seamless handling of queue tasks, along with instant updates reaching users and admins alike. Not just on time but always in step, notifications travel by email or SMS to keep people informed. Built smart, the structure supports fast data work while keeping each part talking clearly to the next.

Performance Testing Table

Parameter	Result
Average Token Generation Time	< 2 seconds
Queue Update Delay	< 1 second
Email Notification Delivery	5–10 seconds
SMS Notification Delivery	5–15 seconds
System Availability	99% uptime
Maximum Concurrent Users Tested	50

The experimental results demonstrate that the proposed system significantly improves queue management efficiency compared to traditional manual queue systems.

VIII. CONCLUSION

Now entering the scene: a fresh method for managing queues at service spots. As time passes, alerts signal

proximity to serving moment. Clarity stays high thanks to an uncluttered display interface. Beneath the surface, links glide without snag - making pauses fade. Lighter than paper logs, what shows up runs on quiet momentum. One clear route ease tension as wait times unwind. The last burden in rush moments.

Notifications go out not by manual trigger but according to changes in foot traffic. As individuals approach service zones, nearby displays shift accordingly. Should groups gather, adjustments occur ahead of major congestion. The entire arrangement responds in real time. When tasks follow a clear sequence, performance improves - no gaps, no redundancy. In high-traffic areas, smoother flow comes from anticipating delays before they form.

Finding out how things work shows shorter waits happen alongside smoother operations. Because the queue is easy to see, visitors understand where they stand without guessing. With employees tackling tasks one after another, mix-ups drop off noticeably.

IX. FUTURE WORK

A shift in perspective reveals another route - tomorrow's systems could bypass present constraints. Performance improves at this moment, though potential lingers past existing frameworks.

Eventually, capabilities might emerge similar to these examples

Future enhancements include:

- Development of a mobile application (Android and iOS)
- Integration of AI-based waiting time prediction
- QR code-based token generation
- IoT-based display systems

A step forward in capability could come from these updates, boosting what the Smart Queue and Token Management System can manage down the line. While minimal in appearance, the adjustments might extend reach across token handling processes. Growth in adoption brings heavier demand on queue efficiency - this much becomes clear over time. Each revision opens room for better response, keeping pace steady even under load. Small refinements here carry weight later, quietly steering how user traffic evolves through the system.

ACKNOWLEDGMENT

A stillness filled the lab, yet it turned into something vital - recognition goes to the Computer Science and Engineering team at Sphoorthy, whose efforts stretched past evening light. Owing to steady direction from them, scattered tries slowly found shape. Progress in each stage rested upon that backing; absent such help, movement forward could have stalled entirely. Thanks go to those nearby - peers, fellow students - their unspoken input quietly shaping every section. During moments when understanding faded, a brief remark often surfaced, softly redirecting thought. Those unplanned talks, arising without schedule, brought small changes across the long stretch of drafting. The result did not rise alone, rather through layered exchanges happening late, offbeat, unexpected.

REFERENCES

[1] I. Sommerville, Software Engineering, 10th ed., Boston, MA, USA: Pearson Education, 2016.
[2] A. S. Tanenbaum and D. J. Wetherall, Computer Networks, 5th ed., New York, NY, USA: Pearson, 2011.
[3] M. Fowler, Patterns of Enterprise Application Architecture, Addison-Wesley Professional, 2002.
[4] Python Software Foundation, "Python Documentation," Available: https://docs.python.org/3/
[5] R. Buyya, C. Vecchiola, and S. T. Selvi, Mastering Cloud Computing: Foundations and Applications Programming, Morgan Kaufmann, 2013.
[6] J. Nielsen, Usability Engineering, San Diego, CA, USA: Academic Press, 1994.
[7] S. K. Sharma and P. Gupta, "Digital Queue Management System for Customer Service Optimization," International Journal of Computer Applications, vol. 182, no. 12, pp. 25–30, 2018.
[8] A. Kumar and R. Singh, "Smart Queue Management System Using Web Technologies," International Journal of Advanced Research in Computer Science, vol. 9, no. 4, pp. 112–117, 2018.
[9] S. Patel and M. Shah, "Web Based Queue Management System for Service Centers,"

International Journal of Computer Science and Engineering, vol. 7, no. 2, pp. 45–50, 2019.

[10] R. Gupta and V. Sharma, "Automated Token Management System Using Web Technologies," IEEE International Conference on Smart Computing, 2020.