RESEARCH ARTICLE OPEN ACCESS

SMART QUEUE MANAGEMENT SYSTEM WITH AI-BASED TIME PREDICTION

Paul Xander Dela Cruz[1], Dwayne Lancelot Fuerte[2], Emerson Daguio Jr[3], Ms. Vivien Agustin[4], Dr. Ronald Fernandez[5]

*(Department of Information Technology, Jesus Reigns Christian College, 811 J. Nakpil St., Malate, Manila
[1]paulxanderdelacruz@jrccmanila.edu.ph)

**(Department of Information Technology, Jesus Reigns Christian College, 811 J. Nakpil St., Malate, Manila
[2]dwaynelancelotfuerte@jrccmanila.edu.ph)

***(Department of Information Technology, Jesus Reigns Christian College, 811 J. Nakpil St., Malate, Manila
[3]emersondaguiojr@jrccmanila.edu.ph)

****(Department of Information Technology, La Consolasion University Philippines, Manila
[4]vivien.agustin@email.lcup.edu.ph)

*****(Department of Information Technology, La Consolasion University Philippines, Manila
[5]ronald.fernandez@email.lcup.edu.ph)

--************************--

Abstract:

Managing customer flow is a cornerstone of operational efficiency in the service industry, yet traditional "take-a-number" methods often result in human error and increased customer stress due to a lack of real-time status updates. While Digital Queue Management (DQM) systems have automated registration, they remain limited by their inability to accurately predict waiting times. This study addresses these critical inefficiencies by proposing a Smart Queue Management System (SQMS) that integrates an Artificial Intelligence (AI)-based time prediction model with multi-platform accessibility. Developed using an Agile Software Development Life Cycle (SDLC) and a client-server architecture, the system synthesizes an on-site kiosk, a mobile-first web portal, and public digital signage into a unified ecosystem. The analytical core of the system utilizes machine learning algorithms, specifically Random Forest and Gradient Boosting, which are trained on both historical arrival patterns and real-time operational data to generate dynamic Estimated Waiting Times (EWT). Results demonstrate that the SQMS provides real-time synchronization across all platforms, allowing users to transition to virtual monitoring via QR codes, thereby reducing physical congestion and waiting anxiety. Furthermore, the system includes an administrative dashboard that offers data-driven insights into staff efficiency and service bottlenecks. Ultimately, the SQMS transforms traditional waiting environments into streamlined digital ecosystems, enhancing both administrative productivity and the overall quality of the customer experience.

Keywords — Smart Queue Management, Artificial Intelligence, Machine Learning, Time Prediction, Agile Methodology, Human-Computer Interaction

--************************--

I. INTRODUCTION

In High-volume service environments such as government offices, academic institutions, and financial facilities, long queues remain a critical operational bottleneck. Traditional physical queue management models require individuals to wait in static locations without accurate insights into their anticipated wait times. This structural limitation leads to reduced customer satisfaction, increased perceived waiting stress, and compromised administrative efficiency.

While digital queue management systems have introduced structured ticketing they rely on rigid first-come, first-served logic that fails to adapt to real-time service disruptions, varying transaction complexities, or staff availability to overcome these challenges, this study presents a smart queue management system that integrates artificial intelligence and machine learning models to dynamically forecast estimated waiting times. By transitioning from a static queuing framework to an intelligent, predictive ecosystem, this system aims to optimize operational workflows, empower users with transparent time forecasting, and minimize physical congestion in waiting areas.

II. STATEMENT OF THE PROBLEM

Current Digital Queue Management (DQM) systems are simple digital ticket dispensers, only automating registration. This lack of accurate, real-time, and predictive AI leads to critical inefficiencies: prolonged, uncertain client waits, low satisfaction, disrupted schedules, and poor transparency due to unreliable estimated service times.

Specifically, this study seeks to address these multifaceted challenges:

1. Individuals or end-users experience extended and uncertain waiting periods due to the absence of accurate time estimation in traditional queue systems. This unpredictability leads to frustration, inefficiency, and poor customer satisfaction.
2. Due to the lack of proper scheduling, time prediction, and notification mechanisms, users tend to arrive simultaneously, resulting in overcrowded waiting areas, increased discomfort during peak hours, and the need to wait physically in line without flexibility in managing their time.
3. Most queue management systems lack AI capabilities to analyze historical and real-time data. Without predictive insights, organizations cannot optimize operations, forecast peak hours, or improve service efficiency.

III. OBJECTIVES OF THE STUDY

The main objective of this study is to design, develop, and evaluate a Smart Queue Management System (SQMS) that integrates an Artificial Intelligence (AI)-based time prediction model with multi-platform accessibility to improve operational efficiency and enhance user experience. This study specifically aims to:

I. To develop, implement, and rigorously validate a machine learning model, specifically an AI model, designed to accurately predict Estimated Waiting Times (EWT) for customers by analyzing and processing real-time, dynamic queue and service operational data.
II. To implement an automated queue management system that integrates real-time scheduling, predictive wait-time analytics, and an Omni channel notification framework to eliminate physical congestion and provide users with time-management flexibility.
III. To integrate a machine learning-based analytical layer into the queue management framework that leverages historical and real-time data to provide predictive insights, enabling proactive staff allocation and the optimization of service efficiency.

IV. REVIEW OF RELATED LITERATURE

This chapter reviews literature and studies that form the theoretical and empirical basis for the proposed Smart Queue Management System (SQMS) with AI-Based Time Prediction. It covers queue management theories, technological evolution, state-of-the-art AI for waiting time prediction, multi-platform integration, Human-Computer Interaction (HCI), and data-driven decision-making. The synthesis identifies a research gap: the lack of an integrated, multi-platform system that combines predictive AI with comprehensive administrative analytics, which this study addresses.

Queuing Theory and Service Models

Queuing Theory is proposed by Agner Krarup Erlang, a Danish mathematician. It is the mathematical study of waiting lines. Queuing theory examines the mathematical principles behind waiting lines and how they behave in different service settings. Foundational models such as the M/M/1 and M/M/c queues describe the relationship between arrival rates, service rates, and queue lengths, providing a framework for analyzing system efficiency (Shortle et al., 2018). Another key concept, Little's Law, connects the average number of customers in a system with their arrival rate and the average time they spend being served.

While these classical models form the foundation of modern queue management theory, they often depend on simplified assumptions, such as Poisson arrivals and exponential service times that may not hold true in real-world environments. In practice, factors like fluctuating demand, unpredictable service durations, and human-driven processes can make queue behavior far more complex.

Digital Queue Management Systems (DQM)

The development of queue management systems has evolved significantly from manual to digital processes over the past few decades. Early systems primarily sought to automate the traditional "take-a-number" method using electronic dispensers and LED displays to organize customer flow. These systems helped maintain fairness and order in serving customers but did little to reduce uncertainty about waiting times or improve the overall user experience.
As technology advanced, Digital Queue Management (DQM) systems began integrating more interactive features, such as self-service kiosks, SMS notifications, and basic reporting tools, allowing users to register, monitor, and receive updates more efficiently (Uddin et al., 2016). These innovations marked an important step toward automating service operations and enhancing transparency in queue handling.

Recent studies have introduced the concept of Virtual Queue Management Systems (VQMS), which extend the capabilities of DQM by enabling remote check-ins and real-time tracking through mobile or web-based platforms (Nguai et al., 2018). These systems not only minimize physical congestion but also align with modern expectations for convenience and accessibility. Together, these developments form the foundation upon which more intelligent, AI-driven queue management systems are being designed today, systems capable of predicting waiting times, analyzing performance data, and improving service delivery dynamically.

Artificial Intelligence in Time-Series Prediction

Predicting waiting time can be viewed as a time-series forecasting problem, where patterns from past data are used to estimate future outcomes. Machine learning provides a wide range of algorithms suited for this type of task, offering more adaptive and data-driven solutions compared to traditional statistical approaches.

Basic models such as Linear Regression often serve as the starting point for estimating waiting times, providing a simple yet interpretable baseline. However, more advanced algorithms like Random Forests and Gradient Boosting can identify nonlinear relationships between variables, for instance, how factors such as the time of day or the number of active service counters influence queue durations (Lim & Zohren, 2021).

Recent studies have extended these approaches through the application of artificial intelligence (AI) and hybrid modeling techniques. These models integrate multiple forecasting methods to enhance accuracy and adaptability in dynamic environments (Fathian et al., 2019). Neural network architectures, such as Recurrent Neural Networks (RNNs), have gained attention for their strength in handling sequential and time-dependent data (Konar & Bhattacharya, 2017). Although these models require substantial data and computational resources, they are highly effective in capturing complex temporal patterns, making them valuable tools for predicting waiting times in queue management systems (Kose, 2019).

Multi-Platform System Integration and Human-Computer Interaction (HCI)

The concept of ubiquitous computing centers on the idea that technology should be available and accessible across multiple platforms to support users in different contexts. This approach ensures that computing is not confined to a single device but is integrated into everyday environments. As discussed by Zou (2025), cross-platform integration plays a vital role in enabling seamless communication between systems, allowing information to move fluidly between devices.

In the field of system development, Huang, Sadeghian, and Kwon (2015) emphasize the importance of integrating multi-platform frameworks to achieve distributed yet synchronized operations. Their study highlights how technological systems can function more efficiently when communication across platforms, such as kiosks, web applications, and mobile devices, is unified through a centralized framework.

From a design perspective, Human–Computer Interaction (HCI) principles are crucial in shaping how users interact with these systems. According to MacKenzie (2024), effective HCI design focuses on usability, accessibility, and reducing cognitive effort for users. Similarly, Card (2018) underscores the psychological aspect of interface design, noting that systems should be intuitive and consistent to minimize user confusion and enhance overall experience.

In queue management, these concepts converge to create a cross-platform experience that connects physical and digital interfaces. A practical example of this is the integration of QR codes, which allow users to transition effortlessly from a kiosk to their personal devices. This simple mechanism bridges the gap between in-person and online interaction, promoting accessibility, convenience, and user satisfaction.

Information Visualization: Principles for Real-Time Digital Signage

Digital signage plays an important role in enhancing communication and user experience within waiting areas. It acts as a public information radiator, allowing real-time updates and visual cues to be displayed clearly to everyone present. According to Garaus and Wagner (2019), effective digital signage can significantly improve customer satisfaction by reducing perceived waiting time and increasing engagement, especially in service environments such as retail and administrative offices

When it comes to designing visual displays, clarity and usability are essential. The design principles established by Edward Tufte, as discussed by Bacon and IDMIL (2015), emphasize the importance of data-ink ratio, presenting data in a way that maximizes clarity and minimizes unnecessary decorative elements often referred to as “chart junk.” Applying these principles to queue management displays ensures that information is communicated efficiently and without distraction.

For queue environments, this means using large, high-contrast fonts, clear information groupings (such as “Now Serving” or “Next Up”), and a minimalist layout that allows users to easily glance at and understand the information being shown. When thoughtfully designed, digital signage not only keeps users informed but also enhances their overall waiting experience by fostering transparency and reducing uncertainty.

Data-Driven Decision Making

In modern service operations, data-driven decision-making has become essential for ensuring efficiency and responsiveness. According to Diván (2017), leveraging data allows organizations to make informed choices rather than relying solely on intuition or static processes. This shift toward evidence-based management helps improve operational accuracy and overall performance.

Within this context, Yadava (2023) highlights the growing role of data analytics in enhancing business intelligence and strategic planning. By systematically collecting and analyzing operational data, organizations can uncover trends, assess performance, and identify areas for improvement. Similarly, Michael et al. (2024) emphasize that the integration of artificial intelligence and data science strengthens data-driven systems by providing real-time insights that can guide managerial decisions effectively.

A key tool that embodies these principles is the administrative dashboard. It is a visual interface that consolidates important metrics into an accessible, real-time overview. As described by Bartlett and Tkacz (2017), dashboards serve as a form of governance by providing transparency, accountability, and structured feedback to decision-makers. Franklin et al. (2017) further explain that dashboards using interactive visualizations, such as charts and graphs, enable managers to monitor throughput, spot trends, and make quick, evidence-based decisions.

In the case of queue management systems, administrative dashboards display essential Key Performance Indicators (KPIs) such as average waiting time, service time, staff utilization, and customer abandonment rate. Presenting these metrics visually allows managers to identify operational bottlenecks, evaluate staff performance, and optimize workflows in real time. Through this integration of data analytics and visualization, service providers can continuously improve their processes and deliver more efficient, customer-centered service.

IV. METHODOLOGY

This chapter outlines the research methodology employed in the design, development, and evaluation of the Smart Queue Management System (SQMS) with AI-Based Time Prediction. It details the two-phased Developmental Quantitative research design adopted for the study, which combines the creation of a technological artifact with the empirical assessment of its impact. Specifically, this chapter presents the systemic approach used for system development, including the underlying architecture, the specifications for hardware and software requirements, and the methods used for the creation and validation of the AI prediction model. Furthermore, it describes the data acquisition strategy, the selection of research participants, the procedures for data analysis, and the evaluation framework used to measure the system's efficacy, efficiency, and user satisfaction, ensuring the research objectives are addressed through an intensive and systematic process.

Research Design

The study adopted a Developmental–Quantitative research design, which was deemed appropriate given its dual focus on the creation and empirical evaluation of a technological solution. As discussed by Clark-Plaskie et al. (2022), developmental research designs were particularly suited for studies that aimed to design, construct, and assess innovative systems that address real-world problems. In this case, the SQMS was developed as a functional artifact to optimize queue efficiency and enhance user experience through AI-driven predictions and multi-platform accessibility.

The Agile methodology was chosen for the Smart Queuing System (SQMS) because it allows for continuous improvement through short, iterative cycles called Sprints. Instead of building the entire system at once, the development is broken down into manageable stages such as setting up the data pipeline, training the AI model, and designing the user interface. This approach is ideal for an AI-based project, as it allows the team to test the accuracy of wait-time predictions in real-time and make immediate adjustments if the model underperforms.

By using Agile, the system stays flexible and user-focused. Regular testing and feedback ensure that the final product is not just technically sound, but also easy for both staff and customers to use. This iterative process reduces the risk of errors and ensures that the AI remains accurate even as service conditions change during deployment.

The research design unfolded in two major phases:

Development Phase - the design and construction of the SQMS prototype using the Agile software development framework, emphasizing iterative refinement and continuous testing.

Evaluation Phase - the quantitative measurement of the system's performance, prediction accuracy, and user satisfaction through statistical analyses and standardized usability testing.

This dual structure ensured that both the technological output and the empirical findings contributed meaningfully to advancing queue management practices.

Developmental Research Framework

The developmental framework followed the Agile Software Development Life Cycle (SDLC), which allowed flexibility and adaptability throughout the creation of the SQMS. The framework consisted of five iterative stages:

Planning

The planning phase serves as the foundation of the Smart Queuing System. In this stage, the development team collaborates with stakeholders to identify common challenges in traditional queuing processes, such as long waiting times, overcrowding, and lack of real-time updates. Instead of attempting to define all requirements at once, the agile approach breaks them down into smaller, user-centered tasks or user stories.

Clear system objectives are established, including improving service efficiency and enhancing user experience. A flexible development timeline is also created, typically organized into short iterations or sprints. This allows the team to adapt to feedback and changing requirements throughout the project.

Design

In the design phase, the team concentrates on organizing how the system will work and appears to users during the design phase. This involves creating the overall system architecture to guarantee smooth communication between various parts, including administrative dashboards, mobile apps, and kiosks.

User interface designs are carefully developed to be easy to use, intuitive, and suitable for a variety of users. Prototypes and initial diagrams are made, and they are regularly improved based on input from users. In addition, data flow diagrams are created to show how queue data is updated and processed in real time, guaranteeing dependability and efficiency.

Development

System features are developed and provided incrementally during the development phase, which is a cycle. Each sprint focuses on completing specific modules of the Smart Queuing System. For example, one round might concentrate on creating the mobile or web application for remote queue access, while another might design the kiosk interface for on-site users.

The kiosk interface, web and mobile platforms, digital signs for queue updates, and administrative dashboards for monitoring and control are important parts of the system. In order to guarantee that newly created features integrate seamlessly with already-existing components and enable the system to develop steadily and effectively, continuous integration is done.

Testing

This phase encompasses functional testing, throughout the development phase, testing is done frequently to ensure the dependability and quality of the system. While integration testing makes sure that every component of the system functions as intended, functional testing verifies that every feature performs as intended.

To find usability problems and opportunities for improvement, user acceptance testing is also done to get input from real users. In order to make the system more reliable and user-friendly over time, any mistakes or issues found during testing are fixed in later iterations.

Evaluation

During the evaluation phase, the Smart Queuing System's overall performance and effectiveness are evaluated. This requires looking at important variables including user satisfaction, system responsiveness, and waiting time reduction.

To determine whether the system achieves its intended goals, user and administrator feedback is carefully reviewed. According to Agile principles, the evaluation does not signal the end of development but rather offers insights for ongoing enhancement, ensuring that the system continues to be effective, flexible, and user-friendly.

Each phase informed the next, allowing continuous improvement based on user feedback and performance testing. The iterative nature of Agile ensured that the prototype evolved in direct response to observed needs and performance gaps (Kramer, 2019). Verification and validation principles were applied throughout the process to maintain software reliability and consistency (Dabney & Arthur, 2019)

A. SYSTEM ARCHITECTURE

The Smart Queue Management System was developed using a client–server architecture, which ensured centralized control and real-time synchronization across all platforms.

The system comprised four core modules:

Kiosk Ticketing Module - Deployed on-site, this module allowed users to register for queue tickets. It communicated with the backend server to generate a queue number and an initial Estimated Waiting Time (EWT), printed along with a scannable QR code.
Web/Mobile Module - Provided a responsive interface for real-time queue tracking and notifications. Users could access their queue position through the QR code or a unique URL.
Digital Signage Module - Displayed “Now Serving” and “Next Up” information to the public, reducing perceived waiting times through continuous visual updates.
Administrative Module - Enabled service staff to manage queues, monitor ongoing transactions, and view performance analytics.

This modular structure ensured scalability, allowing new service counters or features to be integrated seamlessly as organizational needs expanded.

C. METHODS AND TOOLS

The research methodology follows an Agile Prototyping model, focusing on iterative cycles of design, coding, and testing. This approach ensures the Flask backend and React frontend integrate smoothly with the Scikit-learn AI model. By using this method, the system is continuously refined to ensure the hardware can handle multiple user requests with minimal delay.

To support development, GitHub is used for version control and Jupyter Notebooks for training the prediction models. Postman is utilized to test the communication between devices, while Docker ensures the software runs consistently across the server and kiosks. Together, these tools provide a stable framework for delivering real-time queue updates and analytics.

SOFTWARE REQUIREMENTS

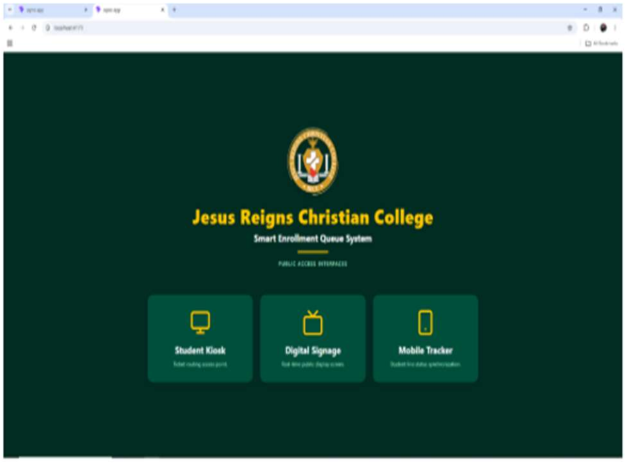
The Software Requirements focus on a database to store records and Python to run the AI prediction models. A web or mobile framework provides live updates to users via an API, while security protocols ensure all data remains protected and the system operates smoothly.

Layer	**Technology**	**Description**
Front-End	React.js, Vue.js	JavaScript frameworks used to build interactive and dynamic user interfaces for web applications.
Back-End	Python (Flask/Django)	Server-side frameworks that handle business logic, API creation, and communication between front-end and database.
Database	PostgreSQL / MySQL	Relational database management systems used to store, manage, and retrieve structured data efficiently.
AI Framework	Scikit-learn	Machine learning library in Python used for data analysis, modeling, and predictive algorithms.
Visualization	Chart.js, D3.js	Libraries for creating visual representations of data such as charts, graphs, and interactive dashboards.
Testing	Jest, Selenium, Postman	Tools for ensuring code quality: Jest for unit testing, Selenium for browser automation, and Postman for API testing.

V. RESULTS AND DISCUSSION

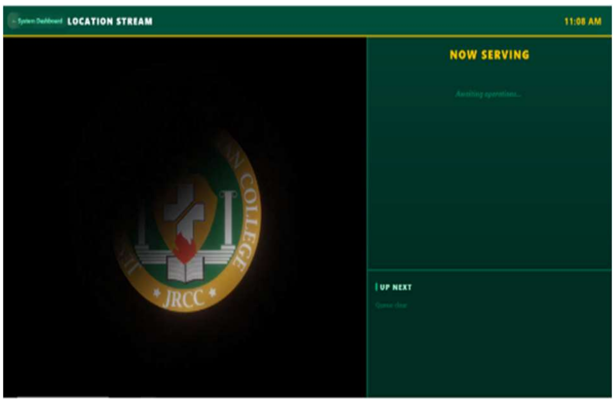
A. DIGITAL SIGNAGE SCREEN

The system consists of a Mobile Tracker that lets students use their mobile devices to track their queue status, a Digital Signage screen that shows real-time queue updates and



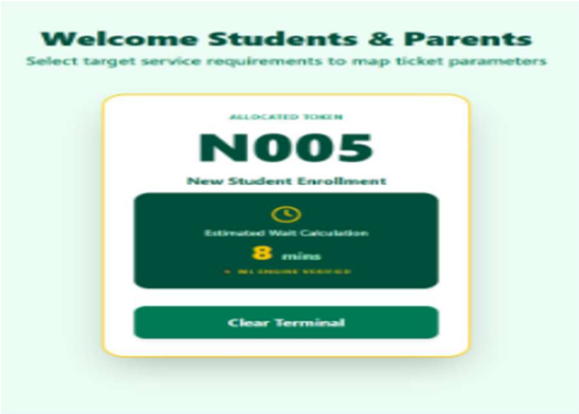
announcements, and a Student Kiosk for obtaining and routing queue tickets. The interface's overall goal is to organize, speed up, and simplify the enrollment process by effectively managing student transactions through digital technology.

B. QUEUE NUMBER



While students and parents wait for their turn, it shows queue numbers, announcements, and real-time updates. This facilitates better organization of the enrollment process and makes it simple for users to keep track of the current queue status and ticket numbers.

C. DIGITAL QUE TICKET



A digital queue ticket screen for a student enrollment system is shown in this picture.
You are given a unique token number by the screen. Your token number in this instance is N005.
Your service needs are instantly categorized by the system. Your service path is displayed as New Student Enrollment.
Your waiting time is determined by an embedded intelligence engine. Your wait time is clearly estimated to be eight minutes on the screen.
The waiting process is made easier by the system. Without physically forming a line, you keep track of your turn. You can reset the screen for the next user by pressing the clear terminal button.

DISCUSSION

You need a queue management system to eliminate long wait times and optimize your operations.
Predictive data confirms machine learning fixes queue delays. Standard systems fail because human processing times change during peak hours. Workers slow down as fatigue sets in. This system tracks those changes.
The machine learning model captures timing patterns. It tracks clerk performance and arrival surges. This transparency lowers customer frustration. Customers use their waiting time outside the office lobby.
Managers view staff performance metrics on live dashboards. They reallocate staff to busy service counters immediately. High traffic volume increases computer processing loads. The system schedules data training during night hours to protect day performance.

VI. CONCLUSION

You need an efficient way to manage visitor flow and reduce lobby congestion.
This project delivers a queue system with live time forecasting. The software connects a Python backend to a user interface. This combination cuts waiting room crowding. It gives users exact appointment windows. Future versions will run on cloud servers and sort customer requests automatically.

VII. ACKNOWLEDGEMENT

The authors thank the IT department at Jesus Reigns Christian College Manila for providing technical guidance and research facilities.

VIII. REFERENCES

1. Ahmad, T., et al. (2023). Investigating the use of Long Short-Term Memory (LSTM) networks for predicting wait times in high-traffic retail environments. Journal of Retail Analytics.
2. Ahmadzadeh, B., Patey, C., Norman, P., Farrell, A., Knight, J., Czarnuch, S., & Asghari, S. (2025). Artificial intelligence solutions to improve emergency department wait times: Living systematic review. The Journal of Emergency Medicine, 75, 174.
3. Ahmadzadeh, B., et al. (2024). AI technologies research protocol for real-time decision-making and emergency department wait time reductions. Clinical Workflow Protocols.
4. Al-Mazaideh, M., & Al-Sabbagh, A. (2026). Queue time predictor (QTP): A hybrid forecasting strategy for real-time queue monitoring. Journal of Software Systems and Platforms.
5. Alvarado-Valencia, J. A., et al. (2017). Detrimental outcomes of queue inefficiencies in service operational frameworks. International Journal of Service Operations Management.
6. Arnold, J. (2022). Consumer psychological perceptions of delay and service attentiveness in modern industries. Journal of Service Industry Research.
7. Bacon, J., & IDMIL. (2015). Applying Edward Tufte's data-ink ratio principles to contemporary visual systems. Information Visualization Review.
8. Badawi, A., & Anjasmara, R. (2024). Web-based queuing system design for service operations using website technology. Journal of Digital Operations.
9. Bartlett, L., & Tkacz, N. (2017). Dashboards as metrics of governance, transparency, and structured feedback. Operational Management Quarterly.
10. Brakewood, C., Macfarlane, G. S., & Watkins, K. E. (2015). The impact of real-time information displays on bus ridership in New York City. Transportation Research Part C: Emerging Technologies.
11. Card, S. K. (2018). Psychological aspects of interface consistency and intuitive system design. Human-Computer Interaction Systems.
12. Chen, L., & Martinez, M. (2024). Machine learning and IoT-based systems for hospital queue wait time

estimations using edge processing. Internet of Things and Smart Applications.

13. Clark-Plaskie, M., et al. (2022). Methodological frameworks for developmental research designs in artifact construction. Educational Technology Research Journal.

14. Dabney, J., & Arthur, L. (2019). Software verification and validation principles for high-reliability systems. Software Engineering Insights.

15. Diván, M. J. (2017). Evidence-based management and data-driven choices in service infrastructure optimization. Journal of Business Intelligence.

16. Ekşioğlu, O. (2016). User experience design and accessibility evaluation of public information kiosk interfaces. International Journal of Human-Computer Interaction Studies.

17. Fathian, M., et al. (2019). Hybrid predictive modeling and multi-forecasting integrations in dynamic queue environments. Applied Intelligence Review.

18. Franklin, D., et al. (2017). Interactive visualization, charts, and graphs for real-time management dashboards. Journal of Visual Analytics.

19. Garaus, M., & Wagner, U. (2019). Let us entertain you: The influence of digital signage on perceived waiting time and customer satisfaction. Journal of Retailing and Consumer Services.

20. Garcia-Sosa, R., et al. (2022). Natural language processing and predictive triage structures in multi-specialty clinical environments. Journal of Medical Systems and Data.

21. Harada, Y., & Shimizu, T. (2020). Impact of a commercial artificial intelligence-driven self-assessment triage tool on clinical wait times. International Medicine Retrospective Journal.

22. Huang, S., Sadeghian, P., & Kwon, S. (2015). Integrating multi-platform frameworks for distributed yet synchronized operations. Distributed Computing Systems.

23. Konar, A., & Bhattacharya, S. (2017). Recurrent neural networks (RNN) for sequential and time-dependent series forecasting. Neural Frameworks.

24. Kose, U. (2019). Advanced temporal deep learning for time-series queue data analysis. Artificial Intelligence Applications.

25. Kramer, M. (2019). Iterati

