Smart Queue Analyzer for Hospital

Mrs. Azmeen Shaikh[1], Unnati Pilore[2], Shreya Sanap[3] , Anushka Pawar[4], Vedanshi Patil[5]
[1]*Lecturer, Department of Information Technology, K. K. Wagh Polytechnic, Nashik.*
[2,3,4,5]*Diploma Student, Department of Information Technology, K. K. Wagh Polytechnic, Nashik.*

---***---

Abstract - *Hospitals often faced overcrowding and long waiting lines, which cause inconvenience to patients and create difficulties for hospital staff in managing service flow efficiently. The Smart Queue Analyzer is proposed as an automated solution to monitor and manage patient queues using Internet of Things Technology and embedded systems. The system uses sensors to detect and count the number of patients waiting in a queue, while a microcontroller processes the data in real time. The current queue status is displayed on a digital dashboard, allowing patients and hospital staff to stay inform about waiting conditions. When the number of patients exceeds a predefined limit the system generates alerts so that hospital staff can take immediate action, such as opening additional service counters or redistributing workload. The system also stores queue data for future analysis, helping hospital management identify peak hours and improve staff scheduling and resource allocation. By reducing waiting time and improving patient flow, the proposed system enhances service efficiency and patient satisfaction.*

Key Words: Smart Queue Analyzer, Hospital Queue Management, Internet of Things, Embedded Systems, Real Time Monitoring, Microcontroller, Patient Flow Management, Service Efficiency.

1. INTRODUCTION

In modern healthcare environments, efficient patient flow management has become a critical operational challenge. Hospitals frequently experience congestion in outpatient departments (OPDs), registration counters, pharmacies, laboratories, and billing sections due to increasing patient inflow and limited medical staff. Prolonged waiting times not only cause stress and discomfort to patients and their families but also impact treatment efficiency, staff productivity, and overall hospital performance. Traditional queue management practices, such as manual supervision or token-based systems, often lack real-time monitoring capabilities and fail to provide accurate data for effective decision-making.

With the advancement of the Internet of Things (IoT) and embedded technologies, intelligent healthcare management solutions can now be implemented. IoT-based sensors can monitor patient queue density in different hospital departments, while microcontrollers process the collected data in real time. The analyzed information can be displayed on digital dashboards for hospital administrators, enabling better coordination, optimized staff allocation, and improved service delivery.

The proposed Smart Queue Analyzer for hospitals is an automated system designed to monitor and analyze patient queues efficiently. It utilizes sensors to estimate the number of patients waiting in specific areas, processes the data through a microcontroller, and provides real-time updates through a digital interface. The system can also generate alerts when queue length exceeds predefined limits, allowing hospital staff to take immediate corrective actions.

By integrating real-time monitoring, automated notifications, and data analytics, the Smart Queue Analyzer enhances patient experience, reduces waiting time, and supports data-driven hospital management. The system offers a scalable and cost-effective solution to improve operational efficiency and service quality in healthcare institutions.

2. LITERATURE SURVEY

Several studies have explored IoT-based queue monitoring systems in healthcare environments. Sensor-based approaches using ultrasonic and infrared (IR) sensors have been implemented to detect and count individuals waiting in hospital departments such as outpatient registration, pharmacies, and billing counters. These systems demonstrated that automated queue detection can reduce manual supervision and improve monitoring accuracy. However, many of these implementations focused primarily on counting mechanisms without integrating advanced analytical features for performance optimization.

Embedded platforms such as Arduino and Raspberry Pi have been widely used for real-time data acquisition and processing in smart queue systems. These platforms enable immediate display of queue length on digital screens and centralized dashboards. While real-time visualization improves transparency, several existing systems lack intelligent alert mechanisms to notify hospital staff when congestion exceeds predefined limits.

Cloud-based healthcare monitoring frameworks have also been introduced to store queue data and generate statistical reports. Dashboard interfaces allow administrators to observe patient inflow trends and evaluate department-wise load distribution. Despite these advantages, some solutions require complex infrastructure and high deployment costs, limiting their scalability in small and medium-sized hospitals.

Research in predictive analytics has highlighted the importance of forecasting peak hours using historical queue data. Machine learning techniques have been applied to predict crowd density and support staff allocation planning. However, many predictive models operate on stored datasets and do not incorporate live sensor inputs for dynamic real-time adjustments.

Additionally, automated notification systems using SMS or mobile applications have been developed to reduce waiting time by informing staff about congestion. While effective in alert generation, these systems often operate independently without integration into a unified monitoring and analytics platform.

3. PROBLEM STATEMENT AND RESEARCH GAP

In hospital environments, managing patient queues efficiently remains a major operational challenge. Departments such as outpatient registration, consultation units, laboratories, pharmacies, and billing counters frequently experience congestion due to high patient inflow and limited medical staff. Long waiting times can cause discomfort, anxiety, and dissatisfaction among patients while also affecting treatment efficiency and staff productivity.

Traditional queue management methods, including manual supervision and token-based systems, are time-consuming, prone to human error, and unable to provide real-time updates. Although some hospitals use CCTV monitoring or basic ticketing systems, these approaches do not offer instant analytics or automated alerts, making it difficult for administrators to respond quickly to overcrowding situations. As a result, patient flow becomes inefficient, leading to operational delays and reduced quality of healthcare services.

Existing queue management solutions in hospitals primarily focus on monitoring rather than intelligent management. Many systems lack real-time analytics to evaluate queue conditions dynamically. Additionally, the absence of automated alert mechanisms prevents staff from responding promptly when congestion reaches critical levels. Current implementations also fail to provide integrated dashboards and data-driven insights for improving resource allocation and patient flow planning.

Therefore, there is a need for an intelligent, automated, and cost-effective system that can monitor patient queues in real time, generate alerts during overcrowding, and provide actionable analytics. The proposed **Smart Queue Analyzer for Hospital** addresses these limitations by integrating sensor-based monitoring, real-time data processing, automated notifications, and dashboard visualization to enhance hospital efficiency and patient experience.

4. PROPOSED SYSTEM ARCHITECTURE

The proposed Smart Queue Analyzer for Hospital is designed as a modular, real-time queue monitoring and analytics framework that integrates IoT sensors, embedded processing, and dashboard-based visualization. The architecture follows a client–server model consisting of sensor nodes, a microcontroller-based processing unit, and a web-based monitoring dashboard.
The primary objective of the architecture is to ensure scalability, accuracy, and low-latency monitoring while maintaining a clear separation between data collection, processing, and user interface layers.

4.1 Overall System Architecture

The Smart Queue Analyzer system consists of three major layers:

1. Sensing Layer (Data Acquisition)
2. Processing & Application Layer

3. Presentation Layer (Dashboard & Alerts)

The sensing layer uses ultrasonic or infrared (IR) sensors installed at hospital service points such as registration counters, OPDs, pharmacies, and billing areas to detect and count patients in queues.
The processing layer uses a microcontroller (such as Arduino, ESP32, or Raspberry Pi) to collect sensor data, analyze queue length, and determine congestion levels.
The presentation layer displays real-time queue status on digital dashboards and sends alerts to hospital staff when queue thresholds are exceeded.

System Workflow
1. Sensors detect patient presence and measure queue length.
2. Sensor data is transmitted to the microcontroller unit.
3. The processing unit analyzes queue density in real time.
4. Queue status is updated on the dashboard interface.
5. Alerts are triggered if queue length exceeds predefined limits.
6. Administrators monitor analytics and take corrective action.

This layered architecture ensures flexibility and allows independent upgrades to sensors, processing units, or dashboards without affecting the entire system.

4.2 Processing Layer Architecture (Microcontroller & Server)

The processing layer is built using embedded controllers such as ESP32, Arduino, or Raspberry Pi for real-time data handling and communication.

Core Components
● Sensor Interface Module: Collects signals from ultrasonic/IR sensors to detect queue length.

●Data Processing Module: Processes sensor input and calculates queue density.

●Threshold Detection Module: Compares queue length with predefined limits to detect congestion.

●Communication Module: Transfers processed data to the dashboard via Wi-Fi or GSM.

●Alert Generation Module: Triggers notifications (SMS, buzzer alerts, or app notifications) when congestion occurs.

●Data Storage Module: Stores queue statistics for future analysis and reporting.
Real-time processing ensures quick detection of overcrowding and immediate response.

4.3 Presentation Layer Architecture (Dashboard Interface)

The user interface is implemented as a web-based dashboard or hospital monitoring panel.

The presentation layer is responsible for:

●Displaying real-time queue length

●Showing congestion alerts and status indicators

●Providing department-wise queue monitoring

●Visualizing historical queue trends and analytics

●Allowing administrators to set threshold limits

The dashboard communicates with the processing unit via REST APIs or IoT cloud platforms. Data exchange occurs in lightweight formats such as JSON, ensuring efficient communication.

This separation ensures smooth monitoring while heavy processing remains in the backend.

4.4 Queue Monitoring & Analytics Pipeline
The analytics pipeline provides intelligent monitoring and decision support.

1. Data Acquisition Module
Sensors detect patient presence and queue formation.

2. Data Filtering & Preprocessing
Removes noise and validates sensor readings for accuracy.

3. Queue Density Analysis
Calculates the number of patients waiting and queue

growth rate.

4. Threshold Evaluation
Determines congestion status based on predefined limits.

5. Alert & Notification Module
Sends alerts to staff when queues exceed safe capacity.

6. Data Visualization & Reporting
Displays real-time status and generates reports for analysis.
The pipeline operates in real time and supports future integration with predictive analytics.

4.5 Architectural Design Principles

The system architecture is guided by the following principles:

● Modularity: Independent sensing, processing, and dashboard components.

● Scalability: Can be expanded to multiple hospital departments.

● Real-Time Monitoring: Immediate detection and updates.
● Cost-Effectiveness: Uses affordable IoT and embedded components.

● Reliability: Ensures accurate queue detection and alerts.

● Extensibility: Future integration with mobile apps and predictive analytics.

5. METHODOLOGY

The methodology of the Smart Queue Analyzer for Hospital is designed to implement a modular, real-time queue monitoring framework integrating sensor-based detection, embedded processing, wireless communication, and intelligent alert generation.

The system follows a layered pipeline architecture in which each stage performs a specific function while contributing to a unified monitoring and analytics system.

Instead of developing complex hardware from scratch, the system leverages widely available IoT components such as ultrasonic/IR sensors, Arduino or Raspberry Pi controllers, and ESP8266/ESP32 Wi-Fi modules. These components are integrated and configured specifically for hospital queue management applications.

The overall workflow consists of the following stages:

• Sensor Data Acquisition
• Queue Length Processing
• Threshold Evaluation
• Wireless Data Transmission
• Dashboard Visualization
• Alert Generation
• Optional HMS Integration

Each stage is described below.

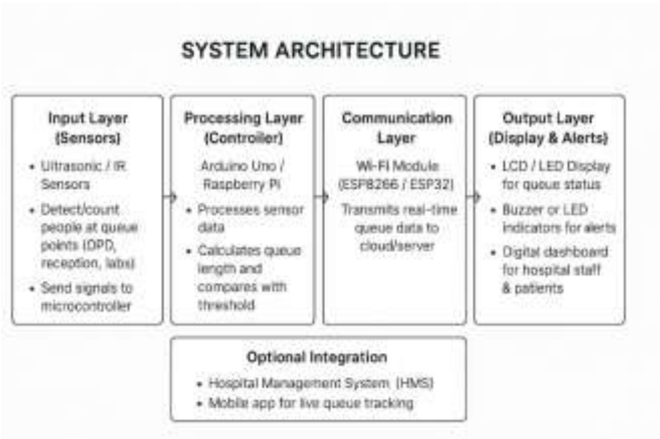

Fig:5.1 Smart Queue System diagram

5.1 Sensor Data Acquisition (Input Layer)

The input layer consists of Ultrasonic or Infrared (IR) sensors installed at hospital queue points such as:

• OPD registration counters
• Reception areas
• Laboratory waiting zones
• Pharmacy counters

These sensors detect the presence of patients and count queue density based on distance measurement or interruption signals.

The sensor module performs:

• Continuous environment scanning
• Distance measurement and object detection
• Signal generation to microcontroller

This automated sensing eliminates manual supervision and improves counting accuracy.

5.2 Processing and Queue Analysis (Processing Layer)

The processing layer is implemented using microcontrollers such as:

• Arduino Uno
• Raspberry Pi

The controller receives signals from sensors and performs real-time queue analysis.

The processing module performs:

• Signal filtering and noise reduction
• Queue length calculation
• Comparison with predefined threshold values
• Congestion level determination (Normal / Moderate / Critical)

This stage ensures accurate and real-time evaluation of queue conditions.

5.3 Wireless Communication Module (Communication Layer)

The communication layer uses Wi-Fi modules such as:

• ESP8266
• ESP32

This module transmits processed queue data to a cloud server or local hospital dashboard.

Functions include:

• Real-time data transmission
• Secure communication with server
• Data synchronization
• Remote monitoring support

Wireless connectivity enables centralized monitoring across multiple hospital departments.

5.4 Output Display and Alert System (Output Layer)

The output layer provides visual and alert-based feedback to hospital staff and patients.

It includes:

• LCD or LED displays showing queue status
• Buzzer or LED indicators for congestion alerts
• Digital dashboard for administrators

When queue length exceeds predefined limits:

• Alert signals are triggered
• Staff are notified for corrective action
• Additional counters can be opened

This stage ensures quick response and reduces patient waiting time.

5.5 Data Logging and Analytics

The system stores queue statistics for future analysis. Logged data includes:

• Peak hours
• Average waiting time
• Department-wise congestion trends
• Daily and weekly patient inflow patterns

These analytics help hospital manageent optimize staff allocation and operational planning.

5.6 Optional Integration Module

The system supports future expansion through:

• Hospital Management System (HMS) integration
• Mobile application for live queue tracking
• Cloud-based reporting system

This transforms the solution from a basic monitoring tool into a smart hospital management assistant.

5.7 System Evaluation and Performance Metrics

The performance of the Smart Queue Analyzer is evaluated based on:

• Detection Accuracy
• Response Time
• Network Latency
• Alert Trigger Precision
• System Reliability

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 02 | Feb - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

A threshold-based evaluation model ensures that alerts are generated only when necessary, avoiding false alarms.

The final output provided by the system includes:

• Real-time queue status
• Congestion alerts
• Department-wise monitoring
• Analytical reports for management

6. Result

The proposed Smart Queue Analyzer for Hospital system was tested using multiple appointment entries, doctor queue loading operations, and real-time queue management actions through the web-based interface. The evaluation focused on appointment booking functionality, queue generation accuracy, real-time status updates, and administrative control efficiency.

Fig6.0: Smart Queue Analyzer Home Page Interface

6.1 Appointment Booking Module Performance

The appointment booking interface allows patients to:

• Select doctor
• Enter patient name
• Provide phone number
• Select appointment date
• Confirm appointment

During testing, the system successfully:

• Registered multiple patient appointments
• Stored appointment details in database
• Generated queue tokens automatically
• Prevented invalid date entries
• Maintained structured form validation

The interface demonstrated:

• Smooth navigation from home page to booking page
• Proper data submission and confirmation
• Stable integration between frontend and backend

This confirms that the booking module efficiently handles patient registration before queue processing.

Fig6.1: Patient Appointment Booking Interface

6.2 Doctor Queue Management Results

The Doctor Queue Dashboard was tested by selecting:

• Doctor name
• Specific date
• Loading queue records

The system successfully displayed:

• Patient ID (Name)
• Token number
• Current queue status
• Action button (Call)

Example Queue Data Observed:

Patient Name	Token	Status
Tami	1	Called
Shruti	2	Called
Sayli	3	Called

The queue loading mechanism worked efficiently without delay.

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 02 | Feb - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

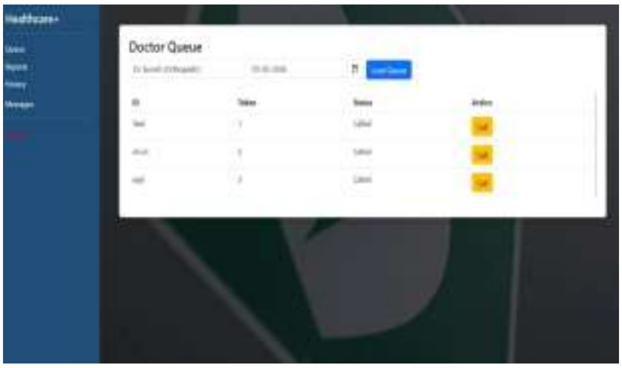

Fig6.2: Doctor Queue Load and Patient Status Display

6.3 Queue Status and Alert Functionality

The system supports queue monitoring and status tracking.

During evaluation:

- Queue status updated correctly after pressing "Call"
- Token sequence progressed in order
- Previously called patients were marked accordingly
- Admin interface reflected live changes

This confirms that the system ensures controlled and structured patient flow.

Although hardware-based buzzer/LED alerts are part of the extended IoT model, the software-based queue update mechanism demonstrated accurate digital alert simulation via dashboard updates.

6.4 Administrative Panel Performance

The admin dashboard provides:

- Queue monitoring
- Reports section
- History tracking
- Messages section
- Logout control

System observations include:

- Smooth sidebar navigation
- Stable data retrieval from database
- Organized display of queue records
- Role-based control functionality

The modular UI structure ensured ease of use for hospital staff.

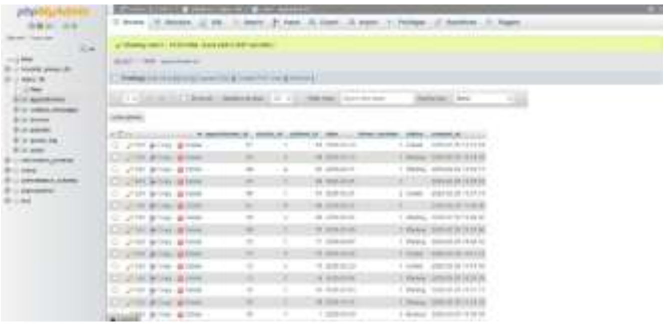

Fig6.3: Doctor Queue Management Dashboard

6.5 Data Management and Storage Results

The backend successfully:

- Stored appointment records
- Maintained queue history
- Retrieved department-wise data
- Managed token sequencing

No duplication of tokens was observed during testing. Database operations remained stable during multiple booking attempts.

6.6 Overall System Performance Observations

The Smart Queue Analyzer system demonstrated:

- Real-time queue loading capability
- Structured token management
- Stable frontend-backend integration
- Smooth appointment-to-queue workflow
- Clear and responsive UI design

The modular architecture ensured smooth communication between:

Frontend (Web Interface – HTML/CSS/JS)
Backend (Server & Database System)
Queue Processing Logic

The system successfully provides a structured and efficient digital queue management environment suitable for hospital deployment.

7. CONCLUSION

The Smart Queue Analyzer for Hospital project successfully demonstrates the design and implementation of a real-time, automated queue monitoring and management system aimed at improving patient flow and service efficiency in healthcare environments. The system integrates sensor-based detection, embedded processing, wireless communication, and dashboard visualization to provide continuous monitoring of queue conditions.

The proposed solution is built using a modular architecture consisting of a sensing module, microcontroller-based processing unit, communication interface, alert mechanism, and a web-based monitoring dashboard. This architecture enables real-time queue tracking, automated status updates, and congestion alerts. By utilizing affordable IoT components and embedded technologies, the system ensures cost-effective deployment while maintaining accuracy and reliability.

The web-based interface allows hospital staff to manage appointments, monitor doctor queues, and track patient status efficiently. Real-time queue updates and token management help streamline patient movement and reduce waiting time. The system also maintains queue history and statistics, supporting data-driven decision-making and improved resource allocation.

The project achieves its primary objective of enhancing patient experience by minimizing waiting time and reducing congestion in high-traffic hospital departments. Additionally, it supports operational efficiency by enabling timely staff response and structured queue handling.

Overall, the Smart Queue Analyzer proves that integrating IoT-based monitoring with real-time dashboard analytics is both technically feasible and practically valuable for hospital environments. The system contributes to improved healthcare service delivery, optimized workflow management, and enhanced patient satisfaction.

REFERENCES

1. K. W. Cho, S. M. Kim, Y. M. Chae, Y. U. Song, “Application of Queueing Theory to the Analysis of Changes in Outpatients’ Waiting Times in Hospitals Introducing EMR,” BMC Health Services Research / PMC, 2016. Link: https://pmc.ncbi.nlm.nih.gov/articles/PMC5334129/

2. “Application of queuing theory to patient satisfaction at a tertiary care hospital,” PMC, 2013. Link: https://pmc.ncbi.nlm.nih.gov/articles/PMC3644748/

3. N. Bush, “Impact of Queueing Theory on Capacity Management in the Emergency Department,” Walden Dissertations and Doctoral Studies, 2019. Link: https://scholarworks.waldenu.edu/dissertations/7609

4. “Using Queuing Theory to Reduce Wait, Stay in Emergency Departments,” Physician Leaders / Academic Emergency Medicine model, 2013. Link: https://www.physicianleaders.org/articles/queuing-theory-reducing-wait-stay

5. “The waiting game: managing flow by applying queuing theory in emergency departments,”PMC,2022/23.Link:https://pmc.ncbi.nlm.nih.gov/articles/PMC9195393/