Smart Queue System

1st Bryan Chen
School of Engineering and Technology
University of Washington
Tacoma, USA
brch@uw.edu

2nd Phuoc Le
School of Engineering and Technology
University of Washington
Tacoma, USA
Phuocle2@uw.edu

***Abstract*—Long waiting times in queues often lead to customer dissatisfaction and inefficiencies in service delivery. The Smart Queue System described in this paper leverages the Internet of Things (IoT) to make queue management processes. The proposed solution employs edge computing devices, cloud services, and real-time data visualization tools to check queue lengths, estimate waiting times, and distribute queue information to customers. By integrating Node-RED for workflow orchestration, Microsoft Azure services for data processing, Azure Custom Vision for crowd-level classification, and Grafana for analytics, the system offers a user-centric approach that optimizes resource allocation, reduces waiting times, and enhances overall customer experiences.**

I. Introduction

A. Background and Motivation

Long, unorganized queues remain a pervasive problem in high-traffic service areas such as hospitals, restaurant, and government institutions. The management systems often rely on manual processes—using overhead speakers or screens to call the next customer. These methods lack the flexibility and real-time adaptability required to handle fluctuating demand. Additionally, customers often face uncertainty regarding wait times, which can lead to frustration and reduced overall satisfaction.

In recent years, IoT solutions have emerged as powerful tools to make process automation and real-time data collection better. By integrating sensor devices, cloud platforms, and interactive dashboards, IoT-driven queue management systems can minimize waiting times, provide transparency, and optimize resource usage. This paper presents a novel "Smart Queue System" that applies IoT principles—including Azure Custom Vision to differentiate crowd levels—to automate generation, manage customer flow, and provide real-time analytics.

B. Project Goals and Objectives

The primary goals of the Smart Queue System are:

- Reduce Waiting Times: Dynamically adjust the queue process and alert customers of their turn through LCD and audio notification.
- Enhance Customer Satisfaction: Might suggest help based on how Custom Image Model classifies situations based on number of customers
- Enable Data-Driven Decisions: Gather and analyze queue-related data to optimize service counters, staffing, and overall operational efficiency.

II. Related Work or Concepts

A. Traditional Queue Management

Traditional queue management systems often involve physical tokens and static displays. While such systems can handle basic distribution, they lack flexibility and scalability. Customers have no accurate insight into waiting times, and service providers cannot dynamically adjust the queue based on real-time data. These limitations often lead to inefficiencies and dissatisfaction.

B. IoT-Enabled Solutions

Recent advancements in IoT have introduced a capabilities such as:

- Remote Monitoring and Control: Using edge devices (e.g., Raspberry Pi or microcontrollers) to collect real-time data from sensors, such as the number of people waiting.
- Cloud Integration: Using platforms like Microsoft Azure IoT Hub to process large volumes of data, enabling more advanced analytics and scalability.
- Data Visualization and Analytics: Tools like Grafana provide dynamic dashboards, real-time analytics, and customizable alerts.
- Computer Vision: Azure Custom Vision models can classify crowd levels or detect the number of people in a waiting area, providing additional data to enhance queue management.

C. Gaps in Existing Solutions

Although multiple queue management products exist, many still rely on proprietary hardware or lack user-centric features such as personalized notifications. Most do not leverage computer vision to assess crowd density, which lead to inaccurate wait-time predictions when crowds grow unexpectedly. The Smart Queue System addresses these gaps by offering a modular, scalable solution that integrates smoothly with existing infrastructure while giving comprehensive data analytics.

III. Proposed IoT Solution

A. System Overview

Node-RED: A flow-based development tool that simplifies data collection, processing, and device control [1]. The Smart Queue System combines an edge device, cloud services, and a user interface to deliver real-time queue updates. The major components include Edge Device: A microcontroller or Raspberry Pi that manages local queue operations, such as interacting with a display module. Node-RED Flows: A flow-based logic controller that orchestrates data transfer, processes incoming sensor data, and sends relevant updates to the cloud. Azure IoT Services: A suite of cloud services for data ingestion, storage, and analysis. Azure IoT Hub ensures secure communication between the edge device and the cloud [2], while Azure Functions can process incoming data in real time. Data Visualization: Local Dashboards provide real-time insights into queue length and usage statistics for Grafana Dashboard. Computer Vision Service: Azure Custom Vision for crowd-level classification, using scheduled photo captures or live camera feeds to determine the waiting area's occupancy and inform the queue management logic.

B. Architecture and Workflows

The high-level system architecture:

Data Capture: The edge device collects queue-related data (e.g., number of people in queue, temperature, humidity, etc.)

Local Processing: Node-RED, running on the edge device, processes these inputs and forwards essential information to Azure IoT Hub.

Cloud Processing: Azure IoT Hub receives the data, which is then stored in a database. Additional analytics can be performed to derive insights such as peak hours.

Visualization: Grafana or Azure Dashboard fetches the processed data to display real-time queue status and historical trends.

User Interaction: Customers will get a checking number if there people waiting in the queue. When crowd levels are high, it shows in local dashboards for manager to see.

C. Hardware Prototype

The prototype includes: A Raspberry Pi or similar board for local control, a small LCD display to show queue information.

D. Software Components

Node-RED: Used for designing logical flows. For instance, when a person is registered, Node-RED increments the queue count and publishes an event to the cloud.

Azure IoT Hub: Acts as a message sender between the local device and the cloud-based services.

Grafana: Queries the database to display real-time and historical queue metrics in a user-friendly dashboard.

IV. Decisions and/or Observations

A. Data Collection and Analysis

The During the prototype phase, we collected queue metrics such as:

- Temperature and humidity: Observed through real-time monitoring of how waiting location feeling so we can turn on and off air conditioner.
- Peak Hours: Identified through timestamp analysis of issuance and service completion.
- Queue Length Patterns: Observed through real-time monitoring of how many customers are waiting at any given time.
- The crowded classifications offered an additional layer of validation, ensuring that the system’s wait-time predictions remained accurate even when unexpected surges occurred.

B. Workflow Efficiency

Node-RED flows allowed for straightforward integration of various sensors, display modules, and cloud endpoints. This simplified system development and enabled rapid iteration. The decision to use a single board computer (e.g., Raspberry Pi) for edge computing ensured that local processing could continue even if the cloud connection was temporarily lost.

C. Cloud vs. Edge Trade-offs

Storing and processing data on the cloud allowed for scalability and the use of advanced analytics, but also introduced latency and dependency on stable internet connectivity. The system design thus balanced edge and cloud processing by performing real-time tasks locally while delegating historical analysis and advanced processing to the cloud [2].

V. Discussion

A. Cloud vs. Edge Trade-offs

One of the key benefits of the Smart Queue System is its ability to notify customers about their queue status in real time. This significantly reduces the perceived waiting time and can improve overall user satisfaction. Also, real-time notifications help prevent congestion around service counters, as customers only need to be present when their turn is imminent.

B. Data Flow and Security Considerations

Data flows from the edge device to the cloud via Azure IoT Hub, which offers secure communication. In addition, access controls within Azure ensure that only authorized personnel can view data. For large-scale deployments, encryption and secure device provisioning become crucial. There will need to be a team to work on it to make sure there is no leakage of data

C. *Scalability and Reliability*

The system's modular design allows it to be scaled to multiple service counters or multiple locations. Node-RED's flow-based approach is intuitive and can handle additional nodes with minimal configuration changes [1]. By relying on Azure's global infrastructure, the solution can maintain reliability and accommodate varying loads.

D. *Limitations*

Hardware Constraints: The prototype relies on stable power and internet connectivity.

Cost: While cloud services can be cost-effective at scale, smaller deployments may find the initial setup costs relatively high.

User: Success depends on customers actively engaging with the system (e.g., wave hand to register their position).

Image Quality: Computer vision accuracy depends on adequate lighting and camera placement. Poor image quality can affect crowd classification results [2].

VI. Conclusion

The Smart Queue System offers a modernized approach to queue management by integrating IoT devices, cloud services, and real-time analytics. The solution addresses the common pain points of traditional queue systems—long waiting times, customer frustration, and inefficiencies in staff allocation. Through the use of Node-RED for workflow automation, Azure IoT services for secure data processing, and Grafana for intuitive dashboards, the system provides actionable insights to both service providers and customers.

Future Work could involve leveraging machine learning models to predict waiting times more accurately and automating counter assignment based on demand. Additionally, integrating third-party scheduling systems or chatbots could further enhance the user experience. Overall, the Smart Queue System demonstrates the potential of IoT in delivering improvements to everyday processes, paving the way for broader adoption in various public and private service sectors.

References

[1] Node-RED, "Node-RED Documentation," [Online]. Available: https://nodered.org/docs/ [Accessed: Mar. 13, 2025].

[2] Microsoft Azure, "Azure IoT Hub Documentation," [Online]. Available: https://docs.microsoft.com/en-us/azure/iot-hub/ [Accessed: Mar. 12, 2025].

[3] Grafana Labs, "Grafana Documentation," [Online]. Available: https://grafana.com/docs/grafana/latest/ [Accessed: Mar. 10, 2025].