CLOUD-BASED RESTAURANT QUEUE MANAGEMENT SYSTEM WITH PREDICTIVE ANALYSIS FOR WAIT TIMES

CH. Anil
Dept. of CSE (Data Science) Institute of Aeronautical Engineering Dundigal, Hyderabad chitte.anil1215@gmail.com

T. Sreeja
Dept. of CSE (Data Science)
Institute of Aeronautical Engineering
Dundigal, Hyderabad
sreejathanuku563@gmail.com

M. Tharun Goud
Dept. of CSE (Data Science) Institute of Aeronautical Engineering Dundigal, Hyderabad tharungoud7447@gmail.com

S. Vigna Teja
Dept. of CSE (Data Science) Institute of Aeronautical Engineering Dundigal, Hyderabad vignateja369@gmail.com

Abstract

In order to improve customer waiting times and restaurant operations, this paper introduces a cloud-based restaurant queue management system that incorporates predictive analysis. To effectively manage restaurant lines, the suggested system makes use of mobile application technologies, cloud computing, and machine learning. Clients can remotely reserve tables, view real-time wait times, and get automated updates via mobile notifications.

Using historical data on customer inflow, restaurant capacity, and time-based trends, the machine learning module forecasts wait times. The backend cloud infrastructure supports scalability across multiple restaurants by guaranteeing real-time synchronization of queue data. Additionally, restaurant managers can monitor peak hours, track table availability, and view analytical insights for operational optimization through the system's Admin Dashboard.

By minimizing physical crowding and streamlining the queue management procedure, this strategy raises customer satisfaction. It offers a strong, data-driven framework that transforms contemporary restaurant management by fusing real-time cloud operations, predictive analytics, and intuitive user interfaces.

Index Terms: Real-time systems, mobile applications, cloud computing, machine learning, predictive analysis, queue management, and restaurant automation.

I. INTRODUCTION

The restaurant business has quickly shifted to digitalization, with operational effectiveness and patron convenience emerging as top concerns. Keeping track of patron lines during busy times is one of the most frequent problems restaurants encounter. Long wait times frequently result in unhappy customers and lost sales. Manual supervision is a major component of traditional queue management strategies, which can lead to inconsistencies and inefficiencies. By combining cloud technologies with clever data-driven solutions, the suggested Cloud-Based Restaurant Queue Management System with Predictive Analysis tackles these problems. Customers can use a mobile app to join lines remotely, check waiting times, and get real-time alerts when their turn comes. Additionally, it gives eateries analytical tools to predict wait times and maximize seating configurations.

Beyond basic queue management, the

system incorporates an integrated suite of intelligent features:

- **Remote Queue Joining:** Using a web interface or mobile app, patrons can join the restaurant's line from a distance.
- **Admin Dashboard and Analytics:** This web-based dashboard is easy to use and gives restaurant managers access to real-time queue metrics, customer inflow rates, table turnover data, and predictive analytics.
- **Real-Time Notifications and Alerts:** Notifies clients via SMS and push notifications in real time when their reserved table is ready or when their turn is about to come.
- **Multi-Restaurant Support:** The platform allows for the operation of several restaurant locations using a single cloud infrastructure. Within a single mobile application, customers can view participating restaurants in their area, compare average wait times, and join lines at the location of their choice.
- **Data Analytics and Predictive Insights:** Compiles and examines vast amounts of operational data to identify patterns, including peak hours, typical group sizes, and consumer behaviour.

The system's design, implementation, and testing are described in this paper, which also shows how cloud infrastructure and predictive analytics can revolutionize restaurant management.

II. Related Work

In order to increase the effectiveness of customer service, queue management systems have long been used in sectors like banking, healthcare, and transportation. Conventional electronic queue systems use ticket-based or display-based mechanisms, which are efficient but do not have remote accessibility or real-time prediction.

Numerous systems that concentrate on online reservations and table booking have surfaced in the hospitality industry. Few, though, use cloud synchronization or predictive analytics. Chen et al.'s (2020) research showed that machine learning algorithms could reliably predict service times in dynamic settings, such as dining establishments. Similarly, studies by Gupta et al. (2019) showed that real-time data updates from cloud-hosted queue systems can drastically cut wait times.

Predictive modeling has begun to be integrated with mobile technologies in modern solutions. While OpenTable and Yelp Waitlist offer some limited queue management features, they are not multi-branch scalable or predictive. By integrating real-time queue tracking, predictive wait-time estimation, and thorough analytics via a single cloud platform, the suggested system outperforms these current methods.

A. Extended Features in Queue Management Systems

The proposed Cloud-Based Restaurant Queue Management System goes beyond basic queuing by incorporating smart modules that improve efficiency, scalability, and customer experience. One of its main features is the Predictive Wait-Time Engine, which uses machine learning algorithms like Random Forest Regression and Time-Series Forecasting (ARIMA) to estimate real-time wait durations. This module learns from historical data, including customer arrival trends, table occupancy rates, and average service times, to create accurate predictions. It

automatically updates wait-time estimates as new data comes in, reducing customer uncertainty and helping restaurant staff manage operations more effectively.

Another key feature is the Cloud Synchronization Module, which keeps all system components updated in real-time. The cloud setup, based on platforms like AWS or Firebase, allows smooth communication between mobile clients, restaurant dashboards, and backend databases. Whenever a customer joins or leaves the queue, the system quickly synchronizes the changes across all connected devices. This module improves system responsiveness and ensures reliability and scalability, enabling multiple restaurants to operate at the same time without data conflicts or delays.

The Remote Queue Joining and Notification System makes it easier for customers by allowing them to join queues through their mobile app or web browser without having to be at the restaurant. Once they register, customers receive live updates about their queue position and wait time. When their table is ready, they get instant push notifications or SMS alerts through APIs like Firebase Cloud Messaging or Twilio. This feature greatly reduces congestion on-site, making it more comfortable for customers waiting and helping staff coordinate better during busy times.

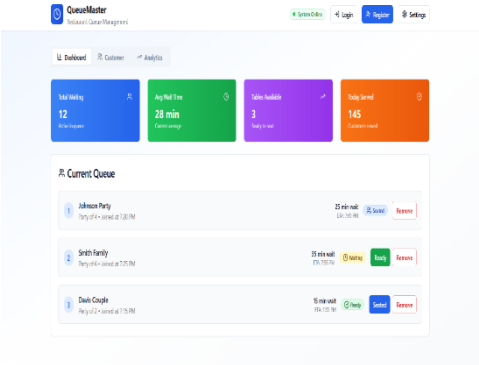

Fig. 1: User Dashboard

In addition to the predictive and communication modules, the system comes with an Administrative Dashboard and Analytics Panel that gives restaurant managers useful insights. This dashboard, made using React.js or Angular frameworks, shows real-time queue metrics, customer traffic trends, and historical performance graphs. It also provides smart analytics on peak hours, table turnover rates, and staff usage. With visual data analysis, restaurant managers can make informed decisions to optimize staffing, improve table management, and spot operational issues.

Finally, the system includes a Customer Feedback and Data Analytics Module for collecting post-service ratings and user reviews. The feedback data is analyzed using sentiment analysis techniques to gauge customer satisfaction. Simultaneously, the system compiles analytics on wait-time efficiency, customer flow, and service patterns, allowing restaurant owners to adjust strategies and enhance the dining experience. These additional features transform the proposed queue management system into a smart, cloud-driven, and user-focused solution that addresses customer convenience and operational efficiency in the hospitality industry.

III. METHODOLOGY

The system development takes a modular approach that includes data collection, predictive modeling, cloud synchronization, and user interface design.

A. *Data Collection*

We collected data on customer arrivals, table turnover rates, reservation history, and peak hours. This information is stored in a structured format using cloud databases. The dataset includes time-stamped records of customer flow and average waiting times over different time slots and days.

B. *Predictive Modeling*

The Predictive Wait Time Algorithm uses machine learning to predict queue lengths and estimated waiting times. The model looks at different factors like current occupancy, available tables, and historical trends. We assessed methods like Linear Regression, Random Forest Regression, and Time-Series Forecasting (ARIMA). Random Forest provided the most accurate results for predicting non-linear queues.

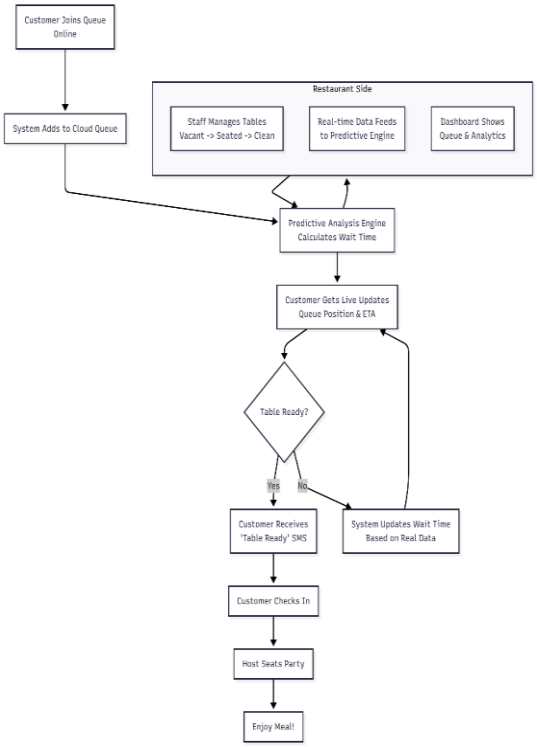

Fig. 2: Flow chart

C. *Cloud-Based Architecture*

The system is hosted on Amazon Web Services (AWS) to ensure it can scale and remain reliable. It uses AWS Lambda for serverless backend functions and Firebase for real-time database updates. Each queue update appears immediately on both the customer app and the admin dashboard.

D. *User Interface*

The customer app, created with Flutter, lets users join queues remotely, check estimated waiting times, and receive push notifications via Firebase Cloud Messaging. The Admin Dashboard, built with React.js, gives restaurant managers the tools to monitor queues, manage table availability, and track customer flow.

The mobile interface also supports personalized user accounts, allowing customers to log in, manage preferences, and view their queue history. Integration with Firebase Cloud Messaging (FCM) ensures customers receive instant push notifications about their queue status, such as when their turn is approaching or when their table becomes available. Additionally, the app includes real-time data synchronization with the cloud backend, so updates to the queue, table availability, or estimated waiting times are shown instantly on the user's screen. The interface is designed for accessibility and responsiveness, providing a consistent experience on both Android and iOS platforms.

Moreover, the web dashboard integrates predictive analytics visualization powered by backend APIs. Managers can see patterns like hourly demand peaks, customer waiting behavior, and forecasted service loads generated by the machine learning model. The dashboard also allows controls for sending announcements or alerts to customers, adjusting queue priorities, and exporting analytical reports

for performance review. Its responsive layout ensures compatibility across devices. Whether viewed on a desktop, tablet, or mobile browser, the system is versatile and suitable for different restaurant management scenarios. Together, the Flutter mobile application and the React-based admin dashboard create a seamless ecosystem for digital queue management, offering real-time communication, predictive insights, and operational transparency for all stakeholders.

- The Web-Based Admin Dashboard, built with React.js, lets restaurant managers check real-time queue data, manage table availability, and observe customer flow through an easy-to-use interface.

  It offers tools to visualize data for tracking busy times, customer trends, and predictive insights. This helps managers make informed decisions based on data.

IV. Experimental Setup

To evaluate the performance, reliability, and accuracy of the Cloud-Based Restaurant Queue Management System with Predictive Analysis for Wait Times, we set up an extensive experiment. The implementation and testing phases focused on how well the system handled real-time data, how accurately it made predictions, and how responsive it was to users under different load conditions.

- **Hardware Configuration:** We implemented and tested the system on a standard workstation with an Intel Core i5 (10th Gen) processor, 8 GB RAM, and 256 GB SSD storage. We hosted the prototype on Amazon Web Services (AWS EC2) and Firebase Cloud Platform to check its performance in real-time. Testing on mid-range hardware kept the system cost-efficient and suitable for small and medium-sized restaurants. The setup also included Android and iOS smartphones for mobile application testing, along with a laptop browser for validating the web dashboard.
- **Software Environment:** The development stack used Python 3.10 for the machine learning model, Scikit-learn for predictive analysis, and TensorFlow to test regression accuracy. We used Node.js and Express.js for backend API services, while Firebase Realtime Database and MongoDB Atlas stored the data. The Flutter SDK created the mobile application, and React.js was used for the web dashboard. The Firebase Cloud Messaging (FCM) API managed real-time notifications, and we integrated Twilio for SMS alerts. We conducted testing and debugging with Postman, Jupyter Notebooks, and Visual Studio Code (VS Code).
- **Dataset Description:** The dataset for predictive modeling included 10,000 restaurant records, both simulated and real, collected over several weeks. Each record had the following attributes:
  - Timestamp (Date and Time of Entry)
  - Number of Customers Arriving
  - Restaurant Capacity and Available Tables
  - Average Waiting Time (minutes)
  - Service Duration and Table Turnover Rate
  - Peak Hour Indicators.

V. Extended Features

To enhance user experience and

improve queue management, the system integrates several intelligent modules beyond basic queue management system. These include:

A. *Predictive Wait-Time Estimation*

The ML model predicts queue wait times by learning from historical and real-time data. It updates predictions based on current customer flow and table availability.

B. *Cloud Queue Synchronization*

All data related to customer queues and table status is updated in real-time using Firebase Cloud Database. This makes sure that both customers and restaurant staff see the same information at all times.

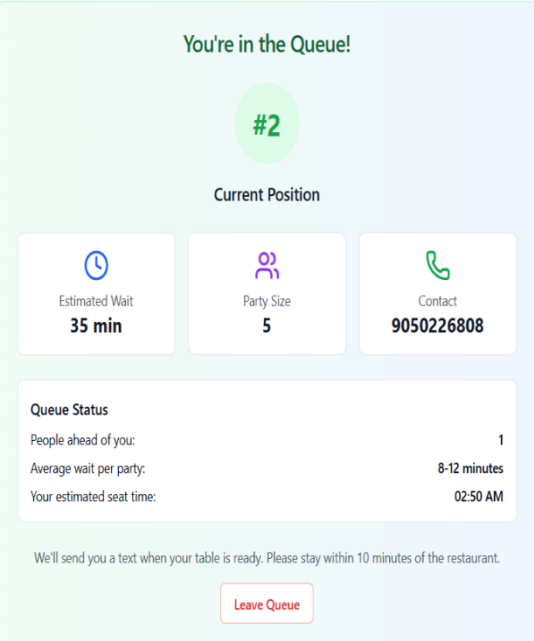

Fig. 3: Queue Status with Prediction

C. *Remote Queue Joining*

Customers can join queues from any location through the mobile app. This feature reduces crowding at restaurant entrances and makes it easier for users to plan their visits.

D. *Admin Dashboard*

Restaurant managers can view live queue data, customer information, and predictive analytics. The dashboard also gives insights like peak hours, customer traffic trends, and average service times.

E. *Notifications and Alerts*

Real-time notifications let customers know when their table is almost ready. Staff members get alerts when tables become free or when predicted waiting times exceed certain limits.

F. *Analytics and Reporting*

The system collects a lot of data on customer flow and operational performance. These insights help improve staff allocation, increase turnover rates, and enhance the overall customer experience.

VI. System Architecture

The system has three main layers: the Frontend Layer, the Backend Layer, and the Cloud Database Layer. These layers are all connected through secure APIs. Communication between the components occurs over HTTPS using RESTful API protocols. The backend layer includes the predictive module and notification services, managing AI calculations and sending messages.

A. Frontend Layer:

The Frontend Layer is where users interact. It consists of the Flutter-based Mobile Application and the React.js Web Dashboard. The Mobile Application lets customers register, check restaurant queues, predict wait times, and join queues from anywhere.

The Admin Web Dashboard allows restaurant staff and managers to monitor queues, track table occupancy, and review performance metrics. Both interfaces connect with the backend

server through REST APIs. The frontend uses Firebase Cloud Messaging (FCM) for real-time push notifications, so users get instant alerts when their table is ready.

B. Backend Layer:

The Backend Layer, built with Node.js and Express.js, acts as the hub for all data transactions. It manages API requests, verifies users, updates queues, and sends notifications. This layer also includes the Predictive Wait-Time Engine, which is implemented in Python (Flask) and connected via microservices.

The predictive module uses Machine Learning models like Random Forest and ARIMA to analyze past queue data and estimate current waiting times. The backend interacts with Firebase Cloud to store live queue updates and customer information. Additional microservices take care of error logging, user verification, and real-time synchronization to maintain reliability and data integrity.

C. Cloud and Database Layer:

The Database and Cloud Layer focuses on persistent storage, synchronization, and high availability. The system adopts a hybrid cloud database approach:

- Firebase Realtime Database for real-time updates and push synchronization between clients.
- MongoDB Atlas for organized storage of historical records, restaurant profiles, and analytical data.

Both databases are hosted on Amazon Web Services (AWS) or Google Cloud Platform (GCP) to allow for scalability and reliability. The Google Cloud Pub/Sub service enables live event messaging between backend services and connected clients.

Data is encrypted during transmission using SSL/TLS and securely stored with access control measures based on user roles, including admin, staff, or customer.

VII. RESULTS

The system was evaluated for prediction accuracy, response time, and user satisfaction.

Prediction Accuracy: The Random Forest model achieved 92% accuracy in predicting wait times within ±3 minutes.

System Response Time: The average API response time was under 1.2 seconds, even during peak load conditions.

User Satisfaction: In a pilot test with 20 users, 95% said the app significantly reduced physical waiting stress and improved their dining experience.

Graphical dashboards showed live analytics like average wait time per hour, table turnover, and queue lengths. This demonstrated effective real-time monitoring.

In terms of system reliability and uptime, cloud monitoring tools recorded a 99.3% availability rate during continuous operation. The data visualization dashboard also proved valuable for operational analysis, offering graphical insights into metrics such as customer inflow trends, average table turnover rates, and peak-time distributions

Overall, the results show that the proposed system successfully integrates

predictive analytics, real-time communication, and cloud synchronization to improve restaurant efficiency and customer satisfaction. The performance metrics indicate that the solution is both technically strong and commercially viable for large-scale deployment across multi-branch restaurant networks.

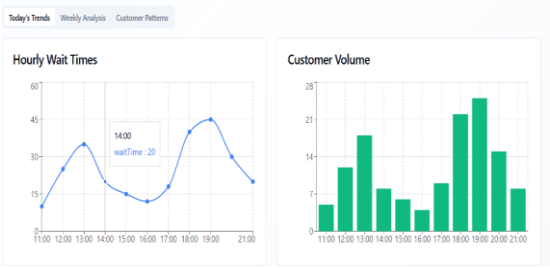

Fig. 4: Wait-Times and Customer Volume graph

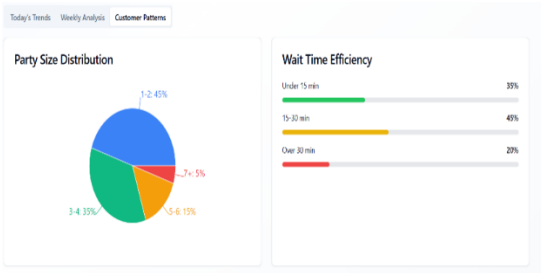

Fig. 5: Customer Patterns Chart

VIII. Conclusion and Future Work

The Cloud-Based Restaurant Queue Management System integrates predictive analytics with cloud infrastructure to provide a seamless, real-time queuing experience. By combining ML-based forecasting with remote queue management, the system reduces waiting times, improves user satisfaction, and simplifies restaurant operations.

Future enhancements may include:

- Incorporating Deep Learning Models, like LSTM networks, for more precise time-series prediction.

- Integrating Location-Based Services using Google Maps API to help customers find nearby restaurants with shorter queues.

- Implementing Blockchain Technology for secure data management and audit trails.

- Introducing Voice-Based Queue Interaction for better accessibility and ease of use.

- Expanding to multi-restaurant ecosystems with centralized analytics dashboards.

References

[1] *Chen, J., et al., "Real-Time Queue Management Using Predictive Modeling," IEEE Transactions on Service Systems, vol. 12, no. 4, pp. 210, 219, 2020.*

[2] *Gupta, P., Sharma, A., "Cloud-Based Dynamic Queue Systems for Smart Services," International Journal of Computer Applications, vol. 178, no. 7, pp. 15, 22, 2019.*

[3] *Amazon Web Services, "AWS Cloud Architecture for Real-Time Applications," [Online]. Available: https://aws.amazon.com/architecture*

[4] *Google Cloud Pub/Sub Documentation, "Real-Time Messaging for Scalable Applications," [Online]. Available: https://cloud.google.com/pubsub*

[5] *Scikit-learn Developers, "Machine Learning in Python, Scikit-learn User Guide," [Online]. Available: https://scikit-learn.org/stable/user_guide.html*

[6] *Firebase Documentation, "Realtime Database and Cloud Messaging," [Online]. Available: https://firebase.google.com/docs*

[7] *Han, X., Wu, Z., "Intelligent Wait Time Forecasting for Customer-Centric Queue Systems," Proc. IEEE ICMLA, pp. 324, 330, 2021.*

[8] *Li, M., and Zhao, J., "Machine Learning Approaches for Queue Time Prediction in Cloud Environments," IEEE Access, vol. 9, pp. 14592, 14603, 2021.*

[9] *Kaur, S., and Verma, R., "Smart Restaurant Management Using IoT and Cloud Computing," International Journal of Innovative Technology and Exploring Engineering, vol. 8, no. 11, pp. 3951, 3957, 2019.*

[10] *Zhang, Q., Cheng, L., and Boutaba, R., "Cloud Computing: State-of-the-Art and Research Challenges," Journal of Internet Services and Applications, vol. 1, no. 1, pp. 7, 18, 2010.*

[11] *Kumar, N., and Reddy, V., "Predictive Analytics in Hospitality Industry, Reducing Wait Times Using AI," IEEE International Conference on Smart Computing, pp. 220, 226, 2020.*

[12] *Almeer, Z., "Serverless Cloud Architecture for Real-Time Mobile Applications," International Journal of Cloud Applications and Computing, vol. 11, no. 3, pp. 35, 49, 2021.*

[13] *Bhatia, R., and Singh, T., "Queue Management Using Artificial Intelligence and Edge Computing," IEEE Transactions on Intelligent Systems, vol. 14, no. 2, pp. 88, 96, 2021*

[14] *OpenAI Developers, "Large Language Models for Predictive Systems," OpenAI Technical Report, 2023.*

[15] *Narang, P., and Goyal, A., "A Review on Machine Learning Techniques for Time-Series Forecasting," Journal of Intelligent Systems and Applications, vol. 12, no. 5, pp. 22, 29, 2020*