**High-Level Design Document:   
Restaurant Order Notification System**

Table of Contents

[Introduction: 1](#_Toc166408376)

[**1.** System Architecture 1](#_Toc166408377)

[3. Component Descriptions: 2](#_Toc166408378)

[4. Data Flow: 2](#_Toc166408379)

[5. Technology Stack: 2](#_Toc166408380)

[6. Deployment Architecture: 2](#_Toc166408381)

[7. Scalability and Performance Considerations: 3](#_Toc166408382)

[8. Security Considerations: 3](#_Toc166408383)

[9. Error Handling and Resilience: 3](#_Toc166408384)

[10. Monitoring and Logging: 3](#_Toc166408385)

[11. Future Enhancements: - 3](#_Toc166408386)

[Screens for this Project 3](#_Toc166408387)

[Login Screen: 3](#_Toc166408388)

[Dashboard: 3](#_Toc166408389)

[Order Entry Screen: 3](#_Toc166408390)

[Send SMS Screen (New Screen): 4](#_Toc166408391)

[Notification Log Screen: 4](#_Toc166408392)

[Settings Screen: 4](#_Toc166408393)

[Profile Screen: 4](#_Toc166408394)

[Logout Screen: 4](#_Toc166408395)

## Introduction:

The High-Level Design Document provides an overview of the architecture and components of the Restaurant Order Notification System. This document outlines the system's design principles, major components, and interactions.

A screenshot of a menu

Description automatically generated

## System Architecture

**The system architecture follows a client-server model, with the following major components**

* **Notification Service:** Responsible for sending SMS notifications to customers.
* **Web Interface:** Allows restaurant owners to enter customer information (Name and telephone number) and trigger order notifications.

## 3. Component Descriptions:

**Notification Service:**

* The Notification Service is responsible for processing orders and triggering SMS notifications to customers.
* It utilizes the Twilio API to send SMS messages, integrating with the Twilio service for message delivery.
* Customer information (name and telephone number) is provided directly by restaurant owners via the Web Interface.

**Web Interface:**

* The Web Interface provides a user-friendly interface for restaurant owners to enter customer information and trigger order notifications.
* It allows restaurant owners to input customer name, telephone number, and order details manually.
* The Web Interface communicates with the Notification Service to trigger order notifications.

## 4. Data Flow:

* Customer information (name and telephone number) and order details are input manually by restaurant owners via the Web Interface.
* The Web Interface forwards the customer information and order details to the Notification Service.
* The Notification Service utilizes the customer information to trigger SMS notifications via the Twilio API.

## 5. Technology Stack:

* **Notification Service:**
  + Programming Language: Python
  + Framework: Flask
  + Third-party Service: Twilio API
* **Web Interface:**
  + Frontend Framework: React.js
  + Backend Framework: Node.js (Express)
  + Security: HTTPS for secure communication

## 6. Deployment Architecture:

* The system will be deployed in a cloud environment, such as AWS or Azure.
* Deployment will utilize containerization technologies, such as Docker, for scalability and flexibility.
* Load balancers and auto-scaling groups will be used to ensure high availability and performance.

## 7. Scalability and Performance Considerations:

* The system architecture is designed to scale horizontally to handle increasing numbers of orders and notifications.
* Load balancing and auto-scaling mechanisms will dynamically adjust resource allocation based on demand to maintain optimal performance.

## 8. Security Considerations:

* Customer information will be encrypted during transmission and storage to ensure confidentiality.
* Access controls and authentication mechanisms will be implemented to restrict access to sensitive data.

## 9. Error Handling and Resilience:

* The system will implement robust error handling mechanisms to handle exceptions gracefully.
* Resilience will be ensured through redundancy and failover mechanisms at both the application and infrastructure levels.

10. Monitoring and Logging: -

Comprehensive monitoring and logging will be implemented to track system performance, identify issues, and troubleshoot errors. - Metrics such as response times, error rates, and system availability will be monitored using tools like Prometheus and Grafana.

## 11. Future Enhancements: -

Potential future enhancements include: - Integration with order management systems to automate customer data retrieval. - Support for additional notification channels (e.g., email, push notifications). - Advanced analytics and reporting capabilities to gain insights into customer behavior.

# Screens for this Project

## Login Screen:

* + Allows staff to log in to the system using their credentials.

## Dashboard:

* + Provides an overview of system activity, such as pending orders and notification status.
  + Includes navigation links to other sections of the system.

## Order Entry Screen:

* + Allows staff to manually input customer details obtained from the order management system.
  + Includes fields for customer name, phone number, and order number.
  + May include validation checks for data accuracy.

Order Management Screen

Customer Name: [John]

Telephone: [212-450-5409]

Order ID: [10024] (from Poc)

[Submit]

## Send SMS Screen (New Screen):

* + This screen allows the restaurant staff to send SMS notifications to customers when their orders are ready for pickup.
  + Below the form is the order list table displaying the current orders.
  + Each row in the table represents an order with columns for Order ID, Customer Name, Telephone, Order Description, Status, and Action.
  + The "Status" column indicates whether the order is pending or ready for pickup.
  + The "Action" column contains a button labeled "Send SMS" for each order, allowing the restaurant owner to trigger an SMS notification when the order is ready.

A menu with numbers and text

Description automatically generated with medium confidence

## Notification Log Screen:

* + Displays a log of sent notifications, including timestamps and delivery status.
  + Allows staff to track the status of notifications and troubleshoot delivery issues if any.

## Settings Screen:

* + Provides options for configuring system settings, such as notification templates and SMS gateway integration.

## Profile Screen:

* + Allows users to view and update their profile information, such as contact details and password.

## Logout Screen:

* + Confirms the user's intention to log out of the system and redirects them to the login screen upon confirmation.