**Software Design Document (SDD): Order Notification System**

Table of Contents

[1. Introduction 1](#_Toc166408658)

[2. Architectural Overview 1](#_Toc166408659)

[System Architecture: 1](#_Toc166408660)

[Component Diagram: 1](#_Toc166408661)

[3. Design Considerations 1](#_Toc166408662)

[4. Detailed Design Specifications 1](#_Toc166408663)

[5. Deployment Architecture 1](#_Toc166408664)

[6. Integration Points 2](#_Toc166408665)

[7. Data Flow Diagram 2](#_Toc166408666)

[8. Assumptions and Constraints 2](#_Toc166408667)

[Assumptions include: 2](#_Toc166408668)

[Constraints include: 2](#_Toc166408669)

[9. Glossary 2](#_Toc166408670)

[Some key terms and acronyms related to this project: 2](#_Toc166408671)

# 1. Introduction

The Software Design Document (SDD) provides a detailed blueprint for the design and implementation of the Order Notification System. This document outlines the architectural design, detailed design specifications, and integration points for the system.

# 2. Architectural Overview

## System Architecture:

The Order Notification System follows a client-server architecture. The server-side component includes a web application for order entry and notification triggering, while the client-side component consists of the SMS gateway service for sending notifications to customers.

## Component Diagram:

The system components include:

* Web Interface: Allows restaurant staff to manually input customer details.
* Notification Service: Triggers SMS notifications to customers.
* SMS Gateway: External service for sending SMS notifications.

# 3. Design Considerations

**Scalability:** The system design supports scalability by using a distributed architecture and load balancing mechanisms to handle increased user volume and system usage.

**Reliability:** To ensure reliability, the system implements fault tolerance mechanisms such as redundancy in server infrastructure and error handling in notification triggering logic.

**Security:** Security measures include data encryption for sensitive customer information and access control mechanisms to restrict unauthorized access to system functionalities.

# 4. Detailed Design Specifications

**User Interface Design:** The web interface follows modern design principles, providing an intuitive user experience for restaurant staff to input customer details efficiently.

**Database Design:** The system uses a relational database (e.g., MySQL) to store customer details and notification logs. The database schema includes tables for customers, orders, and notification logs.

**Notification Triggering Logic:** The system triggers SMS notifications based on order status. When an order is marked as ready for pickup, the notification service fetches the customer's phone number from the database and sends an SMS using the SMS gateway.

# 5. Deployment Architecture

**Deployment Diagram:** The system is deployed in a cloud environment (e.g., AWS) using scalable and resilient infrastructure components. The web application is deployed on virtual machines or containers, while the SMS gateway service is accessed via an API.

**Environment Configuration:** Configuration scripts automate the setup of development, testing, and production environments, ensuring consistency across deployments.

# 6. Integration Points

**SMS Gateway Integration:** The system integrates with an external SMS gateway service for sending SMS notifications to customers. Integration is achieved via an API provided by the SMS gateway provider.

**Order Management System Integration (Future):** Future integration with the restaurant's order management system will automate order entry into the Order Notification System, enhancing operational efficiency.

# 7. Data Flow Diagram

The data flow diagram illustrates the flow of data through the system, depicting how customer details are entered, processed, and used to trigger notifications.

# 8. Assumptions and Constraints

## Assumptions include:

* + Availability of internet connectivity for accessing the web interface.
  + Reliability of the SMS gateway service for delivering notifications.

## Constraints include:

* + Adherence to data privacy regulations when handling customer information.
  + Compatibility with the existing technology stack used by the restaurant.

# 9. Glossary

## Some key terms and acronyms related to this project:

1. **POS:** Point of Sale - The system used by restaurants to manage sales transactions, including order taking, payment processing, and inventory management.
2. **SMS:** Short Message Service - A text messaging service component of most telephone, internet, and mobile device systems.
3. **API:** Application Programming Interface - A set of rules and protocols that allows different software applications to communicate with each other.
4. **UI:** User Interface - The visual elements and interactive components of a software application through which users interact with the system.
5. **UX:** User Experience - The overall experience of a user when interacting with a product or system, including usability, accessibility, and satisfaction.
6. **DB:** Database - A structured set of data organized and stored electronically in a computer system.
7. **JWT:** JSON Web Token - A compact, URL-safe means of representing claims to be transferred between two parties.
8. **ORM:** Object-Relational Mapping - A programming technique for converting data between incompatible type systems using object-oriented programming languages.
9. **CI/CD:** Continuous Integration/Continuous Deployment - A set of practices and tools that enable teams to automate the process of integrating code changes and deploying applications.
10. **SRS:** Software Requirements Specification - A document that describes the requirements and specifications of a software system.
11. **SDD:** Software Design Document - A document that describes the architecture and design of a software system.

.