**Software Requirements Specification (SRS) for \*\*Pizza Hut Management System\*\* using PostgreSQL**

Table of Contents

1. \*\*Introduction\*\*

1. Purpose

2. Scope

3. Definitions, Acronyms, and Abbreviations

4. Overview

2. \*\*Overall Description\*\*

1. Product Perspective

2. Product Features

3. User Classes and Characteristics

4. Operating Environment

5. Design and Implementation Constraints

6. Assumptions and Dependencies

3. \*\*System Features\*\*

1. User Management

2. Menu Management

3. Order Management

4. Customer Management

5. Delivery and Pickup Management

6. Reporting and Analytics

4. \*\*External Interface Requirements\*\*

1. User Interfaces

2. Hardware Interfaces

3. Software Interfaces

4. Communication Interfaces

5. \*\*System Requirements\*\*

1. Functional Requirements

2. Non-Functional Requirements

6. \*\*Database Design\*\*

1. Database Tables

2. ER Diagram

---

### 1. \*\*Introduction\*\*

#### 1.1 Purpose

This document specifies the software requirements for the Pizza Hut Management System using PostgreSQL as the backend database. The system will help automate and manage orders, menu items, customers, and deliveries. It aims to improve service efficiency, track orders, and generate business reports.

#### 1.2 Scope

The Pizza Hut Management System will include functionalities for:

- Managing the pizza menu, including different categories like pizza, drinks, desserts, etc.

- Handling customer orders (dine-in, takeout, and delivery).

- Customer management, including contact details and order history.

- Reporting and analytics for sales, customer trends, and inventory.

- Delivery and pickup status tracking.

The system will be built using PostgreSQL to manage the database and will be accessible to restaurant staff, delivery personnel, and managers.

#### 1.3 Definitions, Acronyms, and Abbreviations

- \*\*SRS\*\*: Software Requirements Specification

- \*\*DBMS\*\*: Database Management System

- \*\*SQL\*\*: Structured Query Language

#### 1.4 Overview

This SRS document provides detailed specifications of the Pizza Hut Management System. It covers system features, functional and non-functional requirements, database design, and external interfaces.

---

### 2. \*\*Overall Description\*\*

#### 2.1 Product Perspective

The Pizza Hut Management System will be an independent application developed to streamline the restaurant’s operations. It will be implemented using a PostgreSQL database to store and manage data efficiently. The system will support role-based access for different types of users, such as managers, waitstaff, and delivery personnel.

#### 2.2 Product Features

Key features of the system include:

- \*\*Menu Management\*\*: Add, update, and delete menu items.

- \*\*Order Management\*\*: Manage customer orders, generate bills, and track delivery.

- \*\*Customer Management\*\*: Store customer details, including order history and contact information.

- \*\*Delivery and Pickup Tracking\*\*: Monitor the status of delivery orders.

- \*\*Reporting\*\*: Generate reports on sales, popular menu items, and customer behavior.

#### 2.3 User Classes and Characteristics

- \*\*Admin/Manager\*\*: Responsible for managing the system, menu items, users, and viewing reports.

- \*\*Waitstaff\*\*: Handles dine-in orders and processes bills.

- \*\*Delivery Staff\*\*: Responsible for tracking delivery orders and updating order status.

- \*\*Customer\*\*: Can view the menu, place orders, and track deliveries (if integrated with a mobile app).

#### 2.4 Operating Environment

- \*\*Operating System\*\*: Cross-platform (Windows, Linux, or macOS).

- \*\*Database\*\*: PostgreSQL.

- \*\*Server\*\*: Apache/Nginx (for web-based applications).

#### 2.5 Design and Implementation Constraints

- The system will use PostgreSQL as the primary database.

- It must support concurrent access by multiple users with role-based access control.

- Data integrity and security must be enforced, especially for customer data and payment details.

#### 2.6 Assumptions and Dependencies

- Internet access may be required for order tracking and customer interaction if integrated with mobile apps.

- The system assumes that all users have basic knowledge of handling digital order management systems.

---

### 3. \*\*System Features\*\*

#### 3.1 User Management

- \*\*Description\*\*: This module allows the admin to manage users of the system, including managers, waitstaff, and delivery personnel.

- \*\*Functionalities\*\*:

- Add new users, assign roles, and update or delete user information.

- Set role-based access permissions.

#### 3.2 Menu Management

- \*\*Description\*\*: This module allows administrators to manage the pizza menu.

- \*\*Functionalities\*\*:

- Add, update, and delete menu items.

- Define item categories (e.g., pizzas, sides, drinks, desserts).

- Set pricing and availability of menu items.

#### 3.3 Order Management

- \*\*Description\*\*: This module manages customer orders for dine-in, takeout, and delivery.

- \*\*Functionalities\*\*:

- Create new orders and assign them to tables (dine-in) or customers (takeout/delivery).

- Update order status (e.g., prepared, in delivery).

- Generate invoices and process payments.

#### 3.4 Customer Management

- \*\*Description\*\*: This module stores customer details and order history.

- \*\*Functionalities\*\*:

- Add and update customer information (name, contact, address).

- Track order history for loyalty programs and personalized service.

#### 3.5 Delivery and Pickup Management

- \*\*Description\*\*: This module tracks the status of delivery and pickup orders.

- \*\*Functionalities\*\*:

- Assign orders to delivery personnel.

- Update delivery status (e.g., out for delivery, delivered).

- Notify customers about the status of their order.

#### 3.6 Reporting and Analytics

- \*\*Description\*\*: This module generates reports for business insights.

- \*\*Functionalities\*\*:

- View sales reports (daily, monthly, yearly).

- Generate reports on the most popular menu items.

- Analyze customer trends and purchasing behavior.

---

### 4. \*\*External Interface Requirements\*\*

#### 4.1 User Interfaces

The system will provide the following interfaces:

- A dashboard for users (admins, waitstaff, delivery staff) to access features according to their roles.

- A form-based interface for menu management and order placement.

- A table-view interface for viewing orders and reports.

#### 4.2 Hardware Interfaces

- \*\*Printers\*\*: To print invoices and order receipts.

- \*\*POS Terminals\*\*: For payment processing.

#### 4.3 Software Interfaces

- \*\*PostgreSQL Database\*\*: Used to manage data for menu items, orders, customers, and sales.

- \*\*Web or Mobile Frontend\*\*: Interfaces with PostgreSQL to manage the system from a web or mobile application.

#### 4.4 Communication Interfaces

- \*\*HTTP/HTTPS\*\*: Communication between the frontend (if web-based) and the backend server.

---

### 5. \*\*System Requirements\*\*

#### 5.1 Functional Requirements

1. \*\*User Login/Authentication\*\*:

- Users must log in with their credentials, and access is role-based (admin, waitstaff, delivery staff).

- The system should support password recovery for users.

2. \*\*Manage Menu Items\*\*:

- Admins can add, update, and remove items from the menu.

- The system will categorize menu items (e.g., pizzas, drinks, sides).

3. \*\*Manage Orders\*\*:

- Waitstaff can create, update, and view orders.

- Orders must be tracked with real-time status updates for delivery and dine-in.

4. \*\*Customer Data Management\*\*:

- Customer details, including name, contact information, and order history, must be stored.

- Customers can be linked to orders for delivery or takeout.

5. \*\*Track Delivery and Pickup Orders\*\*:

- Assign delivery orders to delivery personnel.

- Update order status and notify customers accordingly.

6. \*\*Generate Reports\*\*:

- Admins can view and export reports on sales, popular items, and customer behavior.

#### 5.2 Non-Functional Requirements

1. \*\*Performance\*\*: The system must be responsive and capable of handling multiple user requests simultaneously.

2. \*\*Security\*\*: User data, especially customer contact and payment details, must be securely stored and encrypted.

3. \*\*Usability\*\*: The interface must be intuitive and accessible to users with minimal training.

4. \*\*Reliability\*\*: The system must be reliable and available during restaurant operation hours with a maximum downtime of 1%.

5. \*\*Scalability\*\*: The system should be scalable to support an increasing number of customers, orders, and menu items as the business grows.

---

### 6. \*\*Database Design\*\*

#### 6.1 Database Tables

1. \*\*Users Table\*\*:

- `user\_id` (Primary Key)

- `username`

- `password`

- `role` (admin, waitstaff, delivery)

2. \*\*Menu Items Table\*\*:

- `item\_id` (Primary Key)

- `item\_name`

- `category` (pizza, drinks, desserts, etc.)

- `price`

- `availability`

3. \*\*Orders Table\*\*:

- `order\_id` (Primary Key)

- `customer\_id` (Foreign Key)

- `total\_amount`

- `order\_date`

- `order\_status` (dine-in, delivery, pickup)

4. \*\*Order Items Table\*\*:

- `order\_item\_id` (Primary Key)

- `order\_id` (Foreign

Key)

- `item\_id` (Foreign Key)

- `quantity`

- `item\_price`

5. \*\*Customers Table\*\*:

- `customer\_id` (Primary Key)

- `name`

- `email`

- `phone`

- `address`

6. \*\*Delivery Table\*\*:

- `delivery\_id` (Primary Key)

- `order\_id` (Foreign Key)

- `delivery\_status`

- `assigned\_to` (Delivery Personnel)

#### 6.2 ER Diagram

The ER diagram represents the relationship between entities such as Users, Menu Items, Orders, Order Items, Customers, and Delivery personnel.

---

This concludes the Software Requirements Specification (SRS) for the Pizza Hut Management System using PostgreSQL.