

Software Requirements Specification (SRS) for Restaurant Management System

1. Introduction

This document provides a detailed description of the **Restaurant Management System** that will be used to manage customer reservations in a restaurant. The system will allow restaurant staff to create new customers, make reservations, check table availability, and cancel reservations.

2. System Overview

The Restaurant Management System (RMS) will help restaurant staff to handle the following tasks:

- **Create Customers:** Staff can add new customers to the system by entering their name and email.
- **Make Reservations:** Customers can book tables for specific dates and times.
- **Check Table Availability:** Staff can check whether a table is available for a particular date.
- **Cancel Reservations:** Reservations can be canceled by entering a reservation ID.

3. Functional Requirements

The system will have the following functional features:

3.1 Create Customer

- **Description:** The system allows staff to create a new customer by entering their name and email.
- **Input:**
 - Name (String)
 - Email (String)
- **Process:** When a new customer is added, the system stores their information in the Customer database table.
- **Output:** A confirmation message is displayed with the customer ID generated by the system.

3.2 Make Reservation

- **Description:** This feature allows staff to make a reservation for a customer at a specific table.
- **Input:**
 - Customer ID (Integer)
 - Table ID (Integer)
 - Reservation Date (String - YYYY-MM-DD)
 - Party Size (Integer)
- **Process:**
 - The system checks if the provided table_id exists.
 - If the table exists, the system checks if it is available on the given date.
 - If the table is available, the system creates a reservation for the customer.
- **Output:** Confirmation message that the reservation was successfully created, or a message indicating that the table is unavailable.

3.3 Check Table Availability

- **Description:** The system allows staff to check if a particular table is available for a specific date.
- **Input:**
 - Table ID (Integer)
 - Reservation Date (String - YYYY-MM-DD)
- **Process:** The system queries the database to check if the table has any existing reservations on the specified date.
- **Output:** A message stating whether the table is available or not.

3.4 Cancel Reservation

- **Description:** Staff can cancel a reservation by providing the reservation ID.
- **Input:**
 - Reservation ID (Integer)
- **Process:** The system deletes the reservation entry from the Reservation table based on the provided ID.
- **Output:** A confirmation message that the reservation has been canceled.

4. Non-Functional Requirements

4.1 Usability

- The system should have a simple, user-friendly interface where staff can easily perform all tasks related to customer management and reservations.

4.2 Reliability

- The system should ensure that all customer data and reservations are saved properly in the database.
- It should handle errors gracefully, providing informative error messages when necessary (e.g., if a table does not exist or a reservation cannot be made).

4.3 Performance

- The system should be able to handle multiple simultaneous requests for creating customers, checking availability, making reservations, and canceling reservations.

4.4 Security

- The system must ensure that customer data (name, email) is securely stored.
- Only authorized staff members should have access to create customers or make reservations.

5. Database Design

The following tables will be used to store data in the system:

5.1 Customer Table

This table stores the details of the customers.

Field	Type	Description
customer_id	INT	Unique ID for each customer
name	VARCHAR(255)	Name of the customer
email	VARCHAR(255)	Email address of the customer

5.2 Reservation Table

This table stores details about reservations.

Field	Type	Description
reservation_id	INT	Unique ID for each reservation
customer_id	INT	ID of the customer making the reservation

Field	Type	Description
table_id	INT	ID of the table being reserved
reservation_date	DATE	Date of the reservation
party_size	INT	Number of people for the reservation

5.3 RestaurantTable Table

This table stores information about restaurant tables.

Field	Type	Description
table_id	INT	Unique ID for each table
capacity	INT	Seating capacity of the table

6. User Interface

The system will operate through a command-line interface (CLI) with the following options:

- **1. Create Customer:** Prompts the user to enter the customer's name and email.
- **2. Make Reservation:** Prompts the user to enter the customer ID, table ID, reservation date, and party size.
- **3. Check Table Availability:** Prompts the user to enter the table ID and reservation date to check availability.
- **4. Cancel Reservation:** Prompts the user to enter the reservation ID to cancel the reservation.
- **5. Exit:** Exits the system.

7. System Architecture

The system will follow a simple client-server architecture:

- The **client** side interacts with the user through a CLI and sends requests to the server.
- The **server** side (back-end) will interact with a MySQL database, where all customer and reservation data will be stored.

8. Conclusion

The **Restaurant Management System** will efficiently handle customer creation, reservations, and table availability checking. It ensures ease of use, reliable data

management, and secure processing of customer information. This system will help the restaurant staff manage reservations seamlessly and avoid overbooking.