

Database Design Document - Online Appointment Management System

1. Introduction

1.1 Purpose

This document defines the functional and non-functional requirements of the Online Reservation Management System. The system will provide offline and online platform for customers to make and book tables, while allowing the restaurant staff to manage reservations effectively.

1.2 Scope

The system aims to make reservations easy, enhance customer experience, optimize restaurant data management through automation. Key functionalities will include:

- Online and offline reservation management
- Order Accuracy
- Table tracking and management
- Reservation modifications and cancellations
- Pick-up Service
- Administrative dashboard for restaurant staff
- Reservation reminder through messages or emails

1.3 Target Audience

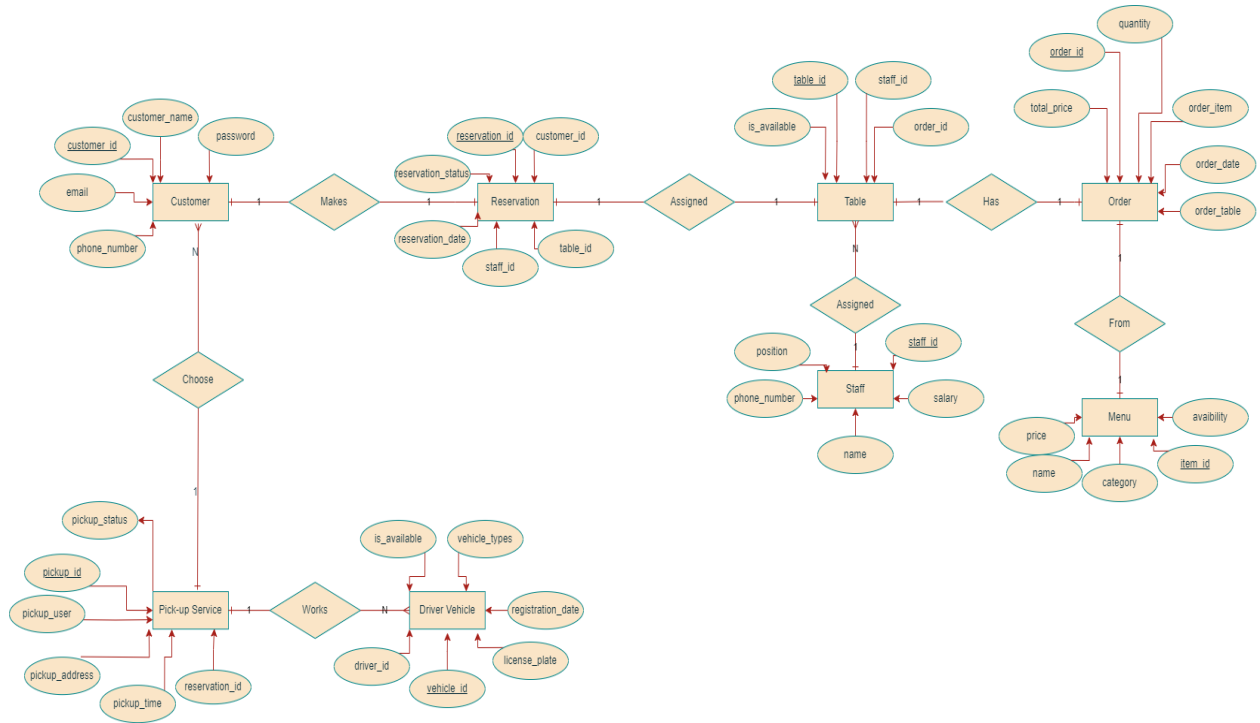
- Software development team
- Test engineers
- Product managers
- Stakeholders (Restaurant Owners & Managers, Restaurant Staff, Customers)

2. Database Architecture

2.1 Database Management System

- **Type:** Relational Database Management System (RDBMS)
- **Preferred DBMS:** MySQL
- **Normalization:** Third Normal Form (3NF) to optimize storage and eliminate redundancy

2.2 ER Diagram



3. Database Schema

3.1 Customers Table

Column	Type	Constraints	Description
customer_id	INT	AUTO_INCREMENT	Unique user identifier
customer_name	VARCHAR(200)	NOT NULL	Full name
phone_number	VARCHAR(50)	NOT NULL UNIQUE	Customer phone number
email	VARCHAR(100)	NOT NULL UNIQUE	Customer email
password	VARCHAR(250)	NOT NULL	Hashed password

3.2 Pickup_service Table

Column	Type	Constraints	Description
pickup_id	INT	AUTO_INCREMENT	Unique pickup identifier
pickup_user	INT	REFERENCES customers(customer_id)	Customer ID
pickup_status	BOOLEAN	DEFAULT FALSE	Pickup status
pickup_address	VARCHAR(250)	NOT NULL	Pickup address
pickup_time	DATETIME	NOT NULL	Pickup time

3.3 Reservation Table

Column	Type	Constraints	Description
reservation_id	INT	AUTO_INCREMENT	Unique reservation identifier
reservation_status	ENUM	DEFAULT ACCEPTED	Reservation status
customer_id	INT	REFERENCES customer(customer_id)	Customer id
table_id	INT	REFERENCES table(table_id)	Table id
reservation_date	DATETIME	NOT NULL	Reservation date

3.4 Vehicles Table

Column	Type	Constraints	Description
vehicle_id	INT	AUTO_INCREMENT	Unique vehicle identifier
driver_id	INT	REFERENCES staff(staff_id)	Staff ID
vehicle_type	VARCHAR(100)	NOT NULL	Vehicle type
licence_plate	VARCHAR(50)	NOT NULL UNIQUE	Licence plate
registration_date	DATE	NOT NULL	Registration date
is_available	BOOLEAN	DEFAULT TRUE	Vehicle's availability

3.5 Tables Table

Column	Type	Constraints	Description
table_id	INT	AUTO_INCREMENT	Unique table identifier
is_available	BOOLEAN	DEFAULT TRUE	Availability of table
staff_id	INT	REFERENCES staff(staff_id)	Staff ID
order_id	INT	REFERENCES order(order_id)	Order ID

3.6 Menu Table

Column	Type	Constraints	Description
item_id	INT	AUTO_INCREMENT	Unique item identifier
category	VARCHAR(50)	NOT NULL	Item category
Name	VARCHAR(100)	NOT NULL UNIQUE	Item name
price	DECIMAL(10,2)	NOT NULL	Item price
Availability	BOOLEAN	DEFAULT TRUE	Availability of

			item
--	--	--	------

3.7 Staff Table

Column	Type	Constraints	Description
staff_id	INT	AUTO_INCREMENT	Unique staff identifier
name	VARCHAR(200)	NOT NULL	Name
position	VARCHAR(150)	NOT NULL	Position
phone_number	VARCHAR(50)	NOT NULL UNIQUE	Phone number
salary	DECIMAL(10,2)	NOT NULL	Salary

3.8 Order Table

Column	Type	Constraints	Description
order_id	INT	AUTO_INCREMENT	Unique order identifier
order_table	INT	REFERENCES tables(table_id)	Order table
order_item	INT	REFERENCES menu(item_id)	Order item
quantity	INT	DEFAULT 1	Quantity
total_price	DECIMAL(10,2)	DEFAULT 0	Total price
order_date	DATETIME	DEFAULT CURRENT_TIMESTAMP	Order date

4. Indexing Strategy

- **Primary Keys:** Used on `id` fields to ensure uniqueness and fast lookups.
- **Foreign Keys:** Enforce referential integrity between tables.
- **Indexes:**
 - **Index on `email` in Customers Table** (for faster login lookups).
 - **Index on `customer_id` in Reservation Table** (for retrieving

customer's reservation quickly).

- **Index on `date` in Reservation Table** (for efficient date-based queries).

5. Security Measures

- **Data Encryption:**
 - Passwords stored using bcrypt hashing.
 - Sensitive data encrypted using AES-256.
- **Access Control:**
 - Role-based access control (RBAC) to restrict unauthorized actions.
- **Audit Logging:**
 - All user actions logged for monitoring and security.

6. Scalability & Performance Optimization

- **Read Replicas:** Database replication for load balancing.
- **Caching Mechanism:** Implement Redis or Memcached for frequent queries.
- **Partitioning Strategy:** Time-based partitioning for logs and history tables.
- **Connection Pooling:** Optimized DB connections for performance.

7. Backup & Disaster Recovery Plan

- **Database Backups:** Daily full backups, hourly incremental backups.
- **File Storage Backups:** Version-controlled backups of static assets.
- **Failover Strategy:** Multi-region deployment for high availability.
- **Recovery Time Objective (RTO):** Less than 15 minutes.
- **Recovery Point Objective (RPO):** Less than 5 minutes.

8. Conclusion

This database design document provides a robust foundation for the Online Restaurant Reservation Management System. It ensures scalability, security, and efficiency while maintaining data integrity and accessibility.