SYMBIOSIS
INSTITUTE OF TECHNOLOGY
॥वसुधैव कुटुम्बकम्॥

Database Management Report

Restaurant Management System (RMS)

RESTAURANT

Foodies Palace Co.

Submitted by:

S.No.	Name	Prn
1	Mayank Sahai	22070122115
2	Dhairya Mehra	22070122118
3	Nakul Kushwaha	22070122130

Under the guidance of:

Prof. Pooja Bagane

Department of Computer Science

SYMBIOSIS INSTITUTE OF TECHNOLOGY, PUNE

Index

Introduction

Effective management is the backbone of a successful and colourful restaurant industry world where flavour meets flair and hospitality is utmost. Constantly adapting to new consumer needs and rising expectations, restaurateurs have to cope with an increasingly hard situation. As different as the food on their menu, restaurant owners and managers must be organizers, efficacious, technologically aware, money conscious, and masters of physical and human things.

In this context, RMS stands out as the guiding light of innovation, providing a revolutionary approach that changes the way restaurants conduct and succeed in business. Emerging out of a marriage between advanced technology and an intimate knowledge of the inner workings of restaurant management, the RMS, which is slated to overhaul all aspects of the dining out experience, from booking to bill, promises truly exciting times for restaurateurs.

Problem Statement

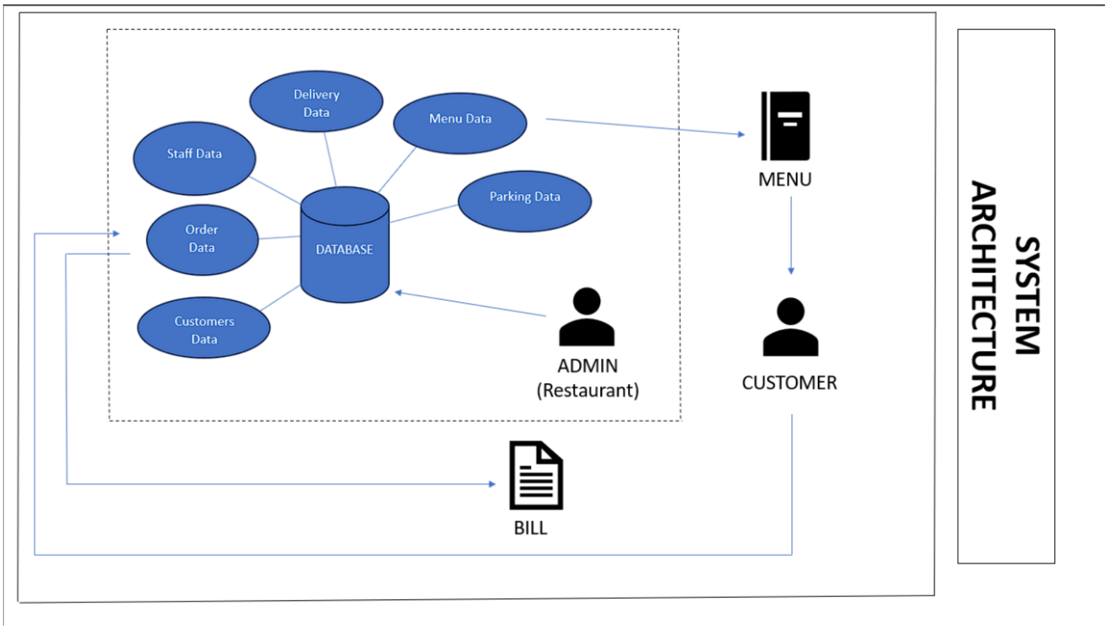
RMS- Restaurant Management System

In the world of restaurant ownership, it is common to feel like a walking, talking culinary highwire act, juggling constantly to keep the various elements of the modern day dining establishment in balance. From the careful choreography of reservation of tables and arrangement of seating to the detailed orchestration of orders and supply, the obstacles are many and complex.

The classic restaurant management practices, based on manual procedures and isolated systems, frequently fail to meet these challenges. Inaccurate inventory counts, overbooked tables, fragmented communication, and poor service are just some of the traps that restaurateurs are mired in, eroding efficiency, profitability, and customer satisfaction.

It has never been more urgent than now to have a holistic solution that is technology driven working on both the front of the house and the back of the house, to empower staff, to delight customers, and to drive growth. The RMS is ready to introduce a new era of information technology in the restaurant space by providing a global, easy to use, and transformational platform that allows restaurateurs to improve every aspect of their business and provide customers with unforgettable dining experiences.

System Architecture



Modules

1. **Customer Management:** Tools that enable the management of customer profiles, preferences and feedback.

2. **Reservation Management:** It handles reservation bookings, cancellations, and status updates.

3. **Order Management:** On the customer side, it helps in placing order, changing one already placed, and it is easy to track the order.

4. **Menu Item Management:** It handles the menu of the restaurant, which involves item specifications, prices, and availability.

5. **Staff Management:** Handle staff profiles and chefs profiles, their roles, schedules, and performance appraisals.

6. **Delivery Management:** The coordinates delivery orders, addresses, and delivery status.

7. **Feedback Management:** The firm collects and analyzes customer feedback in a bid to enhance service quality.

8. **Order Status:** Each order, either delivered or picked up, is also tracked in terms of status.

9. **Parking Management:** The system tracks vehicle data, in and out schedules of the customers vehicle.

Functional Requirement

User Authentication and Authorization:

1. Allow staff and customers to create accounts with unique usernames and passwords.
2. Implement password encryption and secure storage mechanisms.

Customer Interaction:

1. Allow customers to register accounts or place orders as guests and book a reservation with date and time.
2. Enable customers to browse the menu, add items to their cart, and customize orders and let them choose the option for delivery or in-house.
3. Implement feedback forms or rating systems for customers to provide reviews and suggestions.

Reservation Management:

1. Display a calendar view for staff to see existing reservations and available time slots.
2. Enable staff to cancel reservations with proper notification to customers.

Order Processing:

1. Develop an order management system to track orders from placement to delivery.
2. Enable staff to assign orders to specific tables or dining areas.
3. Implement communication channels between staff and kitchen for order coordination.

4. Provide real time updates for staff on order readiness and delivery status.

Feedback Analysis:

1. Collect feedback from customers through surveys, ratings, or comments.
2. Generate reports or visualizations to present feedback analysis to management and staff.
3. Implement mechanisms to respond to customer feedback and address any issues or concerns raised.

Reporting and Analytics:

1. Generate reports on sales performance, including revenue, profit margins, and popular menu items.
2. Analyse customer feedback to identify trends, satisfaction levels, and areas for improvement.
3. Present reports and analytics in a user friendly format, such as charts, graphs, and dashboards.

Entities, Relationships and Attributes

Entities and Relations

Entities	Relationships	Entities
Customer	Can Book	Reservations
Customer	Parks Vehicle in	Parking Slots
Customer	Provides	Feedback
Reservation	Has	Orders
Order	Has	Order_Status
Order	Contains	Menu_item
Chef	Prepares	Order
Chef	Part Of	Staff

Attributes

1. **MenuItem**
    a. Item_ID
    b. Item_name
    c. Price
    d. Category
2. **Order**
    a. Order_ID
    b. Table_no

c. Order_Date_time

d. Total_Amount

e. Payment_status

f. Order_Mode

3. **Customer**
   a. Customer_ID
   b. Name
   c. PhoneNo
   d. Email
4. **Reservation**
   a. Reservation_ID
   b. Noofguest
   c. DateTime
5. **Feedback**
   a. Feedback_ID
   b. Ratings
   c. DateTime
   d. Comments
6. **Parking_Slots**
   a. Slot_No
   b. Vehicle_No
   c. StartTimeDate
   d. EndTimeDate
7. **Order_Status**
   a. Status_ID
   b. Status_Name
8. **Chef**
   a. Chef_ID
   b. Name

    c. Specialization

9. **Staff**

    a. Staff_ID

    b. Name

    c. Role

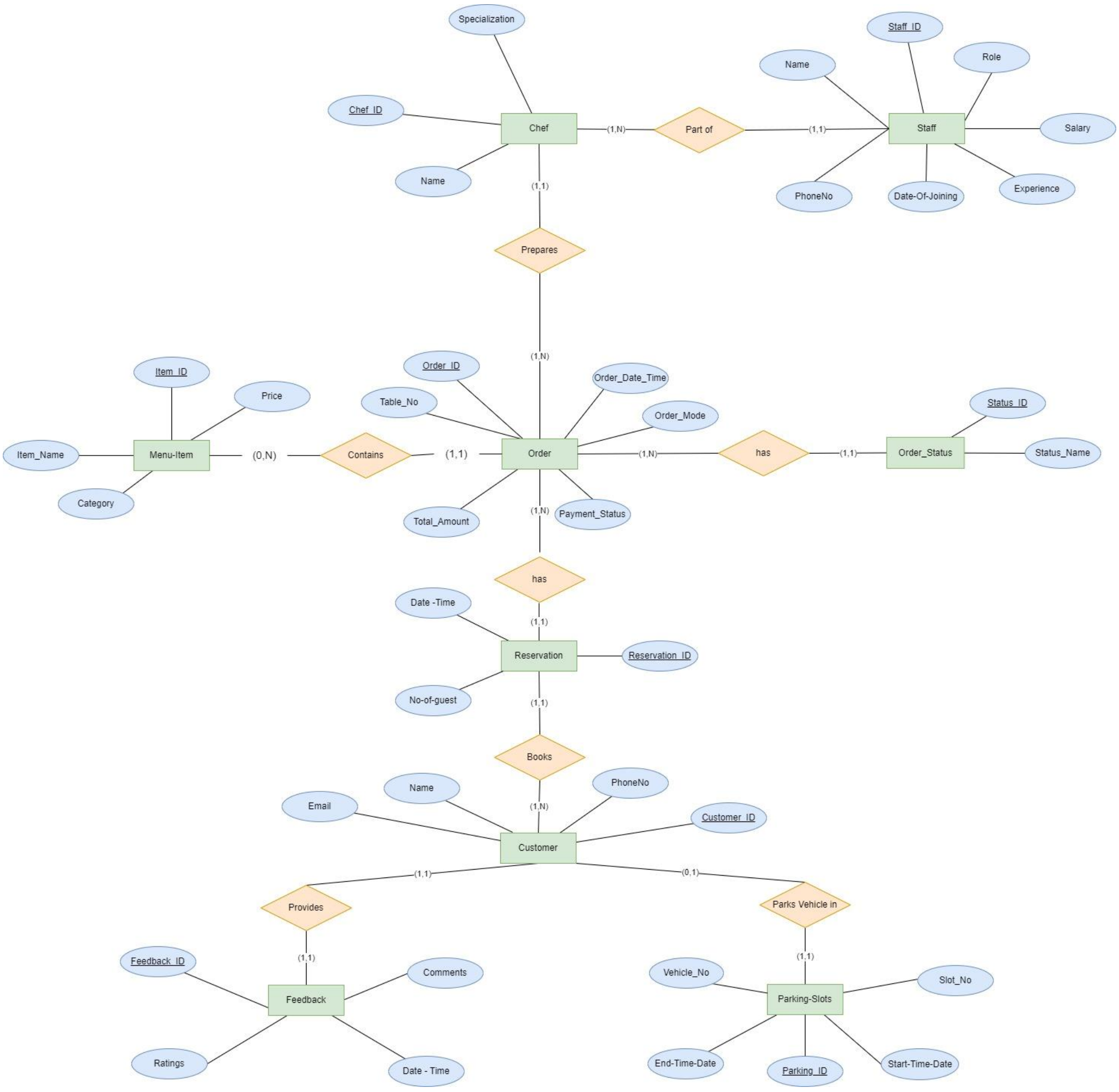
    d. PhoneNo

    e. Salary

    f. DateofJoining

    g. Experience

ER Diagram



Specialization
Staff_ID
Role
Name
Chef_ID
Chef
(1,N)
Part of
(1,1)
Staff
Salary
Name
(1,1)
PhoneNo
Date-Of-Joining
Experience
Prepares
(1,N)
Item_ID
Order_ID
Order_Date_Time
Price
Table_No
Order_Mode
Status_ID
Item_Name
Menu-Item
(0,N)
Contains
(1,1)
Order
(1,N)
has
(1,1)
Order_Status
Status_Name
Category
Total_Amount
(1,N)
Payment_Status
has
Date -Time
(1,1)
Reservation
Reservation_ID
No-of-guest
(1,1)
Books
Name
PhoneNo
Email
(1,N)
Customer_ID
Customer
(1,1)
(0,1)
Provides
Parks Vehicle in
Feedback_ID
(1,1)
(1,1)
Comments
Vehicle_No
Slot_No
Feedback
Parking-Slots
Ratings
Date - Time
End-Time-Date
Parking_ID
Start-Time-Date

Relational Schema

1. **MenuItem** (Item_ID (PK) , Item_name , Price , Category)

Item_ID (PK)	Item_Name	Price	Category

- **Primary Key:** Item_ID
- **Foreign Keys:** Null
- **Candidate Keys:** Item_Name,Item_ID
- **Alternate Keys:** Item_Name

2. **Order** (Order_ID (PK) , Item_ID (FK), Status_ID (FK), Reservation_ID (FK), Chef_ID (FK), Table_No, order_date_time, Total_Amount, Payment_status, Order_Mode)

Order_ID (PK)	Item_ID (FK)	Status_ID (FK)	Reservation_ID (FK)	Chef_ID (FK)	Table_No	Order_date_time	Total_Amount	Payment_status	Order_Mode

- **Primary Key:** Order_ID
- **Foreign Keys:** Item_ID, Status_ID, Reservation_ID, Chef_ID
- **Candidate Keys: (**Order_date_time,Table_No), Order_ID
- **Alternate Keys: (**Order_date_time,Table_No)

3. **Customer** (Customer_ID (PK), Name , PhoneNo , Email)

Customer_ID (PK)	Name	PhoneNo	Email

- **Primary Key:** Customer_ID
- **Foreign Keys:** None
- **Candidate Keys:** PhoneNo, Email, Customer_ID
- **Alternate Keys:** PhoneNo, Email

4. **Reservation** (Reservation_ID (PK) , Customer_ID (FK), Order_ID (FK), Noofguest , DateTime)

Reservatoin_ID (PK)	Customer_ID (FK)	Order_ID (FK)	Noofguest	DateTime

- **Primary Key:** Reservation_ID
- **Foreign Keys:** Customer_ID, Order_ID
- **Candidate Keys:** Reservation_ID
- **Alternate Keys:** Null

5. **Feedback** (Feedback_ID (PK) , Customer_ID (FK), Ratings , DateTime , Comments)

Feedback_ID (PK)	Customer_ID (FK)	Ratings	DateTime	Comments

- **Primary Key:** Feedback_ID
- **Foreign Keys:** Customer_ID
- **Candidate Keys:** Feedback_ID
- **Alternate Keys:** Null

6. **Parking_Slots** (Parking_ID (PK) , Slot_No , Customer_ID (FK), Vehicle_No , StartTimeDate , EndTimeDate)

Parking_ID (PK)	Slot_No	Customer_ID (FK)	Vehicle_No	StartTimeDate

- **Primary Key:** Parking_ID
- **Foreign Keys:** Customer_ID
- **Candidate Keys:** Parking_ID , (Slot_No , StartTimeDate)
- **Alternate Keys:** (Slot_No , StartTimeDate)

7. **Order_Status** (Status_ID (PK), Order_ID (FK), Status_Name)

Status_ID (PK)	Order_ID (FK)	Status_Name

- **Primary Key:** Status_ID
- **Foreign Keys:** Order_ID

- **Candidate Keys:** Status_ID
- **Alternate Keys:** Null

8. **Chef** (Chef_ID (PK), Order_ID (FK), Name , Specialization)

Chef_ID (PK)	Order_ID (FK)	Name	Specialization

- **Primary Key:** Chef_ID
- **Foreign Keys:** Order_ID
- **Candidate Keys:** Chef_ID
- **Alternate Keys:** Null

9. **Staff** (Staff_ID (PK) , Chef_ID(FK), Name , Role , PhoneNo , Salary, Dateofjoining , Experience)

Staff_ID (PK)	Chef_ID (FK)	Name	Role	PhoneNo	Salary	Dateofjoining	Experience

- **Primary Key:** Staff_ID
- **Foreign Keys:** Chef_ID
- **Candidate Keys:** Staff_ID , PhoneNo
- **Alternate Keys:** PhoneNo

Application of Codd’s Rules

1. **Information Rule:** All data must be accessible via the relational table structure.

- **Applicability:** Yes, applicable to ensure data accessibility and organization within the system.

2. **Guaranteed Access Rule:** Each data item must be accessible by specifying a table name, primary key, and column name.

- **Applicability:** Yes, ensures consistent and precise data retrieval within the system.

3. **Systematic Treatment of Null Values:** Null values must be treated consistently, rather than being interpreted in multiple ways.

- **Applicability:** Yes, relevant for maintaining data consistency and accuracy in the system.

4. **Dynamic Online Catalogue Based on the Relational Model:** The database catalogue (metadata) must be stored and accessible like other data.

- **Applicability**: Yes, relevant for maintaining an organized and accessible database structure.

5. **Comprehensive Data Sublanguage Rule:** The system must support a comprehensive language for data definition, manipulation, and access.

- **Applicability:** Yes, ensures the availability of a robust data language for managing the system's data.

6. **View Updating Rule:** Any view that is theoretically updatable should also be updatable by the system.

- **Applicability:** Partially applicable, as the system may involve views, but the requirement for updating views is not explicitly mentioned.

7. **High level Insert, Update, and Delete:** The system must support high level operations for inserting, updating, and deleting data.

- **Applicability:** Yes, relevant for ensuring efficient data manipulation within the system.

8. **Physical Data Independence:** Changes in physical storage should not impact the logical structure of the database.

- **Applicability:** Partially applicable, as the system may not fully support physical data independence due to potential impacts on the logical structure.

9. **Logical Data Independence:** Changes in the logical structure of the database should not require changes to the application's programs.

- **Applicability:** Partially applicable, as the system may not fully support logical data independence due to potential impacts on physical storage.

10. **Integrity Independence:** Integrity constraints should be stored in the catalog and not in application programs.

- **Applicability:** Yes, relevant for maintaining data integrity and consistency within the system.

11. **Distribution Independence:** The system should not require changes to the logical structure of the database when distributed to multiple locations.

- **Applicability:** Not applicable, as the system does not involve distributing portions of the database to multiple locations.

12. **Non subversion Rule:** If the system provides low level access to data, it must also provide high level access to the same data to prevent bypassing security controls.

- **Applicability:** Yes, relevant for ensuring data security and preventing subversion of access controls within the system.