

Café Management System

SG Team

1

Sujal | Devpreet | Yubraj

Table of Contents

[Executive Summary 2](#_Toc195005462)

[Design Analysis Process 3](#_Toc195005463)

[Actors 3](#_Toc195005464)

[Software Development Life Cycle (SDLC) 4](#_Toc195005465)

[User Stories 5](#_Toc195005466)

[Uses Cases 6](#_Toc195005467)

[Use Case Diagram 7](#_Toc195005468)

[Functional and Non – Functional Requirements 8](#_Toc195005469)

[Evidence from Brainstorming 8](#_Toc195005470)

[Entity Relationship Diagram 9](#_Toc195005471)

[Table Designs – Data Dictionary 11](#_Toc195005472)

[Contributions 17](#_Toc195005473)

[References 17](#_Toc195005474)

# Executive Summary

The Café Management System is a structured and user-friendly system developed to manage the operations of café. It covers essential functions such as handling customer details, managing employee roles, process order, collecting customer feedback, menu handling and recording payments.  
  
Built Using SQLite3, the system includes multiple interconnected such as Customer, Orders, Menu Items, Tables, Employees and Payments. Customers can place orders that are linked to specific tables and handled by assigned employees. Each order includes selected menu items with custom instructions, quantity, and pricing.

The system also tracks table availability and customer count, supports multiple payment methods, and stores payment records securely. Menu items are categorized for easy selection, and a feedback module allows customers to rate their experience, helping improve service quality.

Also, this system is an MVP (Minimum Viable Product) of the whole café operations in real – life so it does not cover every aspect of café managed in real – world.

Overall, this system simplifies café operations, reduces manual work, improves service efficiency, and enhances the customer experience through organized data management.

# Design Analysis Process

In design analysis process of our café management system we will discuss about User Stories, Use Cases, Use Case Diagram, Actors, Software Development Lifecycle, Functional and Non – Functional Requirements, Evidence from Brainstorming and References (if any...).

## Actors

1. **Customer:** Customer is also a key user actor of the café management system because they are responsible for placing order as well as engaging in café’s service. Basically, they look after the menu, select their desire food stuffs, drinks etc and place the order and complete payment through various methods such as cash, card or any other digital wallets. They can also track the order status and further they can also give the feedback through which it helps the café for improvisation on services, food and others.

1. **Owner/Manager:** Basically, owners look after the whole entire operation and manage the system of café. They can add, update, remove the menu items, and adjust the pricing looking after the scenario and look after sales report, customer preferences and peak hours. They mainly manage the staff roles and shift of staff. Through, which whole café system runs more effectively, maintaining good services and high customer satisfaction

1. **Staff:** Staff mainly include waiters, chef, and cashier. They play a crucial role in Cafes daily operation. They take the order and prepared order and once the food is ready, they update the time to take the order and delivered it. They co-ordinate to all the members, track the orders and complete orders as well as reduced errors and improved the service.

## Software Development Life Cycle (SDLC)

We are following the Agile Methodology, using this following 7 – steps:

**Note:** **The Needs stated here is based on the real – time systems as we are making MVP so, we are not responsible for fulfilling all requirements.**

**1.Planning:**

**Need:** The system should efficiently manage customers, orders, payments, employees and menu items.

**Approach:** we manage the key features like order processing, table management, payments and planning their development to ensure smooth functionality.

**2.Define Requirements:**

**Need:** The system must support core operations like order handling, menu updates, and customers feedback while maintaining performance.

**Approach:** We define user roles and interactions based on our database structure, ensuring coordination between customers, employees, and orders

**3. Design & Prototyping:**

**Need:** A well-structured database and a natural user interface to simplify order placement, payment processing and employee management.

**Approach:** Using our ERD, we design a logical database structure with clear relationships between customers, orders, menu items, and payments.

**4. Software Development:**

**Need:** A system that allows customers to place orders, employees to manage them, and payments to be processed accurately.

**Approach:** We follow an iterative approach, first implementing core features like order creation and payments, then adding enhancements such as customer feedback.

**5. Testing:**

**Need:** Ensure that all components function correctly and interact smoothly

with each other.

**Approach:** We conduct unit testing on key modules (orders, payments, menu items), to validate relationships between tables. This part will do after we create proper database file.

**6. Deployment:**

**Need:** A stable system rollout that allows real-time feedback and improvements.

**Approach:** We release the system as an MVP of café management further will add more features is needed.

**7. Operations & Maintenance:**

**Need:** Continuous monitoring and updates to enhance performance, security, and user experience.

**Approach:** We track system performance, fix issues as they arise, and introduce upgrades based on customer feedback and business needs.

## User Stories

**1.Customers**

As a customer, I will look after the menu and order it, so that I can get my preferred meal quickly.

As a customer, I want to pay my bill by online payments, so that I can complete my order Fastly and easily.

As a customer, I want to be notified through the email if my order is delayed, so that I can plan accordingly.

As a customer, I want to give feedback and review, so that it helps to improve the café service.

**2.Employee Stories (Waiter, Chef, Cashier)**

As a waiter, I want to assign orders to specific tables so that I can ensure the right customer gets the correct order.

As a waiter, I want to update an order's status to "Delivered" so that I can keep track of completed orders.

As a chef, I want to receive kitchen orders instantly so that I can begin food preparation without delay.

As a chef, I want to see customer special requests (e.g., no sugar, extra cheese) so that I can prepare the food accordingly.

As a cashier, I want to process payments using cash or card so that customers have flexible payment options.

As a cashier, I want to generate digital receipts so that customers receive via email.

**3.Owner Stories (Owner, Manager, Supervisor)**

As a cafe owner, I want to add, update, or remove employee details, so that I can manage staffing and assign roles effectively.

As a supervisor, I want to view all active orders and table occupancy status, so that I can oversee operations and ensure smooth service.

As a manager, I want to access and review all payment transactions, so that I can track revenue and ensure all payments are recorded correctly.

As a supervisor, I want to read customer feedback and ratings, so that I can identify areas for improvement and maintain service quality.

As a cafe owner, I want to generate a summary of daily sales including total orders and income, so that I can assess business performance and make informed decisions.

## Uses Cases

1.View Menu – Customers browse the digital menu categorized by Coffee, Pastries, Snacks, and Desserts.

2.Place Order – Customers select items, specify quantity, and confirm the order.

3.Process Payments – Customers complete payments using Cash or Card.

4.Provide Feedback – Customers rate their experience and leave comments.

5.Manage Employees – Owners add, edit, or remove employee details.

(a) Add Employee

(b) Update Employee Role

(c) Remove Employee

6.Update Menu Items – Owners update the menu by adding new items or changing prices.

(a) Add New Item

(b) Update Item Price

7.Assign Roles – Managers assign roles (Waiter, Chef, Cashier) to employees.

8.Owner - Get the Total Revenue for the Café

9.Owner - Update Ownership Information

10.Customer - Delete a Customer Who Requested Account Removal

11.Order - Get All Orders Along with Their Items

12.Items - Best-Selling Items

13.Order - Track Order Status for a Customer

14.List – List all employees under a specific owner

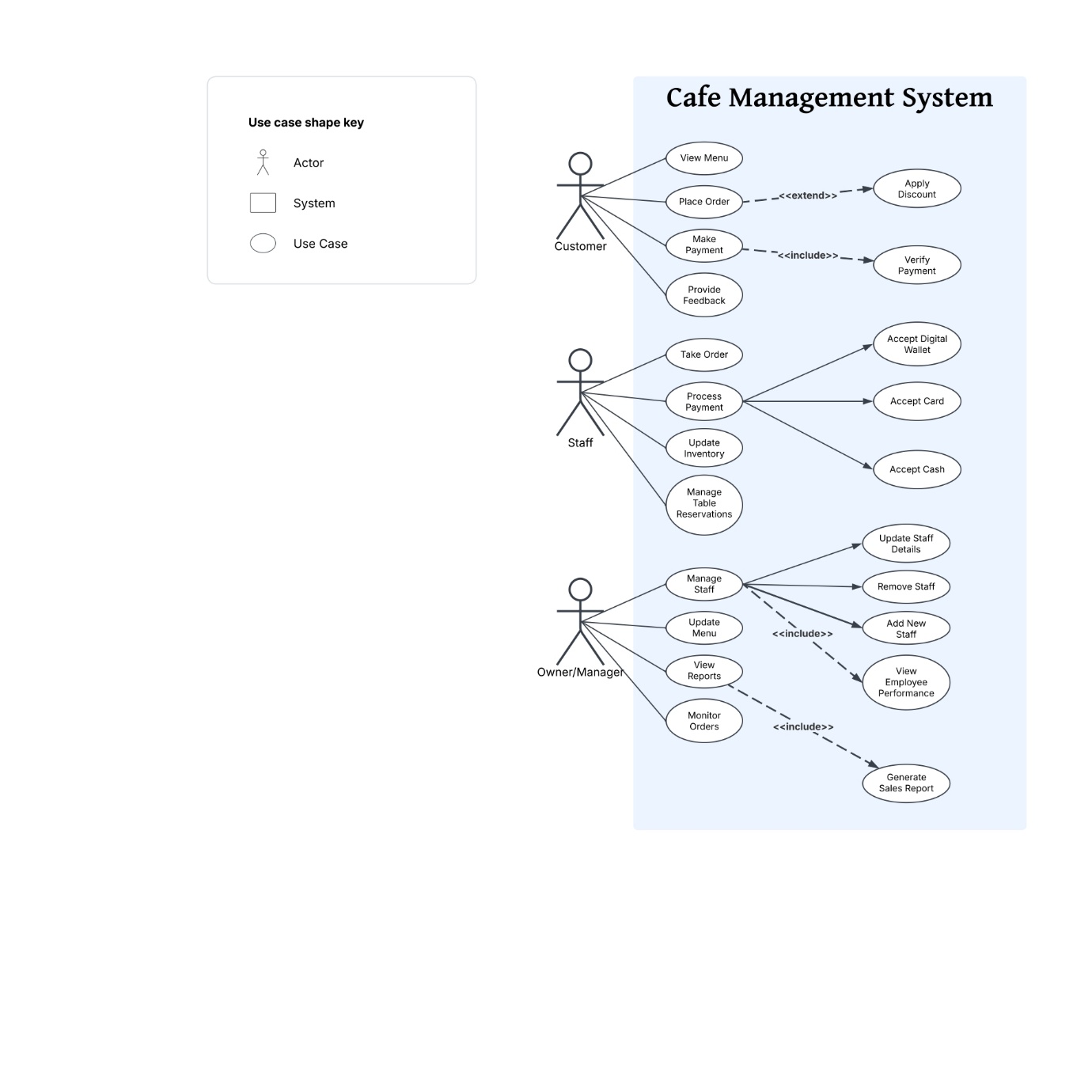
15.Feedback – Find customers who have given a 5 – star rating.

Based on this use cases we have created queries which is attached in queries.txt file.

Also, all queries include the topics that we have learned so far in this studio course listed below:

Where clause, left join, inner join, group by, aliases, update statement, delete statement, insert values etc.

## Use Case Diagram



Based on this Use Case diagram there are use cases given below further implemented to queries.

## Functional and Non – Functional Requirements

The requirements given below is a scope of whole system some requirements are not supported for this MVP.

**Functional Requirements:**

1. Order placement: Browse menu, select items and place order
2. Menu management: Add, edit and delete the menu items as required
3. Inventory management: Track stock levels, receive deliveries, and manage all information. (Not Supported)
4. Table booking: Allow customers to book tables in advance.

**Non-Functional Requirements:**

1. Security: Secure login and password management for staff accounts. (Not Supported)
2. Usability: User-friendly interface for all staff, customer and owner/manager
3. Scalability: Now our café is small but in future the system should be able to handle future growth in terms of users, data, and features.
4. Reliability: Data accuracy and consistency as well as system should be stable and reliable.

## Evidence from Brainstorming

The ERD represents the final structure based on discussions about order processing, payments, and customer interactions.

* Feedback functionality was included after considering customer engagement.
* Employee roles were defined to streamline order and table management.
* Payment methods were diversified based on common cafe transactions.
* Menu categories were structured for ease of use and scalability.
* Reporting features were included after analysing business needs for sales tracking.

# Entity Relationship Diagram

Logical ERD:

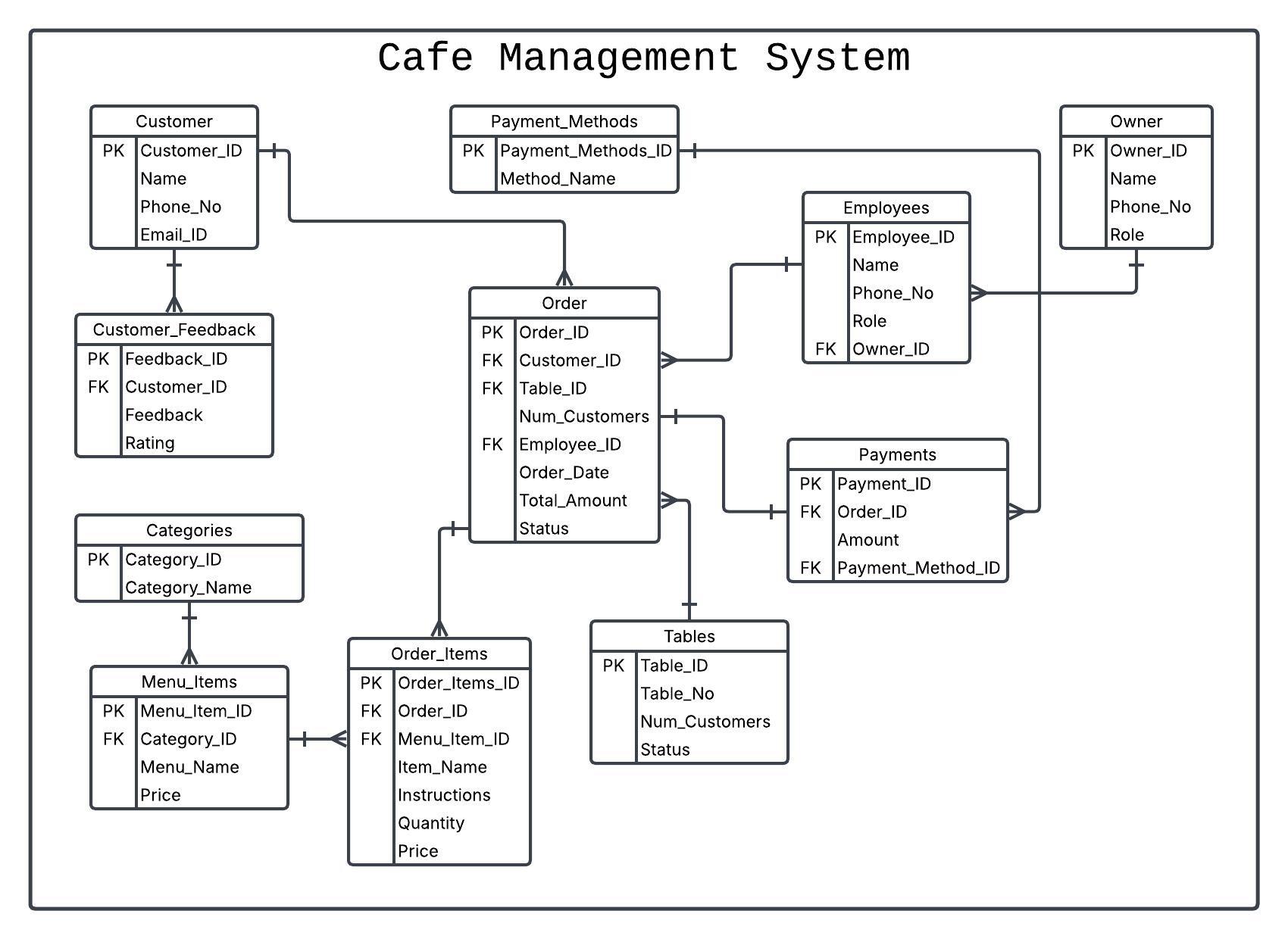
This was our logical approach for café management system which shows a rough Idea and as draft form of our system and after some reforms in this will come up with proper physical diagram which is given below after this.

A diagram of a cafe management system

AI-generated content may be incorrect.

**Physical ERD:**

This is our final physical ER Diagram for café management system consist of total 11 tables which gives a visual representation of my tables and its attributes.



# Table Designs – Data Dictionary

This data dictionary gives a detailed description of table created in our system and their attributes to a proper understanding of our system.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Entity name | Description | | | | |
| Owner | The owner entity includes customer personal information like name, owner id, email and phone number with department information | | | | |
| Field Name | **Description** | **Data type** | **Key field** | **Constraints** | **Example** |
| Owner\_ID | Uniquely Identifies each owner | Int | PK | Not Null, Unique | 01, 02 |
| Name | Full name of each owner | Text |  | Not Null | Jame, Erick |
| Phone\_No | Contact Number of the owner | Text |  | Unique | +64 02040155546 |
| Role | Role of each owner | Text |  |  | Owner, manager |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Entity name** | **Description** | | | | |
| Customer | The customer entity includes customer personal information like name, customer id and phone number with department information | | | | |
| **Field Name** | **Description** | **Data type** | **Key field** | **Constraints** | **Example** |
| Customer\_ID | Uniquely Identifies each Customer | Int | PK | Not Null, Unique | 01 |
| Name | Full Name of the Customer | Text |  | Not Null | Sam |
| Phone\_No | Contact Number of the Customer | Text |  | Unique | +64 02040155547 |
| Email\_ID | |  | | --- | | Email address of the customer |  |  | | --- | |  | | Text |  | Unique | Sam01@gmail.com |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Entity name | Description | | | | |
| Categories | The categories table stores different menu categories such as beverages, desserts, or meals. | | | | |
|  |  | | | | |
| Field Name | **Description** | **Data type** | **Key field** | **Constraints** | **Example** |
| Category\_ID | Unique Category ID | Int | PK | Not Null, Unique | 01 |
| Category\_Name | Menu Category Name | Text |  | Not Null | Coffee |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Entity name | Description | | | | |
| PaymentMethods | the categories table stores different category available | | | | |
|  |  | | | | |
| Field Name | **Description** | **Data type** | **Key field** | **Constraints** | **Example** |
| Payment\_Methods\_ID | Uniquely identifies Payment method | Int | PK | Not Null, Unique | 02 |
| Category\_Name | Name of the payment method | Text |  | Not Null | Card |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Entity name | Description | | | | |
| Tables | The Tables table stores details about seating arrangements and table availability in the cafe. | | | | |
| Field name | **Description** | **Data type** | **Key field** | **Constraints** | **Example** |
| Table\_ID | Uniquely identifies each table | Int | PK | Not null, unique | 10 |
| Table\_No | Table number assigned in the cafe | Text |  | Not null | T5 |
| Num\_Customers | Maximum customers a table can accommodate | Number |  | Not null | 4 |
| Status | Available status of the table | Integer |  |  | Occupied |
| Entity name | **Description** | | | | |
| Employees | This table holds details of employees working in the cafe, including their roles and contact information. | | | | |
| Field Name | **Description** | **Data type** | **Key field** | **Constraints** | **Example** |
| Employee\_ID | Uniquely identifies each employee | Int | PK | Not null, unique | 23 |
| Name | Employee’s name | Text |  | Not null | Emma Watson |
| Phone\_No | Employee’s contact number | Text |  | Unique | +641122334455 |
| Role | Employee’s role in the cafe | Text |  |  | Waiter |
| Owner\_ID | Reference to the owner managing the employee | Integer | FK | Not null | 1 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Entity name | Description | | | | |
| Menu\_Items | This table contains details of items available on the cafe's menu, including category and price. | | | | |
| Field name | **Description** | **Data type** | **Key field** | **Constraints** | **Example** |
| Menu\_Item\_ID | Uniquely Identifies each menu item | Int | PK | Not Null, Unique | 03 |
| Category\_ID | Reference to the category of the menu item | Int | FK | Not Null | 5 |
| Menu\_Name | Name of the menu items | Text |  | Not Null | Cappucino |
| Price | Price of the item | Decimal |  | Not Null | $7.00 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Entity name | Description | | | | |
| Orders | This table tracks orders placed by customers, along with payment status and the employee handling them. | | | | |
| Field Name | **Description** | **Data type** | **Key field** | **Constraints** | **Example** |
| Order\_ID | Uniquely Identifies each order | Int | PK | Not Null | 5001 |
| Customer\_ID | Reference to the customer placing the order | Int | FK | Not Null | 101 |
| Table\_ID | Reference to the table occupied for the order | Int | FK |  | 10 |
| Num\_Customers | Number of customers for the order | int |  |  | 2 |
| Employee\_id | Employee handling the order | Int | FK | Not Null | 23 |
| Order\_Date | Date when the order was placed | Date |  | Not Null | 2025-03-30 |
| Total\_Amount | Total amount of the order | Decimal |  | Not Null | $20.00 |
| Status | Status of the order (pending, Completed, etc) | Text |  |  | Completed |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Entity name | Description | | | | |
| Order\_Items | This table stores details of items included in an order, with quantity and price. | | | | |
| Field name | **Description** | **Data type** | **Key field** | **Constraints** | **Example** |
| Order\_Item\_ID | Uniquely Identifies each order item | Int | PK | Not Null, Unique | 9001 |
| Order\_ID | Reference to the order | Int | FK | Not Null | 5001 |
| Menu\_Item\_ID | Reference to the menu item ordered | Int | FK | Not Null | 35 |
| Quantity | Quantity of the item ordered | Int |  | Not Null | 2 |
| Price | Price of the item at the time of ordered | Decimal |  | Not Null | $9.00 |
| Instruction | Special instructions for preparing the item | Text |  |  | No sugar |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Entity name | Description | | | | |
| Payments | This table records payments made for orders, including the method of payment. | | | | |
| Field name | **Description** | **Data type** | **Key field** | **Constraints** | **Example** |
| Payment\_ID | Uniquely identifies each payment | Int | PK | Not Null, Unique | 1, 2 3 |
| Order\_ID | References to the order being paid | Int | FK | Not Null | 2 |
| Amount | amount paid for the order | Decimal |  | Not null | 35.25 |
| Payment\_Method\_ID | References the payment method used | int | FK | Not Null | 3 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Entity name | Description | | | | |
| Customer Feedback | This table stores customer reviews and ratings for the cafe’s service and menu. | | | | |
| Field name | **Description** | **Data type** | **Key field** | **Constraints** | **Example** |
| Feedback\_ID | Uniquely Identifies each feedback entry | Int | PK | Not Null, Unique | 4, 6 |
| Customer\_ID | References the customer providing feedback | Int | FK | Not Null | 401, 785 |
| Feedback | Comments or review provided by the customer | Text |  |  | Superb! |
| Rating | Rating of service or food given by customer | int |  | Not Null | 2 |

# Contributions

We used Github to manage our documents properly. Given below is my repository link:

<https://github.com/SujalGajera/Cafe_Management_System>

**Café Management System – SG Team (Sujal | Devpreet | Yubraj):**

**Sujal**

I played a crucial role in this project as a leader and my contribution is

SDLC,

User Stories (Employees) / Use Cases (11 to 15),

Use Case Diagram,

ERD (Physical),

Data Dictionary (Orders, Order Items, Categories, Menu Items), and

final Presentation.

**Devpreet**

Executive Summary,

User Stories (Owner) / Use Cases (6 to 10),

Evidence from brainstorming,

Data Dictionary (Tables, Payment and Payments Method)

**Yubraj**

Actors,

User Stories (Customer), / Use Cases (1 to 5),

Functional and Non – Functional Requirements

Data Dictionary (Owner, Customer, Customer Feedback, Employees)

# References

Add references here if you have any.