

FOOD ORDERING AND DELIVERY MANAGEMENT SYSTEM

NAME: SIRI S ARADHYA SRN: PES1UG23CS906

NAME: YOGITHA A S SRN:PES1UG23CS901

SECTION: L

1. User Requirement Specification

1.1 Purpose of the Project

The purpose of this project is to design and develop a Food Ordering System that enables users to browse restaurants, view menus, place orders, and provide feedback, while allowing administrators to manage restaurants, menu items, and customer orders efficiently.

This project aims to simulate the functionality of popular food delivery platforms like Swiggy or Zomato, providing an interactive web-based platform integrated with a relational database. It demonstrates how database concepts such as normalization, referential integrity, stored procedures, and triggers can be implemented effectively in a real-world business scenario.

The system provides an end-to-end food ordering experience — from restaurant browsing and cart management to order placement and payment tracking — all supported by robust backend database operations.

1.2 Scope of the Project

The scope of this project covers both user-facing and administrator-facing functionalities.

- **For Users:**
The system allows registration and secure login using password hashing, browsing multiple restaurants and their menus, adding desired food items to a cart, placing orders, selecting payment methods, and submitting reviews with ratings.
- **For Administrators:**
The system provides a dashboard to manage restaurants, update or delete menu items, track order progress, and update order statuses.

The project ensures data consistency and automation using triggers, procedures, and functions. It demonstrates complete CRUD (Create, Read, Update, Delete) operations across all entities and enforces business rules such as inventory control and order status history.

This application is scalable and can be extended further with delivery tracking, user analytics, and coupon-based discount systems, making it a comprehensive database-driven business solution.

1.3 Detailed Description of the Project

The Food Ordering System is implemented using MySQL as the backend database and Streamlit (Python) as the frontend interface.

It integrates multiple interrelated tables that represent real-world entities such as users, restaurants, menu items, orders, payments, and reviews.

- The database schema includes tables: Users, Restaurants, Menu, Orders, Order_Items, Cart, Payments, Coupons, Delivery_Partners, Reviews, and Order_Status_History.

- The database is normalized and uses foreign key constraints to maintain referential integrity among entities.
- Stored Procedures automate key operations such as order placement and review submission (PlaceOrderFromCart, AddReview).
- Functions such as GetOrderTotal and GetRestaurantAvgRating help compute derived data for business insights.
- Triggers handle automatic updates — adjusting stock levels after each order, recalculating totals, and maintaining a detailed log of order status changes.
- The Streamlit frontend provides a graphical user interface with interactive features for both customers and administrators. Users can browse restaurants with images, view menu categories, and interactively manage their cart, while admins can view and update data seamlessly.

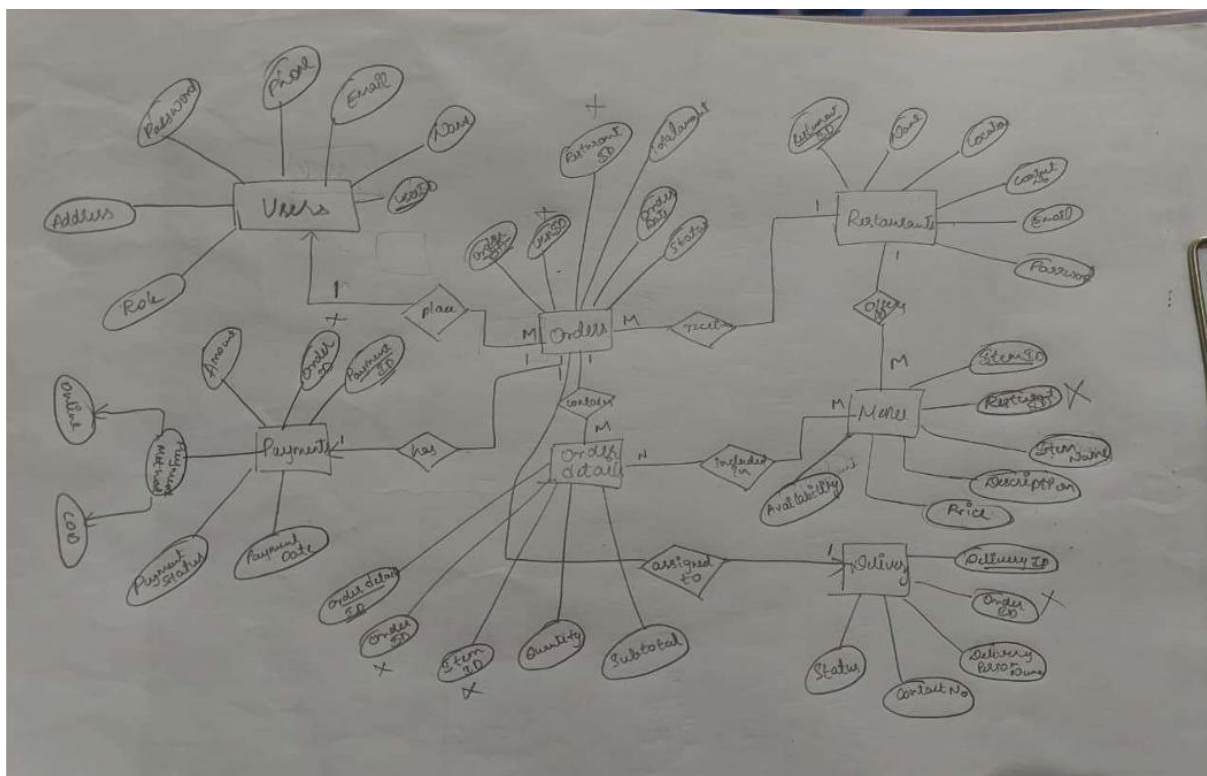
The system thus demonstrates a complete data flow from front-end operations to backend database transactions, ensuring accuracy, security, and ease of management.

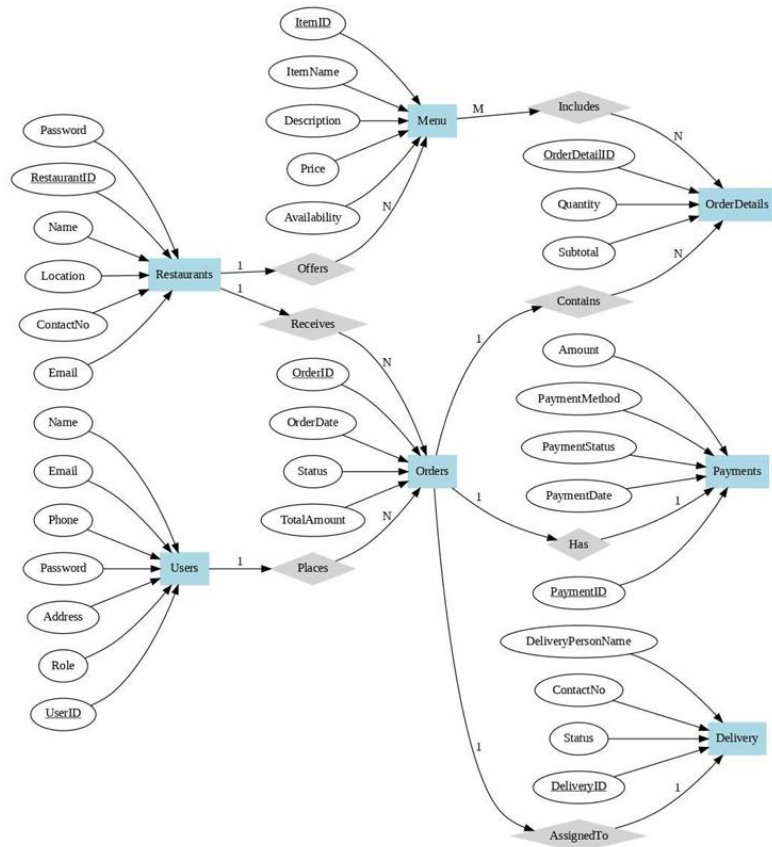
1.4 Functional Requirements

Sl. No.	System Functionality	Description
1	User Registration & Login	Allows users to sign up with name, email, and password. Passwords are hashed using SHA-256 for secure authentication.
2	Restaurant Browsing	Displays a list of restaurants with details and images. Users can select a restaurant to view its menu.
3	Menu Display & Filtering	Fetches menu items dynamically from the database and categorizes them (e.g., Pizza, Drinks, Desserts).
4	Add to Cart	Enables users to add menu items to their cart with a specified quantity. Duplicate entries are restricted by unique constraints.
5	View / Modify Cart	Displays all items in the user's cart with quantities, prices, and totals. Allows removal of items.
6	Place Order	Executes stored procedure PlaceOrderFromCart to create an order, move items from the cart, and compute total amount.
7	Payment Processing	Updates payment details (method and amount) for each order. Supports Credit Card, Debit Card, UPI, Wallet, and Cash.
8	Automatic Stock Update (Trigger)	Triggers automatically adjust Menu.stock when items are ordered or deleted. Prevents negative stock levels.
9	Order Status Tracking	Order statuses are updated by users/admins. A trigger logs every change in Order_Status_History.
10	Review Submission	Users can submit ratings and comments through the AddReview procedure. Validation ensures ratings between 1 and 5.

11	Admin Dashboard	Admins can add/delete restaurants, manage menu items, and update order statuses from a single interface.
12	Coupon Management	Coupon codes provide percentage-based discounts with validation of expiry date and max discount amount.
13	Data Analytics & Insights	Functions and queries provide insights such as average restaurant ratings.
14	Security & Integrity	The system ensures data consistency through foreign keys, constraints, and controlled SQL operations in the application layer.

1.5 ER Diagram and Representation Schema





Representation Schema For web based Food delivery :-

Users :

User ID	Name	Email	Phone	Password	Address	Role
---------	------	-------	-------	----------	---------	------

Restaurants :

Restaurant ID	Name	Location	Contact info	Email	Password
---------------	------	----------	--------------	-------	----------

Menu :

Item ID	Item Name	Description	Price	Availability	Restaurant ID
---------	-----------	-------------	-------	--------------	---------------

Order :

Order ID	User ID	Restaurant ID	Order Date	Status	Total Amt
----------	---------	---------------	------------	--------	-----------

Order Details :

OrderDetail ID	Order ID	Item ID	Quantity	Sub total
----------------	----------	---------	----------	-----------

Payments :

Payment ID	Order ID	Amount	Payment Method	Payment Status	Payment Date
------------	----------	--------	----------------	----------------	--------------

Delivery :

Delivery ID	Order ID	Delivery Person Name	Contact No	Status
-------------	----------	----------------------	------------	--------

