

# **Project Title: Restaurant Management System**

Group Members: 190050126 Valeti Bharat Chandu ,  
190050127 Vedant B ,  
190050070 Nabarun Champramrari

## **Overview of application:**

Our Restaurant Management System is designed to streamline the process of managing a restaurant, making it more efficient and user-friendly. The system will allow users to manage reservations, menus, orders, and billing in one central location. The system will also provide data analytics to restaurant managers, helping them make informed business decisions.

## **Users:**

Restaurant Manager - This user class will have access to all features of the system, including managing tables, menus, orders, and billing. They will also have access to data analytics to make informed business decisions.

Wait & Cashier Staff - This user class will be able to manage reservations, take orders, and process billing.

Kitchen Staff - This user class will be able to view orders and mark them as complete.

Customers - This user class will be able to view the menu, make reservations, and place orders.

## **Data to be stored:**

Table - Table No. , Unique Identifier, Location, Seating Capacity

Menu Item - Name, Description, Price, Ingredients

Order - Table Number, Menu Items, Special Requests, Total Price

Reservation - Date, Time, Table Number, Customer Information

Billing - Order Information, Tax, Total Price

Customers - Name , Place , Address

Waiters - ID , Name

Chefs - ID , Name

Cashiers - ID, Name

**List of entities with attributes , constraints :** [DDL file](#)

**ER diagram:** [link](#)

## **Application structure:**

Backend: The main API functionality includes:

Managing Tables - Create, Update, and Delete tables, menu items, orders, bills, reservations

Handling different queries on the tables

UI: We plan to support a web-based UI using Angular/React for the frontend and Node.js for the backend.

## **API interfaces:**

Login API:

Endpoint: /login

Method: POST

Request parameters: username, password

Return value: status (success/failure), user information (if login successful), error message (if login failed)

Menu API:

Endpoint: /menu

Method: GET, POST, PUT, DELETE

Request parameters: for POST and PUT: item name, description, price, quantity; for DELETE: item ID

Return value: list of menu items (if GET request), status (success/failure), error message (if any)

Order API:

Endpoint: /order

Method: GET, POST, PUT, DELETE

Request parameters: for POST: order details (e.g. customer name, items ordered, delivery person ID); for

PUT: order ID and status update (e.g. "delivered"); for DELETE: order ID

Return value: list of orders (if GET request), status (success/failure), error message (if any)

Billing API:

Endpoint: /billing

Method: GET, POST

Request parameters: for POST: order details (e.g. cashier ID, waiter ID, order ID, total cost, tax, total amount, chef ID); for GET: billing ID

Return value: list of billings (if GET request), status (success/failure), error message (if any)

Add a new menu item:

Endpoint: POST /api/menu

Parameters:

name (string)

description (string)

price (number)

Return value:

If successful, returns a JSON object with the new menu item's ID and a success message.

If unsuccessful (e.g. if the name is already taken), returns a JSON object with an error message.

Place a new order:

Endpoint: POST /api/orders

Parameters:

items (array of objects) - each object should contain the ID of a menu item, the quantity ordered, and any special instructions

customer\_id (string) - the ID of the customer placing the order

order\_type (string) - the type of order (e.g. dine-in, takeout, delivery)

Return value:

If successful, returns a JSON object with the order's ID and a success message.

If unsuccessful (e.g. if an item ID is invalid), returns a JSON object with an error message.

Other Aspects: We plan to generate test data to simulate the use of the system.

### **Key User Interfaces:**

Login Page:

Data Displayed: Login Form

Form Elements: Username, Password

Dashboard for Restaurant Manager:

Data Displayed: Statistics on, orders, and billing

Links: Menu Management Page, Order Management Page, Billing Management Page

Menu Management Page:

Data Displayed: List of menu items, item no , price, quantity

Form Elements: Add Menu Item, Update Menu Item's quantity/price, Delete Menu Item

Order Management Page:

Data Displayed: List of orders , order\_no, order\_date, order\_type, name,  
items,delivery\_person\_id,delievry\_fee,total\_cost, Status

Form Elements: Take Order, Mark Order as delivered

Billing Management Page:

Data Displayed: bill\_no,cashier\_id,waiter\_id,total\_cost,tax,total\_amount,order\_details,chef\_id

Form Elements: Generate Bill

Menu View Page for Customers:

Data Displayed: menu items with their price and description

Form Elements: Add to Cart

Order Page for Customers:

Data Displayed: Order Form

Form Elements: Menu Items, Total Price, order\_no, order\_date, order\_type, ,  
items,delivery\_person\_id,delievry\_fee,total\_cost.