PES UNIVERSITY, Bengaluru
Department of Computer Science and Engineering
B. Tech (CSE) – 5th Semester – Aug-Dec 2024

B.TECH. (CSE)
V Semester
UE22CS341A –Software Engineering

PROJECT REPORT
On
Restaurant Management System

Submitted by:

Aathil Nishad	PES2UG22CS011
Ajaybir Singh	PES2UG22CS042

August – Dec 2024
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
BENGALURU – 560100, KARNATAKA, INDIA

Table of Contents

UE22CS341A - Software Engineering

Project Description For Restaurant Database Management System

The Restaurant Management System (RMS) is designed to improve and digitise the operations of a restaurant by integrating various aspects of restaurant management into a website.

It is a brand-new development project aimed at addressing the challenges faced by small to medium-sized restaurants in managing their day-to-day operations. The system will be highly customizable to cater to the unique needs of different types of restaurants, such as dine-in, takeaway, or delivery-based establishments.

Target Audience:

- **Restaurant Owners and Managers:** Individuals responsible for overseeing the daily operations of the restaurant.
- **Staff (Waiters, Chefs, Cashiers):** Employees who work in the restaurant can interact with the system to manage orders, process payments, and ensure inventory levels.
- **Customers:** End-users who can place orders, make reservations, and provide feedback through the system.

Key Stakeholders:

- **Restaurant Owners:** Interested in a solution that offers easier management and increased efficiency.
- **Staff:** Looking for a user-friendly interface to manage their tasks reducing their workload.
- **Customers:** Desire a seamless dining experience, whether they are dining in, ordering takeout, or booking a table.

Functional Features:

- **Table Management:**
  - View and manage table availability.
  - Assign customers to tables.
- **Order Management:**
  - Take orders digitally and send them directly to the kitchen.
  - Modify or cancel orders in real-time.
  - Track the status of each order (e.g., preparing, ready, served).
- **Reservation System:**

    - Allow customers to book tables online.
    - Manage reservations and cancellations.
- **Menu Management:**
    - Create and update digital menus with pricing, descriptions, and images.
- **Inventory Management:**
    - Automatically update inventory as orders are placed.
    - Generate alerts for low-stock items.
- **Billing and Payment Processing:**
    - Generate itemized bills for customers.
    - Support multiple payment methods (cash, credit/debit cards, digital wallets).
- **Employee Management:**
    - Manage staff schedules and shifts.
    - Track employee attendance.
    - Handle payroll calculations based on hours worked and tips received.

Plan of Work and Product Ownership

- Feasibility Study and Requirement Analysis:
    - First, we will conduct a feasibility study and analyse the project requirements. This step is essential for laying a solid foundation and helping us avoid potential issues as we move forward.
    - Time Allotted: September
- Backend Development and Server Configuration:
    - Next, we will focus on developing the backend, with an emphasis on designing a robust SQL database. This includes defining the schema, establishing relationships, and optimizing query performance. We'll also implement server-side logic for CRUD operations and data validation, along with secure API endpoints to ensure efficient and secure data handling.
    - Time allotted: September and November
- Frontend Development:
    - Lastly, we will develop the frontend, creating a user-friendly interface that integrates seamlessly with the backend, ensuring a smooth and cohesive user experience.
    - Time allotted: November and December

Tasks Assigned:

- Aathil Nishad:
    - Creation of backend and its configuration.
    - Adding functionalities for table management, order management, reservation system, and menu management.
- Ajaybir Singh:
    - Creating front end and design.

- o Adding functionalities for inventory management, billing and payment processing and employee management.

UE22CS341A - Software Engineering

PROJECT PLAN
For
Restaurant Database Management System

Life-cycle followed (PES2UG22CS011)

- Project Initiation:
  - o Project Definition:
    - ▪ Clearly define the project scope and objectives, including the specific functionalities (e.g., table management, order management, billing)
  - o Stakeholder Identification:

     Identify key stakeholders, including restaurant owners, managers, staff, and customers.

     Determine their roles, responsibilities, and how they will be involved throughout the project. o Requirements Gathering: ▪ Collect and document functional and non-functional requirements from stakeholders.

  - o Feasibility Analysis:
    - ▪ Assess the technical and operational feasibility of the project.

- o The project requirements for the Restaurant Management System are characterized by high demands for product functionality, high process specifications, and relatively low resource requirement.
- o The Restaurant Management System requires various functions such as
  - Table Management
  - Order Management
  - Reservation Manager ▪ Menu Management
  - Inventory Management
  - Billing and Payment Processing
  - Employee Management o The deployment model used is Iterative model.

- **Frequent Requirement Changes:**
  - Restaurant management systems often need to adapt to dynamic business needs, such as new features, customer preferences, and changes in the restaurant's workflow and menu changes.
- **Complex and Interconnected Features:**
  - Functions like table management, order management, inventory tracking, and billing are interconnected.
  - The iterative model let us focus on specific parts of the system and gradually introduce other features.
- **Risk Reduction:**
  - In the restaurant business, processes like payment handling or inventory management are crucial and risky.
  - Handling potential issues with usability, system security, or resource management before finalizing the product can be done through iterative model.
- **Flexibility in Deployment:**
  - With the Iterative model, we can deploy partial versions of the website (such as a version that only manages orders and payments) while still working on other features, making the development more flexible and adaptable.

- Initial Project Planning:
  - Our first focus is to build a secure login and authentication page for the users.
    - **Role-Based Access Control:**
      - Create the login system such that it supports different user roles (e.g., administrators, waitstaff, kitchen staff, customers).
  - Next task is to create the important functions sequentially, along with its database in

    MySQL.
    - Each function has its own dedicated database.
  - In our next step, we will prioritize the development of the payment and billing feature due to its critical importance and the necessity for exceptional accuracy in managing associated risks.
    - **Prevent Transaction Errors**
    - **Reduce System Downtime**

Once we have completed all these steps, we will start final part of the project that is

Closing. ➢ **Final Report Submission:**

o Prepare a comprehensive final report detailing the problem statement, objectives, methodology, results, and conclusions.

- **Project Presentation:**

o Create a clear and concise presentation that highlights the project's key aspects.

- **Project Evaluation and Feedback:**

o Be prepared to answer questions from faculty or reviewers regarding our methodology, decisions, and outcomes.

Tools Used for this Project

Identify the tools you want to use throughout the lifecycle, such as planning tools, design tools, version control, development tools, bug tracking, and testing tools.

Deliverables classified as reuse/build components (PES2UG22CS042)

1. User Interface (UI) Design ➢

Deliverables:

o Web-based UI for order placement and reservations o

Admin dashboard for managers and staff.

- **Category**: **Reuse Component**
- **Justification**: The UI can be developed using existing web development frameworks and libraries such as React, Angular, or Bootstrap. These frameworks provide pre-built components like forms, tables, and navigation menus, reducing development time and effort.

2. Backend API Development ➢

Deliverables:

o RESTful APIs for managing orders, reservations, menus, inventory, and employee management.

- **Category**: **Build Component**
- **Justification**: The APIs need to be custom-built to integrate all the required functionalities and ensure seamless interaction between the UI, database, and third-party services. Custom

APIs are essential to handle specific business logic, such as order processing, payment integration, and inventory updates.

3. ***Database Design and***

Management ➢ **Deliverables**:

o Database schema for all

entities (e.g., customers, orders,

tables, inventory).

o Scripts for database initialization and migration.

- **Category**: **Build Component**
- **Justification**: The database schema must be designed to meet the specific requirements of the RMS. Custom tables, relationships, and queries will be created to ensure efficient data storage and retrieval. Although a relational database like MySQL is reused, the schema and logic will be custombuilt.

4. ***Table Management Module*** ➢

Deliverables:

  - Real-time table availability management.
  - Assignment of customers to tables.

- **Category**: **Reuse Component**
- **Justification**: This module could leverage existing libraries or plugins for real-time data handling and visualization (e.g., WebSockets or Firebase for real-time updates). Basic CRUD operations can use reusable components from the backend framework.

5. ***Order Management Module*** ➢

Deliverables:

  - Digital order placement and modification.
  - Real-time kitchen notifications.

- **Category**: **Build Component**
- **Justification**: While some components for order placement can be reused, this module will require custom development to handle specific business processes, such as kitchen notification systems and order tracking.

6. ***Reservation Management Module***

   ➢ **Deliverables**:

   o Online table booking and cancellation.

   - **Category**: **Reuse Component**
   - **Justification**: Common functionalities like booking and cancellation can be implemented using reusable components from existing libraries or plugins (e.g., calendar or booking libraries). However, some customization may be needed to align with specific restaurant policies.

7. ***Menu Management Module*** ➢

   Deliverables:

   o Digital menu creation and updates.

   - **Category**: **Reuse Component**
   - **Justification**: Basic CRUD operations for managing menus can leverage existing UI and backend libraries, reducing development time. However, any specific logic for handling real-time updates will need custom work.

8. ***Inventory Management Module*** ➢

   Deliverables:

   o Inventory tracking and low-stock alerts.

   - **Category**: **Build Component**
   - **Justification**: Inventory management involves specific business rules and logic (e.g., reducing stock levels based on orders) that need to be built from scratch. Although some parts (like notification services) can be reused, the core functionality needs custom development.

9. ***Billing and Payment Processing***

   Module ➢ **Deliverables**:

   o Payment gateway integration.

   o Generation of itemized bills.

   - **Category**: **Reuse Component**
   - **Justification**: Payment processing can be handled using existing payment gateway APIs (e.g., Stripe, PayPal, UPI). Reusing these APIs ensures secure and efficient transactions, while the billing system can use existing libraries for generating itemized bills.

10. Employee Management Module ➢

Deliverables:

o Staff scheduling, attendance tracking, and payroll management.

- **Category**: **Build Component**
- **Justification**: Employee management requires custom logic, such as dynamic scheduling and payroll calculation based on hours worked and tips. While some UI components can be reused, the backend logic must be custom-built.

11. Security and Authentication ➢

Deliverables:

o Secure login and role-based access control.

- **Category**: **Reuse Component**
- **Justification**: Reusable components and libraries (e.g., JWT, OAuth) for authentication and authorization can be implemented to ensure secure access management. These are standard practices and do not require custom development.

12. Testing and Quality Assurance ➢

Deliverables: o Test cases for all

modules.

o Performance, security, and usability testing.

- **Category**: **Build Component**
- **Justification**: While some test automation tools (e.g., Selenium, JUnit) can be reused, the test cases need to be built from scratch to cover the specific functional and non-functional requirements of the RMS.

Effort Estimation (in person - months)

Working days for one person would be 7/21.66 = 0.323 person-months.

Given Information:

- **Project Type:** Semi-Detached
- **Estimated KLOC:** Let's assume **10 KLOC** for calculations (you can adjust this based on your specific requirements).
- **Number of People:** 2
- **Working Days Available:** Approximately 45 days until the submission date.

COCOMO Model Constants:

- a = 3.0
- b = 1.12

Effort Calculation:

E = a × (KLOC)^b

E = 3.0 × (10)^1.12

E = 39.547 person-days

Duration Calculation:

T = 2.5 × (E)^0.38

T = 2.5 × (39.547) ^0.38

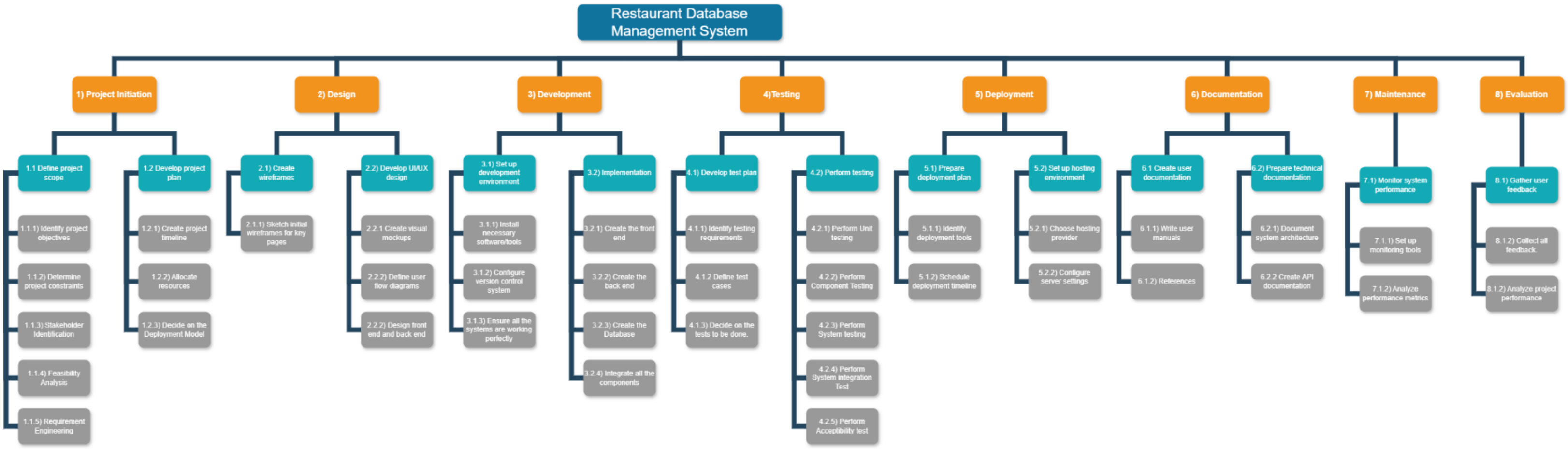
T = 10.11 days

Summary:

- Total Effort Estimate: 39.547 person-days

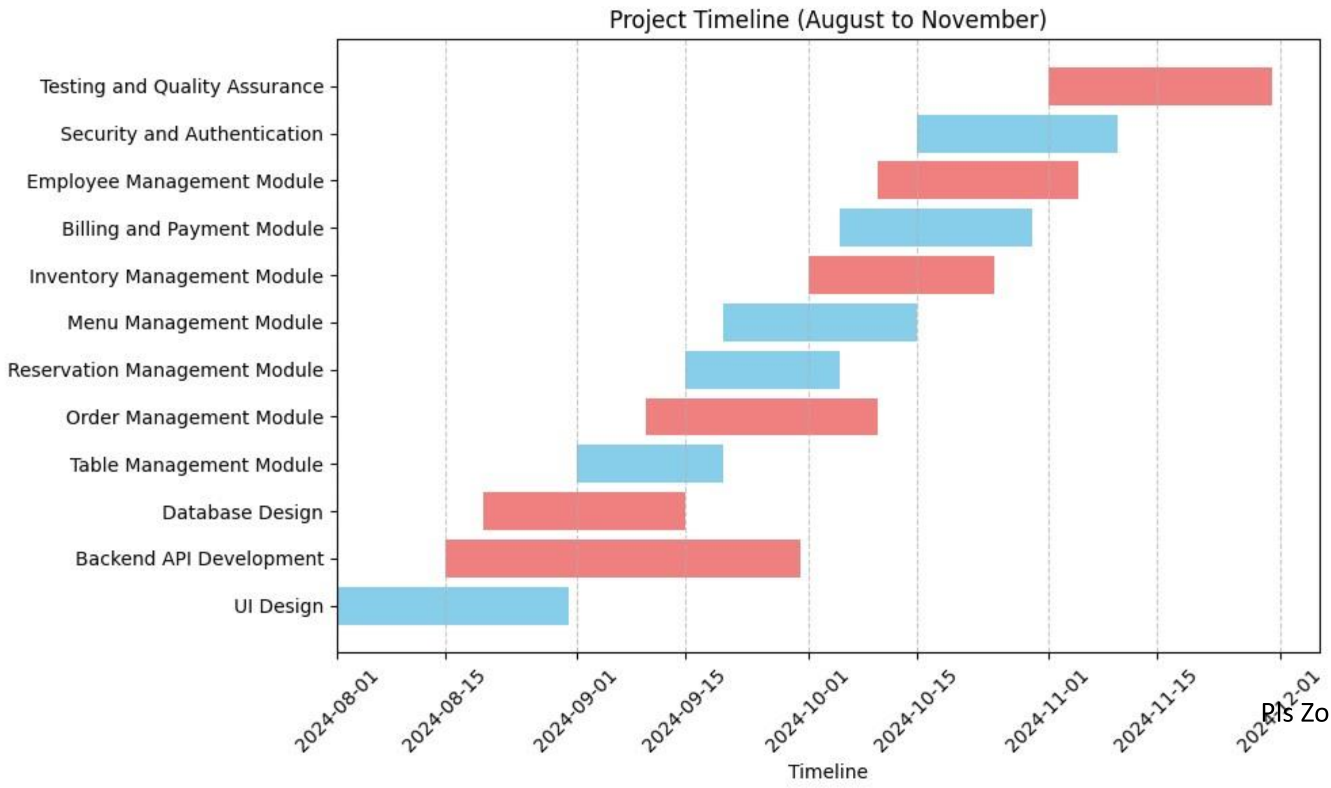
- Total Duration: 10.11 days

Work Breakdown Structure


Restaurant Database Management System
1) Project Initiation
1.1 Define project scope
1.1.1) Identify project objectives
1.1.2) Determine project constraints
1.1.3) Stakeholder Identification
1.1.4) Feasibility Analysis
1.1.5) Requirement Engineering
1.2 Develop project plan
1.2.1) Create project timeline
1.2.2) Allocate resources
1.2.3) Decide on the Deployment Model
2) Design
2.1) Create wireframes
2.1.1) Sketch initial wireframes for key pages
2.2) Develop UI/UX design
2.2.1 Create visual mockups
2.2.2) Define user flow diagrams
2.2.2) Design front end and back end
3) Development
3.1) Set up development environment
3.1.1) Install necessary software/tools
3.1.2) Configure version control system
3.1.3) Ensure all the systems are working perfectly
3.2) Implementation
3.2.1) Create the front end
3.2.2) Create the back end
3.2.3) Create the Database
3.2.4) Integrate all the components
4)Testing
4.1) Develop test plan
4.1.1) Identify testing requirements
4.1.2 Define test cases
4.1.3) Decide on the tests to be done.
4.2) Perform testing
4.2.1) Perform Unit testing
4.2.2) Perform Component Testing
4.2.3) Perform System testing
4.2.4) Perform System integration Test
4.2.5) Perform Acceptibility test
5) Deployment
5.1) Prepare deployment plan
5.1.1) Identify deployment tools
5.1.2) Schedule deployment timeline
5.2) Set up hosting environment
5.2.1) Choose hosting provider
5.2.2) Configure server settings
6) Documentation
6.1 Create user documentation
6.1.1) Write user manuals
6.1.2) References
6.2) Prepare technical documentation
6.2.1) Document system architecture
6.2.2 Create API documentation
7) Maintenance
7.1) Monitor system performance
7.1.1) Set up monitoring tools
7.1.2) Analyze performance metrics
8) Evaluation
8.1) Gather user feedback
8.1.2) Collect all feedback.
8.1.2) Analyze project performance

Pls Zoom

Gantt Chart



Pls Zoom In

UE22CS341A - Software Engineering

System Requirements Specification
For
Restaurant Database Management System

1. Introduction *(Aathil)*

1.1 Purpose

This document specifies the requirements for the **Restaurant Management System (RMS)**. The system is designed to digitize the operations of small to medium-sized restaurants by integrating various aspects of restaurant management, such as table management, order processing, inventory control, employee scheduling, and customer reservations, into a unified web-based platform.

1.2 Scope

The Restaurant Management System intended for use by restaurant owners, managers, staff, and customers. It provides a secure and user-friendly interface for managing restaurant operations, from taking orders to processing payments. The system interacts with a central database and other modules to streamline restaurant operations and improve overall efficiency.

1.3 Definitions, Acronyms, and Abbreviations

RMS: Restaurant Management System

CRUD: Create, Read, Update, Delete

API: Application Programming Interface

UI: User Interface

POS: Point of Service

1.4 References

IEEE Standard for Software Requirements Specifications (IEEE Std 830 1998)

1.5 Overview

The document is structured into sections detailing the functional and non-functional requirements, system features, external interface requirements, and more.

2. Overall Description *(Aathil)*

2.1 Product Perspective

The RMS is a standalone system designed to be integrated into the restaurant's existing infrastructure, including point-of-sale (POS) systems and digital payment gateways. It connects to a central database to manage and synchronize restaurant operations.

2.2 Product Functions

- **Table Management**: View and manage table availability, assign customers to tables.
- **Order Management**: Digital order taking, modification, real-time order tracking, and kitchen notifications.
- **Reservation System**: Online table booking, reservation management, and cancellations.
- **Menu Management**: Create and update digital menus, including pricing, descriptions, and images.
- **Inventory Management**: Track ingredient stock levels, update inventory based on orders, and generate low-stock alerts.
- **Billing and Payment Processing**: Generate itemized bills and support multiple payment methods.
- **Employee Management**: Manage staff schedules, track attendance, and handle payroll calculations.

2.3 User Classes and Characteristics

- **Restaurant Owners and Managers**: Oversee restaurant operations and use the system to monitor performance.
- **Staff (Waiters, Chefs, Cashiers)**: Utilize the system to manage orders, inventory, and other operational tasks.
- **Customers**: Place orders, make reservations, and provide feedback through the system interface.

2.4 Operating Environment

- **Software**: Web Development software such as Node Js, Express Js, HTML, CSS and for the database: MySQL.
- **Hardware**: Devices with internet connectivity such as POS systems, tablets, and desktop computers.

2.5 Design and Implementation Constraints

- Must comply with data protection and privacy regulations.
- Must ensure secure data transmission and storage.
- Must adhere to usability and accessibility standards.

2.6 Assumptions and Dependencies

- Assumes stable internet connectivity for all transactions and communications.

- Requires regular software maintenance and updates.

3. External Interface Requirements *(Aathil)*

3.1 User Interfaces

- Web-based UI for order placement, and reservations.
- Admin dashboard for managers and staff to manage operations.

3.2 Hardware Interfaces

- POS systems for payment processing.
- Tablets or computers for order management and inventory control.

3.3 Software Interfaces

- Integration with digital payment gateways such as UPI's and digital banking.
- APIs for third-party applications like delivery services such as Swiggy or Zomato.

3.4 Communication Interfaces

- Secure protocols like HTTPS for data transmission.

4. System Features *(Ajaybir)*

4.1 Table Management

4.1.1 Description: Manage table availability and customer assignments.
4.1.2 Functional Requirements:

- The system shall allow staff to view table availability in real-time.
- The system shall enable assigning or reassigning customers to available tables.

4.2 Order Management

4.2.1 Description: Digital order placement and kitchen notification.
4.2.2 Functional Requirements:

- The system shall send orders directly to the kitchen upon confirmation.
- The system shall allow modifications or cancellations of orders in real-time.
- The system shall track and display the status of each order.

4.3 Reservation System

4.3.1 Description: Enable online table bookings and cancellations**.**
4.3.2 Functional Requirements:

- The system shall allow customers to book tables via the website.
- The system shall manage reservation availability and handle cancellations.

4.3 Menu Management

4.4.1 Description: Manage digital menus, pricing, and descriptions.
4.4.2 Functional Requirements:

- The system shall allow managers to create and update digital menus.
- The system shall display updated menu items to customers in real-time.

4.4 Inventory Management

4.5.1 Description: Manage food inventory.
4.5.2 Functional Requirements:

- The system shall automatically update inventory levels as orders are placed.
- The system shall generate alerts for items that are low in stock.

4.5 Billing and Payment Processing

4.6.1 Description: Generate bills and process payments.
4.6.2 Functional Requirements:

- The system shall generate itemized bills for customers.
- The system shall support multiple payment methods, including cash and digital wallets.

4.6 Employee Management

4.7.1 Description: Manage staff schedules, attendance, and payroll.
4.7.2 Functional Requirements:

- The system shall allow managers to set and modify staff schedules.
- The system shall track employee attendance and calculate payroll based on hours worked and tips received.

5. Non-Functional Requirements *(Ajaybir)*

5.1 Performance Requirements

- The system shall respond to user inputs within 2 seconds.
- The system shall complete a transaction within 5 seconds, excluding network delays.

5.2 Security Requirements

- The system shall require secure authentication for all user classes.

5.3 Usability Requirements

- The system shall provide an intuitive and user-friendly interface.

5.4 Reliability Requirements

- The system shall have an uptime of 99.9%.

6. Other Requirements *(Aathil)*

6.1 Regulatory Requirements

- The system shall comply with data protection regulations.

Requirements Traceability Matrix (RTM*): (Aathil)*

Requirement ID	Description	Design Specification	Implementation Module	Test Case ID
R-1	Table Management	Design a database to manage table layout, seating arrangement, and track availability in real time.	TableModule	TC-1
R-2	Order Management	Implement order placement, tracking, and status updates with customer-facing and kitchen interfaces.	OrderModule	TC-2
R-3	Reservation System	Develop a reservation system to handle customer bookings, details, and time slot management.	ReservationModule	TC-3
R-4	Menu Management	Design an interface for adding, updating, and deleting menu items, with price and availability controls.	MenuModule	TC-4
R-5	Inventory Management	Create a system to track stock levels, manage inventory additions/removals, and monitor item usage rates.	InventoryModule	TC-5
R-6	Billing and Payment	Implement billing features, payment processing, and generate detailed receipts for customers.	BillingModule	TC-6

R-7	Employee Management	Build a module to manage employee records, scheduling, and assign roles with access control.	EmployeeModule	TC-7

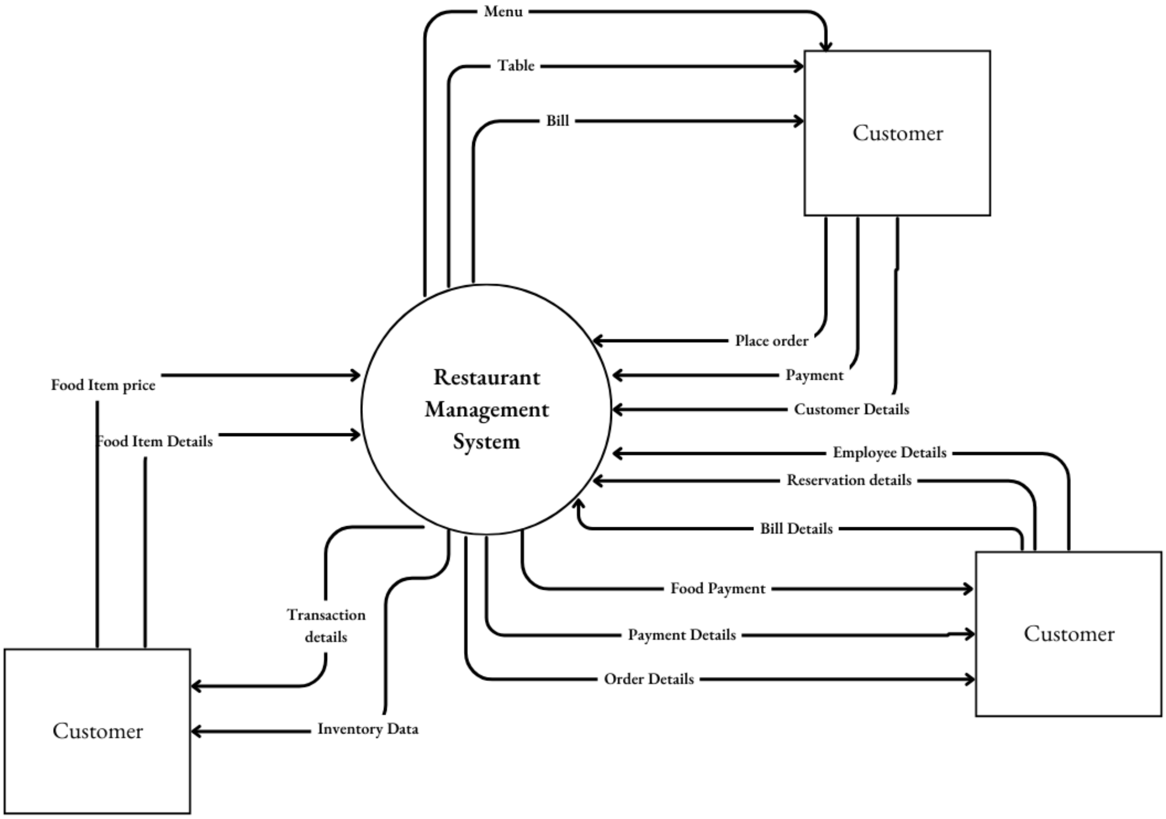
UE22CS341A - Software Engineering

Design Plan For Restaurant Database Management System

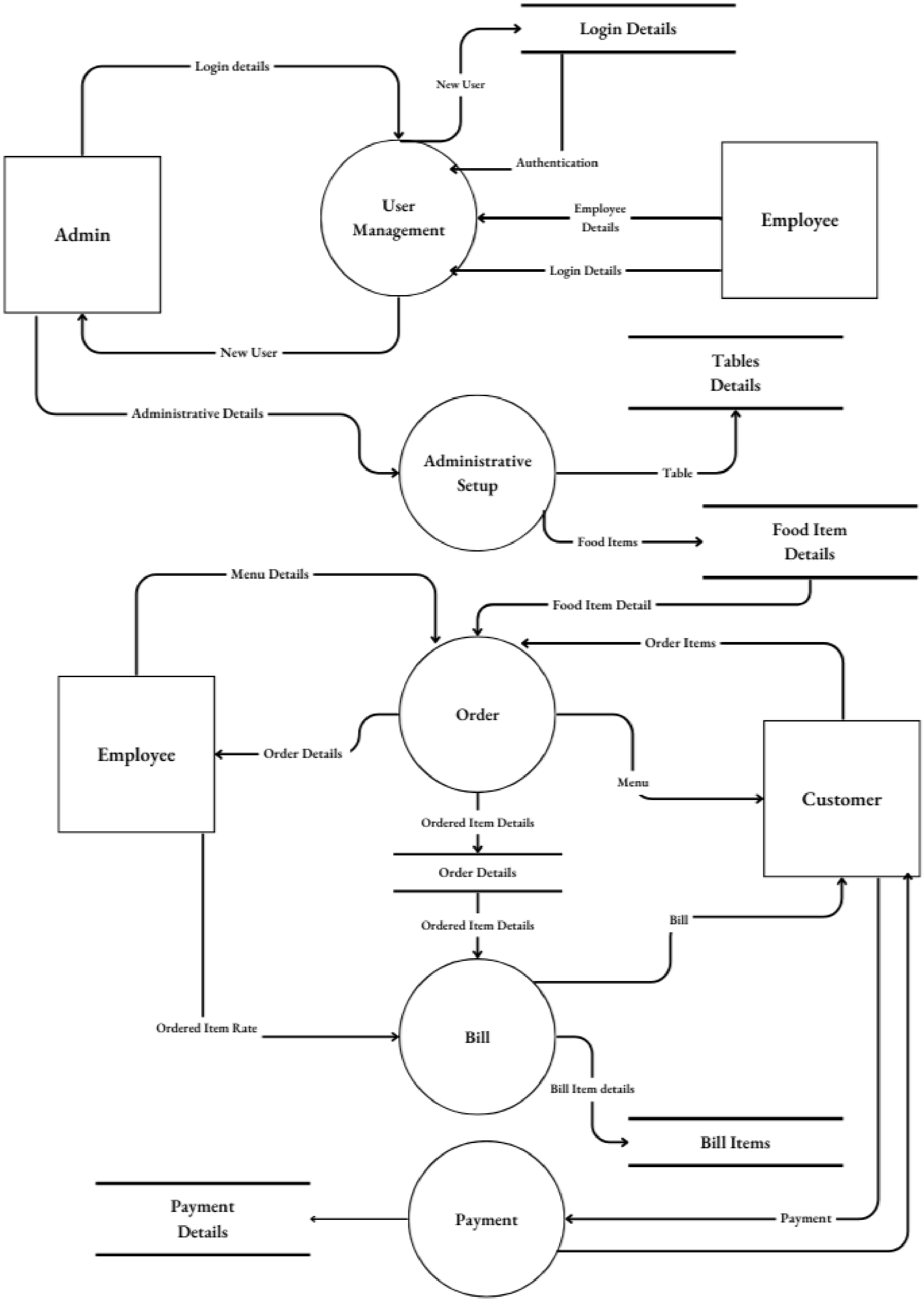
Design Diagrams

Diagrams of Levels of DFD

Level 0



Level 1


Login Details
Login details
New User
Authentication
User Management
Admin
Employee Details
Employee
Login Details
New User
Tables Details
Administrative Details
Administrative Setup
Table
Food Items
Food Item Details
Menu Details
Food Item Detail
Order Items
Order
Employee
Order Details
Menu
Customer
Ordered Item Details
Order Details
Ordered Item Details
Bill
Ordered Item Rate
Bill
Bill Item details
Bill Items
Payment Details
Payment
Payment

Architectural Design

The Restaurant Management System is designed to streamline operations by connecting staff and customers through a ***client-server architecture****.*

The Front End (Client side):

Overview:
The client-side interface allows both restaurant staff and customers to interact with the system through a web app or mobile application. This component is responsible for user interactions, managing orders, reservations, and displaying relevant data and real-time updates.

Key Responsibilities:

- *User Authentication:*
  Manage login and logout functionality for both customers and staff (e.g., servers, managers).
- *Form Submissions:*
  Allow users to submit forms for table reservations, order placements, and feedback.
- *Displaying Real-Time Data:*
  Provide visual representations of key data such as order statuses, table availability, and sales metrics via charts or graphs.
- *Notifications and Alerts:*
  Send real-time notifications for order completions, reservation confirmations, or important updates such as delays or table assignments.

- *Secure Communication:*
  Ensure secure and encrypted communication with the backend server for all data-related interactions such as placing orders, fetching reservations, and managing restaurant operations.

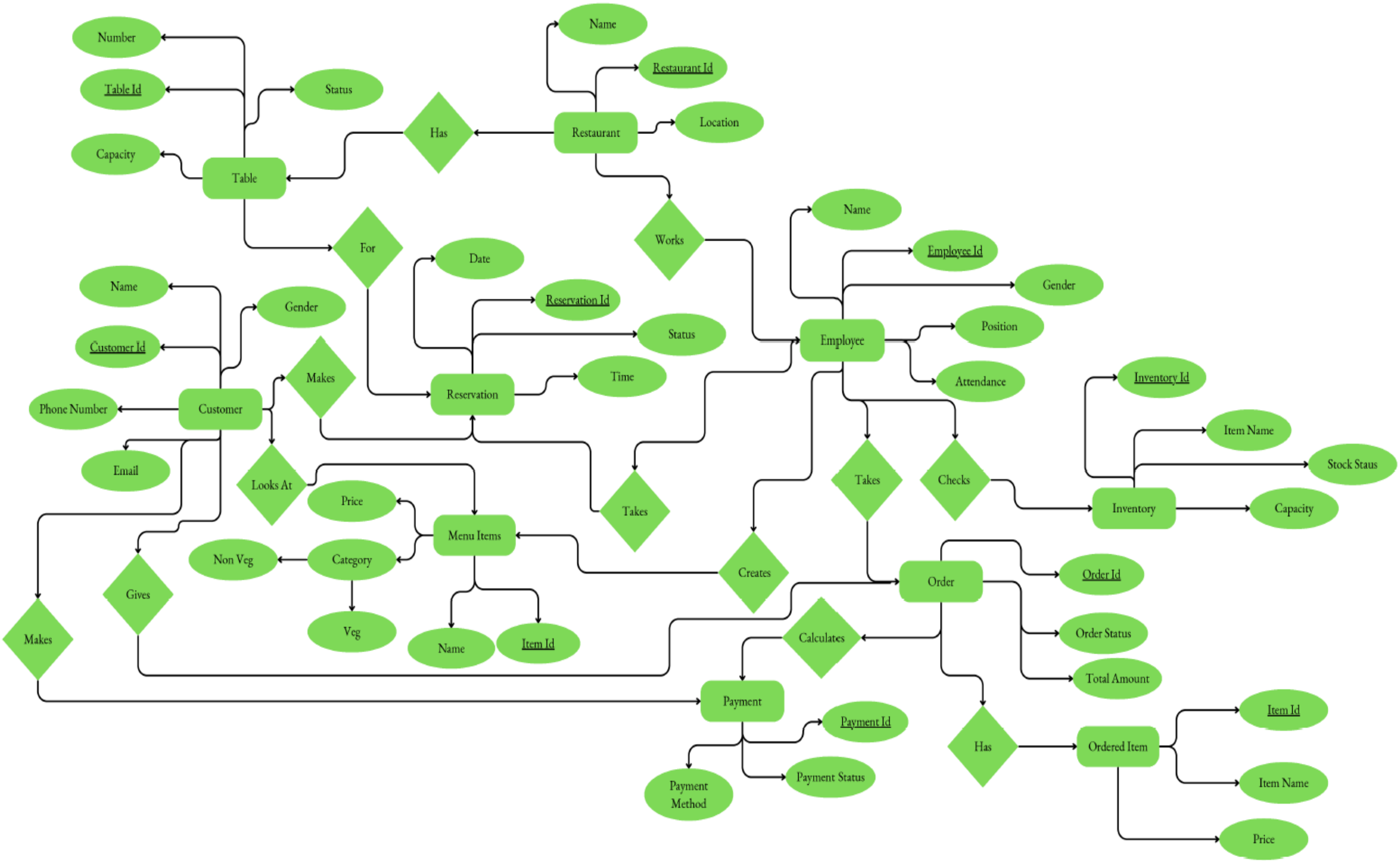
The Backend (Server-Side)

Overview:
The backend is responsible for managing all the business logic, data processing, and communication with the database. It handles order processing, reservations, staff management, and ensuring that the system runs smoothly and securely.

Key Responsibilities:

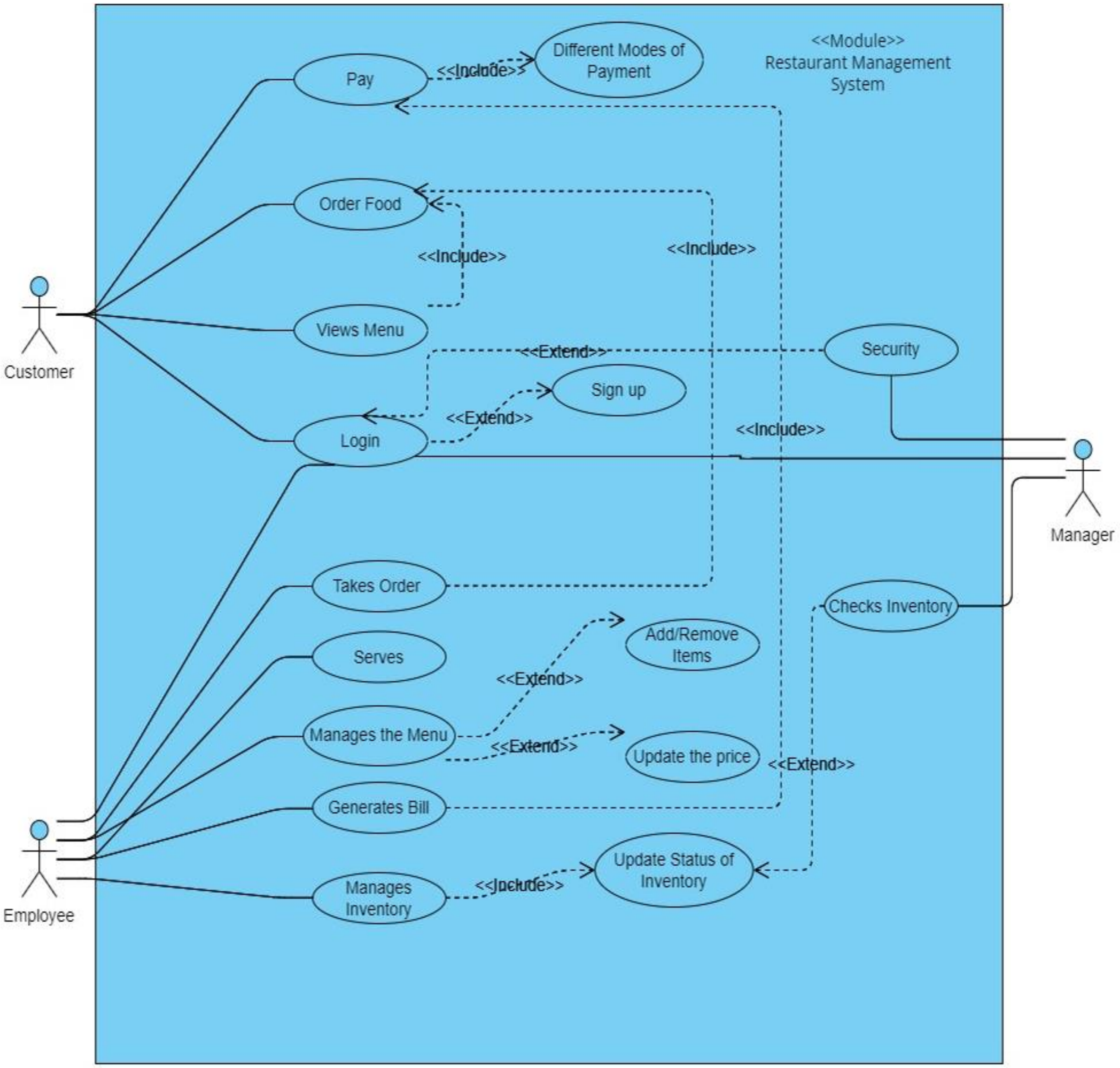
- *User Authentication and Authorization:*
  Verify and authorize user access based on roles (e.g., customers, kitchen staff, managers).
- *Data Processing and Storage:*
  Process and store data related to customer orders, reservations, payments, and staff duties in a centralized SQL database.
- *Data Consistency and Business Rules:*
  Maintain data integrity and ensure that business rules are applied (e.g., alerting staff when a table is overbooked or inventory is running low).

ER Diagram:

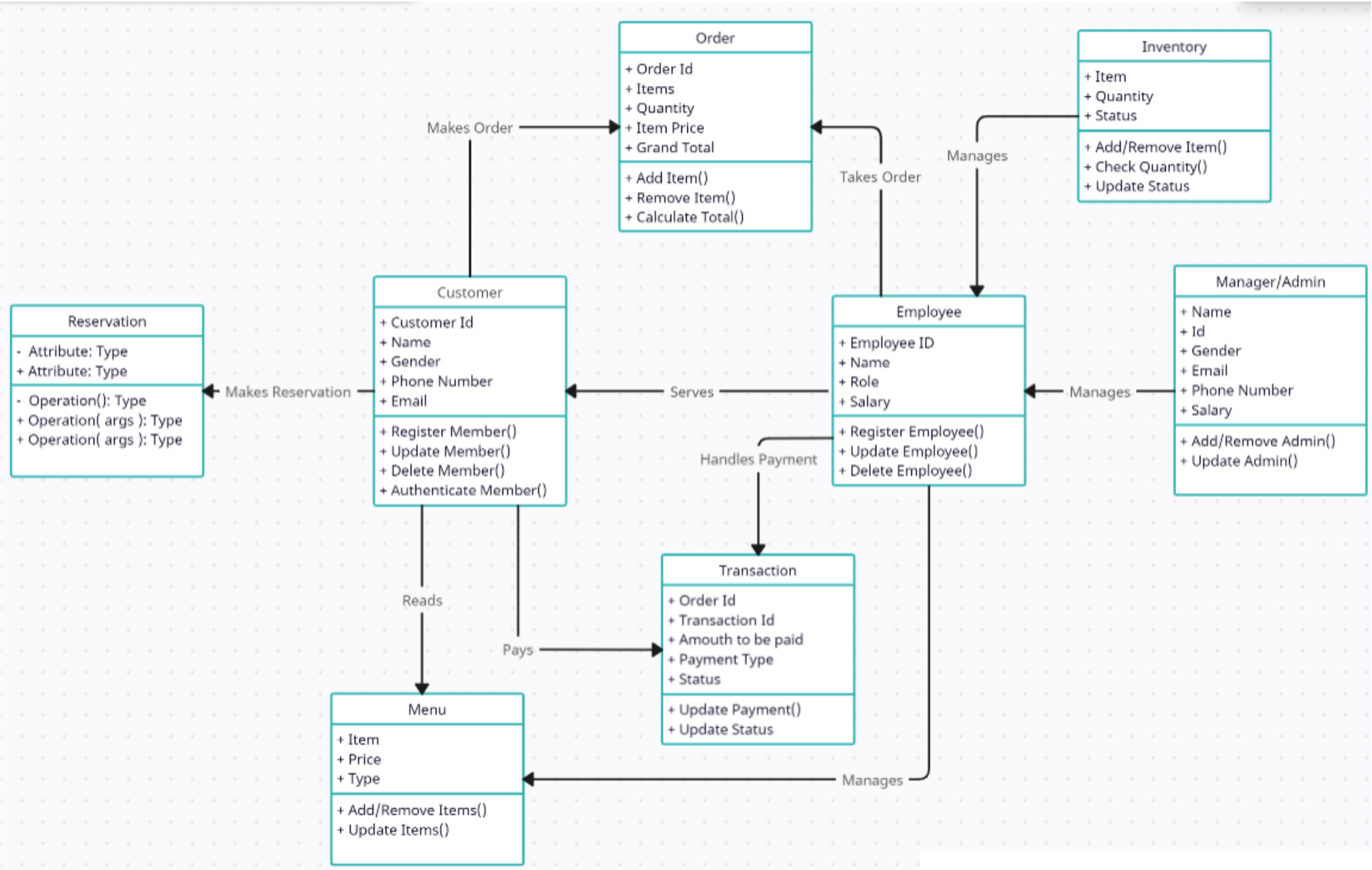

Number
Table Id
Status
Capacity
Table
Has
Name
Restaurant Id
Restaurant
Location
Works
For
Date
Reservation Id
Status
Name
Gender
Customer Id
Phone Number
Customer
Makes
Reservation
Time
Email
Looks At
Price
Menu Items
Takes
Name
Employee Id
Gender
Employee
Position
Attendance
Inventory Id
Item Name
Stock Staus
Takes
Checks
Inventory
Capacity
Non Veg
Category
Veg
Name
Item Id
Gives
Makes
Creates
Order
Order Id
Order Status
Total Amount
Calculates
Payment
Payment Id
Payment Status
Payment Method
Has
Ordered Item
Item Id
Item Name
Price

UML

Use Case Diagram



Class Diagram



UE22CS341A - Software Engineering

Test Plan
For
Restaurant Database Management System

Test Case ID	Description	Pre-Conditions	Test Steps	Test Data	Expected Results	Actual Results	Test Result
UT_01	Validate successful login for staff	Database connection established	1. Enter a valid username (existing staff)	Username: "john.doe"	Login successful with "Welcome, john.doe" message displayed	Login successful with "Welcome, john.doe" message displayed	PASS
UT_02	Validate unsuccessful login for staff (wrong username)	Database connection established	1. Enter an invalid username	Username: "invalid_user"	Login fails with "Invalid username or password" message displayed	Login fails with "Invalid username or password" message displayed	PASS
UT_03	Validate unsuccessful login for staff (wrong password)	Database connection established	1. Enter a valid username (existing staff)	Username: "john.doe"	Password: "incorrect_password"	Login fails with "Invalid username or password" message displayed	Login fails with "Invalid username or password" message displayed
UT_04	Validate login functionality for empty username	Database connection established	1. Leave username field blank	Username: ""	Login fails with "Please fill in all fields" message displayed	Login fails with "Please fill in all fields" message displayed	PASS
UT_05	Validate login functionality for empty password	Database connection established	1. Enter a valid username (existing staff)	Username: "john.doe"	Password: ""	Login fails with "Please fill in all fields" message displayed	Login fails with "Please fill in all fields" message displayed
UT_06	Validate login functionality for empty role	Database connection established	1. Enter a valid username (existing staff)	Username: "john.doe"	Password: "password123"	Role: ""	Login fails with "Please fill in all fields" message displayed

UT_07	Validate database connection error handling		1. Simulate a database connection error (e.g., modify database credentials)	N/A	"Database Connection Error" message displayed	"Database Connection Error" message displayed	PASS
UT_08	Validate handling of unexpected errors	Database connection established	1. Introduce a deliberate error in the login function	N/A	Informative error message displayed (e.g., "An unexpected error occurred. Please try again later.")	Informative error message displayed	PASS

Sample Test Case:

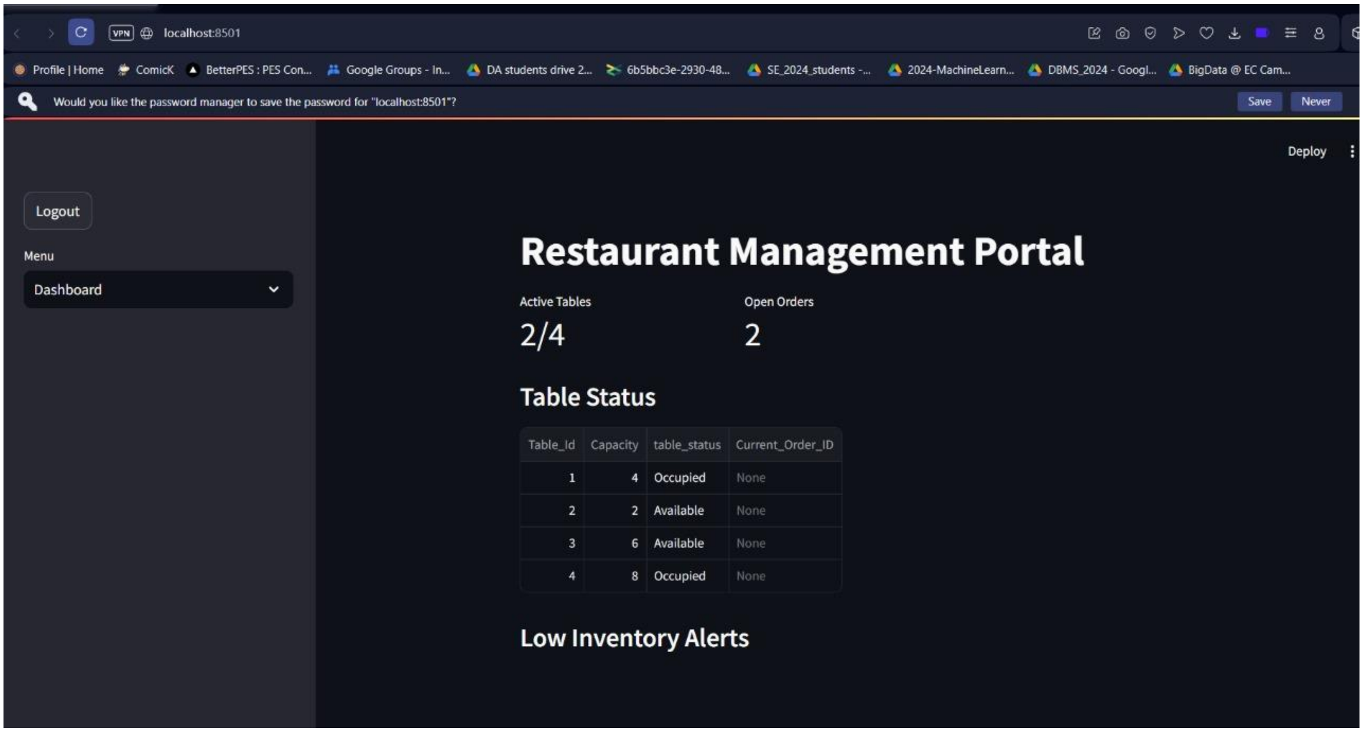
Test Case ID	Description	Pre-Conditions	Test Steps	Test Data	Expected Results	Actual Results	Test Result
IT_01	Validate successful retrieval of menu items	Login as staff	1. Call the get_menu_items function	N/A	All available menu items are retrieved and displayed	All available menu items are retrieved and displayed	PASS
IT_02	Validate successful retrieval of menu items by category	Login as staff	1. Call the get_menu_items function with a specific category	Category: "Appetizers"	Only menu items in the "Appetizers" category are retrieved and displayed	Only menu items in the "Appetizers" category are retrieved and displayed	PASS
IT_03	Validate handling of staff get_menu_items unavailable function menu items	Login as staff	1. Call the get_menu_items function	N/A	Only menu items with Available=True are retrieved and displayed	Only menu items with Available=True are retrieved and displayed	PASS
IT_04	Validate successful update of menu item details	Login as staff	1. Call the update_menu_item function with valid data for an existing item	Updated details (e.g., name, price)	Menu item details are updated in the database and reflected on the UI	Menu item details are updated in the database and reflected on the UI	PASS

IT_05	Validate handling of invalid menu item update data	Login as staff	1. Call the update_menu_item function with invalid data (e.g., negative price)	N/A	Error message displayed indicating invalid data	Error message displayed indicating invalid data	PASS

Test Case ID	Name of Module	Test Case Description	Pre-Conditions	Test Steps	Test Data	Expected Results	Actual Results	Test Result
UT-01	User registration module	To test the login functionality	Access to Chrome Browser	1: Navigate to http://www.demo.com 2: Enter Username and Password 3: Click Submit	User name: PESU Student. Password : pes123	Login should be successful with “welcome message”	Login successful with “welcome message” displayed	PASS

UE22CS341A - Software Engineering

Test Cases
For
Restaurant Database Management System



Would you like the password manager to save the password for "localhost:8501"?

Restaurant Management Portal

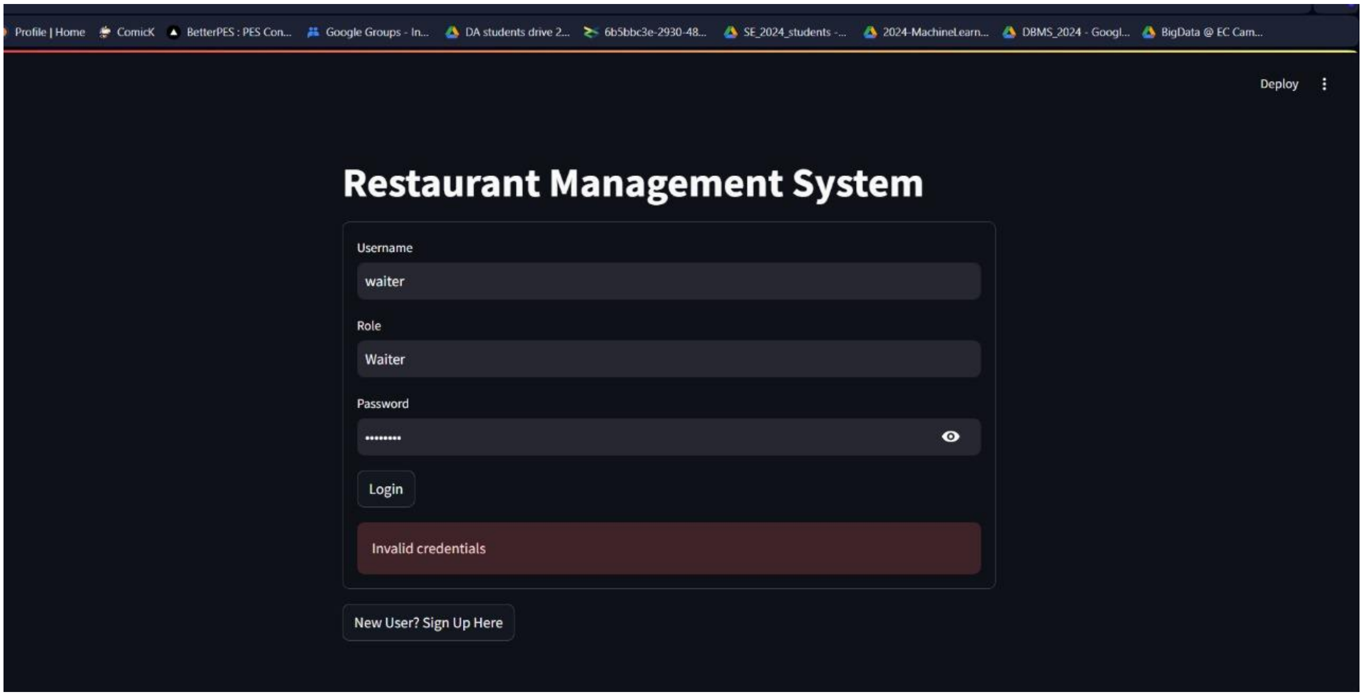
Active Tables
2/4

Open Orders
2

Table Status

Table_Id	Capacity	table_status	Current_Order_ID
1	4	Occupied	None
2	2	Available	None
3	6	Available	None
4	8	Occupied	None

Low Inventory Alerts



Restaurant Management System

Username: waiter

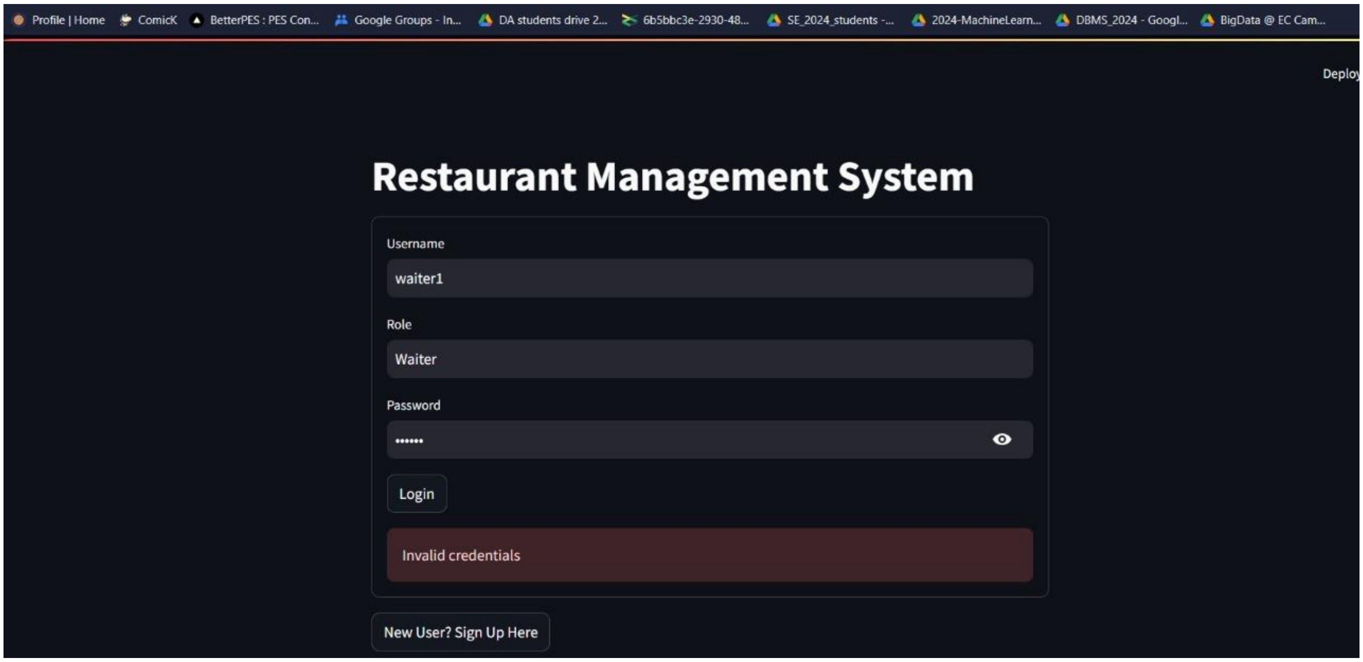
Role: Waiter

Password: ••••••••

Login

Invalid credentials

New User? Sign Up Here



Restaurant Management System

Username: waiter1

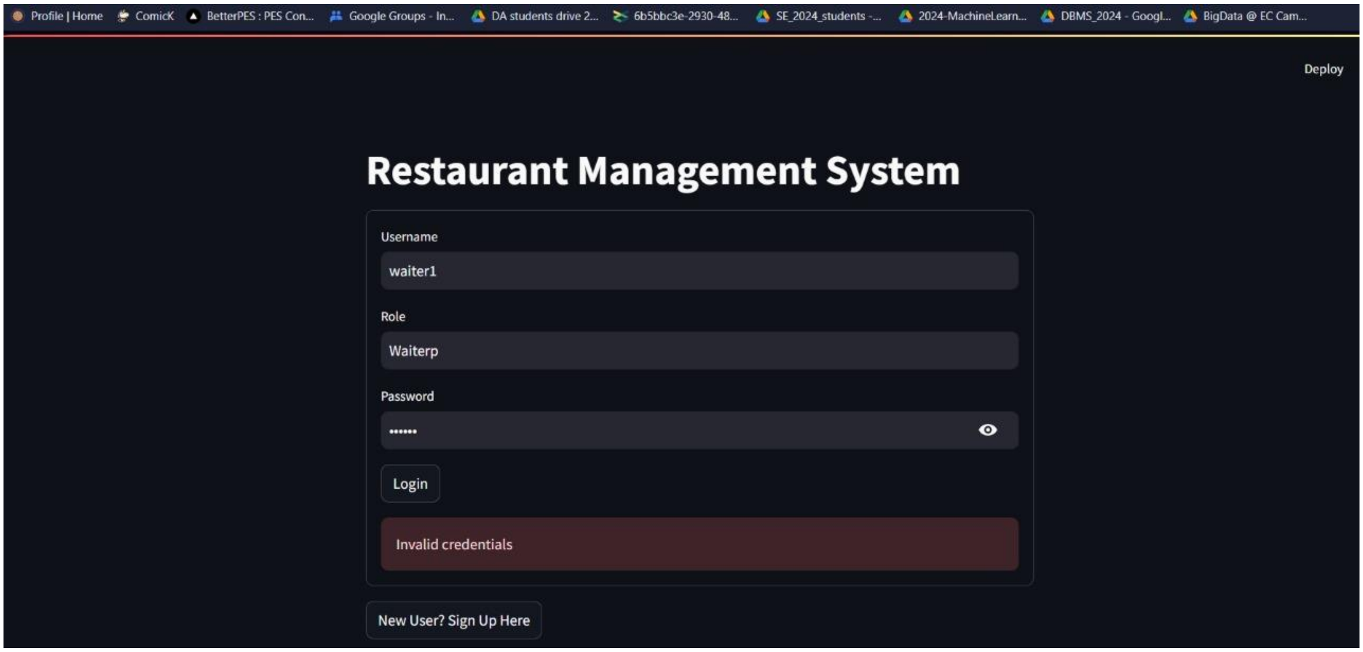
Role: Waiter

Password: ••••••

Login

Invalid credentials

New User? Sign Up Here



Restaurant Management System

Username

waiter1

Role

Waiterp

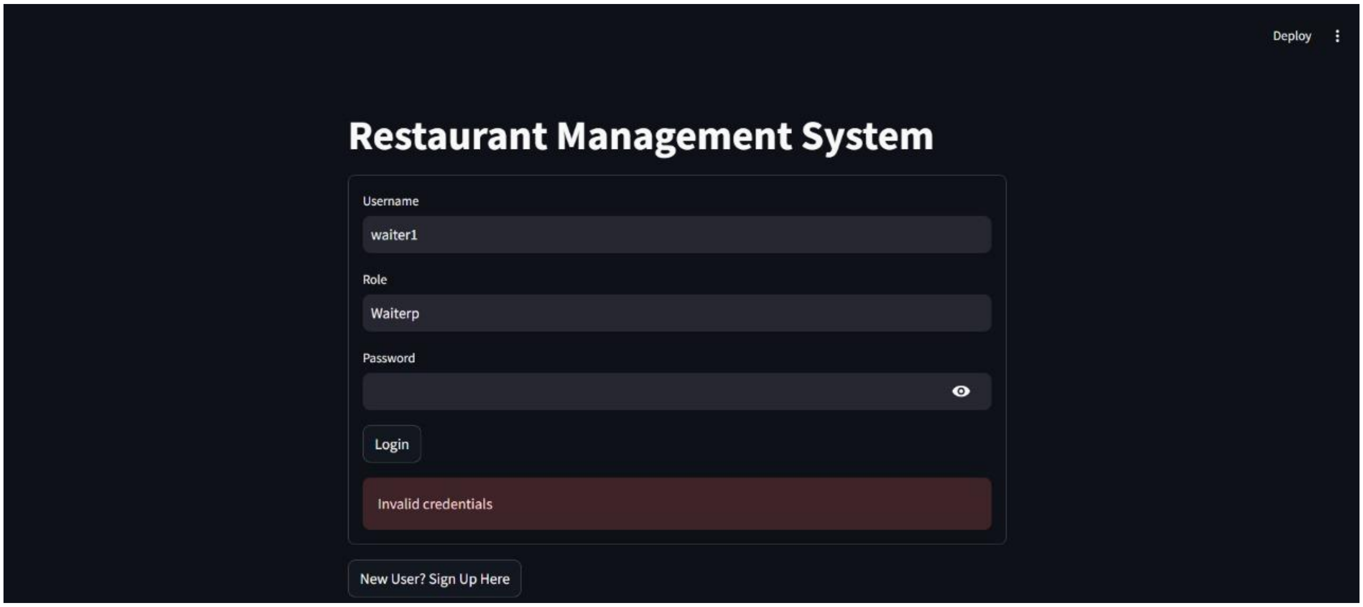
Password

••••••

Login

Invalid credentials

New User? Sign Up Here



Restaurant Management System

Username

waiter1

Role

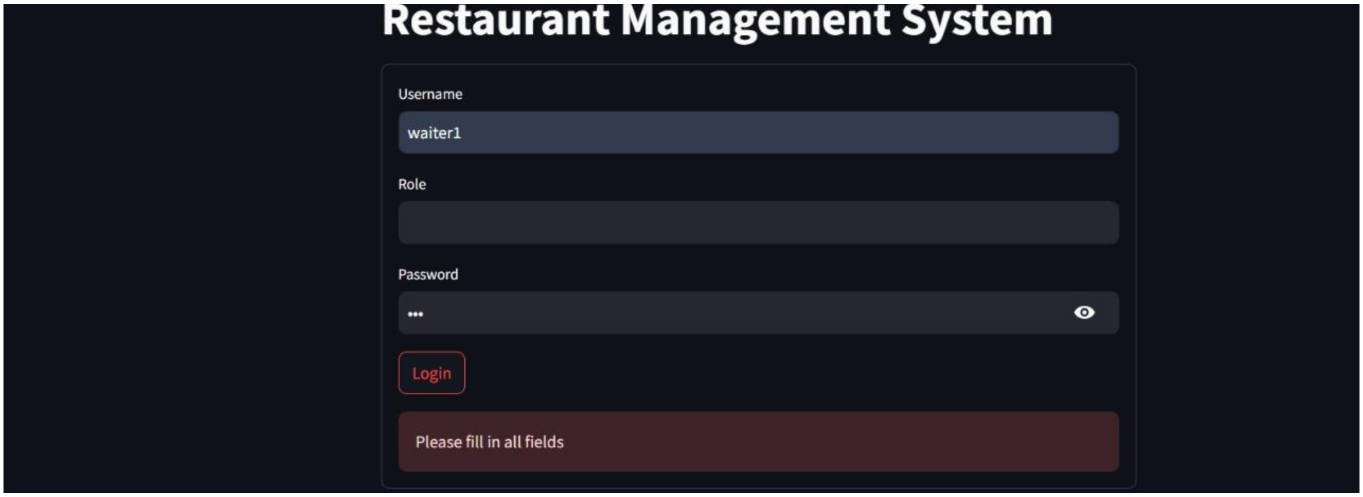
Waiterp

Password

Login

Invalid credentials

New User? Sign Up Here



Restaurant Management System

Username

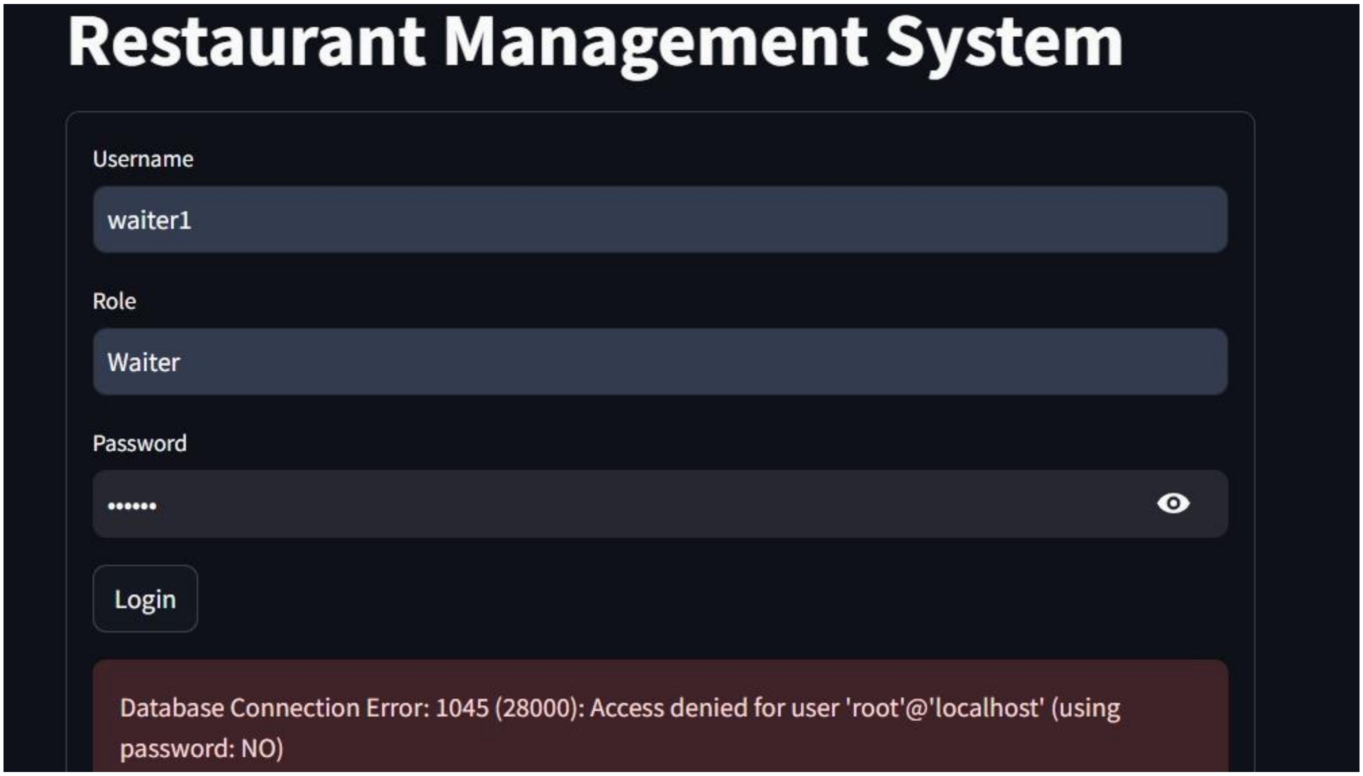
waiter1

Role

Password

•••

Please fill in all fields



Restaurant Management System

Username

waiter1

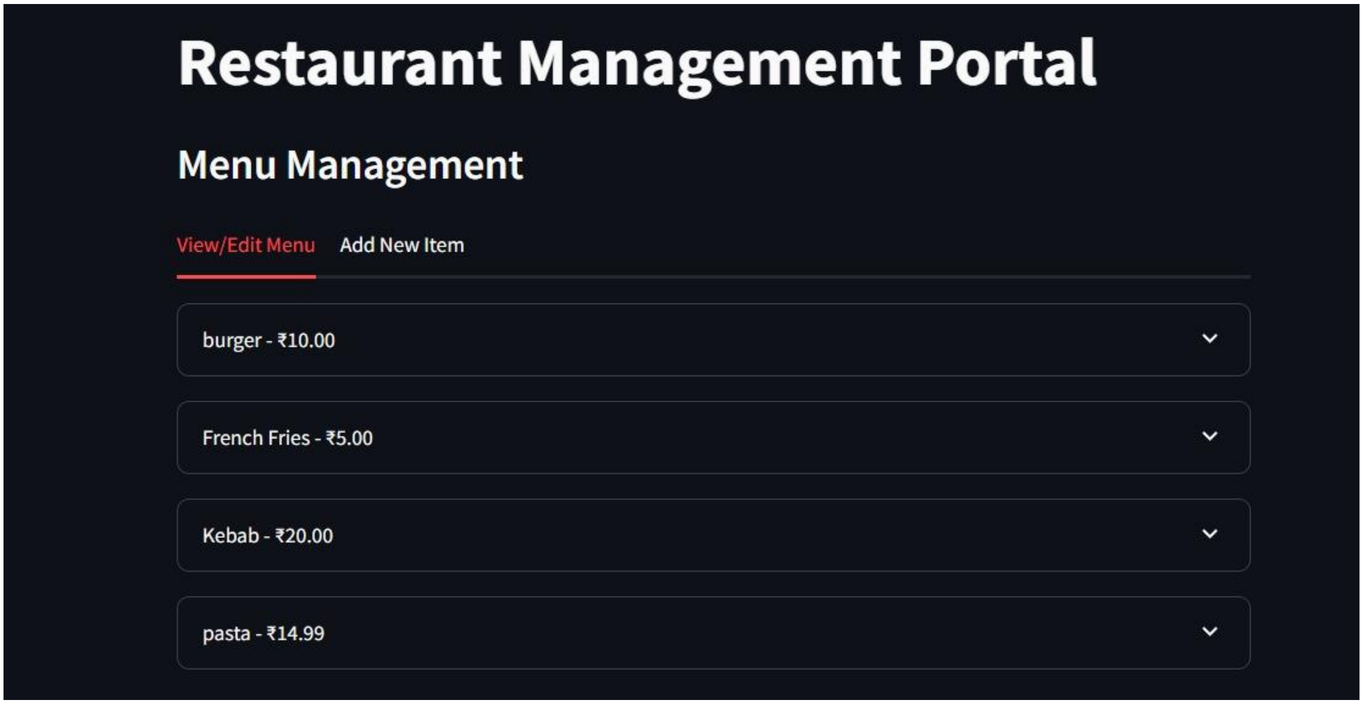
Role

Waiter

Password

••••••

Database Connection Error: 1045 (28000): Access denied for user 'root'@'localhost' (using password: NO)



Restaurant Management Portal

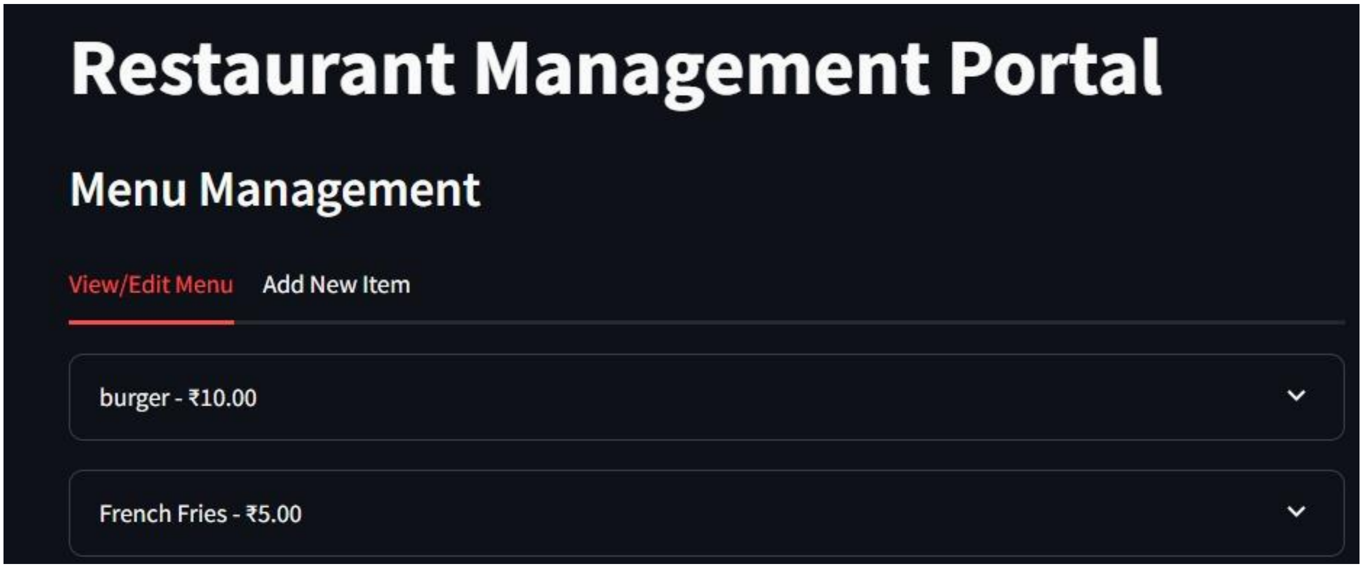
Menu Management

burger - ₹10.00

French Fries - ₹5.00

Kebab - ₹20.00

pasta - ₹14.99

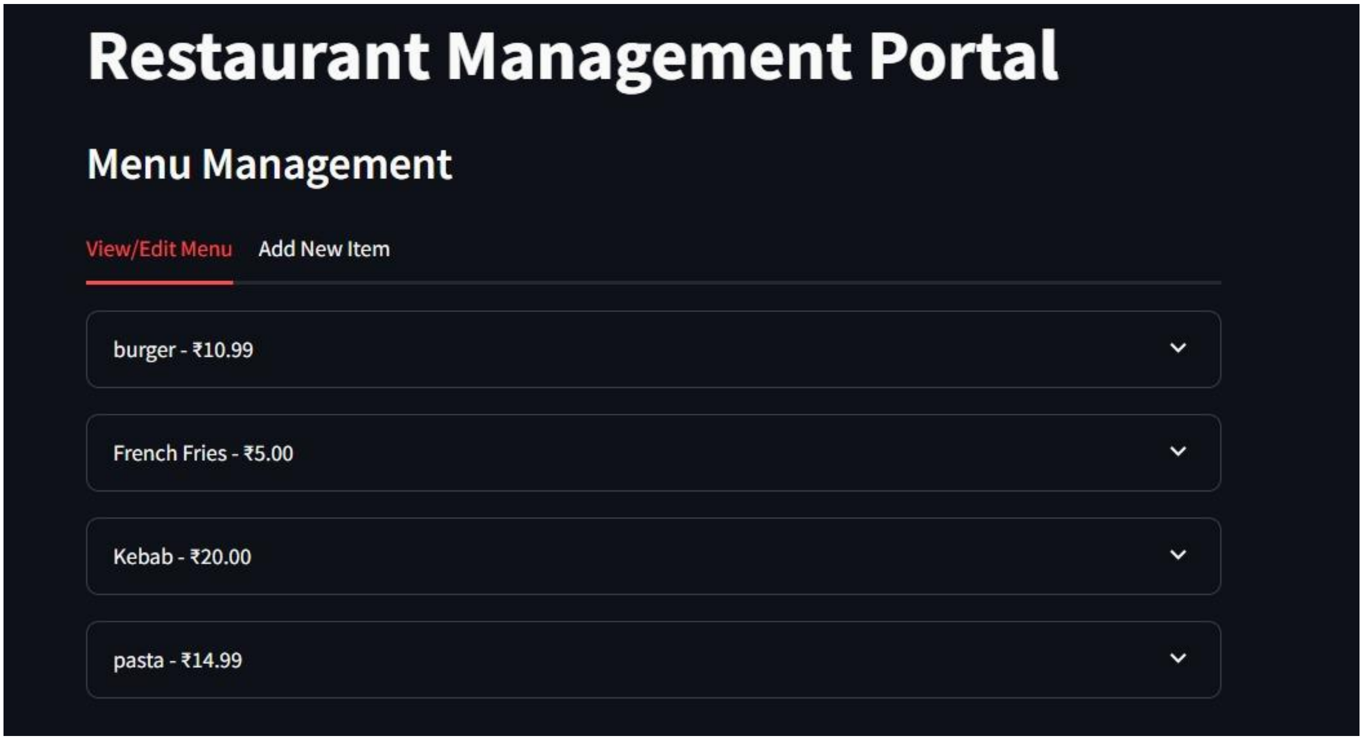


Restaurant Management Portal

Menu Management

burger - ₹10.00

French Fries - ₹5.00



Restaurant Management Portal

Menu Management

burger - ₹10.99

French Fries - ₹5.00

Kebab - ₹20.00

pasta - ₹14.99