1. Preliminary Investigation and its Activities

The preliminary investigation highlights the challenges in current restaurant operations and justifies the need for a centralized, automated Restaurant Management System.

1.1 Problem Identification and Definition

Problem identified

- In many restaurants, order-taking, billing, inventory, and supplier management are handled manually or using separate, unintegrated systems.
- Manual system are time consuming and error can also occur in them.
- Inventory tracking is difficult.
- This causes delays in serving customers, billing errors, inventory mismatches, and poor coordination between staff (waiters, chefs, cashiers, and managers).
- Customers often face long waiting times for placing and receiving orders.
- Management faces challenges in monitoring employee activities, supplier deliveries, table bookings, and payments.
- Lack of a centralized system leads to difficulties in data tracking, reporting, and decision-making.
- In peak hours, customers wait 20 – 30 minutes just for billing.
- Current manual system cause inefficiency in order taking, billing, and inventory.

Definition of Problem: There is a need for a centralized Restaurant Management System (RMS) that integrates customer ordering, table booking, billing & payment, inventory & supplier management, employee monitoring, and reporting into a single platform.

The problem identified is that existing restaurant operations rely on manual processes, which result in inefficiencies in table booking, order taking, billing and inventory management. Therefore, there is a need for a centralized restaurant management system that automate these processes and ensures accuracy, speed and better customer satisfaction.

1.2 Problem Description

The existing restaurant operations are mostly managed through manual processes or partially computerized systems that work in isolation. This creates several challenges:

- Order Management Issues: Waiters take orders manually, which increases the chances of mistakes in recording customer preferences and leads to delays in communication with the kitchen staff.
- Billing & Payment Delays: Manual calculation of bills or using basic POS systems often causes errors in pricing, tax application, or discounts. Multiple payment modes (cash, card, UPI) are not handled efficiently.
- Inventory Mismanagement: Ingredients and raw materials are not tracked in real-time, leading to shortages, wastage, or over-purchasing from suppliers.
- Supplier Coordination Problems: Managing multiple suppliers without an integrated purchase order system causes delays and confusion in procurement.
- Table Booking & Reservation: Without a proper reservation system, tables may remain underutilized or get double-booked, leading to customer dissatisfaction.
- Employee & Role Management: Tasks of waiters, chefs, cashiers, and admin are not clearly tracked, leading to inefficiencies and lack of accountability.
- Reporting & Analysis: Managers cannot easily generate reports about sales, customer preferences, or supplier performance, which limits decision-making.

Summary:

The lack of a centralized, automated, and integrated Restaurant Management System results in inefficiency, errors, longer service time, and reduced customer satisfaction.

1.3 Fact Finding Techniques

To understand the current problems and gather accurate requirements for the Restaurant Management System, different fact-finding techniques were used:

1. Interviews
   - Conducted interviews with restaurant owners, managers, waiters, chefs, and cashiers to identify pain points in daily operations.
   - Example: Managers highlighted difficulties in inventory control, while waiters reported delays in order communication.
2. Questionnaires / Surveys
   - Distributed questionnaires to staff and customers to collect feedback about existing issues.
   - Customers emphasized the need for faster service and online table booking.
3. Observation
   - Directly observed restaurant operations during peak and non-peak hours.
   - Identified long waiting times for customers, manual errors in billing, and poor coordination between staff roles.
4. Document Analysis
   - Reviewed existing billing slips, purchase orders, inventory records, and reservation logs to analyse gaps and inconsistencies.
5. Research & Literature Study
   - Studied similar restaurant management software solutions to understand their features and limitations.
   - Helped identify additional requirements like multi-payment support, notification services, and analytics.

❖ Questionary:

Customer Survey:

How long do you usually wait for bill?

➔ On an average 10 – 15 minutes after finishing food.

Would you prefer online table booking?

➔ Yes, especially during weekends.

Staff Survey

How often do billing mistakes occurs?

➔ Atleast 2 – 3 times a day during rush hours.

Would a mobile/tablet order entry system help?

➔ Yes, it will save time and reduce errors.

Do you face problems in order taking and communication with the kitchen?

➔ Yes, many times chefs misread handwritten orders, which causes delays and wrong dishes.

These techniques ensured that system requirement are based on real-world needs of the restaurant covering all stakeholders (customer, staff, supplier and management).

1.4 Drawbacks of Existing System

The current manual or partially computerized systems used in restaurants have several limitations:

1. Time-Consuming Order Processing
   - Orders are taken manually by waiters and passed verbally or on paper to the kitchen.
   - This leads to delays, miscommunication, and wrong food being prepared.
2. Error-Prone Billing
   - Manual bill preparation increases chances of calculation mistakes, incorrect tax application, and missing discounts.
   - Multiple payment modes (cash, card, UPI) are not integrated.
3. Poor Inventory Management
   - No real-time tracking of raw materials.
   - Restaurants face issues of stockouts (running out of items) or wastage due to over-purchasing.
4. Supplier Handling Difficulties
   - Orders to suppliers are not recorded systematically.
   - Difficult to track which supplier delivered what, leading to disputes and confusion.
5. No Centralized Control
   - Each department (kitchen, billing, staff and inventory) works in isolation.

- Management cannot monitor operations effectively in real-time.

6. Inefficient Table Reservation
   - Customers cannot pre-book tables easily.
   - Often results in overbooking or underutilization of tables.
7. Lack of Reports & Analytics
   - Managers do not get automatic sales reports, inventory usage, or supplier performance insights.
   - This reduces efficiency in decision-making.
8. Customer Dissatisfaction
   - Delays in service, wrong orders, and poor table management directly affect customer experience.

These drawbacks highlight the urgent need for an integrated, automated Restaurant Management System that can streamline operations and improve efficiency.

1.5 Scope of the Proposed System

- Provide a centralized system to manage orders, billing, inventory, suppliers, employees, and reservations.
- Faster and accurate order processing with direct communication between waiter, kitchen, and cashier.
- Automated billing & multi-payment support (cash, card, UPI).
- Real-time inventory tracking with supplier integration.
- Table booking and reservation system for better customer experience.
- Employee management for assigning roles and monitoring tasks.
- Reports and analytics for sales, stock, and supplier performance.
- Customer satisfaction through faster service and error-free operations

The proposed Restaurant Management System will automate and integrate all key restaurant operations including table booking, order taking, billing, payment, and inventory management. Customers will be able to book tables in real time, while waiters can place orders directly into the system, which will instantly notify the chef. Bills will be generated automatically and linked to orders, allowing accurate and quick payments. The inventory module will track stock in real time, alerting the manager about low stock or expired items and enabling smooth purchase order generation. The admin will have access to reports on sales, expenses, and inventory for better decision-making. The system will provide role-based access to customers, staff, and managers, ensuring centralized and efficient restaurant management.

1.6 Feasibility Study

1. Technical Feasibility:
   - The system can be developed using widely available technologies like MERN stack, Django, or .NET.
   - Supports multiple platforms (desktop, tablet, smartphone, POS systems).
   - Database ensures reliability, normalization, and data consistency.
   - Cloud deployment options allow scalability and multi-branch support.
   - Supports integration with APIs (payment gateways, SMS/email, GST compliance).
2. Operational Feasibility:
   - Simple dashboards with role-based access for each type of user.
   - Minimal training required for staff.
   - Automation reduces manual workload.
   - Managers can monitor sales, stock, and performance in real time.
3. Economic Feasibility:
   - Cost-effective compared to manual systems.
   - Saves money by reducing wastage, incorrect billing, and mismanagement.
   - Development and maintenance costs are reasonable.
4. Schedule Feasibility:
   - Can be developed modularly in phases.
   - Estimated development time: 4–6 months.

❖ Risk and mitigation
Risk: Staff may resist using new system -> Sol: Training sessions
Risk: Cost overrun -> sol: Modular development.
Risk: Data security -> sol: Encryption + PCI DSS compliance.

Conclusion: The RMS is technically, operationally, economically, and practically feasible.

2. Requirement Specification

The requirement specification clearly defines the system's functional and non-functional needs, inputs, outputs, and end users to ensure accurate and efficient operations.

2.1 Data Requirements of the System

The system requires the following categories of data to function effectively:

1. Customer Data
   - Customer ID, Name, Contact Number, Email, Address
   - Table Booking details (date, time, number of persons)
   - Order history and payment details
2. Table Booking data
   - Booking ID, Customer ID (linked), Table number, Date and time of booking, Number of guests, Advance payment (if any)
   - Booking status (Confirmed, Cancelled, Completed)
3. Employee Data
   - Employee ID, Name, Role (Waiter, Chef, Cashier, Admin)
   - Contact Information
4. Menu Data
   - Item ID, Name, Category (starter, main course, dessert, beverage)
   - Price, Availability status
5. Order Data
   - Order ID, Customer ID, Ordered items, Quantity, Total amount
   - Order status (Pending, In Progress, Completed, Cancelled)
6. Billing & Payment Data
   - Bill ID, Order ID, Payment Mode (Cash, Card, UPI)
   - Tax, Discounts
7. Inventory Data
   - Item ID, Item Name, Stock Quantity, Reorder Level
   - Supplier ID linked to raw materials
8. Supplier Data
   - Supplier ID, Name, Contact Details
   - Supplied items list, Payment records
9. System Data
   - Login credentials (Username, Password, Role)
   - Access permissions for different users

This ensures that all major operations (ordering, billing, inventory, employee, supplier management, reservations) are covered with accurate and secure data.

2.2 Identify End Users of the System

The RMS will be used by different categories of users, each with specific roles:
The Restaurant Management System (RMS) will be used by different end users. Each user has distinct roles, responsibilities, and access privileges.

1. Customers
- Role: Book tables, place orders, make payments.
- Access: Limited to booking module, order placement, and bill/payment view.

2. Waiters
- Role: Enter customer orders, update booking details, serve customers.
- Access: Order entry, order status tracking and bill viewing (not editing).

3. Chefs / Kitchen Staff

- Role: Receive and process orders sent by waiters.
- Access: Order list, preparation status update.

4. Cashier
- Role: Generate bills, process payments, Book tables, update payment status.
- Access: Billing and payment modules.
- Benefit: Accurate and quick billing, reduces manual calculations.

5. Inventory Manager
- Role: Monitor and manage stock, update received supplies, check low-stock alerts.
- Access: Inventory module, supplier information, reorder tracking.
- Benefit: Prevents shortages/wastage, ensures smooth kitchen operations.

6. Administrator (Admin)
- Role: Manage overall system, maintain user accounts, generate reports, monitor sales and performance.
- Access: Full system access (bookings, orders, billing, inventory, user management, reporting).
- Benefit: Centralized control, better decision-making, improved efficiency.

Summary:
The RMS will be used by customers, waiters, chefs, cashiers, inventory managers, and administrators.
Each end user is assigned a specific role with defined access privileges to ensure secure, efficient, and organized system operation.

2.3 Input Data to the System

1. The Restaurant Management System (RMS) involves several major processes. Each process is specified below with its inputs, operations, and outputs.Customer Inputs
   - Personal details (Name, Contact, Email)
   - Table booking request (Date, Time, Number of seats)
   - Food orders (Menu items, Quantity, Special instructions)
   - Payment details (Mode of payment: Cash, Card, UPI)
2. Waiter / Staff Inputs
   - Customer order details (entered into the system if customer orders offline).
   - Updates to order status (Pending → In Progress → Completed).
   - Table status (Available, Reserved, Occupied).
3. Chef Inputs
   - Order completion status.
   - Ingredient usage updates (linked to inventory).
4. Cashier Inputs
   - Bill generation details (Order ID, Taxes, Discounts).
   - Payment confirmation and transaction records.
6. Manager / Admin Inputs
   - Employee details (ID, Role, Salary, Shift).
   - Menu updates (Add/Edit/Remove items, Prices).
   - Send inventory items purchase order.
   - Supplier details (Name, Items supplied, Contact).
7. Inventory manager
   - Inventory updates (Stock levels, Reorder points).
   - Inventory items details (quantity, expiry).
   - Add items, remove items.

These inputs ensure smooth functioning of all modules: booking, ordering, billing, inventory, and reporting.

2.4 Output Information from the System

- For Customers
  - Booking confirmation (table number, date/time, advance payment).

- Order receipt (items, quantity, price, special requests
- Final bill/invoice (subtotal, tax, discount, total amount, payment mode).
- Menu display (categorized food & drinks).

- For Waiters
  - Assigned tables list.
  - Current orders for their tables (status: preparing, ready, served).
  - Pending customer requests (e.g., modifications, special instructions).

- For Chefs
  - Kitchen order ticket (KOT): list of items to prepare per order.
  - Special requests attached to dishes.
  - Pending vs. completed orders.

- For Cashiers
  - Bill summary (order details, table, amount, payment status).
  - Daily sales report.
  - Discounts and promotions applied.

- For Admin
  - Daily/Monthly sales report.
  - Customer list and booking history.
  - Employee management report (active waiters, chefs, cashiers).
  - Supplier performance report (deliveries, delays).
  - Purchase order report (supplies ordered, received, pending).
  - Inventory report (current stock levels, items nearing expiry).

- For Inventory manager
  - Inventory report (current stock levels, items nearing expiry).
  - Inventory item details.
  - Inventory details.

- For Suppliers
  - Purchase orders (items requested, quantities, delivery deadlines).
  - Payment details for supplies delivered.

2.5 Functional/Non-Functional/Processing Requirements of the system

Functional Requirements

- Table Booking: Book tables, prevent double-booking, generate booking confirmation
- Order Management: Place orders, notify kitchen, update order status.
- Billing and Payment: Auto-generate bills, apply taxes/discounts, accept multiple payment modes.
- Inventory Management: Deduct stock, monitor levels, generate alerts and reorder requests.
- User Management: Role-based login, authentication, access control.
- Report Generation: Generate sales, inventory, booking, and performance reports.

Non-Functional Requirements

- **Reliability**: 99.9% uptime, backups every 12 hrs.
- **Security**: PCI DSS compliance, role-based access, encryption, use tokens & sessions.
- **Usability**: Mobile-friendly UI, simple dashboards for staff.
- **Maintainability**: Modular, easy updates, logging.
- **Scalability**: Support growth (more tables, menu items, and staff).

Processing Requirements

1. Table Booking:
   - Input: Customer request (date, time, number of guests).
   - Process: Check availability → confirm booking.
   - Output: Booking confirmation.
2. Order Cycle:
   - Input: Customer places order via waiter or self-service.
   - Process: Order sent to kitchen → chef updates status.
   - Output: Updated order status.
3. Billing & Payment:
   - Input: Order details, taxes, discounts, payment mode.
   - Process: System generates bill and processes payment.
   - Output: Invoice and transaction record.
4. Inventory Management:
   - Input: Ingredient usage, stock levels, supplier deliveries.
   - Process: Stock updated → reorder alerts → purchase orders.
   - Output: Updated stock report.
5. Supplier & Purchase Orders:
   - Input: Admin/inventory manager creates purchase order.
   - Process: Supplier delivers → stock updated.
   - Output: Updated inventory and supplier records.
6. Employee & User Management:
   - Input: Admin adds employees with roles.
   - Process: Role-based authentication applied.
   - Output: Secure access and monitoring.
7. Reporting & Analytics:
   - Input: Transaction data.
   - Process: Data aggregated → reports generated.
   - Output: Sales, inventory, supplier, and customer reports.

Summary: The system ensures smooth flow from booking → ordering → billing → inventory → suppliers → reporting.

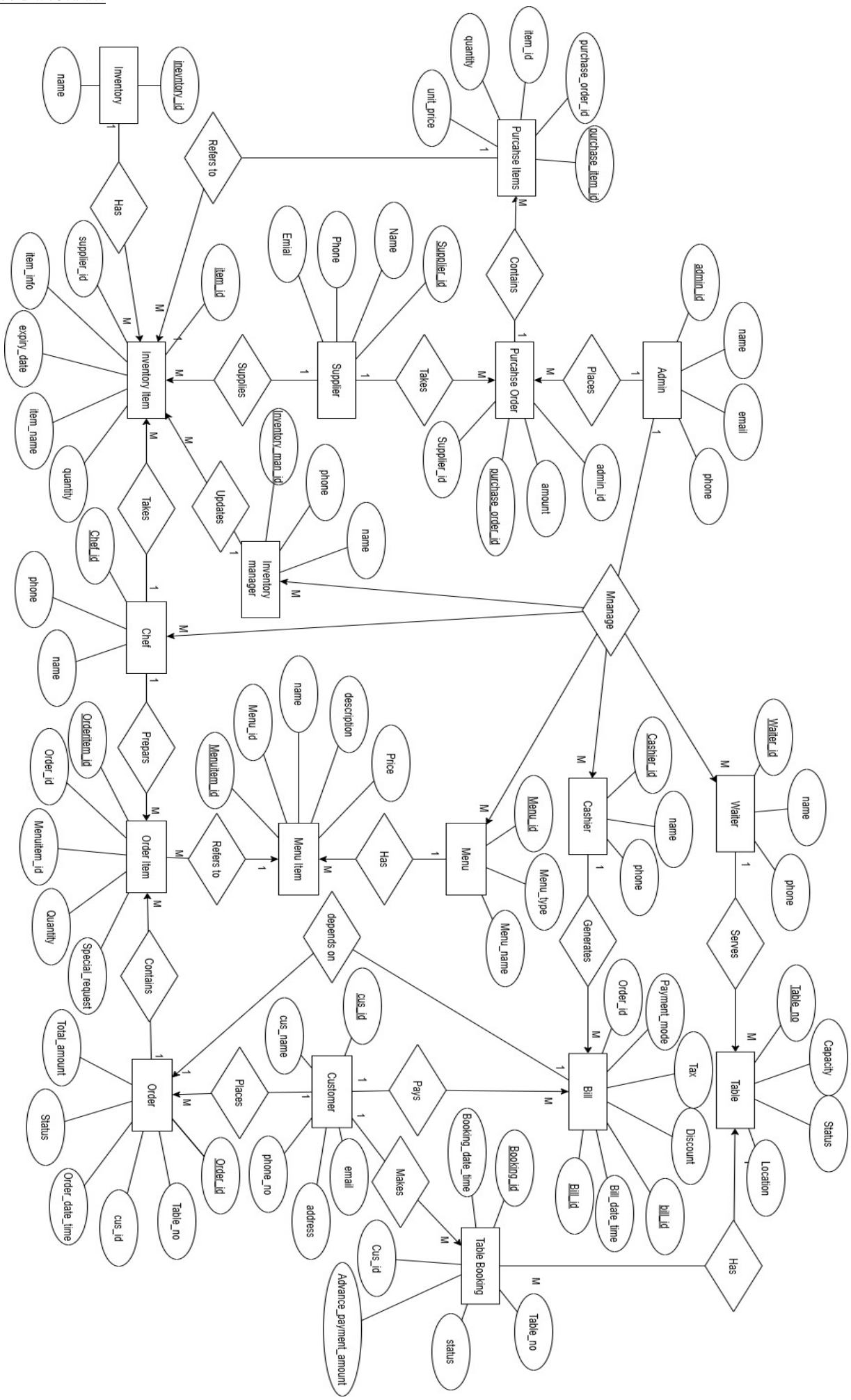
3 DATABASE DESIGN

The database design identifies entities, attributes, relationships, and applies normalization up to BCNF to maintain data integrity and eliminate redundancy.

3.1 IDENTIFY THE ENTITIES AND THE ATTRIBUTES

Sr. NO	Entities	Attributes
1.	Customer	Cus_id (PK), Cus_name, Email, Phone_no, Address
2.	Table	Table_no(PK), Capacity, Status, Location
3.	Table Booking	Booking_id (PK), Cus_id (FK), Table_no (FK), Booking_date_time, Advance_payment_amount, Status
4.	Order	Order_id (PK), Cus_id (FK),Table_no (FK), Order_date_time, Status, Total_amount
5.	Order Item	Orderitem_id (PK), Order_id (FK), Menuitem_id (FK), Quantity, Special_request,
6.	Menu	Menu_id (PK), Menu_name, Menu_type
7.	Menu Item	Menuitem_id (PK), Menu_id (FK), Name, Description, Price
8.	Bill	Bill_id (PK), Order_id (FK), Bill_date_time, Discount, Tax, Payment_mode,
9.	Admin	Admin_id (PK), Name, Phone, Email
10.	Chef	Chef_id (PK), Name, Phone
11.	Waiter	Waiter_id (PK), Name, Phone
12.	Cashier	Cashier_id (PK), Name, Phone
13.	Inventory Manager	Inventory_man_id (PK), Name, Phone
14.	Inventory	Inventory_id (PK), Name
15.	Inventory Item	Item_id (PK),Inventory_id (FK), Supplier_id (FK), Item_name, Item_info, Quantity, Expiry_date
16.	Supplier	Supplier_id (PK), Name, Phone, Email,
17.	Purchase Order	Purchase_order_id (PK), Supplier_id (FK), Admin_id (FK), Amount
18.	Purchase Items	Purchase_item_id (PK), Purchase_order_id (FK), Item_id (FK), Quantity, Unit_price

3.2 E-R Diagram



3.3 IDENTIFYING ALL TABLES, FIELDS, RELATIONSHIP BETWEEN TABLES, ETC.

- Tables and fields

Sr. NO	Entities	Attributes
1.	Customer	Cus_id (PK), Cus_name, Email, Phone_no, Address
2.	Table	Table_no(PK), Capacity, Status, Location
3.	Table Booking	Booking_id (PK), Cus_id (FK), Table_no (FK), Booking_date_time, Advance_payment_amount, Status
4.	Order	Order_id (PK), Cus_id (FK),Table_no (FK), Order_date_time, Status, Total_amount
5.	Order Item	Orderitem_id (PK), Order_id (FK), Menuitem_id (FK), Quantity, Special_request,
6.	Menu	Menu_id (PK), Menu_name, Menu_type
7.	Menu Item	Menuitem_id (PK), Menu_id (FK), Name, Description, Price
8.	Bill	Bill_id (PK), Order_id (FK), Bill_date_time, Discount, Tax, Payment_mode,
9.	Admin	Admin_id (PK), Name, Phone, Email
10.	Chef	Chef_id (PK), Name, Phone
11.	Waiter	Waiter_id (PK), Name, Phone
12.	Cashier	Cashier_id (PK), Name, Phone
13.	Inventory Manager	Inventory_man_id (PK), Name, Phone
14.	Inventory	Inventory_id (PK), Name
15.	Inventory Item	Item_id (PK),Inventory_id (FK), Supplier_id (FK), Item_name, Item_info, Quantity, Expiry_date
16.	Supplier	Supplier_id (PK), Name, Phone, Email,
17.	Purchase Order	Purchase_order_id (PK), Supplier_id (FK), Admin_id (FK), Amount
18.	Purchase Items	Purchase_item_id (PK), Purchase_order_id (FK), Item_id (FK), Quantity, Unit_price

- Relationships

1. **Waiter – Table** → M:N (One waiter serves many tables, one table can be served by many waiters)
2. **Customer – Order** → 1:M (One customer places many orders)
3. **Customer – Bill** → 1:M (One customer pays many bills)
4. **Customer – Table_Booking** → 1:M (One customer makes many bookings)
5. **Cashier – Bill**→ 1:M (One cashier generates many bills)

6. **Bill – Order** → 1:1 (One Bill depens on One order)
7. **Table – Table_Booking** → 1:M (One table can be booked many times)
8. **Order – Order_Item** → 1:M (One order contains many items)
9. **Order_Item – Menuitem** → M:1 (One menu item can appear in many orders)
10. **Menu – Menu_Item** → 1:M (One menu contains many items)
11. **Chef – Order_Item** → 1:M (One chef prepares many order items)
12. **Admin – Menu** → 1:M (One admin manages many menus)
13. **Admin – Employee** → 1:M (One admin manages many employee)
14. **Admin – Purchase_Order** → 1:M (One admin issues many purchase orders)
15. **Supplier – Purchase_Order** → 1:M (One supplier takes many purchase orders)
16. **Supplier – Inventory_Item**→ 1:M (One supplier supplies many Inventory items)
17. **Purchase_Order – Purchase_Item** → 1:M (One purchase order includes many items)
18. **Inventory – Inventory_Item** → 1:M (One inventory contains many items)
19. **Inventory_Item – Purchase_Item** → 1:M (One inventory item can appear in many purchase orders)
20. **Chef - Inventory_Item** → 1:M (One chef uses many inventory items)
21. **Inventory_manager – Inventory_items**→ 1:M (One Inventory managers updates many inventory items)

3.4 Normalize database

To ensure that the database design is efficient and free from redundancy, the RMS database is normalized up to BCNF.

Unnormalized Table

Order_ID	Cus_ID	Cus_Name	Table_No	Order_Item	Total_Amount
O101	C01	Anshu	T05	Pizza,2,200; Coke,1,50;	450

Problem: Order_Item has repeating groups (multiple values in one cell). Total_amount is derived, leading to redundancy.

Step 1: 1NF (First Normal Form)

Definition: A relation is in 1NF, if every row contains exactly one value for each attribute.

Rule: Table columns should contain atomic data.

Converted table (1NF):

Order_ID	Cus_ID	Cus_Name	Table_No	Item_Name	Qty	Price
O101	C01	Anshu	T05	Pizza	2	200
O101	C01	Anshu	T05	Coke	1	50

Repeating groups removed. But still has redundancy (Cus_Name only depends on Cus_ID).

All attributes are atomic. The Table is in **1NF**.

Step 2: 2NF (Second Normal Form)

Definition: A relation is in 2NF, if it is in 1NF and every non-key attribute is fully functionally dependent on the primary key of the relation.

Rule: Must be in 1NF, and no partial dependency.

Split data into separate tables: Order, Order_Item, and Customer.

Order Table:

Order_ID (PK)	Cus_ID (FK)	Table_No
O101	C01	T05

Order_Item Table:

Order_Item_id	Order_ID(FK)	Item_Name	Qty	Price
23	O101	Pizza	2	200
24	O101	Coke	1	50

Customer Table:

Cus_ID (PK)	Cus_Name	Phone
C01	Anshu	9579695156

Partial dependency removed.

No composite primary keys used table is in **2NF**.

Step 3: 3NF (Third Normal Form)

Definition: A relation is in 3NF, if it is in 2NF and no non-key attribute of the relation is transitively dependent on the primary key.

Rule: Must be in 2NF, and no transitive partial dependency.

Price and item_name should not be in Order_item because they depend on Menu_item, not directly on Order.

Menu_Item Table:

Menu_Item_ID (PK)	Item_Name	Price
I01	Pizza	200
I02	Coke	50

Order_Item Table (Updated):

Order_item_id(PK)	Order_ID (FK)	Menu_Item_ID (FK)	Qty
23	O101	I01	2
24	O101	I02	1

Transitive dependencies removed. Tables are in **3NF.**

Step 4: BCNF (Boyce-Codd Normal Form)

Definition: A relation R is said to be in BCNF, if and only if every determinant is a candidate key.

Rule: Must be in 3NF, and no dependency where a non-key attribute determines part of the relation.

Now, all determinants are candidate keys. No anomalies remain.

The Table is already in BCNF.

Final Normalized Schema

Customer(Cus_ID PK, Cus_Name)
Order(Order_ID PK, Cus_ID FK → Customer, Table_No)
Menu_Item(Item_ID PK, Item_Name, Price)
Order_Item(Order_ID PK, FK → Order, Item_ID PK, FK → Menu_Item, Qty)

By following these steps the database is normalized to BCNF.

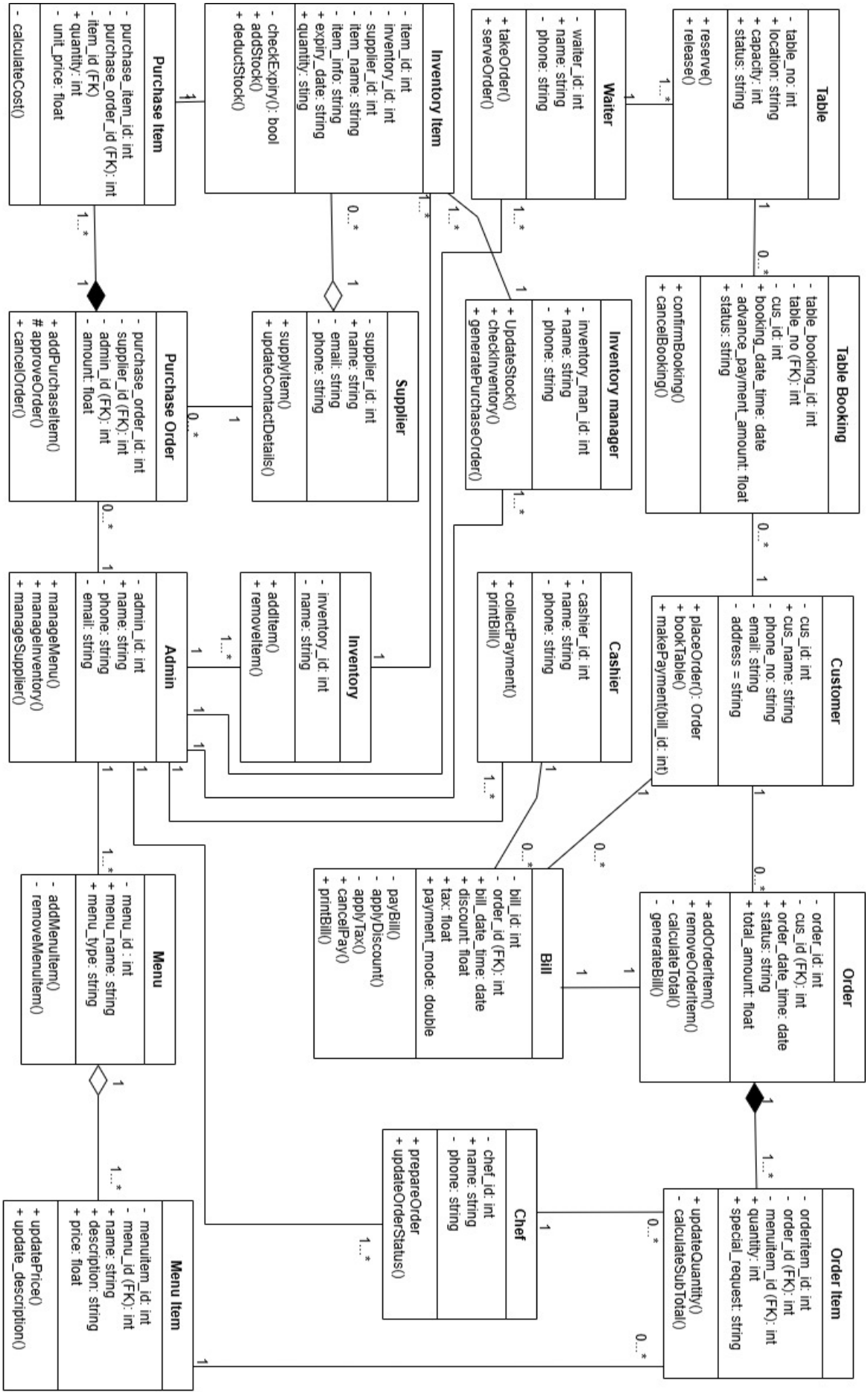
Final Normalized database

Sr. NO	Entities	Attributes
1.	Customer	Cus_id (PK), Cus_name, Email, Phone_no, Address
2.	Table	Table_no(PK), Capacity, Status, Location
3.	Table Booking	Booking_id (PK), Cus_id (FK), Table_no (FK), Booking_date_time, Advance_payment_amount, Status
4.	Order	Order_id (PK), Cus_id (FK),Table_no (FK), Order_date_time, Status, Total_amount
5.	Order Item	Orderitem_id (PK), Order_id (FK), Menuitem_id (FK), Quantity, Special_request,
6.	Menu	Menu_id (PK), Menu_name, Menu_type
7.	Menu Item	Menuitem_id (PK), Menu_id (FK), Name, Description, Price
8.	Bill	Bill_id (PK), Order_id (FK), Bill_date_time, Discount, Tax, Payment_mode,
9.	Admin	Admin_id (PK), Name, Phone, Email
10.	Chef	Chef_id (PK), Name, Phone
11.	Waiter	Waiter_id (PK), Name, Phone
12.	Cashier	Cashier_id (PK), Name, Phone
13.	Inventory Manager	Inventory_man_id (PK), Name, Phone
14.	Inventory	Inventory_id (PK), Name
15.	Inventory Item	Item_id (PK),Inventory_id (FK), Supplier_id (FK), Item_name, Item_info, Quantity, Expiry_date
16.	Supplier	Supplier_id (PK), Name, Phone, Email,
17.	Purchase Order	Purchase_order_id (PK), Supplier_id (FK), Admin_id (FK), Amount
18.	Purchase Items	Purchase_item_id (PK), Purchase_order_id (FK), Item_id (FK), Quantity, Unit_price

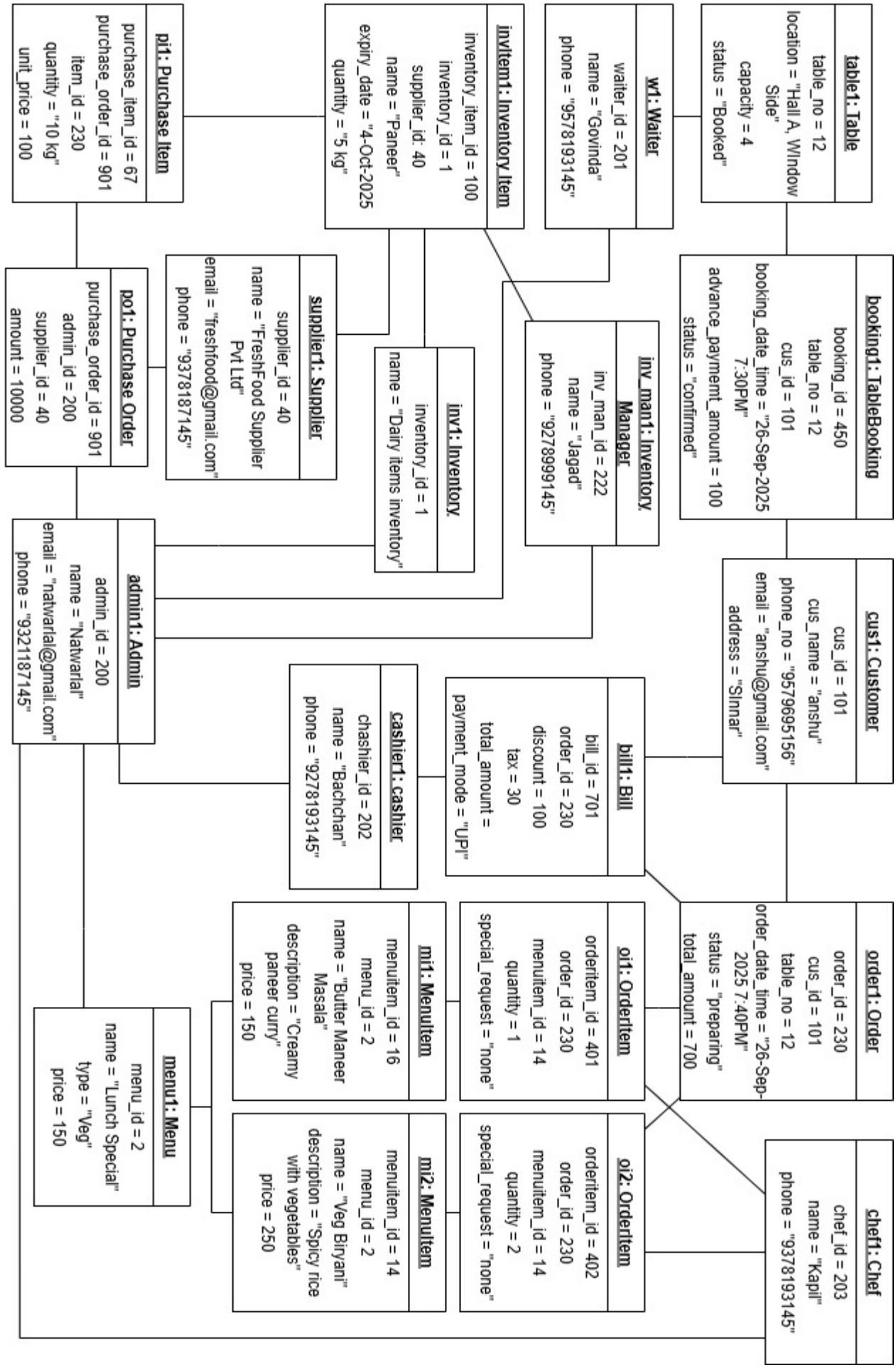
4. SYSTEM DESIGN

The system design translates requirements into UML diagrams—use case, activity, class, sequence, state chart, component, and deployment—that visually represent the complete functioning of the RMS.

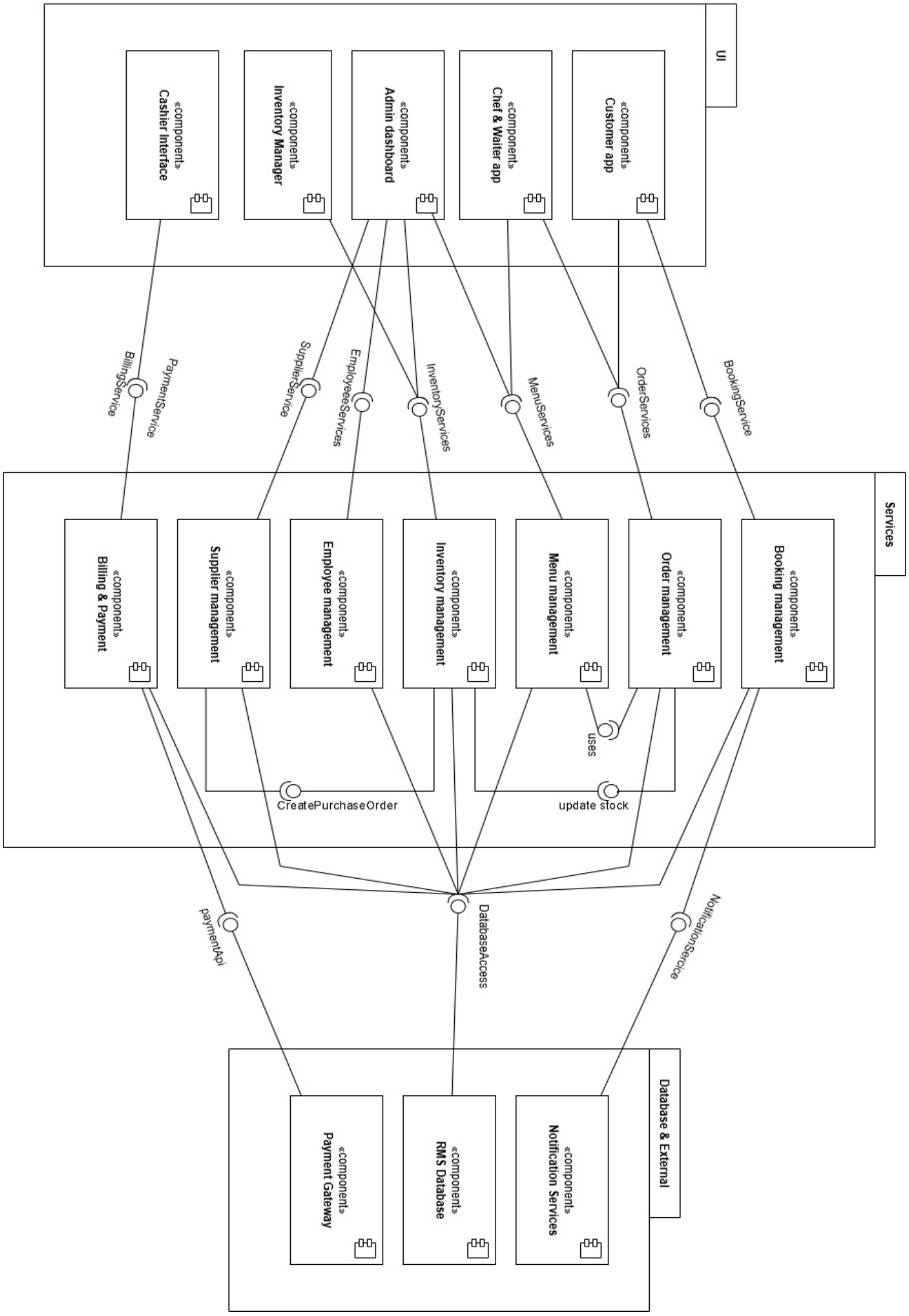
3.4 CLASS DIAGRAM



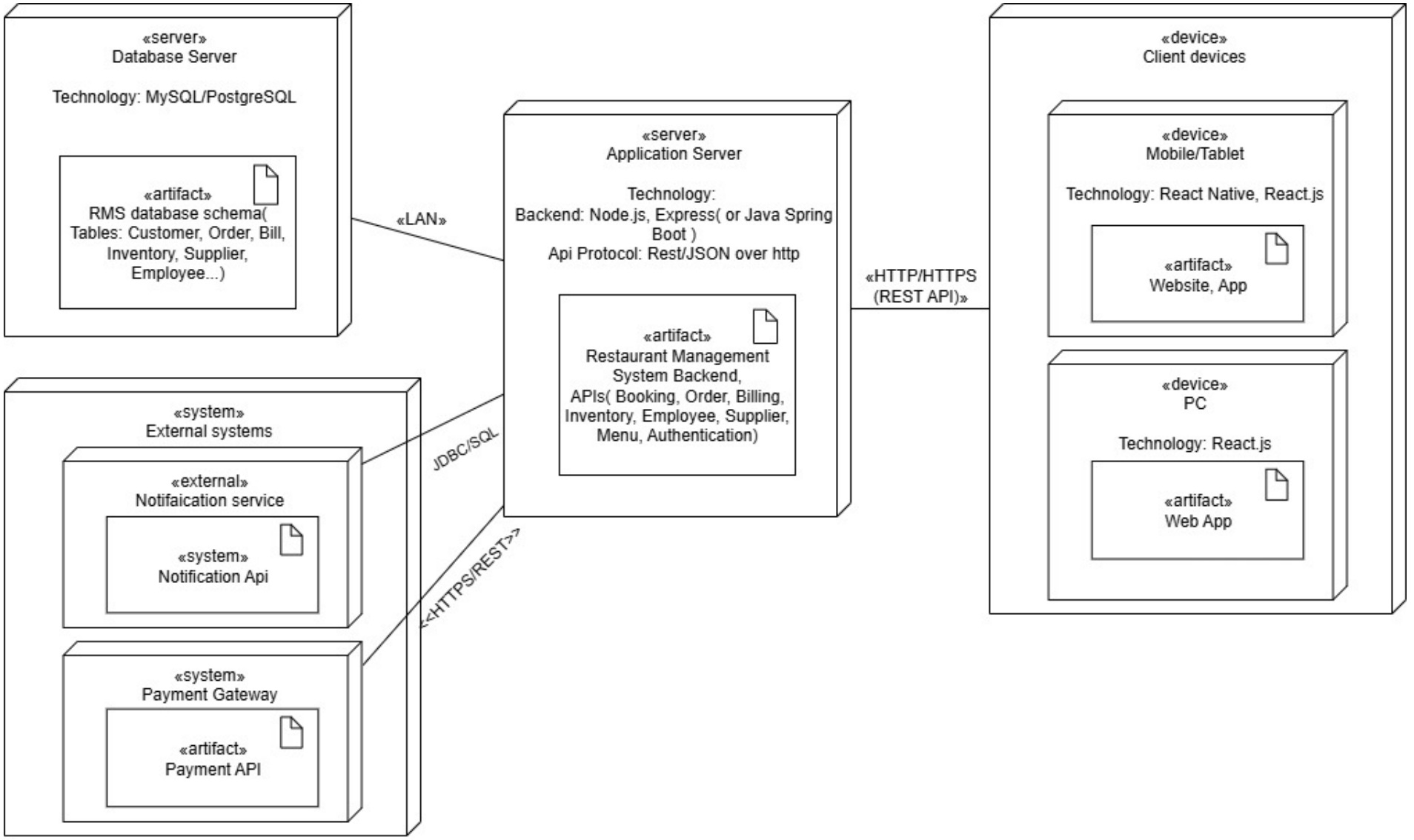
3.5 Object Diagram



3.6 COMPONENT DIAGRAM



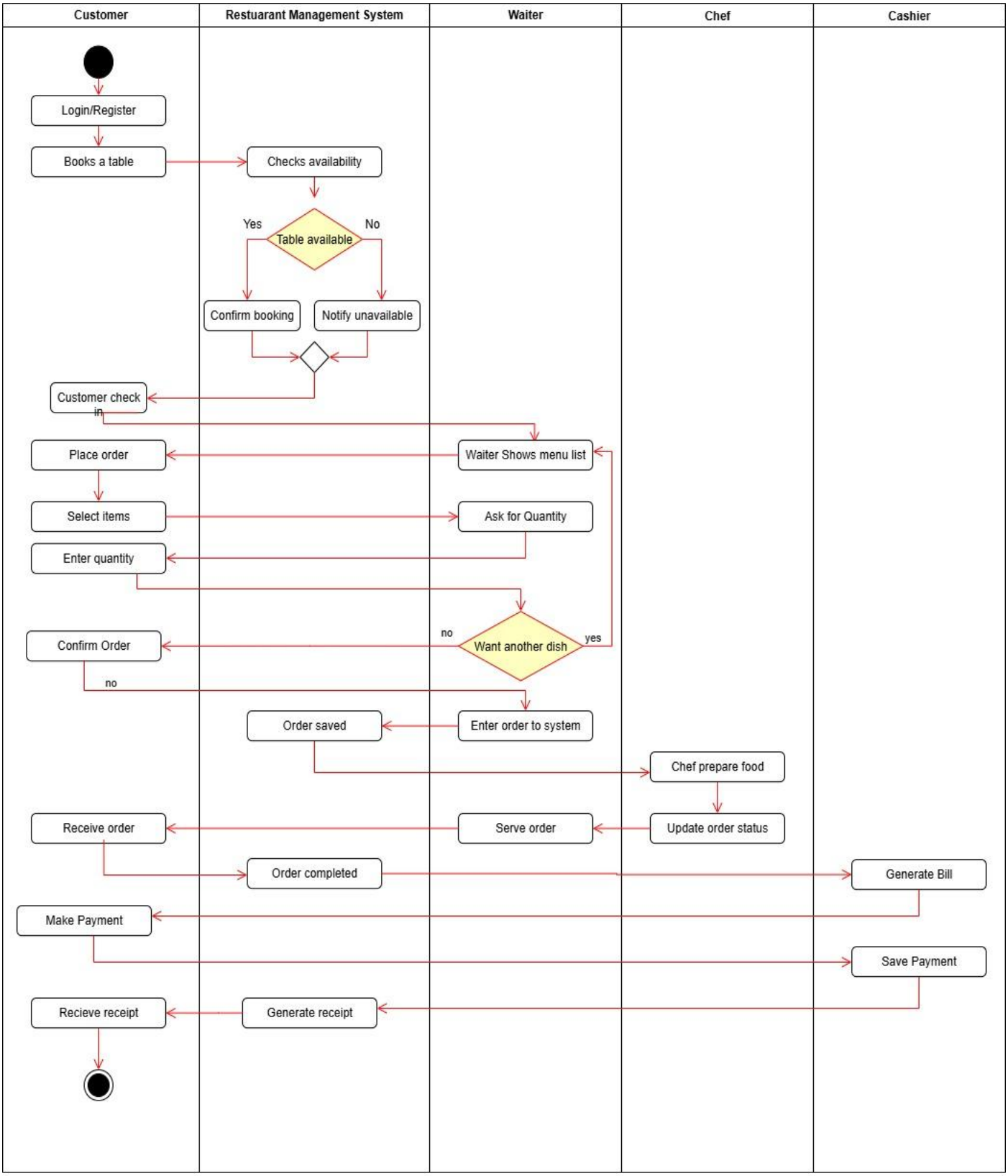
3.7 Deployment Diagram



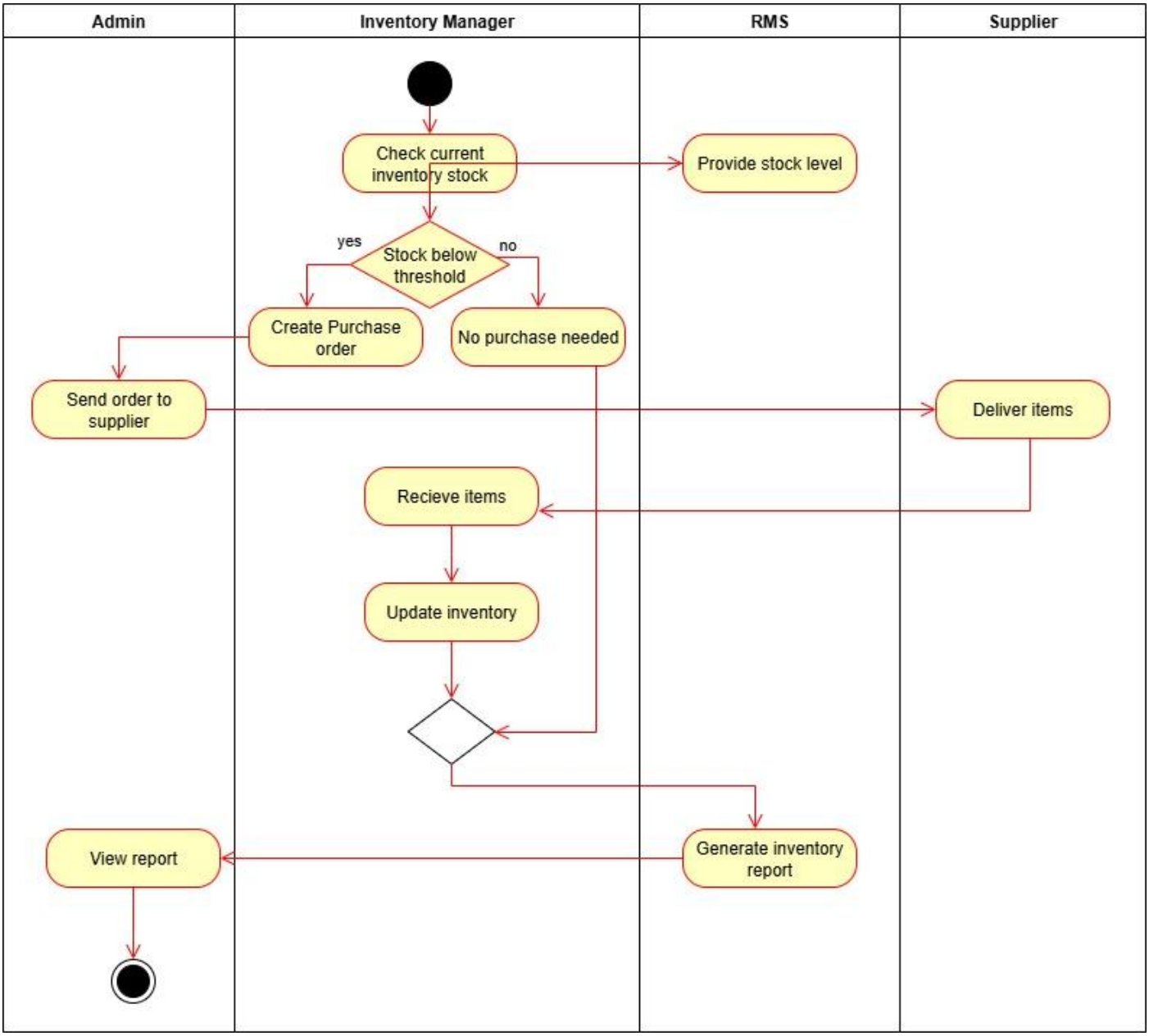
3.8 USE CASE DIAGRAM

3.9 Activity Diagram

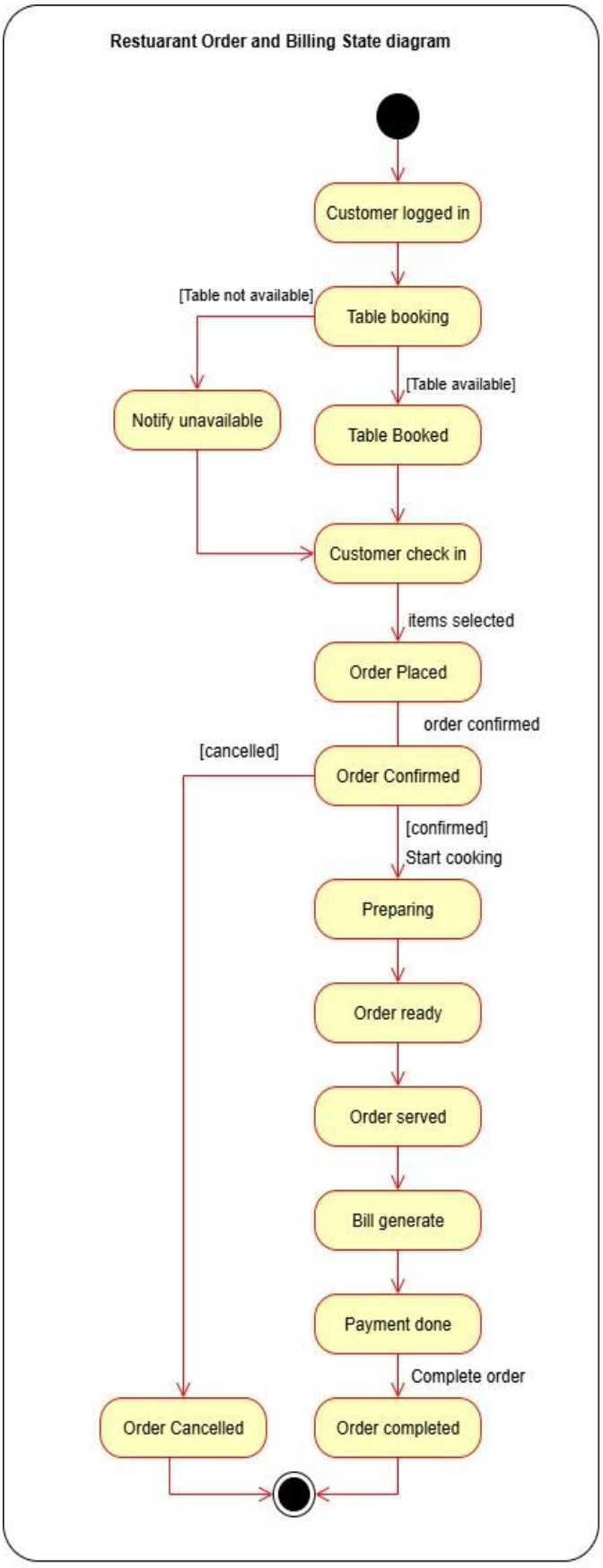
Table Booking and order Activity Diagram

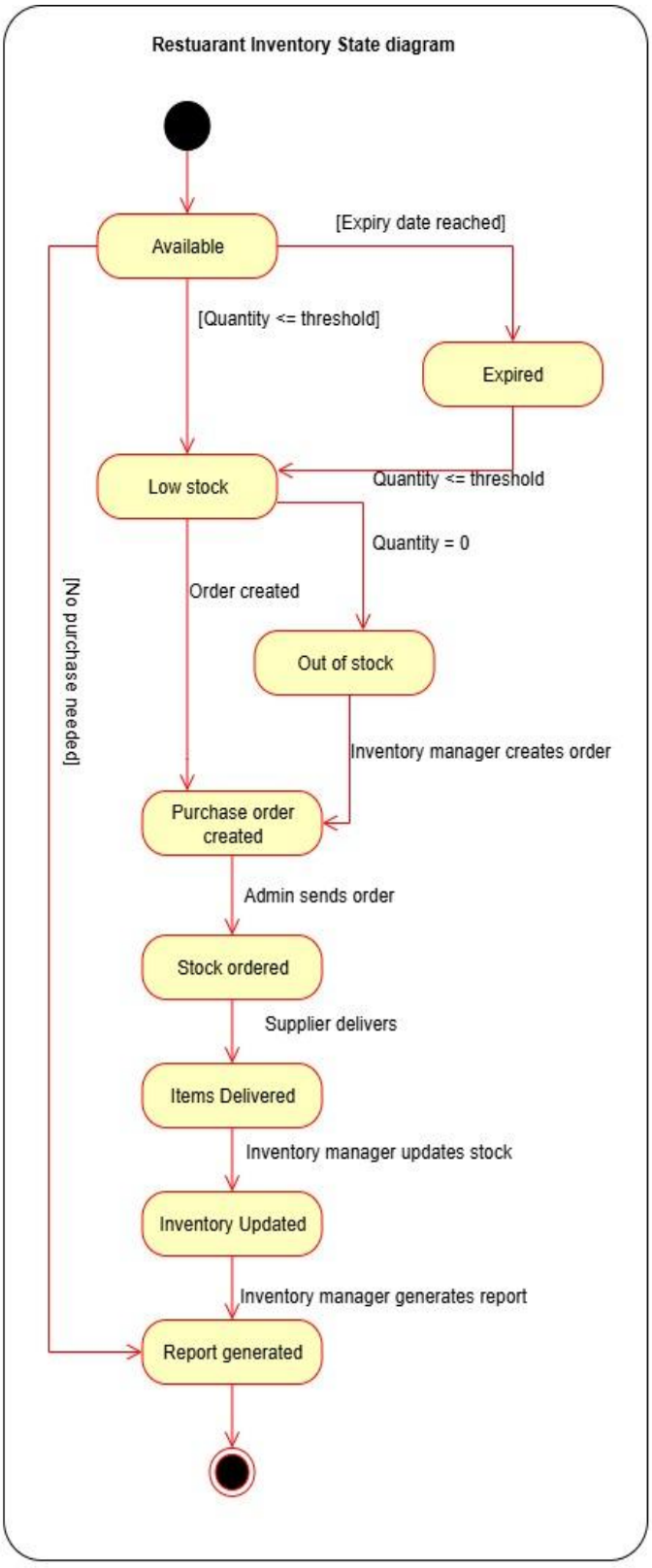


Inventory Activity Diagram

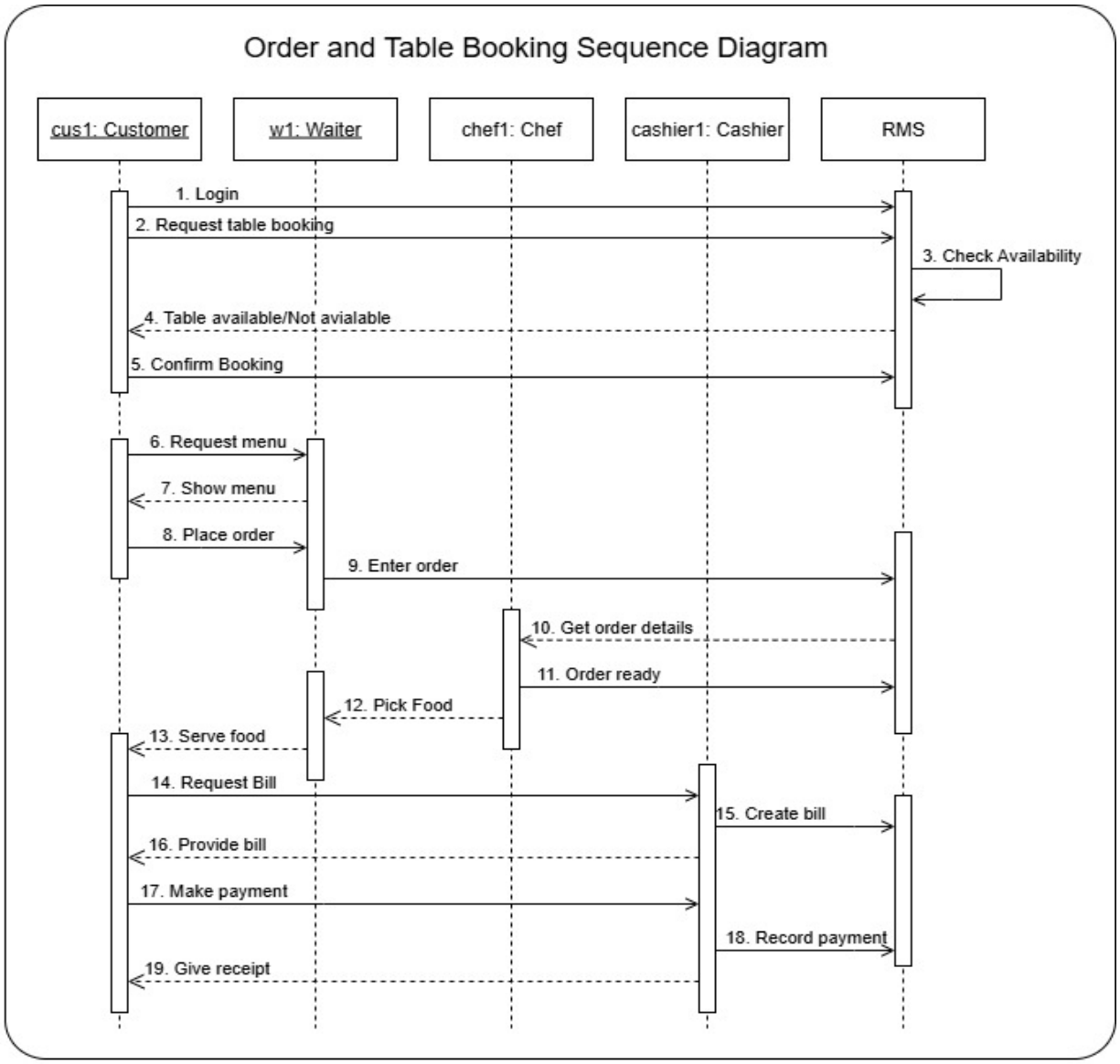


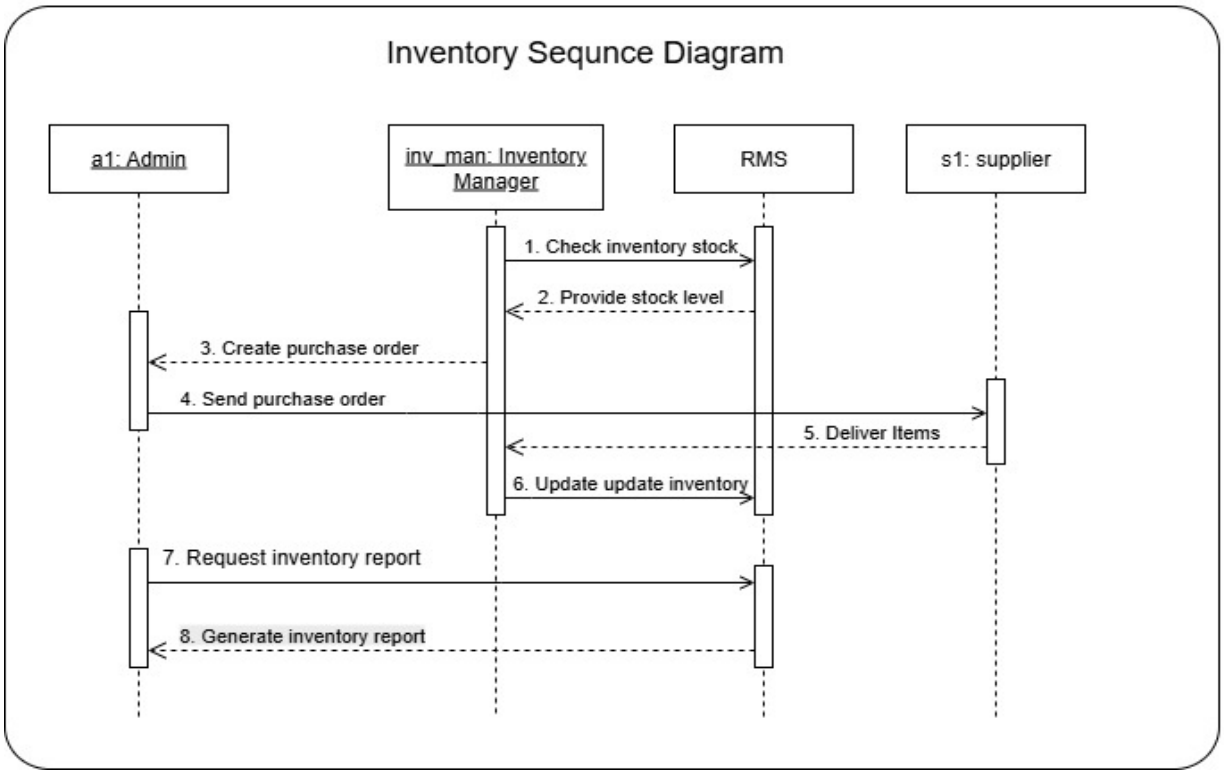
3.10 STATE CHART DIAGRAM



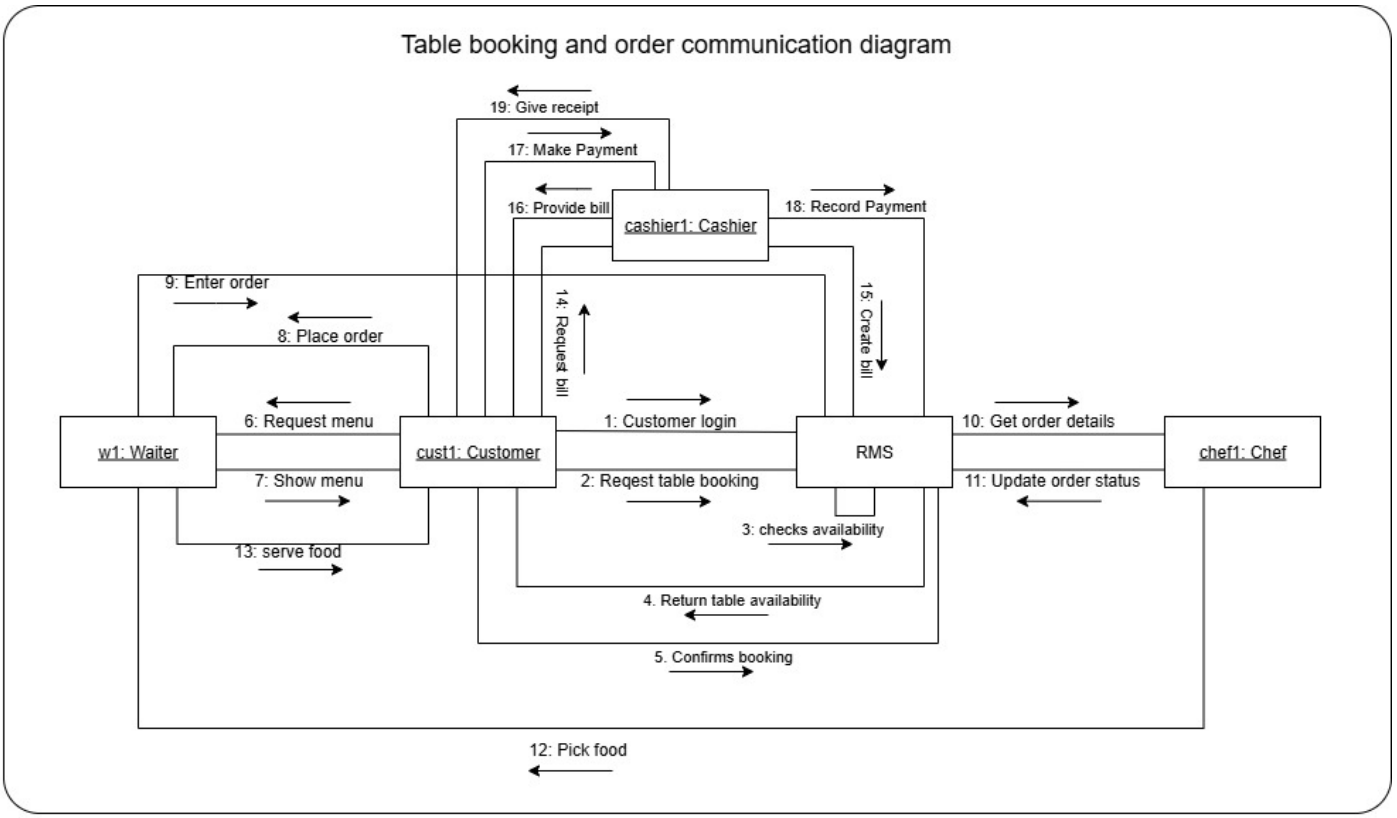


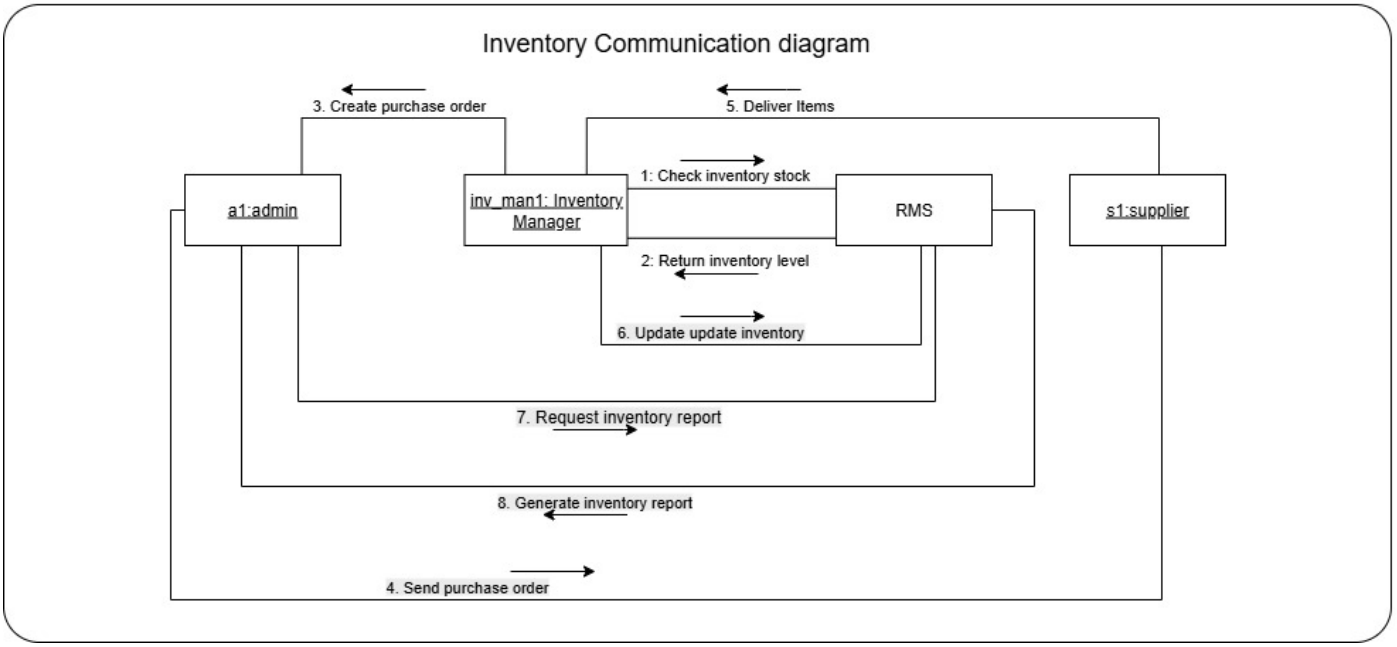
3.11 SEQUENCE DIAGRAM





4.9 Communication/Collaboration Diagram





CONCLUSION

The Restaurant Management System (RMS) designed in this project to addresses the major challenges faced by restaurants in managing their daily operations. By integrating table booking, order management, billing, payment, inventory, supplier coordination, employee monitoring, and reporting into a single centralized platform.

The system is designed to eliminate the inefficiencies of manual or partially computerized methods.

The system ensures:
- Faster and error-free order processing with direct communication between waiters, chefs, and cashiers.
- Accurate billing and support for multiple payment methods.
- Real-time inventory tracking with automated low-stock alerts and supplier integration.
- Organized table booking and reservations to reduce customer waiting time and prevent double-bookings.
- Clear role-based access for customers, employees, and administrators.
- Powerful reporting and analytics to assist management in decision-making.

Through careful requirement analysis, system design, normalization up to BCNF, and development of UML diagrams, the project provides a robust blueprint for implementation. The feasibility study confirms that the system is technically, operationally, and economically viable for deployment.

The proposed RMS will not only enhance operational efficiency but also improve customer satisfaction by providing faster service, minimizing errors, and streamlining restaurant operations. With future enhancements such as multi-branch support, online customer ordering, and advanced analytics.

REFERENCES / BIBLIOGRAPHY

- Korth, Henry F., Silberschatz, Abraham, and Sudarshan, S. Database System Concepts. McGraw-Hill Education, 7th Edition, 2019.
- Sommerville, Ian. Software Engineering. Pearson, 10th Edition, 2016.
- Pressman, Roger S. Software Engineering: A Practitioner's Approach. McGraw-Hill, 8th Edition, 2014.
- Kendall, Kenneth E., and Kendall, Julie E. Systems Analysis and Design. Pearson, 9th Edition, 2019.

- Official Documentation of MySQL Database. [Online]. Available at: https://dev.mysql.com/doc/

- YouTube channels:
    a. 5 Minutes Engineering for diagrams
    b. Edureka for database normalization
    c. Gate smashers for diagrams

- Plantuml, planttext is used for to take the reference and understand diagram and its elements more clearly.