# IT 314 - Software Engineering
## Lab 6 -  Group 14
### Date -  Mar 23, 2023

| Team Members | Student ID |
|---|---|
| Pulkit Khandelwal | 202001120 |
| Shreyansh Khemesara | 202001121 |
| Prajapati Vrutik Navneetbhai(Team Leader) | 202001124 |
| Baraiya Dhruv Pravinbhai | 202001140 |
| Mandaviya Soham Hiteshbhai | 202001142 |
| Shah Nishadhkumar Nirajbhai | 202001151 |
| Buddhdev Harsh Nitesh | 202001157 |
| Chhagani Krunal Ajaybhai | 202001158 |
| Thakor Harshal Dipaksinh | 202001169 |

# Domain Analysis :

## Introduction:
Domain analysis models provide a systematic way to analyze the key components, concepts, and relationships within a specific domain.
Here , the domain would be the automatic management of the restaurant.

## Identify Entity, Boundary and Control objects
**Entity Objects:**
1. **tables:** A table is an entity which is used to specify sitting availability.
2. **orders:** An order is an entity which is used to specify the order details of customers.
3. **users:** A user is an entity which describes the staff member who has taken the order from the customer.
4. **products:** A product is an entity which specifies the different menu items which are available in the restaurant.
5. **categories:** Category entity represents the different categories of the menu items.
6. **order_of_products:** This entity is used to specify the product details corresponding to the specific order.

**Attributes:**
1. **tables:** id, name , occupied
2. **orders:** id, total , isPaid, delivery , note, user_id, table_id
3. **users:** id, name , email, password, is_admin,
4. **products:** id , name , price , category_id, stock
5. **categories:** id, name
6. **order_of_proucts:** id, order_id, product_id, quantity

**Boundary Objects :**

Boundary objects are objects or concepts that can be shared and understood by different communities or groups, allowing them to collaborate and communicate effectively
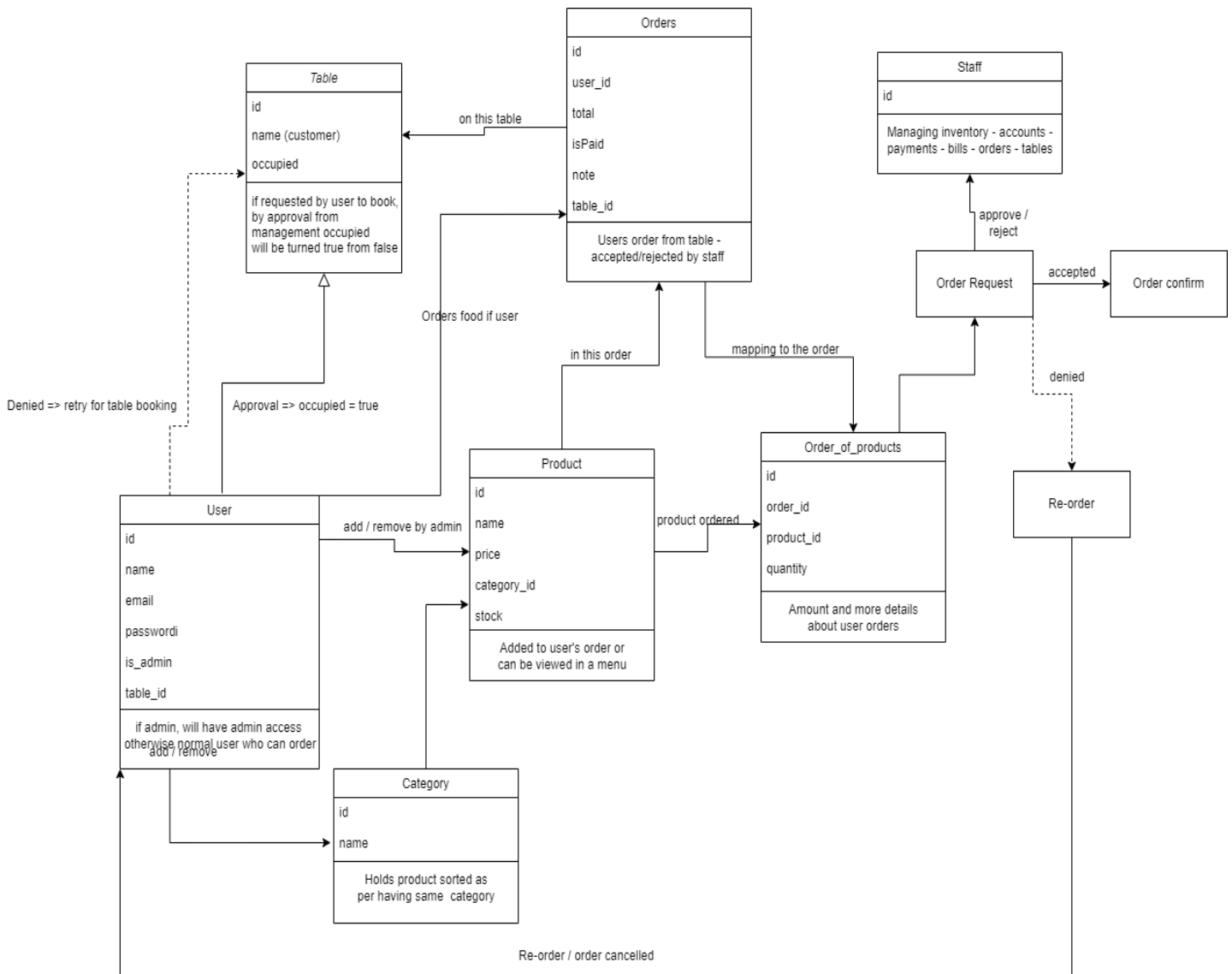
1. **Order management:** An order management allows restaurant staff to receive and process customer orders, track inventory, and manage deliveries. This can be a shared boundary object between restaurant staff, and delivery drivers, as it provides a common interface for communication and coordination.

2. **Menu database:** A menu database is a centralized repository of all the food and beverage items available in a restaurant, along with their prices and descriptions. This database can be accessed by admin and restaurant staff to ensure consistency and accuracy in ordering and preparation.

3. **Payment mechanism :** This is used to process transactions and manage customer payments. This can serve as a boundary object between restaurant staff and customers, as it provides a user-friendly interface for ordering and payment.

4. **Customer feedback mechanism :** A customer feedback mechanism is a tool that allows customers to provide feedback on their dining experience, including food quality, service, ambiance, and overall satisfaction. This can be used as a boundary object between customers and restaurant management, enabling a continuous feedback loop to improve service.

**Control Objects:**

The control object represents the internal processes and rules that govern the behavior of the system. It coordinates the interactions between different objects and manages the flow of data within the system.

1. **Inventory Manager :** This control object helps to track the inventory levels of ingredients, and alerts when stock needs to be replenished.
2. **Customer Relationship Manager:** This control object helps in managing customer data, preferences, and feedback.
3. **Table Manager:** This control object manages table reservations and seating arrangements.
4. **Employee Manager:** This control object manages employee scheduling, attendance, and payroll.
5. **Security Manager:** This control object ensures the safety of the restaurant, staff, and customers.

# Class Diagram

## Table
*Table*

id

name (customer)

occupied

if requested by user to book,
by approval from
management occupied
will be turned true from false

## Orders

id

user_id

total

isPaid

note

table_id

Users order from table -
accepted/rejected by staff

on this table

## Staff

id

Managing inventory - accounts -
payments - bills - orders - tables

approve /
reject

Order Request

accepted

Order confirm

denied

Re-order

Orders food if user

in this order

mapping to the order

Denied => retry for table booking

Approval => occupied = true

## User

id

name

email

passwordi

is_admin

table_id

if admin, will have admin access
otherwise normal user who can order
add / remove

add / remove by admin

## Product

id

name

price

category_id

stock

Added to user's order or
can be viewed in a menu

product ordered

## Order_of_products

id

order_id

product_id

quantity

Amount and more details
about user orders

## Category

id

name

Holds product sorted as
per having same  category

Re-order / order cancelled

# Sequence Diagram

## 1. User Ordering the Food :
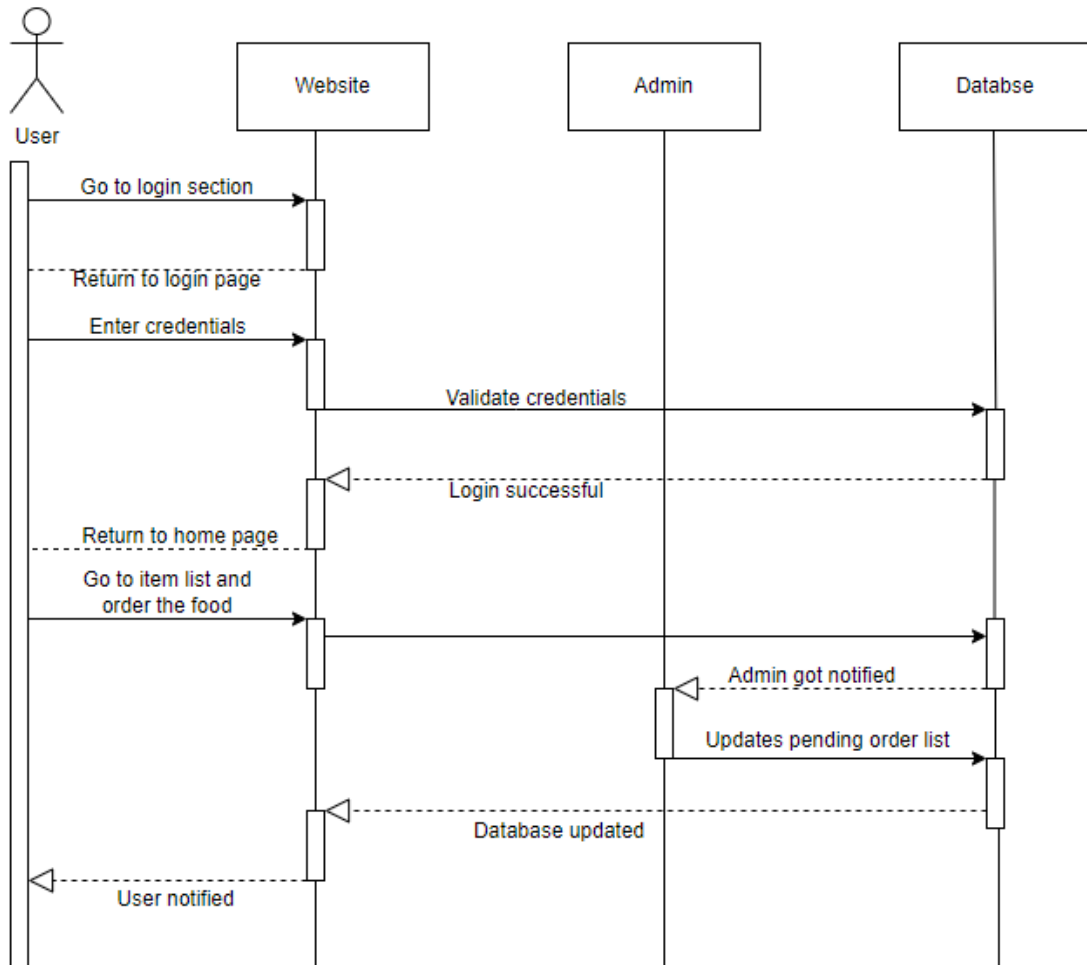
## 2.Admin Managing the System :



Admin

:Website

:Database

Visit Login Page

Enter admin credentials

Validate credentials

Admin Login Sucessfull

Return to Home Page

Goes to inventory page

Returns inventory page

admin selects the item whose
stock is to be checked

Requests for item stock

Returns the stock of requested item

Displays the stock of selected item
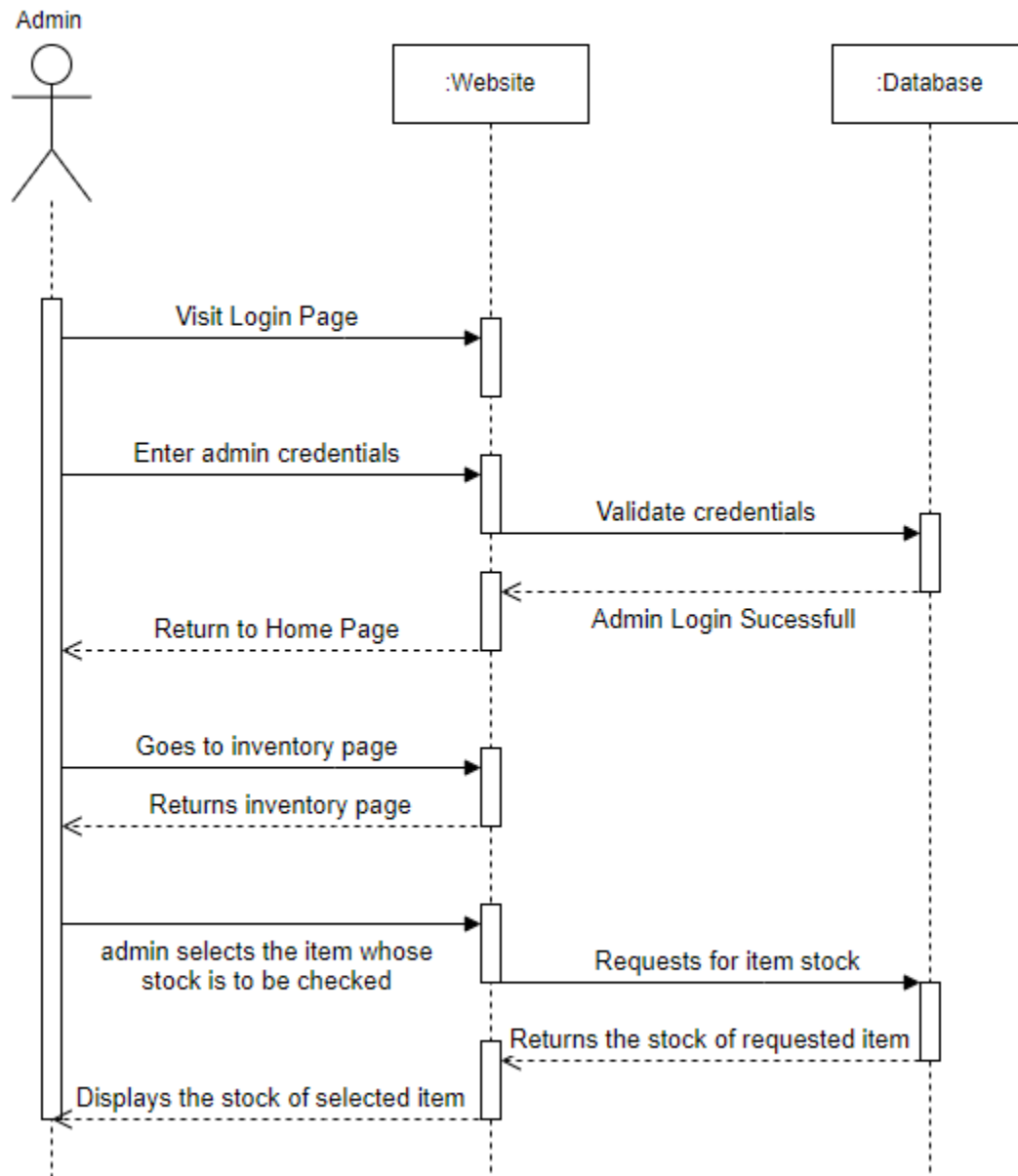
# Design Goals

**Maintenance and Dependability Criteria**
1. **Efficiency**: The system ought to be developed to increase restaurant productivity. This may require speeding up the ordering procedure, shortening wait times, and streamlining the payment procedure.
2. **Accuracy**: The system should be designed to minimize errors. This might require making sure that orders are precise and that payments are handled properly.
3. **Scalability**: The system must be built to support expansion and growth. As the restaurant expands, this can include introducing new capabilities and features..
4. **Security**: The system should be created to guarantee the security of user data, including payment and personal information.
5. **Cost-effective**: The system should be designed to provide a return on investment for the restaurant. This could include lowering labor costs or increasing revenue through increased efficiency.
6. **Robustness:** The system must be able to handle high volumes of transactions and requests without crashing or experiencing downtime. This requires designing the system to be robust and capable of handling unexpected errors or failures.
7. **Integration**: The system should be designed to integrate with existing restaurant technology, such as point-of-sale systems and inventory management systems.
8. **Customization**: The system ought to be created so that restaurants can alter it to suit their unique requirements. Customizing menu items, prices, and order processing may fall under this category.

**Performance Criteria**

1. **ResponseTime:** The system must respond as quickly as possible to user queries.
2. **Throughput:** The system must complete a predetermined number of jobs in a given period of time.

**End-User Criteria**

1. **Usability:** The system should have a simple user interface that can be used by anyone without requiring any special training or technical knowledge
2. **User-friendly**: Both customers and restaurant personnel should find the system simple to use. This can entail offering simple interfaces and clear directions.

# High Level System Design

```
                        ┌─────────────────────────────┐
                        │  Restaurant automation System │
                        └─────────────────────────────┘

┌──────────────────┐                          ┌──────────────────┐
│ Presentation layer │                          │ Business Login layer │
└──────────────────┘                          └──────────────────┘

        ┌──────────────────┐                          ┌──────────────────┐
   ───> │  User Interface   │                     ───> │  Order approval   │
        └──────────────────┘                          └──────────────────┘

        ┌──────────────────┐                          ┌──────────────────┐
   ───> │ User Authentication │                    ───> │ Managing Accounts │
        │  (login / Sign up)  │                        └──────────────────┘
        └──────────────────┘

        ┌──────────────────┐                          ┌──────────────────┐
   ───> │    Order Food     │                     ───> │ inventory display │
        └──────────────────┘                          └──────────────────┘

        ┌──────────────────┐                          ┌──────────────────┐
   ───> │     Payment       │                     ───> │  Table booking    │
        └──────────────────┘                          │    approval       │
                                                       └──────────────────┘
        ┌──────────────────┐   ┌──────────────────┐
   ───> │  Table booking    │   │ Data Access Layer │
        └──────────────────┘   └──────────────────┘

                                   ┌──────────────────┐
                              ───> │ Database connection │
                                   └──────────────────┘

                                   ┌──────────────────┐
                              ───> │   User database   │
                                   └──────────────────┘

                                   ┌──────────────────┐
                              ───> │ Product database  │
                                   └──────────────────┘

                                   ┌──────────────────┐
                              ───> │ Inventory database │
                                   └──────────────────┘

                                   ┌──────────────────┐
                              ───> │  Order database   │
                                   └──────────────────┘
```
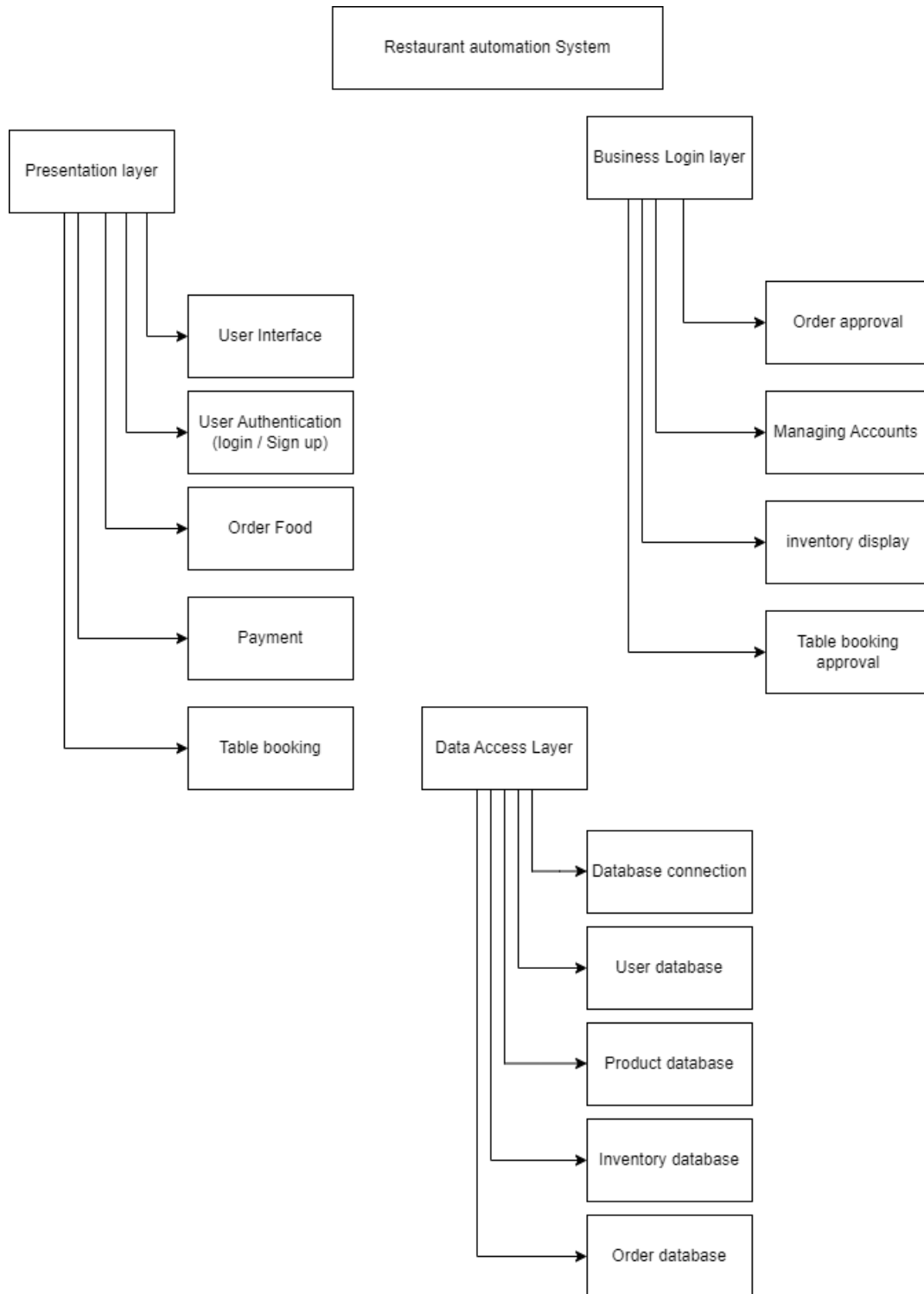
# The layers of the three-tier architecture are:

1. **Presentation Layer :**
   - This layer will handle the user interface and user authentication (login / sign up ) for the system.
   - It also includes the user interface components that will allow users to interact with the system and authentication subsystem that will verify the credentials.
   - It also handles ordering food from the customer , payment mechanism and booking of the table.

2. **Business Logic Layer:**
   - This layer contains the core business logic of the system.
   - It consists of approval of food orders which will approve food orders if they are available , managing staff accounts , inventory display and approval of table booking which will book tables if it is not occupied.

3. **Data Access Layer:**
   - This layer will handle the storage and retrieval of data for the system.
   - It includes Database connection that will establish connection to the database, user database which will store the information about users , product database which will store the information about the items available in the restaurant , inventory database which will store inventory records of the restaurant and order database in which data of all the orders from customer will be stored.

Overall , this architecture provides a proper separation of concerns and will allow easy maintenance and scalability of the system.