



## DU Food Point Management System

Software Requirements Specification and Analysis [SE-2206L]

### Submitted By

#### Group 04

Morshaline Mojumder (BSSE - 1504)

Dhanapoti Roy (BSSE - 1510)

G. M. Rashidul Islam Rahat (BSSE - 1517)

Emam Hassan (BSSE - 1523)

Fahim Shahriar Sizan (BSSE - 1531)

### Submitted To

Dr. Kazi Muheymin-Us-Sakib

Professor,

Institute of Information Technology

University of Dhaka

Submitted On : 24-09-202

# Software Requirement & Specification Analysis of DU Food Point

## List of stakeholders of DU Food Point

- Owner
- Manager
- Customer
- Rider

## Viewpoints of stakeholders

We have met with the stakeholders who are playing a vital role in the entire process to get a better understanding of what they require. We collected their viewpoints regarding this system.

### • Owner

The primary focus should be customer satisfaction and operational efficiency to maximize profit. The system must provide complete administrative control over staff, menu, and finances while being user-friendly to stay ahead of competitors. Requires automation to minimize operational overhead and enable data-driven business decisions.

### • Manager

The system must streamline order fulfillment through centralized delivery coordination. Should function as a command center for assigning deliveries, monitoring performance, and handling exceptions. Must break down traditional coordination barriers with real-time visibility and intelligent assignment algorithms.

### • Customer

Customers face issues with traditional ordering methods like phone calls, limited menu visibility, and lack of tracking. They require an automated system that's convenient and time-saving with instant confirmations, secure payments, real-time tracking, and personalized recommendations available 24/7.

### • Rider

Riders struggle with unclear instructions, inefficient routes, and communication gaps. They need a mobile-optimized system with clear assignments, GPS navigation, customer communication channels, fair

workload distribution, and simple status updates to maximize delivery efficiency.

## **QFD (Quality Function Deployment)**

The list below is the Quality Function Deployment, or QFD, of a DU FOOD POINT Restaurant Management System, aiming to align all the requirements needed for its development. We gathered all these requirements after consulting with all the stakeholders of this system, and organized them into normal, expected and exciting categories.

### **Normal Requirements**

- A predefined account will be given to the restaurant owner. The owner account will have access to the central database containing all user data and complete administrative control.
- The login module will have special accounts for employees (Manager, Delivery Rider) which will have separate interfaces catered to their roles such as order management, delivery assignments, and status updates.
- Customers will provide their personal information including name, address, contact details, and preferences to create their user account.
- The system will display all currently available food items organized by categories with images, descriptions, prices, and availability status.
- Customers can select food items and add them to cart, specify quantities, and apply any available vouchers or discounts before checkout.
- Customers can customize their orders by adding special notes and preferences such as "extra spicy", "leg piece only", or specific delivery instructions.
- Customers can pay via Mobile Financial Services (bKash, Nagad, Rocket) or choose Cash on Delivery. Online payment will require OTP verification and PIN confirmation.
- The admin can manage menu items including adding new items, editing existing ones, updating prices, and managing availability status.
- The manager can assign delivery orders to available riders based on location proximity and workload distribution.
- There will be a communication module where customers can directly contact the owner or employees for order inquiries, complaints, or feedback.
- The system will integrate with MFS APIs for secure payment processing and GPS services for location tracking and route optimization.
- The system must provide order tracking, delivery status updates, and basic analytics for business insights.

## Expected Requirements

- Users should be able to navigate through the ordering process effortlessly, without needing extensive guidance or complex procedures.
- The interface should be responsive across various devices and screen sizes. Whether users access the system on desktop, tablet, or smartphone, it should adapt seamlessly.
- The system should provide secure password storage with OTP-based password reset functionality.
- The system should provide proper data security with encrypted payment transactions and secure user data handling.
- There will be a search functionality where users can find food items by name, category, or price range quickly and efficiently.
- Customers will have access to their order history, allowing them to reorder previous meals with a single click for convenience.

## Exciting Requirements

- The system will feature meme-based food suggestions where admins can upload funny food memes that rotate on the homepage, creating viral marketing moments and making food ordering entertaining and shareable.
- Personalized birthday celebration system that automatically sends special discount notifications to customers on their birthday, along with complimentary dessert offers and personalized birthday-themed food recommendations.
- Smart weather-responsive menu that automatically promotes hot items during cold weather and refreshing drinks during hot days, with weather-based promotional banners.

## User Story

### 1. Registration

A new customer needs to register in the system by providing required personal information such as name, contact details, and credentials. This allows the customer to create an account and access restaurant services. During registration, the customer must provide details like first name, last name, father's name, mother's name, date of birth, present and permanent address, blood group, religion, email, and at least one phone number (a second phone number is optional). The

customer must also set a password for future login. Once the registration is complete, the account will be created and ready for use.

## 2. Login

A registered user (Customer, Employee, or Restaurant Owner) should be able to log in to the system using their registered phone number or email along with a password. Upon successful authentication, the user gains access to features and services based on their role in the system. Additionally, the system provides an option to securely store login credentials, allowing for faster access through one-tap login. If the user forgets their password, they can reset it through OTP verification sent to their registered phone number or email.

## 3. Menu System

The Menu System enables administrators to efficiently manage all food items while providing customers with a dynamic and personalized browsing experience. In the backend, authorized admins can add new menu items, edit existing items, update prices, change availability, and remove items as necessary. When adding a new item, the system requires: Item Name, Category (e.g., Biryani, Drinks, Snacks), Price, Detailed Description, Item Image, Stock Availability Status (Available/Unavailable), and an optional Featured Tag. Featured items are highlighted on the homepage under Smart Suggestions, and all new items automatically appear in the customer interface with a prompt such as “You can try our new item ‘Chap-Polao’.”

Admins can edit item details at any time, including updating descriptions, changing images, adjusting prices, or marking items as temporarily unavailable. Items can be soft deleted (hidden from customers but retained in the database) or permanently deleted, with soft-deleted items restorable if needed. Additional admin features include: category reorganization, promotional scheduling with discount badges, and homepage media uploads such as banners or memes. All changes trigger automatic cache refresh or API sync to ensure immediate updates on the customer interface. All menu modifications are logged in an audit trail with admin ID, timestamp, and action details for accountability.

On the customer-facing side, the menu is displayed in card-style format, showing item image, detailed description, price, and real-time availability. Customers can search for items by name, price range, or category. A smart suggestion engine recommends items based on previous orders, popularity trends, and newly added

items. Available discounts, vouchers, and promotions are clearly displayed, and customers can store items in a virtual cart or review past orders for one-click reordering.

## 4. Order System

The Order System allows customers to select, customize, and finalize food orders efficiently, integrating order management, cart functionality, and real-time updates. Customers select food items from the digital menu, choosing multiple types and quantities of items in a single order. They can apply vouchers or promotional codes and add custom notes or preferences, such as “extra spicy” or “leg piece only”. Selected items are added to a virtual cart, which maintains a running total and allows modification before checkout.

The system tracks order progression in real-time, updating status through predefined stages: Pending → Preparing → Ready → On the Way → Delivered. Once an order is confirmed, the manager assigns a delivery rider, and the system provides live tracking of the rider’s location via GPS. Customers receive automatic notifications at key milestones, including order preparation and rider arrival updates.

The system also supports direct communication between customers and delivery riders, allowing real-time queries or instructions regarding the order. After delivery, customers can submit feedback and rate their experience, ensuring quality control and improving service reliability. This Order System ensures a streamlined, transparent, and interactive experience from selection to delivery, keeping customers informed and engaged at every step.

## 5. Payment System

Customers can choose between online payments through Mobile Financial Services (MFS) such as bKash, Nagad, and Rocket, or opt for Cash on Delivery (COD).

For online payments, customers select their preferred MFS provider and enter their registered mobile number associated with that account. The system integrates securely with the MFS provider’s API to generate a One-Time Password (OTP), which is sent to the customer’s mobile. The system validates the OTP within a limited number of attempts and a time-bound window. Once the OTP is verified, customers enter their MFS PIN, which is encrypted and transmitted securely to the payment gateway. Successful authentication results in a transaction confirmation, marking the payment as complete. A digital receipt is automatically generated, including transaction ID, timestamp, and payment amount, and is sent via SMS.

and/or email to the customer.

If Cash on Delivery is selected, no online transaction is required. The order is placed immediately, with payment status marked as “Pending - COD”. Delivery agents collect cash upon delivery and update the system accordingly.

The system allows customers to apply voucher codes or promotional discounts during checkout. Voucher validation checks criteria such as expiry date, minimum order value, applicable items or categories, and maximum discount limits, ensuring the final payable amount is updated in real-time before confirmation.

In the event of payment failures—such as invalid OTP, insufficient balance, or transaction timeouts—the system provides detailed error messages and options to retry or switch payment methods. All payment activities, including online and COD transactions, are logged in an audit trail to ensure transparency, accountability, and troubleshooting. This payment system ensures a secure, seamless, and user-friendly transaction experience, giving customers confidence while maintaining data integrity and supporting multiple platforms.

## 6. Delivery Management System

Once an order is confirmed, the admin receives delivery assignment requests on a centralized dashboard displaying pending deliveries with complete order details and customer locations. The system shows real-time maps of available riders and provides intelligent assignment suggestions based on proximity, workload, and availability. Assigned riders receive full order details, including customer contact, delivery address, order contents, special instructions, and optimized route guidance.

The Delivery Management System ensures efficient coordination between customers, owner, and delivery riders while integrating GPS tracking and real-time notifications for a seamless delivery experience. Customers specify their preferred delivery location using manual entry, interactive map selection, or automatic GPS detection. The system maintains a history of addresses for quick selection and allows detailed landmarks and special instructions, such as “main door delivery.”

Before confirming a delivery, the GPS system verifies the coordinates against predefined service areas. For valid locations, the system provides estimated delivery times; for out-of-area requests, customers receive a notification like “We are not available in your area” with suggested alternatives.

Customers can track their assigned rider in real-time via GPS, viewing live location, estimated arrival time, and planned route. Location updates occur every 30 seconds, and the system sends automatic notifications at key milestones, such as

“Your rider has picked up your order” and “Your rider is 10 minutes away.” Delivery status progresses through predefined stages: Pending → Assigned → Picked Up → In Transit → Delivered. Riders update status at each stage, while admins manually update exceptional cases. All status changes are logged with timestamps for audit purposes, and unsuccessful deliveries require riders to provide specific reasons for complete transparency.

The system also includes a communication module, allowing logged-in users (customers, employees, and owners) to exchange messages. Customers can select a recipient (owner or employee) and categorize messages as order inquiry, issue/complaint, or general feedback. The owner can respond, mute users, delete inappropriate messages, and filter messages by type to efficiently manage communication.

After delivery, customers can submit feedback and rate their experience, ensuring continuous service quality monitoring. This system ensures transparent, real-time delivery management, integrating rider assignment, tracking, communication, and feedback for a seamless user experience.

## Use Case Diagram

A Use Case Diagram is a visual representation of the functional requirements of a system. It shows how different users (actors) interact with the system to achieve specific goals (use cases).

### **Primary Actor:**

Primary actors interact to achieve required system function and derive the intended benefit from the system. They work directly and frequently with the software.

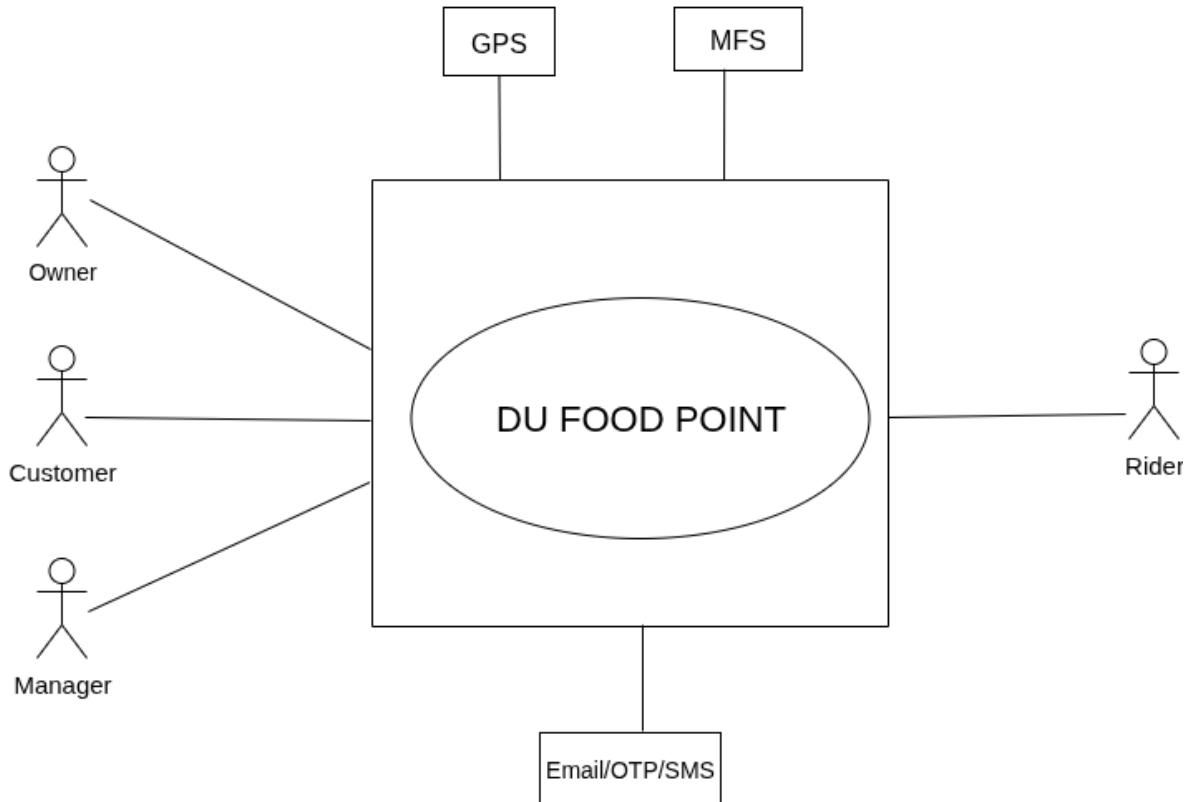
### **Secondary Actor:**

Secondary actors support the system so that primary actors can do their work. They either produce or consume information.

### Level 0 :DU Food Point Management System

**Primary actors:** Owner,Manager,Customer

**Secondary actors:** Rider,MFS,GPS,Email/OTP/SMS

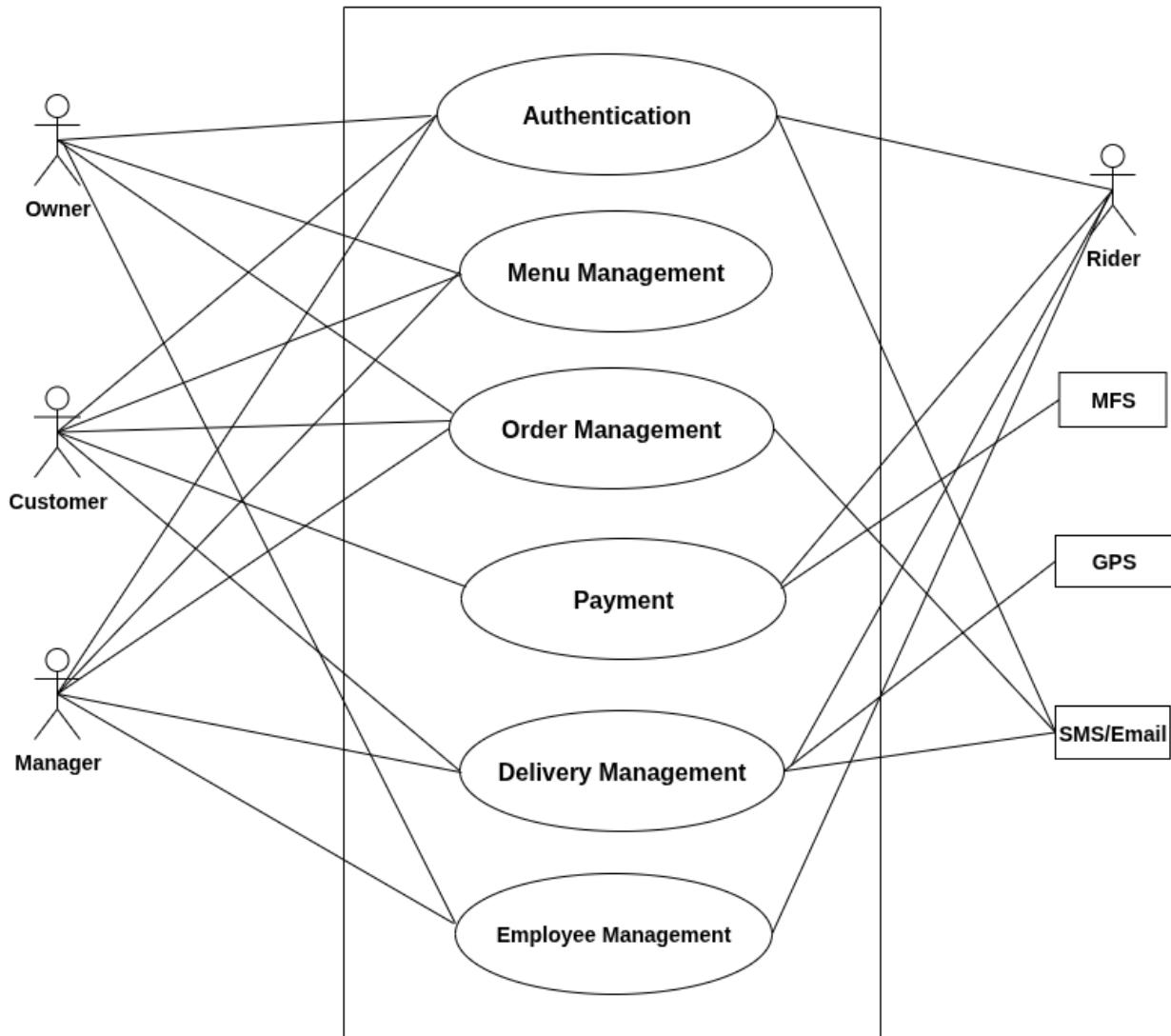


The DU Food Point Management System is a centralized platform that connects multiple stakeholders within a university-based food service environment. The system is represented in the Level 0 DFD with both primary and secondary actors. The primary actors include the Owner, Manager, and Customer, who are directly linked with the system as key participants. The secondary actors consist of the Rider, Mobile Financial Services (MFS), GPS, and Email/OTP/SMS, which provide external support for delivery, payment, location tracking, and communication. At the core, the DU Food Point System functions as the central unit that integrates all actors, illustrating the overall structure of interactions without showing internal processes. This diagram highlights the scope of the system and defines its boundary by showing how external entities remain connected with the central system.

### Level 1 :DU Food Point

**Primary actors:** Owner,Manager,Customer

**Secondary actors:** Rider,MFS,GPS,SMS/Email



### Authentication

This subsystem ensures that only authorized users can access the DU Food Point system. It verifies the login credentials of users such as owners, managers, employees, riders, and customers. The authentication process helps keep the system secure by preventing unauthorized access and protecting sensitive data like payment details, customer information, and employee records.

## **Menu Management**

This subsystem is responsible for managing the entire food menu of the DU Food Point. It allows adding new food items, editing their details, removing outdated or unavailable items, and updating stock availability. This keeps the menu accurate and up to date so that customers can always view the latest offerings. It plays a critical role in customer satisfaction and smooth restaurant operations.

## **Order Management**

The order management subsystem handles the full order process, starting from when a customer places an order until it is completed. It stores order details, tracks order progress, and ensures smooth communication between the kitchen, manager, rider, and customer. This helps to reduce confusion, maintain timely updates, and improve the efficiency of order handling.

## **Payment**

This subsystem manages all payment-related activities. It ensures that when customers pay for their orders, the transactions are securely processed through mobile financial services or other payment methods. It verifies each transaction and updates the order status to “paid” once the payment is confirmed. This ensures a smooth and reliable payment experience for customers and management.

## **Delivery Management**

The delivery management subsystem is responsible for assigning orders to delivery riders, tracking delivery progress, and updating customers on their order status. It uses GPS tracking to locate the rider’s position in real-time and ensures that customers are notified about their delivery timing through SMS or notifications. This helps to improve delivery efficiency and customer satisfaction.

## **Employee Management**

This subsystem handles everything related to employee records, including account creation, role management, and access permissions. It allows managers and owners to add new employees such as riders, chefs, or cashiers. It also ensures that employee details are verified and stored securely. OTP-based verification is used to confirm account authenticity before granting access to the system.

### **Action and Reply:**

**Action:** A user tries to log in with credentials.

**Reply:** The system checks the information and grants or denies access.

**Action:** A manager adds or edits food items in the menu.

**Reply:** The system updates the menu and displays the changes to customers immediately.

**Action:** A customer places a new order.

**Reply:** The system records the order and sends confirmation notifications.

**Action:** Payment is made for an order.

**Reply:** The system verifies the payment and updates the order status to “paid.”

**Action:** A delivery task is assigned to a rider.

**Reply:** The system sends order details to the rider and starts GPS tracking.

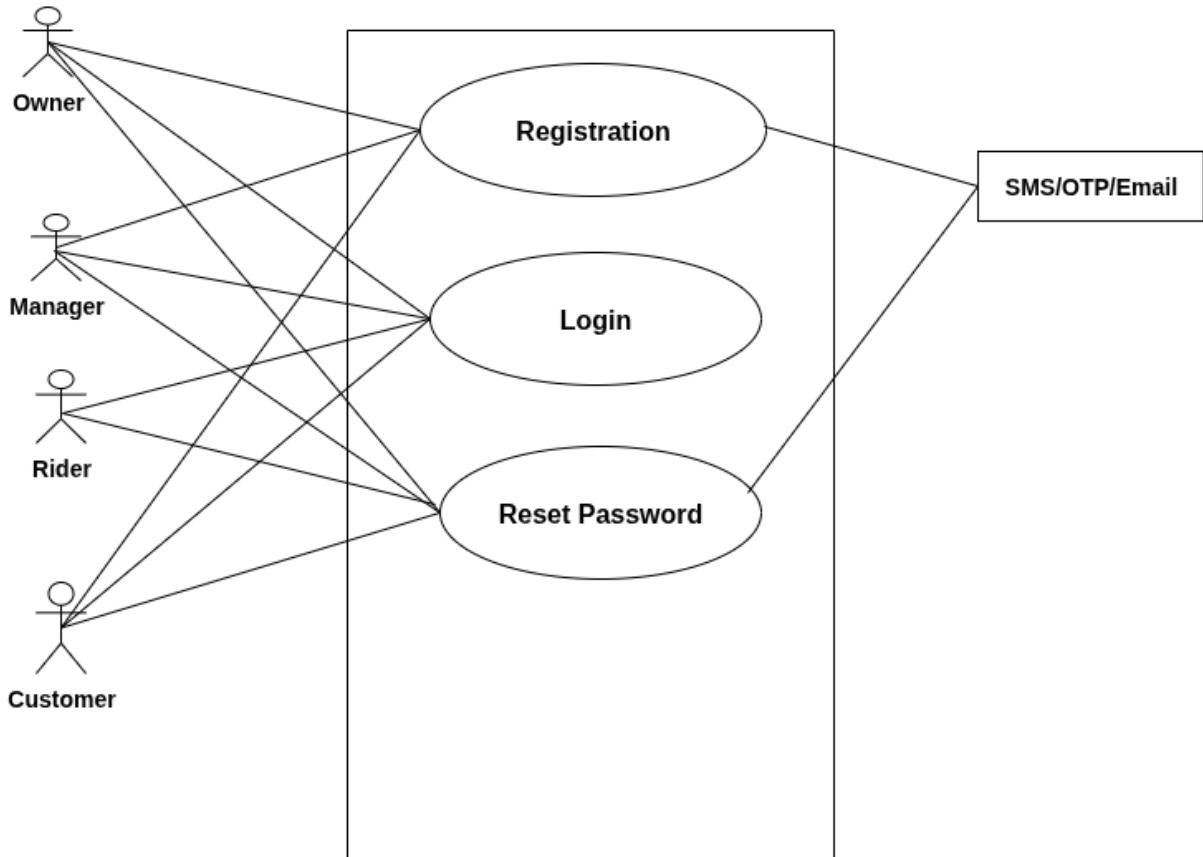
**Action:** A new employee account is created.

**Reply:** The system verifies the details and activates the employee account.

### Level 1.1 :Authentication

**Primary actors:** Owner,Manager,Rider,customer

**Secondary actors:** SMS/OTP/Email



## Registration

The registration process is the entry point for all users including the Owner, Manager, Rider, and Customer. It ensures that only verified individuals can access the system. During registration, the user provides personal information such as full name, email address, phone number, and a chosen password. The system validates the inputs, checking for proper format and uniqueness (for example, ensuring no duplicate accounts exist). To secure the process, the system sends an OTP (One-Time Password) or a verification link via SMS or Email. The user must complete

this verification step before the account is activated. This step prevents fake accounts, strengthens security, and ensures that communication channels are valid for future notifications. Once successful, the system creates a secure user profile in the database, encrypted to protect sensitive information.

## **Login**

Login is the main gateway into the system. After registration, users must provide their correct email/phone number and password to gain access. The system cross-checks the entered credentials with its database records. If the information is valid, the system grants access to the respective dashboard depending on the user's role (Owner, Manager, Rider, or Customer). Incorrect details trigger error messages such as "Invalid Credentials" or "Account Not Found." The system may also include security measures like account lockouts after multiple failed attempts and session management to prevent unauthorized access. This process ensures that only legitimate users can perform operations within the system.

## **Reset Password**

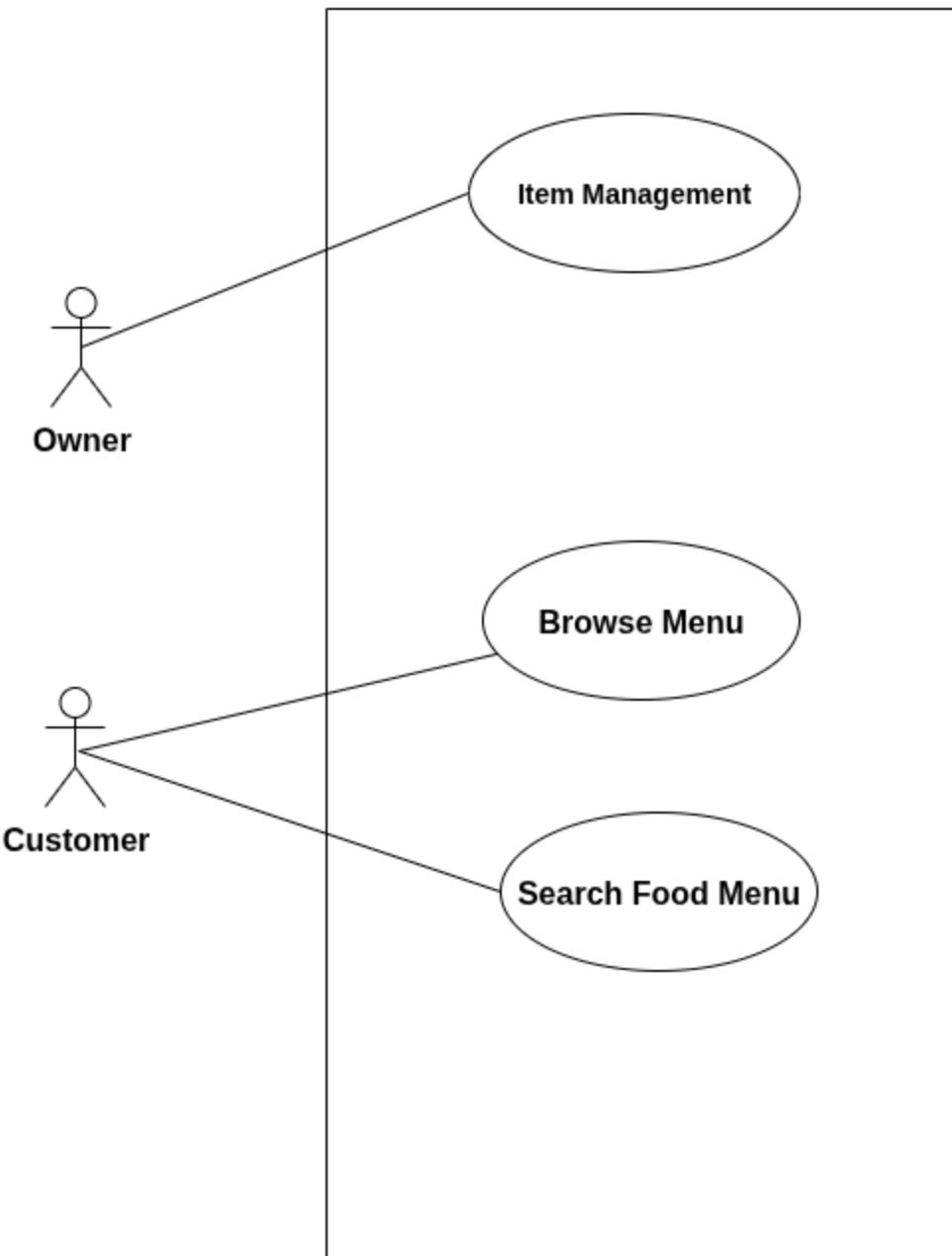
The reset password feature helps users recover access if they forget their login credentials. To initiate a reset, the user provides their registered phone number or email address. The system then verifies identity by sending an OTP or reset link. After successful verification, the user is prompted to create a new password that meets security requirements (such as minimum length, use of special characters, etc.). The system updates the password in the database after encrypting it. This functionality ensures that account recovery is smooth while maintaining a high level of security to prevent unauthorized changes.

## **Actions and Replies**

- **Action 1:** User initiates registration.  
**Reply 1:** System sends OTP/Email verification and creates account.
- **Action 2:** User enters login credentials.  
**Reply 2:** System verifies and grants access.
- **Action 3:** User requests password reset.  
**Reply 3:** System sends OTP and allows password update.

## Level 1.2 :Menu management

Primary actor: Owner,Customer



## Item Management

The Owner has complete control over the restaurant's menu. Through this feature, they can add new dishes, update existing details such as name, price, description, and availability, or remove items that are no longer offered. The system ensures that changes are reflected immediately in the database so customers always see the latest and most accurate information. For example, if a dish is out of stock, the Owner can mark it unavailable, preventing customers from placing incorrect orders. This management process ensures smooth operations and accurate representation of offerings.

## Browse Menu

Customers rely on this feature to explore the available food items. The system provides an organized, easy-to-navigate interface showing item names, detailed descriptions, prices, and whether the item is available or sold out. High-quality images or extra details (such as preparation time or nutritional information) can also be displayed to enhance the browsing experience. This transparency helps customers make informed choices and reduces order mistakes.

## Search Food Menu

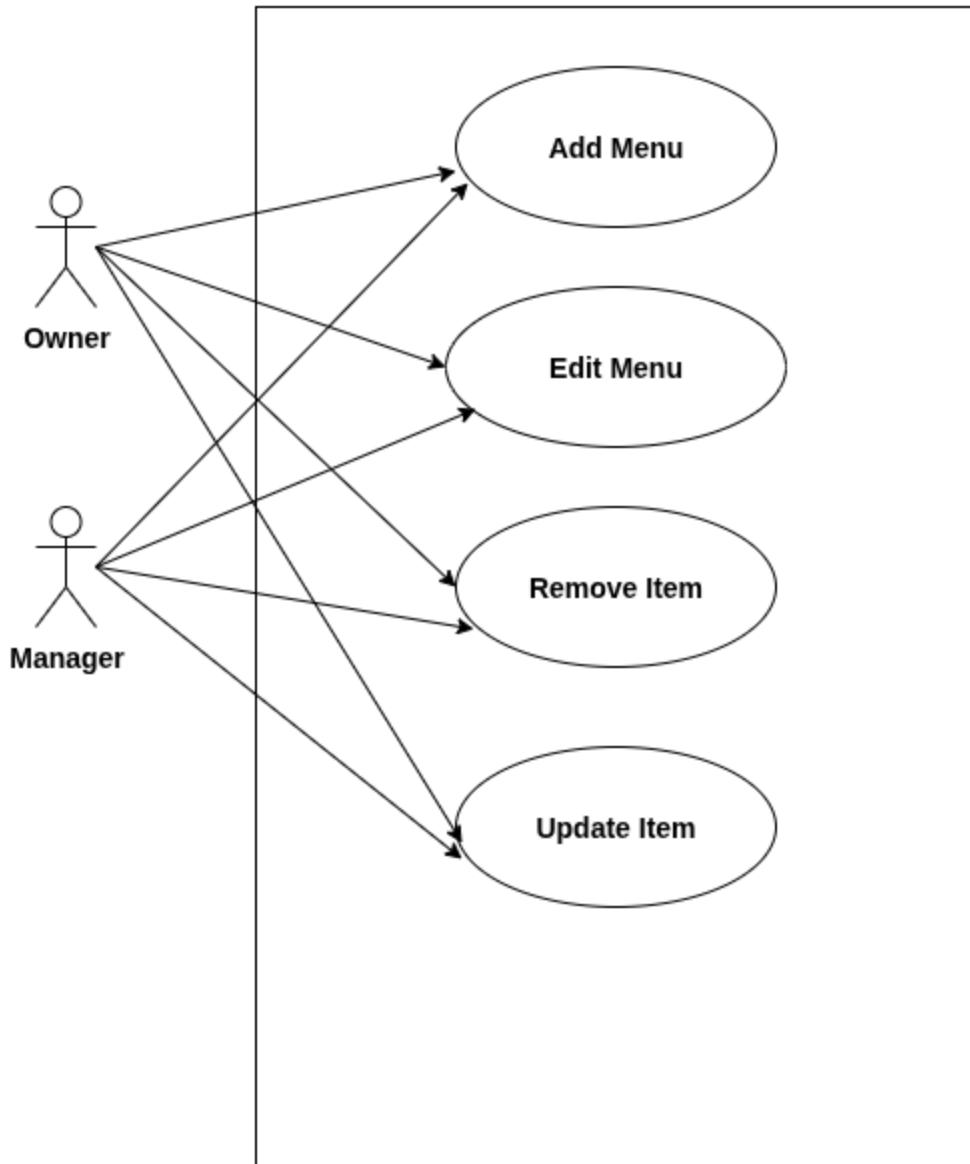
The search option enhances customer convenience by allowing them to find specific items quickly. Customers can search using keywords (like “burger,” “pasta,” or “chicken”) or apply filters such as category, price range, or popularity. The system processes the query and returns matching items instantly. This speeds up the ordering process and makes the customer experience smoother, especially in large menus with many choices.

## Actions and Replies

- **Action 1:** Owner adds or updates menu items.  
**Reply 1:** System updates the menu database.
- **Action 2:** Customer browses the menu.  
**Reply 2:** System displays all items.
- **Action 3:** Customer searches for a food item.  
**Reply 3:** System shows relevant results.

### Level 1.2.1 : Item Management

Primary Actors : Owner, Manager



#### Add Menu

This function allows managers or owners to add new food items to the restaurant's

menu. When adding an item, details like the name, price, description, and category are provided to ensure customers have complete information when browsing.

## **Edit Menu**

Through this function, existing menu details can be modified. Managers or owners can update information such as price changes, updated descriptions, or shifting items to different categories to keep the menu current.

## **Remove Item**

When a food item is no longer available or discontinued, this function allows it to be permanently removed from the menu. This prevents customers from ordering unavailable dishes and keeps the menu clean.

## **Update Item**

This function helps update stock levels and availability status. It ensures that customers always see which items are in stock and prevents them from placing orders for unavailable products.

### **Action and Reply:**

**Action:** A new dish is added to the menu.

**Reply:** The system saves the dish information and shows it in the menu for customers.

**Action:** Menu details are edited, such as price or description.

**Reply:** The system updates the record with the new details.

**Action:** An unavailable item is removed.

**Reply:** The system deletes the item and removes it from customer view.

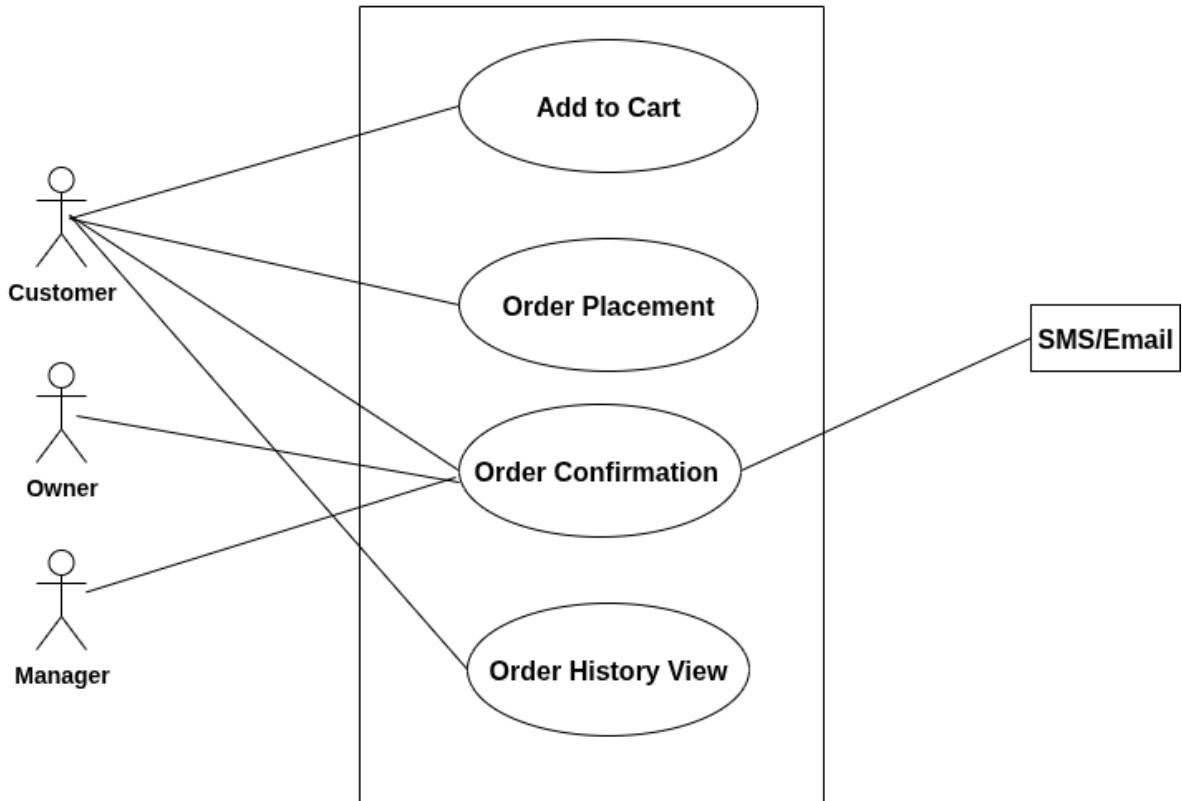
**Action:** Stock levels or availability are updated.

**Reply:** The menu instantly reflects the updated availability status.

### Level 1.3 :Order Management

**Primary actor: Customer,Owner,Manager**

**Secondary actors: SMS/ Email**



### Add to Cart

The Add to Cart feature is where the customer's journey towards placing an order begins. Customers select food items they want, and the system creates a virtual shopping cart. As items are added or removed, the cart dynamically updates the quantity and calculates the total cost in real time. This includes the base item price, discounts, service charges, and applicable taxes. Customers can adjust item quantities at any time, giving them full control over the order before checkout.

### Order Placement

Once satisfied with their cart, customers proceed to order placement. The system verifies that all chosen items are still available. If there are shortages, the system notifies the customer immediately. Otherwise, the system calculates the final bill

(including discounts or vouchers) and securely stores the order in the database. The order is assigned the status *Pending*. This ensures both customer and restaurant staff are aligned before preparation begins.

## Order Confirmation

The system automatically generates an order confirmation message, which is sent to the customer via SMS or Email. This message includes order details, expected preparation time, and the current status. It assures customers that their request has been successfully recorded and is being processed.

## Order Status Check

Customers can continuously track their order through the Order Status feature. The system updates statuses in real time: *Pending*, *Preparing*, *Ready for Pickup*, *Out for Delivery*, *Delivered*, or *Cancelled*. This transparency keeps the customer informed at every stage of the process, reducing uncertainty and increasing trust.

## Order History View

For convenience, customers can view all their previous orders through this feature. The system stores data such as order date, food items, cost, payment method, and delivery outcome. Customers can use this history for quick reordering or for record-keeping. This also helps the restaurant analyze customer behavior.

## Notifications

Notifications play an essential role in keeping customers updated without them having to constantly check the app. Alerts are sent for critical events such as order confirmation, preparation start, when the order is out for delivery, or when it is successfully delivered. If the order is canceled, the system notifies the customer immediately.

## Actions and Replies

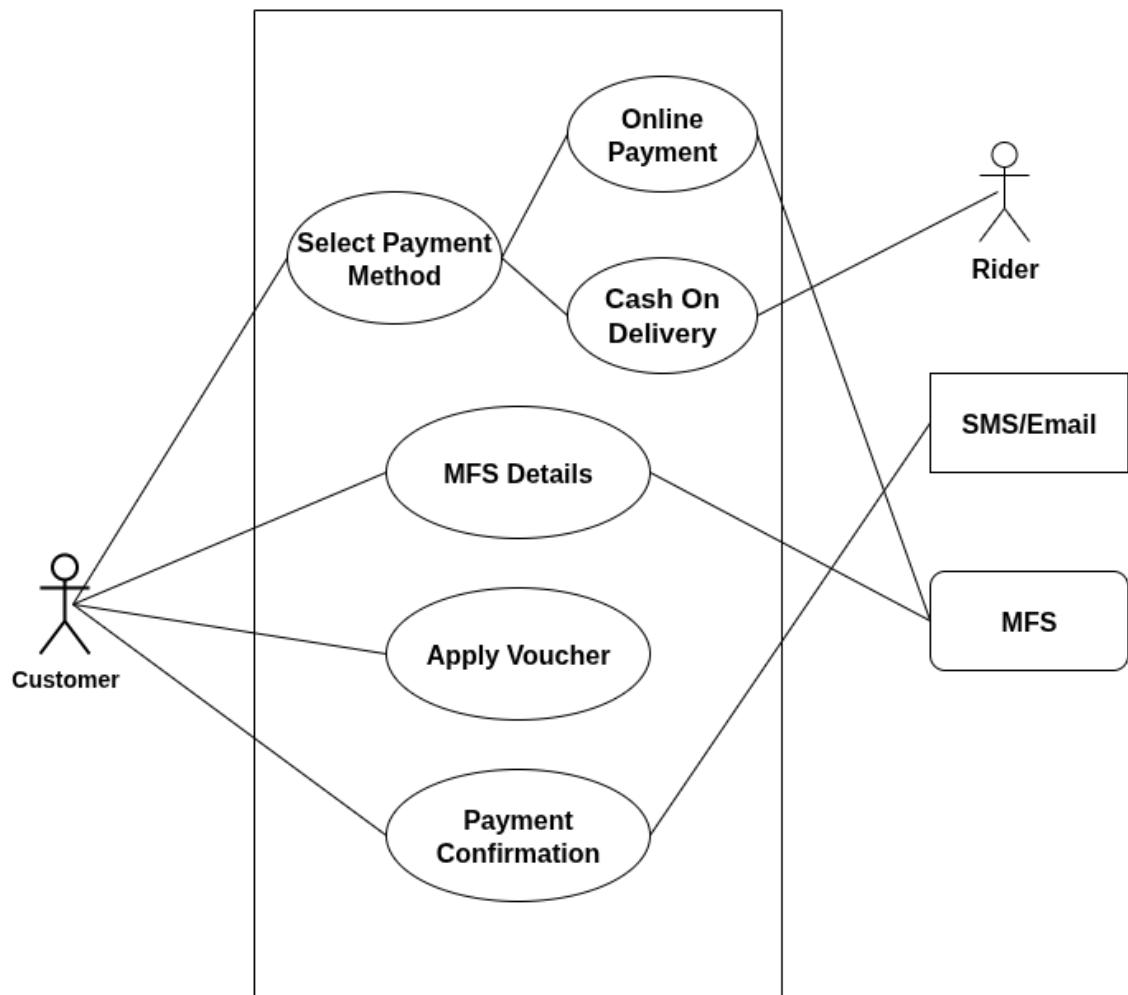
- **Action 1:** Customer adds items to cart.  
**Reply 1:** System updates total price.
- **Action 2:** Customer places order.  
**Reply 2:** System validates and confirms.
- **Action 3:** Customer checks order status.  
**Reply 3:** System displays real-time status.

- **Action 4:** Customer views past orders.  
**Reply 4:** System retrieves history.
- **Action 5:** System sends notifications.  
**Reply 5:** Customer receives updates.

#### Level 1.4:Payment

**Primary actor: Customer**

**Secondary actors: Rider,SMS/Email,MFS**



## Select Payment Method

Customers are provided with flexibility by allowing them to choose a preferred payment option. Options include Online Payment (through Mobile Financial Services such as bKash or Nagad) and Cash on Delivery. This choice ensures that customers with varying needs and preferences can comfortably complete their orders.

### Online Payment

If a customer selects online payment, the system redirects them to a secure MFS gateway. Here, they log in with their registered mobile wallet or linked bank account and authorize the payment. Once the transaction is processed, the gateway sends a confirmation (success or failure) back to the system. Successful payments update the order status to *Payment Received*, while failed attempts trigger error messages and request retries. This integration ensures fast and secure digital transactions.

### Cash on Delivery

If the customer chooses Cash on Delivery, the system immediately records the payment status as *Payment Pending*. When the Rider delivers the order and collects cash, the system updates the status to *Payment Received*. This option is convenient for customers who prefer traditional payment methods.

### MFS Details

Behind the scenes, the system manages and integrates with Mobile Financial Services providers. It stores secure API keys, validates credentials, and configures services to ensure smooth digital transactions. Proper handling of these details ensures reliability and prevents errors in financial processes.

### Apply Voucher

To make the ordering experience more attractive, the system allows customers to enter voucher or promo codes during checkout. The system validates the voucher, ensuring it hasn't expired and meets usage conditions. If valid, the discount is applied instantly, and the final payable amount is updated in real time.

## Payment Confirmation

After payment is complete, the system ensures both the Customer and the Rider are notified immediately. For online payments, confirmation is sent right after the gateway approves the transaction. For Cash on Delivery, confirmation is sent once

the Rider successfully collects the payment.

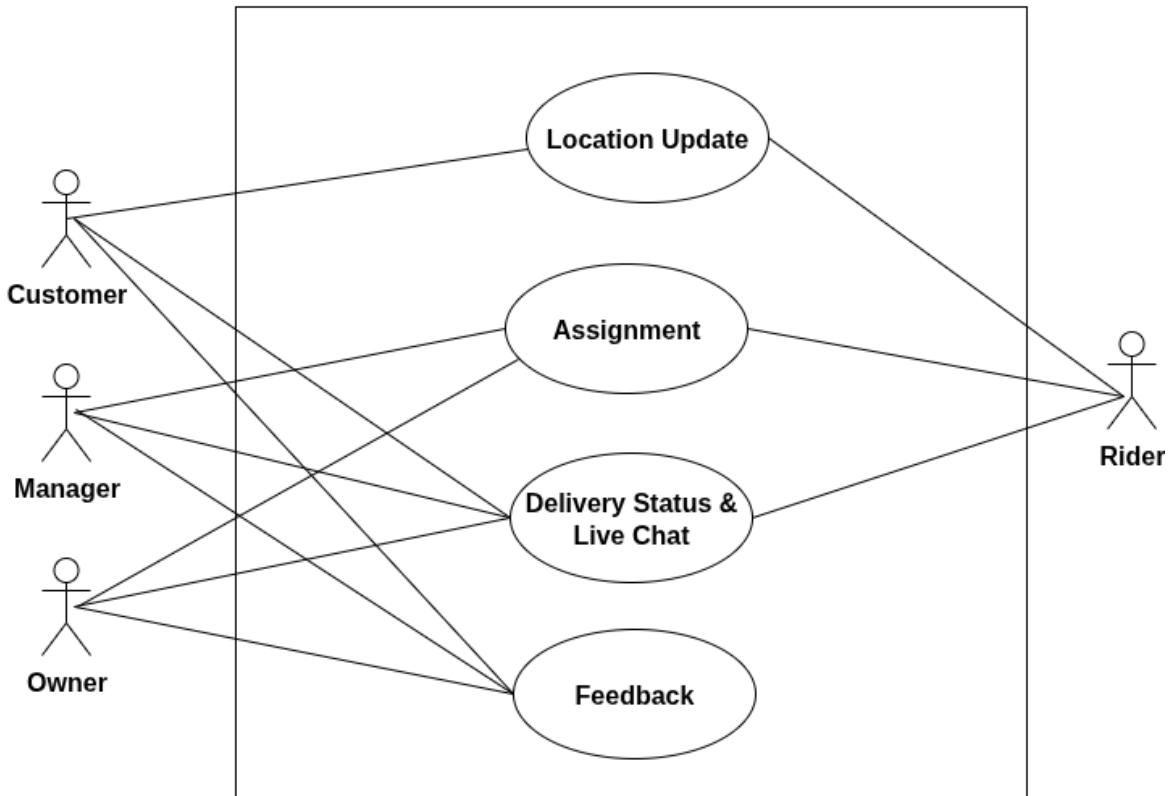
### **Actions and Replies**

- **Action 1:** Customer selects payment method.  
**Reply 1:** System shows options.
- **Action 2:** Customer initiates online payment.  
**Reply 2:** System redirects to gateway.
- **Action 3:** Customer applies voucher.  
**Reply 3:** System validates and applies discount.
- **Action 4:** Payment succeeds.  
**Reply 4:** System confirms to both customer and rider.
- **Action 5:** Payment fails.  
**Reply 5:** System shows error and prompts retry.

### Level 1.5:Delivery Management

**Primary actors:** Customer, Manager, Owner

**Secondary actors:** Rider



## Location Update

The Rider's ability to update their real-time location is critical for successful delivery. Using GPS integration, the Rider's movements are tracked continuously. This information is visible on the customer's app, showing the exact position of their order on a live map. Managers also use this data to oversee Rider activity, manage workloads, and reassign deliveries if necessary. This creates transparency and builds trust, as customers know exactly where their food is at all times.

## Requirements

This feature manages special delivery instructions provided by customers. Examples include "Call before arriving," "Leave at the main gate," or "Handle

package with care.” The system records these requirements and makes them visible to the Rider during delivery. Handling such special conditions ensures personalized service and maximizes customer satisfaction.

## Assignment

Delivery assignment is handled by the Manager, who uses system data about Rider availability, workload, and location to make decisions. The system supports this by automatically suggesting the most suitable Rider for a delivery. Once assigned, the Rider receives the complete delivery details on their dashboard, including customer name, address, contact info, and order ID. This process ensures deliveries are efficient and well-coordinated.

## Delivery Status & Live Chat

**Delivery Status Tracking** allows Riders to mark updates step by step: Picked Up, On the Way, Near Destination, Delivered, or Cancelled. These statuses appear instantly to customers, keeping them informed without the need for extra queries.

**Live Chat** enables direct communication between Customer and Rider. Through a secure messaging interface, customers can clarify instructions, request updates, or make last-minute changes. This improves coordination and minimizes delivery errors.

## Feedback

After the order is delivered, customers can provide feedback through ratings and written reviews. They may rate the overall delivery experience, comment on Rider behavior, and suggest improvements. The system securely stores this feedback in the database, allowing managers to review performance, identify areas for improvement, and ensure consistent service quality.

## Actions and Replies

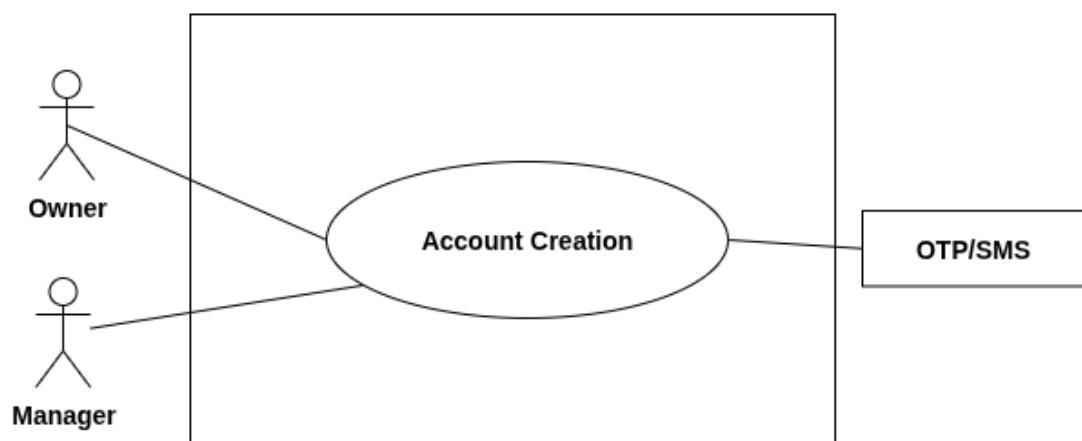
- **Action 1:** Rider updates location.  
**Reply 1:** System displays live tracking.
- **Action 2:** Manager assigns delivery task.  
**Reply 2:** System updates Rider dashboard.
- **Action 3:** Rider updates delivery status.  
**Reply 3:** System shows progress.

- **Action 4:** Customer starts live chat.  
**Reply 4:** System provides secure messaging.
- **Action 5:** Customer submits feedback.  
**Reply 5:** System records and stores it.

### **Level 1.6 : Employee Management**

**Primary actors:** Manager,Owner

**Secondary actors:** OTP/SMS



### **Account Creation**

This process is used to add new employees to the system. The owner or manager enters the employee's details such as name, phone number, and role. Once the

details are submitted, the system sends an OTP via SMS to the provided phone number for verification. After the OTP is confirmed, the account is activated and the employee can log in and start using the system. This verification process ensures that only legitimate employees gain access.

### **Action and Reply:**

**Action:** Employee details are entered into the system.

**Reply:** The system generates an OTP and sends it to the employee via SMS.

**Action:** Employee enters the OTP to verify the account.

**Reply:** The system checks the OTP and confirms if it is correct.

**Action:** OTP verification is successful.

**Reply:** The employee account is activated and ready for login.

## **Activity Diagram**

Activity diagram is an important behavioral diagram in UML diagram to describe dynamic aspects of the system. Activity diagram is essentially an advanced version of flowchart that models the flow from one activity to another activity.

Here are the main reasons why we need activity diagrams:

**Visualize Workflows**

- They provide a clear picture of how a process or use case proceeds step by step.
- Easy for both technical and non-technical stakeholders to understand.

**Model Business Processes**

- Helps to capture how business operations or system processes actually work.
- Useful in requirements analysis to ensure no step is missed.

**Show Parallel and Conditional Flows**

- Unlike flowcharts, activity diagrams handle **decisions**, **loops**, and **concurrent activities** very well.
- This helps to understand complex scenarios.

**Bridge Between Requirements and Design**

- Translates textual requirements (like in an SRS) into a graphical model.

- Developers and testers can use it to ensure the system meets the functional requirements.

**Detect Issues Early**

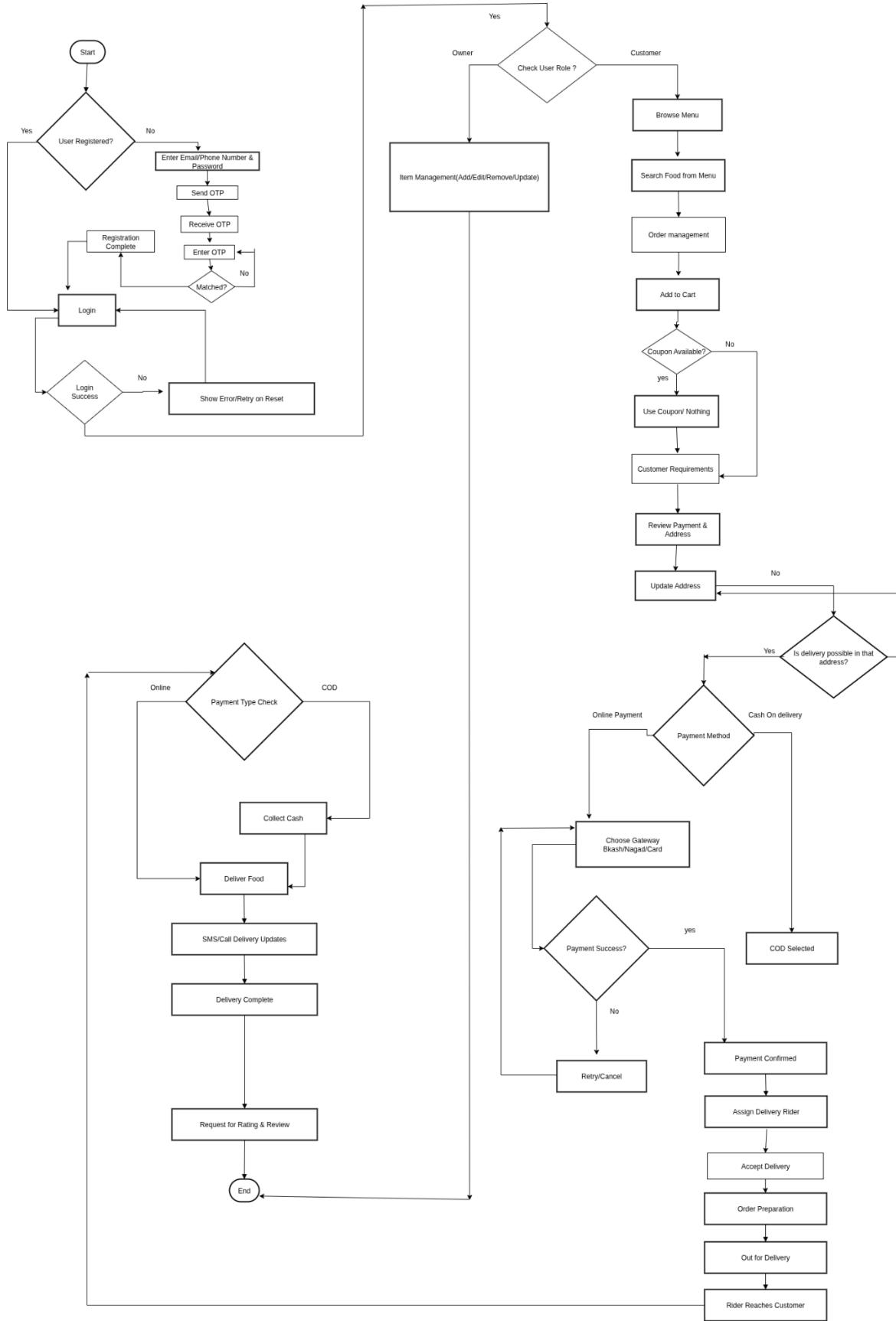
- Makes it easier to find bottlenecks, redundancies, or missing steps in the process before implementation.

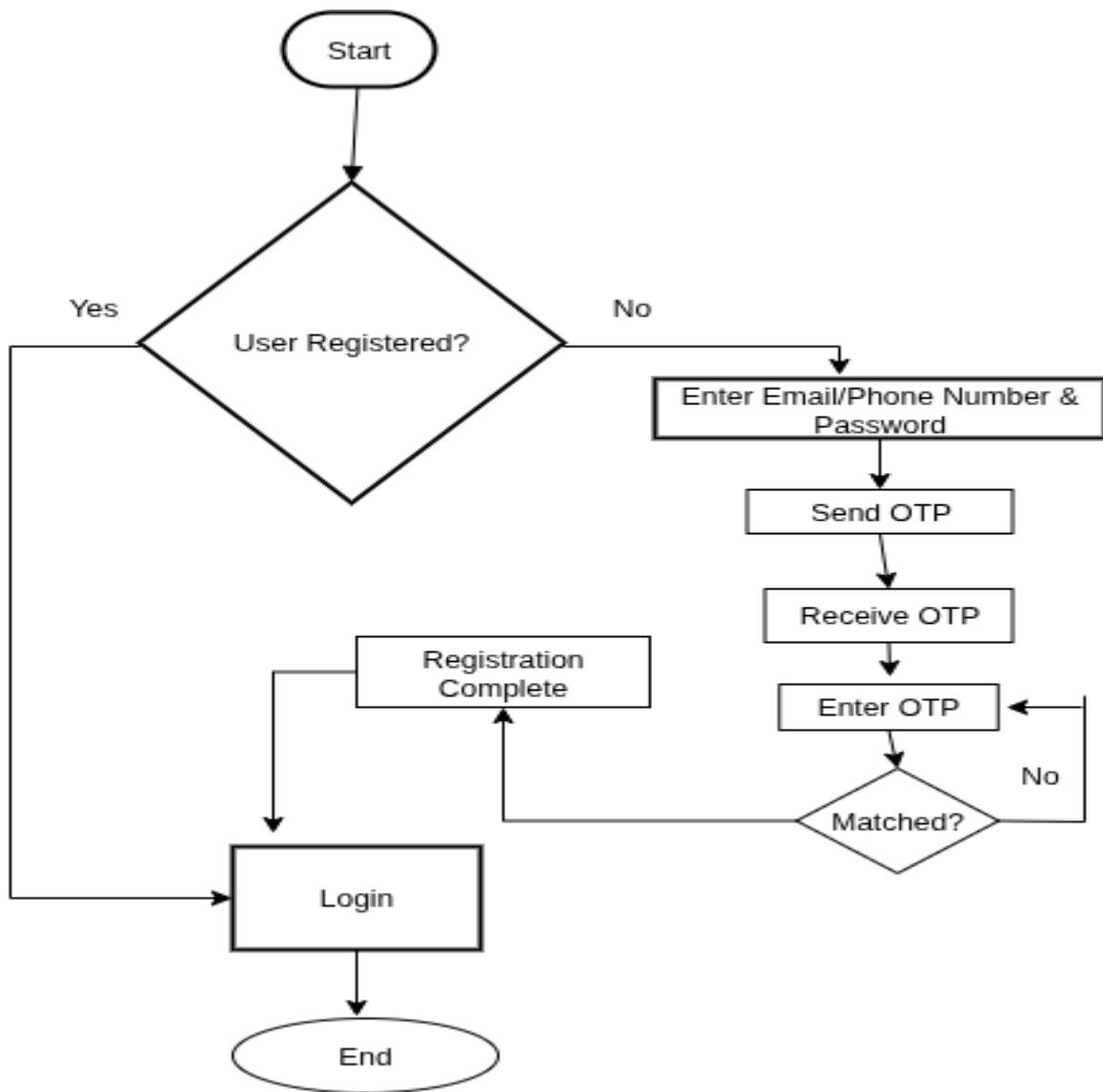
**Support Use Case Realization**

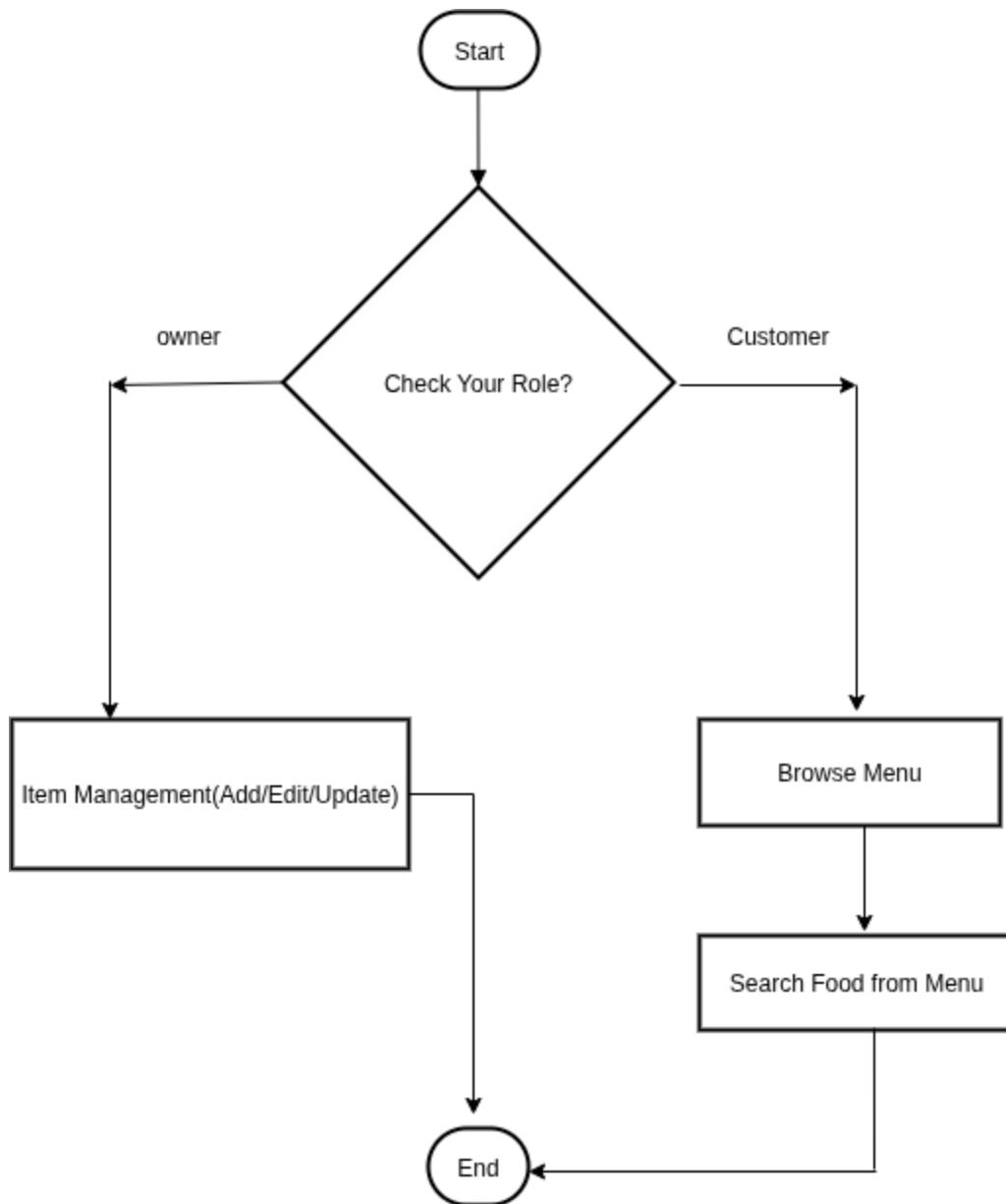
- They are often used alongside **use case diagrams** to show how a use case is actually executed in detail.

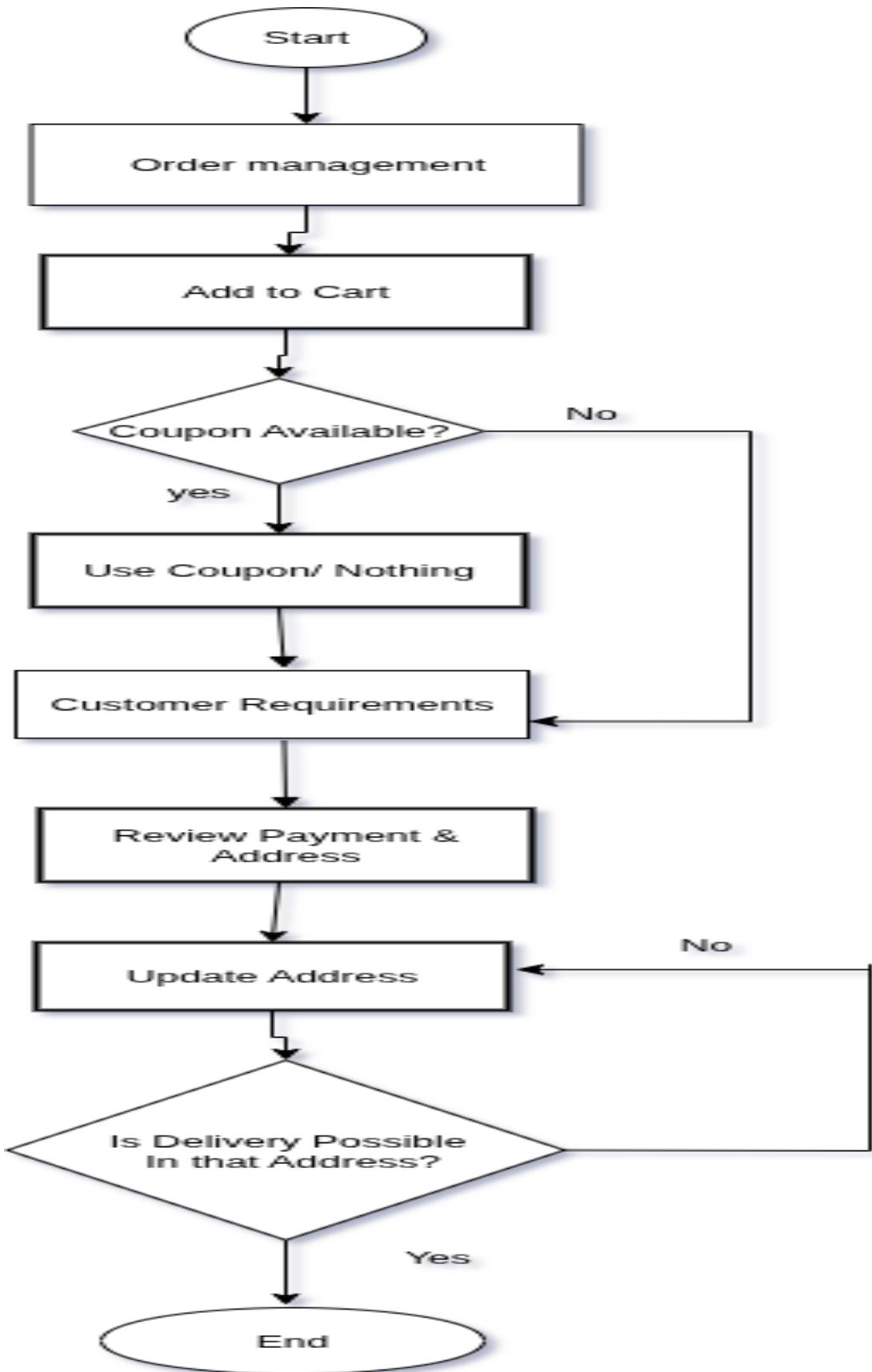
The activity diagrams are: (Reference Use Case Diagram)

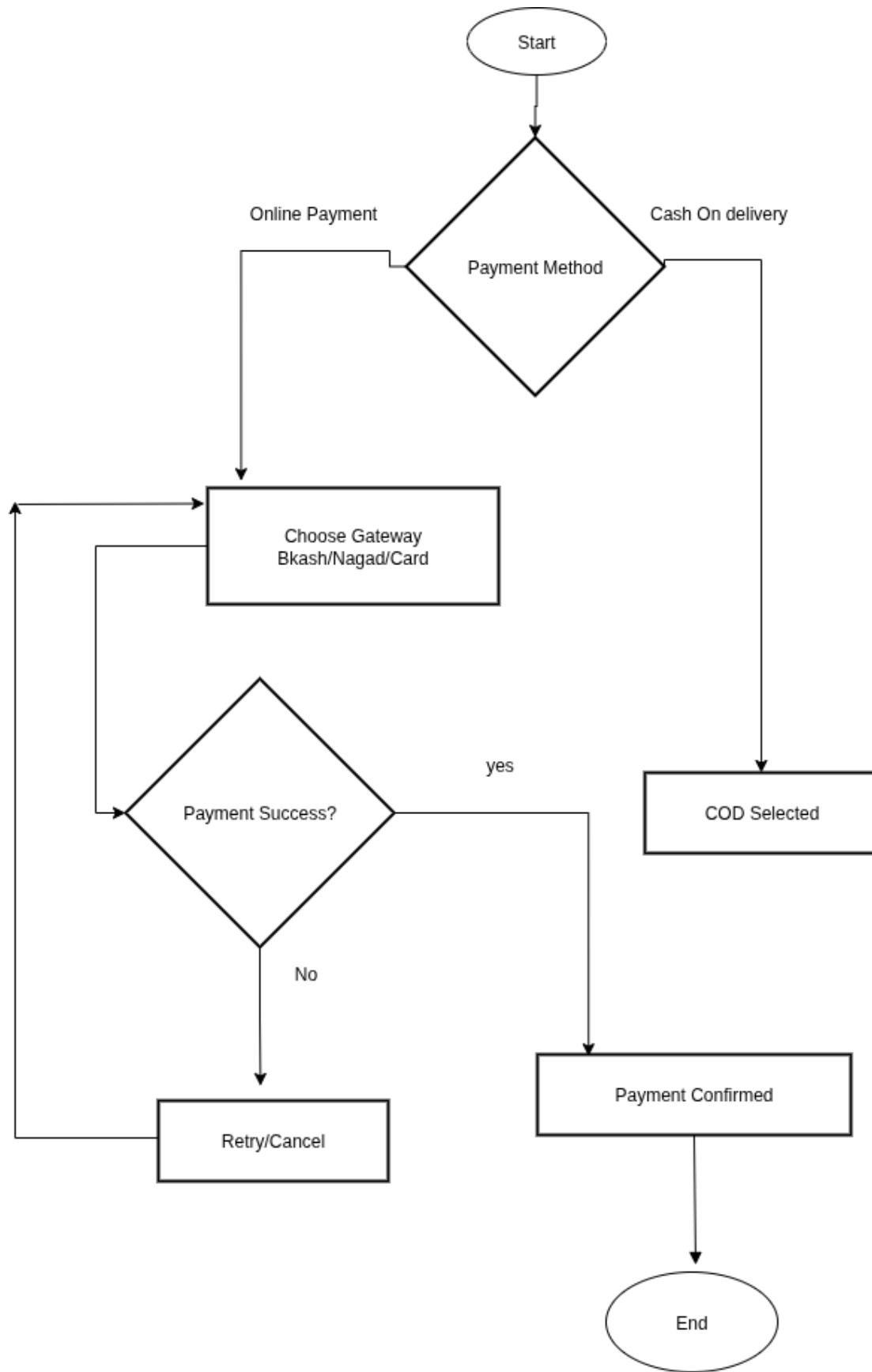
- Level 1: DU Food Point
- Level 1.1: Authentication
- Level 1.2: Menu Management
- Level 1.3: Order Management
- Level 1.4: Payment
- Level 1.5: Delivery Management

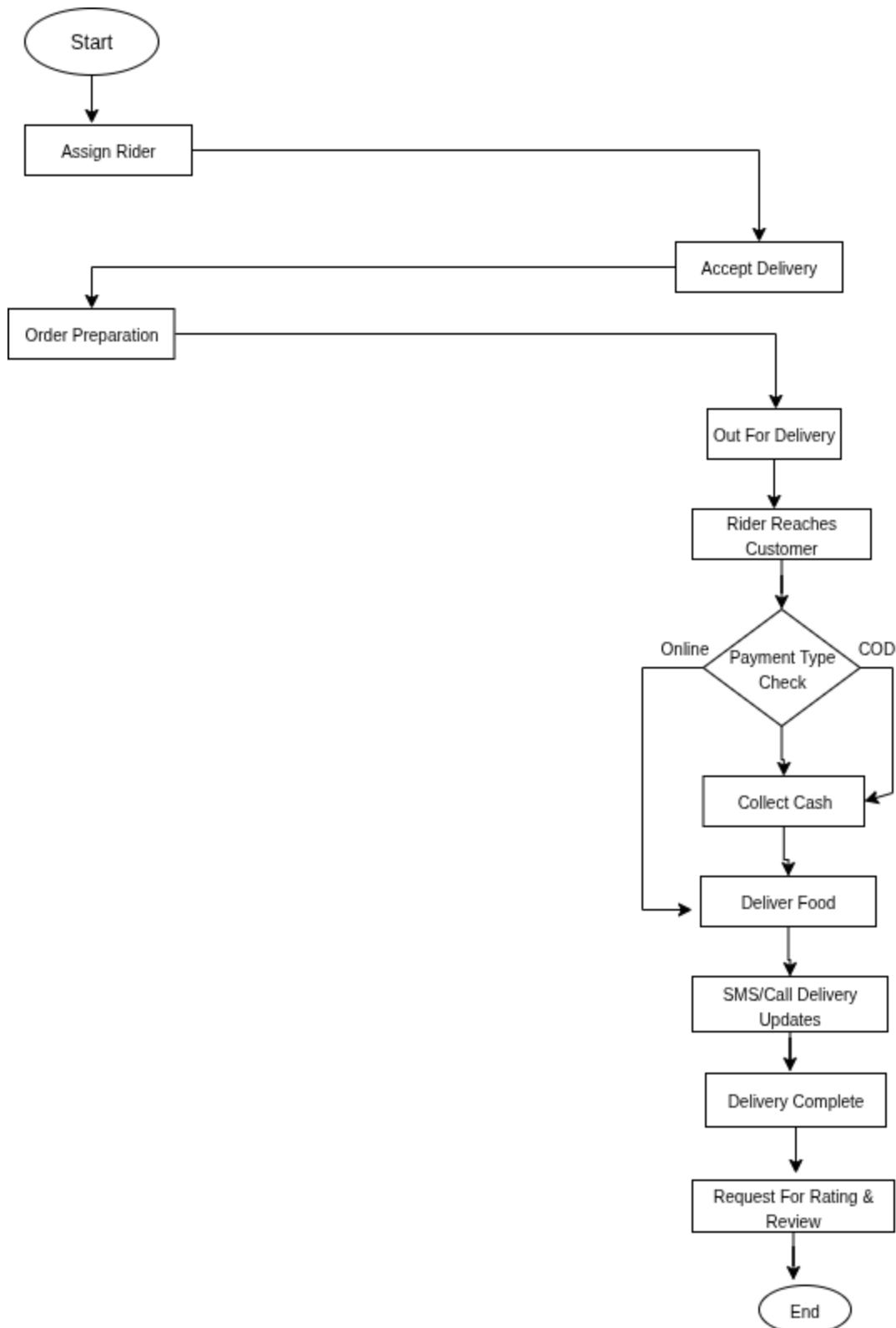










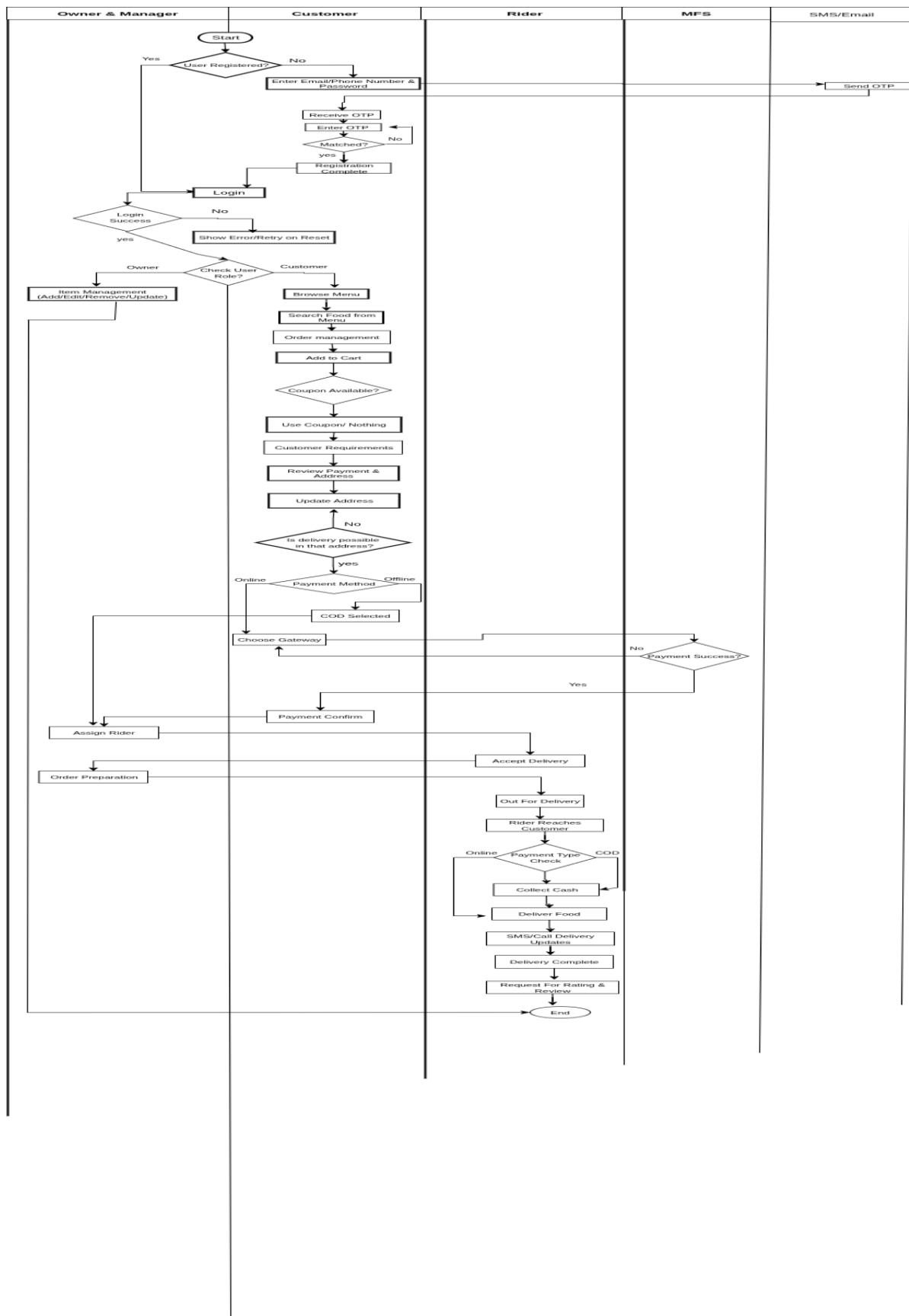


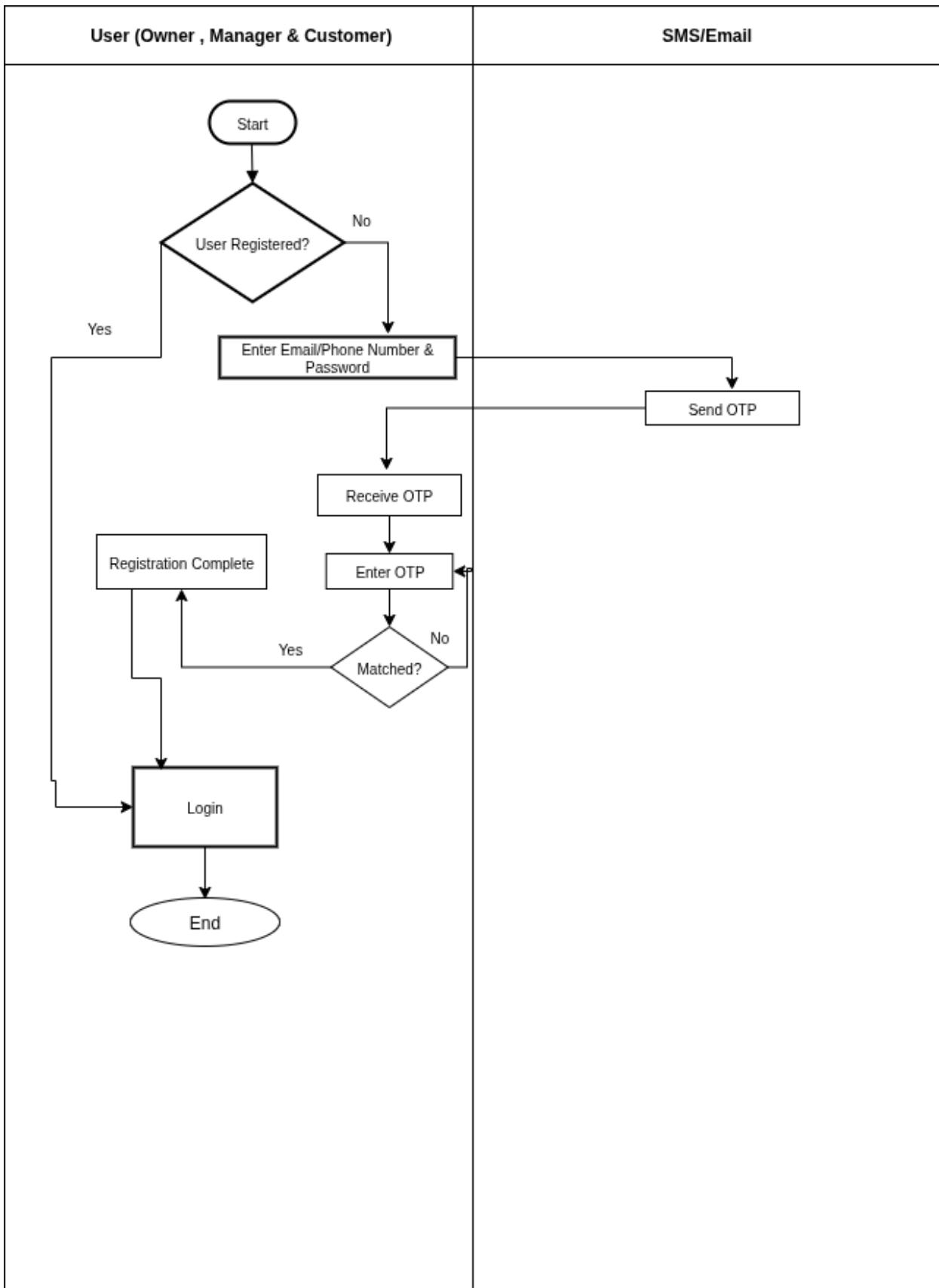
Swimlane diagram

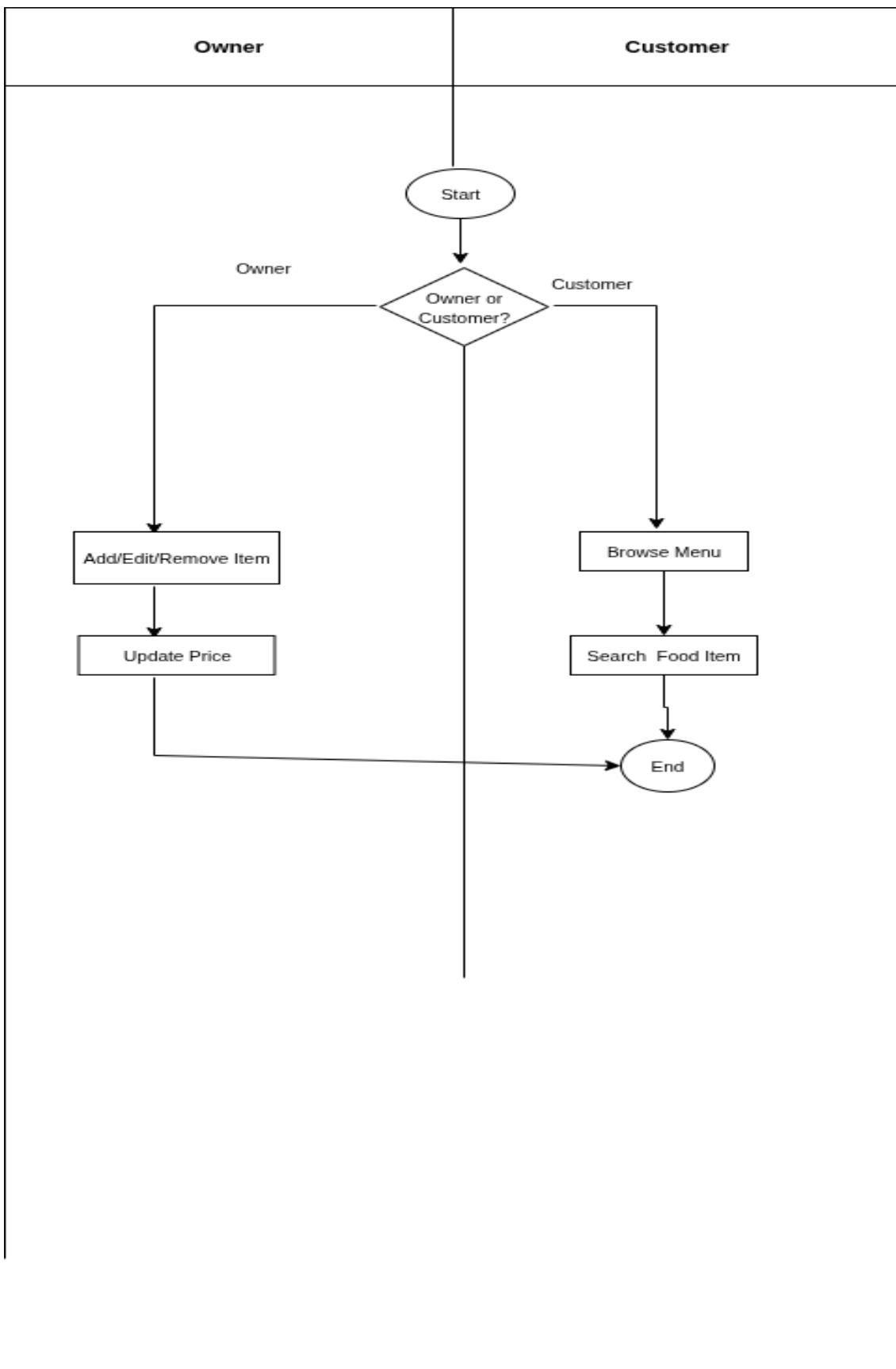
A swimlane diagram is a type of flowchart, which diagrams a process from start to finish, but it also divides these steps into categories to help distinguish which departments or employees are responsible for each set of actions. It is based on the analogy of lanes in a pool, as it places process steps within the horizontal or vertical “swimlanes” of a particular department, work group or employee, thus ensuring clarity and accountability.

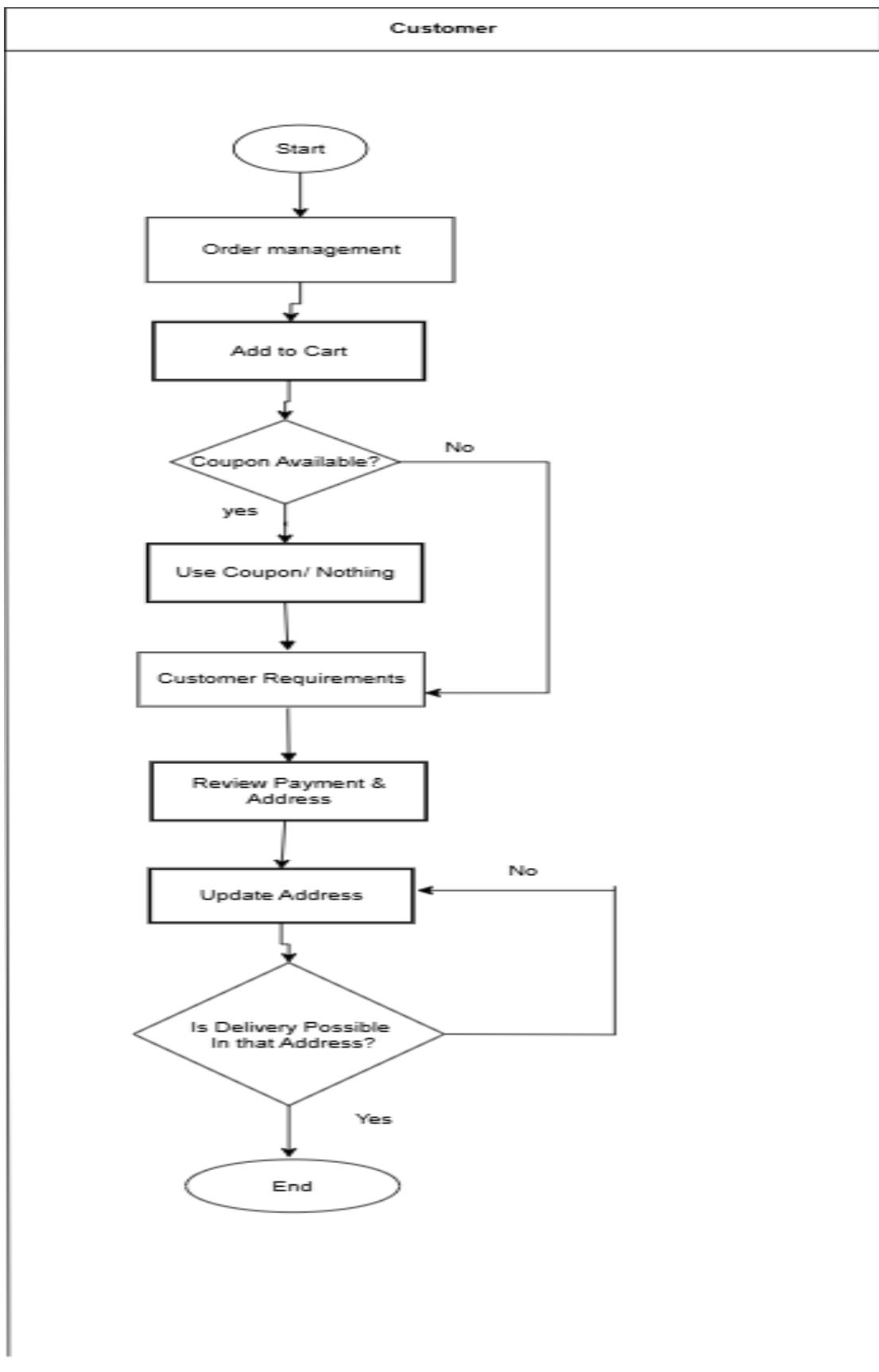
The activity diagrams are:

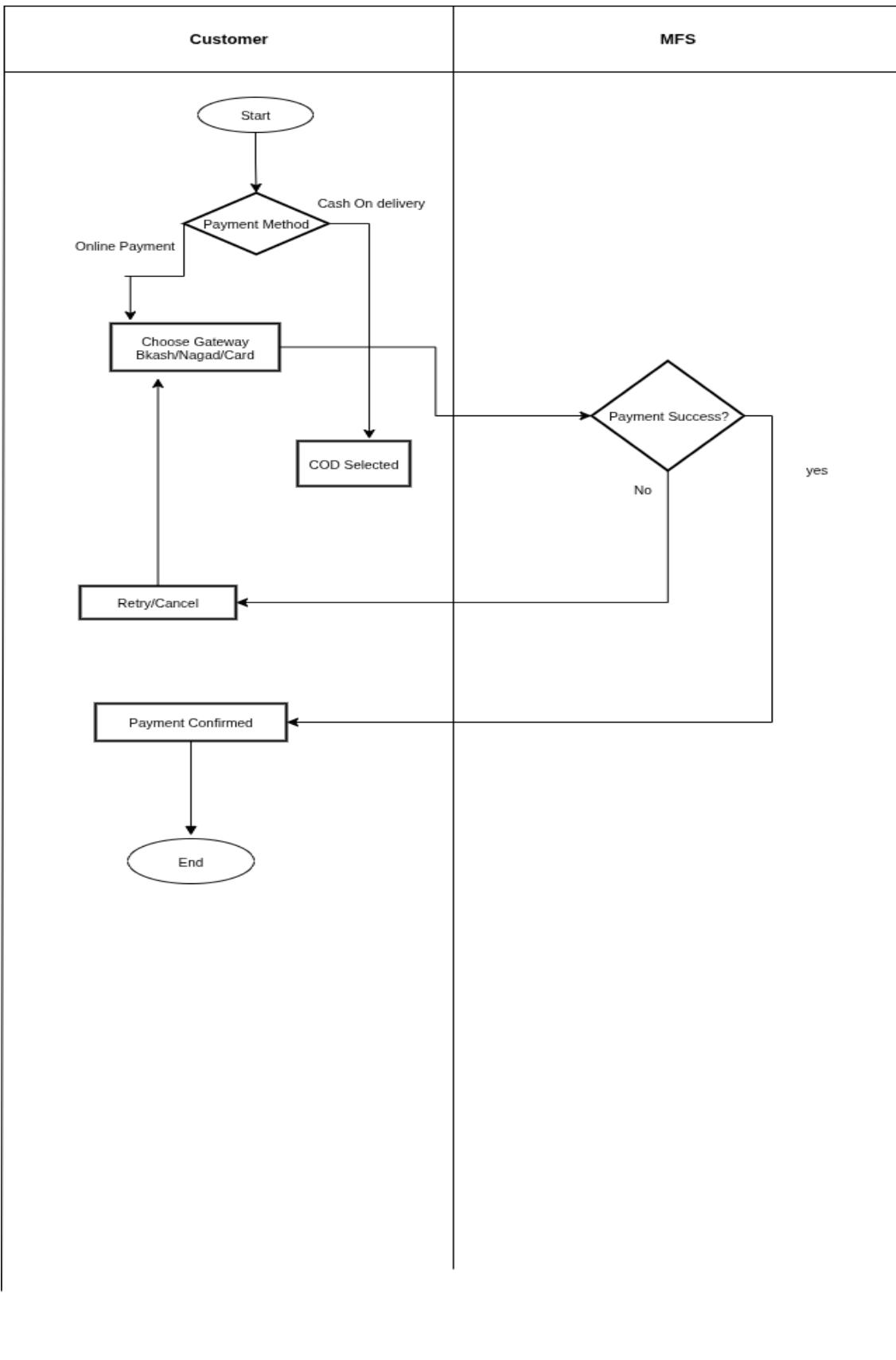
- Level 1: DU Food Point
- Level 1.1: Authentication
- Level 1.2: Menu Management
- Level 1.3: Order Management
- Level 1.4: Payment
- Level 1.5: Delivery Management

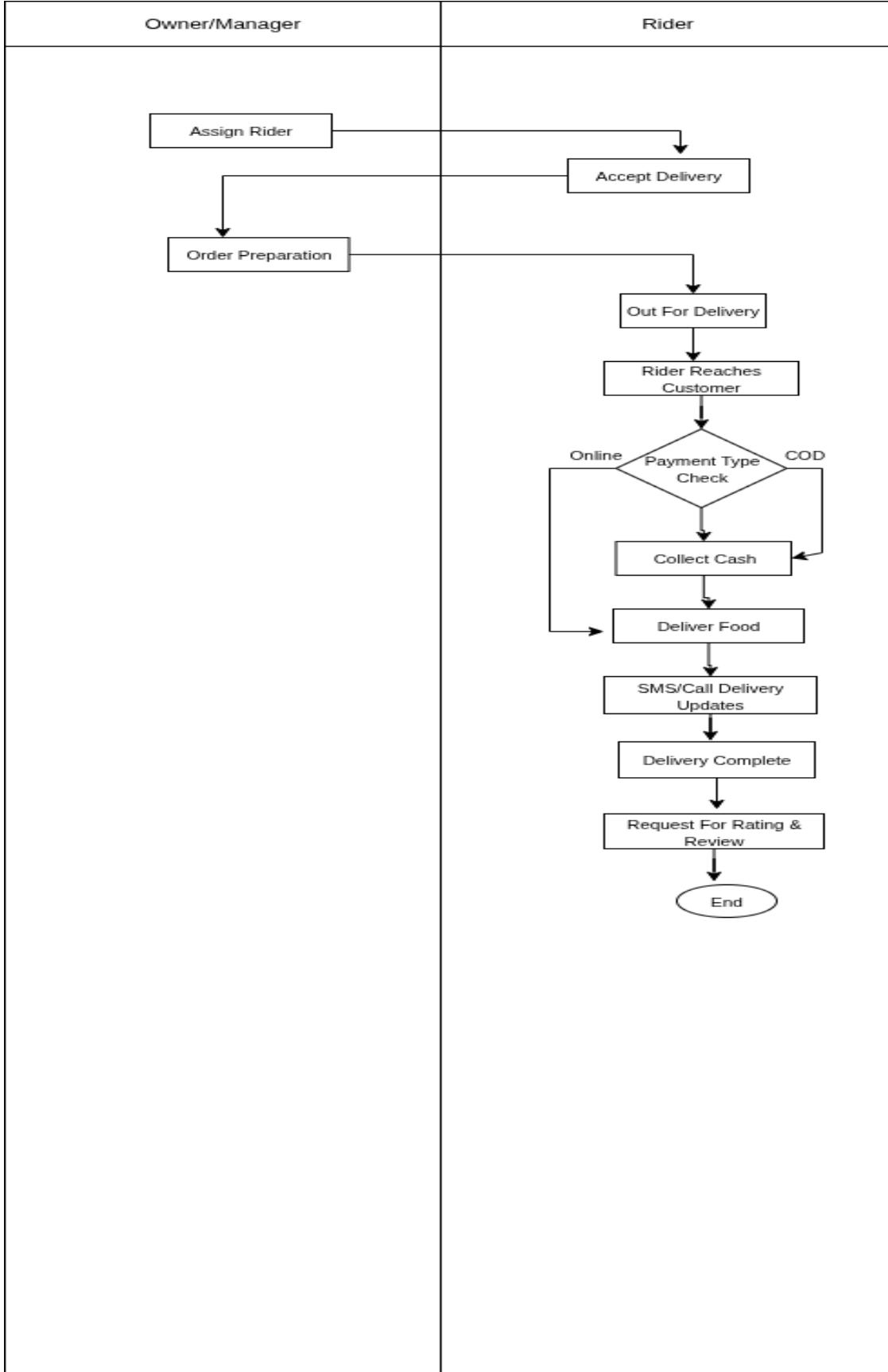












We need a Swimlane Diagram because it not only shows the flow of activities (like a normal activity diagram) but also makes it clear who is responsible for each activity.

Here are the main reasons:

### **1. Clarifies Roles and Responsibilities**

- Each lane shows what a specific actor, department, or system does.
- Prevents confusion about “who does what.”

### **2. Visualizes Interactions Between Actors**

- Shows how activities move from one role/system to another.
- Helps identify dependencies and communication points.

### **3. Improves Understanding of Complex Processes**

- Breaks down workflows by responsibility, making them easier to follow.
- Especially useful when multiple people/systems are involved.

### **4. Highlights Bottlenecks and Handoffs**

- Makes it easier to spot delays or inefficiencies where work passes between lanes.

### **5. Great for Business & System Design**

- In SRS or UML design, it helps both **technical teams** (developers, testers) and **non-technical stakeholders** (managers, clients) to clearly understand the process.

## **Data based modeling**

Database modelling in SRS is the analysis and representation of the data that the system must store, process, and manage. It usually involves identifying entities,

attributes, and relationships based on the requirements gathered.

**Problem Space:** In software engineering, the problem space is the set of user needs, goals, and requirements that describe what the system should do, without considering how it will be implemented. It focuses on the real-world problem the system must solve.

**Solution Space:** The solution space is the set of design choices, technologies, and implementations that describe how the system will fulfill the requirements defined in the problem space. It focuses on the technical realization of the solution.

### Data Object Identification:

SI No.	Noun	P/S	Attributes
1	Customer	S	First name, Last name, Father's name, Mother's name, Date of birth, Present address, Permanent address, Email, Phone number, Password, Account, Login, Credential, Role
2	System	S	
3	Information	P	
4	Contact	P	
5	Details	P	
6	Account	S	
7	Services	P	
8	First name	S	
9	Last name	S	
10	Father's name	S	
11	Mother's name	S	
12	Date of birth	S	
13	Present address	S	
14	Permanent address	S	
15	Email	S	
16	Phone number	S	
17	Password	S	
18	Use	P	
19	Login	P	
20	User	S	
21	Employee	S	
22	Owner	S	First name, Last name, Role, Access, Option, Assignment,

			Dashboard, Menu, Functionality, Update, Management, Action, Communication
23	Authentication	S	
24	Feature	P	
25	Role	P	
26	Access	P	
27	Option	P	
28	Credential	S	
29	Menu	S	
30	OTP	S	
31	Verification	S	
32	Attempt	P	
33	Window	P	
34	Administrator	S	
35	Food item	S	ID, Name, Description, Image, Price, Availability, Status, Tag, Discount, Suggestion, Quantity, Cart
36	Experience	P	
37	Backend	S	
38	Food category	S	ID, Name, Description, Tag, Image, Status, Availability, Price, Trend, Functionality
39	Price	S	
40	Description	P	
41	Image	P	
42	Status	P	
43	Availability	P	
44	Tag	S	
45	Homepage	S	
46	Suggestion	S	
47	Action	S	
48	Cache	S	
49	API	S	
50	Sync	S	
51	Trail	P	
52	ID	S	
53	Timestamp	P	
54	Accountability	P	
55	Format	S	

56	Card	S	
57	Trend	P	
58	Discount	S	
59	Voucher	S	
60	Cart	S	
61	History	P	
62	Order	S	ID, Timestamp, Status, Quantity, Total amount, Stage (Pending, Preparation, Ready), Progression, Checkout, Confirmation, Cart, Voucher, Discount, Order detail
63	Banner	S	
64	Meme	S	
65	Log	S	
66	Management	P	
67	Functionality	P	
68	Update	P	
69	Quantity	S	
70	Preference	P	
71	Note	P	
72	Checkout	S	
73	Progression	P	
74	Stage	P	
75	Pending	S	
76	Preparation	S	
77	Ready	S	
78	Way	P	
79	Delivery	S	
80	Confirmation	S	
81	Manager	S	First name, Last name, Role, Access, Option, Assignment, Dashboard, Menu, Functionality, Update, Management, Action, Communication
82	Rider	S	First name, Last name, ID, Location, GPS, Route guidance, Proximity, Transit, Coordination, Workload, Assignment, Notification, Map, Continuum, Alternative, Time, Delivery, Way
83	Location	S	
84	GPS	S	

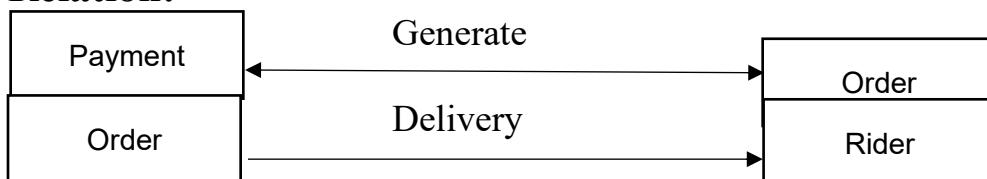
85	Notification	S	
86	Communication	S	
87	Instruction	P	
88	Feedback	S	Rating, Note, Experience, Response, Inquiry, Complaint, Issue, Quality, Monitoring, Communication, Instruction, Reliability
89	Rating	S	
90	Reliability	P	
91	Payment	S	Method, Choice, Online payment, MFS, COD, Provider, Number, Gateway, Transaction, Receipt, Amount, Balance, Timeout, Error, SMS, Message, Failure
92	Method	P	
93	Choice	P	
94	Online payment	S	
95	MFS	S	
96	COD	S	
97	Provider	P	
98	Number	P	
99	Gateway	S	
100	Transaction	S	
101	Receipt	P	
102	Amount	P	
103	SMS	S	
104	Error	P	
105	Message	S	
106	Failure	P	
107	Balance	P	
108	Timeout	P	
109	Activity	P	
110	Transparency	P	
111	Integrity	P	
112	Platform	S	
113	Assignment	P	
114	Dashboard	S	
115	Request	P	
116	Map	S	
117	Proximity	P	
118	Workload	P	
119	Order detail	S	

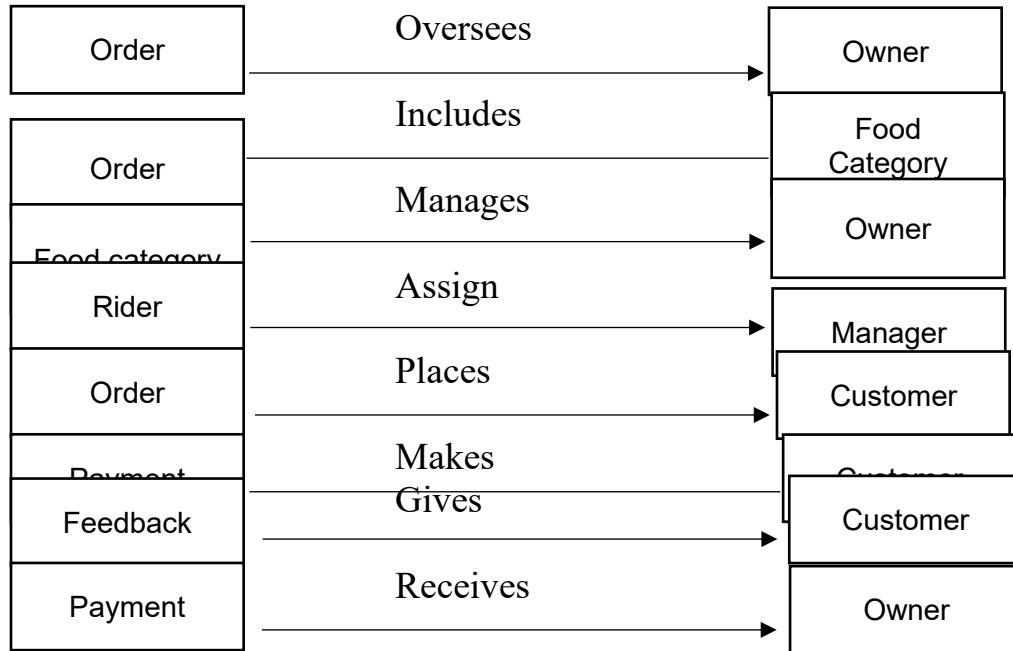
120	Route guidance	S	
121	Coordination	P	
122	Selection	P	
123	Detection	P	
124	Landmark	P	
125	Area	P	
126	Estimate	P	
127	Time	P	
128	Alternative	P	
129	Continuum	P	
130	Transit	P	
131	Audit	S	
132	Case	P	
133	Reason	P	
134	Recipient	P	
135	Inquiry	P	
136	Issue	P	
137	Complaint	P	
138	Response	P	
139	Filter	P	
140	Quality	P	
141	Monitoring	S	

## Final Data Objects:

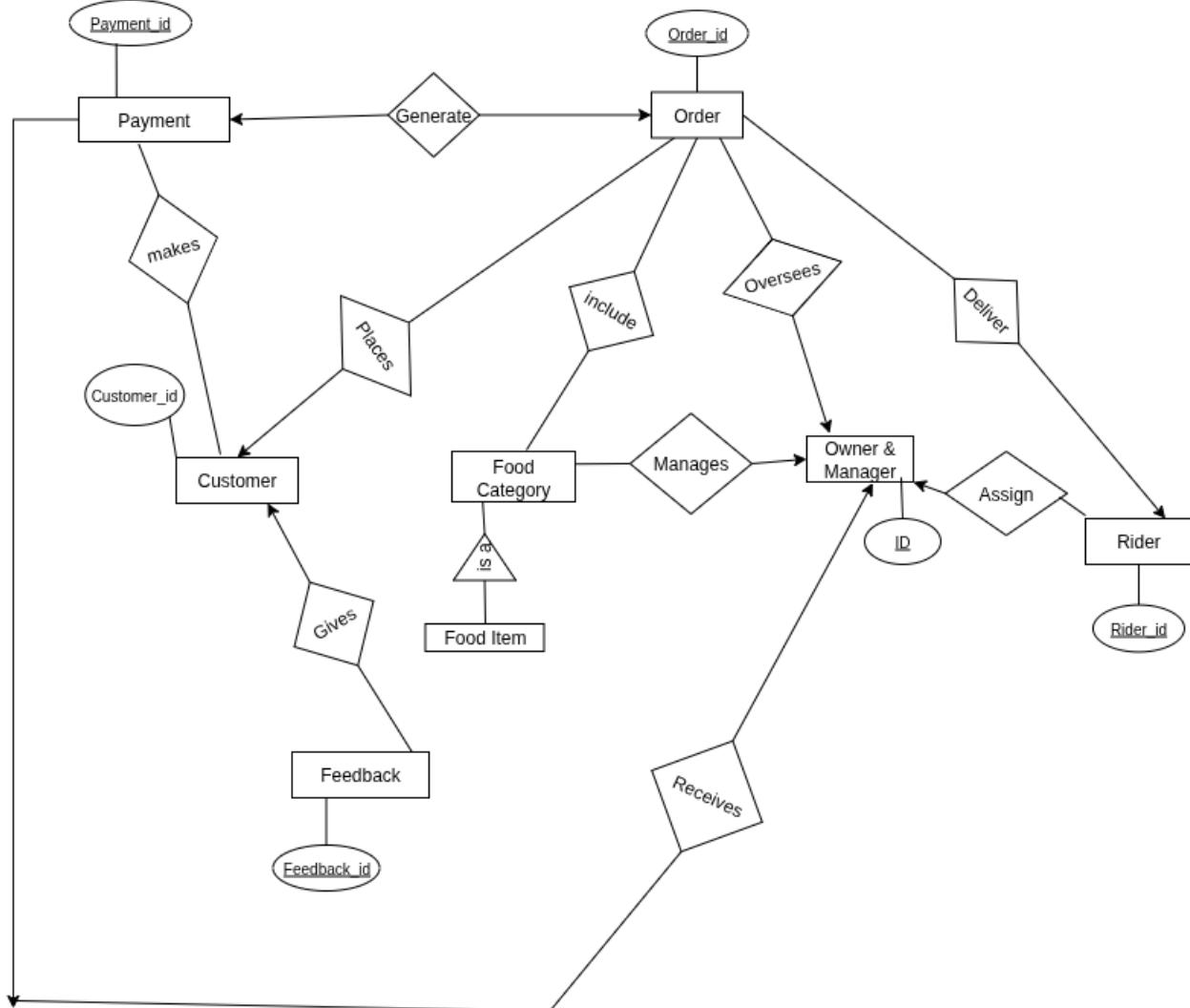
1. Customer
2. Rider
3. Owner/Manager
4. Order
5. Payment
6. Food category
7. Food item
8. Feedback

## Relation:





## ER Diagram:



## Schema:

Customer		
Attributes	Type(Size)	Key
Customer ID	NUMBER	Primary key
First Name	VARCHAR(50)	
Last Name	VARCHAR(50)	
Father Name	VARCHAR(50)	
Mother Name	VARCHAR(50)	
Date of Birth	VARCHAR(50)	
Email	VARCHAR(50)	
Phone Number	VARCHAR(50)	

Present Address	VARCHAR(50)	
Permanent Address	VARCHAR(50)	
Password	VARCHAR(50)	
Account Status	VARCHAR(50)	
Feedback ID	NUMBER	Foreign Key
Order ID	NUMBER	Foreign Key
Payment ID	NUMBER	Foreign Key

<b>Rider</b>		
<b>Attributes</b>	<b>Type(Size)</b>	<b>Key</b>
Rider ID	NUMBER	<b>Primary key</b>
First Name	VARCHAR(50)	
Last Name	VARCHAR(50)	
Email	VARCHAR(100)	
Phone Number	VARCHAR(15)	
Current Location	VARCHAR(100)	
Status	VARCHAR(20)	
Rating	DECIMAL(3,2)	
Total Deliveries	NUMBER	
Password	VARCHAR(255)	
<b>Order ID</b>	<b>NUMBER</b>	<b>Foreign Key</b>

<b>Owner/Manager</b>		
<b>Attributes</b>	<b>Type(Size)</b>	<b>Key</b>
ID	NUMBER	<b>Primary key</b>
First Name	VARCHAR(50)	
Last Name	VARCHAR(50)	
Email	VARCHAR(100)	
Phone Number	VARCHAR(15)	
Role	VARCHAR(30)	
Password	VARCHAR(255)	
<b>Rider ID</b>	<b>NUMBER</b>	<b>Foreign Key</b>
<b>Payment ID</b>	<b>NUMBER</b>	<b>Foreign Key</b>
<b>Order ID</b>	<b>NUMBER</b>	<b>Foreign Key</b>

<b>Order</b>		
<b>Attributes</b>	<b>Type(Size)</b>	<b>Key</b>
Order ID	NUMBER	<b>Primary key</b>
Customer ID	NUMBER	<b>Foreign key</b>
Order Number	VARCHAR(20)	
Order Status	VARCHAR(30)	
Total Amount	DECIMAL(10,2)	
Discount Amount	DECIMAL(10,2)	
Final Amount	DECIMAL(10,2)	
Delivery Address	VARCHAR(200)	
Payment Method	VARCHAR(30)	
Payment Status	VARCHAR(20)	
<b>Rider ID</b>	<b>NUMBER</b>	

<b>Payment</b>		
<b>Attributes</b>	<b>Type(Size)</b>	<b>Key</b>
Payment ID	NUMBER	<b>Primary key</b>
Order ID	NUMBER	<b>Foreign key</b>
Transaction ID	VARCHAR(100)	
Payment Method	VARCHAR(30)	
Payment Gateway	VARCHAR(50)	
Provider	VARCHAR(50)	
Provider Number	VARCHAR(20)	
Amount	DECIMAL(10,2)	
Payment Status	VARCHAR(20)	
COD Status	VARCHAR(20)	
Online Payment Status	VARCHAR(20)	
Customer ID	NUMBER	<b>Foreign Key</b>
Order ID	NUMBER	<b>Foreign Key</b>

<b>Food category</b>		
<b>Attributes</b>	<b>Type(Size)</b>	<b>Key</b>
Category ID	NUMBER	<b>Primary key</b>
Category Name	VARCHAR(100)	
Description	TEXT	
Image	VARCHAR(255)	
Status	VARCHAR(20)	
Display Order	NUMBER	

Manager ID	NUMBER	Foreign Key
Order ID	NUMBER	Foreign Key

Food item		
Attributes	Type(Size)	Key
Food ID	NUMBER	Primary key
Category ID	NUMBER	Foreign key
Food Name	VARCHAR(100)	
Description	TEXT	
Price	DECIMAL(10,2)	
Image	VARCHAR(255)	
Status	VARCHAR(20)	

Feedback		
Attributes	Type(Size)	Key
Feedback ID	NUMBER	Primary key
Customer ID	NUMBER	Foreign key
Overall Rating	NUMBER(1)	
Food Quality Rating	NUMBER(1)	
Delivery Rating	NUMBER(1)	
Service Rating	NUMBER(1)	
Comment	TEXT	
Feedback Type	VARCHAR(30)	

## Class Based Modeling

Class-based modeling represents the objects that the system will manipulate, the operations that will be applied to the objects, relationships between the objects and the collaborations that occur between the classes that are defined.

### List of Verbs in User Story:

SL No.	Verbs
1	Register
2	Provide information
3	Create account
4	Access services
5	Set password
6	Complete registration
7	Log in

8	Use credentials
9	Authenticate user
10	Gain access
11	Store credentials
12	Reset password
13	Send OTP
14	Verify OTP
15	Manage menu
16	Add item
17	Edit item
18	Update item details
19	Change availability
20	Remove item
21	Highlight featured item
22	Appear customer interface
23	Try suggested item
24	Mark item unavailable
25	Delete item
26	Restore item
27	Reorganize categories
28	Schedule promotion
29	Upload media
30	Trigger cache refresh
31	Log activity
32	Display menu
33	Search item
34	Recommend item
35	Review past order
36	Reorder items
37	Select item
38	Customize order
39	Finalize order
40	Apply voucher
41	Maintain cart
42	Modify order/cart
43	Checkout
44	Track order
45	Assign rider
46	Provide order details

47	Receive notification
48	Communicate with rider
49	Deliver order
50	Submit feedback
51	Rate service
52	Choose payment method
53	Enter payment details
54	Integrate with payment API
55	Generate OTP
56	Validate OTP
57	Confirm transaction
58	Collect cash
59	Check voucher validity
60	Retry payment
61	Switch payment method
62	Ensure security
63	Suggest rider assignment
64	Specify delivery location
65	Detect GPS location
66	Verify coordinates
67	Notify customer
68	Update status
69	Respond to message

## General Classification

Candidate classes were then characterized in seven general classes. The seven general characteristics are as follows:

1. External entities
2. Things
3. Events
4. Roles
5. Organizational units
6. Places
7. Structures

SL No.	Noun	General Classification
--------	------	------------------------

1	Customer	4, 5
2	System	2
3	Account	2, 7
4	First name	2
5	Last name	2
6	Father's name	2
7	Mother's name	2
8	Date of birth	2
9	Present address	2, 6
10	Permanent address	2, 6
11	Email	2, 7
12	Phone number	2
13	Password	2
14	User	4, 5
15	Notification	2
16	Owner	4, 5
17	Authentication	3, 7
18	Credential	2
19	Menu	2, 6, 7
20	OTP	2
21	Verification	3
22	Administrator	4, 5
23	Food item	2
24	Backend	6, 7
25	Food category	2, 5, 7
26	Price	2
27	Tag	2
28	Homepage	6, 7
29	Suggestion	2, 3
30	Action	3
31	Cache	2, 6
32	API	2
33	Sync	3
34	ID	2
35	Format	2, 7
36	Card	2, 7
37	Discount	2, 3
38	Voucher	2, 3
39	Cart	2, 6

40	Order	2, 3, 7
41	Banner	2, 7
42	Meme	2
43	Log	2
44	Quantity	2
45	Checkout	3, 6
46	Pending	3
47	Preparation	3
48	Ready	3
49	Delivery	3, 4
50	Confirmation	3
51	Manager	4, 5
52	Rider	4, 5
53	Location	2, 6
54	GPS	1
55	Notification	2
56	Communication	2, 3
57	Feedback	2, 3
58	Rating	2
59	Payment	2, 3
60	Online payment	2, 3
61	MFS	1
62	COD	2
63	Gateway	2, 7
64	Transaction	2, 3
65	SMS	2, 3
66	Message	2, 3
67	Platform	2, 6
68	Dashboard	2, 6
69	Order detail	2, 7
70	Route guidance	2, 6
71	Audit	2, 3
72	Monitoring	2,

### Potential To Be A Class:

1. Customer
2. Account
3. Present address

4. Permanent address
5. Email
6. User
7. Owner
8. Authentication
9. Menu
10. Administrator
11. Food category
12. Backend
13. Homepage
14. Suggestion
15. Cache
16. Cart
17. Order
18. Banner
19. Format
20. Card
21. Discount
22. Voucher
23. Delivery
24. Manager
25. Rider
26. Location
27. Communication
28. Feedback
29. Payment
30. Gateway
31. Transaction
32. SMS
33. Platform
34. Dashboard
35. Route guidance
36. Audit

## Selection Criteria

The candidate classes are then selected as classes by six Selection Criteria. A candidate class generally becomes a class when it fulfills around three characteristics.

1. Retain information
2. Needed services
3. Multiple attributes
4. Common attributes
5. Common operations
6. Essential requirements

Potential general classified nouns to become a class after selection criteria:

<b>SL No.</b>	<b>Noun</b>	<b>Selection Criteria</b>
1	Customer	
2	Account	
3	Present address	
4	Permanent address	
5	Email	
6	User	1, 2, 3, 4, 5, 6
7	Owner	
8	Authentication	1, 2, 3, 4, 5, 6
9	Menu	1, 2, 3, 4, 5, 6
10	Administrator	
11	Food category	1, 2, 3, 4, 5, 6
12	Backend	
13	Homepage	
14	Suggestion	
15	Cache	
16	Cart	1, 2, 3, 4, 5, 6
17	Order	1, 2, 3, 4, 5, 6
18	Banner	
19	Format	
20	Card	

21	Discount	
22	Voucher	
23	Delivery	1, 2, 3, 4, 5, 6
24	Manager	
25	Rider	
26	Location	
27	Communication	
28	Feedback	1, 2, 3, 4, 5, 6
29	Payment	1, 2, 3, 4, 5, 6
30	Gateway	
31	Transaction	
32	SMS	
33	Platform	
34	Dashboard	
35	Route guidance	
36	Audit	

### Selected Class:

1. User
2. Authentication
3. Menu
4. Food Category
5. Cart
6. Order
7. Payment
8. Delivery
9. Feedback

### Attributes and Method Identification:

Class Name	Attributes	Method
User	- firstName - lastName - fatherName - motherName	+ registration() + provideInformation() + logIn() + resetPassword()

	<ul style="list-style-type: none"> <li>-dateOfBirth</li> <li>-presentAddress</li> <li>-permanentAddress</li> <li>-email</li> <li>-phoneNumber</li> <li>-password</li> <li>-userRole</li> </ul>	<ul style="list-style-type: none"> <li>+ accessServices()</li> <li>+ detectGPSLocation()</li> <li>+ respondToMessage()</li> </ul>
<b>Authentication</b>	<ul style="list-style-type: none"> <li>-isAuthenticated</li> <li>-loginAttempts</li> <li>-otpCode</li> <li>-otpExpiry</li> <li>-sessionToken</li> </ul>	<ul style="list-style-type: none"> <li>+authenticateUser()</li> <li>+SendOTP()</li> <li>+verifyOTP()</li> <li>+validateCredentials()</li> <li>+generateSessionToken()</li> <li>+LogActivity()</li> <li>+ensureSecurity()</li> </ul>
<b>Menu</b>	<ul style="list-style-type: none"> <li>-itemName</li> <li>-category</li> <li>-price</li> <li>-description</li> <li>-itemImage</li> <li>-stockAvailability</li> <li>-featuredTag</li> <li>-discountBadge</li> <li>-isAvailable</li> </ul>	<ul style="list-style-type: none"> <li>+manageMenu()</li> <li>+addItem()</li> <li>+editItem()</li> <li>+updateItemDetails()</li> <li>+changeAvailability()</li> <li>+RemoveItem()</li> <li>+appearCustomerInterface()</li> <li>+markItemUnavailable()</li> <li>+deleteItem()</li> <li>+RestoreItem()</li> <li>+schedulePromotion()</li> <li>+uploadMedia()</li> <li>+TriggerCacheRefresh()</li> <li>+displayMenu()</li> <li>+searchItem()</li> <li>+recommendItem()</li> </ul>
<b>Food Category</b>	<ul style="list-style-type: none"> <li>-categoryName</li> <li>-categoryDescription</li> <li>-categoryImage</li> <li>-itemCount</li> <li>-isActive</li> </ul>	<ul style="list-style-type: none"> <li>+reorganizeCategories()</li> <li>+addCategory()</li> <li>+editCategory()</li> <li>+removeCategory()</li> <li>+categorizeItem()</li> </ul>
<b>Cart</b>	<ul style="list-style-type: none"> <li>-cartItems</li> <li>-totalAmount</li> <li>-itemQuantity</li> <li>-customNotes</li> <li>-voucherApplied</li> </ul>	<ul style="list-style-type: none"> <li>+selectItem()</li> <li>+customizeOrder()</li> <li>+maintainCart()</li> <li>+modifyOrder()</li> <li>+applyVoucher()</li> </ul>

	-runningTotal	+checkout() +calculateTotal()
<b>Order</b>	-orderID -customerID -orderItems -orderStatus -totalAmount -customNotes -orderTimestamp -estimatedDeliveryTime -riderID	+finalizeOrder() +TrackOrder() +receiveNotification() +updateStatus() +provideOrderDetails() +reviewPastOrder() +reorderItems() +communicateWithRider() +submitFeedback() +rateService()
<b>Payment</b>	-paymentMethod -transactionID -paymentAmount -mfsProvider -mobileNumber -otpCode -paymentStatus -voucherCode -digitalReceipt	+choosePaymentMethod() +enterPaymentDetails() +integrateWithPaymentAPI() +generateOTP() +validateOTP() +confirmTransaction() +collectCash() +checkVoucherValidity() +retryPayment() +switchPaymentMethod() +generateReceipt()
<b>Delivery</b>	-deliveryAddress -gpsCoordinates -riderID -deliveryStatus -estimatedTime -routeGuidance -serviceArea -landmarks -specialInstructions	+assignRider() +suggestRiderAssignment() +verifyCoordinates() +notifyCustomer() +updateStatus() +deliverOrder() +trackLocation() +provideRouteGuidance()
<b>Feedback</b>	-feedbackID -customerID -orderID -rating -comments -feedbackType -responseStatus	+submitFeedback() +rateService() +collectFeedback() +analyzeFeedback() +respondToFeedback()

## Class Cards

A Class Card is a simple, structured way to describe a class during requirements and design modeling. It helps in identifying the main classes of the system, their responsibilities, and how they collaborate with other classes.

<b>User</b>	
<b>Responsibilities</b>	<b>Collaborators</b>
Register new accounts	Authentication
Provide personal information	Delivery
Manage login credentials	Feedback
Specify delivery locations	

<b>Authentication</b>	
<b>Responsibilities</b>	<b>Collaborators</b>
Verify login credentials	User
Generate OTP	Payment
Handle password reset	
Manage user sessions	

<b>Menu</b>	
<b>Responsibilities</b>	<b>Collaborators</b>
Display food items	Food Category
Manage item categories	Cart
Handle item availability	Order
Process search requests	

<b>Food Category</b>	
<b>Responsibilities</b>	<b>Collaborators</b>
Organize menu categories	Menu
Manage category details	
Support item classification	

<b>Cart</b>	
<b>Responsibilities</b>	<b>Collaborators</b>
Maintain selected items	Menu
Calculate total amounts	Order
Handle item modifications	Payment
Apply vouchers and discounts	

<b>Order</b>	
<b>Responsibilities</b>	<b>Collaborators</b>
Track order status	Cart
Manage order lifecycle	Payment
Handle customer communication	Delivery
Provide order details	Feedback

<b>Payment</b>	
<b>Responsibilities</b>	<b>Collaborators</b>
Process payment transactions	Authentication
Handle multiple payment methods	Cart
Validate payment information	Order
Generate digital receipts	

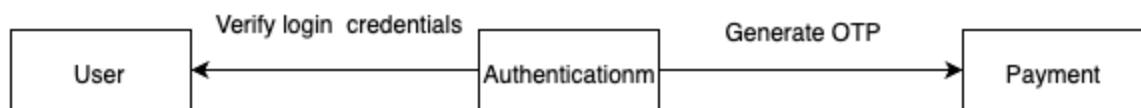
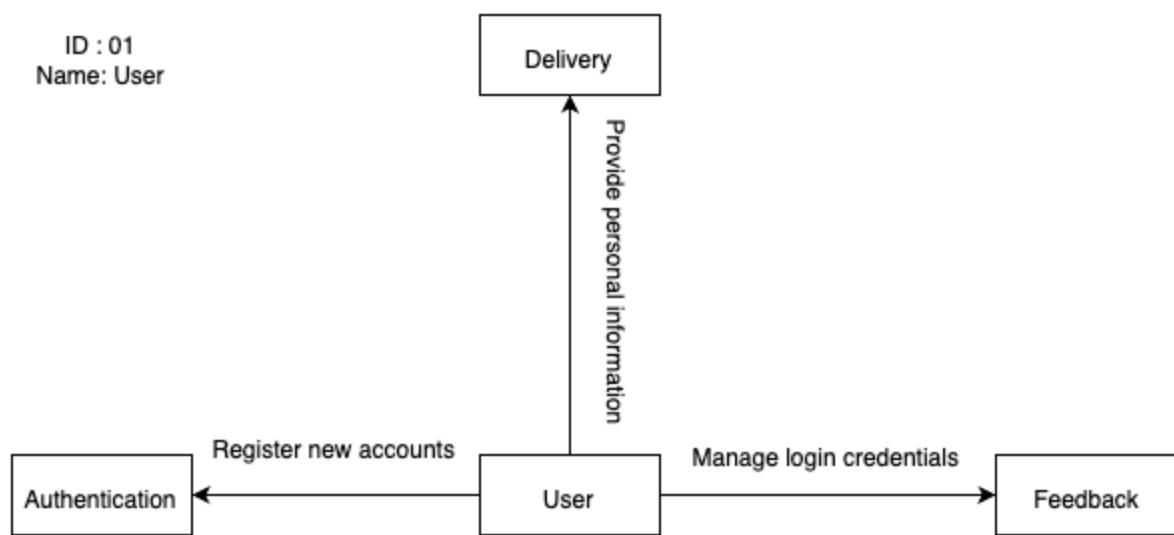
<b>Delivery</b>	
<b>Responsibilities</b>	<b>Collaborators</b>
Assign delivery riders	Order
Track delivery progress	User
Verify delivery locations	Feedback
Provide route guidance	

<b>Feedback</b>	
<b>Responsibilities</b>	<b>Collaborators</b>
Collect customer feedback	Order
Manage service ratings	Delivery
Analyze feedback data	User
Generate feedback reports	

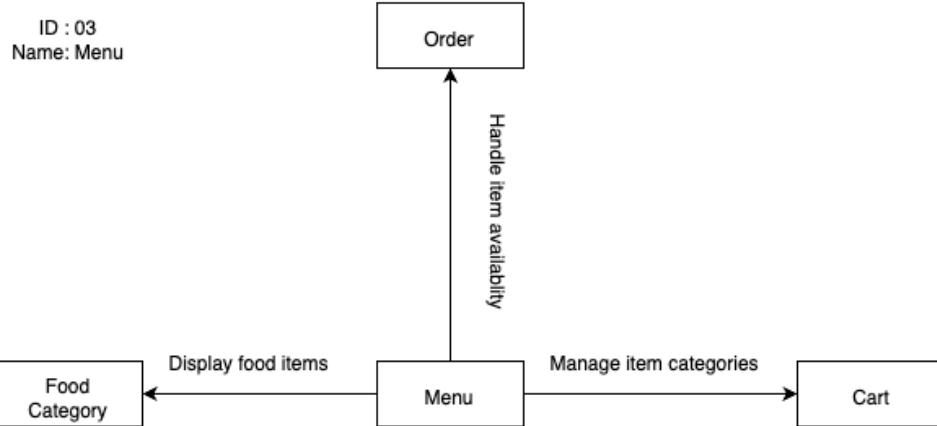
## CRC-Diagram

A CRC diagram (or CRC card model) is a simple, lightweight tool used in object oriented design to capture the essential details of a class.

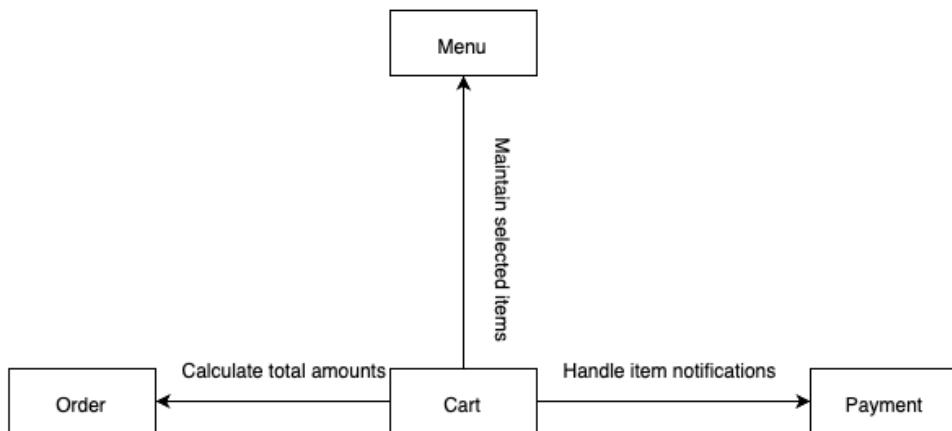
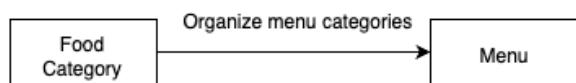
- **Simple & easy to understand** – Quick way to model classes without technical complexity.
- **Encourages collaboration** – Often used in group discussions or brainstorming sessions.
- **Identifies system roles** – Helps decide what each class should do.
- **Defines interactions** – Shows which classes depend on each other.



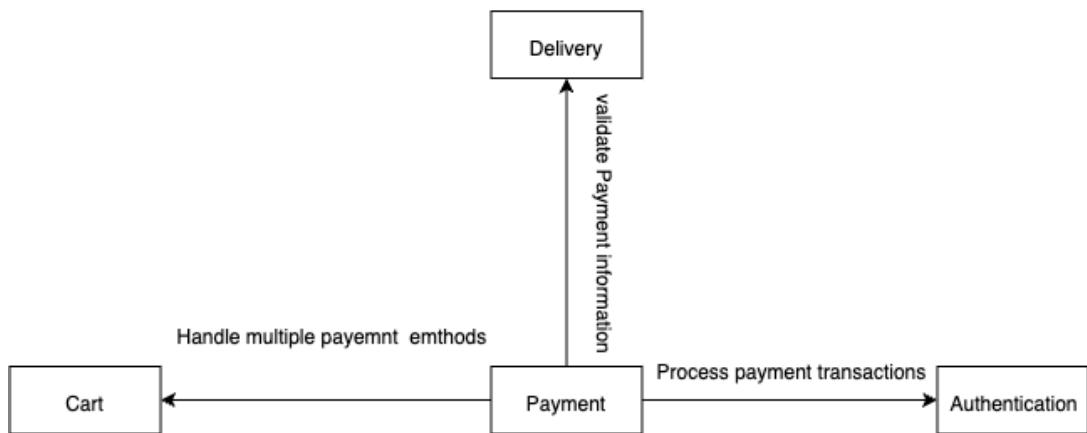
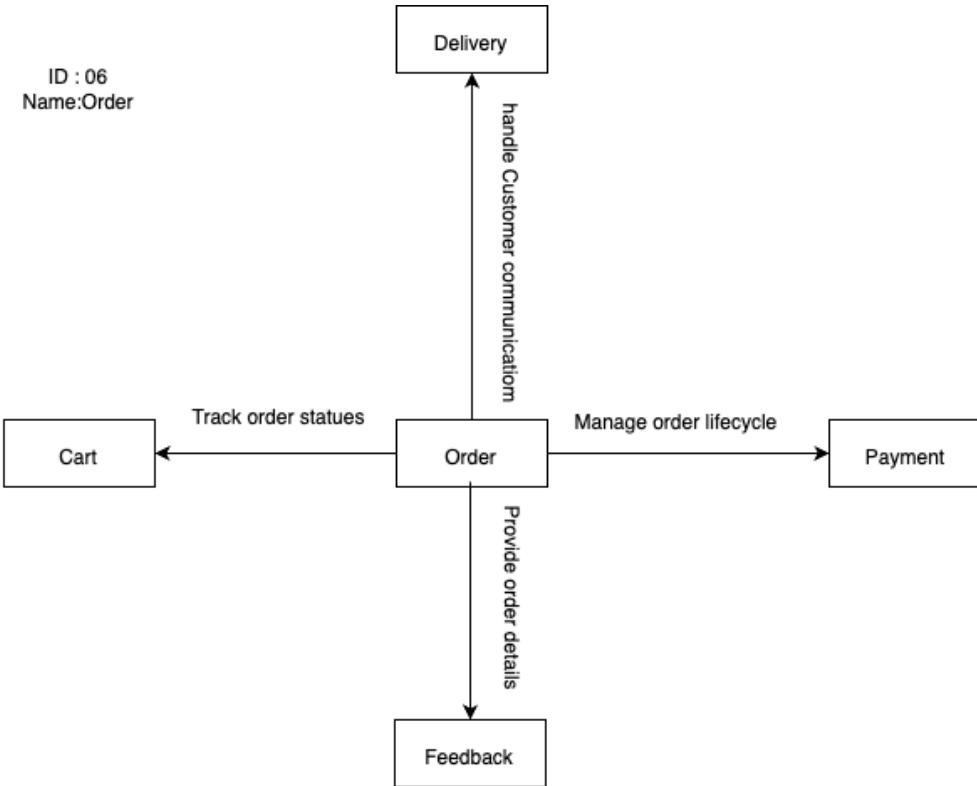
ID : 02  
Name: Authentication



ID : 04  
Name: Food Category

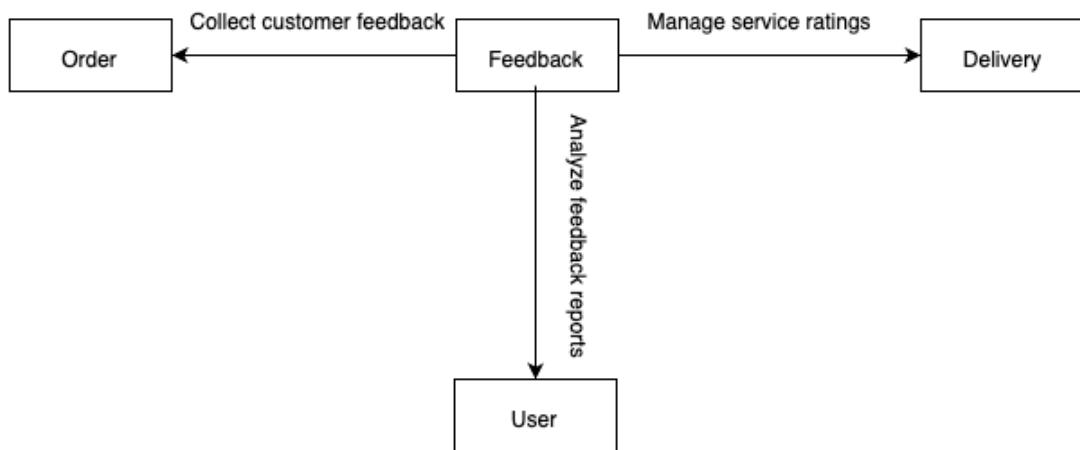
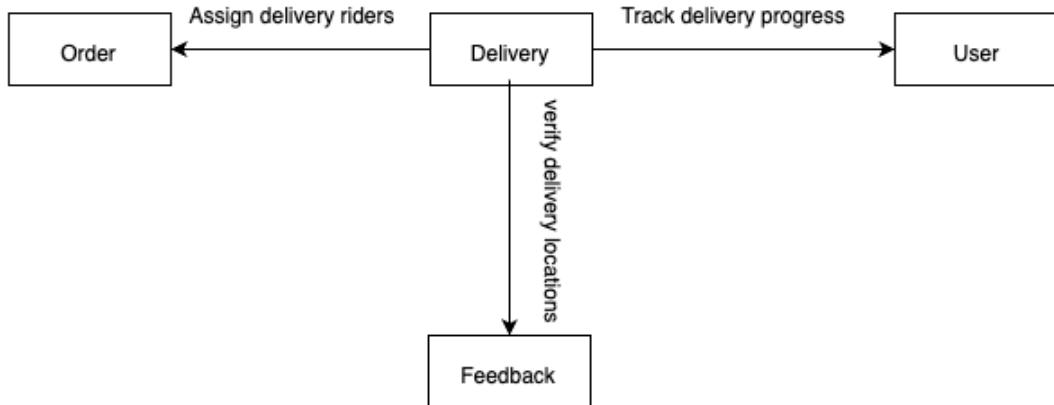


ID : 05  
Name: Cart

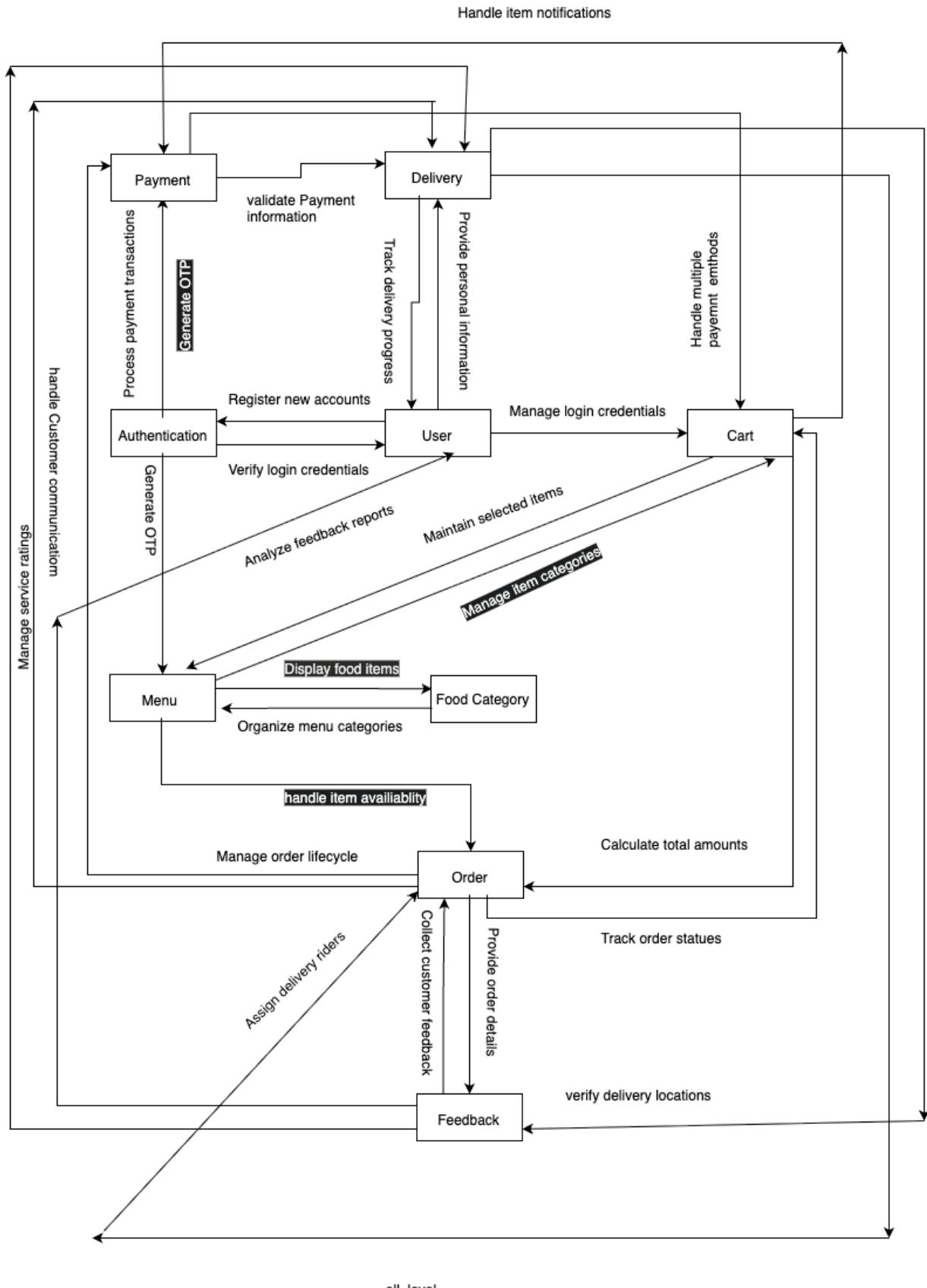


ID : 07  
Name:payment

ID : 08  
Name: Delivery



ID : 09  
Name: Feedback



## Behavioral Modeling

Behavioral modeling is a software engineering technique that represents how a system or its individual components behave in response to internal or external events. It focuses on the dynamic aspects of the system, describing the flow of activities, the changes of states, and the interactions between objects over time. Common tools for behavioral modeling include state transition diagrams, sequence diagrams, activity diagrams, and event tables, which together help to capture the functional behavior of the system beyond its static structure.

No	Initiator	Event	Event Name	Collaborator
01	User	Provide registration details	User Registration	Authentication
02	User	Update profile information	Profile Update	User
03	User	Request password reset	Password Reset Request	Authentication
04	User	View order history	Order History View	Order
05	Authentication	Validate login credentials	User Login	User
06	Authentication	Generate OTP for password reset	OTP Generation	User
07	Authentication	Verify OTP	OTP Verification	User
08	Authentication	Complete password reset	Password Reset Complete	User
09	Authentication	Logout user	User Logout	User
10	Menu	Add new menu item	Menu Item Creation	Food Item
11	Menu	Edit menu item details	Menu Item Update	Food Item
12	Menu	Remove menu item	Menu Item Deletion	Food Item
13	Menu	Update item availability	Availability Update	Food Item
14	Menu	Browse categories	Menu Browse	Food Item
15	Menu	Search items	Menu Search	Food Item
16	Food Item	Create item	Item Creation	Menu
17	Food Item	Update price	Price Update	Menu

18	Food Item	Change description	Description Update	Menu
19	Food Item	Upload image	Image Update	Menu
20	Cart	Add item	Item Addition	Food Item
21	Cart	Remove item	Item Removal	Food Item
22	Cart	Update quantity	Quantity Update	Food Item
23	Cart	Add instructions	Instruction Addition	Food Item
24	Cart	Proceed to checkout	Checkout Initiation	Order
25	Order	Create order	Order Creation	Cart
26	Order	Assign rider	Rider Assignment	Delivery
27	Order	Update status	Order Status Update	Delivery
28	Order	Cancel order	Order Cancellation	Payment
29	Payment	Select method	Payment Method Selection	Order
30	Payment	Process MFS payment	MFS Payment Processing	Authentication
31	Payment	Verify transaction	Payment Verification	Order
32	Payment	Complete payment	Payment Completion	Order
33	Delivery	Accept delivery	Delivery Acceptance	Order
34	Delivery	Track location	GPS Tracking	Order
35	Delivery	Complete delivery	Delivery Completion	Order
36	Feedback	Submit rating	Rating Submission	Order
37	Feedback	Create review	Review Creation	User

## State Transition Diagram (STD)

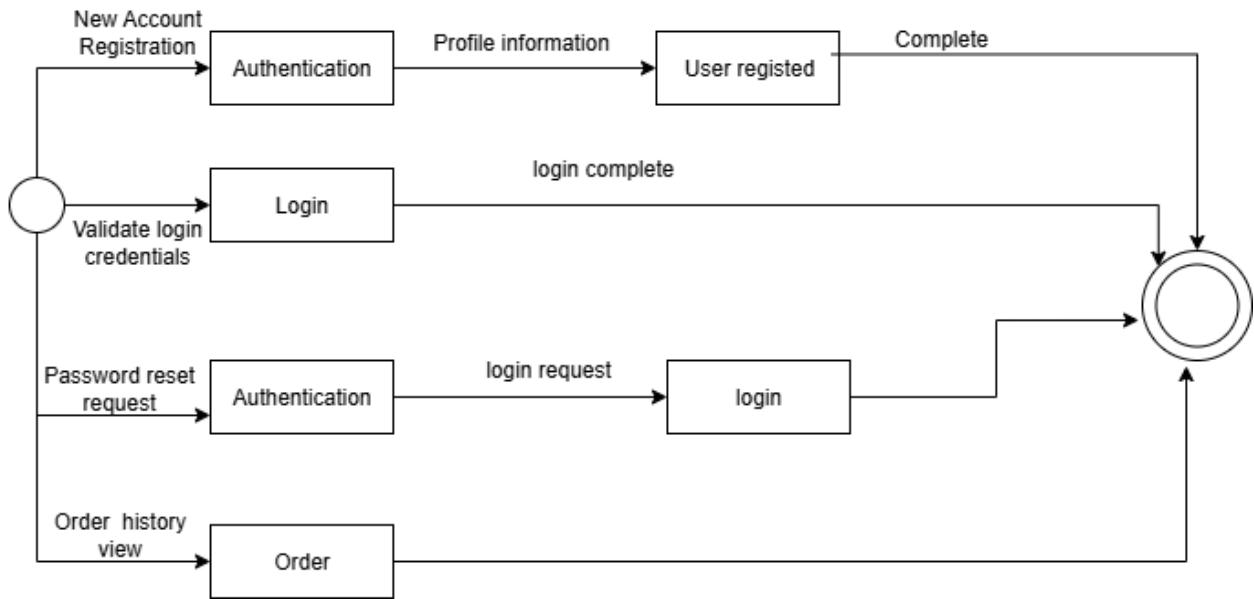
A state transition is the movement of a system from one state to another when a

specific event occurs and (optionally) certain conditions are satisfied.

### **Importance of State Transition:**

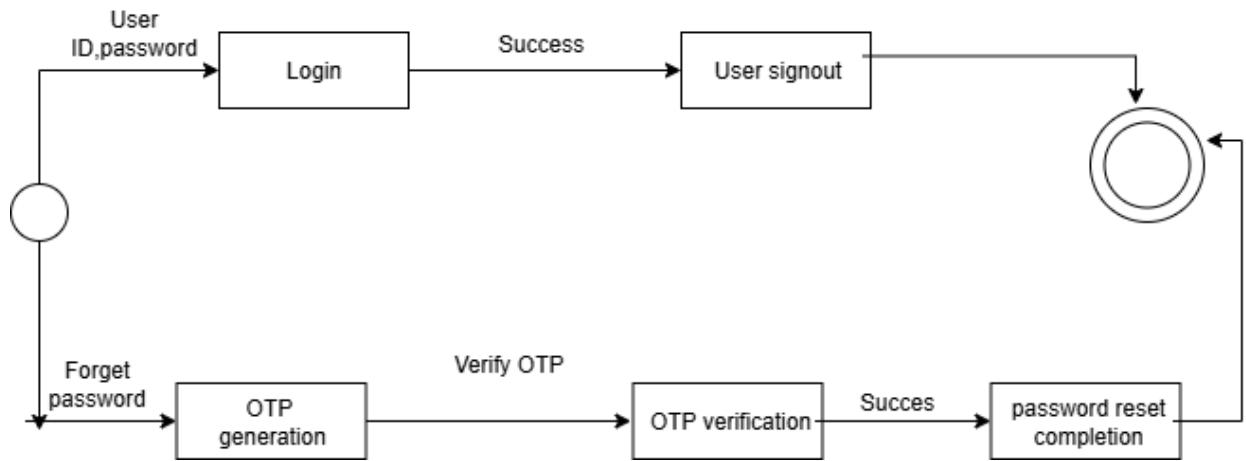
1. **Shows dynamic behavior** – It explains how the system reacts to events and changes over time.
2. **Clarifies system logic** – Makes it easier to understand the sequence of states and what causes them.
3. **Avoids ambiguity** – Ensures that for every event, the system has a defined response.
4. **Helps in testing** – Testers can design cases for each state and transition to ensure correctness.
5. **Improves design quality** – Provides a blueprint for developers to implement event-driven systems correctly.
6. **Handles complex scenarios** – Especially useful where the system has many states.
7. **Supports error handling** – Shows what the system should do in unexpected or invalid states.

ID : 01  
Name : User

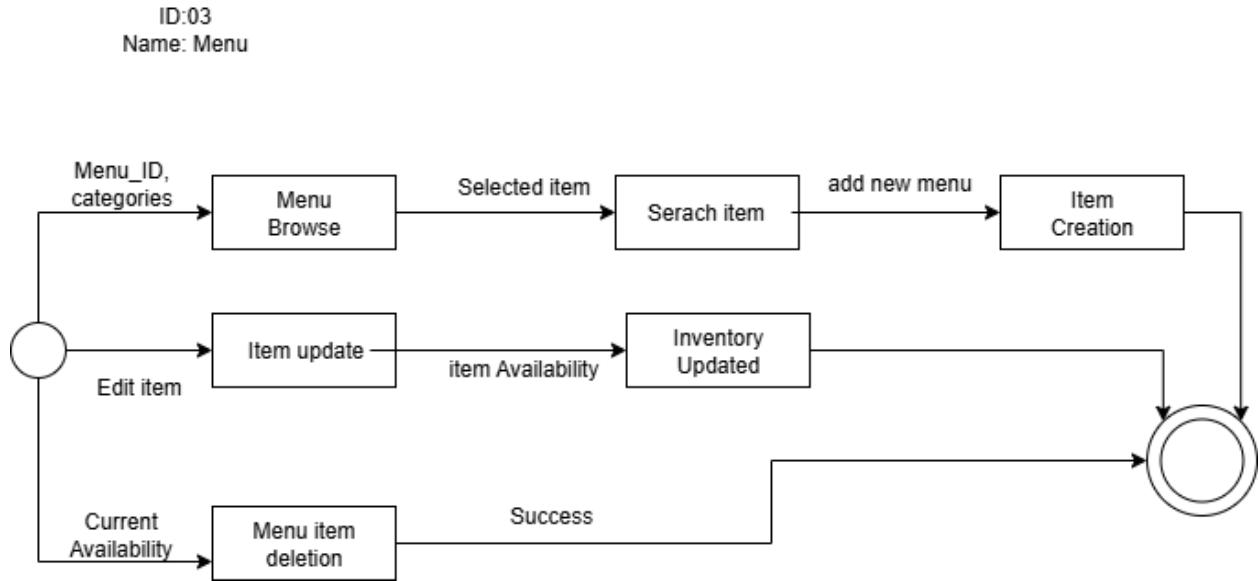


The User module begins with account registration, where a new user provides profile information and is authenticated before being successfully registered. Existing users can log in after validating their credentials, and if a password reset is needed, authentication with OTP is required before resetting. Users are also able to view their order history after logging in. The flow ends once the user completes registration, login, or password reset.

ID:02  
name:Authentication

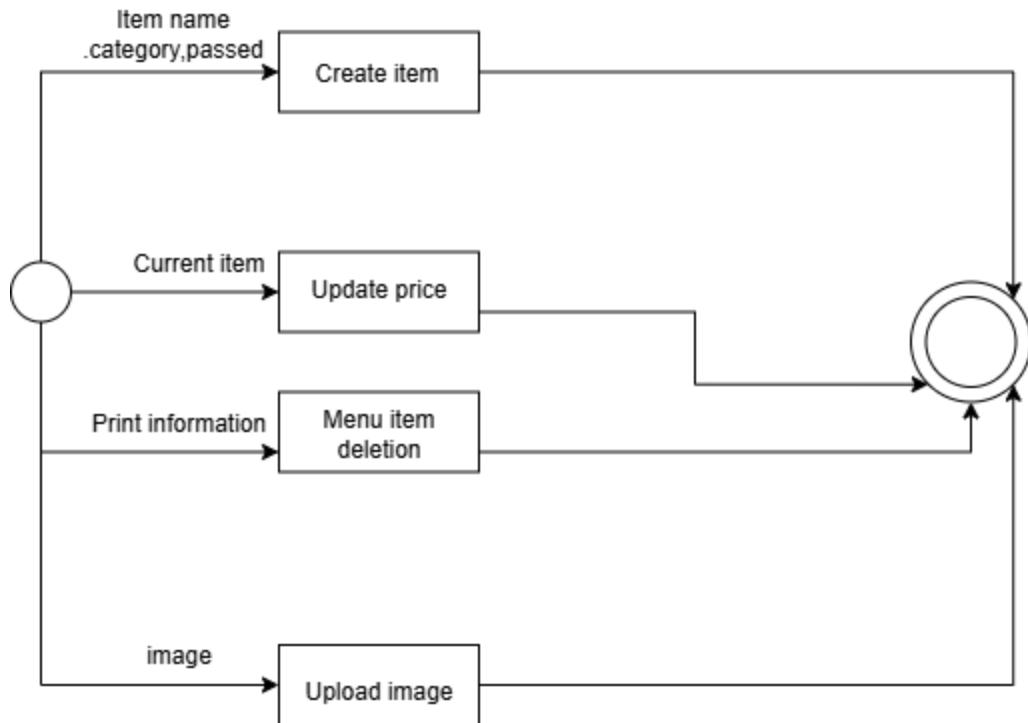


The Authentication process allows users to log in with their ID and password, after which they may sign out successfully. If a user forgets their password, the system generates an OTP, verifies it, and then allows a password reset. After successful completion of any of these processes, the authentication cycle is considered complete.



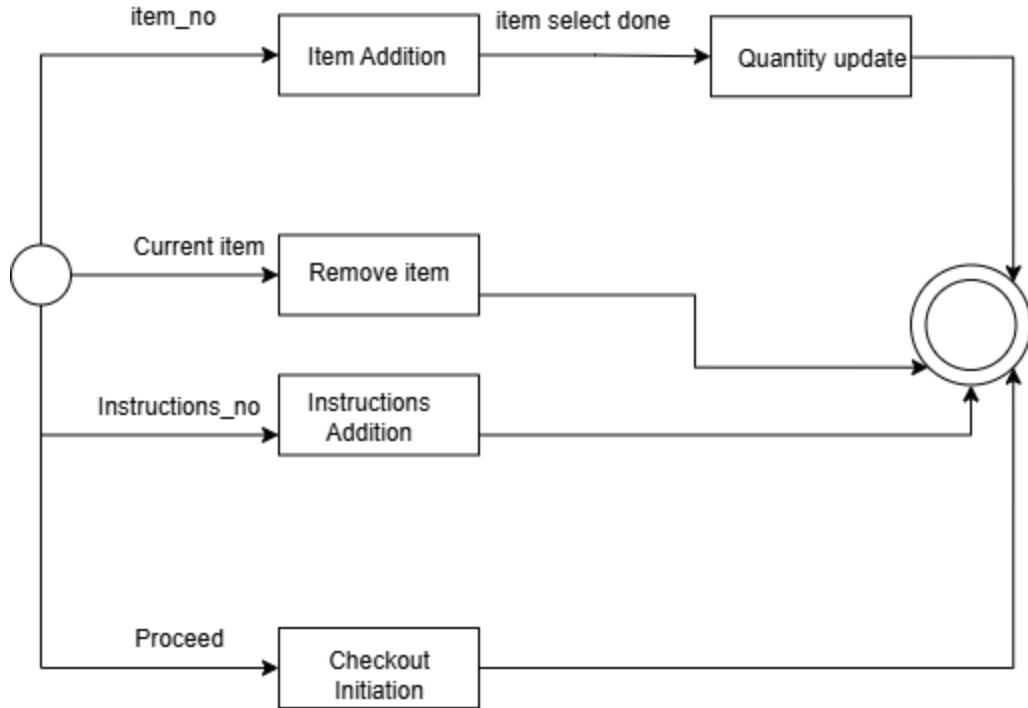
The Menu module enables users or admins to browse items by menu ID or categories. Items can be searched, newly created, edited, or deleted based on current availability. Whenever changes are made, the inventory is updated accordingly. The process concludes once menu items are successfully created, updated, or removed.

ID: 04  
Name : Food Item



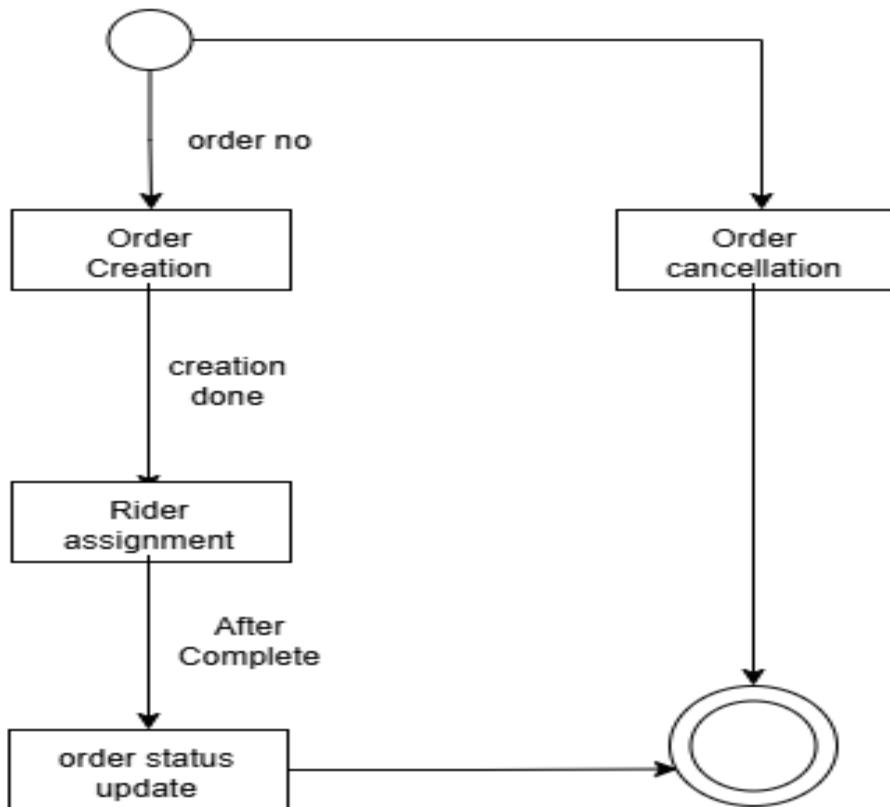
Food items are managed by creating new entries with names, categories, and other details. Existing items can have their prices updated, be deleted, or have images uploaded for better presentation. These actions ensure food items remain current, accurate, and visually appealing for the menu.

ID: 04  
Name : Cart



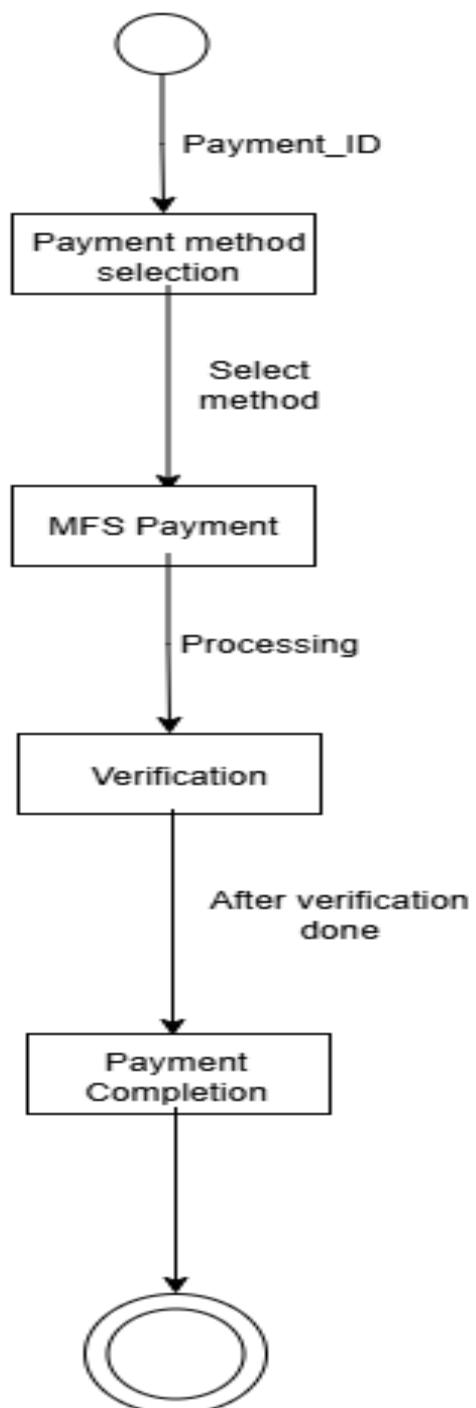
The Cart module allows users to add items, update their quantities, or remove them when no longer needed. Special instructions can be added to specific items, and once satisfied, users can proceed to initiate checkout. The cart's functions end after any of these actions is completed.

ID: 05  
Name : Order



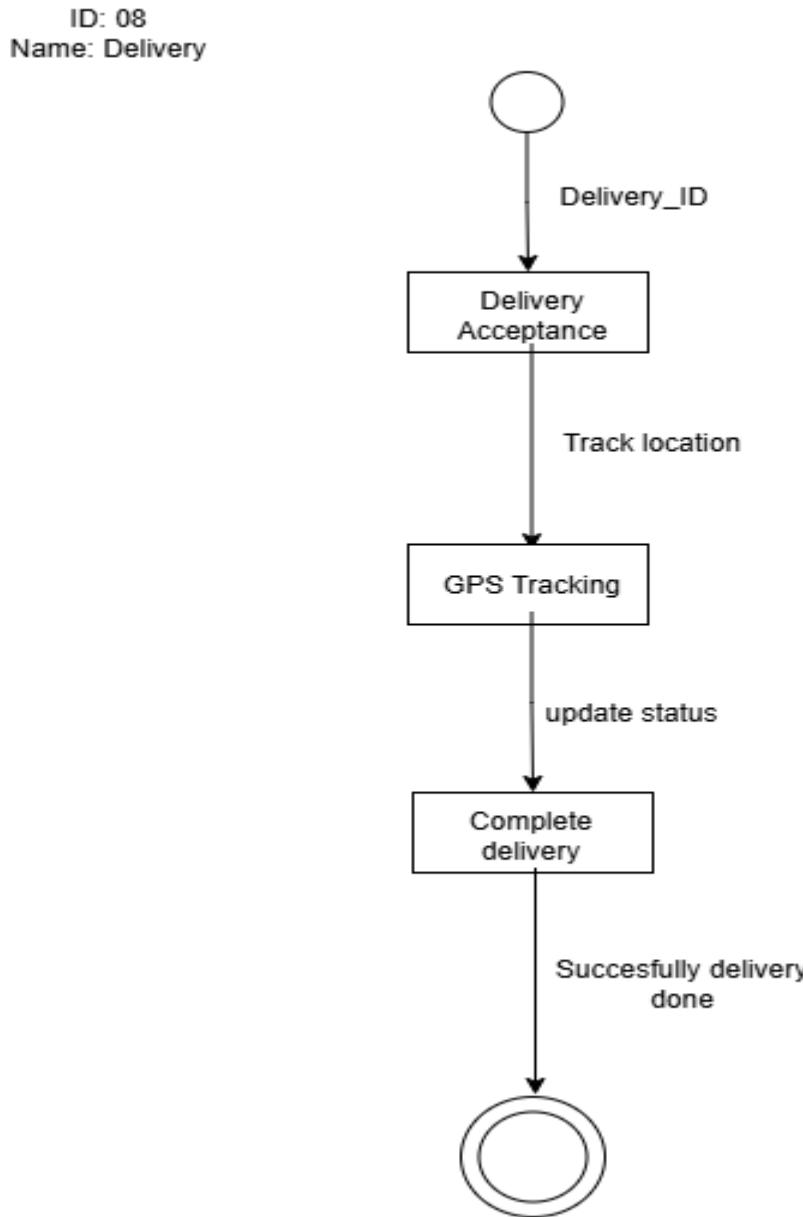
An order begins with its creation and generation of an order number. Once created, a rider is assigned to handle the delivery. The order status is updated after completion, while users retain the option to cancel at any stage. This ensures flexibility in managing customer requests.

ID:07  
Name : payment



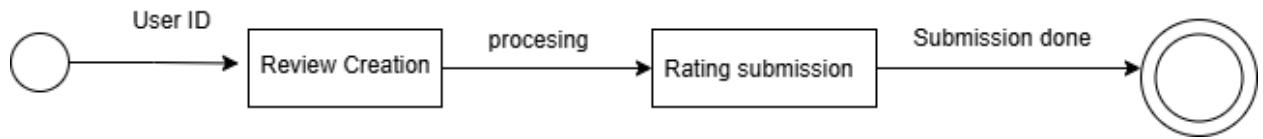
The Payment process starts with a payment ID and involves selecting a preferred method. Mobile Financial Services (MFS) handle processing, followed by

verification to ensure accuracy. Once verified, the transaction concludes with payment completion, marking the process as successful.



The Delivery module begins with acceptance of the task using a delivery ID. The rider's movement is tracked via GPS, and statuses are updated in real time. Upon successful handover to the customer, the delivery is marked complete, closing the process.

ID : 09  
Name : Feedback



The Feedback process starts with the user creating a review linked to their user ID. After the review, they submit a rating, which is processed by the system. Once submitted, the feedback cycle is complete, enabling continuous improvement through customer opinions.

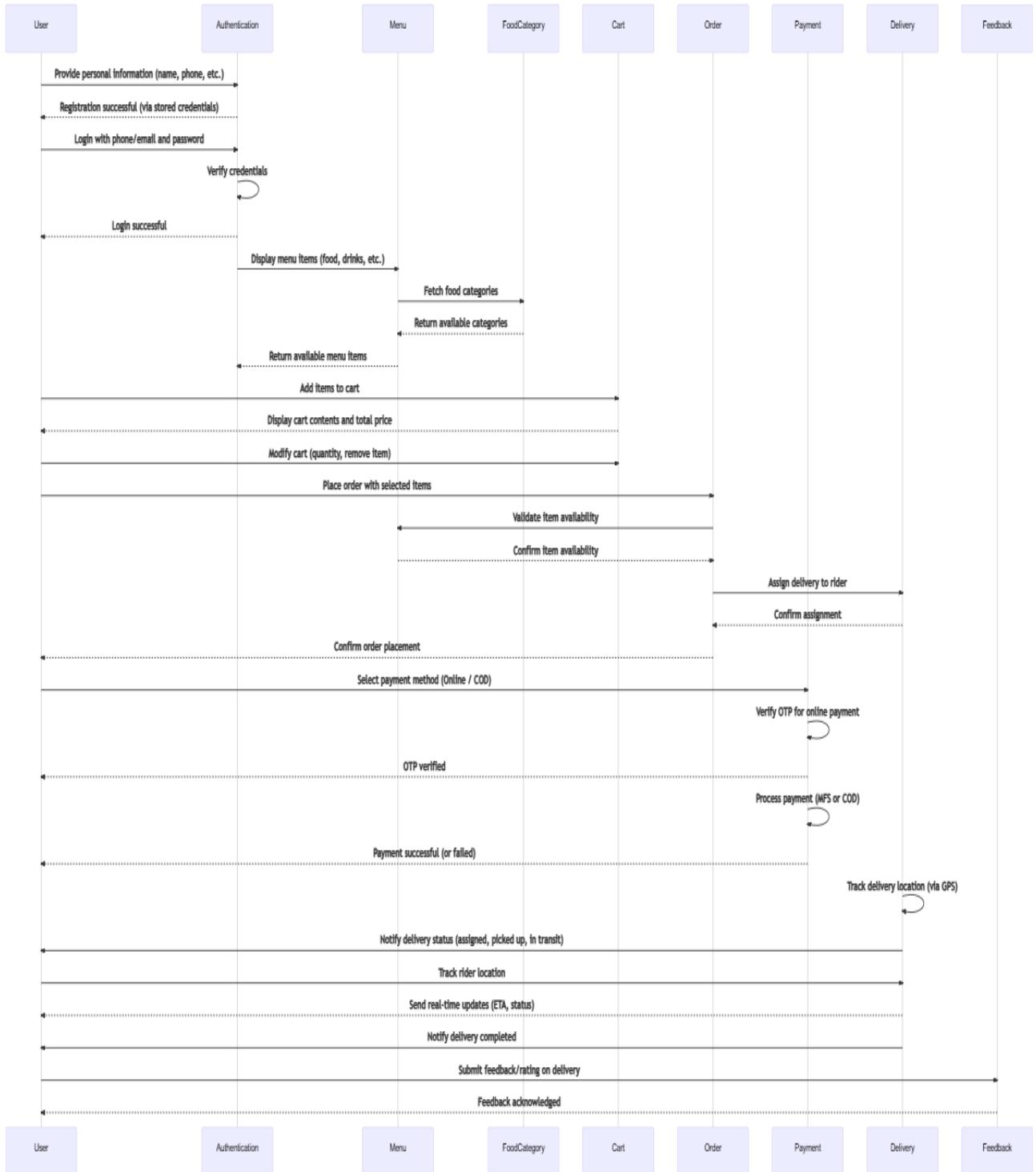
## Sequence Diagram

A Sequence Diagram is a type of UML behavioral diagram that shows the interaction between objects or actors in a system arranged in a time sequence, focusing on the order of messages exchanged to carry out a specific functionality.

### Importance of Sequence Diagram

1. **Visualizes interactions** – Shows how objects/actors communicate step by step.
2. **Clarifies system behavior** – Explains the exact order of operations.
3. **Helps in design** – Guides developers on how to implement the logic.
4. **Detects missing or wrong logic** – Makes errors in flow easy to identify.

5. **Aids in testing** – Testers can create test cases from the message flow.
6. **Simplifies communication** – Easy for both technical and non-technical people to understand.
7. **Handles complexity** – Useful in large systems where many components interact (e.g., online payment, ATM).



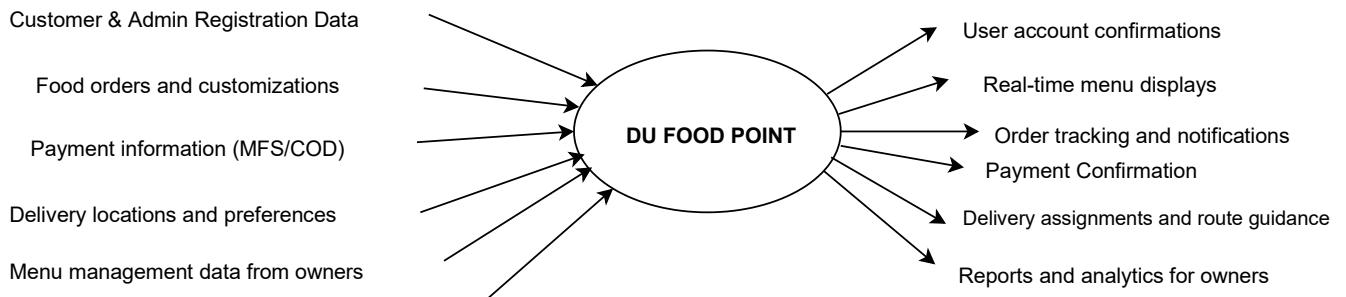
## Data Flow Diagram (DFD)

A Data Flow Diagram (DFD) is a diagram that illustrates the flow of information within a system by showing how input data is transformed into output data through processes, data stores, and data flows.

### **Reasons why DFD is needed:**

1. **Visualize the system** – It gives a clear picture of how data flows, making the system easy to understand for both technical and non-technical people.
2. **Identify requirements** – Helps in analyzing what inputs, outputs, processes, and data stores are necessary.
3. **Improve communication** – Acts as a common language between developers, stakeholders, and clients.
4. **Detect problems early** – By mapping data flow, errors like missing inputs/outputs or redundant processes can be spotted before coding.
5. **Break down complexity** – Using levels (0, 1, 2, etc.), complex systems are divided into smaller, manageable parts.
6. **Documentation** – Serves as part of system documentation for future maintenance and upgrades.

### **Level 0: DU Food Point Management System**



Rider status updates and GPS location

## Central Process

**DU FOOD POINT** - This is the main system being analyzed, represented by the central oval/bubble.

### Input Data Flows (Into the System)

The arrows pointing INTO the system show data that feeds into DU FOOD POINT:

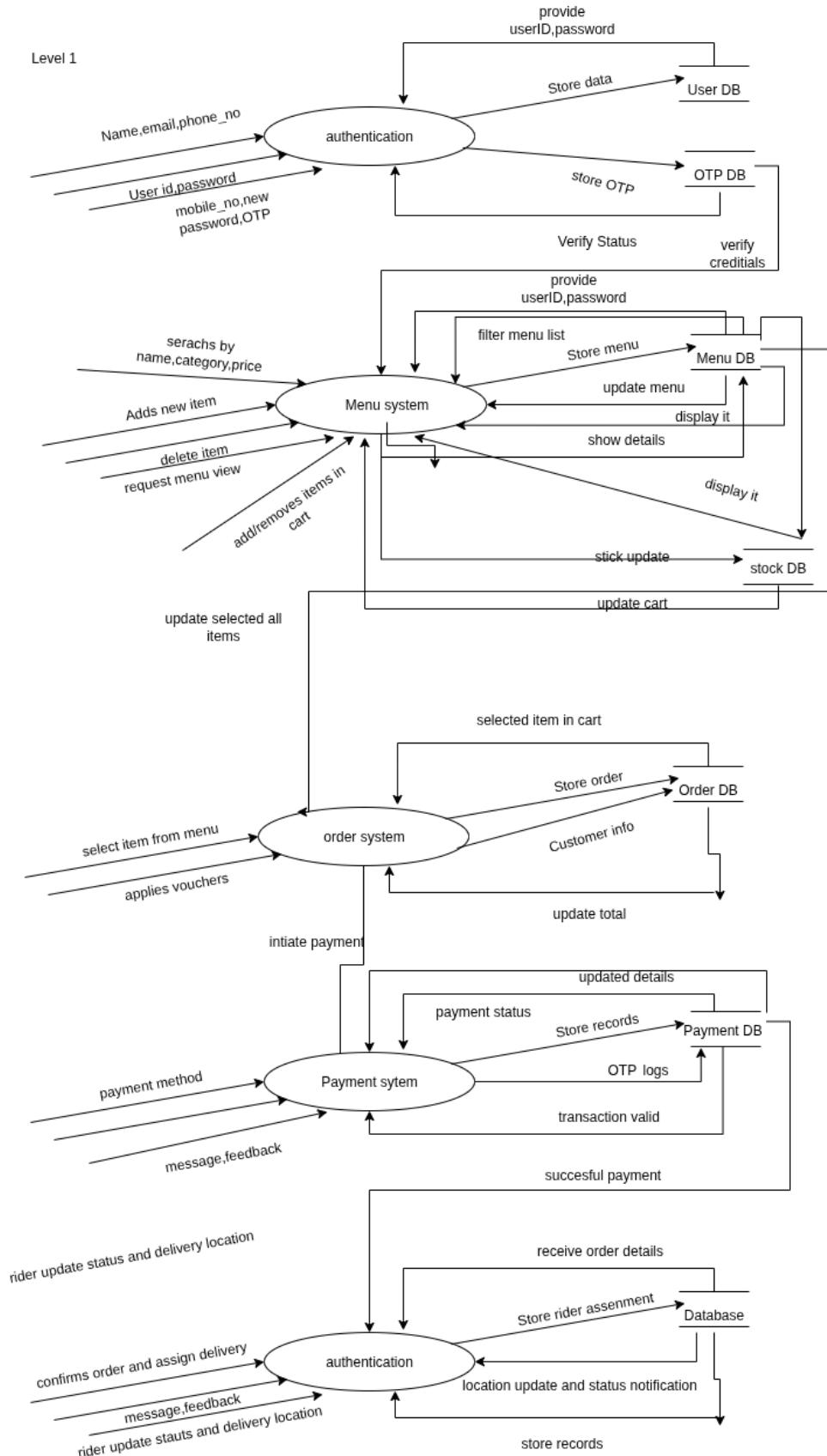
- **Customer & Admin Registration Data** - User sign-up information
- **Food orders and customizations** - Order details and special requests
- **Payment information (MFS/COD)** - Mobile Financial Services or Cash on Delivery payment data
- **Delivery locations and preferences** - Address and delivery specifics
- **Menu management data from owners** - Restaurant/food provider menu updates
- **Rider status updates and GPS location** - Delivery personnel tracking information

### Output Data Flows (From the System)

The arrows pointing OUT OF the system show what the system produces:

- **User account confirmations** - Registration confirmations to users
- **Real-time menu displays** - Updated menu information for customers
- **Order tracking and notifications** - Status updates for customers
- **Payment Confirmation** - Transaction completion notices
- **Delivery assignments and route guidance** - Instructions for delivery riders
- **Reports and analytics for owners** - Business insights for restaurant partners





The authentication process receives inputs of registration data (name, email, phone), login credentials (user ID, password), and recovery information (mobile, new password, OTP), then outputs stored user data to User DB, OTP codes to OTP DB, verification status, and authenticated credentials to enable system access.

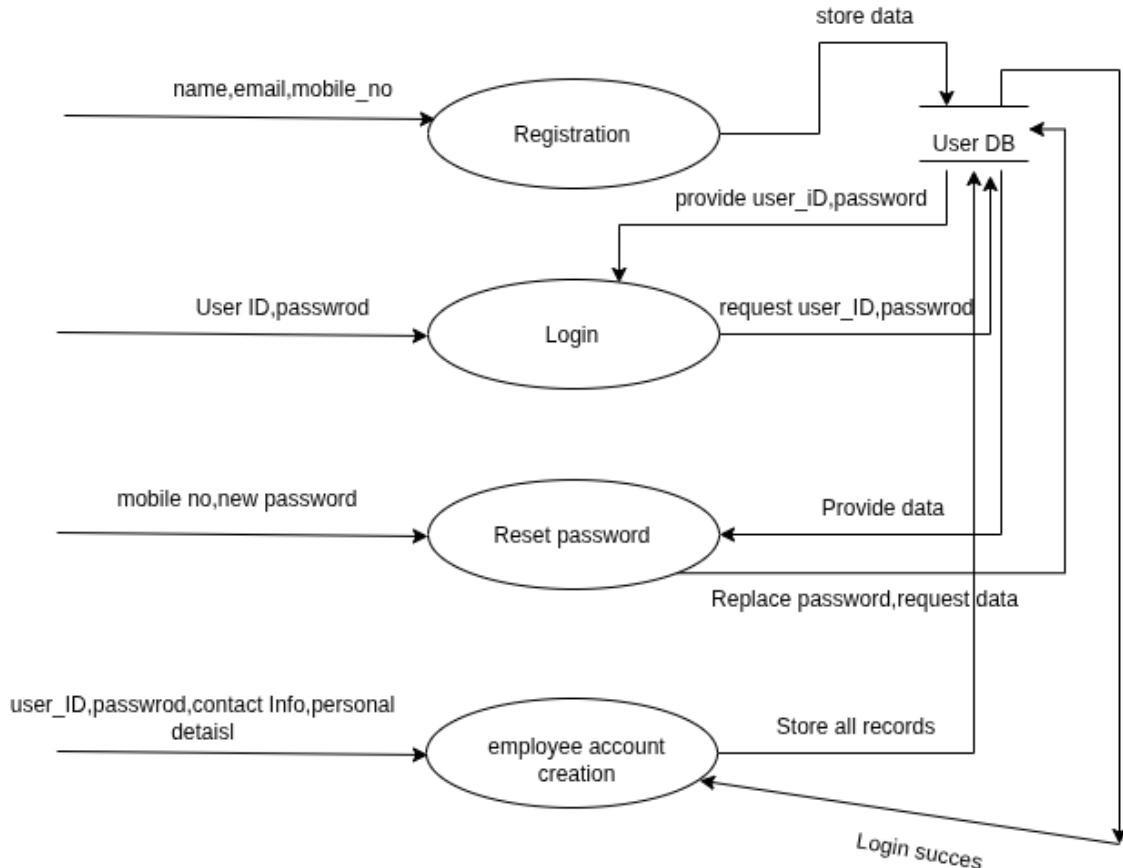
The menu system process accepts inputs including search queries by name/category/price, menu management requests from owners, browsing requests, and cart selections, then generates outputs that store menu information in Menu DB, display filtered menus, send stock updates to Stock DB, and update cart contents for customer selections.

The order system process receives inputs of selected menu items, applied vouchers, and cart contents, then produces outputs including stored orders in Order DB, customer information for payment, calculated totals, and finalized selections for transaction processing.

The payment system process accepts inputs of payment methods, gateway feedback, and payment initiation data, then generates outputs that store transaction records in Payment DB, maintain security logs, provide validity confirmations, and issue successful payment triggers for delivery.

The delivery authentication process receives inputs of rider location updates, order assignments, communication messages, and order details from payments, then produces outputs including rider assignment records in Database, location notifications for customers, and delivery documentation, completing the entire food ordering and delivery workflow from authentication through final delivery tracking.

level 1.1  
Authentication



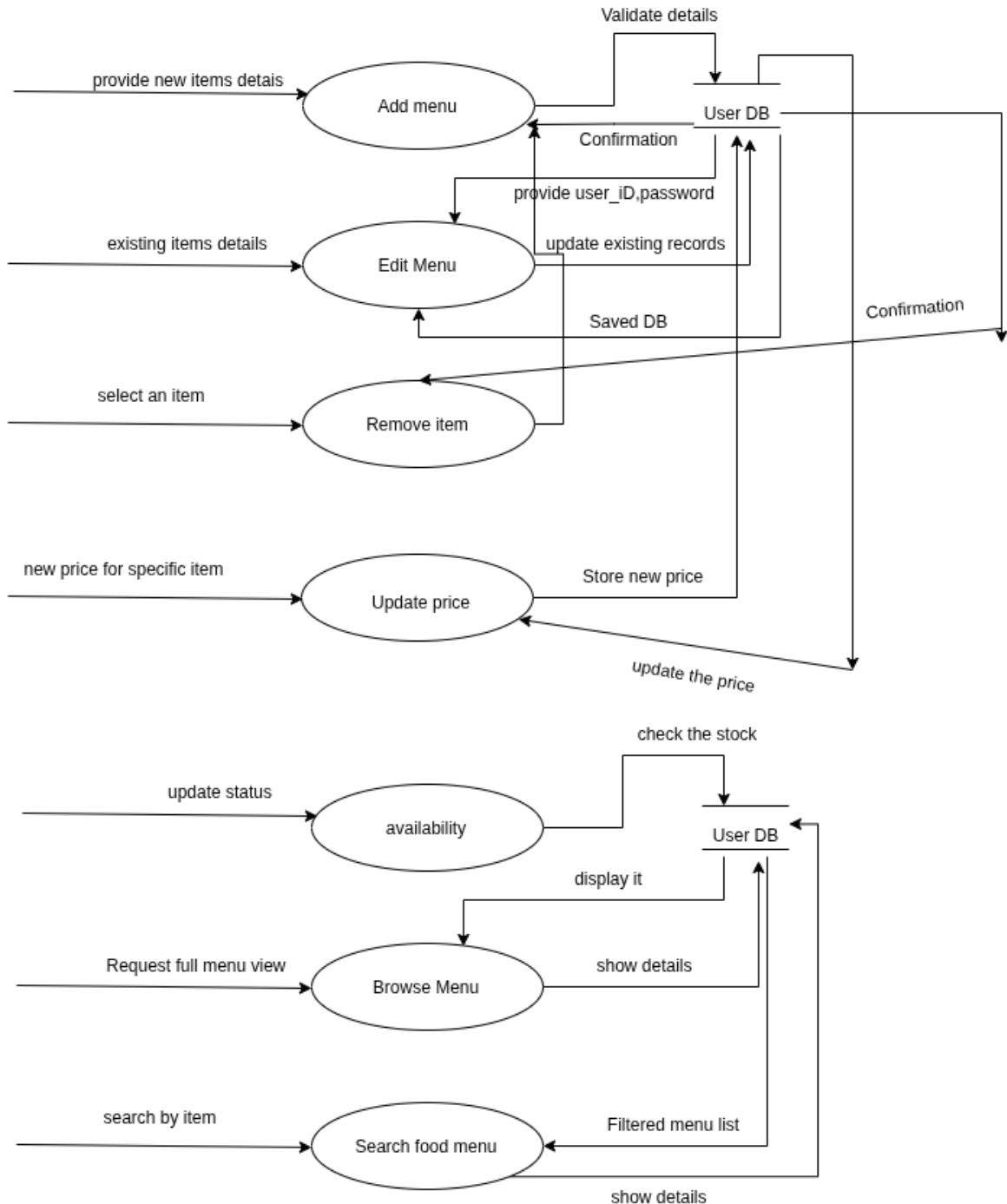
The Registration process initiates user account creation by accepting inputs of name, email, and mobile number from new users, then outputs stored data to the User DB for permanent record keeping and provides user ID and password credentials back to users, establishing their initial system access capabilities.

The Login process handles returning user authentication by receiving inputs of existing user ID and password combinations, then requesting verification data from the User DB to validate credentials and grant authorized system access for established users.

The Reset password process manages account recovery scenarios by accepting inputs of mobile number and new password from users who need credential updates, then interacting with the User DB to provide verification data and outputting replaced password information along with request data to update existing account records.

The Employee account creation process completes the authentication cycle by receiving comprehensive inputs including user ID, password, contact information, and personal details for staff members, then storing all records in the User DB and providing login success confirmation to grant employee access to system administration features. Throughout all steps, the central User DB serves as the primary data repository, maintaining bidirectional communication with each process to ensure secure credential storage, verification, and management across the entire authentication system.

Level 1.2  
Menu system



The Add menu process initiates menu expansion by accepting inputs of new item details from restaurant owners, then outputs validation details and confirmation to

the User DB while providing user ID and password verification, establishing secure menu addition capabilities with proper authentication.

The Edit Menu process manages existing item modifications by receiving inputs of existing item details for updates, then interacting with the User DB to provide user ID and password verification and outputting updated existing records along with confirmation to the Saved DB, ensuring authorized menu changes are properly documented.

The Remove item process handles menu item deletion by accepting inputs to select specific items for removal, then coordinating with the Saved DB and User DB to maintain data integrity and provide confirmation of successful item removal from the menu system.

The Update price process manages pricing modifications by receiving inputs of new price information for specific items, then storing new price data and coordinating price updates across the system to maintain current pricing information.

The Availability process controls item availability status by accepting update status inputs, then checking stock levels in the User DB and displaying current availability information to ensure customers see only available menu items.

The Browse Menu process facilitates customer menu exploration by receiving requests for full menu views, then showing detailed menu information from the User DB to provide comprehensive browsing capabilities for customers.

The Search food menu process completes the menu system by accepting search inputs by specific items, then generating filtered menu lists and showing relevant details to help customers quickly find desired food items through targeted search functionality.