



Food Delivery System

Software Specification and Requirements Analysis

Supervised By

Dr. Sumon Ahmed
Associate Professor
IIT, DU

Submitted By

Md. Jihad Hossain
Roll:BSSE 1413
Md.Main Uddin
Roll:BSSE 1472

2. Quality Function Deployment

Quality Function Deployment (QFD) is a method that translates customer needs into technical requirements for the software. For the **Online Food Delivery System** (one restaurant), the following requirements are identified:

Normal Requirements:

User Interface (UI) & Functionalities

- Users should be able to browse the restaurant's menu.
- A cart system should allow users to add/remove items.
- Users should be able to place an order with a delivery address.
- Users can choose different payment methods (credit card, cash, etc.).
- Order confirmation and tracking should be provided to users.
- A basic feedback system should be available after delivery.

Expected Requirements:

- The website should handle multiple simultaneous users smoothly.
- Users should have the option to save their favorite orders for quick future checkout.
- Integration with maps for order tracking in real-time.
- User login and account creation features, with social login options.

Exciting Requirements:

- **Personalized Offers & Recommendations:** Personalized food suggestions based on order history.
- **Loyalty Program:** Points and rewards for repeat customers that can be redeemed for discounts.
- **Push Notifications:** Real-time alerts for order status, promotions, and special offers.

2.1 Usage Scenario

Scenario: Online Food Delivery for One Restaurant

Actors:

1. **Customer** - The individual using the platform to browse and order food.
2. **Restaurant System(Admin)** - The restaurant's online menu and order management system.
3. **Delivery Personnel** - The person responsible for picking up and delivering the order.
4. **Payment Gateway** - The service managing the payment transaction.

Preconditions:

- The **Online Food Delivery** application or website is accessible to the customer.
- The restaurant's **menu and availability** are updated in real-time on the platform.
- The **Customer** is registered and logged in on the application.
- The **Payment Gateway** is functional and ready to process transactions.
- Delivery personnel is available for order fulfillment.

Main Scenario:

1. **Customer Opens the Application**
 - The customer launches the food delivery app on their device.
2. **Browsing the Menu**
 - The customer navigates through the restaurant's menu, viewing available items, descriptions, and prices.
3. **Adding Items to Cart**
 - The customer selects desired items and adds them to the shopping cart, specifying quantities and any special requests.
4. **Reviewing Order in Cart**
 - The customer reviews the selected items, checks prices, and makes any last-minute changes to the cart.
5. **Proceeding to Checkout**
 - Once satisfied, the customer proceeds to the checkout page to confirm order details and enter delivery information.

6. Choosing Payment Method

- The customer selects a payment method (e.g., credit card, digital wallet) and confirms the payment via the payment gateway.

7. Order Confirmation

- The system confirms the order, and an estimated delivery time is provided. The customer receives a digital receipt and order tracking ID.

8. Order Preparation by Restaurant

- The restaurant system receives the order and initiates preparation. The order status updates to “In Preparation” on the customer’s app.

9. Assigning Delivery Personnel

- Once the order is ready, the system assigns available delivery personnel, updating the customer on the driver’s location and estimated arrival time.

10. Delivery to Customer

- The delivery personnel picks up the order and transports it to the customer. Real-time tracking is provided in the app.
- Upon delivery, the customer confirms receipt, and the order status updates to “Delivered.”

11. Post-Delivery Feedback

- The customer has the option to rate the experience, provide feedback, and leave a tip if desired.

12. Authentication:

- The Admin/manager will be pre-registered to the system. Every customer have to register first to interact with the system with username,password,phone-number ,email etc.The delivery personnel is added by the admin. Every user can log in the system by their username and password.

Postconditions:

1. **Order Status is Updated** - The order status is marked as "Delivered" in the system, and the transaction is recorded.
2. **Customer Feedback is Saved** - If the customer provides feedback or ratings, these are stored in the system for future reference.
3. **Payment Settlement** - The payment gateway processes the transaction, and the funds are directed to the restaurant after any applicable deductions.
4. **Delivery Personnel Status is Updated** - The delivery personnel's availability status is set to "Available" after successful delivery.

Exceptions:

1. Payment Failure

- **Cause:** Insufficient funds, network issues, or payment gateway errors.
- **Resolution:** The system notifies the customer of the payment failure and prompts them to try another payment method or retry the transaction.

2. Item Unavailability

- **Cause:** Items in the cart are out of stock or unavailable after the customer has added them.
- **Resolution:** The system notifies the customer about the unavailability and prompts them to either remove or replace the unavailable items.

3. Order Cancellation by Restaurant

- **Cause:** The restaurant may need to cancel due to unforeseen circumstances (e.g., kitchen issues, staff shortage).
- **Resolution:** The system notifies the customer, and a refund is initiated if payment was completed. The customer can be given an option to reorder later or choose an alternative.

4. Delivery Personnel Unavailability

- **Cause:** Lack of available delivery personnel due to high demand or other issues.
- **Resolution:** The system notifies the customer of the delay and provides an updated delivery time. If the wait is too long, the customer can opt to cancel the order and receive a refund.

5. Order Delivery Failure

- **Cause:** Delivery personnel unable to locate the customer, or the customer is unavailable to receive the order.
- **Resolution:** The delivery personnel contacts the customer to arrange redelivery. If still unsuccessful, the order is marked as "Undelivered," and the restaurant or platform handles the next steps based on policy (e.g., partial refund or redelivery).

6. App or System Crash

- **Cause:** The application or system experiences a crash or outage.
- **Resolution:** The customer is informed of the issue if possible, and the order status is saved when the system resumes. If an order is disrupted, customer support may assist in re-confirming or canceling it.

7. Network Connectivity Issues

- **Cause:** Customer or delivery personnel loses network connection during ordering or delivery tracking.
- **Resolution:** The system retries the operation if possible. If the customer is disconnected during payment, they may need to reinitiate the transaction once connectivity is restored.

3. Scenario Based Modeling

The success of a computer-based system or product is measured in many ways but user satisfaction resides at the top of the list. Understanding how actors want to interact with a system will lead the software team to properly characterize requirements and build meaningful analysis and design models. Hence, requirements modeling begins with the creation of scenarios in the form of use cases, activity diagrams, and swimlane diagrams.

What is a Use Case Diagram?

A use case is a written description of how users will perform tasks on this website. It outlines, from a user's point of view, a system's behavior as it responds to a request. Each use case is represented as a sequence of simple steps, beginning with a user's goal and ending when that goal is fulfilled. Use cases specify the expected behavior (what), and not the exact method of making it happen (how). Use cases once specified can be denoted both textual and visual representation (i.e. use case diagram). A key concept of use case modeling is that it helps us design a system from the end user's perspective. It is an effective technique for communicating system behavior in the user's terms by specifying all externally visible system behavior.

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:

Name: Food delivery System

Primary Actor: Customer, Admin, Delivery Man

Secondary Actor: Payment Gateway

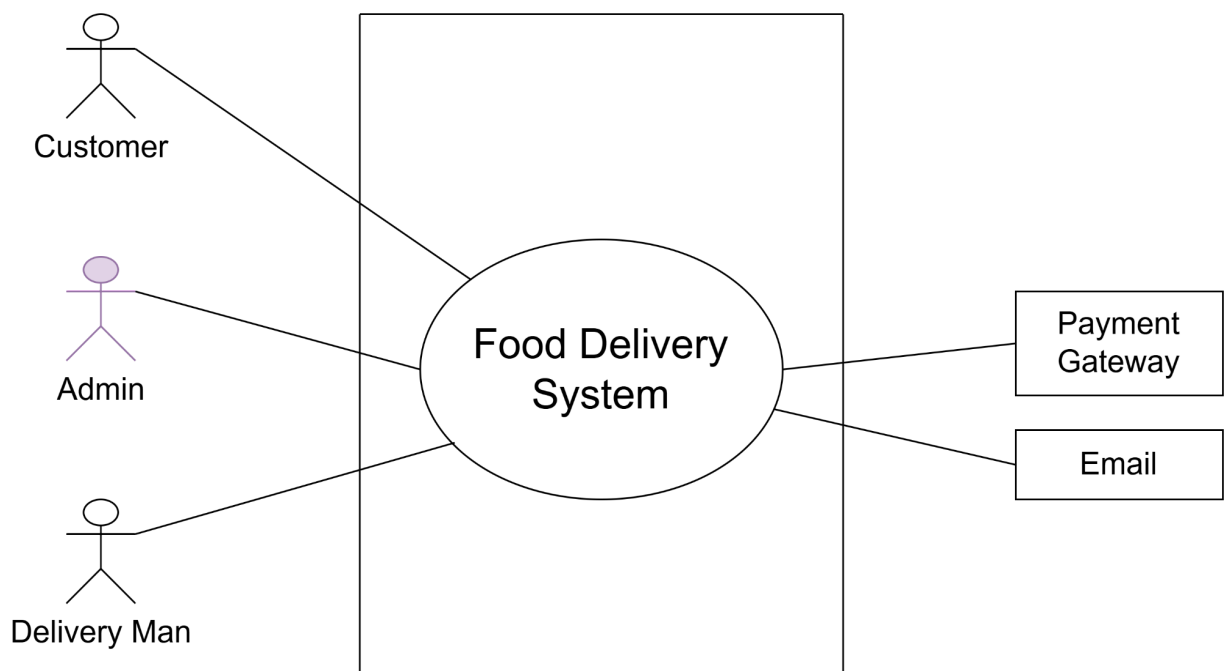


Fig-1: Food Delivery System

Level 1

Name:Details of Food delivery system

Primary Actor:Customer,Admin,Delivery Man

Secondary Actor:Payment Gateway

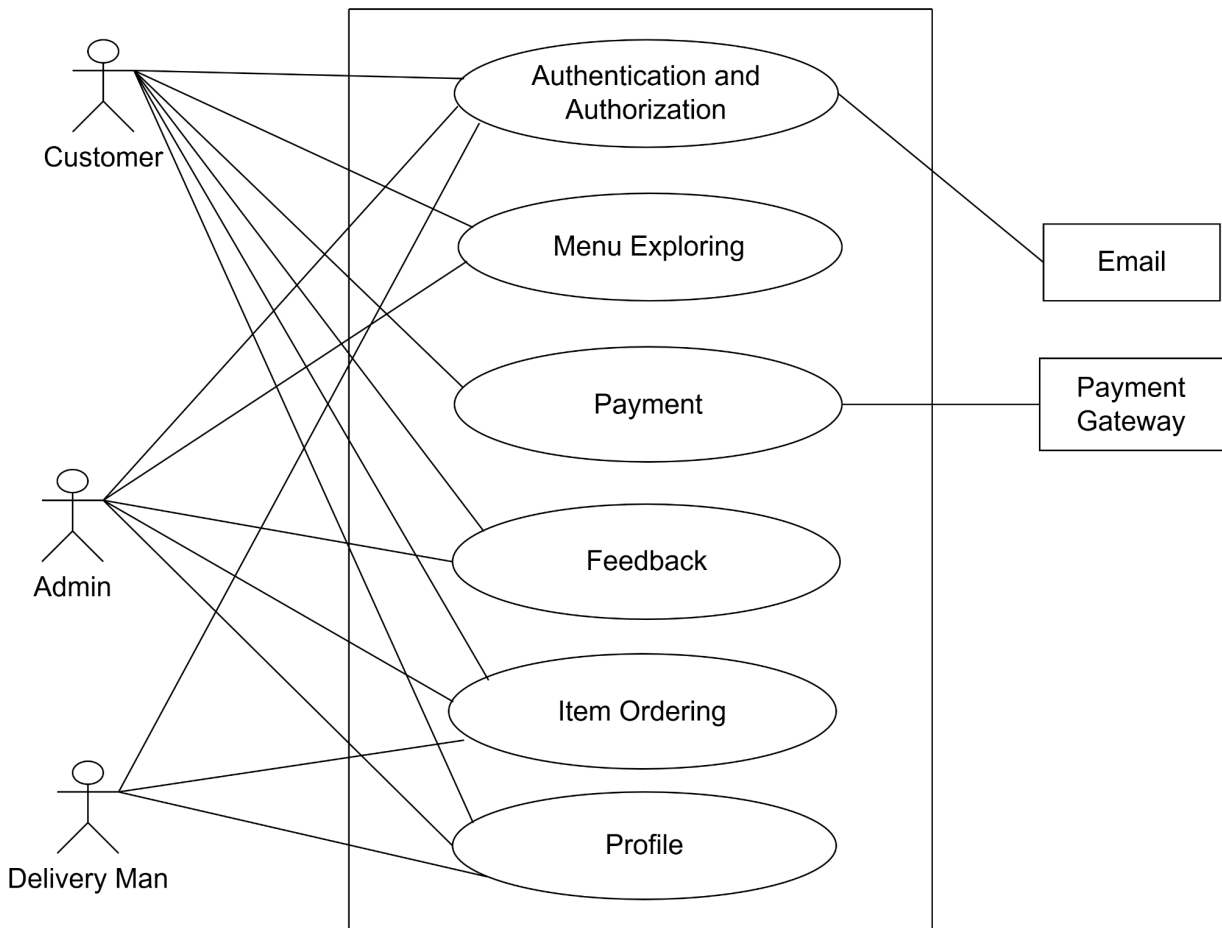


Fig-2:Level 1

Authentication and Authorization

Purpose: To ensure secure access to the system and protect user data.

- **User Registration:** Customers and admins register by providing essential details such as email, password, and name.
- **User Login:** Validates credentials using secure password hashing and grants access.
- **Role-Based Authorization:** Determines permissions for customers (place orders, provide feedback) and admins (manage products, view reports).
- **Forgot Password:** Allows users to reset passwords via email or OTP-based verification.
- **Session Management:** Ensures active sessions are managed securely, with automatic logout for inactivity.

Profile Management

Purpose: To manage user details and enhance their experience.

- **Customer Profile:**
 - View and update personal details (name, address, phone number).
 - Maintain saved addresses for quick order placement.
- **Admin Profile:**
 - Manage credentials and access settings.
- **Data Security:** All profiles are securely stored in the database, ensuring privacy and confidentiality.

Menu Exploring

Purpose: To allow users to browse the restaurant's offerings.

- **Categorized Menu:**
 - Display food items by categories such as appetizers, main courses, desserts, and beverages.
- **Product Details:**
 - Show item name, price, description, and availability.
- **Search Functionality:**
 - Allow users to search for items by name or keywords.
- **Recommendations:**
 - Suggest popular or similar dishes based on user browsing history.

Item Ordering

Purpose: To facilitate smooth and efficient order placement.

- **Add to Cart:**
 - Users can add items to the cart, view the total price, and adjust quantities.
- **Place Order:**
 - Select delivery addresses and payment methods before confirming the order.
- **Order Status:**
 - Provide real-time updates on order status (e.g., confirmed, in preparation, out for delivery).

Payment

Purpose: To enable secure and diverse payment options.

- **Payment Gateway Integration:**
 - Support payment methods such as credit/debit cards, UPI, digital wallets, and cash on delivery.
- **Transaction Details:**
 - Display the total price, including taxes and delivery charges, before payment.
- **Payment Confirmation:**
 - Send receipts via email or SMS after successful transactions.
- **Refund Process:**
 - Handle refund requests for canceled orders or failed transactions.

Feedback

Purpose: To gather customer reviews and improve service quality.

- **Ratings:**
 - Customers can rate products and delivery services on a scale (e.g., 1-5 stars).
- **Comments:**
 - Provide detailed feedback or suggestions for improvement.
- **Feedback Management:**
 - Admins can view and respond to feedback, using it to enhance customer satisfaction.

Action Replies for Actors

1. Customer

- **Action:** Registers for an account.
 - **Reply:** System validates and creates a profile.
- **Action:** Browses menu or places an order.
 - **Reply:** Menu displays, and order details are stored.
- **Action:** Provides feedback.

- **Reply:** Feedback is stored for analysis.

2. Admin

- **Action:** Manages products.
 - **Reply:** Updates reflected in the menu.
- **Action:** Tracks orders.
 - **Reply:** Dashboard updates order statuses.

3. System

- **Action:** Sends notifications for orders.
 - **Reply:** Delivers real-time updates to customers.

4. Delivery Man

Action: Logs into the platform.

Reply:

- The system verifies login credentials provided by the delivery man.
- Grants access to the dashboard where assigned orders, delivery history, and availability status can be managed.

Action: Updates availability status (e.g., Available, Unavailable).

Reply:

- The system updates the delivery man's availability in the database.
- Admins and the order management module are notified of the availability change.

Action: Views assigned orders.

Reply:

- The system retrieves and displays all orders assigned to the delivery man, along with the delivery address, contact details, and estimated delivery time.

Action: Marks order as "Picked Up."

Reply:

- The system updates the order status to "In Transit."
- Real-time order tracking is enabled for the customer, showing the delivery man's location.

Action: Delivers the order to the customer.

Reply:

- The system allows the delivery man to mark the order as "Delivered."
- Updates the order status in the database.
- Notifies the customer and admin about successful delivery.

Action: Reports an issue (e.g., unable to deliver, customer unavailable).

Reply:

- The system logs the issue and notifies the admin for resolution.
 - If necessary, allows the admin to reassign the order or handle escalations.
-

Action: Views earnings or delivery history.

Reply:

- The system retrieves the delivery man's performance metrics, including total deliveries completed, customer ratings, and payment details for completed deliveries.
-

2. Admin Actions and System Replies Related to Delivery Men

Action: Adds new delivery man profiles.

Reply:

- The system validates delivery man details (e.g., name, contact info).
 - Creates a new delivery man profile in the database and sets their initial status as "Available" or "Unavailable."
-

Action: Assigns orders to delivery men.

Reply:

- The system suggests available delivery men based on location, workload, and status.
 - Admin selects a delivery man, and the system assigns the order, updating the delivery man's task list.
 - Notifies the delivery man of the new assignment via their dashboard.
-

Action: Monitors delivery progress.

Reply:

- The system provides real-time tracking and updates for all assigned deliveries.
 - Displays any reported issues for admin intervention.
-

Action: Manages delivery man profiles (e.g., updates, deactivates).

Reply:

- The system allows the admin to edit profile details or deactivate delivery man accounts.
- Updates the database and notifies the affected delivery man of any changes.

Level 1.1

Name: Authentication and Authorization

Primary Actor: Customer, Admin, Delivery Man

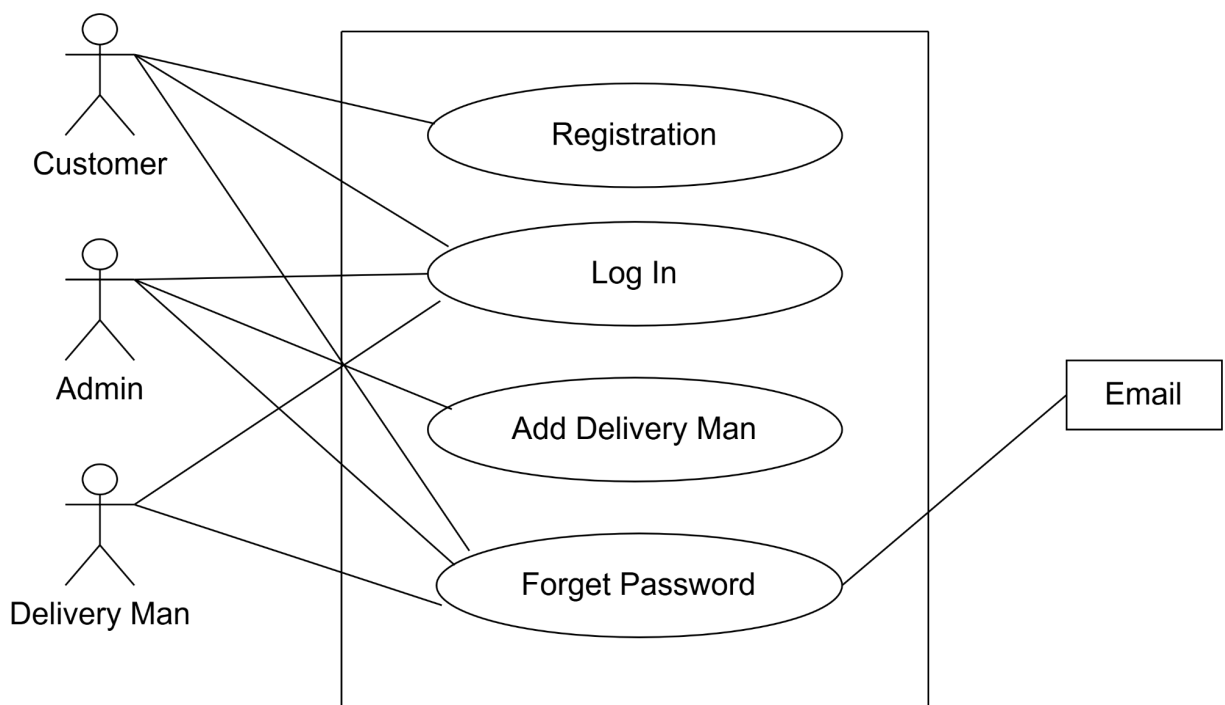


Fig-3: Authentication and Authorization

Customer:

Action1:Request for Registration

Reply1:Get Registration Notification

Action2:Request for Login

Reply2:Get Login Notification

Admin:

Action1:Request for Login

Reply1:Get Login Notification

Action2:Request for Add Delivery Man

Reply2:Get Add Delivery Man Notification

Delivery Man:

Action:Request for Login

Reply:Get Login Notification

Level 1.2

Name:Menu Exploring

Primary Actor:Customer,Admin

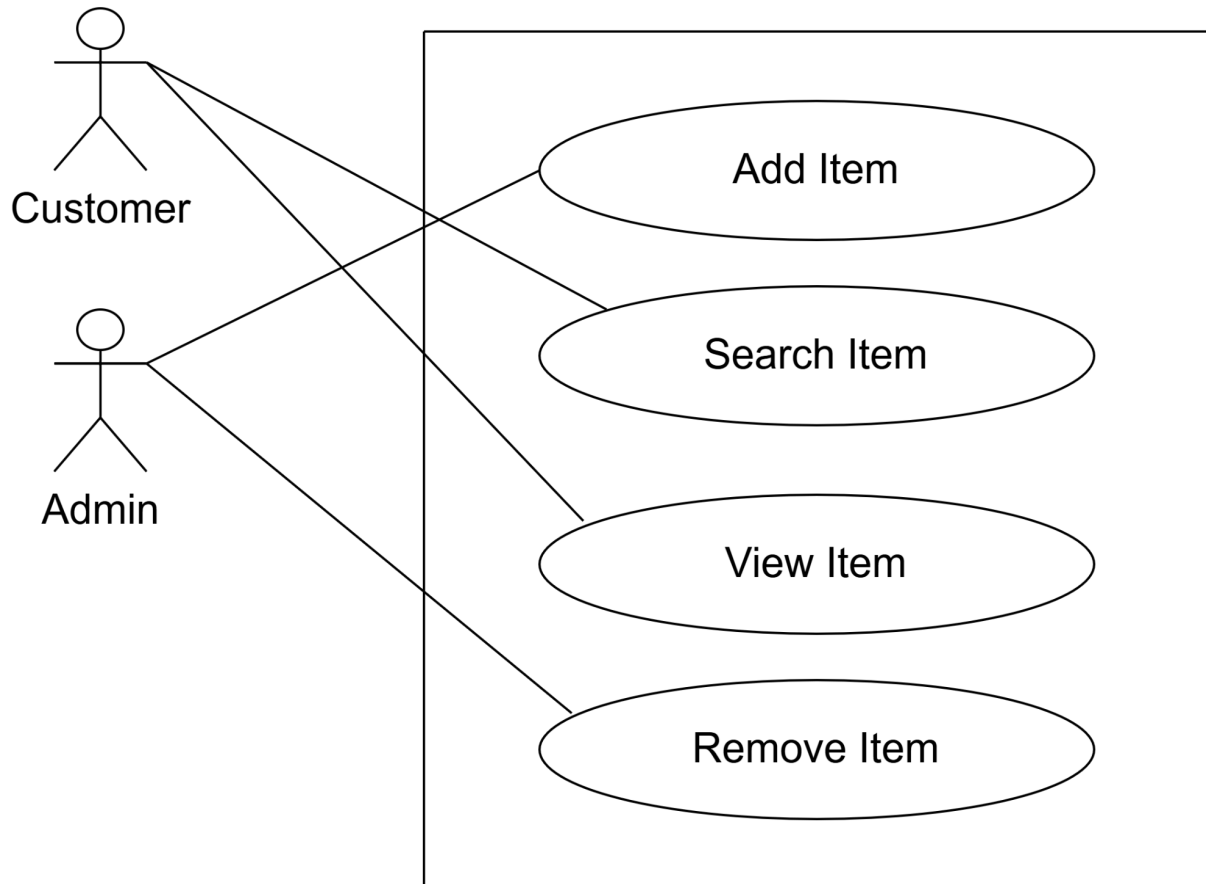


Fig-4:Menu Exploring

Customer:

Action1:Customer request on server for searching an item

Reply1:System provides customer acquire response

Action2:Customer needs to view item

REply2:System provides the essential information

Admin:

Action1:Admin adds item in the system

Reply1:System gives response in terms of the request

Action2:Admin can remove item from the system

Reply2:System gives notification that item has removed from the system successfully

Level 1.3

Name:Payment

Primary Actor:Customer

Secondary Actor:Payment Gateway

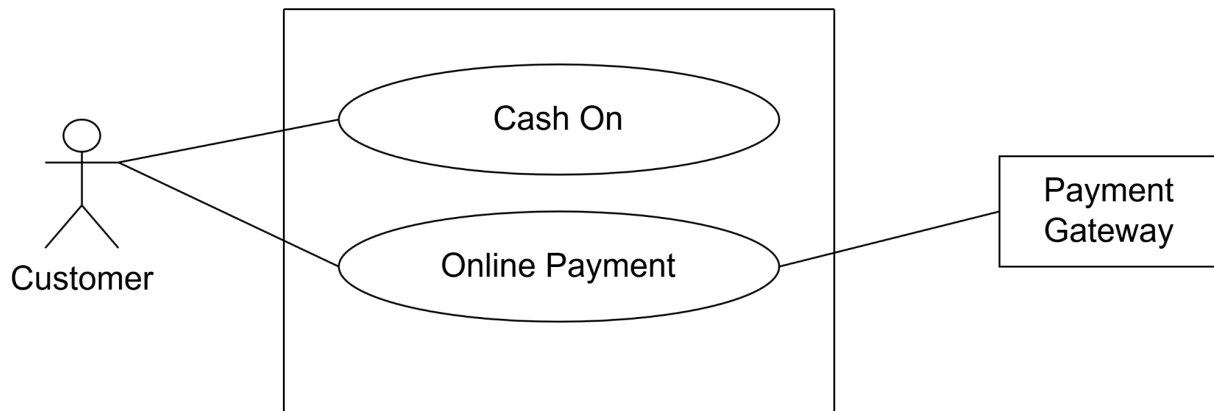


Fig-5:Payment

Customer:

Action1:Customer selects cash on process

Reply1:System gives s a notification

Action2:Customer pays amount via online payment

Reply2:Customer receives a payment notification

Payment Gateway:

Action:It provides an online payment method

Reply:Deliver notification

Level 1.4

Name:Feedback

Primary Actor:Customer,Admin

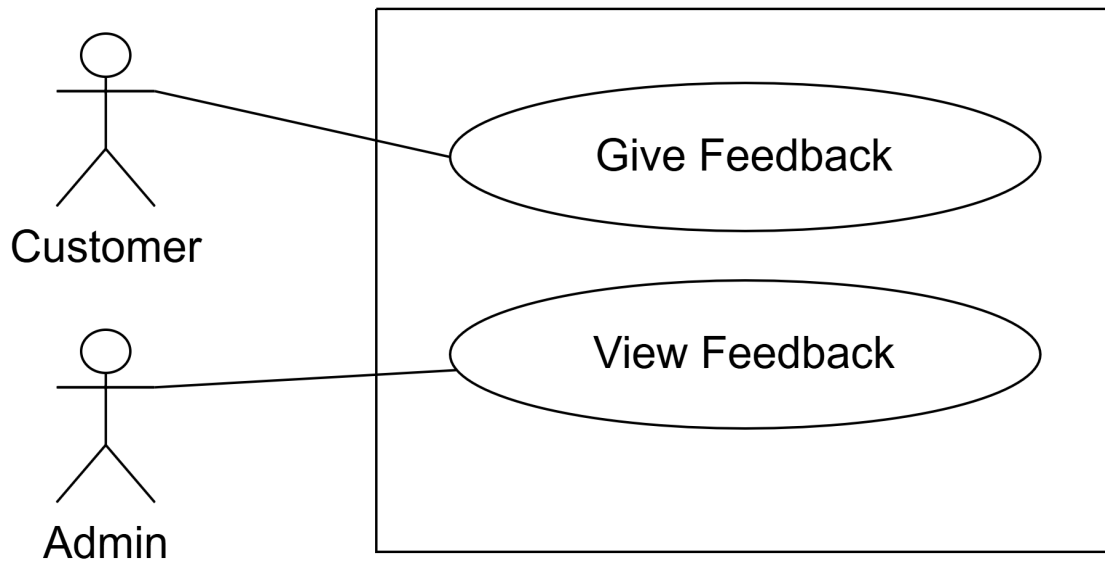


Fig-6:Feedback

Customer:

Action:Customer gives feedback according to the service

Reply:Customer gets the feedback notification

Admin:

Reply:Admin Can show the feedback report from the customer

Level 1.5

Name:Item Ordering

Primary Actor:Customer,Admin, Delivery man

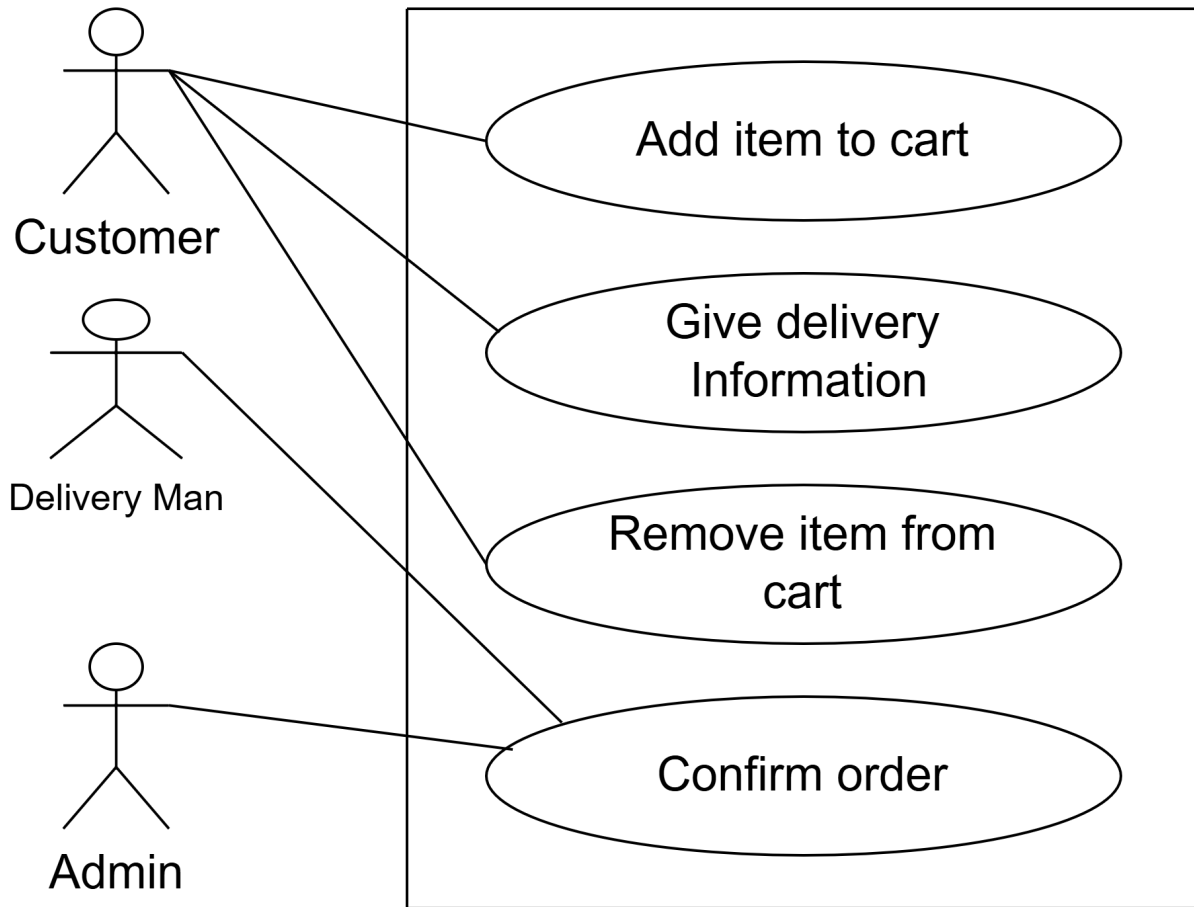


Fig-7:Item Ordering

Level 1.6

Name:Profile

Primary Actor:Customer,Delivery Man

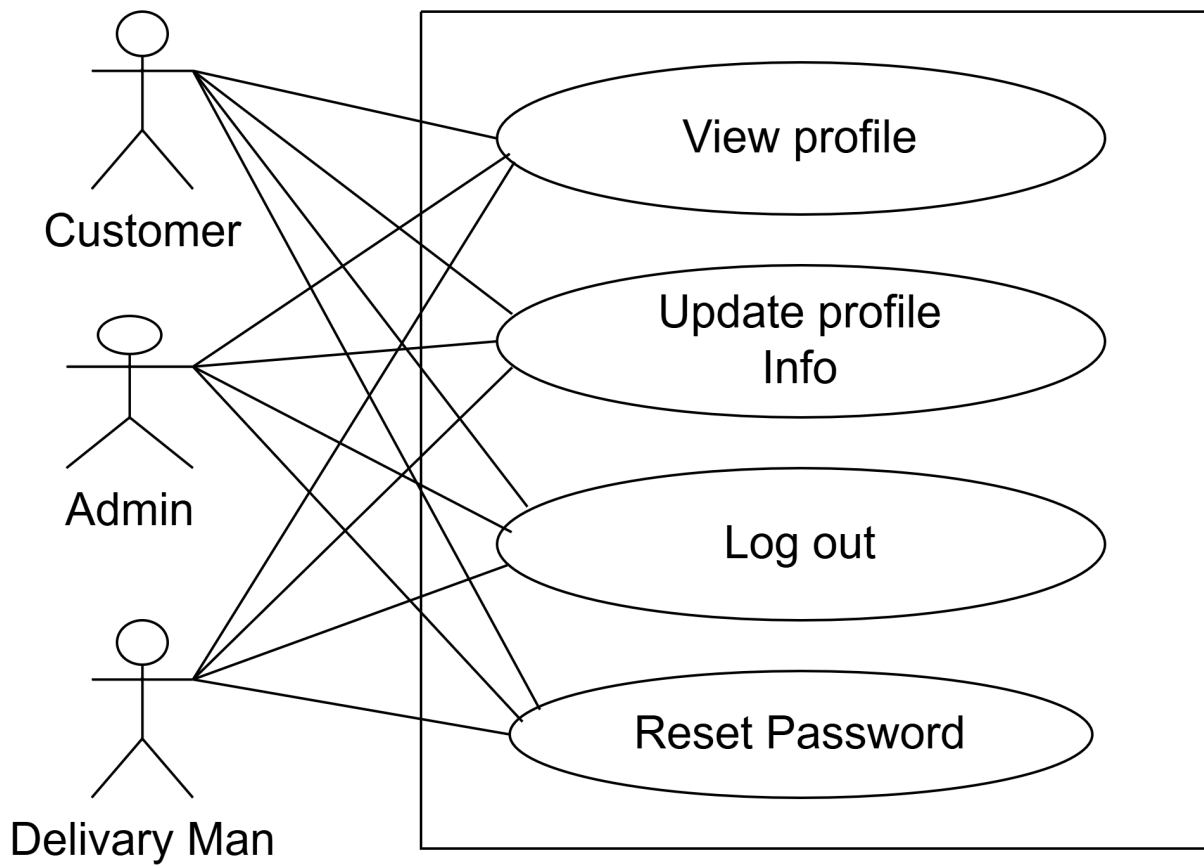


Fig-8:Profile

Level 1.5.4

Name: Confirm Order

Primary Actor: Admin, Delivery Man, Customer

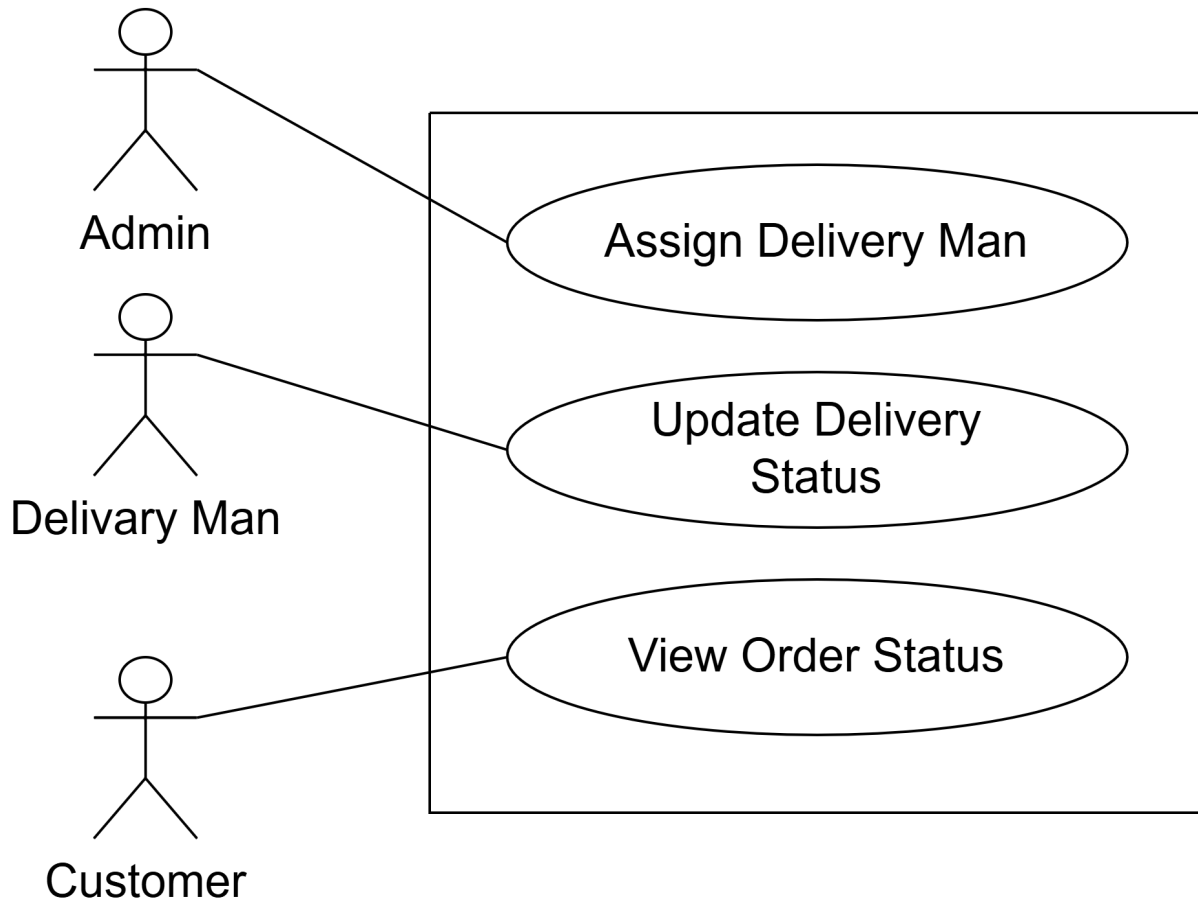


Fig-9: Confirm Order

Customer:

Action: Assign Delivery Man for the order

Delivery Man:

Action: Show the delivery details

Reply: Update delivery details

Customer:

Action: View the delivery

4. Activity diagram

What is an Activity Diagram?

An activity diagram is a type of diagram used in software engineering and requirements engineering to model the flow of activities and actions within a system or a business process. It provides a visual representation of the dynamic aspects of a system, showing how different activities interact and the sequence in which they occur.

The Purpose of Activity Diagram

An activity diagram can prove to be useful to a software engineer for various reasons, such as:

- Drawing the activity flow of a system.
- Describing the sequence from one activity to another.
- Describing the parallel, branched and concurrent flow of the system.

Level: 1
Name: Food Delivery System

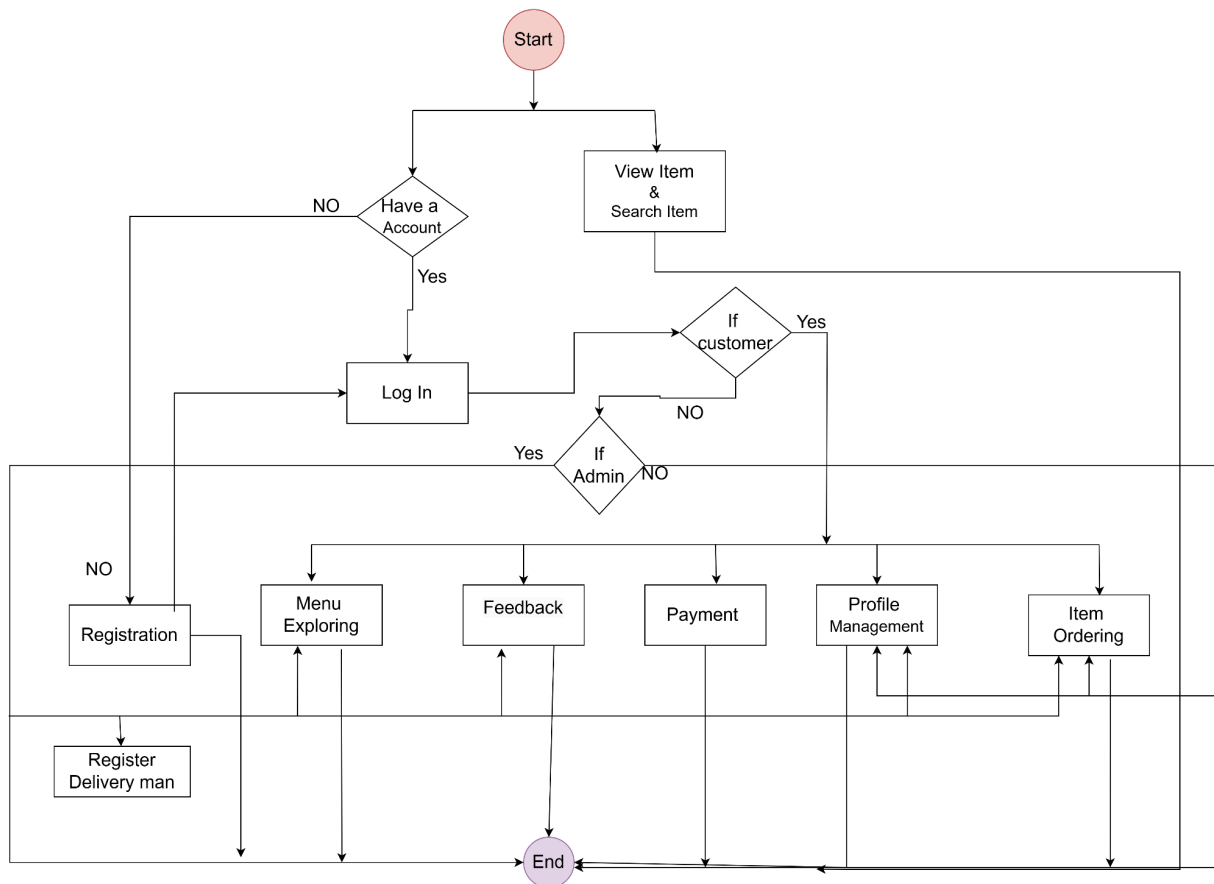


Fig-10: Food Delivery System

Level 1.1

Name: Authentication and Authorization

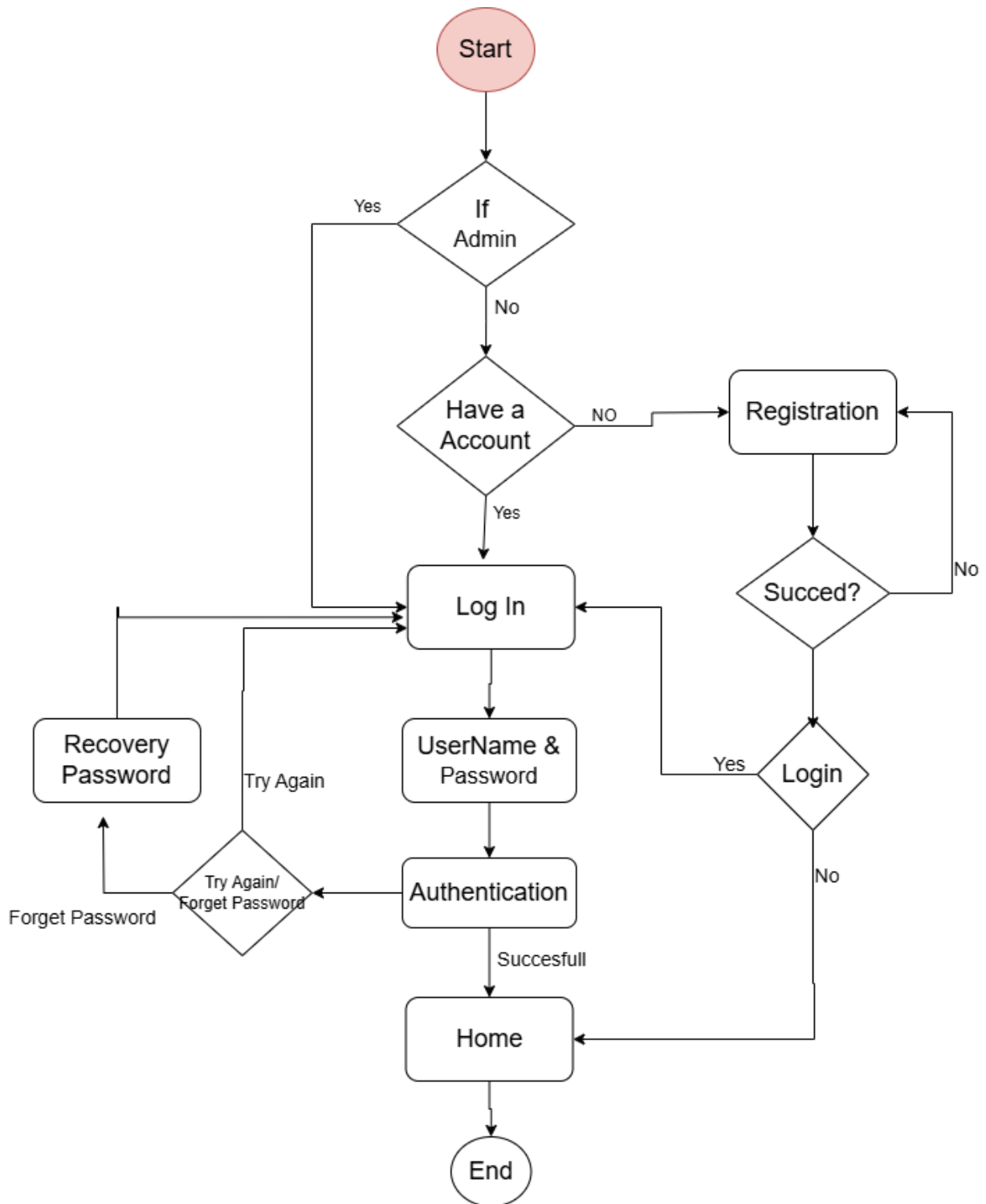


Fig-11: Authentication and Authorization

Level 1.2
Name:Menu Exploring

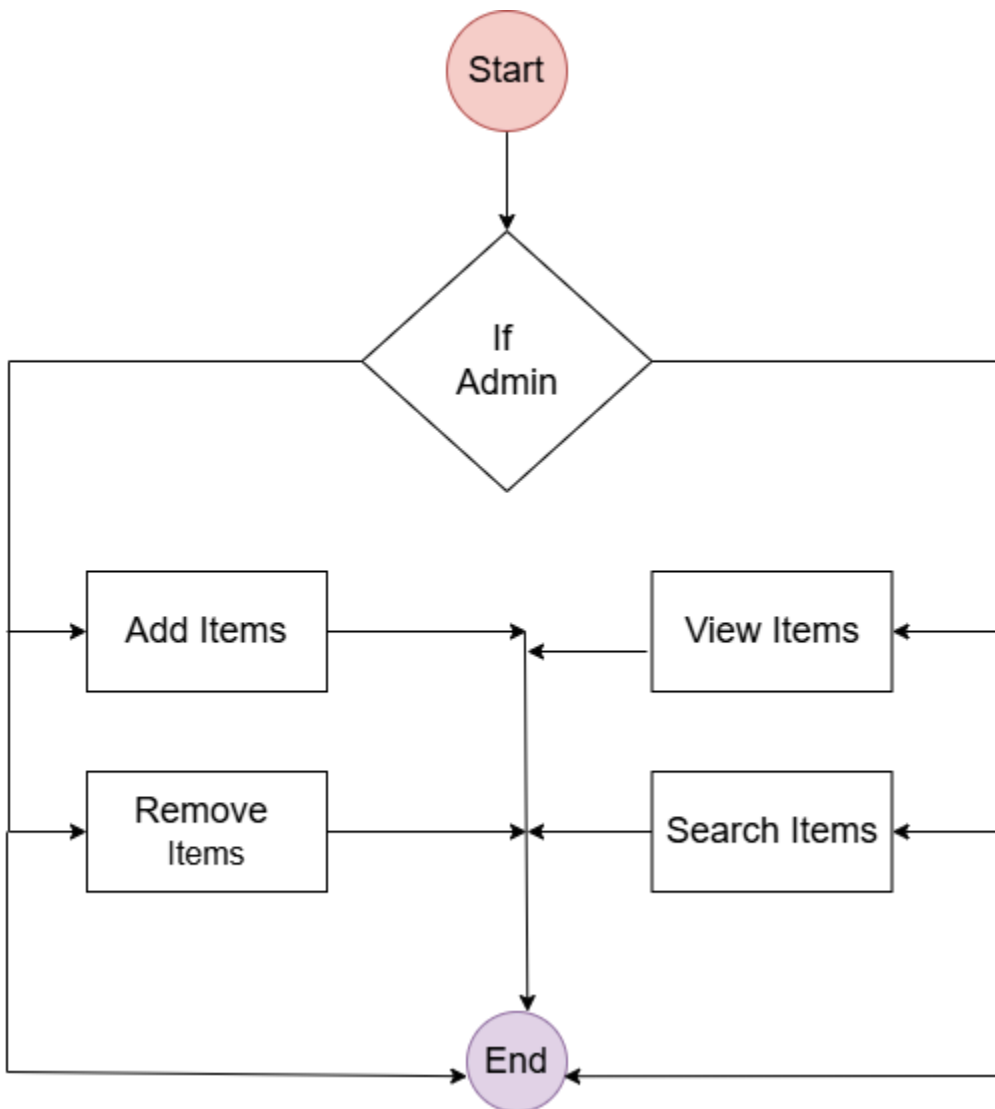


Fig-12:Menu Exploring

Level:1.3
Name:Payment

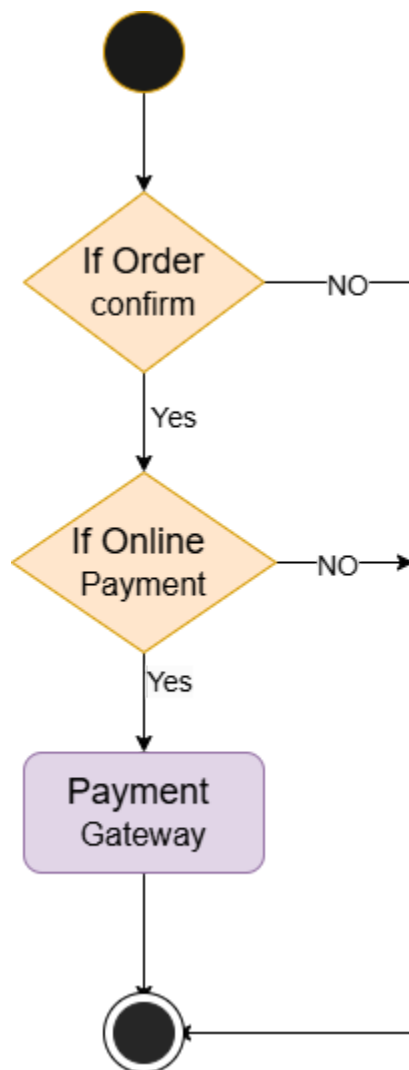


Fig-13:Payment

Level 1.5
Name:Item Ordering

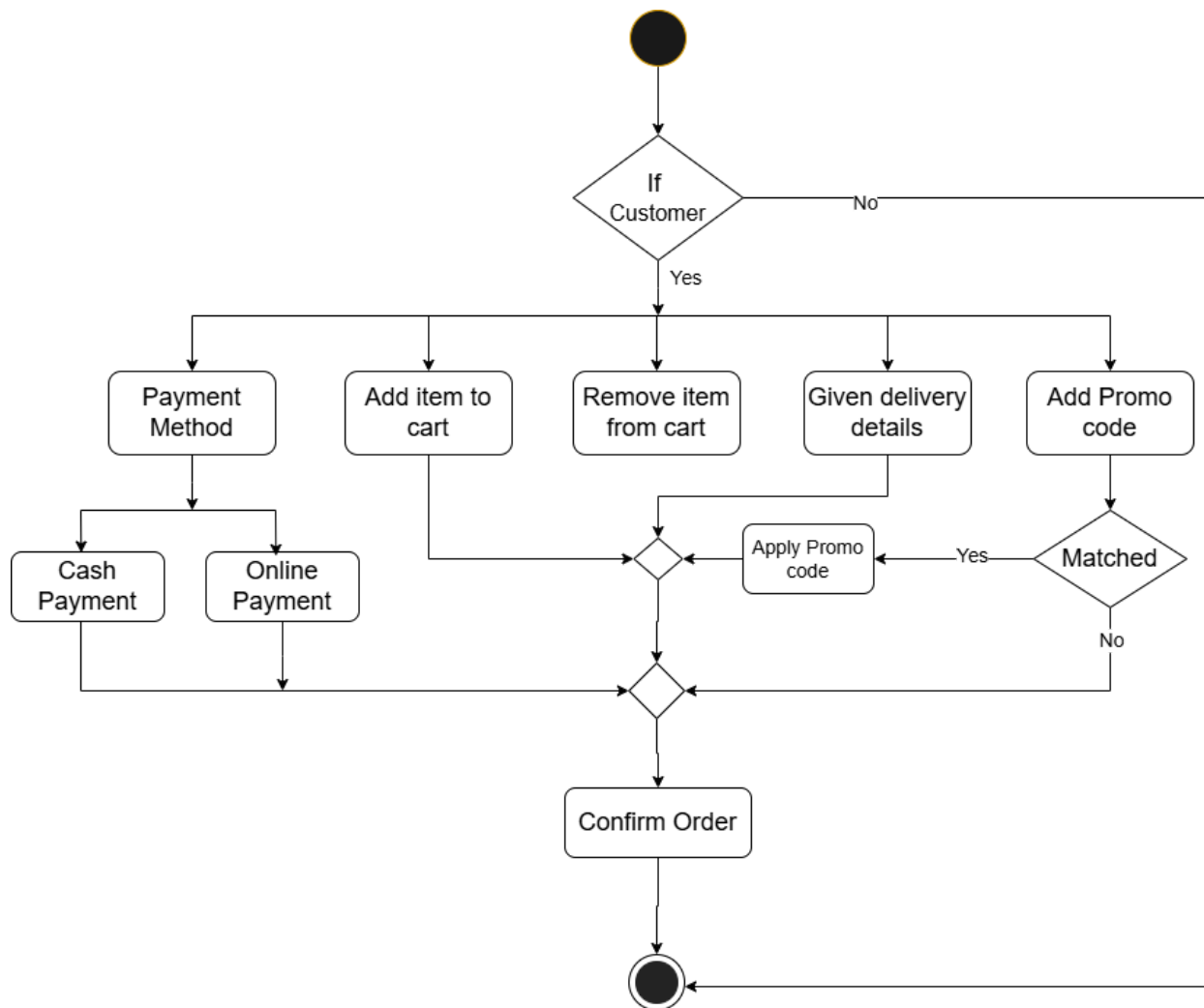


Fig-14:Name:Item Ordering

Level :1.6
Name:Profile

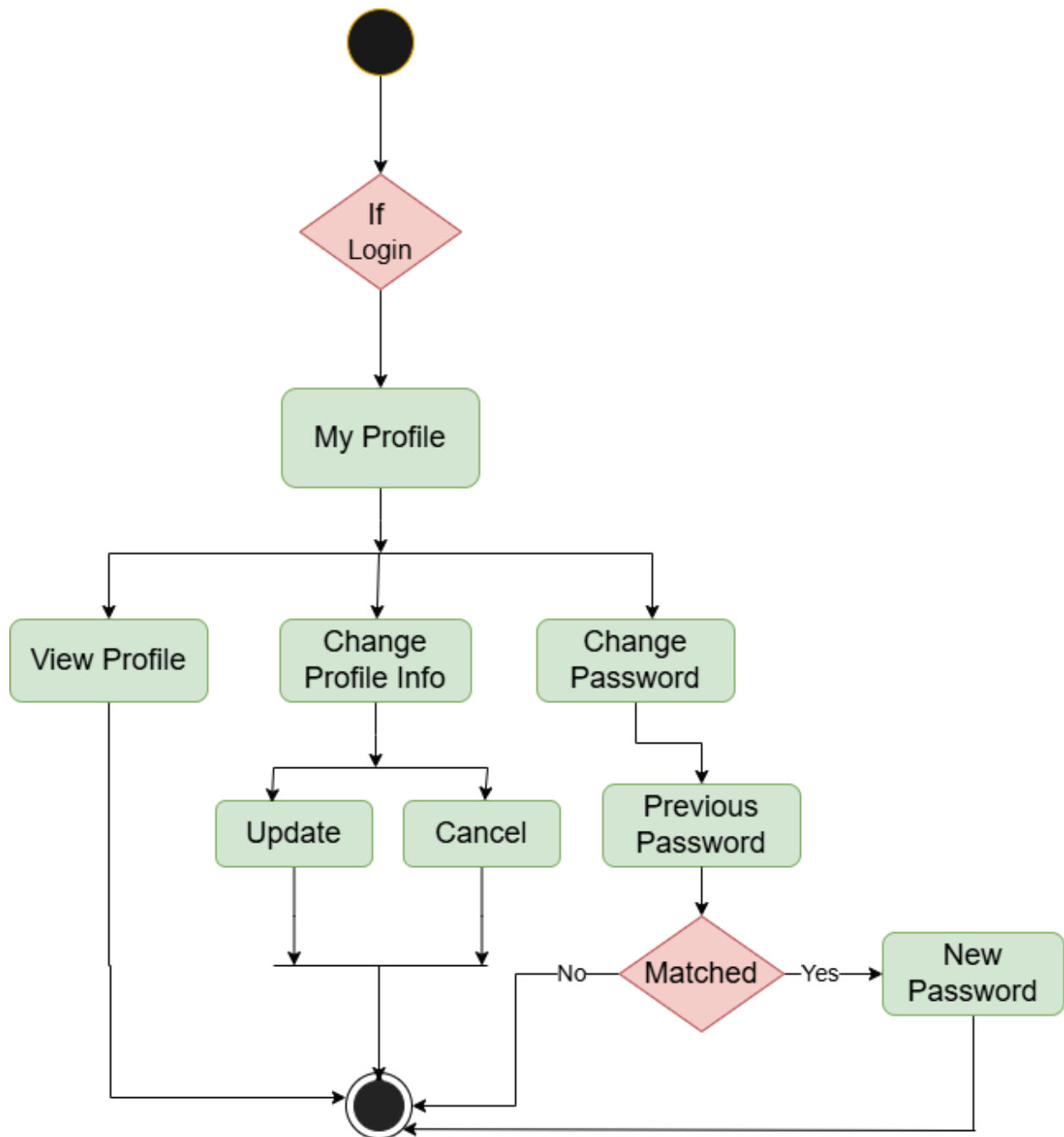


Fig-15:Profile

Swimlane Diagram

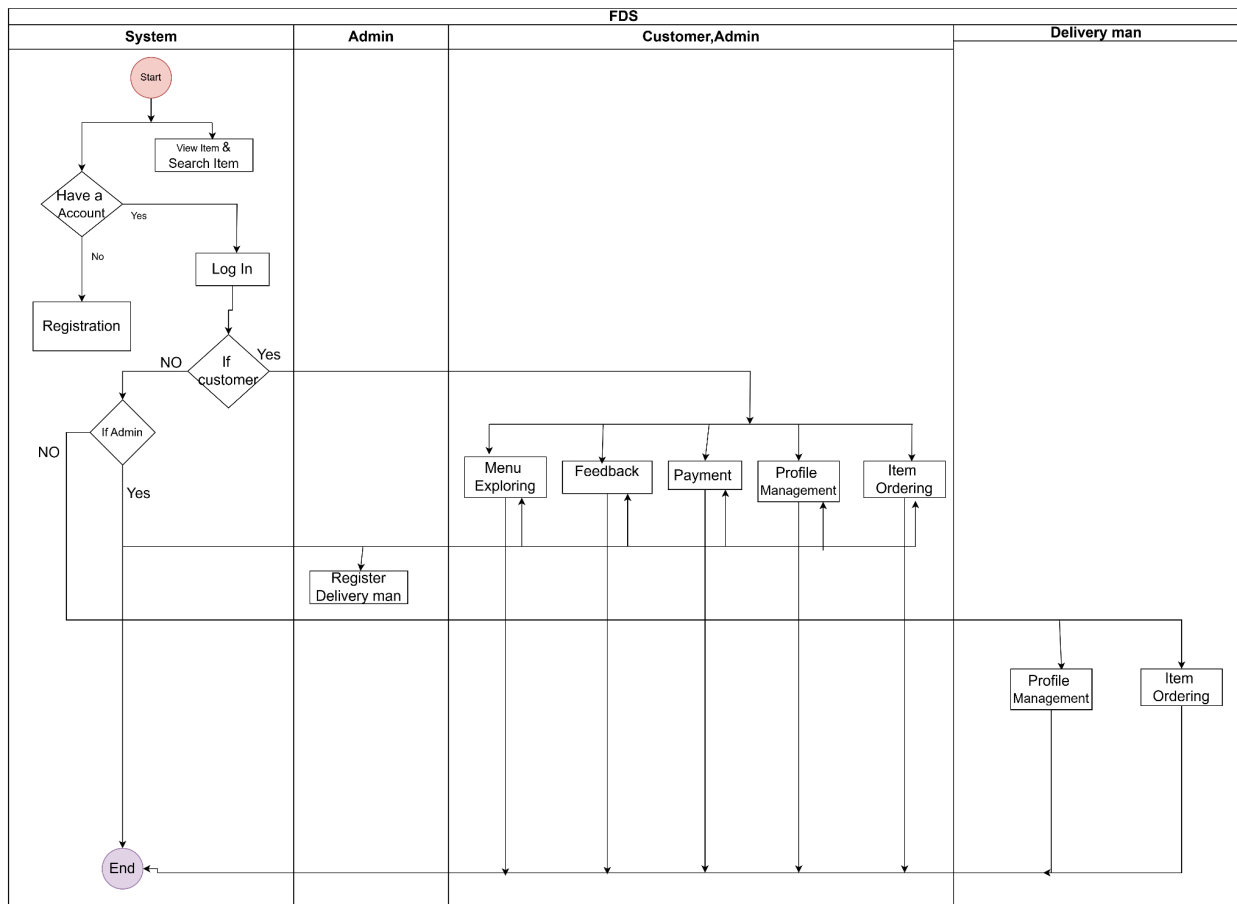
The Swimlane diagram, also known as the Rummler-Brache diagram or across-functional diagram, is a more detailed form of an activity diagram. Swimlanes are sometimes called functional bands. It simply describes who is responsible for the activities being performed in the activity diagram and how they are responsible. The activity diagram only represents the activities being performed, but Swimlane describes who does what in a process or activity performed.

The purpose of a Swimlane diagram:

A swimlane diagram provides various facilities to a software engineer, such as

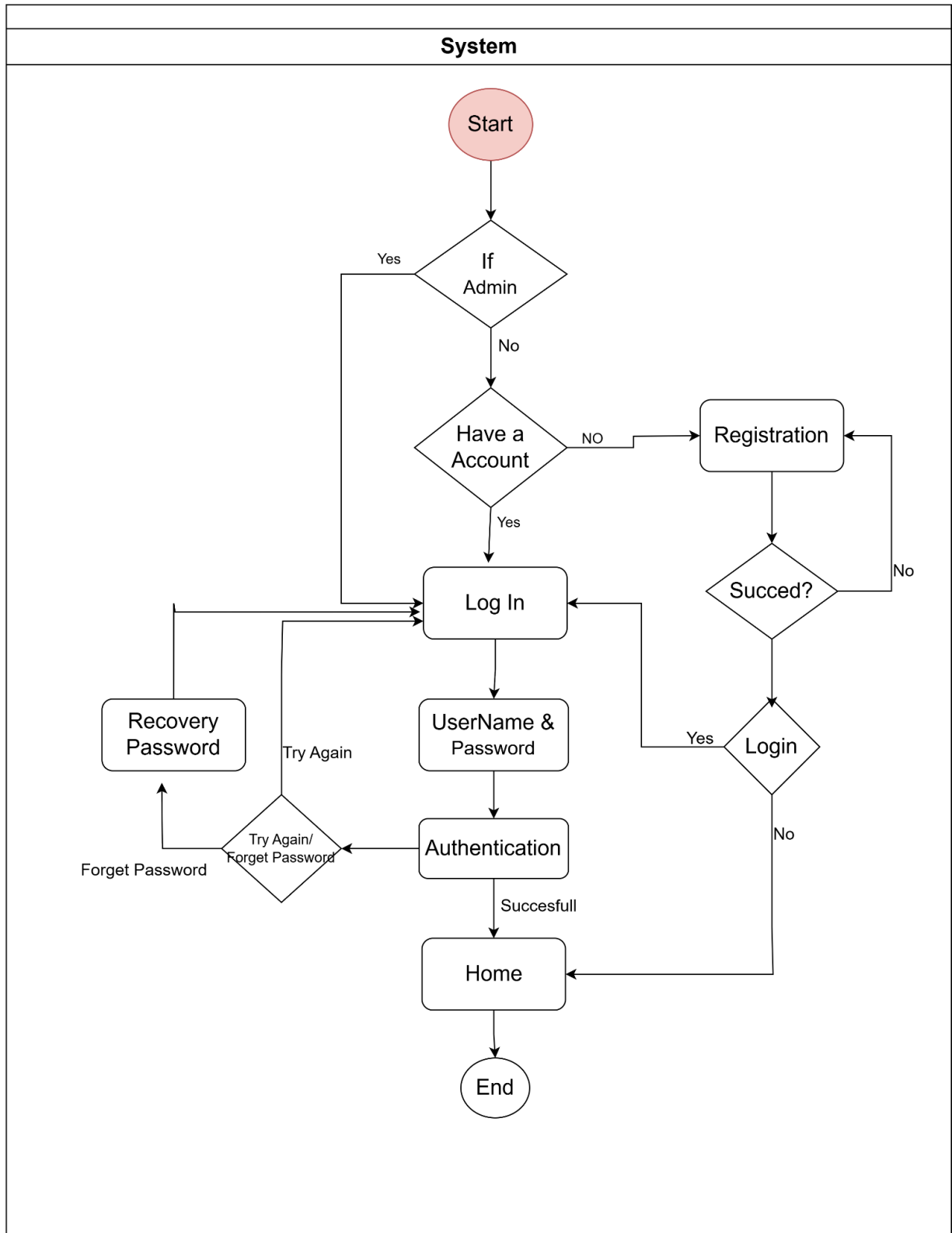
- The separate lanes of the diagram make it easy to delineate responsibilities belonging to certain actors. This helps to clarify complex processes within the software.
- Visualizing processes in this way provides a more thorough overview of an actor's roles within an organization and helps to reduce bottlenecks, redundancies, and extraneous steps.
- Ensures that everyone knows their specific role and avoids collisions.
- Helps to standardize work processes and record them in highly shareable formats that people within the organization can quickly refer to if needed.

Level: 1
Name: Food Delivery System



Level 1.1

Name: Authentication and Authorization



Data Based Modelling

Introduction

A Data Model is an organized view of database concepts and their relationships. The purpose of creating a conceptual data model is to establish entities, their attributes, and relationships. In this datamodeling level, there is hardly any detail available on the actual database structure. The 3 basic elements of Data Modeling are-

- Entity: A real-world thing
- Attribute: Characteristics or properties of an entity
- Relationship: Dependency or association between two entities

The entity relationship diagram (ERD) defines all data objects that are processed within the system, the relationships between the data objects and the information about how the data objects are entered, stored, transformed and produced within the system

Data Object Identification

Serial	Noun	Problem/Solution Space	Attribute
1	Customer	S	13,37,38,49
2	Restaurant System (Admin)	S	13,49,
3	Delivery Personnel	S	13,49
4	Payment	S	13,60,64,36
5	Application	P	
6	Menu	P	

7	Items	S	13,55,60,66
8	Cart	P	
9	Checkout	P	
10	Payment Method	S	
11	Order	S	13,15,27,28,34,60,26,37, ,38,64
12	Receipt	P	
13	ID	S	
14	Delivery	P	
15	Status	S	
16	Feedback	S	13,64,65
17	Inventory	P	
18	Funds	P	
19	Refund	P	
20	Unavailability	p	
21	Cancellation	P	
22	Network Connectivity	P	
23	System	P	
24	Order Delivery	P	
25	Special Requests	P	
26	Delivery Information	S	
27	Order Confirmation	S	

28	Estimated Delivery Time	S	
29	App Crash	P	
30	Delivery Failure	P	
31	Order Tracking	p	
32	Customer Support	P	
33	Redelivery	P	
34	Wait Time	S	
35	Tip	P	
36	Transaction ID	S	
37	Address	S	
38	Customer Location	S	
39	Customer Preferences	P	
40	Order Modification	P	
41	Digital Receipt	P	
42	Service Fee	P	
43	Delivery Zone	P	
44	Payment Confirmation	P	
45	Delivery Attempt	P	
46	Estimated Wait Time	P	
47	Order Queue	P	
48	Order Status	S	
49	User Profile	S	67,78,69,70,74

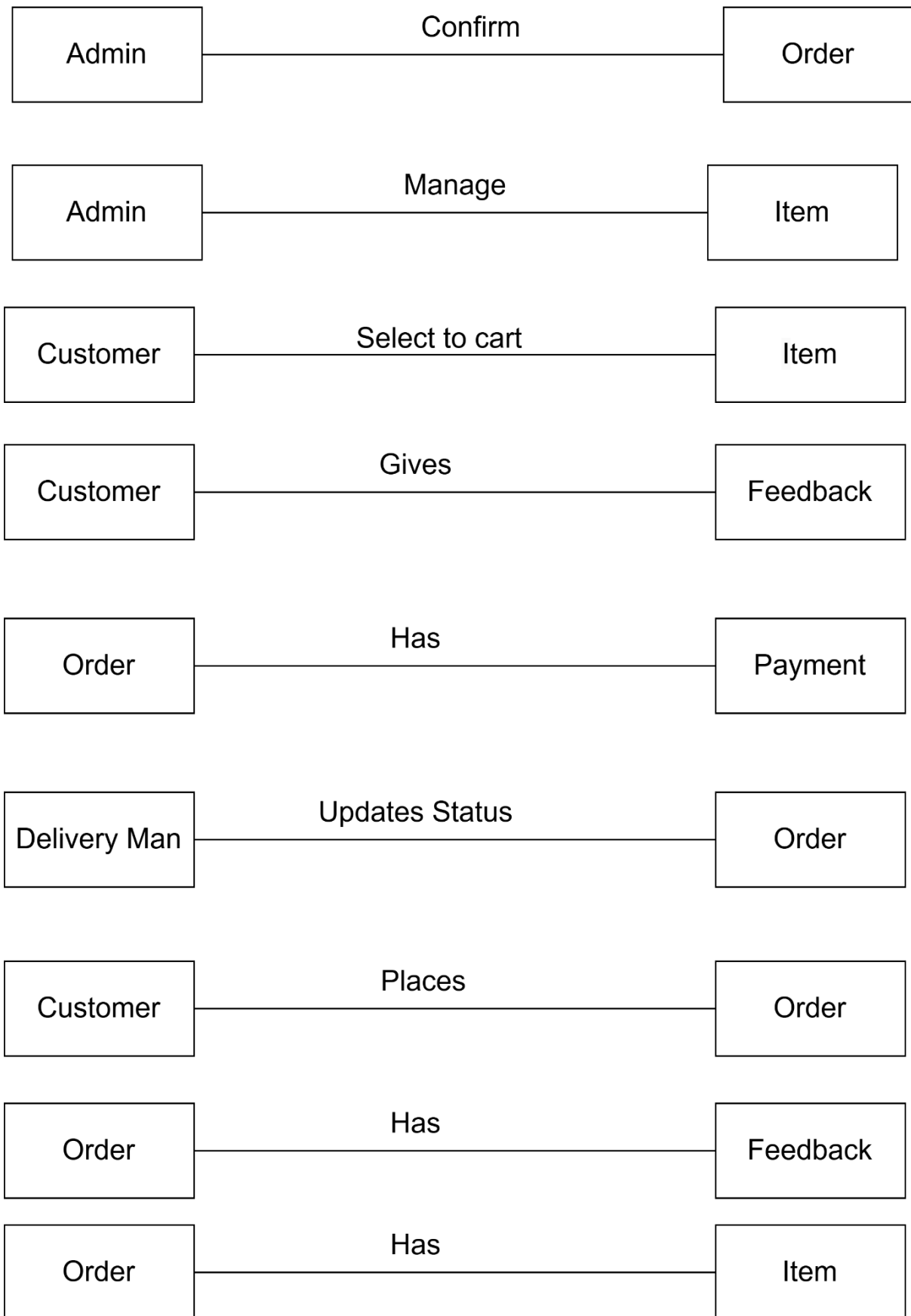
50	Rating	P	
51	Real-time Tracking	P	
52	Order History	P	
53	Item Description	S	
54	Delivery Time Slot	P	
55	Item Price	S	
56	Tax	P	
57	Delivery Distance	S	
58	Order Review	P	
59	Order Modification Request	P	
60	Amount	S	
61	Delivery Confirmation	P	
62	Payment Status	S	
63	Delivery Confirmation Time	P	
64	Date	S	
65	Information	S	
66	Quantity	S	
67	Username	S	
68	Password	S	
69	Phone number	S	
70	Email	S	

71	User	P	
72	System	P	
73	Authentication	P	
74	Name	S	

Final Data Objects

- Customer
- Admin
- Delivery Man
- Payment
- Item
- Feedback
- Order

Relationship Between Data Object



ER Diagram

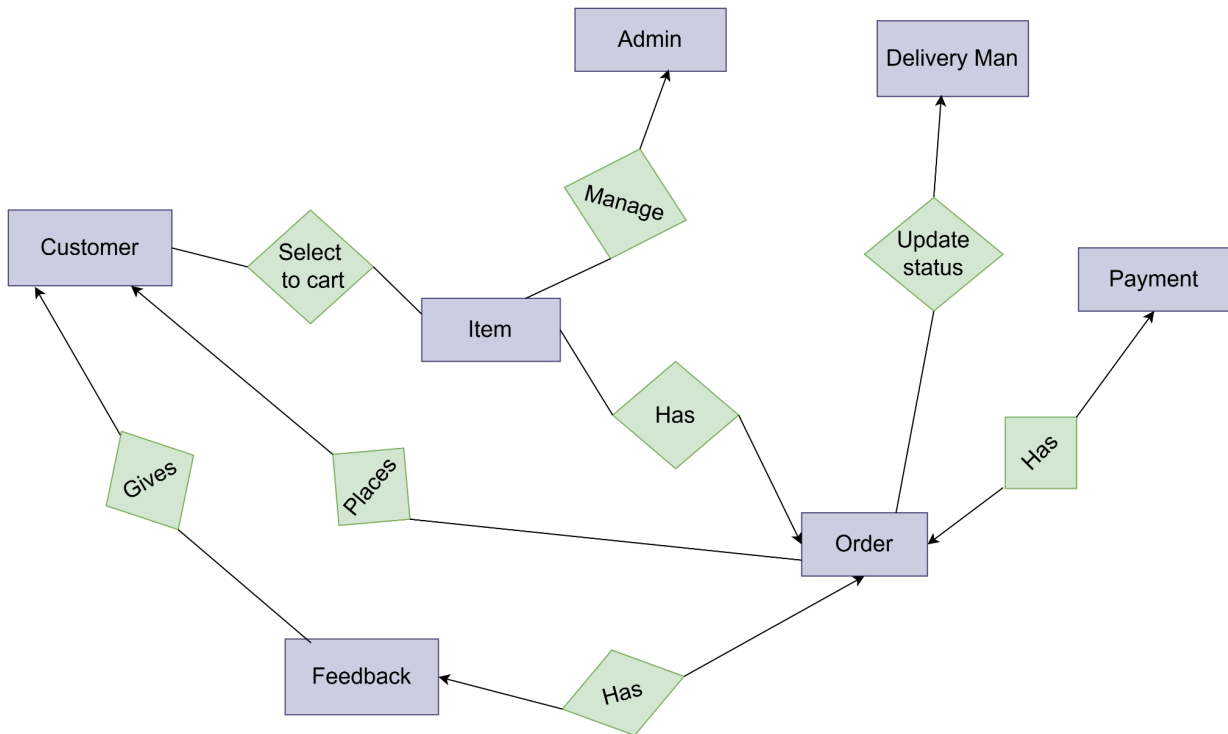


Fig:ER of FDS

Schema diagram

Data Object	Attribute	Type
Customer	<u>customerId</u> username emailId phoneNumber Password address	varchar(10) varchar(20) varchar(25) varchar(15) varchar(15) varchar(100)
Admin	<u>adminId</u> username emailId phoneNumber Password	varchar(10) varchar(20) varchar(25) varchar(15) varchar(15)
Delivery Man	<u>deliveryManId</u> username emailId phoneNumber Password	varchar(10) varchar(20) varchar(25) varchar(15) varchar(15)
Item	<u>ItemId</u> itemDescription itemPrice quantity orderId(foreign key)	varchar(10) varchar(100) number(20) number(10) varchar(10)
Order	<u>orderId</u> status orderConfirmation EstimateDeliveryTime amount deliveryInformation customerLocation date customerId(foreign key)	varchar(10) varchar(50) Bool TIME number(10) varchar(50) varchar(50) DATE varchar(10)

	deliveryManId(foreign key)	varchar(10)
Payment	<u>transactionId</u> amount date	number(20) number(15) DATE
feedback	<u>feedbackId</u> date Information orderId(foreign key)	varchar(20) DATE varchar(20)
SelectToCart	<u>customerId</u> <u>itemId</u> quantity	varchar(10) varchar(10) number(10)

Class Based Modeling

What is Class Based Modeling?

Class-based modeling identifies classes, attributes, and relationships that the system will use. It represents the object. The system manipulates the operations. The elements of the class-based model consist of classes and objects, attributes, operations, and class – responsibility - collaborator (CRC) models. Classes are determined using underlining each noun or noun clause and entering it into the simple table. Attributes are the set of data objects that are defining a complete

class within the context of the problem. The operations define the behavior of an object.

Class Based Modeling: Food Delivery System.

Verb List

Serial	Verb
1	Opens
2	Launches
3	Navigates
4	Viewing
5	Selects
6	Adds
7	Specifying
8	Reviews

9	Checks
10	Makes
11	Proceeds
12	Confirms
13	Enters
14	Selecting
15	Confirms
16	Confirms
17	Receives
18	Provides
19	Receives

20	Initiates
21	Updates
22	Assigns
23	Picks
24	Transports
25	Confirms
26	Updates
27	Has
28	Rates
29	Provides
30	Leaves

31	Is
32	Updated
33	Recorded
34	Provides
35	Saved
36	Updates
37	Processes
38	Directed
39	Notifies
40	Prompts
41	Retry

42	Remove
43	Replace
44	Initiates
45	Cancels
46	Handles
47	Receives
48	Opt
49	Refund
50	Contacts
51	Arranges
52	Marks

53	Informs
54	Saves
55	Assists
56	Re-confirms
57	Cancels
58	Retries
59	Restored

General Classifications

Candidate classes are categorized based on the seven general classification. The analysis classes manifest themselves in one of the following ways:

1. External entities
2. Things
3. Occurrence or Events

4. Roles

5. Organizational units

6. Places

7. Structures

A candidate class is selected for special classification if it fulfills three or more characteristics.

Serial	Noun	General Classification
1	Customer	1,4,5,7
2	Restaurant System (Admin)	1,4,5,7
3	Delivery Personnel	1,4,5,7
4	Payment Gateway	1,3,6
5	Application	1,6
6	Menu	2,6,7
7	Items	2,6,7
8	Cart	2,6,7
9	Checkout	3
10	Payment Method	2
11	Order	2,3,6
12	Receipt	2
13	Tracking ID	2

14	Delivery	3
15	Status	2
16	Feedback	2,3,6
17	Inventory	2,,3,6
18	Funds	2
19	Refund	2
20	Unavailability	2
21	Cancellation	3
22	Network Connectivity	3
23	System	3,7
24	Order Delivery	3
25	Special Requests	3
26	Delivery Information	2
27	Order Confirmation	3
28	Estimated Delivery Time	2
29	App Crash	3
30	Delivery Failure	3
31	Order Tracking	3
32	Customer Support	3
33	Redelivery	3
34	Wait Time	2
35	Tip	2

36	Transaction	2,3,7
37	Delivery Address	2
38	Customer Location	2
39	Customer Preferences	2
40	Order Modification	3
41	Digital Receipt	2
42	Service Fee	2
43	Delivery Zone	2
44	Payment Confirmation	3
45	Delivery Attempt	3
46	Estimated Wait Time	2
47	Order Queue	2
48	Order Status	2
49	User	2,6,7
50	Rating	2
51	Real-time Tracking	3
52	Order History	2
53	Item Description	2
54	Delivery Time Slot	2
55	Item Price	2
56	Tax	2
57	Delivery Distance	2

58	Order Review	3
59	Order Modification Request	3
60	Account Balance	2
61	Delivery Confirmation	3
62	Payment Status	2
63	Delivery Confirmation Time	2

Selection Criteria

The candidate classes are then selected as classes by six Selection Criteria:

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

A candidate class generally becomes a class when it fulfills around three characteristics.

NO	Potential Class	Selection Criteria
1	Customer	1,2,3,4,5
2	Admin	1,2,3,4,5
3	Delivery Personnel	1,2,3,4,5
4	Payment Gateway	1,2,6

5	Menu	1,2,3,4,5
6	Item	1,2,3,4,5
7	Cart	1,2,3,4,5
8	Order	1,2,3,4,5
9	Feedback	1,2,3,4,5
10	Inventory	3,4,5
11	Transaction	1,2,6
12	User	1,2,3,4,5

5.2.4 Selected Classes

- 1.Customer
- 2.Admin (Restaurant System)
- 3.DeliveryPersonnel
- 4.Menu
- 5.MenuItem
- 6.Cart
- 7.Order
- 8.User
- 9.PaymentGateway
- 10.Transaction

Class-Card.

User

Attribute	Methods
Id(String) Name(String) email(String) phone(String) address(String)	updateProfile() viewProfile() resetPassword() Browse menu()
Responsibility	Collaboration
Browse menu.	System

Customer	
Attribute	Methods
customerId(String) orderHistory(List<Order>) cart(Cart)	browseMenu(); addToCart(); viewCart(); checkout(); trackOrder(); giveFeedback();
Responsibility	Collaboration
Browse menu.	Menu

Add items to cart. Review and modify cart. Proceed to checkout. Select payment method. Track order. Provide feedback.	Cart Order PaymentGateway
--	---------------------------------

Admin	
Attribute	Methods
menu (Menu) orders (List<Order>)	updateMenu(); updateOrderStatus(); confirmOrder(); cancelOrder();
Responsibility	Collaboration
Manage menu items (add, update, remove). Monitor and update order status. Handle order cancellations.	Menu Order

DeliveryPersonnel	
Attribute	Methods
personnelId (String) name (String) availabilityStatus currentOrder (Order)	updateDeliveryStatus();
Responsibility	Collaboration
Update delivery status (e.g., "Picked Up," "Delivered").	Order

Menu	
Attribute	Methods
menuItems(List<MenuItem>)	getMenuItems();
Responsibility	Collaboration
Maintain and provide a list of menu items.	MenuItem Admin

Update availability of items.	Customer
-------------------------------	----------

MenuItem	
Attribute	Methods
itemId (String) name (String) description (String) price (double) availability(boolean)	updateAvailability();
Responsibility	Collaboration
Hold details about an individual dish (e.g., name, price, description). Manage availability status.	Menu Admin Cart

Cart	
Attribute	Methods
items(List<menuItem>) totalPrice (double)	addItem(); removeItem();

	<code>updateQuantity();</code> <code>calculateTotal();</code>
Responsibility	Collaboration
Store selected items. Calculate total cost of items in the cart. Allow modifications to item quantities and special requests.	Customer MenuItem

Order	
Attribute	Methods
<code>orderId (String)</code> <code>customer (Customer)</code> <code>items(List<CartItem>)</code> <code>totalPrice (double)</code> <code>deliveryAddress(String)</code> <code>status (String)</code> <code>deliveryPersonnel(DeliveryPersonnel)</code>	<code>updateStatus();</code> <code>assignDeliveryPersonnel();</code>
Responsibility	Collaboration

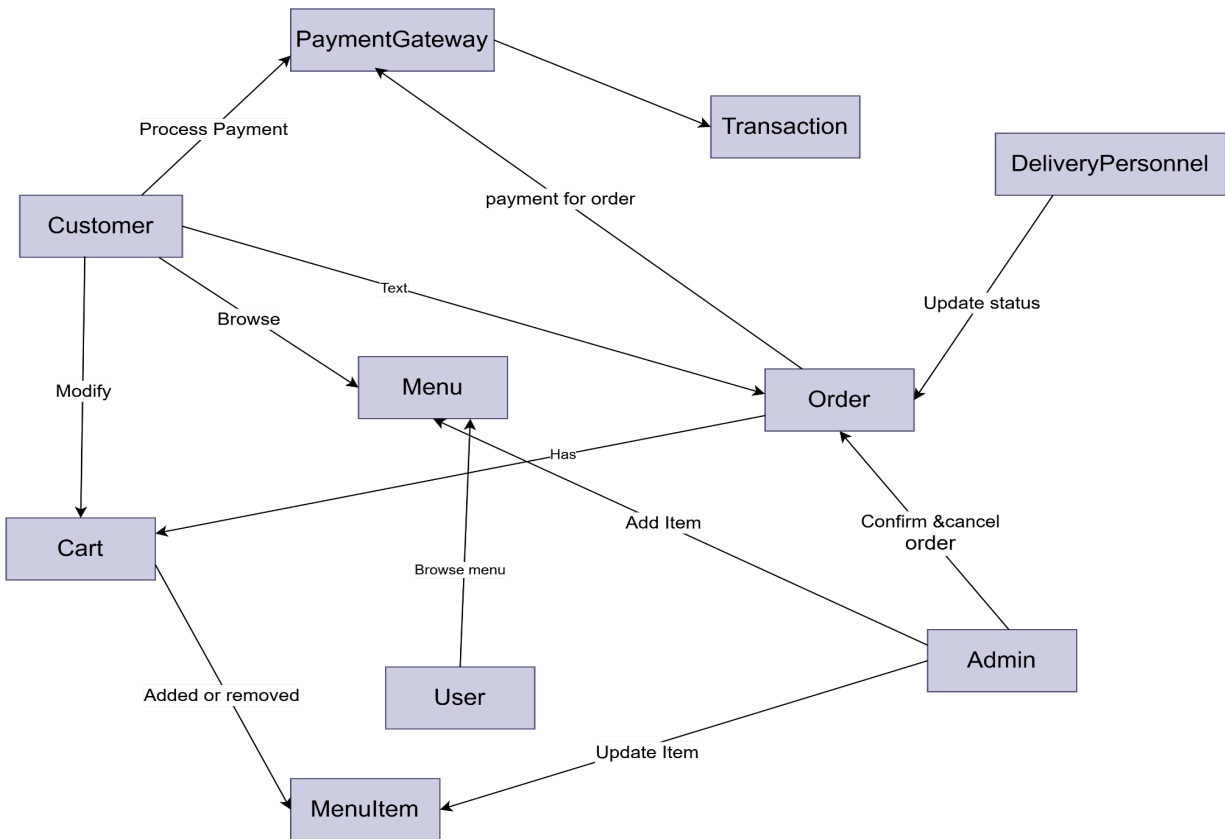
Record order details (items, total price, customer, and delivery info). Track and update order status.	Customer Cart DeliveryPersonnel
---	---------------------------------------

PaymentGateway	
Attribute	Methods
transactionId(String) status (String) amount (double)	processPayment(); refundPayment();
Responsibility	Collaboration
Process payments securely. Handle transaction status (success or failure). Initiate refunds if needed.	Customer Order

Transaction

Attribute	Methods
transactionId(String) status (String) timestamp (DateTime)	
Responsibility	Collaboration
Record payment transaction details (ID, amount, status, timestamp).	PaymentGateway Order

CRC Diagram



Behavioral Modeling

The behavioral model indicates how the software will respond to external events or stimuli. In the context of behavioral modeling, two different characterizations of states must be considered: (1) the state of each class as the system performs its function and (2) the state of the system as observed from the outside as the system performs its function.

State Transition Diagram: Food Delivery System

One component of a behavioral model is a UML state diagram that represents active states for each class and the events (triggers) that cause changes

between these active states. State Transition Diagram represents active states for each class of events (triggers). For this we identified all the events, their initiators and collaborators. In the State Transition Diagram the states are shown in boxed texts, and the transition is represented by arrows. It is also called State Chart or Graph. It is useful in identifying valid transitions.

List of Events

Event	Initiator Class	Collaborator Classes
Browse Menu	Customer	Menu
Add Item to Cart	Customer	Cart, MenuItem
View Cart	Customer	Cart
Modify Cart	Customer	Cart, CartItem
Proceed to Checkout	Customer	Cart, Order
Select Payment Method	Customer	PaymentGateway, PaymentDetails
Process Payment	PaymentGateway	Customer, Order, PaymentDetails, Transaction
Confirm Order	Order	Customer, Admin
Update Order Status (Preparation)	Admin	Order
Assign Delivery Personnel	Admin	DeliveryPersonnel, Order
Update Order Status (Out for Delivery)	DeliveryPersonnel	Order
Real-Time Order Tracking	Customer	DeliveryPersonnel, Order

Deliver Order	DeliveryPersonnel	Order, Customer
Confirm Delivery	Customer	Order
Provide Feedback	Customer	Order, Admin
Update Menu	Admin	Menu, MenuItem
Update Inventory	Admin	Inventory, MenuItem
Cancel Order	Admin	Customer, Order, PaymentGateway
Process Refund	PaymentGateway	Transaction, Customer, Order
Notify Unavailable Items	Cart	MenuItem, Customer
Handle Payment Failure	PaymentGateway	Customer, Order

Pre-register Admin/Manager	System Administrator	User Management System
Register Customer	Customer	Registration System, User Management System
Add Delivery Personnel	Admin/Manager	User Management System
Log in to System	Customer, Admin, Delivery Personnel	Authentication System
Validate User Credentials	Authentication System	User Database
Provide Access to the System	Authentication System	System Interface

State Transition Diagram

Name: Customer

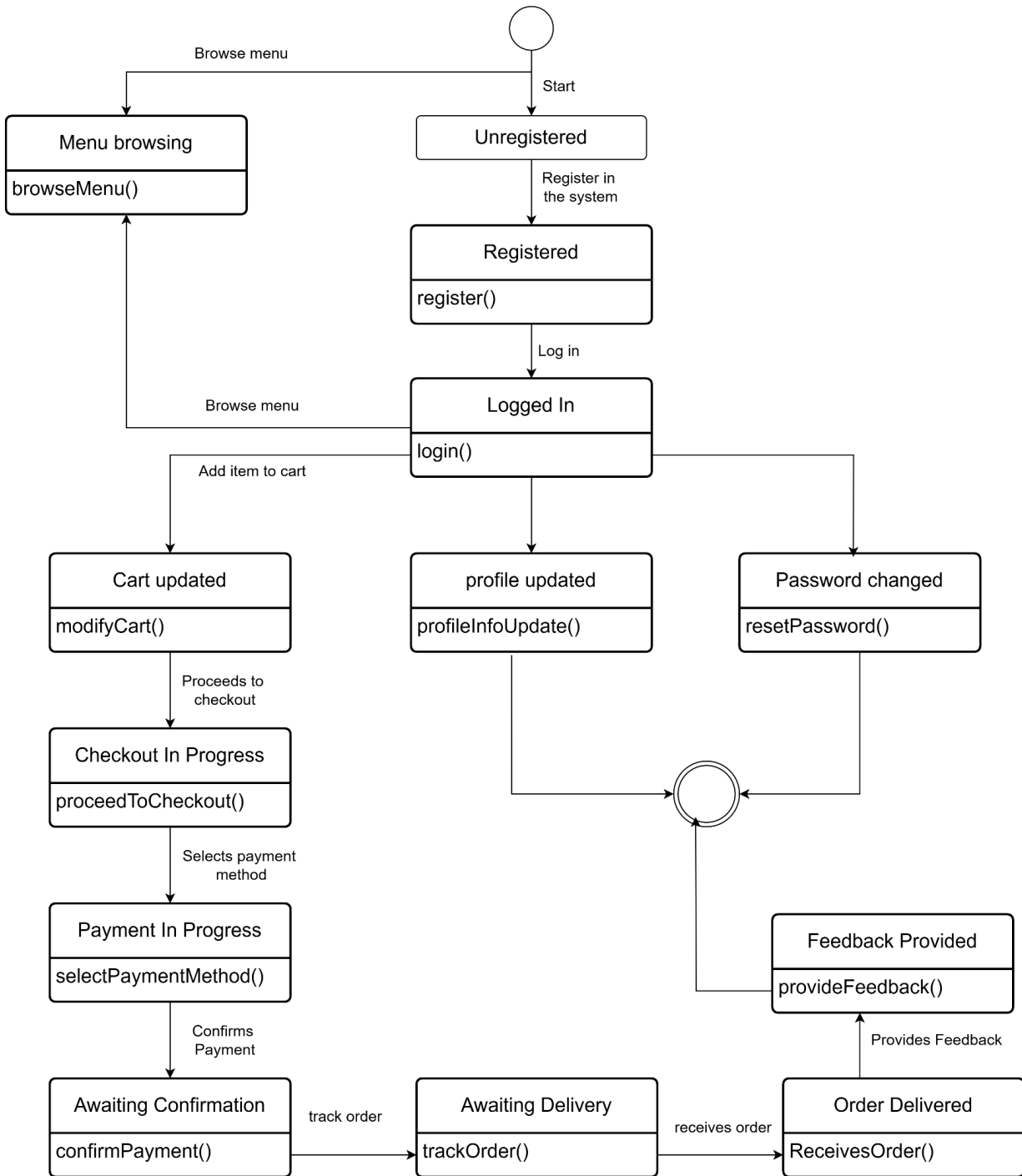


Fig:Customer

Name: Cart

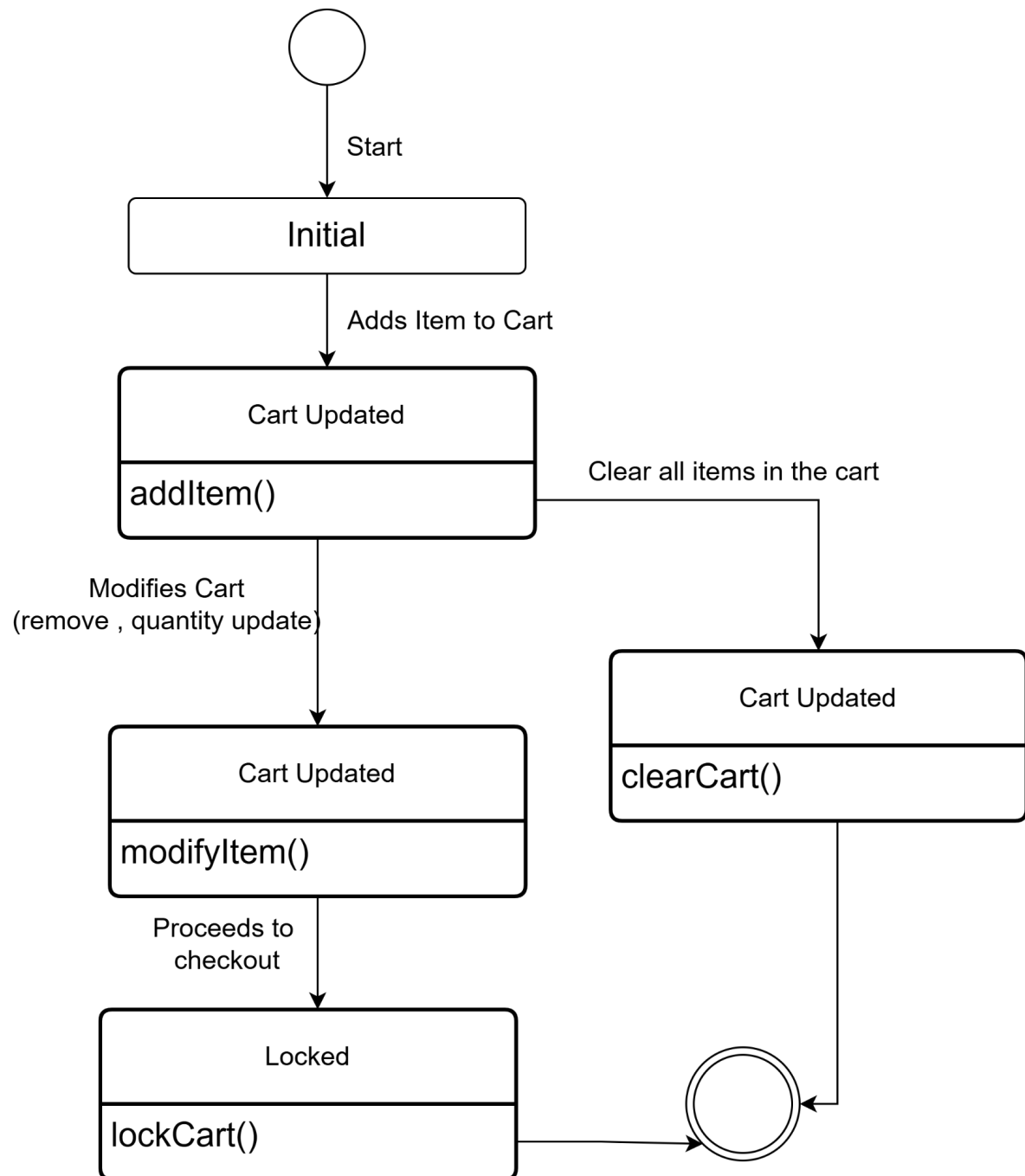


Fig:Cart

Name:Order

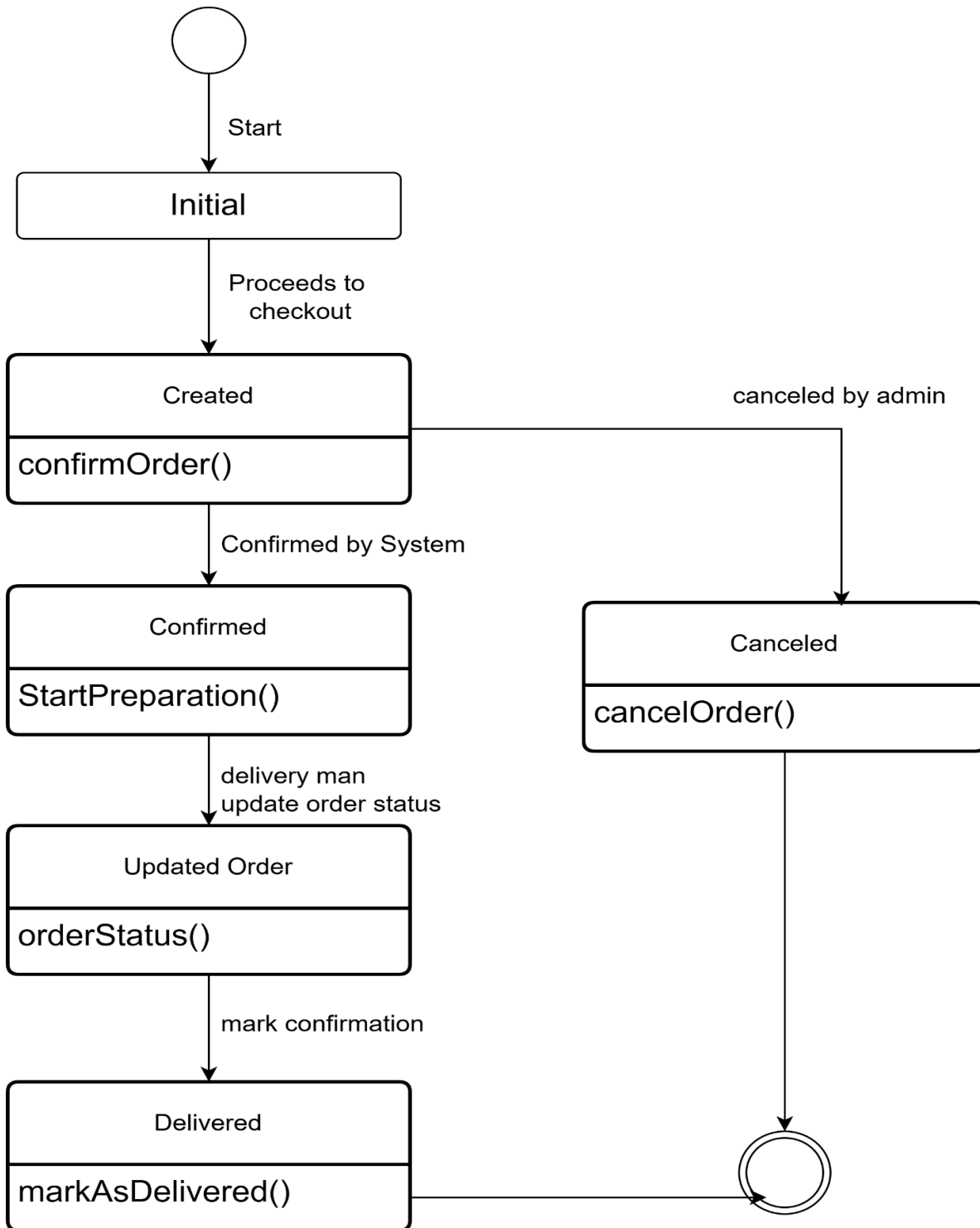


Fig:Order

Name:Delivery Personnel

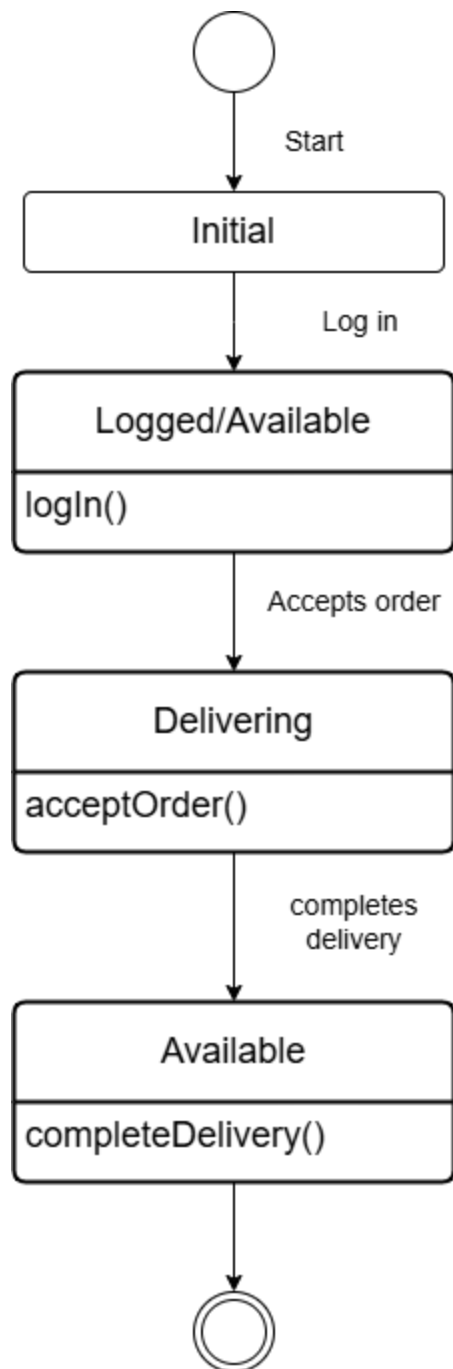


Fig:Delivery Personnel

Name:Payment Gateway

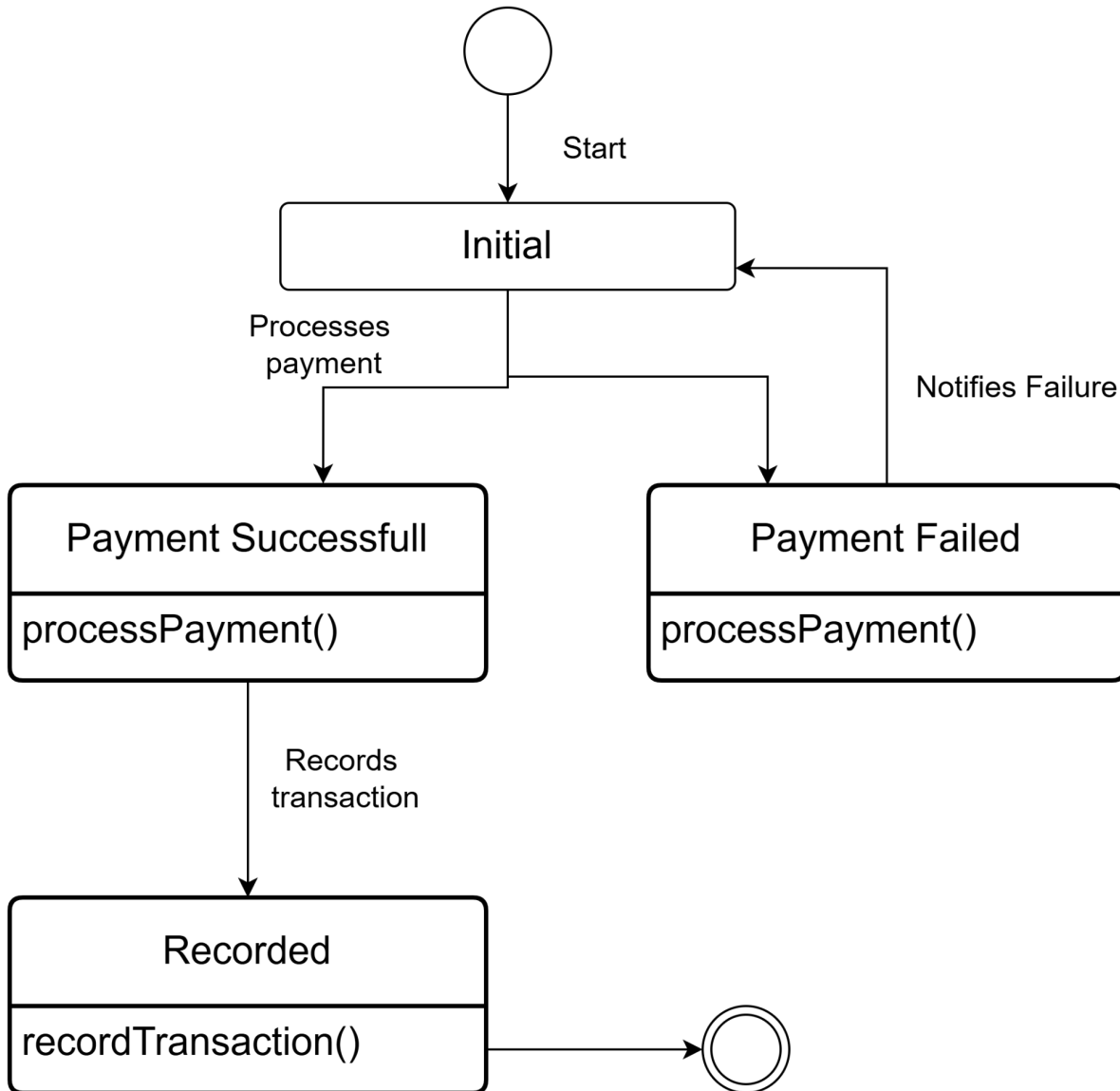
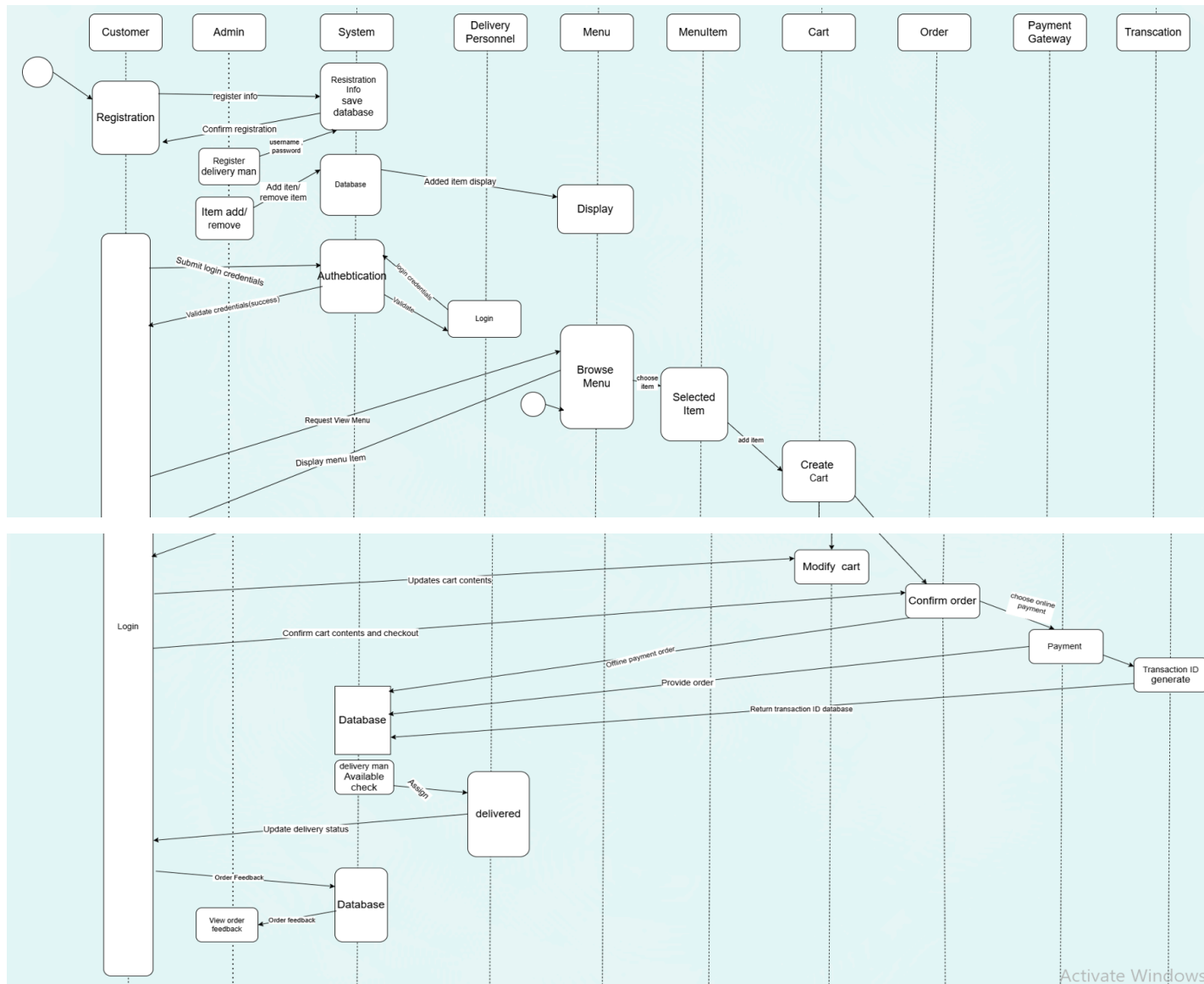


Fig:Payment Gateway

Sequence diagram: Food Delivery System

The second type of behavioral representation, called a sequence diagram in UML, represents how events cause flow from one object to another as a function of time. The sequence diagram is a shorthand version of the use case. It represents vital classes and the events that cause behavior to flow from class to class.

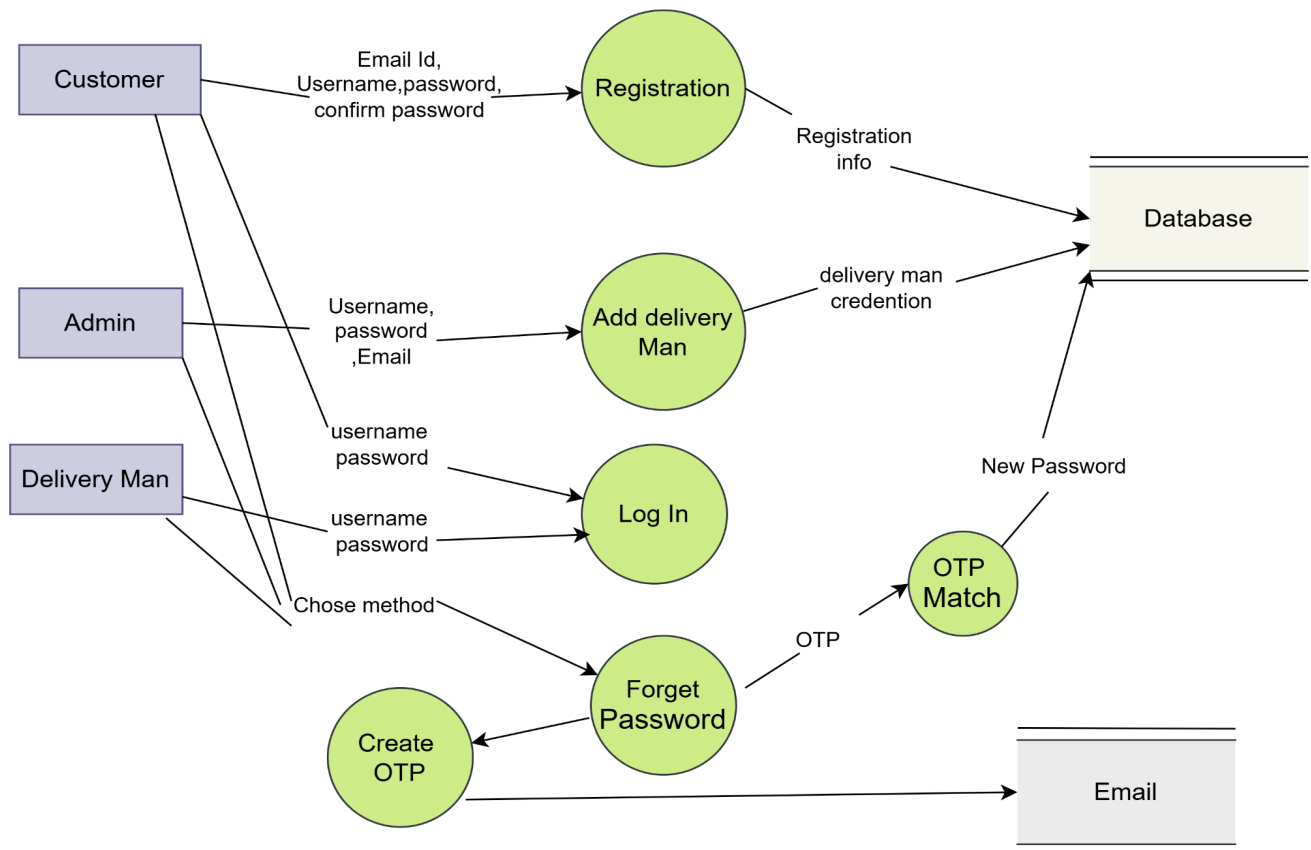


7.Data Flow Diagram (DFD)

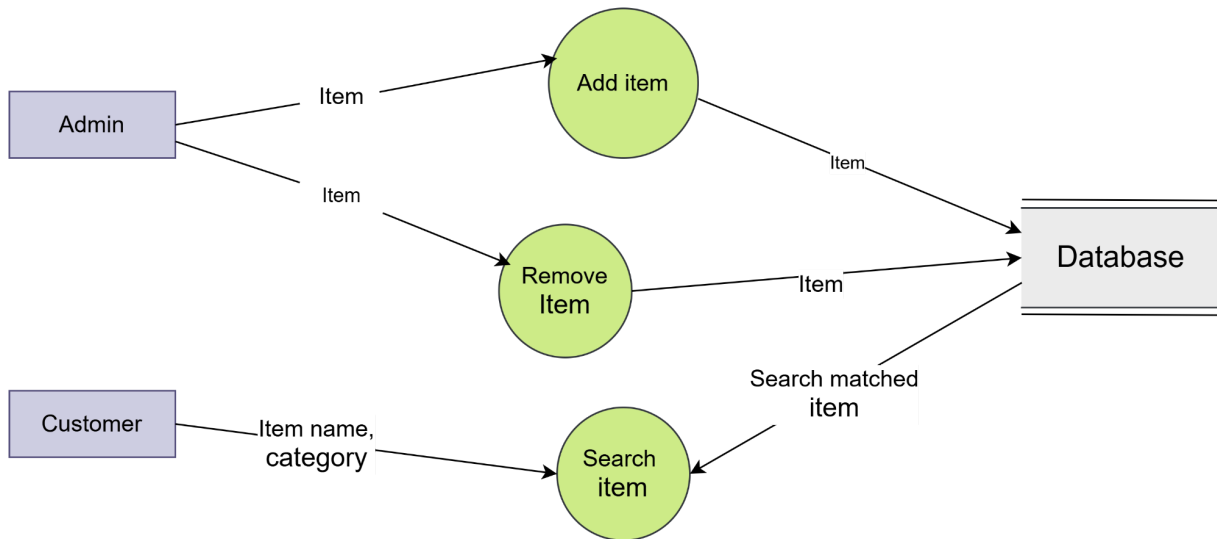
DFD, or Data Flow Diagram, is a visual representation that illustrates the flow of information within a system. It employs various symbols to depict processes, data stores, data flows, and external entities, providing a clear and concise overview of how data moves through a system. DFDs are essential in system analysis and design as they help stakeholders, including analysts and end-users, to comprehend

the information flow and interactions within a complex system. By breaking down a system into manageable components and highlighting data transformations, DFDs facilitate effective communication, aid in identifying potential bottlenecks or areas for improvement, and serve as a foundation for designing and implementing efficient information systems.

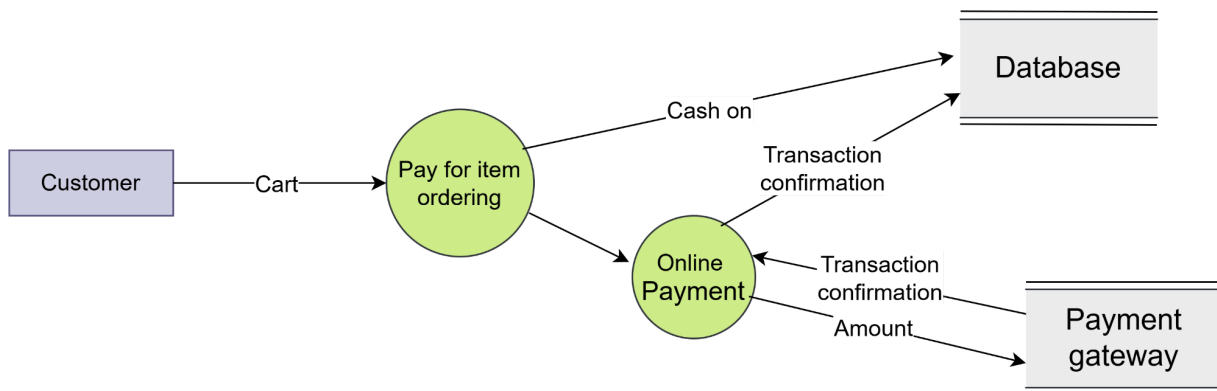
Level 1



Level 2



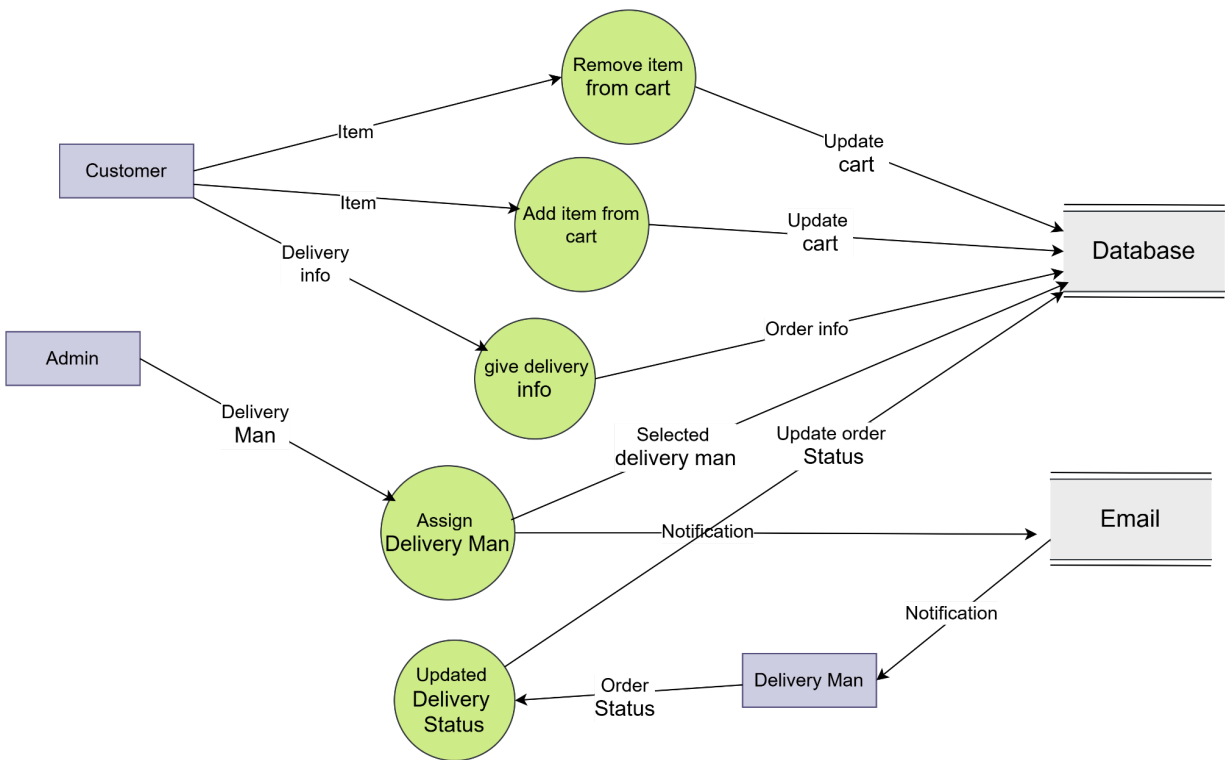
Level 3



Level 4



Level 5



Level 6

