E-commerce system

Ahmed Bahy Yousif	2
Hassan Hussien Azmy	7
Abdelrhman Mahmoud Mohmed	15
Abdallah Hany Ragab	19
Mahmoud Ahmed Mahmoud	32

Supervised by: Dr: Rania Ramdan and

Eng: Mofida Mahmoud

Contents

Abstract

The E-Commerce System is a robust and multi-functional platform designed to streamline online shopping, order management, and administrative operations. The system comprises three primary applications: a Customer Application, an Administrator Application, and a Delivery Application, each serving distinct roles to ensure seamless transactions and efficient order processing.

Customers can register, log in, browse products, and manage their shopping carts. The system validates user input, checks product availability, and processes secure payments. Order cancellation is supported before delivery, triggering inventory updates and refunds.

Administrators have full control over product management, including adding, editing, and deleting items. They can track orders through various states (approved, prepared, archived, or cancelled) and generate detailed reports on sales, inventory, and user activity for data-driven decision-making.

Delivery personnel receive assigned orders, accept suitable ones, and update order statuses, with real-time notifications sent to customers.

Introduction

Our project presents a robust E-Commerce System designed to provide a seamless shopping experience while ensuring efficient management of products, orders, and deliveries. The system addresses the needs of three key user groups—customers, administrators, and delivery personnel.

For customers, the system offers a user-friendly interface where they can register, browse products, add items to their cart, and complete purchases with secure payment processing. Customers can also track their orders and request cancellations if needed, with the system automatically updating inventory and processing refunds. The administrator application provides comprehensive control over the platform, allowing admins to add, edit, or remove products, manage orders, and generate detailed reports on sales, inventory, and customer activity. These reports can be exported in multiple formats for further analysis, helping businesses make data-driven decisions.

The delivery application ensures efficient order fulfillment by allowing delivery personnel to accept orders, update their status, and notify customers in real time. This streamlined process enhances transparency and reliability in the delivery process. By integrating these components, the E-Commerce System optimizes operations, improves customer satisfaction, and supports scalable business growth. Built with security and usability in mind, this platform serves as a modern solution for online retail, meeting the evolving demands of digital commerce.

System description

The e-commerce system is a comprehensive platform composed of three main applications—for customers, delivery personnel, and administrators—alongside support functionalities for customer service and warehouse operations.

The Customer, customers initially should have an account to log in for authenticating. If he doesn't have an account, he must register to the system.

For registration, the customer enters his info then the system validates it, adds the customer to data base of the system and sends the id to him. If customer info is not valid the system displays error messages to him.

Customer can browse and search products, view detailed products, choose quantity needed, system check availability and quantity and add items to his cart. Customers also can remove from cart; first system display cart's items and customers select the item and remove it, if customers want to place orders the system initially retrieves cart items and check availability of items then customers make a payment by entering their credit info, system checks if info valid then system update inventory in the system and sends confirmation message to customer. Customer can cancel order if order isn't delivered yet, if he confirms the cancellation the system updates the status of order, return items to inventory, return the money or payment to customer and send cancellation message to customer.

Admin provides a comprehensive interface for administrators, it allows adding, editing, and deleting products. Manage order state to be approved, prepared, archived, and cancelled. Detailed reports are available across sales, inventory, returns, and user activity, analyzing trends, and export data in multiple formats (pdf, word, etc.) for business insights and operational decisions

Delivery initially receives a list of orders, he either accepts orders that are suitable for him, adds them to his own delivery list, the system updates the status of those orders and sends tracking information to both customers and admin that the order is ready and, in the way, archives the order if it's done.

Event Table

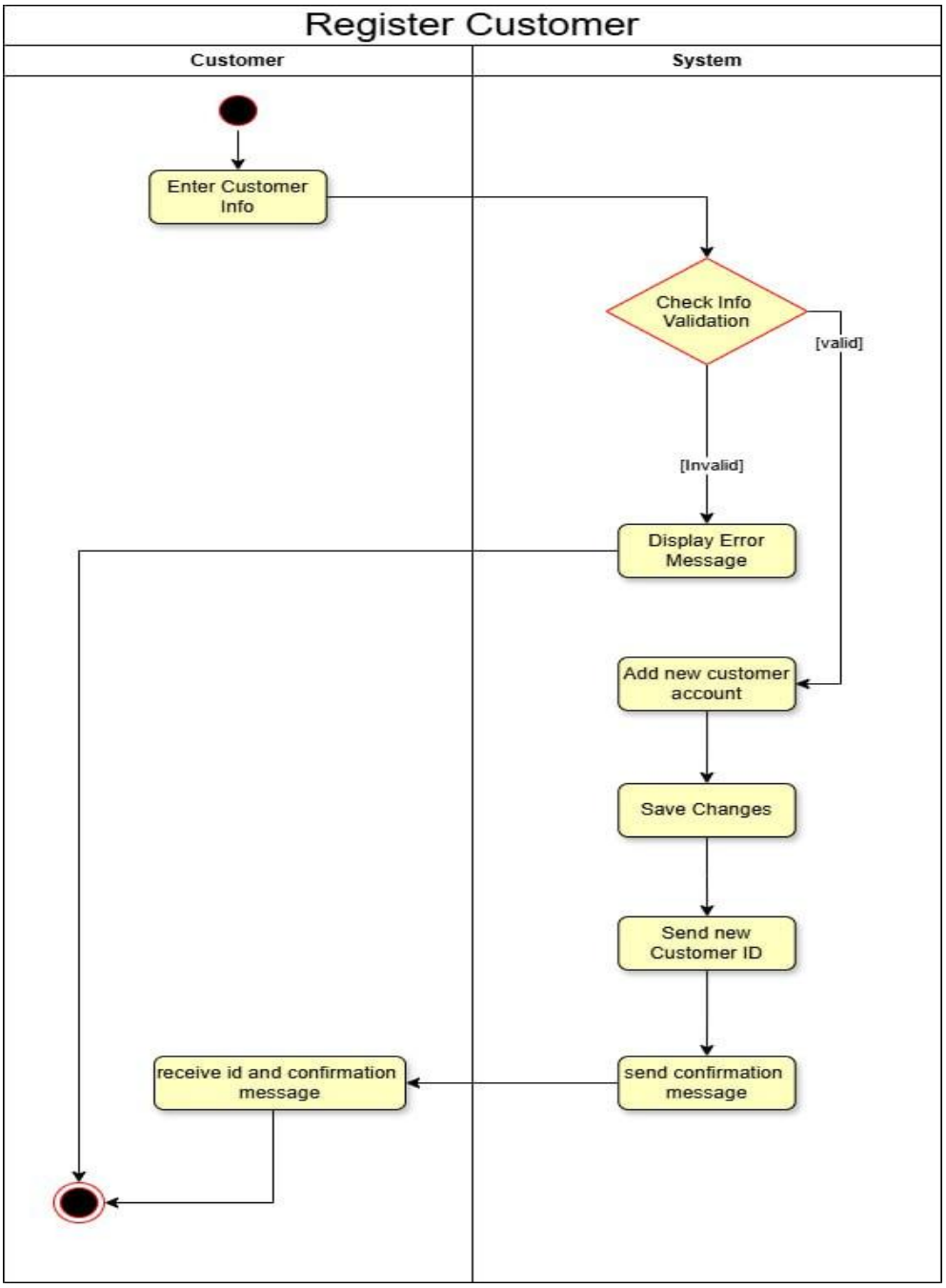
#	Event	Trigger	Source	Use Case	Response	Destination
1	Customer registers a new account	• Customer info • Registration request	Customer	Register a new account	• Customer id Confirmation message	Customer
2	Customer logins to the system	• Login request • email and password	Customer	login	• Authentication Token	Customer
3	Customer modifies shopping cart	• Item addition / deletion request	Customer	Modify Cart	• Updated shopping cart	Customer
4	Customers place an order	• Checkout request • Order info	Customer	place an order	• Order invoice	Customer
5	Customer cancels an order	• Order id • Cancel order request	Customer	Cancel order	• Cancellation message	Customer
6	Admins manage orders	• Order id Approve/ prepare /archive order request	Admin	manage orders	Notification message Updated order status	delivery Customer
7	Admin updates inventory	• Item addition / deletion / update	Admin	update inventory	• updated inventory	
8	Delivery modifies orders state	• Order id Approve/ archive order request	Delivery	modify orders state	• Tracking information • Order state updated in database.	Customer Admin
9	system generate detailed report	• End of day Report request		Generate report	• Daily report	admin

UML

Register e new account

Customers must browse, search and make orders in the system. Customer enters his info; system validates it and sends id to customer. If info is not valid the system sends an error message to the customer.

activity diagram



Scenario

<table>
<tr><td>Use Case Name:</td><td colspan="2">Register New Customer</td></tr>
<tr><td>Scenario:</td><td colspan="2">New customer signs up for an account</td></tr>
<tr><td>Triggering Event:</td><td colspan="2">Customer wants to create a new account.</td></tr>
<tr><td>Brief Description:</td><td colspan="2">customer provides personal information to register a new account. The system checks the validity of the information, creates a customer profile, saves it, and sends a unique customer ID back to the customer.</td></tr>
<tr><td>Actors:</td><td colspan="2">Customer</td></tr>
<tr><td>Related use case</td><td colspan="2">Validate info.</td></tr>
<tr><td>Stakeholders:</td><td colspan="2">Customer</td></tr>
<tr><td>Preconditions:</td><td colspan="2">• Customer is not currently registered.
• The registration system is accessible.</td></tr>
<tr><td>Post conditions:</td><td colspan="2">• A new customer account has been created and stored.
• A unique customer ID is sent to the customer.</td></tr>
<tr><td rowspan="2">Flow of Activities:</td><td>customer</td><td>System</td></tr>
<tr><td>1. Enter personal information

2.Receive customer ID</td><td>1.1 System validates the information

1.2 If valid, proceed to create account

1.3 If invalid, display error message

1.4 Add new customer account

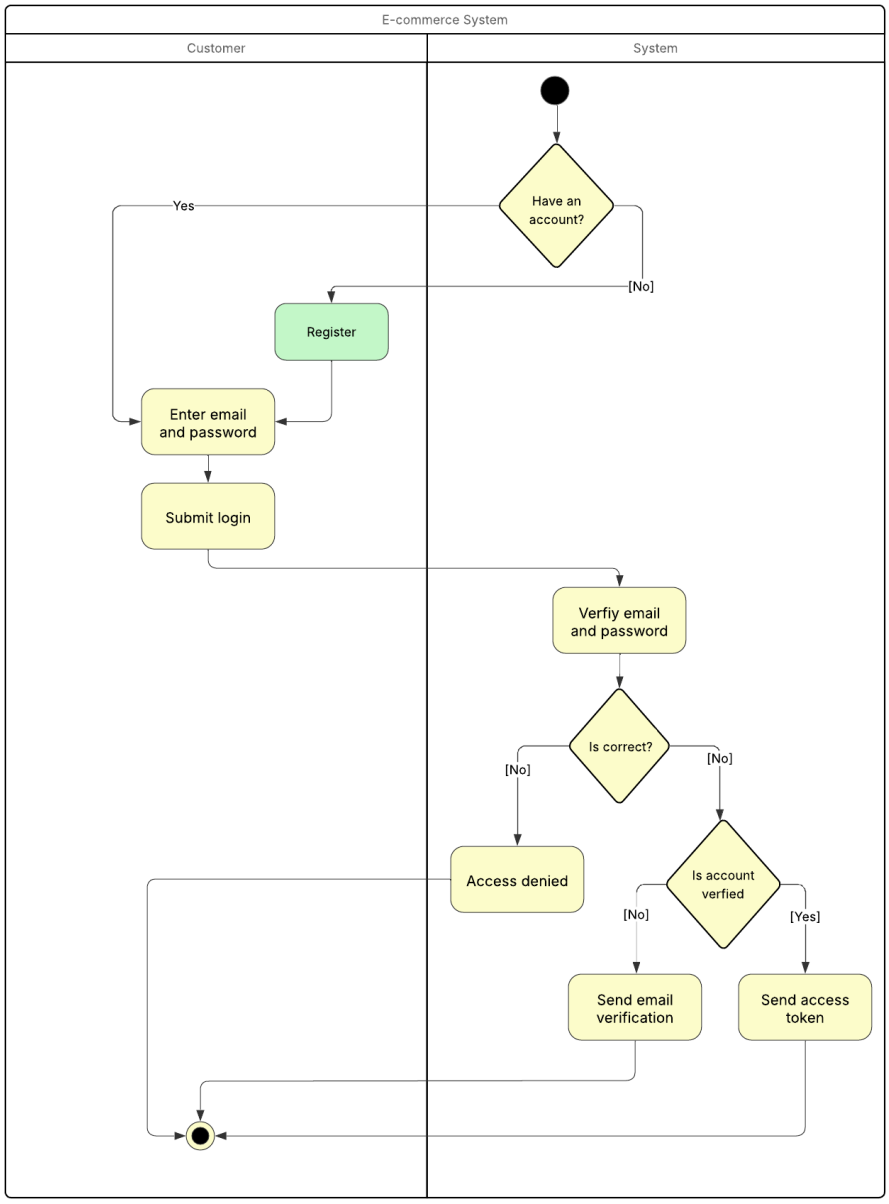
1.5 Save customer data

1.6 Send unique customer ID</td></tr>
<tr><td>Exception Conditions:</td><td colspan="2">1.1 Information is invalid
1.2 Email or ID already exists in the system</td></tr>
</table>

Login

If a customer has an account, he can login by enter his email and password, the system verifies that info and if valid the authentication token is sent to the user.

Activity diagram



Scenario

<table>
<tr><td>Use Case Name:</td><td colspan="2">Login</td></tr>
<tr><td>Scenario:</td><td colspan="2">A user wants to access their account by providing valid login credentials.</td></tr>
<tr><td>Triggering Event:</td><td colspan="2">• The user submits the login form with email and password.</td></tr>
<tr><td>Brief Description:</td><td colspan="2">This use case allows a registered and approved user to log into the system. It validates the credentials and grants access upon successful authentication.</td></tr>
<tr><td>Actors:</td><td colspan="2">• User
System</td></tr>
<tr><td>Related use case</td><td colspan="2">• Verify info</td></tr>
<tr><td>Stakeholders:</td><td colspan="2">• Customer
• Admin
• Delivery</td></tr>
<tr><td>Preconditions:</td><td colspan="2">• The user is already registered.
• The user has a valid and verified email.
• The system is online and accessible.</td></tr>
<tr><td>Post conditions:</td><td colspan="2">• On success: User is authenticated and granted access with a session token.
• On failure: User remains unauthenticated and receives an appropriate error message</td></tr>
<tr><td rowspan="2">Flow of Activities:</td><td>Admin</td><td>System</td></tr>
<tr><td>1. User enters email and password.</td><td>1.1 The system checks if the user exists
1.2 System verifies password
1.3 checks if the user account is approved.
1.4 system returns a success message and session/auth token.</td></tr>
<tr><td>Exception Conditions:</td><td colspan="2">1.1 If the email or password is incorrect Show error: "email or password is not correct."

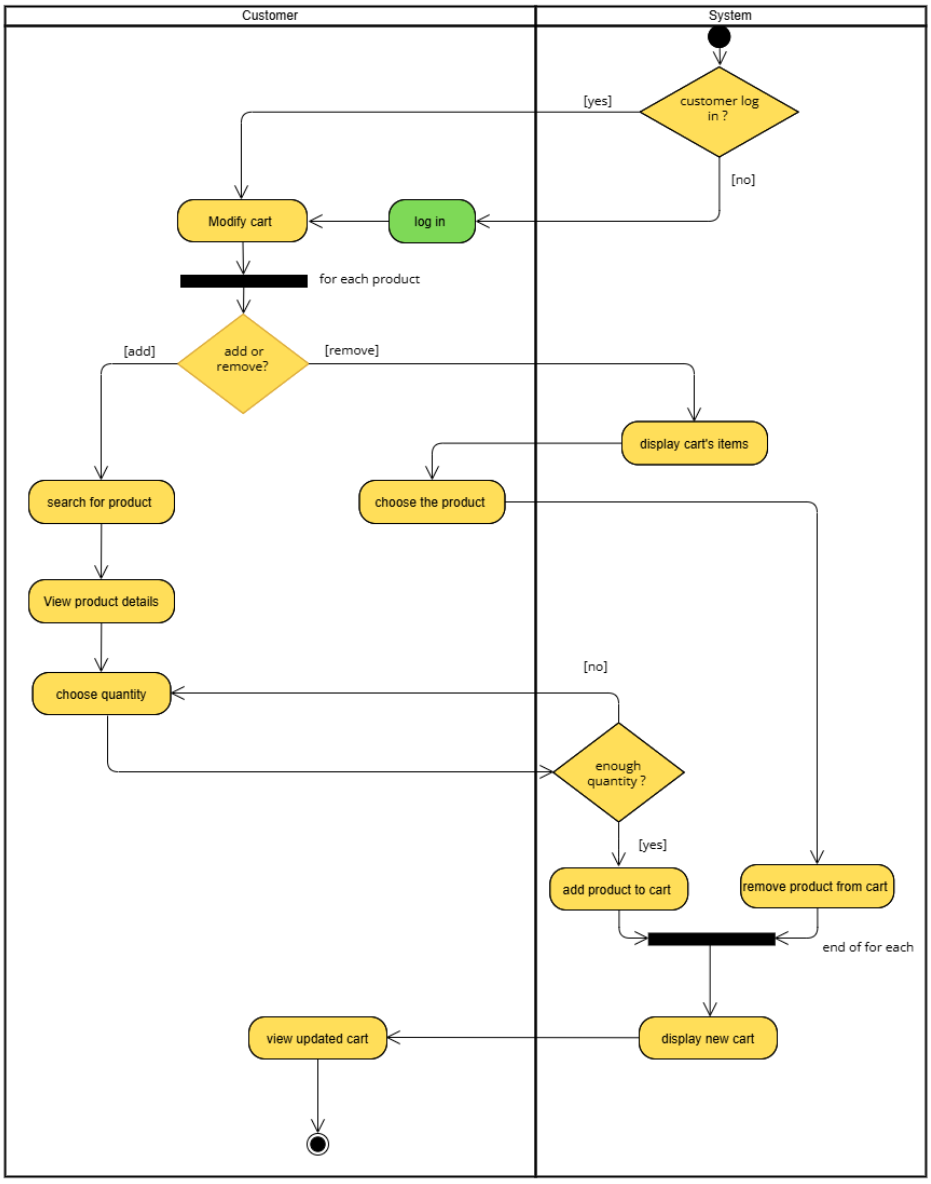
1.2 If the account is not approved → Show error: "Email should be verified." And send verification message

1.3 If database/server is unavailable → Show error: "Login service temporarily unavailable."</td></tr>
</table>

Modify Cart

Customers can make an order by add elements to the cart and then place orders, initially to modify the cart by adding or removing items to the cart and system checks the quantity needed then the system updates the cart and displays new cart

activity diagram



Scenario of add item

<table>
<tr><td>Use Case Name:</td><td colspan="2">Modify Cart</td></tr>
<tr><td>Scenario:</td><td colspan="2">Add items to cart.</td></tr>
<tr><td>Triggering Event:</td><td colspan="2">Customers want to Item addition.</td></tr>
<tr><td>Brief Description:</td><td colspan="2">Customer logs in and interacts with the shopping cart by adding new products, The system validates stock availability, updates the cart accordingly, and displays the new cart.</td></tr>
<tr><td>Actors:</td><td colspan="2">Customer</td></tr>
<tr><td>Related use case</td><td colspan="2">Check quantity.</td></tr>
<tr><td>Stakeholders:</td><td colspan="2">Customer</td></tr>
<tr><td>Preconditions:</td><td colspan="2">• The customer is logged into the system.
• Products exist and available.</td></tr>
<tr><td>Post conditions:</td><td colspan="2">• The shopping cart is updated with the correct item quantities and reflects the current product availability.
• The updated cart is displayed to the customer.</td></tr>
<tr><td rowspan="2">Flow of Activities:</td><td><u>customer</u></td><td><u>System</u></td></tr>
<tr><td>1. customer login into system

2. The customer selects to add an item to the cart.

3. customer search for items and view details
4. customers choose quantity.

5. Customer view new cart</td><td>1.1 The system checks if the user is authorized and has access to modify

4.1 system check if quantity is available

4.2 product added to cart

4.3 system update cart

4.4 system display new cart</td></tr>
<tr><td>Exception Conditions:</td><td colspan="2">1.1 customer is not authorized
4.1 quantity is unavailable</td></tr>
</table>

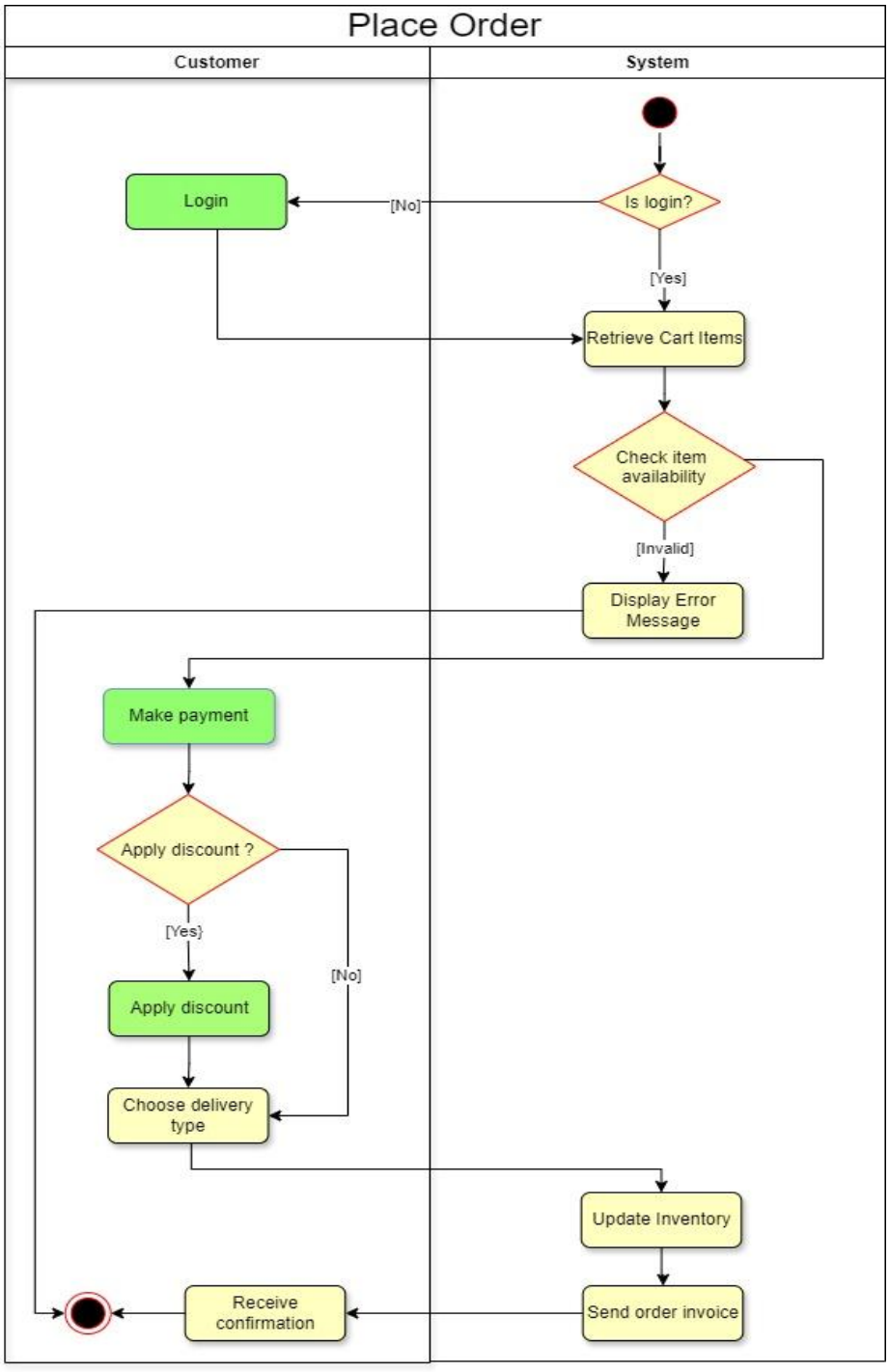
Scenario of remove items

Use Case Name:	Modify Cart	
Scenario:	Remove items from cart .	
Triggering Event:	Customer want to remove item from the cart .	
Brief Description:	customer logs in customer and interacts with the shopping cart by removing existing ones. updates the cart accordingly, and displays the new cart .	
Actors:	Customer	
Related use case	Check quantity.	
Stakeholders:	Customer	
Preconditions:	• The customer is logged into the system.	
Post conditions:	• The shopping cart is updated with the correct item quantities and reflects the current product availability. • The updated cart is displayed to the customer.	
Flow of Activities:	customer	System
	6. customer logs into system 7. Customer selects to remove an item from the cart. 8. customer select product to remove 9. view new cart	1.2 The system checks if the user is authorized and has access to modify 2.1 system display cart 3.1 remove from cart 3.2 system update cart 3.3 system display new cart
Exception Conditions:	1.1 customer is not authorized 2.1 customers choose remove and cart is empty	

Place order

After customer choose items and add them to cart, he can place order and pay for this order then system updates quantity of remaining items and send order confirmation messages to customer

Activity diagram



Scenario

<table>
<tr><td>Use Case Name:</td><td colspan="2">Place an order</td></tr>
<tr><td>Scenario:</td><td colspan="2">places an order for items in cart</td></tr>
<tr><td>Triggering Event:</td><td colspan="2">Customer decides to purchase items from their shopping cart.</td></tr>
<tr><td>Brief Description:</td><td colspan="2">The customer logs in, the system retrieves the cart, checks item availability, processes payment, updates inventory, and sends confirmation.</td></tr>
<tr><td>Actors:</td><td colspan="2">• Customer</td></tr>
<tr><td>Related use case</td><td colspan="2">• Apply discount
• Make payment</td></tr>
<tr><td>Stakeholders:</td><td colspan="2">• Customer</td></tr>
<tr><td>Preconditions:</td><td colspan="2">• Customer has an account.
• Items are added to the shopping cart.</td></tr>
<tr><td>Post conditions:</td><td colspan="2">• Order is placed successfully, and confirmation is sent.
• Inventory is updated.
• If payment fails or items are unavailable, error messages are shown.</td></tr>
<tr><td rowspan="2">Flow of Activities:</td><td>Customer</td><td>System</td></tr>
<tr><td>1. Customer must login

2. Customer initiates checkout.

3. Customer enters payment details.

4. Customer receives confirmation.</td><td>1.1 System checks if customer is logged in.

2.1 System retrieves cart items.

2.2 System checks item availability.

2.3 System prompts for payment info.

3.1 System processes the payment.
3.2 system updates inventory.

3.3 System sends order confirmation.</td></tr>
<tr><td>Exception Conditions:</td><td colspan="2">1.1. customer is not authorized.

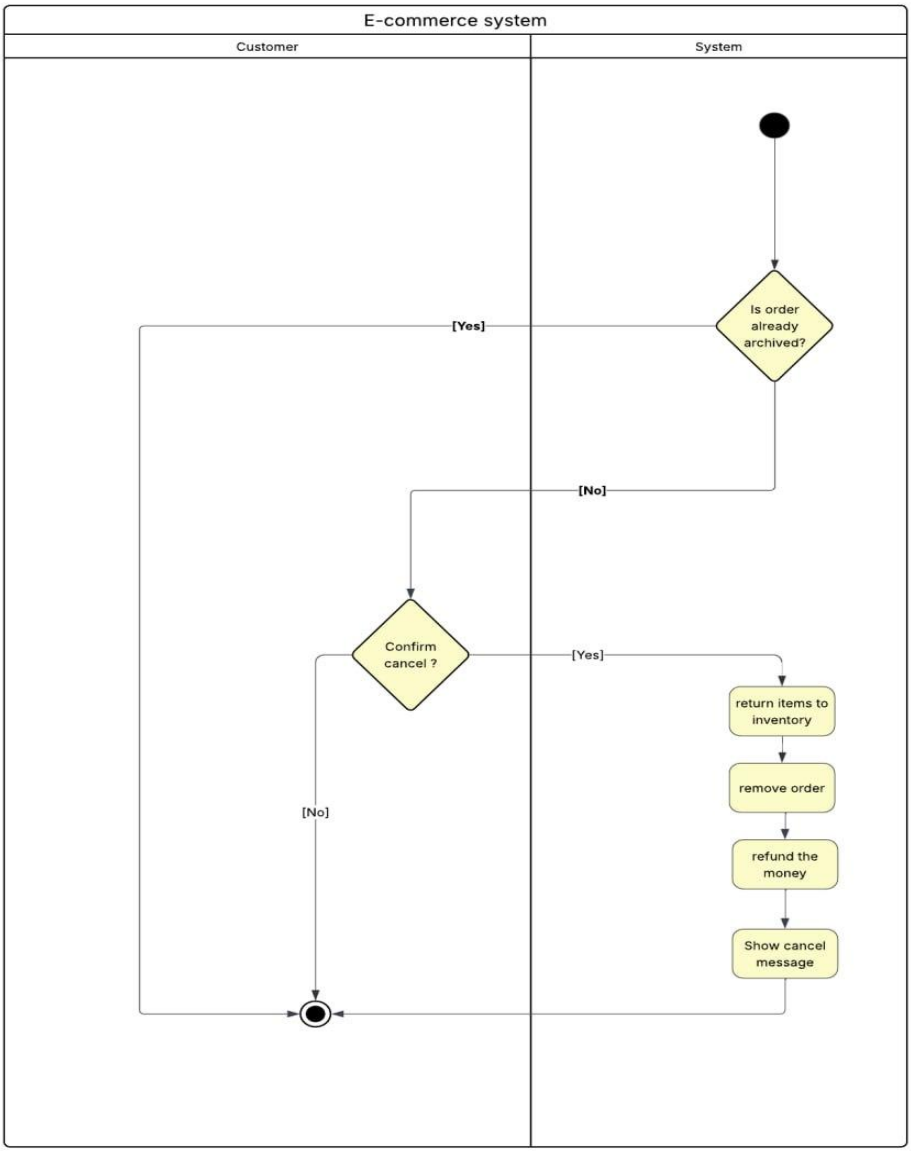
2.2 if the items aren’t available, then the system displays an error.

3.2 if the payment failed, then the system displays payment error.</td></tr>
</table>

Cancel order

customer can cancel the order if the order isn’t archived yet, if he confirms the cancellation the system will return the items to inventory, remove order, refund the money and show cancellation message to customer

Activity diagram



`

Scenario

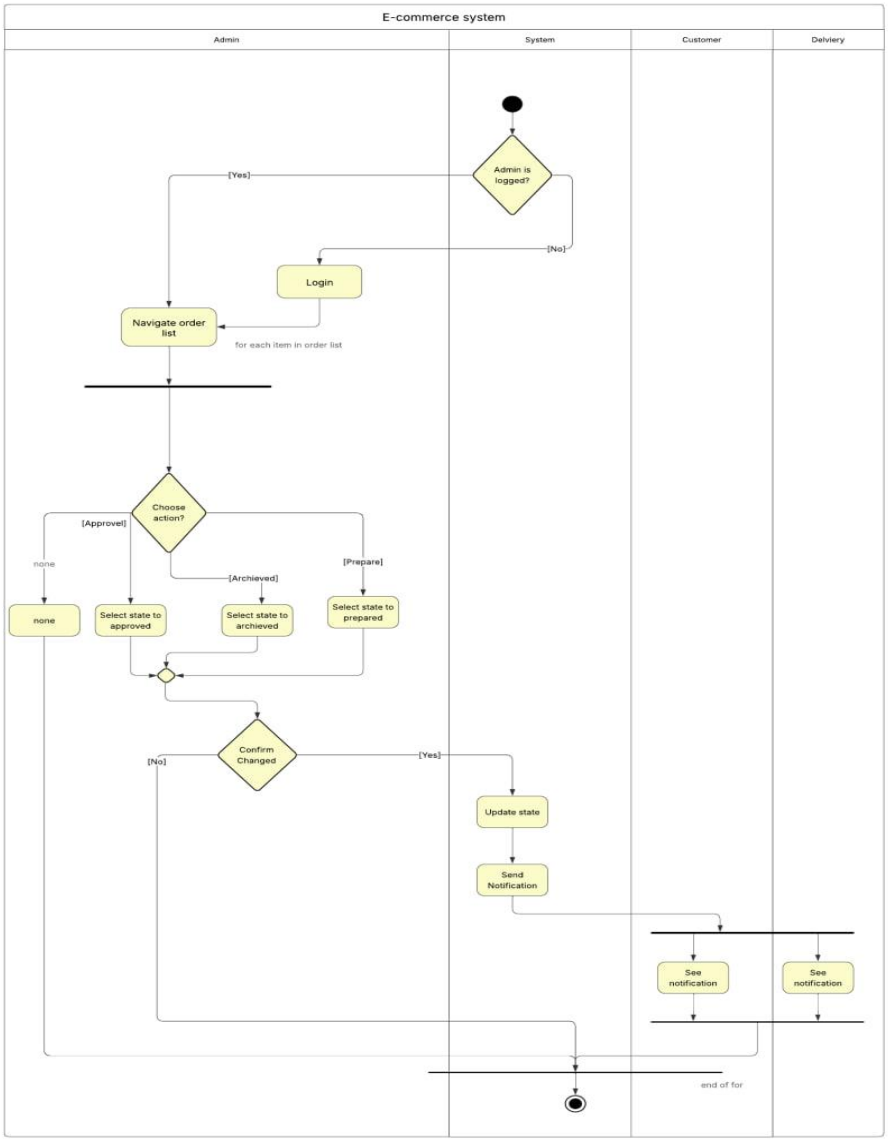
Use Case Name:	Cancel Order	
Scenario:	Customer Cancel Order	
Triggering Event:	• Customer initiates a cancellation request from their order history. • Order info	
Brief Description:	This use case describes how a customer cancels an existing order that has not yet been processed or shipped, using the eCommerce platform.	
Actors:	Customer	
Stakeholders:	Customer	
Related use Case	none	
Preconditions:	• The customer is logged in. • The order exists and is in a cancellable state (e.g., "Pending" or "Processing")	
Post conditions:	• The order status is updated to "Cancelled". • The inventory is adjusted accordingly. • A cancellation confirmation is sent to the customer. • If payment is already made, a refund process is initiated.	
Flow of Activities:	Admin	System
	1. Customer logs into their account. 2. Customer navigates to "My Orders 3. Customer selects an order to cancel. 4. Customer clicks the "Cancel Order" button. 5 Customer confirms the cancellation.	1.1 The system checks if the user is authorized and has access to modify 4.1 System checks if the order is cancelled. 5.1 System updates order status to "Cancelled". 5.2 System adjusts inventory. 5.3 System refund the money to customer 5.4 System sends cancellation confirmation.

Exception Conditions:	1.1 Login fails → prompt users to re-enter credentials 4.1 Order is not in a cancellable state (e.g., already shipped) → display error. 5.1 System update fails → display error message and retries 5.4 Refund process fails → notify support and customer.

Manage order

The admin can view, update, and control all customer orders in the system. This includes managing the entire order lifecycle from approve to prepare and achieved

Activity diagram



Scenario

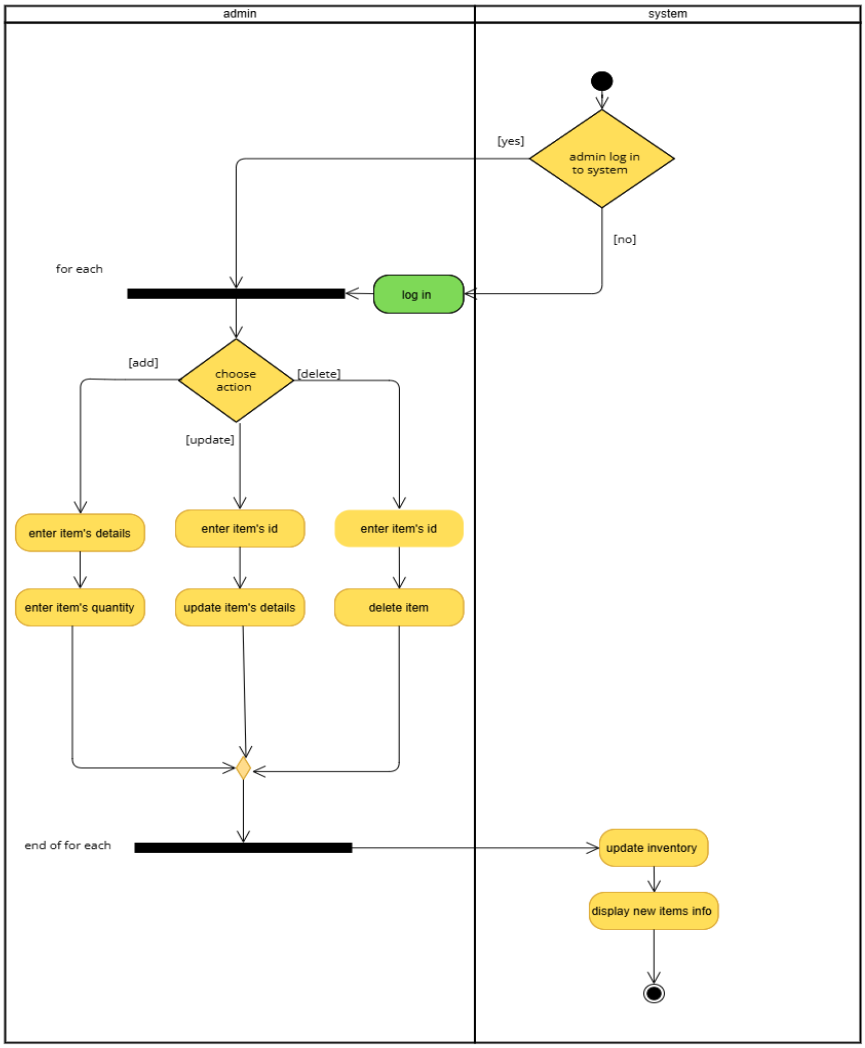
Use Case Name:	Manage Order State	
Scenario:	Admin Manage Order State	
Triggering Event:	Order Id Approve/ prepare/cancel/archive order request	
Brief Description:	This use case describes how an admin updates the status of a customer's order (e.g., "Approved", "Prepared" or, "Archived") through the admin panel of the eCommerce platform.	
Actors:	Admin	
Related use case	none	
Stakeholders:	Admin	
Preconditions:	• Admin is authenticated and logged into the system. • There is at least one existing order in the system	
Post conditions:	• The selected order's status is updated in the database. • A confirmation message is sent to the Admin. • Notification is sent to the Customer. • Notification is sent to the delivery	
Flow of Activities:	Admin	System
	1. Admin logs into the system. 2. Admin navigates to the "Orders Management" section. 3. for each of the orders in order Admin selects a specific order to modify. 4. The system displays the current order details and status. 5. Admin selects a new status from the available list. 6. Admin confirms the change.	1.1 The system checks if the user is authorized and has access to modify 5.1 The system updates the order status in the database. 6.1 System notifies the customer about the update. 6.2 System sends confirmation to the admin.

Exception Conditions:	1.2	Admin is not authorized → show access denied message
	3	Order does not exist or is locked → show error.
	5	Invalid statuses selected → prompt admin to select a valid one
	5.1	Database update fails → show system error.
	6.1	Notification system unavailable → log issue for retry.

Update inventory

Admin can update inventory by adding new item to the system, update detailed of item or removing items from the system.

Activity diagram



Scenario of add item

<table>
<tr><td>Use Case Name:</td><td colspan="2">update inventory</td></tr>
<tr><td>Scenario:</td><td colspan="2">add an item in the inventory system.</td></tr>
<tr><td>Triggering Event:</td><td colspan="2">An administrator needs to add item to the inventory</td></tr>
<tr><td>Brief Description:</td><td colspan="2">Admin interacts with the system to manage the inventory. The admin can add new products by entering item details and quantity. Once operation is completed, the inventory is updated and reflected in the system.</td></tr>
<tr><td>Actors:</td><td colspan="2">Admin</td></tr>
<tr><td>Related use case</td><td colspan="2">None.</td></tr>
<tr><td>Stakeholders:</td><td colspan="2">• Admin
• Customer</td></tr>
<tr><td>Preconditions:</td><td colspan="2">• Admin must be logged into the system with appropriate access rights.
• The admin must be have access to the inventory database.</td></tr>
<tr><td>Post conditions:</td><td colspan="2">• Inventory database is updated to reflect the added, modified, or removed items.
• New item information is visible in the system and available to customers if applicable.</td></tr>
<tr><td rowspan="2">Flow of Activities:</td><td>customer</td><td>System</td></tr>
<tr><td>1. Admin login into system

2. Admin chooses "Manage Inventory."

3. Admin selects an add

4. Admin enters new item details and quantity.</td><td>1.3 The system checks if the user is authorized and has access to modify

2.1 System displays inventory management options (add, update, delete).

4.1 System confirms the updates

4.2 System displays the new inventory</td></tr>
<tr><td>Exception Conditions:</td><td colspan="2">1.2 customer is not authorized
4.1 system doesn’t have extra memory to add new items</td></tr>
</table>

Scenario of delete item

<table>
<tr><td>Use Case Name:</td><td colspan="2">update inventory</td></tr>
<tr><td>Scenario:</td><td colspan="2">delete an item from the inventory system.</td></tr>
<tr><td>Triggering Event:</td><td colspan="2">An administrator needs to delete an item from the inventory</td></tr>
<tr><td>Brief Description:</td><td colspan="2">Admin interacts with the system to manage the inventory. The admin can delete products by entering item id and delete it . Once operation is completed, the inventory is updated and reflected in the system.</td></tr>
<tr><td>Actors:</td><td colspan="2">Admin</td></tr>
<tr><td>Related use case</td><td colspan="2">None.</td></tr>
<tr><td>Stakeholders:</td><td colspan="2">• Admin
• Customer</td></tr>
<tr><td>Preconditions:</td><td colspan="2">• Admin must be logged into the system with appropriate access rights.
• The admin must have access to the inventory database.</td></tr>
<tr><td>Post conditions:</td><td colspan="2">• Inventory database is updated to reflect the added, modified, or removed items.
• New items’ information is visible in the system and available to customers if applicable.</td></tr>
<tr><td rowspan="2">Flow of Activities:</td><td>customer</td><td>System</td></tr>
<tr><td>1. admin login into system

2. Admin chooses "Manage Inventory."

3.. Admin selects delete option

4.Admin enters the item ID

5.Admin remove item.</td><td>1.4 The system checks if the user is authorized and has access to modify

2.1 System displays inventory management options (add, update, delete).

4.1 system check if it is valid

5.1 System confirms the updates

5.2 System displays the new inventory.</td></tr>
<tr><td>Exception Conditions:</td><td colspan="2">1.1 customer is not authorized
5.1 item does not exist in the system</td></tr>
</table>

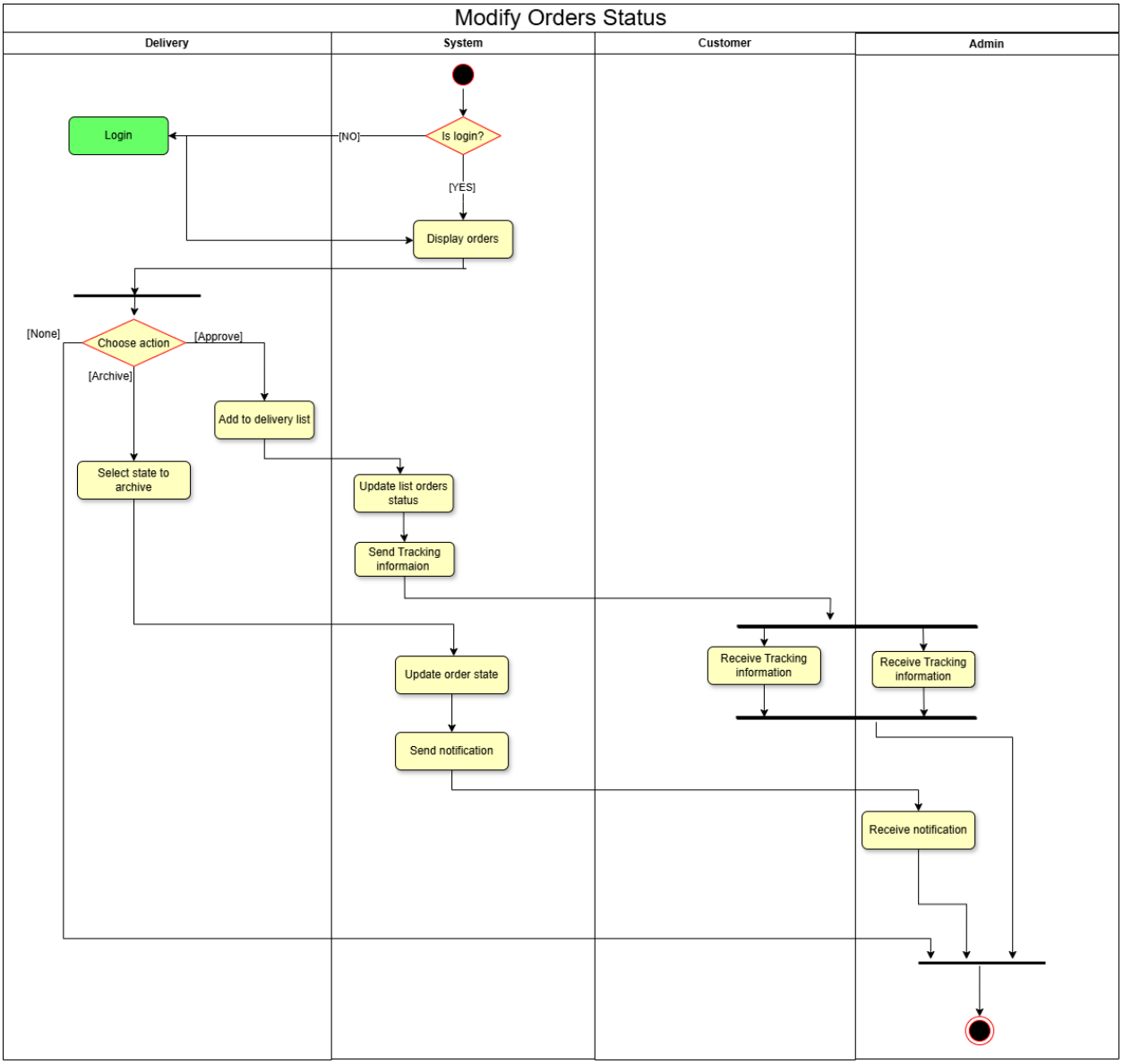
Scenario of update item

Use Case Name:	update inventory	
Scenario:	Update an item in the inventory system.	
Triggering Event:	An administrator needs to update item to the inventory	
Brief Description:	Admin interacts with the system to manage the inventory. The admin can update products by entering item new details and quantity. Once operation is completed, the inventory is updated and reflected in the system.	
Actors:	Admin	
Related use case	None.	
Stakeholders:	• Admin • Customer	
Preconditions:	• Admin must be logged into the system with appropriate access rights. • The admin must be have access to the inventory database.	
Post conditions:	• Inventory database is updated to reflect the added, modified, or removed items. • New item information is visible in the system and available for customers if applicable.	
Flow of Activities:	customer	System
	1.admin logs into system	1.5 The system checks if the user is authorized and has access to modify
	2.Admin chooses "Manage Inventory."	2.1 System displays inventory management options (add, update, delete).
	3. Admin selects update item	
	4.Admin enters the item ID	4.1 system check if id is valid
	5.Admin updates the details	5.1 System confirms the updates
		5.2 System displays the new inventory .
Exception Conditions:	1.1 customer is not authorized 4.1 item is not in the system	

Modify order state

delivery have a list or orders and can modify order stat by appropriate some orders, sends tracking information to admin and customer then send order to customer

Activity diagram



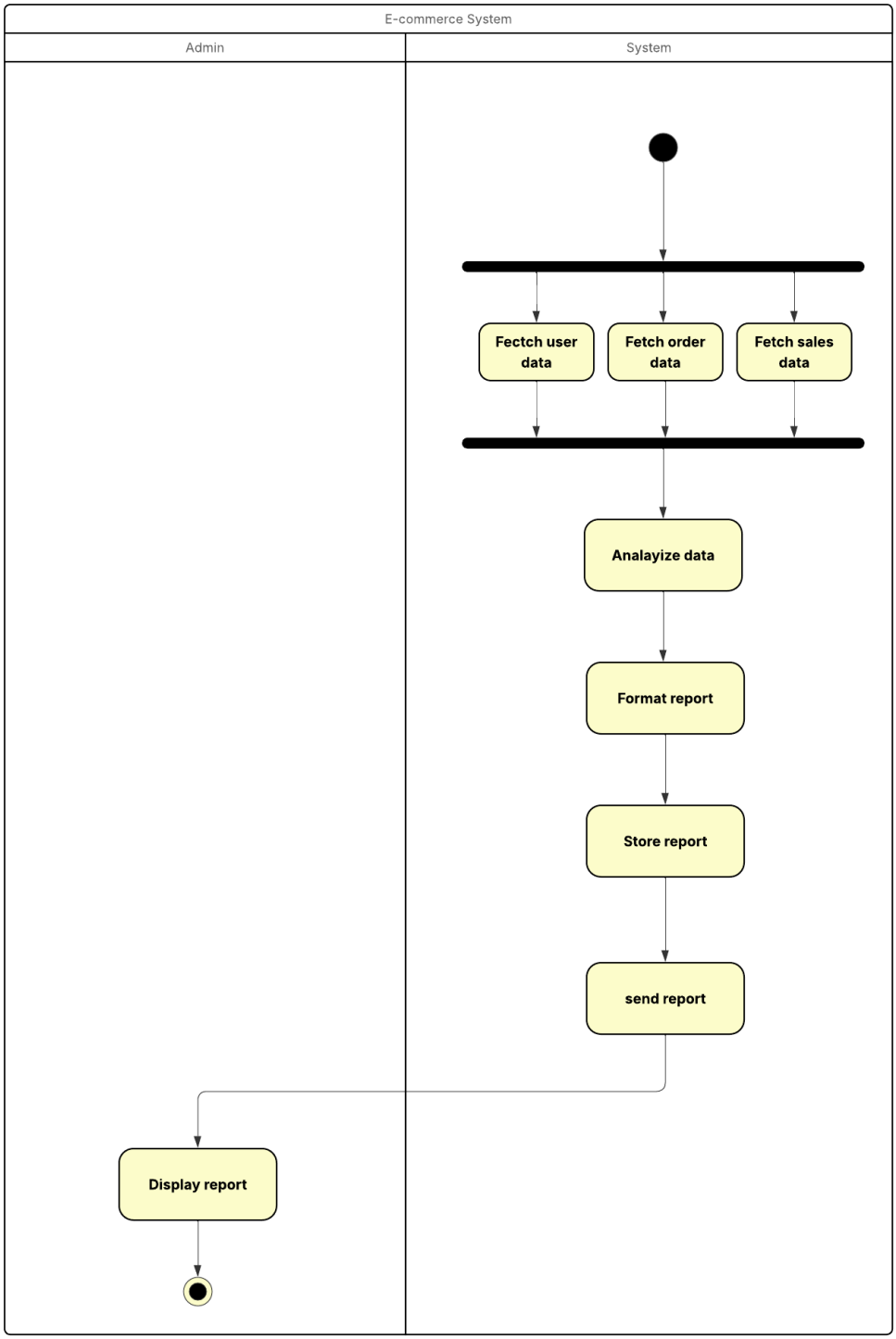
Scenario

Use Case Name:	Modify orders status	
Scenario:	A delivery staff member modify order status.	
Triggering Event:	A delivery staff member accesses the system to update the status of new or in-progress orders.	
Brief Description:	The system ensures the delivery user is logged in, then shows the list of orders. The delivery staff selects appropriate orders and updates their status. The system confirms and records the update.	
Actors:	- Delivery	
Related use case	None .	
Stakeholders:	- Delivery - Customer	
Preconditions:	- Delivery staff is registered and has access credentials.	
Post conditions:	- Order status is updated in the system. - A confirmation message is sent to relevant parties.	
Flow of Activities:	**Delivery**	**System**
	1. Delivery must login 2. Delivery evaluates each order. 3. Delivery adds orders to delivery list.	1.1 System checks if customer is logged in. 1.2 System displays list of orders. 3.1 System updates order status. 3.2 System sends a confirmation message.
Exception Conditions:	1.1 delivery is not authorized 1.2 If there is a system error occurs during status update, then rollback or retry.	

Generate report

system generates daily report for admin activity diagram of Generate report

Activity diagram



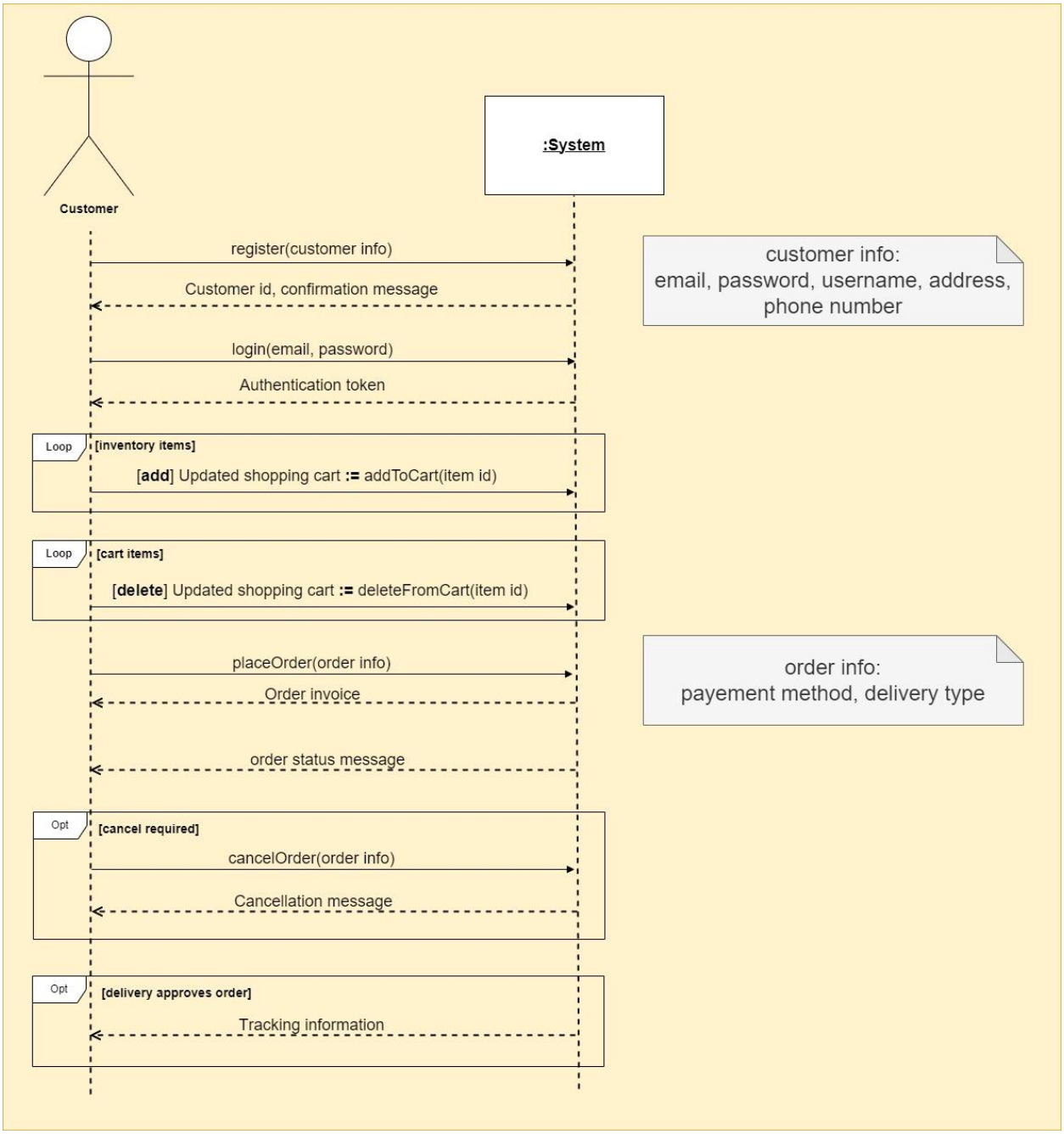
Scenario

Use Case Name:	**Generate Delayed Report**	
Scenario:	**System Automatically or Manually Generates Delayed Report**	
Triggering Event:	**1. End of day** **2. Requests report generation**	
Brief Description:	**The system fetches delayed orders, user data, and sales metrics, compiles them into a structured report (PDF, Word, etc.), stores it, and allows the admin to view/download it.**	
Actors:	**Admin (manual trigger**	
Related use case	**None**	
Stakeholders:	**Admin, Management Team**	
Preconditions:	• **System is connected to the database** • **Orders exist in the system (optional for empty reports)** • **Admin is authenticated (for manual triggers).**	
Post conditions:	• **Report is generated, formatted, and stored.** • **Admin can access the report via the admin panel** • **Notification sent to admin (optional).**	
Flow of Activities:	**Admin**	**System**
	5.1 View/download report from admin panel	**1. Fetch Data** **2. Generate Report** **3. format report** **4. Store Report** **5. send report**
Exception Conditions:	**1.3 No delayed orders found → Generate empty report or skip.** **1.1 Database connection fails → Retry 3 times, then log error.** **1.2 Formatting fails (e.g., PDF lib error) → Fallback to CSV.** **1.3 Storage full → Alert admin and abort.**	

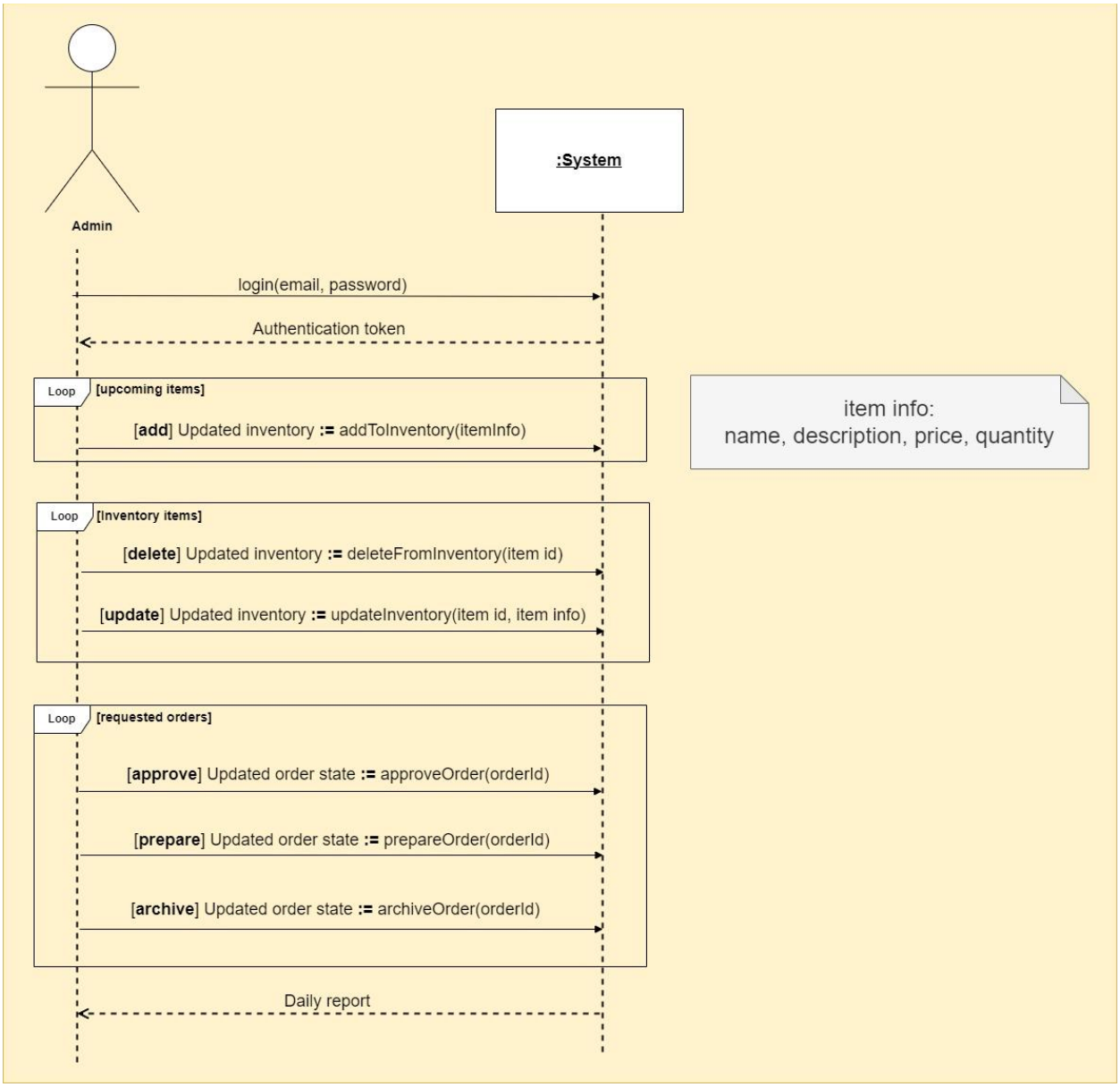
Sequence diagram

A sequence diagram is a type of UML (Unified Modeling Language) diagram that shows how objects interact in a particular scenario of a system. It focuses on the sequence of messages exchanged between objects over time to accomplish a specific functionality.

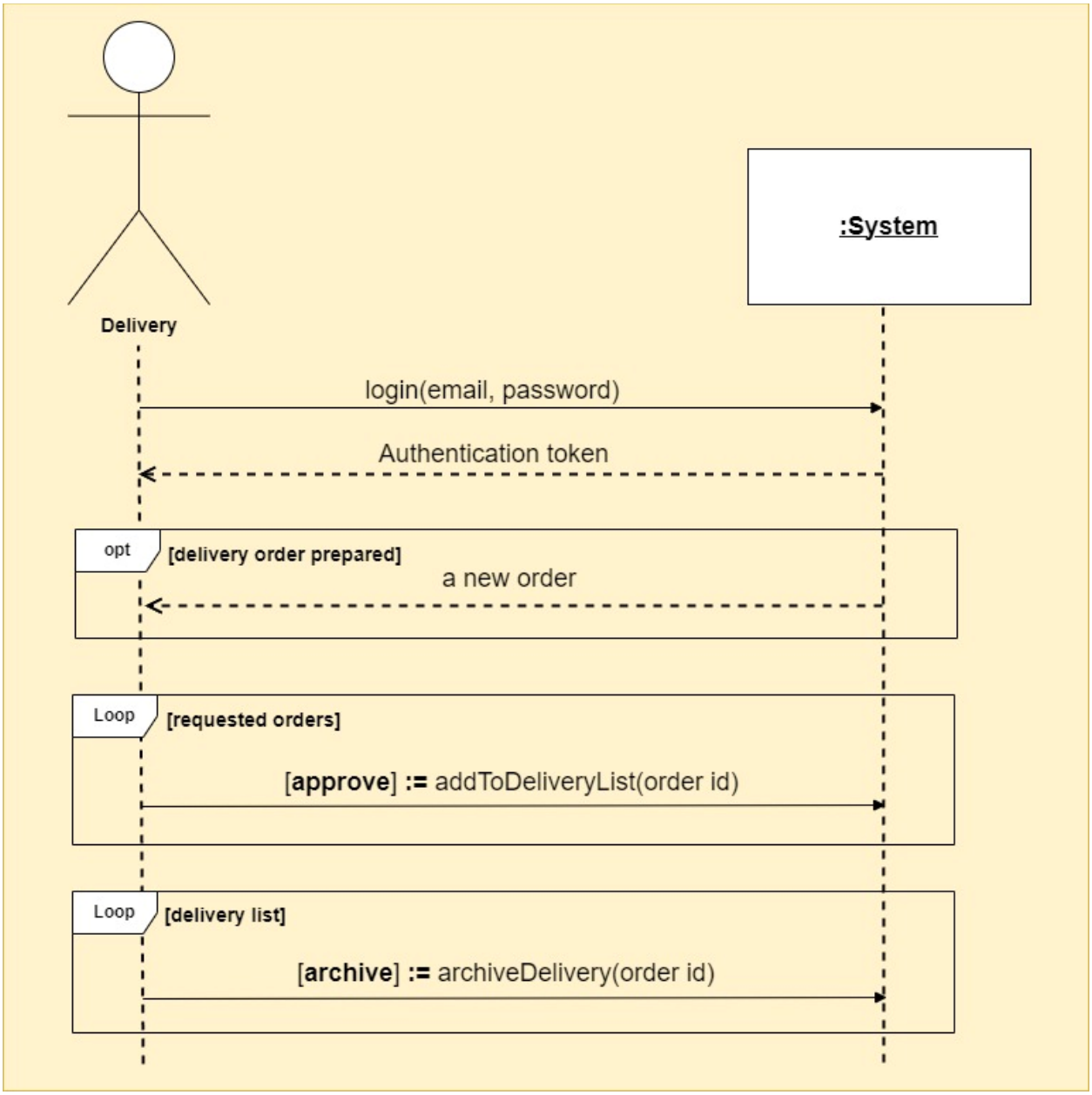
Customer sequence diagram



Admin sequence diagram



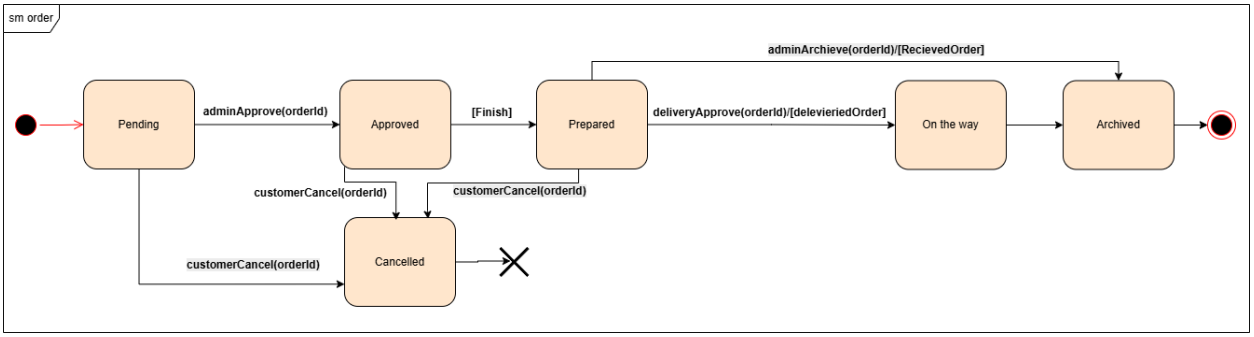
Delivery sequence diagram



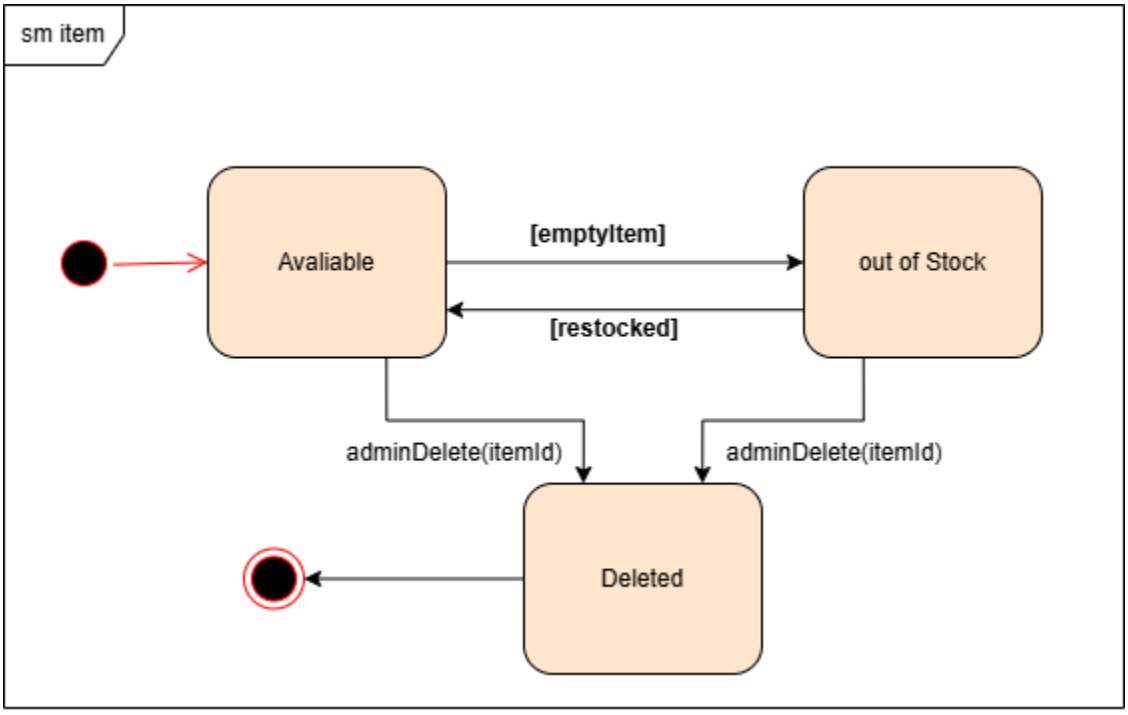
State machine diagram

A State Machine Diagram (also called a State Diagram) is a type of UML diagram that shows the different states an object can be in and how it transitions from one state to another based on events or conditions.

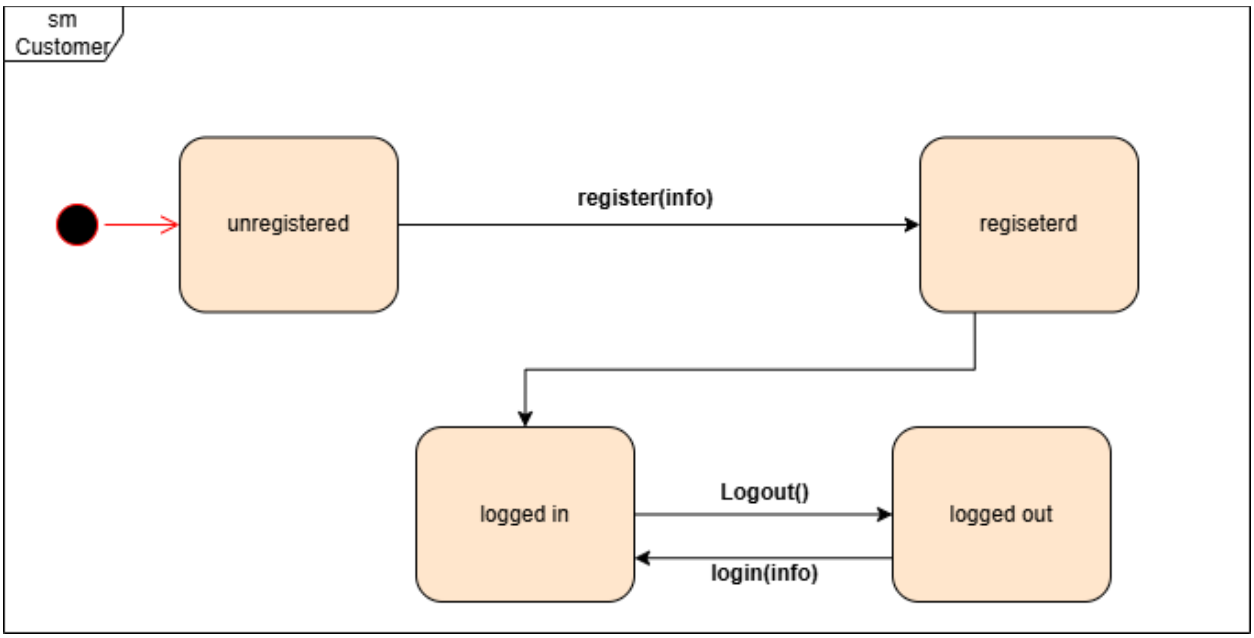
State of order



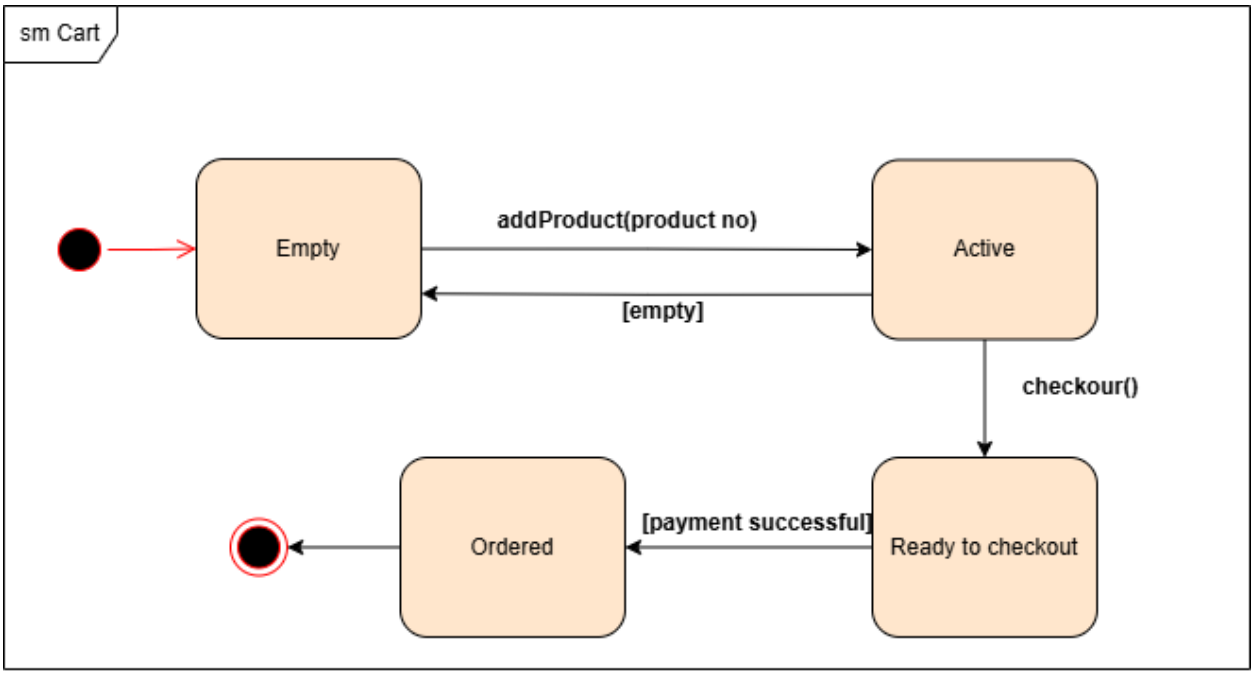
State of Item



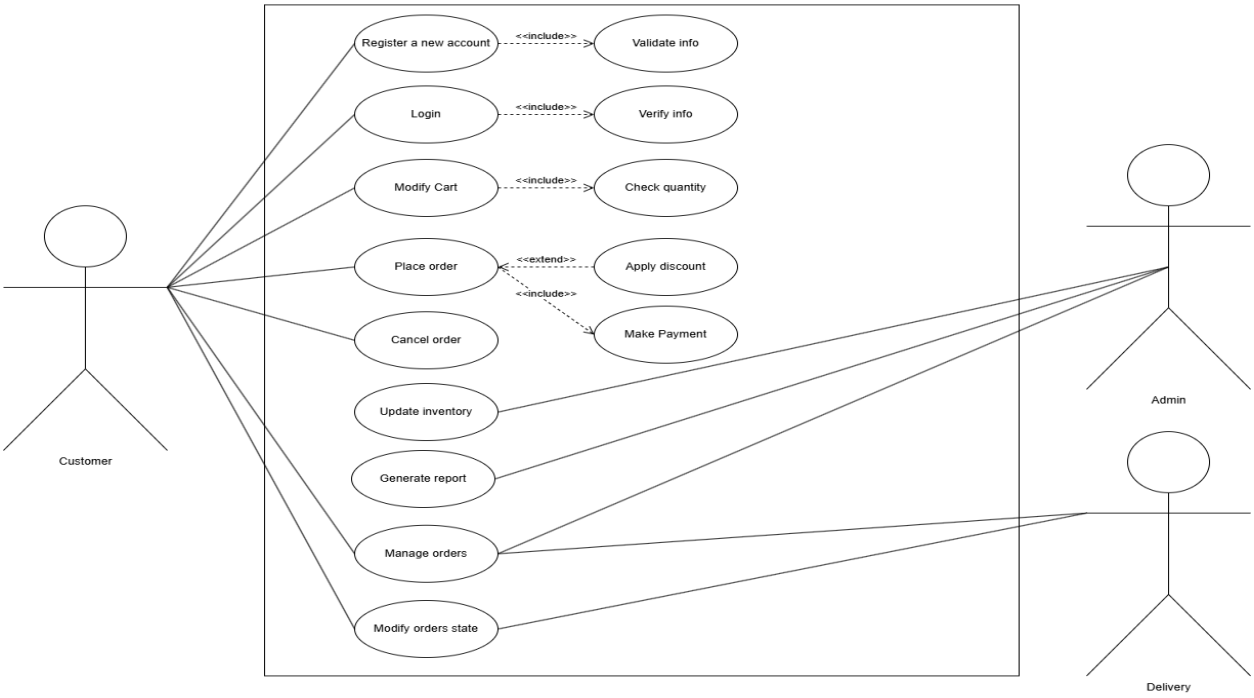
State of Customer



State of cart

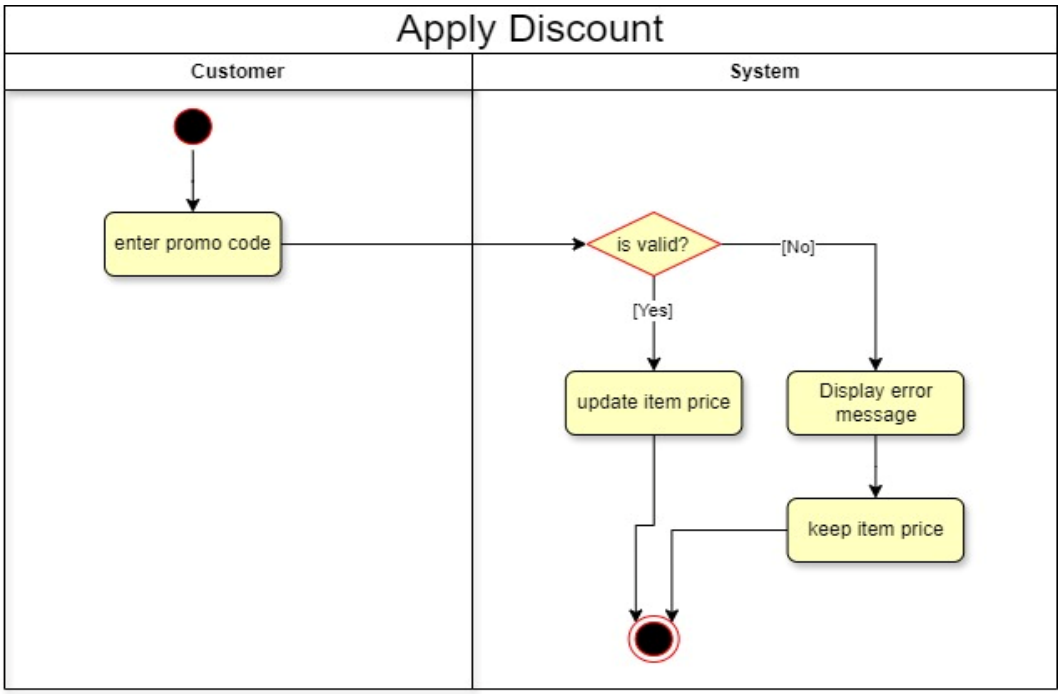


Use case diagram

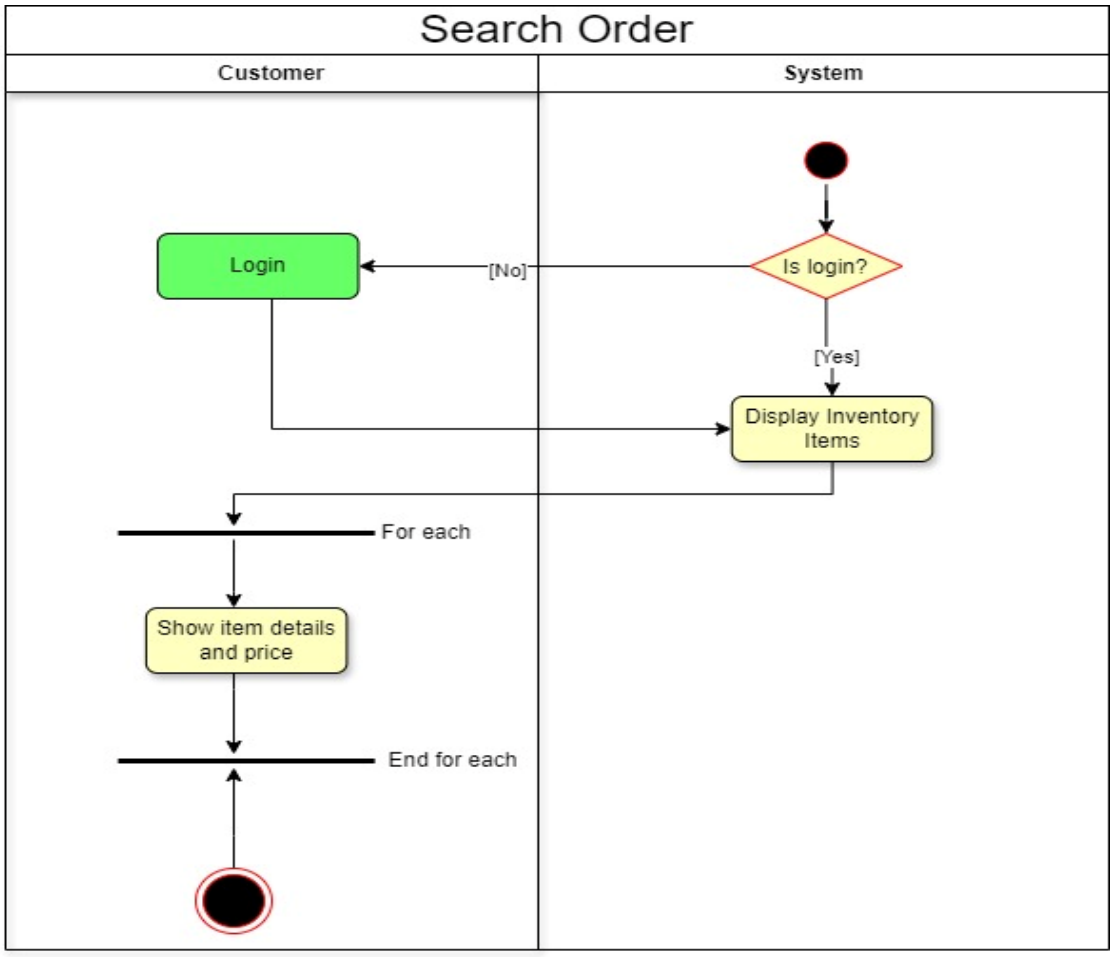


Extended and included activity diagrams

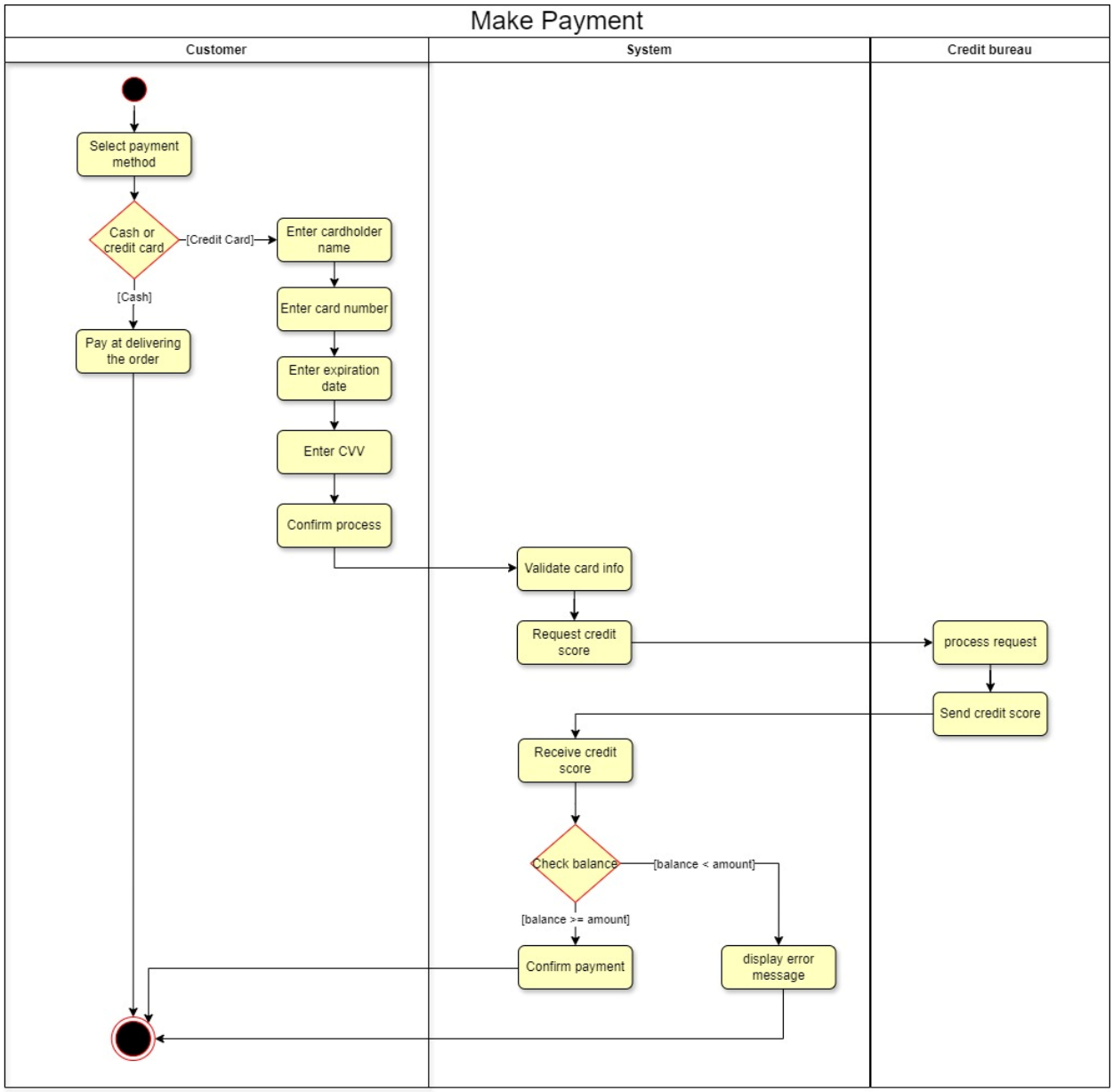
Apply discount



Search order



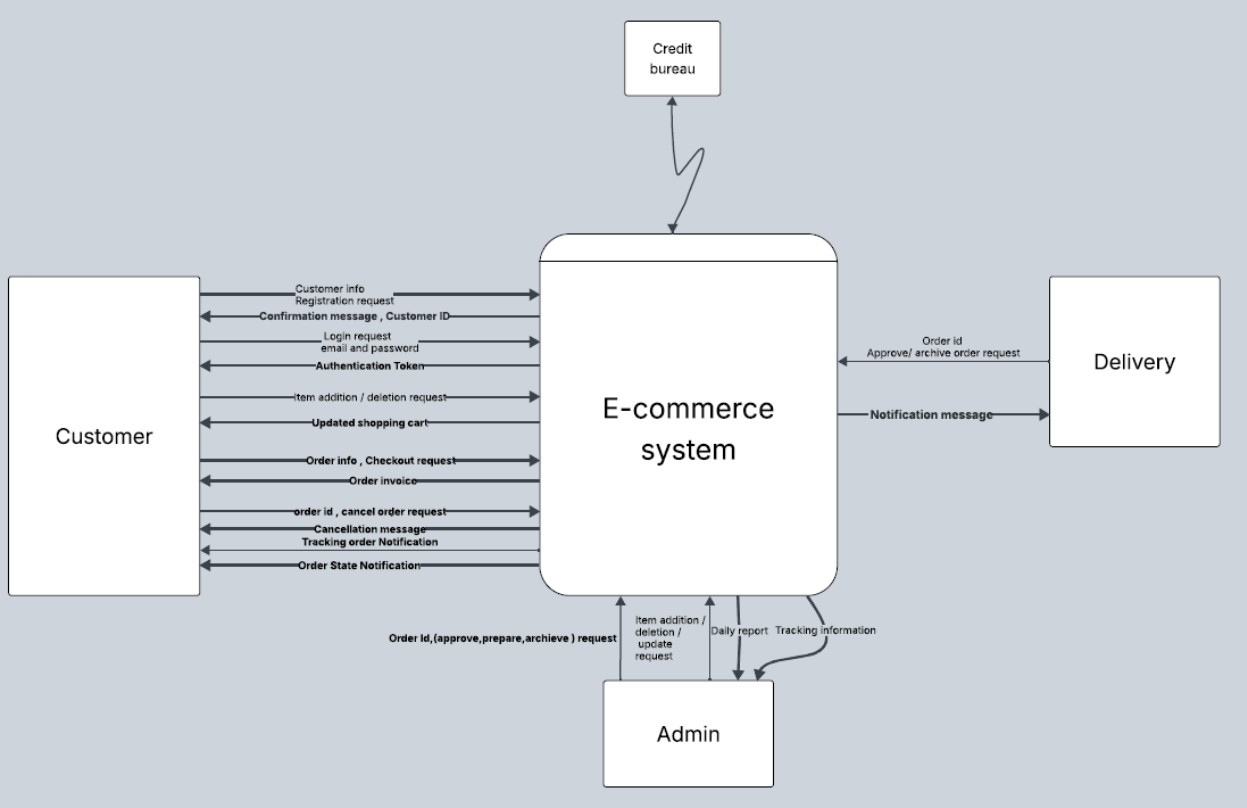
Make payment



Data flow diagram (DFD)

Context diagram

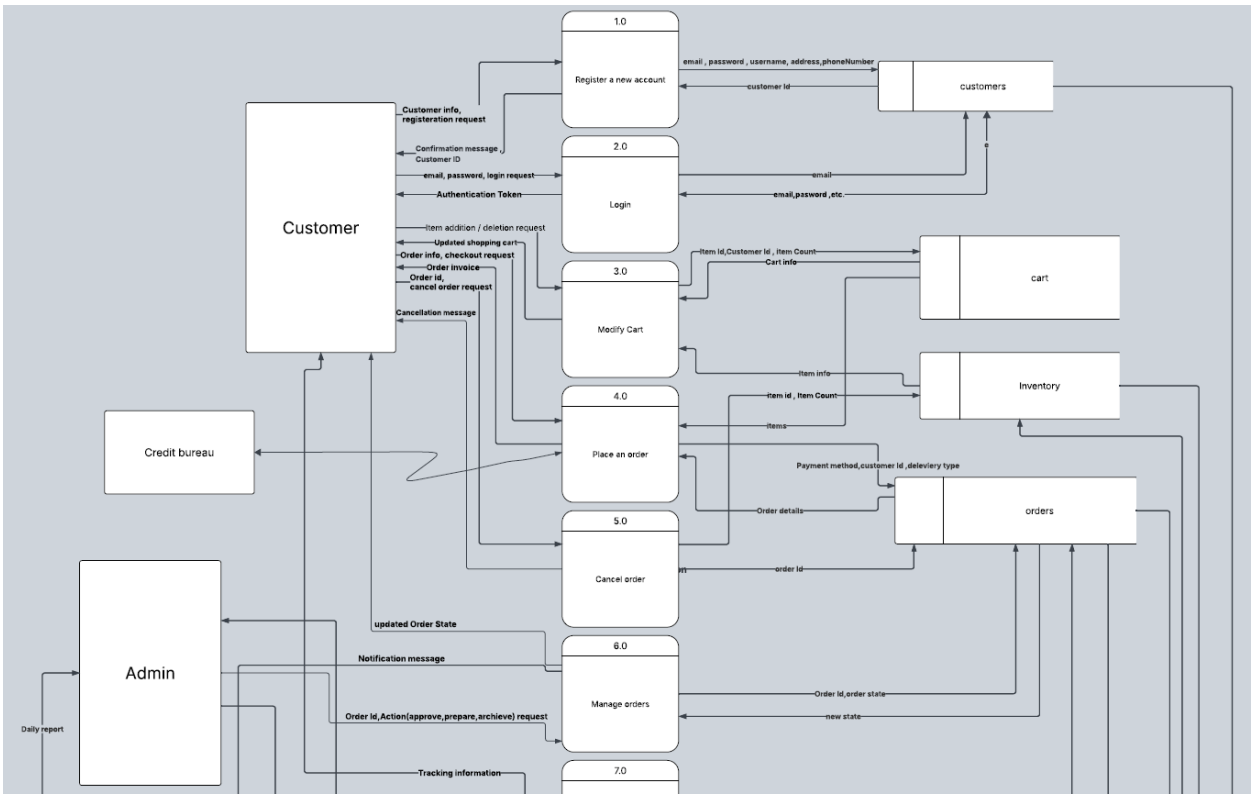
A **context diagram** is a high-level, visual representation of a system and how it interacts with external entities (such as users, other systems, or organizations). It shows the system as a single process and its data exchanges (inputs and outputs) with external entities, without diving into internal details

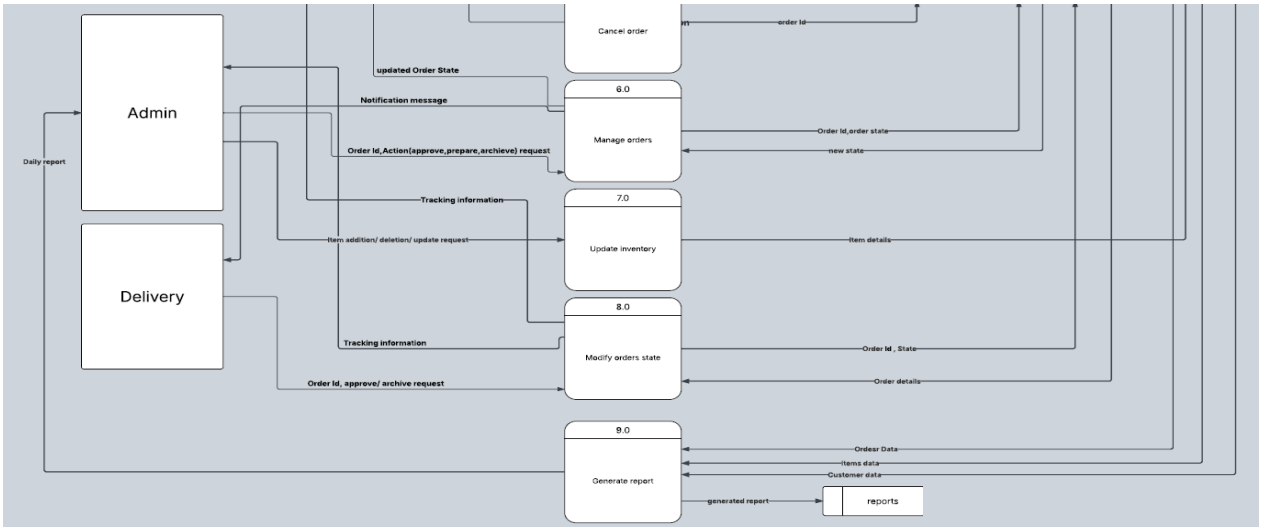


Overview diagram

An overview diagram is a high-level visual representation of a system that provides a broad understanding of its major components and how they interact. It's similar to a block diagram or architecture diagram and is often used early in system design to give a clear picture of:

1. The main modules or components
2. How they connect or interact





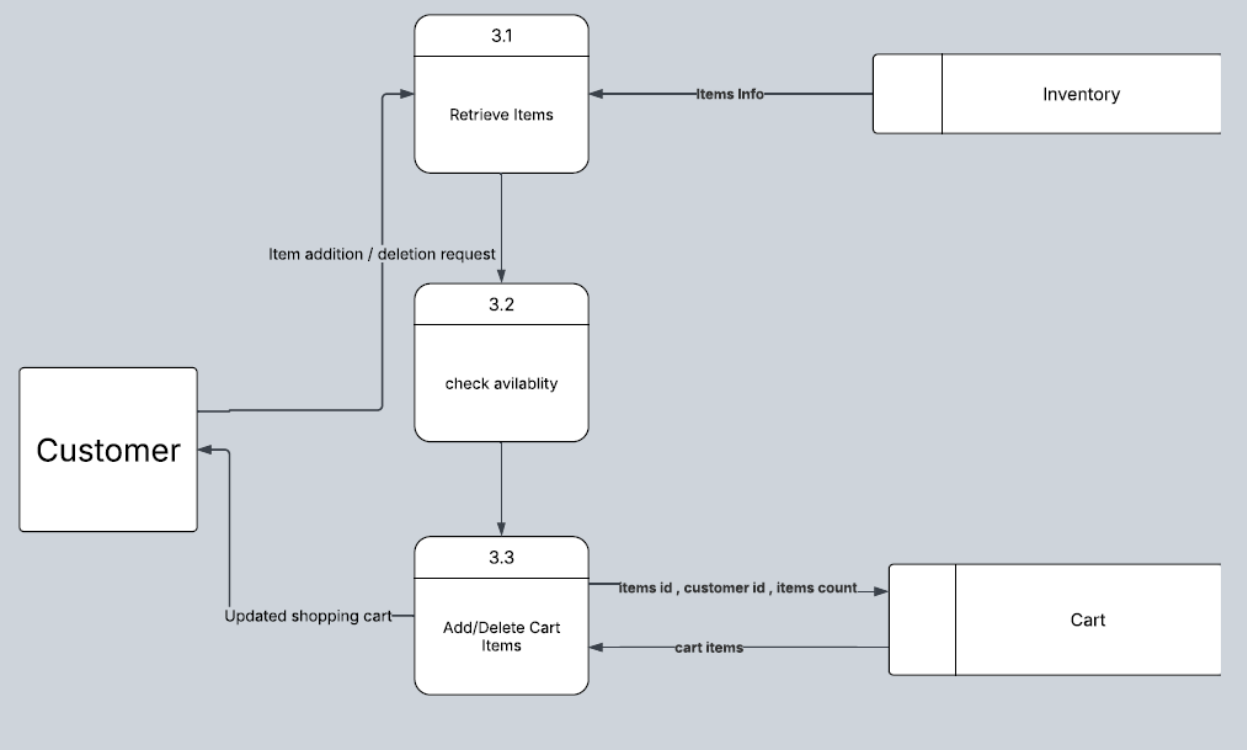
Detailed diagram

A detailed diagram provides a deeper view into how a system works internally. Unlike a context diagram, which shows only the system boundaries and external interactions, a detailed diagram breaks the system into multiple internal processes, data stores, and interactions.

Modify cart

The Cart detailed diagram illustrates the internal structure and relationships of the Cart module within the system. It defines the key entities, their attributes, and interactions required to manage user-selected items before order placement.

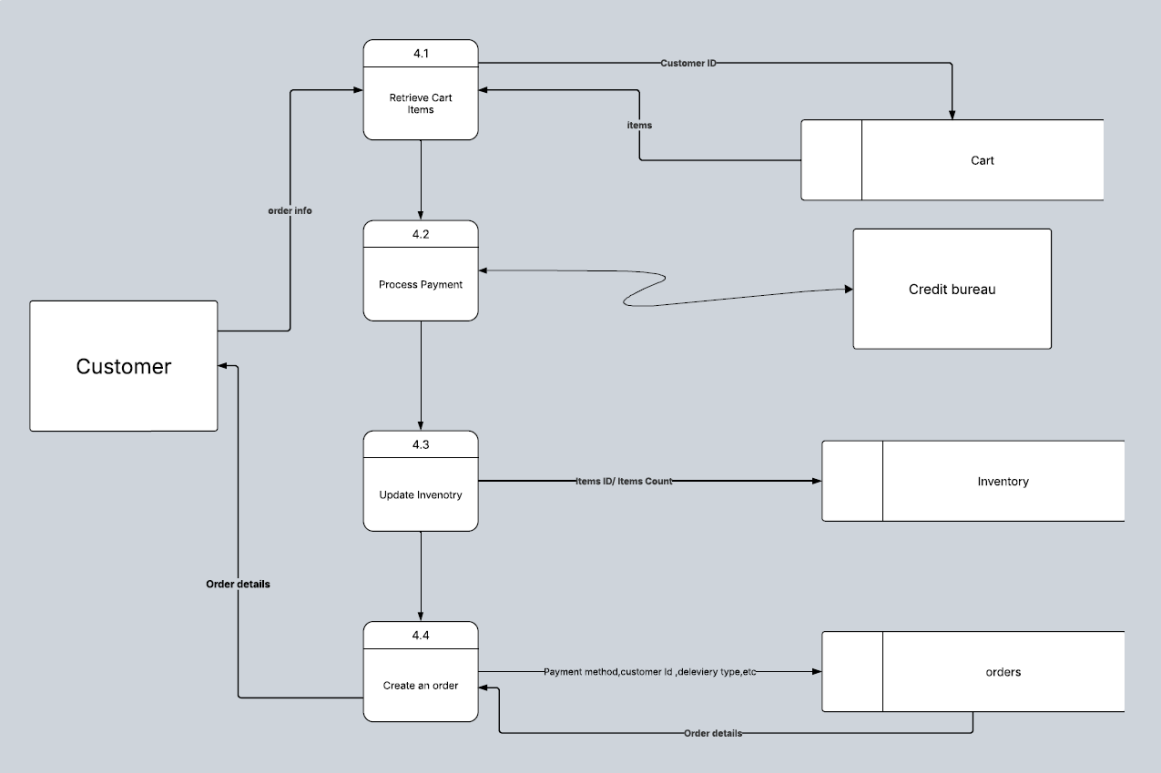
At the core is the Cart entity, which maintains a reference to the user and contains one or more CartItem records. Each CartItem is linked to a specific product or menu item and tracks quantity, price, and any applicable discounts. The diagram also shows associations with the User and Product (or MenuItem) entities, establishing the necessary data relationships for maintaining cart contents.



Place an order

The **Place an Order** detailed diagram represents the structure and workflow involved when a user proceeds to finalize their cart and creates an order in the system. This process transitions the cart contents into a persistent Order entity, initiating the fulfillment process.

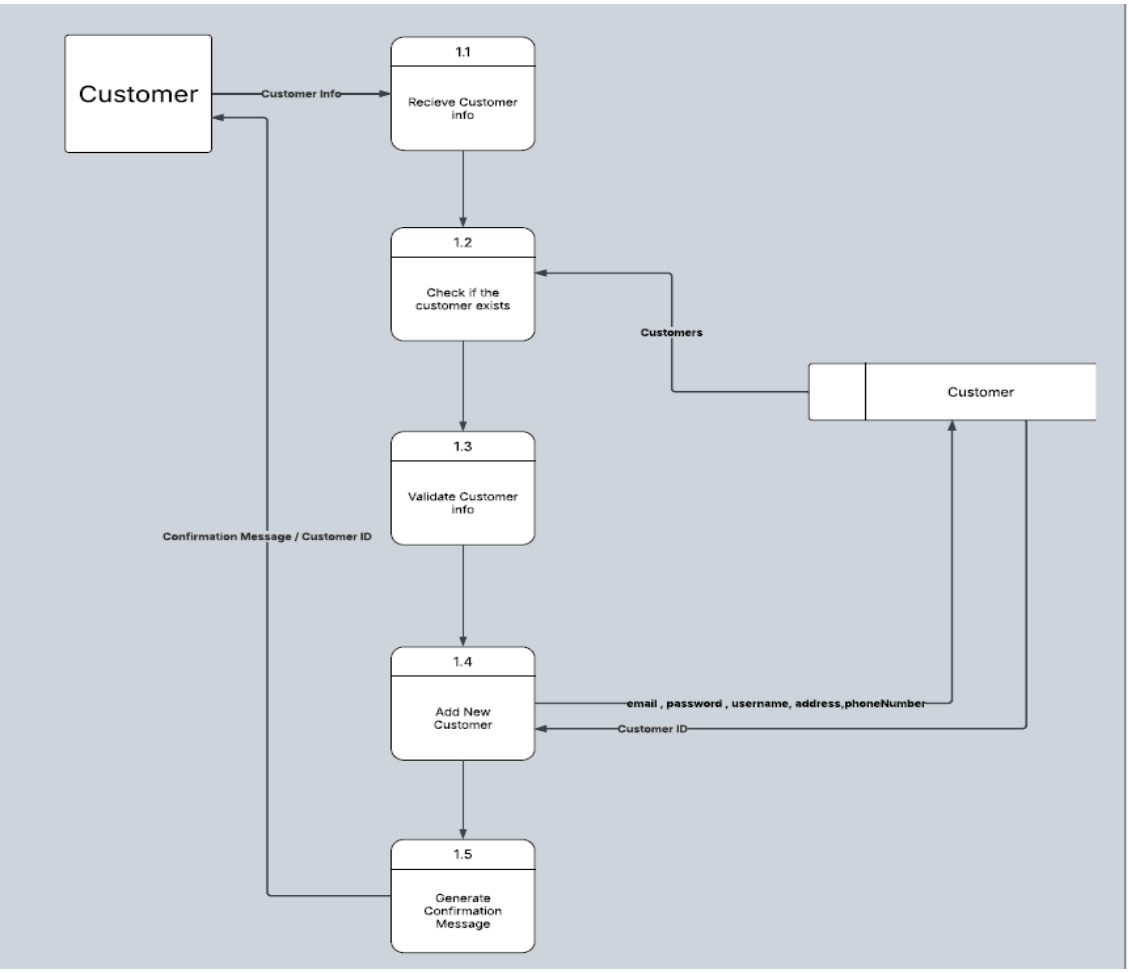
The diagram outlines how the system gathers information from the Cart, User, and Payment components to construct a complete Order. Each Order includes one or more OrderItem entities, which reflect the individual items selected by the user at the time of checkout. The diagram also shows associations with Address (for delivery), PaymentMethod, and OrderStatus, providing a comprehensive view of all data points required for processing the order.



Register

The **Register** detailed diagram models the structure and interactions involved in the user registration process. This process enables new users to create an account within the system, providing necessary personal and authentication details.

The diagram highlights the interaction between the RegistrationForm, User, and supporting components such as ValidationService and UserDatabase. Upon form submission, the system validates user input (e.g., name, email, password), checks for uniqueness (e.g., email not already used), and securely stores the user's credentials and profile data.

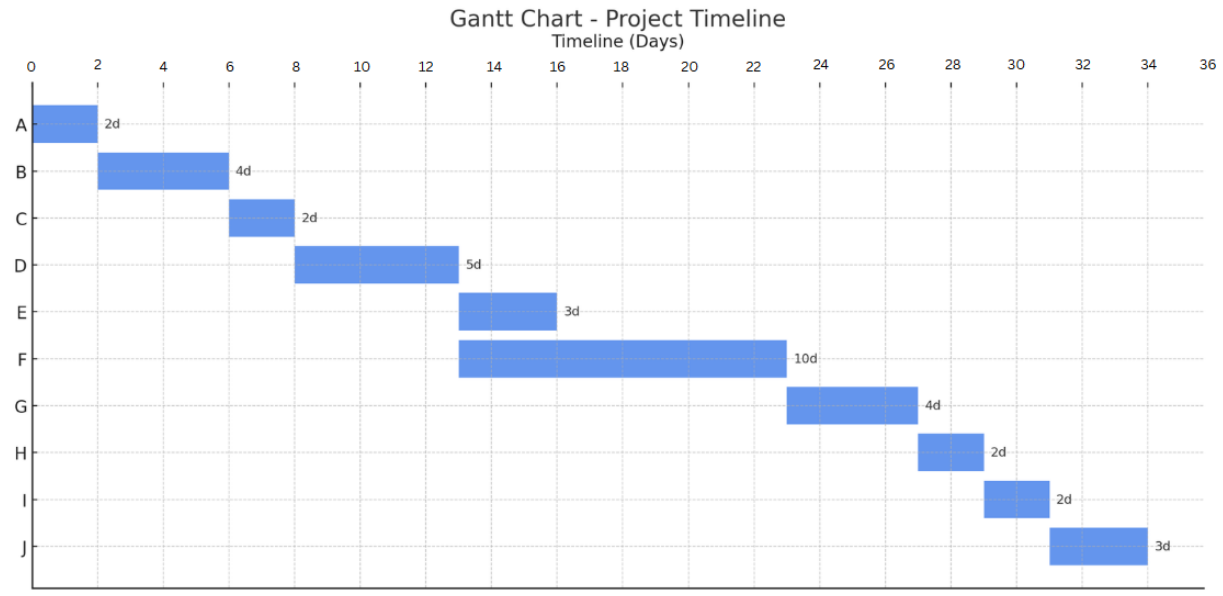


Gant chart

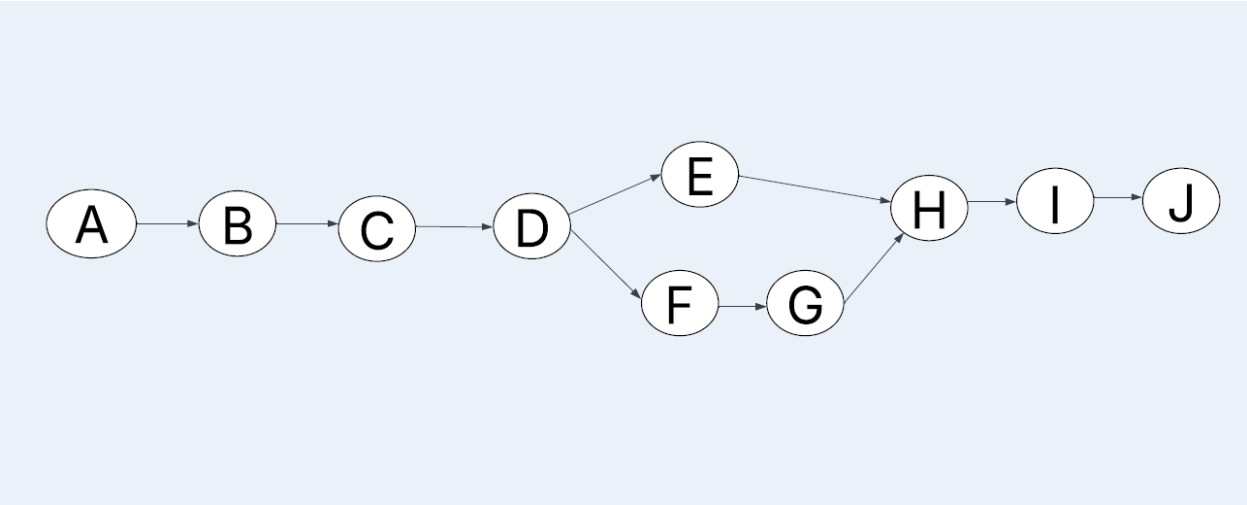
Project Task table

Task ID	Task Name	Duration (Days)	Dependency
A	Identify system requirements and events	1	—
B	Create the event table	1	A
C	Draw the Use Case Diagram	1	B
D	Draw activity diagrams and write use case descriptions	4	C
E	Draw the sequence diagram	1	B
F	Draw the context diagram	1	B
G	Draw the overview diagram	3	D
H	Draw detailed diagrams for each process	2	G

Gantt chart



Network diagram



Critical path → A-B-C-D-F-G-H-I-J

Conclusion

In conclusion, the e-commerce sector continues to reshape the landscape of modern retail, offering unprecedented convenience, scalability, and global reach. This report has examined the critical components of successful e-commerce operations, including website functionality, user experience, payment systems, logistics, and digital marketing strategies. The findings highlight the importance of adopting a customer-centric approach, leveraging data analytics, and maintaining robust security measures to stay competitive. As technology advances and consumer behavior evolves, businesses must remain agile and innovative to capitalize on new opportunities in the digital marketplace.