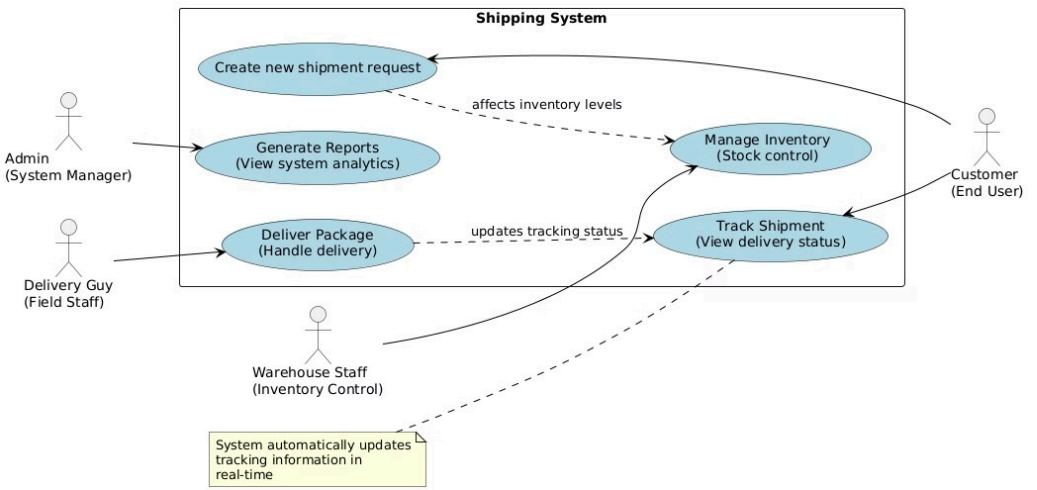
Online Shipment Tracking System: A Software Design Presentation

This presentation showcases the essential diagrams for the **Online Shipment Tracking System**. These visualizations, including the **Use Case**, **Activity**, and **Sequence** diagrams, guide the development process and ensure the best customer experience.

By : OverClocked Team

Meet OverClocked Team

- Abdelrehman E. Fouad
- A'laa Elnageh
- Dai Yasser
- Amina Mohamed
- Nahla Ahmed
- Marwan Mohamed



Business Use Case

A Business Use Case Diagram illustrates how various actors interact with the system, highlighting its key functionalities and purpose. It helps visualize roles and processes for better understanding.

Business Use Case Diagram

Actors

The actors in the diagram are the different roles who interact with the system.

- Admin
- Delivery Guy
- Warehouse Staff
- Customer

Use Cases

The use cases represent the different functions of the system.

- Shipment requests
- Tracking
- Reporting
- Deliveries
- Inventory management

Interactions

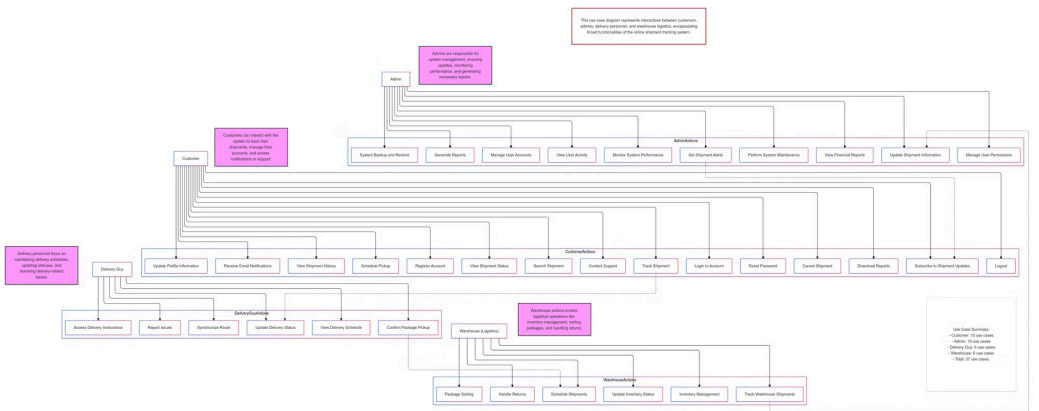
The diagram shows how the actors interact with the system to perform the use cases.

- Delivery update tracking
- Shipment requests
- Inventory management

System Goals

The use case diagram helps to understand the overall goals of the system.

- Real-time tracking
- Efficient delivery
- Stock control



Use Case Diagram

A **Use Case Diagram** is a visual representation of the interactions between users (actors) and the system. It outlines the system's functionalities and helps in understanding the relationship between various roles and use cases.

Key Points of the Use Case Diagram

Customer Actions

Customers have 15 use cases, including searching and tracking shipments, viewing shipment history, managing accounts, and subscribing to updates.

Admin Responsibilities

Admins manage the system with 10 use cases, such as updating shipment information, monitoring performance, managing user accounts, generating reports, and setting shipment alerts.

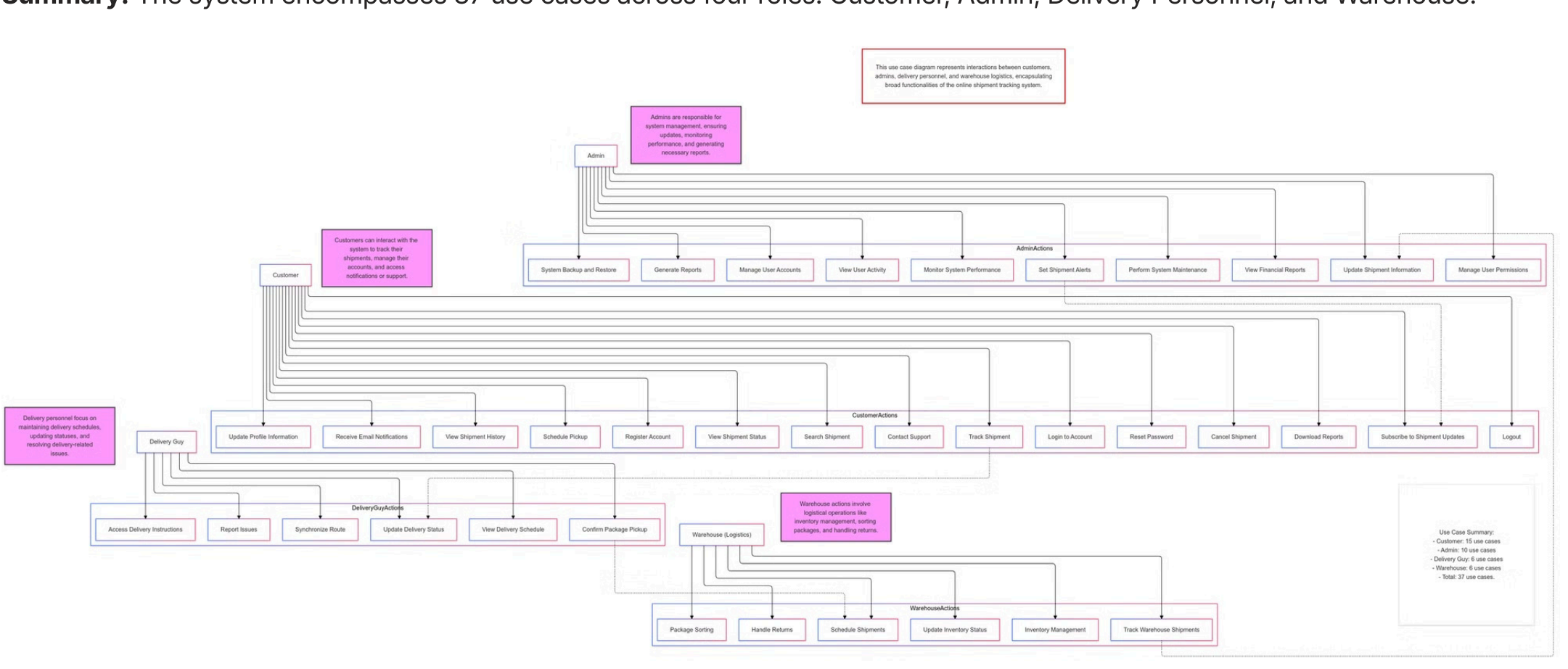
Delivery Personnel

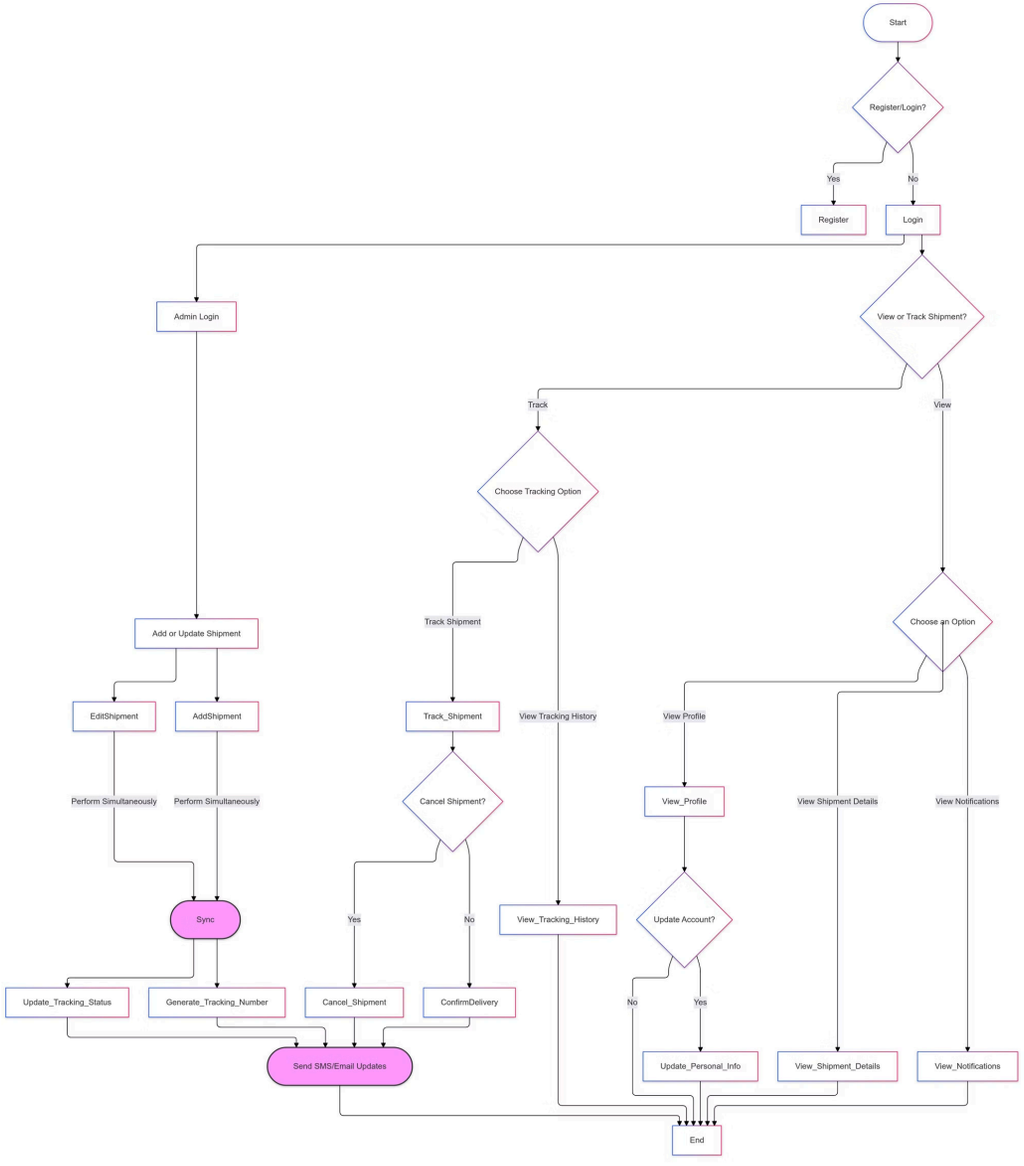
Delivery personnel have 6 use cases focused on operational tasks like viewing delivery schedules, updating delivery statuses, confirming package pickups, and synchronizing routes.

Warehouse Operations

The warehouse team has 6 use cases related to inventory management, package sorting, scheduling shipments, and handling returns.

Summary: The system encompasses 37 use cases across four roles: Customer, Admin, Delivery Personnel, and Warehouse.





Activity Diagram

An **Activity Diagram** is a flowchart used to visualize the sequence of actions, decision points, and processes within a system. It provides a clear view of how different activities interact and flow from start to finish.

Key Points of the Activity Diagram

1

User Registration and Login

Users start by either registering or logging into the system. Admins have a separate login path for managing shipment updates.

2

Shipment Tracking and Management

Customers can view shipment details, track shipments, or review their tracking history. Admins handle shipment updates and can generate tracking numbers.

3

Profile Management

Users can view their profiles and update personal information if necessary.

4

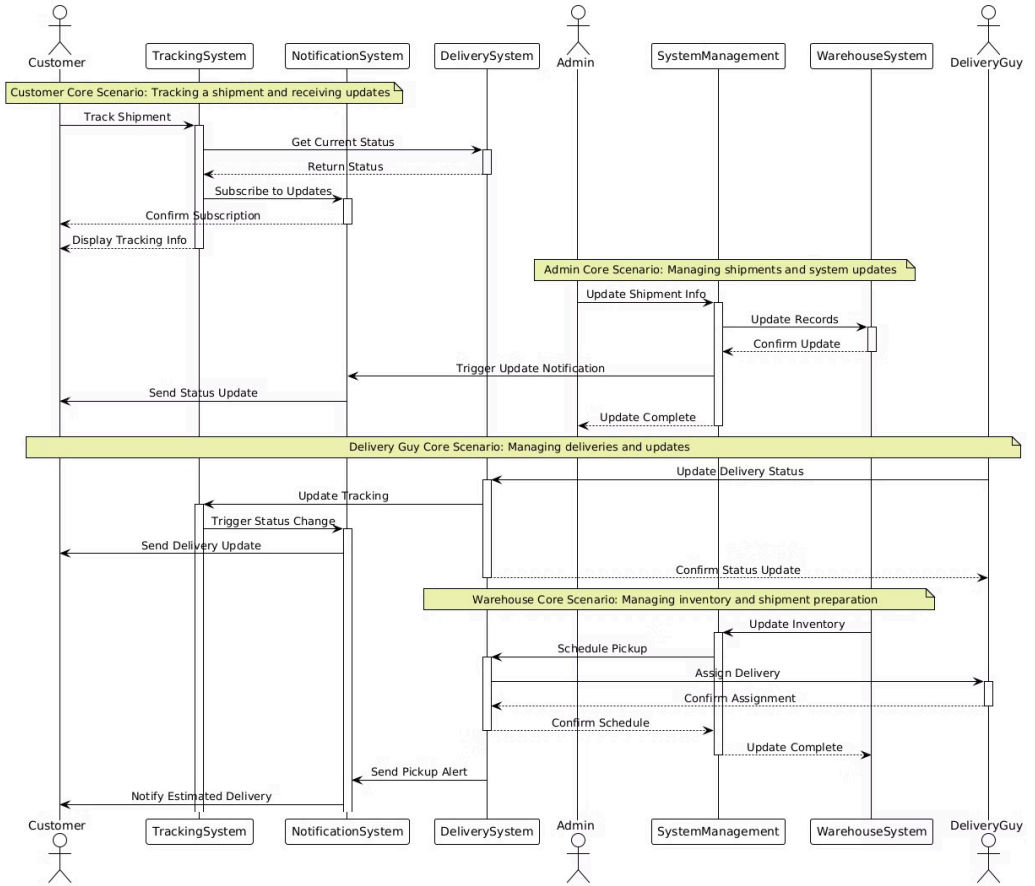
Notifications

Notifications, such as SMS or email updates, are triggered after key actions like shipment tracking, cancellations, or updates.

5

Synchronization of Shipment Data

Admin actions like adding or editing shipment information synchronize simultaneously to ensure tracking numbers and statuses are updated promptly.



Sequence Diagram

A sequence diagram is a type of interaction diagram that models how processes operate with one another by showing the sequence of messages exchanged between objects over time. It provides a clear view of both the dynamic behavior and the flow of control and data within a system.

Key Points of the Sequence Diagram

1 Customer Scenario:

The customer interacts with the tracking and notification systems to track shipments, subscribe to updates, and receive real-time status notifications.

2 Admin Scenario:

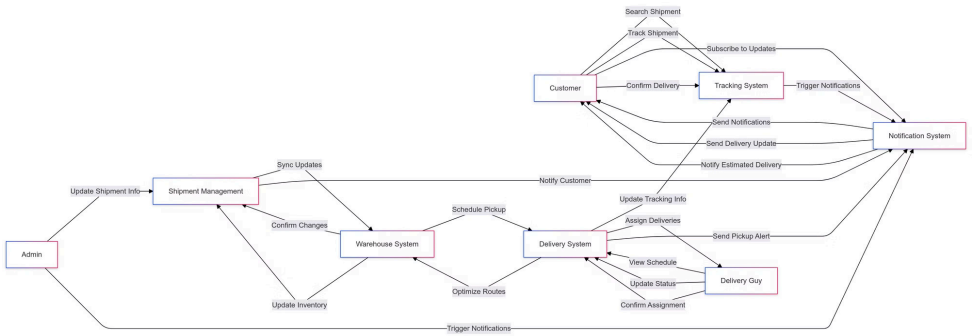
Admins manage shipment information by updating records in the system, which triggers notifications to ensure all parties stay informed about changes.

3 Delivery Workflow:

The delivery personnel update the delivery status and communicate changes with the tracking and notification systems to keep customers informed.

4 Warehouse Operations:

The warehouse system handles inventory updates, schedules pickups, assigns deliveries, and confirms schedules, ensuring efficient shipment preparation.

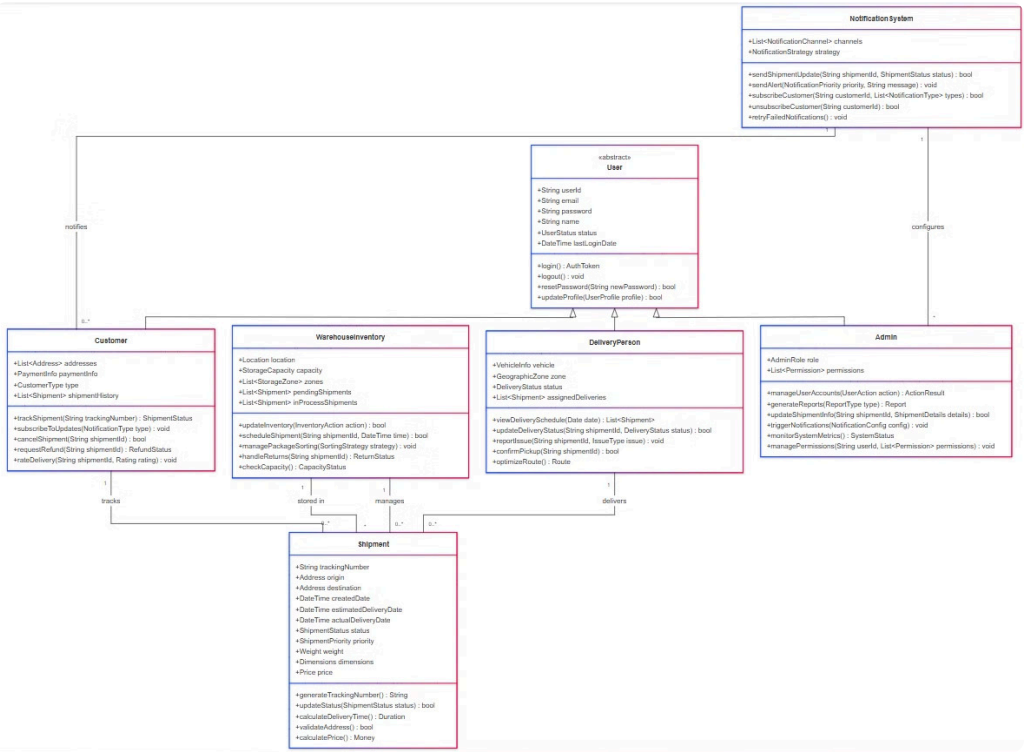


Collaboration Diagram

A **collaboration diagram** illustrates the relationships and interactions between objects or components within a system. Its main purpose is to show how different components work together to achieve system functionalities through communication and coordination.

Key Points of the Collaboration Diagram

- **Customer Interactions:**
  - The customer communicates with multiple systems (Like Tracking and Notification) to search for, track shipments, subscribe to updates, and confirm deliveries.
- **Admin Role:**
  - Admins manage shipment updates via the System Management system, which syncs updates with the Warehouse System and triggers customer notifications.
- **Delivery Workflow:**
  - The delivery personnel interact with the Delivery System to view schedules, update statuses, and confirm delivery assignments, ensuring timely and accurate operations.
- **Warehouse System Functionality:**
  - The Warehouse System updates inventory, schedules pickups, and optimizes routes, working closely with the Delivery System to streamline shipment processes.
- **Notification Flow:**
  - The Notification System acts as a central hub, sending updates and alerts to customers based on triggers from the Tracking, Delivery, and System Management systems.

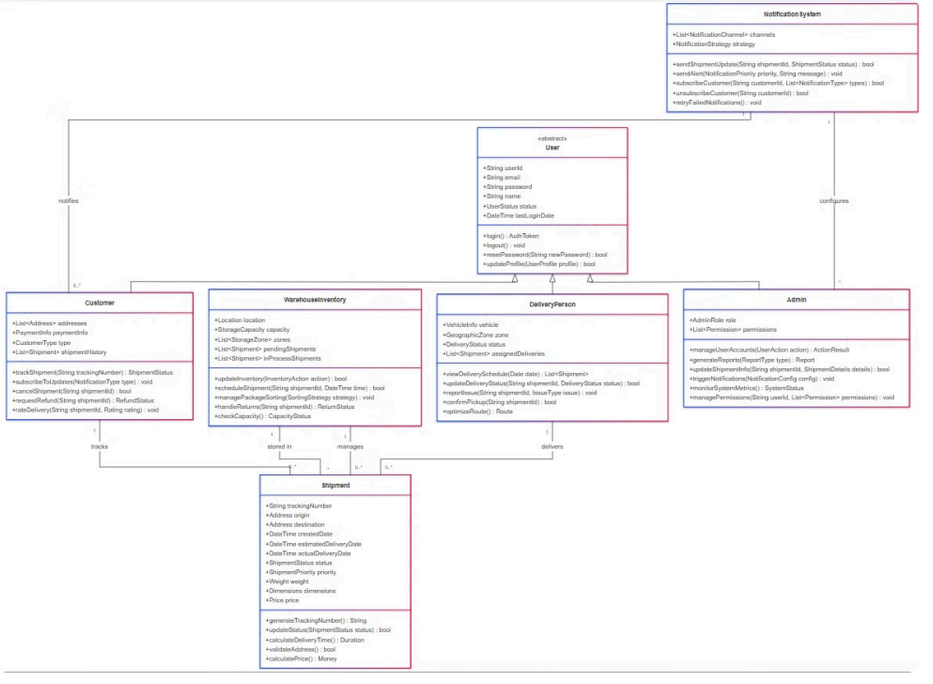


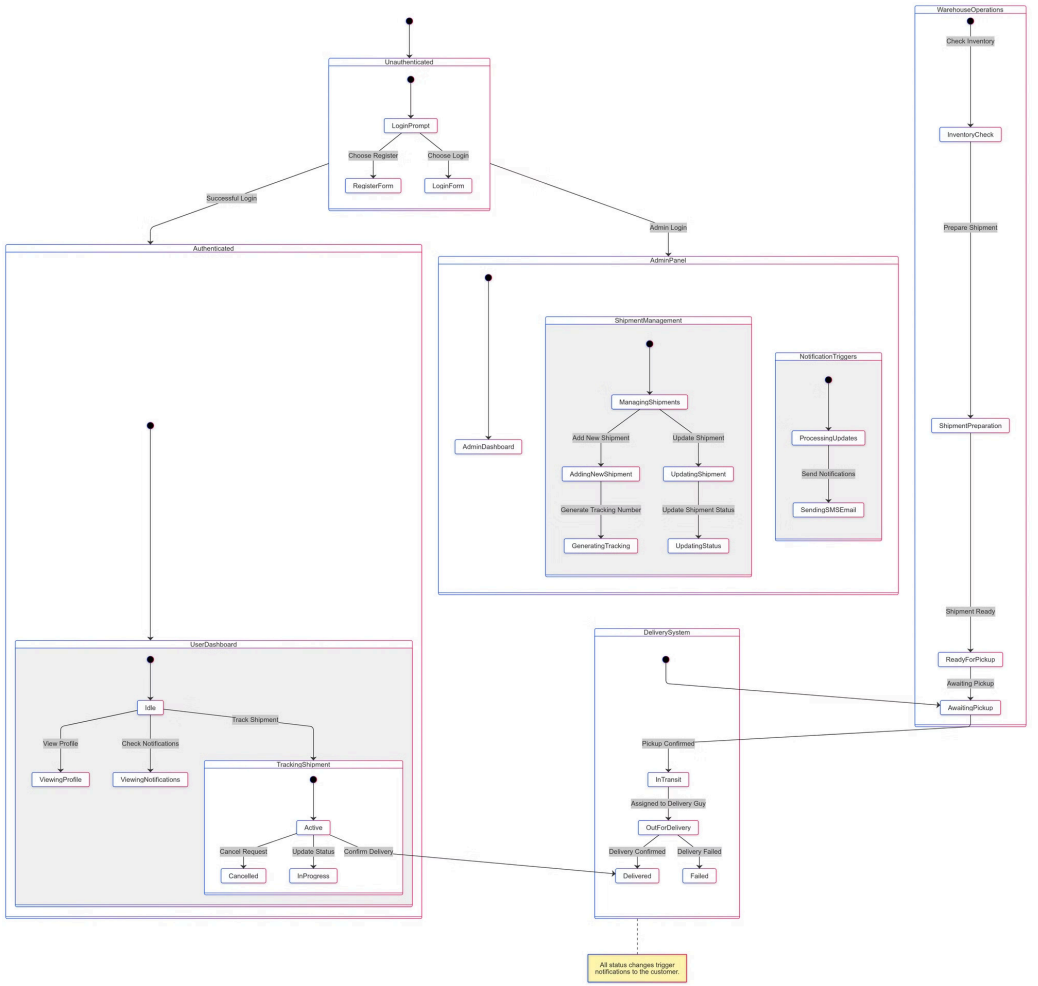
Class Diagram

The **Class Diagram** visually represents the system's structure by showing its classes, attributes, methods, and relationships. Its main purpose is to design and explain the logical structure of a system, particularly in object-oriented programming.

Key Points of the Class Diagram

- User Class:
  - Acts as a parent class for Customer, DeliveryPerson, and Admin with common attributes (e.g., userId, email) and methods like login and resetPassword.
- Customer Class:
  - Manages customer-specific details like addresses, paymentInfo, and methods for tracking shipments or requesting refunds.
- WarehouseInventory Class:
  - Handles inventory management with attributes like location and capacity and methods for scheduling shipments and managing packages.
- Shipment Class:
  - Represents individual shipments with details like origin, destination, and price, and methods like generateTrackingNumber.
- Relationships:
  - Highlights associations such as: Customer tracks shipments. WarehouseInventory stores and manages shipments. NotificationSystem communicates updates to customers.





StateChart Diagram

Statechart Diagrams visually depict a system's states and transitions, clarifying its behavior and workflow for documentation.

Key Points of the StateChart Diagram

1

Login Authentication:

Users start in the **Unauthenticated** state with options to log in or register. A successful login transitions them to the **Authenticated** state.

2

User Dashboard:

In the authenticated state, users can access the **Dashboard**, where they can view profiles, check notifications, or track shipments. Tracking shipments includes sub-states like active, cancelled, or delivered.

3

Admin Panel:

Admins manage shipments and notifications through sub-states like **Adding New Shipment**, **Updating Shipment**, and sending notifications via SMS or email.

4

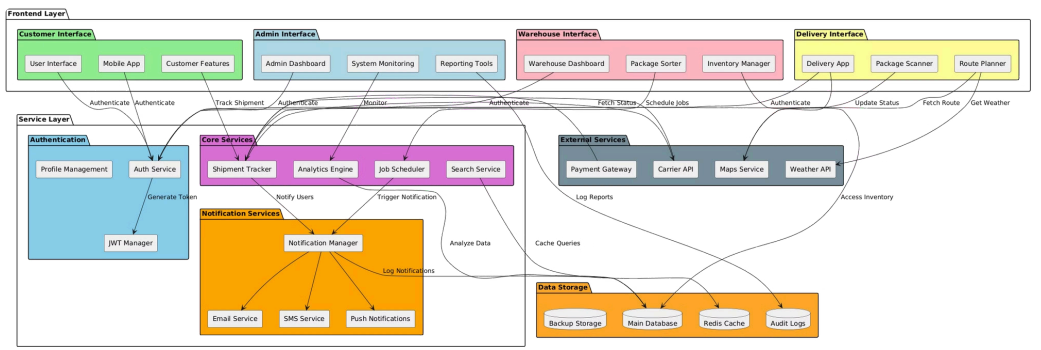
Delivery System States:

The delivery process progresses from **Awaiting Pickup** to **In Transit**, then **Out for Delivery**, and finally **Delivered** or **Failed**. Status changes trigger customer notifications.

5

Warehouse Operations:

Warehouse workflows include **Inventory Check**, **Shipment Preparation**, and transitioning shipments to the delivery system via the **Awaiting Pickup** state.



Component Diagram

A **Component Diagram** illustrates the structural relationships and interactions between software components in a system. It helps in understanding how the system is organized and how different parts collaborate to perform functionalities.

Key Points of the Component Diagram

1 User Interface Component:

Represents the front-end interface where users interact with the system, handling user input and displaying outputs.

2 Business Logic Component:

Contains the core functionality of the system, processing user requests and coordinating actions between components.

3 Database Component:

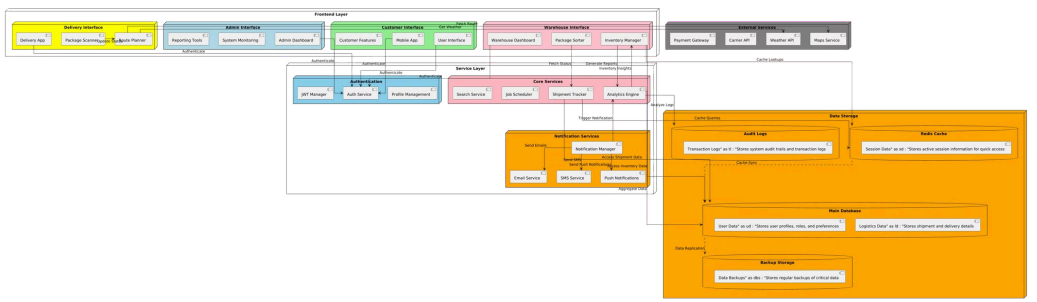
Manages data storage and retrieval, ensuring data integrity and availability across the system.

4 External API Integration:

Shows how the system connects with third-party services for features like shipment tracking or notifications.

5 Component Relationships:

Highlights the connections and communication flow between components, ensuring seamless collaboration within the system.



Deployment Diagram

A **Deployment Diagram** represents the physical architecture of a system, showing how hardware nodes and software components interact. It helps in understanding the system's deployment environment and resource allocation.

Key Points of the Deployment Diagram

Client-Server Architecture

Illustrates the interaction between client devices and the server hosting the system.

Database Node

Shows the location of the database and its connections to other components for data storage and retrieval.

Web Server Requests

Depicts the web server's role in handling requests and delivering responses to users.

External Services

Highlights integrations with third-party services for features like shipment tracking or notifications.

Network Communication

Displays the communication flow between nodes, ensuring seamless data exchange across the system.

Thank You 😊