

E-Commerce Store

System Documentation

Prepared by:

Abd Allah Emad

December 2025

1. Executive Summary

This document provides comprehensive documentation for the E-Commerce Store system, a multi-vendor marketplace platform designed to enable online buying and selling of products. The system supports multiple user roles including customers, vendors, administrators, content managers, support agents, and delivery drivers.

The platform addresses the growing need for digital transformation in retail, providing customers with a convenient shopping experience while enabling vendors to reach a wider audience. The system features secure payment processing, order tracking, product management, and comprehensive administrative controls.

1.1 Key Features

- Multi-vendor marketplace architecture
- Comprehensive user authentication and authorization
- Advanced product search and filtering capabilities
- Secure payment gateway integration (Credit Card, PayPal, Cash on Delivery)
- Real-time order tracking and status updates
- Product review and rating system
- Vendor management and product approval workflow
- Administrative dashboard with full system control
- Delivery management system with driver mobile app
- Promotional offers and coupon management

1.2 Target Users

The system serves six distinct user roles, each with specific capabilities and responsibilities:

- **Customers:** Browse and purchase products, manage orders, write reviews
- **Vendors:** Manage product inventory, fulfill orders, track sales
- **Super Administrators:** Full system control, financial management, user oversight
- **Content Managers:** Product approval, vendor verification, review moderation
- **Support Agents:** Customer service, ticket management, return processing
- **Delivery Drivers:** Package pickup and delivery, cash collection for COD orders

2. Project Overview

2.1 Project Definition

This project develops a comprehensive e-commerce platform enabling users to browse, purchase, and receive products online. The platform features a user-friendly interface, secure payment methods, and an efficient order tracking system. The store operates as a multi-vendor marketplace, allowing multiple sellers to list and sell their products through a centralized platform.

2.2 Problem definition

The retail landscape has undergone significant transformation with the expansion of e-commerce. Traditional stores struggle to adapt to digital transformation, while customers demand convenient, fast, and secure shopping experiences. Key challenges addressed by this project include:

- Lack of effective digital platforms for small and medium-sized businesses
- Insufficient product information and difficult purchase processes on existing platforms

- Poor after-sales service and customer support
- Limited access to international products and vendors
- Lack of trust and transparency in online transactions

2.3 Project Goals

The primary objectives of this project are:

- Design an integrated, user-friendly online shopping experience
- Implement interactive product display with comprehensive information
- Provide multiple payment and delivery options
- Enable real-time order tracking capabilities
- Support cross-platform access (Web, Android, iOS)
- Enable effective inventory and sales management for vendors
- Implement authentic product review and rating system
- Create a flexible, secure, and competitive e-commerce platform comparable to global marketplaces

2.4 Project Scope

2.4.1 Included in Scope

The project encompasses the following deliverables and features:

Internal Deliverables:

- **Project Plan:** Comprehensive scope, schedule, cost, resources, and risk analysis
- **Business Requirements Specification:** Detailed functional requirements covering all system capabilities
- **Quality Plan:** Security compliance standards, performance benchmarks, and testing procedures
- **Communication Plan:** Stakeholder communication protocols and feedback mechanisms
- **Maintenance Plan:** System update procedures, security audit schedules, and operational costs

External Deliverables:

- **Website Platform:** User-friendly interface with product catalog, shopping cart, and checkout system
- **Mobile Applications:** Native Android and iOS applications for customers and drivers
- **Payment Gateway Integration:** Secure integration with PayPal, credit card processors, and cash on delivery
- **Order Management System:** Complete order tracking, status notifications, and invoice generation
- **Digital Marketing:** SEO optimization, social media integration, and email marketing campaigns
- **Customer Service:** FAQ system and live chat support

2.4.2 Excluded from Scope

The following features are explicitly excluded from the current project phase:

- Windows and Mac OS desktop applications
- AI-powered recommendation system
- Multi-language support

3. Project Planning

3.1 Project Life Cycle

The project follows a structured development lifecycle with the following phases:

- Database Design
- GUI Design
- Frontend Development
- Backend Development
- Network Design
- Testing and Quality Assurance
- Deployment and Maintenance

3.2 Technology Stack

3.2.1 Development Tools and Languages

- **Database:** MySQL with MySQL Workbench for design and management
- **GUI Design:** Adobe XD for interface prototyping
- **Frontend:** JavaScript, HTML, CSS with Visual Studio Code
- **Backend:** Python, C++, PHP with appropriate IDEs
- **Network:** Google Cloud Platform, SSH clients, CDN integration

3.2.2 Development Methodology

- **Prototyping:** Using Adobe XD or Figma for initial design
- **Lean Startup:** Using Lean Canvas for validation of ideas
- **Agile:** Using Asana or Trello for project task management

3.3 Project Schedule

The project timeline spans 30 weeks with the following major phases:

Phase	Duration	Weeks
Planning	5 weeks	Weeks 1-5
Requirements Collection	2 weeks	Weeks 6-7
Analysis	5 weeks	Weeks 8-12
Design	5 weeks	Weeks 13-17
Development	8 weeks	Weeks 18-25
Testing & Maintenance	4 weeks	Weeks 26-29
Delivery & Installation	1 week	Week 30

Critical Path: Writing Report → Database Design → UI Design → Coding → Testing → Maintaining

3.4 Budget Allocation

Total project budget: \$150,000 with the following allocation:

Component / Resource	Estimated Cost (USD)
UI/UX Design Tools	\$2,000
Developer Salaries	\$48,000
Database & Server Hosting	\$5,000
Domain & SSL Certificate	\$5,000
Testing & QA Tools	\$10,000
Payment Gateway Integration	\$5,000
Security Enhancements	\$10,000
Hardware (PCs, Routers, etc)	\$25,000
Contingency Reserve (10%)	\$15,000
Total	\$125,000

4. System Requirements

4.1 Functional Requirements

Functional requirements define the specific behaviors and functions of the system. The following table summarizes the core functional requirements:

ID	Description
FR-01	User registration with email and password
FR-02	Login and logout functionality with session management
FR-03	Product search by name, category, and attributes with filtering options
FR-04	Shopping cart management (add, remove, update quantities)
FR-05	Checkout process with multiple payment options
FR-06	Order tracking with real-time status updates
FR-07	Product review and rating system
FR-08	Vendor registration and verification system
FR-09	Product inventory management for vendors
FR-10	Administrative dashboard for system management

4.2 Non-Functional Requirements

Non-functional requirements specify the quality attributes and constraints of the system:

4.2.1 Performance Requirements

- System response time must not exceed 3 seconds for standard operations
- Support for 1,000+ concurrent users without performance degradation
- Database queries optimized for sub-second response times
- Image loading optimized with lazy loading and compression

4.2.2 Security Requirements

- All user data must be encrypted both in transit (SSL/TLS) and at rest
- Password storage using industry-standard hashing algorithms (bcrypt)
- Payment information handled through PCI-DSS compliant gateways
- Role-based access control (RBAC) for all system functions
- Regular security audits and vulnerability scanning

4.2.3 Usability Requirements

- Intuitive interface requiring minimal training for end users
- Mobile-responsive design for all web pages
- Compatible with major browsers (Chrome, Firefox, Safari, Edge)
- Accessibility compliance with WCAG 2.1 guidelines

4.2.4 Scalability Requirements

- Architecture designed to scale horizontally with increased load
- Database partitioning and replication support for growth
- Cloud-based infrastructure allowing elastic scaling

4.3 Requirements Priority

Requirements are prioritized to guide development efforts and resource allocation:

ID	Requirement	Priority
FR-01	User Registration	HIGH
FR-02	Login Functionality	HIGH
FR-03	Search Feature	MEDIUM
FR-04	Responsive Design	MEDIUM
FR-05	Dark Mode Support	LOW

5. System Use Cases

This section documents the primary use cases that define system functionality. The system includes 26 detailed use cases covering all major interactions between actors and the system. Below are selected key use cases representing core system capabilities.

5.1 UC-01: User Registration

Use Case ID	UC-01
Use Case Name	User Registration
Actors	User
Pre-Conditions	User is not pre-registered and has good internet connection
Post-Conditions	User becomes registered in the system and can log in
Normal Flow	<ol style="list-style-type: none">1. User opens the registration page2. User enters username, email address, and password3. User clicks on "Registration" button4. System verifies information and saves account5. System displays registration success message
Alternative Flows	<ul style="list-style-type: none">• Email already exists → System displays error message• Password is weak → System requests stronger password• Field is blank → System requests field completion

5.2 UC-07: Payment / Checkout

Use Case ID	UC-07
Use Case Name	Payment / Checkout
Actors	User, Payment Gateway
Pre-Conditions	User has items in cart and is logged in
Post-Conditions	Order is created, payment processed, and inventory reduced
Normal Flow	<ol style="list-style-type: none">1. User clicks "Proceed to Checkout"2. User enters shipping address3. User chooses payment method (Credit Card/PayPal/COD)4. User clicks "Place Order"5. System sends payment request to gateway6. Gateway authorizes transaction7. System confirms order and sends email receipt
Alternative Flows	<ul style="list-style-type: none">• Payment failed → Gateway rejects card → System displays error• Cash on Delivery selected → System skips gateway steps → Sets status to Pending

5.3 Complete Use Case Summary

The system includes 26 detailed use cases covering all interactions. The complete list includes:

ID	Use Case	Primary Actors
UC-01	User Registration	User
UC-02	Log In & Log Out	All Users
UC-03	Browse Products	User
UC-04	Search For Products	User
UC-05	Add To Shopping Cart	User
UC-06	Remove From Shopping Cart	User
UC-07	Payment / Checkout	User, Payment Gateway
UC-08	View Order History	User
UC-09	Account Management	User
UC-10	Product Verification & Approval	Content Manager
UC-11	Admin Order Monitoring	Super Admin
UC-12	Product Review & Rating	User
UC-13	Offers & Coupons Management	User
UC-14	Notifications & Messages Management	User
UC-15	Payment Management	Super Admin & The System & Payment Gateway (External System)
UC-16	Vendor Registration	Vendor
UC-17	Add Products	Vendor
UC-18	Order Fulfillment	Vendor
UC-19	User Reviews Moderation	Content Manager
UC-20	Verify Vendor Documentation	Content Manager
UC-21	Manage Employee Accounts	Super Admin
UC-22	Platform Configuration & Commission Settings	Super Admin
UC-23	Manage Support Tickets	Support Agent
UC-24	Manage Return Requests (RMA)	Support Agent
UC-25	Delivery Fulfillment (Pickup & Drop-off)	Driver
UC-26	Cash Collection (COD)	Driver

6. System Diagrams

This section provides an overview of the system diagrams that illustrate the system architecture, workflows, and data models. The complete set of diagrams includes Activity Diagrams, State Machine Diagrams, Class Diagrams, and Sequence Diagrams.

6.1 Activity Diagrams

Activity diagrams illustrate the operational workflow within the system sequentially. They serve as a "roadmap" depicting the complete order journey, starting from the customer browsing products, through vendor processing, up to the final delivery by the driver. Swim lanes are utilized to segregate responsibilities, clearly defining the role each actor (User, System, Vendor, Driver, Payment Gateway) plays in successfully completing this cycle.

6.2 State Machine Diagrams

State Machine Diagrams focus on the lifecycle of key entities within the system, such as Orders, Products, Payments, and Vendor Accounts. They visualize the various states an object transitions through (e.g., how an Order moves from "Processing" to "Shipped" to "Delivered") and precisely define the events (triggers) that cause these transitions, ensuring a clear understanding of the system's logic in handling status changes across different modules.

6.3 Class Diagram

The Class Diagram represents the static structure of the system, illustrating the database schema and the relationships between system entities. An Inheritance model is utilized for the "User Management" module, where a central User Account class defines common attributes (ID, Email, Password), while specialized classes (Vendor, Driver, Content Manager) inherit these properties and extend them with role-specific data. This ensures a modular and scalable architecture capable of supporting the Multi-Vendor Marketplace logic.

6.4 Sequence Diagrams

Sequence Diagrams illustrate the dynamic interactions between system actors and components over time. The "End-to-End Order Processing" scenario has been divided into three distinct phases:

- Checkout & Payment: Covers the User's order placement and interaction with the Payment Gateway
- Vendor Fulfillment: Details the Vendor's process of validating, packing, and preparing items for shipment
- Logistics & Delivery: Focuses on the Driver's assignment, package handover, and final delivery/cash collection

Collectively, these diagrams demonstrate how the Multi-Vendor architecture components are logically synchronized across the entire order lifecycle.

7. Risk Management

Risk management is critical to project success. The following table identifies key risks, their probability, impact, and mitigation strategies:

Risk Description	Probability	Impact	Strategy
Delay in UI/UX Design Completion	Medium	Medium	Set strict deadlines and monitor progress weekly
Backend API Integration Issues	High	High	Schedule buffer time; hold integration testing early
Hosting/Server Downtime During Deployment	High	Medium	Choose reliable hosting provider and prepare backup plans
Payment Gateway Failure	Medium	Medium	Use widely supported gateways and conduct early testing
Developer Unavailability	High	Medium	Cross-train team members and document progress continuously
Budget Overrun	Medium	Medium	Track expenses regularly and include contingency reserve
Data Loss or Corruption	Medium	High	Use version control (Git) and automated backups
Security Vulnerabilities	High	Medium	Perform security testing in parallel with development
Scope Creep	Medium	High	Freeze scope after planning; evaluate any change with impact
Poor User Feedback	Medium	Medium	Conduct early UI/UX testing with users and iterate

8. Assumptions and Constraints

8.1 Assumptions

8.1.1 Basic Assumptions

- Internet Connectivity: Customers have reliable internet access
- Basic User Knowledge: Users have basic knowledge of using internet and online shopping
- Payment Gateway Stability: Online payment gateways integrated are stable and secure
- Shipping Service Availability: Shipping services are available and reliable
- Legal Compliance: The online store complies with all local and international regulations

8.1.2 Technology-Related Assumptions

- Browser Compatibility: Online store is compatible with most modern web browsers
- Server Stability: Web server hosting the online store is stable and secure
- API Availability: Necessary APIs for integrating payment gateways and shipping services are available

8.1.3 Market and Customer-Related Assumptions

- Product Demand: There is sufficient demand for the products to be sold
- Affordability: Customers can afford the costs of products and shipping
- Trust in Online Shopping: Customers are confident in online shopping

8.2 Constraints

- **Time:** Project delivery deadline, website launch schedule, time limitations on project phases
- **Cost:** Project budget limitations, website development costs, hosting and maintenance costs
- **Scope:** Required website features and functionalities, number of products to be displayed
- **Quality:** Required website quality standards, security and protection requirements
- **Resource:** Availability of developers and designers, availability of necessary hardware and software

9. Resource Allocation

The project requires various specialized resources with specific skill sets. The following table outlines the resource allocation and timeframes:

Resource	Total	Skill Set Requirements	Timeframe
System Analyst	1	Analyze business needs and translate them into system requirements	1/4/2025 to 14/4/2025
UI/UX Designer	2	Experience in designing user-friendly interfaces for e-commerce platforms	15/4/2025 to 5/5/2025
Backend Developers	3	Knowledge of server-side languages (Python, PHP, Node.js)	6/5/2025 to 16/6/2025
Frontend Developers	3	HTML, CSS, JavaScript, React or Vue	2/6/2025 to 6/7/2025
Database Engineer	1	Design & manage relational databases (MySQL)	15/4/2025 to 5/5/2025
Network Setup	2	Setup of routers, switches, local network	7/7/2025 to 21/7/2025
Server & Hosting	2	Handle 200+ concurrent users, secure, scalable	22/7/2025 to 4/8/2025
Payment Gateway Integrator	1	Integration of services like PayPal, Stripe	7/7/2025 to 21/7/2025
Security Specialist	1	Implement SSL, data encryption, user authentication	5/8/2025 to 2/9/2025
Tester	1	Detect and report bugs, conduct system testing	3/9/2025 to 8/10/2025

Note: Buffer period from 9/10/2025 to 27/11/2025 is allocated for delays and contingencies.

10. Future Enhancements

The following features are identified for future development phases to enhance the platform capabilities:

1. Windows & Mac OS Desktop Applications
2. AI Recommendation System for personalized shopping
3. Multi-Language Support for international markets
4. Advanced Search & Filters with AI-powered suggestions
5. Real-time Chat Support with chatbot integration
6. Loyalty & Rewards Program for customer retention
7. Inventory Management With Analytics and predictive insights
8. Additional Payment Gateways for broader coverage
9. Order Tracking Via GPS In More Countries
10. Enhanced Security with Multi-Factor Authentication (MFA)
11. Cloud Migration & Scalability.

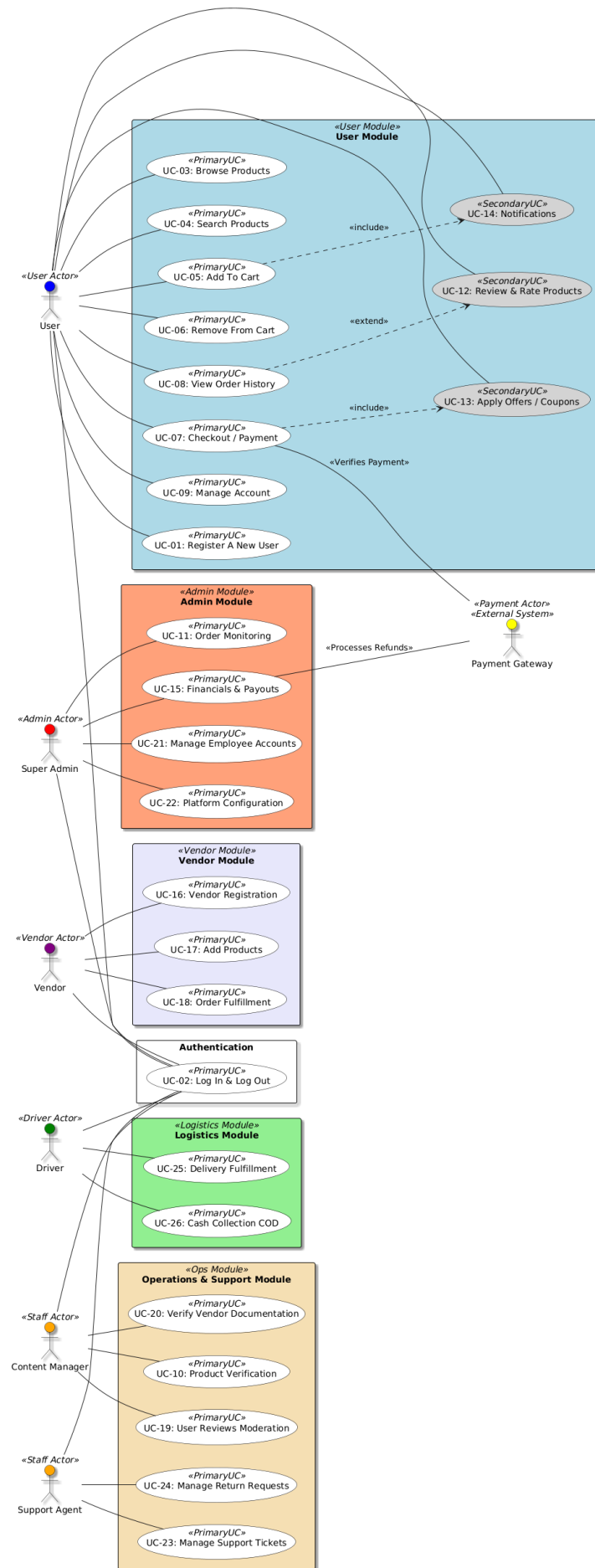
11. Conclusion

This documentation provides a comprehensive overview of the E-Commerce Store system, covering all aspects from project planning and requirements to system design and use cases. The project aims to deliver a robust, scalable, and secure multi-vendor marketplace that addresses the evolving needs of modern online retail.

The system architecture is designed with flexibility and extensibility in mind, allowing for future enhancements and feature additions. By following industry best practices in software development, security, and user experience design, the platform is positioned to compete effectively with established global e-commerce marketplaces.

The careful attention to risk management, resource allocation, and project scheduling ensures that the development process remains on track and within budget. The comprehensive use case documentation and system diagrams provide clear guidance for implementation teams and stakeholders.

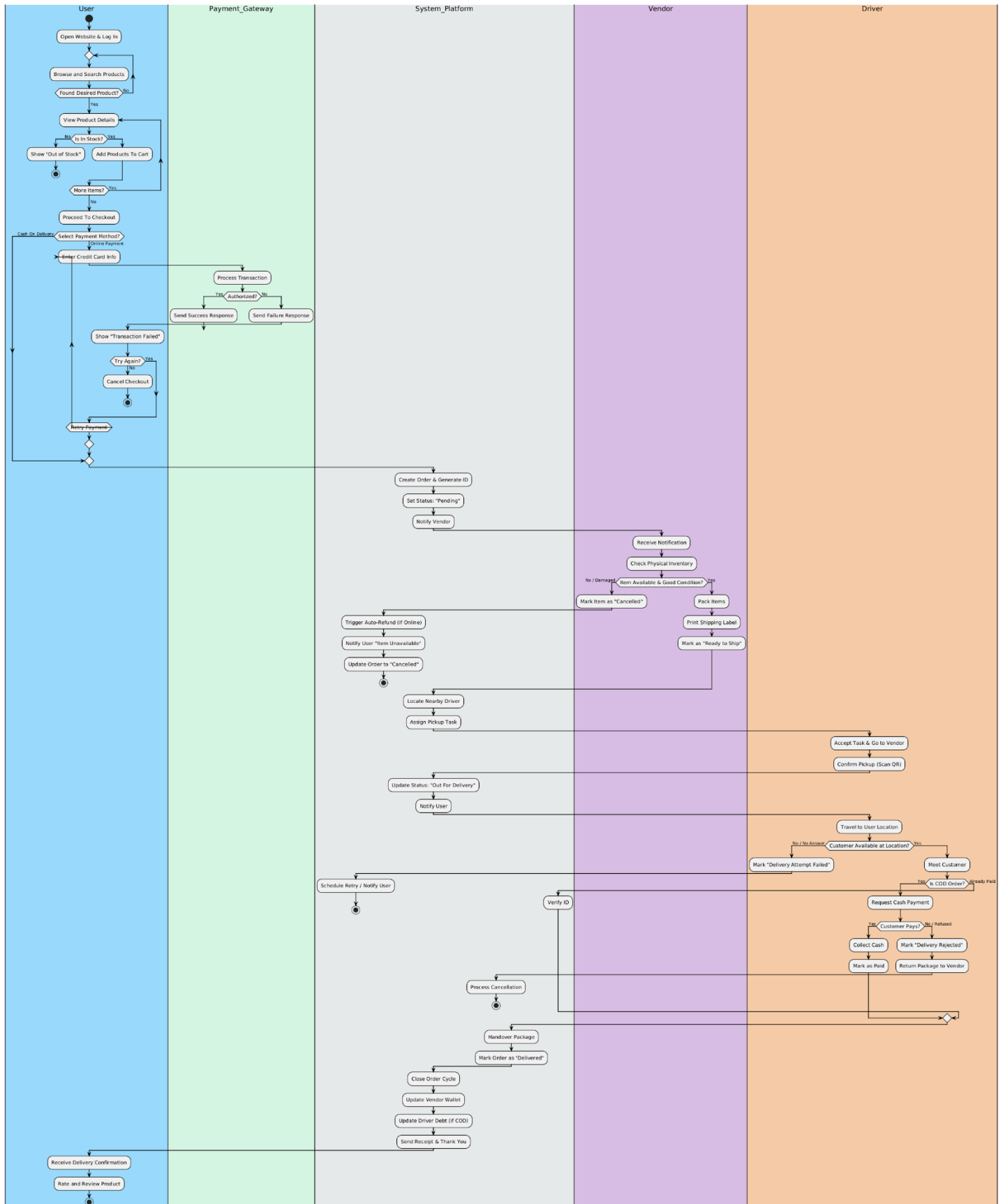
Success of this project will be measured by achieving the defined goals: providing an intuitive user experience, ensuring secure and reliable operations, enabling vendor success, and delivering value to all stakeholders in the e-commerce ecosystem.



Activity Diagram Introduction

"This diagram illustrates the operational **workflow** within the system sequentially. It serves as a 'roadmap' depicting the complete order journey, starting from the customer browsing products, through vendor processing, up to the final delivery by the driver.

Swim lanes are utilized to segregate responsibilities, clearly defining the role each actor (User, System, Vendor, Driver, Payment Gateway) plays in successfully completing this cycle."

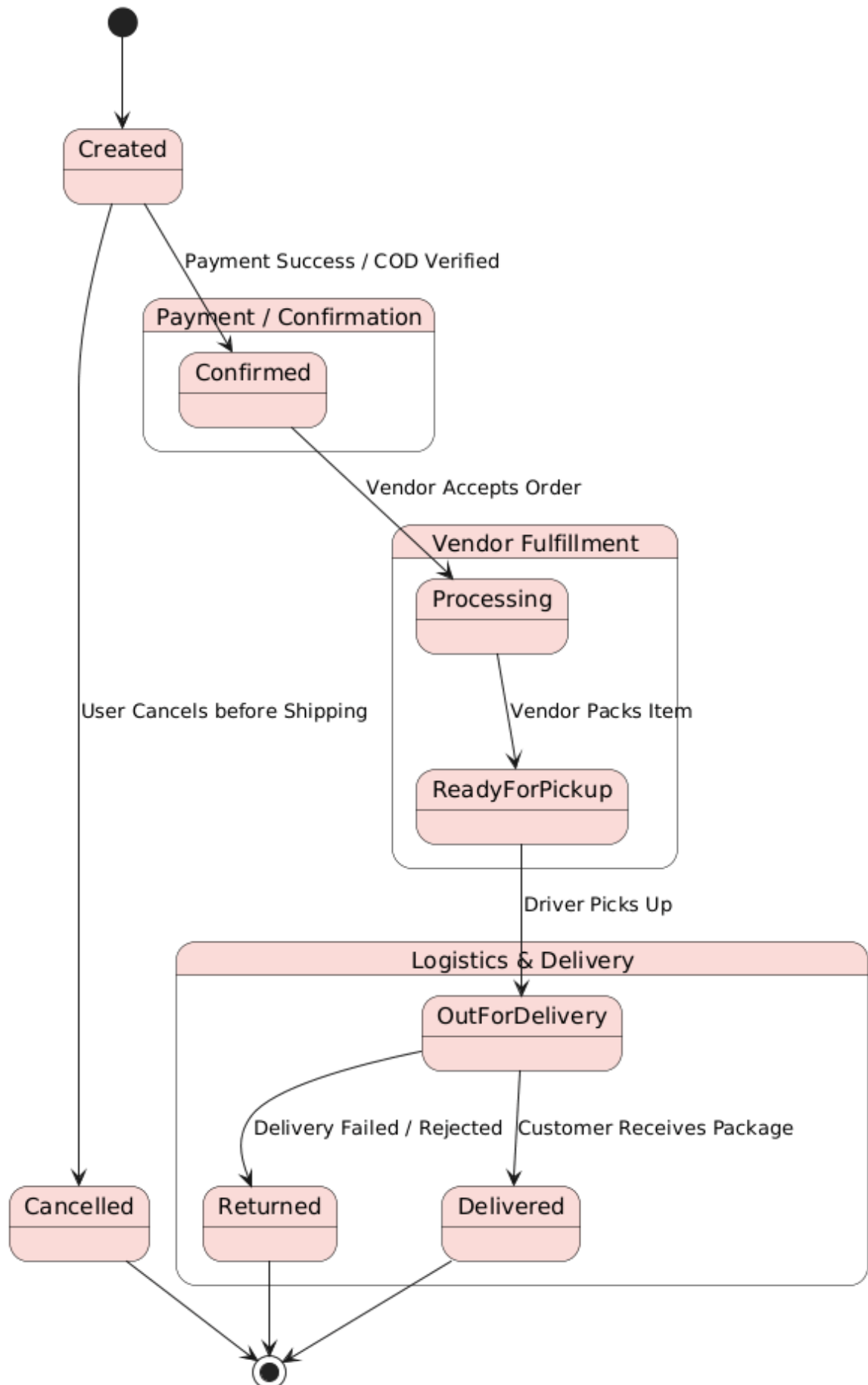


State Machine Diagrams Introduction

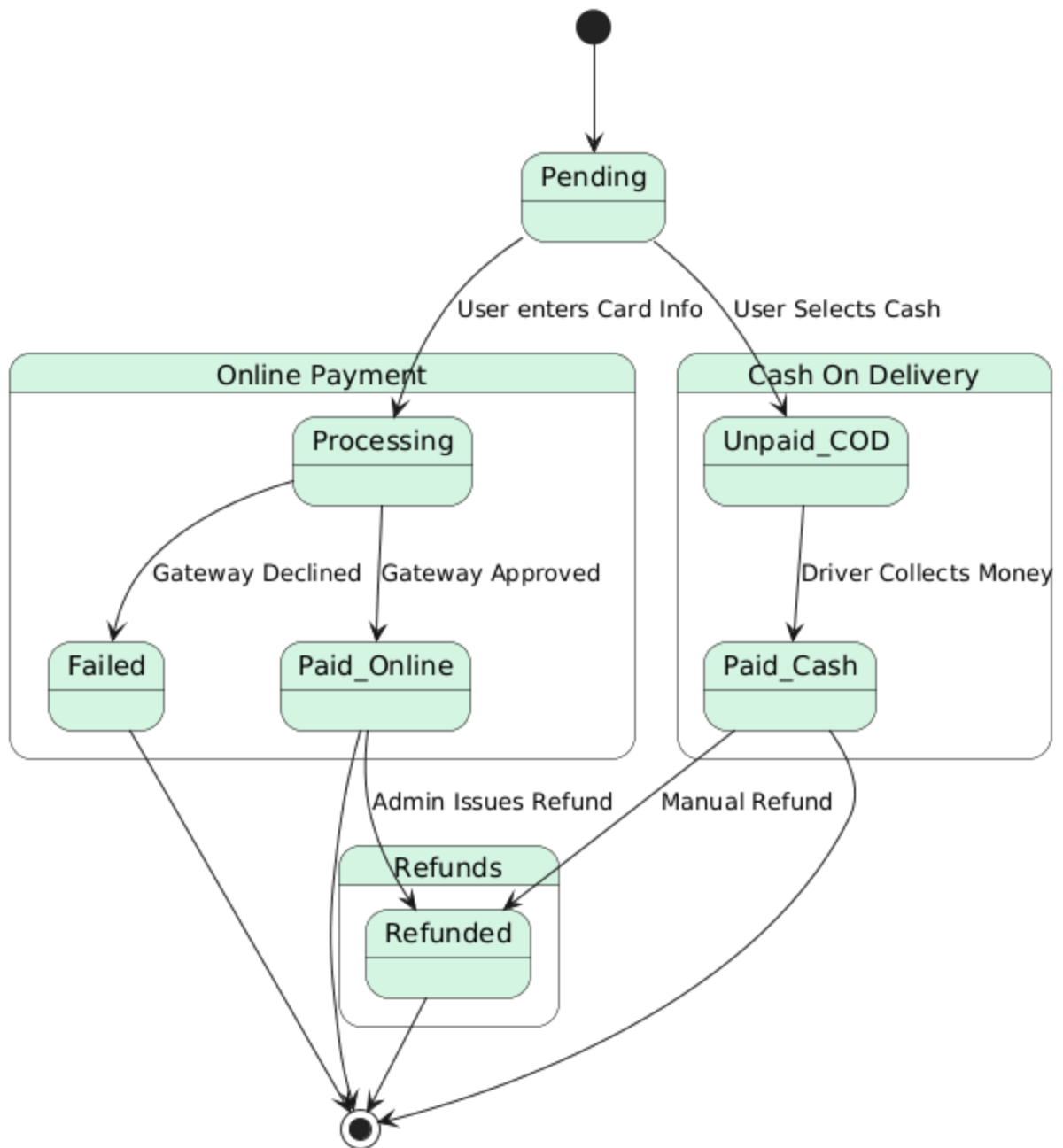
"These diagrams focus on the lifecycle of key entities within the system, such as Orders, Products, Payments, and Vendor Accounts.

They visualize the various states an object transitions through (e.g., how an Order moves from 'Processing' to 'Shipped' to 'Delivered') and precisely define the events (triggers) that cause these transitions, ensuring a clear understanding of the system's logic in handling status changes across different modules."

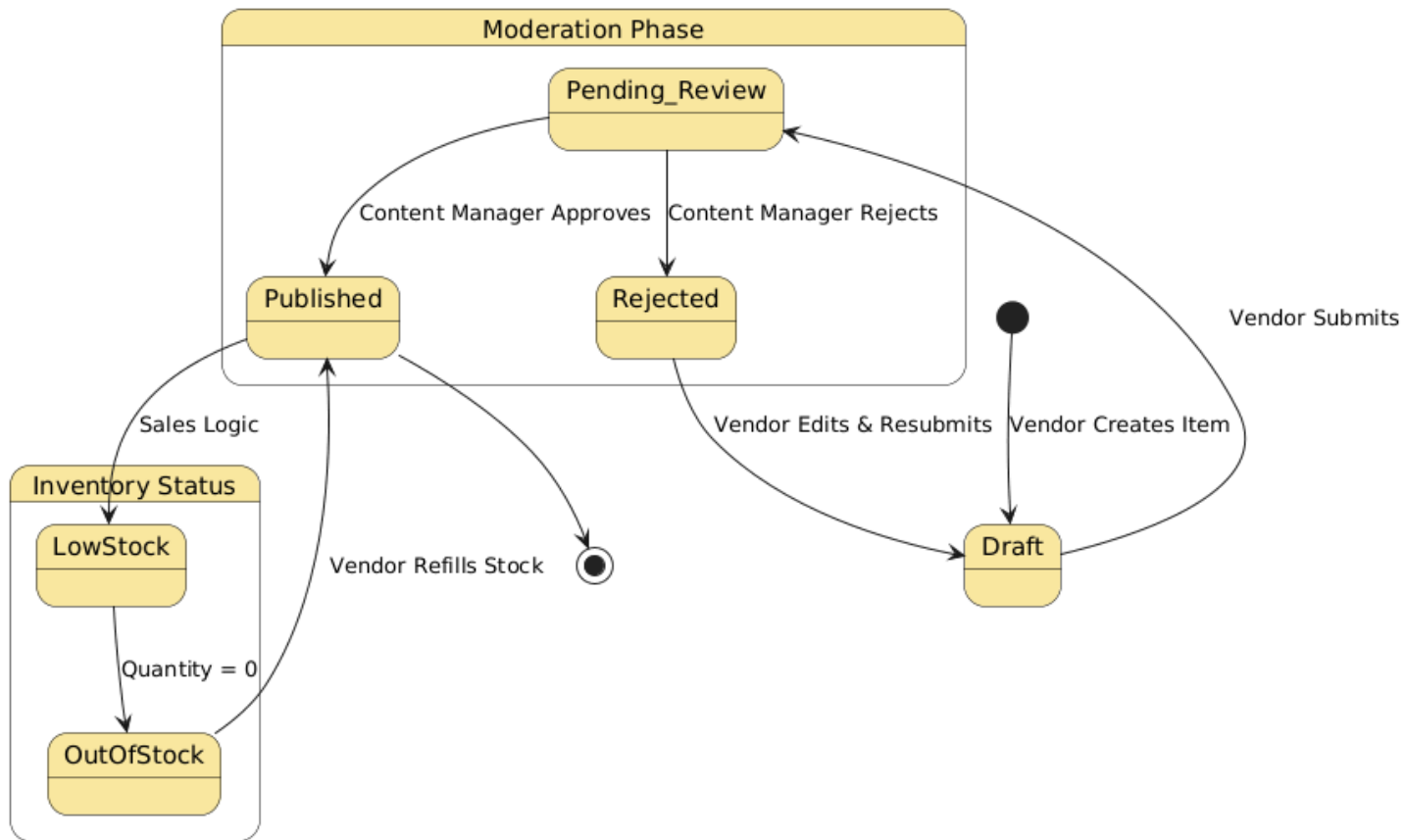
2. Order Lifecycle (Multi-Vendor & Driver)



3. Payment Lifecycle (Online & COD)



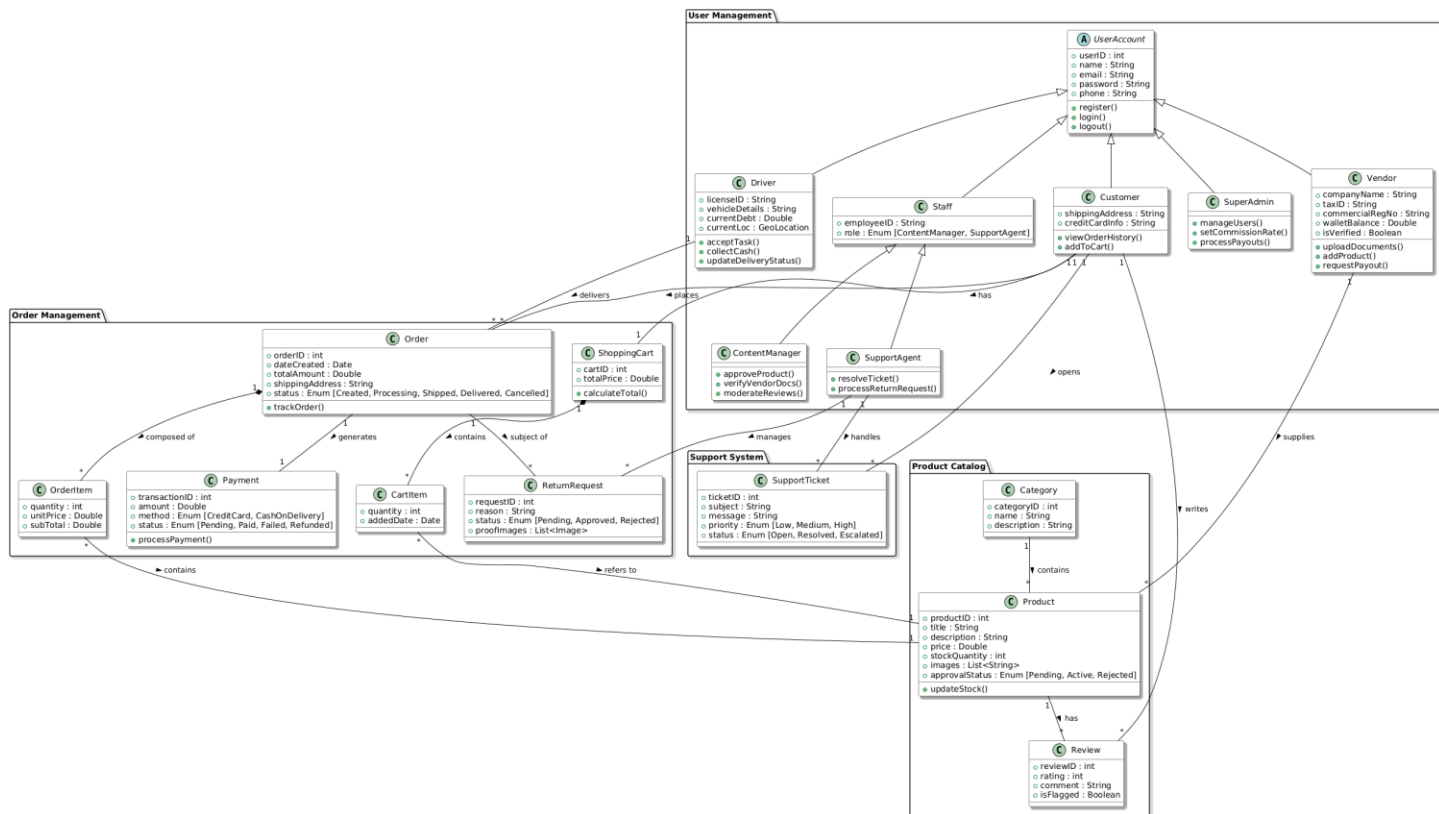
4. Product Publication & Stock Lifecycle



Class Diagram Introduction

"The Class Diagram represents the **static structure** of the system, illustrating the database schema and the relationships between system entities.

In this design, an **Inheritance** model is utilized for the 'User Management' module, where a central User Account class defines common attributes (ID, Email, Password), while specialized classes (Vendor, Driver, Content Manager) inherit these properties and extend them with role-specific data. This ensures a modular and scalable architecture capable of supporting the **Multi-Vendor Marketplace** logic, including distinct handling for vendor wallets, driver debts, and product moderation workflows."



Sequence Diagrams Introduction

This section illustrates the dynamic interactions between system actors and components over time. To ensure clarity and readability, the 'End-to-End Order Processing' scenario has been **divided into three distinct phases**:

1. **Checkout & Payment:** Covers the User's order placement and interaction with the Payment Gateway (or COD verification).
2. **Vendor Fulfillment:** Details the Vendor's process of validating, packing, and preparing items for shipment.
3. **Logistics & Delivery:** Focuses on the Driver's assignment, package handover, and final delivery/cash collection.

Collectively, these diagrams demonstrate how the Multi-Vendor architecture components are logically synchronized across the entire order lifecycle.

