

System Analysis & Design for an E-Commerce Web Application

1. Introduction

1.1 Purpose

This document provides a comprehensive analysis and design outline for an e-commerce web application with automated DevOps deployment. It covers requirements, architecture, data design, and the deployment pipeline.

1.2 Scope

1. Project Overview
2. Problem Statement & Objectives
3. Use Case Diagram & Descriptions
4. Functional & Non-Functional Requirements
5. Software Architecture
6. Data Flow & System Behavior
7. UI/UX Design & Prototyping
8. System Deployment & Integration
9. Additional Deliverables
10. Conclusion

1.3 Audience

The primary audience includes:

- **Developers:** who need to understand the system's architecture and codebase.
- **DevOps Engineers:** responsible for creating and maintaining the automated deployment pipeline.
- **Project Managers:** overseeing timelines, deliverables, and resource allocation.
- **QA/Test Engineers:** validating functional and non-functional requirements.

2. Problem Statement & Objectives

2.1 Problem Statement

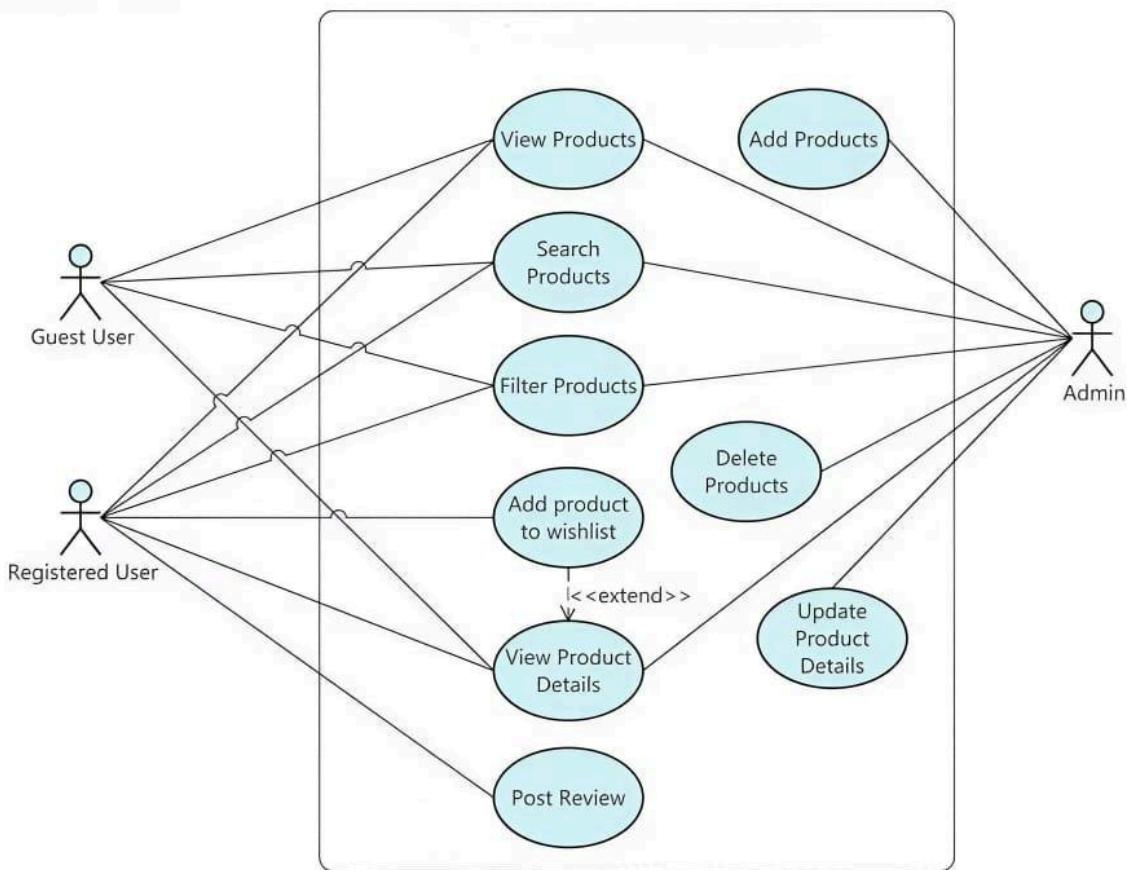
Manual deployment processes are time-consuming and error-prone, often leading to inconsistent environments and deployment failures.

2.2 Objectives

- Automate the deployment of an e-commerce web application using modern DevOps tools.
 - Reduce errors and downtime by establishing a robust CI/CD pipeline.
 - Improve scalability and performance by containerizing services and using orchestration platforms.
-

3. Use Case Diagram & Descriptions

2.1 Use Case Diagram



Actors & Their Roles

- **Guest User can:**
 - **View Products**
 - **Search Products**
 - **Filter Products**
 - **View Product Details**
- **Registered User can perform all guest actions plus:**
 - **Add Product to Wishlist**
 - **Add Product to Cart**
 - **Place Orders**
 - **Track Orders**
 - **Make Payments**
 - **Return Products**
 - **Post Reviews**
- **Admin can perform all registered user actions plus:**
 - **Add Products**
 - **Update Product Details**
 - **Delete Products**
 - **Manage Orders**
 - **Oversee Payment Transactions**

4. Functional & Non-Functional Requirements

4.1 Functional Requirements

- User Authentication and Authorization
- Product Listing, Searching, and Filtering
- Shopping Cart and Wishlist
- Secure Checkout and Payment Processing
- Order Tracking and Management
- Product Reviews and Ratings
- Admin Panel for Product & Order Management

4.2 Non-Functional Requirements

- Performance: Support up to [X] concurrent users.
- Security: Secure data transmission and storage.
- Scalability: Efficiently scale with user demand.
- Reliability: Implement failover and rollback mechanisms.

5. Software Architecture

4.1 High-Level Design

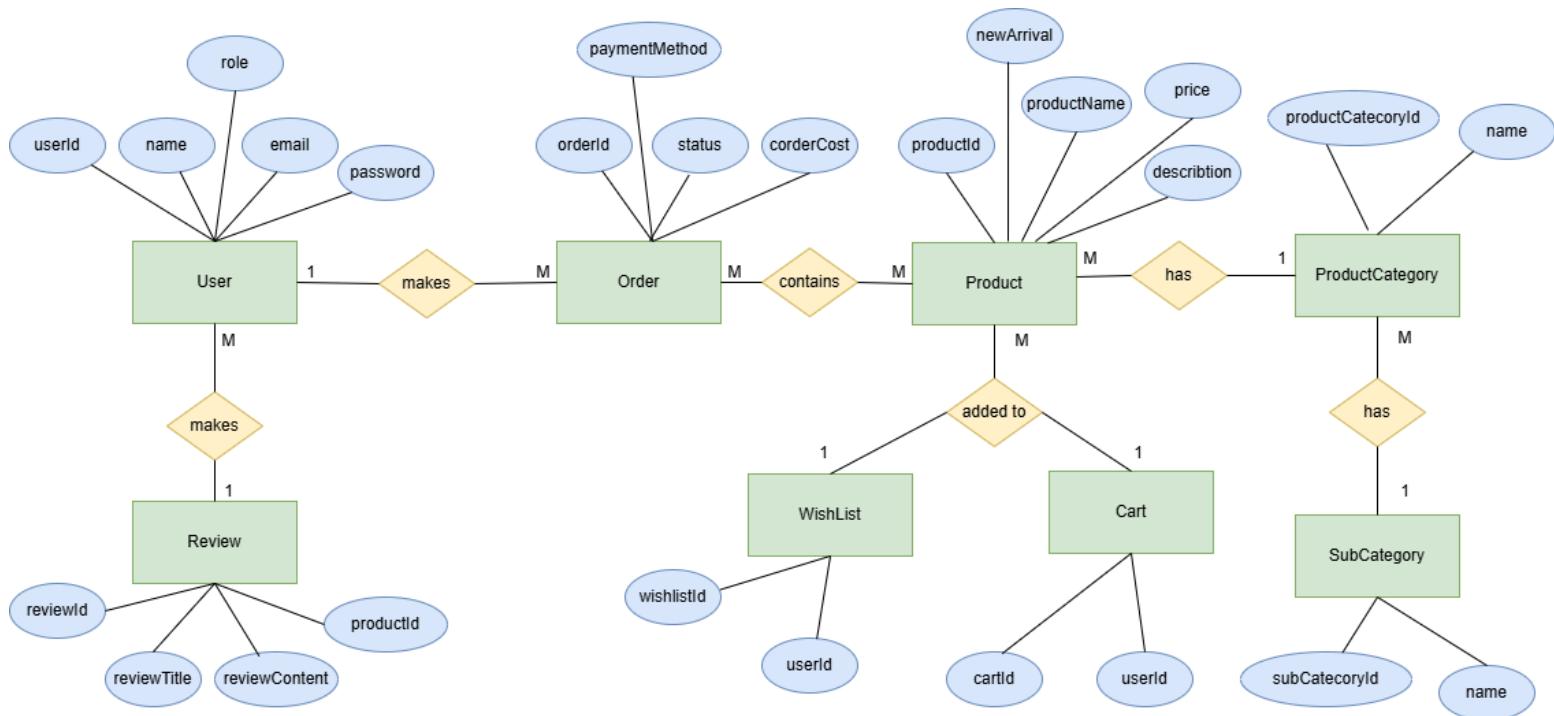
- **Frontend:** React.js /vite
- **Backend:** Node.js
- **Database:** MongoDB
- **DevOps Tools:** Jenkins, Docker, Kubernetes, Ansible

4.2 Architecture Style

- **Monolithic Architecture:** The entire application is deployed as a single containerized system for simpler management and deployment.

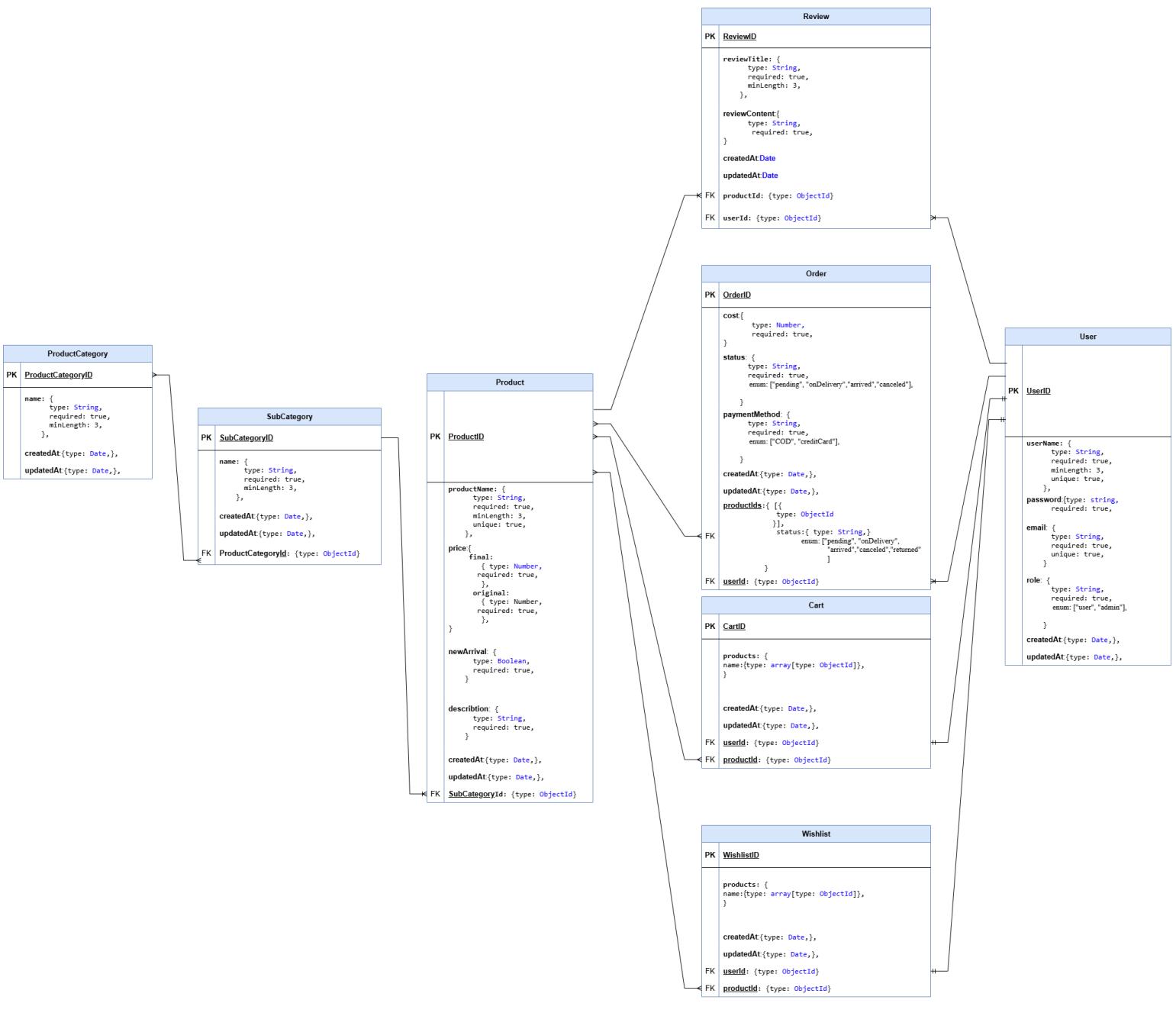
5. Database Design & Data Modeling

5.1 ER Diagram



5.2 Logical & Physical Schema

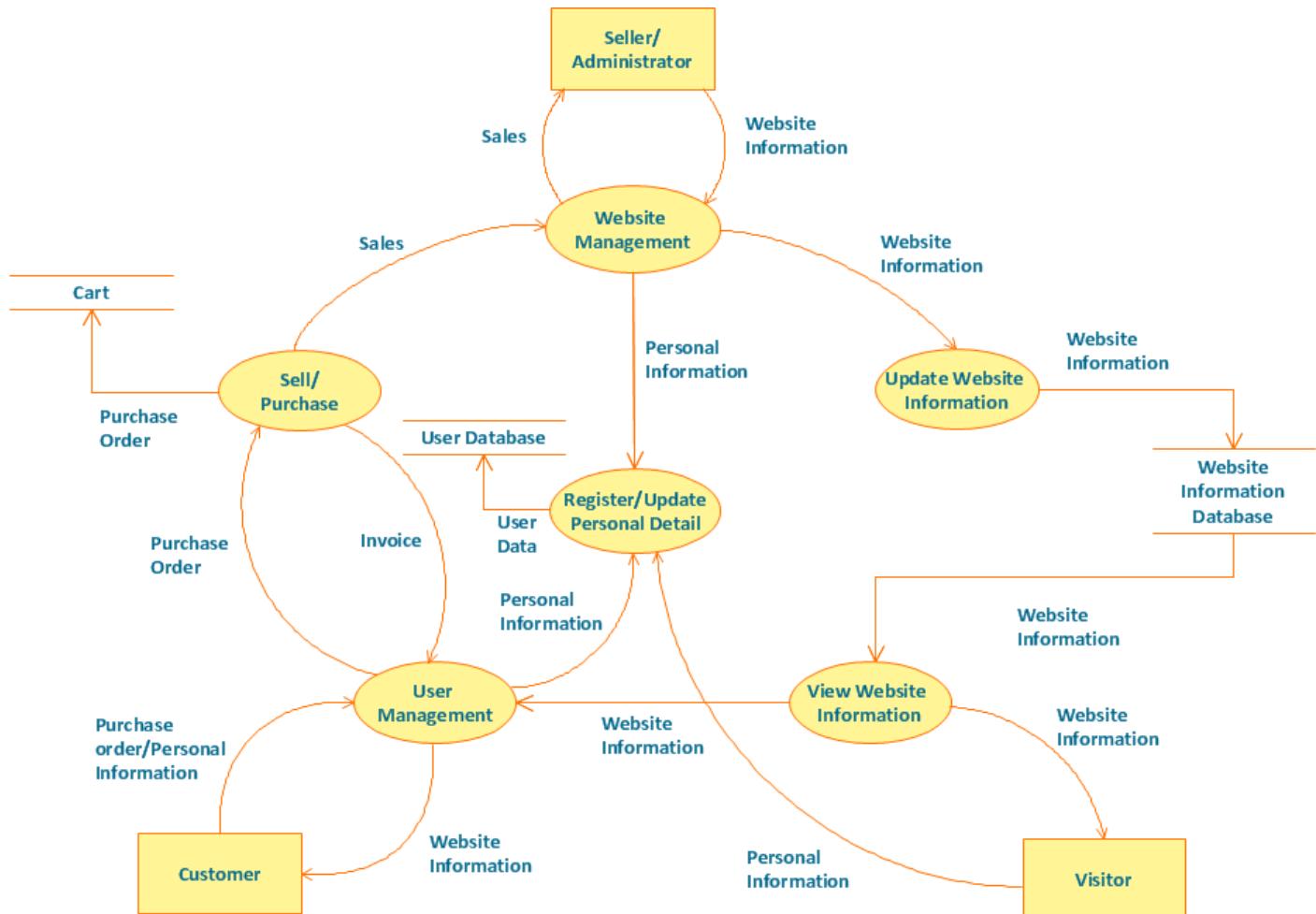
- Tables, attributes, keys, and normalization considerations.



6. Data Flow & System Behavior

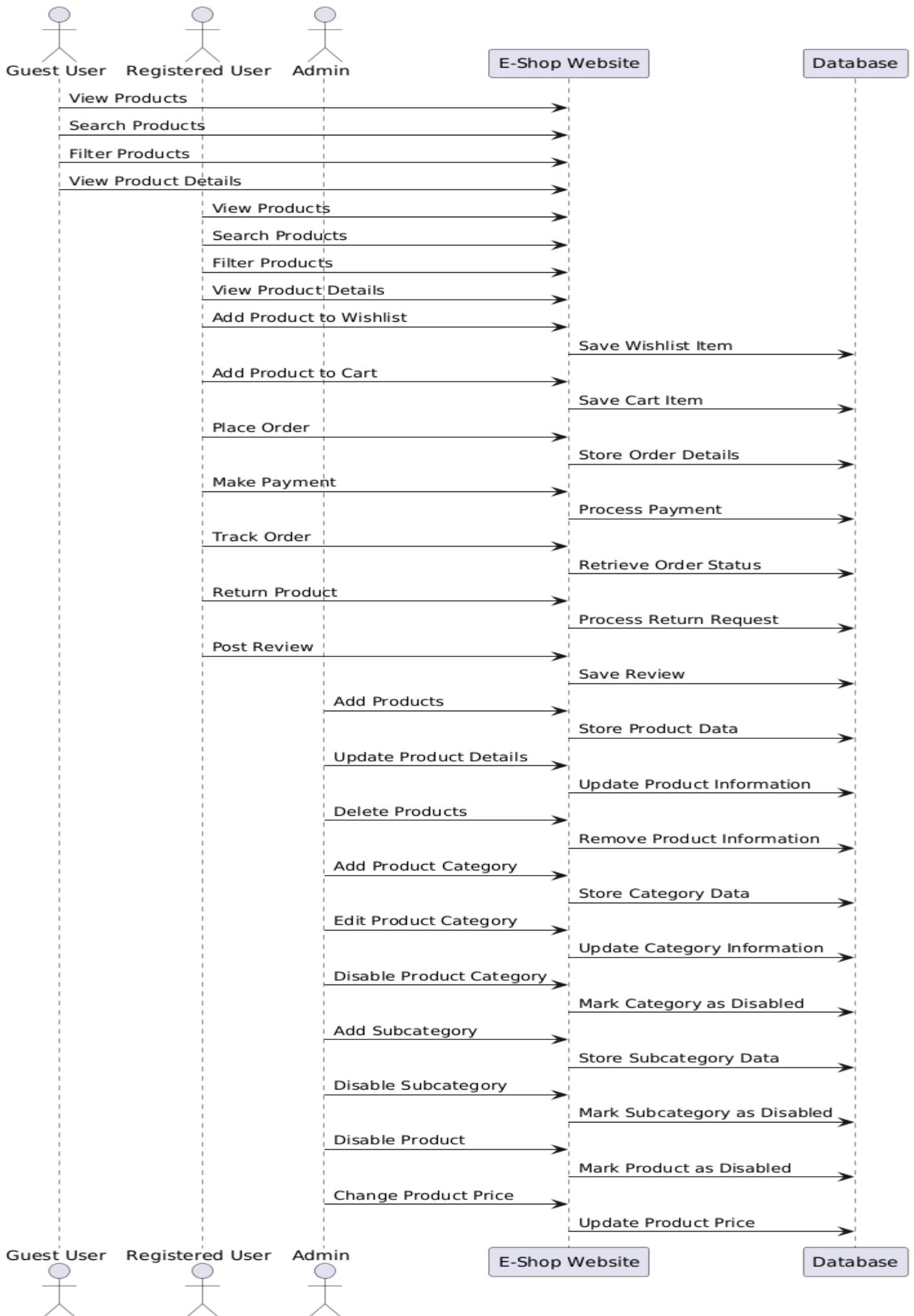
6.1 Data Flow Diagram (DFD)

- Context-level and detailed-level diagrams showcasing data movement.



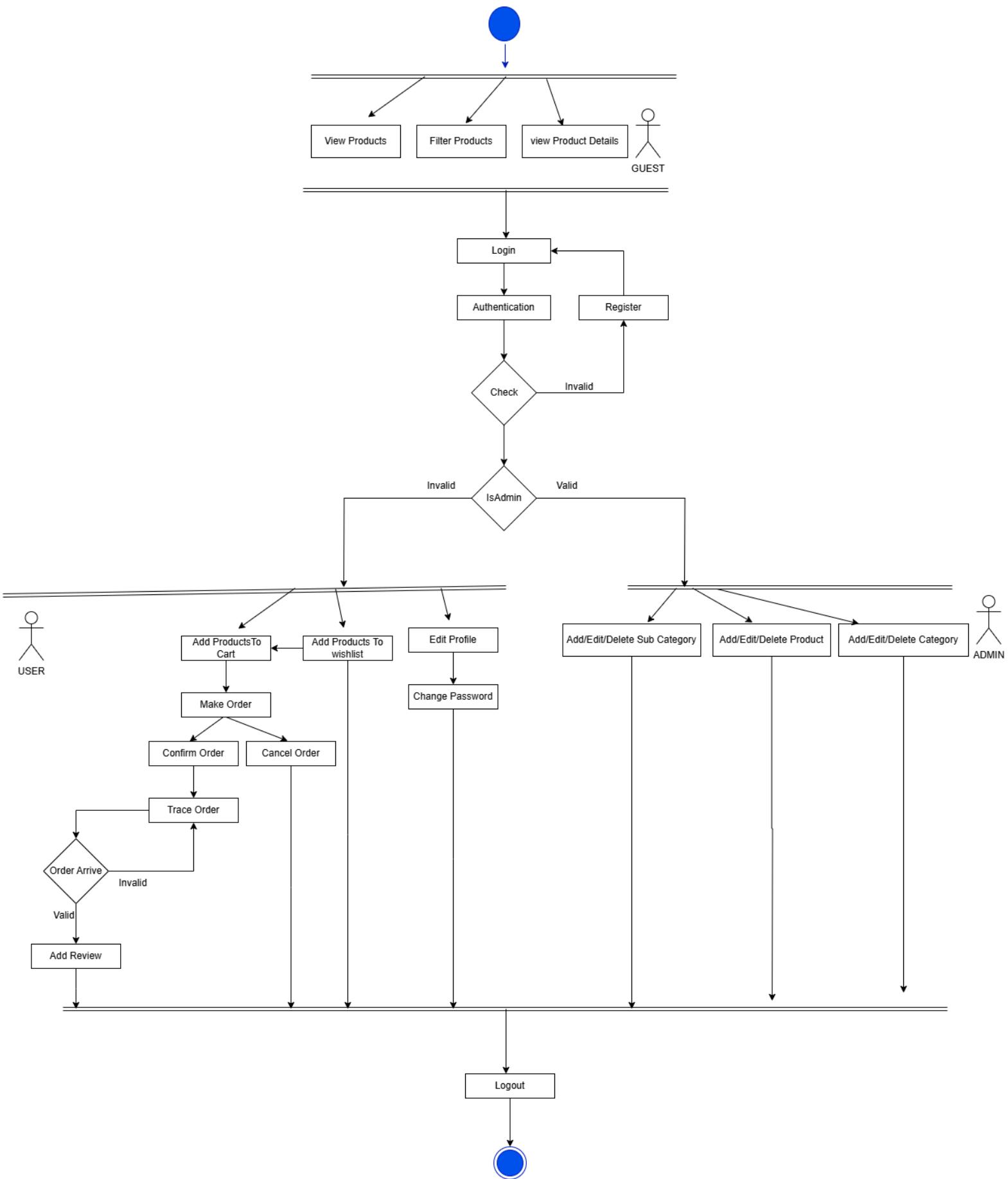
6.2 Sequence Diagrams

- Process flow representation of key interactions.



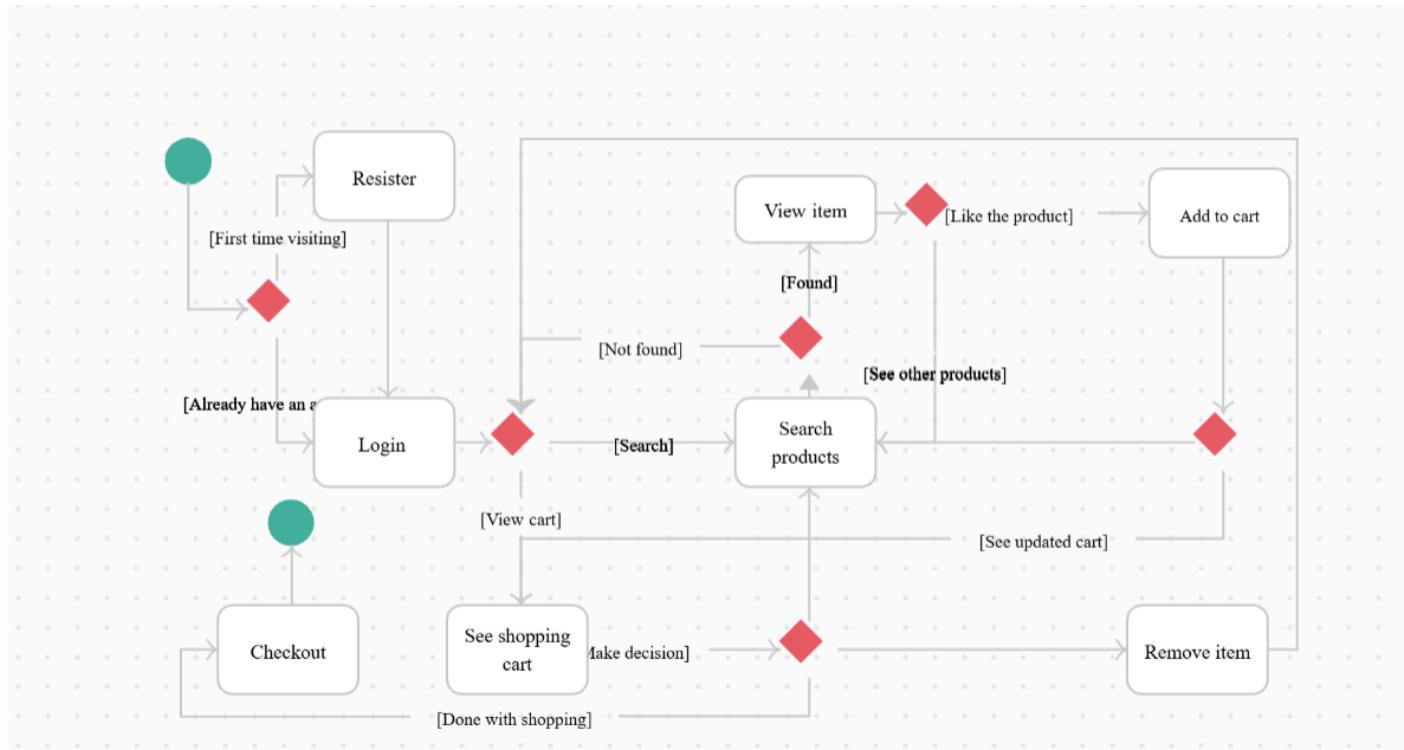
6.3 Activity Diagram

- Workflow visualization of system processes.



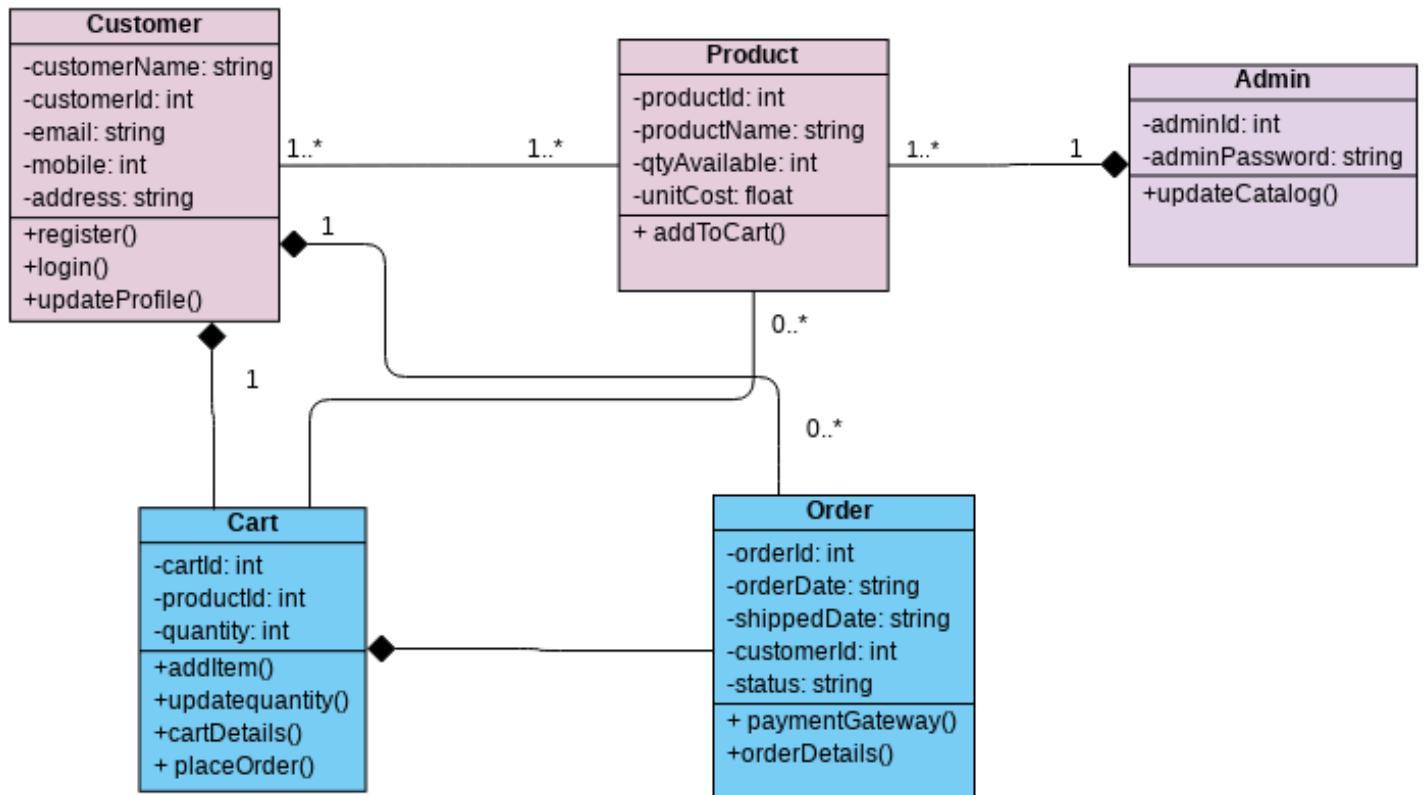
6.4 State Diagram

- Represents different states of system objects.



6.5 Class Diagram

- Defines system structure through classes, attributes, methods, and relationships.



7. UI/UX Design & Prototyping

7.1 Wireframes & Mockups

- Homepage & store page: Displays featured products, search bar, categories.
- Product Detail Page: Shows product images, descriptions, reviews, and “Add to Cart” button.
- Cart & Checkout: Summarizes items, totals, and captures shipping/payment info.
- Order Confirmation: Displays receipt, estimated delivery, and order tracking.

The screenshot shows a mobile application for an e-commerce platform named "ELEGANCE". The top navigation bar includes links for Home, About, Store, and Contact, along with icons for a heart (3 items), a shopping bag (5 items), and a user profile. The main banner features a close-up of men's suits and ties, with a central promotional overlay for a "Suits Offer" of up to 50% off, noting limited stock and unbeatable prices, and a "Shop Now" button. Below the banner, a section titled "Explore Our Categories" displays four categories with images and "Explore Now" buttons: "Fashion Women" (a woman in a red beret), "Fashion Men" (a rack of men's shirts), "Fashion Kids" (children's clothing), and "Health & Beauty" (various skincare and beauty products). At the bottom, a "New Arrivals" section shows four thumbnail images of different products.

ELEGANCE

Home About Store Contact

Heart (3) Bag (5) Profile

Suits Offer!
Up to 50% OFF!
Don't Wait - Limited Stock at Unbeatable Prices!
Shop Now

Explore Our Categories

Fashion Women
Explore the latest trends.
Explore Now →

Fashion Men
Explore the latest trends.
Explore Now →

Fashion Kids
Explore the latest trends.
Explore Now →

Health & Beauty
Explore the latest trends.
Explore Now →

New Arrivals

ELEGANCE

Home About Store Contact

NEED HELP?

If you have inquiries or need assistance, do not hesitate to chat with us.
We are available 7 Days a week between 9am and 8pm.

LIVE CHAT

You can also reach us on 123456.
From Sunday to Thursday, between 10am and 7pm.

Quick Assist

- How to make an order?
- How to track your order?
- How to cancel your order?
- How to return your order?

Contact Us

Enter your name
Enter your email
Enter your message

Send Message

ELEGANCE

Elegance is your go-to destination for the latest trends in fashion. We offer high-quality clothing and accessories to elevate your style.

Quick Links

Home Privacy Policy
About Terms of Service
Store FAQ
Contact Us Shipping Info

Customer Service

1128 456-7890
support@fashionelegance.com

All rights reserved. Elegance © 2025.

Explore Our Store

Fashion Women

- Bags
- Clothes
- Watches
- Shoes

Fashion Men

- Clothes
- Shoes
- Watches
- Sunglasses

Fashion Kids

- Clothes
- Shoes
- Toys
- Accessories

ELEGANCE

Home About Store Contact

Your Cart

- Product 1**
\$19.99
- Product 2**
\$29.99
- Product 3**
\$9.99

Cart Summary

Total Items: 6
Total Cost: \$99.94

Checkout

7.2 UI/UX Guidelines

- Design principles, color schemes, typography, and accessibility considerations.
-

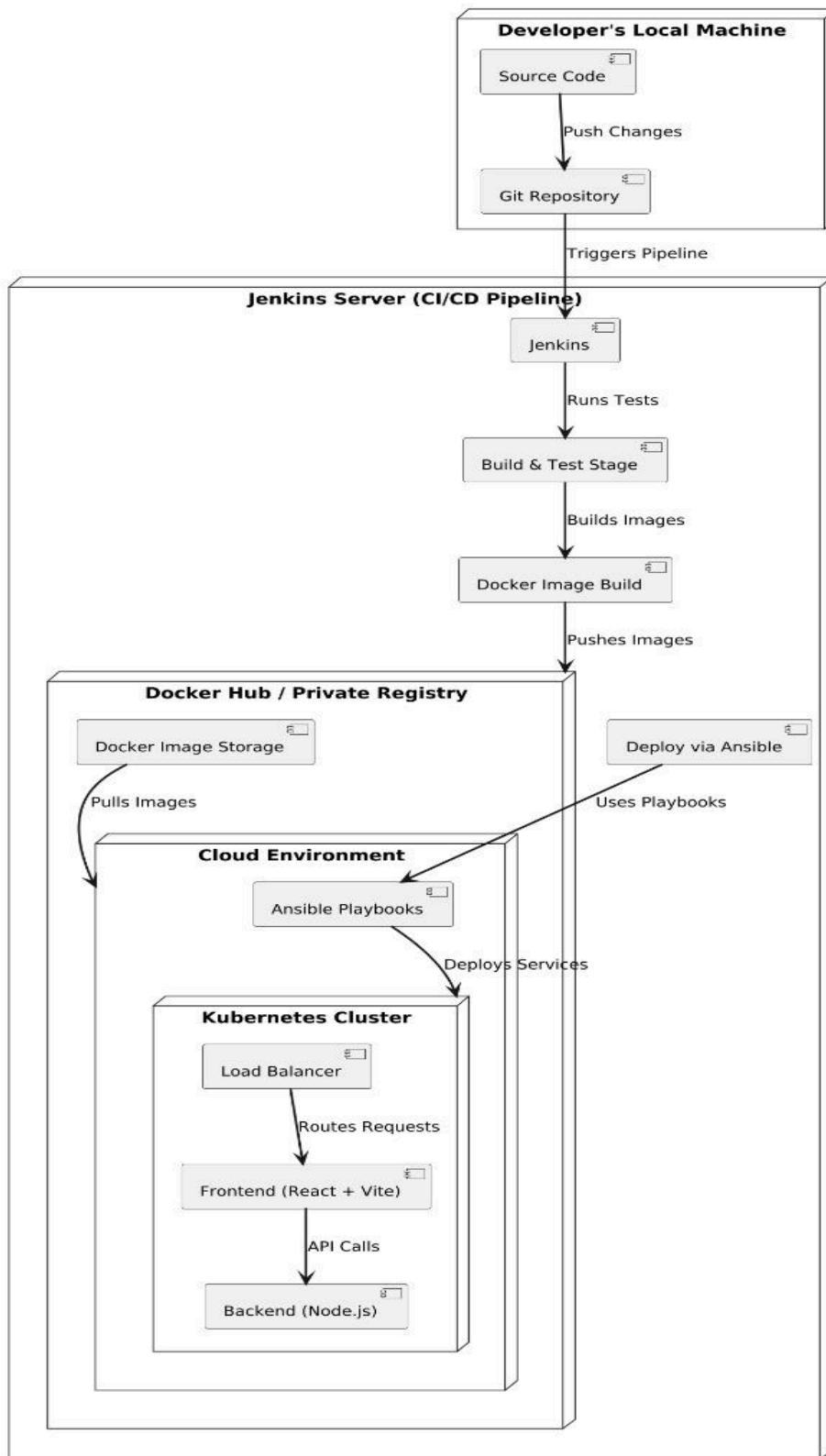
8. System Deployment & Integration

8.1 Technology Stack

- **Frontend:** React + Vite
- **Backend:** Node.js
- **Database:** MongoDB
- **DevOps:** Jenkins (CI/CD), Docker (containers), Ansible (configuration management), Kubernetes (orchestration)

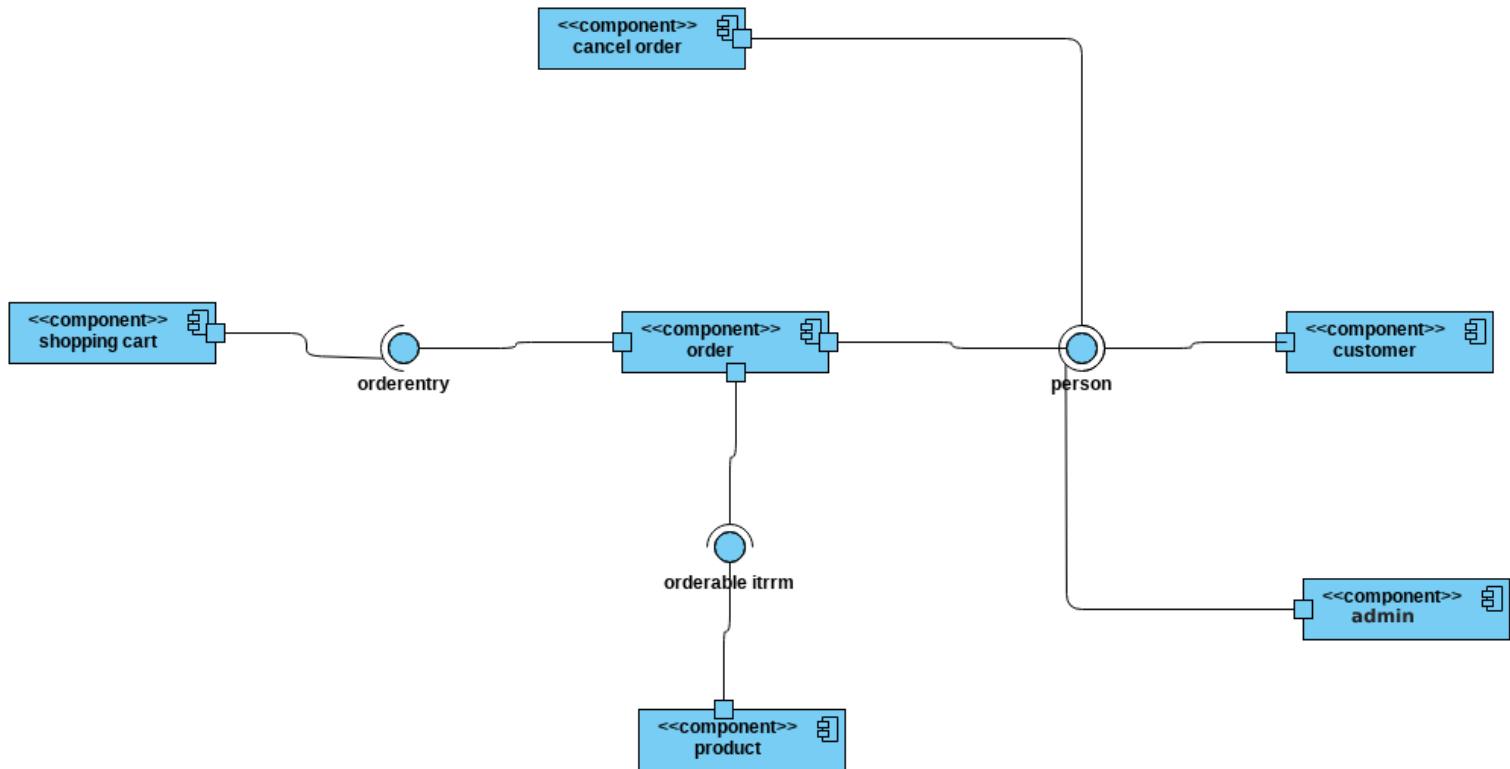
8.2 Deployment Diagram

- **Jenkins:** Polls Git repository for changes.
- **Docker:** Builds images for frontend & backend.
- **Ansible (optional):** Manages infrastructure configs or application environment.
- **Kubernetes:** Deploys Docker containers, manages scaling, load balancing, and service discovery.



8.3 Component Diagram

- High-level system components and their dependencies.



9. Additional Deliverables

9.1 API Documentation

- If the system includes APIs, provide documentation for endpoints and usage.

9.2 Testing & Validation

- Unit tests, integration tests, and user acceptance testing plan.

9.3 Deployment Strategy

- **Staging Environment:** Automated pipeline triggers build and test. If successful, deploy to staging.
- **Production Environment:** Manual approval or automated if tests pass. Rolling updates or blue-green deployment.
- **Monitoring & Logging:** Integrate with tools like Prometheus, Grafana, or ELK Stack for real-time metrics and logs.

9.4 Maintenance & Future Improvements

- **Automated Rollback:** If a deployment fails health checks, revert to the last stable version.
- **Infrastructure as Code:** Use Terraform or Ansible for consistent environment provisioning.
- **Security Hardening:** Regular scans for vulnerabilities, SSL certificates, and secrets management.

10. Conclusion

This documentation outlines the system analysis and design for an automated DevOps deployment of an e-commerce web application. By adhering to these guidelines—covering architecture, database design, DevOps integration, and UI/UX—the project will be well-structured, maintainable, and scalable.