

# **TechXpress E-commerce Platform Project**

# **Project Planning & Management:**

# 1. Project Proposal

## 1.1 Overview

TechXpress is a scalable and maintainable **e-commerce web application** for selling electronics. It will enable users to browse products, add them to a shopping cart, **and complete purchases** using integrated payment gateways. The platform includes an **admin panel** for product, category, and order management.

## 1.2 Objectives

- Develop a full-featured e-commerce platform using **ASP.NET Core**.
- Implement **NTier Architecture** for maintainability and scalability.
- Ensure **secure authentication** and role-based access control.
- Provide **efficient order and payment processing** with Stripe integration.
- Deliver a **user-friendly UI** and responsive design.
- Deploy the application to **Microsoft Azure**.

## 1.3 Scope

- **Customer features:** Browsing products, cart management, checkout, order tracking.
- **Admin features:** Product, category, and order management.
- **Security:** ASP.NET Identity-based authentication and role management.
- **Payments:** Secure payment processing via Stripe.
- **Technologies:** ASP.NET Core MVC, Entity Framework Core, Microsoft Azure, Stripe API, JQuery, Bootstrap.

## 2. Project Plan

### 2.1 Timeline (Gantt Chart)

Week	Task
Week 1: Initial Setup and Product Listings	<ul style="list-style-type: none"><li>• NTier Architecture Setup</li><li>• Database Design</li><li>• Repository Pattern Implementation</li><li>• User Authentication Setup</li></ul>
Week 2: Shopping Cart, Role-Based Access Control, and Admin Panel	<ul style="list-style-type: none"><li>• Shopping Cart Functionality</li><li>• Role-Based Access Control (RBAC)</li><li>• Admin Dashboard Development</li></ul>
Week 3: Order Placement, Stripe Payment Integration, and User Profiles	<ul style="list-style-type: none"><li>• Order Placement</li><li>• Stripe Payment Integration</li><li>• User Profiles and Order History</li></ul>
Week 4: Final Testing, UI Enhancements, and Deployment	<ul style="list-style-type: none"><li>• UI Enhancements</li><li>• DataTables Integration</li><li>• Final Testing and Bug Fixes</li><li>• Deployment</li></ul>

2.2 Milestones

- Week 1: Architecture and Authentication Setup Completed
- Week 2: Shopping Cart and Admin Panel Functional
- Week 3: Order Placement and Payment Integration Completed
- Week 4: Final Deployment and Documentation

2.3 Deliverables

- Fully functional e-commerce platform with admin panel.
- Integrated Stripe payment gateway.
- Responsive UI and live deployment.

2.4 Resource Allocation

- Developer 1: Backend Development (Business Logic, Data Access)
- Developer 2: Frontend Development (UI, Admin Panel)
- QA Engineer: Testing and Bug Fixes
- Project Manager: Timeline and Deliverables Oversight

3. Task Assignment & Roles

Role	Responsibilities
Project Manager	<ul style="list-style-type: none"><li>● Oversee project progress and ensure timely delivery of milestones.</li><li>● Coordinate between team members and stakeholders.</li><li>● Manage risk assessment and mitigation plans.</li></ul>

Backend Developer	<ul style="list-style-type: none"> <li>• Implement the Business Logic Layer and Data Access Layer using ASP.NET Core and Entity Framework Core.</li> <li>• Develop repositories and the Unit of Work pattern for database interactions.</li> <li>• Integrate Stripe payment gateway for secure transactions.</li> </ul>
Frontend Developer	<ul style="list-style-type: none"> <li>• Design and implement the Presentation Layer using ASP.NET Core MVC, Bootstrap, and custom CSS.</li> <li>• Enhance UI/UX with JQuery, DataTables, and Toaster JS for dynamic and interactive elements.</li> <li>• Ensure responsiveness and cross-browser compatibility.</li> </ul>
Database Administrator	<ul style="list-style-type: none"> <li>• Design and manage the database schema using Entity Framework Core.</li> <li>• Optimize database queries and ensure data integrity.</li> <li>• Monitor database performance and scalability.</li> </ul>
QA Engineer	<ul style="list-style-type: none"> <li>• Perform comprehensive testing of all functionalities, including shopping cart, role-based access control, and order placement.</li> <li>• Ensure the system meets performance and security standards.</li> <li>• Document and report bugs for resolution.</li> </ul>
DevOps Engineer	<ul style="list-style-type: none"> <li>• Set up and manage deployment pipelines for Microsoft Azure or Hostinger.</li> <li>• Monitor system uptime and performance post-deployment.</li> <li>• Ensure smooth deployment and rollback processes.</li> </ul>

## 4. Risk Assessment & Mitigation Plan

Risk	Mitigation
Delays in development due to unclear requirements	Conduct detailed requirement analysis and regular stakeholder meetings to clarify doubts.
Payment gateway integration issues	Use Stripe's official documentation and sandbox environment for thorough testing before production deployment.
Security vulnerabilities in user authentication and payment processing	Implement ASP.NET Identity for secure authentication and follow OWASP guidelines for secure coding practices.
Performance issues with high user traffic	Optimize database queries, use caching mechanisms, and perform load testing.
Deployment failures or downtime	Use CI/CD pipelines and maintain a backup of the application and database for quick recovery.

## 5. Key Performance Indicators (KPIs)

KPI	Metric
Response Time	Ensure the average response time for user actions (e.g., browsing products, adding to cart) is under 2 seconds.
System Uptime	Maintain a system uptime of 99.9% post-deployment.
User Adoption Rate	Achieve a 70% user adoption rate within the first three months of deployment.
Error Rate	Keep the error rate below 1% for all functionalities.
Admin Efficiency	Ensure admin tasks (e.g., adding products, managing orders) can be completed within 2 minutes using the admin dashboard.

Payment Success Rate	Maintain a payment success rate of 95% or higher through Stripe integration.
-------------------------	--

## Literature Review:



## 1. Feedback & Evaluation

The **TechXpress E-commerce Platform** will be assessed based on the following criteria:

- **Project feasibility:** How well the project aligns with the given objectives and scope.
- **Code quality:** Maintainability, scalability, and adherence to best practices.
- **Functionality:** Implementation of core features such as product listings, shopping cart, order processing, and payment integration.
- **User experience:** Responsiveness, ease of navigation, and overall UI/UX design.
- **Security measures:** Authentication, role-based access control, and payment security.

## 2. Suggested Improvements

The project can be further enhanced by:

- **Adding AI-powered recommendations:** Implementing machine learning to suggest products based on user behavior.
- **Multi-vendor support:** Allowing third-party sellers to list products.
- **Mobile App Integration:** Expanding the platform to mobile devices using Flutter.
- **Performance optimization:** Using caching and CDN for faster page loads.
- **Additional payment methods:** Supporting PayPal, Apple Pay, or cryptocurrency payments.

### 3. Final Grading Criteria

Category	Percentage (%)	Assessment Criteria
Documentation	20%	Completeness, clarity, and structure of project reports.
Implementation	40%	Functional correctness, code quality, and best practices.
Testing & Debugging	20%	Unit testing, bug fixes, and security measures.
Presentation	20%	Clarity of explanation, UI/UX design, and demonstration quality.
Documentation	20%	Completeness, clarity, and structure of project reports.

## **Requirements Gathering:**

# 1. Stakeholder Analysis

Below is a breakdown of primary stakeholders, their roles, and expectations:

Stakeholder	Role	Needs & Expectations
Customers	End-users who browse and purchase products.	<ul style="list-style-type: none"><li>● User-friendly and responsive UI.</li><li>● Secure payment options (Stripe, PayPal).</li><li>● Order tracking and history.</li><li>● Personalized recommendations.</li></ul>
Admin	Manages product listings, categories, and orders.	<ul style="list-style-type: none"><li>● Intuitive admin panel for easy management.</li><li>● Role-based access control for security.</li><li>● Data analytics and sales reports.</li></ul>
Developers	Backend & frontend engineers building the platform.	<ul style="list-style-type: none"><li>● Clear architecture (NTier, Repository Pattern).</li><li>● Maintainable and scalable codebase.</li><li>● API documentation for integrations.</li></ul>
QA Testers	Ensure system reliability and security.	<ul style="list-style-type: none"><li>● Automated testing framework.</li></ul>

		<ul style="list-style-type: none"> <li>● Functional, performance, and security testing.</li> </ul>
Business Owners	Investors or company executives.	<ul style="list-style-type: none"> <li>● Scalability for future expansion.</li> <li>● Revenue growth via optimized checkout flows.</li> <li>● Customer retention strategies</li> </ul>
Payment Providers	Third-party payment gateways (Stripe, PayPal).	<ul style="list-style-type: none"> <li>● Secure transactions.</li> <li>● Compliance with PCI-DSS standards.</li> <li>● Reliable API integrations.</li> </ul>
Hosting Provider	Microsoft Azure for deployment.	<ul style="list-style-type: none"> <li>● 99.9% uptime guarantee.</li> <li>● Scalable cloud infrastructure.</li> <li>● Security and data privacy compliance.</li> </ul>

## 2. User Stories:

### 1-User Authentication & Account Management.

- As a customer, I want to sign in to my account so that I can access my personal information.

- As a customer, I want to manage my account so that I can update my personal details and preferences.
- As an admin, I want to log in securely so that I can manage the platform's operations.

## **2-Product Browsing & Wishlist.**

- As a customer, I want to search for products so that I can quickly find items I am interested in.
- As a customer, I want to view product details so that I can learn more about the items before purchasing.
- As a customer, I want to add products to my wishlist so that I can save them for future purchases.

## **3-Shopping Cart & Order Management.**

- As a customer, I want to add products to my cart so that I can review them before purchasing.
- As a customer, I want to view my shopping cart so that I can see what I have selected.
- As a customer, I want to update the quantity of items in my cart so that I can adjust my order before checkout.
- As a customer, I want to make an order so that I can complete my purchase.
- As a customer, I want to cancel an order before it is shipped so that I can change my decision.

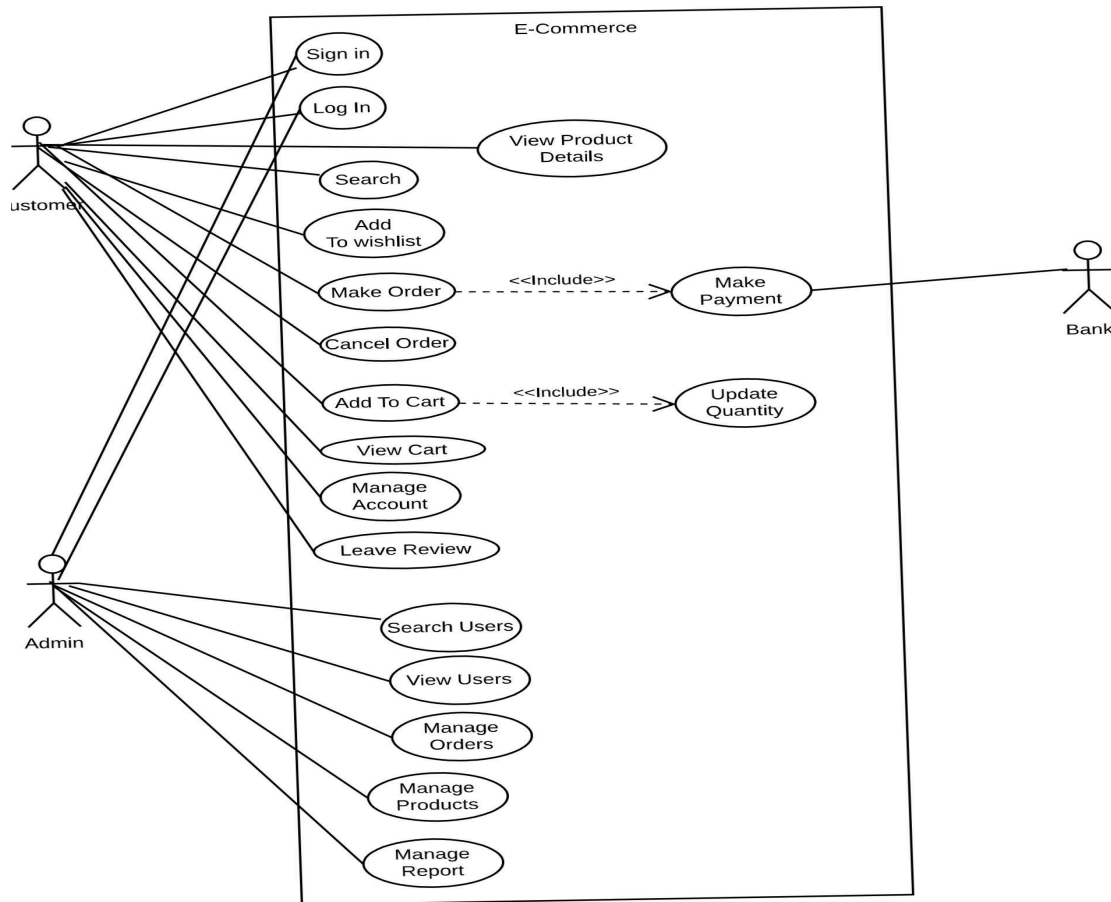
## **4-Payment Processing**

- As a customer, I want to make a payment securely so that I can complete my order transaction.
- As a bank system, I want to process payments so that customers' orders are validated and completed.

### **5-Reviews & Feedback**

- As a customer, I want to leave a review on a product so that I can share my experience with other shoppers.

## **3. Usecase Diagram**



## 4. Functional Requirements

### 1. User Authentication & Account Management

- The system must allow customers and admins to log in securely.
- Users must be able to manage their accounts (update personal details, change passwords).
- The system must validate login credentials before granting access.

### 2. Product Browsing & Wishlist

- Users must be able to search for products using keywords.



- Users must be able to view detailed product information (name, price, description, stock availability).
- Users must be able to add products to their wishlist for future reference.

### **3. Shopping Cart & Order Management**

- Customers must be able to add products to their shopping cart.
- Users must be able to update item quantities in the cart before checkout.
- Customers must be able to remove items from their cart.
- The system must allow customers to place orders.
- Customers must be able to cancel orders before they are shipped.

### **4. Payment Processing**

- The system must support secure online payments through an integrated payment gateway.
- Payment transactions must be validated before order confirmation.
- The system must store transaction details for future reference.

### **5. Reviews & Feedback**

- Customers must be able to leave reviews on purchased products.
- Reviews must be displayed on product pages for other users to see.

### **6. Admin Operations**

- Admins must be able to manage products (add, update, delete).
- Admins must be able to manage orders (view, update status, cancel if necessary).

- Admins must be able to search and view user details.
- Admins must be able to generate and manage reports related to sales and user activity.

## **5. Non-Functional Requirements**

### **1. Performance & Scalability**

- The system must handle at least 1,000 concurrent users without performance degradation.
- Product searches must return results within 2 seconds.

### **2. Security**

- The system must encrypt user passwords using a secure hashing algorithm (e.g., bcrypt).
- Payments must be processed using HTTPS and PCI DSS-compliant payment gateways.
- Admin functionalities must only be accessible to authorized users.

### **3. Usability & Accessibility**

- The platform must be user-friendly and have a responsive design for mobile and desktop users.
- Users must be able to navigate the platform easily with minimal learning effort.

### **4. Reliability & Availability**

- The system must be available 99.9% of the time.
- Automated database backups must be performed daily to prevent data loss.

## **5. Maintainability & Extensibility**

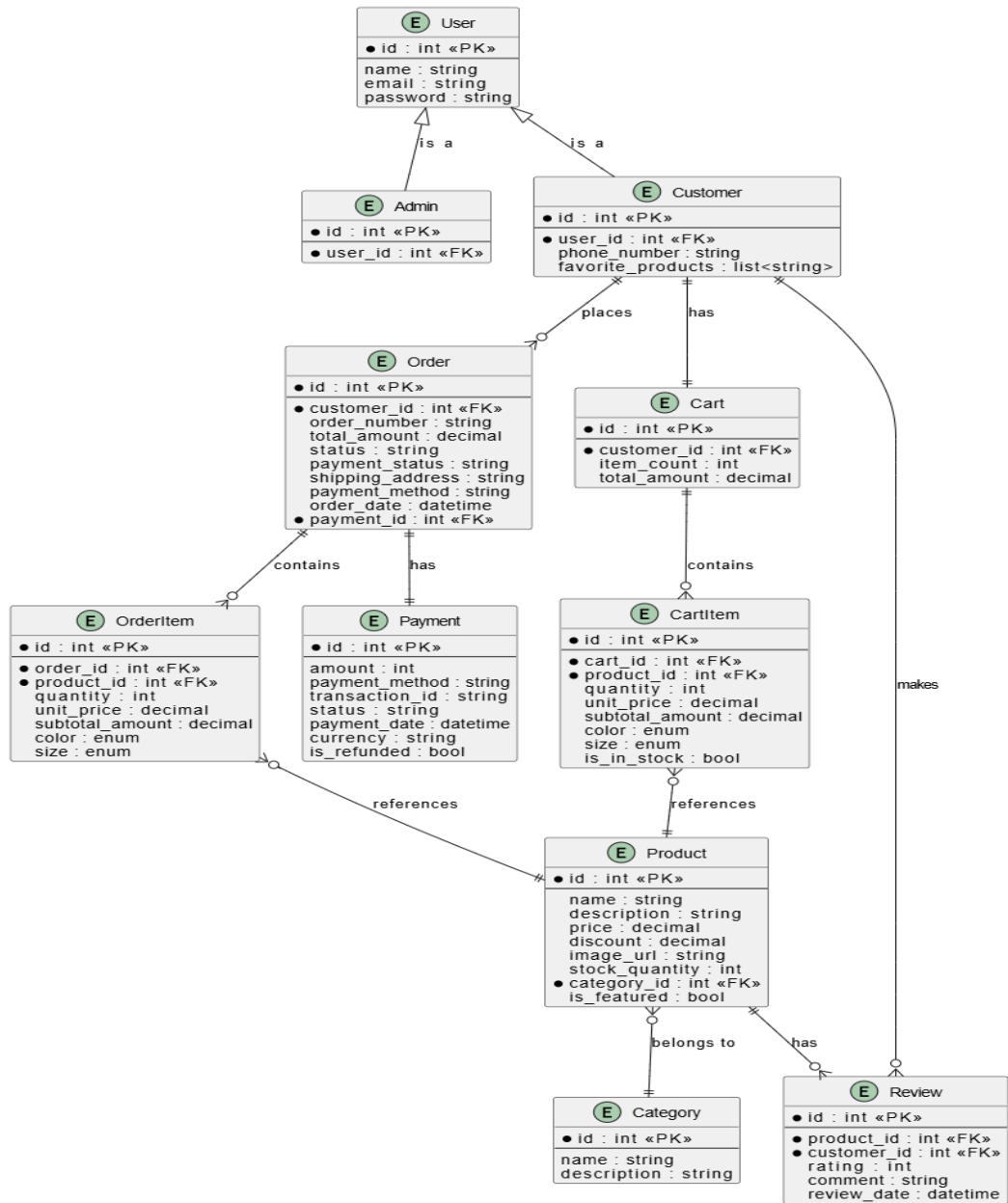
- The system must follow a modular architecture (NTier, Repository Pattern) to allow future enhancements.
- The platform must support easy integration with third-party services (e.g., new payment gateways).

## **6. Compliance**

- The platform must comply with GDPR to handle user data.
- The payment system must meet financial regulations such as PCI DSS.

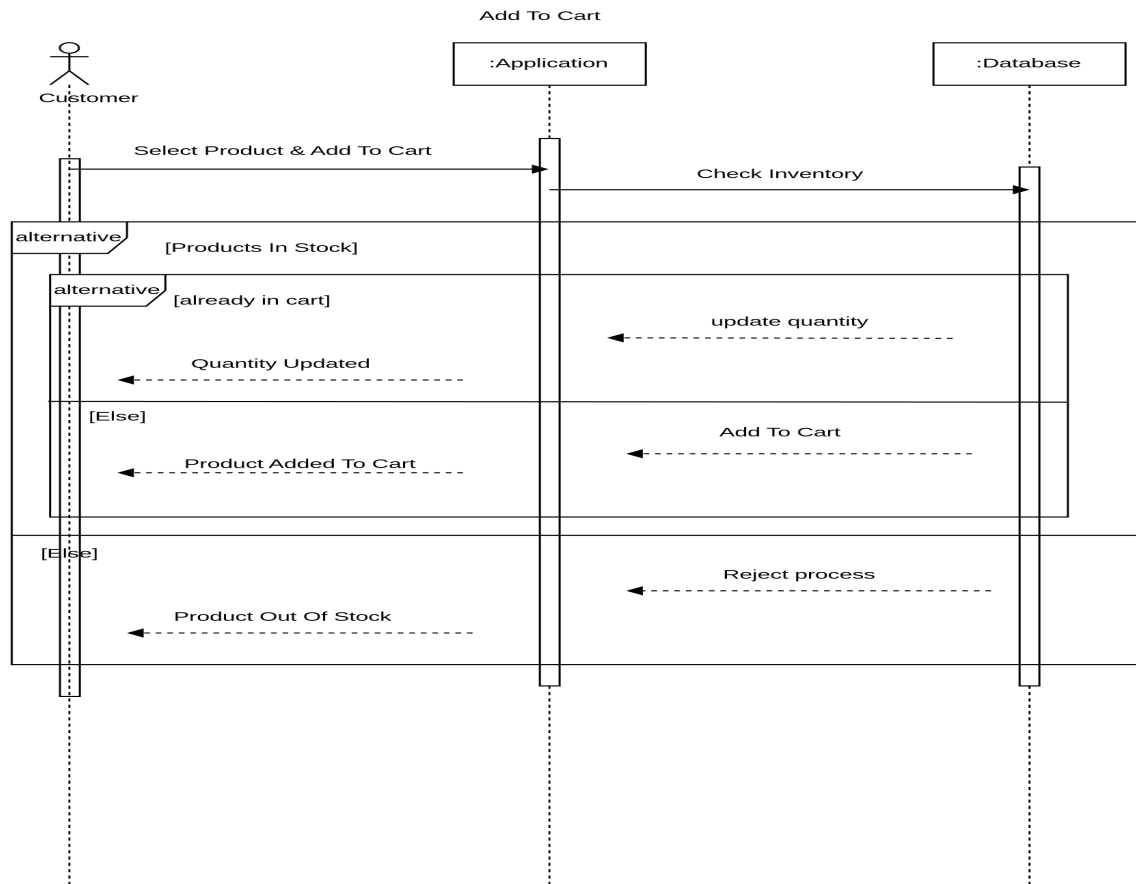
# **System Analysis & Design:**

## 1. ERD:

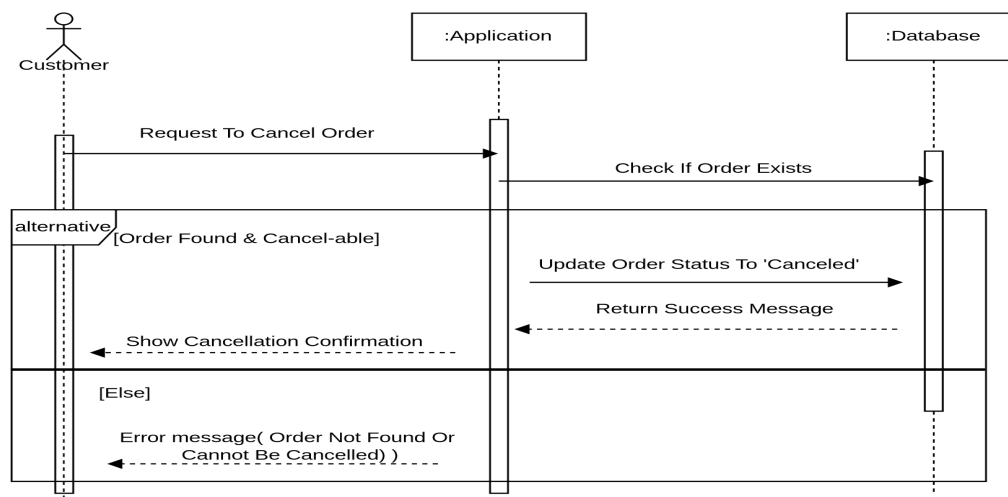


### 3. Sequence Diagrams :

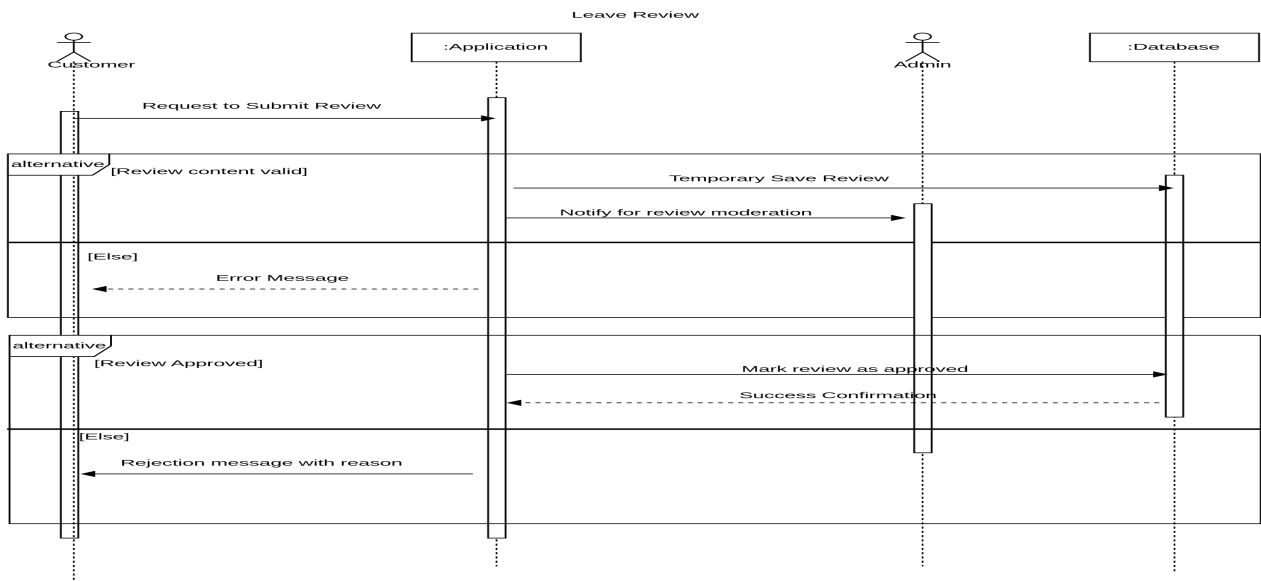
#### 3.1 Add Cart Sequence



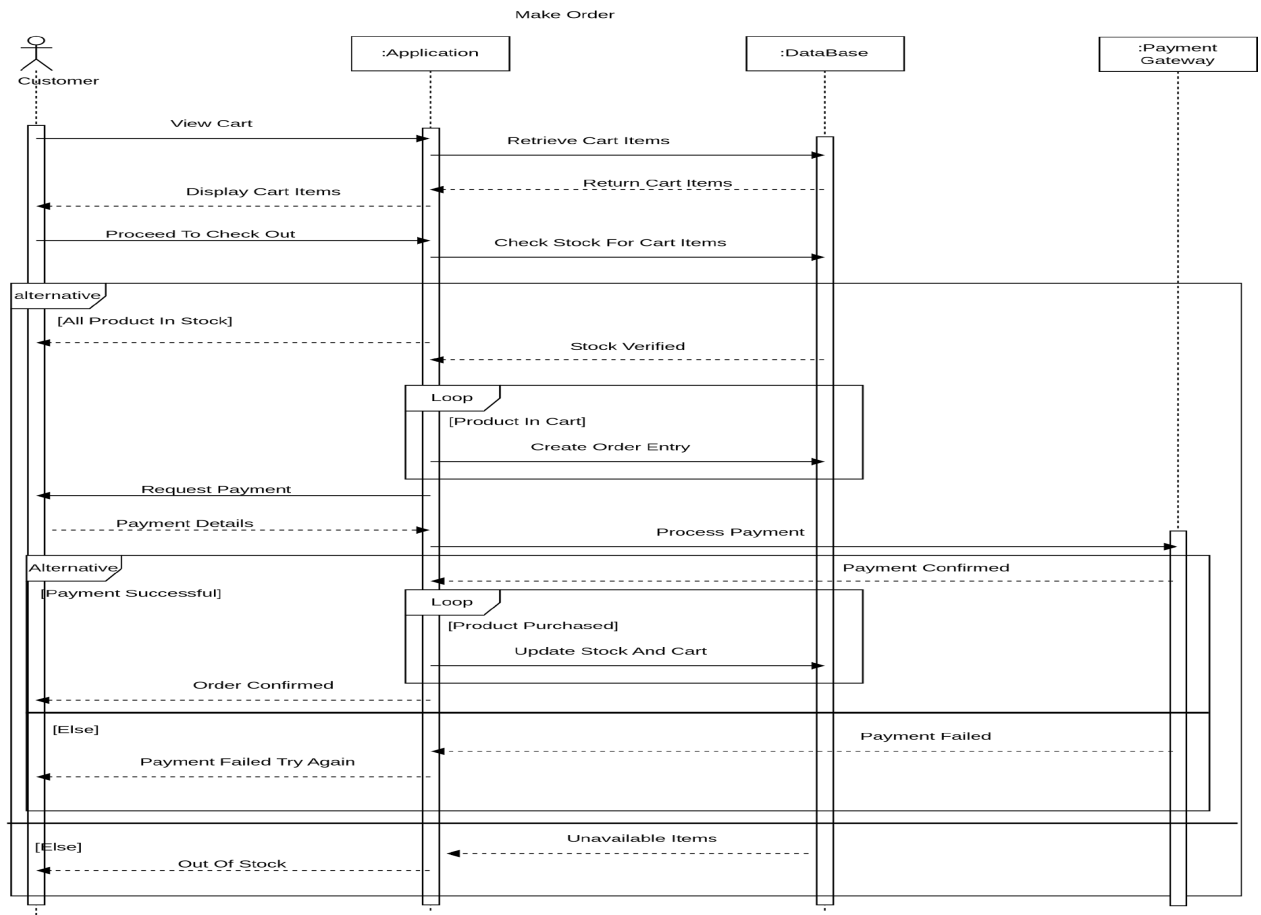
#### 3.2 Cancel Order Sequence



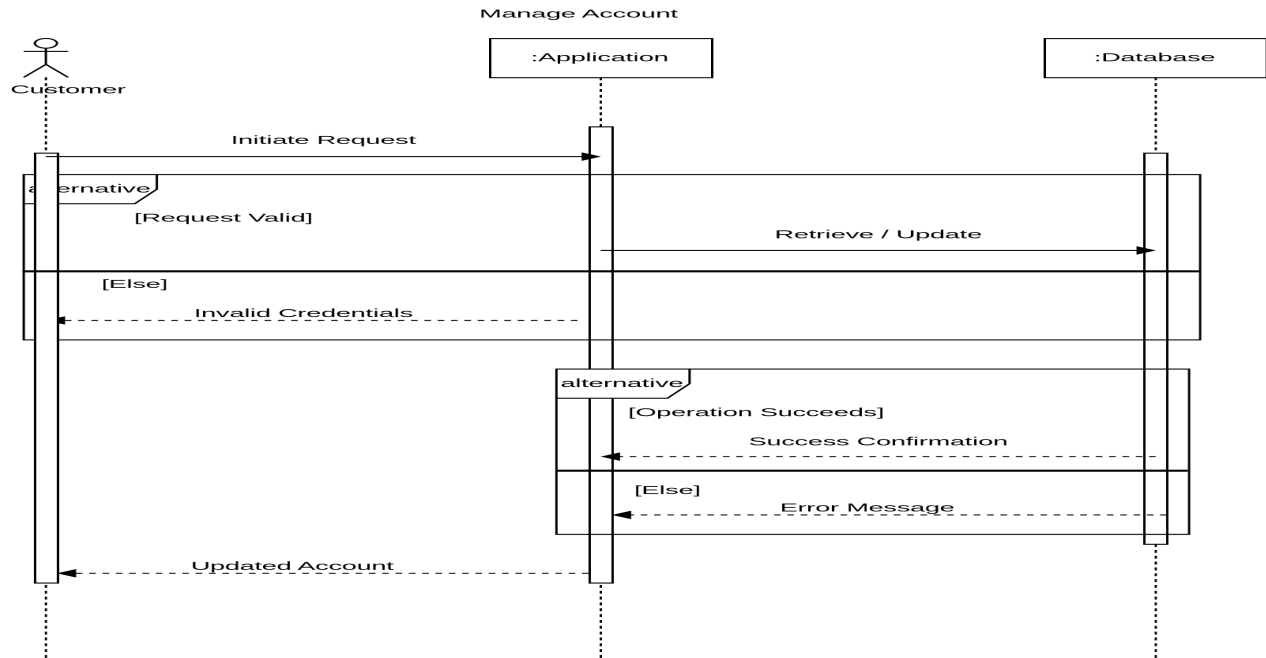
### 3.3 Leave Review Sequence



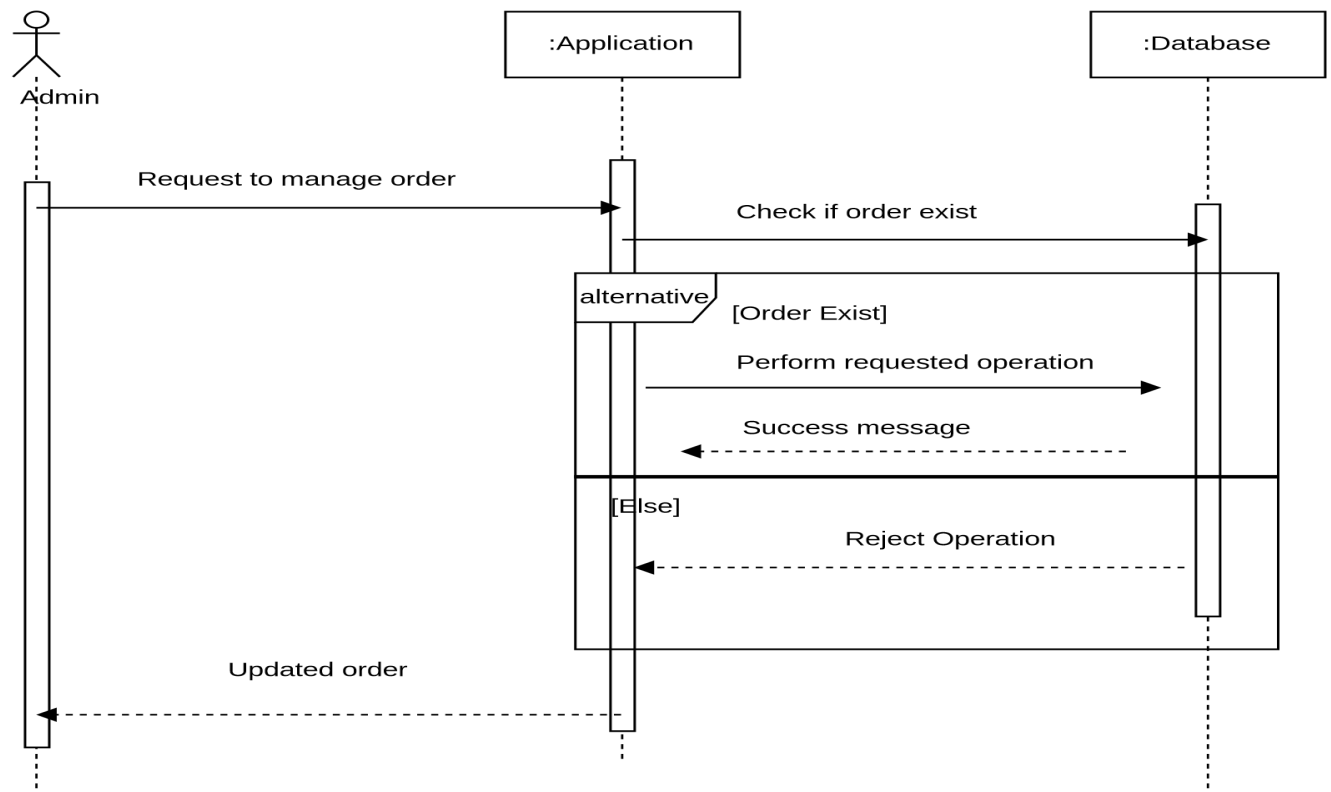
### 3.4 Make Order Sequence



### 3.5 Manage Account Sequence

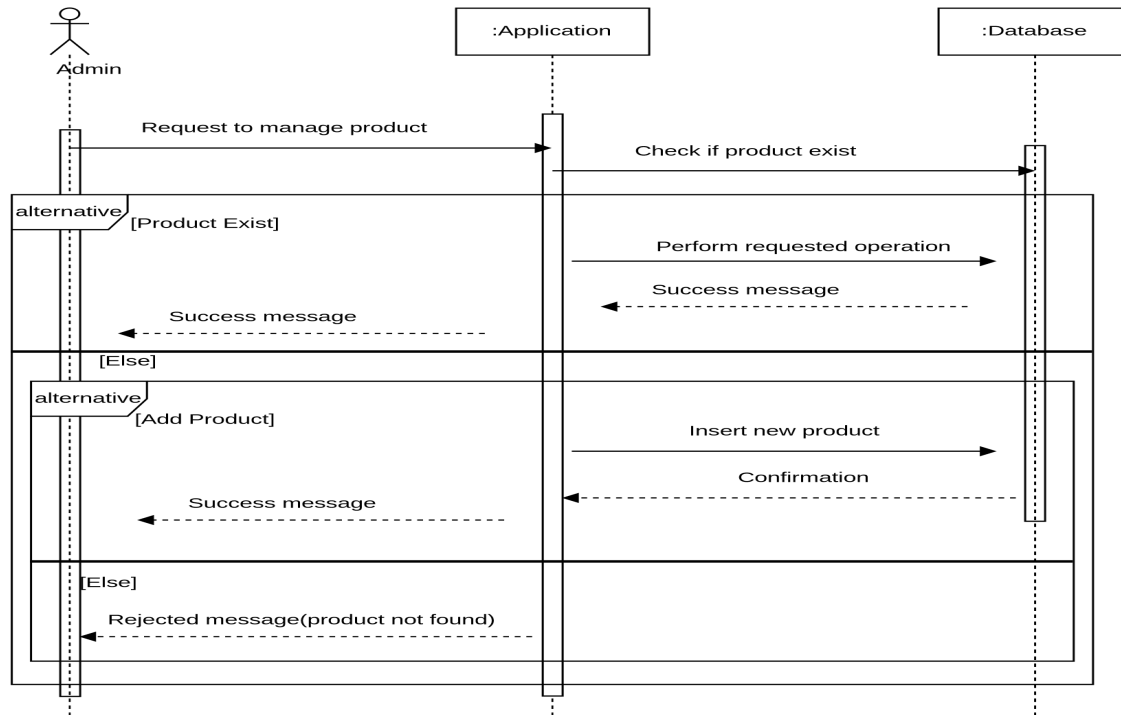


### 3.6 Manage Order Sequence

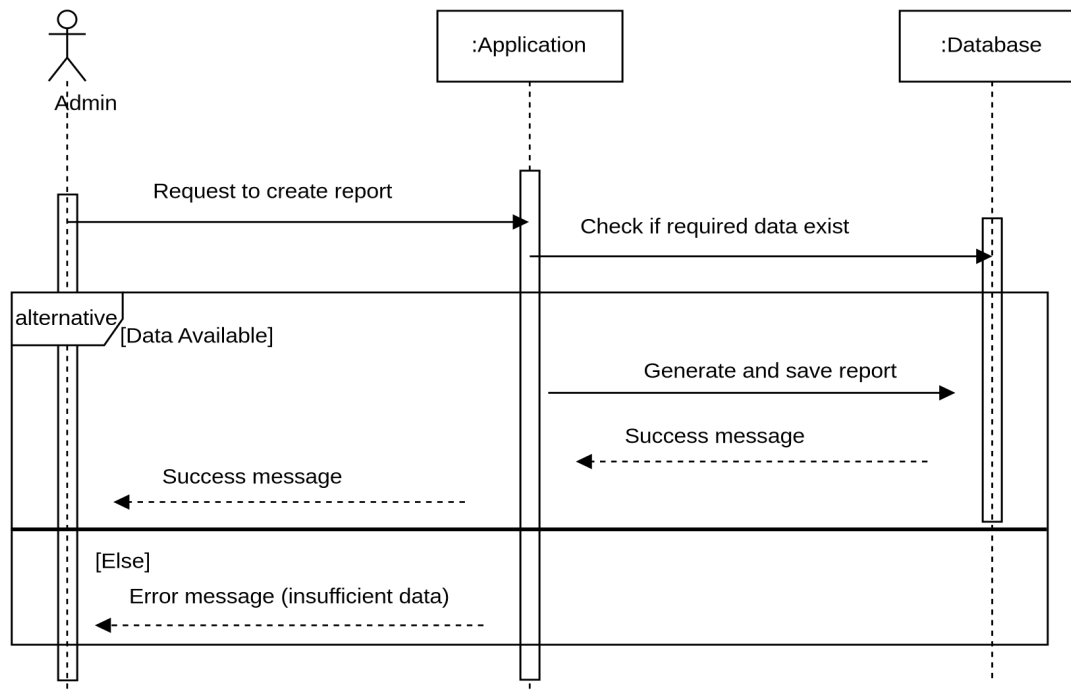




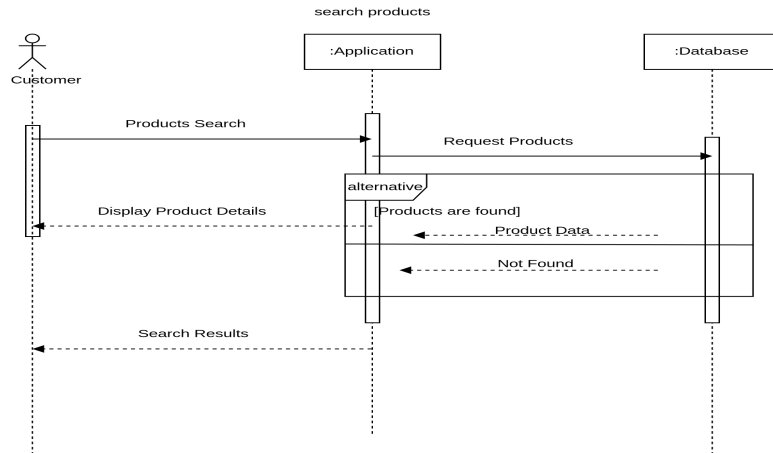
### 3.7 Manage Product



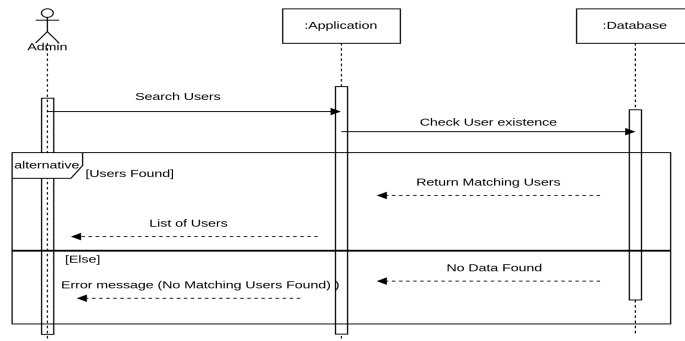
### 3.8 Manage Report Sequence



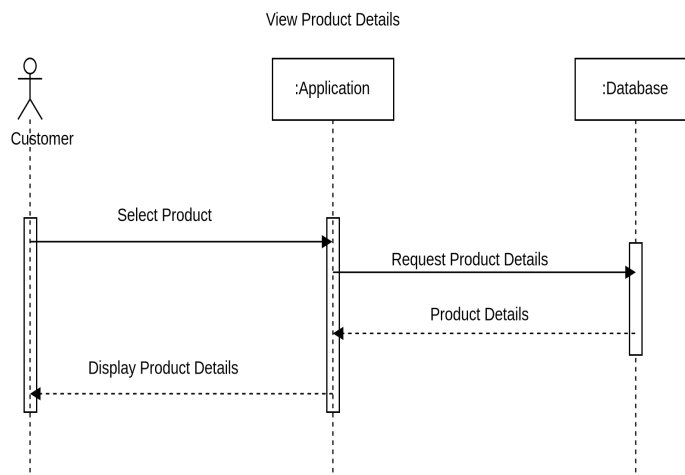
### 3.9 Search Products Sequence



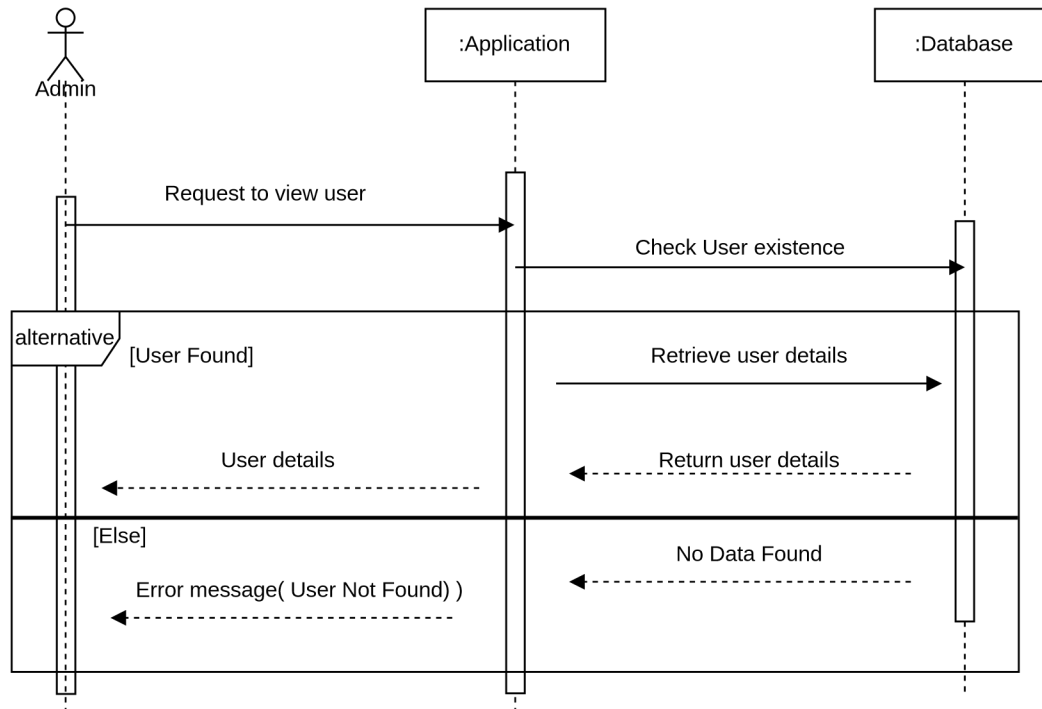
### 3.9 Search User Sequence



### 3.10 View Product Sequence

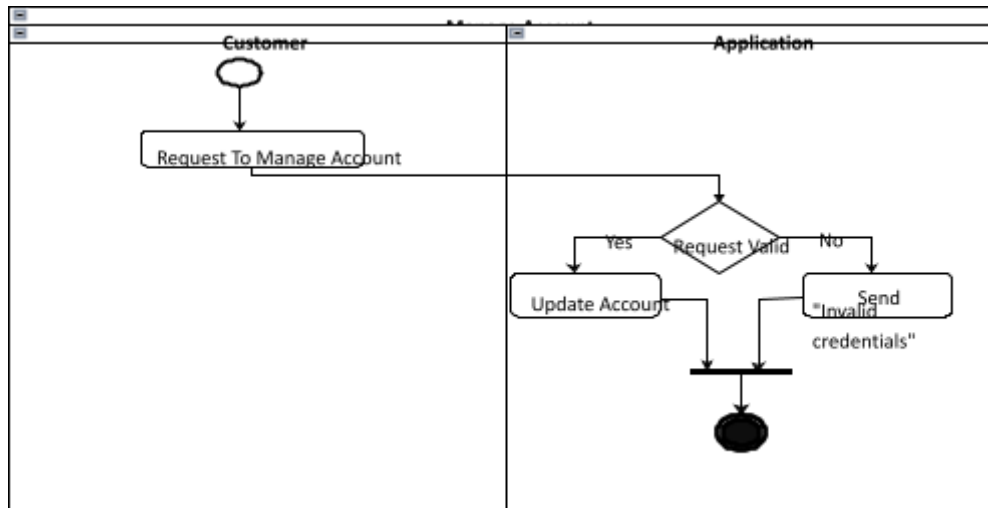


### 3.11 View Users Sequence

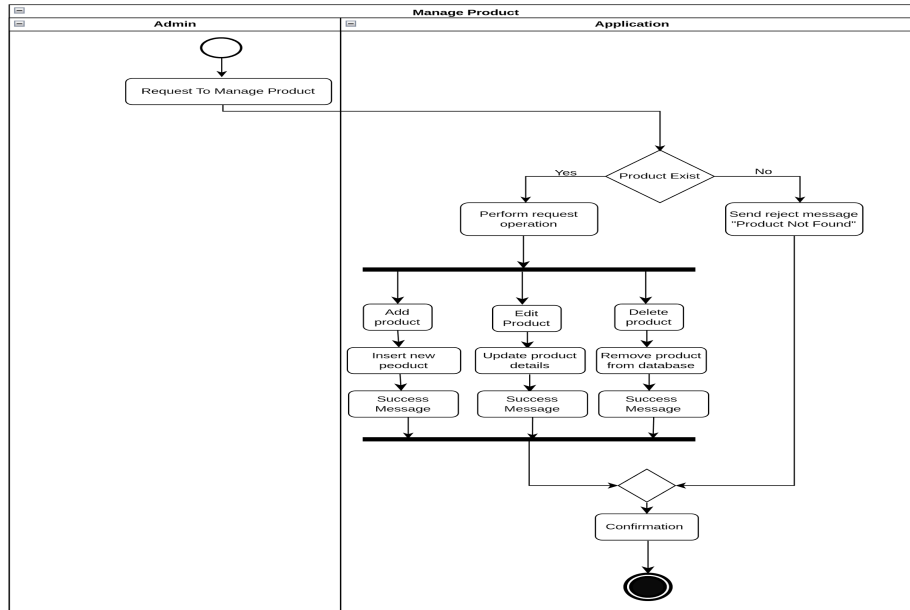


## 4. Activity Diagram

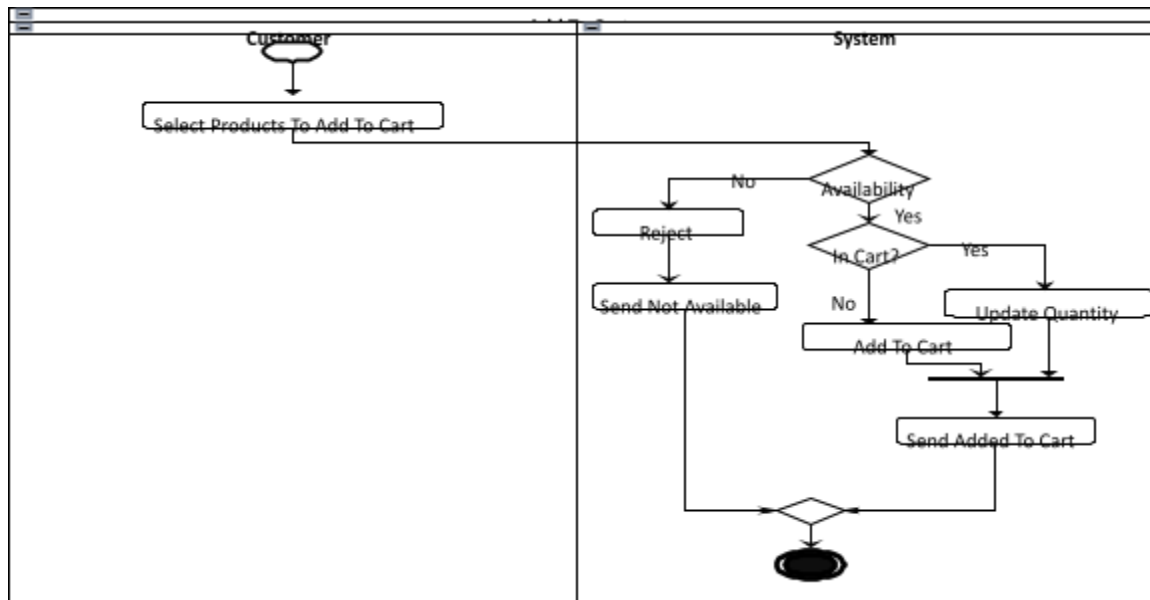
### 4.1 Manage Account Activity



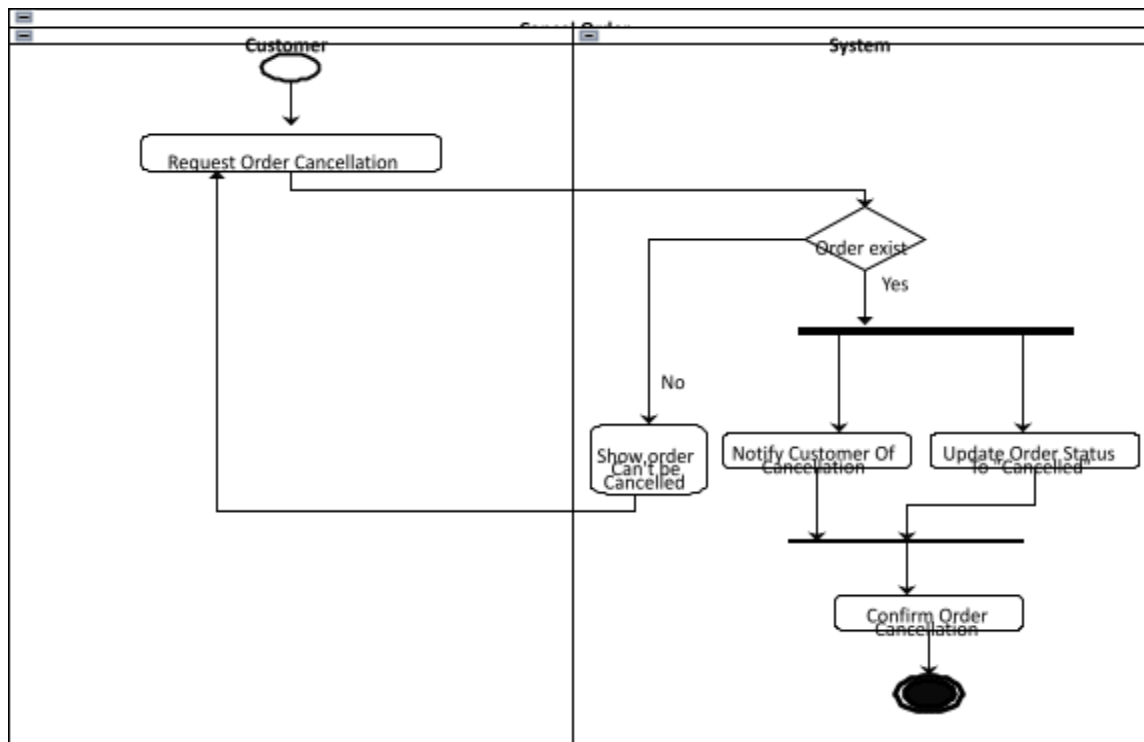
### 4.2 Manage Products Activity



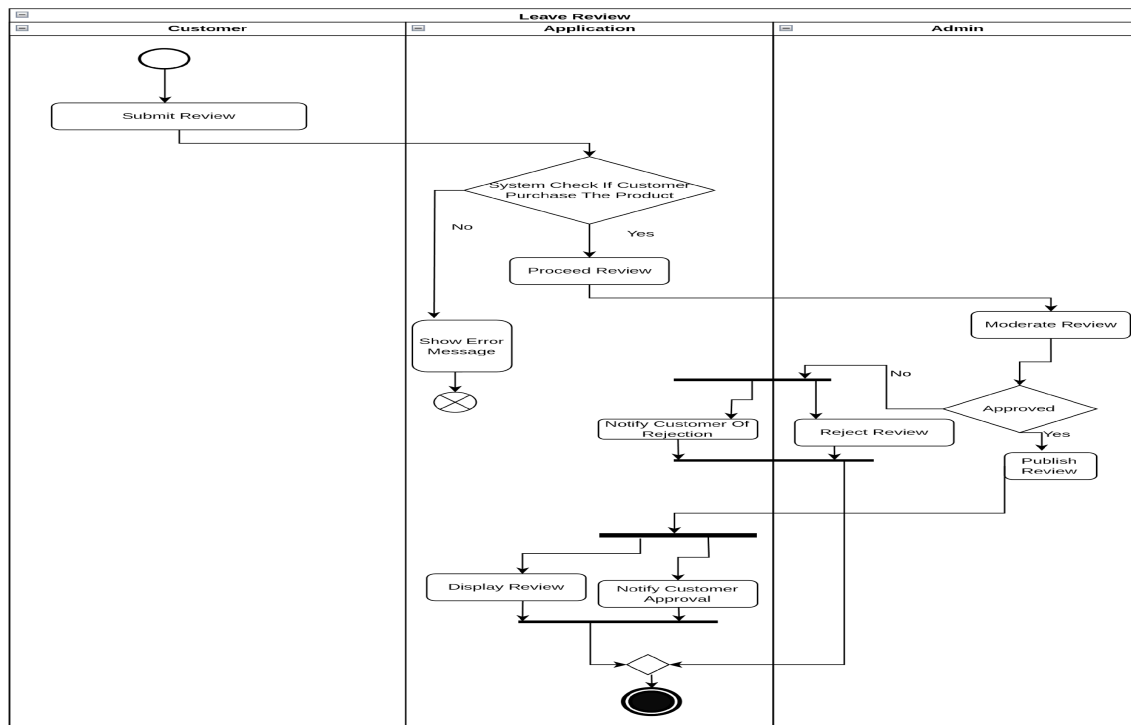
### 4.3 addTo Cart Activity



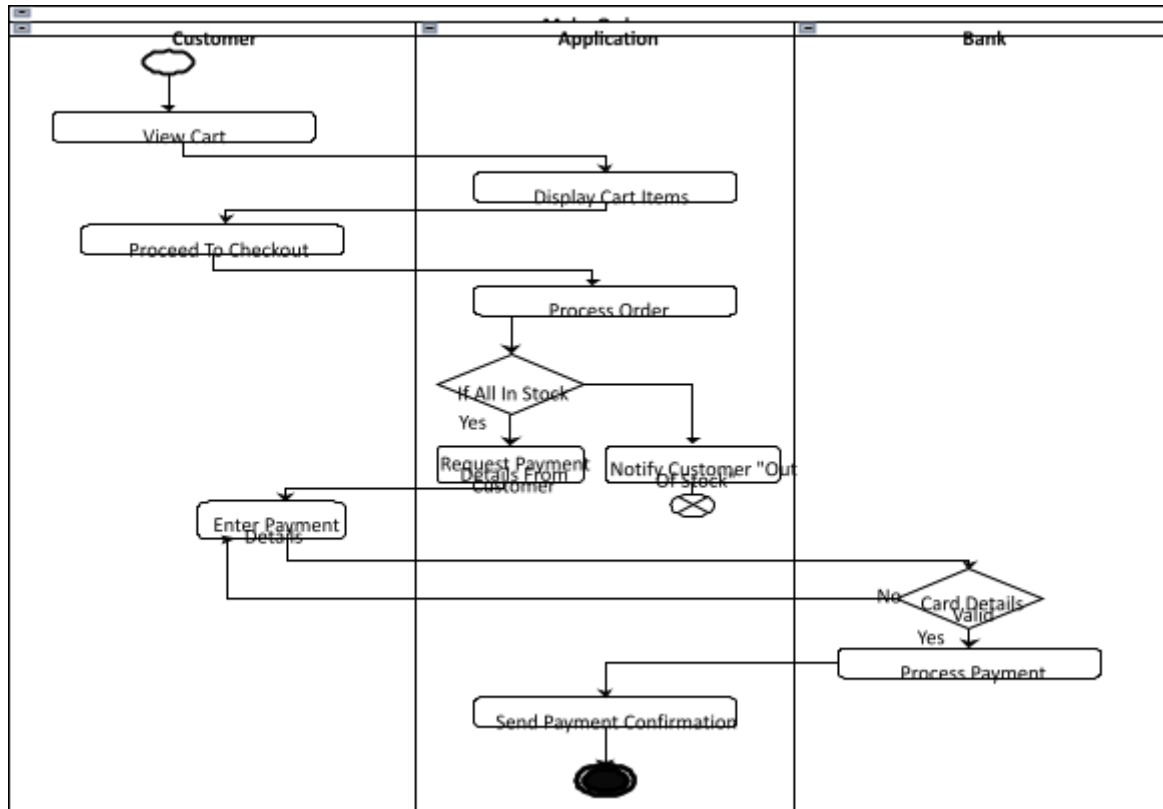
### 4.4 Cancel Order Activity



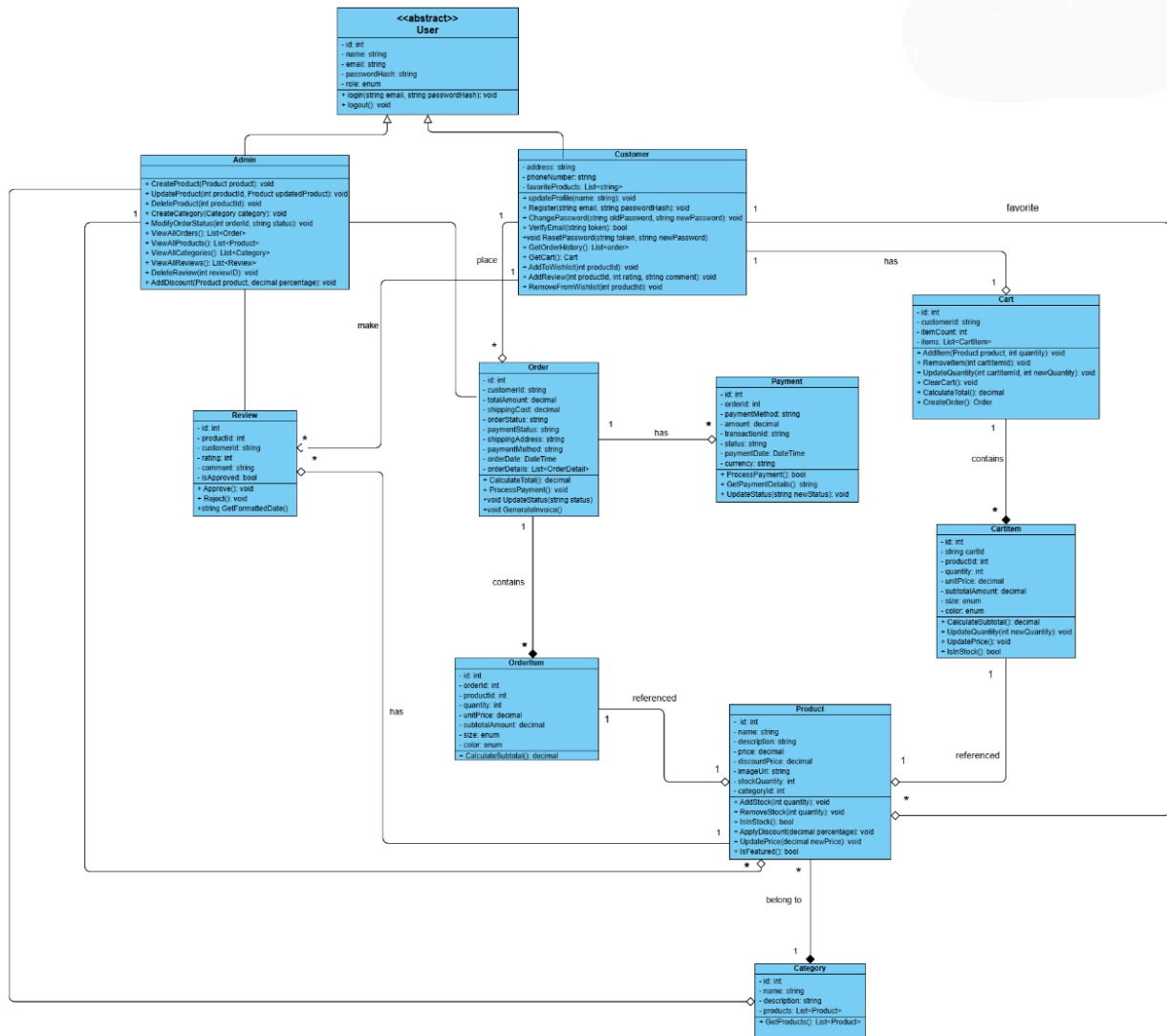
## 4.5 leave Review Activity



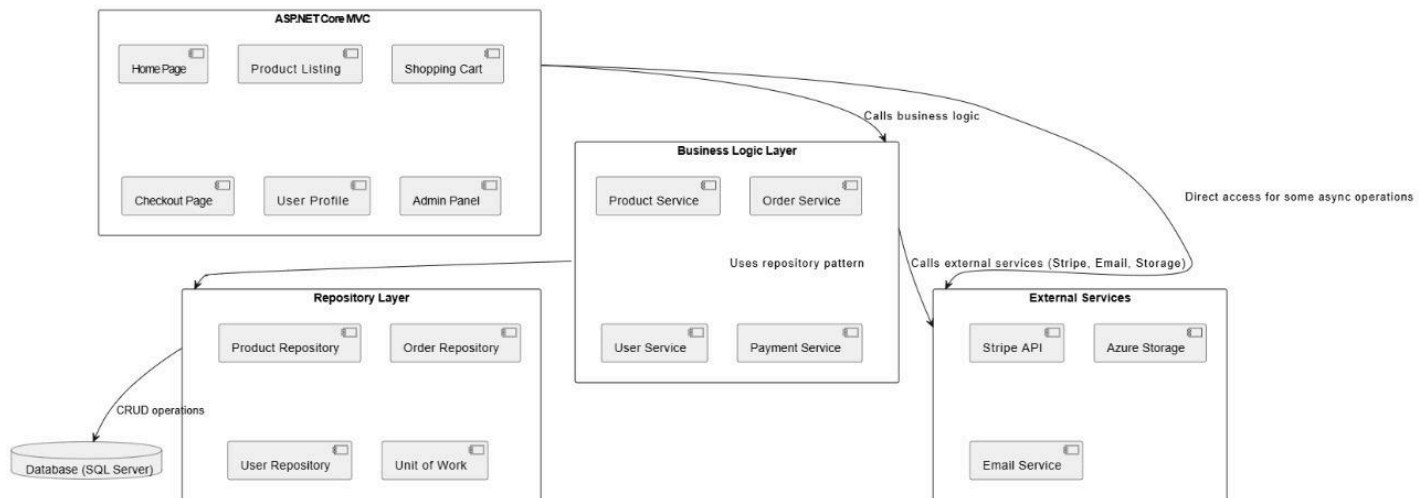
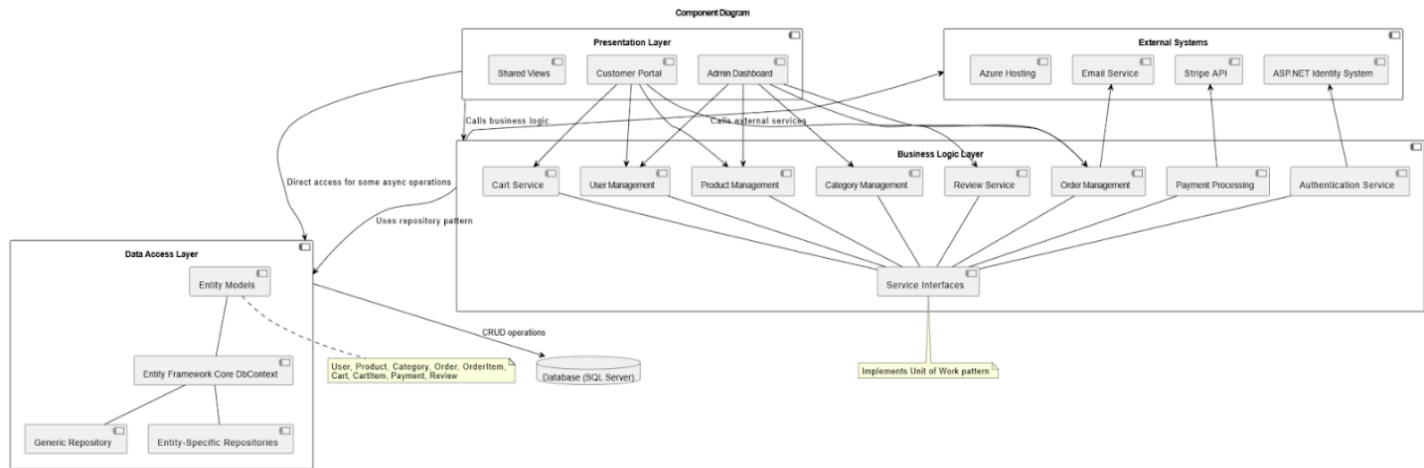
## 4.6 Make Order Activity



## 6. Class Diagram:



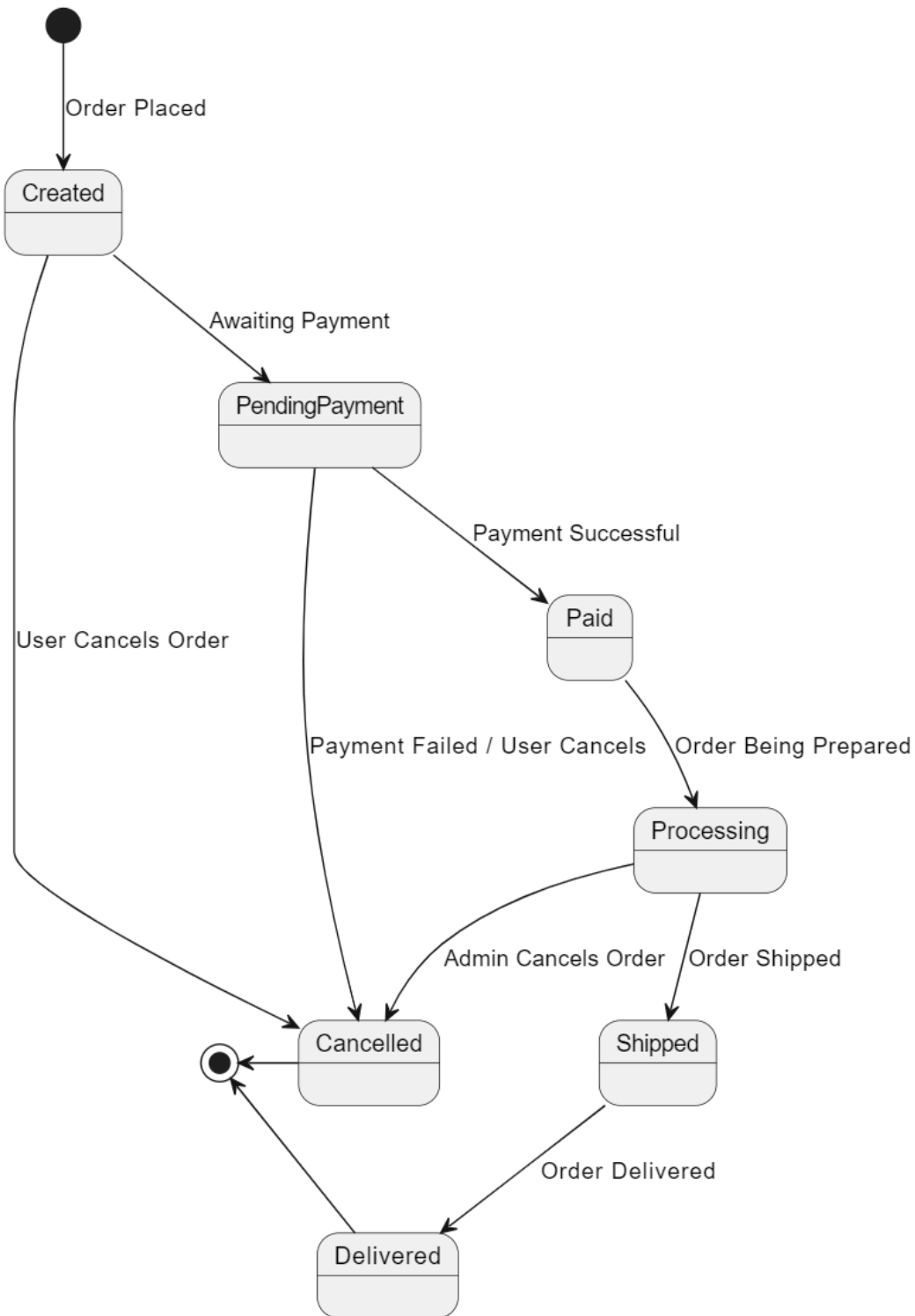
## 7. Component Diagram:



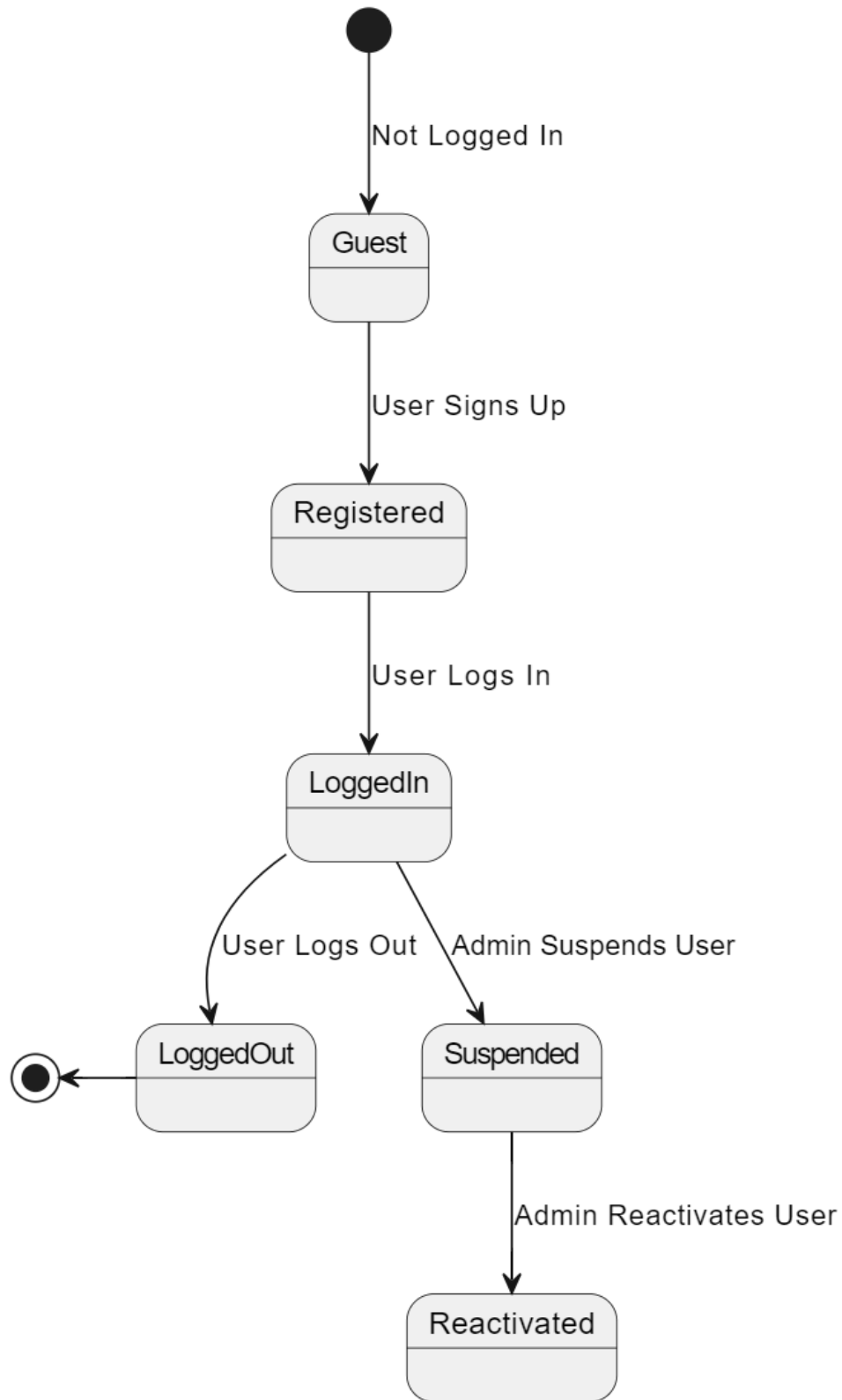


## 8. State Diagram:

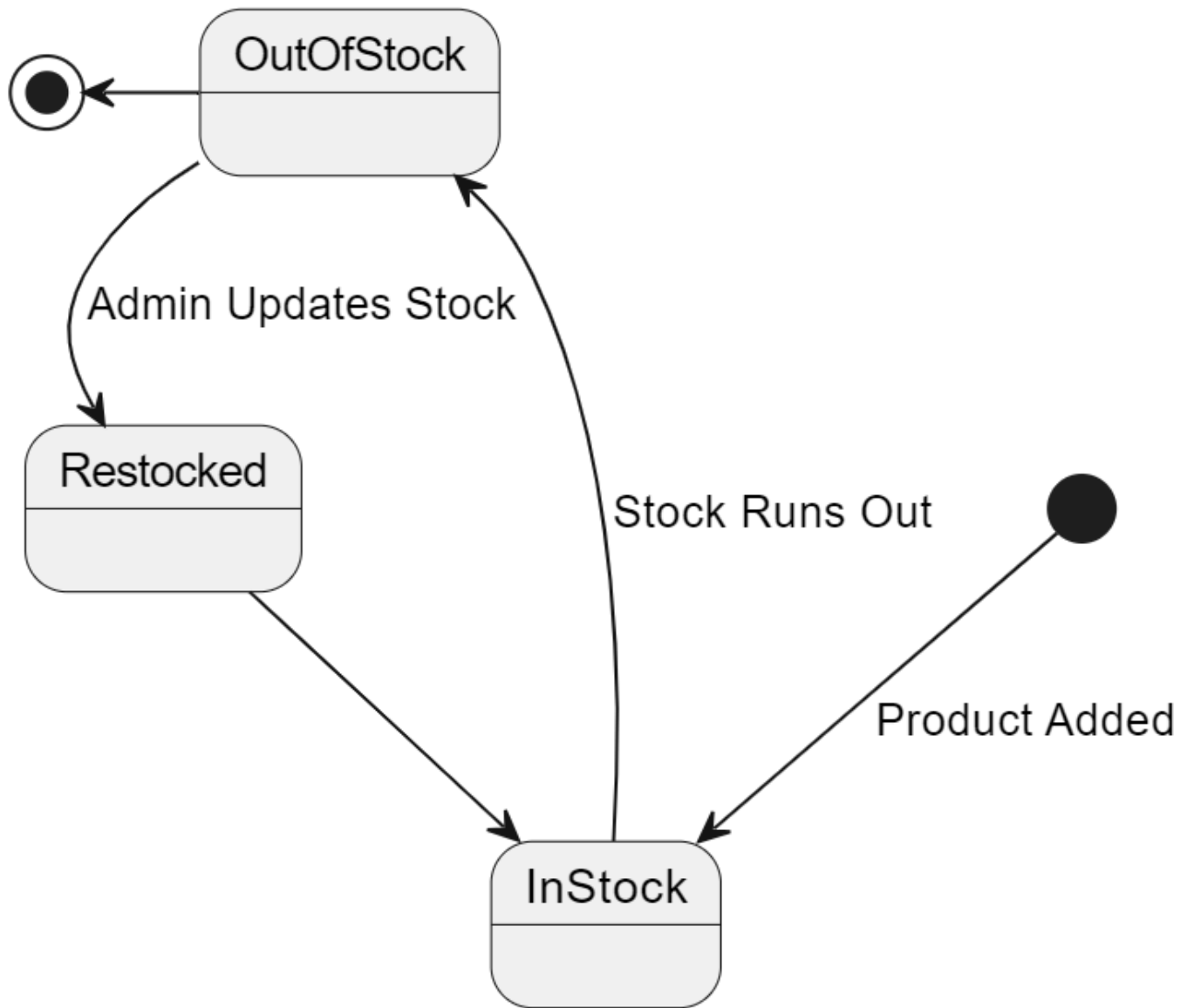
### 8.1 Order State Diagram



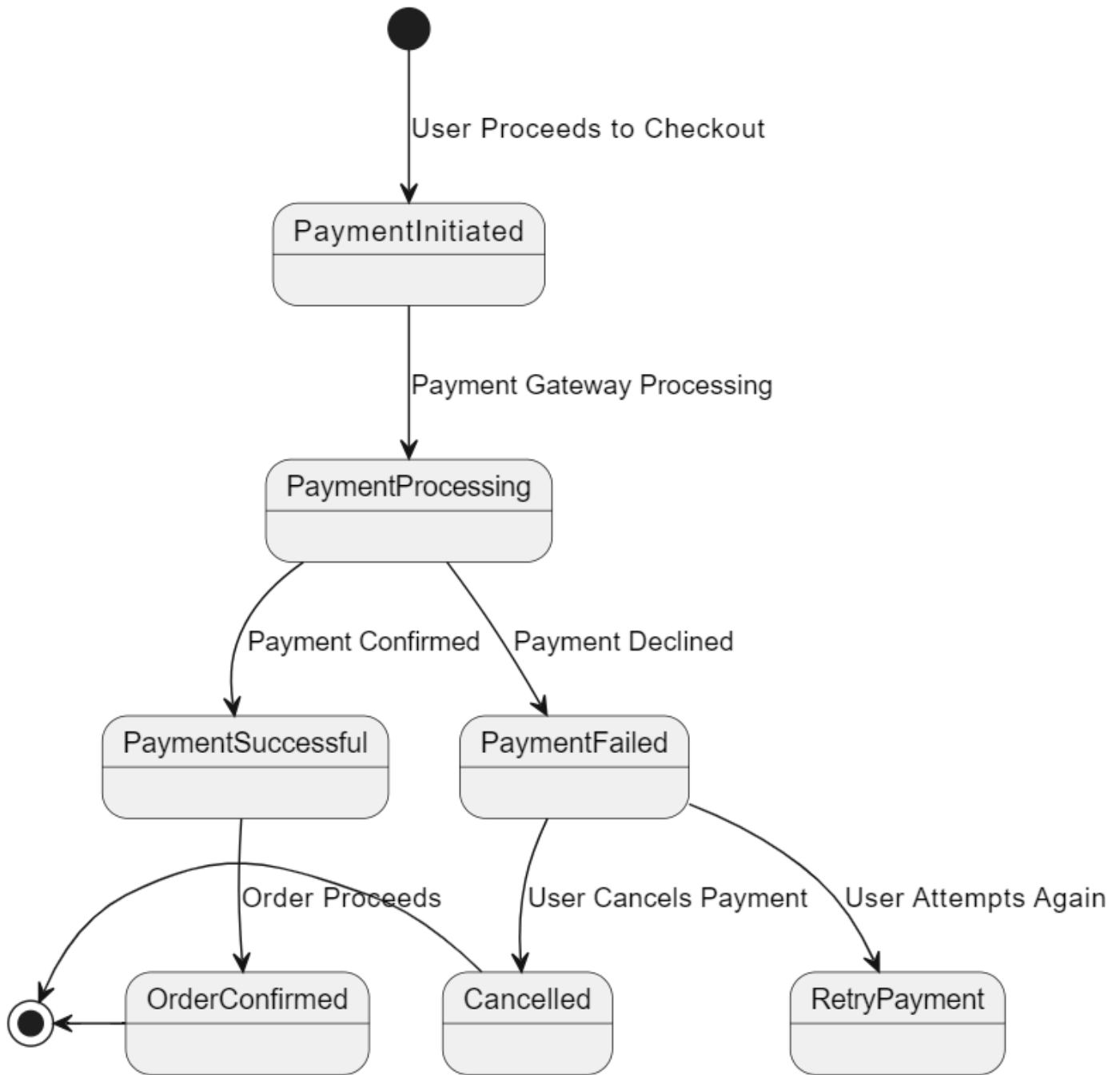
## 8.2 User Authentication State Diagram



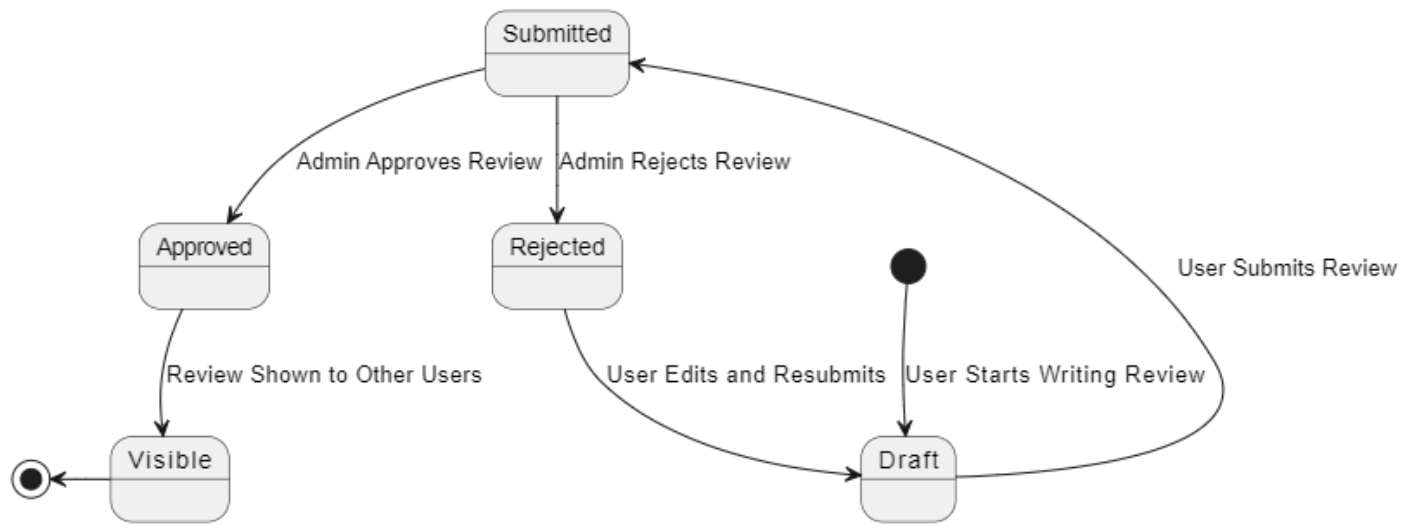
### 8.3 Product Availability State Diagram



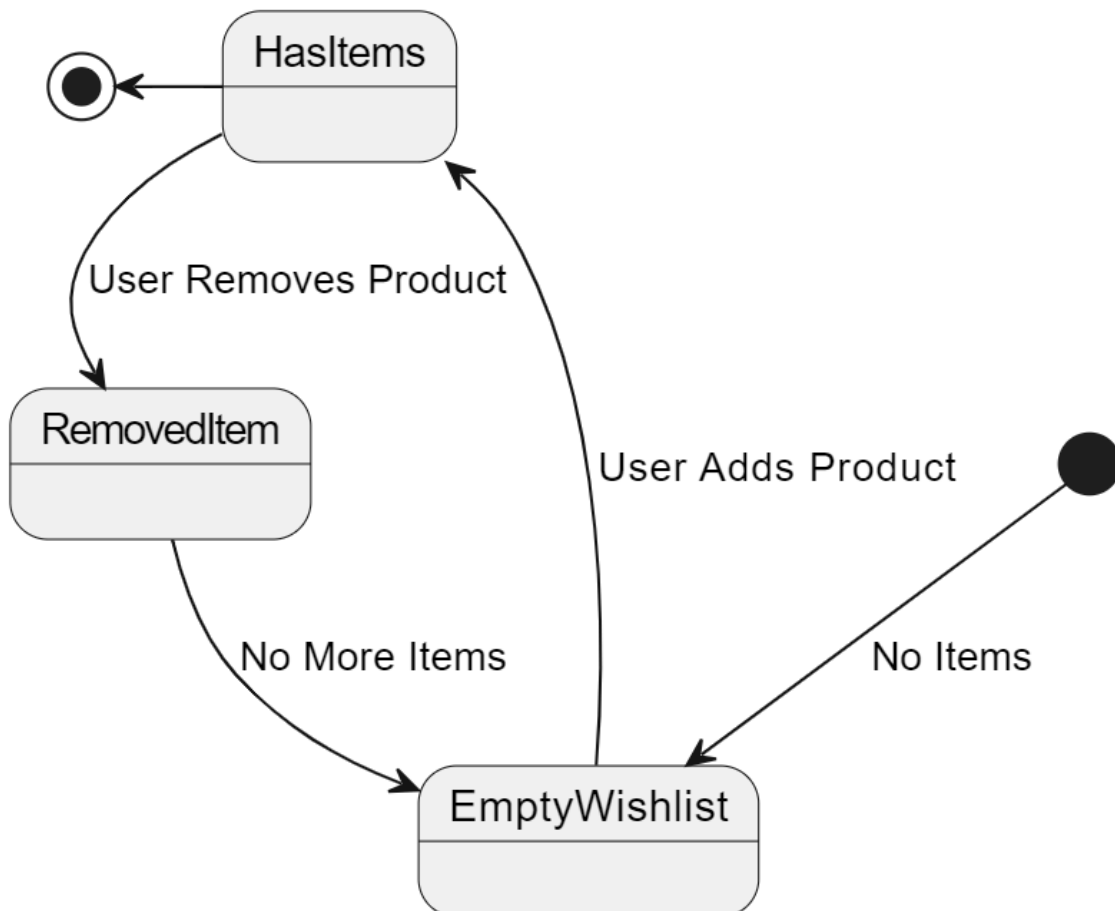
#### 8.4 Payment State Diagram



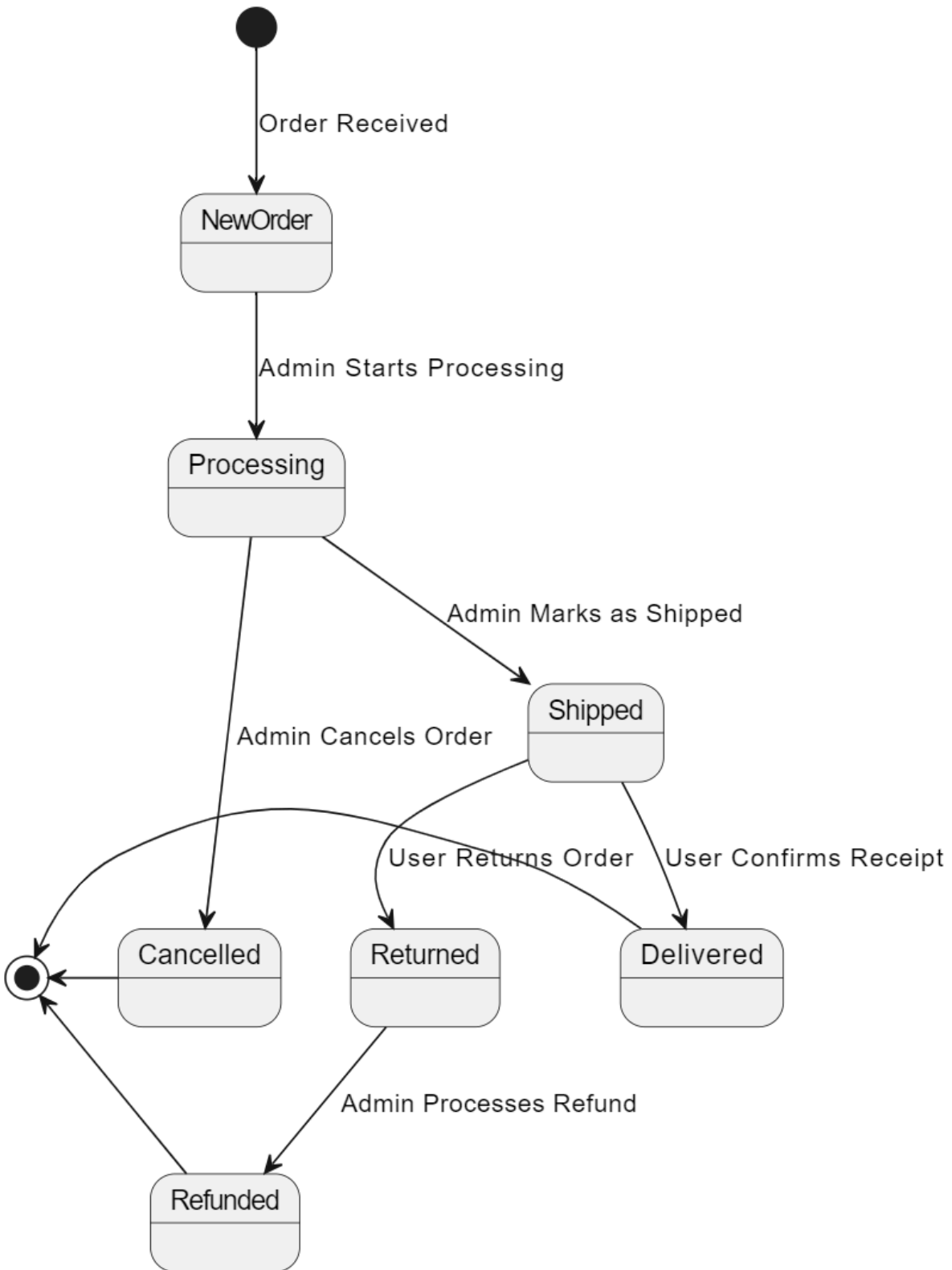
### 8.5 Review State Diagram



### 8.6 Wishlist State Diagram

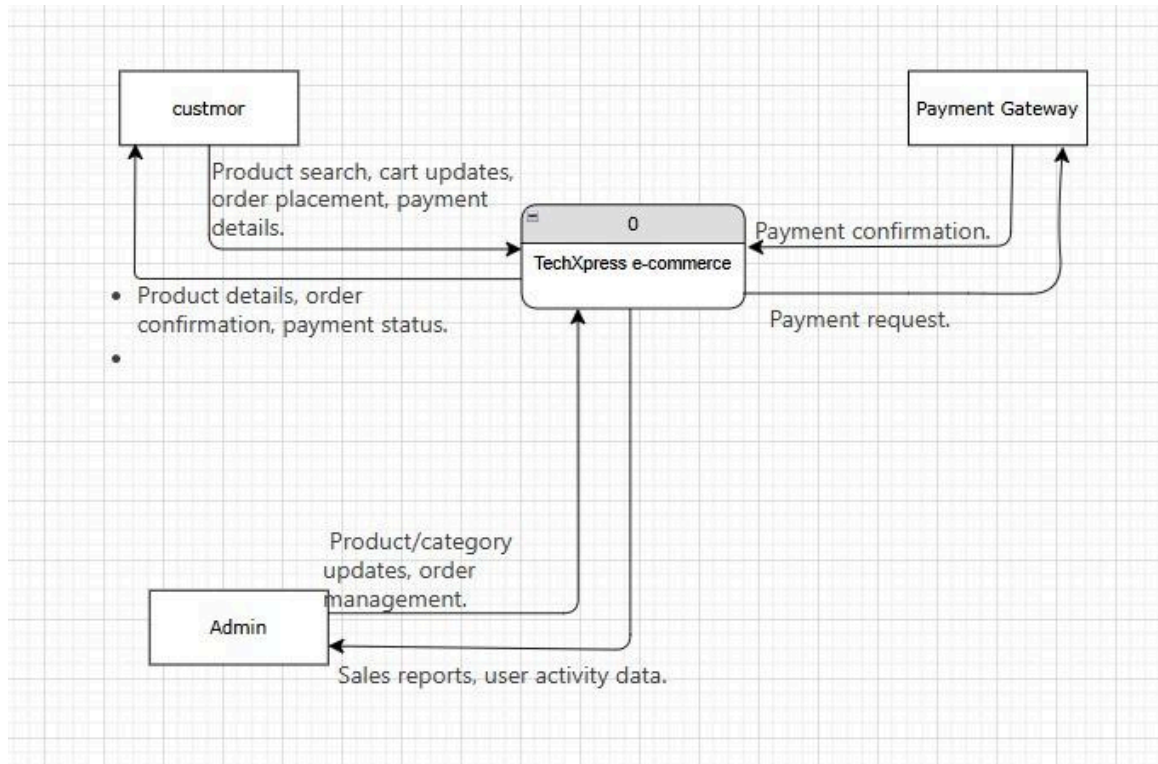


## 8.7 Admin Order Management State Diagram



## 9. Data flow diagram:

### 9.1 Context diagram



### 9.2 Data flow level 0

