# Justin Nestle CS 255 System Design Document

## UML Diagrams

### UML Use Case DiagramA diagram of a flowchart AI-generated content may be incorrect. A diagram of a person with blue circles AI-generated content may be incorrect. A diagram of a flowchart AI-generated content may be incorrect. A diagram of a person with a blue circle AI-generated content may be incorrect.

### UML Activity Diagrams *Payment System - Process Payment A diagram of a payment method AI-generated content may be incorrect.* Instructor – Manage Schedule

A diagram of a program

AI-generated content may be incorrect.

### UML Sequence Diagram

*Process Payment   
A diagram of a payment system

AI-generated content may be incorrect.*

### UML Class Diagram

*A screenshot of a computer

AI-generated content may be incorrect.*

## Technical Requirements

To implement the system effectively, specific hardware, software, tools, and infrastructure are required. The system needs a dedicated or cloud-based server with an \*\*Intel Xeon or AMD EPYC processor, 16GB RAM, and 500GB SSD storage\*\* to handle transactions and database queries. A \*\*100 Mbps internet connection\*\* ensures smooth operations, while \*\*workstations with 8GB RAM and dual-core processors\*\* support retailer and administrator access.

For software, the system should run on \*\*Windows Server or Linux (Ubuntu/CentOS)\*\* with a \*\*MySQL, PostgreSQL, or MongoDB database\*\*. The backend can be developed using \*\*Node.js, Java Spring Boot, or Python Django\*\*, while the frontend uses \*\*React, Angular, or Vue.js\*\*. Secure payments are handled via \*\*Stripe, PayPal, or Square APIs\*\*.

Development tools include \*\*Visual Studio Code, IntelliJ IDEA, or Eclipse\*\* for coding, \*\*GitHub or GitLab\*\* for version control, and \*\*Microsoft Visio, Lucidchart, or draw.io\*\* for UML modeling. \*\*Postman or Swagger\*\* assists with API development and testing.

For infrastructure, \*\*AWS, Microsoft Azure, or Google Cloud\*\* provide scalable hosting. Security measures include \*\*SSL/TLS encryption, role-based access control (RBAC), and regular backups\*\*. \*\*Load balancing and auto-scaling\*\* ensure the system handles traffic efficiently. Meeting these requirements ensures a \*\*secure, scalable, and efficient\*\* system for managing orders, payments, inventory, and customer interactions.