Object-Oriented Design and UML Diagrams for Amazon.com Book Subsystem

Course number: CS374

Antoine Gaton

November 2, 2024

Contents

[Introduction (Copy Over the Requirements) 3](#_Toc140507849)

[OOD for the UML Component Diagram 4](#_Toc140507850)

[UML Component Diagram 1 (Copied Over From the Unit 3 IP) 4](#_Toc140507851)

[List of Components and Object Classes (1 Object Class per Component) 4](#_Toc140507852)

[Is-A Relationship Class Diagram (2 Object Classes, Notes) 4](#_Toc140507853)

[Has-A Relationship Class Diagram (2 Object Classes, Notes) 4](#_Toc140507854)

[UML Sequence Diagram (At Least 2 Object Classes, Notes) 4](#_Toc140507855)

[Object Class Name, Method Name, and UML Activity Diagram (Notes) 4](#_Toc140507856)

[Object Class Name and UML State Machine Diagram (notes) 4](#_Toc140507857)

[References 5](#_Toc140507858)

# Introduction (Copy Over the Requirements)

This document presents the object-oriented design (OOD) for Amazon.com’s book search and purchasing subsystem. It extends the Unit 3 IP with additional UML diagrams, including class, sequence, activity, and state diagrams. Each component is represented by a corresponding class, with "Is-A" and "Has-A" relationships illustrated to show inheritance and composition, and detailed UML diagrams depicting system interactions and behaviors

# OOD for UML Component Diagram

The system utilizes microservices architecture, with distinct services for search, shopping cart, user management, payment processing, and order handling. Each service is modeled as a separate component, promoting modularity and scalability. Below is the UML component diagram for the system.

## UML Component Diagram 1 (Copied Over From the Unit 3 IP)

A screenshot of a computer

Description automatically generated

## List of Components and Object Classes (1 Object Class Per Component)

|  |  |
| --- | --- |
| **Component** | **Corresponding Object Class** |
| UserManagement | User |
| SearchService | Search |
| BookCatalog | Book |
| ShoppingCart | Cart |
| OrderProcessing | Order |
| PaymentService | Payment |

## Is-A Relationship Class Diagram (2 Object Classes, Notes)

A diagram of a user login and login

Description automatically generated

***Notes***: *PremiumUser* extends the *User* class, providing additional features exclusive to premium members, demonstrating an "Is-A" relationship.

## Has-A Relationship Class Diagram (2 Object Classes, Notes)

A diagram of a payment method

Description automatically generated

***Notes***: The *Order* class contains an instance of *Payment*, showing that an order "Has-A" payment, necessary for completing a transaction.

## UML Sequence Diagram (At Least 2 Object Classes, Notes)

A diagram of a book service

Description automatically generated

***Notes***: This sequence diagram shows the interaction between *User*, *SearchService*, *BookCatalog*, and *ShoppingCart* to search for and add a book to the cart.

## Object Class Name, Method Name, and UML Activity Diagram (Notes)

A diagram of a payment process

Description automatically generated

***Note***: The activity diagram represents the *processPayment* method in the Payment class, checking available funds before processing and updating the order.

A diagram of a process

Description automatically generated

***Note***: This activity diagram illustrates the *processOrder* method in the *Order* class. It checks inventory, reserves items if available, updates the status, and sends the order to fulfillment. If inventory isn’t available, it notifies the user.

## Object Class Name and UML State Machine Diagram (Notes)

A diagram of a process

Description automatically generated

***Note***: The state machine diagram captures the lifecycle of an Order object from initialization to delivery.

A diagram of a payment process

Description automatically generated

***Note***: This state machine diagram represents the lifecycle of a *Payment* object. Starting from *Pending*, it transitions to *Authorized* upon successful authorization. From *Authorized*, it can either move to *Completed* if the payment is successful or *Failed* if there’s an error.

# References

Amazon Web Services. (n.d.). *Amazon.com – Early Days Architecture*. AWS Architecture Blog. Retrieved from <https://aws.amazon.com/blogs/architecture/>

Fowler, M. (2014, March 25). *Microservices*. Martin Fowler. Retrieved from <https://martinfowler.com/articles/microservices.html>

Microsoft. (n.d.). *Microservices architecture style*. Microsoft Azure Architecture Center. Retrieved from <https://learn.microsoft.com/en-us/azure/architecture/guide/architecture-styles/microservices>

Open Web Application Security Project. (n.d.). *Web Security Architecture*. OWASP. Retrieved from <https://owasp.org/www-project-web-security-testing-guide/>

Nielsen Norman Group. (2021, January 10). *E-Commerce User Experience*. Retrieved from <https://www.nngroup.com/articles/e-commerce-usability/>