An Online Bookstore System
Software Architecture Document

Version <1.0>

Revision History

Date	Version	Description	Author
23/11/23	1.0	Introduction + Architectural Goals and Constraints	Nguyen Le Tan Thanh
25/11/23	1.1	Edit Use-cases Model	Doan Anh Khoa Le Phuong Chi
27/11/23	1.2	MVC Model	Nguyen Chi Luong
30/11/23	1.3	Logical View: Component	Doan Anh Khoa Le Phuong Chi
01/12/23	1.4	edit Logical View: Component	Nguyen Le Tan Thanh

Table of Contents

Software Architecture Document

1. Introduction

The Software Architecture Document (SAD) is to provide a detailed document about the architecture of a software system. This document helps development team members, project managers, and users understand the structure, design, and operation of the system. To provide preventive and handling measures if risks occur and to support the maintenance and expansion of the system in the future.

1.1 Overview

This document is presented according to the following structure:

1. **Introduction:** This section provides an overview of the entire Software Architecture Document, including its purpose, scope, definitions, acronyms, abbreviations, and references.
2. **Architectural Goals and Constraints:** This section describes the *software requirements* and *goals* that have some significant impact on the architecture.
3. **Use-Case Model:** The Use-Case Model section presents a comprehensive overview of the software's functional requirements from the perspective of users. It identifies and describes the primary use cases that the software is intended to support.
4. **Logical View:** This section provides detailed descriptions of individual system components, their functionalities, interfaces, and interactions with other components.

1.2 Purpose

The purpose of the Software Architecture Document is to:

1. **Communicate the Architecture:**
   - Provide a common source of information about the system architecture for all members of the development team.
   - Helps everyone understand the system's architecture and how components interact with each other.

2. **Manage progress and divide tasks:**
   - Provide work progress information, what tasks have been completed or have not been completed.
   - Break down the architecture into individual components for easier development and testing.
   - Supports the distribution of work among team members.

3. **Support Design decisions:**
   - Identify key design decisions and explain why they were chosen.
   - Supports decisions about the overall structure of the system and other key decisions such as the choice of technologies, frameworks, and design models.

4. **Development and Maintenance Manual:**
   - Provide guidance to development teams on system deployment, integration, and

      maintenance.
- Supports the maintenance and expansion of the system in the future.

1.3 Scope

The scope of this Software Architecture Document includes the entire architecture of both functional and non-functional aspects of the software, key design decisions, patterns, and technologies used.

1.4 References

1. [The subject's youtube channel](#)

# 2. Architectural Goals and Constraints

2.1 Architectural Goals

1. **Accuracy:** The program must run exactly as expected without serious errors.
2. **Search Engine Optimization:** Optimize this website to enhance searchability, ensuring that your products can be easily found on search engines.
3. **Effective sales:** The main goal is to increase sales and profits by providing a convenient and engaging shopping experience for customers. Use this website to promote book brands and provide detailed information about new titles, authors, and trends.
4. **Good User Experience:** Create a user-friendly, easy-to-use interface and optimize the user experience to encourage customer engagement and retention.
5. **Customer care:** Provide communication channels such as online chat, email, or hotline to support and answer customer questions.

2.2 Constraints

1. **Information security:** Ensure the security of personal information and customer accounts through security measures such as SSL, data encryption, and access management.
2. **Inventory management:** Maintain accurate inventory information to ensure the availability of books as needed.
3. **Compliance with laws:** Ensure compliance with legal regulations on copyright, privacy, and other regulations related to online business.
4. **Cross-Platform Compatible:** Design your website for a smooth shopping experience on multiple devices, from computers to mobile phones.
5. **Website Performance:** Make sure the website operates smoothly and has fast loading times so as not to degrade the user experience.
6. **Scalability:** The website should be able to handle a large volume of users with the ability to scale up or down as needed.
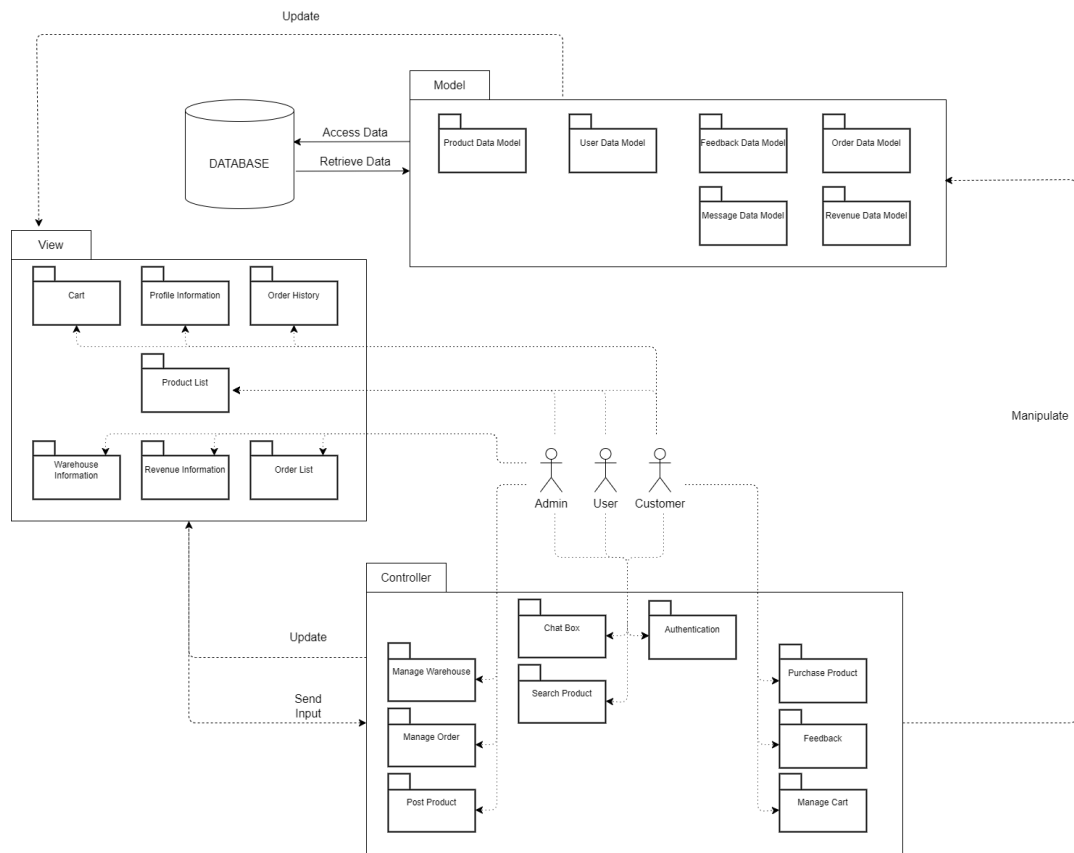
An Online Bookstore System	Version: <1.4>
Software Architecture Document	Date: <01/12/23>
<document identifier>	

3.   Use-Case Model
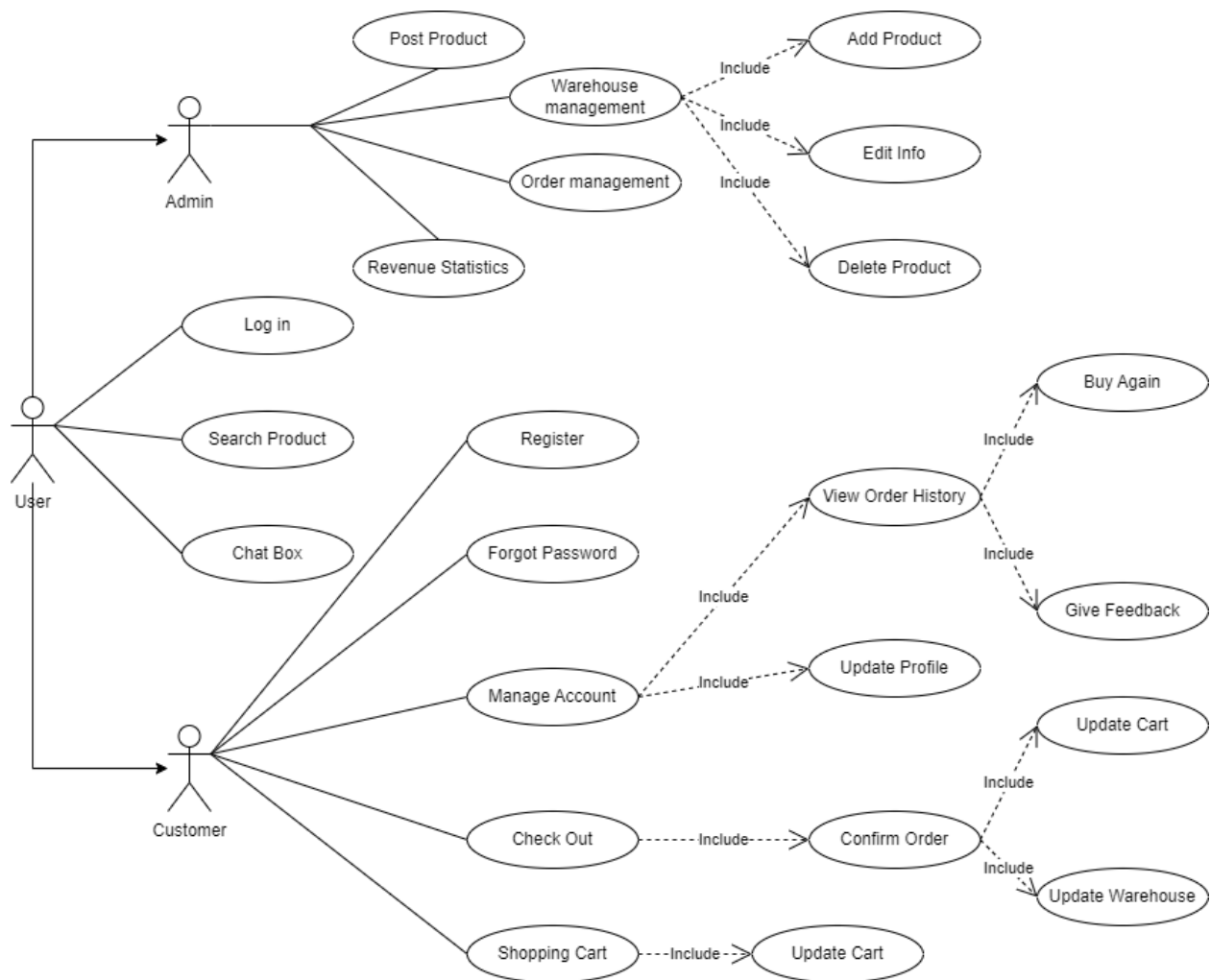
| An Online Bookstore System	Version:	<1.4>
Software Architecture Document	Date: <01/12/23>	
<document identifier>		

4.   Logical View

Every relationship of these components uses HTTP as a means of communication.

- Component: Cart
    + This component represents a list of selected products.
    + The component provides objects for the Manage Cart Component to handle.
- Component: Chatbox
    + Account for the communication of User and Administrator.
- Component: Feedback
    + Account for the user feedback after using the product.
- Component: Manage Cart
    + This component will apply the user's actions on the Cart Component.
- Component: Manage Order
    + This component monitors orders.
- Component: Manage Warehouse
    + This component monitors the inventory of the bookstore.
    + When a book is uploaded by the Post Product Component. You can find out about it with this component.
- Component: Post Product
    + This component will add a product to the warehouse.
    + The Manage Warehouse Component can monitor the product that has been added to the warehouse.

An Online Bookstore System	Version: <1.4>
Software Architecture Document	Date: <01/12/23>
<document identifier>	

+ This component also provides books for the Search Product Component to find.
- Component: Purchase Product
  + This component will remove a number of books from the inventory of the bookstore.
  + The Manage Warehouse Component can monitor the product that has been removed from the warehouse.
- Component: Search Product
 + This component finds books that have been added to the bookstore by the Post Product Component.

4.1 Component: Authentication
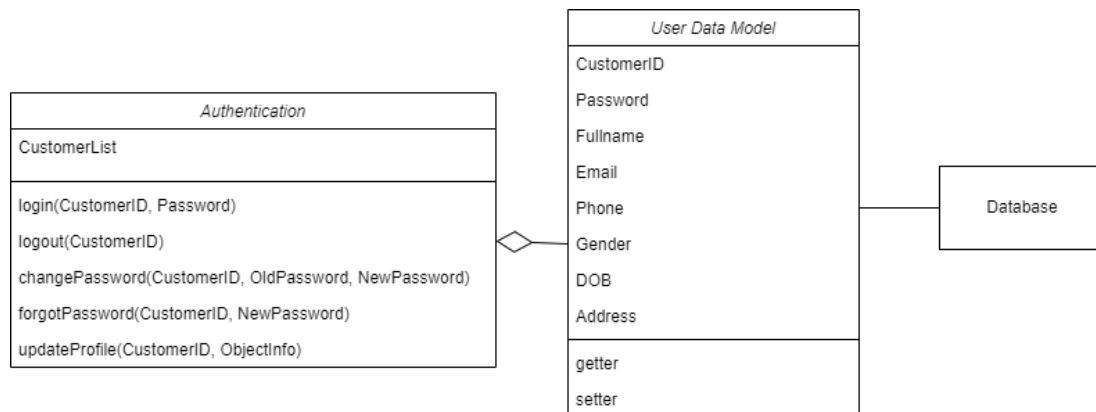
● Authentication class:
  ○ Attributes:
    ■ CustomerList: A list of customers extracted from the database, including personal and account information, used when a user logs in and uses the website.
  ○ Methods:
    ■ login: The login method will allow the user to start using the website. When the user logs in, this method will search for the CustomerID in the CustomerList attribute. If this CustomerID exists, then compare the password. If the password is correct, the user logs in successfully, if not, the login fails.
    ■ logout: This method will end the current account's session, logging out all account information from the website.
    ■ changePassword: This method helps users change their current password. Users need to enter the correct current password (old password) and enter a new password to update.
    ■ forgotPassword: This method is used to change the password without entering the old password (forgot password). However, you must verify that this is indeed your account.
    ■ updateProfile: Method to help update account information.
● User Data Model:
  ○ CustomerID: is a unique identifier for each customer.
  ○ Password: This is the password field, provided by the user when creating the account, used to secure login.
  ○ Fullname, email, NumberPhone, Gender, DayOfBirth, and Address: are the pieces of information provided by the user when creating the account.

4.2 Component: Chatbox
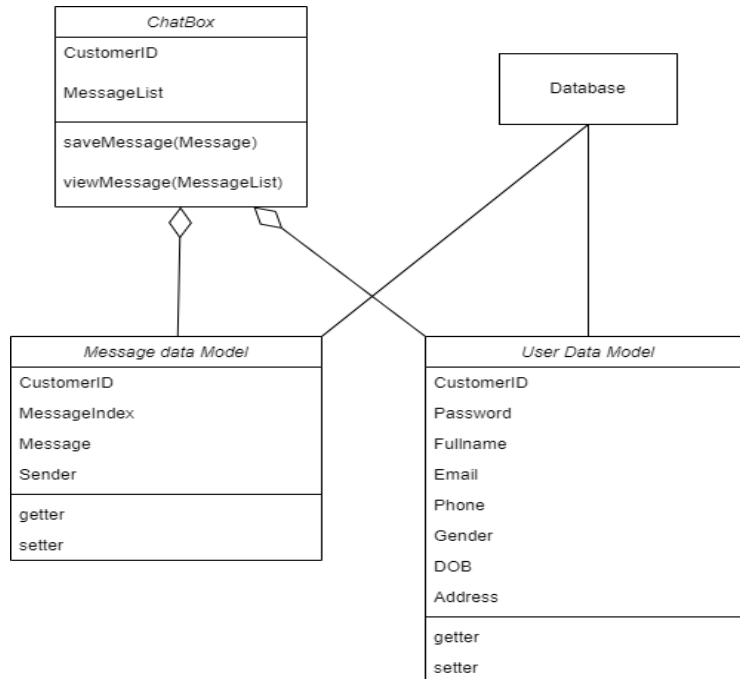
- ChatBox class:
  - Attribute:
    - CustomerID: is a unique identifier for each customer.
    - MessageList: A list of conversations the customer has participated in and the content of those.
  - Methods:
    - saveMessage(): when a user sends a message to the chat box, the message content is immediately saved so that old messages can be retrieved.
    - viewMessage() helps users review old messages.

- Message data Model:
  - CustomerID: CustomerID, so the admin knows which customer is messaging.
  - MessageIndex: Index of a message package, which helps us arrange the order of message packages.
  - Message: is the content of a message package.
  - Sender: flag to identify who sends the message package.
- User data Model: similar to User data Model in components Authentication

4.3 Component: Feedback
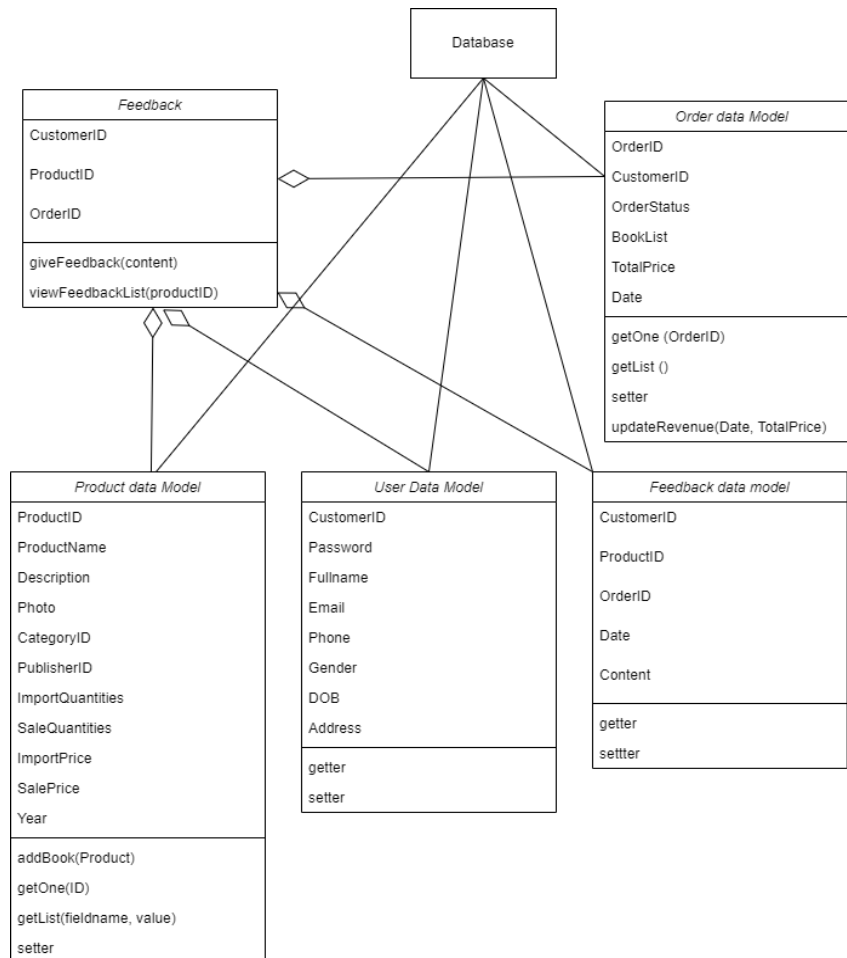
- Feedback class:
  - Attributes:
    - CustomerID: to identify who wrote this review.
    - ProductID: to identify the product being evaluated.
    - OrderID: to determine the purchase order. Once the user has purchased the product, they can rate it.
  - Methods:
    - giveFeedback(): to leave feedback, including score and review content, for a product.
    - viewFeedbackList(productID): to view all feedback of buyers with a designated book.
- Order data Model:
  - Attributes:
    - OrderID: a unique identifier for each order.
    - CustomerID: ID of the customer who made the purchase.
    - OrderStatus: the status of a designated order.
    - BookList: list of books sold in this order.

- ■ TotalPrice: total amount for the entire order, this field is used to calculate the store's revenue.
- ■ Date: the time the order is purchased.
  - ○ Methods:
    - ■ getOne: to get an order based on OrderID.
    - ■ getList: to get a list of orders.
    - ■ updateRevenue: to update revenue statistics on a designated date by adding the total price of an order.
- ● Product data model class:
  - ○ Attributes: The book information includes
    - ■ ProductID: a unique identifier for each product.
    - ■ ProductName: the name or title of the product.
    - ■ Description: the brief description related to the product
    - ■ Photo: an image file representing the product.
    - ■ CategoryID: a reference to the category to which the product belongs.
    - ■ PublisherID: a reference to the publisher or manufacturer of the product.
    - ■ ImportQuantities: the number of books imported into inventory.
    - ■ SaleQuantities: the number of books sold into inventory.
    - ■ ImportPrice: the cost per book when the book is imported.
    - ■ SalePrice: the price at which the product is sold to customers.
    - ■ Year: the year in which the product is released.
  - ○ Methods:
    - ■ The addBook method is to input book information into the warehouse database.
    - ■ The getOne method is to get book information based on ProductID from the database.
    - ■ The getList method is to get a book list based on the field name which is one of the above attributes with the specific value of the fieldname from the database.
- ● User data Model: similar to User data Model in components Authentication
- ● Feedback data Model:
  - ○ CustomerID: Identify who wrote this review.
  - ○ ProductID: Identify the product being evaluated.
  - ○ OrderID: Determine the purchase order.
  - ○ Date: time of giving feedback.
 - ○ Content: Feedback content includes scores and review content.

4.4   Component: Manage Cart
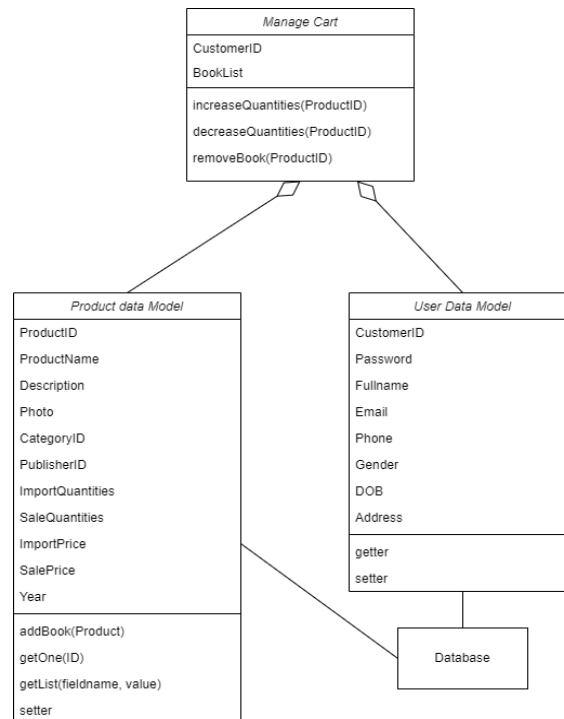
- Manage Cart class:
  - Attributes:
    - CustomerID: Determine which customer's shopping cart belongs to.
    - BookList: List of books and the number of each book in the shopping cart.
  - Methods:
    - increaseQuantities(): Increase the number of books to buy in the shopping cart.
    - decreaseQuantities(): Decrease the number of books to buy in the shopping cart
    - removeBook(): remove a product from the cart.

- Product data Model: similar to Product data Model in components Feedback.
- User data Model: similar to User data Model in components Authentication.

4.5   Component: Manage Order
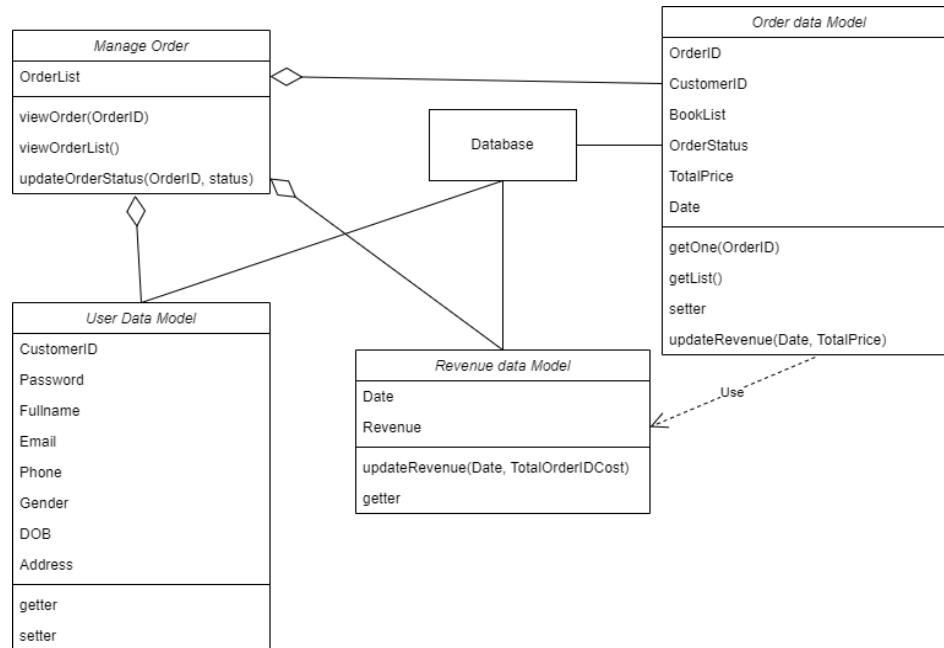
- Manage Order class:
  - Attributes:
    - OrderList: List of OrderIDs that have been purchased.
  - Methods: This method is exclusive to the admin.
    - viewOrderList(): to allow the administrator to view information about orders and their status.
    - updateOrderStatus(): to allow the administrator to update order information or order status.
- Order data Model: similar to Order data Model in components Feedback.
- Revenue data Model:
  - Attributes:
    - Date is the date on which the administrator collects revenue statistics.
    - Revenue is the financial value associated with a particular date, indicating the amount of income earned.
  - Methods:
    - updateRevenue(): to proceed to update revenue when a designated order is confirmed, which means the order is completely delivered. This method will add the old total revenue of a day with the total price of the confirmed order.

4.6   Component: Manage Warehouse
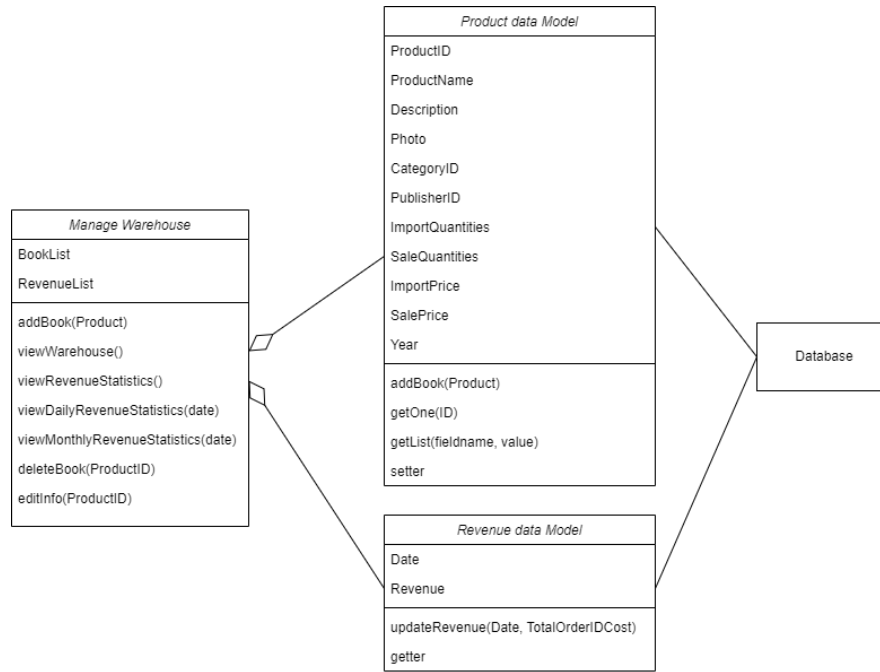
- Manage Warehouse class:
  - Attributes:
    - BookList: List ProductID of book in stock includes information related to each product.
    - RevenueList: The attribute stores sales revenue history.
  - Methods: This method is exclusive to the admin.
    - addBook(): method of adding a product to a warehouse.
    - viewWarehouse(): A method to help the admin look up and check inventory. The products present in the BookList will be displayed on the screen.
    - viewRevenueStatistic(): Based on purchase history (RevenueList), this method calculates store revenue and allows admins to view them in the form of visual charts and figures.
    - viewDailyRevenueStatistic(): to get a list of daily revenue statistics for a designated month to visualize data by drawing a chart and viewing it.
    - viewMonthlyRevenueStatistic(): to get a list of monthly revenue statistics for a designated year to visualize data by drawing a chart and viewing it.
    - deleteBook(): remove a product from inventory (BookList). ProductID is required to know which products to discard.
    - editInfo(): Allows admins to edit order information such as (selling price, import price, product name, category, etc.). ProductID is required to know which products to edit.

- Product data Model: similar to Product data Model in components Feedback.
- Revenue data Model: similar to Revenue data Model in components Manage Order.

An Online Bookstore System	Version: <1.4>
Software Architecture Document	Date: <01/12/23>
<document identifier>	

### 4.7 Component: Post Product
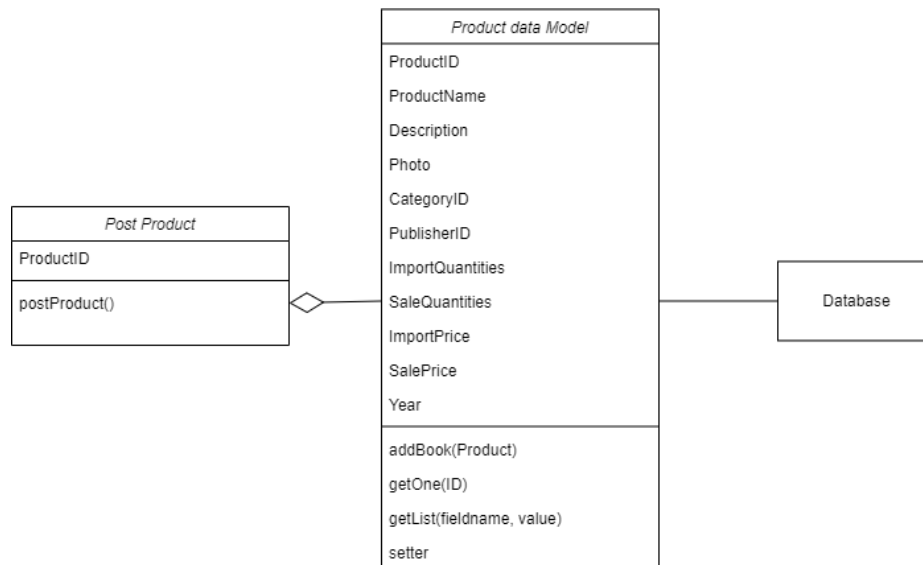
- Post Product class:
  - Attributes:
    - ProductID: a unique product identifier which is used to determine which products will be posted.
  - Methods:
    - postProduct(): This method is used to post products to the warehouse, the admin needs to provide all important information about the product before posting. Once a product is posted to the warehouse, the addBook( ) method in the Manage Warehouse will be automatically executed.

- Product data Model: similar to Product data Model in components Feedback.

An Online Bookstore System	Version: <1.4>
Software Architecture Document	Date: <01/12/23>
<document identifier>	

4.8 Component: Purchase Product
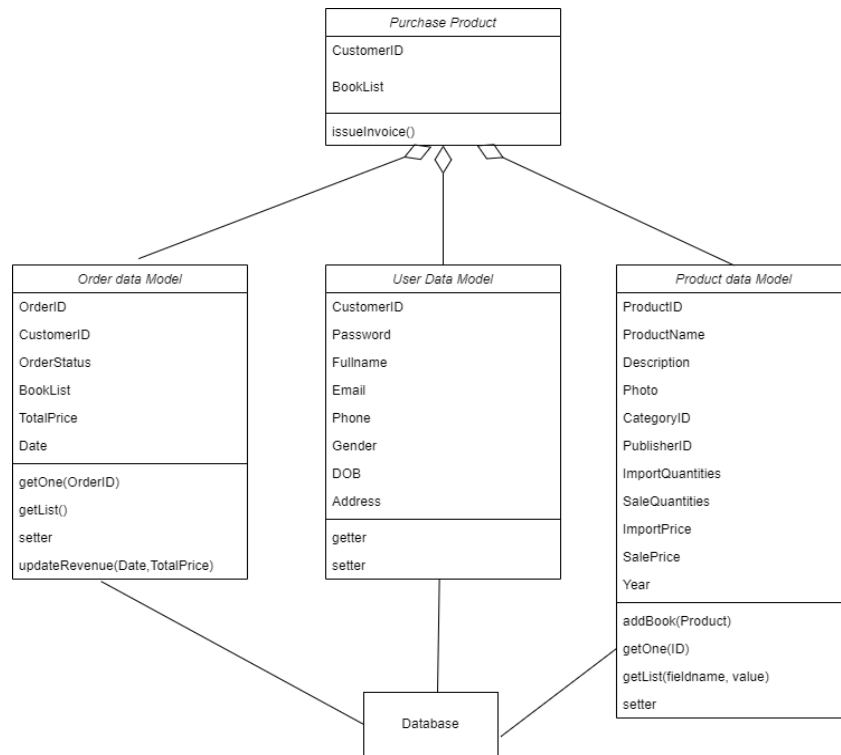
- Purchase product class:
  - Attributes:
    - CustomerID: A unique identifier assigned to each customer.
    - BookList: the list of books that the customer will purchase.
  - The issueInvoice method: this method generates an invoice that has OrderID, customer's information, list of their desired books, total price, and purchase date.
- Product data Model: similar to Product data Model in components Feedback.
- Order data Model: similar to Order data Model in components Feedback.
- User data Model: similar to User data Model in components Authentication.

4.9 Component: Search Product

- Search Product class:
  - Attributes:
    - Keysearch: the characters that a customer enters to look for some desired books.
    - BookList: the list of books which is the result list searched by the customer.
  - Methods:
    - The search method with the Keysearch argument is to return the relevant book lists.
    - The viewBookList method is to show a search list on an interface web.
- Product data Model: similar to Product data Model in components Feedback.