Design Document

for

Bookstore Web-App

Version 1.1, Approved by Tran Dinh Phu

Prepared by Nguyen Tien Dat, Bo Quoc Trung, Tran Dinh Phu

Group 13, 2324II INT2208E 23, VNU-UET

April 20, 2024

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Draft #1	2024-04-25	Initial Version	0.1
Version 1.0	2024-05-05	First (relatively) complete version of the Design Document	1.0

Version 1.1	2024-05-05	Adjusted/fixed some Sequence Diagrams and Class Diagram; adjusted margins and formatting	1.1

1. INTRODUCTION

1.1. Purpose

This software design document describes the architecture and system design of **Bookstore Web-App.**

1.2. Scope

Bookstore Web-App is an online platform that facilitates the purchase of books over the Internet. Its primary purpose is to provide a convenient, easy-to-understand, user-friendly marketplace where users can easily browse, search and purchase books from a vast catalog spanning various genres and categories

The key goals of Bookstore Web-App are:

- Provide an efficient shopping experience
- Enable access to a wide variety of books from various publishers and authors

The essential key features of the E-Commerce include

- CRUD functionality for admin (Create, Read, Update, Delete), view and confirm book order
- Users can views book details and add, delete books from their shopping cart, checkout and the web-app will generate a confirmation order for them
- Guest users can view the details of books (Authors, Summary, Price, etc…)

1.3. Overview

This document outlines the design specifications for Bookstore Web-App. It details the functionalities, system overview, system architecture, data design, component design and user interface (UI) considerations for the application.

2. SYSTEM OVERVIEW

Functionality:

The core functionalities of the bookstore web app include:

- **Extensive Book Catalog**: Users can browse a wide variety of books from various genres and categories.
- **Search and Discovery**: A robust search engine and filter options allow users to find books by title, author, genre, keyword, or other criteria. Users can also discover books through curated recommendations, bestseller lists.

- **User Accounts**: Account registration enables users to manage their wishlists, track purchase history, and save preferred settings.
- **Shopping Cart**: Users can add and remove books from their cart, update quantities, and view order totals before checkout.
- **Secure Checkout**: The checkout process is streamlined and secure, offering multiple payment options (e.g., credit card, PayPal) with robust security measures in place.

Design:

The web app is designed with a user-friendly interface, prioritizing clear navigation, intuitive features, and visually appealing aesthetics. The Model-View-Template (MVT) architecture with Django framework provides a structured and scalable foundation for development. The design emphasizes accessibility, ensuring a smooth experience for users with diverse technical backgrounds and abilities.

3. SYSTEM ARCHITECTURE

3.1. Architectural Design

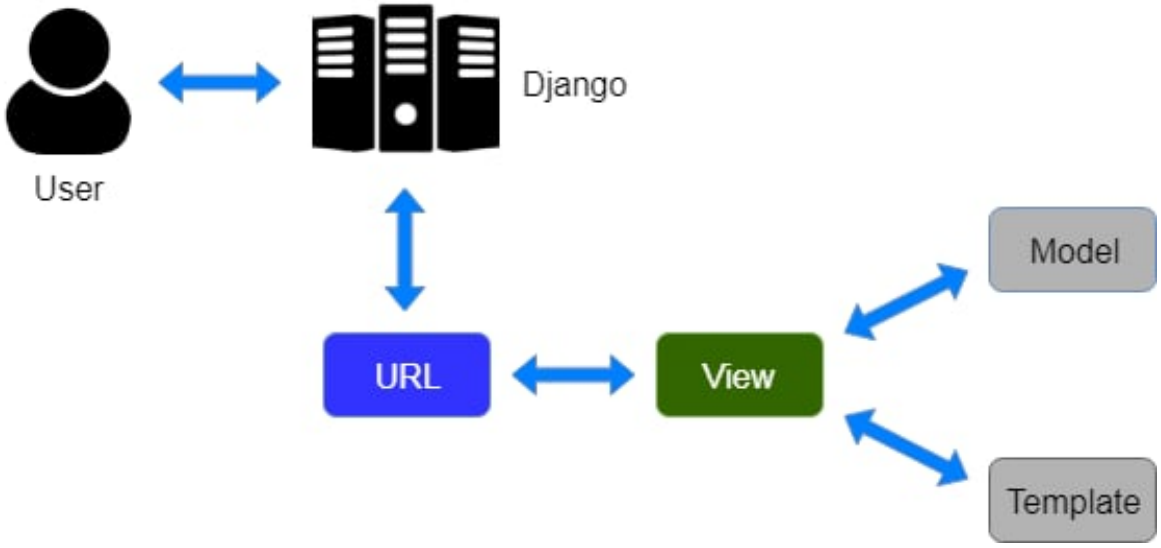
MVT structure: When a user requests a resource to Django, **Django works as a controller** and checks the available resource in the URL.

URLS: define URL patterns to map incoming requests to views.

Models : Interface to app's data. Responsible for maintaining data and the Data Structure behind the entire application.

Views: Execute the logic and interact with Models to carry data and render a template.

Template: User Interface - What the user sees when rendering a website. Consist of HTML/CSS/Javascript files.



3.2. Decomposition Description

Architectural Approach: Object-Oriented (OO)

Subsystems:

- **User Management:** Handles user registration, login, authentication, and user profile management.
- **Inventory Management:** Manages book information, categories, and book updates.
- **Shopping Cart Management:** Facilitates adding, removing, updating items in the cart, and calculating cart totals.
- **Order Processing:** Processes user orders including order confirmation, payment processing, and order fulfillment.
- **Search and Browse:** Enables users to search for books by title, author, genre, or other criteria, and browse book categories.
- **Payment Gateway:** Integrates with a secure payment gateway for processing online transactions.

Object Models:

- **UserPayment**
- **Cart**
- **CartItems**
- **Order**
- **OrderDetails**
- **User**
- **Books**
- **UserAddress**

Template (HTML):

- **checkout**
- **order-summary**
- **bookdetail**
- **home**
- **user**
- **booklist**
- **search**
- **base**

3.3. Design Rationale

Advantages:

- **Separation of Concerns**: MVT promotes a clear separation of concerns between models (data), views (business logic and presentation), and templates (user interface). This modular design improves code maintainability, reusability, and testability.

- **Rapid Development**: Django's MVT structure offers pre-built functionalities for common web development tasks like user authentication, database access, and URL routing.

- **Scalability**: The MVT architecture can be easily scaled to accommodate a growing user base and increased complexity. Additional models, views, and templates can be readily integrated into the existing structure.

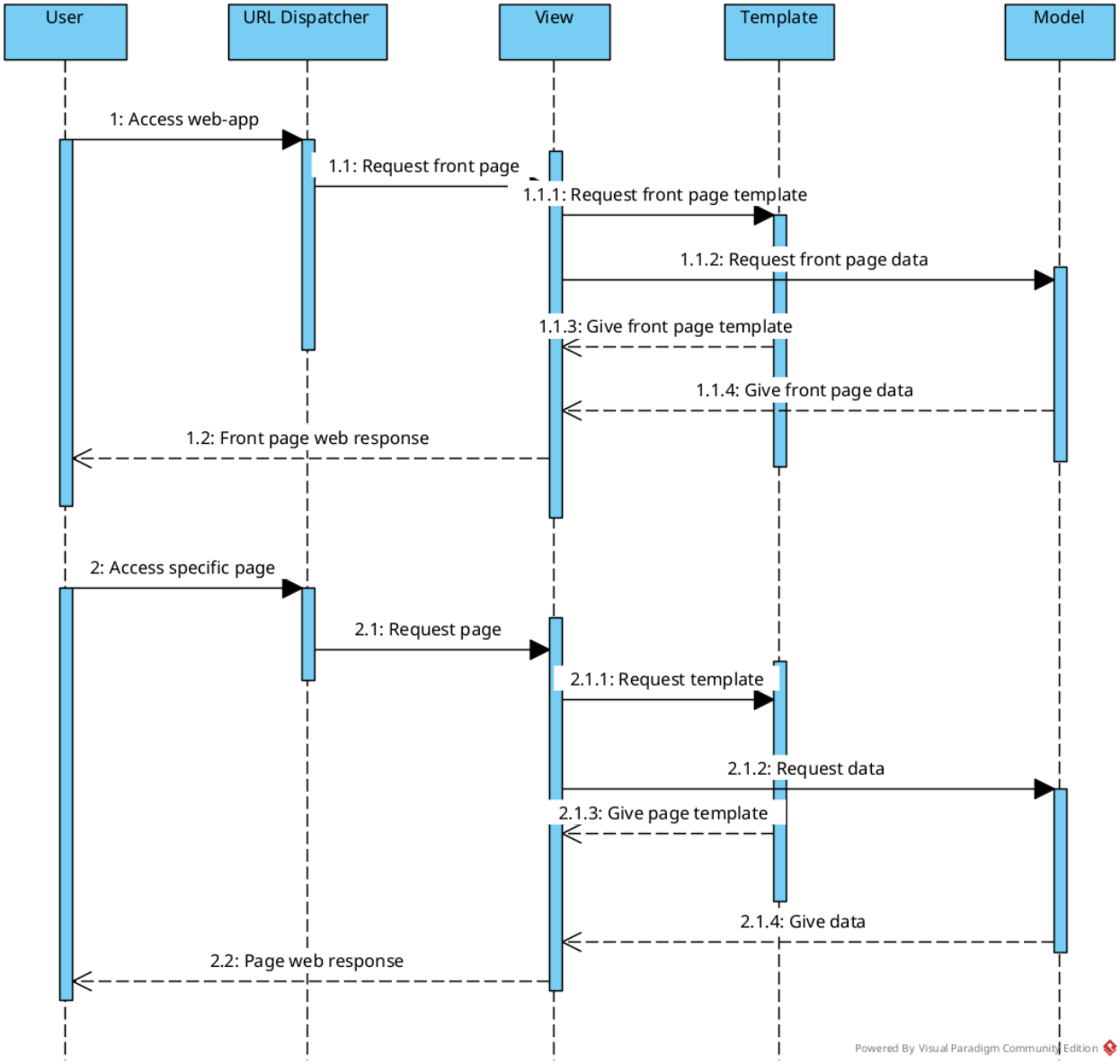
- **Security**: Django enforces security best practices, such as user authentication and authorization, data validation, and protection against common web vulnerabilities.

Trade-offs:

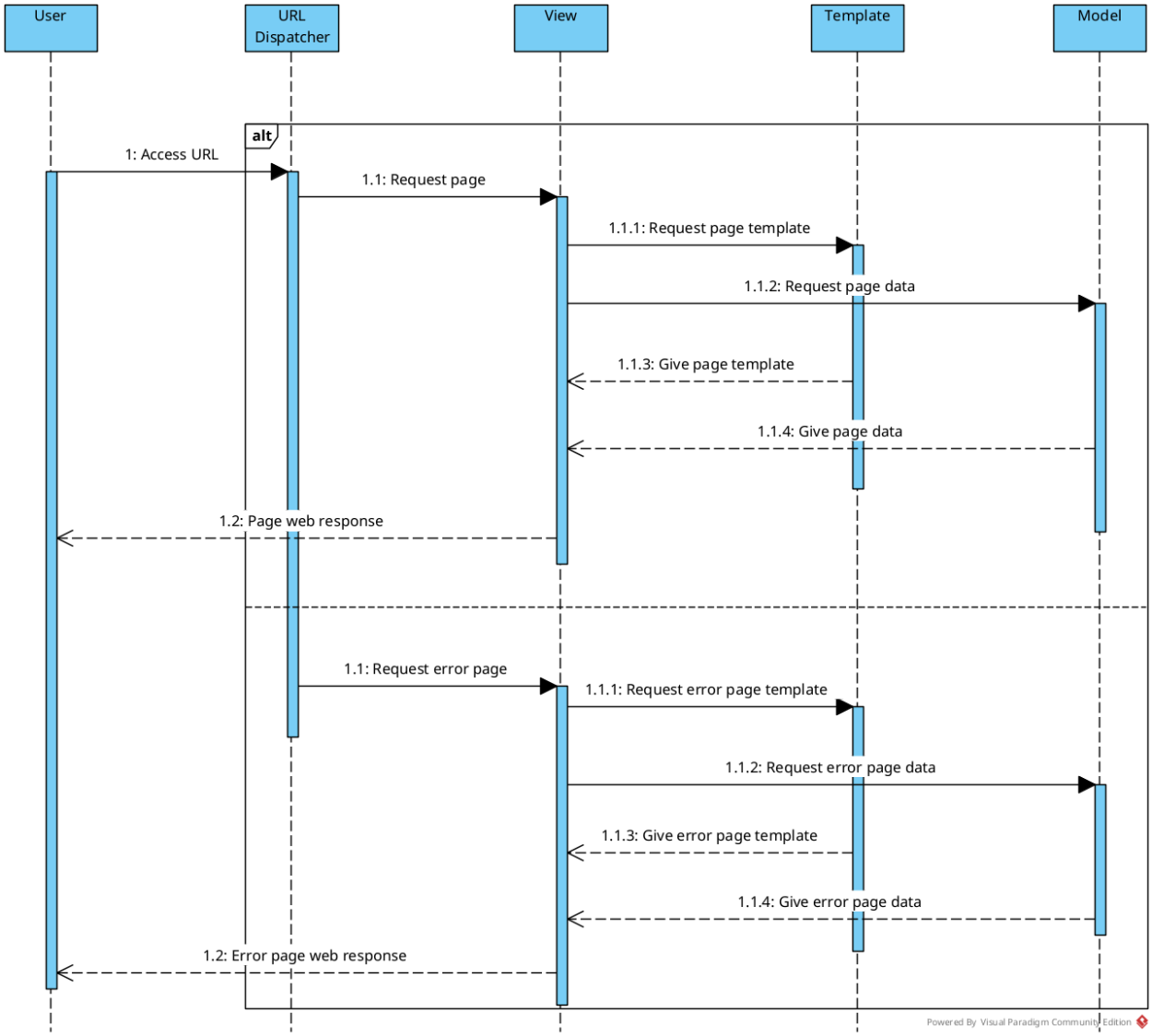
- **Learning Curve:** Developers unfamiliar with MVT or Django may have an initial learning curve. However, Django's extensive documentation and large community resources make it easy to learn and adopt.
- **Flexibility:** While MVT offers a robust structure, it might be less flexible compared to completely custom architectures. However, Django's built-in functionalities and extensibility through plugins and custom code address most development needs for a bookstore web app.

3.4. Sequence Diagrams

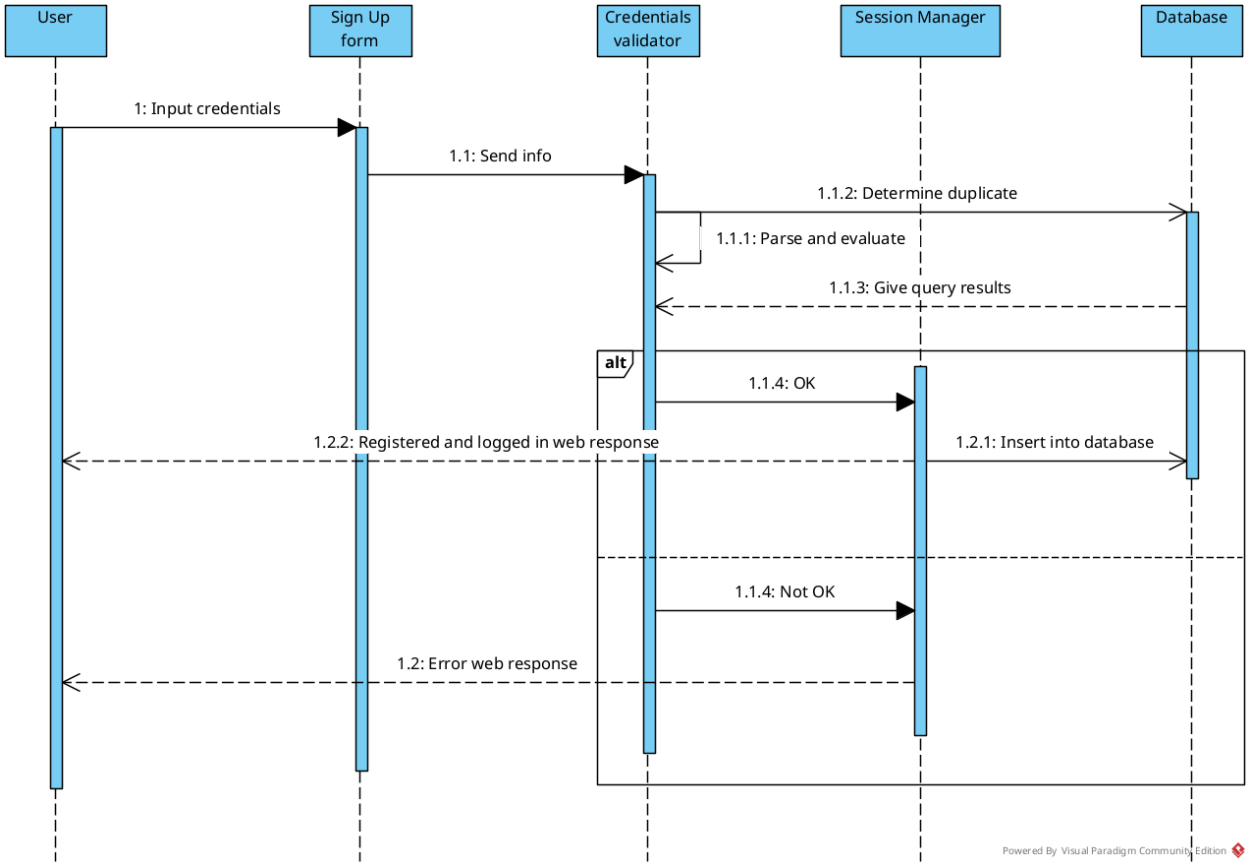
Access and navigate web-app



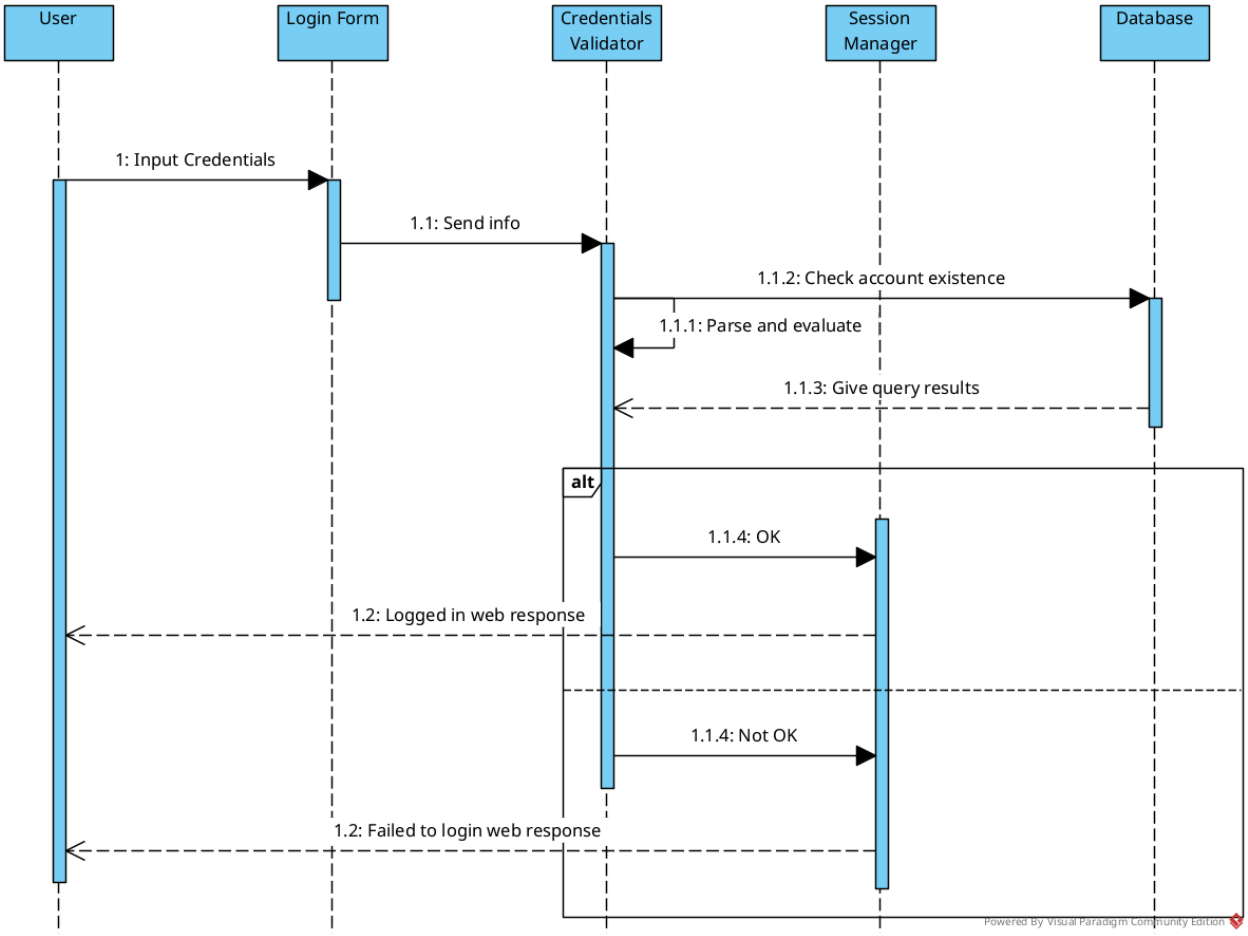
Navigate using URLs



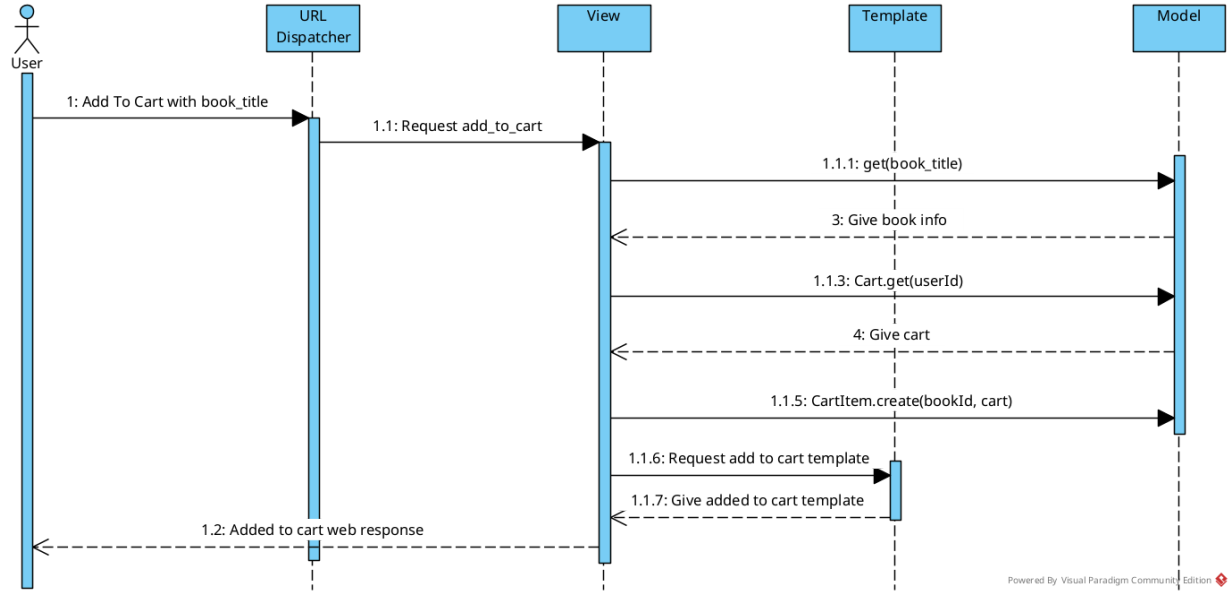
Register (Account)



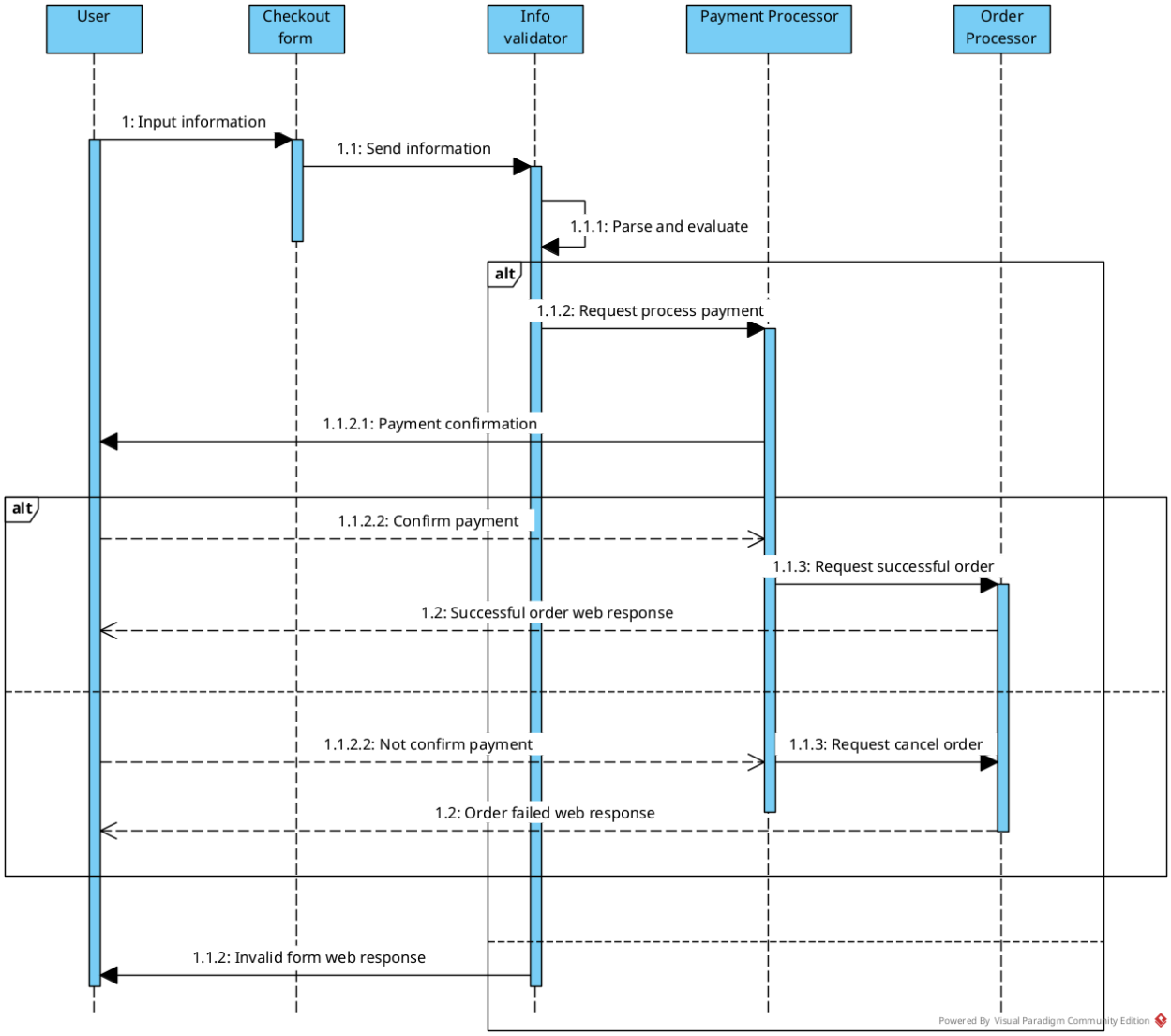
Login



Add to cart



Checkout



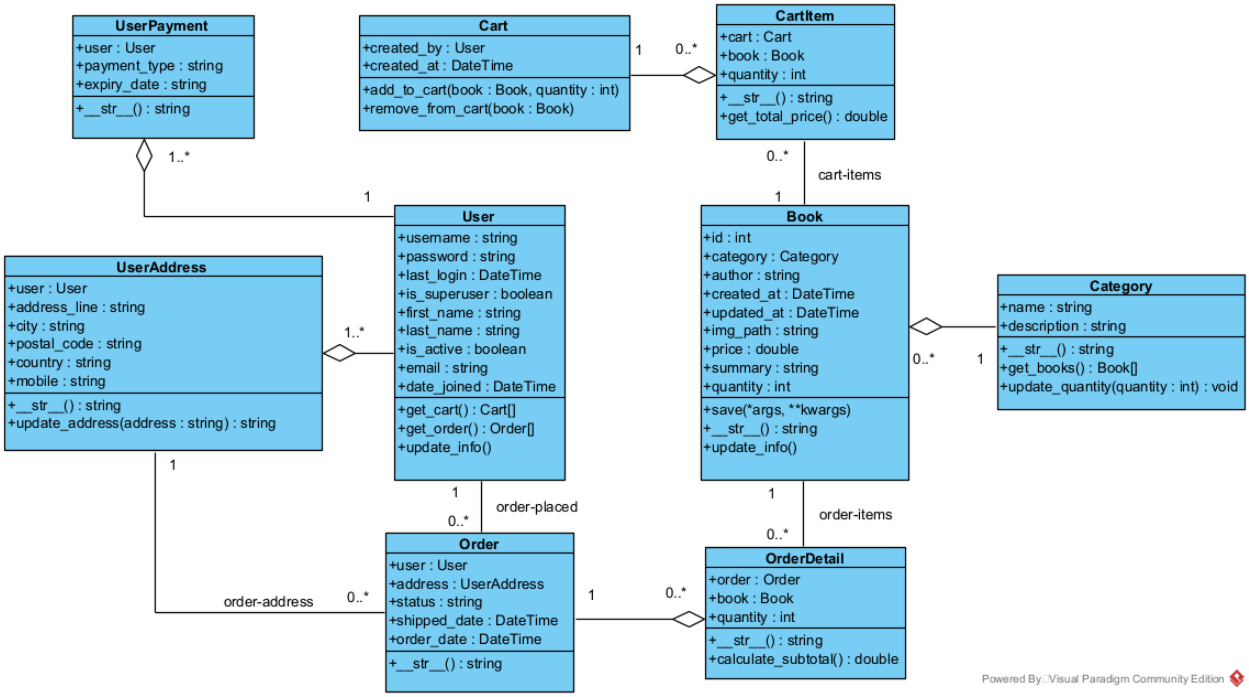
4. DATA DESIGN

- **UserPayment:** Represents the payment methods associated with each user, including options such as credit card, debit card, PayPal, or cash on delivery. Each payment method is linked to a specific user, and an optional expiry date can be provided.
- **UserAddress:** Stores the address details of users, including address line, city, postal code, country, and mobile number. Each address is associated with a user and can be used for shipping purposes.
- **Cart:** Represents a user's shopping cart, which stores the items selected by the user for purchase. Each cart is created by a specific user and includes a timestamp indicating when it was created.
- **Category:** Defines different categories or genres for books, with a name and description for each category.
- **Book:** Represents individual books available in the system, including details such as title, author, category, price, quantity, summary, and image path. The quantity of each book is randomized upon creation to simulate stock availability.
- **CartItems:** Associates books with shopping carts, indicating which books are added to each cart along with the quantity of each book.

- **Order:** Represents an order placed by a user, including details such as the user who placed the order, the shipping address, order status, shipped date, and order date. The order status can be pending, processing, shipped, delivered, or cancelled.
- **OrderDetail:** Associates books with orders, indicating which books are included in each order along with the quantity of each book.

5. COMPONENT DESIGN

5.1. Class Diagram



User Account Requirements:

- Every user account must have at least one registered address and payment method for their transactions.

Shopping Cart Functionality:

- Users can create shopping carts to gather items for purchase.
- Within each cart, users can add multiple items, each storing information about the book(s) selected and the quantity desired.

Order Processing:

- When a user proceeds to checkout, the system generates an order.

- The order includes the user's ID and the selected delivery address.
- Cart items from the user's shopping cart are automatically transferred to the order details within the generated order.

6. HUMAN INTERFACE DESIGN

6.1. Overview of User Interface

The user interface of the bookstore web app is designed to provide a seamless and intuitive experience for users to browse, search, purchase books, manage their account, and track their orders. Here's an overview of the main functionalities from the user's perspective:

Registration and Login:

- New users can register for an account by providing basic details such as username, email, and password.
- Registered users can log in using their credentials.

Browsing and Searching Books:

- Users can browse through the catalog of books available in different categories.
- A search functionality allows users to find books by title, author, or category.

Viewing Book Details:

- Users can click on a book to view its details, including the title, author, price, summary, and availability.

Adding Books to Cart:

- Users can add books to their shopping cart by specifying the quantity they wish to purchase.

Managing Shopping Cart:

- Users can view their shopping cart to review the items added, update quantities, or remove items.
- The cart subtotal and total are displayed for user reference.

Checkout Process:

- Once satisfied with their cart, users can proceed to checkout.
- Users can select a saved address and payment method, or enter new details if necessary.

Order Placement:

- Users can place their order after confirming the details.
- A confirmation message is displayed upon successful order placement.

Order Tracking:

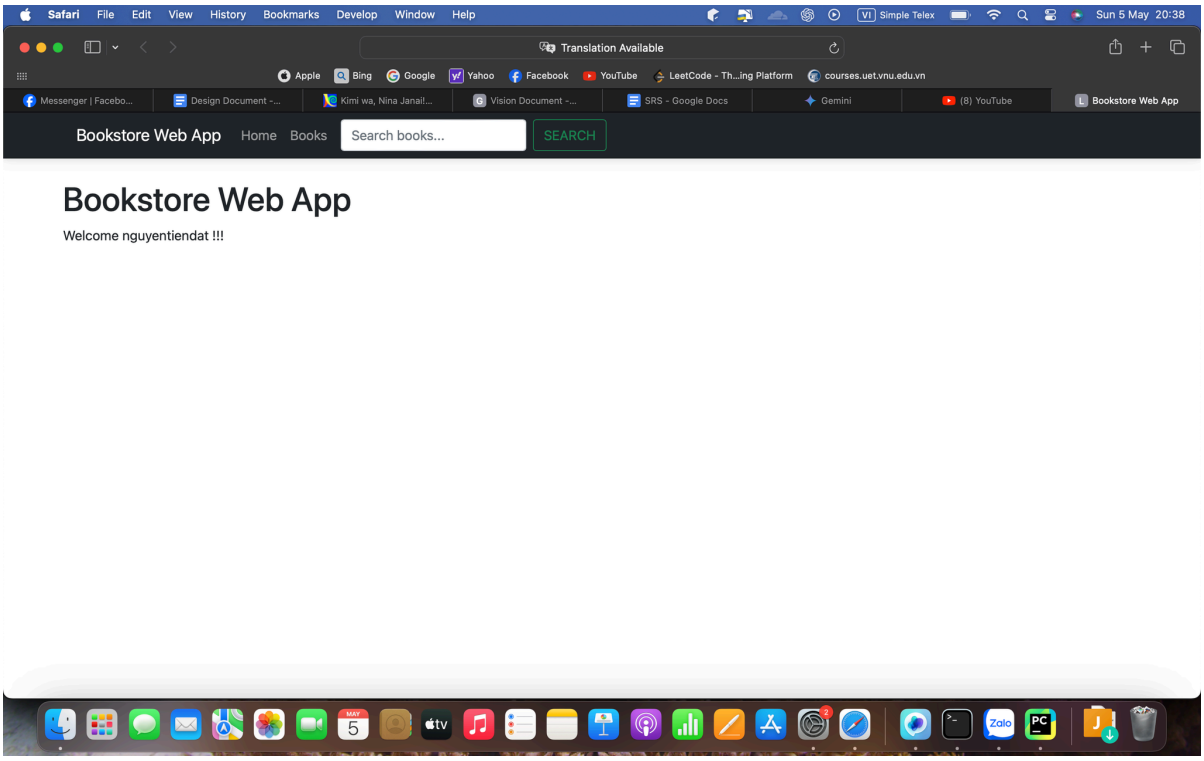
- Users can track the status of their orders, including pending, processing, shipped, delivered, or canceled.
- Order details such as order date and shipped date are provided for each order.

Account Management:

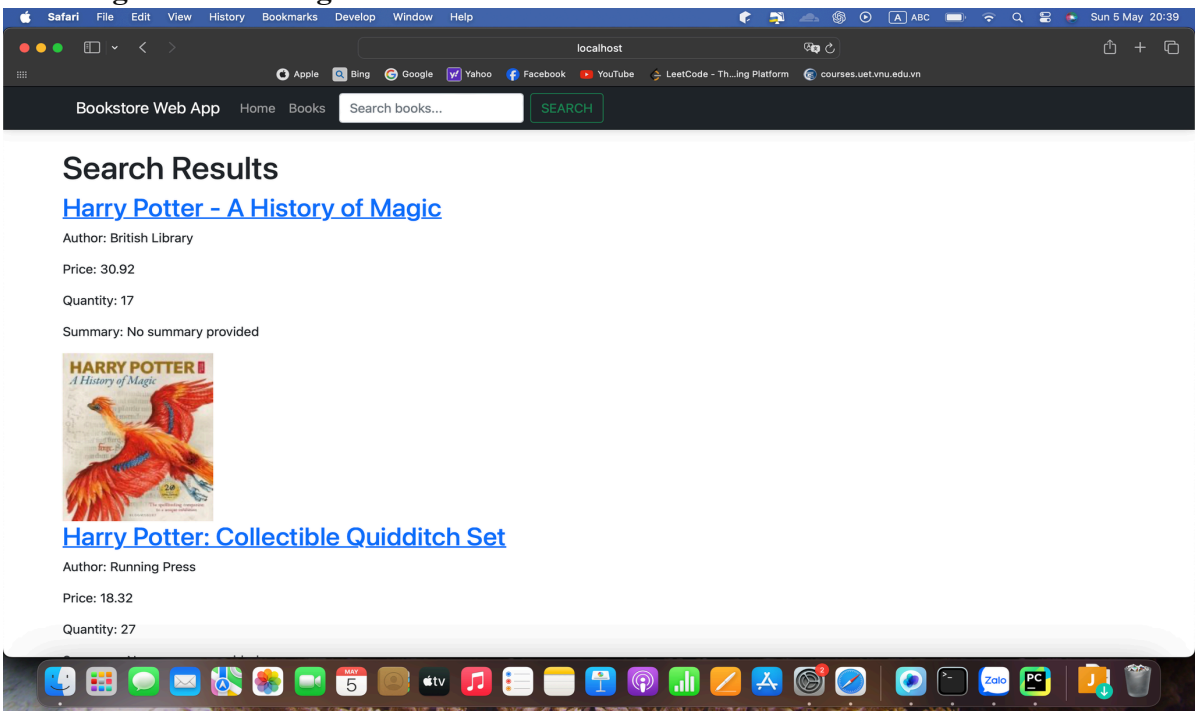
- Users can manage their account settings, including profile information, addresses, and payment methods.
- Options to edit, add, or delete addresses/payment methods are available.

6.2. Screen Images

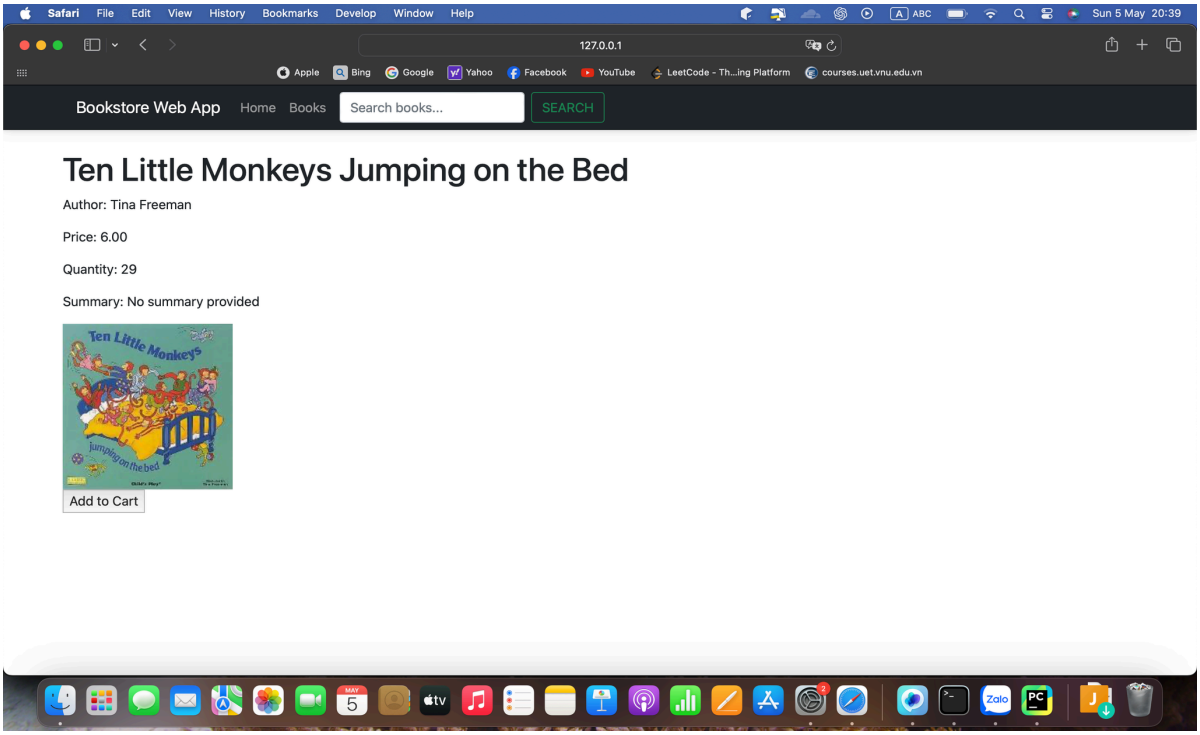
Registration and Login:



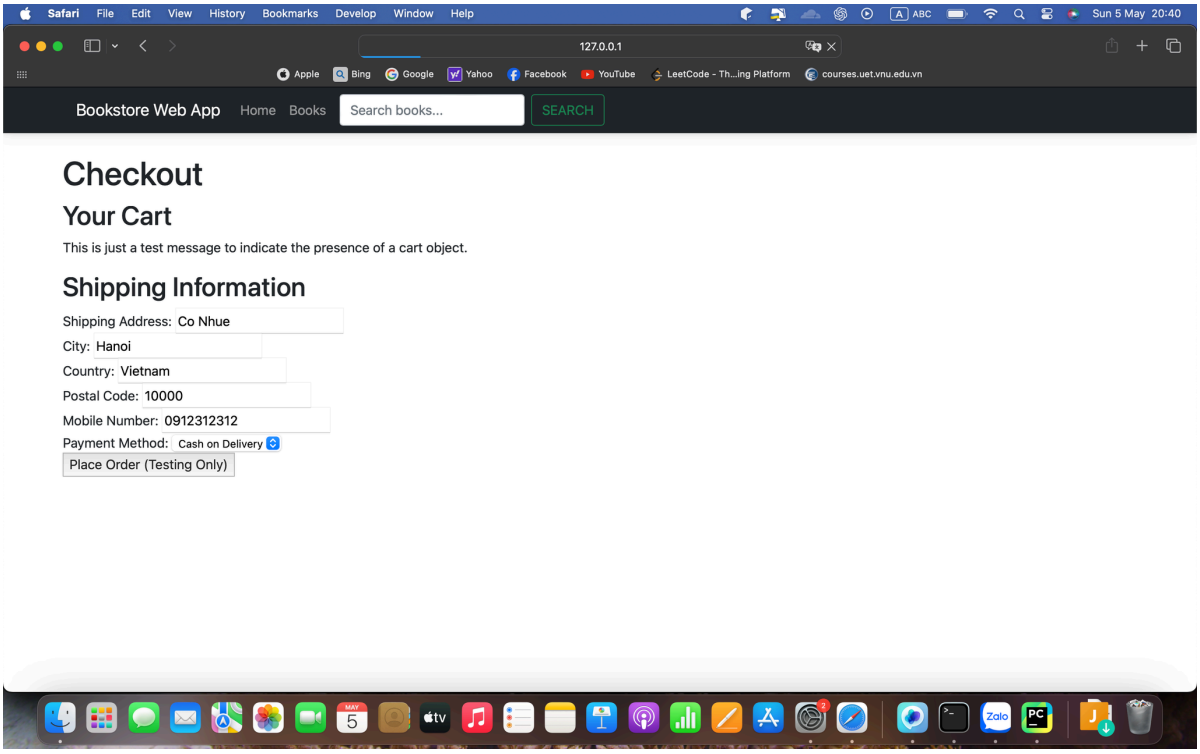
Browsing and Searching books:



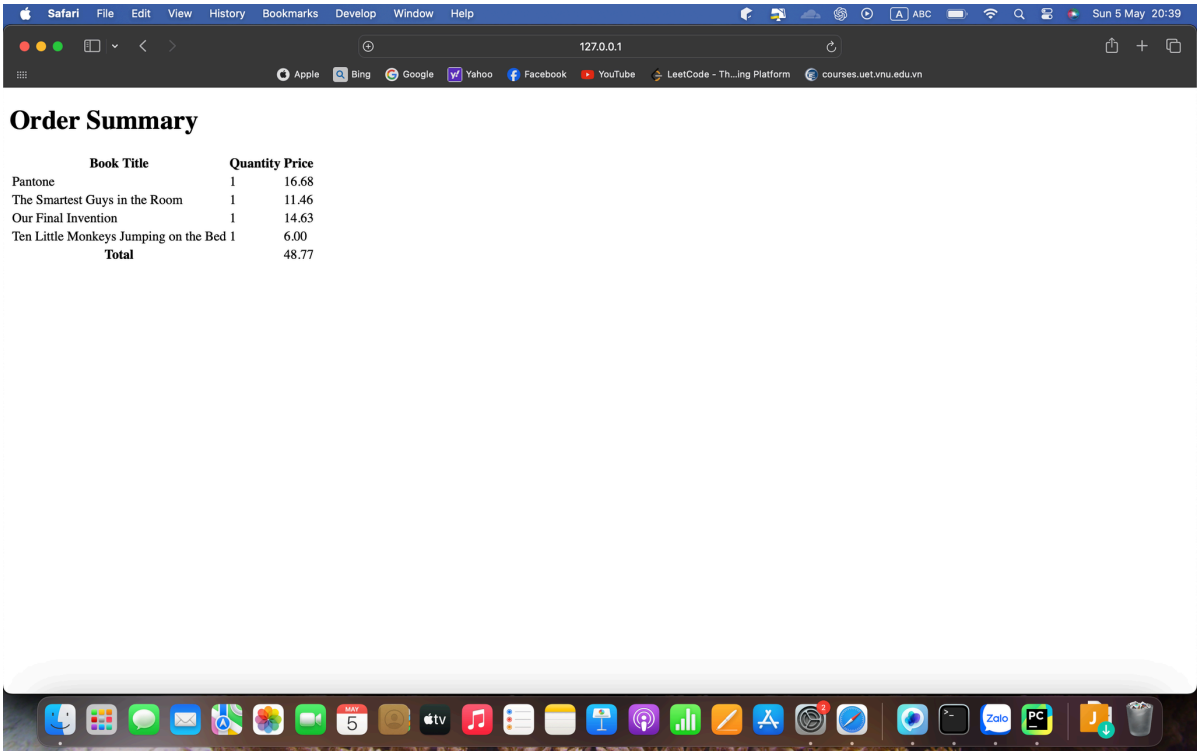
Viewing Book's Detail:



Checkout Process:



Order Summary:



7. EXTERNAL INTERFACES

- **Django Framework (>=5.0.0):** This Python web framework will provide the foundation for building the web application. Django will handle user requests, interact with the database, and generate dynamic web pages.
- **PostgreSQL Database (>=16.0):** This relational database management system will store all application data, including book information, user accounts, and potentially order history.
- **Operating System:** The web application will run on and potentially interact with any of the designated operating system environments, for instance, via system-provided software libraries.
- **Web browser communication** relies on HTTP(S) protocol, HTML/CSS/JS for UI, prioritizes security with HTTPS for sensitive data, and optimizes data transfer for faster loading. It may involve email and user forms handled securely on the server-side.

8. DEPENDENCIES

- **Browser's cookies functionality:** The app may utilize browser cookies and local storage mechanisms for storing user preferences, session data, and shopping cart information. It is assumed that users' browsers support and allow the use of cookies and local storage for these purposes.

- **Railway**: cloud database service for the web-application.