Polsko-Japońska Akademia Technik Komputerowych

Online Book Store Documentation

Sliusarenko Valeriia

S20972

Table Of Contents

1 Introduction

1.1 Purpose

Purpose of this project is to develop simple and convenient to use system for online book store. The system is to be created to help users save their time when choosing and buying books online.

1.2 Scope

The system will provide user-friendly graphical interface which will allow users to visit site anytime to view available books and add reviews to them. User can have opportunity to search book on his taste through categories or search form. The selected books can be added to the shopping cart and purchased for online reading on the web-site. At the checkout stage, books in the shipping cart will be presented as an order. Customer will need to give information about billing address, which will be stored securely on our site. The current system will securely store information about users, books and moderators of the site.

1.3 Definitions, Acronyms and Abbreviations

- **UML** – Unified Modelling Language, is a standard language for writing software blueprints. UML is used to visualize, specify, construct and document
- **HTTP** – hypertext transfer protocol. It is a service protocol.
- **GUI** - Graphical User Interface.
- **Unregistered user** - a person who can only enter the site and view basic books information.
- **Registered user** – person who provided login and password and whose personal data is stored in the site system.
- **HTML** - Hypertext Markup Language. It is language which helps to create web pages.
- **XML** – extensible markup language. It is software- and hardware-independent tool for storing and transporting data.

1.4 References

- Wikipedia – www.wikipedia.com
- Object Oriented Modeling and Design with UML-Michael Blaha, James Rambaugh
- Software Engineering, Seventh Edition, Ian Sommerville
- Database Management Systems – Navathe
- Complete Reference – J2EE – Keogh
- Java – www.sun.com

1.5 Overview

Section 2 contains a general text description of the project, user requirements, characteristics of the users of this project, non-functional requirements. Section 3 contains diagrams and analytical models for the project, use case diagrams, use case scenarios, complexity estimates. Section 4 provides design GUIs for the graphical user interfaces and use cases.

2. Overall Description

2.1 User requirements

1. The system must store information about customers: name, surname, address, email. The customer need to be divided to registered user and unregistered user. To have full access to website user need to register in system.

2. Unregistered user can enter the site and view books description, main image, reviews and price.

3.Registered user have full user access to site: adding books to wishlist, commenting books, buying books, reading books on the website.

4.The system must store information about books: Title, Description, Author, Publish date, rating and price. Books will be organized by categories, which will help to search them more faster and conveniently. Books ratings for example can be: “best-seller”, “very-popular”, “popular”, “brand new”.

5. The system must store information about moderators: name, surname, login, password, contact information. Moderator can change books description and add new books, change price of books and delete books from site.

6. The system must store information about books publishers: name, surname. The system must allow to search books by authors.

7. The system must allow for users to read books online on site after purchasing the book.

8. The system must store information about customers and their orders history. Order must contain this information: order number, order date, list of books.

9. The system must provide users ability to give rating to books, write comments under the books and delete them.

10. The system must allow users to add books to shipping cart, make a payment and have access to read books while user is registered in the system.

2.2 Users characteristics

- **Unregistered user** - a person who can only enter the site and view basic books information. Can become registered user, if he provide login and password or create account.
- **Registered user** – person who provided login and password and whose personal data is stored in the site system, have full user access to site: can addbooks to wishlist, comment books, buy books, read books if user have a bought books on the website.
- **Moderator** – person who can have access to modifying books description, delete book or add a new book, etc.
- **System –**

2.3 Functional requirements

1. **All** users can view books, their description based on category and can search books based on category, title, author, date of release.

2. **Unregistered user** can log into his account or create new account.

3. **Registered user** can add book to wishlist and check his wishlist.

4. **Registered user** can add books to the cart and check his cart.

5. **Registered user** can make order on his cart and receive order number. After payment user have email confirmation about successful payment.

6.**Registered user** can edit their shipping cart before the order.

7. **Registered user** can view his list of purchased books and read books on the site, if only they were purchased by this user.

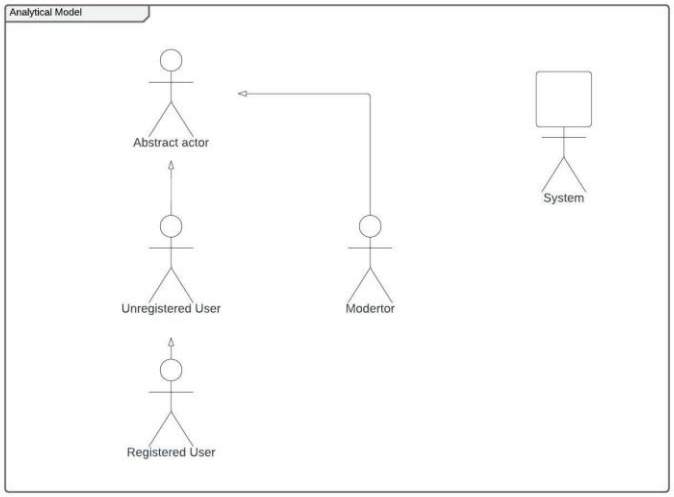
8. **Registered user** can leave a review under the books and delete his reviews.

9. **Moderator** can add new books, delete books and change description of books and their prices. Also moderator can add categories and add books to this categories.

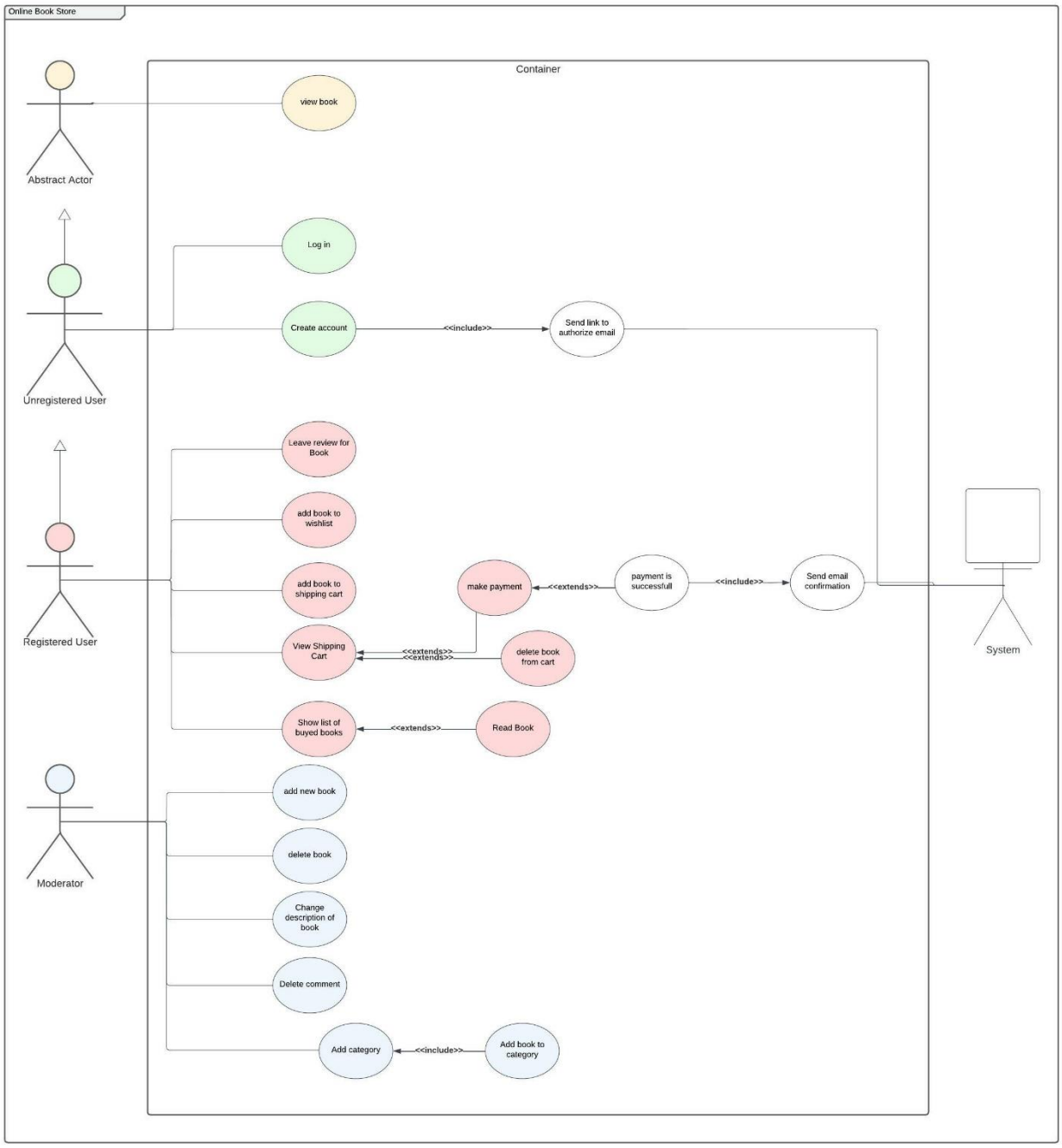
10. **Moderator** can delete comment under books, if they contain inappropriate content.

11. **System** need to send email confirmation when payment was successful and send link for email authorization while user creates account.

3.1 Analytical Model



3.2 Use Case Diagram



3.2.1 Use Case Scenario: Make payment

Actors: Registered user, System

Precondition: user is registered in the system

Description: describes process of adding book to shopping cart, viewing shopping cart and buying products from it.

Successful completion: user receives notification about successful payment from system.

Initiating business events**:** To make a purchase

Main scenario:

1. User view a book page with comments.

2. User adds book to shopping cart.

3. User view his shopping cart.

4. User taps a button to make a purchase.

5. User choses type of payment.

6. System verifies if order was successfully payed.

7. User receives confirmation about successful payment.

Alternative Scenario :

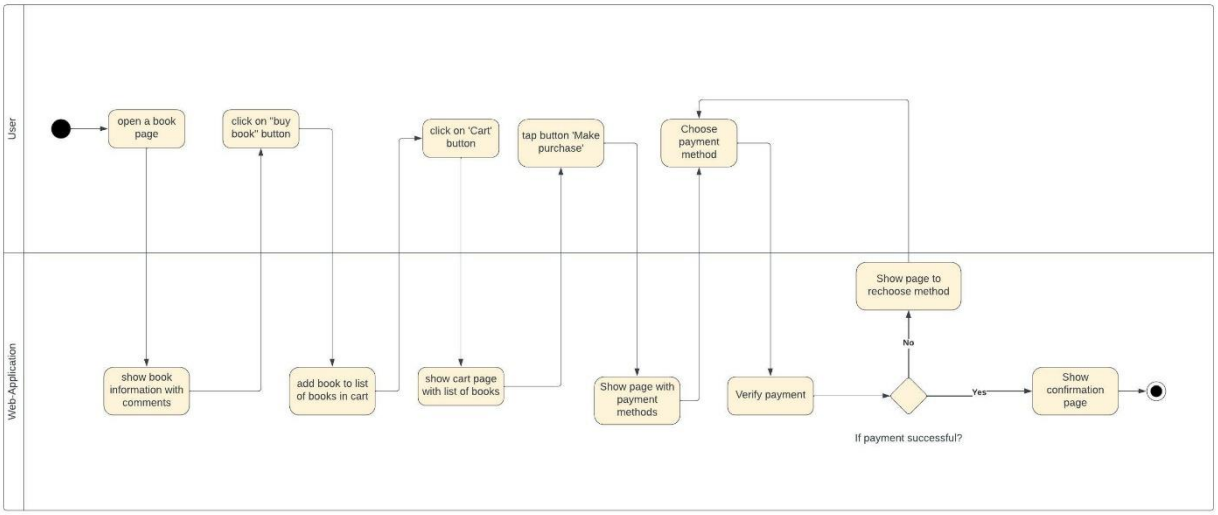
6a. System notifying user that payment was unsuccessful

6b. User chose another payment method.

Post-condition:

User purchased book or list of books.

3.2.2 Activity diagram: Make payment

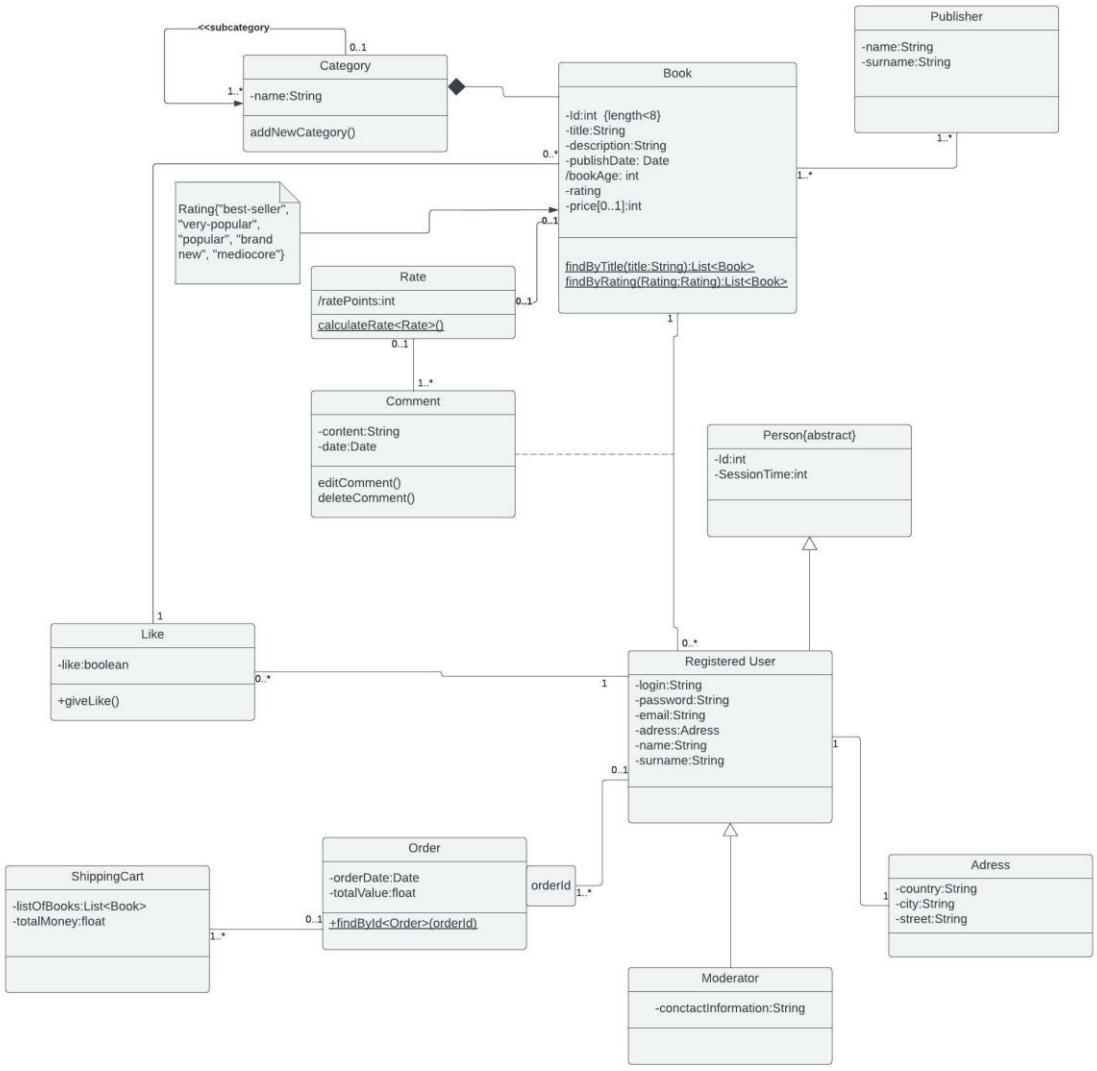


3.3 Non-functional requirements

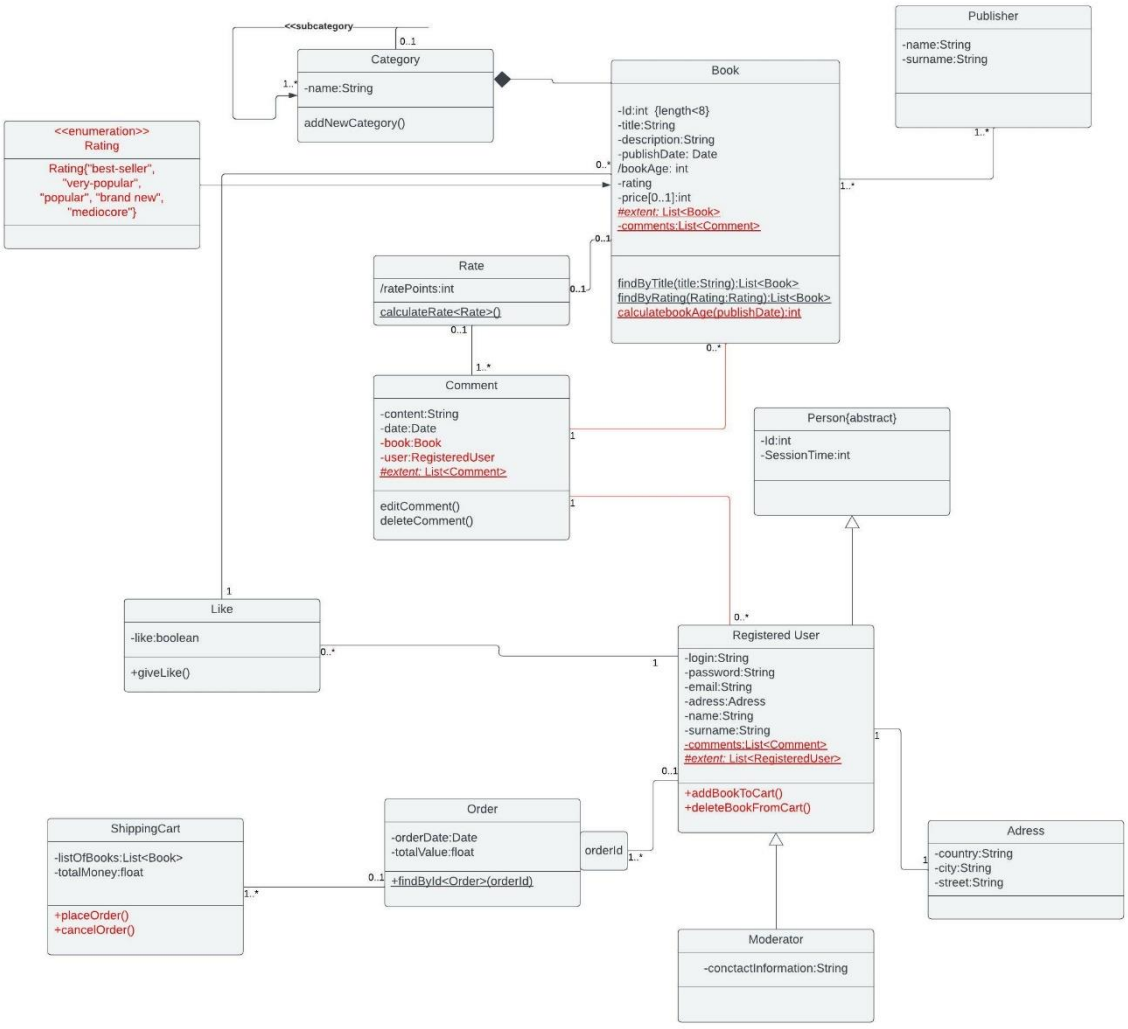
1. System must have good response speed, send it to user in less than 5 seconds
2. Easy to use, efficient, and accessible (The user interface must be quite clear and pleasurable for the beginner user).
3. Database must safely store data of users.
4. Application will have to validate all data of the user
5. Application must be available as a website on different screen sizes of devices and have responsive design

3.4 Class Diagram

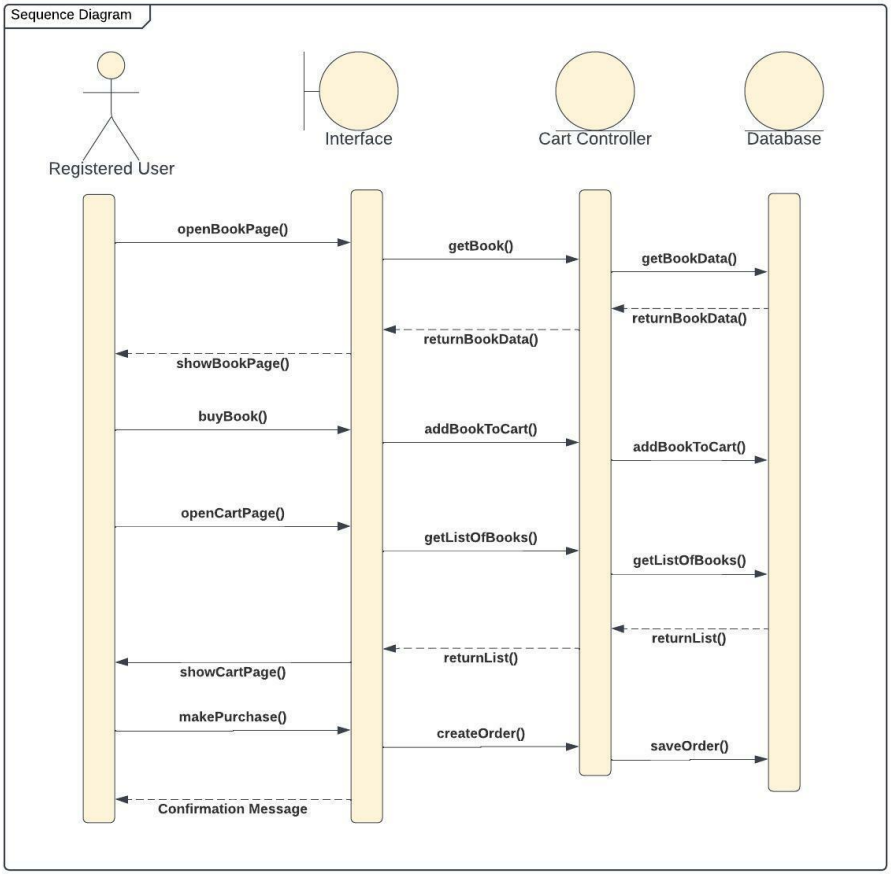
3.4.1 Analytical Class Diagram



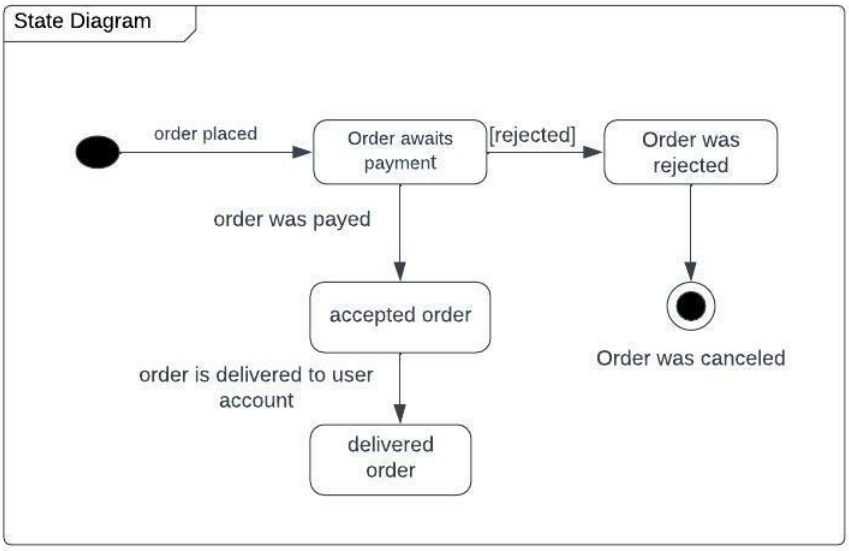
3.4.2 Design Class Diagram



3.5 Sequence diagram

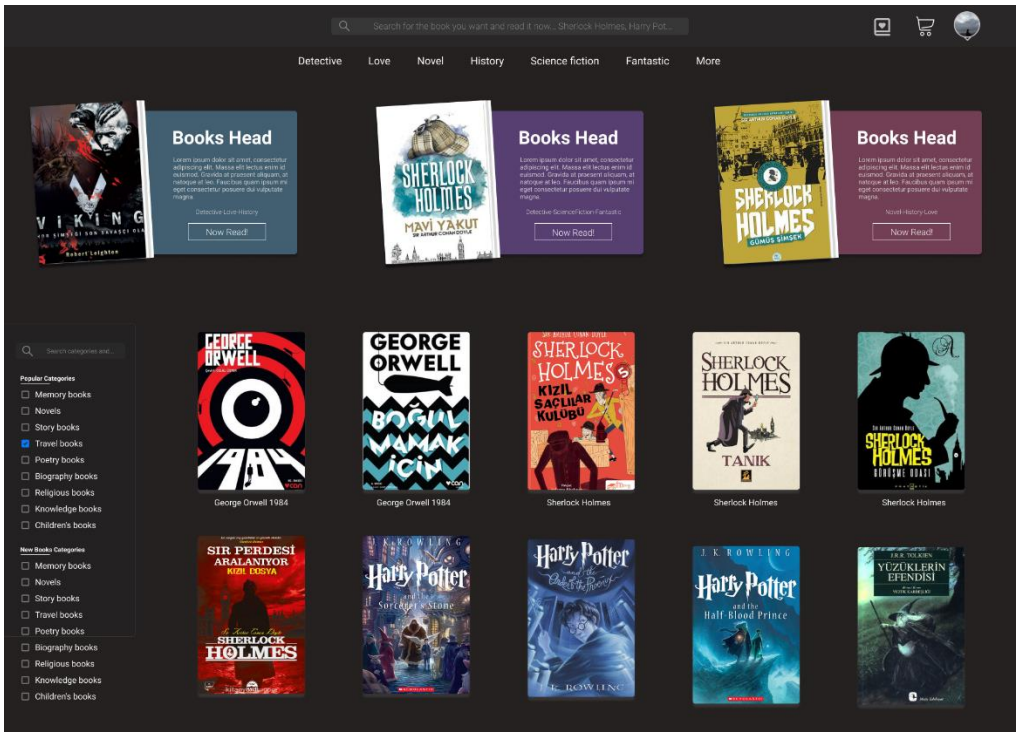


3.6 State Diagram

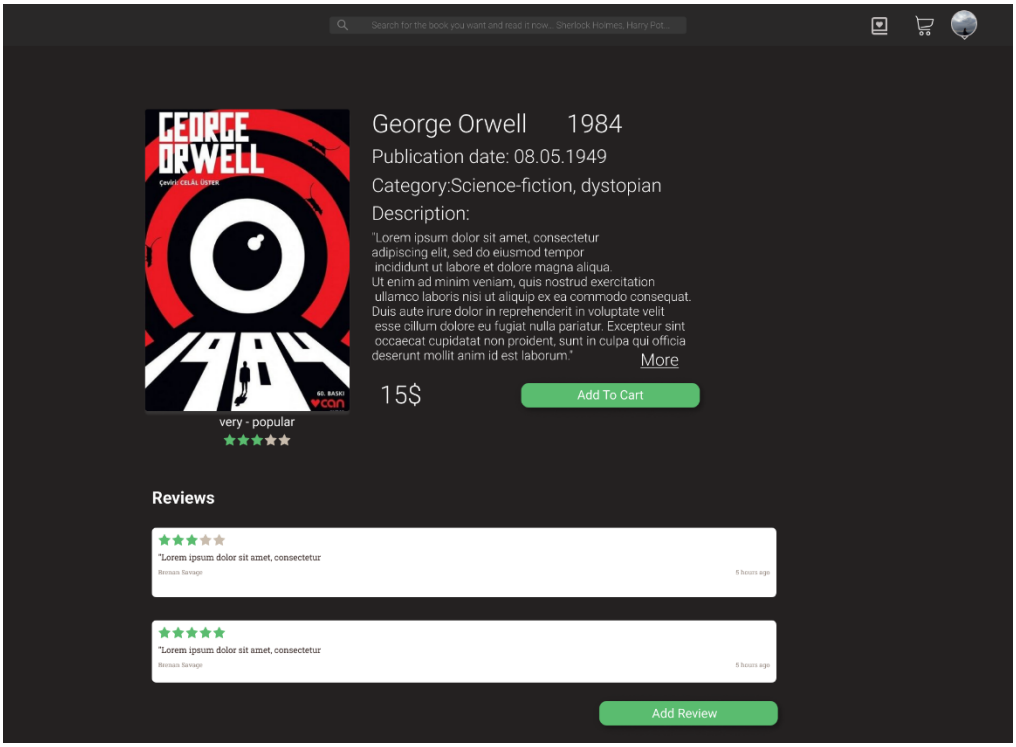


4. Graphical User Interfaces(GUI)

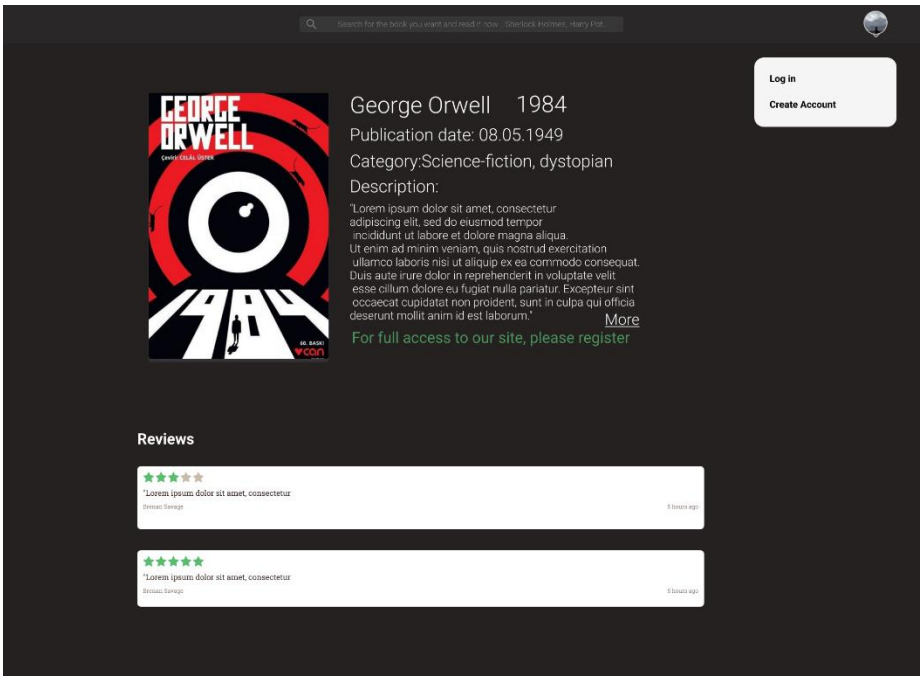
Main Page



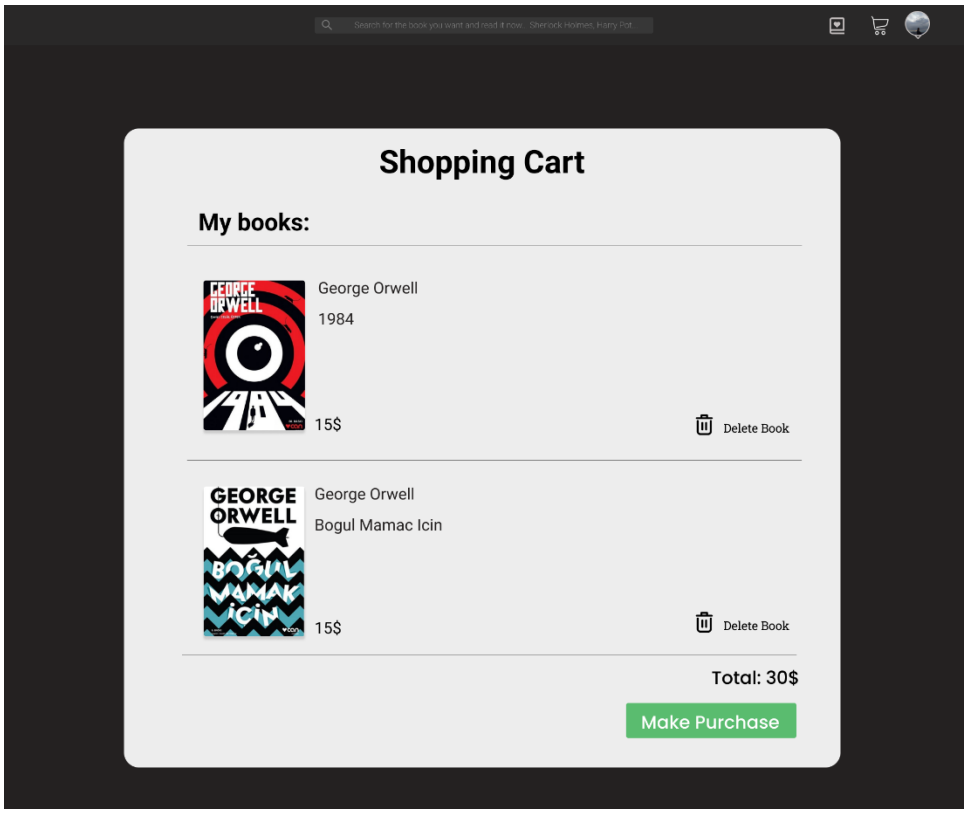
Registered User Book View



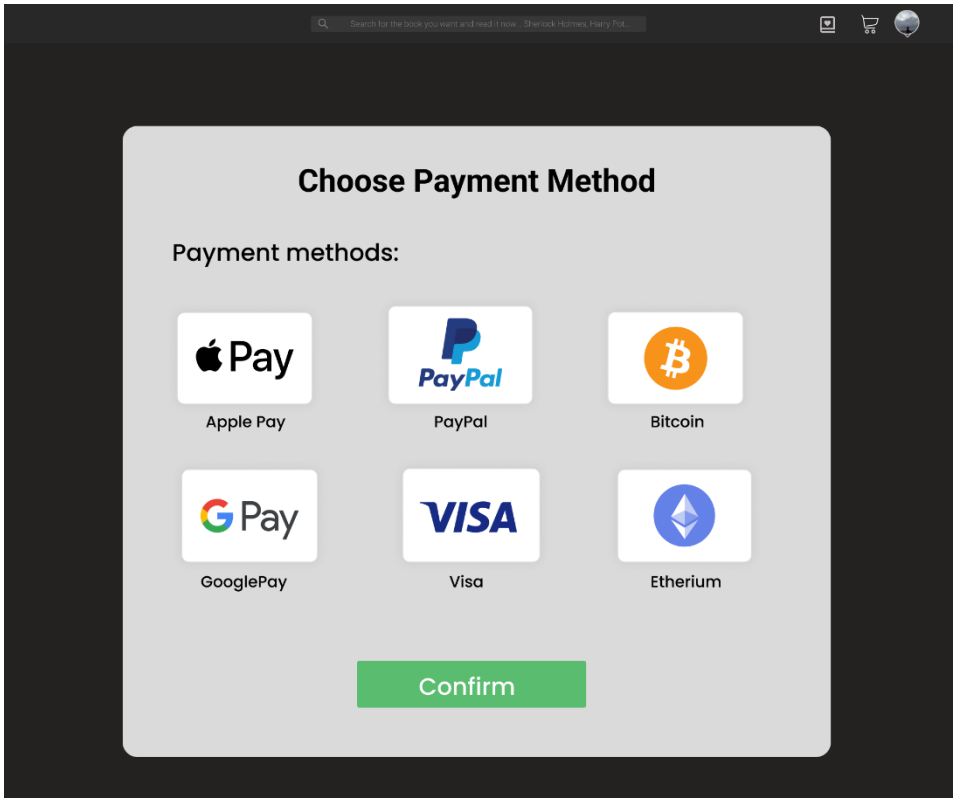
Unregistered User Book View



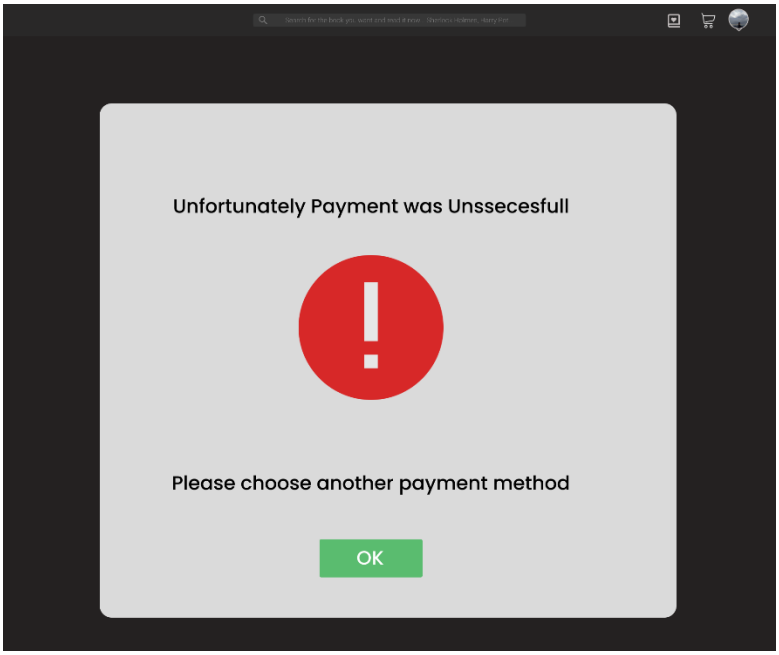
Shopping Cart Page after adding books



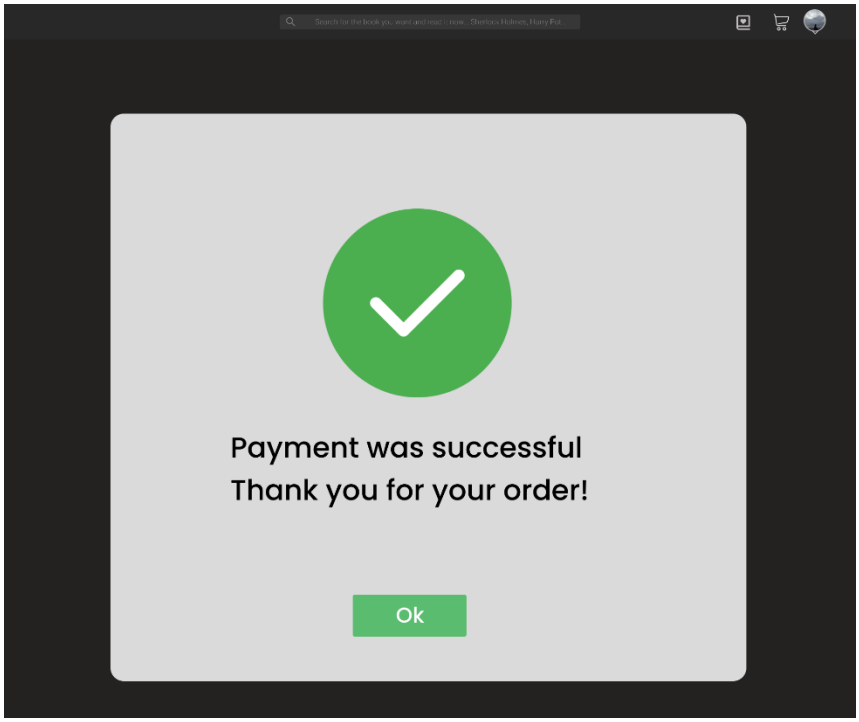
Choose Payment Type Page:



Page to rechoose payment method



Successful payment page



5 Project Decisions

The project implementation will be as Web-application, using:

- Java
- Java Spring 2.5.0
- Spring MVC
- MySQL Database Engine
- HTML
- CSS
- XML

Modification of the analytical model allows for the introduction of functionalities - possibility of adding, deleting books from shipping cart before making payment. Such methods have been introduced in Registered User class:

+addBookToCart()

+deleteBookFromCart()

And additional attributes:

-comments:List<Comment>

#extent: List<RegisteredUser>

And in Shipping Cart Class to cancel and place order from shipping cart:

+placeOrder()

+cancelOrder()

To the Book class were added this new methods:

calculatebookAge(publishDate):int

Project will have this method in Book Model, to help me sort books.

And additional attributes:

#extent: List<Book>

-comments:List<Comment>

I have changed association with attribute, because we cant implement this association in Java directly. So I have made Many to One relationships between Book-Comment-Registered User.

Also I gave changed tagged value on enumerate type which I will use for rating attribute of class Book in Java.

And have added class extend to Comment Class.

For class extends in each class where it was added on design diagram we will have 3 methods:

getExtent() - which return unmodifiable collection of class instances,

saveExtent() – which saves our collection of class instances in persistence format, in file

loadExtent() – which will load our extend collection from already created file which contains previously saved extent.