End-to-End Online Bookstore Application

Sushant Kumar, Vipul Verma

April 20, 2024

Abstract

This contains the features and transaction detalis (for deadline 6) of our application. We have included all major features we had commited to. Publisher, shipper, authors are integral to database but have no role as an user to our portal. The "further inclusion" of analytics based on customer attributes have been implemented.

Contents

1 Features

1.1 User Authentication

- Login with username and password.
- Blocking after 3 unsuccessful attempts.

1.2 Book Display

- Display books with ISBN, title, genre, author, language, availability, price, edition, and offers.

1.3 Rating System

- Ability to give star ratings to a book.

1.4 Cart Management

- Add a book to cart if it is in stock.
- Move to cart and refresh to view books.
- Remove a book with dynamic total amount change.
- Concurrency control to prevent saving cart if another user saved cart with those books beforehand.

1.5 Payment Processing

- Save the cart and proceed to buy.
- Refreshing payment portal and display of details for order.

1.6 Navigation

- Return to the home page.

1.7 Admin Analytics

- Add, Delete, Update a book
- Check trending status for books on day, month, all time
- cutomer analytics based on age, gender, country etc.
- customers' orders history

2 Architecture

2.1 Frontend

JavaFX, fxml with scene builder, command line(minor use)

2.2 Backend

Java, socket programming, fastxml framework, sql connector, MySQL

2.3 Transactions

Conflicting Cart Saving and order placement

- T1 : Adding the book to the cart of a customer
- T2 : Updating book's availability to "Out-of-Stock"

  Clearly T1 and T2 are conflicting and need to be serialized (T2 ¿ T1). Further concurrency control has been implemented by querying whether book has not gone out of stock before adding to cart on the absence of which, the connection is rolled back.
- T3 : Updating wallet for cash
- T4 : Making Final payment

  T3 and T4 are conflicting and T3 should be scheduled before T4. Connection rolled back if minimal cash not in wallet and then T is executed.

- T1 : Add a book
- T2 : Update a book
- T3 : Delete a book
- T4 : Book review update by customer

  Clearly, T1, T2, T3, T4 are independent and non conflicting