CSCI6370 Database Management Final Project Report

Book Store Management

Team:

V H Sowndarya Nookala

Rohith Lingala

Krishna Chaitanya Velagapudi

Sai Subha Sree Nadimpalli

Bavesh Chowdary Kamma

Project Overview

Bookstore Management is a versatile platform designed to facilitate seamless interaction between stakeholders in the book industry. Publishers, authors, and distributors can effortlessly add their books to the application, expanding the catalog and reaching a broader audience. Meanwhile, customers, the end-users, enjoy a user-friendly interface where they can browse, search, and purchase books. The platform not only enables sales but also fosters a community-driven experience, allowing readers to review and recommend titles, creating an engaging space for literary enthusiasts. With its intuitive design and comprehensive book offerings, Bookstore Management aims to bridge the gap between book providers and eager readers, promoting a love for literature while embracing the digital landscape.

Project Objectives

1. User Accessibility:
    - Enable users to access and explore available books without mandatory login.
    - Facilitate a seamless experience for registered users to view the complete book inventory and place orders.
2. Personalization and Filtering:
    - Implement personalized recommendations based on user preferences.
    - Offer filtering options to streamline book searches by genre, author, or specific criteria.
3. Order Management:
    - Allow users to add desired books to their carts and proceed to checkout effortlessly.
    - Provide order tracking functionalities for registered users to monitor delivery status.
4. Multiple Payment and Address Options:
    - Incorporate various payment gateways to accommodate diverse user preferences.
    - Enable users to manage and choose from multiple delivery addresses for each order.
5. Security and User Management:
    - Ensure robust security measures for user data and transactions.
    - Implement user authentication and authorization functionalities for secure access.
6. Scalability and Performance:
    - Design the application to handle a growing number of users and books without compromising performance.
    - Ensure the platform remains responsive and reliable even during peak usage.
7. User Engagement and Interaction:
    - Encourage user engagement through book reviews, ratings, and recommendations.

- Foster a community-driven platform for readers to interact and share insights about books.

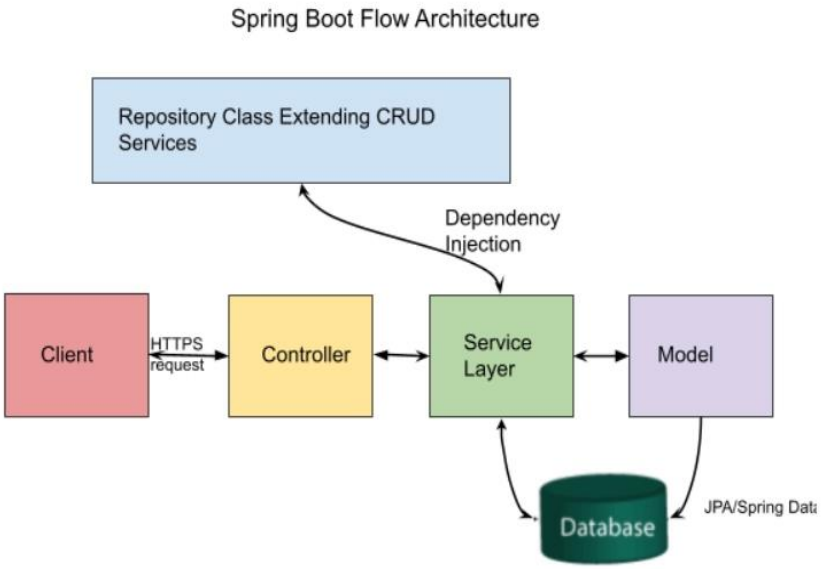
Database Schema:

- Our database schema has the following entities.
- Users, shipping_address, payment_cards, shopping_cart, books, orders, cart_books, book_inventory.

Technologies:

- Backend Programming: Java
- Web Framework: Spring Boot
- DBMS Access: Hibernate
- Database: MySql
- Frontend Programming: Angular

Software Architecture:



UML Class Diagram

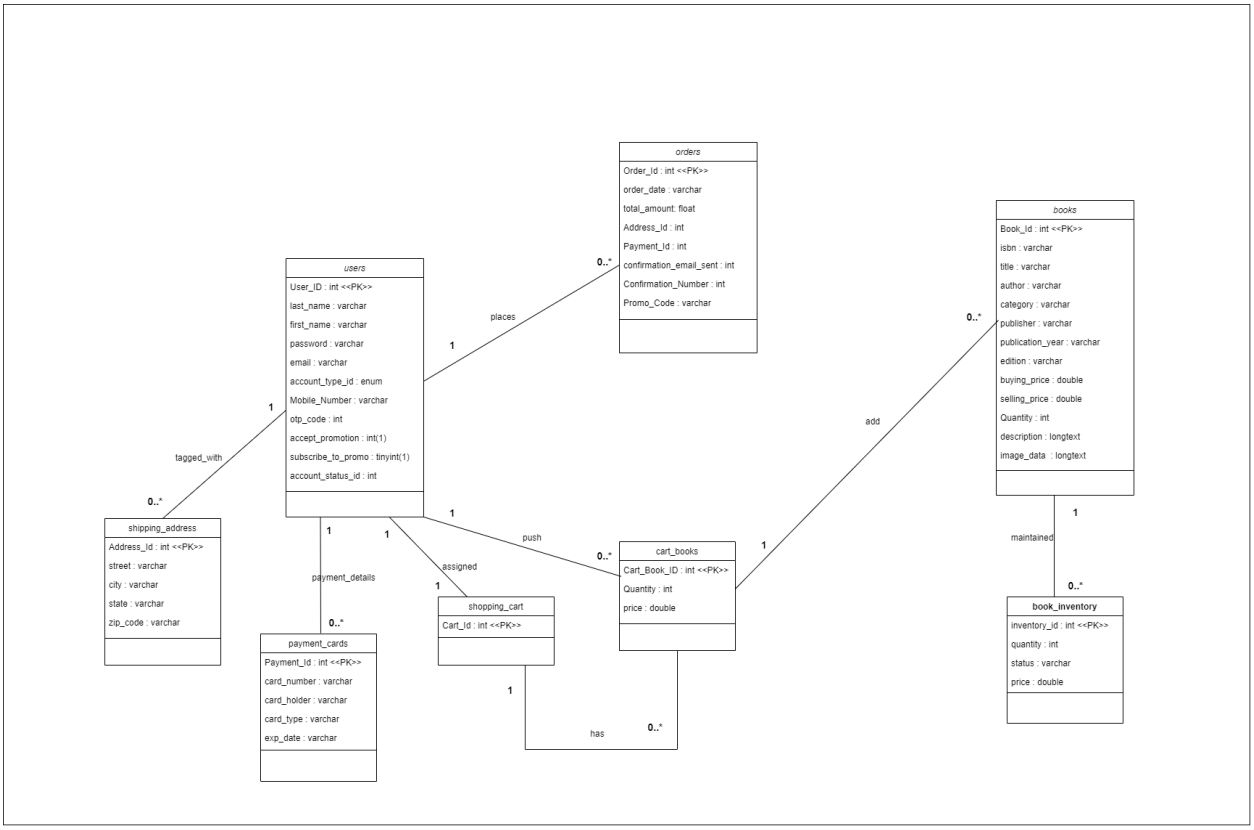
Our UML class diagram consists of 8 classes namely users, books, orders, shipping address, payment cards, shopping cart, cart books, book inventory. Below are the count of relationships in our tables.

0 many to many relationships

8 one to many relationships

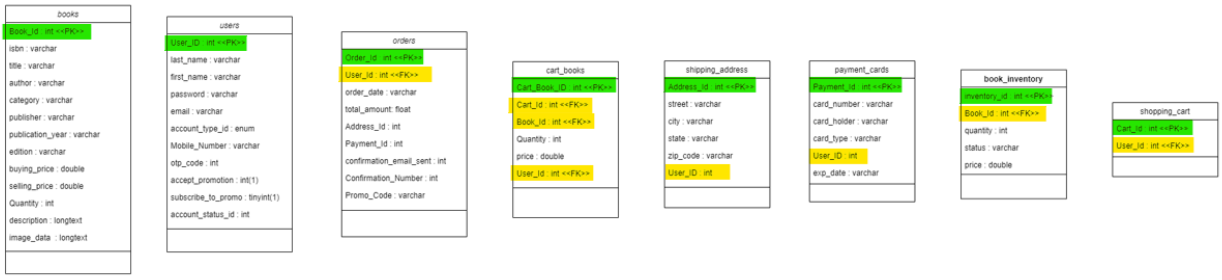
8 primary keys

8 classes



UML Class diagram

Relational Model



- There are 8 number of Primary Keys which are highlighted in Green.
- There are 8 number of Foreign Keys which are highlighted in Yellow.

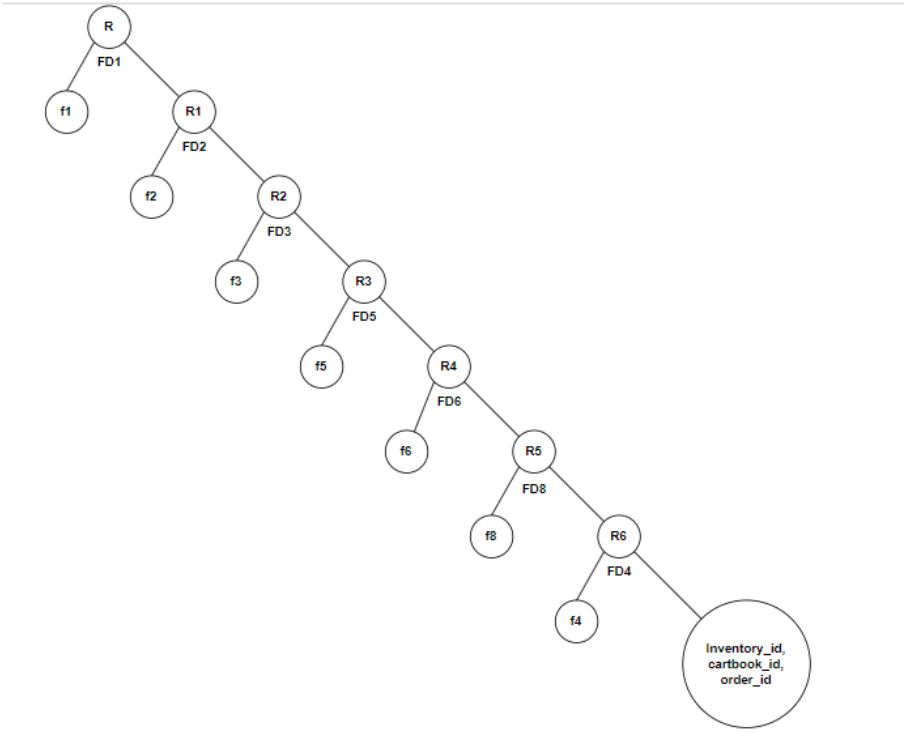
Functional Dependencies

The following are the functional dependencies:

- FD1: Book_id →ISBN, title, author, category
- FD2: Inventory_id →Book_id, Quantity, Status, price
- FD3: Cart_book_id → Cart_id, Quantity, Price
- FD4: Order_id → User_id, order_date, total_amt, add_id, pay_id, ord_sts, confirmation_email, confirmation_number
- FD5: Payment_id →Card_no, Card_holder, Card_type, expiry_date
- FD6: Add_id → street, city, state, zipcode
- FD7: Cart_id → User_id
- FD8: User_id → lname, fname, email, pwd, phone_no, otp

BCNF Decomposition

- F1: Book data (Book_id,ISBN, title, author, category)
- F2: Inventory data (Inventory_id ,Book_id, Quantity, Status, price)
- F3: Cart data (Cart_book_id , Cart_id, Quantity, Price)
- F4: Order data (Order_id , User_id, order_date, total_amt, add_id, pay_id, ord_sts, confirmation_email, confirmation_number)
- F5: Payment data (Payment_id, Card_no, Card_holder, Card_type, expiry_date)
- F6: Address data (Add_id , street, city, state, zipcode)
- F8: User data (User_id, lname, fname, email, pwd, phone_no, otp)



BCNF Decompostion for the Relation R

3NF Synthesis

Step 1: cleanup FD's

1.1: Singleton

Book_id →ISBN

Book_id→title

Book_id→author

Book_id→category

Inventory_id→status

Inventory_id→quantity

Inventory_id→price

Cart_Book_id→cart_id

Cart_Book_id→quantity

Cart_Book_id→price

Cart_Book_id→user_id

Add_id→street

Add_id→city

Add_id→state

Add_id→zipcode

Order_id→user_id

Order_id→order_date

Order_id→total_amt

Order_id→add_id

Order_id→payment_id

Order_id→order_status

Order_id→confirmation_email

Order_id→confirmation_number

Payment_id→card_no

Payment_id→card_holder

Payment_id→card_type

Payment_id→expiry_date

Cart_id→user_id

User_id→lname

User_id→fname

User_id→email

User_id→pwd

User_id→phone_no

User_id→otp

1.2 : Extremeous data : No Extremeous data

1.3 : Redundant Data : Using Transitivity property there is one redundant data which is highlighted in yellow colour using the singleton FD's highlighted in Green.

Step 2: Merge FDs with same LHS

- FD1: Book_id→ ISBN, title, author, category
- FD2: Inventory_id → Book_id, Quantity, Status, price
- FD3: Cart_book_id → Cart_id, Quantity, Price
- FD4: Order_id→ User_id, order_date, total_amt, add_id, pay_id, ord_sts, confirmation_email, confirmation_number
- FD5: Payment_id →Card_no, Card_holder, Card_type, expiry_date
- FD6: Add_id→street, city, state, zipcode
- FD7: User_id →lname, fname, email, pwd, phone_no, otp

Step 3: Form Tables from the FDs

- R1 = Book_id, ISBN, title, author, category
- R2 = Inventory_id, Book_id, Quantity, Status, price
- R3 = Cart_book_id, Cart_id, Quantity, Price
- R4 = Order_id, User_id, order_date, total_amt, add_id, pay_id, ord_sts, confirmation_email, confirmation_number
- R5 = Payment_id, Card_no, Card_holder, Card_type, expiry_date
- R6 = Add_id, street, city, state, zipcode
- R7 = User_id, lname, fname, email, pwd, phone_no, otp

Step 4: Find the subset tables

- There are no subset tables.

Step 5: Lossless Test and adding a global key

If there is a table for each functional dependency and $(R_i)^+ = R$, then we can say that the data is lossless.

Let us apply closure for each table.

Here, we can observe that,

$(R_1)^+ = R_1$

$(R_2)^+ = R_2$

$(R_3)^+ = R_3$

$(R_4)^+ = R_4$

$(R_5)^+ = R_5$

$(R_6)^+ = R_6$

$(R_7)^+ = R_7$

As there is no $(R_i)^+ = R$ Therefore, from the above steps, we can say that the data is lossy.

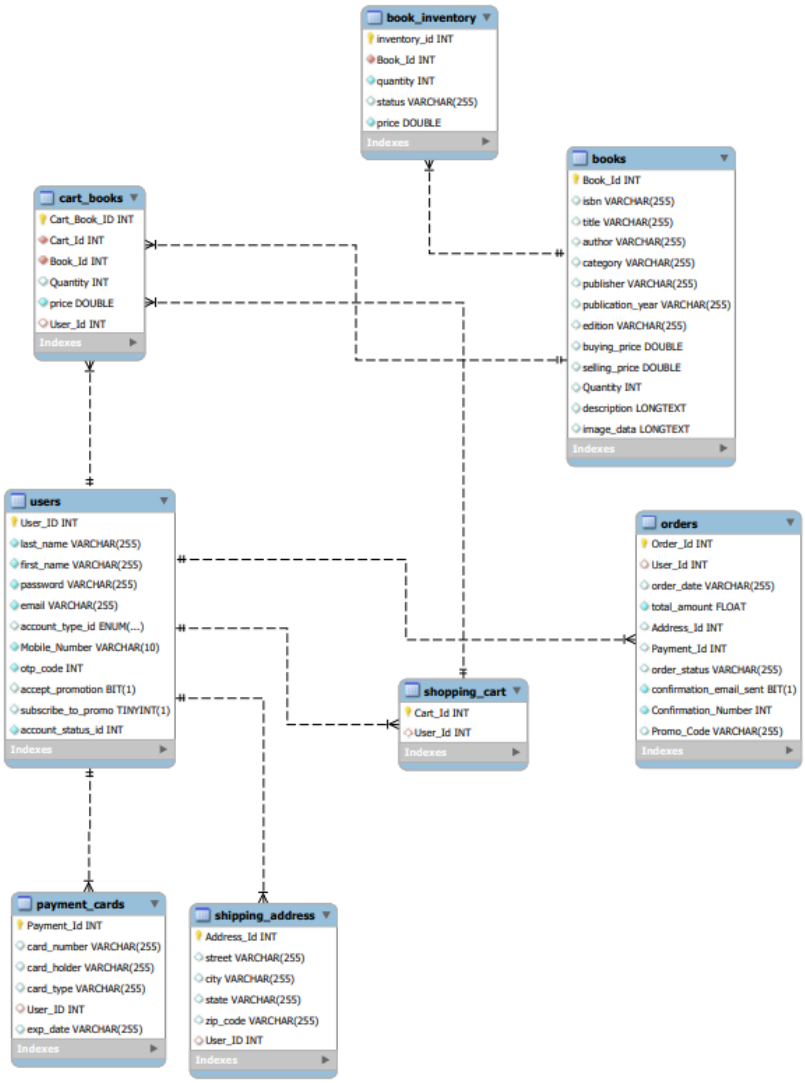
We Impose a global key i.e. “Inventory_Id, Cartbook_Id, Order_Id.”

Comparison of schemas:

	UML	BCNF	3NF
Reduction in Redundancy	**Good**	**Good**	**Good**
Avoidance of Anomalies	**To an extent**	**Yes**	**Yes**
Losslessness/Lossy	**Lossy**	**Lossless**	**Lossless**
Dependency Preservations	**Yes**	**No**	**Yes**
Overall Quality	**Satisfactory**	**Good**	**Good**

Comparing all the schemas we finalized to choose BCNF schema decomposition as a template to design the database.

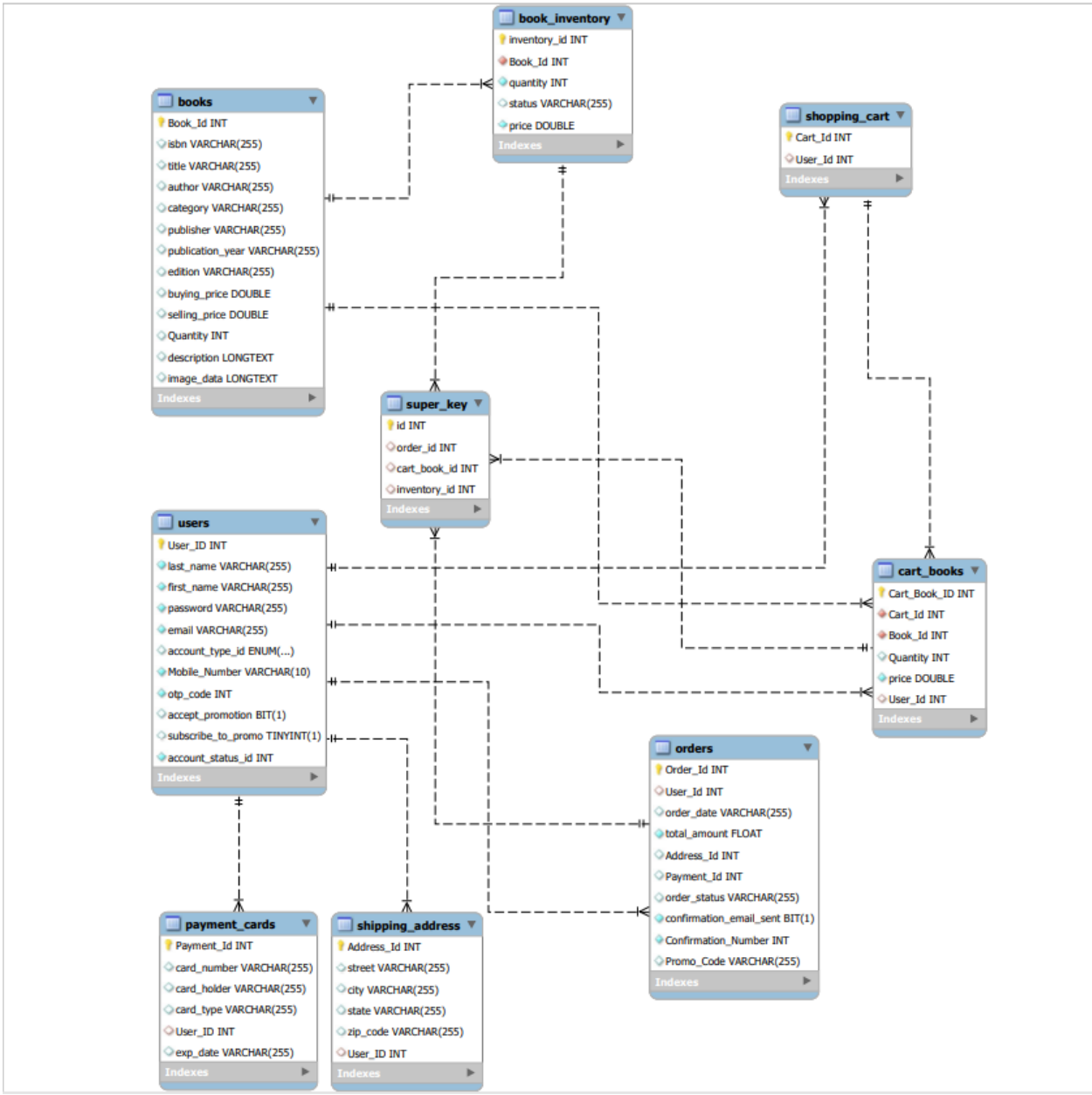
Final Schema:



In the above figure, the constraints that are enforced on the schema are:

- Foreign Key Constraint
- Primary Key (Unique and not null)

Note: Optimized database schema maintaining a table with super key relation table.



Security Handling

- To achieve security, we used JWT barrier tokens which are restricted to the sessions but not to the User.
- Spring boot the tech stack which we used internally provides inbuilt security methods which are used to encrypt the passwords.
- Methods internally provide SHA-256 for encryption.

Conclusion

In conclusion, the Bookstore Management Application endeavors to revolutionize the book-buying experience by seamlessly connecting stakeholders within the literary realm. With a user-centric approach, the platform aims to empower both readers and providers. It enables effortless browsing and ordering for users while offering personalized recommendations and a community-driven space for book enthusiasts. By prioritizing user accessibility, diverse payment options, robust security measures, and scalability, this application seeks to redefine how books are discovered, purchased, and enjoyed. Ultimately, it aspires to create a digital haven where the love for literature thrives and where users find delight in discovering, sharing, and embracing the world of books.