Advanced Database Concepts
Project Milestone 3

Online Bookstore System

Date of Submission: December 8, 2011

By

GROUP 4

Rohit Alekar

Murali Karthik

Jayesh Kawli

Niranjana Ravi

Shiva Ray

Fei Teng

Abstract:

The project online bookstore system deals with the application of World Wide Web to provide a book purchasing service to consumers so that they can perform transactions from anywhere and get a good deal with combination of service and cost.

In this project task of buying a book is simplified from a customer's point of view. A customer who wishes to buy a book will simply visit the website. He can look for books by various criteria such as author, title or category. Also, a shopping cart can store a user's book selections between sessions.

After customer has finished adding books to his cart, he can simply complete the transaction by giving additional details such as payment and billing information. Payment information will be authorized by system by checking customer payment details from another authorized source. Once this process is completed, customer will be notified about confirmation and the details of the transaction will be stored in system for future analysis and evaluation.

Administrators will have control over the major functionalities of the website. One or more administrators can add, or delete book to database or update the inventory details.

In exceptional cases, if books required by the customer are not found in database, the corresponding book will be added to the list of books currently not available in database. This list can then be used by administrator to add books into the database.

INDEX

1. Introduction

The increased use of Internet has led to many real world applications being deployed on the World Wide Web for easy access, high efficiency and fast processing. An online book store application takes advantage of these features and provides users with the facility to easily purchase books from the comfort of home.

Online book store system is very much similar to a real world book store but it is offered virtually. There is no need for a customer to go to a book shop, search and buy books. It is convenient for people who live in far off places or are too busy to visit a real bookshop.

A facility is provided by the website developers so that a customer can browse/search books and add them to the cart for later purchase. A facility for discounts is also provided. In case a particular book is not found in the database, the details of book will be noted into another table which contains list of all the books requested by customers but not available for purchase. This table will be reviewed by administrators so that corresponding books will be added into the database.

Once customer finished adding books to cart, next process is checking out. Customers have to give payment and billing details. Payment details will be verified by the system and once this is done, verified confirmation page will be displayed to user. This information will be added to order database which contains order details such as order id, number of items, customer details and shipping address for delivery.

The details of order will be stored in database purchase history for later evaluation and analysis by the system so that administrators can see shopping trends among various customers as well as evaluation of purchase of different books to check popularity.

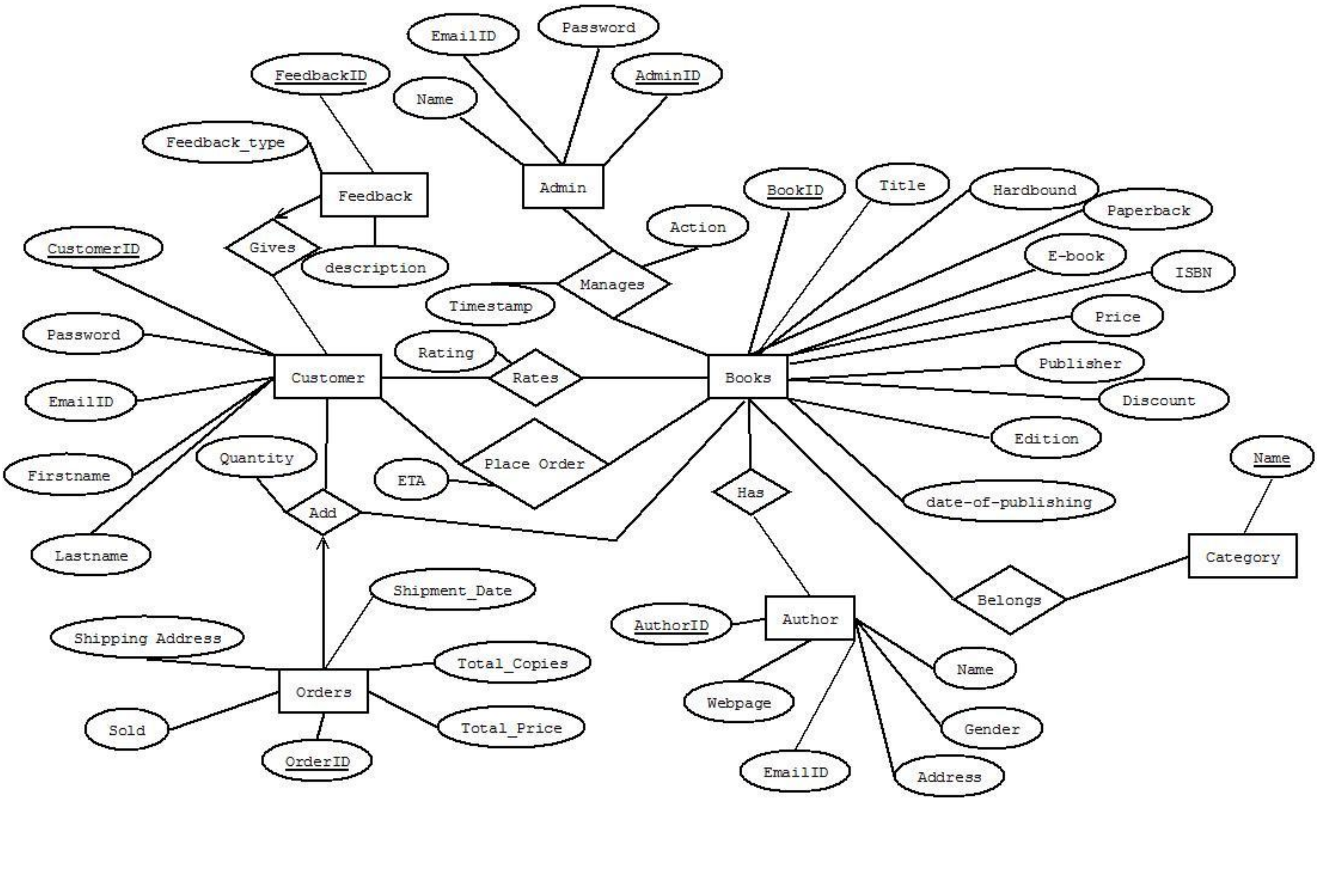
2. Design Diagrams:

Design of the system involves various models for high level representation of the system. Before creating actual system data models give high level representation of the system to be built. Following are the data models used in the project design phase.

2.1 E-R diagram

E-R diagrams represent conceptual and abstract level of system design in the initial phase of system development. They are used to represent various entities, their attributes and relationships among entities in graphical manner. These diagrams are used for data modeling in relational database which then are used to create database schema.

E-R diagram is a conceptual data model which is then modified into logical data model such as relational model.

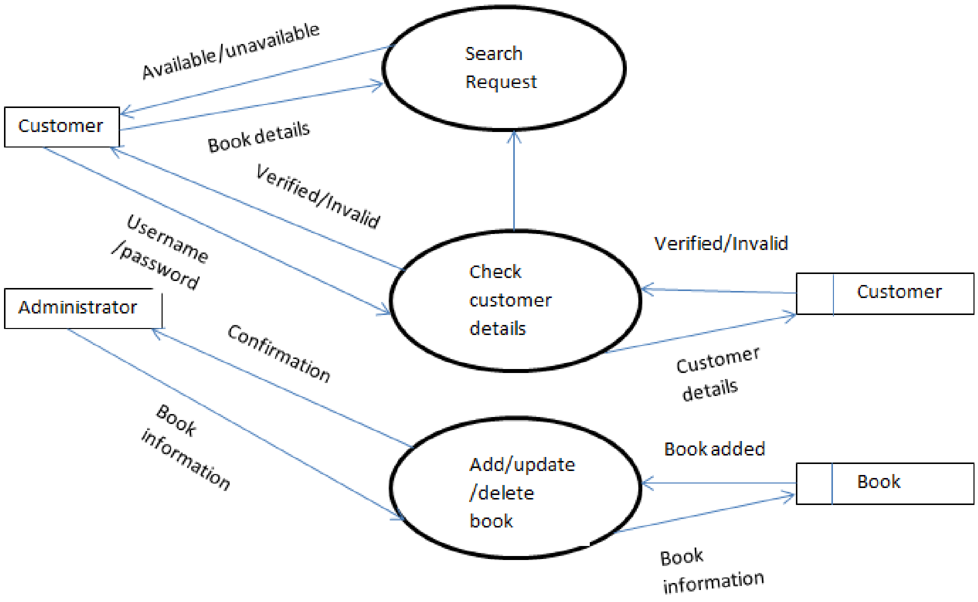


E-R Diagram

2.2 Data flow diagram

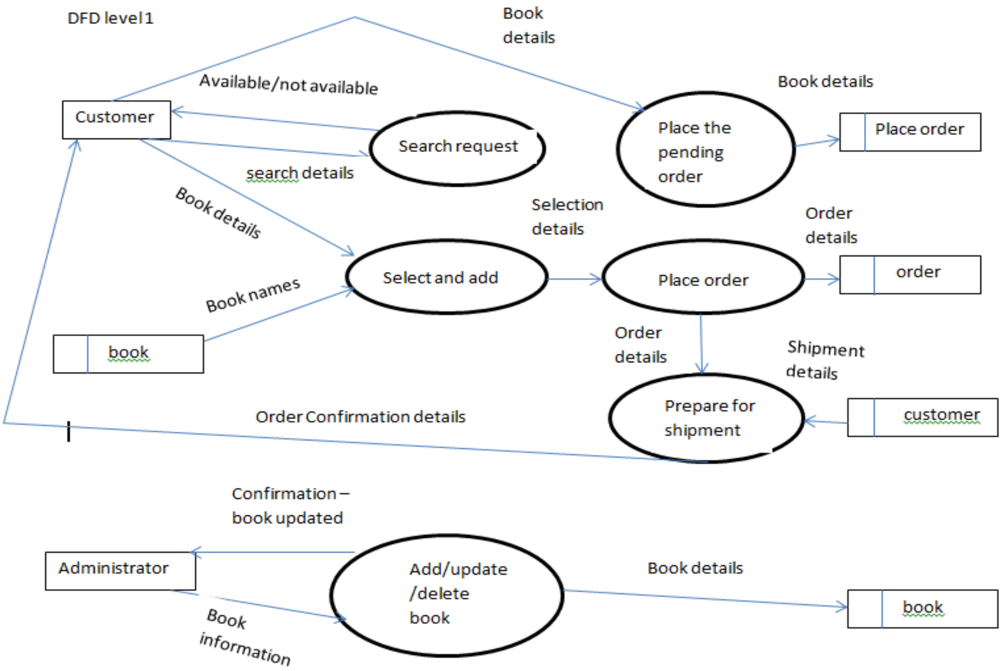
Data flow diagram as name suggests will give representation for data flow between different processes. It also provides means for data representation and visualization. It contains various processes and data as well as external users which uses the data flowing in the system. Processes will operate on data to give out outbound values which will be forwarded to another process. Once all the processing is finished this data will be stored into the database or repository.

Data Flow diagram



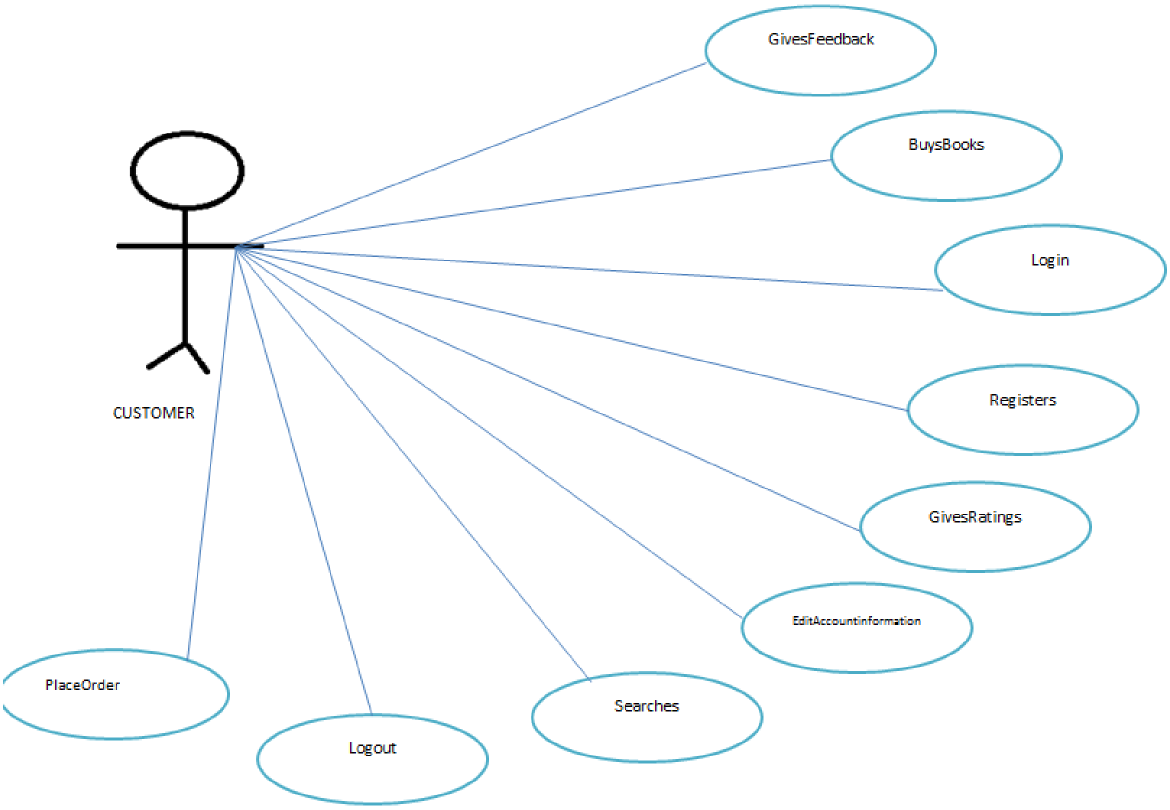
DFD Level 0

DFD level1



2.3 Use-case diagram

Use case diagrams are behavioral diagrams used to represent functionality of system. The external entity associated with the system is represented as an actor. Various tasks and goals actor is responsible to perform shown by the use-cases in diagram. Use case diagram shows dependencies between actors and use-cases as well as the tasks that are assigned to actor within the system boundary.



Use-case diagram 1

Use Case (Actor → customer):

1. Register-Customer registers on the website by providing his/her details.

2. Login-The Customer login with a Username and Password.

3. Searches-The Customer searches the books on the website.

4. BuysBooks-The Customer buys books from the website.

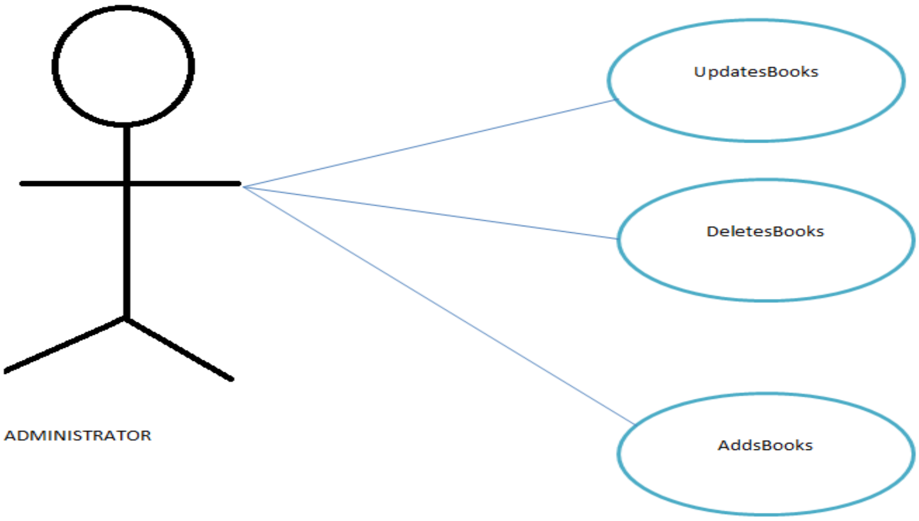
5. PlaceOrder-The Customer places the order if a particular book is out of stock.

6. EditAccountInformation-The Customer can edit his/her details provided to the Website.

7. GivesRating-The Customer can give ratings to a book on the website on the scale of 1 to 5

8. Feedback-The Customer can give a feedback to the website.

9. Logout-Customer logs out of the website by clicking the logout button and the website will flush all the session information.



Use-case diagram 2

Use Case (Actor → Administrator):

1. AddsBooks-The Administrator can add thee new available books in the Books table.

2. Deletes-It can delete the entries in the book database.

3. Updates-The entries in the book database can be updated by the Administrator

3. Detail design and implementation

3.1 Most popular books

We design a simple algorithm to select most popular books from database and show them on the main page in the popular book section. This most popular book selection is based on the rating information given by our customers, involving the rating table and book table. This algorithm works as below:

Step 1. Setting a threshold value N, a book_id can only be picked out of the rating table when the rating given to it by customers is above N.

Step 2. Setting another threshold value M, for each book chosen in step one, check its ratings and pick those that has average rating greater than M.

We implement the algorithm in one SQL:

```
SELECT * FROM BOOK B

WHERE B.BOOK_ID IN

(SELECT BOOK_ID FROM RATES R

GROUP BY R.BOOK_ID

HAVING COUNT(DISTINCT COSTOMER_ID) > N AND AVG(RATING) > M);
```

With the query result, we can have the necessary book information and then display them on the web.

3.2 Use Ajax technique

We use Ajax technique in header.php file to improve the usability of add book to cart. When users add book into his cart, he can stay on the same page after adding and to continue browsing other books. The same thing happens when users remove books from cart.

3.3Credit card validation

To avoid malicious behavior on credit card and alert users about which part their credit card information is incorrect. We add a credit card validation feature. We collect credit card validation algorithm from [2] and implement it for six types of credit card including most used ones such as Visa, Master and Discovery. When users type in incorrect credit card, error message about which part goes wrong, for example card expiration, will be displayed on the same page. So users can have a good experience and correct their input.

3.4Best-selling books

We leverage the customer purchase history information stored in orders table to implement the best-selling books feature. Obviously, best-selling books are those sold best. Plus each order_id in

Orders table relates just on book, then by a simple group by operation on book_id we can find the books sold with the most accumulative total copies number. After retrieving these book_ids, we put them into a PHP container and use a loop to retrieve book information iteratively. The SQL we use to fetch the book_ids are like following:

```
SELECT BOOK_ID

FROM ORDERS

GROUP BY BOOK_ID

ORDER BY SUM(TOTAL_COPIES) DESC LIMIT 5;
```

Here, the constant 5 means that we just want to show five best-selling books on the webpage. We can change it into any value with the growth of our data.

4. References:

[1] http://www.wikipedia.org/

[2] http://www.beachnet.com/~hstiles/cardtype.html

[3] Database Management Systems, Third Edition by Ramakrishnan and Gehrke

[4] http://www.webdesignerwall.com/demo/css3-search-form.html

[5] http://www.lipsum.com/

[6] http://woork.blogspot.com

[7] http://css-tricks.com/

[8] http://www.myjqueryplugins.com/jRating

Appendix - Relational Model

Relational model is a logical data model which is derived from E-R diagram. Purpose of the relational model is to represent various kinds of data as well as their values and attribute. Relational model represents data in more specific form by representing their default values and constraints on it. This data model is then converted into database. In the relational model, the variables along with their data type and default values or constraints if any is collectively called as relational schema.

Table details

ADMIN: admin information stored here permanently.

Admin-id (int)	Admin-name (varchar)	Password(varchar)	Address (varchar)	Email-address (varchar)

Addbook: when an user adds a book into the cart, one more record will be added into the table.

Customer_id(int)	Book_id(int)	Order_id(int)	Quantity(int)

AUTHOR: author information stored here permanently.

Author_id (int)	Name (varchar)	Gender (varchar)	Address (varchar)	Homepage (varchar)	Email (varchar)

BELONGS: store a book and its category information.

Book_id(int)	Name(varchar)

BOOK: book information stored here permanently.

Book-id (int)	BookName(Varchar)	Hardbound(int)	Paperback(int)	e-book (int)	Price (double)	Publisher (varchar)	Edition (int)	Date-of-publishing (date)	Discount (int)	ISBN (varchar)

CATEGORTY: book category information stored here permanently.

Category_id(int)	Name(varchar)

CUSTOMER: customer information stored here permanently.

Customer-id (int)	First name (varchar)	Last name (varchar)	Gender (varchar)	Password (varchar)	Address(varchar)	Email-ID (varchar)

FEEDBACK: feedback information stored here permanently.

Feed-id (int)	Customer-id (int)	FeedbackType(varchar)	Feedback-content (varchar)

HAS: book-author relation table.

Book_id(int)	Author_id(int)

ORDERS: book order information stored here permanently.

Order-id (int)	Book_id (int)	Customer_id (int)	Shippment-address (varchar)	Shipping-data (date)	Total_copies (int)	Total_price (int)	Sold (bit)

PLACE_ORDER: customer's place-order information for unavailable books stored here permanently.

Book_id	Customer_id	ETA

RATES: book ratings from different customers stored here permanently.

Book_id (int)	Customer_id (int)	Rating (int)