**ONLINE BOOK STORE WITH GUI AND SEARCH ENGINE**

**1. Abstract**

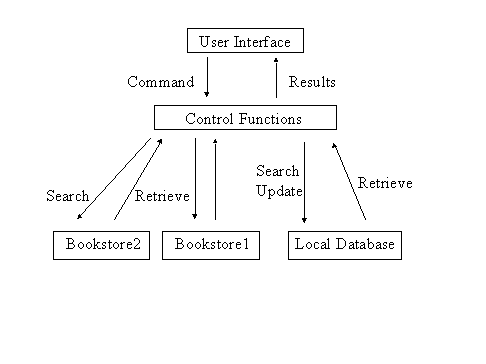
 The goal of this master's project is to design an online bookstore named TechBooks.com that mainly sells computer and technical books. The book inventories are stored in Oracle database in UB. Customers can access the bookstore web site through the World Wide Web. Customers will be able to search the database to find the books they want, check the availability, and place the order to buy the book using their credit cards.

This bookstore also provides a software bridge to two real commercial online bookstores: Bookpools.com and Fatbrain.com. This bridge allow customer to search the inventory of these real bookstores, and display the searching results such as the title, the price and availability of the book.

**2. Design and Methodology**

In this project, several technical approaches are used:

* HTML language for WebPages design and implementation.
* MySql for relational database system.
* Java JDBC / ORM / JPA technology for database connection.
* SQL, structured Query language is used to access and manipulate database.
* Spring Boot / RestPAI for client-server communication.
* Spring MVC Session tracking to implement the Shopping Cart
* Tomcat Webserver



This project designs an online bookstore provides a web-based interface. It is the graphical user interface. It have a form for user to input query information to search the database. The users interface then pass the input to the control function, which implemented in Java. The control function is designed to process the input from the users interface, generate the searching query and then gets data from the database and returns to the users interface. The control functions also have bridges to search some external real online bookstores. It can generate the query based on the user input and pass them to the commercial bookstore. The bridge can also handle the response from the commercial bookstore and parse the useful information and display to the user in HTML forms.

**3. Implementation**

**I. Database overview:**

The online bookstore database sit on the Oracle is made of 4 tables.

**Fbooks:**

Define the inventory of books. It has the fields of: ISBN(primary key), Title, Author, ImageType, Price, Publisher and NumberInstock.

**Fcutomers:**

It saves all the information of the customers. It has the fields of: CusomterID(primary key), User, Passwd, Fname, Lname, Address, City, State, ZipCode, Phone, Fax and Email.

**Forders:**

It has the order information. The fields are: OrderID (primary key), CustomerID, TotalPrice, OrderDate, CardName, CreditCard, CreditType and ExpDate.

**Forderdetail:**

It saves the information of every book on a multiple book order. The fields are: OrderID, ISBN, Quantity and ShipDate. Primary key is the combination of OrderID and ISBN.

**II. Java class file overview**

This project consists of a total of 13 java files.

**BookDetails.java**

Define the object for an individual book. Every book has ISBN number, the Title and Price.

**ShoppingCartItem.java**

Every shoppingCartItem holds an object, which is a book and the quantity of the book.

**ShoppingCart.java**

ShoppingCart is implemented as a hashtable which can hold a book in ShoppingCartItem and number of different books in the hashtable. Add method is to add one ShoppingCartItem into the hashtable. Remove methold remove one from the hashtable.

**Cashier.java**

Cashier class hold an object of shopping Cart and has the method to calculate the total price of all books in the shopping Cart plus the tax(if any).

**BookpoolSearch.java**

This class worked as a bridge between the client's query and the online bookseller Bookpool.com. This can handle cilent's search by title and by author. After getting client input either by title or author, a query in the format of URL string is generated and sent to bookpool.com using java's standard URL class. Bookpool.com will respond to this query by returning a long string of html source code containing books of their database matching the query provided by the client. Boolpool.com returns 25 books in a single page, so the goal of this class is to parse all the useful strings from this long html source code which including the books ISBN, title, price and availability. After getting all these strings, regenerate another html code including all the book infomation and display on client's screen using Java Servlet.

**FatbrainSearch.java**

Another bridge class works similar to the BookpoolSearch.java. It will connect to the online bookseller Fabrain.com. This can handle the user search query by title, author, subject and publisher.

**BookSearch.java**

This class handle client's query to search the local Oracle database by Title, Author, ISBN and Publisher using Java JDBC. And using Java Servlet to generate html source code to display on client's screen. ShoppingCart was implemented by the Servlet Session Tracking technology. It takes the input from the bookstore main page, generates the searching query, searches the database and pastes the book found to user's screen. User can add any displayed books into the shopping cart by pressing the 'add to cart' icon.

**ISBNSearch.java**

This class handles client's query to search by ISBN and display detailed book information for the user. This include the full title of the book, the image of the book and also the author, publisher and numbers in stock. The image is handle in a simple way. All the image files are saved in the server's /images directory. And the image names are their ISBN number follow by the suffix of either .gif or .jpg. In the oracle database, the ISBN and the ImageType will determine the full image name. So, there is no need to save the binary image file into oracle database.

**ShowCard.java**

This class will display all the books user add to his shopping cart. Through the Servlet Session Tracking API, session was found by checking the session ID. Books information was extracted from the hashtable of the shoppingCart object. Html code was generated and send to user's screen by the Servlet doGet method.

**Login.java**

Once the user decide to checkout to buy books, user are asked to input his registration information. If the user has account on the bookstore before, he just need to input his userID and password. This class collects the userID and password and then search to see if the info was valid against the Oracle database using Java JDBC. If the info was matched to the record, user was redirect to continue checkout. Otherwise, user was asked to input some personal info to create an account.

**SignIn.java**

This class is to create a new account to the first time customer, the information will be inserted into the Fcustomer table by using Java JDBC.

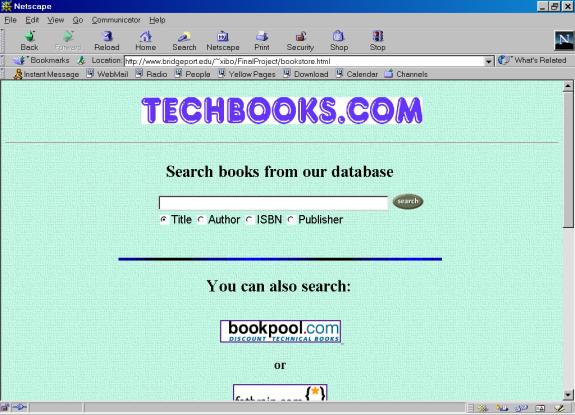
**CheckOut.java**

This class first displays all the contents in the shopping cart, then prompt user to input his credit card information which include the name on card, credit card number, the type of credit card and expiration date. If either of them is empty, let user go back to input again.

**SubmitOrder.java**

This class is called once user has input all his credit card information. An order is generated in the Forders table by assigning a new orderID. Total price and credit card information will be inserted to the table. Also in Forderdetail table, every book in the shopping cart will be an item on the table. Finally, numberinstock value in the Fbooks table has to be updated since the user already buys the book.

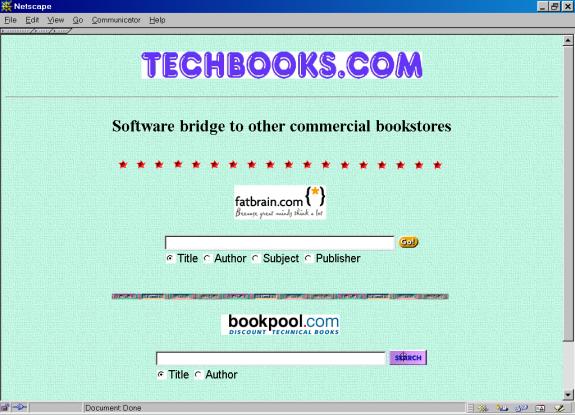
**4. Screen shot**



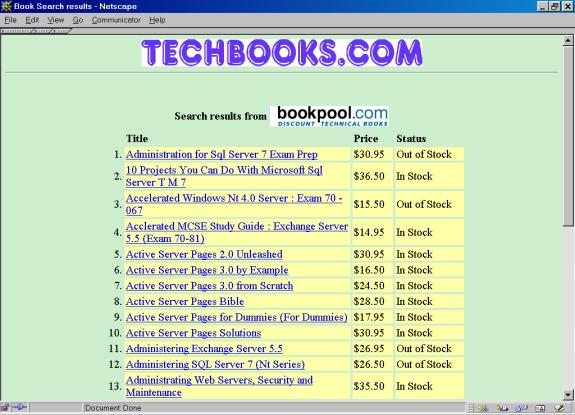
Upper half of the bookstore home



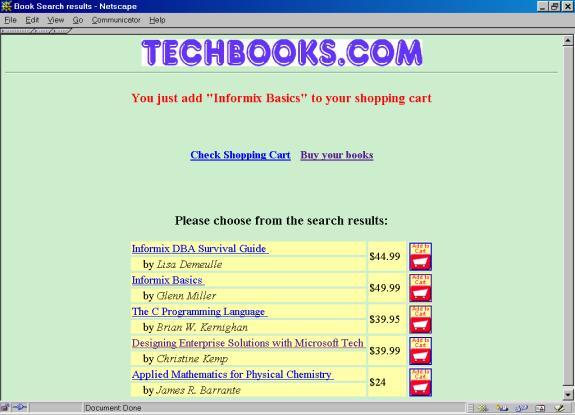
Bottom half of bookstore home



Bridge interface to fatbrain.com and bookpool.com



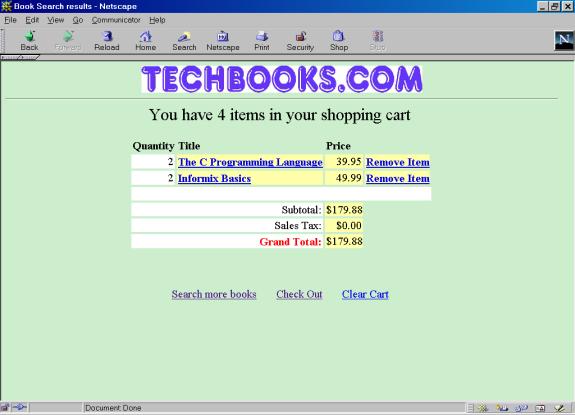
Searching results from bookpool.com



Search results from own database and add one book to shopping cart



Display the book details including the image



Display of the shopping cart



Customer Login Screen



Order preview and input credit card information



Display after successfully process the order



Customer sign in page