

## Online Bookstore System

Over the past several years, ecommerce has transformed how people buy and sell online. In fact, online shopping has become so popular that many vendors sell only online with no physical location. An online bookstore can reach customers anywhere in the world, even with no physical location.

### Problem Statement:

The online Bookstore has few number of employees and keeps information about thousands of books and registered customers. The bookstore will also manipulate and store information about few number of suppliers and shipment agencies.

The system should enable the business manager and the system administrators to have real time statistics of sales and book inventory by producing end of day sales reports and end of day low-inventory notices automatically. It will enable users to view the real time inventory and exact book information.

The system will also keep information about its registered customers. Hence, the system should provide a registration facility to its users. As an online system, the bookstore system, is available to all web users. Web users should be able to browse available books and to search for books, but only registered users (customers) can place orders by adding books to their shopping cart and checking out.

The system should allow users to pay their orders online. ~~After payment, the shipping employee will be notified of a shipment order. The shipment employee will select a shipment agency and change the status of the order. Upon delivering the order to the customer, the shipping employee will be notified, via an email, and accordingly he/she will change order status.~~

### Development Process:

We will the develop the software according to a set of stable and predefined requirements. Our process model will be a hybrid model that combines both the waterfall and agile methodologies. We will apply Waterfall for the overall project planning; setting phases and pre-planned deliverables, on the other hand, we will take advantage of the flexibility of Agile by injecting Agile practices within our planned phases to produce an early working version of the system. For example, we will schedule coding as set of sprints -user stories, we will also benefit from Agile by incorporating testing in the development; to ensure that a quality product will be produced.

Teams can work “Agile” while making sure the processes adhere to the plans created in the initial phase, based on our “Waterfall” planning. Development phases that can be handled simultaneously; shouldn’t have to wait for each other. For example, two team members might be working on UI design while other members can start building database tables.

Prototypes will be used to help designing the UI to gain feedbacks and more customer involvement which will reduce the risk of project failure and/or customer dissatisfaction.

Once the system is completely implemented, it will be tested according to the developed software components by applying *adequate* subsystem testing, integration testing, and acceptance testing (through final project demonstration).

Deadlines for all project deliverables will be announced on the course announcement page on eLC. Team leaders should submit deliverables to the corresponding assignment folder on eLC.

Final project demonstration will be around 04/20/2020.

## High Level System Requirements:

1. The system must allow the system administrator to enter book information. For each book, the bookstore must record its ISBN, category, authors' names, title, cover picture, edition, publisher, publication year, quantity-in-stock, minimum threshold, buying price, and selling price.
2. The system must allow the system administrator to delete or update book information. The admin must also be able to add new employees and authorize them, in some cases he might suspend a member account.
3. The system must allow the system administrator to add promotions to the systems. The system must automatically send email promotions, offered by the system administrator, to all registered users who have subscribed for promotions. Each promotion has a promo-code, percentage, start date and expiration date.
4. The system must provide a browse and search catalog facility, to all users, that shows the list of available books sorted by title. The list should include book title, author, book cover picture, book rating and the price. The system must provide an interface to search for and view books that are available in the store categorized by subject, title, ISBN, or author.
5. The system must allow web users to register for the system. To register, users should provide their information (name, phone number, email address, and password. Optionally users can store their shipping address info (street, city, state and zip code) and payment info (card type, number, expiration date).
6. User information should be stored permanently in the database. Passwords and card numbers should be encrypted.
7. Upon registration the user status should be saved as Inactive until confirming their email address.
8. The user's registration is verified by entering a verification code that is generated by the system and sent to the user's email address upon completing the registration form and after confirmation, the customer will be assigned a unique account ID and the user status will be changed to active.

Registered users (Customers) can login using their account ID or email address and password. Each email address will be linked to one account ID and each account must belong to exact one customer.

9. Only registered users can place orders.
10. Registered users can subscribe/unsubscribe for promotions and latest news offered by the bookstore administrator.
11. Registered user must be able to view and modify their user profile at any time.
12. The system must provide a secure forget-password facility.

13. The system must provide a shopping cart to each registered user. Each customer ID must be linked to one shopping cart.
14. Registered users should be able to place orders by adding books to their shopping cart. They must be able to view and manipulate their shopping cart at any time before checking out.
15. The system must provide a secured checkout facility. The system must allow customers to use their promotion codes in order to take advantage of current discount promotions.
16. The system must send order confirmation to the customer's email. Order confirmation should include Customer name, Confirmation number, Order ID, Order Date, Shipping information, ordered items and total amount in dollars.
17. Customers must be able to view their order history and they should be able to track their order status.
18. Users should be able to reorder a previously ordered item.
19. For each order, the system must record order information: who made the order (Customer ID), order ID, order time and date, order status, total price, shipping address, payment method, and ordered books.
20. The system must enable managers and administrators to pull predefined reports; namely End of Day sales reports, low-inventory notices, book sales report and publishers' sales report.
21. The system must provide multi-user access, assuring correct concurrent behavior.
22. The system should maintain suitable authorization information and authenticate access for different types of users (Administrator, manager, web users and registered users). User authentication should be implemented by checking user ID and password.
23. The system must have an easy-to-use user interface (UI) with screens designed for each part of the system's functionality and must conform to users' authorization.
24. The user interface must either be accessible from a common Web browser (such as: Internet Explorer, Google Chrome, Mozilla Firefox, and Safari).
25. The system must use a persistent data store (MySQL RDBMS) for all of the relevant data.
26. The system should use accepted standards whenever possible (HTML, CGI, Servlet API, JDBC, ODBC, SQL, etc.). The system must be coded in either Java, Python, or C++, possibly incorporating some smaller parts coded in other languages, such as PHP and JavaScript, if needed.