PROJECT:

ONLINE BOOKSTORE

Group: *Max 5 members;* ***Report Due****: In class on Tue of Week 14;*

Demo Due*: make an appointment in Week 14 (before final exam)*

B&N is a large bookstore and it has decided to go online. Your team has been called in to help B&N design and implement the prototype of its online shop. The owner of B&N has thought extensively about what he wants and offers a concise summary:

Data Specification

The system should manage information about books in the bookstore, inventory, (registered) customers and books they have ordered. It should also store information about customer opinions and book ratings.

Books: An ISBN for each book, title, author(s), publisher, year of publication, number of copies in inventory, price, format (hardcover, softcover), keywords, subject (Sci-Fi, western, history, etc.). You can add additional information if necessary.

Customers: For each registered customer, you need to maintain: his/her full name, login name, password, major credit card number, address, phone number, books he/she has ordered and information regarding the order (e.g., order_date, order_status).

Opinions: Users can provide 'feedback' for a book, as a score (1-10) along with optional short text. Users can also rate other users' feedback as 'useless', 'useful', 'very useful'.

System Functionality

The following set of events and queries should be handled by your system:

1) [5pts] *Registration*: a new user has to provide necessary information; he/she can pick a login-name and a password. The login name should be checked for uniqueness.
2) [5pts] *Ordering*: After registration, a user can order one or more books. A user may order multiple copies of a book, one or more times. (The charging of the credit card and the shipment of the books are outside the scope of this project).
3) [15pts] *User record*: upon user demand, you should print the full record of a user:
   - his/her account information
   - his/her full history of orders (book name, number of copies, date etc.)
   - his/her full history of feedbacks
   - the list of all the feedbacks he/she ranked with respect to usefulness
4) [2pts] *New book*: The store manager records the details of a new book, along with the number of new books that have arrived in the warehouse.
5) [3pts] *Arrival of more copies*: The store manager increases the number of copies in inventory.
6) [2.5pts] *Feedback recordings*: Users can record their feedback for a book. You should record the date, the numerical score (0= terrible, 10= wonderful), and an

optional short text. No changes are allowed; only one feedback per user per book is allowed.

7) [2.5pts] *Usefulness ratings*: Users can assess other uses' feedback, give a numerical score 0, 1, or 2 ('useless', 'useful', 'very useful' respectively). A user is not allowed to rate his/her own feedback.
8) [20pts] *Book Browsing*: Users may search for books, by asking conjunctive queries on the authors, and/or publisher, and/or title, and/or subject. Your system should allow the user to specify that the results are to be sorted a) by year, or b) by the average score of the feedbacks.
9) [5pts] *Useful feedbacks*: For a given book, a user could ask for the top *n* most 'useful' feedbacks. The value of *n* is user-specified (say, 5, or 10). The 'usefulness' of a feedback is its average 'usefulness' score.
10) [10pts] *Book recommendation*: Like most e-commerce websites, when a user orders a copy of book 'A', your system should give a list of other suggested books. Book 'B' is suggested, if there exist a user 'X' that bought both 'A' and 'B'. The suggested books should be sorted on decreasing sales count (i.e., most popular first); count only sales to users like 'X' (i.e. the users who bought both 'A' and 'B').
11) [10pts] *Statistics*: Every month, the store manager wants
    - the list of the *m* most popular books (in terms of copies sold in this month)
    - the list of *m* most popular authors
    - the list of *m* most popular publishers

Deliverables and Grading

1. A report (20%) that contains two parts. The first part should contain the ER diagram for the application and the relational schema (in SQL DDL code). The second part should contain implementation details of your application; sample and representative SQL code of the functions it helps to implement; 2 or 3 representative screen dumps of the Web/mobile app interface.
2. A demonstration of your application (80%). This will be strictly graded based on the functionalities of your application (according to the points assigned above).