Project: development of database-based system with Django

Group: Max 5 members. Please submit your groups online here by October 9th.

Topic selection: If you want create a website on a topic other than the book store described below, please email your proposal to dorien_herremans@sutd.edu.sg **and** jon_wong@sutd.edu.sg by October 14th, together with a brief description of the suggested web application and functionalities to be implemented.

Report Due: Monday of Week 14

Demo Due: Week 14 (before final exam)

Submission: Source code and report through eDimension

Description: Create a web application with a database backend using Django that implements the features described below. The server can run locally on your machine (e.g. through python manage.py runserver, or online). You can use all of Django's libraries, however, you need to create and access your database using **raw sql**, with this interface, so *not* with Django's ORM as you are being graded on your knowledge of SQL.

More info: Will be given in the class of Week 6. Interested students can already take a look at Django tutorials online, such as this one. Be sure to use *raw sql* in your project.

1 Features that should be included in the system

Note: groups who want to create a different type of system (not a book store) are encouraged to do so for extra credits. However, the system needs to link all of the below features to their own topic, thus applying the same database concepts. If you are creating a different system, e.g. an online bidding website or something else, your report should link each of the below points to your system and briefly explain how these database concepts are reflected.

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. Use Django's auth mode and session DB module for this.

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

2 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 code of the functions it helps to implement for each of the above points; 2 or 3 representative screen dumps of the app interface. *If your website is about a different topic than a book store,*

then please explain in the report how each required feature is reflected in your system (applied to a different topic).

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).

3. **Extra credits** will be given to students who implement a novel topic (other than the book store project) or who can demonstrate some best practices in web development:

    - Django's test framework
    - CSS preprocessors like LESS/SASS
    - Javascript UI libraries like jQuery
    - CSS frameworks like Bootstrap/Foundation
    - Git-tracked development workflow. See Jon Wong's tutorial.