CS5530 Spring 2013: Project

1 Overview

In this project, you will design, implement, and document a database system for an electronic videostore (e.g, `http://www.netflix.com`).

2 Application Specification

A description of the major functions and data items follows. Some functions will be provided to the customers and others to the manager of the videostore.

2.1 Basic Data

The system should manage information about videos in the videostore, inventory, (registered) users and videos they have ordered. It should also store information about user opinions and video ratings. While you are working, keep in mind that these items are a minimal set of requirements.

Video Data: An ISBN for each video, title, author(s), year of production, number of copies in inventory, price, format (DVD, Blue-Ray, HD-DVD, VCD), keywords, subject, rating (G, PG, etc). You should add additional information if necessary.

Customer Data: For each registered customer, you need to maintain: his/her full name, login name, password, major credit card number, address, phone number, videos he/she has ordered and information regarding the order.

Opinions: Users can provide 'feedback' for a video, as a score (1-10) along with optional short text. Users can also rate other users' feedback as 'useless', 'useful', 'very useful'; finally, they are allowed to declare other users as 'trusted' or 'not-trusted'.

2.2 Functionalities of the system

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

1) [3pts] Registration: Registration: a new user has to provide the appropriate 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 video tapes. The total amount of its order is reported to him/her. A user may order multiple copies of a movie, one or more times. (The charging of the credit card and the shipping of the tapes are outside the scope of this project).

3) [15pts] User record: upon user demand, we should print the full record of a user:

- all his/her personal data
- the full history of sales (movie name, number of copies, date)
- his/her full history of feedbacks
- the list of all the feedbacks he/she ranked with respect to usefulness
- the logins of the 'trusted' and 'not-trusted' users, along with the corresponding dates.

4) [2pts] New movie: The store manager records the details of a new movie, along with the number of tapes that have arrived in the warehouse.

5) [3pts] Arrival of more tapes: The store manager increases the appropriate counts.

6) [3pts] Feedback recordings: Users can record their feedback for a movie. We should record the date, the numerical score (0= terrible, 10= masterpiece), and an optional short text. No changes are allowed; only one feedback per user per movie is allowed.

7) [3pts] Usefulness ratings: Users can assess a feedback record, giving it a numerical score 0,1, or 2 ('useless', 'useful', 'very useful' respectively). A user should not be allowed to provide a usefulness-rating for his/her own feedbacks.

8) [3pts] Trust recordings: A user may declare zero or more other users as 'trusted' or 'not-trusted'.

9) [20pts] Movie Browsing: Users may search for movies, by asking conjunctive queries on the actors, and/or director, and/or title-words, and/or rating (G, PG, etc). Your system should allow the user to specify that the results are to be sorted (a) by year, or (b) by the average numerical score of the feedbacks, or (c) by the average numerical score of the trusted user feedbacks.

10) [5pts] Useful feedbacks: For a given movie, 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.

11) [8pts] Buying suggestions: Like 'netflix.com', when a user orders a tape of movie 'A', your system should give a list of other suggested movies. Movie 'B' is suggested, if there exist a user 'X' that bought both 'A' and 'B'. The suggested movies should be sorted on decreasing sales count (i.e., most 'popular' first); count only sales to users like 'X'.

12) [15pts] 'Two degrees of separation': Given two actor names, determine their 'degree of separation', defined as follows: Two actors 'A' and 'B' are 1-degree away if they played in at least one movie together; they are 2-degrees away if there exists an actor 'C' who is 1-degree away from each of 'A' and 'B'; and so on.

13) [10pts] Statistics: Every semester, the store manager wants

- the list of the m (say $m = 10$) most popular movies (in terms of tapes sold in this semester),
- the list of m most popular directors and
- the list of m most popular actors

14) [5pts] User awards: At random points of time, the store manager also wants to give awards to the 'best' users; thus, the manager needs to know:

- the top m most 'trusted' users (the trust score of a user is the count of users 'trusting' him/her, minus the count of users 'not-trusting' him/her)
- the top m most 'useful' users (the usefulness score of a user is the average 'usefulness' of all of his/her feedbacks combined)

3 Phase 1: Data Modeling with ER and the Conversion to the Relational Model

Initially, you must design and create a database that organizes and stores data about the videostore system. This includes designing the ER diagram, and then converting your designs to relational tables/schemas to store this data, and designing rules (when necessary) for how these tables relate to each other so that the data they store can be combined in meaningful ways. As part of the report for this phase, you will turn in an E-R diagram of your database design, and the translation to the relational schema. Also, you have to provide all the constraints and the SQL code that you used to enforce them (CHECK or ASSERTION clauses).

Due Date: Monday, February 25th in class.

4 Phase 2: A Workstation/Desktop System with Java

In this phase, you should write the SQL queries and stand-alone Java code that implements a (simple) user interface to your database and executes the queries. Your system should provide all the functionalities as specified in Section 2.2. Sample codes will be provided to you at the beginning of this phase.

A command-line based interface will be accepted. A GUI is not necessary, but you are welcome to do so (no bonus points for a GUI though if implemented).

Due Data: Monday March 18th 10am, using the Handin system. Your submission in this phase will be handled by the Handin system. In particular, you can submit your zipped project folder as a single zip file by either of the following options:

1) Log in using your CADE account at: `https://cgi.eng.utah.edu/webhandin` (the class name is cs5530); then select phase 2 .

or 2) use the command line on the linux machines (from the CADE lab): handin cs5530 phase2 /path/to/file.

Your implementation in this phase will be used as APIs to facilitate your implementation in the next phase for the web interface.

5 Phase 3: Web-based System with JSP

Finally, you will create a web portal to your database system. Users will connect to the web site and will be able to execute various functionalities from the web interface. This will be achieved via JSP+Tomcat. Note that you are not allowed to use servlet, or any other programming frameworks. Only JSP is allowed (since it allows you to call your Java classes in Phase 2 in an extremely simple fashion). You will be provided sample JSP codes to start with, as well as sample initial set up to your web folder under the Tomcat server. Details will be provided on the class website when this phase begins.

Due Data: Monday April 8th 10am, using the Handin system. You will stop editing any files on your account on our webserver when this phase is due. You also need to submit a single zip file (for your entire web folder) through the Handin system; again, details will be provided in the course website when it is close to the due date of this phase.

6 Grading of the Project

Phase 1 takes 30% of the project, and Phase 2 and Phase 3 will each take 35% of the project. Phase 1 will be graded in the same way as how we have graded your HW1 and quiz 1. Phase 2 and Phase 3 will be strictly graded based on the points assigned to each functionality as shown above in Section 2.2.