CSE480: Database Systems, Spring 2003
Project Two

A Web-based Student Enrollment System

Design Document Due: April 14 in class
Implementation Due: April 21, 11:59 p.m.
(This project will be done by the same two person student group as that in project one)

1 Description:

In this project you will design and implement a Web-based Student Enrollment Information Management System to allow students and the administrators to browse, enroll and manage enrollment data. The goal of this project is to give you hands-on-experience, using a real-world application, of various important database concepts and techniques. You must read the description of the requirements given below, carefully. Your design must incorporate all the requirements given here. However, if you need to make any assumptions for detailing some of the requirements, you must include them in the detailed requirements to be submitted with the design specifications.

Some of the major database concepts to be applied in this project are:

1. database modeling
2. Automatic ID generation for key attributes
3. Ad-hoc queries
4. Views
5. PL/SQL, procedure and packages
6. Transactions, save points, rollback and isolation levels
7. Statement and row-level triggers

2 Project Requirements:

There are two types of users in this system, Student user and Administrative user. Notation [%] used below sums up to [50%] for all sql queries, views, etc. in the design document.
As a Student user, you will be able to do the following:

1. Student information: [10%]
   (a) Login/Logout and change password (project one).
   (b) Student Personal Information:
       Display student’s own information, such as name, student ID, age, address, major and student type (Undergraduate, graduate, probation).
   (c) Student Course Information:
       Display all the courses the student has taken or currently taking. Course information includes: course title, description, prerequisites, number of credits, instructor, textbook, TA, semester, grade. Display total credits completed and the GPA. GPA must be a **derived attribute** and should be implemented by triggers. GPA is defined as:
       
       $$\text{SUM} \text{ of all} \ (\text{course grade} \times \text{course credit}) \ \text{DIVIDED by total credits.}$$
   (d) General Information: Display general information about courses, departments, instructors and the programs.
   (e) EER-diagram developed in the labs should be extended as necessary, and used for data for this project. Your design must reflect any clarification that is given through Question/answer.

2. Online Course Enrollment: [20%]
   Keep in mind that multiple students may request to enroll in the same course at the same time. You will implement them using database transactions. You must use at least one long duration transaction and a rollback in your transaction design. Long duration transactions must be efficiently implemented so that locking, if any, is done appropriately. You must indicate the isolation level(s) you are using in your design document with a brief justification.

   For on-line course enrollment you need to consider the following:
   (a) There should be a deadline for course enrollment.
   (b) A student will be able to find available courses based on semester, department and partial courseno, such as ‘cse1*’. Display the course information as well as the maximum number of seats and the number of currently available seats.
   (c) You may select one or more courses to enroll in one transaction. The enrollment request may fail if any of the following happens:
       i. You did not take all the prerequisite courses
       ii. No available seats
iii. Not a student in the same department as the department offering the course.

The system will display reports indicating if certain course enrollments have failed and the reasons for failure.

**As an Administrator, you will be able to do the following:** [20%]

1. [0%] Login/Logout and change password.

2. [5%] Add a new student. The system should be able to automatically generate a student ID. The student ID is in the format of XX12345, where XX is your initial, 12345 represents the sequence number. The new student sequence number should be the current maximum sequence number in your database plus one.

3. [10%] Search student information based on one or any combination of name, student id, courseNo, major and student type. You should be able to search student name and student Id based on pattern match. You may select a student from the search result to update that student’s information.

4. [5%] Insert student grade information by giving studentId, course sequence number and grade. As a result of this, if an undergraduate student’s GPA goes below 1.5, the system will automatically make the student a probational student. If a probational student’s GPA becomes equal to or above 1.5, the student should become a regular student again. For graduate students the threshold should be 2.5. The start and end date for probation should be maintained in the database.

You may use the sample database given in the labs, but will need to include additional data to completely test your system. For demo we may use a totally different data base. TA may answer reasonable questions about the project and help you understand and work on the project during his office hours. Please start working on the project based on the concepts already learned.

**User Interface Design:**

All user interfaces should be point/click type where appropriate. Refer to lab 9 for implementation.

**A student question/answer file will be posted on the course web page**

**3 Submission:**

The submission of this project has two parts, design document and implementation. **Note that the design must be presented in significant details**
and will require a great deal of effort. The design document should include sql statements, triggers, transactions, etc. The design document and the implementation carry equal weight (50% each).

- The design document is due on April 14. In your design document, you should present the following:

- 10% Detailed Project Requirements including window snap shots and description of window functions.
  10% All User Interface Design.
  10% EER Diagram.
  10% Database Schema (all relation schemas, constraints, etc.).
  5% Foreign Key Reference Diagram.
  5% Other design issues not included above.
  50% All SQL queries, views, transactions, triggers and PL/SQL functions and procedures you are going to use. You MUST include at least one view, one function or procedure, one long duration transaction and a statement and a row level trigger in your design.

- **SUBMISSION OF EXECUTABLE CODES:**

  The implementation part is due on April 21 by 11:59 p.m.. You must send an email to the TA by April 21 (11:59 p.m.) with a link to your login page. TA will check the time stamp of your project files for submission deadline. Attach all your source codes, a student user name/password and the administrator user name/password that you have created for the system, with the email. Include some data in your system for the TA to test your program.

  The grade distribution for this part of the project is based on the following criteria:

  1. Correctly run all project functions – 75%
  2. Interface design – 20%
  3. Coding practice – 5%

**Grade distribution between the two projects:**

Project 1: 20%
Project 2: 80%