CS3425 Final Project

Fall 2018

100 points, 20% of the overall grade

Report Due date: 11:59pm, Tuesday May 1st.

Besides the report, you also need to schedule a 10-minute meeting with our TA on Monday or Tuesday during Exam week. During the meeting, the TA will ask you to run selected functions, which will be 20% of your project grade.

Project description:

In this project, you will design and implement a simplified version of online registration system for Ice Skating Camps. The system will be used by the skating camp coordinator, coaches and campers. **If there is any ambiguous requirement to you, write down your assumptions in your project report.**

a) The organizer creates the camp info: camp name, start date, end date, and location. We also need to store a brochure, which is a pdf file about the camp. The organizer will decide when the registration begins and when it ends. The database only need to support one camp for this project.

b) The organizer decides the registration fee and ice fees per ice session

c) The organizer creates ice sessions. Each session has a date, start_time, end_time and comments. Each session is general 45 minutes.

d) The organizer creates teaching slots all sessions. Each session may have multiple slots. Each slot is generally 15 minutes long. We need to record start time, and end time for each slot.

e) The organizer creates accounts for coaches. The system tracks coach's name, email, mail address, phone number.

f) The coaches decide how much they charge per 15 minute slot, and pick which time slots they offer lessons.

g) The campers create their own accounts. The system keeps track of camper's name, email address, mail address, phone number and emergence contact.

h) The campers register a camp during the registration period, and then pick the lesson time from coaches. The camper can take lessons with multiple coaches. The lessons are one to one, which means no two campers are allowed to choose the same time slot with the same coach.

i) The system creates the bill and schedule for each camper.

j) The system creates the lesson schedule with student name for each coach.

Your Tasks:

1. E-R Model and Relational Schema
   1) Construct an E-R diagram representing the conceptual design of the database. 2) Be sure to identify primary keys, relationship, cardinalities, etc.
   3) Create relational schema from your E-R design, and refine it based on the principles of relational design.

2. SQL Schema

   Create the tables in Mysql with PK, FK, unique, not null, etc. constraints as appropriate.

3. You shall design and implement the following functions for the instructor and the students.

The interfaces for organizer's tasks can be implemented inside database as Stored Procedure or function. The interface for students must be accessible through Web Browser.

1) **Organizer**
   i. Create camps: name, date, registration fee, ice fee per session, etc.
   ii. Create sessions and slots
   iii. Create coaches with initial password, information and lesson fee

2) **Coaches**
   i. Login to the system
   ii. Pick slots to offer lesson
   iii. Display the current lesson schedule
   iv. Display the charges to each student

3) **Campers**
   i. Register and login to the system
   ii. Choose a camp, choose coaches and lesson slots.
   iii. Display the lesson schedule
   iv. Display the bill (registration fee, ice fee, lesson fee per coach)

Let me repeat: the interfaces for Organizer's tasks can be implemented inside database as Stored Procedure or function. The interface for Coaches and students must be accessible through Web Browser.

What to turn in to canvas:

To speed up the grading, please submit two files.

1. **FinalReport.pdf.** Put all the following contents into a pdf file **FinalReport.pdf**
   1) E-R diagram and relational schema, plus any explanatory notes if you will.

   2) SQL schema script. That is the create table statement.

   3) Do the following three things for each tasks(total 5) below as user cs3425gr, not your account, which means you need to grant permission to cs3425gr appropriately.

   A.The detailed instruction of how to perform the following task
   B. The output of each task (if GUI, please attach screen shot),
   C. Show the related data inside the database before and after the operation to proof that it works correctly
   Please make sure you give clear instruction and include the exact input and output for each step because TA will do the test following your example here.

   i. Create two coach user **<u>Alice and Joe</u>**

   ii. Create a new camp **Summer18**

   iii. Create 2 days 12 sessions, 36 slots

iv. Have **Alice** login to the system with password

v. Take the Summer18 Camp, show the ice slots and decide if he/she will offer lesson in that slot.

vi. Do the same for coach Joe

vii. Have two campers Mary and Olivia to register and choose 5 lessons from Alice and Joe

viii. Show Mary, Olivia's schedule and bills

ix. Show Alice and Joe's lesson schedule and charges to each student

2. **Source Code files (source.zip or source.tar)**

Grading:

ER design and relation schema: 5 points

SQL schema, including constraints and indices: 5 points

User interfaces in the report: 50 points

Coordinator's functions: 10 points
3.3 points for each function

Coach's function: 20 points
Login: 5p
Lesson Slot: 5p
Teaching schedule: 5p
Charges: 5p

Student's functions: 20 points
login: 5 points
choose coach and time slots: 5 points
display schedule: 5
display bill: 5

(Important note! You need to show that the above functions actually work correctly. You could display the related database data before and after the operation, then explain what had happened. You will only get 20% for each problem if there is no proof that the function works as expected)

Use transaction correctly: 5 points

- Update the data in one transaction

Encrypted password: 5points

- No clear text in database, in network transmission

SQL injection: 5 points

- Consider the SQL injection attack wherever user input is typed

Programming style: 5points

- Clear code and concise, adequate comment

Selected user interfaces checking by TA: 20 points

Extra points (up 10 points)

1. You may get to up 5 points for your user interface design. A nice interface includes the following
   a. Self explained – user does not need to be trained for long time to use it
   b. Efficient - user does not need to enter much information in different places
   c. User friendly - nice look, it is enjoyable for user to use it

2. You may earn up to 5 points for using OAuth 2.0 to authenticate the users of your web application.

   In the specification, above, we have two functions:
   - Organizer creates coaches and set the initial password for coach user.
   - Coach login with their password.

   To do so, you will create a coach account table to record their account name and password. Then you will need to ask coach's account and password to allow them login.

   But with Google OAuth2.0 API, we will delegate the user authentication to Google's OAuth Server. The OAuth 2.0 server will authenticate the user with their email address and ISO password. We can also specify the redirected URL once user has been authenticated.

   What this means: we may still list valid coach account (gmail address) in our table, but we don't need to save passwords in the table, and we don't handle login. Instead we let the OAuth 2.0 server do it, and tell the OAuth server where to go (redirect url) after user is authenticated.

   See the instructions here: https://developers.google.com/api-client-library/php/auth/web-app