Project. Build an Information System for Course Assignments

1 Objectives, important dates and deliverables

You are expected to develop an information system to 1) record which instructor teaches which course(s) in a given semester with what resources, and 2) enable users to answer questions about the instructors and courses etc.

Important milestones and deliverables

1. **Every sunday night from Apr 5**. Weekly report. The report includes things done in the previous week: meetings, things discussed and done, questions, plan for next week etc. Submit to `weeklyreport` folder on Piazza with text/pdf file with name: team last names + "-wk"#, e.g., smithParkerZhang-wk1.text. You can certainly attach files to your post.

2. **Apr 7.** Understanding of the project and produce the ER diagram. Submit a PDF file including the drawing and explanation of the ER diagram and the design of schema of the tables to the Piazza folder *project*. File name: team last names+"-ER", e.g., ParkerSmiZhang-ER.pdf

3. **Apr 21.** Intermediate report and prototype system (source code and databases). Submit a ZIP file, with team last names + "-mid", e.g., smithParkerZhang-mid.zip, including the report in pdf and the prototype system to the Piazza folder *project*. Face to face interview.

4. **May 5.** Final report and system (source code). Submit a ZIP file including the report in pdf and the system to the Piazza folder *project*. Face to face interview.

2 Preliminaries and background

A course has a course code (e.g., CS4354), and a title (e.g., Concepts of Database Systems for CS4354). A course is either a required course or an elective course. In a given semester, one or more lecture sections of a course may be offered. A lecture section is either a regular face to face lecture or a distance lecture. A regular lecture section is numbered from `001`. A distance lecture is numbered from `D01` where `D` denotes distance. In the university information system, there is a unique five-digit number associated with each lecture section of an offered

course. For example, for CS1411, we offered 3 lecture sections (numbered from 001 to 003). The first section has a number 14319 which is unique with respect of all lecture sections of all courses. Note that this number may change every semester (e.g., in anothe semester, the number for section 001 of CS1411 may be 26453).

An *instructor* of a lecture section of a course is a professor, a graduate part time instructor (GPTI), or a full time instructor (FTI). A professor is either *untenured* or *tenured*.

For the courses of CS1411, CS1412, and CS2413, we also have lab sections which starts from 501 (denoting the first lab section). Each lab section has a unique 5-digit number too, similar to a lecture section.

For each section of a course in a semester, we have the enrollment (e.g., 20 students in the class), an upper limit of seats (not more than 50 seats) for it.

In any semester, a number of courses are offered. Some courses will have only one lecture section while some others may have more than one lecture sections. Some we may have more than one lecture sections and lab sections. An instructor, a time slot and weekdays (either MWF or TTH), and a classroom are assigned to a section of a course. A classroom is identified by its building name and room number (at most 4-digit).

A course is *new* for a professor if this professor never taught it before.

For each instructor and a lecture section assigned to the instructor, there may be a TA assigned to help this instructor on this section for a certain number of hours per week (smaller than 20). For example, a TA is assigned to help the instructor of the 001 section of CS1411 for 7 hours per week.

See the file `sampleCourseAssignment.xlsx` (downloadable from `http://redwood.cs.ttu.edu/~yzhang/cs4354/program/`) for sample information on courses offered in all semesters (spring, summer I, summer II, and Fall) of 2013 at CS department of TTU and how other resources are assigned to the courses. From the blog of Apr 2 in https://cs4354volunteerteam.w you can find ways to import sample data to mySQL (thanks to Albor writing up this note).

For courses and personnel, we restrict to computer science department only in this project. For course names and their codes, use the latest TTU catalog. For building names and room numbers, follow the convention used in the sample course assignment file.

3 Problem

The problem is to build a web based application using `PHP` and `mySQL` so that with your application:

1. A department faculty/staff can login (**optional**) and update the following information.

(a) When there is a new professor / FTI / GPTI joining the department, add the information of the new people into the system. The information including the joining date (which semester of which year, e.g., spring 2013), tenured or untenured, and title (e.g., assistant/associate/full professor). However for FTI/GPTI, we don't have title or tenure information.

(b) For each semester and every section of an offered course, input who is the instructor, where and when the section is taught, the capacity limit of the section and the enrollment of this class. For example, for CS4354 section 001, time (10am to 10:50am) days (MWF), room (204), building (ENGCTR), capacity (60), and enrollment (35). Note that for a section of a course, there may be more than one instructor.

(c) For each section of a course, input the TA/Grader name, and hours the TA/Grader will assist this course.

(d) The course list of computer science. For each course, there is an attribute of catalog whose value is year. E.g., a course with catalog 2014 means that the course is a course in 2014 catalog.

2. A professor can login (**optional**) and do the following tasks

(a) Update her/his preferences of the courses to teach in a given academic year (e.g., Fall 2013 to Summer II 2014). As for the user interface, all courses will be displayed with the professor's preference value (1-3) for that year (if the preference value is not there yet, use the previous year's preference values). The professor can edit the values. A preference value of NULL represents that the corresponding course is not preferred by the professor.

(b) Update her/his teaching load distribution (only in fall and spring) preference: more load in fall or more load in spring or don't care.

(c) Input special request for a given year: course code, title, justification (< 200 words).

(d) For each section of a course assigned to this professor in a given semester, input the text books with the following information: Text Title, Author, Edition, ISBN #, Publisher. Here we assume the assignment of the given semester is already inside the database. If the course was taught before by this professor, the most recent text information should be displayed as the default text information.

(e) See the courses assigned to them in the next semester. For each course, display its course code, time, days, room and building.

3. A business manager can obtain the following information:

(a) For any given instructor and a number n (normally $n = 5$), list all the courses (course code, title, semester, enrollment, building), in reverse chronicle order, that the instructor has taught in the last n years.

For a given instructor and a number n, the manager may also want to know the number of distinct courses with the times a course is repeated, average enrollment of this course, average TA hours (per week) for this course, and the ratio between the average TA hours and the average enrollment of this course. As an example,

the information to show is something like: (CS4354, 5, 20, 4, 0.2), (CS5285, 1, 10, 0, 0). The first tuple means that the instructor has taught CS4354 5 times with average enrollment 20, average TA hours 4 (per week). Certainly it helps if you show the information in a table with the meaning of each column (and/or row) given explicitly. The undergraduate courses should be shown before the graduate courses. A clear separation of undergraduate courses from graduate courses is preferred. The required courses should also be made distinguishable from the elective courses.

(b) For a given n, show a table which contains the following summary information for each professor: the ratio between the total number of TA hours and the total enrollment of all undergraduate courses (and graduate courses respectively) this professor taught in the past n years, the number of all distinct courses and the number of new courses taught in the past n years, the total number of undergraduate courses (*not* just distinct ones) in n years and the total number of graduate courses in n years.

(c) Given a number n, for each section of a special course (CS5331/CS5332), list its title, instructor, offered date (e.g., fall 2012), and enrollment.

(d) For any given course and a number n, list all of its offerings in the last n years. For each offerring, list its section number, instructor, enrollment and date, in reverse chronicle order.

(e) See the preferences of all professors of any given year.

(f) See the text(s) used by a professor for a given course (code). If the professor has taught this course several times, list all texts and their corresponding semester that are used before in reverse chronicle order.

(g) Given a number n, list all summer (I and II) courses with course code, instructor and enrollment in the last n years.

(h) Given a number n, show the following statistics:

i. For each course, the total enrollment for each regular semester, i.e., fall or spring, in the last n years.

ii. For each level (1000, 2000, 3000, and 4000, and 5000), the total enrollment for each regular semester in the last n years.

4 The deliverables

1. The source code of a web application using `PHP` and `mySQL`. Your web application should be executable, i.e., user can use your application through a browser.

2. A readme on how to install the and use your web application.

3. A project report (for both intermediate and final) including

   (a) ER diagram. You may check if there is any UML or ER model tools to help draw the diagram although you don't have to.

   (b) Relation schemas and explanations on why you design such schemas.

(c) SQL pseudo code on how to answer all the queries given in the problem section.

(d) The design/pseudo code of your `PHP/mySQL` program.

(e) User interface design (simple yet friendly).

(f) Contributions of each team member.

5 Organization and Deadlines

You must form a team of three.

Your are supposed to finish all the design and build a prototype system by **Apr 21**. Deliverables: your report and the prototype system. There will be a face to face interview where you explain your design and demo your system. Up to 40% of the total project grade will be awarded to your work.

Your final report and system will be due **May 5**. There will also be a face to face interview. Up to 60% of the total project grade will be awarded to your final work.

6 Design of your web application

Here we provide more information on the design of your web application. You are expected to pay special attention to the design of the database and the `PHP` (or `Python`) program, and the design of the user interface. Your design should be documented in your project report in the way defined in the previous section.

6.1 Database and the control program

In your report, pay special attention to

1. ER diagram,
2. Schemas design and justification, and
3. SQL queries to answer all questions given in the problem section.

You are expected to give the pseudo-code and design ideas of your `PHP` (or `Python`) program.

6.2 User Interface Design

The user interface of your web application is expected *simple yet friendly*.

To obtain motivation and inspiration on the design of user interface, there is no better products to study than the apple products. Read chapter 12 (THE DESIGN – Real Artists Simply) and chapter 26 (DESIGN PRINCIPLES – The Studio of Jobs and Ive) of the book *Steve Jobs* by *Walter Isaacson* (kindle edition 2011-10-24) published by Simon & Schuster. You can find the book here `http://www.maismac.net/steve_jobs_by_walter_isaacson.pdf`.

Here are some excerpts from the chapters.

" 'The main thing in our design is that we have to make things intuitively obvious,' Jobs told the crowd of design mavens."

"To be truly simple, you have to go really deep. For example, to have no screws on something, you can end up having a product that is so convoluted and so complex."

"Ive [the chief designer of Apple]'s goal is 'to make things so simple that they often seem undesigned.'"

"At most other companies, engineering tends to drive design. ..... In the early days of Apple, Jobs had approved the design of the case of the original Macintosh, and the engineers had to make their boards and components fit."

7 Database systems to use to develop the web application

You are expected to develop your web application using `PHP` and `mySQL`. In the following, we provide some information on web application development and `PHP/mySQL`.

7.1 General information on web application development

The sections 9.1-3 and 9.7 of the text provide an excellent introduction to develop web applications with a database backend.

7.2 `PHP` and `mySQL`

Installation of the tools. Use the all-in-one installer XAMPP (`http://www.apachefriends.org/en/xampp.html`) to install `PHP`, webserver, and MySQL. For windows installation, go to `http://www.apachefriends.org/en/xampp-windows.html`. The installation/test instruction there is very easy to follow.

Develp a simple web application using `PHP`.

1. Create a directory `myWebPages` under the `htdoc` directory of the directory where your `XAMPP` application resides.
2. Under `myWebPages`, create a file named `hello.php` with content as follows

```
<html>
 <head>
  <title>PHP Test</title>
 </head>
 <body>
 <?php echo '<p>Hello World</p>'; ?>
 </body>
</html>
```

3. In the URL bar of your browser, type `http://localhost/myWebPages/hello.php`, you can see the output of the program.

4. References. Now you can go to `PHP` manual/tutorials, e.g., `PHP` simple tutorial `http://www.php.net/manual/en/tutorial.php` and `PHP` manual `http://php.net/manual/en/`.

Database system `mySQL`

1. The `XAMPP` application offers an web application to access `mySQL`. Run the `XAMPP` control panel (you have to quit Skype before you run this application). Click the admin button in the row of `mySQL` on the panel. As a result, you'll be brought to a web page where you have an interactive user interface to create and access (by issuing SQL queries) databases.

2. A tutorial on how to use web front end of `mySQL` can be found at `http://www.siteground.com/tutorials/phpmyadmin/`.

   (a) Create databases (interactive manner) `http://www.siteground.com/tutorials/phpmyadmin/phpmyadmin_create_database.htm`.

   (b) Run SQL on the database created:
   `http://www.siteground.com/tutorials/phpmyadmin/phpmyadmin_mysql_query.htm`

   (c) Detailed SQL syntax specification:
   `http://dev.mysql.com/doc/refman/5.1/en/sql-syntax.html`, or `http://downloads.mysql.com/docs/refman-5.5-en.a4.pdf` (large file, need patience.)

Develop database backed `PHP` **programs**

1. The diagram in the textbook (P370) seems to be useful to give one an overview on the development of database backed `PHP` program.

2. A simple introduction of `PHP` and `mySQL` by W3school can be found at `http://www.w3schools.com/php/php_mysql_intro.asp`.

   In the web page above, search `PHP Database`, you will find the topics, below the entry `PHP Database`, from connecting to a database, creating a database to using various SQL queries.

8 Progress report

Submit weekly report on time.

9 Aknowledgement

Our TA Alaa Darabseh and stduents Albot Benjamin, Lazo Diego, Rex, Tara Parker have contributed to the design and document of this project. Jon Bastnagel, Bailey Everts and William Ray also point to me interesting systems.