Course Rank System

Dharma Teja Bankuru    Sai Kiran Katuri    Sairam Bade
130050049             130050051             130050079

October 11, 2015

Software Requirement Specification
Database and Information Systems
Dept. of Computer Science
IIT BOMBAY
1 Introduction

- Create an interface where students can review courses and plan their academic program by accessing official university information and previous year's grading statistics.

- Students can comment under a course asking for some more information about the course and instructors or other students who had done the course can reply to it.

- Students who already took the course can rate the course and give their review about it.

- A professor teaching a course can update or modify its course content accordingly during his teaching period.

2 Users of Application and Volumes

There are four types of users to this application namely Admin, Instructors, Students and Spectators.

ADMIN :
Admin can add new courses or delete existing courses from the course structure and approves the requests of students or instructors. He also updates the timetable of the institute every semester based on instructor assignment to courses for teaching and registration done by students.

INSTRUCTOR :
Instructor teaches some courses(stored in teaches table of database updated by admin of institute) and he can modify the content of his courses and can also reply to the queries posted by students under the course.

STUDENT :
Students can view the courses and post their queries regarding courses below the course description page. He/She can also review and rate the course after completing the course in previous offerings.

SPECTATOR :
Spectators are people not from institute who has to sign up in order to view courses. After signing up, they can view the contents of the course and comment on them if needed.

In general taking our university as an example, which contains about 5000 students and 500 instructors and IITB being a very notorious institute, course information will be viewed by many users around the world. Hence, there will be around 100000 views per day and around 10000-15000
comments made (globally over all courses). However, during registration this count can go very high as many students of IITB will look at grading statistics and course reviews before registering for some courses. Hence, at peaks there will be around 100-200 database accesses per second.

3 Tools we are planning to use

We are planning to use postgresql database for storing the data. We are planning to use JDBC to make this application. We are planning to use eclipse IDE that we learnt in our labs.

4 Inputs, Data transactions and processing

- User is asked to enter the user name and password in order to log-in to interface.
- He can sign up if not registered already. He is allowed to choose an user name which must be unique. These will be added as spectators.
- If an instructor who joined recently into institute want to sign up he can sign up using his current LDAP id.
- Admin will confirm his identity and add him to instructors table (more on this tables later).
- Instructor can update the modified content of a course if he is currently teaching that course which can be known using the teaches table present in database which will be updated by the admin.
- When an instructor updates modified content updates must be made in the mod_content table of database after checking for satisfaction of constrains.
- Students of institute can review courses that were already done by them. The reviews will be added to the reviews table of database.
- After the semester ends, grading statistics of the courses will be added to statistics table of database making use of takes table.
- Name of instructor who had given the grades is also added to this database. This can be retrieved from teaches table.
• Users can make comments under each course’s page. Peers can also give replies to others comments.

• A comment made will be added to comment table of database. All the replies will be added to reply table of database.

• Deletion of comments is permitted to only admins.

5 Tentative Schema of Our Database

Kindly note that we came with this schema without starting up with ER diagram. This is tentative and we came up with this just to get started. Original one will be submitted in the detailed design document next week.

| Table Name | Attribute1, Attribute2, ...
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>people</td>
<td>ID, password, name, address, admin_ins_std_spec, approved</td>
</tr>
<tr>
<td>admin</td>
<td>ID, qualification</td>
</tr>
<tr>
<td>student</td>
<td>ID, dept_name, tot_credits</td>
</tr>
<tr>
<td>instructor</td>
<td>ID, dept_name, salary integer</td>
</tr>
<tr>
<td>spectator</td>
<td>ID, dept_name, salary</td>
</tr>
<tr>
<td>department</td>
<td>dept_name, building, budget</td>
</tr>
<tr>
<td>course</td>
<td>course_id, name, dept_name, credits, course_type, half_full_sem</td>
</tr>
<tr>
<td>prereq</td>
<td>course_id, prereq_id</td>
</tr>
<tr>
<td>takes</td>
<td>ID, course_id, semester, year, grade</td>
</tr>
<tr>
<td>teaches</td>
<td>ID, course_id, semester, year</td>
</tr>
<tr>
<td>org_content</td>
<td>course_id, org_lec_credits, org_tut_credits, org_practical_credits, org_self_study_credits, org_text_ref, org_description</td>
</tr>
<tr>
<td>mod_content</td>
<td>course_id, mod_lec_credits, mod_tut_credits, mod_practical_credits, mod_self_study_credits, mod_text_ref, mod_description, ID, modified_at</td>
</tr>
<tr>
<td>statistics</td>
<td>course_id, semester, year, tot_students, grade_stats</td>
</tr>
<tr>
<td>reviews</td>
<td>ID, course_id, semester, year, tot_students, grade_stats</td>
</tr>
<tr>
<td>comment</td>
<td>comment_id, ID, course_id, comment, created_at</td>
</tr>
<tr>
<td>reply</td>
<td>comment_id, ID, course_id, comment, created_at</td>
</tr>
</tbody>
</table>
6 A walk through our Database(tentative)

Kindly note that we came with this schema without starting up with ER diagram. This is tentative and we came up with this just to get started. Original one will be submitted in the detailed design document next week.

A detailed schema diagram of this database is present in previous page.

people:
- people(ID, password, name, address, admin_ins_std_spec, approved)

  Everybody is given an unique ID while signing up and password which is encrypted will be stored along with it in the table. It also contains name, address. An integer admin_ins_std_spec which specifies whether a person is an instructor or a student or an admin or a spectator. It also has a Boolean "approved" specifying whether the person is verified as an student, instructor belonging to this institute.

admin:
- admin:(ID, qualification)

  These table contains all the admins along with their qualifications.

student:
- student(ID, dept_name, tot_credits)

  This table contains all the students from this institution approved by admin.

instructor:
- instructor(ID, dept_name, salary integer)

  This table contains all the instructors from this institution approved by admin.

spectator:
- spectator(ID, dept_name, salary)

  This table contains people not belonging to this institute along with their institute name and degree.

department:
- department(dept_name, building, budget)

  This table contains all the departments present in the institute.
course(course_id, name, dept_name, credits, course_type, half_full_sem)

This table contains all the courses offered in the institute along with number of credits, the type of course (either theory or lab) and whether the course is offered full semester or half semester.

prereq:
prereq(course_id, prereq_id)
This table contains all the prerequisites of a given course.

takes:
takes(ID, course_id, semester, year, grade)
This table gives the information about student taking a course in a semester and year along with the grade he/she obtained in that course.

teaches:
teaches(ID, course_id, semester, year)
This table gives information about all the courses an instructor is teaching in a given semester and year.

org_content:
org_content(course_id, org_lec_credits, org_tut_credits, org_practical_credits, org_self_study_credits, org_text_ref, org_description)
This contains original information of a given course.

mod_content:
mod_content(course_id, mod_lec_credits, mod_tut_credits, mod_practical_credits, mod_self_study_credits, mod_text_ref, mod_description, ID, modified_at)
This contains modified or updated information about a course and ID of the person who modified it and time at last it is modified.

statistics:
statistics(course_id, semester, year, tot_students, grade_stats)
This table contains the number of students who obtained a particular grade obtained in a course in a given semester and year.

reviews:
reviews(ID, course_id, review, rating, difficulty_level)
This table gives the rating and review given by a student for a course along with its difficulty level.
comment:
comment(comment_id, ID, course_id, comment, created_at)
This table contains all the comments under a course along with the time at which it is created.

reply:
reply(comment_id, ID, course_id, comment, created_at)
This table contains all the replies to a particular comment under a course along with the time at which it is created.

7 References Used

(1) https://www.mysql.com/products/workbench/design/
(2) http://www.postgresql.org/
(3) http://www.latextemplates.com/