University Database

Team 5

Ee Yieng Zheng
Raymond Hong
Christine Pham

Section 3
09/28/2017

Table of Content

Project Description

Goal

The application is a private database for a university to store information. The database may be ordered by relation. Information may include departments, staff and faculty members, buildings, courses, students, or more. Users will be able to query the database for specific information. Students can add/drop courses and professors can assign grades.

Motivation

A university is complex. There must be an easy way to structurally organize all members associated with a university, whether it be students, staff, or outsiders. Using relational techniques, we can organize a university based on the function of its members.

Stakeholders

The database will store the university's numerous departments, staffs and faculties, courses, and students information, therefore the stakeholders are faculty and students.

Application Domain

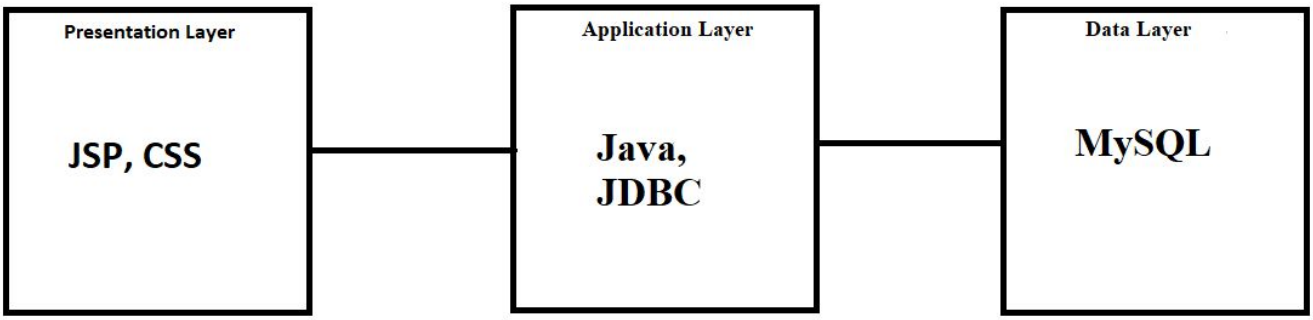
The database is relevant to the field of education.

Benefits

Having a private database that stores university information is highly beneficial to both students and professors alike, because storing that much information without a database would be daunting. The main benefit of a database is its query functionality. Students will be able to quickly and efficiently query the database for information about their courses. They can look up available courses and obtain useful information about them such as the current professor, the time, and the location of the class. This allows students to easily plan out their future class schedules. Students can also utilize the database to drop themselves from courses when their workload is too much for them to handle. Professors will be able to see their classes and students within each course. They can also assign grades to students.

System Environment

System Structure



The system will follow a 3-tiered architecture with 3 layers: a presentation layer, an application layer, and a data layer. The user will interact with the presentation layer. The presentation can pass user input to the application layer. All business logic will be in the application layer. The application layer will then pass a query to the data layer. After the database returns a response, the application layer will process the response and send the formatted output to the presentation layer to be displayed.

Hardware

We will be testing the system environment on Macs, Linux, and Windows.

Software

We will be using different software environments such as Atom/Eclipse for Java/JDBC control, and Brackets for HTML/CSS control.

RDBMS

We will be using MySQL 5.7.19 as the database.

Application Languages

We will be using Java, HTML, CSS, XML, and SQL.

Functional Requirements

User Access

- Fr.1 User Login
    - Fr 1.1 Users can login by typing in their username and password
        - Description: Users can login to the system after inputting a valid username and password.
    - Fr 1.2 The system shall provide different user interface options depending on the account type that was detected.
        - Description: Upon login, the system shall provide different UI options. System admins will have the most control. Presidents and deans will have more control than the head of departments. The head of departments will have more control than professors. Professors will have more control than students.

- Fr.2 Update Profile Information
    - Fr 2.1 Users can update their own personal profile information
        - Description: Users can edit their own profile information including preferred email, home address, and phone number.
    - Fr 2.2 Users can change their own password
        - Description: Users can change their own password by inputting a new one. Before the password change is approved, the user will be asked to enter their current password. The current password must be valid before the user can change their password.

- Fr.3 Search for Courses
    - Fr 3.1 The system shall allow users to query the course database for existing courses.
        - Description: Users can search courses by course name, time, or department. Users can also specify whether they want to see waitlisted/closed courses or not.
    - Fr 3.2 The system shall allow users to query the course database for courses that can fulfill a GE requirement.
        - Description: Users can see a list of courses that fulfill a particular GE requirement. For example, users can see a list of area B2 courses that they are currently available.

- Fr.4 Student Functions
    - Fr 4.1 Students can add courses to their current schedule

        - Description: Upon selecting a course, a student can choose to add the course to their schedule. The system will check to ensure that the student has met the prerequisites and that there are no time collisions. If so, the course will be added if it is not full.
    - Fr 4.2 The system shall allow students to drop courses.
        - Description: Students can go to their course list to drop a course.
    - Fr 4.3 Students can see their course list.
        - Description: Students can see a list of courses that they are currently enrolled in as well as courses that they taken in the past. The course information including the professor, the location, the time, and the number of students enrolled will be provided for current courses.
    - Fr 4.4 Students can access their grades in each of their courses.
        - Description: Students can view their grades in their past courses.

- Fr.5 Professor Functions
    - Fr 5.1 Professors can add/drop students from their courses.
        - Description: Professors can add students to their courses by inputting the student's student ID. Professors can add students even if the student does not meet the prerequisites or the class is full. Professors can also drop students from their courses.
    - Fr 5.2 Professors can view the courses that they are currently teaching
        - Description: Professors can see a list of courses that they are teaching. The course information including the location, the time, and the number of students enrolled will be provided.
    - Fr 5.3 Professors can assign grades to each of their students after the course has ended.
        - Description: If a class has ended, professors can assign grades for every student.

- Fr.6 System Administrator Functions
    - Fr 6.1 System administrators have the power to do everything that students and professors can.
    - Fr 6.2 System administrators can disable/remove currently existing user accounts.
        - Description: Students who are no longer enrolled or professors who no longer work at the university can be removed from the system by system admins.
    - Fr 6.3 System administrators can edit the profile details of other users.

        - Description: System admins can edit other users' names, addresses, emails, phone numbers, grades, etc.

- Fr.7 Head of Department Functions
    - Fr 7.1 The Head of Department is responsible for assigning a professor for each courses. They can also hire/fire a professor, promote and increase/decrease wage.
        - Description: add/remove students to major, add/remove professor, power over professors.
    - Fr 7.2 Provide services to students: graduation, change of major, double major, petition, report students for plagiarism, etc.
        - Description: ability to modify students, courses, rooms, and professors relations.

- Fr.8 President of the University and Deans of Colleges
    - Fr8.1 Responsible for the university's finances
        - Description: increase/decrease fundings to specific departments. Assign/reassign deans.
    - Fr8.2 Deans allocate funds to college departments. Assign/reassign head of department.

Non-Functional Issues

Graphical User Interface

The graphical user interface will be built using HTML and CSS through JSP. It will display on modern web browsers such as Mozilla Firefox and Google Chrome.

Security

The system can only be accessed by inputting a valid username and password. Passwords will be hashed and stored for security.

Access Control

Access to the database is limited depending on a user's account type. The system will have a hierarchical structure. Based on this hierarchical structure, different users will have different permissions. For example, the head of department will have the ability to control professors' pay, or professors can control the students' grades. Non-members will have no access, and students will have access to view their courses. System administrators have complete control of the system and can access all information besides user passwords.