

COURSE MANAGEMENT

Abstract

The Course Management Database aims at giving the students an idea about the offered courses. Students often face a dilemma while choosing a course. They have many questions regarding each subject such as, what is the subject about? What is the syllabus of the subject? Who is handling the subject? How will it be useful for us? And so on. The answers to these questions lie in this database. With this information at hand students can make an informed decision.

The Database has four entities, Department, Faculty, Course and Student. Each course in the database is categorized into two different groups, Open Courses (offered by other departments) and Closed Courses (offered by Computer Science Department). A Course is characterized by Course Code, Course Name, Prerequisites, Number of credits allotted to the course, the semester for which it is offered and the proposed Syllabus.

The Department has attributes such as, Department Code, name and Department head. The faculties are characterized by their Social Security Number (SSN), Faculty Name, Years of Experience, Designation and the Department to which they belong to. Students have University Seat Number (USN), Student name, Semester number and the Department Number as their attributes.

The Database not only helps the students to choose a course, but it also gives a provision to register for the course. It gives the faculty handling a course the option of changing the course credits, prerequisites and syllabus. The faculty handling a particular course can view the number and details of the students who have enrolled for that particular course. The Administrator has to provide authorization to modify details of a course, faculty or a student (add, delete and update). The Administrator can create a new administrator account or a faculty account.

In conclusion, The Course Management Database makes selection and management of courses simple and easy to implement.

Chapter 1

Introduction

1.1 Objectives of the project

This project was designed with the purpose of making the process of choosing courses (both open and closed) for the students of the Computer Science Department, MSRIT more systematic and convenient. Its primary aim is to computerize the whole system and allow students to view the courses offered along with the course attributes and register for a course. The faculty should be able to view the students registered for a course and have privileges to edit course attributes such as credits, prerequisites and syllabus. The Administrator of the Department should be given privileges to insert, delete or update student, faculty or a course. Both these sections of the application is, of course, password protected, so as prevent illegal access and modification of details of courses, students and faculty.

1.2 Current Scope

Usually, when the offered courses are announced, course details are not provided. In such a case, a student fails to make the right choice and may take a course, which may not interest him once he gets to know what the subject is all about. In order to avoid such a situation, the course management database allows students to view the offered courses and various details associated with the courses which will help them to make the right choice. Generally, the process of choosing a course involves actually going to the concerned department, filling out a form, and submitting it, which is tedious work. To counteract this problem the student can register for a course of their choice through this database.

1.3 Future Scope

This application will be available either as a link from or as one of the web pages on the official MSRIT website (www.msrit.edu).

Chapter 2

Software Requirements Specification

2.1 Introduction

This section of the document is designed to specify the software requirements to be met by any system to run the application Course Management Scheme. We specify the general and specific requirements here.

2.2 General Description

2.2.1 Project Perspective

This project is to be viewed as an application made for the students of the Computer Science and Engineering department of M. S. Ramaiah Institute of Technology for the purpose of registering for courses from both their parent department and from other departments. Students have the freedom of researching about a certain course and making a clean and informed decision about the courses. The project also allows faculty to change the details of courses, mainly syllabus.

2.2.2 Product overview

The project has been developed basically to make the overall process of choosing and registering for courses as easy as possible.

2.2.3 End User expectation

The end user, if he or she is a student, can expect the following:

1. View the courses available for a particular semester
2. Register for single or multiple courses from either the parent department or from other departments

If the end user is a faculty member, he or she can expect to be able to:

1. View the details of the students registered for the course(s) currently being taught by him or her.
2. Change the details of the course courses being handled by him or her, such as course number, prerequisites, and credits

The administrator is granted the following rights:

1. Assigning a teacher to a certain course.
2. Add/Delete/Modify a course/faculty/student.
3. Register a student for a certain course. This option has been specified if a department decides to not implement the process of the student registering themselves using the application and uses the usual way of students going to the department and registering. The administrator can add all students on a whole to the course.

2.3 Assumptions and Dependencies

1. Each department has an administrator who has a unique loginID and password, and only he/she can update details for that particular department.
2. The loginID of each faculty is their Social Security Number.

2.4 Specific Requirements

2.4.1 Functional Requirements of the end product

The following are the functions the application is expected to be capable of performing:

1. Every Department offers courses.
2. Courses are of two types, open and closed.
3. Each department offers open courses to other departments.
4. A student has to choose atleast one open and one closed course.
5. A student cannot register for more than one course in the same group.
6. Students are identified by their USN, name, and semester.
7. The faculty is characterized by SSN, name, designation, qualification, and years of experience.
8. Courses are identified by code, name, type, group, pre-requisites, credits and syllabus.
9. The student should only be able to view the offered courses and register for only one course in a group.
10. The faculty in charge of a course should be able to view the students who have enrolled for that particular course.
11. The faculty in charge of a course should be able to edit the details of the course.
12. The Department Administrator should authenticate himself before using the application.

13. The Administrator should be able to insert/delete/update faculty, student and course details.
14. The administrator should be able to allot a course to a student.

2.4.2 Software system Requirements

1. XAMPP version 1.7.3a, for Ubuntu or Windows.
2. Web browser.

2.4.3 Hardware system Requirements.

1. 128 MB RAM
2. Windows 2000, XP (Server 2003), Vista (Server 2008), Windows 7, Ubuntu.
3. 70 MB of free space for Linux Operating systems, 300 MB of free space for Windows operating systems.
4. x86, x64 bit processors.

Chapter 3

Data Model Design

3.1 Identified Entities, Attributes and Relationship

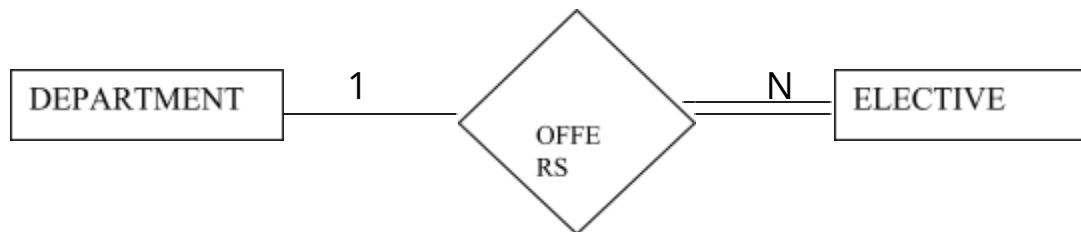
Identified Entities: Department, Course, Faculty, Student

Attributes:

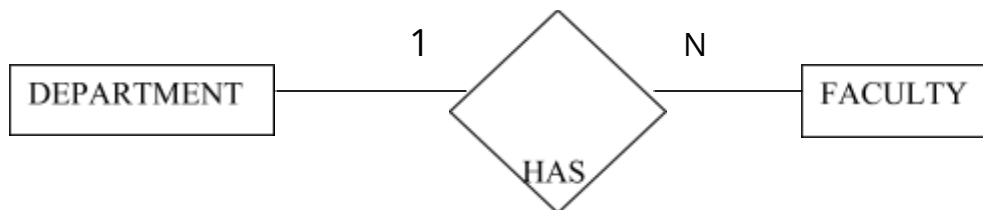
1. Department : DCode, DName, DHead.
2. Course : ECode, EName, Credits, PreRequisites, DNumber, FSSN, Sylsum, Type, EGroup, Semester.
3. Faculty : SSN, FName, DOB, DOJ, YearExp, Desig, DNumb.
4. Student : USN, SName, Sem, DOB, DNum.

Relationships

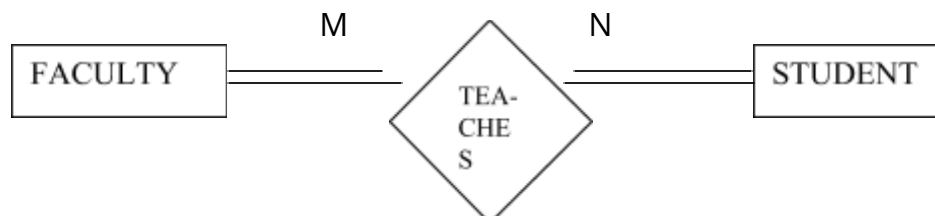
A Department can offer many courses whereas a particular course can be offered by only one department. The relation is total towards course.



A Department employs many faculty but a teacher can be employed by only one department. The relation is total both sides.



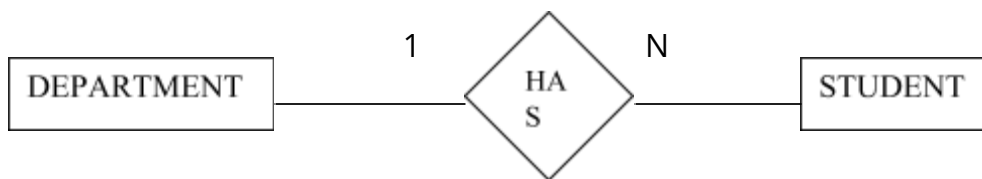
A Faculty can teach many students and a student can be taught by many teachers. It is a total relation at both sides.



A course may be chosen by many students likewise a student may choose many courses. It is a total relation towards course.



A Department has many students where one student can belong to only one Department. It is a total relation at both sides.



3.2 Entity-Relationship Model (E-R Diagram)

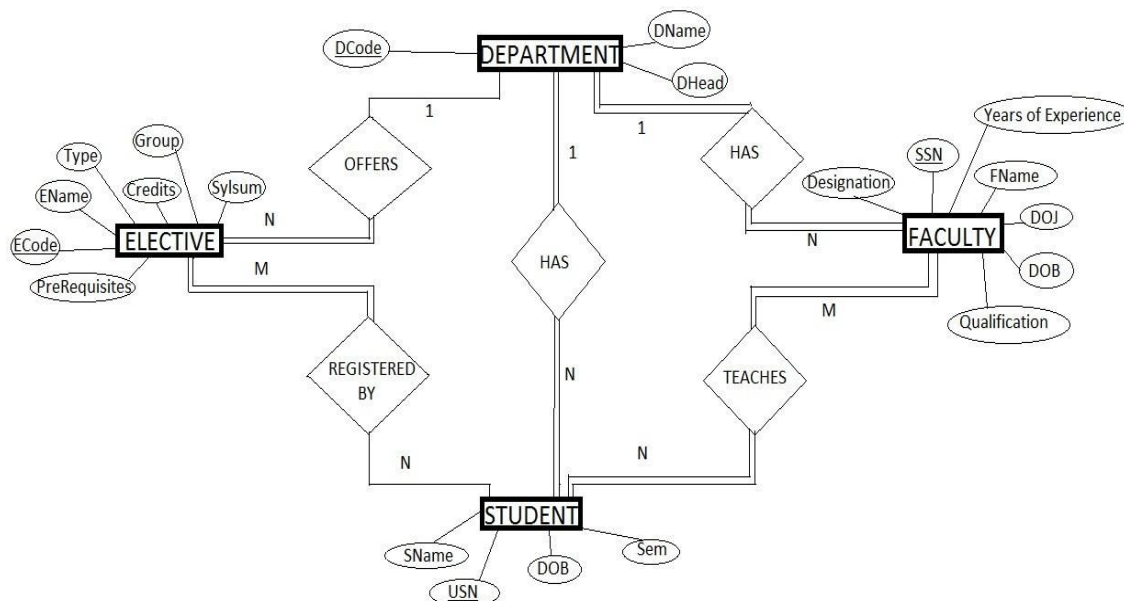


Fig 3.2 ER Diagram

3.3 Relational Schemas

Department

<u>DCode</u>	DName	DHead
--------------	-------	-------

Course

<u>ECode</u>	ENAME	Credits	PreRequisites	DNumber	FSSN	Type	EGroup	Sem
--------------	-------	---------	---------------	---------	------	------	--------	-----

<u>SSN</u>	FName	OB	DOJ	YearExp	Desig	DNumb
------------	-------	----	-----	---------	-------	-------

Faculty

Stud_Fac

<u>StuUSN</u>	<u>FacSSN</u>
---------------	---------------

Stud_Elec

SUSN	ECOD
------	------

Student

<u>USN</u>	SName	Sem	DOB	DNum
------------	-------	-----	-----	------

Chapter 4

Design and Implementation

This software which can be used for course management was build using the scripting language PHP as the front end and the XAMP server which is free and open source cross-platform web server package. Database queries were executed in MySQL. Some values for reference and verification purposes were inserted into the database from the back end. Since XAMPP is cross-platform the whole project has been implemented on Ubuntu, as there would be no difference from the Windows version of XAMP. The project is easily portable as long as XAMPP is running on the computer.

The basic purpose of this project was to the make the process of choosing and registering for courses as user-friendly as possible with the maximum functionality. PHP allowed us to do the coding with utmost ease. Coding with PHP is similar to coding with any scripting language or any procedural language, there are local variables and SESSION variables which can be considered as global variables.

A student views the list of courses he's eligible for by entering his semester and department and the course type he's interested in. The database is queried with a select statement using these parameters, and the resulting array of rows is displayed. The student can register immediately viewing the courses. When he does so, an entry is immediately made into the database using the insert query. Various extreme cases, such as a non-existent student, double registration (in the case of the Group constraint) have been checked by querying the database accordingly. On viewing the list of courses the student can also view all the details of the course he/she is interested in by entering the course code. The table 'Course' is queried and the details of the course are displayed in a table format. The table format is shown using the *table* tag of HTML embedded in the echo statement of PHP. Further, the student can view the details of the faculty handling the subject, the table 'Faculty' is queried with the Faculty SSN and the results are displayed.

When the faculty tries to access the software, he/she can do so only with a FacultyID and a password. This authentication is done. The faculty can view the student details of the courses he/she is handling. This is done using the query which checks the FacultyID with the course table and outputs the result, as a drop down menu from which the faculty can choose the course

he/she wants to view the details of. A session variable for the FacultyID is set, to see to it that the faculty can manipulate/view details about his/her course only. The Faculty can change the credits, pre-requisites and the syllabus of the course he/she's handling. This is done using the update query using the details entered in the HTML form.

An administrator has complete access of the database. He logs in with a loginID and password. An administrator can add accounts i.e. Admin accounts or Faculty accounts. This is done again using an insert option which is reflected upon in the database. Whenever an account is created an entry is made with the Username mapped with the Password in the database. Administrator also can add, delete, and modify the details of a course, faculty or a student. Deletion and Insertion are done using the Delete and Insert queries, for the respective tables. Besides the primary key i.e. the course code the administrator can modify any other detail of the course. Faculty's qualification, designation, years of experience, Date of Birth and Date of Joining can be modified. If an account has been created with certain fields as NULL, for example Date of Birth, then those values can be inserted using this modify function. A student's semester, or date of Birth can be modified. Radio Buttons are used to show which detail of the course/faculty/student is modified. The modified values are reflected upon the database using the update query.

The entire project has been implemented using PHP, the results of all the queries have been displayed in table form, drop down menus using HTML tags in PHP. The main objective of this project was to club in the maximum functionality with constant error checking, and a easily understandable user interface.

Chapter 5

Conclusion and Future Enhancements

This project was made with the purpose of making the process of choosing courses (both open and closed) for students of the CSE department of MSRIT more systematic and convenient. Its primary aim is to computerize the whole system and allow students to choose their courses online through the use of an application based on our project. Students tend to choose a course with respect to what their peer group does, usually not doing any research on their own, this happens because one, the student knows nothing about the course other than the name, and two, choosing the course must be done within a very short span of time usually two days. To avoid this, a website showing the details of various courses, importantly the syllabus, which has been updated atleast a fortnight before the courses have to be chosen, would be very beneficial for the students.

The students also have the provision of registering online, so a student need not hop from department to department signing up for courses, everything can be done from one place. If there is a limit on the number of students that can register for a course, the software shows that the registrations are closed for a course. This allows the student to simply register for another course, and does not have to wait for the department to count the number of application forms and choose the first 60 students and reject the rest, after the last date for the registration of courses.

There are provisions for the faculty-in-charge to modify course details, like syllabus, credits and prerequisites. The faculty can view the students in his/her course class, so the faculty-in-charge also has a helpful online database of the students in his course.

Of course as with website/database there needs to be an administrator for maintaining, and updating the database. The administrator has complete control of the database, so as and when required, that is when a semester is about to end, he/she can update the database, adding details of new courses, modifying details of old ones i.e. if the same course is offered again but for a different semester of students. The administrator also assigns faculty to a course. Administrators can also update/insert/delete student and faculty details.