

CS 2541W Team Project

Registration System (REGS)

Phase 1

The REGS system provides a course registration system.

- Students use your online system to register for courses. After a course is taken, instructors are able to submit grades, and a transcript can be viewed showing courses and grades.
- The REGS must support course registration by students, grade entry by instructors and Grad Secretary (GS), and both students and GS can search for transcripts.

You will eventually deploy your system on our production server using MySQL and Python Flask. Remember that you will need to integrate your application with other modules in Phase 2 – so be careful about what other software you use.

User Interface Design: For the final project we will evaluate your user interface more stringently. In Phase 1 the emphasis is correctness but we expect good user interface design methods applied to your project (such as user friendly system, input form validation, etc.).

Description of the REGS System

The REGS system provides an online course registration system that allows students to register (add/drop) for courses, check for their registration record (courses they have taken and grades received – similar to a transcript), and allows faculty instructors to enter grades, and grad secretary (GS) to search for transcripts and enter/change grades.

REGS workflow:

You must implement the workflow below. For specific data needed for this application, refer to the information below as well as your analysis of what other data may be required.

- Each graduate student should be able to create an account that enables them to log into the system.
- A user registered in the system as a graduate student can enroll for graduate courses using a web registration system.

- A graduate student has personal information that identifies them.
 - Each student has a unique university ID (UID) which is an 8 digit number.
 - The system must store the last and first name of the student, and other personal information such as address.
 - A student can be enrolled in the Masters program or the PhD program; the system must store this information.
 - The system must be able to provide a login for each student in the university.
- The system stores course enrollment information for each student. This information reflects the functionality provided by the registration system; i.e., courses taken by the student, the semester and year taken, the final grade for the course (if completed), number of credit hours. In other words, information that is typically found on a transcript.
- The web registration system enables students to enroll for courses, and faculty (or the GS) to assign grades. A student will be able to use the online registration system to register for courses, and a faculty (or the GS) can use the system only to enter the final grades. The valid final grades are (A, A-, B+, B, B-, C+, C, F).
- At any time the GS, a faculty member, or the student can query the system for the current transcript of the student. Courses currently in progress show up with a grade of IP (in progress).
- The registration system must store information on the courses, the faculty teaching the course, the grades (if a final grade is not assigned then a grade of IP should show up), the schedule (time and day). The course registration application must check for schedule conflicts.
 - To simplify the registration system you can assume the following:
 - Each course has a department (subject), course number, title and number of credit hours associated with the course. No two departments can have the same department name and within a department the course number is unique. The schedule should include the course information above, a section number, the semester it is offered in, and the day and time. *For Phase 1 you can make a simplifying assumption that each course has only one section* (but your design should allow changes to allow multiple sections if requested in Phase 2).
 - The courses that are scheduled are part of a university course catalog. The course list specifies the pre-requisites for each course. The initial data for the course catalog is provided in the appendix in this document.
 - To simplify the registration system, you MUST implement the schedule provided in the appendix. You can choose to populate with additional courses if you choose. The schedule specifies a day and time, and a course number as explained in the appendix. (In Phase 2 you may be required to extend this system so that an authorized user can add courses to the schedule of classes.)

- The enrollment/registration information must store the course, the semester, the class time, day, the student identification, the instructor information, and the grade. Note that a course offering is unique when one considers the department, course number and the semester & year (and section if there are multiple sections) that it was offered in – this is what a CRN captures in Banner (but we will stick to a simpler format). For simplicity you can assume that courses are scheduled on M, T, W, R, F and each course meets once a week (like a lot of our grad courses), and the valid time slots are 3-5:30pm, 4-6:30pm and 6-8:30pm only (i.e., only three time slots). Assume that the same schedule is applied every semester (i.e., the schedule shown in the list is assumed to be in place every semester). To simplify matters, your system can allow students to take two consecutive classes that are 30 minutes apart in their end time and start time.
- Additional parameters that the registration system could require in the final product are location (room) and capacity (max class capacity). While this is not required in Phase 1, think of how your system may scale.
- For further simplification, assume that (a) there is no limit on the number of students in the class, and (b) assume that PhD students can only register for graduate credit courses (they are numbered 6000 level). You will get extra points if you implement a more complex system.
- A student can register for courses by adding a course. The system must also support the feature of dropping a course.
- A faculty instructor teaching the course can submit grades. Once submitted the grades cannot be changed by the faculty.

Note on planning for scalability (i.e, for future enhancements and features) and Important relevant information: In Phase 2, you will also need to implement a number of queries/reports (specified at a later time). Keep this in mind during your table designs. Further, for the final system, there are different types of common users each with specific functionality (and authorization) that must be satisfied by the system at each phase even though some of the queries will be implemented in Phase 2.

Sample queries that may be required later: In addition to the workflow process, additional queries may be submitted to the system in order to generate specific reports. Some examples include:

- Generate total list of current students (by degree or by admit year).
- Generate a complete transcript (list of courses and the current GPA); although you will be implementing most of the functionality for this during Phase 1.
- For a faculty, find all courses they are teaching.

Users and Roles:

Observe that there are different categories of users of the REGS system, and each type of user has specific roles and authorizations.

- Systems administrator
 - Has access to everything and must create the different types of users
- Grad Secretary (GS)
 - Has complete access to current student's data. Note that they cannot create new users.
- Faculty Instructors
 - They can enter grades for the students in the courses they are teaching (i.e., courses for which they are the instructor) only once. Only the GS can override/change the grade.
- Graduate Students
 - They can view their enrollment information (such as courses taken and grades) and can register for courses (add/drop). They can update their personal information (address, email etc.) but no other information.

APPENDIX A

Course Catalog:

The system maintains a course catalog (i.e., the academic bulletin) which lists the courses by subject (i.e., department), course number, title, credit hours, and pre-requisites. No two courses in a department can have the same course number.

Course pre-requisite(s): Each course can have one main pre-requisite and one secondary pre-requisite. In a complete registration system, a student should not be able to register for a course if they have not taken ALL the pre-requisites for the course. For example, a student cannot register for CSCI 6286 if they have not taken CSCI6232 and CSCI6283 (which also implies they have taken CSCI 6212).

Below is the course catalog for the university (we use subject and dept interchangeably.) -- similar to what you find in the GW bulletin. Only a course that appears in the catalog can be scheduled during any one semester. This information must be stored in the system.

DEPT	Course Number	Title	Credits	Pre-requisite1	Pre-requisite 2
CSCI	6221	SW Paradigms	3	None	None
CSCI	6461	Computer Architecture	3	None	None
CSCI	6212	Algorithms	3	None	None

CSCI	6220	Machine Learning	3	None	None
CSCI	6232	Networks 1	3	None	None
CSCI	6233	Networks 2	3	CSCI 6232	None
CSCI	6241	Database 1	3	None	None
CSCI	6242	Database 2	3	CSCI 6241	None
CSCI	6246	Compilers	3	CSCI 6461	CSCI 6212
CSCI	6260	Multimedia	3	None	None
CSCI	6251	Cloud Computing	3	CSCI 6461	None
CSCI	6254	SW Engineering	3	CSCI 6221	None
CSCI	6262	Graphics 1	3	None	None
CSCI	6283	Security 1	3	CSCI 6212	None
CSCI	6284	Cryptography	3	CSCI 6212	None
CSCI	6286	Network Security	3	CSCI 6283	CSCI 6232
CSCI	6325	Algorithms 2	3	CSCI 6212	None
CSCI	6339	Embedded Systems	3	CSCI 6461	CSCI 6212
CSCI	6384	Cryptography 2	3	CSCI 6284	None
ECE	6241	Communication Theory	3	None	None
ECE	6242	Information Theory	2	None	None
MATH	6210	Logic	2	None	None

Course Schedule (Registration Data):

This list of courses must be included in your registration system. Only courses that are in the catalog can be scheduled. For each course, the system specifies a course ID, department/subject, course number, title and credit hours. In addition, since it is a 'schedule', the semester, day and time for each course must be specified; typically each course has a faculty instructor and a scalable design would permit multiple sections. You can assume/add any other information that you deem necessary (section, rooms, room capacity etc.). Note that the course number along with the semester and year will be unique; i.e., this combination (course ID + semester + year) is more like the CRN that you see on Banner where the CRN changes each semester for the same course. Question- Think of how you would implement designs that would allow multiple sections to be scheduled.

As a starting point, below is the schedule that must be implemented by your course scheduling "application for each semester. We are making a simplifying assumption here – that schedule is the same every semester. A more flexible/realistic system would allow different schedules. We are also working with just three time bands on each day – 1500–17:30 (3pm to 5:30pm), 1600–1830 (4pm-6:30pm), and 1800–20:30 (6pm to 8:30pm). *Question: What other information is usually provided in the class schedule ?*

CourseID	DEPT	Course Number	Title	Credit Hours	Day	Time
1	CSCI	6221	SW Paradigms	3	M	1500–1730
2	CSCI	6461	Computer Architecture	3	T	1500–1730
3	CSCI	6212	Algorithms	3	W	1500–1730
4	CSCI	6232	Networks 1	3	M	1800–2030
5	CSCI	6233	Networks 2	3	T	1800–2030
6	CSCI	6241	Database 1	3	W	1800–2030
7	CSCI	6242	Database 2	3	R	1800–2030
8	CSCI	6246	Compilers	3	T	1500–1730
9	CSCI	6251	Cloud Computing	3	M	1800–2030
10	CSCI	6254	SW Engineering	3	M	1530–1800
11	CSCI	6260	Multimedia	3	R	1800–2030
12	CSCI	6262	Graphics 1	3	W	1800–2030
13	CSCI	6283	Security 1	3	T	1800–2030
14	CSCI	6284	Cryptography	3	M	1800–2030
15	CSCI	6286	Network Security	3	W	1800–2030
16	CSCI	6384	Cryptography 2	3	W	1500–1730
17	ECE	6241	Communication Theory	3	M	1800–2030
18	ECE	6242	Information Theory	2	T	1800–2030
19	MATH	6210	Logic	2	W	1800-2030
20	CSCI	6339	Embedded Systems	3	R	1600--1830