CS 2541 Team Project
Advising System (ADS)
Phase 1

The ADS provides functions that help with advising and graduation requirements The REGS system provides a course registration system. Students enroll in courses at GWU using an online registration system to register for courses and their transcript is recorded in the database (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. The ADS provides functions that help with advising and graduation requirements. Each student fills out an online Form 1 which lists the courses they will take to meet graduation requirements. When the student applies for graduation, the system must check to see if all graduation requirements are met (i.e., the student has taken the courses listed on the Form 1 and met GPA and course requirements). Once they are met, and the student is cleared to graduate they are then added to an alumni list. The ADS must support graduation audit by students, faculty advisor, and Grad Secretary (GS), search for transcripts, and graduation of a student by the GS.

You will implement your system on the SEASCF machines using MySQL and PhP. 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 examine and evaluate your user interface. In Phase 1 the emphasis is on correctness but we will expect good user interface design methods applied to your project (such as user friendly system, input form checking, etc.).

Description of the ADS System

The ADS component provides some advising functions (for the student and advisor) including checking if degree requirements are met.

ADS 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.

- A graduate student in the university is assigned a faculty advisor by the GS.
- Each graduate student has an account that enables them to log into the system.
- A graduate student has personal information that identifies them.
    - Each student has a unique university ID (UID) which is a 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.
- We assume that the student has taken some courses, and the system stores course enrollment information for each student. This information includes 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.
- A student must specify their entire program of study by filling out a Form 1 and having a faculty advisor view the form. This lists the courses that they will take to meet the Degree requirements (this is somewhat similar to the curriculum sheets that undergraduates must follow to meet their degree requirements.). A sample of a Form 1 is provided in the Appendix in this document.
- After completing the requirements for the degree to which they are admitted, the student formally applies for graduation by visiting a URL, "Apply for Graduation", and entering their student number and selecting the degree to which they are applying. If you want to make a simplifying assumption for Phase 1, at risk of losing some points, then assume students are only applying for the MS degree.
- Since you will need to look up their enrollment information (transcript), assume that they have taken courses only from the course catalog provided in this document (in the Appendix).
    - Assume that the valid final grades are (A, A-, B+, B, B-, C+, C, F).Courses currently in progress show up with a grade of IP (in progress.
- Once a student has applied for graduation, the system performs an 'audit'. Specifically the system checks to see if the student has satisfied all the degree program requirements:
    - This requires that the system check the courses the student has taken and compare them with the program requirements (both course requirements and GPA requirements) and compares them with the courses the student listed on their Form 1. (For example, if they have taken a different set of courses than listed on their Form 1 then they will not be cleared for graduation). You could simplify the process by checking for program requirements when they submit their Form 1 -- i.e., check if the courses listed on their Form 1 meet the course requirements of the MS program; thus, the application for graduation will only test if they have filed a Form 1 and if they satisfy the GPA rule.
    - If you want to simplify the project in Phase 1 (at the risk of losing points), assume that only only MS students will apply for graduation).The program requirements for both the MS and PhD are provided in the appendix.
    - In general, program requirements for the degree (in your project, only the MS degree) are stored in the system. This will allow changes to the program requirements to be made if necessary.
- Once a student is cleared for graduation, the GS formally process their application and they "graduate". Note that a student can be cleared for graduation but they do not

actually graduate until the GS, or another authorized user, enters this information into the system and formally clears them.
    - The process of graduation must be automated; i.e., the GS need only check the "cleared for graduation" students and approve their graduation by clicking on some selection. (In practice the GS actually looks through their folder and transcript, and their accounts payable balance.)
- When a student "graduates" they are removed from the Graduate student table and their information must be entered into an Alumni table. Note that only a summary of their academic information should be kept in the Alumni table.
    - In a real system, the enrollment information for a student is not removed since they may re-enroll at GWU for another degree. Thus, a graduation process would only require that their data be tagged to indicate that they have graduated with a degree while keeping all their information intact.
- An alumni can only edit their personal information (such as email, address) by logging on to a URL using their student number (again, in practice it is better to give them a unique password rather than create privacy problems by using their student number.

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 statistics on total number of graduates, filtered by different parameters.
- Generate list of graduating students (select by year/semester or other?)
- Change advisor of student
- Search utilities to find advisees, alumni etc.

Users and Roles:

Observe that there are different categories of users of the ADS 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. They are responsible for assigning an advisor and for graduating a student. Note that they cannot create new users.
- Faculty advisors

    - These are faculty in the department and can review Form 1; for PhD students they have to approve (pass) the PhD thesis.
    - They can view their advisees transcript but cannot update the transcript. This is the only access they are given.
    -
- Graduate Students
    - They can view their enrollment information (such as courses taken and grades) but cannot update their grades. They enter the Form1 data, and can apply for graduation. They can update their personal information (address, email etc.) but no other information.
- Alumni: They can log into the system and edit their personal information only.

APPENDIX A: Degree Requirements and Course Catalog

MS Degree Requirements

To earn a MS in Computer Science at this university, a student must have satisfied all of the following requirements:

- Completed all 3 core courses required for MS: CSCI 6212, CSCI 6221, and CSCI 6461
- Minimum GPA of 3.0
- Completed at least 30 credit hours of coursework
- Taken at most 2 courses outside the CS department as part of the 30 credit hours of coursework
- Not more than 2 grades below B

(For Phase 1, you need to implement the MS degree 'audit' only.)

PhD Degree Requirement

- Minimum 3.5 GPA
- Completed at least 36 credit hours
- Taken at least 30 credits in CS
- Not more than one grade below B
- No required core courses.
- Pass thesis defense – approved by the advisor.

Suspension from the program: If a student has three grades below B then the student will be under an academic suspension.

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.) This data 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

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.) This data 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. 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 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 for each semester. We are making a simplifying assumption here – that schedule is the same very 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 ?*

Note: You do not have to implement a registration system; but your system needs to store course registration data (course, grade etc.) based on the table below.

CourseID	DEPT	CourseNumber	Title	CreditHours	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

APPENDIX B: Sample Form 1. For MS degree

Form1: Program of Study for MS in Computer Science

Please enter the courses you plan to take to earn your MS degree in Computer Science. You must enter at most 12 courses, and your Form 1 must meet the degree requirements.

Univ ID	**Last Name**	**First Name**
12345678	Coltrane	John
Courses In Program:	**DEPT/SUBJECT**	**CourseNumber**
	CSCI	6212
	CSCI	6221
	CSCI	6461
	CSCI	6232
	CSCI	6233
	CSCI	6283
	CSCI	6284
	CSCI	6286
	CSCI	6241
	CSCI	6242