CS 2541W Team Project Phase 2
Project ARG

You have been assigned to a 3 person team and tasked with delivering a specific product that requires integrating all three modules you have each built – the Application, Registration and Advising (ARG) system.

Project Integration Experience

In Phase 1, each team was assigned a specific module to be designed and implemented; specifically, you implemented an Application System (APPS), a Registration System (REGS), and an advising system (ADS). In Phase 2 of the project, you will integrate these different modules to provide the final product. As part of the team based design, integration, and testing we are essentially mimicking the design of a product by integrating different modules designed by different design teams. In Phase 2, the different components in the final system are no longer the three modules you built in Phase 1 – ideally, they should no longer look like three distinct modules when you deliver the product. Once again, you must implement your system using Amazon RDS in MySQL and Python. You will still need to refer to the Phase 1 specification document while you work through Phase 2. **Important:** *Please go over your Phase 1 project description again – there were features that were listed as optional in Phase 1 but required in Phase 2.*

What needs to be done in Phase 2

During Phase 1 you worked on three very different modules. In Phase 2 you should be working towards building the complete system (project) you have been assigned, but using the modules you built in Phase 1. Much of the functionality has been built in Phase 1, but you still have work left to complete the final project -- so plan your time carefully. Here are some specific things you need to be doing (this is only a partial list -- there are more things to be done as you will realize during this final phase) -- they are listed in a recommended order of completion.

- Reading: Before you start, each of you should read through the project description document for each module from Phase 1 and get your team member who built that module to explain it to the whole team. **The specifications listed on those project details page should serve as a minimum -- you need to use your basic CS and logical analysis/design skills to figure out other design decisions. Making unreasonable, or unrealistic, assumptions will cost you points.**
- Your Project Assignment: You have been assigned to a new team of three persons. Your project integrates all three modules and adds new functionality. You are responsible for integrating the module you worked on during Phase 1 into the final system; you

CANNOT collaborate outside of your group, as normal, which means you cannot consult with your Phase 1 team members. If you need to consult with them after that, please check with your mentor or one of the course instructors.

- Integration: The first thing you should all be working on (with the 'deadline' of one week) is on integrating the different modules to provide a skeleton of the complete system.
    - For example, the application system at the end admits an applicant. This application becomes a student once the student matriculates. A student matriculates when they (1) accept their admission and then (2) they send in their acceptance fee (deposit). You need to think of how a student can accept admission and send their deposit. The simplest case (this is the simplest possible solution you can provide - so think of other ways) is when the GS matriculates them, i.e., the student mails their acceptance fee as a check and the GS clicks on some button to make them into a student).
    - Once they become a student, they should be able to use the Registration and Advising systems – i.e., their data should now be available for the REGS and ADS systems.
    - The registration information (courses, grades, etc.) should be available to the ADS system. Be careful about your data -- you should not be replicating the data (some amount may be unavoidable)--it is the same student who is registering for courses as well as filling out Form 1, and their personal data comes from the APS modules.
        - Note: A complete redesign of a module built in Phase 1 is NOT a good solution and therefore will get points taken off.
        - Why do we require this: imagine that these are huge pieces of code written by someone who is no longer there -- in addition, it is possible that you may not have gotten the authorization to change the code).
- Check Workflow: Once you get an idea of how to best integrate your modules, go over (with a fine tooth comb!) the workflow of the entire system description as it pertains to the project you have been assigned. Identify what changes need to be made to the various components. Also take a look what did not work in Phase 1 (i.e., what the "evaluators" told you) and fix those problems first -- yes, you will get points taken off again for the same errors. Make sure your integrated system now meets the exact specifications of your final project.
- Design, Implement and Test new functionality/code: Once you check the workflow and the project specs, you will realize that new functionality needs to be built. For example, multiple online recommendation letters and advising forms (to lift registration hold) have to be built into the system now. In addition, specific Phase 2 reports/queries that MUST be provided for the client are listed here. You have to figure out which queries your "product" can support. Design and implement these new functions, and make sure you test them and verify that the workflow is met.
- User Interface: Once you have figured out everything that needs to be done to have the entire project working), you should take a critical look at your user interface. Work on developing an easy to use interface, and one that moves seamlessly from one application to the other. At the same time, make sure that an application cannot see the

student functions (and similarly with the other roles). Your final project evaluation will take into account your user interface design.
    - Make sure you do not use banners from commercial sites unless you have a license agreement-- this is a copyright infringement. Ideally, your main project page should state that this is a class project.
- Final Testing: After the Phase 1 demos, you have some idea of the testing that needs to be done. Before you come to the final demo, make sure you have run enough tests to catch at least the common errors. You should populate your database with a number of sample cases as described in the additional phase 2 submission instructions which will be posted in the next few days.
- Grading: The grading criteria will be similar but adds a new criteria to Phase 1 - i.e., implement correct workflow and all functionality specified in the project specification -- but we will also reserve up to 20% of the grade for "bells and whistles" and user interface design. For this 20% -- grade will be based on additional functionality you provide, effective software design, user interface. Essentially, the most competitive project is the "product" that we will be "buying".
- Phase 2 Queries and Reports: As part of the system, several report generation functions must be provided in the system. The minimum set of queries, and features, are specified later in this document.
- Other steps:
    - Report Writing: You will be required to submit a report of the formal descriptions of your final project -- the specifications, design, testing steps, and labor costs (do what you can on labor costs). This is part of the final project deliverables.
    - Teamwork: Each of the persons in a team is responsible for the module they built in Phase 1. We will use this for our grading. However, all team members must work together to ensure correct workflow and to help test the entire system.

The Application, Registration and Graduation System (ARG)

In this project your three person team is tasked with integrating the three modules from Phase 1 to provide a complete application-to-registration-to-graduation (ARG) system. The ARG system will process new student applicants, provide a course registration system, and provide advising services (by checking their degree status). Refer to your Phase 1 description for more details on each individual module.

The system stores information about graduate applicants to CS@GW, students currently enrolled in the department, alumni, faculty, administrative users (such as GS, Admin), and courses that the department offers. The client (i.e., instruction team and the client's payment will be in the form of a grade!) demands the following workflow (this is an outline of the workflow and details have to be fleshed out if needed):

1. A graduate applicant goes to the website and applies for admission to a graduate program. The department receives an application, and an entry is made into the

database (indicating the date the application was received, and the data for the student including academic data such as GRE, GPA etc.).
    - The system must provide an online recommendation letter submission system; i.e., applicants will enter names of up to three references and their email contact and the system will email the letter writers and check their submission. The letter submitted must be accessible by the faculty reviewers.
    - While you can still use the solution for how their transcript is submitted, you must add the option of an online transcript transmission if the applicant chooses that option.
        - An applicant will specify if the transcript will be mailed from their school OR they will enter the link to their school's registrar.
        - If they choose mailing of a transcript, then the GS will have to mark it as received (same as Phase 1).
        - If they choose an online submission, then a transcript will be emailed or submitted online (like the letters).
2. The graduate admissions committee (i.e., faculty) reviews the application and makes a recommendation, which can be Admit or Reject. An admit can be 'admit with aid' or 'admit without aid' – the admission date is also included in the data. We want this review process to be automated, wherein the faculty reviewer can enter their scores into a review form. The Grad Secretary (GS) (or Department Chair/ CAC) would then admit the applicant. The system must support multiple faculty reviews and an average review can be viewed by the CAC. Refer to Phase 1 description for more details. The graduate admissions committee reviews applications and makes a decision -- Admit, Admit with Aid, or Reject.
3. An applicant can check on the status of their application by going to the website and logging into the system. The status is "Application Received and Decision Pending", or "Application Materials Missing" or "Admission Decision: Accepted" or "Admission Decision: Rejected". Feedback should be provided on what is missing -- similar to Phase 1.
4. An admitted applicant can choose to accept the admission at GWU (i.e., matriculates) -- such an applicant would then become a current graduate student in our system and can register for courses. To decide on how a student will matriculate and who does what, refer to the discussion on integration earlier in this document. Note an applicant cannot enroll in courses unless they have matriculated. A matriculated student will be assigned a faculty advisor by the GS.
5. A student registers for courses using the registration system. All the requirements of the registration system are described in Phase 1.
    - The client has indicated that an additional feature of an "advising hold" would be desirable; the first time a student registers for courses (i.e., as soon as they are matriculated) they must submit an advising form listing the courses they will take and the advisor must "sign off" electronically on their advising form before they can register (to simplify matters, assume that this is not required for their second semester onwards). The client has indicated that the advising form can be the same as the Form 1 but more flexible models are preferred.

6. A current enrolled student must fill out a Form 1, and can apply for graduation at any time. The Form 1 must satisfy the requirements for the degree.
7. If cleared for graduation (the system does an ‘audit’ to check if the student meets degree requirements, and the GS clears the student if no other holds) the student is no longer a current graduate student and now becomes an alumni.
8. Alumni can log into the system to check their transcript and update their personal contact information, but cannot register for courses.

In addition to the workflow above, several functions (queries) and reports are specified by the client as a basic requirement. The client has indicated that additional features are desirable and will determine the final cost of the product.

Optional but cool additional feature: A user with the role of The Registrar works with the department to schedule courses. Courses can be scheduled only if they are in the catalog of courses (i.e., the bulletin). They can add or delete courses from the schedule of classes.

- Optional: The client is keen on having a system that will program in not just the time but the room and the room size (i.e., class capacity). During registration, the system must close the class if the room capacity has been reached.

Queries and Reports

Your system must support generation of reports (which may correspond to specific queries) which can be submitted by specific types of users. Note that the queries may be considered to belong to one (or more) of the modules from Phase 1 and apply to a class of users (though a user, such as GS had different functions in each Phase 1 module). The minimum set of reports/queries (you should consider adding additional relevant queries as your bells and whistles) are listed below and are essential to providing a complete system.

- Search for an applicant using their last name or student number. This query can be submitted by the GS or by a faculty reviewer.
- Update applicant’s academic and personal information – an applicant may choose to update their information at any time (or can be simplified by having only the GS perform this but this is not an ideal solution).
- An applicant or current student or Alumni should be able to update their personal information at any time. Think of which other types of users you want to give this privilege to.
- Given the Semester, or Year or Degree program, generate the list of graduate applicants. This query can be submitted by the GS.
- Given the semester or year or degree program, generate the list of admitted students. This query can be submitted by the GS.
- Given the semester or year or degree program, generate statistics such as total number of applicants, total number admitted, total number rejected, average GRE score for admitted applicants, etc. This query can be submitted by the GS.
- Given the semester or year or degree program, generate the list of graduating students (i.e., those cleared for graduation). This query can be submitted by the GS.

- Given the semester or year or degree program, generate the list of alumni and their email address. This query can be submitted by the GS.
- Generate a total list of current students (by degree or by admit year). This query can be submitted by the GS.
- Change the advisor for a student, given the student's student number. This query can be submitted by the GS.
- Given a student's student number, generate the transcript (list of courses, credits, and the current GPA). This query can be submitted by the GS or by the faculty advisor or by the student.
- For a faculty advisor, generate a list of all advisees. This query can be submitted by the GS or the faculty advisor.
- For a faculty instructor, generate their course roster (i.e., list of students enrolled in their class) given a specific course they are teaching.

Users and Roles

Observe that there are different categories of users (in both the systems), and each type of user has specific roles and authorizations:

- System administrator
    - Has access to everything and can create different types of users.
- Grad Secretary (GS)
    - Has complete access to applicant's data and to current student's data. They (1) can update status of applicant but they cannot review the application, (2) matriculate a student (i.e., changing an admitted applicant to a current student once the student enrolls at GW), and (3) clear a student for graduation. Note that they cannot create new users.
- Chair of Admissions Committee (CAC)
    - Is a faculty in the department.
    - Has complete access to applicant's data and can enter reviews, and makes final decision of admit or reject.
- Faculty Reviewers (including Chair of the admissions committee -- the CAC)
    - These are faculty in the department.
    - They can review the student's application – they have access to all the applicant's information. They enter their review into a review form which is stored in the system.
- Faculty Advisors
    - These are faculty in the department.
    - They can view their advisee's transcript but cannot update the transcript. This is the only access they are given to the transcript. They cannot view the transcript of students who are not their advisee.
    - They can view all information about an applicant but cannot enter a review.
    - They will need to lift the advising hold of a new graduate student.
- Faculty Instructors
    - These are faculty in the department

    - They can enter grades for the students in the courses they are teaching (i.e., courses for which they are the instructor).
    - Option: They can view transcript of students in their class (but not other students)
- (graduate student) Applicants.
    - They can enter their application form information, and can check on the status of their application. They can edit their personal/contact information. They are not permitted to perform any other functions.
- Current Graduate Students
    - They can view their transcript (but cannot update it), enter Form 1, register for classes, and can apply for graduation. They can update their personal information (address, email etc.) and optionally the Form 1 but no other information.
- Alumni
    - They can log into the system, view their transcript, and edit their personal information only.
- Registrar: (Optional)
    - They can create a new schedule of classes, i.e., they can add/delete classes from the schedule but the course must be in the course list (i.e., in the bulletin).
    - They can view grades and transcripts of current students.
    - They can update the grade for a student -- no other user (GS, Faculty, etc.) has this authorization.

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