OMSCS 6310 – Software Architecture & Design
Assignment #6 [150 points]: Course Management System – Group Design (v1)
Summer Term 2017 – Prof. Mark Moss
Due Date: Monday, July 17, 2017, 11:59 pm (AOE)

Submission:

- This assignment must be completed as part of a group.
- You must ensure that you have submitted the list of your group member's names to the OMSCS 6310 Team via Piazza before this beginning this assignment.
- Submit your answers via T-Square.
- You must notify us via a private post on Piazza BEFORE the Due Date if you are encountering difficulty submitting your project. You will not be penalized for situations where T-Square is not encountering significant technical problems. However, you must alert us before the Due Date – not well after the fact.

Scenario: The clients at the university are ready to enter the final phase of the project. During this phase, they are going to provide more information on how they would like you to structure the overall system. More specifically, they will provide details about the overall flow of the application, and how it will be employed by the administrators and students to manage instructor hiring and assignments, monitor enrollment requests, and assign grades. Your task will be to update your previous design to incorporate these new changes; and, to ensure that your design is consistent with the user's requirements and the systems as currently implemented.

Disclaimer: *This scenario has been developed solely for this course. Any similarities or differences between this scenario and any of the programs at Georgia Tech programs are purely coincidental.*

Deliverables: This assignment doesn't require you submit any specific deliverables: your main requirement is to coordinate and conduct an online meeting with your assigned TA, who will: (1) act as your project evaluator; and, (2) also be able to answer your questions from the client's perspective.

- You should plan to spend 45 minutes – 1 hour during your meeting. Requests for more time are at the discretion of your TA.
- You must conduct your meeting on or before the due date, unless you have received permission for a later date from your TA.
- Your team should spend the first 15 – 20 minutes explaining your general approach to completing this project; and, more specifically from an architecture and design perspective, how you believe that the new requirements introduced below will affect your design.
- Your team will be evaluated on the quality (clarity and completeness) of your presentation, along with your answers to the questions posed by your TA.
- Though you don't have to formally present any documents – class models, state charts, deployment diagrams, etc. – during this meeting, you are highly encouraged to share drafts of these documents. Don't be concerned that they aren't perfect – this is an excellent opportunity to get direct feedback from your TA.
- If you are going to share any documents, then you should provide an advance copy of them to your TA at least 48 hours before your scheduled presentation to help facilitate the discussion.

Client's Problem Description/Update: The clients are very satisfied with the functionality that you have developed so far, and they want to begin to assemble these capabilities into an interactive application for the administration and students. More specifically, they would like you to develop an application that will help them simulate the process of managing course offerings, enrollments and evaluations throughout the academic year. The application will support a continuous cycle of generating seat allocations for various courses; determining which student course requests can be granted based on those allocations and other policies (e.g. prerequisites); and, updating the student records with the most recent grades.

You've already developed solutions for much of this "core business logic" individually in Assignment #4, and your team will build on these efforts for the final project. Listed below are the key areas of new functionality that your team will need to develop and implement for your final, working system, along with some clarification of the existing functionality from the clients based on previous questions and discussions in the course. We might provide more specific technical details later; however, my intent is to provide at least enough details here so that you can really begin to develop your final system.

User Interface: Your system must provide a much more user-friendly interface than the current text-based command line interface in the previous assignments.

- You are still permitted to have a command line interface (CLI) in your system to support your development, maintenance, troubleshooting, etc.; however, you must support all of the functionality from the previous assignments, in addition to the new functionality mentioned above, via one or more graphical web- or application-based pages.
- We are giving your team fairly wide latitude in selecting a framework to support the development of your user interface, given that you are responsible for installing and configuring that framework on the course VM.
- At this point, plan on your interface having at least two distinct pages: one from the administrator's (or instructor's) viewpoint; and, a different page for the student's viewpoint. Each page should present the functions most common for that viewpoint, along with the data that would be relevant and timely. Also, (tentatively) you are welcome to include other pages as well if it makes sense for your design.
- Your state charts will be based on the operation of these interface pages. The intent is that your interface pages will provide a solid "target" for your state charts, similar to the examples based on wristwatch interfaces as mentioned in the Udacity lectures, etc.

Error Checking: Given that your final system is intended to provide an interface for human administrators and students, the clients would like you to incorporate more robust error and warning messages into the system, as discussed here:

- You should create a ***course_report*** instruction that displays whether or not a course is being offered during the term, along with the number of remaining available seats if appropriate.
- When a ***request_course*** instruction is denied because of missing prerequisites, then your system must display an appropriate message that provides more detail than the current "missing

prerequisites" message; in particular, it must provide (at a minimum) a list of all of the missing prerequisites.
- Also, ensure that all of the instructor, student and course IDs referenced from instructions are valid, or display messages that alert the users and prevent null pointer errors, etc.

Data Persistence: Your final system will eventually manage multiple years of data. Therefore, your team must implement a data persistence solution.
- The recommended system is MySQL, which is already installed on the course VM. Other persistence mechanisms will also be supported – ask your TA during the meeting if you're concerned/unsure about whether or not a specific system will be accepted.
- Using simple, plain-text files will not be accepted as a viable solution.
- For instructors, your system must save all of the courses offered by an instructor for each term. This information must be tagged (enriched) with the term and year data to support later analysis.
- For students, your system must save all of the courses taken and grades for each student. This information must also be tagged with the term and year enrolled to support later analysis.
- As an added capability to support the ***request_course*** function, your system must maintain an ordered wait list (first come, first served) of students that have made otherwise valid course requests (e.g. the course is being offered, and the student has satisfied all of the prerequisites) but been denied because of a lack of available seats.
- Your system must maintain this wait list across terms; and, if more seats are allocated to the current course (or to a new session of the course during a later term), then your system must fill seats from the wait list before accepting other, newer course requests.

Data Analysis: The clients want to use your system to support long-term analysis of the collected data. This will aid them in making better decisions about instructor hiring, offering courses, etc. The external support system that you must implement to support these types of analysis is a data mining system named Weka, which should already be installed on the course VM.
- Your system must use Weka to access the collected data, and perform some basic summary and trend analysis of the instructor allocations and student requests.
- Your system will need to translate the data from the internal format you use for storage into the ARFF format, which is used by Weka. ARFF format is very similar to a basic CSV format which uses headers (@name, @attributes and @data) to define and partition the file contents.
- Your system must leverage one of the Weka capabilities – for example, classification, clustering, frequent itemset analysis, etc. – to provide some analysis of the data to the clients. Please feel free to research Weka (and the ARFF format) on the web to consider which kind of functionality your team would like to implement. Weka capabilities can normally be invoked by saving the data to be analyzed to a file, and then using a system call from within Java to access the Weka CLI.
- I understand that this is not a data mining course, and I don't expect your data analysis or Weka output to be overly-complicated or technically deep. One intent of this requirement is to allow your team explore how interacting with an external support system affects your architecture.

Closing Comments & Suggestions: This is the information that has been provided by the client so far. The meeting with your TA is intended to be a great opportunity not only to present your design

plans going forward, but also for you to get some of your questions and concerns answered and clarified. We (the OMSCS 6310 Team) will also conduct Office Hours where you will be permitted to ask us questions in order to further clarify the client's intent, etc. The clients might also add, update, and possibly remove some of the requirement details, but the overall requirements are relatively fixed at this point. One of your main tasks will be to ensure that your architectural documents and related artifacts remain consistent with the problem requirements – and with your implementations – over time.

Quick Reminder on Collaborating with Others: Since this is a group project, you may (and should) communicate freely with all of your group members. However, your group should not communicate with any other groups while working on this project. Please use Piazza for your questions and/or comments, and post publicly whenever it is appropriate. If your questions or comments contain information that specifically provides an answer for some part of the assignment, then please make your post private (your group members and OMSCS6310 TAs/Instructors only) first, and we (the OMSCS 6310 Team) will review it and decide if it is suitable to be shared with the larger class.

Best of luck on to you this assignment, and please contact us if you have questions or concerns.
Mark