

Supplement to SIS's Academic Requirements System

A Technical Report
presented to the faculty of the
School of Engineering and Applied Science
University of Virginia

by

Noah Dela Rosa

May 5, 2021

On my honor as a University student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments.

Noah Dela Rosa

Technical advisor: Daniel G. Graham, Department of Computer Science

Abstract

I propose to make a new tool, namely a web application that seeks to improve upon the way students at UVA can track their progression through their major. This tool seeks to address deficiencies in UVA's Student Information System (SIS), namely: poor user interface design, which makes navigating curriculum requirements more difficult than necessary, and linking the functionality of a schedule builder to provide students a more easily accessible way to plan for their next semester. The motive behind creating this tool came from informal talks with other students who had their frustrations with the current way SIS displays academic requirements. I plan to use page scraping (of likely both students' Academic requirements and Lou's List) and the fundamentals of programming websites to create an intuitive mode of staying on top of one's required courses.

Introduction

Despite the recent overhaul of the user interface of SIS, the "Academic Requirements Report" page hasn't received the same treatment. The report page produced by SIS is cluttered and lacks the ease of use and intuitiveness such a critical support system should have. While it becomes easier to navigate the current system as you progress through your major, due to completed sections of the curriculum automatically collapsing upon completion of all section requirements; it can be an immense challenge for first-years or even second-years to begin piecing together an understanding of the courses they need to take.

With the current system, students are strictly presented with the requirements for their degree, broken down into several categories. For example, as a student majoring in Computer Science through the School of Engineering, such sections included are: "Engineering Undergraduate First

Semester Required Courses”, “Computer Science Lower Level Required Course”, and “Applied Mathematics Courses”. This barebones representation of academic requirements makes the process of developing a plan for future semesters take longer than it should. Furthermore, the current solution does not make readily available resources that could expedite the planning process easier, such as undergraduate handbooks that some major departments provide to their students. The lack of such resources reveals at first glance what can be assumed to be a lack of connection between SIS admins and major departments.

Background

This project relates to the fields of web design and development or more generally, user experience engineering. This project was built using HTML, CSS (primarily through the Bootstrap 4 framework), PHP, and a locally hosted MySQL database.

Related work

There is no system that UVA uses to compute academic requirements other than SIS, UVA does provide a schedule builder that allows students to select up to ten courses for a semester and possible combinations of those classes are provided. Since students can choose which courses they’d like to see combinations of in different schedules, this tool performs the job nearly perfectly. However, this tool can be further personalized by including the option to import courses from a list of their required courses they are eligible to take. Slight improvements can be made to the system as well such as specific error messages, rather than “There are no schedules to display. Try rebuilding schedules”. Specific error messages could include which courses are giving the tool issues (i.e. time conflicts between two classes, a certain class isn’t being offered

that semester, etc.). To be clear, my system would by no means overwrite the current schedule builder in SIS, rather, it would improve upon the existing system as well as directly link it to the academic requirements report.

System design

This project was designed as its standalone system apart from SIS, though a final implementation would include its integration into SIS overall. The front-end of the system was designed and constructed from HTML and CSS while the back-end consists of PHP and a locally hosted MySQL database.

Some of the design challenges faced when constructing this implementation of the system were how to read the SIS-generated academic requirements report for individualized students' curriculum needs, safety considerations for students' private information (i.e., course grades, SIS ID, GPA, full name, major, etc.), and how to better display and stratify academic needs.

One of the biggest design challenges was figuring out how to identify the needs of a student based on their personally generated academic requirements report. Due to my implementation of this system being done solely by myself, I didn't have the time to implement this portion of the system as I'd wanted, but my initial thoughts were to implement this through scraping the PDF for tables of requirements that have been satisfied, as well as those that still require completion. The implementation of this would scrape the PDF and look for tables of courses used to satisfy certain requirements. Then, to get the requirements that the student still needs, the system would have to search the PDF for lines such as this:

Computer Science Capstone

Not Satisfied:

· **Units: 3.00 required, 0.00 used, 3.00 needed**

The tool would then identify the string that starts with “· Units” and check if the number of needed credits for the section isn’t zero. If so, this section “Computer Science Capstone” would be appended to the list of other sections/courses the student still needs to complete.

The next big challenge was the issue of information security, more specifically, how to prevent the tool and/or malicious users from accessing data they don’t need or don’t have the authorization to see. Since students upload their academic requirements report PDF, the system ought to destroy the file after all of the necessary information has been extracted. This way, no malicious user could attempt to hack into a database of these files and gain access to a student’s entire academic history here at UVA. Furthermore, when extracting the student’s information from the PDF, specifications in the program would designate the following columns of each table not to be saved: “Grade”, “Type”, and “Additional Information”.

Lastly came the challenge of how to better display the academic requirements than the method currently used in SIS, a series of collapsible/expandable sections. In my partial implementation, my improvement upon the current solution is to separate requirements that have been completed and those that still need to be accomplished. When displaying these two sections, those courses that still need to be taken are shown first as they are typically what students need to know.

Procedure

To use this system, students must either sign up for an account or log in if they have already made one. Then from the home screen, the call-to-action changes and tells the student to upload the PDF of their academic requirements report. Upon successful upload, the student is automatically redirected to a page that details the summary of the classes they’ve taken and those

they still need to take (this is where the personalized computation comes in). Then from that page, the student can also view a page that would detail possible schedules for the next semester which are populated by the most immediate requirements that the student can satisfy. If the classes displayed aren't to the student's liking, they have the option to choose different classes than the ones currently populating the schedule.

Results

Since this new system is only a partial implementation, the results detailed here only hint toward how well this solution remedies the deficiencies in SIS's current form. When comparing this proposed system versus the way SIS currently accomplishes the task, students were pleased with how readily available pertinent information was displayed. One student said, "[they] especially liked how clean the page was. When I try and use SIS to do this it's easy to miss details because of the small font and messy organization." Another student also commented that while they prefer the schedule builder in SIS, they wished that it had an option to import classes from the list of requirements they still need to meet. The overall reception among test users was positive. Most users cited their frustrations with SIS as being primarily from a "messy interface". Suggestions from test users on other ways to display this information were to display the requirements as in SIS through collapsible sections of required courses but add better spacing between each or to implement an entire page strictly to those requirements that still need to be fulfilled.

Conclusions

A system was designed to solve mutual feelings of frustration with the current way UVA's SIS prepares and displays a student's academic requirements. The system sought to remedy these frustrations through two means (a) a cleaner, more organized method of displaying academic requirements, both satisfied and unsatisfied, and (b) combining this system with a schedule builder for future semesters, based on either student-selected courses or requirements the student can immediately fulfill.

When test users used the implementation of the proposed system, the reception was overall positive. Students said they were better able to see their requirements and the interface of the new system gave way to an easier and more intuitive experience when determining their requirements to determine the courses they'd like to take in the future. From this new system, we were able to pinpoint some of the misgivings that students have with SIS currently. These sources of frustration were due to SIS's "messy interface" which reinforce the need for such a supplemental system to SIS.

Future work

With more time, a more complete implementation of the proposed system could have been developed. Features that the system would benefit from that were not included in this implementation include functionality for advisors to see the requirements and other information for each of their advisees, support for PDF scraping discussed above, as well as pages that include resources specific to a student's major such as samples of course schedules for eight semesters and major-specific undergraduate handbooks which are provided usually by each major department.