CS486C – Senior Capstone Design in Computer Science

Project Description

Project Title: Teaching Assistant Scheduling and Management System	
Sponsor Information:	Viacheslav Fofanov, Ph.D. Associate Professor Viacheslav.Fofanov@nau.edu School of Informatics, Computing, and Cyber Systems Northern Arizona University Flagstaff, AZ

Project Overview:

Practicum labs are a staple of modern undergraduate pedagogical experience. These labs typically provide a small-group learning environment that supplements larger lectures, allowing for hands-on, practical exposure to the subject matter with greater instructor / student interaction. In computer and engineering sciences, labs are usually taught by Teaching Assistants (TAs), which are typically graduate students that are closely supervised by the primary instructor for the course. When done correctly, labs are staffed by TAs that are both knowledgeable and passionate about the subject matter, with relatively recent experiences in learning the subject (and thus a keen personal perspective on where the common learning bottlenecks are), and paid, dedicated time to devote to one-on-one and small group teaching.

Academic units hosting degree programs with a significant number of lab courses attached typically have a designated *Lab Coordinator*, a faculty member who is tasked with the difficult challenge of staffing and providing overall supervision for the unit's lab courses. As the number of lab courses to schedule increases, the complexity of finding optimal staffing for the unit's lab courses increases exponentially. In essence, the problem is a multi-dimensional constraint resolution and optimization problem: there is a pool of potential TAs available in the unit, each with certain background, skills, experience levels…and, of course, their own unique scheduling commitments (TAs take classes too!). On the other side, there is a pool of lab courses to staff, each scheduled at certain times, and requiring certain levels of proficiency in some set of skills required to teach the lab. The job of the Lab Coordinator is to identify the optimal TA for each lab, taking into consideration the TA's qualifications, previous lab leadership experiences (and student reviews of those experiences) and, of course, their availability schedule. As courses and numbers of TAs in the pool grow, it rapidly becomes extremely challenging to find *any* workable staffing solution, much less finding one that optimally deploys TAs, i.e., a maximum number of TAs are assigned to labs they are maximally qualified, interested, and experienced in teaching. As a result, staffing labs in any sizeable unit in a "manual" fashion is time-consuming, frustrating, and error-prone. Furthermore, once the staffing plan is created, any changes to GTA availability can have significant cascading effects, disrupting multiple labs and staffing assignments; in the worst case, the staffing problem must be solved anew from scratch.

Solution Vision: A web-based tool for automating and optimizing course staffing assignments.

What is needed is a sophisticated, secure web application that (a) engages TAs to spread the effort of entering/updating their skills and availability data; (b) supports a database of courses to be covered, along with some characterization of expertise and experience (both minimum and "desired") needed to lead the course; and (c) provides a powerful graphical interface to allow a Lab Coordinator to easily maintain an overview and quickly schedule courses. A basic version would focus on *assisting* a human Lab Coordinator in solving the staffing problem, e.g., by showing calendared courses, assigned/un-assigned TAs, and TAs the could teach each course in a simple calendar layout. A more sophisticated solution would add some level of automation, using a variety of techniques (hill-climbing, Monte Carlo, etc.) to explore the "search space" of possible TA assignments and suggest possible solutions that meet/optimize all constraints. Specifically, we envision a secure web2.0 web application driven by a well-designed GUI interface that includes the following features:

<u>The bare basics: Minimum usable product:</u>

- A secure web application, including a user management interface with role-based permissions.
- Allows system administrator to create new account shells for TA users.
- TA users can login in, change password, and update their profiles: expertise in various (pre-determined areas), weekly availability schedule, etc.
- Lab Organizer (Sys admin) can define all courses that are taught by TAs in the unit, and associated (minimal/desired) levels of skill/experience needed for each.
- Lab Organizer can create a new "Semester" object, and populate that semester by "instantiating" offerings of courses being offered that term.
- Lab Organizer can populate a Semester object by selecting from the TA pool to designate which TAs are active and available that term.
- A graphical interface (e.g. calendar view) that lays out the courses to be covered, and quickly shows which TAs might be available for each. Lab Organizer can manually "assign" TAs to courses…and watch the layout update to see rippled implications. This should allow rapid "assisted manual" development of a complete staffing plan.

<u>A complete, usable solution would add:</u>

- Dashboard views that allow Lab Organizer to quickly overview current and past Semesters, view the current TA pool and status of each (i.e. have they completed all of their profile information), and can easily overview courses in a semester to see status (staffing).
- An algorithm to optimize a staffing plan based on a set of user-definable requirements (non-breakable rules) and preferences (rules that incur penalties when violated).
- An algorithm to optimize staff replacement (e.g. when GTA is unable to perform their duties) that will minimize disruptions to the existing staffing plan.
- Ability to engage multiple faculty, staff, and administrators in the TA staffing tool. Would involve defining various permission levels, e.g., "Lab Organizer", "Schedule-viewer", "TA editor", etc.
- A more efficient account creation system: system administrator can create new accounts by entering emails (individually or bulk); those users are sent account creation links (basically one-time tokens) that allow them to create their own accounts.
- Lab Organizer can easily review TA performance, both by semester, or by overview of whole history for a TA. Graphical tools to visually summarize and juxtapose terms would be a bonus.
- More sophisticated settings screens for system administrator, e.g., allows easy graphical management of the constraints, expertise areas associated with various courses, and other settings.
- Provides easy way to view experience and teaching history for each TA. Automatically updates this history based on each semester's course assignments.
- A "course evaluation" mechanism: allows course evaluations to be configured (might connect to Qualtrics surveys to avoid reinventing a wheel). When opened, TAs see a link they can send to their course students to take the survey. Students log in using their NAU accounts; the system ensures that each login submits a single survey. Results are returned, stored for the course offering and appear in the TA's teaching record.
- Basis for expanding access and/or monetization: Add an additional, enclosing layer to the architecture, so that, at the top level, system admins can create new "Departments", along with a "local administrative user"

for each (i.e. the Lab Coordinator for that department). That Lab Coordinator can then do all of the above for his/her own department. Note that all courses/users/data for each such "department" must be carefully and securely be kept separate from all other departments to avoid inadvertent or malicious data exposure.

Additional features: Stretch goals

- Allow GTAs to interact with the scheduling system if they wish to swap / modify their assignments, within the rules established by staffing plan creator
- An ability to export schedules, and supporting information into formats chosen by user.

Impact of a successful solution. Lab courses are a common feature in just about every STEM degree program across all universities in the U.S., and even globally. From Biology, to Geology, to Engineering, to Health Sciences, lab courses provide critical hands-on learning experiences for future graduates. Although we have focused here on staffing lab courses taught by TAs, it should be clear that the system could easily be used for the more general course staffing problem faced by large academic units everywhere. A successful software solution produced by this project would therefore have an enormous potential market, saving literally tens of thousands of hours of efforts and many thousands of scheduling errors.

Knowledge, skills, and expertise required for this project:

- Programming and software development skills for modern web interfaces and databases.
- Algorithm development for scheduling optimization problems
- Development of graphical user interfaces in web-based systems, including end-user testing/refinement.

Equipment Requirements:

- There should be no equipment or software required other than a development platform and software/tools freely available online.
- Client will provide basic algorithm and rules for staffing optimization algorithm

Software and other Deliverables:

- A complete software product, fully tested, installed and demonstrated on a server platform of client's choice
- A strong as-built report detailing the design and implementation of the product in a complete, clear and professional manner. This document should provide a strong basis for future development of the product.
- Users guide to running the code, appropriate for someone with little software background.
- Complete professionally-documented codebase, delivered both as a repository in GitHub, BitBucket, or some other version control repository; and as a physical archive on a USB drive.