

S&DS 662 Final Project - ULA App

Katherine Brumberg, Maria Gargiulo, Margaret Luo, Cameron Pringle, Jon Seyhun

2018-05-07

Overview

This year, the Yale S&DS Department adopted the Undergraduate Learning Assistant (ULA) program in many of its classrooms. Students serving as ULAs have both grading and teach responsibilities and serve in very similar capacities to the department's graduate student TAs and TFs. One problem currently facing the ULA program is the fact that no system currently exists to organize the recruitment and class assignment processes, which is frustrating for students applying to be ULAs, faculty hoping to have ULAs in their classrooms, and administrators who oversee the process and hire students. As more and more departments adopt the ULA program there will be a greater need for an organized (and potentially centralized) system to manage recruitment and assignment. This Shiny application is a first attempt at doing just that.¹

The structure of this project is divided into 5 main parts: a student interface, a faculty interface, a stable matching algorithm, an administrator interface, and a notification portal. The student interface allows students interested in serving as ULAs to provide some basic information about themselves and the courses they've taken, and rank their preferences in terms of the classes that they would like to ULA. This information (minus preferences) feeds into the faculty interface, allowing faculty members to rank which students they would like to be their ULAs. The rankings from both of these interfaces will be fed into some stable matching algorithm and the resulting pairings will be displayed on the administrator interface where they can be approved or manually changed, if necessary. After administrators finalize assignments, students will be able to log back in and accept or reject their offer.

Current Progress

Katherine

- Created log in functionality for both students and faculty. For students, this loads in their previous responses. For faculty, this creates a tab for each class they are teaching and loads in the information from the students who ranked that class and enables the faculty to rank those students. I also added error messages for the various log in failures.
- Helped Margaret and Cameron set up the inputs in the student ranking table so that they are interpreted by RShiny correctly and also made it possible to enable and disable inputs based on previous inputs.
- Set up a ranking system for faculty where there are drop down lists of students for each ranking that update so they only include the students who have not yet been ranked.
- Created a summary page for faculty which displays their rankings and number of desired ULAs.
- Created submit functionality on faculty page that writes to a CSV.
- Helped debug and merge other peoples' code together.
- Helped plan the course of the project, adding issues to the git as necessary to guide peoples' work.

Maria

- Set up basic initial structure for the student side of the application by creating the tabs that are used to collect student meta-data, record their course preferences, and summarize their inputs and set up all

¹You can see our current progress and test the app for yourself at: <https://github.com/kkbrum/ULA-app>

numeric and text fields on the meta-data tab.

- Created basic error checks for all tabs of the student side of the application to ensure that responses are non-blank and follow the correct input format (ex. 4-digit pin is actually 4 digits.) When input is not correct students see an error message or warning and they may only submit their application, which writes to their personal preferences csv, when all input issues have been resolved.
- Built the summary tab with Jon. In particular, I made the code more concise and provided two HTML formatting options depending on whether or not a particular field had a proper response.
- Helped debug and merge other peoples' code together.
- Helped plan the course of the project by contacting administrators, creating an outline of the general structure of the app, organizing team meetings, and adding issues to the git as necessary to guide peoples' work.

Margaret

- Worked on the student's preferences page with Cameron. This involved displaying the courses available as a table (information pulled from a csv) and creating a selection button below to allow students to choose the classes they would like to ULA. Once the student submits their course selections, a ranking table is created with the correct number of rows for the number of courses selected.
- With Katherine's help, built a student information input table that collected data in the correct form (numeric for what year they've taken the class/preference ranking, character for typed in inputs, etc.) and correct ranges of values for numeric inputs. I also worked on adjusting the enabling of inputs - students must first select whether or not they have taken a course before being able to fill out any further information on the table.
- A feature to be added is an autofill feature (the same as the Student meta-data page) where if a student signs in, all of the information a previous session that was saved to a csv is pulled into the App and automatically fills in the Student Preference page.

Cameron

- Worked with Margaret on the Preferences tab to allow students to submit their course choices.
- Created 'error checkers' to ensure the sheet was filled in appropriately. This involved taking the input and making sure that certain tabs were non empty and giving specific error messages that corresponded to the input problem.
- May need to add a different table to display the courses once the list of courses increases. At the moment the current table will display options as a long list. This will cause issues with space when more courses are added.

Jon

- Built the 'summary' tab with Maria.
- Contributed to the 'summary' tab by going through the 'my info' and 'course preferences' tabs and incorporating for each field various checkpoints which need to be completed before the student is informed in the 'summary' tab that their response in the particular field is sufficient.
- General error checking and adding functionality for aesthetics and to improve smoothness of operation for the user.
- Examples include: Displaying an error message if 'select' is pressed without and courses selected, requiring course input to be populated before the user can proceed through the 'course selection' tab, informing users that all fields in the 'my info' tab are required, and others.
- Contributed to the 'issues' page with functionality that needs to be added in the future.

Future Work

The student and faculty interfaces have all major functionalities in working order, so further work done on either of these parts will be to improve efficiency or make the application more aesthetically pleasing. The next steps in this project are figuring out the details of the stable-matching algorithm that will be used to assign student ULAs to classes, creating a visual display to communicate matchings on the administrative interface, and to create an interface where students can log in to view and accept or reject their final assignment. These pieces will then need to be deployed to the S&DS Department server where they will be accessed and maintained.