Team riteSnicKerz— Bayan Berri, Terry Guan, Yuyang Zhang, Brian Leung

SoftDev2 pd 07

Final Project

2018-05-15

Project Daedalus:
Better Course Selection and Program Changes

I. Brief Description

Course selection has been known to be a chaotic and unpredictable process. Many mistakes are typically made in the process of selecting students: lack of a lunch period, core class, physical education class, automatic removal from a given spot in a class, and losing a spot in an AP to a student who is less qualified.

Program changes are even more chaotic as students have to wait on lines that can stretch throughout the building, obtain signatures from multiple individuals, find students who want to swap periods, etc. Our project attempts to address this situation by building an all-in-one course selection and program changes website. We attempt to automate student selection into courses based on factors like overall average, subject average, recommendation, etc. By automating this process we remove many possibilities for human error. The online system will also diminish the need to be physically present at program changes.

II. Detailed outline

Course selection will be done by evaluating every student who applied to a specific course using a weight system that uses these factors: overall average, subject average, and teacher recommendation. Each student will be given a number based on the weight system. The highest ranked students will be prioritized and put into a specified course.

Program changes will be done online. Every student can submit a request to be put into a specific class with a valid reason. The assistant principal (or the admin) will be given the students' information (transcript, OSIS, Stuy number, etc) and a reason in a FIFO order (this produces the effect of first come first serve: if a person submits it earlier then his or her application will be shown first and read first). Students who are waitlisted and fail to put in a request will not be considered for the class. This is to demonstrate continued interest in the course. Waitlisted students who do not show interest won't be put in the course.

There will still remain administrative powers to override the system to address special circumstances. For example, an assistant principal can allow a more advanced student skip courses (like a freshman who can skip to trigonometry instead of taking geometry).

When the user first enters our site they will be given the option to log in or view our about us page (Students cannot register because the Admin accounts must make accounts when students enter as freshman).

The About us page will just give some background information on the website and how it works. We will have both admin accounts and student accounts.

If a user logs into an admin account they will be able to do a variety of things. For one, the admin is the one who can create student accounts. We don’t really want to give any one the ability to create accounts— this limits it to only Stuy students. Admins can also view specific students, courses, and a list of program change requests. Admins include assistant principals and guidance counselors (the people who usually sign off on program changes). The Admins go through their inbox which is, once again, in the form of a stack and approves the different requests. If the request is not possible to meet they will have some sort of alert (In the case of a full class). However, they won’t be completely dismissed. Down the line, there is probably someone else who wants to drop the class that’s full. In cases like these, the admin can revisit old cases and make sure to accept the requests. Admins can also change their passwords and view their account settings.

A student account is allowed to do multiple things as well. For one it allows students to submit program change requests. The student can also select courses for the next semester. Course selections will be similar to the google form that we used to fill out in the past. It will ask for two elective choices, a math class, an english elective if they’re upcoming juniors/seniors etc (For the MVP, we won’t have that many things going on. Probably just one elective and a gym elective or something simple). If the application for course selection hasn’t been released yet and it’s not program changes season, the student can view their schedule. Other than that, they can change their password and view account settings as well.

We plan to use databases throughout our project quite often. It’s the best way to keep track of a lot of these details such as schedules, grades, transcripts, etc. (See database schema below)

III. Component map

IV. Site map

V. Database Schema: {MongoDB/(Any Relational DB) Document style]

Student db pseudo schema

- OSIS
- Email
- Background information
- SCHEDULE
- Overall average
- All subject averages
- Pre reqs
- Graduation requirements
- Religious preference
- Year
- Recommendations
- Class selection preference

Class pseudo schema

- Max number of people
- Teachers teaching the class
- Basic Info (Course code, name, description, room)
- Waitlist
- Pre Reqs needed to be filled
- Applicant Pool and/or Queue

VI. Division of tasks

Bayan:

- Project Manager
- Second frontend handler

Terry:

- Database handling
    - Providing difference between admin accounts and student accounts
- Leading combination of frontend and backend
- Python data handling (from database)

Yuyang:

- Main frontend handler: JS, CSS, HTML and any necessary jinja templating
- Work on creating Admin accounts

Brian:

- Python routing
- Python data handling (from database)
- Work on creating Admin accounts
- Snake Handler