

## AP/Elective Selection Website Project

### Overview

As requested by the Programming Office, Team Team Platek Prophit (TTPP for short) will be creating a website on which students will be able to make AP course selections. The courses available for selection will be individualized for each student based on the prerequisites met.

Administrators will be able to setup the database with information about students and courses by uploading files, either in CSV or XLSX format [TBD]. After initially setting up the database, administrators can update it through the user-friendly administrator interface. They can add, remove, or modify information for an individual student or course or add or remove an entire cohort of students.

We will first focus on developing a site to support AP Selection, then move onto elective selection if time permits.

### Project Roles

- Project Manager: Reo
- Database Master: Haley
- Framework Boss: Amy
- Javascript Wizard: Vanna

### Project Components (blue for administrative pages, purple for student pages)

- **app.py**
  - Routes the HTML pages.
  - Acts as main liaison between frontend and backend.
  - Admin ability to shutdown system.
  - Login authentication and sessions code for admins and students.
- **/utils/csvparser.py**
  - Takes data (probably as CSV files) and processes them to add to the database.
  - Parses through student transcripts to build database.
  - For each student, calculate overall average, average for each department

- **/utils/requirements.py**
  - Write tests to see if students qualify for courses based on grades and prereqs.
- **/utils/login.py**
  - Contains functions for:
    - creating new accounts (hashing sensitive data and storing in database).
    - checking whether a user's inputted username and password matches database data.
- **/utils/database.py**
  - Find and return which APs students can take based on prerequisites (uses requirements.py).
  - [Extended functionality] Find and display which electives students can take based on prerequisites and space left in schedule; save export this information, as well.
  - Export database information (in form to be determined by Mr. Cheng)
  - Add, remove, or modify individual classes and students.
  - Add or remove cohorts.
- **/templates/base.html**
  - Contains the flask template which the other pages will inherit from. Will contain elements that recur on every page like navigation bars .
- **/templates/adHome.html and /templates/stuHome.html**
  - The respective home pages of an administrator and student. Also where students will see the option to sign up for courses.
- **/templates/adLogin.html and /templates/stuLogin.html**
  - The respective login pages of the administrator(s) and students. The two categories of users have their own distinct login page to mitigate the issue of students logging in once course selection has been shut down.
- **/templates/add.html**
  - Allows administrator add an entire cohort, an individual student, or an individual course, respectively.
- **/templates/remove.html**
  - Allows administrator remove an entire cohort, an individual student, or an individual course, respectively.
- **/templates/modifyCourse**
  - Allows administrator to add, remove, or modify prerequisites for existing courses.
- **/static/style.css**
  - Overarching style sheet to be applied to most/all pages

- **/static/script.js**
  - Javascript function/event listeners suitable for most pages -- used to make AJAX calls and for dynamic design (page alteration without reloading)

## **Tentative Schedule**

Tuesday, 5/16

- Set up basic Flask app -- all requisite HTML files and simple login system

Friday, 5/19

- Implement transcript file reading functionality + calculate averages

Tuesday, 5/23

- Storing and modification of student information based on both administrative and student input
- Processing student information based on course requirements/constraints + display appropriate form that reflects student info, course limit, etc.

Friday, 5/26

- Storing inputted information from course selection form in database
- Add course selection form "on/off" feature

Tuesday, 5/30

- Fine tuning core Javascript functionality

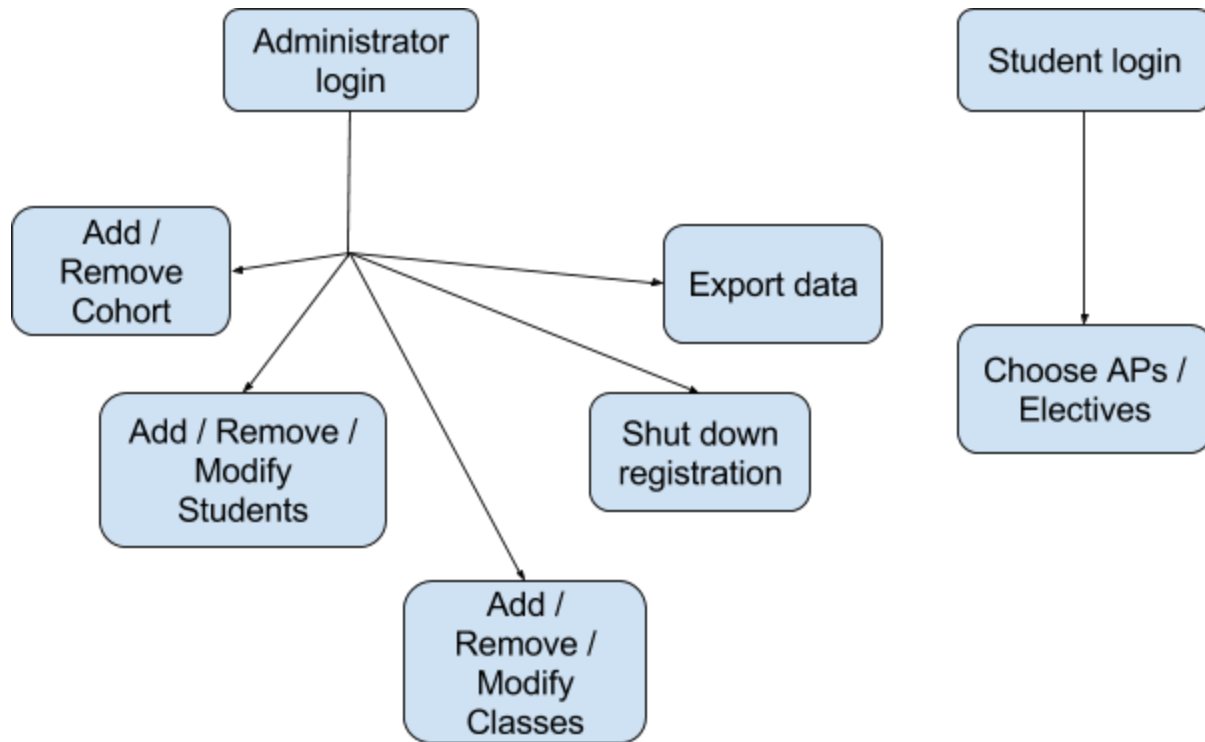
Friday, 6/2

- Styling!

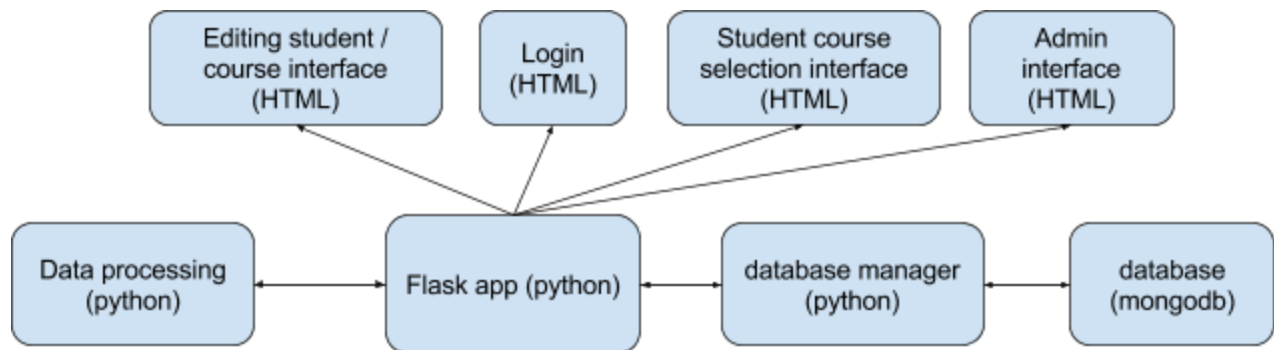
Tuesday, 6/6

- Implement reach goals (e.g. modal login) if time permits

## Site Map



## Component Map



## Database Schema

MongoDB

Collections: students, courses, admin accounts, site status

### students

- OSIS (string)
- First name (string)
- Last name (string)
- Middle name (string)
- stuy.edu email (string)
- Cohort (string)
- Classes taken (dictionary)
  - keys: department names (string)
  - values: classes taken in that department (list of course codes (strings))
- Departmental averages
  - keys: department names (string)
  - values: departmental average (float)
- Overall average (float)
- Classes selected (list of course codes (strings))

### courses

- Course code (string)
- Department categorization (string)
- Prerequisite courses (list of course codes (strings))
- Prerequisite departmental average (float)
- Prerequisite overall average (float)

### admin accounts

- username (string)
- password, hashed (string)

### site status

- registration open? (int, 0 or 1)

## Frontend Simple Mockups

### Student Login

stuy.edu username:

OSIS:

### Student Interface

Overall Average:	??	AP Selection 1
AP Limit:	??	<input type="text"/>
Department Averages		AP Selection 2
English:	??	<input type="text"/>
Social Studies:	??	AP Selection 3
Mathematics:	??	<input type="text"/>
Sciences:	??	AP Selection 4
Foreign Language:	??	<input type="text"/>
Technology:	??	
Arts/Music:	??	
Health:	??	

## Administrator Login

username

password:

Login

## Administrator Interface

logout

Student / Course lookup

type ▼

student  
course

criteria type ▼

OSIS  
name  
username

course code  
name  
department

search

Add Student

choose file

add

Add Cohort

cohort name:

choose file

add

Remove Cohort

cohort ▼

2017  
2018  
2019  
2020

remove

Close registration

Edit Course / Student		logout
<STUDENT NAME>	edit	
<OSIS>	edit	
<STUY.EDU USERNAME>	edit	
AP Limit: ??	edit	
AP Selection 1 (other slots grayed out until previous slot is filled)		
<AP COURSE>	edit	
AP Selection 2		
<AP COURSE>	edit	
AP Selection 3		
<AP COURSE>	edit	
SAVE		

printed by Rick.