Identification

* Project Name: Class 5
* Members:
  + Ethan Seide (Leader): eseide@princeton.edu
  + Arin Champati: champati@princeton.edu
  + Chaz Bethel-Brescia: chazb@princeton.edu
* Repository: <https://github.com/EthanSei/Cos-333-Project-Class-5>
* The lead instructor and our TA have read-only access to our source code repository.

Elevator Speech

They say college is the best 4 years of your life, so how can you make the most of it? By taking the best courses! And how do we find the best courses? Class 5. Class 5 is the solution that helps Princeton students find the best courses for them with the click of a button. First, a user inputs a selection of some courses they like and some courses they dislike for Class 5 to build a user profile. Class 5 then utilizes preferences from other users to suggest the best courses. In this way, students receive personalized suggestions, based on historical data, about what courses they will most likely enjoy.

Overview

The Class 5 app intends to make Princeton students more aware of courses they might enjoy based on their interests. Based on the user’s own preferences, combined with preferences of other students with similar interests or academic paths, each user receives suggested courses that will guide their course selection decisions. We hope not only to ease the class-hunting process, but to expose students to quality courses that they might not have taken otherwise. The user will be guided smoothly between the Home page, Preferences page, and Suggestions page, where they will sign in with CAS authentication, input 5 liked and disliked courses, and instantly receive a curated list of courses.

### Requirements

The system should function as a tool to assist Princeton students in course selection - namely, suggesting courses based on their user preferences in relation to other users’ preferences. The system’s intended users are Princeton students, and more specifically, Princeton students who are having trouble finding courses or are curious about what courses are ‘objectively’ the best for them.

The system will provide a selection of course suggestions that are based on the user’s preferences and their relation to ‘historical’ Princeton student preferences. Simplified,, if a student likes class A, and many other students who liked class A also like class B, the student will be suggested class B. This system benefits the users as it provides a fast, simple experience to find potential courses to take. This, of course, is in contrast to existing methods - namely, manually reading course reviews, asking friends what to take, comparing numerical course ratings, etc. The problem with the existing solution is that it is both inefficient and uninformed. The user would have to manually track each course review somehow, and the user may not know exactly what courses to look for / compare. These problems are what Class 5 aims to solve.

### Functionality

1. User logs in through CAS
2. User fills out list of most liked and least liked courses as well as confirms other information (year, major, certificates, etc)
   1. Stretch goal: user will be able to go back to this page and change preferences for their courses, change reactions to course suggestions, and change their major, year, etc.
3. User receives a selection of courses they may also enjoy based on their input preferences (see design for suggested implementation)
   1. Stretch goal: user can query with basic filters (i.e. filter by distribution, course level, department, etc.)
   2. Stretch goal: user can query with advanced filters (i.e. filter suggestions by major, certificate, or class year). For example, if user searches for suggestions with a COS major filter, then user will see suggestions that COS majors are getting.
   3. Stretch goal: if user selects class in the suggested courses list, basic information about the class will show up (i.e. cross listings, professors, reviews, etc.)
      1. Stretch goal: user will see additional advanced information such as percentage of positive/neutral/negative reviews as analyzed by a sentiment model.
4. User can remove courses from the suggestion list (i.e. if they have already taken it or don’t like the result), and new suggested courses will fill their place.
5. General
   1. Web application will be optimized for a desktop window
      1. Stretch goal: web application will have different modes for desktop, tablet, and phone.

**Scenario 1: Janice – the Jaded Junior**

Janice is now a junior at Princeton, and her schedule is *busy* to say the least. Amidst the growing stress from internship recruiting season and Independent Work, she doesn’t want to spend a ton of time finding that extra class to add to her schedule for spring semester. This is when she remembers Class 5! During her daily coffee break, she logs into Class 5, quickly jots down some classes she liked and disliked, and *boom*. She has five classes tailored to her needs, chosen based on people who have already taken similar courses. She peeks at a couple of their descriptions, decides on something that seems fun, and her schedule is good to go. Just in time for her next precept.

**Scenario 2: Fred – the Frazzled First-Year**

Fred is thrilled to begin his first semester at Princeton, but finds himself overwhelmed by the number of courses at his disposal. He has already browsed through course reviews on princetoncourses.com yet is still having trouble narrowing down potential options from the wide range of opinions and feedback. Already apprehensive about his course-load for his prospective major, he wants a class that will round off his schedule and provide him with a new experience, while also suiting his interests. Then he remembers Class 5. Even without any prior knowledge about the Princeton courses, he’s met with suggestions based on the interests and discoveries of other like-minded students, and picks one that looks super cool. Now he can kick off the semester in style, without having to dig through endless pages of the course catalog. No stress!

### Design

**Frontend**: Bootstrap website (JS, HTML/CSS)

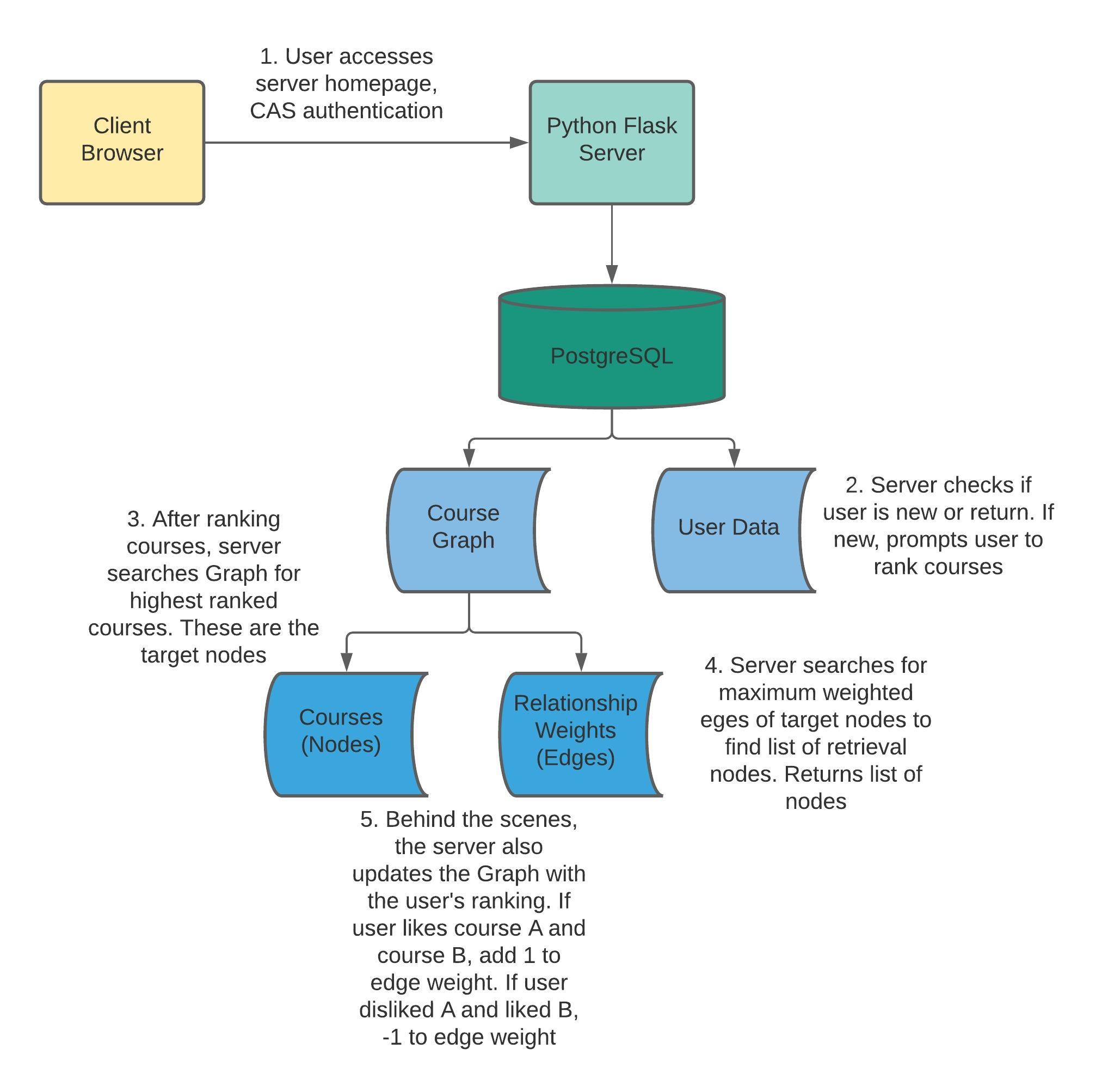
* CAS Authentication
* Homepage
* Prompt user to rank courses or update rankings if returning
* Display suggested courses
* Jinja2 Templates

**Backend**: Python Flask Server (Python)

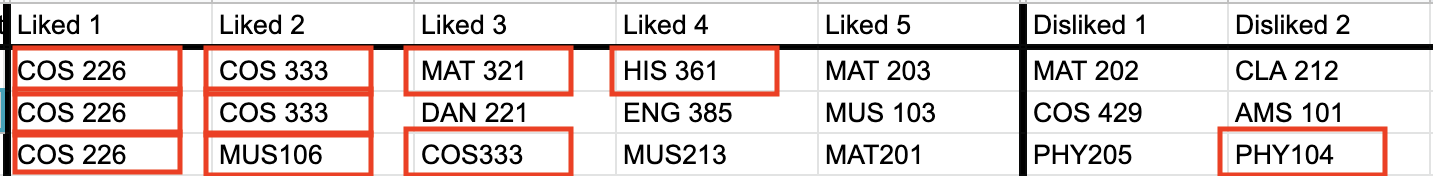
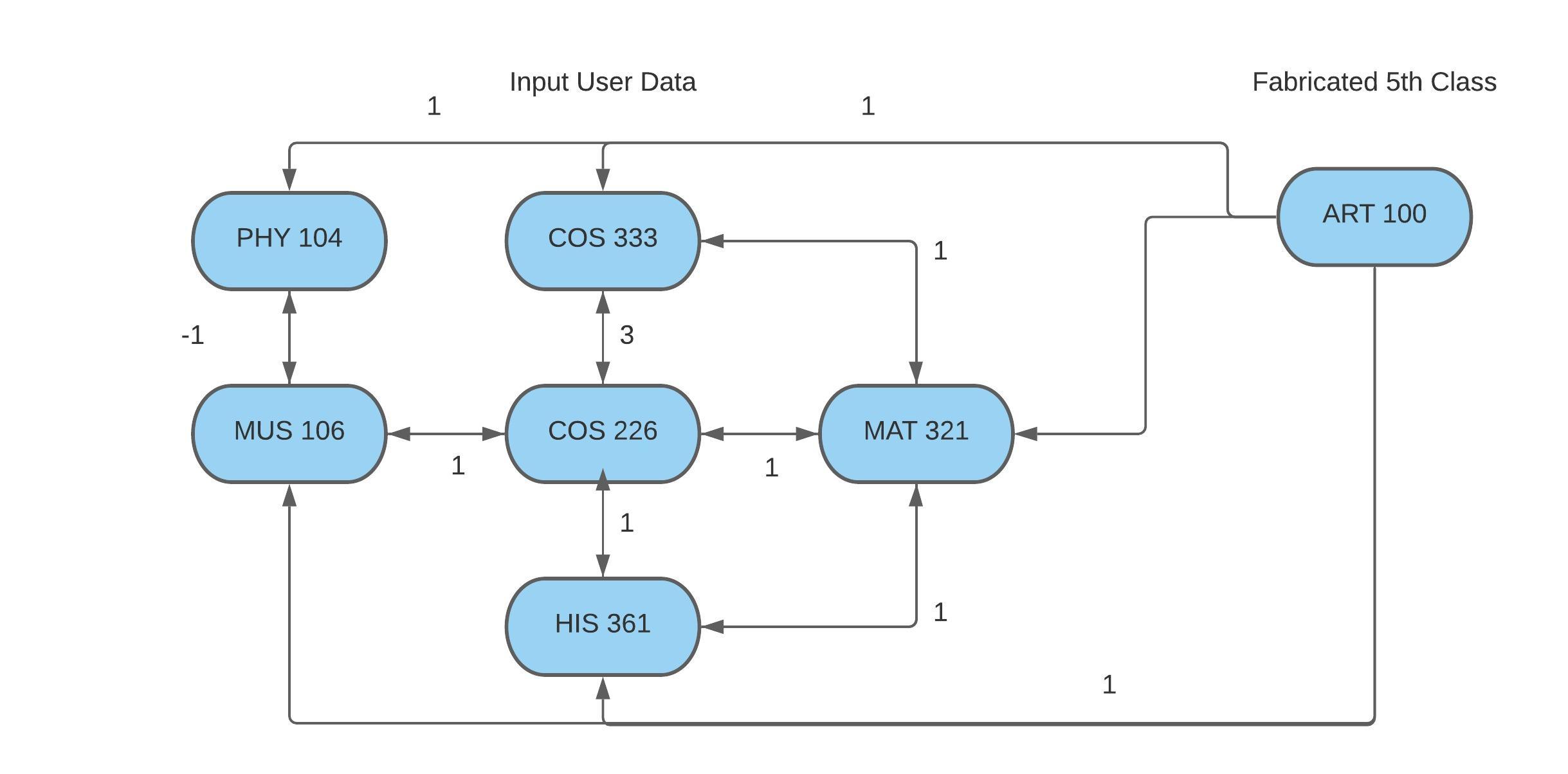
* Framework for web application
* Retrieve data from database
* Business logic to search for correct courses in DB (see diagram below)

**Database**: PostgreSQL (\*\*Best option to store an edge-weighted graph?)

* Store user input data
  + Past rankings
  + Past courses suggested
  + User id, major, year, etc
* Store Course Graph
  + Course Nodes
  + Edge weights for utility between courses



Graph Structure (Red-boxed Courses Shown)

****

Initially, nodes are courses with edges of weight 0 (or no edges). Add edges with weight 1 between “fabricated 5th courses” which are courses we determine\* are good 5th classes. This way, if a user likes classes that no one else has taken, it has initial suggestions.

Every user inputs 5 (or more) courses they liked and 5 they disliked. For each pair of courses A, B that a user likes, the weights of edges A-B and B-A increase by 1 (or initialized with weight 1). For each course A that a user likes and course B that a user dislikes, the weights of edge A-B and B-A decrease by 1.

\*Initial fabricated 5th courses are determined by examining Princeton Courses and asking friends. When the initial survey goes out, we can also ask individuals for their favorite 5th course.

### Milestones

**MVP** (self contained, but copied from [link](https://docs.google.com/document/d/1tyWL84Gw-WDyGLzO-LSKRTyN297qvEAzpbjw_wvpZbs/edit?usp=sharing))

For the purposes of describing the MVP, I will walk through an example of what a user would see. First, the user would be greeted by a homepage with the option to login or see an about page. The user can then log in with CAS authentication. If this is the first time a user is logging in, they will be sent to a preferences page where their major and class year will appear, and the user can confirm these are correct and modify to their liking. The user can then input a list of courses they have liked at Princeton and courses they have disliked at Princeton. Then the user can continue on to the suggestions page. In the MVP, the suggestions page will just contain a list of suggestions. This is the functionality of the MVP.

**Minimum Viable Product**

* Home page - sleek, nice
* About page
* CAS authentication
* Ask for info:
  + Select classes that you could possibly have taken (calculated with the year we get from your data)
    - Depending on class year, require a minimum of 3 likes, 3 dislikes, and max of 10.
  + Ask personalized questions (makes sure the pulled data is correct): major, class year, **Stretch**: certificates?, interests
* Calculate a list of ~10 5th courses to show you and display to the user
  + Can do a re search
  + Can select one to remove and research for another one (or n)
  + **Idea**: can also allow student to pick *any* course
* We should have a page of classes you say you like and don’t like
  + You should be able to update these preferences on this page
* **How we get initial data:**
  + Google sheet: send a google sheet to our friends where the list 5 of their favorite courses, and 5 of their disliked courses.
    - We want at least 30-40 datapoints
    - Could send to our clubs, or even on listserv if we are feeling extra
  + Text: could also text people the same thing

**Stretch Goals**

1. Being able to change your preferences for liked/disliked classes as well as reactions for the Class5 suggestions
2. Resizing for tablet and phone
3. Filters
   1. Class based filters
   2. By major, or by class year, or by certificate
4. Displaying information about the courses
   1. Numerical reviews (ratings)
      1. Our own metrics? I.e. sentiment analysis
   2. Written reviews
   3. Registrar link
5. Sharing the suggested classes via email or text
6. Semester suggestions

**Potential Design**

|  |  |  |
| --- | --- | --- |
| Homepage | Preference Page | Suggestions Page |
|  |  |  |

**Timeline** (see Timeline document)

### Risks

* Getting initial user data:

Our first major task will be collecting user data to fuel our class searching-algorithm. We’ve created a Google slides specifically for this purpose – cataloging an individual’s class year, major, certificate, favorite “5th course,” 5 Liked courses, and 5 Disliked courses – and we’re beginning to disperse it to our respective friend groups at Princeton, which will hopefully yield around 50 completed responses. However, we must also consider the possibilities that:

1. We don’t obtain enough data for the algorithm to organically generate a 5th course without being predictable or repetitive.
2. Our sample will likely not be representative of the entire undergraduate population.
3. Users with similar academic profiles may have similar preference for courses outside their major, which could lower the number of unique links between nodes in our graph, and yield a graph that is not very interconnected.

To account for an initial lack of data, we may use generic suggestions (likely guided reviews on princetoncourses.com) to get our system up and running. To account for the potential lack of interdispersedness between graph nodes, we could initialize edge weights between known popular 5th courses and all other courses, to add some variety to our selections. If our algorithm is still not functioning at a sufficient level, we would then pivot to utilizing the already-existing course data princetoncourses.com to construct a more populated graph.

* Time-sensitivity: Learning new tools, languages, systems

We each have various levels of experience in web programming, especially with using PostgreSQL and Flask. We are aiming to divvy up our responsibilities so that we each spend some time working in the front end and back end, but we’ll also find a balance such that each person has time to overcome the learning curve to get practice using a new technology, but also gets to spend time working on a component they’re more comfortable with.