Project Timeline

**October 5, 2020 (Theme: Prep)**

* Finalize MVP and Project Overview [Team]
* Deploy something on Heroku
* Start collecting data
* Nail down pseudocode algorithm
* **Arin:** Set up Heroku app hello world, looked into matrix factorization, finalized data entry form, and sent it out to a few people.
* **Chaz:** Began outreach for data collection
* **Ethan:** Created flow charts to visualize data. Setup timeline

**October 12, 2020 (Theme: Prep)**

* \*Partner with other COS 333 team to be each other’s users [Team][Done]
* Create OIT account (for CAS authentication)
  + ActiveDirectory (princeton student info)
  + MobileApp (course info)
* Send out survey to Princeton student friends, teammates, club members to obtain at least 50 user profiles on course rankings [Team][Done]
* Determine roles and responsibilities [Team]
* Create final Heroku Account [Ethan]**[Done]**
* \*Set up PostgreSQL DBMS → start inputting data **[Done]**
* **Arin:** Monitored data collection sheet – edited the entries to make sure that there were uniform conventions, manually added netIDs and names after we realized that we needed them, worked with Chaz to flesh out the adjacency list idea, figured out details for the recommendation system. Also wrote script to convert csv file to postgresql.
* **Chaz:** Continued outreach for data collection; began sketches for PostgreSQL database structure; worked with Arin to refine graph structure (using adjacency lists to streamline important information).
* **Ethan:** Reached out to friends/clubs for data collection; Created Heroku Project Account; Setup Flask and Jinja2 framework; Created initial graph tech spec; Midterms this week slowed me down → next week will be better

**October 19, 2020 (Theme: Plan)**

* Send out survey to Princeton student friends, teammates, club members to obtain at least 50 user profiles on course rankings [Team][Done]
* Meet with George R. Kopf and discuss getting course reviews like in PrincetonCourses.com [Done]

Set up initial Flask framework[Done]

* Create entire 3-tier system flowchart [Team]
* Set up PostgreSQL DBMS → finish inputting data [Done]
* Prototype Version → rudimentary functionality
* **Arin:** I worked on both the Database class as well as the data sources. I went through the princeton data sources and created deeply nested dictionaries of all the information we wanted - the dicts represent tables. I then went through our csv of data, and I converted that data into nested dicts that represent database tables. I also thought ahead and automated a process that lets us pull the data we need from a range of terms, as opposed to just one term. I converted the csv data into data in the form we want (i.e. dept, coursenum mapped to courseid). I also created the logic for pre-populating the database with multiple tables, only a few left to go.
* **Chaz:** I worked on fleshing out the structure of our Database class, writing methods allowing us to add a new user’s information to the table, remove a user’s information, update a user’s liked/disliked courses, or get a user’s liked/disliked courses.
* **Ethan:** Did error handling for the database, set up initial structure, created flask application that displays user information, created templates, connected everything to Heroku.

**October 26, 2020 (Theme: Create)**

* Create course graph database layout and get everyone’s approval [Done]
* Create the course graph class itself [Done]
* Display a list of suggestions for a class [Done]
* **NOTE:** We need to figure out what we’re doing with favorite fifth classes in regards to unitary weights
* **Arin:** Populated database with our existing data. Created tables in the database that store unitary weights and edges. Worked with chaz to finish the architecture for CourseGraph and helped create the first suggestion algorithm. Put it all together in the flask application that Ethan made.
* **Chaz:** I fleshed out the CourseGraph and CourseNode, and wrote a constructor that connects to the database and takes in unitary weight and edge information. Arin and I both worked out a preliminary suggestion algorithm using a PriorityQueue ranked by each suggested courses’s composite score (a combination of its unitary weight and edge weight to that course)
* **Ethan:** Helped get the CourseGraph and CourseNode classes started and laid out. Created a new index page for viewing course suggestions.

**November 2, 2020 (Theme: Create)**

* Create course graph database layout and get Princeton staff’s approval for data usage [Done]
* Display a list of suggestions for a given user [Done]
* Basic UI where a user can login, register or update their information, and receive suggestions [Done]
* Alpha Version → all basic functionality [Done]
* **Arin:** This week, I worked on the Flask logic, the database logic, the data source logic, etc. I worked with Chaz on allowing the database to have update methods instead of just add methods, so that changes can propagate through the system. I got the Flask logic to work more like what it will be for the final project: i.e. the logic with CAS. I fleshed out Ethan’s code to make it work for the alpha version.
* **Chaz:** Worked on logic for getUserSuggestions() method – which gets all suggestions for a specific user – so that it eliminates duplicate suggestions, and tracks maximum scores of each suggestion in a dict before populating the PriorityQueue. Wrote methods for updating certificates, crosslistings, and coursedetails.
* **Ethan:** Worked on setting up the frontend UI for the login, signup, and main page for suggestions and connecting the frontend with the CourseGraph. Helped remove courses the user already liked from the list of suggestions.

**November 9, 2020 (Theme: Debug + Beautify UI)**

* System Security
* Additional methods:
  + Most liked / disliked / disputed classes: sorted by weight and # people
  + For a course, get % of people who liked or disliked it
* \*\*\***Finish ALL backend functionality + features**
* Begin to beautify UI for login, profile, preferences pages
* **Arin:** Implemented CAS Authentication into our application, redid our session logic for the database - now we are calling sessionmaker() once and passing in sessions when we need them. Also took care of session rollbacks when an exception occurs. Started front end: I had designed a basic version of what the site would look like. I coded all of the homepage and styled all of the profile page. I also did all of the form validation in the profile page.
* **Chaz:** Added more functionality to getSuggestions() method in CourseGraph, so that if the algorithm doesn’t find enough related suggestions for a user, it pulls from the most popular courses. Also began to make dashboard interactive with JQuery so that a user can remove a suggestion and the list repopulates with updated suggestions.
* **Ethan:** Added autocomplete features for profile page using AJAX and accessor methods in database for retrieving lists of courses and major similar to an input. Began working on the User’s/Programmer’s Guides

**November 16, 2020 (Theme: Test + Evaluate)**

* Roughly finalize UI [Done]
* Iron out bugs in major features [Done]
* Test MVP (other COS 333 group, friends) + get feedback [Done]
* Beta Version → All functionality with passable UI [Done]
* **Arin:** Helped Chaz debug the dashboard suggestion logic, and also helped create a new approach. Styled the dashboard - added buttons, search bar, made the suggestions clickable, displayed information (required database methods). Finished the profile page by adding certificate section and also disliked suggestions section.
* **Chaz:** Reworked logic for removing suggestions so the page is much quicker and more responsive – rather than repopulate the entire list, fetch one more disliked suggestion and append in a new <div> at the end of the list. Also manipulated the event handling so when a user clicks on a disliked suggestion, they cannot dislike another suggestion until the first response is handled (avoids overriding any requests)
* **Ethan:** Added autocomplete features to certificates (AJAX, flask endpoint, and new database method). Added Most Liked courses filter. Allows users to select the most popular courses in the system and includes the functionality to add these courses to their disliked suggestions list. Created a google form to get feedback from test users

**November 23, 2020 (Theme: Evaluate)**

* Extra week
* Clean up code + flesh out documentation [Done]
* Continue to incorporate feedback [Done]
* High level and lower level diagrams (do not need code to be perfect) [Done]
* Heuristic Evaluation on System
* Stretch goals?
* **Arin:** Assisted Chaz with the user guide. Stylized the button Chaz added and the filters that Ethan added.
* **Chaz:** Worked on the User Guide, and added a button to clear all disliked suggestions, so a user does not have to repeatedly click the delete button to remove numerous disliked suggestions.
* **Ethan:** Added new endpoints for filters and re-organized how /dashboard pulls suggestions. Finished up basic filters and fixed some bugs. Helped clean up the code and built the first draft of the About page.

**November 30, 2020 (Theme: Evaluate)**

* Extra week
* Clean up and modularize code [Done]
* Continue to incorporate feedback [Done]
* Completed beta version of application [Done]
* Presentation on Thursday, Dec 3 [Done]
* **Arin:** There was a major bug that all of us had to work together to solve. The dynamic script was messing up in the dashboard page. I also worked with Chaz to create the presentation slides, and of course, I prepared for the presentation. I also did some individual testing on the system to make sure it was working properly.
* **Chaz:** Worked with Arin to prepare slides for presentation, found bug in numbering of course labels for recommended suggestions (helped Arin debug), wrote questions and answers for FAQ page.
* **Ethan:** Mainly debugging and asking users to run through test cases. Created course graph and overall architecture boxes and arrow diagrams to use in the presentation

**December 8, 2020 (Theme: Finalize)**

* User’s Guide [Done]
* Programmer’s Guide [Done]
* ProductEval [Done]
* Go over Production Quality, Modularized Code
* **Arin:** Worked on Programmer’s Guide, styled the FAQ page, worked on the Product and Project eval.
* **Chaz:** Worked on Heuristic evaluation, Project Eval, and Programmer’s Guide.
* **Ethan:** Worked on the Programmer’s Guide, ProjectEval, and helped clean up all other documentation.

-----------------------------------------------------DONE-------------------------------------------------------

Thanks for a wonderful semester Nick!