Team Name: NPRTS
Report Number: 11
Reporting Week: Week11, Week of Mar 20

Summary of work completed in prior week:

- Base of the core scheduling algorithm is complete. It generates all possible schedules as a tree given a list of courses and pulling the corresponding prerequisite requirements for those courses from the database. Currently, it only works for the computer science major, including high-level CS electives and knowledge area courses.

Summary of work planned for next week:

- For the scheduling algorithm, there are multiple outstanding issues for the next couple of weeks:
    - Decide on how we want to implement knowledge area courses and other miscellaneous electives. Since you can basically take most of these courses at any time, and there are five knowledge areas required, the algorithm currently generates 5! additional schedules, which is computationally way too inefficient.
    - Implement the other major and both minors. This should be the same algorithm, but we need to hammer out the specific list of courses for each of these, as well as figuring out how majors and minors combine with each other.
    - Implement transfer credits. This should be straightforward, as it essentially just means giving the same list of required courses, but instead of starting from the root of the tree as an empty list of courses completed, start with the ones that a student is transferring in.
    - Format the list of possible schedules as something that can be passed to the front end. Since each branch of the tree is one possible schedule, we might just do a list of all branches.
    - Make the schedules such that UNIV 190 and FY 100 are always taken in the first semester of freshman year.
    - Calculate potential schedules in the algorithm based on the assumption of a credit limit rather than a number of courses limit.
    - Handle the weird cases of course offerings, like even fall and odd spring.
- Get data from the backend to the frontend of the application, in order to fill out pages such as the course catalog, course history, and advisors/advisees.
- Divide up the frontend by type of user — present different pages to the user if they are a student vs. advisor vs. administrator.
- Implement frontend signup and login.
- Plan for the design presentation on April 4.

Open issues and action plan to resolve them:

- The team website eventually needs to be updated with all of our reports and documentation.
- The full requirements document needs to be written. It should be adapted from the presentation and some of our other existing documents, with more details given and in the IEEE format (already in the Drive).

Project Management Summary

Team member	**Tasks completed**	**Hours worked for week**	**Total hours**
Tyler Yankee	Worked on the scheduling algorithm. See details above.	16	44
Peter Dorovitsine	Attended meeting and helped plan out future work	1	22
Robert Licata	Worked with Niall to clean up the front end.	2	25
Niall Pepper	Attended weekly meeting and continued working on cleaning up details on the front end.	2	34
Sangwon Youn	Attended weekly meeting and continued working on admin page	4	25

Meeting Summary

For each team meeting provide the date, time, and location of the meeting. Indicate which members attended the meeting and action items discussed in the meeting.

March 22, 2023, 4 p.m. ERC. All members present.

- Discussed current status coming out of the break in terms of both front end and back end, and what each of us are working on in the next week.