SmartScheduler — Sprint 2 Requirements Artifact

1 Summary

We're building SmartScheduler, a tool that lets students pick desired courses and instantly see all conflict-free schedule combinations in a weekly view. Students can pin preferred sections and block times they can't attend to refine results.

2 Problem

Students currently create their class schedules manually by searching the course catalog, selecting desired classes, and tracking options in spreadsheets. They must identify and compare potential section combinations on their own, which is time-consuming, error-prone, and often leads to invalid or suboptimal schedules. SmartScheduler automates this process, generating all valid schedule combinations and significantly reducing the time and effort required to plan a semester.

3 Goals

- Reduce schedule planning time and stress for students

- Eliminate the possibility of a student creating an invalid schedule

- Improve quality of schedules by making all combinations visible

- Respect the constraints of the user

- Make results exportable

4 Functional Requirements

- F-1: The system will allow users to search for classes based on its key and class number. Set up a basic search bar with fake data for now that gets the results and puts them in a dropdown menu.

    - Ex: EECS 581, HIST 101

- F-2: This system will generate all possible combinations of the users selected classes

- F-7: The system will display the title of class, credit hours, a short description, and the instructor in the card that will display after the user selects the class from the search

5 Non-Functional Requirements

- NFR-1: Works in modern browsers (Chrome, Edge, Firefox, etc)

- NFR-2: Colors follow WCAG 2.11 contrast guidelines

- NFR-3: Response time for the HTTP call the university server should be under 2-3 seconds

-NFR-4: The system's front-end architecture should use standard React conventions (component-based, reusable, documented) to ensure maintainability

6 Constraints & Assumptions

- Frontend stack: React + TypeScript, built with Vite
- Universities server with class information is accessible
- Server returns needed information to display to the students
- No server of our own, everything is handled client side
- No PII handled for this project

7 Out of Scope

There are three big features that we wanted to add to this system but they will have to be added later as we will not be able to get to them in this project. First we wanted to add a way for the user to create an account and use this account to login on later uses of the system. We would use this account to allow users to save their favorite schedule to their account so they could view it again later. Having accounts would allow us to do the second feature which would be allowing the student users to send their proposed schedule to their advisor. This feature could possibly eliminate or at least make meetings with your academic advisor more efficient. The last feature is in that same area, it would allow the user to enter their DPR(Degree Progress Report) and then we would suggest classes that they should take that they still need to take and are being offered in the next semester.

8 Glossary

- SmartScheduler - This application; generates conflict-free schedules from selected courses.
- EECS & HIST - These are shorthand for class departments specifically 'Electrical Engineering and Computer science' and History these will be used by the user to have a better search
- Degree Progress Report - Is a file that the student could get from the university that shows what classes they have taken as well as classes they still need to take to earn their degree
- Credit Hours - Unit of academic credit associated with a course/section.
- React - UI library for building component-based web interfaces
- Vite - Front-end build tool/dev server with fast hot-reload for React/TS projects
- PII (Personally Identifiable Information) - Data that can identify a person (e.g., name, student ID, email)