

Design Change Document #1  
2/20/24

- We are going to change how the calendar view lists multiple items blocked during the same time period. We are now going to display them side by side in the same way they would appear on Google Calendar
  - The reason for this is so that all information will be readily available without having to click for additional information (e.g. Needing to click to see the course codes for overlapped courses)
- The calendar will no longer display certain dates, instead displaying a generic Monday-Friday schedule.
  - We chose this because the dates don't matter when courses are the same each week throughout the semester.
- When creating a schedule for a new semester, the user will be able to import a previous semester's schedule or start fresh if they want
  - Import last year's equivalent schedule by default (with option to switch). For example, creating Spring 2025 will by default import Spring 2024's schedule
  - This is to make creating new semester schedules much quicker
- The design will be focused on the admin creating or editing the schedule. A "view mode" for other faculty is much lower priority
  - This change is because it doesn't really matter if other non-scheduling faculty is able to view the app, therefore, a view mode should only be considered as a stretch goal.
- We have an increased focus on creating help functionalities for the users since we expect them to be less familiar with our app, as it is not used very frequently. This will primarily involve having "Help" popups or basic function descriptions of the different pages and uses of the app.
  - This is because our target user base will not necessarily be "good" at technology or familiar with our app and may need help after we have left the project
- We are going to change how we think about the constraints list
  - Using the database, have automatic constraints for the following:
    - Two courses should not be using a room at the same time
    - A course's size should not exceed its room's capacity
    - An instructor should not be teaching two courses at the same time
  - Manual constraints we add and build into the system. These are primarily based on what we know about what courses students commonly take in the same semester.
    - Example: CS2311 and CS2321 should not be scheduled at the same time
  - The reason for this change is because we discussed the important constraints with Dr. Ott and settled on this list.
- We will keep track of "instances" of constraints (each time when a constraint is broken).
  - This is to allow these instances to be quickly retrieved and allow them to be "ignored"

- We now have a clear priority that we are going to follow for main features of the app based on feedback from Dr. Pastel and Dr. Ott, which is displayed here. We're adding this because it's a great idea to have clear goals and an order to follow them.

*Priority List*

1. *Be able to view all classes and instructor in calendar and list*
2. *Implement primary constraints. By primary I mean delineated. I know you thought maybe add new conflicts. The only new conflicts I think should be added would be something like cohort class.*
3. *View the conflicts. We discussed multiple ways. Certainly do the easy one first*
4. *Enter classes and instructors.*
5. *Reach goals?*
  - a. *Help Page/how a user can use the product*
    - i. ***Likely trivial and should be done no matter what if we keep good documentation***
  - b. *Add spring security*
    - i. ***Different levels of security lower priority***
    - ii. ***Base level of security and login functionality is important - we don't want anyone with the URL to be able to edit the app***
  - c. *Single Sign on*
    - i. ***This only really matters along with view mode access***