Criterion A: Planning

Defining the Problem

My client was having trouble budgeting time for schoolwork, and was forgetting deadlines for relatively important work. At his school, the penalization for this performance can be flexible, but is typically dire. My client tried recording this information in a planner, but keeping up with the planner's physical location and remembering to check it also became an issue. Upon discussing the issue further with my client, I figured that a simplistic but functional scheduling application that manages coursework and assignments, ranging from schoolwork to extra curricular activities, would be helpful, and possibly even fix the problem entirely.

I met again with my client to see if there were specific things that could be included that would specifically benefit them, and as expected, the ability to filter assignments by their respective due date was a critical necessity. Because of the nature of the problem, the application relies mainly on sorting algorithms - sorting by date, mainly, but an additional sort by alphabetical order might help to just display remaining work in a more general manner. By splitting the program into two sections, it can remain functional and minimalistic at the same time. The two parts would involve adding new assignments with their relevant name, course, date, and possibly a description of the work, whereas the other part would focus on sorting and displaying.

Rationale for the Proposed Solution

For the purposes of this project, I will be using Object Oriented Programming in Java, as it has been the basis of my computer science studies in my courses. I believe that the intrinsic "easily modifiable" value of OOP will benefit the lifespan and relevance of the project, too, as making changes based on client demands is a reasonable and possible expectation. The multi-platform capabilities of Java can also lead to cross-platform availability, which means that my client could form their schedule from multiple devices with ease.

In order to make the program usable, the interface will include a form-like window to add the projects with the option to mark fields as "N/A" (in the case of them being irrelevant), a window to count the total number of assignments recorded, as well as respective "tabs" that contain lists of the results of different sorts. By clicking on the elements in a respective list, I

would like to create a pop-up that shows all provided information for the specific assignment.

Starting Success Criteria

- There are two buttons on the main window that allow the user to navigate to the two other parts of the program.
- There is a window with several text fields that will receive user input to add assignments.
- The window includes a "submit" button that will record this information, and the button will have to send the user back to the main window.
- There will be a window that includes an assignment counter, with a "back" button to navigate to the original window.
- The program should display a list of projects, sorted by their respective due date.
- The program should display a list of projects, sorted alphabetically.
- The program should display a list of projects, sorted by when they should be started based on date due as well as how many prospective days the project would take to finish.
- By clicking on an assignment within a displayed list, the program should create a new window to display all of the information recorded for the specific assignment.