Outline Description

Services:

- Students will be able to add/drop courses, search for courses, and print their schedule
- Instructors will be able to print their schedule and class list, as well as search for courses
- Admins will be able to add/remove courses from the system, add/remove users, add/remove students from a course, and search/print rosters and courses

Constraints:

- Database must work for 100 students, 10 instructors, and 1 admin
- Users should not be able to access functions outside of their classification

Goals:

- System should provide a scheduling service which will allow students, instructors, and admins to add courses, search for courses, print schedules, etc.
- System should include multiple semesters, print-out of schedule, and scheduling preferences
- Needs both a database of users and a database of courses

Estimated duration:
5/11/23 - 5/19/23

Concurrent Activities

Specification

Start off with the most bare-bones version of the code which still incorporates all of the requirements from the outline description. The code created for Assignment 1 essentially is this already, where all of the functions are there and able to be called, but none of the functionality has been coded in yet.

For a true initial version, a preliminary database should be added and interfaced using some of the functions.

Estimated duration:
5/19/23 - 6/2/23

Development

After the initial version, true development on the project should begin. This means each new feature that is added should be intended as the final version of that function, and then testing can determine if there are any issues with its implementation. This should be done over and over again with each necessary requirement/goal, creating multiple intermediate versions of the project that slowly add new functionality over time.

Additionally, if there comes a time where a new feature needs to be added that was not specified in the outline description, it should be added to the specifications as a way to document its existence and then development can continue.

Estimated duration:
6/2/23 - 7/28/23

Validation

During the development of the intermediate versions, each new feature should be both developer and user tested. The testing should continue until all issues with this new version have been addressed and the new feature is functioning as intended.

After enough versions, final testing should begin in a few different forms as a way to validate the system. Although both user testing and developer testing have been in effect through the whole project due to the nature of this process model, it should be the main focus before shipping the final version since no new functionality is being added. Once these tests have been completed and the system is sufficiently validated, the final version can be shipped.

Estimated duration:
6/9/23 - 7/28/23

Initial Version

The initial version should be a skeleton of all the requirements specified in the outline description. In a way, this has already been mostly completed in Assignment 1, but it is missing a database, which is needed in order for most of the functions to be tested in any sort of meaningful way. Since later down the line there will be many intermediate versions, the database in this initial version can be very rudimentary and will only server the purpose of testing if some of the functions work and can access a database without issue.

Estimated duration:
5/19/23 - 6/2/23

Intermediate Versions

Intermediate versions of the code should slowly add functionality and fixes to bugs and issues found in previous versions. Initial versions of the system will be fairly limited and likely interfaced solely through the command window. Each new version should include some new completed goal; this could be adding a new database, adding functionality to a certain user, creating the scheduling system, etc.

Later down the line, versions can possibly start to include stretch goals, such as a clean GUI and additional options not listed in the basic requirements.

Estimated duration:
6/2/23 - 7/28/23

Final Version

The final version of the code should have all requirements filled and ideally every goal/stretch goal reached. Additionally, the system should be fairly well tested at this point as it has, in essence, been tested through each new iteration of the system.

Estimated duration:
Week of project due date

*****Estimated durations are tentative since most steps are repeated for every iteration of the system and therefore are done for essentially the whole length of the project**