Detailed Use Cases

Schedule Generator

- **Summary:** The user requests to build a schedule for their classes, and the program will generate all the class schedule options that fit with their time and professor that they want to take.
    1. User types a course ID in the search text box.
    2. User selects a course from the search results.
    3. If you would like to select a number of courses from a list of electives, check the option that says "Optional?" and add your electives.
    4. After selecting your courses, user can choose to omit certain sections or activities based on their location, professor, or if they're full.
    5. User selects a sort order: Earliest start, shortest days, least days per week.
    6. The program starts to generate.
    7. Program asks to choose option of different available schedules.
    8. User chooses the best option.
    9. System will automatically save the schedule.
- **Alternative Paths:** There are no special requirements associated with this use case.
- **Exceptions Paths:** In step 1, if user types invalid course ID, then courses will not show up.
- **Precondition:** Student should be able to generate a proper class schedule in accordance to their needs.
- **Postcondition:** The class schedule is generated and the course is ready for submission.

GPA Calculator

- **Summary:** The user requests a simulation of their GPA by inputting certain information, and the program will generate the GPA.
- **Basic Course of Events:**
    1. The program asks the user for his or her current GPA based on the 4.0 scale.
    2. The program asks the user to input his or her current classes.
    3. The program asks the user to input the weights (amount of credit hours) for each corresponding class.
    4. The program asks the user to input the letter grades received or estimated to receive for each corresponding class.
    5. The GPA is calculated based on the 4.0 scale and displayed.
- **Alternative Paths:** N/A
- **Exceptions Paths:** In step 2, if the GPA entered is greater than 5.0, an error message about overbounding the GPA will be displayed and step 2 will be repeated. In step 4, if the credit hour for a class is above 5, then an error message about overbounding the class weight will display and step 4 will be repeated. In step 5, if a letter other than “A”, “B”, “C”, “D”, or “F” is used, an error message will display, or if a symbol other than “+” or “-” is used, then an error message will display.
- **Precondition:** User’s GPA should be calculated according to students input.
- **Postcondition:** User’s GPA is calculated, and the menu is prompted again.

Final Grade Calculator

- **Summary:** The user requests a simulation of their final grade for their course by inputting certain information, and the program will generate the grade.
- **Basic Course of Events:**
    1. The program asks the user for the name of the course..
    2. The program asks the user to input the types of grades (Quizzes, Tests, Assignment, etc.) and the corresponding grade distribution.
    3. The program asks the user to input the weights (amount of credit hours) for each corresponding class.
    4. The program asks the user to input the average of the grades in each grade category.
    5. The final grade will be displayed.
- **Alternative Paths:** N/A
- **Exceptions Paths:** In step 2, if user puts a grade distribution that goes over 100 percent, the program could miscalculate and mislead the user. In step 3, if user puts more than four credit hours for a course, then an error message would appear.
- **Precondition:** The user is able to calculate their final grade using the Final Grade Calculator.
- **Postcondition:** The user receives their final grade and the menu will prompt again.

Class Route Generator

- **Summary:** Student requests direction to class by choose the class that they're taking to generate the route to their class.
- **Basic Course of Events:**
    1. Student determines which class and building they need to get to.
    2. Student inputs their location of where they are at the moment.
    3. Student inputs location of their selected class into generator.
    4. Route generates and shows student how to get there from the location the student is at currently.
    5. Student reaches location of their class by following the route generated from Google Maps API.
- **Alternative Paths:** Student can use a physical map to get to their class.
- **Exceptions Paths:** In step 2, if student doesn't have accurate location of where they are, then generator will not be able to effectively show student the best way to reach their class.
- **Precondition:** User is able to generate the route to their classes through Google Maps API
- **Postcondition:** User gets the class route that they need through the route generator.

Select Operation

- **Summary:** User should be able to select operations on what they want to do on the main page.

- **Basic Course of Events:**
    1. User goes onto main page of the application.
    2. User chooses the selection from a drop down menu of what they need to accomplish at the time.
    3. User goes through the steps of what was mentioned in the previous use cases based on whichever task they had selected.
    4, User accomplishes task successfully.
    5. User is able to go back to the main page for more selections of tasks if needed.
- **Alternative Paths:** N/A
- **Exceptions Paths:** If user does not go through the steps accordingly to what was mentioned in the previous use cases then they will not be able to succeed in accomplishing the task successfully.
- **Precondition:** User is able to choose between multiple options of their desired task they need to complete.
- **Postcondition:** User gets the required solution to what they needed and had selected on the main page.

Use Cases from Assignment 2

- The main section, the class scheduler, allows students to select in the GUI, which option they would want to view. From this menu, the students will have options, such as a GPA calculator at first, and then when the students provide the system with more information, the more options will unlock and allow the students to do more things with the program.

- The GPA calculator system allows the student to insert his or her classes, and the weights of the classes, and the letter grade (A, B, C, D, F) and allows the students to see a simulation of the GPA based on the calculations. As an example, the student selects two classes and then puts in the weight of the classes, which are 3 credit hours each. Student then puts in grades of B and F in the calculator. The GPA calculator will then calculate a GPA of 1.5 as an average of the B and F grades.

- The final grades calculator system allows the student to input their class' grades into the program and then, calculate the student grade at the end of the semester. As an example, the student inserts grades with the same weights of 60, 70, and 80. The final grade calculator would then process the information and show an average of 70 for the final grade. The student can then add or insert more grade after the calculation to affect this final grade

- The maps system will incorporate the Google Maps API and help students find the route to the correct building of the class they are taking. This is a great part of the application, especially for new students who are unsure of where they are supposed to go. As an example, the student wants to walk to his or her next class, so the student selects the class they want to go to and let the program do the rest of the work. The program will then triangulate the location of the student and give them a route to the building of the next class.

- The list of subjects system and class selection system will import the database of subjects that are available to take during the semester and allow the student to view this information in a clear format. The systems will also allow students to select the class that he or she wants to pick. The data may be collected and imported, possibly as a CSV file or other simple database file. As an example, the student will select SOFTWARE ENGINEERING as a class. The student will then be able to view this information. The student then decides to select this class as an option. The student decides to select more classes. The student selects OPERATING SYSTEMS, ROBOTICS, and COMPUTER ARCHITECTURE. This combined list of classes will be shown in an easy-to-view format for the student to view. The student looks at the credit hours and see that he/she is taking 12 credit-hours and is satisfied. The student then saves the configuration and moves on with his/her day.

- The class timings system or scheduling conflict system allows the students to settle class conflicts on the fly. The student will be able to view the class conflicts in the calendar GUI and make adjustments to the schedule without the hassle of remembering the CRN of the class. As an example, the student selects a class that has the schedule Monday,

Tuesday, Wednesday, Thursday 9:00am-9:50am. The student then selects a class that has the schedule Monday, Tuesday, Wednesday, Thursday 9:30am-10:20am. The student will be notified that there is a class conflict and will give more details, including the name of the course, that the class is having a conflict with.

- The notification system will notify a student when there is, by default, ten minutes remaining until the class begins. This setting can be changed, of course, within the notification system. As an example, we have a student who is always running late for class. The student sees the setting in the notification system that the notification is set for ten minutes before class. The student then sets this setting to 30 minutes before class and has rarely been late to class since then.