Test Results

During the implementation and testing part of the project we have faced a lot of bugs. Most of them were related to the fact that we expected all of the inputs to be perfect, hence did not cover the cases when the given input was wrong. For example, if for some reason the class object that saved the information about courses in a dictionary format had a wrong key, the program would not be able to continue execution. The reason is that all parts of the code expected the format to be perfect. Some of the bugs were related to the fact that the input was empty. One example is if the user does not select any courses our program would crash, or when the json files that are necessary for the algorithm (selected.json, courses.json) are empty the program would crash with error.

The bugs occurred on the following test cases: Test Case 6, Test Case 18, Test Case 20, Test Case 25, Test Case 26, Test Case 27, Test Case 30, Test Case 33, Test Case 35, Test Case 36, Test Case 37, Test Case 38. Some of the test cases raised the edge values of the input. For example, test case 18 occurred when we chose 9 courses which is the maximum number of courses that we expect students to take. As stated in the test case document, we resolved it by changing the loop iteration to include the edge case.

Test cases 25, 26, and 33 revealed the error of having an empty file. In order to fix that, we added the “try except” blocks to the code. Hence, if the file is empty the program catches the error, displays the error message, and terminates.

Test cases 27 and 30 revealed the error of having the file with the unexpected format. If the saved dictionary would have a different key from the one saved in the “course_class.py”, the program would crash. We fixed it by adding a “try except” block inside all the functions that try to access the keys of the dictionary. In case the key does not exist, the program continues the algorithm if possible.

Test case 20 revealed the error of having an empty input when the user tries to generate the schedule. We solved this problem by not allowing the schedule to be generated if no courses were selected.

Test cases 36 and 37 reveal the error of not being able to change from edge calendars (first and last) to another edge (last and first). This problem is not crucial. It is simply made to improve user experience. We fixed it by modifying the part of code that increments/decrements current schedule’s id, so that when the left button is pressed if the current calendar is first it transfers to the last. Same done with the last calendar.

Lastly, test case 6 failed when we tried to launch localhost from a different address. This does not really affect our project because in the future we are planning to host the website and launch it, so there will be no need to add specific localhost addresses to the database. We resolved this problem by adding that address to the Google Cloud Consoel, because we are using Google API.