Criterion E - Evaluation

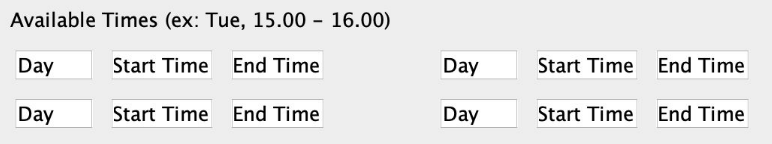
Evaluation of Product Based on Success Criteria

Success Criteria from Criterion A (Excluded from Word Count)	**Y/N**	**Comments**
The user will be able to		
Import CSV files	Y	
Record available time, subject of tutees and tutors	Y	
See matches for one tutee ranked properly	Y/N	(1)
Allow user to input number of wanted pairs and produce the best pair match ups for that quantity	N	(2)
Receive a list that matches up the subjects and schedule of tutees and tutors	Y/N	(3)
Receive a list that matches well with the user's manual list	N	(2)
Delete previous inputs	Y	
Change data input	Y	
Display information about the pairs	N	(2)
User-friendly features		
Intuitive GUI	Y	
Automatic CSV file import	Y	
User note box for each tutee/tutor to note special attributes not indicated in the program	Y	
Presentation of data using tables instead of text field	Y	
User doesn't have to change parameters to get better matches	N	(2)
Already paired tutees and tutors no longer appear in the ranking	Y	
Labelling of the attributes of tutors and tutees in my program matches the labelling Mr.Callahan uses	Y	

Error/exception handling		
If the wrong data is input, the program notifies the user	N	(4)
If the subject choice is not present in the options, there is an ‘other’ button where the user can add a new subject	N	(4)

Comments on Success Criterias Not Met

1. Instead of ranking the list tutors generated for a single tutee, I opted to generate a filtered list of available tutors with the same subject due to the difficulty of the ranking program. So, my program allowed Mr.Callahan to “see as many matches as possible for one tutee or tutor” but not ranked.
2. Initially, my program was supposed to have a mass pairing tab that pairs several tutors and tutees at the same time and a ranking tab that generates the best tutor match for the tutee searched in order. Again, I opted for a much more simple filtering tab than the aforementioned two tabs. Thus, all success criterias related to those tabs were not met
3. As I faced errors with the timeTable array I only made the program to allow Mr.Callahan to “receive a list that perfectly matches up the subjects of tutees and tutors”.
4. After I finished the program, I realized that the labels and embedded text in the text fields made it clear to Mr.Callahan what had to be input for the data, thus I thought I might not need to notify the user as they could quickly see the problem on their own. In addition, I realized that an indefinite number of ‘other’ buttons would be needed so I instead included all possible subjects which could be needed.



Word Count: 231

Overall User Feedback

Despite the downgrading of the final product compared to the initially conceived design, Mr.Callahan was satisfied with the product that was produced. To directly quote him, he said

> " I like the way you got the pairing set up over there. Checked if they are available. That stands out very well and helps to keep track of who is gainfully employed or not. That's pretty good and the ability to pair them up is a strength of the program. " [1]

He seemed to approve the pairing tab despite it being quite simple and he also seemed to appreciate the special notes tab I made for him to add notes potentially useful when pairing the tutors and tutees in the future. [2] He also didn't have any specific suggestions to improve the program, showing that the program fulfilled its purpose.

Lastly, he added "Yeah it is a nice presentation. It looks like it is easy to work with" verbatim, so it seemed that I was even successful in making the GUI intuitive.[3]

Word Count: 104 (excluding quotations from interview)

Recommendations for Further Improvement

After I finished showing the different tabs of the program, I told Mr.Callahan about several ways the program could be improved and he agreed to them by showing signs of approval. The different suggestions I had were geared towards completing the program as it was intended to be. To list them in order of difficulty, I suggested I could...

1. Make the program visually cleaner removing as many flag values when presenting data
2. Make the import file button read the different days the tutees and tutors are available,
3. Create a ranking algorithm that would not only filter tutors by subject but also by available time and judge rank each option based on how good they match with a tutee
4. Create a mass pairing algorithm that matches several tutees at the same time by optimizing the tutor option for each tutee.

The removable of 'not set yet's, for example, could reduce the time wasted scrolling through subjects, and the last 3 recommendations would all be essential in increasing the overall capacity and usefulness of the program.

Word Count: 175

[1] Kevin Callahan, interviewed by author, Bangkok, December 8, 2019, audio recording 11:00- 11:35, Appendix 7

[2] Kevin Callahan, interviewed by author, Bangkok, December 8, 2019, audio recording 3:40 - 3:50, Appendix 7

[3] Kevin Callahan, interviewed by author, Bangkok, December 8, 2019, audio recording 15:51 - 16:03 Appendix 7

Criterion E Word Count: 510

IA Word Count: 166+302+ 1017+ 510 = 1995