Criterion E: Evaluation

The evaluation of the success criteria was carried out with the client in the testing process of the program. The client's assessment is presented in the interview transcript [appendix 4].

1) The client can input new words, its definition, translation, pronunciation, and "links" to the vocabulary list.
   Met: the program's text boxes allow the user to input text .

2) The client's inputs (vocabulary words) can be saved in the program.
   Met: the vocabulary words and its attributes are saved in a JSON file after the "save" button is clicked.

3) The client is able to remove old vocabulary words. When removed, the word, definition, translation, pronunciation, and "links" will be deleted from the file.
   Met: the words in the JSON file can be removed by clicking the "delete word" button. The changes will be updated.

4) The vocabulary words are saved in an array, and sorted and displayed in alphabetical order [3] for the client to find a word in an easier and faster way to improve time efficiency.
   Met: the vocabulary words are in an array. Their first letters are compared to sort and display them in alphabetical order for efficiency.

5) A message appears in the controller if a new input/word already exists in the array.
   Met: when new input is already in the JSON file, the controller outputs "Already exists".

6) When the client clicks on a word, the vocabulary words and its attributes are displayed.
   Met: by double-clicking a word, the text boxes will display the vocabulary word and its attributes.

7) When the user clicks "search", an API dictionary's definition is displayed.
   Met: the "search" button retrieves a word's definition from the dictionary API and displays it.

8) Have a game to test the clients memory on their vocabulary words [3]. The goal is to guess the letters of the vocabulary words until it's completed.

-> Notify “Right!” if the letter guessed is correct.

-> Notify “Wrong” if the letter guessed is not correct.

-> Notify “No integers” if the input is an integer.

-> Notify “Done” when all the letters in the word were guessed correctly.

Met: in the game, the if-else statements verify if the input is valid (a letter) and is correct (if it's in the word they’re trying to guess). After the “enter letter” button is pressed, a message saying “Right!”, “Wrong”, or “No integers” appears. A message saying “Done” is displayed once the user guesses all the word.

9) The scores from the game are correctly and successfully added or subtracted after each input and displayed. 100 points are added for correct guesses, 10 points are subtracted for incorrect guesses.
   Met: in the game, for correct input, 100 points are added, while for wrong input, ten points are subtracted and displayed.

10) The buttons/text that is viewed by the user should have simple/short vocabulary, easy for a beginner English student to understand [3].
    Met: the buttons and labels were easy for the client to understand .

11) The program should work on the client’s laptop, it should be designed with an interface for a laptop.
    Met: the program can be successfully executed in other laptops.

12) The program includes methods, if-else statements, and loops that are annotated with comments for further development.
    Met: the code is annotated stating what it does.

Recommendations for Further Development

-Make the dictionary API not only retrieve the definition of a vocabulary word but its pronunciation as well, so the user can have time efficiency and have an accurate pronunciation displayed in the textbox and as audio.

-The game could provide and display a hint (based on the user's searched definitions or links) if five wrong guesses are made in a row.

-Make an algorithm that can classify the vocabulary words into three levels of difficulty so that the user can choose the game's level of difficulty by selecting vocabulary words from a specific English level.

-Use a timer in the game to have a limit of time to guess the word selected from the vocabulary list. This would further challenge the user's knowledge and abilities to remember the vocabulary words.

Improvement suggestions from to the client:

-Be able to upload and display images for vocabulary words [4]. Viewing images that are associated with the vocabulary words could improve the client's learning and memorization of the words' meanings.

-Make an algorithm that checks the correct spelling of the new vocabulary word to be added before being able to save it to the JSON file [4]. This can correct possible spelling mistakes made by the user while writing in the text box and avoid issues when a definition for the word wants to be retrieved but can't because the word is spelled incorrectly.

-Use an extra API that will retrieve the translation of a vocabulary word so my client won't have to input that themselves or risk the translation being wrong [4].

Word count: 497

[4] M. "English Student - Evaluation." Personal interview. 16 Jun 2023. See appendix **4**.