Criterion A: Planning

The Scenario:

My client, Magaly, is a friend who is currently learning English with online lessons. During our discussion in the interview, she stated that she finds learning English difficult because she struggles to learn new key words, what they mean, how to write them, and how to pronounce them correctly, but that it's very important for her reading, writing, and speaking skills. She said that she struggles to "keep consistency if (she doesn't) remember the previous key words" [1]. Her "studying methods are ineffective" [1] because they fail to help her memorise the words' meanings and how to pronounce them correctly. She said "it's time consuming not being able to move on to new words without remembering old ones" [1].

My client currently relies on online translators which often have errors. The notes she takes on paper are easily lost and she can't keep track of old and new content as she doesn't have it all in one place. The consequences from not solving this issue are having to revise her old notes every time she can't remember a word in time which are also disorganised. This is ineffective for her learning as it slows down her progress pace. She also claims that she's not engaged enough with the content she's learning. This issue is currently solved by associating/connecting new words with other things, which she claims helps her remember either meaning or pronunciation, but the effectiveness of this current solution is limited to the lack of organisation of her notes.

Rationale:

My client uses her home computer to do her online lessons, so an appropriate solution that helps improve her problem/situation would require the program to be used in both her school Mac laptop and her home Window desktop computer [3]. The programming language used will be Java. Java is platform-independent unlike other languages like C++ [2] which would be ideal for my client's devices. The GSON Java library also works with JSON that will help save information after the user makes any changes in the program and allow my client to easily copy the information from her Mac laptop to her Windows desktop [3].

After gathering information about the requirements to help my client, I decided to make a program that will allow her to learn new keywords in an engaging way that allows her to easily remember what she needs to learn. Unlike existing vocabulary building programs like Duolingo that have built in vocabulary that the user needs to learn before moving on, my program will let the user input new words, their meaning/definition, pronunciation, and things the user links the words to. Additionally, using a program on a laptop instead of her phone will reduce potential distractions like notifications that disrupt her learning. Through object-oriented programming, dividing the program into objects will provide structure for the program's development. This is useful to save classes (in this case the vocabulary words and their attributes) to a JSON file.

Word Count: 486

Success criteria:
After interviewing my client we came up with the following success criteria [3]

1) The client can input new words, its definition, translation, pronunciation, and “links” to the vocabulary list.
2) The client’s inputs (vocabulary words) can be saved [3] in the program.
3) The client is able to remove old vocabulary words [3]. When removed, the word, definition, translation, pronunciation, and “links” will be deleted from the file.
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.
5) A message appears in the controller if a new input/word already exists in the array.
6) When the client clicks on a word, the vocabulary words and its attributes are displayed.
7) When the user clicks “search”, an API dictionary’s definition is displayed.
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.
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.
10) The buttons/text that is viewed by the user should have simple/short vocabulary, easy for a beginner English student to understand [3].
11) The program should work on the client’s laptop, it should be designed with an interface for a laptop.
12) The program includes methods, if-else statements, and loops that are annotated with comments for further development.

[1] M. “English Student - The Scenario.” Personal interview. 17 Aug 2022. See Appendix **1**.
[2] Website comparing Java and C++ - Rationale. See Appendix **2**.
[3] M. “English Student - Success Criteria.” Personal interview. 9 Sep 2022. See appendix **3**.