**Computer Science Internal Assessment: French Vocabulary Memorization Software**

CRITERION A - PLANNING

**Defining The Problem:**

Each year, more and more people learn a foreign language, and my client--my mother--is trying to learn French through textbooks and reading French news articles. She eventually hopes to pass the DELF/DALF A2 exam--an official test of French proficiency administered by the French Ministry of Education. However, she is struggling to memorize the hundreds of French words she needs to know for the exam.

The client’s current study method relies on pen and paper: When she encounters a new word, she looks up its meaning online before jotting it down in a notebook. Occasionally, she reads through this notebook and tries to memorize the meanings of as many words as she can. According to the client, this method is tedious and ineffective. Additionally, the client has no way of knowing whether the words she’s learning will ultimately be helpful in the exam.

In our correspondence, the client also states that digitizing her learning process for the exam will be “extremely beneficial.” This is because, at the moment, the client spends a large proportion of her time on administrative tasks like maintaining her ‘vocabulary notebook’. Instead, she wants to focus on growing comfortable with French, making the learning process enjoyable, and tracking her progress. Having studied French and passed DELF A2 myself, the client asked me to create a vocabulary memorization software that is customized to her needs, so that she can prepare for the exam in an informed, efficient, and fun manner.

See Appendix A for client interactions.

, ss

*Word count - 249*

**Rationale For Solution:**

As a solution, I have proposed a French Vocabulary Memorization Software that will randomly generate vocabulary tests for the client to take in preparation for DELF A2. The software will also track the client’s progress since the previous test, offer tests of varying difficulty levels (and recommend a difficulty level based on the client’s past performance), and include an additional ‘fun’ way to revise verbs: competing against the software on a shared test.

Although the internet has many resources for learning French vocabulary, none of them are tailored specifically to the vocabulary useful for DELF A2, which is what the client wants. Additionally, they offer no way for the client to easily monitor her learning progress, and thus her level of preparation for DELF A2. My software will be a comprehensive solution that resolves both these problems.

ns

To build this software, I will be using the Flutter SDK in conjunction with the Dart programming language. Flutter is an ideal choice as it comes with a large collection of ready-to-use widgets, allowing for detailed customization of the user interface, making the final product suited to the client’s preferences and thus easy to use. Furthermore, Flutter will let me prototype new features quickly as per the client’s request, due to its low compilation time. I also plan to use Dart to code the software’s logic, as it is a well-known, stable programming language with plenty of high-quality documentation, making it easy to debug faulty code.

*Word count - 244*

**Solution Success Criteria:**

1. **Home Screen -** A screen displaying the various types of tests the user can take (tests of different difficulty levels, tests that ask for the English equivalent of a French word and vice versa etc.)
2. **Test Generation -** Once the user has made their selection on the home screen, they should be directed to the first question of the appropriate, randomly-generated test.
3. **User Input -** The user should be able to enter and submit their answer to each question in the test.
4. **Feedback on Correct Answers -** If the user’s answer is correct, they should be told so and directed to the next question in the test after a short time delay.
5. **Feedback on Incorrect Answers -** If the user’s answer is wrong, they should be told so, shown the expected answer, and directed to the next question in the test after a short time delay.
6. **Score Screen -** At the end of each test, the user should be directed to a screen where they can see their score.
7. **View Progress -** The user should be able to check their progress for each test (i.e. the change in their score from the previous test they took).
8. **Difficulty Level Recommendations -** The software should display which difficulty level it thinks the user is suited to, based on their latest test score.
9. **Difficulty Levels in Battle Mode -** The user should be able to choose the difficulty level of their ‘battle’ against the software.
10. **Battle Mode -** The user should be able to compete, or battle, against the software on a shared test.
11. **Navigate to the Home Screen -** The user should be able to navigate to the home screen from any other screen.