

1.1: Scenario/Issue

My client is the parent of a student at my school who helps his son with homework from different classes. Although my client has enough knowledge on most basic subjects, he has difficulty being able to keep up with his son's language classes due to the amount of memorisation needed and not always knowing his son's curriculum, (See interview #1). While interviewing the client, I found out that he has difficulty using language applications such as Duolingo or Babble because they lead someone down a specific path instead of letting them learn what they need, and because they have a learning curve to them to be able to fully utilize their potential. Additionally, he informed me that manual methods of trying to learn are both costly in time and effort for giving very little reward. Hence, I chose to discuss possible solutions for the parent's problem to solve some of the pain of language acquisition.

1.2: Computational Solution

After consulting with the client, it was obvious that a computational solution was most viable because of the nature of the problem. Analog methods of studying such as using paper or flashcards take a large amount of time and do not allow for diversifying methods of studying vocabulary without requiring an obscene amount of effort. Other than the time consumption, the creation of such study methods is incredibly repetitive and difficult to continue for a long time. The automation that is allowed by using a computational solution would cut down the time by a lot and it would allow for much more diverse studying methods. Other than fixing the problems of analog solutions, a custom computational solution would allow the user to make their own decisions of what to learn instead of going along with what the application prescribes for them.

For this application, I will be using C++ with the Qt desktop application framework. I am using C++ because of the fact that it is a multi-paradigm programming language, allowing for procedural, functional, and object oriented programming. This flexibility, as well as the memory management of C++ allows for efficient, but not overly convoluted code. The Qt framework is

ideal for a desktop application because of it being universal across different platforms, allowing for a program to be on Windows, Mac, and Linux. For the backend database, I will be using SQLite. SQLite is a local database which does not take up a lot of space. This is perfect because it allows the user to avoid having a connection without the tradeoff of losing storage. SQLite also does not require installation, only being two files that can be included with the project. Apart from that, it allows for data persistence and quick look up times through SQL. My choice of IDE will be Visual Studio Code as well as Qt Creator. Visual Studio Code allows for many different file types to be edited and also just being a comfortable editor, while Qt Creator allows for the creation of GUI elements without too much of a hassle with code.

1.3: Success Criteria

- The application allows the user to create and manage custom vocabulary lists with capabilities of holding over 1000 words in a list
- The system supports at least 3 world languages such as Spanish, Latin, and German.
- The application must provide at least 3 testing modes: Showing a word and presenting multiple choice options for its meaning/translation, showing a meaning/translation and requiring the user to provide the word, flashcard study methods.
- The application supports bidirectional testing, from one language to another and the other way around
- The system must have a spaced repetition system with an algorithm for prioritizing studying certain words based on certain parameters: last date studied, correct/incorrect, time taken to answer.
- Definitions/meanings of words should include multiple ways of being stored: synonyms and antonyms, contextual interpretations, and other methods of recognition outside of dictionary-style definitions
- The user must be able to view a report/summary of the words/list they have been studying or currently learning
- The AI integration allows the user to generate new vocabulary lists through prompts
- The AI feature must allow the user to automatically provide definitions for a user-provided list of words
- The AI feature must be able to extract unknown words from a block of text provided by the user that are not yet added to other lists