Software requirement specification document for English language learning application Tap Learn

Boghos Youseef

Alex Borgognoni

Computer Science Bachelor

University of Luxembourg

October 11, 2020

-Index

1- Introduction

Globalization has led to the discovery of many talents across countries and cultures. Yet, language barrier remains the number one hindrance for fully efficient exploitation of those talents.

1.1 Idea

Usually people who learn a new language translate every new word they learn into their native tongue. This creates a delay in learning the language. People learning languages tend to translate first into their mother-tongue and then understand the meaning intuitively. This creates a hurdle in the way of people who wish to learn the language for work necessities. What if there was a way to transcend this old-fashioned way of learning languages?

What if users could learn English in a similar way as a native English speaker? The idea is to inject the ideas and thoughts and concepts directly from the English lexicon into the brain, without the need for the delayed translation in-between languages in the brain. This is the basic Idea of Tap Learn.

2. Product description

2.1 Pre-requisites

The app was developed using Flutter frame work with the programming language Dart. The app can be transformed into a standalone executable but for now it is in beta phase which requires the following pre-requisites in order to get the app to work:

a) flutter sdk
b) dart language package
c) an emulator (in our case we used the built-in emulator of android studio)

2.2 Design

Tap Learn works by implementing a card system that allows users to choose the type of vernacular they would like to learn based on the environment in which that vernacular is used. each card presents a single word with a photo included to showcase the word being presented to the user. Most apps nowadays that offer a full course on any targeted language usually push users to give up after forcing them to go through the typical “boring” first stages of language learning and not giving them the free agency to choose what they wish to learn. Tap Learn circumvents that by immediately immersing the user in the environment of their choice, giving them access to potentially hundreds of words that are suited for each particular user’s immediate need.

3. Requirements

The only requirements presented by the client were the following:

- An app that teaches the English language.

- Must include, or have the possibility to scale to, 500 words to teach the users.

Therefore, functional and non-functional requirements were introduced by my colleague and I, Tap Learn developers.

3.1 Functional Requirements (FR)

3.1.1 The application must connect to the internet to showcase photos.

3.1.2 The application must allow users to easily switch between environments through the scrolling functionality.

3.1.3 The application must allow users to easily switch between tabs and enter into environments with a single tap but also allow swiping between tabs.

3.1.4 The application must implement a user-friendly interface.

3.1.5 The application must implement a cards system with tap-able components.

3.1.6 The application must display to users their choice of word environments upon launching the application.

3.1.7 The application must present a tab for environments and a tab for exercises related to those environments.

3.1.8 The application must contain a large data-set such that it satisfies a wide range of users and their needs.

3.1.9 The application must implement audio files in buttons on each card with the proper pronunciation, recorded and ready to play, of the words presented on the cards.

3.1.10 The application must implement an action button that allows the addition of extra or new environments.

3.1.11 The application should be a standalone executable that can run on any android device.

3.1.12 The application should be deployable through playstore.

.

3.2 Non-functional Requirements (NFR)

3.2.1 The application must be completed with a full requirements analysis document in the duration of 48 hours.

3.2.2 The application must present the users with their choice of English words depending on the environment they are most commonly used in.

3.2.3 The application must accurately represent the written concepts in images.

3.2.4 The cards implementation should be done so that the user can easily swipe right for the next card or left to go back to the previous card.

3.2.5 The application must not be CPU extensive.

3.2.6 The application must allow the easy and user-friendly navigation through scrolling and swiping functionalities.

4. Evaluation

4.1 Functional Requirements Evaluation

--The application fully satisfies the following functional requirements:

- 3.1.1
- 3.1.2
- 3.1.3
- 3.1.4
- 3.1.5
- 3.1.6
- 3.1.7

--The application satisfies partially the following functional requirements:

- 3.1.9 The application implements the audio buttons but due to lack of suitable audio data-base the audio files are not present.
- 3.1.10 The application implements the button but due to time constraint there were no additional environments prepared for addition.

--The application does not satisfy the following functional requirements

- 3.1.8 The application does not have yet a big enough data-base due to time constraint on producing the application (NFR 3.2.1)
- 3.1.11 The time constraint led to concentration of effort on basic idea behind the app and not the fully operational version.
- 3.1.12 since the application is not a standalone executable yet it cannot be deployed through playstore.

4.2 Non-functional Requirements Evaluation

--The application fully satisfies the following non-functional requirements:

- 3.2.1
- 3.2.4
- 3.2.5
- 3.2.6

--The application satisfies partially the following non-functional requirements:

- 3.2.2 The application implements the words-accessibility mentioned in NFR 3.2.2 but it due to lack of a complete and working data base it does not provide a full spectrum of word choices to users.
- 3.2.3 some written concepts cannot be accurately represented in images (e.g Agency of humans, free will, equality, etc)

5. Evolution and future development

The application can easily be implemented while satisfying all of the NFRs and FRs if given enough time.

The application has many future directions it can take. Here are some examples:

- The application can implement a social networking aspect where users share their achievements and or play games together.
- The application can also implement a video calling and texting capabilities with other users.
- The application can be evolved to include videos to explain concepts that cannot be fully represented with images and audio files.