



University of Groningen

Bachelor's Thesis

Building Blocks for Online Language Practice Platforms

Martin Avagyan

Primary Supervisor

Mircea Lungu

Secondary Supervisor

Alexander Lazovik

July 2017

Abstract

This thesis describes the process of designing and implementing a family of interactive exercises, with the aid of relevant research from the field of linguistics. The author of this work has chosen a way of implementation, the reasons for which will be described in this thesis, along with the chosen architecture. Furthermore, this work also aims to illustrate how the selected approach provides the family of exercises with flexibility and maintainability, including the re-usage of their building blocks to construct new exercises. Finally, the paper will present how the evaluation of these exercises by more than one hundred users proved the project's usability.

Contents

1	Introduction	1
1.1	Structure	2
2	Related work	4
2.1	Vocabulary acquisition	5
2.2	Related applications	6
2.2.1	LinguaLeo	6
2.2.2	Babbel	7
2.2.3	Anki	8
2.3	Zeeguu	9
2.3.1	Zeeguu exercises	10
2.3.2	Zeeguu UMR	10
2.3.3	Zeeguu Chrome Extension	11
3	Problem Description	12
4	Realization	14
4.1	Starter page	14
4.2	Exercises	15
4.2.1	Exercise 1: Find	15
4.2.2	Exercise 2: Choose	17
4.2.3	Exercise 3: Match	18
4.2.4	Exercise 4: Translate	19
4.3	Exercise components	20
4.3.1	Text pronunciation	20
4.3.2	Exercise options	21
4.3.3	Feedback	21
4.3.4	Exercise visual feedback	22
4.4	Exercise sets	23

4.5	No exercises and 404 feedback	24
5	Architecture	25
5.1	Back-end	25
5.2	Front-end	26
5.2.1	Memory leak prevention	26
5.2.2	Architecture overview	27
5.3	Exercise architecture	28
5.4	Design blocks	29
5.4.1	ProgressBar class	30
5.4.2	Speaker class	30
5.4.3	Feedback class	30
5.5	Publish subscribe module	31
5.6	Generator module	32
5.7	Validator module	33
5.8	Static classes	33
6	Results and Evaluation	35
6.1	Heat-map analyses	35
6.2	General observations	37
6.3	User activity	38
6.3.1	User interaction	39
6.3.2	Hint usage	40
6.3.3	In-exercise feedback	41
6.4	Learning experience	42
6.4.1	Learning process	43
6.5	Explicit Perceived Usefulness	45
7	Conclusion	47
7.1	Future work	47
A	Setup Overview	49
	Bibliography	50

Chapter 1

Introduction

Learning a new language can be a challenging task, which leading experts in the field have tried to tackle from different angles. One of these approaches involves Internet-based learning applications. In the past decade, there have been many successful implementations of these tools, which, overall, seems to be based on the same basic principles: helping the learner master a new language by providing them with a predefined set of exercises to improve their vocabulary, grammar, and correct usage of words. However, this approach has apparent limitations, one of them being the fact that there is no average user, and that, therefore, the predefined set of words does not always reflect the knowledge of the user. These applications, thus, become dependent on algorithms that aim to determine the user's proficiency level. Moreover, even if they are suited for the user's language level, they might fail at reflecting their interests, making the overall activity less enjoyable.

An alternative approach to overcome this limitation is to use personalized learning, a well-discussed topic in the scientific community. One of the focuses of the project described here was, therefore, to provide exercises that adapt to the users' preferences, which can be achieved through the development of a language practice tool in the form of an online practice platform that can integrate with other APIs. Keeping in line with the idea of personalized learning, the goal was to remain flexible enough to accommodate custom exercises, and to combine different types of exercises in the process of customization. This was to be achieved through determining the optimal exercise combinations by analyzing user feedback.

This project, attempts to address those issues in the following research questions. The first research question focuses on the software design of a family of language

practice exercises, while the second research question focuses on the usability and usefulness of the platform:

RQ1 *Can we identify basic building blocks that can enable the construction of a large variety of engaging language practice exercises, while also utilizing the user input for further computation?*

RQ2 *How useful are these identified building blocks as the basis for an exercise platform?*

2.1 *How do people use the platform?*

2.2 *How do people actually learn using the platform?*

1.1 Structure

This document describes the Online Practice Platform. The next sections are focused on describing the process of analyzing, developing and evaluating the essential and minimal building blocks for a language practice tool.

Thus, this paper is structured in the following manner:

1. **Related work**

This section investigates the research done in the field, the related applications that solve similar problems, and the direction of this project.

2. **Problem description**

This part defines the problem and lists the necessary requirements to solve the problem.

3. **Realization**

This section describes the solution to the problem and its visual implementation.

4. **Architecture**

This section explains the underlying architecture behind the platform.

5. **Results and Evaluation**

This part discusses the evaluation with over one hundred users learning a new language. It describes the insights gained from the validation process and the improvements derived from it.

6. **Conclusion**

This section reiterates the highlights of this paper and draws the final conclusions.

7. **Future Work**

This section discusses possible enhancements for the system.

Chapter 2

Related work

In 1620, the English scientist and author Francis Bacon wrote:

"If you read a piece of text through twenty times, you will not learn it by heart so easily as if you read it ten times while attempting to recite from time to time and consulting the text when your memory fails."[1]

The idea suggests that stimulating the active recalling of words while learning will help to improve vocabulary acquisition. This scientific finding has been termed the testing effect, which suggests that, in order to increase the efficiency of one's long-term memory, one needs to allocate some learning time for retrieving the to-be-remembered information through testing with proper exercises.[3]

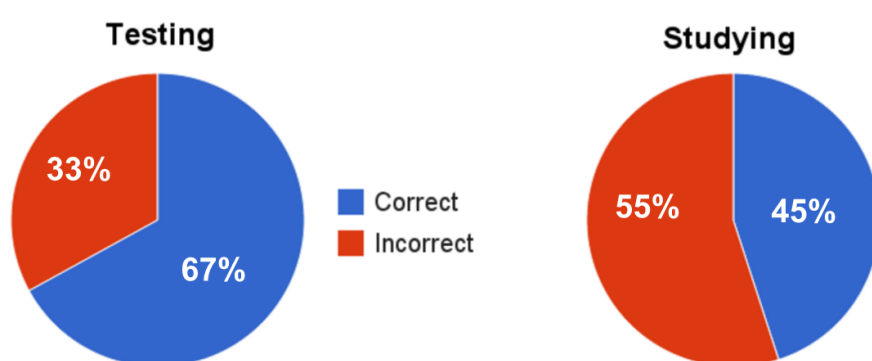


Figure 2.1: Study conducted to show the effectiveness of the testing effect

This is particularly important for individuals who wish to practice their vocabulary while learning a new language. A recent study [1] suggests that one can improve the correct recalling of words from 45 percent to 67 percent if, alongside learning

new information, sufficient time is allocated for testing knowledge (Figure 2.1). This idea, the creation of a set of tools which could provide the user with exercises that strengthen long-term memory, served as the inspiration for this project. The following subsection will discuss how this problem could be approached from the linguistic point of view.

2.1 Vocabulary acquisition

There has been a significant amount of research in the field of linguistics discussing the best way for vocabulary acquisition. However, for the sake of brevity, the following description will focus on highlighting the main ideas of one of the leading researchers, Paul Nation. He proposes that "[k]nowing a word involves knowing its form, meaning and use." [7] By breaking down this finding, we derive the following three aspects about knowing a word.

- **The form**

In order to know the form of a word, one needs to:

- Know what the word sounds like and looks like.
- Know how the word is pronounced, written and spelled.
- Recognize parts of the word.
- Recognize the minimum parts needed to express the word's meaning.

- **The meaning**

In order to know the meaning of the word, one needs to know:

- The meaning the word signals.
- What is included in the concept of the word.
- What items can the concept refer to.
- The other words the reader associates with the learned word.
- Alternative words to replace the word.

- **The use**

In order to know how to use the word, one needs to know:

- The patterns that the word occurs in and patterns that must be used with the word.

- The words that occur with the learned one and the words that must be used with it.
- The word’s frequency of occurrence.
- The word’s frequency of usage.

The above-mentioned summary, therefore, provided the linguistic basis for this project, and served as the starting ground for understanding the minimal blocks required for the practice exercises.

2.2 Related applications

As it has been discussed above, when learning a new language, it is paramount to the learner to continuously practice their skills, ideally through engaging exercises that challenge their memory. This has been proven by a large number of applications that support such activities. However, most of these applications serve the goal of accumulating profit (Duolingo¹, Babbel², etc.), or are standalone applications that cannot interact with other platforms. Besides web-based applications there have been number of reasearch into e learning research tools however they come with limitations, such as the fact that they usually focus on a single language, and limited set of exercises [5]. In this section, we will briefly discuss the relevant work that proved to be successful in the past years, including both traditional and non-traditional approaches to vocabulary acquisition.

2.2.1 LinguaLeo

LinguaLeo³ is a web-based platform that offers an English learning service for multiple languages. It personalizes the user’s learning program with the help of multimedia content, such as news articles and videos. The user can practice their language via learning exercises.

¹<https://www.duolingo.com/>

²<https://www.babbel.com/>

³<https://lingualeo.com>

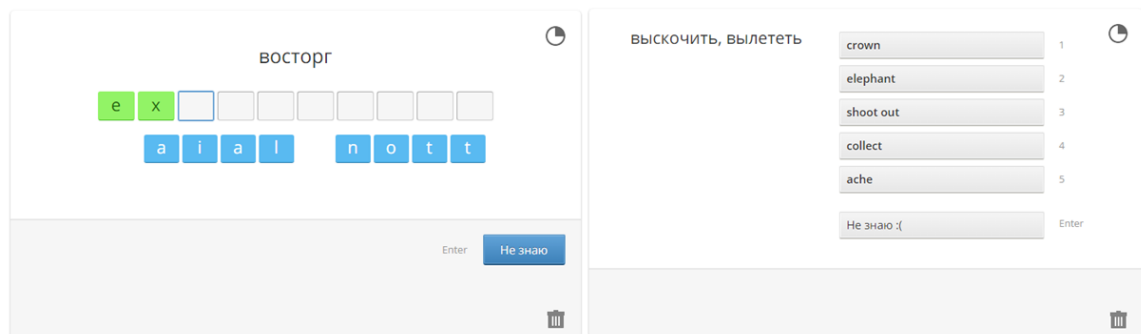


Figure 2.2: Two of the exercises offered by LinguaLeo

The personalizing aspect of the platform is similar to the concept that this project aims to achieve; however, LinguaLeo is a closed-source, freemium application that provides only English learning experience to its users.

- **Advantages**

1. Personalized learning by letting the user build their vocabulary list.
2. Interactive variety of exercises.

- **Limitations**

1. Closed-source,
2. Freemium platform.
3. Only English learning experience.

2.2.2 Babbel

Similar to LinguaLeo, Babbel is an e-learning platform. It provides language learning exercises that are designed by educators and linguists. The main method of practice is through predefined exercises. While the success that this platform enjoys is undeniable, the limitations of this approach have been discussed in the previous chapters - there are no average users, therefore, tailoring the learning experience to a non-existent average user may decrease the effectiveness of the method.

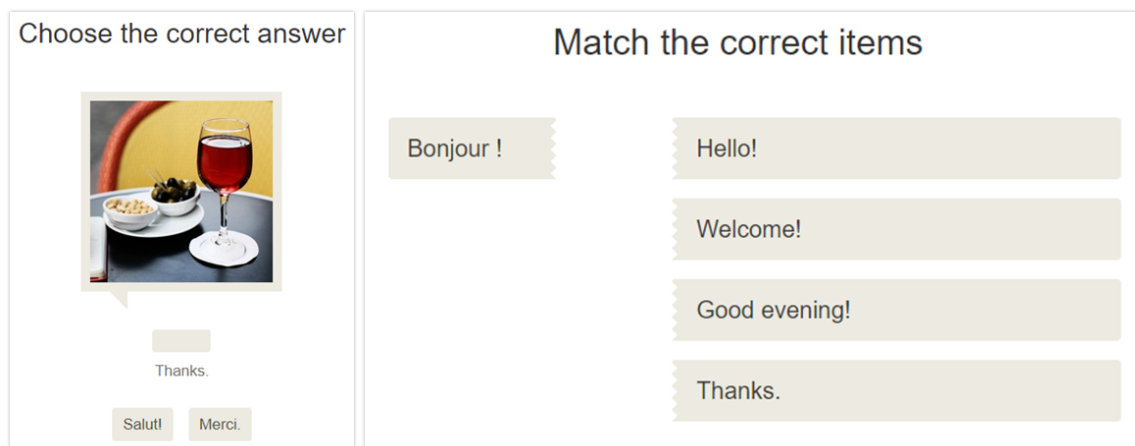


Figure 2.3: Two of the exercises offered by Babbel

Another difference between this method and an language platform is the fact that the platform is premium and closed source.

- **Advantages**

1. Tailored variety of practice exercises.
2. Interactive variety of exercises.

- **Limitations**

1. Closed-source.
2. Premium platform.
3. Predefined set of exercises.

2.2.3 Anki

Anki is a flashcard program. The program employs the space repetition method for learning activities. Each card is constricted by three fields, the expression field in the target language, the pronunciation field an the meaning of expression in desired language.

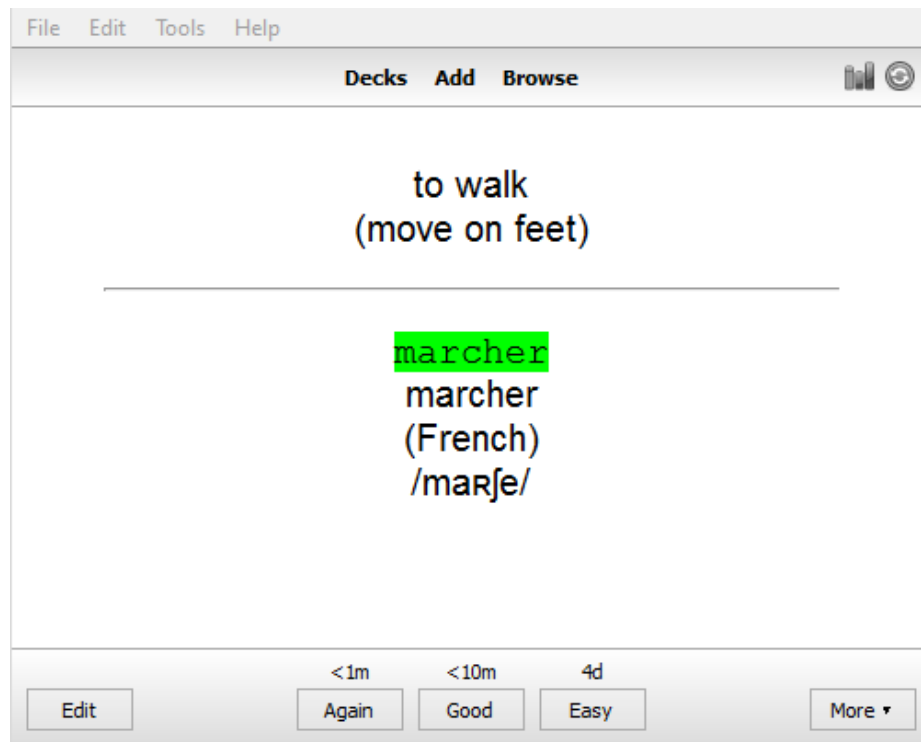


Figure 2.4: Flashcard example from Anki

- **Advantages**

1. Variety of personal exercises constricted by the user.
2. Open source platform.

- **Limitations**

1. Stand alone desktop application, not available for the web.

2.3 Zeeguu

Zeeguu[6] is an open source language learning platform. Currently, Zeeguu offers language learning with the aid of reading articles and practicing vocabulary. The next sections briefly describe two services offered by Zeeguu that are directly related to this project.

2.3.1 Zeeguu exercises

Zeeguu exercises were the initial implementation of a practice service implemented for Zeeguu platform. The service provided contextual personalized exercise, however there was only a single type of exercise implemented in there.

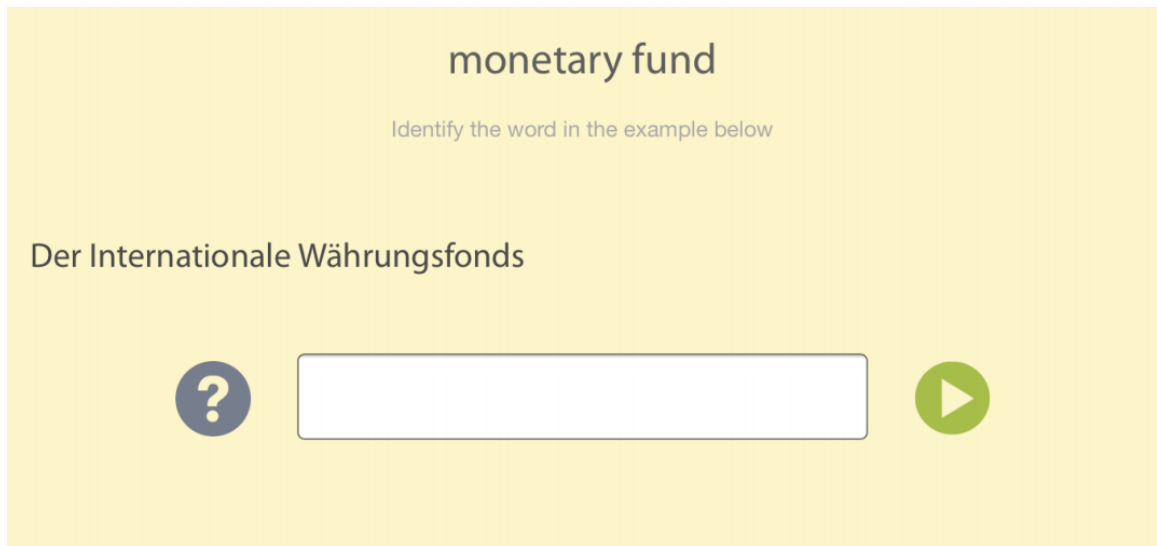


Figure 2.5: Initial implementation of Zeeguu exercises.

- **Advantages**

1. Contextually personalized exercise.

- **Limitations**

1. Only one exercise type available.

2.3.2 Zeeguu UMR

Zeeguu UMR (Unified-Multilanguage-Reader) is a web application developed by two students, described in more detail in the following Bachelor's thesis: L.A.H van den Brand [9] and D. Chirtoaca [2]. The application implements a multilingual article reading and language learning service. The interface allows for translations of unknown encountered words. Those translations, or bookmarks,⁴ serve as an input for the practice platform.

⁴Translations are also known as bookmarks, hereinafter, bookmarks refer to user translations.

2.3.3 Zeeguu Chrome Extension

Similar to Zeeguu UMR, the chrome extension allows the user to translate words from any website if the extension is activated for particular page. Those translations then can be bookmarked, and become an input for practice platform. ⁵

⁵<https://chrome.google.com/webstore/detail/zeeguu/ckncjmaednfephhbpeookmknhmjjodcd?hl=en>

Chapter 3

Problem Description

In the previous sections, the limitations and flaws of available systems were discussed. This section lays down the functional(FR) and non-functional requirements (NFR) that were devised for the project. In order to attain the goal, the platform should adhere to the following functional requirements.

1. The learning platform should be comprised of exercises, which should be based on textual interactions.
2. The context gathered from the past readings of the user should serve as inputs for the system.
3. The system should be able to accommodate the majority of the vocabulary learning aspects from Section 2.1.
4. In addition, the designed system should be flexible enough to accommodate a combination of exercises (of different types), in order to provide customization for different users.
5. The system should provide a way in which user feedback is collected within each exercise for further analyses. The analysis will help to improve the algorithm behind the exercises, and to address issues related to them.
6. The user should have control over individual exercises, such as removing the exercise or reporting a shortcoming.

Alongside the functional requirements, the system should prioritize the following non-functional requirements.

1. The platform should prioritize four main system design goals, which are the design requirements that the system should adhere to. The system design should:

- (a) minimize complexity.
 - (b) be intuitive, and should result in an easily extendable and maintainable code.
 - (c) prioritize efficiency.
 - (d) consider easy integration with external API's.
2. The platform should be web based to enable cross-platform usage.
 3. All the user interface (UI) elements should have a responsive design to allow usability on various screen sizes. The design has to maximize the screen estate on small screen sizes by altering UI components.
 4. The user interface of the platform should be intuitive for the user to minimize the provided textual guidance.
 5. All the platform-dependent calculations and generations should be instantaneous¹ to provide seamless behaviour. While waiting for external requests to return resources, the system should provide visual loading reference.

Keeping this in mind, and taking into account the problem description in the introduction, the primary goal of this project is, therefore, to face these limitations by producing an **open, multilingual practice platform that can be integrated with external API's, while also providing the user with an engaging experience.**

To address and validate all the requirements, this project was integrated with Zeeguu-API ². It utilizes the resources of the platform, namely, the bookmarks described in Section 'Related work.'

¹less than 200 ms.

²<https://github.com/mircealungu/Zeeguu-API>

Chapter 4

Realization

This chapter is dedicated to describing the solution to our problem (defined in the previous section) by looking at the individual components that comprise the practice platform. While designing those components, two design principles were followed:

1. **Discoverability:** when looking at a design element, the user should be able to discover what operations the visual reference can perform.
2. **Feedback:** each visual element should respond to user's interactions by signaling what happened.

These are the fundamental principles that are derived from the Human-Centered-Design (HCD)[4] framework.

It is important to note that all the design components presented in this section have a responsive design, therefore, they adapt to any screen size. However, this won't be discussed in detail here, in order to keep the document concise.

4.1 Starter page

The starter page is the first screen the users are exposed to, therefore, a deeper understanding of how users interact was derived from usage data. The goal was to create a minimal user interface that is both promising and intuitive. On the page, the user is presented with options to practice exercise sets. Exercise sets are combinations of multiple exercises that will be described in the sections below. There are three practice sessions available, offering 2, 3, and 5 minutes of practice and containing 10, 15 and 25 bookmarks, respectively.

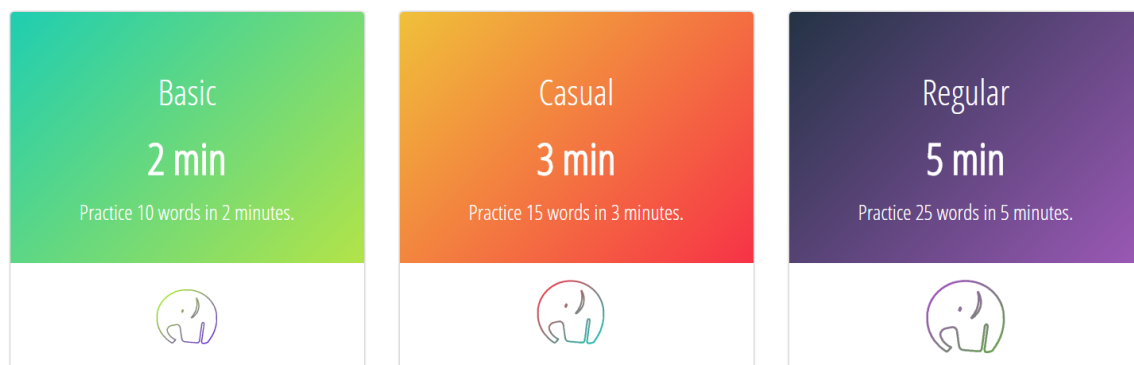


Figure 4.1: Starter screen

Figure 4.1 depicts this implementation. Each package has three main components: name, time, and information. As stated above, the time suggests an approximation of how long it will take for the user to complete the exercise. The information provides the user with minimal guidance as to what to expect when a package is clicked on. The validation section below will further elaborate on how the adapted design proved to be more efficient than an alternative implementation.

4.2 Exercises

In order to discover possible basic interactive components that can allow generation of new types of exercises, a decision was made to start by implementing four different types of exercises in order to then analyze their commonalities and differences. The following sections provide a description of those exercises.

4.2.1 Exercise 1: Find

In the 'Find' exercise (Figure 4.2), the user is presented with the translated word and is asked to find the original word in the context of the original text.

Precondition

The minimum number of bookmarks required to generate the exercise is one (precondition 1 (P1)). As it can be seen from the example below, the exercise depends on the translated word, the context, and the original word (P2).

Interactive Elements

The exercise card is equipped with the following interactive elements (IE):

IE1: Ask for hint¹.

IE2: Check the answer.

IE3: Word pronunciation².

IE4: Control over the exercise card: report or delete the exercise.

IE5: Select-able text.

IE6: Input box.



Figure 4.2: The user interface for exercise 1 (find).

Considerations

To improve the visual appeal of the exercise, user feedback was considered. If the context text is multiline, the context alignment format will be formatted as *justify*. Otherwise, the context text is aligned to the *center* of the card. The same principle is applied to exercises with similar visuals.

¹The hint strategy for the exercise is to provide the expected answer.

²Some of those elements were added later on after considering the user feedback.

4.2.2 Exercise 2: Choose

The 'Choose' exercise (Figure 4.3) is designed to provide the user with the context and 3 options to complete that context. The user is tasked with choosing the correct word that fits the context.

Precondition

It is required to have a minimum of 3 bookmarks for generating the exercise (P1). For one of the bookmarks the exercise utilizes the context for all 3 of them it uses the translation and the original word (P2).

Interactive elements

The exercise card is equipped with the following interactive elements:

IE1: Ask for hint.³

IE2: Selection of any of the provided options.

IE3: Word pronunciation.

IE4: Control over the exercise card: report or delete the exercise.

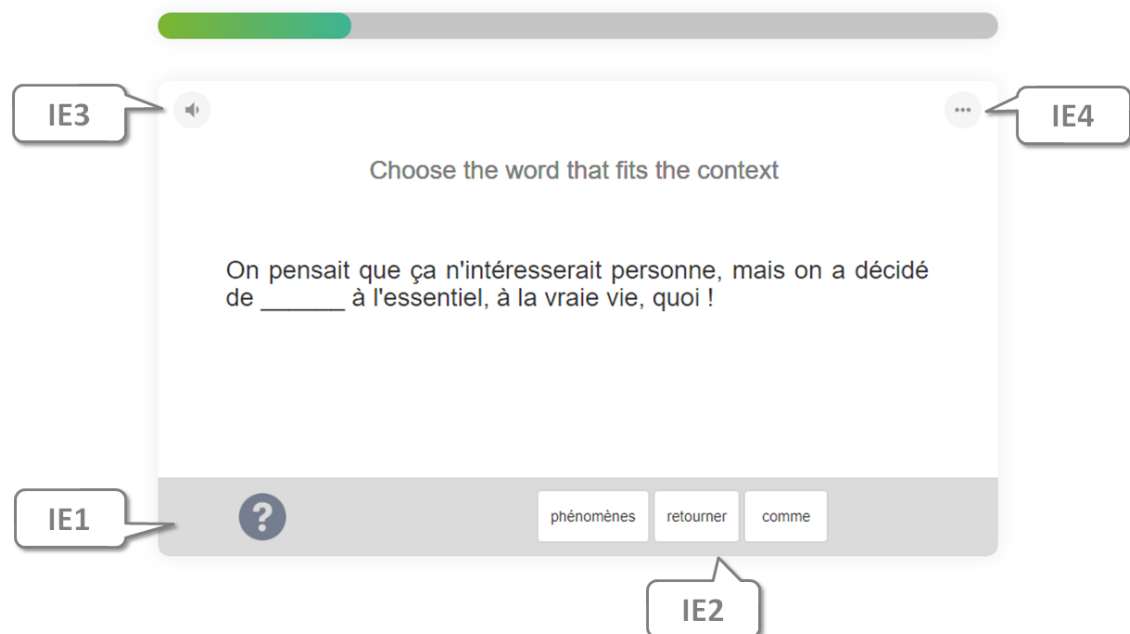


Figure 4.3: The user interface for exercise 2 (choose).

³The hint strategy for this exercise is to reduce available options by one.

Considerations

The principle applied in exercise 'Find' is used here as well, to *center* or *justify* the exercise context.

4.2.3 Exercise 3: Match

The 'Match' exercise (Figure 4.4) provides the user with 3 words in the original language and in the translated language, and the user is tasked with matching the word with its translation.

Precondition

Naturally, the exercise requires a minimum of 3 bookmarks for generation (P1). For each bookmark the exercise utilizes the translation and the original word (P2).

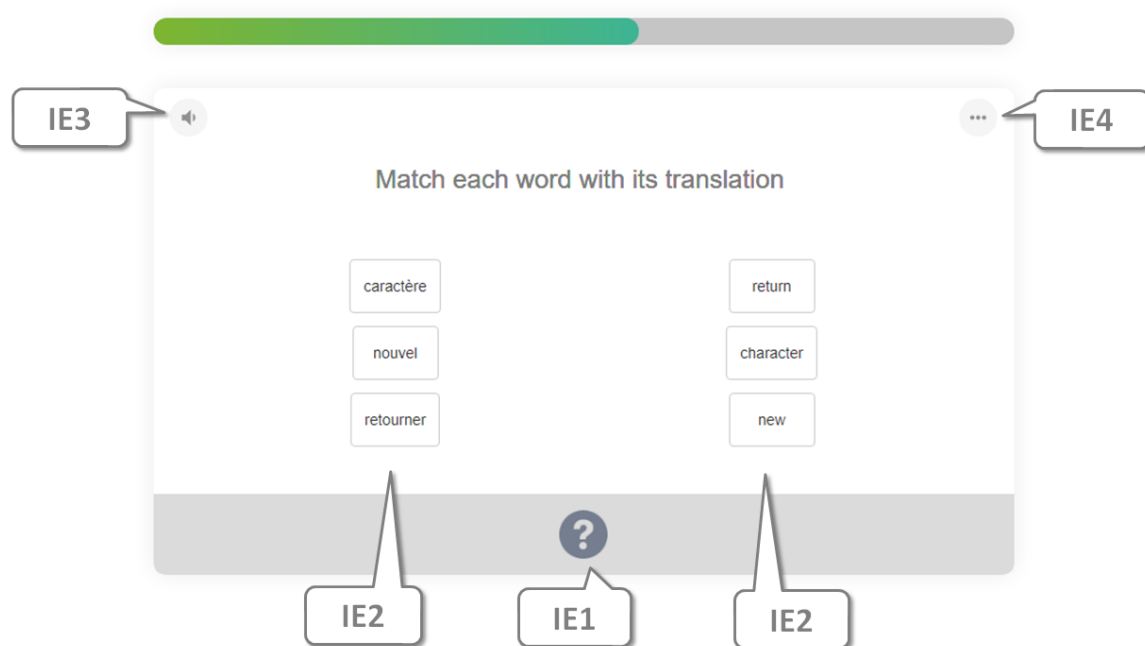


Figure 4.4: The user interface for exercise 3 ('Match')

Interactive elements

The exercise card is equipped with the following functions:

IE1: Ask for hint⁴.

⁴The hint strategy here disables one of the pairs.

IE2: Selection of an option from the left/right column and from the right/left column in order to match the given pairs.

IE3: Word pronunciation.

IE4: Control over the exercise card: report or delete the exercise.

Considerations

The options in each column are displayed vertically. An alternative implementation that used horizontal display proved to be less usability on different sized screens.

4.2.4 Exercise 4: Translate

The 'Translate' exercise (Figure 4.5) provides the user with the the word in the original language, and asks the user to translate the word.

Precondition

The user is also presented with the context in which the word appears, where the to-be-translated word is highlighted in the context (P1). The context serves as a visual aid for the user to grasp the idea of the context, and to derive the word's meaning. The minimum number of bookmarks required to generate the exercise is one (P2). The exercise card is equipped with the following functions:

Interactive elements

IE1: Ask for hint⁵.

IE2: Check the answer.

IE3: Word pronunciation.

IE4: Control over the exercise card: report or delete the exercise.

IE5: Input box.

⁵The hit strategy here gives the expected answer in the input box.

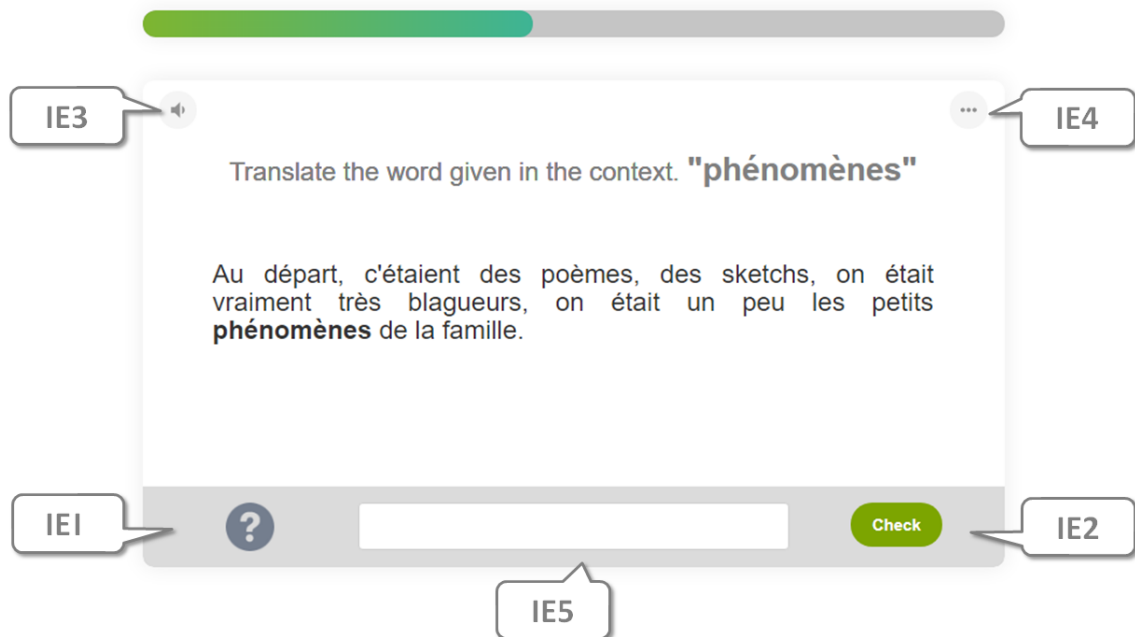


Figure 4.5: The user interface for exercise 4 ('Translate')

Considerations

The principle applied in exercise 'Find' is used here as well, to *center* or *justify* the exercise context.

4.3 Exercise components

The previous section explained the composition of each exercise type and described the functionality provided by each card. In this section, the blocks and components that comprise these exercises are described.

4.3.1 Text pronunciation

In the 'Related work' section, it was discussed that, in order to know the form of the word, the learner has to be able to know how the word sounds like. Therefore, word pronunciation became a feature that each exercise needed to include. Each exercise card (Figure 4.6) includes a sound icon, which, when clicked on, will pronounce the relevant word for each exercise type.

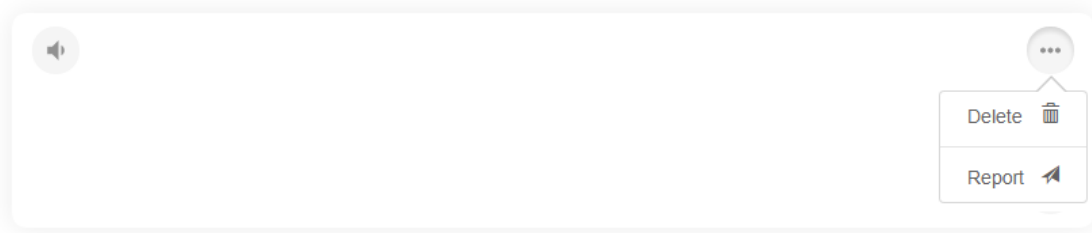


Figure 4.6: Exercise card header, sound icon to the left and option panel to the right.

4.3.2 Exercise options

In order to give the user control over the exercise card, there was a need to have an intuitive visual reference. The option module is a shared component among all the exercises, and is present at each exercise card. The module provides the common functionalities to the exercises, such as giving a feedback or deleting the exercise card. This functionality list can be extended for each exercise type.

4.3.3 Feedback

An important aspect of the platform is to give the user the possibility to provide feedback to the system. The feedback (Figure 4.7) is divided into two categories: predefined feedback and custom feedback. The predefined options are used to improve the the algorithm that provides the bookmarks. It includes feedback options, such as "the exercise was too easy" or "I knew the exercise". The custom feedback is evaluated manually, and conclusions are drawn in order to improve the platform. An example of an improvement based on user feedback is presented in the 'Validation' section of this paper.

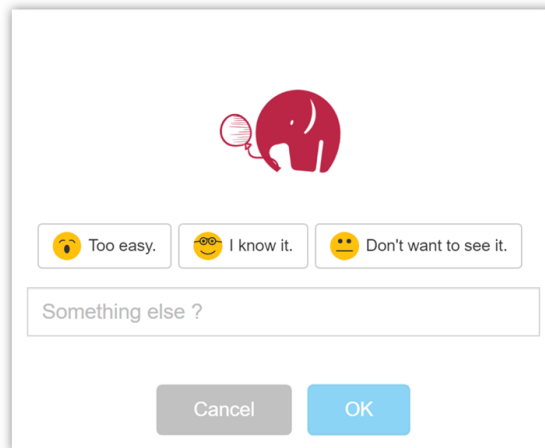


Figure 4.7: Exercise feedback

Feedback can be provided before or after the exercise is completed. Once the exercise is completed, visual reference to the feedback is displayed to the user. This is done by replacing the exercise footer with a secondary footer (Figure 4.8). This makes sure that the user is aware of the feedback option even after the exercise is completed.

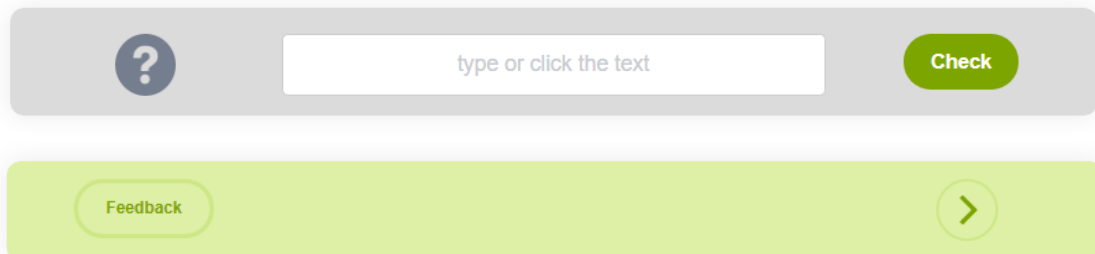


Figure 4.8: The primary footer of the exercise (top) is replaced by a secondary footer (bottom) when the exercise is successfully completed.

4.3.4 Exercise visual feedback

Following the second principle of HCD, each exercise has to display visual feedback for given actions. Therefore, when the user gives the wrong answer to an exercise, a wrong answer animation is displayed (Figure 4.9 top). The animation shakes the focused element for 1 second as an indication that the answer is not accepted.

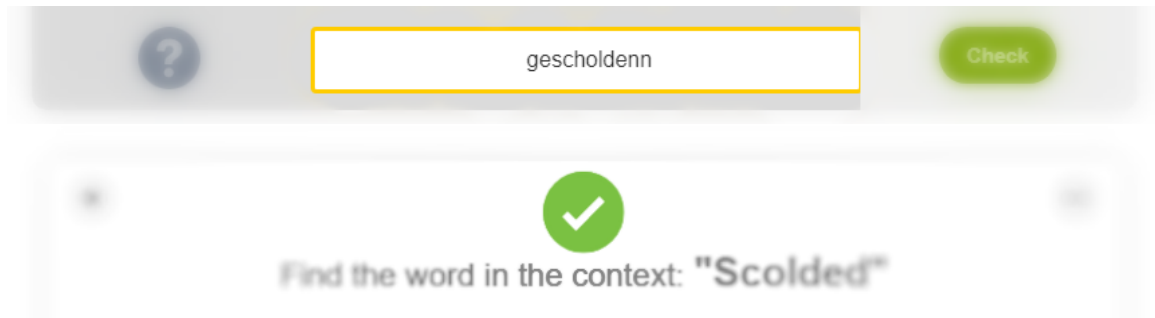


Figure 4.9: Wrong answer animation shake (top), correct animation bottom.

When the user chooses the correct solution, a check-mark animation (Figure 4.9 bottom) is presented to the user, which later continues with the progress bar animation.

4.4 Exercise sets

As it was discussed in the section "Problem description" it is required to combine different types of exercises into one package, those are called exercise sets. In the home screen in Section 4.1, the displayed exercise packages correspond to exercise sets. The exercise examples presented in Figures 4.2, 4.3, 4.4 and 4.5 show a progress bar present above each exercise card. The progress bar aims to provide the user with visual feedback on their current state in the exercise set. When the exercise set is completed, the bar is filled.



Figure 4.10: Progress bar

Once the exercise set is completed, the user is presented with the option block (Figure 4.10). The screen has 2 main components: the time taken for the user to complete the exercise set, and the possible actions that can be performed. There are two possible actions that the user can take: 'go to home screen' and 'repeat the exercise set.'

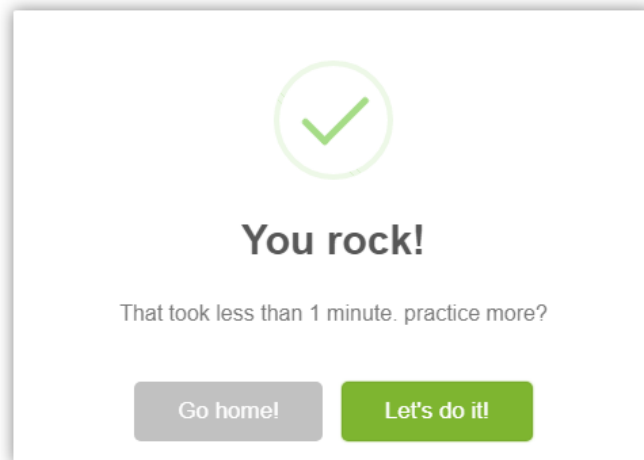


Figure 4.11: Exercise completed page

4.5 No exercises and 404 feedback

It was mentioned in the 'Problem description' section that the platform should provide the user with visual feedback at all times. Since the platform depends on input that comes from external APIs, it is possible that, at certain times, there won't be any input available, or the input will not be usable for the platform. To prevent poorly generated exercises, the platform analyzes the incoming input, and then provides the user with appropriate visual feedback, such as the examples in the figure below. In the first case, the user is presented with the 'not enough input' page. This happens when the input received from the external API does not satisfy the minimum requirements for generating exercises. In the second case, when typing a wrong URL, the user is redirected to the 'not found' page.



Figure 4.12: The empty pages.

Chapter 5

Architecture

The practice platform is built as a web-based service. On the back-end, the platform relies on the Python Flask micro framework, and on the front-end, it relies on ECMAScript 2015 (ES6). The setup overview in Appendix A briefly describes how these tools are bundled in order to create a powerful development environment.

To meet all the requirements listed in the previous sections, the platform was developed in the form of a single-page application. This approach allows the system to be flexible, and gives greater control over the user interface. Besides, focusing on the client's side allows for the full usage of JavaScript, and thus opens the option of benefiting from the object-oriented aspects of the language. In addition, one big advantage of the approach is load-balancing, which lowers the load on the server's side, as it heavily relies on the JavaScript that runs on the client's browser.

5.1 Back-end

The back-end acts as a Flask Blueprint that has a single endpoint.¹ The endpoint expects HTTP GET requests, and directs all the requests by loading the *html* template containing the main entry point. This leaves the client's side responsible for the further handling of the requests. Given the nature of the application, the main objective of the server is concluded by template-loading and by validating users, in case the session is not found in the cookies of client's browser.

¹The source code has two endpoints, the second is dedicated to debugging locally, therefore, it is not part of the framework.

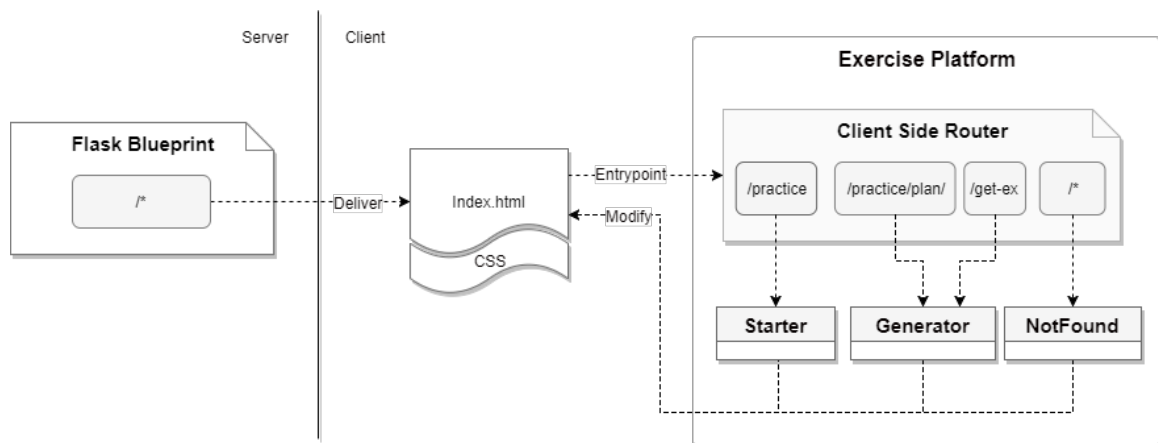


Figure 5.1: System overview

5.2 Front-end

The application starts from the server loading the *index.html* template in the browser. Within that file, the application's main entry point *app.js* is loaded along with the related styles. From this point on, all the relevant *html* content is loaded by JavaScript, using Ajax requests. The application relies on client-side routing in order to maintain functional URLs while also avoiding a full-page reload. For that, the server-side and the client-side have to cooperate. Figure 5.1 depicts this process by presenting the server-client interaction. The server-side makes sure that all the URLs are redirected to the same template *index.html*, which loads the main entry point, *app.js* that implements client-side routing. All the possible routes that the platform has are defined here. All the unrecognized routes are directed to the 'not found' page.

To provide the functionality of client side router the front end uses *Page.js* library. The motivation behind the choice of the library is in its simplicity, and the traditional approach for the routing. It functions by intercepting all the "*href*" attributes from the *html* "*a*" tags.

5.2.1 Memory leak prevention

Despite the advantages of implementing the platform as a single-page application, the approach comes with a set of challenges which force the front-end to take precautions. One such challenge is memory leaks. In order to avoid memory leaks, special care is

taken in modularizing the application, so that all the references to an object that no longer exists are removed, thereby allowing the garbage collector to delete the obsolete objects. For this reason, each instance class has a 'terminate' method, which is called once the class is no longer needed. This removes all the object references within the class. Once the class itself is removed, all possible memory leaks are prevented.

5.2.2 Architecture overview

An overview of the architecture of the front-end is provided below. The static and singleton classes that are used throughout the platform are left out from the schema to have clarity and keep the complexity down, depicting only the important aspects of the architecture.

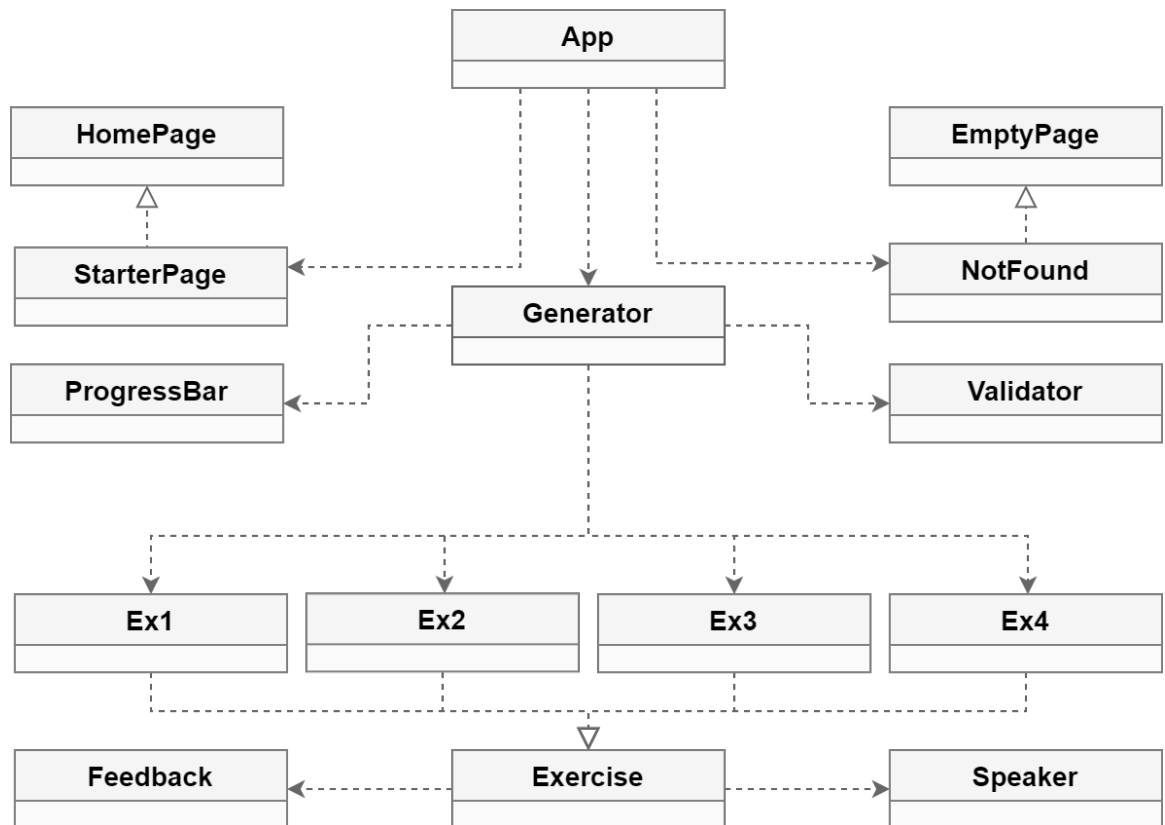


Figure 5.2: Platform architecture overview

The next sections focus on explaining the individual components of the system.

5.3 Exercise architecture

In order to create a maintainable exercise platform, the various components comprising the exercises had to be analyzed, to determine whether these components are shared or are particular to a specific exercise.

The decision was made to divide exercises into layers. Each layer is a visual representation for a set of functionalities. This section describes how the exercises are composed, visually. Later, the next sections dive into the technical aspect of the project by explaining the code composition.

On the first layer, the exercises are composed of the **generator layer**, followed by the **exercise layer**, and concluded by the **exercise type layer**². The composing layers are presented in Figure 5.3.

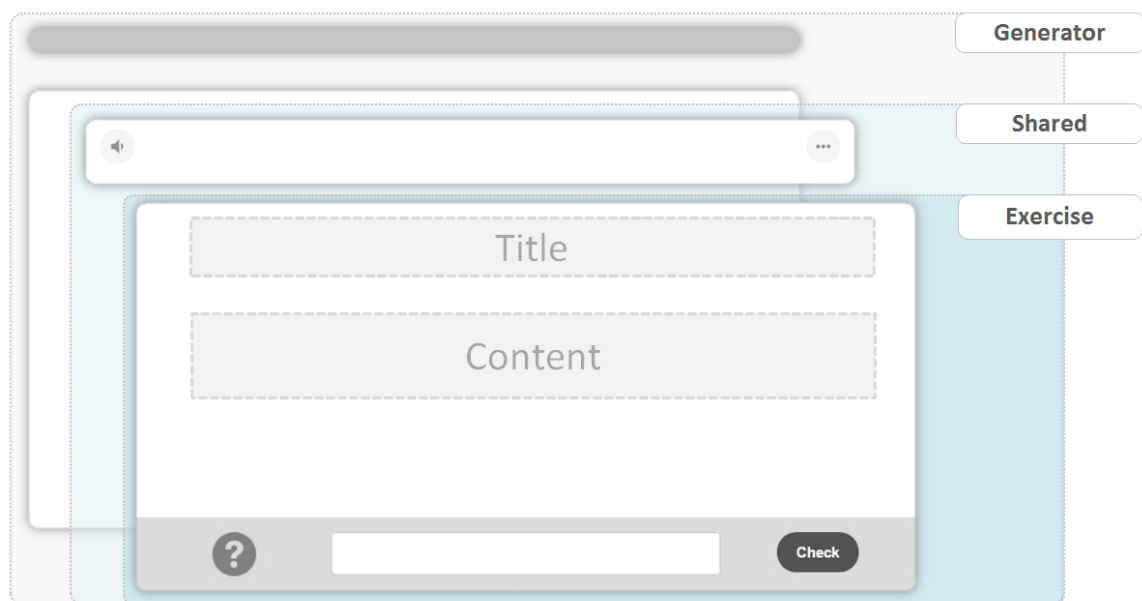


Figure 5.3: Composing layers of the exercises

The **generator layer** is comprised of the essential templates for the exercises. These are the progress bar and the exercise card, therefore, the generator acts as a binder for the two essential blocks.

²Also called as custom exercise layer.

On the second step, the exercise card is extended by the **shared layer**, which provides the generalized functionality for each exercise. These include the option panel and the sound bar. If the need arises, the layer can be extended to accommodate more generalized properties for the exercises.

The final step is the **exercise layer**. Each exercise has its own unique visual components, however, the basic structure is shared among all. Thus, each exercise type is composed of the header, the content, and the footer. In Figure 5.4, one can observe the differences and commonalities among the various types of the exercises.

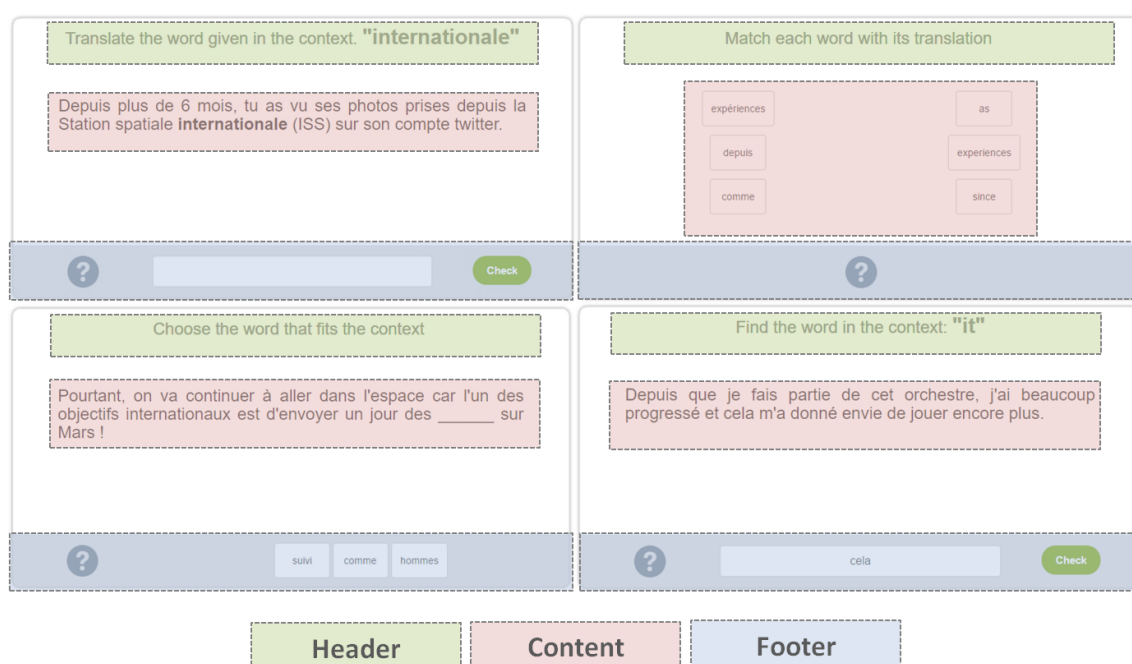


Figure 5.4: The commonalities of exercise types

In case the platform needs to be extended to accommodate new types of exercises, by keeping this structure, one can define the header, the content and the footer of each exercise by reusing the design blocks that are described in the next section.

5.4 Design blocks

It was discussed that each exercise is comprised of layers, and besides that, each layer is a combination of blocks. Examples are the sound block, the feedback block, the option block, etc. These blocks provide the necessary functionality for the exercises to give the user a meaningful experience. Some blocks are shared with all the exercises,

as it was presented while dissecting the exercises into layers, and some are custom only to a specific exercise. The next sections describe the functionality of the blocks that the exercises rely on, and explain their place in the architecture.

5.4.1 ProgressBar class

The progress bar class is present on the Generator layer. The class is defined as a JavaScript object literal. The class initialization requires two parameters: the starting point of the progress bar (how much the progress bar should be advanced from the starting point in percentages), and the size of the progress bar (or the amount of moves that is required to complete the progress bar). The class has public methods for initializing, restarting, moving, and terminating. The primary method that is used to advance the visual indicator of the progress is the 'move' method: when 'move' is called, the progress bar will move by a certain amount, based on the initialized size.

5.4.2 Speaker class

The objective of the speaker class is to convert text to speech in a provided language. The class uses SpeechSynthesisUtterance API that is provided by major browsers³. The class requires two parameters: the text that needs to be pronounced, and the language to use for the action.

As a building block, the sound class is used in the Exercise layer. Therefore, each instance of the exercises will have the module instance, and will override the text source and the language of the source to be pronounced. By default, the exercise prototype uses the bookmarks' source word for the sound module.

5.4.3 Feedback class

The feedback class is part of the Exercise layer. The class is designed to collect the user reactions related to a specific bookmark. The purpose of this functionality is to provide a feedback loop that will improve the personalization of the exercises for each user. In this project, the algorithm that deals with personalization and with providing bookmarks shall be treated as a black-box, as it is outside of the scope of the platform. The class uses the SweetAlert library, as well as the Mustache.js templating engine.

³Internet Explorer and Chromium browsers are not supported.

5.5 Publish subscribe module

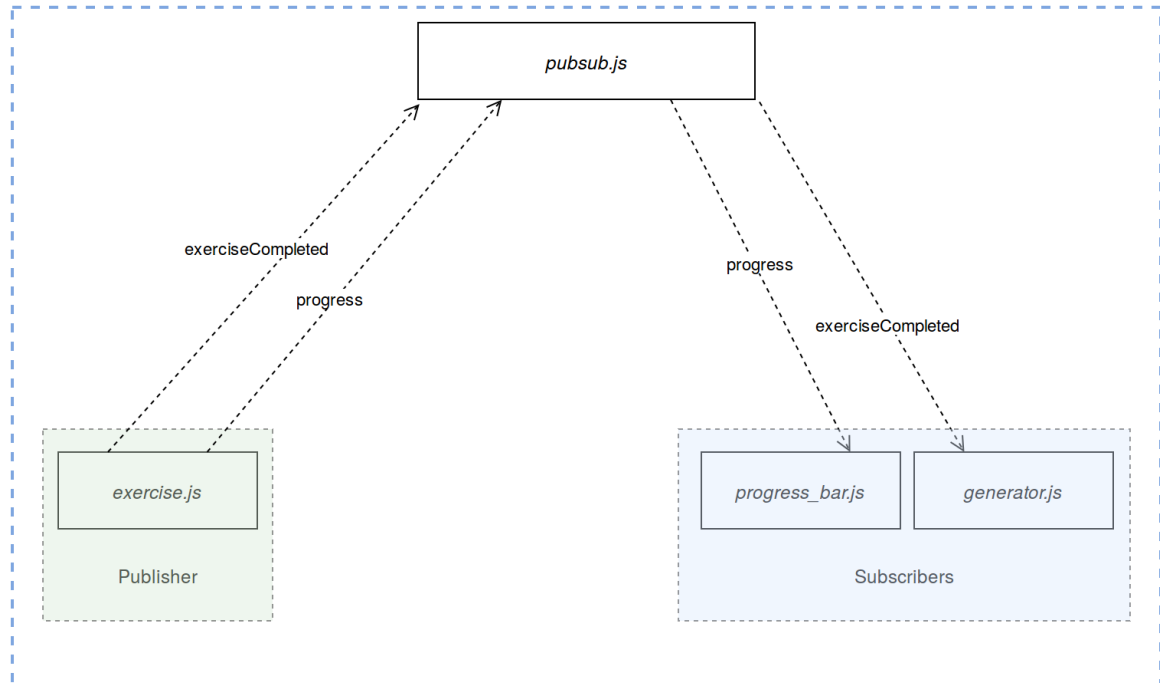


Figure 5.5: Publish subscribe module

The publish-subscribe (pubsub) module is an important part of the application. It is a singleton class that gives flexibility in controlling the communication of different modules without coupling them. The class follows the 'observer-observable' pattern.

In order to explain it's functionality, one should consider the Figure example presented above. The example presents two of the main contexts in which the pattern is used:

- The progress bar in the framework is independent from other modules, and to retain the independence but also allow communication between different modules, we employ the concepts of emitter and listener. In the example presented above, the exercise acts as the emitter, whereas the progress bar is the listener. Once an exercise is successfully completed, the emit signal is sent to the pubsub class from the exercise module. The pubsub module on its turn will run the function that is associated with the signal (in this case, the "move" method of the progress bar). Throughout the communication, the progress bar is unaware of its emitter, namely, the existence. This way, each running exercise sends a

'progress' signal whenever it was successful. The progress bar calls the move signal. The progress bar listens to the signal and updates when it receives.

- Another example for using publish-subscribe is seen in the generator module. When a set of exercises is completed, an 'exerciseCompleted' signal is sent to the publish-subscribe module. Upon receiving the signal, the generator moves on to the next exercise type, or terminates if there are no planned exercises. The generator receives the signal and creates the next set.

In addition, the pattern is very similar to the observer-observable pattern in other languages. The exercise has no knowledge of the existence of any subscribers, and the subscribers do not know the origin of the signal they receive. The fact that the subscribers and the publisher do not need to know about each other maintains the modularity of the application.

5.6 Generator module

The generator module is the first layer that comprises the exercises platform. It is created when the user selects an exercise set from the starter page. The module is responsible for generating exercises based on the received input. The input represents a Map that contains two parameters - the type of exercise and the number of times the exercise has to appear consecutively. An example of an input for generator can be seen below:

```
[[1, 2], [2, 2], [4, 1]]
```

From the example above, the input requires the generator to create an exercise set comprising of 3 exercises. The first exercise is exercise 1 (Find) that has to accrue 2 times, followed by exercise 2 (Choose) 2 times and finished by exercise 4 (Translate) 1 time. However the platform depends on external input therefore it cannot make sure to receive proper input at all times. Therefore the validator module requests for input from Zeeguu API then validates the input before generating. The approach lands an ideal ground for developing a personalized set of exercises by utilizing the 'word within a context' concept.

The generator class will first use the validator module to analyze the input. Based on the return value from the validator, the class will construct exercise sets.

5.7 Validator module

As it was briefly discussed, the validator module is responsible for handling the input from generator class. The main feature of the validator module is to check the input before supplying it to the generator module. It requests for bookmarks from Zeeguu API bookmarks-to-study endpoint. Based on the result, it recalculates the appropriate input for the generator. There are three possible ways for the validator to report, those are discussed in the following listing:

1. If number of bookmarks is zero, then the generator will terminate and the application will be directed to no bookmarks page.
2. If the number of bookmarks is less than the request total number of bookmarks for the generator, the validator will return a new input for the generator that can be accommodated by the received amount of bookmarks. The generator then will generate the exercise set according to the new input.
3. If the number of bookmarks is more than or equal to the requested number, the validator will return the same set, and the generator will generate the requested number of exercises.

5.8 Static classes

- **CookieHandler** class is responsible for retrieving and setting cookies provided the name of the cookie.
- **Session** class is dedicated to retrieving the user session from the client's browser cookies. It utilizes the CookieHandler class during this process.
- **Loader** module is an ES6 static implementation. It functions as a centralized place for template-loading within any class.
- **Settings** All the external API requesters are placed in the settings module. This separation allows for extending the platform for other API's in the future.
- **LoadingAnimation** is a singleton class, meaning that only one instance of the object will be available for the framework. The objective of the loading animation is to provide a visual interface for the user while the requested content is being loaded. The provided architecture that allows for the loading animation can be requested from any module. It hides the main content of the screen and

displays the loader, and once the loading is complete, the content is displayed once again.

Chapter 6

Results and Evaluation

In order to ensure that the minimal building blocks are correctly identified (**R1**), and that the users find the platform useful (**R2**), and for the further improvement of the platform, a study with the participation of over one hundred users was conducted for the duration of 24 days. The following sections discuss some observations that were made by evaluating user data, and show the conclusions and improvements that were made based on these observations.

6.1 Heat-map analyses

The significance of the starter page cannot be understated, since it is the first page the users get in contact with when accessing the platform. For this reason, it was analyzed with the help of heat-maps and user activity recordings to devise a better design for the page.

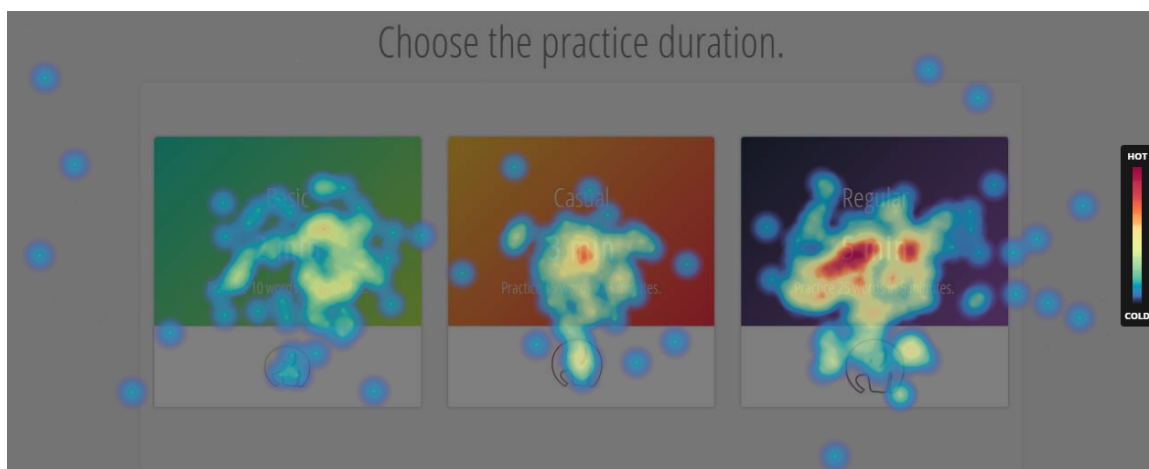


Figure 6.1: Starter page click heat-map.

The current implementation of the page holds 3 practice packages. From the packages, almost half of the clicks, 48%, were made on the 3rd package (5-minute practice). In comparison, 21% of the clicks fell on the 3-minute package, and 24% on the 2-minute one (Figure 6.2). From this observation, it can be deduced that our users are more likely to take longer practice sessions. This insight might be utilized to tailor the program more to the users' needs, in the future versions of the platform.

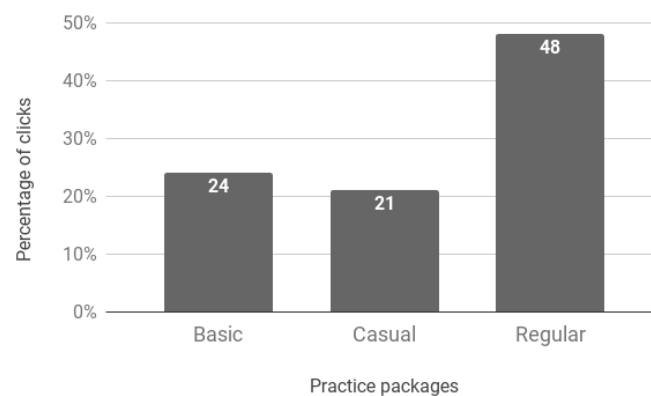


Figure 6.2: Usage statistics on starter page.

More importantly, the design presented above solves a more pressing issue that was present previously, when the platform had a different implementation for the starter page Figure 6.3.



Figure 6.3: Previous version of home screen

It had been observed before that the users would spend a significant amount of time on this screen, trying to decide which practice package to choose from 7 options. Therefore, the observation suggested that the users seem to be confronted with a phenomenon called *the paradox of choice* [8]. In addition, heat-map evaluation suggested that some practice options were barely clicked on. The currently adapted design remedied these shortcomings successfully.

6.2 General observations

The platform was used by 104 people, for a period of 24 days. During this time, a total of **14257 exercises were correctly solved**, in 4 different languages. Since the primary user groups came from two sources, from the Gomarus College (high school students learning French), and from the University of Groningen Language Center (participants learning Dutch), these two languages, Dutch and French, were the most practiced ones. (Figure 6.4). As these two groups were in majority, the following discussion shall be focused on them.

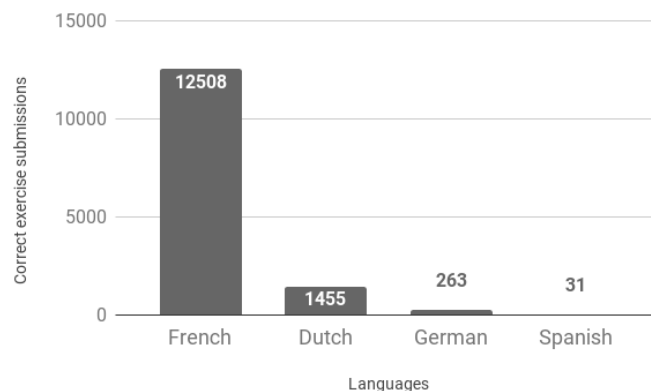


Figure 6.4: Number of correctly solved exercises per language.

During this time, on average, each person solved 137 exercises correctly. This gives an average of 7 words per day. Most of the activity came from the students of Gomarus College (the French-learning primary user group).

Language	Correctly solved exercises	Number of Users	Average per user
French	12,508	74	170
Dutch	1,455	26	56
German	263	3	88
Spanish	31	1	31
Total	14,257	104	137

Table 6.1: Interaction with exercises for different languages for the duration of 24 days.

6.3 User activity

This section focuses on answering the second research question, namely: "How do people use the platform?" The evaluation period showed that users would regularly return to the platform, demonstrated by the diagram below, which depicts user activity per day, throughout a period of 24 days.

User activity vs day

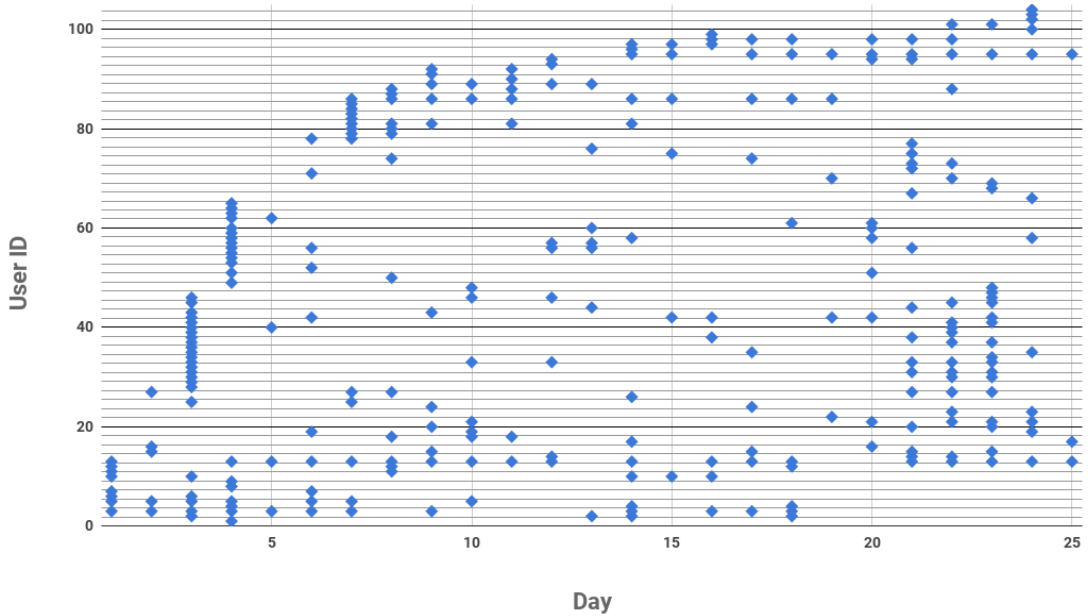


Figure 6.5: User activity throughout the validation period of 24 days.

The diagram also evidences that some users were more engaged with the platform than others. At the same time and more importantly, it also shows that, on average,

user interactions were spread out throughout the validation period. This evidences the fact that users continued to utilize the platform, as opposed to trying it out once but abandoning it. The analysis of user behavior has also shown that about 50% of the users have completed at least one hundred exercises during the validation period, which can be observed from the histogram presented below.

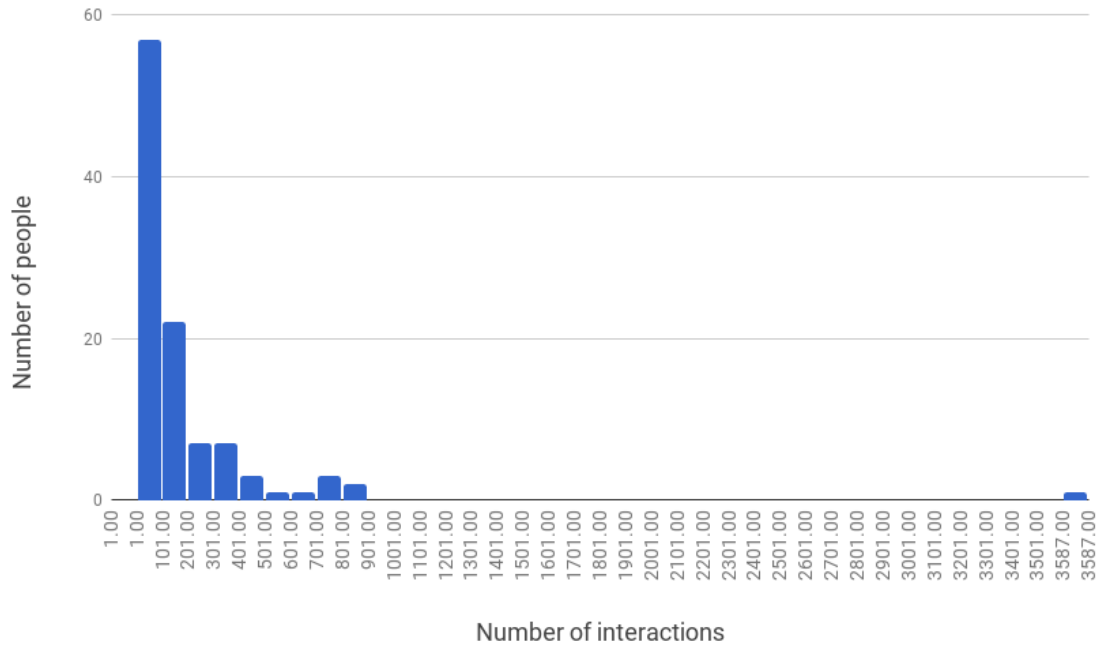


Figure 6.6: User activity histogram.

Additionally, students who completed more than one hundred exercises were more likely to return to the platform (at least once per week). Naturally, the presence of extremes has also been observed, such as a student who performed more than 3500 exercise interactions in a single day.

6.3.1 User interaction

The types of interactions tracked - **correct exercise submissions**, **wrong submissions**, **hint usage** and **feedback usage** - are presented in the figure below:

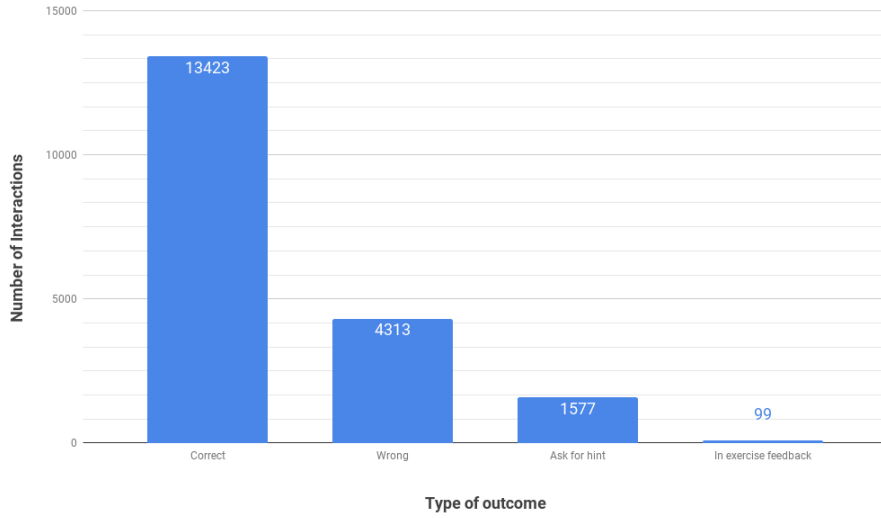


Figure 6.7: Number of interactions and type of outcome for French and Dutch learners.

The next sections will describe these four interactions in more detail.

6.3.2 Hint usage

Observing the different types of exercises demonstrated that not all the interactions were used evenly among them. The table below presents the four types of exercises and the corresponding number of hints used for each of them:

	Find	Choose	Translate	Match
Total interactions	6,885	7,563	2,862	2,099
Ask for hint	565	42	949	21
Percentage	8.21%	0.56%	33.16%	1.00%

Table 6.2: Hint usage among different exercises.

For the exercise types *Choose* and *Match*, the learners used the least amount of hints - 0.56% and 1%, accordingly. On the other hand, for the *Translate* exercise, hint usage was 33.16%. The hint usage for the *Find* exercise was in between these two extremes, but was still significantly larger than the first two - 8.21%. From this observation, one can conclude that the reason for the diverging hint usage was the different strategy behind the hints for the different types of exercises. The hint strategy for the latter group (*Translate*, *Find*) gives away the whole answer, therefore, users are more motivated to 'skip' the answer by using hint. On the other hand, the hint strategy

for the exercises *Choose* and *Match* eliminates one of the wrong options, therefore, users do not find the need of using the hint pressing enough and prefer to test the options themselves. An alternative hypothesis would be that the hint usage is related to the difficulty of the exercises themselves, therefore the *translate* exercise can be considered the hardest exercise type.

6.3.3 In-exercise feedback

In total, for the duration of usage, in-exercise user feedback was received on 99 occasions. The distribution of user feedback is presented below.

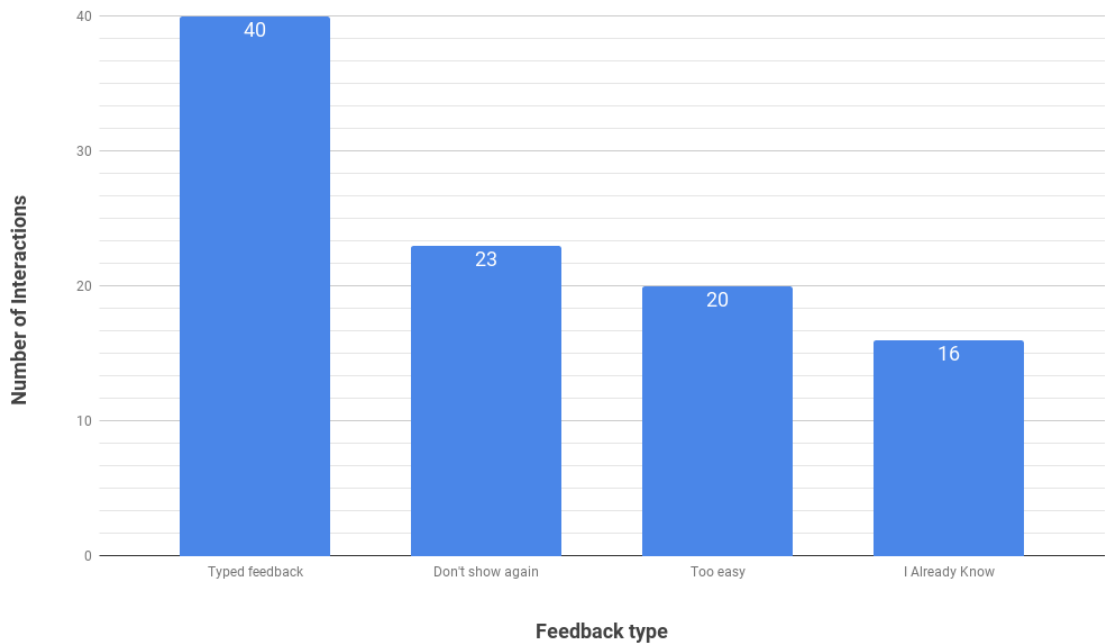


Figure 6.8: In-exercise feedback distribution.

As it was discussed before, there were two main methods for the user to submit feedback within an exercise: the predefined option and the manual (typed) option. 40% of user feedback was done manually, while 60% was done by using the predefined options. From this, an assumption can be made that users prefer to use the predefined options. Nevertheless, in the future, new predefined types of feedback can be distilled from the analysis of the manual feedback, in order to understand user preferences and to see if the users find it useful to continue sending feedback. The users might perceive this step as redundant, due to the lack of an immediate effect after submitting the

feedback. Nevertheless, while analyzing the feedback, it became clear that users who engaged in providing feedback used this feature multiple times.

6.4 Learning experience

In order to answer the second part of the research question (that is, how people actually learn using the platform), this section will consider the correct and wrong usage of words learned. The column chart presented below shows the correct and wrong interactions with the exercises for each user. On average, among all user groups, the percentage¹ of correct answers was 72%. On average, it was observed that the ratio of correct usage among Dutch learners was slightly lower than that of French learners - 71.5% as opposed to 73.2%. This difference could be attributed to a variety of factors (the difficulty of the language, the level of proficiency of the learner, etc.).

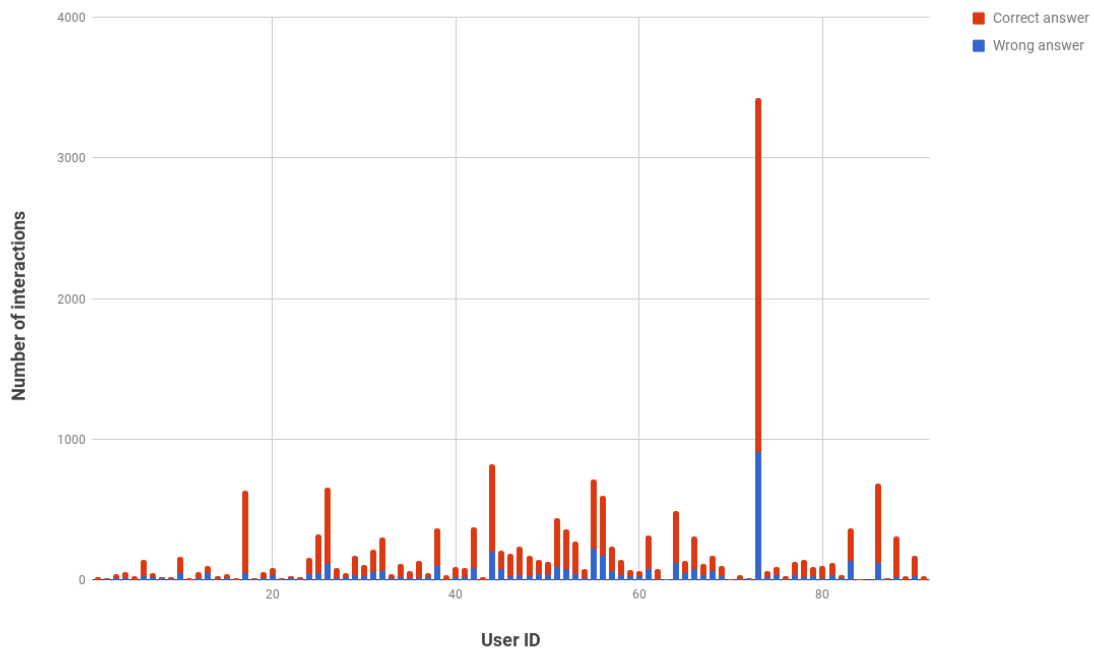


Figure 6.9: Wrong vs. correct exercise outcomes for each user.

Another factor that can come into play when interacting with the platform is the different types of the exercises that the users are presented with. The table below

¹The percentage of correctly answered exercises from the sum of correct and wrong submissions.

contains the usage data for three types of exercises.²

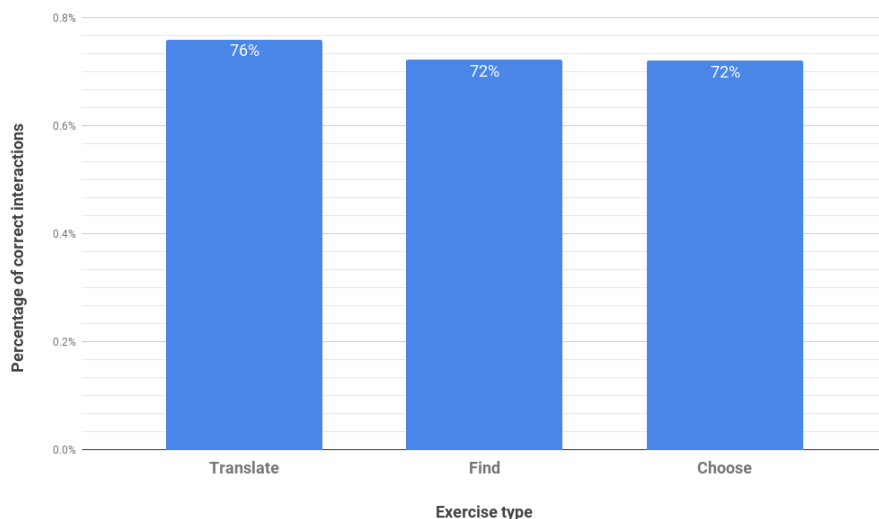


Figure 6.10: The exercise type and correct vs wrong ratio.

Among these three, we can observe that the *Translate* exercise had a slightly higher average percentage of correct solutions (76%) than the other two exercises, *Find* and *Choose* (72%). Alternatively considering that the exercise type *Translate* had the highest number of hint usage we can conclude that when users did not feel confident about the exercise they 'skipped' it by using hint option. This resulted in higher percentage for correct solutions.

6.4.1 Learning process

In total 5149 bookmarks were used in the exercises platform during the learning period. The bookmarks that were correctly solved for the final iterations were identified as learned. This group comprises of 4110 bookmarks. According to this derivation, the users learned in total 80% of the bookmarks that they have interacted with using the platform.

It is important to note that the learning process followed certain patterns that repeated among different users. For instance, for 725 bookmarks, the users answered wrong during their first interaction with the bookmark, after which the final solutions were correct, showing a natural learning process. For 3385 bookmarks the users

²The exercise 'Match' has a different approach for submitting the wrong outcome, therefore, it is not discussed in this section.

answered correctly from the beginning and the answers still remained correct for the final iterations. In total 1037 (20%) of the bookmarks still remained not learned, since the final outcomes showed wrong interaction with the bookmark. Besides these general patterns, there were common learning patterns observed that are presented in the figure below.

Learning pattern	Frequency
C	1,212
CC	646
CCC	393
WC	190
W	188
CCCC	158
CCCCC	94
WWC	90
CWC	83
WCC	81
WW	80
CCCCCC	70
CW	66
CCWC	53
CWCC	50

Table 6.3: Frequency of pattern occurrence.

The character *C* stands for correct interaction and character *W* stands for wrong interaction. The left most character of the pattern is the first interaction whereas the right most character is the last interaction with the bookmark. From this observation two conclusions can be drawn. Firstly, many of the bookmarks were solved correctly from the beginning and remained correct for the rest of the learning process. This observation implies that bookmarks were already part of the user’s vocabulary. Secondly, the bookmarks that were not correctly solved in the beginning, or that had wrong solutions in the middle, eventually yielded correct interaction for final iterations. The patterns presented in the figure above apply to more than 67% of the bookmarks.

From the above discussion it is safe to conclude that the platform was a useful tool for the users, both in strengthening their vocabulary and in helping them learn new words.

6.5 Explicit Perceived Usefulness

During the evaluation period, explicit user feedback was also collected. There were two sources of the feedback - Hotjar polls and the final survey that was sent to all the users that participated in the validation period. An example of a user feedback collected via Hotjar polls is presented below.



<u>OS</u>	<u>DATE</u>	<u>How did you find our site?</u>
	6 days ago	It is very useful. It helps to improve my French vocabulary.
	6 days ago	Really good, it's easy to understand and made beautifully.

Figure 6.11: Explicit user feedback.

The final user survey revealed explicit user satisfaction. Some of the positive remarks are presented below:

Question: What did you **like** the most about the exercises?

- "You learn a lot."
- "Writing the word."
- "The variety of the type of questions."
- "It's a fun way to study words, it looks like a game. You don't consider it like a process of studying."

Besides the positive remarks, it is important to identify the aspects of the platform that do not fit users' preferences. To this end, the final survey included a question on the most disliked aspects of the platform.

Question: What did you **dislike** the most about the exercises?

- "The translations were not so good."
- "The translations are in English."
- "Nothing."
- "There aren't translations."
- "Doesn't give the translations."
- "Nothing really."
- "The exercises would only focus on the first 5-6 articles and then repeat the questions on the same articles."
- "You practice your whole vocabulary instead of just the vocabulary from the text you read."

The users' answers point to some shortcomings that are both part of the platform as well as part of the API that the platform relies on. Therefore, in order to resolve these problems, a future cooperation between the two platforms is necessary. On average, the platform was rated 3.75 out of 5, presented in the figure below.

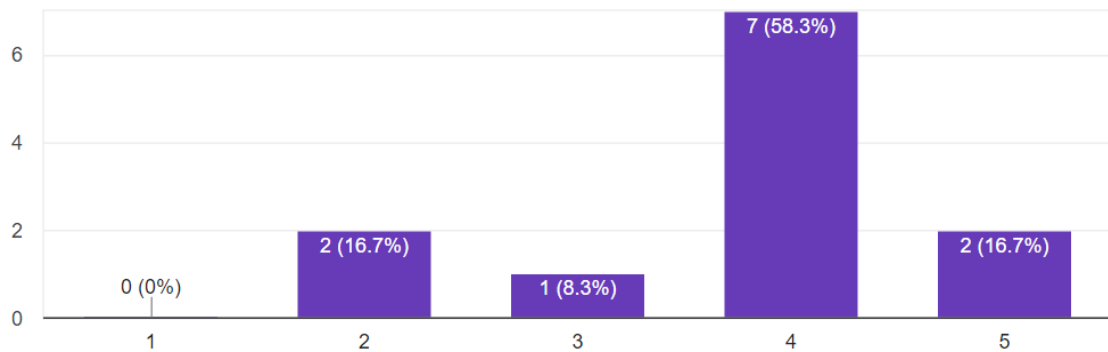


Figure 6.12: Rating of the platform by 12 users.

Chapter 7

Conclusion

The user interaction with the platform shows that an intuitive design was achieved. Iterations over the design components helped to accomplish more fluent experience for the user. The heat-maps and recordings suggest practically no learning curve involved while interacting with the system. This is thanks to the human centered design approach that was adapted by the project early on during the development. The platform also proved to be capable of providing user feedback in a way that is easy to parse manually and draw conclusions. This helped to quickly overcome the shortcomings that the framework faced.

Taking into consideration the massive usage in a short period of time, it can be concluded that some users found the platform engaging and useful. During this testing period with the users, feedback is received, suggesting enhancements for the platform. These enhancements will be addressed in the future work. A questionnaire asking for explicit user feedback suggested user satisfaction.

Taking into consideration the above mentioned outcomes, and referencing the requirements that were set for the platform, we can conclude that all functional and non-functional requirements are successfully met. The framework proved to be maintainable and flexible when extending, and applying enhancements, thus we estimate a strong basis for the future development.

7.1 Future work

The project has a set of enhancements in the backlog. Types of those enhancements are discussed in the next subsections.

Usability Improvements

Usability improvements include enhancements for the current exercise types, general improvements for usability of the platform and new types of exercises. By using Hotjar polls, a large amount of user feedback is collected that can also become potential features for future iterations. Those enhancements are primarily for improving the usability of the platform or enhancing the platform with more features.

Personalizing exercises

Given that the project has proven to be ready for testing with actual users, a new developments can be conducted in the future in order to answer possible questions. One of such question is - *What is the optimal strategy for scheduling various types of exercises ?*. This can be achieved by personalizing the starter page based individual user activity. This activity can be analyzed using the feedback that users provide from the exercises. Additionally the user can be presented with reaction possibility (such as a like and dislike buttons) for each exercise. This will give us more insight about user preferences. This bring another questions - *How to ensure that the platform adapts to the individuals preferences?*. In order to accommodate more insights, the system can use more thorough logging of user activity, such as logging clicks on every interactive UI element.

Appendix A

Setup Overview

In order to achieve modularity in the application, the front-end is build using EcmaScript 6. This is the EcmaScript's native way of defining modules and reusing them in different parts of the project through import and export statements. However, since EcmaScript 6 (ES6) is not the standard of the web, Babel transpiler is used to compile the code to JavaScript¹, also knows as Vanilla JavaScript, that is the adopted standard of the web.

In order to bundle the above mentioned ES6 modules together we use Webpack module bundler. We create a single entry point for the framework, by using Babel transpiler. This all is coupled together using the default package manger for Node.js, the NPM. The resulting combination of technologies and methods gives powerful ground for developing web frameworks.

¹The best-known implementation of ECMAScript

Bibliography

- [1] Akresh-Gonzales. What Is the Testing Effect, and How Does It Affect Learning, Knowledge, and Retention? 2015.
- [2] D. Chirtoaca. Apollo: Simplicity and intuitiveness in an article reading tool. 2017.
- [3] E. Bruce Goldstein. Cengage Learning. In *Cognitive Psychology: Connecting Mind, Research, and Everyday Experience.*, page 231, 2004.
- [4] LUMA Institute. Innovating for people: Handbook of human-centered design methods. 2012.
- [5] S. Jager. Towards ict-integrated language learning: Developing an implementation framework in terms of pedagogy, technology and environment. 2009.
- [6] Mircea F. Lungu. Bootstrapping an ubiquitous monitoring ecosystem for accelerating vocabulary acquisition. In *Proccedings of the 10th European Conference on Software Architecture Workshops*, ECSAW '16, pages 28:1–28:4, New York, NY, USA, 2016. ACM.
- [7] P. Nation. Learning vocabulary in another language. 2001.
- [8] Barry Schwartz. The paradox of choice, 2004.
- [9] L.A.H. van den Brand. Prometheus: efficiency and usability in an article related feed manager. 2017.