A Gamified Japanese Language Learning App

Student Name: Shaurya Shekhar Dey (s.dey2@ncl.ac.uk)

Student Number: 220025500

Supervisor: Dr Konrad Dabrowski (konrad.dabrowski@ncl.ac.uk)

Timescale: June 2023 – August 2023

Deadline: August 28 2023

Word count: 16856

Abstract

In today's interconnected world, learning a new language has become a valuable skill. The pandemic saw a rise in the number of online language learners using language learning applications. This project aims to address the challenges faced by language learners by developing a gamified Japanese language learning application that combines effectively structured learning paths with engaging gamified elements.

The application leverages the power of gamification to create an immersive and interactive learning experience. Drawing from contemporary research on language acquisition and gamification principles, the app transforms the traditional language learning approach into an enjoyable and effective journey for users of all ages and skill levels.

Acknowledgements

I would like to thank my project supervisor Dr Konrad Dabrowski for his help during this project. His guidance has been invaluable.

Table of Contents

Table of Figures

Table of Tables

1.0 Introduction

There was a huge surge in interest in language learning during the lockdown period. Numbers show that due to the social distancing phenomenon, language learning apps have claimed increased usage of more than 200 per cent while many report sales up more than 50 per cent.

Data suggest that there has been a year-on-year increase in annual revenue generated by the language learning app sector between 2016 and 2021 [1]. Another report projects that the language learning market will be worth 31.81 billion dollars by 2029 [2]. Duolingo, the most popular app in the sector, surpassed Babbel, a competitor, to generate 250 million dollars in revenue, eclipsing Memrise, Tandem and others in the process.

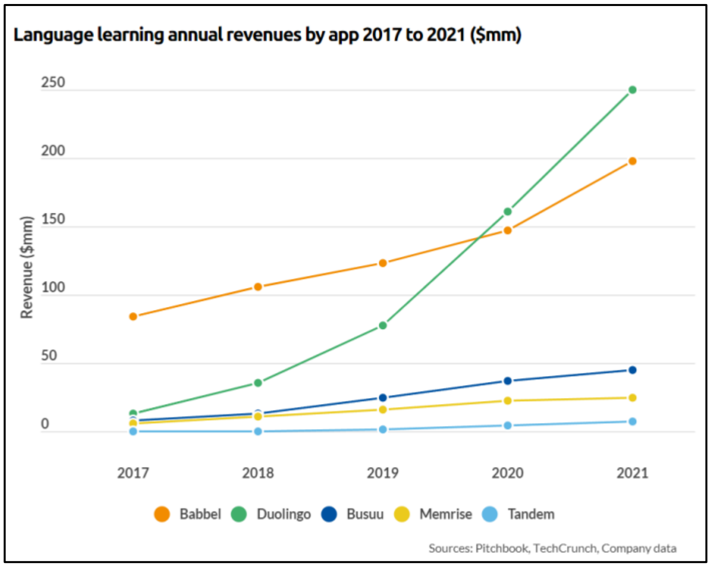


Figure *1* - Language Learning App Revenues[1] *(Pitchbook, TechCrunch)*

While these apps offer a good starting point, they fall short of providing enough content for intermediate and advanced learners. They might offer a structured learning path, but the content is often limited. This has prompted many users to use flashcard-based learning apps like Anki or sentence-based learning apps like Clozemaster. Anki uses a highly effective spaced-repetition system in addition to far greater customizability. This makes Anki a much more desirable learning tool. Its strongest suit is its customizability, allowing users to create and add their flashcards and decks. Research has shown that SRS (Spaced Repetition System), if used correctly, shows greater gains in language acquisition compared to other methods. At the very least, learners report a learning experience that is positive enough to motivate them to learn the language further. [3]

This project will design and create an app that builds on the algorithmic groundwork laid by the spacedrepetition system and add gamification to the experience so that language learners find it motivating to keep coming back to it. This project will also learn from what has made apps like Duolingo

[1] "Language Learning App Revenue and Usage Statistics (2023)," n.d., https://www.businessofapps.com/data/language-learning-app-market/

successful and add structured learning paths along with an interactive and gamified user experience. The application will deliver an experience that will attract new learners but will also cater to advanced learners, allowing users to create and add their flashcards.

This project will focus on one language, Japanese. This choice is intentional as the content is less important for the project relative to the features it has to offer. Nonetheless, it will detail clear learning paths that the users can start with and enough built-in content to cater to intermediate learners. The Japanese Language Proficiency Test, henceforth referred to as the JLPT, has been decided upon as the foundation that the core content will be based on.

Despite the benefits that SRS software provides, there is still a gap between the ease of setup between apps like Duolingo and Anki. While the former requires no setup but fewer features, the latter provides a much wider breadth of content but requires reading tutorials, knowledge of SRS and setting up their decks. Research has also shown that people were often bored with the UI that Anki had to offer, which could potentially inhibit the process of learning. Thus, the lack of a more personalized gamified experience is recognized as one of the key issues.

1.1 Aim

This project aims to create a feature-rich language learning web application that employs a Spaced Repetition System (SRS) to optimize the learning process for second-language acquisition. The application will focus on content for the Japanese language. The application will consist of several ‘learning tracks’ that will structure the content based on frequency, grades, and current proficiency level. The application will also make the experience entertaining by implementing leaderboards, achievements, and streaks.

1.2 Objectives

The objectives of this project are as follows:

1. Develop an optimized spaced repetition system for second-language acquisition using the SuperMemo2 algorithm.
2. Design an interactive and user-friendly interface to increase user engagement.
3. Conduct a usability evaluation to ensure the application aligns with fundamental usability principles.
4. Incorporate a leaderboard system allowing sorting based on streaks, days studied, and mastered Kanji.
5. Integrate personalized learning tracks as predefined decks of flashcards, tailored to learner proficiency.
6. Administer a brief test to determine if the learner is a complete beginner.
7. Populate each flashcard deck with audio corresponding to the displayed text.
8. Enable the user to input their answers for specific sections of the application.
9. Ensure application responsiveness across all screen sizes and adherence to accessibility standards.
10. Provide support for major browsers, excluding Internet Explorer.

1.3 Motivation

As demonstrated by background research and literature review, there are many applications out there that have revolutionized language learning for the digital age. This application seeks to piece together everything and fill in the gaps left by existing language learning apps. It should not be a gamified experience OR a sophisticated algorithm. This project will be about curating each person's language learning journey with interactive exercises, progress tracking, optimizing reviewing-timing and maximizing retention.

1.4 Project Scope

The scope of this project will be bound by the following constraints and assumptions:

- This application will be optimized for use on the web. There are no plans or expectations to create it on multiple platforms.
- This application is designed first and foremost with a heavy focus on its features and functionality that deals with gamification. Although the UI will be clean and minimal, there are no expectations to curate an application with commercial-grade UI.
- The application will focus on exploiting existing spaced-repetition algorithms and will not try to optimize an existing one or derive a new one.

1.6 Outline of Material

The thesis consists of 5 chapters that are organized as follows.

1.0 Introduction – This introduces the project topic and objectives.

2.0 Background Research – This explains the nuances of the Japanese Language, an introduction to the spaced repetition concept, gamification, and its use in language learning. It also discusses the features of existing language-learning apps.

3.0 Building the App – This section details the agile development process that went behind making the app. It starts with the technologies used in the app and their relevance to the app. Following that are the functional and non-functional requirements and the app design and prototype. The Implementation section is the most detailed as it talks about the high-level implementation of the app along with low-level details pertinent to the app's development. Every important component of the application is talked about here, including the Flashcards, Scheduler and Gamified elements. Finally, the section concludes with details of testing and the success of the implementation.

4.0 Results and Evaluation – This section wraps up the implementation by reflecting and evaluating the outcome. It assesses the app's validity as an actual tool for language learning as well as talks about the current limitations.

5.0 Conclusion – The final chapter is the epilogue that talks about the possibility of an extended implementation if the time constraints weren't there.

2.0 Background Research

This section contains the relevant background research that was conducted to aid with application design and development. Herein lies the literature review and an evaluation of existing applications used. This also includes the significance of spaced-repetition models for language learning and how it helps people remember things more effectively as opposed to cramming the material.

Different platforms have been researched to select the most appropriate one that can be easily accessed by all language learners as well as comply with the functional requirements defined by the project.

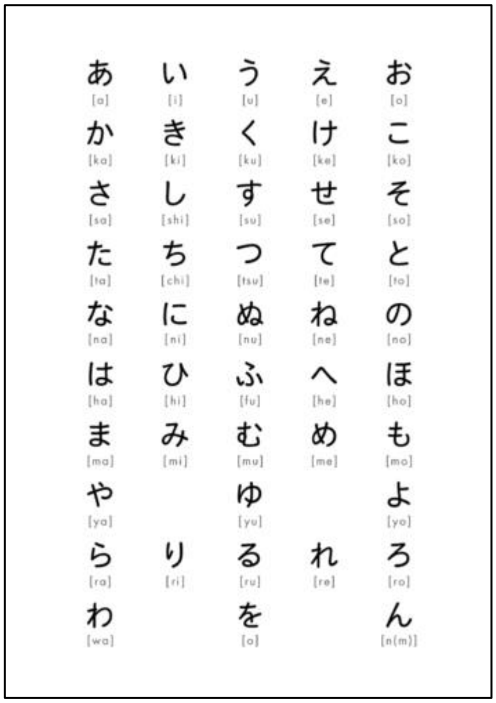
Furthermore, it will also give a reason for the decision to choose Japanese as the initially supported language and the need to structure content around the Japanese Language Proficiency Test rather than a more generalized approach. It will also show why there is a need for a bespoke application that combines the desirable qualities of the applications that inspire it.

2.1 A Brief Introduction to the Japanese Language

The Japanese Language is a captivating and intricate language that interests learners with its unique blend of scripts, grammar structure, and linguistic nuances. One of its distinguishing features is the employment of three scripts: Kanji, Hiragana and Katakana.

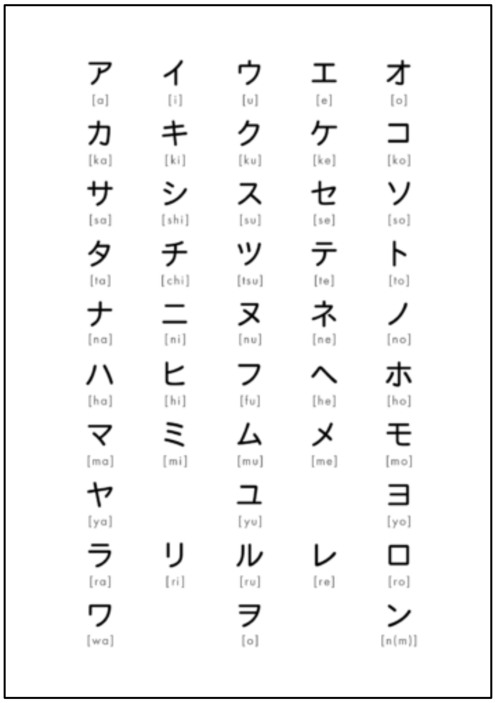
Kanji, originating from Chinese characters, form the backbone of Japanese writing. These intricate characters encompass a wide array of meanings and are essential for understanding more complex texts, ranging from literature to official documents.

Complementing Kanji are Hiragana and Katakana, syllabic scripts designed for native and foreign words, respectively. Hiragana graces the pages with its fluid curves, representing the basic phonetic sounds of the language. Katakana, with its angular strokes, is the gateway to foreign words and onomatopoeia, adding vibrancy to Japanese expression.



あ [a]	い [i]	う [u]	え [e]	お [o]
か [ka]	き [ki]	く [ku]	け [ke]	こ [ko]
さ [sa]	し [shi]	す [su]	せ [se]	そ [so]
た [ta]	ち [chi]	つ [tsu]	て [te]	と [to]
な [na]	に [ni]	ぬ [nu]	ね [ne]	の [no]
は [ha]	ひ [hi]	ふ [fu]	へ [he]	ほ [ho]
ま [ma]	み [mi]	む [mu]	め [me]	も [mo]
や [ya]		ゆ [yu]		よ [yo]
ら [ra]	り [ri]	る [ru]	れ [re]	ろ [ro]
わ [wa]		を [o]		ん [n(m)]

Figure 2 - Hiragana Chart [2]



ア [a]	イ [i]	ウ [u]	エ [e]	オ [o]
カ [ka]	キ [ki]	ク [ku]	ケ [ke]	コ [ko]
サ [sa]	シ [shi]	ス [su]	セ [se]	ソ [so]
タ [ta]	チ [chi]	ツ [tsu]	テ [te]	ト [to]
ナ [na]	ニ [ni]	ヌ [nu]	ネ [ne]	ノ [no]
ハ [ha]	ヒ [hi]	フ [fu]	ヘ [he]	ホ [ho]
マ [ma]	ミ [mi]	ム [mu]	メ [me]	モ [mo]
ヤ [ya]		ユ [yu]		ヨ [yo]
ラ [ra]	リ [ri]	ル [ru]	レ [re]	ロ [ro]
ワ [wa]		ヲ [o]		ン [n(m)]

Figure 3 - Katakana Chart [3]

[2] https://nihongoflashcards.com/hiragana-katakana-chart/

[3] https://nihongoflashcards.com/hiragana-katakana-chart/

Japanese grammar structure, while initially perplexing to newcomers, underpins the language's elegance. The subject-object-verb (SOV) order contrasts with many Indo-European languages, granting a distinctive rhythm to sentences. Particles, small grammatical units, provide context and nuances to phrases, ensuring precise communication even within succinct expressions. [4]

In the realm of language proficiency assessment, the Japanese Language Proficiency Test (JLPT) stands as a pivotal benchmark. Organized into five levels, from N5 (beginner) to N1 (advanced), the JLPT evaluates listening, reading, and grammar comprehension skills. This standardized assessment aids learners in gauging their linguistic progress and serves as a credential for academic, professional, and personal pursuits involving the Japanese language.

Clearing JLPT is one of the essential requirements for many exams and accreditations in Japan and for anyone willing to move there, it proves a level of competency for the language required to go around the country. [5]

Why Japanese for this project?

As highlighted above, the Japanese language is very intricate and offers a lot in the way of developing a language learning app, it offers the most flexibility in creating and showcasing multiple learning paths that cover the breadth of the language. Also, being unfamiliar to most learners, in future testing of the app, it presents novel content that can help in ascertaining the efficacy of the spaced repetition system; like how Ebbinghaus attempted to test his ability to recall the list of ‘nonsense syllables’ which has been described in detail in the next sub-section. Moreover, the author has previously studied Japanese and made use of spaced repetition to aid their learning, making it the ideal language to build the application with.

2.2 Enhancing Language Learning Via Spaced Repetition and Mnemonics

Spaced repetition is a powerful technique that can be leveraged to enhance language learning outcomes. “It is a memory technique that involves recalling information at optimal spacing intervals until the information is learned at a sufficient level” [6] Through strategically reviewing language material at specific intervals we can optimize memory retention and promote effective learning. It can help reduce time spent learning while retaining information better.

2.2.1 The Ebbinghaus forgetting curve

Hermann Ebbinghaus in his attempt to test his ability to remember a list of nonsense syllables discovered a few key aspects of memory like, memories tend to weaken over time and there is a massive drop in retention soon after learning something.

Ebbinghaus’ Nonsense Syllables

Ebbinghaus' nonsense syllables were a series of three-letter combinations that were designed to be devoid of any existing meaning or associations. These syllables were used by German psychologist Hermann Ebbinghaus in his pioneering research on human memory and learning in the late 19th century.

Ebbinghaus used 2300 *nonsense syllables* [7] to study the process of memorization and forgetting. By using syllables that had no inherent meaning, he aimed to eliminate any prior knowledge or associations that could affect the results of his experiments. This allowed him to investigate the fundamental principles of how information is acquired, retained, and forgotten over time.

Ebbinghaus' experiments involved memorizing lists of these nonsense syllables and then testing his memory at different intervals to see how much of the material he could recall. He discovered patterns of forgetting and retention, which he famously represented with the "forgetting curve." This curve showed that memory rapidly deteriorates shortly after learning but then levels off over time, the first few repetitions being the most effective.

Ebbinghaus' work laid the foundation for the scientific study of memory and learning, and his research methods and findings have had a lasting impact on the field of psychology.

If what is being learned is completely novel to the learner, they are more likely to forget it if they don’t understand it. Finally, how the information is presented also has an impact on recall ability. In conclusion, he learned that the difference in performance can be explained by mnemonic representation skills.

Ebbinghaus found that the following methods are the most optimal for increasing the strength of memory [8]:

- Better memory representation (with mnemonic techniques)
- Repetition-based active recall (like spaced repetition)

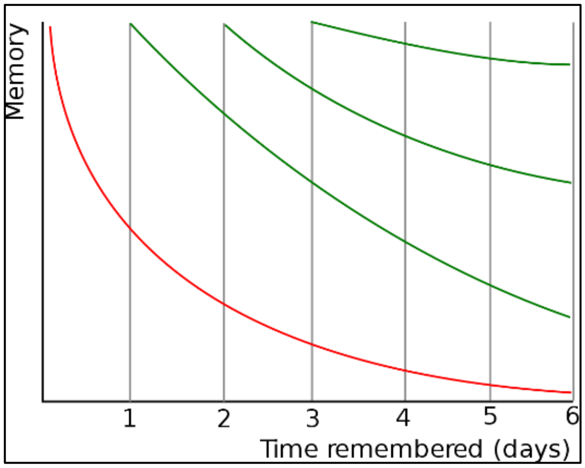


Figure 4 – A Typical Representation of The Forgetting Curve

(Wikipedia, The original uploader was Icez at English Wikipedia., Public domain, via Wikimedia Commons, 2007) [4]

The methods for increasing the strength of memory will be focused on. Mnemonic techniques will be reinforced using flash cards. Repetition based on active recall will be implemented using a spaced repetition algorithm.

Later research also pointed out that besides the two factors proposed by Ebbinghaus, higher “original learning” would also produce a slower rate of forgetting. [9]

Spending some time each day to recall and remember information will lead to a drastic decrease in the effects of the forgetting curve. Learning consultants also claim reviewing materials in the first 24 hours of learning is the optimum time to actively recall the content. This also resets the forgetting curve. There is also ample evidence that shows that waiting 10 to 20 per cent of the time towards when said information will be needed is most likely to be the best time to review it. [10]

2.3 Using Gamification to Maximize User-retention and Promote Learning

The literature review reveals that there have been many successful attempts at gamifying certain activities to capture attention even outside education. The winner of the Biggest Energy Saver Contest by San Diego Gas and Electric achieved as much as 46.5 per cent energy savings, equal to 1,356-kilowatt hours for her family of three, and those who used the same energy-saving gaming app achieved 20 per cent savings on average, compared to 9 per cent by those who used only the device without the app [11] It promotes a win-win strategy that can result in a positive social impact along with self-improvement.

In a forum-based setting, Stack Overflow is a popular example of how gaming elements such as points, badges and privileges have prompted users to ask and answer questions which ultimately benefits the community using it. [12]

Adobe created *LevelUp*, which gamified the process of learning the image-editing software Photoshop. It assigned missions and rewarded users with badges and awards. [13]

A study by the University of Florida found that students who played math computer games reported an increase in their math performance on core number sense tasks, along with a marked improvement in their confidence in their mathematical abilities. [14]

Nanyang Technological University in Singapore also examined the effectiveness of today’s gamified second language learning apps and explored the benefits of efficient learning and the strong initial motivation for these learning apps.[15]

[4] https://en.wikipedia.org/wiki/Forgetting_curve

There is a myriad of research findings that point towards gamification having positive psychological benefits [16] positive attitude towards complex subjects and lead to overall cognitive development. [17] [18]

2.4 Platform for Application

Background research reviews that multiple suitable platforms can be considered as one to build the application on. However, it has been found that a web-based solution is probably the most suitable. The only other alternative was a native mobile application. However, because of the following reasons, a web application stands out as the better choice:

- **Platform independence**: One of the biggest strengths of web-based applications is that they can be accessed from any device with a web browser. Regardless of which operating system, web apps can be made accessible to the widest possible range of users without needing to develop separate versions for different platforms. Given the limitation of manpower in this project, it only makes sense to go with a web app since it makes it easier to test as well as streamline development cycles. It is also understood that the app will be made responsive so that it conforms to variable resolution (widescreen to mobile)
- **Ease of Updates**: It is essential to note that while this application will focus on Japanese for the time being, it shall be developed in a way that other languages can be seamlessly accommodated and fit into the gamified app. This would mean that simple updates deployed to the server allow users to access all the content.
- **Faster Development**: Given how popular JavaScript and its libraries are, there are a lot of resources online that can help speed up the development process.
- **Discoverability**: Through Search Engine Optimization (SEO), web apps can easily attract search results and organic traffic.
- **Responsiveness and Accessibility**: Accessibility and responsiveness do not need to take a backseat. Web apps are easy to use and to be made user-friendly.

It is recognized that it would mean a lack of offline progress and consequently require the user to be online while learning but research on existing applications shows that all existing applications have a web-based version and their native or desktop versions *still heavily rely on using the internet* to sync progress across devices.

2.5 Existing applications

Building on the foundation of spaced repetition learning and what makes it effective, the functionality of the application to find what works the best was also researched.

Ariza and Hancock [19] found that it was only through a process of personally accommodating information into previously existing knowledge structures, that it was possible to build knowledge and it was vital to present appropriate content and promote interaction between the learner and the content. A critical design feature is to show as much comprehensible input as possible. To achieve this, one can go about two different models, an integrated model with illustrations, explications, and a variety of genres to drive motivation and accessibility, and a concurrent model that focuses on vocabulary, text, rhetorical structure, and support for the learner.

2.5.1 Duolingo

Duolingo is ahead of every other language-learning website by a substantial margin. As can be seen in *Figure 3* more than 60 per cent of installs are for Duolingo alone. The numbers for app usage and cumulative downloads are similar, Duolingo is far ahead of everyone else.

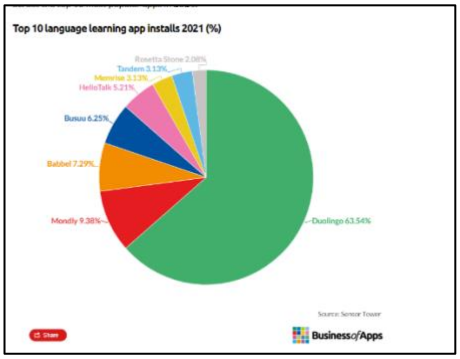
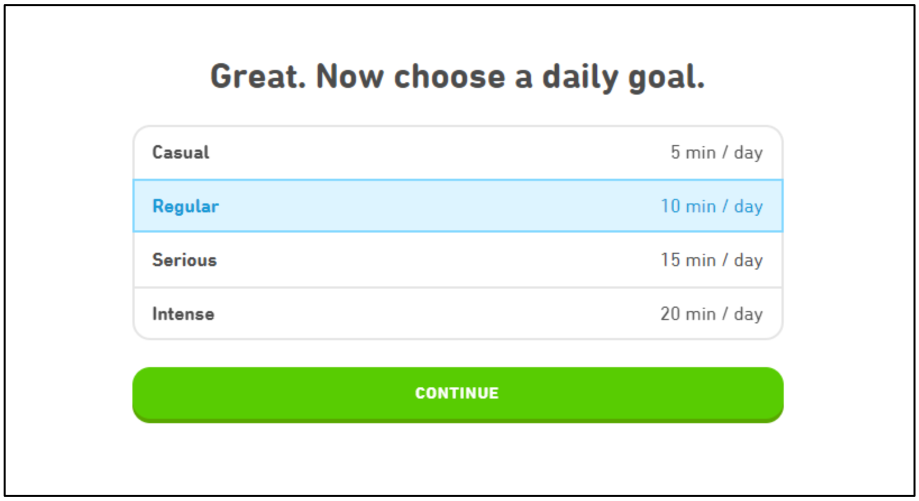


Figure 5 - Top Apps by Number of Downloads (Business of Apps) [5]

Upon closer inspection of the website, the following has been observed:

- The user is immediately greeted by a modal that asks for the intensity of their study sessions.



Great. Now choose a daily goal.

Casual 5 min / day

Regular 10 min / day

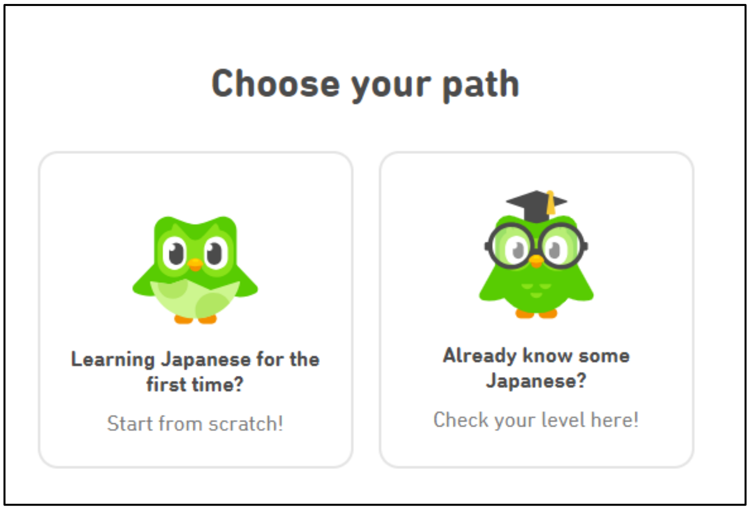
Serious 15 min / day

Intense 20 min / day

Figure 6 - Duolingo Daily Goals (Duolingo)

- Following that, a baseline for learning is set. If the user is already familiar with the language, then they are asked to take a small test so that their level of Japanese can be gauged.

[5] ("Language Learning App Revenue and Usage Statistics (2023)," n.d.) https://www.businessofapps.com/data/language-learning-app-market/



Choose your path

Learning Japanese for the first time?

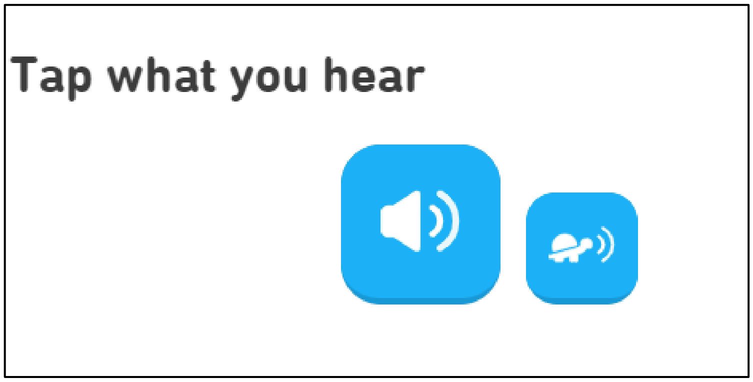
Start from scratch!

Already know some Japanese?

Check your level here!

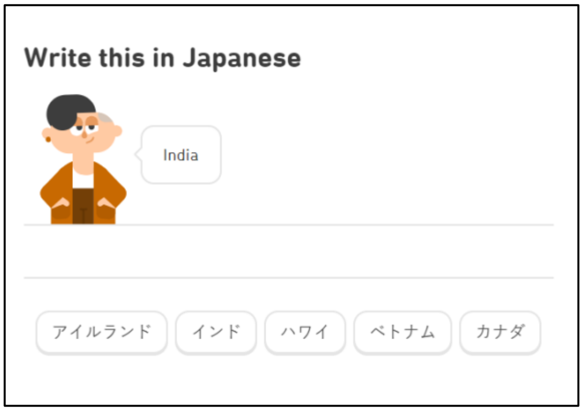
Figure 7 - Duolingo Learning Paths (Duolingo)

- The way it places and assesses is that if the user can correctly answer the questions, the difficulty of the questions is increased.
- There are several audio and text-based questions.



Tap what you hear

Figure 8 - Duolingo Audio (Duolingo)

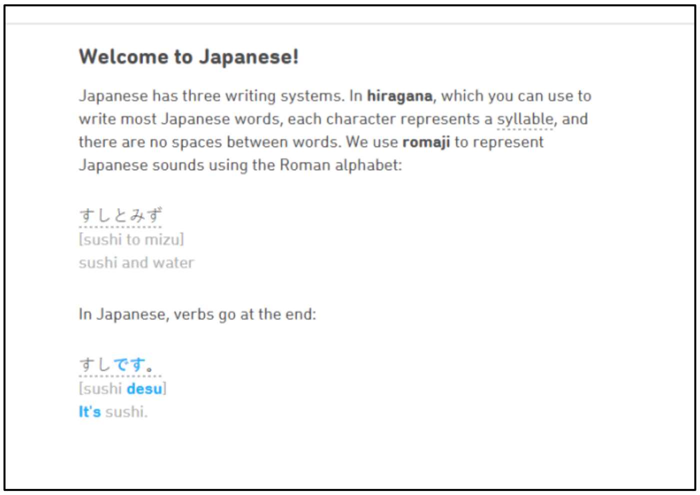
- 

Write this in Japanese

India

Figure 9 - Duolingo Question Sample (Duolingo)

- For new learners, it shows tips in the beginning which seem to be what the content they show is based on.



Welcome to Japanese!

Japanese has three writing systems. In **hiragana**, which you can use to write most Japanese words, each character represents a syllable, and there are no spaces between words. We use **romaji** to represent Japanese sounds using the Roman alphabet:

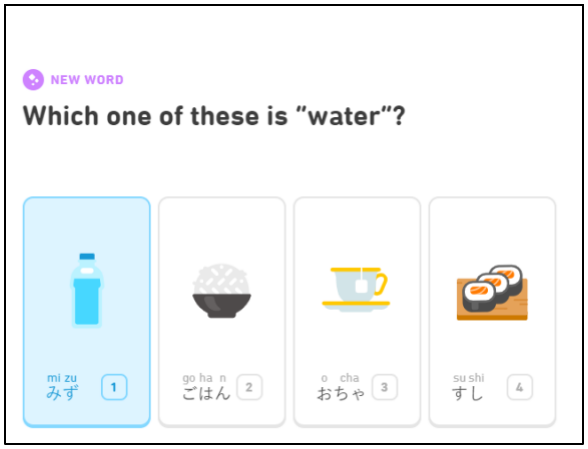
すしとみず
[sushi to mizu]
sushi and water

In Japanese, verbs go at the end:

すしです。
[sushi **desu**]
It's sushi.

Figure 10 - Duolingo Beginner Screen (Duolingo)

- Immediately after the user is thrown into a series of multiple-choice questions **without** learning anything about **grammar**, **Hiragana** or **Katakana** – the building blocks of the Japanese language, as can be seen in *Figure 9*

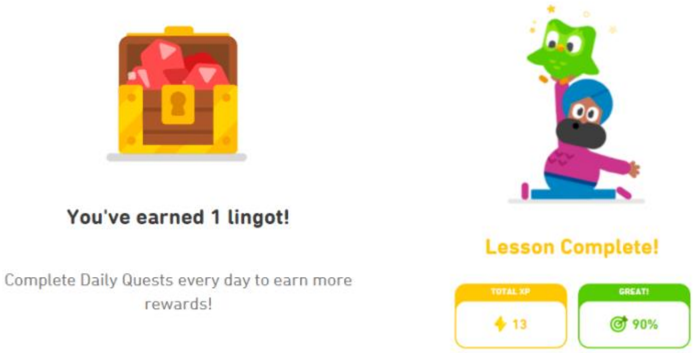


NEW WORD

Which one of these is "water"?

Figure 11 - Duolingo select reading (Duolingo)

- It tries to drill hiragana with the help of Romaji and pictures.
- Gamification comes in with its animations, and streaks on correctly answering questions in a row. On completing each lesson, the user is awarded a certain amount of experience and the 'accuracy' of their answers. They also earn a virtual currency called 'Lingots' which can be earned by completing more lessons.



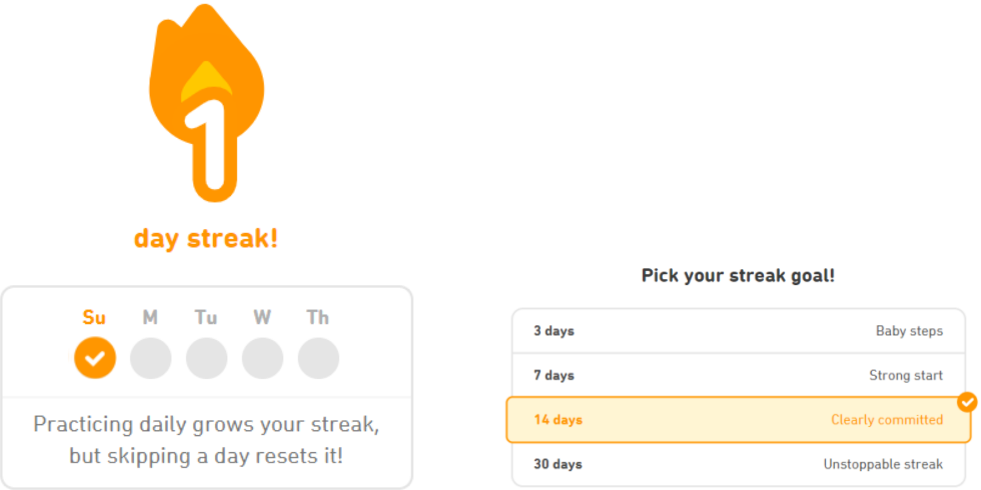
You've earned 1 lingot!

Complete Daily Quests every day to earn more rewards!

Lesson Complete!

Figure 12 – Lingots in Duolingo (Duolingo)

- The user is encouraged to grow their daily streak and the platform clearly states that their streak will be reset if they skip a day. This gives the user incentive to be consistent with their learning.



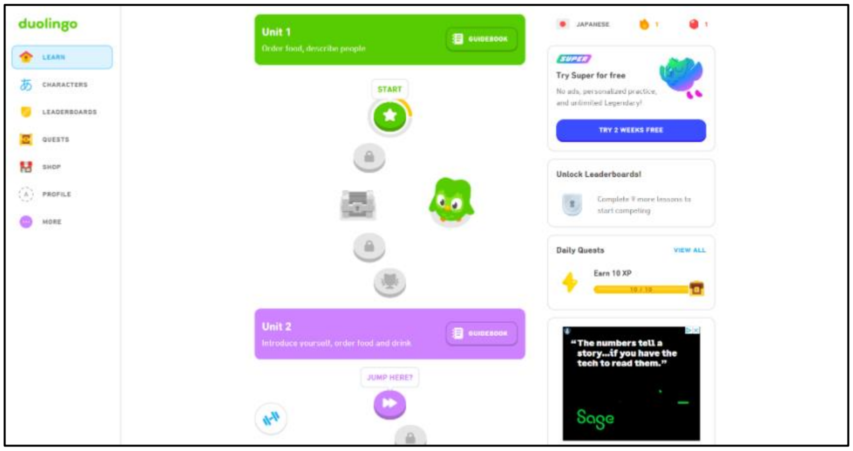
Practicing daily grows your streak, but skipping a day resets it!

Pick your streak goal!

3 days	Baby steps
7 days	Strong start
14 days	Clearly committed
30 days	Unstoppable streak

Figure 13 - Streaks in Duolingo (Duolingo)

The Dashboard



Unit 1
Order food, describe people

Unit 2
Introduce yourself, order food and drink

Try Super for free
No ads, personalized practice, and unlimited Legendary!

Unlock Leaderboards!
Complete 9 more lessons to start competing

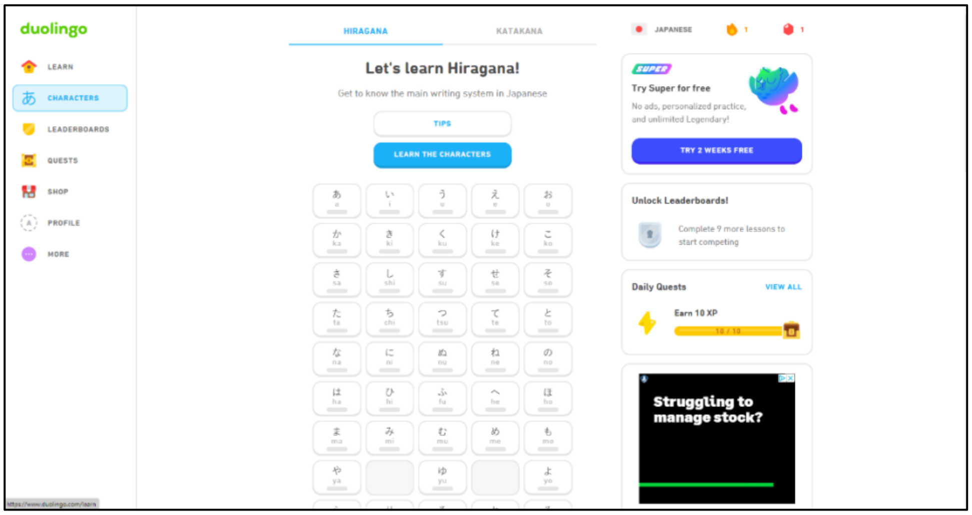
Daily Quests
Earn 10 XP

"The numbers tell a story...if you have the tech to read them."

Figure 14 - Duolingo dashboard (Duolingo)

Learn: This section of the dashboard contains 125 'Units' with 10 lessons in each unit except for the first one. The user can choose to jump to any unit, but they'll have to complete a test to advance to that unit.

Characters: Here the user can learn Hiragana and Katakana with their stroke orders and corresponding audio.



Let's learn Hiragana!

Get to know the main writing system in Japanese

あ a	い i	う u	え e	お o
か ka	き ki	く ku	け ke	こ ko
さ sa	し shi	す su	せ se	そ so
た ta	ち chi	つ tsu	て te	と to
な na	に ni	ぬ nu	ね ne	の no
は ha	ひ hi	ふ fu	へ he	ほ ho
ま ma	み mi	む mu	め me	も mo
や ya		ゆ yu		よ yo

Try Super for free

No ads, personalized practice, and unlimited Legendary!

Unlock Leaderboards!

Complete 9 more lessons to start competing

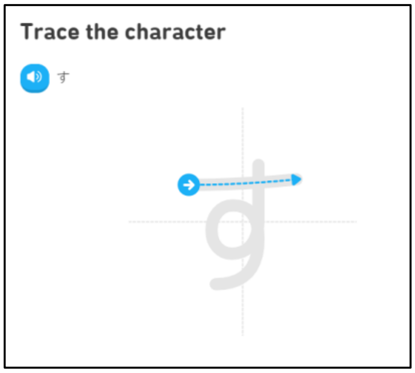
Daily Quests

Earn 10 XP

10 / 10

Struggling to manage stock?

Figure 15 - Duolingo Hiragana (Duolingo)



Trace the character

す

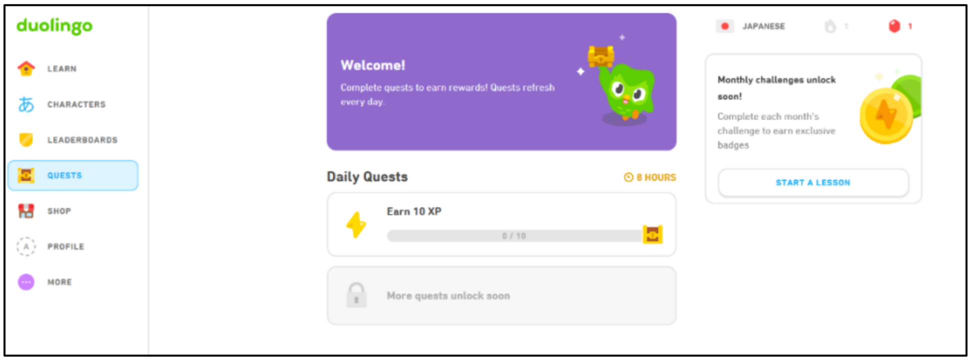
Figure 16 - Duolingo trace character (Duolingo)

Leaderboards: This section contains a tier-based leaderboard system that is unlocked after completing 9 lessons. The tiers are as follows:

Bronze, Silver, Gold, Sapphire, Ruby, Emerald, Amethyst, Pearl, Obsidian, and Diamond

There is no tier higher than Diamond. Bronze is the lowest.

Quests: The quests section lists daily and monthly quests that can be completed for experience points that help unlock badges.



Welcome!

Complete quests to earn rewards! Quests refresh every day.

Daily Quests

Earn 10 XP

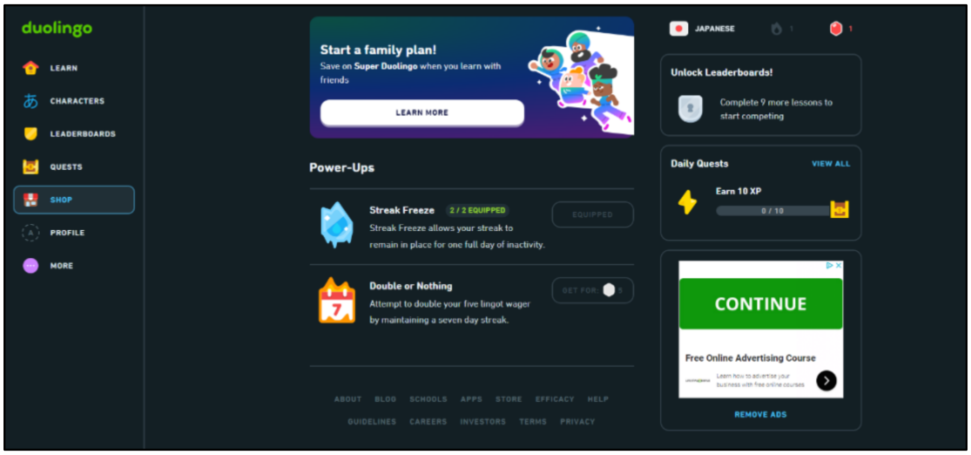
More quests unlock soon

Monthly challenges unlock soon!

Complete each month's challenge to earn exclusive badges

Figure 17 - Daily quests in Duolingo (Duolingo)

Shop: The shop allows the user to exchange virtual currency for certain *power-ups.* These specific power-ups are essentially gamified elements that allow the user to wager their streak or freeze it if they need to take a break. Thus, turning the entire learning experience into one that is goal-oriented. Although these 'goals' are not overtly related to specific learning objectives, they help the user stay consistent with their learning.



Start a family plan!

Save on **Super Duolingo** when you learn with friends

Power-Ups

Streak Freeze

Streak Freeze allows your streak to remain in place for one full day of inactivity.

Double or Nothing

Attempt to double your five lingot wager by maintaining a seven day streak.

Unlock Leaderboards!

Complete 9 more lessons to start competing

Daily Quests

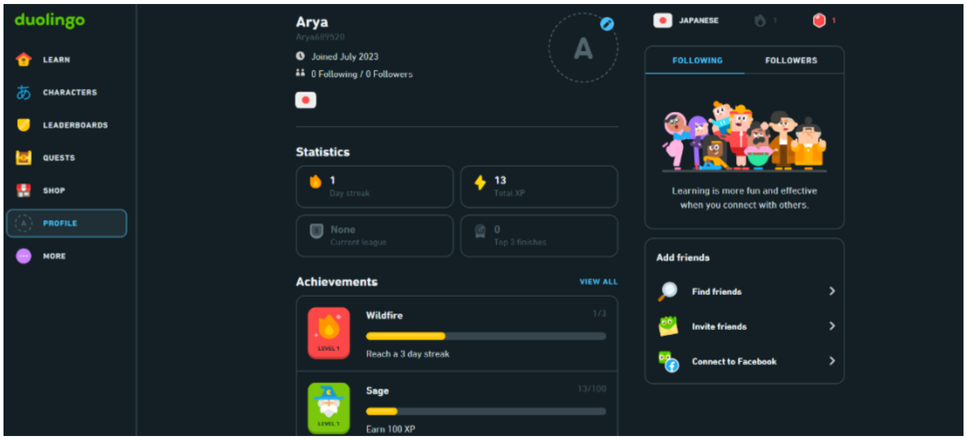
Earn 10 XP

Free Online Advertising Course

Learn how to advertise your business with free online courses

Figure 18 - Shop in Duolingo (Duolingo)

Profile: Finally, we come to the profile section which shows a user their statistics including their current achievements, streak and amount of experience earned. Each user can change their profile picture.



Arya
Arya689520
Joined July 2023
0 Following / 0 Followers

1 Day streak
13 Total XP
None Current league
0 Top 3 finishes

Wildfire 1/3
Reach a 3 day streak

Sage 13/100
Earn 100 XP

Learning is more fun and effective when you connect with others.

Figure 19 - Profile Page in Duolingo (Duolingo)

Overall, Duolingo is a highly gamified website. With attractive animations and a colourful UI, it tries to retain users with its highly interactive user experience which numbers already show is working. It is obvious why it remains the most popular language-learning alternative. However, for serious learners, all the bells and whistles don't meet their expectations as there is a serious lack of a spaced repetition system in Duolingo. This makes it less efficient compared to other alternatives that have been researched.

2.5.2 Anki

Of the four solutions reviewed here and otherwise, Anki is considered by many to be the strongest spaced repetition software out there today. Several studies have observed the potential of Anki's spaced learning system to correlate with better performance in assessments.[20]

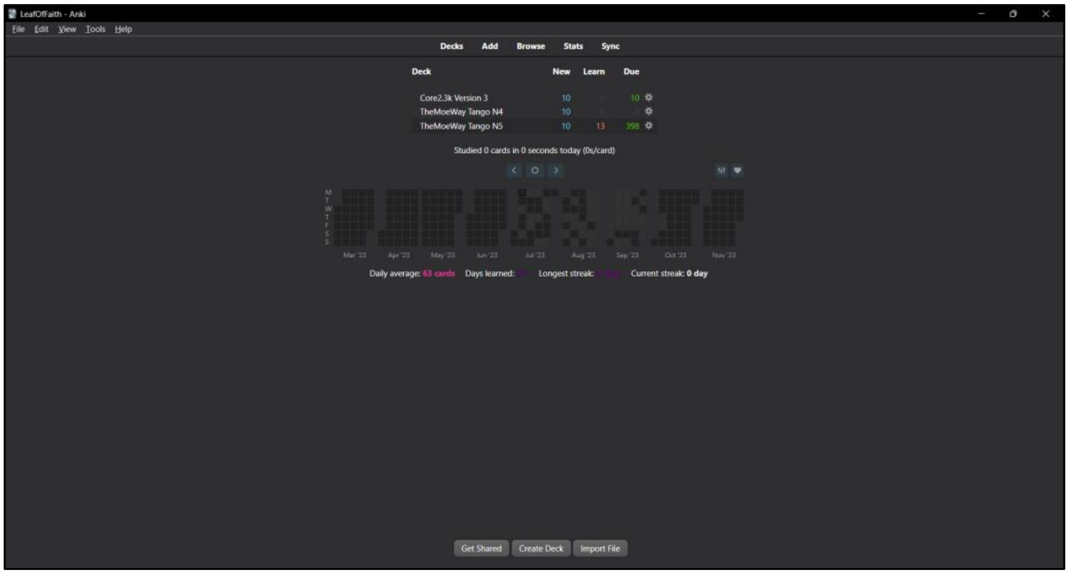
It was suggested by a meta-analysis in 1999 that learning information by spaced repetition allows learners to outperform 67% of mass presentation learners given the same number of practice sessions. [21] It was also suggested that it will vary according to the "*nature of the task being practised*", the inter-trial time interval, and the interaction between these two variables [22]

Anki utilizes two concepts to maximize learning:

Active Recall: Put simply, it is when we're asked a question and forced to remember the answer to it. It differs from passive learning in the sense that we're not sitting there only consuming information passively.

Spaced Repetition: Anki's strongest suit, based on the 'The Ebbinghaus Forgetting Curve' shown earlier in figure[number], the user is shown flashcards that are spaced out using a spaced repetition algorithm and shown to them again and again after an increasing wider interval till it becomes familiar to them.

The UI



Deck	New	Learn	Due
Core2.3k Version 3	10	0	10
TheMoeWay Tango N4	10	0	0
TheMoeWay Tango N5	10	13	398

Studied 0 cards in 0 seconds today (0s/card)

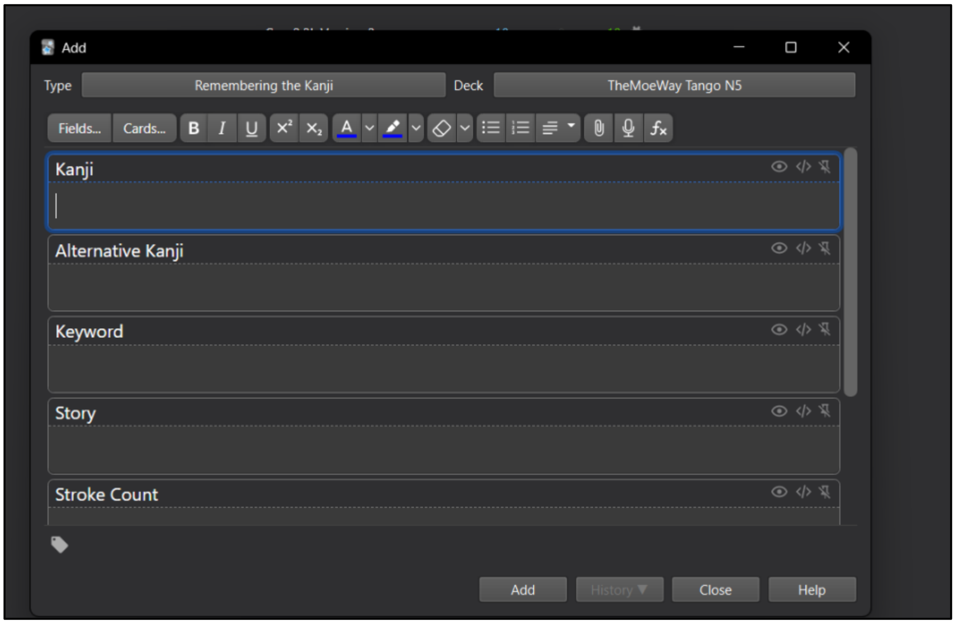
Daily average: 63 cards Days learned: Longest streak: Current streak: 0 day

Figure 20 - Anki dashboard

In stark contrast to what we saw for Duolingo, the Anki desktop application is extremely simple and barebones. The navbar consists of:

Decks: Refers to a collection of flashcards the format for which is elaborated in the figures below.

Add: Allows us to add flashcards to any deck. It auto-formats the flashcard being added to the corresponding deck. In the figure below, we can see that we can add ‘fields’ to flashcards.



Type: Remembering the Kanji
Deck: TheMoeWay Tango N5

Kanji

Alternative Kanji

Keyword

Story

Stroke Count

Figure 21 - Add card in Anki

Fields: Fields are essential ‘components in Anki’s flashcard design. They are essentially what make up Anki flashcards. Each flashcard can have multiple fields that go on the front and back of the card. This helps in organizing the information that needs to be learned. Fields and their flexibility help in customizing the flashcard to suit our study needs. Text-based fields, images and audio files can be added making it possible to study a variety of subjects.

On first using Anki, the user encounters a *Learning and Review Phase* where the user will encounter new flashcards. During this phase, they are introduced to new information and begin the learning process. Once the user has learned a flashcard, it enters the review phase. Each flashcard has an associated *Review Interval*, which determines how frequently the user will see that particular card. After successfully reviewing a flashcard, its review interval increases, and it will be seen less frequently. If the user struggles with a flashcard or makes a mistake, its review interval decreases, and the user will encounter it more often to reinforce their memory

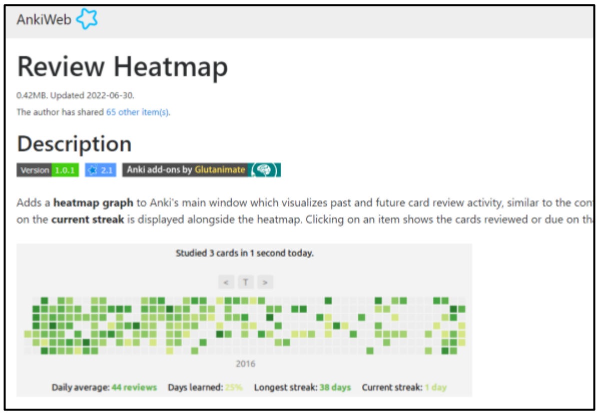
Anki's SRS is designed to maximize memory retention by timing the repetition of flashcards at optimal intervals. The system calculates the ideal time for reviewing each flashcard based on factors like the user’s past performance, the difficulty of the card, and the time elapsed since the user last reviewed it. This ensures that the user reviews cards when they are most likely to be forgotten, reinforcing your memory effectively.

Anki’s algorithm, according to their website is based on SM-2 or SuperMemo2 which is a widely adopted spaced repetition language learning model. However, it differs from it in a few ways most notably:

1 It is less strict with its learning intervals. It recognizes the need to see a card before one can memorize it so it doesn’t punish the user for the first few failures.
2 It uses 1 fail choice instead of 3.
3 It tries to avoid the pitfall of ‘low interval hell’ by factoring in successive failures which it claims is missing in SM-2. [23]

Browse allows the users to browse all available decks. There are no premade or built-in decks that are available in Anki and the user is expected to either create their flashcards or download premade decks from the AnkiWeb website. This makes the process rather tedious and there’s often a learning curve involved in understanding how the software works. Moreover, without a learning plan, a beginner language learner might find the process all too daunting to start with, inhibiting the willingness to learn which is most important.

It is important to note that although the Anki dashboard doesn’t contain Leaderboards or Virtual Currency. Anki is extremely feature rich because of an extensive library of plugins made by the community that fill in a lot of the gaps. This includes a learning heatmap and leaderboard.



Review Heatmap

0.42MB. Updated 2022-06-30.
The author has shared 65 other item(s).

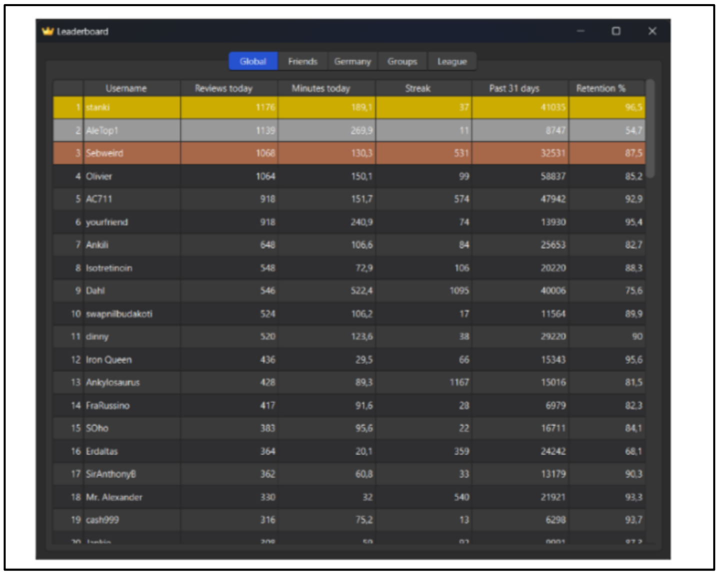
Description

Adds a **heatmap graph** to Anki's main window which visualizes past and future card review activity, similar to the con
on the **current streak** is displayed alongside the heatmap. Clicking on an item shows the cards reviewed or due on th

Studied 3 cards in 1 second today.

Daily average: 44 reviews Days learned: 25% Longest streak: 38 days Current streak: 1 day

Figure 22 - Study heatmap in Anki



	Username	Reviews today	Minutes today	Streak	Past 31 days	Retention %
1	stanki	1176	189,1	37	41035	96,5
2	AleTop1	1139	269,9	11	8747	54,7
3	Sebweird	1068	130,3	531	32531	87,5
4	Olivier	1064	150,1	99	58837	85,2
5	AC711	918	151,7	574	47942	92,9
6	yourfriend	918	240,9	74	13930	95,4
7	Ankili	648	106,6	84	25653	82,7
8	Isotretinoin	548	72,9	106	20220	88,3
9	Dahl	546	522,4	1095	40006	75,6
10	swapnilbudakoti	524	106,2	17	11564	89,9
11	dinny	520	123,6	38	29220	90
12	Iron Queen	436	29,5	66	15343	95,6
13	Ankylosaurus	428	89,3	1167	15016	81,5
14	FraRussino	417	91,6	28	6979	82,3
15	SOho	383	95,6	22	16711	84,1
16	Erdaltas	364	20,1	359	24242	68,1
17	SirAnthonyB	362	60,8	33	13179	90,3
18	Mr. Alexander	330	32	540	21921	93,3
19	cash999	316	75,2	13	6298	93,7

Figure 23 - Leaderboard in Anki (AnkiWeb) [6]

Overall, this is the most polished application algorithmically. However, it lacks the colourful and engaging user experience that is offered by the likes of Duolingo. This makes the learning process rather dry for users who have been using it for a while. The lacklustre user experience is made up for by a plethora of community-created content.

[6] https://ankiweb.net/shared/info/41708974

2.5.3 Clozemaster

Clozemaster is yet another language learning platform that focuses on improving vocabulary and comprehension through context-based sentence completion exercises.

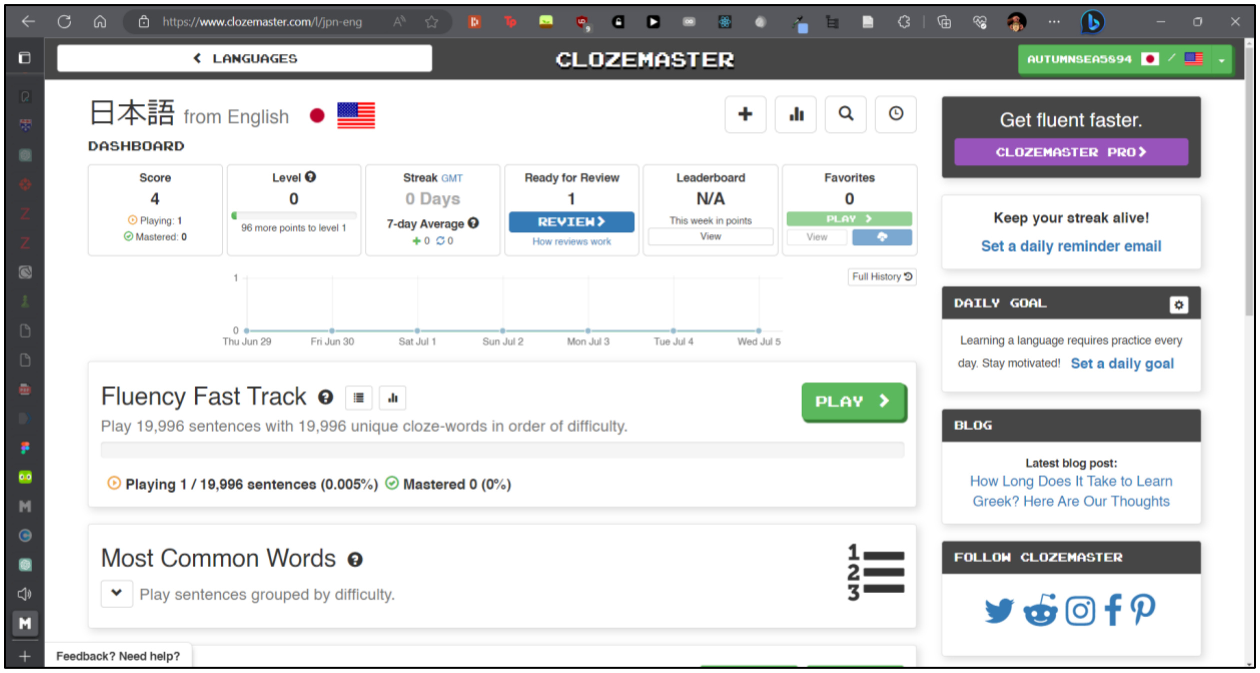


Figure 24 - Clozemaster dashboard (Clozemaster)

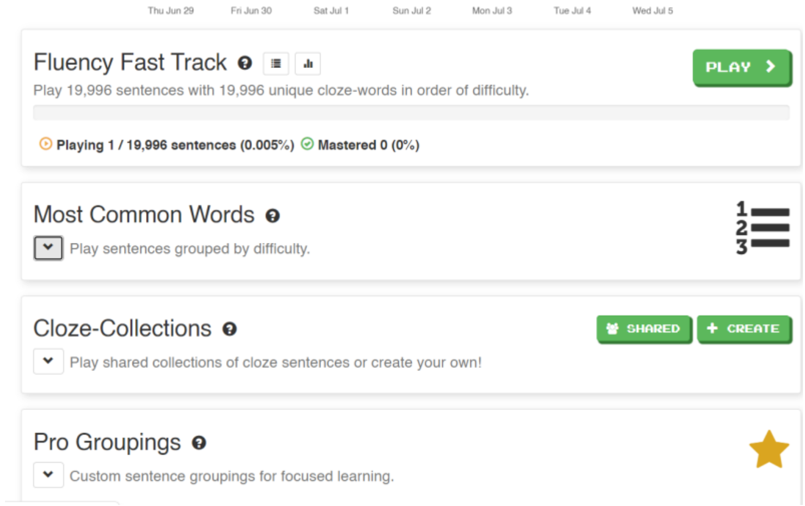
The user is greeted with a dashboard that is not as minimal as Anki but like what was observed in Duolingo. ‘Score’ and ‘Level’ increase with the number of reviews and ‘Streak’ is the same as in Duolingo.

In Clozemaster, a "Cloze" refers to a specific type of exercise where you are presented with a sentence in the target language and one or more words or phrases within the sentence are replaced with blanks. Your task is to fill in the missing word or phrase based on the context provided. [24]

The term "Cloze" is derived from the linguistic concept of a "cloze test." In language learning, a cloze test is an assessment technique that involves removing words or phrases from a text, and the learner must fill in the missing elements. Clozemaster adapts this concept and incorporates it into its sentence completion exercises. This is although in many cases the ‘Cloze Procedure’ is not acknowledged or as effective as other learning methods. [25]

The essential components of Clozemaster are listed as follows:

Learning Tracks: Offering multiple learning tracks instead of a singular unit, a 100s of unit long track gives the user a much more structured approach than Duolingo



Fluency Fast Track

Play 19,996 sentences with 19,996 unique cloze-words in order of difficulty.

Playing 1 / 19,996 sentences (0.005%) Mastered 0 (0%)

Most Common Words

Play sentences grouped by difficulty.

Cloze-Collections

Play shared collections of cloze sentences or create your own!

Pro Groupings

Custom sentence groupings for focused learning.

Figure 25 - Learning tracks in Clozemaster (Clozemaster)

Reviewing: Clozemaster breaks down its review process on its website as follows

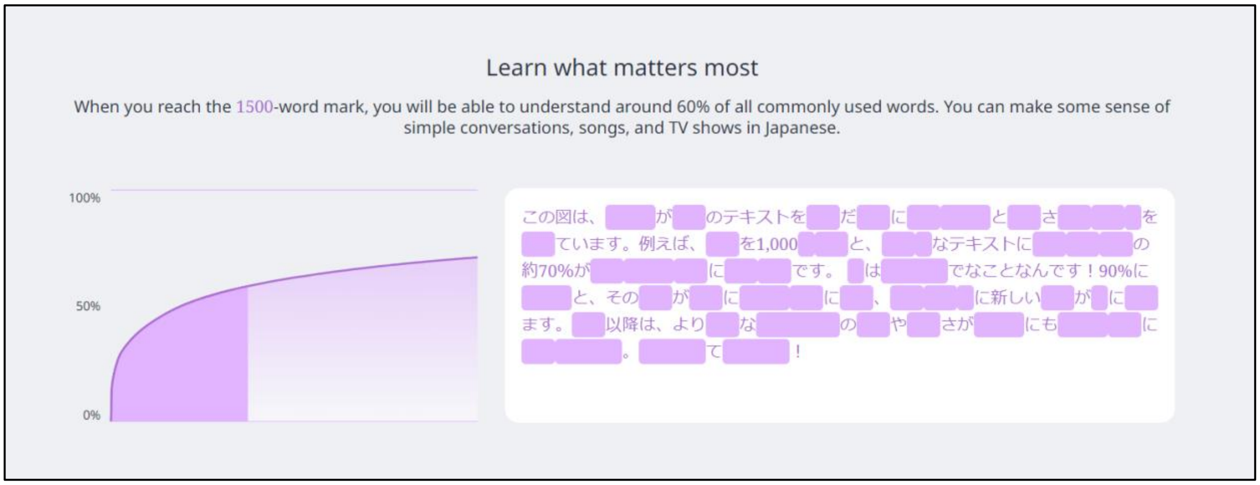
Sentences undergo assessment based on spaced repetition intervals like the scheduling principles employed by platforms like Memrise and Anki. The default intervals consist of 1 day, 10 days, 30 days, and 180 days, which depend on the consecutive correct responses to a sentence. Subscribers with a Pro membership have the flexibility to adjust these intervals. For instance, a newly introduced sentence that is answered correctly enters the review cycle after one day, signifying 25% mastery.

Upon successive correct responses, the interval extends to 10 days (50% mastery) and subsequently 30 days (75% mastery). Achieving correctness again results in the sentence being considered fully mastered (100%) and placed on a 180-day review cycle. This periodicity persists as long as the sentence continues to receive correct answers. However, an incorrect response at any stage resets the sentence's mastery progress to 0%

2.5.4 Lingvist

The last of the language-learning applications reviewed, Lingvist promises vocabulary learning for as little as 10 minutes a day. This differs from the approach of Anki where review times can quickly ramp up depending on how many flashcards the user has previously reviewed.

As can be seen from the figure below, Lingvist greets the user with an entirely different concept of learning the language through 'the smart and efficient learning of vocabulary'.



Learn what matters most

When you reach the 1500-word mark, you will be able to understand around 60% of all commonly used words. You can make some sense of simple conversations, songs, and TV shows in Japanese.

100%
50%
0%
この図は、 が のテキストを だ に と さ を
ています。例えば、 を1,000 と、 なテキストに の
約70%が に です。 は でなことなんです！90%に
と、その が に に 、 に新しい が に
ます。 以降は、より な の や さが にも に
。 て ！

Figure 26 - Lingvist initial screen (Lingvist)

It gives an idea of what different stages of vocabulary learning look like. It also does a very good job of introducing learners to the different scripts that are used in the Japanese language.

The user can immediately start typing the answers in both Romaji and Hiragana

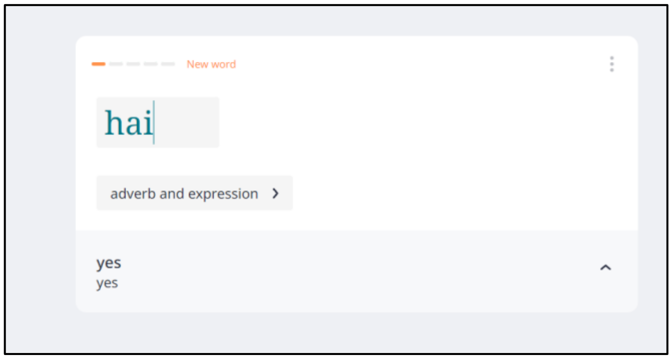


Figure 27 - User input in Lingvist (Lingvist)

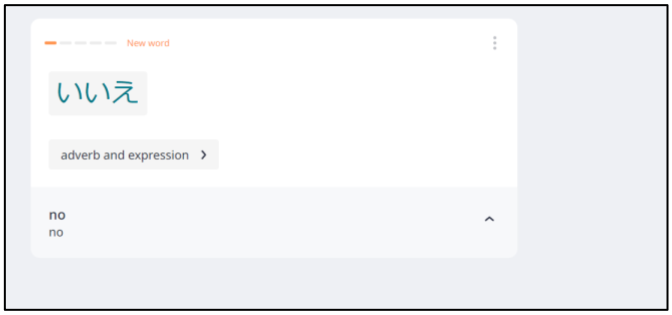
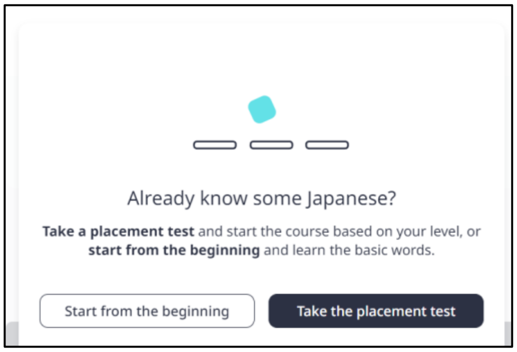


Figure 28 - User input in Lingvist 2 (Lingvist)

In case the learner correctly answers the first few questions, it asks them to take a placement test. If not, they can choose to start from the beginning.

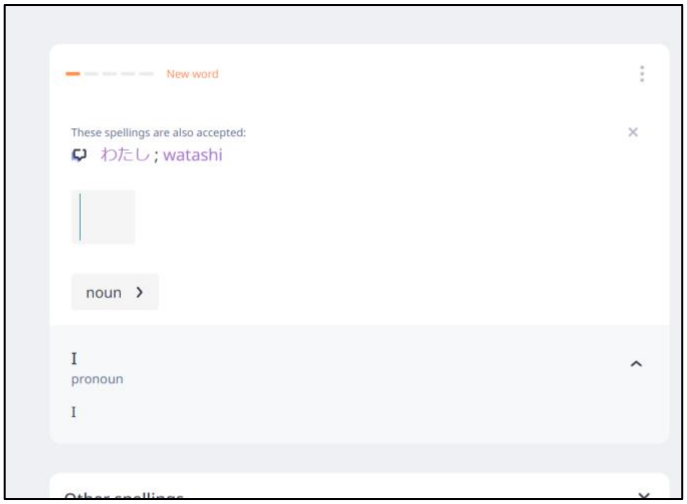


Already know some Japanese?

Take a placement test and start the course based on your level, or **start from the beginning** and learn the basic words.

Figure 29 - Learning path Lingvist (Lingvist)

Should the user start from the beginning of the course, they are asked to type in the romaji, kanji, hiragana, or katakana spelling of the word in front of them. If they answer incorrectly, they are shown the correct answer.



These spellings are also accepted:

わたし ; watashi

Figure 30 - User input Lingvist (Lingvist)

Lingvist also offers a detailed grammar tips section that can be used to understand grammar patterns in the language.

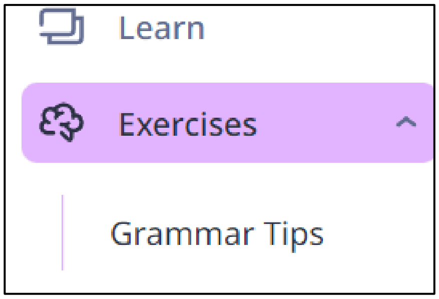
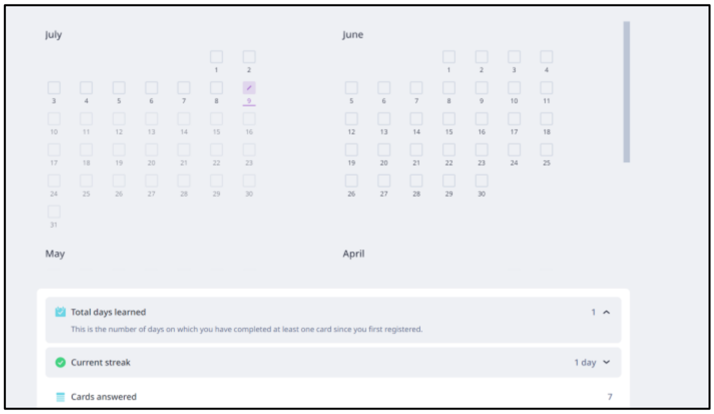


Figure 31 - Grammar section Lingvist (Lingvist)

The insight section shows statistics about total days learned, current streak and cards answered.



Total days learned 1

This is the number of days on which you have completed at least one card since you first registered.

Current streak 1 day

Cards answered 7

Figure 32 - Statistics section in Lingvist (Lingvist)

The Lingvist website details how Hermann Ebbinghaus discovered patterns in memorization and learning/forgetting curves. The application also used a spaced repetition system that is used to gradually increase intervals to create the most optimal learning schedule for each learner. It is at the heart of Lingvist and they about their Knowledge Mapping Engine or KME as they like to call it. They claim it adapts to the learner's current level of knowledge and predicts the words they don't know. It boasts a very clean user interface as well. [26]

In conclusion, it is certainly a more polished overall experience than Duolingo. It misses out on a key feature which is gamification, however, it does the job of optimizing language learning well. Since the user is shown a ton of new vocabulary without learning the scripts, it presumably has a steeper learning curve than the other applications we've seen but at the same time, is one of the most polished applications out there.

Table 1 - Table comparing functionalities of different language-learning apps/websites

Application Name	Spaced-Repetition	Gamification	Flashcards/Mode of learning	Audio	Content
Duolingo	SRS Algorithm used but not as effective	Yes. Highly gamified. Very engaging. Progress tracking	Methods like matching pairs used, along with sentence completion	Yes, audio is provided for most sentences.	Provided, structured.
Anki	Yes. Highly flexible algorithm and proved to be efficient	None unless plugins are used.	Highly customizable flashcards that the user has control over	Can be used, can be added by users, or provided in decks.	None are provided out of the box; users must look for decks or create flashcards.
Clozemaster	Utilizes SRS	Yes, marketed as a 'game to rapidly expand vocabulary'	Sentence-based learning with multiple choices offered	Yes, audio was provided for sentences after answered	Provided, structured.
Lingvist	Utilizes SRS	Isn't overly gamified. Simple interface	Sentence based learning	Yes, audio for vocabulary provided	Provided, structured.

The table above shows that most of the apps use Spaced Repetition in some form or the other, but the level of gamification tends to vary. The mode of learning also differs from app to app but all of them offer options to get audio. Besides Anki, every other learning app offers a structured program. Anki is dependent on the decks the learner might create or pre-existing decks put out by the Anki community.

2.6 Spaced Repetition Algorithm

Spaced repetition is an evidence-based approach to learning, based on the principle that there should be scientific evidence behind education practices. It is performed primarily with flashcards. New and original flashcards are shown more often; older and already-seen flashcards are less frequently shown thus exploiting the spacing effect.

For this project, it was researched and found that there are 3 primary families of scheduling:

- Neural network based.
- Leitner system
- SuperMemo family of algorithms

One of the biggest limitations of this project is testing whether the app can successfully emulate spaced repetition learning. This is especially true if a new Spaced Repetition algorithm is written from scratch.

To work with this limitation, it was found that it is best to stick with a tried and tested spaced repetition algorithm. SuperMemo's SM-2 algorithm is the most well-documented of all algorithms out there.

The two most popular apps for second-language acquisition were compared, namely Duolingo and Anki. Duolingo published a paper [27] where they presented their 'half-life regression' model for second-language acquisition but another paper [28] challenges Duolingo's rigid model to Anki's highly customizable SM-2-based algorithm. Results show that Anki has a more reliable and trustworthy algorithm when it comes to short-term and mid-term performance. Duolingo is not very accurate in working out the 'forgetting curve'. Duolingo is the most popular app because of its gamification, which is far more entertaining than most apps out there.

For this project, it was decided that it would be more efficient to focus on implementing something like Anki and SM-2.

The following algorithm will be used [29]

```
algorithm SM-2 is
    input:  user grade $q$
            repetition number $n$
            easiness factor $EF$
            interval $I$
    output: updated values of $n$, $EF$, and $I$

    if $q \geq 3$ (correct response) then
        if $n = 0$ then
            $I \leftarrow 1$
        else if $n = 1$ then
            $I \leftarrow 6$
        else
            $I \leftarrow \text{round}(I \times EF)$
        end if
        increment $n$
    else (incorrect response)
        $n \leftarrow 0$
        $I \leftarrow 1$
    end if

    $EF \leftarrow EF + (0.1 - (5 - q) \times (0.08 + (5 - q) \times 0.02))$
    if $EF < 1.3$ then
        $EF \leftarrow 1.3$
    end if

    return $(n, EF, I)$
```

The variables are defined as follows:

n – In this algorithm, n represents the repetition number or the number of times a specific piece of information has been presented to the user and reviewed.

EF (Easiness Factor): The Easiness Factor represents how well the user remembers a particular piece of information. It's a value that is adjusted after each review session based on the user's response grade (q). A higher EF indicates that the user finds the material easier to remember and therefore can review it at longer intervals. The algorithm uses a formula to adjust the EF based on the user's response grade and other factors.

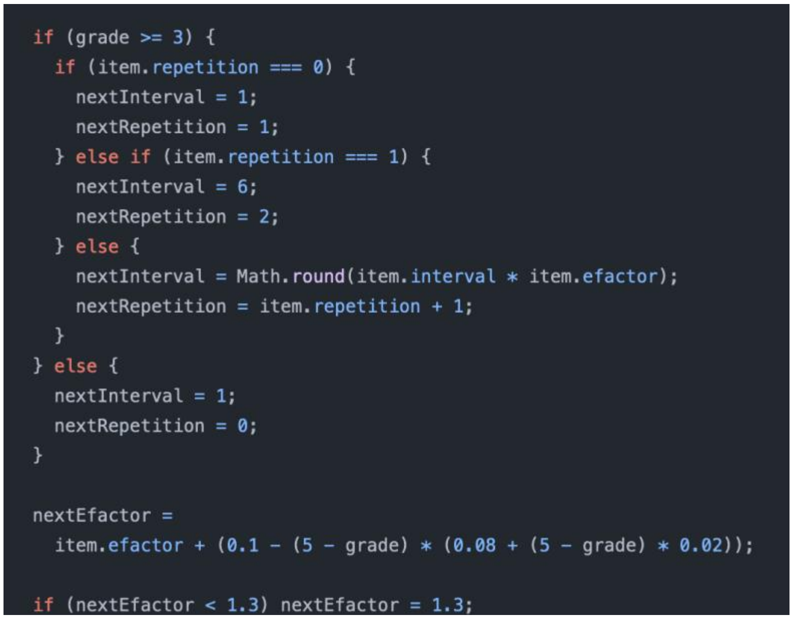
I (Interval): The Interval is the period between consecutive reviews of the material. It represents how long the user should wait before reviewing the information again. The algorithm calculates the interval based on the repetition number n, the EF, and certain conditions. As the user reviews the material more times, the interval increases for better retention. The SuperMemo algorithm is a learning technique designed to optimize the process of memorization and long-term retention of information. This approach has served since its inception as the foundational principle in numerous spaced repetition software applications.

Fundamentally, the algorithm's concept revolves around strategically timing review sessions to optimize memory recall. This principle draws from the psychological phenomenon known as the 'spacing effect', wherein information is more effectively remembered through spaced-out reviews.

The algorithm is broken down as follows:

- **Initial Learning**: When encountering new information to be memorized, such as a term or concept, users input it into the SuperMemo system as a flashcard or item.
- **First Review**: Following the initial study, the information undergoes the first review. Successful recall prompts scheduling for the subsequent review, whereas errors lead to an earlier review. This is determined by the ease factor (EF). The time until the next review is the interval (I)
- **Review Scheduling**: The algorithm determines the timing of subsequent reviews based on prior performance. Accurate recall extends the interval (I) until the next review, while difficulty in recall shortens the interval.
- **Spacing Intervals**: Intervals between reviews are calculated based on the "Ebbinghaus forgetting curve," which outlines how memory degrades over time without reinforcement. By intelligently spacing out reviews at increasing intervals, the algorithm leverages the spacing effect to bolster memory retention.
- **Continual Reviews**: Ongoing reviews prompt the algorithm to adjust intervals according to individual learning patterns and memory capabilities. It considers variables like review frequency and past performance and where n (successful recalls)
- **Review Priority**: The algorithm also accommodates the volume of information to be memorized. It prioritizes items due for review and those posing greater difficulty, enabling users to focus on the most crucial content or content they are most likely to forget immediately.

This project builds on a TypeScript implementation of a Delphi implementation of the SuperMemo2 algorithm. Wozniak, while working on his original implementation found that he was able to acquire 270/items/year/min with an overall retention of 89.3%. The SM-2 algorithm marked a major improvement of the original SuperMemo method. [29] The typescript implementation that builds on the original Delphi implementation is as follows.



```
if (grade >= 3) {
  if (item.repetition === 0) {
    nextInterval = 1;
    nextRepetition = 1;
  } else if (item.repetition === 1) {
    nextInterval = 6;
    nextRepetition = 2;
  } else {
    nextInterval = Math.round(item.interval * item.efactor);
    nextRepetition = item.repetition + 1;
  }
} else {
  nextInterval = 1;
  nextRepetition = 0;
}

nextEfactor =
  item.efactor + (0.1 - (5 - grade) * (0.08 + (5 - grade) * 0.02));

if (nextEfactor < 1.3) nextEfactor = 1.3;
```

Figure 33 - TypeScript implementation

3.0 Building the App

3.1 Gamified Spaced-repetition-based language learning app

The function of the spaced-repetition-based language learning app is to optimize the learning process for users. The app will focus initially on the Japanese language while focusing on Kanji, vocabulary, and grammar retention. The application's core feature is the spaced repetition algorithm which will ensure efficient and long-term retention of language knowledge. As users progress through the lessons, the app will intelligently schedule the words and Kanji after strategic intervals. The algorithm is based on SuperMemo2 which has been widely used and tested.

Furthermore, the app will provide a sense of competition and progress through its intuitive leaderboard system and achievements. As the user learns more and completes milestones, the app will reward the user with badges to celebrate their progress.

In terms of user experience, the app will be designed with a user-friendly interface in mind. This entails fluid animations, minimal loading times and easy access to the most important features. With its effective spaced repetition system and comprehensive language resources, the app aims to empower its users to achieve their language learning goals enjoyably and efficiently.

3.2 Agile Development

Agile is a software development methodology that focuses on flexible and iterative development. It aims to efficiently deliver a good product by responding to changing requirements throughout the development process.

A set of objectives were proposed to produce this app, and a comprehensive set of requirements was drawn up. These requirements were then followed up by defining the approach toward design. The design phases were planned, and then, implementation was achieved based on the chosen programming language that best suited the purpose of the app. All iterations of the app were built along these requirements and these requirements helped keep the development of the app in check.

An agile mode of development was particularly important when the technologies used in developing the app changed and evolved. The process of development was cyclical, and the author had to come up with solutions while staying true to the functional requirements. By iteratively developing the app and testing it after each cycle, the author was able to land on the most optimal solutions.

For testing, unit tests as well as overall tests were carried out for each stage of the development cycle with testing libraries. The app was iteratively made following an agile model of development. An agile method of development was preferred over a more linear method like a waterfall as the rigidity of the latter was inefficient. An agile method also allowed for flexibility in changing libraries and technologies used in the app. [30]

3.2 Technologies Used

3.2.1 React

The front end of the app was built using React [31] , a highly popular and supported JavaScript library for building user interfaces. React allows for the creation of interactive and dynamic components, making the app highly responsive and engaging for learners.

This app follows the component-based architecture of React and ensures code reusability, maintainability, and scalability of the user interface and state. The app has a lot of moving parts that need fine-grained optimisations offered by React. At the core of React's philosophy lies the idea of crafting components that are both reusable and modular. This guiding principle paved the way for the development of intricate and versatile components that seamlessly melded into the application's framework. These components encompassed a spectrum of elements, spanning from buttons and forms to intricate data presentations. Meticulously designed to encapsulate their unique functionalities and aesthetics, these components became self-contained units, forming the foundation of the application's architecture. This emphasis on modularity ultimately leads to elevated code structure, heightened potential for reuse, and enhanced ease of maintenance which was deemed to ideal for this application that has a lot of potential to be worked on further.

3.2.2 Typescript

To further enhance code quality and introduce strict type-checking, TypeScript was leveraged. [32] TypeScript is a programming language developed by Microsoft that extends and enhances JavaScript by adding static typing and other advanced features. It is often referred to as a "superset" of JavaScript because it builds upon the syntax and functionality of JavaScript while introducing additional capabilities.

TypeScript code can be ‘transpiled’ (converted) into regular JavaScript code, making it compatible with existing web browsers and environments. Testing the app becomes a much less arduous process with its static typing, improved tooling, and detection of potential errors.

When addressing the aspect of testing, TypeScript plays a pivotal role in simplifying the process. With static typing in place, a significant portion of potential bugs and errors are caught at compile-time, reducing the need for extensive runtime testing. This leads to a more efficient testing phase, as developers can focus their efforts on scenarios that might not be covered by the “type” system.

This application makes use of TypeScript to ensure that when flashcards are being generated by the system, they follow the interface of a flashcard defined within the application. Every prop and every object that is passed between components is type-checked to ensure sanitized code.

3.2.3 Supabase

For handling users and creating a database, Supabase [33] ,an open-source alternative to Firebase was employed. The integration process is seamless, and it makes adding authentication to any application incredibly swift a process.

Furthermore, Supabase now comes packaged with PostgreSQL which allowed the author to do away with the use of a separate server. PostgreSQL (often referred to as Postgres) is a type of SQL database. SQL stands for Structured Query Language, which is a standard language used to manage and manipulate relational databases. PostgreSQL is an open-source relational database management system that uses SQL as its query language. Data synchronization is maintained through React itself. It was decided that PostgreSQL is better than using a non-relational database as the schema with unique data for each user, is complex enough to warrant a relational database that can handle complex queries. PostgreSQL is widely established, making it easier for developers to debug problems with all the resources and help that are available online.

For this application, the author sought a solution that satisfied the need for both a database as well as an authentication service and Supabase was the answer to it. It comes with extensive documentation and support, as well as a website that provides all the SQL queries the user needs to fetch data through their client. Supabase also helps in easily visualizing data for easy access and debugging.

3.2.4 Tailwind CSS

Finally, Tailwind [34] helped streamline the app's design and styling process. Optimized for all reasonable viewports, the app boasts compatibility with most modern devices that open it on the web. Tailwind CSS is a utility-first CSS framework designed to expedite web development through a collection of predefined classes. Unlike conventional component-centric methodologies, Tailwind CSS promotes the construction of user interfaces by combining compact utility classes, which directly apply distinct styling attributes to HTML elements.

This strategy empowers developers to swiftly generate adaptive and adaptable designs without resorting to extensive personalized CSS coding. Covering an extensive array of design and arrangement requirements, Tailwind CSS's utility classes have garnered popularity for simplifying the development journey and ensuring the uniform creation of visually pleasing web applications.

Tailwind CSS is being used in this app to create style guides that can be easily adopted by any future developers. It has extensive documentation, and it makes creating a responsive application – an application that adapts to all screen sizes, very simple.

Backendless Approach

This project uses a Backendless approach over a traditional application with a backend. A backendless app is an application that operates without the traditional need for a dedicated server-side backend infrastructure. Backendless apps are popular for smaller-scale apps and good reason. They are quick to build and easy to scale. It is made possible in this case through React and Supabase which allows the application to make all calls through the front and render information as and when needed, and where the services and operations and handled by a cloud-based service. This also alleviates the need to maintain backend code which is not written in the application.

To summarize the benefits of a backendless app:

- It helps streamline the development process and ensure rapid development which given the time constraint of this project, was found to be the most optimal approach.
- This helps keep the app online and hosted for others to make use of or build upon.

This application in particular makes use of Supabase to achieve that end by offering solutions for the following functionality:

- Supabase offers a powerful and scalable database service based on PostgreSQL. It allows you easy set up and manage the database without the need to configure and maintain a separate database server. It can be used to create tables, define relationships, and perform complex queries using PostgreSQL.
- Supabase includes a comprehensive authentication system that takes care of email/password login and signup along with OAuth flows for Google and GitHub.
- All queries to Supabase are performed using the React frontend.

To set realistic and achievable goals, a set of functional requirements were defined so that the app could be measured against clear, simple, and unambiguous objectives.

3.3 Functional Requirements

The functional and non-functional requirements are divided into three categories — P1, P2 and P3. These represent how critical the corresponding requirement is while developing the app.

- **P1** — The most important of the 3. It represents the core functionality of the website and requirements that are deemed to be indispensable and uncompromisable.
- **P2** — These requirements supplement the core functions of the app but are not given precedence over P1 requirements. For the most part, these requirements build on P1 requirements.
- **P3** — These are the least important requirements that are still in the scope of the app but represent some extended functionality of the app that does not affect the primary functions of the app itself. Not meeting these requirements does not hinder the primary objectives the author set out to complete.

Table 2 - Functional Requirements

	Functional Requirements	Priority
R1	**Content is to be displayed in the form of flashcards.** All content in the application will be shown to the learner in the form of flashcards	**P1**
R2	Each flashcard will be shown to the user for a **maximum of 60 seconds** after which the flashcard will be automatically marked as failed.	**P1**

R3	**Spaced Repetition System** The app should incorporate a spaced repetition system which optimizes learning by presenting flashcards at intervals based on the user's learning progress and retention rate.	**P1**
R4	**Gamified Elements** The application should incorporate gamification techniques to enhance the language learning journey. This includes integrating elements such as rewards, achievements, and progress tracking to simulate a game-like experience.	**P1**
R5	**Achievement Recognition** Users' accomplishments, such as completing a set of lessons or reaching certain skill levels, should be acknowledged through achievements. These achievements can serve as milestones and markers of progress in their language-learning journey.	**P2**
R6	**Reward System** The app should implement a reward system where users receive virtual rewards, badges, or tokens upon completing lessons, achieving specific goals, or demonstrating proficiency.	**P3**
R7	**Progress Tracking** Learners should have access to detailed progress reports and analytics, showcasing their strengths and areas for improvement. This feature will allow learners to monitor their language proficiency and set goals accordingly.	**P2**
R8	The app should include **sentence completion** questions that promote user input and engagement	**P2**
R9	The application should have **built-in decks** for different JLPT levels	**P1**
R10	**Login** A user should be allowed to log in with their email and password, a magic link or GitHub.	**P1**
R11	If a user is not signed in, they are prompted to go to sign-up.	**P1**
R12	**Review** When a user starts learning a new deck, cards that need to be reviewed in that deck are shown first.	**P1**
R13	The app should not allow more than **100 review cards for next time.** This limit should be achieved by stopping the user from learning new cards once they hit 100 review cards due	**P2**
R14	The **leaderboard** should contain **3 filters**, study streak, kanji mastered, and words mastered	**P1**
R15	The **leaderboard** should be **globally accessible**	**P1**
R16	The profile page should display all user stats and how many cards they have mastered	**P2**
R17	The profile page should display all essential profile information including email ID, password, and profile picture	**P2**

3.2 Non-functional Requirements

Table 3 - Non-functional requirements

	Non-functional Requirements	**Priority**
NF1	**Response time** The application should have fast response times and smooth navigation. There should be minimal delay when transitioning from one part of the application to another.	**P1**
NF2	**Usability T**he user interface should adhere to Nielson's usability guidelines so that the app is user-friendly.	**P2**

NF3	**UI Guidelines** The app should refer to Material Design guidelines while styling components and the UI to ensure adherence to readability and accessibility	**P2**
NF4	The app should be designed to handle multiple users.	**P2**
NF5	WCAG (Web Content Accessibility Guidelines) should be adhered to	**P2**
NF6	User progress should be successfully backed up on the cloud and data loss should be prevented	**P2**
NF7	The app should work seamlessly across all kinds of devices if they have a web browser. Thus, ensuring a consistent experience for all users.	**P2**
NF8	The app should adapt to different screen sizes, resolutions, and viewports, providing an optimal user experience on various devices including smartphones, tablets, and desktops.	**P1**

3.4 Design

An early prototype of the system was necessitated because of the choice of an agile mode of development. The stages of designing the application were broken down as follows:

- Interface design: This involved designing the user interface (UI) and user experience (UX). The goal was to create an intuitive and at the same time visually appealing interface that would enhance user interaction and engagement and build on the shortcomings of some of the researched apps that boasted superiors SRS systems but lacklustre UI.
- Functional design: In this phase, the author outlined the specific functionalities and features that the application would offer. This included defining user roles and permissions, as well as the various tasks and actions users could perform within the system.
- Architectural design: The architectural design was crucial for determining the overall system's structure, including the choice of technologies, frameworks, and platforms. It also involved outlining the system's modules and their interactions.
- Error handling and recovery design: To enhance the application's robustness, the team devised strategies for handling errors and implementing recovery mechanisms in case of unexpected failures.
- Database design: This stage focused on defining the structure of the application's database, including the tables, relationships, and data schema. It allows the app to track user progress, store learning materials, and facilitate seamless interactions between users and the application.

3.4.1 App Design

In designing the different sections of the application, an initial sitemap of the website was created. This gave a clear idea of what to include on the website so that the user is greeted by a succinct-looking UI that is not bloated.

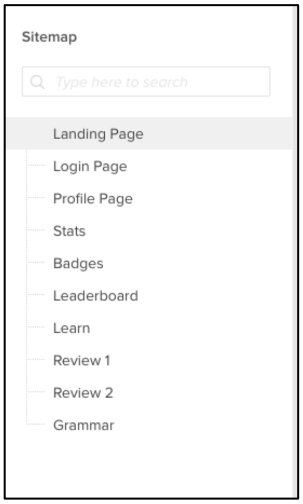
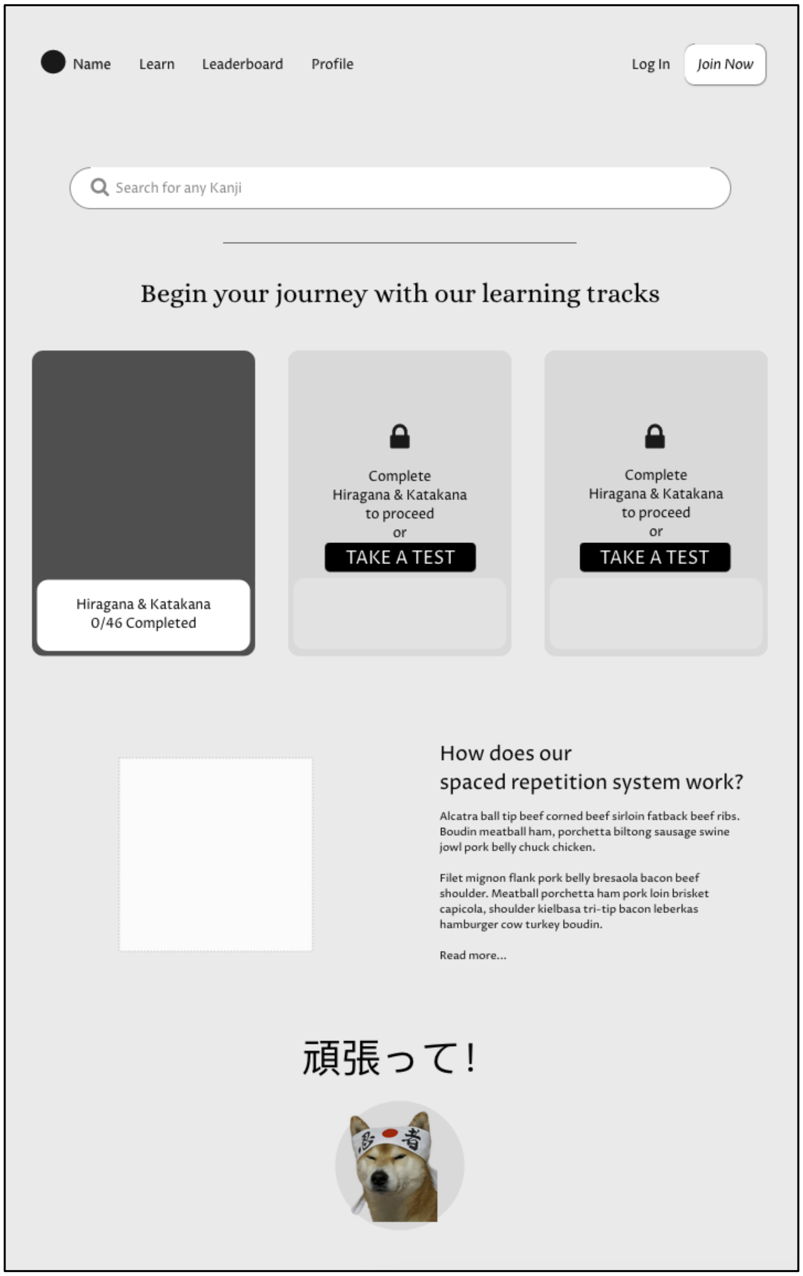


Figure 34 - Sitemap for app

Prioritizing the 4 golden rules of design; consistency and usability, comfortable and clear interactions, error prevention, don’t overwhelm users — the user interface was meticulously crafted to deliver a clean and seamless experience. [35]

3.4.1.1 The Landing Page



Begin your journey with our learning tracks

Hiragana & Katakana
0/46 Completed

Complete
Hiragana & Katakana
to proceed
or

Complete
Hiragana & Katakana
to proceed
or

How does our
spaced repetition system work?

Alcatra ball tip beef corned beef sirloin fatback beef ribs. Boudin meatball ham, porchetta biltong sausage swine jowl pork belly chuck chicken.

Filet mignon flank pork belly bresaola bacon beef shoulder. Meatball porchetta ham pork loin brisket capicola, shoulder kielbasa tri-tip bacon leberkas hamburger cow turkey boudin.

頑張って!

Figure 35 - App landing page

The landing page was designed with simplicity in mind, aiming to create an impactful and intuitive first impression for visitors and users alike. The core aspects of the website are put at the forefront with a strategic emphasis on the most essential feature: the pre-built decks and learning tracks. The navbar takes the user to every section of the website.

Following that is a short paragraph elaborating on how the spaced repetition system works with a link to read more about the SuperMemo2 algorithm in detail.

3.4.1.2 The Learn Page

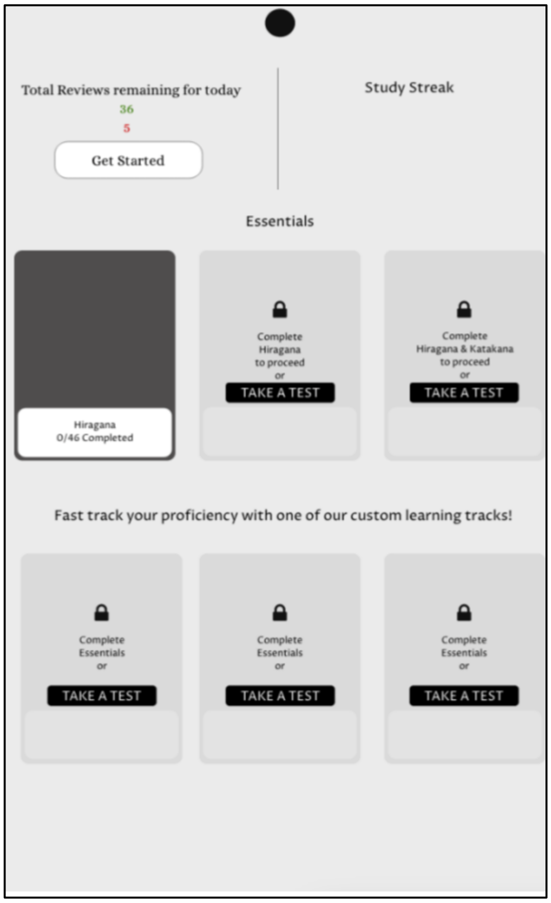


Figure 36 - App Learn page

The learn page details the learning tracks shown on the landing page along with any pending reviews that the user may have for the day. It is important to remember that this page should be inaccessible unless the user signs in. Through react, the component will be conditionally rendered and if the user is not signed in then they will automatically be redirected to the login/sign-up page.

On first opening the app, the user is prompted to learn Hiragana and Katakana before proceeding to anything else since all content on the website requires the user to learn Hiragana and Katakana. In case the user is already familiar with Hiragana and Katakana they can choose to take a quick test that allows them to unlock all content on the website.

3.4.1.3 The Leaderboard Page

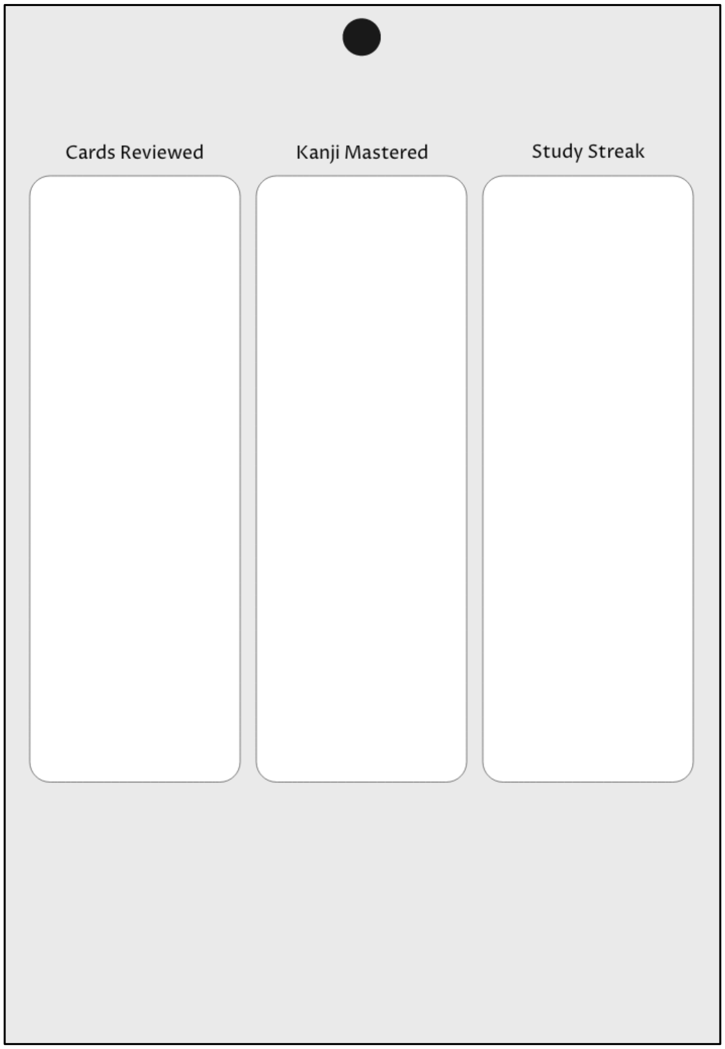


Figure 37 - App leaderboard page

We then come to the leaderboard page that allows the user to see where they stand compared to other users on the website. The leaderboard page is designed to add an aspect of gamification to the website coupled with the ‘achievement’ mechanic discussed in the pages below. The leaderboard page allows for 3 kinds of filters, total cards reviewed, kanji mastered and study streak — it is updated regardless of whether a user opens the leaderboard page as the React component renders it based on data available on the cloud database.

3.4.1.4 The Profile Page

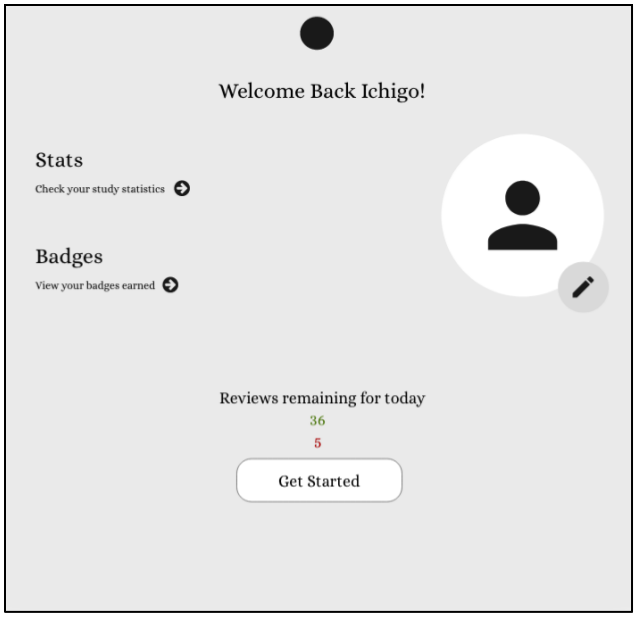


Figure 38 - App profile page

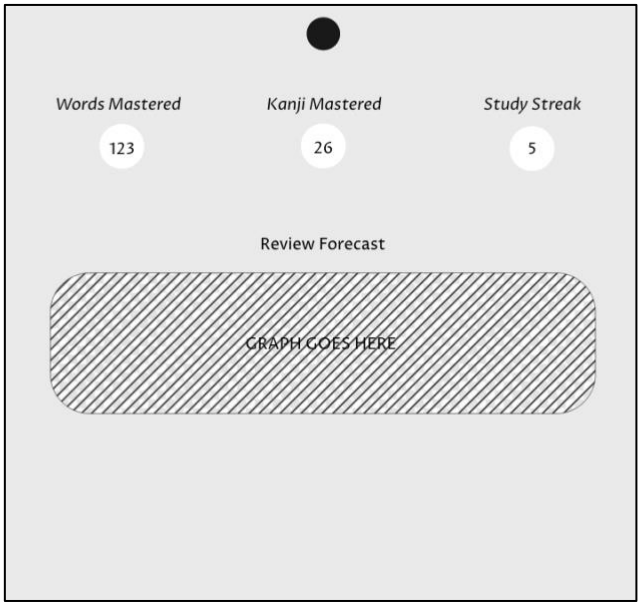


Figure 39 - App statistics page

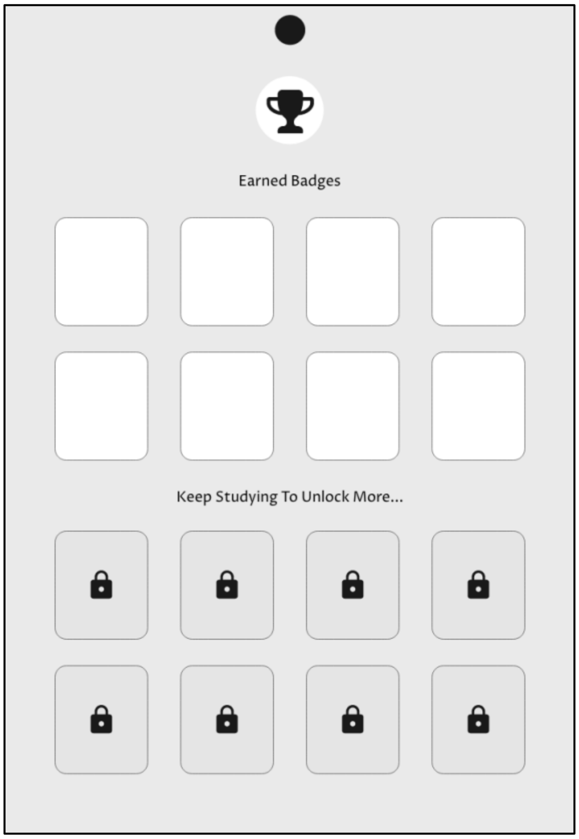


Figure 40 - App achievements/badges page

The profile page cleanly displays user-specific content including their profile picture, personal stats, and badges/achievements they have unlocked by being consistent with their studies. Research has shown that showing statistics and social gamification greatly increased grades and retention for learners. [36]

Visualizing personal statistics helps promote a sense of progress among learners.

3.4.2 Use-case Diagram, Class Diagram and Database Design

A use-case diagram is a powerful tool used in software development and system analysis to visualize and understand the functional requirements of a system from an end-user's perspective. It represents the interactions between actors (users or external systems) and the system itself, illustrating the specific functionalities the system provides and how users interact with it.

Use-case diagrams bridge the gap between functionality and the initial design ideas to facilitate an understanding of the system's scope and requirements. It helps in identifying all possible use cases and developing a robust and well-defined system. In doing so, the boundaries of the system are also laid out guiding the design process and ensuring that the system being designed fulfils its intended purpose.

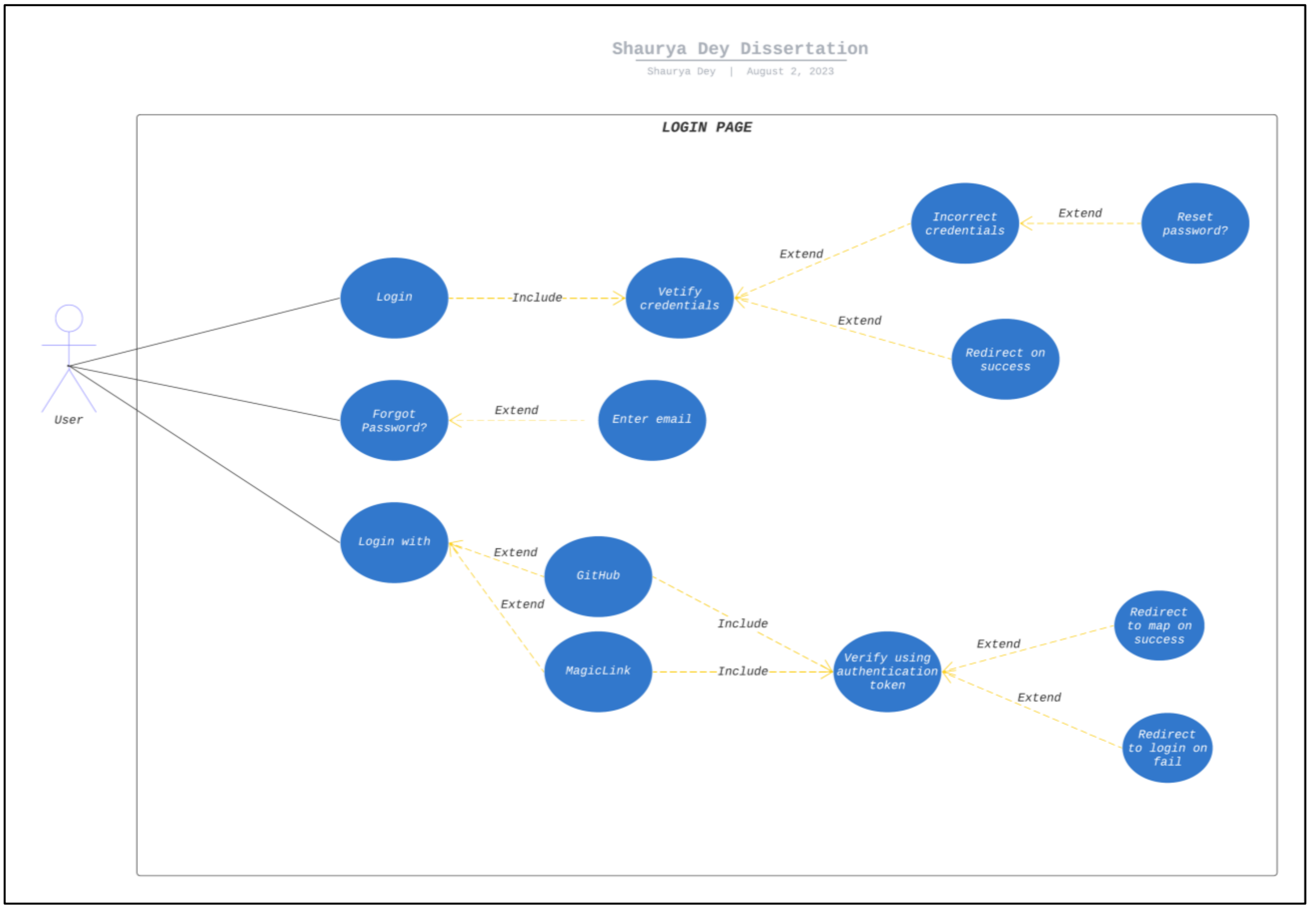


Figure 41 - Use-case diagram for the Login page

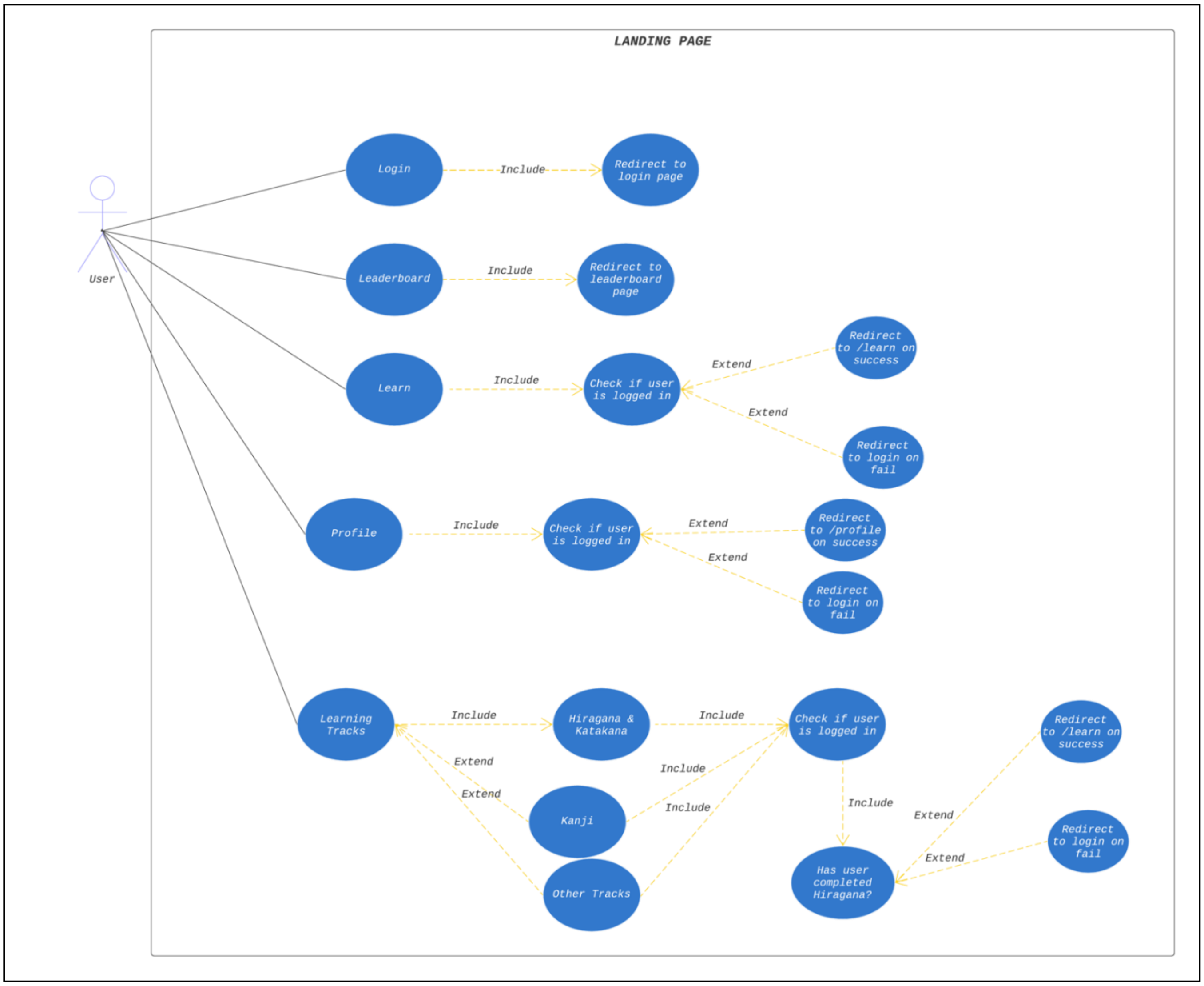

LANDING PAGE
User
Login
Include
Redirect to login page
Leaderboard
Include
Redirect to leaderboard page
Learn
Include
Check if user is logged in
Extend
Redirect to /learn on success
Extend
Redirect to login on fail
Profile
Include
Check if user is logged in
Extend
Redirect to /profile on success
Extend
Redirect to login on fail
Learning Tracks
Include
Hiragana & Katakana
Include
Check if user is logged in
Extend
Extend
Kanji
Include
Include
Other Tracks
Include
Has user completed Hiragana?
Extend
Redirect to /learn on success
Extend
Redirect to login on fail

Figure 42 - Use-case diagram for Landing page

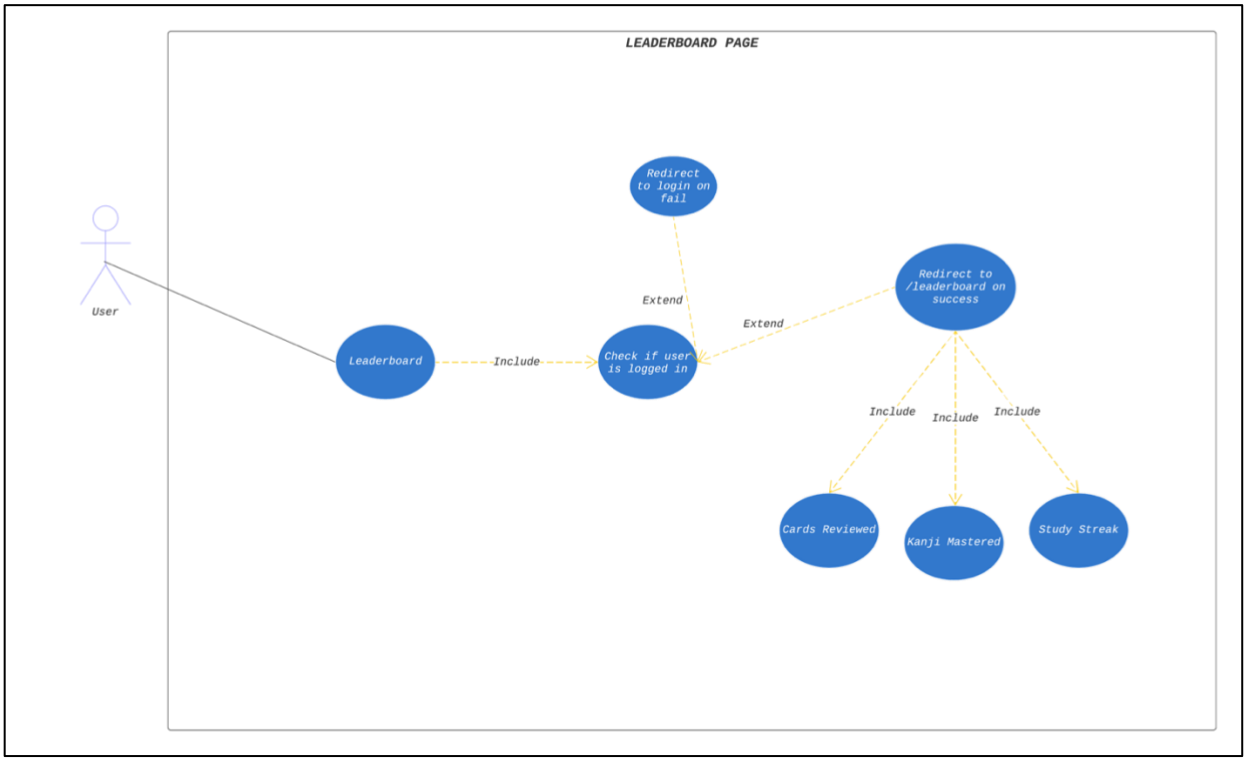


Figure 43 - Use-case diagram for the Leaderboard page

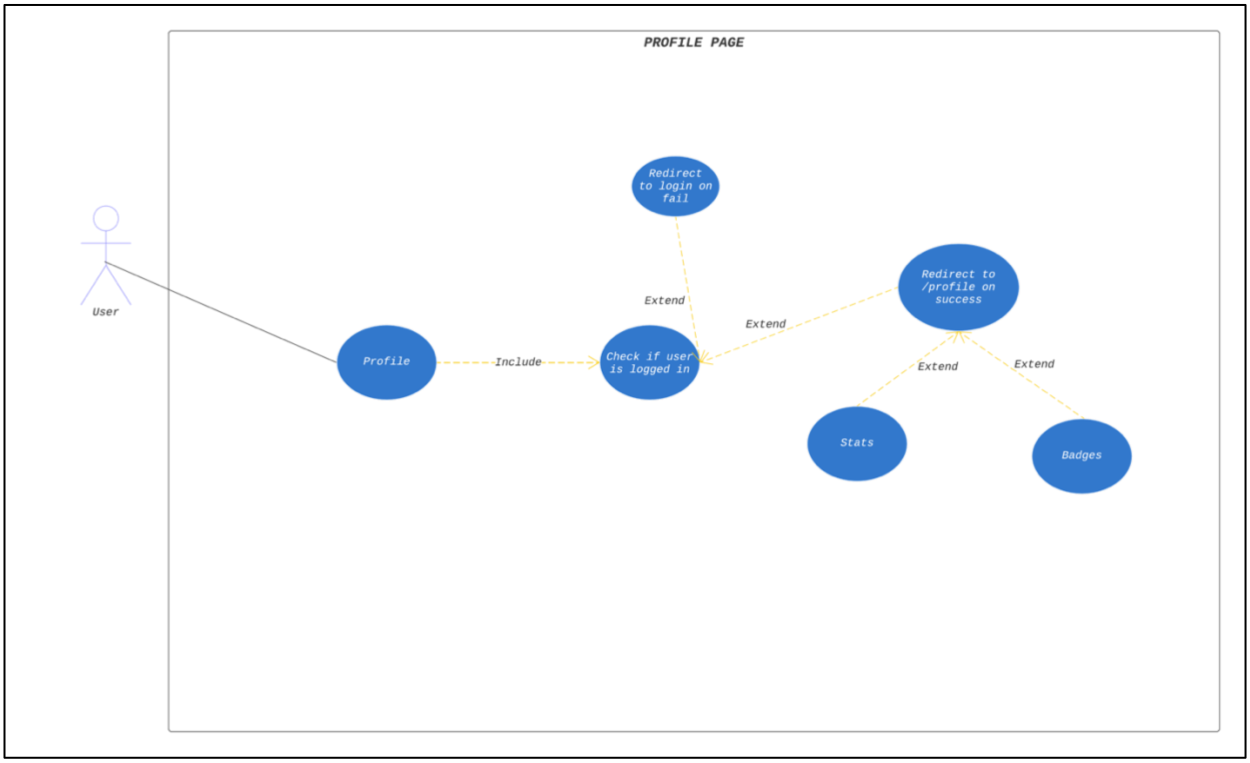

PROFILE PAGE
User
Redirect to login on fail
Extend
Redirect to /profile on success
Extend
Profile
Include
Check if user is logged in
Extend
Extend
Stats
Badges

Figure 44 - Use-case diagram for the Profile page

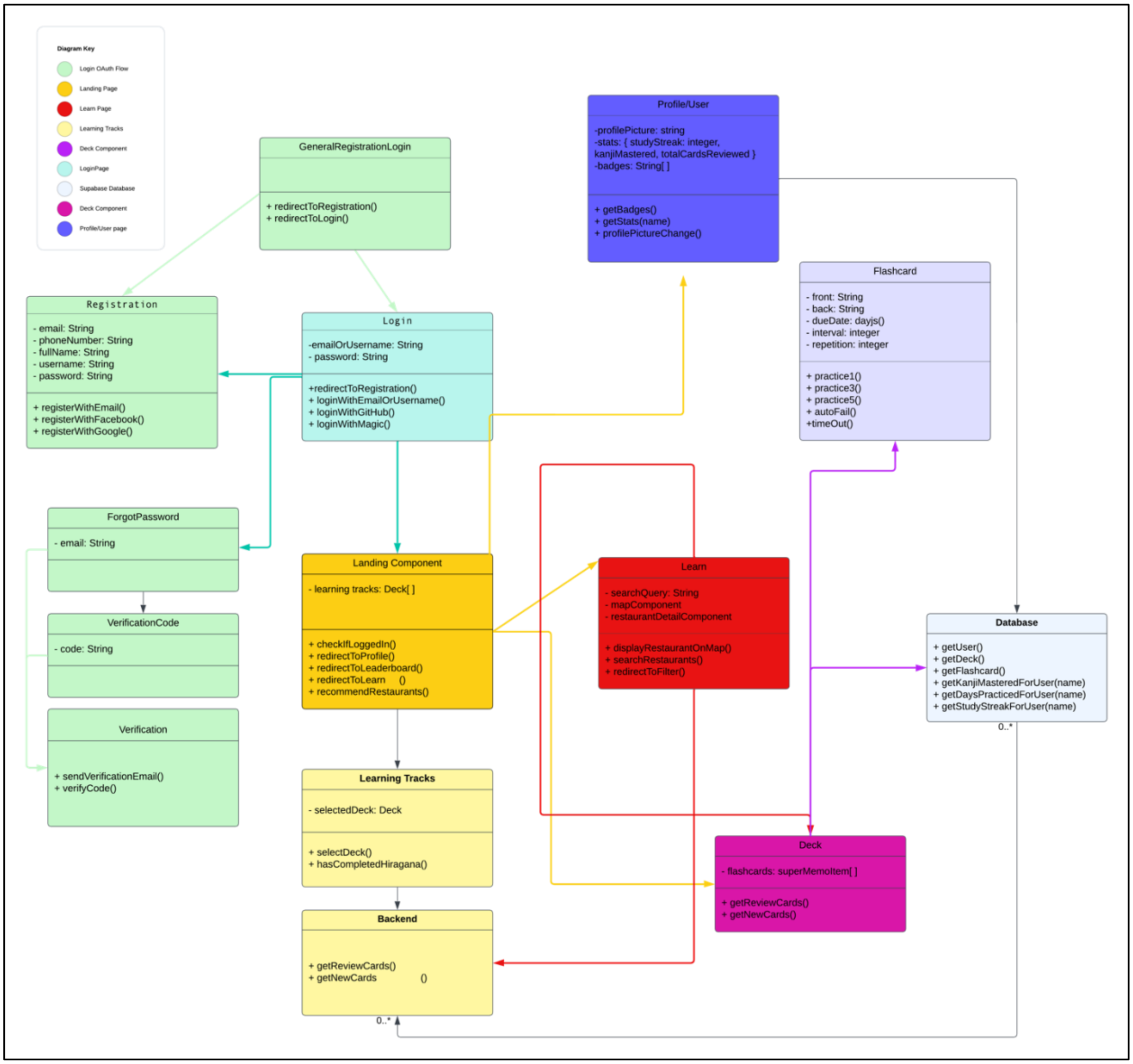


Figure 45 - Class diagram for the app

Another essential tool in modelling the application was creating a class diagram, used to represent a 'static structure' of a system. It illustrates in detail the classes/components with their variables and functions used therein.

A class diagram as ascribed above denotes the relation between the different parts of the web application making it incredibly useful in depicting how data flows within the application and facilitates design patterns. It helps identify areas where code can be reused, and modules repurposed to keep components under 100 lines long and easy to maintain in the long run. Each part of the application is split into functional components that contribute to ease during testing otherwise unachievable if the components are too long. This does not take away from the fact that the application is comprehensive and fully realizes the objectives that had been initially defined and set. In an agile environment, such maintainability is irreplaceable.

3.5 Implementation

The application's implementation is guided by the overarching aim to provide a comprehensive and engaging gamified language learning experience. Central to this implementation is the utilization of flashcards, which serve as foundational tools for fostering effective learning. Flashcards are structured to encompass both traditional word-to-meaning scenarios and sentence-based questions, creating a well-rounded approach to language acquisition. This dynamic approach is integral to enabling users to grasp language nuances comprehensively.

Moreover, the implementation includes the creation of various learning "tracks" designed to cater to users of different skill levels and objectives. These tracks are thoughtfully structured into distinct levels, closely aligned with the Japanese Language Proficiency Test (JLPT) levels. This strategic segmentation ensures that users progress through the learning journey systematically, gradually advancing their language proficiency.

A key aspect of the implementation revolves around the concept of study streaks, which play a pivotal role in user motivation and engagement. Study streaks are determined by the user's consistent engagement with the app, reflecting their commitment to learning. Users can accumulate study streaks by consistently completing learning sessions, and the mechanism for maintaining this streak is clearly defined within the app.

The Flashcard Review process, a cornerstone of the app's learning methodology, is carefully detailed. Users encounter various question types, including traditional flashcards and sentence-based queries. Feedback mechanisms are built-in, allowing users to assess their grasp of the content and categorize their responses as "easy," "medium," or "hard," tailoring the learning experience to their proficiency level.

The Profile page furnishes users with insights into their learning progress, highlighting mastered words, Kanji, and study streaks. Stats and badge pages contribute to a comprehensive view of accomplishments and achievements, encouraging consistent learning and achievement.

Furthermore, the Leaderboard page promotes a sense of healthy competition, motivating users to excel in their learning endeavours. This page ranks users based on metrics like total cards reviewed, Kanji mastered, and study streaks, fostering a dynamic and engaging community.

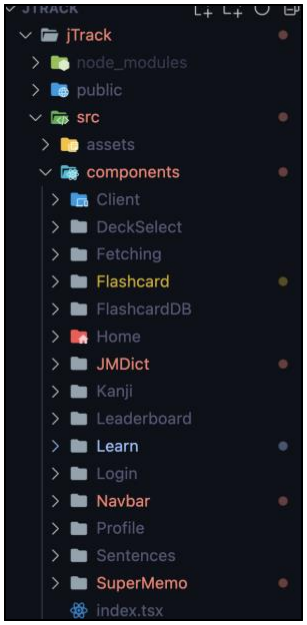
MASTERED CARD

The app aims to establish the baseline for when the user is expected to have mastered a card. It follows Anki’s definition of a mastered card as one that has its next review more than 21 days from the current date.

COMPONENT-BASED DEVELOPMENT

Components are a fundamental building block of the React library. In React, components are encapsulated, reusable units that allow us to structure the UI into smaller, manageable pieces. They promote modularity, reusability, and maintainability in the application's codebase.

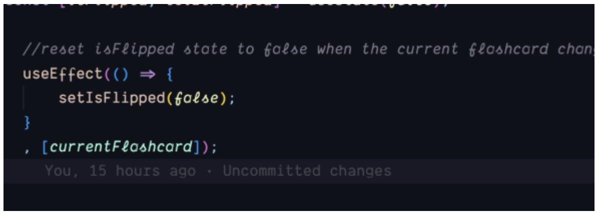
As can be seen below, the entire project was split up into several components. These components are functional components that return JSX (JavaScript XML) elements, which define the structure and content of a user interface.



```
JTRACK
jTrack
  node_modules
  public
  src
    assets
    components
      Client
      DeckSelect
      Fetching
      Flashcard
      FlashcardDB
      Home
      JMDict
      Kanji
      Leaderboard
      Learn
      Login
      Navbar
      Profile
      Sentences
      SuperMemo
      index.tsx
```

Figure 46 - Folder Structure

Unlike class components, functional components do not have access to traditional lifecycle methods (e.g., componentDidMount, componentDidUpdate). Instead, they use the useEffect hook to manage side effects and perform actions during different phases of the component's lifecycle.



```
//reset isFlipped state to false when the current flashcard chan
useEffect(() => {
    setIsFlipped(false);
}
, [currentFlashcard]);
  You, 15 hours ago · Uncommitted changes
```

Figure 47 - Use State

To understand how hooks work in React, we look at 'State'. In React, **State** is a JavaScript object that represents the current and *mutable data* of a component. It allows components to keep track of changing information and update the UI accordingly. State enables components to be dynamic and interactive, as it allows them to respond to user input, API responses, and other events. The `useEffect` hook is a fundamental feature in React that allows functional components to perform side effects in response to changes in the component's props, state, or the component's mounting and unmounting. Simply put, hooks and State allow React components to be very ‘Reactive’ as the name suggests, moulding and changing according to how the user interacts with the UI.

In the figure above, we can see a very good example of useEffect being used in the app. Whenever there is a change in ‘currentFlashcard’ or the current flashcard being studied, the ‘flipped’ status of the flashcard is reverted to false so that the answer is hidden for the new flashcard.

Functional components are typically more concise than class components, making them a preferred choice for building simple and modular UI elements.

3.4.1 Detailed High-level explanation of the user flow

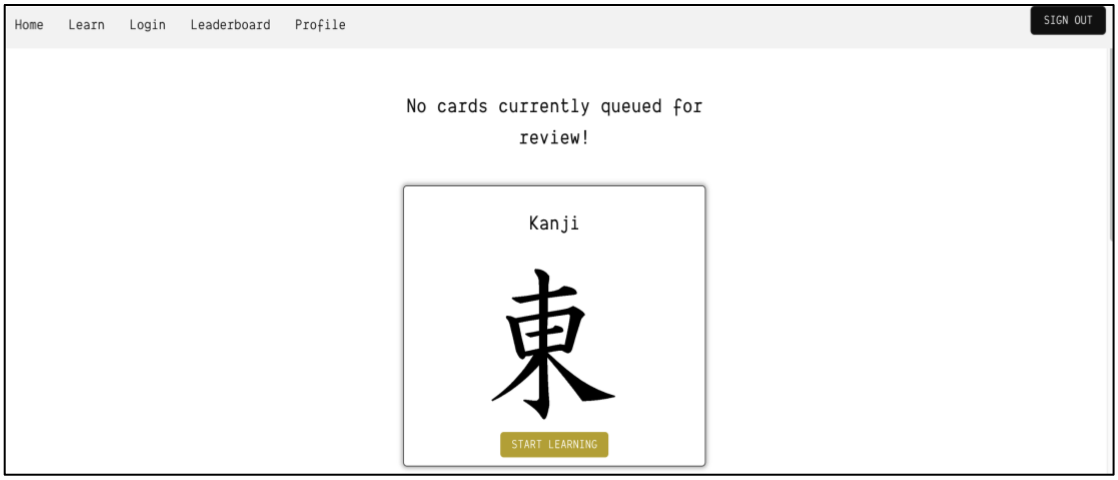
Described below is a high-level overview of how a user interacts with the language learning app.

1. **Landing Page:**
    - **a.** The user arrives at the landing page of the app.
    - **b.** On the first launch, they are presented with a welcome message and a brief overview of the app’s features. This includes pre-built decks and how the content is broken down, the importance of learning Hiragana and Katakana before tackling vocabulary. This also includes in slight detail the idea behind a spaced-repetition system and why it works better than traditional learning that involves cramming information only to quickly forget it.
    - **c.** On the landing page, the user is shown multiple learning ‘tracks’ that offer the user a structured path to follow. There are 3 different ‘levels’ prescribed that correspond to JLPT test levels.
    - **d.** Each ‘track’ which is a card component on-click redirects the user to the /learn route of the website where the pre-built decks reside.
    - **e.** The navbar at the top allows the user to navigate to leaderboards, learn or profile

Figure 48 - Navbar

2. **Learn Page:**
    - **a.** On initial render the app first checks if the user is logged in since it is required that the user logs in so that their progress may be saved.
    - **b.** If the user is not logged in, they are redirected to the login page where they are prompted to log in or sign-up. This guarantees that the progress is uploaded to the cloud database. The login page is detailed on point number 5 in the user flow.

c. After confirming the user has logged on and is in session, they are shown whether they have cards that need to be ‘reviewed’.



No cards currently queued for review!

Figure 49 - Deck Select – Learn Page

d. Reviewed cards mean the user has previously studied before and so they are prompted to review cards that are due either on or before that date. (It is important to note that it shows the total number of cards that are due for revision across all decks.)

e. Right below it is a section that maps all decks that have cards available for review and displays them at the top, prompting the user to ‘continue learning’.

f. When the user clicks on ‘start learning’ on a deck, they are directly taken to a page that lets them start reviewing and learning new cards based on their desired JLPT level.

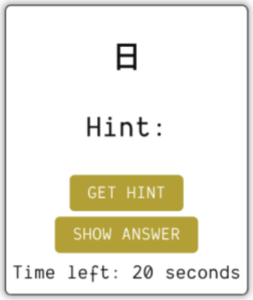
g. **Reviewing Flashcards**

   i. When the user clicks "Start Learning," they are taken to a page where they have a combined review and learning session.

   ii. The app displays the first all the flashcards that are due for review.

   iii. Each flashcard has a word or phrase and its translation or meaning on the other.

   iv. Two kinds of questions can come up – the first kind is where a word or phrase is shown on the front and the back reveals the meaning or answer — a **traditional flashcard**. Their recall ability comes in on the second attempt which is where the spaced repetition algorithm comes in. The flashcard is shown for a maximum of 60 seconds before it is auto-failed.



Time left: 20 seconds

Figure 50 - Normal Flashcard not flipped

v. There’s an option of getting a hint in the case of Kanji flashcards which shows a reading of the Kanji above the character.

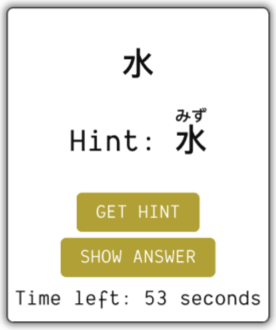
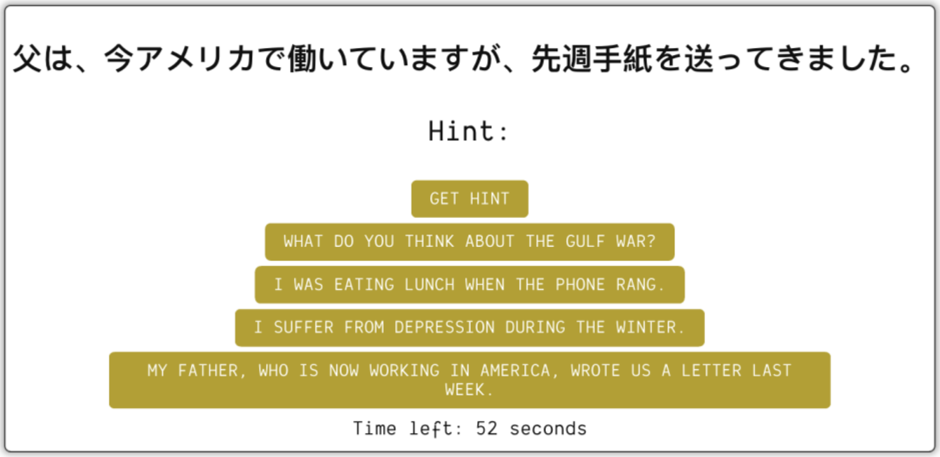


Figure 51 - Normal Flashcard with Hint

vi. The second kind of question is where a user is given a sentence with a hint and 4 options of which only 1 is the correct answer.



父は、今アメリカで働いていますが、先週手紙を送ってきました。

Hint:

WHAT DO YOU THINK ABOUT THE GULF WAR?

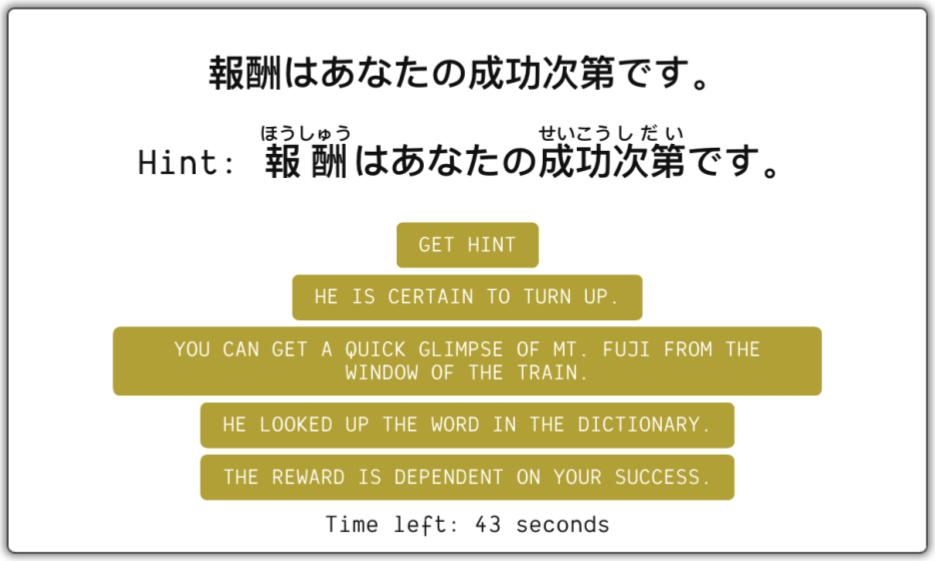
I WAS EATING LUNCH WHEN THE PHONE RANG.

I SUFFER FROM DEPRESSION DURING THE WINTER.

MY FATHER, WHO IS NOW WORKING IN AMERICA, WROTE US A LETTER LAST WEEK.

Figure 52 - Multiple-choice flashcard

vii. Optionally, the user is allowed a hint which makes the sentence a little easier but automatically caps the ease factor to ‘medium’ instead of ‘easy’



報酬はあなたの成功次第です。

ほうしゅう　せいこうしだい

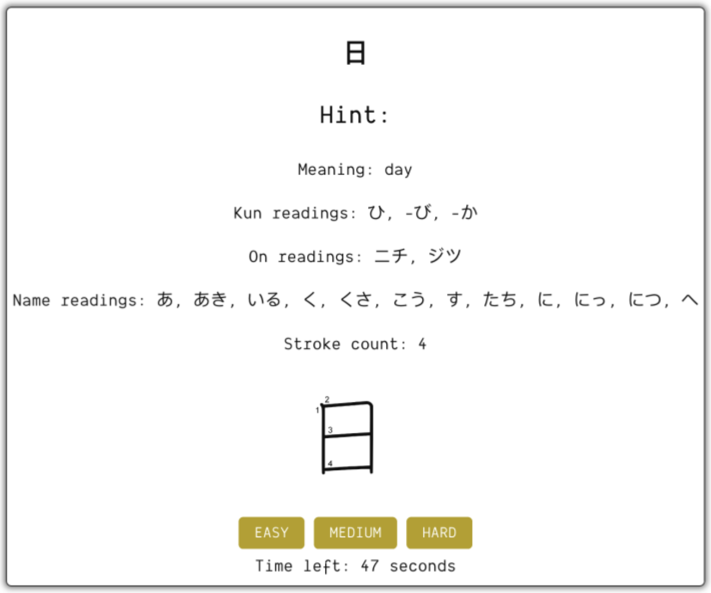
Hint: 報酬はあなたの成功次第です。

GET HINT
HE IS CERTAIN TO TURN UP.
YOU CAN GET A QUICK GLIMPSE OF MT. FUJI FROM THE WINDOW OF THE TRAIN.
HE LOOKED UP THE WORD IN THE DICTIONARY.
THE REWARD IS DEPENDENT ON YOUR SUCCESS.

Time left: 43 seconds

Figure 53 - Multiple-choice with a hint

viii. In the case of **flashcards**, the user is shown the word or phrase and checks if they can recall the meaning, then they can click “Show Answer” to show the answer and see if they were correct. It is then up to them to select Easy, Medium, or Hard depending on how much they struggled to answer.



日

Hint:

Meaning: day

Kun readings: ひ, -び, -か

On readings: ニチ, ジツ

Name readings: あ, あき, いる, く, くさ, こう, す, たち, に, にっ, につ, へ

Stroke count: 4

Time left: 47 seconds

Figure 54 - Normal flashcard flipped

ix. The reason for letting users have control over the ease factor is so that their progress for a flashcard isn't reset too far back if they 'struggle' to answer the questions. Anki tries to fix this by having 5 options but that is far too many for the user the gauge for themselves.

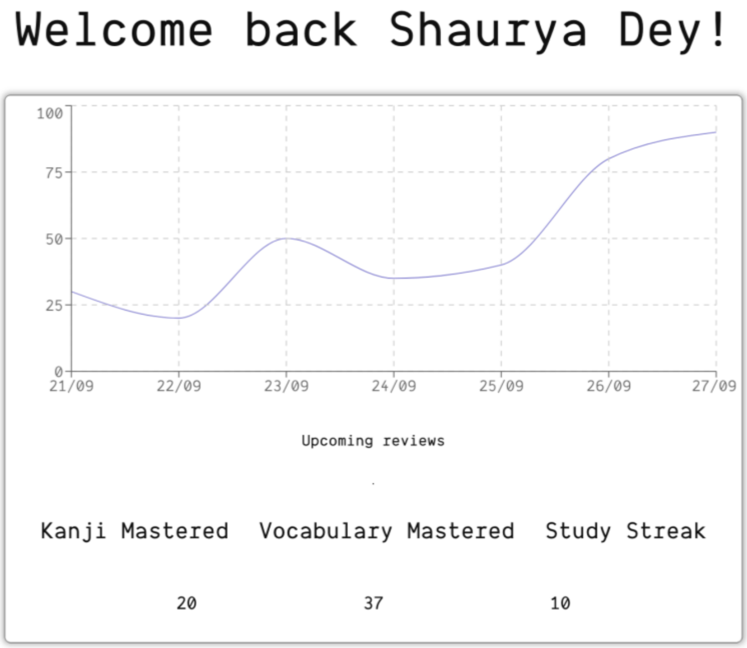
x. After completing the review queue, the user is shown a maximum of 10 new flashcards that they will learn on that day. This is true for both multiple-choice questions as well as Kanji, Hiragana and Katakana flashcards.

xi. The users are not shown new flashcards if they have more than 100 reviews due for the next time to ease the learning process.

3. **Profile Page:**

a. Like the learn page, the app first checks if the user is logged in and redirects them to the login page if they are not.

b. On initially loading the profile page, they are shown three different things, a stats component, a badges component and the option to view and change their profile data.



Welcome back Shaurya Dey!

Kanji Mastered	Vocabulary Mastered	Study Streak
20	37	10

Figure 55 - Profile page with stats

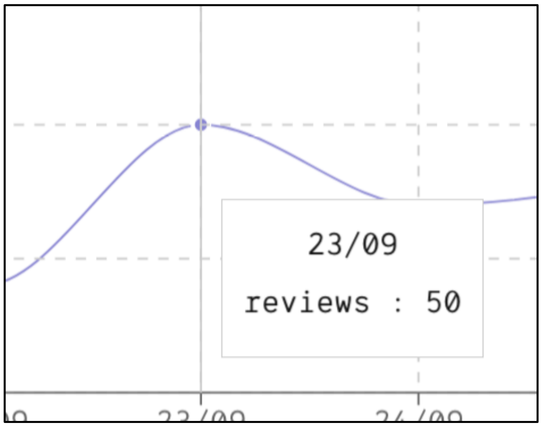
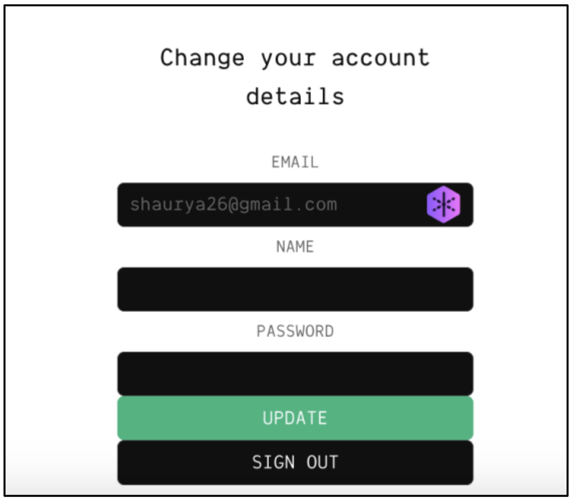


Figure 56 - Review count on hover



Change your account details

EMAIL

shaurya26@gmail.com

NAME

PASSWORD

Figure 57 - Change account details

c. The data for the user is fetched and stored in the database. This includes the card they must review, the total number of cards they have reviewed, the cards they have mastered, the total Kanji they have mastered, and their study streak.

d. **Stats Component**

   i. This component shows 4 things:

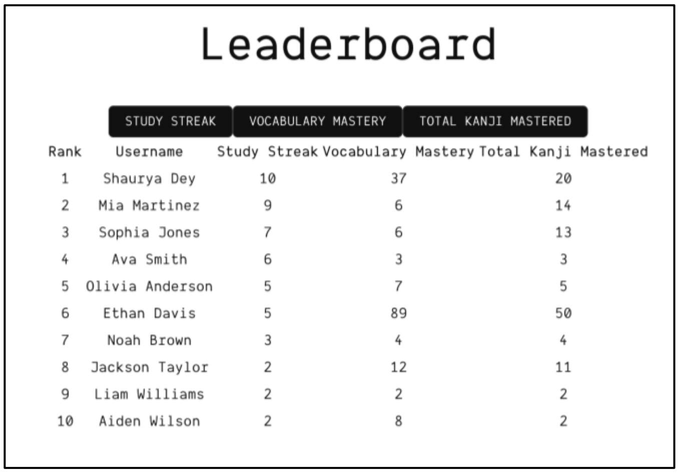
      1. The number of words mastered.
      2. The number of Kanji mastered.
      3. The user's study streak.
      4. The user's upcoming review forecast is in the form of a graph.

e. **Badges Component**

   i. This component shows which badges or achievements the user has unlocked. These achievements are unlocked by the user as they learn and practice consistently.

   - **ii.** It will show achievements the user has unlocked as well as one's they've unlocked along with a section that shows ones that they're very close to unlocking.

4. **Leaderboard Page:**
   - **a.** This page does not check if the user is logged in but uses user data in the backend. It sorts users based on 3 metrics:
     - **i.** Total Cards Reviewed
     - **ii.** Number of Kanji Mastered
     - iii. Study Streak



Leaderboard

Rank	Username	Study Streak	Vocabulary Mastery	Total Kanji Mastered
1	Shaurya Dey	10	37	20
2	Mia Martinez	9	6	14
3	Sophia Jones	7	6	13
4	Ava Smith	6	3	3
5	Olivia Anderson	5	7	5
6	Ethan Davis	5	89	50
7	Noah Brown	3	4	4
8	Jackson Taylor	2	12	11
9	Liam Williams	2	2	2
10	Aiden Wilson	2	8	2

Figure 58 - Leaderboard page with 3 different filters

5. **Login Page:**
   - **a.** The user is redirected to the login page if they are logged out or new and try to access /learn or /profile.

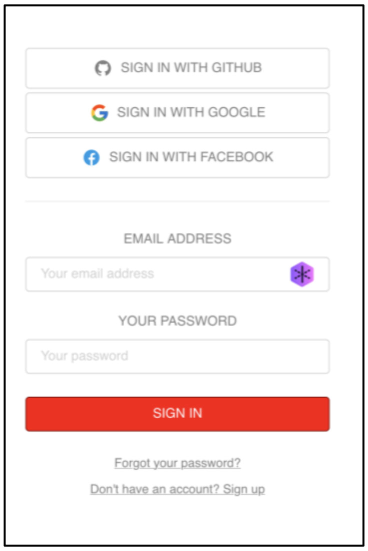


Figure 59 - Login Component in Login Page

b. The app offers 3 different methods to log in:

   i. GitHub
   ii. Magic Link
   iii. Email and Password

c. The login auth is handled by Supabase which handles the authentication for GitHub and one-time password for the Magic Link. In case the user does not have their credentials in the database, they are asked to sign up. The Magic Link option only works if the user already has previously logged in with their email ID and password.

3.4.2 Database

In the initial stages of development, a PostgreSQL database hosted on a NodeJS server was utilized to manage our data. This involved concurrently running the React application and fetching data using the NodeJS server..

While this setup provided a solid foundation for storing and retrieving data, the author recognized the need to enhance the application's functionality and security while limiting the number of components. Consequently, the author transitioned the database infrastructure to Supabase, a platform that seamlessly integrates PostgreSQL's robust data querying capabilities with the added advantage of built-in authentication features. This migration not only allowed for greater reliability but also maintainability. By leveraging Supabase's authentication system to streamline user management and bolster the overall security of the application. This strategic shift in technologies paved the way to create a more secure, efficient, and user-friendly experience for the application's users.

To setup a connection to Supabase, the following npm package was installed and saved to the project

```
npm install @supabase/supabase-js
```

Then a client was set up as described in the Supabase documentation.

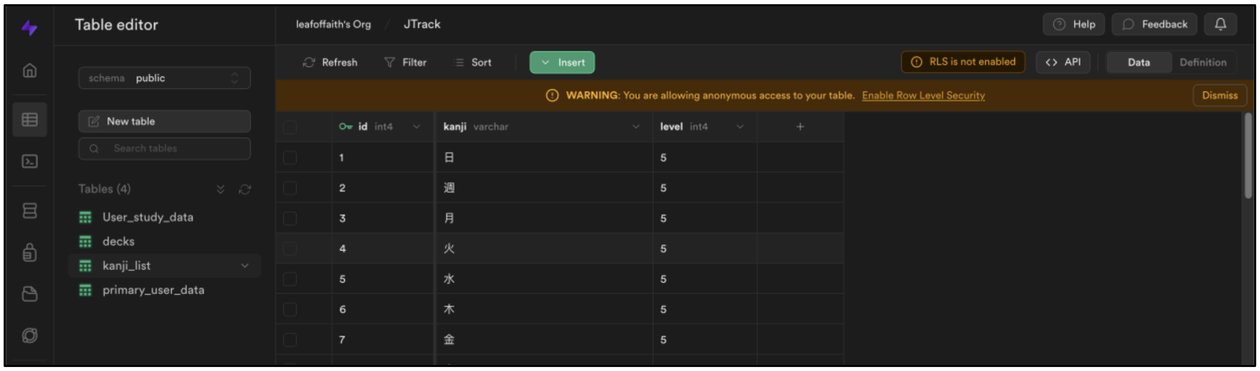
This client was exported so that it could be used in all other components that were making use of it, showcasing yet again the modularity of React.

Initial iterations of the app included making calls to the Postgres server running with Node in an exported module. However, it was later replaced with the same Supabase client that we saw earlier. This is an example of how the app was slowly migrated to stop using a backend server and make all calls using the front end of React.

STORING AND FETCHING KANJI AND SENTENCE DATA

While first creating the app, the Kanji data was stored and bundled as part of the app itself. This was not considered to be a problem as the number of Kanji per JLPT level wasn't that big. Regardless of the data structure it was stored in, it offered similar loading times in every case.

With the inclusion of Supabase, it was found that to separate the data from the codebase, it would be easier to store the Kanji in the database itself. The corresponding table in the Supabase is shown below.



WARNING: You are allowing anonymous access to your table. Enable Row Level Security

id int4	kanji varchar	level int4
1	日	5
2	週	5
3	月	5
4	火	5
5	水	5
6	木	5
7	金	5

Figure 60 - Supabase database

This is stored along with other user data that exist in separate tables which allows us to pull from the same Postgres database.

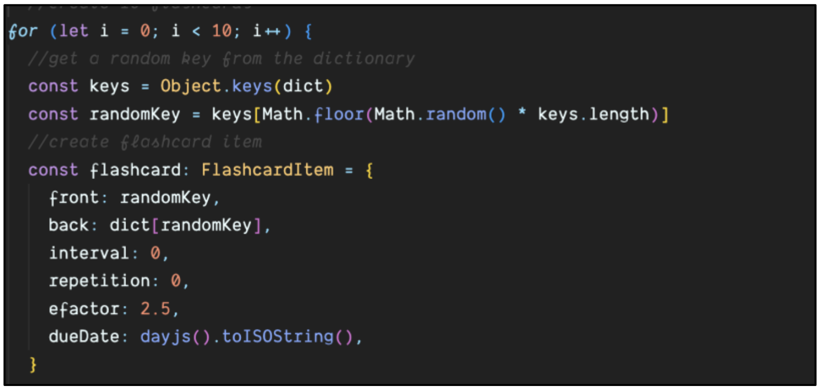
Things are different with the sentence data. As it was found, the best resource for sentence data was the Tatoeba corpus which allows for sentence pairs to be downloaded in a. tsv format. Sentences in English were translated into the target language – in this case, Japanese and stored in a .tsv file. Tatoeba is a collection of sentences and translations that is free to use and offers its sentences as part of a creative commons license. This .tsv file could be served to the application in one of three ways –

1. As a local file that can be fetched using a Node backend
2. By converting and importing it as a JSON (JavaScript Object Notation) file in React itself.
3. Copy the file onto the PostgreSQL database and insert the values therein.

The first alternative was not favourable as it required the need for a backend which the author worked to remove entirely. The second option was also not the most efficient, this is because the larger file size of JSON would mean a larger overhead and hence a slower app. The only other alternative was to copy the JSON onto the PostgreSQL database and insert the values therein.

While on paper it seemed that the third option was the quickest, due to inconsistent loading times while fetching from the cloud database, bundling the JSON along with the app proved to be the quickest and most consistent option.

The following code was used to create the sentence flashcards once read in. The flashcards were created at random from the database, and a maximum of 10 were created at a time.



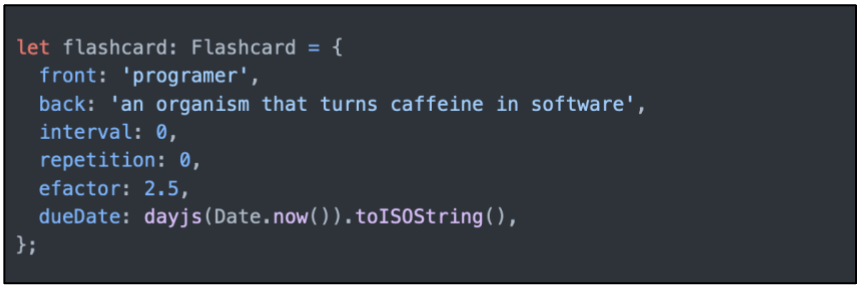
```
  //create 10 flashcards
for (let i = 0; i < 10; i++) {
  //get a random key from the dictionary
  const keys = Object.keys(dict)
  const randomKey = keys[Math.floor(Math.random() * keys.length)]
  //create flashcard item
  const flashcard: FlashcardItem = {
    front: randomKey,
    back: dict[randomKey],
    interval: 0,
    repetition: 0,
    efactor: 2.5,
    dueDate: dayjs().toISOString(),
  }
```

Figure 61 - Sentence flashcard creation

Once the Kanji characters were fed to the app, details about the Kanji including their meaning, reading level and stroke count were retrieved using an API called kanjiapi.
Kanjiapi is an API that provides kanji data from an extensive kanji dictionary using EDICT and KANJIDIC dictionary files. The data it provides through its endpoints is served in JSON format. The dictionary files used by the Kanjiapi website are property of the Electronic Dictionary Research and Development Group and use of them is in conformance with the Group's licence. [37]

3.4.3 Flashcards

Flashcards in the app are designed like flashcards in Anki albeit simpler, they retain the idea of having a 'front' and 'back' much like traditional flashcards. The user is shown the front of the flashcard and prompted to recall what the meaning is.



```
let flashcard: Flashcard = {
  front: 'programer',
  back: 'an organism that turns caffeine in software',
  interval: 0,
  repetition: 0,
  efactor: 2.5,
  dueDate: dayjs(Date.now()).toISOString(),
};
```

Figure 62 - Flashcard Item interface

A React flashcard component was created that was further subdivided into a front and a back. Flashcard information is passed through props when rendering a flashcard through the scheduler. This separation was made purely because of the separation of concerns and code modularity. It helps in keeping the code clean and allows for more flexible styling. Furthermore, developers that may work on this app in the future, or the author themselves will find it easier to read code that has been separated, commented on, and kept organized.

The flashcard component also includes 'useEffect' hooks that run a timer that auto-fails a card when the timer completes. The timer is set to and reset to 60 seconds whenever a new card is shown.

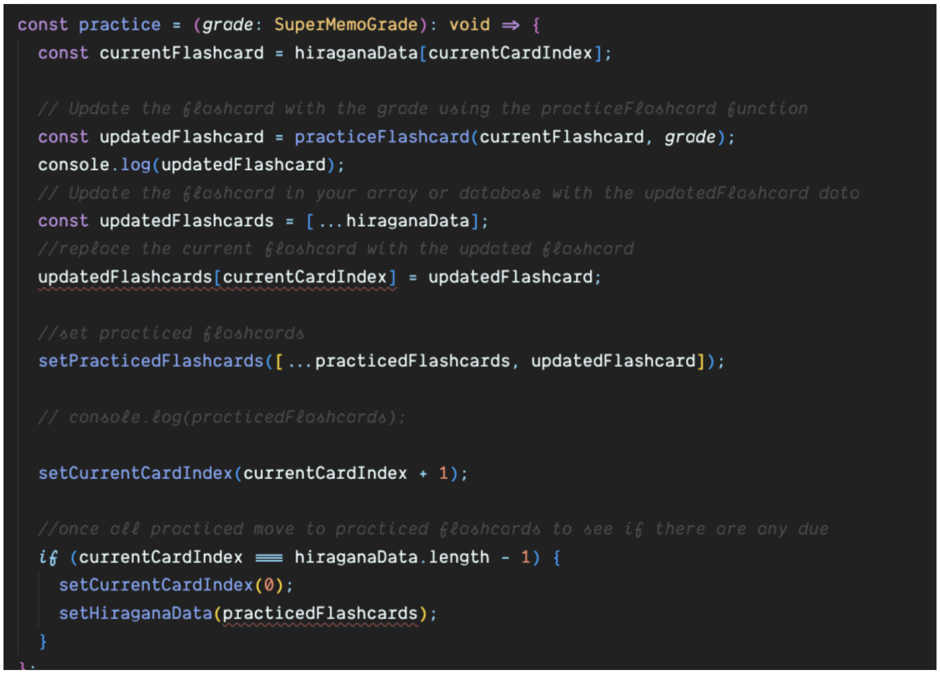
The ‘flashcard item’ is also defined in a separate file that ensures that every flashcard created conforms with the expected structure of a flashcard. This is achievable mainly through typescript. It extends the implementation of a SuperMemo flashcard item that has been defined in an imported npm (node package manager) package implementing the SuperMemo2 algorithm in typescript. [38]

3.4.4 Scheduler

The scheduler is where the logic of the application resides. The scheduler uses a TypeScript implementation of the Delphi implementation of the supermemo2 algorithm, which is the same algorithm that has been mentioned in section 2.5 of the report. The scheduler optimizes memory retention by scheduling review just before forgetting is likely to occur, gradually increasing the intervals as items become more ingrained in the user’s memory. The theory behind scheduling it according to the SuperMemo2 algorithm is strongly grounded in previously conducted research.

In the app, the Kanji scheduler, Sentence scheduler, Hiragana scheduler and Katakana scheduler handle the nuances of the slightly different kinds of flashcards. However, the main difference between the two is the way the information is fed to the scheduler. While the Kanji, Hiragana and Katakana pull data from the database, the Sentence scheduler pulls data from the JSON that is bundled with the app. To separate the two and potentially avoid any conflicts, it was found that keeping 2 separate components and states was better for the app.

The following function was defined to ‘practice’ a flashcard. It takes in a grade with which to practice, in this project’s case, there are 3 grades 1,3 & 5 where 1 is the easiest. To make sure the state persists, the flashcards are stored and set using the **setPracticedFlashcard** setter after practice.



```
const practice = (grade: SuperMemoGrade): void => {
  const currentFlashcard = hiraganaData[currentCardIndex];

  // Update the flashcard with the grade using the practiceFlashcard function
  const updatedFlashcard = practiceFlashcard(currentFlashcard, grade);
  console.log(updatedFlashcard);
  // Update the flashcard in your array or database with the updatedFlashcard data
  const updatedFlashcards = [...hiraganaData];
  //replace the current flashcard with the updated flashcard
  updatedFlashcards[currentCardIndex] = updatedFlashcard;

  //set practiced flashcards
  setPracticedFlashcards([...practicedFlashcards, updatedFlashcard]);

  // console.log(practicedFlashcards);

  setCurrentCardIndex(currentCardIndex + 1);

  //once all practiced move to practiced flashcards to see if there are any due
  if (currentCardIndex === hiraganaData.length - 1) {
    setCurrentCardIndex(0);
    setHiraganaData(practicedFlashcards);
  }
};
```

Figure 63 - Practice flashcard function

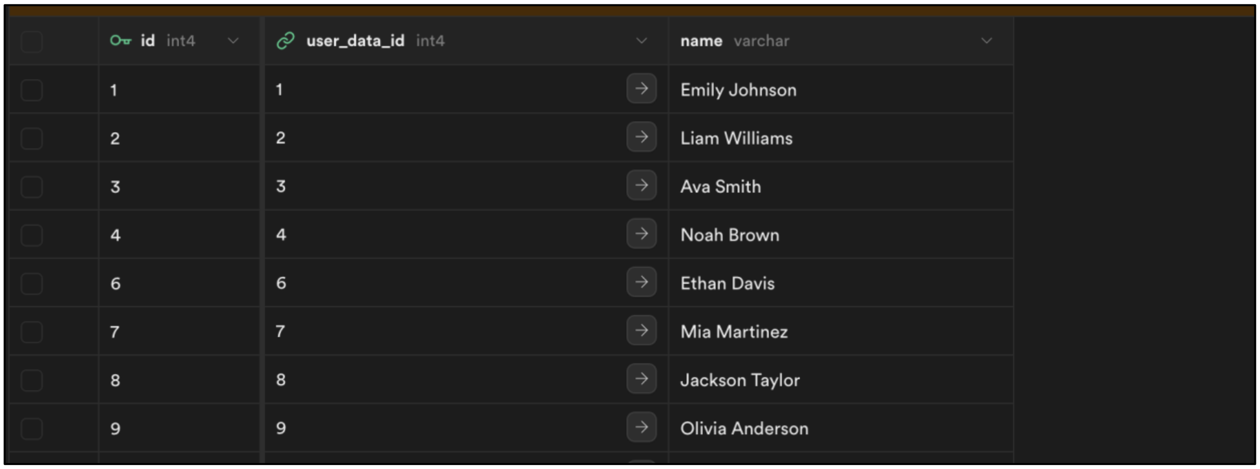
3.4.5 Leaderboard and Achievements

Incorporating gamified elements was one of the primary objectives of this application to build on the highly interactive nature of apps like Duolingo. These gamified features tap into our most basic desires for rewards and feedback. [39]

Incorporating a leaderboard within the app introduces an element of gamification that imbues users with a distinct sense of competition. By showcasing the rankings and scores of participants, the leaderboard cultivates a dynamic atmosphere where individuals are motivated to excel in comparison to their peers. This competitive inclination inspires users to surpass their boundaries, invest in increased dedication, and consistently interact with the app's materials.

As learners witness their positions ascend on the leaderboard, a strong aspiration to achieve higher ranks and gain acknowledgement, a constructive feedback loop is started, where the pursuit of excellence evolves into an intrinsically rewarding journey. The existence of the leaderboard transcends individual learning into a collective voyage for mastery, establishing a synergy between constructive rivalry and the refinement of skills.

In this application, the information for the leaderboard is pulled from Supabase's 'primary_user_data' table that provides three kinds of metrics – study streak, kanji mastered, and words/vocabulary mastered. The concept of a mastered flashcard is defined at the beginning of section 3.5



id int4	user_data_id int4	name varchar
1	1	Emily Johnson
2	2	Liam Williams
3	3	Ava Smith
4	4	Noah Brown
6	6	Ethan Davis
7	7	Mia Martinez
8	8	Jackson Taylor
9	9	Olivia Anderson

Figure 64 - User data in the database

3.6 Testing

The language-learning app is built from TypeScript-React components, which are composed of functions that return HTML. Most of these components use React's State to manage what data is being displayed. It was found that unit testing would be the ideal way to test out the functionality of the application.

Unit testing involves checking each component independently to ensure that they operate correctly and return the. tsx React components as expected. This involves testing buttons to make sure that they execute the correct functions or take users to the desired part of the application. TypeScript was very useful to ensure that flashcards, decks, and primary data structures were structured according to the interfaces defined by the authors.

The React Testing Library was a cornerstone tool for conducting comprehensive testing on a language learning application, ensuring its functionality, reliability, and user experience. It is recommended by the developers who created React and is widely used.

Tests were conducted to ensure the schedulers were correctly marking cards for review and setting the due dates as expected. Four TypeScript files entrusted with fetching sentence data, kanji data, hiragana data, and katakana data were subjected to rigorous examination as were the 4 scheduler components that displayed the flashcards and scheduled them accordingly. Whenever the scheduler 'practised' a flashcard, an updated flashcard was returned which made it easy to check if the flashcard was being correctly practised with the ease factor and due date.

This scrutiny validated their capacity to fetch precise and current data and the tests provided confirmation of the robustness of these data-fetching components.

Integral to the app's success was the meticulous structuring of sentence data into a dictionary format. The sentence data was downloaded from Tatoeba [40] as part of their Creative Commons license. The data was then read into a dictionary in the required format. Rigorous unit tests were carefully devised to evaluate the integrity and accessibility of this pivotal data structure. The dictionary was used to generate sentence flashcards and their corresponding multiple-choice questions.

Black box testing

Black box testing is a software testing methodology that focuses on evaluating the functionality of a software application without knowledge of the internal code. For this application, black box testing was deemed most appropriate after conducting the unit tests. The unit tests were conducted to make sure the code was free of errors and the functions working as expected. Black box testing would help eliminate remaining errors in the functionality that are visible.

This testing was conducted by the author alongside conducting unit tests.

Detailed below are the tests that were applied to the system

3.6.1 Login component

Table 4 - Login component test

Test Description	Expected Outcome	Actual Outcome	Pass/Fail
User Registration	The user should be able to create an account with a valid email and password	The user successfully registered with a valid email and password	Pass
Login	The user should be able to log in with valid credentials	The user was able to log in with valid credentials	Pass
Forgot Password	The user should be able to reset the password with a valid email	The user successfully resets the password a with valid email	Pass
Profile Update	The user should be able to update profile information	User successfully updated the profile information	Partial Pass
Profile and Learn Page Visibility if the user is not logged in	The profile and Learn page should not be visible unless the user is logged in	Profile and Learn page were not visible unless the user were logged in	Pass

Profile Update (Partial Pass) – The profile-updating function was buggy which didn't allow for users to occasionally not be able to update their email address and password. This had more to do with the service offered by Supabase rather than the function defined by the author. The author tried to debug the error but it didn't reliably pass the test.

3.6.1 Schedulers

Table 5 - Kanji scheduler test

Test Description	Expected Outcome	Actual Outcome	Pass/Fail
Kanji Scheduler displays flashcards	The Scheduler should be able to successfully display the flashcards	Scheduler successfully displayed the flashcards to be practiced with Kanji details and stroke order	Pass
Kanji Practice easy medium or hard buttons	The Scheduler should be able to successfully 'practice' a flashcard when easy medium or hard is pressed	The scheduler was able to successfully 'practice' a flashcard when easy medium or hard was pressed.	Pass
Kanji Due Date updated on practice	The Scheduler should be able to update the flashcards with the next due date	The Scheduler should be able to update the flashcards with the next due date	Pass
Kanji Hint by pressing on	The Scheduler should allow the user to get a furigana hint above the kanji.	The Scheduler allows the user to get a furigana hint.	Pass
Kanji Timer for a max of 60 seconds	The Scheduler should auto fail after 60 seconds	The Scheduler auto fails after 60 seconds	Pass

Table 6 - Hiragana scheduler test

Test Description	Expected Outcome	Actual Outcome	Pass/Fail
Hiragana Scheduler displays flashcards	The Scheduler should be able to successfully display the flashcards to be practiced	The Scheduler successfully displayed the flashcards to be practiced with the Hiragana scheduler	Pass
Hiragana Practice easy medium or hard buttons	The Scheduler should be able to successfully 'practice' a flashcard when easy medium or hard is pressed	The Scheduler was able to successfully 'practice' a flashcard when easy medium or hard was pressed.	Pass

Hiragana Due Date updated on practice	The Scheduler should be able to update the flashcards with the next due date	The Scheduler should be able to update the flashcards with the next due date	Pass
Hiragana Timer for a maximum of 60 seconds	The Scheduler should auto fail after 60 seconds	The Scheduler auto fails after 60 seconds	Pass

Table 7 - Katakana scheduler test

Test Description	Expected Outcome	Actual Outcome	Pass/Fail
Katakana Scheduler displays flashcards	The Scheduler should be able to successfully display the flashcards to be practiced	The Scheduler successfully displayed the flashcards to be practiced	Pass
Katakana Practice by pressing easy medium or hard buttons	The Scheduler should be able to successfully 'practice' a flashcard when easy medium or hard is pressed	The Scheduler was able to successfully 'practice' a flashcard when easy medium or hard was pressed.	Pass
Katakana Due Date updated on practice	The Scheduler should be able to update the flashcards with the next due date	The Scheduler should be able to update the flashcards with the next due date	Pass
Katakana Timer for a max of 60 seconds	The Scheduler should auto fail after 60 seconds	The Scheduler auto fails after 60 seconds	Pass

Table 8 - Sentence scheduler test

Test Description	Expected Outcome	Actual Outcome	Pass/Fail
Sentence Scheduler displays flashcards	The Scheduler should be able to successfully display the flashcards to be practiced	The Scheduler successfully displayed the flashcards to be practiced	Pass
Sentence Multiple Choice questions displayed by the scheduler	The Scheduler should be able to display 4 options out of which only one is the correct one.	The Scheduler correctly displays 4 options out of which only one was correct	Pass

Sentence Correct Choice on selecting the correct choice	The Scheduler should be able to successfully fail a card when a wrong option is selected and pass if the correct option is selected.	The Scheduler was able to successfully fail a card when a wrong option was selected and pass if the correct option was selected.	Pass
Sentence Due Date updated on practice	The Scheduler should be able to update the flashcards with the next due date	The Scheduler should be able to update the flashcards with the next due date	Pass
Sentence Timer for a max of 60 seconds	The Scheduler should auto-fail after 60 seconds	The Scheduler auto-fails after 60 seconds	Pass
Sentence Hint by pressing on Get Hint button	The Scheduler show allow the user to get a hint for the sentence using the button	The Scheduler allows the user to get a hint for the sentence using the button	Pass

3.6.2 Profile Page

Table 9 - Stats component test

Test Description	**Expected Outcome**	**Actual Outcome**	**Pass/Fail**
Stats component displayed on page load	The stats component within the profile page should be visible showing the current user's stats and review forecast	The stats component within the profile page correctly shows the current user's stats	Pass
Change user data by filling out the form	The profile page should allow the user to change their email address and password	The profile page allows the user to change their email address and password	Pass
Achievements displayed on page load	The achievements component should show the current achievements that the user has	The achievements component shows the current achievements the user has	Pass

Summarizing the tests

In summary, the tests carried out as part of the black box tests showed that the app was functioning as expected as far as the core functionality was concerned. The navigation between all the pages handled using React was functioning as expected.

4.0 Results and Evaluation

This section discussed the results of the project and to what extent the requirements were met. The developer was able to meet the core functional requirements that were set at the beginning of the project thus ensuring the app's functioning was as close as it could be to the applications it was inspired by and building on.

4.1 Meeting functional requirements

Table 10 - Functional requirements review

	Functional Requirements	**Priority**	**Pass/Fail**
R1	**Content is to be displayed in the form of flashcards.** All content in the application will be shown to the learner in the form of flashcards	**P1**	**Pass**
R2	Each flashcard will be shown to the user for a **maximum of 60 seconds** after which the flashcard will be automatically marked as failed.	**P1**	**Pass**
R3	**Spaced Repetition System** The app should incorporate a spaced repetition system which optimizes learning by presenting flashcards at intervals based on the user's learning progress and retention rate.	**P1**	**Pass**
R4	**Gamified Elements** The application should incorporate gamification techniques to enhance the language learning journey. This includes integrating elements such as rewards, achievements, and progress tracking to simulate a game-like experience.	**P1**	**Pass**
R5	**Achievement Recognition** Users' accomplishments, such as completing a set of lessons or reaching certain skill levels, should be acknowledged through achievements. These achievements can serve as milestones and markers of progress in their language-learning journey.	**P2**	**Pass**
R6	**Reward System** The app should implement a reward system where users receive virtual rewards, badges, or tokens upon completing lessons, achieving specific goals, or demonstrating proficiency.	**P3**	**Fail**
R7	**Progress Tracking** Learners should have access to detailed progress reports and analytics, showcasing their strengths and areas for improvement. This feature will allow learners to monitor their language proficiency and set goals accordingly.	**P2**	**Partial Pass**
R8	The app should include **multiple-choice sentence** questions that promote user input and engagement	**P2**	**Pass**
R9	The application should have **built-in decks** for different JLPT levels	**P1**	**Pass**
R10	**Login** A user should be allowed to log in with their email and password, a magic link or GitHub.	**P1**	**Pass**
R11	If a user is not signed in, they are prompted to go to sign-up.	**P1**	**Pass**
R12	**Review** When a user starts learning a new deck, cards that need to be reviewed in that deck are shown first.	**P1**	**Pass**
R13	The app should not allow more than **100 review cards for next time.** This limit should be achieved by stopping the user from learning new cards once they hit 100 review cards due	**P2**	**Partial pass**

R14	The **leaderboard** should contain **3 filters**, study streak, kanji mastered, and words mastered	**P1**	**Pass**
R15	The **leaderboard** should be **globally accessible**	**P1**	**Pass**
R16	The profile page should display all user stats and how many cards they have mastered	**P2**	**Pass**
R17	The profile page should display all essential profile information including email ID, password	**P2**	**Pass**

R6 – A detailed Rewards system was not implemented as the author prioritized other functionalities given the time constraint. This meant that the user will not fully experience a Duolingo-like gamified UI as the author had initially intended.

The Rewards system would have comprised virtual currency along with more incentive for the user to come back to the application.

R7 – The progress tracking function doesn't show areas that the learner is weak in. This would have been completed if the author/developer had more time. The progress tracking feature would have also included in percentage, the cards remaining in the decks.

R13 – This was tested in code by hard-coding it and running a unit test to check it, but it could not be checked in practice as that is not possible given the duration of this project.

4.2 Meeting non-functional requirements

Table 11 - Non-functional requirements review

	Non-functional Requirements	**Priority**	**Pass/Fail**
NF1	**Response time** The application should have fast response times and smooth navigation. There should be minimal delay when transitioning from one part of the application to another.	**P1**	**Pass**
NF2	**Usability T**he user interface should adhere to Nielson's usability guidelines so that the app is user-friendly.	**P2**	**Pass**
NF3	**UI Guidelines** The app should refer to Material Design guidelines while styling components and the UI to ensure adherence to readability and accessibility	**P2**	**Partial Pass**
NF4	The app should be designed to handle multiple users.	**P2**	**Pass**
NF5	WCAG (Web Content Accessibility Guidelines) should be adhered to	**P2**	**Pass**
NF6	User progress should be successfully backed up on the cloud and data loss should be prevented	**P2**	**Pass**
NF7	The app should work seamlessly across all kinds of devices if they have a web browser. Thus, ensuring a consistent experience for all users.	**P2**	**Pass**
NF8	The app should adapt to different screen sizes, resolutions, and viewports, providing an optimal user experience on various devices including smartphones, tablets, and desktops.	**P1**	**Partial Pass**

NF3 – Material design guidelines were advised but not followed. There was not enough time for the author to adhere to specific style guides and it was not prioritized over other core features.

NF8 – The app works on Firefox, Edge & Chrome, in all viewports. It was not tested for Edge or any other browser.

Overall, the application follows a minimal yet aesthetic design. The focus of the project was not on user experience, but the application isn't barebones either. It also details how the app works in the About section of the home page. So, the user knows how to make use of the spaced repetition system.

4.3 Evaluating the system as a tool to learn Japanese

Being a language learner who has previously used language learning apps to supplement their Japanese learning, the author was attuned to the intricacies of the language and how the content needed to be structured to ease the learning process.

Furthermore, the author's own learning journey contributes to their discernment of potential pitfalls or shortcomings in language learning platforms. They could identify issues such as the lack of gamification in Anki or the lack of a reliable spaced repetition system in Duolingo.

There are several things the application does well while there are a handful of things that are missing. The application can schedule cards and optimize the learning frequency using the spaced repetition system built into the application. The application also structures the content based on JLPT levels as originally intended. JLPT is the most widely recognized Japanese proficiency test, so the learners will be reassured of the fact that they are learning content that will be relevant to them should they decide to prepare for the test.

The application also does well to show statistics for the users which makes their progress tangible in a way. By showing study streaks, words and kanji learned, the learner can measure their achievements. These stats are carried forward and displayed on the leaderboard which is visible to everyone. The leaderboard allows users to compete with other users for the top spot in the respective filters, encouraging them to learn more. Since new flashcards are only shown up to a maximum of 100, users can't cram and are instead encouraged to be consistent with their studies and take it slow. Users are greeted with an easy-to-use interface and at no point are they shown more information that they know what to do with. The desktop version also has custom CSS based on mouse hover which gives the user feedback and helps with the overall user experience. The experience on phones is also smooth and consistent.

Recognition over recall

Flashcards in the app look and function the same way as it does in Anki, focusing on recognition over recall. This is especially true for the multiple-choice questions while learning sentences which helps in easier recognition of tougher sentences.

Limitations

The efficacy of the app could not be tested due to time constraints. If given more time the author could have asked users to use the app for over a month to confirm the functioning of the spaced repetition system and receive user feedback on the gamified elements. Although background research has shown that gamification usually corresponds to a more engaging application, it was not in the scope of this project.

Despite the restrictions, the results of testing and evaluating the app have been reassuringly positive.

Summary of Testing

Most of the functional and non-functional requirements were met leaving room only for the parts that could be completed if there was more time. The functionality of the app works and is as it was intended

5.0 Conclusion

In this project, the author built a gamified language learning web application that uses a spaced repetition system, specifically the SuperMemo2 algorithm and for the Japanese language. Its mode of learning is based on flashcards that are based on different JLPT levels. According to the results, the app will allow language learners to pick up the Japanese language by following a structured path using its built-in decks.

Firstly, background research was conducted to understand language learning, gamification, and the nuances of Japanese. It helped develop the basis for the project. In addition, it helped in understanding what made the already successful applications so good or easy to use, and what the shortcomings were.

The research helped ascertain the requirements that the final system should meet. The requirements were drafted based on the background research. Next, the author started working on the design, implementation, and testing. The approach used for the application development was Agile. The project was built primarily using React and TypeScript. The implementation process took up the most time, as the author spent time trying to optimize content for the website while learning how TypeScript works.

The author spent time trying to find an optimal solution for the backend of the website before landing on Supabase to handle authentication as well as querying Hiragana, Katakana and Kanji. The pre-built sentence decks had to be bundled with the app itself. The sentences and their translations were retrieved from Tatoeba that gave the author an easy to use. json format to work with.

Then time was spent on testing the application to make sure the basic functionality of the application was working as intended. Although unit tests were conducted iteratively throughout the entire process, a final 100%-line coverage test was attempted using the built-in React testing library which made sure the flashcard, the leaderboard and achievements were working as intended.

Criticisms

As mentioned earlier, the requirements were met for the application but there were some parts that could have been worked on more. A slightly more accessible design might have been possible. The app is only supported on the most popular browser, had there been more time, the app could have been optimized for less popular options like OperaGX, Safari and so on.

It is to be noted that instead of focusing too much on the actual content, it would have been more beneficial for the author to focus on the functionality to reduce the time taken to implement the application. Given the time constraints, it was unrealistic to expect to create content for an application. The primary focus of the project was to create a system for language learners first. Although Japanese was selected for this project due to the author's familiarity with it, in practice the language learning system can be used to learn any language. So, in hindsight, excessive focus on the content resulted in more time being taken for the implementation portion than expected.

Although the spaced repetition system works as intended, writing a more efficient algorithm might have been possible if the author focused less on the content of the website.

Extended Implementation

If given more time, the application could have been worked on a lot more. There is room for improvement in the application both in terms of functionality as well as user experience.

- **Advanced Progress Tracking**: Detailed insights into their strengths and areas needing improvement could be provided, helping users tailor their learning strategies. Leeches are not included in the app, which in a future implementation could be shown in the statistics dashboard
- **Audio**: A big component of researched language learning apps is that they include audio which was not implemented in this application. It was attempted but there were performance issues that could be fixed in time for it to be presentable.
- **Offline access**: The app is clearly an online one without support for offline functionality. A native application could be made that mitigates this by providing and packaging content for offline functionality.
- **User-added decks:** User-added decks were not shipped with the application as it currently stands. Anki allows users to add their own decks which helps them personalize their content and learning further, something that is missing in this implementation. Perhaps integration with Anki decks could be explored.
- **Advanced Rewards:** Rewards like virtual currency were not implemented and can be worked on in the future. An idea for a RPG (Role-playing game) like gamification was considered but ultimately scrapped in favor of more essential features.

The current version of the language learning app provides valuable tools for language acquisition, a longer development timeline would have been ideal to implement a few or all the proposed extended features. These enhancements would have contributed to a more immersive, adaptive engaging experience

Summary

Language learning, according to the author, is a very essential skill that allows for cultural immersion. The language learning website offers an excellent platform for new learners to start with Japanese by offering structured learning paths with gamified elements that keep the user interested in the material. The application can serve as a good tool for existing learners to brush up on their existing knowledge as well. The spaced repetition system optimizes learning and gives the users a sense of progress that will serve to motivate them further to learn not just the Japanese language but other languages as well.

REFERENCES

[1] "Language Learning App Revenue and Usage Statistics (2023)," *Business of Apps*. https://www.businessofapps.com/data/language-learning-app-market/ (accessed Aug. 09, 2023).

[2] M. M. R. P. Ltd, "Online Language Learning Market Worth $31.81 Billion by 2029 - Exclusive Report by Meticulous Research®," *GlobeNewswire News Room*, Nov. 16, 2022. https://www.globenewswire.com/en/news-release/2022/11/16/2557157/0/en/Online-Language-

Learning-Market-Worth-31-81-Billion-by-2029-Exclusive-Report-by-Meticulous-Research.html (accessed Aug. 09, 2023).

[3] A. E. Seibert Hanson and C. M. Brown, “Enhancing L2 learning through a mobile assisted spaced-repetition tool: an effective but bitter pill?,” *Comput. Assist. Lang. Learn.*, vol. 33, no. 1–2, pp. 133–155, Jan. 2020, doi: 10.1080/09588221.2018.1552975.

[4] Babbel.com and L. N. GmbH, “An Introduction To The Japanese Language,” *Babbel Magazine*. https://www.babbel.com/en/magazine/guide-to-japanese-language (accessed Aug. 09, 2023).

[5] “Advantages of JLPT | JLPT Japanese-Language Proficiency Test - https://www.jlpt.jp/e/about/merit.html.” https://www.jlpt.jp/e/about/merit.html (accessed Aug. 09, 2023).

[6] S. Tamm, “Spaced Repetition: A Guide to the Technique,” *E-Student*, Jan. 10, 2023. https://e-student.org/spaced-repetition/ (accessed Aug. 09, 2023).

[7] “Hermann Ebbinghaus: The First Psychologist to Study Learning and Memory - Edublox Online Tutor,” Jan. 07, 2023. https://www.edubloxtutor.com/hermann-ebbinghaus/ (accessed Aug. 09, 2023).

[8] “Forgetting curve,” *https://en.wikipedia.org/wiki/Forgetting_curve*. Jul. 30, 2023. Accessed: Aug. 09, 2023. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Forgetting_curve&oldid=1167963953

[9] G. R. Loftus, “Evaluating forgetting curves,” *J. Exp. Psychol. Learn. Mem. Cogn.*, vol. 11, no. 2, pp. 397–406, 1985, doi: 10.1037/0278-7393.11.2.397.

[10] H. Pashler, D. Rohrer, N. J. Cepeda, and S. K. Carpenter, “Enhancing learning and retarding forgetting: choices and consequences,” *Psychon. Bull. Rev.*, vol. 14, no. 2, pp. 187–193, Apr. 2007, doi: 10.3758/bf03194050.

[11] SDG&E, “SDG&E Celebrates San Diego’s Biggest Energy Savers.” https://www.prnewswire.com/news-releases/sdge-celebrates-san-diegos-biggest-energy-savers-136748638.html (accessed Aug. 09, 2023).

[12] “Stack Overflow - Where Developers Learn, Share, & Build Careers - https://stackoverflow.com/,” *Stack Overflow*. https://stackoverflow.com/ (accessed Aug. 09, 2023).

[13] “Level Up! Gamification,” *Adobe Slate*. https://express.adobe.com/page/dHrG4/ (accessed Aug. 09, 2023).

[14] S. Hamiyet, “The effects of computer games on the achievement of basic mathematical skills,” *Educ. Res. Rev.*, vol. 10, no. 22, pp. 2846–2853, Nov. 2015, doi: 10.5897/ERR2015.2172.

[15] “Examining the Use of Gamification in Language Learning Apps and Its Effect on Second Language Acquisition | Second Language Acquisition - https://blogs.ntu.edu.sg/hss-second-language-acquisition/wiki/chapter-10/#Introduction.” (accessed Aug. 09, 2023).

[16] C. Wright, “Amon, K., & Campbell, A. (2008). Can children with AD/HD learn relaxation and breathing techniques through biofeedback video games? Australian Journal of Educational & Developmental Psychology, Vol. 8, pp. 72-84.,” 2008.

[17] K. Walsh, “8 Research Findings Supporting the Benefits of Gamification in Education,” *Emerging Education Technologies*, Dec. 05, 2012. https://www.emergingedtech.com/2012/12/8-research-findings-supporting-the-benefits-of-gamification-in-education/ (accessed Aug. 09, 2023).

[18] B. Kim, “Chapter 4. Gamification in Education and Libraries,” *Libr. Technol. Rep.*, vol. 51, no. 2, Art. no. 2, Mar. 2015.

[19] E. N. Ariza and S. Hancock, “Second Language Acquisition Theories as a Framework for Creating Distance Learning Courses,” *Int. Rev. Res. Open Distrib. Learn.*, vol. 4, no. 2, Oct. 2003, doi: 10.19173/irrodl.v4i2.142.

[20] A. Lambers and A. J. Talia, “Spaced Repetition Learning as a Tool for Orthopedic Surgical Education: A Prospective Cohort Study on a Training Examination,” *J. Surg. Educ.*, vol. 78, no. 1, pp. 134–139, Jan. 2021, doi: 10.1016/j.jsurg.2020.07.002.

[21] J. J. Donovan and D. J. Radosevich, “A meta-analytic review of the distribution of practice effect: Now you see it, now you don’t.,” in *Journal of Applied Psychology*, Oct. 1999, pp. 795–805. doi: 10.1037/0021-9010.84.5.795.

[22] C. Nickson, “Learning by Spaced Repetition,” *Life in the Fast Lane • LITFL*, Jul. 01, 2019. https://litfl.com/learning-by-spaced-repetition/ (accessed Aug. 09, 2023).

[23] “What spaced repetition algorithm does Anki use? - Frequently Asked Questions - https://faqs.ankiweb.net/what-spaced-repetition-algorithm.html.” (accessed Aug. 09, 2023).

[24] “Cloze Procedure | Cloze Exercise | Teaching Wiki,” *Twinkl*. https://www.twinkl.co.uk/teaching-wiki/cloze-procedure (accessed Aug. 09, 2023).

[25] K. L. Jensen and C. Elbro, “Clozing in on reading comprehension: a deep cloze test of global inference making,” *Read. Writ.*, vol. 35, no. 5, pp. 1221–1237, May 2022, doi: 10.1007/s11145-021-10230-w.

[26] “Repetition Works! How To Learn a New Language with Spaced Repetition,” *Lingvist*. https://lingvist.com/blog/spaced-repetition-in-learning/ (accessed Aug. 09, 2023).

[27] B. Settles and B. Meeder, “A Trainable Spaced Repetition Model for Language Learning,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Berlin, Germany: Association for Computational Linguistics, 2016, pp. 1848–1858. doi: 10.18653/v1/P16-1174.

[28] F. M. Vilamu, “Duolingo – optimization of the Spaced Repetition System to improve long-term memorization”.

[29] “SuperMemo 2 algorithm: source code - https://www.super-memory.com/english/ol/sm2source.htm.” (accessed Aug. 09, 2023).

[30] “Agile vs. Waterfall Methodology – Forbes Advisor - https://www.forbes.com/advisor/business/agile-vs-waterfall-methodology/.” (accessed Aug. 09, 2023).

[31] “React.” https://react.dev/ (accessed Aug. 23, 2023).

[32] “JavaScript With Syntax For Types.” https://www.typescriptlang.org/ (accessed Aug. 09, 2023).

[33] “Supabase | The Open Source Firebase Alternative,” *Supabase*. https://supabase.com/ (accessed Aug. 23, 2023).

[34] “Tailwind CSS - Rapidly build modern websites without ever leaving your HTML.,” Nov. 15, 2020. https://tailwindcss.com/ (accessed Aug. 23, 2023).

[35] “4 Golden Rules of UI Design Every Designer Should Know | Springboard,” *Springboard Blog*, May 12, 2021. https://www.springboard.com/blog/design/golden-rules-of-ui-design/ (accessed Aug. 02, 2023).

[36] M. Krause, M. Mogalle, H. Pohl, and J. J. Williams, “A Playful Game Changer: Fostering Student Retention in Online Education with Social Gamification,” in *Proceedings of the Second (2015) ACM Conference on Learning @ Scale*, Vancouver BC Canada: ACM, Mar. 2015, pp. 95–102. doi: 10.1145/2724660.2724665.

[37] “ELECTRONIC DICTIONARY RESEARCH AND DEVELOPMENT GROUP - http://www.edrdg.org/edrdg/licence.html.” http://www.edrdg.org/edrdg/licence.html (accessed Aug. 23, 2023).

[38] V. Dinh, “SuperMemo.” Aug. 22, 2023. Accessed: Aug. 23, 2023. [Online]. Available: https://github.com/Maxvien/supermemo

[39] J. F. Vermeir, M. J. White, D. Johnson, G. Crombez, and D. M. L. Van Ryckeghem, “The Effects of Gamification on Computerized Cognitive Training: Systematic Review and Meta-Analysis,” *JMIR Serious Games*, vol. 8, no. 3, p. e18644, Aug. 2020, doi: 10.2196/18644.

[40] “Tatoeba: Collection of sentences and translations - https://tatoeba.org/en/.” https://tatoeba.org/en/ (accessed Aug. 23, 2023).