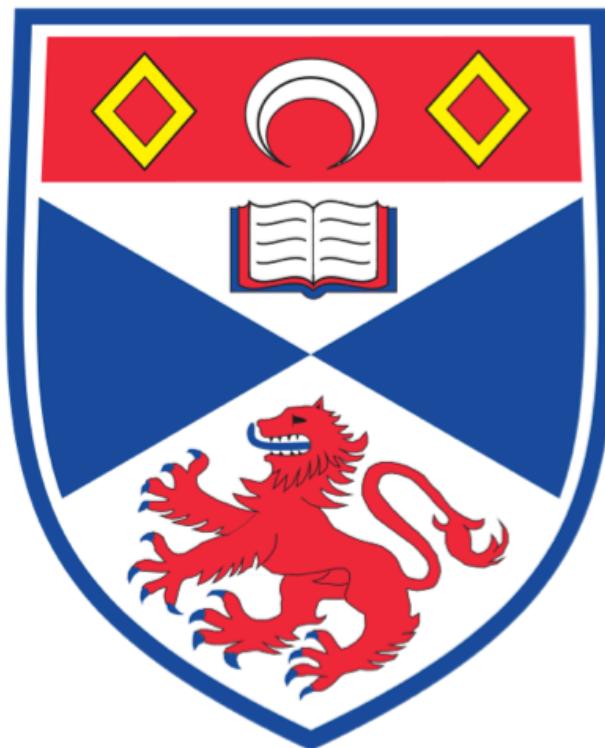


LANGUAGE LEARNING APP FOR INTERMEDIATE LEARNERS

Yue Ning Law

210002926



Supervisor: Angela Miguel

School of Computer Science

University of St Andrews

22nd March 2024

DEDICATION

To my grandma, mum and dad.

ABSTRACT

This project presents the design, development and evaluation of a full-stack language application curated for intermediate Chinese learners.

Research was conducted through surveys and interviews with language learners and teachers to identify the most important aspects of language learning. The final result contains an app that uses natural language processing to segment video transcripts from YouTube into vocabulary words for students to study. Students are able to watch YouTube videos in real-time with the transcripts and their translations, create multimedia flashcards enhanced with images from Google, personal notes, audio pronunciations and stroke order animations, and play short, 5-minute games to re-inforce learning based on a spaced-repetition algorithm. These games include exercises such as matching words to images, testing speech pronunciation, translating sentences, fill-in-the-blank sentences and stroke order practice.

Lastly, the app was evaluated by 37 students which indicated that the app was engaging, fun and effective, with 24.3% of students rating the app a 10/10.

DECLARATION

I declare that the material submitted for assessment is my own work except where credit is explicitly given to others by citation or acknowledgement. This work was performed during the current academic year except where otherwise stated. The main text of this project report is 21004 words long, including project specification and plan. In submitting this project report to the University of St Andrews, I give permission for it to be made available for use in accordance with the regulations of the University Library. I also give permission for the title and abstract to be published and for copies of the report to be made and supplied at cost to any bona fide library or research worker, and to be made available on the World Wide Web. I retain the copyright in this work.

CONTENTS

1. Introduction	6
2. Software engineering process	7
3. Ethics	9
4. Context survey	10
4.1. Analogical thinking	11
4.2. Context based learning and self-evaluation	12
4.3. Mobile learning	13
4.4. The current language apps	14
4.5. Utilising social media	16
4.6. Summary	16
5. Design	18
6. Requirements specification	31
7. Implementation (Backend)	33
7.1. Endpoints	36
7.1.1. Processing a video	36
7.1.2. Obtain a video	36
7.1.3. Add a study date	37
7.1.4. Get study streak	37
7.1.5. Get video library	37
7.1.6. Get cards to review today	37
7.1.7. Update the spaced repetition (SRS) system	37
7.1.8. Creating flashcards	38
7.1.9. Updating flashcards	38
7.2. Worker threads	39
7.2.1. Video processing	39
7.2.2. Obtaining YouTube transcripts	40
7.3. Architecture & Hosting	41
8. Implementation (Frontend)	43
8.1. Flutter	43
8.2. Frontend architecture	44
8.3. Frontend features	45
8.4. MVP	47
9. Evaluation	61
9.1. Student feedback	61
9.2. Evaluation against objectives	69
9.3. Critical appraisal	70
10. Conclusion	72
11. Appendix	73
11.1. Tests	73
11.2. User manual	77
11.3. Other	77
References	82

1. INTRODUCTION

Many existing language apps today cater to beginners, leaving intermediate learners struggling to find effective and personalised resources. This project aims to identify the most important aspects of language learning and create a full-stack application that leverages YouTube content, flashcards, multimedia, spaced-repetition and game-based learning to aid intermediate learners. Personal investigations such as surveys and interviews will be conducted to complement existing research.

Currently, I am on the journey of studying Mandarin Chinese, but I am struggling to find resources to aid my learning. More specifically, there is a large amount of beginner-friendly content, but not enough for intermediate learners. The content topics are generic and are not personalised to my interests. Through asking other language learners, I have found that many of them face the same issue.

I therefore propose an app solution based on language learning research that seeks to integrate best language learning practices for intermediate learners of Mandarin Chinese using YouTube. Users can create flashcards from new vocabulary found from these YouTube videos and through a spaced repetition (SRS) system, be able to review these words through review questions provided by the app. Additionally, research will encompass my own personal investigations through survey and interviews, research papers and analysis on the success of language apps such as Duolingo.

The objectives of this project are stated as follows:

Objective	Objective type
Create a minimal viable product of a language learning app	Primary
Transcripts can be generated from the app	Primary
Flashcards can be created from the transcripts	Primary
App can generate review questions relevant to the user	Secondary
A user evaluation form sent out to obtain user feedback at the end of the project	Secondary

2. SOFTWARE ENGINEERING PROCESS

For this project, Agile was used as the software engineering methodology.

Agile allows for the continuous delivery of valuable software. As the end-goal is to create a full-stack application with iterative feedback from interviews throughout the building process, Agile is useful for producing many working iterations of a software solution quickly and also involving a wide range of stakeholders at all phases of the development process [1].

This means that on an ongoing basis, a product can be tested, examined and adjusted, rather than building a single product at the very end. Agile consists of the following practices (some were removed as it presumed the existence of teams instead of a solo project) [2]:

1. Daily meetings (short meetings should be introduced to keep everyone up to date with progress)
2. Demo (at the end of an iteration, a working product should be demonstrated to other stakeholders)
3. Iteration planning (for example a sprint backlog, to break down requirements to smaller work items, and planning of what features should be included in coming releases)
4. Iterative and incremental development (development is done in sprints. In each sprint, increments are added to a working piece of software)
5. Retrospectives (a way to reflect on what went well each sprint)
6. Task board (where progresses of tasks are visualised)

Since this project is to be done individually, the only meetings required to achieve (1) are weekly meetings with my supervisor. In this meeting, I will discuss my progress as well as setbacks I have faced that week. This will also achieve (5), where I reflect on each weekly sprint.

To achieve (2), after the development of a minimal-viable-product (MVP), interviews will be conducted per week to obtain user feedback. After the feedback, the MVP will be re-iterated over in the next sprint.

For (3), Notion, a note-taking app, will be used to track and plan new features brought up by interviewees. Each larger feature will be broken down into atomic levels so that each task is more manageable.

(4) will be achieved through weekly sprints, where the Notion page is reviewed and upcoming features planned for the sprint are tracked.

Finally, (6) is will be achieved through a progress log on Notion that will track the work done every time there is a new feature or bug fixed.

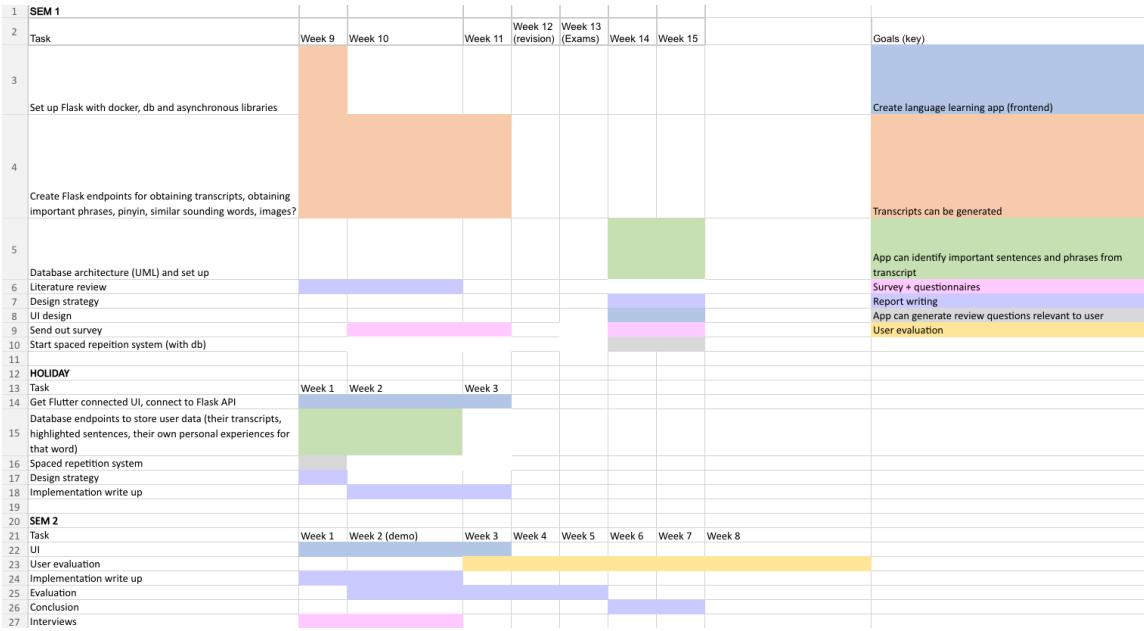


FIGURE 1. Initial project plan

The initial project plan seen in a GANTT chart (see Figure 1). As shown, there is a lot of leeway to allow for project flexibility. This is because the app is planned to evolve depending on user feedback during the building process. As you will find discussed below in the design and implementation sections, the project has diverted from the initial plan and has extended to contain many more features than what was initially expected.

The reason why weeks 1-8 were blank were because these weeks were used to understand the problem space and get familiar with Flask and Flutter.

You can read the progress log in Appendix Figure 31.

3. ETHICS

Since this project involves the use of YouTube content it is important to be aware of YouTube's developer policies.

Regarding YouTube's developer policies [3] it is important to consider both ethical and legal aspects, including copyright issues. I will ensure that:

1. Content queried from YouTube is used only for language learning purposes, and the app will not misuse the content in any way.
2. Content queried from YouTube will be explicitly stated, as the video will be embedded into the app.
3. To protect the rights of content creators, the app will clearly credit the original creators of the video.
4. Videos will stop being shown if a content creator does not want their video shown on the app.
5. My app will transform the copyrighted material by incorporating it into a language learning context.
6. My app will not allow videos to be downloaded or be temporarily stored. The video will be transformed into transcripts designed just for language learning.
7. My app will only use publicly available content through the YouTube API.

This project will also require surveys and interviews from participants. Surveys will be conducted on Qualtrics and only accessible to students within the University of St Andrews. Every participant involved will be given a consent form to fill out as well as a Participant Information sheet, which will describe how their data will be used and for what purposes. This data will be anonymised and deleted after the project submission date.

The signed ethical approval document can also be seen in the Appendix (Figure 32).

4. CONTEXT SURVEY

It has been argued that languages did not necessarily evolve from speech, but from the innate human instinct of communication. At first, humans did not have words, but they expressed themselves with body gestures and hand movements. As they innovated, creating fire and inventing tools, only then did they begin to communicate with their mouths, and thus along came the need for words [4]. Language therefore emerged from mimicking others, which we can see from studying the brains of monkeys; the same areas of their brains light up when watching another monkey perform a set of grasping movements [5]. Interestingly, the brain region for monkeys mimicking each other is the same brain region that lights up for human language.

What we can distil from this research is that we learn languages from copying and observing each other. By just observing another's movements, our own brain can help us infer their goals and intentions. This therefore gives us meaning behind their movements. If we relate this to language, by listening to a lot of content in that language and watching their body movements, we can unconsciously infer the meaning of the words.

A video summarizes language learning into four principles [6]:

1. Seek relevance
2. Obtain the content's basic meaning
3. Focus on only what you can understand
4. Build it into memory

For example, you may be fluent in English, but if you were thrown into a PhD study of physics and did not come from a physics background, you would also be lost in its terminology. The content has no personal relation to you; to build this knowledge into memory, you would first need to filter out the principles of the content and understand the basics. After that, you would have to work out how you can fit this new content into your existing knowledge.

Our brains work unconsciously and are constantly seeking out new patterns. In order to learn, our brains categorise the content into something we can group, then abstract it away such that we can form relationships. The methodology works because similar words are located close to each other in your brain. To do this, we need to constantly be seeing and using this content in different contexts (a method called ‘interleaving’) and creating analogies [7]. The idea of analogies links back to how we move this knowledge into our long-term memory, which is by creating relevance around this content.

If we take this through the lens of computer science, we can understand this idea of abstraction through how we code. If we have new words such as ‘car’, ‘motorbike’, ‘truck’, we can categorise these into a class called ‘vehicle’. As we build up these classes and abstract them away, we can more easily draw high-level relationships between them.

How can we apply this to achieving fluency? Fluency occurs when words fit together automatically. We do not necessarily have to think about the next word in a sentence, because our brain has already found the patterns and intuitively knows what words should come next.

Gabriel Wyner, the author of *Fluent Forever* [8], discusses how we can actively use language learning principles and apply them.

1. Pronunciation
2. Get the most frequent words to learn
3. Use comprehensible input
4. Output

With pronunciation, it is important to learn the rhythm of the words and the flow of the words. At the same time, we should be training our ears and our mouth to learn how to differentiate similar sounding words, and how to actually pronounce these words with our mouths. By getting the flow of the language, this applies to principle (2) of ‘obtaining the basic meaning’. Here, we are understanding the gist of the language.

The next step is getting the most frequent words to learn. This helps with principle (1) and (3) - seeking relevance and focusing learning on what you can understand. Before we read a whole chunk of text, we need to create anchors in our brains to latch on to. We have to give our brain context before learning new things, which can let the brain draw connections between ideas and the new words. It is also important that learners do not get bogged down by the nitty gritty of the language; they should instead try and understand the context and overall meaning of the text. This principle is used in most textbooks; students are given the most important keywords to learn before the article, helping them prioritise certain words.

The third is using comprehensible input, relating to principle (1) (seeking relevance) and (4) (building it into memory). By choosing content that we like, we can draw connections in our brains. Wyner talks about how when we get new words, we can make it comprehensible by turning it into stories. We have previously discussed how analogies (and therefore stories) help turn new information into long-term memory. Wyner stresses how comprehensible input does not mean simply translating the words to understand it. He emphasises how we should directly link stories to images. For example, to learn the phrase ‘she is’, we can use the phrase ‘she is a doctor’ and have a picture of a woman in a doctor suit.

Fourth, is output. Outputting is important because this is where we use the new words in different contexts. We can start to use the words in sentences that have a personal relation to us, reinforcing the words in our minds. Playing with the words in our minds also lets us deepen connections with words in that context, helping us draw links between words and start speaking fluently.

4.1. Analogical thinking.

To further investigate the importance of creating analogies in our brain, a study [9] explored how analogical processes in human thinking and learning improved a person’s learning relational retrieval and transfer.

The paper focuses on mapping. In simplified terms, two situations or concepts are aligned to find commonalities and make inferences between them [10]. This theory was previously studied in a paper published in 1890 [11], which aimed to create a Structure-Mapping Engine (SME), a cognitive simulation for analogical matching. There are two important aspects; support, which measures how much an inference is based on the analogy you are making (where more support from the analogy is better) and extrapolation, measuring how much your inferences goes beyond what the analogy directly provides.

Only after mapping does ‘directionality’ emerge, where the meat of the understanding takes place as we explore the analogy further. An example given from paper [10] is the word ‘jail’. A similar word could be ‘prison’. An analogy could be ‘job’. Thus, if the learner was learning ‘jail’, instead of just showing them the word countless times, we can ask them to generate metaphors with the word and talk about their experiences or feelings with that word.

A paper on structural alignment [12] takes this further, exploring why analogies are so effective for memorisation. As we have discussed, our mental representations are hierarchical (we prioritise certain things) and are made up of categories with relations to each other. By comparing two ideas, there is a structural alignment to find a maximal structurally consistent match between representations. The system favours interpretations that preserve a maximal connected related structure; in other words, we remember new information easier if we can draw many links to another idea. Taking this idea into language learning, we can see this in similar sentence structures, or the negation of certain words. For example, ‘I like’ and ‘I don’t like’ are different, but we can remember them due to their alignable differences. Antonyms also work the same way; ‘up’ is related to ‘down’ even though they have opposite meanings.

Intermediate learners can benefit from analogical thinking, because they have a larger vocabulary range and a better understanding of grammar, enabling them to draw better connections. A current issue intermediate learners face is the lack of speaking and listening practice students get in a classroom setting [13]. However, it is through listening where learners can create analogies as they see the words in more contexts, and it is through speaking where learners can develop their own self-confidence in that language. The self-confidence ties in with pronunciation, as students are worried that they sound too foreign.

4.2. Context based learning and self-evaluation.

The most common methods of language learning taught in schools are usually through reading. To investigate the effectiveness of reading in the target language, I explored a paper on the effect of exposure frequency on vocabulary acquisition [14]. The research confirms that reading does serve as a significant source of vocabulary development, but in quite a surprising manner. Although the vocabulary growth is modest, it highlights that reading creates cumulative knowledge, and has a long-term positive impact on adult vocabulary growth. What is paramount is the exposure of the new word in different contexts, which allows the learners to infer word meanings. We can link this back to the idea of how our brains remember better if they work through finding patterns themselves, which is why learning languages through pure translating back to your native language is not the most effective.

If reading only contributes to a modest impact on vocabulary, what are some of the strategies that have been proven effective for vocabulary retention? As already discussed, learning vocabulary is enhanced when we encounter words in context. Flashcards, mnemonics and translations are very common approaches for this, but to prove this assumption, we should assess vocabulary learning through immersion. VocabulARy, a study on learning vocabulary through augmented reality (AR) [15], is an AR application for vocabulary learning that visually annotates objects in AR, in the user’s surroundings, with the corresponding words. The study took two groups, one that used the VocabulARy prototype and another that had an alternate AR system which did not show any additional

visualisation of the keyword. Showing visualisations outperformed the other group in short-term retention, mental effort and task-completion time, and also scored significantly higher in delayed recall and learning efficiency.

Vygotsky, a plurilingual speaker, emphasises the importance of learning in social contexts, where knowledge is acquired from interacting with each other and environments [16].

Through this, he developed a social constructivist theory, where he believed community plays a central role in the process of ‘making meaning’. This links back with our previous research where we discovered the importance of comprehensible input and using context clues to learn new vocabulary. Furthermore, utilising the community aspect of language learning enables constant assessment of a one’s own performance. A community can help with self-evaluation, where learners can ask for feedback on their pronunciation, grammar and vocabulary.

Interestingly, the advantages of self-evaluation can especially be seen with speaking. Speaking is an important aspect of language learning where learners focus on outputting different words in different contexts.

Speaking is known for its difficulty due to message conceptualization and articulation [17]. In this paper, they studied the strategies students took when practising speaking. Participants used extensive planning to structure their arguments and to select appropriate vocabulary, then continuously monitored their performance by assessing their grammar, vocabulary and pronunciation. After, they completed self-evaluation. What we can gauge is that a good speaker actually comes from the planning and thought behind their sentences and arguments, rather than just constantly outputting speech. Bound with this, is the constant assessment of their own performance.

Therefore, having the ability to evaluate and assess one’s own performance is crucial in language learning, as it helps a learner self-correct and identify their own mistakes.

4.3. Mobile learning.

Many language resources today are available through mobile apps. In language learning, mobile devices are great for quickly looking up new words, translating sentences or even finding answers to questions. It also provides a convenient means to access videos, music and content that can be applicable to language learning. With the billions of content available online, learners can also be exposed to different cultures and languages whenever they want. In learning, there is a ‘forgetting curve’ [18] which states that the process of forgetting gets slower and slower over time, as long as you keep repeating the content at timed intervals. Thus, having access to content anytime means that learners can harness the power of repetition and utilise it to their advantage. This is in contrast to the learning environment in schools, where students have to wait each week for a lesson to review their knowledge.

A case study we can use is X (previously Twitter), which is a microblogging platform. A paper looked especially at using X to foster foreign language learning [19], aiming to gain insights about learners’ perceptions of the use of X in language learning and how they feel about tweeting as an extracurricular activity throughout four weeks. X helps educators and learners benefit because the platform enhances student collaboration and interaction.

Students can engage in meaningful communication and get immediate feedback, which as discussed previously, can help students learn more effectively.

4.4. The current language apps.

With this in mind, we can assess the current language learning apps on the market. In particular, I will be discussing the big three language learning apps, Babbel, Duolingo and Rosetta Stone.

Babbel focuses on achieving fluency through immersion of real-life dialogues [20]. It uses the idea of how a native learner would learn, by teaching vocabulary and grammar through practical dialogue examples in conversation. It guides the brain to connect the dots passively by learning new information based on the dialogue context. At the same time, it trains the learner's pronunciation skills. It also emphasises the use of a spaced-repetition system, where you revisit the words in different contexts, spaced out over time.

Cognitive psychology has repeatedly shown the benefits of using short repetition practices in order to put new knowledge into long term memory [21]. Ebbinghaus stated that 'with any considerable number of repetitions a suitable distribution of them over a space of time is decidedly more advantageous than the massing of them at a single time'. The umbrella term for this phenomenon is the 'distributed practice effect' or 'spacing effect'.

Very early on, Ebbinghaus conducted experiments on himself to determine how to minimize the amount of time it took to relearn a set of material. He discovered that spacing the study of simple verbal material across several days rather than all in one day resulted in fewer relearning trials. From this initial study, many other studies branched from learning words, sentences and text passages.

In Glenberg's experiments, he discovered that increasing the amount of time between recall sessions benefitted retention to a point; after this point is reached, the additional intervals lead to a poorer retention. In other words, learners should start from shorter interstudy intervals until a certain interval, and maintain this.

Duolingo is a game-based learning app that stresses the importance of 'learning-by-doing' through interactive lessons [22]. Similarly to Babbel, it uses the idea of your brain picking up patterns passively; thus, Duolingo does not teach grammar rules. It instead pushes learners to figure out the conjugation rules by themselves. By utilising AI, Duolingo adapts lessons to individuals, where the AI model tracks and adjusts the order and difficulty of exercises. The topics chosen to teach are based on school and institutions standards. However, Duolingo is most known for its high engagement, due to its bite-sized lessons and gamification streaks, helping learners to stay motivated.

Finally, Rosetta Stone relies on dynamic immersion [23]. The idea is to use human senses in order to move new words into long-term memory. For example, learners are not given translations, but are instead encouraged to learn the words through the pictures. To achieve grammar, Rosetta Stone gives a few examples of a grammatical concept, then the words the learner should focus on gets highlighted. To achieve speech, they offer services that allow learners to read aloud, and as they do so, their pronunciation gets corrected.

Each language app has their own unique advantages and disadvantages. The use of forcing your brain to draw their own connections to recognise different patterns aligns with the research I have conducted above, and building content based on personal relevance (as Babbel and Duolingo do) has been proven to help retain information. Babbel's spaced repetition system is also very effective, especially as it reminds you of the words in a different context every time. A common misconception of learning is to constantly repeat

information with flashcards, however not only is this boring, the brain starts to recognise the cards and trick itself into thinking it has the card remembered, when it actually has not. The spaced repetition system of Babbel means that the cards for the same word are not always the same, which forces the brain to recognise the words in different contexts.

Duolingo's strength comes with the gamification aspect. Its bite-sized lessons take less than 5 minutes to do, and learners can undergo 'quests' with their friends, allowing them to push each other and support each other in learning. It also makes them feel more accountable when they miss their lessons, because their progress can be shown to their friends. As a personal user of Duolingo, I like how each exercise brings in new vocab to learn, but at the same time is very easy to complete. Furthermore, Duolingo consists of badge systems and rewards, in the hopes that students become more motivated and engaged in their content. Students should be intrinsically motivated to learn [24] (where the desire to learn comes from the student, rather than from external factors such as parental pressure), as this leads to better information retention. However, a substantial body of research suggests that the way we attempt to increase intrinsic motivation should be cautioned [25], because tangible rewards (such as badges) can shift a student's motivation from intrinsic to extrinsic. What they instead conclude is that gamification should be used for the fast feedback to the students.

Duolingo is also famous for its microlearning aspect. Microlearning is similar to gamification, but the critical difference lies with learning goals being masked as game-like activities. The gamification encourages students to participate more in the learning activities by providing them more enjoyable approaches. Duolingo is one of the most popular language apps, but how effective is it? We can back up Duolingo through a study done by Fang [26], which looks at English learning. The study begins with emphasising that despite the exponential growth of knowledge and information with the internet, the traditional 'classroom + textbook' learning mode has failed to satisfy people's needs of seeking knowledge. The new micro-learning approach has become popular among college students due to their ability to use mobile handheld devices, such as a mobile phone, and micro-learning concentrates on brief and independent messages. College students are therefore provided with the information they need anytime.

A paper on microlearning [27] shows that gamification, infographics, videos, apps and social media may all be leveraged to provide this. Microlearning allows lessons to be given in a short length of time and can be accessible at any time and from anywhere. It further explores how microlearning can increase student comprehension and retention, especially when lessons are broken down into digestible pieces. In the modern day, TikTok (a platform for short videos) are extremely easy to remember because of their short, effective nature, where all important information is concisely condensed into a few seconds. This is a great example of microlearning; the only downside is that microlearning is not effective when dealing with a narrow, intricate and complicated issue requiring an in-depth discussion.

Rosetta Stone's strength comes from the fact it uses sensory input. Different languages may contain words that do not exist in the learner's primary language. Thus, there is no direct translation. Therefore, images are incredibly useful for this, because it strips away the intermediary step of having your brain translate to a different language domain, where sometimes the word does not exist. Its stress on pronunciation and having a direct feedback loop is also useful in achieving fluency, because many learners struggle with pronunciation

the most. Learning new languages requires you to train new mouth muscles, muscles that may not be used in the learner's primary language.

4.5. Utilising social media.

When designing a language app, we can also stretch beyond what other language apps do and take a look at social media. Social media is addictive; among the 7.91 billion people as of 2022, the average time individuals spent using the internet was 6h 58 minutes per day, with an average use of social media platforms of 2h and 27 mins [28]. What sets social media apart is the personalization aspect, human connection and the fact people are free to exchange ideas which build up on top of each other. We can see this through the most popular apps of January 2022, which are TikTok, Instagram, Facebook and WhatsApp, whose basic goal is to enable users to share and create content with each other.

The idea of social media is simple: help humans establish relationships [29]. As establishing relationships interferes with necessary life activities such as sleep, nutrition and work, its overuse can be seen as any other addiction as it can dominate a person's life. To establish relationships, people need to be able to connect and engage with others through the sharing of experiences and ideas. In fact, the definition of social media by Merriam-Webster states social media is a 'form of electronic communication (such as websites for social networking and microblogging) through which users create online communities to share information, personal messages, and other content (such as videos)'.

Drawing aspects of social media with language learning is an interesting field to explore; social media gives opportunities for students to understand more about other cultures through videos and blogs. On the other hand, there are many criticisms of using social media, one of the main reasons being addiction. In the context of language learning though, a study in Algeria investigated the effect of social media in writing [30]. In particular, a discussion was drawn between writing formally and informally, where on social media 90% highlighted their use of slang words and abbreviations. When asked about the reasons for informality in writing, the students gave reasons ranging from the fast and easy to use nature, as well as it being helpful in being able to express themselves. It gave them more freedom in writing in contrast to the formal way that obliged them to follow strict rules. The paper concluded that social media did indeed have an overall negative impact on the way students write, as the relaxing environment of social media encouraged students to write through abbreviations, symbols and slang words. In a social media context, this writing style is accepted, but inappropriate from an academic perspective.

However, I would criticise this paper in that this paper was written through the lens of an academic context. There are many goals within the field of language learning; not everyone learns for the sake of academic writing. Social media's strength lies in getting quick ideas across and sparking conversation. Social media's personalisation aspect also means learners can choose what and who they want to learn from; there is plentiful content for those in the academic space, and others in entertainment.

4.6. Summary.

Many of the language learning applications today are focused on beginners, whose aim is to just familiarise themselves with the language. This means there is more emphasis on creating content around general, surface-level topics. Basic vocabulary and grammar structures are not enough to make students confident in a language. Improvement comes from constant self-evaluation and having an expansive vocabulary range and grammar

bank, as well as a fast, personalised feedback loop which does not come from traditional class lessons, as many students only see their teachers once a week.

The solution to this problem is thus a mobile app (due to the advantages of handheld devices for language learning) that will cater to intermediate learners by utilising social media's personalisation and community aspect to generate custom content to the user; techniques such as microlearning, spaced repetition, gamification, sensory learning methods, self-evaluation (especially in the context of speaking); finally, flashcards that promote context-based learning and analogical thinking through the ability to add images and personal notes.

5. DESIGN

To understand the language learning space, we must first identify the main pain points of learners and what is currently missing from the apps available.

I sent out a survey using Qualtrics to language learners of Asian languages and teachers in this area. The survey assessed the most commonly used language apps currently and what features specifically users liked the most. I also wanted to identify how certain apps kept loyal users and contrast them to other language apps that people are not very consistent with.

However, these questions gives a false sense that creating language apps is the only solution to learning a language. In fact there are many other resources available, such as using textbooks, attending in-person classes and more. Thus, the questions in the survey were not just limited to apps, but also learning methods students liked to use and why.

The results can be seen below. Also note that the bullet points in the infographics are comments provided by the survey-takers, where they were given the opportunity to write more in detail about a certain opinion.

Based on my survey, Duolingo was the most used language learning app, followed by Anki and Memrise. Anki is a general flashcard platform and Memrise is a language app also famous for its gamification. Students mentioned that the most useful functionalities were being able to listen to pronunciations, the gamification aspects (the ability to compete against friends through leaderboards and the short lessons which gives a feeling of progression), as well as spaced repetition. These findings match with my research seen in the context survey, but what was emphasised in the survey was the importance of User Experience (UX) and the app's User Interface (UI).

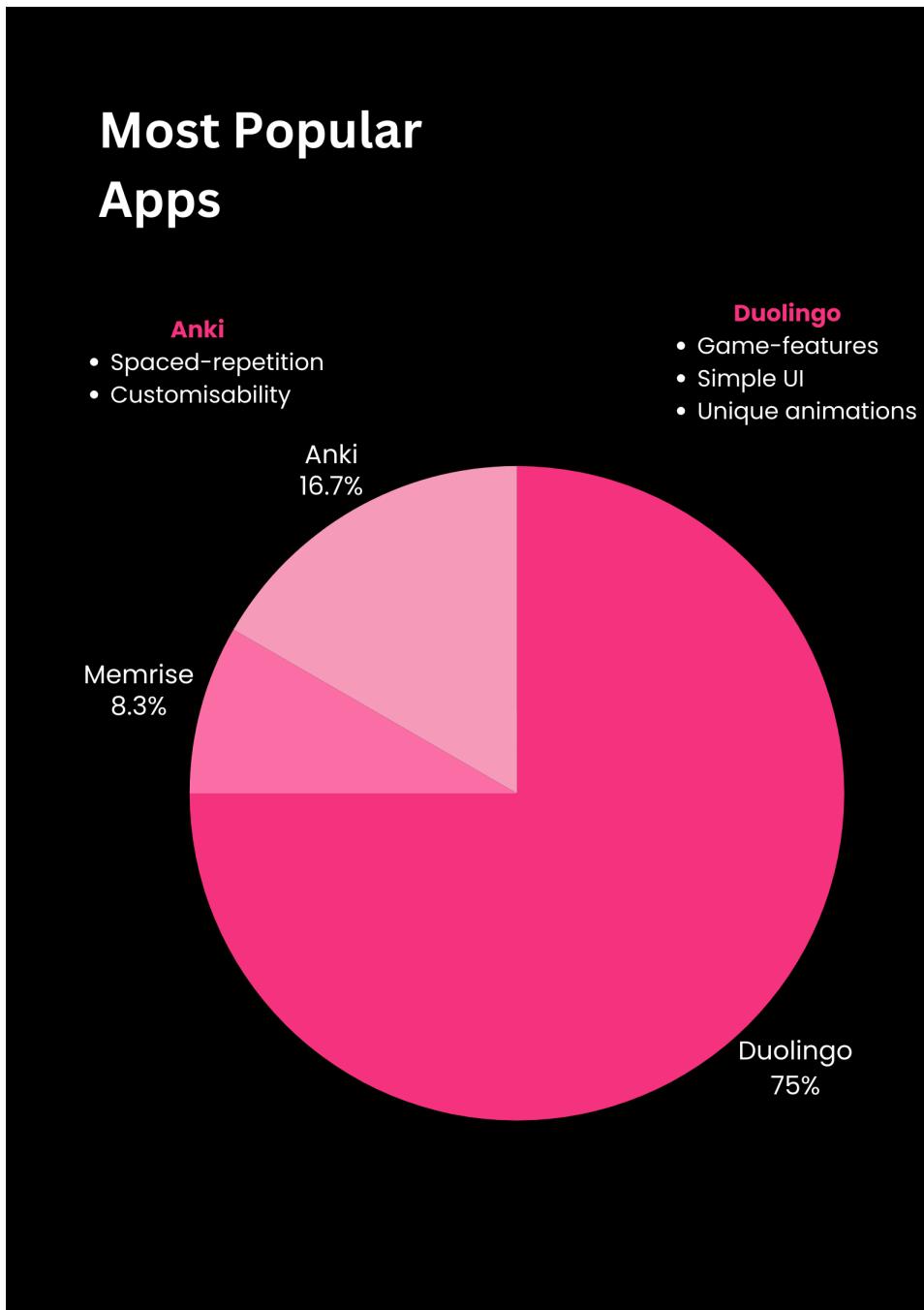


FIGURE 2. Most popular apps

Compared to the interfaces of competing language apps, Duolingo's design is attractive due to its simple animations, clean and intuitive design and game-like features. Those who preferred Anki generally prioritised more their spaced-repetition system and its practicality, however the percentage of those who preferred Duolingo to Anki was 75% to 17%. The customisability aspect of Anki comes from how users can create their own cards, fill it with their own sentences, audio and words. When done well, this provides a optimized experience for long-term memory retention due to its multi-sensory capabilities.

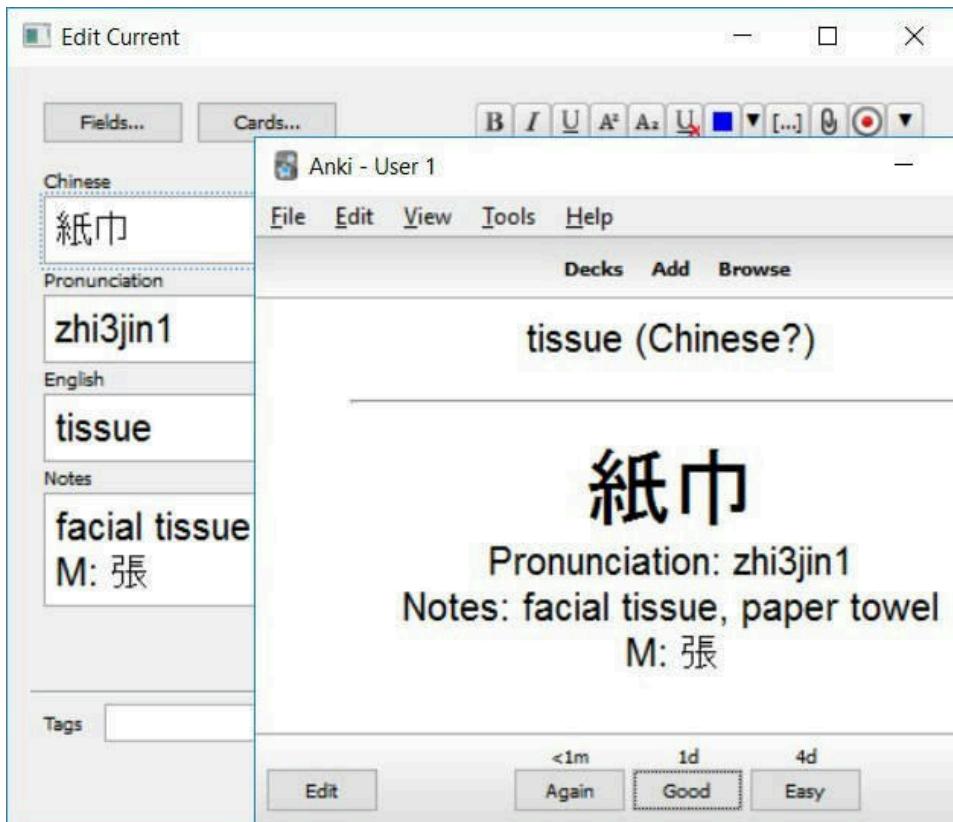


FIGURE 3. Example card creation in Anki

However, Anki may be challenging for beginners due to its complexity [31]. It requires a long setup time to understand the interface if a user is to design their own review deck. Thus, many learners opt to browse and find available pre-made decks online instead. Yet, the majority of the learning experience comes from creating your own flashcards, re-emphasized in a study aimed to promote active learning. The new strategy used is called the ‘Flashcards-Plus’ (FP) [32] where, like traditional flashcards, students identify bold-faced terms from a textbook and write them on one side. Students then write the textbook definition on the other side. However, the FP strategy goes further; students write a definition for the same key word in their own words, and also generate a realistic example of the key term from their own lives that will increase retention.

The study concluded that students who used FP improved their exam performance more than those who did not use the strategy. The downside of this study lies within the possibility of those being exposed to the FP strategy are already actively searching for ways to improve their own grades and study habits, rather than their own improved grades stemming from this new FP strategy alone.

On the other hand, this study aligns with the idea of relating new learnings back to our own pre-existing knowledge.

From my survey findings, users appreciated the listening features of the apps the most. Interestingly, this matches with one of the challenges identified by teachers who took the survey, which was pronunciation.

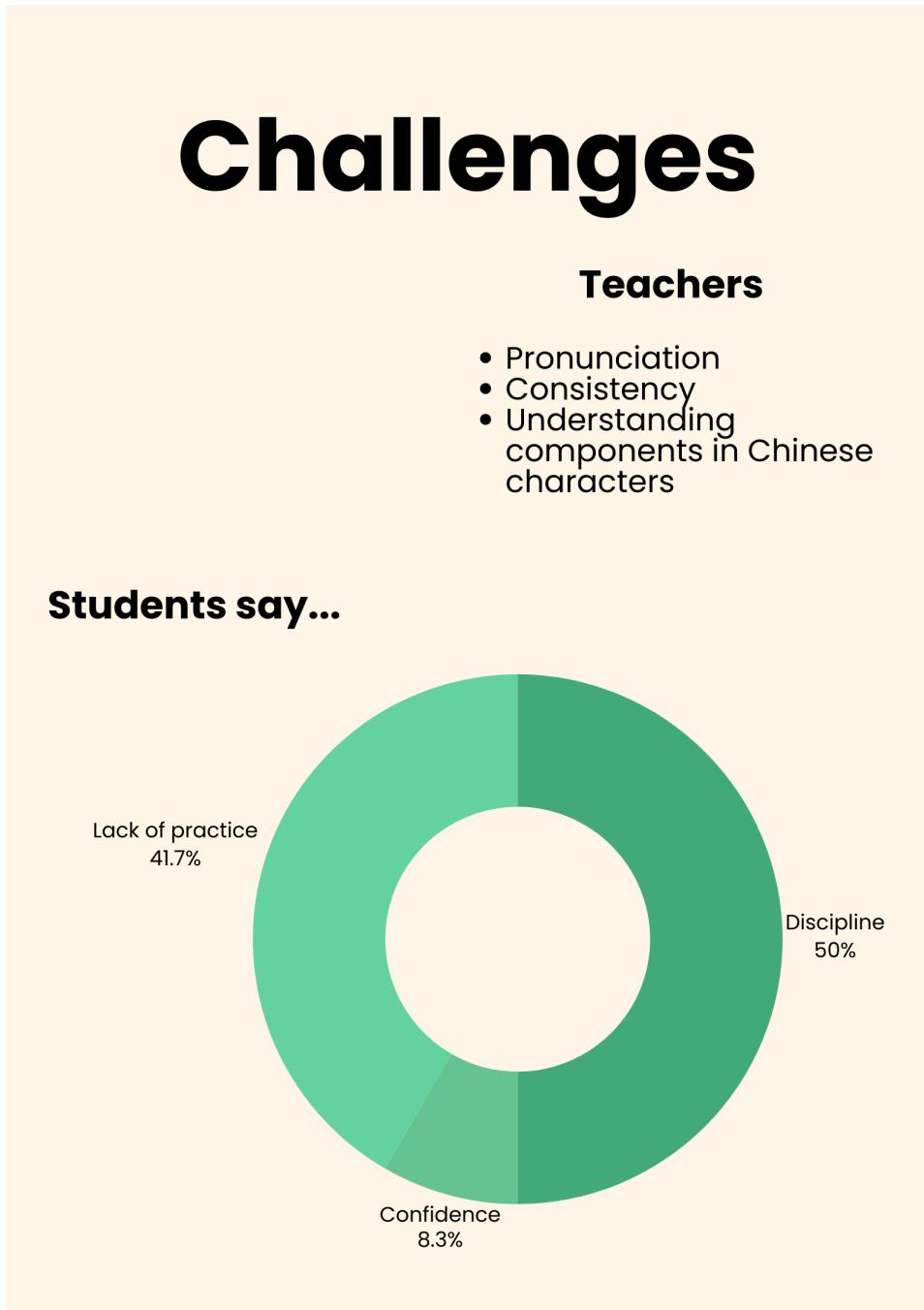


FIGURE 4. Challenges of language learning

In the context survey, Gabriel Wyner (the author of ‘Fluent Forever’) emphasised that pronunciation is one out of four main pillars of language learning. It is thus essential that we therefore include features in our own app to facilitate pronunciation practise. In the language apps listed above, a common functionality is the ability to press on a word and listen to an automated recording of it. Whilst many apps stop there, Duolingo takes this further by incorporating speaking exercises into its games that assesses a user’s pronunciation as they repeat a sentence shown on their screen.

Speak this sentence



CAN'T SPEAK NOW

FIGURE 5. Duolingo's speech exercise

The teachers and students also align when talking about tackling consistency. 50% of the students surveyed mentioned discipline as being one of the main obstacles in language learning. I took this further, conducting semi-structured interviews where I asked Duolingo app users with streaks over 100 days what made them so consistent with the app.

They answered that they liked the competitive nature where they can compete against friends, the short lessons and the feeling of progression. In the context survey, I discussed the advantages and disadvantages competitive learning has on students, and so whilst my survey results do illustrate more of the benefits of competitiveness in learning, it does not give us the full picture. Duolingo's gamification encompasses both types of learners - those who want to compete, and those who want to work together to unlock chests and experience points (XP), through a feature where you can combine XP with friends. This

feature was also mentioned by another survey-taker in a separate interview, where it was stated it was their favourite attribute of the app.

According to [33], autonomy is one of the basic human needs, which contributes to a student's motivation for learning and achievement. It is a strong predictor of student engagement. In order to achieve autonomy, students need to find some degree of meaningful choice and purpose in their learning.

One method to achieve autonomy is by acknowledging a student's interests and building choice in a school's curricula. To understand students' interests, I invited the students to talk about their motivations for learning a language in the survey.

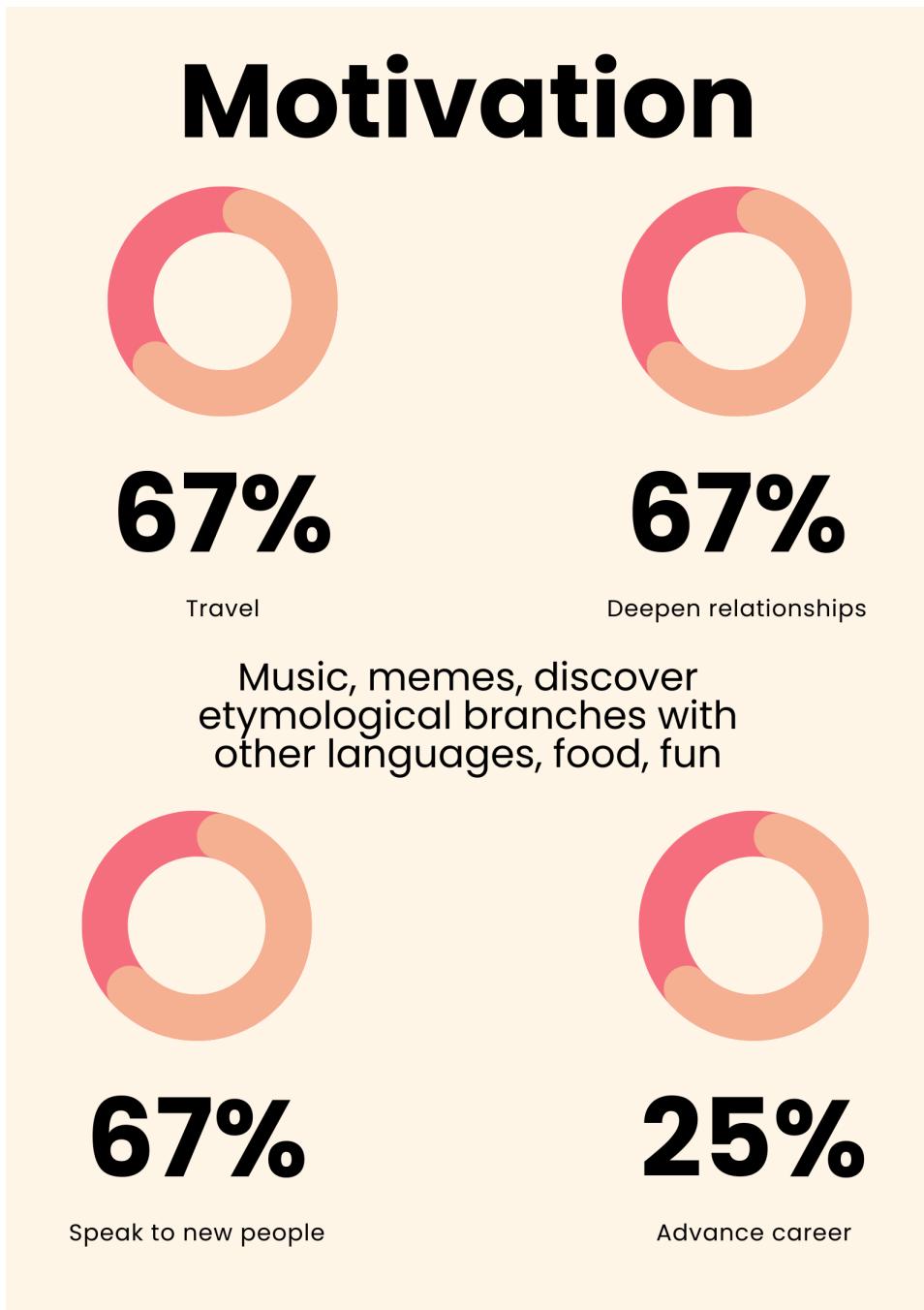


FIGURE 6. Language motivations

67% of the students are motivated by travel, speaking to new people and to deepen relationships. 33% went into ‘other’, and they expanded on this as music, meme culture and for general knowledge.

Current language apps do touch upon these topics, however they provide very general vocabulary. The topic ‘travel’ is expansive and thus it is impossible for an app to devise a curricula that would be relatable for every language learner. Mentioned in my context survey section, cultivating stories that are personal to each person enables deeper thinking

and memorisation. Therefore, learning words that have no personal value may seem ‘irrelevant’ and ‘random’, which are words used to describe the curriculum from my survey results.

The question ‘What do you want to improve the most in?’, also gave interesting results:

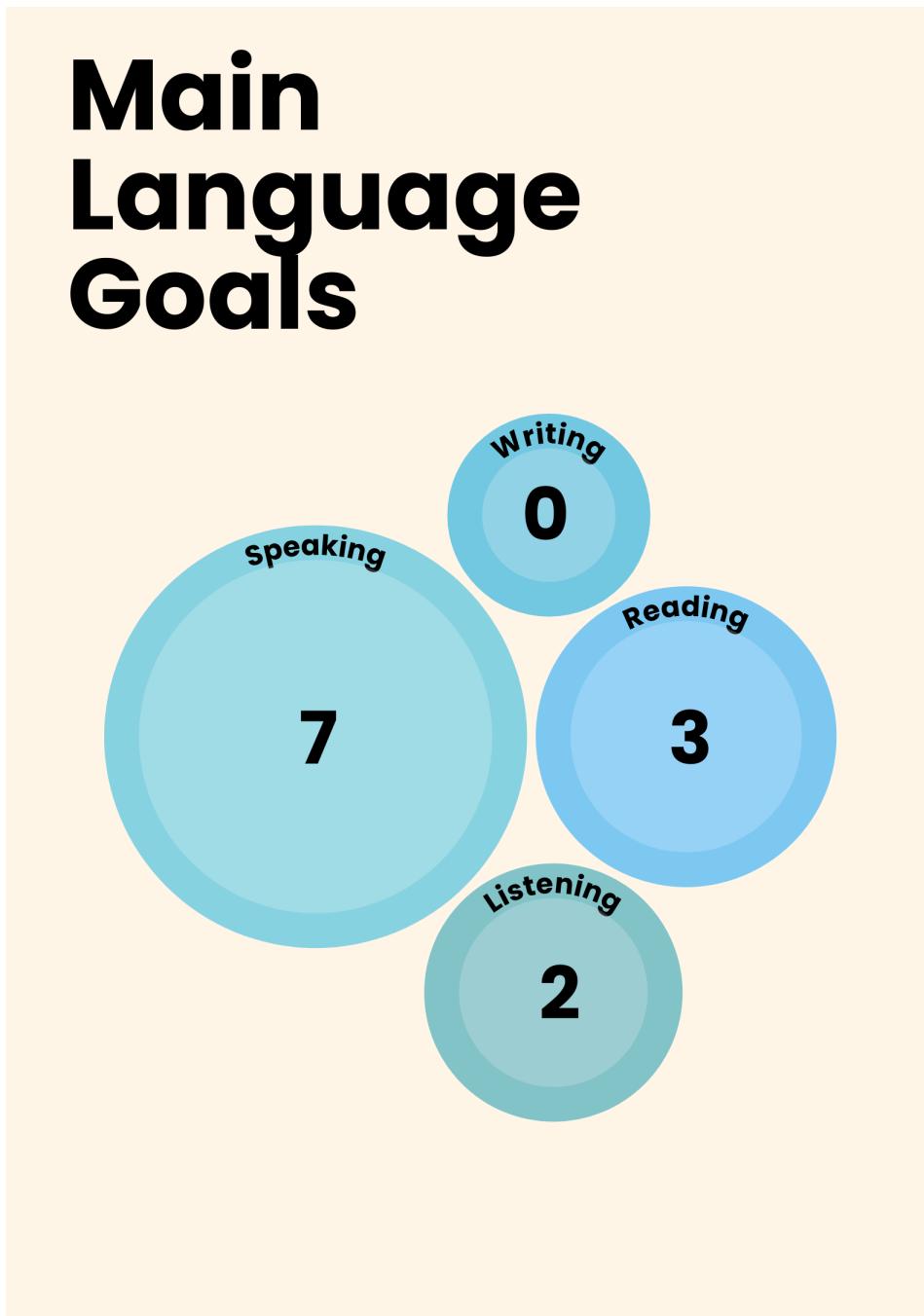


FIGURE 7. Language goals

Students were also given an opportunity to expand on this, being asked what features they wanted to see on this app that would help them progress in their language goals. In the time of reporting this survey, 3 responses were given:

1. Listening to words in a sentence. It's easy to just learn the words, but [not] how the words are used
2. Spaced repetition system
3. Drills, practice and games

In further questions, students emphasised how they wanted more flexibility with their listening tasks and being able to change the speech speed or be able to converse with the app and let it correct your pronunciation. The idea of having not enough real-speaking practice came up multiple times, which went hand-in-hand with how the sentences taught by the apps were irrelevant and the words they suggested them to learn did not identify with them. From this, we learn that when designing the app we should think about how we can make the content as personalised as possible to the user, instead of them following a syllabus written by somebody else.

In fact, one of my survey questions asked students what other resources they use in language learning.

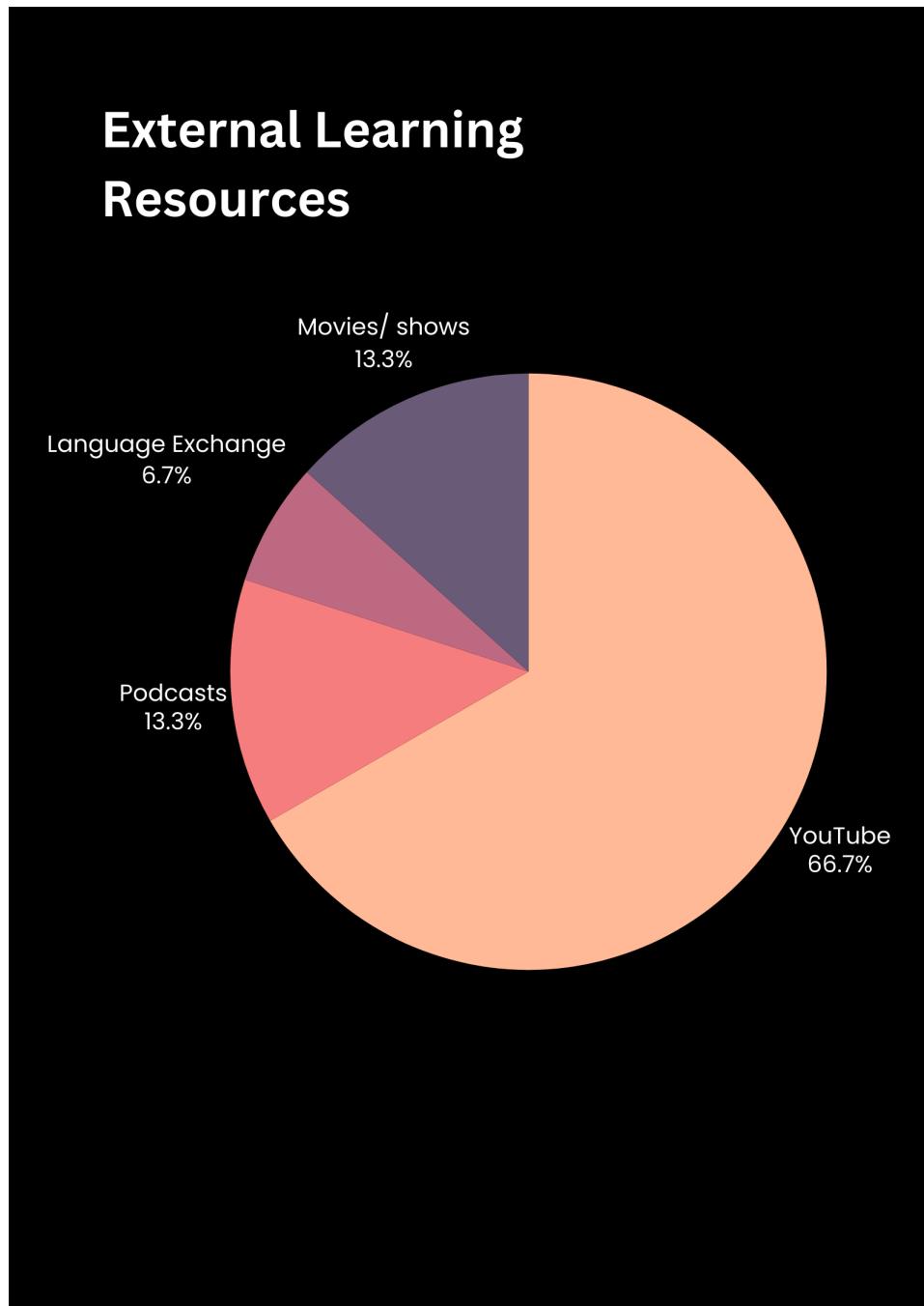


FIGURE 8. External resources

Improving speaking requires a foundational level of vocabulary in that certain topic; this may be the reason why app language learners struggle to achieve ‘fluency’, because the vocabulary taught by the app is not sufficient enough for their daily-life conversations or for them to understand the content they consume, which is unique to each and every person.

Duolingo’s course follows the school system, which gives a limited set of vocabulary. For example, you are more likely to learn ‘I play football’ if you are a footballer rather than a

basketballer. The survey results display that students learn from online resources and content rather than textbooks, where they have unlimited access to niche topics that are personalised to them. Therefore, creating an app that utilises these personalised content can make their learning more enjoyable, improving their language learning consistency and also help them achieve fluency by widening their vocabulary scope to more specialised words unique to them.

On the other hand, teachers were asked about what language apps could do better that can help students. Some suggestions include:

1. Being able to correct when one gets it wrong
2. Using free resources such as Google

While teachers focused on the functionalities, students who were consistent with language apps mentioned more UI-centric capabilities that they would like to see improved. This includes having an easy to navigate, minimally distractive UI (minimal colour usage, large, intuitive icons) and having the ability to customise flashcards.

Building on this, we can combine Duolingo's gamification techniques (short lessons and sense of progression) with Anki's spaced repetition system and custom sentences that can provide a more personalised learning experience to the user. To keep the user experience fun and motivational, which will assist in consistency, the content provided should be unique to each user and provide vocabulary that is relevant to them. Through using online video platforms such as YouTube, students will be able to listen to the content through native users and also see how words are applied in relevant contexts.

Regarding teachers' comments on being able to self-correct, a feature should be included such that exercises get repeated if it has been previously wrong. The accuracy will be used to calculate a 'score' for the certain word, which will then be fed as input for the spaced repetition system.

The survey questions thus far have been biased to learning languages from apps, but this may not be the best approach. Therefore, I later on ask the user about their favourite learning methods.

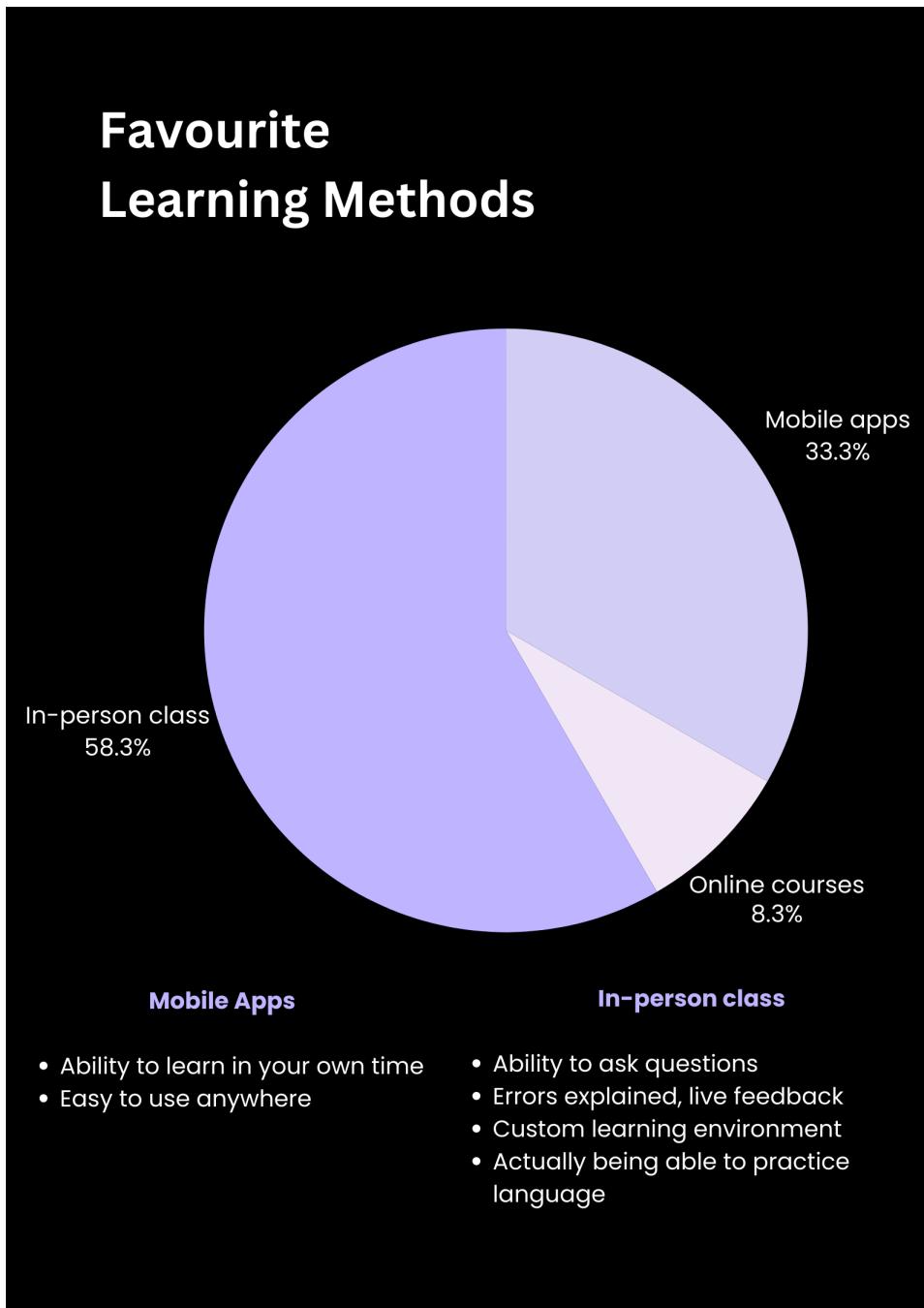


FIGURE 9. Best learning methods

By far, in-person classes are the most favourite language learning approach. The ability to ask questions, interact with others and get instant feedback (such as error explanations) directly from speaking the language in class is the most advantageous when learning languages. Unfortunately, many language apps fall flat with this, and so we see many platforms start to emerge where users can start question threads.

On the other hand, in-person classes can be costly and your curriculum is limited. The advantages of mobile apps is that you can learn in your own time and anywhere.

The question therefore is how can we merge the advantages of in-person classes to the app?

Looking back at previous responses, students emphasised their wish for pronunciation correction. Like most existing language learning apps, the app should have the functionality to listen to how words should be pronounced. It can further be expanded to also prompt the user to output in that language, and assess their pronunciation with an accuracy score. Unfortunately, this will not be the same as an in-person environment where an experienced teacher can correct the learner, but it gives instantaneous feedback. The app can also facilitate a custom-learning experience by providing relevant content and vocabulary to the user that is not restricted by a syllabus.

Issues that have not been highlighted by the survey takers also include the dangers of in-person learning. Students who do not contribute in class will not be able to practice their speaking muscle and thus will not reap the benefits stated above. Furthermore, students may pick up bad habits or mistakes from other students during interaction practices.

Overall, the survey has highlighted the importance of:

1. Gamification in learning with short lessons (improves motivation and consistency due to the feeling of progression)
2. Relevant content that does not just follow a general syllabus (improves autonomy and thus motivation and consistency)
3. Responsive UI that corrects users instantaneously, is easy to use and has a clean design
4. Listening exercises to improve pronunciation and speaking
5. Interaction with native speakers

While my app currently may not be able to achieve (5), further implementation of my app could include using chat bots such as Bing Chat or Chat-GPT to role-play as a language buddy.

6. REQUIREMENTS SPECIFICATION

So far, we have seen importance of personalized, relevant content, microlearning, spaced-repetition, multimedia learning through images, audio and animations, self-evaluation for speaking, and gamification. Gamification and relevant content have also been re-iterated through the initial surveys sent out to language learners and teachers.

One of the main issues of language learning is not being able to find appropriate content. Content in textbooks and apps today follow a particular school system, which contains some topics that may not be of interest to the user. Therefore, social media platforms can be leveraged to provide content that is more relevant to the user, due to their personalization algorithms. A popular social media video-sharing platform is YouTube, where users can find content that is more niche and specific to their interests.

However, just obtaining transcripts from YouTube videos does not help with language learning. Users should be able to make flashcards out of certain sentences and words of interest to learn from. Flashcards should incorporate multimedia such as images, audio and animations to help a student's analogical thinking. The creation of these flashcards would then be re-surfaced to the user in a spaced repetition system, which studies have shown to be effective in long-term memory retention. Interactive learning, gamification and microlearning have also been proven to keep up engagement and motivation, as well as provide a sense of progression. Consequently, the app should incorporate short games with fun exercises to test the user's flashcards.

Those who took the survey also mentioned the importance of a responsive UI that would correct users instantaneously. Games can help achieve this, with exercises including listening exercises that would enhance a user's understanding and speaking, as highlighted by the survey results. Self-evaluation techniques would be utilised in the speaking exercises, where users can record themselves speaking and the app would provide feedback on their pronunciation.

Therefore, the functional requirements for the implementation is as follows:

Functional Requirement	Features required	Priority
As a user, I want to obtain transcripts from interesting videos to study from.	<ul style="list-style-type: none"> • Connect to the YouTube application programming interface (API). • Transcribe the YouTube video. • Translate the selected YouTube video. 	Should
As a user, I want to identify certain sentences and words to make flashcards from.	<ul style="list-style-type: none"> • Incorporate NLP models to do word segmentation on the transcript. • Create a database to store the user created flashcards. 	Should

<p>As a user, I want to be able to review the flashcards in a spaced repetition system.</p>	<ul style="list-style-type: none"> • Create a spaced repetition system where the tested vocabulary gets surfaced to the user at optimal times. • Allow the user to create their flashcard on the app. 	<p>Could</p>
<p>As a user, I want to be able to improve listening and speaking skills.</p>	<ul style="list-style-type: none"> • Games that allows users to listen to the sentence pronunciation to test listening skills. • Games that allows users to speak the sentence to test speaking skills. 	<p>Could</p>
<p>As a user, I want to be engaged with the app and have fun while learning.</p>	<ul style="list-style-type: none"> • Incorporate progression aspects into the app, such as streaks. 	<p>Could</p>

7. IMPLEMENTATION (BACKEND)

Flask, a lightweight Python-based microframework, was chosen for the backend implementation due to its simplicity and flexibility, its active community and updated documentation. Flask also utilises Python, which contains many useful libraries for word segmentation and other natural language processing (NLP) tasks.

Figure 10 is a UML diagram displaying the relationships between my Flask models. SQL database was used because SQL can be accessed with Flask through SQLAlchemy, a Python SQL toolkit and object relational mapper (ORM). Flask-SQLAlchemy provides ways to interact with and gain access to the database's SQL functionalities [34]. The ORM aspect allows for easy querying by using simple Python objects and methods rather than writing SQL statements.

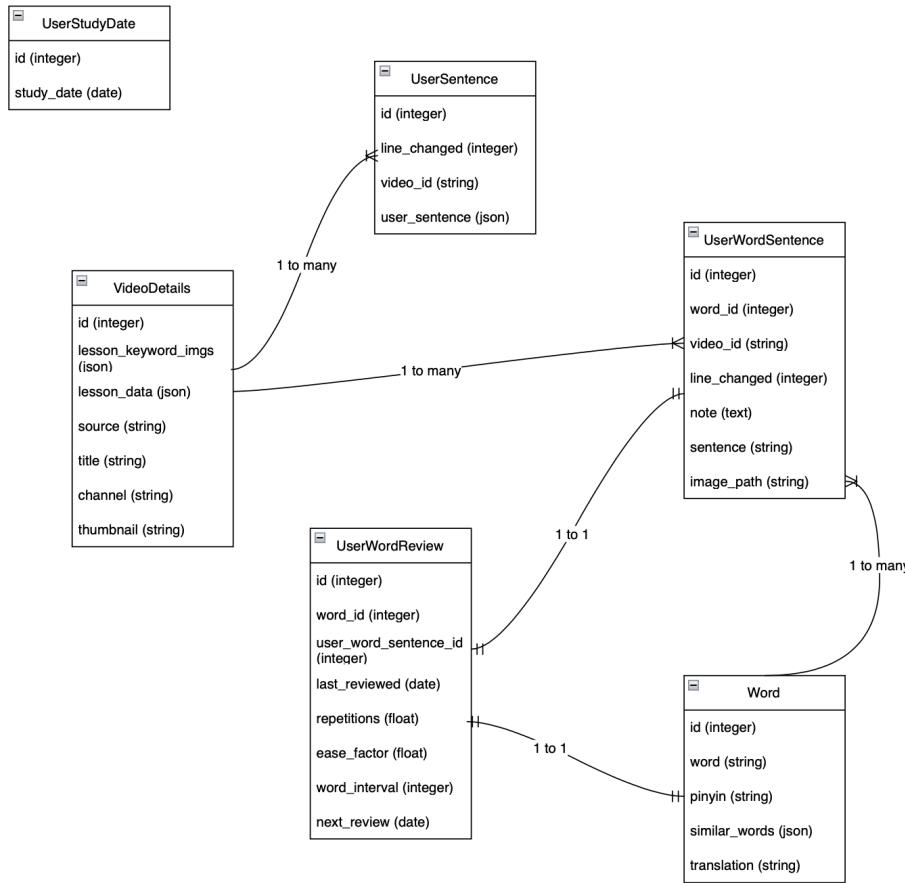


FIGURE 10. UML diagram

Figure 10 displays the relationships between these tables. The `UserStudyDate` is a table in charge of dealing with user streaks. The table gets updated whenever a user finishes a game lesson and a new `study_date` entry gets added.

The `VideoDetails` contains the video information from YouTube. The model saves the YouTube video id; a dictionary of the video's keywords and their respective images (the implementation of which is discussed below); the `lesson_data` consisting of the video's

transcript now segmented into chinese characters, their pronunciation, translations and similar sounding words; the source of the video (in this case, it is assumed to be YouTube, however if this app were to be expanded it could contain videos from other mediums); finally the video's title, channel and thumbnail.

Displayed beneath is an example JSON dictionary containing the 'keywords_img' as well as the 'lesson_data'. A video gets separated into different 'lessons' as video transcripts can be hours long if users want to transcribe long-form content such as podcasts, which Stanza cannot handle. Therefore, the transcripts get segmented into smaller chunks before being processed and appended into the 'lessons' array.

```
{
    "keywords_img": [
        {
            "img": <imgurl>,
            "keyword": "亚洲"
        },
        ...
    ],
    "lessons": [
        {
            "segment": {
                "duration": 1.291,
                "segment": "加州留学生的生活",
                "sentences": {
                    "entries": [
                        {
                            "pinyin": "jia zhou",
                            "similarounds": [
                                "甲胄",
                                ...
                            ],
                            "translation": [
                                [
                                    "加",
                                    [
                                        "to add",
                                        "plus",
                                        ...
                                    ]
                                ],
                                ...
                            ],
                            "upos": "PROPN",
                            "word": "加州"
                        },
                        ...
                    ],
                    "sentence": "加州留学生的生活"
                },
                "start": <starttime>
            }
        },
        ...
    ],
    "video_id": <videoid>,
    "source": "YouTube",
    "title": <title>,
}
```

```

    "thumbnail": <thumbnail>,
    "channel": <channel>
  }
};


```

There is a one-to-many relationship from VideoDetails to UserSentence, as a video contains many sentences. This contains fields such as the sentence id which gets automatically incremented, the line number of this sentence in the context of the video, the id of the video that this sentence is related to, and finally the actual sentence, which again is a json containing the details of each word in the sentence, its translation, pronunciation, other similar sounding words and their part-of-speech (POS).

```

{'sentence': '加州留学生的的生活',
 'entries': [
  {
   'word': '加州',
   'upos': 'PROPN',
   'pinyin': <pinyin>,
   'translation': <translation>,
   'similarwords': <otherwords>
  },
  ...
  {
   'word': '生活',
   'upos': 'NOUN',
   'pinyin': <pinyin>,
   'translation': <translation>,
   'similarwords': <otherwords>
  }
 ]
}

```

When a video gets downloaded, the sentences are processed using Stanza (this will be discussed more in detail below, under the section ‘word processing’), before being uploaded into the UserSentence table.

The UserWordSentence is used for the creation of flashcards and has a one-to-many relationship with a ‘Word’. A ‘Word’ is a character or set of characters in chinese that gets saved by the user to be tested later using a spaced-repetition algorithm (an algorithm that would resurface the word at optimal intervals to re-inforce learning). In this schema we have a foreign-key relationship to the word being tested, the video that the sentence is part of, the line of the video the sentence being tested comes from, a personal note that would help with recall, the actual sentence string in order to keep the word’s context, and finally an image url that the user chooses from the frontend as the flashcard multimedia. There is a many-to-one relationship with Word, because a user can use the same sentence to test many different words (there are many words in a sentence).

The Word model simply contains strings for the actual chinese characters (word), its pronunciation (pinyin), a json list of similar sounding words (to help with pronunciation) and a json list of possible translations.

Finally, the UserWordReview model is utilised to store information for the spaced repetition algorithm.

This algorithm is inspired by Anki's own implementation, where a user ranks a certain flashcard's difficulty from 0-5, where 5 is a perfect response. In my implementation, this is calculated from how many of these exercises per word are correct (seen later when discussing frontend, each word gets reviewed 5 times through 5 different exercises). A perfect 5/5 score would thus give a user a ranking of 5. The calculation also requires the number of previous repetitions of this flashcard, its previous ease factor (a floating point number generated by the last iteration of the spaced repetition algorithm to determine the number of days before the next review), and the previous interval that the user has seen the word. Additionally, I added a field called 'next_review' so that querying is easier. I simply have to query all reviews where next_review is less than or equal to today, which can then immediately be shown to the user.

The algorithm spits out a new interval (the number of days for the next review), increments the number of repetitions, and calculates the new ease factor that has been adjusted based on how well the flashcard was remembered. All of this information gets sent to the server after every 'lesson' gets completed.

7.1. Endpoints.

In this section we discuss the URL routes that the frontend application can query.

7.1.1. Processing a video.

The aim of this endpoint is to allow users to obtain the YouTube transcript from YouTube. This gets sent to a worker thread which will then do word processing, described in the next section.

This is a POST request to <http://projectvm05.cs.st-andrews.ac.uk:8080/vid> requiring the following fields:

```
{
    "video_id":<videoid>,
    "source": "YouTube",
    "forced":<booleantruefalse>,
    "title": <title>,
    "channel": <channelname>,
    "thumbnail":<thumbnailurl>
}
```

The 'forced' field is required if a user wants to re-download a video and overwrite the existing one in the database.

7.1.2. Obtain a video.

The aim of this endpoint is to allow users to see the json data for a particular video. This will query the VideoDetails table to obtain the relevant row with a particular video_id. At the same time it will also query the UserSentence table to get all the relevant sentences that are related to this video_id.

By combining these two queries, we obtain the mapping for the video's keywords and their image urls, as well as all the sentence data for the video, where each word contains information about their pronunciation, translation, POS and similar sounding words.

This is a GET request to http://projectvm05.cs.st-andrews.ac.uk:8080/getlesson/<video_id>.

7.1.3. Add a study date.

The aim of this endpoint is to update the UserStudyDate table so that we can later calculate the study streak of the user. Whenever this endpoint is called, the server will calculate the current time in Europe/London time, to maintain consistency. Then, it will check if the study_date already exists in the table; if not, it will add a new entry to the table.

This is a GET request to the endpoint `http://projectvm05.cs.st-andrews.ac.uk:8080/addstudydate`.

7.1.4. Get study streak.

The aim of this endpoint is to motivate a user by supplying the current study streak of the user. To achieve this, we first query all the unique study dates from the UserStudyDate table, and order them from descending from today.

In the case the user just opened the app, they would want to be shown the streak starting from yesterday. Thus, we start from yesterday's date and iterate through the study dates until there is a gap bigger than 1 (it is no longer consecutive). When this is done, we check whether there exists a study_date equal to today; if so, we add one to the current count.

This is a GET request to the endpoint `http://projectvm05.cs.st-andrews.ac.uk:8080/getstreak`.

7.1.5. Get video library.

The aim of this endpoint is to query all of the previously processed videos for the user. It is a simple query that obtains all videos from VideoDetails and returns this to the user.

This is a GET request to the endpoint `http://projectvm05.cs.st-andrews.ac.uk:8080/getlibrary`.

7.1.6. Get cards to review today.

The aim of this endpoint is to obtain all of the flashcards to be tested depending on the spaced-repetition algorithm (as previously discussed).

To achieve this, we join the UserWordReview and Word tables on word_id, and filter all reviews where the next_review field is less than or equal to today.

When this is done, we loop through the review and the word. Here, the word is isolated and does not come with the sentence it is part of, giving the user no context. Therefore, logic is required to also query the UserWordSentence table to obtain all the words with the specific word_id being tested and obtain a random sentence out of all of those options. This is then used to obtain the relevant UserSentence entry.

Now, we can obtain all relevant information to test the user with - the image related to that word, a sentence the word appears in, and the personal note aligned with that word. Along with this is the review information, such as the last_reviewed fields, repetitions (the number of times this word has been used to test the user), its ease factor and other details regarding the spaced-repetition algorithm.

This is a GET request to the endpoint `http://projectvm05.cs.st-andrews.ac.uk:8080/getcardstoday`.

7.1.7. Update the spaced repetition (SRS) system.

A user will have to update the review information of a card when they have completed a game lesson. Each lesson consists of 5 words; instead of sending 5 network requests to the server, once per word, it would be better do utilise batching, minimising overhead.

This is achieved by the server iterating through all of the words, obtaining their word_id, number of repetitions, the previous ease factor, the previous word interval and the quality of how well the user recalled the word, calculated by the frontend. Passing this to the function `update_user_word_review` in the ModelService class, these parameters are fed into the spaced repetition algorithm and each word obtains a new ease factor, number of repetitions, the interval for the next review as well as the next review date. Finally, this gets updated in the UserWordReview table.

This endpoint is called via a POST request to `http://projectvm05.cs.st-andrews.ac.uk:8080/batchupdatereviews` with the fields:

```
[  
  {  
    "word_id": <wordid>,  
    "last_repetitions": <lastrepetitions>,  
    "last_ease_factor": <lasteasefactor>,  
    "word_interval": <wordinterval>,  
    "quality": <quality>  
  },  
  ...  
]
```

where quality is a score from 1-5 based on how many of the exercises the user got correct.

7.1.8. Creating flashcards.

When creating a flashcard, a new record must be inserted into the Word table with its relevant pinyin (pronunciation), similar sounding words and translations. This is done after a check to see if the word does not already exist.

After this, we must create a new UserWordSentence where the word_id is equal to the id of the Word just created, and initialise the review information for this word with the default values of last_reviewed being the current date, repetitions as 0, ease_factor as 2.5 to follow the spaced repetition algorithm, word_interval as 1 and the next_review set to be the same day.

This is a POST request to `http://projectvm05.cs.st-andrews.ac.uk:8080/addnewreview` with the following fields:

```
{  
  "word": <word>,  
  "pinyin": <pinyin>,  
  "similar_words": <similarsoundingwords>,  
  "translation": <translation>,  
  "video_id": <videoid>,  
  "line_changed": <linechanged>,  
  "sentence": <sentence>,  
  "note": <personalnote>,  
  "image_path": <imageurl>  
}
```

7.1.9. Updating flashcards.

There are two ways to update flashcards. The personal aspects of the flashcards are the image for the flashcard, as well as the note written by the user. These fields are updated by simple POST requests. For updating an image url, it is <http://projectvm05.cs.st-andrews.ac.uk:8080/updateimagepath>. Consequently, the endpoint for updating the note is <http://projectvm05.cs.st-andrews.ac.uk:8080/updatenote>.

The fields required for `updateimagepath` are as follows:

```
{
    "video_id": <videoid>,
    "word_id": <wordid>,
    "line_changed": <linenumber>,
    "image_path": <updatedimagepath>
}
```

The fields required for `updatenote` are as follows:

```
{
    "video_id": <videoid>,
    "word_id": <wordid>,
    "line_changed": <linenumber>,
    "note": <updatednote>
}
```

The fields ‘`video_id`’, ‘`word_id`’ and ‘`line_changed`’ are required to uniquely identify a row in the `UserWordSentence` table.

7.2. Worker threads.

The worker threads are in charge of processing the video transcripts.

An interesting aspect about Chinese is the fact there are no spaces between words. Word segmentation is thus difficult, as we cannot simply split sentences based on the whitespace character. Chinese characters additionally can have their own meaning by themselves, or be combined with other characters to form different words.

One example is given from an article on chinese word segmentation [35]. The phrase: ‘你们研究所有十个图书馆’ can have multiple meanings depending on which characters you combine.

One interpretation is:

你们(‘you’)/研究(‘to study’)/所有(‘all’)/十(‘ten’)/个(classifier)/图书馆(‘library’), meaning ‘you go study all the 10 libraries!’.

Another interpretation could be:

你们(‘you’)/研究所(‘institute’)/有(‘to have’)/十(‘ten’)/个(classifier)/图书馆(‘library’), meaning ‘your institute has 10 libraries!’

Therefore, I decided to use a library for Chinese segmentation. In this process, I tried both Jieba and Stanza, however settled with Stanza due to the more advanced features Stanza offered, such as position of word (POS) tagging (categorizing a word as an adjective, adverb etc), lemmatization (finding the word’s root), as well as segmentation.

7.2.1. Video processing.

Incorporating Stanza [36] into the app was by creating a YouTubeHelper class with functions that initialized stanza, such as translating the youtube transcript into simplified chinese, processing each segmented word to obtain their POS, pinyin (pronunciation of the chinese word), their possible translations and similar sounding words, and lastly, splitting the transcript into manageable, processable chunks for Stanza to compute.

Chinese characters can be written in two forms; traditional and simplified. The majority of chinese speakers use simplified characters, however some transcripts from YouTube are only available in traditional chinese, as seen in videos from Taiwanese speakers. Therefore, I used an API called HanziConv [37] for the specific purpose of converting all chinese characters between the two types. When the whole transcript is in simplified form, the phrases in the transcript are traversed and stanza is used to segment each word. At this time, the word gets put into a function called ‘process_words’ which would query an API called hanzidentifier [38] to check if this word is a valid chinese character, before calling a pinyin library [39] to obtain the word’s pronunciation and possible translations. Obtaining the similar sounding words is also done in an akin manner, where dimsim [40] is instead the library used. All of these libraries are readily available Python modules.

From the lens of scaling this app, it would be beneficial to use caching to prevent the repetitive calls for common words. Constantly calling words such as ‘the’ or ‘he/she’ introduces a lot of overhead. Thus, Redis, an in-memory data structure store, was used for the purpose of caching the pinyin, translation and similar word requests to speed up processing.

Later on in the frontend we discuss the implementation of syncing the YouTube video to the captions and their respective translations and pinyin. This is prepared in the main logic of the YouTubeHelper class, which saves the time that phrase is spoken as well as its duration, which is supplied by the YouTube API itself.

Additionally, the YouTubeHelper class utilises additional APIs such as the TextRazor API [41] to get the keywords for the transcript, as well as the PyUnSplash API [42] to get corresponding images for these keywords. These keywords are displayed in the order of most frequent to the least frequently used in the transcript. Regarding the PyUnsplash API, the image URLs are saved with the rest of the transcript data, such that the frontend can look it up via a simple network request supplied by Flutter. To obtain the YouTube transcript itself, a YoutubeTranscriptApi [43] was used. Additional logic was added to ensure that only videos with chinese captions could be processed, to avoid any errors.

A downside of relying on these APIs are the limitations proposed by the data controllers. PyUnsplash, for example, limits image queries to 50 images per hour. Therefore, if the app were to scale to more users, an alternative API may have to be used, or costs will have to be involved for the best user experience. Otherwise, some videos displayed to the user will not contain images of the keywords for the transcript, which may hinder learning.

7.2.2. *Obtaining YouTube transcripts.*

The worker is also in charge of dealing with YouTube videos. When a user requests to process a new YouTube video, this is passed to Celery, a distributed task queue system of multiple workers and brokers to enable high availability and horizontal scaling [44]. This architecture is shown in Figure 11. Celery was used due to the long processing time of

Stanza, which requires asynchronous work outside of the usual HTTP request-response cycle which the rest of the Flask backend could handle.

Thus, when the backend route is called to process a video, this gets taken up by a celery worker which will do the video processing as explained in the above step.

Celery workers are also used for obtaining the keywords and their image urls from PyUnsplash, again due to the long waiting time. If a user were to wait for a response, the users would most likely quit the app due to a bad user experience. Both of these steps are achieved from the aforementioned ‘YouTubeHelper’ class.

Once both these steps are done, the data gets added to the VideoDetails table. Celery contains useful methods such as ‘group’ which allows tasks to be done in parallel, depending on the amount of worker threads available. In my system, I have provided 2 threads; for the purposes of the minimal-viable-product (MVP), only one user can use the app at a time, so the maximum number of threads open at a time would be 2 (one for obtaining the keywords and their images, another for the stanza word processing).

A chord function, in-built in celery, is also used such that after these two tasks are done in parallel, the obtained data gets immediately added to the VideoDetails table.

Thus, the user can utilise the other features of the app as the video processing tasks run in the background.

7.3. Architecture & Hosting.

The backend architecture consists of different services: the main Flask backend, the SQL database, NGINX and the celery workers. One of the main issues of implementing the backend was realising how to combine all services together, such that they could speak to one another. Each celery worker had to be passed tasks from the Flask app and thus required their own instance of the Flask app context. Furthermore, each celery worker must also have their own instance of the SQL database so that they could add new videos into the VideoDetails table.

Therefore, the celery service contains some repetition of code from the Flask app service, such as the database models. This is a common downside to microservices architecture, however it is a necessary step because it allows decoupling of logic. Microservices allow for better scaling because compared to a monolithic application, if one service goes down, the rest of the services stay running [45].

The next step is to containerize the whole of the backend, so that from the point of view of the frontend, it only talks to one service. This is achieved through the orchestration tool, docker-compose, which allows us to run multi-container applications. In one command, all of the microservices can be created and started, just from one configuration file [46].

Docker enables the creation of a portable environment which means that your application can run in any server.

Finally, this backend must be hosted such that any frontend client can access the endpoints. Utilising University of St Andrews’ virtual machines, the docker-compose network was spun up. This means that any device connected to the University’s eduroam can connect to the endpoints discussed above.

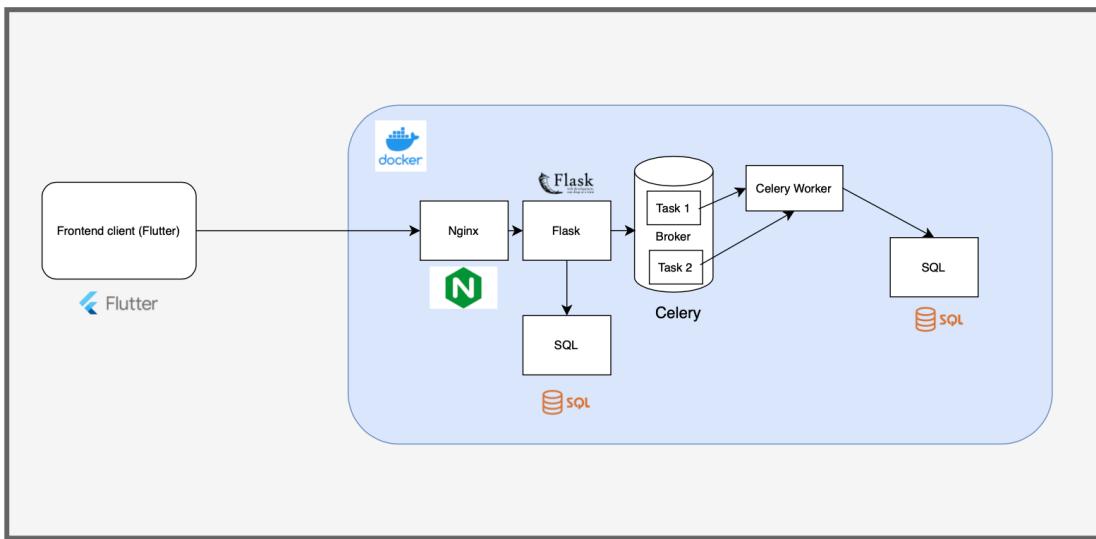


FIGURE 11. High-level architecture

8. IMPLEMENTATION (FRONTEND)

8.1. Flutter.

The decision to create a mobile app over a web application was chosen because of its accessibility. Although the backend will be hosted online (on a virtual machine supplied by the University), certain data will be cached locally on the mobile device itself. Whereas, a website requires constant WiFi connection and inhibits practicing a language on the go. Adding on to the advantages of mobile apps explored in the context survey, mobile phones can help students draw better connections between vocabulary due to their portable nature. With the case of learning the word ‘coffee’ in a coffee shop, having a mobile app readily available at any location is very beneficial, helping to provide a deeper and more authentic learning experience [47].

In another case, Lu and her colleagues (2014) designed a mobile app [48] presenting Chinese characters along with pinyin (chinese pronunciation) and illustrations on the stroke order in writing Chinese characters. Additionally, games such as ‘Pinyin Match’ and ‘Hearing and Match’ were implemented, with gamification elements such as levelling up and competition. The final results showed that students positively engaged with the mobile app to practice and learn Chinese characters. Teachers also reported that the mobile app accommodated a variety of students’ learning abilities.

After the study, further recommendations for the app were recorded, such as providing animated hints on the correct stroke order as students wrote on the screen, offering instant feedback from the games and also archiving and organising learning artefacts as the students’ learning portfolios.

These recommendations are interesting as it relates to the teachers’ results from the initial survey conducted. Teachers thought that modern language apps today lacked the ability for students to self-correct which could be helped by providing animated hints on stroke order and instant feedback from games.

The same paper also mentioned that mobile apps provided opportunities for learners to practice the language outside the classroom. One example is when students were tasked to take appropriate photos to interpret certain idioms - these students then reported that it enhanced their understanding of these Chinese idioms.

Through the initial surveys, the idea of gamification was stressed and thus my initial intention was to use Unity. Unity is a cross-platform game engine since 2005, and is popular with many mobile game app developers. Ideas of creating in game characters that the app user could talk to were proposed through utilising Large Language Models (LLM) models, however this idea was pushed to the backlog due to not being able to test the LLM’s accuracy. A user may pick up something wrong from the LLM, which is detrimental to the user’s learning. Instead, the learning in the app should be similar to Anki, where there is a set answer to check with. The learning overhead for Unity was very big. Not only did art need to be involved (you create your own models to put into your game), it is also not a very friendly platform for customizability, for example adding extra packages that is beneficial to my language app.

Thus, I took a look at other app development platforms, such as Android Studio and XCode. Android Studio and XCode are native to platforms (android and iOS respectively),

which means that scaling this application to a wide audience would be difficult. Luckily, I came across a cross-platform alternative, Flutter.

Flutter brings along many additional advantages. Flutter contains many readily available widgets that allows developers to focus more on the application logic rather than the user interface. These widgets are available on different OS versions and thus will bring fewer compatibility issues when distributing the app.

The active Flutter community means that packages are constantly being updated. As seen later in this paper, features such as tracking a user's stroke to learn a character's stroke order is an example of a pre-made package that developers can use, instead of creating the feature from the ground up. This allows for a faster and simpler development time.

8.2. Frontend architecture.

Before building Flutter apps, it is important to first decide how to structure it. A normal flutter app follows an architecture made up of four distinct layers; the presentation layer, application layer, domain layer and data layer.

Starting from the bottom up, the data layer represents the data sources, in our case all the json queried from the backend. The domain layer speaks to the data layer and converts this json into models that can be used by the rest of the frontend. This can be achieved by the use of repositories that deal with data serialization and data parsing. The application layer contains services that deal with the application logic and can access the many repositories, and finally the presentation layer consists of the widgets and controllers that deal with the interface.

This approach is great for very simplistic apps. When we begin to deal with larger applications, we need to take this architecture a step further and think about a feature-first or layer-first approach [49]. This is because scalable apps consists of a lot of different features. We can think of a feature as an action a user must take to achieve a goal.

A layer-first approach would look something like this:

```

src
  presentation
    feature1
    feature2
  application
    feature1
    feature2
  domain
    feature1
    feature2
  data
    feature1
    feature2

```

However one of the pitfalls of this approach is when adding a new feature3; we would have to make changes in each folder.

Instead, we can take a layer-first approach, as follows:

```

features
  feature1
    presentation

```

```

application
domain
data
feature2
presentation
application
domain
data

```

This is more logical because when adding a new feature we can focus on just one folder and it is easier to make changes to one feature.

The next step is therefore to decide how to split the app into the different features.

8.3. Frontend features.

Based on the user feedback and research as shown previously, we have identified the importance of context-based learning, using multimedia to reinforce understanding of words, the usefulness of spaced repetition and how gamification can increase a user's motivation and learning consistency.

To achieve context-based learning, obtaining YouTube videos for users to learn would be beneficial as users can understand and 'shadow' the native YouTuber in real-time.

Shadowing is a technique used by language learners to bridge the gap between listening and speaking. Whilst a learner listens to speech, they simultaneously repeat what they attended to.

Shadowing brings the benefits of bringing a learners' attention to the phonological aspects of what they hear, rather than the meanings, as there is very little time lag [50]. The same study showed that text-presented shadowing, where learners shadow together with a written script of a target passage, may improve reading skills and possibly pronunciation.

The disadvantage of shadowing is that a learner cannot hear themselves speak, as the attention is on listening to input and reproducing it orally. Thus, the paper emphasises the importance learners recording themselves for self-reflection, re-iterating what we have seen in the context survey.

Thus, providing YouTube transcripts that sync with the videos allow learners to practice shadowing. Users should be able to search for videos and save videos so that the learner can re-visit the same video. Later on, we also touch upon self-reflection through recording the user's voice in the provided games.

Using multimedia combined with spaced-repetition is a combination mainly found with flashcard apps, such as Anki. Flashcards mainly contain a sentence or a keyword to be learned; when flipped, it contains the translation. For a flashcard to be fully utilised, images, sound and animations should be comprised in the flashcard. Therefore, my app will contain a user flow allowing for users to design their own flashcard by accessing images from online, be able to listen to the new words and also see animations of their stroke order.

Gamification has shown a direct correlation to a user's engagement to language learning, and thus their motivation and consistency. Learners should undergo a diverse set of exercises that uses as many senses as possible, such as visual, audio and kinetic simulation.

My app will therefore contain exercises that will use all aspects of language (speech, writing, reading, translating), broken down into small, gamified lessons.

Lastly, learners mentioned their struggle with staying consistent with their learning. To mitigate this, the app should include features such as streaks and progress tracking of how many lessons they have completed that day.

Overall, the app would be split into 5 different features:

1. lessonoverview: this feature deals with obtaining all the videos you want to study from and displaying the transcripts and keywords to the user.
2. makereviews: this feature deals with creating new flashcard and updating flashcard images and notes.
3. spacedrepetition: this feature deals with obtaining all the words to be reviewed that day, updating all the reviews and application logic dealing with the game exercises.
4. useroverview: this feature deals with calculating streaks and tracking a user's progress for that day.
5. youtubeintegration: this feature deals with showing the YouTube player and UI aspects for searching for YouTube videos.

Lessonoverview feature example

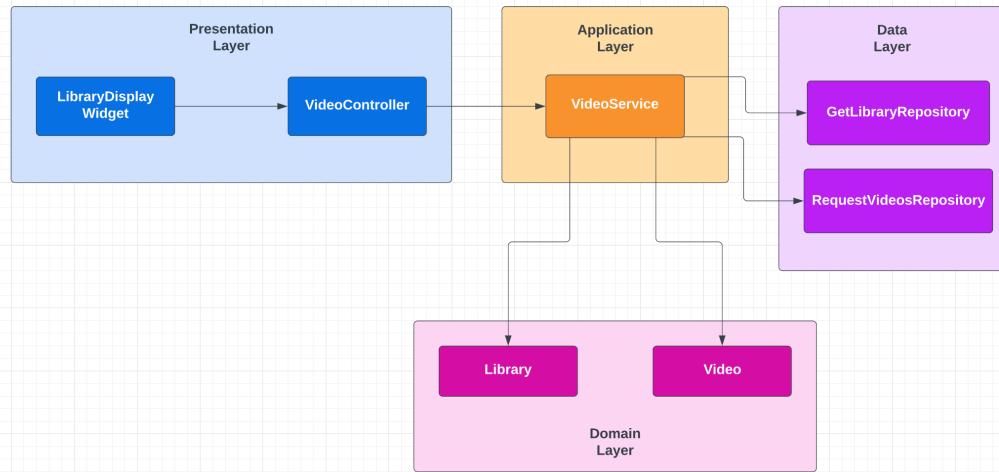


FIGURE 12. Example of lessonoverview feature

Riverpod providers wrap around these services, which controllers in the presentation layer access. This is so that widgets in the presentation layer which use certain controllers can listen to state changes and automatically refresh the interface.

For example, the VideoController accesses methods in the VideoService to obtain the previously processed videos. When obtained, this gets put into a Library model (a list of all the videos they have previously processed), which Riverpod watches for any changes for. If any changes are detected, the widget using this controller will also be updated.

Riverpod is not only a state management framework, but also useful for performing reactive caching to easily update the UI and helps developers through catching programming errors at compile-time rather than at runtime [51].

8.4. MVP.

Now that the backend has been implemented, the next steps are to create an Minimal Viable Product (MVP) to use for user feedback. We can assess the MVP design using Nielson's 10 usability heuristics [52], which consists of:

1. Visibility of System status (the deisgn should always keep the users informed of what is going on, through appropriate feedback)
2. Match between the system and the real world (words, phrases and concepts should be familiar to the user)
3. User control and freedom (users should have a clearly marked ‘emergency exit’ to leave unwanted action without hassle)
4. Consistency and standards (the app should follow platform and industry conventions)
5. Error prevention (best designs carefully prevent errors in the first place)
6. Recognition rather than recall (minimise user’s memory load by making the elements and options visible. i.e. help in context, rather than giving a long tutorial)
7. Flexibility and efficiency of use (shortcuts may speed up the interaction for the expert user. Examples are keyboard shortcuts, touch gestures and customization)
8. Aesthetic and minimalist design (interfaces should not contain information that is irrelevant or rarely needed)
9. Help users recognise, diagnose and recover from errors (error messages should be expressed in plain language and constructively suggest a solution)
10. Help and documentation (perhaps provide some documentation to help users understand how to complete their tasks)

From Figure 13, the original MVP design we can already see the app lacking in many areas. Regarding heuristic (1), the home page with the animated gif holding a review sign is actually a pressable button to show a user’s review lessons for that day. This is unlabelled and not immediately clear. The home page also does not contain a clear functionality that can instantly tell the user what they can do. This is also seen in the lesson overview page, where despite having a box with a camera icon and another text box to add a personal note, interviews have shown that users still get confused with the meaning of these boxes.

Original MVP

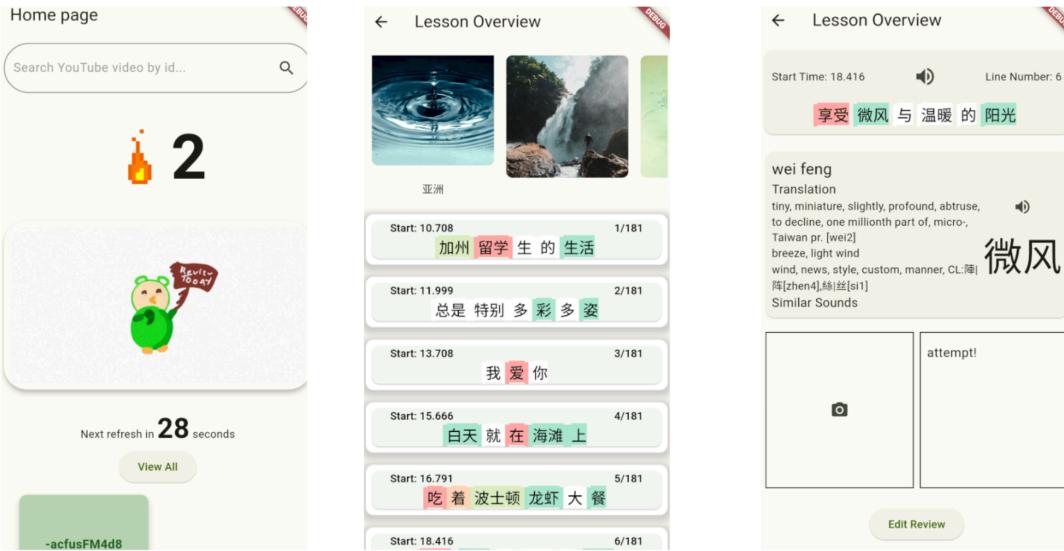


FIGURE 13. MVP design

Regarding the second heuristic, the words and phrases may be unfamiliar to the user. The main issue of the app is that it assumes that the user knows what to do. As an example, the rightmost page supplies the pinyin (pronunciation), translation and similar sounds to that word, but this is not clear due to the lack of labelling.

(3) is also a problem because there is no instantaneous escape route for a user when they accidentally follow a certain navigation path. To get back to the home screen, the user simply has to constantly press the back button. Sometimes, the back button in the app bar does not even exist. Thus, a navigation bar would be a suitable solution to this issue.

In the search bar widget (Figure 14) in the home page, (when searching for a new YouTube video to study from), the pop-up shows a widget with a YouTube thumbnail, name and channel.

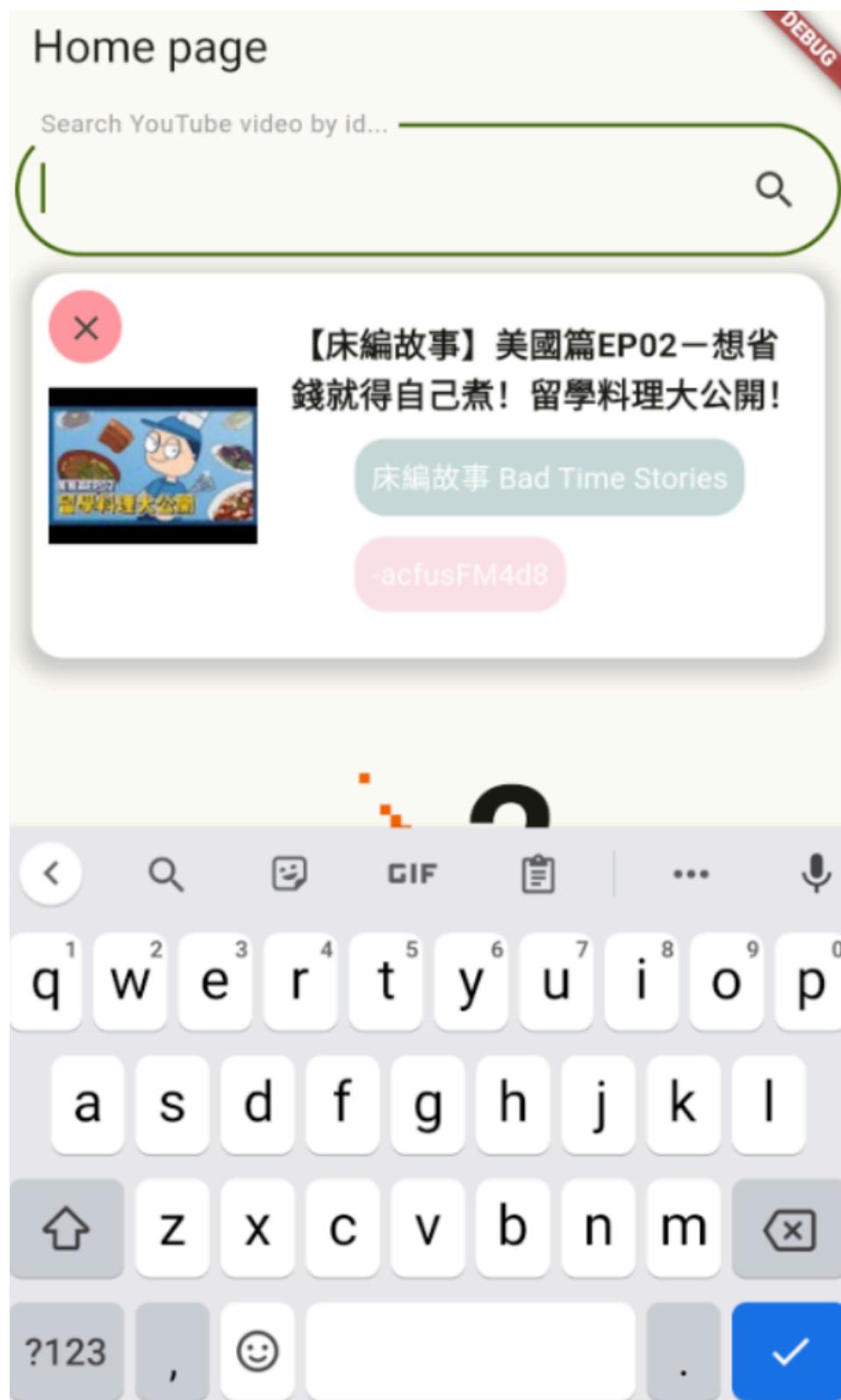


FIGURE 14. Initial video search

However, the exit button is on the top left, which goes against convention (the exit button is normally seen on the top right). This design therefore goes against Nielson's 4th heuristic, which emphasises the importance of consistency and standards. Building on top of this, there is also a lack of exit buttons in general. This means that when a user hits an error, it is difficult to navigate away from this, leading to user frustration. Overall, this impacts Nielson's 5th, 9th and 10th heuristic.

Finally, for the 6th and 7th heuristic (minimising user's memory load by making options visible as well as introducing shortcuts to speed up user interaction), more icon buttons can be used. It is important that these icon buttons are big enough to be visible, but also fit the UI aesthetic (Nielson's 8th heuristic).

Taking this a step further, I also wanted to hear what other students had to say. With this MVP, I conducted semi-structured interviews and conversations.

Most of the responses mentioned the lack of an intuitive UI, such as certain actions not being clear, widgets being too clustered and the lack of a uniform structure and colour scheme. However, they enjoyed the game aspect and the ability to download transcripts from YouTube to study from, mentioning how in certain apps today there is an inability to learn from online articles and content.

something duolingo lacks is prob ability to learn and practice something u see on internet
Heart 1
maybe a bar to copy and paste and then u revise off an article

FIGURE 15. Lack of utilisation of online resources

In the next iteration of the UI design, I drew from online inspirations and apps that I use personally and created inspiration boards. By using these ideas, I would further enhance my UI, re-iterateively conduct interviews and improve upon it. One example is shown below, where I used one interesting design to re-implement the creation of a flashcard page.

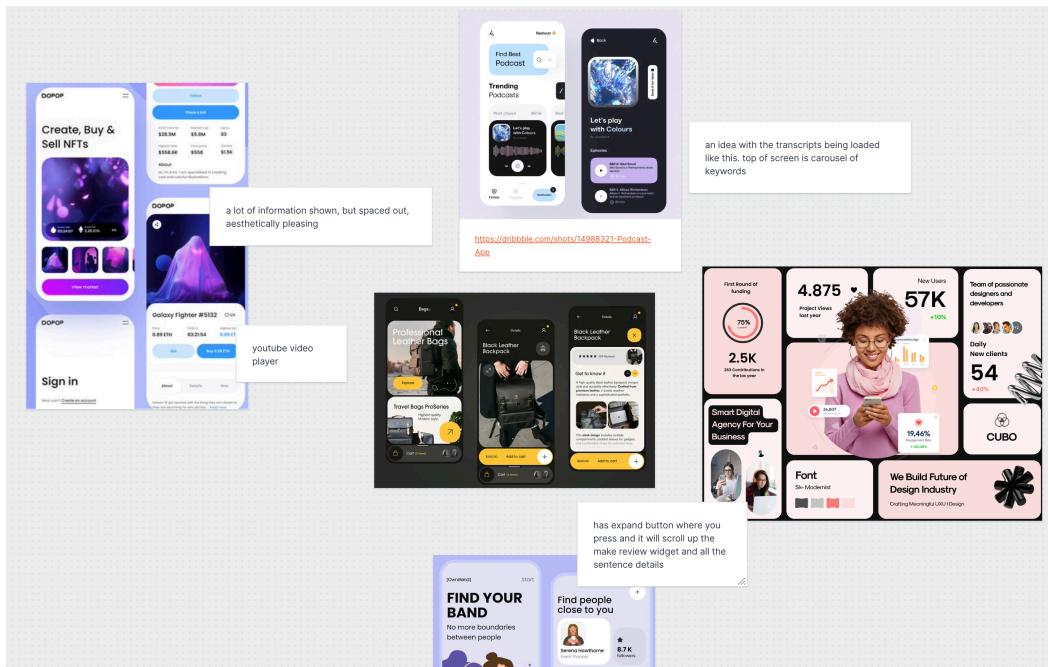


FIGURE 16. Inspiration board example

Make review page (old)

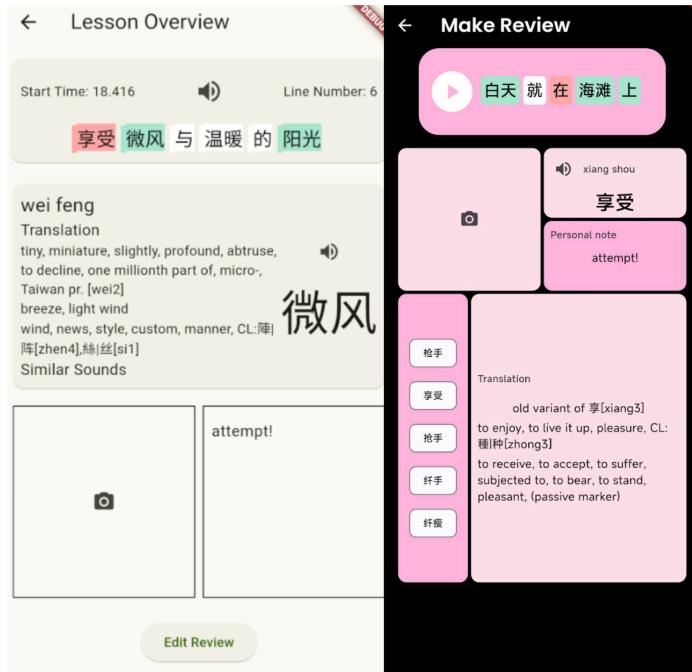


FIGURE 17. An iteration of make review page

Although the colour theme has improved, feedback included the lack of a focus on the page. The eye is not immediately attracted to any area of the screen, and it is not easy to know where to start to look. This also highlighted the importance of balancing usability and aesthetics.

Another critique was the lack of headers and instructions on the page. For example, on the left-most side is a list of similar sounding words, to help users with pronunciation (by identifying similar sounding words, users can be more aware with common pitfalls and mitigate further mistakes in pronunciation down the line). However, from a user perspective, this just looks like a list of random words. It would be best to label what this is exactly.

In the same interview, I asked them which app's design they liked the most, and why. They answered with Headspace's UI design (Headspace is a meditation app).

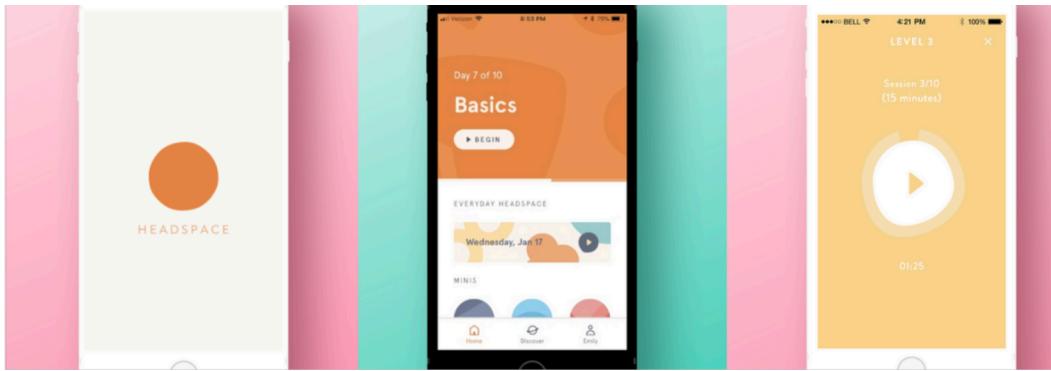


FIGURE 18. Headspace UI design

More specifically, the UI is spacious and calm to look at, which fit their branding. This can be seen in contrast to my initial home page design, where there were too many functionalities in one page. This creates a similar issue to Figure 17, where there is no particular focus on a page, leading to bad usability.

From this conversation, I concluded that each page should lead a user down a particular path through the app and be intuitive. It should follow common practices such as having the close button on the top right rather than the top left, as seen in Figure 14.

Home & Video page

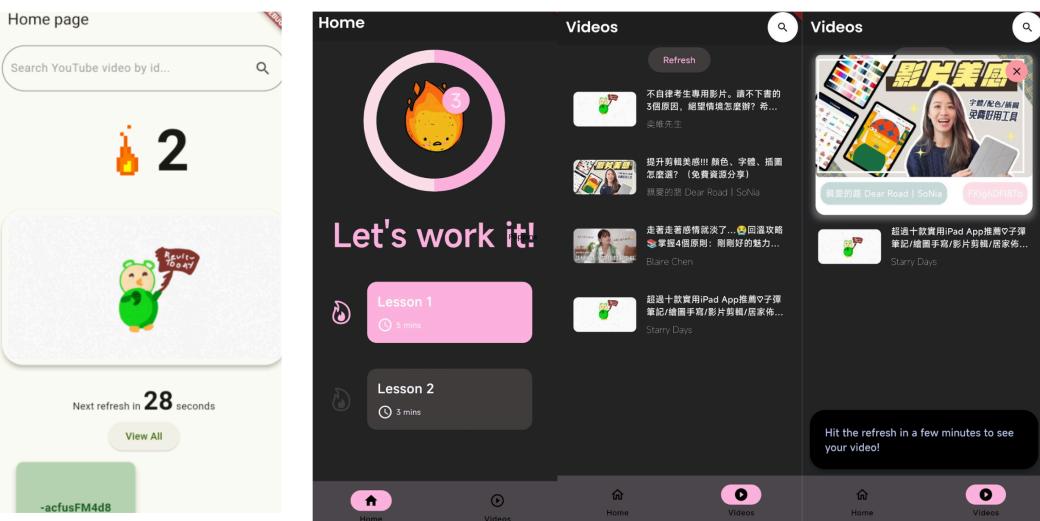


FIGURE 19. Home and Video pages

On the left hand side is the initial home page design. This consists of a search bar functionality, the streak number, a card widget for all the flashcards to review today, and finally the pre-downloaded transcripts of searched YouTube videos.

When the user searches a video and selects one they want to study, this will be shown at the bottom of the home screen. In the first MVP, you may notice a countdown. Every 30 seconds, the frontend polls the server. It is implemented this way because obtaining the transcript of the YouTube video, running the Stanza NLP model on it, querying APIs to

get the most critical words from the transcript and their respective image links, is not instantaneous. In the later iterations, this was changed for the user to manually press the refresh button only when they expect a new video to be downloaded, reducing the server load.

Additionally, the app has been revamped to contain navigations to a home page and video page, instead of cluttering all the information together. This separates out the different features of the app and allows each page to have a certain focus. From conducting interviews with users, it was suggested that the main focus should be the user's lessons to do that day, rather than all the videos in their library. Therefore, the home page now solely concentrates on a user's progress of lessons that day, their streak number, and all the exercises for them to complete.

Duolingo, as mentioned previously, utilises small, bite-sized lessons so that users can learn languages on the go and more conveniently. Inspired by this, each lesson shown on the home page only tests 5 new words, allowing users to do 5-minute lessons whenever they have time. Smaller chunks of learning has also proven to increase motivation and discipline. The large fire con at the top was also introduced in response to Nielson's 6th heuristic, giving the app a more decluttered look.

The right most image displays the new search widget after typing in the YouTube ID in the search bar on the video page. There is a larger thumbnail and the exit button is on the top right, following conventions and thus Nielson's 4th heuristic.

Transcript page

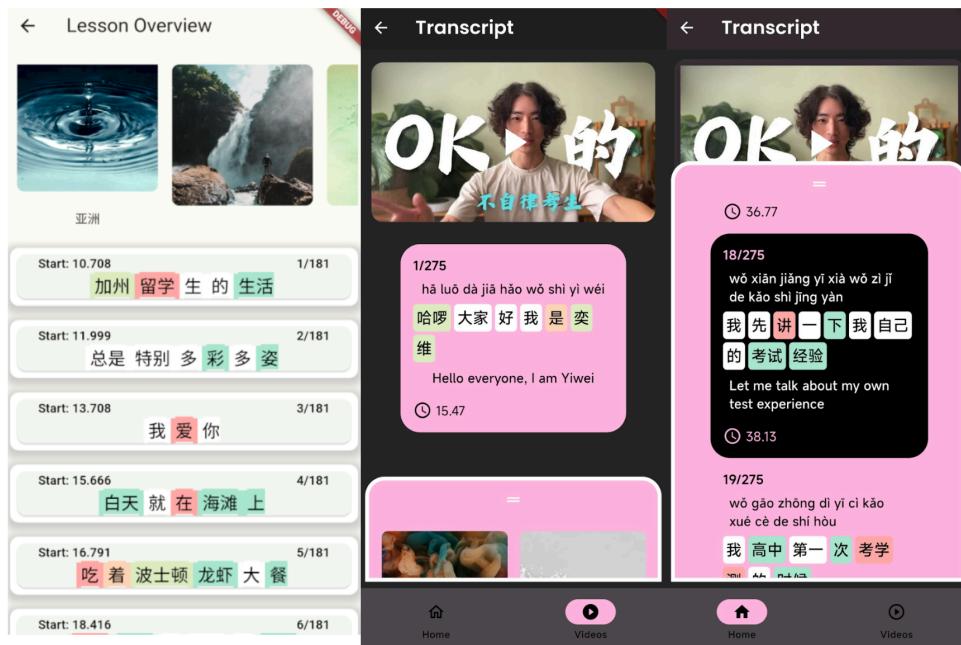


FIGURE 20. Transcript page

Figure 20 displays the full transcript of the video. Initially, this page just showed the keywords of the video with their respective images, then a scrollable list of all phrases from the video, with the times they are spoken (after ‘Start’) and the line number on the top right hand side. The highlighted words show their POS (parts of speech), such as whether it is a noun, verb, adjective etc.

Paper [53] discusses the approach that should be taken when developing course materials for technology-mediated Chinese language learning. Zhang Shenglan recommends that ‘for each chapter or unit, the learning objectives should be given at the beginning, so that students understand what is expected of them’. Since the app does not follow a course but the content of a YouTube video, this can be achieved through adding these keywords and their images.

By presenting these keywords before the transcript itself, it provides the most ‘pay-off value’ possible. These keywords can then be expanded upon in the transcript. At the same time, it provides context to the learner, depicting what the video is about, who are the characters in this video and who the video is for.

In the next few interviews, students discussed how it would be beneficial to also watch the YouTube video in real-time as they follow along the transcript. In my context survey, it was discovered that people learn from body language, not just from the words themselves. Being able to draw from visual contexts to decipher new words has proven to be effective in learning.

Thus, in the new iteration of the app, a real-time listener has been implemented which matches the transcripts to the video as it plays (see Figure 20, middle image).

From adding in this feature, the app now consists of visual and audio simulation. Rather than using a robotic voice (used by Anki and other language apps), students can learn the intonation and speech styles of their favourite content creators. This makes the learning experience more enjoyable and also provides a method for users to learn more native ways of speaking. As seen from the surveys, language learners value learning from relevant content rather than mechanistic textbook ways of speaking.

Along with this, a draggable scrollable sheet can also be seen peeking from the bottom. Exit buttons have also been incorporated into the app in case the user has navigated to this page accidentally, as well as gestures. By taking Nielson’s 7th heuristic (adding shortcuts) into account, users can easily scroll up to see the draggable widgets. Otherwise, these widgets are hidden at the bottom of the screen, which gives the screen more space and improves its overall aesthetic. As well as the keywords of the image, it contains the rest of the transcript, allowing the user to have the whole script at their fingertips instead of having to wait for the video to play at a certain point before seeing a particular phrase of interest. Moreover, this scrollable sheet allows users to jump to certain parts of the video, which is beneficial for the shadowing technique explained previously.

In later iterations, we can see that each phrase also contains the full pinyin (with their tones rather than just the romanized English characters) and a translation beneath. These add-on features are achieved through the Google Translate package and Pinyin package that Flutter offers.

Many YouTubers that teach Mandarin online include the Chinese subtitles, their pinyin and their direct translation in their videos. This has proved to be very beneficial to the

community, as sometimes words in the sentence may be misheard and thus learned with the wrong pronunciation. With the pinyin shown immediately to the user, we can mitigate this risk. Furthermore, having a direct translation beneath the phrase is also extremely beneficial, as it strips away the need for the learner to navigate away from the app, translate the whole sentence, and come back on the app. Previously, each word in the sentence had to be translated independently, however this takes a lot of time if the whole sentence is made up of new words; users are more likely to be discouraged from learning mandarin and would quit the app.

Each of these phrases belong to pressable card widgets, enabling users to create their own flashcards.

Make review page (new)

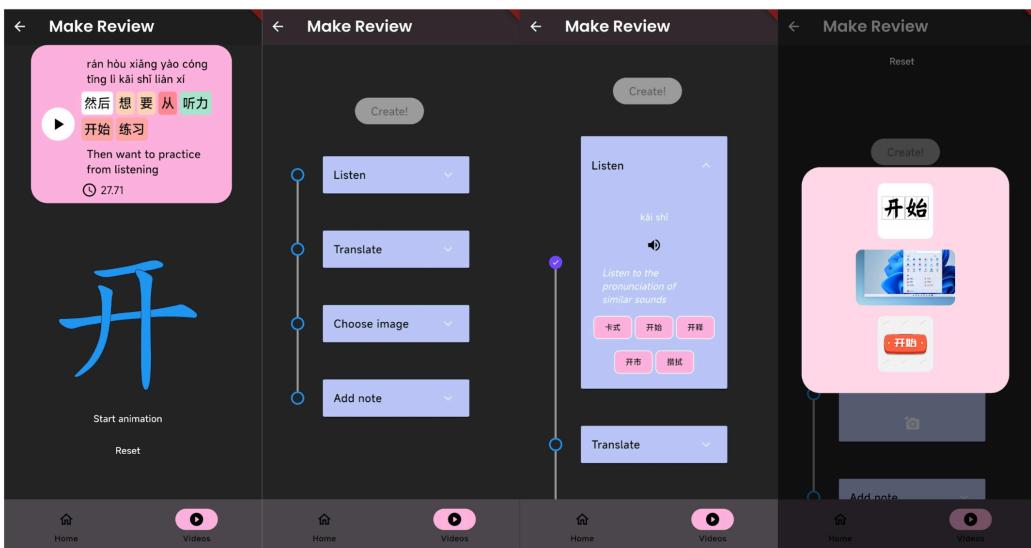


FIGURE 21. New review page

When one of these words get clicked, a user journey is created for a user to create a flashcard. This flashcard will save the word the user wants to review, as well as the sentence it is a part of, to keep its context information.

An improvement from the previous iteration includes the introduction of a stroke-order animation. This idea came from an interview with a beginner-learner of chinese. Stroke-order is incredibly important in the chinese language, but is easily overlooked. Adding animations to the UI also enhances the user experience by making it more fun.

In fact, the inclusion of multimedia has been shown to enhance learning [54]. Multimedia is a combination of more than one type of media such as text, symbols, images, pictures, audio, video and animations usually with the aid of technology for the purpose of enhancing understanding or memorisation. In the same paper, the use of multimedia was summarized to provide the following benefits:

1. Ability to turn abstract concepts into concrete concepts

2. Ability to present large volumes of information within a limited time with less effort
3. Ability to simulate students' interest in learning

The paper is based on assumptions are that learners have many channels to separate visual and auditory information, and each channel has a limited load capacity. Thus, multimedia is beneficial as it separates the load from one channel, such that the learners are not overwhelmed from too much information.

From the semi-structured interviews conducted as well as the initial surveys, we can already see how impactful animations and listening to sound pronunciations are. These features stood out especially which, from my findings, can explain why Duolingo is much more popular than its competitor, Anki.

Furthermore, following Nielson's 8th heuristic (an aesthetic and minimalist design), I have improved the user journey when they create a flashcard (see Figure 21, 2nd image).

By making this information linear, users can easily identify their next steps to create a flashcard. There is no information overload and users can decide to open and close certain expandable widgets. A checkbox at the side also marks the user's progress so they can track which widgets they have seen already.

Previously, users were prompted to take their own images to represent a word. However, many interviewers discussed the overhead with this approach, as they would have to find a relevant photo from their own photo gallery. Furthermore, some words simply are difficult to find an image for. Therefore, a new API call to Google Images has been used, where the top 3 images are shown to the user instead (see Figure 21, 4th image).

The user simply has to press on their favourite image; this provides a much more seamless feel to the app. Teachers that took the initial survey also mentioned their desire to incorporate Google into the app - they emphasized how utilizing these free resources can be very beneficial, and many language apps today do not use its advantage. By adding these Google Images we can use free resources online submitted by native speakers of Chinese, which can strengthen the memory of words in learners.

An example is if we think about learning the word for cat. The British idea of a cat is very different to the image of a cat in the East. When learning English, we already associate 'cat' with a certain type of cat. However, when learning a new language, we should associate this new word with a language that aligns more with their culture, as it can provide more context to the learner, strengthening connections in the brain.

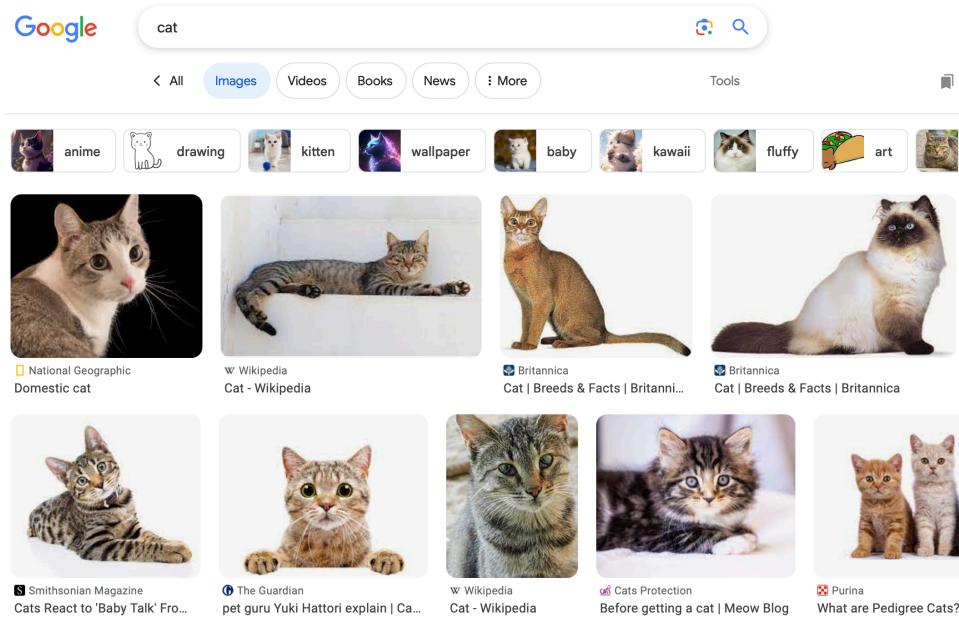


FIGURE 22. Cat (in english)

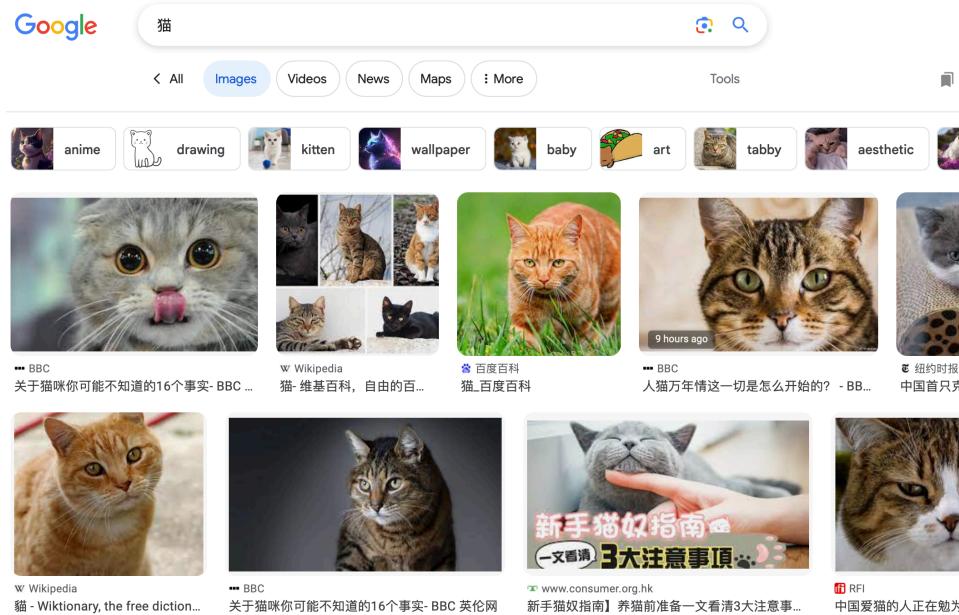


FIGURE 23. Cat (in chinese)

Although similar, we can already see the slight differences between the european cats and the cats when we search in chinese. Google Images also gives a much larger library of images that a user can choose from compared to their own personal photo gallery. Giving the learner the ability to choose their favourite image also means they spend more time thinking about that particular word, strengthening their connection with that word.

Lastly, a game functionality was added with five exercises:

1. Fill in the blank

2. Write the stroke order
3. Match the image
4. Translate the sentence
5. Speaking exercise

Game exercises

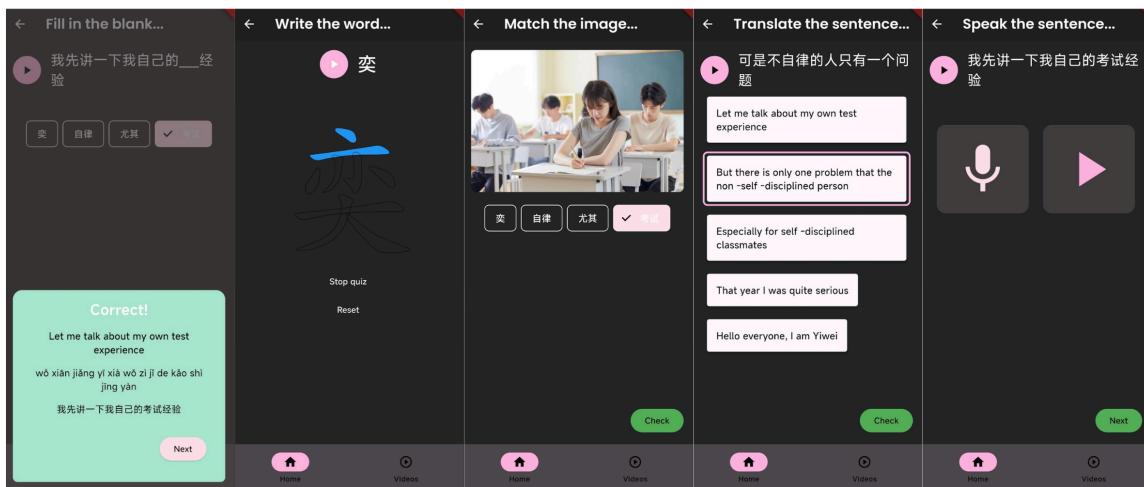


FIGURE 24. Games

Now that we have the ability to add a word, we must incorporate this into a spaced-repetition system.

According to Rigeney (1978) [55], learning strategies are the ‘actions, behaviours, steps, or techniques - such as seeking out target language conversation partners, or giving oneself encouragement to tackle a difficult language task - used by learners to enhance learning’.

To make the learning of the vocabulary effective, we can utilise previous research and design short, fun games for the learner that motivates and encourages them, even through mistakes. The exercises in the games will include using images and multimedia to help users link the information with pre-existing knowledge. The words will also be tested with sentences, to help users see the words in context and also help them analyze and classify these new vocabulary. Finally, the exercises will be challenging enough where learners are prompted to think critically before answering, for example by having translation and fill-in-the-blank exercises, where not all words are known by the learner.

The CoCAR model is also an interesting concept [56] that emphasises using action to enhance understanding. When designing the games, we therefore need to create engaging exercises that allows students to manage unknown situations. Achieving this can be through using multiple ways of testing, such as through visuals, sound, kinetic (testing stroke order) and understanding (translations). For deep understanding of vocabulary, this is achieved from the ability of users to add images and personal notes to their flashcards,

where users are prompted to think back to relevant experiences or images that remind them of that word.

Thus, I have distilled the learning aspect into 5 exercises. The first exercise is to fill in the sentence with the correct vocabulary; this exercise aims to teach the user how to use a certain word and where to put it in a sentence. In Figure 24, image 1, you can see the popup that shows when the user gets an exercise correct. This provides the user with pronunciation as well as the sentence's meaning, giving the user extra information to aid with their learning.

Exercise 2, testing stroke order, is an interactive method for learners to learn how to write the characters. This idea was brought up by one of the interviewees learning Chinese. They mentioned how difficult stroke order was and how they enjoyed this aspect of the language the most. Adding additional multimedia reinforces learning. Knowing the stroke order also ensures that the learner does not simply recognise the character, which is a common pitfall many Chinese learners fall for. Furthermore, writing Chinese characters is very complex and can lead to a lack of motivation. Especially for new vocabulary, students may feel lost while writing a complicated character. Therefore, this app also provides stroke hints when a certain number of wrong strokes have been calculated, to keep the student's morale high and enabling instantaneous self-correction.

In a study conducted in an elementary school Chinese immersion language program, students were given iPads to encourage writing [57]. Inputting Chinese were through the handwriting input method, allowing the students to improve their own Chinese writing ability through writing stories. At the end of the study, the students showed a net positive attitude.

From our previous research, we already have seen the importance of creating stories to remember a certain vocabulary. This can be achieved by using the matching image to word exercise (Figure 24, 3rd exercise), where the image itself is a story of its own. The personal note that users put onto their flashcards also can help with memory retention.

Fourthly is translating a sentence, which allows a user to test their understanding. This is a technique seen in many flashcard applications, such as Anki.

Lastly is a speaking exercise. Apps such as Duolingo utilises AI to check the accuracy of a learner's pronunciation. When using this feature myself, I realised that Duolingo's AI allowed you to cheat to system, where you may pass the exercise despite pronouncing a few words wrong. Furthermore, the research we have conducted explains the importance of self-reflection in the language learning process. By recording your own self saying the words and comparing this recorded audio with the correct pronunciation, users can recognise their own mistakes themselves and they are less likely to make the same mistake again.

All these exercises utilises multimedia, which we have seen has proven to be effective for learning. In a study examining the use of multimedia on E-cards for students learning Chinese as a foreign language [58], it was revealed that the more visual and verbal a flashcard is, the better the retention. Oxford (1990) recommended the use of word learning techniques through creating mental linkages (an example being grouping words by themes), applying images and sounds and finally employing an action to help learn words.

Traditional flashcards fall flat where audio cannot be integrated. Furthermore, it cannot access the internet and use visual media online. The optimal flashcard and way to be tested

is therefore by testing with the sounds and images incorporated with that word. This is why we have such exercises as shown above.

9. EVALUATION

9.1. Student feedback.

This app was reviewed by 37 students at the University of St Andrews, coming from a variety of backgrounds. The process was as follows: students were given a quick demonstration of my app, and then were asked to fill out an online survey using Qualtrics.

The aim of the survey was to gather feedback on the app's usability, user interface (UI) and user satisfaction, as well as gather suggestions for future improvements. The first half of the survey was a series of questions to gather a student's previous language learning experience; the second half was a series of questions to gather feedback on the app.

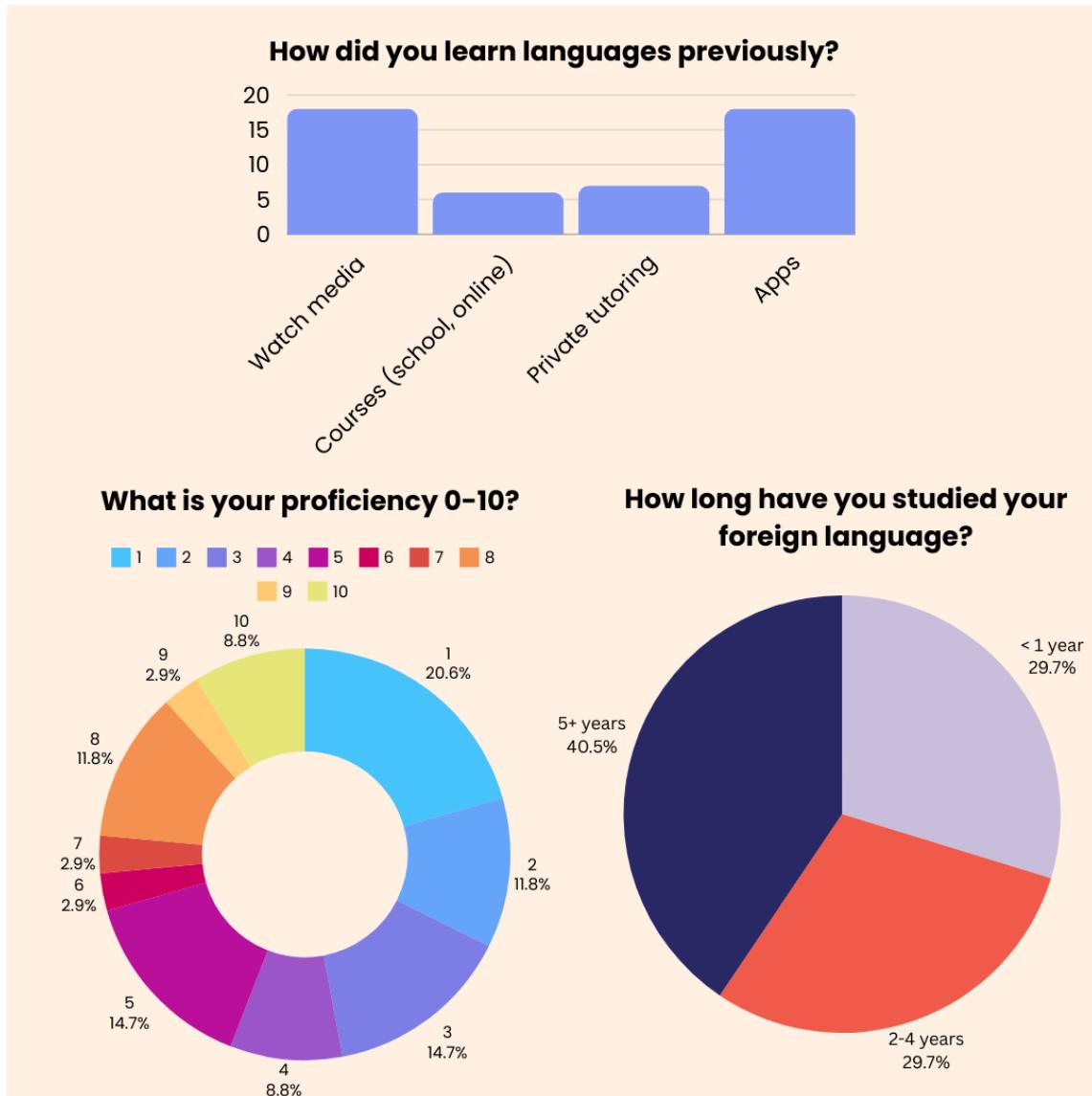


FIGURE 25. User previous language learning experience

In Figure 25 we can see that the majority (40.5%) of students have studied their foreign language for over 5 years, yet only 8.8% of the students ranked their language proficiency

as ‘fluent’ (a 10/10). In this scale, 0 means that they cannot speak the language at all, 5 means they can hold a conversation with a stranger on the street, and 10 means they can understand complicated concepts in that language. The majority of the students surveyed ranked their proficiency as between 1-5, which is a beginner to intermediate level. This is the demographic that my app aims to target. Additionally, the students in question primarily used study techniques such as apps and watching media online.

Next, the students were asked about their favourite language learning methods.

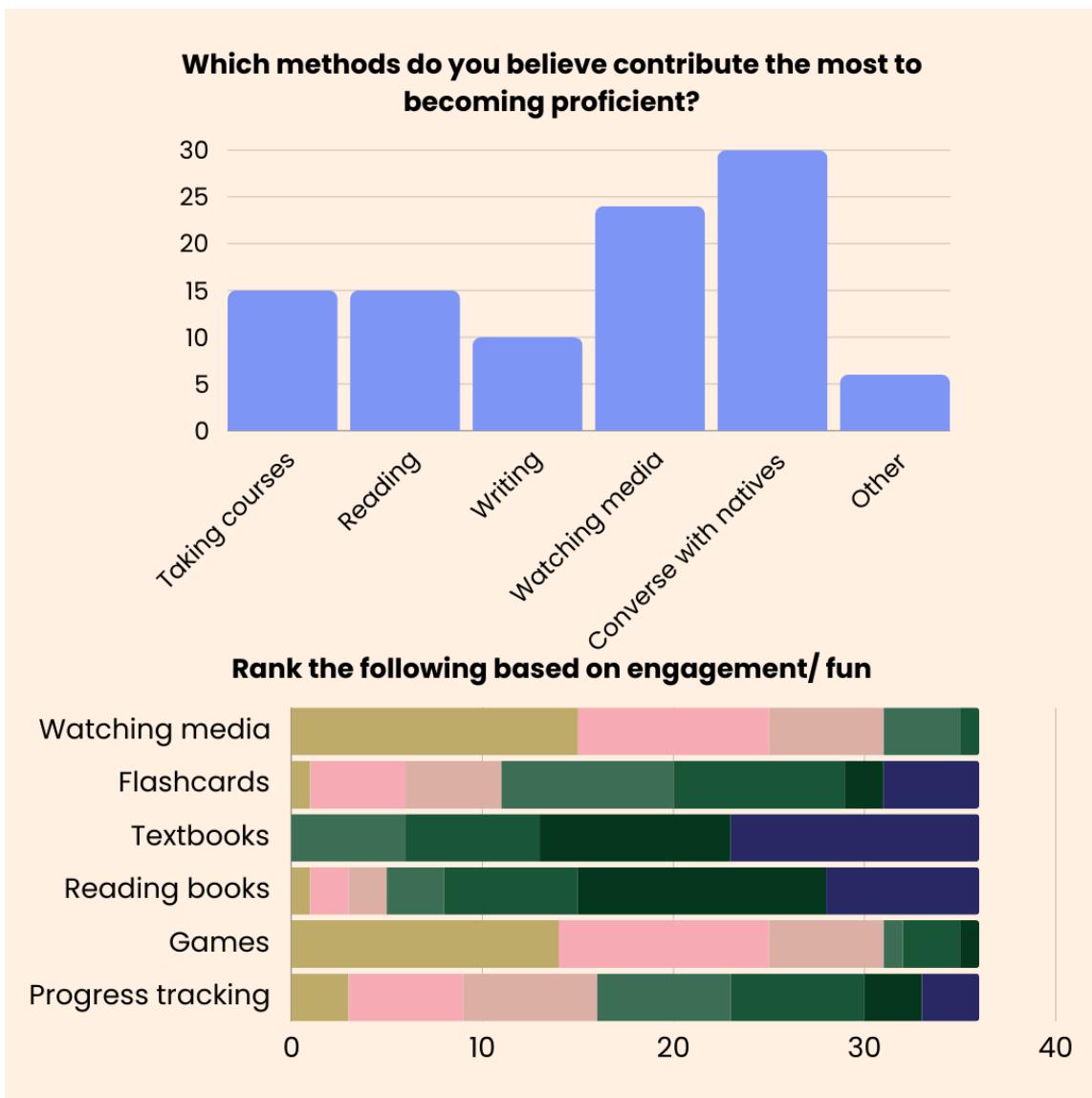


FIGURE 26. Favourite language learning methods

The majority of students believed that conversing with native speakers was the most effective method of learning a language. This was followed by watching online media, such as YouTube videos, which aligns with the features of my app, such as the ability to watch YouTube videos and shadow the native speaker. Students who selected ‘other’ also mentioned listening to music in that foreign language, going to the country where the

language is spoken and having motivation to learn that language. This goes hand-in-hand with the idea of going to the country where the language is spoken, as surrounding yourself in a culture with that language can keep motivation high.

Motivation also stems from being engaged and having fun when undergoing language learning. Therefore, I asked students to rank the methods of language learning based on engagement and fun. Watching media was ranked the highest. Textbooks, on the other hand, were ranked last, proving the point that students prefer to learn from relevant content and multimedia, rather than the static, mechanical teachings from a textbook.

Surprisingly, learning through flashcards was ranked mainly in the 4th and 5th place, which is lower than I had expected. Despite the book ‘Fluent Forever’ [8] pushing flashcard learning as one of the most effective methods of learning, it is clear that students do not feel as engaged compared to other methods. To investigate further, I asked the students about how often they used flashcards in their language learning.

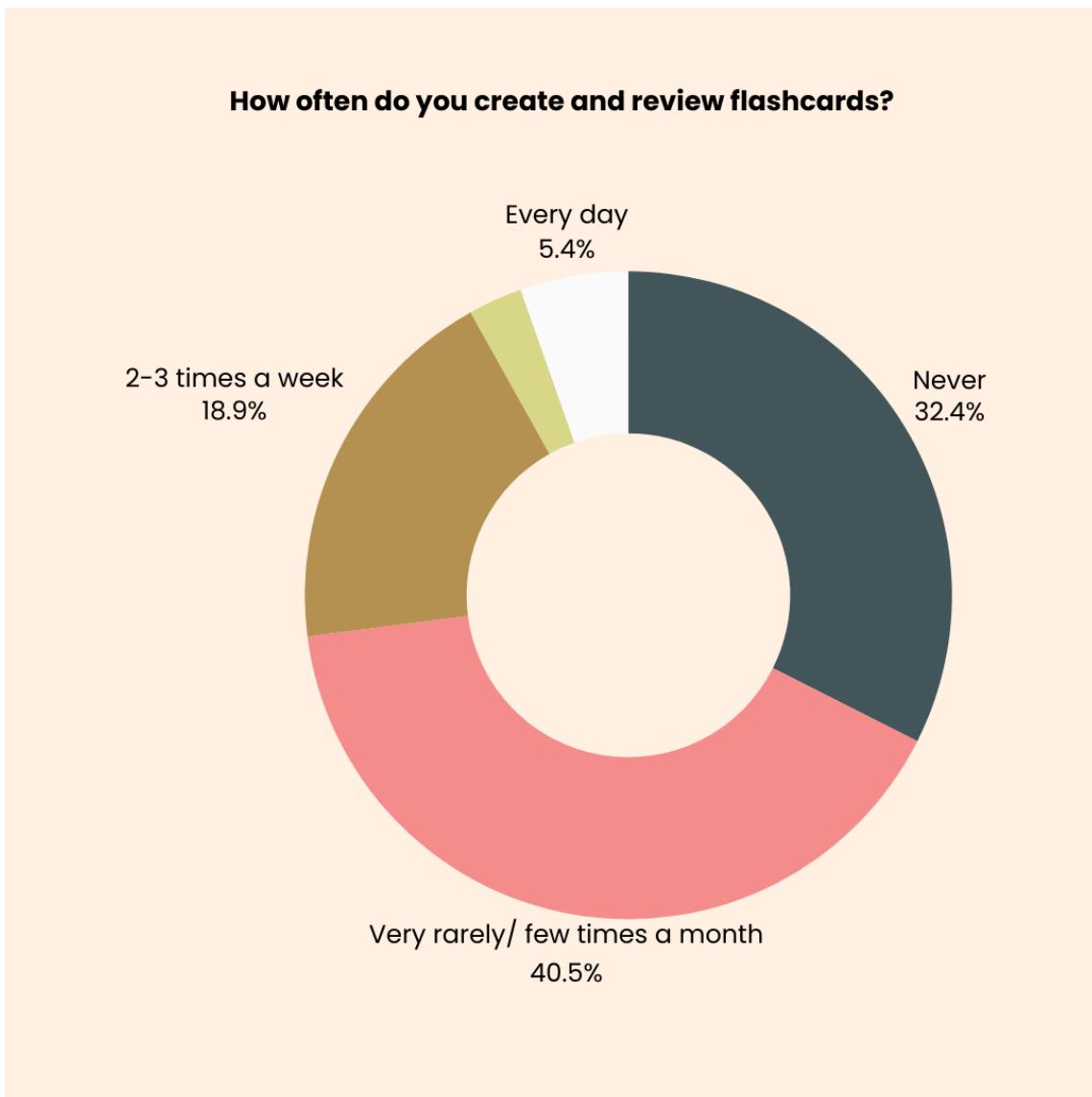


FIGURE 27. Flashcard usage

Figure 27 displays that the majority of students do not use flashcards at all, and if they do it is extremely rare (40.5% of the students selected ‘very rarely/ few times a month’). Of the students that did use flashcards every day, they mentioned that they used Anki, a popular flashcard app. However, there has been much discourse on the engagement factor of Anki. Anki can be seen as quite a mechanical way of learning; students mentioned that Anki was ‘boring’. The students who did enjoy Anki mainly mentioned its effectiveness rather than how fun it was.

This is where my app comes in. My app aims to make flashcard learning more engaging and fun. The flashcard contains multimedia, such as images and sound, and students get tested through exercises utilising images, sound and animations, which is in contrast to Anki where students simply flip a card to see its translation.

However, to see the effectiveness of this approach, the second half of the survey was a series of questions to gather feedback on the app.

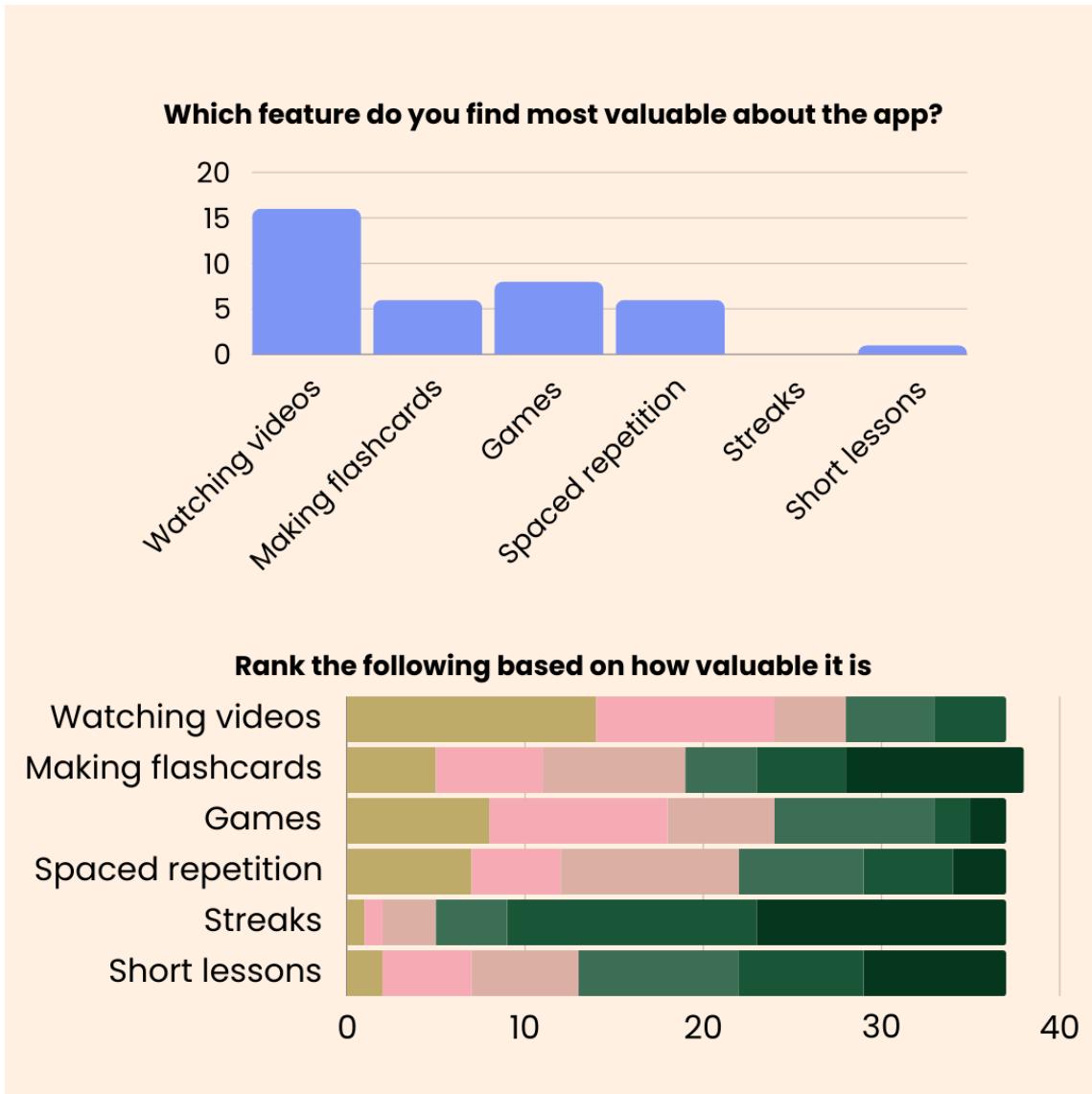


FIGURE 28. Favourite app features

Straight away we can see that students enjoyed the watching videos feature the most. Of those who commented, they mentioned how it was an ‘innovative’ and ‘fresh’ idea, and how it was the most ‘engaging’ feature if the app. Another student mentioned how the ability to have the transcription and translation at the same time meant it saved them time from having to look up words in the browser.

Games were generally ranked second highest, getting comments such as ‘I like games’ and ‘GAMES ARE COOL’, re-emphasising the importance of making learning fun. Whilst students were ranking the app features, many students struggled to decide the order between games and the short lessons. Those who ranked short lessons (where each ‘lesson’ only tests 5 new words) before games mentioned how it was ‘more realistic for busy public users’, explaining how much more practical it was.

Finally, those who ranked creating flashcards first mentioned how easy it was and how they could ‘click a word to add notes directly’. This is a feature that is not present in Anki, which gives an empty template to the user and expects them to fill in the information themselves. For students who are not aware of how to utilise flashcards and make it as engaging as possible, they may create a flashcard empty of multimedia and context, leaving no personal relation to the flashcard, and thus are less likely to remember it.

It is also important to consider what would drive students away from using the app every day.

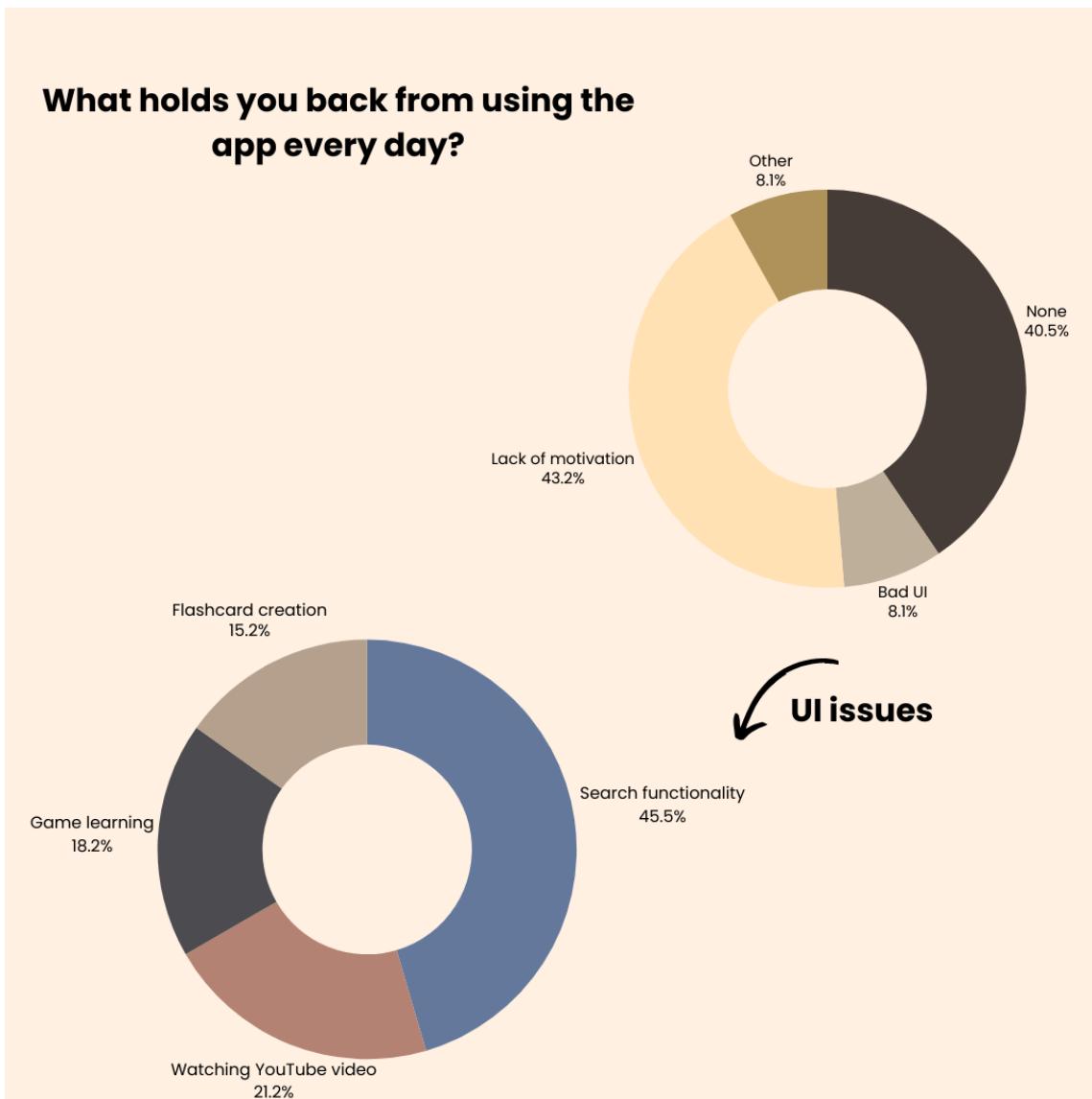


FIGURE 29. Least favourite app features

The vast majority of students (43.2%) mentioned their own lack of motivation. They mentioned that ‘outside the app, there aren’t many reasons to use the language. A community of just people interested in the same language would be what I consider a primary source of motivation’. Similarly, another student wrote, ‘I want to be able to see

the progress of friends and other people!'. We can therefore see a relation between what motivates people - a community of people learning the same language.

Those who selected 'other' described their fears of being busy and falling behind. Another student mentioned how they did not want to overuse their brain. This is a common fear amongst students, and is why the app is designed to have short, 5-minute lessons. However, the app does not currently have a feature to support students who are falling behind. Due to the spaced-repetition algorithm, students who fall behind will have to review more words the next day, which can be overwhelming.

However, I wanted to investigate how students found the user interface of the app. This can be seen in the pie chart in Figure 29 (2nd chart). Most students discussed how the search functionality was laborious, as the students would need to know the YouTube ID of the video in order to download it. Initially, this was implemented in the case students could not type in Chinese and therefore could not search up the video. When this was brought up, students mentioned how they would rather search in English and have the app translate the search for them.

One interesting comment was how 'as a beginner, it is very hard to know what to even search in the first place'. Since we are searching YouTube videos, the user would need to have a background in watching Chinese YouTube to know what interests them. If random videos are shown to them, they may not be interested in the content and thus not be motivated to study from it. This would be a good feature to add in the future, where the app can suggest videos to the user based on their interests. As stated from another student, 'it would be very unlikely that I would know or be interested in YouTubers that speak in a language I am trying to learn. It would be nice if the app recommended videos related to the type of content I would be interested in, at least for starters to get me familiar with that area of YouTube'. Students also mentioned having 'a feature showing recommended search terms would be helpful because as a language learner you do not even know what to search. For example maybe we can search "cooking videos" in English and it would bring up the equivalent in Chinese.' Additionally, students mentioned they would like to see custom videos, instead of being limited to YouTube.

Students who mentioned the 'watching YouTube video' feature as their least favourite interface feature mentioned how although they could see the segmentation of the words, there is currently no way to identify what the different colours mean. It would be better, they suggested, to categorise the words into nouns, verbs, adjectives etc. They also found the layout of all the previously watched videos to be cluttered and unorganised, suggesting playlists to organise by categories, ratings, themes or dates. Some mentioned a bookmarking feature for videos they struggled with, so they can come back to it later. Others wanted better ways to track their progress of videos such as their watch time and how many words or flashcards they have made on that video, linking back to the idea of progress tracking increasing motivation. Lastly, when watching the video, students recommended having a feature that allowed automatic pausing of the video at every caption change to allow them to read the transcript at their own pace.

Regarding 'flashcard creation', many students pointed out that the stroke order animation could be difficult to see at times, due to the colour clash with the dark background. Furthermore, the app currently does not allow students to see all their previously made flashcards as the idea was to re-surface the flashcard at optimal times based on a spaced

repetition system through game-based learning. However, students mentioned how they would like to see all their flashcards in one place, as well as the ability to edit and delete them. In a previous iteration of the MVP, I had initially allowed for students to take their own images for the flashcards. This was later changed to query Google Images API and obtain the top 3 images for students to select from. It would be beneficial to have options for both of these features, as some students mentioned how they would like to take their own images for the flashcards.

Lastly, the ‘game learning’ feature was regarded generally highly. Students identified its similarities to Duolingo and expressed how they would like to see more Duolingo-like features such as earning experience points and introducing characters to the games. Some students wanted to see game aspects that leaned more towards competing with friends, such as every ‘accomplishment can give you foot soldiers - attack your friends!’ In a similar vein, students wanted time-based challenges to bring in a competition aspect to it. There were some discrepancies with this feedback, however, as others stressed they would rather have simple, manual flashcards rather than all the game features. Thus, a later iteration of the app would be a balance between these two, where the user can choose to have a more game-like experience or a more traditional flashcard experience.

In conclusion, the app was rated very highly, with all students rating the app a 6 or higher, and 24.3% rating the app a 10/10.

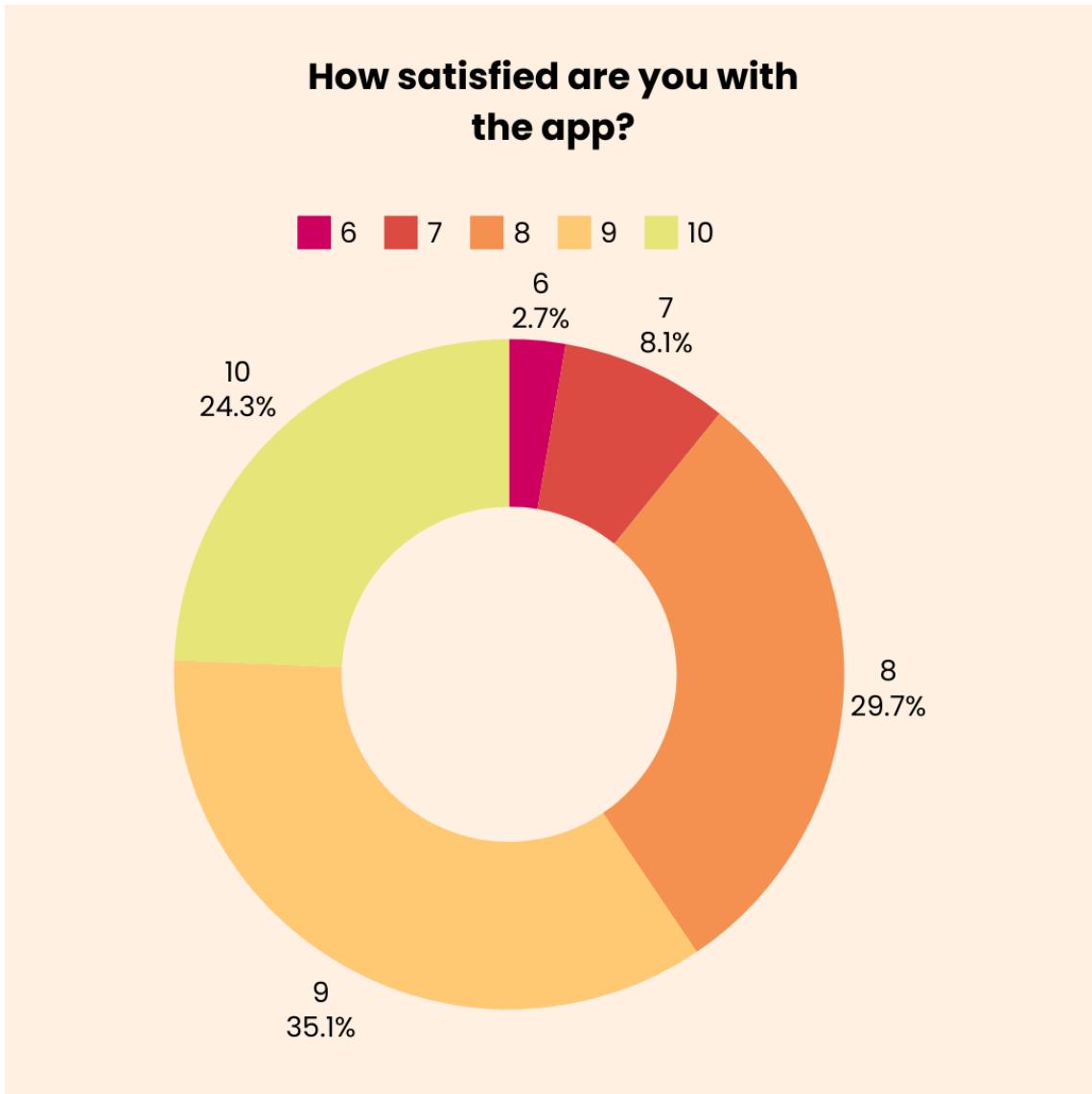


FIGURE 30. App Rating

Students who were interested in further development of the app answered the question ‘Are there any future features you would like to see on the app?’ with ideas such as a continuous assessment of the user’s vocabulary. This was interesting, as the student who mentioned this came from a background of language courses and did not use many apps. They discussed how schools would have small tests every week, and how this would be beneficial to have. Many other students’ ideas fell into the idea of social-networking, where they could make friends and have chat rooms to practice their language, create their own user profiles, have ranked leaderboard that compare your progress with friends. Additionally, it was mentioned from a few others to have more game-features such as having more characters and perhaps virtual reality (VR) integration.

9.2. Evaluation against objectives.

The original set out objectives required creating a minimal viable product of a language app, having the ability to generate transcripts and flashcards from YouTube videos as well as review questions relevant to the user, and finally creating a user evaluation form to obtain user feedback at the end of the project.

These objectives have been achieved, with the full-stack app being able to successfully query the YouTube API for a video with Chinese captions, transcribe and translate it with word segmentation, and create flashcards for each word a user wants to learn. The creation of a flashcard has been extended based on user feedback and personal research to include multimedia aspects such as images, audio and animations. To achieve this, the app has the ability to query Google Images to obtain the top 3 images for a particular word, for the user to select as their flashcard image. Audio has been integrated through text-to-speech Flutter packages and animations have been integrated to show the stroke order of a character.

The initial objective was for the app to generate review questions relevant to the user. This has been achieved through the creation of 5 exercises, which tests the user's understanding of the word through fill in the blank, matching the image, translating the sentence, recording their own voice to check their pronunciation and also testing their stroke order of characters by allowing users to write on the screen. Gamification techniques were also implemented to make the learning experience more fun, by the introduction of streaks and short game lessons where each lesson only tests a maximum of 5 words.

Furthermore, the app has been evaluated by 37 students at the University of St Andrews, and the feedback has been positive, with 24.3% rating the app a 10/10. Throughout the building process, interviews were conducted with students and teachers to gather iterative feedback. This feedback has been used to improve the app, such as the introduction of a real-time listener to match the transcript to the video as it plays and other UI changes.

9.3. Critical appraisal.

My app's uniqueness draws from the utilisation of YouTube videos and context-based learning, which is not seen in many language apps today. Combined with ideas from Duolingo, Rosetta Stone and Anki, the app is a mix of multimedia (Rosetta Stone), spaced-repetition (Anki) and game-based learning (Duolingo).

The strengths of my app comes from the incorporation of visuals through YouTube videos, images from Google to create flashcards and Chinese character stroke animations. Audio is also incorporated through videos and text-to-speech packages, facilitating better retention and recall. Furthermore, the spaced repetition algorithm optimises effective learning and the game-based learning increases a user's motivation and engagement.

However, there are many areas which the app falls short. As the app encompasses a mixture of elements from the different apps, it is not as strong in certain aspects as the individual apps. For example, the spaced-repetition algorithm may not be as optimised as Anki's implementation, and there is not a lot of user freedom for them to create their own flashcard. Anki, for example, contains features for learners to hide certain parts of an image and do labelling.

For multimedia learning, the app cannot ensure quality of the YouTube videos. As there are no professional Chinese speakers working on the app, the app cannot ensure that all of

the NLP segmentation is correct. Certain words may not be segmented correctly and there is currently no way for the user to correct this.

Regarding game-based learning, the app's user interface is not as polished as Duolingo's. The app does not have social networking aspects such as leaderboards and features to compete with friends. The app also does not have a feature to track the user's progress, such as how far through the video you have watched, or how many flashcards you have created, which is a big factor of motivation for certain users. Duolingo is also known for its characters and stories, which my app does not have.

On the other hand, the app is tailored specifically to language learners, compared to Anki, which is a general-purpose flashcard app. Anki does not prompt the user to create flashcards with audio, images and other multimedia aspects, which has proven to increase retention. Furthermore, the app allows users to quickly create flashcards from YouTube videos rather than from scratch, which speeds up the process of creating flashcards. At the same time, it allows users to create flashcards from authentic contexts, preparing them for more real-life situations. Combined with the short lessons for microlearning, the app is more practical for busy learners, who may not have time to sit down and study for long periods of time.

Since the users can choose which YouTube video to study from, the app can be tailored to the user's interests, which is an advantage to Duolingo and Rosetta Stone where the content is pre-determined. Potentially, the app could incorporate more social features such as virtual study groups and leaderboards, as well as more game-based learning features such as characters and stories. These features would also fit with the app's branding of learning off authentic content, as users can share their own real-life stories as in-game characters.

10. CONCLUSION

In conclusion, my app successfully achieves and exceeds all set out objectives. Through conducting personal research, interviews with language learners and more, I have been able to create an app that utilises effective language learning techniques to create a product that is engaging, fun and useful, as seen by the results from the final survey. The initial app has been improved through constant iterations from interviews with students and teachers, and the final product consists of features such as YouTube transcript syncing, obtaining flashcard images from Google Images, and a game-based learning system, on top of the original plan of obtaining a video's transcript and creating flashcards to review from.

Currently, the app is designed for one user. However, the app could be expanded to include a social networking aspect, as mentioned by the students. This would make the app more engaging and motivating, as we have discovered that a community of people learning the same language can help motivate people. It would also be interesting to incorporate large language models (LLMs) through the avatar of an in-game character to help with students who do not have access to a network of people speaking that foreign language. Future iterations of the app could bring in new game-features that help with building community, such as leaderboards and clans, to make the app more fun and boost motivation. Finally, the app could be expanded to include more languages, as the app is currently only designed for mandarin. Some students who took the survey mentioned how they would love to see the app in the foreign language they are currently studying.

11. APPENDIX

11.1. Tests.

Tests were split into 3 different types: backend unit tests that tested the API calls and their logic, tests that ensures the Flutter frontend handles API responses correctly, and finally end-to-end (E2E) tests that tested the complete application flow, which involves users interacting with the app.

Unit tests

The API tests were achieved using Postman, an API platform for building and using APIs. These tests were run on localhost and the logs were checked via DockerHub.

Test	Expected	Result	Pass
Test that the API call to get the YouTube transcript is successful with a valid YouTube ID. (Use endpoint '/vid'). The YouTube video has chinese captions	The server logs should show the YouTube transcript	Server successfully logged the chinese transcript	Pass
Test that word segmentation works on the YouTube transcript. (Use endpoint '/vid')	The server logs should show the segmented words	Server successfully logged the segmented words	Pass
Test that the segmented words can successfully obtain their pronunciation information, similar word information and translation information. (Use endpoint '/vid')	The server logs should show the pronunciation, similar words and translation of the segmented words	Server successfully logged the pronunciation, similar words and translation of the segmented words	Pass
Test that the YouTube transcript, when processed as above, is saved onto the database. (Use endpoint '/vid' to submit a valid YouTube video, then	The database should contain the YouTube transcript	Database successfully contains the YouTube transcript	Pass

check the database contains this through endpoint /getlesson/<videoid>)			
Test that the TextRazor API call is successful and obtains the lesson keywords for that transcript. Also seen from endpoint /vid	The server logs should show the lesson keywords	Server successfully logged the lesson keywords	Pass
Test the PyUnsplash API can obtain image urls for the lesson keywords. Also seen from endpoint /vid	The server logs should show the image urls	Server successfully logged the image urls	Pass

In the Flutter frontend, the following unit tests used MockHttpClient to test the logic without actually hitting the real endpoints. This is seen under the ‘tests’ folder in the Flutter app.

Test	Expected	Result	Pass
Test gets all videos from the server	When the server is called, the server should return a list of videos with response 200	Server returns a list of videos and has response 200	Pass
Test that the app can send a POST request to request a YouTube video	When a YouTube video is requested, the server should return a response 200	Server returns a response 200	Pass
Test that a 404 error is thrown if a YouTube video is not found	When a YouTube video is not found, the server should return a response 404	Server returns a response 404 for a particular YouTube video request	Pass
Test that data for a specific video is returned when a GET is sent	When a GET request is sent for a specific video, the server should return a	Server returns a response 200 with the video data	Pass

	response 200 with the video data		
Test that the app can send a POST request to create a flashcard	When a flashcard is created given the correct inputs, the server should return a response 200	Server returns a response 200	Pass
Test that the app can successfully send a POST request to update a flashcard	When a flashcard is updated given the updated note and image url, the server should return a response 200	Server returns a response 200	Pass
Test that the app can query all words to be reviewed today through a GET request	When a GET request is sent to the server, the server should return a response 200 with the words to be reviewed today	Server returns a response 200 with the words to be reviewed today	Pass
Test that the app can obtain the streak today from a GET request	When a GET request is sent to the server, the server should return a response 200 with the streak today	Server returns a response 200 with the streak today	Pass

End-to-end tests

These end-to-end tasks test a user interacting with the app.

Test	Expected	Result	Pass
Test that app only shows YouTube video if chinese captions are available	When the YouTube id of an english video is placed, app shows no results	App shows no results	Pass
Test that app only shows YouTube video if chinese captions are available	When the YouTube id of a chinese video is placed, app shows the video	App shows the video	Pass

Test that the app can successfully show the transcript of the video	When the YouTube id of a chinese video is placed, app shows the transcript	App shows the transcript	Pass
Test that the app can successfully shows the segmented words of each phrase in the transcript	When a video is selected, the words are segmented through the use of different background colours	App shows the segmented words	Pass
Test that the transcript page displays all relevant information	When a video is clicked, the lesson keywords and their images are shown, as well as the rest of the transcript	App shows the lesson keywords and their images, as well as the rest of the transcript	Pass
Test that the app allows for the creation of flashcards for each word in the transcript	When a word is clicked, the app allows the user to create a flashcard by displaying a page with the word stroke animation, its pronunciation, translation and similar words	App shows the flashcard creation page	Pass
Test that the app can successfully query the top 3 images from Google Images for a word when creating a transcript	When a word is clicked and a user wants to choose a flashcard image from Google Images, the app shows the top 3 images	App shows the top 3 images	Pass
Test that the app can add a day to a streak when a lesson is finished	Starting from a streak of 0, when a lesson is finished, the streak should show 1	Streak shows 1 when a lesson is finished	Pass
Test that the app can determine whether a flashcard must be created or be updated	When a flashcard is created, the app's button should turn to 'Update'	Button turns to 'Update' when a flashcard is created	Pass

Test that the flashcard can update its image and notes	When a flashcard is updated, the image and notes should change	Image and notes change when a flashcard is updated	Pass
Test that all 5 exercises are surfaced to the user in a game lesson	In a game, all 5 exercises should be shown to the user for a specific word	All 5 exercises are shown to the user	Pass
Test that the YouTube transcript works in real-time and different times of the video can be jumped to	When the user clicks on a phrase, the video jumps. The transcript also changes in real-time	The video jumps and the transcript changes in real-time	Pass
Test that the app successfully shows the word at correct intervals based on spaced-repetition	When a word is added, the word gets re-surfaced at different days based on the spaced-repetition algorithm	The word gets re-surfaced at different days based on the spaced-repetition algorithm	Pass

11.2. User manual.

The app's server is currently hosted on a virtual machine at the University of St Andrews, thus the app will only serve requests if the Flutter client is connected to the University's Eduroam. The app is currently not available on any stores, but can be run locally by downloading the github repository and running the Flutter client. The github repository is private and can be accessed by contacting the author.

To download the Flutter client when the repository is accessed, run the following:

```
cd frontend/workspace/flutterapp
flutter pub get
flutter run
```

11.3. Other.

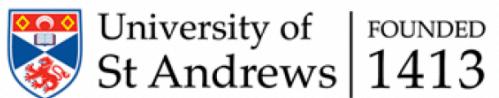
Progress

Date	Log
27/9	<ul style="list-style-type: none"> - Read papers on LLMs and made notes - Set up git repository and initial folders for API part
30/9	<ul style="list-style-type: none"> - Trying to set up again, using ASP .NET core → APIs to bilibili - Using podcast api instead of bilibili! (turns out must pay for transcripts)
8/10	<ul style="list-style-type: none"> - Flask set up YouTube endpoints - pip install opencc, pinyin to transform unicode to simplified and get pinyin
15/ 10	<ul style="list-style-type: none"> - More pinyin, word segmentation - timestamps to start of sentence
16/10	<ul style="list-style-type: none"> - able to get the sentences and then the translations and pinyin - word segmentation works! - word segmentation takes a long time; need to do some caching and use rabbitMQ perhaps
18/10	<ul style="list-style-type: none"> - Looking at rabbitMQ for better scaling of processing transcripts - Studying how to deploy to heroku with multiple workers
28/10	Integrated Nginx and Docker. Finished Fluent Forever book for literature review
05/11	<ul style="list-style-type: none"> - Introduced to "celery" which integrates rabbitMQ with docker. - Still issues with stanza, because it's such a big module and the docker hub stanza is only for german. - Can use TextRazor to get the most important words from transcript
19/11	<ul style="list-style-type: none"> - Wrote more literature review. - Set up NGINX but issues with Celery. - Trying to add some images using google search... issues - Able to incorporate the words with similar pronunciation (similar sounding words)
11/12	<ul style="list-style-type: none"> - Celery memory issue.
24/12	<ul style="list-style-type: none"> - Rebuilt architecture so now app is cleaner and uses micro-services rather than monolithic - Backend routes work for spaced repetition (SRS), etc!

26/12	<ul style="list-style-type: none"> - app context working with celery! the issue was because the worker did not have access to the db. the contextTask in backend was able to set up celery, but each of the workers needed their own Flask app instance, each with their own instance of the db instances (that all connect to the same hosted db!). - now issues with the json and submitting the video data.
27/12	<ul style="list-style-type: none"> - Expanding to not just youtube videos - also able to overwrite previous videos (deleting all reviews etc). - polling does not work. user will have to manually refresh/ reload.
28/12	<ul style="list-style-type: none"> - Flutter set up... Just with emulator.
29/12	<ul style="list-style-type: none"> - Small change — added source to videos, know if it comes from YouTube or somewhere else. - Designing app on Figma so then we can create presentation, application, data layer diagram
02/01	<ul style="list-style-type: none"> - Been working on the domain and data repositories of the flutter app. - Update user sentence works. - Testing works.
08/01	<ul style="list-style-type: none"> - Working on service providers and services, controllers, application and presentation layers. - Got initial UI for obtaining sentences of the transcript, and you can now go onto make review page (can't make review yet) but you can tap on every word and it'll initialise new flash card for you
12/01/24	<ul style="list-style-type: none"> - Game logic (had some issues with the states, now fixed. And also, can batch update lessons now!). - Have to update the db to have a "loading" value in querying videos...
13/01/24	<ul style="list-style-type: none"> - Added timer that would periodically check if next video is ready!
14/01/24	<ul style="list-style-type: none"> - Did streaks! - Did the whole process of update the streak whenever lesson gets completed (checks local storage so not a lot of requests to server) when adding update date. - Merged to main branch, so main all updated now!
16/01/24	<ul style="list-style-type: none"> - YouTube search bar, it now can query youtube id! there were some hardships with using YouTube api.
18/01/24	<ul style="list-style-type: none"> - UI changes. - Failed attempt at trying to host onto Digital Ocean

21/01/24	<ul style="list-style-type: none"> - On the 19th, 20th, been struggling with hosting backend onto digital ocean. Many bugs and turmoil, have emailed <redacted> for help. - Working on improving UI, and writing up dissertation and get design feedback
25/01/24	<ul style="list-style-type: none"> - Working on writing the report the past few days. - Requested for a school VM to host. - Updated UI such that it can sync with YouTube video!!
26/01/24	<ul style="list-style-type: none"> - Can see stroke order animation now!
31/02/24	<ul style="list-style-type: none"> - Updating UI, improving the make review page (with timeline), conducted the 'semi-structured' interviews, been writing up the dissertation
04/02/24	<ul style="list-style-type: none"> - Got the obtaining images from google working! (issues with state management however) - Added the animated search bar and also the whole app navigation
07/02/24	<ul style="list-style-type: none"> - Can connect to the VM! - Phrase translations to the sentences and also their pinyin on top. - Many bugs with making review, updating review... fixed now.
08/02/24	<ul style="list-style-type: none"> - Revamped the home UI so that it shows streak and also a progress indicator showing how many lessons you got done that day!
09/02/24	<ul style="list-style-type: none"> - Improved search functionality (search widgets, also connect to backend to obtain videos!)
10/02/24	<ul style="list-style-type: none"> - New games! (and also progress bar works with it). - Can update SRS now. - Added the stroke order game, fill in blank and also match image to word
14/02/24	<ul style="list-style-type: none"> - Working on bugs like sorting the transcript properly. - Writing up report - Incorporated skipping through the video in lesson overview
17/02/24	<ul style="list-style-type: none"> - Streak bug (gets wrong streak) - Bug with navigation to home page after exercises completed
17/02/24	<ul style="list-style-type: none"> - Streak fixed! had to change backend to sort by unique descending, and also streaks should be calculated from the day before (in case not studied today), THEN we check if studied today and then we + 1.
26/02/24	<ul style="list-style-type: none"> - Exercise to record voice and hear back working!
28/02/24	<ul style="list-style-type: none"> - Translation exercises are all done now! Coding is finished :)

FIGURE 31. Progress log



School of Computer Science Ethics Committee

08 November 2023

Dear Yue Ning,

Thank you for submitting your ethical application which was considered by the School Ethics Committee.

The School of Computer Science Ethics Committee, acting on behalf of the University Teaching and Research Ethics Committee (UTREC), has approved this application:

Approval Code:	CS17314	Approved on:	08.11.23	Approval Expiry:	08.11.28
Project Title:	Chinese Language App for Intermediate Learners				
Researcher(s):	Yue Ning Law				
Supervisor(s):	Angela Miguel				

The following supporting documents are also acknowledged and approved:

1. Application Form
2. Participant Information Sheet
3. Participant Consent Form
4. Participant Advert
5. Sample Questions

Approval is awarded for 5 years, see the approval expiry date above.

If your project has not commenced within 2 years of approval, you must submit a new and updated ethical application to your School Ethics Committee.

If you are unable to complete your research by the approval expiry date you must request an extension to the approval period. You can write to your School Ethics Committee who may grant a discretionary extension of up to 6 months. For longer extensions, or for any other changes, you must submit an ethical amendment application.

You must report any serious adverse events, or significant changes not covered by this approval, related to this study immediately to the School Ethics Committee.

Approval is given on the following conditions:

- that you conduct your research in line with:
 - the details provided in your ethical application
 - the University's [Principles of Good Research Conduct](#)
 - the conditions of any funding associated with your work
- that you obtain all applicable additional documents (see the '[additional documents' webpage](#) for guidance) before research commences.

You should retain this approval letter with your study paperwork.

Yours sincerely,

Natasha Bazeley

SEC Administrator

School of Computer Science Ethics Committee

Dr Olexandr Konovalov/Convenor, Jack Cole Building, North Haugh, St Andrews, Fife, KY16 9SX

Telephone: 01334 463273 Email: ethics-cs@st-andrews.ac.uk

The University of St Andrews is a charity registered in Scotland: No SC013532

FIGURE 32. Ethics approval

REFERENCES

1. Team, I. C. S.: How Agile Plays Into Software as a Layered Technology, <https://www.computer.org/publications/tech-news/trends/agile-and-software-as-a-layered-technology>
2. Lagerberg, L., Skude, T., Emanuelsson, P., Sandahl, K., Ståhl, D. D.: The Impact of Agile Principles and Practices on Large-Scale Software Development Projects: A Multiple-Case Study of Two Projects at Ericsson. 348–356 (2013)
3. YouTube: YouTube API Services - Developer Policies, <https://developers.google.com/youtube/terms/developer-policies>
4. Corballis, M.: The Origins and Evolution of Language, <https://www.youtube.com/watch?v=nd5cklw6d6Q>
5. P.F.Ferrari, L.Bonini, L.Fogassi: From Monkey Mirror Neurons to primate behaviours: possible "direct" and "indirect" pathways, <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2865083/>
6. Lonsdale, C.: How to learn any language in 6 months, <https://www.youtube.com/watch?v=d0yGdNEWdn0>
7. Sung, J.: Using “Interleaving” for Language Learning, <https://www.youtube.com/watch?v=ZFyPMRzPNTE>
8. Wyner, G.: Fluent Forever: How to learn any language fast and never forget it, (2014)
9. Gentner, D., Colhoun, J.: Analogical Processes in Human Thinking and Learning, https://link.springer.com/chapter/10.1007/978-3-642-03129-8_3
10. Gentner, D.: Mechanisms of analogical learning. Department of Computer Science, University of Illinois at Urbana-Champaign (1987)
11. Forbus, K., Gentner, D., Everett, J., Wu, M.: Towards a Computational Model of Evaluating and Using Analogical Inferences, https://www.qrg.northwestern.edu/papers/Files/CogSci97_analogical_inference_paper.pdf
12. Gentner, D., Markman, A. B.: Structural Alignment in comparison: no difference without similarity, <https://www.jstor.org/stable/40063088?seq=1>
13. L.Harlow, L., Muyskens, J. A.: Priorities for Intermediate-Level Language Instruction, <https://www.jstor.org/stable/329004?seq=1>
14. Rott, S.: The Effect of Exposure Frequency on Intermediate Language Learners' Incidental Vocabulary Acquisition and Retention through Reading. Studies in Second Language Acquisition. 21, 589–619 (1999). <https://doi.org/10.1017/S0272263199004039>
15. Weerasinghe, M., Biener, V., Grubert, J., Quigley, A., Toniolo, A., Pucihar, K. Č., Kljun, M.: VocabulARy: Learning Vocabulary in AR Supported by Keyword Visualisations. IEEE Transactions on Visualization and Computer Graphics. 28, 3748–3758 (2022). <https://doi.org/10.1109/TVCG.2022.3203116>
16. Mcleod, S.: Vgotsky's Theory of Cognitive Development, <https://www.simplypsychology.org/vygotsky.html>
17. Pawlak, M.: Investigating the use of speaking strategies: in the performance of two communicative tasks: the importance of communicative goal, <https://files.eric.ed.gov/fulltext/EJ1183957.pdf>
18. Team, M. T. C.: Ebbinghaus's Forgetting Curve, <https://www.mindtools.com/a9wjrw/ebbinghauss-forgetting-curve>
19. Eylem Koral Gumusoglu, A. T. &, Aydin, D.: Fostering Foreign Language Learning With Twitter: Reflections from English Learners, <https://files.eric.ed.gov/fulltext/EJ1165854.pdf>
20. Babbel: The Babbel Method, https://about.babbel.com/en_GB/
21. E., M. W. C., Kupper-Tetzl, Weston, T., S.N.Kim, A., V.Kapler, I., Foot-Seymour, V.: Enhancing the quality of student learning using distributed practice, <https://www.cambridge.org/core/books/cambridge-handbook-of-cognition-and-education/enhancing-the-quality-of-student-learning-using-distributed-practice/88D92B0BC94535226DB7B39C8026EF31>

22. Freeman, C.: The Duolingo Method: 5 key principles that make learning fun and effective, <https://blog.duolingo.com/duolingo-teaching-method/>
23. Team, R. S.: How we teach: Discover the Rosetta Stone Method, <https://www.rosettastone.co.uk/magazine/how-we-teach/>
24. Smiderle, R., Rigo, S. J., Marques, L. B., Miranda Coelho, J. A. P. de, Jaques, P. A.: The impact of gamification on students' learning, engagement and behaviour based on their personality traits, <https://slejournal.springeropen.com/articles/10.1186/s40561-019-0098-x>
25. Hanus, M. D., Fox, J.: Assessing the effects of gamification in the classroom: A longitudinal study on intrinsic motivation, social comparison, satisfaction, effort and academic performance, <https://www.sciencedirect.com/science/article/pii/S0360131514002000?via%3Dihub>
26. Fang, Q.: A Study of College English Teaching Mode in the Context of Micro-learning. Presented at the (2018)
27. Ghafar, Z., Abdulkarim, S., Lanya, Mhamad, M., Rezhin, Kareem, A., Pshtewan, Rasul, A., Tablo, Mahmud, I.: Microlearning As a Learning Tool for Teaching and Learning in Acquiring Language: Applications, Advantages, And Influences on the Language. Canadian Journal of Educational and Social Studies. 3, 45–62 (2023). <https://doi.org/10.53103/cjess.v3i2.127>
28. Kemp, S.: Digital 2022: Global Overview Report, <https://datareportal.com/reports/digital-2022-global-overview-report>
29. Milošević-Đorđević, J. S., Žeželj, I. L.: Psychological predictors of addictive social networking sites use: The case of Serbia. Computers in Human Behavior. 32, 229–234 (2014). <https://doi.org/https://doi.org/10.1016/j.chb.2013.12.018>
30. Ghouali, K., Benmoussat, S.: Investigating the Effect of Social Media on EFL Students' Written Production: Case of Third-Year EFL Students at Tlemcen University, Algeria. 24–39 (2019). <https://doi.org/10.24093/awej/Chief1.3>
31. Scholarly: Anki: Pros and Cons of Using Spaced Repetition Software, <https://scholarly.so/blog/anki-pros-and-cons-of-using-spaced-repetition-software>
32. Senzaki, S., Hackathorn, J., Appleby, D. C., Gurung, R. A. R.: Reinventing Flashcards to Increase Student Learning, <https://doi.org/10.1177/1475725717719771>
33. León, J., Núñez, J. L., Liew, J.: Self-determination and STEM education: Effects of autonomy, motivation, and self-regulated learning on high school math achievement, Learning and Individual Differences, <https://doi.org/10.1016/j.lindif.2015.08.017>
34. Dyouri, A.: How to Use Flask-SQLAlchemy to Interact with Databases in a Flask Application, <https://www.digitalocean.com/community/tutorials/how-to-use-flask-sqlalchemy-to-interact-with-databases-in-a-flask-application>
35. Leung, W. C.: Chinese Word Segmentation, <https://medium.com/@ching.achterwinter/chinese-word-segmentation-2e28feee87fe>
36. Stanza: Stanza, <https://stanfordnlp.github.io/stanza/>
37. Pypi: hanziconv, <https://pypi.org/project/hanziconv/>
38. Pypi: hanzidentifier, <https://pypi.org/project/hanzidentifier/>
39. Pypi: pinyin, <https://pypi.org/project/pinyin/>
40. Pypi: dimsime, <https://pypi.org/project/dimsime/0.2.1/>
41. TextRazor: TextRazor, <https://www.textrazor.com/>
42. Developers, U.: Py, <https://unsplash.com/developers>
43. Pypi: youtube-transcript-api 0.6.2, <https://pypi.org/project/youtube-transcript-api/>
44. Celery: Introduction to Celery, <https://docs.celeryq.dev/en/stable/getting-started/introduction.html>

45. Academy, C.: Advantages and Disadvantages of Microservices Architecture, <https://cloudacademy.com/blog/microservices-architecture-challenge-advantage-drawback/>
46. Docker: Docker Compose overview, <https://docs.docker.com/compose/>
47. Ezra, O., Cohen, A.: How teachers can guide students towards the increased use of contextualised MALL in target and non-target countries: The case of L2 Chinese adult students, <https://doi.org/10.14742/ajet.5015>
48. LYU, B., QI, X.: A review of research on technology-assisted teaching and learning of Chinese as a second or foreign language from 2008 to 2018, <https://link.springer.com/article/10.1007/s11516-020-0006-8>
49. Bizzotto, A.: Flutter Project Structure: Feature-first or Layer-first?, <https://codewithandrea.com/articles/flutter-project-structure/>
50. Hamada, Y.: Shadowing: What is It? How to Use It. Where Will It Go?. RELC Journal. 386–393 (2019)
51. Bizzotto, A.: Flutter Riverpod 2.0: The Ultimate Guide, <https://codewithandrea.com/articles/flutter-state-management-riverpod/>
52. Molich, R., Nielsen, J.: Improving a human-computer dialogue, Communications of the ACM, (1990)
53. Kubler, C. C.: Developing course materials for technology-mediated Chinese language learning, <https://doi.org/10.1080/17501229.2018.1418626>
54. MD, A., N, F., A, O., NT, S.-B., LA, O., OV, M., YO, I.-F., AO, F., AL., A.: Multimedia tools in the teaching and learning processes: A systematic review. Heliyon, <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7644889/>
55. Hardan, A. A.: Language Learning Strategies: A General Overview, <https://www.sciencedirect.com/science/article/pii/S1877042813048179>
56. Lan, Y.-J., Chang, K.-E., Chen, N.-S.: 1038CoCAR: An online synchronous training model for empowering ICT capacity of teachers of Chinese as a foreign language, <https://ajet.org.au/index.php/AJET/article/view/808/106>
57. Eubanks, J.-F., Yeh, H.-T., Tseng, H.: Learning Chinese through a twenty-first century writing workshop with the integration of mobile technology in a language immersion elementary school, <https://doi.org/10.1080/09588221.2017.1399911>
58. Li, J.-T., Tong, F.: Multimedia-assisted self-learning materials: the benefits of E-flashcards for vocabulary learning in Chinese as a foreign language, https://www.researchgate.net/publication/327739809_Multimedia-assisted_self-learning_materials_the_benefits_of_E-flashcards_for_vocabulary_learning_in_Chinese_as_a_foreign_language