Packle: A Discord Study Bot

Mariyam Yasmeen

S1800367 | 19020255

BSc (HONS) Computer Science

Villa College

Abstract

The ability to learn and retain information is a vital skill. If we wish to develop our minds and have a better understanding of the world around us, we must be able to acquire and retain information; retain being a key part of this process.

Retesting and spaced learning have been proven to be effective methods of retaining memories long term. This project explores the development of Packle to allow students to study, both individually and with their peers, by utilising these scientifically proven effective methods of learning. Packle will be implemented as a bot on Discord; a popular and free online platform which can be used as a powerful educational tool.

Acknowledgements

I would like to express my appreciation to my supervisor Udhma Latheef for her help and guidance through every stage of this project.

I would also like to extend my deepest gratitude to Sam and Jim for mentoring, inspiring, and motivating me. I am also grateful to my friends Nahy, Shaffan, and Ziaan for their encouragement and feedback.

I would also like to thank my family for their support and the cats for the comfort they offered me through this project.

Table of Contents

Table of Figures

Table of Tables

1. Introduction

While there is still much to be understood about the human brain and memory, learning is said to occur when information flows from short-term to long-term memory, while leaving a lasting trace. A reversal of this flow happens when we try to recall learned information (Martinez, 2012). We are extremely dependent on our ability to learn and recall information since this allows us to understand the world around us. With knowledge being an important part of our existence, it is crucial to identify and hone the skills that allow us to learn and remember key information effectively over a long period of time; one proven method being the use of retesting and spaced repetition.

This report will highlight the development of Packle; a Discord bot which will allow students to use retesting and spaced repetition to study. Section one will highlight the scope and objectives of Packle. This will be followed by the research done for the project and the functional and non-functional requirements. Section three will include the requirements analysis and section four will highlight methodology used. The design will be discussed in section five, implementation in section six and testing in section seven. Section eight includes an evaluation of the final project and section nine concludes the paper.

1.1. Aim

The goal of this project is to provide an effective learning method, with an easy user interface, implemented on a popular application which can provide educational support.

1.2. Problem Identification

These main problems that Packle targets include:

- Lack of awareness in students identifying retesting and spaced repetition as effective methods of studying.
- Cost of applications creating a digital divide for students who cannot afford distraction-free paid applications.

- Limited reach of web applications where additional marketing would be required for students to discover the tool.

1.2.1. Provide awareness regarding retesting and spaced repetition

The ability to learn and retain information is a life-long skill that will help in formal education, personal development, and career advancements. One of the main problems identified while researching for this project is that students are unaware of how effective retesting is.

Students intuitively, albeit incorrectly, assume that learning is only achieved through studying and that testing is useful for evaluation (Pyc and Rawson, 2010). The implementation of Packle will allow students to discover the effectiveness of retesting and spaced learning.

1.2.2. Provide a free solution without advertisements

Currently there are a number of free tools available for students to make their learning process easier. Free tools play an important role in increasing digital literacy (Danica, 2018) and this in turn bridges the digital divide (Rowsell, Morrell and Alvermann, 2017). Free tools are necessary to provide all students with equal accessibility. However, many free tools are funded by advertisements. Since advertisements are a distraction on an application where students need to concentrate, these tools come at an alternative cost.

Quizlet (2021) is a free flashcard web application which is difficult to use since it constantly reminds students to upgrade to the paid tier. One-third of the study progress page is also occupied by advertisements. StudyStack (2021) and Flashcard Machine (2021) are similar free tools filled with advertisements.

From a business perspective, it is understandable that companies are hoping to make a profit from providing a service or need advertisements for running costs. However, this poses a problem for students who cannot afford the ‘luxury’ of studying without distractions.

1.2.3. Provide a service on a popular application

One of the main issues considered during the conception of Packle was whether the study tool would be best implemented as a web application. A considerable amount of marketing

would be required to help students discover a web application and getting students to adopt and continue using it would be difficult. Therefore, implementing Packle on a popular platform for students would be more effective.

The proposed application for the project implementation is Discord. It was chosen for the following benefits:

- Popular among students and so a majority of students would have an account and experience using it.
- Natively offers features that allow students to collaborate and learn, such as servers for study groups and text chat, voice chat or screen share functionality which creates an enriching experience.
- Discord's open application programming interface (API) allows developers to create bots which makes it easy to implement Packle.
- Disboard (2021) has thousands of servers tagged with 'study', which makes it easy to find servers which would benefit from having Packle.
- While Discord offers purchasing 'Nitro' which allows users some additional functionalities (2021a), all the useful functionalities of the application are provided for free.

1.3 Stakeholder Identification

Figuring out the users of the bot will help to establish requirements that will fulfil their needs. Due to the nature of the bot and its ability to cater to a wide variety of subjects there could potentially be more special groups of users that Packle can cater to. For this implementation, I have considered the following as the key users:

1.3.1 Students

The main goal of this project is to help students use flashcards to retain information and so students are the main priority. A student is regarded as anyone formally or informally pursuing education in any field. Since spaced-repetition was found to be an effective

learning method when used in special needs education, Packle could potentially be used to aid in teaching vocabulary to students with special needs as well (Al-Jumeily *et al.*, 2016).

When considering and evaluating the needs of students I will not be limiting them to a certain age group. However, since Discord's Terms of Service includes that users must be 13, Packle will be restricted to students that fit this age limit. However, younger children can still use their parent's account on Discord under adult supervision.

1.3.2 Teachers

Teachers can use Packle as an interactive teaching tool. Discord is the perfect platform for online learning where teachers can make a server and stream, voice chat and video chat to easily communicate with students. They can use Packle to test how well students understood a lesson.

Additionally, by making flashcard packs, teachers would have an extensive library of key questions related to the subjects that they teach which can serve as a great resource for when students need to study for exams.

2. Literature Review & Research

The purpose of this project is to implement an effective method of studying on a platform which is free and popularly used by the target demographic. Since the success of the project depends on the effectiveness of and ease of access to the method implemented, the main areas of research will focus on understanding the study methods and technology proposed for the implementation of the project.

2.1. Effectiveness of retesting and spaced repetition

Research was done to determine whether retesting and spaced learning is effective across variables such as different age groups, content complexity, and a digital environment.

2.1.1. Studies conducted to evaluate spaced learning and retesting

Hermann Ebbinghaus, a German psychologist, discovered how memory is subject to a forgetting curve. From 1880 to 1885 he performed a series of experiments to memorize and

recall random syllables. Ebbinghaus published his findings in a book translated to English as ‘Memory: A Contribution to Experimental Psychology’ (1913). His experiments confirmed that memory and time are strongly correlated.

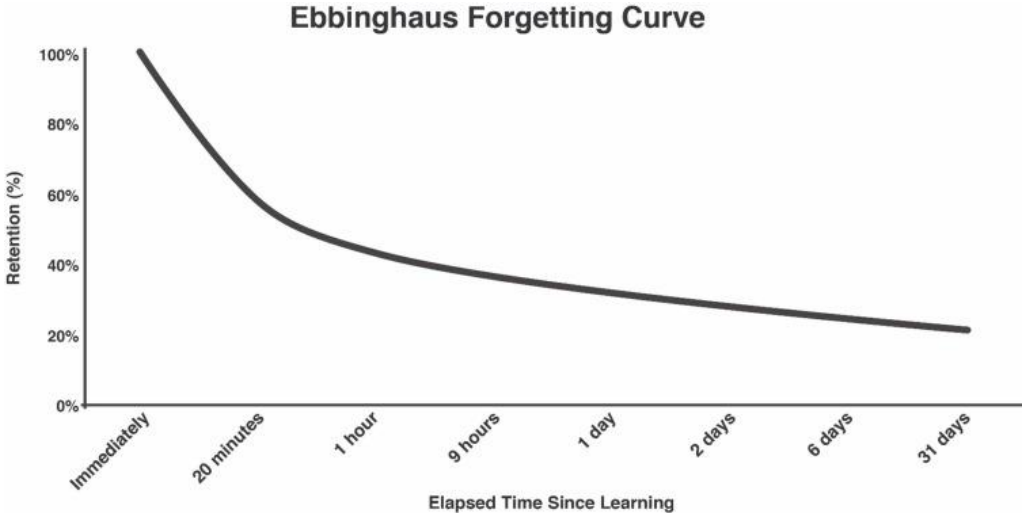


Figure 1: The Ebbinghaus Forgetting Curve (Shail, 2019)

Ebbinghaus’ Forgetting Curve is plotted as the amount of information retained against the number of days elapsed since learning. By the first day over 60% of the information is potentially forgotten. A study by Murre and Dros (2015) replicated the experiment done by Ebbinghaus. The results obtained were similar to the forgetting curve experiment conducted in 1880, supporting Ebbinghaus’ findings.

The method to combat this by spacing was also identified by Ebbinghaus (1913). His study was paramount to understanding the effectiveness of recollection to retain memories and how spacing can decrease memory deprecation. Shail (2019) discusses how the process of micro-learning by reintroducing information allows the forgetting curve to not degrade, shown in Figure 2. He suggests that breaking down information into brief learning moments allows learners to concentrate on specific information and that educators can harness this to design mobile applications.

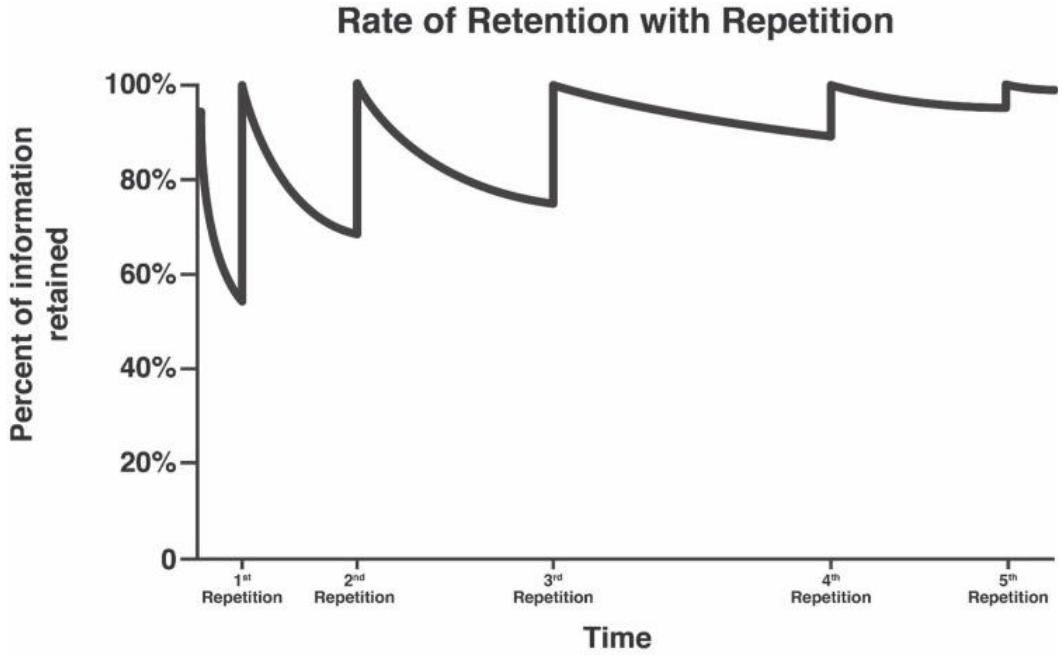


Figure 2: Hypothetical example of how spaced repetition affects memory retention (Shail, 2019)

Karpicke and Roediger's (2008) study on effectiveness of relearning and retesting is frequently cited in the field of effective learning methods. Their research was conducted to test college students on their ability to learn a foreign language. The study also let students predict what their outcomes were going to be which led to an interesting revelation that college students were unaware of how effective retesting was when compared to restudying. The research showed that when students were required to recall the material through retesting, they were able to correctly answer 80% of the questions compared to 33-36% scored by students that used methods where retesting was dropped.

Similar studies conducted on elementary level students showed that retrieval may be an effective learning strategy for children with varying levels of reading comprehension and processing speed (Karpicke, Blunt and Smith, 2016; Goossens *et al.*, 2014). Another study monitored the effects of spaced repetition learning for STEM (science, technology, engineering, and mathematics) students. They were provided with access to a web application with practice questions delivered through a personalised spaced repetition calendar. Results show that the students who practiced spaced repetition scored a mean of 70%. This is a significant improvement compared to the students who revised right before the exam scoring 64% and students who did not use the application at all scoring 61% (Voice

and Stirton, 2020). Soderstrom, Kerr and Bjork (2016) observed that restudying seems to produce better results when done in a spaced manner as well.

Spaced repetition and flashcards are also effectively used by medical students (Lambers and Talia, 2021), which shows that these learning methods are suitable for a variety of age groups and complexity levels of subjects. Medical students also seem unaware of the benefits of retesting compared to restudying. It is important to note that both the learning methods resulted in poor recall within a span of six months and so students must revisit topics throughout their study period to consistently retain the information they learn (Schmidmaier *et al.*, 2011).

A study by Pastötter and Bäuml (2014) also shows evidence that retrieval practice may actually enhance the ability to remember subsequent new information in the future. They state that this forward effect of testing could reduce memory deficits and enhance the ability for students to learn.

2.1.2. Rate of spaced repetition

The rate at which spaced repetition is performed determines its effectiveness. Therefore, it is important to identify the optimal rate. Whelan (2019) suggests using the Leitner system with flashcards where the cards get sorted into five levels of repetition and the length of repetition is daily, every two, four, nine and fourteen days. In this way, information that is harder to retain will be practiced frequently.

Kang (2016) states that identifying the timing of practice sessions is significant to data retention and that combining spaced retrieval with retesting is beneficial. His research identifies that retrieval slows down the process by which content is forgotten, attempts that occur close to the initial learning bear greater results, and that subsequent attempts can be pushed further apart in time. The effectiveness of these learning methods vary depending on the complexity of the material being learnt, retrieval method and time frame.

Based on findings such as the initial study by Ebbinghaus and Kang's identification of retrieval close to initial learning and spacing of subsequent retrieval attempts, Packle will implement a modified Leitner system with three levels of proficiencies at the proposed optimal rate for reminders being daily, the third and the seventh day within a week.

Following the depiction of Ebbinghaus' forgetting curve and cycle of review by Chun and Heo (2018) and the study by Shail (2019), the proposed rate of retrieval in Packle will hypothetically allow the retention of memory depicted in Figure 3. This system ensures that daily revisions will be for the material that the user finds the hardest to remember and that any information the user can recollect easily will be retested with increased spacing between the attempts.

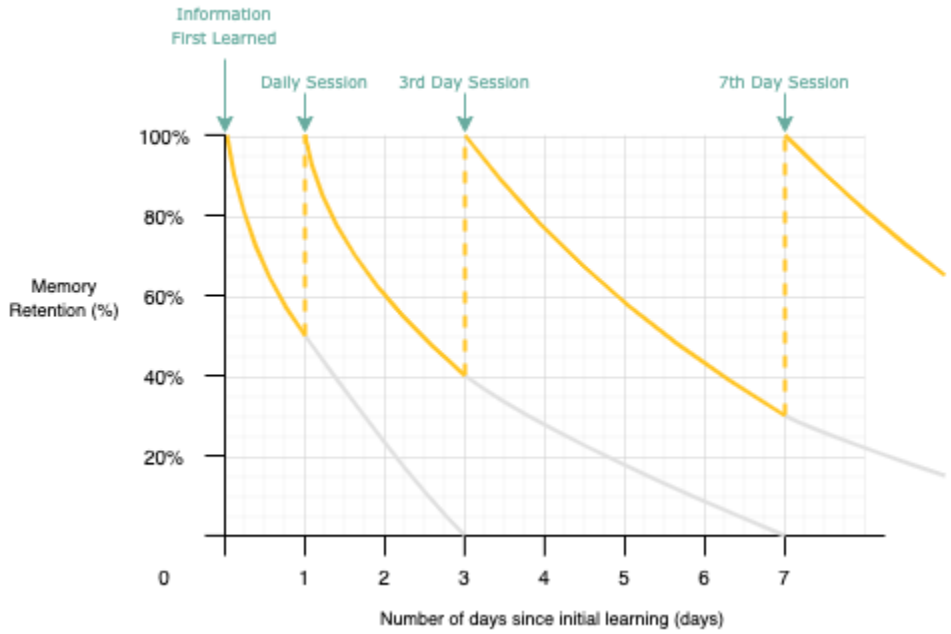


Figure 3: Hypothetical effect of the spaced repetition reminder on memory retention

2.1.3. Competition-based Learning

Research done by Whitman, Tanzer and Nemec (2019) on a flashcard quiz application 'Quizlet' shows evidence of how a quiz game can benefit students. Results showed that participants who used the application scored an average of 94.1% compared to the average score of 86.9% for students who did not use it. A study by Hwang and Chang (2016) also confirms a significant increase in the interest and attitudes of students that studied using a peer competition-based learning approach. Therefore, the quiz mode in Packle can be used as a way to motivate student learning.

2.1.4. Digital vs Paper

A majority of studies regarding the use of flashcards and testing in general use traditional methods of learning via printed material. Since Packle will be implemented and used via digital methods, it was important to determine whether the effectiveness of retesting and spacing was limited to traditional methods.

A study by Chen and Chan (2019) compared augmented reality flashcards with traditional flashcards when used in early childhood education. Their study showed that both methods significantly improved the ability for children to learn with no significant differences between the two.

Another study conducted by Sage *et al.* (2020) on undergraduate students compared the effectiveness of learning via paper, smartphones and laptops. This study discusses how students may prefer paper over digital flashcards to minimize technical issues or screen time. However, Ashcroft, Cvitkovic and Praver (2018) observed that digital learning methods benefitted students with a lower level of English proficiency suggesting that digital flashcards may offer deeper learning opportunities. Another benefit is that applications can send reminders to students which encourages persistent engagement.

One of the downsides to studying over digital applications may be that they distract students. Tossell *et al.* (2015) states that while smartphones can be a useful learning tool, students perceive it as a hindrance to their educational goals. Therefore, students would need to play a role in ensuring that they use their time studying effectively. One of the ways in which this can be done when using Packle is by using Discord's native ability to set the user's availability. Students can set their availability to 'Do not disturb' or even 'Invisible' to minimise distractions.

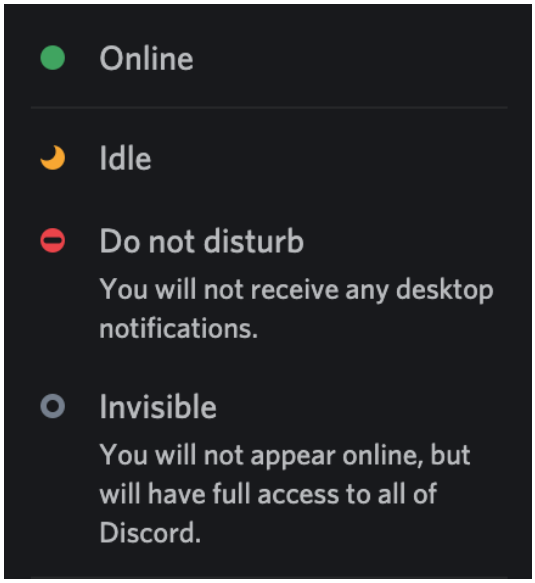


Figure 4: Discord's user status menu

Sage *et al.* (2020) tested a few hypotheses to understand differences between paper and digital flashcards. Their results showed that memory retention, cognitive load and satisfaction did not vary depending on the platform. However, students did spend more time reviewing the paper cards. This may be due to how digital flashcards are faster and easier to flip through.

Based on these studies, it is understood that the digital implementation of Packle could offer students similar or better results when learning compared to using traditional methods.

2.1.5. Limitations of spaced learning and retesting

While there are a number of studies which test and confirm the effectiveness of retesting compared to restudying (McConnell, St-Onge and Young, 2015; Roediger and Karpicke, 2006), Goossens *et al.* (2016) states that the success of those studies may apply only to lab sessions and do not function as well implemented in real life.

Another factor that could affect the ability to retrieve learned information is when the information needs to be retrieved while under high-pressure. Hinze and Rapp (2014) studied the effects of performance pressure on retrieval by making participants take high or low stake quizzes and then a final test. While the results of the quizzes were similar, participants

that took the low-stake quiz performed better on the final indicating that pressure may disrupt the ability to retrieve information. While Packle has a quiz mode this is only done as practice and would not be similar to the pressure that participants were subjected to in this study.

2.2. Technology

Since Discord was chosen as the potential platform for Packle, research needed to be done to confirm if this would be viable.

2.2.1. Discord's use as an education tool

Discord is a free application with support for text, video, voice and streaming. It is available cross platform on desktops and mobile and can be accessed through web browsers and as a stand-alone application (Kiene, 2020). While it was initially developed for gamers, it is now used by people to create community spaces for a variety of interests. Discord states that it has 150 million monthly active users and 19 million active servers (2021b).

Discord's popularity seems to have risen through the Covid 19 pandemic, with institutions such as the University of Ploiesti in Romania successfully utilising Discord as the main online study tool for Computer Science students. They even developed their own bot 'info-AI' which allowed the university to do useful tasks such as keeping track of attendance, recording assignment submissions and conducting exams (Vladoiu and Constantinescu, 2020).

The application was used in a similar manner at Polytechnic of UBAYA, a university in Indonesia. Students appreciated the useful functionalities native to Discord, its low resource requirements and cross-platform support (Bernad, 2021). Another study done by Wahyuningsih and Baidi (2021) on elementary students identified that Discord was an effective replacement for face-to-face classes. Students and teachers stated that learning over the application was an enjoyable and interactive experience. The main challenge identified by this study is how unfamiliarity leads to difficulties in the beginning. Therefore, beyond developing the bot, it will be crucial for Packle to have an easy to understand knowledge base which will allow new users to understand the technology.

A study conducted by Odinokaya *et al.* (2021) was aimed at testing the scores of students learning English as a foreign language. The students using Discord significantly outperformed the control group with students benefiting from the organization, group interactions and daily learning exercises. This study supports the fact that implementing study reminders on Packle may help students to understand that this repetitive yet small effort put in daily can lead to much better learning.

2.2.2. Implementing Packle as a Bot

The flexible APIs provided by Discord allows developers to create useful and powerful bots which can be added to servers to enhance them in creative ways (Kiene, 2020). Discord's extensive knowledge base on bot building (Discord Developer Portal, 2021) and specifically discord.py, the python API wrapper for Discord bots created by Rapptz (2021), allows the development of useful bots with endless possibilities. The consistent updates by Discord also allows continual improvement of the bot.

2.3. Known existing bots with similar functionality

To ensure that Packle is a useful and innovative solution, it was necessary to conduct research to look for existing Discord bots that offered flashcard functionality or spaced-repetition. The limited options that were discovered are outlined in Table 1.

Table 1: Known bots with similar functionality

#	Name	Available At	Pros	Cons
1	Flashcards Bot	https://github.com/insertteamhere/flashcardsbot	- Code available on Github	- Users need to locally build the bot - Very limited functionality to view and quiz flashcards
2	Flash Cards	https://top.gg/bot/543203701314027531	- Flashcard bot where you can create and study cards	- List of commands mentions modules but provides no command to create one - Bot is not online and cannot be used or tested

3	Kotoba Discord Bot	https://kotobaweb.com/bot/quiz	- Impressive knowledge base - Various custom UI/UX settings	- The bot has an extensive library of flashcards but they are made specifically to teach Japanese - To create custom cards you need to use their website
4	Belta Flashcards	https://github.com/katagatame/Protomolecule/wiki/BeltaFlashcards	- Code available on Github	- Choice format instead of retesting through flashcards - Users can't add cards and the content teaches a fictional language from a TV show

To the extent of my knowledge, Packle will be the first of its kind available on Discord with the following functionalities:

- Offers spaced repetition and retesting functionality
- Fully operational on Discord itself without requiring the user to visit other websites
- Allows users to easily create their own content instead of being limited to pre-set data

3. Requirements Analysis

Requirements analysis is the clear definition of the problem and the solution that fulfils the needs of stakeholders. It is crucial to successful software development and allows developers to solve a problem by building the right solution (Meyer, 2014, p.32). Quality requirements are unambiguous, singular, and concise (Giachetti, 2010, p.188). The requirements for Packle were determined using the research in section 2 and a survey done to understand the needs of stakeholders.

3.1. Questionnaire

A questionnaire was designed as a Google form survey that could be distributed online. The complete survey can be found in Appendix A-G. Since the survey was anonymous and did not collect any personal information, it would not require an ethical review.

The only requirement to be a participant was experience with studying in either a formal or personal setting. The goal of the questionnaire was to survey students to understand information regarding their study methods and use of Discord.

3.1.1. Analysis of results

The survey received a total of 71 responses where 49.3% was for the age group 13-20 and 42.3% was for the age group 21-30.

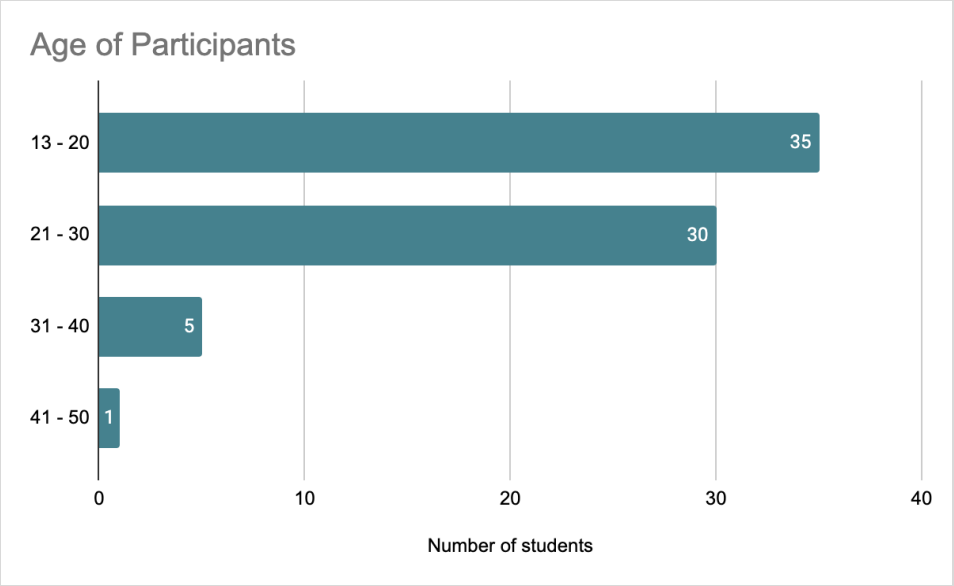


Figure 5: Participant's ages

The results were analysed to understand the information provided by participants for the following research questions.

3.1.1.1. What are the student's preferred methods of studying?

40.8% of participants stated that they learnt the material close to the exam and 39.4% of participants stated that a combination of studying and testing is preferred. Additionally, 12.7% prefer repetitive re-learning, 5.6% prefer self-testing to identify weaknesses and one student stated that they do not study.

This information aligns with the studies explored in section 2.1.1. where students intrinsically believe that studying close to the exam is the best way to retain information.

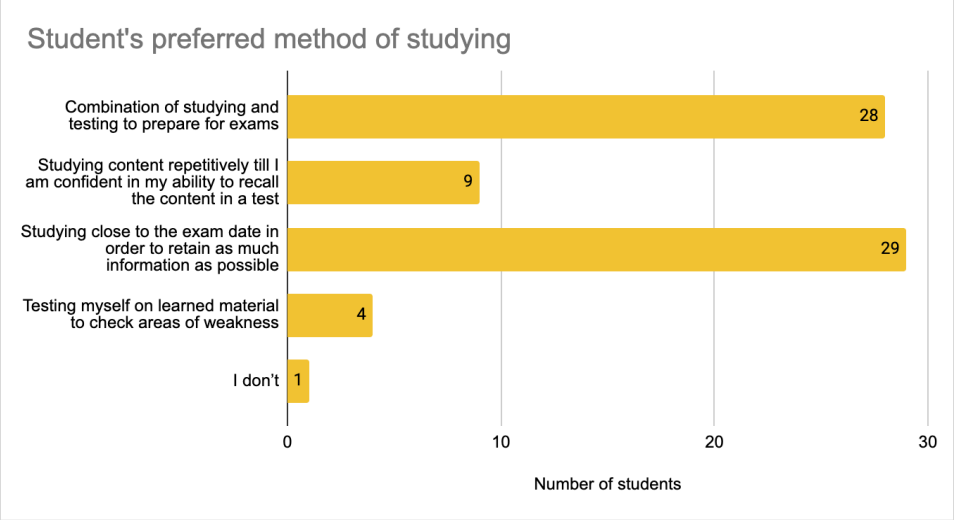


Figure 6: Student's preferred method of studying

3.1.1.2. Are students aware of retesting and spaced learning?

Of the 71 participants, 84.5% were aware of retesting as a learning method compared to 59.2% that were aware of spaced repetition as a learning method. 71.8% of the participants said that they have used either retesting or spaced learning. This shows that not all participants who are aware of these methods are utilising it in their learning process.

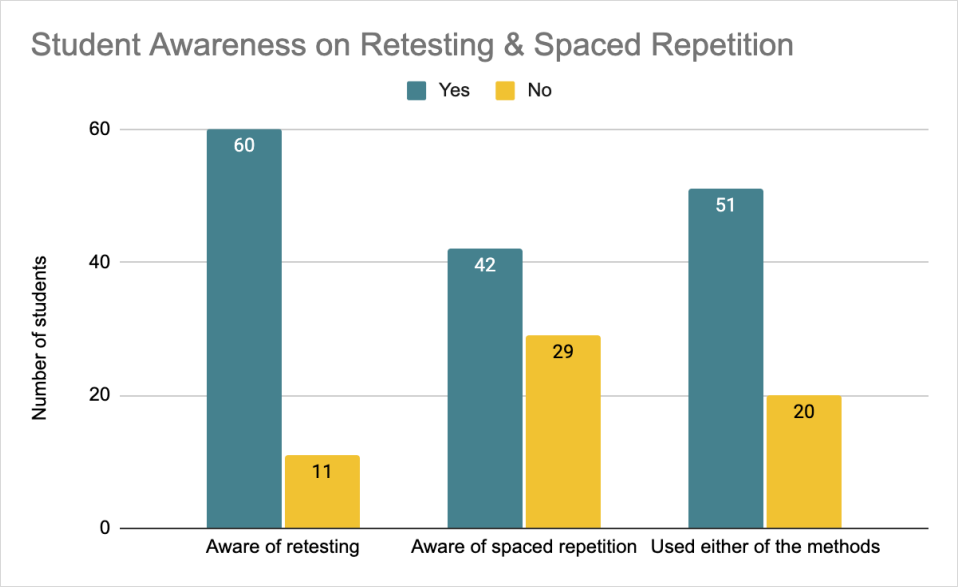


Figure 7: Student's awareness of retesting, spaced repetition and usage

3.1.1.3. If students used retesting or spaced learning, were these methods effective?

When presented with the statement ‘retesting is an effective method of learning’, 90.2% of the participants stated that they strongly agree or agree with the statement and similarly 76.5% of students strongly agreed or agreed with the statement ‘Spaced learning is an effective method of learning’. This shows that similarly to the results of the studies explored in section 2.1.1., a significant majority of students benefit from using these methods.

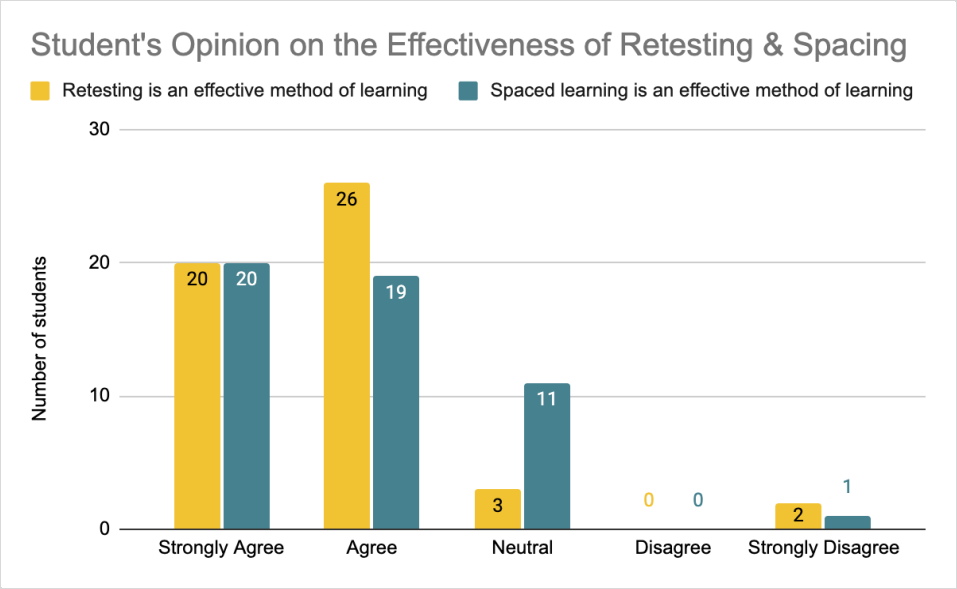


Figure 8: Student’s opinion on the effectiveness of retesting and spaced repetition

3.1.1.4. If students used retesting, what was the most useful method?

60.8% of participants stated that they use revision questions, 25.5% stated that they use flashcards, and 7.8% said they use quizzes. The low percentage for quizzes might be due to the fact that it's difficult to find quizzes that cater specifically to the content that is being taught and students may find it inconvenient to create their own.

The past exam paper option is similar to revision questions and Quizlet is a form of a quiz. One participant had also stated that they take short notes although it is unknown whether this involves retrieval of memory. Since Packle’s implementation would use flashcards, studying them would be similar to the use of revision questions to retrieve information.

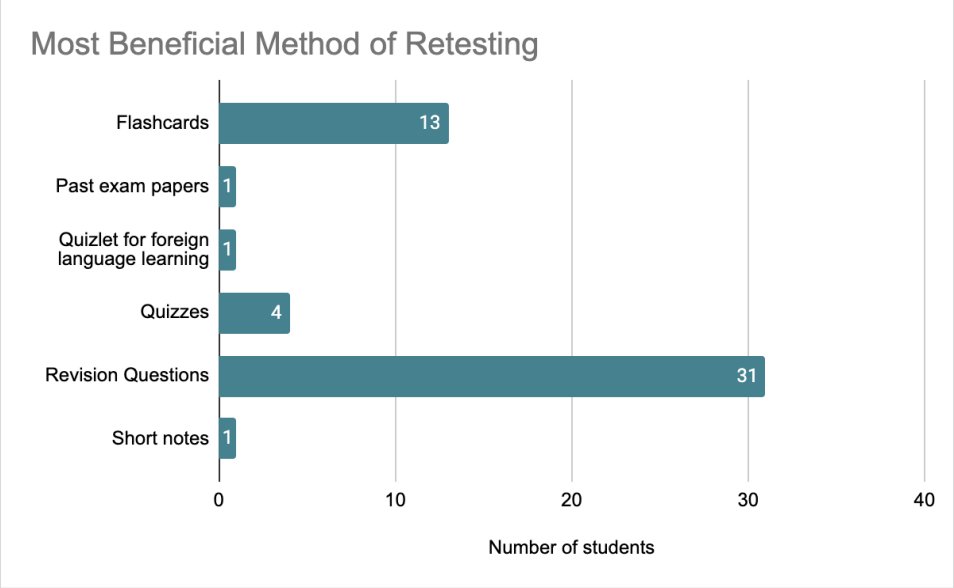


Figure 9: Survey results for most beneficial method of retesting

3.1.1.5. If students used spaced learning, what was the most effective time frame?

While participants added in multiple levels of repetition that suited them best, 43.1% of students chose ‘daily’ and 41.2% of students chose ‘weekly’. Since Packle will be implementing a system that sends a reminder every 1st, 3rd and 7th day of a week, this aligns with the current study habits of students where a majority prefer revisiting content daily and weekly.

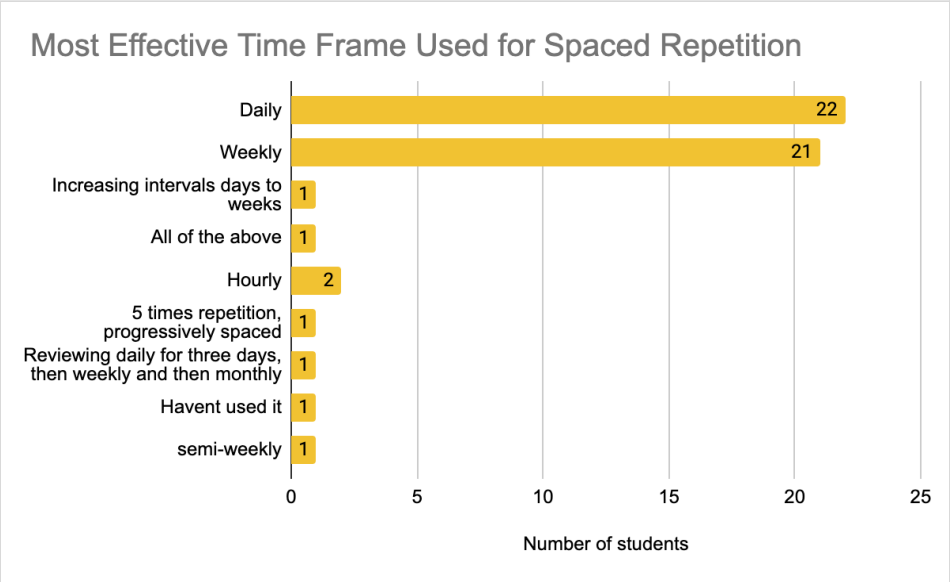


Figure 10: Survey results for most effective time frame of spaced repetition

3.1.1.6. Would students be willing to try these methods?

A majority of participants that had not yet tried these methods show a willingness to try both retesting and spaced repetition. Packle would provide an easy way for students to test these methods.

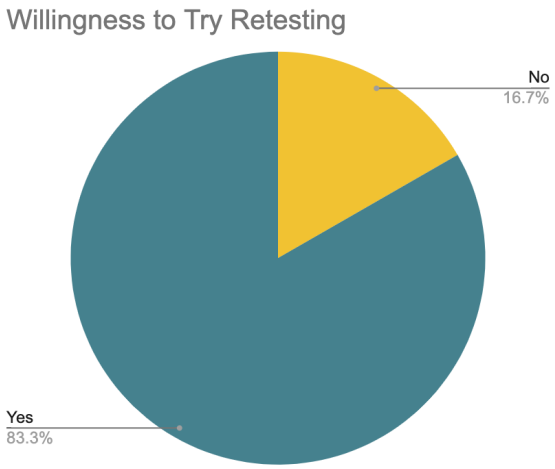


Figure 11: Student's willingness to try retesting

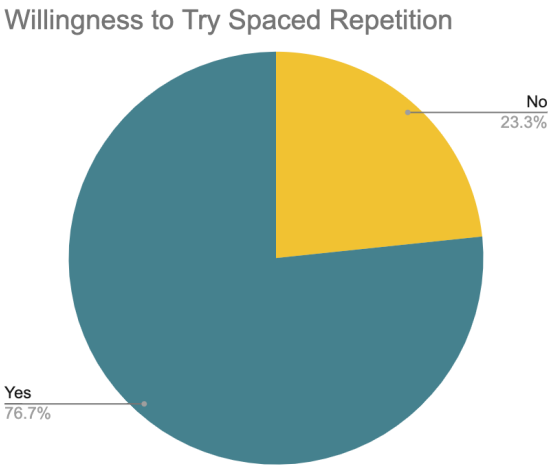


Figure 12: Student's willingness to try spacing

3.1.1.7. Do students use Discord as an educational tool?

57.9% of participants had not used Discord for studying, which is expected since Discord is used for a wide variety of purposes. However, this does not negate the fact that 42.1% of participants have used it for educational purposes. Additionally, 77.2% of participants strongly agree or agree that Discord can be used as an educational tool. Packle's implementation on Discord would allow students to enhance their study experience on the platform.

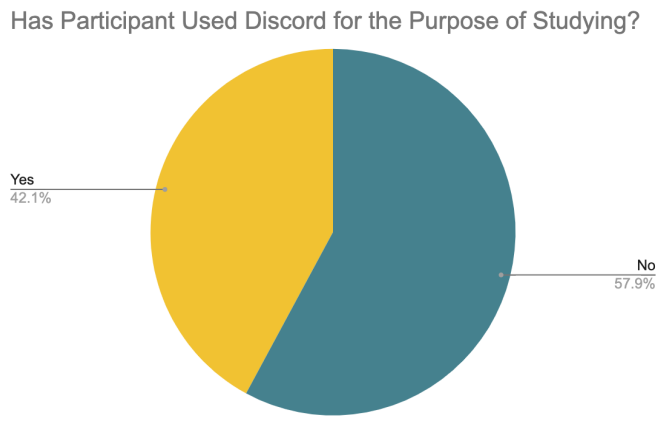


Figure 13: Survey result for student's use of Discord as educational tool

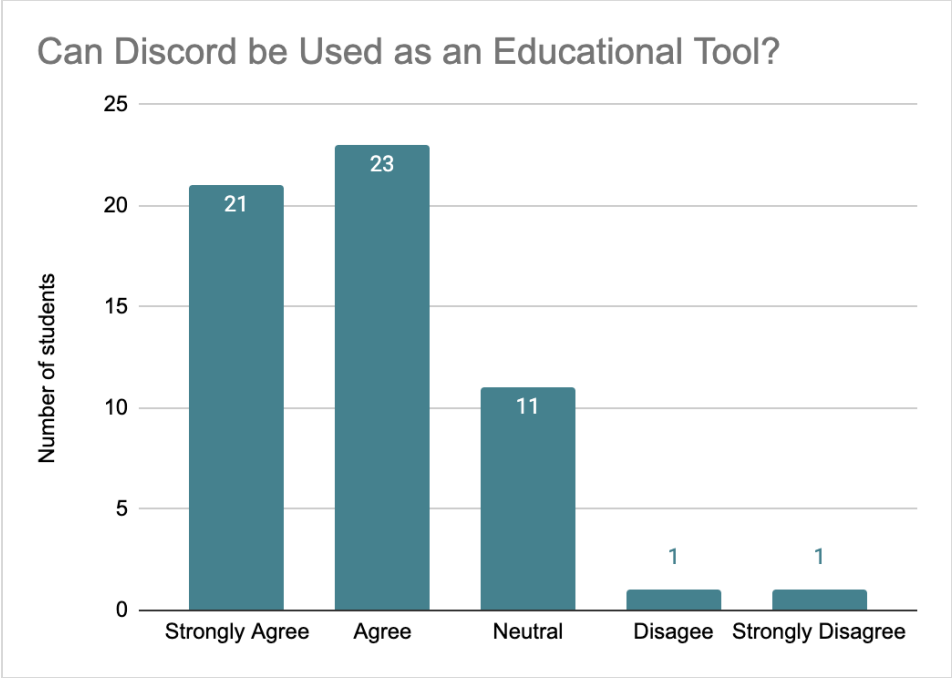


Figure 14: Student's opinion on whether Discord can be used as an educational tool

3.1.1.8. How familiar are Discord users with bots and how likely are they to use one?

It was important to identify if Packle would be used by students. The most interesting result obtained from this study is the relationship between participant's familiarity with using Discord bots and their willingness to use one as a learning tool. Therefore, this aligns with the findings in section 2 regarding the importance of making sure users have a proper knowledge base to depend on so that the use of new technology is not hindered or rejected due to unfamiliarity.

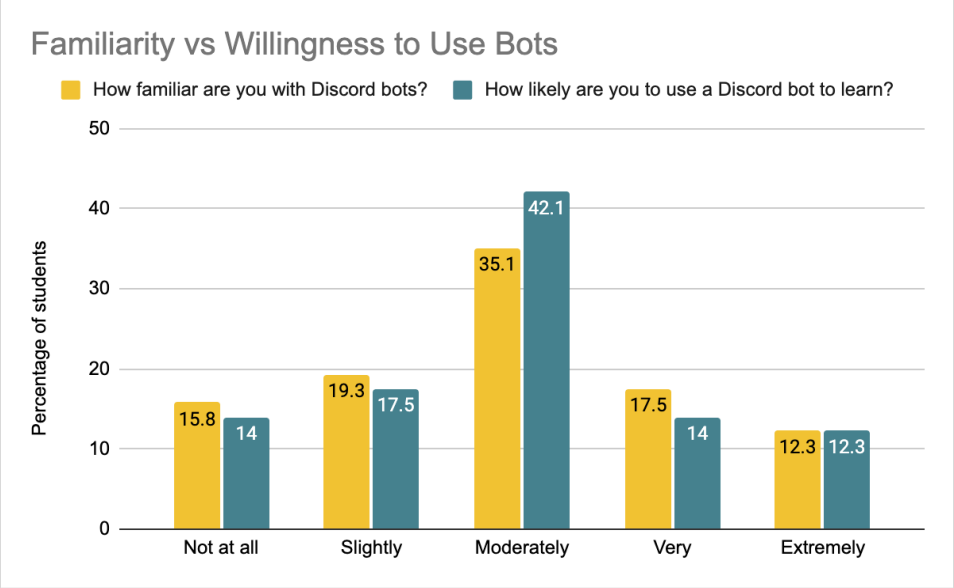


Figure 15: Familiarity against willingness to use Discord bots

3.1.1.9 If the student is not a Discord user, would they be willing to try it now?

When participants without a Discord account were asked if they would use it in the future to help with their education 57.1% stated 'No' and 42.9% stated 'Yes'. This shows that a considerable percentage of non-users would be willing to try the application.

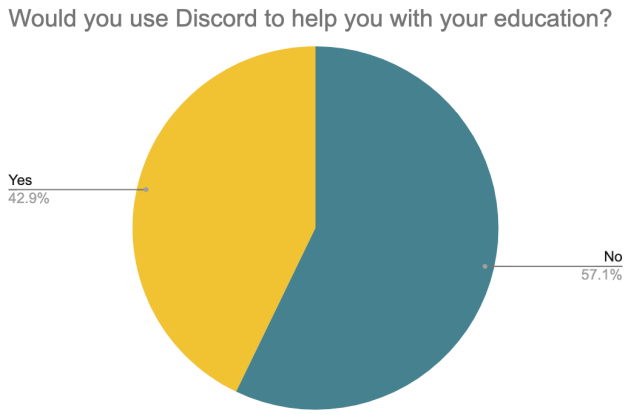


Figure 16: Survey result for non-user's willingness to use Discord

3.1.1.10 Conclusion

A majority of students seem to prefer bulk-learning before an exam which might mean that students are unaware of more effective methods. However, not all participants that are aware of these learning methods are using it and a majority of participants that are not

aware are willing to try it. Therefore, Packle might be a useful and easy tool to spread awareness and allow students to have effective study sessions.

While the method of retesting that most students preferred is revision questions, these can easily be done in the form of flashcards which would also allow students to proactively make questions from the content learnt and have a database of these questions readily available for them to study at a later date. Additionally, students may use quizzes more when Packle provides an easier way to create them.

The proposed reminder system for Packle aligns with current student habits and a majority of students agree that Discord can be used as an educational tool. Additionally, it was observed that unfamiliarity may deter students so the implementation must be simple with useful guides to help students understand how it works. Overall, the results of the survey show that Packle will be a useful tool for a majority of students.

3.2. Requirements

Using the research from section 2 and the survey in section 3.1., the functional and non-functional requirements of Packle have been captured in Table 2 and Table 3. The tables include an identifier, requirement, its rationale, and MoSCoW rule to prioritise what functionalities the system must, should, could and would not have (Apperly *et al.*, 2003, p.97).

3.2.1 Functional Requirements

Functional requirements determine the actions that the system must be capable of handling to support stakeholder goals (Giachetti, 2010, p.186). Since they define what the system should or should not do it is easy to determine the complete implementation of a functional requirement.

Table 2: Functional requirements of Packle

Identifier	Requirement	Rationale	MoSCoW Priority
FR01	Accept questions	Allows users to input question/answer pairs	Must have

FR02	Manage packs	Allows users to add useful information such as ‘category’ for packs	Could have
FR03	Study card packs	Allows users to study cards	Must have
FR04	Enable/disable reminders	Allows users to choose if they want the spaced learning feature	Should have
FR05	Send scheduled study reminders	Allows users to get spaced reminders	Must have
FR06	Adjust pack and card data based on study session	Allows implementation of the Leitner system for cards and spaced learning in packs by adjusting card proficiency level	Must have
FR07	Display information about packs	Allows users to view pack information such as number of questions	Could have
FR08	Initiate a quiz	Allows users to use cards for competition-based learning	Should have
FR09	Track user progress for quiz	Allows users participating in quiz to understand how well they did	Could have
FR10	Base system logic on findings from research	This allows the system to implement an effective method	Should have
FR11	Data persistence	This would allow students to not lose progress when the bot goes offline	Would not have

3.2.2 Non-Functional Requirements

Non-Functional requirements are the properties of the system that are important for stakeholders and affect their satisfaction with the system. They cannot always be measured and are subjective since they are qualities of the system such as reliability, safety, performance, scalability, or usability (Giachetti, 2010, p.187).

Table 3: Non-functional requirements of Packle

Identifier	Requirement	Rationale	MoSCoW Priority
Reliability			
NFR01	The bot would always be online	This would allow uninterrupted access	Should have
NFR02	The bot should be able to support multiple users	This allows simultaneous use of the bot without issues	Should have
NFR03	The bot should be able to accept unlimited card packs	This allows users to submit as many packs as they want without being restricted	Could have
Safety			
NFR04	The bot should not abuse its listening access to text messages	This would allow users to trust using the bot without fear of losing privacy	Must have
NFR05	The data of users and information submitted should be kept securely and not shared with anyone beyond the original user	This would allow users to trust using the bot without fear of losing privacy	Must have
NFR06	The CSV files output by the bot must be safe and secure for users to download	This ensures protection of users	Must have
Performance			
NFR07	The bot should respond to commands quickly and efficiently, within a maximum delay of 3 seconds	This allows users to effectively use the bot without lags	Should have
NFR08	The bot should catch any errors with user's inputs	This would allow users to understand why their input lead to unexpected results	Should have
Scalability			
NFR09	The bot should be upgradable with changes to both Discord and discord.py	This allows the bot to keep up with changes and implement new features	Must have

Usability			
NFR10	The bot should provide information that allows users to understand its use completely	This allows users with varying experience to effectively use the bot	Must have
NFR11	The bot should use commands that are self explanatory and easy to remember	This allows users with varying experience to effectively use the bot	Should have
NFR12	The user interface of the bot must be optimised for all platforms and devices	This allows the UI to best fit all device and settings	Could have

4. Methodologies

Governing of the development process through a set of principles that define the tasks, methods, deliverables, and best practices is known as the methodology (Giachetti, 2010, p.86). For this project the waterfall methodology will be used to schedule its development.

The general downside to using this approach in projects is how problems may occur from poor communication, incomplete requirements capturing or poor integration of component interoperability (Giachetti, 2010, p.88). However, since this is a solo, small scale project with well defined requirements, the waterfall approach will suit the project better than a method such as Agile which is designed for teams.

In this methodology, each well-defined step of the process will be completed before moving to the next. This creates a cascade which starts at requirements capturing and moves through the design, implementation and testing process at which point the system can be released. These processes will be managed in the form of a Gantt chart shown in Figure 17. A larger version of this can be found in Appendix H.

The only changes made to the waterfall methodology will be the overlapping of literature review, testing and bug fixing processes. This is because literature review is a time consuming process that allows the development of a solid system based on research.

Overlapping of testing and bug fixing allows the development of an error-free system within the short time period available.

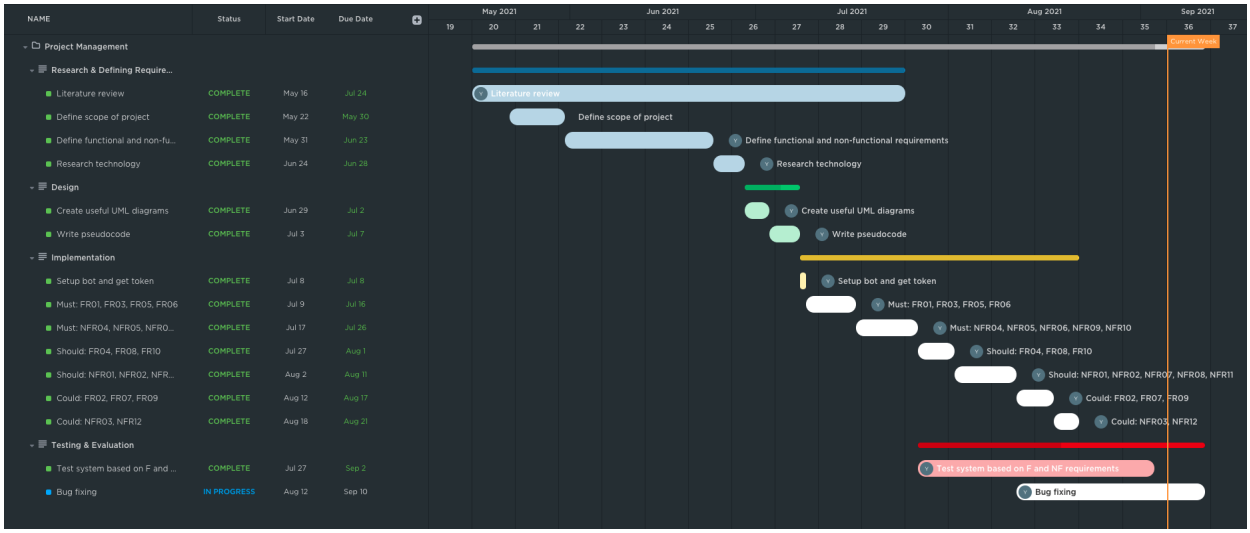


Figure 17: Gantt chart used to schedule the project

5. Design

Architectural system design is a crucial part of the development process which allows stakeholders to identify how the system will be organised and structured. Through this process all the system's components and the connections between them are determined (Sommerville, 2016, p.168).

Some benefits stated by Bass, Clements and Kazman (2021, p.45-56) to creating a concise system architecture include being able to:

- Determine the quality of a system
- Understand whether changes can easily be made
- Allow stakeholders to communicate
- Define system constraints
- Develop better scheduling and costing estimates

5.1. Repetition & Spaced Learning System

One of the most important components of Packle is the repetition and spaced learning logic. The flashcard style used as a retesting method in Packle allows users to practice active recollection of memories. Additionally, users studying 1-on-1 sessions can enable reminders to utilise spaced learning based on the Leitner System (Olesea, 2019).

5.1.1. System Logic

Packle uses the following components within its learning system

- Card: The question / answer pair submitted to Packle by the user
- Pack: A collection of cards
- Proficiency: Proficiency level of a card which determines the rate of repetition

When a user uploads cards to create a pack, they will be initialised at proficiency 1. The proficiency determines the rate of repetition as follows within a 7 day loop:

- Proficiency 1: daily reminders
- Proficiency 2: reminder on the 3rd day
- Proficiency 3: reminder on the 7th day

Therefore, within a week students will be sent reminders for specific proficiencies and when a student initiates a study session, the relevant cards for the day are displayed. When a user studies a card, correct answers will increment and incorrect answers will decrement the proficiency. If the proficiency is 1 and the answer is incorrect, the card will remain at 1. If the proficiency is 3 and the answer is correct the card will move to ‘mastered level’ so that the card no longer appears in further study sessions.

In this way, content that a student knows will be studied at longer time frames of spaced learning until it is mastered and content that the student finds difficult will be repeated at a higher frequency. Additionally, students can reset a pack so that all cards are set to proficiency 1.

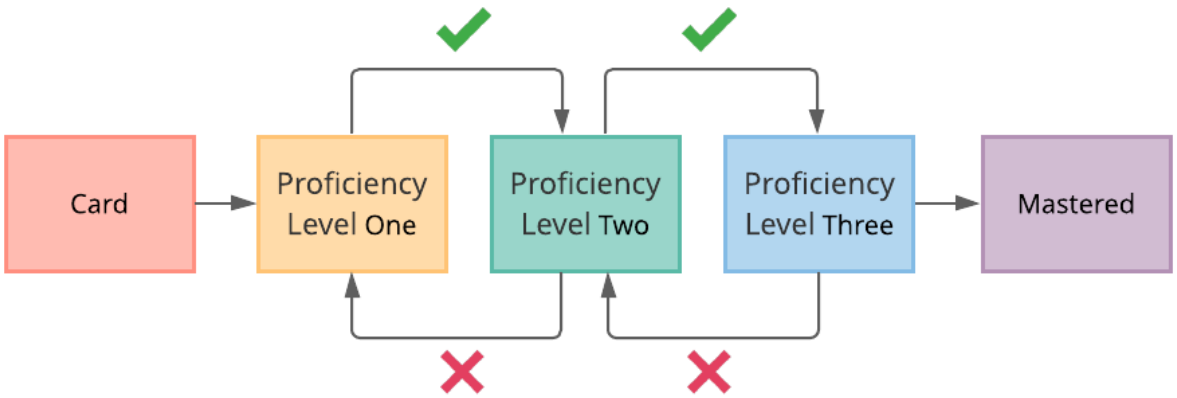


Figure 18: Flow of the proficiency system in Packle

5.2. System Architecture

A system architecture diagram allows stakeholders to view a high-level abstract version of the project that encompasses the overall functionalities of the system. Users will be able to send commands to Packle to which Packle will send responses. Packle will be able to initiate a study or a quiz session for users to participate in.

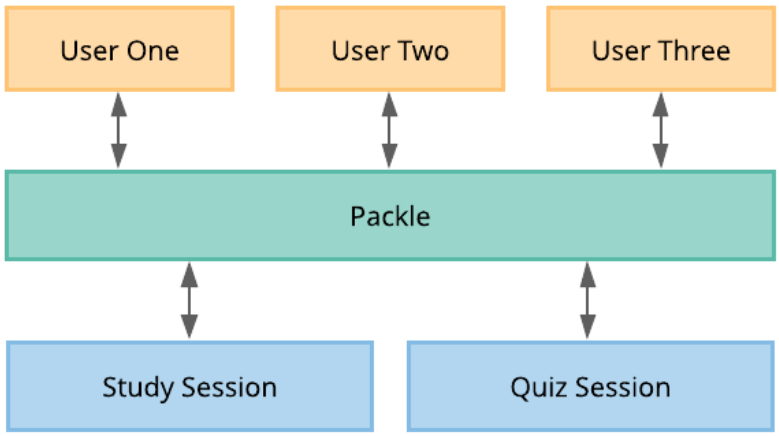


Figure 19: System architecture diagram for Packle

At the most basic level, Packle must be able to let multiple users interact with it simultaneously for study sessions or quizzes.

5.3. Use case diagram

Use case diagrams allow detailing the interactions between the user (actor) and the system. Effective use cases are brief and as simple as possible (Fowler, 2004, p.146-148). The main functionalities for a Packle user will be to add cards, change system settings, and study alone or with multiple users using the quiz mode. All the functions are performed using commands that require a specific format and parameters, which is why Packle will check if it's correct before executing the command.

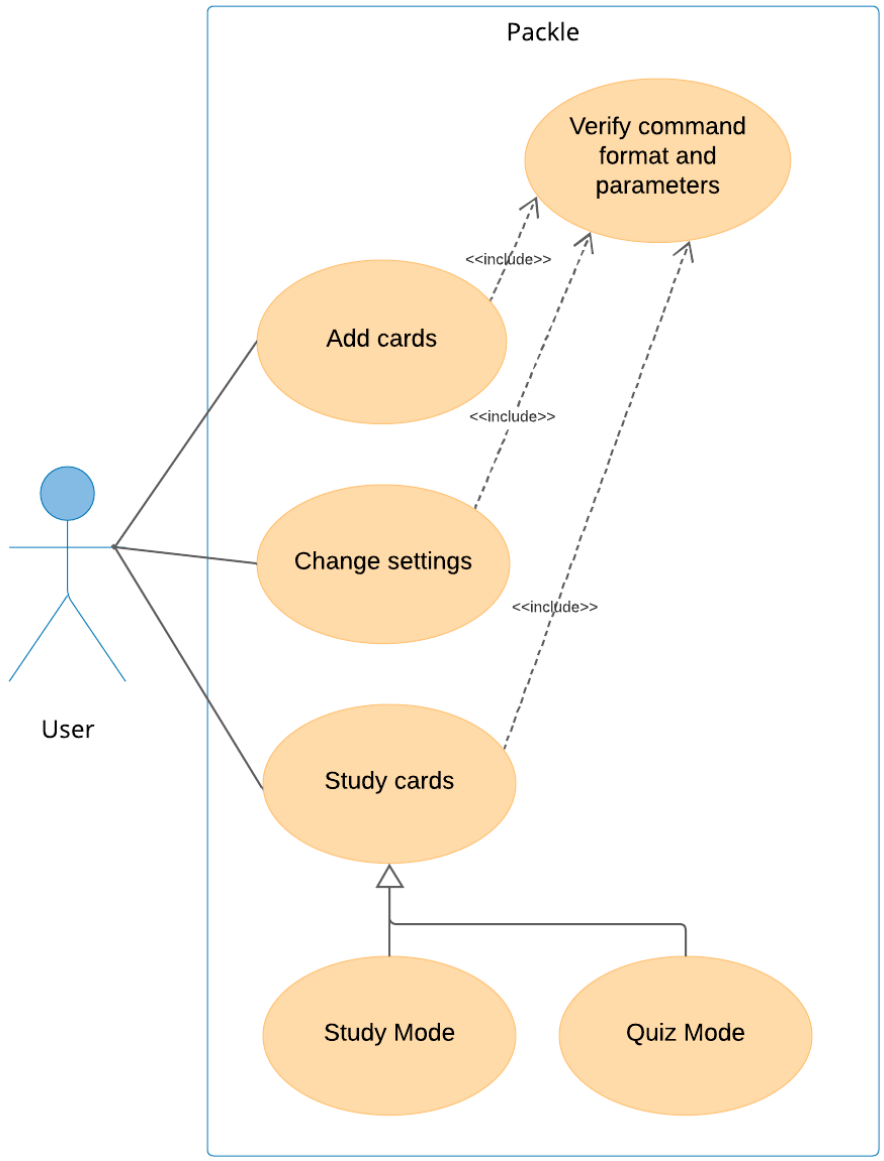


Figure 20: Use case diagram for Packle

5.4. Sequence diagram

A sequence diagram shows the interactions between the user and the system in chronological order for a use case (Sommerville, 2016, p.146).

5.4.1. Sequence Diagram for Creating Packs

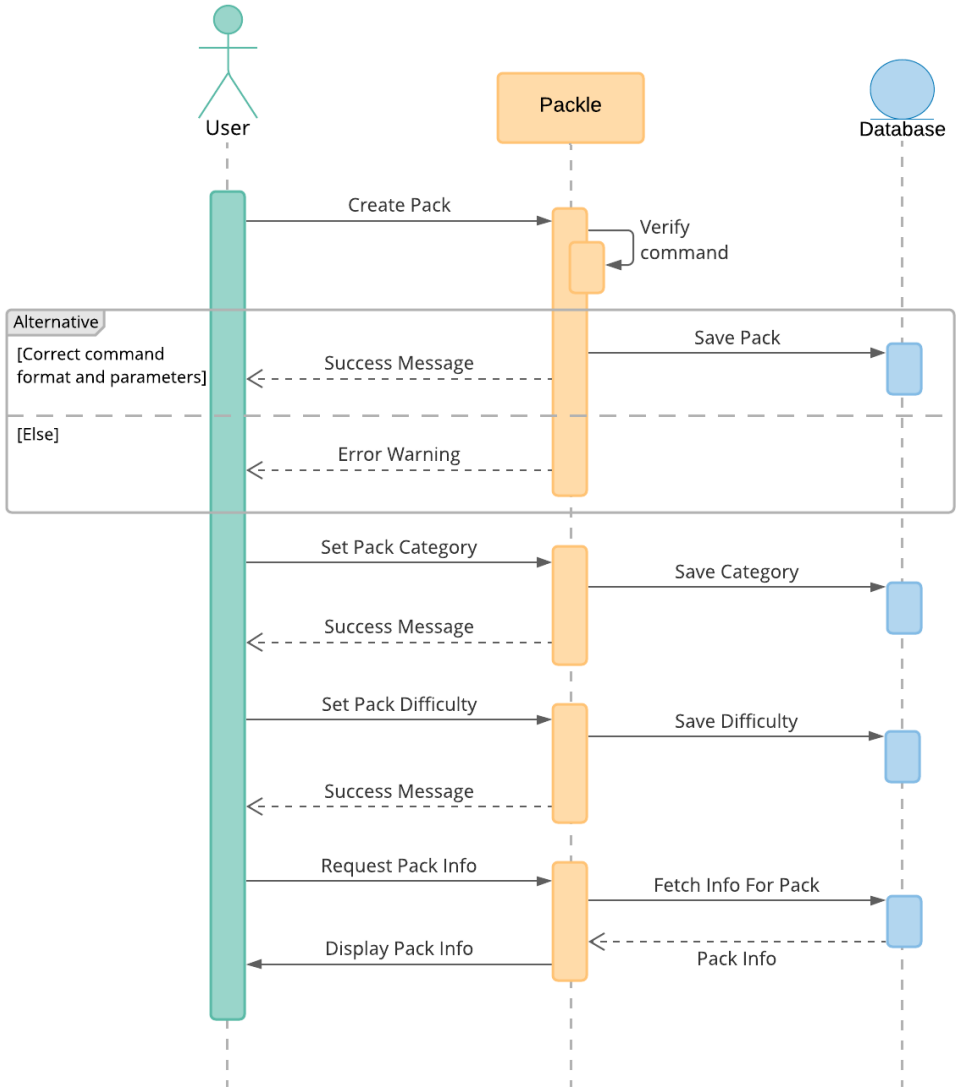


Figure 21: Sequence diagram for packs

Figure 21 shows some of the interactions that a user can have with the system to create, edit and view packs. While verifying commands is an action that will happen with every user interaction, it has been shown once to avoid repetition.

5.4.2. Sequence Diagram for Study Mode

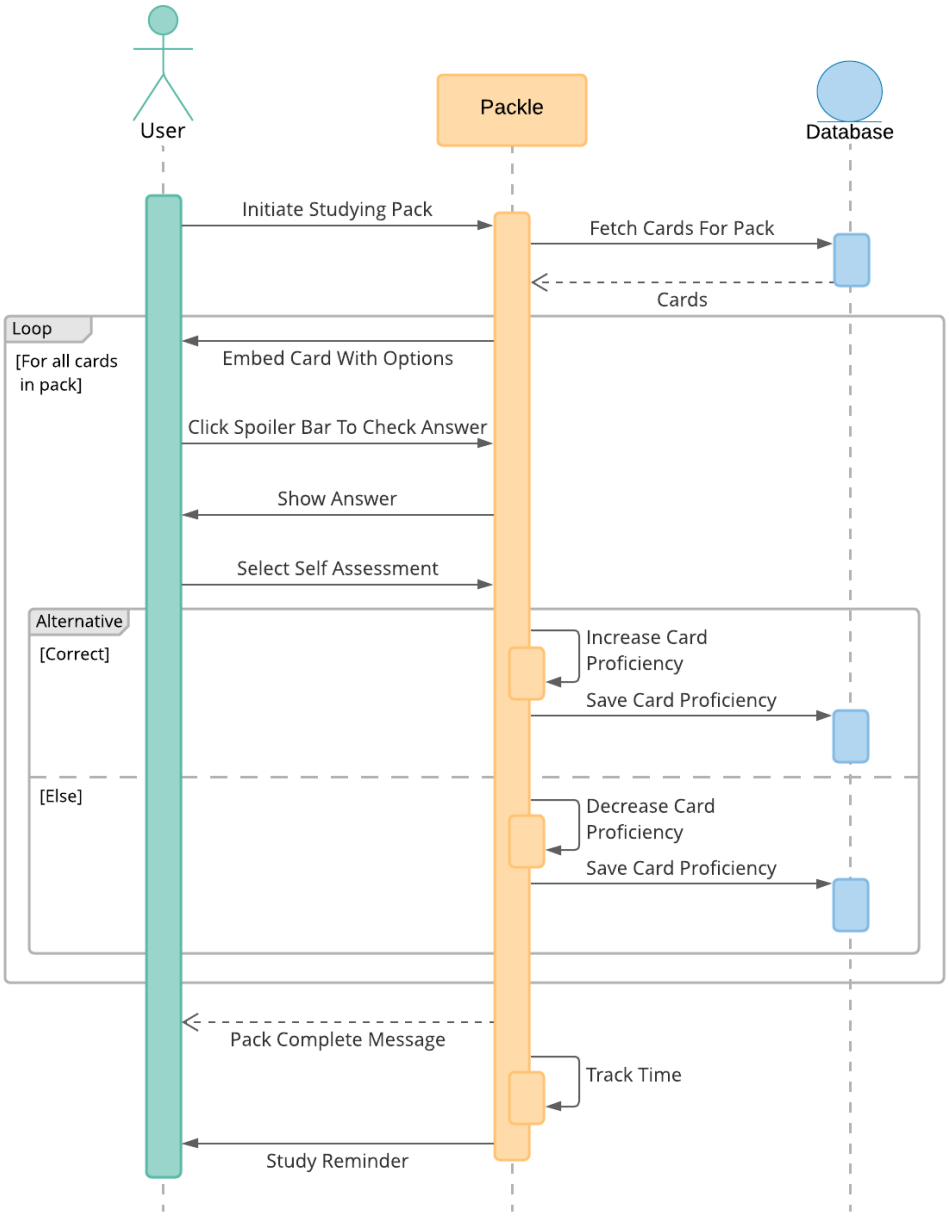


Figure 22: Sequence diagram for study mode

Figure 22 shows how a saved pack is studied. The user would initiate a study session by interacting with Packle in a direct message. When the user selects a pack to study, Packle will embed its cards in a loop. The card would have the question, the hidden answer, and stickers the user can click. The user will read the question, retrieve the answer from memory, reveal the answer and then conduct a self assessment to determine if their answer is correct or wrong.

Once the answer is selected, if it was correct the proficiency for the particular question will increase or if it was wrong it would decrease and the system will store this information. Once all the cards for the day have been looped through, Packle will let the user know their study session is done for the day. It will then track time until 24 hours at which point it will send the relevant study reminder to the student.

5.4.3. Sequence Diagram for Quiz Mode

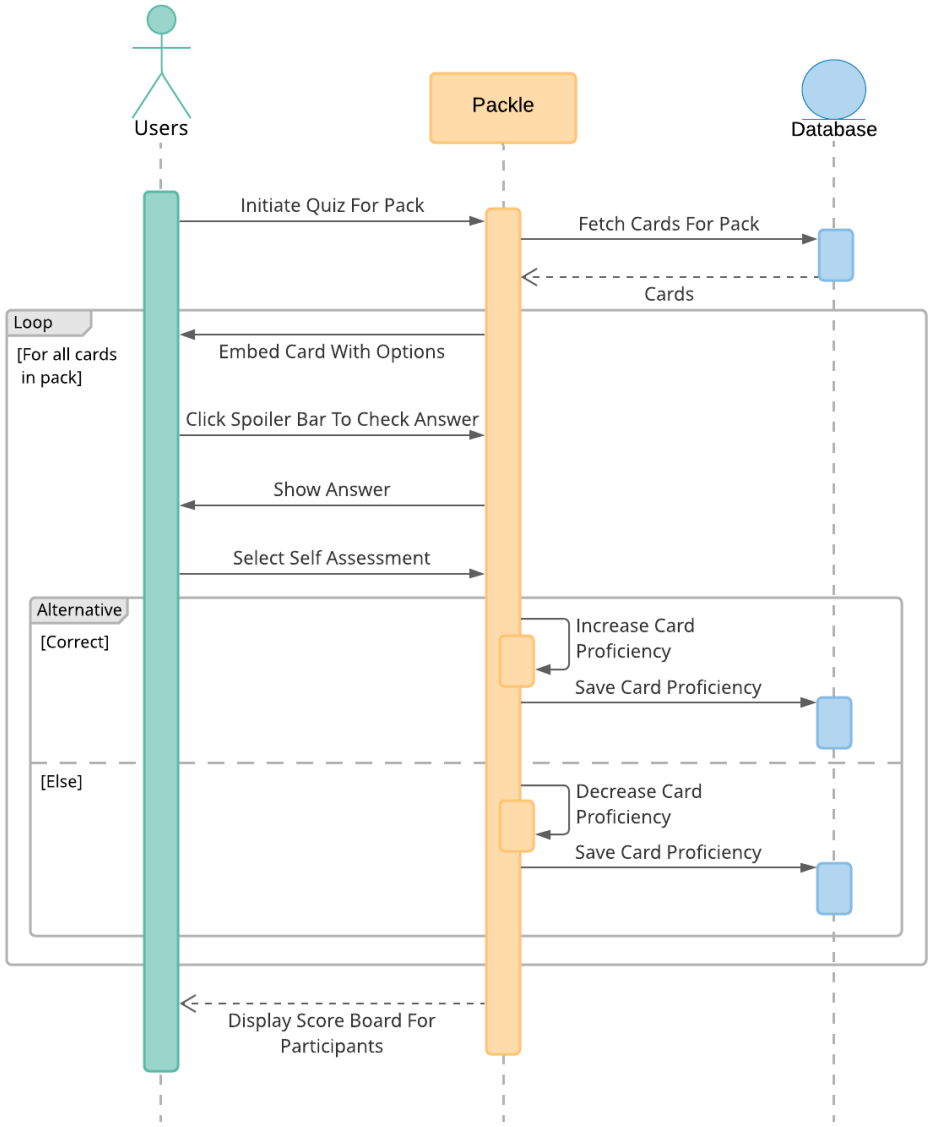


Figure 23: Sequence diagram for quiz mode

Figure 23 shows how a saved pack is used as a quiz. The user would initiate a quiz session by adding Packle to a Discord server. Packle will execute the quiz mode the same way study mode is conducted. The differences are that in the server, any of the users can participate and submit whether they got the answer right or wrong and quiz mode would not have a reminder as it is a one time event. The cards will also be subjected to a timer that the quiz initiator can set. Once all the cards have been looped through, Packle will tally the score and present a scoreboard with the results.

5.5. User Interface Design

Since Packle is a bot implemented on Discord, there were limitations to what could be implemented user interface (UI) wise. However, since UI design is an important part of development, Packle was carefully designed to be intuitive and easy for users with varying levels of experience with bots. For example, a minimalistic help menu will help users remember the Packle commands and the card has clickable stickers with intuitive emojis.

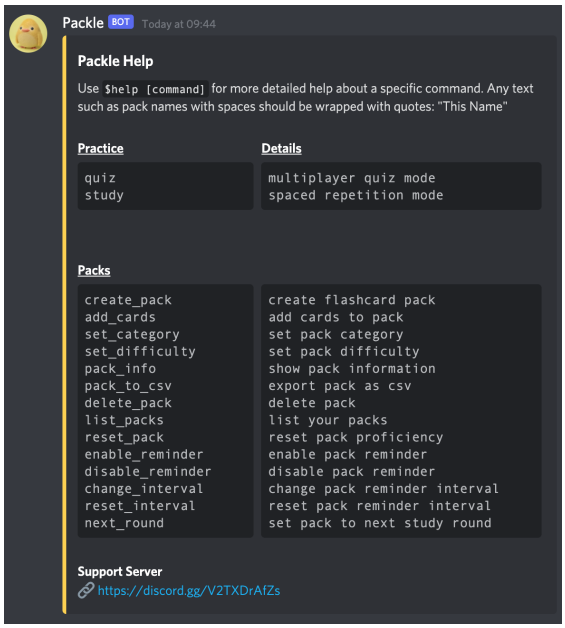


Figure 24: Help menu for Packle

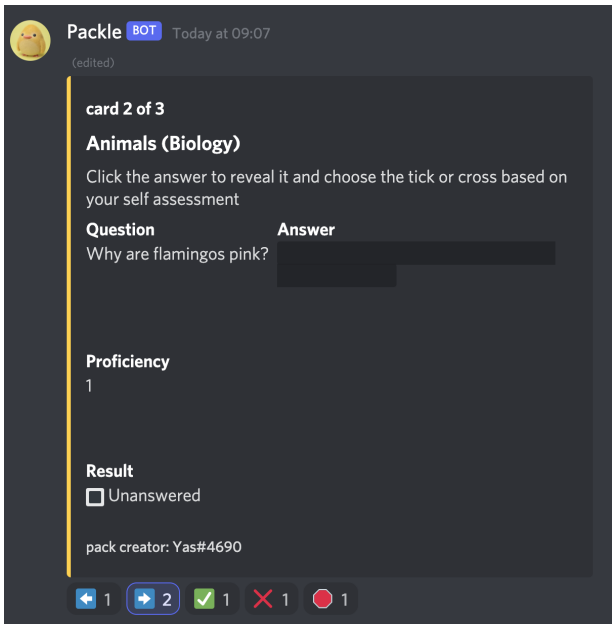


Figure 25: Card embed in Packle

5.5.1. Branding

The name ‘Packle’; a play on the use of packs in the system, was chosen since it is easy to pronounce and remember. The colour yellow was chosen due to how vivid, bright colours tend to catch attention (Kauppinen-Räisänen, 2014). The Packle mascot ‘Chonky’ was created to give Packle a trustworthy and friendly appearance (Cayla, 2013). Additionally, a fun easter egg was created for users to send the command ‘$quack’ to receive a randomised duck gif.

Figure 26: Packle’s mascot Chonky

5.6. Test design

Testing ensures whether the components that make up the functional requirements and any non-functional requirements that could possibly be tested work as intended or if the system behaves beyond intention. Table 4 was used to test as many cases as possible with an identifier, the case, expected outcome and actual outcome with a status to denote if the test passed or failed.

Table 4: Test table with sample

Identifier	Test Case	Expected Result	Actual Result	Status
FR01				
C01	Study cards	Packle should initiate studying	Packle initiates studying	Pass

6. Implementation & Development

The implementation of Packle targets three main goals. Make an application that:

- Increases student awareness on spaced learning and retesting,
- Is free
- Is available on a platform which is popular and suitable for education.

This was done in the form of a flashcard based bot with a spaced learning reminder that users can easily and freely access on Discord.

6.1. Python and discord.py

Initially the technology chosen for the project was React and Node. However, this does not fulfil the third target and further research done in section 2 showed evidence of Discord being a suitable platform. After considering the libraries discord.js and discord.py, the latter was chosen due to higher familiarity and the syntactical simplicity offered by Python. The project was built using python version 3.9.2 and discord.py 1.7.3. A Dictionary, which in python is called a ‘dict’ was used as a database to store all data as easily retrievable key / value pairs where the keys are unique (Python Software Foundation, 2021).

6.2. Commands

Discord bots work with the use of simple commands. These commands can initiate a task and accept parameters. The prefix chosen for Packle is ‘$’. All commands must start with the prefix to trigger Packle to perform a task. Code snippets for some main functions of the system can be found in Appendix I - L. Table 5 details the functions implemented in Packle.

Table 5: Functions Implemented

Command	Parameter(s)	Action
create_pack	pack_name: str, csv_text: str	Create a new flashcard pack by: - Checking if there is any viable csv text in the message text or if a csv is attached (and if both it will get merged) - Dict is created for the user if one doesn't exist and the pack is added to it. The dict allows users to use submitted packs in direct messages and any server where Packle exists. - Success message is sent and the initial request is deleted to avoid showing the user their answers
add_cards	pack_name: str, csv_text: str	Add cards to an existing flashcard pack by: - Checking if the pack exists - Checking if there is any viable csv text in the message text or if a csv is attached (and if both it will get merged) - Add card(s) into selected pack - Success message is sent and the initial request is deleted to avoid showing the user their answers
set_category	pack_name: str, category: str	Set category for the selected pack and send success message
set_difficulty	pack_name: str, difficulty: str	Set difficulty for the selected pack and send success message
list_packs	-	List all the flashcard packs that were added by the user as an embed
pack_info	pack_name: str	Show information about a flashcard pack as an embed
pack_to_csv	pack_name: str	Export existing flashcard pack as a CSV file by: - Building buffer - Create discord.File and send it along with pack information

reset_pack	pack_name: str	Set proficiency level of all cards in a pack to 1 and sends success message
delete_pack	pack_name: str	Delete the selected pack and sends success message
enable_reminder	pack_name: str	Enables spaced reminders for a pack and sends a success message. Default spaced repetition rate is proficiency one daily, proficiency two every three days and proficiency three every seven days.
disable_reminder	pack_name: str	Disables reminders for a pack and sends success message
change_interval	pack_name: str, seconds: float, minutes: float, hours: float	Changes the default spaced learning reminder from 24 hours to a custom value determined by seconds, minutes, and hours.
reset_interval	pack_name: str	Reset pack spaced reminder to default values
next_round	pack_name: str	Manually advance to the next round with cards
study	pack_name: str	Initiate a study session for the selected pack in a direct message with Packle. Packle would proceed by embedding the cards available for the day. The embed includes the question, hidden answer, and selected answer and attaches stickers that users can react to. These include: - Forward / Backward: Navigate cards - Tick / Cross: Self assessment of answer - Stop: Stop study session Packle will change the proficiency level of the cards depending on the answer selected by the user, remind the user how to restart a session if there is a timeout or end the session for the day once all cards have been studied

quiz	pack_name: str or pack_name: str, interval: float	Initiate a study session for the selected pack in a server with Packle. Packle would proceed by embedding the cards in the pack with 10 second intervals unless the quiz was initiated with a custom interval. The embed includes the question, hidden answer, and selected answer and attaches stickers that users can react to. These include: - Forward / Backward: Navigate cards - Tick / Cross: Self assessment of answer - Stop: Stop study session One or more users can react to the cards and once all the cards are done Packle will embed a highscore board with the results of the quiz which has the rank, name and score achieved
help	Null / another command	If help is sent with no parameters Packle will embed the general help menu with a list of all the commands with short descriptions. If the help command is sent with another command Packle will embed the detailed explanation for it

6.3. Error Handling

Error handling was implemented on all functionalities of the bot to assure that all errors were caught and that the user was made aware of what they were doing wrong. Packle will check to verify if the command is in the correct format, has the required parameters, and if the resources such as the selected pack exists before execution. Table 6 shows some of the errors that Packle will look for when the create_pack command is executed.

Table 6: Error Handling

Function	Error Checks
create_pack	- If the user did not submit a question/answer pair in the form of a text then it checks to see if the message has a CSV. If neither was submitted then the command is terminated - If the pack already exists return error - If the pack name is missing return error - If the format is wrong (eg. missing answer) return error

6.4. Packle in Use

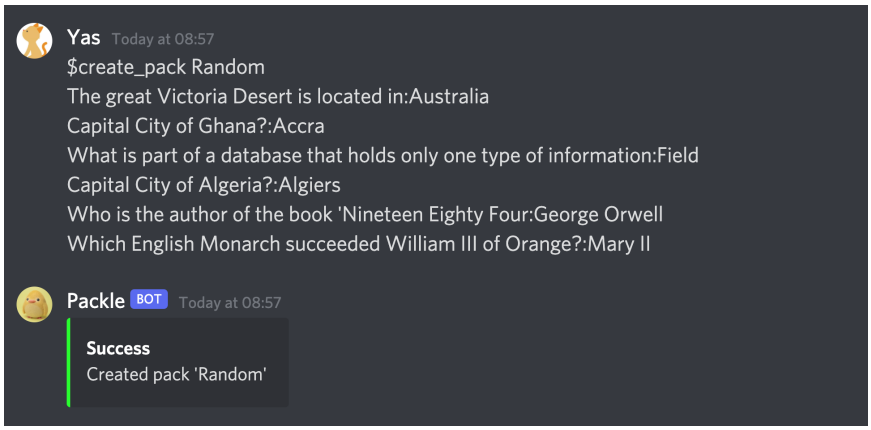


Figure 27: Creating pack with text

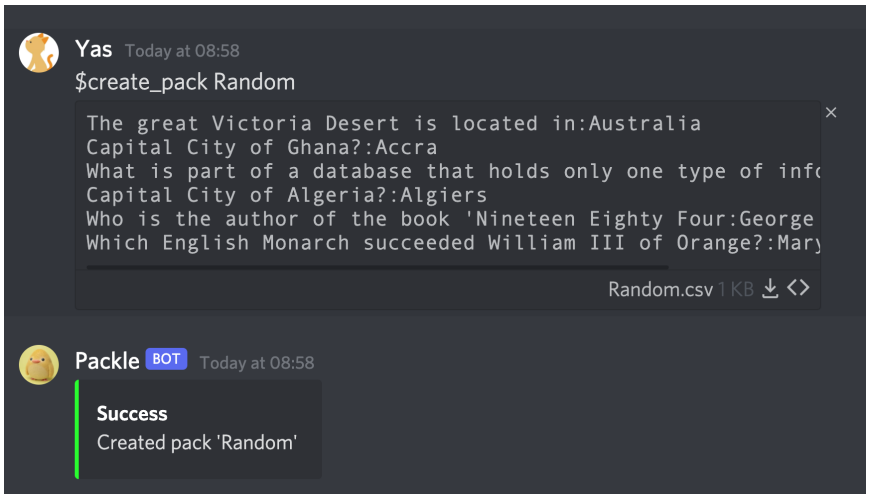


Figure 28: Creating pack with csv

Figure 29: Modifying and viewing pack

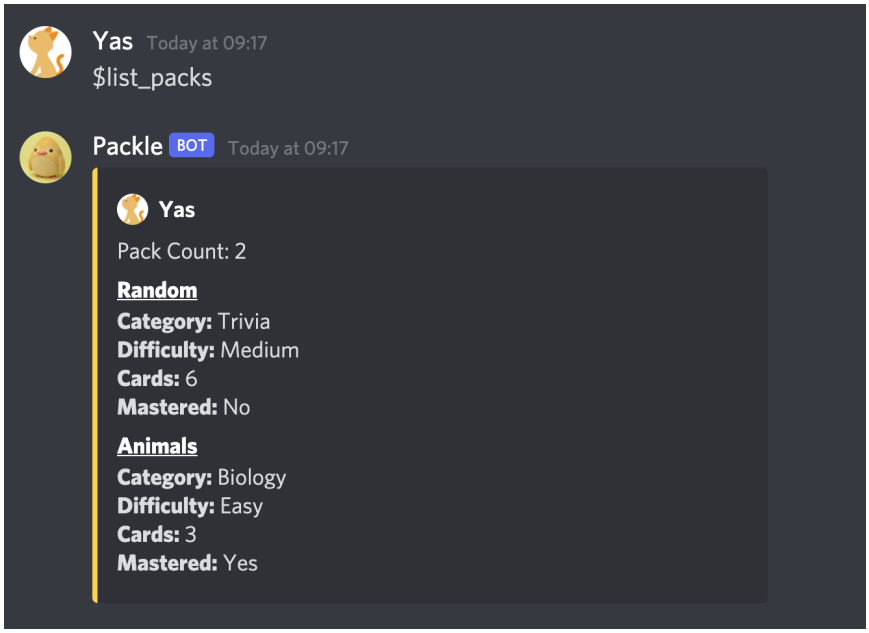


Figure 30: Listing packs

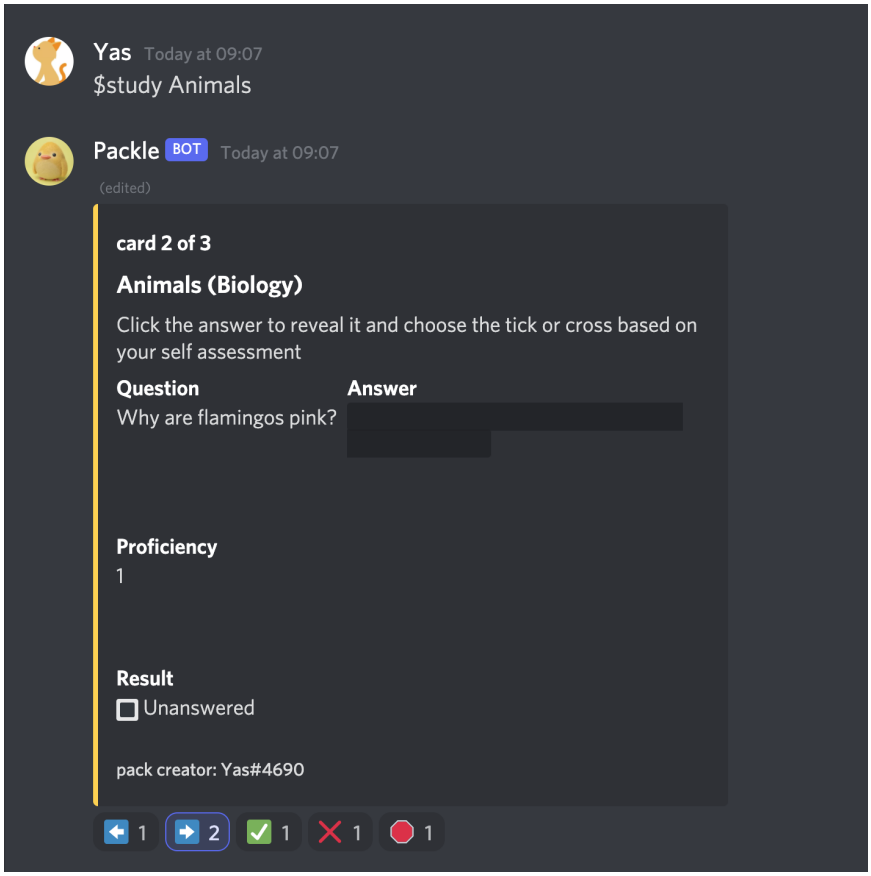


Figure 31: Studying pack

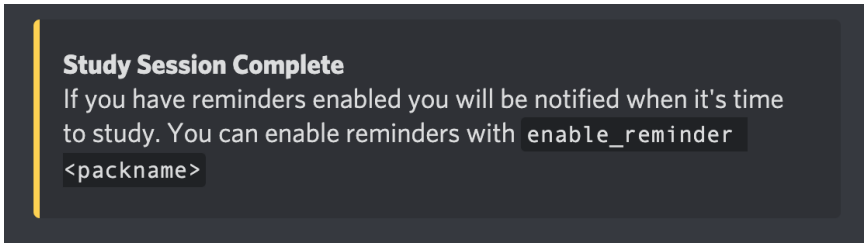


Figure 32: Notification for finishing a study session

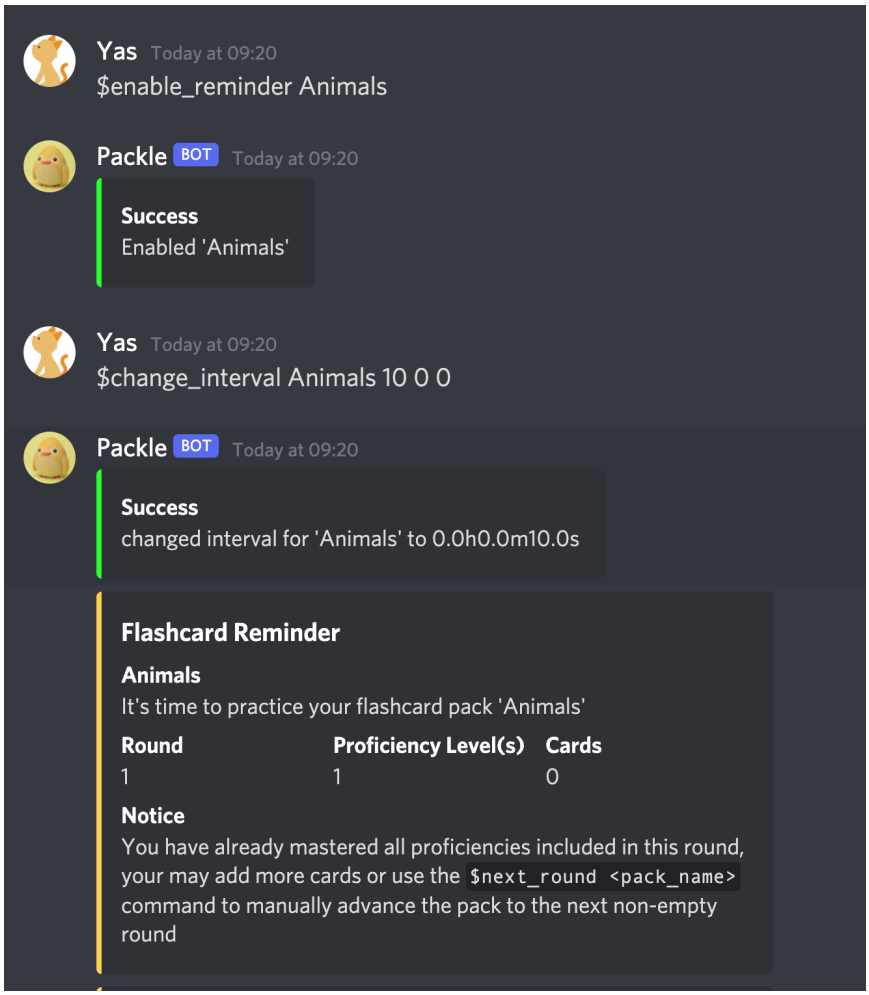


Figure 33: Enabling spaced learning reminder, customising it and receiving the reminder

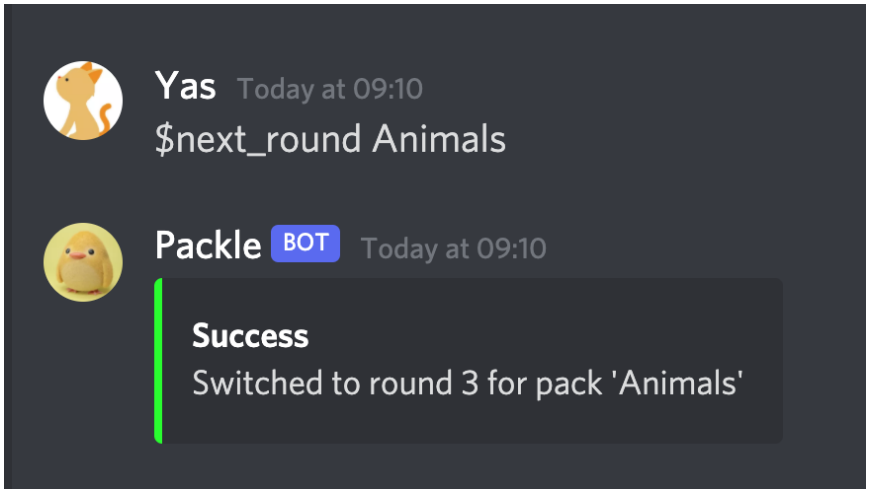


Figure 34: Forcing progress to next available round with cards

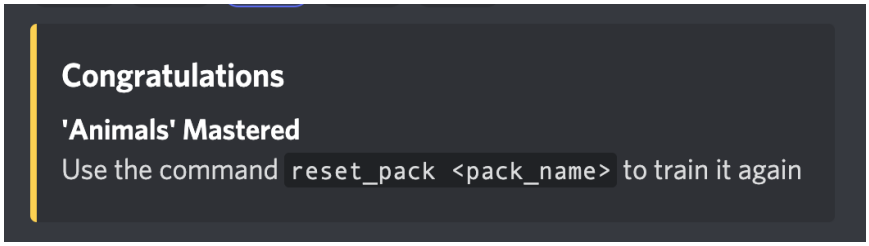


Figure 35: Notification for mastering a pack

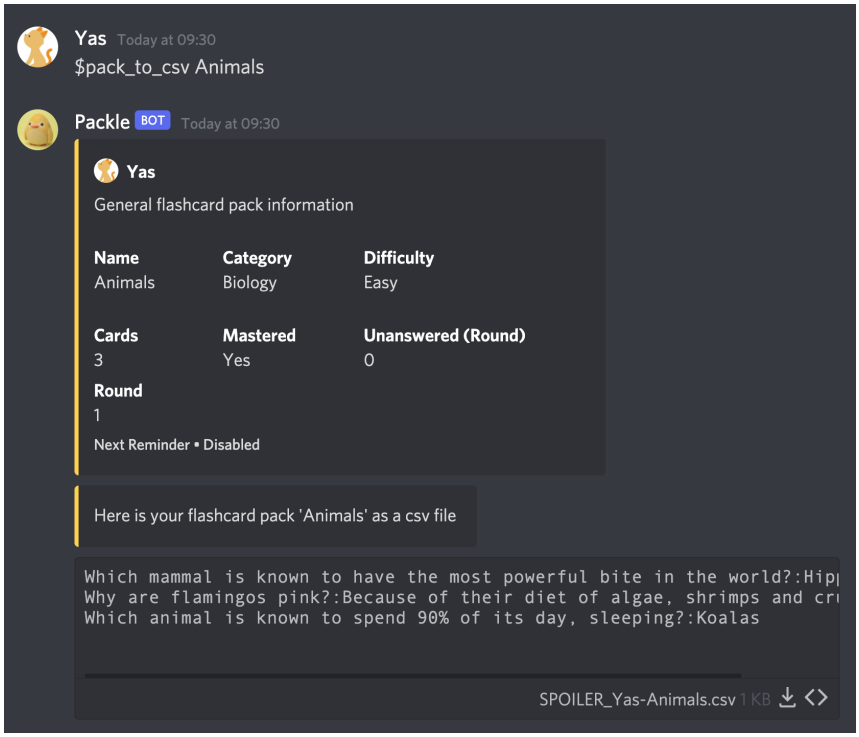


Figure 36: Export pack

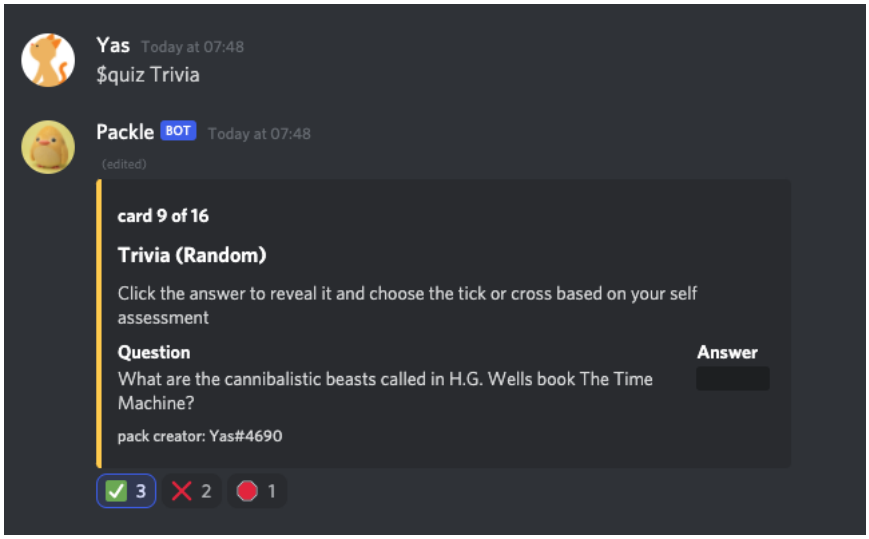
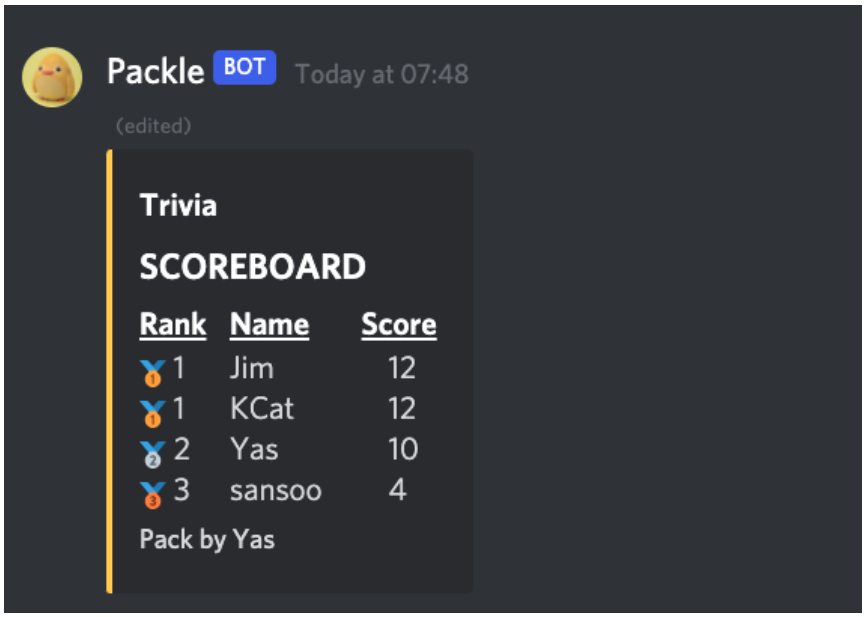


Figure 37: Quiz mode



(Names used with permission)

Figure 38: Quiz scoreboard with two users that got the same score

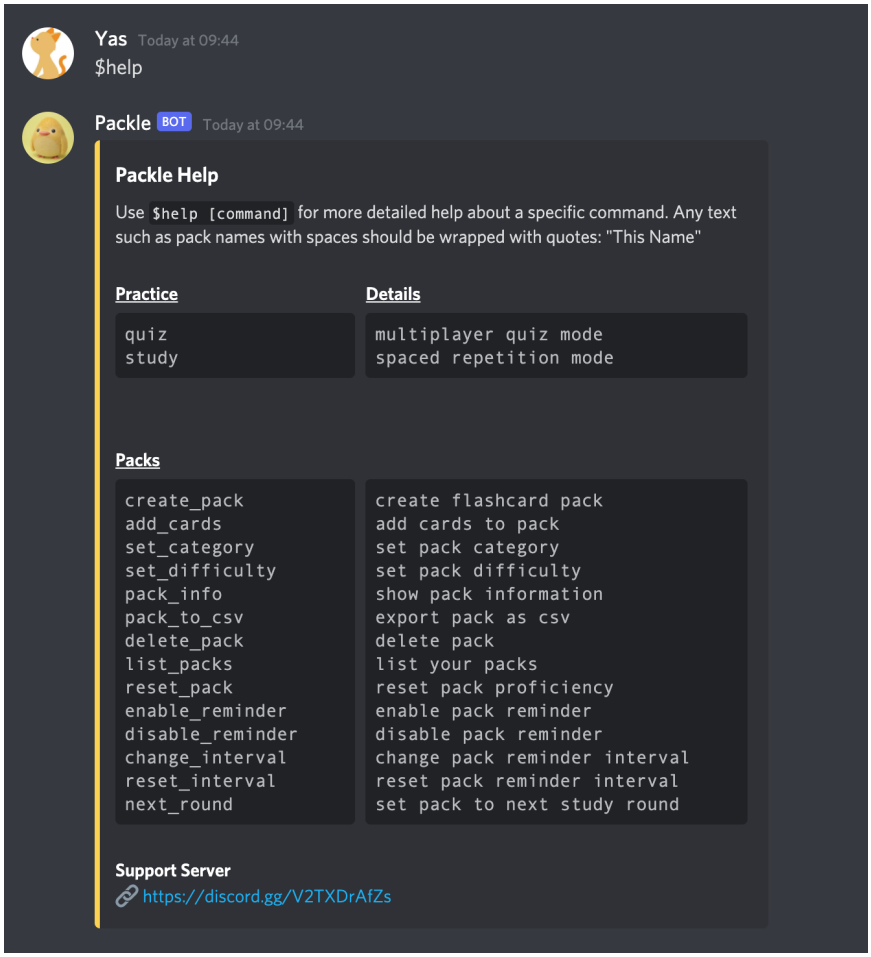


Figure 39: General help

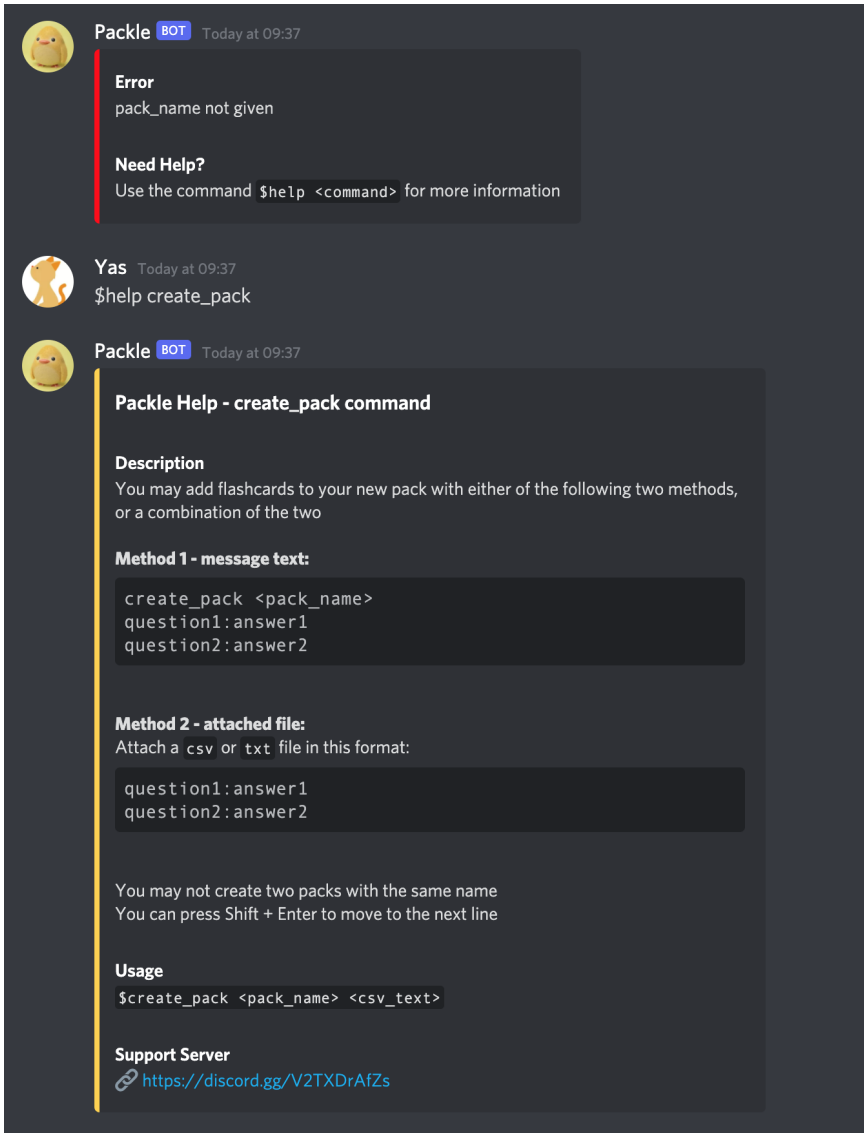


Figure 40: Error message and detailed help menu for the command

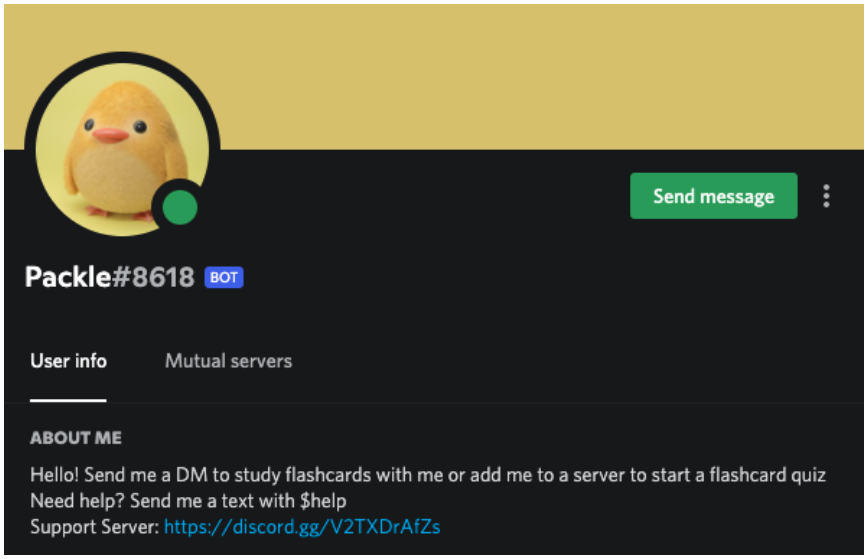


Figure 41: Packle with link to support server

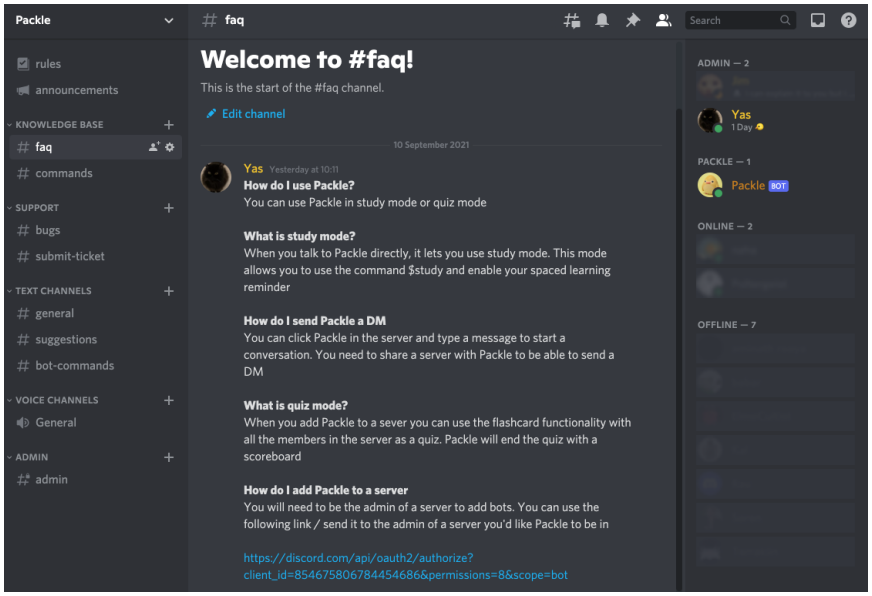


Figure 42: Packle support server

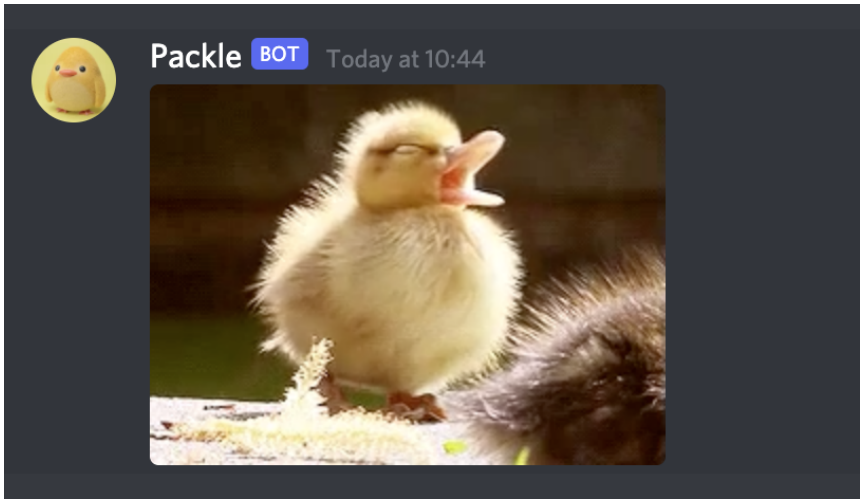


Figure 43: Easter egg that sends a random duck gif

6.5. Summary

The implementation of Packle on Discord has met all the targets successfully. Packle functions as an easy to use study tool which can benefit any current or new users of Discord that wish to use retesting and spaced learning for their study sessions.

7. Testing

Testing allows developers to improve the quality of a system. It is carried out by running the system with the intention of finding as many errors as possible. However, it would be impractical to create test cases for every possibility, therefore priority will be given to designing the cases with the best probability of detecting errors in the implementation of system requirements defined in section 3. The purpose of the test cases will be to examine if the system does not perform as expected but also to verify if the system performs a function it is not supposed to do (Myers, Badgett and Sandler, 2012, p.5-13).

7.1. Functional Requirements

Table 7: Functional Requirements Test Case Table

Identifier	Test Case	Expected Result	Actual Result	Status
FR01				
C01	Submit question/answer pair as text	Packle should create a pack	Packle creates a pack	Pass
C02	Submit question/answer pair as CSV	Packle should create a pack	Packle creates a pack	Pass
C03	Submit question/answer pair as text and CSV	Packle should create a pack with merged data	Packle creates a pack with merged data	Pass
C04	Submit question only	Packle should inform user that format is incorrect	Packle informs user that format is incorrect	Pass
C05	Submit question/answer pair as text into existing pack	Packle should add the card into the pack specified	Packle adds the card into the pack specified	Pass
C06	Export submitted pack as a CSV	Packle should export the selected pack with all original and edited content as a CSV file	Packle exports the selected pack with all original and edited content as a CSV file	Pass
FR02				
C07	Set pack category	Packle should update pack category	Packle updates pack category	Pass
C08	Set pack difficulty	Packle should update pack difficulty	Packle updates pack difficulty	Pass
C09	Change pack category	Packle should update pack category	Packle updates pack category	Pass
C10	Change pack difficulty	Packle should update pack difficulty	Packle updates pack difficulty	Pass
C11	Delete a pack	Packle should delete the selected pack	Packle deletes the selected pack	Pass
FR03				
C12	Send command to study a pack with pack name	Packle should initiate study mode for selected pack	Packle initiates study mode for selected pack	Pass

C13	Send command to study a pack without pack name	Packle should inform the user that pack name is missing	Packle informs the user that pack name is missing	Pass
C14	Send command to stop studying a pack	Packle should terminate the current study session	Packle terminates the current study session	Pass
C15	Set answer for card as correct	Packle should log the answer	Packle logs the answer	Pass
C16	Set answer for card as incorrect	Packle should log the answer	Packle logs the answer	Pass
C17	Not submit an answer to a card	Packle should time out after 120 seconds and inform the user	Packle times out after 120 seconds and informs the user	Pass
C18	Study multiple packs at the same time	Packle should allow users to study multiple packs at the same time	Packle allows users to study multiple packs at the same time	Pass
C19	Send command to study in a server instead of a direct message	Packle should notify the user that the study command is only for direct messages	Packle notifies the user that the study command is only for direct messages	Pass
FR04				
C20	Enable study reminder	Packle should enable the reminder	Packle enables the reminder	Pass
C21	Disable study reminder	Packle should deactivate the reminder	Packle disables the reminder	Pass
C22	Change reminder time frame	Packle should allow the user to set a custom reminder	Packle allows the user to set a custom reminder	Pass
C23	Reset reminder time frame	Packle should reset the spaced reminder to default	Packle resets the spaced reminder to default	Pass
FR05				
C24	Passage of time through reminder time frame set for a user with reminders enabled	Packle should send the relevant study reminder	Packle sends the relevant study reminder	Pass

C25	Passage of time through reminder time frame set for a user with reminders disabled	Packle should not send study reminders	Packle does not send study reminders	Pass
FR06				
C26	User reacts with ‘correct’ for a card	If proficiency < 3 Packle should increase the proficiency for the card by +1 and If proficiency = 3 the card should be moved to the ‘mastered’ deck	If proficiency < 3 Packle increases the proficiency for the card by +1 and If proficiency = 3 the card is moved to the ‘mastered’ deck	Pass
C27	User reacts with ‘incorrect’ for a card	If proficiency > 1 Packle should decrease the proficiency for the card by -1 and If proficiency = 1 the card should remain at proficiency 1	If proficiency > 1 Packle decreases the proficiency for the card by -1 and If proficiency = 1 the card remains at proficiency 1	Pass
C28	User studies all cards in the pack for the day	Packle should end the study session	Packle ends the study session	Pass
FR07				
C29	User lists packs	Packle should list packs	Packle lists packs	Pass
C30	Send command to display pack information	Packle should display information for selected pack	Packle displays information for selected pack	Pass
FR08				
C31	Send command to start a quiz with pack name	Packle should initiate quiz for selected pack	Packle initiates quiz for selected pack	Pass
C32	Send command to start a quiz without pack name	Packle should inform the user that pack name is missing	Packle informs the user that pack name is missing	Pass
C33	Send command to stop quiz	Packle should terminate the current quiz	Packle terminates the current quiz at end of posted question	Pass
C34	Participants click ‘correct’ for a card	Packle should log the answers for all participants	Packle logs the answers for all participants	Pass
C35	Participants click	Packle should log the	Packle logs the	Pass

	‘incorrect’ for a card	answers for all participants	answers for all participants	
C36	No participants submit any answers	Packle should time out after going through all cards	Packle times out after going through all cards	Pass
C37	Send command to start a quiz in direct messages instead of a server	Packle should notify the user that the quiz command can only be used in servers	Packle notifies the user that the quiz command can only be used in servers	Pass
FR09				
C38	1 participant submits answers until quiz is completed	Packle should tally the score and display the scoreboard for a single user	Packle tallies the score and displays the scoreboard for a single user	Pass
C39	> 1 participant submits answers until quiz is completed	Packle should tally the scores and display the scoreboard. Users with equal scores will get the same rank number	Packle tallies the scores and displays the scoreboard. Users with equal scores get the same rank number	Pass
FR10				
C40	Spaced learning logic for reminder system	The default reminders for studying using packle should be daily for proficiency 1, on the 3rd day for proficiency 1 and 2 and on the 7th day for proficiency 1, 2, and 3	The default reminders for studying using packle is daily for proficiency 1, on the 3rd day for proficiency 1 and 2 and on the 7th day for proficiency 1, 2, and 3	Pass
C41	Retesting logic for cards	Packle should output questions with hidden answers to allow users to attempt retrieval of the answer before self assessing	Packle outputs questions with hidden answers to allow users to attempt retrieval of the answer before self assessing	Pass
C42	Leitner system based logic for card proficiencies	Packle should initiate all cards as proficiency one which should increment or decrement based on user responses	Packle initiates all cards as proficiency one which increments or decrements based on user responses	Pass

7.2. Non-Functional Requirements

Additional tests were done on the non-functional requirements which could be tested to some level. However, as highlighted in section 3, these are characteristics of the system which are difficult to test. While it cannot be entirely confirmed if a goal was truly met, doing some tests allows us to understand if the bot can be used effectively by students when it is deployed.

Table 8: Non-functional Requirements Test Case Table

Identifier	Test Case	Expected Result	Actual Result	Status
NFR02				
C43	Multiple users interact with Packle simultaneously	Packle should be able to handle requests from multiple users without crashing	Packle is able to handle requests from multiple users without crashing	Pass
NFR03				
C44	Add 10 different packs in with a minimum of 10 questions per pack	Packle should be able to store all 10 packs	Packle stores all 10 packs	Pass
NFR06				
C45	Export a pack as a CSV	Packle should attach a plain CSV file	Packle attaches a plain CSV file	Pass
NFR07				
C46	Enter commands to interact with Packle	Packle should output the relevant response in about 3 seconds without a noticeable delay	Packle outputs a relevant response in less than 3 seconds without a noticeable delay	Pass
NFR10				
C47	Submit the help command	Packle should output clear instructions with a link to the support server	Packle outputs clear instructions with a link to the support server	Pass

7.3. Reflection of Results

As discussed in section 4, the system was developed with frequent testing to ensure that the implemented functions worked as expected, therefore the completed system performed as expected. The only changes made after testing was changing some messages so that they were more concise and changing some of the timings, since the testing provided some insight into real use.

8. Evaluation

With the implementation of all the functional requirements and meeting the objectives of the project, Packle's implementation was a great success.

8.1. Research

A majority of the project time was spent on research due to the vast number of studies available for spaced learning and retesting. It was challenging to identify the most relevant content to include in the report. The research also ensured the suitability of Discord for the project.

The research also allowed solid foundations on which Packle could be built such as determining the best rate of repetition. While further research would need to be done to understand if the logic chosen improves the retention capabilities of students, I am confident that based on the evidence found regarding the chosen rate, students that stick to the default study reminders would benefit greatly from it. The research also helped me understand that having an extensive help menu and support server would be crucial to Packle's success. This was reflected in the results of the survey where a trend was detected between familiarity with bots and willingness to use it.

8.2. Requirements

The survey helped to identify student habits and along with the research done, it was easy to make a scope for Packle in such a way that all the project goals could be accomplished.

The only limitation was that Packle was being implemented as a bot on an application and this had to be taken into consideration when determining the requirements.

8.3. Methodology

The adjusted waterfall methodology chosen for the project worked well since the functional and non-functional requirements for the project were static. The goals were well defined from the beginning and therefore it was easy to go through the stages to completion.

8.4. Design, Implementation & Testing

Since user experience was a priority, Packle was designed to cater to users with the least possible level of experience with Discord and bots. Extensive error handling will ensure that students are aware of what they did incorrectly or what they can do to achieve their targets. The commands are also built in such a way that students get multiple options such as in the case of creating packs with text, csv and both combined.

The process of implementing Packle was an interesting challenge. The choice of using discord.py allowed Packle to be developed in a much shorter time frame than it would have taken if I had built Packle as a web application as initially planned. Additionally, the extensive testing carried out increased my confidence in Packle before its release.

Packle functions extremely well with a fast response rate even when simultaneous users are interacting with it. One of the issues identified during implementation and testing was that since the Leitner system depends on certain proficiency levels for different days, we may have an edge case where a student has no cards for a day. This issue was quickly rectified with the implementation of the ‘next round’ command so that students who did not want to spend a day without studying were prompted to either add new cards or force an advancement.

8.5. Further Work

Although a dict being used to store data works within the confines of the time available and the project being a prototype, the most crucial next step for Packle would be to implement a method of data persistence so that it does not lose user data. Packle needs better hosting, preferably on a cloud server so that it is always available for users. Since Packle has a support server, user feedback will be the most useful resource by which the future enhancements of Packle will be decided and implemented.

8.6. Feedback

My supervisor was a great source of support through the project. As a stakeholder for the project, her insight was very valuable. Meetings were held on a need-to basis and the feedback received helped prioritise tasks and keep the project on the right track.

9. Conclusion

Packle was developed to allow students to have access to an effective method of learning, provided on a free platform that they were familiar with in a way that was intuitive for users of varying levels of experience. Extensive research was done to understand the findings of studies and a survey was conducted to collect additional information. Based on this the requirements for the project were determined and a Gantt chart was created to schedule the project.

The project was then designed using relevant UML diagrams so that the architecture of the system could be determined before implementation. This preparation helped to make the implementation run smoothly and since bug fixing was run simultaneously, the project successfully passed the testing phase. All of the requirements for the project and its main targets were accomplished leading to a successful implementation of a learning tool that will allow students to incorporate retesting and spaced learning into their study sessions.

10. References

1. Al-jumeily, D., Hussain, A.J., Abuelmaatti, O., Fergus, P. and Lunn, J. (2016) An Open Learning System For Special Needs Education. *Knowledge Management & E-learning* [online]. 8 (1), pp. 68-85. [Accessed 20 June 2021].
2. Apperly, H., Hofman, R., Latchem, S., Maybank, B., McGibbon, B., Piper, D. and Simons, C. (2003) *Service- and Component-based Development: Using Select Perspective and UML* [online]. Great Britain, UK: Pearson Education Limited. [Accessed 25 July 2021].
3. Ashcroft, R.J., Cvitkovic, R. and Praver, M. (2018) Digital Flashcard L2 Vocabulary Learning Out-performs Traditional Flashcards at Lower Proficiency Levels: A Mixed-methods Study of 139 Japanese University Students. *The EuroCALL Review* [online]. 26 (1), pp. 14-28. [Accessed 19 July 2021].
4. Bernad, B. (2021) Discord to Support Synchronous Communication in Distance Learning, *Proceedings of the 2nd Annual Conference on blended learning, educational technology and Innovation (ACBLETI 2020)*, Padang, Indonesia, 23-24 October 2020. Atlantis Press [online]. Available from: https://www.atlantis-press.com/proceedings/acbleti-20/125957911 [Accessed 16 August 2021].
5. Bass, L., Clements, P. and Kazman, R. (2021) *Software Architecture in Practice* [online]. 4th ed. USA: Addison-Wesley Professional. [Accessed 24 July 2021].
6. Cayla, J. (2013) Brand Mascots as Organisational Totems. *Journal of Marketing Management* [online]. 29 (1), pp. 86-104. [Accessed 20 July 2021].
7. Chen, R.W. and Chan, K.K. (2019) Using Augmented Reality Flashcards to Learn Vocabulary in Early Childhood Education. *Journal of Educational Computing Research* [online]. 57 (7), pp. 1812-1831. [Accessed 19 July 2021].
8. Danica, S. (2018) Teaching with an Index Card: The Benefits of Free, Open-source Tools. *The Chronicle of Higher Education* [online]. 65 (8), p. B32. [Accessed 17 August 2021].
9. Disboard (2021) *Discord Servers.* Available from: https://disboard.org/servers/tag/study [Accessed 20 July 2021].
10. Discord (2021a) *Nitro.* Available from: https://discord.com/nitro [Accessed 20 July 2021].
11. Discord (2021b) *Our Metrics.* Available from: https://discord.com/company [Accessed 20 July 2021].
12. Discord Developer Portal (2021) *Documentation.* Available from: https://discord.com/developers/docs/intro [Accessed 12 August 2021].

13. Ebbinghaus, H. (1913). Memory: A Contribution to Experimental Psychology. Translated from the German by Henry A. Ruger and Clara E. Bussenius. New York: Teachers College, Columbia University.
14. Flashcard Machine (2021) *Welcome to Flashcard Machine.* Available from: https://www.flashcardmachine.com/ [Accessed 19 May 2021].
15. Fowler, M. (2004) *UML Distilled: A Brief Guide to the Standard Object Modeling Language* [online]. 3rd ed. Boston: Pearson Education. [Accessed 24 July 2021].
16. Giachetti, R.E. (2010) *Design of Enterprise Systems: Theory, Architecture, and Methods* [online]. Florida, USA: CRC Press. [Accessed 26 July 2021].
17. Goossens, N.A.M.C., Camp, G., Verkoeijen, P.P.J.L. and Tabbers, H.K. (2014) The Effect of Retrieval Practice in Primary School Vocabulary Learning. *Applied Cognitive Psychology* [online]. 28 (1), pp. 135-142. [Accessed 19 July 2021].
18. Goossens, N.A.M.C., Camp, G., Verkoeijen, P.P.J.L., Tabbers, H.K., Bouwmeester, S. and Zwaan, R.A. (2016) Distributed Practice and Retrieval Practice in Primary School Vocabulary Learning: A Multi-classroom Study. *Applied Cognitive Psychology* [online]. 30 (5), pp. 700-712. [Accessed 27 July 2021].
19. Hinze, S.R. and Rapp, D.N. (2014) Retrieval (Sometimes) Enhances Learning: Performance Pressure Reduces the Benefits of Retrieval Practice. *Applied Cognitive Psychology* [online]. 28 (4), pp. 597-606. [Accessed 14 July 2021].
20. Hwang, G. and Chang, S. (2016) Effects of a Peer Competition-based Mobile Learning Approach on Students' Affective Domain Exhibition in Social Studies Courses. *British Journal of Educational Technology* [online]. 47 (6), pp. 1217-1231. [Accessed 21 July 2021].
21. Kang, S.H.K. (2016) Spaced Repetition Promotes Efficient and Effective Learning: Policy Implications For Instruction. *Policy Insights From the Behavioral and Brain Sciences* [online]. 3 (1), pp. 12-19. [Accessed 19 July 2021].
22. Karpicke, J.D. and Roediger, H.L.R. (2008) The Critical Importance of Retrieval For Learning. *Science* [online]. 319 (5865), pp. 966-968. [Accessed 19 June 2021].
23. Karpicke, J.D., Blunt, J.R. and Smith, M.A. (2016) Retrieval-based Learning: Positive Effects of Retrieval Practice in Elementary School Children. *Frontiers in Psychology* [online]. 7, p. 350. [Accessed 19 July 2021].
24. Kauppinen-Räisänen, H. (2014) Strategic Use of Colour in Brand Packaging. *Packaging Technology and Science* [online]. 27 (8), pp. 663-676. [Accessed 20 July 2021].
25. Kiene, C. (2020) *Challenges and Adaptations to Technological Change in Online Communities* [online]. MA, University of Washington. Available from:

https://digital.lib.washington.edu/researchworks/handle/1773/45469 [Accessed 02 August 2021].

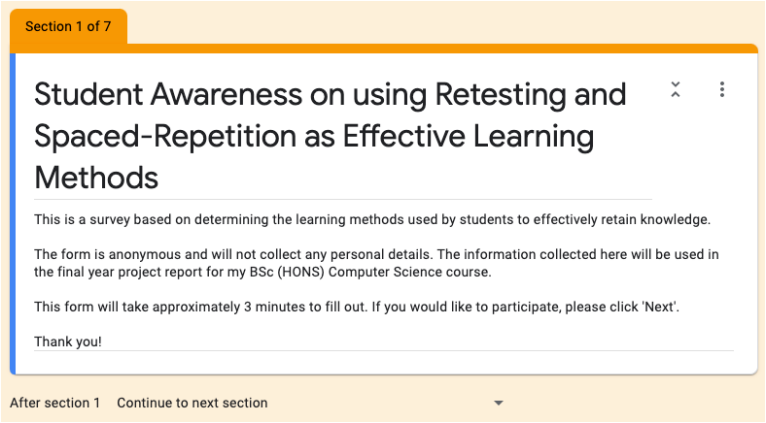
26. Lambers, A. and Talia, A.J. (2021) Spaced Repetition Learning as a Tool For Orthopedic Surgical Education: A Prospective Cohort Study on a Training Examination. *Journal of Surgical Education* [online]. 78 (1), pp. 134-139. [Accessed 19 July 2021].
27. Martinez, M.E. (2012) Human Memory the Basics. *Phi Delta Kappan* [online]. 91 (8), pp. 62-65. [Accessed 10 June 2021].
28. McConnell, M.M., St-Onge, C. and Young, M.E. (2015) The Benefits of Testing For Learning on Later Performance. *Advances in Health Sciences Education* [online]. 20 (1), pp. 305-320. [Accessed 25 July 2021].
29. Memory (no date) *Welcome to Memory.* Available from: https://www.memory.com/ [Accessed 19 May 2021].
30. Meyer, B. (2014) *Agile! the Good, the Hype and the Ugly* [online]. Switzerland: Springer International Publishing. [Accessed 23 July 2021].
31. Murre, J.M.J. and Dros, J. (2015) Replication and Analysis of Ebbinghaus' Forgetting Curve. *PLoS ONE* [online]. 10 (7) [Accessed 13 July 2021].
32. Myers, G.J., Badgett, T. and Sandler, C. (2012) *The Art of Software Testing* [online]. 3rd ed. New Jersey, USA: John Wiley & Sons, Inc. [Accessed 25 July 2021].
33. Odinokaya, M.A., Krylova, E.A., Rubtsova, A.V. and Almazova, N.I. (2021) Using the Discord Application to Facilitate Efl Vocabulary Acquisition. *Education Sciences* [online]. 11 (9) [Accessed 18 August 2021].
34. Olesea, C. (2019) The Concept of Personal Learning Pathway For Intelligent Tutoring System GeoMe. *Computer Science Journal of Moldova* [online]. 27 (3), pp. 355-367. [Accessed 02 July 2021].
35. Pastötter, B. and Bäuml, K.T. (2014) Retrieval Practice Enhances New Learning: The Forward Effect of Testing. *Frontiers in Psychology* [online]. 5, p. 286. [Accessed 19 July 2021].
36. Pyc, M.A. and Rawson, K.A. (2010) Why Testing Improves Memory: Mediator Effectiveness Hypothesis. *Science* [online]. 330 (6002), p. 335. [Accessed 19 June 2021].
37. Python Software Foundation (2021) *Built-in Types.* Available from: https://docs.python.org/3/library/stdtypes.html#typesmapping [Accessed 10 August 2021].
38. Quizlet (2021) *Become your most unstoppable self.* Available from: https://quizlet.com/en-gb [Accessed 19 May 2021].
39. Rapptz (2021) *API Reference.* Available from: https://discordpy.readthedocs.io/en/stable/api.html [Accessed 15 August 2021].

40. Roediger, H.L. and Karpicke, J.D. (2006) Test-enhanced Learning: Taking Memory Tests Improves Long-term Retention. *Psychological Science* [online]. 17 (3), pp. 249-255. [Accessed 19 July 2021].
41. Rowsell, E., Morrell, E. and Alvermann, D.E. (2017) Confronting the Digital Divide: Debunking Brave New World Discourses. *The Reading Teacher* [online]. 71 (2), pp. 157-165. [Accessed 13 August 2021].
42. Sage, K., Piazzini, M., Downey, J.C. and Ewing, S. (2020) Flip It Or Click It: Equivalent Learning of Vocabulary From Paper, Laptop, and Smartphone Flashcards. *Journal of Educational Technology Systems* [online]. 49 (2), pp. 145-169. [Accessed 19 July 2021].
43. Schmidmaier, R., Ebersbach, R., Schiller, M., Hege, I., Holzer, M. and Fischer, M.R. (2011) Using Electronic Flashcards to Promote Learning in Medical Students: Retesting Versus Restudying. *Medical Education* [online]. 45 (11), pp. 1101-1110. [Accessed 19 July 2021].
44. Shail, M.S. (2019) Using Micro-learning on Mobile Applications to Increase Knowledge Retention and Work Performance: A Review of Literature. *Cureus* [online]. 11 (8), p. e5307. [Accessed 12 July 2021].
45. Soderstrom, N.C., Kerr, T.K.K. and Bjork, R.A.B. (2016) The Critical Importance of Retrieval—and Spacing—for Learning. *Psychological Science* [online]. 27 (2), pp. 223-230. [Accessed 19 July 2021].
46. Sommerville, I. (2016) *Software Engineering* [online]. 10th ed. Essex, England: Pearson Education Limited. [Accessed 24 July 2021].
47. StudyStack (2021) *Search free flashcards shared by students and teachers.* Available from: https://www.studystack.com/ [Accessed 19 May 2021].
48. Tossell, C.C., Kortum, P., Shepard, C., Rahmati, A. and Zhong, L. (2015) You Can Lead a Horse to Water But You Cannot Make Him Learn: Smartphone Use in Higher Education. *British Journal of Educational Technology* [online]. 46 (4), pp. 713-724. [Accessed 20 July 2021].
49. Vladoiu, M. and Constantinescu, Z.C. (2020) Learning During COVID-19 Pandemic: Online Education Community, Based on Discord, *2020 19th RoEduNet Conference: Networking in Education and Research (RoEduNet)*, Bucharest, Romania, 11-12 December 2020. IEEE [online]. Available from: https://ieeexplore.ieee.org/document/9324863 [Accessed 10 August 2021].
50. Voice, A. and Stirton, A. (2020) Spaced Repetition: Towards More Effective Learning in Stem. *New Directions in the Teaching of Physical Sciences* [online]. 15 (1) [Accessed 19 July 2021].
51. Wahyuningsih, E. and Baidi, B. (2021) Scrutinizing the Potential Use of Discord Application as a Digital Platform Amidst Emergency Remote Learning. *Journal of Educational Management and Instruction* [online]. 1 (1), pp. 9-18. [Accessed 17 August 2021].

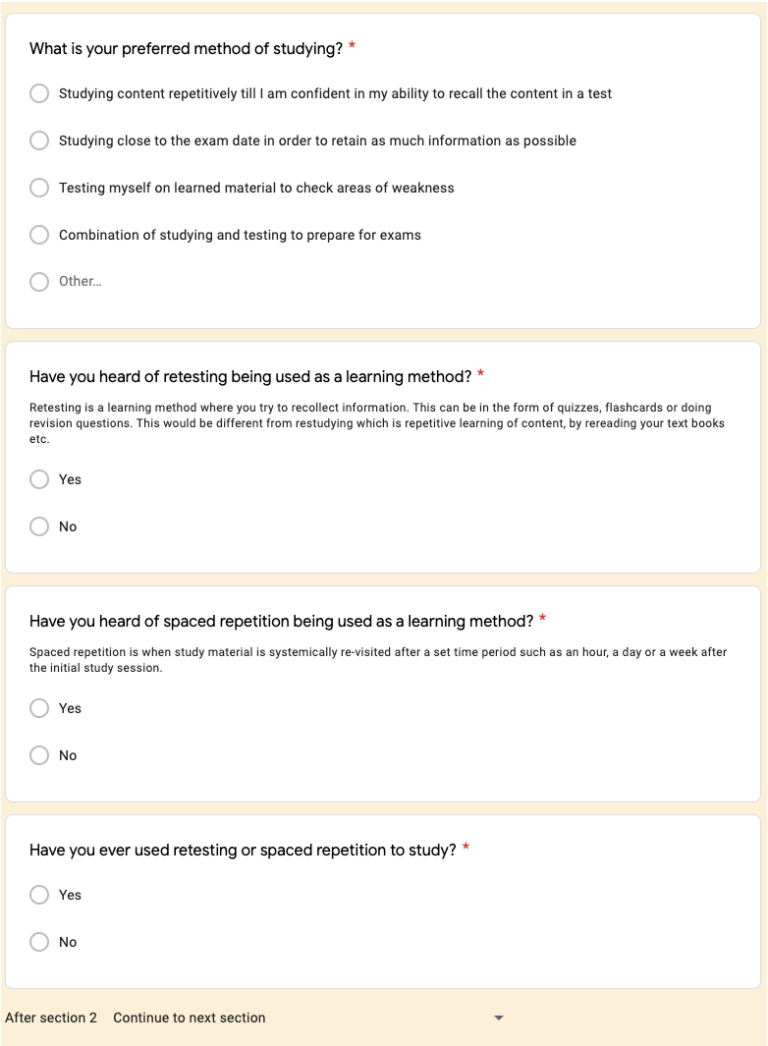
52. Whelan, J. (2019) *Using the Leitner System to improve your study.* Available from: https://jessewhelan.medium.com/using-the-leitner-system-to-improve-your-study-d5edafae7f0 [Accessed 19 July 2021].
53. Whitman, A.C., Tanzer, K. and Nemec, E.C. (2019) Gamifying the Memorization of Brand/generic Drug Names. *Currents in Pharmacy Teaching and Learning* [online]. 11 (3), pp. 287-291. [Accessed 21 July 2021].

11. Appendices

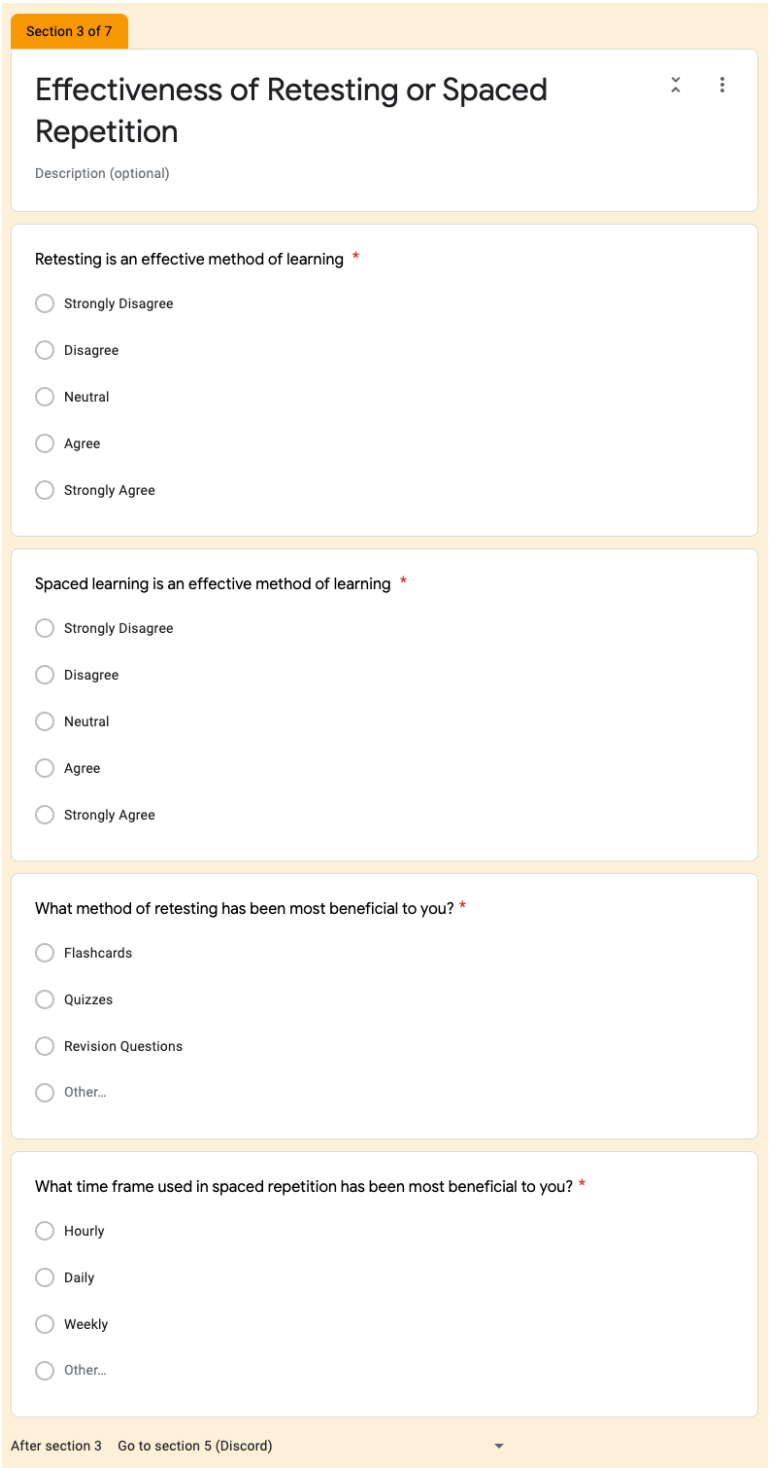
Appendix A: Questionnaire Section 1



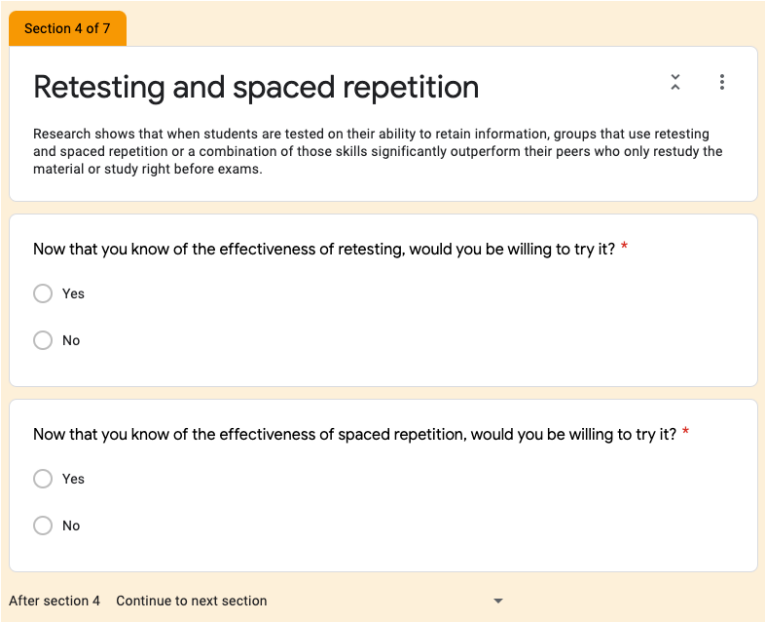
Appendix B: Questionnaire Section 2



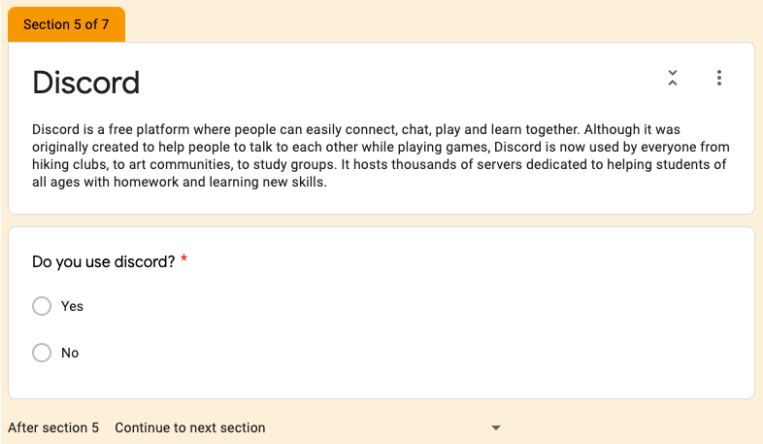
Appendix C: Questionnaire Section 3



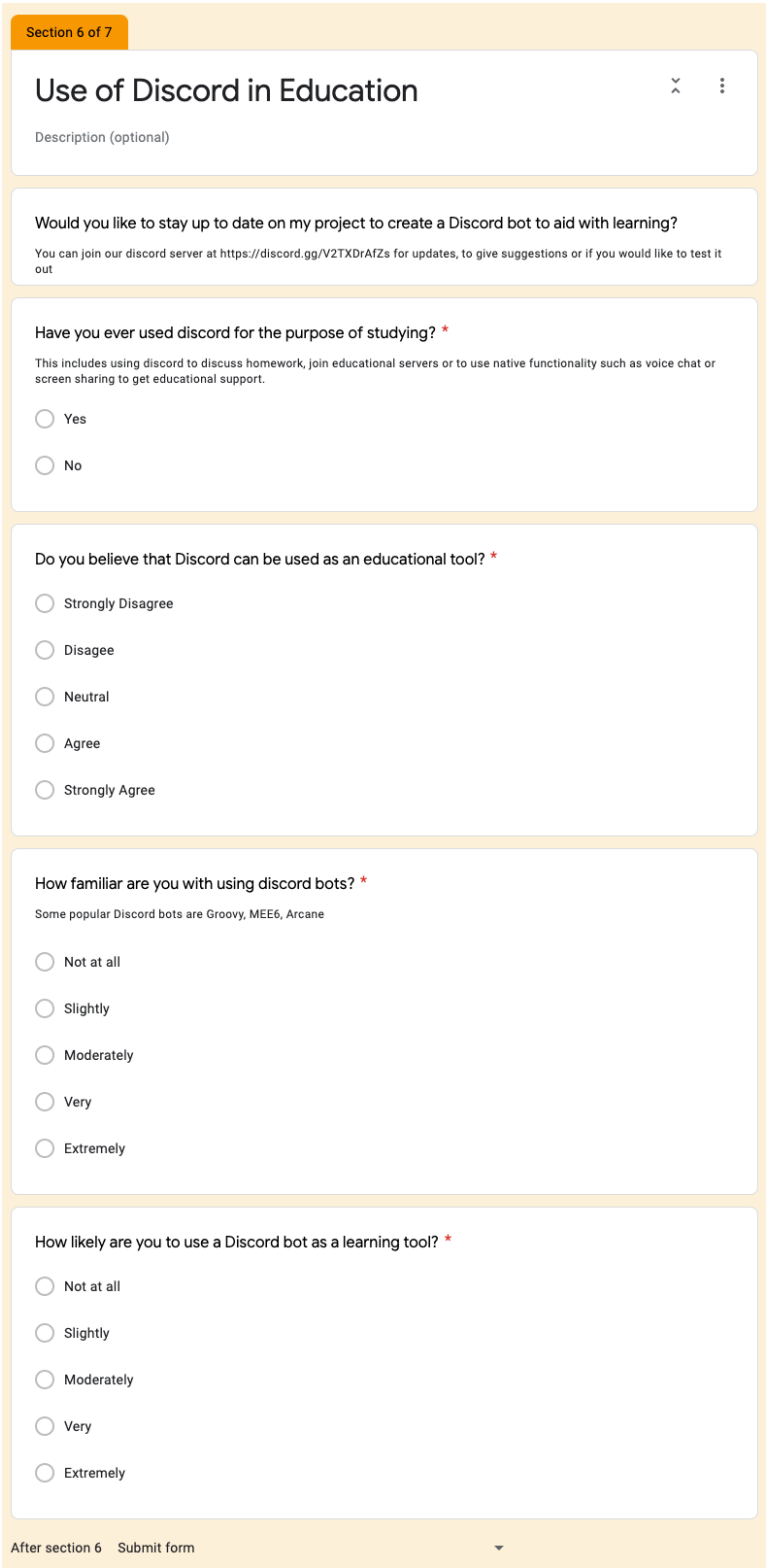
Appendix D: Questionnaire Section 4



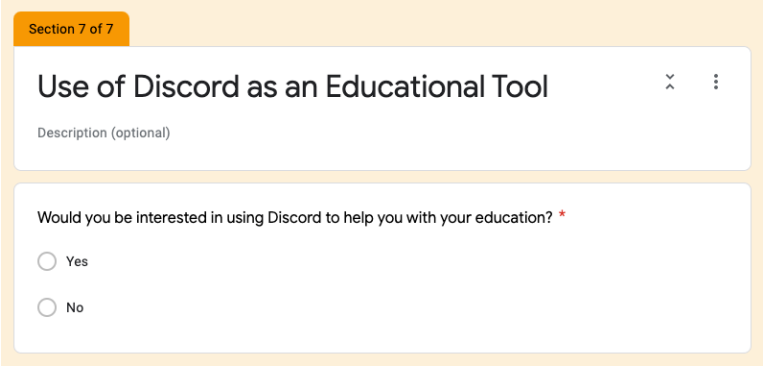
Appendix E: Questionnaire Section 5



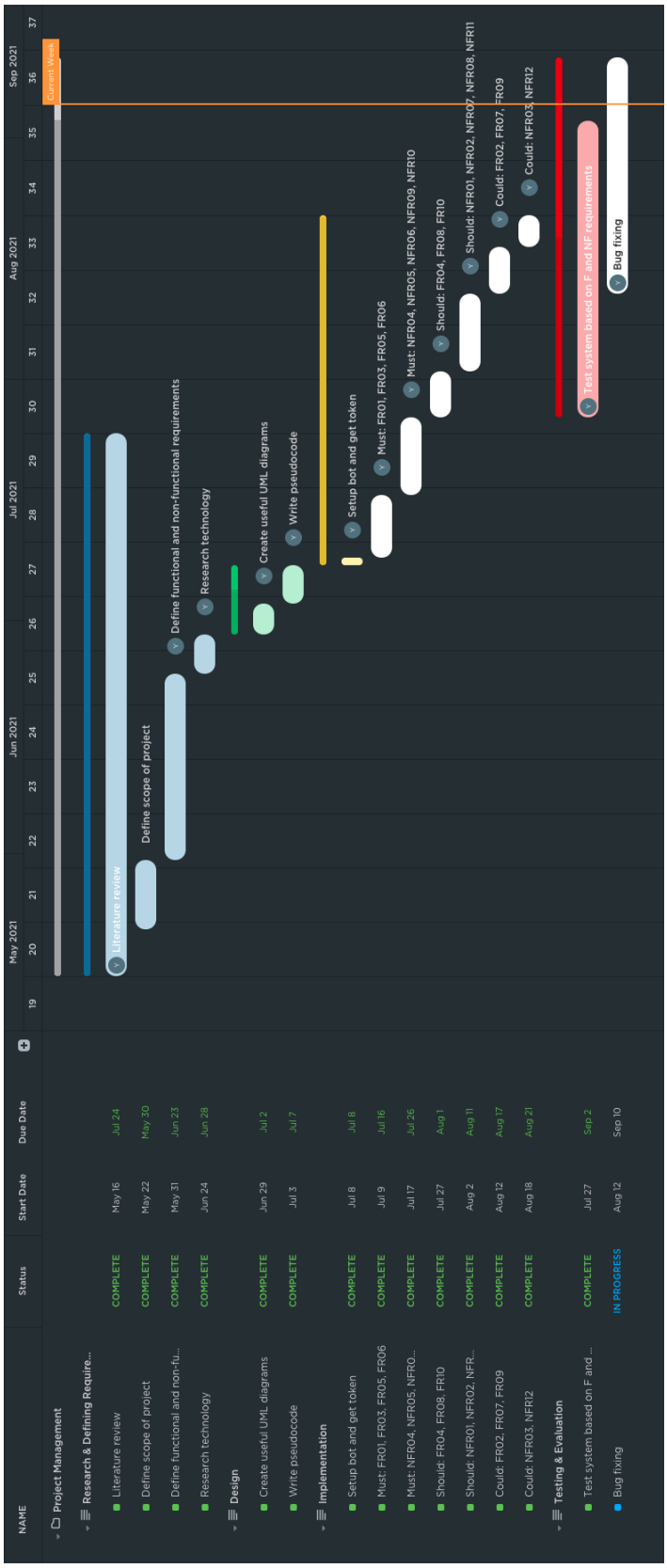
Appendix F: Questionnaire Section 6



Appendix G: Questionnaire Section 7



Appendix H: Gantt Chart



Appendix I: Code snippet for creating packs

```python
    async def create_pack(self, ctx: commands.Context, pack_name: str, *, csv_text: str) -> None:
        """
        create a new flashcard pack
        """

        # send error message and return if pack already exists
        if ctx.author.id in self.bot.packs and pack_name in self.bot.packs[ctx.author.id]:
            msg = f'pack {repr(pack_name)} already exists'
            return await send_error_msg(ctx, msg)

        # check if there is any viable csv text in the message text itself
        csv_lines = await self.parse_text(csv_text)

        # check if there is viable csv in any of the message attachments,
        # and add that to what if any we already have
        attachment_lines = await self.parse_attachments(ctx.message)

        if attachment_lines:
            csv_lines.extend(attachment_lines)

        # if no viable q/a csv can found in message text or attachments, command fails and stops
        if not csv_lines:
            msg = 'no viable question,answer csv data in message text or attachments'
            return await send_error_msg(ctx, msg)

        # create the packreturn
        pack = await self._create_pack(
            csv_lines,
            name=pack_name,
            author=ctx.author,
        )

        # create dict for user if it doesn't exist
        if ctx.author.id not in self.bot.packs:
            self.bot.packs[ctx.author.id] = {}

        # add the pack to the user's dict
        self.bot.packs[ctx.author.id][pack_name] = pack

        # send success message
        msg = f'Created pack {repr(pack_name)}'
        await send_success_msg(ctx, msg)

        # try to delete input message to prevent spoiling
        await delete_context_message(ctx)
```

Appendix J: Code snippet for studying a pack

```
    @commands.dm_only()
    async def study(
            self,
            ctx: commands.Context,
            pack_name: str = None,
    ) -> None:
        """
        spaced repetition study mode (dms) channel only)
        """

        # check for missing pack_name and handle it
        if pack_name is None:
            msg = 'Missing required argument `<pack_name>`'
            return await send_error_msg(ctx, msg)

        # send error message and return if pack doesn't exist
        if (
            ctx.author.id not in self.bot.packs
            or pack_name not in self.bot.packs[ctx.author.id]
        ):
            msg = 'Pack not found'
            return await send_error_msg(ctx, msg)

        # convenience reference to the CardPack
        pack: CardPack = self.bot.packs[ctx.author.id][pack_name]

        # send error message and return if pack has no cards
        if not pack:
            msg = f'Your pack {repr(pack.name)} is empty, try adding some cards'
            return await send_error_msg(ctx, msg)

        # pack mastered, send info and return
        if pack.mastered:

            # disable the reminder if it's running
            reminder_was_running = pack.reminder.is_running()
            if not pack.reminder.is_being_cancelled():
                pack.reminder.cancel()
            if reminder_was_running:
                while pack.reminder.is_running():
                    await asyncio.sleep(0.1)

            # send success message
            msg = (
                f'Your have mastered your pack {repr(pack.name)}, '
                'use the `reset_pack <pack_name>` command to start it again'
            )
```

Appendix K: Code snippet for a quiz

```
    @commands.guild_only()
    async def quiz(
            self,
            ctx: commands.Context,
            pack_name: str = None,
            interval: float = None,
    ) -> None:
        """
        multiplayer quiz mode (server channels only)
        """

        # check for missing pack_name and handle it
        if pack_name is None:
            msg = 'Missing required argument `<pack_name>`'
            return await send_error_msg(ctx, msg)

        if interval is None:
            interval = self.quiz_interval_default
        else:
            interval = min(60.0, interval)
            interval = max(5.0, interval)

        # send error message and return if they already have a quiz session
        if ctx.author.id in self.quiz_sessions_by_user_id:
            msg = "You're already running a Quiz session"
            return await send_error_msg(ctx, msg)

        # send error message and return if pack doesn't exist
        if (
            ctx.author.id not in self.bot.packs
            or pack_name not in self.bot.packs[ctx.author.id]
        ):
            msg = 'pack not found'
            return await send_error_msg(ctx, msg)

        # send error message and return if the pack doesn't exist
        if ctx.author.id not in self.bot.packs or pack_name not in self.bot.packs[ctx.author.id]:
            msg = 'Pack not found'
            return await send_error_msg(ctx, msg)

        # get the pack
        pack = self.bot.packs[ctx.author.id][pack_name]

        # start the quiz
        return await self._quiz(ctx, pack, interval)
```

Appendix L: Card snippet for spaced learning reminder

```python
    async def reminder() -> None:
        """
        sends spaced repetition reminder to practice specified CardPack
        """

        r: tasks.Loop = pack.reminder
        if r.current_loop == 0:
            return

        # if the last round wasn't completed
        if pack.round.active:

            # skips it and sets up the next one
            pack.next_round()

        # otherwise current round was setup when the previous one was completed
        else:

            # activates current round
            pack.round.active = True

        # format proficiency the levels of the round
        proficiency_levels = [str(x) for x in CardPack.Round.ROUNDS[pack.round_index]]
        if len(proficiency_levels) > 2:
            start = ', '.join(itertools.islice(proficiency_levels, len(proficiency_levels) - 1))
            proficiency_levels = f'{start}, and {proficiency_levels[-1]}'
        else:
            proficiency_levels = ' and '.join(proficiency_levels)

        # create the embed
        title = 'Flashcard Reminder'
        embed = discord.Embed(
            color=Colors.embed,
            title=title
        )

        name = pack.name
        value = f"It's time to practice your flashcard pack {repr(pack.name)}"
        embed.add_field(name=name, value=value, inline=False)
```