EXAM COMPANION: A MOBILE APPLICATION TO SUPPORT STUDENTS IN HIGHER EDUCATION DURING REVISION.

A PREPRINT

Atul Ghandhi
Final Year Undergraduate Project 2018/19
School of Electronic Engineering and Computer Science
Queen Mary University of London
ec16358@qmul.ac.uk

Usman Naeem
Project Superviser
School of Electronic Engineering and Computer Science
Queen Mary University of London
u.naeem@qmul.ac.uk

January 18, 2020

ABSTRACT

Students in higher education spend much of their time revising course content to achieve better grades in their examinations. Not all revision techniques are equally effective, the techniques used by students can affect their learning, and in turn, their grades. The project aims to produce a software application which implements features that will aid students in transitioning their revision techniques to more effective ones. This report also investigates the effectiveness of revision techniques used most frequently by students, as well as alternative revision methods used less often using relevant academic papers. The results found that spaced repetition and practice testing were the most effective revision techniques for test preparation. In addition, the Pomodoro technique was found to be an effective method of managing time during individual revision sessions. The Exam Companion application includes features that are built upon research on the Pomodoro technique, spaced repetition and gamification.

Keywords Pomodoro technique · Spaced repetition · Gamification · Practice testing

This report, with any accompanying documentation and/or implementation, is submitted as part requirement for the degree of BSc Computer Science at Queen Mary University of London. It is the product of my own labour except where indicated in the text. The report may be freely copied and distributed provided the source is acknowledged.

Contents

1 Introduction

1.1 Background

University students studying for their Examinations rely heavily on passive studying methods such as highlighting or rereading large amounts of text to memorize required knowledge (Dunlosky, et al., 2013). Such methods, however, are ineffective at helping to understand complicated concepts or even to simply memorize information (Prince, 2004). Passive studying methods cause students to have a lower intrinsic motivation, lower conceptual learning scores and lower engagement in their courses (Carl A. Benware, 1984). Students using these methods could spend their time more productively using more effective studying techniques.

Furthermore, students can manage their time poorly by studying for long hours with few breaks in hopes of learning as much content as possible, known as ‘cramming’. This has been proven in multiple studies to be less effective than carefully managing how time is spent and allowing for breaks (Weng, et al., 2010). Methods of time management such as the Pomodoro technique can be used to complete more overall work in a given time frame by reducing the amount of time spent unfocused and distracted (Ruensuk, 2014).

1.2 Problem Statement

Students revising for their examinations should spend their time in a way that maximises productivity to improve their academic performance and, in turn, their future prospects. Instead students in higher education often rely on ineffective revision techniques which undermine their achievements or do little to improve them.

1.3 Aim

This project aims to aid students in changing their revision techniques to researched methods of studying proven to improve learning per unit time in the form of a software application. In doing so, the project aims to improve the grades achieved by university students in their examinations.

1.4 Objectives

- To understand the studying and time management methods currently employed by university students by conducting interviews.
- To use academic papers from journals and conferences to discover the most effective revision and time management techniques that can help students to improve their revision sessions.
- To produce a software application that implements features that allow students to revise more effectively by improving their studying and time management techniques.

1.5 Research Questions

- How can the revision methods of students be altered so that their productivity is maximised?
- How can time be managed by students when studying so that they achieve the maximum amount of learning/work for a given time frame?
- How can the Pomodoro technique utilised by students to improve time management within individual revision sessions?
- How can Spaced Repetition be utilised to aid students in memorising required course by managing time over their course leading up to exams?
- How can learning methods be modified to utilise Active learning, and how would this change the academic performance of a student.

Section 2 of this report reviews the current literature on the effectiveness of various revision techniques. Section 3 assesses how the research findings can be used to achieve Exam Companions aim. Sections 4, 5 and 6 address the applications requirements, design, implementation and validation (testing). Section 9 concludes this report with a summary of findings and future work.

2 Literature Review

2.1 Conventional revision techniques

2.1.1 Taking notes

Bretzing & Kulhavy (1979) investigated the effectiveness of four levels of note taking. 180 students were assigned one of five learning conditions (including control) and given a fictitious piece of text to study using their assigned technique in 30 minutes. The students were also limited to writing a maximum of three sentences of notes regardless of the technique being employed. A control group simply read the piece of text without taking any notes. The students were then tested on key facts from the text either immediately after studying it or one week later.

The four levels of note taking were as follows:

1. Summarising; condensing a larger amount of text into fewer words by excluding repetitive or unimportant information.
2. Paraphrasing (Note-Taking); rewriting key parts of the text such that it retains its meaning and volume but uses different words.
3. Verbatim copying; Copying the important parts of a text word for word.
4. Letter search; recording in notes all words that begin with capital letters.

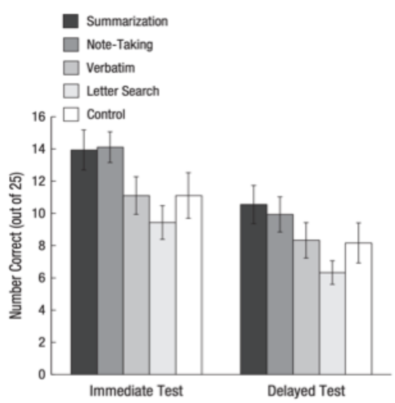

Figure 1: “Mean number of correct responses on a test occurring shortly after study as a function of test type (immediate or delayed) and learning condition in Bretzing & Kulhavy (1979). Error bars represent standard errors.” Dunlosky, et al. (2013)

Figure 1 shows the average number of correct responses out of 25 for each of the methods of note taking tested, both immediately after reviewing the information and with a delayed test.

Bretzing and Kulhavy’s results showed that with a delayed test scenario, most relevant to students in higher education, the students that summarised the text performed best, with those that paraphrased the text coming second. Students that copied the text word for word did as well as the control group that only read the text and students that used ‘letter search’ performed worse than had they not taken any notes at all.

Dunlosky et al. (2013) suggests the "active act of summarising or paraphrasing the information helped students perform better in their test than the passive acts of verbatim copying, reading or writing key words". The authors note “Additional time required for note-taking was only worthwhile when meaningful notes were taken” (Bretzing & Kulhavy, 1979), this ties in with Einstein, et al. (1985) which found the most successful students were those that recorded the greatest number of high importance propositions in their notes.

Similarly, Allison King (1992) found that students that summarised their lecture notes when reviewing them or used self-questioning strategies (Using question templates such as “how does x relate to y” to form and answer questions using their notes) outperformed those that simply reread or highlighted their notes.

These studies show that while taking notes is an effective method of revision, the techniques used when taking and reviewing notes can greatly influence the long-term retention of that information.

2.1.2 Practice questions

Practice testing was found to improve long term retention of information over 100 years ago (Abott, 1909) and in the hundred years since, hundreds of experiments have found it to enhance learning and recall (Dunlosky, et al., 2013).

Some studies, such as Runquist (1983), showed that testing improved retention as opposed to not testing. More recently, Pastötter & Karl-Heinz (2014) found “retrieval practice on previously studied information, compared to restudy of the same material, renders the information more likely to be remembered in the future”. This shows that using practice questions as a form of revision is more beneficial to delayed recall of that information than traditional forms of studying, such as rereading or summarising notes. Likewise, Dunlosky, et al. (2013) carried out an in-depth analysis of 10 commonly used revision techniques, including rereading and summarising notes, and found practice testing to be the most effective revision technique.

Improving recall of previously studied material is known as the backward effect of testing. The forward effect of testing states that “recall testing of previously studied information can enhance learning of subsequently presented new information” (Pastötter & Karl-Heinz, 2014).

Roediger and Karpicke (2006b) gave undergraduates a short text for initial study followed by either a second study session or by a practice test. One week later, recall was considerably better among the group that had taken the practice test than the group that had restudied (56% versus 42%).

The above shows that practice questions are clearly an excellent form of revision for students, as they consistently enhance recall more than almost any other revision technique.

2.2 Comprehensive revision techniques

There are countless ways to tweak revision techniques giving countless methods of revision. This report will therefore only analyse revision techniques that have been sufficiently studied, such that a clear conclusion can be drawn regarding their effectiveness. This will allow them to be better compared so that the software application that results from this research will incorporate features that most improve its users studying quality and thus grades.

2.2.1 Active learning

Learning in higher education today is generally done passively. Passive learning can be defined as being teacher focused, where the teacher presents information and the student sits and absorbs the presented information, without actually doing anything with it. On the other hand, active learning involves applying newly learned information immediately in some way. While there is no exact point where learning becomes active or passive, the more interaction there is with presented information and the more feedback that's provided on that interaction, the more active learning becomes.

Passive learning includes revision methods most typically employed by students, such as rereading notes/textbooks and watching recorded lectures; in these methods students try and absorb information without using it in any active way. There are many methods of active learning that involve doing something stimulating with the information to be learned. These include using the Feynman technique, where students describe a concept as simply as possible and relearn any part of the concept they cannot simplify. Creating flashcards or charts to represent the information in a different form or summarising notes as opposed to just rereading them are also active revision techniques.

Carl A. Benware (1984) investigated whether students that learned course content with an active orientation would perform better in class than those that learned with a passive orientation. The work notes that several studies (Cloward, 1967; Allen & Feldman, 1972) found that when students volunteered to tutor other students, the improvement in learning in the tutors was as great or greater than that of the students being tutored. It is further noted that other studies such as The National Commission on Resources for Youth (1972) and Goldschmid (1970) found the gains for teachers were more than just academic; tutoring also enhanced the perceived competence, motivation and self-esteem of the tutors. Benware hypothesised that these changes may be partly psychological due to the process of learning itself being different when learning to teach, in which case the improvements to the tutor should be present before any tutoring actually takes place. The work tests this hypothesis by having some students learn content (an article on brain functioning) with the expectation of teaching it to another student and by having other students learn the same content with the expectation of being tested on it, thus creating an active orientation (learning to teach) and a passive orientation (learning to be examined). Note that the work does not require students to use any specific revision technique, instead, the independent variable is the mindset of students when learning the content.

The results of the study found that those students that learned to teach were more "intrinsically motivated and perceived themselves to be more actively engaged with their environment" than those students that learned to be examined. The students that learned to teach also perceived the 'activeness' of learning to teach as much higher than learning to be examined, whereas those that learned only to be examined gave similar activeness/passiveness scores to both types of learning. Both groups of students were given an examination on the content and, as table 1 demonstrates, while the groups performed similar on rote learning questions, those that learned in order to teach performed significantly better on questions requiring conceptual understanding.

The above work covers the difference having an active orientation can make when revising examinable content, but it doesn't show how any specific revision technique can be used by students to improve conceptual understanding or recall.

Dunlosky, et al. (2013) compared the effectiveness of several popular revision methods and found spaced practice (aka spaced repetition; analysed below) and practice testing to be the most effective studying techniques. The use of flash cards, an active learning method, can utilise both of these techniques. Creating flash cards requires content to be

Table 1: Means and standard deviations of rote and conceptual learning scores (Carl A. Benware, 1984)

	Experimental (n=19)	Control (n=21)
Rote learning score	18.21	16.24
	(4.58)	(4.13)
Conceptual learning score	18.84	10.76
	(4.89)	(4.23)

summarised to fit onto small cards which can then be reviewed multiple times conveniently to memorise the information on them. They can utilise practice testing by being double sided, with a prompt on one side and a piece of information that needs to be learned on the other. The student can then view the prompt on one side of the card and test how well they know the information on the other side of the card before seeing it.

Forbes, et al. (2013) investigated whether time would be more productively spent learning unknown flash cards alongside already learned cards or if students would be better off revising only unlearned cards. The study measured the learning rate (number of new cards learned per minute) of three students and found that interspersing additional known cards in flash card review sessions reduced learning rates compared to using only unlearned cards. Kupzyk, et al. (2011) found similar results, though it should be noted that both of these works involved very few participants (3 and 4 respectively) of a young age and the participants in Forbes, et al. (2013) all had learning disabilities. Therefore, the results of these studies may not be generalisable to students in higher education.

2.2.2 Pomodoro technique

The Pomodoro technique is a time management method, originally created by Francesco Cirillo, that aims to help overcome distractions when working to improve productivity. Using the Pomodoro technique, a person will work for a slot of 25 minutes on a given task during which they will only work on that task and ignore any other pending tasks or distractions. After the 25-minute slot, called a Pomodoro, the person will take a 5-minute break, followed by another 25-minute Pomodoro session. After every 4 consecutive Pomodoro's, a larger 15 to 30-minute break is given to 'refresh' the mind.

The Pomodoro technique has several advantages. It allows the amount of time spent on a task to be easily quantified (by counting the number of Pomodoro's completed for a task) and it allows for larger tasks to be split into smaller chunks. The primary objective of the time management method, however, is to improve productivity.

Ruensuk (2014) investigated the effectiveness of utilising the Pomodoro technique in an agile software development environment. This is relevant to students in higher education as both environments require focused work to meet strict deadlines.

This work measured the productivity of developers as time spent to complete a task, this was measured both before and after introducing the Pomodoro technique. The study also measured the number of internal and external interruptions that affected the developers, an internal interruption being something the developer was responsible for (i.e. checks Facebook) and an external interruption being something out of their control (i.e. boss or team member asks for help). Any Pomodoro that was interrupted was considered void and would begin again when the developer returned to work. The results collected over a two-month period (split into 'sprints' of two weeks each) found that the number of hours required to complete a similar task dropped by a third when using the Pomodoro technique over working with no time management.

Furthermore, as figure 2 illustrates, internal interruptions dropped significantly while external interruptions also decreased in number. Internal interruptions may have dropped due to developers being more conscious of those interruptions and making an effort to stop them to prevent a Pomodoro slot being declared void. It is also likely that after

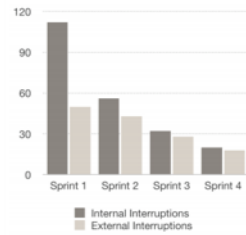

Figure 2: Mean number of internal and external interruptions during work. (Ruensuk, 2014)

using the Pomodoro technique for a while, developers were able to adapt the time frame of the Pomodoro's and breaks to suit them; thus, reducing the number of distractions during their Pomodoro sessions. External interruptions likely dropped as a result of developers postponing dealing with interruptions until their Pomodoro slot was over whenever possible.

The drawbacks of the technique, as mentioned by the developers after the two-month period, were that they had to wait for team members to finish their Pomodoro slots if they needed help and that external interruptions could not always be managed and would result in a void Pomodoro.

The study gives an insight into a practical implementation of the Pomodoro technique, its effectiveness and drawbacks. However, it was only conducted within a single company (Think Blue Data Co., Ltd.) and with a small team of developers. As such the results may not be representative of all people that use the technique.

Weng, et al. (2010) also investigated the impact of the Pomodoro technique, or lack of it, with an agile distributed software development team based in Italy. The study found that using the Pomodoro technique enhanced the team's productivity and helped with tracking the progress of a task and estimating the time required to complete it. The authors state the Pomodoro technique works because it is based on time-boxing; the idea of assigning a task an unchangeable time interval. When necessary, an unfinished task can be relegated to the next time interval. The time-box method works when it is 'protected', i.e. once begun its length does not change and distractions are deflected. As such, any definitive interruptions to a Pomodoro, internal or external, make it void. This improves focus and concentration on work by reducing distractions, time-wasting and anxiety linked to the passage of time. Moreover, Weng notes the Pomodoro technique allows "a sustainable working pace to be obtained through the alternation of work and rest and the combination of short breaks and longer pauses".

The only issues found with the technique were that communication between team members was reduced as colleagues could not distract each other during a Pomodoro slot. This was solved with collective Pomodoro's where the entire team used a single timer and took collective breaks. Another issue found was that some team members would use their break to discuss the task just completed in the Pomodoro, making the break as taxing as the Pomodoro slot and nullifying the core of Pomodoro time management. This was somewhat solved by incorporating team activities during the break sessions such as '5-minute fitness club'. Both of these issues would not be relevant to the average student independently studying for exams, the benefits of increased productivity with time management, however, would be relevant to them.

The above studies show that the Pomodoro technique is an effective and easy to implement method of improving productivity during intensive work. And yet, while there are multiple studies of the techniques implementation in agile software development, it has not been utilised in an educational context. While some online websites may recommend the technique to students, its use is far from widespread and many students that could benefit from the Pomodoro technique are unaware of it.

2.2.3 Spaced repetition

Our brains memorise information more firmly with more repetitions of that information and each time a piece of information is repeated it takes longer for our brains to forget it. Repeated reviews of information can take place in a 'massed' heap (where information is reviewed multiple times consecutively) or reviews can be spaced out over time. 254 studies involving 14,811 participants found spaced practice was far more effective for recall than massed practice for all retention intervals (Cepeda, et al., 2006).

Table 2: Retention of information when tested; for massed and spaced practice among participants of various studies

	% Correct			
Retention interval	Massed	Spaced	No. of studies	No. of participants
1-59 s	41.2	50.1	96	5,086
1min - <10min	33.8	44.8	117	6762
10min - <1 day	40.6	47.9	10	870
1 day	32.9	43.0	15	1,123
2-7 days	31.1	45.4	9	435
8-30 days	32.8	62.2	6	492
31 days +	17.0	39.0	1	43
All retention intervals	36.7	47.3	254	14,811

There is little consensus to account for the retrieval advantages of spaced practice though several theories exist. A popular one, backed by empirical evidence, is the study-phase retrieval theory (Braun & Rubin, 1998; Thios & D'Agostino, 1976) which states that restudy of information prompts a retrieval of the memory trace of the first presentation. This causes an elaboration on the first memory trace enhancing long term recall. Massed practice does not yield the same results since the first memory trace is active during the restudy of information, so the information is not retrieved.

The above studies show that spaced repetition is an effective method of improving recall and students trying to memorise information should therefore space out their review of each concept over time instead of massing the practice of each concept in one go. Students are often constrained for time and as such, spaced repetition will be most beneficial to them when restudy of information has to occur as infrequently as possible while the information recall is still relatively high.

Cepeda et al. (2008) tested 10 different spacing intervals (called lag or gap) between initial practice and review of trivia facts; ranging from 0 to 150 days.

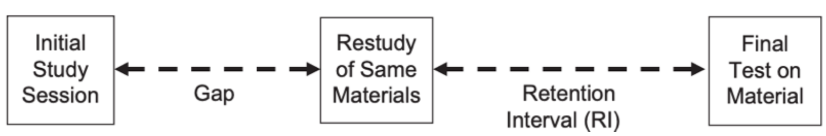

Figure 3: Structure of most studies investigating the impact of the spacing effect on learning.

A final test was administered between 7 and 350 days after the final review session; this period is known as the retention interval. The experiment found that retention of the facts was highest when the lag was between 10 to 20% of the retention interval. Since students are limited in time, taking the upper bound of 20% means that according to this study, a student revising for something on the first day of term knowing they will be tested on that content in an exam in 9 months (270 days) should review that information every 54 days (20% of 270) for it to remain memorised.

Unfortunately, the above strategy assumes all the content a student needs to learn will be first reviewed on their first day, which is not the case. For spaced repetition to have maximum efficiency in an educational context, review scheduling must balance the competing priorities of introducing new information and reviewing old information such that the rate of learning is maximised.

Reddy et al. (2016) discusses the Leitner System as a solution to solve review scheduling. The system relies on two sided flashcards, with a prompt on one side (i.e. key word) and information that needs to be learned on the other (i.e. definition of the key word). The cards are arranged in a series of decks where new information is added to 'deck 1' and the cards in each deck are reviewed in a 'First-In-First-Out' queue with the first deck having the highest priority and each following deck having a lower priority. Each time a card is reviewed, if the prompt is answered correctly (i.e. the person correctly recalls the definition of the key word) the card is moved a single step down to a lower priority deck and if the prompt is answered incorrectly then the card is moved one step up to a higher priority deck. Cards incorrectly recalled in deck 1 stay in deck 1 and cards correctly recalled in the final deck are removed from the system. Essentially, this forms an ordered list of flash cards to be learned such that cards will be repeatedly reviewed until memorised.

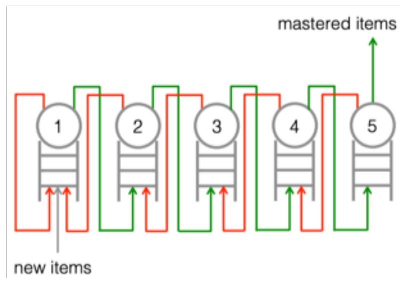

Figure 4: Structure of most studies investigating the impact of the spacing effect on learning.

Figure 4 gives an illustration of how the Leitner system can be implemented. New information is added to deck 1 to be memorised. Red lines show transition of cards to higher priority decks and green lines show transition of cards to lower priority decks. Cards in deck 1 are reviewed each learning session and cards after deck 5 are 'retired' so no longer need to be reviewed. Cards in decks 2, 3, 4 and 5 are reviewed every 2nd, 3rd, 4th and 5th learning session respectively. The interval between the learning sessions would be up to each individual user; though they would typically be several days apart to utilise the benefits of spaced repetition.

The research studies analysed agree that spacing out information reviewal over time leads to superior recall in a test environment than 'massed revision'.

2.2.4 Gamification

Gamification is the concept of applying typical gaming elements such as gaming mechanics, aesthetics and game thinking, to non-gaming activities such as education (Borges, et al., 2014). Digital games are capable of motivating the user to keep playing, holding their interest and convincing them to solve internal conflicts and problems within the game. In theory, gamifying revision would increase student motivation to study and help reduce boredom induced procrastination. The majority of studies on gamification as applied to education concern higher education as the benefits of gamification have sparked a recent growing interest in developing eLearning environments equipped with gamified elements (Borges, et al., 2014). O'Donovan, et al. (2013) claims higher education is already much like a game where "students start as novices (freshmen). As they progress, they go on quests (lectures) to learn skills (coursework) and are then tested on these through challenges (tests and assignments) that determine whether they qualify for the final boss-battle (exams) in order to level up (pass the year) or lose the game (failure)."

Butler & Ahmed (2016) gamified an introductory Computer Science course which new students often struggled with. The game used a 'space-ship' like interface to teach students concepts such as stacks and selection sort. The work assessed the effectiveness of gamification using surveys and discussions with the students. A majority of students claimed they learned better with an active learning style and agreed that learning difficult concepts through a gamified environment helped with understanding them. A large number of students said they lost the motivation to learn concepts that were too difficult to understand. The study found that using games to learn increased motivation, retention and conceptual understanding in students. While this work showcases an example of how gamification can be utilised in an educational environment and presents student opinions of gamifying learning, little quantitative analysis is provided.

Nah, et al. (2014) carried out review of papers relating to gamification in educational and learning contexts. The reviews included Brewer, et al. (2013), which found gamification in the form of points and prizes for tasks completed increased the motivation of students (aged 5 - 7) and task completion rose from 73% to 97%. Gibson, et al. (2015) explained that using digital badges alongside points and leader boards would "motivate some learners to continuously engage with online materials and activities". Goehle (2013) integrated levels and achievements in an online calculus course. The work found higher engagement levels from students on the course, with half the students completing more homework than required to gain additional achievements. In an optional survey the majority of students said they felt the system was rewarding and appreciated the extra acknowledgement for doing their homework.

O'Donovan, et al. (2013) gamified a games development course. Students were given experience points for lecture attendance, performance on quizzes, participation in class exercises and other activities. Gaining experience points also led to gaining in-game currency which could be used to buy things like assignment extensions, giving the gamification real life impact. Several other gamification elements were also implemented including progress bars, leader boards and badges. In a questionnaire, students felt gamification improved their understanding of and engagement with the course. The results found leader boards to be the most effective at motivating students to engage with the course. They also found a significant increase in lecture attendance, much higher than other computer science courses in the same department and most importantly the results found a statistically significant improvement in student grades from the previous year. It should be noted that gamifying a games development course may have skewed the results as the participants would likely be avid gamers already and more likely to engage in a gamified course than the average student.

While these papers show gamification to be a promising method of improving the learning experience of students, not all studies have found it to be effective. Barata, et al. (2015) gamified a computer science course in Information Systems with multiple game design elements such as points, levels, leader boards, challenges, and badges. The results indicate gamification of the course led to higher student engagement and participation as well as an increased attendance in lectures implying increased interest or enjoyment in the course. However, this did not lead to a significant improvement in student grades.

Berkling & Thomas, (2013) even found gamification to be detrimental to learning. A computer science course in Software Engineering was gamified by including achievements, points, progress bars and leader boards. Questionnaires administered to the students revealed that a majority of them did not consider themselves to be gamers and many of them found gamification of the online platform a hindrance to their learning rather than an asset. The overall feedback received from students showed no consensus with positive and negative views being balanced. Features such as points and public recognition for helping others through a forum were functional but not utilised by students. The work suggests that this could be partly due to students finding the transition between teaching styles required additional unnecessary effort and partly due to a strong culture of formal, traditional schooling. The work found that explicitly naming the platform "gamified" discouraged some students that were opposed to the idea of learning through games and suggested that a shift in the learning environment towards active learning would need to be implemented slowly to be accepted.

The above studies demonstrate that while gamification can help motivate and engage students in their work, overhauling a system to make it completely gamified can adversely affect student engagement with the course. Gamification appears to be most effective when it is not explicitly named as such and is subtly embedded into a system rather than being the focus point.

2.3 Student reflection

Interviews were conducted with 24 students, from multiple universities, to investigate current approaches to exam revision and time management.

Students were asked about their approaches to revision when distant from examinations and during exam season. When distant from examinations, 91.6% of the students mentioned in some form "reading or writing notes using lecture slides and/or recordings"; with ~25% of those students only reading the notes provided. In follow up question's students said they copied or reworded lecture slides with the intention of rereading them when examinations approached. Very few students used their notes in an active manner; such as summarising their notes or using them to create charts or revision cards.

Revision techniques used by students changed when exams approached, with approximately 71% of students switching partly or entirely to completing practice exam questions, with 29% spending the run up to exams taking and/or reading more notes on course content. Students also claimed they spent significantly more time revising as exams approached, taking fewer breaks and leisure time off revision.

Since practice exam questions are one of the most effective revision techniques (Dunlosky, et al., 2013), this means the majority of students spend most of the year revising with less effective revision methods and switch to more effective ones leading up to exams.

The students were then asked how they timed individual revision sessions and how they scheduled revision across their timetable. The majority of students (66.7%) said they did not time revision sessions except when completing past papers in a set amount of time. For other forms of study such as taking or reading notes revision was generally done until students got distracted, tired or had a timetabled activity to attend. The remaining third of students mentioned timing revision sessions by assigning a block of time to each module or subject for the day to make sure all subjects were covered.

Regarding scheduling revision sessions, a quarter of the students said they did not schedule revision sessions beforehand, instead working on whichever task seemed most pressing at the time. Of the students that used a timetable to schedule their revision sessions, the scheduling used was largely centred around ensuring each subject had an equal amount of time dedicated to it and largely came into effect during exam season when lectures stopped. During semester most students worked on whichever module they felt most needed their attention, either due to a lack of understanding or a deadline approaching for an assessed task.

The interview responses show time management is not to a concern for many students as they are unaware of its immense potential to improve their productivity when studying. Utilising the Pomodoro technique and spaced repetition would be hugely beneficial to students that do not time their revision at all.

Once exam season approached, over 70% of students used a digital timetable on their smart phones with a calendar or timetable application. A small number (~4%) used a written timetable to schedule their events. Of the students using a timetable, 80% dedicated each day to a different module or examination and the remainder split each day into several slots for different modules. Here again, the majority of students said they revised subject knowledge in blocks (one lesson a day) instead of splitting up practice over several days which would utilise spaced repetition and greatly improve their recall of the revised content. Students do not have to spend more time studying to improve recall, they simply need to schedule the time they already spend differently.

2.4 Existing approaches

2.4.1 Flash cards Application

There are many digital flash card applications online though the most popular of them is ANKI, available on several platforms (Android, Windows, IOS, OSX). In Anki, the flash cards are stored in 'decks' and users can review the cards of any deck. The order of cards shown is decided by a spaced repetition algorithm based on the SM-2 algorithm where the interval between each review increases if the card is correctly recalled and is reduced if it is not recalled. Upon each review of a card, the user can select 'Again' (bad), 'Good' or 'Easy'. The easier the recall of the card, the longer the interval before it is next presented for review.

Anki: List of decks	Anki Example card	Anki: Example card	Anki: Statistics

Figure 5: Screenshots of the Anki application running on Android.

While Anki includes a large set of features relating to flash cards (share decks with other users, sync decks/cards across platforms, view statistics of applications use), the application does not include any form of gamification to motivate its users to create cards and does not include any time management features, such as Pomodoro, to create and review flash cards. Most importantly, there is no way for users to browse the Anki cards they have created. They only way to access cards is by reviewing the cards the algorithm decides a user needs to review.

2.4.2 Pomodoro timer application

The most comprehensive Pomodoro timer available on Android, IOS and OSX is called "Focus To-Do: Pomodoro timer and To-Do List". The application includes a Pomodoro timer with adjustable time intervals and allows the user to add 'projects' to a list, projects can be marked completed and new projects can be added. Additionally, a Pomodoro session can be linked to a project so that the user can measure how much time was spent on the project, although this is not necessary. The application provides statistics on its use and includes some gamified elements, for example using the application 'generates sunlight' which is then used to grow a virtual forest, which can be viewed from within the applications user interface. A larger forest also leads to a higher level.

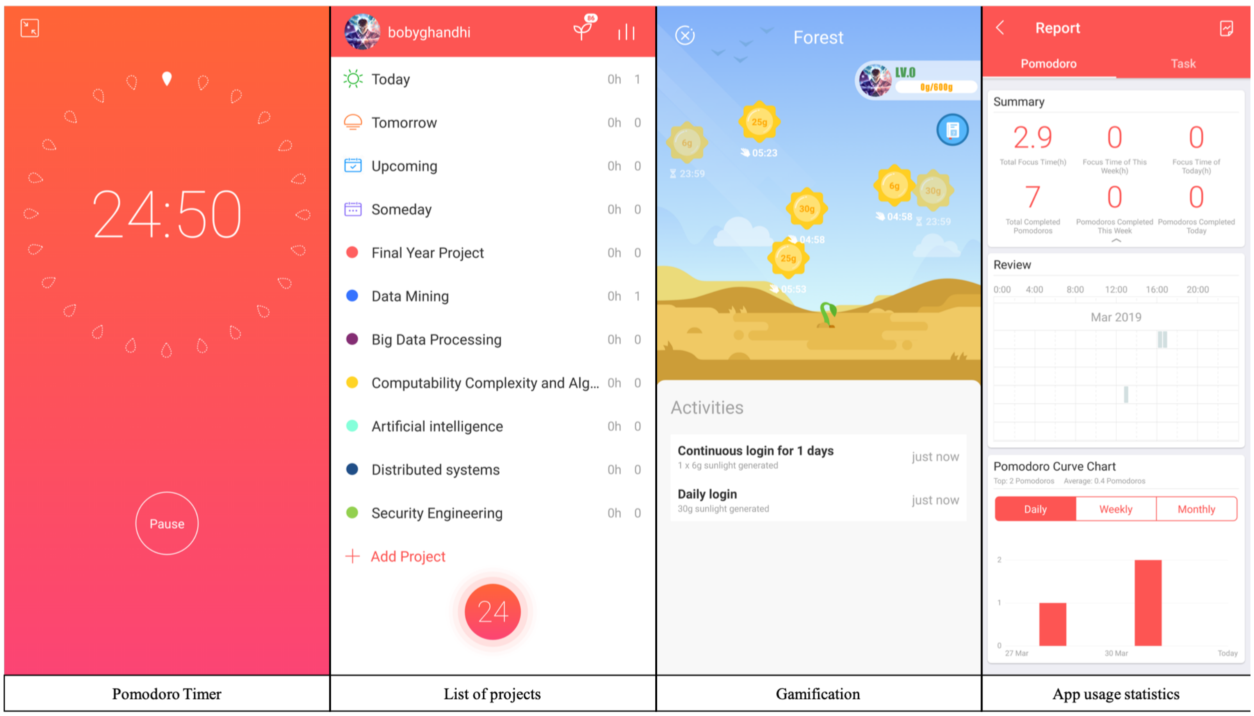

Figure 6: Screenshots of the 'Focus To-Do' application running on Android.

While the forest section of the application provides an original take on gamification, the application does not focus on this and the user may not know about 'sunlight' unless they clicked on the unlabelled forest button. The statistics also do not include sunlight as a measure of work done and a user is not notified of sunlight generated when using the application, making the gamification redundant. Gamification works by rewarding the user for completing work, but if a user is not notified of the reward received or their progress, than it will not make a difference. This was likely a choice made by the applications developers because it is marketed towards adults and obviously gamified elements may discourage an older target audience from using the application.

The application not being directly aimed at students also means that subjects or modules cannot be added so that a student can view how much work they're completed for a given module. Projects can be added instead, which can be labelled complete when done. Furthermore, tasks can be added in the To-Do section of the application but a user cannot enter a deadline for the task as the application only allows "Today, Tomorrow, Someday". Students with assignments that have strict deadlines would find it far more useful to be able to add deadlines to tasks and view their pending work in order of urgency.

2.5 Application of research

Exam Companions aim is to enable students to be more productive with their revision techniques. To achieve this, the Exam Companion application must be easy to use and understand. If students found the application too complicated due to an abundance of features then they would be less likely to use the application and benefit from enhanced revision techniques.

The Pomodoro technique is shown to consistently improve productivity during focused work and provides an easy method of breaking up larger tasks and reviewing the amount of time spent on a given subject. The Pomodoro technique, while relevant to students, is not widely used in an educational context being mostly implemented in agile software development companies. The application will therefore include a Pomodoro timer aimed specifically at students to make their revision sessions more productive. The timer will be designed for students by allowing them to set the module a Pomodoro is being completed for and view a history of Pomodoro's completed, as well as what work was completed during each Pomodoro.

Dunlosky et al. (2013) found spaced repetition and practice questions to be the most effective methods of revision. All other studies looked at concurred with the statement that both spaced repetition and practice questions are effective methods of learning information. Flash cards are a method of revision that can combine both both spaced repetition and practice testing. Flash cards can be used to test the recall of information and reviewing them with suitable intervals also takes advantage of the benefits of spaced repetition. The application will include a flash cards page that will allow students to create digital flash cards and store them in decks. Students will then be able to review flash cards, one at a time, testing their knowledge of each card.

The application will include a schedule page where students can schedule all of their academic activities including time tabled lectures and revision slots. The schedule page will be linked to the Pomodoro page, where any revision slot will send a notification to the user to use the Pomodoro timer for their revision and likewise, using the Pomodoro timer will require a revision slot to be scheduled at that time.

Lastly, the application will include gamified elements to encourage students to use the application and motivate them to revise more. The application will reward the user for certain activities within the application with digital points. Activities such as creating digital flash cards and using the Pomodoro timer to revise will earn the user points, which will affect their ranking in the games leader board. This will encourage users to use the Pomodoro timer to revise more often and to create more digital flash cards.

3 Requirements

Table 3: Exam Companion application requirements analysis

Id	Requirement	Type	Priority	Use case
1	The system shall allow the user to set a Pomodoro timer that alternates between 25, 5 and 15 minutes.	Functional	Core	Pomodoro timer
2	The system shall allow the user to change the Pomodoro timer intervals.	Functional	Core	Pomodoro timer
3	The system shall allow the user to add their modules from the Pomodoro timer page, and thereafter tie each Pomodoro completed to the module it was completed for.	Functional	Core	Pomodoro timer
4	The system shall allow the user to record a target before a Pomodoro begins and a summary of work completed at the end of a Pomodoro.	Functional	Core	Pomodoro timer
5	The system shall store on the device, and allow the user to view, a history of Pomodoro's completed with their targets and summaries where applicable.	Functional(D)	Core	Pomodoro timer
6	The system shall grant the user one point for every 5-minutes of a Pomodoro session completed, with four additional points for completing four consecutive Pomodoro's and two additional points for adding a target and summary for the Pomodoro session.	Non-functional usability	Optional	Gamification
7	The system shall allow the user to compete with other users for points by sharing their score on a global leader-board.	Non-functional usability	Core	Gamification
8	The system shall allow the user to create digital flash cards on their device.	Functional	Core	Flash cards
9	The system shall grant the user one point for each flash card created.	Functional	Core	Gamification
10	The system shall store created flash cards on the device and in the cloud.	Functional(D)	Core	Flash cards
11	The system shall allow the user to organise flash cards into 'decks' .	Functional	Core	Flash cards
12	The system shall allow the user to delete created flash cards.	Functional	Core	Flash cards
13	The system shall allow grant the user two points for each flash card created.	Non-functional usability	Optional	Flash cards
14	The system shall allow the user to add events in a schedule.	Functional	Core	Schedule
16	The system shall display the schedule in a chronological list. This list of events shall be stored on the device and in the cloud	Functional(D)	Core	Schedule
17	The system shall allow the user to set events to repeat after each given time interval.	Functional	Core	Schedule

Table 3: Exam Companion application requirements analysis

Id	Requirement	Type	Priority	Use case
18	The system shall allow the user to create an unlimited number of events in the schedule.	Non-functional usability	Core	Schedule
19	The system shall allow the user to understand its features and navigate their way through the application.	Functional	Core	-
20	The system shall be written in java and run on the android mobile operating system.	Non-functional compliance to standards	Core	-
21	The system shall allow each user to create an account using their email or existing Google log-in. This account will be used to store all user data, including events and Pomodoro history, in the cloud.	Functional	Core	-
22	The system shall be secure and not allow any user to access the data of other users stored in the cloud	Functional	Core	-

4 Design

4.1 Wireframes

Since Exam Companion's purpose is to boost the productivity of its users, the user interface was designed to be simple, easy to use and above all, not distracting. The applications log in page implements the default Firebase authentication user interface to allow a user to create an account with their email address or sign in with their existing Google account. As the application is built upon android, it is reasonable to assume that any user will have a Google account, making the first time log-in of most users a short one click process.

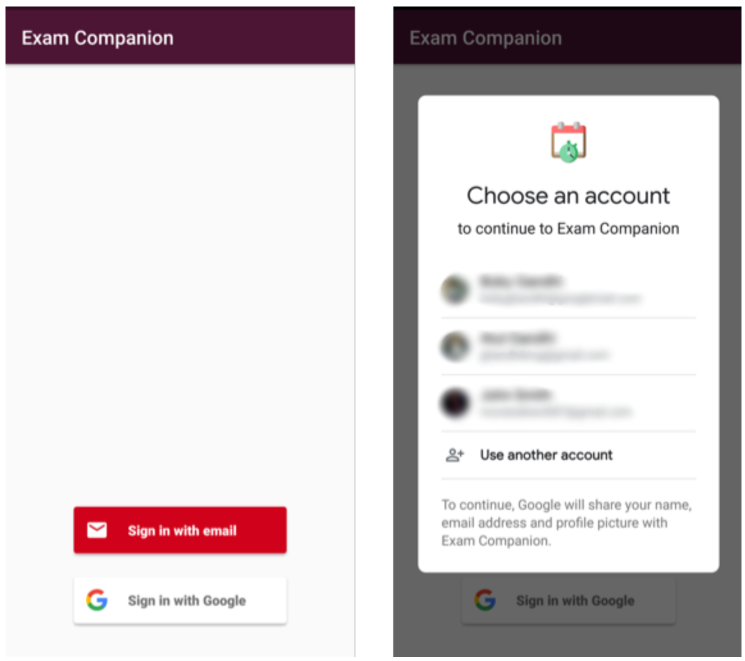

Figure 7: Initial log-in activities when user first opens application, alongside log-in using Google account; application prompts user to select Google account to log-in with.

4.1.1 Home activity

As the application has 4 distinct features, the home page a user is first greeted with after logging in or reopening the application has 4 plain buttons.

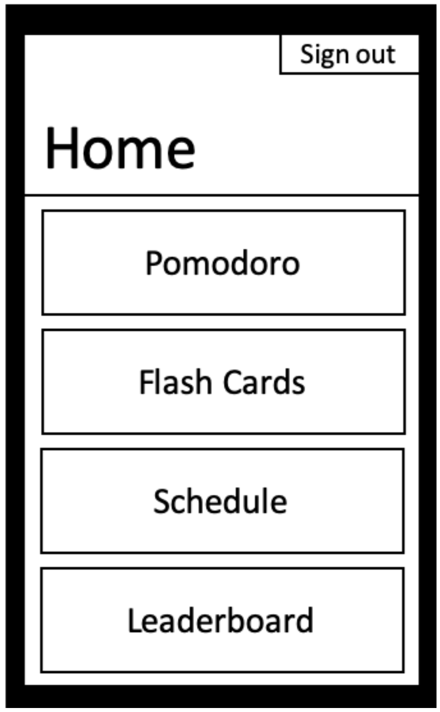

Figure 8: Wireframe of Exam Companion Home activity.

4.1.2 Pomodoro Activity

The most productivity-boosting feature of Exam Companion is the Pomodoro timer, so it is the first button on the list. Next is flash cards, which would allow the user to create or review flash cards as part of their revision, while a Pomodoro timer was running. Next is a schedule that would allow a user to add or view their upcoming academic events and the final button is the dashboard. This will take the user to a page that displays their points, shows leader-board tables of users with the most points, and displays statistics of their use of the application.

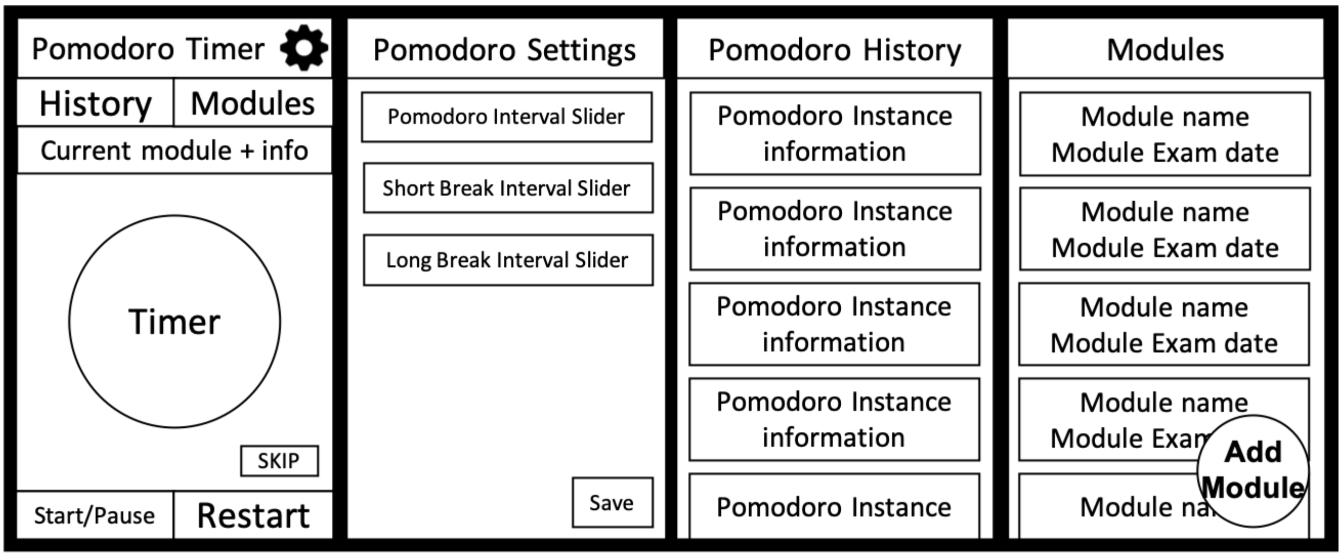

Figure 9: Wireframes for the four Pomodoro activities.

For the Pomodoro timer to fulfil all of its requirements it needed four screens. The main screen that opens when the Pomodoro button from the Home screen is clicked is for the timer itself. This screen needed buttons to use the timer, including a Start/Pause button, a restart button (to restart current timer) and a skip button (to skip the current Pomodoro iteration to the next break or work interval). In addition the Pomodoro screen also needed buttons the other three Pomodoro related page including timer Settings, Pomodoro History and Modules. The remaining space on the main Pomodoro screen is taken up by a circular progress bar and countdown text along with a button to choose the module the Pomodoro is for with additional information about the current or next Pomodoro iteration.

The Pomodoro settings screen needed a method for the user to easily adjust the time intervals for Pomodoro sessions, and short and long breaks in-between the Pomodoro's. The easiest method to do this was sliders, as these would also allow a maximum limit to be easily set on timer intervals. A Pomodoro session that lasts too long would nullify the benefits of using the Pomodoro technique so using a slider with a maximum value to set the interval length helps prevent that.

The Pomodoro history and modules screens include only scrolling lists of the users Pomodoro session history and their modules. The module screen also allows users to add new modules, as a Pomodoro session cannot begin unless the user has at least one module stored to complete the Pomodoro for.

4.1.3 Schedule activity

The schedule screen only needed to display users events and allow users to add new events. A scrolling list view of upcoming events takes up most of the page with a floating button that takes the user to the add event screen. The add event screen needed to include all the variables an event may have, including location, type, repeat, notes and colour (for organisation). While not all fields are relevant or mandatory for each event, they all have to be displayed when creating an event.

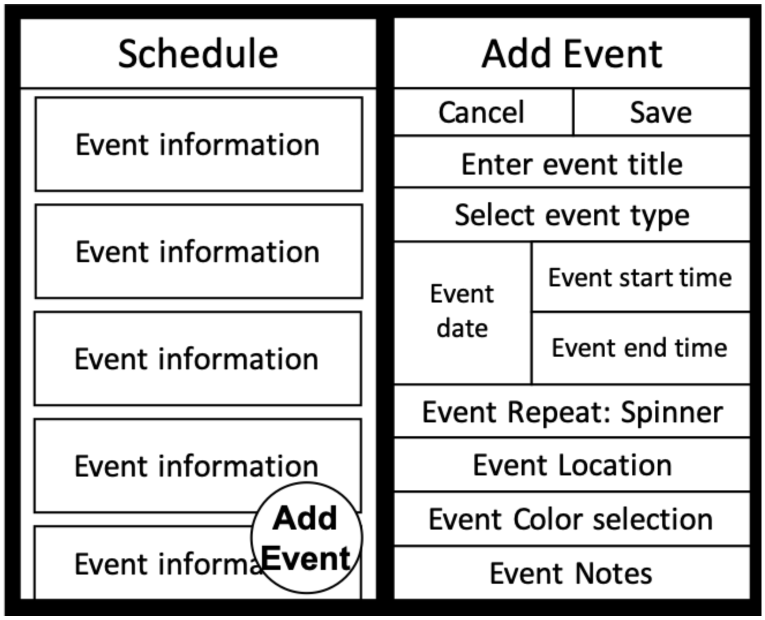

Figure 10: Wireframes for the two Schedule activities.

4.1.4 Flash-Cards activity

The flash cards activities are also tailored towards University students. When a user first clicks the flash cards button from the home screen they are lead to a list of modules, these are the same modules from the Pomodoro screen, although the activity is different as it displays only module names and new modules can be added directly via an alert dialog (opened when floating button is clicked). Clicking a module takes the user to a similar list of decks for that module, so that the flash cards for each module can be organised into decks (where each deck is a module topic). Selecting a deck takes the user to a RecyclerView scrolling list of cards, with each card showing the card prompt (front of card). A floating button allows the addition of a new card. Clicking a card will take the user to a another activity where only the card prompt is visible and clicking the prompt then reveals the card content (back of card).

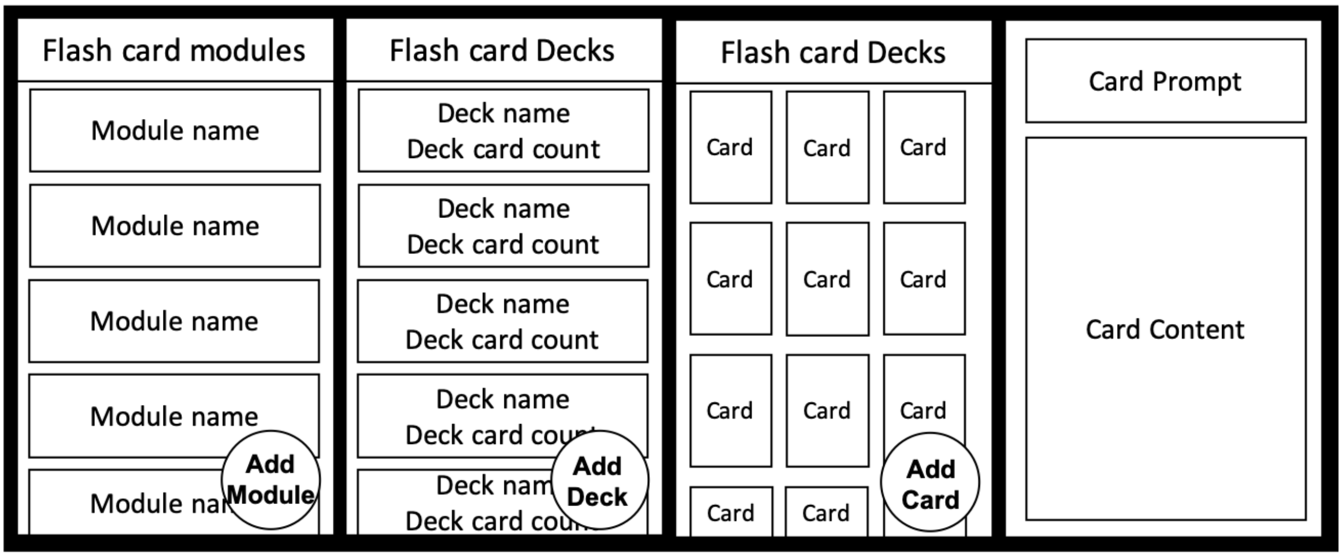

Figure 11: Wireframes for the flashcard activities.

4.1.5 Leaderboard activity

The leaderboard activity is a scrollable listview of application users, with each row displaying the users image, name and points. The list is sorted by points, with users that have the most points at the top.

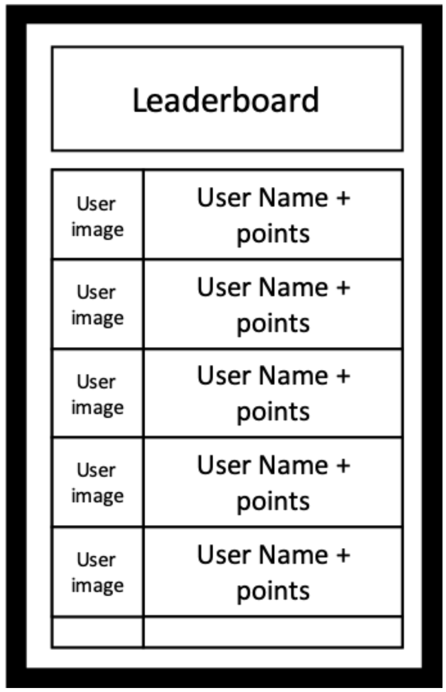

Figure 12: Wireframes for the Leaderboard activity.

5 Implementation

5.1 Pomodoro

The Pomodoro activities main feature is the Pomodoro timer. It utilises the Pomodoro technique to iterate between timer a 25-minute work session (a Pomodoro), followed by a 5-minute break. This helps improve the productivity of its users as the short breaks reduce distractions and boredom induced procrastination. After every four Pomodoro's, a user can take a longer 15-minute break. These are the default intervals, recommended by the created of the Pomodoro technique, however these intervals can be changed to suit each individual user.

As the timer is aimed specifically at students each Pomodoro session can be assigned to a specific module. Modules can be added by the user by clicking the modules button which takes the user to the 'Modules' activity. This activity includes a ListView of modules, populated by an adapter, displaying each module's name and exam date. Clicking the floating button on the bottom right of the page opens a dialog that allows the user to add a module to the list. Adding a module creates a ModuleObject which is uploaded to the Firebase Realtime Database and stored in JSON file format. This database is used to populate the modules ListView.

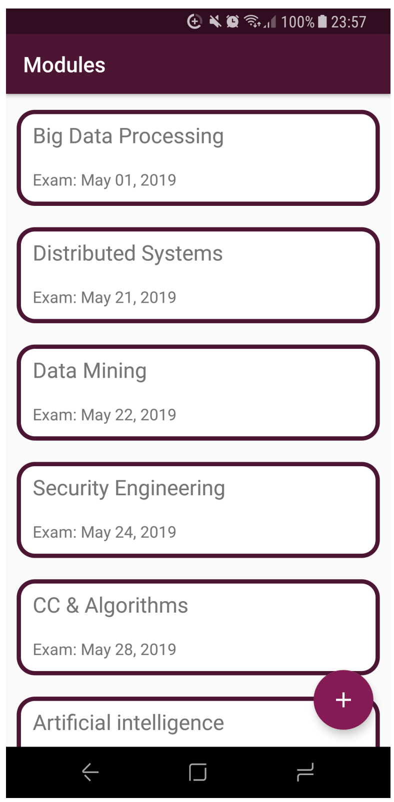

Figure 13: A screenshot of the Modules activity, accessed through the Pomodoro page.

On the Pomodoro page a spinner, populated by a spinner adapter accessing data from the firebase database, shows a module from the modules list above the countdown text. Clicking this module shows a list of all modules, which the user can select. That way the user can select a specific module for each Pomodoro they complete. When a Pomodoro timer begins a Toast (short popup message) asks the user to make sure they've selected the current module from the list of modules. If the user has not added any modules, then the timer will not begin until at least one has been added; and the user will be notified by another Toast to add a module. Having a message reminding the user to select the correct module each time a Pomodoro starts will help ensure each Pomodoro is assigned to the correct module.

The timer itself was built with the use of a video tutorial referenced in the code comments, this tutorial helped build a fixed length simple timer using the android CountDownTimer class (android.os.CountDownTimer).

This basic timer was then modified into a Pomodoro timer. The timer needed to support 3 lengths, the actual Pomodoro (work period), short break and long break . Three 'long' variables hold these values in milliseconds. These variables have the default values mentioned above (25, 5 and 15 minutes) but they can also be set by the PomodoroSettings activity. This activity is opened by clicking the settings cog on the top right of the Pomodoro page and presents the user with three labelled Seek bars to change the timer intervals and a TextView explaining how many points the user will earn per Pomodoro with their current intervals.

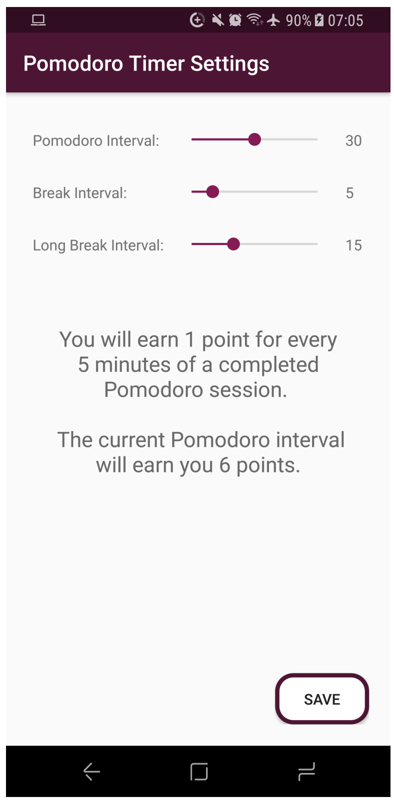

Figure 14: A screenshot of the Modules activity, accessed through the Pomodoro page.

The timer also needed to iterate between Pomodoro sessions (for work) and break sessions. A Boolean called 'breakOrWork' keeps track of this with 'true' meaning the current iteration is a break and 'false' meaning the current iteration is a Pomodoro. When the activity is first opened the first iteration is, of course, a Pomodoro and not a break. The Pomodoro class uses a 'breakCount' integer to keep count of the number of breaks a user has had. After each break session, the breakCount is incremented by one and this value is used to work out which break (short or long) to give the user. If 'breakCount%4 == 0', meaning if breakCount mod 4 is equal to zero, then the user has had a multiple of 4 breaks, and therefore 4 Pomodoro sessions and the timer is set to long break. Otherwise, the timer is set to short break (assuming breakOrWork=true).

When the start button is clicked, an alert dialog opens asking the user to enter a target for the Pomodoro (an extra point is awarded for the Pomodoro if a target is entered). The user can choose to skip entering a target and the timer countdown then begins. When the timer ends a summary alert dialog opens asking the user to enter a summary and a check a checkbox if the target was achieved. This too can be skipped. A PomodoroInstance object containing the target, summary, success and date/time of the Pomodoro is saved to the database at the end of a Pomodoro session along with the points the user has earned; though this does not happen if the timer is skipped and the session does not run to the end as only complete Pomodoro's are recorded.

A user can view their revision history by clicking the History button on the Pomodoro page which takes them to the PomodoroHistory activity. This page uses another scrolling ListView populated by a ListView adapter (PomodoroAdapter.java). The page shows all Pomodoro's a user has completed with their lengths, targets, summaries and modules. A red row indicates a failed Pomodoro and a green row indicates a successful one.

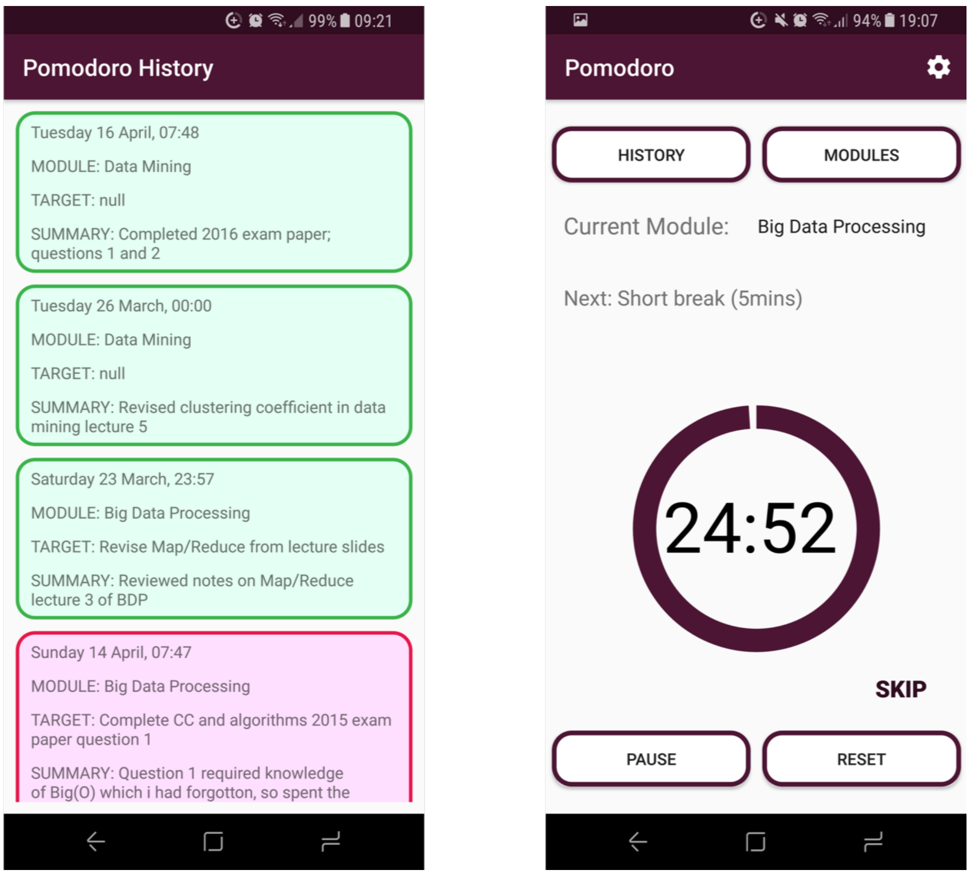

Figure 15: Screenshots of the PomodoroHistory activity (left), accessed through the Pomodoro page and the main Pomodoro activity (right).

As mentioned, only a Pomodoro session that reaches the end of its time is recorded. Clicking the skip button will skip the current iteration of the Pomodoro timer and delete any target or other information for that Pomodoro currently recorded. A timer can be paused and the remaining Pomodoro can be continued later which will retain the Pomodoro's information. Clicking the reset button will reset the current iteration of the Pomodoro timer (break or work) without moving to the next stage.

To prevent the timer from resetting as soon as the user leaves the timer activity, the onStop method is overridden and uses SharedPreferences to store timer variables (timeLeft, breakCount etc.). The onStart method, called after onCreate then reinstates these values allowing the timer to continue. The timer sounds a bell at the end of each iteration. This bell sounds even if the user has left the application; however, if the application is deleted from recent apps then the timer will not sound.

5.2 Flash-cards

The flash cards page allows users to create double sided flash cards, with a prompt and an answer. For flash cards to be used by students they needed a method of organisation. So, the first page a user see's when clicking the flash cards button from the home activity is a scrollable list of modules. This ListView is populated by an ArrayList adapter using data from the same list as the modules screen in the Pomodoro page. Meaning a user can use added modules to create flash cards for them and complete Pomodoro sessions for those modules. A floating action button on this screen opens an alert dialog allowing a user to add a new module.

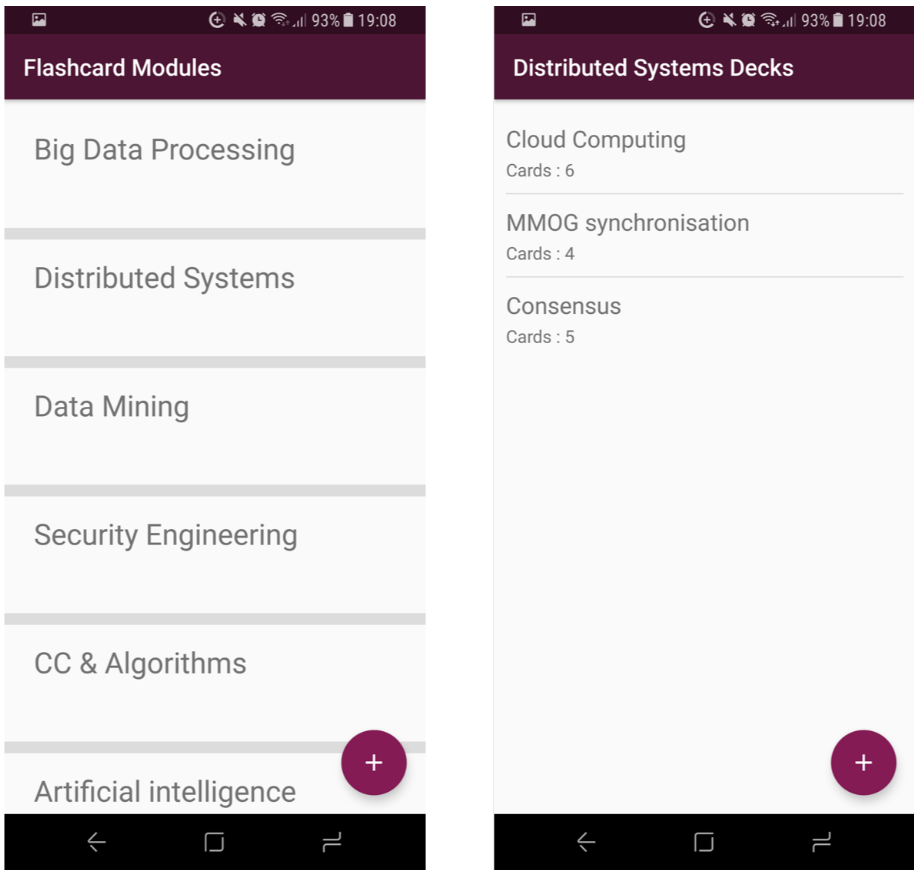

Figure 16: Screenshots of the Flash-cards modules activity (left) and the Flash-cards decks activity (right).

Clicking any of the modules in the flash cards list will take the user to a similar list of decks. Each module a student studies will have many topics, and these topics can be represented by the decks of cards. The list is another ListView populated by an adapter accessing data from firebase database. New decks can be added via alert dialog by clicking the floating '+' button.

Once the user selects a deck, they are then taken to a grid like view of all cards in that deck. The layout is a RecycleView populated by an adapter using firebase data. The grid of cards was built with the aid of a video tutorial, referenced in code, which demonstrated how to build a similar page for displaying books. Each card in the RecycleView shows the card prompt to identify the card. Selecting a card takes the user to another activity showing only the cards prompt so that the user can try to remember the content (utilising the benefits of practice testing). The user can then click the prompt to view the remainder of the card and test how well they remembered it. A floating button on the bottom right of the RecycleView allows users to add a new card using the CreateCard activity which then stores new cards in the firebase database.

5.3 Schedule

The Schedule activity presents the user with a scrolling list of events. Events are ordered by their date and time, with events that are nearest at the top of the list. When the system time passes an event end time, the event is automatically removed from the list. The events are read from the firebase database and stored in an ArrayList. Each event read from the database has its end time checked against current time, and if the end time is passed that event is not passed to the EventsAdapter. The events are sorted using the Collections framework (via Collections.sort); first ordered by (start) time and then ordered by date. This gives a list of events ordered by both time and date. The ArrayList is passed to the EventsAdapter class which inflates the xml layout for the ListView row and populates the ListView in schedule with information from the eventObjects passed to it in the ArrayList. Clicking an event opens an alert dialog showing all details of that event, some of which (notes and location), are not shown in the ListView.

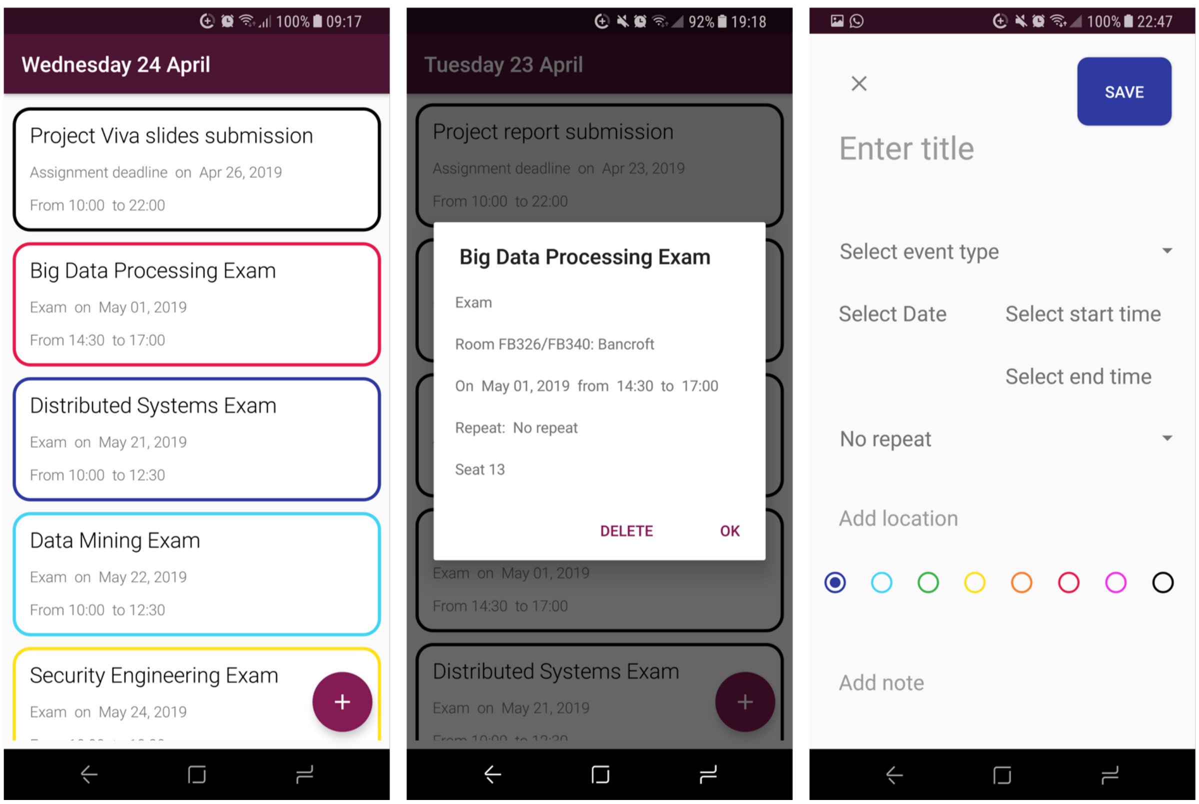

Figure 17: Screenshots of the Schedule activity (left), an alert dialog opened by clicking a schedule event (middle) and the CreateEvent activity (right).

New events are created by clicking the floating '+' button which leads the user to a CreateEvent activity. This activity uses a GridView to arrange a series of TextView's, EditText's, Spinners and Radio buttons to allow the user to enter an event. Date and time of events are entered through date-picker and time-picker dialogs which open up when the 'Select date' or 'Select time' text is clicked. The event type and repeat can be selected from a spinner. If an event is repeated, it repeats for a total of 14 weeks which is the length of the longest university semester. This will make it easier for

University students to use ExamCompanion to save their time table. When the save button is clicked, the CreateEvent class checks that all fields entered are valid (date is not in the past, event ends after it begins etc.) and if so, it creates an EventObject and uploads this to the firebase database so it can show in the Schedule ListView. If the event is set to repeat, the application saves the event multiple times using for-loops and adding the required time to the event in each iteration.

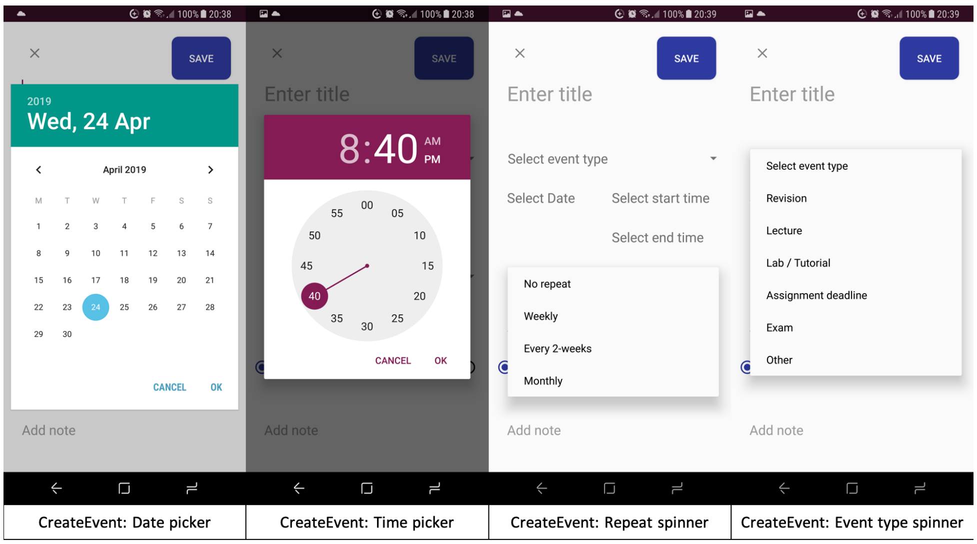

Figure 18: Screenshots of CreateEvent activity date-picker dialog, time-picker dialog, event repeat spinner and event type spinner.

5.4 Leaderboard

The Leaderboard activity shows a list of all users of the application, ordered by the number of points they have. When a new user first logs in to the application, the Home activity creates a UserObject and uploads this to the firebase database. This user object can be read by any other user, so it does not contain any information that is not publicly available. The users profile name (from their Google account) is saved along with a URL to their profile picture and a point count. Each user can only write to their own UserObject but can read all of them. This is necessary to populate the ListView in the Leaderboard activity with users. The ListView reads UserObjects from firebase, adds them to an ArrayList which is inputs as an argument to the LeaderboardAdapter class. The adapter then inflates each row of the listview with each user's image, name and points. Points are earned by users for creating flash cards (two points per flash card created) and completing Pomodoro's (one point for every 5-minutes of a complete Pomodoro). When points are earned, users are alerted by a Toast message and the points are uploaded to firebase so that the Leaderboard can be immediately updated.

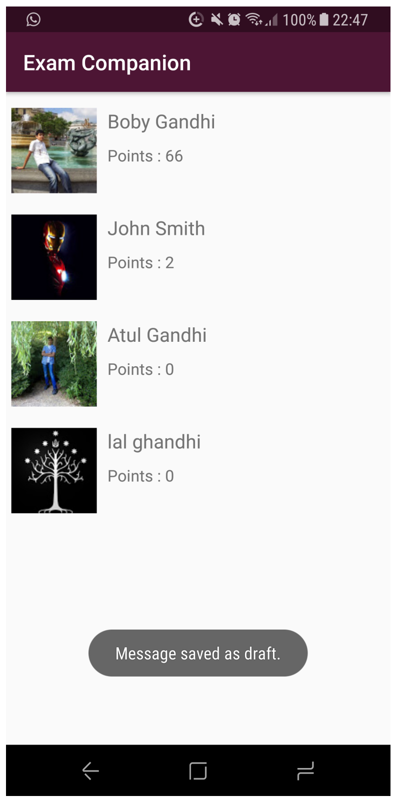

Figure 19: Screenshot of the Leaderboard activity displaying 4 users and their points.

6 Validation

6.1 Black Box Testing

Two university students with no knowledge of Exam Companions code were selected to test various features of the application using black box testing methods. The aim of black box testing was to discover any remaining bugs in the application that would affect its functionality. In addition; this test also sought to discover the complexity of using the application for a user with no previous knowledge of it and to gain some feedback on what users found was difficult or missing in the final product. The applications purpose and key features were explained to the testers, but they were not given any direction on how to use or access the features. The application was also already signed into a new account with no user information as the sign in functionality was not being tested.

6.1.1 Pomodoro

Testers were first asked to enter the Pomodoro page, change the timer's intervals and begin a Pomodoro revision session. Testers found the Pomodoro page from the Home activity with ease. They also navigated to the settings button on the top right of the Pomodoro screen to change the timer's intervals. One tester said the 'TextView' explaining how many points would be gained was helpful and both testers said changing the timer's intervals with sliders bars was an easy to understand and convenient method. The testers then returned to the main Pomodoro activity and tried to begin a timer but since the new account had no modules saved, instead of starting, the timer displayed a Toast message stating "Add a module in the modules page, then select it by clicking the text next to 'Current Module' ". This lead both users to the 'Modules' activity using the 'Modules' button on the Pomodoro page, as intended. Users added a module using the alert dialog and returned to the Pomodoro activity. Clicking the 'Start' button this time opened the target dialog, where both users entered text and clicked save. This started the timer and displayed another Toast message: "Make sure you've selected the correct module name". Since only one module was added and it was already selected by the 'Spinner', this message was irrelevant to both users. It did, however, lead one of the testers to click the 'Spinner' text showing their added module, opening a dialog with a 'list' showing their added module. Users were informed that clicking the skip button would skip their current Pomodoro without saving it and clicking reset would reset the current Pomodoro time. Both testers skipped their Pomodoro and a Long break began.

This was because when each timer iteration began, the 'setTime' method in the Pomodoro.java file checked if 'breakCount%4 == 0' to discover if 4 iterations of Pomodoro had happened and therefore a long break was warranted. While this worked to discover each time 4 Pomodoro's had been completed, since 'breakCount' was initialised as zero and '0 mod 4' is also equal to zero, the timer was making the users first break long instead of short. This issue was fixed by initialising 'breakCount' as one and incrementing 'breakCount' after each break session rather than before.

6.1.2 Flash-cards

The Testers were next directed to create a flash card. Selecting the flash cards page from the Home activity took them to a list holding the module they added in the Pomodoro page. Clicking this module took them to a blank page labelled 'Decks' with a floating '+' button. Both testers used this to add a deck via the alert dialog asking for a deck name; the added deck appeared on the page which they clicked to get to another black page with a floating '+' button labelled 'Cards'. Clicking this one took users to a 'CreateCard' activity, where they entered a card prompt and content and saved, going back to the Cards activity (which displayed a Toast message: "You have earned 2 points") where their card was displayed with its prompt showing. Selecting this card expanded the prompt with a Toast message stating "Click card prompt to show content". The testers then clicked the prompt to view the content.

The flash cards page worked as intended and without error. Testers found it easy to create new modules, decks and cards stated the overall interface was simple. One of the testers said it was cumbersome to go through modules and decks to get to the card (requiring several clicks to read a card); however, the multiple pages design was intentional.

The flash cards section of the application is designed to hold many cards and it would be most useful to students if the cards are organised in a similar fashion to their degrees. With many modules, and each module containing many decks (for module topics/lectures) and each deck containing the cards for that module. While a tester that created one flash card may find it cumbersome to get to the card from the Home activity, in real world use the benefits of organising cards into modules and decks will outweigh the drawbacks of going through several activities to reach a card.

6.1.3 Schedule

The testers were then asked to create an event in Schedule. They accessed the Schedule activity (currently a blank activity with the floating button) and clicked the floating '+' button to add an event. This took them to the 'CreateEvent' activity where they entered the events details and clicked the save button; taking them back to Schedule with a row showing their new event. Clicking this event displayed a dialog window showing all event details. The testers were asked to create several more events to ensure the ordering of events by date and time was functional and there were no unknown input errors.

The Schedule activity was also fully functional. The testers did not find the 'CreateEvent' activity to be overwhelming; a concern during the designing process due to the activities many input fields. Testers found using spinners to select event repeats and types instead of typing them was more convenient and they mentioned that using dialogs to enter date and time made doing so easier. One of the testers commented that events could have start times and end times but only one date, meaning an event could not span for longer than a day, he also noted that there was no way to set a full-day event. These options were initially not included in the 'CreateEvent' activity to prevent it from becoming too complicated for the average user by having to many fields to fill out. Since the Schedule was designed for academic events, it also seemed unnecessary as academic events such as lectures, assignment deadlines and exams will have specific times and will not last longer than a day. The suggestions were, however, added to possible further work for the application – to expand the schedule activity so that it could be more easily used for non-academic events as well as academic ones.

6.1.4 Leaderboard

Finally, the testers were asked to view the Leaderboard. The Leaderboard held two accounts created for white box testing purposes in addition to the accounts created for the testers. The testers accounts both showed 2 points for the flash card they created and were in the list of four accounts ordered by points. The Leaderborard activity worked as intended and the testers did not make any further suggestions concerning it.

6.2 White Box Testing

White box testing was used to detect use case paths that might cause errors or bugs in the application, including invalid inputs into fields. The class for an activity that accepts an input, directly or via alert dialog, includes a 'checkIfValid' method to ensure inputs are not null, empty or otherwise invalid. White box testing was used to ensure this method was functional and prevented all invalid inputs from being saved or crashing the application.

All activities or dialogs where inputs could be added were tested with plain text input (where possible), empty input and numerical input. Each attempt lead to the 'checkIfValid' method stopping the input from being saved to the database and returned a Toast message stating why the input entered was incorrect. The 'CreateEvent' acivity was tested by creating events the had already ended or ended before they began; both of these invalid inputs were detected by the 'checkIfValid' method.

In addition to testing input errors, many repeating events were added to the schedule (symbolising a timetable) which were correctly sorted and displayed. The Pomodoro section also functioned correct, with short breaks and long breaks occurring at the correct Pomodoro iterations and complete Pomodoro's being saved to the History page with a red or green background depending on if the 'Target Achieved' checkbox was ticked in the Pomodoro summary dialog.

7 Conclusion

7.1 Summary

This project reviewed literature concerning revision methods used in education to discover how those methods could be changed to improve the learning and productivity of students in higher education. The project found methods such as the Pomodoro technique were effective at improving productivity during intensive work periods and active learning methods in education helped improve understanding and recall over passive learning methods.

Over the course of the project, the research questions first posed were answered, and a complete and functional software application was built that utilises the research to improve the productivity and learning of its users. The Pomodoro timer can be used by students to track and time their revision and to lower the number of distractions that affect their revision. The flash cards pages can be used to create and review flash cards multiple times, utilising active learning (creating cards), spaced repetition (reviewing cards) and practice testing (trying to remember card content using just prompt).

References

[1] Robert S. Allen, Vernon L.; Feldman. Learning through tutoring: Low-achieving childern as tutors. Wisconsin Univ., Madison, 1972. Wisconsin Research and Development Center for Cognitive Learning, Wisconsin Research and Development Center for Cognitive Learning.

[2] Gabriel Barata, Sandra Gama, Joaquim Jorge, and Daniel Gonçalves. Gamification for smarter learning: tales from the trenches. *Smart Learning Environments*, 2, 12 2015.

[3] Carl A Benware and Edward L Deci. Quality of learning with an active versus passive motivational set. *American Educational Research Journal*, 21(4):755–765, jan 1984.

[4] Kay Berkling and Christoph Thomas. Gamification of a software engineering course and a detailed analysis of the factors that lead to it's failure. pages 525–530, 09 2013.

[5] Simone Borges, Vinicius Durelli, Helena Reis, and Seiji Isotani. A systematic mapping on gamification applied to education. 03 2014.

[6] Burke H. Bretzing and Raymond W. Kulhavy. Notetaking and depth of processing. *Contemporary Educational Psychology*, 4(2):145–153, apr 1979.

[7] Robin Brewer, Lisa Anthony, Quincy Brown, Germaine Irwin, Jaye Nias, and Berthel Tate. Using gamification to motivate children to complete empirical studies in lab environments. In *Proceedings of the 12th International Conference on Interaction Design and Children - IDC 13*. ACM Press, 2013.

[8] Shannon Butler and Dewan Tanvir Ahmed. Gamification to engage and motivate students to achieve computer science learning goals. In *2016 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE, dec 2016.

[9] Nicholas J. Cepeda, Harold Pashler, Edward Vul, John T. Wixted, and Doug Rohrer. Distributed practice in verbal recall tasks: A review and quantitative synthesis. *Psychological Bulletin*, 132(3):354–380, 2006.

[10] Nicholas J. Cepeda, Edward Vul, Doug Rohrer, John T. Wixted, and Harold Pashler. Spacing effects in learning. *Psychological Science*, 19(11):1095–1102, nov 2008.

[11] Robert D. Cloward. Studies in tutoring. *The Journal of Experimental Education*, 36(1):14–25, sep 1967.

[12] John Dunlosky, Katherine A. Rawson, Elizabeth J. Marsh, Mitchell J. Nathan, and Daniel T. Willingham. Improving students' learning with effective learning techniques. *Psychological Science in the Public Interest*, 14(1):4–58, jan 2013.

[13] Edwina E. Abott. On the analysis of the factors of recall in the learning process. *Psychol. Monogr*, 11, 01 1909.

[14] Gilles O. Einstein, Joy Morris, and Susan Smith. Note-taking, individual differences, and memory for lecture information. *Journal of Educational Psychology*, 77(5):522–532, 1985.

[15] Bethany E. Forbes, Christopher H. Skinner, Michelle P. Black, Jared Yaw, Joshua Booher, and Jean Delisle. Learning rates and known-to-unknown flash-card ratios: Comparing effectiveness while holding instructional time constant. *Journal of Applied Behavior Analysis*, 46(4):832–837, aug 2013.

[16] David Gibson, Nathaniel Ostashewski, Kim Flintoff, Sheryl Grant, and Erin Knight. Digital badges in education. *Education and Information Technologies*, 20(2):403–410, nov 2013.

[17] Geoff Goehle. Gamification and web-based homework. *PRIMUS*, 23(3):234–246, mar 2013.

[18] Marissa K. Hartwig and John Dunlosky. Study strategies of college students: Are self-testing and scheduling related to achievement? *Psychonomic Bulletin & Review*, 19(1):126–134, nov 2011.

[19] Sean H. K. Kang. Spaced repetition promotes efficient and effective learning. *Policy Insights from the Behavioral and Brain Sciences*, 3(1):12–19, jan 2016.

[20] Alison King. Comparison of self-questioning, summarizing, and notetaking-review as strategies for learning from lectures. *American Educational Research Journal*, 29(2):303–323, jun 1992.

[21] Sara Kupzyk, Edward J Daly, and Melissa N Andersen. A COMPARISON OF TWO FLASH-CARD METHODS FOR IMPROVING SIGHT-WORD READING. *Journal of Applied Behavior Analysis*, 44(4):781–792, 2011.

[22] Therese H. Macan, Comila Shahani, Robert L. Dipboye, and Amanda P. Phillips. College students time management: Correlations with academic performance and stress. *Journal of Educational Psychology*, 82(4):760–768, 1990.

[23] Fiona Fui-Hoon Nah, Qing Zeng, Venkata Rajasekhar Telaprolu, Abhishek Padmanabhuni Ayyappa, and Brenda Eschenbrenner. Gamification of education: A review of literature. In *Lecture Notes in Computer Science*, pages 401–409. Springer International Publishing, 2014.

[24] Siobhan ODonovan, James Gain, and Patrick Marais. A case study in the gamification of a university-level games development course. In *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference on - SAICSIT 13*. ACM Press, 2013.

[25] Bernhard PastÃ¶tter and Karl-Heinz T. BÃuml. Retrieval practice enhances new learning: the forward effect of testing. *Frontiers in Psychology*, 5, apr 2014.

[26] Michael Prince. Does active learning work? a review of the research. *Journal of Engineering Education*, 93(3):223–231, jul 2004.

[27] P. RAMSDEN and N. J. ENTWISTLE. EFFECTS OF ACADEMIC DEPARTMENTS ON STUDENTS APPROACHES TO STUDYING. *British Journal of Educational Psychology*, 51(3):368–383, nov 1981.

[28] Siddharth Reddy, Igor Labutov, Siddhartha Banerjee, and Thorsten Joachims. Unbounded human learning. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD 16*. ACM Press, 2016.

[29] Kelly Braun David C. Rubin. The spacing effect depends on an encoding deficit, retrieval, and time in working memory: Evidence. *Memory*, 6(1):37–66, jan 1998.

[30] Mintra Ruensuk. An implementation to reduce internal/external interruptions in agile software development using pomodoro technique. In *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*. IEEE, jun 2016.

[31] Willard N. Runquist. Some effects of remembering on forgetting. *Memory & Cognition*, 11(6):641–650, nov 1983.

[32] Samuel J. Thios and Paul R. DAgostino. Effects of repetition as a function of study-phase retrieval. *Journal of Verbal Learning and Verbal Behavior*, 15(5):529–536, oct 1976.

[33] Xiaofeng Wang, Federico Gobbo, and Michael Lane. Turning time from enemy into an ally using the pomodoro technique. In *Agility Across Time and Space*, pages 149–166. Springer Berlin Heidelberg, 2010.

[34] Kathryn T. Wissman, Katherine A. Rawson, and Mary A. Pyc. How and when do students use flashcards? *Memory*, 20(6):568–579, aug 2012.

8 Appendix

8.1 Risk Assessment

Table 4: Project Risk-assessment

Description of risk	Description of impact	Likelihood rating	Impact rating	Preventative actions
Illness or accident	Could case project deadlines to be missed and cause delays in completion of project resulting in an unfinished final product.	Low	High	Risk can be reduced by taking greater precautions to ensure personal health and safety. This would include promptly checking up any medical issues with a doctor to prevent a long-term illness that would impact progress on the project, as well as avoiding unnecessary risks such as jogging at night when visibility for cars is low and accidents can happen.
Poor time management	Missing deadlines and managing time poorly could result in an unfinished project, lacking in basic features and poorly analysed and tested with software bugs and errors.	Medium	High	Project progress will be reviewed on a weekly basis and noted down in a log along with any issues that set back the project – so that in the following week those issues can be addressed. Project progress will also be compared to the time plan on a weekly basis, and where the project diverts from the time plan, it will be remodelled to fit in all required work by the deadlines set.
Failure to acquire participants for surveys, interviews or testing	Could cause an inability to conduct important research for the project resulting in a final product that is unfit for its target demographic.	Low	High	The methods used to attract participants for surveys and interviews can be diversified. Online websites and social media platforms can be used to advertise surveys where required to gain a large sample size for research conducted. Participants for interviews can be gained through advertising to peers at Queen Mary's as well as by contacting students from other institutions through social media to participate in the study. If the above two methods still do not attract the required number of participants for research, financial incentive could be provided though websites such as Google Surveys.

Table 4: Project Risk-assessment

Description of risk	Description of impact	Likelihood rating	Impact rating	Preventative actions
Failure to acquire required information	If suitable research articles and journals do not exist or cannot be accessed, there could be a lack of research available relevant to the project. This could make the final product unable to fulfil the projects objectives.	Low	High	Multiple online sources will be used to search for and find relevant research materials. In addition, research sections of public and academic library's will be used so that a suitable number of relevant studies can be used in the product research.
Inadequate testing	Poor testing methods and analysis could lead to bugs and errors in the final application passing through the testing phase without being fixed.	Medium	High	The completed application will be debugged and tested by the developer and other testing participants using black-box and white-box testing methods to ensure thorough testing.