Final Report for the Student Planner Mobile App Project.

Ronald Grant

***Abstract*—Time management for students is a significant challenge for students. Evidence indicates that without adequate tools or support, many students struggle with time management, leading to increased procrastination, increased workloads, missed deadlines, and poorer academic outcomes. While mobile time management apps exist, they often present steep learning curves and demand significant time investments to be effective. The Student Planner App project aims to address these barriers by developing a mobile app featuring a calendar-based heat map to provide an intuitive visualisation of a student's current workload and future periods of increased workload. This visualisation highlights areas of increased workload and thus provides feedback on students' time management across their courses to provide them with proactive time-management opportunities. Additionally, the app utilises gamification principles to enhance student engagement with the application and their studies. The app's design allows the potential for integration with academic platforms to automate configuration and minimise the effort required to use the app. Therefore, the Student Planner App project can enhance student academic achievement and mental well-being by offering an intuitive solution to student time management challenges. User testing showed positive responses to the usability and functionality of the app. This positive feedback suggests that the SPA effectively supports students in managing their time and tasks, potentially leading to improved academic outcomes and decreased stress. Potential avenues for future development involve expanding the application's gamification features, such as adding level progression, unique items, and experience points to enhance user engagement further.**

***Index Terms*—Keywords: Mobile Application, User Experience, Learning.**

I. Introduction

TIME management is a significant challenge for undergraduate students, who must manage meeting the obligations of multiple courses over a set period. A student's development of time management skills is expected to include creating strategies to complete their study tasks, making time for lectures, tutorials, assignments, test revision and other contributing tasks [1]. Frequently, procrastination is the outcome of poor time management. [2]. Procrastination delays completing course activities and contributes to study-related stress and poorer mental health outcomes [3]. Both procrastination and time management are significant factors in academic performance [4]–[6]. The difficulty in effective time management can be attributed to many factors affecting students. Research suggests that undergraduate students' last-minute work results from a lack of understanding of using unscheduled periods of their time alongside balancing academic and non-academic commitments [7]. Furthermore, despite wanting to be better organised, students may need more strategies or skills to manage themselves better and set goals [4]. In contrast, adequate support and training of students in their time management skills have shown an increase in their perceived control of time, decreased stress levels and contributed to their general welfare [8], [9]. Furthermore, effective time-managing behaviours in students lead to a decrease in procrastination, distraction, and making better use of their time overall [10].

Despite the benefits of effective time management on student well-being, evidence shows that practising effective time management without aid or support remains challenging for many students. Various technologies have been suggested to support students in their time management. An example of such technology is a digital time management tool proposed by Raadt and Dekeyser [11], which showed success in supporting students' time management and engagement. Utilising mobile technology to support students in educational contexts is of particular interest. Smartphones enable the convenience of accessing course materials and resources at any time through the internet, enhancing students' independence, enhancing engagement with their learning, and providing distance learning opportunities [12], [13]. Mobile apps bring opportunities for more personalised learning opportunities and deeper understanding through discussion on social media [14]. Specialised mobile apps can visualise complex systems, such as human anatomy, or create personalised study materials that allow students to learn at their own pace, such as quiz apps or note-taking apps [15]. An example is shown through using apps for daily journaling, which positively impacted self-reliant learning in conjunction with online learning [16].

Mobile devices hold the potential to support student's academic activities. As mobile devices are seeing widespread use by students to support educational activities and learning via internet resources or mobile apps [15], [17], it is necessary to consider utilising the mobile platform to help students with effective time management. Recognising this need, this project introduces the Student Planner App (SPA), a tool to support students with their time management challenges. SPA's goal is to provide the student with a way of intuitively finding their progress within each course, encouraging engagement with their courses, and highlighting areas of increased workload. SPA aims to make courses more manageable by providing an overview of the workload over set periods and giving recommendations on which to prioritise and manage course activities. SPA also aims to encourage building time management skills for those who use it. This goal is crucial for first-year undergraduates, who may need help establishing good time

This project was supervised by Thomas Kühne.

management skills when transitioning from secondary school [4], [7]. The SPA's design aligns with the United Nations goals of Sustainable Development. By building skills to improve students' well-being, decrease academic barriers and promote better learning outcomes, the SPA aligns with the "Quality Education" goal and partly with the "Good Health and Well-being" goal [18].

The final product of this project utilises a heat map to visualise workload to aid students in finding and preparing for periods of increased workload in their university schedule. Busy workloads are determined by combining a task 'health' metric, which considers the completion stage of a task and how long it is until the task is due, with another metric that calculates simultaneous tasks for that day. High workload days are visualised as a deeper red colour. Student progression is visualised, using the average health metric for the student's active task to communicate how well the user keeps up with all of their work. Automated setup and configuration of the user's calendar based on their enrolled university courses minimises the effort to use SPA. Task prioritisation through a task list allows students to find which tasks they should focus on quickly. Finally, challenge popups motivate users to engage with tasks that are high priority or need more attention. The evaluation was performed for this project through user testing. User testing of SPA showed positive results, with users expressing high satisfaction in all areas evaluated via Likert-scale questions. Users found SPA's interface intuitive and responsive and appreciated its functionality for aiding task organisation, time management, planning, and motivation, indicating high usability. Reception to the activity list screen, calendar workload visualisation, and challenge popups were positive. User feedback also highlighted areas for improvement for SPA. The red-gradient colour scheme of the workload visualisation was noted as potentially anxiety-inducing or unclear for distinguishing workloads. Users also suggested enhancing the challenge feature by displaying more challenges at a time and introducing push notifications outside of the app.

II. Background and Related Work

A. Mobile Technology to Support Education

The utilisation of mobile technology within education is a widely explored area of research. The development of mobile technologies, faster processing speeds, ubiquitous online connectivity and students' widespread usage of these mobile devices [15] facilitate its increased adoption within learning environments to support students and staff. The m-Advisor, implemented by Shambour et al. [19], is a mobile application solution for students' time-consuming task of course-advising. Reducing the time-consuming nature of course advising can provide students and course advisors more space for other types of productivity. Since m-Advisor directly communicates with a university database, the program can retrieve relevant student data and academic program information without manual configuration. Through the m-Advisor's UI, a student can determine the availability, requirements, and details of courses, giving the student all the necessary information needed to make decisions for the future of their academic endeavours. The m-Advisor demonstrates how a mobile app implementation reduces the time and effort investment needed in course advising, which could include booking appointments with course advisors. Importantly, m-Advisor makes academic information more accessible and personalised, which is critical in enhancing student independence [12]. The M-advisor provides a basis for the functionality of the SPA, specifically in applying the automated configuration of the application based on the user's enrolled courses and assigned tasks. This aspect of SPA's functionality is critical in minimizing the time commitment required to make the SPA usable.

B. Mobile Technology for Time Management

Personalised learning advice is essential in online-based learning, which demands skills in self-directed learning. However, online-based learning lacks the learning support present with in-person higher education institutions [20]. The MyLearningMentor (MLM) app implemented by Alario et al. [20] recommends adaptive planning advice based on a user's current standing within their courses to build fundamental work and study habits. MLM uses the metrics of Preferences, Priorities and Performance (how much time has been committed to previous tasks). Notably, the MLM achieves personalised advice through adaptive planning, in which the planner updates its recommendations based on the previous performance and priorities of the user. MLM emphasises engagement through interaction with the app. Specifically, users provide new information to MLM, including the time committed to tasks and task completion, which will determine performance. Furthermore, advice can be given through tips, hints and suggestions based on performance and collected user data. MLM focuses on massive online massive open online courses (MOOCs). Nevertheless, Alario et al.'s findings can be applied to the increasing digitisation of higher education. The SPA, while providing functionality similar to MLM, will extend the work ideas presented by MLM by visualising the project workload of students over set periods.

Existing time management apps often require manual setup and configuration of tasks. Liyange et al. acknowledge that this paradigm is accessible for more organised students and is not accessible to many who may not have the perseverance necessary to configure these apps enough to make them useful [21]. This barrier is a problem since many existing time management tools require time-consuming setups, limitations in task scheduling, and steep learning curves for using the tools effectively [22]. A time-management assistant app prototype was created by Liyange et al., which addresses the limitations of manually configured time-management apps [21]. The prototype emphasises the need for a user-friendly design that minimises users' input to use the app. Minimisation of inputs includes only requiring the user to input information once a week, automating the prioritisation of tasks, and providing quick access to relevant information and suggestions based on user preferences. This emphasis informs the design of tasks within the prototype, which communicates the amount of time needed per week, when their deadlines are, and orders them by priority. Furthermore, the user can track the progress of all current activities on a 0-100 scale, allowing a more comprehensive view of progress across all activities, which, combined with minimising user effort, dramatically contributes to the app's ease of use. These quality-of-life features are valuable to include in the SPA to maximise its immediate usefulness to students.

Similarly, the design of ScheduleME aims to address the usability issues present in existing time management tools [22]. The automated setup of ScheduleME and the minimisation of user input are achieved through machine learning. As information and course data are available online, ScheduleME uses data scraping to collect course-related data such as assignments, deadlines and exam timetables. A genetic algorithm then streamlines the collected data into schedulable tasks, deadlines, events, and important dates. Personalised scheduling of tasks is implemented using reinforcement learning, which can adapt to a user's behaviour, identify free time slots, and reallocate tasks if users do not complete them. Compared to other time management tools, which have been observed as difficult and time-consuming to learn [22], the automation ScheduleME significantly decreases the inputs required for a user to set up the app. While the utilisation of machine learning may be valuable in generalising the SPA's applicability to any institution, it is an engineering problem that is out of this project's scope.

C. Application of Gamification Features

Within the context of mobile applications, gamification provides game elements such as leaderboards, rewards systems, challenges and feedback to the experience. Increasing engagement through the gamification of time management applications can contradict the purpose of time management. However, gamification of educational applications aims to increase engagement with the course content rather than the app itself. Gamification within educational contexts has seen positive benefits in student motivation, participation, and overall academic outcomes [23], [24]. Educational Applications such as DevHub and Stack Overflow utilise gamification through badges, user rewards and reputation to increase engagement with the application [25]. Pechenkina et al. [24] evaluated the effectiveness of a gamified learning mobile app utilising progression trees, leaderboards, badges and notifications as gamified incentives. Usage of the gamified app showed increases in student retention rates and academic performance. Furthermore, higher scores indicating engagement with the app correlated to higher academic achievement. The authors conclude that gamified apps contribute to the growing demand for personalised learning. Furthermore, the app provided helpful feedback to students through badges to mark milestones and leaderboards to measure their performance compared to their peers.

The inclusion of gamification also shows the potential to promote effective time management. Study planners can be extended to add interaction elements, such as utilising animations to reward completed tasks. Furthermore, tasks can be represented as games to increase user engagement. One example represents progress as a Tetris game and rewards Tetris blocks for every completed task to progress a Tetris game to incentivise regular engagement with the application [26]. Adaptive companions as a gamified feature could provide personification to hints and feedback, similar to those found in Duolingo [27]. Cassells and Broin implemented a gamified interface utilising an adaptive companion to enhance the experience of using study planner applications. The companion serves the role of helping with learning the study planner through instructions on how to use the application. The companions also provide personalised feedback, encouragement to complete goals, recognition of goals met, and negative reinforcement when behaviours such as procrastination arise. Similar to other time management apps, visual representations of the user's progression are provided. However, feedback recommendations and important events or dates are delivered through the companion, providing an aspect of personality to the notifications. Including a companion can increase the amount of enjoyment and motivation when prioritising tasks without requiring more effort from the user [27] and can potentially use other gamification techniques to optimise engagement. The SPA can utilise gamification to increase user engagement when using the application and, therefore, motivate them to engage with their time management. Furthermore, gamification can recontextualize otherwise trivial features, such as notifications and progression meters, into more exciting features, such as challenges or progression tree rewards. Although it is crucial to ensure that the combination of gamification techniques in a time management application does not end up distracting users through over-emphasis on in-app progression or other gamified effects [28]. Care when designing gamified activities is needed to ensure that they contribute to academic engagement and motivation to complete real-world tasks.

III. Tools and Methodology

A. Mobile Application Frameworks

Using hybrid frameworks is a popular form of development for mobile applications. The adoption of hybrid frameworks can be attributed to the increase in development efficiency afforded by cross-platform development using a single code base, code reuse through open-source plugins, and simplification of the packaging, deployment phases of development [29], [30] when compared to native development. The streamlined aspects of hybrid development facilitate the rapid development of mobile applications and are especially useful for small teams of developers [31]. However, these benefits are a trade-off with the overall performance and size of the application [29], which could negatively impact the application's usefulness depending on the platform and hardware. Therefore, choosing a practical hybrid framework is essential in maximising the benefits of hybrid development and mitigating the inherent drawbacks. Comparisons to the Apache Cordova, Flutter and React-Native frameworks show that Flutter provides a more effortless development experience, more extensive documentation and more straightforward configuration [30], [32], [33]. However, functionality and overall performance comparisons between Flutter and React-Native largely do not favour either framework [32], [34]. After considering these findings with the selection of hybrid frameworks, Flutter was chosen as the hybrid application framework for developing SPA. This decision to choose Flutter was influenced by its offering of libraries that greatly aided in developing SPA, in addition to previous experience, which demonstrated its effectiveness. Flutter's implementation language (Dart) also provided helpful programming language features such as type and null safety, making SPA less susceptible to bug propagation due to incorrect typing or null pointers.

B. Libraries and Tools

In developing the SPA, a selection of libraries and tools was made to aid the functionality and user experience of the app. Below is an overview of the libraries instrumental in SPA's development and their role within SPA.

- **Syncfusion Flutter Calendar:** The Syncfusion Flutter Calendar library provided the framework from which SPA's calendar UI is built. This library provided configurable widgets that handle the rendering of a calendar UI whilst providing methods of making any calendar view highly configurable. The usage of this library significantly decreased the amount of time in implementing the calendar view for SPA as it provided built-in logic for an interactive calendar UI, which included month, week and day views, each customised to fit the needs of SPA's implementation. The individual cells, representing days, were configured to communicate periods of increased workload by configuring their colouration.
- **Google Firebase:** Google's Firebase was chosen as it provides several services integral to SPA's functioning. Furthermore, Flutter has the FlutterFire library, which contains easy-to-implement APIs to interact with Firebase's services. Firebase Auth is used to implement user authentication through email and password. This service simplified the required logic needed to implement authentication and session persistence for SPA. Firebase storage (Firestore) is the database for course and user information SPA uses. Firestore is able to auto-scale its read-write infrastructure to match demand, which is especially helpful in ensuring the SPA database is scalable [35]. Utilising the services, in addition to the easy-to-implement APIs given by the FlutterFire library, greatly aided in simplifying the development of the SPA. Most significantly, it allowed SPA to use a consolidated library to utilise these related services. A separate database or authentication solution would have likely increased complexity and configuration when implementing SPA, especially if each required its own API libraries to function.

C. Development methodology

A development methodology must be adopted to set the development stages, pace and milestones for the SPA project to develop successfully. Both Agile and Waterfall methodologies aim to achieve this. While Agile is not typically suited for individual development, as Scrum and Kanban are intended for solving collaborative problems [36], it has been noted that teamwork and collaboration aspects of agile can be eliminated to make the framework more useful for singular developers [37]. However, Agile development focuses on delivering software iterations with feedback informing new development priorities. This delivery style is less suited for development in which multiple iterations are not feasible, and Waterfall may be more suited given that the project requirements are well-defined [38]. The evaluation of SPA's performance is conducted through a user testing period and will likely not have an opportunity to be iterated upon after delivery. Considering the use cases for both development methodologies, the Waterfall methodology was more appropriate for SPA's development, as it suited given the project's time constraints and well-defined project requirements.

IV. Design

Development of the SPA begins with establishing the requirements and constraints of the project. Since the SPA is intended for usage by university students, much consideration has been given to basing requirements on needs unique to university students.

A. Functional Requirements

A considerable barrier to utilising time management apps is the manual configuration of tasks, which can take considerable effort to make these apps usable [21]. The SPA should automate the creation of tasks, sub-tasks and deadlines to mitigate this barrier. To this end, users should have a schedule ready upon login to assist with time management. Automating this process necessitates storing data regarding students enrolled in courses and the tasks involved in completing a course prepared for retrieval. The SPA must be able to visualise the workload volume of these tasks over a specific period. Visualising this workload will give the user a quick and digestible overview of how many simultaneous tasks they need to manage over a set period so that they may plan their time accordingly. Furthermore, the visualisation needs to reflect the early completion or postponement of tasks that will impact workloads in the future. For example, postponing tasks will increase workloads at later dates when other new tasks appear in the schedule. The SPA must have a function to track the progress of tasks and allow users to record when tasks and sub-tasks are completed. A progress tracker that takes aspects of gamification should be used to boost engagement and communicate the completion across single or multiple tasks. The visualisation should also communicate how well the user keeps up with their workload over all their courses and tasks within a time period. SPA must also support users in prioritising tasks and visualising workloads and progression. To this end, gamification-inspired notifications or reminders should also be included in the SPA to increase user emphasis on completing tasks with upcoming deadlines or committing time to tasks that the user may be neglecting. A candidate for these gamified notifications is challenge notifications that carry some form of reward upon completion.

B. Non-Functional Requirements

The SPA must provide an intuitive and easy-to-navigate user interface to maximise usability. Minimising the actions required to access the main functions of the SPA is crucial in avoiding common barriers present in other time management apps, including steep learning curves or unintuitive user interfaces [22]. SPA's usability can be measured by examining how many actions are required to access the primary functional components of the app and how easily users can navigate to features. Good performance is another concern for the user interface of the SPA, as it must be responsive and provide minimal latency when loading the app or its content. This requirement contributes to the usability aspects of the application and is crucial in enhancing the user experience with the application. The functional components of a SPA must ensure a positive user experience. The workload visualizations and progression tracking should be easily interpreted and provide sufficient information to aid decision-making and planning. Furthermore, notifications or reminders should motivate users rather than being a source of annoyance or distraction while using the SPA.

Furthermore, the application must be lightweight in its size and hardware demands, which will require selecting efficient libraries and frameworks during development and optimising algorithms and processes that could potentially be resource-intensive. Minimising resource usage will ensure the application's UI and functionalities are responsive to users, even on devices with limited hardware capabilities. The SPA needs to be available at any hour of the day to accommodate the varying work habits of students, which is vital in facilitating their personal study needs and independence. The availability of SPA will likely be most affected by the availability of its external services (database, authentication service). Therefore, selecting reliable external services or utilising local storage is necessary. The database solution must also be fast to ensure the responsiveness of SPA is maintained. The SPA must not ask for or store a user's personal information beyond what is necessary to determine the user's enrolled course information and the progress of their courses. This user information must be kept private through secure authentication methods. Finally, updates and maintenance must be made possible without disrupting the application's primary functions. Furthermore, there is potential for future iterations and the sustainability of the SPA after this project life-cycle. Ensuring the maintainability of the SPA necessitates that the code should conform to the standards of the chosen framework and the appropriate use of documentation.

C. Internal Architecture

Figure 1 illustrates the design for architecture between the internal components of SPA's solution. The user interface will consist of several views to give students various ways of planning their courses. The following are the aspects of the requirements that are addressed in this design. For this description, *tasks* will refer to any time-sensitive activity students must undertake in their courses, such as assignments, tests and exams. A core aspect of the SPA is to provide a visual overview of the tasks and workload assigned to a student. This visualisation can be achieved using a calendar and heat map fusion. Combining these allows for an intuitive representation of the student's schedule and workload intensity. Tasks and deadlines are plotted on the calendar, creating a timeline of course-related activities, a standard function of calendar views. The heat map overlay adds a layer of visual insight, with the intensity of colour indicating the workload volume on specific days or periods. An underlying metric will be employed to calculate the approximate workload of a given day, which will inform the colouration of the individual days within the calendar view. This metric considers multiple factors, including the number of simultaneous tasks on a given day, the stage of completion for tasks, and how long until the task is due. Another visualisation will communicate the user's progress within their active tasks and display a progress meter. While complete gamification of this visualisation, which could involve adding level progression or items, is not in the scope of SPA, it will take gamification aspects to enhance engagement. Using gamification involves re-contextualising the progress meter as a form of self-competition, such as beating one's high score or winning a race by keeping up with workloads, represented by leading the race. These visualisations will allow students to identify areas of increased workload at a glance and understand their overall standing within all of their active tasks.

Beyond visualisation, SPA should also directly assist students in prioritising tasks. To this end, a similar metric must also be employed to determine which tasks to prioritise. This metric will be referred to as task *health*, which considers the remaining work for the task and how long until the task is due. Using this health metric will help prioritise tasks with low health, indicating a potentially unsatisfactory state of completion or impending deadline. A dedicated view of these prioritised tasks will display a list of the tasks by calculated priority and various other filters for flexibility. Furthermore, engagement with tasks with low health is further emphasised by introducing *challenges*. Similar to the progression view, challenges utilise gamification aspects to enhance engagement. Challenges will offer the user one or more tasks to complete within a time limit, potentially with a reward for completion. Encouraging users to engage with neglected tasks and refocusing their attention on different courses is the primary purpose of these challenges, especially in cases where students are spending too much time on one course or task.

The programming architecture will use a Model-View-Controller architecture (MVC) to separate the user interface, controller and database logic. This allows a separation of concerns between these elements to improve the maintainability and sustainability of the code base. Student data will be represented using a user model generated by an external database or previously saved information in local storage. Controllers for separate student data elements (tasks, challenges and enrolled courses) are used to update the relevant parts of the user model as students update task completion or accept challenges.

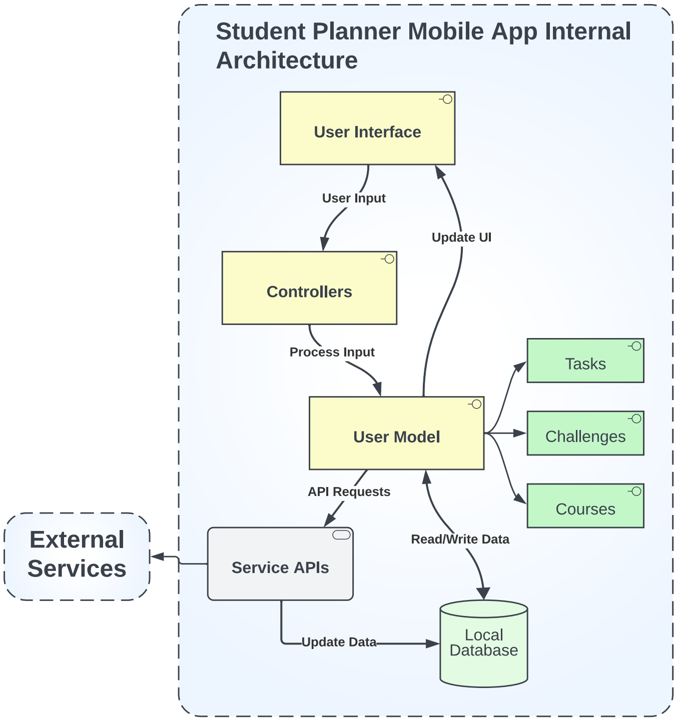

Fig. 1: Architecture of SPA Internal Components

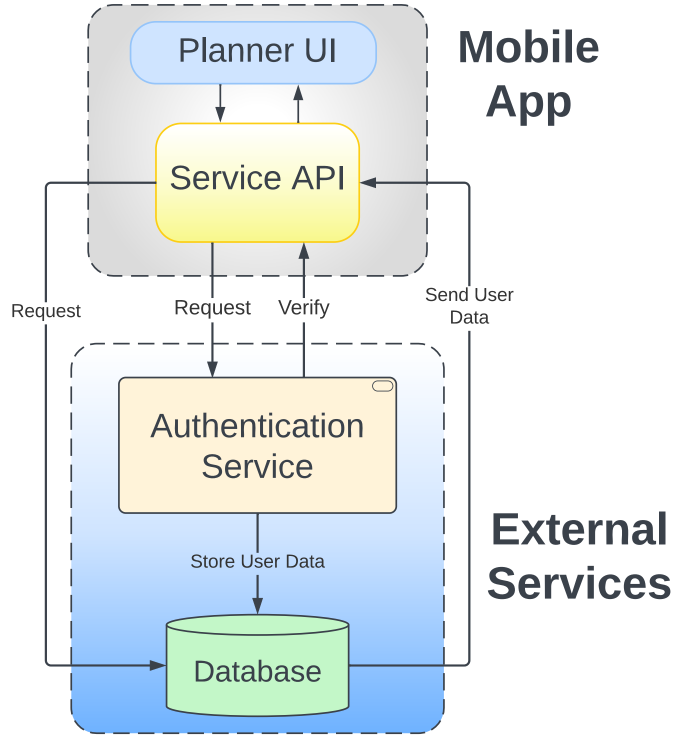

Fig. 2: Architecture of SPA External Components

D. External Architecture

Figure 2 illustrates the interaction between the external components in the SPA's solution, providing an overview of the architecture. Automating task creation and scheduling is essential in minimising the barriers to using SPA. Authenticating students is necessary to automate these processes and store and retrieve data for their enrolled courses. Significantly, this authentication step will help ensure the student's privacy regarding enrollment and completion data when using SPA. Only the credentials, enrolled course data, task completion data, and challenge data are tracked within the authentication and database services to minimise student data collection. The students must authenticate themselves using university credentials provided by the institution (student IDs and student emails). The SPA web API sends requests containing the user's student credentials to an authentication service for verification. Successful verification will allow the SPA to access the student's enrolled courses through a database. The database schema will map the student ID to a list of enrolled courses. Enrolled courses then map to the tasks, tests and associated deadlines needed to complete the course. The SPA's web API receives responses containing course information from the database and parses them so the SPA's planner user interface can visualise them. This process requires no additional actions from the student, assuming the correct credentials are inputted. After entering their credentials, and once they are authenticated, they can instantly view their workloads, tasks, and challenges. There are cases where internet access may not be available to students. Even in these cases, SPA still needs to be usable to cater to the diverse work habits of students. Some form of local storage as a backup will ensure that the changes can be stored locally and that SPA is still usable without an internet connection.

E. Sustainability Considerations

The SPA primarily promotes the social aspects of sustainability goals outlined by the UN. SPA's design aligns with the "Quality Education" and "Reduce inequality" goals [18]. By aiding students in managing their academic workload, SPA could improve academic outcomes, decrease academic stress, build practical time management skills and contribute to higher-quality education, which is crucial in improving equality [39]. However, it is important to note that SPA primarily focuses on supporting students within tertiary education, which is more self-reliant. In terms of providing quality education, developing nations struggle with secondary school completion rates [40]. Furthermore, SPA is a mobile application that relies on access to modern smartphone hardware to function, whose ownership is growing in developing countries but is significantly lower than in developed economies [41]. When considering these prominent factors contributing to lower-quality education, the impact of SPA in its current iteration is limited as it primarily provides support for students in tertiary education, potentially leaving gaps in usefulness for those in secondary education and accessibility for those who lack the necessary technology. SPA's economic and technological sustainability is ensured by utilizing actively maintained open-source libraries, which reduces application development costs and ensures that the implemented libraries are actively maintained. This consideration extends to the database and authentication solution for SPA. It is essential to avoid incurring significant operation or database hosting charges to prevent accumulating costs, which could significantly impact the long-term sustainability of SPA. These considerations will prevent potential financial barriers arising during further SPA development, such as renewing product licenses or charges for accessing the database.

V. Implementation

A. Model, Database and Authentication

Google's Firebase is used to implement the authentication service through Firebase Auth. Firebase is additionally used for database components through Firestore. Firebase Auth and Firestore are directly supported in Flutter through the FlutterFire API library. When authenticating a student through Firebase Auth, the standard process involves entering the student's email address (preferably given by the institution) and password as credentials. Firestore is a type of NoSQL database that offers the advantage of flexibility with regard to the data types that are stored. Unlike other SQL databases requiring pre-defined schemes, Firestore's NoSQL database allows for a broader range of data structures, such as JSON, XML, or other structured formats. While SPA currently only uses JSON for its database solution, NoSQL will enable the use of different types of data structures that may be required in future iterations, thereby increasing the flexibility of future development. The student data JSON uses the student's pre-defined user ID (UID) as a key to access their data. UIDs are generated upon account creation. Each student data value holds references to three important pieces of data:

- Their enrolled course codes are used to access course data.
- A challenges document stores information on each challenge the user has accepted or postponed.
- The activities progression for each course the student is enrolled in, including user-made activities.

Once student data is retrieved from Firestore, it is stored within SPA as a UserModel, which encapsulates all the currently logged-in student information needed for SPA to function. Flutter's provider library is used to access this model and handle the user interface state. The provider is used as it allows the model to be accessible from any part of the Flutter widget tree and is easier to maintain and implement than Flutter's default state management system. Course JSON data is a collection of assignments, tests and exams each with its own sub-tasks, start dates and due dates, simulating a course outline. For the purposes of this iteration of this project.

Using a single API library for Firebase's authentication and database services made implementing these services much more straightforward. If other external services were used, separate APIs may have been required to complete the same operations, which would have added significant complexity to the logic of the SPA's API. Using Firebase may raise concerns regarding the cost of operations, which, according to Google's billing information, can become expensive at large scales [42]. However, this is not an issue for SPA since it is only a prototype. It is improbable that the app's usage will ever reach such a high scale.

B. Task Planner

Figure 3 represents the planner screen that gives students an overview of the workload over a calendar month. The colouration of days determines the cumulative workload across each of their tasks that are active during that day. Syncfusion's Flutter Calendar library is used as a base for this calendar, as it provides considerable flexibility within its calendar framework. Within the calendar. the more saturated red colourisation indicates a higher workload for that period. A combination of two metrics determines this colouration. Firstly, the individual task health is calculated, which is factored into the completion stage of the task and the time remaining before the task is due, outlined in algorithm 1.

Algorithm 1 Calculate Task Health

```
function CALCULATETASKHEALTH
    {Calculate the total number of subtasks and the number of completed subtasks}
    totalSubtasks ← count(subtasks)
    completedSubtasks ← count(Completed subtasks)
    {Calculate the completion factor}
    completionFactor ← completedSubtasks/totalSubtasks
    {Calculate the total time and time passed for the task}

    totalTime ← difference(taskDueDate, taskStartDate)
    timePassed ← difference(todaysDate, taskStartDate)
    {Calculate the time factor}
    timeFactor ← timePassed/totalTime
    {Calculate the health of the task based on the completion and time factors}
    if completionFactor == 1.0 then
        health ← 1.0
    else
        health ← 1.0 + completionFactor − timeFactor
    end if
    {Set the health field in the Activity object}
    setHealth(health)
end function =0
```

The task health metrics (Algorithm 1) for the active tasks during each calendar day are accumulated. In other words, a task is considered active on a particular day if its start time (when the task is released) is on or before that day and its due date is after or on the same day. These accumulated tasks are then used to find the average daily health value to determine the colouration of individual cells using the following and are implemented in algorithm 2. This metric allows SPA to determine the overall workload for a given day. A penalty is applied for each simultaneous task that is active during a particular day, which helps to represent busy periods caused by many assignments or tests that students need to manage.

When using these health metrics, the visualisation gives users an overview of periods of increased workloads and allows them to better prepare and plan their time for these periods. This visualisation also motivates users to start work early to minimise the impact of these high workload periods. Selecting a day on the planner will display a dialogue containing a list of drop-down tiles for each task active during that day

Algorithm 2 Calculate Daily Average Health

```
function DailyAverageHealth
    {Calculate the average health for each day}
    dailyAverageHealth ← empty map
    for each (date, healthValues) in dailyHealthValues do
        averageHealth ← sum(healthValues)/length(healthValue

        {Apply a penalty for concurrent activities}
        penalty ← ConcurrentActivityPenalty
        averageHealth ← averageHealth × penalty
        dailyAverageHealth[date] ← averageHealth
    end for
    return dailyAverageHealth
end function =0
```

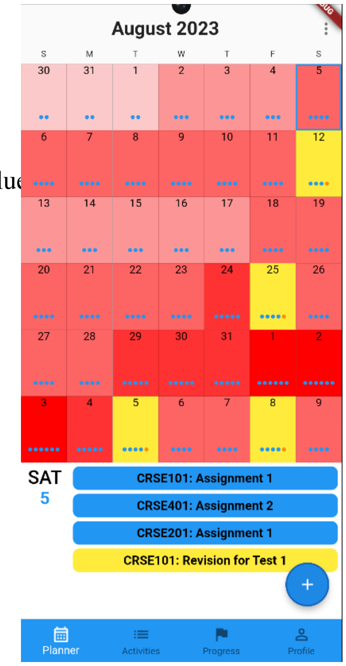

Fig. 3: Planner Heat Map

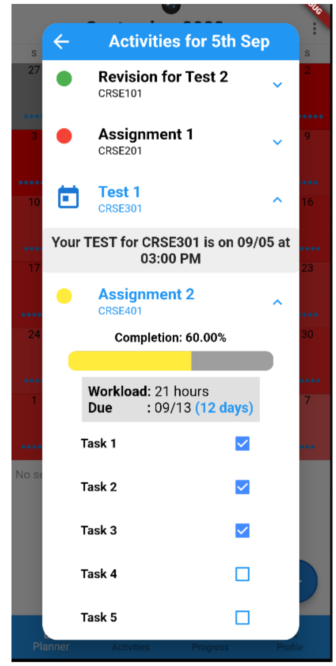

Fig. 4: Planner Day Activities

(Figure 4. A Flutter dialogue window is used here rather than navigating to a separate screen to prevent potential latency between screens. The task list displays the next sub-task to complete, the overall progress, and a button to bring up further details of the task. Coloured circles on each communicate the health of the task, determined using the health metric described in algorithm 1. Tasks that are marked green indicate that good progress has been made on them, with health values greater than 0.75. Tasks that require more work are marked yellow, with health values between 0.75 and 0.5. Finally, tasks that have inadequate progress, need attention or are due soon are marked red, with health values less than 0.5. Tasks that have passed their due date are coloured grey. Selecting any of these activities within the list displays a drop-down of three key features.

- A progress bar indicating how much of the task is completed, which is coloured based on the calculated health value. This communicates not only the completion stage of the task but also if the completion stage is satisfactory.
- A description of the estimated remaining workload and how long until the task is due, which, combined with the progress bar, assists in informing the user of how much work remains for the selected task.
- A checklist of sub-tasks that make up the task. Completing each sub-task completes the overall task. By checking the sub-tasks, the health value for the task and the daily average health values are updated to reflect the new completion status. This feature enables users to update tasks and view the impact of completing them on both the task health and the workload projection in the planner. Each checkbox selection also triggers the Firestore API to update the student's data to reflect the new information. As there is potential for many successive presses of the sub-tasks, debouncing is used to prevent excessive calls to the Firestore API and instead only send changes once one second has passed since the last sub-task update. Doing this is necessary to prevent latency issues within SPA or excessive read and write operations in Firestore, which could increase the cost of operation.

Additionally, users may add their own activities to the calendar through the activity creator (figure 5, which lets them set the details and dates for their activity. Their custom-made events are added to the calendar and do not have their own health values but contribute to the cumulative number of tasks during that day. In the current iteration of SPA, these user-created activities cannot be modified when created.

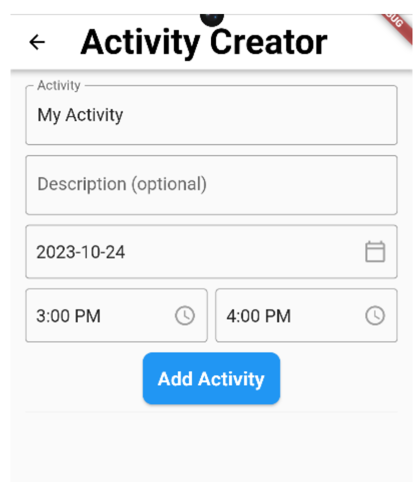

Fig. 5: Activity Creator

C. Activity List Screen

Figure 7 represents the activity list screen, providing students with an alternative method of planning tasks. Within this view, tasks can be sorted by priority based on health values, with the lowest values appearing at the top of the list. The tasks on the top of the list are likely to be the highest priority to commence work on to prevent delays or failure in task completion before the task deadline. This prioritized listing aids students in selecting which tasks to allocate their time to by directing their focus towards tasks that are more important given the time context. Furthermore, selecting a task (figure 6) provides the same task view as seen in Figure 4, allowing students to make updates to their task completion from either of these screens.

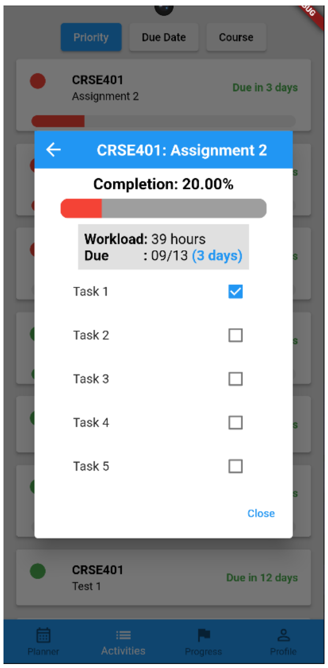

Fig. 6: Activity List Screen

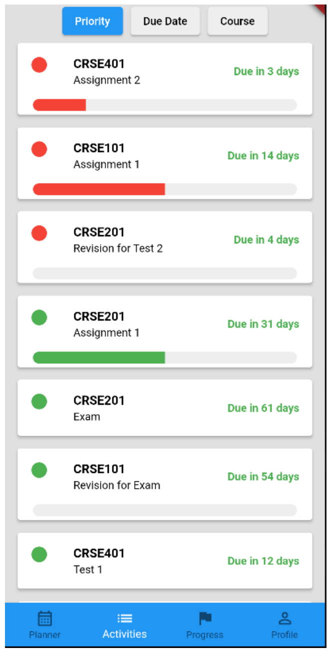

Fig. 7: Activity List Screen Sorted by Priority

D. Challenge Popups

Challenge offers are implemented as popups using Flutter's dialogue feature (figure 8) and focus on challenging students to undertake work on a task that is of low health. Challenge tasks are also selected based on tasks due by the end of the week or revising for upcoming tests. Challenge tasks are assigned based on a weighted system between the three criteria. The weighting of tasks due soon is calculated using the inverse of the number of days until the task is due, represented by equation 1. The same calculation is used to determine the weighting of the forthcoming tests and exams. The weighting of the low-health tasks are calculated according to equation 3. Each of the weights is normalised based on the sum of the weights. Each weight is then normalised by dividing each number by the sum of each weight to get the proportion of each weight between 0 and 1 (equation 4. The final selected challenge task is based on randomly generating a random number R between values 0 and 1, then selecting the task based on the generated outlined in equation 5. Choosing a challenge based on these weights allows a chance for each challenge type to be selected whilst prioritising tasks with very low heath or tasks and tests that are due very soon.

$$W_{\text{due}} = \frac{1}{\text{Days Until Due}}, \quad (1)$$

$$W_{\text{test}} = \frac{1}{\text{Days Until Test}}, \quad (2)$$

$$W_{\text{health}} = 1 - \text{Task Health} \quad (3)$$

$$W_{\text{total}} = W_{\text{test}} + W_{\text{due}} + W_{\text{health}} \quad (4)$$

$$\text{Selected} = \begin{cases} \text{Next test,} & \text{if } R < W_{\text{test}} \\ \text{Due next,} & \text{if } R < W_{\text{test}} + W_{\text{due}} \\ \text{Lowest health,} & \text{otherwise} \end{cases} \quad (5)$$

The weights must pass a threshold for a challenge to be created to prevent excessive challenges offerings. Furthermore, a student may have a maximum of three active challenges at once, and multiple Challenges are not assigned to a single task. Upon the first offer, challenges may not be declined, only postponed. This has been implemented to ensure that students can offset challenges for a short period but not eliminate them, as they are based on tasks that should have more work put into them. Postponing challenges sets a postponed Boolean flag to true, which will be offered again in two days. Challenge offers for previously postponed challenges have the option to be declined and will not be offered again to prevent potential irritation from unwanted repeated challenge offers.

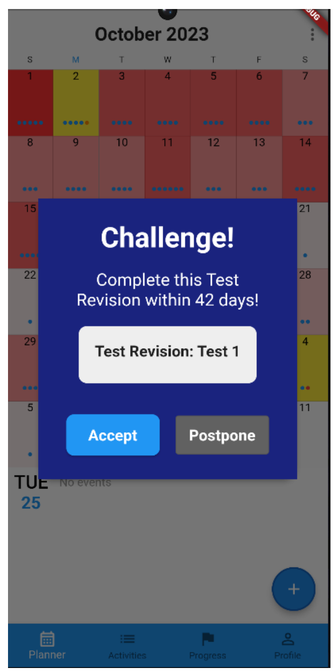

Fig. 8: Challenge Offer

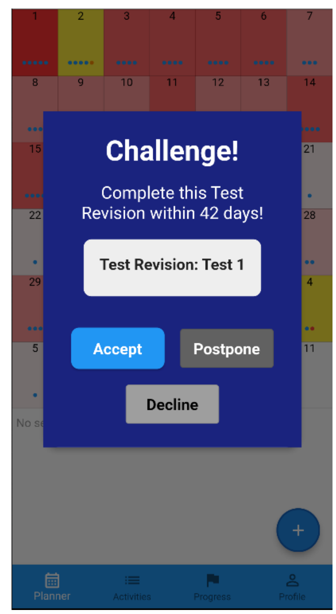

Fig. 9: Postponed Challenge Re-offer

E. Progression Screen

The progression screen visualises how well the student keeps up with their workload (Figure 10). Student progress is represented as a car in a 'race', with Non-Player Character (NPC) cars competing in the same race. One NPC is marked as a 'goal', which is ahead of the player and encourages students to improve their time management to catch up to the goal NPC. The average health value for each student's currently active tasks, normalised to be between 0 (poor health for all functions) and 1 (perfect health for all tasks), is used to place their car onto the progress bar. This screen provides the student with an intuitive and engaging visualisation of their course

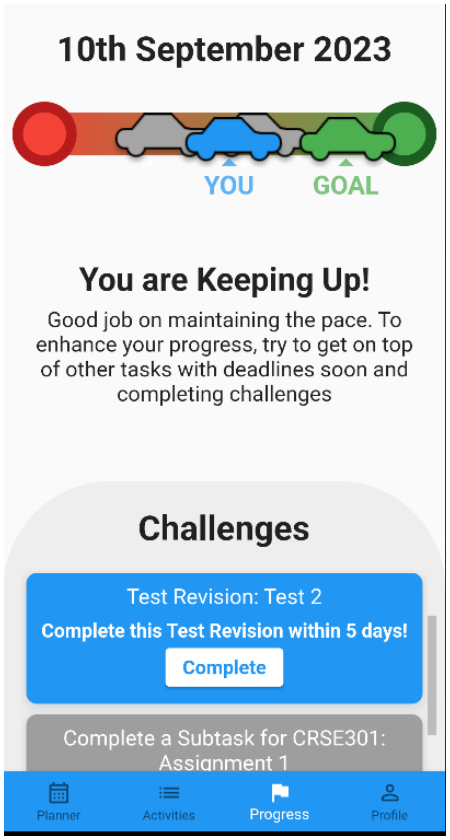

Fig. 10: Progress Screen

progress.
Finally, challenges are listed within the progress screen, within a scrollable view. Completing challenges is accomplished simply by selecting the 'complete' button to remove them from the challenge list. Additionally, postponed challenges are displayed and are coloured in grey with no 'complete' button, indicating their postponed status.

VI. Evaluation

The evaluation of SPA aims to measure SPA's usability and effectiveness in enhancing academic productivity and time management skills. This section highlights SPA's positive aspects and limitations in its current iteration, which can be used as focus points for future development. User testing was conducted to evaluate SPA usability and effectiveness. User testing consisted of quantitative data using nine 5-point Likert-scale questions. These questions focused on the usability of SPA, its ease of use, how well the visualisation aided in identifying busy periods, and whether users thought the app could improve their time management and planning. The Likert scale ranged from "Strongly Disagree" to "Strongly Agree". Table I provides a comprehensive list of the Likert-scale questions. Additionally, qualitative data was gathered through open-answer questions in which participants provided feedback in whatever words they chose. These questions asked for feedback on the positive and negative aspects of using SPA, providing insights from user experiences that could point out potential usability issues and enhance the quantitative data findings. In particular, the open-answer questions asked users to highlight the positive and negative aspects of SPA. A copy of the user testing sheet, with all questions included, is provided in the appendix. Testing was conducted on four participants, all fourth-year students at the Victoria University of Wellington. Each participant gave explicit consent by signing a consent form.
The results of the Likert scale questions are outlined in figures 11 and 12. Likewise, the results of the open-answer questions are outlined in tables II and III. Based on the participants' responses to the open-answer questions, common themes were identified that highlight both the positive aspects of the user's experience with SPA and areas that require improvement. In addition to these themes, we have included specific quotes from participants related to these themes. This analysis provides valuable insights into how the user experience of SPA can be improved.

The results obtained from user testing show very positive responses towards SPA. According to the Likert-scale question results, no neutral, disagree or strongly disagree responses were given for any of the nine Likert-scale questions, indicating high satisfaction across each evaluated aspect. Users found it easy to start using SPA and reported that its user interface is responsive and intuitive. Users affirmed that SPA's comprehensive functionality and visualisations could aid their task organisation. Additionally, users believed that SPA's functionality could support their time management, planning and motivation, indicating SPA's potential to enhance productivity. The positive aspects found in open-answer question results corroborate these positive responses from the Likert scale questions. Users responded positively to the time-management features, especially the activity list screen and the calendar workload visualisation. Users also highlighted that the challenge popups were an effective motivator and appreciated the time saved from the automated calendar setup. However, it is important to iterate that this project does not directly interact with any university infrastructure; Course data has been created explicitly mocked. While this lays the groundwork, it limits the capability of this project to represent real-world university data, which could likely differ significantly depending on the institution's systems and infrastructure.

The responses to the open-answer questions also highlighted additional areas of improvement for SPA and its potential limitations (Table III. The visual design of the workload visualisation is an area for improvement, particularly its chosen red-gradient colour scheme. Users noted that the red colouration could be anxiety-inducing and counterproductive for some. Furthermore, users noted that the colour gradient could make distinguishing low to medium workloads difficult. Another limitation users indicate is the inability to update user-created activities after creation. Some critical pieces of feedback were given for the challenge feature also. Firstly, the current challenge list within the progression screen only shows one or two challenges at a time. Users suggested extending this view to show more accepted or postponed

challenges simultaneously. Additionally, push notifications were suggested as an extension of the challenge system, as it is currently limited to offering challenges only when opening the app.
The evaluation of the SPA is limited by its evaluated cohort. Each participant was a fourth-year university student who had likely already had ample chance to develop their time management skills and routines. A future testing opportunity is to test its usability on first-year students, who may be struggling more with time management issues.

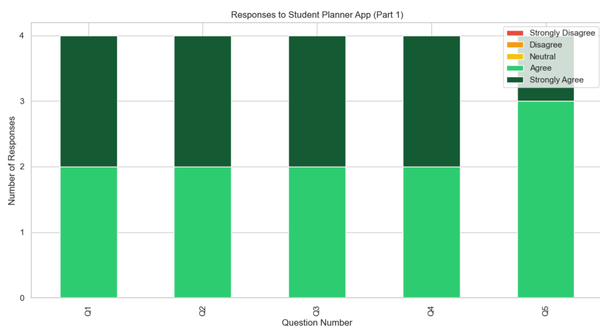

Fig. 11: Responses for Questions 1-5

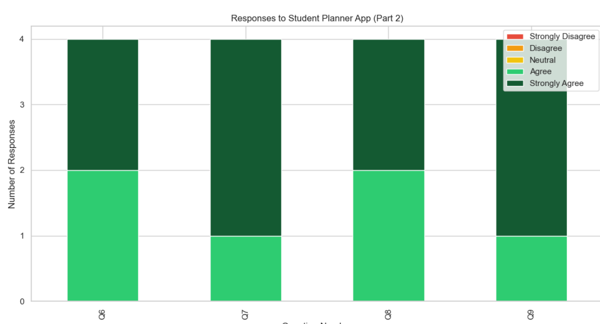

Fig. 12: Responses for Questions 6-9

VII. Conclusions and Future Work

Students, especially those transitioning to university, face significant challenges in time management. Existing time management tools often require time-consuming setups, steep learning curves, limited task scheduling and workload visualisation functionality, making existing solutions less suited to busy university schedules. SPA aimed to address these usability issues present in current solutions whilst providing personalised task prioritisation and workload visualisation tailored to aid university students in time management and planning. The final implementation of SPA showed positive results in user testing, with users expressing high satisfaction in all areas of its evaluated aspects. Users found SPA's interface intuitive and responsive and appreciated its functionality for aiding task organisation, time management, planning, and motivation. Reception to the activity list screen, calendar workload visualisation, and challenge popups were positive. User feedback also highlighted areas for improvement for SPA and potential avenues for future development. The red-gradient colour scheme of the workload visualisation was noted as potentially anxiety-inducing or unclear for distinguishing workloads. This may be improved by using an alternative colour scheme or separate colour thresholds for workloads, such as green for low, yellow for medium, and red for high workloads. Users also suggested enhancing the challenge by introducing push notifications outside of the app.

Further avenues for future development include introducing more gamification aspects into SPA for better user engagement. One such example is extending the progression screen race, which is simplistic in this iteration. Adding items, unlocks, or level progression to the progression screen could enhance user engagement with the app and motivate them to be proactive in their studies. Alternatively, adaptive companions, such as that seen in other learning apps such as Duolingo, have been shown to enhance enjoyment and engagement from the user [27]. An adaptive companion within SPA could provide the user motivation, encouragement, or task-planning advice, enhancing the overall engagement and satisfaction of using the app.

References

[1] K.-L. Krause and H. Coates, "Students' engagement in first-year university," *Assessment & Evaluation in Higher Education*, vol. 33, no. 5, pp. 493–505, 2008.
[2] T. Hailikari, N. Katajavuori, and H. Asikainen, "Understanding procrastination: A case of a study skills course," *Social Psychology of Education*, vol. 24, no. 2, pp. 589–606, 2021.
[3] S. G. Nayak, "Impact of procrastination and time-management on academic stress among undergraduate nursing students: A cross sectional study.," *International Journal of Caring Sciences*, vol. 12, no. 3, 2019.
[4] R. V. Adams and E. Blair, "Impact of time management behaviors on undergraduate engineering students' performance," *Sage Open*, vol. 9, no. 1, p. 2158244018824506, 2019.
[5] J.-E. Kim and D. A. Nembhard, "The impact of procrastination on engineering students' academic performance," *International Journal of Engineering Education*, vol. 35, no. 4, p. 1008, 2019.
[6] I. S. Jones and D. C. Blankenship, "Year two: Effect of procrastination on academic performance of undergraduate online students.," *Research in Higher Education Journal*, vol. 39, 2021.
[7] J. Van der Meer, E. Jansen, and M. Torenbeek, "'it's almost a mindset that teachers need to change': first-year students' need to be inducted into time management," *Studies in Higher Education*, vol. 35, no. 7, pp. 777–791, 2010.
[8] A. Häfner, A. Stock, L. Pinneker, and S. Ströhle, "Stress prevention through a time management training intervention: An experimental study," *Educational Psychology*, vol. 34, no. 3, pp. 403–416, 2014.
[9] A. Häfner, A. Stock, and V. Oberst, "Decreasing students' stress through time management training: An intervention study," *European journal of psychology of education*, vol. 30, pp. 81–94, 2015.
[10] H. E. Douglas, M. Bore, and D. Munro, "Coping with university education: The relationships of time management behaviour and work engagement with the five factor model aspects," *Learning and individual Differences*, vol. 45, pp. 268–274, 2016.
[11] M. de Raadt and S. Dekeyser, "A simple time-management tool for students' online learning activities," in *Proceedings ASCILITE 2009: 26th Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education: Same Places, Different Spaces*, 2009.
[12] M. K. K. Singh, N. A. Samah, *et al.*, "Impact of smartphone: A review on positive and negative effects on students," *Asian Social Science*, vol. 14, no. 11, pp. 83–89, 2018.
[13] V. M. Rambitan, "The effect of smartphone on students' critical thinking skill in relation to the concept of biodiversity," *American Journal of Educational Research*, vol. 3, no. 2, pp. 243–249, 2015.

Label	Question
Q1	It was easy to start using the planner
Q2	The planner's user interface is intuitive
Q3	The planner offers all the features I expect
Q4	The planner helps to organize and plan my tasks
Q5	I believe the planner would motivate me to complete my scheduled tasks
Q6	I think the planner will positively impact my productivity and time management
Q7	I could update my in-app tasks quickly
Q8	The planner provided a smooth user experience
Q9	I would recommend the Student Planner App to others

TABLE I: Likert Scale Evaluation of SPA

Aspect	Quotes
Task Management Features	"Activity screen can show assignments in the order of both due date and priority." "The activity screen has good UI and an intuitive layout." "In the Activity tab, it is easy to shift priority."
Workload Visualization	"The calendar very clearly shows the workload." "Liked the due dates of assessments in yellow." "The colour-coding obviously shows "panic days"."
Integration with University Course Outline	"Liked that assessment dates and workload are pre-determined based on the Course outline."
Challenges	"The challenges are a good reminder and motivator to work on assessments."
Overall User Experience	"Clear layout of assessment progress." "Consistency of design throughout the app."

TABLE II: Positive Aspects of SPA

Aspect	Quotes
Visual Design of Heat Map	"I personally found the red colour gradient to indicate the active workload quite overwhelming at first." "The gradient of the colours may become confusing." ""
Notification System	"Push notifications for challenges would be useful."
User Interface for Challenges	"The challenge list screen could be changed so that the challenges are not in a small window at the bottom." "Only the bottom part of the challenges section is scrollable. The user may want to see more challenges at once."
Task Modification Features	"Add the ability to modify added external tasks (e.g., change the date of an appointment)."

TABLE III: Areas of Improvement for SPA

[14] A. S. Drigas and P. Angelidakis, "Mobile applications within education: An overview of application paradigms in specific categories.," *International Journal of Interactive Mobile Technologies*, vol. 11, no. 4, 2017.

[15] M. K. Foti and J. Mendez, "Mobile learning: How students use mobile devices to support learning," *Journal of Literacy and Technology*, vol. 15, no. 3, pp. 58–78, 2014.

[16] J. Broadbent, E. Panadero, and M. Fuller-Tyszkiewicz, "Effects of mobile-app learning diaries vs online training on specific self-regulated learning components," *Educational Technology Research and Development*, vol. 68, pp. 2351–2372, 2020.

[17] J. T. Boruff and D. Storie, "Mobile devices in medicine: a survey of how medical students, residents, and faculty use smartphones and other mobile devices to find information," *Journal of the Medical Library Association: JMLA*, vol. 102, no. 1, p. 22, 2014.

[18] "The 17 goals." , https://sdgs.un.org/goals, United Nations. , last accessed on 30/05/2023.

[19] Q. Shambour, S. Fraihat, *et al.*, "The implementation of mobile technologies in higher education: a mobile application for university course advising," *Journal of internet technology*, vol. 19, no. 5, pp. 1327–1337, 2018.

[20] C. Alario-Hoyos, I. M. Estévez Ayres, M. P. San Agustin, D. A. Leony Arreaga, and C. Delgado Kloos, "Mylearningmentor: A mobile app to support learners participating in moocs," 2015.

[21] A. Liyanage, S. Siriwardana, S. Reyal, and M. Mithsara, "Design-thinking and ucd combination for designing effective time management assistant mobile app.," in *RoCHI*, pp. 111–118, 2021.

[22] A. Liyanage, W. D. Jayarathne, S. Siriwardana, S. Reyal, *et al.*, "Scheduleme-smart digital personal assistant for automatic priority based task scheduling and time management," in *2021 2nd Global Conference for Advancement in Technology (GCAT)*, pp. 1–6, IEEE, 2021.

[23] A. N. Saleem, N. M. Noori, and F. Ozdamli, "Gamification applications in e-learning: A literature review," *Technology, Knowledge and Learning*, vol. 27, no. 1, pp. 139–159, 2022.

[24] E. Pechenkina, D. Laurence, G. Oates, D. Eldridge, and D. Hunter, "Using a gamified mobile app to increase student engagement, retention and academic achievement," *International Journal of Educational Technology in Higher Education*, vol. 14, pp. 1–12, 2017.

[25] F. F.-H. Nah, Q. Zeng, V. R. Telaprolu, A. P. Ayyappa, and B. Eschenbrenner, "Gamification of education: a review of literature," in *HCI in Business: First International Conference, HCIB 2014, Held as Part of HCI International 2014, Heraklion, Crete, Greece, June 22-27, 2014. Proceedings 1*, pp. 401–409, Springer, 2014.

[26] T. Cassells, B. Daire'O, and K. Power, "The effect of gamification on time-management in tertiary education," in *European Conference on Games Based Learning*, p. 881, Academic Conferences International Limited, 2016.

[27] T. Cassells and D. O'Broin, "The difference in intrinsic motivation when completing a prioritization task in a standard and gamified interface.," in *GamiFIN*, pp. 167–178, 2019.

[28] S. Diefenbach and A. Müssig, "Counterproductive effects of gamification: An analysis on the example of the gamified task manager habitica," *International Journal of Human-Computer Studies*, vol. 127, pp. 190–210, 2019.

[29] T. Vilček and T. Jakopec, "Comparative analysis of tools for development of native and hybrid mobile applications," in *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pp. 1516–1521, IEEE, 2017.

[30] M. Gonsalves, "Evaluating the mobile development frameworks apache cordova and flutter and their impact on the development process and application characteristics," 2019.

[31] R. Enihe and J. Joshua, "Hybrid mobile application development: A better alternative to native," *Global Scientific Journal*, vol. 8, no. 5, 2020.

[32] S. Stender and H. Åkesson, "Cross-platform framework comparison: Flutter & react native," 2020.

[33] W. Wu, "React native vs flutter, cross-platforms mobile application frameworks," 2018.

[34] A. E. Fentaw, “Cross platform mobile application development: a comparison study of react native vs flutter,” 2020.
[35] R. Kesavan, D. Gay, D. Thevessen, J. Shah, and C. Mohan, “Firestore: The nosql serverless database for the application developer,” 2023.
[36] T. McKenzie, M. Morales-Trujillo, S. Lukosch, and S. Hoermann, “Is agile not agile enough? a study on how agile is applied and misapplied in the video game development industry,” in *2021 IEEE/ACM Joint 15th International Conference on Software and System Processes (ICSSP) and 16th ACM/IEEE International Conference on Global Software Engineering (ICGSE)*, pp. 94–105, IEEE, 2021.
[37] M. de León-Sigg, B. J. Pérez-Valenzuela, S. Vázquez-Reyes, and J. L. V. Cisneros, “Adaptation of the initial software development method for a single developer,” in *2018 6th International Conference in Software Engineering Research and Innovation (CONISOFT)*, pp. 35–41, IEEE, 2018.
[38] T. Thesing, C. Feldmann, and M. Burchardt, “Agile versus waterfall project management: decision model for selecting the appropriate approach to a project,” *Procedia Computer Science*, vol. 181, pp. 746–756, 2021.
[39] J. Walker, C. Pearce, K. Boe, and M. Lawson, “The power of education to fight inequality-oxfam,” *THE POWER OF EDUCATION TO FIGHT INEQUALITY: How increasing educational equality and quality is crucial to fighting economic and gender inequality. Retrieved December*, vol. 6, p. 2021, 2019.
[40] U. Nations, “The sustainable development goals report 2023,” 2023.
[41] P. R. Center, “Smartphone ownership on the rise in emerging economies,” 2018. Accessed: 2023-10-10.
[42] Google, “See a cloud firestore pricing example,” 2023. Accessed: 2023-10-11.