StudyMon

NUS Orbital 2025
Final Submission

Chan Jun Jie, Year 1 Business Analytics
Gabriel Simundo, Year 1 Computer Science

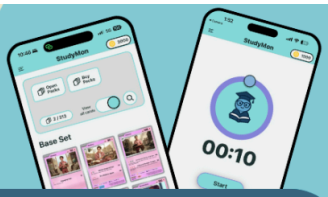
Contents

Posters

Liftoff Poster: 7236.png
Milestone 1 Poster: Poster.pdf
Milestone 2 Poster: 7236.png
Milestone 3 Poster: StudyMon.pdf

Final Submission Poster

StudyMon



About

Members

- Chan Jun Jie
- Gabriel Simundo Ponce

StudyMon is a study productivity app which encourages students to stay focused on their planned tasks and study sessions, rewarding them for their efforts.

Problem Statement

We recognize how difficult it can be to study continuously without frequent distractions from our mobile phones. It can be tempting to take a quick look at our devices while studying, breaking our focus and decreasing our productivity. We want to provide a solution to mitigate this issue, motivating us to study more frequently, and with better focus.

SWE PRACTICES

- **Component Based Architecture**
- **Agile Methodology** to develop our app over 2 week sprints.
- **Version Control with Git**
- **Branching and Pull Requests**
- **Unit Testing**

Tech Stack


React Native


Expo


Firebase


Figma


Github


Assets generated by ChatGPT and Gemini

Features

1. **User Account Authentication**
2. **Studying Motivation Timer**
3. **Card Collection**
4. **Exchange SM coins at the Store for Card Packs**
5. **Pack Opening to obtain cards**
6. **Schedule Interface**
7. **Algorithmic Scheduling for Tasks and Events**
8. **Friend System to view others stats and collection**
9. **Profile page for users to review their statistics and studying habits.**

Future Plans

- Trading Cards System between friends
- A playable card game using StudyMon cards
- Achievements system

TrY Us

Download Expo Go and Scan the QR code

Proof of Concept

Refer to video demonstration:

Final Submission Video - Made with Clipchamp_1755149521998.mp4

User testing link:

1. Download **Expo Go** from the App Store or Google Play
2. Scan the QR code below

3. Log in with
   - email: email@email.com
   - password: password

(or create your own account with any email!)

Test Accounts

Email	Password	Remarks
email@email.com	password	Developer Account
test1@email.com	password	Developer Account

Proposed Level of Achievement

Apollo 11

Project Scope

StudyMon aims to build a gamified productivity mobile app for NUS students that combines a focus timer, intelligent task and event scheduling, and a collectible trading card reward system to promote distraction-free studying.

Aim

We aim to develop a study productivity app which encourages students to stay focused on their planned tasks and study sessions, rewarding them for their study efforts. By implementing a focus timer, intelligent task scheduling algorithm, and a trading card system, we can provide an engaging way for students to manage their time effectively and stay motivated. The app can create a sense of achievement and community through this system, while enhancing users' productivity.

Motivation

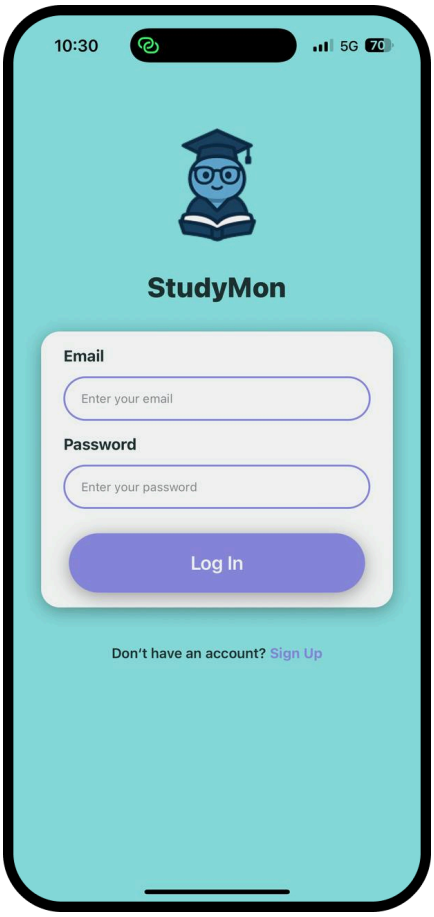
As NUS students, we recognise how difficult it can be to study continuously without frequent distraction from our phones. It can be tempting to take a quick look at our devices while studying, breaking our focus and decreasing our productivity. Thus, we want to provide a solution to mitigate this issue, even motivating us to study more frequently, and with more focus.

User Stories

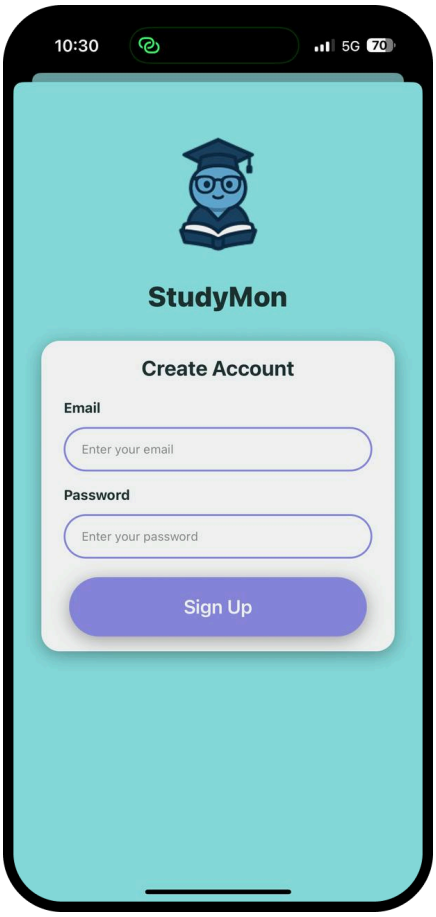
1. As a student who wants to achieve better study habits, I want to be able to study without being distracted by my phone.
2. As a student using the study timer, I want to earn in-game currency upon completion of each study session to reward me for the effort.
3. As a student using the app, I want to be able to exchange this currency for StudyMon card packs to motivate me to do more study sessions.
4. As a student with a hectic schedule, I want a personal scheduling tool to view all my tasks and events in one convenient place.
5. As a student with different tasks with different importances, I want a tool to help me schedule them efficiently to ensure I fulfil all my deadlines.
6. As a NUS student, I want to be able to sync my NUSMods timetable, and insert all my lessons into my schedule.
7. As a student, I want to see a summary of my study timer usage to ensure that I am spending my time productively
8. As a student using the app, I want to add my friends and stay updated with their study sessions, further motivating me to study more.
9. As a user with duplicate or wanted cards, I want to be able to trade these cards with my friends and the general public, to get the ones I want.
10. As a student using the app, I want to unlock achievements after hitting studying milestones, receiving bonus rewards to motivate my continued usage of the app.

Features

1. User Account Authentication [Completed]



Log-In Screen



Sign-up Screen

Description

Each user will be authenticated with his/her own unique account. Utilising the Firebase SDK Authentication, users can log in via Email and Password. They will be able to create an account/sign up if the user does not have an existing account.

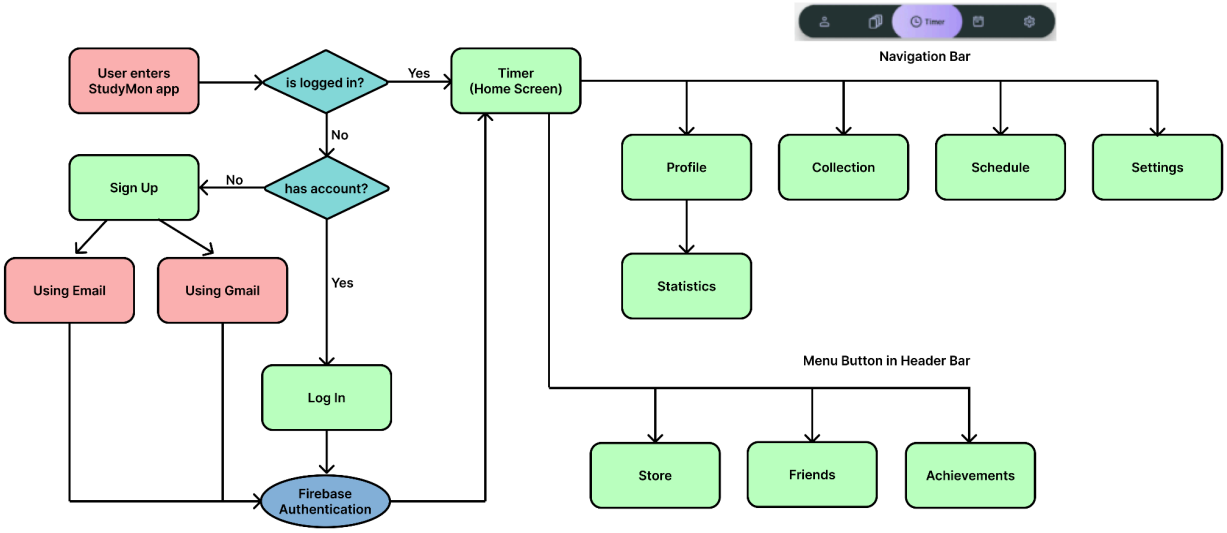
Based on the user's action, the relevant call to the Firebase authentication instance is made. For instance, registering with email and password calls **createUserWithEmailAndPassword**. The FirebaseAuthExceptions are handled within a try-catch block and an alert bar is displayed when the user keys in an invalid input.

Only upon successful login or account registration will the user be routed to the next page and the account reflected in the Firebase authentication.

Implementation philosophy

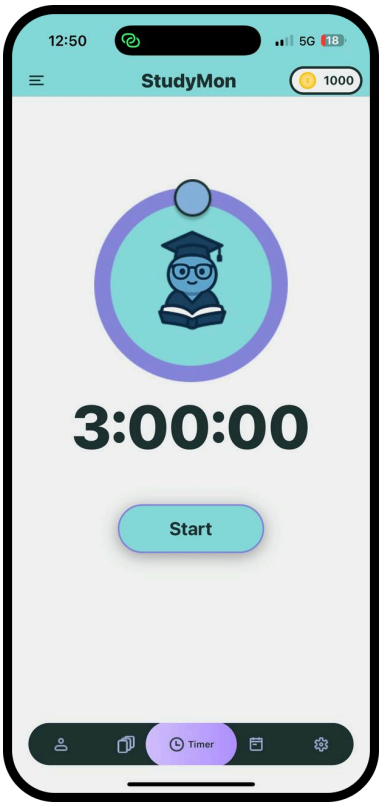
The authentication controller includes a listener that continuously monitors the user's sign-in status. When the mobile app is launched, it first checks whether the user is logged in.

If the user is already signed in, they are immediately directed to the Home page.
If not, the system performs an additional check to see if a user profile exists for the account. When such a profile is found, the user is taken directly to the Timer home page. Otherwise, they are routed to the Login page.

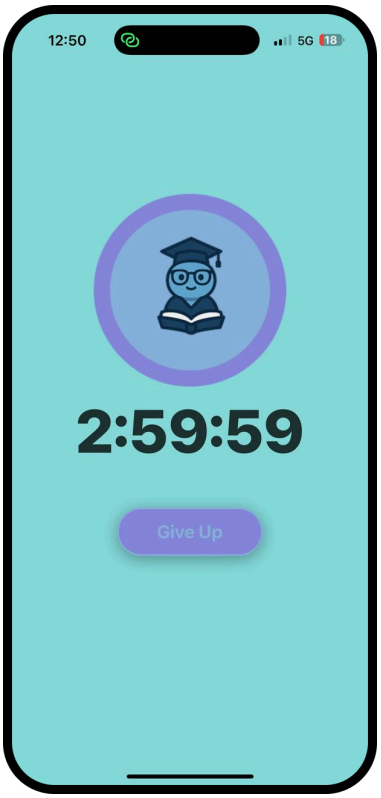


User Authentication Diagram

2. Studying Motivation Timer [Completed]



Timer Screen (Home Screen)

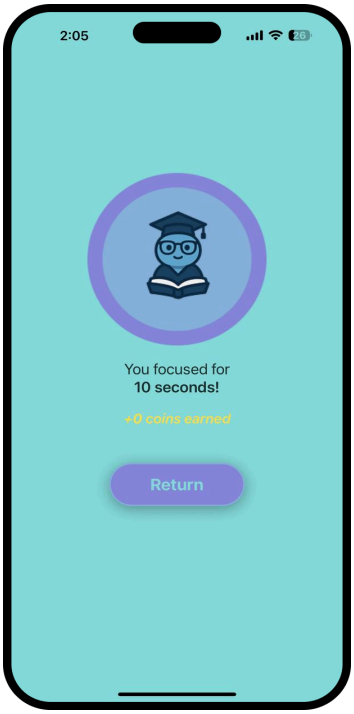


Active Timer Screen

Description

The Study Motivation Timer is a core feature designed to help students stay focused during study sessions by limiting distractions and rewarding consistency. Users can set a timer for a specific study duration, during which they are not allowed to leave the StudyMon app or use other apps, as any attempt to do so results in lost progress and no rewards.

Upon successful completion of the timer, users earn in-game currency that can be used to purchase card packs from the app's collectible trading card system. This mechanic not only encourages discipline and commitment but also gamifies the study experience to make productivity more engaging and rewarding.



Completed Timer Screen

Rewards

Upon completion of a study session, StudyMon Coins are given to the user depending on how long the timer was set to, according to the following table. The rate of SM coins awarded is different for the first, second and third hour **within the same timer session**.

Session Duration	SM Coins awarded
1 to 60 minutes	1 SM coin every 5 minutes
61 to 120 minutes	1 SM coin every 4 minutes
121 to 180 minutes	1 SM coin every 3 minutes

For example, a user who completed a 1h 30min study session with the timer will receive 12 SM coins for the first hour and 7 SM coins for the next 30 minutes for a total reward of 19 SM coins for the session. This system encourages users to commit to longer study sessions, thus improving focus and efficiency of study sessions.

However, to prevent misuse of the timer to gain large amounts of SM coins while one is not actually studying, such as when users are sleeping, we set the maximum timer duration to 3 hours. This also encourages users to take breaks in between study sessions which can improve productivity.

3. Trading Card Collection [Completed]

StudyMon's trading cards serve as our main reward system to incentivise users to continue using our app and stay motivated. While using our app, users will receive in-game currency (StudyMon coins) which can be exchanged at our Shop for StudyMon card packs.

Each pack can be opened to build a collection of NUS-themed trading cards of various rarities. These cards are added to the user's personal collection, providing a fun and rewarding system that encourages long-term engagement and goal completion through gamification.

These cards form a theoretically playable mobile trading card game (TCG) with a set of rules heavily inspired by the Pokemon TCG. We incorporate aspects of popular trading card games such as the Pokemon over-the-board paper TCG and the Hearthstone mobile game while designing our cards.

While we did not plan for the mobile trading card game to actually be playable, due to massive effort to create a fully playable mobile card game being equivalent to an entire Orbital project in itself, we thought it would be more meaningful for users building up their collection of StudyMon cards if the cards were to be well thought out and theoretically playable in a fully planned out game. In the future, we may extend our app to include a fully playable mobile TCG using the StudyMon cards collected by users.

Card Design

We designed each card individually using Figma, making use of Figma's Component system to create templates of various styles of card,

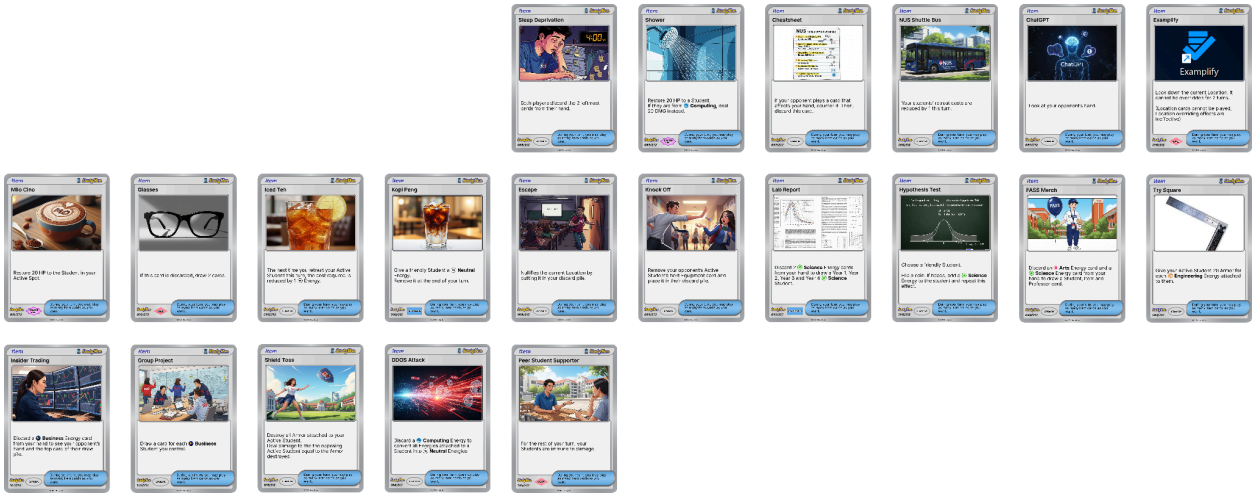
Student Card Templates	Item Card Template	Equipment Card Template	Professor Card Template	Location Card Template

This fulfills the Don't Repeat Yourself principle and allows us to easily make changes in format and arrangement of elements to the template card which carry forward the changes to all instances.

Card List

For StudyMon's release, we plan to have a base set of 150 cards comprising 84 Student cards, as well as 21 Item cards, 11 Equipment cards, 17 Professor cards and 17 Location cards.

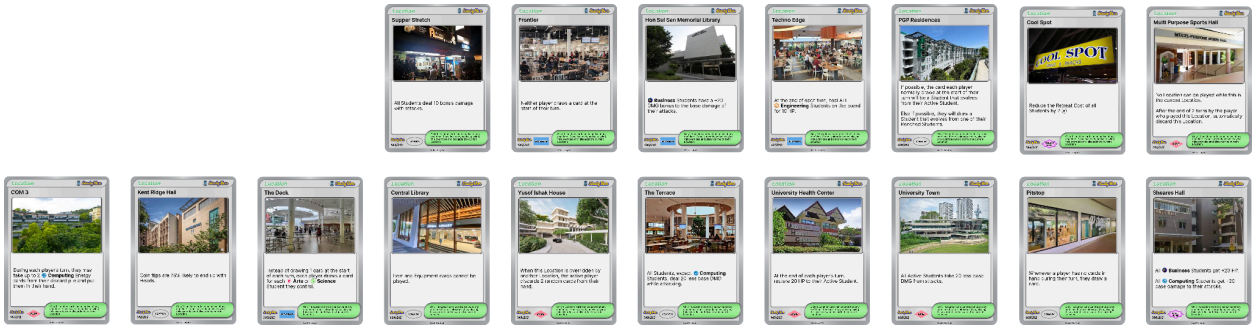
84 Student Cards (001 - 084)



Item Cards (085 - 105)

Equipment Cards (106 - 116)

Professor Cards (117 - 133)



Location Cards (134 - 150)

For future updates, we may add expansions to introduce cards to represent students from more courses, new Items and Equipment cards recognisable by NUS students, and other NUS locations not already included.

Generating Card Art

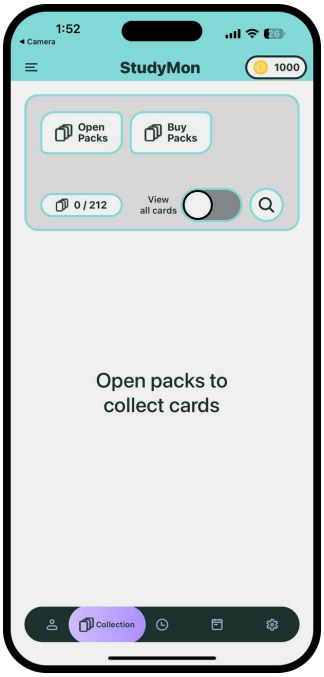
We use generative AI tools, namely ChatGPT and Gemini, to generate our card arts. Below is an example template prompt we use to generate most of our card arts.

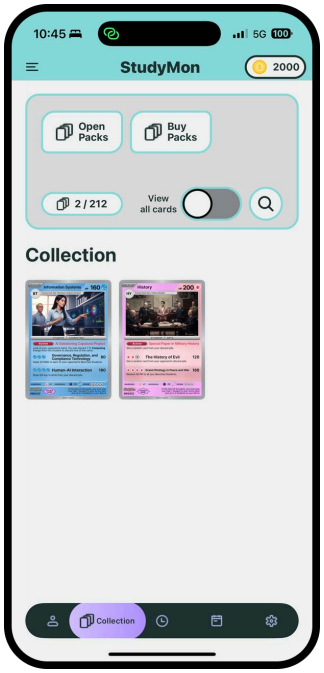
Prompt for Student card art

Generate 5 new pictures of a year [YEAR] NUS [male/female] [COURSE] student. Don't give me a generic student wearing a shirt that says NUS or [COURSE] on it, I don't want any title with the descriptions on the images. Give them nice clothes with an [COLOUR] colour theme. I want details that show they are a [FACULTY] student studying [COURSE]. Make them seem [DESCRIPTIVE DETAILS]. Give me a full picture image with NO weird white banners. I want the 5 pictures in digital art theme and rectangular format.

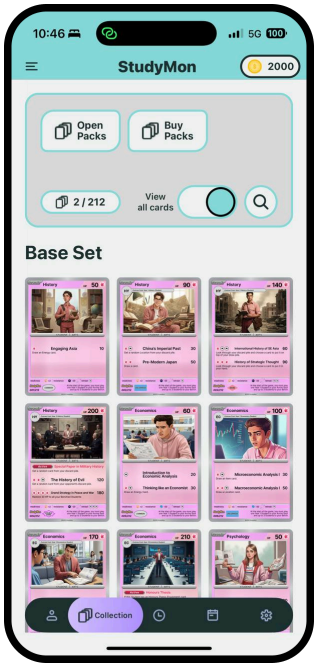
This will usually produce 0-3 decent card arts per prompt. We readjust and repeat these prompts until we get enough card arts for a line of Student cards for a particular course, to represent students from Year 1 to Year 4.

Collection Screen



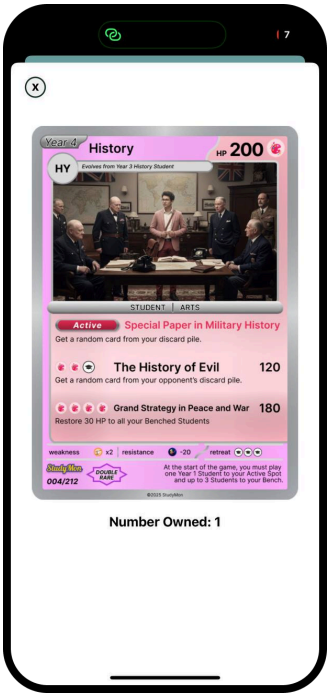


View Owned Cards Only



View All Cards

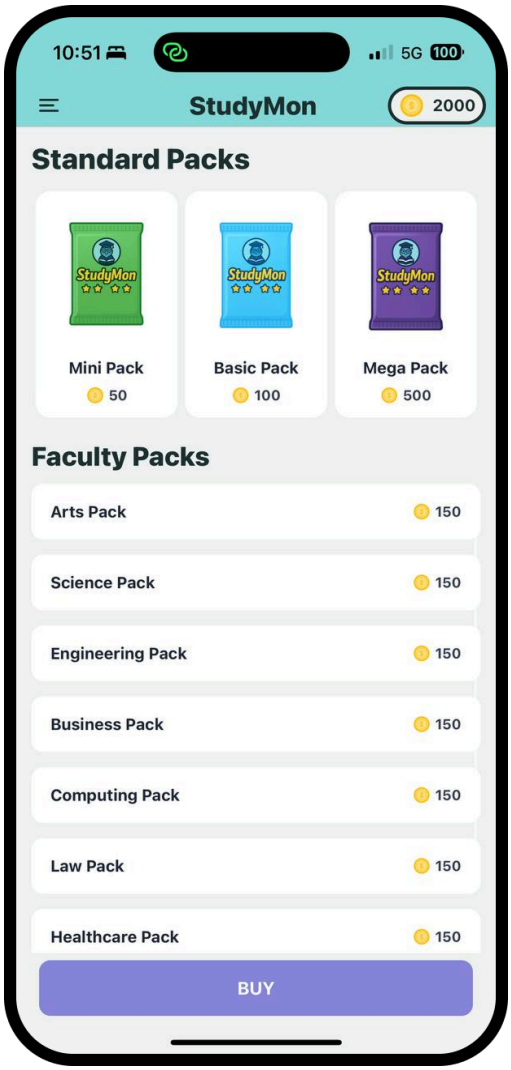
From this screen, users can view all the StudyMon cards they have collected from opening packs. By clicking each card, they will see a focused view on each card to read the card text more easily.



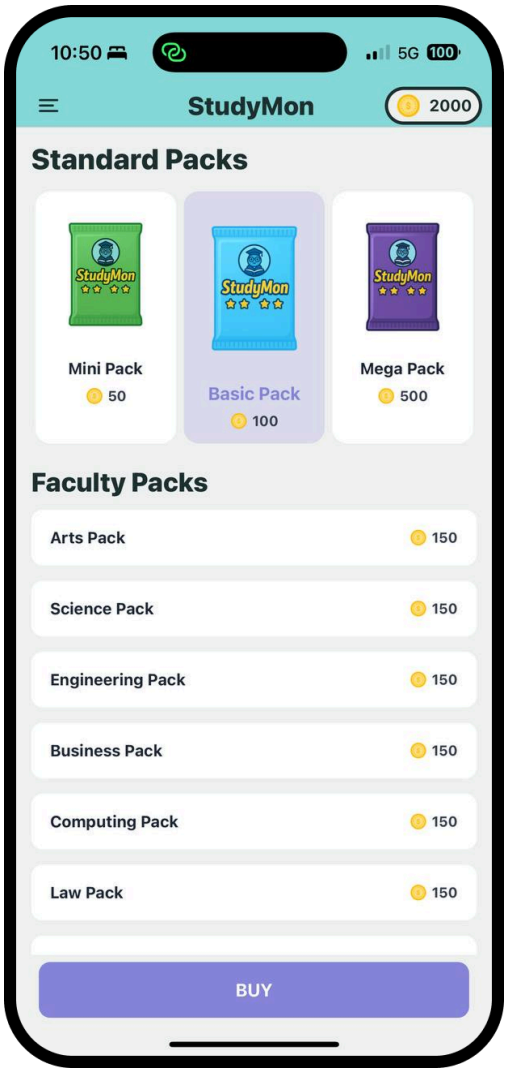
Focused View

4. Exchange StudyMon coins at the Shop for Card Packs

Shop Screen



Shop Screen



Focused view on pack

Shop Catalogue

Currently we have one Base Set of 150 cards which will mainly be sold in 3 tiers of packs: Mini, Basic and Mega.

Pack Algorithm

Each pack contains 5 cards, with each card having an independent probability to be of a certain rarity. The table below illustrates how cards are determined each time a user opens a pack.

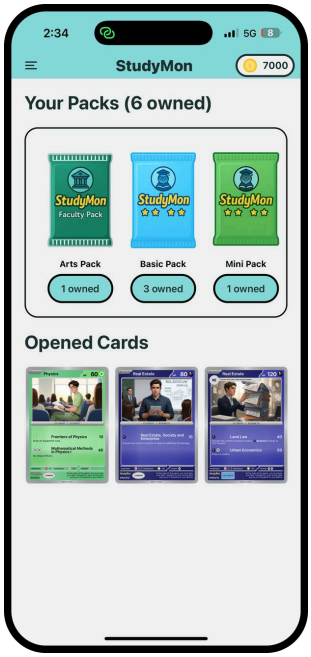
Algorithm Step 1: Generate the rarity of each card

Rarity	Probability
Common	74%
Uncommon	20%
Rare	5%
Double Rare	1%

In addition, each pack has a guaranteed chance to contain at least a Rare or better. If the first step of the algorithm does not produce any Rare or better cards, one of the cards will be replaced by a guaranteed Rare or Double Rare according to the following table:

Algorithm Step 2: Guaranteed Rare or better

Rarity	Probability
Rare	90%
Double Rare	10%



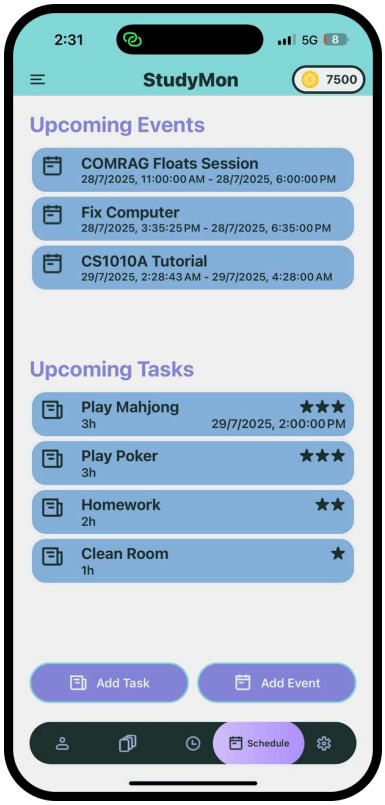
Card Opening Screen

5. Schedule Interface

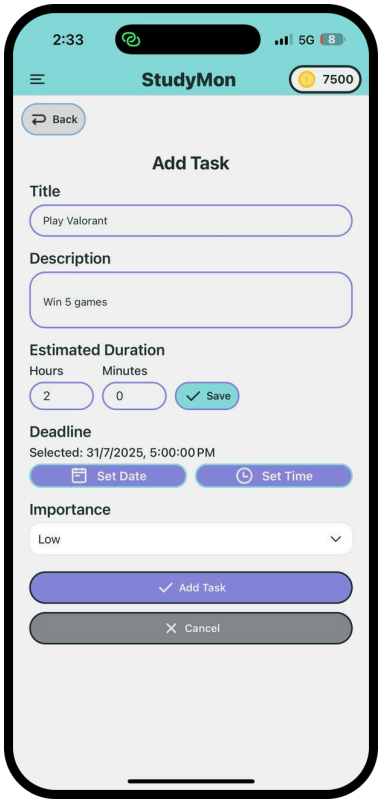
Description

This feature allows users to input their academic tasks and events, along with their deadlines, importance, and estimated durations.

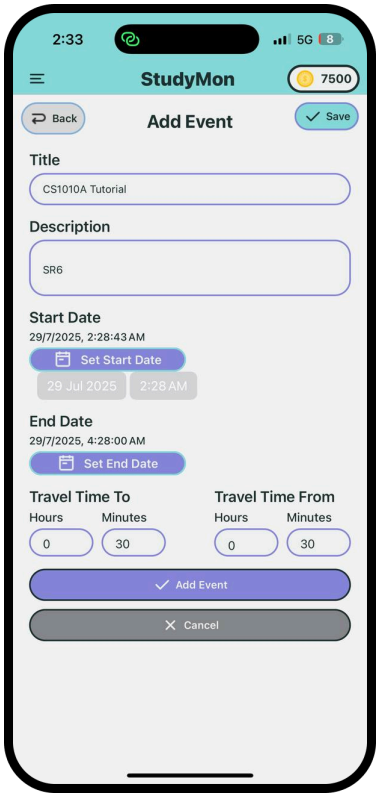
Tasks are objectives the user has to complete by a certain deadline, taking an estimated amount of time to complete as input by the user. Events are responsibilities the user has from a fixed start time to end time. StudyMon's algorithms intelligently schedules its users' tasks and events, taking into account travel time and free periods.



Schedule Screen



Add Task Screen

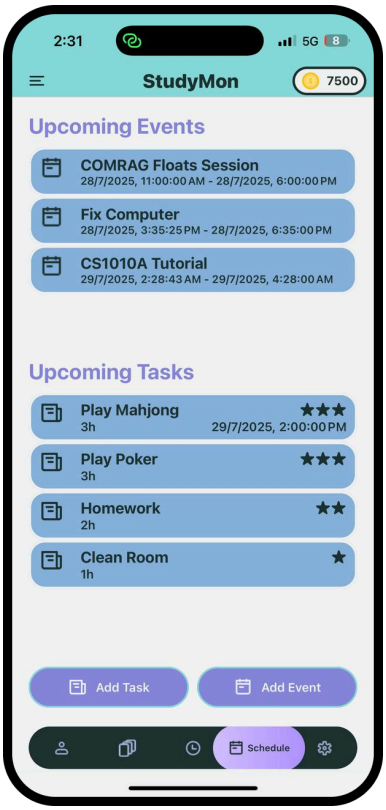


Add Event Screen

6. Algorithmic Scheduling for Tasks and Events

Description

As users complete tasks or events earlier or later than expected, the scheduling algorithm dynamically recalibrates to ensure all deadlines are still met in the most efficient way possible, helping students make the best use of their time.



Algorithmic Scheduling of Tasks

We keep track of each user's tasks and events in Firestore. Every user document has a tasks subcollection and events subcollection, each holding task and event documents respectively. The fields contained by each Task and Event document are listed below.

```typescript
interface Task {
  id: string;
  title: string;
  description: string;
  estimatedDurationMinutes: number;
  deadline?: Timestamp | null;
  importance: 'low' | 'medium' | 'high';
  completed: boolean;
}
```

```typescript
interface Event {
  id: string;
  title: string;
  description: string;
  startDate: Timestamp;
  endDate: Timestamp;
  createdAt: Timestamp;
  travelTimeTo: number; // in minutes
  travelTimeBack: number; // in minutes
  source: 'manual' | 'NUSMods';
}
```

Task and Event fields in Firestore Database

StudyMon's TaskSort Algorithm

Our algorithm primarily comprises of a function named **taskPriorityComparator** which takes 2 Tasks (task1 and task2) and returns -1 or 1 based on whether task1 or task2 is ranked more important.

taskPriorityComparator ranks tasks by performing checks in the specified order:

1. Prioritize tasks with deadlines that would be failed if not done immediately
2. Prioritize tasks with close deadlines depending on their importance level
3. Prioritize tasks by importance (High > Medium > Low)
4. Prioritize tasks with deadlines
5. Prioritize tasks by earlier deadlines
6. Else, if all other checks consider the 2 tasks equal, just return task1

Every time the Upcoming Tasks overview is updated, i.e. when a new task is added, or when tasks are edited, completed or deleted, StudyMon's TaskSort algorithm will run to arrange the tasks such that the user always sees their tasks presented in the optimal completion order.

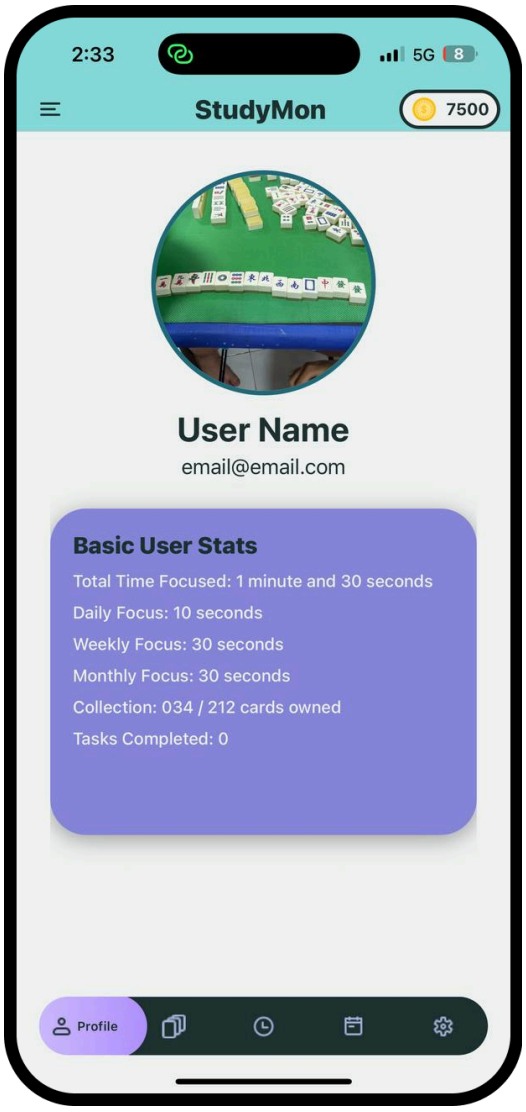
NUSMods API Integration

StudyMon can take in users' NUSMods Timetable to automatically populate their schedule with events representing their lessons.

7. Profile and Statistics Page

Description

The profile page displays personalized statistics such as total study time, monthly/weekly/daily focus time, completed tasks, and card collection progress. By visualizing their productivity stats, users can better understand their learning patterns, stay motivated, and set meaningful personal goals for improvement.

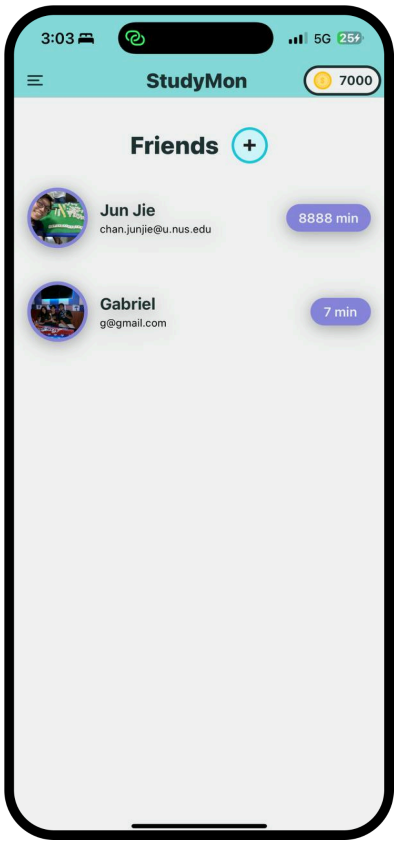


Profile and Statistics Page

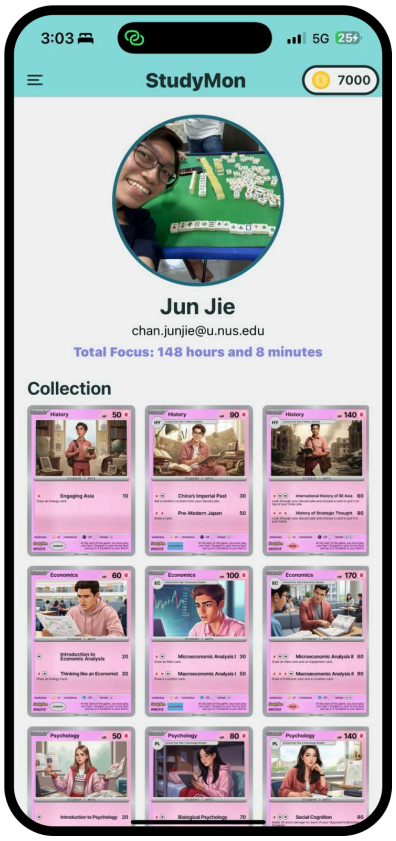
8. Friend System

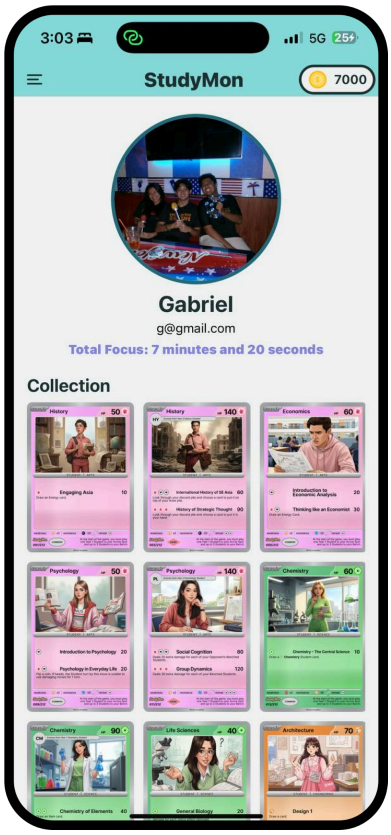
Description

Users can add friends within the app to build a sense of community and accountability. The friend system enables users to view each other's collections and total focus time, leading to a more interactive and enjoyable experience with StudyMon.

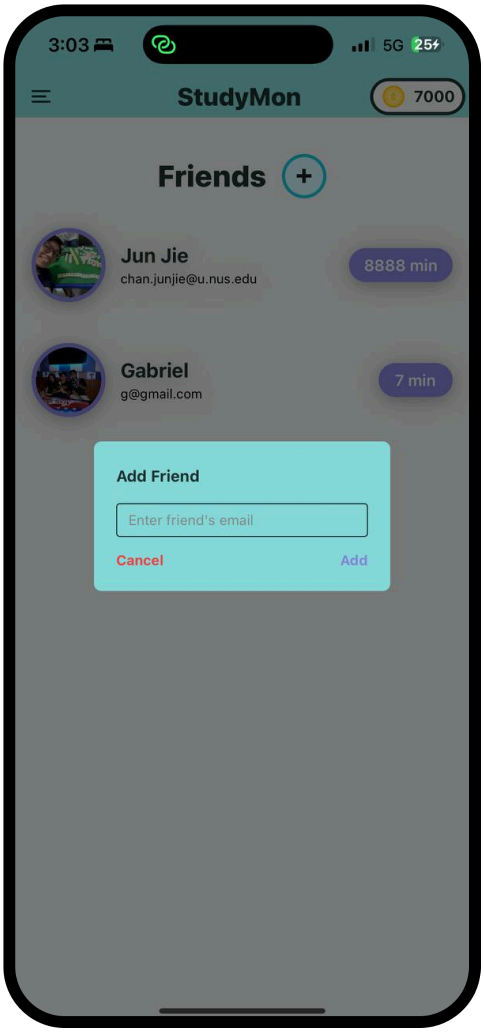


Friends Screen



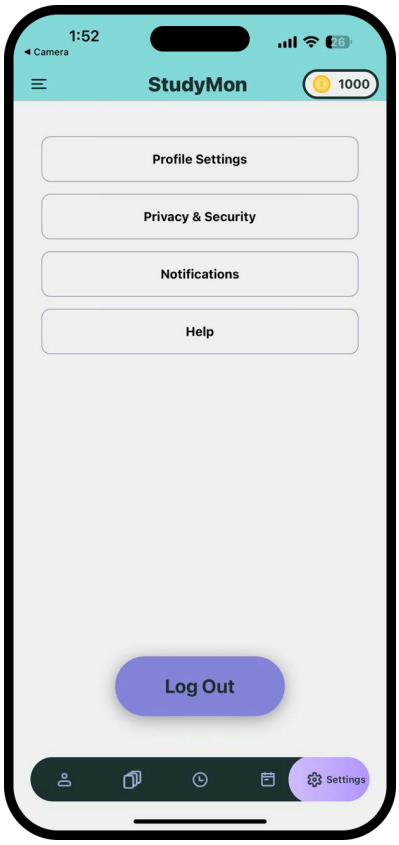


Detailed View on Friends


3:03
5G
StudyMon
7000
Friends
Jun Jie
chan.junjie@u.nus.edu
8888 min
Gabriel
g@gmail.com
7 min
Add Friend
Enter friend's email
Cancel
Add

Add Friends

9. Settings (To be done)



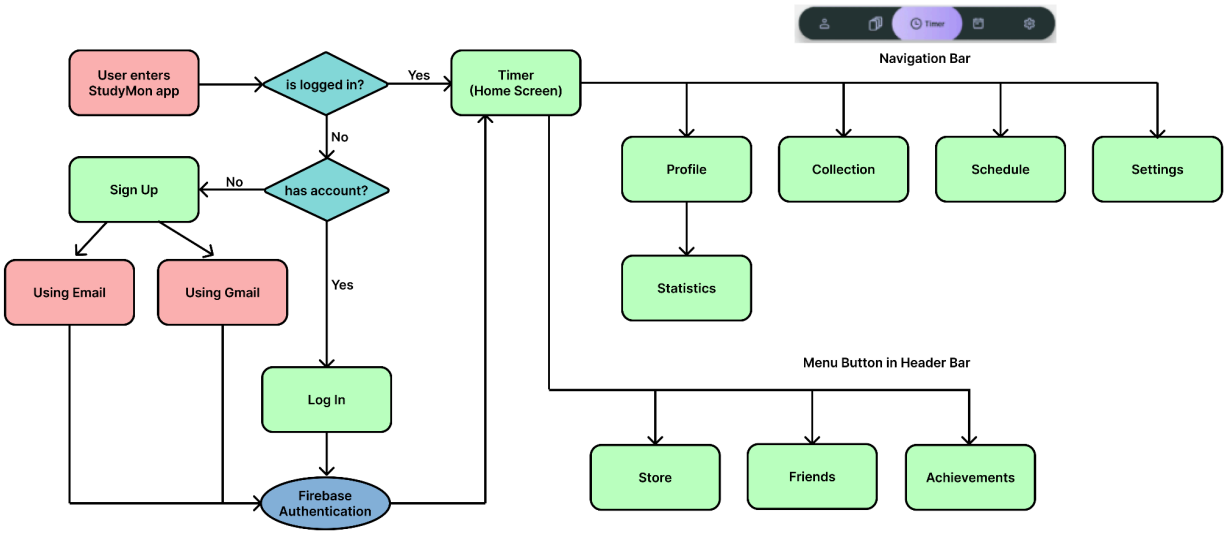
Settings Screen

Timeline

Below is a rough timeline and development plan with key goals and deadlines. Our Project Log provides more specific details.

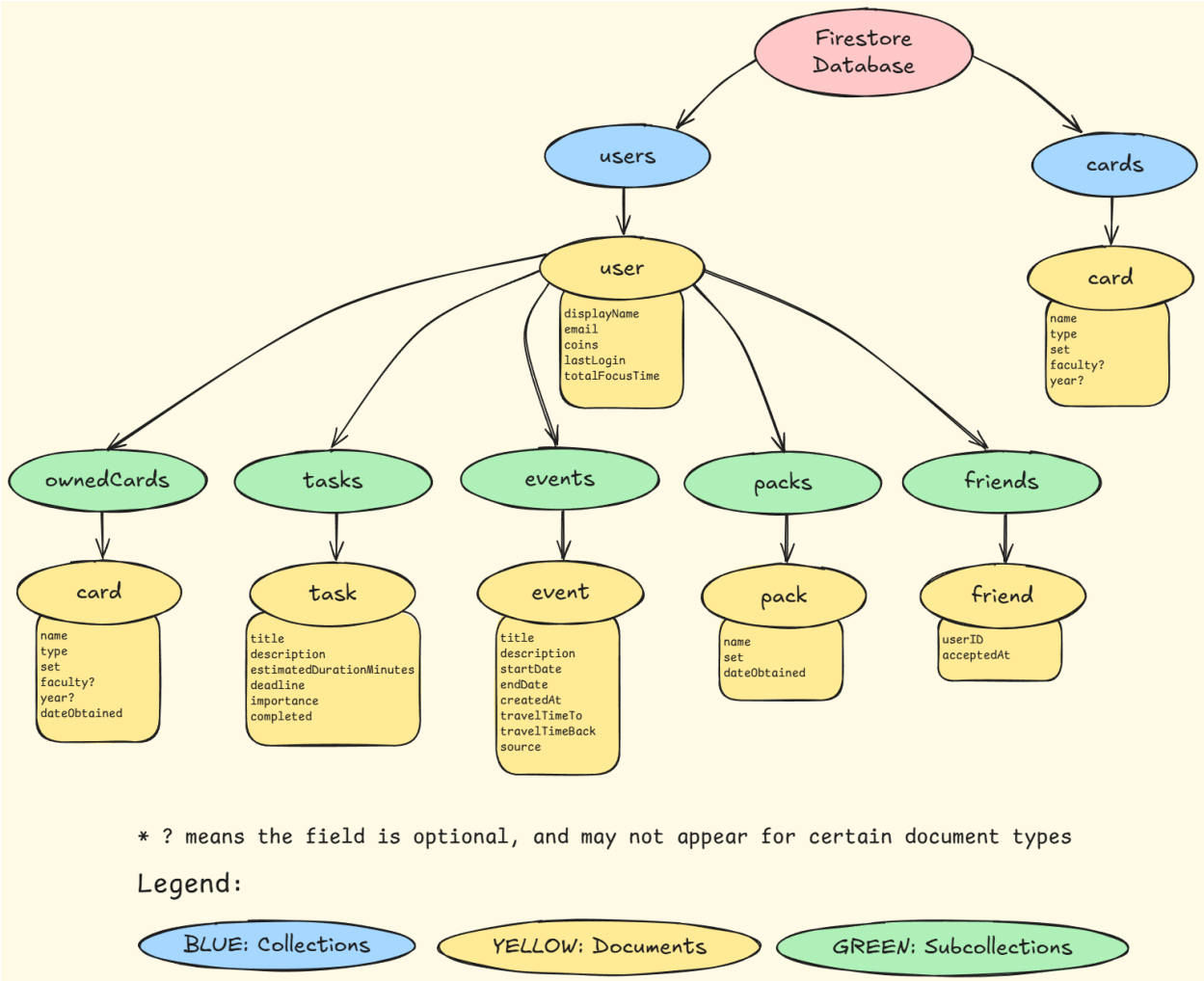
Sprint	Date	Objectives
0	12 May - 19 May (1 week)	• Solidify plans for the StudyMon app • Learn all the tools in our tech stack by following YouTube tutorials
19 May: Liftoff Submission		
1	19 May - 1 Jun	• Design the UI/UX for **User Authentication** and **Timer** screens on Figma • Implement **User Authentication** logic • Implement **Log In**, **Sign Up** and **Timer** screens
2 Jun: Milestone 1 Submission		
2	2 Jun - 15 Jun	• Plan and design **StudyMon cards** • Implement and design **Firestore** database • Design the UI/UX for **Shop**, **Collection**, **Profile** screens on Figma • Implement **Shop**, **Collection**, **Profile** screens
3	16 Jun - 29 Jun	• Design the UI/UX for **Schedule**, **Settings** and **Statistics** screens on Figma • Implement **Schedule**, **Settings** and **Statistics** screens
30 Jun: Milestone 2 Submission		
4	30 Jun - 13 Jul	• Design the UI/UX for **Trading**, **Friends** and **Achievements** screens on Figma • Implement **Trading**, **Friends** and **Achievements** screens
5	14 Jul - 27 Jul	•
28 Jul: Milestone 3 Submission		
6	28 Jul - 10 Aug	• Fix all bugs • Improve UI
7	11 Aug - 24 Aug	• Quality of life updates • Improve UI
27 Aug: Splashdown		

User Flow Diagram



Firestore Database Diagram

As Firestore is a NoSQL database, we created a Collection-Document Diagram to represent how StudyMon's database is designed.



For cards, which are represented as documents in the cards collection, we keep track of the name, category, set, faculty (only Student cards have faculty), year (only Student cards have faculty) and cardArt (a link to the card art).

Software Engineering Practices

Component Based Architecture

This paradigm allows us to break down complex UI features into smaller, reusable pieces. Each component is modular, making it easy to reuse and update components, fulfilling the Don't Repeat Yourself principle.

Agile Methodology and 2-week Sprints

We use the tried and tested Agile Methodology approach to develop our app over 2 week sprints. Before each sprint, we will meet up in person to evaluate our progress and discuss what we want to implement in the coming sprint. This will allow us to efficiently plan our targets, so as to meet the requirements of each Milestone.

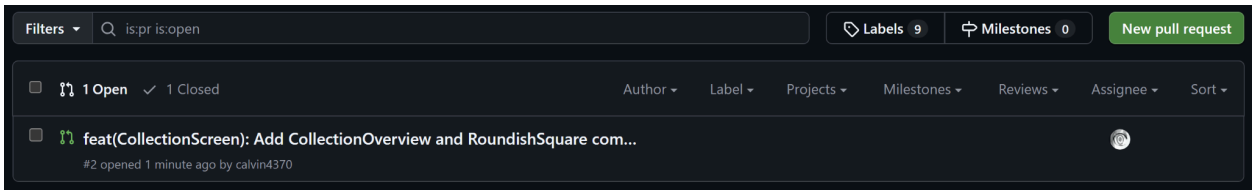
Secure Storage of Secret Keys

To prevent unauthorized access to our Firebase backend, we store all secrets such as our Firebase API key, Firebase Messaging Sender ID and Firebase App ID in a .env file. Our .gitignore includes this file, thus preventing our secret keys from being pushed to our github repository.

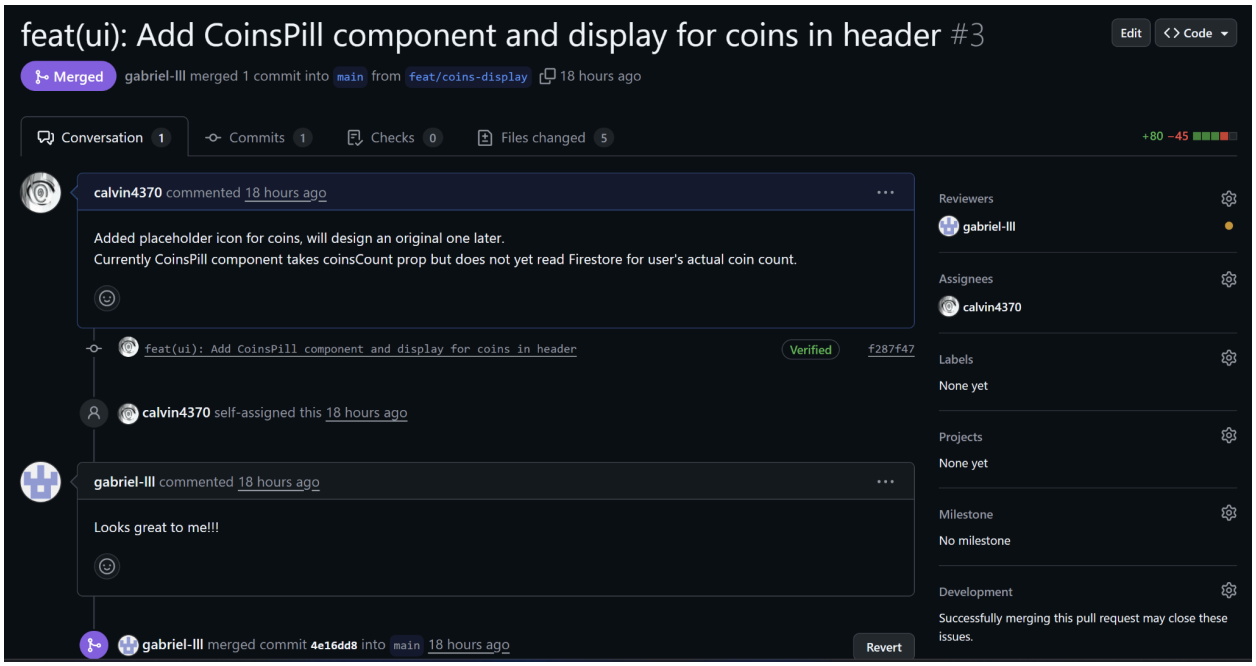
Version Control

Branching and Pull Requests

We use Git and Github for version control. Before every change to the codebase, we pull the latest code from the main branch, and start a new branch for each new feature. We then commit our changes to the feature branch and open a pull request for the other member to review and approve the proposed changes.



Only after thoroughly checking through the addition, will the reviewer approve the pull request and merge the new feature into the main branch. This ensures that both of us are caught up with each other's additions, as well as to properly resolve merge conflicts.



Testing

Unit Testing

Unit testing is a software testing method where we test individual pieces of code in isolation, such as functions and components, to make sure they behave as expected.

```
PASS app/__tests__/TimerComplete.test.tsx
  TimerComplete basic functionality
    ✓ formats the completed timer duration correctly (1 ms)
    ✓ reads duration from route params
    ✓ navigates to home when return is pressed

PASS constants/__tests__/helperFunctions.test.ts
  helperFunctions
    padWithZero
      ✓ adds leading zero to single-digit numbers
      ✓ does not add leading zero to double-digit numbers
      ✓ works with numbers greater than 99
    formatTime
      ✓ converts seconds to [hours, minutes, seconds] format
      ✓ handles large values correctly
    formatTimeAsSentence
      ✓ formats time with hours, minutes, and seconds (1 ms)
      ✓ formats time with seconds only
    formatTimeAsShort
      ✓ formats time with hours, minutes, and seconds
      ✓ formats time with seconds only
      ✓ handles zero correctly

PASS app/__tests__/TimerActive.test.tsx
  TimerActive handleGiveUp function
    ✓ handleGiveUp navigates to tabs and logs message

PASS components/__tests__/AuthGate.test.tsx
  AuthGate logic
    ✓ redirects to login screen when user is not authenticated (1 ms)
    ✓ does not redirect when user is authenticated
    ✓ does not redirect when authentication is loading
```

Some examples of basic automated unit testing we utilised to test specific components and pages of the program to ensure reliability.

User Testing

We made use of user testing to find bugs in our app, and to seek suggestions on how to improve UI elements and functionalities.

Telegram Message

Hi! We're Jun Jie and Gabriel, NUS students developing StudyMon 📚, a fun productivity app designed to help you stay focused and motivated while studying!

With StudyMon, you can:
⏱️ Set focus timers to keep you on track
📅 Organize tasks with our intelligent scheduler
🃏 Earn and collect StudyMon cards as rewards
🛍️ Buy card packs from our in-app shop
📈 Track your study stats and progress
👯 Add friends and check out their collection

We're currently in the testing phase and would love your help! 🎉

To try it out:
1. Download Expo Go from the App Store or Google Play
2. Scan the QR code below
3. Log in with
email: email@email.com
password: password
(or create your own account with any email!)

Once you're in, feel free to explore the app, try out all the features, and let us know:
🐛 Any bugs you spot
✨ Features you liked or think we should add
💬 General feedback or just a shoutout!

Share your thoughts here:
https://docs.google.com/forms/d/e/1FAIpQLScxNHO3apbTdZNse1oWNx5eW_iNsCTofVh0JSgff1pbIJoT8w/viewform?usp=dialog

Thanks for helping us make StudyMon better! Let's level up our study game together! 💪🎓

Tech Stack

- **Frontend**: React Native and Expo
- **Backend**: Firebase
- **Database**: Firestore (Firebase)
- **User Authentication**: Firebase SDK Authentication
- **UI Design**: Planned with Figma
- **Version Control**: Git and Github
- **CI/CD**: Github Actions

We use ChatGPT and Gemini to generate assets for our app, including the StudyMon logo, icons and card arts.

Project Log

Project Log

Activity Log

Sprint	Date	Name	Activity / Feature	Hours	Status
Liftoff	1 - 26 May 25	Jun Jie Gabriel	Learn React Native, Expo and how to create mobile apps by following various youtube tutorials	20	Done
Sprint 1	30 May 25	Jun Jie	Set up Expo project for React Native Frontend	1	Done
Sprint 1	31 May 25	Jun Jie	Setup Tailwind CSS and Nativewind	2	Done
Sprint 1	31 May 25	Jun Jie	Finish basic structure of the frontend, including tab bar with 5 screens, 2 of which are implemented while 3 are placeholders for now	5	Done
Sprint 1	1 Jun 25	Gabriel	Set up Firebase backend	2	Done
Sprint 1	1 Jun 25	Jun Jie	Update ProfileScreen and TimerScreen, and make Timer feature functional	6	Done
Sprint 1	1 Jun 25	Gabriel	Integrate auth users to log in and sign up screens	2	Done
Sprint 1	2 Jun 25	Jun Jie	Implement complete timer flow with custom input to adjust timing and completion screen	6	Done
Sprint 2	4 - 8 Jun 25	Jun Jie	Design templates for different StudyMon Student cards	10	Done
Sprint 2	8 Jun 25	Jun Jie	Design StudyMon card backs	2	Done
Sprint 2	9 Jun 25	Gabriel	Finish settings screen and added new pages	2	Done
Sprint 2	9 Jun 25	Gabriel	Update creating account and logging in features to be fully functional	2	Done
Sprint 2	8 - 10 Jun 25	Jun Jie	Design templates for different StudyMon Item cards	4	Done
Sprint 2	10 - 12 Jun 25	Jun Jie	Design templates for different StudyMon Professor, Equipment and Location cards	4	Done
Sprint 2	10 - 12 Jun 25	Jun Jie Gabriel	Learn how to use Firebase's Firestore database	4	Done
Sprint 2	11 Jun 25	Jun Jie	Create first 20 StudyMon Cards (20/212)	3	Done
Sprint 2	12 Jun 25	Jun Jie	Create 20 StudyMon Cards (40/212)	3	Done
Sprint 2	13 Jun 25	Jun Jie Gabriel	Design Firestore database	4	Done
Sprint 2	13 Jun 25	Gabriel	Add functional log out button	1	Done
Sprint 2	13 Jun 25	Jun Jie	Create 20 StudyMon Cards (60/212)	3	Done
Sprint 2	14 Jun 25	Gabriel	Implemented Auth Gate to only allow authorised users into the main app	3	Done
Sprint 2	14 Jun 25	Gabriel	Set up Firestore backend	4	Done
Sprint 2	14 Jun 25	Gabriel	Integrate Firestore database upon signing up	3	Done
Sprint 2	14 Jun 25	Jun Jie	Create 20 StudyMon Cards (80/212)	3	Done
Sprint 2	15 Jun 25	Jun Jie	Create 20 StudyMon Cards (100/212)	3	Done
Sprint 2	15 Jun 25	Jun Jie	Design Collection Page on Figma	5	In Progre...
Sprint 3	19 Jun 25	Jun Jie	Create 25 StudyMon Cards (125/212)	3	In Progre...
Sprint 3	20 Jun 25	Gabriel	Implement functional drawer into the app with additional pages	4	Done
Sprint 3	22 Jun 25	Gabriel	Finish design for drawer	2	Done
Sprint 3	25 Jun 25	Gabriel	Added functionality to timer linking it to rewards	2	Done
Sprint 3	28 Jun 25	Gabriel	Created and designed shop	5	Done
Sprint 3	30 Jun 25	Gabriel	Added basic schedule ui	3	Done
Sprint 3	30 Jun 25	Jun Jie	Implement Collection Screen	10	Done

Credits and References

- ChatGPT and Gemini were used to generate assets for our app, including the StudyMon logo, some icons and card arts.
- StudyMon card formats were inspired by that of Pokemon TCG cards.
- Some Icons were sourced from ionicons, Icons8 and Iconamoon
- Prettier is used to standardise code formatting