Sleep and Screen Time Tracker

Group 11, CM12005 Coursework 2

02/05/2024

Group Member	Username	Degree Course
Felix Nacu	fn361	BSc Computer Science and Artificial Intelligence with Professional Placement
Matthew Mather	mm3465	BSc Computer Science and Artificial Intelligence
Kan Chit Ho Matthew	mk2805	Computer Science with Professional Placement
Aleena Shaiju	as5458	BSc Computer Science with Professional Placement
Zhiming Zhang	zz2393	BSc Computer Science and Artificial Intelligence with professional placement
Vladislav Toder	vt438	Computer Science and Artificial Intelligence
Sebastian George Bodean	sgb55	BSc Computer Science
Yusuf Hassan	yh2452	BSc Computer Science
Bhuvan	bk597	Computer Science

Group Member	Email	Contribution	Comment	Signature
Felix Nacu	fn361@bath.ac.uk	8	-Attended all meetings -Worked on report and some code	F.N.
Matthew Mather	mm3465@bath.ac.uk	8	Fully contributed, working on report, code and testing	M.N.Mather
Kan Chit Ho Matthew	mk2805@bath.ac.uk	8	-Attended all meetings -Worked on report	M.K.
Aleena Shaiju	as5458@bath.ac.uk	8	-Attended all meetings -Worked on report and had role of Product Owner	A.S
Zhiming Zhang	zz2393@bath.ac.uk	8	-Attended all meetings -Worked on database code and testing.	Zhiming Zhang
Vladislav Toder	vt438@bath.ac.uk	8	Attended all meetings, worked on database and had role of scrum master	V.T.
Sebastian George Bodean	sgb55@bath.ac.uk	10	-Attended all meetings -Worked on the app and integrating the functions	S.B
Yusuf Hassan	yh2452@bath.ac.uk	8	attended most meetings (missed several due to illness), worked on and edited report	Y.H.
Bhuvan	bk597@bath.ac.uk	0	Attempted contact with them, but no response. As a result, no contributions were made by this member.	

Sleep and Screen Time Tracker

Authors: Felix Nacu, Aleena Shaiju, Matthew Mather, Kan Chit Ho Matthew, Vladislav Toder, Zhiming Zhang, Sebastian Bodean George, Yusuf Hassan

Abstract

The problem we decided to address was the growing issue of spending too much time on your mobile device causing a lack of sleep. Our research showed that there was a correlation between these variables, especially for university students, hence why we chose this issue. Our solution was to create an app that tracks the user's sleeping hours and screen time, allows them to input their goals, and notifies them when they need to go to bed and/or get off their phone. So far, our system is able to accurately keep record of sleeping hours and screen time input by the users and produce graphs comparing the weekly figures to the user's goal. There is also a trophy feature that is a count of all the times a user has achieved their goal. To evolve our system, we would like to use a health API to take the sleep and screen time data from the user so that they don't have to manually enter it. We also have yet to implement the actual notifications system - currently, you can view the data regarding when to sleep/get off your phone within the app itself.

Contents

1.Introduction

1.1 Issues the system is intended to tackle

The software system we are developing as a part of this coursework is aimed to assist university students with issues pertaining to how a large amount of screen time can alter and disrupt someone's sleep schedule. An increased amount of screen time can be associated with an unhealthier sleep schedule and lower quality of sleep, which in turn can have an impact on the overall mental health of university students (Alam, Naeem and Asad, 2023). The software aims to motivate a user to reduce their screen time and improve their sleep schedule, potentially having positive effects on their mood and productivity. The software should do this by collecting and providing information about the user's screen time and sleep time over a period of time, providing the user with the information to improve themselves.

1.2 Personal Informatics

This system will take the form of a personal informatics system which can be defined as a system to "help people collect personally relevant information for the purpose of self-reflection and gaining self-knowledge." (Li and Dey and Forlizzi, 2010). Personal informatics systems can work with a wide range of information; for example, health applications tracking someone's physical activity, self journaling (where the user keeps track of relevant events and their mood), and productivity monitoring.

At its core, a personal informatics system can be split into five sections according to a stage-based model: preparation, collection, integration, reflection, and action (Li and Dey and Forlizzi, 2010). These cover the main steps of a user and a personal informatics system. However, the personal informatics system only has to allow a user to input the information they are tracking in such a way that it can be processed to give a meaningful output for the user to reflect upon and could then take action from that point.

1.3 What makes a Personal Informatics System effective

While the concept of a personal informatics system is simple at its core, consistent user engagement is dependent on the system being built effectively. There are four common reasons behind people forgetting to use these systems: forgetting to track, difficulty managing upkeep, intentionally skipping entries, and suspending tracking (Epstein, et al., 2015). For a personal informatics system to be effective it must then have a method to prevent each of these issues.

To help the user remember to track their data, a reminder could be given in the form of a notification; alternatively, the data could be collected automatically to eliminate the user needing to track it at all. This would also make the system simpler to use and reduce the difficulty of upkeep, and the use of an intuitive user interface could reduce this difficulty even further.

The other common reasons can be seen as a lack of motivation from the user to keep using the system. Many methods are implemented into personal informatics systems to improve user retention.

One method is to give the user goals to aim for. These goal motivated personal informatics systems aim to keep a user engaged through either allowing them to set their own goal to aim for or by giving the user goals to aim for. The intent is to make the user want to see themselves progressing towards their goal, and therefore want to use the system more. Another method to improve user motivation is in-system rewards. This gives the user a sense of achievement for each item they collect and hopes to motivate the user to keep collecting these rewards.

1.4 System idea

The main idea for the system is for it to be an application (currently on Android to prototype) to allow a user to track and see the correlation between their screen time and time spent sleeping over a set intervals of time. The screen time and sleep time data will be gathered from Android's Health Connect API which allows us to access health data stored on the phone. The user will be able to set goals and will be provided notification reminders by the system to encourage them to reduce screen time and help make their sleep more consistent. This is to be developed in a group using the scrum framework.

1.5 Questionnaire and interview

We decided to conduct a questionnaire in order to collect data on the screen time and sleep patterns of our target audience. The questionnaire will use the Likert scale which will help give a holistic view of students' patterns (Willott, L, 2021). Additionally, studies found that light emitted from screens can lead to a reduction in the production of the sleep hormone melatonin (Chang, et al., 2014) which can make it harder to fall asleep and lead to disrupted sleep. Therefore, as part of the questionnaire one of the questions asks our potential users if they experience any sleep related issues such as difficulty falling asleep, trouble staying awake, waking up frequently during the night and waking up feeling unrefreshed. Another key role the questionnaire should achieve is to find potential features for our system, therefore the last question asks if they use any other applications/systems that monitor sleep or screen time and if they do what features do they like in particular. A link to the questionnaire and summarised results can be found in Appendix C.

We then carried out an interview with two students (transcripts can be found in Appendix C). From this, we found that too much screen time correlated with a negative impact on the quality of sleep. One respondent claimed that they slept better on nights where they didn't use their phone right before sleeping. They also mentioned that they wanted to decrease their screen time in general. This proves how useful our app could be - it will help encourage people to reduce their screen time hours, especially before sleeping.

2. Agile Software Process Planning and Management.

In order to achieve this system we used the scrum framework. Within this framework we organised our work into sprints. In total, we conducted 2 sprints with each sprint lasting roughly 3 weeks.

Roles

Scrum Master (Vladislav Toder) - Responsible for facilitating the scrum process and leading meetings. They planned Sprints and ensured they were conducted effectively by setting up frequent meetings. We chose him as he had great leadership and communication skills.

Product Owner (Aleena Shaiju) - Responsible for managing the product backlog and prioritising the work done by the development team based on the value to our clients. Aleena would also conduct interviews with potential clients and make sure our backlog would reflect their feedback. For example, due to feedback received we implemented a trophy system for completed goals. A record is kept each time the user completes a goal and the user is awarded a 'trophy'. This would encourage the users to use the app and create new goals to achieve.

Development Team - Consisted of individuals responsible for implementing and delivering the app. We decided our system had 3 main parts to it: the user interface, a database, and features that we received from the Product Owner. Each one of these sections had 3-4 different people assigned to them, who worked closely together on implementing their specific section.

Use of Jira

In order to keep track of our sprints we used the software Jira which is a tool for agile teams to plan, track and manage their work efficiently. In Jira we created a project backlog and created different tasks and stories that were assigned to individual members of our team. Jira also allowed us to create a board in which we could see at which stage each task was helping with the tracking of our sprints. The four stages were To Do, In Progress, Testing and Done.

2.1 Sprint 1

Dates: 25/03/2024 to 14/04/2024

We had a planning session on 25/03/2024 where we decided on our goals for the first Sprint. Our aim was to do some more background research so that we knew what features to implement, could find an API to use, set up a database and have a template for different scenarios the user would have and how they can set their goals.

When planning our first sprint we created a Sprint Backlog using Jira (found in Appendix E) which included tasks split into the following 4 categories which were each given a priority level in order to make sure we at least achieved the most important tasks:

1) Research sleep and screen time monitoring. This included finding an API for both sleep and for screen time monitoring and working on implementing the API. **Priority: Low**, at this stage we felt we could use fake data and an API is not needed.
2) Do some background research into existing systems and find out user needs by conducting elicitation interviews with students and creating a questionnaire. **Priority: High**, at an early stage of the development process this is important in order to figure out what we want our system to achieve.
3) Create a database where each user will have their sleep and screen time stored. **Priority: High**, the database is the foundation of our system so this is a high priority task.
4) Create a method for users to store and update their goals regarding screen time and sleep.**Priority: Medium**, we felt this would be more important to build the database first but this should still be aimed to be achieved this sprint.

Sprint 1 Review and Retrospectives

After completing our first sprint, we held a retrospective meeting on 15/04/2024. Here we discussed what worked for us and what we should improve in order to further progress. We found the database implementation successful, and though we didn't plan on creating a UI initially at this stage we managed to create a prototype using Android studio. We decided we should aim to meet more often during our sprint in order to more easily adapt our objectives if needed as during Sprint 1 we decided we should delay the API integration and focus on core functionality. Our Product Owner also conducted another interview where she demonstrated our current program to our client. From this we gathered we should aim to improve our user interface and include additional functionalities such as data visualisation tools and a trophy system that rewards users if they meet their goals.

2.2 Sprint 2

Dates: 15/04/2024 to 1/05/2024

During the meeting on 15/04/2024, we came to an agreement that we needed to split some of the pre-existing tasks into smaller subtasks as well as creating new goals to cover areas that were overlooked before. The Sprint 2 backlog can be found in Appendix E.

1) UI - Utilising the template of the google form questionnaire, query users of where their current goals are set. **Priority**: **High,** carrying over from Sprint 1 improving on the prototype made in Android Studio.
2) Graphs and diagrams - Graph that allows users to visualise their progress in comparison to their goal. Use case & UML diagrams as a schematic representation of the structure of our system. **Priority: Medium,** crucial for explaining and showcasing our system.
3) Notification - Reminding the user is an element of what makes a PI system effective. **Priority: Medium,** although still important but not as much in comparison to the other outstanding tasks.
4) Trophy implementation - A visual form of encouragement for the user, open the app to see a trophy if the goal is met and plus 1 to the total amount they have amassed.

Priority: Low, shouldn't take long according to one of our group members and moreso a bonus feature.

Sprint 2 Review and Retrospectives

On 29/04/2024, we concluded our second sprint and held another extensive meeting lasting almost two hours to consider our progression. The user interface achieved the intended user-friendly experience as well as the goal setting procedures and features such as setting the local time along with their intended goal. Users will also have access to comparison graphs of their planned goal in contrast to the data collected. The trophy reward system was integrated smoothly and displayed whenever the user achieves their goal. Although feedback proposed having a notification pop up, we decided that the notification system hinders the app's basic function so after some discussion it was ultimately scrapped. As discussed in Sprint 1 retrospectives we succeeded in holding more frequent meetings, which helped us reconsider our aims for the sprint.

3. Software Requirements Specification

We used various methods of research to establish our system requirements. Using initial requirements, we tried to decide upon two different kinds of PI data. As the main target audience of our product was university students, we thought of areas that we personally could all improve on. The number of hours we sleep at night and our screen time were the two data fields we chose to use. By carrying out questionnaires and interviews, we came to the conclusion that other students also felt that these were two areas they wanted to improve on in their daily lives. Research showed that up to 60% of all university students have poor sleep quality (Schlarb, Friedrich and Claßen, 2017), with students spending anywhere from 50.2% to, where remote learning is present such as in lockdown, 77.6% of their time using screens in some cases (Safranek, 2020). We felt that there was a need for an app to focus on improving these figures.

Firstly, we determined what exactly we wanted our system to do. These functionalities were:

- Taking in two fields of data (sleeping hours and screen time).
- Allowing the user to insert goals for both fields of data.
- Notifying the user each time they:
    (1) go over the screen time limit
    (2) need to go to bed in order to achieve their sleeping hours goal.

We decided upon creating an app over a website, as we want our system to be able to use the personal information - specifically screen time and hours slept - stored in most mobile phone health apps. A website would make it much harder to access that data stored within apps. To acquire the data from the health apps, we initially planned on using Android's Health Connect API.

For the app, we decided to create an Android app using Android Studio. We chose to create an Android app due to there being a higher number of Android users than iOS users (Backlinko, 2024), making Android apps more easily accessible by students. We chose to use Android Studio as it allowed us to test our app using the Virtual Machine tool, and the

IDE had performance analysis tools to help optimise the code. A limitation of using Android Studio is that the app will not be available on iOS devices, but that is something that we can work on in the future. A benefit of our app is that there will be no hardware requirements, beyond having an Android device.

3.1 Research

To build upon our initial functionalities, we looked into what else we wanted our app to do. In Interview 2, it was mentioned that it can be really tempting for the student to use their phone even when wanting to sleep early. The respondent mentioned wanting notifications that stayed at the top of the screen to encourage them to not use their device. However, we felt that this may not be a good idea as the notification may cover a part of the screen that is important for the user to access. As a result, permanent notifications would not be in the best interest of the user and we decided to not implement it.

The respondent also mentioned that they think our system should have an incentive/reward system to encourage people to actually achieve their goals. They mentioned an app that would ‘grow’ a plant whilst the app was open (encouraging you to do something else, like studying), and if you ever left the app it would ‘kill’ the plant. This was a very unique feature that made us recognise the need for an incentive in our own app. As a result, we have decided to incorporate a trophy system into our app. Every time the user reaches their goal, the counter of trophies they’ve achieved will also increase.

The specification also recommended that we create graphs to show correlations between data. We decided to display a graph showing the sleeping hours and screen time of the user throughout the past week on the app. In Interview 2, the respondent also felt that it would be a nice feature to add and that it would make it easier for them to understand their trends and patterns. This was important feedback as it supported our decision to add this feature.

As a part of our research to develop the requirements for our system, we investigated similar systems that are already publically available. One of these systems was the SleepScore app that focuses on improving a user’s sleep. By looking at this system we found several features aimed to improve a user’s experience of the app. Many of these features were designed to give the user summarised information reducing the work required to obtain relevant information as to how a user should improve their sleep. These features include giving an overall number score to each night’s sleep and rating them on varying criteria, incentivising a user to improve as it's simply just increasing a number almost like a game. Due to SleepScore obtaining sleep data in a different way, these kinds of ratings wouldn’t be possible to introduce into our system; however, this helped us develop the idea of having a trophy system to incentivise the user to reach goals. The application also displays graphs showing trends in the user’s sleep pattern over different time intervals being one week, several weeks, or months. This feature inspired us to include similar graphs in our system. Furthermore, the system could then offer advice for improving a user’s sleep schedule. Due to the complexity and scientific knowledge required to give correct advice for improving sleep, we opted to only send simple suggestions and motivational messages through the use of notifications.

Another feature common across many existing apps or phone features (such as Apple Screen Time, Google Family Link, or Qustodio) is that they all have the option for parents to

set daily screen time limits for their children. It also allows parents to block the use of other apps at certain times and schedule the phone to be disabled at certain times, and these can be done remotely. Whilst we felt these were really important features, we didn't believe they were necessary for our app as our target audience was university students.

There were shortcomings with those apps that we felt we wanted to avoid. There were complaints that Apple Screen Time was difficult to set up and use, and so we knew to make sure our app was easy to use. Qustodio also required users to pay a monthly subscription ($55 a year). This was not something we wanted to implement as students would not be able, or more importantly willing, to pay such a high price for our app.

3.2 Requirements

Our initial priority was to create a functioning database that would store all of the data and carry out the functional calculations (e.g. 'has the user achieved their goal?'). Our next main priority was the app, and creating a user interface that was simple and intuitive. We needed to ensure that the data input would also produce the right graphs, and correctly send notifications to the user. Only afterwards would we focus on incorporating an API to extract the data from the user's device.

However, we realised that sending notifications to the user is quite difficult to complete in our allocated time frame, and thus it is a feature we will implement in the future. Similarly, we decided against using an API for now as it is also quite complicated to integrate into our system. This left us with the following final list of functional requirements:

Requirement Number	Description	Priority/Coursework Specification Reference
1 Database Functionalities		
1.1	Inserting data into the database	Priority Level: High Reference: 5.1, 5.3
1.2	Deleting data from the database	Priority Level: High Reference: 5.1
1.3	Retrieving data from the database	Priority Level: High Reference: 4.2, 5.2
1.4	Updating the data in the database (e.g. if you actually slept less hours the day prior)	Priority Level: High Reference: 4.2, 5.3
2 Goals and Graphs		
2.1	Returning the number of hours slept at a specific date	Priority Level: Medium Reference: 6.1

2.2	Returning the average sleeping hours between two dates	Priority Level: High Reference: 6.2
2.3	Returning the screen time at a specific date	Priority Level: Medium Reference: 6.1
2.4	Returning the average screen time between two dates	Priority Level: High Reference: 6.2
3 User Interface		
3.1	Allowing the user to input a goal for screen time	Priority Level: High Reference: 7.1, 7.2
3.2	Allowing the user to input a goal for sleeping hours	Priority Level: High Reference: 7.1, 7.2
3.3	Creating a graph that shows hours slept and screen time across the past week, with lines that show what the goal is	Priority Level: Medium Reference: 6.1, 6.2
3.4	Displaying the graph to the user	Priority Level: Medium Reference: 6.1, 6.2
3.5	A trophy count that increments each time a user achieves a goal	Priority Level: Low Reference: 7.4
3.6	Allowing the user to update their goals once a new one is reached.	Priority Level: High Reference: 7.3

Non-functional requirements:
These are requirements we focused on throughout the project, and so we did not number them. They apply to all sections of our project.

- **Usability**: We want our app to be a very intuitive system. The app interface should be designed to be very learnable – it should require minimal training to be able to use it, even if the user is to go without using the app for a long period of time. There is no need for any user documentation.
- **Security**: The app should be very secure as we do not plan on using a centralised server to store all user information. We plan on storing all information locally on the device, hence why we have not included a registering/log-in system in our functional requirements. As a result, there is no risk of someone gaining unauthorised access into a database and retrieving the personal information of all users. Unless the user's device itself has a security breach, our app should be very secure and users shouldn't feel worried about their private information. For the data that we do collect,

it is only the minimal amount of data needed for our system to function. The data is also anonymous on the system and we as developers have no way to access it. This means our system follows GDPR regulations.

- **Performance**: The system shouldn't have a problem with multiple users due to it not being a centralised database. As a result, it should be quite scalable. New users can join and there will be no need to redesign the system, and no significant negative impact on the performance of the app.

3.3 Use Cases and Scenarios

We have modelled our use cases into use case diagrams in order to provide a concise summary of what the system should achieve at an abstract level.

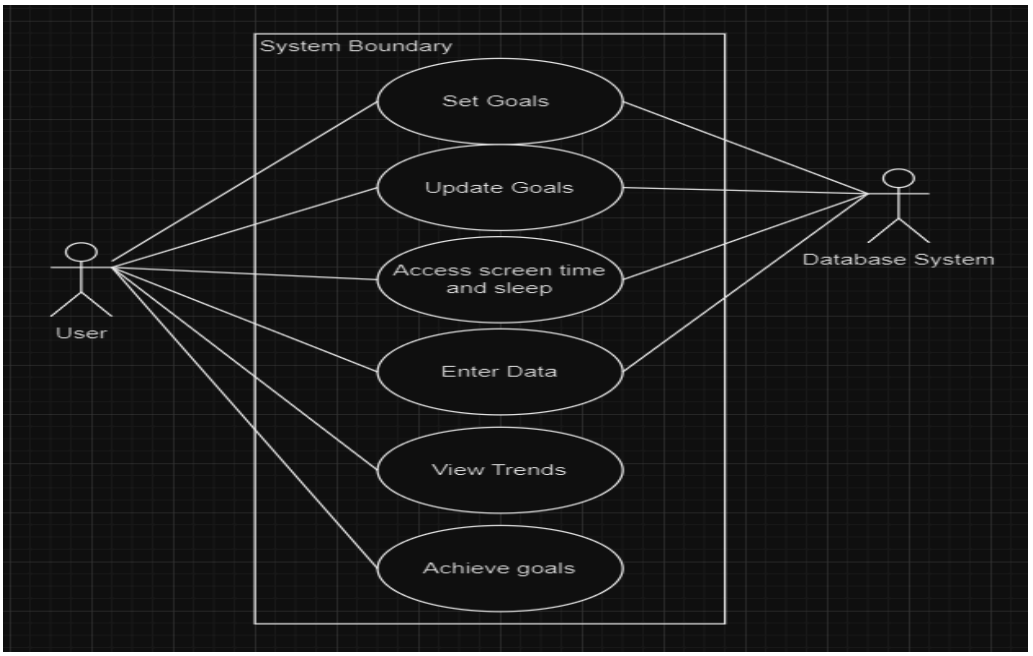


Figure 1

Use Case Scenario 1: Set Goals

Actor: User, Database System

Summary: The user sets a personal goal for daily sleeping hours and screen time directly through the app interface.

Steps:

1) The user opens the app and navigates to the 'Goals' section.
2) The user inputs a target value for daily sleeping hours / screen time (e.g., 8 hours) and the time frame for the goal.
3) The app saves these values in the database and confirms to the user that the goals have been set.
4) These goals are now used to track and notify the user accordingly.

Post-Conditions: The user's goals are stored in the database system and used for daily tracking. The user will then also be able to update these goals.

Use Case Scenario 2: View Trends

Actor: User

Summary: The user views graphical trends of their sleeping hours and screen time over selected periods.

Steps:

1) The user selects the 'History' option in the app.
2) The user chooses a timeframe (e.g., week) to view data trends.
3) The app retrieves the relevant data from the database.
4) The app displays a graph showing sleeping hours and screen time, highlighting any discrepancies from set goals.
5) The user reviews their activity patterns to make necessary lifestyle adjustments.

Post-Conditions: The user gains insight into their behaviour over time, aiding in self-awareness and behaviour modification.

Use Case Scenario 3: Achieve Goals

Actor: User

Summary: The user archives a trophy once they achieve their goals.

Steps

1) At the end of the timeframe of the goal the app calculates if the user has met their daily goal consistently.
2) If the goals have been met the app generates a trophy as a reward.
3) When the user next logs into the app they will be able to see their new trophy, boosting their motivation.

Post-Conditions: Rewards encourage ongoing commitment to achieving set goals, as found from our elicitation interview, enhancing user engagement.

4. Design

4.1 Use of Java

For our implementation we have chosen to use Android Studio as previously mentioned alongside Java. We chose to use Java because it is the second most widely used language in the app development industry (Anon, 2023), which offers us valuable industry-like experience. Another advantage of Java is that it is an object oriented language. This makes allocation of tasks particularly easy as individuals can work on separate classes and then easily integrate them together once done. Java's encapsulation and inheritance further

ensure that each component can be developed, tested, and reused effectively, and if needed, expanded on in the future.

4.2 Database

The application's database is organised around a primary table, which uses date as the primary key and includes two additional columns: "sleeping hours" and "screen time". Each user maintains a personal database to record those metrics chosen. The functions of the database are described in a Java interface (referenced to "Figure 2 Function Table") and implemented in a Java file (referenced to "Figure 2 Queries"). This file contains methods for executing SQL queries.

For example, if a user needs to edit a their sleep time for a particular date, the user can insert said date and another sleep time through the user interface and then the application triggers the relevant method at the user interface inside, which executes an SQL query to update the sleep time for the specified date (Functional Requirement 1.4). Using the date as the primary key facilitates user inputs, so that all relevant data for a particular date can be accessed by simply entering the date into the application (Functional Requirement 1.3).

The database also includes a delete function in the interface (referenced to "Figure 2 Function Table") designed to delete the data on a monthly basis to free up storage space (Functional Requirement 1.2). Furthermore, the insertion function is multifunctional; The user's current personal information can be automatically used for new entries or allow manual entry of past data in screen time or days through the interface (Functional Requirement 1.1).

4.3 Goals

One of the main parts of our system was the creation and validation of goals. The following UML class diagram represents how this part of our system functions and interacts.

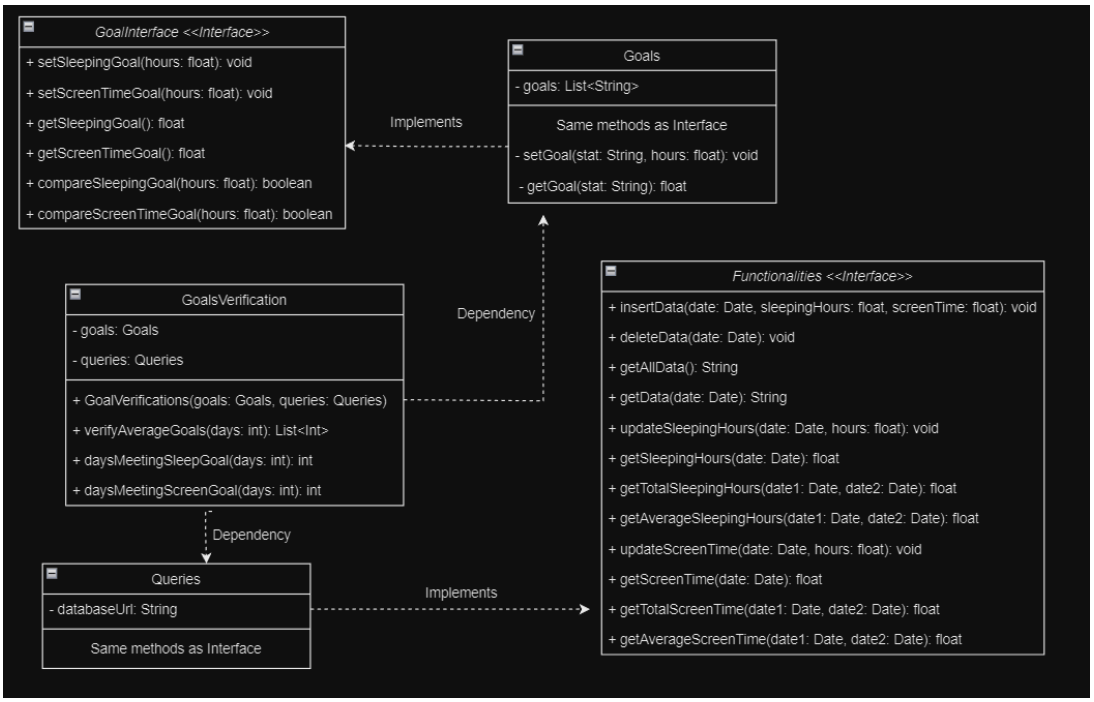


Figure 2

Our Goals section, which can be seen in the UML class diagram Figure 2, consists of 3 classes and 2 interfaces for those classes which would be used if we are to further expand our system in the future. The Queries class is a gateway between our system and the database. This allows us to retrieve key stats from the database like screen/sleep time between 2 dates, screen/sleep time at a particular date (functional requirement 2.1 and 2.3) and average sleep/screen time(functional requirement 2.2 and 2.4). The Goal class is used to create daily goals (functional requirement 3.1 and 3.2) and update their goals (functional requirement 3.6). Goals are stored in a .txt file which is accessed through the goals class. The goal verification class then leverages both these classes and will verify if the user is on track with their goals by allowing other parts of the system to see how many times their goal was met in the past x days for both sleep and screen time, and also to see if on average they are meeting their goal in the past x days from which the system can then award the user a trophy (functional requirement 3.5) if at the end of the week they have achieved their goal on average.

4.4 User Interface

The GUI was designed and developed to be intuitive and quick to pick up, in order to fit the non-functional requirement of usability. We opted for simple self-explanatory buttons at the bottom of the screen. In the actual implementation, we may make the buttons bigger to make them even clearer to the user. Similarly, we chose to keep the interface simple. Too much on one screen could become overwhelming for the user, which would not encourage them to continue using the app. These features can be seen demonstrated in the app wireframe below:

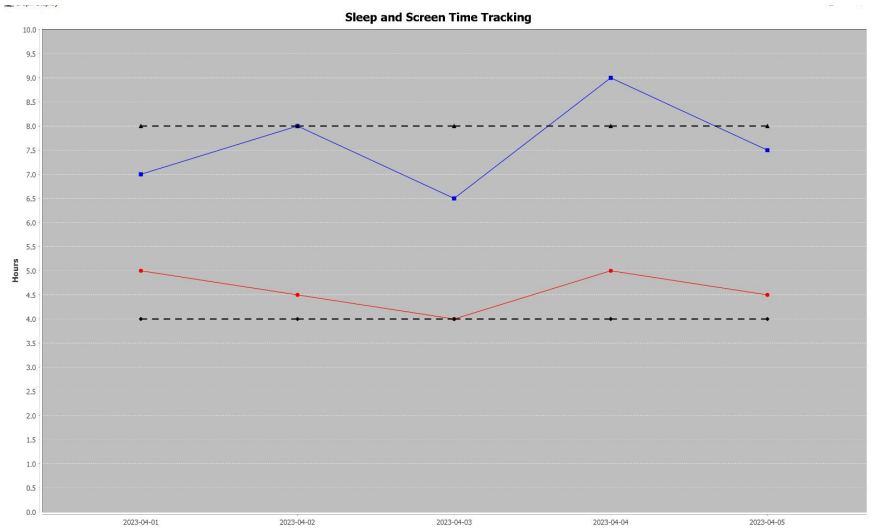
Figure 3

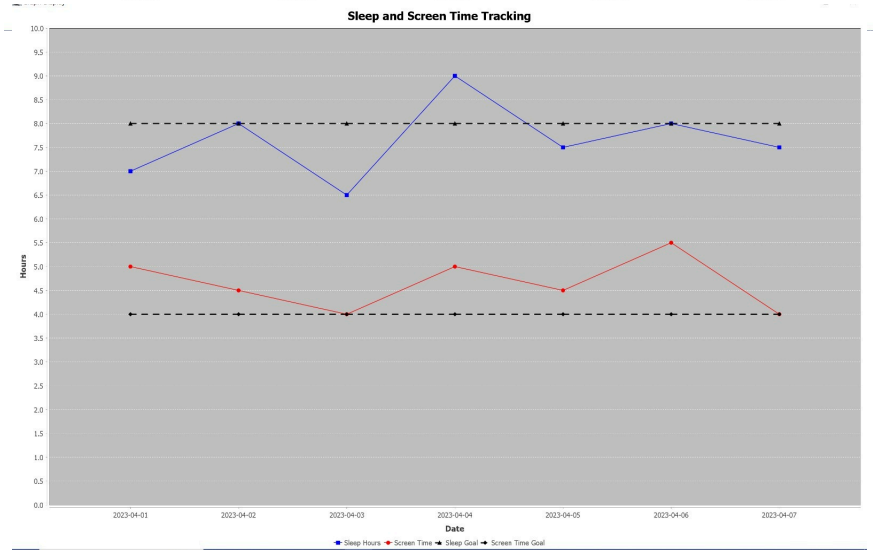
The ‘App Name’ frame is the home screen of the app. It will display the time and then state how long the user has before going to bed. As well, there will be two progress bars that display how close a user is to reaching their goal, one bar for sleep time and screen hours respectively. The ‘Add Data’ frame asks the user if they’d like to enter data as hours slept, which would be used when entering data from the previous night (functional requirement 1.1). This is likely to be a toggle button in the actual implementation. The ‘Set Goals’ screen is used, as the name suggests, to set goals. The user can enter the time they usually go to

bed at, and the time they want to start going to bed at (functional requirement 3.1 and 3.2). The latter of the two is then used for the ‘You have x time left before going to sleep’ message on the home screen. If the user wishes to update one of their goals they can simply redo this process and their previous goal will be overwritten (functional requirement 3.6). The ‘Screen Time’ button will also display the options related to screen time (in the design we have only included the page for ‘Sleep Schedule’). Finally, the ‘History’ screen will display the graphs we create on the user’s data over the past week. This is also in accordance with our software requirements. Throughout the design process, we included progress bars on the ‘App Name’ page for the user to see their own progress. Due to the feature not being of a high priority, and due to time constraints, this was not added in the final implementation. It is something we would include in the future.

4.5 Graph

A key part of our system was data representation. We decided the best and most concise method of achieving this would be through a visual representation of a graph, this graph is created through the use of the jfreechart library plotting two horizontal lines to showing the users screen time and sleep hours goals along with two lines showing the users stored data in the systems database. The graph allows the user to see their progression in sleep/screen time alongside their goal and hopefully motivate them to improve. The graph should also help users realise the correlation between screen time and sleep. As can be seen in Figure 4, they can realise how screen time is negatively impacting their sleep.





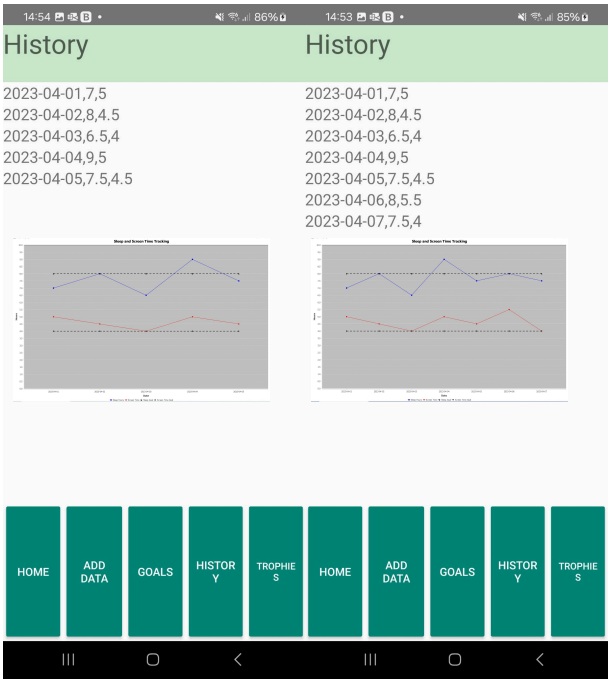


Figure 4

5. Software Testing

5.1 Testing Plan

To make sure that the requirements were met in the specification and that no errors appeared in the code, we are going to test each method individually in unit tests as well as with integration tests to ensure each module works as a cohesive system. Our test plan acts as an outline so the testing is thorough. To make the test plans simple and get results quickly to compensate for the issue of not knowing how long sections are likely to take, the deadlines for test plans are very flexible in nature and are generally assigned to the original writer of the code.

5.1.1 I/O Database

Before integrating Java classes and functionalities into Android Studio, we tasked two team members with developing an I/O database. This database supports five key operations: delete, select, insert, and update, along with an additional I/O function. Testing these database and graph classes in advance proved invaluable, as it allowed the team to make necessary improvements and adjustments before the actual app implementation (This can be seen in Appendix D).

5.2 Unit and Integration Tests

As part of our testing of the system, we completed unit tests on each method and integration testing when a class requires another class to work. Here is a summary of the testing done on the Goals class using JUnit. Having one issue in the first test (referenced to 2.1 in the table).

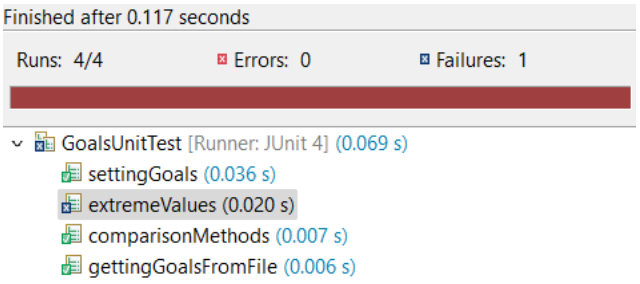


Figure 5

Test No.	Requirement Tested	What is being tested?	Data Inputted	Result	Comment
1	3.1, 3.2	Setting goals with the Goals class using acceptable values	ScreenTimeGoal = 1.3 SleepingGoal = 8.56	Pass Goals equaled those initially set	No issues

2.1	3.1, 3.2	Setting goals with erroneous data	ScreenTimeGoal = -10, 26 SleepingGoal = -1, 26	Fail set goals of negatives = 0 but accepted 26	Added an extra line to try fix the issue
2.2	3.1, 3.2	Setting goals with erroneous data	ScreenTimeGoal = -10, 26 SleepingGoal = -1, 26	Pass Goals set to the most extreme in possible range.	No issues.
3	2.1, 2.2, 2.3, 2.4	Testing the comparison method returns the correct values for different inputs	ScreenTime =1.3 compared to 1.1,1.4 Sleeping =8.7 compared to 8.2, 9.2	Pass True returned for 1.1 and 9.2, false returned for 1.4, 9.2	No issues
4	2.1, 2.2, 2.3, 2.4	Testing goals are retrieved from the file so all goals objects can access the goals.	ScreenTimeGoal = 1.3 SleepingGoal = 8.56	Pass New instance of Goals class retrieved the goal values	No issues

5.3 System Testing

Test No.	Req. No.	What is being tested?	Data Inputted	Result	Comments
1	2.1, 2.3	Record of user's sleep and screen time is displayed on the History tab.	screenTime = 8.0 sleepingTime = 2.0 for both dates Date1 = 2023-10-10 Date2 = 2022-10-10	Pass Records and displayed in chronological order	No issues
2	2.1, 2.3	User is able to edit their stored data from the History page.	Date = 2023-10-10 newScreenTime = 20.0	Pass New screentime is equal to the new value	No issues
3	3.1	User is able to input a goal for screen time.	Date = "2024-10-10" screenGoal = 4.0	Pass System successfully saves the user's goal	No issues
4	3.1	User is unable to input invalid screen	Date = "2024-10-10"	Pass System	No issues

		time goals (i.e. non-integers/non-floats).	screenGoal = “asfads”	returns an error message stating the issue	
5	3.2	User is not allowed to enter unreasonable data for the screen time goal	Date = “2023-10-10” screenGoal =“71678.0”	Pass System returns an error message	No issues
6	3.2	User is not allowed to enter negative values for the screen time goal	Date = “2023-10-10” screenGoal =“-4.0”	Pass System returns an error message	No issues
7	3.2	User is able to enter a goal for hours slept.	Date = “2023-10-10” sleepingGoal = 8.0	Pass User is allowed to enter and save the data	No issues
8	3.2	User is unable to input invalid sleep time goals (i.e. non-integers/non-floats).	Date = “2024-10-10” sleepingGoal =“asfads”	Pass System returns an error message.	No issues
9	3.2	User is not allowed to enter unreasonable data for the sleep goal	Date = “2023-10-10” sleepingGoal =“71678.0”	Pass System returns an error message	No issues
10	3.2	User is not allowed to enter negative values for the sleep goal	Date = “2023-10-10” sleepingGoal =“-4.0”	Pass System returns an error message	No issues
11	3.3, 3.4	Users can view a graph displaying hours slept and screen time across the past week from the History page.	Open the History tab.	Pass System returns a graph on History page	No issues
12	2.1, 2.3	A bar showing the user’s progress towards their current goals is displayed on the home page.	Open the Home tab.	Fail The display remains the same regardless of progress	This issue appeared almost at the last moment, we were not able to fix it
13	3.6	Users can set a new sleep time/screen time goals upon	Complete a goal, then open the goals tab and	Pass Capable of setting a new	No issues

		completing their current one.	attempt to set a screen time/sleep time goal.	goal	
14	3.5	A trophy count representing the number of completed goals is displayed on the Trophies page.	Open the Trophies tab.	Pass Trophy tab displays intended information	No issues
15	3.5	The trophy count is incremented when a goal is completed.	Complete a goal, and then reopen the Trophies tab.	Pass Trophy count increased by 1 upon completion	No issues

As part of our database system testing we also created a testing class which can be found in Appendix D Figure 6.

6. Reflection and Conclusion

6.1 Review of System

Software Requirements:

Throughout our project, we managed to successfully complete our list of software requirements. When we realised that it was difficult to complete something within the required timeframe or that we lacked knowledge in that area, we were able to quickly adapt our requirements. We did this by holding regular group meetings, the minutes of which can be found in Appendix B.

There were two features that we had to remove from our initial requirements due to time constraints. These were the health API and notifications to let the user know when to go to bed/stop using their phone. Using an API would have meant the user no longer had to manually enter in their own data, which would make it even easier for them to use the app. An automatic notification would let the user know when to sleep/put their phone away, instead of manually opening the app to check (which can be tedious and an easily forgotten task). We would have liked to also add notifications with words of encouragement (e.g. 'You've achieved your goal twice in the past week!') to keep users motivated. These are features we believe would elevate our app by making it more user-friendly.

Another issue is that our app is only available for Android devices. Whilst Android devices are popular, a wider range of people could benefit from the app if it was also available to iOS devices. In the future, we would use a cross-platform framework to create an app that would be compatible with all operating systems.

One element we did not consider for our non-functional requirements is system reliability. Our system currently has no back up process. If the user was to lose their device, or accidentally delete the app, the data would be gone and unrecoverable. This is due to our system having no centralised database. By running the system and storing the data on a server, with a login system for all users, we could also implement a backing up system for the data.

Design:

For this section of the project, we may have benefitted from starting earlier. By creating rough sketches sooner, it may have been easier to create and design the layout of the user interface. At the very least, it would have been a much quicker process than it was. Due to design not being completed earlier, not all members of the group had input into the final design. Despite this, we managed to keep consistent with the colours and fonts used and the interface looked cohesive and presentable.

To further improve the design of our interface, we would like to add a dark mode and a light mode. Dark mode especially is easier on the eyes, and would make the app more accessible to a wider audience. As people will be checking this app at night to know when they need to sleep, a dark mode would make it easier for people to use the app in an environment with less lighting.

Another area to be improved is the trophy counter. Our trophy counter is a single variable that does nothing but display the number of times a user has achieved a goal. This may get boring for the user after achieving a few goals, especially as the trophy image we have used is a plain black trophy. To make it more exciting, the trophy counter could be changed to a score counter. Each time a score milestone is achieved (e.g. fifteen goals achieved), then the trophy is upgraded (e.g. from bronze to silver to gold, or getting bigger in size). This would make the app more like a game and encourage the users to constantly keep trying to achieve their goals. The change that every score milestone makes would make the user interface for that page more exciting.

Software Testing:

One major challenge came from the specific requirements of Android Studio. Our initial application was written in Java and had to be adapted because Android Studio did not have native support for SQLite. This limitation required us to use Android Studio's internal functions and modify several classes, which was a time consuming process. This adaptation process highlighted the importance of understanding the capabilities and limitations of the development environment early in the project lifecycle. To avoid such problems in the future, it would be better to conduct further research into the software we use.

In addition, our team faced logistical issues due to hardware limitations. Only one team member owned an Android device, forcing the rest of the team to rely on emulators. This setup led to a number of problems, such as compatibility issues with service providers and complications when initialising data storage files. These obstacles significantly hampered effective collaboration and code sharing between team members, as only two developers were able to consistently share and test code effectively. This situation highlighted the critical need for a robust testing infrastructure capable of simulating a range of environments and devices.

Despite these challenges, there was plenty of success within the team responsible for the Android application's user interface. They found that integrating new classes and making changes to the application's UI was facilitated by straightforward command line operations. Once team members became familiar with the procedures for modifying the application's UI, they were able to efficiently implement enhancements, including the addition of new sections, images and colour schemes. This ease of modification was a positive aspect, demonstrating the adaptability of our chosen development tools and the benefits of a well-designed architectural framework.

6.2 Review of Agile Process

Due to this project being a new way of working for many members of the group, we experienced many issues that needed to be overcome, e.g. communication and lack of knowledge.

To combat difficulties in communication and understanding, we tried to have frequent meetings within each sprint and learnt to use software such as Jira and Github to manage and share work across the group. Jira allowed each member of the group to individually (or as a group) be assigned to tasks. It also allowed everyone to know what stage of completion each task was at to better coordinate workloads. While these software helped, due to the

inexperience of the group with using these software, we feel they were not utilised to their fullest extent. This created delays in the development process. However, as the project progressed and members became more familiar with the process, delays decreased and the group acted more cohesively.

While these methods helped, communication ended up being especially difficult over the two week Easter break, and due to it being near the beginning of the project, some members found it hard to work on the project as not all details had been fully discussed and understood by all group members. This led to a less productive two weeks. We may have completed more tasks over the break if more meetings were carried out during these two weeks.

One aspect of managing workload affected by the inexperience of the group was assigning specific roles to group members, with sections of a sprint backlog being split into larger tasks assigned to a single member. This required certain members to be constantly changing what aspect of work they were completing, e.g. development to design to testing and back to development. While this gave flexibility to the group, it also resulted in certain tasks being more error prone. Instead, we could have assigned roles such as developer, tester, designer at certain points to ensure a clearer development path. Splitting these larger tasks into even more subtasks could have also helped this issue of group members being unsure of the direction of the project.

Agile:

The agile methodology splits the process into sprints. During our project we completed two sprints, where alongside developing and implementing sections of the system, we adapted our requirements and plan to better fit an end goal. These changes came from learning more with research of other available systems and our own primary research as shown in sections 1.5 & 3.1 of the report.

At the beginning of the project we may have been overly ambitious. We wished to implement features into the final system that, through the first and second sprint, we discovered were not vital to the core functionality of the system and causing other issues. These were the API and notification system, which were taking a large amount of time and effort away from getting other system functions working. Due to this we first delayed when the implementation of an API would take place in the first sprint, before removing both features from the requirements list in the second sprint. As a result, while initially this acted as a set back to the group, this helped us gain an idea of how long certain aspects of a project can take, and how to effectively allocate time to tasks in the future. Learning that it is better to delay and change priorities of certain features if they are causing issues, this is something that we would discuss and manage better using agile again.

Throughout our project we held discussions about the priorities of our system, which was done very early on, so that we knew what our end goals were. We believe we decided on the priorities of the system well, as even in reflection we do not think we'd change our priority order (other than considering API and notifications in the beginning). The priority order allowed us to focus on important tasks that were needed for other tasks to be completed and

were vital to the system's functionality. This helped us complete all major tasks within the project's time frame and ensure they were of high quality.

References

Alam, A., Naeem, A. and Asad, M., 2023. *The Impact of Screen-time on Sleep Quality and Mental Health among University Students.* Faisalabad: Government College University Faisalabad.

Anon, (2023). *Top 12 Popular Java Applications Examples in Real-World - Scalo* [Online]. Available from: https://www.scalosoft.com/blog/top-12-popular-java-applications/#:~:text=It%20is%20the%20second%20most [Accessed 1 May 2024].

Backlinko, 2024. *iPhone vs Android Statistics.* [Online]. Available from: https://backlinko.com/iphone-vs-android-statistics#iphone-vs-android-phone-sales [Accessed 25 April 2024].

Chang, A.M., Aeschbach, D., Duffy, J.F. and Czeisler, C.A., 2014. Evening use of light-emitting eReaders negatively affects sleep, circadian timing, and next-morning alertness. Proceedings of the National Academy of Sciences, 112(4), pp.1232–1237.

Epstein, D.A., Ping, A., Fogarty, J. and Munson, S.A., 2015. A lived informatics model of personal informatics. *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 7-11 September 2015, Osaka, Japan. New York, USA: Association for Computing Machinery, pp. 731-742.

Gulotta, R., Forlizzi, J., Yang, R. and Newman, M.W., 2016. Fostering Engagement with Personal Informatics Systems. *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*, 4-8 June 2016, Brisbane, QLD, Australia. New York, USA: Association for Computing Machinery, pp. 286-300.

Li, I., Dey, A. and Forlizzi, J., 2010. A stage-based model of personal informatics systems. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 10-15 April 2010, Atlanta, GA, USA. New York, USA: Association for Computing Machinery, pp. 557–566.

Safranek, C., 2020. Stanford students now spend four-fifths of the waking day staring at a screen; is this the new college normal?. *The Stanford Daily,* 8 July.

Schlarb, A.A., Friedrich, A. and Claßen, M., 2017. Sleep problems in university students - an intervention. *Neuropsychiatric Disease and Treatment*, 13, pp. 1989–2001.

Willott, L. 2021. *Likert Scale: What You Need to Know.* [Online]. Available from: https://www.customerthermometer.com/customer-surveys/likert-scale-examples-questions-what-you-need-to-know [Accessed 20th April 2024].

Appendices

Appendix A Meeting Minutes

Minutes for meeting on **18/03/2024** at **15:15**.

The meeting took place in person in **CB 4.17**

Attendees - Aleena, Felix, Matthew K, Matthew M, Sebastian, Vlad, Zhiming

Objective: Initial meeting of group for discussion of coursework

Points discussed:

1. Coursework specification and requirements of the system.

Each member of the group read through the specification of the coursework. Any queries regarding the specification were discussed as a group.

2. Aims of the system to be developed.

A group discussion took place suggesting ideas of what to base our personal informatics system on. This discussion included what data could be collected and compared, along with the possible source of the data (e.g. Fitbit, Smartphones) along with how a user could set self goals. As well, the main goal of the system and how this matches the specification was also considered.

3. Future communication between group members.

Phone numbers of all present members were collected and formed into a WhatsApp group and a google doc setup for collaborative work.

Actions:

1. All members should sign up for GitHub and gain some familiarity with it by the next meeting.
2. All members should research an article on either general “Personal Informatics software systems” or a system related to our current main idea of tracking sleep, steps and calories burnt. Results should be placed on the group google docs by the next meeting.

Next meeting will take place on 25/03/2024.

Minutes for meeting on **25/03/24** at **15:15**.
The meeting took place in person in **CB 4.17**
Attendees - Aleena, Felix, Matthew K, Matthew M, Sebastian, Vlad, Zhiming, Yusuf
Objective: Start Sprint 1

Points discussed:

1. Jira

The group all signed up to Jira to keep on track of the tasks and manage progress.

2. List of tasks

By analysing the overall project, the group managed to break down the project into a list of smaller, more manageable tasks. There were three main tasks the project could be divided into: acquiring data from the database, applying functions to the data (such as comparisons), and then performing an action (notifying the users). The aim of the first sprint is to focus on the first two categories, and split them into further smaller tasks.

3. Prioritised the tasks

After creating the list, the tasks were ordered in terms of priority. If the project is created following this order, it will allow for the most important parts of the project to be completed. This way, less important tasks only need to be added if it is within our capabilities/allocated time frame (and no time is wasted on something unnecessary).

4. Allocated group members to the tasks

Everyone selected which tasks they were more comfortable with working on.

Actions:

1. Conduct further background research into features to implement into our system (e.g. which API to use).
2. Set up a database with functionalities (inserting data from user, updating data, deleting data, etc).

Minutes for meeting on **15/04/24** at **15:15**
The meeting took place in person in **CB 4.17**
Attendees - Aleena, Felix, Matthew K, Matthew M, Sebastian, Vlad, Zhiming
Objective: Start Sprint 2.

Points discussed:

1. Progression with tasks from Sprint 1

Everyone discussed how much progress was made with the tasks from sprint 1. This allowed the group to understand what had been completed, and which tasks need to be prioritised for the next sprint.

2. Project Revision

By looking at the workload and the amount of time left to finish the project, the group decided to withdraw looking at implementing an API for the time being. As well, the group discussed the details of what the user interface of the app should look like and which features would be included/discarded.

3. New tasks set out.

The group discussed the next most important tasks to be completed and were assigned to the tasks.

Actions:

1. Complete tasks set out in sprint 2 as listed on Jira.

Minutes for meeting on **19/04/24** at **11:15**
The meeting took place in person in **EB 1.1**
Attendees - Vlad, Sebastian
Objective: Two of our group members went to speak to our lecturer to ensure our project was on the right track.

Minutes for meeting on **22/04/24** at **15:15**
The meeting took place in person in **CB 4.17**
Attendees - Aleena, Felix, Matthew K, Matthew M, Sebastian, Vlad, Zhiming
Objective: Reconfirm and consider transfer more people to write the report

Points discussed:

1. Re-distribute resources to other parts of the project
   The group decided that the report requires more attention and had more people work on it.

Minutes for meeting on **26/04/24** at **11:15**.

The meeting took place in person in **EB 1.1**

Attendees - Felix, Matthew K, Matthew M, Sebastian, Vlad

Objective: Discuss technical difficulties that occurred

1. Report
   Needs to be completed within the weekend, aim to have a semblance of completion by Monday.
2. Implementing SQL in android studio
   Addressing issues with android studio not using SQLite. Imported database library and successfully implemented graphs.

Minutes for meeting on **29/04/24** at **15:15**

The meeting took place in person in **CB 4.17**

Attendees - Aleena, Felix, Matthew K, Matthew M, Sebastian, Vlad, Zhiming, Yusuf

Objective: Recap and summarise sprint 2

Points discussed:

1. Trophy
   Auxiliary interface functionality, if the user has met their goal then they will receive a trophy as a motivation and +1 to the total trophy count which they can view when they first open the app.
2. Report: Reflection
   Let those who worked on parts of the software such as the database to write their account on the development process.
3. Additional graphs
   Line graph to indicate the user's current progress in comparison to their initial goal.
4. Organising an extra meeting
   Last push before the deadline, a supplementary meeting before Friday in order to bring this project to a conclusion.

Minutes for meeting on **01/05/24** at **14:15**

The meeting took place in person in **Library**

Attendees - Aleena, Felix, Matthew K, Matthew M, Sebastian, Vlad, Zhiming, Yusuf

Objective: Tie up loose ends

Points discussed:

1. Report

   Reflection part reconfirmation due to overlapping information, polishing previous sections.

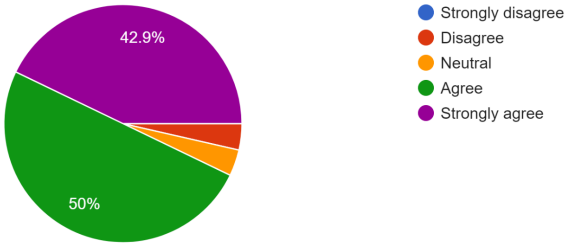
Appendix B Evidence of Primary Research

Questionnaire Results

Link to the questionnaire: https://forms.gle/sjpYhGvHBfPaHBk69

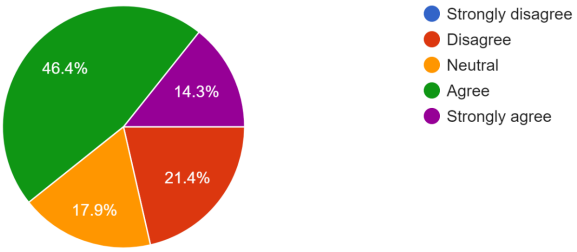
I spend a significant amount of time on screens before going to bed.

28 responses



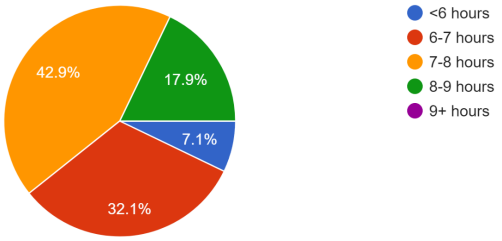
My screen time affects the quality of my sleep.

28 responses



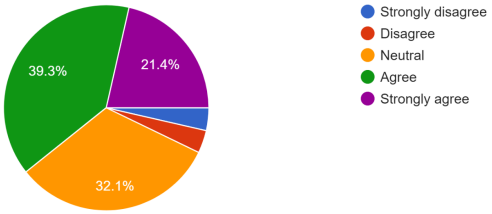
How many hours of sleep do you get on average?

28 responses



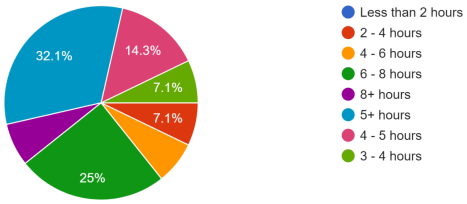
I believe reducing screen time before bed could improve my sleep quality.

28 responses



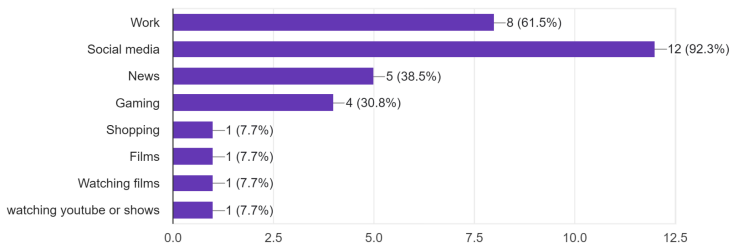
How many hours a day do you spend on screens? (TV, Phone, Laptop, Gaming consoles)

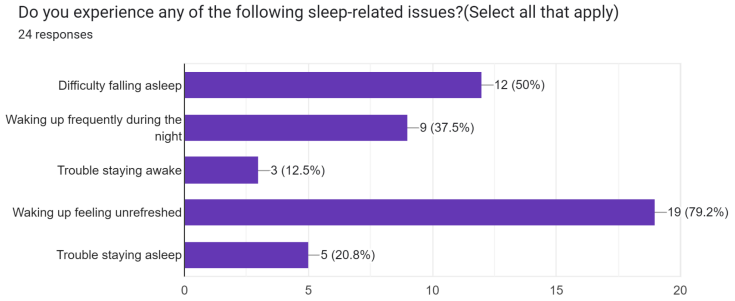
28 responses

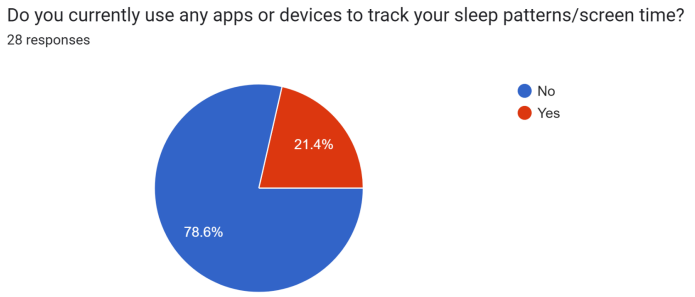


What do you spend your screen time on?

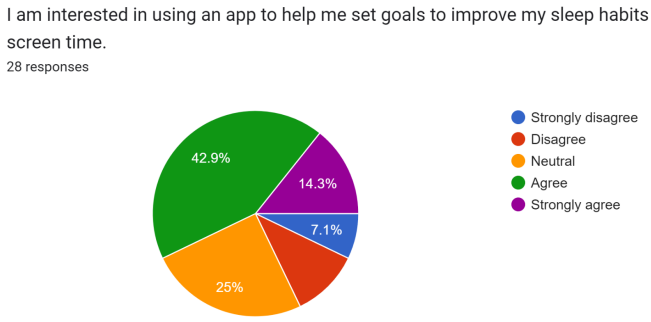
13 responses







The following question was what features do you like in particular to which the responses were along the lines of summarised statistics, average screen time hours and suggestion on time to go to sleep.



Interview Transcripts

Interview 1

Interviewer: Product Owner

Respondent: Student (Anonymous)

Date: 20/03/204

Product Owner: Do you believe that using your phone negatively impacts your sleep?
Student: Definitely, I believe that if I could get off my phone earlier I'd be able to fall asleep earlier and easier. Especially because I don't use my phone for work or uni, it's more for social media.
Product Owner: I see. Do you think you sleep better on nights where you didn't use a phone before going to bed?
Student: I think so. A few times I've tried reading beforehand, and I have slept better. I want to stick to sleeping without using my phone right before… but it's really tempting to go back on my phone. I need some kind of incentive to not go back on it.
Product Owner: Our idea is an app that would take in your sleeping hours and screen time, and allow you to set goals for each. Would this be an app you're interested in using?
Student: Yes, it would be interesting to see what my sleep and screen time patterns are like. It may help me plan my time better. I also really want to cut down on my screen time.

Interview 2

Interviewer: Product Owner

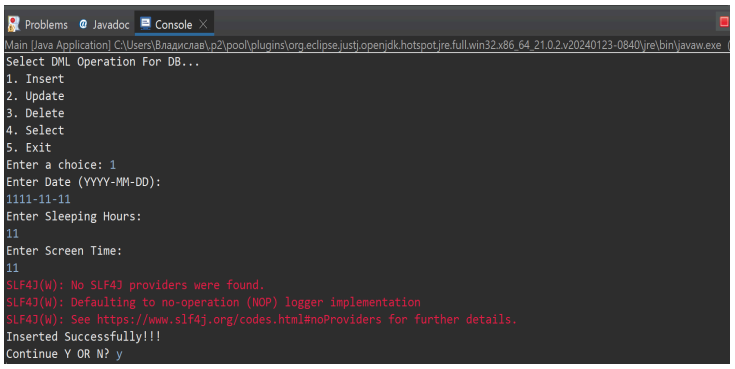
Respondent: Student (Anonymous)

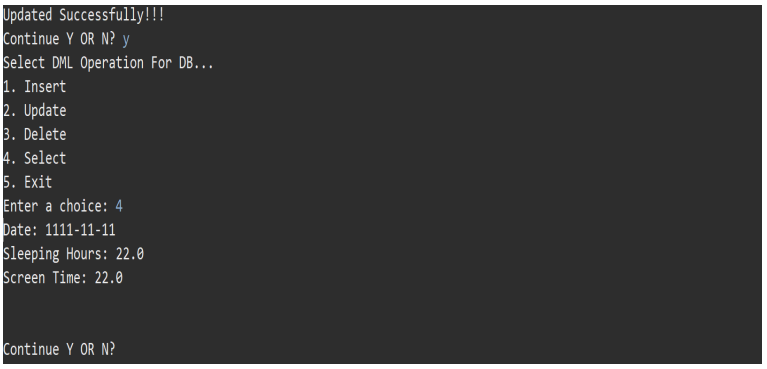
Date: 11/04/2024

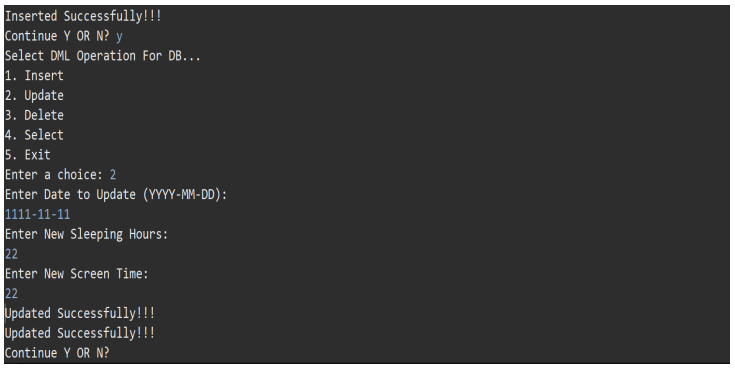
Product Owner: What do you think would make this app better?
Student: I think the app would be more helpful if it automatically locked down apps on the phone when it's bedtime. Current apps give you the option to ignore it, which makes it really easy to just keep using those apps. Or you could use notifications or banners that appear at the top of the screen if you are on a phone after a certain time past bedtime. It should be on there constantly so that I can't swipe it away.

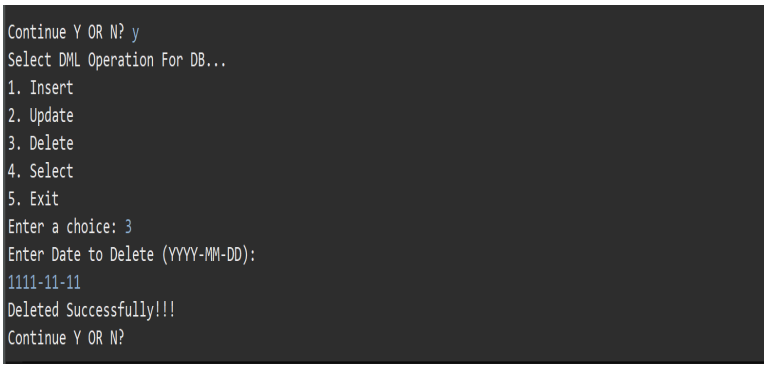
Product Owner: But what if you needed to use something on the screen that's now being covered by the banner? If you're unable to swipe it away, it might hinder the usage of other apps or features on your phone.
Student: That's true. I do think it would be nice to have an incentive to actually reach goals. Sort of like the studying app that grows a tree and kills it if you don't study.
Product Owner: Like a reward system?
Student: Yes, exactly.
Product Owner: Okay. Another feature we're thinking about is displaying your data in the form of a graph when you open up the app. Would you like that?
Student: I don't think it's a main feature, but it would be nice. It'll make it easier for me to see and understand my data and trends in my sleep or screen time. It might make the main page look nicer, too.

Appendix C Design/Testing Extra Evidence

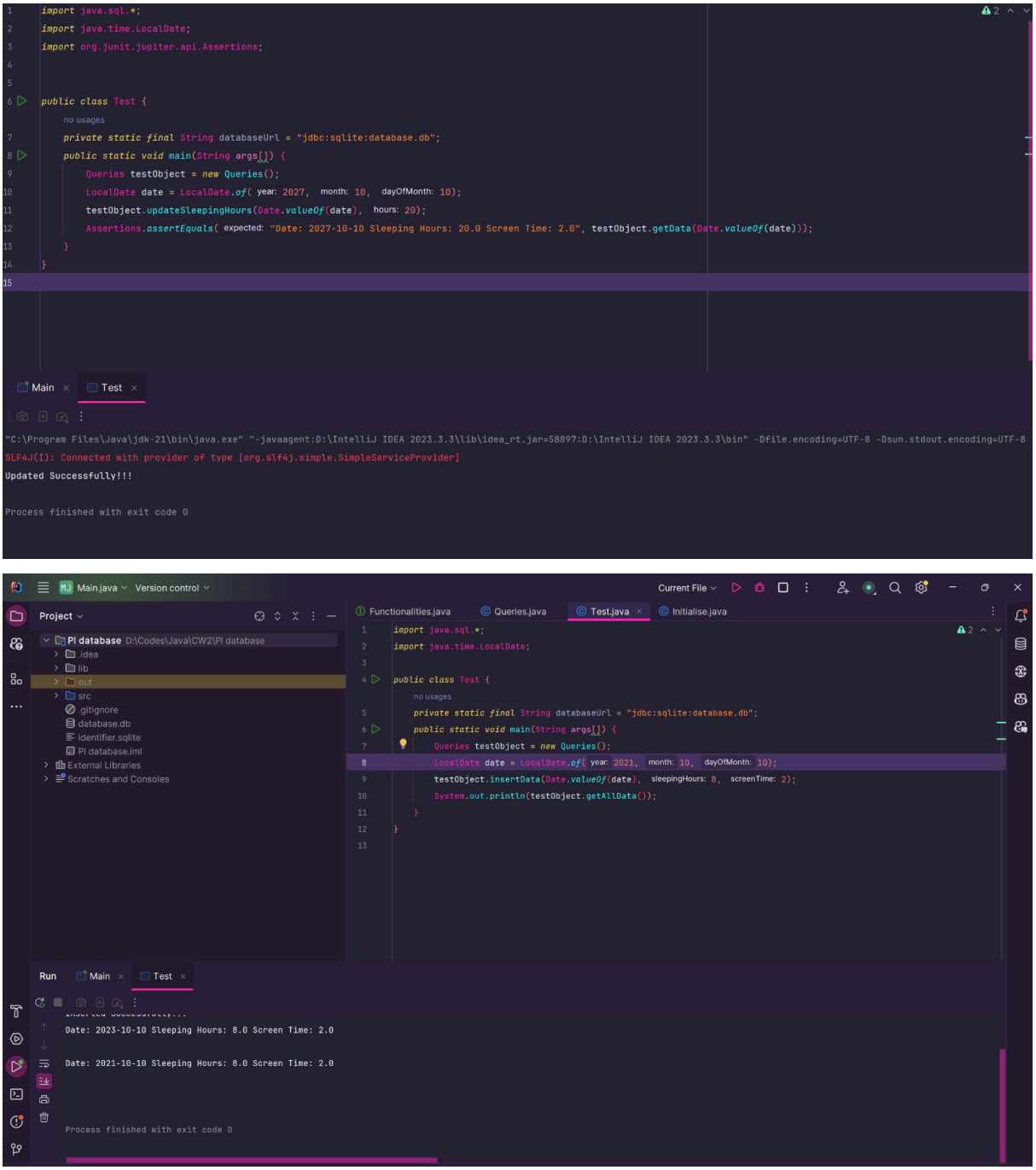








IO Database Figure

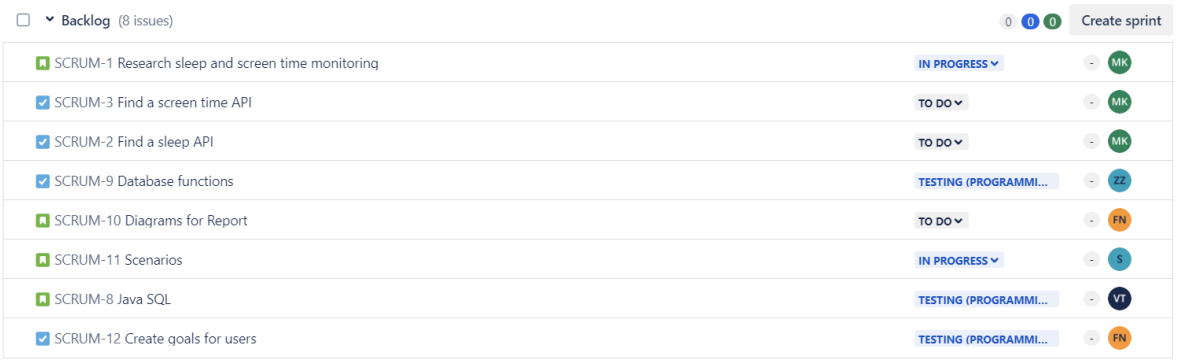

import java.sql.*;
import java.time.LocalDate;
import org.junit.jupiter.api.Assertions;
public class Test {
no usages
private static final String databaseUrl = "jdbc:sqlite:database.db";
public static void main(String args[]) {
Queries testObject = new Queries();
LocalDate date = LocalDate.of(year: 2027, month: 10, dayOfMonth: 10);
testObject.updateSleepingHours(Date.valueOf(date), hours: 20);
Assertions.assertEquals(expected: "Date: 2027-10-10 Sleeping Hours: 20.0 Screen Time: 2.0", testObject.getData(Date.valueOf(date)));
}
}
Main
Test
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:D:\IntelliJ IDEA 2023.3.3\lib\idea_rt.jar=58897:D:\IntelliJ IDEA 2023.3.3\bin" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8
SLF4J(I): Connected with provider of type [org.slf4j.simple.SimpleServiceProvider]
Updated Successfully!!!
Process finished with exit code 0

Main.java
Version control
Current File
Project
PI database D:\Codes\Java\CW2\PI database
.idea
lib
out
src
.gitignore
database.db
identifier.sqlite
PI database.iml
External Libraries
Scratches and Consoles
Functionalities.java
Queries.java
Test.java
Initialise.java
import java.sql.*;
import java.time.LocalDate;
public class Test {
no usages
private static final String databaseUrl = "jdbc:sqlite:database.db";
public static void main(String args[]) {
Queries testObject = new Queries();
LocalDate date = LocalDate.of(year: 2021, month: 10, dayOfMonth: 10);
testObject.insertData(Date.valueOf(date), sleepingHours: 8, screenTime: 2);
System.out.println(testObject.getAllData());
}
}
Run
Main
Test
Date: 2023-10-10 Sleeping Hours: 8.0 Screen Time: 2.0
Date: 2021-10-10 Sleeping Hours: 8.0 Screen Time: 2.0
Process finished with exit code 0

Figure 6

Appendix D Project Backlogs

Sprint 1 Backlog



Sprint 2 Backlog

SCRUM Sprint 2

☐		SCRUM-13	UI (goals) ask user goals quiz for users with android app	DONE	SCRUM Sprint 2
☐		SCRUM-14	Scenarios, Average statistics and checking goals	DONE	SCRUM Sprint 2
☐		SCRUM-15	Notifications	TO DO	SCRUM Sprint 2
☐		SCRUM-16	Background research	DONE	SCRUM Sprint 2
☐		SCRUM-17	graph representation	DONE	SCRUM Sprint 2
☐		SCRUM-18	Class diagrams	DONE	SCRUM Sprint 2