UNIVERSITY
OF OULU

FACULTY OF INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING

Ville Kylmämaa
Janne Kerola

MOBILE-BASED STUDY SUPPORT SYSTEM FOR CONDUCTING ESM STUDIES

Bachelor's Thesis
Degree Programme in Computer Science and Engineering
May 2022

Kylmämaa V., Kerola J. (2022) Mobile-Based Study Support System for Conducting ESM Studies. University of Oulu, Degree Programme in Computer Science and Engineering, 50 p.

ABSTRACT

The experience sampling method (ESM) is a widely adopted research method, where subjects are asked to report their thoughts and feelings multiple times a day over a longer study period. Due to a variety of causes, such as changes in life-style, habitat and routine, subjects often display a drop in response rate over the course of the study and sometimes stop responding altogether. Many of these causes are outside the study managers' control. One such cause we are interested in, is the fluctuation of sleep quality over the course of the study. Poor sleep quality could affect the participants' interest-level and memory during the study, causing a drop in the ESM response rate.

This thesis details the design, implementation and testing of our mobile-based study support system for conducting ESM studies. This system entails a mobile application which provides subjects notifications of available questionnaires and a way to conveniently answer multiple questionnaires a day, and a web application to streamline the configuration of the parameters and questions of the study. The thesis further details the design, organization and the conducting of a proof-of-concept study to test the study support system in a realistic scenario. We combine, process and analyze the data gathered via our study support system and commercial sleep measuring devices. We discuss the results of the analysis and present our conclusions. Finally, we discuss possible ways our system can be developed further and present future research topics based on our findings.

Keywords: Experience sampling method, mobile-based study support system, digital sleep tracking, user study

Kylmämaa V., Kerola J. (2022) Diplomityön teko-ohjeet. Oulun yliopisto, Tietotekniikan tutkinto-ohjelma, 50 s.

TIIVISTELMÄ

Kokemusten näytteistysmenetelmä (ESM) on laajasti käyttöönotettu tutkimusmetodi, jossa osallistujat raportoivat ajatuksiaan ja tunteitaan useita kertoja päivässä tutkimuksen aikana. Usein osallistujien vastausmäärät laskevat tutkimuksen aikana tutkimuksien järjestäjien hallitsemattomissa olevien muuttujien takia, kuten muutokset elämäntyylissä, elinympäristössä ja rutiinissa. Yksi tällainen muuttuja on unenlaatu tutkimuksen aikana. Huono unen laatu voi vaikuttaa osallistujien mielenkiintoon ja muistiin negativiisesti, aiheuttaen vastausmäärän laskemista tutkimuksen aikana.

Tämä työ esittelee mobiili-pohjaisen ESM-tutkimustukijärjestelmän suunnittelun, kehityksen ja testausvaiheen. Tukijärjestelmä koostuu mobiilisovelluksesta, joka ilmoittaa käyttäjille saatavilla olevista kyselyistä sekä helpottaa vastausten lähettämistä eteenpäin. Toinen komponentti on verkkosovellus tutkimusten suoraviivaista alustusta varten. Työ esittelee myös testitutkimuksen suunnittelun ja järjestelytyön, sekä tutkimuksen tulokset ja arvionnin. Yhdistämme, käsittelemme ja analysoimme tukijärjestelmämme sekä kaupallisten unimittauslaitteiden kautta saatua dataa. Keskustelemme analyysin tuloksista ja esitämme johtopäätöksemme. Lopuksi keskustelemme järjestelmän mahdollisista jatkokehitysväylistä, sekä annamme osviittaa tuleville tutkimuspoluille.

Avainsanat: ESM-menetelmä, kokemusten näytteistysmenetelmä, puhelin-pohjainen tutkimustukijärjestelmä, digitaalinen unenseuranta, käyttäjätutkimus

TABLE OF CONTENTS

FOREWORD

This Bachelor's thesis project was completed during the spring semester 2022 at the University of Oulu. We are grateful to our friends, family, and our supervisor Aku Visuri, for supporting us during the project. Special thanks to Aku Visuri for providing us with an interesting project topic with expandable scale and free choice of implementation. This project enabled us to learn and showcase our various skills in software engineering, design and data analysis.

Oulu, May 18th, 2022

Ville Kylmämaa
Janne Kerola

LIST OF ABBREVIATIONS AND SYMBOLS

ESM	experience sampling method
API	application programming interface
REST	representational state transfer
UI	user interface
SPA	single-page application

1. INTRODUCTION

The experience sampling method (ESM) is a widely used and adapted method for sampling self-reported user experiences in various studies. By proactively prompting the user for input at specified intervals, the method is designed to reduce the loss and misrepresentation of data caused by the subject recalling past events with inaccuracy. Researchers interested in human behaviour may use the method to gather data from activities of daily living produced in a natural, non-laboratory environment. [1]

Wide-spread adoption of smart-phones has allowed researchers to conduct highly automated, remote and non-supervised studies using the method. Electronic questionnaire platforms allow for precise measurement of elapsed time, response rate and other related statistics. Researchers can utilize the built-in sensors and geo-location functionality of smart-phones to gain further insight into the environment and its effect on human behaviour. [1]

ESM studies generally note a drop in the number responses and active participation from users as time goes on. The causes are often uncontrollable by study managers, such as changes in the subjects' life, or general disinterest and forgetfulness. In the worst-case scenario, users stop responding completely. [1]

ESM is particularly useful for studying daily fluctuations in subjective health, well-being, behaviour and similar issues. Sleep, as an activity, generally occurs every night, and its duration and quality have notable health and well-being related consequences for the following days. We are particularly interested in sleep and its effects on response frequency in ESM studies. [2].

Our thesis aims to:

1. Develop a mobile-based support system for conducting ESM studies.
2. Design a proof-of-concept ESM study to test the system.
3. Organize and assess user response frequency and retention rate over a period of 14 days.
4. Investigate possible correlations and underlying causes.

The main contributions of this thesis are the design and implementation of a mobile-based study support system adaptable for different kinds of ESM studies, the design and conducting of the proof-of-concept study on the effects of sleep on response rate, and the evaluation of the study results. Section 3 discusses the design of the system and the proof-of-concept study. Section 4 gives an overview of the development process and key factors of our implementation. Section 5 details the study conducted with the developed system. Section 6 discusses the achievements and limitations of our system, possible avenues for future work, and the results of the study we conducted. Finally, in Section 7, we give our thoughts on future directions for research and potential research gaps.

2. RELATED WORK

There have been many sleep studies conducted utilizing the ESM with various differing protocols.

Takano et al. (2014) studied the relationships between repetitive thought, mood and sleep problems. ESM was utilized to record the thought content and mood of the participants 8 times a day at random intervals for a sampling period of 1 week. To encourage continuous participation, the participants received financial compensation and a report of their personal results at the end of the study. 6 out of 49 participants were excluded and the mean response rate was 78.9%. [3]

Das-Friebel et al. (2020) studied the effect of bedtime social media use on sleep and well-being. ESM was utilized to record social media use, and sleep duration and satisfaction in a single questionnaire in the morning, and momentary affective well-being 5 times later during the day for a sampling period of 2 weeks. To encourage continuous response rate, the participants could receive £2.50 for each day of participation. The participants had to respond to at least 67% of all the prompts to receive the full compensation amount. Out of the 116 participants who were able to participate in the study, the data of 4 participants was excluded due to low participation rate, and 4 other participants - one reported the study to be too tedious and intrusive - quit the study after it had commenced. [4]

Sznitman et al. (2020) studied the effect of cannabis use and sleep start time on sleep continuity. ESM was utilized to record cannabis use and sleep indicators 3 times a day for a sampling period of 7 days. A lottery for 12 vouchers worth 500 Israeli new shekels was held to encourage participation. Despite the monetary encouragement and the short study duration, 84 out of 138 participants were excluded due to answering less than 30% of all reports. [5]

Kammerer et al. (2021) studied the effect of sleep on persecutory symptoms in patients with psychosis and prevailing delusions. ESM was utilized to record subjective sleep quality in a single questionnaire in the morning, and momentary assessment of affect and persecutory symptoms 10 times later during the day for a sampling period of 6 days. The participants received a compensation of 40€ for completing the study and an extra 5€ for having higher than 70% response rate. The patients in the study had an average response rate of 71.72% and the healthy controls 74.20%. [6]

Block et al. (2019) studied the effect of anticipatory stress, and openness and engagement on perceived sleep quality. ESM was utilized through 6 questionnaires a day to record sleep quality, anticipatory stress, and openness and engagement for a sampling period of 1 week. Only 5 out of 290 participants were excluded due to ESM participation rate of less than 50%, and overall, 93.16% of all the questionnaires were completed. This is a peculiarly high participation compliance compared to other similar studies. The participants were diagnosed with either major depressive disorder or social phobia and the study claims that psychiatric populations generally have good compliance rates. To prevent overburdening the participants, some questionnaires had only one question. [7]

Outside of just sleep studies, Vachon et al. (2019) conducted a meta-analysis on how different design characteristics were associated with participant retention and compliance in studies investigating major depressive disorder, bipolar disorder, and psychotic disorder. The higher sampling frequencies were associated with lower participant compliance and retention, but interestingly study duration did not have a significant association with either. [8]

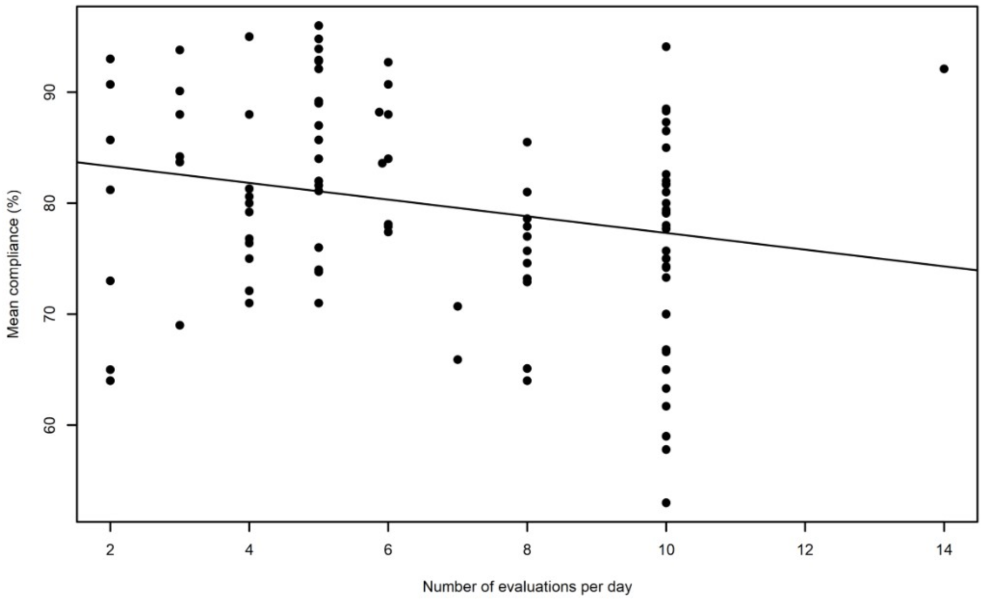


Figure 1. Vachon et al. (2019), graphical representation of the relationship between the compliance of experience sampling method studies and the frequency of daily self-evaluations. Image used under the terms of the Creative Commons Attribution License. [8]

Participant compliance varies wildly between the referenced ESM sleep studies and the studies in the referenced meta-analysis. Some studies manage higher compliance with a sampling frequency of 10 times a day than other studies with a sampling frequency as low as 2 times a day.

However, the study populations, the questionnaires, the amount of compensation incentives, and other study methods also vary to such an extent that the studies are not directly comparable. For example, some studies have more questions per questionnaire than others. Some studies consider a questionnaire unanswered if the participant did not respond in 15 minutes while in other studies the length of this answering period was as long as 2.5 hours. Some studies offer monetary compensation for completing the study with high enough response rate while some offer no compensation of any kind. The number of questions per questionnaire and the complexity of the questions asked also vary between the studies. All this makes conducting meta-analyses problematic which makes individual studies specifically studying participant retention useful.

3. DESIGN

3.1. ESM Application Design

To run an ESM study, we require the following:

1. Consistent, always available method of prompt delivery
2. Mechanism for collecting user responses and related data
3. Safe and reliable storage for the collected data

We considered different ways of implementing these requirements, with varying levels of quality. We deliberated their effectiveness contrasted with the difficulty of implementation. For the delivery method of the questionnaires, we considered either utilizing timed emails as the easy option, or developing a mobile application as the proper software engineering option (Table 1).

Delivery Method	Pros	Cons
Timed emails	Easy to implement.	Limited monitoring capabilities. Notifications reliant on the email application. Vulnerable to spam-filters.
Mobile application	More customizable, full control over notifications to draw attention. Allows monitoring of events before, during and after prompt triggers.	Much higher work load.

Table 1. Considered prompt delivery methods

We also considered using existing web survey platforms for collecting and storing the response data and compared them with either utilizing a commercially available cloud database service or building our own fully custom back-end from scratch (Table 2).

Management Method	Pros	Cons
Online survey (Webropol, Zef, etc.)	Automatic analysis of responses. Easy management of questions and responses.	Limited collection of data points during prompt.
Available cloud database service (Amazon Web Service, Firebase, etc.)	Quicker to setup than custom web server. Ready-made analytics about usage and traffic. Ready-made solutions for many security issues.	Medium amount of work Not as customizable as a custom web server.
Fully custom web back-end	Completely customizable. Complete structural control of the data.	Requires the most work.

Table 2. Considered data management methods

In the end, we chose the combination of a mobile application connected to a commercially available cloud database service. Between these choices, we valued customizability, which gave us the opportunity to create something unique. The development of the mobile application would be our main goal and contribution in this thesis. With the majority of the time allocated for implementation going in to the development of the mobile application, we chose to utilize a commercial cloud database service over developing a fully custom back-end.

We chose to develop the mobile application with Flutter [9] because it allows for cross-platform development for both Android and iOS from a single codebase. This means that we do not have to develop Android and iOS applications separately. The application running on both operating systems would be very beneficial, as subjects with Android and iOS-based devices can be included as participants. For our cloud database, the choice was Cloud Firestore through Google's Firebase service, which when compared to a completely custom web server built from scratch would save a lot of time and effort. These choices are further discussed in the Technology Choices section 4.2.

3.1.1. Minimum Requirement Specifications

We define the minimum requirement specifications of our mobile application as the following:

1. In-application questionnaires with a custom user interface (UI).
2. Delivery of a notification when a questionnaire should be answered. The notification can be tapped to open the questionnaire.

3. Questionnaires are only available during their intended time period, and each questionnaire can only be answered once.
4. Responses to the questionnaires and their related timestamps are stored in a cloud database.
5. The user is kept up to date about the progression of the study with different main views for the start, in progress, and end of the study.
6. The user can choose the start time of the daily questionnaires.

3.1.2. Additional Feature Specifications

If time constraints allow, we aim to also implement the following:

1. The user can change the start time of the daily questionnaires to counteract changes in their sleep schedule.
2. Study parameters and questions are fetched from the cloud database in order to make them configurable instead of hard-coded.
3. Web dashboard to easily configure the study parameters and questions.
4. The application runs on both Android and iOS.

3.2. Study Design

We set out to design a proof-of-concept study to validate the functionality of our mobile-based system. The core focus of this study is on the effect of sleep on the response rate and response quantity over time when employing the experience sampling method. By providing subjects with sleep measuring devices and consistently scheduling automated prompts to answer, we should also able to study the subjects' perceived sleep quality when compared to the measured amount.

We considered the number of scheduled prompts to be a critical factor. Too many prompts may cause subjects to tire out quickly, while too few may lead to an insufficient amount data for further analysis. The content and delivery method of these prompts is also likely to affect subjects' perception and responses to the study.

Initially we considered open text input type questions. Open questions allow for a wide variety of answers, however this form of response cannot easily be analyzed via statistical analysis. Open text field questions complicate the processing of the data as high participant counts quickly produce a high number of unique answers. Manual revision of hundreds or even thousands of responses is out of scope for this work. Text-based answers also require more time and effort from the participant to answer which might result in lower response rates. These open text field questions could be left optional, however participants may feel an obligation to answer them or even guilt for leaving them empty. Frustration over answering the same questions every day or multiple times a day could also lead to a decrease in the quality of the answers over time. The advantage of open questions is that they can enable the participants to give more detailed and nuanced answers. However, we deemed this level of granularity unnecessary for this study and not worth the possible disadvantages.

In comparison, scales provide many useful benefits for smartphone-based study of this type. Collecting and analyzing scale values is perfectly suited for statistical analysis, with data points becoming easily graph-able for visual representation. In addition, the mechanism for responding does not change even if we change the parameters of the scale, e.x. "Sad to Happy" functions similarly to "1 to 100". We are able to change the effort requirement by changing the question format, without affecting mechanical difficulty.

We also considered allowing the subjects to skip questions they do not want to answer. This could tempt the subjects to skip questions too easily, resulting in a lower response rate. However, not being able to skip questions may sometimes result in the participant not completing a questionnaire at all, which they would have otherwise completed at least partially. Being forced to answer questions that the participant might at the time wish to skip could also lead to lower quality answers. Furthermore, partially answered questionnaires also give us the opportunity to evaluate the subjects' tendencies to skip certain questions. Therefore, we decided to allow skipping questions in this study.

We chose to frame our questions so that there is a clear focus on momentary experiences rather than reports across a period of time, due to subjects' memories being relative. Specific wording was used to query the subjects current emotional state and thoughts, which should in theory result in a more accurate overview timeline. In other words, we preferred the questions worded as "Describe your current mood" over "Describe your mood since the last prompt".

After prolonged discussion, our questionnaire design consists of repeatable 5-point Likert-scale questions. The number of questions directly correlates with the aspects we wanted to focus on, namely the correlation between sleep quality and the subjects' ability to focus during the day. We discussed the potential dangers of leading statements vs. open ended questions (('I feel energetic' vs. 'How energetic do you feel currently?') and chose the latter to prevent external misdirection with the results. We also believe the Likert-scale to be an appropriate format, as the questions require some introspection to answer yet are mechanically convenient to respond to.

The questions for the questionnaires were chosen by considering different aspects of the human ability to focus. We consider the following concepts critical aspects of the subjects focusing ability for evaluation: **Vitality, Mood, Health,** and **Effective tasking.**

Vitality is the subjects' perception of their own energy level at different points of the day. High energy subjects are expected to have better capability to focus than tired, low energy subjects. By keeping track of self-reported energy levels, we are able to compare perceived vitality in comparison to measured sleep quality.

- How energetic do you feel currently?

Mood should reflect the subjects' emotional state at the time of the prompt. By prompting the user to balance their current emotional state between two opposites, such as Happy-Sad or Stressed-Relaxed, we are able to link the quality of sleep with possible resulting emotional states. We used a circumplex model of affect to choose the mood pairings in our question array. [10]

- Evaluate your current mood:
  - on a scale from "Sad" to "Happy"
  - on a scale from "Bored" to "Alert"
  - on a scale from "Nervous" to "Calm"
  - on a scale from "Stressed" to "Relaxed"

Health is a supporting aspect for vitality. By considering the subjects' general health, we are able to separate low vitality days caused by possible injury, ailment or disease. We opted not to ask specifics in order to reduce the burden.

- How healthy do you feel currently?

Effective tasking is a direct consequence of focusing ability. Links between sleep quality and effective tasking can be established by monitoring these attributes.

- Evaluate your effectiveness in your current task (work, school, hobby, etc.).

In addition, the first questionnaire of the day would have an additional question about the sleep quality of the last night. This provided us with subjective sleep quality data in order to not rely solely on the sleep measuring devices.

- How well did you sleep last night?

3.2.1. Study Duration and Prompt Frequency

For the study duration, we discussed either one or two weeks. We considered that one week would make it easier to find participants and we could possibly run two or more batches in the allocated time. However, one of the goals of the study was to see if response rate would decrease over the duration of the study, and we considered that one week would not be sufficiently long to study this. The Vachon et al. (2019) meta-analysis found that the study duration was not associated with the participant compliance or retention rate [8]. We hope to observe this in our own study.

With the sampling frequency in ESM studies generally ranging between 2 and 10 times a day [8], we decided that five questionnaires per day was a reasonable middle-ground amount to balance between getting enough data and overburdening the participants. The five questionnaires were set three hours apart to divide them evenly over a 12-hour period. We considered a 30-minute response window to be enough time to allow for flexibility, while still being current enough. After 30 minutes the questionnaire would expire and the response considered unanswered if the participant didn't answer it.

3.2.2. Role of the Mobile Application in the Study

The application was designed to deliver a notification at the time when the participant was supposed to answer a questionnaire. The notifications could be clicked to open the application, or the participant could open the application through their own means to find the questionnaire. After answering a questionnaire, the application displays the time the next questionnaire would become available.

Because people have different sleep schedules, we decided to allow the participants to choose the start time of the daily questionnaire cycle. For standardization, we instructed the participants to choose a start time which would be 1 hour from their usual wake-up time. The 12-hour period during which all the questionnaires would be prompted would then start at the chosen time.

Data was collected from the answers to the questionnaires, the time the questionnaire was prompted, the time the participant opened the questionnaire, the time the participant completed the questionnaire, and the total amount of questionnaires answered by the participant. The response and timestamp data were sent to our back-end database for storage and analysis. The main interests for the analysis were the percentage of total questionnaires answered by each participant and whether or not the amount of answered questionnaires decreased over time. We were also interested in the correlation between sleep amount and the answers to the questions about vitality, health, mood and effective tasking.

3.3. Sleep Measuring Devices

Sleep measuring devices were incorporated in order to have an objective source of sleep data from the participants. Two types of sleep measuring devices were available: Fitbit Versa 3 wristwatch and Withings Sleep Analyzer under-mattress sleep tracker.

3.3.1. Withings Sleep Analyzer

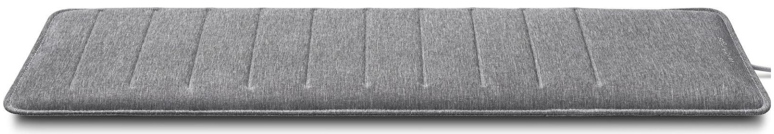


Figure 2. Withings Sleep Analyzer, a sleep measuring device used in our study. [11]

The Withings Sleep Analyzer is placed under the mattress and it measures sleep cycles and sleep time by measuring heart rate, respiratory rate and body movements via pneumatic sensor, and snoring and cessations in breathing via sound sensor [12]. The sleep data from the device could be examined with the Withings Health Mate mobile application [13]. However, in order to export the collected data, the participants are required to do it via the Withings Online Dashboard, or it could be accessed via the Withings Developer API [14].

3.3.2. Fitbit Versa 3 Fitness Tracker

Figure 3. Fitbit Versa 3, a sleep measuring device used in our study. [15]

The Fitbit Versa 3 is worn on the wrist, where it measures sleep stages and score passively via a motion sensor, heart rate sensor and more [12]. The device requires pairing to a smartphone device via Bluetooth before collecting data. The FitBit application is available on the Google Play Store [16], requiring users to create an account to pair new devices. After pairing, the device is ready for use and collects data automatically. In order to export collected data from the service, users are required to log in to the FitBit web portal. The portal allows users to choose the time-span and data category they wish to export. The data is also available through the FitBit Developer API [17].

Device	**Withings Sleep Analyzer**	**Fitbit Versa 3 Fitness Tracker**
Timestamps	X	X
Total sleep amount		X
Light sleep	X	X
Deep sleep	X	X
REM sleep	X	X
Snoring	X	
Breathing cessations	X	
Heart Rate	X	X

Table 3. Comparison of the data points tracked by the devices.

4. IMPLEMENTATION

4.1. Implementation Process

We chose to utilize a ongoing agile development process somewhat similar to Scrum, with weekly meetings between members to discuss potential issues arising during development. We also scheduled progress report meetings with our project supervisor.

We chose to operate on a merge-request based workflow, where new features, fixes and other tasks began development in a new branch before being submitted to be reviewed by other developers. Upon passing the review, the changes were merged directly into the main branch. We outlined the key tasks in the project Kanban-board according to our design parameters and created a new empty Flutter [9] project as the base. We utilized the Kanban-board and the meetings to coordinate between tasks and assign reviewing duties between members.

In order to encourage future development of the project, we left instructions on how future developers could easily adopt the repository by supplementing their own API keys. We documented the code and implemented many quality control mechanism to ease the adoption process, as detailed in the next section.

4.2. Technology Choices

We chose to develop our mobile application with Flutter. This decision was based on our previous experience with the framework and Flutter's cross-platform capabilities in order to develop simultaneously for Android and iOS. Supporting both Android and iOS allowed us to search for participants from larger pool for our study. Higher performance would likely be possible with native programming, for example using Kotlin [18] to develop for Android. However, our application is not very resource intensive and questionnaires rendering slightly faster will not make a difference to the end user.

For our DevOps platform, we chose GitLab[19] due to being familiar with its robust CI/CD pipeline system, as well as its built-in features in project management. Utilizing the issue tracking and management features, we were able to easily divide and choose appropriate tasks during development, as well as organize tasks by priority. Thanks to the platform's messaging system, we were able to discuss changes and design decisions without direct contact, leaving behind a trail of our thoughts which we utilized in writing this thesis.

In order to utilize automatic changelog generation and semantic versioning in our build pipelines, we chose to name our merge-request following the conventional commits [20] convention specification. Conventional commits offers commit message templates for common commit types, such as new features, bug-fixes and documentation changes. Automation tools are able to parse these "commit types" into sectioned, stylized changelogs or release notes for human consumption according to predefined style configurations. For example, a merge request titled "docs: add section about conventional commits" would get parsed into a change log with a separate documentation section, with the message "add section about conventional commits" as a bullet point. As we utilized squashing of commits upon being merged, individual commit style would not make a difference in the end result.

Conventional commits also allowed us to use automatic semantic versioning [21] to manage software updates. We chose semantic-release [22] as the parsing tool to automatically determine when a version "bump" was necessary, triggering further pipelines to build, upload and release our application.

To ease the development process, we implemented robust test, build and deploy pipelines for the project. Any new merge request would have to pass the testing pipeline before being eligible for merging. Any failed test would result in a merge-block, until the failing test was passed. When code was deemed appropriate for a merge, automation tools would parse the title and select an appropriate versioning bump before triggering the build stage. If no version bump was detected, the build stage would be skipped and the subsequent upload stage would fail with built no files to upload. If the upload stage succeeded, the final deployment stage would trigger, publishing a neat GitLab release with automatically generated release notes for testing.

To further enforce systematic development and quality control, we implemented multiple pre-commit [23] hooks to enforce unified stylistic formatting and enact automatic static code analysis. Pre-commit hooks function by executing individual pre-defined analysis tools on the currently modified files before allowing or preventing committing based on the result. Common hooks used in our project include the Prettier [24] and Flutter format [25] hooks for automatic formatting, and the Flutter Analyze [26] hook to catch potential bugs via static code analysis.

For our database, we chose Firebase Cloud Firestore [27] to store questionnaire questions and responses, study parameters, and timestamps. Cloud Firestore is a cloud database that is scalable, quick to setup, and much less work than creating a custom back-end solution. Cloud Firestore allows syncing of data across devices in real-time without having to program custom listeners. This syncing could be used for example to quickly and easily modify questions even during the study. We were able to utilize Cloud Firestore fully for free due to the small scope of our study. We considered the advantages presented to be worth it over developing our own backend server database. Firebase also offers another option for a database: Realtime Database. We chose Cloud Firestore service over the Realtime database due to Firebase's promise of Firestore having faster queries and better scaling [28].

The web application was developed with React[29] as the JavaScript library of choice, because of the ease it offers for creating single-page applications (SPA), or in summary, web applications which update elements without having to refresh the page. Both of us also had previous experience with React.

We also utilized many packages provided by the open-source Flutter and Dart developer community at Pub.dev [30].

- **Hive** [31] is a key-value based database replacement package, which we chose for its ease of implementation. Thanks to hive, we are able to store persistent app-specific settings on the device.

- **device_info_plus** [32] is a library for querying information about the current device. We utilize the device identifier to match events in our collected data before anonymization.

- **Flutter Workmanager** [33] is a library used for executing headless dart code in the background. This was used to schedule local notifications for each day and manage cancellations upon a setting wipe.
- **flutter_local_notifications** [34] is a library for creating device native notifications. These are utilized for prompting users of questionnaire availability.
- **app_settings** [35] is a package used to manage app-specific permission on android and iOS. Due to system design choices, we required permission to ignore battery optimisation on android devices in order for the notifications to work as intended.

For a full list of utilized packages and version numbers, please see the source repository pubspec.yaml file.[1]

4.3. System Architecture

Our project consists of three parts: the mobile application, the web dashboard, and the Cloud Firestore database. The cloud database uses a NoSQL data model and any communication with it is done via its representational state transfer (REST) API. The mobile application and the web dashboard both communicate with the cloud database. They do not communicate directly with each other as all the synchronization happens through the cloud database.

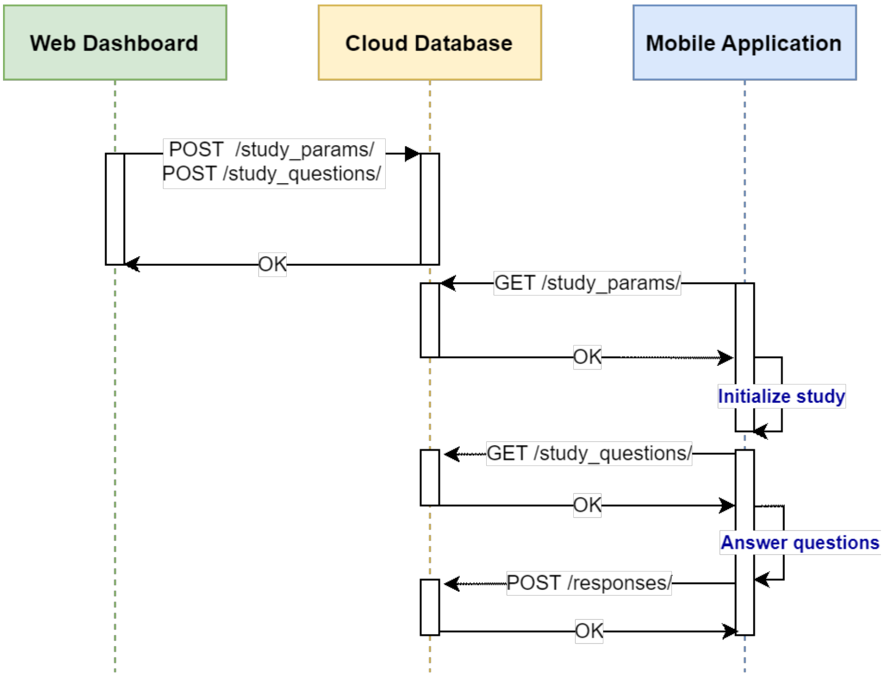


Figure 4. Diagram of component interactions in the implementation.

[1] https://gitlab.com/esm-padawans/esm-mobile/-/blob/main/pubspec.yaml

4.3.1. Operation Mechanism

When the mobile application is first run, it will fetch the study parameters (Table 4) from the cloud database. These parameters include the duration of the study in days, the amount of questionnaires per day, the time between each questionnaire in minutes, and the time after which a questionnaire will time out. These parameters are then stored locally and will not update even if the values are changed in the cloud database. This is to prevent mistakes such as accidentally changing one parameter wrong and disrupting the study for all currently participating, while also enabling easier management of groups with different study parameters.

The scheduling of questionnaires is handled completely locally based on the configured study parameters. The application calculates whether a questionnaire should be shown based on the given daily questionnaire frequency, the amount of time a questionnaire should be available for, and the time between each questionnaire. Therefore, multiple kinds of study windows can be configured by adjusting these parameters, granting flexibility in adapting our application to different kinds of studies.

For our study, we chose to set questionnaire frequency to five prompts per day and the study duration to 14 days. Questionnaires were spaced 180 minutes apart from each other, with each having a 30-minute timeout window in which users can respond to the questionnaire.

Study Parameters	**Default**	**Details**
questionnaireFrequency	5	The number of questionnaire prompts per day.
questionnaireTimeOutPeriod	30	The number of minutes each questionnaire is available before automatically closing.
timeBetween	180	The number of minutes between individual questionnaire prompts.
studyDuration	14	The number of days the study is planned to last.

Table 4. Configurable study parameters. Defaults are set as a fallback if the fetching of the parameters from the Cloud Firestore fails.

When the user launches the application to respond to a questionnaire, the collection of questions is fetched from the cloud database (Table 5). Question definition requires providing a title string variable, for example: "How did you sleep last night?", a label string variable under which the response should be stored as when sent, for example: "sleep", a Boolean variable (true/false) determining whether the question should be a multi-part question or not, and a string variable determining in which daily questionnaires the questionnaire should be shown. For example, a "2" would only be shown in the second questionnaire of each day and "all" would be shown in every questionnaire. Top and side labels for each question can also be provided. Top labels will show on top of the slider and side labels on each side horizontally. Both top and side labels are optional, and you can choose to include either one of them or both.

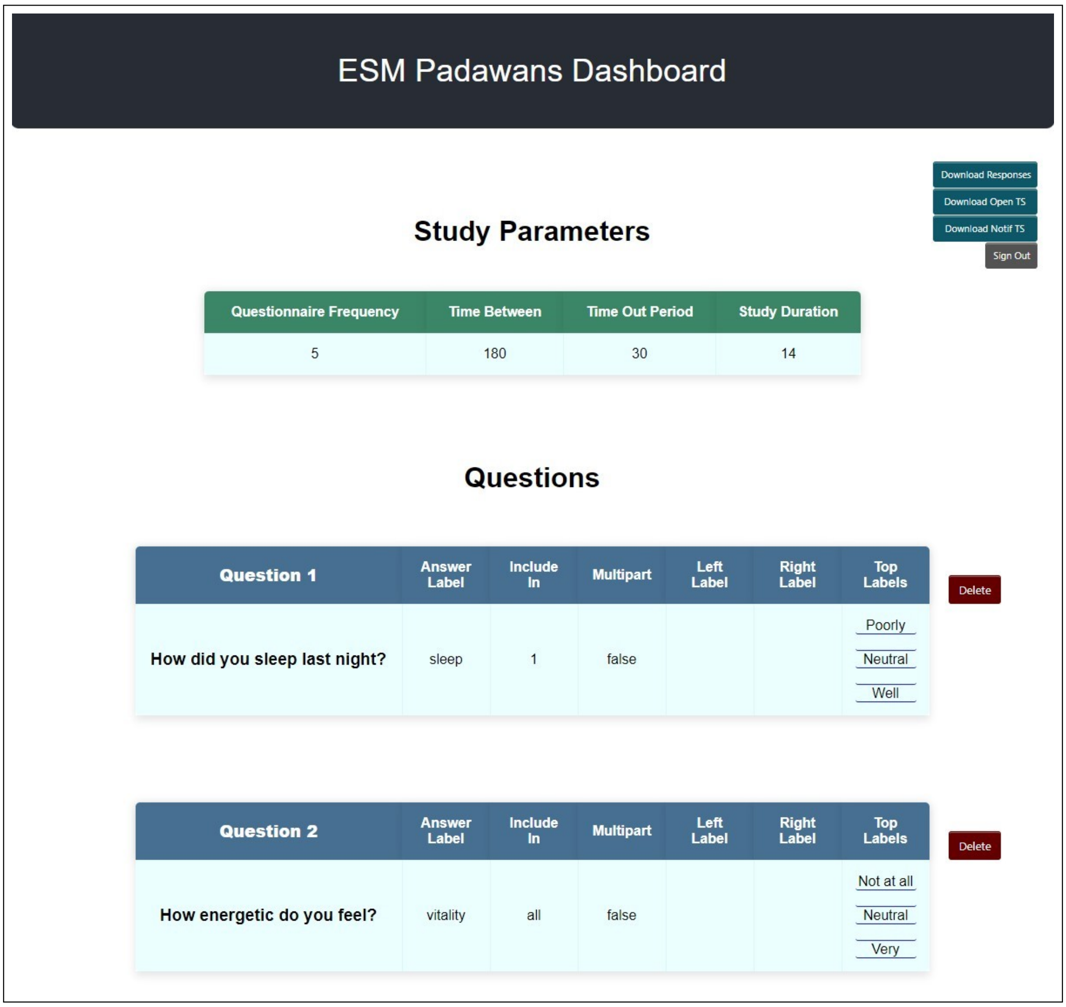
Question parameters	Required	Type	Details
answerLabel	yes	string	The label under which the response value is stored.
includeIn	yes	string	String indicating the index of the questionnaires it should appear in. Use "all" for all.
multipart	yes	boolean	Whether this question should be rendered as a multipart or single view question.
question	yes	string	The actual question displayed to the user. Empty string allowed.
topLabels	no	array of strings	Labels displayed on top of questions pips. Indexes 0-4 indicate left to right. Empty strings allowed.
leftLabel	no	string	Label displayed on the left side of the slider widget.
rightLabel	no	string	Label displayed on the right side of the slider widget.

Table 5. Configurable question parameters.

When the user elects to send their response to a questionnaire, the responses to each question will be stored in the cloud database under the label each question was specified to use. The unique device identifier is sent with the response in order to differentiate which responses belong to the same participant. The index representing which questionnaire of the day the response corresponds to is also sent. Timestamps are sent from the time when a notification was delivered to the user, the time when the user opened the questionnaire and the time when the user elected to send the questionnaire. The timestamps include the unique device identifier and the questionnaire index, which allow us to identify exactly where each questionnaire correlates to in the subjects' expected routine.

The mobile application delivers a notification to the user whenever a questionnaire becomes available. The user can then either tap the notification or open the application directly to view the questionnaire. Unused notifications will be automatically dismissed after the questionnaire times out. The notification functionality is implemented locally in the application, which reduces the amount of API calls to the cloud database, which in turn improves scalability. This also enables further development, potentially allowing the study to continue seamlessly for a user even when the network connection of their device is lost. In future, the responses could also be stored locally until they can be sent when the network connection is regained.

4.3.2. Web Dashboard



ESM Padawans Dashboard

Study Parameters

Questionnaire Frequency	Time Between	Time Out Period	Study Duration
5	180	30	14

Questions

Question 1	Answer Label	Include In	Multipart	Left Label	Right Label	Top Labels
How did you sleep last night?	sleep	1	false			Poorly Neutral Well

Question 2	Answer Label	Include In	Multipart	Left Label	Right Label	Top Labels
How energetic do you feel?	vitality	all	false			Not at all Neutral Very

Figure 5. UI of the web dashboard with which you can configure different kinds of ESM studies for our application.

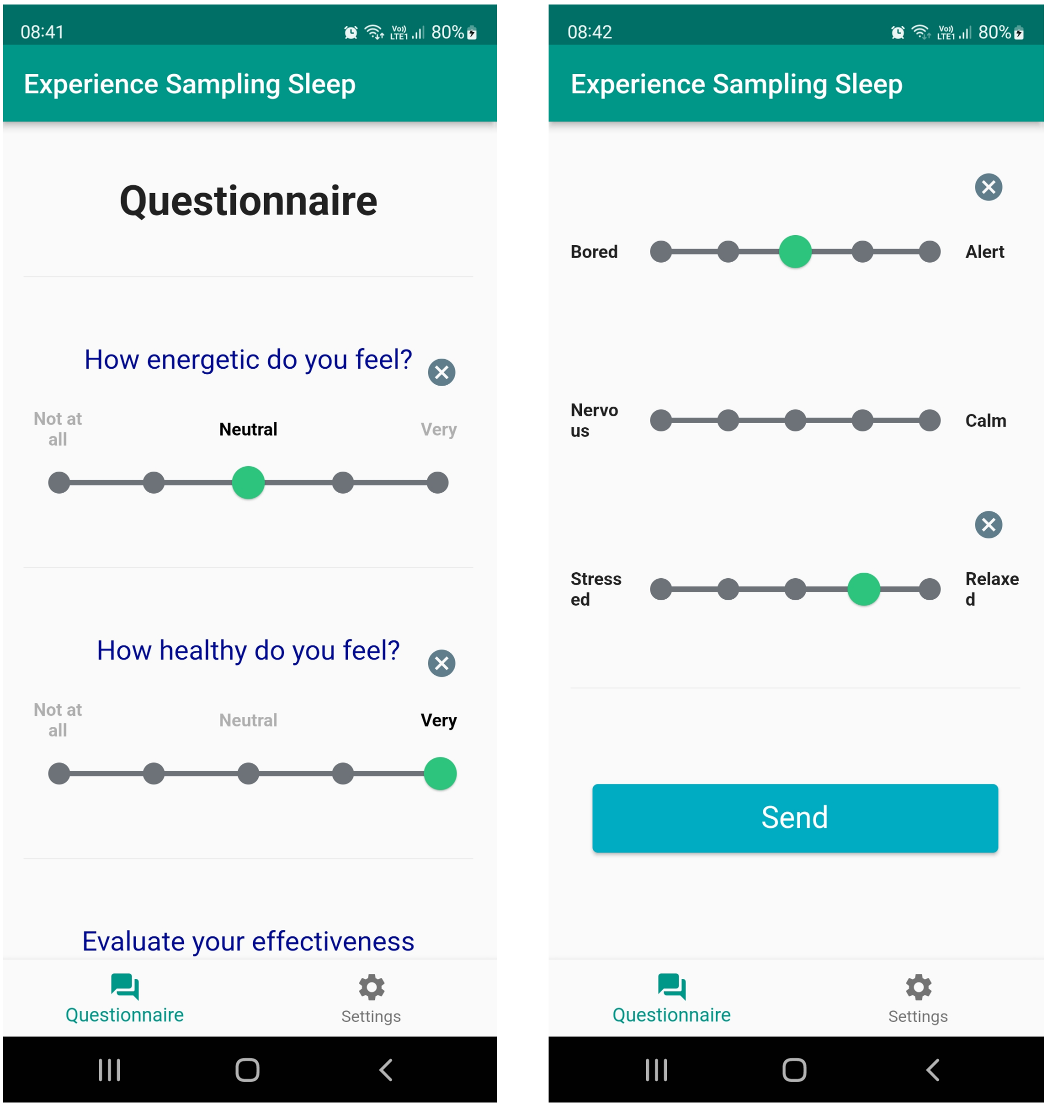
The purpose of the web dashboard (Figure 5) is to provide an easy-to-use user interface (UI) to add, edit and delete items from the cloud database. More specifically, it is used to edit the parameters of a study (Table 4) and create the questions (Table 5) for it.

To add questions without the web dashboard, one would create them directly in the Firebase console. This would involve first creating a table for the questions, then manually adding seven different columns to each question with predefined names and data types. To prevent erroneously configured parameters from crashing the application, we implemented a rudimentary error handling system which will display an appropriate error informing of the cause in the mobile application, for example if any of the predefined names or data types was input incorrectly in the Firebase console.

The web dashboard adds another safeguard layer, with predefined and tested processes for fast and convenient creation of new questions, while disallowing operations which result in error-prone configurations. Furthermore, data from the Cloud Firestore database can be exported in JSON format through the web dashboard. The web dashboard is also hosted on Firebase.

4.4. User Interface

The Flutter framework is designed around the principles of Googles' material design[36] guide. Flutter calls individual components, such as buttons and text, "widgets". The existing widget library covers most use-cases for mobile user-interface development, with the added benefit of cross-platform support for both iOS and Android. We chose to use the default stock widgets, with minimal styling changes for most of the application. For questionnaire sliders, we came across an open-source example[37] utilising an effective way to customize the slider widget, and adapted this into our project.



08:41

Experience Sampling Sleep

Questionnaire

How energetic do you feel?

Not at all — Neutral — Very

How healthy do you feel?

Not at all — Neutral — Very

Evaluate your effectiveness

08:42

Experience Sampling Sleep

Bored — Alert

Nervo us — Calm

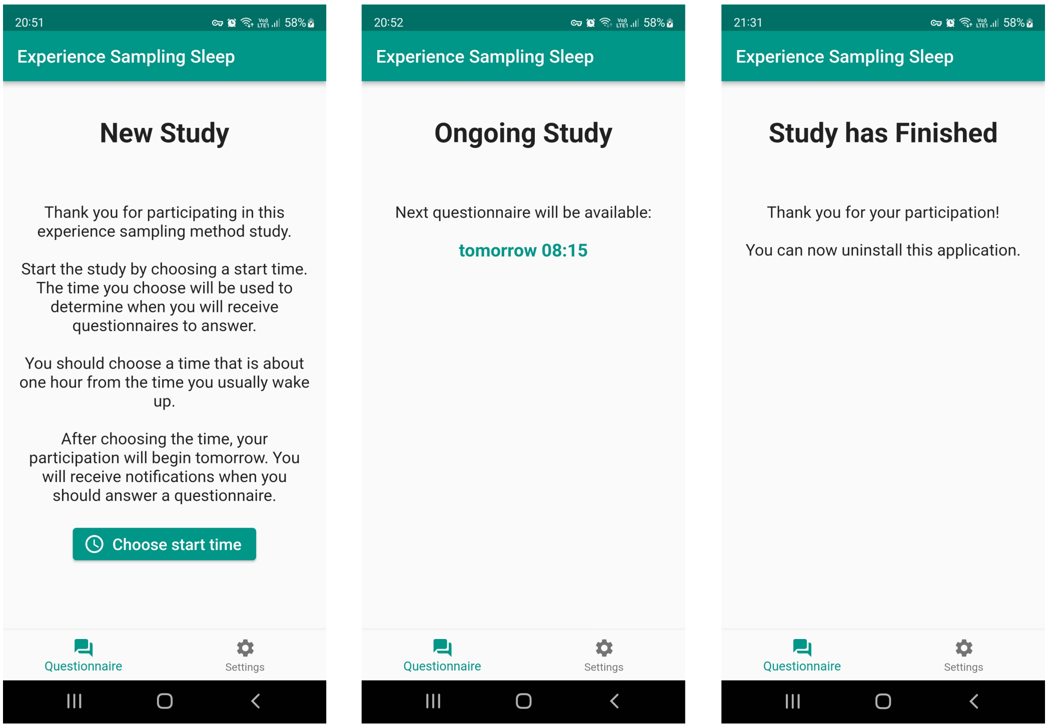
Stress ed — Relaxe d

Figure 6. Screenshot of the study questionnaire in the mobile application.

For the user-interface design, our goal was to create an easy-to-use, simple and clean UI. We opted for a standard bottom navigation bar design, with one tab delegated for questionnaire content and the other for system settings. We chose to display generic guidance intended to the subjects upon first installing the application. Subjects could utilize widely-adapted gestures and icons to navigate the application and when prompted, provide responses to the questionnaire.

The questionnaire UI (Figure 6) was designed to be as clear as possible. We used ample white-space and dividers between the questions for clear separation. The "Send" button is located at the bottom of the page as the user would expect. An "Undo" button for each question is displayed beside each slider once the slider has been "tapped". The button allows the user to take back their answer if they wish to leave said slider blank.

The states of the application's main view guide the user through the study (Figure 7). There are three states: start of the study, ongoing study and finished study. At the start of the study, the main view guides the user to choose a start time to start the study. This start time corresponds to the time of day the user wishes for the questionnaire prompts to start. In our study, we recommend to choose a time approximately one hour from the moment subjects usually awaken. When the start time is chosen, the main view will display the time of the next available questionnaire to the user in the ongoing study. After the last questionnaire on the final day of the study, the main view will display a message indicating the study has finished and users are free to uninstall the application.



20:51

Experience Sampling Sleep

New Study

Thank you for participating in this experience sampling method study.

Start the study by choosing a start time. The time you choose will be used to determine when you will receive questionnaires to answer.

You should choose a time that is about one hour from the time you usually wake up.

After choosing the time, your participation will begin tomorrow. You will receive notifications when you should answer a questionnaire.

20:52

Experience Sampling Sleep

Ongoing Study

Next questionnaire will be available:

tomorrow 08:15

21:31

Experience Sampling Sleep

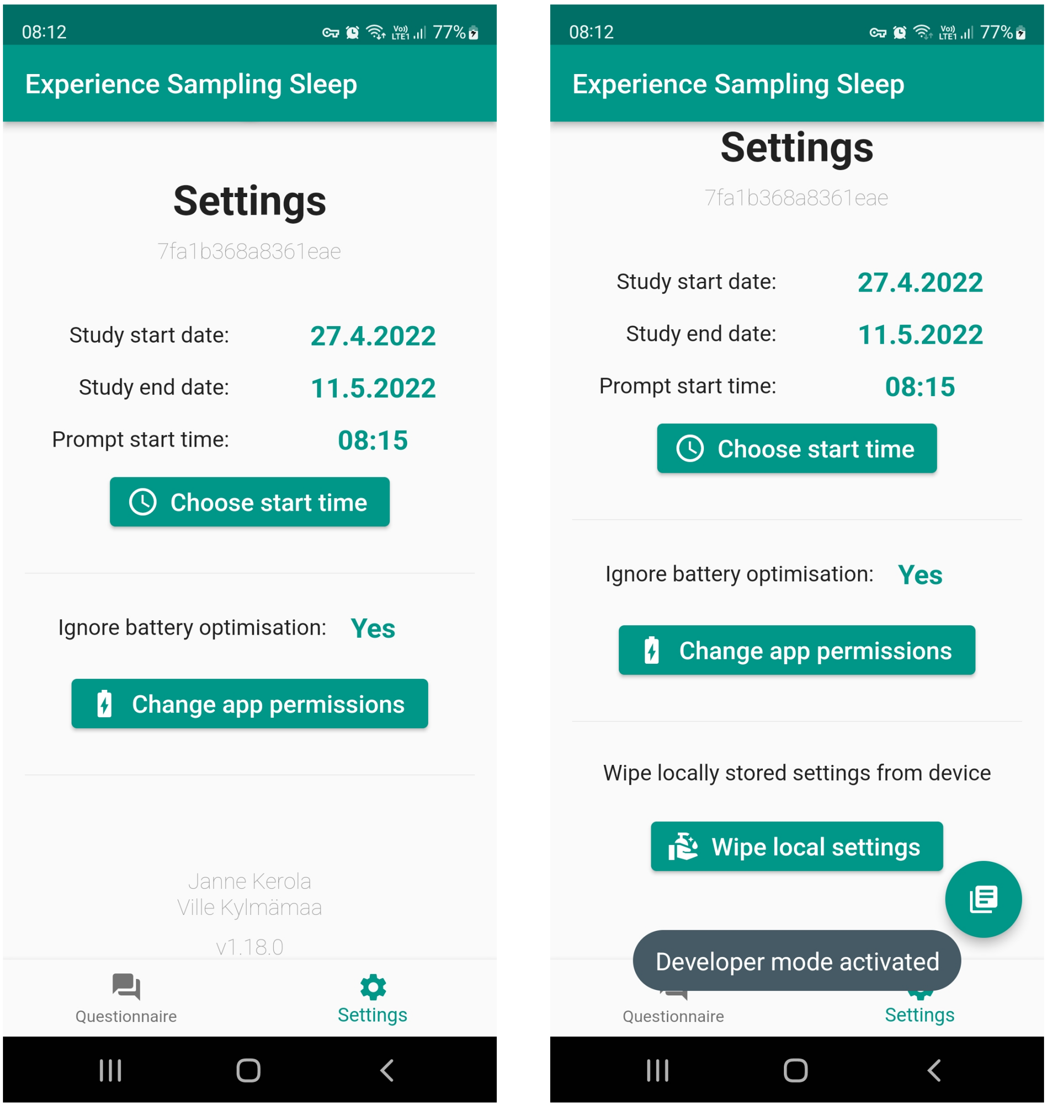
Study has Finished

Thank you for your participation!

You can now uninstall this application.

Figure 7. Different states of the mobile application during a study.

During development, we utilized an always present hover-button to display a separate screen of current database contents. This button was intended to speed up development and was modified to only display notifications sent from the parent device before the study began. We chose to hide this button behind a so called "developer mode", which is accessible by tapping the version build number in the settings panel 6 times in succession.



08:12 77%

Experience Sampling Sleep

Settings

7fa1b368a8361eae

Study start date: 27.4.2022

Study end date: 11.5.2022

Prompt start time: 08:15

Ignore battery optimisation: Yes

Janne Kerola
Ville Kylmämaa

v1.18.0

08:12 77%

Experience Sampling Sleep

Settings

7fa1b368a8361eae

Study start date: 27.4.2022

Study end date: 11.5.2022

Prompt start time: 08:15

Ignore battery optimisation: Yes

Wipe locally stored settings from device

Developer mode activated

Figure 8. Settings page of the mobile application. Normal view on the left, developer mode activated on the right.

The settings tab (Figure 8) contains information about the device, study start- and end-dates, as well as options for changing application specific permissions. We also chose to include a "wipe local settings" option to speed up testing in our development. Local settings consist of key-value pairs stored locally, for example the start date of the study and the time of day that the questionnaires start. The button to wipe local settings was available in the release version provided to the participants in our study, accessible under developer mode to ensure the participants would not press it without our instruction.

4.5. Third Party Materials

As a development device, we utilized a Nokia model 5.1 smartphone outfitted with the Android 9 operating system. We also utilized personal smartphones as test devices to a degree, but most of the development was done on an emulator bundled with Android Studio[38] running android 11 or API-level 30 equivalent.

For the objective sleep measurement, we used two devices: Fitbit Versa 3 wristwatch and Withings Sleep Analyzer under-mattress sleep tracker, as detailed in the Study Design section 3. The devices come with their own mobile applications, which are required to access and export their sleep quality data.

4.6. Security and Privacy

To be able to make calls to our Firebase Cloud Firestore, the sender must have the correct Google services API key and must be using either our Android application or our web dashboard. Authentication has been implemented in the web dashboard, requiring users to login to gain access. As the system currently doesn't yet support running multiple different studies at the same time from the same database, accounts can only be created directly in the Firebase project's console by the project's owner and editors, making this an easy and a certain way to secure the account creation process.

For an additional layer of security, we have programmed specific security rules to limit the types of calls that can be made to each table in the Cloud Firestore database:

Database Table	**Read**	**Create**	**Write**
responses	all	all	authentication
study_params	all	authentication	authentication
study_questions	all	authentication	authentication
timestamp_notification	all	all	authentication
timestamp_open	all	all	authentication

Table 6. Cloud Firestore database security rules.

- **Read:** The ability to get data from a table of the database, but not modify it in any way. The permission to read from any table is allowed to all.
- **Create:** The ability to add an item to a table of the database, but not being able to edit or delete anything. The permission to create study parameter and study question tables is allowed only to authenticated users.
- **Write:** The ability to add, edit and delete an item. The permission to write is granted only to authenticated users. Authenticated users are allowed to write to any table.

- Note that each of these actions also requires the correct Google services API key for the database.

Authentication is required to access the Firebase console and the web dashboard and to add, modify, or delete any data in the cloud database. As per the security rules (Table 6), there is currently no way to edit or remove any data from the database through the mobile application. In addition to read rights, the mobile application only has creation rights for the specific tables required for participating in the study. Thus, important data is protected without requiring authentication within the mobile application.

To be able to directly modify the Cloud Firestore database in the Firebase console, the accessing user-account has to have been appointed "Owner" or "Editor" in the Firebase project. As long as the owners of the Firebase project do not add members to the project who could compromise it, the system is secure.

Concerns about the privacy of their data was raised by the participants. The collected questionnaire data is sensitive, especially because it contains health and mood data. The data is first identified by the unique device identifier of the user. This is mostly anonymous, although not completely. Though unlikely, one could gain knowledge of someone's unique device identifier. Thus, in data processing we switch from this identifier to completely anonymous data group identifiers.

Data collected from the sleep measuring devices can also be considered sensitive. Regarding privacy, the weakest link in our procedure is the combining of data from the questionnaires to the data of the sleep measuring devices. This is a crucial step as the data needs to be combined for analysis. Unfortunately, there is currently no implemented procedure which would assuredly allow the participant stay anonymous to us. The main issue is that there are only two available devices, and these devices are different from each other. Therefore, only two participants may participate at a time and keeping track of devices is trivial for maintainers.

4.7. Risk Assessment

For the duration of our development process and study, we utilized Firebase as our backend. Due to an oversight on our part at the start of the project, we chose to include the accompanying google-services.json file containing the API key in the project repository, which could expose our development backend to malicious third parties, if the repository were made public during the study. We have therefore decided to take down our development Firebase project before publishing our source-code to the public. This will render the google-services.json file, which can still be found in the version history, outdated and useless for future malicious parties who chance upon our project. However, even in the case of a malicious attacker gaining access to our Firebase API key, our Cloud Firestore security rules (Table 6) would prevent the attackers from modifying or deleting any important data, because of the authentication requirement which the API key alone does not grant. Although, they could still spam the database with false responses.

A minor drawback in our design is the requirement for ignoring battery optimisation. This increases the risk of bugs which may cause excessive power drain without notifying the user. However, we tested our software rigorously and thus far have not noticed anything of the sort.

The switch from winter time to summer time occurred during our development process. This caused the notification time and the time the application displays the next questionnaire should be available, to go out of sync with the time that the questionnaire actually becomes available. This was caused due to the notification time and displayed time calculating elapsed time from the moment they were set, while questionnaire availability is directly scheduled with date and time. This can be fixed during a study with the current functionality, by instructing the user to update the questionnaire start time, but in the future this should be resolved automatically by the application.

Updates to mobile operating systems pose another issue. For example, future Android and iOS versions may require new explicitly given permissions causing conflict with the core functionality of our application. Maintenance is likely needed to ensure the long-term reliability of our application.

Regarding the reliability of our application in a study setting, our greatest concern is the reliability of the notifications being delivered from the background when the application has been closed for a long time. A high number of missed questionnaires, due the user not receiving notifications in time, could seriously undermine a study. The limiting factor in testing the reliability of notifications is our lack of time. Ideally, tests should be executed with a real device over multiple days. As our whole development took place over six consecutive weeks, we were not able to allocate enough time to test all possible configurations for an extended period of time to determine which solution suited us the best, as there are many different ways to implement the notifications.

5. EVALUATION

5.1. Evaluation Plan

In order to evaluate the functionality of our mobile application, we will conduct an ESM proof-of-concept study. The focal point of our study lies in evaluating the effects of sleep on the ESM response rate, with secondary interest in the responses themselves. Therefore, our plan is to collect response and sleep quality data from our subjects and evaluate their effective response rate over the period of two weeks. We will use statistical analysis to determine if any links between sleep amount and response rate, and sleep amount and the actual responses exist in our data.

Our pool of subjects is limited due to time and device constraints and we offer no monetary incentives to attract potential candidates. We will select from available persons two subjects able to commit to the study over a two-week period in early April, 2022.

We will give an introductory guide to the study to the subjects and present them with sleep monitoring devices before their respective study start dates and provide technical assistance over the duration of the study. We will then manually collect the sleep quality data from the subjects' devices before grouping it together with data generated by our application.

The reliability of notifications playing from the background will be evaluated by comparing the total amount of successful notifications to the total amount (70 per participant over 2 weeks) that should have occurred during the study. This is possible thanks to our application recording timestamps when a notification is successfully delivered.

5.2. Study Execution

We recruited two participants to our ESM study through personal relations. In this thesis, we will call them subject A and subject B. Both participants are students in the University of Oulu. Subject A is a 22-year-old male, and subject B is a 21-year-old female. We offered no compensation to our participants other than the sleep data that they would receive from the sleep measuring devices.

At the start of the study, we provided and familiarized the participants with our mobile application and the sleep measuring devices. We explained to them that this is a study utilizing the ESM method to study the effects of sleep on the subjects' mood, energy levels and health queried in the questionnaires. We did not tell them that we were more interested in their response rate and other metadata more than the answers themselves. We considered this obfuscation is essential because explaining that our main focus is on the participants' answering rate could affect their responses in an unwanted way.

Both participants started the study on the 6th of April and, with the 14-day duration, had their final questionnaire on the evening or night of 19th of April. Both had late sleep schedules and chose quite late starting times for their questionnaires, between 11 AM and 12 AM. Even so, both missed most of the 1st questionnaires of the day often due to still being asleep. Nevertheless, this showcases the usefulness of the participant being able to choose the starting time. If our participants were forced to start at 8 AM for example, they would have likely missed even more of the earlier notifications or they might have declined to participate in the study altogether. With the study window set to start between 11 AM and 12 AM, it would subsequently end between 11 PM and 12 PM.

During the study, we monitored the incoming responses and timestamps through our back-end to ensure that the study was going well. We had also requested the participants to report any bugs they might encounter. One such bug reported was in how the application checks for the availability of a questionnaire where the last questionnaire of the daily window would not show if the time went past midnight, thus changing the day. We fixed the bug immediately and provided the updated version to the subjects. When subject B contacted us about not receiving notifications, we advised subject B to change the starting time for the application to reschedule the notifications. However, we are unsure if this helped or not. Subject A required essentially no communication related to the study. During development, our supervisor reported an issue, where the notifications eventually stopped delivering from the background if the application wasn't opened for a whole day or two. Therefore, for our study, we asked the participants to launch the application if they missed a whole day of questionnaires without launching the application.

Finally, after the study was over, we asked the subjects in a neutral way how they experienced the ESM protocol conducted with 5 questionnaires for each day. Both commented that the experience was easy, since the questionnaires were so short and because one could simply answer how they felt in the moment instead of having to think about the answers over longer periods of time. When asked more directly if they found the study invasive or annoying, both reported that they didn't experience it that way thanks to how easy it was. One of the participants commented that the window to answer each questionnaire could have been longer, one hour for example, and that the question about sleep which was only asked in the first questionnaire of the day could have been asked in every questionnaire until it had been answered for the day.

5.3. Data Management and Pre-Processing

We collected all of the data after the study had ended on the 21st of April. The questionnaire response and timestamp data were obtained as a JSON dump of the Firestore database through our web dashboard, while sleep data was obtained directly from the subjects themselves as CSV data.

We utilized the open-source JavaScript libraries Chart.js [39] and D3.js [40] to combine and manage study data and produce all visualizations present in this thesis. The benefit of using JavaScript over more traditional tools is the possible inclusion of visualizations directly into the web dashboard (Figure 5) to allow better monitoring of an ongoing study.

Before analysis, we performed pre-processing by cleaning erroneous data. We removed notification timestamps for notifications which were unsuccessfully delivered at the wrong time. This anomaly was present in subject B's case when their device would block the notifications from being delivered from the background until the application was opened. When the application was finally opened, all the blocked notifications for questionnaires which had already passed would all be delivered at once. We also removed duplicate timestamps of which two were present in the case of subject A. The duplicate timestamps could have been possibly produced by the WorkManager task responsible for the timestamp running twice due to a slow response from the back-end resulting from the Workmanager retrying unfinished tasks at certain time intervals. In reality, the application did not actually deliver two notifications for these duplicate timestamps. We deemed that only successfully delivered notifications arriving on the correct time for the corresponding questionnaire should be used for our data analysis.

5.4. Study Results and Analysis

We mainly base our analysis on daily averages. Our data could also be used to study daily fluctuations as we have multiple answers to the same questions during each day. However, we decided that daily averages would be granular enough for our purposes.

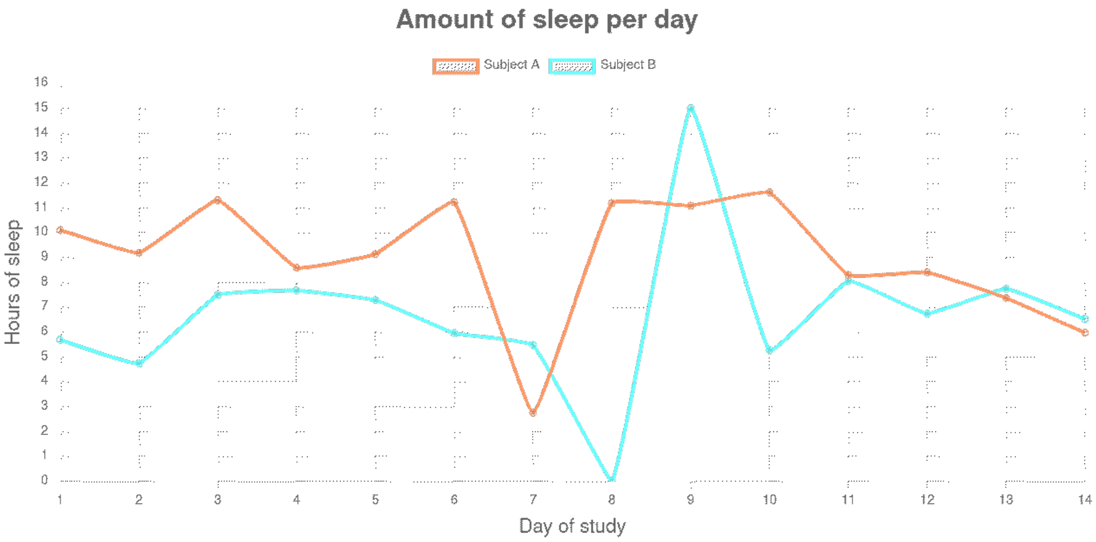


Figure 9. Chart depicting the average amount of sleep subjects had each day.

The amount of sleep was measured differently for each subject. Subject A's sleep data was measured with the Fitbit fitness tracker, which lists the duration of each stage of sleep (light, deep, REM) in seconds. For the visualization, the sum of the of the duration of these parts was used to graph the daily total amount of sleep had. For subject B, the Withings sleep mattress device lists the total amount of sleep as a separate statistic, which was directly used in the chart. Due to the personal nature of sleep cycles, we ran into visualization problems caused by the subjects entering sleep at any point during day 1, or in some cases the early hours of day 2. We solved this by inserting the sleep amount on the day the subject woke up, instead of when they fell asleep. This meant even if sleep was entered on day 1, all of the time spent asleep would be marked under day 2.

Interestingly, the changes seen in Figure 9 in sleep amount seem to mirror each other on some days. We are unsure if the subjects share these ticks out of sheer coincidence.

The major sleep disturbances in the middle of the study, and the shifting of the sleep windows (Figure 10 and Figure 11) on the latter half of the study, may have been caused by the start of the May Day events and festivities in the university.

The subjects had atypical amounts of sleep on some nights. For example, subject A's unexpected drop to just under three hours on day seven of the study, or subject B's 15 hours of sleep following a night of no sleep. While we left the outliers visible in our charts, we chose to exclude these values before performing mathematical operations on the data-set. After excluding the heavy outliers in data cleaning, the subjects had 8.02 hours of sleep on average. By themselves, subject A averaged 9.49 hours of sleep with consistent repetition, and subject B averaged 6.55 hours of sleep.

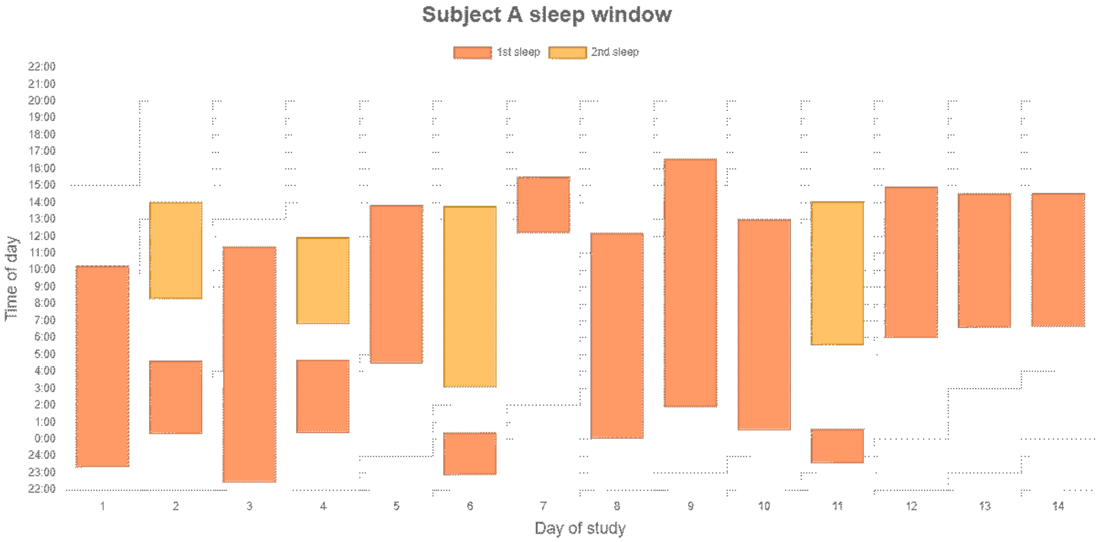


Figure 10. Chart depicting the sleep window for subject A.

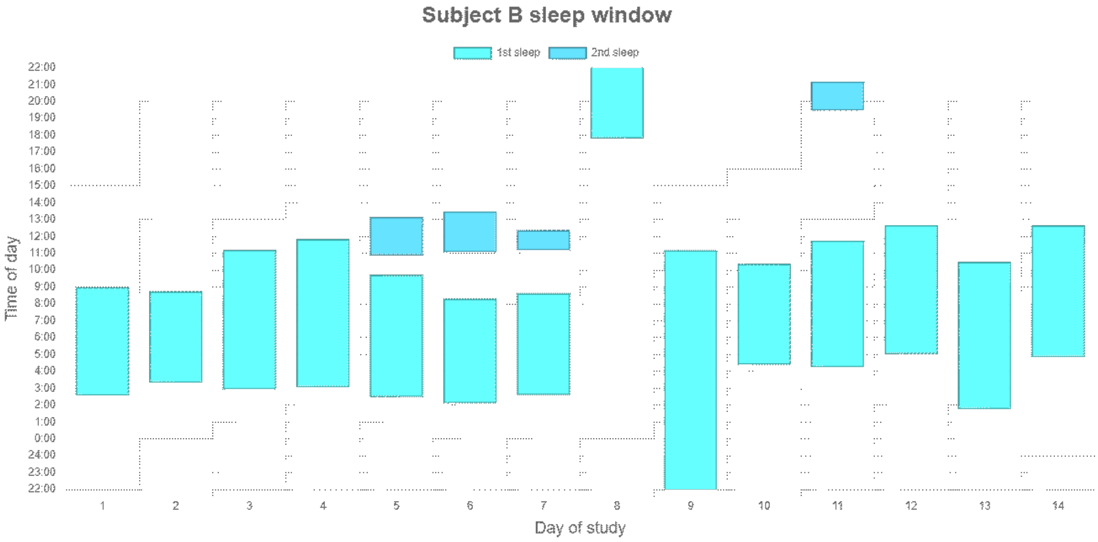


Figure 11. Chart depicting the sleep window for subject B.

Figure 10 and Figure 11, which visualize the subjects' sleep windows, further showcase the atypical occurrences in the subjects' sleep patterns. At the start of the study, subject A fell asleep between 23 PM and 1 AM, but on the last four days the subject fell asleep between 5 AM and 7 AM, moving the sleep schedule much later. Subject B's sleep schedule also moved a few hours later comparing the earlier and the latter half of the study. On some of the days, both subjects had biphasic sleep, during they woke up, stayed up for a few hours and went back to sleep again. The 15.0 hours of sleep recorded out of 17.4 hours in bed by subject B began at 5:50 PM on the late afternoon of the 8th day and lasted until almost midday 11:10 AM on the 9th day. This period was preceded by 29 hours of being awake.

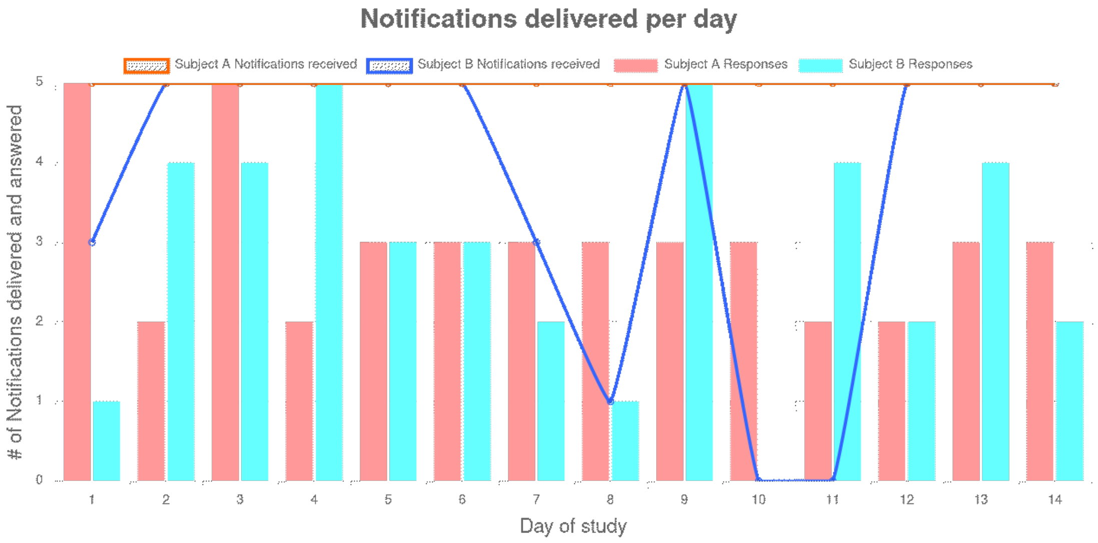


Figure 12. Chart depicting the amount of daily notifications that were successfully delivered during the time window of the corresponding questionnaire to each subject and the number of responses received for each subject.

As seen in Figure 12, notifications were successfully delivered 100% of the time to subject A. Notifications were successfully delivered 78.6% of the time to subject B. On subject B's personal device there were clear problems with the delivery during days 1, 7, 8, 10 and 11 of the study, while on 8 days out of the 14 total there were no issues. The study average for the response rate was 89.3%.

Post-study, we conducted a short 3-day test with subject B utilizing a different method of delivering notifications. This method scheduled notifications with Android AlarmManager. During the 3-day test, 100% of the notifications were successfully delivered to subject B. However, 3 days is not enough testing as our study showed that these issues can arise even after multiple days of perfect operation.

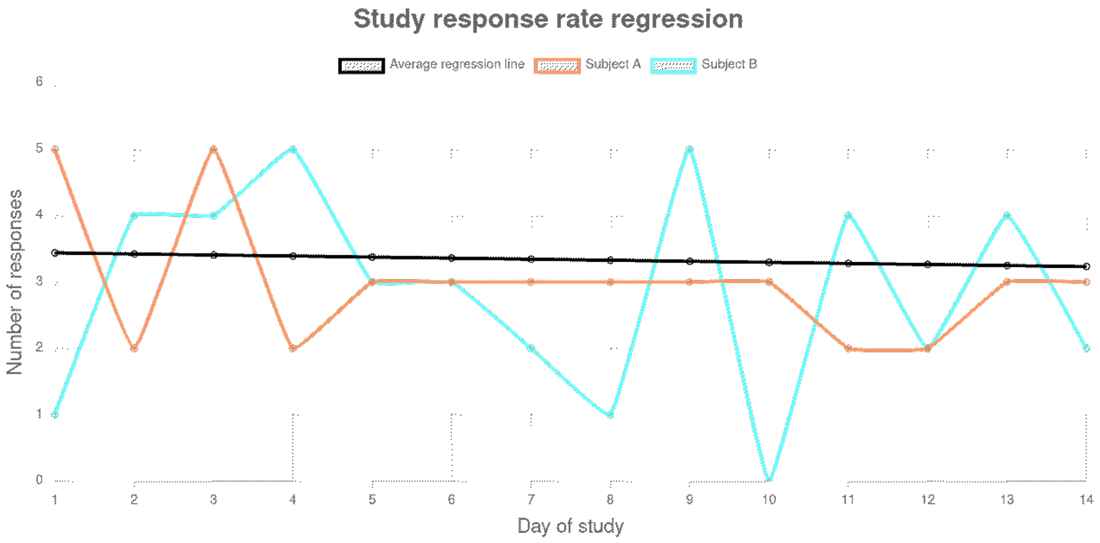


Figure 13. Chart with linear regression line depicting the study average.

As is shown by the linear regression line in Figure 13, the response rate for both subjects had only a very minute decrease during the study. The derivative of the linear regression line was roughly -0.016 and the decrease in response rate was about 6.0% from start to the end of the study.

On average, both subjects managed to respond to roughly three prompts each day. Note that some or all of the notifications being unsuccessfully delivered to subject B during days 1, 7, 8, 10 and 11 (as seen in Figure 12) most likely had an effect to the response rate during these days. Although no notifications were successfully delivered during day 11, subject B still managed to respond to four questionnaires.

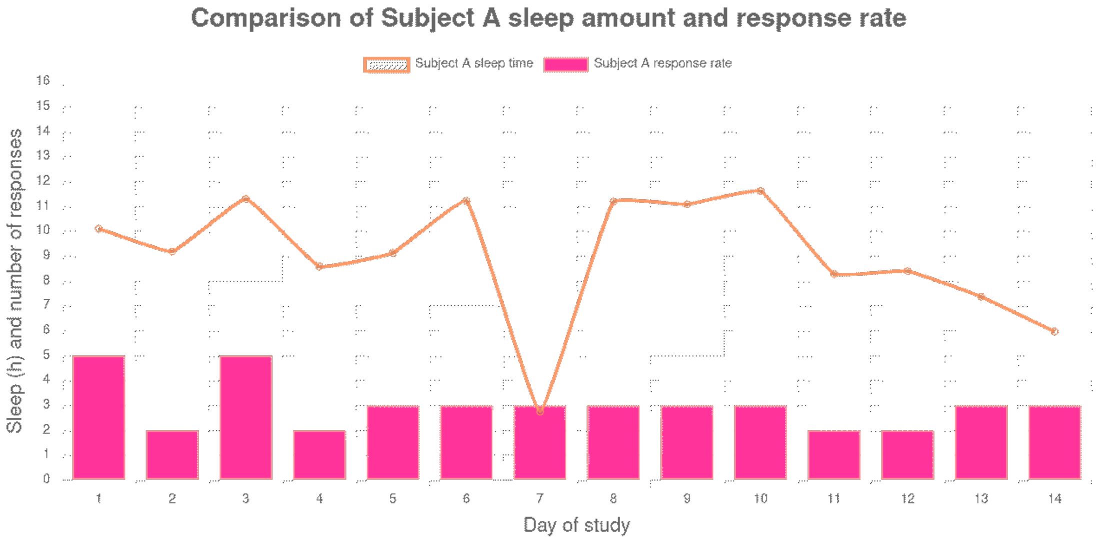


Figure 14. Chart comparing Subject A's number of daily responses with the amount of sleep had.

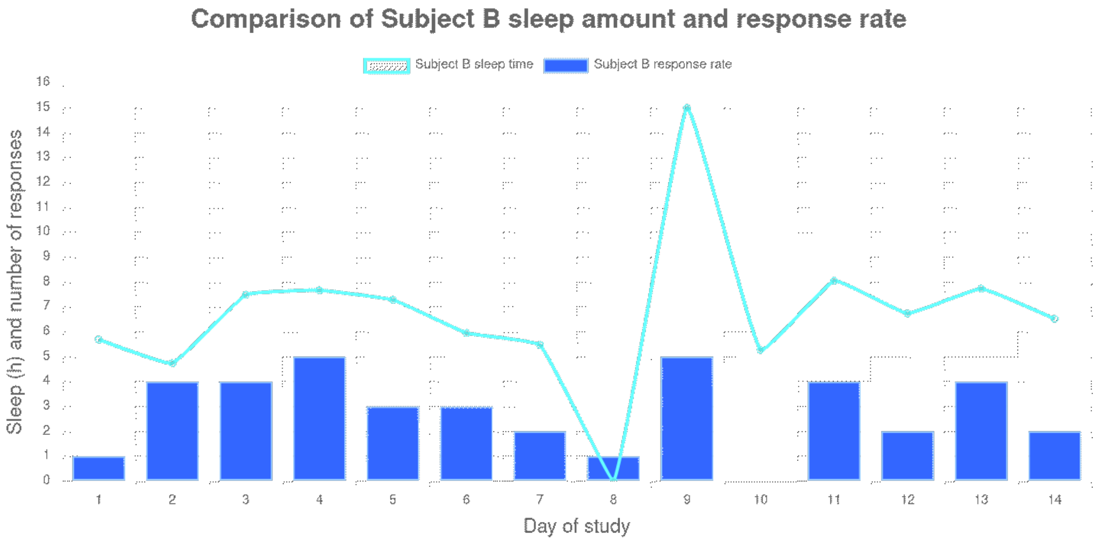


Figure 15. Chart comparing Subject B's number of daily responses with the amount of sleep had.

The charts in Figure 14 and Figure 15 offer some insight about the relationship between sleep amount and response rate. Especially subject B's response rate seemed to follow the amount of sleep detailed.

We used LibreOffice [41] to calculate correlation coefficients and p-values for the subjects' sleep amount and response rate. The correlation coefficient for Subject A was 0.395 with a p-value of 0.181, and for subject B the correlation coefficient was 0.588 with a p-value of 0.044. Sleep's effect on response rate was statistically significant (p-value $\leq$ 0.05) only for subject B, and not for subject A or the combined p-value.

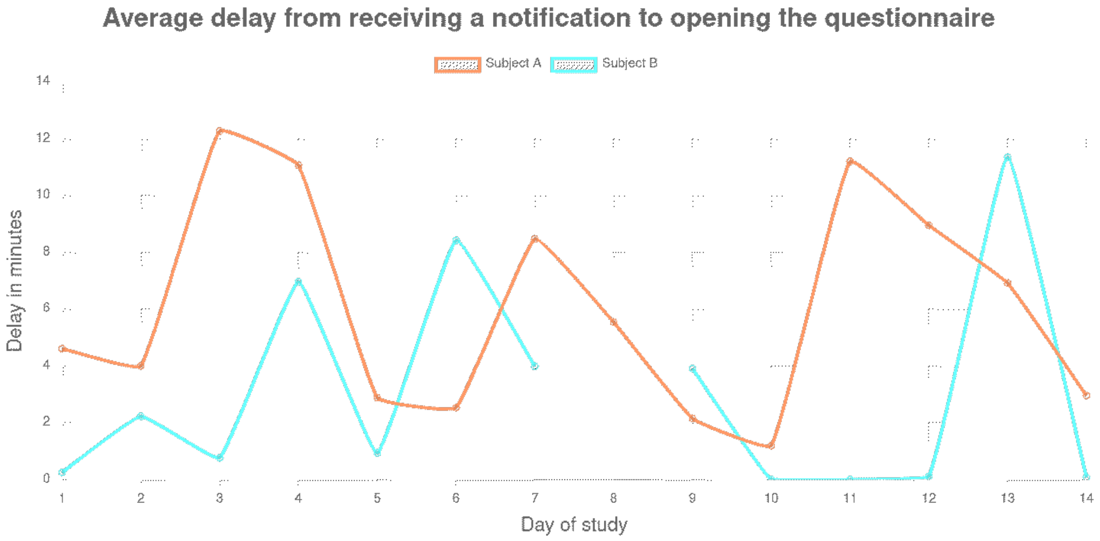


Figure 16. Chart depicting the average delay from subjects receiving notifications to opening them. Days where no notifications were delivered are blank.

The combined average delay for both of the subjects during the study from the moment a notification was delivered, to the moment the subject opened the application was 4.4 minutes. As seen in Figure 16, there was high variance in the delay with the daily average ranging between under a minute up to about 12 minutes. The highest delay was 26.3 minutes for subject A, and 27.8 minutes for subject B, when the subjects managed to answer a questionnaire at the last minutes of the 30-minute answer window for a questionnaire.

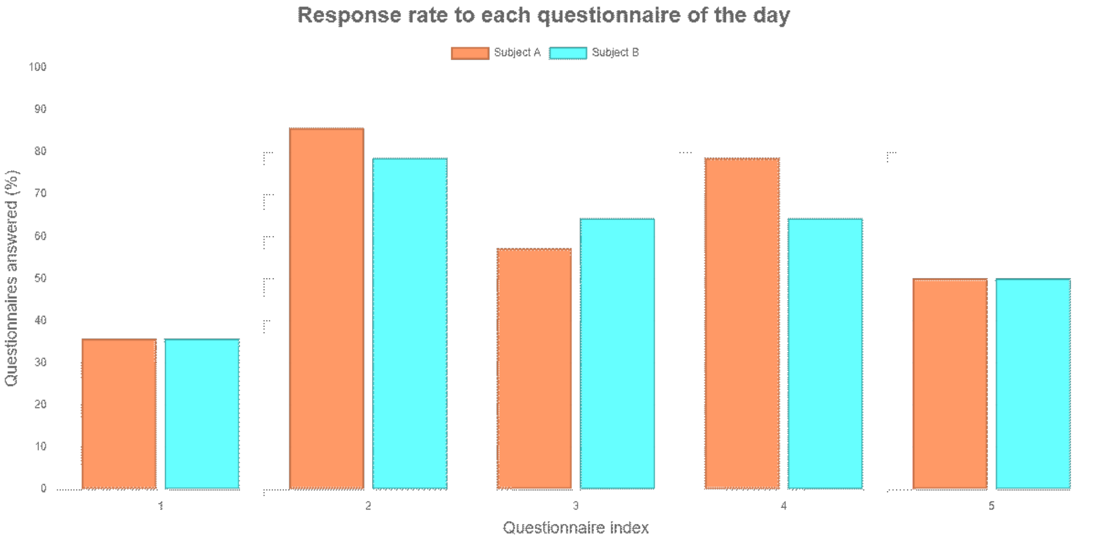


Figure 17. Chart depicting the response rate to each specific questionnaire of the day.

As seen in Figure 17, the first questionnaire of the day, answered on 35.7% of the study days, was by far the most commonly missed questionnaire by both subjects. This was due to the subjects often oversleeping (Figure 10 and Figure 11), even though both had set the starting time quite late between 11 AM and 12 AM. The second most common questionnaire to miss was the last one of the day, answered 50% of the time.

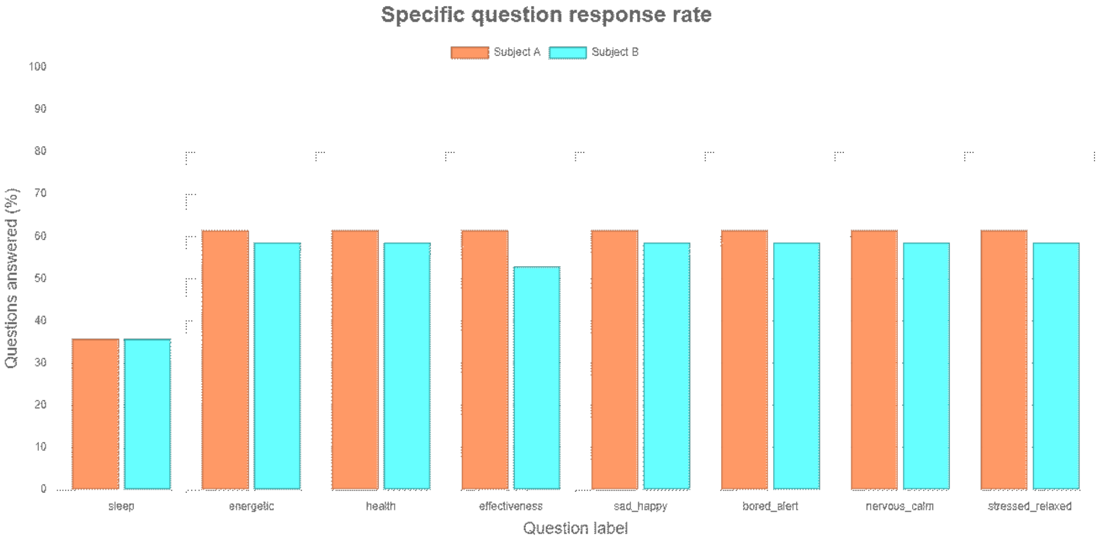


Figure 18. Chart depicting the how many times each question was answered.

In figure Figure 18, the potential maximum answer count to the sleep question was 14, due to only being present in the first questionnaire of the day. Both subjects answered the sleep question 5 times which is 35.7% of the potential maximum. Of the rest, roughly 60% were answered of the maximum of 70 responses.

Other than the effectiveness question in the case of subject B, each question was answered each time it was presented to each subject. Note the near equal amount each question was answered. This means that the subjects mostly did not skip questions, though the opportunity was present in each questionnaire. The only exception was subject B skipping the question "Evaluate your effectiveness in your current task (work, school, hobby, etc.)", which they skipped 4 times in situations where they considered their current task not sensible evaluating effectiveness for.

We excluded the question about subjective sleep quality from further analysis due to it receiving only a few responses, as seen in Figure 18.

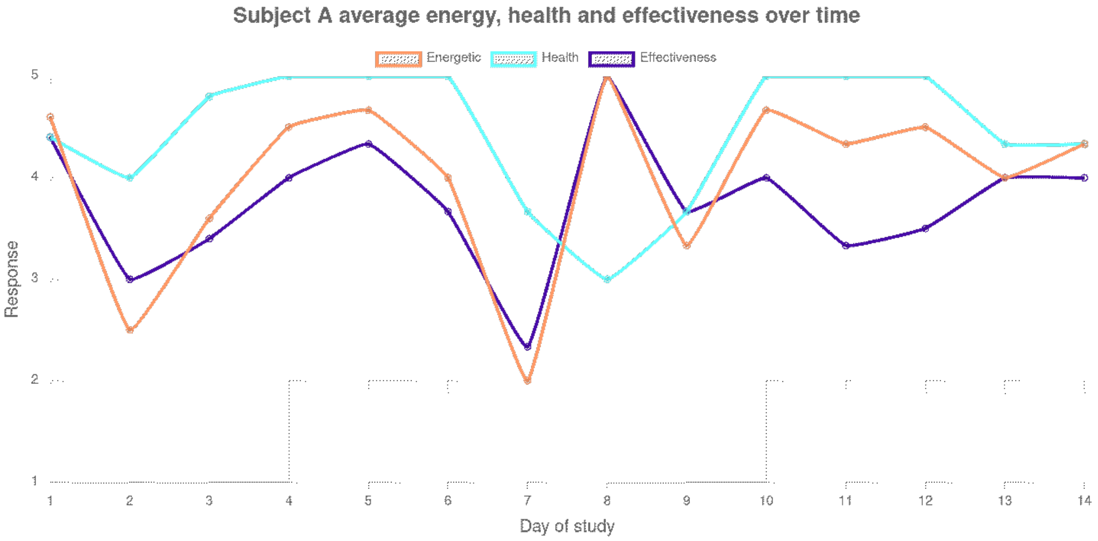


Figure 19. Chart depicting the daily average energy level, health and effectiveness responses from subject A.

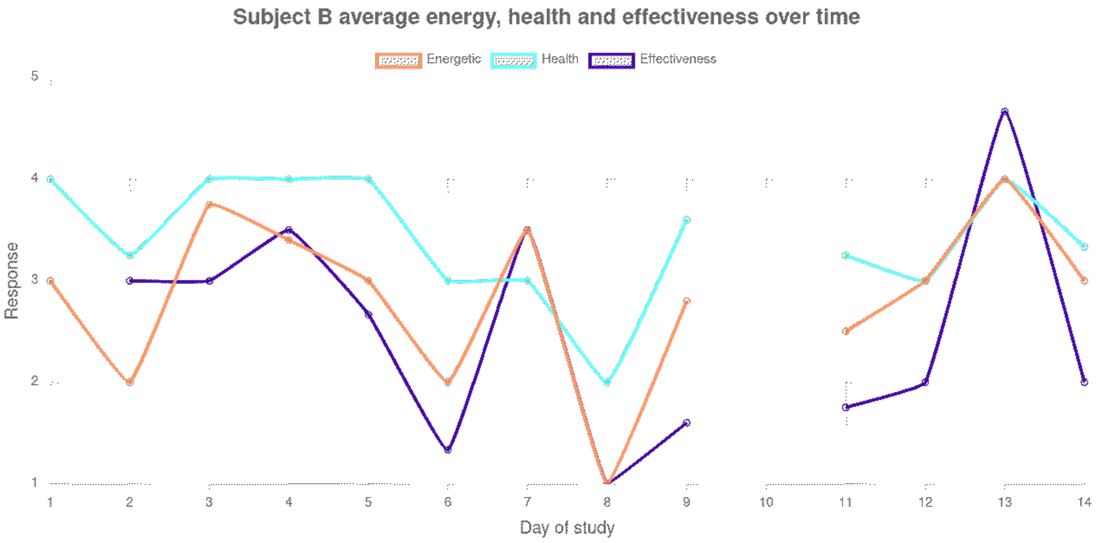


Figure 20. Chart depicting the daily average energy, health and effectiveness responses from subject B. The gap on day 10 is due to subject B answering no questionnaires on that day.

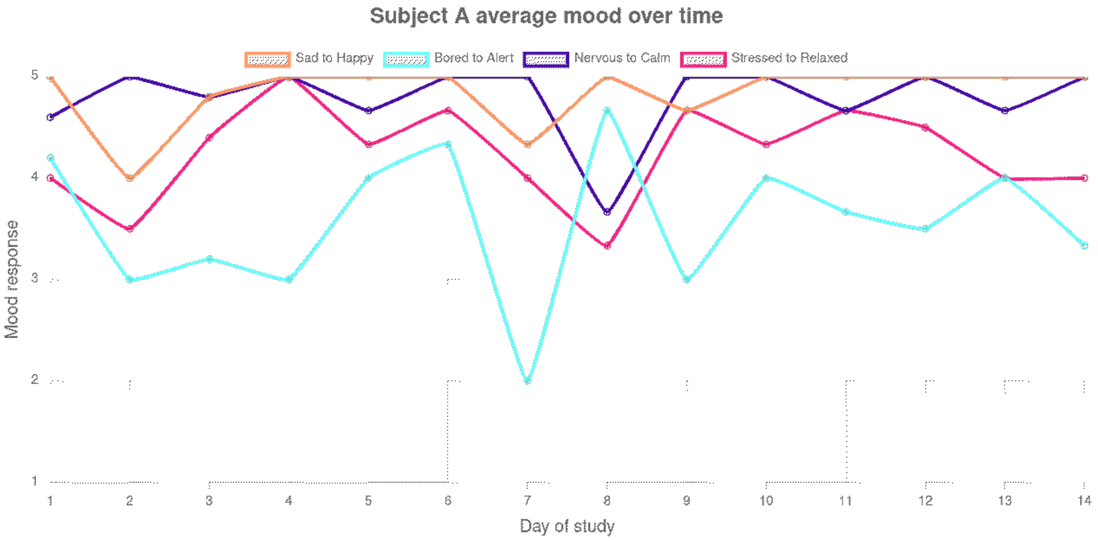


Figure 21. Chart depicting the daily average mood from subject A.

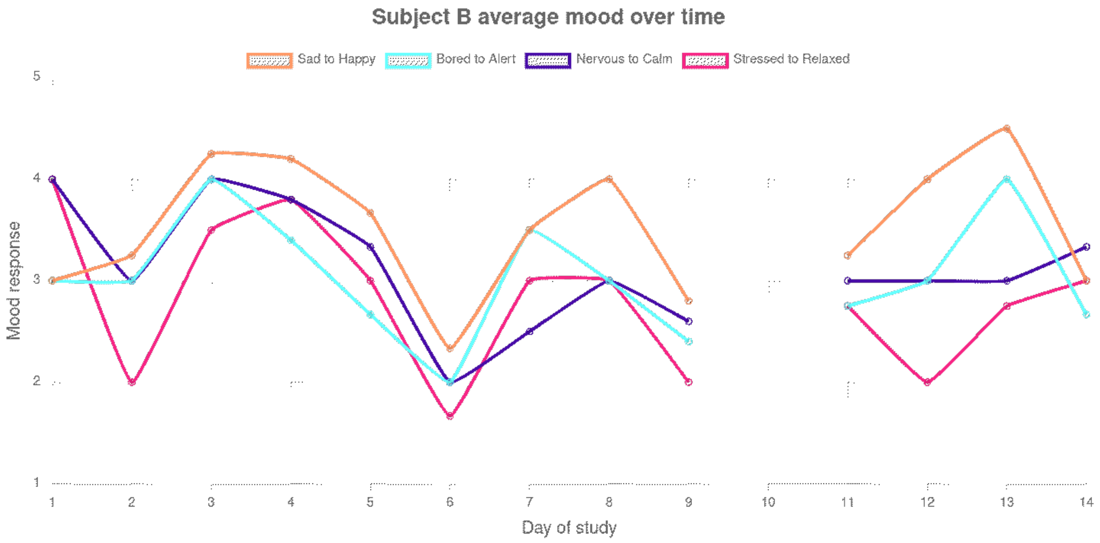


Figure 22. Chart depicting the daily average mood from subject B. The gap on day 10 is due to subject B answering no questionnaires on that day.

We present Figure 19, Figure 20, Figure 21 and Figure 22 to showcase some of the data which can be collected with our application. While the resulting visualizations are interesting, the statistical power of a two subject study is too low for deeper analysis.

Correlation between	**Subject A correlation**	**Subject A p-value**	**Subject B correlation**	**Subject B p-value**
Sleep and response rate	0.395	0.181	0.588	0.044
Sleep and energy level	-0.047	0.879	0.559	0.059
Sleep and health	-0.160	0.602	0.506	0.093
Sleep and effectiveness	0.116	0.706	0.417	0.177
Sleep and sad-happy	-0.097	0.753	0.587	0.045
Sleep and bored-alert	0.280	0.353	0.452	0.140
Sleep and nervous-calm	-0.207	0.496	0.465	0.128
Sleep and stressed-relaxed	0.028	0.928	0.457	0.135

Table 7. Table with correlation coefficients of the study.

Collected in Table 7 are the correlations calculated between sleep and response rate, and sleep and the answers to the questions asked in the questionnaires. The correlation between sleep and response rate, which was one of our main goals to investigate, had the strongest correlation coefficient and p-value.

However, the results are not statistically significant as most the p-values cross the 0.05 limit. In any case, the results could not have been generalized with only two participants in the study, and the high p-values were expected.

Statistic	**Subject A**	**Subject B**	**Study Average**	**Unit**
Average amount of sleep per day	9.49	6.55	8.02	Hours
Notifications delivered successfully	100%	78.6%	89.3%	Percentage
Average response rate	3.0	2.83	2.92	Responses per day
Average response rate as percentage	60.0%	56.7%	58.3%	Percentage
Response rate change from study start to end	-7.4%	-4.4%	-6.0%	Percentage
Average delay from receiving to opening	6.05	2.79	4.42	Minutes

Table 8. Table with statistical analysis values of the study.

Finally, Table 8 contains statistical key values collected from our study. Values involving sleep have been calculated using cleaned up data, where the outlier spikes as seen in Figure 9 have been omitted.

6. DISCUSSION

6.1. Reflection

Our main goal was to develop a mobile application with which an ESM study could be conducted with. We were successful in this goal and in many ways broadened the originally intended scope. Importantly, we developed our mobile application from the start with the idea that it should be possible to conduct many different kinds of ESM studies with it. Thus, the study parameters (Table 4) and the questions (Table 5) were designed to be fetched from the back-end database to make them configurable.

Additionally, we wanted to make the study parameters and questions configurable easily and error-free by anyone even without programming or database knowledge. Thus, we developed a web dashboard (section 4.3) to handle all the operations required with the back-end database to run an ESM study with our mobile application. In addition to configuring a study, these operations include downloading the data after the study has finished.

Furthermore, we added functionality to our mobile application to make it possible for the participant to choose when their daily questionnaire cycle starts. Instead of forcing every participant to adhere to the same daily time cycle, this feature enables ESM studies ran with our mobile application to adhere to the participating individual's sleep cycle. This allows for different kind of standardization, for example by instructing all participants to set up their start time one hour from when their usual wake-up time.

The user can also change their start time during the study in order to react to sudden shifts in their sleep cycle in the mobile application. Our subjects consistently overslept the time window allotted for the first ESM questionnaire of the day despite setting the start time on their own accord. This in turn prevented us from comparing the reported subjective sleep quality with the measured. The subjects in our study could have utilized this start time changing feature to counteract the changes in their sleep schedule (Figure 10 and Figure 11). However, they opted not to. Perhaps we did not advertise this feature and its purpose enough.

During development, we aimed to follow best practices in all of our programming and we took care to produce a scalable code-base for our mobile application and web dashboard. Both follow a component-based structure to make future additions and learning the code-base easier. In addition, we set up a DevOps environment to automate parts of the development life-cycle of the mobile application. However, we omitted writing unit tests for the mobile application and the web dashboard due to time constraints and tests not being included in the project scope. App-store testing was also ruled out, due to compulsory registration fees on both Google's developer console and Apple's developer's program, though delivery of the application could have been simplified by utilizing them.

Overall, we are proud of the results of our efforts in this Bachelor's thesis project. Our successes far outweigh the few shortcomings.

6.1.1. Study Evaluation

To evaluate the functionality of our implementation, we conducted a proof-of-concept study to test if our mobile application is suitable for the intended purpose. With only two sleep measuring devices available for our study, and the time requirements for this thesis allowing for only a single 2-week study period, our study was limited in statistical power. However, its main purpose was to serve as a proof-of-concept study for our implementation, and it was successful in this regard.

Subjects reported that the 30-minute response window we used could have been longer. We considered this time-span to be sufficient during our planning stage, with enough time to respond in most situations yet not too much to consider the resulting response data as ’outdated’. In retrospect, we realize that subjects may check their mobile phones upon receiving the notification, yet choose not to respond immediately due to activity related reasons, for example driving. Without additional reminders subjects may then forget to respond at all, resulting in subjects missing this study window. Subjects may also simply not hear the notification, and a longer response window could offer more chances to respond whenever subjects check their phones. However, response window should still be short enough to keep the questionnaires properly separated from each other. Rather than increase the response window, future developers could implement multiple reminders during each window to counteract distractions and forgetfulness.

If future studies include questions which require at least one answer per day, future developers should consider implementing safeguards to ensure that the questions presented in early prompts can be responded to, even if the subjects miss the initial window. For example, the important questions could be persisted in all questionnaires until a response is received for that day. Although, regarding questions about subjective sleep quality, optimally they should be answered close to the wake-up time and persisting the questions might result in participants answering them in the evening.

The features present in our mobile application which allow the user to choose and modify the start time by themselves appear to be unique in ESM research. Therefore, enabling these features could be made optional. A researcher conducting an ESM study with the application should be able to chose whether or not they want to allow the participants to choose the start time and reschedule it on their own. The choice to either enable or disable these features would be done on the web dashboard when configuring the study.

6.1.2. Mobile Application Limitations

A limitation that emerged during the study, was that the method utilized for delivering notifications in our mobile application does not work as intended on some mobile devices (Figure 12). 100% on notifications were successfully delivered to subject A but only 78.6% were successfully delivered to B. On the devices which have issues with notification delivery, these issues can arise even after multiple days of flawless operation. We are not sure why this occurs. This limitation resulted from us having only few test devices and only few full test weeks before starting our study. Optimally, the reliability should be tested with different devices and different operating system versions.

Our first implementation of notifications utilized Android's AlarmManager [42] and iOS's UILocalNotification [43] through the Flutter flutter_local_notifications package [34] to schedule all the notifications of the study at once. Later we started testing the Flutter Workmanager plugin [33] which utilizes Android's WorkManager [44] and iOS's performFetchWithCompletionHandler [45] to schedule background tasks, in order to launch notifications from the background.

On Android, AlarmManager may be more reliable than the WorkManager which was used during our proof-of-concept study. Functions to configure scheduled notifications utilizing AlarmManager already exists in the source code. With AlarmManager, setting up single notifications performed reliably while periodic notifications did not. This, and the limit of 64 simultaneously scheduled local notifications on iOS [43], is why we chose to abandon this method initially. However, it is possible to schedule all the notifications as single notifications, either all at once, or in batches for a few days at a time to circumvent performance issues or device specific limits. In a short 3-day test after our study, 100% of notifications were successfully delivered with the AlarmManager method to subject B, who had problems with the notifications during the study. However, we could not produce timestamps during this short test to ensure that the notifications were delivered. Rather, this relies on the report of the participant. Furthermore, 3 days is not a long enough duration as problems arose even after multiple days of flawless operation during our study Figure 12.

Another method of delivering notifications, which we could not allocate enough time to test, is sending push notifications from a remote server, for which the natural choice would be Firebase Cloud Messaging [46] given our Firebase back-end.

We had the desire to support iOS devices as well, which was one of the reasons why we chose to develop with Flutter. Unfortunately, the required macOS and Xcode dependencies prevented us from pursuing this path further [47]. We did not own devices with macOS and thus could not test our mobile application with iOS devices. Therefore, we have not tested what kind of platform specific solutions need to be developed in order for the mobile application to run on iOS.

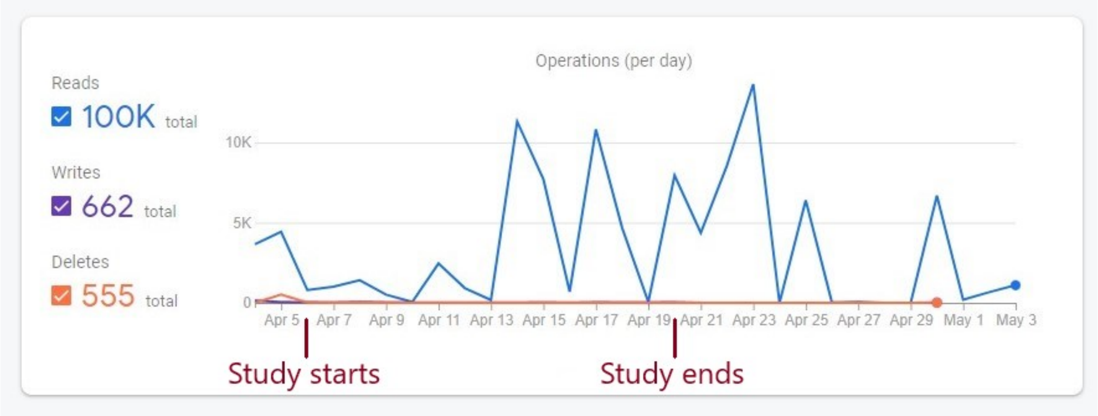


Figure 23. The amount of calls made to our Firebase Cloud Firestore database during April 2022.

The mobile application currently makes more read calls to the Cloud Firestore database than it needs to. This is due to the application fetching the questions from the cloud database each time the application is launched during a questionnaire window. This was helpful during development and did not matter in our small-scale study. However, for higher scalability fewer calls would be beneficial. Observing the days at the start of our study during which additional development did not occur, roughly between 500 to 1500 read calls were made daily, depending on how many questionnaires were answered by the two participants (Figure 23). Days surpassing 5,000 read calls result from development causes, such as reloading the data multiple times in rapid succession. As of writing, 3rd of May 2022, Cloud Firestore is free for up to 50,000 read calls per day after which the cost rises to $0.06 per 100,000 read calls [48]. The amount of read calls can be optimized by storing the questions locally in the device at the start of the study, as is done with the study parameters. The amount of write and delete calls is minuscule compared to the read calls, with under 60 write calls per day and 0 delete calls from the two participants during our study.

Every question in our study used 5-point Likert scale, and therefore we did not require implementing additional types of response methods. However, open-ended text fields, numeric value inputs, and multiple-choice questions would add value to the system as a study tool.

6.2. Study Results Compared to State of the Art

Due to the low number of subjects, the statistical power of the proof-of-concept study is low. Our choice of subjects also resulted in highly varied data, which created many unknown variables to consider in interpreting the results. Thus, confidence in the data is low and the results cannot be generalized.

Nevertheless, we take interest in comparing our results to the state of art. In the meta-analysis by Vachon et al. (2019), study duration had no significant association with response rate [8]. We observed essentially the same result. In our data, there was a minute 6% decrease in the response rate over 14 days, but this decrease was not statistically significant, and subject B not receiving all of the notifications successfully might have negatively affected the response rate.

The average response rate in our study was 58.3%. This appears to be somewhat low compared to other sleep studies which utilized ESM. Das-Friebel et al. (2020) had a study protocol close to ours: 14-day study duration with a questionnaire frequency of 6 per day, and the participants were undergraduates. In their study, Das-Friebel et al. required over 60% response rate, or the participant's responses would be excluded out of the data. Only 4 participants out of 116 were excluded. However, a major difference in this study was that the participants were free to answer to a prompted questionnaire at any time during the day. In other words, their response window was over 12 hours for the early questionnaires while we used a constant 30-minute response window. Furthermore, an incentive £35 in total was offered for achieving 67% response rate while we offered no monetary compensation. [4]

Takano et al. (2014) an average response rate of 78.9%, and Kammerer et al. (2021) had an average response rate of 71.72% in the patient group and 74.20% in the control group. However, both of these studies also offered monetary compensation based on response rate [3][6]. Block et al. (2019) had a particularly high average response rate at 93.16%, and for Sznitman et. al (2020) it was particularly low as 84 out of 138 participants were excluded due a response rate lower than 30% [7][5]. Other than the one exception, the response rate in our study was lower than in the referenced studies. Although, due to the numerous differences in monetary compensation, response window, questionnaire frequency, and other numerous details, our results are not directly comparable with the referenced studies.

Some notifications not being successfully delivered most likely had some effect on our response rate. We also believe, with the start of May Day events at the University of Oulu in the middle of our study, that the study took place during an atypical period in the participants' lives which among other issues caused anomalies in sleep affecting not only the responses but also the response rate itself. In an ideal setting, subjects may on average enjoy a more routine driven lifestyle.

6.3. Future Work

For our mobile application, the most important functionality to improve is the delivery of notifications. Further development should ensure that notifications are delivered reliably on as many mobile devices as possible.

In order to reach a wider subject pool, the mobile application should also be tested on iOS devices. We could not test this ourselves due to lacking the required devices. There may yet be further configuration left to do that we could not implement, and some features may require a platform specific solution to function.

Only one study can be run simultaneously with the current implementation. To expand on this, a system could be developed to allow multiple studies to be ran simultaneously, and these studies should all be possible to set up through the web dashboard. One example architectural solution to achieve this would include functionality for researchers to able to generate new study configurations, and edit or delete previously created ones on the web dashboard. A unique identifier is assigned to each study configuration, which the participants of the study then enter on the mobile application to direct which study configuration they should participate in. All the data collected from the participant is then collected under the specific study configuration identifier, visible and exportable only to the study owner.

We monitored our study directly through the Firebase Cloud Firestore database which is unwieldy. Thus, a section for monitoring an ongoing study could be added to the web dashboard. It could include information about the participants and visualizations of the data being collected.

7. CONCLUSION

Subjects engaged in ESM studies tend to display a drop in response rate over the duration of the study. While the causes of this drop are varied and some are uncontrollable by study managers, common causes can be minimized or eliminated by providing subjects core mechanism to enable participation.

To address these common causes, ease management, and further enable participation in these studies, we designed mobile-based study support system for conducting ESM studies. We developed the system over a period of 6 weeks in March 2022, expanding the scope of the project to include a user-friendly web dashboard to ease study management. We tested the application rigorously before setting out to conduct a proof-of-concept study to test the system in a real study situation.

For our proof-of-concept study, we chose to focus on the effects of sleep on this response rate. We designed this study with the intent of comparing sleep amount and response rate, and observing changes in the response rate over the course of the study.

We then conducted a initial test study on two subjects using our study support system, measuring response and sleep data over the study duration period of 14 days. We analyzed the results, disregarding statistical outliers present in the data and presented them in detail in the Evaluation section 5. Our system was able to provide us with rich data, proving its usefulness as a tool in future research.

While the scope of our study was too small to allow for conclusive results, we found interesting similarities in the visualization of the subjects sleep and response rate. We found that subjects often overslept the first prompt, leaving important questions present in them unanswered.

Future research topics include increasing the scope of the study to a far larger number of subjects, as well as implementing and observing the effects of reminder mechanisms on prompts yet to receive a response.

8. REFERENCES

[1] van Berkel N., Ferreira D. & Kostakos V. (2017) The experience sampling method on mobile devices. ACM Comput. Surv. 50. URL: https://doi.org/10.1145/3123988.

[2] Jasper Palmier-Claus Gillian Haddock F.V. (ed.) (2019) Experience sampling in the study of sleep and wakefulness. Routledge, London, 176 p.

[3] Takano K., Sakamoto S. & Tanno Y. (2014) Repetitive Thought Impairs Sleep Quality: An Experience Sampling Study. Behavior Therapy 45, pp. 67–82. DOI: https://doi.org/10.1016/j.beth.2013.09.004.

[4] Das-Friebel A., Lenneis A., Realo A., Sanborn A., Tang N., Wolke D., von Mühlenen A. & Lemola S. (2020) Bedtime social media use, sleep, and affective wellbeing in young adults: an experience sampling study. The Journal of Child Psychology and Psychiatry 61, pp. 1138–1149. DOI: https://doi.org/10.1111/jcpp.13326.

[5] Sznitman S., Shochat T. & Greene T. (2020) Is time elapsed between cannabis use and sleep start time associated with sleep continuity? An experience sampling method. Drug and Alcohol Dependence 208. DOI: https://doi.org/10.1016/j.drugalcdep.2020.107846.

[6] Kammerer M., Mehl S., Ludwig L. & Tania M. (2021) Sleep and circadian rhythm disruption predict persecutory symptom severity in day-to-day life: A combined actigraphy and experience sampling study. Journal of Abnormal Psychology 130, pp. 78–88. DOI: https://doi.org/10.1037/abn0000645.

[7] Block V., Meyer A., Miché M., Mikoteit T., Hoyer J., Imboden C., Bader K., Hatzinger M., Lieb R. & Gloster A. (2019) The effect of anticipatory stress and openness and engagement on subsequently perceived sleep quality–An Experience Sampling Method study. Journal of Sleep Research 29. DOI: https://doi.org/10.1111/jsr.12957.

[8] Vachon H., Viechtbauer W., Rintala A. & Myin-Germeys I. (2019) Compliance and Retention With the Experience Sampling Method Over the Continuum of Severe Mental Disorders: Meta-Analysis and Recommendations. Journal of Medical Internet Research 21. DOI: https://doi.org/10.2196/14475.

[9] Flutter framework. URL: https://flutter.dev/.

[10] Poner J., Russell J. & Peterson B. (2005) The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology. Development and Psychopathology 17. DOI: https://doi.org/10.1017/S0954579405050340.

[11] Withings sleep analyzer image. URL: https://www.withings.com/nl/en/sleep-analyzer.

[12] Niemi J., Risto R. & Salo S. (2021) Digital sleep : expert evaluation of commercially available digital sleep trackers. University of Oulu URL: http://urn.fi/URN:NBN:fi:oulu-202106238725.

[13] Withings mobile application. URL: https://play.google.com/store/apps/details?id=com.withings.wiscale2&hl=fi&gl=US.

[14] Withings developer api. URL: https://developer.withings.com/.

[15] Fitbit versa 3 image. URL: https://www.fitbit.com/global/us/products/smartwatches/versa3.

[16] Fitbit mobile application. URL: https://play.google.com/store/apps/details?id=com.fitbit.FitbitMobile.

[17] Fitbit developer api. URL: https://dev.fitbit.com/build/reference/web-api/.

[18] Kotlin. URL: https://kotlinlang.org/.

[19] Gitlab. URL: https://www.gitlab.com/.

[20] Conventional commits specification. URL: https://www.conventionalcommits.org/en/v1.0.0/.

[21] Semantic versioning specification. URL: https://semver.org/.

[22] Semantic release. URL: https://semantic-release.gitbook.io/semantic-release/.

[23] Pre-commit: A framework for managing and maintaining multi-language pre-commit hooks. URL: https://pre-commit.com/.

[24] Prettier - opinionated code formatter. URL: https://prettier.io/.

[25] Flutter format pre-commit hook. URL: https://github.com/Cretezy/flutter-format-pre-commit.

[26] Flutter analyze pre-commit hook. URL: https://github.com/dluksza/flutter-analyze-pre-commit.

[27] Firebase cloud firestore. URL: https://firebase.google.com/docs/firestore.

[28] Choose a database: Cloud firestore or realtime database. URL: https://firebase.google.com/docs/database/rtdb-vs-firestore.

[29] React. URL: https://reactjs.org/.

[30] Pub.dev; the official package repository for dart and flutter apps. URL: https://pub.dev.

[31] Hive: Fast, enjoyable and secure nosql database. URL: https://pub.dev/packages/hive.

[32] device_info_plus. URL: https://pub.dev/packages/device_info_plus.

[33] Flutter workmanager. URL: https://pub.dev/packages/workmanager.

[34] flutter_local_notifications. URL: https://pub.dev/packages/flutter_local_notifications.

[35] app_settings. URL: https://pub.dev/packages/app_settings.

[36] Material design guidebook. URL: https://material.io/design.

[37] Milke J., Slider example on github. URL: https://github.com/JohannesMilke/slider_example.

[38] Android studio. URL: https://developer.android.com/studio.

[39] Chart.js. URL: https://www.chartjs.org/.

[40] Data driven documents. URL: https://d3js.org/.

[41] Libreoffice suite. URL: https://fi.libreoffice.org/.

[42] Android alarmmanager. URL: https://developer.android.com/reference/android/app/AlarmManager.

[43] ios uilocalnotification. URL: https://developer.apple.com/documentation/uikit/uilocalnotification.

[44] Android workmanager. URL: https://developer.android.com/topic/libraries/architecture/workmanager.

[45] ios performfetchwithcompletionhandler. URL: https://developer.apple.com/documentation/uikit/uiapplicationdelegate/1623125-application.

[46] Firebase cloud messaging. URL: https://firebase.google.com/docs/cloud-messaging.

[47] Build and release an ios app. URL: https://docs.flutter.dev/deployment/ios.

[48] Firestore pricing. URL: https://cloud.google.com/firestore/pricing.