Axel Kjønsberg og Erlend Ydse

Supporting Time Management among Students with the Aid of Digital Tools

Master's thesis in Datateknologi
Supervisor: Trond Aalberg
July 2022

Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

Axel Kjønsberg og Erlend Ydse

Supporting Time Management among Students with the Aid of Digital Tools

Master’s thesis in Datateknologi
Supervisor: Trond Aalberg
July 2022

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

Abstract

Students at universities and colleges have a considerable amount of different tasks to deal with throughout a semester. For many students, efficient planning as well as good habits and routines are necessary to stay on top of their mandatory course work. Good management of time has also been shown to strongly correlate with the performance and well being of students. To facilitate the behaviours related to time management, a tool which supports automatic task planning for students was designed and implemented. The design is based on existing theory of scheduling problems as well as a survey of student's study preferences. Afterwards, the tool was tested through the use of qualitative and quantitative methods, and the results were used to discuss the utility of such a tool in addition to deriving requirements and principles to aid in the design of similar tools.

It was eventually shown that students have several needs which must be taken into consideration when planning. Many of these needs may be automatically accommodated for by a scheduling algorithm, assuming that the underlying scheduling problem is modelled befittingly. The more peculiar planning needs of students, however, are not as easily accounted for by classical scheduling algorithms, and therefore introduce additional requirements for tools which should support planning.

Keywords: time management, task management, task planning, short-term planning, study support tool, learning support tool, learning technology, scheduling algorithms, scheduling in practice

Sammendrag

Studenter ved høyskoler og universiteter må håndtere en stor mengde oppgaver gjennom et semester. For mange studenter er det nødvendig å planlegge effektivt og opprettholde gode vaner for å holde tritt med deres obligatoriske studiearbeid. Det har blitt vist at god tidsstyring korrelerer sterkt med studenters akademiske prestasjon og velvære. For å tilrettelegge for oppførsel knyttet til tidsstyring ble det implementert og designet et verktøy som støtter automatisk oppgaveplanlegging for studenter. Designet er basert på eksisterende teori angående planleggingsproblemer i tillegg til en spørreundersøkelse om studenters studievaner. Etterpå ble verktøyet testet gjennom kvalitative og kvantitative metoder, og resultatene ble brukt til å diskutere nytten av et slikt verktøy i tillegg til å avlede krav og prinsipper for å støtte designet av lignende verktøy.

Det ble etterhvert vist at studenter har flere behov som må tas i betraktning under planlegging. Mange av disse behovene kan støttes automatisk via en planleggingsalgoritme så lenge det kan antas at det underliggende planleggingsproblemet har blitt modellert på en passende måte. De mer særegne planlegningsbehovene til studenter er dog ikke like enkle å imøtekomme via klassiske planleggingsalgoritmer, og introduserer derfor ytterlige krav til andre aspekter ved verktøy som skal støtte planlegging.

Acknowledgements

We would like to thank our supervisor Trond Aalberg for being able to entrust us with the thesis which we proposed for him, and also for providing an appropriate amount of guidance during our work.

Also, thanks to our fellow students who have provided encouragement and shown enthusiasm for our work throughout this semester, especially to our fellow master's graduate Eivind Yu Nilsen who, in addition to being of good support, has been a great sparring partner for discussing theory and proposing several algorithms.

Contents

List of Figures

List of Tables

Acronyms

EDD Earliest Due Date. 14, 30, 35, 38, 68, 70

MOOC Massive Open Online Course. 5

NTNU The Norwegian University of Science and Technology. 20, 25, 34, 39, 44–46, 74

PC Principal Component. 54, 59, 60

PCA Principal Component Analysis. 54, 59, 60

SPTF Shortest Processing Time First. 14, 18, 35, 38

UI User Interface. 6, 26

VLE Virtual Learning Environment. 25, 26, 29, 34, 35

WSPTF Weighted Shortest Processing Time First. 14

CHAPTER 1

Introduction

The everyday of an average student is filled with several tasks stemming from a wide array of responsibilities. A student will usually enrol in several courses per semester, where each course, more often than not, provides a considerable amount of mandatory coursework. It also not unusual to partake in activities such as part-time jobs, voluntary work and organized training in addition to studies throughout the day. In order for a student to keep track of all of these different tasks, deadlines and liabilities, some sort of planning is generally called for.

1.1 Motivation

For many students, the act of planning might feel like a task in and of itself, and planning therefore tends to be pushed aside when it, perhaps, should not. The idea of a student which never plans for anything, but rather gambles on getting things done in the nick of time, is certainly not unheard of either. These types of behaviours emerge in several studies, with, for instance, Rabin, Fogel, and Nutter-Upham [55] asserting that *dilatory behaviour*, commonly known as *procrastination*, is prevalent among 30 to 60 percent of all students. Steel [65] claims that these numbers might even be higher. Moreover, Steel [65] asserts that procrastination may negatively impact academic results, with Macan, Shahani, Dipboye, and Phillips [34] and Nordby, Wang, Dahl, and Svartdal [43] among others supporting this claim. Adding to the case against procrastination is its connection to mental health issues such as depression and anxiety [35, 61]. All in all, one might say that procrastination is something most students could manage better without. Conversely, the behaviours of *time management* have been shown to correlate with the absence of procrastination [22, 59]. Thus, if the goal is to reduce the amount of procrastination among students, a step in the right direction might be to facilitate time management behaviours.

If digital tools may be used to support *task management*, for instance by the use of to-do list applications (of which the prevalence should not be understated [67]), perhaps similar tools can be used for *time management* as well. However, this might be easier said than done, as the act of planning and managing time may be more complex than it first might seem. Not only do people exhibit different behaviours when managing their time [33], but their preferences for how to perform these behaviours also differ significantly. Nonetheless, the prevalence of digital calendars, planners as well as to-do lists hints at a need for such tools to aid in time management behaviours.

1.2 Research Questions

The overarching theme for this thesis may be described as **how task and time management among students can be facilitated through the use of digital tools**. Additional emphasis is placed on how time management can be supported through automatic planning of tasks. Furthermore, in order to gain insight into the problems and presupposition which have been presented thus far, the following research questions are proposed:

RQ1: What is the state-of-the-art of tools for supporting task and time management?

RQ2: What needs and preferences of students should be considered by tools for supporting task and time management?

RQ3: What kind of scheduling algorithms may be applied in the domain of study related tasks?

RQ4: What are students' perception of a tool for supporting task and time management?

The first research question RQ1 is for the most part self-explanatory. In order to gauge the standard among tools which support time and task management, some kind of survey needs to be performed. However, it is not as straight forward as only looking at what has been done in similar research; a considerable amount of work done on designing such tools comes from the domain of *commercially* available tools and applications. Therefore, both related research as well as a state-of-the-art commercial tools will be presented in chapter 2.

The second research question RQ2 relates to what *needs* students may exhibit when partaking in the behaviours of task and time management. This question mainly considers the needs which are prevalent among students, but also encompasses the needs which are generally required for planning among the overall population. These are needs which should be supported by a tool which is meant to facilitate planning. A method for uncovering these needs will be to survey the study preferences and habits of students, the results of which will be presented in chapter 4. Another method is to let students test and evaluate an actual tool which supports time management, thereby providing the possibility of gaining further insight through the students' feedback. The process and results of this evaluation is presented in chapters 7 and 8.

The third research question RQ3 concerns which algorithms from the field of *scheduling* may be applied to solve the problem of planning study related tasks. Moreover, in order to derive such algorithms, it is of much help if the problem can be properly formulated as a scheduling problem. This makes it possible to connect and relate the problem at hand to other known problems, which in turn opens the door for building upon previous solutions and algorithms. A summary of scheduling theory, as well as the usual ways of solving certain scheduling problems, will therefore be provided as part of the background in chapter 3.

The fourth research question RQ4 is composed of two smaller sub-questions. The first of which is aimed specifically at the plan which may be suggested to a student by a planning tool. Given that the student's general needs for planning are already met by the tool, what is required from the suggested plan in order for it to be accepted by the student? More specifically, what is required from the presentation and user interaction with a generated personal plan. The other sub-question relates to the perceived *utility* of such a tool. Are there perhaps any usability issues which are inherent to this type of tool specifically? Like with RQ2, insights into these questions have been the focus of the evaluation process which is presented in chapter 7 and chapter 8.

CHAPTER 2

Tools to Aid Task & Time Management

A search was made for state-of-the-art solutions related to the problem of managing study related tasks. The process of searching encompassed both relevant literature from previous research as well as commercial tools and services which were reportedly used by students. The literature was found through Google Scholar using a collection of different keywords. The selected sample of commercially available tools was mostly based on what students reported to using in the survey presented in chapter 4 as well as during the evaluation processes of chapter 7 and chapter 8.

2.1 Definitions of Task and Time Management

A few criteria were used when choosing which works, tools and solutions to consider. Firstly, they needed to be related to either **task management** or **time management** in some way. Task management was simply defined as behaviours which either raise *task awareness* (to-do lists), support *task monitoring* (for instance by providing notifications) or help with *task decomposition*. The definition of time management, however, is a little bit more tricky, as time management may be considered a *result* of several composite behaviours rather than a behaviour in and of itself. For the purpose of this thesis, the definition of time management suggested by Claessens, Eerde, Rutte, and Roe [11] is preferred. Here, time management is defined through a set of behaviours including **time assessment**, **time planning** and **time monitoring**.

Time assessment behaviours refer to behaviours which raise awareness of one's use of time. In turn, these behaviours may help in deciding whether or not additional tasks and responsibilities should be accepted, as well as answering questions like "do I actually have time for this?" and "will I be available at this hour or not?"

Furthermore, *time planning* behaviours are behaviours which aim at making the most out of one's time, in addition to ensuring that tasks are performed at an appropriate point in time. These behaviours may include the act of setting goals, prioritizing and grouping tasks, as well as setting aside time in one's calendar for specific tasks.

Finally, *time monitoring* behaviours revolve around observing one's use of time during activities. Following this behaviour, it is possible garner useful insights into how time might be used more effectively and whether or not *more* time needs to be dedicated to certain tasks during future planning.

2.2 Related Work

In the realm of time monitoring behaviours, de Raadt and Dekeyser [14] proposed the inclusion of a simple progress bar on the course pages of a learning platform. The proposed progress bar illustrated how many of the required assignments the student had completed, and how many they had left. The inclusion of such a tool was well received by the students, with 85 percent of the 139 respondents answering that they would like to use the tool in other courses as well, and 71 percent of the respondents stating that the tool helped them manage their time.

In order to counteract the lack of direct support given to Massive Open Online Course (MOOC) learners, in addition to promoting self-regulated learning, Alario Hoyos, Estévez Ayres, Pérez Sanagustín, Leony, and Delgado Kloos [3] have proposed **MyLearningMentor** as a mobile application to support learners. The tool delivers a collection of tasks based on the MOOCs in which the learner has enrolled. It then suggests a schedule for how the learner should proceed for getting these tasks done. Additionally, hints for how to adopt certain time management behaviours are provided to the learner from a separate hint database. The tool assumes that the student works on one task at a time and measures the time spent. The spent time is used to personalize all estimated completion times for a given student, as well as in crowd-sourcing for improving estimated completion times for the completed tasks for other students.

A similar approach in terms of crowd-sourcing is found in the **TaskGenies** system proposed by Kokkalis, Köhn, Huebner, Lee, Schulze, and Klemmer [27]. However, the main purpose of the tool is *Action planning* [27, 33] rather than time management. The act of action planning is usually thought of as an extension of task management, as it consists of breaking down tasks into smaller sub-actions which require less effort to act upon. In the case of TaskGenies, the process of decomposing tasks is performed through crowd-sourcing, where the crowd-sourcing is used to review how previous tasks containing similar keywords were broken down. The results of Kokkalis, Köhn, Huebner, Lee, Schulze, and Klemmer [27] showed that providing action plans significantly increased task completion rates for high-level tasks, and that the participants who re-used action plans through crowd-sourcing completed more tasks than the control group. However, the usability of many of the crowd-sourced plans was not great, as the system often would find and recommend actions completely unrelated to the given task.

As an effort to further encourage self-regulated learning in MOOCs, Pérez-Álvarez, Maldonado-Mahauad, Sapunar-Opazo, and Pérez-Sanagustín [45] proposed **NoteMyProgress**, a browser plugin which monitors activity and time spent on a MOOC platform. The plugin contains a dashboard which visualizes the aggregated data with interactive graphs, with the aim of helping the users get an overview of their activity in their courses. This is an example of a tool which supports time monitoring, but not necessarily time assessment or planning. In a pilot study of the beta version of the tool, the average usability evaluation of the tool was above 3.67 on a 5-point Likert scale. Participants in the study suggested that the tool should notify them on the activities planned for each week and display their assessment scores.

Lund and Wiese [33] conducted a study on how people currently engage in the process of short-term planning. More specifically, they analyzed how participants use software and paper-based tools to engage in different planning behaviours. Using their results to elicit requirements for tools to support planning, they further proposed a User Interface (UI) for such a tool. Among the elicited requirements were the ability to rearrange items in the existing plan, use of imprecise times, set reminders, mark tasks as done and moving tasks to the later days. Similar conclusions were reached by Häfner, Oberst, and Stock [22] in their article on students' habits. They showed that tools to support planning must allow for flexibility, such as allowing imprecise times, moving tasks to the next day, and transferring tasks from other sources. They also found the need for marking tasks as complete to help motivation, and getting reminders to prompt beginning a task or preparing for an event.

2.3 Commercial Tools and Services

Taskito [68] is an example of a typical commercial *to-do list* application. A to-do list is perhaps the most simple example of a tool which supports task management. Taskito works as an extension of the standard to-do list concept, providing some functionality outside of the standard to-do list features. The user enters tasks which they want to complete, along with extra information which helps in organization, prioritization and further planning. Some of the extended functionality provided by the tool is the ability to monitor tasks which have been previously completed, grouping tasks by what time they need to be performed, and providing notifications to the user about upcoming tasks.

Timeliness [69] is another example of a to-do-list application with extended functionality. Timeliness has typical to-do-list functionality, and moreover, it keeps track of and displays an overview of upcoming deadlines and "do dates" (dates when tasks are planned to be performed). Thus, Timeliness provides additional support for time assessment behaviours outside of its basic to-do list features.

Jira [5] is an example of a commercial tool which supports *project management*. The tool exhibits the features of a standard to-do list application, but is extended with functionality with the aim of helping a team or an individual manage tasks in a project. Tools which support project management may, for instance, allow users to map dependencies between tasks, loosely plan when tasks should be started and completed, organize tasks through project boards or road-maps, outside of the basic functionality of to-do lists.

Google Calendar [17] is an example of a simple digital calendar and a general purpose tool for assessing time. Digital calendars let users manually create and edit temporal plans, which may in turn be used to assess how much time they have to spare and when they are available. Digital calendars are also commonly used for *timeblocking* which is a special behaviour of time assessment and planning, consisting of blocking out specific periods of time for tasks based on the time which is available.

Akiflow [1] is a newer tool which illustrates a trend of *combining* to-do lists with the environment of digital calendars. In this tool, the tasks are collected in a standard to-do list fashion, and alongside the list of tasks is a digital calendar. Tasks can be freely dragged from the to-do list and over into the calendar, which can be used for time assessment and timeblocking of these tasks. Akiflow also exemplifies the need of gathering all of one's tasks in a single place. This is facilitated through the tool's many ways of integrating with preexisting task management tools.

Clockwise [12] is a used to aid time assessment and planning of tasks, more specifically of meetings among people belonging to the same organization. The schedules of other "team members" is compared in order to propose meeting times and lunch breaks. The remaining time in the calendar can then be scheduled as "focus time".

Shovel [62] is a tool specifically aimed at supporting students and their needs for time assessment, time monitoring and planning. In a similar fashion to Akiflow, it also follows the trend of integrating to-do lists into digital calendar environments. The calendar can thus be used for easier time assessment. However, contrary to Akiflow, Shovel is specifically designed around study related tasks, which renders the tool less applicable for the management of more general tasks. In terms of task management, additional information about the user's semester and courses is used to speed up the process of adding tasks and assessing available study time. In addition to the time assessment and planning capabilities, Shovel also provides time monitoring of the tasks which have been completed, and can thereby provide insight into what tasks the user needs to assign more time to when planning.

Launched in the spring of 2022, **Reclaim AI [56]** is a calendar based tool which supports time assessment, time monitoring and time planning through automation. The tool lets the user specify *habits*, for instance lunch breaks, which are used to automatically block out time in the calendar. After time has been set aside for habits and other important events, calendar events dedicated to tasks which the user has specified will be *scheduled automatically* inside of the remaining available time. The user can then accept the plan which has been proposed by the tool or choose to edit it themselves. Judging by the amount of time dedicated to each task through the calendar, the tool will also try to monitor the amount of time spent on each task and provide progress reports. In addition to its time management features, Reclaim also supports integration with several digital calendar and task management tools, with the aim of speeding up the process of assessing time and adding tasks.

2.4 An Overview

After having examined the tools and solutions which have been presented thus far, it was discovered that they fell along two axes which explained many of their differences. These axes were:

1. The **generality** of the tool – how specific are the use cases which the tool provides support for?
2. The degree of **automation** present in the tool – among the task and time management behaviours which are supported by the tool, to what degree are they supported through automation?

The resulting plot from attempting to place the tools along the two axes can be seen in Figure 2.1. As one might expect, the tools which originate from related research are much more specific in their use cases compared to their commercial counterparts. Also, it is important to note that the two axes are not independent of one another. For instance, the more specific the use case, the more assumptions the tools are allowed to make about the user's needs and preferences. In turn, this opens up the possibility for more automation of the tools functionality. This assumption is a possible explanation of the "*generality-automation*" *front* illustrated by Google Calendar, Akiflow, TaskGenies, Clockwise, Shovel and MyLearningMentor.

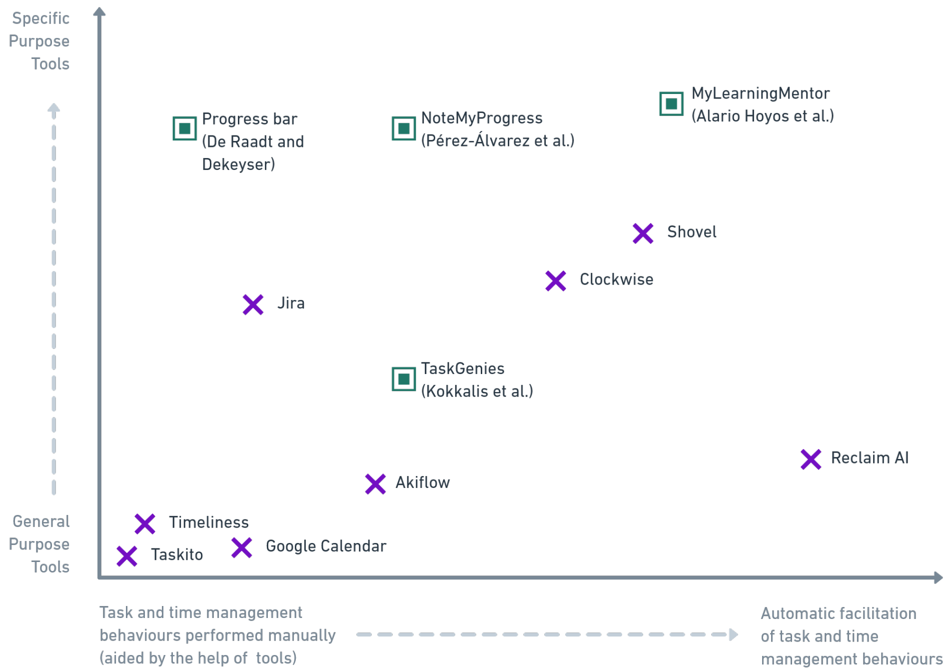

Figure 2.1: Plot of State-Of-The-Art Solutions along Two Axes.
Tools from related research are represented by a **square**; commercially available tools are represented by a **cross**.

An interesting trend which was observed among the examined tools is how they would become seemingly more *opinionated* when moving further out on each of the two axes. Naturally, moving along the generality axis would restrict *what* the tools may be used for. On top of that, moving along the automation axis would influence *how* the time management behaviours would need to be performed. For instance, in order to automate many of the time management behaviours, the tool Reclaim builds upon several presumptions of how its users would like to plan and monitor their time.

As previously mentioned, people show different preferences for how they want to partake in the various time management behaviours [33]. Tools which aim to support these time management behaviours therefore risk alienating certain users when they become more opinionated about *how* the users should partake in time management. However, this is only an overall trend found among the specific tools which were examined, and might not dictate how these kinds of tools are designed in general. As is illustrated by the plot of Figure 2.1, relatively few tools with a high amount of automation were examined. Therefore, the possibility of automated tools to support time management behaviours, which are also non-opinionated, should not be disregarded.

CHAPTER 3

Scheduling Theory

This chapter will provide a brief introduction into the family of problems known as *scheduling problems*, alternatively called *sequencing problems*. Scheduling problems are perhaps more common than one might first imagine, as they tend to emerge in many different, and seemingly unrelated, fields of work. The handling of shipping containers at ports, the assignment of working hours to nurses at a hospital, and perhaps most (in)famously, the scheduling of computer processes to be handled by CPUs are all problems which relate to scheduling in some way or another. Naturally, this chapter will give an overview of the scheduling theory limited to what may be applied in scheduling of *study related tasks*. Nonetheless, a general description of scheduling problems will first be provided, as general knowledge will be convenient for understanding the more special cases of scheduling problems which will be presented later on.

3.1 Classification of Scheduling Problems

A notation for describing scheduling problems, which has later become a common way of classification, is introduced by Graham, Lawler, Lenstra, and Kan [18]. The notation has later been used in the extensive descriptions of scheduling problems by Pinedo [49], Brucker [6] and Lawler, Lenstra, Rinnooy Kan, and Shmoys [29]. Here on, the following notation will be used to refer to different scheduling problems:

$$\alpha \mid \beta \mid \gamma$$

where:

α = a specification of the "machine environment"
β = a set of constraints and special characteristics
γ = the "objective function" to be minimized

The upcoming sections will give an explanation of the above notation as well as some extensions to the original notation which may be relevant to the problem of scheduling study related tasks. However, first and foremost, a description of the *jobs* to be scheduled will be given, as most of the later explanations will use this description as a basis.

Jobs

A job, sometimes referred to as a "task", is usually defined through a set of parameters which together describe its different aspects. For a job J_i, a *processing time* p_i is most often defined as an integer to describe how many units of time the job needs before it can be completed. Likewise, the completion time of a job C_i describes the absolute point in time when the job is completed.

A *deadline*, or due date, d_i is also used in most cases to describe when a job needs to be completed. If the job's completion time C_i surpasses its deadline d_i ($C_i > d_i$), the deadline will be considered "not met", and the job will be considered "tardy". On a slight side-note, in terms of scheduling, "tardiness" and "lateness" are not interchangeable, as *lateness* is defined as the general difference between a job's time of completion and its deadline ($C_i - d_i$), while *tardiness* is not considered unless the time of completion actually surpasses the deadline ($\max\{0, C_i - d_i\}$).

In some cases, it is preferred to ascribe a *soft* deadline d_i to some of the jobs, whilst ascribing a *hard* deadline $\overline{d}_i$ to the remaining jobs [13, 29]. The penalty given for a job not meeting its hard deadline is comparatively much harsher than the penalty for a job not meeting its soft deadline. Usually, the entire potential value of a job is lost the very moment it becomes unable to meet its hard deadline, while in the case of a soft deadline, a milder *cost function* is applied.

A cost function f_i is used to measure the incurred cost of completing a job at its completion time C_i. In the case of tardiness, for instance, f_i may be used to determine the consequential penalty of not meeting the deadline. A commonly used cost function for a job's tardiness is:

$$f_i = \max\{0, C_i - d_i\} \tag{3.1}$$

The function 3.1 is often denoted by T_i (T as in "Tardiness"), and results in a *linear* cost for a job which has surpassed its deadline.

A *weight* w_i is in many cases given to a job in order to provide a notion of prioritization. The weight of a job may, for instance, be factored into the job's cost function in order to *weigh* the costs of the job against the cost of other jobs.

A *release time* might be used to describe how a job will not be available for scheduling before a certain point in time r_i. The introduction of release times also has a tendency of increasing the complexity of problems which would otherwise be simple to solve [6, 29, 49]

Machine Environment (α)

The field describing the so-called machine environment contains just a single entry. For instance, $\alpha := 1$ will describe the simplest case where a *single* machine (or *resource* as it is sometimes called) is available to process the jobs. The single machine case will be the main focus of most of the scheduling problems discussed from here on out, as the case of scheduling study related tasks (usually) only involves a single student. However, there also exist more complicated cases were several machines are available for use; these should not be fully disregarded when examining the possibilities for scheduling study related tasks. There are several ways of notating multi-machine scheduling problems, depending on the relationship between the machines and the circumstances of the jobs. When all the jobs can be processed by any of the multiple machines, the machine environment may be described by one of the following:

- Pm when the processing time of a job is *identical* on all of the parallel machines.
- Qm when the machines are *uniform*, meaning a job's processing time is only dependent on the processing speed of the machine.
- Rm when the machines are *unrelated*, meaning a job's processing time is dependent on both which machine it is run on and the machine's overall processing speed.

There are also cases where a job might require several distinct *operations*, where perhaps the operations are sorted by precedence, or certain operations need to be performed by dedicated machines. These cases are often referred to as *shop* environments, where G is used to denote the general shop case. Many special cases of the general shop exist including Om for "open shop", Jm for "job shop" or Fm for "flow shop". The exact details of these classifications will not be explained in detail. Nonetheless, general knowledge of shop environments might provide a useful "tool" for modelling more complex scheduling problems.

Constraints and Characteristics (β)

The β field is used to further specify the constraints and characteristics of the jobs and machines on which the jobs will be processed. Both Graham, Lawler, Lenstra, and Kan [18] as well as Brucker [6] are quite meticulous about exactly what this field may be used to express, however, in several works done on scheduling problems, this field has also been used to express a wide array of additional characteristics.

The perhaps most common job characteristic described by the field β is the inclusion of *release times*, which have been explained earlier in this chapter. Often paired with release times is the notion of *preemption*, shortened as either *prmtn*, *prmp* or *pmtn*, which is a characteristic of jobs which can be temporarily put on hold and resumed afterwards without affecting the processing costs. If there, however, *are* costs for resuming a job, it will usually be specified in some way or another. It has also been mentioned earlier how release times often complicate scheduling problems. Conversely, the introduction of preemption may in some problems mitigate the complexity introduced by release times [29, 49].

The notion of *precedence relations* between the jobs is noted as *prec*. In short, if a job j_i has precedence over another job j_k, a schedule is feasible if, and only if, the completion time of j_i is before the start time of j_k. In addition to basic precedence relations, there are special cases of precedence constraints. If every job has at most a single successor and a single predecessor, the precedence relation is known as a *chain*; if every job has at most a single successor, it is known as an *intree*; and if every job has at most a single predecessor, it is known as an *outtree*.

In many scheduling problems, there are required *setup times* associated with different sets of jobs. The existence of a *sequence dependent setup time* is often denoted by s_{jk} and represents a mandatory setup time between jobs j_i and j_k. *Family dependent setup times* might also be required between the processing of different *job families* (sometimes called *job classes* [20]), which are a way of classifying jobs. The introduction of job families, usually denoted by *fmls*, may require a setup time s_g when the schedule switches to the job family g, or require a setup time s_{gh} when switching from family g to family h.

In much of the early work on scheduling theory, it was often assumed that the machines would be continuously available. In later work, this availability is rarely considered as a given. The notion of *breakdowns*, noted as *brkdwn*, are used to describe this aspect of scheduling, either by introducing periodical periods of unavailability [32, 40] or by describing the potential of breakdowns through a stochastic variable or function of time [15].

Less common is the use of *reject* to denote the option of rejecting jobs upon arrival [2, 13]. One might assume that rejection is always allowed, however, this is not necessarily the case under some of the more special circumstances which will be discussed later on in this chapter, and it would, therefore, be useful to specify.

Objective Function to be Minimized (γ)

As explained earlier in the chapter, the notation T_i is used as a shorthand for the linear cost function for tardiness. Likewise, L_i is used as a shorthand for the *lateness* cost function of a job, where lateness is defined through $C_i - d_i$. In the case of hard deadlines, a unit penalty U_i is often used as a cost function instead of L_i or T_i. The unit penalty cost function U_i is defined as:

$$U_i = \begin{cases} 1 & \text{if } C_i > d_i \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

So far, three different cost functions have been described. When describing an *objective function*, however, a function which takes every cost function f_i should be taken into consideration (unless care should be given to only *one* of the many jobs job). The most commonly used objective functions are derived either by looking the maximum among the cost functions of all jobs, for instance L_{max}, or by taking the sum of all the cost functions, as in $\sum T_i$. Graham, Lawler, Lenstra, and Kan [18] provides the classification $\gamma \in \{f_{max},\ \sum f_i\}$ for objective functions. Although it might not be entirely clear from this classification, the *average* of a cost function over all jobs, $\frac{1}{n}\sum_{i=1}^{n} f_i$, may also be used as an objective function. The weight of the jobs can also be accounted for, for instance, in the case of total weighted tardiness ($\sum w_i T_i$). In fact, a substantial amount of different cost functions have been formulated throughout the work on scheduling problems, although, the objective function is often composed of some linear combination of either the sum or maximum of U_i, T_i or L_i.

By using the notation which has been presented up until now, it is possible to formulate many of the common problems in scheduling. The problem of "minimize the total weighted completion time of a single machine environment with release times, preemption and resumable jobs”, for instance, can be formulated and communicated through a single line of notation:

$$1 \mid r_i, \, pmtn \mid \sum w_i C_i$$

It is useful to be able to explain even simple scheduling problems in a clean and explicit manner, as many of their solutions have lead to heuristics which have also proven useful for solving more complex problems. When the heuristics deal with the question of “what job to schedule next”, they are referred to as *dispatch rules*. Below is a couple of dispatch rules which lead to optimal solutions for some of the most fundamental scheduling problems, but also tend to give good solutions in the case of more complex problems:

- **Shortest Processing Time First (SPTF)**: Formalized by Smith [64] and commonly known as “Smith’s principle” or “Smith’s ratio rule”. The rule leads to optimal solutions for the problem of $1||\sum C_i$. The generalized rule also solves for the case of $1||\sum w_i C_i$ by dividing the weight of the jobs by their affiliated processing times (w_i/p_i), which is often described as Weighted Shortest Processing Time First (WSPTF).
- **Earliest Due Date (EDD)**: Also known as “Jackson’s rule” after the work of Jackson [24]. Initially proved to solve the problem of $1||L_{max}$, with modifications to the rule shown to be optimal for other problems. Furthermore, if there exists a schedule where all jobs meet their deadlines, then EDD is optimal.

By slightly modifying the EDD dispatch rule, Moore [37] developed a polynomial time algorithm for solving $1||\sum U_i$ which is commonly known as the *Moore-Hodgson* algorithm. The algorithm can be proven to not only be optimal in the basic case, but also extending to the case of $1||\sum w_i U_i$ [9]. The generalized problem solved by the algorithm has also been shown by Lawler and Moore [28] to be equivalent to the weighted knapsack-problem, thereby providing a general proof as well as valuable insight on how scheduling problems can be solved through dynamic programming. All in all, the Moore-Hodgson algorithm has become an important cornerstone in scheduling, as many of its extensions and modifications are useful tools in the solving of more complex scheduling problems.

An overview of the hardness of some of the most fundamental scheduling problems is presented in Figure 3.1. Many of these problems can be solved by clever use of the dispatch rules which have just been discussed, and optimal solutions for each of these problems have been known for decades. However, a seemingly endless amount of scheduling problems still remain to be solved for, especially under the more special circumstances (which may quickly arise when scheduling makes the leap from theory to practice).

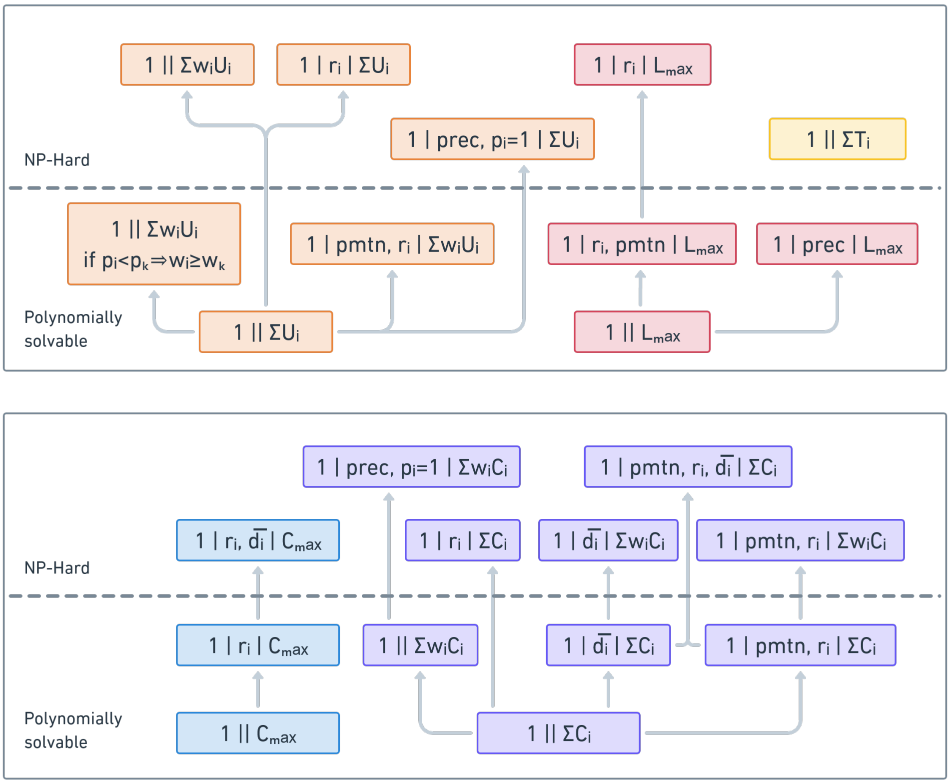

Figure 3.1: Illustration of the Hardness and Reducibility among Some Common Single-Machine Scheduling Problems (Based on works by Pinedo [49] and Lawler, Lenstra, Rinnooy Kan, and Shmoys [29])

3.2 Scheduling under Special Circumstances

Consider the case of scheduling study related tasks, or maybe the case of scheduling everyday tasks in general (if the former seems too far removed). This is a case of scheduling "in practice" where the requirements and circumstances, more often than not, are fuzzy and uncertain. For instance, a simple "processing time" for a given job can no longer be considered as *just* a processing time, but rather a number which, to a limited degree, estimates the *actual* processing time. A computer may perhaps be able to calculate the exact amount of time its processor will require to finish its given tasks; a student, however, is rarely able to the exactly estimate the amount of hours needed to complete their given coursework (or write their master's thesis). Only *after* the work has already been completed will they be able to provide a proper estimate. During a semester, the amount of course work is not necessarily known either, and will only become apparent gradually as the semester progresses. And what exactly is the end-goal, or objective, of the student? Is it to maximize the amount of assignment deadlines which are met throughout their coursework? Or do they wish to keep the overall tardiness of their work to a minimum? Are there perhaps schedules which will satisfy both of these objectives? And will a student, contrary to a machine, not require breaks? The following sections will provide a discussion on how these type of special circumstances can be considered when formulating and solving scheduling problems.

Multiple Objectives

So far, several objective functions have been introduced, and some simple solutions for how to optimize under them have been discussed. However, there also exists the possibility of a schedule which may optimize for *several* objectives at once, for instance both $\sum U_i$ and $\sum T_i$. Given that such schedules exists, it would be beneficial to examine and design algorithms are able to search for them in an effective manner. The notion of *multi-objective functions* are introduced in order to formalize these types of multi-objective optimization problems, and is usually denoted by expanding the field γ into several sub-fields $\gamma^{(1)}$, $\gamma^{(2)}$, ... , $\gamma^{(n)}$. With this notation, the objectives are most often interpreted *hierarchically*, for instance, as in "optimize for $\gamma^{(2)}$, but only under the optimal which has already been found for $\gamma^{(1)}$". It is also possible to express multi-objective functions which are *not* inherently hierarchical, for instance: $\gamma := a \sum f_i^{(1)} + b \sum f_i^{(2)}$, where a and b express a different weighting, or preference, for the objective functions $\sum f_i^{(1)}$ and $\sum f_i^{(2)}$ respectively.

Often, the notion of a "non-dominated" or "Pareto optimal" solution is used to explain *optimality* when multiple objectives are considered. These terms describe solutions where there exist no other feasible schedule which improves upon one objective without also impairing one of the other objectives [41, 75]. More formally, a non-dominated schedule σ' can be defined using the following conditions:

- For every feasible schedule σ_i, schedule σ' performs no worse than σ_i in any objective.
- For every feasible schedule σ_i, schedule σ' is strictly better than σ_i by at least one objective.

Finding these optimal solutions, which from here on will be referred to as "non-dominated", are usually the end-goal of algorithms made to solve multi-objective optimization problems. In the case where *several* non-dominated solutions are found through a single run of an algorithm, a "decision maker" or "expert" will need to decide which of the solutions is the most desirable. The decision maker may be yet another algorithm with its own heuristics or an actual human.

It might not come as a big surprise that most multi-objective scheduling problems are, in fact, hard to solve. Scheduling is, however, not the only field of computer science where hard problems arise. If the scheduling problem can be properly formalized as, for instance, a linear program or a mixed-integer linear program, methods from constrained optimization may be worth considering. If the problem is defined in terms of linear constraints, the polynomial run-time algorithms for solving linear programs may be used effectively. In most cases, however, the scheduling problem does not lend itself to be described through continuous variables alone, but rather as an mixed integer program. Nonetheless, by relaxing these integer programs, effective *approximation* algorithms may still be devised through methods of linear programming [29, 51].

Additionally, several meta-heuristic algorithms like *local search* techniques [48, 57], *genetic algorithms* [48, 73, 76] and *swarm optimization* [8, 38, 77] have been applied to the harder scheduling problems whilst producing satisfying results.

A scheduling problem which may seem related to scheduling study related tasks is the problem of "timetable scheduling" at universities and colleges [4, 10, 16, 63, 70]. Despite both problems considering the schedule of students, it needs to be emphasized that they are far from identical. The timetable problem is, however, an example of scheduling in the realm of education where the presence of many different objectives often leads to $\mathcal{NP}$-hardness. Seemingly, as a result of the multi-objectives which define the timetable problems, meta-heuristic algorithms are especially popular for deriving satisfying solutions.

Maintenance, Breaks and Unavailability

Theory aside, in a realistic setting, one might expect whatever (or whoever) is performing the jobs to require breaks and maintenance in order to prevent unexpected breakdowns; this was briefly discussed earlier in this chapter. A machine or computer cannot work indefinitely into the future without a period of maintenance; neither can a student work tirelessly on the same task without breaks. These breaks can be accounted for by scheduling them *before* the scheduling of tasks, thereby modelling these down-times as *availability constraints* or *periodic maintenance*. The introduction of such constraints generally adds complexity to scheduling problems. Although, if the jobs are *resumable*, meaning they do not incur additional costs by preempting, much of the added complexity can be mitigated by slightly altering the existing solutions of the problems. The problem characteristic of "availability constraints with **resumable** jobs" is denoted by r-a in the work of Lee [30]. Likewise, "availability constraints with **non-resumable** jobs" is denoted by nr-a. Lee [30] also explains how Moore-Hodgson's algorithm still solves for the problem of $1|r\text{-}a|\sum U_i$ simply by regarding the length of the unavailable period together with the jobs' completion time. This line of thinking holds for most single machine problems with resumable jobs and availability constraints. However, one need still beware, as a few otherwise simple problems like $1||\sum w_iC_i$ suddenly become $\mathcal{NP}$-hard when subjected to r-a. A further discussion of the case of *semi-resumable* jobs and some extended notation has also been made by Graves and Lee [19].

Online Scheduling Problems

An *online* problem is characterized by how only a *part* of the problem is known upon inception. The remainder of the problem will only become revealed as the solution progresses. In other words, an online algorithm will need to be able to process its input piece-by-piece as the input becomes gradually known. The algorithm is therefore not able to take the entire input into consideration when it begins solving a problem, which usually gives it some impairment compared to *offline* algorithms for the same problem. When comparing online algorithms to their offline counterparts, some notion of *competitiveness* is often mentioned. A k-competitive algorithm describes an online algorithm where its objective value is never more than k times the optimal value of the solution to the offline problem.

Usually, by simply following one of the dispatch rules mentioned earlier, a "decent" level of competitiveness can be achieved for online scheduling algorithms. For instance, the SPTF heuristic, when modified to work with online problems, has been shown by Vestjens [72] to provide a 2-competitive algorithm for the problem described by $1 \mid online \mid \sum C_i$, which is also the *lower bound* for this online problem when using deterministic algorithms. However, the use of *randomized* algorithms, where parts of the solution is left up to chance, may for several problems lead to an even lower bound for the competitive ratio than their deterministic counterparts [60, 66, 72]. For example, the aforementioned problem of $1 \mid online \mid \sum C_i$ has a lower bound of 1.582 when randomized algorithms are applied [72].

The field of online scheduling is certainly broad. It will, however, not be discussed here in further detail. Nonetheless, the existence of such problems, as well as the limitations they introduce, are important to keep in mind when regarding the real-time aspects of scheduling study related tasks.

Scheduling in the Face of Uncertainty

First and foremost, there are several ways of modelling the notion of "uncertainty", and the ways of thinking about scheduling problems may depend on which model is chosen. In the case of *stochastic* models for scheduling, it is assumed that the uncertain variables (usually the processing times and release dates of jobs) come from known *distributions* of probability. Since the objective function no longer can be expressed in terms of "non-stochastic" numbers, it will need to be modified. Often, it is done by considering the *expected* value of whatever variable the goal is to minimize. In the case of $1||\sum C_i$, for instance, the corresponding problem with a stochastic completion time may be expressed as $1||\sum E(X_i)$, where $E(X_i)$ denotes the expected completion time[1]of a job j_i.

As with the deterministic problem of $1||\sum C_i$, the stochastic problem of $1||\sum E(X_i)$ can be solved by using the SPTF dispatch rule if the shortest *expected* processing time is considered. This is true regardless of which probability distribution the completion time is described by [52]. However, in most other cases, the underlying probability distribution will significantly affect how the problem needs to be solved. For instance, Akker and Hoogeveen [2] have shown that if the processing times are described by either a *gamma distribution* or a *negative binomial distribution*, the problem of $1||\sum U_i$ can practically be solved as its deterministic counterpart. However, if the distribution is *normal* or *uniform*, the problem goes from being solvable in polynomial time to $\mathcal{NP}$-hard, and it therefore requires different algorithms to solve.

The work of Akker and Hoogeveen [2] is also an example of how, in the case of uncertainty, objective functions may be derived through something other than estimated values. Instead, they define their objective function in a *stochastic* sense by looking at how many jobs are "stochastically on time". A job is deemed to be *risky* if the chance of it being on time is lower than some predefined probability; the job will in this case be *rejected* before the scheduling process begins. The notion of risky

[1] With stochastic scheduling, X_i is often used instead of either p_i or C_i as these notations are imbued with different meanings in the related fields of *statistics* and *queue theory* [52].

jobs can be used to modify, for instance, Moore-Hodgson's algorithm to also solve for stochastic scheduling problems.

When the probability distribution can be assumed, the estimated values can also be used to formulate mixed-integer programs [26]. As mentioned earlier in the chapter, the integer programs can be relaxed and solved as linear programs in order to derive polynomial time estimation algorithms.

Using probability distributions to model the problem stochastically is, however, not the only option of dealing with uncertainty. Much work has also been done in formulating scheduling problems through the use of *fuzzy* numbers and membership functions [23, 31, 39]. The notion of fuzziness also lends itself for devising objective functions which focus on a schedule's *robustness*. If the amount of overlap between the fuzzy job intervals can be reduced, the schedule will be more robust in the cases were the performance is lower than expected.

Lastly, it is important to keep in mind the possibility that the uncertainty cannot be modelled in any applicable manner. If one cannot assume the distribution of the variables or model them through some fuzzy membership function, the best solution may be to lean back on a simpler approach; as Pinedo [50] describes it: The more complicated the randomness, the simpler the solution ought to be. The scheduling can also be iterated on *reactively* when the uncertainty causes a drift from the proposed schedule [21, 25, 36, 53].

CHAPTER 4

Study Preferences and Habits

At the beginning of the spring semester of 2022, a survey was conducted in order to garner insight into RQ2, which relates to the study habits of students and their essential needs when planning. In addition to providing information directly related to the research, the survey results would also be used to improve certain usability aspects of the planning tool, of which include appropriate default values for many of the tool's user settings.

4.1 Description of the Survey

Students at The Norwegian University of Science and Technology (NTNU) were the main target for the survey. In order to recruit students, several methods were applied. Firstly, a poster was designed, printed and posted near popular lecture halls around campus. Secondly, recruitment messages were posted online on various student message boards. These messages contained an explanation of the survey and its purpose as well as a hyperlink to an online form.

The survey was structured around several questions, the first of which was about how many hours per day the student would spend on course mandatory work (like assignments and projects) in addition to how much time they would spend on non-mandatory study activities. The students' "best-case estimates" for the hours spent on these activities was also asked about. Additionally, the students would describe to what degree they plan each study day, how far ahead they planned, and to what degree they would adhere to the plans they had made. In relation to the students' study habits and routines, they were asked about when their study day usually starts and ends, how long they usually work for before taking a break, as well as how long a break usually lasts.

Some questions were aimed at what the students prioritize when planning their study days. These questions included how much they prioritize meeting deadlines, variation in their work and the frequency of breaks, which the students were asked to rate on a Likert scale from 1 to 5. A more open-ended question was also asked regarding any other factors which the students might consider when planning. Ultimately, the students were asked if they already use a calendar or similar tool during their planning.

4.2 Results

By the end of the survey process, a total of 66 submissions from students (30 female and 36 male, age ranging from 19 to 27) had been collected. Some of the responses were clearly from 5th year master's graduate students, who were hypothesized to have distinctively different habits and preferences than other students. A point of consideration therefore arose around whether or not to include these data points. In the end, the master students showed little to no difference in study preferences compared to the rest of the population, and they were therefore included in the final results. Few confirmatory statistical tests were performed on the resulting data, however, several exploratory methods were utilized in order to gauge overall trends among the students. Descriptive statistics on the survey is presented in Table 4.1.

Statistical Summary of Study Preferences

Variable	n	$\hat{\mu}$	Std. Dev	Sample Median
Length of Study Day	66	8h 9m 36s	1h 55m 12s	7h 55m 12s
Start of Study Day	66	09:07	1h 12m	09:07
End of Study Day	66	17:17	1h 55m 12s	17:16
Actual Time Spent Doing Exercises Per Day	66	3h 23m 24s	1h 46m 12s	3h
Optimal Time Spent Doing Exercises Per Day	66	4h 5m 4s	1h 56m 24s	4h
Actual Additional Study Time Per Day	66	1h 28m 48s	1h 24m	1h
Optimal Additional Study Time Per Day	66	2h 23m 24s	1h 45m	2h
Study Session Length Minimum	66	1h 45m 50s	1h 12m 4s	2h
Study Session Length Maximum	66	1h 55m 50s	1h 10m 28s	2h
Break Length Minimum	66	24m 46s	21m 25s	20m
Break Length Maximum	66	32m 16s	31m 44s	20m
Degree of Planning	66	2.95	1.01	3.00
Number of Days Planned For At Once	66	4.00	2.55	3.00
Degree of Plan Adherence	66	3.14	0.88	3.00
Importance of Deadlines	66	4.47	0.79	5.00
Importance of Variation	66	2.47	1.08	2.00
Importance of Break Frequency	66	2.82	1.20	3.00
Number of Courses	66	3.68	1.36	4.00
Age	66	22.08	1.69	22.00

Table 4.1: Statistical Summary of Responses from the Survey on Study Preferences

Notably, when comparing the different prioritizations for planning a day of studies, it was shown that 55 of the 66 respondents rated the **importance of deadlines** higher than both **importance of variation** and **importance of break frequencies**. In contrast, only 2 of the 66 respondents rated **importance of variation** the highest.

A more in depth look at the results in the form of bar charts can be found in section A.1 of the appendix. Furthermore, the R code used to calculate the results presented in this chapter is provided in section B.1.

A Search for "Types" of Students

It was hypothesized that there might exist different groups, or types, of students which display different study preferences and habits. In an effort to uncover these kinds of patterns, both correlation tests and k-means clustering were utilized.

Correlations were tested between every continuous variable using the Spearman method. The resulting correlation matrix, with every non-significant correlation removed, can be seen in Figure 4.1. Many of the strong correlations are as one might expect, for instance the correlation between the start and end times of the student's study day, as well as the correlation between the student's number of courses and their degree of planning. However, a few of the correlations, albeit not as strong, are perhaps more notable, for instance the correlations between the students' degree of planning and the importance they place on variation ($r(66) = .30$, $p = .0149$).

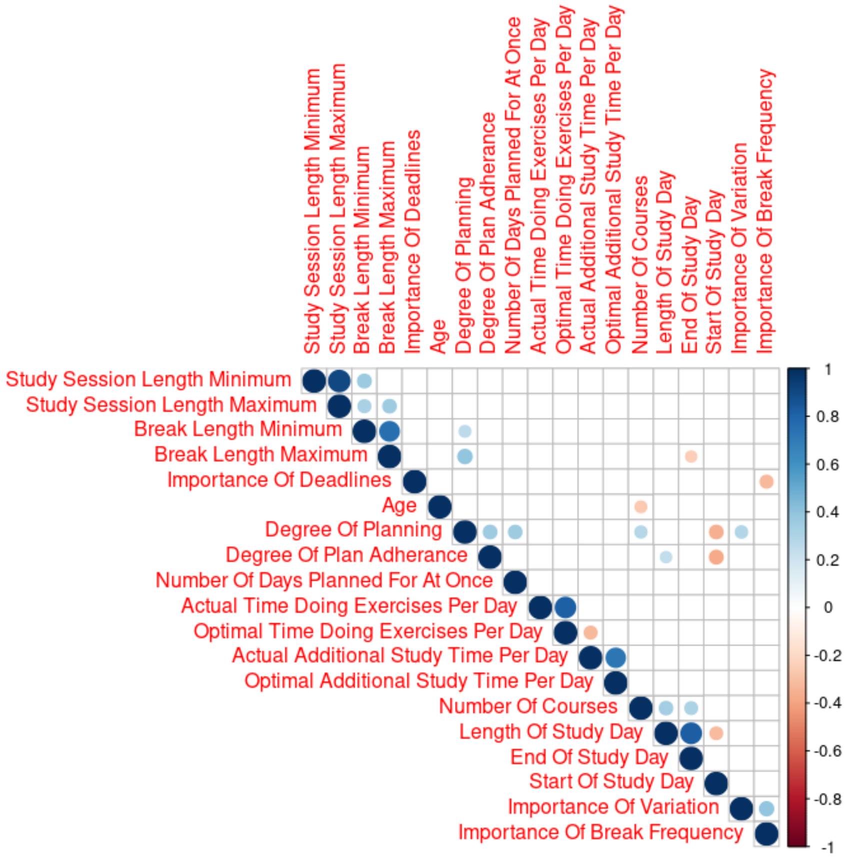

Figure 4.1: Correlation Matrix with Non-Significant Correlations Excluded

As an exploratory method to gauge if there exists different types of students with different preferences, k-means clustering was performed on the data. Several methods were used to evaluate what number of clusters should be used, among which were the *elbow method*, the *silhouette method* and the *gap statistic method*. The elbow method did not provide any clear result, whilst the gap statistic method suggested only a single cluster (which naturally would not provide any new information). Therefore, in the end, a number of 2 clusters were chosen based on the result from the silhouette method. The results of the clustering is displayed as a projection onto a 2-dimensional scatter-plot in Figure 4.2. Additionally, the coordinates of the 2 cluster centres (along normalized axes of the data) are shown in Table 4.2.

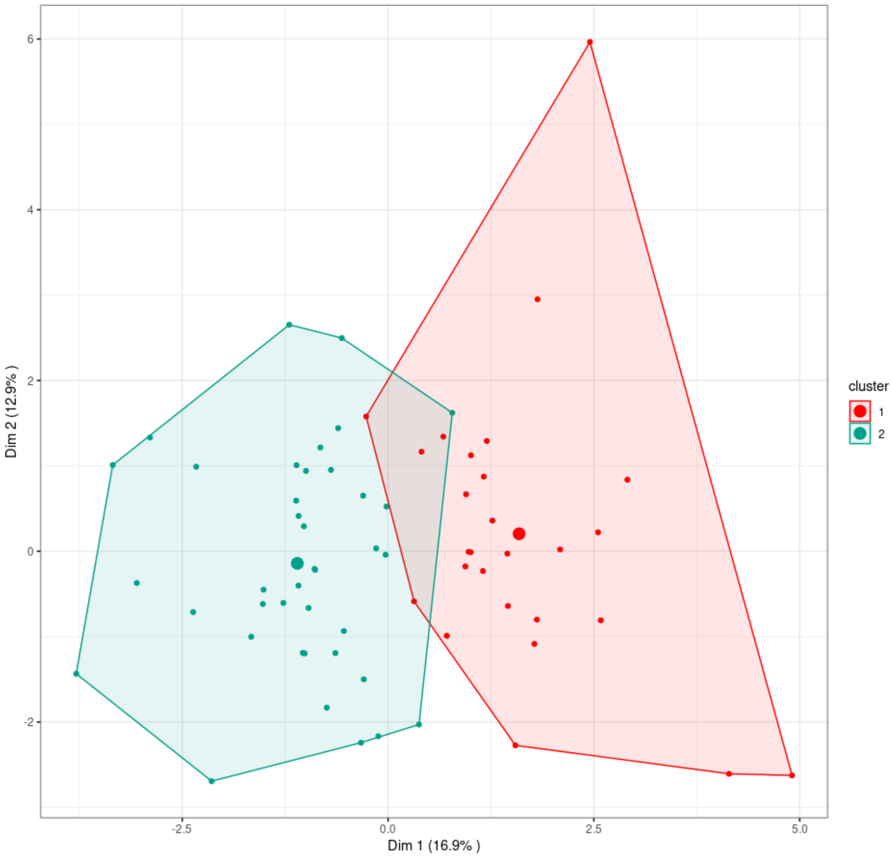

Figure 4.2: 2-Means Clustering of Study Preferences.
(The results have been projected onto the span of the two strongest principal components derived from the data)

Centre of the 2 Clusters Derived through 2-Means Clustering

Variable	Centre of Cluster 1	Centre of Cluster 2
Actual Time Doing Exercises Per Day	0.22	-0.32
Optimal Time Doing Exercises Per Day	0.19	-0.28
Actual Additional Study Time Per Day	-0.36	0.51
Optimal Additional Study Time Per Day	-0.34	0.49
Start Of Study Day	0.01	-0.02
End Of Study Day	0.01	-0.01
Study Session Length Minimum	-0.47	0.68
Study Session Length Maximum	-0.49	0.71
Break Length Minimum	-0.39	0.56
Break Length Maximum	-0.37	0.54
Degree Of Planning	-0.31	0.45
Number Of Days Planned For At Once	-0.07	0.10
Degree Of Plan Adherence	-0.07	0.10
Importance Of Deadlines	-0.01	0.01
Importance Of Variation	-0.36	0.52
Importance Of Break Frequency	-0.30	0.43
Number Of Courses	0.01	-0.01

Table 4.2: Centre of the 2 Clusters Derived through 2-Means Clustering. (The coordinates for every variable have been normalized, meaning they lie on the interval of $[-1, 1]$)

There is a seemingly large difference between the average **break lengths**, **study session lengths** and **study time per day** between the two clusters. Perhaps more notably, among the variables which describe what the students prioritize when planning, the variable **Importance of Variation** shows the largest distance between the two clusters, suggesting that there is a specific group of students who prefer variation and another group of students who are indifferent. In comparison, each cluster is centred around nearly the exact same value of **Importance of Deadlines**, suggesting that planning for deadlines is about equally important for each cluster of students.

A Summary of Free-Text Responses

When examining the free-text responses which were included in the survey, they were found to contain information related to several different topics. The responses were therefore categorized and sorted through a qualitative method. The method yielded the following themes:

- How do students choose what task to work with next?
- External needs which need to be accounted for when planning
- The ability to cooperate on tasks
- Tasks which need to be planned for *besides* the ones which are directly study related
- Tasks or events with predetermined time periods

Considering how students choose what task to work with next, two of the respondents mentioned that they would look for the tasks with the *shortest completion time* rather than considering the closest deadline. A couple of respondents also said they would want to focus on the task which requires the most amount of "brain power" to get done, especially when starting their day.

Among the external needs when planning, the *location* where the tasks needed to be performed was mentioned twice. It was also mentioned by others that they would need to consider their current mood and "energy level" when planning.

The "availability of other groups members" was mentioned by a couple of respondents as something they consider during planning of their day of studies.

On the topic of other mandatory tasks besides the ones which are directly study related, two respondents mentioned that they need to set aside time for voluntary work for the student association which they participate in. Others also mentioned how they plan to set aside time for going to the store as well as training.

Finally, looking at tasks and events with predetermined times, 8 respondents mentioned that they would need to plan around meetings which are present in their agenda. Two respondents mentioned how they plan around their regular lectures for every week, while a couple more respondents stated that they plan around the time when they always do their training.

CHAPTER 5

Design

This chapter will propose a hypothetical design for a tool meant to support a selection of time management behaviours among students. However, first and foremost, description of how this design was derived will be provided. Of special interest is how **time assessment** and **time planning** of study related tasks may be facilitated. Consequently, support for task management will also be discussed, as it is assumed that in order for the student to plan around their tasks, the tasks will need to be managed by the tool in some way or another. This assumption is largely based on an observation made in chapter 2, where every tool which supports planning was found to also facilitate task management. In summary, the "goal" of the tool is to support students during their process of planning study related tasks and assess what time they have available.

5.1 Task Management

The inception of the tool's design came from a simple to-do list. The rest of the design would build upon this premise in order to bridge the gap between a basic list of tasks and a fully fledged tool to support time assessment and planning. However, regardless of the tool's eventual functionality, the first step would be to enable the student to load the tool with their study related tasks. Two methods were proposed for achieving this. Firstly, the student should be able to manually enter their tasks along with the information which should be considered during planning. Secondly, the tool should be able to take advantage of the tasks which are already defined in the student's Virtual Learning Environment (VLE). At NTNU, "Blackboard Learn" is the learning environment which is currently in use, and by establishing a connection between the VLE and the planning tool, it may be used to automatically query the collection of tasks which have been given to the student.

5.2 Assessing the Student's "Available Study Time"

The next step in designing the tool was to figure out how the tool would be informed of the student's available time. It was presumed that the tool would not be able to suggest satisfying plans to the student without also knowing when the student is available to study. Therefore, several methods were conceptualized for enabling students to specify their available time. The first of which was to allow the student to connect the tool to their personal calendar in some way. From the survey on study habits and preferences presented in chapter 4, it was discovered that 50 out of the 66 respondents (76 percent) already used a calendar for planning their study days. The reasoning is similar to connecting the tool to the student's VLE; by integrating with their calendar, the tool can automatically acquire information on when the student is available to study.

One should also regard the remaining 24 percent of students who reportedly do not use a calendar to plan for events and chores. Additionally, there might be students who do not feel comfortable providing data from their personal calendar, or students who perhaps use a calendar daily, but do not bother to enter every single task. Naturally, these students would require other methods of informing the tool about their available time. Some potential methods would be to let the student specify their preferred study time through the tool's UI, either through a dedicated *settings* menu or by answering questions during an *on-boarding* section. The student would then suggest which days of the week they want to study, as well as a dedicated study period for each day. Another possibility is to enable the student to add elements directly to their personal plan after it has been generated, thereby letting them specify periods of unavailability.

5.3 Scheduling of Study Related Tasks

If the student's available time and tasks are known, it will become possible for the tool to schedule these tasks algorithmically and thereby propose a personal plan for the student. Note that there is made a distinction between the *schedule* generated by the tool and the *personal plan* which is eventually presented to the student. From here on, the "personal plan" will refer to what the student will observe and interact with through the tool's UI, whilst the "schedule" is the output of the algorithm representing a sequence of tasks. This section will mainly discuss the latter, as well as how a scheduling algorithm may be designed given the current knowledge of students' habits and preferences.

Students Are Not Machines

First and foremost, it is important to recognize that students are not, in fact, machines. The underlying scheduling problem may therefore not be modelled with the level of certainty which may otherwise be assumed in the case of more "orthodox" scheduling problems. If a student is to perform one of their tasks, there is no completely fail-safe method for estimating how much time the task will need to be completed. The simplest way of achieving an "adequate" estimate might be to simply ask the student directly, thereby letting them enter into the tool their own estimated completion time for every task. However, as has been observed time and time again, most people tend to be overly optimistic when asked to estimate times for tasks [7, 47], and students are not assumed to be exceptions to this rule.

Dealing with Uncertainty

As was mentioned in chapter 3 on the act of scheduling under uncertainty, how the problem may be solved is dependent on *how* the uncertainty can be modelled. It did, however, not feel appropriate to assume much about the uncertainty which is present in this problem, especially the completion times of tasks. Aside from people's tendency to underestimate completion times of tasks, little was known about how exactly these estimated completion times would compare against the tasks' actual completion times. As discussed in chapter 3, if the *probability distribution* of the estimated completion time is known, there *are* indeed several known solutions which are known to perform well. Conversely, given the current lack of knowledge on completion times and their intrinsic uncertainty, the seemingly most feasible solution is to entrust the student with providing appropriate completion times.

Dealing with Breaks

Another concern is regarding the students' need for taking **breaks** in between study sessions, and how this behaviour consequentially affects the time which is available for scheduling. Every single respondent from the survey of chapter 4 reported that they take breaks throughout their day. There are mainly two possibilities (perhaps more) for how these breaks may be taken into consideration when the tool schedules a plan for the student. The possibility which was first considered was to model these breaks as their own type of job – a "break job" – and schedule them as any other study related task. For every study day, a set of "break jobs", each with their own release time and soft due date, would be added to to the list of jobs to be scheduled, and an objective function would ensure that these breaks are not too tardy (while still allowing for some flexibility). However, as discussed in chapter 3 on scheduling, the introduction of release times and non-preemptable jobs would increase the complexity of the problem. Therefore, an alternative method of handling study breaks was considered. The method consists of modelling the breaks as *periodic times of unavailability* based on the student's study preferences. These breaks may then be regarded alongside the other periods of unavailability in order to infer the student's available study time – an arguably simpler approach than the one which was first suggested. In other words, the breaks are not scheduled in the same sense as study related tasks, but are rather considered and *prearranged* before the actual scheduling has begun.

Variation

Aside from the need for breaks, another important observation was made from the survey on study preferences. There is seemingly a need for "**variation**" among certain groups of students, and evidently, this need is stronger among the students who reportedly plan their day more thoroughly. Therefore, enabling some sort of variation in the schedule might be important to consider. However, the survey's respondents did not specify exactly what kind of variation they would prioritize when planning. Do the students wish to work with several different *courses* per day, or would they, perhaps, rather work with a certain task for some amount of time before switching to a different one? Without making further inquiries, it cannot be known for certain how this variation should be modelled. Some type of variation may, however, be *assumed* in order to formulate hypotheses, which may in turn be tested.

With the Student as a Decision Maker

In contrast to the inherent uncertainty, the need for breaks and the requirement of variation, there are other special characteristics of the problem which may make the scheduling *easier* rather than more complex. If, for a moment, one would take a step back and view the larger problem at hand – not only the scheduling problem in isolation, but the problem of supporting a student's time management with the help of a digital tool – one may observe that an "expert" decision maker is available at all times. All things considered, the student is the one who will initiate the scheduling process, and the student is the one who will eventually accept or reject the resulting plan. With this observation in mind, it may be possible to relax many of the problem's constraints.

It may be assumed that there exists students who exhibit additional needs for planning outside of the "normal" needs of the general population. For instance, in the survey of chapter 4, it was uncovered that some students would like to work with certain tasks at a specific point in time in order to cooperate with other students. It would be possible to model the scheduling problem after every peculiar need which a student might exhibit, and eventually design an algorithm to solve for this inevitably complex problem. Given that such a perfect algorithm exists, the student may spend some time to specify all of their peculiar needs as constraints before a schedule is generated, and they would thereby entrust the algorithm with producing a satisfying result. However, given appropriate methods, the student may just as quickly make the required modifications to the plan *after* the scheduling has been performed. To summarize, the scheduling algorithm cannot account for every possible need which students may have without requiring an endless amount of questions before scheduling. The students, however, may account for their own needs as long as the the tool provides them with the correct tools for editing their personal plan. Therefore, the overall goal of the scheduling algorithm becomes not to generate an *optimal* schedule, but rather an *adequate* one, which may be further modified by the student in order to achieve a satisfying plan.

In order to further simplify the scheduling problem, it may be assumed that the student would want to generate a new schedule every time they are made aware of additional tasks. Consequently, the scheduling problem may be thought of as a problem where every task has a release time of zero. There is therefore no need to model the jobs with release times. The "online-ness" of the problem, however, does not disappear under this assumption, as the scheduling algorithm may still be able to account for new tasks possibly being added in the future. Exactly how the online-ness may be dealt with depends on how the scheduling problem is modelled in the end. Nevertheless, if it may be assumed that students will want to iterate on their schedule when new tasks are added, this behaviour may function as a good heuristic for improving the competitiveness of the algorithm.

Additionally, the thought of a schedule which is frequently iterated upon translates well into the notion of *reactive* scheduling, which was mentioned in chapter 3. Considering the amount of uncertainty surrounding the problem, and how it should be dealt with, the possibility of scheduling reactively might be the best way of handling this dilemma. The student may realize that they might have provided a wrong estimation of a task's completion time. Therefore, the tool should enable the student to easily update the estimated completion time and iterate on the schedule given their updated knowledge.

Jobs, Constraints and Characteristics

Students have several different types of tasks to deal with. One of them are the mandatory tasks provided through the student's VLE. These tasks were used as a basis for modelling the jobs to be scheduled by the tool. This leads to the following attributes being associated with each job:

- A processing time p_i, provided by the student
- A hard deadline $\overline{d}_i$
- A cost function U_i
- (A release time $r_i \coloneqq 0$)

Additionally, jobs are *preemptive* and assumed to be *resumable* with a negligible amount of incurred cost. As discussed earlier, the student will also have periods of unavailability, which are inferred by events in their calendar, their preferences for breaks, their preferred study time as well as other unavailable periods which they have manually entered into the tool. The scheduling problem will therefore be described by r-a, as is suggested by Lee [30].

Objective

It is assumed that the students' highest priority is to minimize the amount of tasks which exceed their deadline. In this case, the natural way of describing the objective function of the problem would be $\sum U_i$. Moreover, one should not disregard the possibility of tasks being of different importance to the student. For instance, the student may partake in a course with many obligatory tasks and hand-ins; however, relatively few tasks are needed to pass the course. On the other hand, the student may partake in another course with relatively few tasks, but which are all highly important in order to pass. The student would in this case place a higher *weight* on the tasks from the last course than the ones from the first. The objective function is then described by $\sum w_i U_i$.

Lastly, there is the concern of accommodating an appropriate amount of variation in the student's plan. Based on the results from the survey of chapter 4, it may be assumed that this should not be prioritized over meeting the deadlines of tasks. Achieving a required amount of variation is, however, a valid objective in its own right. Should both of these criteria be achieved, the problem is best described as a multi-objective optimization problem. However, it is not easy to tell how exactly the need for variation should be described as an objective function. Nonetheless, regardless of *how* the notion of variation is eventually described, the corresponding objective function will be denoted by $\mathcal{V}$. Thus, the multi-objective function of the problem is described as:

$$\sum w_i U_i^{(1)},\ \mathcal{V}^{(2)}$$

Possible Ways of Modelling the Scheduling Problem

Considering what has been discussed so far, there seems to be a number of ways in which the problem of scheduling study related tasks may be modelled. The model depends mostly on what the students wish to prioritize, although many other aspects of the problem are also subject to interpretation. Additionally, briefly discussed was the potential trade-off between the accuracy of the model and the amount of constraints which need to be specified by the student before the scheduling can be performed. In the end, three different models of the problem, with different levels of complexity, are suggested to be implemented for testing:

1. $1 \mid r\text{-}a \mid L_{max}$
   Minimize the maximum lateness over all jobs. An optimal solution is found by following the dispatch rule of EDD.
2. $1 \mid r\text{-}a,\, \overline{d}_i \mid \sum U_i$.
   Minimize the amount of tardy jobs given hard deadlines. An optimal solution is found by using the modified Moore-Hodgson's algorithm.
3. $1 \mid r\text{-}a,\, \overline{d}_i \mid \sum w_i U_i^{(1)},\, \mathcal{V}^{(2)}$
   Minimize the amount of tardy jobs while also accommodating enough "variance". A feasible solution may be found through the use of meta-heuristic algorithms.

In the best case, solutions for *several* of the modelled scheduling problems would be implemented in order to test which is the most befitting. Nevertheless, some actions need to be accommodated for by the tool regardless of which model is chosen. These include the user's ability to enter tasks alongside a deadline and an estimated completion time, the ability to set periods of unavailability and prearrange breaks, and lastly, the ability to reorder tasks and iterate on their schedule after the scheduling has been performed.

5.4 Presenting the Student with an Automatically Generated Plan

So far, this chapter has examined the scheduling of study related tasks, in addition to how this problem may be modelled and solved. The solution to the underlying scheduling problem is regarded as *what* the student should be presented with. The remainder of the chapter will, however, be dedicated to exploring *how*, as in: How should the suggested solution to the scheduling problem be presented to the student?

The Personal Plan and its Population of Elements

One concern is how the problem needs to be conceptualized in order to derive a satisfying scheduling algorithm. Another concern is how the problem should be conceptualized for the sake of making it understandable for the student. These two conceptualizations are not necessarily the same. A certain model of a problem might work well for solving it algorithmically; however, this model might not be the best way to communicate the problem to a student who has their own idea of how it should be solved. The way in which the resulting schedule will be presented is through a timetable-like plan. The appearance of this plan needs to communicate what role the different elements play and what they represent, whilst plan's behaviour upon interaction must be intuitive in order to let the student shape it to their liking.

Study Sessions

This is regarded as one of the most fundamental elements of the plan. As a result of the preemptive scheduling and periods of unavailability, each job will most likely need to be divided into *several* sessions. Each session occupies a specific period of time in the plan, the sum of which constitutes the required time for completing their affiliated job. The study sessions should also be interactable, as to enable the student to move them around as they see fit. Some basic "drag'n'drop" functionality should be implemented in order to make the interaction possible, as this is a method most users of digital calendars should be familiar with.

Breaks

Breaks are another fundamental element of the plan, as it is something every student plans around (as may be assumed by the survey results). The breaks are prearranged according to the student's preferences before the scheduling is performed, and therefore, the resulting study sessions will fall in between the breaks and the rest of the unavailable periods. The breaks should, however, be communicated differently from the other periods of unavailable time. A distinction should perhaps be made between the periods of time which are absolutely unavailable for *both* the scheduler and student, and periods of time which the student has *decided* should be unavailable. The notion of breaks is better described by the latter. In a way, breaks are "slack", in the sense that they are conceptually unavailable times for the scheduler, however, the student might do with them as they wish. The same way a student may want to rearrange the placement and length of study sessions, they might want to change the placement of certain breaks in their plan. Therefore, it must be communicated that the functionality for editing study sessions is also applicable for breaks.

Time Outside of Preferred Study Time

For each day of the week, the student should be able to specify a period of time for which they would like to study. The remaining time *outside* of these time periods needs to be communicated to the student as "time which they themselves have decided should be unavailable". This is similar to how the breaks are conceptualized. Unlike the breaks, however, it does not make sense for the student to move these periods around in their personal plan; not in the same way at least. In short, these unavailable periods may be described as just as "slack" as breaks, but not interactable in the same way.

Calendar Events and Manually Added Unavailable Periods

Together with breaks and the outside of preferred study time, the calendar events constitute the unavailable time of the student. Unlike the rest of the unavailable time, which are "slack", the events imported from the student's calendar are considered "strict", meaning that they are not directly editable by the student without some sort of forewarning. In addition to importing events from their calendar, it was also argued that the student should be able to add similar types of unavailable time to their plan manually. These will behave in the same way as calendar events, although presented through a slightly different appearance in order to distinguish the two types of plan elements.

Locked Elements

The notion of *locked elements* refers to either study sessions or breaks which the student has purposely and manually moved to their current placement in the plan. If a student makes the conscious choice of having a certain study session or break at a certain point in time, the scheduling algorithm should be able to respect this decision when it is asked to generate a new schedule. The student should, in turn, be given proper indication that these elements are indeed *locked* and marked as unavailable time for future runs of the scheduling algorithm.

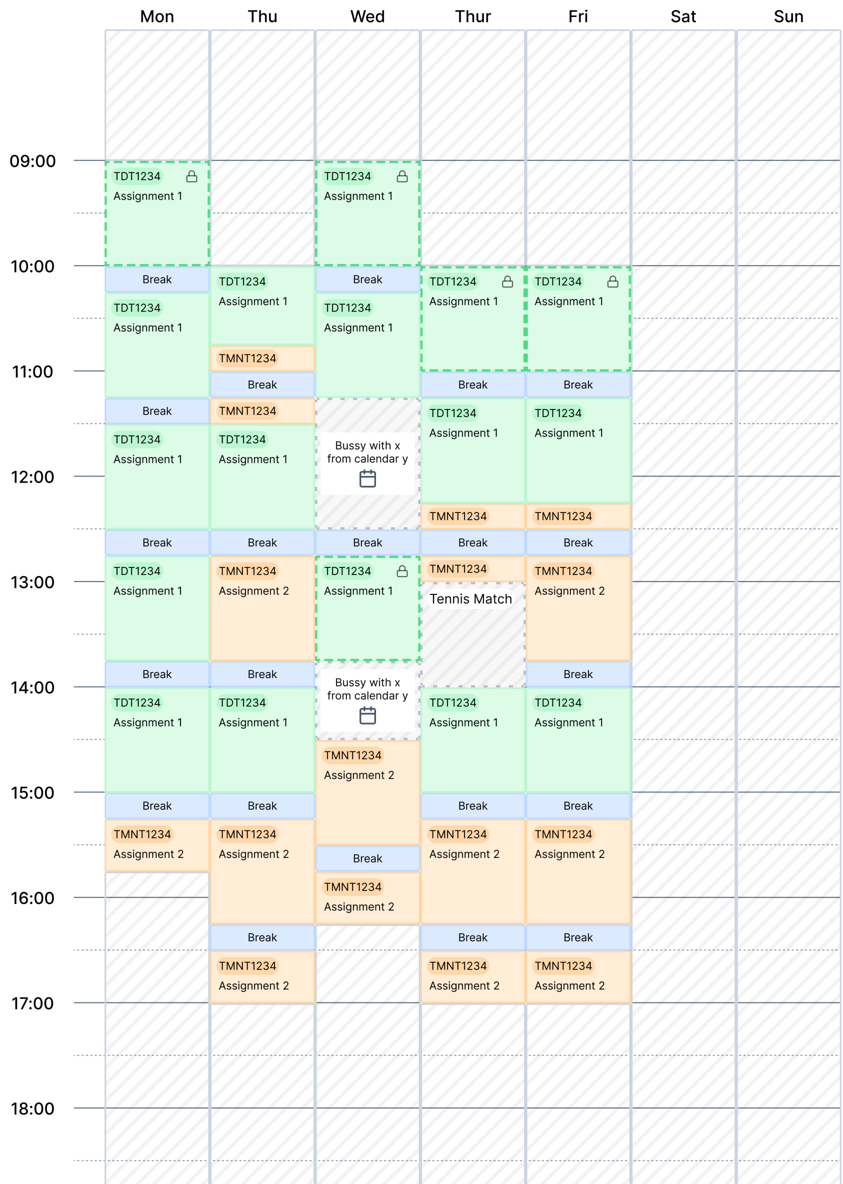

Figure 5.1: A Conceptual Illustration of the Personal Study Plan and its Inherent Elements

Figure 5.1 shows an illustration made to help visualize what a personal plan might look like containing the different plan elements. These elements include **study sessions** (in colours like green or orange), **breaks** (in blue), **locked study sessions** and **locked breaks** (marked by a stippled outline), **calendar events** and **periods of unavailability** (marked by a striped background and stippled outline) and **time outside preferred study time** (marked by a striped background).

Rules of Interaction between Elements

A few questions might arise if one tries to imagine what happens when certain plan elements are placed on top of one another. How should the tool react to this type of interaction? Should this type of action simply be rejected? As was briefly mentioned in the descriptions of the different plan elements, certain elements may be considered *strict* while others may be considered *slack*. It was also mentioned how the student would not be able to edit a calendar event without forewarning. This forewarning is also applied for the rest of the strict elements like locked study sessions, locked breaks and manually added times of unavailability. These strict elements, which have been intentionally marked as unavailable, should refrain from being edited by accident. If the student is aware of this restriction, they may be assured that they will not make unintentional mistakes while they modify their personal plan. Thus, the student will be able to freely move around the rest of the plan elements without worry.

There are, however, some nuances to the behaviour which was just explained. Study sessions may be considered as strict under certain circumstances. If the plan is *active*, which in scheduling terms means that the plan represents the only feasible schedule, there will no longer be enough time to perform any given task if one of its study sessions are removed. Study sessions of an active schedule may therefore also be considered strict.

Additionally, if a strict element is placed upon another strict element, the result will, unsurprisingly, become two strict elements placed on top of each other. There is however, no possibility of lessening the amount of unavailable time after such an action. The action of placing a strict element on top of another strict element, or a slack element for that matter, may be performed by the student without forewarning.

The Self-Adjusting Behaviour of the Personal Plan

In order to further aid the student in the process of editing their personal plan, it might be worth letting the plan continuously adjust itself based on the edits which the student makes. There are many ways of going about this, although the simplest method is perhaps to let the scheduling algorithm re-schedule the rest of the plan after the student makes an edit. The motivation behind this method is that the student, after having used the tool for a while, will already be familiar with how the scheduling is performed; adding additional, separate behaviours for self-adjustment of the plan might only lead to confusion. In the case of locked study sessions and breaks, these would be considered as constraints of the available time when the re-scheduling is performed. Moreover, the time which has been set aside for locked study sessions would need to be deducted from the estimated processing time of the jobs which the sessions are dedicated to.

CHAPTER 6

Implementation

This chapter will describe the process of implementing the ideas and concepts proposed in chapter 5 in the form of a tangible tool. The overall goal of the implementation process was to provide a tool which may be further used to evaluate the ideas for scheduling and planning functionality which were discussed in the chapter on design. At the end of this chapter, a description of the final tool and how it may be used by a student is also provided.

6.1 The Design of a Graphical User Interface

Before the design described in chapter 5 was realized as a fully-fledged tool, many of the design ideas were implemented as a low-fidelity prototype. Along with this prototype, a graphical user interface was introduced in order to communicate the design's ideas and features to the student. The graphical user interface and overall presentation of the prototype was similar to the conceptual drawing presented in Figure 5.1 of chapter 5. After the prototype was made ready, it was used to perform usability tests on a total of 9 students from the Computer Science course at NTNU. Eventually, several issues related to usability were improved upon based on the results from these tests, although very few of the fundamental design ideas were changed.

Some resulting **points of consideration** are also worth mentioning from the usability tests which were performed. Firstly, a potential integration with the Blackboard VLE was overall well received (although the integration was entirely simulated in the prototype). The act of adding tasks manually was perceived as cumbersome by many, and therefore, they would rather rely on the automatic task-gathering through the connection with Blackboard.

The importance of properly explaining the concepts of “available time” should not be understated. Much confusion seemed to arise from the calendar integration specifically. The test subjects wondered what exactly they were importing from their calendar. Some thought of the calendar as a means to import additional tasks, in a manner similar to the integration with their VLE. Some test subjects, however, made the connection between their calendar events and their available time, although seemingly, they did not think of how these two concepts would be used in tandem for scheduling their tasks.

In an effort to counteract the confusion around the calendar integration, the process of connecting the tool to the student's personal calendar was added as part of an **on-boarding process**, which new users of the tool may undertake in order to set their basic preferences. Not only did much of the confusion surrounding the calendar integration seem to disappear following its inclusion in the on-boarding process, the on-boarding process was also seemingly efficient for explaining other important concepts of the tool. For instance, by letting the student set their preferences for breaks just before they would be asked to connect to their calendar, the students seemed to make up a mental model of the tool where both calendar events as well as breaks were connected to their available time.

It was also observed that test subjects, in an attempt to edit their plan, would instinctively perform gestures associated with drag'n'drop functionality. This type of functionality was therefore heavily prioritized while implementing the final tool.

6.2 The Tool and its Features at the End of Implementation

Other than the additions to the tool which have been described in the section on the graphical user interface, the tool would be implemented to accommodate for most of the functionality presented in chapter 5. A few features were, however, altered by the end of the implementation process. These alterations should be given some attention:

(1) The connection with the Blackboard VLE in order to automatically fetch the students tasks was omitted from the implementation. Thus, the only way for the student to add their tasks would be to add them manually.

(2) Proper integration with several digital calendars became too demanding. For instance, the process of integrating with Google Calendar (outlined in section C.1 of the appendix) would not be completed in time for testing the tool. Therefore, the acting calendar integration would be through the use of manual upload and download of ICS-files.

(3) The default model of the underlying scheduling problem was $1|r\text{-}a|L_{max}$, and a simple scheduling algorithm based on the EDD dispatch rule was therefore used. Additionally, other dispatch rules, such as SPTF, were also made available for the student through the their personal to-do list in the tool.

An Overview of the Tool's Different Views

In total, four distinct views were implemented to fit the functionality of the tool. These views include: The "Tasks" view, the "Study Preferences" view, The "Manage Calendar" view and the view of the "Personal Plan". Additionally some views for on-boarding purposes were implemented in order to let new users of the tool specify their study preferences and habits before moving on to using the rest of the tool. Figure 6.1 illustrates the group of views and how they connect with each other during use of the tool.

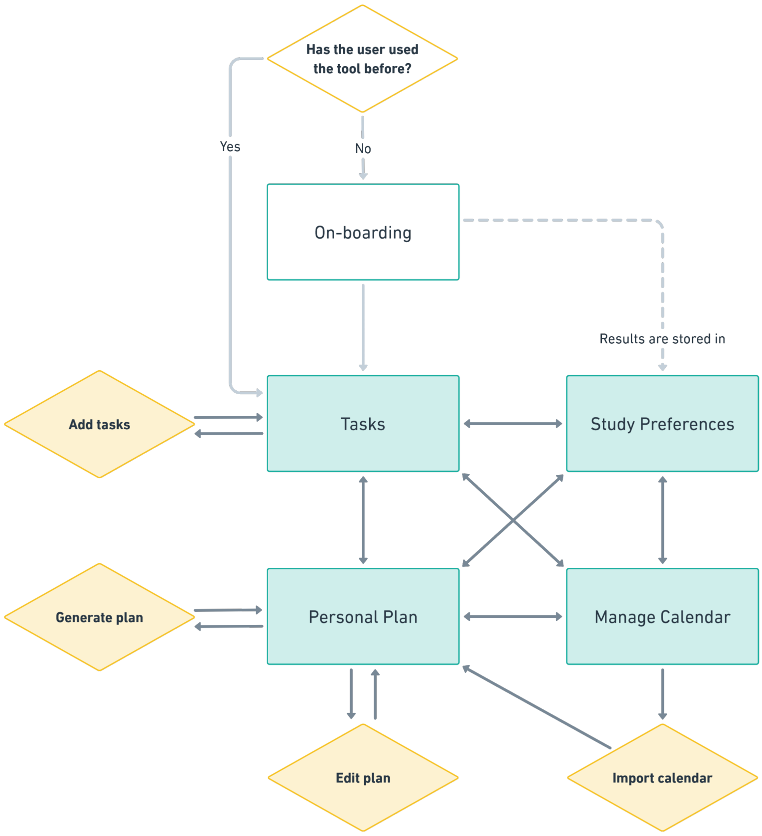

Figure 6.1: A Flowchart Illustrating How the Different Views of the Tool Are Connected

The On-Boarding Process

Figure 6.2 and Figure 6.3 display two of the screens which the student is presented with during the on-boarding process of the tool. In total, the on-boarding consists of 5 questions which were structured as follows:

1. Study preferences and habits
   a) For how long do you prefer to work with a task/course before you take a break?
   b) For how long do you prefer your breaks to be?
2. Available time
   a) Which days of the week do you prefer to work with your studies?
   b) When do you prefer to start and end a day of studies?
   c) In order to provide more accurate times for when you are available, you may also upload events from your calendar (a prompt to upload an ICS-file is displayed).

In order to accelerate the on-boarding process, the answer fields for several of these questions were pre-filled using the median values from the survey on study preferences presented in chapter 4.

Figure 6.2: A view of the on-boarding process. The student is asked how frequent and how long they prefer their breaks to be.

Figure 6.3: A view of the on-boarding process. The student is asked how long they prefer to study each day.

Figure 6.4: The View of "Tasks". The title says "Your study related tasks", while the subtitle informs the student that they may rearrange tasks to change their priority. The order which is displayed in the figure has been sorted by *earliest due date first* (EDD). The "Legg til oppgave" button at the bottom of the screen enables the student to add additional tasks to their to-do list.

The To-Do List

After completing the on-boarding process, the student is sent to their list of tasks, or the *to-do list* as many would call it. This is also where recurring users are sent upon logging into the tool. Initially, this list will be empty, as it requires the student to manually add their study related tasks.

After some tasks have been added to the list, as is shown in Figure 6.4, they may be sorted according to the student's liking, where the sorting will function as the dispatch rule during scheduling. The default sorting, based on the survey of study preferences in chapter 4, is EDD. A few other pre-defined sortings like SPTF were added based on some of the "rarer" study preferences which were reported.

The list of tasks also allows for marking the tasks as "done" when they have been completed by the student. Additionally, tasks with a relatively close deadline would have their text highlighted in red, while tasks with an exceeded deadline would have their text displayed with a ~~strike-through~~.

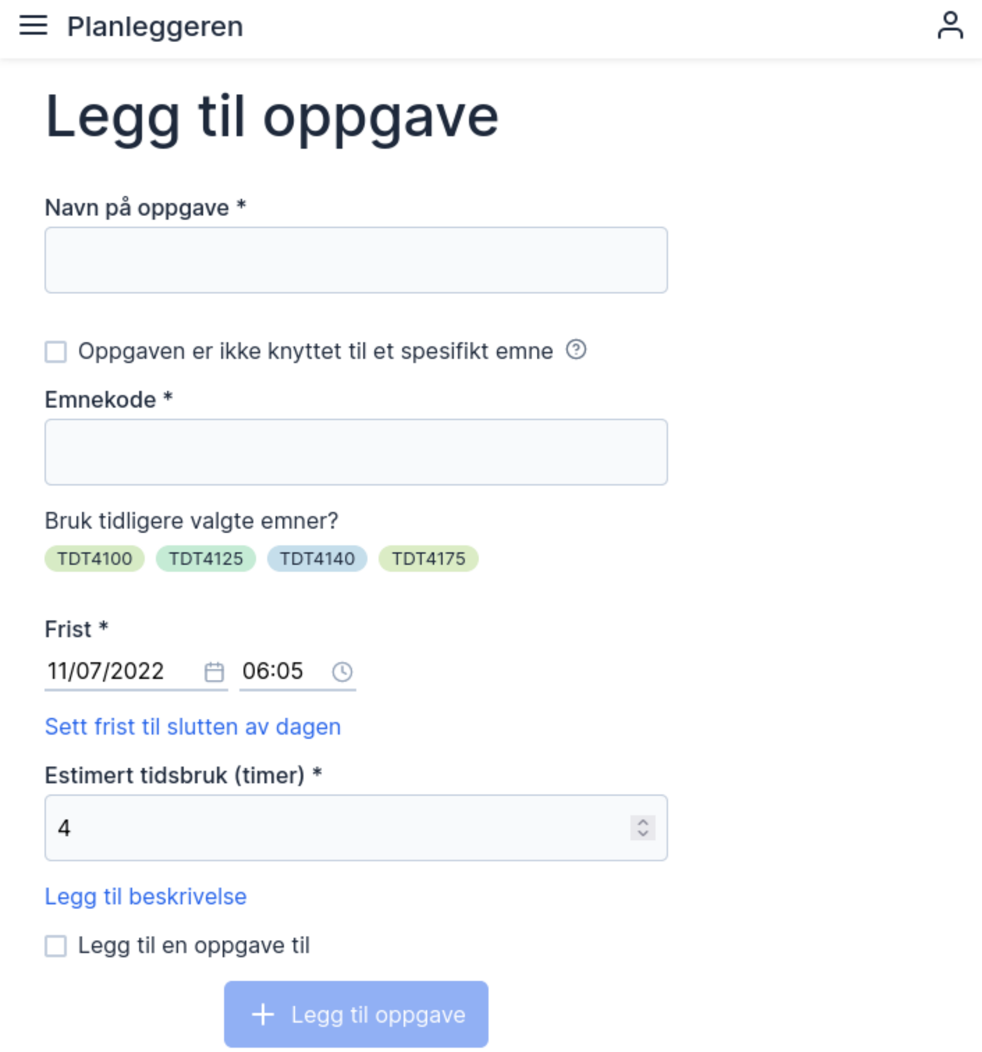

Figure 6.5: The “Add Task” View

When the student wants to add a new task, they would need to do so through the form which is presented in Figure 6.5. This form requires a name for the task which is to be added, a deadline and an estimation of how long the task will take to complete. Optionally, the student might connect the task to a given course by appending the course's *course code*, either by entering the code manually or choosing among the codes which have previously been used for previously added tasks. If the student chooses to enter the code manually, while they type, they will be provided with “quick-fill” options from which they may choose; these options correspond to existing courses at NTNU.

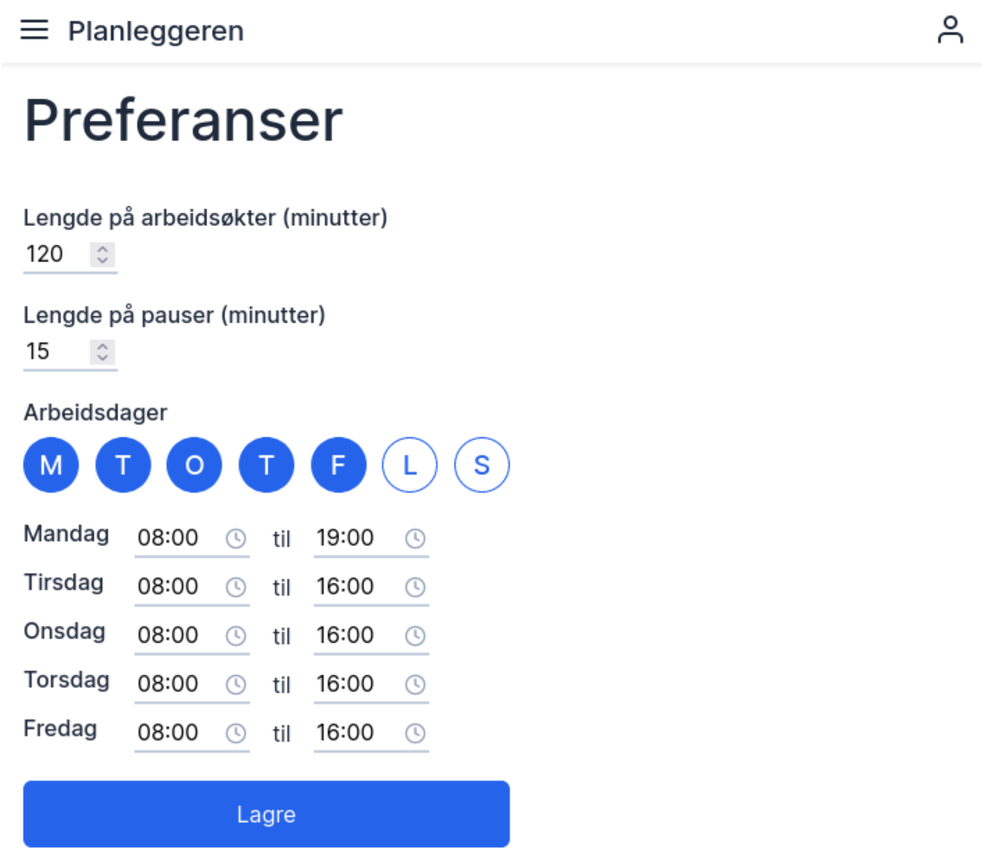

Figure 6.6: The "Study Preferences" View. The fields are labelled "Length of work sessions (minutes)", "Length of breaks (minutes)" and "Work days". The student may specify different start and end times per work day.

The Student's Study Preferences

The view of study preferences presented in Figure 6.6 is where the student would navigate to in order to change the preferences which they entered during the on-boarding process. Additionally, in contrast to the on-boarding process, where the student would want to move through quickly, the view of study preferences allows the student to be more *specific* about their preferences. In the current implementation, this mainly refers to how the student may change their preferred study period individually for each day of the week. The values which the student has specified through these fields are what will eventually determine how their plan will look like, as these values decide the prearranging of breaks and the availability of time for scheduling.

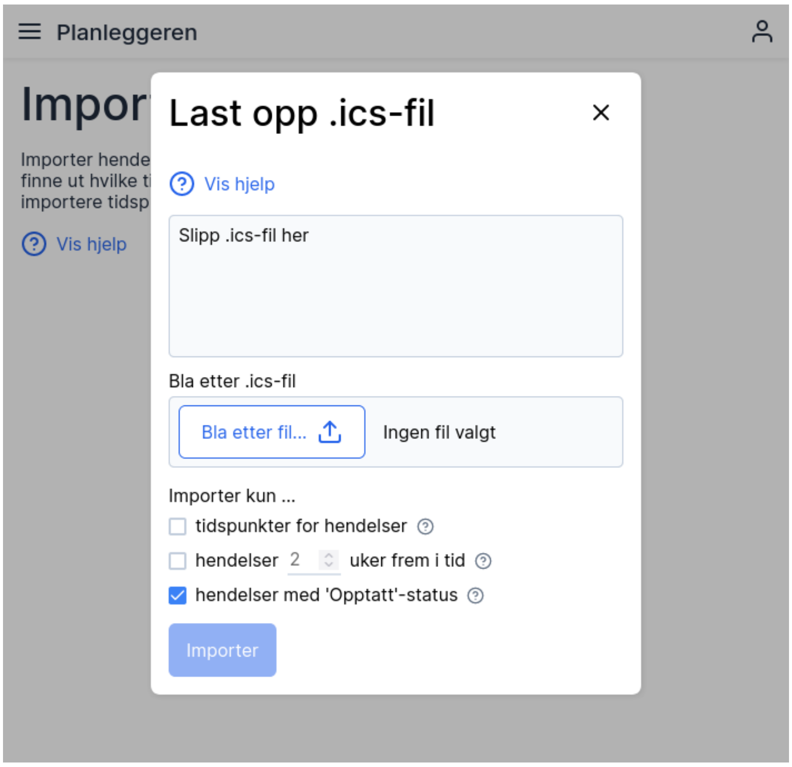

Figure 6.7: The "Manage Calendar" View. The title of the modal says "Upload .ics-file". The button below translates to "Show help". Below the help-button is a prompt for the student to upload an ICS-file. Moving further down, there are few check-boxes which specify what data should be imported from the ICS-file.

The Calendar Integration

Another separate view was implemented in order to let the student import their personal calendar. The method of importing is through the use of an ICS-file, which the student will need to export manually from their calendar tool, for instance, Google Calendar. In the calendar management view, as shown in Figure 6.7, a few extra features were integrated. The first of which is a "show help" button, which would display text for guiding the student through the process of exporting and importing their calendar. Another feature is the ability to further specify which of the events from the calendar should be imported. It was noted during the earlier usability tests that certain students are considerably specific about which calendar events they mark as "busy" and which they mark as "free". The option to only import "busy" events is therefore provided.

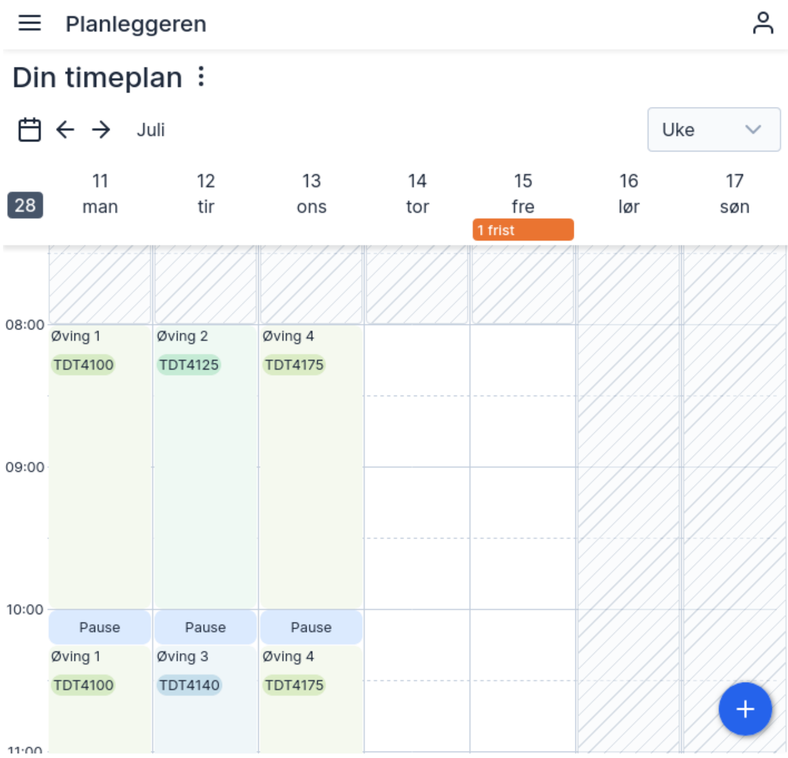

Figure 6.8: The View of the "Personal Plan". The smaller, blue coloured elements of the plan represent *breaks*, while the other distinctly coloured elements represent *study sessions* dedicated to the tasks "Assignment 1", "Assignment 2", "Assignment 3", and "Assignment 4". One of these tasks are due on Friday the 15th, which is represented by a *deadline* in orange placed above this day in the plan.

The Personal Plan

The view of the personal plan may, informally speaking, be described as the "meat and bones" of the tool. This is where the student is provided a personal study plan based on the tasks which they have entered and the study preferences which they have specified. Moreover, following the rules of interaction between plan elements described in chapter 5 on design, the plan enables the student to perform further modification. In short, the tool enables the user to drag and drop elements of the plan around, and thus alter their it as they see fit, without the worry of making unintentional planning mistakes.

Figure 6.8 shows how the personal plan of a student might look like. In addition to the methods for modifying the plan and the self-adjusting behaviour of elements, as described in chapter 5, the view of the plan provides a few other features. One of which is the ability to manually add elements such as extra study sessions and breaks to the plan by the use of the "add" button located located in the bottom right corner. These will then be regarded as "locked" elements upon placement.

CHAPTER 7

Qualitative Evaluation

This chapter aims at explaining the *qualitative* methods which were used to evaluate the tool which has so far been described through chapters 5 and 6. The next few sections will provide a description of the methodology which was followed throughout the evaluation process, whilst the last section of this chapter, the "Results", will report the resulting findings from the evaluation.

The main purpose for the qualitative evaluation was to uncover information and gain insight into RQ2, which regards potential needs among students who use tools for time management. Additional attention was also given to RQ4 and its two sub-questions. The first of these sub-questions, **RQ4A**, was defined as: "Among students, what is required in order to accept an auto-generated task plan?". The other part of the question regards the students' perceived usefulness of a tool for supporting time management. **RQ4B** was defined as: "Among students, what is the perceived usefulness of a tool for supporting task planning?".

The tests were performed as a cross between interviews and usability tests. The format of the interviews which were conducted can be categorized as an *informal conversational interview* combined with an *interview guide approach.* The characteristics of informal conversational interviews according to Patton [44] are that questions emerge from the immediate context and are asked in the natural course of things; there is no predetermination of question topics or wording. In an interview guide approach, the topics and issues which shall be covered are specified in advance, in outline form; the interviewer decides the sequence and wording of questions in the course of the interview. In other words, a list of questions and tasks were prepared in advance, but the wording and sequence of these questions were allowed to vary according to the situation, and questions which emerged during the tests were asked along with follow-up questions.

7.1 Circumstances of the Tests

The tests were all conducted in a meeting room located at campus where one researcher took the role as an interviewer, while another researcher took notes while helping participants if they got stuck with tasks performed outside the tool. A single student would participate at a time, and they were given access to test the tool through a web browser of their choice on their personal computer. The participants were not given a prepared computer, as the tests would ask the participants to import their personal calendars, which would require them to log in to their personal calendar. Login information and exported calendar files would then be stored on the provided computer, which would be a breach of privacy and data laws. Participants were supplied with pre-made user accounts which would be erased after the tests.

The test participants were recruited from NTNU, mostly from computer science related majors, and the tests were performed late in the semester in May, at which time most courses had finished their compulsory assignments.

7.2 Test Format and Conduction

Every test which was conducted followed the following format:

1. The participant was welcomed and introduced to the test and the researchers. They were shown to their seat and were told to use their personal computer for the test. The purpose of the test was explained. The participant was told to think aloud and to ask any questions they would have during the test. They were assured that the tool was what was being tested, and not the participant's abilities.
2. The participant was informed that a user had been made in advance which would be deleted after the test. Permission for making an audio-recording of the test for the purpose of transcription and analysis was asked and granted.
3. The participant was asked to log into the tool using the login information of their pre-made user account.
4. The participant was allowed to step through the on-boarding process in the tool. The on-boarding process consisted of introducing the tool and to ask the participant about when they preferred to start and end a study day, how often and how long they preferred to take breaks, and which days of the week they preferred to study.
5. The participant was asked to enter their study related tasks for the coming week.
6. The participant was asked whether they used a digital calendar to organize their everyday events. If they used such a calendar, they were asked to import it into the tool.
7. The participant was asked to generate a personal plan.
8. The participant was asked whether they would use the generated personal plan, and if not, why.
9. The participant was asked whether they wanted to make any changes in the generated personal plan, and if the answer was yes they were asked to attempt to make those changes.
10. The participant was asked about how much control they felt over the resulting personal plan.
11. The participant was asked about whether they felt that there was something the tool did not account for while generating the personal plan.
12. The participant was asked whether they would use such a tool in their everyday lives for planning their studies. If the answer was yes, they were asked about which type of tasks they would use it for.
13. The participant was asked about how comfortable they felt about having a program plan their studies.

14. The participant was asked about how useful it felt to have such a personal plan generated for them.
15. The participant was asked about potential advantages using such a tool could provide.
16. The participant was asked about potential disadvantages using such a tool could provide.
17. The participant was asked to fill out a System Usability Scale (SUS) form (as described by usability.gov [71]).
18. The test was concluded and the participant was given compensation for participating in the test.

7.3 Limitations

As described by Patton [44], a weakness of an informal conversational interview is that it may require a greater amount of time to collect systematic information because it may take several conversations with different people before a similar set of questions has been posed to each participant in the setting. The topics covered by the prepared questions did not cover all the topics which arose naturally during the tests. Because of this, not all participants were able to comment on the same topics, which made for less focused results during analysis. Such an interview format should therefore require a greater number of tests in order to account for its unstructured nature.

Not all planned features could be implemented in time for the qualitative tests. One of these missing features was the ability to drag and drop study sessions and breaks in order to move them in the timetable. Another missing feature was a calendar integration which would remove the need for the student to import their calendar manually by exporting and then importing an ICS-file and instead allow them to synchronize their events from their calendar automatically. Likewise, a request for an integration with BlackBoard – the online learning platform used at NTNU – was denied, which would have let the tool retrieve the student's tasks and reduce the need to enter them manually. These missing features may potentially have lead to poorer usability during the tests, which in turn could affect how the participants chose to respond to certain questions.

There were some bugs in the tool at the time of testing. Some bugs lead to participants getting stuck while performing tasks, necessitating help from the researchers in order to troubleshoot and solve the problems. Troubleshooting situations occurred in 3 of the 8 tests. These troubleshooting situations may have given the participants a skewed impression of how the tool functions in practise, which may have lead to biased responses.

The sample population, as represented by the test participants, was not necessarily representative for the population which would actually use such a tool, which would consist of students from different universities and fields. As mentioned earlier, the participants mainly consisted of students from the Computer Science course taught at NTNU. Students from other fields may have different approaches to study planning, and may have differing proficiency with using these kinds of tools.

In an experiment, *demand characteristics* refer to an experimental artifact where the participants form an interpretation of the experiment's purpose and subconsciously change their behaviour to fit that interpretation [58]. One such artifact is the *Good-Subject Effect*, where the participant exhibits behaviours designed to support what they perceive to be the preferable hypothesis [42]. Due to the lack of counter-measures against demand characteristics in the qualitative testing, it is possible that the results were affected by unnatural behaviour.

The tests took place in a controlled testing environment in a room located at the university campus. Under realistic conditions of usage, the tool might be used in high stress environments or in areas with poor internet connections. The testing environment was therefore not completely realistic.

7.4 Processing of the Results

The results were processed by first transcribing the recorded audio from the interview tests. Informative quotes were then extracted from the resulting transcriptions. Finally, the quotes were compared in order to determine common themes which were further categorized.

7.5 Results

In total, 8 students at NTNU participated in the tests. Several bugs and usability issues were uncovered, and feedback was used to improve the tool before the next testing phase.

Goals for Usage

On the topic of *why* they would use such a tool, the participants expressed several imagined benefits. One of the benefits imagined was *cognitive offloading*, as in letting the student offload parts of the planning process onto the tool instead of having to keep track of the entire process themselves.

Other participants imagined *time saving* as a potential benefit, as the tool would let them "skip" large parts of the planning behaviour associated with time management.

Another benefit, which was frequently brought up, was the ability to quickly attain an overview of tasks and their associated deadlines. If all tasks, due dates and planned events are made clearly visible by the tool, the student no longer needs to make a large effort to assess what needs to be done and when.

> *"I think it's nice that I can collect my tasks. I don't necessarily need to create a plan, but I like adding what I need to do and to choose priorities. That way, I can see the due dates – it's a lot more tidy than Blackboard at least."*

Several participants mentioned that they would spend a considerable amount of time creating a personal plan by hand, and that they would save time by using such a tool.

If the student can also trust the plan which is generated, they do not have to worry about finishing tasks before their due dates.

> *"If you trust it [the tool] and are confident that it works, it works well to save time spent on planning."*

> *"You don't have to worry so much about whether or not there is enough time before the due dates, as long as you can trust the plan. It may be nice to not have to think about it [due dates]."*

Participants also mentioned that using a tool like this would help them structure their work better and that it could help them become more disciplined.

One participant described that they see benefits in how the tool makes a clear distinction between free time and study time. They suggested that it would increase focus on the study sessions, as the tool would colour the periods in the plan grey if the student is not available to study.

Three participants mentioned that such a tool could help them decompose tasks, suggesting that large tasks that were decomposed would be easier to get started on.

Usage Areas

Regarding what types of tasks the participants would plan for using a tool like this, four of the participants explained that they would want to use it for all concrete tasks with clear due dates. One participant could even envision using the tool for planning *everything* in their lives and not just tasks related to their studies. Another participant mentioned that they could to use the tool for personal projects, and two participants mentioned that they would use the tool for practising for exams.

Potential Pitfalls

During the interview tests, 6 of the participants expressed possible pitfalls regarding future use of the tool or a similar tool. One of the mentioned pitfalls was that the personal plan might lose its purpose if the user misses a study session or is not able to work properly during the session. Regarding this situation, one participant suggested that it should be possible to mark a study session as *missed* or *unsuccessful*, so that the tool can adapt the rest of the personal plan according to the lost time.

Two participants also mentioned that such a tool can induce a false sense of security. If the user relies completely on such a tool, but has forgotten to add tasks or unavailable times, or diverges from the personal plan, they may accidentally miss deadlines.

Comments on the Generated Personal Plan

The participants had much to say about the personal plan which the tool had suggested. This feedback could be categorized under one of the following: "distribution of study sessions", "break preferences" or "coordination with other students".

Distribution of Study Sessions

Five of the participants expressed a need for *variation* among the tasks in their personal plan, as studying a single subject for an entire day would become monotonous or demotivating. One of the participants also suggested that the plan should suggest tasks from at least two courses per day.

Two participants expressed dissatisfaction about the personal plan being “front heavy”, where all study sessions were planned as early as possible. They suggested that the tool should spread the required study sessions across less packed days in their personal plan, as long as the tasks could still be finished before their due date.

Two participants wanted to be able to specify varying lengths for certain study days; this with regard to how they would not be able to work for the same duration every day, or wanted to finish early on Fridays before the weekend. One participant also wanted to specify a custom study session duration per task in order to account for the task difficulty.

Finally, two participants wished to specify which days or time spans they wanted to work with a particular task in order to account for the availability of their “group members”, with which they share the responsibility of completing the task. The availability of help with the tasks, for instance from a student assistant, was also mentioned.

Break Preferences

Six of the participants were not satisfied with all breaks being the same lengths, and they wanted to be able to choose a fixed time each day for a longer break. This longer break was intended to be used as a lunch break and a way to relax before continuing their studies.

One participant also mentioned that it was very common to start breaks at whole hours.

Coordination with Other Students

Two participants said that the tool should in some way account for group work. One of them suggested that this could be done by specifying available times for a task. Another participant mentioned that they wanted to be able to coordinate their plans with the schedules of available help.

Requirements for Accepting the Generated Plan

The participants mentioned several factors explaining whether or not they would be accepting of an auto-generated personal plan.

One of the factors is regarding how they would be able to *trust* such a tool. One of the participants mentioned how they would probably double-check their personal plan when using the tool for the first couple of times, although they would likely become more comfortable after having used the tool for a while. As a requirement for using the tool, another of the participants explained how they would need to trust the tool not to misuse their personal data which is needed to generate the plan.

A second factor concerned the *control* of the personal plan. Three participants explained how being able to edit their personal plan was a condition for using the tool. One participant expressed worry about the thought that if they adopted a large degree of automation in their planning, errors in the planning could occur without their knowledge. It was also mentioned that if the tool would require them to spend too much time editing their personal plan, they might as well perform the planning manually using pen and paper instead.

One final factor had to do with the student's *familiarity* with the tool and existing tools fulfilling similar needs. One participant explained how their familiarity with Google Calendar and Google Keep would make them reluctant of using something new, as they had already established strong habits and routines with these tools. Another participant mentioned how one of the tool's disadvantages was that it was "another app", and how this would raise the threshold for beginning to use it.

CHAPTER 8

Quantitative Evaluation

In similar fashion to chapter 7 on the qualitative evaluation process, this chapter will describe the *quantitative* evaluation process of the tool. First, the methodology will be described following the next few sections, and finally, the results will be presented.

The main purpose of the quantitative tests was to gain further insight into RQ4 regarding how students would perceive a tool such as this which supports time management. The sub-questions related to RQ4, as described in chapter 7, would also be points of attention during this evaluation process. Additionally, when considering RQ2, there was also a possibility of uncovering more needs of students which were not discovered during the qualitative evaluation of chapter 7 or from the survey on study preferences of chapter 4. Another motivation for performing a quantitative evaluation was to circumvent some of the limitations of the qualitative tests as described in chapter 7. Letting the students use the tool on their own accord for an extended period of time, in an environment of their choosing, would hopefully produce less biased results than the results of the qualitative evaluation.

8.1 Circumstances of the Tests

The testing process was advertised as an "open beta" where the participants would be asked to answer a survey at the end. After the "open beta" had been advertised, there were 40 students who had showed their interest in participating in testing as well as answering the subsequent survey. These participants were emailed an introduction to the testing process along with a link to a beta-version of the tool.

8.2 Test Format and Conduction

The participants then allowed to use the tool as frequently as they pleased, and they were informed that they had the option of giving feedback on the tool after a few days of use. At the end of the testing period, the participants were provided a hyperlink to a survey, which was provided to them both by email and as a prompt inside of the tool. In the light of RQ2 and RQ4, several assumptions, or hypotheses, about the tool were made, which were then structured and formulated into the survey-questions which have been summarized below. The full list of survey questions can be found in section C.2.

1. Do you normally use tools such as this, or have you previously used similar tools to plan your work/studies?
2. On a scale from 1 to 5, were 1 means "strongly disagree" and 5 means "strongly agree", what scores would you give the following statements about the tool:
    - I liked that the tool made a distinction between *study time* and *other time*.
    - I think it is important that the tool knows when I am available to study.
    - I think the settings for "preferred study time" worked well to show my available time.
    - I think that manually adding "unavailable periods" worked well for showing my available time.
    - I think the way in which the *calendar functionality* was integrated worked well to show my available time.
3. How "tightly integrated" would you prefer for such a calendar functionality to be in this type of tool?
4. (. . .) what scores would you give the following statements about the tool:
    - I feel there are a satisfying amount of ways to show when I am available to study (in the tool in its current state).
    - I used an unnecessary amount of time moving between the different views of the tool (for instance between the overview of tasks and the view of my personal plan).
    - I spent lots of time adjusting/making changes to my personal plan after it had been generated.
    - It was easy to make the changes which I wanted to my personal plan.
    - I feel there are a satisfying amount of ways to to change my personal plan after it was generated.
5. Are there any ways to make changes to your personal plan that you feel are missing at the moment?
6. (. . .) what scores would you give the following statements about the tool:
    - I liked how elements in my personal plan automatically adjusted themselves after I had manually made changes.
7. Would you like if more elements in the schedule were adjusted automatically?
8. (. . .) what scores would you give the following statements about the tool:
    - I feel that the plan which was generated fit me and my needs.
    - I felt I was missing an overview of how much time I had already spent on my tasks.
    - I wish the tool would plan for more varied workdays.
9. Do you have any thoughts on what it would mean to have a "varied" study plan?
10. Is there anything about the planning of study days which the tool did not account for, but that you wish it accounted for while generating schedules?
11. (. . .) what scores would you give the following statements about the tool:
    - I could imagine myself continuing to use this tool to structure my studies.
    - I could imagine myself using this tool to structure my tasks in general (not just study related tasks).
12. I could imagine myself continuing to use a similar tool, but only if . . .
13. Is there anything else you want to say about the tool?

8.3 Limitations

As with any statistical analysis, this process of evaluation is also susceptible to "data snooping", which is also referred to as "p-hacking" and sometimes "data dredging". Data snooping is often described as selectively choosing results and tests to favour a specific hypothesis or outcome. As described by White [74], data snooping is a dangerous practice which is to be avoided, even though it is endemic, and perhaps unavoidable, in most branches of science. This was kept in mind while analyzing the results, but it is important to recognize the possibility of the evaluation process, although subconsciously, adhering to this practice.

The quantitative testing period started late in the semester. Therefore, most students had already finished their assignments and mandatory coursework, and their remaining tasks for the semester consisted only of preparing for exams. As a consequence, the conditions for the evaluation were not ideal, as the tool was primarily aimed at helping students plan their busy schedules in the middle of the semester. This may also have hindered recruiting participants for the testing process, as students might not have had enough time to participate in the testing, which in turn might have led to the overall small sample size of 17 respondents. A small sample size limits the types of statistical tests which can be conducted as well as the overall strength of tests.

During the qualitative testing, follow-up questions could be used to clarify how exactly the respondent had interpreted a question, and thereby affirming the basis for the given response. With the current testing format, this affirmation was not possible, and there is a chance that the respondents of the survey could interpret the same question differently. This could potentially produce answers which are less fit for comparing against each other. With Likert type questions, there is also the chance that the middle "neutral" value is not actually interpreted as a neutral response by the respondents, and that the answers may be skewed towards either side of the scale for each question.

A few bugs which severely affected the usability of the tool were discovered by participants during the testing period. These bugs were fixed shortly after the testing began, but may have discouraged further use of the tool, leading to responses which were not necessarily based on "normal" usage of the tool.

8.4 Processing of the Results

The approach which was used to analyze the results is a form of null hypothesis testing; more descriptively, it is an approach which Perezgonzalez [46] would describe as "Fisher-leaning null hypothesis testing". As with the standard Fisher approach to data analysis, a level of significance (alpha level) was decided upon *a priori* and used for all statistical tests. The default level of significance used for every test was .05. Since it was difficult to control how large the sample size would be after the testing period, the test power (and beta level) of the tests was not decided upon prior to the conduction of the tests. However, the power of every test was eventually considered in order to use tests which, given the final sample size and distributions, would minimize the chance of wrongly rejecting the alternative hypotheses (type II errors). Also, following the conventions of the Fisher approach to data analysis, no hypotheses were assumed unless they resulted in a p-value lower than the level of significance which had been decided upon.

The results of the statistical tests described below were calculated using the R programming language. The affiliated code is provided in the appendix under section B.2.

Testing for Normality

Normality tests were conducted on the distribution of every Likert type variable. This was done in order to determine whether the data was applicable to use with parametric tests, which are known to generally have greater test power than their non-parametric counterparts. As a result of the overall small sample size of 17, neither the variance nor mean of the data may be assumed. Therefore, the Lilliefors test was used to test for normality instead of the more powerful parametric Shapiro-Wilks and Shapiro-Francia tests.

The “Shape” of the Distributions

For every Likert type variable, it would be beneficial to determine whether the distribution leans towards the lower or higher end of the scale. If certain statements about the tool are deemed to be particularly important, they could be used to garner further insight into RQ2. In order to make proper assumptions regarding either the mean or median of each variable’s distribution, appropriate tests need to be conducted. For normally distributed variables from the data, a confidence interval from the befitting t-distribution would be used to make assumptions about the population mean. For non-normal distributions, the Wilcoxon Signed-Rank test was used to make assumptions about the population median. However, as the Wilcoxon test also assumes symmetry of the distribution, the Sign test can be used in the case of asymmetry according to Pratt and Gibbons [54].

Examining Categorical Variables

For the few variables which are categorical rather than Likert type, a simple summation was made with the result showcased as a bar diagram. This process does not necessarily help with assuming or rejecting hypotheses; nonetheless, it can give insight on which categories are the most frequent among the general population of students.

Correlation Tests

The strength of correlations between every continuous variable, as well as the significance of the correlation, was tested – either by using the Pearson or Spearman method depending on whether or not the distributions were assumed to be normal. Of special interest is the correlation between “I could imagine using a similar tool in the future” and other variables, as the associated responses are heavily related to RQ4B: “Among students, what is the perceived usefulness of a tool for supporting task planning?” The same goes for the variable “I could imagine using the tool for general planning”, as the associated responses also relates to the perceived usefulness of the tool. The correlation between “Initial satisfaction with the generated plan” and other variables is also of a particular interest in order to garner insight on RQ4A: "Among students, what is required to accept an auto-generated task plan?"

Principal Component Analysis

In order to further examine what groups of variables might correlate and vary together, methods from multivariate statistics can be used. However, several of the methods commonly used in multivariate statistics, like MANOVA and MANCOVA, make several assumptions about the data to be analyzed. For instance, following rules like the "Law of large numbers", assumptions may be made when the sample size is considerably large. Conversely, with a small sample size, these kinds of assumptions are generally advised against.

A more general purpose method from multivariate statistics is Principal Component Analysis (PCA), which is performed by rotating the space spanned by the variables around a set of Principal Components (PCs). In and of itself, the process of PCA does not make any assumptions about the overall population, as it consists of simply applying rotations on the already existing data. The process of *interpreting* PCA results, however, relies on a few assumptions and should be used with caution. If the purpose of the PCA is to reduce the dimensionality of the data, or to deduce PCs which are supposed to describe something about the overall population, the analysis needs to be performed with an adequately large sample size; the available sample size of 17 is far from adequate for these purposes. If the sample size of this analysis was, for instance, 18 instead of 17, the resulting PCs would most likely have drastically different orientations. This observation illustrates how the PCs are highly arbitrary with a sample size this small, and that they therefore should not be the focus of this analysis.

In the end, the PCA would not be used to test for hypotheses or to assume any formal results, but rather as an alternative way of representing the 16-dimensional data as a 2-dimensional plot. In other words, the results of the PCA are to be used *exploratory* rather than confirmatory.

Analysis of Full-Text Responses

Besides examining the continuous and categorical variables, which can be done through a statistical approach, there is also the case of the more nuanced full-text responses. To analyze these responses, an approach similar to what is described in chapter 7 was used.

8.5 Results

This section will provide an overview of the most significant findings from the analytical process on the quantitative data. Further details about the results and the analytical process is showcased in section A.2 of the appendix. Moreover, as reported in section A.2 of the appendix, none of the continuous variables would be deemed as normally distributed. Therefore, only results from non-parametric tests on the data will be considered hereafter.

Likert Type Responses

An overview of the scores of every Likert-like question can be seen in Figure 8.1. The questions are sorted from "most agreed upon" at the top of the plot to "least agreed upon" at the bottom. Some responses, particularly the "need for the tool to know when I am available to study", seems to be agreed upon across the board. Additionally, some of the responses show a clear divide between the ones who agree and those who disagree, for instance those who "need more variation in their plan". However, in order to be able to assume any of these results, further statistical analysis needs to be done.

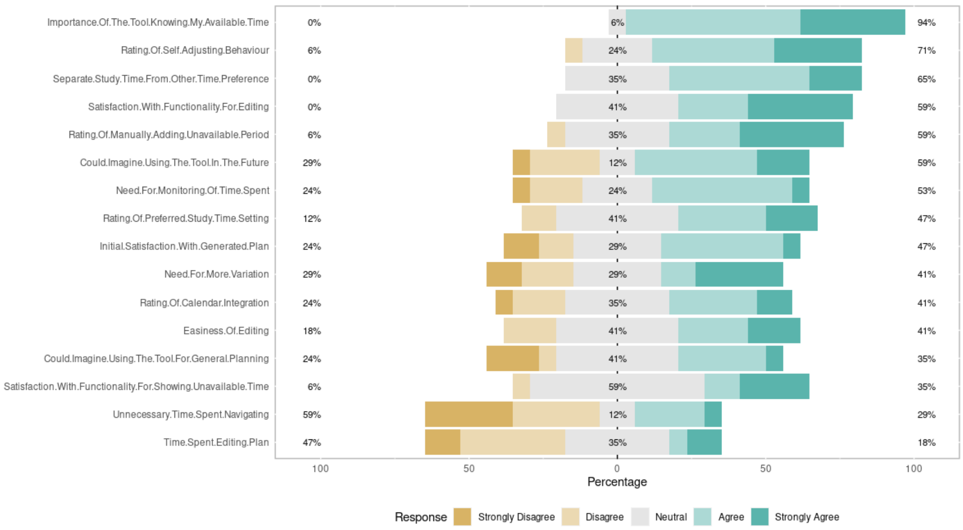

Figure 8.1: Likert Plot of the Likert Type Variables from the Quantitative Testing

Sign and Wilcoxon Signed-Rank Tests

As a result from the Sign tests, the following Likert scores were found to have a median significantly *greater* than the neutral value (3):

- *Importance Of The Tool Knowing My Available Time*

$$p = .0002$$

- *Rating Of Self Adjusting Behaviour*

$$p = .0017$$

- *Separate Study Time From Other Time Preference*

$$p = .0005$$

- *Satisfaction With Functionality For Editing*

$$p = .0010$$

- *Rating Of Manually Adding Unavailable Period*

$$p = .0059$$

Similarly, the following Likert score was found to have a median significantly *lower* than the neutral value (3):

- *Unnecessary Time Spent Navigating the Tool*

$$p = .0002$$

A few variables were deemed to have an even higher median by the Wilcoxon Signed-Rank tests. The most notable discrepancy was "Importance Of The Tool Knowing My Available Time", where the Wilcoxon test suggested a median of 5! However, judging by the resulting histogram of the variable, its distribution is most likely not symmetric. In general, none of the variables which produced discrepancies between Sign test and Wilcoxon Signed-Rank test showed signs of symmetry (as can be seen in section A.2 of the appendix). For these variables, the assumption of symmetry under the Wilcoxon Signed-rank test does not hold, and its result should be disregarded. The result from the Sign test, however, still holds, as it does not assume symmetry of the distributions. The entire result of the Wilcoxon and Sign tests can be viewed in Table A.4 in the appendix, including the highest significant median for each variable, as well as the discrepancies between the two tests.

Correlation

After calculating the correlations between every continuous variable using Spearman, and thereby producing a 16×16 correlation matrix, a minority of the correlations were shown to be significant. The significant correlations, as well as their degree of correlation, can be seen in Figure 8.2.

As previously stated, regarding RQ4, the most interesting variables to test for correlation are:

- "I could imagine using a similar tool in the future"
- "I could imagine using the tool for general planning"
- "Initial satisfaction with the generated plan"

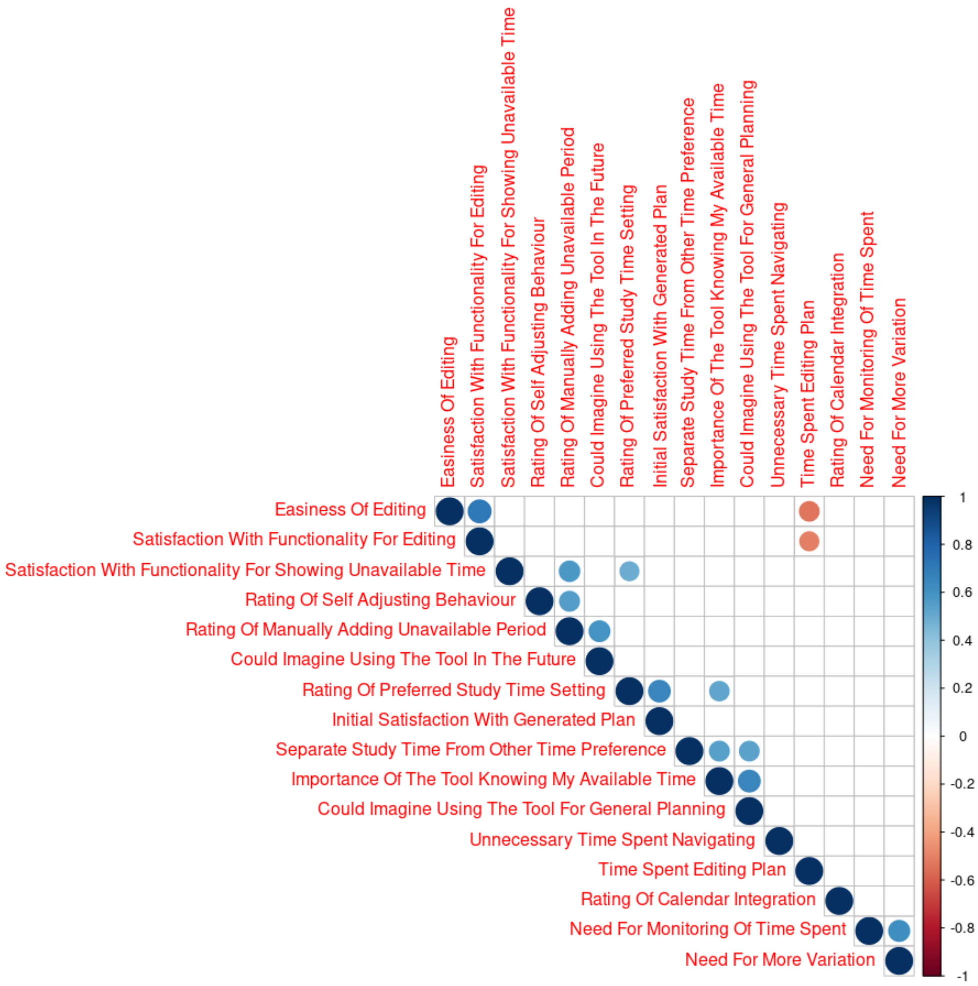

Figure 8.2: Correlation Matrix of Likert Type Variables (with Non-Significant Correlations Removed)

After calculating the correlation values for these variables individually, a few hypotheses were assumed:
"**I could imagine using a similar tool in the future**" strongly positively correlates with "**Rating of manually adding unavailable periods**".

$$r(17) = .60, \ p = .0111$$

"**I could imagine using the tool for general planning**" strongly positively correlates with "**Preference for separating study time from general time**".

$$r(17) = .54, \ p = .0260$$

"**I could imagine using the tool for general planning**" strongly positively correlates with "**Importance of the tool knowing my available time**".

$$r(17) = .66, \ p = .0041$$

"**Initial satisfaction with the generated plan**" strongly positively correlates with "**Rating of settings for preferred study time**".

$$r(17) = .65, \ p = .0047$$

Categorical Data

Below is a series of bar diagrams made by summing how many of the respondents belonged to each categorical variable.

Preferred Method of Calendar Integration

Three different preferences for calendar integration were eventually reported by the respondents. Among the sample population, "Automatically keeping the calendar and plan synced" (with 58.8%) is seemingly the most preferred way of integrating the calendar into the tool. The frequency of every preference regarding method of calendar integration is shown in Figure 8.3.

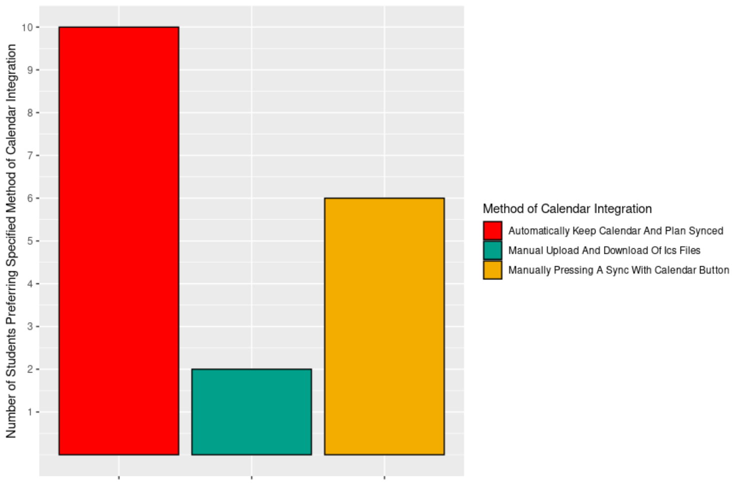

Figure 8.3: Preference for Method of Calendar Integration

Preference of Self Adjusting Behaviour

There were three different choices for self adjusting behaviour for which the test subjects could report their preference. Among the sample population, "I would like for my plan to adjust itself after editing, but only if I understood how the plan adjusted itself" (with 70.6%) was the most frequent preference which was reported. The frequency of every preference regarding automatic adjustment of the plan is shown in Figure 8.4.

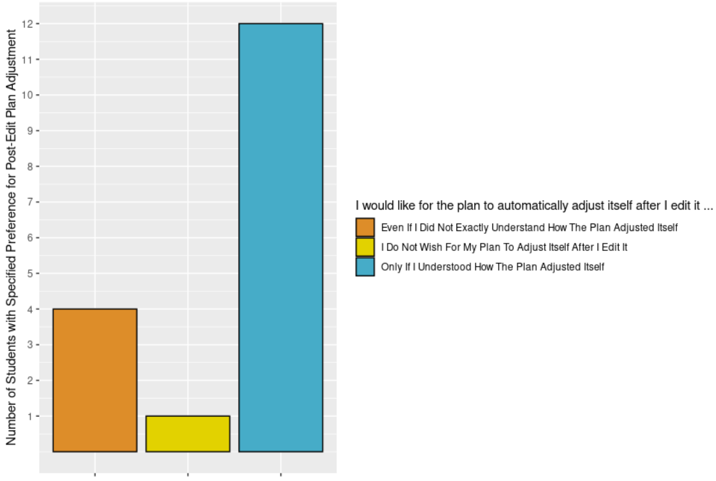

Figure 8.4: Preference for Automatic Plan Adjustment Post-Editing

Requirements for Wanting to Use a Similar Tool

In the end, there were four different types of requirements for continuing to use the tool which were reported by the test subjects. Among the sample population, "a tool better suited to personal needs" (with 64.7%) is the most frequent requirement for wanting to use such a tool. The frequency of every requirement for wanting to use a similar tool is shown in Figure 8.5.

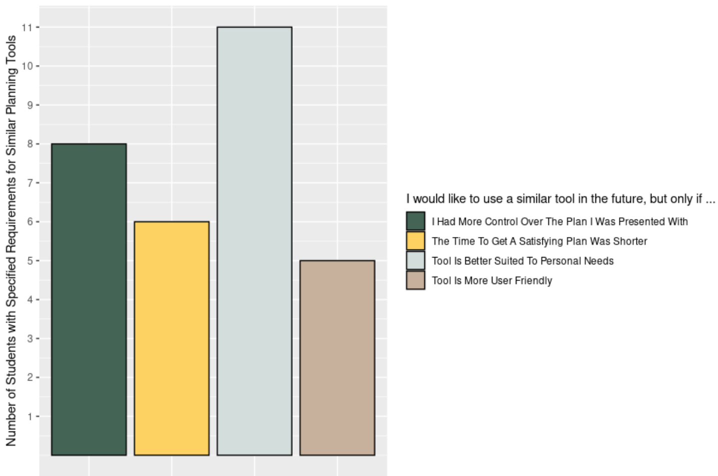

Figure 8.5: Requirements for Wanting to Use a Similar Tool

Principal Component Analysis

A set of 16 PCs was derived by performing the PCA, among which the two PCs with the highest explanation rate were used in the dimensionality reduction of the data. The cumulative explanation of variance from the first two PCs is summarized in Table 8.1, and the full report of the eigenvalues and eigenvectors can be found in section A.2 of the appendix. For all of the results presented in this section, it is also important to consider the limitations of the PCA which are imposed by the small sample size. In short, the PCs which are presented here cannot be described as representative for the overall population, however, the plots produced by the dimensionality reduction may still be useful by providing alternative perspectives on the data.

The First Two Eigenvalues from the PCA

Principal Component	Eigenvalue	Explanation of Variance	Cumulative Explanation of Variance
PC1	3.7	22.8%	22.8%
PC2	2.9	18.0%	40.9%

Table 8.1: The First Two Eigenvalues from the Principal Component Analysis

When projecting the base vectors of the 16 original variables onto the subspace spanned by PC1 and PC2, as shown in Figure 8.6, it is possible to observe how the different variables contribute to the overall variance in this subspace. The longer the base vector stretches out from the centre of the projection, the more of the variation it contributes to, which is important to keep in mind when viewing Figure 8.7 and Figure 8.8 presented later in this section.

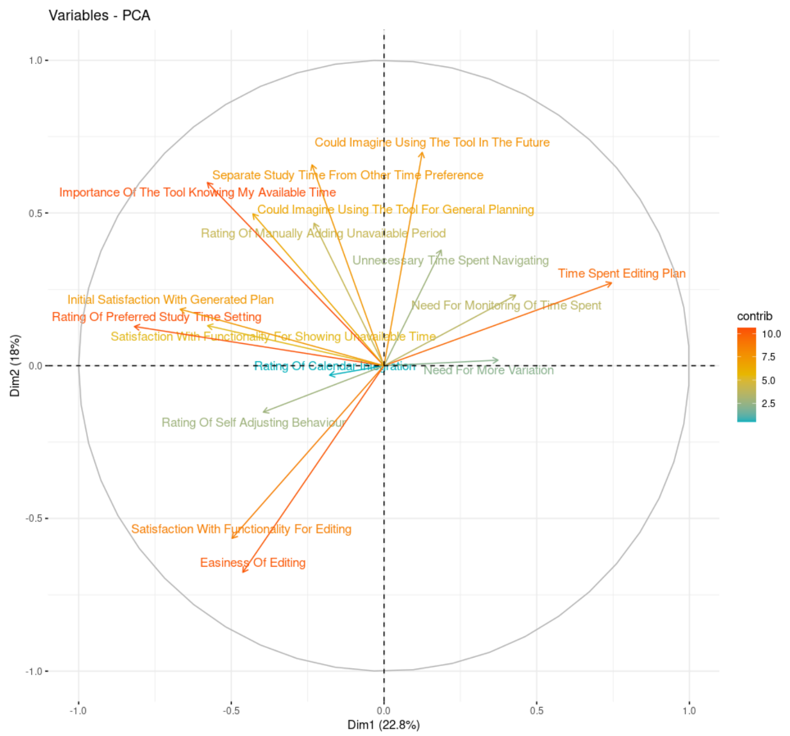

Figure 8.6: Contributions of each Continuous Variable Projected onto the Span of PC1 and PC2

In order to further investigate RQ4B a few derived data categories were used in the examination of the dataset. Combined with the plot of the projection onto PC1 and PC2, these categories can potentially give an idea of which variables correlate with the different categories of respondents. The respondents were considered as part of these categories if they had answered 4 or higher on certain questions. The derived categories are as follows:

1. Respondents who "could imagine to use the tool in the future" and those who could not
2. Respondents who "could imagine to use the tool in the future" *in addition to* "use it for general planning" and those who could not

Figure 8.7: Projected onto the span of PC1 and PC2: Students Who Could (and Could Not) Imagine Using a Similar Tool in the Future

When viewing these categories together with the projection of the data, a clear pattern emerges. As can be seen in Figure 8.7, even though the data has been subject to strong dimensionality reduction, there is a nearly clear-cut divide between the respondents who could imagine using a tool like this in the future and the respondents who could not. Excluding the variable "Could imagine using the tool in the future", which obviously contributes to this divide, a combination of the following variables seem to contribute to the want of using a similar tool in the future:

- "Preference for separating study time from other time"
- "Importance of the tool knowing my available time"
- "Satisfaction with functionality for showing unavailable time"
- "Rating of preferred study time setting"
- "Rating of adding unavailable period"
- "Could imagine the tool for general planning"
- "Initial satisfaction with the generated plan"
- "Unnecessary time spent navigating the tool"

Likewise, lower ratings of these variables from the respondent can indicate that they would *not* want to use a similar tool in the future. Despite how "nice" the predictive capabilities of these results may seem, it is worth noting that the correlation between these variables does not tell much, if anything, about the underlying causality. Also, if another sample was added, it is no way to be sure about where in the plot it would have ended up this has the potential of drastically altering how the result might be interpreted.

When displaying the same projection, but with the category of respondents "could imagine to use the tool in the future *in addition to* use it for general planning", a similar observation is made. When viewing Figure 8.8, there seems to be a divide between the two categories, although not as clear as the divide between the previous categories showcased in Figure 8.7.

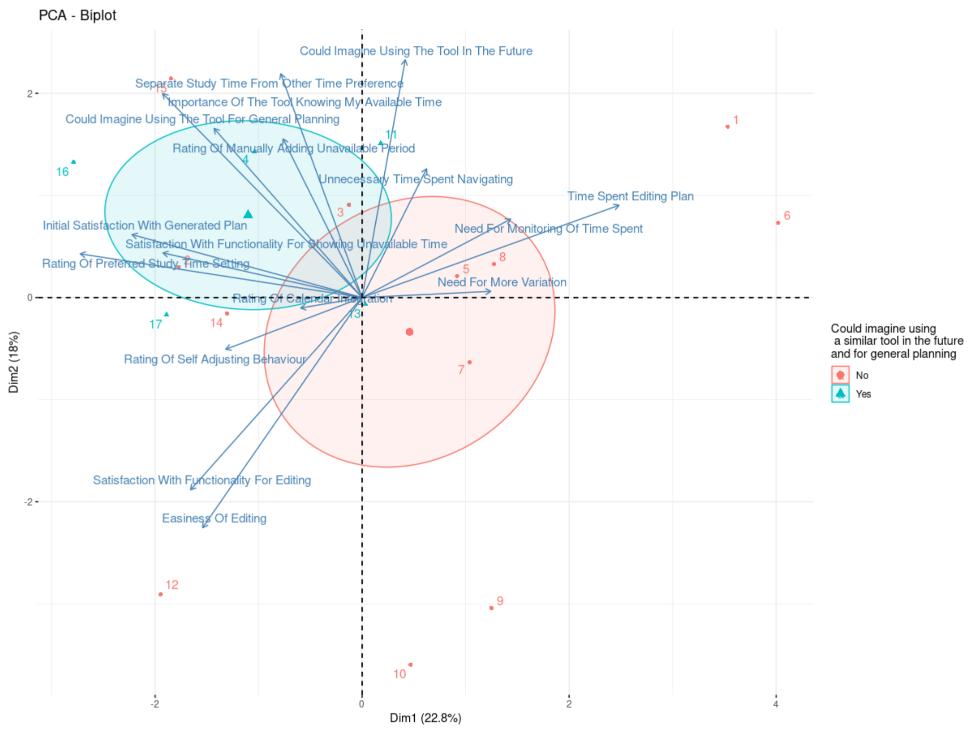

Figure 8.8: Projected onto the span of PC1 and PC2: Students Who Could (and Could Not) Imagine Using a Similar Tool in the Future *and* Use it for General Planning

Summary of Free-Text Responses

When considering the responses to the free-text questions, it was observed that they could be categorized under three distinct themes:

- **General requirements needed to be fulfilled by this type of tool**
- **Requirements regarding the usability of this type of tool**
- **Requirements specifically regarding the generated plan**, which can be further sub-categorized as:
  - Methods required to edit the personal plan
  - “variation” required of the personal plan

General Requirements Needed to be Fulfilled by this Type of Tool

Two respondents explicitly stated that they would like for the tool *not* to require the names or code of courses when adding new tasks; this in order to let them use the tool for general planning, not necessarily related to their studies.

> *“I don’t think it felt very natural to add other tasks like ‘buying groceries’ etc. These are tasks which I would very much like to have gathered in a single tool.”*

Another respondent expressed how they missed some functionality for monitoring tasks and time, especially in the form of notifications for upcoming tasks.

Requirements Regarding the Usability of this Type of Tool

Several respondents stated that adding tasks to the tool should be quick. This is especially important for students who might have several smaller task appear throughout the span of a single day. It was also mentioned that if the process of adding tasks is not quick enough, they might as well perform the planning manually.

> *“I could imagine using this tool if it hadn’t been for the large amount of time which is currently needed to input tasks into the tool. It is faster for me at the moment to set up my study plan by hand.”*

Another respondent was frustrated about how the tool did not set aside the “correct” amount of time for each task. They did not feel that the duration of work sessions displayed in their personal plan properly correlated with the “estimated completion time” which they had entered when inputting the task.

One of the respondents stressed the importance of being able to specify the amount of “buffer” between the current point in time and the next task of the generated plan. They did not particularly like how their personal plan would be generated with an empty period of one hour starting from the current point in time. This empty period was initially placed in every generated plan to give the students a buffer between their planning process and when they would be inclined to start studying. The respondent also agreed with this sentiment, although personally, they would rather have no buffer at all.

There was also a mention of the need for editing tasks whose deadline had already been surpassed. This would not only require the ability of monitoring tasks with an upcoming deadline, but also a log of previous tasks. This feature was already implemented in the tool when the testing began; nonetheless, its importance was stressed by a couple of the respondents.

A few respondents explained their need to view their entire personal plan without the need of zooming or scrolling. At the moment of testing, in most web browsers, their personal plan would only be partially visible along the vertical axis. Some of the respondents further elaborated on this need by explaining how they want to quickly assess if a generated plan satisfies their needs, or if they would need to edit the plan or generate a new one.

Lastly, one respondent found it especially difficult to add certain tasks to the tool, as they were either too “large” or “vaguely defined”. The student would therefore skip adding these tasks into the tool and rather focus on smaller tasks which they were able to perform.

Methods Required to Edit the Personal Plan

Regarding the **methods of editing the personal plan**, several suggestions were made by the respondents. Most frequently brought up was that the tool should allow placing a study session *after* the deadline of the corresponding task. The respondents argued that not all tasks have clear-cut deadlines, for instance tasks which only need to be completed “sometime in the future”, and that it would be beneficial if the tool could handle these tasks more loosely.

Despite the existing “quick add” functionality for adding study sessions and breaks manually, a few of the respondents expressed their need for an even faster way of manually adding study sessions to their personal plan. This need especially concerns study sessions belonging to tasks which have not yet been added to the tool. In this situation, in order to add their desired study session, the student would first need to navigate away from their plan-view and back to the overview of tasks, then input a new task in the task-view, and finally navigate back to the plan-view. As a means to further speed up the process of adding tasks and study sessions, one of the respondents also described their need for specifying keyboard shortcuts.

Another respondent suggested a colouring option for tasks in the personal plan. In the version of the tool which the respondents had tested, the colouring of each study session was determined by hashing the name and course code of the affiliated task, which would return a usable colour described in hexadecimal. As a consequence, two or more neighbouring study sessions in the plan would sometimes have similar colours, even though they belonged to different tasks. Thus, the respondent found it hard to distinguish between some of the different elements of their personal plan, which in turn made it difficult to quickly asses whether or not the generated plan was satisfying.

Type of "Variation" Required of the Personal Plan

When presented with the prompt of **"what it would mean to have a 'varied' study plan"**, the respondents had befittingly varied responses. Four of the respondents explained how a varied study plan would let the student work with more than a certain amount of distinct *courses* per day. Another four respondents answered, similarly, that a varied study plan would let the student work with more than a specified amount of distinct *tasks* per day. In other words, among the sample population, the number of distinct courses/tasks worked with per day is what determines whether or not a plan is satisfyingly "varied" for the majority of students.

An alternative way of describing a satisfyingly "varied" study plan came from one of the remaining respondents. In their response, they described how a study plan would have enough "variation" only if the time spent on a single task concurrently does not exceed a certain limit. Consequently, after having worked with a single task for a certain amount of time, the plan should tell the student to preempt the current task and start working on something else for a period of time. In their response, however, it was not specified how long this period of time should be before the student can resume a previous task after preemption.

Another respondent suggested that a "varied" study plan would entail variation in the "type" of tasks performed by the student. Examples of the types of tasks include reading, writing and going to lectures.

Lastly, one respondent explained how a variation in the length of study breaks can benefit the variation of a study plan.

Time Planning Tools Used by the Students

Alongside the other free-text questions, there was also a question regarding what kinds of tools and combinations thereof which the respondents already have used for task planning. Among the 17 respondents, the following combinations of tools were mentioned as having been used for task planning:

- Google Calendar (mentioned by **5** respondents)
- Google Calendar in combination with Google Docs
- Google Calendar in combination with Notion
- Their phone's built in calendar and notification features (mentioned by **2** respondents)
- Akiflow
- Todoist
- Milanote
- The "1024 time table generator"
- Microsoft OneNote

CHAPTER 9

Discussion

After the processes of qualitative and quantitative evaluation, as well as results garnered from the survey of chapter 4, the a significantly large "heap" of results were produced. This discussion chapter aims at digesting the results which have been presented thus far, as well as relating them to the research questions which were posed in chapter 1.

9.1 RQ1 – What is the State-of-the-Art of Tools for Supporting Task and Time Management?

There are a few things to say about the state-of-the-art of tools for task and time management. Firstly, as was briefly mentioned earlier in the thesis, seemingly every tool which supports some sort of time management also supports task management in some way or another. In chapter 5, it was discussed how task management may be seen as a necessity for the tool to facilitate time management.

Moreover, tools which facilitate for more than one type of time management behaviour (among the behaviours defined by Claessens, Eerde, Rutte, and Roe [11]), seem to also provide a *synergetic effect* between these behaviours. Better time monitoring may, for instance, help in future time planning, as monitoring the time which has been spent on current tasks may inform future time planning. Time planning may, in turn, support both time assessment and time monitoring. Most of the tools which were examined, with few exceptions, included support for several of the defined time management behaviours.

An observation was also made of how tools with more specific use cases would be able to make several assumptions, which in turn provide possibilities for easing the process of several time management behaviours. Most notably was the trend of the tools' "generality" and the amount of automation they exhibited in their functionality. The newly introduced tool Shovel is an instance of this trend, where the tool makes several assumptions on how students would like to perform the *timeblocking* of their study related tasks, while the tool also makes this behaviour efficient and "streamlined" to perform.

Newer tools like Shovel, Akiflow and Reclaim also exemplify how the realm of commercial tools for time management is rapidly changing. Just a few years ago, it would be hard to find tools which support the behaviours of time assessment and planning in such an advanced manner; perhaps the most advanced, commonly used tool at the time would have been Google Calendar. Nevertheless, in the domain of tools to support time management, the general rule seems to be that commercial tools set examples, and the theory follows thereafter. In turn, there seems to be much unexplored potential for future research on these types of tools. Of special interest might be how the simpler and more straight-forward steps of certain time management behaviours may be facilitated through automation.

9.2 RQ2 – What Needs and Preferences of Students Should Be Considered by Tools for Supporting Task and Time Management?

Judging by several of the results of the quantitative evaluation, including the many sign tests which were performed, it may be assumed that a tool which supports time planning among students should also know when the student is available to study. A perhaps more interesting question is provided by asking "how?" How should a tool such as the one which was tested be informed of the student's available time? If one were to, again, observe the results of the quantitative evaluation, it may be assumed that the way in which "preferred study time" was specified is a good answer to the aforementioned question. Another functionality of the tool which was rated favourably by the students was the ability to be more specific about available time through adding periods of unavailability directly to the plan. The final method of specifying unavailability, as provided by the tool, was the calendar integration. This method, however, was not rated as highly as the two other methods. Although, it is also worth considering that the large majority of the students would want a completely different way of integrating their calendar than the current method provided by the tool. It may be assumed that the calendar integration in fact *is* a good idea, although it would need to be substantially more seamless than the current use of ICS-files. In the end, a tool such as this should most likely offer a couple of ways of connecting to a student's calendar. However, regardless of *how* the connection is done, one may assume that the calendar should connect to the tool in a more autonomous manner.

Another important dilemma, which was made clear from both the the survey of chapter 4 and the results of chapter 7, is the fact that certain students would like to schedule for more than *just* study related tasks. A tool which is to support time planning and time assessment among students should also consider the possibility that some "non-study related tasks" also need to be performed in the span of a study day. These non-study related tasks may include training or more "chore-like" tasks. The tool which was evaluated *did* provide the possibility of adding tasks without also connecting them to any existing courses, thereby enabling the student to add more general types of tasks. However, the fact that adding tasks through course codes was used by default apparently stunted a few students who wanted to add more general types of tasks. At the moment, there are seemingly few reasons why also "non-study related tasks" should not be supported by tools like this. One may argue that there are possible benefits of only handling study related tasks, as these tasks follow the same form of "template", and useful assumptions may then be made by the tools design. It should, however, be possible to make assumptions about the study related tasks while *also* allowing for more general tasks to be added to the tool.

Considering the fact that not every study related task, or "non-study related tasks" for that matter, is performed at the same *location* also provides an interesting requirement for scheduling. Reportedly, several students take the location into account when planning their days of studying. There is also the case of more peculiar needs as, for instance, the *availability* of collaborators for group oriented tasks. These needs do not necessarily need to be supported by the tool in the form of automation. However, the tool should enable the student to plan around these types of needs in some other way.

During the qualitative evaluation of chapter 7, the notion of *lunch breaks* and other daily habits were also brought up. These habits are something which a tool should let the student accommodate for. The idea of lunch breaks, for instance, is something which may easily be added to the instructions used by the tool for prearranging breaks.

Lastly, there is an important difference between tasks which students are *commissioned* with, for instance through their virtual learning environment, and tasks which the students have imposed on themselves; the latter of which will usually have a much less strict deadline than the former. One may imagine the case of a student setting a goal of reading up on a certain topic before next week. It is not be the end of the world if this task is completed on Monday instead of the previous Sunday. On the other hand, the thought of exceeding a deadline for an obligatory assignment might seem like the end of the world for some students. A tool for supporting time planning should therefore be able to differentiate between these two types of deadlines and how they should be prioritized.

9.3 RQ3 – What Kind of Scheduling Algorithms May be Applied in the Domain of Study Related Tasks?

As discussed in the chapters on scheduling and design, exactly how the problem of scheduling study related tasks should be solved depends entirely on how it should be modelled. The problem is characterized by an inevitable amount of uncertainty, and there is much leeway for making different assumptions about the problem. Some attributes of the scheduling problem, however, are much more certain than others, and some fundamental model of the problem may be assumed.

The Fundamental Scheduling Problem

Scheduling for minimizing the number of exceeded deadlines, seems to be highly prioritized among most students judging by the results of the survey of chapter 4, as well as the satisfaction with the default EDD sorting of tasks used in the tool. However, as was just discussed in the previous section, not all deadlines are created equal. The student may want to add extra weight to some deadlines, while other deadlines are not as "strict" as the rest. The basic problem of scheduling study related tasks is then, perhaps, best modelled trough the following multi-objective function:

$$\sum w_i U_i^{(1)}, \ \sum w_i T_i^{(2)}$$

Given the way of classifying scheduling problems which was described in chapter 3, this notation may be translated to: Minimize the sum of weighted, tardy jobs with hard deadlines; under this criteria, minimize the sum of weighted tardiness.

It may also be assumed that students are not always available to perform their tasks, although they are very much capable of preempting these tasks and resuming them later on. The fundamental model of the scheduling problem may thus be expressed as:

$$1 \mid r\text{-}a,\, \overline{d}_i \mid \sum w_i U_i^{(1)},\ \sum w_i T_i^{(2)}$$

This does not seem much different from the standard problem of 1|r-a| $\sum w_i U_i$, which is solved by Moore-Hodgson's algorithm. By finding a solution which minimizes the amount of tardy jobs, one would most likely not be far away from a solution which also minimizes the amount of tardiness. If this may be assumed, a meta-heuristic algorithm will have a high chance of finding feasible solutions to the overall problem.

Extending the Scheduling Problem

Now that the basic needs for scheduling are covered by the model of the problem, it may be further extended to include more peculiar needs. For instance, there was a somewhat common need among students for taking the *location* of tasks into account. By connecting each distinct location to a corresponding *job family*, one may model the time it takes to travel between two locations as a *family dependent setup time*. Promising solutions have previously been found for these types of problems [57], so it should be possible to account for the locations where the tasks need to be performed, as well as the additional time which is required for travelling between them.

Another important extension to the model is the accommodation of "variation" in the study plan. This need is made apparent from both the survey results of chapter 4 as well as the results from the quantitative evaluation. In chapter 5 the objective function for enabling the wanted amount of variance was denoted by $\mathcal{V}$. However, due to a lack of information about what students would regard as variation, this objective was not being discussed any further. Conversely, it is now possible to continue this discussion in the light of the results from the quantitative evaluation. The student's provided much useful insight on *what* this variation would actually entail. The most common way of defining this variation was as "the number of distinct courses/tasks worked with per day". The student would ultimately need to choose between the option of either "courses" or "tasks" in order to define how the variation should be measured. This choice may simply be implemented in the tool as a specific study preference. However, there is something lacking from the above definition.

One may imagine a student who wants to work with 2 courses per day in order to have enough variation in their work. Their day of studies begins at 8 o'clock, and they finish their first task of the day by 9. The next available task is a considerably larger one, and will last throughout the rest of the day. Even though the variation was specified according to the above definition, the student still ended up with a considerably large amount of time spent doing the same task. Therefore, a slightly different definition of variation needs to be derived. Based on one of the other answers from the quantitative evaluation: "A study plan would have enough 'variation' only **if the time spent on a single task concurrently does not exceed a certain limit**. Consequently, after having worked with the same task for a certain amount of time, the plan should tell the student to preempt the task and start working on something else for a period of time."

Given the definition which was just provided, it seems possible to concretely describe the objective function for variation $\mathcal{V}$. For the following description, it is assumed that the student wants to vary the time spent on *tasks*; the same description, however, also accounts for the case of variation in *courses* if the wording is slightly changed. Let $\mathcal{G}_i$ be a group of study sessions dedicated to the same task where each study session is within at most one break's length of another. In other words, $\mathcal{G}_i$ denotes a group of neighbouring tasks, only separated by breaks. Let k be the number of such groups which exist in the schedule, and let $t(G_i)$ be a function which returns the total study time contained by one of these groups. Let v be the amount of time the student may work without switching tasks before they consider their work as "not varied". The objective function for achieving enough variation may then be defined as:

$$\mathcal{V} \coloneqq \sum_{i=1}^{k} \max\{0,\, t(G_i) - v\}$$

This way of modelling variation may seem overly convoluted. It is, however, not as convoluted for a computer which would attempt to calculate the objective function $\mathcal{V}$ as any other objective function. If the above function may be evaluated programmatically in a way which does not require too much time to calculate, an evolutionary algorithm, for instance, may be able to find good solutions using this objective as a fitness function. One concern of this objective function, however, is that a possible solution to the corresponding scheduling problem would simply be to switch tasks as often as possible. Among the population of solutions which may be derived from a meta-heuristic algorithm, one would therefore need to choose a solution with generally few task switches.

Spreading the Required Workload across the Plan

As was stated by some of the students during the qualitative evaluation of chapter 7, there is a need to spread the required workload of tasks across the available time of the plan instead of performing every task as soon as possible. A simple solution is provided for this problem: Suppose that there is enough time to perform all tasks (otherwise, the student would not have the need to spread them across their plan). In this case, it is known that choosing tasks greedily by the EDD rule will provide an optimal schedule. In the proposed solution, every task is still chosen greedily according to the EDD rule, however, instead of scheduling a task as soon as possible, one may choose to spread the task across every study day between the current point in time and the task's deadline. The amount of time dedicated to a given study day may be weighted by the amount of study time which is available for that day. Since the EDD rule is followed, and the scheduling is performed under the condition that there is enough time for every task, an optimal solution is guaranteed. This scheduling method may be described as a modified EDD which spreads the required workload rather than heaping the work at the beginning of the plan. This option may be provided to the student using the tool, given that there, in fact, is enough time for performing every task.

9.4 RQ4 – What are students' perception of a tool for supporting task planning?

A – Among students, what is required to accept an auto-generated task plan?

As some students mentioned during the qualitative and quantitative evaluations, the process of obtaining a usable plan should be fast – at least faster than it would take the student to produce the plan manually. Additionally, several students mentioned the importance of them being able to quickly *assess* if they would use the plan which the tool has suggested, or if they would need to make modifications. For instance, the entire plan needs to be viewable after it has been generated by the tool. The ability to clearly distinguish study sessions, as well as other plan elements, by colour was also brought up by the students. The design of such a tool may, therefore, consider methods for viewing and navigating the plan quickly, in addition to colouring options for elements.

During the quantitative evaluation, the students' satisfaction with the generated plan was assumed to strongly correlate with their rating of the functionality for defining their "preferred study time". It may be further assumed that, in order for the student to want to use their given plan, their preferred study time needs to be accurately reflected by the tool.

Another element of the plan which needs to be accurately represented is the amount of time the student wants to dedicate to a given task. For instance, during the quantitative evaluation one of the students reported that they did not feel that their given plan had set aside the amount of time which they had specified for certain tasks, and that the resulting uncertainty was detrimental to their planning process. It is therefore important that the student is made aware of the estimated completion time of tasks which is actually used during the scheduling of their plan. If the tool were to introduce a more "sophisticated" form of completion time estimation, it is assumed that the student would need to be made aware of the estimated completion time which has been derived by the tool. In order to achieve this, the tool may, for instance, present the student with a *suggested estimate* for the completion time of a task while it is entered by the student. The student may then choose to either accept or reject this suggestion. Regardless, the student will be aware of which estimation of completion time is being used in the end.

B – Among students, what is the perceived usefulness of a tool for supporting task planning?

The current layout and design of the tool, as described by chapters 5 and 6, may be assumed to satisfy the student's needs for usability. Among the requirements for continuing to use a similar tool in the future, which was queried during the quantitative evaluation, the minority of students expressed that the tool would need to be more usable. The results from the sign test also suggest that the current way of navigating the tool is satisfying enough. Nonetheless, there is still room for improvement, as was suggested by a few students during the qualitative and quantitative evaluations. The process of adding tasks, for instance, may be designed to be both easier and quicker.

As a suggestion for how to improve upon this aspect, the view of the task list and the view of plan may be merged into one. This type of design is also found in other state-of-the-art tools like Shovel and Akiflow. Tasks may then quickly be added while viewing the plan, while custom study sessions may be added to the plan by dragging tasks directly into the plan from the task list.

The overall functionality for editing the plan seems to be well received by the students. Among the more experimental functionalities which were introduced by the design were the method of dragging and dropping plan elements around, the notion of "locked elements" and the self-adjusting behaviour displayed by the plan after having been modified. These were all highly rated among the students. Moreover, the addition of an on-boarding section proved to be useful for providing the student's with a clearer mental model of the tool.

Several students, however, still felt that the tool was not well enough suited for their own needs. The tool mostly focused on supporting the behaviours of time assessment and time planning, while time monitoring functionality was kept to a minimum. One may, however, suspect that the addition of support for time monitoring would be well received by students. The "need for monitoring of time spent on tasks" was not rated high in the quantitative evaluation; it was, however, not rated low either. Moreover, a few students suggested that the tool should provide notifications for upcoming tasks, which would be considered a time monitoring behaviour. Proper support for time monitoring would potentially also let the students provide better estimations for the completion times of tasks.

CHAPTER 10

Conclusion and Future Work

This thesis has examined the possibilities of supporting the time management of students through the use of digital tools. A design for such a tool, focusing on the behaviours of time assessment and time planning, was discussed. This tool was later implemented in order to further examine certain ideas and concepts of the design, of which include the tool's ability to automatically suggest personal study plans for the student. Throughout the work on this tool, as well as the subsequent evaluation process, several observations have been made.

A tool which is to support *time management* should, first and foremost, build upon basic functionality for supporting *task management*. To further support the behaviours of time assessment and planning of students, the tool needs to be informed on the student's availability. The tool which is proposed in this thesis suggests several methods by which this "available study time" may be derived. These methods include the student's ability to specify their preferred study time, importing events from their personal calendar, as well as manually adding specific periods of unavailability to their personal plan. All of these methods were well received by students during evaluation of the tool.

Students also exhibit several planning needs which need to be accounted for, of which include: the ability to prioritize both *hard* and *soft* deadlines, the accommodation of variation in the student's plan, and specifying lunch breaks as well as other daily habits. A tool which aims at supporting the planning behaviours of the student will therefore need to consider these ways in some way or another. In the case of an automatically generated plan, which the tool suggested in the thesis provides, many of these needs may be accounted for. However, some needs, such as the *location* of where tasks will be performed, may be considered too peculiar to be taken into account by a scheduling algorithm. Therefore, in order to cover the possible planning needs which a student may exhibit, the tool must provide proper methods to enable the student to edit their personal plan. So-called drag'n'drop functionality may be considered an essential method for enabling editing. Outside of the basic editing methods, the proposed tool suggests a few advanced methods for editing the personal plan, including *locked plan elements* and a self-adjusting behaviour after the plan has been edited. These methods were well received by the students who evaluated the tool.

A tool for supporting time management of study related tasks may also take into account the more general tasks of the student. It should be possible to schedule the more general tasks while also benefiting from the specificity of study related tasks.

A tool for improving time management among students may also support time planning by automatically suggesting schedules for the student. These schedules will be the result of some sort of scheduling algorithm, and therefore, the problem of scheduling study related tasks needs to be properly modelled as a scheduling problem. Following the common notation for describing scheduling problems, the following model of scheduling study related tasks is proposed:

$$1 \mid r\text{-}a,\, \bar{d}_i \mid \sum w_i U_i^{(1)},\ \sum w_i T_i^{(2)},\, \mathcal{V}^{(3)}$$

The problem is inherently *multi-objective*, as the student will prioritize the minimization of several variables including: the amount of late jobs with hard deadlines, the overall amount of tardiness in their plan, and the amount of work performed without enough variation (described by the objective $\mathcal{V}$).

By finding feasible solutions to the underlying scheduling problem, a tool may propose adequate study plans to a student. In order for students to accept such an automatically generated study plan, however, the plan needs to properly reflect the estimated completion times of tasks which the student has specified. Furthermore, the way in which the tool will provide the student with their final plan needs to be faster than if the student were to perform the planning manually. Finally, the plan needs to be presented in a way which enables the student to quickly assess whether or not they may want to actually use it, or if it requires further editing.

10.1 Future Work

The resulting tool from the implementation process seems to be well received by many students. However, there were a few features which may have made the tool substantially more usable. The first of which is a connection with the students' virtual learning environment. The virtual learning environment currently in use at NTNU is Blackboard Learn, and through integration with this environment, the tool will be able to automatically fetch many of the students' tasks, which in turn reduces the need to add tasks manually. The second feature would consist of a more seamless integration with the student's personal calendar. The current method of manually importing and exporting calendar data through ICS-files is unfavoured by the majority of students.

By properly implementing the features which have been described above, many of the usability related problems of the tool may be substantially reduced. This may be of much help in further evaluation of the tool, as these usability issues are not necessarily what should be the focus of evaluation.

Several advanced objectives for modelling the scheduling problem have also been discussed. Of special interest is the model of variation proposed in chapter 9 and examining how this model may affect the resulting solutions to the scheduling problem.

Thus, the problem needs to be solved as a multi-objective optimization problem, which in turn would suggest a need for meta-heuristic algorithms.

Lastly, the rules for specifying the prearranging of breaks in the current tool are considerably simple. Since the beginning of this research, several students have expressed their needs for more advanced break regiments while working. Many students, for instance are accustomed to the so-called Pomodoro technique, which includes a variation in the lengths and frequencies of breaks. A method for specifying more advanced break prearrangements may therefore be worth exploring.

Bibliography

[1] Akiflow. Akiflow - centralize your schedule. Available online at: https://akiflow.com/, last accessed on 2022-07-10.

[2] Akker, J. and Hoogeveen, J. Minimizing the number of late jobs in case of stochastic processing rimes with minimum success probabilities, Jan. 2004.

[3] Alario Hoyos, C., Estévez Ayres, I., Pérez Sanagustín, M., Leony, D., and Delgado Kloos, C. My learning mentor: a mobile app to support learners participating in moocs. *Journal of Universal Computer Science*, volume 21:735–753, July 2015.

[4] Alghamdi, H., Alsubait, T., Alhakami, H., and Baz, A. A review of optimization algorithms for university timetable scheduling. *Engineering, Technology & Applied Science Research*, volume 10number 6:6410–6417, Dec. 2020.

[5] Atlassian. Jira cloud. Available online at: https://www.atlassian.com/software/jira, last accessed on 2022-07-10.

[6] Brucker, P. *Classification of scheduling problems*. In *Scheduling Algorithms*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pages 1–10.

[7] Buehler, R., Griffin, D., and Ross, M. It's about time: optimistic predictions in work and love. *European Review of Social Psychology*, volume 6number 1:1–32, 1995.

[8] Chandrasekaran, K. and Simon, S. P. Multi-objective scheduling problem: hybrid approach using fuzzy assisted cuckoo search algorithm. *Swarm and Evolutionary Computation*, volume 5:1–16, 2012.

[9] Cheriyan, J., Ravi, R., and Skutella, M. A simple proof of the moore-hodgson algorithm for minimizing the number of late jobs. *Operations Research Letters*, volume 49number 6:842–843, 2021.

[10] Chu, S.-C., Chen, Y.-T., and Ho, J.-H. Timetable scheduling using particle swarm optimization. In *First International Conference on Innovative Computing, Information and Control - Volume I (ICICIC'06)*, volume 3, pages 324–327, 2006.

[11] Claessens, B., Eerde, W., Rutte, C., and Roe, R. A review of time management literature. *Personnel Review*, volume 36, Feb. 2007.

[12] Clockwise. Clockwise - time management for teams. Available online at: https://www.getclockwise.com/, last accessed on 2022-07-10.

[13] de Weerdt, M., Baart, R., and He, L. Single-machine scheduling with release times, deadlines, setup times, and rejection. *European Journal of Operational Research*, volume 291number 2:629–639, 2021.

[14] De Raadt, M. and Dekeyser, S. *A simple time-management tool for students' online learning activities.* PhD thesis, 2009, pages 194–199.

[15] Federgruen, A. and Mosheiov, G. Single machine scheduling problems with general breakdowns, earliness and tardiness costs. *Operations Research*, volume 45number 1:66–71, 1997.

[16] Ghaemi, S., Taghi Vakili, M., and Aghagolzadeh, A. Using a genetic algorithm optimizer tool to solve university timetable scheduling problem. In *2007 9th International Symposium on Signal Processing and Its Applications*, pages 1–4, 2007.

[17] Google. Google calendar. Available online at: https://calendar.google.com, last accessed on 2022-07-10.

[18] Graham, R., Lawler, E., Lenstra, J., and Kan, A. Optimization and approximation in deterministic sequencing and scheduling: a survey. In Hammer, P., Johnson, E., and Korte, B., editors, *Discrete Optimization II.* Volume 5, Annals of Discrete Mathematics, pages 287–326. Elsevier, 1979.

[19] Graves, G. H. and Lee, C.-Y. Scheduling maintenance and semiresumable jobs on a single machine. *Naval Research Logistics (NRL)*, volume 46number 7:845–863, 1999.

[20] Herrmann, J. W. and Lee, C.-Y. Solving a class scheduling problem with a genetic algorithm. *ORSA Journal on Computing*, volume 7number 4:443–452, 1995.

[21] Herroelen, W. and Leus, R. Robust and reactive project scheduling: a review and classification of procedures. *Katholieke Universiteit Leuven, Open Access publications from Katholieke Universiteit Leuven*, volume 42, Apr. 2004.

[22] Häfner, A., Oberst, V., and Stock, A. Avoiding procrastination through time management: an experimental intervention study. *Educational Studies*, volume 40:352–360, Mar. 2014.

[23] Itoh, T. and Ishii, H. Fuzzy due-date scheduling problem with fuzzy processing time. *International Transactions in Operational Research*, volume 6number 6:639–647, 1999.

[24] Jackson, J. *Scheduling a production line to minimize maximum tardiness.* Research report. Office of Technical Services, 1955.

[25] Janak, S. and Floudas, C. Production scheduling of a large-scale industrial batch plant. ii. reactive scheduling. *Industrial & Engineering Chemistry Research - IND ENG CHEM RES*, volume 45, May 2006.

[26] Janak, S. L., Lin, X., and Floudas, C. A. A new robust optimization approach for scheduling under uncertainty: ii. uncertainty with known probability distribution. *Computers & Chemical Engineering*, volume 31number 3:171–195, 2007.

[27] Kokkalis, N., Köhn, T., Huebner, J., Lee, M., Schulze, F., and Klemmer, S. R. Taskgenies. *ACM Transactions on Computer-Human Interaction*, volume 20:1–25, Nov. 2013.

[28] Lawler, E. L. and Moore, J. M. A functional equation and its application to resource allocation and sequencing problems. *Management Science*, volume 16number 1:77–84, 1969.

[29] Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A. H., and Shmoys, D. B. Chapter 9 sequencing and scheduling: algorithms and complexity. In *Logistics of Production and Inventory*. Volume 4, Handbooks in Operations Research and Management Science, pages 445–522. Elsevier, 1993.

[30] Lee, C. Machine scheduling with an availability constraint. *Journal of Global Optimization*, volume 9:395–416, 1996.

[31] Litoiu, M. and Tadei, R. Real-time task scheduling with fuzzy deadlines and processing times. *Fuzzy Sets and Systems*, volume 117number 1:35–45, 2001.

[32] Liu, M., Wang, S., Chu, C., and Chu, F. An improved exact algorithm for single-machine scheduling to minimise the number of tardy jobs with periodic maintenance. *International Journal of Production Research*, volume 54number 12:3591–3602, 2016.

[33] Lund, J. R. and Wiese, J. Less is more: exploring support for time management planning. In Designing Interactive Systems Conference 2021, June 2021.

[34] Macan, T. H., Shahani, C., Dipboye, R. L., and Phillips, A. P. College students' time management: correlations with academic performance and stress. *Journal of Educational Psychology*, volume 82:760–768, 1990.

[35] McCown, W. G., Johnson, J. L., and Petzel, T. Procrastination, a principal components analysis. *Personality and Individual Differences*, volume 10:197–202, 1989.

[36] Mihoubi, B., Bouzouia, B., and Gaham, M. Reactive scheduling approach for solving a realistic flexible job shop scheduling problem. *International Journal of Production Research*, volume 59number 19:5790–5808, 2021.

[37] Moore, J. M. An n job, one machine sequencing algorithm for minimizing the number of late jobs. *Management Science*, volume 15number 1:102–109, 1968.

[38] Moslehi, G. and Mahnam, M. A pareto approach to multi-objective flexible job-shop scheduling problem using particle swarm optimization and local search. *International Journal of Production Economics*, volume 129number 1:14–22, 2011.

[39] Muhuri, P. and Shukla, K. Real-time task scheduling with fuzzy uncertainty in processing times and deadlines. *Applied Soft Computing*, volume 8:1–13, Jan. 2008.

[40] Nesello, V., Subramanian, A., Battarra, M., and Laporte, G. Exact solution of the single-machine scheduling problem with periodic maintenances and sequence-dependent setup times. *European Journal of Operational Research*, volume 266number 2:498–507, 2018.

[41] Ng, Y.-K. *Pareto optimality.* In Jan. 2004, pages 26–46.

[42] Nichols, A. L. and Maner, J. K. The good-subject effect: investigating participant demand characteristics. *The Journal of General Psychology*, volume 135:151–166, Apr. 2008.

[43] Nordby, K., Wang, C., Dahl, T., and Svartdal, F. Intervention to reduce procrastination in first-year students: preliminary results from a norwegian study. *Scandinavian Psychologist*, volume 3, June 2016.

[44] Patton, M. Q. *Qualitative Research and Evaluation Methods: Integrating Theory and Practice*. Sage Publications, Inc, 4th edition, 2015.

[45] Pérez-Álvarez, R., Maldonado-Mahauad, J. J., Sapunar-Opazo, D., and Pérez-Sanagustín, M. Notemyprogress: a tool to support learners' self-regulated learning strategies in mooc environments. In pages 460–466. Data Driven Approaches in Digital Education, Sept. 2017.

[46] Perezgonzalez, J. D. Fisher, NeymanPearson or NHST? A Tutorial for Teaching Data Testing. *Frontiers in Psychology*, volume 6:223–223, Mar. 2015.

[47] Pezzo, M. V., Litman, J. A., and Pezzo, S. P. On the distinction between yuppies and hippies: individual differences in prediction biases for planning future tasks. *Personality and Individual Differences*, volume 41number 7:1359–1371, 2006.

[48] Pinedo, M. L. *Advanced single machine models (deterministic)*. In *Scheduling: Theory, Algorithms, and Systems*. Springer International Publishing, Cham, 2016, pages 71–111.

[49] Pinedo, M. L. *Deterministic models: preliminaries*. In *Scheduling: Theory, Algorithms, and Systems*. Springer International Publishing, Cham, 2016, pages 13–32.

[50] Pinedo, M. L. *Modeling and solving scheduling problems in practice*. In *Scheduling: Theory, Algorithms, and Systems*. Springer International Publishing, Cham, 2016, pages 431–458.

[51] Pinedo, M. L. *Single machine models (deterministic)*. In *Scheduling: Theory, Algorithms, and Systems*. Springer International Publishing, Cham, 2016, pages 33–69.

[52] Pinedo, M. L. *Single machine models (stochastic)*. In *Scheduling: Theory, Algorithms, and Systems*. Springer International Publishing, Cham, 2016, pages 265–291.

[53] Portoleau, T., Artigues, C., and Guillaume, R. Robust predictive-reactive scheduling: an information-based decision tree model. In Lesot, M.-J., Vieira, S., Reformat, M. Z., Carvalho, J. P., Wilbik, A., Bouchon-Meunier, B., and Yager, R. R., editors, *Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 479–492, Cham. Springer International Publishing, 2020.

[54] Pratt, J. W. and Gibbons, J. D. *Concepts of Nonparametric Theory*. New York, Ny Springer New York, 1981.

[55] Rabin, L. A., Fogel, J., and Nutter-Upham, K. E. Academic procrastination in college students: the role of self-reported executive function. *Journal of Clinical and Experimental Neuropsychology*, volume 33number 3:344–357, 2011. PMID: 21113838.

[56] Reclaim. Reclaim | make time when there is none. Available online at: https://reclaim.ai/, last accessed on 2022-07-10.

[57] Rego, M., Souza, M., and Arroyo, J. E. Multi-objective algorithms for the single machine scheduling problem with sequence-dependent family setups. In volume 223, pages 142–151, Nov. 2012.

[58] Rosenthal, R., Kazdin, A. E., and Rosnow, R. L. *Artifacts in behavioral research : Robert Rosenthal and Ralph L. Rosnow's classic books : a re-issue of Artifact in behavioral research, Experimenter effects in behavioral research and the volunteer subject.* Oxford University Press, 2009.

[59] Schouwenburg, H. C. *Academic procrastination.* In *Procrastination and Task Avoidance: Theory, Research, and Treatment.* Springer US, Boston, MA, 1995, pages 71–96.

[60] Seiden, S. Randomized online scheduling with delivery times. *Journal of Combinatorial Optimization*, volume 3:399–416, 1999.

[61] Senécal, C., Koestner, R., and Vallerand, R. Self-regulation and academic procrastination. *Journal of Social Psychology - J SOC PSYCHOL*, volume 135:607–619, Oct. 1995.

[62] Shovel. Shovel study planner. Available online at: https://shovelapp.io/, last accessed on 2022-07-10.

[63] Sigl, B., Golub, M., and Mornar, V. Solving timetable scheduling problem using genetic algorithms. In *Proceedings of the 25th International Conference on Information Technology Interfaces, 2003. ITI 2003.* Pages 519–524, 2003.

[64] Smith, W. E. Various optimizers for single-stage production. *Naval Research Logistics Quarterly*, volume 3number 1-2:59–66, 1956.

[65] Steel, P. The nature of procrastination: a meta-analytic and theoretical review of quintessential self-regulatory failure. psychol bull 133: 65-94. *Psychological bulletin*, volume 133:65–94, Feb. 2007.

[66] Stougie, L. and Vestjens, A. P. Randomized algorithms for on-line scheduling problems: how low can't you go? *Operations Research Letters*, volume 30number 2:89–96, 2002.

[67] Task management - android apps on google play. Available online at: https://play.google.com/store/search?q=task%20management&c=apps&gl=NO, last accessed on 2022-06-30.

[68] Taskito. Taskito. Available online at: https://taskito.io/, last accessed on 2022-07-10.

[69] Timeliness. Timeliness: task planner. Available online at: https://timeliness.app/, last accessed on 2022-07-10.

[70] Torres-Ovalle, C., Montoya-Torres, J., Quintero-Araujo, C., Sarmiento, A., and Luna, M. University course scheduling and classroom assignment. *Ingenieria y Universidad*, volume 18:59–75, June 2014.

[71] usability.gov. System Usability Scale (SUS), 2019. Available online at: https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html, last accessed on 2022-07-10.

[72] Vestjens, A. *On-line machine scheduling.* English. PhD thesis, Mathematics and Computer Science, 1997.

[73] Wang, S., Liu, M., Chu, F., and Chu, C. Bi-objective optimization of a single machine batch scheduling problem with energy cost consideration. *Journal of Cleaner Production*, volume 137:1205–1215, 2016.

[74] White, H. A reality check for data snooping. *Econometrica*, volume 68number 5:1097–1126, 2000. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/1468-0262.00152.

[75] Wierzbicki, A. P. A mathematical basis for satisficing decision making. *Mathematical Modelling*, volume 3number 5:391–405, 1982. Special IIASA Issue.

[76] Xanthopoulos, A., Koulouriotis, D., and Tourassis, V. *Multi-objective scheduling on a single machine with evolutionary algorithm.* In Apr. 2011.

[77] Zhang, Y., Zhu, H., and Tang, D. An improved hybrid particle swarm optimization for multi-objective flexible job-shop scheduling problem. *Kybernetes*, volume ahead-of-print, Dec. 2019.

Appendices

APPENDIX A

Report of Statistical Values and Plots

A.1 Statistics from the Survey on Study Preferences

Statistical Summary of Study Preferences Given a Sample Size of n

Variable	n	$\hat{\mu}$	Std. Dev	Sample Median
Length Of Study Day	66	0.34	0.08	0.33
Actual Time Doing Exercises Per Day	66	3.39	1.77	3.00
Optimal Time Doing Exercises Per Day	66	4.09	1.94	4.00
Actual Additional Study Time Per Day	66	1.48	1.40	1.00
Optimal Additional Study Time Per Day	66	2.39	1.75	2.00
Start Of Study Day	66	0.38	0.05	0.38
End Of Study Day	66	0.72	0.08	0.71
Study Session Length Minimum	66	105.83	72.07	120.00
Study Session Length Maximum	66	115.83	70.48	120.00
Break Length Minimum	66	24.77	21.42	20.00
Break Length Maximum	66	32.27	31.73	20.00
Degree Of Planning	66	2.95	1.01	3.00
Number Of Days Planned For At Once	66	4.00	2.55	3.00
Degree Of Plan Adherence	66	3.14	0.88	3.00
Importance Of Deadlines	66	4.47	0.79	5.00
Importance Of Variation	66	2.47	1.08	2.00
Importance Of Break Frequency	66	2.82	1.20	3.00
Age	66	22.08	1.69	22.00
Number Of Courses	66	3.68	1.36	4.00

Table A.1: Statistical Summary of Survey Responses

Statistical Summary of Study Preferences Calculated in Alternative Units of Measurement				
Variable	n	$\hat{\mu}$	Std. Dev	Sample Median
Length of Study Day	66	8h 9m 36s	1h 55m 12s	7h 55m 12s
Actual Time Doing Exercises Per Day	66	3h 23m 24s	1h 46m 12s	3h
Optimal Time Doing Exercises Per Day	66	4h 5m 4s	1h 56m 24s	4h
Actual Additional Study Time Per Day	66	1h 28m 48s	1h 24m	1h
Optimal Additional Study Time Per Day	66	2h 23m 24s	1h 45m	2h
Start of Study Day	66	09:07	1h 12m	09:07
End of Study Day	66	17:17	1h 55m 12s	17:16
Study Session Length Minimum	66	1h 45m 50s	1h 12m 4s	2h
Study Session Length Maximum	66	1h 55m 50s	1h 10m 28s	2h
Break Length Minimum	66	24m 46s	21m 25s	20m
Break Length Maximum	66	32m 16s	31m 44s	20m
Degree of Planning	66	2.95	1.01	3.00
Number of Days Planned For At Once	66	4.00	2.55	3.00
Degree of Plan Adherence	66	3.14	0.88	3.00
Importance of Deadlines	66	4.47	0.79	5.00
Importance of Variation	66	2.47	1.08	2.00
Importance of Break Frequency	66	2.82	1.20	3.00
Age	66	22.08	1.69	22.00
Number of Courses	66	3.68	1.36	4.00

Table A.2: Statistical Summary of Survey Responses With Recalculated Units of Measurement

Plots of Variables from the Survey on Study Preferences

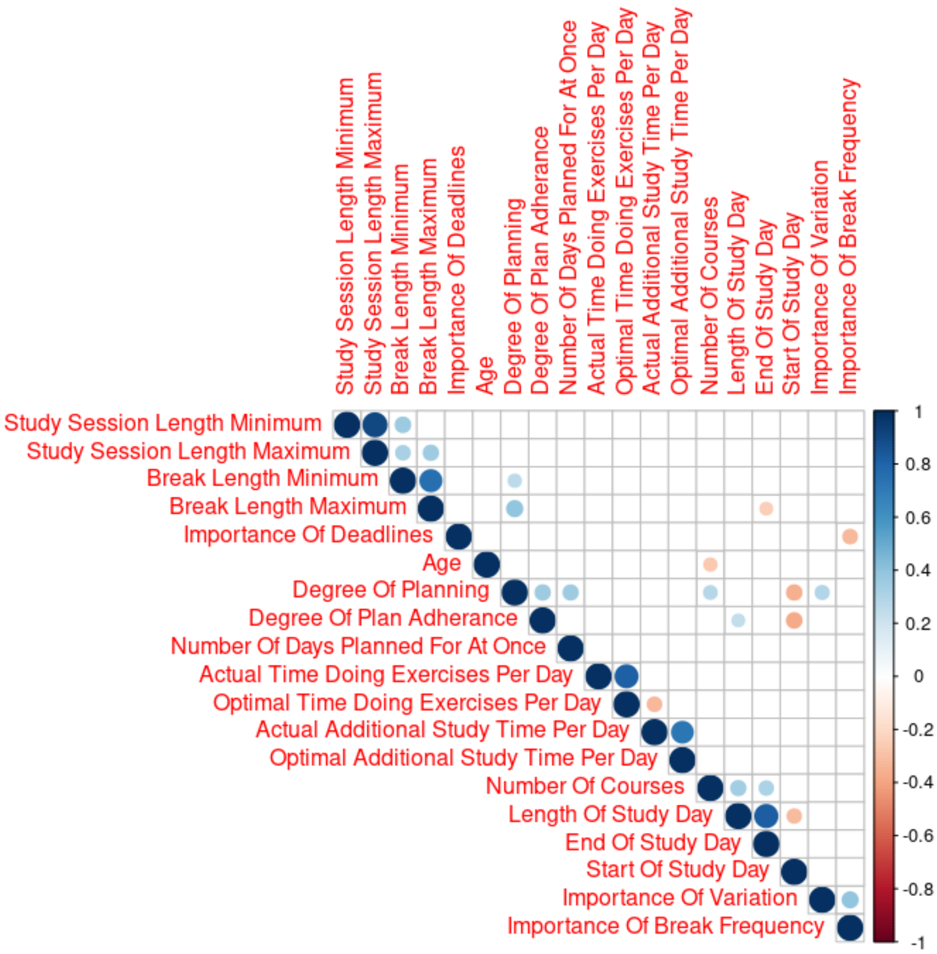

Figure A.1: Correlation Matrix with Non-Significant Correlations Excluded

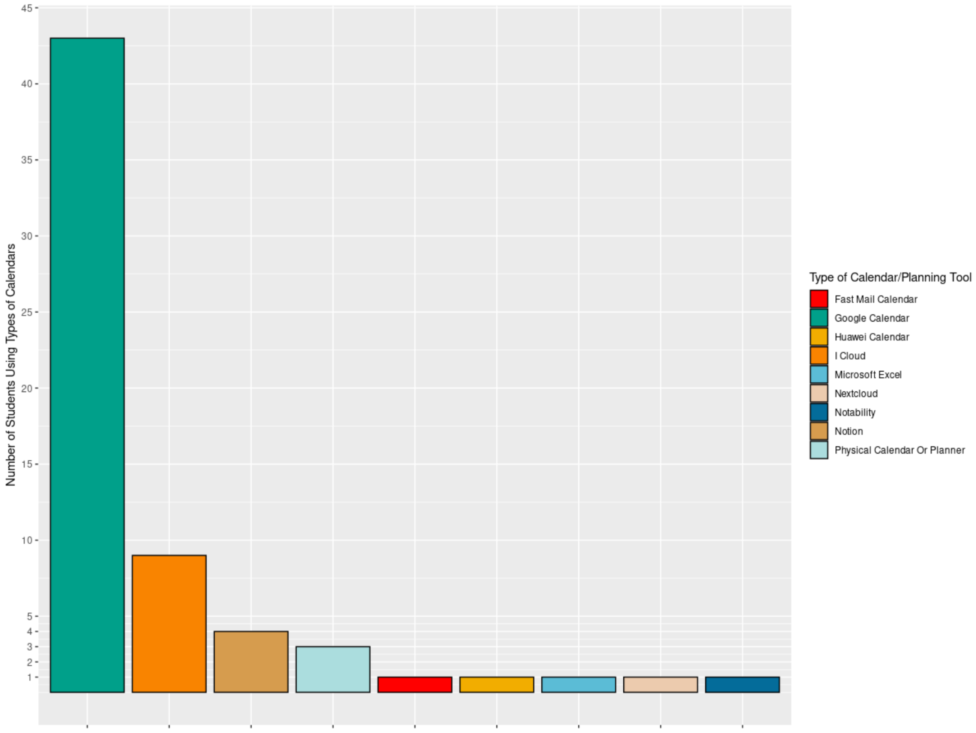

Figure A.2: Types of Calendars and Planning Tools in Use as Reported by the Survey

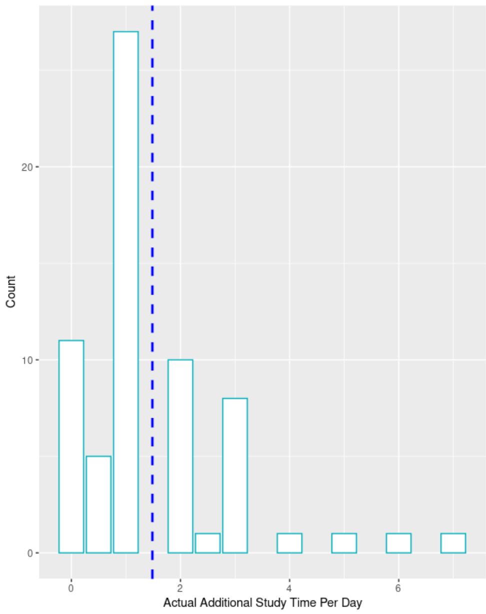

Figure A.3: Actual additional study time per day

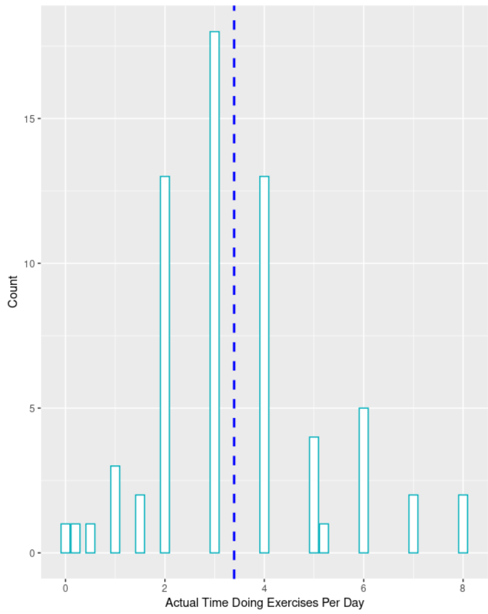

Figure A.4: Actual time doing exercises per day

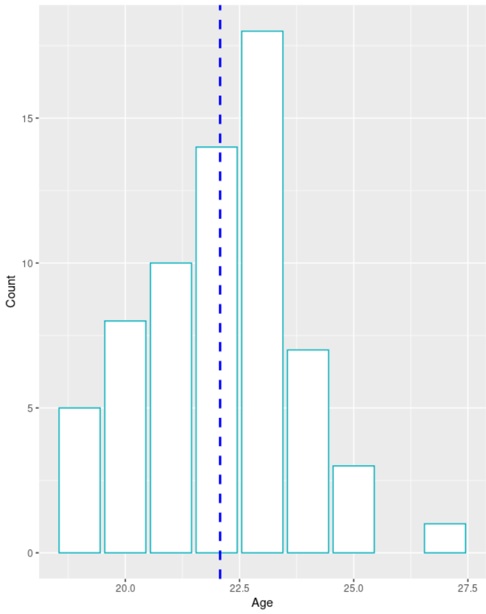

Figure A.5: Age

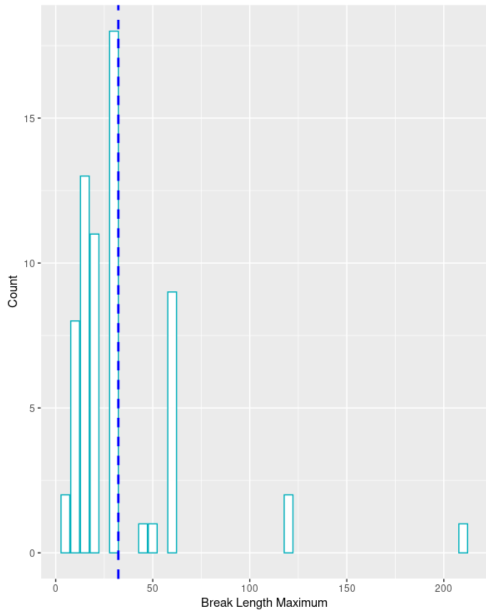

Figure A.6: Break length maximum

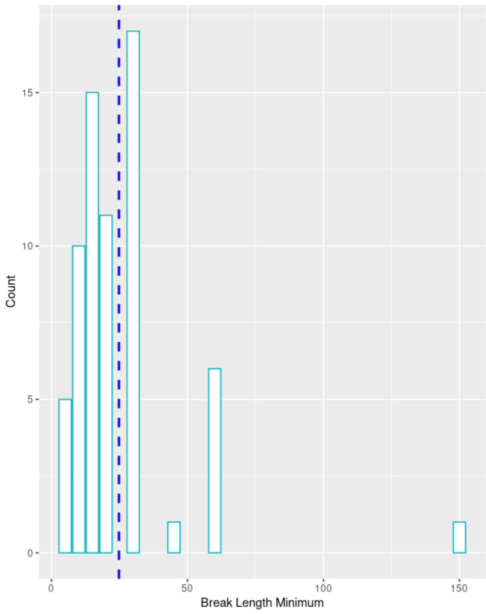

Figure A.7: Break length minimum

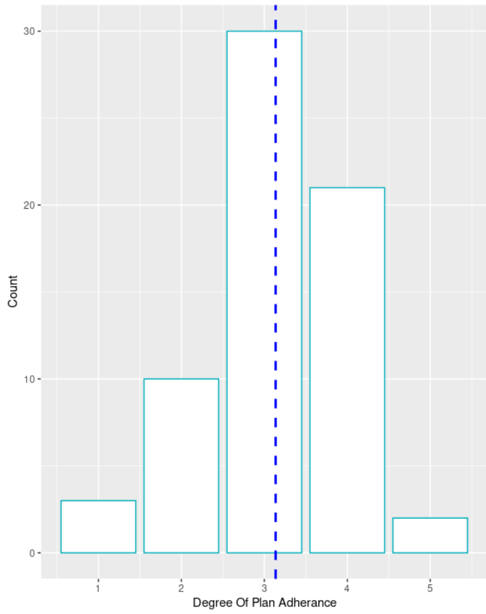

Figure A.8: Degree of plan adherence

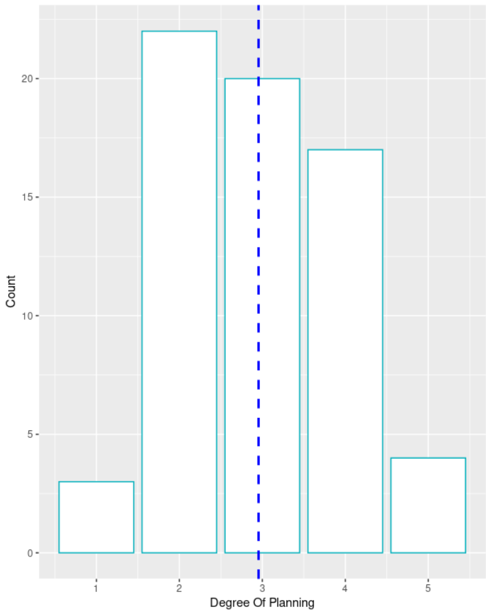

Figure A.9: Degree of planning

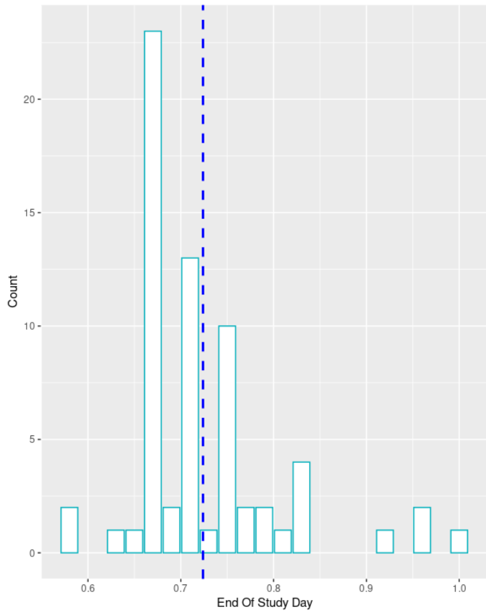

Figure A.10: End of study day

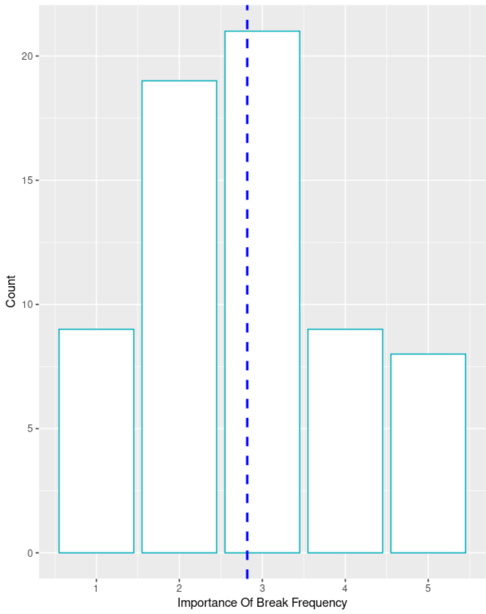

Figure A.11: Importance of break frequency

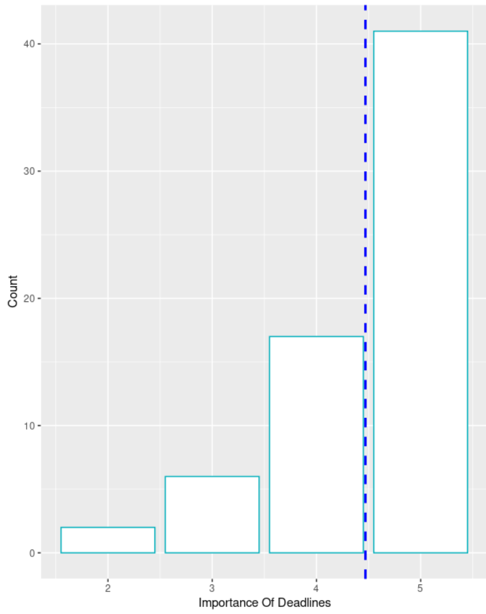

Figure A.12: Importance of deadlines

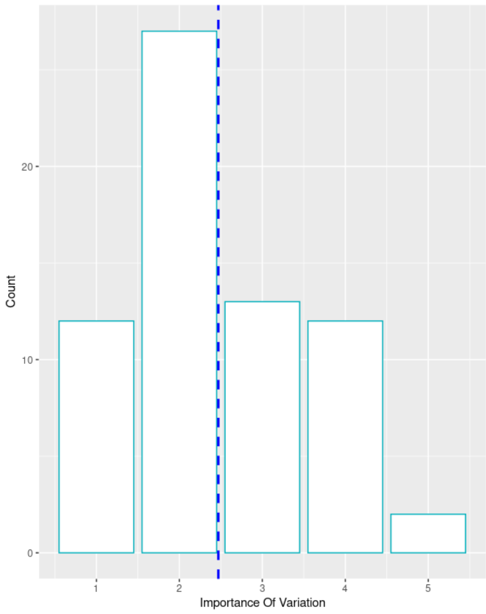

Figure A.13: Importance of variation

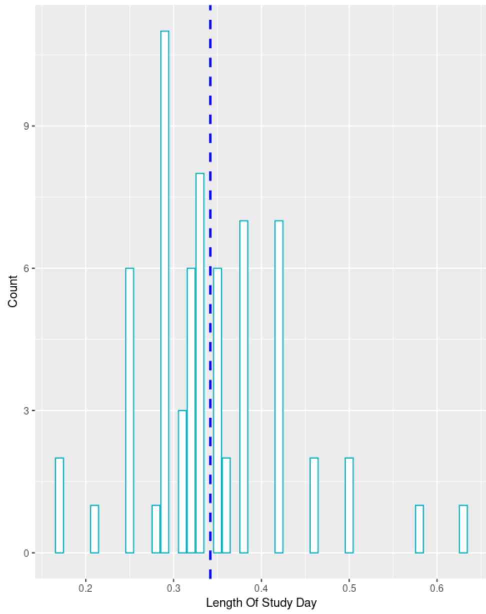

Figure A.14: Length of study day

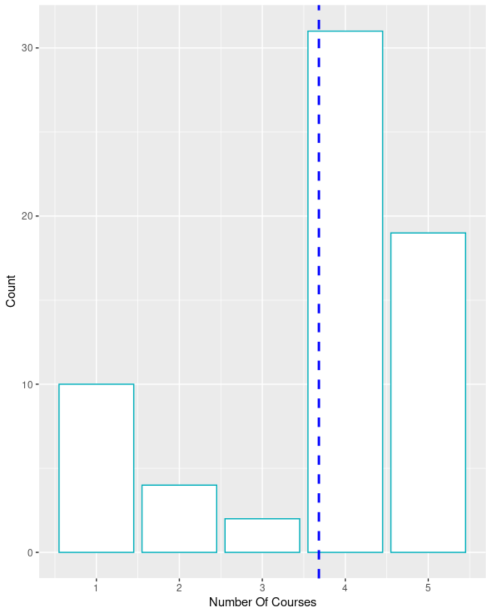

Figure A.15: Number of courses

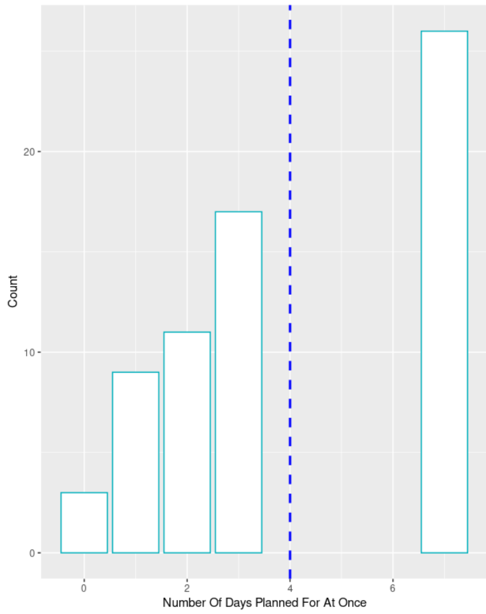

Figure A.16: Number of days planned for

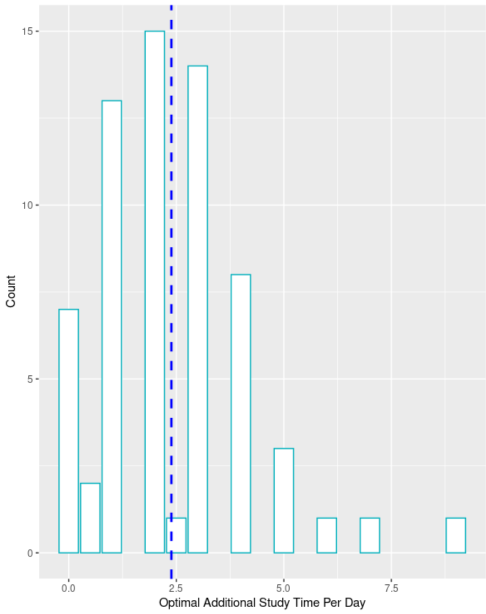

Figure A.17: Optimal additional study time per day

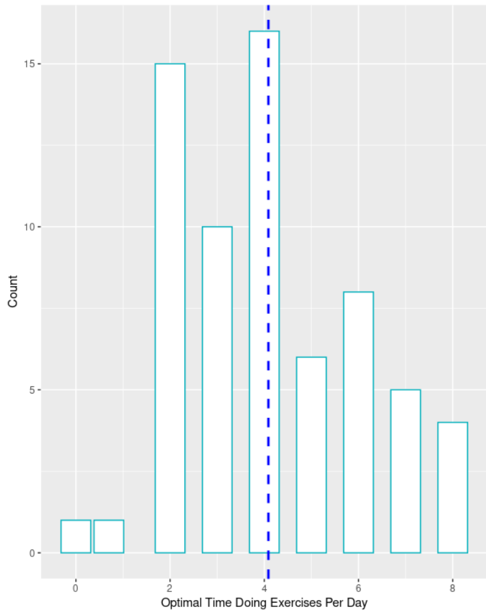

Figure A.18: Optimal time doing exercises per day

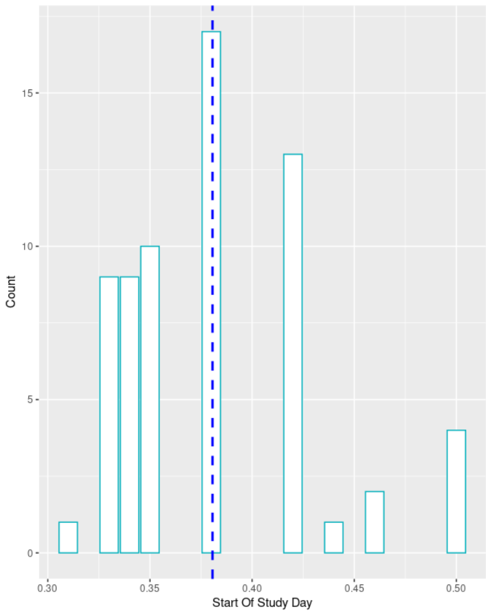

Figure A.19: Start of study day

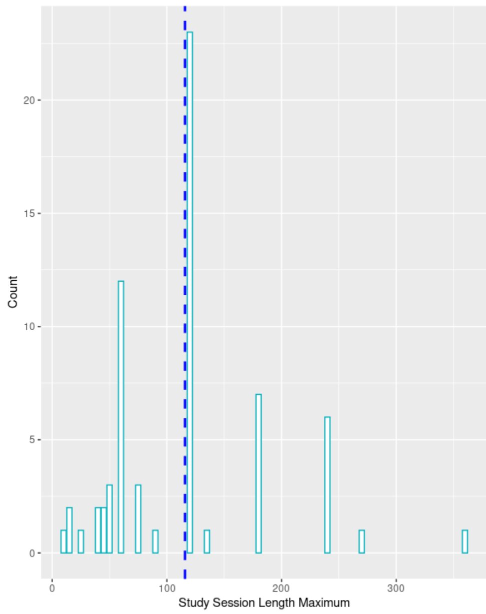

Figure A.20: Study session length maximum

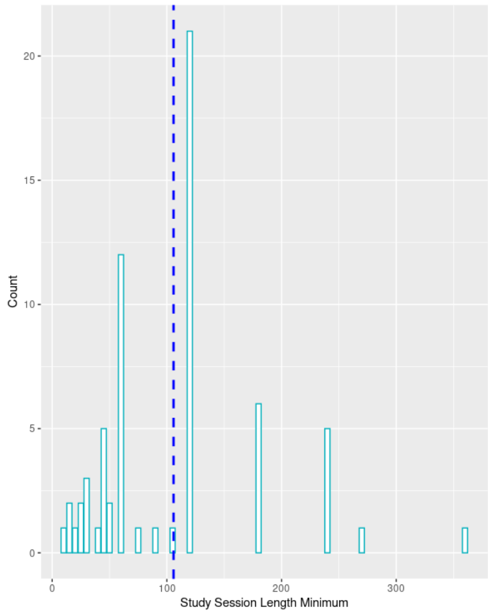

Figure A.21: Study session length minimum

A.2 Statistics from the Quantitative Evaluation

Statistical Summary of Quantitative Evaluation Given a Sample Size of n				
Variable	n	$\hat{\mu}$	Std. Dev	Sample Median
Seperate Study Time From Other Time Preference	17	3.82	0.73	4
Importance of the Tool Knowing My Available Time	17	4.29	0.59	4
Rating of Preferred Study Time Setting	17	3.53	0.94	3
Rating of Manually Adding Unavailable Period	17	3.88	0.99	4
Rating of Calendar Integration	17	3.24	1.09	3
Satisfaction with Functionality for Showing Unavailable Time	17	3.53	0.94	3
Unnecessary Time Spent Navigating the Tool	17	2.47	1.33	2
Time Spent Editing Plan	17	2.71	1.16	3
Easiness of Editing	17	3.41	1.00	3
Satisfaction with Functionality for Editing	17	3.94	0.90	4
Rating of the Plan's Self Correcting Behaviour	17	3.94	0.90	4
Initial Satisfaction with the Generated Plan	17	3.18	1.13	3
Need For Monitoring of Time Spent on Existing Tasks	17	3.29	1.05	4
Need for More Variation	17	3.29	1.40	3
Could Imagine Using the Tool in the Future	17	3.41	1.23	4
Could Imagine Using the Tool for General Planning	17	3.00	1.17	3

Table A.3: Statistical Summary of Likert Type Variables

Normality of the Data

Results of the Lilliefors test indicated that there is a non-significant difference from the normal distribution for the following variables:

1. *Rating of Calendar Integration*

$$D(17) = .182,\ p = .141$$

2. *Need For More Variation*

$$D(17) = .179,\ p = .155$$

However, after considering the QQ-plots and histograms of each distribution, as shown in Figure A.22, Figure A.23, Figure A.24 and Figure A.25, the two distributions were deemed as non-normal.

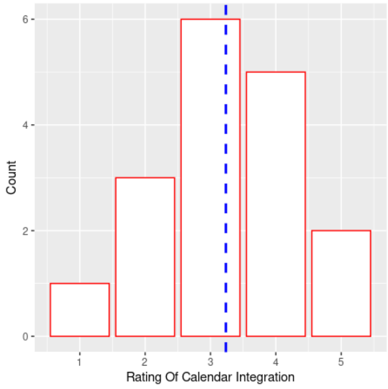

Figure A.22: Histogram Showing the Distribution of “Rating of Calendar Integration”

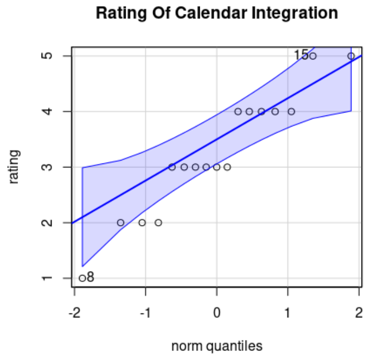

Figure A.23: QQ-plot Showing the Normal Quantiles of “Rating of Calendar Integration”

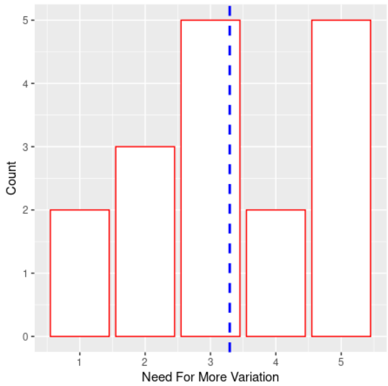

Figure A.24: Histogram Showing the Distribution of “Need for More Variation”

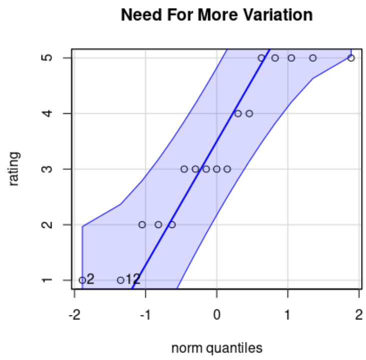

Figure A.25: QQ-plot Showing the Normality Quantiles of “Need for More Variation”

Results of Wilcoxon Signed-Rank Test and Sign Test

Variable	Sign Test	p-Value of Sign Test	Wilcoxon Signed-Rank Test	p-Value of Wilcoxon
Importance Of The Tool Knowing My Available Time	$m > \mathbf{3}$	1.526e-05	$m > \mathbf{4}$	.0363
Rating Of Self Adjusting Behaviour	$m > 3$	.0017	$m > 3$	.0017
Separate Study Time From Other Time Preference	$m > 3$	.0005	$m > 3$	.0012
Satisfaction With Functionality For Editing	$m > 3$	.0010	$m > 3$	.0023
Rating Of Manually Adding Unavailable Period	$m > 3$	.0059	$m > 3$	.0035
Could Imagine Using The Tool In The Future	$m > 2$	.0017	$m > 2$	.0011
Need For Monitoring Of Time Spent	$m > 2$	.0009	$m > 2$	.0008
Rating Of Preferred Study Time Setting	$m > \mathbf{2}$	3.052e-05	$m > \mathbf{3}$	.0219
Initial Satisfaction With Generated Plan	$m > 2$	.0037	$m > 2$	.0014
Need For More Variation	$m > 2$	.0065	$m > 2$	.0024
Rating Of Calendar Integration	$m > 2$	.0009	$m > 2$	.0011
Easiness Of Editing	$m > 2$	6.104e-05	$m > 2$	.0004
Could Imagine Using The Tool For General Planning	$m > 2$	.0106	$m > 2$	.0032
Satisfaction With Functionality For Showing Unavailable Time	$m > \mathbf{2}$	1.526e-05	$m > \mathbf{3}$	.0231
Unnecessary Time Spent Navigating the Tool	$m > 1$	.0002	$m > 1$	.0011
Time Spent Editing Plan	$m > 2$	.0327	$m > 2$	.0145

Table A.4: Results of the Sign and Wilcoxon Signed-Rank Tests for the Likert Type Variables

Plots of Likert Type Variables

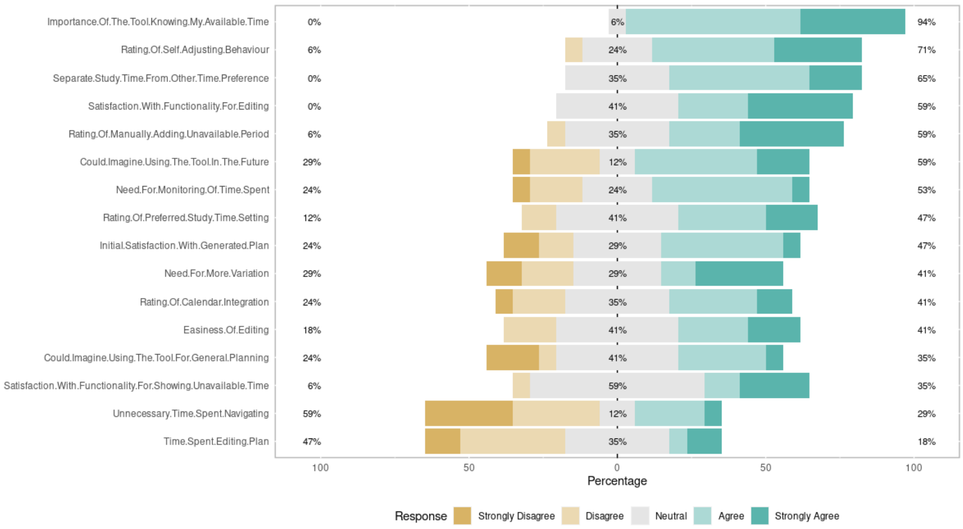

Figure A.26: Likert Plot of the Likert Type Variables from the Quantitative Testing

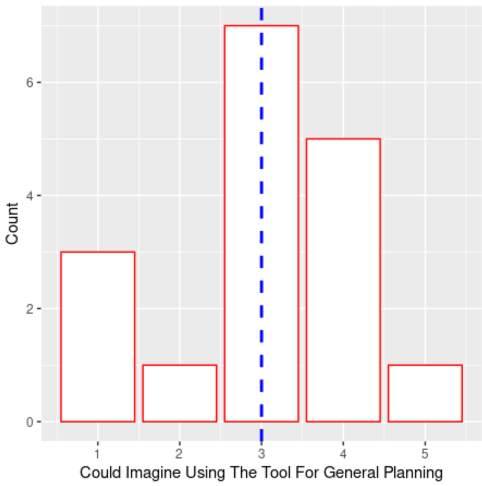

Figure A.27: Could Imagine Using The Tool For General Planning

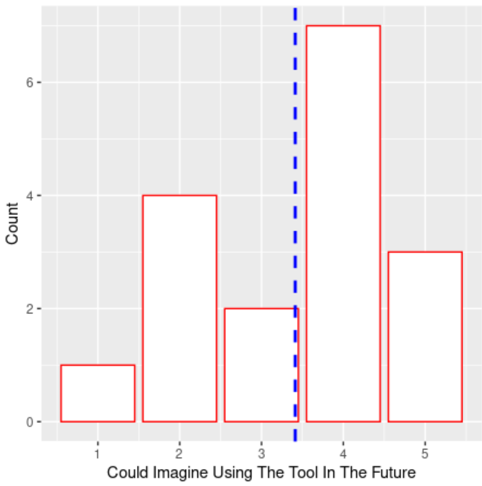

Figure A.28: Could Imagine Using The Tool In The Future

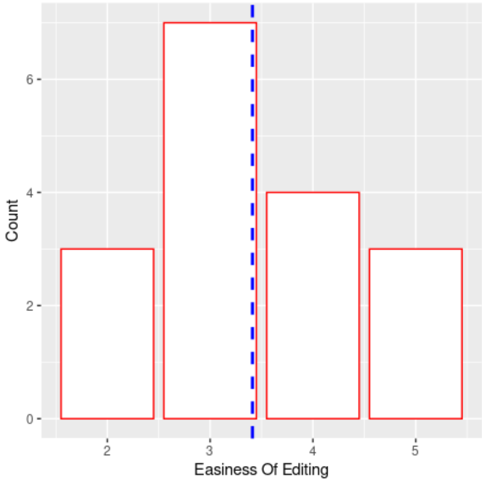

Figure A.29: Easiness Of Editing

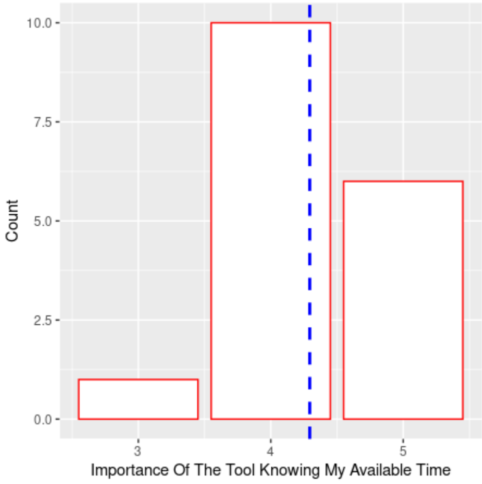

Figure A.30: Importance of the Tool Knowing my Available Time

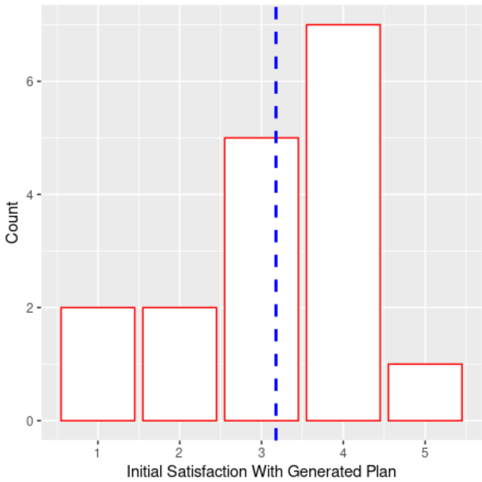

Figure A.31: Initial Satisfaction With Generated Plan

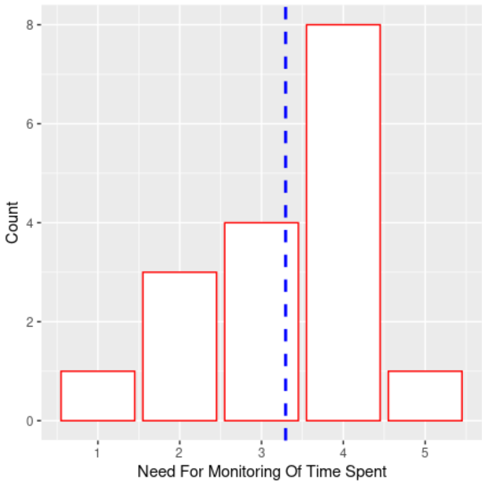

Figure A.32: Need For Monitoring Time Spent

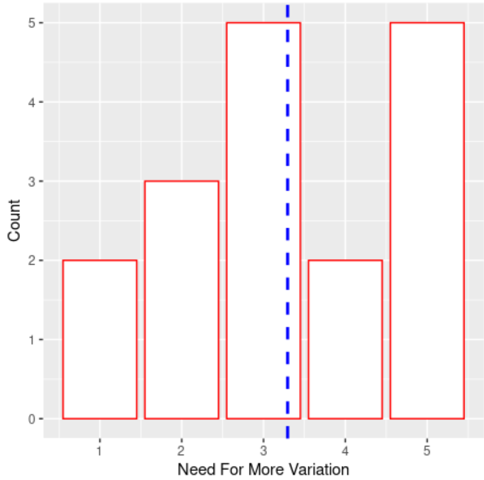

Figure A.33: Need For More Variation

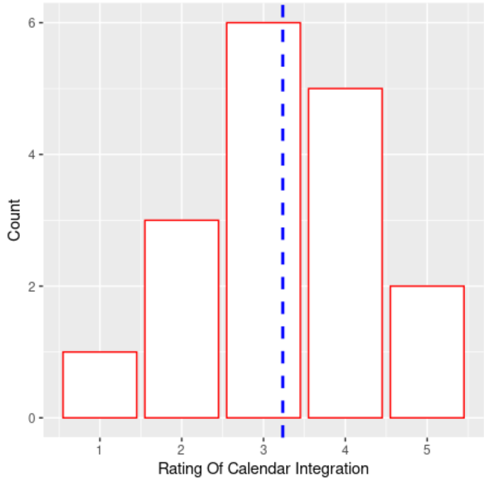

Figure A.34: Rating Of Calendar Integration

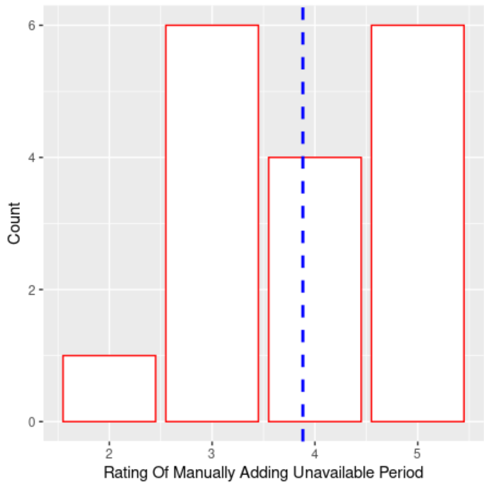

Figure A.35: Rating Of Manually Adding Unavailable Period

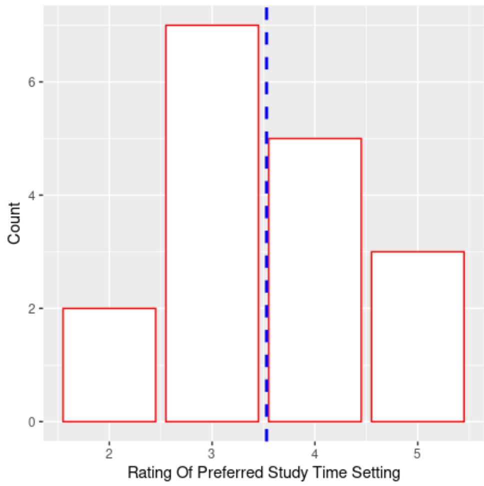

Figure A.36: Rating Of Preferred Study Time Setting

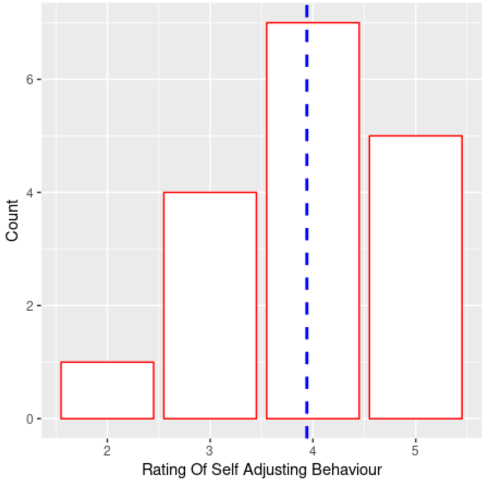

Figure A.37: Rating Of Self Adjusting Behaviour

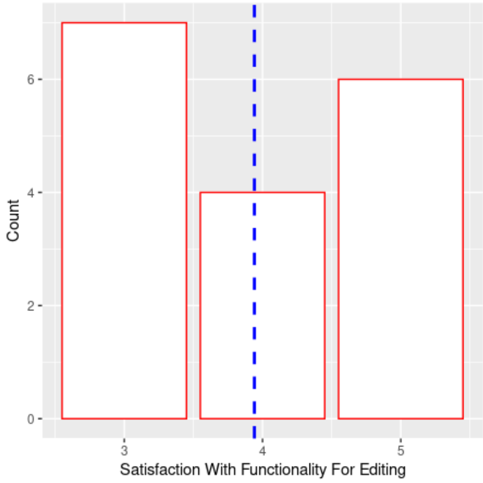

Figure A.38: Satisfaction With Functionality For Editing

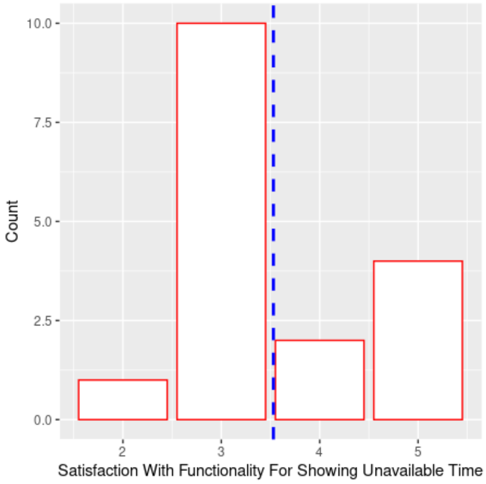

Figure A.39: Satisfaction With Functionality For Showing Unavailable Time

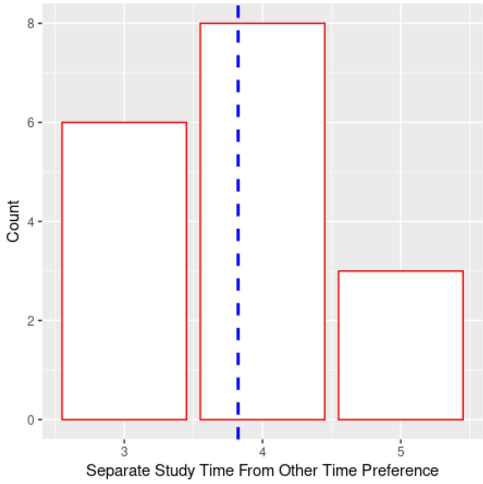

Figure A.40: Separate Study Time From Other Time Preferences

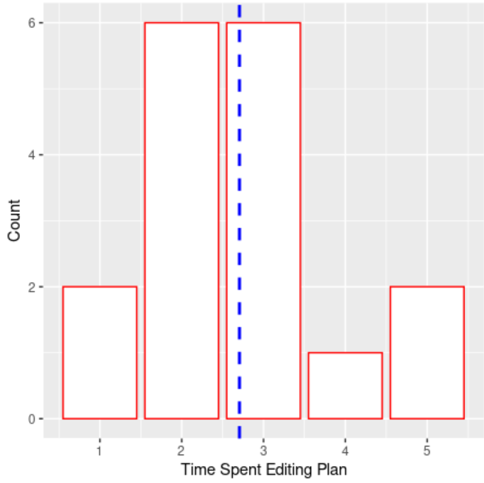

Figure A.41: Time Spent Editing Plan

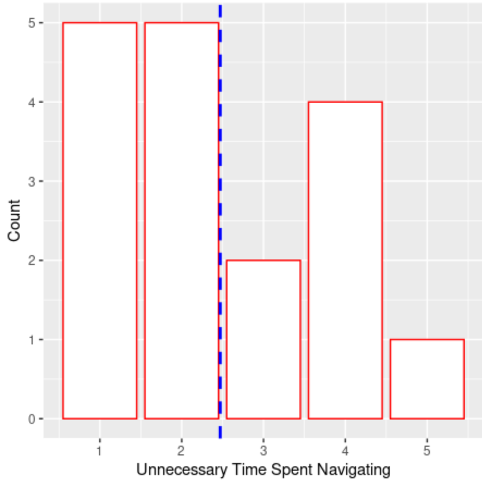

Figure A.42: Unnecessary Time Spent Navigating

Categorical Data

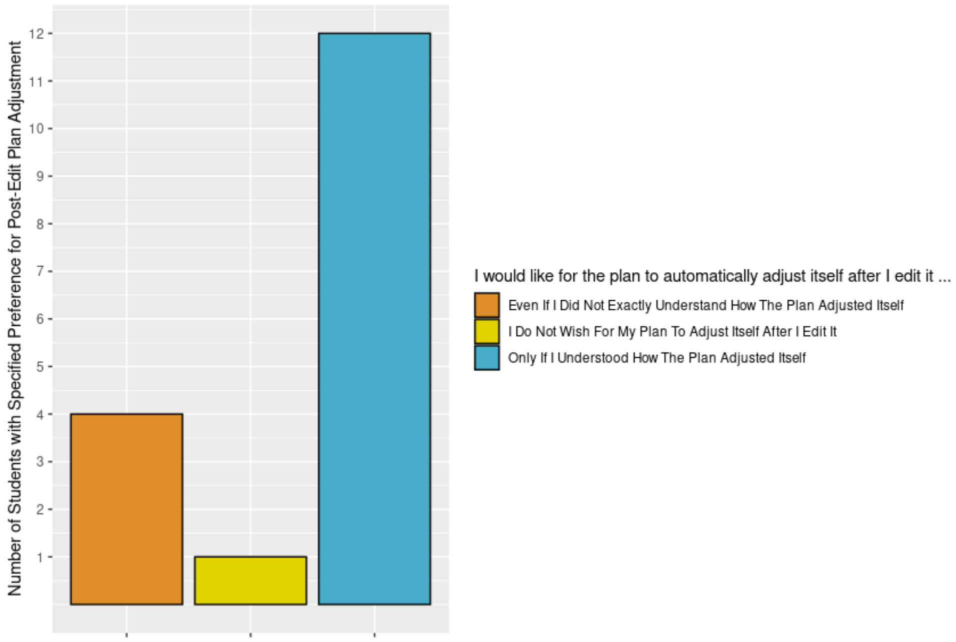

Figure A.43: Preference for Post-Edit Plan Adjustment

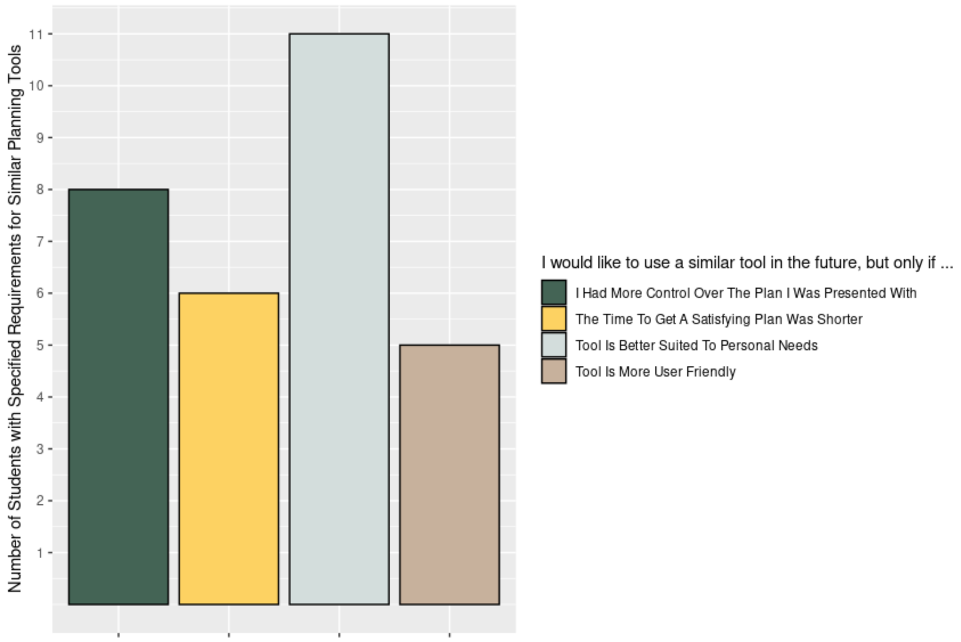

Figure A.44: Requirements for Wanting to Use a Similar Tool

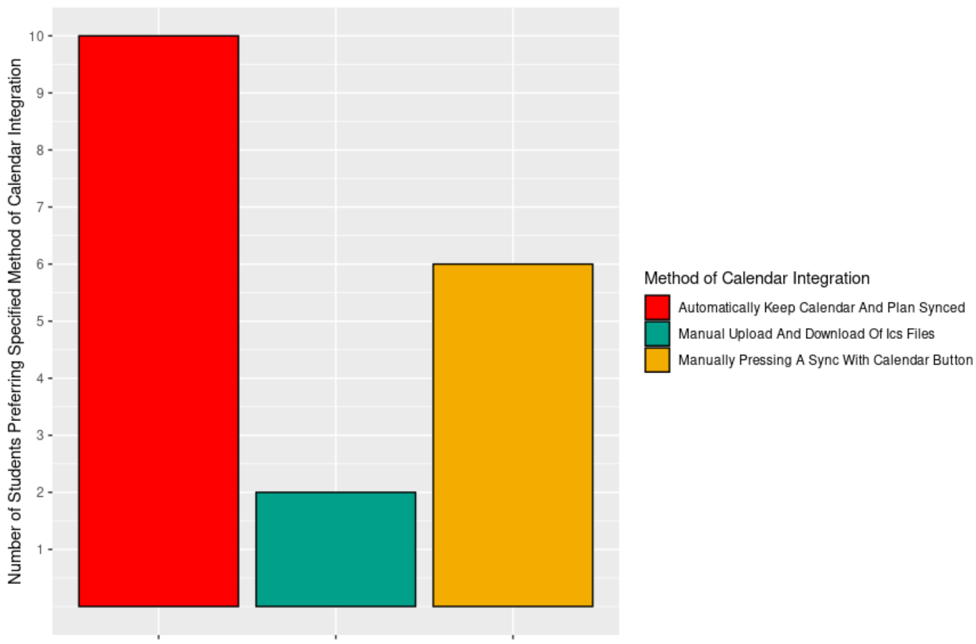

Figure A.45: Preference for Method of Calendar Integration

Principal Component Analysis

Eigenvalues from the Principal Component Analysis

Principal Component	Eigenvalue	Explanation of Variance	Cumulative Explanation of Variance
PC1	3.7	22.8%	22.8%
PC2	2.9	18.0%	40.9%
PC3	2.4	15.2%	56.0%
PC4	1.8	11.0%	67.1%
PC5	1.4	8.8%	75.9%
PC6	1.1	6.8%	82.6%
PC7	0.8	4.7%	87.4%
PC8	0.6	4.0%	91.4%
PC9	0.5	3.1%	94.6%
PC10	0.3	2.2%	96.7%
PC11	0.2	1.4%	98.1%
PC12	0.2	1.0%	99.1%
PC13	0.1	0.4%	99.6%
PC14	0.1	0.3%	99.9%
PC15	0.0	0.1%	100.0%
PC16	0.0	0.0%	100.0%

Table A.5: Eigenvalues from the Principal Component Analysis

Eigenvectors 1-8 from Principal Component Analysis

Variable	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8
Separate Study Time From Other Time Preference	-0.12	0.39	-0.27	0.18	-0.04	0.06	-0.48	0.29
Importance Of The Tool Knowing My Available Time	-0.30	0.35	-0.03	0.08	-0.18	0.37	-0.01	-0.02
Rating Of Preferred Study Time Setting	-0.43	0.08	-0.05	0.00	0.21	-0.05	0.32	-0.25
Rating Of Manually Adding Unavailable Period	-0.12	0.27	0.38	-0.41	-0.02	-0.12	0.03	0.11
Rating Of Calendar Integration	-0.09	-0.02	0.25	0.43	0.12	0.22	0.52	0.48
Satisfaction With Functionality For Showing Unavailable Time	-0.30	0.08	0.38	-0.09	0.25	0.30	-0.13	0.08
Unnecessary Time Spent Navigating	0.10	0.22	-0.27	-0.17	-0.42	0.44	0.31	-0.23
Time Spent Editing Plan	0.39	0.16	-0.09	-0.05	-0.01	0.02	0.40	0.23
Easiness Of Editing	-0.24	-0.40	0.00	0.14	-0.21	0.32	-0.19	0.03
Satisfaction With Functionality For Editing	-0.26	-0.33	0.20	0.05	-0.37	0.03	0.10	-0.27
Rating Of Self Adjusting Behaviour	-0.21	-0.09	0.15	-0.48	-0.33	-0.25	0.07	0.38
Initial Satisfaction With Generated Plan	-0.35	0.11	-0.19	0.21	0.25	-0.38	0.16	-0.17
Need For Monitoring Of Time Spent	0.23	0.14	0.43	0.22	0.00	0.09	-0.20	0.01
Need For More Variation	0.20	0.01	0.33	0.39	-0.35	-0.27	0.00	-0.18
Could Imagine Using The Tool In The Future	0.07	0.41	0.31	-0.04	0.05	0.01	0.03	-0.44
Could Imagine Using The Tool For General Planning	-0.22	0.29	-0.08	0.24	-0.44	-0.34	0.02	0.18

Table A.6: Eigenvectors 1-8 from Principal Component Analysis

Eigenvectors 9-16 from Principal Component Analysis

Variable	PC9	PC10	PC11	PC12	PC13	PC14	PC15	PC16
Separate Study Time From Other Time Preference	-0.08	0.29	0.17	-0.02	-0.19	-0.28	0.02	-0.42
Importance Of The Tool Knowing My Available Time	0.03	-0.45	-0.08	-0.28	0.14	-0.31	-0.29	0.34
Rating Of Preferred Study Time Setting	-0.40	-0.11	0.36	-0.02	0.35	-0.04	0.26	-0.32
Rating Of Manually Adding Unavailable Period	0.13	0.08	-0.35	-0.07	0.29	0.13	-0.34	-0.45
Rating Of Calendar Integration	0.34	0.01	-0.01	0.07	-0.10	-0.07	0.12	-0.17
Satisfaction With Functionality For Showing Unavailable Time	-0.26	0.18	0.27	0.05	-0.35	0.43	-0.19	0.25
Unnecessary Time Spent Navigating	-0.04	0.00	-0.06	0.43	-0.22	0.21	0.00	-0.21
Time Spent Editing Plan	-0.48	0.43	0.01	-0.24	0.13	-0.16	-0.22	0.21
Easiness Of Editing	-0.01	0.40	-0.07	0.30	0.54	-0.07	-0.10	0.11
Satisfaction With Functionality For Editing	-0.14	0.23	-0.22	-0.47	-0.39	-0.17	0.07	-0.18
Rating Of Self Adjusting Behaviour	-0.07	-0.09	0.20	0.31	-0.14	-0.38	0.16	0.19
Initial Satisfaction With Generated Plan	-0.08	0.13	-0.39	0.39	-0.22	-0.16	-0.31	0.15
Need For Monitoring Of Time Spent	-0.49	-0.22	-0.41	0.25	0.03	-0.10	0.34	-0.02
Need For More Variation	-0.04	-0.12	0.45	0.16	0.02	-0.03	-0.46	-0.14
Could Imagine Using The Tool In The Future	0.36	0.41	0.13	0.05	0.06	-0.26	0.30	0.24
Could Imagine Using The Tool For General Planning	-0.01	0.10	-0.08	-0.13	0.14	0.52	0.28	0.23

Table A.7: Eigenvectors 9-16 from Principal Component Analysis

APPENDIX B

Code Samples

B.1 R Code for Analyzing the Results from Survey on Study Preferences

Calculating Summations of Categorical Data, and Testing for Correlations and Normality

```
library(ggplot2)
library(wesanderson
library(nortest)
library(car)
library(stringr)
library(psych)

# Define significance level
significance = .05

# Read data
answers <- read.csv("/cloud/project/study_preferences_survey(66).csv", header = TRUE)

# "Nice-ify" column names
library(stringr)
colnames(answers) <- sapply(
  colnames(answers),
  function(x) str_to_title(str_replace_all(x, "_", " "))
)

# Plot Sum of Categorical Data on Calendars
sumdata <- data.frame(
  colSums(
    answers[, c("Google Calendar",
                "I Cloud",
                "Notion",
                "Physical Calendar Or Planner",
                "Fast Mail Calendar",
                "Huawei Calendar",
                "Nextcloud",
                "Notability",
                "Microsoft Excel")]
  )
)
colnames(sumdata) <- "Number of Students Using Types of Calendars"
sumdata$"Type of Calendar" <- row.names(sumdata)
ggplot(
  data=sumdata,
  aes(
    x = reorder(‘Type of Calendar‘, -‘Number of Students Using Types of Calendars‘),
    y=‘Number of Students Using Types of Calendars‘,
    fill=‘Type of Calendar‘
  ),
) + geom_bar(colour="black", stat="identity") +
  theme(axis.text.x=element_blank(), axis.title.x = element_blank()) +
  scale_y_continuous(breaks=c(1,2,3,4,5,10,15,20,25,30,35,40,45,50))+
  scale_fill_manual(values = c(
    wes_palette(n=5, name="Darjeeling1"),
    wes_palette(n=4, name="Darjeeling2"))
  )

```

```r
# Before statistical testing designed for continuous variables,
# drop variables which are either categorical or were synthetically derived
dropCategoricalVariables <- function(answers) {
  answers[sapply(
    c(
      "actual_study_time_per_day",
      "study_session_length_stdv",
      "break_length_stdv",
      "gender",
      "google_calendar",
      "i_cloud",
      "notion",
      "physical_calendar_or_planner",
      "fast_mail_calendar",
      "huawei_calendar",
      "nextcloud",
      "notability",
      "microsoft_excel"
    ),
    function(x) str_to_title(str_replace_all(x, "_", " "))
  )] <- list(NULL)
  return(answers)
}
numericalAnswers <- dropCategoricalVariables(answers)

i <- 1
while(i <= ncol(numericalAnswers)) {
  # Get variable to test on (name and data)
  column.data <- numericalAnswers[, i]
  column.name <- colnames(numericalAnswers)[i]

  column.plot <- ggplot(numericalAnswers, aes(x = column.data)) +
    geom_bar(fill = "white", color = "blue") +
    geom_vline(aes(xintercept=mean(column.data)), color="blue", linetype="dashed", size=1) +
    labs(y = "Count", x = column.name)
  plot(column.plot)

  # Get statistical description of column data using 'psych'
  column.description <- psych::describe(column.data)

  qqPlotted <- FALSE

  # Perform Shapiro-Wilks test for normality
  shapiroWTest <- shapiro.test(column.data)
  if(shapiroWTest$p.value >= significance) {
    if(!qqPlotted) qqPlot(column.data, main = column.name, ylab="rating")
    qqPlotted <- TRUE
    print(
      paste(
        "Results of the Shapiro-Wilks test indicated that there is a non-significant difference from the normal
              distribution",
        paste("p =", substring(toString(round(shapiroWTest$p.value,3)), 2))
      )
    )
  }

  # Perform Shapiro-Francia test for normality
  shapiroFTest <- sf.test(column.data)
  if(shapiroFTest$p.value >= significance) {
    if(!qqPlotted) qqPlot(column.data, main = column.name, ylab="rating")
    qqPlotted <- TRUE
    print(
      paste(
        "Results of the Shapiro-Francia test indicated that there is a non-significant difference from the normal
              distribution",
        paste("p =", substring(toString(round(shapiroFTest$p.value,3)), 2))
      )
    )
  }

  # Perform Pearson Chi-square test for normality
  pearsonTest <- pearson.test(column.data)
  if(pearsonTest$p.value >= significance) {
    if(!qqPlotted) qqPlot(column.data, main = column.name)
    qqPlotted <- TRUE
    print(
      paste(
        "Results of the Pearson chi-square test indicated that there is a non-significant difference from the
              normal distribution",
        paste("p =", substring(toString(round(pearsonTest$p.value,3)), 2))
      )
    )
  }

```

```
  i <- i + 1
}

# Calculate and plot correlation results
answersForCorrTest <- dropCategoricalVariables(answers)

library("Hmisc")
library(corrplot)
correlations <- rcorr(as.matrix(scale(answersForCorrTest)), type = "spearman")
corrplot(correlations$r,
         type="upper",
         order="hclust",
         p.mat = correlations$P,
         sig.level = 0.05,
         insig = "blank"
)
```

Clustering using 2-Means Clustering, and Plotting Data in the span of PC1 and PC2, and the span of PC1, PC2 and PC3

```
library(ggpubr)
library(factoextra)
library(wesanderson)
library(scatterplot3d)

numericalAnswers <- dropCategoricalVariables(answers)

# Compute k-means
numberOfClusters <- 2
res.km <- kmeans(scale(numericalAnswers), numberOfClusters, nstart = 500)
# K-means clusters showing the group of each individuals
res.km$cluster

# Dimension reduction using PCA
res.pca <- prcomp(numericalAnswers, scale = TRUE)
# Coordinates of individuals
ind.coord <- as.data.frame(get_pca_ind(res.pca)$coord)
# Add clusters obtained using the K-means algorithm
ind.coord$cluster <- factor(res.km$cluster)
# Add Degree of Planning groups from the original data set
ind.coord$‘Degree Of Planning‘ <- factor(numericalAnswers$‘Degree Of Planning‘)
# Data inspection
head(ind.coord)

# Percentage of variance explained by dimensions
eigenvalue <- round(get_eigenvalue(res.pca), 1)
variance.percent <- eigenvalue$variance.percent
head(eigenvalue)

ggscatter(
  ind.coord,
  x = "Dim.1",
  y = "Dim.2",
  color = "cluster",
  palette = wes_palette(n=numberOfClusters, name="Darjeeling1"),
  ellipse = TRUE,
  ellipse.type = "convex",
  size = 1.5,
  legend = "right",
  ggtheme = theme_bw(),
  xlab = paste0("Dim 1 (", variance.percent[1], "% )"),
  ylab = paste0("Dim 2 (", variance.percent[2], "% )")
) +
  stat_mean(aes(color = cluster), size = 4)

library(plotly)
library(dplyr)
fig <- plot_ly(ind.coord,
               x = ~Dim.1,
               y = ~Dim.2,
               z = ~Dim.3,
               color = ~cluster,
               colors = c("#FF0000", "#00A08A")
               ) %>%
  add_markers(size = 12)

fig <- fig %>%
  layout(title = "2-Means Clustering in PC1xPC2xPC3",
         scene = list(bgcolor = "#ffffff",
                      xaxis=list(title = paste0("Dim 1 (",variance.percent[1],"% )")),
```

```
                      yaxis=list(title = paste0("Dim 2 (",variance.percent[2],"% )")),
                      zaxis=list(title = paste0("Dim 3 (",variance.percent[3],"% )"))))
fig
```

B.2 R Code for Analyzing Data from the Quantitative Evaluation

Code Which Was Run before Each Test

```
# Define significance level
significance = .05

# Read data
answers <- read.csv("/cloud/project/beta(17).csv", header = TRUE)

# "Nice-ify" column names
library(stringr)
colnames(answers) <- sapply(
  colnames(answers),
  function(x) str_to_title(str_replace_all(x, "_", " "))
)

# Before statistical testing designed for continuous variables,
# drop variables which are either non-numerical or were synthetically derived
dropCategoricalVariables <- function(answers) {
  # drop non-likert type variables
  answers[sapply(
    c(
      "tool_is_better_suited_to_personal_needs",
      "tool_is_more_user_friendly",
      "the_time_to_get_a_satisfying_plan_was_shorter",
      "I_had_more_control_over_the_plan_I_was_presented_with",
      "even_if_I_did_not_exactly_understand_how_the_plan_adjusted_itself",
      "only_if_I_understood_how_the_plan_adjusted_itself",
      "I_do_not_wish_for_my_plan_to_adjust_itself_after_I_edit_it",
      "manual_upload_and_download_of_ics_files",
      "manually_pressing_a_sync_with_calendar_button",
      "automatically_keep_calendar_and_plan_synced"
    ),
    function(x) str_to_title(str_replace_all(x, "_", " "))
  )] <- list(NULL)
  return(answers)
}
```

Plotting Barplots and Testing for Normality

```
# Repeat normality tests for all continuous variables
library(car)        # For qq-plots
library(nortest)    # For lilliefors normality test
library(ggplot2)    # For plotting graphs

answersForNorTest <- dropCategoricalVariables(answers)

i <- 1
while(i <= ncol(answersForNorTest)) {
  # Get variable to test on (name and data)
  column.data <- answersForNorTest[, i]
  column.name <- colnames(answersForNorTest)[i]

  # Plot barplot of column data
  column.plot <- ggplot(answersForNorTest, aes(x = column.data)) +
    geom_bar(fill = "white", color = "red") +
    geom_vline(aes(xintercept=mean(column.data)), color="blue", linetype="dashed", size=1) +
    labs(y = "Count", x = column.name)
  plot(column.plot)

  qqPlotted <- FALSE

  # Perform Lilliefors test for normality
  lilliefors <- lillie.test(column.data)
  if(lilliefors$p.value >= significance) {
    qqPlot(column.data, main = column.name, ylab="rating")
    qqPlotted <- TRUE
    print(
      paste(
        "Results of the Lilliefors test indicated that there is a non-significant difference from the normal distribution",
        paste("D(", length(column.data), ") =", substring(toString(round(lilliefors$statistic, 3)), 2)),
```

```
        paste("p =", substring(toString(round(lilliefors$p.value,3)), 2))
      )
    )
  }

  i <- i + 1
}
```

Plotting of Likert Variables

```
# Drop non-likert type variables
answersForLikert <- dropCategoricalVariables(answers)

# Define Likert levels
likertLevels <- c('Strongly Disagree', 'Disagree', 'Neutral', 'Agree', 'Strongly Agree')

# Apply Likert levels
answersForLikert <- lapply(answersForLikert, function(x) factor(x, labels = likertLevels, levels = 1:5))

# Plot
library(likert)
plot(likert(as.data.frame(answersForLikert)))
```

Sign and Wilcoxon Sign-Rank Tests

```
answersForWilcoxAndSign <- dropCategoricalVariables(answers)

# Create an empty data frame for test results of Wilcoxon and Sign tests
wilcoxonSignResults <- read.csv(text="Sign Test Result,Sign Test p-value,Wilcoxon Result, Wilcoxon p-value",
                                colClasses = c("double", "double", "double", "double"))

# Use Sign test from BSDA
library(BSDA)

# Repeat One-sided Wilcoxon and Sign tests for all continuous variables
i <- 1
while(i <= ncol(answersForWilcoxAndSign)) {
  column.data <- answersForWilcoxAndSign[, i]
  column.name <- colnames(answersForWilcoxAndSign)[i]

  # Booleans to record whether a significant sign or wilcox result has been found
  significantWilcoxonTest <- FALSE
  significantSignTest <- FALSE

  # Column to record median and p-value from the two tests
  column.tests <- c(0, 1, 0, 1)

  # Perform one-tail Wilcoxon and Sign for 4 hypothesized medians in decreasing order
  for(hypothesized_median in 4:1) {
    # Perform one-tail one-way Wilcoxon signed-rank test greater than hypothesized median
    if(!significantWilcoxonTest) {
      wilcoxon <- wilcox.test(column.data,
                              correct = TRUE,
                              mu = hypothesized_median,
                              alternative = "greater")
      if(wilcoxon$p.value < significance) {
        significantWilcoxonTest <- TRUE
        column.tests[4] <- toString(round(wilcoxon$p.value, 8))
        column.tests[3] <- hypothesized_median
      }
    }

    # Perform one-tail one-way Sign test greater than hypothesized median
    if(!significantSignTest) {
      signTest <- SIGN.test(column.data,
                            md = hypothesized_median,
                            alternative = "greater")
      if(signTest$p.value < significance){
        significantSignTest <- TRUE
        column.tests[2] <- toString(round(signTest$p.value, 8))
        column.tests[1] <- hypothesized_median
      }
    }
  }

  # If any significant results were made, add results to table
  if(significantWilcoxonTest | significantSignTest) {
    wilcoxonSignResults[nrow(wilcoxonSignResults) + 1, ] <- column.tests
    row.names(wilcoxonSignResults)[nrow(wilcoxonSignResults)] <- column.name
```

```
  }

  i <- i + 1
}

# "Nice"-ify the names of variables with significant names
colnames(wilcoxonSignResults) <- sapply(
  colnames(wilcoxonSignResults),
  function(x) str_to_title(str_replace_all(x, "\\.", " "))
)
```

Calculation of Correlation Matrix

```
answersForCorrTest <- dropCategoricalVariables(answers)

library("Hmisc")
library(corrplot)
correlations <- rcorr(as.matrix(answersForCorrTest), type = "spearman")
corrplot(correlations$r,
         type="upper",
         order="hclust",
         p.mat = correlations$P,
         sig.level = 0.05,
         insig = "blank"
)
```

Plot Summaries of Categorical Data

```
library(ggplot2)        # For plotting
library(wesanderson)    # For nice plot palettes

# Plot Sum of Categorical Data on Calendar Integration Preference
sumdata <- data.frame(
  colSums(
    answers[ , c(
      "Automatically Keep Calendar And Plan Synced",
      "Manually Pressing A Sync With Calendar Button",
      "Manual Upload And Download Of Ics Files"
    )]
  )
)
colnames(sumdata) <- "Number of Students Preferring Specified Method of Calendar Integration"
sumdata$"Method of Calendar Integration" <- row.names(sumdata)
ggplot(
  data=sumdata,
  aes(
    x = `Method of Calendar Integration`,
    y=`Number of Students Preferring Specified Method of Calendar Integration`,
    fill=`Method of Calendar Integration`
  ),
) + geom_bar(colour="black", stat="identity") +
  theme(axis.text.x=element_blank(), axis.title.x = element_blank()) +
  scale_y_continuous(breaks=c(1,2,3,4,5,6,7,8,9,10))+
  scale_fill_manual(values=wes_palette(n=3, name="Darjeeling1"))

# Plot Sum of Categorical Data on
# "I would like to use a similar tool in the future, but only if ..."
sumdata <- data.frame(
  colSums(
    answers[ , c(
      "Tool Is Better Suited To Personal Needs",
      "Tool Is More User Friendly",
      "I Had More Control Over The Plan I Was Presented With",
      "The Time To Get A Satisfying Plan Was Shorter"
    )]
  )
)
colnames(sumdata) <- "Number of Students with Specified Requirements for Similar Planning Tools"
sumdata$"I would like to use a similar tool in the future, but only if ..." <- row.names(sumdata)
ggplot(
  data=sumdata,
  aes(
    x = `I would like to use a similar tool in the future, but only if ...`,
    y=`Number of Students with Specified Requirements for Similar Planning Tools`,
    fill=`I would like to use a similar tool in the future, but only if ...`
  ),
) + geom_bar(colour="black", stat="identity") +
  theme(axis.text.x=element_blank(), axis.title.x = element_blank()) +
  scale_y_continuous(breaks=c(1,2,3,4,5,6,7,8,9,10, 11, 12)) +
```

```
  scale_fill_manual(values=wes_palette(n=4, name="Chevalier1"))

# Plot Sum of Categorical Data on Plan Adjustment Preference
sumdata <- data.frame(
  colSums(
    answers[ , c(
      "Even If I Did Not Exactly Understand How The Plan Adjusted Itself",
      "Only If I Understood How The Plan Adjusted Itself",
      "I Do Not Wish For My Plan To Adjust Itself After I Edit It"
    )]
  )
)
colnames(sumdata) <- "Number of Students with Specified Preference for Post-Edit Plan Adjustment"
sumdata$"I would like for the plan to automatically adjust itself after I edit it ..." <- row.names(sumdata)
ggplot(
  data=sumdata,
  aes(
    x = `I would like for the plan to automatically adjust itself after I edit it ...`,
    y=`Number of Students with Specified Preference for Post-Edit Plan Adjustment`,
    fill=`I would like for the plan to automatically adjust itself after I edit it ...`
  ),
) + geom_bar(colour="black", stat="identity") +
  theme(axis.text.x=element_blank(), axis.title.x = element_blank()) +
  scale_y_continuous(breaks=c(1,2,3,4,5,6,7,8,9,10, 11, 12)) +
  scale_fill_manual(values=wes_palette(n=3, name="FantasticFox1"))
```

Principal Component Analysis

```
answersForPCA <- dropCategoricalVariables(answers)

# Create categories of data
answersAndGroupings <- answers

answersAndGroupings$`Could Imagine Using The Tool In The Future` <-
  factor(as.factor(answers$`Could Imagine Using The Tool In The Future` > 3),
         labels = c("No", "Yes"))

answersAndGroupings$`Could Imagine Using The Tool For General Planning` <-
  factor(as.factor(answers$`Could Imagine Using The Tool For General Planning` > 3),
         labels = c("No", "Yes"))

answersAndGroupings$`Could Imagine Using The Tool In The Future And For General Planning` <-
  factor(as.factor(answers$`Could Imagine Using The Tool In The Future` > 3 &
                   answers$`Could Imagine Using The Tool For General Planning` > 3),
         labels = c("No", "Yes"))

# Calculate PCA results (with normalized variances)
library("factoextra")
pca <- prcomp(answersForPCA, scale = TRUE)

# Plot explanation strength of principal components
fviz_eig(pca)

# Plot contributions
fviz_pca_var(
  pca,
  axes = c(1, 2),
  col.var = "contrib", # Color by contributions to the PC
  gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
  repel = TRUE     # Avoid text overlapping
)

## Plot bi-plot of PCA1xPCA2 with groupings
fviz_pca_biplot(
  pca,
  axes = c(1, 2),
  col.ind = answersAndGroupings$`Could Imagine Using The Tool In The Future And For General Planning`, # color by
      groups
  addEllipses = TRUE, # Concentration ellipses
  ellipse.type = "confidence",
  legend.title = "Could imagine using\n a similar tool in the future\nand for general planning",
  repel = TRUE,
)

## Plot groupings on PCA1xPCA2
fviz_pca_ind(
  pca,
  axes = c(1, 2),
  col.ind = answersAndGroupings$`Could Imagine Using The Tool In The Future And For General Planning`, # color by
      groups
  addEllipses = TRUE, # Concentration ellipses
```

```
  ellipse.type = "confidence",
  legend.title = "Could imagine using\na similar tool for general planning",
  repel = TRUE
)
```

APPENDIX C

Miscellaneous

C.1 An excerpt from Google's brand verification process

1. Ensure your app complies with the Google APIs Terms of Service and Google's API Services User Data Policy.
2. Confirm your app doesn't fall under any of the use cases listed in the Exceptions to verification requirements.
3. If you use Google Sign-In Scopes in your app, ensure that your app complies with the branding guidelines.
4. Verify ownership of your project's authorized domains using the Search Console. Use an account that is either a Project Owner or a Project Editor of your Cloud Console project.
5. Make sure all branding information on the OAuth consent screen, such as the project name shown to users, support email, homepage URL, privacy policy URL, and so on, accurately represents the app's identity.
    - Make sure that your homepage meets the following requirements:
        - Your homepage must be publicly accessible, and not behind a sign-in page.
        - Your homepage must make clear its relevance to the app you're verifying.
        - Your homepage must be accurate, inclusive, and easily accessible to all users.
        - Links to the Google Play Store or Facebook are not considered valid application homepages.
    - Make sure that your app's Privacy Policy meets the following requirements:
        - The Privacy Policy must be visible to users, hosted within the domain of your website, and linked from the OAuth consent screen on the Google API Console.

- The Privacy Policy must disclose the manner in which your application accesses, uses, stores, or shares Google user data. Your use of Google user data must be limited to the practices disclosed in your published Privacy Policy.

6. Prepare a detailed justification for each requested scope as well as an explanation for why a narrower scope wouldn't be sufficient. For example: *My app will use https://www.googleapis.com/auth/calendar to show a user's Google calendar data on the scheduling screen of my app, so that users can manage their schedules through my app and sync the changes with their Google calendar.* Your requested scope must be as granular as possible (if your requested scope goes beyond the usage needed, then we will either reject your request or suggest a more applicable scope).

7. Prepare a video that fully demonstrates the OAuth grant process by users and shows, in detail, the usage of sensitive scopes in the app.

   - Show the OAuth grant process that users will experience, in English (the consent flow, and, if you use Google Sign-in, the sign-in flow).
   - Show that the OAuth Consent Screen correctly displays the App Name.
   - Show that the URL bar of the OAuth Consent Screen correctly includes your app's Client ID.
   - Show how the data will be used by demonstrating the functionality enabled by each sensitive and restricted scope you request.
   - Upload the video to YouTube. You'll need to provide a link to the video as part of the verification process. Let us know if your app requires registration or features a local login. If any of your OAuth clients are not ready for production, we suggest you delete or remove them from the project requesting verification. You can do this in the Google Cloud Console.

C.2 The Full Questions from the Quantitative Survey

1. Do you normally use tools such as this, or have you previously used similar tools to plan your work/studies? If so, which ones?
2. On a scale from 1 to 5, were 1 means "strongly disagree" and 5 means "strongly agree", what scores would you give the following statements about the tool:
   - I liked that the tool made a distinction between *study time* and *other time*.
   - I think it is important that the tool knows when I am available to study.
   - I think the settings for "preferred study time" worked well to show my available time.
   - I think that manually adding "unavailable periods" worked well for showing my available time.
   - I think the way in which the *calendar functionality* was integrated worked well to show my available time.
3. How "tightly integrated" would you prefer for such a calendar functionality to be in this type of tool?
   - As it is now: Let me manually upload unavailable periods, as well as exporting the my personal plan, by the use of ICS-files.
   - It should let me synchronize my calendar and my personal plan by the use of a "synchronize calendar" button
   - It should continuously keep my calendar and personal plan up to date with each other (completely automatic, nothing done manually).
   - It should be integrated in another way than the alternatives mentioned above (if so, please specify).
4. On a scale from 1 to 5, were 1 means "strongly disagree" and 5 means "strongly agree", what scores would you give the following statements about the tool:
   - I feel there are a satisfying amount of ways to show when I am available to study (in the tool in its current state).
   - I used an unnecessary amount of time moving between the different views of the tool (for instance between the overview of tasks and the view of my personal plan).
   - I spent lots of time adjusting/making changes to my personal plan after it had been generated.
   - It was easy to make the changes which I wanted to my personal plan.
   - I feel there are a satisfying amount of ways to to change my personal plan after it was generated.
5. Are there any ways to make changes to your personal plan that you feel are missing at the moment? (If so, please specify)
6. On a scale from 1 to 5, were 1 means "strongly disagree" and 5 means "strongly agree", what scores would you give the following statements about the tool:

- I liked how elements in my personal plan automatically adjusted themselves after I had manually made changes.

7. I would like if more elements in the schedule were adjusted automatically ...
   - ... but only if I understood the rules which my personal plan followed when adjusting itself.
   - ... even if I did not exactly understand how my personal plan adjusted itself.
   - I do not wish for my personal plan to automatically adjust itself after I make changes.
8. On a scale from 1 to 5, were 1 means "strongly disagree" and 5 means "strongly agree", what scores would you give the following statements about the tool:
   - I feel that the plan which was generated fit me and my needs.
   - I felt I was missing an overview of how much time I had already spent on my tasks.
   - I wish the tool would plan for more varied workdays.
9. Do you have any thoughts on what it would mean to have a "varied" study plan? (If so, please specify)
10. Is there anything about the planning of study days which the tool did not account for, but that you wish it accounted for while generating schedules? (If so, please specify)
11. On a scale from 1 to 5, were 1 means "strongly disagree" and 5 means "strongly agree", what scores would you give the following statements about the tool:
    - I could imagine myself continuing to use this tool to structure my studies.
    - I could imagine myself using this tool to structure my tasks in general (not just study related tasks).
12. I could imagine myself continuing to use a similar tool ...
    - but only if it was better tailored for my needs.
    - but only if it was generally more user friendly.
    - but only if it took less time to get started and to get a usable plan.
    - but only if I had more control over the plan I ended up with.
    - but only if it fulfilled other requirements I have (if so, please specify).
13. Is there anything else you want to say about the tool? (It could be anything.)

Axel Kjønsberg og Erlend Ydse

Supporting Time Management among Students with the Aid of Digital Tools

NTNU
Norwegian University of
Science and Technology