Linköping University | Department of Computer and Information Science
Master's thesis, 30 ECTS | Datateknik
2023 | LIU-IDA/LITH-EX-A--23/082--SE

Implementation and evaluation of a web-based learning management system

Implementation och utvärdering av en webb-baserad lärplattform

Mathilde Hennings
Emil Nilsson

Supervisor : Aseel Berglund
Examiner : Erik Berglund

Linköpings universitet
SE–581 83 Linköping
+46 13 28 10 00 , www.liu.se

Upphovsrätt

Copyright

Abstract

Modern education is more and more dependent on technical solutions to assist student learning. Systems specifically designed for this purpose are called learning management systems and provide the link between students and teachers, in the sense that a significant part of their communication is handled within the system. With a learning management system as a core part of a course, the demands on the usefulness and usability of the system are high. The usefulness of an in-house developed system was evaluated in a real world scenario from a teacher's perspective. The evaluation concludes that the implemented changes increased the usefulness of the system and simplified the work for the teachers.

Acknowledgments

We would like to express our gratitude to our supervisor Aseel Berglund and our examiner Erik Berglund who have assisted us throughout the project. In addition we would like to give a special thanks to research assistant Izabella Jedel who has provided us with significant help in conducting this study.

Finally we would like to thank the course assistants in the course TDDI83 (Professionalism for Engineers) for participating in the study and reporting any issues which arose while using the system in the course.

Contents

List of Figures

All names in the figures are made-up.

List of Tables

1 Introduction

Learning management systems (LMSs) are a relatively new area of research, and therefore it is not heavily studied. This thesis paper presents the implementation of an administrative tool for an LMS and evaluates its usefulness and usability to course assistants. The following sections present the motivation behind the research topic, the aim of this thesis, the research question as well as relevant delimitations.

1.1 Motivation

As the world develops and the importance of technology increases, demands are placed on education to adapt accordingly. Technology within the educational system has been a topic of interest for many years, and has been shown to improve both teaching and learning [1]. One common way to introduce technology to education is through LMSs, which moves learning to a digital environment. An LMS can, for example, include presentation of learning material, quizzes and student submissions. The use of LMSs also enables distance studies, which has become more popular in recent years, partly as a consequence of the Covid-19 pandemic.

LMSs are used to automate aspects of course administration, such as tracking progress, reporting results and handling learning materials [2]. This makes the work for course assistants more efficient, and resources can be prioritized more effectively. Tasks such as organizing submissions, giving feedback and getting an overview of student progression are facilitated with a digital system. It is of great importance to evaluate the system's usefulness in order for it to improve the education, instead of creating unnecessary obstacles and frustration. There are some evaluations on usability of LMSs from the perspective of the students, but they are rarely discussed from the perspective of the teachers or administrators.

1.2 Aim

The aim of this thesis project is to investigate how an administration tool for an LMS can be designed to give high usefulness and improve the ease of use for its users. To be able to investigate this a web application for the course *Professionalism for Engineers* at Linköping University known as Eprof was used, to which new features were implemented, as well as improvements to existing ones. These changes were evaluated to assess the usefulness and

ease of use of the platform from the perspective of its users, specifically the course assistants. In addition, the project aims to give insight into how learning analytics can be implemented in an LMS and its affects on the users.

1.3 Research question

To reach the aim of the project the following question will be answered in this thesis report:

- How can an administration tool for an LMS be implemented to give high *usefulness* and *usability* for the users?

There is no standard definition for usefulness in the context of human-computer interaction (HCI). Although this is the case there have been attempts to create a common definition. MacDonald and Atwood [3] propose a definition of usefulness to be *"the extent to which a system's functions allow users to complete a set of tasks and fulfill specific goals in a particular context of use."*. Usefulness is thereby closely related to usability, which according to the ISO 9241-11 standard definition [4], is defined as *"the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use"*.

1.4 Delimitations

This thesis project focuses on the administration tool of the LMS. This means that the student frontend will not be evaluated. The user tests are only conducted with users of the administration view of the system, more specifically the course assistants. The evaluation participants only include people who have no prior experience using the system, since no course assistance had previous experience of using the system. Furthermore, the research only evaluates the platform using one specified course. This means that the usage of this platform in the context of another course with a different structure is not investigated or evaluated.

2 Background

Within this chapter details about the context to this thesis are described. This includes describing the Professionalism for Engineers course, existing software systems at Linköping University, the Eprof e-learning platform, as well as tools and frameworks used in the implementation of the platform.

2.1 Professionalism for Engineers

Professionalism for Engineers is a collection of courses at Linköping University aimed to develop computer science students' soft skills. The first edition of the course was developed by Berglund and Heintz [5], who argue that the course has a positive effect on students and improves their view on the importance of soft skills. This edition of the course is divided into twelve different parts, which are part of four main categories: personal effectiveness, personal development, social competence and the engineering profession. Another smaller course edition for other education programs was introduced later, it is described by Berglund [6] to have been successful for opening the course to more students. This course has the course code TDDI83[1] and is in focus during this study.

The TDDI83 course edition is categorized into three main modules of similar structure, which are titled:

1. Personal management
2. Communication
3. Teamwork and engineering

The main modules consist of a lecture, reading material, a writing assignment and some supplementary assignments. They all end with a seminar where a small group of students discuss their experiences in the module. The lecture and the seminar are conducted in person face-to-face. The concept for the larger course edition is the same, although the modules differ.

Before the e-learning platform was introduced, these tasks were carried out using a combination of three separate platforms: the course webpage, Lisam and Webreg. As the

[1] https://studieinfo.liu.se/kurs/tddi83

course is given in different editions for different education programmes, the e-learning platform has only been introduced in the TDDI83 edition so far. Therefore, this thesis evaluates primarily how the system is perceived in that course edition. This course edition is faster paced than the other edition and thereby allows for quicker evaluation. During the Covid-19 pandemic the TDDI83 course was adapted to be run in a fully remote environment as described by Berglund [7]. Although it was successfully converted to a fully online course, most aspects which were changed have since been reverted, as there were benefits to the face-to-face version which could not be replicated in the online version.

Although the primary focus in this thesis is the TDDI83 course edition, the end goal is that the concepts described in this thesis should be adaptable to other editions of Professionalism for Engineers as well as other courses.

2.2 Platforms at Linköping University

Before the development of Eprof, the course TDDI83 used a course webpage, Lisam and Webreg. These platforms are also commonly used together in other courses at the Department of Computer and Information Science (IDA) at Linköping University.

2.2.1 Course webpage

For many courses at the university there exists a course webpage. This page includes information about the course which is relevant for the student, such as study material, assignments and lectures. There is no possibility of interaction between students and teachers. The course webpage does not require authentication and is therefore accessible by everyone; you do not need to be registered to the course or even be a student at the university to access it.

2.2.2 Lisam

The platform Lisam is a collaborative space containing files, videos, submissions and quizzes. It is an interactive website since it enables students to upload documents and take quizzes among other things. Course supervisors can also give feedback to submissions directly in Lisam. This platform is not open to everyone; authentication is needed as well as being registered to the course. Lisam is based on *Microsoft SharePoint* and is designed to be able to act as an all in one hub for courses.

2.2.3 Webreg

The department of Computer and Information Science (IDA) uses a platform called Webreg for registration to laborations, projects and seminars. The registration can be individual or by group. This platform is then used to give feedback to the students who have signed up, and for the students to monitor their results and grades.

2.3 Eprof

Eprof is the name of the web-application used in the thesis. It was first developed in 2021 when the first version was implemented as part of a thesis project at the university. Thereafter, several thesis project have been conducted, investigating areas like usability, responsiveness and gamification of the platform. The following list contains the complete collection thesis projects ordered by publication.

- *Enriching the user experience of e-learning platforms using responsive design: a case study* by Björn Möller Ehrnlund [8].

- *Design and Evaluation of an E-learning Platform to support Active Learning* by Emma Algotsson [9].
- *Implementation And Evaluation of Gamification Elements In An E-learning Platform For Engineering Education* by Erica Ahlqvist [10].
- *Implementing a gamified elearning platform to teach soft skills to bachelor students* by Jakob Sjöqvist and Oscar Sandberg [11].
- *Implementation and usability evaluation of a responsive e-learning administration tool* by Johannes Klasson and Christoffer Henriksson [12].

2.3.1 Eprof concepts

In previous thesis work on the Eprof platform much focus has been placed on the student frontend and improving the learning experience. This has resulted in gamification elements such as a point system for completing tasks and a progress bar to track the student's progress. The student view is shown in Figure 2.1 where a summary of the student's points and their course progress is displayed in the top right corner.

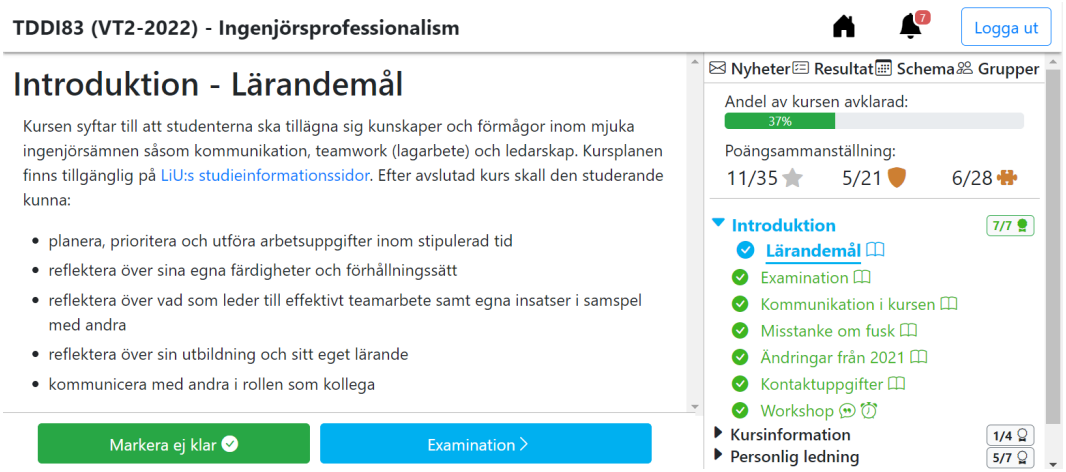

Figure 2.1: Eprof student view

Point system

There are three types of points: *course points*, *master points* and *order points*. These are earned through completing course activities. Course points are mandatory and are obtained when tasks and submissions are completed. When a task is completed with high quality it is rewarded with master points. Order points are given when administrative tasks are done correctly, e.g. submissions done in time and handed in through the Eprof platform. To pass the course, a certain distribution of points must be met. There is also a reward for students who collect more points than required in the course. These thresholds, as well as the maximum amount of points, are shown in Table 2.1. This point system has previously been evaluated from a student's perspective by Berglund and Jedel [13], who found that the point system positively improved the students' hedonic perception of the system.

Course structure

Eprof supports the building of courses according to a certain structure. Each course is divided into course parts which is a container listing course activities. Course activities contain the

tasks and instructions. The different types of course activities supported by the system are reading, writing submission, file submission, seminar, video and quiz. A course activity can either be gradable or not and this can be edited in the course activity settings.

Table 2.1: Points and thresholds in the TDDI83 course

	Course points	**Master points**	**Order points**
Pass the course	35	0	14
Pass the course with reward	35	18	17
Max points	35	21	21

2.3.2 Previous implementation of Eprof

Eprof is divided into three separate modules, a student frontend, a teacher frontend, and a backend which is shared by both frontend applications. These have been continuously developed throughout the previous thesis projects. When work on this thesis started in January 2023, there were several limitations of the system. One main limitation of the system was that due it only having been actively used once, there was no proper implementation of a system to rerun a course for a new year, for example multiple courses could not share the same course code. Furthermore, the teacher frontend did not provide the adequate functionality to do all the required tasks of running a course.

Student view

The student view was inherited from the state described by Sandberg and Sjöqvist [11]. It was developed to a state where the functionality was mostly finished and had been used in a previous instance of the course. The potential for improvement was mainly in small details. The student view is developed as a single page application using Angular. The user interface is largely based on the ng-bootstrap[2] component library.

Teacher view

The teacher view was a later addition to Eprof initially developed by Henriksson and Klasson [12]. This view was in a less mature state than the student view and while the functionality it had worked, many features that could be expected from what is in the student view were missing in the teacher view. Throughout the teacher view there were many bugs which were most likely a consequence of that the previous course instance data had not been built using the teacher view, leaving some functionality mostly untested. In the previous course iteration most of the course data was configured using a seeding script which bypassed the need of an administration view.

The teacher view is, just like the student view, developed as a single page application using Angular. The user interface is largely based on the Angular Material component library, which is further described in 2.4.5.

Shared backend

The backend is shared between the student and teacher frontends and has been developed continuously alongside them. It is written in the Python based Flask framework and connected to an SQLite database.

[2]https://ng-bootstrap.github.io

2.4 Tools and Frameworks

The Eprof platform had partly been implemented and therefore had a fixed set of tools and frameworks which it used. These tools and frameworks are further presented in this section.

2.4.1 Angular

Angular[3] is a frontend framework for building web applications with TypeScript and can be used to develop complex single page applications. In Angular everything is built using components which can be seen as modular building blocks for an application, allowing for reuse throughout the application.

2.4.2 TypeScript

TypeScript[4] is a superset of JavaScript allowing for statically typed programming at compile time, after which it can be executed as JavaScript.

2.4.3 Flask

Flask[5] is a popular backend web framework for writing web applications in Python. It is a barebone framework, but has a vast ecosystem of libraries specifically made to extend the functionality of Flask.

2.4.4 SQLite

SQLite[6] is a simple Database management system (DBMS) made to store databases in a single file. To execute SQL queries in Python programs the object relational mapper (ORM) SQLAlchemy can be used.

2.4.5 Material UI

Material UI[7] is a user interface component library providing ready made components designed to follow common principles of usability and accessability. It is developed according to the material design specification[8].

[3] https://angular.io/
[4] https://www.typescriptlang.org/
[5] https://flask.palletsprojects.com/
[6] https://www.sqlite.org/
[7] https://material.angular.io/
[8] https://m3.material.io/components

3 Theory

This chapter presents relevant theory and previous research within the topics of LMSs, learning analytics, usefulness and usability as well as technology acceptance.

3.1 Learning Management Systems

LMSs, or e-learning platforms, are described by Alias and Zainuddin as "a software application or web-based technology used to plan, implement, and assess a specific learning process" [14, p. 28]. This generally includes the possibility for a teacher to create and distribute information, observe the participation of students and assess their performance online. An LMS can be used for *blended learning*, where a traditional classroom experience is enhanced with online elements.

Having an LMS can allow for effective operation of a course, but the system itself will not inherently improve learning efficiency without being used efficiently. Chaw and Tang [15] have conducted a study examining whether the system quality, information quality, and service quality of an LMS affects the learning effectiveness. As the system can be seen as a mandatory part of course participation for a student, being satisfied with an LMS is not a prerequisite for them using it. However, Chaw and Tang conclude that an LMS of high quality can affect learning effectiveness positively.

An LMS has to be evaluated from two perspectives, the student's and the teacher's. According to Lonn and Teasley [16] students often believe efficiency is the most valuable benefit of IT systems for education, while instructors found communication to be the most valuable benefit. According to Castro-Schez et al. [17] LMSs can be used both to make a subject more interesting and to help students grasp a topic. At the same time, it allows a teacher to store and monitor each student's work throughout the course.

Adopting an e-learning platform can sometimes be restricted by the willingness of the staff to adopt a new platform that affects the way they work. Cifuentes et al. [18] describe factors that can improve the staff attitude of an e-learning platform. At first, a platform needs to provide a clear advantage which can grab their attention. When staff proceed to start using the system it is stated by Cifuentes et al. that naysayers can provide useful information about what needs to be improved, and should therefore be listened to. They further state that when the teachers experience student success from using the system they will become confident users.

Alias and Zainuddin [14] conducted a study which evaluated lecturers' attitude towards the use of an LMS at university level. They stated the importance of the lecturers acceptance towards and integration into the LMS in order for it to be useful. Even though the lecturers were mostly using traditional teaching techniques, they were open-minded and had a positive attitude towards the use of an LMS. The study also showed that the usage of an LMS lead to better organization of lecture material, time saving, more effective evaluation of students and better interaction between lectures and students by linking the LMS to the email system used by the university.

3.2 Learning analytics

The use of LMSs has provided the opportunity to collect large amounts of education related data [19]. This has resulted in several new research areas, including learning analytics [20]. Learning analytics is aimed to understand and optimize the learning experience for students through analysing LMS data. Some benefits to learning analytics, as mentioned by Mothukuri et al. [21], is the possibility to monitor student progress by visualizing data through the use of dashboards. Tissenbaum et al. [22] state that many systems display dashboards for students and teachers after the fact, and instead they investigated how progress can be displayed in a meaningful way in real-time.

A teachers' understanding of learning analytics can sometimes be an obstruction for the effectiveness of the analytics. According to van Leeuwen [23] the intervention which should be taken due to a learning analytic can be unclear, often depending on a teacher's level of experience. Therefore, learning analytics can be improved by suggestions of which actions should be taken as a consequence of a metric. There is also a possibility that the meaning of the metric itself can not be interpreted by the teacher.

3.2.1 Student outcome prediction

Learning analytics within higher education is a growing area of research ever since online learning has increased in use. As the use of e-learning grows, so does the difficulty for teachers to gain insight into the students' performance. To facilitate this, learning analytics can be applied together with predictive models to estimate student outcome, and especially identify at risk students, and through this information support student success. Smith et al. [24] suggest two methods to predict student output. The student output consisted of three warning levels meant to indicate the likelihood of a student passing the course. A few key factors were defined that affect the prediction of student success, such as grades, site engagement, login frequency, pace and a few non-LMS related factors. Another prediction model developed by Zacharis [25] shows that four main factors were highly effective in predicting student outcome in blended learning environments, these were: reading and posting messages, content creation contribution, quiz efforts and the number of files viewed.

3.2.2 Learning analytics and gamification

Gamification within the field of education relates to the use of game elements to increase student motivation [20]. These elements include levels, badges, progress bars, and many more. Even though there is much research showing positive results, it is according to Tenório et al. [26] important to enable monitoring and adaptation of game elements to obtain the positive effects.

Dichev et al. [27] developed and evaluated a gamified learning platform with learning analytics. They found that the platform motivated students and that teachers could track progress and notice students who were falling behind and might need help. The aim with implementing learning analytics in a gamified learning platform was to capture, analyze and

visualize learning data from student performance and behaviour to optimize the learning and teaching experience. The data analyzed was divided into three categories:

- **Action related**: data about actions performed by the students.
- **Content related**: data about content which the student interacted with or produced.
- **Result related**: data about learning or gamification outcomes of student activities.

By making data such as points, completed tasks, number of attempt and last login available to teachers, Dichev et al. [27] state it aids teachers in understanding students' interaction with the course activities.

3.3 Usefulness and usability

Usability is a central concept of human-computer interaction and is a measurement of effectiveness, efficiency and satisfaction of a system. This is described in ISO 9241-11 [4], which is an international standard that provides guidance on usability. The standard defines usability as the extent to which a product can be used by specific users to achieve their goals with efficiency, effectiveness, and satisfaction in a given context and emphasizes a user centered approach to design. Effectiveness focuses on the accuracy and completeness of users goal with using the system. Efficiency evaluates the speed and resources required to complete tasks. Satisfaction encompasses the subjective experience and overall user satisfaction with the system. ISO 9241-11 provides a framework for evaluating usability, promoting user-centered design, and guiding organizations in creating interactive systems that provide enhanced user experiences. The standard highlights the importance of involving users throughout the design process and conducting iterative usability evaluations.

Usefulness is a crucial factor in determining the success of a software system and plays a significant role in enhancing the user experience and the overall user satisfaction. According to MacDonald and Atwood [3] usefulness has largely been overshadowed by usability which has dominated the field of research. Furthermore, they claim that a system's "goodness" can be linked to usefulness, a system which is deemed useful is often correlated with being deemed good. MacDonald and Atwood further describe that usefulness is closely related to usability.

Grudin [28] argues that a system which is usable might not be useful, for example a system might be very easy to learn and to use, but if it serves no purpose to the user it is not useful, providing no utility. Thereby both the usability and utility have to be considered in parallel for a systems usefulness to be properly evaluated. Furthermore Johannessen and Hornbæk [29] describe how utility is seldom at focus when evaluating usefulness, instead only the usability is in focus. Thereby both usefulness and usability are important considerations in human-computer interaction. A system may be useful but lack usability if it is difficult to learn or inefficient to use. Conversely, a system may be highly usable but lack usefulness if it doesn't fulfill users' requirements. The end goal should be to design systems that are both useful and usable, providing value to users while being easy and efficient to use.

3.4 Technology Acceptance Model

Technology acceptance model (TAM) is an evaluation model used to measure the user acceptance of a system and was developed by Davis [30]. It was developed in 1989 and has since become a leading model to evaluate system acceptance, according to Maragunić et al. [31]. Davis further describes how the model measures two variables, *perceived usefulness* (PU) and *perceived ease of use* (PEOU), which is said to influence system use. The model consists of six questions for each of the two variables.

PU is defined as how useful the user deems the system to be, to which extent would the system improve their work performance. PEOU is related to how easy to use the user finds the system. A user can find a system easy to use but not perceive it as helpful in performing better at their job. Likewise, a system could enhance the user's job performance but simultaneously be difficult to use.

3.4.1 Validity

To analyse the validity of TAM, King and He [32] conducted a statistical meta-analysis, where 88 published articles were studied. The findings were that TAM is a "powerful and robust predictive model", and that "the effect of ease of use on behavioral intention is primary through usefulness".

3.4.2 Technology acceptance in LMSs

TAM was applied in a study conducted by Masrom [33] where 122 students answered the questionnaire to evaluate e-learning platforms. The results showed that the perception of a system's usefulness is more important than the acceptance of it when it comes to the intention to use a system. Therefore, they claim that it is more important to present the advantages of using a system in regards to effectiveness and efficiency for student learning than it is to teach how to use the system. They also found that TAM is useful to predict the acceptance of a system, but not for getting descriptive data.

Mohamad et al. [34] conducted a study where TAM was used to evaluate e-learning. The study indicated that PU and PEOU made a significant impact on the technology acceptance of e-learning. They validate that TAM gives a good representation of technology acceptance within the context of higher education.

4 Method

The method of the thesis project is divided into three parts: pre-study, implementation and evaluation, which are described in this chapter.

4.1 Pre-study

The pre-study included exploration of the Eprof platform and its features to investigate what could be improved. This was done by first discussing the platform with people who were involved in the development of it last year, who possessed knowledge from user testing of the platform, as well as the examiner of the course. With regard to what was discussed, tasks of both course assistants and administrators were mapped out and explored through testing, simulating system usage from their perspectives. After the exploration of the platform was conducted, a list of three main problems was identified. Solutions to these three problems were outlined and discussed with the course examiner before continuing with the implementation phase.

4.2 Implementation

The e-learning platform Eprof was already under development and therefore the architecture, tools and frameworks were already decided and implemented. The project consists of two frontend applications, one backend and a database to handle data storage. The platform was implemented using Angular, Flask and SQLite, as well as many other less significant dependencies.

The implementation phase was divided into two parts. The first part was to adapt the system to handle new course starts and fix bugs while keeping the main user experience intact. The second part consisted of implementing a new view in the teacher frontend with concepts based on the results from the pre-study.

During the implementation of the platform code quality was an important consideration. The platform was not built as a prototype, but rather as a system that was actively used throughout a course by both students and course assistants. It is also a system which hopefully will be used and maintained in the future. To improve and maintain code quality and maintainability as the code base inevitably grows, some measures were taken to improve the existing code standard to avoid bugs and accommodate future development.

As a first measure all the dependencies in all three modules were updated to newer versions and some dependencies which were deprecated were phased out.

Angular is developed specifically for usage with the TypeScript language and thereby has great support for it. Although this is the case, the "any" keyword was widely used throughout the code bases. This is likely to result in lower code quality as stated by Bogner and Merkel [35]. To address this flaw in both frontend modules the strictness of TypeScript usage was improved by reducing the usage of the "any" keyword and adding a lint rule for warning when the "any" type was used. In addition to this linting rule, several standard linting rules for Angular and TypeScript were added to inform users of bad practices in the code.

In the API endpoints between the two frontend applications and the backend, the initial version had many inconsistencies with how different objects were represented, the same database model had numerous of different representations in different API endpoints. To address this inconsistency the library Marshmallow[1] was added, which allowed the definition of schemas of how a database model should be represented in the API. It also allowed input data to be validated into the same schema. Since the standards of letter casing in Python and TypeScript differ, another aim was to unify in which letter casing API data was formatted. By using Marshmallow, an automatic transformation between snake case and camel case in the API could be achieved. Consistency in the API allowed for better integration with TypeScript in the two frontend applications.

4.3 User evaluation

User evaluation was performed twice, once before the new view was implemented and once after. The evaluations were made using a questionnaire consisting of a combination of questions aimed to collect user opinions and questions from TAM.

The people used for the evaluation were course assistants, or mentors, in the course TDDI83. In total, the course had six course assistants. They used the platform continuously during the two months the course was taught. The course assistants graded one course part before the new view was implemented, and one after it was implemented. The course parts consisted of the same types of course activities, both containing a number of file submissions, one written assignment and finally a seminar. Thereby, each evaluation was based on the same amount of interaction with the platform. The questionnaire was distributed when all the course assistants had graded their students' submissions, usually one to two weeks after the deadline of the assignment. The first submission was two and a half weeks into the course, and the second one just over two weeks after that. Five course assistants participated in the evaluation. Both the first and second data collection was conducted with the same course assistants, and they had no prior experience using the platform and were not anonymous.

4.3.1 Opinions on Eprof

The part of the questionnaire aimed to assess opinions of the Eprof platform consisted of a combination of open ended questions and questions answered on a Likert scale from one to seven. Each question answered on a scale was accompanied by a text field to motivate the answer.

The two iterations of the questionnaire consisted of mostly the same questions, with a few additions and deletions. In more detail, these questions and which questionnaire round they were included in are shown in Table 4.1. Each question marked with an "X" in the table entails its participation in the corresponding round.

[1] https://marshmallow.readthedocs.io

Table 4.1: Questionnaire for iteration one and two

Question	**Type of answer**	**Questionnaire round**	
		1	**2**
What is your opinion of the teacher view in Eprof?	Scale of very negative to very positive	X	X
What is your opinion of the changes made to Eprof?	Scale of very negative to very positive		X
What is your view on correcting students' tasks in Eprof?	Scale of very negative to very positive	X	X
What can be improved with the correction process in Eprof?	Text	X	
How easy or difficult is it to keep track of how the students are doing in the course with the help of Eprof?	Scale of very difficult to very easy	X	X
What can be improved in Eprof when it comes to seeing how the students are doing in the course?	Text	X	
How do you experience the visualization of students' progress and scores in Eprof?	Scale of very negative to very positive		X
Something else you want to add?	Text	X	X

Table 4.2: TAM questions

Question	**Category**
Using Eprof in my job as a mentor enables me to accomplish tasks more quickly.	PU
My interaction with Eprof is clear and understandable	PEOU
Using Eprof makes it easier for me to do my work as a mentor	PU
I find it easy to get Eprof to do what I want it to do	PEOU
I find Eprof useful in my work as a mentor	PU
Eprof helps me identify students who are behind in the course and who may need more support from me as a mentor	Solution objective
Using Eprof enhances my effectiveness in my work as a mentor	PU
Learning to operate Eprof is easy	PEOU
I find Eprof to be flexible to interact with	PEOU
It is easy to become skillful at using Eprof	PEOU
Using Eprof improves my job performance	PU
Using Eprof increases my productivity	PU
Eprof helps me get an overview of how the students are doing in the course	Solution objective
I find Eprof to be easy to use	PEOU

Table 4.3: TAM scale

Scale	Points
Strongly disagree	1
Moderately disagree	2
Somewhat disagree	3
Neutral (neither disagree nor agree)	4
Somewhat agree	5
Moderately agree	6
Strongly agree	7

4.3.2 Technology acceptance of Eprof

To test the technology acceptance of Eprof TAM was used. It included PU and PEOU from TAM, see section 3.4, which uses a seven step Likert scale, see Table 4.3.

In addition to the standard PU and PEOU questions in TAM, two new questions were added with the same Likert scale. The aim with the additional questions was to give insight regarding the platforms ability to support the course assistants in their work as mentors. The new questions include "Eprof helps me identify students who are behind in the course and who may need more support from me as a mentor" and "Eprof helps me get an overview of how the students are doing in the course". The additional questions and the standard PU and PEOU questions are presented in Table 4.2 in the order they appear in the questionnaire.

4.3.3 Thematic analysis

The questionnaire used to evaluate Eprof contained open-ended questions. To analyse the qualitative results from these and identify common patterns, a thematic analysis method was used. A thematic analysis also made it possible to draw conclusions from qualitative data. In the thesis project a thematic analysis method as presented by Howitt [36] was used. This method is based on the one created by Braun and Clark [37]. The method consists of the following six steps:

1. Data familiarisation
2. Initial coding generation
3. Search for themes based on initial coding
4. Review of themes
5. Theme definition and labelling
6. Report writing

Data familiarisation is a key step in thematic analysis to gain insight into the data collected. It can be done by transcribing interviews or reading through the collected data several times. Data familiarisation facilitate the researcher in finding patterns and themes that could be used in the analysis.

After data familiarisation the initial coding is generated. Coding is defined as a label to describe one or two lines of textual data [36]. The code should summarize the important points from that line of data. Braun and Clarke [37] state that coding generation can be performed with either a data-led or a theory-led approach. In this project a data-led approach is taken, which entails that the coding generation is led by the textual data instead of theories applied to the research.

The third step in thematic analysis is to search for themes based on the initial coding created in the previous step. Some codes are more connected to each other than others, and this step aims to find patterns amongst the codes.

Next, the themes are reviewed in relation to the original data. The data was previously arranged around the codings, but in this step it is to be organized into the different themes. At this stage, the themes do not have to be fully defined. Depending on the data distribution amongst the themes, some themes may not seem relevant anymore whilst others may need to be split into multiple separate themes.

In the fifth step, the themes are defined and labeled. In defining and labelling the themes it is important that they can be distinguished conceptually from each other. When adding clear definitions to the themes, it is very likely that all the previous steps in the model need to be reexamined.

The final step in the model described by Howitt [36] is to write a report on the results from the thematic analysis.

According to Howitt [36] the model does not necessarily need to be followed as a straight line. In actual use it is common to move between the steps as needed to improve the analysis.

5 Results

This chapter presents the results of the thesis divided into pre-study, implementation and evaluation.

5.1 Pre-study

During the pre-study the previously developed platform was tested from the perspective of the two types of users: course assistants and administrators. These results were combined with feedback from course assistants with previous experience of using Eprof, as well as discussions with the course examiner, and resulted in the following list of main problems:

- Cannot handle rerun of courses.
- Difficult to track student progress.
- Cluttered grading process.

As a result of the pre-study a decision was made to improve the grading process for the course assistants. To better display student progress in a way adapted to course assistants a new view was designed. The main focus of the new view was to contain everything relevant for a specific course assistant to grade students and display their progress in a clear way. What was to be included in the new view was evaluated through examining exactly which parts of the system would be relevant in the role of a course assistant, and making a paper sketch of a new view as a proof of concept. Although the view was designed for a single assistant to see their own students' progress, it was also seen as important that an assistant should be able to customize which subset of students is viewed. For example, if a course assistant wants to see another course assistant's view or wants to get a complete overview of the students in the course, the new view should facilitate this.

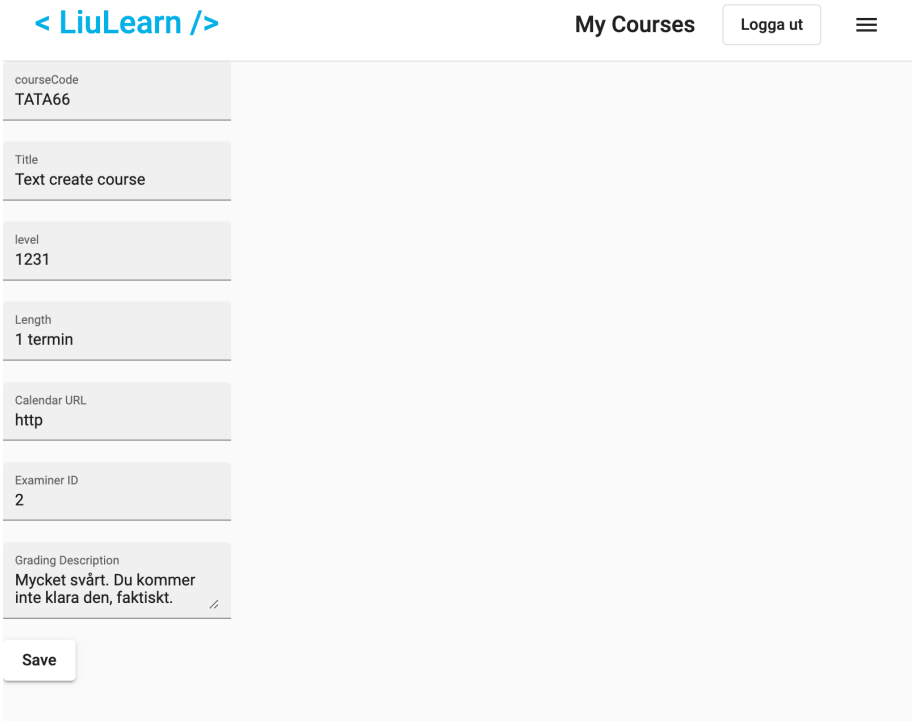

Figure 5.1: The old version of the course creation page

5.2 Implementation

The implementation has been categorized into two major categories, firstly the general upgrades, described in section 5.2.1. These are changes that were made before the new view was evaluated, or were inconsequential to the new view. The other category is the implementation of things related to the new view which are described in section 5.2.2. The implementation of the new view is what was in focus during the evaluation part of the results.

5.2.1 General upgrades

The main priority of the general upgrades was to ensure that the course could be rerun for another year, which included ensuring that a course can be cloned into a new instance for each rerun as well as removing any bugs that arose.

Course reruns

At the initial stage the application enforced unique course codes for different courses, meaning you could not rerun a course in the system. Therefore, the first task became adapting the system to allow course reruns. To facilitate this the old “Create new course” page, which can be seen in Figure 5.1, was improved. A stepped interaction sequence was added for creating a course, which can be seen in Figure 5.2. The main new feature of this page is allowing copying of old courses for a rerun. When copying courses all course material is cloned, as well as the structure of group collections, while all student related data such as course participants and their submissions are ignored.

To differentiate between different reruns of a course the concept of "study periods" was added to represent different semesters. These study periods are reusable between courses to allow for a shared start and end date that applies to all courses in that period.

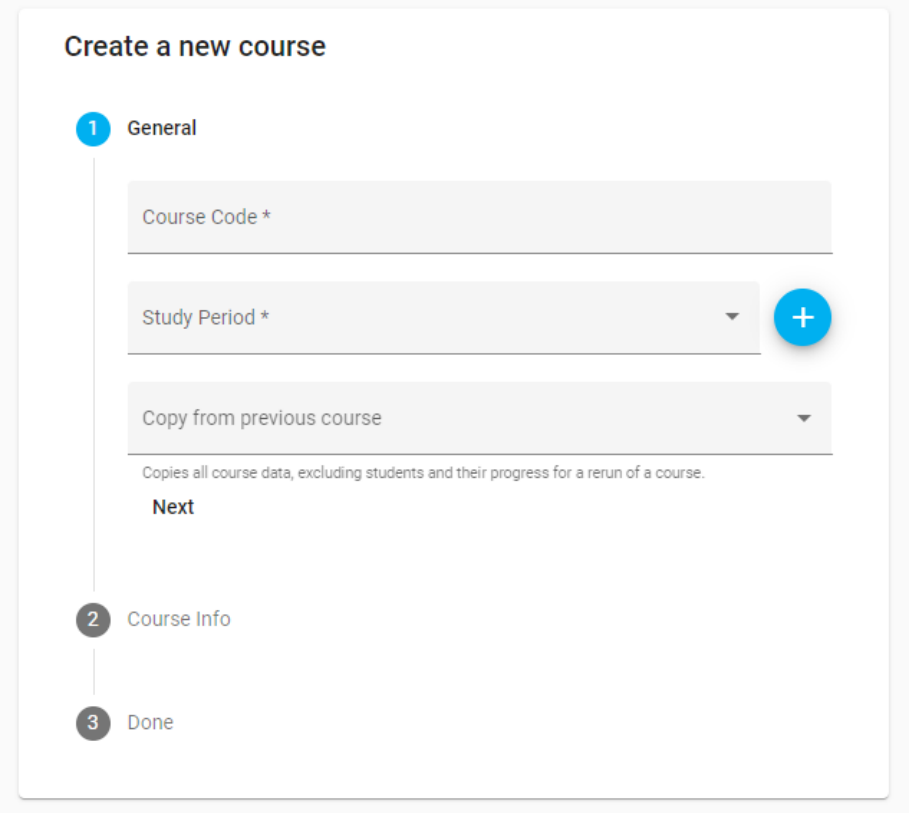

Figure 5.2: The create course sequence at its first step.

As a course is copied it will need updated deadlines, and to facilitate this an extra step was added to the step sequence where users can update the deadlines to suit the new course. New deadlines are automatically suggested based on start dates of the old study period and the selected study period for the new course. If the suggested deadlines are deemed unsuitable by the user they can easily be overwritten. The view for updating deadlines can be seen in Figure 5.3

In addition to the new features some of the previously implemented functionality of the course creation page was improved. This includes the "select examiner" field which was updated to include a searchable selection field, as a replacement to the old field where you manually entered an examiner's user id, which is not easily accessible to a user.

The process to add new students to a course was improved, as a newly created course will always be empty of participants. Multiple features for adding students were implemented. Firstly, the previously unfinished functionality of importing users from the Webreg API was finished. Secondly, a feature to import users from CSV files was added, files which can be downloaded from Linköping University's group manager tool[1]. Both import features can be used to import course assistants or students to the course.

As there were now more courses, the homepage of the teacher view was updated to only display courses the user is actually a teacher for. When a new course is created, the creator and the selected examiner are automatically added as teachers to the course.

Internally, the application was largely based on using course codes as identifiers for different courses, but as multiple courses should be able to have the same course code, this had to be changed throughout the entire system to the unique course ids.

[1]https://gm.liu.se

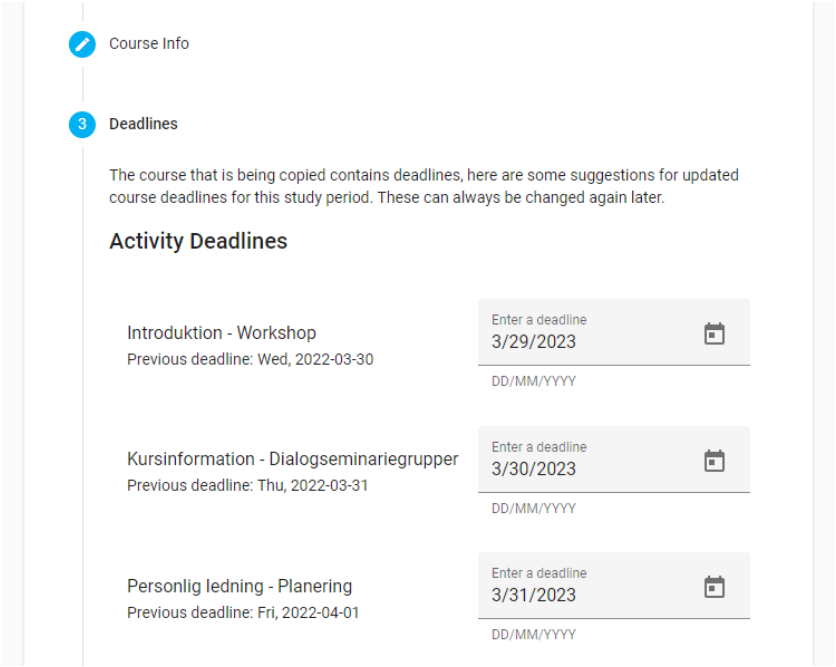

Figure 5.3: The view to update deadlines while copying a course.

Grading view

As part of the general upgrades, the view for grading writing submissions was updated to enable a better grading experience. The updated view can be seen in Figure 5.4. The new grading view was designed to make better use of the horizontal space on bigger screens, by having a two column layout with the question in the first column and the student's answer in the second column. This had the secondary aim to give the appearance of a more free flowing text, without the questions interrupting. On smaller screen sizes a stacked view is used similarly to before, although with slightly changed appearance to differentiate the questions from the answers.

Besides the updated layout there were a few new additions in the grading view. Specifically a toggle to see which sentences have been edited from the last submission when grading a complementary submission, a word count of the submission text and a button for downloading a PDF version of a submission. As the PDF can also be useful for students to facilitate printing their submissions it was also added to the student frontend.

The Groups and Submissions tabs

As a course assistant with a responsibility for grading there were initially two main course tabs of interest, Groups and Submissions. These two tabs both provide the ability to grade submissions through the shared grading view which has been described in Section 5.2.1. When grading through these tabs the difference is how a course assistant locates new submissions. The Submissions tab is a table showing all submissions to the course, which can be seen in Figure 5.5. The Groups tab is a grid showing progress of all students in a group, which can be seen in Figure 5.6. Of these two tabs, only the Groups tab allows a course assistant to grade activities which are not submitted through Eprof, for example confirming participation at a seminar.

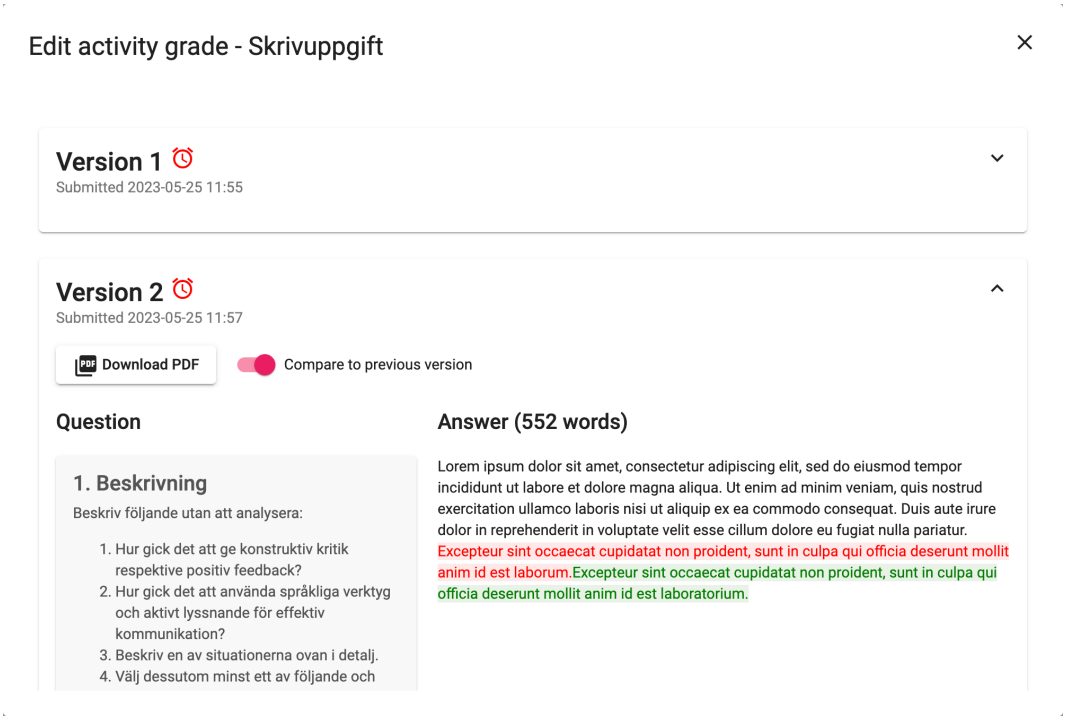

Figure 5.4: The new grading view for a writing submission

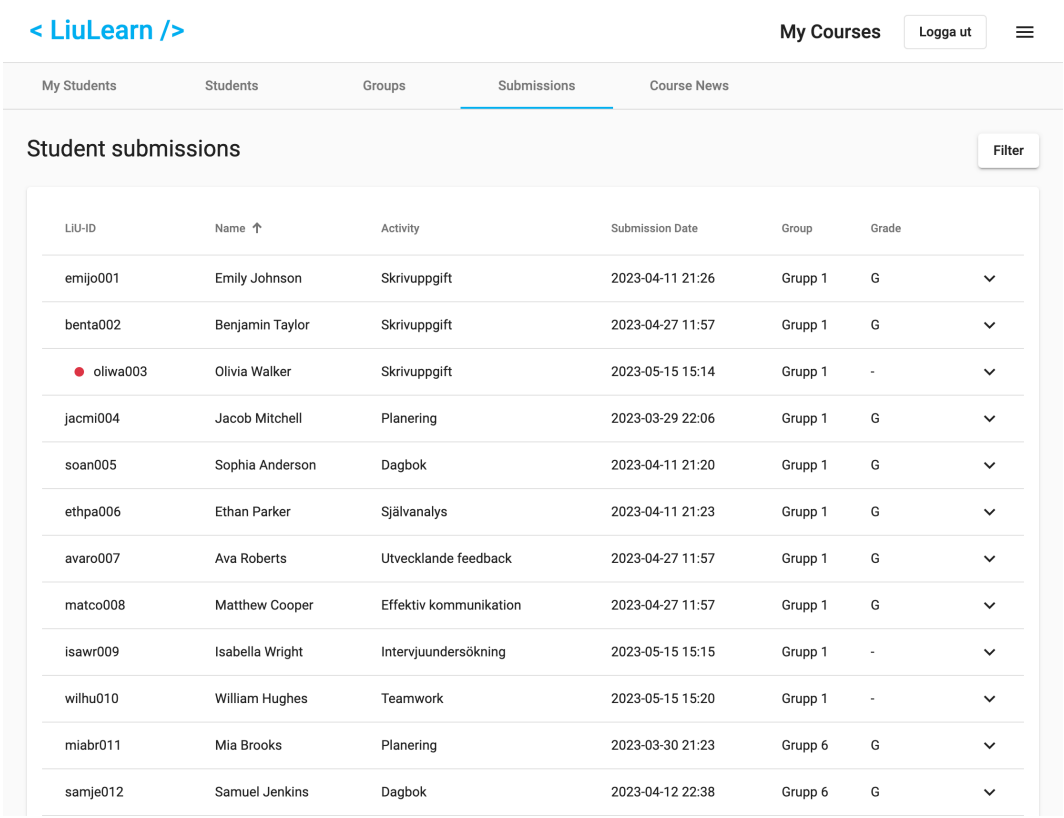

Figure 5.5: The preexisting Submissions tab which was used before the new view.

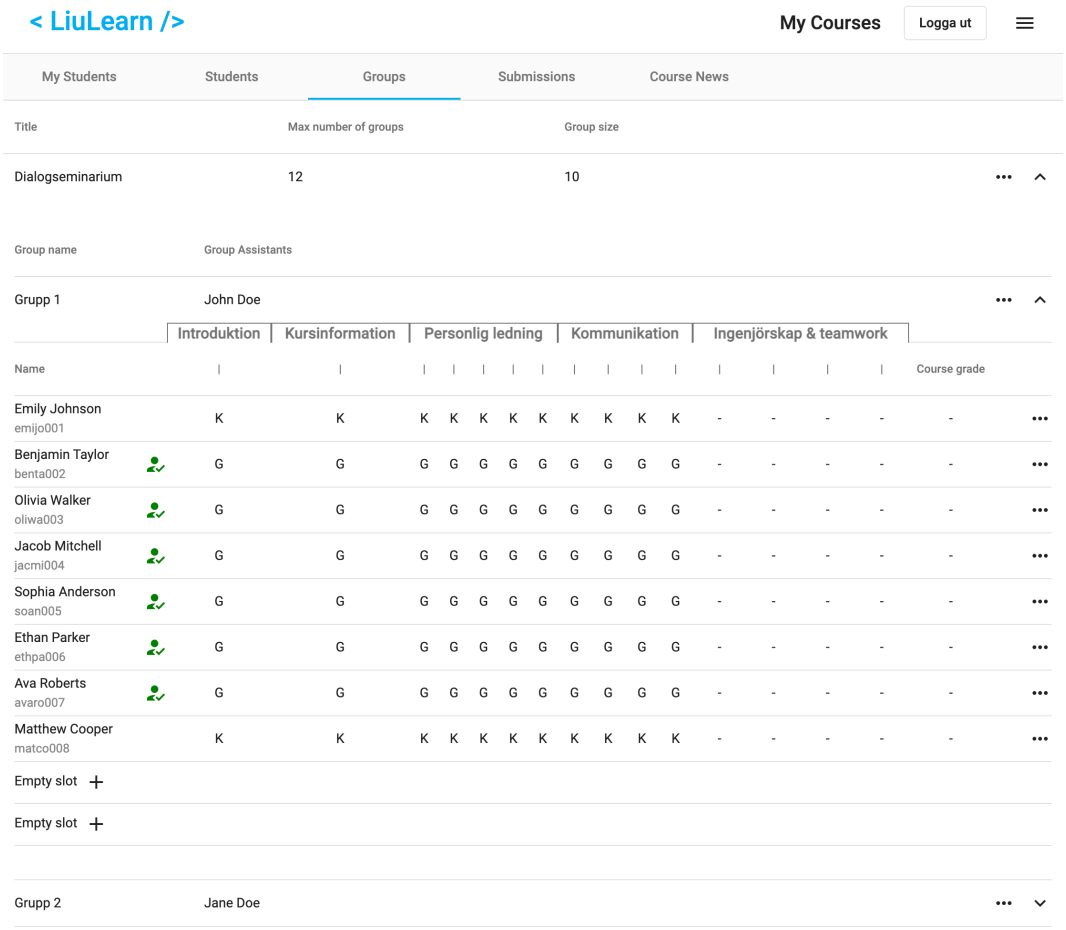

Figure 5.6: The preexisting Groups tab which was used before the new view.

Miscellaneous changes

The progress meters for each student which are visible in both the student's and the teacher's view were initially set to linearly update based on the amount of completed activities, an issue with this was that the reading activities which a student mark as completed by themselves skewed the progress significantly. To address this a value for each course activity was added to denote it's "progression weight". This new value can now be set in the teacher view and allows a user to set how much each course activity should influence a student's course progression. Along with this the progression calculation was changed and optimized by turning it into a database query.

In the system all course activities are connected to a course group collection, which in turn gives every student an assigned course assistant for each course activity. Although this connection existed in the previous implementation, it was not configurable through the teacher view, as a consequence of the unclear connection there were unexpected bugs when the connection was missing. Another feature which was missing was the ability to configure who is the assistant of a specific group. To address these flaws, fields were added in the course settings to configure the connections. Labels were also added to clarify who is the assistant of which group in the group view.

In the previous version of the teacher view there was no way for a user to create course news posts, which are displayed on the home page of the student view for the course. To

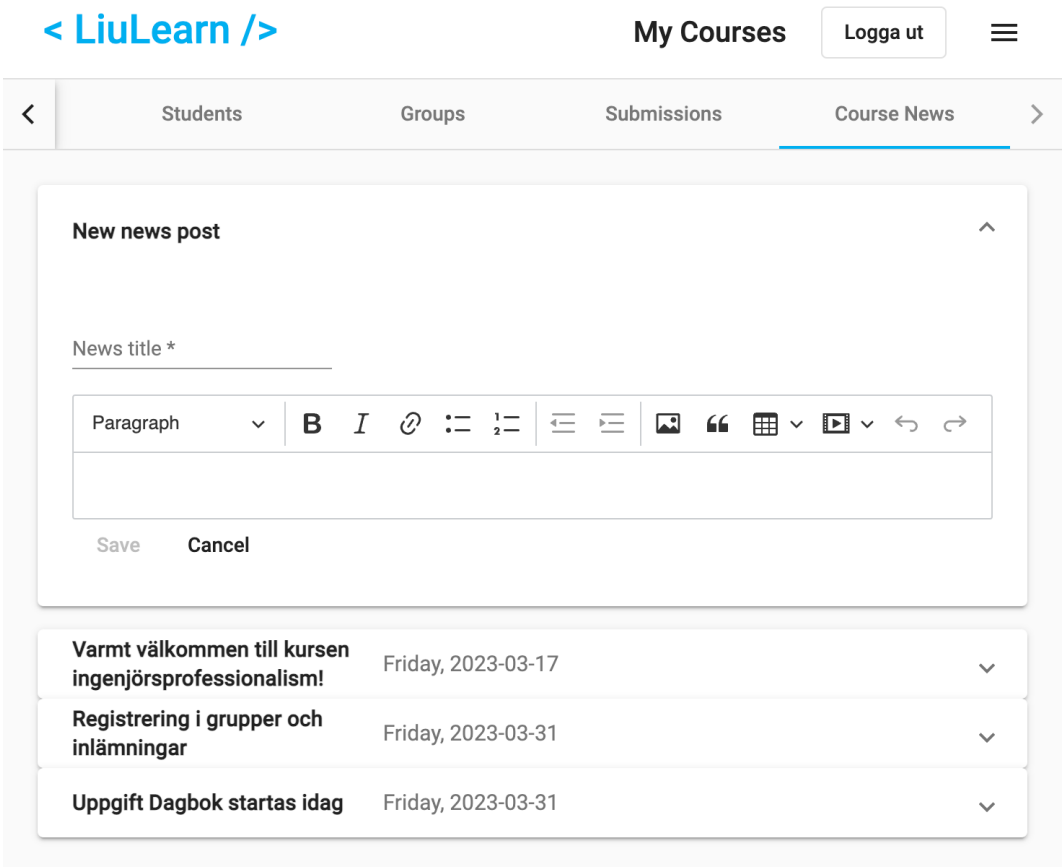

Figure 5.7: The new tab for editing course news to display on the student home page.

address this a page for viewing, adding, editing and deleting course news was added, so that users can easily manage course news. The course news editor tab is shown in Figure 5.7.

As there are still several features lacking in the teacher view, which sometimes need to be accessed, a new admin interface was added to edit any database model, made possible by the module Flask-Admin[2]. This admin interface is only intended to be used by the developers to perform uncommon, non repeated tasks, to save time from developing specific features for these tasks. In addition to the new admin interface a command line interface for improving management possibilities of the database was added. This command line interface enables a system of version control for the database schema and the possibility of switching between versions. The database management is enabled through the library Flask-Migrate[3]. The command line interface was also extended to include an updated database seeding command for use in development, where the database is filled with mock data. Personal information is avoided in the mock data by generating random names for all users.

To facilitate the testing of new features that were added the concept of feature flags was implemented to allow for A/B testing. The primary use case for these has been to make sure new features are working properly before being widely deployed. The feature flags can be added to specific users to give them early access to a specific feature.

To improve the general user experience many small changes were made throughout the system. These changes include better feedback for errors, improved labels throughout the system, some table headers now stay in view when scrolling through long tables, selected

[2]https://flask-admin.readthedocs.io
[3]https://flask-migrate.readthedocs.io/

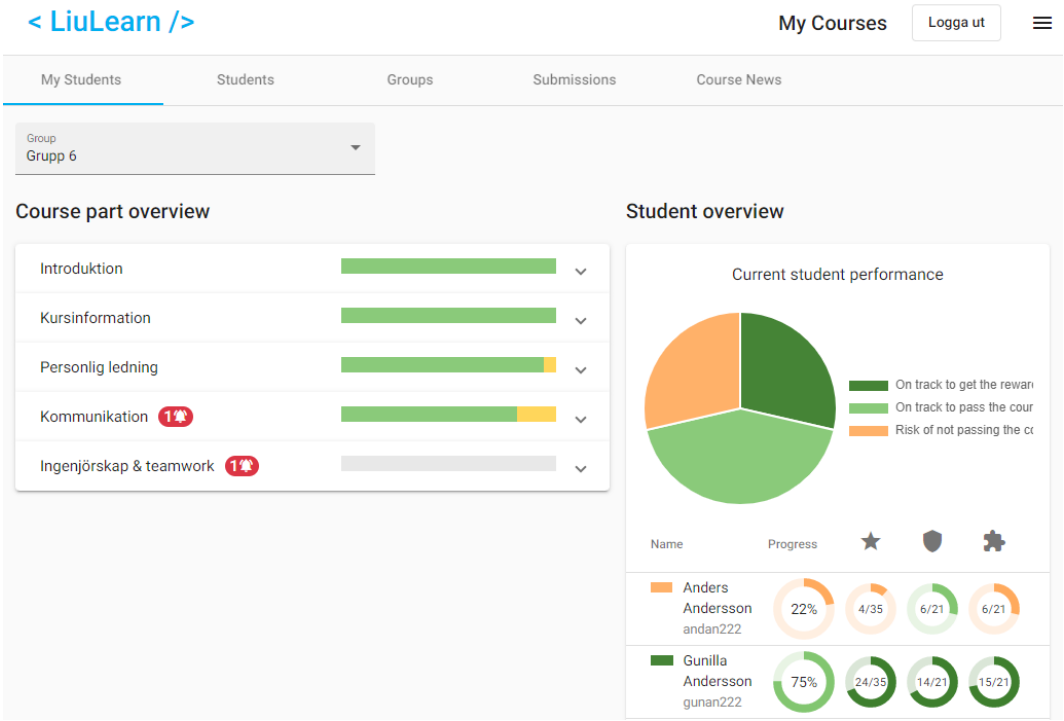

Figure 5.8: My students view

tabs are added to the browser navigation history, teachers no longer have to be added as students in a course to view the student page for the course and other minor stylistic changes.

Lastly, many bugs were found and resolved. Major bug fixes included making time zone data consistent throughout the system, fixing student confirmations of assigned groups and making sure that the user interface updates to reflect changes made using the API in some places where it was missing.

5.2.2 The new view

In accordance with the results from the pre-study, a new view was created for easier grading and a clearer overview of student progression. The view contains a course part overview, a student overview and a drop down list to filter the groups for which the data is shown in the overviews, see Figure 5.8. The group filter was implemented to default to the course groups which the currently logged in course assistant is assigned to. If the currently logged in course assistant is not assigned any groups, course part overview and student overview will not be displayed. Instead, a message will be shown telling the user to select groups in the drop down filter manually. The view displays the course part overview and the student overview in a two column layout, which becomes stacked on small screens.

Course part overview

The course part overview serves the purpose of grading submissions and tasks. Each course part is displayed in the list of expansion panels, see Figure 5.8. Progress bars indicate the grading progress for each course part. Next to the course part title are red badges that display how many new ungraded submissions that specific course part contains. The progress bar in combination with the badges were added to give an overview of the grading and submissions without needing to expand the panel.

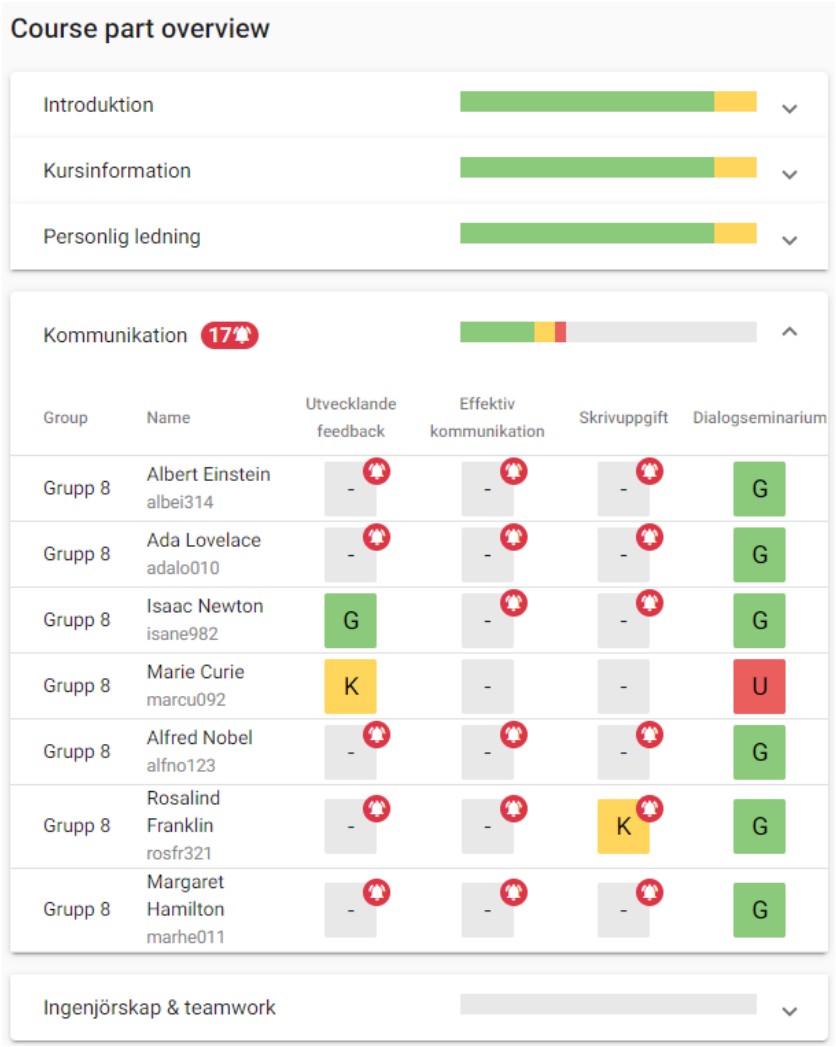

Figure 5.9: Course part overview

Upon expansion of a course part panel a table is shown, see Figure 5.9. This table contains every student in the chosen groups along with their results in each graded activity within the expanded course part. The results are color coded according to their grade, where green is passed, yellow is complementary, red is failed and gray is ungraded. The letter abbreviations correspond to the Swedish translations of the grades (*Godkänd*, *Komplettering* and *Underkänd*). See Table 5.1 for an overview of the grades, their abbreviations and colors respectively.

Table 5.1: Grades and their assigned colors

Grade	**Abbreviation**	**Color**
Passed	G	Green
Complementary	K	Yellow
Failed	U	Red
Ungraded	-	Gray

Red circles with bell icons inside represent ungraded submissions their respective activity. This red badge is displayed together with the colored grade button, and visualizes where the course assistants have grading to do. As seen in Figure 5.9, the red badge can be applied both

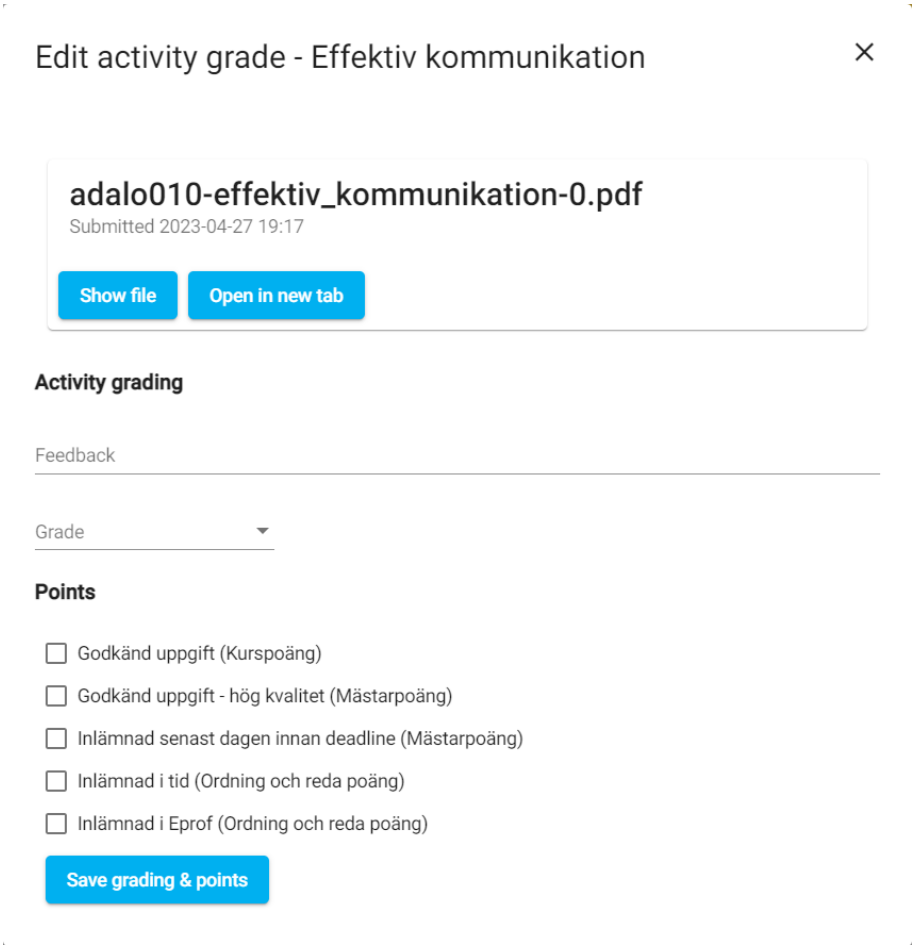

Figure 5.10: Course activity grading modal for a file submission

on activities which have not been graded before, and activities that have been graded but have a new updated submission version.

To grade an activity the colored grade button is pressed and a pop-up modal is opened. Inside the modal the submission is displayed. The submission type is either a file, like in Figure 5.10, or a writing assignment, like in Figure 5.4. The difference in the grading view is that a writing assignment is displayed directly in the modal alongside the instructions to the assignment. There are input fields for feedback and grading as well as a list of checkboxes for the points that can be given for that activity. The available points vary between different activities. When the *Save grading & points* button is pressed the results will be saved and the student will be notified.

Student overview

The right column of the My students view is a component labeled as “Student overview”, shown in Figure 5.11, which contains information about how students in the currently selected group constellation are currently performing. At the top of this component all of its data is summarized into a pie chart with three different categories. Directly below there is a table with the overview for each individual student in the course.

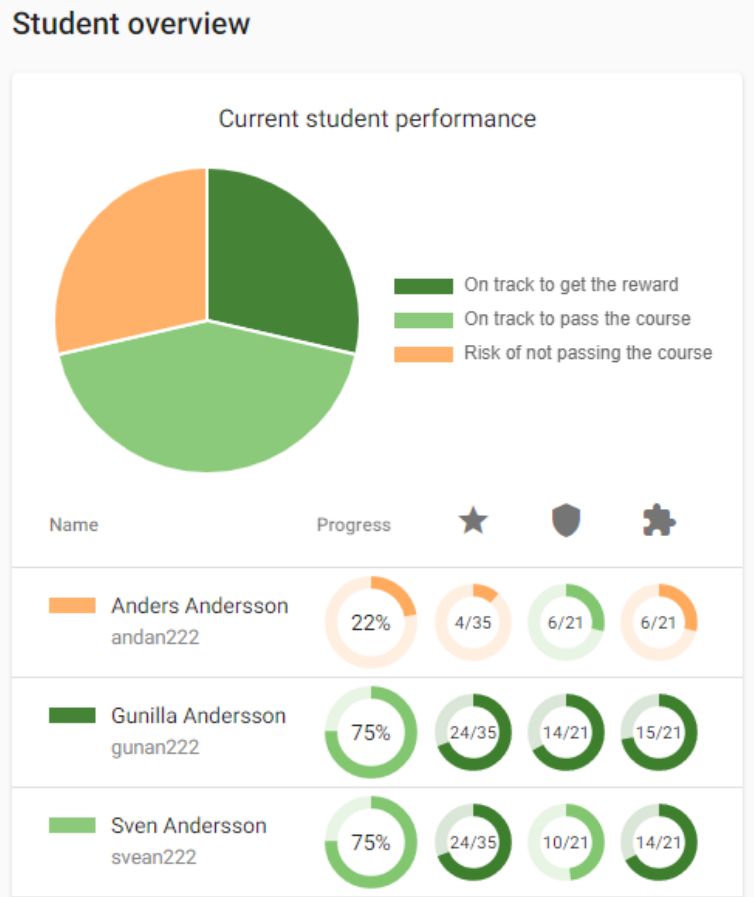

Figure 5.11: Student overview pie chart and table

The pie chart displays how many students are in each of the three following categories, each of which will be referred to as their corresponding color:

- On track to get the reward (dark green)
- On track to pass the course (light green)
- Risk of not passing the course (orange)

Which category an individual student belongs to is decided through the data in the student table, and is also displayed as a colored rectangle before the student's name in the table, the color is decided from the worst result the student has in any point category. For example, if a student has orange color for any point category, they risk not passing. If they instead have light green color for any point category but no orange, they are on track to pass. Lastly, if all the point categories are dark green, they are on track to pass the course with the extra reward.

Which color each individual point category is assigned for each student is decided by several factors. A point threshold for each color corresponding to each point type is calculated based on two main factors, time elapsed in course and point requirements for the course.

The time elapsed factor is based on the deadlines in the course, together with a seven day delay to factor in that assignments will not be immediately graded after the deadline. The assignments that do not have deadlines are counted as if they had the same deadline as the next upcoming assignment which has a deadline. If no upcoming assignment has a deadline, the assumed deadline will be the end date of the course's study period.

The point requirements factor is based on how many points of a certain category a student needs to get to for each color threshold. For the TDDI83 professionalism for engineers course in particular, it adapts to the points thresholds shown in Table 2.1. This factor is used to adjust for courses where you don't need all the points to complete the course.

With the time and point factors combined, the expected threshold will be the amount of points expected at the current date, after subtracting any points that are not required by the course point requirements. Thereby every student will start out expecting to do great in the course, but as time passes this will automatically adjust and start to show more students falling behind.

The table of students was designed as a shared component with the previously existing table in the Students tab. Therefore the layout of the old tab was also updated accordingly. The table component has a few parameters to adapt slightly to the different views. The main addition to the old Students tab was displaying the sum of points for each student, which was previously not possible to do without manually visiting each individual activity grading and manually calculating the sum. Along with the different point types being added to the table, the progress indicator was kept from the Students tab.

5.3 Evaluation

This section presents the results of the questionnaires used to evaluate the changes made to the platform. Data collection was performed before and after the changes to enable comparison. The people used for the evaluation were course assistants, or mentors, in the course TDDI83 and used the platform continuously as the course was taught. The course assistants were between the ages 22–29 and consisted of 40% women and 60% men. The course assistants graded one course part before the new view was implemented, and one after it was implemented. The course parts consisted of the same types of course activities, both containing a number of file submissions, one written assignment and finally a seminar. Thereby, each evaluation was based on the same amount of interaction with the platform.

5.3.1 Opinions on Eprof

The first set of questions in the evaluation questionnaire were related to the users' opinions of Eprof. The results of the questionnaire and the thematic analysis of the open-ended questions and motivation behind the answers on the scale are presented in this section.

General opinions

The results before and after the changes implemented is shown in Figure 5.12. The figure indicates an improvement of the course assistants' opinion of the teacher view in Eprof after the implementation of the new view. A clearer understanding of the course assistants' perception of the changes is given in Figure 5.13, where all course assistants considered the changes to be very positive. They thought the changes were a big improvement that made it easier both when grading and in obtaining an overview of student state. One on the course assistants said that "the new update has really made a difference" and that "it is much easier to navigate the teacher view now".

In the first version, some assistants stated that hey lacked the functionality of viewing only their own students. In the second version, the ability to filter the data on groups, and specifically only showing the course assistants own groups, was received as a positive change mentioned several times in the questionnaire.

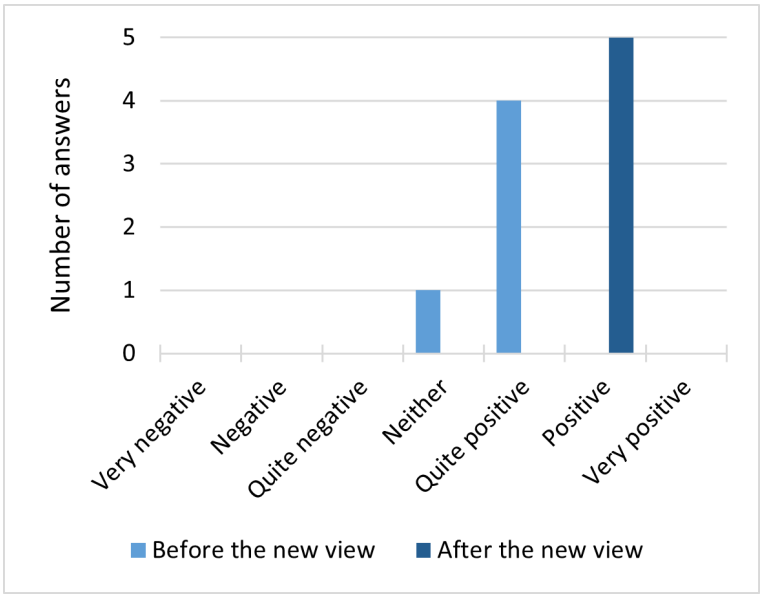

Figure 5.12: Answers to the question "What is your opinion of the teacher view in Eprof?"

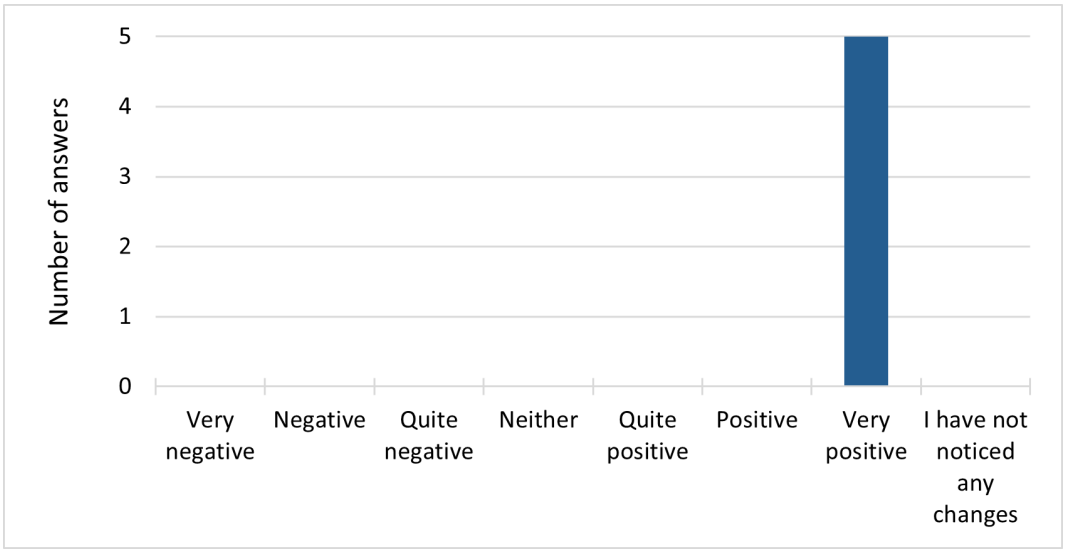

Figure 5.13: Answers to the question "What is your opinion of the changes made to Eprof?"

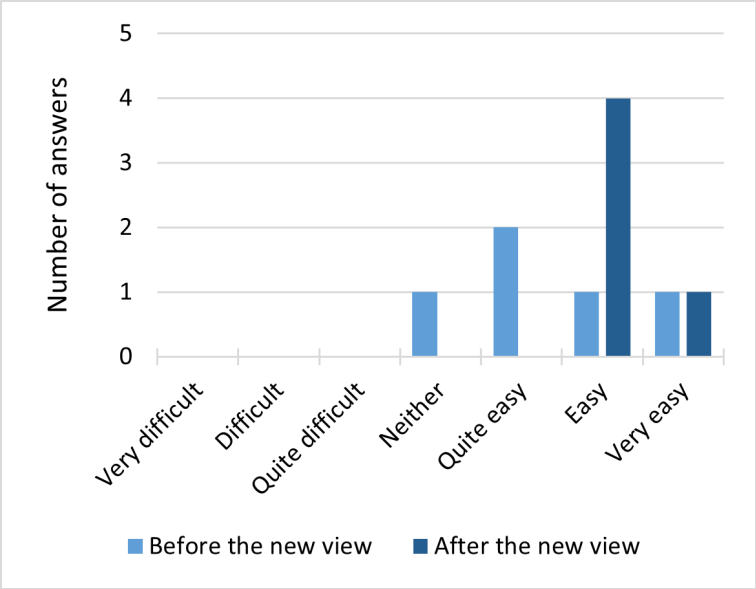

Figure 5.14: Answers to the question "How easy or difficult is it to keep track of how the students are doing in the course with the help of Eprof?"

Student progression and visualization

Regarding the platforms ability to convey student progress, the course assistants mentioned the difficulty in interpreting the progress data in the student tab in the first version of the platform. The ability to get an overview of student progression was largely improved with the new view, which can be shown in Figure 5.14.

The new visualizations were seen as "positive" and "very positive", see Figure 5.15. The color coded estimated level was according to the course assistants an appreciated addition. The pie chart in combination with the table of students and their points were said to present a clear overview of student progression. Furthermore, this made it easier for course assistants to notice any student falling behind. However, it was mentioned that there were too many students marked with risk of not passing.

Grading

The grading process was in the first version of the system viewed quite positively, see Figure 5.16. Factors mentioned that the course assistants enjoyed was the list of checkboxes used to give points to the assignments. The grading modal was kept without modification in the new view.

A mentioned improvement was the overview of grades given for the course activities. When the groups tab was used for grading, the course assistants mentioned the difficulty in finding the correct button to press because of the compressed layout leading to the course activity titles not being displayed. In the new view these titles were spread out and visible, which was mentioned as an improvement that made the grading process easier. Also, the color coded buttons were mentioned to improve the overview.

When asked about future improvements to the first version, some course assistants mentioned they wanted a way to see which course activities had new submissions to it. This was implemented in the new view with positive feedback from the course assistants and seen as a "very good improvement". Other future improvements mentioned were automatic grading for points such as "handed in before deadline" and "number of words in assignment is within a required interval".

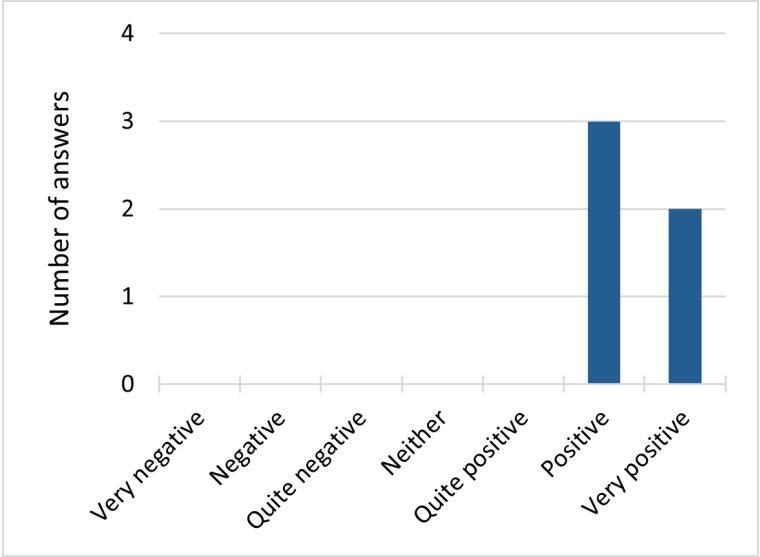

Figure 5.15: Answers to the question "How do you experience the visualization of students' progression and scores in Eprof?"

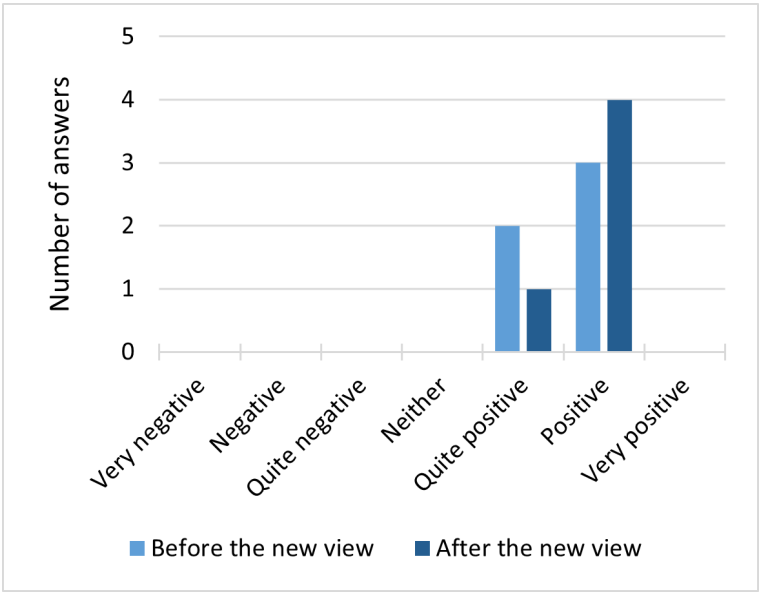

Figure 5.16: Answers to the question "What is your view on correcting students' tasks in Eprof?"

5.3.2 TAM

The second part of the questionnaire included questions from TAM as well as two additional questions as presented in section 4.3.2. Figure 5.17 shows the results from the TAM questionnaire both before and after the new view was implemented. Each question has two bar charts where the top one represents the answers before the new view was implemented, and the bottom one represents the answers after the new view was implemented. The average score from the TAM analysis is shown in Table 5.2, where each step of the scale corresponds to a score of one to seven, as shown in Table 4.3. A table of the average score from TAM grouped on type of measurement (PU, PEOU and solution objective) is shown in Table 5.3. Both tables present the average before and after the new view was implemented. The grouped table indicates a slight improvement in PU, PEOU and solution objective respectively.

Table 5.2: TAM questions and average answer before and after the new view was implemented

Nbr	**Question**	**Average before**	**Average after**
1	Using Eprof in my job as a mentor enables me to accomplish tasks more quickly	6.4	6.8
2	My interaction with Eprof is clear and understandable	6.4	6.4
3	Using Eprof makes it easier for me to do my work as a mentor	6.4	6.8
4	I find it easy to get Eprof to do what I want it to do	5.6	6.0
5	I find Eprof useful in my work as a mentor	6.4	6.8
6	Eprof helps me identify students who are behind in the course and who may need more support from me as a mentor	5.4	6.4
7	Using Eprof enhances my effectiveness in my work as a mentor	6.2	6.6
8	Learning to operate Eprof is easy	6.4	6.4
9	I find Eprof to be flexible to interact with	5.2	5.8
10	It is easy to become skillful at using Eprof	6.2	6.4
11	Using Eprof improves my job performance	6.0	5.8
12	Using Eprof increases my productivity	5.8	6.2
13	Eprof helps me get an overview of how the students are doing in the course	6.0	6.8
14	I find Eprof to be easy to use	6.2	6.2

Table 5.3: TAM questions average before and after the new view, grouped on category

Category	**Average before**	**Average after**
PU	6.2	6.5
PEOU	6.0	6.2
Solution objective	5.7	6.6

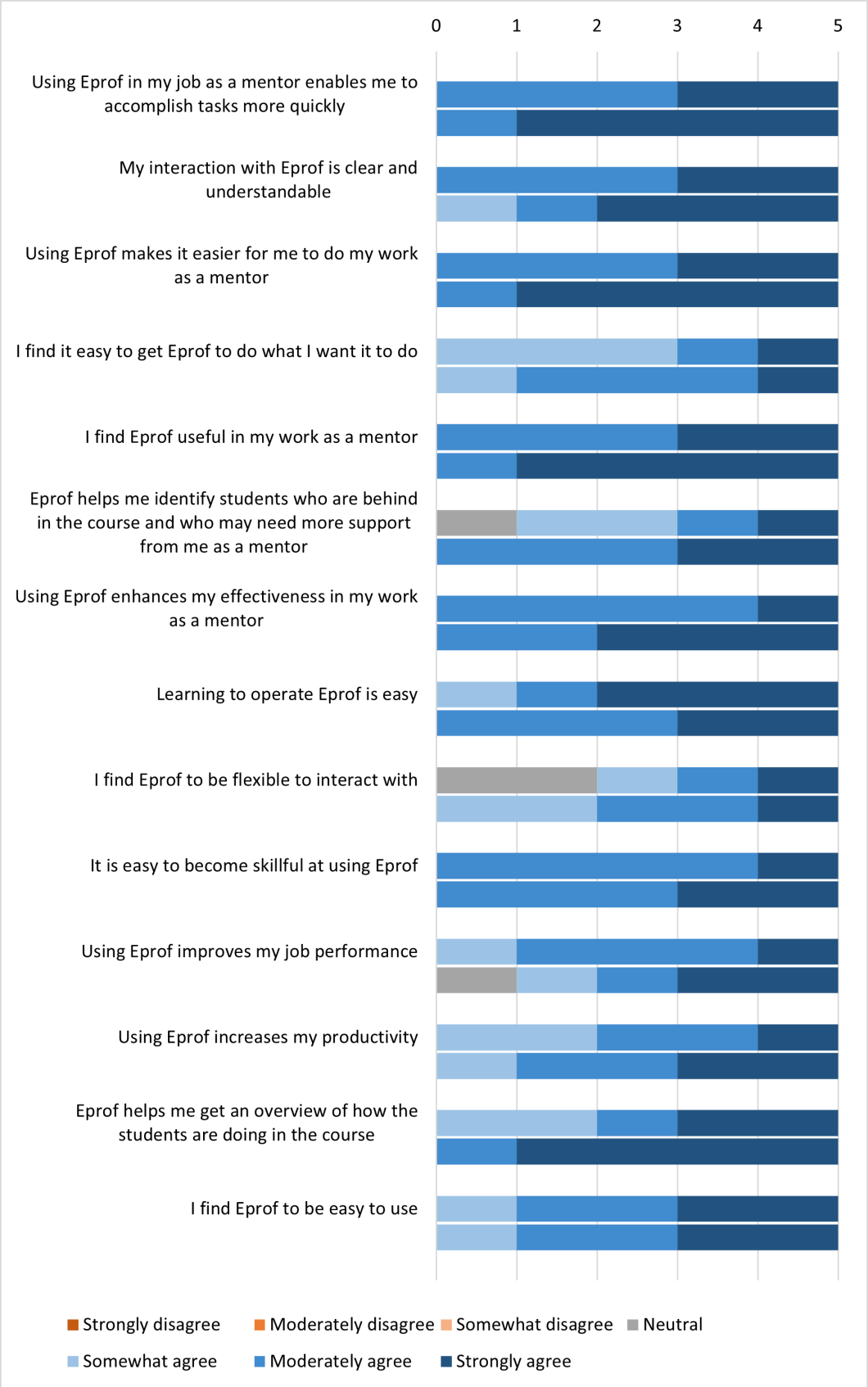

Figure 5.17: TAM results before and after the new view was implemented

6 Discussion

This chapter discusses the results in section 5, the method in section 4, and finally how this work relates to a wider context.

6.1 Results

This section aims to discuss the results of the study, primarily by focusing on how the results could have been improved based on the findings in this study.

6.1.1 Pre-study

The results of the pre-study involved finding three main issues to be resolved in the application. Namely the inability to properly do course reruns, the difficulty of tracking student progress and the cluttered grading process. As a result of the last two issues, a new view was designed to accommodate these needs.

As the new view was released to the course assistants after the first set of questionnaires had been conducted, the opinions of the course assistants on the previous way of grading largely matched with the results of the pre-study. Thereby the results of the pre-study can be verified to have been relevant. Whether or not the view proposed in the sketched design was a favourable way to solve these issues is further validated through the second questionnaire round.

6.1.2 Implementation

One aspect which is lacking in the implementation is the ability to see a detailed overview of a single student's progress, for example in the case where a single point is missing, it can be hard to pinpoint which point it is without manually going through every submission. A solution to this could be to have a detailed result view for each student, for example in a modal window when a user's name is pressed. This view could include an overview of each activity and give information on which of the available points have been given for each individual activity. Furthermore, the modal window could be used when implementing a feature to set the final course grade of a student.

When implementing the new view, the old progress percentage value was kept beside the different point type indicators, although as a slightly bigger indicator to help differentiate the two metrics. This can be seen in Figure 5.11. The progress indicator has shown to have limited usefulness for a course assistant while still serving a purpose for the students. With the ability to mark a reading activity as completed, and instantly receiving a small bump in progress, a student can get a sense of instant accomplishment. In the course assistants' perspective of grading a student, only the point type indicators will be of interest. Some students might even be shown as being below the expected progress to the course assistant due to not having the desire of marking every activity as completed. The findings by van Leeuwen [23] suggest that the progress indicator could be interpreted as a confusing metric, where the course assistants could become unsure in which way to intervene to a low progress indicator, when the indicated value might in fact be out of their control and insignificant. Furthermore, the complexity the progress indicator adds to the student overview justifies the case that the removal of the progress indicators in the teacher view should be considered. At the same time the progress indicator has previously been shown to serve an important gamification purpose from a student perspective in the study by Sandberg and Sjöqvist [11], which signifies that the progress indicator should still be preserved for displaying the student view. If a more detailed result view for a specific student is at some point implemented, as previously discussed, the progress indicator could be shown to the course assistant in a less significant manner in the detail view, preferably in a design inspired by how it is displayed on the student view.

When calculating if a student is on track to pass the course, or if the student is at risk of not passing, a seven day delay was implemented to take into account that a course assistant will not instantly grade the submission. Although grading will almost never be done without delay, what is to say that the course assistant will grade within seven days specifically. During the evaluation process there was a point where seven days had passed after a deadline and no course assistant was completely done grading. This resulted in the student overview showing that all students risk failing the course, which certainly was not the case. When the course assistants then finished grading the numbers were again more convincing. To address this inconsistency in the predicted outcome there are several solutions that can be proposed. Initially removing the set time delay after deadline and introducing another solution to the problem of delayed grading should be done. The solution could be as easy as assuming that an ungraded submission is perfect until proven otherwise. To improve on that a predictive model could be applied to predict the statistical probability that each available point from a submission will be received. Thereby the difference in expectations before and after grading should be minimized, and the timing of grading is no longer important. A course assistant submitting a grading to a submission will instead only slightly increase the accuracy, based on how good the initial prediction was. As previously discussed in section 3.2 there have been studies on how to more accurately predict student outcomes. For example Smith et al. [24] describe how factors such as log-in frequency, site engagement, pace, assignment grades and more can be effective to predict course outcome, even just eight days into a course in their scenario. All these factors could be applied in the Eprof scenario as well, while the factors described by Zacharis [25] are less applicable as they rely on features which are not part of Eprof.

6.1.3 Evaluation

The opinions of Eprof can in general be seen as positive. The opinions were quite positive even before the changes, but the graphs and the comments from the course assistants indicates an improvement. Regarding Figure 5.12, the general opinion of Eprof is improved from quite positive to positive. Even though the changes were deemed as "very positive" in Figure 5.13 by all course assistants, which is the highest available alternative, the overall opinion was not "very positive". This could be caused by the missing functionality in Eprof

that was mentioned by the course assistants in the open-ended questions. Many of these were described as nonessential, but with them implemented the results might be improved.

The difference in the opinions of the grading process were minor. This is reasonable since the actual grading process was not changed. However, what is considered included in the grading process can be ambiguous, which might explain why the results of the question were not unchanged. There were changes to the display of given grades more clearly and less compressed, which could be interpreted as a part of the grading process. These changes could also possibly affect the answers regarding the visualization of progress and how easy it is to keep track of how students are doing in the course. These two graphs indicate a large improvement, but whether or not they are related to the visualization of grades depends on the course assistants' interpretation of the questions. Observing the text answers it is however apparent that the changes improved their work as course assistants.

Through the questionnaire, it was found that the visualizations introduced were appreciated by the course assistants and that it helped them get an overview of how the students were doing in the course. As a result, they could more easily identify students who were not on track and might needed extra help. This coincides with the previous research mentioned in section 3.2.2 where Dichev et al. [27] states that a gamified learning platform combined with learning analytics help teachers track student progress and notice students who are falling behind.

Regarding the technology acceptance of Eprof and the results from TAM, the scores were high for both before and after the new view was implemented. It can be argued that the improvements were small, 0.3 for PU and 0.2 for PEOU. However, since the initial scores were high there was little room for improvement. The initial score for PU was 6.2 and the maximum possible score is 7, leaving room for an improvement of 0.8. Considering this, an improvement of 0.3 can be seen as quite good. The same applies to the PEOU, where the first score was 6.0 out of 7, and after the new view was applied the score was 6.2. This is an improvement of 0.2 out of 1.0 possible.

Regarding the two additional question added to TAM, the improvement were even better. In identifying students who are behind and need more support, the score increased from 5.4 to 6.4, an increase of 1.0 out of 1.6 possible. The second additional question regarding the ability to get an overview of how the students are doing in the course, the score went from 6.0 to 6.8, meaning an improvement of 0.8 out of 1.0 possible.

6.2 Method

This section aims to discuss the method of the study, primarily by focusing on how the method could have been improved based on the findings in this study.

6.2.1 Pre-study

As the course started late March while the work on the thesis started in the middle of January there was a large delay until the system could properly be evaluated through the course assistants. The first questionnaire was conducted in the middle of April as the students need time to get started with the course before the first grading can be done. This made it difficult to use the help of the course assistants earlier as they would not be well-versed in the course or system yet. The effect of this was a distorted time plan where the first opportunity to evaluate came far into the thesis time plan. The first useful information from the course assistants was deemed to come after the first grading round, in this case mid April. Had the course started earlier the course assistants could have been used to conduct the pre-study, instead of just verifying what the more limited pre-study concluded. The course has three main parts with submissions, but due to the limited time, the last round of submissions could not be used for any evaluation in the scope of time for this thesis.

6.2.2 Implementation

The implementation phase could have been more clearly defined from the beginning, in particular what was the prioritized task in relation to the method. Instead to a large extent the focus of the initial implementation phase was some general improvements throughout the system, rather than focusing on the goal of the thesis. As an example, the new grading view was added before the first questionnaires, thus having it not be part of the evaluation of the differences between the old and new experience. With a clearer definition of the implementation goals from the beginning, and categorization of every task to its relevance of evaluation, more of the implemented features could have been subject to the evaluation process. However, there was still a system for prioritization of tasks to implement in the form of a kanban board.

6.2.3 Evaluation

By using the course assistants as participants in the evaluation process there was a very limited amount of participants, in this case five. Although they were few, they allowed for a real world scenario as they were involved throughout the entire course, able to thoroughly use the system when grading real students. Although the real course could be used for evaluation the entire course duration could not be used for the study as previously mentioned. At the time of writing this, the final course results are not set, therefore the learning analytics from the expected student performance pie chart can not be fully evaluated. If the course results were final, the expected student performance at different points of the course could have been evaluated against the actual outcome.

Furthermore, the participants in the evaluation were not anonymous. They were in addition paid employees, hired as course assistants. These two factors could possibly result in them having a positive bias during the evaluations.

The thematic analysis of the text answer responses in the survey could show a bias in the interpretation of the person performing the analysis. Although this is the case, the text answers should be seen as a complement to the answers on the Likert scale, with a main objective to provide context for the results in the scale. In some of the text answers, it can be difficult to interpret exactly what the respondent meant, in this case the answer had to be ignored in the analysis.

The user evaluation with a questionnaire did provide much information about the use and opinions of Eprof. However, it could be beneficial to also investigate how the system is actually used by the course assistants to find the flaws in the system which were not presented in the questionnaire. The questionnaire is likely to bring forth the most urgent flaws as thought by the course assistants, while observing usage may reveal remaining and possibly more hidden faults. This would also entail the possibility to tailor the platform to real usage.

6.3 The work in a wider context

Implementing learning analytics into an LMS can be an ethical question. If too much data is collected it can be a privacy concern. A simple example of an analytic could be how much time a student spends on a task, where a teacher could use this information to identify that a student might be in need of help. From the student's perspective this could either be seen as a positive act giving them extra attention, but they might also find it intrusive to their privacy. A study by Korir et al. [38], with the aim to compare how students perceive privacy concerns in different contexts, they find that students can often be more trusting when sharing data with their university than an e-commerce site, especially in trusting that data is not shared with a third party. To avoid these ethical issues in Eprof, no additional data other than what is required for grading is collected or used, even though extra data collection could improve

the learning analytics, and some analytics could even be implemented through anonymous data.

Taking privacy a step further, an LMS has the potential to make further anonymization possible. For example, by not showing which individual is graded for certain submissions at the point of grading, any biases regarding ethnicity, gender or personal connections are removed. This is something that has already been done when grading exams at many universities. A study on anonymous grading by Bygren [39] showed that anonymizing grading has an effect, even if the biases seem quite insignificant or even reversed. On the other hand any anonymization an LMS contributes to can lead to further detachment in the connection between a student and a teacher, compared to a traditional teaching environment.

7 Conclusion

This thesis presents the implementation and evaluation of the further development of an LMS. The main focus of the development and evaluation was the administration view, in particular the grading process for course assistants. The evaluation was performed in a real world scenario where the course assistants of a course in professionalism for engineers were the evaluators. The evaluation aimed to conclude if a new view specifically designed for course assistants would help improve the usefulness and usability of the system to the assistants.

7.1 Research question

The research question answered in this thesis project was **how can an administration tool for an LMS be implemented to give high usefulness and usability for the users**. Through the use of a questionnaire emerged the conclusion that the changes had made an overall positive impact on the usefulness and usability from the perspective of course assistants. More specifically, the thesis demonstrates that the use of data visualization of learning analytics can help course assistants in grading and supporting students, and thereby improve the usefulness. Such visualizations include students' points, prediction of student outcomes and their progress in course parts. Through the data visualization a course assistant can obtain rapid insights into both a student group in its entirety and the students as individuals, which has proven to increase the usefulness of the system. The addition of predicting student outcome was shown to be useful as a concept, but the importance of accuracy in the predictions was evident. For student output prediction to be more useful the prediction method has to be improved.

By implementing a new grading view for course assistants, all aspects of the grading process for a course assistant were integrated into a single page, while hiding unrelated features such as modifying groups and managing students assigned to other assistants. This was done in addition to adding more information through the data visualizations. Gathering relevant features for course assistants into a single view is deemed to have improved the usability of the system.

7.2 Future work

The implemented system has been proven useful in the specific course which has been evaluated, but for broader conclusions on how the implemented features could be incorporated in other courses, the system would have to be evaluated in more contexts. No attempts have been made to incorporate the system in another course, but being developed with multiple courses in mind, the system could be adapted for and evaluated in other contexts. Future studies can be done evaluating how the system could be broadened and how it can be adapted for wider usage in courses with different needs, by making use of the changes described in this thesis.

Bibliography

[1] Rahat Raja and P. C. Nagasubramani. “Impact of modern technology in education”. In: *Journal of Applied and Advanced Research* 3.1 (2018), pp. 33–35.

[2] Ryann K Ellis. “Learning management systems”. In: *Alexandria, VA: American Society for Training & Development (ASTD)* (2009).

[3] Craig M. MacDonald and Michael E. Atwood. “What Does It Mean for a System to Be Useful? An Exploratory Study of Usefulness”. In: *Proceedings of the 2014 Conference on Designing Interactive Systems*. DIS ’14. Vancouver, BC, Canada: Association for Computing Machinery, 2014, pp. 885–894. ISBN: 9781450329026. DOI: 10.1145/2598510.2598600.

[4] ISO 9241:2018(En). *Ergonomics of human-system interaction - Part 11: Usability: Definitions and concepts*. Standard. 2018.

[5] Aseel Berglund and Fredrik Heintz. “Integrating soft skills into engineering education for increased student throughput and more professional engineers”. In: *Pedagogiska inspirationskonferensen-Genombrottet* (2014).

[6] Aseel Berglund. “Professionalism For Engineers: Soft Skills In Engineering Education To Prepare For Professional Life”. In: *14th International CDIO Conference*. 2018.

[7] Aseel Berglund. “Online Courses for Teaching Engineering Professionalism”. In: *19th International CDIO Conference*. 2023.

[8] Björn Möller Ehrnlund. “Enriching the user experience of e-learning platforms using responsive design: a case study”. MA thesis. Linköping University, 2021.

[9] Emma Algotsson. “Design and Evaluation of an E-learning Platform to support Active Learning”. MA thesis. Linköping University, 2021.

[10] Erica Ahlqvist. “Implementation And Evaluation of Gamification Elements In An E-learning Platform For Engineering Education”. MA thesis. Linköping University, 2021.

[11] Oscar Sandberg and Jakob Sjöqvist. “Implementing a gamified elearning platform to teach soft skills to bachelor students”. MA thesis. Linköping University, 2022.

[12] Johannes Klasson and Christoffer Henriksson. “Implementation and usability evaluation of a responsive e-learning administration tool”. MA thesis. Linköping University, 2022.

[13] Aseel Berglund and Izabella Jedel. “Higher Education Students’ Perceptions of the Game Element Points in a Gamified Learning Management System”. In: *The 7th International GamiFIN Conference*. 2023.

[14] Nor Aziah Alias and Ahmad Marzuki Zainuddin. “Innovation for better teaching and learning: Adopting the learning management system”. In: *Malaysian online journal of instructional technology* 2.2 (2005), pp. 27–40.

[15] Lee Yen Chaw and Chun Meng Tang. “What Makes Learning Management Systems Effective for Learning?” In: *Journal of Educational Technology Systems* 47.2 (2018), pp. 152–169. DOI: 10.1177/0047239518795828.

[16] Steven Lonn and Stephanie D. Teasley. “Saving time or innovating practice: Investigating perceptions and uses of Learning Management Systems”. In: *Computers Education* 53.3 (2009), pp. 686–694. ISSN: 0360-1315. DOI: https://doi.org/10.1016/j.compedu.2009.04.008.

[17] Jose Jesus Castro-Schez, Ester del Castillo, Julian Hortolano, and Alfredo Rodriguez. “Designing and Using Software Tools for Educational Purposes: FLAT, a Case Study”. In: *IEEE Transactions on Education* 52.1 (2009), pp. 66–74. DOI: 10.1109/TE.2008.917197.

[18] Lauren Cifuentes, Rinki Suryavanshi, and Alexandra Janney. “Motivating Instructors and Administrators to Adopt e-Learning”. In: *Leading and Managing e-Learning: What the e-Learning Leader Needs to Know*. Ed. by Anthony A. Piña, Victoria L. Lowell, and Bruce R. Harris. Cham: Springer International Publishing, 2018, pp. 189–202. ISBN: 978-3-319-61780-0. DOI: 10.1007/978-3-319-61780-0_14.

[19] Larian M Nkomo and Muesser Nat. “Discovering students use of learning resources with educational data mining”. In: *2016 Honet-Ict*. IEEE. 2016, pp. 98–102.

[20] Dhiraj Upadhyaya and Anchal Garg. “Leveraging on gamification and learning analytics for improved student learning”. In: *2019 International Conference on Computing, Power and Communication Technologies (GUCON)*. IEEE. 2019, pp. 132–138.

[21] Uday Kumar Mothukuri, B Viswanath Reddy, P Naveen Reddy, Sarada Gutti, Kumar Mandula, Ramu Parupalli, CHAS Murty, and E Magesh. “Improvisation of learning experience using learning analytics in eLearning”. In: *2017 5th National Conference on E-Learning & E-Learning Technologies (ELELTECH)*. IEEE. 2017, pp. 1–6.

[22] Mike Tissenbaum, Camillia Matuk, Matthew Berland, Leila Lyons, Felipe Cocco, Marcia Linn, Jan L Plass, Nik Hajny, Al Olsen, Beat Schwendimann, et al. “Real-time visualization of student activities to support classroom orchestration”. In: Singapore: International Society of the Learning Sciences, 2016.

[23] Anouschka van Leeuwen. “Teachers’ perceptions of the usability of learning analytics reports in a flipped university course: when and how does information become actionable knowledge?” In: *Educational Technology Research and Development* 67.5 (2019), pp. 1043–1064. ISSN: 1556-6501. DOI: 10.1007/s11423-018-09639-y.

[24] Vernon C Smith, Adam Lange, and Daniel R Huston. “Predictive modeling to forecast student outcomes and drive effective interventions in online community college courses.” In: *Journal of asynchronous learning networks* 16.3 (2012), pp. 51–61.

[25] Nick Z. Zacharis. “A multivariate approach to predicting student outcomes in web-enabled blended learning courses”. In: *The Internet and Higher Education* 27 (2015), pp. 44–53. ISSN: 1096-7516. DOI: https://doi.org/10.1016/j.iheduc.2015.05.002.

[26] Kamilla Tenório, Diego Dermeval, Mateus Monteiro, Aristoteles Peixoto, and Alan Pedro da Silva. "Exploring Design Concepts to Enable Teachers to Monitor and Adapt Gamification in Adaptive Learning Systems: A Qualitative Research Approach". In: *International Journal of Artificial Intelligence in Education* 32.4 (2022), pp. 867–891.

[27] Christo Dichev, Darina Dicheva, and Keith Irwin. "Gamification driven learning analytics". In: *Proceedings of the 13th International Conference on e-Learning*. 2018, pp. 70–76.

[28] Jonathan Grudin. "Utility and usability: research issues and development contexts". In: *Interacting with computers* 4.2 (1992), pp. 209–217.

[29] Guðrún Hulda Jónsdóttir Johannessen and Kasper Hornbæk. "Must evaluation methods be about usability? Devising and assessing the utility inspection method". In: *Behaviour & Information Technology* 33.2 (2014), pp. 195–206. DOI: 10.1080/0144929X.2012.751708.

[30] Fred D Davis. "Perceived usefulness, perceived ease of use, and user acceptance of information technology". In: *MIS quarterly* (1989), pp. 319–340.

[31] Nikola Marangunić and Andrina Granić. "Technology acceptance model: a literature review from 1986 to 2013." In: *Universal Access in the Information Society: International Journal* 14.1 (2015), pp. 81–95. ISSN: 1615-5289.

[32] William R King and Jun He. "A meta-analysis of the technology acceptance model". In: *Information & management* 43.6 (2006), pp. 740–755.

[33] Maslin Masrom. "Technology acceptance model and e-learning". In: *Technology* 21.24 (2007).

[34] Mohamad Amiruddin Mohamad, Mohd Talmizie Amron, and Nur Hidayah Md Noh. "Assessing the Acceptance of E-Learning via Technology Acceptance Model (TAM)". In: *2021 6th IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE)*. Vol. 6. IEEE. 2021, pp. 1–5.

[35] Justus Bogner and Manuel Merkel. *To Type or Not to Type? A Systematic Comparison of the Software Quality of JavaScript and TypeScript Applications on GitHub*. 2022. arXiv: 2203.11115 [cs.SE].

[36] Dennis Howitt. "Thematic analysis". In: *Introduction to qualitative methods in psychology*. Prentice Hall, 2010, pp. 163–185. ISBN: 978-0-13-206874-1.

[37] Virginia Braun and Victoria Clarke. "Using thematic analysis in psychology". In: *Qualitative research in psychology* 3.2 (2006), pp. 77–101.

[38] Maina Korir, Sharon Slade, Wayne Holmes, Yingfei Héliot, and Bart Rienties. "Investigating the dimensions of students' privacy concern in the collection, use and sharing of data for learning analytics". In: *Computers in Human Behavior Reports* 9 (2023). ISSN: 2451-9588. DOI: https://doi.org/10.1016/j.chbr.2022.100262.

[39] Magnus Bygren. "Biased grades? Changes in grading after a blinding of examinations reform". In: *Assessment & Evaluation in Higher Education* 45.2 (2020), pp. 292–303. DOI: 10.1080/02602938.2019.1638885.